Transportation Research Part C 149 (2023) 104057

Contents lists available at ScienceDirect

Transportation Research Part C

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

Integrated electric bus timetabling and scheduling problem

Xiaoming Xu, Yanhong Yu, Jiancheng Long *

School of Automotive and Transportation Engineering, Hefei University of Technology, Hefei 230009, China

ARTICLE INFO

Keywords:
Timetabling
Vehicle scheduling
Time-space network
Lagrangian relaxation heuristic

ABSTRACT

Vehicle timetabling and scheduling in a public transit system are usually performed separately, with the output of timetabling serving as the input of scheduling. An obvious drawback of this sequential planning method is that the trade-off between bus timetables and vehicle schedules may be neglected when determining solutions, which in turn results in that the obtained solutions may be inferior to those produced using an integrated framework. For example, a well-planned timetable may result in a schedule that requires a large vehicle fleet size with more operational cost, while a well-planned schedule may reduce the quality of a bus timetable by limiting the use of vehicles. In this paper, we introduce a time-space network-based framework for integrating electric bus timetabling and scheduling, with minimum and maximum headway times, depot requirements, deadheading and vehicle battery capacities considerations. The underlying time-space network is constructed with well-designed inventory arcs that represent multiple operations a bus may execute, thus decreasing the network size. Using the constructed network, we formulate the considered problem with a multi-commodity network flow model and develop a Lagrangian relaxation heuristic that consists of three phases, including generating relaxed solutions, making relaxed solutions feasible, and improving feasible solutions, to solve the integrated model. Tests on a set of instances confirm that the proposed integrated solution method can efficiently produce bus timetables and schedules with valid bounds, indicate that the integrated method can produce better solutions where the profit is increased by 5.29%–20.28%, and show how the headway times, service trip profit and operating cost settings affect the solution.

1. Introduction

Transit networks play an important role in cities, especially in metropolitan areas, as they are effective and low-carbon means of alleviating traffic pressure. Advances in electric vehicle technology are enabling the increased use of electric buses in public to reduce greenhouse gas emissions. For example, in the bus system of Hefei city, China, almost all of the commercial vehicles are electric buses which work in daytime and return to assigned depots for charging at night. The transit planning process is usually divided into five sequential planning stages (see, e.g., Ceder and Wilson, 1986): line planning, frequency/headway setting, timetabling, vehicle scheduling and crew rostering. We study an integration optimization problem of the frequency/headway setting, timetabling and vehicle scheduling stages in an electric bus system as they are tightly coupled to one another. For example, a well-planned bus timetable may ensure an efficient public transit service and a well-planned vehicle schedule may reduce the operational cost of the line. However, there is a trade-off between a well-planned bus timetable and a well-planned vehicle schedule. That is, a well-planned timetable may produce a vehicle schedule with a large bus fleet size as the connection between the arrivals and departures of each vehicle at the same bus terminus is usually neglected when determining timetables. A large fleet size increases the operating cost.

* Corresponding author.
E-mail address: jianchenglong@hfut.edu.cn (J. Long).

https://doi.org/10.1016/j.trc.2023.104057
Received 14 July 2022; Received in revised form 6 January 2023; Accepted 8 February 2023
Available online 17 February 2023

Conversely, a well-planned vehicle schedule may reduce the quality of a bus timetable by limiting the vehicle fleet size or minimizing the number of vehicles used. Thus, modeling and solving the timetable generating problem and vehicle scheduling problem (VSP) separately may produce lower-quality solutions than those produced using an integrated optimization framework.

1.1. Related literature

In this section, we review related existing studies in which the studied problems can be divided into three categories: the transit network timetabling (TNT), vehicle scheduling problem and the integration optimization problem of TNT and VSP.

1.1.1. Transit network timetabling

The main purpose of the TNT is to maximize the number of transported passengers while minimizing the operating cost; therefore, passengers' convenience and cost and operators' operating cost and profit are usually taken into consideration when determining transit network timetables. Some studies on the TNT problem consider only decisions regarding departure times from lines' starting stations. For example, de Palma and Lindsey (2001), Castelli et al. (2004) and Li et al. (2010) have studied the TNT problem of coordinating different lines' starting times to minimize the operator cost and/or passenger cost. Mesa et al. (2014) studied the TNT problem of determining each trip's departure times from the original station, taking the vehicle fleet size into account. Gkiotsalitis and Alesiani (2019) considered a robust timetable optimization for bus lines that minimized the possible loss in a worst-case scenario, in which the uncertain nature of travel times and passenger demand was taken into consideration and predetermined planned dispatch times were required. The decision variables were the deviations from each predetermined vehicle dispatch time. Some studies on the TNT problem have generated detailed timetables by determining the arrival and departure times at each visited stop. For example, Zhao and Zeng (2008) analyzed an integrated transit network problem consisting of line planning, headway setting and timetabling problems, in which timetabling decisions at stops were made with random variables after headway decisions were made. Fonseca et al. (2018) studied an integrated timetabling and vehicle scheduling (ITTVS) problem, in which the arrival and departure times at each stop were not based on headway decisions but were instead set as decision variables. Chu et al. (2019) studied a bus timetabling problem considering transfer synchronization, in which bus timetabling decisions, such as the departure times at each stop and the running time between two stops, and passenger choices regarding travel paths were simultaneously optimized.

1.1.2. Vehicle scheduling problem

Given a determined timetable, the VSP aims to assign vehicles covering all trips in the timetable, while satisfying practical constraints such as layover time requirements between trips. In practice, the quality of vehicle schedules, which is usually measured by the fleet size (Ceder, 2007), affects not only the vehicle usage and maintenance costs but also the staffing cost. The simplest version of the VSP is the single-depot single-type VSP (SDVSP), which determines vehicle schedules such that all trips are covered by vehicles departing (returning) from (to) the same location, i.e., the depot (see, e.g., Ibarra-Rojas et al., 2015). An extended version of SDVSP is the multi-depot vehicle scheduling problem (MDVSP), in which vehicles can depart (return) from (to) different locations (see, e.g., Dell'Amico et al., 1993; Oukil et al., 2007; Kulkarni et al., 2018). Some researchers have studied the MDVSP taking fuel consumption into consideration; see, e.g., Haghani and Banihashemi (2002). Zhang et al. (2021) analyzed an electric bus fleet scheduling problem taking battery degradation and the nonlinear charging profile into consideration. In their problem, the charging strategy, which determine the trip chain of a bus and the times when it charges its battery, had to be determined. Some researchers have studied the MDVSP with time windows that determine when the services must begin; see, for example, Desaulniers et al. (1998) and Kliewer et al. (2012). Other researchers have studied the MDVSP taking the vehicle type into consideration; see, e.g., Guedes and Borenstein (2015), Kliewer et al. (2006), and Hassold and Ceder (2014).

1.1.3. Integrated TNT and VSP

Integrating timetabling and vehicle scheduling may enable a favorable trade-off between service quality and operating cost. Some studies addressing the ITTVS problem can be found in the literature, e.g., Ibarra-Rojas et al. (2014), and Ceder (2001) considered the first to examine the ITTVS problem (Fonseca et al., 2018; Carosi et al., 2019). Most studies on the ITTVS problem have used ideal, desired, or predetermined timetables (see, e.g., Guihaire and Hao, 2010; Petersen et al., 2013), in which timetabling decisions are often made by shifting and/or canceling line runs. There are two typical application scenarios for various versions of the ITTVS problem. One application scenario minimizes the negative influence of a disruption, such as a vehicle breakdown, traffic accident, medical emergency, or road work, by allowing trips to be both shifted and canceled when rescheduling vehicles; see, e.g., the vehicle rescheduling problem studied in Li et al. (2007, 2009). The other application scenario is to reduce the operating cost by allowing trips to be shifted and vehicle routes to be rescheduled while limiting the deviation between the original and newly-generated schedules and timetables. For example, Petersen et al. (2013) studied the ITTVS problem of minimizing the vehicle scheduling and passenger transfer costs by developing a large neighborhood search metaheuristic. In their study, sets of metatrips were constructed based on the given original trips, with exactly one trip from each metatrip being covered in the solution. The idea underlying this strategy is that choosing alternative departure timings may reduce passenger-transfer-related waiting times. Ibarra-Rojas et al. (2014) used two integer linear programming models and an ϵ-constraint method to jointly solve the combined timetabling and vehicle scheduling problems.

More recently, Liu and Ceder (2017) solved an ITTVS problem involving transfer path routing and even-headway departure decisions by using a deficit-function-based sequential search method. Laporte et al. (2017) studied an integrated problem comprising

Table 1
Comparison of some closely related studies.

Reference	Infrastr-uture	Objective(s)	Headway restrictions	Original timetable	Mileage/elec-tricity limits	Deadhead	Methodo-logy
Guihaire and Hao (2010)	General network	Minimize weighted service quality and resource utilization	Not considered	Given	Not considered	Allowed	Iterated local search
Petersen et al. (2013)	General network	Obtain a trade-off between passenger service and operating cost	Not considered	Given	Not considered	Allowed	Large neigh-borhood sea-rch
Ibarra-Rojas et al. (2014)	General network	Obtain a trade-off between passenger service and operating cost	Not considered	Given	Not considered	Allowed	ϵ-constraint method
Liu and Ceder (2017)	General network	Minimize passenger in-vehicle, waiting, transfer time, load discrepancy and vehicle size	Maximum	Given	Not considered	Not allowed	Deficit func-tion based heuristic
Laporte et al. (2017)	General network	Minimize user inconvenience, line run cost, and fleet size cost	Not considered	Not given	Not considered	Not allowed	ϵ-constraint method
Fonseca et al. (2018)	General network	Minimize a weighted sum of operating and passenger cost	Minimum maximum	Given	Not considered	Allowed	Matheuristic
Desfontaines and Desauln-iers (2018)	General network	Minimize a weighted sum of operating cost, shifting penalties, headway and timetable deviation	Not considered	Given	Not considered	Allowed	two-phase matheuristic
Tong et al. (2017)	General network	Minimize the number of unserved passengers and routing costs	Not considered	Not given	Not considered	Not consider	Lagrangian relaxation
Carosi et al. (2019)	Single depot	Minimize deviation from ideal timetable, vehicle operating cost	Minimum maximum	Not given	Not considered	Allowed	Diving-type matheuristic
This work	General network	Maximum service profit less operating cost	Minimum maximum	Not given	Considers	Allowed	Lagrangian relaxation

timetabling, vehicle scheduling and user routing problems, in which the number of runs along each line was restricted and deadheading was not considered. Desfontaines and Desaulniers (2018) studied the MDVSP with given targeted timetables, allowing trips to be shifted when determining vehicle routes. Tong et al. (2017) studied a customized bus service design problem for finding optimal stop locations, routes and timetables for each vehicle, wherein passenger demand and vehicle capacity were explicitly considered. They formulated their problem using a multi-commodity network flow optimization model, and developed a Lagrangian relaxation heuristic (LRH) to decompose the integrated problem into a general assignment problem associated with passenger-to-vehicle assignment decisions and a time-dependent shortest path problem associated with vehicle routing decisions. To efficiently solve these problems, they developed a space–time prism-based method solution algorithm to reduce the solution search space. Fonseca et al. (2018) addressed the ITTVS problem tactically, making timetabling decisions by shifting the departure time and increasing the dwell time at intermediate stops to maximize transfer opportunities. They developed a matheuristic to iteratively solve a bi-objective mathematical formulation of the ITTVS problem, wherein at each iteration, timetabling and vehicle scheduling decisions are determined in a preliminary and a secondary step, respectively. Carosi et al. (2019) studied the ITTVS problem of optimally balancing the regularity (i.e., minimizing the deviation from the desired headways) and service provider cost, wherein no timetable was given but desired headways together with deviation cost were provided. They proposed an ingenious multi-commodity flow-type model for the considered ITTVS problem using pure compatibility, pure time-space and mixed-type underlying graph. An effective diving-type matheuristic was developed to solve their model, in which the continuous relaxation could be solved using either a linear programming solver or Lagrangian techniques at each iteration.

Table 1 summarizes the characteristics of studies closely related to ours. For a comprehensive literature review on transit network planning, please refer to Ibarra-Rojas et al. (2015). As concluded in Carosi et al. (2019), almost all the previous studies on the ITTVS problem used meta-heuristics to obtain solutions, save for Carosi et al. (2019) and Fonseca et al. (2018), where matheuristic approaches are developed to solve the studied problems. We use a multi-commodity network flow model on a time-space network and develop an LRH to obtain feasible solutions and valid bounds for the ITTVS problem. Our work differs from previous research in at least two aspects. First, almost all of the previous works, except Laporte et al. (2017) and Carosi et al. (2019), have used ideal, desired, or original timetables, whereas we do not use a predetermined timetable that may increase the solution space and obtain better solutions while increase the computational burden. Second, we consider the battery capacity constraints of buses that has been neglected in most TNT models but has been considered in VSP studies; see, e.g., Haghani and Banihashemi (2002), and Zhang et al. (2021), while battery charge scheduling for electric vehicles is beyond the scope of this paper. The frequency of service (reciprocal of the headway) may be the simplest one to measure the qualities of timetables from the viewpoint of users of the transportation service. Similar to Carosi et al. (2019) and Fonseca et al. (2018), in our work headway times should be determined and are limited to respect minimum headway and maximum headways. This is consistent with operator practice, as it can reduce the work of solving the ITTVS problem. For example, from the given passenger flow data, operators in the study area, such as Hefei Public Transport

Group, use their experience to estimate approximate values of the minimum and maximum headway times at different times. Our study differs from Fonseca et al. (2018) in that valid bounds can be quickly obtained using the LRH, whereas Fonseca et al. (2018) provide valid bounds using the CPLEX solver which requires a computational time of 24 h. Our work differs from Carosi et al. (2019) in that we consider the maximum battery capacity constraint for each individual bus. As addressed in Carosi et al. (2019), their VS subproblem would be a minimum cost network flow problem if it could be solved independently, where some constraints on the vehicle routes, such as those depending on the total time/distance traveled by the vehicle may not be capable of being expressed.

1.2. Contributions and overview of the paper

We make two contributions to the research. First, we develop a time-space network-based framework for integrating TNT and VSP. To the best of our knowledge, our problem is the first that considers minimum and maximum headway times, depot requirements, deadheading and vehicle battery capacities. The underlying time-space network is smaller than the traditional one, as the former is constructed with well-designed inventory arcs that represent multiple operations a bus may execute. Second, we formulate the considered problem with a multi-commodity network flow model and develop a Lagrangian relaxation heuristic that consists of three phases, including generating relaxed solutions, making relaxed solutions feasible, and improving feasible solutions to solve the integrated model. The developed heuristic can efficiently handle different operator-specified headway patterns and the battery capacity limitation of electric buses. A computational study in which our method is applied to various line topologies confirms that the proposed solution method produces bus timetables and vehicle schedules with valid bounds and report the benefits obtained by integrating vehicle timetabling and scheduling decisions.

The rest of this paper is organized as follows. Section 2 presents a detailed description of our integrated bus timetabling and scheduling problem. Section 3 describes the construction of a time-space network with inventory arcs, based on which an integer program for the considered problem is formulated. A Lagrangian relaxation-based solution method is developed in Section 4. Section 5 conducts a computational study to examine the effectiveness of the modeling and solving methods. Some conclusions are made in Section 6.

2. Problem description

Given some lines and bus depots, the problem we study is to simultaneously determine bus timetabling and routing decisions that maximize the operational profit while minimizing the operational cost, such that certain operational requirements are satisfied. Timetabling decisions determine the departure headways between two adjacent service trips along the same line. These departure headways are used to specify the bus timetable, which comprises the starting and ending times of all trips. Routing decisions determine a series of activities for each bus during the considered planning horizon, which can be represented by a time-space trajectory, i.e., a bus route, in the bus system. Along a route, a bus may leave a depot and arrive at a terminus, make a service or deadhead trip, arrive at another terminus where it may wait before departing for the next service or deadhead trip, after which it returns to the depot it had left at the start.

2.1. Input data

Four types of objects are considered in the bus system: depots, termini, bus lines and buses. The input data corresponding to these objects are summarized in Table 2 and explained below in detail. The considered planning horizon is discretized and given by $\{0, 1, \dots, T\}$, in which each time unit is 1 min.

Depot and terminus data: Let $D = \{d_1, d_2, \dots, d_{|D|}\}$ be the set of all considered bus depots, where the buses are parked and battery charging operations are performed overnight. The capacity of each depot is assumed to be unlimited. Let $S = \{s_1, s_2, \dots, s_{|S|}\}$ be the set of all considered bus termini. Each terminus represents the starting or ending point of a bus line. To begin its regular service trips to transport passengers, a bus performs a pull-in action to travel from the depot to the starting point of the line. After completing its assigned trips, the bus performs a pull-out action to travel from the terminus to the same depot it had left to commence its trips. Buses may also perform deadhead trips to reposition themselves. Here, a *service trip* is the journey of a bus along a line, which is similar to the *line run* presented in Mesa et al. (2014) and Laporte et al. (2017), and *deadhead trips* are the journeys a bus makes between termini without carrying any passengers.

Bus line data: Let $L = \{l_1, l_2, \dots, l_{|L|}\}$ be the set of all considered directed bus lines. The input data for each *bus line* (or *line*) l are given by $(p_l, q_l, \tau_l, f_l, g_l, h_l)$. Specifically, each line l has one starting point p_l and one ending point q_l, where $p_l, q_l \in S$. A pair of *opposing lines* is defined as two lines along which buses travel between two termini in opposite directions (see, e.g., Castelli et al., 2004). The time for service trips between the starting and ending points of a line l is always τ_l; i.e., there is exactly one unidirectional line if lines exist between two termini in one direction. The opposing line l' of line l may have a different service trip time. Without loss of generality, we denote φ_l as the ratio of $\tau_{l'}$ to τ_l; i.e., $\varphi_l = \tau_{l'}/\tau_l$. The line operator receives a profit of f_l for every service trip performed for line l (see, e.g., Caprara et al., 2002; Cacchiani et al., 2008). Moreover, to guarantee the service quality of line l, its service trip frequency cannot be less than a predetermined value. The interval between the departure times of two consecutive trips along the same line l must be no greater than g_l time units; this is the maximum departure headway restriction. However, considering the limited passenger demand and safety requirements, the interval between the departure times of two consecutive trips along the same line l must be at least h_l time units; this is the minimum departure headway restriction. In

Table 2
Summary of input data.

Type of data	Notation	Description
Depot and terminus data	D	set of all bus depots, $D = \{d_1, d_2, \dots, d_{\|D\|}\}$
	S	set of all bus termini, $S = \{s_1, s_2, \dots, s_{\|S\|}\}$
Bus line data	L	set of all bus lines, $L = \{l_1, l_2, \dots, l_{\|L\|}\}$
	p_l	starting terminus of bus line l
	q_l	ending terminus of bus line l
	τ_l	travel time for a service trip between line l's starting and ending termini
	f_l	fixed profit achieved by performing a service trip for line l
	g_l	maximum time difference between departures from line l's starting terminus
	h_l	minimum time difference between departures from line l's starting terminus
	φ_l	ratio of $\tau_{l'}$ and τ_l, where l' is the opposing line of line l, i.e., $\varphi_l = \tau_{l'}/\tau_l$
	$L(s)$	set of lines originating at the same terminus $s \in S$
	$\chi(s)$	minimum service trip time among the lines in $L(s)$
Bus data	K	set of all buses ($K = \{k_1, k_2, \dots, k_{\|K\|}\}$)
	$E_{\max}$	maximum battery capacity
	e_k	electricity consumed per unit time for a running bus $k \in K$
	c_f	fix cost of using a bus
	c_d	unit cost of a bus's time spent on deadheading between termimi
	$\alpha_{ss'}$	time for a deadhead trip between termini $s, s' \in S$
	α'_{ds}	time for a pull-in operation between depot $d \in D$ and terminus $s \in S$
	α''_{sd}	time for a pull-out operation between terminus $s \in S$ and depot $d \in D$
	β	minimum dwell time between performing two consecutive bus trips
	ψ_l	ratio of α_{p_l,q_l} to τ_l, i.e., $\psi_l = \alpha_{p_l,q_l}/\tau_l$

addition to these input data, we denote $L(s)$ as the set of lines originating at the same terminus s, and $\chi(s)$ as the minimum service trip time among the lines in $L(s)$.

Bus data: Let $K = \{k_1, k_2, \dots, k_{|K|}\}$ be the set of all the considered buses, which are homogeneous electric buses whose maximum battery capacity is $E_{\max}$, and the electricity consumed per unit time for a running bus k is e_k. Note that a bus is either assigned work or is idle during the considered planning horizon and it is not allowed to charge its battery between two pieces of works. This is consistent with the operator practice in Hefei bus system where most buses are not suitable for quick charging. A fixed cost of c_f is incurred if bus k is assigned work. A cost of c_d is incurred per unit time for a deadhead. The time for a deadhead trip between two termini $s, s' \in S$ is $\alpha_{ss'}$. As a bus usually runs faster when it is deadheading than when performing a service trip, $\alpha_{ss'}$ is no greater than the time of the service trip from termini s to s'. We denote ψ_l as the ratio of α_{p_l,q_l} to τ_l; i.e., $\psi_l = \alpha_{p_l,q_l}/\tau_l$. Similarly, the pull-in (pull-out) time between a depot $d \in D$ (terminus $s \in S$) and terminus $s \in S$ (depot $d \in D$) is given by α'_{ds} (α''_{sd}). Buses may also wait at termini before performing the next service or deadhead trip. For operational safety, each bus must be assigned at least β time units of dwell time between performing two consecutive bus trips.

2.2. Objective and constraints

As mentioned above, the aim of the considered problem is to simultaneously generate a feasible bus timetable and determine bus routes in a planning horizon, such that the total profit is maximized. The total profit can be calculated as the difference between the operational profit of providing bus service trips (e.g., profit from the bus fare) and the operational cost of buses (e.g., the fixed and deadhead costs). Therefore, the objective of our problem is to obtain a solution that optimally balances the operational profits and costs. More specifically, the operational profit can be maximized by maximizing the number of service trips, while minimizing the operational cost by minimizing the number of buses used and the travel time of deadhead trips, for which the following five types of constraints must be satisfied:

- Service constraints: Each service trip can be performed by at most one bus, and each bus can perform at most one service trip at a time. • Headway time constraints: For each pair of adjacent service trips along the same line, the headway time between these trips should be no less than the minimum headway time and no greater than the maximum headway time.
- Break requirements: Each bus must be assigned a dwell time after performing a service trip.
- Depot requirements: After completing its assigned trips, each bus is required to return to the same depot it had left to commence its trips.
- Maximum battery capacity constraints: For each bus, the quantity of electricity consumed in its assigned trips cannot exceed its maximum battery capacity $E_{\max}$.

3. Mathematical formulation

In this section, we formulate the considered problem using a multi-commodity flow model with certain restrictions, in which each commodity represents a bus. The underlying network is an acyclic directed time-space network $G = (V, A)$, where depots and termini with time instants are represented by vertices and service or deadhead trips are represented by (inventory) arcs.

3.1. Time-space network construction

A time-space network with inventory arcs was used by Kulkarni et al. (2018) to model their MDVSP as an inventory formulation, in which a feasible solution must cover all the trips in the input bus timetable. In the time-space network, inventory arcs are constructed to represent a trip from a certain location and time to the end location and time. This inventory arc construction may help reduce the size of the time-space network by reducing the number of vertices and the corresponding arcs, which in turn may help reduce the computational burden. Following Kulkarni et al. (2018), we construct a time-space network with inventory arcs for the problem under consideration. As the timetable in our problem is unknown that should be determined, our time-space network contains all the possible service trips that may be scheduled in the timetable. The "time" dimension of our time-space network G comprises the instants $0, 1, 2, \dots, T$. The "space" dimension of G comprises the vertices $\bar{\rho}_d$ and ρ_d for each depot $d \in D$, which represent the arrival at and departure from depot d, respectively. Each terminus $s \in S$, together with time point $t \in \{0, 1, \dots, T\}$ corresponds to vertices $\bar{\rho}_{st}$ and ρ_{st}, where vertex $\bar{\rho}_{st}$ represents the arrival of a bus at terminus s at time t and vertex ρ_{st} represents the departure of a bus from terminus s at time t. For the common line l sharing the same origin and destination, i.e., $p_l = q_l$, we also construct independent arrival vertices $\bar{\rho}_{p_l,t}$, $\bar{\rho}_{q_l,t}$ and independent departure vertices $\rho_{p_l,t}$, $\rho_{q_l,t}$. Let vertex $\bar{o}$ and vertex $\bar{d}$ be an artificial source and sink, respectively, for the multi-commodity flow; the vertex set V of the time-space network G can then be mathematically written as

$$V = \{\bar{o}, \bar{d}\} \cup \{\rho_i, \bar{\rho}_i \mid i \in D\} \cup \{\rho_{st}, \bar{\rho}_{st} \mid s \in S;\ t = 0, 1, \dots, T\}.$$

The arc set A of G contains several types of arcs. A cost coefficient $c(u, v)$ associated with each arc $u \to v \in A$ represents the cost for a bus to traverse arc $u \to v$. These types of arcs and their associated cost coefficients are described below.

3.1.1. Starting, ending, pull-in, pull-out and non-working arcs

A starting arc connects the dummy source vertex $\bar{o}$ and a departure vertex ρ_d for each $d \in D$. A bus traversing this starting arc represents a bus that is starting its work during the considered planning horizon and incurring a fixed cost c_f; we then have $c(\bar{o}, \rho_d) = c_f$. An ending arc connects an arrival vertex $\bar{\rho}_d$ and the dummy sink vertex $\bar{d}$ for each $d \in D$. A bus traversing this ending arc represents a bus that has completed its work during the considered planning horizon and is incurring no cost; we then have $c(\bar{\rho}_d, \bar{d}) = 0$. Starting arcs and ending arcs can be used to ensure that the operational requirements of buses are satisfied. A pull-in arc $\rho_d \to \rho_{st}$ such that $t \geq \alpha_{ds}$ and a pull-out arc $\rho_{st} \to \bar{\rho}_d$ such that $t \leq T - \alpha_{sd}$ exist for each pair consisting of a depot $d \in D$ and a terminus $s \in S$. A bus traversing a pull-in (pull-out) arc represents a bus that is performing a pull-in (pull-out) action to go from (return to) its depot d to (from) a terminus s, thereby a cost $c_d \cdot \alpha'_{ds}$ ($c_d \cdot \alpha''_{sd}$) is incurred. A non-working arc $\bar{o} \to \bar{d}$ connects the dummy source and sink vertices. A bus traversing this non-working arc represents a bus that is idle (i.e., is not used) during the considered horizon and is incurring no cost; i.e., $c(\bar{o}, \bar{d}) = 0$.

3.1.2. Inventory, deadheading, transfer and waiting arcs

Two types of inventory arcs ensure that buses can perform service trips between depots to transport passengers. One type is the inventory arc $\rho_{st} \to \rho_{s't'}$ for each line $l \in L$ and time instants $t, t' = 0, 1, \dots, T$ such that $s = p_l, s' = q_l$ and $0 \leq t \leq t' = t + \tau_l + \beta \leq T$. The other type is the inventory arc $\rho_{st} \to \bar{d}$ for each line $l \in L$ and time instant $t = 0, 1, \dots, T - \tau_l$ such that $s = p_l$. A bus traversing an inventory arc represents a bus that is performing a service trip for line l in which a profit of f_l is achieved and no cost is incurred. For each $l \in L$ and $t, t' = 0, 1, \dots, T$, if $s = p_l, s' = q_l$, and $0 \leq t < t' = t + \tau_l + \beta \leq T$, we have $c(\rho_{st}, \rho_{s't'}) = -f_l$; otherwise, $c(\rho_{st}, \rho_{s't'}) = +\infty$. For each $l \in L$ and $t = 0, 1, \dots, T - \tau_l$, if $s = p_l$ we have $c(\rho_{st}, \bar{d}) = -f_l$; otherwise, $c(\rho_{st}, \bar{d}) = +\infty$. Our inventory arcs, which are similar to those presented in Kulkarni et al. (2018) may help reduce the scale of the time-space network. For example, the inventory arc $\rho_{st} \to \rho_{s't'}$ represents two bus operations: a service trip between the origin and destination termini and a dwell at the end terminus, that are usually represented by two or more arcs; see, e.g., travel and dwelling arcs in Xu et al. (2021). Another example is a bus traversing the inventory arc $\rho_{st} \to \bar{d}$, which represents a bus that is performing a service trip, after which it may wait at the end terminus for some time before returning to the nearest depot and completing its work at time T. These operations are usually represented by a travel arc, a dwelling arc, many waiting arcs, a pull-out arc and an ending arc in a classical time-space network (see, e.g., Xu et al., 2021). The size of the time-space network can be further reduced by assigning at most one inventory arc in $\{\rho_{p_l,t} \to \rho_{q_l,t'}, \rho_{p_l,t} \to \bar{d}\}$ to each line $l \in L$ and $t \in [0, T]$. To achieve this, for each $l \in L$, let $T_l = \max\{0, T - (\tau_l + \beta) - \min_{s \in S \setminus \{q_l\}} \{\chi(q_l), \alpha_{q_l,s} + \chi(s)\}\}$. By Proposition 1, for each time instant $t = 0, 1, \dots, T_l$, we construct only one inventory arc $\rho_{st} \to \rho_{s't'}$ such that $s = p_l, s' = q_l$ and $0 \leq t < t' = t + \tau_l + \beta \leq T$, and for each time instant $t = T_l + 1, T_l + 2, \dots, T - \tau_l$, we construct only one inventory arc $\rho_{p_l,t} \to \bar{d}$. As the minimum dwell time β is incorporated into these inventory arcs, the requirement that each bus must be assigned at least β time units of dwell time between two consecutive trips is implicitly satisfied.

Deadheading arcs are constructed to ensure that buses can travel between different termini to reposition themselves. In our time-space network, a deadheading arc $\rho_{st} \to \bar{\rho}_{s't'}$ exists for each pair of termini $s, s' \in S$ such that $s \neq s'$ and $t' - t = \alpha_{ss'}$. A bus traversing a deadheading arc represents a bus that is performing a deadhead trip to reposition itself from terminus s to terminus s'

Table 3
Summary of arc types.

Type of arcs	Form	Domain	Situation	Cost
Starting arcs	$\bar{o} \to \rho_d$	$d \in D$	A bus starts its work at depot d	c_f
Ending arcs	$\bar{\rho}_d \to \bar{d}$	$d \in D$	A bus completes its work at depot d	0
Pull-in arcs	$\rho_d \to \rho_{st}$	$d \in D;\ s \in S;\ t = 0, 1, \ldots, T$	A bus performs a pull-in action	$c_d \cdot \alpha'_{ds}$
Pull-out arcs	$\rho_{st} \to \bar{\rho}_d$	$d \in D;\ s \in S;\ t = 0, 1, \ldots, T$	A bus performs a pull-out action	$c_d \cdot \alpha''_{sd}$
Non-working arcs	$\bar{o} \to \bar{d}$	/	A bus is not assigned any work	0
Inventory arc	$\rho_{p_l t} \to \rho_{q_l t'}$	$l \in L;\ t, t' = 0, 1, \ldots, T_l$ s.t. $0 \le t \le t' = t + \tau_l + \beta \le T$,	A bus performs a service trip for line l and a dwell operation, $T_l = \max\{0, T - (\tau_l + \beta) - \min\{\chi(q_l), \alpha_{q_l,s} + \chi(s)\}\}$	$-f_l$
	$\rho_{p_l t} \to \bar{d}$	$l \in L;\ t = T_l + 1, T_l + 2, \ldots, T - \tau_l$		
Deadheading arcs	$\rho_{st} \to \bar{\rho}_{s't'}$	$s, s' \in S;\ t, t' = 0, 1, \ldots, T$ s.t. $s \ne s';\ t' - t = \alpha_{ss'}$	A bus performs a deadhead trip to reposition itself	$c_d \cdot \alpha_{ss'}$
Transfer arcs	$\bar{\rho}_{st} \to \rho_{st}$	$s \in S;\ t, t' = 0, 1, \ldots, T$	A bus change its status from arrival to departure	0
Waiting arcs	$\rho_{st} \to \bar{\rho}_{s,t+1}$	$t = 0, 1, \ldots, T$ s.t. $t + 1 \le \min\{T, \max_{\{l \in L \text{ s.t. } q_l = s\}}\{T_l + (\tau_l + \beta)\}\}$	A bus is waiting at terminus s	0

in which a cost $c_d \cdot \alpha_{ss'}$ is incurred and no profit is achieved, as the bus does not transport passengers. Thus, for each pair of termini $s, s' \in S$ and $t, t' = 0, 1, \ldots, T$, if $s \ne s'$ and $0 \le t < t' = t + \alpha_{ss'} \le T$, we have $c(\rho_{st}, \bar{\rho}_{s't'}) = c_d \cdot \alpha_{ss'}$; otherwise, $c(\rho_{st}, \bar{\rho}_{s't'}) = +\infty$. A transfer arc $\bar{\rho}_{st} \to \rho_{st}$ follows each deadheading arc. A bus traversing this arc represents a bus whose status changing from arrival to departure. No cost is incurred and no profit is achieved; i.e., $c(\bar{\rho}_{st}, \rho_{st}) = 0$. Moreover, for each terminus $s \in S$ and each time instant $t = 0, 1, 2, \ldots, \min\{T, \max_{\{l \in L \text{ s.t. } q_l = s\}}\{T_l + (\tau_l + \beta)\}\} - 1$, a waiting arc connects two departure vertices ρ_{st} and $\rho_{s,t+1}$. A bus traversing a waiting arc represents a bus that is waiting at terminus s for 1 min without incurring any cost; i.e., $c(\rho_{st}, \rho_{s,t+1}) = 0$.

Table 3 summarizes these types of arcs, in which the "Form" column indicates the representation of the considered arcs, the "Domain" column displays the domains of indexes used in these arc representations, the "Situation" and "Cost" columns, respectively, present the situation and the cost $c(u, v)$ of a bus traversing the arc $u \to v$. In addition to the starting, ending, pull-in, pull-out, non-working and inventory arcs of the second type, each arc of network G takes the following forms: $u \to v \in A$, where the time index of v is greater than the time index of u; see, e.g., inventory arcs of the first type, waiting arcs, and deadheading arcs. Therefore, the time-space network G is acyclic when all time-related parameters are positive.

Proposition 1. *For each line l, a bus cannot perform another bus service trip after performing a bus service trip ℓ of line l in case of trip ℓ's departure time is later than* $\max\{0, T - (\tau_l + \beta) - \min_{s \in S \setminus \{q_l\}}\{\chi(q_l), \alpha_{q_l,s} + \chi(s)\}\}$.

Proof. In case of a bus performs a bus service of line l, we consider the following two situations: (i) the bus performs a bus service of another bus line l' whose origin terminus is l's destination terminus; and (ii) the bus first repositions itself to another terminus s ($s \ne p_l$ and $s \ne q_l$) and then serves another bus line l' whose origin terminus is s. It is easy to calculate that the minimum required operational times for the first and second situation are, respectively, $\chi(q_l)$ and $\alpha_{q_l,s} + \chi(s)$. Combine situations (i), (ii) and the corresponding minimum required operational times, we can conclude that the bus cannot perform another bus service after time instant $\max\{0, T - \min_{s \in S \setminus \{q_l\}}\{\chi(q_l), \alpha_{q_l,s} + \chi(s)\}\}$. Furthermore, consider the minimum break time of β and bus service trip time of τ_l along line l, it is easy to deduce that if the departure time of a bus service trip ℓ of line l is later than time instant $\max\{0, T - (\tau_l + \beta) - \min_{s \in S \setminus \{q_l\}}\{\chi(q_l), \alpha_{q_l,s} + \chi(s)\}\}$, a bus cannot perform another bus service trip after performing bus service trip ℓ. This completes the proof. □

3.2. Constraints

For each bus $k \in K$, a path from vertex $\bar{o}$ to vertex $\bar{d}$ in this time-space network corresponds to a bus route. The route from $\bar{o}$ to $\bar{d}$, except for the route consisting of the non-working arc $\bar{o} \to \bar{d}$, satisfies the break constraints discussed in Section 2.2 if any inventory arc is traversed. Given the constructed network G, we aim to determine one unit of flow for each of the $|K|$ commodities from vertex $\bar{o}$ to vertex $\bar{d}$ to construct routes for all buses, such that the maximum total profit is achieved. However, a flow in this network may not satisfy the departure headway constraints discussed in Section 2.2. Hence, in addition to the standard network flow constraints, such as flow balance constraints, supply and demand constraints and siding constraints for an individual path, (e.g., the same depot restriction and maximum battery capacity constraint), our network flow model also includes the following departure headway constraints for various flows.

Minimum (maximum) departure headway time constraints: for each pair of adjacent trips of line $l \in L$ in the determined timetable, the headway time between the trips should be greater (less) than the minimum (maximum) headway time h_l (g_l). Hence, for each line $l \in L$ and each $t_1 = 0, 1, 2, \ldots, T - h_l + 1$, no more than one bus (at least one bus) can depart from p_l during the time

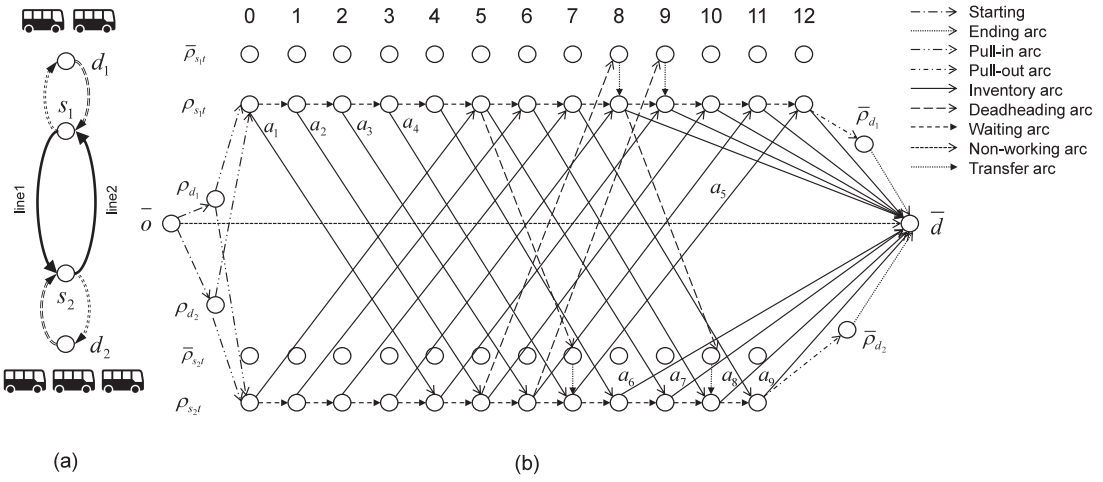


Fig. 1. Line topology and the time-space network in the example.

interval $[t_1, t_1 + h_l - 1]$ ($[t_1, t_1 + g_l - 1]$). Thus, we impose the constraint that the total flow along the inventory arcs in the arc subset

$$C^1_{lt_1} = A \cap \Big[\{\rho_{p_l,t} \to \rho_{q_l,t'} \mid t, t' = 0, 1, \ldots, T;\ t_1 \le\ t \le t_1 + h_l - 1,\ \text{s.t. } t' = t + \tau_l + \beta\} \cup \{\rho_{p_l,t} \to \bar{d} \mid t_1 < t \le t_1 + h_l - 1\} \Big]$$

must be at most 1. We also impose the constraint that the total flow along the inventory arcs in the arc subset

$$C^2_{lt_1} = A \cap \Big[\{\rho_{p_l,t} \to \rho_{q_l,t'} \mid t, t' = 0, 1, \ldots, T;\ t_1 \le\ t \le t_1 + g_l - 1,\ \text{s.t. } t' = t + \tau_l + \beta\} \cup \{\rho_{p_l,t} \to \bar{d} \mid t_1 < t \le t_1 + g_l - 1\} \Big]$$

must be at least 1. Let

$$C_1 = \{C^1_{lt} \mid l \in L;\ t = 0, 1, 2, \ldots, T - h_l + 1\} \text{ and } C_2 = \{C^2_{lt} \mid l \in L;\ t = 0, 1, 2, \ldots, T - g_l + 1\}.$$

The total bus flow along arcs in any arc set $C \in C_1$ must then be no greater than 1, and the total bus flow along arcs in any arc set $C \in C_2$ must then be at least 1.

3.3. An example

Consider a line topology with two depots $\{d_1, d_2\}$, two termini $\{s_1, s_2\}$, two bus lines $\{l_1, l_2\}$ and five buses, as depicted in Fig. 1(a). The service trip times along l_1 and l_2 are, respectively, 3 and 4 min. The deadhead times of $\alpha_{s_1 s_2}$ from s_1 to s_2 and of $\alpha_{s_2 s_1}$ from s_2 to s_1 are 2 and 3 min, respectively. Without loss of generality, the deadhead times between the depots and termini are set as $\alpha'_{d_1,s_1} = \alpha''_{s_1,d_1} = 0$, $\alpha'_{d_2,s_2} = \alpha''_{s_2,d_2} = 0$, $\alpha'_{d_1,s_2} = \alpha''_{s_1,d_2} = \alpha_{s_1,s_2}$, and $\alpha'_{d_2,s_1} = \alpha''_{s_2,d_1} = \alpha_{s_2,s_1}$. In addition, the minimum dwell time β is set to 1 min, and the considered planning horizon is set to $[0, 15]$; i.e., $T = 15$.

Fig. 1(b) depicts a partial time-space network corresponding the line topology presented in Fig. 1(a), where various vertices and types of arcs are shown. In this time-space network, the maximum allowed departure time for bus service trips along lines l_1 and l_2 are, respectively, $T - \tau_{l_1} = 15 - 3 = 12$ and $T - \tau_{l_2} = 15 - 4 = 11$. For line l_2, by Proposition 1, after time 7 $\big(= \big[T - (\tau_{l_2} + \beta) - \min\{\chi(s_1), \alpha_{s_1,s_2} + \chi(s_2)\}\big] = \big[15 - (4 + 1) - \min\{3, 2 + 4\}\big]\big)$, only inventory arcs of the form $\rho_{s_2 t} \to \bar{d}$ exist; see, e.g., inventory arcs a_6, a_7, a_8, a_9 in Fig. 1(b). Also, for time $t = 0, 1, \ldots, 7$ only inventory arcs of the form $\rho_{s_2 t} \to \rho_{s_1 t'}$ exist. In addition, after time 12, there is no waiting arc corresponding to terminus s_1, as inventory arcs a_6, a_7, a_8 and a_9 can represent these waiting arcs as discussed in Section 3.1.2. Similarly, for line l_1, for time $t = 0, 1, \ldots, 7$ only inventory arcs of the form of $\rho_{s_1 t} \to \rho_{s_2 t'}$ exist, after time 7 only inventory arcs of the form of $\rho_{s_1 t} \to \bar{d}$ exist, and there is no waiting arc at terminus s_2 after time 11.

With the constructed time-space network, we are now able to describe the solution using a set of bus routes. Suppose the vehicle battery capacity is large enough, one feasible solution of the example can be represented by the five bus routes depicted in Fig. 2, where we can see that each bus returns to the same depot it had left to commence its trips, i.e., the depot requirements are satisfied. Moreover, for the minimum and maximum headway times as 2 and 3 min, respectively, we can construct arc sets $C^1_{l_1,0} = \{a_1, a_2\}$, $C^1_{l_2,5} = \{a_5, a_6\}$, $C^2_{l_1,0} = \{a_1, a_2, a_3\}$, $C^2_{l_2,5} = \{a_5, a_6, a_7\}$ and so on. In the solution, a bus k_1 may traverse a path p_1 consisting of starting arc $\bar{o} \to \rho_{d_1}$, pull-in arc $\rho_{d_1} \to \rho_{s_1,0}$, inventory arc a_1, four waiting arcs, and inventory arc a_7. Another bus k_2 may traverse a path p_2 consisting of starting arc $\bar{o} \to \rho_{d_1}$, pull-in arc $\rho_{d_1} \to \rho_{s_1,0}$, three waiting arcs, inventory arc a_4, four waiting arcs, and inventory arc a_9. It can be seen that both the minimum and maximum headway times between paths p_1 and p_2 satisfy the requirements; these two paths thereby are compatible. Further, it can be observed that in current solution there are 10 service trips with 8 headway

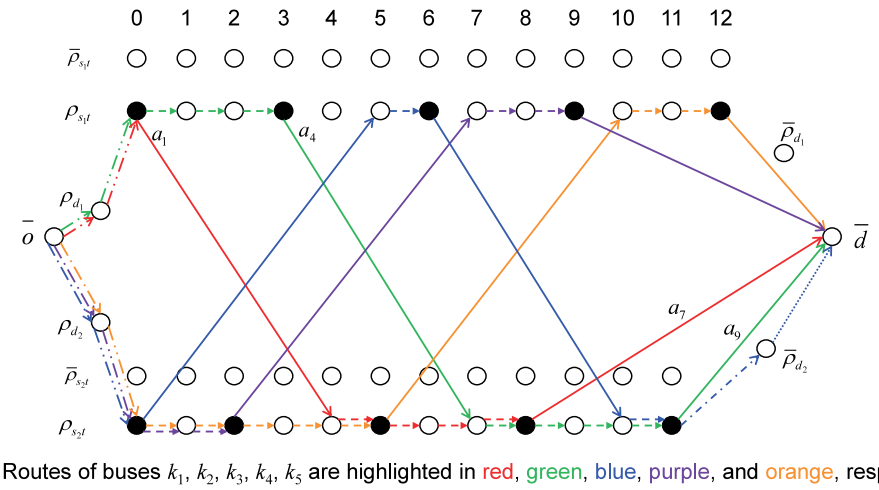


Fig. 2. Routes of buses on the time-space network.

times, while 7 headway times are the same 3 min. This implies that to achieve more profit one can insert more service trips into current solution by scheduling better vehicle routes using more buses.

Note that although set C_2 is specified using the time intervals $t = 0, 1, \ldots, T - g_l + 1$, the generated set C_2 may not cover all the corner cases at the end of the planning horizon. This gap in coverage occurs because impossible inventory arcs are not constructed for the end of the time horizon, and so there is no arc set C^2_{lt} for line $l \in L$ and time interval t at the end of the planning horizon, where the maximum headway constraint need not to be considered. In addition, the set C_2 covers all the corner cases at the beginning of the planning horizon. For example, arc sets $C^2_{l_1,0} = \{a_1, a_2, a_3\}$ and $C^2_{l_1,1} = \{a_2, a_3, a_4\}$ in set C_2 cover the corner cases at time intervals 0 and 1. However, in set C_2, there is no arc set corresponding to the end of the time horizon $[0, 15]$, e.g., time intervals 14 and 15.

3.4. Integer programming formulation

For each bus $k \in K$ and arc $u \to v \in A$, define decision $x^k_{uv} = 1$ if arc $u \to v$ is traversed by bus k in the determined solution; otherwise, let $x^k_{uv} = 0$. The multi-commodity network flow problem with the above restrictions can then be formulated as the following integer program.

$$\mathbf{P}_{\max}: \quad \text{Maximize} \quad \sum_{k\in K}\sum_{u\to v\in A} -c(u,v)x^k_{uv} \tag{1}$$

$$\text{subject to} \quad \sum_{\{v:\bar{o}\to v\in A\}} x^k_{\bar{o}v} = 1, \qquad \text{for all } k \in K \tag{2}$$

$$\sum_{\{u:u\to \bar{d}\in A\}} x^k_{u\bar{d}} = 1, \qquad \text{for all } k \in K \tag{3}$$

$$\sum_{\{v:u\to v\in A\}} x^k_{uv} = \sum_{\{v:v\to u\in A\}} x^k_{vu}, \qquad \text{for all } k \in K, u \in N \setminus \{\bar{o}, \bar{d}\} \tag{4}$$

$$\sum_{\{u\to v\in A^+_d\}} x^k_{uv} = \sum_{\{u\to v\in A^-_d\}} x^k_{uv}, \qquad \text{for all } k \in K, d \in D \tag{5}$$

$$\sum_{u\to v\in A} e_{uv}x^k_{uv} \le E_{\max}, \qquad \text{for all } k \in K \tag{6}$$

$$\sum_{k\in K}\sum_{u\to v\in C} x^k_{uv} \le 1, \qquad \text{for all } C \in C_1 \tag{7}$$

$$\sum_{k\in K}\sum_{u\to v\in C} x^k_{uv} \ge 1, \qquad \text{for all } C \in C_2 \tag{8}$$

$$x^k_{uv} \in \{0,1\}, \qquad \text{for all } k \in K, u \to v \in A \tag{9}$$

The objective function (1) is the total profit of the solution, which is the difference between total profit achieved from assigning bus services to buses and the operational costs derived from the fixed and deadheading costs. Constraints (2) require the total flow of each bus emanating from vertex $\bar{o}$ to be 1, and constraints (3) require the total flow of each bus entering vertex $\bar{d}$ to be 1. Constraints (4) are flow balance constraints for buses. Constraints (5) require that a bus returns to the same depot it left to begin its trips, where A^+_d denotes the set of all pull-in arcs corresponding to depot d (e.g., pull-in arc $\rho_{d_1} \to \rho_{s_1,0}$ in Fig. 1) and A^-_d denotes the set of all pull-out and inventory arcs corresponding to depot d (e.g., pull-out arc $\rho_{s_1,12} \to \bar{\rho}_{d_1}$ and inventory arcs a_6, a_7, a_8 and a_9 in Fig. 1). Constraints (6) require that the electricity consumed in a bus route is no greater than the maximum battery capacity of the bus, where e_{uv} denotes the quantity of electricity consumed on arc $u \to v$. Constraints (7) and (8), respectively, cover all minimum and maximum headway time constraints presented in Section 3.2. As the minimum cost multi-commodity network flow problem is more widely researched in the literature, in the following, we reformulate problem $\mathbf{P}_{\max}$ as problem $\mathbf{P}_{\min}$ for the convenience

of presentation.

$$\mathbf{P}_{\min}: \quad \text{Minimize} \quad \sum_{k\in K}\sum_{u\to v\in A} c(u,v)x_{uv}^{k}$$
$$\text{subject to} \quad \text{constraints (2)–(9).}$$

4. Lagrangian relaxation heuristic

In this section, we present a Lagrangian relaxation heuristic to solve the proposed model $\mathbf{P}_{\min}$. Lagrangian relaxation heuristic has a nice property which permits us to decompose a complex problem into a series easy-solved problems, and it has been widely used to solve bus/vehicle routing and timetabling problems, see, e.g., Carosi et al. (2019), Xu et al. (2018), and Lu et al. (2022).

4.1. Lagrangian relaxation

We use the Lagrangian method to relax constraints (7) and (8) of problem $\mathbf{P}_{\min}$, and denote $\lambda_C \geq 0$ ($C \in C_1$) and $\mu_C \geq 0$ ($C \in C_2$) as their respective Lagrangian multipliers. Incorporating these relaxed constraints into the objective function with Lagrangian multipliers λ and μ, we obtain the Lagrangian relaxed problem $\mathbf{P}(\lambda,\mu)$ (see below), where λ and μ are the vectors of the λ_C and μ_C values, respectively.

$$\begin{aligned}\mathbf{P}_{\min}(\lambda,\mu): \text{Minimize} \quad & \sum_{k\in K}\sum_{u\to v\in A} c(u,v)x_{uv}^{k} + \sum_{C\in C_1}\lambda_C\Big(\sum_{k\in K}\sum_{u\to v\in C} x_{uv}^{k} - 1\Big) \\ & + \sum_{C\in C_2}\mu_C\Big(1 - \sum_{k\in K}\sum_{u\to v\in C} x_{uv}^{k}\Big) \\ \text{subject to} \quad & \text{constraints (2)–(6) and (9).}\end{aligned}$$

As all relaxed constraints are related only to inventory arcs, we let A_{inv} denote the set of all inventory arcs in our time-space network G.

After removing the constant $\sum_{C\in C_2}\mu_C - \sum_{C\in C_1}\lambda_C$ from the objective function of $\mathbf{P}(\lambda,\mu)$, this Lagrangian relaxation problem can be decomposed into $|K|$ independent subproblems. The subproblem corresponding to each $k \in K$ is as follows:

$$\begin{aligned}\mathbf{P}_k(\lambda,\mu): \text{Minimize} \quad & \sum_{u\to v\in A} c(u,v)x_{uv}^{k} + \sum_{C\in C_1}\sum_{u\to v\in C}\lambda_C x_{uv}^{k} - \sum_{C\in C_2}\sum_{u\to v\in C}\mu_C x_{uv}^{k} \\ \text{subject to} \quad & \sum_{\{v:\bar{o}\to v\in A\}} x_{\bar{o}v}^{k} = 1 \\ & \sum_{\{u:u\to \bar{d}\in A\}} x_{u\bar{d}}^{k} = 1 \\ & \sum_{\{v:u\to v\in A\}} x_{uv}^{k} = \sum_{\{v:v\to u\in A\}} x_{uv}^{k}, && \text{for all } u \in N\setminus\{\bar{o},\bar{d}\} \\ & \sum_{\{u\to v\in A_d^{+}\}} x_{uv}^{k} = \sum_{\{u\to v\in A_d^{-}\}} x_{uv}^{k}, && \text{for all } d \in D \\ & \sum_{\{u\to v\in A\}} e_{uv}x_{uv}^{k} \leq E_{\max}, \\ & x_{uv}^{k} \in \{0,1\}, && \text{for all } u\to v \in A\end{aligned}$$

Each subproblem $\mathbf{P}_k(\lambda,\mu)$ is a constrained shortest path problem with revised arc lengths δ_{uv} which are determined as follows:

$$\delta_{uv} = \begin{cases} c(u,v) + \sum_{\{C\in C_1: u\to v\in C\}}\lambda_C - \sum_{\{C\in C_2: u\to v\in C\}}\mu_C, & \text{if } u\to v \in A_{inv}; \\ c(u,v), & \text{otherwise.}\end{cases}$$

We then develop a relaxed solution heuristic using a labeling algorithm to obtain the constrained shortest path.

Given any vectors λ and μ, a lower bound on the optimal objective value of problem $\mathbf{P}$ can be obtained by solving the relaxed problem $\mathbf{P}_{\min}(\lambda,\mu)$. A tight lower bound can be obtained by solving the optimization problem

$$\max_{\lambda,\mu} \mathbf{P}_{\min}(\lambda,\mu),$$

which is referred to as the Lagrangian multiplier problem associated with the original optimization $\mathbf{P}_{\min}$ (see, e.g., Ahuja et al., 1993, p. 606), where the subgradient optimization procedure presented in Section 4.4 is used to find the optimal Lagrangian multipliers. The solution of problem $\mathbf{P}_{\min}(\lambda,\mu)$ enables the upper bound of problem $\mathbf{P}_{\max}$ to be obtained. If these buses are considered homogeneous, the $|K|$ subproblems are identical. We then need to solve only one of them when we determine the lower bound on the optimal objective value of problem $\mathbf{P}_{\min}$, which is equal to $\big($the length of the shortest path $\times|K| + \sum_{C\in C_2}\lambda_C - \sum_{C\in C_1}\mu_C\big)$.

To obtain tighter lower bounds and better feasible solutions, we adopt a subgradient optimization procedure to find the near-optimal Lagrangian multipliers, as shown in Section 4.4. At each iteration of the subgradient optimization procedure, the following steps may be executed. First, determining the lower bound of $\mathbf{P}_{\min}$ by solving the relaxed problem with the developed relaxed solution heuristic that is presented in Section 4.2. Second, determining the upper bound by the feasible solution heuristic that is presented in Section 4.3. Third, updating Lagrangian multipliers using the method in Section 4.4 with the obtained lower and upper bounds. Algorithm 1 summarizes the entire Lagrangian heuristic, where we execute these steps iteratively, until the number of iterations hits a prespecified limit $MaxIter$. To save the computational time, we may execute the feasible solution heuristic with different probabilities at different iterations, as the feasible solution heuristic is time-consuming and it will be less likely to be able to obtain a better upper bound solution at later iterations. Moreover, we reduce the value of step size θ if the incumbent best lower bound has not been improved for certain consecutive iterations.

Algorithm 1 Lagrangian relaxation heuristic

Input: System data, initialize iteration number $i = 1$, optimality gap $gap = 100\%$, the best upper bound $UB_{\min} = +\infty$, best lower bound $LB_{\min} = -\infty$, maximum iteration number $MaxIter$, step size θ
while $i < MaxIter$ **do**
  Step 1: Use the *relaxed solution heuristic* (see Algorithm 2 in Section 4.2) to obtain lower bound lb, let $LB_{\min} \leftarrow \max\{lb, LB_{\min}\}$, reduce θ if $LB_{\min}$ is not improved for certain iterations
  Step 2: Use the *feasible solution heuristic* (see Algorithm 3 in Section 4.3) to obtain upper bound ub with different probabilities, let $UB_{\min} \leftarrow \min\{ub, UB_{\min}\}$
  Step 3: Update Lagrangian multipliers (see Section 4.4), let $gap \leftarrow \frac{UB_{\min} - LB_{\min}}{LB_{\min}} \times 100\%$, $i \leftarrow i + 1$
end while
Output: $LB_{\min}$, $UB_{\min}$ and the best feasible solution

4.2. Relaxed solution heuristic

Labeling algorithms have been widely used to find constrained shortest paths; see, e.g., Tong et al. (2017), Righini and Salani (2008), Lozano et al. (2016) and Xu et al. (2018). Our relaxed solution heuristic (see Algorithm 2) also includes a labeling algorithm in which for each $v \in V$, we introduce a two-dimensional label (L_v, E_v), where L_v denotes the length of a path from vertex $\bar{o}$ to v, and E_v denotes the electricity consumed along this path. We first initialize a label set $\mathcal{L} = \{(L_{\bar{o}}, E_{\bar{d}})\} = \{(0, 0)\}$ and denote the shortest path length by $L_{\min} = +\infty$. For each depot $d \in D$, we first set the lengths of all pull-in, pull-out and inventory arcs equal to those in $\{\delta_{uv} \mid u \to v \in A\}$ and temporarily set the lengths of all pull-in, pull-out and inventory arcs corresponding to depot $d' \in D \setminus \{d\}$ to $+\infty$, which implicity satisfies the depot constraints. We then execute the while loop in Algorithm 2. In each loop, we consider label $(L_u, E_u) = \arg\min\{L_u + \tilde{L}_u \mid (L_u, E_u) \in \mathcal{L}\}$, where $\tilde{L}_u$ is the length of the shortest path (with arc length $\{\delta_{uv} \mid u \to v \in A\}$) from u to $\bar{d}$, if $u = \bar{d}$, we update $L_{\min} = \min\{L_v, L_{\min}\}$ and break the while-loop. Otherwise, if the length of arc $u \to v$ is finite, we extend to vertex v the label (L_v, E_v), where $L_v = L_u + c_{uv}$ and $E_v = E_u + e_{uv}$. We can then obtain new labels for all vertices in u's succeeding vertex set $Succ(u)$.

Algorithm 2 Relaxed solution heuristic

Input: Network G, arc length $\{\delta_{uv} \mid u \to v \in A\}$
  initialize label set $\mathcal{L} = \{(L_{\bar{o}}, E_{\bar{o}}) = (0, 0)\}$, shortest path length $L_{\min} = +\infty$
for $d \in D$ **do**
  set the lengths of all pull-in, pull-out and inventory arcs as the same as that in $\{\delta_{uv} \mid u \to v \in A\}$, while reset the lengths of all pull-in, pull-out and inventory arcs corresponding to depot $d' \in D \setminus \{d\}$ as $+\infty$
  while label set $\mathcal{L}$ is not empty **do**
    select the label $(L_u, E_u) = \arg\min\{L_u + \tilde{L}_u \mid (L_u, E_u) \in \mathcal{L}\}$
    if $v = \bar{d}$ **then**
      update $L_{\min} = \min\{L_v, L_{\min}\}$; break;
    else
      generate $(L_v, E_v) = \{L_u + c_{uv}, E_u + e_{uv}\}$ to each v in vertex u's succeeding vertex set $Succ(u)$
      for each new generated label (L_v, E_v) for $v \in Succ(u)$:
        (1) if $L_v + \tilde{L}_v > L_{\min}$ or $E_v + \tilde{E}_v > E_{\max}$ or (L_v, E_v) is dominated by other labels, then prune label (L_v, E_v); otherwise, add label (L_v, E_v) into label set $\mathcal{L}$
        (2) if some labels in $\mathcal{L}$ are dominated by (L_v, E_v), then remove these labels from $\mathcal{L}$
    end if
  end while
end for
Output: shortest path length $L_{\min}$

To speed up the labeling algorithm, two dominance pruning strategies are developed to handle the scale of the extended labels according to the characteristics of our constrained shortest path problem. In the first pruning strategy, let (L_v^1, E_v^1) and (L_v^2, E_v^2) be two labels associated with the same vertex $v \in V$. Label (L_v^1, E_v^1) can be said to dominate label (L_v^2, E_v^2) if the following conditions hold: (i) $L_v^1 \le L_v^2$ and (ii) $E_v^1 \le E_v^2$, in which (L_v^2, E_v^2) would thereby be pruned. In the second pruning strategy, for each $v \in V$, label (L_v, E_v) would be pruned if $E_v + \tilde{E}_v > E_{\max}$ or $L_v + \tilde{L}_v > L_{\min}$, where $\tilde{E}_v$ is the minimum required quantity of electricity consumed

among the paths from v to $\bar{d}$. Each newly generated label will be added to label set $\mathcal{L}$ if it is not pruned. Similarly, some labels in set $\mathcal{L}$ will also be removed from $\mathcal{L}$ if they are dominated by the newly generated labels.

4.3. Feasible solution heuristic

We now present an upper bound heuristic that generates feasible solutions to the problem of determining the upper bound of $\mathbf{P}_{\min}$, where the Lagrangian dual information derived from the nonnegative vectors λ and μ used in the lower bound solution method is adopted. The lower bound of problem $\mathbf{P}_{\max}$ can be derived using the feasible solution to problem $\mathbf{P}_{\min}$. The heuristic has three phases: the first phase constructs a bus schedule and a bus timetable satisfying the minimum departure headway constraints (7), the second phase attempts to make the aforementioned relaxed solution feasible by inserting more bus services satisfying the maximum headway time constraints (8) and the third phase aims to improve the quality of the obtained feasible bus timetable. We describe these three phases in detail below and summarize this heuristic in Algorithm 3.

Algorithm 3 Feasible solution heuristic

Input: Network G, arc length $\{\delta_{uv} \mid u \to v \in A\}$
Phase 1: Generate relaxed solution
for $k := k_1, k_2, \ldots, k_{|K|}$ **do**
    apply the relaxation solution heuristic to construct a route for k with arc length δ^k_{uv}, see equation (10), where the routes of bus determined before k remain unchanged
end for
Phase 2: Make relaxed solution feasible
for each pair of scheduled adjacent trips m, n for the same line $l \in L$ such that the time of $t_m + g_l$ (t_m is trip m's departure time) is earlier than n's departure time t_n, i.e., $t_m + g_l < t_n$ **do**
    if $t_m + h_l \le t_n - h_l$ **then**
        execute the *insert* (see Algorithm 4) operation
    else if $t_m + h_l > t_n - h_l$ **then**
        execute the *shift* (see Algorithm 5) operation
    end if
end for
Phase 3: Improve feasible solution
for each $k \in K$ that has been assigned to work **do**
    if there exists a bus route $r_{k'}$ that can be performed by bus k after completing its route r_k **then**
        combine routes r_k and $r_{k'}$, and make bus k' idle by updating $K \leftarrow K \setminus \{k'\}$
    end if
end for
Output: A feasible solution or a sufficient big cost in case of no feasible solution is found

Phase 1: We construct routes for buses one by one in a predetermined sequence (e.g., bus ID), where each route corresponds to a path from vertex $\bar{o}$ to vertex $\bar{d}$ in network G. When we construct bus k's route, the routes determined before the construction remain unchanged. To ensure that the route for bus k does not violate the minimum departure headway time constraints (7) of problem $\mathbf{P}_{\min}$, we need to prevent its corresponding path in network G from traversing arcs that are incompatible with the existing paths. To this end, using the existing paths, we construct a set $\tilde{C}_1$ to cover all possible incompatible arc sets, in which each incompatible inventory arc set $C^1_{lt} \in \tilde{C}_1$ is the same as that defined in Section 3.2. Specifically, for each l, if there exists a scheduled bus service trip with departure time of t, we add the incompatible inventory arc set C^1_{lt} to set $\tilde{C}_1$. To ensure that no arc in $\tilde{C}^1_l$ can be traversed by bus k, we revise the arc lengths in the set to $+\infty$. Letting δ^k_{uv} denote the revised arc length of $u \to v \in A$, we set

$$\delta^k_{uv} = \begin{cases} +\infty, & \text{if } u \to v \in C, \text{ for each } C \in \tilde{C}_1; \\ \delta_{uv}, & \text{otherwise.} \end{cases}$$

Using the arc lengths $\{\delta^k_{uv} \mid u \to v \in A\}$, we construct a route for bus k by obtaining the shortest path from vertex $\bar{o}$ to vertex $\bar{d}$ in network G using a dynamic programming algorithm. The collection of the paths (or routes) constructed in Phase 1 for all the buses forms an initial relaxed solution to problem $\mathbf{P}_{\min}$ that satisfies the minimum departure headway constraints; see, e.g., constraints (7), which however may not satisfy the maximum departure headway constraints; see, e.g., constraints (8). For each line $l \in L$, let C_{lt} be the arc set of the inventory arcs whose departure time is in $[t - g_l, t]$ ($t \in \{g_l, T - \tau_l\}$). To ensure that a bus service trip is generated during the first (last) g_l time instants, at the beginning of Phase 1 we set the length of the inventory arc in C_{l,g_l} ($C_{l,T-\tau_l}$) to a negative value that is small enough to ensure a bus traverses these arcs. Once an inventory arc in C_{l,g_l} ($C_{l,T-\tau_l}$) is selected, we reset the length of the inventory arc in C_{l,g_l} (respectively $C_{l,T-\tau_l}$) to its original arc length δ^k_{uv}. This may reduce the computational

burden of repairing the obtained relaxed solution in Phase 2. Moreover, we can further refine the relaxed solution by deleting routes that contain only non-working arc $\bar{o} \to \bar{d}$.

Phase 2: For each line l, considering each pair of adjacent scheduled bus service trips m and n such that the difference between their departure times t_m and t_n is larger than the maximum departure headway time g_l (i.e., $t_m + g_l < t_n$), we can execute an *insert* operation or a *shift* operation to modify the current solution. When $t_m + h_l \le t_n - h_l$, we use the insert operation presented in Algorithm 4 to try insert a new bus service ℓ such that $t_m + h_l \le t_\ell \le t_n - h_l$ and modify some bus routes to ensure their feasibility. In this insert operation, we first check all the existing bus routes, and if a route r_k corresponding to bus k exists that can pick up a newly generated bus service ℓ without deadhead trips, we add this service trip ℓ to route r_k and insert it in the current bus timetable. Otherwise, we generate a new bus service trip ℓ such that $t_\ell = t_m + h_l$ and construct a new bus route by assigning this trip to an idle bus that has not been assigned to any work in the relaxed solution. After executing an insert operation for the second situation, the existing service trip m and the newly generated service trip ℓ must satisfy the maximum headway time constraint as $t_\ell = t_m + h_l$, while the newly generated service trip ℓ and the existing service trip n may satisfy the maximum headway constraint as $t_\ell \le t_n - h_l$.

Algorithm 4 Insert operation

Input: current solution, two adjacent trips m, n and corresponding departure time t_m and t_n
for $t \in [t_m + h_l, t_n - h_l]$ **do**
  if there exists a bus k that has been assigned work can perform a newly generated bus service trip ℓ whose departure time is t without deadhead trips **then**
    add trip ℓ to bus k's route r_k, **break**
  end if
end for
if no new service trip is generated **then** generate a new service trip ℓ with departure time of $t_m + h_l$, and assign trip ℓ to an idle bus k
end if
Output: revised solution

When $t_m + h_l > t_n - h_l$, we cannot directly insert a new bus service trip between service trips m and n, but we can change their departure times using the shift operation presented in Algorithm 5 to reduce the difference between t_m and t_n and satisfy the maximum departure headway time constraints. We can also increase the difference between t_m and t_n to enable new bus service trips to be inserted between service trips m and n. For convenience of presentation, we let $R(m)$ denote the bus route that covers service trip m. Here, we consider the following three situations: (i) if t_m can be delayed for 1 min while $R(m)$ and the trip pair (m, n) remain feasible, we delay t_m by 1 min, and accordingly update $t_m = t_m + 1$; (ii) if t_n can be advanced by 1 min while $R(n)$ and the trip pair (n, ℓ) remain feasible, where ℓ is the next bus service trip to n, we advance t_n by 1 min, and accordingly update $t_n = t_n - 1$; (iii) if it is not advantageous to delay of advance service trip m or n's departure time by 1 min, we change service trip m's departure time to $t_n - h_l$ to possibly increase the departure headway time between service trips ℓ' and m, where ℓ' is the scheduled service trip that precedes m; meanwhile, we assign the new service trip m to an idle bus k and accordingly update $R(m) \leftarrow R(m) \setminus \{m\}$ and $r_k \leftarrow \{m\}$.

Algorithm 5 Shift operation

Input: current solution, two adjacent trips m, n and corresponding departure time t_m and t_n
if t_m can be delay for one minute while the feasibilities of trips pair (m, n) and the route $R(m)$ that contains trip m are still ensured **then**
  set $t_m = t_m + 1$
else if t_n can be advanced for one minute while the feasibilities of trips pair (m, n) and the route $R(n)$ that contains trip n are still ensured **then**
  set $t_n = t_n - 1$
else
  set $t_m = t_n - h_l$, $R(m) \leftarrow R(m) \setminus \{m\}$, and assign bus service trip m to an idle bus
end if
Output: revised solution

Phase 3: If Phase 2 generates a feasible solution, we try to improve its quality by reducing the bus fleet size as much as possible. This reduction can be achieved if, for each bus $k \in K$, another bus $k' \in K$ with the same depot exists such that the starting time

of bus route $r_{k'}$ is later than the ending time of bus route r_k and one of the following conditions holds: (i) the ending depot of r_k and the starting depot of $r_{k'}$ are the same or (ii) the difference between the starting time of $r_{k'}$ and the ending time of r_k is long enough to accommodate the deadhead trip time of k from r_k's ending depot to $r_{k'}$'s starting depot. If these conditions hold, we use bus k for bus k''s service trips and make bus k' idle, which saves the cost c_f of using k'.

4.4. Subgradient optimization procedure

In this section we adopt a widely used subgradient optimization procedure to find the near-optimal Lagrangian multipliers λ and μ to obtain tighter lower bound. Let $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_{|C_1|}, \mu_1, \mu_2, \dots, \mu_{|C_2|}\}$ be the set of all Lagrangian multipliers in vectors λ and μ, and $\eta = \{\eta_1, \eta_1, \dots, \eta_{|C_1|+|C_2|}\}$ be the subgradient vector corresponding to Lagrangian multiplier vector Λ. The subgradient optimization procedure is an iterative approach, in which each component in Λ is initialized as 0 at the first iteration. Let λ_i^κ, μ_i^κ and η_i^κ be the ith components of vector λ, μ and η at κth iteration of the subgradient optimization, respectively. Lagrangian multipliers λ and μ then can be updated according to the following formulation (see, e.g., Held and Karp, 1971):

$$\lambda_i^\kappa \leftarrow \max\{\lambda_i^{\kappa-1} + \theta \cdot \frac{UB_{\min} - LB_{\min}(\lambda, \mu)}{\|\eta^\kappa\|^2} \cdot \eta_i^\kappa, 0\} \quad (i = 1, 2, \dots, |C_1|)$$

and

$$\mu_i^\kappa \leftarrow \max\{\mu_i^{\kappa-1} + \theta \cdot \frac{UB_{\min} - LB_{\min}(\lambda, \mu)}{\|\eta^\kappa\|^2} \cdot \eta_{i+|C_1|}^\kappa, 0\} \quad (i = 1, 2, \dots, |C_2|)$$

where $UB_{\min}$ is the objective value of the incumbent best feasible solution for problem $\mathbf{P}_{\min}$, $LB_{\min}(\lambda, \mu)$ is the optimal objective value of $\mathbf{P}_{\min}(\lambda, \mu)$ at current iteration with current Lagrangian multiplier values λ and μ, and parameter θ is a preset step size of the subgradient optimization procedure.

5. Computational study

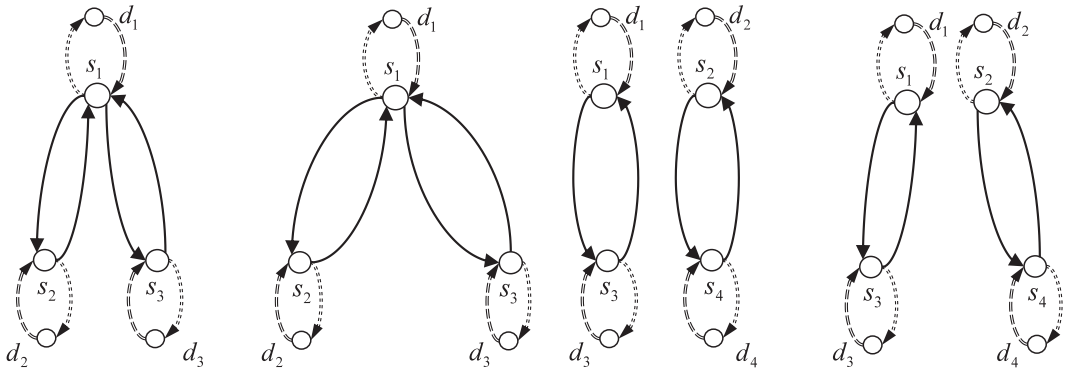
In this section, we conduct a computational study to examine the effectiveness and performance of our modeling and solving methods. The developed solution method is implemented using C# language, and all the test instances are solved on a personal computer with a 3.60 GHz Intel Core i7-7700 processor and 16 GB RAM. Section 5.1 introduces the method for generating test instances, followed by the computational result reports, analysis and discussions in Sections 5.2 and 5.3. In our implementation of the subgradient optimization procedure, parameter $MaxIter$ is set to 15,000. The step size θ is initialized as 2.0 and reduced by 10% if $LB_{\min}$ is not improved for 160 or more consecutive iterations. We execute the feasible solution heuristic with probability of 1.0 (respectively 0.2) if the number of iterations is no greater than 50 (respectively greater than 50).

5.1. Instance generation

Our computational study can be divided into two parts. In the first part, we conduct four sets of test instances using a simple line topology with two depots and two opposing lines; see Fig. 1(a). The first set consists of small-size test instances that can be solved using the CPLEX solver and the proposed LRH, to investigate the effectiveness of our solution method. The second set of test instances uses data from Line 119 operated by Hefei Public Transport Group. The third set consists of test instances conducted to observe the computational results with various values of minimum and maximum headway times. The fourth set consists of test instances conducted to check whether the computational results are influenced by different combinations of the ratio of "fixed cost c_f" to "service profit f_l" and the ratio of "deadhead cost $c_d \times \alpha_{p_l,q_l}$" to "service profit f_l". In these two instance sets, for each (combination of) tuned parameter(s), we randomly generate three test instances, where 80 buses (i.e., $|K| = 80$) are considered in each instance.

In the second part of our computational study, we test our modeling and solving method using four complex line topologies, as shown in Fig. 3, where deadheading between each pair of termini is allowed. The first line topology, denoted by LT-1, consists of three termini, three depots, and two pairs of opposing lines, where one terminus is shared by all of the four lines, and the maximum deadhead trip time between termini is less than the minimum service trip time of all the lines, see Fig. 3(a). The second line topology, denoted by LT-2, is similar to the first, except that the maximum deadhead trip time between termini is at least equal to the maximum service trip time, see Fig. 3(b). The third line topology, denoted by LT-3, consists of four termini, four depots, and two independent pairs of opposing lines, in which the maximum deadhead trip time between termini is less than the minimum service trip time; see Fig. 3(c). The fourth line topology, denoted by LT-4, is similar to the third, except that the maximum deadhead trip time between termini is at least equal to the maximum service trip time; see Fig. 3(d). For each line topology, we randomly generate five instances, where 180 buses (i.e., $|K| = 180$) are considered in each instance.

We generate the data of each instance as follows, unless otherwise specified. The planning horizons in our test instances are the same $[0, 1080]$, e.g., from 6 : 00 to 24 : 00, in which there are two peak periods $[7:00, 9:00)$ and $[17:00, 19:00)$ and three off-peak periods, including $[6:00, 7:00)$, $[9:00, 17:00)$ and $[19:00, 24:00]$. Without loss of generality, for each terminus of each line $l \in L$, we assume that there exists an adjoining depot. The service trip time τ_l is randomly selected from $\{60, 65, \dots, 85, 90\}$, the probability of selecting a particular component being 1/7. For each line l with service trip time τ_l, its opposing line l''s service trip time is set to the round value of $\tau_l \cdot \varphi_l$, where φ_l is randomly selected from $\{1.00, 1.01, \dots, 1.19\}$ and the probability of each value being selected is 1/20. For example, when $\tau_l = 65$ min and $\varphi = 1.10$, $\tau_{l'} = 65 \cdot (1.10) \approx 72$ min. Similarly, the deadhead trip



(a) LT-1: the maximum deadhead trip time between termini is less than the minimum service trip time

(b) LT-2: the maximum deadhead trip time between termini is at least equal to the maximum service trip time

(c) LT-3: the maximum deadhead trip time between termini is less than the minimum service trip time

(d) LT-4: the maximum deadhead trip time between termini is at least equal to the maximum service trip time

Fig. 3. Four line topologies.

time is $\tau_l/\psi_{p_l,q_l}$ rounded to the nearest integer, where ψ_l is randomly selected from $\{0.61, 0.62, \ldots, 1.00\}$, the probability of selecting a particular component being 1/40. The pull-in and pull-out times (in minutes) between the depot and its adjoining terminus are randomly selected from $\{5, 6, 7, 8, 9\}$, the probability of selecting a particular component being 1/5. The pull-in (pull-out) time between the depot and other more distant termini is set to the sum of the pull-in (pull-out) time and the deadhead time between its adjoining terminus and other more distant termini. Moreover, we set the maximum battery capacity $E_{\max}$ to 100 and the electricity e_k consumed per unit time while running to 0.1, which means that each electric bus can run at most $(100/0.1/60) \approx 16.67$ hours when one unit time is defined as 1 min. When the minimum headway time h_l is determined, the maximum headway time g_l is generated as $g_l = h_l + \Delta_h$. The minimum dwell time is set to 5 min; i.e., $\beta = 5$. Moreover, we set the fixed cost $c_f = 2.0 \cdot f_l$ and the deadhead cost $c_d \times \alpha_{p_l,q_l} = 0.5 \cdot f_l$. For simplicity, the monetary unit is scaled such that $f_l = 1$. For the second part of our computational study, the minimum headway time h_l for each line $l \in L$ in the peak and off-peak periods is set to 3 and 6 min, respectively. The value of Δ_h is randomly selected from $\{3, 4, 5\}$ and each component is selected with probability of 1/3. The Appendix describes in detail the method of generating the deadhead trip time between two termini that belong to different pairs of lines.

For each test instance, we record the numbers of generated service trips and deadhead trips. As a feasible solution may not be obtained for every execution of the solution method presented in Section 4.3, we record the ratio of the number of times a feasible solution was obtained to the number of times the feasible solution heuristic was executed (see the "Average LB generation rate" column in Tables 8 and 9). For each instance, we also record the optimality gap and profitability index, which are defined as

$$Gap = \frac{UB - LB}{LB} \times 100\%, \text{ profitability index} = \frac{\text{total profit}}{\text{total cost}},$$

where UB is the objective value of problem $\mathbf{P}_{\max}$ with the optimal solution of $\mathbf{P}_{\min}(\lambda, \mu)$, LB is the objective value of problem $\mathbf{P}_{\max}$ with a feasible solution to problem $\mathbf{P}_{\min}$, and the profitability index can be used to measure the profitability/attractiveness of running the bus lines.

5.2. Computational study on the simple line topology

We first report the results of the first part of the computational study to evaluate the performance of our solution method and observe how the headway times, service trip profit and operating cost settings affect the solution.

5.2.1. Results from small-scale test instances

We randomly generate five small-scale test instances using the aforementioned methods and parameters, except for the planning horizon, which is set to $[0, 360]$ rather than to $[0, 1080]$. We solve these small-scale instances using the commercial mixed-integer linear programming solver CPLEX 22.1.0 and our LRH, where for each instance the running time limit of the solver is set to 10 h and the maximum number of iterations of the LRH is set to 10,000. We define the optimality gaps Gap^1 and Gap^2 as follows

$$Gap^1 = \frac{UB_{LRH} - UB_{Cplex}}{UB_{Cplex}} \times 100\% \text{ and } Gap^2 = \frac{LB_{LRH} - LB_{Cplex}}{LB_{Cplex}} \times 100\%,$$

where UB_{LRH} and UB_{Cplex} (respectively LB_{Cplex} and LB_{LRH}) are the upper bounds (the objective values) to problem $\mathbf{P}_{\max}$ obtained by the CPLEX solver and LRH.

Table 4 summarizes the computational results from these small-scale test instances, where "$|K|$" column report the numbers of buses used in different solutions and "CT(s)" columns report the CPU time consumed in solving the considered problems. The

Table 4
Computational results obtained by different methods.

Instance	CPLEX solver				Lagrangian relaxation heuristic						Gap^1	Gap^2
index	UB	LB	Gap	CT(s)	UB	LB	Gap	No. of trips	$\|K\|$	CT(s)		
1	58.67	51.00	15.03%	36000	51.67	51.00	1.30%	102	24	146.16	−13.55%	0.00%
2	59.50	51.50	15.53%	36000	51.58	51.50	0.16%	103	25	131.98	−15.35%	0.00%
3	57.50	50.50	13.86%	36000	51.25	49.50	3.54%	101	25	110.84	−12.19%	−2.02%
4	57.00	50.50	12.87%	36000	51.25	48.00	6.77%	99	24	113.61	−11.22%	−5.21%
5	64.50	61.00	5.74%	36000	61.52	55.50	10.85%	98	19	94.82	−4.84%	−9.91%
Average:	59.43	52.90	12.61%	36000	53.45	51.10	4.61%	100.6	23.4	119.48	−11.18%	−3.52%

Table 5
Computational results obtained by sequential and integrated optimization methods.

Instance	TNT (CPLEX)		VSP (CPLEX)			TNT+VSP (SEQ)		Integrated framework (LRH)					Gap^3
index	OptV	CT(s)	OptV	$\|K\|$	CT(s)	OptV	CT(s)	UB	LB	Gap	$\|K\|$	CT(s)	
1	410.0	0.09	−233.0	58.0	2839.58	177.0	2839.67	220.2	210.0	4.87%	35.0	1425.59	18.64%
2	415.0	0.07	−200.5	50.0	2528.95	214.5	2529.02	241.2	231.5	4.18%	33.0	2432.68	7.93%
3	412.0	0.08	−232.0	58.0	2583.30	180.0	2583.37	222.1	216.5	2.57%	35.0	1454.60	20.28%
4	410.0	0.08	−233.0	58.0	2600.17	177.0	2600.25	220.2	209.0	5.38%	36.0	941.91	18.08%
5	416.0	0.08	−189.0	47.0	2893.22	227.0	2893.29	247.7	239.0	3.66%	32.0	2380.92	5.29%
Average:	412.6	0.08	−217.5	54.2	2689.04	195.1	2689.12	230.3	221.2	4.11%	34.2	1727.14	13.38%

average optimality gap of the CPLEX solver is much greater than that of the proposed LRH, and the LRH consumes significantly less CPU time than the CPLEX solver. These results confirm both effectiveness and efficiency of our modeling and solving methods. Moreover, the "UB" columns show that the LRH can provide much tighter upper bounds for problem $\mathbf{P}_{\max}$ than the CPLEX solver. The "Gap^2" column indicates that although the LRH on average produces worse feasible solutions than those produced by the CPLEX solver, for two instances the feasible solutions produced by them have the same quality. It can be also observed that the last instance shows a gap of 10.85% of LRH, the number of buses used in the last instance is 19 which is much less than those in other instances, while the number of generated trips in the last instance is slightly less than that in other instances. A plausible explanation for this result is the last instance may produce an optimal solution with less buses and 100 or more service trips, while the LRH generates an ordinary solution with less trips and less profit which in turn results in a worse optimality gap. In conclusion, the computational results show that the proposed LRH can provide tighter bounds for the considered problems, but the heuristic that produces feasible solutions needs to be improved with further optimization techniques.

5.2.2. Trade-off between the bus timetable and vehicle schedule

In this section we investigate the advantage of the developed integrated optimization method. We generate five instances using the aforementioned method and parameters, except for the fixed cost c_f, which is $4.0 \cdot f_l$ rather than $2.0 \cdot f_l$. We solve TNT and VSP using CPLEX solver in a sequential manner, with the output of TNT as the input of VSP. The TNT aims to schedule as much service trips as possible that can be formulated as

$$\begin{aligned} \mathbf{P}_{TNT}: \text{Maximize} \quad & \text{objective function (1)} \\ \text{subject to} \quad & \text{constraints (7)–(9).} \end{aligned}$$

Let $\hat{A}_{inv}$ denote the set of all inventory arcs that are selected in the timetable solution of the TNT. The VSP aims at using least cost to cover all obtained service trips that can be formulated as

$$\begin{aligned} \mathbf{P}_{VSP}: \text{Maximize} \quad & \text{objective function (1)} \\ \text{subject to} \quad & \textstyle\sum_{k\in K}\sum_{u\to v\in \hat{A}_{inv}} x_{uv}^k = 1, \\ & \text{constraints (2)–(6), (9),} \end{aligned} \tag{10}$$

where for each inventory arc $u \to v \in A$, its cost $c(u, v)$ is set to 0 (respectively $+\infty$) if this arc is (respectively not) in inventory arc set $\hat{A}_{inv}$.

Table 5 summarizes the results obtained by the aforementioned sequential method SEQ and the developed integrated optimization methods LRH, where "OptV" columns report the optimal objective values obtained by the CPLEX solver. The "Gap^3" column shows the optimality gaps between solutions obtained by SEQ and LRH, where the optimality gap Gap^3 is defined as $100\% \times (LB_{LRH} - OptV_{SEQ})/OptV_{SEQ}$. From "$Gap^3$" and "CT(s)" columns, we can observe that compared with SEQ, the developed LRH can produce better solutions using less computational time and the improvement gaps range from 5.29% to 20.28%, which reports the significance of integrating TNT and VSP decisions and demonstrate the effectiveness of LRH. Moreover, the $|K|$ columns indicate that in the solutions obtained by method SEQ, more buses are required to cover the obtained optimal bus timetable, thus increasing the operational cost.

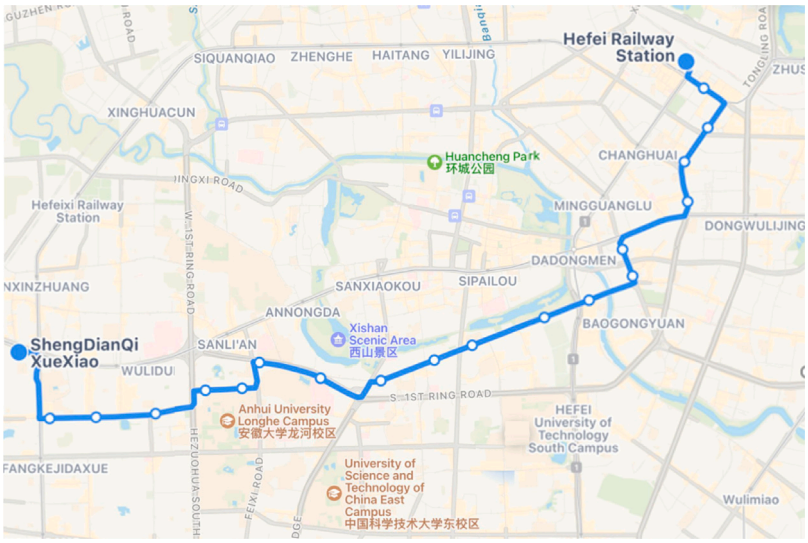


Fig. 4. Bus line 119 in Hefei city (Google Maps: https://www.google.com/maps/place/Hefei,+Anhui,+China).

Table 6
Computational results on realistic test instances: Part 1.

Fixed cost	*UB*	*LB*	*Gap*	No. of used buses	No. of trips	Profitability index	CPU time(s)
1.00	303.53	294.00	3.24%	29.00	324.00	10.80	2458.11
2.00	274.66	269.00	2.10%	28.00	326.00	5.72	2514.54
3.00	245.88	242.00	1.60%	28.00	326.00	3.88	2562.05
4.00	217.20	214.00	1.50%	28.00	326.00	2.91	2356.69
Average:	260.32	254.75	2.11%	28.25	325.50	5.83	2472.85

Table 7
Computational results on realistic test instances: Part 2.

$E_{\max}$	*UB*	*LB*	*Gap*	No. of used buses	$\max_{k\in K}\{N_k\}$	$\min_{k\in K}\{N_k\}$	Sample deviation of N_k	CPU time(s)
70	274.70	262.00	4.85%	31.00	12.00	4.00	4.72	2840.21
80	274.67	269.00	2.11%	28.00	14.00	8.00	8.31	2738.56
90	274.66	270.00	1.73%	28.00	16.00	6.00	19.57	2285.69
100	274.66	269.00	2.10%	28.00	16.00	6.00	18.90	2463.61
110	274.66	269.00	2.10%	28.00	16.00	6.00	18.90	2465.75

5.2.3. Results from realistic test instances

As shown in Fig. 4, Bus Line 119 in Hefei city connects Hefei Railway Station (HRS) and Anhui Electrical Engineering School (AEES, "Sheng DianQi XueXiao" in Chinese pronunciation). The application comprises 275 service trips covered by 24 buses; i.e., each bus operates about 11.46 trips on average. There are 27 off-peak and peak periods (11 in one direction and 16 in the other), and in each period the time span ranges from 30 min to 5 h, and the headway varies over a small range. For the realistic test instances, the travel time along the service (deadhead) trip from HRS to AEES is set to 55 (27) minutes, and the travel time along the service (deadhead) trip from AEES to HRS is set to 50 (25) minutes. The planning horizon is $[0, 1020]$, e.g., from 6:00 to 23:00, in which there are two peak periods $[6:00, 8:00)$ and $[14:00, 18:00)$ and two off-peak periods $[8:00, 14:00)$ and $[18:00, 23:00]$. The minimum and maximum headways in the peak (off-peak) periods are set to 4 and 7 min (8 and 13 min). Moreover, all the pull-in and pull-out times are set to 0. Other data in the realistic test instances are generated using the above method unless otherwise specified.

Table 6 presents the computational results from the realistic test instances with four fixed cost settings. The results show that the proposed LRH effectively solves the realistic cases and provide high-quality feasible solutions with an average optimality gap of only 2.11%. The average computational time is 2472.85 s, i.e., 0.69 h, indicating that the LRH is suitable for practical off-line transit planning. In the instances with fixed costs of 2.00, 3.00 and 4.00, the LRH generates 326 trips operated by 28 buses. That is, each used bus serves 11.64 trips, which is slightly greater than the current value of 11.46. Thus, our LRH may produce better vehicle schedules, which would improve the bus utilization. Moreover, the profitability index tends to decrease as the fixed cost increases, because the total cost becomes higher as the value of vehicle fixed cost increases, and the influence of the profits is small as the number of service trips varies over a small range.

Table 7 presents the computational results from the realistic test instances with five $E_{\max}$ settings. We can see that when $E_{\max} = 70, 80$, the optimality gaps, upper and lower bounds are inferior to that of $E_{\max} = 90, 100, 110$, indicating that as the maximum

Table 8
Computational results with different minimum and maximum headway times.

Instance set	h_l in peak periods	h_l in off-peak periods	Δ_h	Average no. of service trips	Average no. of deadhead trips	Average LB generation rate	Average UB	Average LB	Average Gap	Profitability index	CPU time(s)
1	3	6	3	397.33	0.67	34.10%	304.46	296.33	2.73%	3.99	2341.62
2	3	6	4	391.33	0.67	40.92%	288.01	281.00	2.51%	3.55	2124.03
3	3	6	5	394.33	0.33	40.92%	303.94	296.17	2.64%	4.02	2422.34
4	3	6	6	372.33	1.67	40.88%	282.42	274.17	3.01%	3.84	1689.58
Average:				388.83	0.83	39.21%	294.71	286.92	2.72%	3.85	2144.39
5	4	7	3	327.00	1.67	37.48%	256.75	251.50	2.08%	4.34	2224.42
6	4	7	4	320.33	0.33	45.13%	258.05	250.83	2.88%	4.61	2031.56
7	4	7	5	321.67	0.33	51.55%	252.62	246.83	2.38%	4.37	2352.72
8	4	7	6	323.67	0.33	49.71%	257.87	251.50	2.57%	4.55	2216.72
Average:				323.17	0.67	45.97%	256.32	250.17	2.48%	4.47	2206.35
9	5	8	3	275.00	0.33	22.97%	216.97	210.83	2.93%	4.31	2023.13
10	5	8	4	278.67	1.33	44.87%	225.95	220.67	2.39%	4.81	2564.64
11	5	8	5	276.00	1.33	51.20%	228.59	222.67	2.67%	5.19	2525.58
12	5	8	6	274.00	1.33	58.46%	220.66	214.67	2.79%	4.73	2639.92
Average:				275.92	1.08	44.38%	223.04	217.21	2.69%	4.76	2438.32
13	6	9	3	242.67	2.67	21.31%	190.83	186.00	2.61%	4.31	1998.93
14	6	9	4	242.67	0.67	43.91%	196.59	192.33	2.21%	4.94	2202.79
15	6	9	5	242.33	2.33	56.12%	192.61	187.83	2.56%	4.47	2125.61
16	6	9	6	244.00	0.67	56.43%	200.84	197.00	1.95%	5.21	2590.24
Average:				242.92	1.58	44.44%	195.22	190.79	2.33%	4.73	2229.39

battery capacity increases, the negative influence of the battery capacity limit on the problem becomes less. We can see that in the instances with small $E_{\max}$, the number of used buses is more than that of instances with bigger $E_{\max}$. This is because a bus can perform less trips as the battery capacity is small, as shown in the "$\max_{k\in K}\{N_k\}$" column, where N_k is the number of service trips covered by bus k. As shown in the "CPU time(s)" column, using more buses requires more computational time as more constrained shortest path problems should be solved when determining feasible solutions. From the "Sample deviation of N_k", one can observe that the instances with small battery capacity may produce paths whose lengths vary over a narrower range.

5.2.4. Results with various headway parameter settings

Table 8 summarizes the computational results with various minimum and maximum headway times, in which each column reports the average result of three random instances. Table 8 shows that high-quality solutions can be obtained within 1 h and the average optimality gaps of all test instance sets, except instance set 4, are less than 3%, which confirms the stability of our modeling and solution methods when applied to randomly generated test instances on simple line topology. The number of generated service trips tends to decrease as the headway times increase, which is reasonable. Moreover, the number of generated deadhead trips is nearly zero in our test instances, implying that deadhead operations are undesirable when generating feasible solutions. The "Average LB generation rate" column shows that the LB (recall that LB is defined as the lower bound to problem $\mathbf{P}_{\max}$) generation rate tends to fall as Δ_h decreases. This fall may be explained by the fact that when the time difference between the maximum and minimum headway times becomes small, the coupling tendency becomes stronger, which makes it more difficult to find feasible solutions using the feasible solution heuristic. The profitability index is stable with various headway times, and as the headway times increase, the profitability index increases slightly. A possible explanation for this slight increase is that as the timetable is sparse, one bus may perform more service trips, which may help produce compatible pairs of trips.

5.2.5. Results with various cost and profit settings

In this section, we report the results obtained with various cost and profit settings. For convenience, we define the following ratios for the same line l,

$$\text{RFP} = \frac{\text{fixed cost } c_f}{\text{service profit } f_l}, \quad \text{RDP} = \frac{\text{deadhead cost } c_d \times \alpha_{p_l,q_l}}{\text{service profit } f_l}.$$

Setting RFP $= 1, 2, 3, 4$ and RDP $= 0.2, 0.4, 0.6, 0.8$, allows 16 combinations of RFP and RFP. As described in Section 5.1, for each combination, we generate three random instances, thus producing another 16 sets of test instances in which there are 48 random instances.

Fig. 5(a) shows how the optimality gap is affected by various combinations of RFP and RDP. We can observe that the average optimality gaps obtained with RFP $= 1, 2$ are superior to that with RFP $= 3, 4$ in most instance sets, which implies that the more the "RFP", the larger the optimality gap. Fig. 5(b) shows the average LB generation rate with various combinations of RFP and RDP, and it shows that the generation rate increases as the value of RFP increases. A plausible explanation for this behavior is that as RFP increases, the number of buses used decreases; this shortage of used buses makes it difficult to obtain feasible solutions. Fig. 5(c) shows that the profitability index decreases when the fixed cost increases. This is reasonable as the maximum headway time constraints are imposed in our problem, making the numbers of generated service trips substantially the same because the headway times settings are approximately equal to each other for different instances; see Table 8. As the numbers of service trips are approximately the same, the numbers of used buses are also approximately the same. Therefore, the higher the fixed cost of using a bus, the lower the profitability index.

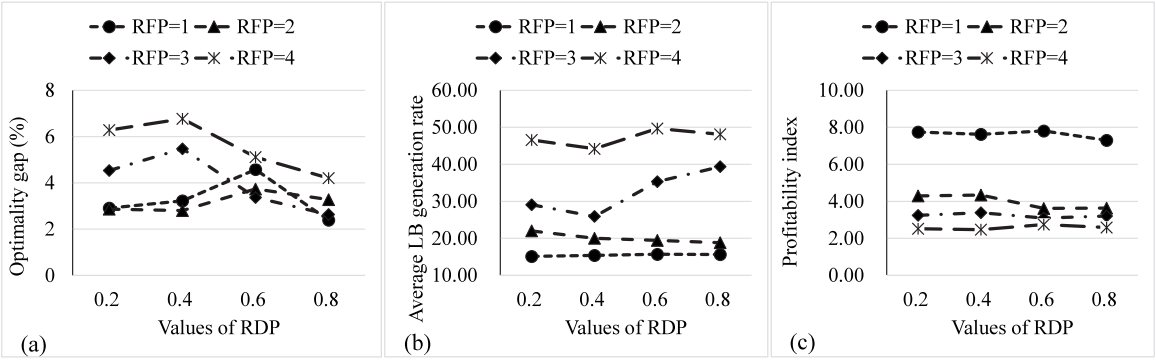


Fig. 5. Average optimality gaps, LB generation rates, profitability indexes with various RFP and RDP.

Table 9
Computational results on different line topologies.

Instance set	Line topology	Δ_h in two pairs of lines	LB generation rate	UB	LB	Gap	CPU time(s)
33	LT-1	(4.0,3.0)	24.59%	656.45	613.00	7.09%	17709.93
		(5.0,3.0)	22.61%	623.31	583.50	6.82%	18000.00
		(5.0,5.0)	28.11%	590.86	556.00	6.27%	15254.65
		(3.0,3.0)	21.57%	618.90	577.50	7.17%	18000.00
		(4.0,4.0)	25.79%	580.90	545.00	6.59%	17649.17
Average:		(4.2,3.6)	24.53%	614.08	575.00	6.79%	17322.55
34	LT-2	(4.0,5.0)	23.19%	573.91	530.50	8.18%	13742.21
		(4.0,4.0)	23.97%	603.73	564.00	7.05%	15214.45
		(5.0,5.0)	25.55%	569.33	536.00	6.22%	14579.65
		(3.0,3.0)	22.63%	599.15	562.50	6.52%	16705.84
		(5.0,5.0)	25.62%	591.02	561.00	5.35%	14314.85
Average:		(4.2,4.4)	24.19%	587.43	550.80	6.66%	14911.40
35	LT-3	(4.0,4.0)	19.31%	634.72	560.00	13.34%	18000.00
		(5.0,4.0)	19.89%	615.28	557.50	10.36%	18000.00
		(3.0,3.0)	15.85%	647.31	573.00	12.97%	18000.00
		(5.0,5.0)	19.23%	648.17	577.00	12.33%	18000.00
		(3.0,5.0)	15.40%	638.60	572.50	11.55%	18000.00
Average:		(4.0,4.2)	17.94%	636.82	568.00	12.11%	18000.00
36	LT-4	(4.0,4.0)	23.20%	591.13	532.00	11.11%	18000.00
		(4.0,4.0)	22.81%	589.27	534.00	10.35%	18000.00
		(5.0,4.0)	21.35%	615.09	563.50	9.15%	18000.00
		(5.0,5.0)	20.93%	623.30	557.00	11.90%	18000.00
		(3.0,5.0)	17.72%	617.02	559.00	10.38%	18000.00
Average:		(4.2,4.4)	21.20%	607.16	549.10	10.58%	18000.00

5.3. Computational study on complex line topologies

As introduced in Section 5.1, in the second part of our computational study, we test four set of instances (20 instances in total) on four complex line topologies as shown in Fig. 3.

5.3.1. Computation results

Table 9 reports the computational results obtained for various line topologies. The "*Gap*" column shows that although the optimality gaps obtained for various complex line topologies are greater than those for the simple line topology (see Table 8), most optimality gaps for test instances of LT-1 and LT-2 are less than 8%, demonstrating that our solution method can also effectively solve these problems on such two-line topologies. The optimality gaps for test instances of LT-3 and LT-4 are higher than 8% and range from 9.15% to 13.34%, which would be resulted from that our solution is less likely to generate feasible solutions at each iteration for such test instances; see the "LB generation rate" column of Table 9.

The "CPU time(s)" columns in Tables 8 and 9 show that much more computational time is used to solve a test instance for complex line topologies than for a simple line topology, implying that the more complex a line topology is, the more time is required to obtain a feasible solution, because of the larger solution space.

5.3.2. Results on shared depots

Fig. 6(a) shows the numbers of used buses in feasible solutions. The obtained bus fleet sizes in feasible solutions are quite stable in various random test instances, confirming the robustness of our solution methods for problems involving complex line topologies. Fig. 6(b) shows that the numbers of generated service trips for various line topologies range from 740 to 780, which may imply that such problems have similar characteristics (e.g., complexity, scale). Fig. 6(c) shows the line changes obtained for various line topologies, where a line change represents a bus that is switching from one pair of opposing lines to another pair of opposing lines.

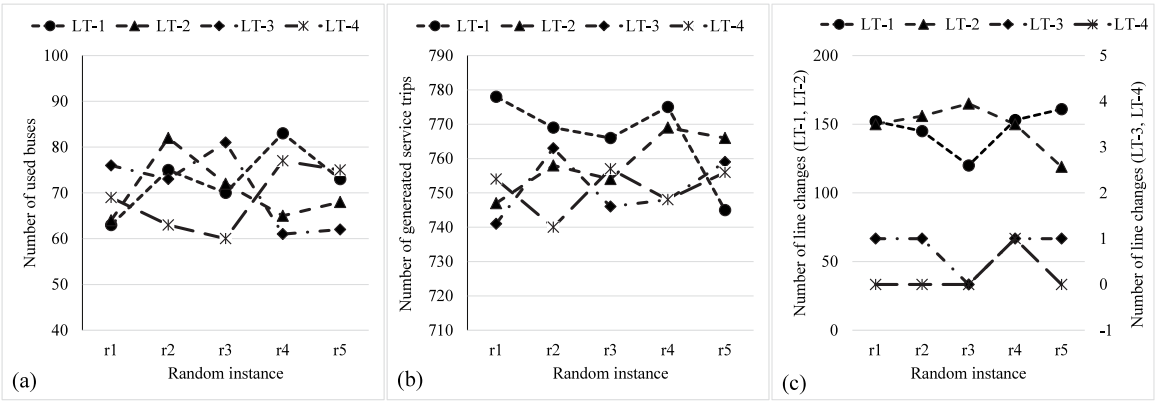


Fig. 6. Obtained indicators in different instances involving complex line topologies.

Many line changes occur in instances of LT-1 and LT-2, whereas at most one line change occurs in one test instance of LT-3 and LT-4. A plausible explanation for this discrepancy is shared depot in LT-1 and LT-2 (see, e.g., Figs. 3(a)–(b)), which may help generate more line connections between different pairs of opposing lines. Similar to the results obtained for the simple line topology, no deadhead trip is produced for complex line topologies, again indicating that our integrated optimization method avoids performing deadhead trips to reduce the operational cost.

6. Conclusions

We studied an integrated electric bus timetabling and scheduling problem with minimum and maximum headway times, depot requirements, deadheading and vehicle battery capacities considerations. The problem was formulated using a multi-commodity network flow model on a developed time-space network where inventory arcs representing multiple bus operations were constructed. A Lagrangian relaxation heuristic was developed to solve the model. The computational study on various line topologies shows that solving bus timetabling and scheduling problem using an integrated framework performs better than that in a sequential manner and the average optimality gaps of almost all instance sets on simple line topology are less than 3%, while the more complex a line topology is, the larger optimality gap is produced. Moreover, the results also show that more line connections between different pairs of opposing lines may be generated in instances of line topologies with shared depots, implying one would like to avoid performing deadhead trips to reduce the operational cost.

Our solution method, however, may obtain a solution with irregular headways. One way of obtaining a regular timetable is to set the difference between the minimum and maximum headway times as small as possible. Another way is to penalize a deviation from the desired headways in the objective function. However, the developed solution method may not solve the model effectively with such a penalty, because our relaxed problem is a constrained shortest path problem, and no dual information about such penalty can be used to find the constrained shortest path with revised arc lengths (see Section 4.1). Therefore, modeling and solving the integrated problem taking headway regularity into consideration is an interesting topic requiring further research. Further, we focused on developing a model and solution method to simultaneously determine bus timetabling and routing decisions, without considering operational details and siding constraints, such as the difference between travel times in peak and off-peak hours and the depot capacity constraint. Incorporating the time difference and depot capacity constraint would be easy and would not change the nature of our model. The travel time difference can be imposed by setting different time lengths on inventory arcs during peak and off-peak hours. The depot capacity constraint can be modeled by imposing that the total flow along each pull-in arc or pull-out arc be no greater than the capacity of the corresponding depot, and such a linking constraint can also be relaxed using Lagrangian method with an additional Lagrangian multiplier, thus requiring more computational time. Developing more efficient solution methods such as meta-heuristics for the ITTVS problem is an interesting topic that requires further research.

Data availability

Data will be made available on request.

Acknowledgments

The authors thank four anonymous referees for their helpful comments. This study is supported by the National Natural Science Foundation of China (Nos. 72071059, 71925001), China Postdoctoral Science Foundation (No. 2019M662144), and the major science and technology projects in Anhui Province (No. 202003a05020009). Moreover, thanks are due to Hefei Public Transport Group and Jiaoxin Technology Co., Ltd for having provided the operational data used in the paper.

Appendix. The method for generating deadhead trip times

Let s and s' be two termini that belong to different pairs of opposing lines, e.g., s_2 and s_3 in Fig. 3, $\tau_{\min}$ (respectively $\tau_{\max}$) be the minimum (respectively the maximum) value of service trip times among different lines, ν_1 and ν_2 be two coefficients. For instances of LT-1, we set deadhead trip time of $\alpha_{ss'}$ between termini s and s' as the round value of $\tau_{\min} \times \nu_1$, where coefficient ν_1 is randomly selected from $\{0.61, 0.62, \ldots, 1.00\}$ and each component is selected with probability of 1/40. The deadhead trip time of $\alpha_{s's}$ between termini s' and s as the round value of $\alpha_{ss'} \times \nu_2$, where coefficient ν_2 is randomly selected from $\{0.81, 0.82, \ldots, 1.00\}$ and each component is selected with probability of 1/20. Similarly, for instances of LT-2, we respectively set deadhead trip times $\alpha_{ss'}$ and $\alpha_{s's}$ as the round values of $\tau_{\max} \times \nu_1$ and $\alpha_{ss'} \times \nu_2$, where coefficients ν_1 and ν_2 are randomly selected from $\{1.01, 1.02, \ldots, 1.40\}$ and $\{1.01, 1.02, \ldots, 1.20\}$, respectively.

For instances of LT-3 and LT-4, we need to consider four pairs of termini that belong to different pairs of opposing lines as there is no shared depot. Let s and s' be a pair of adjacent termini that belong to different pairs of opposing lines, e.g., the pair of s_1 and s_2 and the pair of s_3 and s_4 in Figs. 3(c) and 3(d). For instances of LT-3, we set deadhead trip time of $\alpha_{ss'}$ between termini s and s' as the round value of $\tau_{\min} \times \nu_1$, where coefficient ν_1 is randomly selected from $\{0.41, 0.42, \ldots, 0.60\}$ and each component is selected with probability of 1/20. The deadhead trip time of $\alpha_{s's}$ between termini s' and s as the round value of $\alpha_{ss'} \times \nu_2$, where coefficient ν_2 is randomly selected from $\{0.81, 0.82, \ldots, 1.00\}$ and each component is selected with probability of 1/20. Similarly, for instances of LT-4, we respectively set deadhead trip times $\alpha_{ss'}$ and $\alpha_{s's}$ as the round values of $\tau_{\max} \times \nu_1$ and $\alpha_{ss'} \times \nu_2$, where coefficients ν_1 and ν_2 are randomly selected from $\{1.01, 1.02, \ldots, 1.20\}$.

Let s and s' be a pair of diagonal termini to different pairs of opposing lines, e.g., the pair of s_1 and s_4 and the pair of s_2 and s_3 in Figs. 3(c) and 3(d). For instances of LT-3, we set deadhead trip time of $\alpha_{ss'}$ between termini s and s' as the round value of $\tau_{\min} \times \nu_1$, where coefficient ν_1 is randomly selected from $\{0.81, 0.82, \ldots, 1.00\}$ and each component is selected with probability of 1/20. The deadhead trip time of $\alpha_{s's}$ between termini s' and s as the round value of $\alpha_{ss'} \times \nu_2$, where coefficient ν_2 is randomly selected from $\{0.81, 0.82, \ldots, 1.00\}$ and each component is selected with probability of 1/20. Similarly, for instances of LT-4, we respectively set deadhead trip times $\alpha_{ss'}$ and $\alpha_{s's}$ as the round values of $\tau_{\max} \times \nu_1$ and $\alpha_{ss'} \times \nu_2$, where coefficients ν_1 and ν_2 are randomly selected from $\{1.21, 1.22, \ldots, 1.40\}$ and $\{1.01, 1.02, \ldots, 1.20\}$, respectively.

References

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., 1993. Network Flows: Theory, Algorithms, and Applications. Prentice Hall, Englewood Cliffs, NJ.

Cacchiani, V., Caprara, A., Toth, P., 2008. A column generation approach to train timetabling on a corridor. 4OR 6, 125–142.

Caprara, A., Fischetti, M., Toth, P., 2002. Modeling and solving the train timetabling problem. Oper. Res. 50 (5), 851–861.

Carosi, S., Frangioni, A., Galli, L., Girardi, L., Vallese, G., 2019. A matheuristic for integrated timetabling and vehicle scheduling. Transp. Res. B 127, 99–124.

Castelli, L., Pesenti, R., Ukovich, W., 2004. Scheduling multimodal transportation systems. European J. Oper. Res. 155 (3), 603–615.

Ceder, A., 2001. Efficient timetabling and vehicle scheduling for public transport. Springer, pp. 37–52.

Ceder, A., 2007. Public Transit Planning and Operation: Theory, Modeling and Practice. Butterworth-Heinemann, MA.

Ceder, A., Wilson, N.H.M., 1986. Bus network design. Transp. Res. B 20 (4), 331–344.

Chu, J.C., Korsesthakarn, K., Hsu, Y.-T., Wu, H.-Y., 2019. Models and a solution algorithm for planning transfer synchronization of bus timetables. Transp. Res. E 131, 247–266.

de Palma, A., Lindsey, R., 2001. Optimal timetables for public transportation. Transp. Res. B 35 (8), 789–813.

Dell'Amico, M., Fischetti, M., Toth, P., 1993. Heuristic algorithms for the multiple depot vehicle scheduling problem. Manage. Sci. 39 (1), 115–125.

Desaulniers, G., Lavigne, J., Soumis, F., 1998. Multi-depot vehicle scheduling problems with time windows and waiting costs. European J. Oper. Res. 111 (3), 479–494.

Desfontaines, L., Desaulniers, G., 2018. Multiple depot vehicle scheduling with controlled trip shifting. Transp. Res. B 113, 34–53.

Fonseca, J., van der Hurk, E., Roberti, R., Larsen, A., 2018. A matheuristic for transfer synchronization through integrated timetabling and vehicle scheduling. Transp. Res. B 109, 128–149.

Gkiotsalitis, K., Alesiani, F., 2019. Robust timetable optimization for bus lines subject to resource and regulatory constraints. Transp. Res. E 128, 30–51.

Guedes, P.C., Borenstein, D., 2015. Column generation based heuristic framework for the multiple-depot vehicle type scheduling problem. Comput. Ind. Eng. 90, 361–370.

Guihaire, V., Hao, J.-K., 2010. Transit network timetabling and vehicle assignment for regulating authorities. Comput. Ind. Eng. 59 (1), 16–23.

Haghani, A., Banihashemi, M., 2002. Heuristic approaches for solving large scale bus transit vehicle scheduling problem with route time constraints. Transp. Res. B 36 (4), 309–333.

Hassold, S., Ceder, A., 2014. Public transport vehicle scheduling featuring multiple vehicle types. Transp. Res. B 67, 129–143.

Held, M., Karp, R.M., 1971. The traveling-salesman problem and minimum spanning trees: Part II. Math. Program. 1, 6–25.

Ibarra-Rojas, O.J., Delgado, F., Giesen, R., Muñoz, J.C., 2015. Planning, operation, and control of bus transport systems: A literature review. Transp. Res. B 77, 38–75.

Ibarra-Rojas, O.J., Giesen, R., Rios-Solis, Y.A., 2014. An integrated approach for timetabling and vehicle scheduling problems to analyze the trade-off between level of service and operating costs of transit networks. Transp. Res. B 70, 35–46.

Kliewer, N., Amberg, B., Amberg, B., 2012. Multiple depot vehicle and crew scheduling with time windows for scheduled trips. Public Transp. 3, 213–244.

Kliewer, N., Mellouli, T., Suhl, L., 2006. A time-space network based exact optimization model for multi-depot bus scheduling. European J. Oper. Res. 175 (3), 1616–1627.

Kulkarni, S., Krishnamoorthy, M., Ranade, A., Ernst, A.T., Patil, R., 2018. A new formulation and a column generation-based heuristic for the multiple depot vehicle scheduling problem. Transp. Res. B 118, 457–487.

Laporte, G., Ortega, F.A., Pozo, M.A., Puerto, J., 2017. Multi-objective integration of timetables, vehicle schedules and user routings in a transit network. Transp. Res. B 98, 94–112.

Li, J.-Q., Borenstein, D., Mirchandani, P.B., 2007. A decision support system for the single-depot vehicle rescheduling problem. Comput. Oper. Res. 34 (4), 1008–1032.

Li, Z.-C., Lam, W.H.K., Wong, S.C., Sumalee, A., 2010. An activity-based approach for scheduling multimodal transit services. Transportation 37, 751–774.

Li, J.-Q., Mirchandani, P.B., Borenstein, D., 2009. A Lagrangian heuristic for the real-time vehicle rescheduling problem. Transp. Res. E 45 (3), 419–433.

Liu, T., Ceder, A., 2017. Integrated public transport timetable synchronization and vehicle scheduling with demand assignment: a bi-objective bi-level model using deficit function approach. Transp. Res. Procedia 23, 341–361.
Lozano, L., Duque, D., Medaglia, A.L., 2016. An exact algorithm for the elementary shortest path problem with resource constraints. Transp. Sci. 50 (1), 348–357.
Lu, J., Nie, Q., Mahmoudi, M., Ou, J., Li, C., Zhou, X., 2022. Rich arc routing problem in city logistics: Models and solution algorithms using a fluid queue-based time-dependent travel time representation. Transp. Res. B 166, 143–182.
Mesa, J., Ortega, F., Pozo, M., 2014. Locating optimal timetables and vehicle schedules in a transit line. Ann. Oper. Res. 222, 439–455.
Oukil, A., Amor, H.B., Desrosiers, J., Gueddari, H.E., 2007. Stabilized column generation for highly degenerate multiple-depot vehicle scheduling problems. Comput. Oper. Res. 34 (3), 817–834.
Petersen, H.L., Larsen, A., Madsen, O.B.G., Petersen, B., Ropke, S., 2013. The simultaneous vehicle scheduling and passenger service problem. Transp. Sci. 47 (4), 603–616.
Righini, G., Salani, M., 2008. New dynamic programming algorithms for the resource constrained elementary shortest path problem. Networks 51 (3), 155–170.
Tong, L., Zhou, L., Liu, J., Zhou, X., 2017. Customized bus service design for jointly optimizing passenger-to-vehicle assignment and vehicle routing. Transp. Res. C 85, 451–475.
Xu, X., Li, C.-L., Xu, Z., 2018. Integrated train timetabling and locomotive assignment. Transp. Res. B 117, 573–593.
Xu, X., Li, C.-L., Xu, Z., 2021. Train timetabling with stop-skipping, passenger flow, and platform choice considerations. Transp. Res. B 150, 52–74.
Zhang, L., Wang, S., Qu, X., 2021. Optimal electric bus fleet scheduling considering battery degradation and non-linear charging profile. Transp. Res. E 154, 102445.
Zhao, F., Zeng, X., 2008. Optimization of transit route network, vehicle headways, and timetables for large-scale transit networks. European J. Oper. Res. 186 (2), 841–855.