Article:
Wu, W, Lin, Y, Liu, R orcid.org/0000-0003-0627-3184 et al. (1 more author) (2022) The multi-depot electric vehicle scheduling problem with power grid characteristics. Transportation Research Part B: Methodological, 155. pp. 322-347. ISSN 0191-2615

https://doi.org/10.1016/j.trb.2021.11.007

Please cite the paper as:
Wu W., Lin Y., Liu R. and Jin W. (2021) The multi-depot electric vehicle scheduling problem with power grid characteristics. Transportation Research Part B. In press.

The multi-depot electric vehicle scheduling problem with power grid characteristics

Weitiao Wu[a1], Yue Lin[a], Ronghui Liu[b], Wenzhou Jin[a]

a. School of Civil Engineering and Transportation, South China University of Technology, Guangzhou 510641, China

b. Institute for Transport Studies, University of Leeds, Leeds, LS2 9JT, United Kingdom

Abstract: Electric buses can bring significant environmental and social benefits in the future public transportation systems. However, the large-scale adoption of electric buses faces major technical challenges caused by not only the limited running range and long charging time, but also the complex power grid characteristics, such as time-of-use (TOU) electricity tariffs and peak load risk. On one hand, the operation cost is determined by the TOU pricing and vehicle schedule. On the other hand, the unbalanced charging demand resulting from the vehicle schedule will cause peak load risk and pose a potential threat to the power grid safety. With the increasing penetration of electric buses, there is a real need to carefully design and manage electric bus scheduling to not only reduce the system costs but also ensure power grid safety. In this paper, we introduce a bi-objective multi-depot electric vehicle scheduling problem, a new generalization to the vehicle scheduling problem where the effects of TOU pricing and peak load risk are explicitly considered. The dual objectives are to minimize the total operation cost and to minimize the peak load resulting from concurrent recharging activities, as constrained by the running range of the electric buses and the capacity of charging depots/stations. A time-expanded network model is devised to represent this problem, while the bi-objective optimization model is reformulated by the lexicographic method. We propose a tailored branch-and-price method to solve the problem. Heuristics and a trip chain pool strategy are embedded into the branch-and-price method to expedite the computation time. Our method is validated through a benchmark network and a real-world bus network in Guangzhou, China. The results demonstrate that our method is effective in cost savings and peak load leveling, and far outperforms the off-the-shelf solver with respect to solution quality and computation efficiency. The real-world application results show that compared to state-of-the-practice, the peak load can be significantly reduced, on top of cost and fleet size savings.

Keywords: Public transport; Vehicle scheduling problem; Power grid characteristics; Peak load risk; Branch-and-price; Heuristics

[1] Corresponding author, E-mail: ctwtwu@scut.edu.cn; Tel: (+86)13763327125

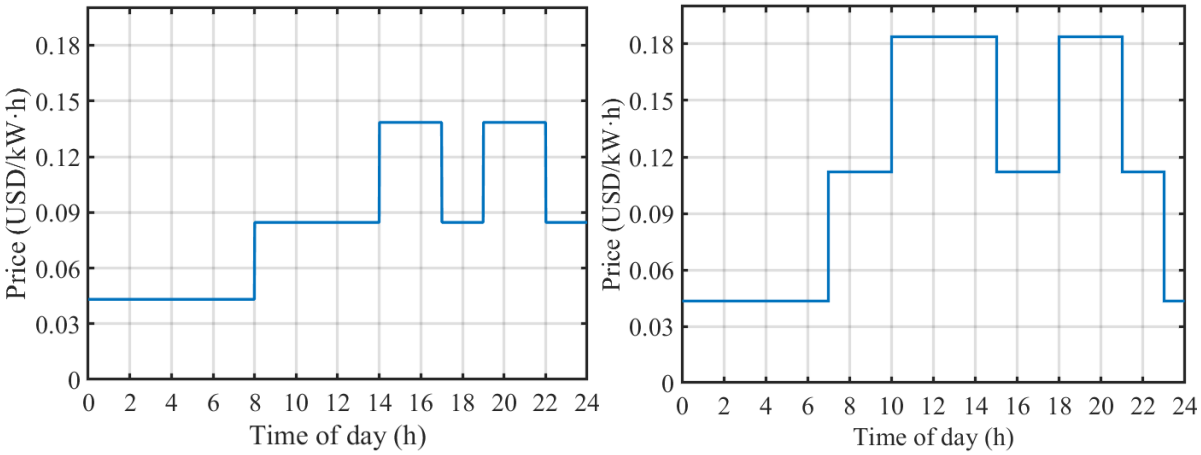
1. Introduction

Public transport is vital to address urban traffic congestion problems and to support sustainable urban development. While conventionally-fueled diesel buses are prone to low energy efficiency and high pollutant emissions, the advancements in renewable energy and battery technology have led to increasing adoption of electric buses by transit agencies. In 2019, the total number of electric buses around the world was about 425,000, an increase of 32% from 2018. China is leading the way in adopting electric buses. For example, in Shenzhen, all heavy diesel buses have been replaced by electric ones. In the United Kingdom, electric bus fleets have been implemented in a number of cities including London and Liverpool (Ayre, 2018).

Recent developments in electric buses have also created new challenges in transportation. Despite their environmental advantages, the relatively short running range and long recharging time remain the major factors that hamper the adoption of electric buses. At present, there are three recharging methods: battery swapping, in-motion wireless charging, and station charging. Battery swapping has low layover time as the depleted batteries can be replaced by fully charged ones almost instantly. However, such a recharging method requires professional labor and costly equipment, such as additional batteries and loading robots. Using dynamic wireless power transfer technology, wireless lane-based charging allows electric buses to be charged on-route. Although this method can reduce the size of batteries and charging time, it requires high construction and maintenance costs and is still in the exploratory stage. Two types of chargers are available for station-based charging: fast chargers and slow chargers. Even though the buses can be fully charged through slow charging overnight, electric buses often cannot complete a full-day's operation without recharging. In practice, even the design running range cannot be fully utilized due to possible road gradient, traffic congestion, air-conditioning, and heating. Typically, an electric bus with a design driving range of 250 km can run for only 175 km with actuated air-conditioning. Moreover, to prolong the battery life, the running range would be deliberately reduced by the public transport operator to avoid deep discharge. Therefore, daytime fast charging is often needed to maintain the operation of electric buses for a whole day, particularly for buses with a small battery capacity.

Another challenge of electric bus operation stems from the complex power grid characteristics associated with the operation cost and power system safety. In the context of time-of-use (TOU) electricity tariffs, the peak tariff is approximately two or three times its low counterpart (Wu et al., 2020). Fig. 1 displays the TOU pricing in two megacities in China, which shows clearly that the charging costs are heavily dependent on the time of day when bus recharging is required, and therefore on the vehicle schedule. If more recharging activities can be undertaken in the low-tariff period, charging costs would be lower but charging demand for that period would be higher. A highly centralized electric power demand has the potential to overload the electricity grid and induce unanticipated voltage drops and poor power quality. The mass adoption of electric buses and technological advancements in battery capacity and high-power fast-charging devices would increase total energy demand. The regional aggregate power demand from both residential and industrial consumption is more likely to exceed the tolerable charging station capacity in the season with peak demand (e.g., summer), given the physical constraints of charger

equipment. For example, the power capacity of a transformer in North America is limited to 25kVA. An added challenge for vehicle scheduling of electric buses, therefore, is how to reduce the peak load as well as overall operation costs.



(a) Guangzhou (b) Beijing

Fig. 1 Time-of-use electricity tariffs in two megacities

The vehicle scheduling problem (VSP) is fundamental in public transport planning and transportation science. A well-designed vehicle schedule for electric buses would be expected to reduce peak load risk for power grids in a cost-efficient manner. Although the VSP has been extensively studied previously, the literature on multi-depot electric vehicle scheduling problems is quite rare due to its complexity. With the increasing penetration of electric buses, there is an increasing need to carefully design and manage electric bus scheduling to not only reduce the system costs but also to ensure power grid safety. To address the above challenges, this paper aims to study the multi-depot electric vehicle schedule problem (MDEVSP) under TOU electricity tariffs that meet the two objectives: (a) minimizing the total operation cost; and (b) minimizing the peak charging load. We propose an exact tailored branch-and-price algorithm to tackle this problem, and we validate the proposed method through a benchmark network and a real-world bus network in Guangzhou, China. Based on the results, new findings and insights are provided.

2. Literature review and main contributions

The advancements of electric vehicles have given rise to a number of new research streams in transportation science. In what follows, we begin by reviewing the electric vehicle (EV) charging infrastructure planning, proceed to review vehicle scheduling problems, and finally point out the objective and potential contributions of this paper.

2.1 EV charging infrastructure planning

A stream of research on EV has centered on the charging infrastructure design. This problem aims to optimally deploy the charging infrastructure considering various factors. Generally, the research subjects can be categorized into two types: light-duty cars and heavy-duty buses. The studies on the former are

extensive, and they include optimizing the charging station location, the number of chargers for plug-in EVs, and the size of fast charging stations (see a review in Shen et al. (2019)).

The behaviour of heavy-duty buses is quite different from that of light-duty cars. For a start, most of the bus services follow fixed routes, and thus they have limited recharging options, as compared to cars. The energy consumption for buses is much higher than that for cars, and therefore they need recharging more frequently. As such, the optimization models developed for light-duty cars cannot be directly adapted to the electric bus systems. Studies addressing the electric bus charging infrastructure are relatively scarce, and are mostly confined to the sphere of transportation assuming unlimited power supply. Wei et al. (2018) investigated the spatial and temporal deployment optimization problem for battery-electric bus systems, with the aim to minimize the deployment cost of replacing conventional diesel and compressed natural gas buses. Rogge et al. (2018) jointly optimized the battery-electric bus scheduling scheme, fleet sizing, and charging infrastructure planning. Their considerations are limited to overnight in-depot charging. Lin et al. (2019) developed a multi-period bus charging infrastructure planning model under the transportation network and power grid system. He et al. (2019) addressed the deployment of fast-charging stations for a battery-electric bus system. The model jointly determines the locations and types of fast-charging stations, the battery sizes, and whether to install energy storage systems. However, the model is deterministic by assuming that the energy consumption is predetermined. An (2020) developed a stochastic integer program for optimal siting of charging stations and fleet sizing with uncertain demand. A few studies focus on the layout of wireless lanes for electric bus systems. Liu and Song (2017) studied the robust optimization problem of locations of wireless charging facilities and battery sizes for battery-electric bus systems. Bi et al. (2018) proposed an optimization model for locating wireless charging bus stops, which undertake opportunity charging when loading and unloading passengers.

2.2 Operational planning in public transport

Public transport planning includes network design, frequency setting, timetabling, and vehicle/crew scheduling (Ceder, 2004; Wu et al., 2019). Besides those planning procedures, another stream of research is dedicated to the design of control strategies at the operation phase, such as holding control (Wu et al., 2017) and speed adjustment (Argote-Cabanero et al., 2015; Laskaris et al., 2020). As this paper is primarily concerned with the vehicle scheduling at the planning phase, in what follows we review only existing relevant research on this direction.

The vehicle scheduling problem has been studied since the early 1970s and remains a prominent research topic because of its inherent complexity. The problem aims to assign a fleet of buses to complete a given set of trips within pre-specified start and end times. The vehicle scheduling problem can be divided into two categories: the single-depot vehicle scheduling problem (SDVSP) and the multiple-depot vehicle scheduling problem (MDVSP). In the MDVSP, some deadheading trips are allowed to share fleet resources. The SDVSP can be solved in a polynomial time, whereas the MDVSP has been proven to be NP-hard. Hadjar et al (2006) addressed the MDVSP with a branch-and-cut approach. Shen et al. (2016) developed a model for vehicle scheduling with random trip time by refining the compatibility of trip pairs. Uçar et al. (2017) proposed a recovery method to handle the disruptions in the MDVSP. He et al. (2018) addressed the

vehicle scheduling problem with random trip time using an approximate dynamic programming approach. Kulkarni et al. (2018) devised a new inventory formulation for the MDVSP, coupling with heuristics based on multiple-commodity network flow and Dantzig-Wolfe decomposition.

There also exist a handful of studies on the integrated optimization of vehicle scheduling and other procedures, such as integrated timetabling and vehicle scheduling (Petersen et al., 2013; Desfontaines and Desaulniers, 2018; Carosi et al., 2019) and integrated vehicle and crew scheduling (Huisman and Wagelmans, 2006; Boyer et al., 2018).

2.3 Electric vehicle scheduling problems

With the rising adoption of electric buses around the world, the electric vehicle scheduling problem has attracted increased interest recently to support public transport management. The electric vehicle scheduling problem is a variant of the classic vehicle scheduling problem, where the characteristics of electric buses are explicitly considered, such as the limited running range, long charging time, and fixed charging stations. Li (2014) developed an SDVSP for electric buses considering limited energy. Li et al. (2019) formulated an MDVSP with a mixed bus fleet composed of electric and diesel buses. Tang et al. (2019) studied both static and dynamic versions for electric bus scheduling under stochastic traffic conditions, where the static model involves a buffer-distance strategy and the dynamic model reschedules the bus fleet. Yang et al. (2018) optimized the charge scheduling for in-motion wireless charging systems for electric buses. The objective is to minimize the total charging cost. Yao et al. (2020) optimized electric bus scheduling considering multiple vehicle types using a genetic algorithm. Perumal et al. (2020) integrated vehicle and crew scheduling for electric bus systems.

Another line of research concentrates on the recharging schedules and recharging process management for electric buses considering the charged amount (Kooten et al., 2017; Janovec and Kohání, 2019). Wen et al. (2016) presented an adaptive large neighborhood search algorithm to address the SDVSP, where both full recharging and partial recharging are allowed. Wang et al. (2017) developed a framework to optimize electric bus recharging schedules for both the planning and operation levels. Recently, He et al. (2020) investigated the optimal charging scheduling problem for the on-route fast-charging battery-electric bus system. Liu and Ceder (2020) investigated the battery-electric bus scheduling problem with consideration of the non-linear battery charging function, with the aim to minimize the required fleet size and battery chargers.

2.4 Objectives and contributions

The existing literature reveals that the electric vehicle scheduling problem is a new and important research topic from both theoretical and practical perspectives. Most existing studies are concerned with solving a transportation problem, assuming unlimited electricity supply and ignoring the potential risk in electricity peak load. As far as we are aware, there is no explicit analysis on the effect of power grid characteristics (e.g., TOU pricing and peak load risk) in the electric vehicle scheduling problem, even though it is highly relevant and critical for transportation electrification. The main contributions of this paper are therefore threefold. First, we explicitly consider TOU electricity tariffs and formulate a bi-objective electric transit vehicle scheduling problem in the generalized context of multi-route schedule

coordination to minimize the operation cost and peak load risk. Second, we reformulate the bi-objective optimization model by a powerful and practical lexicographic method. This method contributes to peak charging load leveling while reducing operation costs. Third, we propose an exact solution method based on a tailored branch-and-price method that embeds heuristics and a trip chain pool strategy to expedite the computation time. Our method is validated through a benchmark network and applied to a real-life bus network.

In the next section, the mathematical models are formulated. In Section 4, solution methodologies are described. In Section 5, computational experiments and an application are presented. Finally, the conclusions and future works are provided.

3. Modeling approach

3.1 Problem description

The electric vehicle scheduling problem aims to optimally assign a fleet of buses to complete a given set of trips, where each trip has a specific start time, end time, and corresponding terminals (including charging stations). The electric vehicles are allowed to be recharged at any of the given charging stations. A single overnight charging will be insufficient to perform a full day's schedule, indicating that within-day charging is essential to the schedule. The objective is to first minimize the operation cost considering the TOU pricing, and secondly to minimize the charging peak load. The operation cost includes the fixed transportation cost, variable transportation cost, waiting time cost, and charging cost.

To facilitate model development and without loss of generality, the following assumptions are made:

(A1) The electric buses will be fully charged overnight. Fast charging will be conducted during the operation period, and the buses will be fully charged once plugged into the grid. The charging power can be lower than the maximum power to reduce the degradation of the vehicles' batteries.

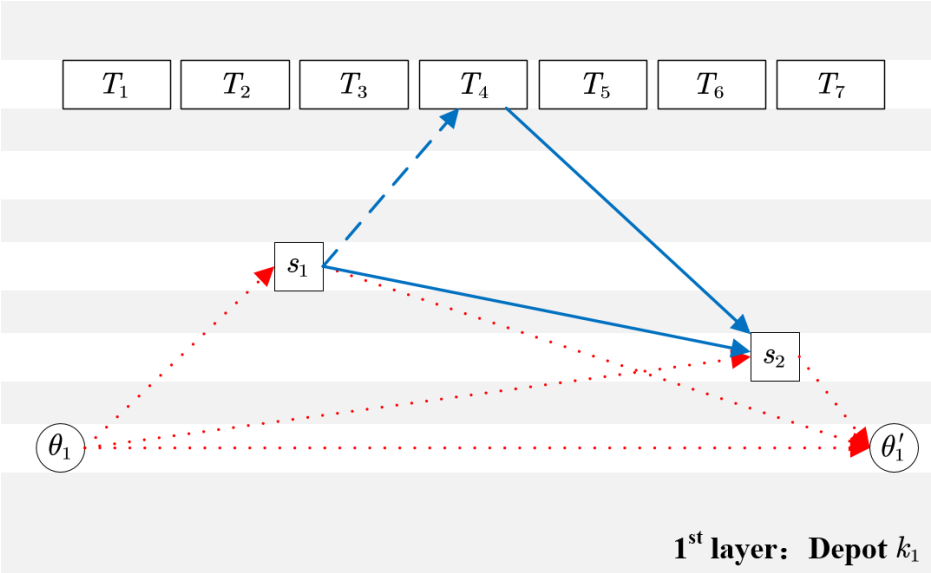
(A2) The charging time of each bus equals the full charging time. In practice, the charging time of each bus is nearly identical as the overall charging process includes fast charging (with large power) and trickle charge (with small power). In the latter process, the bus continues to draw electricity immediately after fast charging for a period of time to prolong the battery life. Even if the time spent in fast charging may differ depending on the state of charge (SoC), the trickle charge will continue until the required charging time is over.

(A3) The travel times are deterministic depending on the trip starting time. In practice, bus travel times can vary due to the stochastic nature of public transit attributes, such as fluctuating demand and changeable weather. Nevertheless, the effect of stochastic travel times can be alleviated by incorporating a proper slack time into scheduled travel time in the planning phase. For example, the scheduled travel time is commonly set as 85-percentile observed travel time to achieve a reliable scheme (Wu et al., 2016; Muller and Furth, 2000). In addition, the layover time at terminals can mitigate the stochastic travel times to a large extent.

(A4) The charging stations can be located at either the terminals or any specific location to which the buses go off-route.

3.2 Time-expanded network

In this section, we describe the electric transit system and our formulation of the MDEVSP with the effect of power grid characteristics in a time-expanded network. To represent the recharging activities at different times of the day, the planning horizon is discretized into a set of time windows. For simplicity, the length of each time window and the interval between time windows are set as the full charging time. We define a combination of charging station and time window (i.e., 'charging station-time window') as a charging activity/node. Let D denote the depot set; T denote the set of charging activities; S denote the trip set; Θ and Θ' denote the set of origin depots and destination depots, respectively. Each trip $s \in S$ can be defined as a tuple (st_s, et_s, sl_s, el_s), where st_s represents the trip starting time; et_s represents the trip ending time; sl_s is the origin depot; el_s is the destination depot. Likewise, each charging activity $u \in T$ can be defined as a tuple (st_u, et_u, sl_u, el_u), where $et_u - st_u$ equals the full charging time. The trip starting station is usually different from the trip ending station for a specific trip (i.e., $sl_u \neq el_u$), whereas the trip starting station is identical to the trip ending station for a charging activity (i.e., $sl_u = el_u$).



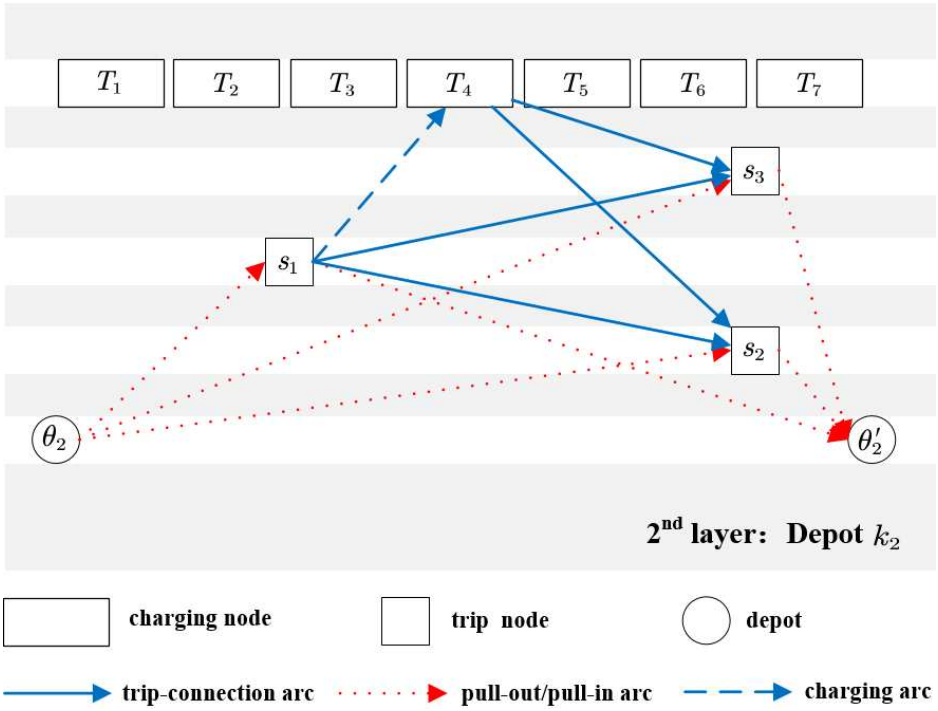


Fig. 2 Time-expanded network for MDEVSP

Since the buses need to pull out and pull in from a certain depot, it will be difficult to track the depot associated with a specific vehicle with only one layer. To realistically model the potential movement of vehicles, we construct a set of layers, with each one corresponding to a specific depot. As a result, the vehicle schedule can be represented by a directed graph $G_k = (V_k, A_k)$, in which V_k and A_k represent the set of nodes and arcs of layer k, respectively. Let Θ_k and Θ_k' represent the origin depot and destination depot of layer k, respectively. Fig. 2 depicts the parallel time-expanded networks of two layers. In the time-expanded network model, there is a network layer for each depot. A trip will be executed by a bus from a specific depot. For example, trip s_1 and s_2 can be executed by the bus dispatching from depot k_1, while trip s_1, s_2 and s_3 can be executed by the bus dispatched from depot k_2. A node stands for a specific activity, including trip dispatch from the depot, trip return to the depot, a bus trip, or a charging activity. The trip node and charging node are characterized by the starting time, ending time, starting station, and ending station, where the starting and ending stations are physical locations.

An arc represents a feasible connection between two activities, and there are pull-out arcs, pull-in arcs, trip-connection arcs, and charging arcs. The arc capacity is 1, ensuring that each arc can only be executed by one vehicle. The pull-out arc connects a depot to a trip node, which represents a vehicle dispatching from the depot to begin its first trip. The pull-in arc connects a trip node to a depot, which represents a vehicle completing the trip task and returning to the depot. The trip-connection arc connects two trip nodes or a trip node and a charging node. There are two possible scenarios for the trip-connection arc: (a) a vehicle is held at the current depot and executes the next trip; (b) a vehicle runs empty to another depot and executes the next trip after holding. The charging arc connects a trip node to a charging node, which indicates a vehicle running empty to the charging station and undertaking a recharging activity.

For a feasible trip-connection arc $\{(i,j) \in A_k | i \in S \cup T, j \in S\}$ or charging arc $\{(i,j) \in A_k | i \in$

$S, j \in T\}$, the following condition should be met: $st_j - et_i - DH_{ij} \geq 0$, that is, the trip starting time should exceed the prior trip ending time plus the possible deadheading time between the predecessor and the current node, where DH_{ij} represents the deadheading trip time from the ending point of node i (el_i) to the starting point of node j (sl_j). A feasible trip chain consists of a number of trip nodes and charging nodes, with starting depot being Θ_k and ending depot being Θ'_k, while satisfying the maximum running range constraint. The objective of MDEVSP is to optimally find a set of trip chains to cover trip tasks for a whole day, while satisfying a series of constraints such as depot and charging station capacity.

3.3 Arc costs

In the MDEVSP, the system costs are comprised of fixed transportation cost, variable transportation cost, waiting time cost, and charging cost. The fixed transportation cost depends on the number of vehicles required, while the variable transportation cost is determined by the travel distance. The charging cost consists of fixed charging cost and variable charging cost. The fixed charging cost is associated with the number of charge cycles, which results from the maintenance cost, crew cost, and battery degradation. On the other hand, the variable charging cost is dependent on the charging time, charging power, and TOU electricity tariffs. Evidently, with TOU pricing, the charging cost will be higher if recharging activities are to be undertaken during a high-tariff period. Therefore, the vehicle scheduling scheme under TOU pricing may exert great influence on the charging cost. For a charging activity $u \in T$, the charging cost can be calculated as follows:

$$c_b^u = P \cdot \int_{st_u}^{et_u} W(t)dt + c_f \tag{1}$$

where P is the charging power; $W(t)$ is the electricity price at time t; st_u and et_u are the starting time and ending time of charging activity u, respectively; c_f is the fixed charging cost.

As a result, the arc cost takes the following piece-wise function:

$$c_{ij}^k = \begin{cases} c_d \cdot d_{ij}^k + c_v, & pull-out\ \ arc \\ c_d \cdot d_{ij}^k + c_w \cdot ID_{ij}, & trip-connection\ \ arc \\ c_d \cdot d_{ij}^k + c_w \cdot ID_{ij} + c_b^j, & charging\ \ arc \\ c_d \cdot d_{ij}^k, & pull-in\ \ arc \end{cases} \tag{2}$$

where d_{ij}^k denotes the deadheading distance from the ending point of node i (el_i) to starting point of node j (sl_j). We note that the distances between the depot and each trip node can vary among different depot layers, although the deadheading distances between trip nodes (or between the trip node and the charging node) are the same. In other words, the lengths of deadheading trip-connection arcs (or charging arcs) are the same for different depot layers, whereas the lengths of deadheading pull-out arcs and pull-in arcs can vary among different depot layers. c_d denotes the transportation cost per unit kilometer; c_w denotes the waiting time cost per unit time with the waiting time being $ID_{ij} = st_j - et_i - DH_{ij}$; c_v denotes fixed transportation cost per unit vehicle; c_b^j denotes the variable charging cost of activity (node) j.

3.4 Optimization models

As the power required for during-the-day fast charging is very high, the scheduling scheme of electric buses should consider the grid load of power systems to distribute charging demand throughout different times of the day so as not to overload the grid. This would however require that some electric buses are charged during the high-tariff period, incurring excess charging costs. Thus there are trade-offs between the objectives of operation cost minimization (i.e., charging in the low-tariff period) and charging load leveling. In this paper, we formulate the MDEVSP with power grid characteristics as a bi-objective optimization problem, to address the trade-off of these dual objectives.

Prior to the optimization model formulation, the decision variables and associated parameters are introduced. Let x_{ij}^k denote a binary decision variable, $x_{ij}^k = 1$ indicates that a bus is assigned to node j after node i in the layer for depot k; otherwise, it is 0. Let g_i^k denote a continuous decision variable standing for accumulated distance traveled to node i in the layer for depot k, which is to ensure that the accumulated distance does not exceed the maximum running range. Let c_{ij}^k denote the cost of arc (i,j) in the layer for depot k; d_{ij}^k denote the distance from node i to node j in the layer for depot k; G denote the maximum running range that a fully-charged bus allows; N denote the fleet size; V_{Θ_k} denote the capacity of depot k; V_{T_j} denote the charging station capacity corresponding to charging activity T_j; L is a variable that denotes the charging peak load, which refers to the maximum number of buses charged simultaneously at charging stations.

The objectives of MDEVSP are to find the vehicle-to-trip assignment and charging plan that minimize the total cost from the system perspective, while reducing the peak charging load. With the costs discussed above, the MDEVSP can now be formulated as the following bi-objective mixed-integer nonlinear programming:

$$min \ \sum_{k\in K}\sum_{(i,j)\in A_k} c_{ij}^k x_{ij}^k \tag{3}$$

$$min \ L \tag{4}$$

s.t.

$$\sum_{k\in K}\sum_{i:(i,j)\in A_k} x_{ij}^k = 1 \qquad \forall j \in S \tag{5}$$

$$\sum_{i:(i,j)\in A_k} x_{ij}^k - \sum_{i:(j,i)\in A_k} x_{ji}^k = 0 \qquad \forall j \in S \cup T; \ k \in K \tag{6}$$

$$g_j^k \le g_i^k + d_{ij}^k + \left(1 - x_{ij}^k\right)\cdot M \qquad \forall j \in S; \ k \in K; \ (i,j) \in A_k \tag{7}$$

$$g_j^k \ge g_i^k + d_{ij}^k - \left(1 - x_{ij}^k\right)\cdot M \qquad \forall j \in S; \ k \in K; \ (i,j) \in A_k \tag{8}$$

$$g_i^k = 0 \qquad \forall i \in \Theta_k \cup T, \forall k \in K \tag{9}$$

$$g_i^k + d_{ij}^k \le G + \left(1 - x_{ij}^k\right)\cdot M \qquad \forall j \in T \cup \Theta_k'; \ k \in K; \ (i,j) \in A_k \tag{10}$$

$$\sum_{k\in K}\sum_{j:(i,j)\in A_k} x_{ij}^k \le N \qquad \forall i \in \Theta_k \tag{11}$$

$$\sum_{j:(i,j)\in A_k} x_{ij}^k \le V_{\Theta_k} \qquad \forall i \in \Theta_k; k \in K \tag{12}$$

$$\sum_{k\in K}\sum_{i:(i,j)\in A_k} x_{ij}^k \le V_{T_j} \qquad \forall j \in T \tag{13}$$

$$\sum_{k\in K}\sum_{i:(i,j)\in A_k} x_{ij}^k \le L \qquad \forall j \in T \tag{14}$$

$$x_{ij}^k \in \{0,1\} \qquad \forall (i,j) \in A_k; \ k \in K \tag{15}$$

$$g_i^k \ge 0 \qquad \forall i \in \Theta_k \cup S \cup T \cup \Theta_k'; \ k \in K \tag{16}$$

$$L \in N_+ \tag{17}$$

Objective (3) is to minimize the total operation cost. Objective (4) is to minimize the charging peak load. Constraints (5) enforce trip covering such that each trip is assigned to only one vehicle. Constraints (6) enforce the flow conservation for the nodes of one layer other than the depots, thereby rendering each vehicle able to find a feasible path on the time-expanded network. Constraints (7) and (8) determine the accumulated distance traveled to node j, which equals the accumulated distance traveled to the predecessor plus the distance between the predecessor and the current node. Constraints (9) ensure that the accumulated travel distance is 0 when a bus immediately is dispatched from the depot or completes the recharging activity at a station. Constraints (10) ensure that the maximum running range cannot be exceeded. Constraints (11) guarantee that the total number of buses required should not exceed the fleet size. Constraints (12) force that the number of dispatched buses should not exceed the depot capacity. Constraints (13) guarantee that the total number of buses with the same charging activity does not exceed the charging station capacity. Constraints (14) guarantee that the peak load restriction cannot be violated for each charging station. Constraints (15)-(17) state the attributes of the decision variables.

3.5 Model reformulation

The problem formulated in (3)-(17) is a bi-objective optimization model. There are a few solution methods for this type of problem, such as the weighted sum method, epsilon-constrained method, and lexicographic method. Interested readers are referred to Deb (2001) and Ehrgott (2005) for a detailed description of these methods. A good vehicle scheduling scheme should achieve win-win situations benefiting two stakeholders, that is, a transit agency and a power system. Meanwhile, for the vehicle scheduling problem, operation cost saving is usually the top priority from the perspective of public transport authorities. The lexicographic method assumes that the objectives can be ranked in the order of importance, and solves a sequence of single-objective optimization problems without degrading the prior objective. Moreover, the lexicographic method does not require that the objective functions be normalized, as opposed to the weighted sum method. For this reason, we adopt the lexicographic method to address the bi-objective optimization model with different dimensions (cost and load) and set minimizing operation cost as the primary objective.

Thus, we first optimize the primary objective function (minimize cost) without considering the other objective function (peak load) or constraints (14) to obtain the minimum cost, denoted z_1^*. We then optimize the second objective function (minimize peak load) with constraints (5)-(17) plus an added constraint to ensure the operation cost is no greater than z_1^*. To find a set of Pareto solutions, we relax the added constraint with a parameter $rgap \geq 0$, so that the cost is no greater than $z_1^*(1 + rgap)$. The value of $rgap$ is the relative gap that controls the upper bound of total operation cost. Minimizing the peak load with different values of $rgap$ allows a set of Pareto solutions to be found.

The cost-minimization problem is described mathematically as follows:

[P1]

$$min \ \sum_{k \in K} \sum_{(i,j) \in A_k} c_{ij}^k x_{ij}^k$$

s.t.

Eqs: (5)-(13)

Eqs: (15)-(16)

Using the optimal solution $x_{ij}^{k}{}^{*}$ of problem [P1], we formulate the peak load minimization problem [P2] as follows:

[P2]

$$min\ \ L \tag{18}$$

s.t.

$$\sum_{k\in K}\sum_{(i,j)\in A_k} c_{ij}^{k} x_{ij}^{k} \le \sum_{k\in K}\sum_{(i,j)\in A_k} c_{ij}^{k} x_{ij}^{k}{}^{*}\,(1+rgap) \tag{19}$$

Eqs: (5)-(17)

When $rgap = 0$ the solution of [P2] provides the minimum peak load L for the minimum cost solution. [P1] and [P2] are mixed-integer linear programming problems, with binary decision variables x_{ij}^{k} and continuous decision variables g_i^k. Problem [P2] also includes the integer variable L. Problem [P1] is an NP-hard problem, where small instances can be readily solved by commercial solvers (such as CPLEX). However, solving larger instances of problem [P1] by commercial solvers is not an easy job because of the formidable number of variables in the model. According to our preliminary experiments, CPLEX fails to obtain a feasible solution for a medium-size instance (400 trips, see Table 1) even after two days of computation. In the following section, we devise a branch-and-price method to solve [P1]. Problem [P2] is a mixed-integer linear programming problem, but it can be solved with an off-the-shelf solver.

Remark 1: The solution approach above can be viewed as a form of the epsilon-constrained method where the right-hand side of constraints (19) is the epsilon value (ε). Thus, the set of Pareto solutions can be obtained by adjusting the value of ε, or $rgap$. For discussion on implementation of the epsilon-constrained method, see Mavrotas (2009).

4. Branch-and-price method

4.1 Preliminaries

The multi-depot vehicle scheduling problem (MDVSP) has been proven to be NP-hard. Therefore, MDEVSP is also NP-hard since MDVSP is only a special case of MDEVSP. In this study, we devise an exact algorithm to address this NP-hard problem. The cost-minimization problem [P1] is recast as a trip chain selection problem, where each trip chain satisfies constraints pertaining to fleet size limitation, depot/charging station capacity, and trip covering (see Section 4.2). Given a large number of possible trip chains, it is almost impossible to enumerate all feasible trip chains in an acceptable amount of time. Nevertheless, the problem can be tackled by a well-designed branch-and-price method (Barnhart et al., 1998). In this part, we devise a tailored branch-and-price method to address MDEVSP. In particular, we devise a couple of reinforcements and heuristics to help to find approximate solutions in a reasonable time for large-scale scenarios.

As a critical component of the branch-and-price method, column generation is an efficient algorithm for solving large-scale linear optimization problems and is widely used to solve vehicle and aircraft scheduling problems (Li, 2014; Zeighami and Soumis, 2019). Motivated by this fact, we use the column

generation algorithm to efficiently find high-quality trip chains. Instead of handling all trip chains, the column generation algorithm solves the problem using only a subset of incumbent trip chains, which we refer to as the restricted master problem (RMP). In the column generation algorithm, if the newly generated trip chain contributes to improving the RMP, it will be added onto the trip chain pool (see Section 4.7); otherwise, the process is terminated.

To illustrate the concept of trip chain generation, as shown in Fig. 3, suppose there is a matrix where each column represents a feasible trip chain satisfying trip-connection constraints. A trip chain is composed of a depot set, a trip set, and a charging station-time window set, which represents the origin depots, service trips, and charging activities, respectively. These elements of a trip chain can be mapped into the nodes of the time-expanded network. The service trips in the trip set are sorted in chronological order according to the departure time. Since each trip can be executed by one electric bus in a specific period, the trip chain and the corresponding cost can be uniquely determined given the set elements. Through solving the RMP, the dual variables of constraints can be obtained. On the left side of Fig. 3, there are 6 trip chains in the trip chain pool. Suppose the incumbent optimal (feasible) solutions are trip chains 3 and 4. By solving the sub-problem using dual variables, a new column with minimum reduced cost that contributes most to improving the master problem can be identified (7^{th} column). This column is then added to the master problem, yielding the optimal solution: trip chains 2 and 7, as shown on the right side of Fig. 3. The new column can improve the master problem only with a negative reduced cost. This process is repeated iteratively until the master problem cannot be improved.

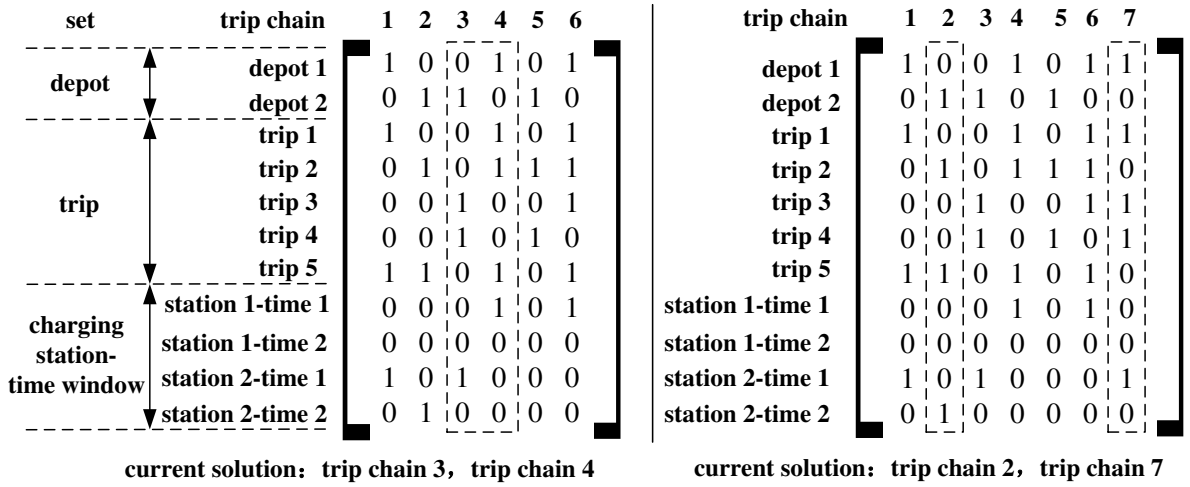


Fig. 3 Illustration of column generation for MDEVSP

Note that the solutions to the master problem by column generation cannot guarantee integrality. To address this issue, the branch-and-bound algorithm can be used to obtain integral solutions. Integration of a branch-and-bound algorithm and column generation yields the branch-and-price algorithm. Nevertheless, as noted by Barnhart (1998), such integration is not a trivial task since the branching constraints are likely to destroy the structure of the sub-problems. However, to improve the algorithmic efficiency, some customized strategies can be devised and embedded into the branch-and-price method.

Fig. 4 shows the flowchart of the branch-and-price method. The set of initial columns (trip chains) are constructed by heuristics (see Section 4.6.1) and form a trip chain pool, which corresponds to the root node

in the search tree. In the initialization, the upper bound is set as a sufficiently large value, and the root node is added to a list that is used for recording unexplored nodes in the search tree. If the nodes on the list remain, then one of them is selected and the corresponding linear programming relaxation of the set partitioning problem is solved by column generation, which provides a tight lower bound. If the solution is worse than the upper bound, the node is removed. If the solution is better than the upper bound and integer feasible, the upper bound is updated and the incumbent node is removed. If the solution is better than the upper bound and integer infeasible, a heuristic (see Section 4.6.2) is adopted to seek a better integer solution, thereby branching on this node.

Solving the linear relaxation problem independently for each node in the branch-and-price tree is usually time-consuming. If the trip chains can be known in advance, the redundant computation can be avoided, which reduces the computation burden. To this end, a trip chain pool strategy is proposed to handle column generation sub-problems and trip chains generated by heuristics (see Section 4.6). At the beginning of the column generation, feasible trip chains with branching information (see Section 4.5) are drawn from the trip chain pool and added onto the RMP. Note that the trip chains with branching feasible are identified one-by-one. Specifically, the branching conditions (see Section 4.5) will prevent the vehicle from going through specific arcs on the time-expanded network; if these arcs are not included in the trip chain, then the trip chain ensures branching feasibility.

When new trip chains are generated by the sub-problem, they are added onto the trip chain pool and RMP again. The stop condition can be associated with: (i) empty list; (ii) maximum allowable computational time; and (iii) the maximum number of visited nodes.

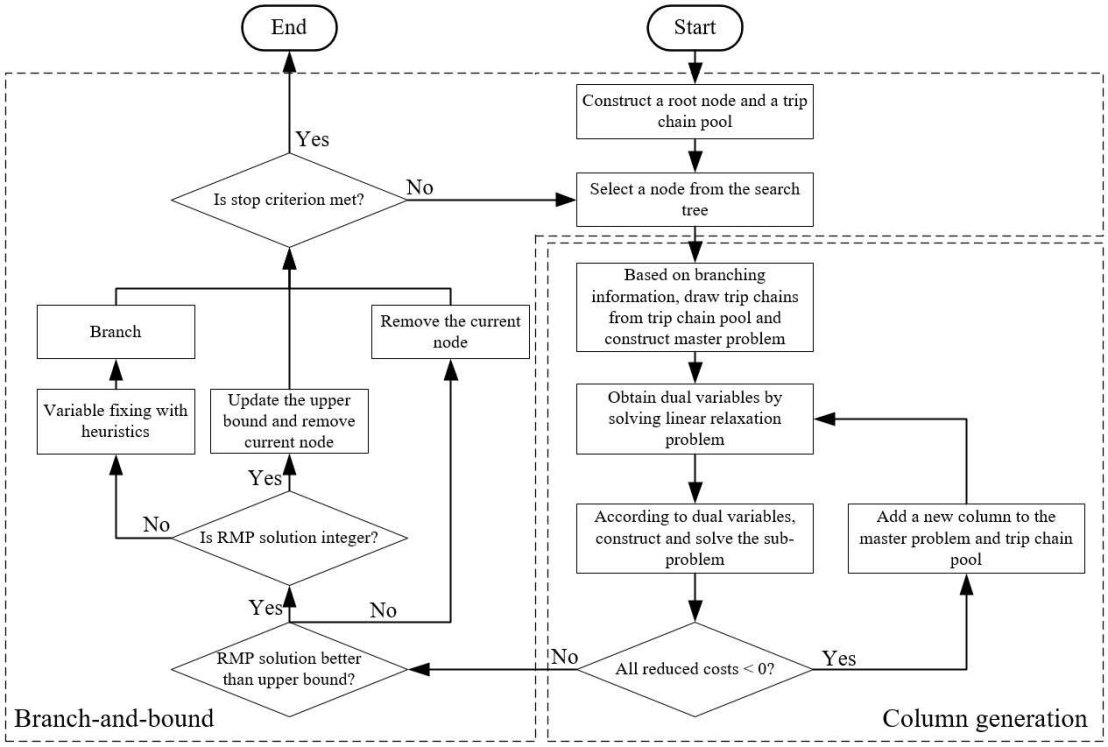


Fig. 4 Flowchart of the branch-and-price method

4.2 The column generation master problem

Recall that a trip chain is a sequence starting and ending with a depot and containing charging stations and service trips. Let R be the set of feasible trip chains in the RMP. Then, the RMP can be formulated as the following set partitioning problem:

$$min \sum_{r\in R} c_r z_r \tag{20}$$

s.t.

$$\sum_{r\in R} \delta_k^r z_r \le V_{\Theta_k} \qquad \forall k \in K \tag{21}$$

$$\sum_{r\in R} \delta_s^r z_r = 1 \qquad \forall s \in S \tag{22}$$

$$\sum_{r\in R} \delta_u^r z_r \le V_{T_u} \qquad \forall u \in T \tag{23}$$

$$\sum_{r\in R} z_r \le N \tag{24}$$

$$z_r \in \{0,1\} \qquad \forall r \in R \tag{25}$$

where z_r, $r \in R$ denotes the binary decision variable, and $z_r = 1$ if trip chain r is undertaken by an electric bus in the fleet. c_r denotes the total operation cost of trip chain r. δ_k^r, $k \in K$ denotes the trip chain incidence coefficient equal to 1 if trip chain r starts from depot k. δ_s^r, $s \in S$ denotes the trip chain incidence coefficient equal to 1 if service trip node s is covered by trip chain r. δ_u^r, $u \in T$ denotes the trip chain incidence coefficient equal to 1 if charging node u is covered by trip chain r. K, S, and T denote the set of depots, service trips, and charging activities, respectively. Note that δ_k^r, δ_s^r and δ_u^r are state variables, of which the values can be acquired given a trip chain. The trip chain generation method will be discussed later in Section 4.3.

In the RMP, the objective function given by Eq. (20) represents the total operation costs. Constraints (21) guarantee that the station capacity should be satisfied. Constraints (22) denote that each trip can be served only once. Constraints (23) ensure the charging station capacity should be satisfied. Constraints (24) ensure that the number of vehicles required is no more than the fleet size. Constraints (25) stand for the attribute of decision variables.

To generate the dual variables corresponding to constraints, the linear relaxation of RMP can be repeatedly solved by the column generation algorithm with only a subset of trip chains. In this way, the binary variable z_r becomes a continuous variable ranging from 0 to 1.

4.3 The column generation sub-problem (pricing problem)

For the minimization problem, only the columns with negative reduced costs contribute to improving the solutions. In other words, when the reduced costs of all feasible columns are larger than 0, any newly added column cannot improve the results, and the problem has been solved to optimality. The purpose of the column generation sub-problem (pricing problem) is to generate columns with negative reduced costs and feed them to the RMP. Let dual variables α_k, β_s, γ_u, $\pi(k \in K, s \in S, u \in T)$ correspond to constraints (21)-(24), respectively, then the reduced cost of trip chain $r \in R$ is:

$$c_r - (\sum_{k\in K} \delta_k^r \alpha_k + \sum_{s\in S} \delta_s^r \beta_s + \sum_{u\in T} \delta_u^r \gamma_u + \pi) \tag{26}$$

The trip chain cost c_r can be formulated by the arcs of the time-expanded network:

$$\begin{aligned} c_r &= \sum_{k\in K} \sum_{(i,j)\in A_k} c_{ij}^k x_{ij}^k \\ &= \sum_{k\in K} \sum_{(i,j)\in A_k, i\in\Theta_k} c_{ij}^k x_{ij}^k + \sum_{k\in K} \sum_{(i,j)\in A_k, i\in S} c_{ij}^k x_{ij}^k + \sum_{k\in K} \sum_{(i,j)\in A_k, i\in T} c_{ij}^k x_{ij}^k \end{aligned} \tag{27}$$

where the trip chain incidence coefficients δ_k^r, δ_s^r, and δ_u^r are equivalent to the following forms as a function of the variable x_{ij}^k:

$$\delta_k^r = \sum_{(i,j)\in A_k, i=\Theta_k} x_{ij}^k \tag{28}$$

$$\delta_s^r = \sum_k \sum_{(i,j)\in A_k, i=s} x_{ij}^k \tag{29}$$

$$\delta_u^r = \sum_k \sum_{(i,j)\in A_k, i=u} x_{ij}^k \tag{30}$$

Substituting Eqs. (27)-(30) into Eq. (26) yields the reduced cost of trip chain $r \in R$ in the form of the time-expanded network:

$$\begin{aligned} &\sum_{k\in K} \sum_{(i,j)\in A_k, i\in\Theta_k} (c_{ij}^k - \alpha_k) x_{ij}^k + \sum_{k\in K} \sum_{(i,j)\in A_k, i\in S} (c_{ij}^k - \beta_i) x_{ij}^k \\ &+ \sum_{k\in K} \sum_{(i,j)\in A_k, i\in T} (c_{ij}^k - \gamma_i) x_{ij}^k - \pi \end{aligned} \tag{31}$$

Therefore, the reduced cost of arc (i,j) corresponding to depot k can be formulated as the piecewise function:

$$\overline{c_{ij}^k} = \begin{cases} c_{ij}^k - \alpha_k, \forall j \in \Theta_k & \forall k \in K \\ c_{ij}^k - \beta_j, \forall j \in S & \forall k \in K \\ c_{ij}^k - \gamma_j, \forall j \in T & \forall k \in K \end{cases} \tag{32}$$

The pricing sub-problem can be formulated based on the arc-specific reduced cost. The purpose is to optimally find a path with minimum reduced cost, i.e., the shortest path. Since the constraint on travel distance range is enforced, the pricing problem becomes an elementary shortest path problem with resource constraints (ESPRC), which can be formulated as follows:

(SUB_COST)

$$min \ \sum_{k\in K} \sum_{(i,j)\in A_k} \overline{c_{ij}^k} x_{ij}^k - \pi \tag{33}$$

s.t.

$$\sum_{k\in K} \sum_{i:(i,j)\in A_k} x_{ij}^k = 1 \qquad \forall i \in \Theta_k \tag{34}$$

Eqs: (6)-(10)

Eqs: (15)-(16)

In the pricing problem, the decision variables are x_{ij}^k and g_i^k, where x_{ij}^k and g_i^k are respectively binary variables and continuous decision variables, whose specifics have been described in Section 3.2. Objective (33) is to minimize the reduced cost of trip chains. Constraints (34) ensure that exactly one of the trip chains starts from an origin depot.

4.4 Multi-label correcting method for solving ESPRC

With multiple layers in the time-expanded network, the shortest path from depot θ_k to θ_k' can be calculated independently for each layer $k \in K$, and can be accomplished by parallel computing. Once the shortest path of each layer is found, the optimal solution of the sub-problem can be determined by taking the path with the minimum cost among all layers. Due to the presence of possible negative reduced cost $\overline{c_{ij}^k}$, arcs with negative weights may exist in the network. As such, the label-setting method is inappropriate to solve the ESPRC, and the multi-label method should work in a label-correcting way, whose specifics are described as follows:

4.4.1 Label definition

The premise for the label-correcting method is to define problem-specific labels. Let $\mathcal{p}_i^m$ denote the m-th path from source (origin depot) θ_k to node i. A label ℓ_i^m with two sources (c_i^m, g_i^m) is associated with $\mathcal{p}_i^m$. c_i^m represents the total cost of the m-th path from source θ_k to node i, that is, $c_i^m = \sum_{(i,j)\in\mathcal{p}_i^m} \overline{c_{ij}^k}$. g_i^m represents the accumulated distance traveled from source θ_k to node i of m-th path.

4.4.2 Label extension

The label ℓ_i^m is maintained by using forward dynamic programming. In the label-correcting algorithm, label ℓ_i^m of node i is extended to label $\ell_j^{m'}$ of node j along arc (i,j) on path $\mathcal{p}_i^m$. The label extension rule is described as follows:

$$c_j^m = c_i^m + \overline{c_{ij}^k} \tag{35}$$

$$g_j^m = \begin{cases} 0 & \forall j \in T \\ g_i^m + d_{ij}^k & others \end{cases} \tag{36}$$

The extension is performed only when the following maximum travel distance constraint is met:

$$g_i^m + d_{ij}^k \le G \tag{37}$$

Note that the accumulated travel distance should be reset if node j is a charging node.

4.4.3 Label domination

The efficiency of the label-correcting algorithm depends on the number of labels. When a label belongs to neither an optimal nor a feasible solution, the label and the corresponding path will be eliminated using label domination rules to expedite the computation time. Label ℓ_i^m dominates $\ell_i^{m'}$ if and only if

$$c_i^m \le c_i^{m'} \tag{38}$$

$$g_i^m \le g_i^{m'} \tag{39}$$

If label $\ell_i^{m'}$ is dominated by label ℓ_i^m, then label $\ell_i^{m'}$ and the corresponding path $\mathcal{p}_i^{m'}$ do not belong to an optimal solution. This is because each feasible extension to $\mathcal{p}_i^{m'}$ can also apply to $\mathcal{p}_i^m$ in a more cost-effective way. Hence, $\ell_i^{m'}$ and $\mathcal{p}_i^{m'}$ will be eliminated.

With the aforementioned components, the principle of the label-setting method can be described as follows. First, initialization is implemented before the commencement of the label-correcting method. Here, the label of source θ_k is set as $(0,0)$, which indicates that the cost at the origin depot and accumulated travel distance are both 0. In the label-correcting method, the node indices of updated labels are recorded, and one of the arcs originating from a node is selected before the next iteration. Here, the first-in-first-out principle is adopted, and thus the 'QUEUE' is used to store nodes. To handle a node, all the labels of the node are compared pairwise, and the redundant and inefficient labels are eliminated by domination rules. Subsequently, the label extension of the node is performed when the constraints (37) are satisfied. At last, when the QUEUE is empty, the optimal path and associated solution are obtained by a backtracking method. The pseudocode is provided in Algorithm 1:

Algorithm 1 Pseudocode of label-correcting method for ESPRC

Initialize $\ell_i^1 \leftarrow (0,0)$, for each $i \in \theta_k$
QUEUE = [θ_k]
While i = pop(QUEUE) % pop( ) withdraws the first element of the QUEUE
  $\{\ell_i^1, \ell_i^2, \dots, \ell_i^m\} \leftarrow$ LABEL_dominance($\{\ell_i^1, \ell_i^2, \dots, \ell_i^{m'}\}$)
  For each arc (i,j)
    For each path m in node i
      If $g_i^m + d_{ij}^k \le G$ % label extension

$$\ell_j^m \leftarrow \begin{cases} \left(c_i^m + \overline{c_{ij}^k}, 0\right) & \forall j \in T \\ \left(c_i^m + \overline{c_{ij}^k}, g_i^m + d_{ij}^k\right) & others \end{cases}$$

      End
    End
    If label extension is performed
      QUEUE.add(j)
    End
  End
End
Backtrace( )

4.5 Branching

To ensure integral optimal solutions, column generation is integrated into a branch-and-bound search framework and actuated at each node of the search tree, which results in a branch-and-price algorithm. When the optimal solution of the RMP is not integral once the column generation completes, the branch-and-bound process is conducted.

When developing the branch-and-price approach, the branching rule will have a great influence on the tree size and resulting algorithmic performance. To obtain an integral optimal solution in the MDEVSP, an alternative is to branch on the column variable z_r (trip chains) of the master problem. However, it will be difficult to branch on the selection scheme of z_r in a straightforward manner, since in this way the sub-problem structure would be destroyed, and the tree size will become cumbersome to solve. In practice, when the decision variable of a column is fixed to be 0, a similar column may be generated in the sub-problem, resulting in low branching efficiency. To reduce the size of the search tree, we adopt the branching rule on the connection between trip nodes, instead of branching on decision variables directly.

In doing so, when the solution of the linear relaxation of RMP is integer infeasible, the solution of the set partitioning formulation $\tilde{z}_r$ can be directly converted to that of the time-expanded network $\tilde{x}_{ij}^k$. More specifically, for each arc $(i,j) \in A_k$, let $\delta_k^r = 1$, $\delta_i^r = 1$, $\delta_j^r = 1$ if arc $(i,j) \in A_k$ is traveled in trip chain $r \in R$ on the time-expanded network. Then, the value of x_{ij}^k equals to the summation of $\tilde{z}_r$ of all trip chains going through arc $(i,j) \in A_k$, that is,

$$\tilde{x}_{ij}^k = \sum_{r \in R} \delta_k^r \delta_i^r \delta_j^r \tilde{z}_r \qquad \forall (i,j) \in A_k;\ k \in K \tag{40}$$

Thereafter, we can branch on the variables that are not fixed. Note that a fixed variable either equals 0 or 1, depending on the branching information. As a result, two sub-problems can be obtained, with

branching information relating to $\tilde{x}_{ij}^k = 0$ (called 0-branch) and $\tilde{x}_{ij}^k = 1$ (called 1-branch). For the 0-branch, the associated arc $(i,j) \in A_k$ can be simply disregarded from the time-expanded network. For the 1-branch, the attributes of node i and j are first identified. If node i is a trip, then the arcs in the set $A_i^+ = \{(i,h) \in A_k | h \neq j\}$ are disregarded; if node j is a trip, then the arcs in the set $A_j^- = \{(h,j) \in A_k | h \neq i\}$ are disregarded. Finally, if these disregarded arcs are included in the column, then this column should be removed from the master problem, where the details of column management are referred to the trip chain pool strategy (Section 4.7). With such a branching strategy, the branching information can be easily represented by the time-expanded network.

There is a global upper bound recording the best integral solution for the branch-and-bound tree. The integral solution results from heuristics or the nodes of RMP with integral solution. To obtain an integral solution quickly, the depth-first search rule is adopted. The node with $\tilde{x}_{ij}^k = 1$ is always selected when about to select a node on the same level.

4.6 Heuristics

As shown in the section above, the upper bound can be updated only when the solution of the linear relaxation of RMP is integral feasible. This raises a critical issue for the standard branch-and-price method in that, with the increase of problem size, it becomes more difficult to achieve an integral solution, since the linear relaxation of RMP is more likely to become integral infeasible. Although an exact algorithm can solve the problem to global optimality, it can only cope with a limited scale. In comparison, heuristics can achieve near-optimal solutions in a more time-efficient way. In this part, we devise a new heuristic algorithm that can be handled on the time-expanded network, such that the generated solution meets the branching condition in the underlying network. The principle is to reset arc costs continuously and obtain a trip chain by solving the shortest problem of the time-expanded network, whose solution approach is the same as that of the sub-problem (label-correcting method). After adding the generated trip chain onto the set of trip chains, some trips may be covered more than once resulting in infeasibility. Then, we remove the over-coverings until all the trips are covered only once.

As such, it is crucial to set appropriate arc costs. The fundamental idea for arc cost setting is that: i) the value is close to the fractional solution; ii) each trip chain covers as many trips as possible; and iii) the trip chain cost (and deadheading cost) should be as low as possible.

4.6.1 Heuristic for initial solution generation

The root node should be provided with an initial solution before the commencement of the algorithm. A good initial solution can maintain the master problem feasible and reduce the number of iterations of column generation. Generally speaking, the fixed transportation cost constitutes a large proportion of the total cost. Therefore, a good initial solution should require as few vehicles as possible, such that each vehicle can cover as many trips as possible. Taken together, the purpose of initial solution generation is that each vehicle covers as many trips as possible, while reducing the cost and satisfying the capacity and fleet size constraints. Based on this principle, the main steps for the initialization heuristic are described as follows:

Step 1 Construct the time-expanded network. The cost of arc (i,j) in the layer for depot k is set as $\mu + \nu \cdot c_{ij}^k$, where ν is a small value (here $\nu = 0.002$). If node j is a service trip node, then $\mu = -1$, and $\mu = 0$ otherwise. The fundamental idea of adding μ is to cover as many trips as possible.

Step 2 Obtain a trip chain by solving the shortest path in the underlying network. Then, the over-coverings are removed from the network, including the over-covered nodes, depot, and charging nodes with over capacity.

Step 3 If all the trips are covered, the algorithm is terminated. Otherwise, go to Step 2.

Remark 2: The trip chain generated by the heuristic satisfies trip connection and maximum running range constraints, while not necessarily satisfying the constraints of the master problem. There are two possible reasons for this. First, depot capacity constraints are set too strictly. In Step 2, the depots with excess capacity will be disregarded, and an extreme case may exist where all depots are removed and the shortest path cannot be found, although its probability may be very small. In this extreme case, the heuristic will be terminated in advance. Second, fleet size constraints are set too strictly. Although the trip chain generated by the heuristic can satisfy trip connection and depot/charging station capacity constraints, the fleet size constraints may be violated, such that the master problem is infeasible. Therefore, we add a costly artificial trip chain satisfying all constraints to guarantee the feasibility of the master problem. Specifically, for this trip chain such that $\delta_s^r = 1, \forall s \in S$, the corresponding cost is set as a large value (here it is 10,000,000).

4.6.2 Upper bounding heuristic

The solution of the sub-problem at each node of the search tree is not necessarily integral feasible. To obtain an integral solution quickly, here we devise heuristics to convert the fractional values into integral ones. This also facilitates providing the upper bound and reducing the tree size of branch-and-bound search. The fractional solution $\tilde{z}_r$ obtained from the RMP (19)-(24) can be converted into the arc-based fractional solution $\tilde{x}_{ij}^k$ through Eq. (39). In principle, this fractional solution should be close to its nearest integer value. However, the solution resulting from traditional rounding operations (i.e., rounding up, rounding down, and rounding off) could easily violate the equation constraints such as (5) and (6). Moreover, similar to the initial solution generation, the constructed trip chain should cover as many trips as possible, while reducing the cost and satisfying the capacity and fleet size constraints. Based on these considerations, we propose a variable fixing method, and the main steps are provided as follows:

Step 1 Select the trip chain having not ever been used with the largest fractional value.

Step 2 Construct the time-expanded network. The cost of arc (i,j) in the layer for depot k is set as $\lambda + \mu + \nu \cdot c_{ij}^k$ (the values of μ and ν are in line with Section 4.6.1). If node j is a node covered by the trip chain, then $\lambda = -10$, and $\lambda = 0$ otherwise. The fundamental idea of adding λ is to make the fixed trip chain 'resemble' the original one.

Step 3 Obtain a trip chain by solving the shortest path of the time-expanded network. Then, the over-coverings are removed from the network, including the over-covered nodes, depot, and charging nodes with over capacity. The resulting trip chain satisfies the maximum running range constraint without over-coverings.

Step 4 If all the trips are covered, the algorithm is terminated. Otherwise, go to Step 1.

Remark 3: The trip chain generated by the heuristic may not satisfy the constraints of the master problem. The reasons are similar to those discussed in Remark 2. In this case, the upper bounding heuristic can be simply skipped.

4.7 Acceleration technique: trip chain pool strategy

In the branch-and-bound tree search, column generation will be frequently undertaken at each node, which results in a great computational burden. If we can make use of the available information at the parent node, the redundant calculation can be avoided and the problem solving for the child nodes can be accelerated. Motivated by this fact, we devise an acceleration strategy, called trip chain pool R_{pool}, to collect all the trip chains generated by nodes of the search tree. Let the current node be p, then the initial trip chains R_p of the master problem of node p can be obtained from R_{pool} by removing those trip chains with branching infeasibility.

$$R_p = R_{pool} \backslash \{r: r \in R_{pool}, \text{with } r \text{ being branching infeasible for node } p\} \tag{41}$$

Typically, for the root node of search tree $root$, $R_{pool} = \emptyset$, R_{root} is comprised of two components: i) trip chains generated by heuristics (Section 4.6.1); and ii) an artificial trip chain with a large cost.

After solving the problem for node p, the trip chains by column generation and the trip chain pool are merged, resulting in a new trip chain pool:

$$R_{pool} = R_{pool} \cup R_p \tag{42}$$

5. Computational experiments and application

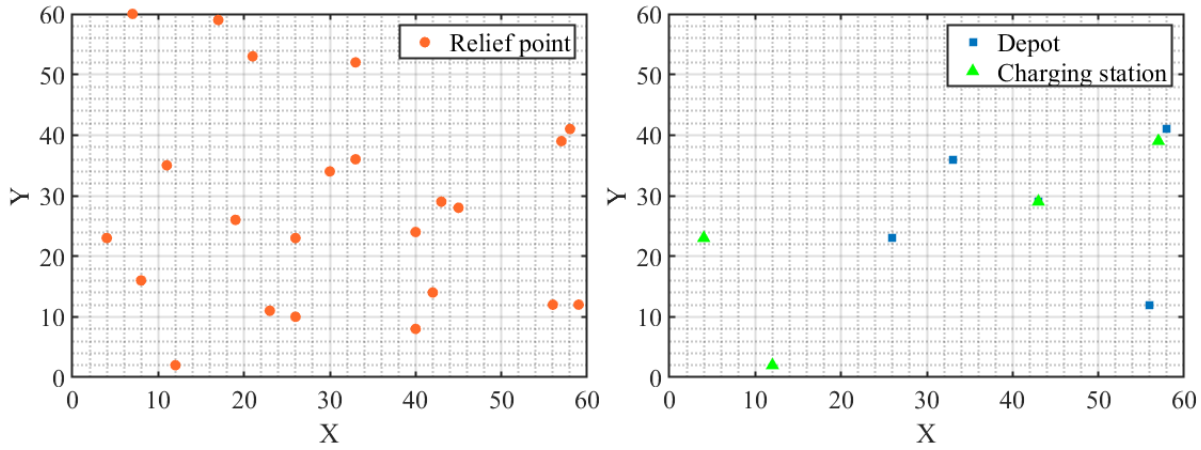
In this section, we first test the bi-objective optimization model and tailored branch-and-price algorithm in two ways: we use a benchmark network to evaluate the algorithm under different sizes. Then, we apply the proposed method to the real-world bus network in Guangzhou, China. For all experiments, the optimization problems are solved on an Intel(R) Core(TM) i7-8700 CPU @ 3.20 GHz core processor, 20GB RAM, and the system is Windows 10, using MATLAB R2018b and the CPLEX 12.9.0 solver with standard tuning.

5.1 Numerical test

5.1.1 Benchmark instances

We randomly generated data for testing using the method as proposed by Adler and Mirchandani, (2017). We considered the scenarios with $|S|$ trips, l charging stations, and $|K|$ depots. To this end, we first randomly generated a set of relief points v from a 60*60 grid ($v \in (\sqrt{|S|}, 2\sqrt{|S|})$, where v is an integer. The relief points served as a set of potential starting and ending locations for the service trips as well as potential locations for the depot and charging stations. Among these relief points, we randomly selected $|S|$ trips, l charging stations, the starting point, and the ending point of the $|K|$ depots. Note that the depots, the charging stations, the starting/ending points of the trips can share the same relief point, whereas the starting point and ending point of the trips cannot.

As an illustration, suppose there are 5 depots and 4 charging stations to be generated. As shown in Fig. 5(a), a total of 23 relief points are first generated. 5 depots and 4 charging stations are then randomly selected from these relief points, and the results are shown in Fig. 5(b). The coordinates of depots are (26,23),(33,36),(43,29),(56,12),(58,41), and the coordinates of charging stations are (4,23),(12,2),(43,29),(57,39), where the relief point (43,29) acts as both a depot and a charging station.



(a) (b)

Fig. 5 Instance generation: (a) relief points; (b) depots and charging stations

For each trip $s \in S$, we assume that its travel distance $dist(sl_s, el_s)$ is the Euclidean distance between two points, with start time st_s being a value chosen from [420,480] with a probability of 15%, (480,1020] with a probability of 70%, and (1020,1080] with a probability of 15%. The trip travel time is randomly chosen from $(dist(sl_s, el_s) + 5, dist(sl_s, el_s) + 40)$.

In the MDEVSP, deadheading trips are often needed to connect trips for interlining services or to recharge buses. We also assume that the travel distance and travel time for deadheading are equal to the Euclidean distance between two points. The depot capacity is an integer randomly chosen from $(3 + |S|/(3.5|K|), 3 + |S|/(2.5|K|))$, the charging station capacity is an integer randomly chosen from $(3 + |S|/(3.5l), 3 + |S|/(2.5l))$, and the fleet size is an integer randomly chosen from $(dist(sl_s, el_s) + 5, dist(sl_s, el_s) + 40)$.

For the operational setting of electric buses, following the literature (An, 2020), the maximum running range G is set as 250 km. This setting is in accordance with the BYD K7 bus with a vehicle capacity of 58 pax and charging time of 60 min. The operation cost of an electric bus is adapted from Li (2014), where the fixed transportation cost c_v is taken as 100 USD per vehicle, and the variable transportation cost c_d is taken as 0.4 USD/km. The waiting time cost c_w is 0.1 USD/min, and the fixed charging cost c_f is 5 USD per time. The charging power is taken as 200 kW. The time-of-use electricity price in Guangzhou is adopted, as shown in Fig. 1(a). The differential price can be divided into different periods, with the low tariff from 24:00 to 08:00 and the high tariff from 14:00 to 17:00 and from 19:00 to 22:00.

5.1.2 Solution comparison under different instance sizes

In order to test the algorithm under different instance sizes, we use various combinations of numbers of trips, depots, and charging stations. Specifically, we set the number of service trips $|S|$ as 50, 100, 200, and 400, the number of depots $|K|$ as 2, 5, and 8, and the number of charging stations as 2, 4, 6, and 8. As

a result, there are totally $4 \times 3 \times 4 = 48$ possible combinations. For all these instances, the maximum number of visited nodes is set as 200. The maximum computational time is set as 12 h for the cases of 50, 100, and 200 trips $|S|$, and 24 h for the cases of 400 trips $|S|$ and 2 depots $|K|$, and 48 h for the cases of 400 trips $|S|$ and 5, 8 depots $|K|$. The branch-and-price algorithm is terminated when either of the above two criteria is satisfied. The computational results of the first objective are shown in Table 1.

Table 1 Computational results of the problem for the first objective

			CPLEX					Tailored branch-and-price				
$\|S\|$	$\|K\|$	l	The number of variables	Lower bound (USD)	Obj (USD)	Gap (%)	CPU(s)	The number of nodes visited	Lower bound (USD)	Obj (USD)	Gap (%)	CPU(s)
50	2	2	7684	2883.69	2884.75	0.04%	144	61	2884.75	2884.75	0.00%	165
		4	11072	2632.69	2638.66	0.23%	77	200	2636.91	2638.66	0.07%	771
		6	14780	2814.02	2815.77	0.06%	563	200	2813.61	2815.77	0.08%	1286
		8	18284	2704.25	2704.25	0.00%	2	35	2704.23	2704.23	0.00%	367
	5	2	19430	2882.46	2883.92	0.05%	53	200	2883.07	2883.92	0.03%	1232
		4	27520	2603.13	2603.30	0.01%	10	71	2603.13	2603.13	0.00%	1053
		6	37250	2730.50	2730.71	0.01%	28	1	2730.71	2730.71	0.00%	401
		8	45680	3049.91	3054.20	0.14%	87	200	3053.70	3059.74	0.20%	3215
	8	2	30736	2616.18	2616.18	0.00%	26	1	2616.18	2616.18	0.00%	282
		4	43808	2573.23	2574.31	0.04%	1410	1	2574.31	2574.31	0.00%	437
		6	60224	2017.34	2023.09	0.28%	43200	200	2021.78	2023.09	0.06%	5468
		8	71776	2873.32	2914.65	1.42%	43200	200	2895.23	2904.12	0.31%	5004
100	2	2	22492	6445.54	6446.34	0.01%	1146	163	6446.34	6446.34	0.00%	2176
		4	29824	5584.53	5596.37	0.21%	43200	200	5590.76	5592.23	0.03%	2991
		6	36972	5886.23	5886.84	0.01%	29748	200	5886.67	5887.60	0.02%	4717
		8	44728	3885.65	3885.65	0.00%	47	149	3885.65	3885.65	0.00%	7315
	5	2	55450	5142.64	5191.02	0.93%	43200	200	5169.68	5178.68	0.17%	6682
		4	74750	5444.28	5445.26	0.02%	18049	200	5444.48	5452.08	0.14%	9738
		6	90950	6590.12	6590.67	0.01%	131	200	6590.31	6590.67	0.01%	12977
		8	108980	4665.88	4665.88	0.00%	164	153	4665.88	4665.88	0.00%	16713
	8	2	91776	5023.41	5024.01	0.01%	21397	200	5023.51	5024.03	0.01%	12196
		4	116544	6113.47	6137.26	0.39%	43200	200	6120.90	6126.31	0.09%	15756
		6	146464	5183.19	5183.19	0.00%	258	1	5183.19	5183.19	0.00%	3499
		8	174704	5007.17	5007.43	0.01%	417	200	5007.43	5007.43	0.00%	22984
200	2	2	76624	10865.16	10988.47	1.12%	43200	200	10886.38	10937.29	0.47%	16810
		4	89920	9838.20	10636.13	7.50%	43200	200	9924.27	9944.60	0.20%	27765
		6	101068	11517.03	11526.68	0.08%	43200	200	11524.93	11526.74	0.02%	23892
		8	117872	10733.87	10734.54	0.01%	399	200	10733.87	10733.87	0.00%	36063
	5	2	187820	10089.45	15789.55	36.10%	43200	165	10105.91	11405.94	12.86%	43200

	4	220400	10572.91	15455.33	31.59%	43200	113	10586.09	11247.25	6.25%	43200
	6	256980	-	-	-	43200	90	9738.95	10686.76	9.73%	43200
	8	290620	-	-	-	43200	84	9796.68	10765.79	9.89%	43200
8	2	307760	9933.76	10205.96	2.67%	43200	69	9938.45	10744.68	8.11%	43200
	4	352000	-	-	-	43200	71	9832.51	10726.74	9.09%	43200
	6	407968	-	-	-	43200	38	10744.17	11733.65	9.21%	43200
	8	474192	9062.20	9100.03	0.42%	43200	27	9062.20	9974.93	10.07%	43200
400	2	270152	-	-	-	86400	136	19455.33	21483.23	10.42%	86400
	4	296492	-	-	-	86400	152	21027.27	23068.37	9.71%	86400
	6	330240	-	-	-	86400	67	19323.98	21151.25	9.46%	86400
	8	356140	-	-	-	86400	67	18903.31	20757.35	9.81%	86400
	2	674790	-	-	-	172800	45	20482.27	22343.92	9.09%	172800
	4	741430	-	-	-	172800	38	20919.62	23152.58	10.67%	172800
5	6	821590	-	-	-	172800	45	20146.15	21933.04	8.87%	172800
	8	891120	-	-	-	172800	16	20455.71	22416.49	9.59%	172800
	2	1072944	-	-	-	172800	1	-	22650.28	-	172800
	4	1204336	-	-	-	172800	1	-	19730.04	-	172800
8	6	1291536	-	-	-	172800	1	-	23376.07	-	172800
	8	1436960	19419.86	26182.67	25.83%	172800	1	-	20882.61	-	172800

In Table 1, the first three columns denote the number of service trips, the number of depots, and the number of charging stations, respectively. The fourth to eighth columns denote the results calculated by CPLEX, where the fourth column denotes the number of variables of the bi-objective integer programming model; the fifth column denotes the lower bound; the sixth column denotes the optimal integral solution found; the seventh column denotes the gap between the optimal solution found and the lower bound, calculated as $Gap = \frac{Obj - lower\ bound}{lower\ bound} \times 100\%$; the eighth column denotes the CPU time for the program calculation. The ninth to thirteenth columns are the results of the tailored branch-and-price method, where the ninth column denotes the number of visited nodes, and the implications of the tenth to thirteenth columns are identical to the fifth to eighth columns. The symbol '-' from the fifth to seventh columns represents a difficult instance where CPLEX fails to find a feasible solution within a given duration. The symbol '-' in the tenth and twelfth columns indicates that the calculation of the master problem at the root node has not been completed, such that the lower bound and the gap are not obtained. Nevertheless, thanks to the heuristics at the root node, our tailored branch-and-price method can still obtain a feasible solution.

As we can see, as the network size increases, the number of nodes to be explored in the search tree does not necessarily increase. However, the computational time of the branch-and-price method increases significantly with the network size and the instances with 400 trips cannot be solved to optimality within 24 hours. This is because of the large number of columns to be generated and more time spent in solving the pricing problem. The solution quality, which is represented by the gap, appears to be most sensitive to the number of depots. For example, for the instance with 200 trips, as the number of depots increases from 2 to 5, the average gap increases significantly from almost 0 to higher than 6% and 31% for the

branch-and-price method and CPLEX, respectively. This suggests that to ensure the solution quality for the MDEVSP, one should pay more attention to the number of bus routes.

The optimal solution can be obtained by CPLEX quickly for small-size instances. However, when the number of variables exceeds 20,000, it is difficult for CPLEX to find the optimal solution or even a feasible solution. In comparison, although the tailored branch-and-price method obtains the optimal solution slower than CPLEX in small-size instances, it can output a feasible solution in large-size instances, with the average gap value below 10%. Nevertheless, in cases with more than 1 million variables, both CPLEX and the branch-and-price method have difficulty in finding feasible solutions. The gap increases with the number of trips, whereas the gaps by the branch-and-price method are lower than those by CPLEX in most scenarios. More importantly, the lower bounds obtained by the branch-and-price method are generally better than those obtained by CPLEX.

For the large-size instances with 400 trips and 8 depots, it is difficult for the branch-and-price method to complete solving the root nodes (and thus obtain the gaps and lower bounds) within limited computational time. Nevertheless, the tailored branch-and-price method can still find the initial integral solutions by the heuristics. Although the heuristic algorithms are deterministic (i.e., the results will not change under different runs given the same input), the performance of the tailored branch-and-price method under large-size scenarios may become unstable and more sensitive to the input (the layout of depots and charging stations). In other words, the solution quality will largely depend on the input data under large-size scenarios. Besides, the layout of depots and charging stations are randomly generated. This may explain why the results with 4 charging stations yield a substantially better total cost than any of the other scenarios with a higher number of charging stations.

5.1.3 Impact of the length of the time window

While the model assumes that the length of each charging time window equals the period necessary to fulfill a full charge in the time-expanded network for simplicity, in practice the charging time might not perfectly match those time windows. We now investigate how the length of the time window affects the system performance. To this end, each charging node is further discretized into a set of equidistant time steps. Let $p^k(t)$ denote the k-th time step of charging node t. The impact of charging time on the station capacity and charging load is handled by $p^1(t), p^2(t), \dots, p^U(t)$. As a result, the constraints (13) and (14) are modified as follows:

$$\sum_{k\in K}\sum_{i:(i,j)\in A_k} x_{ij}^k + \sum_{k\in K}\sum_{i:(i,j)\in A_k} x_{ip^1(j)}^k + \cdots + \sum_{k\in K}\sum_{i:(i,j)\in A_k} x_{ip^U(j)}^k \le V_{T_j} \qquad \forall j \in T \tag{43}$$

$$\sum_{k\in K}\sum_{i:(i,j)\in A_k} x_{ij}^k + \sum_{k\in K}\sum_{i:(i,j)\in A_k} x_{ip^1(j)}^k + \cdots + \sum_{k\in K}\sum_{i:(i,j)\in A_k} x_{ip^U(j)}^k \le L \qquad \forall j \in T \tag{44}$$

To evaluate the length of the time window on the solutions, we set the length of the time window from 10 min to 60 min with an interval of 10 min, where 60 min is the baseline in this example. To conduct a fair comparison, the gaps for all instances are set as 1%, indicating that the program will be terminated when the gap is immediately lower than 1%. 3 instances with 200 trips are selected for testing. All the instances are run by CPLEX, and the results are summarized in Table 2.

Table 2 Comparison of solutions under different lengths of time windows

$\|S\|$	$\|K\|$	l	Time window (min)	The number of variables	Lower bound (USD)	Obj (USD)	Gap (%)	CPU(s)
200	2	2	10	141556	10865.86	10974.38	0.99%	262221
			20	102632	10865.52	10971.12	0.96%	268865
			30	89640	10866.12	10969.65	0.94%	258015
			40	83128	10865.80	10972.69	0.97%	137857
			50	79196	10865.23	10974.30	0.99%	114557
			60	76624	10866.35	10973.41	0.98%	387847
200	2	8	10	375444	10733.87	10811.12	0.71%	506
			20	220896	10733.87	10760.04	0.24%	546
			30	169348	10733.87	10821.11	0.81%	129
			40	143492	10733.87	10810.21	0.71%	1661
			50	128104	10733.87	10737.25	0.03%	255
			60	117872	10733.87	10839.08	0.97%	1152
200	8	2	10	571024	9933.67	10027.99	0.94%	2606
			20	413120	9933.76	10020.75	0.87%	164060
			30	360480	9933.76	10027.99	0.36%	307316
			40	334304	9933.76	10033.95	1.00%	344009
			50	318464	9933.76	10029.52	0.95%	508658
			60	307760	10865.86	10974.38	0.67%	410848

We can see in Table 2 that, as the length of the time window increases, the number of variables decreases, while the effect on the lower bounds and optimal solutions are minimal. Mathematically, the feasible solutions under the time window of 20 min are included in those of 10 min. However, the experiment shows that the solutions under the time window of 10 min are not necessarily the best. This is because the model has not been sufficiently solved to optimality given the terminated gap. Notwithstanding that, the lower bounds under different values of time windows are quite close and the corresponding gaps are all lower than 1%. This suggests that the impact of the length of the time window on the model solutions is limited. Therefore, our model is scalable to incorporate the possibility of opportunistic charging without requiring the vehicle to spend the full charge time.

In addition, for this example, the CPU time is not necessarily proportional to the number of variables. These results are closely associated with the mechanism of an off-the-shelf solver. The possible reason is that for some typical instances the program could find the lower bound more readily.

5.1.4 Impact of the second objective

We now investigate the effect of the second objective, i.e., peak load minimization, by using a lexicographic method. To be representative, the results for a large-size instance with 400 trips are used as the initial solution for the second objective. We call CPLEX with the computer configuration as described above. To obtain a better solution, we set the frequency of RINS as 20, where RINS is a powerful but

expensive test for finding high-quality feasible solutions embedded in the off-the-shelf solver. The maximum computation time is set as 48 h, and the results are provided in Table 3. Note that the peak load is represented by the number of electric buses being charged concurrently. Naturally, a larger number of electric buses being charged simultaneously indicates a higher peak load for the power grid.

Table 3 Bi-objective results by using a lexicographic method

$\|S\|$	$\|K\|$	l	First objective			Second objective			
			Total cost (USD)	Peak load	Fleet size	The number of nodes visited	Total cost (USD)	Peak load	Fleet size
400	2	2	21483.23	7	104	4435	21482.43	4	104
		4	23068.37	4	116	4671	23066.09	3	116
		6	21151.25	4	109	663	21140.44	1	109
		8	20757.35	3	108	2945	20748.16	1	104
	5	2	22343.92	5	117	0	22338.94	0	118
		4	23152.58	3	112	20	23152.58	3	112
		6	21933.04	3	112	15	21933.04	3	112
		8	22416.49	5	117	8	22416.49	5	117
	8	2	22650.28	4	127	2	22650.28	4	127
		4	19730.04	3	101	0	19730.04	3	101
		6	23376.07	4	125	0	23376.07	4	125
		8	20882.61	2	111	0	20882.61	2	111

As we can see that, after the reschedule by the second objective, the peak load of charging demand has been considerably reduced without a significant increase in fleet size. Commendably, the total cost has also been reduced slightly, which indicates a win-win situation. The reduction of peak load is quite evident when the number of depots is relatively small (i.e., 2 and 5). The possible reason is that the number of variables will be reduced with a smaller number of depots, such that a better solution can be found with less computational time. This suggests that the effect of the second objective is closely related to the instance size and computation capacity. If the instance size is small, the optimization of the first objective is sufficient, and the result from the second objective could be unchanged since there is no room for optimization. If the instance size is medium, the optimization of the first objective would be insufficient, such that the second objective works since there is still room for optimization. If the instance size is too large, the optimization of the first objective would be insufficient, such that the optimization of the second objective cannot be processed and the result from the second objective is also unchanged. For the results in Table 3, the size of the first five instances is medium, and the gaps for the first objective solutions are 9-11%, such that there is still room for optimization for the second objective. However, for the last seven instances, the number of variables exceeds 700,000. When the instance size is larger, the computation time at each node will become longer, such that a smaller number of nodes can be visited within a limited time, and it is more difficult to find a better solution. As shown in Table 3, the number of nodes visited for the first four instances exceeds 500, while the number of nodes visited for the last seven instances is less than 20. Typically, for the instance with $|K| = 5$ and $l = 2$, the number of nodes visited is 0 and the peak load decreases to 0, which is because the heuristic of root nodes in CPLEX has found the optimal solution. The

number of nodes visited is 0 for the last three instances, which is because CPLEX has not completed root node processing.

We note also that for the instance with $|K| = 5$ and $l = 2$, the fleet size has been slightly increased (from 117 to 118). The reason is that there exist cost trade-offs between the charging cost and additional transportation cost. The number of charging stations for this instance is the smallest (only 2) among all the instances. Since a recharging activity requires an additional deadheading trip, the variable transportation cost for recharging activities may be relatively large compared to other instances. In this instance, the peak load is reduced to 0 at the expense of a greater fleet size, which implies that the charging cost is not sufficient to overweigh the additional transportation cost.

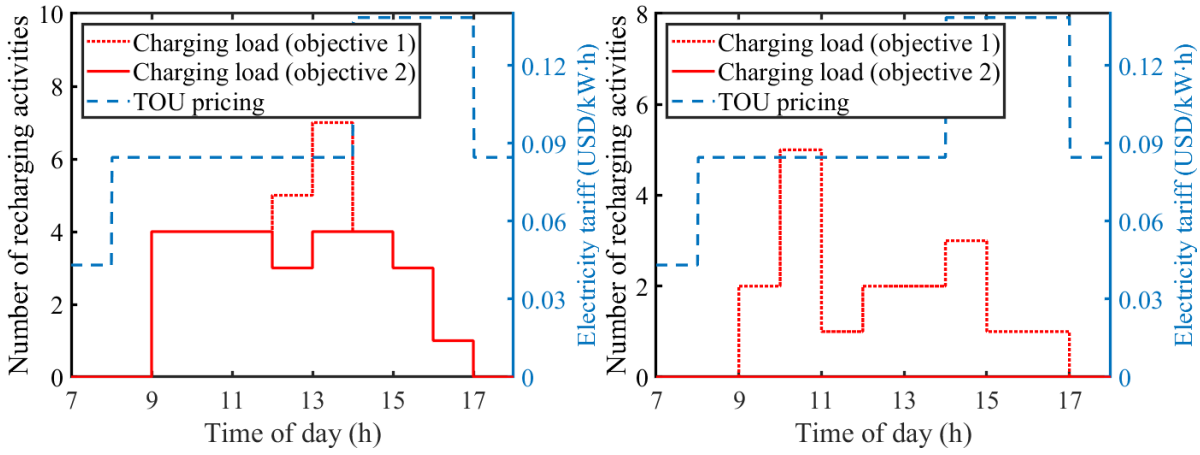


Fig. 6 Temporal distribution of recharging activities over time of day under (a) 400 trips, 2 depots, and 2 charging stations; (b) 400 trips, 5 depots, and 2 charging stations

Fig. 6 depicts the temporal distribution of recharging activities over time of day for two typical instances corresponding to Table 3. For the first instance with 400 trips, 2 depots and 2 charging stations (Fig. 6(a)), after rescheduling by the second objective, the peak load during 12:00-14:00 has diminished. For the second instance with 400 trips, 5 depots and 2 charging stations (Fig. 6(b)), the overall charging demand has diminished.

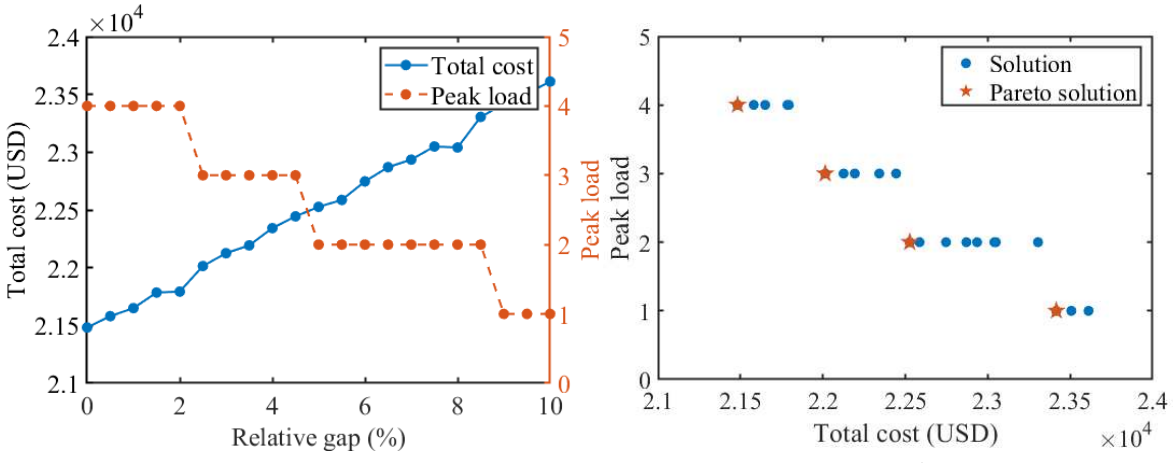


Fig. 7 Analysis of relative gap for a typical instance: (a) Total cost and peak load; (b) Pareto solutions

As the lexicographic method is another form of the epsilon-constrained method by adding the relative

gap, the impact of the relative gap on the solutions deserves some discussion. The instance with 400 trips, 2 depots, and 2 charging stations is selected for analysis, and the results are shown in Fig. 7. As shown in Fig. 7(a), as the relative gap increases, the constraints on the cost will be relaxed, resulting in the higher total cost and lower peak load. The combinations of total cost and peak load for each relative gap yield the solutions in Fig. 7(b). As we can see, there are a few weak-dominated solutions with identical peak load but different costs. This is because adding a relative gap to the constraint cannot guarantee to exclude the inferior solutions with the same peak load but higher cost. Another possible reason is that the second objective cannot guarantee to be solved to global optimality. Nevertheless, the Pareto solutions can be obtained by collecting the non-dominated solutions, as shown by the solid line in Fig. 7(b).

5.2 Case studies

5.2.1 Case description

In this section, we report on the applicability of our model and algorithm via a real-world bus network, Guangzhou Higher Education Mega Center, and compare the performance of MDEVSP with that of the state-of-the-practice. Guangzhou Higher Education Mega Center is located in Xiaoguwei Street, Panyu District, Guangzhou City, with an area of 34.4 square kilometers on both sides of the Zhujiang River. There are more than 20 bus lines in Guangzhou Higher Education Mega Center. We select four island lines to test our model (Loop 1, Loop 2, Pan No.201, and Pan No.202). The four lines have a total of five depots: the Guangzhou National Archives Hall, the Outer Ring W Rd Station, the Guangzhou Daxuecheng Stadium, the Suishicun Station, the Guangdong Science Center Station. An electric bus charging station is located at the Guangzhou National Archives Hall. The map of bus lines is shown in Fig. 8.

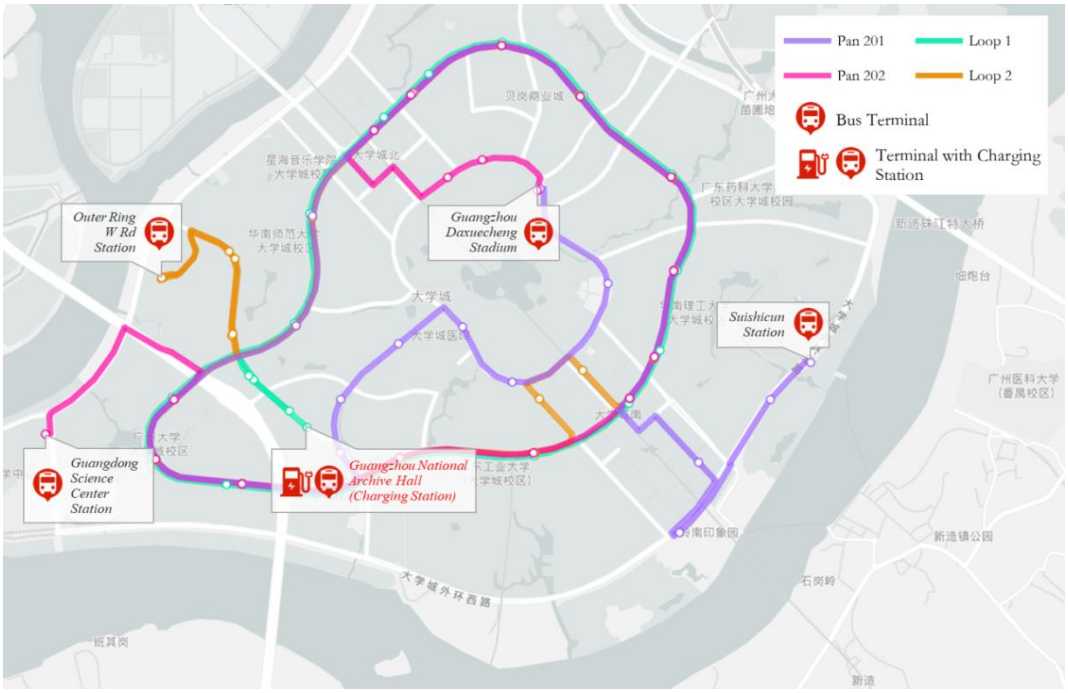


Fig. 8 The map of bus lines

In the MDEVSP, the information of deadheading time and trip information is required. The lengths of the bus lines are measured using the Baidu map, and the one-way trip time is calculated assuming an

average bus operating speed of 20 km/h. Table A.1 lists the resulted deadheading distance and travel time between stations. The overall service time is 6:30-22:30, and there are a total of 379 trips a day. The peak periods are between 7:00 to 9:00 in the morning and 16:00 to 18:00 in the afternoon, and the low peak period is after 21:00 in the evening. The information of each bus line is given in Table A.2. With the headway and trip time information provided in Table A.2, the corresponding bus trip information (i.e., trip duration) is provided in Table A.3. We assume that the depot capacity and charging station capacity are 30 vehicles per depot, and the fleet size is 100 vehicles.

In this area, the vehicle type used is the BYD K& having a design running range of 260 km and a vehicle capacity of 44 pax. However, in practice, the design running range cannot be fully utilized due to possible road gradients, traffic congestion, and air-conditioning. Moreover, to prolong the battery life, the running range would be deliberately reduced by the public transport operator to avoid deep discharge. In view of this, we set the running range of the vehicles as 130 km. The fast charging time is set as 30 minutes, which can be achieved for the fast-charging type charging device, such as BYD EVA080KG. The working voltage is AC342V-440V (three-phase), and the input/output power is less than 200 kW. In addition, the fixed transportation cost is taken as 285.714 USD/vehicle (2000 RMB/vehicle). The variable transportation cost is taken as 0.429 USD/km (3RMB/km). The waiting time cost is taken as 0.014 USD/min (0.1 RMB/min), and the fixed charging cost is taken as 7.143 USD per time (50 RMB per time).

In this section, we consider two simpler approaches with and without TOU pricing considerations as the benchmark against MDEVSP. When the TOU pricing is not considered, in practice the operator would not choose to recharge unless necessary, and buses are usually recharged only when the remaining running range is insufficient to complete the next trip task. In this circumstance, the vehicle trip chains can be constructed using the first-in-first-out (FIFO) principle, which is widely implemented in state-of-the-practice. More specifically, a bus first arriving at a depot will first dispatch to the other ending point of the route based on the timetable. However, such a FIFO rule only applies to a single bus route and cannot be applied to the multi-route scenario, since in the MDVSP it is difficult to evaluate the remaining running range in the provision of different route lengths and deadheading trip mileage. Therefore, the single-line vehicle schedule, where each bus line operates independently using the FIFO rule, is adopted as a benchmark. Each bus route is assigned to its nearest charging station. The overall performance is achieved by aggregating those of each line. Note that the FIFO rule may be adapted by the deficit function approach with deadheading trips, under which circumstance the fleet size may be reduced at the expense of additional deadheading trip costs (Ceder, 2007). The deficit function is however beyond the scope of this study as it is also difficult to evaluate the remaining running range in the provision of deadheading trip mileage.

To consider the recharging event, the traditional FIFO principle proposed by Ceder (2007) is strengthened by a safety running range that is sufficient to return to the depot or charging station. That is, for any trip node $i \in S$, the following conditions hold simultaneously:

$$g_i + \min_{e \in E} DH(i,e) \leq DMAX \tag{45}$$

$$g_i + DH(i,\theta') \leq DMAX \tag{46}$$

where condition (45) denotes that the remaining running range is sufficient to return to the nearest charging station, and condition (46) denotes that the remaining running range is sufficient to return to the depot. The pseudocode of the trip-connection method of the extended FIFO rule is described in Algorithm 2.

Algorithm 2 Pseudocode of the extended FIFO rule

1	For each trip node i, find trip node j according to the FIFO rule.
2	Connect trip node i to trip node j only if the conditions for safety running range (Conditions (45) and (46)) are met; otherwise, go to Step 3.
3	Select the charging station with a distance of $\min_{e \in E} DH(i,e)$, and the nearest charging node to the current time.
4	If trip tasks still exist after the recharging activity, then continue to find the next trip node based on the FIFO rule; otherwise, go to Step 5.
5	Return to the depot after serving trip node j.

Given that the aforementioned extended FIFO rule is not designed to respond to TOU pricing, we also consider another benchmark with awareness of TOU pricing by the enhancement of the extended FIFO rule (FIFO-TOU). The FIFO-TOU rule identifies the safety running range in the same way as the extended FIFO rule (Conditions (45) and (46)), but allows the operator to prioritize recharging buses during the low-tariff period. Specifically, when a bus needs to recharge, the low-tariff period is prioritized; if the trip chain fails to connect after the recharging activity during the low-tariff period, the bus will be recharged during the high-tariff period. By modifying Algorithm 2, the pseudocode of the trip-connection method of the FIFO-TOU rule is described in Algorithm 3.

Algorithm 3 Pseudocode of the extended FIFO-TOU rule

1	For each trip node i, find trip node j according to the FIFO rule.
2	Connect trip node i to trip node j only if the conditions for safety running range (Conditions (45) and (46)) are met; otherwise, go to Step 3.
3	Select the charging station with a distance of $\min_{e \in E} DH(i,e)$, and the nearest charging node during the low-tariff period to the current time.
4	If trip tasks still exist after the recharging activity, then continue to find the next trip node based on the FIFO rule; otherwise, go to Step 5.
5	Select the charging station with a distance of $\min_{e \in E} DH(i,e)$, and the nearest charging node during the high-tariff period to the current time.
6	If trip tasks still exist after the recharging activity, then continue to find the next trip node based on the FIFO rule; otherwise, go to Step 7.
7	Return to the depot after serving trip node j.

5.2.2 Results and discussion

As a result, under the extended FIFO rule, the required fleet size is 39 vehicles and the total cost is 13884.2 USD. Under the FIFO-TOU rule, the required fleet size is 43 vehicles and the total cost is 14974.7

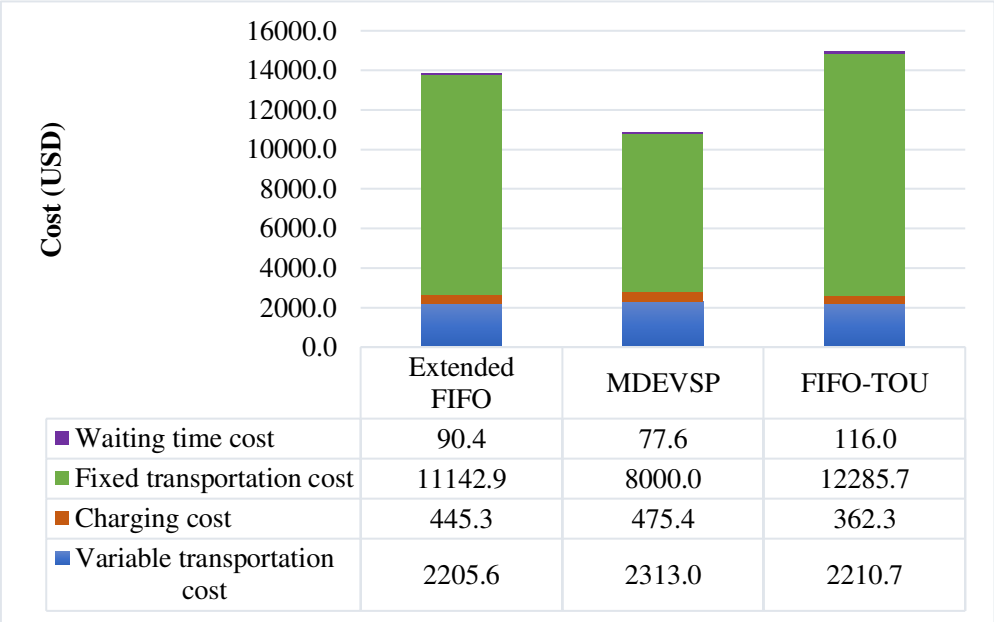
USD. Under the MDEVSP, the branch-and-price algorithm has visited a total of 88 nodes with a gap value of 6.14%, and the resulting fleet size is 28 vehicles with a total cost of 10866 USD. For this example, the branch-and-price method far outperforms the benchmarks in both total cost and fleet size.

Table 4 Optimal vehicle scheduling scheme by the branch-and-price algorithm

Vehicle No.	Trip chain	Travel time (min)	Waiting time (min)	Trip distance (km)	Deadh-eading (km)	Charging cost (USD)
1	A-2-DH-20-43-71-DH-89-104-Charge-129-DH-148-165-177-216-244-268-287-316-Charge-344-359-374-377-A	745	151	249.4	13.4	36.59
2	A-3-DH-26-49-76-97-DH-121-DH-145-174-Charge-213-227-248-271-292-319-334-DH-350-369-A	736	134	243.6	10.8	20.99
3	A-18-39-56-DH-86-118-DH-144-Charge-181-196-DH-228-256-279-312-329-352-A	608	192	201.8	11.5	15.60
4	A-19-47-74-105-136-DH-158-173-DH-191-Charge-230-255-276-293-DH-327-345-A	628	152	207.6	10.8	20.99
5	A-188-Charge-263-DH-306-A	150	143	50.5	10.2	20.99
6	B-5-32-55-82-99-122-143-161-Charge-199-220-247-270-295-320-337-358-B	764	64	253.2	3.6	20.81
7	B-10-31-48-75-DH-111-125-DH-153-176-Charge-222-249-277-304-322-343-360-B	677	159	224.8	12.5	20.99
8	B-11-34-DH-64-137-152-Charge-189-212-231-262-285-B	449	199	148.8	3.6	15.60
9	B-17-40-65-90-113-130-151-Charge-197-214-241-264-291-314-335-366-B	717	111	237.6	3.6	15.60
10	B-24-41-62-DH-84-115-138-159-Charge-221-246-269-294-DH-324-DH-354-B	572	224	190.4	17.0	20.99
11	C-1-16-DH-37-60-87-103-116-140-169-Charge-206-DH-237-260-296-331-DH-351-365-375-378-C	727	241	244.9	21.7	20.99
12	C-4-25-DH-52-DH-81-91-106-Charge-149-162-193-225-253-272-299-338-DH-364-C	613	224	207.3	19.3	15.60
13	C-6-28-50-78-100-124-146-Charge-183-198-219-DH-245-267-290-315-339-361-C	659	181	218.6	11.0	15.60
14	C-8-38-70-88-102-119-154-178-200-235-Charge-298-325-DH-342-357-367-376-C	638	262	213.1	14.7	15.78
15	C-12-36-59-80-DH-101-DH-142-Charge-194-211-223-251-274-302-317-333-346-371-C	651	186	218.6	15.8	15.60
16	C-13-35-68-95-DH-112-127-Charge-156-172-202-238-265-288-DH-313-DH-330-362-C	614	211	206.0	16.5	15.60
17	C-14-45-67-94-123-147-171-Charge-217-239-257-DH-286-311-321-336-DH-355-373-C	638	196	213.0	14.6	20.99
18	C-22-DH-46-63-83-98-114-Charge-164-180-192-234-280-308-DH-328-DH-348-DH-368-C	606	219	204.3	20.0	15.60
19	C-27-51-77-DH-107-128-Charge-167-DH-187-203-232-258-DH-284-DH-307-C	538	117	181.7	21.6	15.60
20	C-30-57-132-Charge-170-DH-204-240-261-00278-305-C	393	263	134.0	20.8	15.60
21	C-54-DH-120-166-184-205-DH-233-273-297-C	379	242	127.0	12.1	0.00
22	C-61-182-207-DH-254-DH-303-C	240	358	80.3	8.8	0.00
23	D-21-44-73-85-109-133-DH-150-Charge-175-190-DH-208-DH-229-DH-259-283-309-341-353-D	638	144	213.1	14.7	15.60
24	D-58-DH-134-DH-168-DH-201-DH-224-250-281-D	346	228	118.2	22.4	0.00
25	E-7-DH-33-DH-72-DH-92-DH-135-Charge-160-DH-186-DH-218-242-266-289-E	502	160	171.1	28.9	15.60
26	E-9-29-53-79-96-110-126-141-157-Charge-195-210-226-252-275-300-318-332-349-363-Charge-379-E	757	160	252.7	12.9	31.38

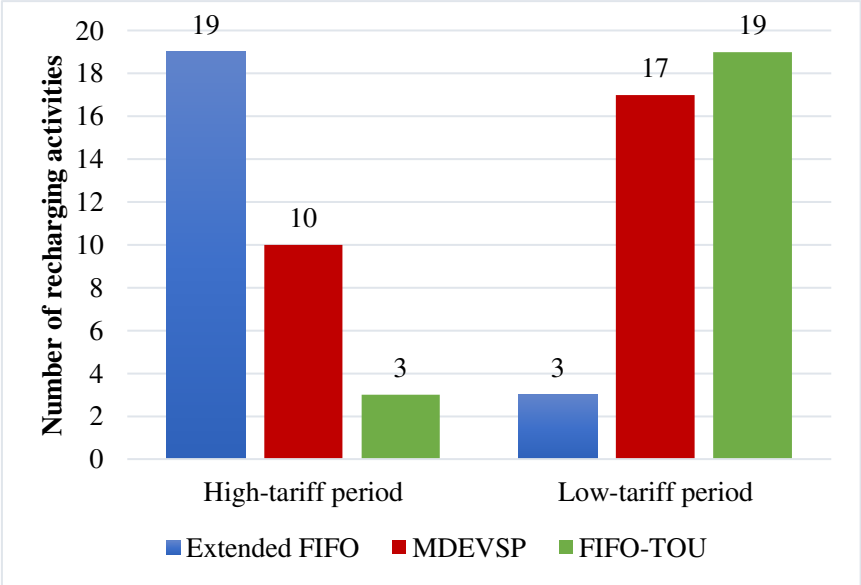
27	E-15-42-69-108-DH-131-155-179-Charge-215-243-282-310-323-347-370-E	591	258	198.5	15.7	20.99
28	E-23-66-DH-93-117-139-163-185-209-236-Charge-301-326-340-356-372-E	562	253	187.0	9.6	15.78

Table 4 presents the optimal vehicle scheduling scheme by the branch-and-price method. The first column represents the vehicle numbers. The trip chains for the electric buses are given in the second column, where the letters A-E represent the bus terminus Guangzhou National Archives Hall, Outer Ring W Rd Station, Guangzhou Daxuecheng Stadium, Suishicun Station, and Guangdong Science Center Station, respectively; and 'DH' and 'Charge' denote the deadheading trip and recharging activity, respectively; and the trip numbers are in accordance with Table A.3. Given these trip chains, the total travel time, waiting time, overall trip distance, deadheading trip distance, and charging cost can be calculated readily.



	Extended FIFO	MDEVSP	FIFO-TOU
Waiting time cost	90.4	77.6	116.0
Fixed transportation cost	11142.9	8000.0	12285.7
Charging cost	445.3	475.4	362.3
Variable transportation cost	2205.6	2313.0	2210.7

(a) System costs



(b) Recharging activity arrangement

Fig. 9 Performance of extended FIFO, FIFO-TOU and MDEVSP

Fig. 9 shows the system costs and the distribution of recharging activities for the extended FIFO rule, FIFO-TOU rule, and the MDEVSP calculated by the branch-and-price algorithm. As shown in Fig. 9(a), the waiting time cost has been reduced considerably by our model. This is due to the coordinated fleet management and resulting less idle time. According to the fixed transportation cost, the fleet size has been saved by using our model at the expense of a larger number of recharging activities and deadheading trips. Compared to the extended FIFO rule, the number of recharging activities increases in the MDEVSP calculated by the branch-and-price algorithm (27 vs. 22 activities), whereas a larger number of recharging activities are distributed in the low-tariff period. Therefore, the charging cost increases slightly for the MDEVSP. This suggests that the TOU pricing scheme tends to shift recharging activities from the high-tariff period to the low-tariff period, which reduces the peak load for power grids. Although the number of recharging activities during the high-tariff period and the charging cost are the lowest under the FIFO-TOU rule, the fixed transportation cost (and thus fleet size), waiting time cost, and total cost are the highest. This is because, in the provision of TOU awareness, a number of buses would be forced to wait at the charging stations or depots to avoid recharging activities during the high-tariff period (see Fig. 10), such that a number of trip tasks during the high-tariff period are skipped to avoid high-tariff recharging activities, and vehicles should be added to execute these skipped trip tasks.

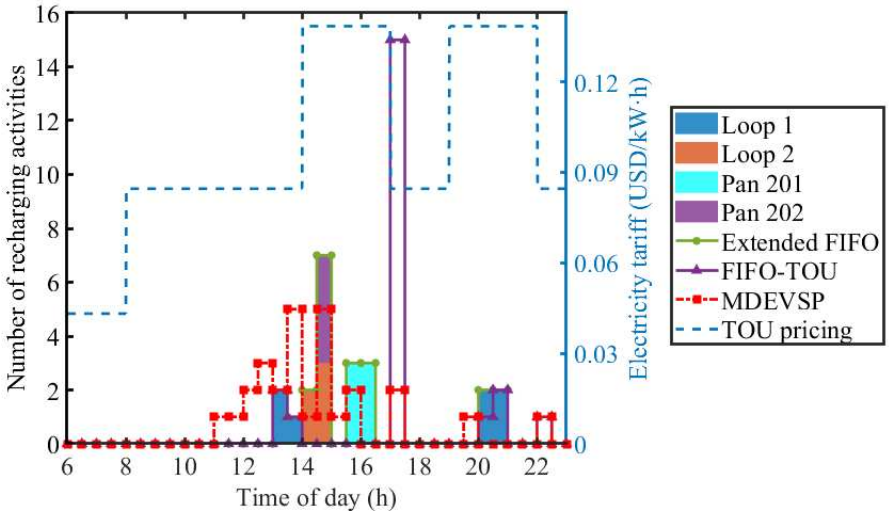


Fig. 10 Recharging periods and electricity tariff distributions

As the charging schedule is integrated into the vehicle scheduling problem, it is interesting to examine how the recharging activities are distributed over the time of day and the associated charging loads. Fig. 10 shows the electricity tariff distribution (TOU pricing), recharging periods, and the corresponding charging load. As we can see, under the extended FIFO rule, most of the recharging activities are performed during the high-tariff period. The recharging periods differ among different lines. For the bus number Loop 1, the recharging period ranges from 13:00 to 14:00 and from 20:00 to 21:00. The recharging periods of Pan 202 and Loop 2 primarily range from 14:00 to 15:00, while that of Pan 201 distributes around 16:00. Under the FIFO-TOU rule, the recharging period ranges from 13:00 to 14:00, from 17:00 to 17:30, and from 20:00 to

21:00. Only 3 buses are recharged at the low-tariff period (13:00-14:00) and the high-tariff period (20:00-21:00), respectively. Most of the recharging activities (totally 15) are performed immediately after the high-tariff period (17:00-17:30), which indicates that these buses are waiting at the charging stations or depots to avoid recharging activities during the high-tariff period. The charging load of MDEVSP is calculated by the branch-and-price algorithm. One can see that the peak load under the FIFO-TOU rule is the highest, being approximately two and three times that of extended FIFO and MDEVSP, respectively. This suggests that the FIFO-TOU could induce centralized charging demand that is not desirable in practice. Compared to the two benchmark scenarios, the charging demand under the MDEVSP is spread around a wider range of low-tariff periods, along with a smooth hump. Although the charging cost has increased respectively by 6.8% and 31.2 % compared to that of the extended FIFO rule and FIFO-TOU rule (see Fig. 9), the peak load has been dramatically reduced by 40% and 200%, respectively, which reveals a promising application result.

6. Concluding remarks

Vehicle scheduling is an important part of public transport planning. With electric buses being introduced by more and more cities around the world, there is an increased need for developing a modeling and solution framework for the electric vehicle scheduling problem that takes into account the electric bus charging demand and the power grid characteristics. This paper introduces a multi-objective multi-depot electric vehicle scheduling problem explicitly considering the TOU pricing and peak load risk. The objectives are to minimize the total operation cost and the peak charging load. We mathematically model this problem on a time-expanded network and reformulate the multi-objective optimization model by a practical lexicographic method. In particular, we propose an exact tailored branch-and-price method to solve the problem. Particularly, the heuristics and a trip chain pool strategy are embedded into the branch-and-price method to expedite the computation time.

The proposed model and solution approach are tested by a benchmark and a real-world bus network in Guangzhou, China. Results demonstrate the superiority of our method over the off-the-shelf solver with respect to solution quality and computation time. More importantly, our model can achieve operation cost savings and peak load leveling. In future research, more exogenous factors can be included in our modeling framework. For example, while this paper concentrates on MDEVSP with a single vehicle type, future research may concern the schedule of mixed fleets comprised of both conventional diesel and electric buses. Another line of future research is to investigate further the scalability of the exact method.

Appendix A

Table A.1 Deadheading distances/time (km/min) between stations

	Guangzhou National Archives Hall	Outer Ring W Rd Station	Guangzhou Daxuecheng Stadium	Suishicun Station	Guangdong Science Center	Charging Station

	Station					
Guangzhou National Archives Hall	0	1.8/6	3.4/8	4.7/14	3.2/9	0
Outer Ring W Rd Station	1.8/6	0	3.8/10	6.5/16	3.9/11	1.8/6
Guangzhou Daxuecheng Stadium	3.4/8	3.8/10	0	3.1/8	5.9/14	3.4/8
Suishicun Station	4.7/14	6.5/16	3.1/8	0	6.1/12	4.7/14
Guangdong Science Center Station	3.2/9	3.9/11	5.9/14	6.1/12	0	3.1/9
Charging Station	0	1.8/6	3.4/8	4.7/14	3.2/9	0

Table A.2 Bus line information

Line	Direction	Origin depot	Destination depot	Line length(km)	Trip time(min)	Time of day	Headway(min)
Loop 1	-	Guangzhou National Archives Hall	Guangzhou National Archives Hall	12	36	6:30-22:30	8 (High peak) 12 (Flat peak) 15 (Low peak)
Loop 2	-	Outer Ring W Rd Station	Outer Ring W Rd Station	15.6	47	7:00-21:00	10
Pan 201	Up	Guangzhou Daxuecheng Stadium	Suishicun Station	13.5	41	7:00-21:00	15 (High peak) 20 (Flat peak)
	Down	Suishicun Station	Guangzhou Daxuecheng Stadium	13.5	41	7:00-21:00	10 (High peak) 15 (Flat peak)
Pan 202	Up	Guangzhou Daxuecheng Stadium	Guangdong Science Center Station	11.2	34	7:00-21:00	15 (High peak) 20 (Flat peak)
	Down	Guangdong Science Center Station	Guangzhou Daxuecheng Stadium	12.7	38	7:00-21:00	15 (High peak) 20 (Flat peak)

Table A.3 Bus trip information

Trip No.	Bus line	Trip duration	Direction	Trip No.	Bus line	Trip duration	Direction
1	Loop 1	6:30-7:06	-	203	Pan 202	14:40-15:24	Down
2	Loop 1	6:42-7:18	-	204	Loop 1	…	-
3	Loop 1	6:54-7:30	-	…	…	21:45-22:21	…
4	Loop 1	7:00-7:36	-	376	Loop 1	22:00-22:36	-
…	…	…	…	377	Loop 1	22:15-22:51	-
201	Pan 201	14:40-15:21	Down	378	Loop 1	22:30-23:06	-
202	Pan 202	14:40-15:14	Up	379	Loop 1	14:40-15:24	-

Acknowledgements

This work is jointly supported by the Science and Technology Program of Guangzhou, China (Project No. 201904010202), National Science Foundation of China (Project No. 72071079, 61703165, 71890972, 71890970), and the Science and Technology Program of Guangdong Province (Project No. 2020A1414010010), and the UK-PACK project EUM151.

References

Argote-Cabanero J., Daganzo, C.F., Lynn, J.W., 2015. Dynamic control of complex transit systems. Transportation Research Part B, 81, 146-160.

Adler J.D., Mirchandani P.B., 2017. The vehicle scheduling problem for fleets with alternative-fuel vehicles. Transportation Science, 51(2), 441-456.

Ayre, J., 2018. New BYD ADL Electric Bus Fleet Deployed In London-Route 153 Now Fully Electric.https://cleantechnica.com/2018/02/13/new-byd-adl-electricbus-fleet-deployed-london-route-153-now-electric/. (Accessed March 15, 2019).

An K., 2020. Battery electric bus infrastructure planning under demand uncertainty. Transportation Research Part C, 111, 572-587.

Barnhart C., Johnson E. L., Nemhauser G. L., Savelsbergh M. W. P., Vance P. H., 1998. Branch-and-price: Column generation for solving huge integer programs. Operations Research, 46(3), 316-329.

Bi Z., Keoleian G., Ersal T., 2018. Wireless charger deployment for an electric bus network: A multi-objective life cycle optimization. Applied Energy, 225, 1090-1101.

Boyer V., Ibarra-Rojas O.J., Ríos-Solís Y., 2018. Vehicle and crew scheduling for flexible bus transportation systems. Transportation Research Part B, 112, 216-229.

Ceder A., 2007. Public transit planning and operation: theory, modelling and practice, Elsevier.

Carosi S., Frangioni A., Galli L., Girardi L., Vallese G., 2019. A matheuristic for integrated timetabing and vehicle scheduling. Transportation Research Part B, 127, 99-124.

Deb K., Multi-objective optimization using evolutionary algorithms: John Wiley & Sons, 2001.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation, 6(2), 182-197.

Desfontaines L., Desulniers G., 2018. Multiple depot vehicle scheduling with controlled trip shifting. Transportation Research Part B, 113, 34-53.

Ehrgott, M., 2005. Multicriteria Optimization, second ed. Springer, Berlin, pp. 323.

Hadjar A., Marcotte O., Soumis F., 2006. A branch-and-cut algorithm for the multiple depot vehicle

scheduling problem. Operations Research, 54(1), 130-149.
He F., Yang J., Li M., 2018. Vehicle scheduling under stochastic trip times: An approximate dynamic programming approach. Transportation Research Part C, 96, 144-159.
He Y., Song Z., Liu Z., 2019. Fast-charging station deployment for battery electric bus systems considering electricity demand charges. Sustainable Cities and Society, 48,101530.
He Y., Liu Z., Song Z., 2020. Optimal charging scheduling and management for a fast-charging battery electric bus system. Transportation Research Part E, 142(23), 102056.
Huisman D., Wagelmans A.P.M., 2006. A solution approach for dynamic vehicle and crew scheduling. European Journal of Operational Research, 172, 453-471.
Janovec M., Kohání M. Exact approach to the electric bus fleet scheduling. Transportation Research Procedia, 2019, 40, 1380-1387.
Kliewer N., Mellouli T., Suhl L., 2006. A time-space network based exact optimization model for multi-depot bus scheduling. European Journal of Operational Research, 175, 1616-1627.
Kooten N.M.E., Akker J. M., Hoogeveen J.A. Scheduling electric vehicles. Public Transport, 2017, 9:155-176.
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. Transportation Research Part B, 118, 457-487.
Laskaris G., Seredynski M., Viti F., 2020. Enhancing bus holding control using cooperative ITS. IEEE Transactions on Intelligent Transportation Systems, 21(4), 1767-1778.
Li J., 2014. Transit bus scheduling with limited energy. Transportation Science, 48(4), 521-539.
Li L., Hong K.L., Xiao F., 2019. Mixed bus fleet scheduling under range and refueling constraints. Transportation Research Part C, 104, 443-462.
Liu T., Ceder A., 2020. Battery-electric transit vehicle scheduling with optimal number of stationary chargers. Transportation Research Part C, 114, 118-139.
Lin Y., Zhang K., Shen Z., Ye B., Miao L., 2019. Multistage large-scale charging station planning for electric buses considering transportation network and power grid. Transportation Research Part C, 107, 423-443.
Liu Z.C., Song Z.Q., 2017. Robust planning of dynamic wireless charging infrastructure for battery electric buses. Transportation Research Part C, 83, 77-103.
Mavrotas G., 2009. Effective implementation of the e-constraint method in Multi-Objective Mathematical Programming problems. Applied Mathematics and Computation, 213, 455-465.
Muller, T.H.J., Furth, P.G., 2000. Integrating Bus Service Planning with Analysis, Operational Control, and Performance Monitoring. In: Proceeding of Intelligent Transportation Society of America Annual Meeting. Boston, Mass.
Petersen H., Larsen A., Madsen O., Petersen B., Ropke S., 2013. The simultaneous vehicle scheduling and passenger service problem. Transportation Science, 47(4), 603-616.
Perumal S.S.G., Dollevoet T., Huisman D., Lusby R.M., Larsen J., Riis M. Solution Approaches for Vehicle and Crew Scheduling with Electric Buses. Econometric institute research papers, Erasmus University Rotterdam, Erasmus School of Economics (ESE), Econometric Institute, January, 2020.
Rogge M., Hurk E., Larsen A., Sauer D., 2018. Electric bus fleet size and mix problem with optimization of charging infrastructure. Applied Energy, 211, 282-295.
Shen Y., Xu J., Li J., 2016. A probabilistic model for vehicle scheduling based on stochastic trip times. Transportation Research Part B, 85, 19-31.

Shen Z., Feng B., Mao C., Ran L., 2019. Optimization models for electric vehicle service operations: A literature review. Transportation Research Part B, 128, 462-477.

Tang X., Lin X., He F., 2019. Robust scheduling strategies of electric buses under stochastic traffic conditions. Transportation Research Part C, 105, 163-182.

Uçar E., Birbil S˙I., Muter ˙I., 2017. Managing disruptions in the multi-depot vehicle scheduling problem. Transportation Research Part B, 105, 249-269.

Wang Y., Huang Y., Xu J., Barclay N. Optimal recharging scheduling for urban electric buses: A case study in Davis. Transportation Research Part E, 2017, 100, 115-132.

Wen M., Linde E., Ropke S., Mirchandani P., Larsen A., 2016. An adaptive large neighborhood search heuristic for the Electric Vehicle Scheduling Problem. Computers & Operations Research, 76:73-83.

Wei R., Liu X., Ou Y., Fayyaz S.K., 2018. Optimizing the spatio-temporal deployment of battery electric bus system. Journal of Transport Geography, 68, 160-168.

Wu W., Liu R., Jin W., 2016. Designing robust schedule coordination scheme for transit networks with safety control margins. Transportation Research Part B, 93, 495-519.

Wu W., Liu R., Jin W., 2017. Modelling bus bunching and holding control with vehicle overtaking and distributed passenger boarding behaviour. Transportation Research Part B, 104, 175-197.

Wu W., Liu R., Jin W., Ma C., 2019. Simulation-based robust optimization of limited-stop bus service with vehicle overtaking and dynamics: A response surface methodology. Transportation Research Part E, 130, 61-81.

Wu W., Lin Y., Liu R., Li Y., Zhang Y., Ma C., 2020. Online EV charge scheduling based on time-of-use pricing and peak load minimization: Properties and efficient algorithms. IEEE Transactions on Intelligent Transportation Systems, doi: 10.1109/TITS.2020.3014088.

Yang C., Lou W., Yao J., Xie S., 2018. On charging scheduling optimization for a wirelessly charged electric bus system. IEEE Transactions on Intelligent Transportation Systems, 19(6), 1814-1826.

Yao E., Liu T., Lu T., Yang Y., 2020. Optimization of electric vehicle scheduling with multiple vehicle types in public transport. Sustainable Cities and Society, 52, 101862.

Zeighami V., Soumis F., 2019. Combining Benders' decomposition and column generation for integrated crew pairing and personalized crew assignment problems. Transportation Science, 53(5), 1479-1499.