Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Transit Bus Scheduling with Limited Energy

Jing-Quan Li

To cite this article:
Jing-Quan Li (2014) Transit Bus Scheduling with Limited Energy. Transportation Science 48(4):521-539. https://doi.org/10.1287/trsc.2013.0468

TRANSPORTATION SCIENCE
Vol. 48, No. 4, November 2014, pp. 521–539
ISSN 0041-1655 (print) | ISSN 1526-5447 (online)

informs®
http://dx.doi.org/10.1287/trsc.2013.0468

Transit Bus Scheduling with Limited Energy

Jing-Quan Li
California PATH, University of California, Berkeley, Richmond, California 94804, jingquan@path.berkeley.edu

In this paper, we propose a vehicle-scheduling model for electric transit buses with either battery swapping or fast charging at a battery station, and a vehicle-scheduling model with the maximum route distance constraint for compressed natural gas, diesel, or hybrid-diesel buses. Both of these scheduling models are NP-hard. We develop column-generation-based algorithms to solve the scheduling problems. We conduct extensive case studies based on real-world instances and instances randomly generated in a practical setting. Our computational experiments show that our algorithms demonstrate very good computational performances. We also use real-world transit data to systematically analyze the number of buses needed, the total operational costs, and the vehicle emissions generated when compressed natural gas, diesel, hybrid, or electric buses are used in service.

Keywords: emission and fuel economy estimation; battery swapping; fast charging; column generation; vehicle scheduling
History: Received: January 2012; revisions received: November 2012, February 2013; accepted: February 2013. Published online in *Articles in Advance* June 25, 2013.

1. Introduction

The vehicle-scheduling problem consists of assigning transit buses to serve a series of timetabled trips with the objective of minimizing fleet size and/or operational costs. Heavy diesel buses were primarily used by transit agencies before the 2000s. To reduce air pollution from heavy diesel buses, manufacturers have developed buses with alternative power sources, including compressed natural gas (CNG) buses, biodiesel buses, and hybrid-diesel buses. In 2011, the percentage of diesel buses was reduced to 69% in the United States (National Transit Database 2013). Recently, battery-electric buses have been used on a large scale in several countries. Battery-electric buses produce zero emissions. However, a fully charged battery often allows an electric bus to run a relatively short distance. The current battery needs to be either charged or swapped with a fully charged one at a battery service station before it depletes. CNG, diesel, and hybrid buses also have the maximum route distance constraint because of their limited tank capacity. Therefore, it is desirable to incorporate the maximum-route distance constraint into the vehicle-scheduling model and consider battery swapping or fast charging if electric buses are used in service. It is also expected to evaluate bus emissions after the scheduling model is solved. These issues are the major motivation for this paper.

The vehicle-scheduling problem can be categorized into two groups: the single-depot vehicle-scheduling problem (SDVSP) and the multiple-depot vehicle-scheduling problem (MDVSP). The SDVSP is polynomially solvable, and a large number of approaches have been proposed, e.g., Paixão and Branco (1987); Song and Zhou (1990); Freling, Wagelmans, and Paixão (2001b). The MDVSP is NP-hard (Bertossi, Carraresi, and Gallo 1987); solution approaches include both exact algorithms (e.g., Carpaneto et al. 1989; Forbes, Holt, and Watts 1994; Ribeiro and Soumis 1994; Löbel 1998; Hadjar, Marcotte, and Soumis 2006; Kliewer, Mellouli, and Suhl 2006) and heuristics (e.g., Ball, Bodin, and Dial 1983; Bodin et al. 1983; Bianco, Mingozzi, and Ricciardelli 1994; Pepin et al. 2009). We refer interested readers to Desaulniers and Hickman (2007) and Bunte and Kliewer (2009) for recent reviews.

Air pollution caused by heavy diesel vehicles has already been recognized. Elkins et al. (2003) report that heavy diesel trucks and buses contribute more than 45% of nitrogen oxides and 75% of particulate matters. A few studies have considered bus emissions in the vehicle scheduling. Li and Head (2009) develop a scheduling model to investigate the reduction of bus emissions for a mixed fleet of transit buses with different power sources. Stasko and Gao (2010) incorporate potential engine improvement options into an optimization model to reduce bus emissions, but published schedules were not included in the model. Neither Li and Head (2009) nor Stasko and Gao (2010) consider electric buses in their studies.

Battery-electric buses have a unique advantage: zero emissions. They have actually been used as early as the 1990s, aiming at reducing emissions in populous downtown areas (e.g., see the case of Tennessee, Argonne National Laboratory 1997). The large-scale use of electric buses may have started during the Beijing Olympics (EV World 2008) and Shanghai

World Expo (United Nations Environment Programme 2009). For example, more than 80 electric buses were used in the Shanghai World Expo (United Nations Environment Programme 2009). However, a fully charged battery only allows a limited running distance. It was reported in the United Nations Environment Programme (2009) that the operational distance of an electric bus is 150 km when an air conditioner is on. In all the aforementioned studies, the current battery is exchanged with a fully charged one at a battery service station. Although electric buses have been used for real operation in Beijing and Shanghai, no scheduling models were proposed, possibly because buses there are generally operated by frequency (Shen and Xia 2009). CNG, diesel, and hybrid buses also have a limited operational distance because of their limited tank capacity. Scheduling electric, CNG, diesel, or hybrid transit buses is named *transit bus scheduling with limited energy* in this paper.

The transit bus scheduling problem with limited energy belongs to the vehicle-scheduling problem with route constraints (called *VSP-RC* in Bunte and Kliewer 2009). The route constraints of the transit bus scheduling problem include (1) the maximum route distance for CNG, diesel, or hybrid buses, and (2) the maximum distance before battery renewal for electric buses. An important VSP-RC problem is the vehicle-scheduling problem with the maximum route time constraint, where the total time during which a vehicle is away from its depot is no more than a prespecified threshold (Bodin et al. 1983; Freling and Paixão 1995; Haghani and Banihashemi 2002; Wang and Shen 2007). Such a vehicle-scheduling problem can take advantage of a special problem structure to characterize a feasible path: the total route time depends on the first and last trips rather than the intermediate trips. Certain studies use this useful structure to build the optimization models, for example, the back-arc approach proposed by Bodin et al. (1983) and several graph models by Freling and Paixão (1995). However, in the transit bus scheduling problem with the maximum route distance constraint, all the trips in the route contribute to the total distance, which makes it impossible to exploit this useful structure. We will show in §2 that the use of the total traveled distance may model fuel consumption more accurately, because waiting usually does not consume fuels. Kliewer, Gintner, and Suhl (2008) study another VSP-RC problem, where bus companies impose a practical constraint on bus line changes on vehicle routes.

Another problem relevant to the VSP-RC is the integrated vehicle and crew scheduling problem, which has been considered as early as the 1980s (e.g., see Ball, Bodin, and Dial 1983) and has gained increasing attention since the late 1990s (e.g., see Freling 1997; Friberg and Haase 1999; Freling, Huisman, and Wagelmans 2001a; Haase, Desaulniers, and Desrosiers 2001; Huisman, Freling, and Wagelmans 2005). We refer interested readers to the introduction section in Steinzen et al. (2010) for a recent review. The integrated vehicle- and crew-scheduling problem has to consider practical constraints on the route, for example, the maximum working hours without a break. Because of the common constraint on bus routes, certain techniques that are proposed in integrated vehicle and crew scheduling have also been used in this paper, such as time-space network modeling (Steinzen et al. 2010) and column generation-based algorithms (Freling 1997; Haase, Desaulniers, and Desrosiers 2001; Sandhu and Klabjan 2007; Steinzen et al. 2010). Other relevant studies include column generation and constrained shortest-path modeling appearing in train and airline scheduling (Lusby et al. 2011; Dunbar, Froyland, and Wu 2012; Petersen et al. 2012; Saddoune et al. 2012).

The contributions of this paper are as follows: First, we provide an extensive review on operational and environmental aspects of CNG, diesel, hybrid, and electric buses. Second, we propose (1) a vehicle-scheduling model for electric transit buses with either battery swapping or fast charging, and (2) a vehicle-scheduling model with the maximum route distance constraint for CNG, diesel, or hybrid buses. Third, we develop column-generation-based algorithms to solve the proposed optimization problems. We use real-world instances and instances randomly generated in a practical setting to examine the performance of our algorithms and compare with CPLEX. Finally, based on computational results, we systematically analyze the number of buses needed, total operational costs, and vehicle emissions when CNG, diesel, hybrid, or electric buses are used in service.

2. Operational, Energy, and Environmental Aspects of Transit Buses

In this section, we present certain operational, energy, and environmental aspects of electric, CNG, diesel, and hybrid-diesel buses.

2.1. Electric Buses and Battery Service Stations

Electric buses have been used in transit agencies since the 1990s, for example, in Santa Barbara, California (Griffith and Gleason 1996) and Chattanooga, Tennessee (Dugan 1994; Argonne National Laboratory 1997). One of the most important components in an electric vehicle is its battery. Haggis and Beback (2010) listed rechargeable batteries used in electric transit buses from a number of manufacturers; the battery types include nickel cadmium, lithium ion, lithium iron phosphate, and sodium nickel chloride. A fully charged battery allows an electric bus to run a limited distance. Haggis and Beback (2010) showed that the maximum distance ranges from 190 km to 500 km, but no real-world tests were reported. The maximum operational distance was rather short for electric buses used in the 1990s. For example, it ranged from 64 km to 96 km for electric buses with 22–25 seats in Chattanooga (Argonne National Laboratory 1997). The use of an air conditioner substantially reduces the maximum operational distance. It was reported that the maximum distance was 130 km in the Beijing Olympics (EV World 2008) and 150 km in the Shanghai World Expo (United Nations Environment Programme 2009) with the air conditioner on. With the air conditioner off, the maximum distance was 250 km in the Shanghai World Expo (United Nations Environment Programme 2009).

The limited operational distance makes an electric bus difficult to run for a whole day. A major solution is to use a fully-charged battery to replace the existing battery before it is depleted. Such a replacement is called *battery swapping* in this paper. Battery swapping has been employed in real-world service, such as in Chattanooga (Argonne National Laboratory 1997), Beijing (EV World 2008), and Shanghai (United Nations Environment Programme 2009). The limited distance can also be remedied by fast charging at a station. Kim (2011) shows that nine electric buses were used in 2011 in Seoul with fast charging, and the maximum distance was about 80 km. An electric bus has also been used in service in Foothill Transit of Los Angeles since 2010; Barry (2010) shows that the bus can run for three hours before fast charging.

Both battery swapping and fast charging are generally conducted at a battery service station. From the points of view of operations, a battery service station has two important aspects: *the battery service time* and *the station capacity*. It is reported that the time of each battery swapping was 10–15 minutes in Chattanooga (Argonne National Laboratory 1997), and about 10 minutes in Beijing and Shanghai (Chang 2010). The fast charging time is about 10 minutes in Foothill (Barry 2010) and from 20 to 30 minutes in Seoul (Kim 2011). A battery service station also has a limited capacity, i.e., the maximum number of buses whose batteries can be serviced simultaneously. Battery swapping can be done for eight buses in Shanghai (Chang 2010).

Another important issue is the costs of charging depleted batteries. Haggis and Beback (2010) show that it can take about eight hours for a battery charger with 400V/100A to fully charge a battery whose operational distance is approximately 310 miles. If the average electricity price is 10.26 cents per kWh (Energy Information Administration 2011), the electricity cost per kilometer can be estimated as 6.61935 cents. It is worth mentioning that battery swapping has another advantage in comparison with fast charging: replaced batteries can be charged in the night at a discounted electricity price, for example the case in Shanghai (United Nations Environment Programme 2009).

Table 1 The Tank Capacity of Some Transit Bus Engines

Manufacturer	Power	Engine	Seats	Tank capacity (gallons)
VOLVO	Diesel	7900	31	56.797–79.252
VOLVO	CNG	7900 CNG	31	56.533
VOLVO	Hybrid	7900 HYBRID	33 or 37	56.797–66.043
NABI	Diesel	31-LFW	25	85.000
NABI	Diesel	35-LFW	30	125.000
NABI	Diesel	40-LFW	40	125.000

Note. The VOLVO engine data are from http://www.volvobuses.com/bus/global/en-gb/products/City buses/volvo_7900_hybrid/Pages/downloads.aspx, and the NABI engine data are from http://www.nabiusa.com/nabi/nabi-LFW-specs.htm.

2.2. CNG, Diesel, and Hybrid Buses: Tank Capacity, Fuel Economy, and Emissions

CNG, diesel, and hybrid buses also have a limited operational distance because of the restriction of tank capacity. Table 1 lists the tank capacity for a number of bus engines. Some field studies have been conducted to estimate the fuel economy of transit buses with different power types (e.g., see Cohen, Hammitt, and Levy 2003; Environment Canada 2004; Clark et al. 2007). Columns 1 and 2 in Table 2 list the bus type and the fuel economy, respectively. The average fuel price can be obtained from a website of Department of Energy, as seen in column 3 in Table 2. Then, the average fuel cost per kilometer can be simply calculated, as shown in column 4. The tank capacity data in Table 1 (column 5, rows 1–3) are used as the given tank capacity, as seen in column 5 in Table 2. Based on the fuel economy and tank capacity (columns 2 and 5), we can estimate the maximum operational distance in column 6. In practice, transit agencies often use the 80% rule: before 80% fuel is consumed, the

Table 2 The Fuel Efficiency, Fuel Price, and Estimated Maximum Operational Distances

				Estimated distances		
Bus type	Fuel economy (km/gallon)	Fuel price ($/gallon)	Fuel cost ($/km)	Given tank cap (gallon)	Max dist (km)	80% max dist (km)
CNG	5.2626[a]	2.09[c]	0.397142	56.533	298	238
Diesel	5.6327[b]	3.81[c]	0.676407	56.797	320	256
Hybrid	7.3708[a]	3.81[c]	0.516905	56.797	419	335

[a]From Clark et al. (2007).
[b]From Table 2 of Cohen, Hammitt, and Levy (2003).
[c]From the website of US Department of Energy (http://www.afdc.energy.gov/afdc/pdfs/Oct_2011_AFPR.pdf).

Table 3 The Estimation of Bus Emission Rates

	Emission rate (g/km)		
Bus type	NO_x	PM	CO_2
CNG[a]	1.7647	0.0143	1,354.5892
Diesel[b]	17.8334	0.1988	1,739.8393
Hybrid[a]	2.5725	0.0106	1,161.3428

[a]From Clark et al. (2007).
[b]From Table 2 of Cohen, Hammitt, and Levy (2003).

bus should return to the depot; column 7 in Table 2 provides such information.

As mentioned before, transit buses generate a large amount of emissions. The chassis dynamometer is a widely used technique to estimate vehicle emissions, consisting of a predescribed driving cycle to simulate typical driving conditions. The Central Bus District (CBD) cycle is the most commonly used cycle. Table 3 presents the emission rate on particulate matters (PM), nitrogen oxides (NO_x), and carbon dioxide (CO_2) for CNG, diesel, and hybrid-diesel buses using the CBD cycles (Cohen, Hammitt, and Levy 2003; Clark et al. 2007). The emission rate is in unit of grams per kilometer. Table 3 clearly shows that diesel buses generate much more emissions than CNG and hybrid buses.

3. Models

In this section, we first present an optimization model for transit bus scheduling with either battery swapping or fast charging. We then propose an optimization model for transit bus scheduling with the maximum route distance constraint.

3.1. Electric Bus Scheduling with Battery Renewal

Given the ending time of each trip and the travel time between any pair of locations, the arrival time from a trip to each battery station can be determined. We define a maximum waiting time at each battery station to avoid unnecessarily long waiting. Then, we can determine the range of service starting time at a battery station for each trip. As mentioned in §2.1, two major factors of a battery service station are the station capacity and the battery service time. To model the station capacity, we use a discretization approach to transform the range of service start times into a set of discretized nodes. Such discretized nodes are called *time-expanded station nodes* in this paper. The sampling time step is set as one minute.

Let B be the set of available battery stations. For each $b \in B$, let T_b be the set of time-expanded station nodes associated with b, and $T = \bigcup_{b\in B} T_b$ is the set of all time-expanded station nodes. Let battery swapping (or fast charging) time at node $t \in T$ be U_t. For each $t \in T$, let a_t be service starting time, and $a_t + U_t$

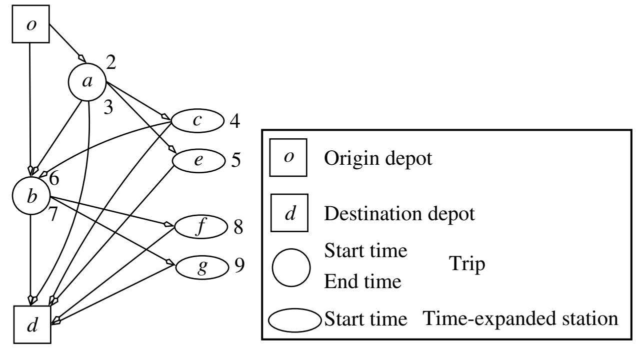

Figure 1 An Example of the Underlying Network

service ending time. For each $t \in T$, let V_t be the station capacity and $p^k(t)$ be the time-expanded node that is k time steps ahead of node t. Let S be the set of service trips. For each trip $s \in S$, let a_s be trip starting time and b_s be trip ending time. We consider the case with a single depot. For the notational convenience, let o and d simply represent the origin and destination depot, respectively. Let r_{ij} be the travel time from $i \in \{o \cup T\}$ or the starting point of trip $i \in S$ to $j \in \{d \cup T\}$ or the ending point of trip $j \in S$. Let the set of arcs be A, which includes the arcs: (1) from o to $i \in S$; (2) from $i \in S$ to $j \in S$ if $b_i + r_{ij} \leq a_j$; (3) from $i \in S$ to d; (4) from $i \in S$ to $t \in T$ if $b_i + r_{it} \leq a_t$; (5) from $t \in T$ to $i \in S$ if $a_t + U_t + r_{ti} \leq a_i$; and (6) from $t \in T$ to d. The travel (1) from the ending point of a trip to the starting point of another trip, battery stations, or the destination depot, or (2) from the origin depot to the starting point of a trip is called *deadheading*.

An example is used to illustrate the process of constructing the underlying network. In Figure 1, nodes a and b are trips, and there is one battery station. Let the travel time along any arc, the maximum waiting time at the station, and the battery service time be one unit. The arrival time from trip a to the battery station is 4. The range of service starting time for trip a at the battery station is $[4, 5]$. If the sampling interval is 1, two time-expanded station nodes are generated: one is with starting time 4 and ending time 5 (node c), and the other is with starting time 5 and ending time 6 (node e). Similarly, nodes f and g are time-expanded station nodes for trip b. Nodes c and e are two consecutive time-expanded nodes. Therefore, $p^1(e) = c$. Note that there is no connection from node e to node b, since the arrival time is 7, which is later than 6, the starting time of trip b.

We now introduce decision variables. Let x_{ji} be a binary decision variable, with $x_{ji} = 1$ if a vehicle is assigned to node i directly after node j, and $x_{ji} = 0$ otherwise. Let g_s be a continuous decision variable, indicating the accumulative distance traveled to the ending point of trip $s \in S$ since the latest battery renewal. We then introduce parameters. Let d_{ij} be the distance traveled from node i to node j, which equals the deadheading distance from i to j plus the trip distance of j if $j \in S$. Let D be the maximum distance that a fully-charged battery allows. Let w_{ij} be the waiting time on node j. Let c_d be the cost of unit traveled distance, and c_w be the cost of unit waiting time. Let C_v be the fixed-vehicle maintenance cost, and C_b be the cost for each battery swapping (or fast charging) service. The cost of arc (ji) is defined as: (1) $c_{ji} = c_d d_{ji} + C_v$ if $j = o$; (2) $c_{ji} = c_d d_{ji} + c_w w_{ji} + C_b$ if $i \in T$; and (3) $c_{ji} = c_d d_{ji} + c_w w_{ji}$ for other arcs. Let K be the maximum number of vehicles that can be used.

The single-depot vehicle-scheduling problem for electric buses with battery swapping (or fast charging) is formulated as follows:

(Model ARC-E)

$$\min \sum_{(j,i)\in A} c_{ji} x_{ji} \tag{1}$$

$$\text{s.t.} \sum_{j:(j,i)\in A} x_{ji} = 1 \quad \forall i \in S, \tag{2}$$

$$\sum_{i:(j,i)\in A} x_{ji} - \sum_{i:(i,j)\in A} x_{ij} = 0 \quad \forall j \in S \cup T, \tag{3}$$

$$g_i = \sum_{j:(j,i)\in A} (g_j + d_{ji}) x_{ji} \quad \forall i \in S, \tag{4}$$

$$g_i = 0 \quad \forall i \in T \cup \{o\}, \tag{5}$$

$$(g_j + d_{jt}) x_{jt} \le D \quad \forall t \in T \cup \{d\}, (j,t) \in A, \tag{6}$$

$$\sum_{j:(j,t)\in A} x_{jt} + \sum_{\substack{i=p^1(t)\\ j:(j,i)\in A}} x_{ji} + \cdots + \sum_{\substack{i=p^{U-1}(t)\\ j:(j,i)\in A}} x_{ji} \le v_t \quad \forall t \in T, \tag{7}$$

$$\sum_{i:(j,i)\in A} x_{ji} \le K \quad j = o,$$

$$x_{ij} \in \{0,1\} \quad (i,j) \in A. \tag{8}$$

Objective (1) is to minimize total operational costs. Constraints (2) and (3) represent covering and flow conservation. Constraints (4) determine the accumulative distance traveled since the latest battery renewal. Constraints (5) guarantee that the accumulative distance is 0 when a vehicle just starts from the depot or finishes the battery service at a station. Constraints (6) force that the maximum route distance restriction cannot be violated. Constraints (7) ensure that the station capacity constraint has to be satisfied. The impact of battery service time on the station capacity is handled by $p^1(t), \ldots, p^{U-1}(t)$ in constraints (7). Constraints (8) guarantee that the total number of vehicles used is no more than the upper bound.

It is easy to show that Model ARC-E is NP-hard: if a very large penalty is imposed for each arc ji with $i \in T$, no node in T will appear in any optimal solution. Therefore, this special case of Model A is equivalent to the vehicle-scheduling problem with the maximum route distance. Bodin et al. (1983) show that any resource-constrained vehicle-scheduling problem is NP-hard. Therefore, Model ARC-E is NP-hard.

Model ARC-E is suitable to both battery swapping and fast charging. In the current technology, the time used in battery swapping is similar to the time used in fast charging (see §2). An electric bus with battery charging generally has much longer operational distances than an electric bus with fast charging. However, there are no major model differences between them.

3.2. Transit Bus Scheduling with the Maximum Route Distance Constraint

Transit bus scheduling with the maximum route distance constraint is mainly suitable for diesel, CNG, and hybrid transit buses, which have much longer operational distances than electric buses.

All the set S, o, d, parameter D, and variables x_{ji} and g_s have the same definitions as in Model ARC-E. The major difference is that there are no nodes or arcs representing a battery station. The set of arcs (A) includes arcs (1) from o to $i \in S$; (2) from $i \in S$ to $j \in S$ if $b_i + r_{ij} \le a_j$; and (3) from $i \in S$ to d. The single-depot vehicle-scheduling model with the maximum route distance constraint is formulated as follows:

(Model ARC-NE)

$$\min \sum_{i:(j,i)\in A} c_{ji} x_{ji} \tag{9}$$

$$\text{s.t.} \sum_{j:(j,i)\in A} x_{ji} = 1 \quad \forall i \in S, \tag{10}$$

$$\sum_{j:(j,i)\in A} x_{ji} - \sum_{i:(i,j)\in A} x_{ij} = 0 \quad \forall j \in S, \tag{11}$$

$$g_i = \sum_{j:(j,i)\in A} (g_j + d_{ji}) x_{ji} \quad \forall i \in S, \tag{12}$$

$$g_o = 0, \tag{13}$$

$$(g_j + d_{jd}) x_{jd} \le D \quad \forall (j,d) \in A, \tag{14}$$

$$\sum_{i:(j,i)\in A} x_{ji} \le K \quad j = o,$$

$$x_{ij} \in \{0,1\} \quad (i,j) \in A. \tag{15}$$

In comparison with Model ARC-E, there are no time-expanded station nodes (T) and battery capacity constraints (7). As a resource-constrained vehicle-scheduling problem, Model ARC-NE is also NP-hard (Bodin et al. 1983). An alternative formulation is to use a multicommodity model by introducing a three-index variable x_{ji}^k, where k represents a specific vehicle. Constraints (14) can be replaced by $\sum_{(j,i)\in A} d_{ji} x_{ji}^k \le D$ for each vehicle. The number of constraints (14) is then equal to the number of potential vehicles, which is generally much smaller than the one in Model ARC-NE. However, the disadvantage is the introduction of a large number of binary variables.

An instance with about 100 trips with this multicommodity model could not be solved by CPLEX in our preliminary tests, whereas CPLEX solved the same instance with Model ARC-NE very quickly.

It is easy to show that the bus-scheduling problem with the maximum route distance constraint is a special case of the electric-bus-scheduling problem with the battery renewal. However, it is more challenging to solve the electric-bus-scheduling problem with battery renewal because of (1) the shorter maximum route distance; (2) a large number of time-expanded station nodes involved; and (3) the additional constraints on the station capacity.

4. Algorithms

In this section, we propose certain column generation-based algorithms for electric transit bus scheduling: a branch-and-price algorithm for small-sized instances and a truncated column generation algorithm for medium- and large-sized instances. These algorithms are then used to solve the bus-scheduling problem with the maximum distance constraint, which is a special case of the electric-bus-scheduling problem.

4.1. A Branch-and-Price Framework

In electric bus scheduling, the restriction on the maximum route distance before battery renewal results in a large number of additional constraints ((4)–(6)). These constraints can be effectively handled by reformulation and Dantzig-Wolfe decomposition.

Let R be the set of all feasible paths from o to d, satisfying the maximum distance constraint before battery renewal. Let c_r be total operational costs of path $r \in R$. Let δ_i^r equal 1 if node $i \in S \cup T$ is covered by path r, and equal 0 otherwise. Let $n^k(t)$ be the time-expanded node that is k time steps later than node t; $n^k(t)$ plays a complementary role of $p^k(t)$. Let binary variable y_r be 1 if path r is selected in the solution. The path-based formulation of electric bus scheduling is as follows:

(Model PATH-E)

$$\min \sum_{r\in R} c_r y_r \tag{16}$$

$$\text{s.t.} \sum_{r\in R} \delta_i^r y_r = 1 \quad \forall\, i \in S, \tag{17}$$

$$\sum_{r\in R} \delta_t^r y_r + \sum_{r\in R} \delta_{p^1(t)}^r y_r + \cdots + \sum_{r\in R} \delta_{p^{U-1}(t)}^r y_r \le v_t \quad \forall\, t \in T, \tag{18}$$

$$\sum_{r\in R} y_r \le K,$$

$$y_r \in \{0,1\} \quad \forall\, r \in R. \tag{19}$$

Objective (16) is to minimize the total operational costs. Constraints (17) ensure that each trip is served once; constraints (18) guarantee that the station capacity constraint has to be satisfied at any moment; and constraint (19) imposes an upper bound on the total number of vehicles used in service.

The linear relaxation of Model PATH-E is solved through column generation by repeatedly solving (1) a *restricted master problem* (RMP) with a subset of columns $\bar{R} \subset R$, and (2) a pricing subproblem to produce columns with negative reduced cost. Even if the linear relaxation of Model PATH-E is solved to optimality by column generation, it is not guaranteed that the resulting optimal solution is integral. Therefore, column generation is generally embedded into a branch-and-bound search framework and is executed at every node of the search tree, resulting in a branch-and-price algorithm (e.g., see Barnhart et al. 1998; Lübbecke and Desrosiers 2005).

4.2. Pricing

Let dual variable α correspond to constraints (16), and let β and π, respectively, correspond to constraints (17) and (18) with the form "$\ge$." The reduced cost of path $r \in R$ is

$$c_r - \sum_{i\in S} \delta_i^r \alpha_i + \sum_{i\in T} \delta_i^r \big(\beta_i + \beta_{n^1(i)} + \cdots + \beta_{n^{U-1}(i)}\big) + \pi.$$

The reduced cost of arc ji, $\bar{c}_{ji}$, then equals: (1) $c_{ji} - \alpha_i$ if $i \in S$; (2) $c_{ji} + \beta_i + \beta_{n^1(i)} + \cdots + \beta_{n^{U-1}(i)}$ if $i \in T$; and (3) c_{ji} if $i = d$. The pricing subproblem consists of finding a path from origin depot o to destination depot d with the maximum distance constraint before battery renewal, which can be formulated as follows:

$$\min \sum_{(j,i)\in A} \bar{c}_{ji} x_{ji} \tag{20}$$

$$\text{s.t.} \sum_{i:(j,i)\in A} x_{ji} = 1 \quad \forall\, j = o, \tag{21}$$

$$\sum_{j:(j,i)\in A} x_{ji} = 1 \quad \forall\, i = d, \tag{22}$$

$$\sum_{i:(j,i)\in A} x_{ji} - \sum_{i:(i,j)\in A} x_{ij} = 0 \quad \forall\, j \in S \cup T, \tag{23}$$

$$g_i = \sum_{j:(j,i)\in A} (g_j + d_{ji}) x_{ji} \quad \forall\, i \in S, \tag{24}$$

$$g_i = 0 \quad \forall\, i \in T \cup \{o\}, \tag{25}$$

$$(g_j + d_{jt}) x_{jt} \le D \quad \forall\, t \in T \cup \{d\},\ (j,t) \in A,$$

$$x_{ij} \in \{0,1\} \quad (i,j) \in A. \tag{26}$$

It can be shown that a special case of the pricing subproblem is the constrained shortest-path problem, which is NP-hard (Garey and Johnson 1979). Therefore, the pricing subproblem is also NP-hard.

As a dynamic programming algorithm, label correcting has been widely used to solve the resource-constrained shortest-path problem (e.g., see Feillet et al. 2004; Righini and Salani 2006). We then design a label-correcting algorithm by considering special problem aspects: maximum distance and battery renewal. Each state in the extension context is represented by a label, (i, g), where i is the last reached node and g represents the accumulative distance traveled since the last battery renewal. The cost of label (i, g) is $\bar{c}(i, g)$, representing the accumulative cost from origin depot o. The first label in the extension is $(o, 0)$, which starts from the depot with distance 0 and cost 0. Now consider that label (i, g) is extended to label $(j, \bar{g})$. If $g + d_{ij} \le D$, the extension is feasible, and $\bar{g}$ is determined as follows: $\bar{g} = 0$ if $j \in T$, and $\bar{g} = g + d_{ij}$ otherwise. Then let us discuss the label dominance. Consider that both label (i, g) and label $(i, \bar{g})$ have node i as their last reached node; label (i, g) dominates label $(i, \bar{g})$ if (1) $\bar{c}(i, g) \le \bar{c}(i, \bar{g})$, and (2) $g \le \bar{g}$. The label extension and dominance are placed into a framework of the general label-correcting algorithm (e.g., see Ahuja, Magnanti, and Orlin 1993, Chap. 5). The label operations here are relatively simple because the underlying network of the vehicle-scheduling problem is acyclic.

When the pricing subproblem is solved, we attempt to generate multiple columns that correspond to the shortest path as well as near-shortest paths if their reduced costs are negative. Extracting multiple paths is easy in dynamic programming if we keep multiple labels in destination depot d. If the number of columns in the pool exceeds a given threshold, we will remove columns from the pool based on their reduced costs.

4.3. Stabilization and Lower Bounding

A major difficulty in column generation may be slow convergence when the solution is near the optimum, which is called the *tailing-off effect*. A widely used approach that mitigates the tailing-off effect is through computing alternative Lagrangian lower bound and applying early termination (e.g., see Vanderbeck and Wolsey 1996). By associating dual variable $\bar{\alpha}$ and $\bar{\beta}$ with constraints (16) and (17) respectively, the Lagrangian dual problem (PATH-LD) can be stated as follows:

$$\max_{\bar{\alpha},\bar{\beta}} \min \left\{ \sum_{r\in R}\left(c_r - \sum_{i\in S}\delta_i^r \bar{\alpha}_i + \sum_{i\in T}\delta_t^r\left(\bar{\beta}_i + \bar{\beta}_{n^1(i)} + \cdots + \bar{\beta}_{n^{U-1}(i)}\right)\right) y_r + \sum_{i\in S}\bar{\alpha}_i - \sum_{i\in T} v_t \bar{\beta}_t) \right\}$$

$$\text{s.t.} \sum_{r\in R} y_r \le K, \quad y_r \in \{0,1\} \quad \forall r \in R.$$

If the optimal value of the pricing subproblem $\bar{c}_r^*$ is negative, it can be shown that $K\bar{c}_r^* + \text{Obj}_{\text{RMP}}$ is a lower bound on the linear relaxation of Model PATH-E, where Obj_{RMP} is the optimal value of the current RMP (e.g., see Farley 1990; Dell'Amico, Righini, and Salani 2006). If this lower bound is greater than or equal to the best incumbent integer solution, the current node is fathomed.

Another important technique for mitigating the tailing-off effect is column generation stabilization. Notable approaches include the primal-dual strategy (du Merle et al. 1999) and the interior point method (Rousseau, Gendreau, and Feillet 2007). Klose and Drexl (2005) and Lübbecke and Desrosiers (2005) provide an extensive review on column generation stabilization. In this paper, we use a stabilization approach proposed by Wentges (1997), named as *weighted Dantzig-Wolfe decomposition*, which has been successfully used in facility location (Klose and Drexl 2005; Klose and Görtz 2007), machine scheduling (Pessoa et al. 2010), and production order batching (Tang et al. 2011). The weighted Dantzig-Wolfe decomposition combines the dual solution from the restricted master problem and Lagrangian multipliers from subgradient search. The overall stabilization approach is outlined as follows.

Step 1 (*Subgradient phase*). Run a given number of iterations on the Lagrangian dual problem (PATH-LD) by (1) solving the pricing subproblem, and (2) updating the Lagrangian dual by subgradient search. Let $\bar{\alpha}^*$ and $\bar{\beta}^*$ correspond to the dual with the best Lagrangian lower bound.

Step 2 (*Column generation phase*).

Step 2.1 (*Restricted master problem*). Solve the restricted master problem in Model PATH-E and let $\hat{\alpha}$ and $\hat{\beta}$ correspond to the optimal dual solutions.

Step 2.2 (*Pricing subproblem*). Let $\alpha = \gamma\hat{\alpha} + (1-\gamma)\bar{\alpha}^*$ and $\beta = \gamma\hat{\beta} + (1-\gamma)\bar{\beta}^*$, where weight γ is between 0 and 1 (0.6 in this paper). Solve the pricing subproblem with α and β as dual variables. If some columns generated have negative costs at dual prices $\hat{\alpha}$ and $\hat{\beta}$, go to Step 2.3; otherwise, go to Step 2.4.

Step 2.3 (*Intermediate subgradient search*). Conduct a number of subgradient steps (10 in this paper), including solving the pricing subproblem and updating Lagrangian dual by subgradient search.

Step 2.4 (*Line search*). Set weight $\gamma^- = \gamma$; increase γ^- by a small amount (0.1 in this paper); set $\alpha = \gamma^-\hat{\alpha} + (1-\gamma^-)\bar{\alpha}^*$ and $\beta = \gamma^-\hat{\beta} + (1-\gamma^-)\bar{\beta}^*$, and solve the pricing subproblem again; repeat these steps until γ^- approaches 1.

Step 2.5 (*Termination criterion*). If a termination criterion is satisfied, stop column generation.

Column generation is stopped in Step 2.5 if (1) there are not any columns with negative reduced cost found in Step 2.4; (2) a small optimality gap has been reached (0.1% for small instances and 0.5% for large instances in this paper); or (3) a given limit of column generation iterations has been reached.

4.4. Branching

If the solution of the restricted master problem is not integral when column generation terminates, a branch-and-bound procedure is applied. We use the standard branching rule on the context of column generation proposed by Ryan and Foster (1981): (1) the solution of path-based model PATH-E is converted into an equivalent solution of the original compact model ARC-E; (2) fractional arc flows (x_{ij}) are then identified, and branching is applied on a fractional arc flow. The underlying network is changed according to $x_{ij} = 1$ or $x_{ij} = 0$. With such a branching scheme, the pricing subproblem structure is preserved after branching. To obtain a feasible solution quickly, we apply the depth-first node selection strategy. When selecting a node on the same level, we always select the node with $x_{ij} = 1$.

4.5. Upper Bounding

4.5.1. A Construction Heuristic to Generate Initial Solutions. An initial solution is needed in column generation. We generate an initial solution using a greedy strategy. We keep a data structure, called the *current node*, which is defined as n. Define *dist* to be the accumulative distance to n since obtaining the latest fully-charged battery. Starting from n, the heuristic attempts to find the next trip. Get a fully-charged battery before the current battery depletes. The vehicle returns to the depot unless there are no other options. The overall framework is outlined as follows.

Step 1. Set the origin depot as n and $dist = 0$. Find an uncovered trip with the earliest starting time, make this trip as n, and update *dist*.

Step 2. Start from n and find an uncovered trip i such that $(ni) \in A$ and c_{ni} is minimal. If no i is found, or n is a trip that has been visited before, go to Step 2.1. Otherwise, go to Step 2.2.

Step 2.1. Examine if returning to the depot from n satisfies the maximum distance constraint. If so, return to the depot and go to Step 3. Otherwise, move backward, update n and *dist*, and go to Step 2.

Step 2.2. Update n and *dist*. If $dist \leq D$, go to Step 2. Otherwise, go to Step 2.3.

Step 2.3. Move backward and update n and *dist*. Among time-expanded station nodes in T, find t with the available station capacity, feasible accumulative route distance, and minimal arc cost. If t is found, go to Step 2.4. Otherwise, go to Step 2.3.

Step 2.4. Update n, set $dist = 0$, and increase the station occupancy. Go to Step 2.

Step 3 (*Termination check*). If at least one trip is uncovered, go to Step 1. Otherwise, stop and examine if the total number of vehicles is more than K.

Our preliminary experiments show that generating multiple initial solutions may be helpful for speeding up the column generation in transit bus scheduling. To produce different initial solutions, we randomly disable some percentage of arcs that appear in the initial solution from the last iteration (we used 20% in case studies) and run the construction heuristic again on the restricted underlying network.

4.5.2. An Iterative Rounding Heuristic. As is well known, good upper bounding is effective to reduce the tree size of branch-and-bound search. We develop an iterative rounding heuristic based on the solution of the restricted master problem: if there exist variables with a fractional value greater than a given threshold (0.7 in this paper), these variables are fixed as 1; otherwise, we fix the variable with the highest fractional value as 1. This process iterates until either an integral solution is obtained or the problem becomes infeasible.

Our preliminary computational results show that such a rounding method often leads to an infeasible model for large instances. The infeasibility is mainly caused by equality constraint (17) ($\sum_{r\in R} \delta_i^r y_r = 1$). After some variables are fixed, some trips may have to be covered more than once by existing columns, thereby resulting in infeasibility. In vehicle routing problems, constraint (17) can generally be relaxed to $\sum_{r\in R} \delta_i^r y_r \geq 1$ without the loss of optimality. However, we may not simply use $\sum_{r\in R} \delta_i^r y_r \geq 1$ to replace $\sum_{r\in R} \delta_i^r y_r = 1$ in vehicle-scheduling problems, because the triangularity is often unsatisfied because of imposed waiting costs. A simple approach is (1) first relaxing $\sum_{r\in R} \delta_i^r y_r = 1$ to $\sum_{r\in R} \delta_i^r y_r \geq 1$, (2) then solving the overall problem, and (3) finally removing overcoverings using a greedy way. We tested this strategy and found that there are often a large number of overcoverings, resulting in a poor final solution.

We design the following penalty-based approach that can effectively reduce overcoverings:

(Model PATH-E-RELAX)

$$\min \left\{ \sum_{r\in R} c_r y_r + \sum_{i\in S} P z_i \right\} \tag{27}$$

$$\text{s.t.} \quad \sum_{r\in R} \delta_i^r y_r = 1 + z_i \quad \forall\, i \in S, \tag{28}$$

$$\sum_{r\in R} \delta_t^r y_r + \sum_{r\in R} \delta_{p^1(t)}^r y_r + \cdots + \sum_{r\in R} \delta_{p^{U-1}(t)}^r y_r \leq v_t \quad \forall\, t \in T, \tag{29}$$

$$\sum_{r\in R} y_r \leq K,$$

$$y_r \in \{0,1\} \quad \forall\, r \in R,$$

$$z_i \in \{0,1\} \quad \forall\, i \in S. \tag{30}$$

We (1) first relax constraint $\sum_{r\in R} \delta_i^r y_r = 1$ as $\sum_{r\in R} \delta_i^r y_r = 1 + z_i$ with z_i as a binary variable in constraints (27), and (2) introduce a penalty for each violation in the objective with P as a penalty constant.

If overcoverings still exist, we remove them in a greedy way. Some preliminary tests are needed to determine a good value of P. In our implementation, P is set about 20 times the average path cost.

4.6. The Heuristic Version for Large-Scale Instances

Although the branch-and-price algorithm has shown its success in many combinatorial optimization problems, it may still be difficult to solve some large-scale instances. A column generation-based heuristic is then developed for large-scale problems. First, we design a truncated column generation approach with variable fixing. Second, we propose a local search method to make improvements based on the solution from the truncated column generation.

4.6.1. Truncated Column Generation with Variable Fixing. Truncated column generation with variable fixing has been successfully used in crew scheduling (Gamache et al. 1999), multiple-depot vehicle scheduling (Pepin et al. 2009), vehicle routing (Prescott-Gagnon et al. 2010), and integrated vehicle and crew scheduling (Steinzen et al. 2010). In our implementation, column generation is first used to solve the path-based formulation PATH-E, which consists of solving the restricted master problem and pricing subproblem, and conducting dual stabilization. After column generation terminates, we will apply variable fixing based on the fractional values of the latest restricted master problem: all variables y_r with fractional values greater than or equal to a given threshold value (0.7 in the implementation) are fixed as 1; if no y_r is fixed, the variable with the largest fractional value is fixed as 1. The trips that are covered by the fixed variables are removed from the problem. Column generation is repeated on the reduced problem until all the trips are covered. The major difference from the branch and price is that there is no tree search in truncated column generation.

However, our preliminary results show that such an approach may make the restricted master problem infeasible for large-scale instances after a number of variables are fixed. The infeasibility is also mainly caused by constraints (17): some trips may have to be covered more than once with the generated columns. We mitigate the potential infeasibility issue by generating some extra paths as follows: (1) initially, we use Model PATH-E to solve the restricted master problem; (2) if the current master problem is infeasible, we utilize the relaxed model PATH-E-RELAX to find a feasible solution; and finally, (3) we examine if some trips are covered more than once. If so, we list the set of paths (called set A) that covers these overcovered trips. For each path in A, we generate some new paths as follows: (i) if there is only one overcovered trip, we simply remove this trip and generate a new path (see Figure 2(a) for an example); (ii) if there are multiple overcovered trips, we generate a new path by removing all the overcovered trips and generate a few paths by removing each overcovered trip respectively (see Figure 2(b) for an example). We did not consider other complicated combinatorial options. This strategy of generating new paths is effective to handle the infeasibility issue we experienced in certain large instances.

We also attempted to use Model PATH-E-RELAX directly in column generation. The performance of this strategy is quite poor because the dual solution of PATH-E-RELAX is very sensitive and cannot provide useful information. We feel that the large value of P and problem degeneration may be the cause of the sensitivity.

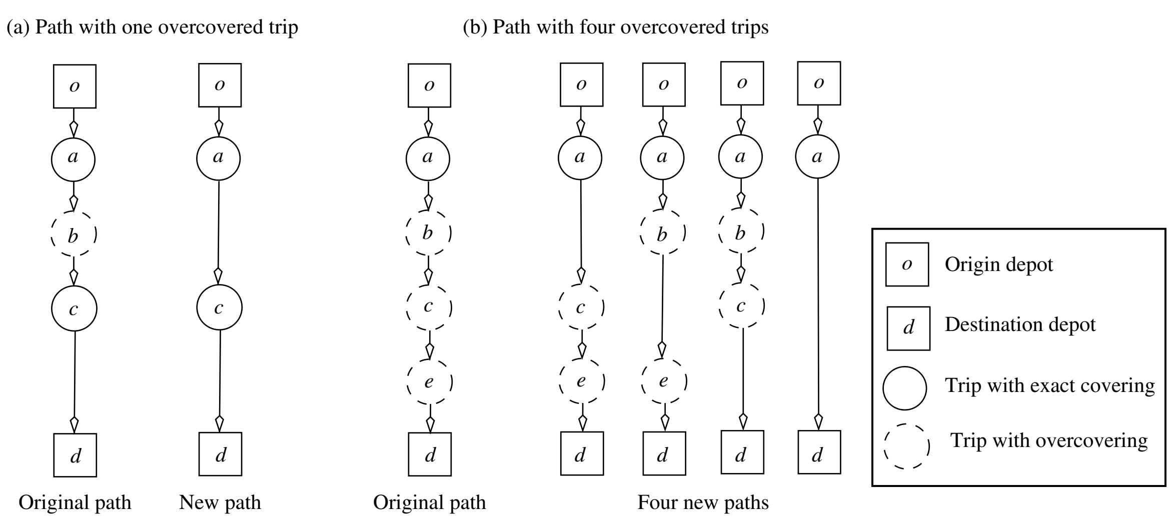

Figure 2 **An Example of Newly Added Paths**

4.6.2. Local Search. The solution generated by truncated column generation is then improved by local search. Local search has been widely used for solving the vehicle-routing and scheduling problem (see Bräysy and Gendreau 2005 for a recent review). Common neighborhood in local search includes interroute operation (e.g., see Lin 1965) and intraroute relocation (e.g., see Or 1976). We customize the classic interroute operations: 2-opt* (Lin 1965; Potvin et al. 1996; Ibaraki et al. 2005) and cross exchange (Taillard et al. 1997). The special problem aspects are considered, including the time-expanded station nodes, station capacity, and maximum route distance constraint. We also develop an integrated interroute and intraroute operation, *relocation first–exchange second*. Consider that in path 1, a bus needs to visit a set of trips, renew the battery from a station, and then continue to serve other trips. If the bus travels toward the battery station slightly earlier or later, it is possible that the total cost is decreased. The major issue of the relocation operation is that one trip around the time-expanded station node may not be covered because of violating the connection constraint. Our approach is to exchange with another path (path 2) to attain the covering feasibility: the trips with the violated connection constraint are inserted into the appropriate positions in path 2, and the removed trips from path 2 are inserted into path 1. An improved solution is found if such exchange is feasible, and overall costs are decreased. The sequence of three operations in the local search is the relocation first–exchange second, cross-exchange, and 2-opt*. The whole local search terminates if there is no improvement.

5. Case Studies

The case studies consist of three parts: in the first part, we discussed data acquisition and parameter estimation; in the second part, we tested the performance of the branch-and-price algorithm and truncated column generation heuristic and compared with the MIP solver of CPLEX 11.2; and in the third part, we investigated total operational costs and bus emissions when CNG, diesel, hybrid, or electric buses were used.

5.1. Data Acquisition and Parameter Estimation

In our studies, we used transit data from the network of County Connection, a transit agency in the Bay Area. The data are publicly available at http://www.gtfs-data-exchange.com/agency/county-connection in the GTFS format, including the locations of stops and the scheduled times of trips at each stop.

To calculate the distance of deadheading arcs, we determined the shortest distance between all pairs of locations. We first mapped a stop to the nearest intersection using point-to-curve matching (Li 2012). We then solved a one-to-one shortest-path problem between each intersection with the Dijkstra algorithm based on the road geometry network data provided by NavTeq. The deadheading times were then determined using the obtained shortest distances and the average vehicle speed, which was set as 25 mph in this study. Fixed vehicle cost C was set as \$100. Note that in this study, the fixed vehicle cost mainly means the maintenance cost instead of the vehicle purchase cost. Unit waiting cost c_w was set as \$0.1 per minute, and unit travel cost c_d was set as \$0.397142 per km for CNG buses, \$0.676407 per km for diesel buses, \$0.516905 per km for hybrid buses, and 6.61935 cents per km for electric buses, respectively (see §2 for the details). The cost of each service at a battery station was set as \$5. The maximum route distance was set as 238 km for CNG buses, 256 km for diesel buses, and 335 km for hybrid buses, respectively. Because the maximum operational distance for electric buses is much shorter, we tested two different values: 120 km and 150 km. The battery service time was set as 10 minutes. These settings are in accordance with real-world testing. It is worth mentioning that although there are no model differences between battery swapping and fast charging, such parameter settings actually represent the use of battery swapping instead of battery charging. In the vehicle-scheduling problem for electric buses, the maximum waiting time at a battery station was set as 30 minutes. We assume that there exists one battery service station with capacity 2 that is located at the depot.

The data from County Connection include two sets of timetabled trips: weekday schedules and weekend schedules. To conduct more tests, we divided each schedule into smaller parts and also generated certain instances by randomly shifting the trip starting time within $[-10, 10]$ minutes from the original value. The ending time was also adjusted by keeping the same trip duration. Table 4 describes the problem instances, where columns 3 and 7 provide the number of time-expanded station nodes for electric bus scheduling.

When determining the value of the maximum number of vehicles (K), we first ran the construction heuristic and obtained the number of vehicles in operation. We then set K based on this number with a slight increase: (1) for instances Sq1, Sq2, Sq3, and Sq4, K was set as 10; (2) for instances Sh1, Sh2, and variants, K was set as 15; (3) for instance S and its variants, K was set as 25; (4) for instances Wh1, Wh2, and their variants, K was set as 70; and finally, (5) for instance W and its variants, K was set as 140 for electric buses with maximum distance 120 km, whereas in other cases, K was set as 100.

5.2. Computational Experiments

We implemented our algorithm in C++ on a Linux Workstation with 2 GHz CPU and 16 GB of RAM.

Table 4 Description of Problem Instances

Inst	No. of trips	No. of time-expanded nodes	Description	Inst	No. of trips	No. of time-expanded nodes	Description
S	242	952	Published weekend schedule	W	947	1,132	Published weekday schedules
Sa		951	S is randomly adjusted	Wa		1,136	W is randomly adjusted
Sb		943	Same as above	Wb		1,136	Same as above
Sc		930	Same as above	Wc		1,130	Same as above
Sh1	117	768	First half of S	Wh1	463	1,073	First half of W
Sh1a		761	Sh1 is randomly adjusted	Wh1a		1,087	Wh1 is randomly adjusted
Sh1b		761	Same as above	Wh1b		1,074	Same as above
Sh1c		779	Same as above	Wh1c		1,084	Same as above
Sh2	125	900	Second half of S	Wh2	484	1,132	Second half of W
Sh2a		904	Sh2 is randomly adjusted	Wh2a		1,131	Wh2 is randomly adjusted
Sh2b		892	Same as above	Wh2b		1,124	Same as above
Sh2c		909	Same as above	Wh2c		1,126	Same as above
Sq1	67	838	First quarter of S	Sq3	58	725	Third quarter of S
Sq2	63	741	Second quarter of S	Sq4	54	659	Fourth quarter of S

We used g++ 4.4 to compile C++ codes. When implementing Model ARC-E and Model ARC-NE in CPLEX, we used the big-M technique to linearize $g_j x_{ji}$ appearing in these models. The default solver of GUROBI 5.1 with the saved basis was used to solve the linear programs. All the CPU time reported here is in the unit of seconds.

5.2.1. Electric Bus Scheduling with Battery Swapping. We discuss the results of the electric-bus-scheduling problem in this section. Table 5 presents computational efficiency for instances Sq1, Sh1, Sh2, S, and their invariants. Columns 1, 2, 3, and 4 provide the problem instance, the number of trips, the number of time-expanded nodes, and the maximum distance before battery swapping, respectively. Columns 5–8 present the computational results on Model ARC-E by CPLEX: the lower bound, the objective value, the optimality gap, and the CPU seconds, respectively. Columns 9–15 report the results by the branch and price: the lower bound, the objective value, the optimality gap, the number of nodes visited, the CPU seconds spent in the root node, the CPU seconds spent in solving the restricted master problems, and the total CPU seconds, respectively. The optimality gaps presented in columns 7 and 11 are defined as (corresponding objective value − best lower bound)/best lower bound, where the best lower bound is the maximum one between two lower bounds presented in columns 5 and 9. We set the maximum running time of CPLEX as 2, 4, and 6 hours for different instances. The branch-and-price algorithm is terminated when the maximum number of tree nodes visited is reached (200 for Sq1–Sq2; 50 for Sh1, Sh2, and their invariants; and 100 for S and their invariants) or the optimality gap is sufficiently small (0.1% for Sq1, Sh1, Sh2, and their invariants; and 0.5% for S and their invariants).

As shown in columns 5–8 in Table 5, whereas CPLEX found optimal or near-optimal solutions for some small instances (Sq1 and invariants), it often resulted in a large optimality gap for the other instances. CPLEX generally failed to obtain a feasible integer solution for the instances with more than 200 trips (S and invariants). The maximum route distance has a great impact on the CPLEX performance: when the maximum route distance is 150 km, CPLEX obtained feasible integer solutions for four instances (Sh2, Sh2a, Sh2b, and Sh2c), even though with a large optimality gap. However, CPLEX obtained an integer solution for only one instance (Sh2a) when the maximum route distance is 120 km. In comparison with CPLEX, the branch and price found optimal or near-optimal integer solutions for all the instances. The worst one happens in instance Sb with the maximum distance 120 km, where the optimality gap is 1.04%. All the solutions from the branch and price are better or the same as ones from CPLEX. A lower bound from the branch and price is generally stronger than the corresponding lower bound from CPLEX. In general, the smaller route distance also leads to the longer computational time of the branch-and-price algorithm. However, the solution quality is not much different. On average, the computational time of the branch and price is much shorter than the time of CPLEX. Table 5 also shows that the majority of the CPU time is consumed in solving the pricing problems.

It is worth mentioning that the columns can be generated from three sources in our paper: (1) the multiple runs of the construction heuristic with some percentage of arcs being randomly disabled (see §4.5.1); (2) the multiple columns generated in each run of the pricing subproblem (see §4.2); and (3) the additional columns generated using Model PATH-E-RELAX (see §4.6.1). Therefore, the number

Table 5 The Branch-and-Price Algorithm for Electric Bus Scheduling

				ARC-E by CPLEX				Branch and price						
Inst	Trip	Time node	Max dist	Lower bound	Obj	Gap (%)	CPU	Lower bound	Obj	Gap (%)	Nodes visited	Root CPU	RMP CPU	Total CPU
Sq1	67	838	150 km	10,193	10,205	0.12	7,200	10,184	10,202	0.09	200	82	98	661
Sq2	63	741		11,517	11,517	0.00	2	11,508	11,517	0.00	3	63	3	79
Sq3	58	725		8,377	8,378	0.01	6,571	8,365	8,378	0.01	59	111	16	356
Sq4	54	659		11,451	11,451	0.00	2	11,449	11,451	0.00	36	38	7	96
Sh1	117	768		13,083	13,316	1.63	14,400	13,104	13,116	0.09	1	419	52	436
Sh1a		761		15,133	15,167	0.22	14,400	15,121	15,134	0.01	1	337	22	354
Sh1b		761		14,172	14,334	1.01	14,400	14,191	14,210	0.13	50	402	87	1,788
Sh1c		779		14,315	14,362	0.32	14,400	14,316	14,351	0.24	50	369	117	2,847
Sh2	125	900		10,998	13,113	18.21	14,400	11,093	11,132	0.35	50	1,075	329	5,844
Sh2a		904		14,527	15,670	7.39	14,400	14,592	14,609	0.12	50	552	167	3,060
Sh2b		892		13,497	16,116	18.42	14,400	13,609	13,625	0.12	50	730	218	4,245
Sh2c		909		13,652	17,822	30.37	14,400	13,670	13,706	0.26	50	707	475	2,875
S	242	952		17,625	24,144	36.47	21,600	17,692	17,776	0.47	54	1,641	840	6,671
Sa		951		20,265	+	+	21,600	20,413	20,511	0.48	48	1,254	920	5,847
Sb		943		19,969	+	+	21,600	20,129	20,237	0.54	100	1,150	798	8,889
Sc		930		20,048	+	+	21,600	20,187	20,318	0.65	100	1,030	970	8,993
Sq1	67	838	120 km	10,211	10,307	0.94	7,200	10,233	10,268	0.34	200	110	140	998
Sq2	63	741		11,517	11,517	0.00	1,251	11,502	11,517	0.00	200	63	139	638
Sq3	58	725		8,376	8,425	0.59	7,200	8,413	8,425	0.14	200	107	105	832
Sq4	54	659		11,452	11,453	0.01	10	11,451	11,453	0.01	3	29	4	47
Sh1	117	768		13,083	17,107	30.76	14,400	13,212	13,228	0.12	50	449	295	2,714
Sh1a		761		15,134	16,480	8.89	14,400	15,196	15,220	0.16	50	352	305	2,423
Sh1b		761		14,172	16,903	19.27	14,400	14,312	14,319	0.05	14	387	129	1,122
Sh1c		779		14,315	+	+	14,400	14,433	14,452	0.13	50	587	282	2,748
Sh2	125	900		10,999	+	+	14,400	11,197	11,221	0.21	50	820	577	4,656
Sh2a		904		14,528	20,003	37.69	14,400	14,720	14,748	0.19	50	609	543	4,672
Sh2b		892		13,498	+	+	14,400	13,716	13,739	0.17	50	813	577	5,688
Sh2c		909		13,563	+	+	14,400	13,801	13,847	0.33	50	469	609	5,556
S	242	952		17,626	+	+	21,600	17,904	18,024	0.67	100	2,102	1,396	10,193
Sa		951		20,266	+	+	21,600	20,639	20,764	0.61	100	1,586	1,722	11,285
Sb		943		19,972	+	+	21,600	20,379	20,590	1.04	100	1,549	1,600	13,544
Sc		930		20,049	+	+	21,600	20,457	20,627	0.83	100	1,329	1,150	10,170

Note. +: CPLEX did not find a feasible integer solution.

of the total columns does not exactly represent the column generation process. We then decide not to report the number of the total columns generated. Instead, the difference between the total CPU seconds and the CPU seconds spent in solving the restricted master problems can provide the information of the time consumed in the pricing subproblems.

Although the branch-and-price algorithm performed well for small-sized instances with less than 300 trips, it spent a very long time solving the medium-sized instances with more than 400 trips (Wh1, Wh2, W, and their invariants). We then switched to the column-generation-based heuristic: truncated column generation with variable fixing. The results are shown in Table 6. Columns 1–4 have the same definition as in Table 5. Columns 5 and 6 provide the lower bound and the CPU seconds by CPLEX. CPLEX was run for 12 hours. Columns 7–13 present the results by the truncated column generation: the lower bound, the objective value, the optimality gap, the number of column generation iterations, the CPU seconds spent in the root node, the CPU seconds spent in solving the restricted master problems, and the total CPU seconds, respectively. The optimality gap presented in column 9 is defined as (objective value − best lower bound)/best lower bound, where the best lower bound is the maximum one between two lower bounds presented in columns 5 and 7. Columns 14–16 report the results by the local search improvement based on the solutions obtained from the truncated column generation: the objective value, the optimality gap, and the CPU seconds.

CPLEX failed to find an integer solution for any of the medium-sized instances with more than 400 trips, whereas the truncated column generation obtained very satisfying solutions for these instances: the optimality gap is less than 1% for most of them. The computational time of the truncated column generation is generally more than four hours. Table 6 also shows that the shorter maximum route distance results in

Table 6 The Truncated Column Generation for Medium-Sized Instances

				ARC-E by CPLEX		Truncated column generation							Local search imp		
Inst	Trip	Time node	Max dist	Lower bound	CPU	Lower bound	Obj	Gap (%)	CG iter	Root CPU	RMP CPU	Total CPU	Obj	Gap (%)	CPU
Wh1	463	1,073	150 km	44,140	43,200	44,971	45,225	0.56	29	5,333	2,512	14,761	45,225	0.56	10
Wh1a		1,087		48,303	43,200	49,099	49,435	0.68	35	8,078	3,630	18,338	49,435	0.68	16
Wh1b		1,074		48,771	43,200	49,569	49,854	0.57	30	8,216	6,624	20,629	49,854	0.57	6
Wh1c		1,084		46,140	43,200	46,941	47,283	0.73	35	5,824	2,754	16,707	47,283	0.73	10
Wh2	484	1,132		43,251	43,200	44,201	44,486	0.64	30	6,486	3,365	17,484	44,486	0.64	10
Wh2a		1,131		45,188	43,200	46,119	46,444	0.70	37	5,440	3,270	17,260	46,444	0.70	11
Wh2b		1,124		45,789	43,200	46,752	47,111	0.77	34	5,954	3,322	16,566	47,111	0.77	12
Wh2c		1,126		45,461	43,200	46,528	46,800	0.58	32	7,288	3,953	16,962	46,800	0.58	14
Wh1	463	1,073	120 km	44,142	43,200	45,718	46,086	0.80	33	11,998	19,202	33,953	46,082	0.80	31
Wh1a		1,087		48,303	43,200	49,777	50,264	0.98	37	8,239	13,495	34,274	50,259	0.97	32
Wh1b		1,074		48,772	43,200	50,332	50,692	0.72	32	12,769	25,417	44,935	50,692	0.72	8
Wh1c		1,084		46,140	43,200	47,675	48,031	0.75	26	7,843	5,972	21,268	48,031	0.75	14
Wh2	484	1,132		43,251	43,200	44,988	45,353	0.81	31	9,248	23,824	41,624	45,348	0.80	34
Wh2a		1,131		45,191	43,200	46,911	47,243	0.71	28	12,675	18,079	35,518	47,243	0.71	9
Wh2b		1,124		45,790	43,200	47,595	47,990	0.83	35	15,340	35,436	58,272	47,985	0.82	33
Wh2c		1,126		45,641	43,200	47,335	47,722	0.82	36	9,024	9,297	27,974	47,699	0.77	81

Note. CPLEX did not find an integer solution for any of these instances.

longer computational time. Because the solutions of the truncated column generation have already been near-optimal, the local search achieves only small improvements (see column 14), although its computational time is very short.

Table 7 presents the computational results on the large-sized instances with 947 trips and more than 1,100 time-expanded station nodes. The column definition is the same as in Table 6. We ran CPLEX (CPXlpopt) for 12 hours to solve the linear programming relaxation for an instance (W with 150 km). However, even the linear programming relaxation was not solved to optimality by CPLEX. We then did not continue the branch-and-bound process. As shown in Table 7, when the maximum route distance is 150 km, the truncated column generation obtained good solutions: the optimality gap is less than 5%. The maximum route distance has a great impact on the performance: the optimality gap the truncated column generation achieved is increased by a percentage between 7% and 15% when the maximum route distance is 120 km. The local search made small improvements based on the solutions from the truncated column generation. In the current implementation, the optimality gap of the linear relaxation at the root node is often about 10%; we stop the column generation process when a given time limit is reached. If we spend more time in the column generation process at the root node, the lower bound is expected to increase, thereby reducing the overall optimality gap. However, we have to balance the improved solution quality and the increased computational time. For these large-sized instances, the total computational time ranges from 58 to 86 hours. The long computational time is not surprising, considering the difficulty of these problems and the solution quality we obtained. Observe that the truncated column generation spent a large amount of time in solving the restricted master problems. An effective and simple method to reduce the computational time is to

Table 7 The Truncated Column Generation for Large-Sized Instances

				ARC-E by CPLEX		Truncated column generation							Local search imp		
Inst	Trip	Time node	Max dist	Lower bound	CPU	Lower bound	Obj	Gap (%)	CG iter	Root CPU	RMP CPU	Total CPU	Obj	Gap (%)	CPU
W	947	1,132	150 km	#	#	77,072	78,725	2.14	54	33,653	65,434	212,551	78,713	2.13	135
Wa		1,136		#	#	82,509	84,444	2.35	65	29,205	80,586	250,074	84,418	2.31	134
Wb		1,136		#	#	79,971	83,325	4.19	66	30,378	72,369	224,600	83,305	4.17	211
Wc		1,130		#	#	82,435	84,647	2.68	63	27,977	74,956	229,566	84,494	2.50	172
W	947	1,132	120 km	#	#	74,543	81,994	10.00	63	28,728	69,521	308,028	81,889	9.85	426
Wa		1,136		#	#	76,560	87,695	14.54	69	32,520	131,761	293,536	87,442	14.21	417
Wb		1,136		#	#	78,349	84,200	7.47	64	26,539	113,679	261,233	84,157	7.41	195
Wc		1,130		#	#	76,236	86,465	13.42	66	31,718	118,288	293,202	86,427	13.37	96

Note. #: CPLEX (CPXlpopt) failed to solve the linear relaxation in 12 hours, so we did not continue the branch and bound.

use the parallel processing functionality in the linear programming solver.

5.2.2. CNG, Diesel, and Hybrid Bus Scheduling with the Distance Constraint. In this section, we discuss the results on the scheduling problem with the maximum route distance constraint. Table 8 presents the computational results on Model ARC-NE by CPLEX. Columns 1 and 2 provide the problem instance and the number of trips. Columns 3–6 report the results when CNG buses are used: the lower bound, the objective value, the optimality gap that is defined as (objective value − lower bound)/lower bound, and the CPU seconds. Columns 7–14 report the corresponding results for diesel buses and hybrid buses. We set the maximum running time of CPLEX as four hours.

Table 8 shows that CPLEX performs well for the vehicle-scheduling problem with the maximum route distance. For the instances with less than 300 trips (Sh1, Sh2, S, and their invariants), CPLEX obtained an optimal solution instantly. For the instances with trips ranging from 300 to 500 (Wh1, Wh2, and their invariants), CPLEX also found optimal or near-optimal solutions within an acceptable amount of time. CPLEX experienced the problem of insufficient computer memory when solving the instances with more than 900 trips; it failed to find an integer solution for CNG and diesel buses (see the last four rows). For the instances with hybrid buses, CPLEX obtained integer solutions for three of four instances, but was still terminated because of insufficient memory in two instances. The maximum route distance has a notable impact on the performance of CPLEX: the instances with a shorter distance require much more computational time (see columns 6 and 14).

It is unlikely for the branch and price or the truncated column generation to outperform CPLEX for small-sized instances (Sh1, Sh2, S, and their invariants) because the pricing subproblem needs to be solved repeatedly, and there is the tailing-off effect in column generation. We then make comparisons for the medium- and large-sized instances. The comparisons for the medium-sized instances are shown in Table 9, where columns 6 and 7 are the objective value and the optimality gap, and the definitions of other columns are the same as in Table 6. Table 9 shows that the truncated column generation obtained near-optimal solutions for all these medium-sized instances: the optimality gap is very small. The local search then made small improvements and found optimal solutions for two instances (Wh2b for CNG and Wh2a for hybrid). In particular, the truncated column generation obtained much better solutions for four instances (Wh1b, Wh2, and Wh2a for CNG, and Wh2a for diesel), where CPLEX generated a large optimality gap. Regarding computational time, CPLEX requires

Table 8 CPLEX Results on Bus Scheduling with the Maximum Distance Constraint (ARC-NE)

		CNG (238 km)				Diesel (256 km)				Hybrid (335 km)			
Inst	Trip	Lower bound	Obj	Gap (%)	CPU	Lower bound	Obj	Gap (%)	CPU (s)	Lower bound	Obj	Gap (%)	CPU
Sh1	117	17,068	17,068	0.00	1	19,996	19,996	0.00	1	18,345	18,345	0.00	1
Sh1a		19,250	19,250	0.00	1	22,299	22,299	0.00	1	20,562	20,562	0.00	1
Sh1b		18,480	18,480	0.00	1	21,569	21,569	0.00	1	19,823	19,823	0.00	1
Sh1c		18,507	18,507	0.00	1	21,630	21,630	0.00	1	19,849	19,849	0.00	1
Sh2	125	15,307	15,307	0.00	1	18,604	18,604	0.00	1	16,722	16,722	0.00	3
Sh2a		19,381	19,381	0.00	2	22,960	22,960	0.00	1	20,928	20,928	0.00	1
Sh2b		18,402	18,402	0.00	1	22,043	22,043	0.00	2	19,974	19,974	0.00	2
Sh2c		18,486	18,486	0.00	1	22,075	22,075	0.00	2	20,043	20,043	0.00	1
S	242	24,638	24,638	0.00	23	30,103	30,103	0.00	43	26,974	26,974	0.00	20
Sa		28,362	28,362	0.00	10	34,386	34,386	0.00	4	30,951	30,951	0.00	11
Sb		27,836	27,836	0.00	25	33,752	33,752	0.00	9	30,366	30,366	0.00	9
Sc		27,947	27,947	0.00	15	33,809	33,809	0.00	12	30,449	30,449	0.00	11
Wh1	463	66,446	66,446	0.00	5,435	82,199	82,199	0.00	2,371	73,192	73,192	0.00	141
Wh1a		71,733	71,733	0.00	2,274	88,215	88,215	0.00	172	78,809	78,809	0.00	222
Wh1b		72,368	73,073	0.97	14,400	89,247	89,247	0.00	751	79,619	79,619	0.00	1,741
Wh1c		69,268	69,268	0.00	4,595	85,885	85,885	0.00	1,210	76,394	76,394	0.00	223
Wh2	484	66,535	67,585	1.58	14,400	83,561	83,561	0.00	5,765	73,831	73,831	0.00	3,887
Wh2a		68,813	71,249	3.54	14,400	85,881	87,293	1.64	14,400	76,116	76,116	0.00	566
Wh2b		70,303	70,303	0.00	5,939	87,969	88,317	0.40	14,400	77,884	77,884	0.00	551
Wh2c		69,416	69,416	0.00	6,849	86,448	86,448	0.00	3,365	76,708	76,708	0.00	199
W	947	119,033	*	*	*	149,982	*	*	*	132,274	133,219	0.71	*
Wa		125,349	*	*	*	157,498	*	*	*	139,149	139,149	0.00	3,846
Wb		123,887	*	*	*	155,803	*	*	*	137,577	138,888	0.95	*
Wc		125,321	*	*	*	157,755	*	*	*	139,238	*	*	*

Note. *: CPLEX was terminated because it was out of memory.

Table 9 **Truncated Column Generation Heuristic for Medium-Sized Instances**

				ARC-NE by CPLEX				Truncated column generation							Local search imp		
Bus type	Inst	Trip	Max dist	Lower bound	Obj	Gap (%)	CPU	Lower bound	Obj	Gap (%)	It	Root CPU	RMP CPU	Total CPU	Obj	Gap (%)	CPU
CNG	Wh1	463	238	66,446	66,446	0.00	5,435	66,154	66,447	1.50E–03	5	2,704	1,968	3,201	66,447	1.50E–03	1
	Wh1a			71,733	71,733	0.00	2,274	71,442	71,734	1.39E–03	12	2,379	2,372	4,397	71,734	1.39E–03	1
	Wh1b			72,368	73,073	0.97	14,400	72,096	72,374	8.29E–03	12	2,190	1,554	2,873	72,372	5.53E–03	1
	Wh1c			69,268	69,268	0.00	4,595	68,963	69,269	1.44E–03	21	2,171	1,690	3,435	69,269	1.44E–03	1
	Wh2	484		66,535	67,585	1.58	14,400	65,976	66,539	6.01E–03	8	2,567	1,999	3,375	66,539	6.01E–03	1
	Wh2a			68,813	71,249	3.54	14,400	68,564	68,818	7.27E–03	20	2,176	1,831	3,197	68,817	5.81E–03	1
	Wh2b			70,303	70,303	0.00	5,939	69,972	70,303	0.00	24	3,080	2,470	4,729	70,303	0.00	1
	Wh2c			69,416	69,416	0.00	6,849	69,131	69,417	1.44E–03	17	2,668	2,477	4,340	69,417	1.44E–03	1
Diesel	Wh1	463	256	82,199	82,199	0.00	2,371	81,827	82,205	7.30E–03	10	1,897	1,346	2,645	82,204	6.08E–03	1
	Wh1a			88,215	88,215	0.00	172	87,851	88,234	2.15E–02	16	2,423	1,879	3,102	88,226	1.25E–02	1
	Wh1b			89,247	89,247	0.00	751	88,924	89,253	6.72E–03	8	2,262	1,686	3,002	89,250	3.36E–03	1
	Wh1c			85,885	85,885	0.00	1,210	85,594	85,888	3.49E–03	13	1,883	1,113	2,891	85,888	3.49E–03	1
	Wh2	484		83,561	83,561	0.00	5,765	83,220	83,572	1.32E–02	18	3,055	2,273	4,340	83,571	1.20E–02	1
	Wh2a			85,881	87,293	1.64	14,400	85,381	85,889	9.32E–03	23	2,246	2,029	3,723	85,887	6.99E–03	1
	Wh2b			87,969	88,317	0.40	14,400	87,631	87,978	1.02E–02	11	2,629	1,557	3,593	87,974	5.68E–03	1
	Wh2c			86,448	86,448	0.00	3,365	86,131	86,451	3.47E–03	13	2,513	1,829	3,343	86,450	2.31E–03	1
Hybrid	Wh1	463	335	73,192	73,192	0.00	141	72,897	73,197	6.83E–03	12	2,332	1,735	3,072	73,197	6.83E–03	1
	Wh1a			78,809	78,809	0.00	222	78,482	78,813	5.08E–03	13	2,038	1,344	2,627	78,810	1.27E–03	1
	Wh1b			79,619	79,619	0.00	1,741	79,305	79,645	3.27E–02	17	2,850	2,616	4,500	79,620	1.26E–03	1
	Wh1c			76,394	76,394	0.00	223	76,070	76,402	1.05E–02	13	2,143	2,362	4,692	76,398	5.24E–03	1
	Wh2	484		73,831	73,831	0.00	3,887	73,607	73,834	4.06E–03	13	2,543	2,049	3,199	73,833	2.71E–03	1
	Wh2a			76,116	76,116	0.00	566	75,787	76,117	1.31E–03	15	2,583	2,146	3,365	76,116	0.00	1
	Wh2b			77,884	77,884	0.00	551	77,555	77,887	3.85E–03	14	2,879	2,258	3,999	77,886	2.57E–03	1
	Wh2c			76,708	76,708	0.00	199	76,354	76,711	3.91E–03	19	2,639	2,156	3,600	76,710	2.61E–03	1

Note. Bold indicates CPLEX failed to get an optimal solution.

much shorter time for the instances with hybrid buses, whereas the truncated column generation performs faster for the instances with CNG buses.

Table 10 reports the computational results on the large-sized instances with 947 trips, where CPLEX experienced the problem of insufficient memory. The column definition is the same as in Table 9. Table 10 shows that the truncated column generation performed very well for most of these instances. The worst cases occur at instances W and Wc for CNG buses with the optimality gaps as 1.00% and 1.18%, respectively, whereas near-optimal solutions are obtained for other instances. The total computational time is long, from 8 to 20 hours for most of the instances. The worst case happens at instance Wc with hybrid buses, where the computational time is more than 32 hours. Note that CPLEX failed to find an integer solution for this instance, but it successfully obtained near-optimal solutions for the other three instances with hybrid buses. Therefore, instance Wc

Table 10 **Truncated Column Generation Heuristic for Large-Sized Instances**

				ARC-NE by CPLEX				Truncated column generation							Local search imp		
Bus type	Inst	Trip	Max dist	Lower bound	Obj	Gap (%)	CPU	Lower bound	Obj	Gap (%)	It	Root CPU	RMP CPU	Total CPU	Obj	Gap (%)	CPU
CNG	W	947	238	119,033	*	*	*	117,263	120,223	1.00E+00	54	8,552	32,385	48,897	119,667	5.33E–01	1
	Wa			125,349	*	*	*	123,461	125,387	3.03E–02	55	14,057	55,391	71,719	125,356	5.58E–03	1
	Wb			123,887	*	*	*	122,112	123,913	2.10E–02	46	10,228	27,312	40,725	123,895	6.46E–03	1
	Wc			123,887	*	*	*	123,816	125,349	1.18E+00	61	10,579	38,278	62,587	125,328	1.16E+00	1
Diesel	W	947	256	149,982	*	*	*	148,528	150,012	2.00E–02	32	12,524	18,782	31,771	149,996	9.33E–03	1
	Wa			157,498	*	*	*	154,125	157,524	1.65E–02	54	13,704	25,082	39,017	157,516	1.14E–02	3
	Wb			155,803	*	*	*	152,892	155,808	3.21E–03	45	14,434	26,338	39,712	155,805	1.28E–03	2
	Wc			157,755	*	*	*	155,196	157,769	8.87E–03	42	15,472	19,191	34,302	157,763	5.07E–03	1
Hybrid	W	947	335	132,274	133,219	0.71	*	130,773	132,277	2.27E–03	45	13,846	24,323	36,064	132,276	1.51E–03	2
	Wa			139,149	139,149	0.00	3,846	136,794	139,224	5.39E–02	55	12,546	34,376	47,237	139,166	1.22E–02	1
	Wb			137,577	138,888	0.95	*	135,295	137,580	2.18E–03	53	12,148	17,691	31,869	137,578	7.27E–04	2
	Wc			139,238	*	*	*	137,478	139,261	1.65E–02	62	9,946	40,976	117,409	139,248	7.18E–03	2

Note. $*$: CPLEX was terminated because it was out of memory.

Table 11 Operation, Energy, and Emission Analysis for CNG, Diesel, Hybrid, and Electric Buses

Inst	Trips	Bus type	No. of bus	Veh cost	Waiting cost	Deadhead (%) / Trip dist	Dist cost	No. of swap	Swap cost	Total cost	Emissions NO_x (g)	Emissions PM (g)	Emissions CO_2 (kg)
Sh1	117	CNG-238	11	1,100	194.6	53.48	412.2	0	0	1,706.8	1,864.1	15.1	1,430.9
		Diesel-256	11	1,100	210.4	49.07	689.2	0	0	1,999.6	18,295.6	204.0	1,784.9
		Hybrid-335	11	1,100	208.3	49.56	526.2	0	0	1,834.5	2,647.9	10.9	1,195.4
		Elec-150	11	1,100	100.3	148.31	106.3	1	5	1,311.6	0.0	0.0	0.0
		Elec-120	11	1,100	94.9	151.18	107.9	4	20	1,322.8	0.0	0.0	0.0
Sh2	125	CNG-238	9	900	170.4	39.97	460.3	0	0	1,530.7	2,073.5	16.8	1,591.6
		Diesel-256	9	900	174.1	39.31	786.3	0	0	1,860.4	20,854.6	232.5	2,034.6
		Hybrid-335	9	900	170.5	40.00	601.7	0	0	1,672.2	3,023.2	12.5	1,364.8
		Elec-150	9	900	93.6	100.98	104.6	3	15	1,113.2	0.0	0.0	0.0
		Elec-120	9	900	86.8	103.40	105.3	6	30	1,122.1	0.0	0.0	0.0
S	242	CNG-238	15	1,500	201.4	27.83	762.4	0	0	2,463.8	3,446.1	27.9	2,645.3
		Diesel-256	15	1,500	212.0	26.39	1,298.3	0	0	3,010.3	34,434.8	383.9	3,359.5
		Hybrid-335	15	1,500	207.0	26.96	990.4	0	0	2,697.4	4,989.5	20.6	2,252.5
		Elec-150	15	1,500	117.8	62.48	149.8	2	10	1,777.6	0.0	0.0	0.0
		Elec-120	15	1,500	121.4	59.28	146.0	7	35	1,802.4	0.0	0.0	0.0
Wh1	463	CNG-238	34	3,400	1,031.4	28.38	2,213.2	0	0	6,644.6	9,954.1	80.7	7,640.8
		Diesel-256	34	3,400	1,035.3	28.19	3,784.6	0	0	8,219.9	100,442.0	1,119.7	9,799.2
		Hybrid-335	34	3,400	1,032.1	28.34	2,887.1	0	0	7,319.2	14,505.5	59.8	6,548.4
		Elec-150	34	3,400	265.2	140.06	662.3	39	195	4,522.5	0.0	0.0	0.0
		Elec-120	34	3,400	266.4	137.49	661.3	56	280	4,607.7	0.0	0.0	0.0
Wh2	484	CNG-238	34	3,400	859.5	33.13	2,394.4	0	0	6,653.9	10,762.2	87.2	8,261.1
		Diesel-256	34	3,400	874.2	32.55	4,081.9	0	0	8,356.1	108,284.7	1,207.1	10,564.3
		Hybrid-335	34	3,400	862.8	32.97	3,120.3	0	0	7,383.1	15,669.6	64.6	7,074.0
		Elec-150	34	3,400	221.4	123.30	652.2	35	175	4,448.6	0.0	0.0	0.0
		Elec-120	34	3,400	233.7	119.47	635.4	52	260	4,529.1	0.0	0.0	0.0
W	947	CNG-238	61	6,100	1,507.0	23.82	4,359.7	0	0	11,966.7	19,609.6	158.9	15,052.4
		Diesel-256	60	6,000	1,559.6	23.39	7,440.0	0	0	14,999.6	197,478.5	2,201.4	19,266.1
		Hybrid-335	60	6,000	1,557.9	23.42	5,669.7	0	0	13,227.6	28,493.6	117.4	12,863.3
		Elec-150	60	6,000	319.6	112.15	1,201.7	70	350	7,871.3	0.0	0.0	0.0
		Elec-120	61	6,100	549.3	94.39	1,094.6	89	445	8,188.9	0.0	0.0	0.0

with hybrid buses may be very difficult to solve. The local search made small improvements in short computational time.

5.3. Economic and Environmental Analysis

We are now ready to present total operational costs and bus emissions when CNG, diesel, hybrid, or electric buses are used in service. In Table 11, columns 1–3 provide the problem instance, the power type, and the number of trips. Columns 4 and 5 report the number of vehicles in operation and the vehicle maintenance costs. Column 6 shows the waiting costs. Column 7 presents the percentage between the total deadheading distance and the total trip distance, using the total trip distance as a basis. Note that the total trip distance is a constant for a specific problem instance. Column 8 shows the total distance costs. Columns 9 and 10 report the times of battery swapping and the corresponding costs. Column 11 presents the overall costs. Columns 12, 13, and 14 report the amounts of NO_x, PM, and CO_2, respectively, emitted by buses. The analysis presented in Table 11 is based on the best solution we have obtained for each instance in §5.2.

We first discuss the number of buses needed. There are no noticeable differences among CNG, diesel, and hybrid buses for all the problem instances, which shows that the maximum operational distance of 238 km may be long enough. Even if this distance is increased, it may not be significantly helpful for reducing the number of vehicles. The use of electric buses does not yield a notable increase in the number of vehicles needed. This may be because of the characteristics of the timetabled trips used in the case studies. Note that the waiting time with CNG, diesel, and hybrid buses is much longer than the waiting time with electric buses, which shows that the timetabled trips may not be efficiently designed.

We now discuss the traveled distance, the distance costs, and the total operational costs. Electric buses generate a much longer deadheading distance than CNG, diesel, and hybrid buses (see column 7). The longer deadheading distance is expected because electric buses need to travel toward the battery station to obtain a fully-charged battery. However, the distance costs and the total operational costs of electric buses are smaller than the ones of CNG, diesel, and hybrid buses because the average electricity price is much lower than the average fuel price. The total costs of CNG buses are smaller than the costs of hybrid buses and diesel buses because of the lower price of CNG. The total costs of hybrid buses are smaller than the costs of diesel buses because of the higher fuel economy of hybrid buses. The total costs of electric buses are significantly smaller than the total costs of CNG, diesel, and hybrid buses for all the instances because of the lower electricity price. In addition, the mild decrease of the maximum route distance, from 150 km to 120 km, does not have a notable impact on the total operational costs of electric buses.

We then discuss emissions generated by transit buses. Electric buses are certainly the best: zero emissions. As expected, diesel buses generate much more emissions than CNG and hybrid buses. CNG buses produce less NO_x than hybrid buses, whereas hybrid buses emit less PM and CO_2. In this study, our objective is to investigate the amounts of bus emissions rather than examining the trade-off among the vehicle costs, the operational costs, and the bus emissions. Interested readers are referred to Li and Head (2009), where the reduction of bus emissions for a mixed fleet of transit buses with different power sources is investigated.

Finally, although electric buses with the battery swapping result in significantly smaller total operational costs and fewer emissions, it cannot be concluded that the use of electric buses is a better choice for the economic and social purposes. First, the use of electric buses requires building at least one battery service station for battery swapping or charging, which is an expensive infrastructure investment. The use of CNG buses also needs depot modification (Clark et al. 2007). Second, we did not include vehicle purchase costs. As shown in Port and Atkinson (2006), the costs of a 40-foot low-floor hybrid transit bus are about $414,000, approximately $90,000 more than a CNG bus and $113,000 more than a conventional bus in 2004. The price of an electric bus may be even higher: Barry (2010) reports that the 35-foot electric bus used in Los Angeles costs about $1 million. In the real world, the bus purchase is a complicated process and depends on the policies in specific countries. For example, in the United States, there is a suggestion called "Minimum Life-Cycle Cost Replacement Ages and Mileage" by Federal Transit Administration (Laver et al. 2007). It is not trivial to incorporate such requirements into the vehicle-scheduling model. We then concentrate on minimizing the total operational costs in this paper. Finally, we did not monetize the emissions. As argued by Levy (2003), many of the uncertainties in estimating social damage costs by emissions have not been answered by the current literature.

6. Conclusion and Future Research

In this paper, we propose a vehicle-scheduling model for electric transit buses with battery swapping or fast charging at a battery station, and a vehicle-scheduling model with the maximum route distance constraint for CNG, diesel, or hybrid buses. We develop a branch-and-price algorithm and its heuristic version with a truncated column generation and a local search method. We use real-world instances and instances randomly generated in a practical setting to examine the performance of the developed algorithms. We run the branch-and-price algorithm on small-sized instances and run the truncated column generation and the local search on medium- and large-sized instances. Our algorithms are compared with the MIP solver of CPLEX 11.2. We conduct extensive computational studies.

We first summarize the computational results on electric bus scheduling with battery swapping. CPLEX obtains near-optimal solutions for only a few instances with fewer than 100 trips. If the problem size becomes larger, CPLEX generally fails to obtain an integral solution. For the large-sized instances with more than 900 trips, CPLEX fails to solve the linear relaxation to optimality in 12 hours. In comparison, our algorithms perform very well: (1) for small-sized instances, the branch-and-price algorithm obtains optimal or near-optimal solutions quickly; (2) for medium-sized instances, the truncated column generation finds near-optimal solutions; and (3) for large-sized instances, the truncated column generation generates very good solutions with the optimality gap less than 5% when the maximum route distance is 150 km; the optimality gap is between 7% and 15% if the maximum route distance is 120 km. The computational time of the branch and price and truncated column generation is generally in units of hours for medium- and large-sized instances. The local search makes small improvements based on the solutions from the truncated column generation.

We then discuss the results on transit bus scheduling with the maximum route distance constraint for CNG, diesel, or hybrid buses. CPLEX finds optimal or near-optimal solutions for small- and medium-sized instances quickly. However, CPLEX experiences the problem of insufficient computer memory in solving the instances with more than 900 trips and cannot find an integer solution for CNG or diesel buses. The truncated column generation obtains near-optimal solutions for all these medium- and large-sized instances with a very small optimality gap. For large-sized instances, CPLEX requires much shorter time for the instances of hybrid buses, whereas the truncated column generation performs faster for the instances of CNG buses.

In our case studies, the use of electric buses with battery swapping does not yield a notable increase on the number of vehicles needed. Electric buses generate much more deadheading distances than CNG, diesel, and hybrid buses, because electric buses need to travel toward a battery station for exchanging with a fully-charged battery. However, the total operational costs of electric buses with battery swapping are generally much smaller than the costs of other buses because the average electricity price is much lower than the average fuel price. The use of CNG buses results in the lower operational costs than diesel and hybrid buses because of the lower CNG price. It is worth mentioning that these findings are based on the timetabled trips used in our case studies, and therefore may not be generalized into other cases. Electric buses generate no emissions, whereas diesel buses produce more emissions than CNG and hybrid buses, in particular NO_x and PM. Finally, although electric buses with battery swapping result in smaller total operational costs and significantly fewer emissions, it cannot be concluded that the use of electric buses is a better choice for the economic and social purposes, because of the requirement of additional capital investments, such as vehicle purchase and battery station construction. The goal of our studies is to provide some useful information for decision making.

Future research can go in several directions. A natural extension is to investigate the models and algorithms when there are multiple depots. An important topic is to study the location problem of battery service stations. Improving the algorithm efficiency for large-sized problems with the shorter maximum-route distance is another direction.

Acknowledgments

The author thanks the editor, the guest editor, and the anonymous referees for pointing out the use of column generation as the solution approach, and valuable suggestions on the paper presentation. The author would also like to thank Andreas Klose at Aarhus University for explaining the use of subgradient search in column generation.

References

Ahuja RK, Magnanti TL, Orlin JB (1993) *Network Flows: Theory, Algorithms, and Applications* (Prentice Hall, Englewood Cliffs, NJ).

Argonne National Laboratory (1997) Electric buses energize downtown Chattanooga. Accessed December 5, 2011, http://www.afdc.energy.gov/afdc/pdfs/chatt_cs.pdf.

Ball M, Bodin L, Dial R (1983) A matching based heuristic for scheduling mass transit crews and vehicles. *Transportation Sci.* 17(1):4–31.

Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH (1998) Branch-and-price: Column generation for solving huge integer programs. *Oper. Res.* 46(3):316–329.

Barry K (2010) Quick-charge electric bus rolls into L.A. County. Accessed January 25, 2012, http://www.wired.com/autopia/2010/09/proterra-ecoride-foothill-transit.

Bertossi AA, Carraresi P, Gallo G (1987) On some matching problems arising in vehicle scheduling models. *Networks* 17(3):271–281.

Bianco L, Mingozzi A, Ricciardelli S (1994) A set partitioning approach to the multiple depot vehicle scheduling problem. *Optim. Method. Software* 3(1–3):163–194.

Bodin L, Golden B, Assad A, Ball M (1983) Routing and scheduling of vehicles and crews: The state of the art. *Comput. Oper. Res.* 10(2):63–211.

Bräysy O, Gendreau M (2005) Vehicle routing problem with time windows, part I: Route construction and local search algorithms. *Transportation Sci.* 39(1):104–118.

Bunte S, Kliewer N (2009) An overview on vehicle scheduling models. *Public Transportation* 1(4):299–317.

Carpaneto G, Dell'Amico M, Fischetti M, Toth P (1989) A branch and bound algorithm for the multiple depot vehicle scheduling problem. *Networks* 19(5):531–548.

Chang C (2010) Business models and public policies for EV charging stations in China. Accessed December 5, 2011, http://www.changce.org/attachments/559_evstation_businessmodel_changce_20101018.pdf.

Clark NN, Zhen F, Wayne WS, Lyons DW (2007) Transit bus life cycle cost and year 2007 emissions estimation. Technical report FTA-WV-26-7004.2007.1, National Renewable Energy Laboratory, Washington, DC.

Cohen JT, Hammitt JK, Levy JI (2003) Fuels for urban transit buses: A cost-effectiveness analysis. *Environ. Sci. Tech.* 37(8):1477–1484.

Dell'Amico M, Righini G, Salani M (2006) A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transportation Sci.* 40(2):235–247.

Desaulniers G, Hickman MD (2007) Public transit. Barnhart C, Laporte B, eds. *Handbooks in Operations Research and Management Science, Transportation* (North-Holland, Amsterdam), 69–127.

du Merle O, Villeneuve D, Desrosiers J, Hansen P (1999) Stabilized column generation. *Discrete Math.* 194(1–3):229–237.

Dugan T (1994) Electric buses in operation: The Chattanooga experience. *Transport. Res. Record* 1444:3–9.

Dunbar M, Froyland G, Wu C-L (2012) Robust airline schedule planning: Minimizing propagated delay in an integrated routing and crewing framework. *Transportation Sci.* 46(2):204–216.

Elkins J, Hemby J, Rao V, Szykman J, Fitz-Simons T, Doll D, Mintz D, et al. (2003) National air quality and emission trends report 2003 special studies edition. Technical report EPA-454/R-03-008, U.S. Environmental Protection Agency, Research Triangle Park, NC.

Energy Information Administration (2011) Prices and factors affecting prices. Accessed December 5, 2011, http://www.eia.gov/energyexplained/index.cfm?page=electricity_factors_affecting_prices.

Environment Canada (2004) Orion VII transit bus equipped with BAE systems HybriDrive propulsion system emissions and fuel economy. Technical Report ERMD-2004-18, Environmental Technology Advancement Directorate, Environment Canada, Ontario, Canada.

EV World (2008) Beijing readies electric buses for Summer Olympics. Accessed December 5, 2011, http://www.evworld.com/news.cfm?newsid=18730.

Farley AA (1990) A note on bounding a class of linear programming problems, including cutting stock problems. *Oper. Res.* 38(5):922–923.

Feillet D, Dejax P, Gendreau M, Gueguen C (2004) An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks* 44(3):216–229.

Forbes MA, Holt JN, Watts AM (1994) An exact algorithm for multiple depot bus scheduling. *Eur. J. Oper. Res.* 72(1):115–124.

Freling R (1997) Models and techniques for integrating vehicle and crew scheduling. Ph.D. dissertation, Erasmus University, Rotterdam, The Netherlands.

Freling R, Paixão JMP (1995) Vehicle scheduling with time constraint. Daduna J, Branco I, Paixão JMP, eds. *Computer-Aided Transit Scheduling. Lecture Notes in Economics and Mathematical Systems*, Vol. 430 (Springer, Berlin), 130–144.

Freling R, Huisman D, Wagelmans APM (2001a) Applying an integrated approach to vehicle and crew scheduling in practice. Voß S, Daduna J, eds. *Computer-Aided Transit Scheduling. Lecture Notes in Economics and Mathematical Systems*, Vol. 505 (Springer, Berlin), 73–90.

Freling R, Wagelmans APM, Paixão JMP (2001b) Models and algorithms for single-depot vehicle scheduling. *Transportation Sci.* 35(2):165–180.

Friberg C, Haase K (1999) An exact branch and cut algorithm for the vehicle and crew scheduling problem. Wilson NHM, ed. *Computer-Aided Transit Scheduling. Lecture Notes in Economics and Mathematical Systems*, Vol. 471 (Springer, Berlin), 63–80.

Gamache M, Soumis F, Marquis G, Desrosiers J (1999) A column generation approach for large-scale aircrew rostering problems. *Oper. Res.* 47(2):247–263.

Garey MR, Johnson DS (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman, San Francisco).

Griffith P, Gleason G (1996) Six years of battery-electric bus operation at the Santa Barbara Metropolitan Transit District. *13th Internat. Electric Vehicle Sympos., Osaka, Japan*.

Haase K, Desaulniers G, Desrosiers J (2001) Simultaneous vehicle and crew scheduling in urban mass transit systems. *Transportation Sci.* 35(3):286–303.

Hadjar A, Marcotte O, Soumis F (2006) A branch-and-cut algorithm for the multiple depot vehicle scheduling problem. *Oper. Res.* 54(1):130–149.

Haggis S, Beback A (2010) Transit bus technology feasibility study. Accessed December 5, 2011, http://www.advancedenergy.org/transportation/additional_efforts/TransitBusFeasibilityStudy_Final.pdf.

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

Huisman D, Freling R, Wagelmans APM (2005) Multiple-depot integrated vehicle and crew scheduling. *Transportation Sci.* 39(4):491–502.

Ibaraki T, Imahori S, Kubo M, Masuda T, Uno T, Yagiura M (2005) Effective local search algorithms for routing and scheduling problems with general time-window constraints. *Transportation Sci.* 39(2):206–232.

Kim L (2011) Namsan electric bus cruises into international limelight. Accessed December 5, 2011, http://www.cnngo.com/seoul/life/namsan-electric-bus-cruises-limelight-262862.

Kliewer N, Gintner V, Suhl L (2008) Line change considerations within a time-space network based multi-depot bus scheduling model. Hickman M, Mirchandani P, Voß S, eds. *Computer-Aided Transit Scheduling. Lecture Notes in Economics and Mathematical Systems*, Vol. 600 (Springer, Berlin), 57–70.

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

Klose A, Drexl A (2005) Lower bounds for the capacitated facility location problem based on column generation. *Management Sci.* 51(11):1689–1705.

Klose A, Görtz S (2007) A branch-and-price algorithm for the capacitated facility location problem. *Eur. J. Oper. Res.* 179(3):1109–1125.

Laver R, Schneck D, Skorupski D, Brady S, Cham L (2007) Useful life of transit buses and vans. Technical report FTA VA-26-7229-07.1, Federal Transit Administration, Washington, DC.

Levy JI (2003) Issues and uncertainties in estimating the health benefits of air pollution control. *J. Toxicol. Environ. Health Part A* 66(16–19):1865–1871.

Li JQ (2012) Match bus stops to a digital road network by the shortest path model. *Transportation Res. C-Emer.* 22:119–131.

Li JQ, Head KL (2009) Sustainability provisions in the bus scheduling problem. *Transportation Res. D-Tr. E* 14(1):50–60.

Lin S (1965) Computer solutions of the traveling salesman problem. *AT&T Tech. J.* 44(10):2245–2269.

Löbel A (1998) Vehicle scheduling in public transit and Lagrangean pricing. *Management Sci.* 44(12):1637–1649.

Lübbecke ME, Desrosiers J (2005) Selected topics in column generation. *Oper. Res.* 53(6):1007–1023.

Lusby R, Larsen J, Ryan D, Ehrgott M (2011) Routing trains through railway junctions: A new set-packing approach. *Transportation Sci.* 45(2):228–245.

National Transit Database (2013) Annual national transit summary and trends (NTST). Accessed January 10, 2013, http://www.ntdprogram.gov/antdprogram/data.htm.

Or I (1976) Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking. Ph.D. dissertation, Northwestern University, Evanston, IL.

Paixão JM, Branco I (1987) A quasi-assignment algorithm for bus scheduling. *Networks* 17(3):249–269.

Pepin A-S, Desaulniers G, Hertz A, Huisman D (2009) A comparison of five heuristics for the multiple depot vehicle scheduling problem. *J. Scheduling* 12(1):17–30.

Pessoa A, Uchoa E, Poggi de Aragão M, Rodrigues R (2010) Exact algorithm over an arc-time-indexed formulation for parallel machine scheduling problems. *Math. Program. Comp.* 2(3–4):259–290.

Petersen JD, Sölveling G, Clarke J-P, Johnson EL, Shebalov S (2012) An optimization approach to airline integrated recovery. *Transportation Sci.* 46(4):482–500.

Port D, Atkinson J (2006) Bus futures 2006. Report, INFORM, New York. http://www.informinc.org/busfut06.php.

Potvin J-Y, Kervahut T, Garcia B-L, Rousseau J-M (1996) The vehicle routing problem with time windows part I: Tabu search. *INFORMS J. Comput.* 8(2):158–164.

Prescott-Gagnon E, Desaulniers G, Drexl M, Rousseau L-M (2010) European driver rules in vehicle routing with time windows. *Transportation Sci.* 44(4):455–473.

Ribeiro CC, Soumis F (1994) A column generation approach to the multiple-depot vehicle scheduling problem. *Oper. Res.* 42(1):41–52.

Righini G, Salani M (2006) Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optim.* 3(3):255–273.

Rousseau JM, Gendreau M, Feillet D (2007) Interior point stabilization for column generation. *Oper. Res. Lett.* 35(5):660–668.

Ryan DM, Foster BA (1981) An integer programming approach to scheduling. Wren A, ed. *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling* (North-Holland, Amsterdam), 269–280.

Saddoune M, Desaulniers G, Elhallaoui I, Soumis F (2012) Integrated airline crew pairing and crew assignment by dynamic constraint aggregation. *Transportation Sci.* 46(1):39–55.

Sandhu R, Klabjan D (2007) Integrated airline fleeting and crew-pairing decisions. *Oper. Res.* 55(3):439–456.

Shen Y, Xia J (2009) Integrated bus transit scheduling for the Beijing bus group based on a unified mode of operation. *Internat. Trans. Oper. Res.* 16(2):227–242.

Song T, Zhou L (1990) A new algorithm for the quasi-assignment problem. *Ann. Oper. Res.* 24(1):205–223.

Stasko TH, Gao OH (2010) Reducing transit fleet emissions through vehicle retrofits, replacements, and usage changes over multiple time periods. *Transport. Res. D-Tr. E.* 15(5):254–262.

Steinzen I, Gintner V, Suhl L, Kliewer N (2010) A time-space network approach for the integrated vehicle- and crew-scheduling problem with multiple depots. *Transportation Sci.* 44(3):367–382.

Taillard É, Badeau P, Gendreau M, Guertin F, Potvin J-Y (1997) A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Sci.* 31(2):170–186.

Tang L, Wang G, Liu J, Liu J (2011) A combination of Lagrangian relaxation and column generation for order batching in steelmaking and continuous-casting production. *Naval Res. Logist.* 58(4):370–388.

United Nations Environment Programme (2009) UNEP environmental assessment Expo 2010 Shanghai, China. Technical report DCP/1209/NA, United Nations Environment Programme, Nairobi, Kenya.

Vanderbeck F, Wolsey LA (1996) An exact algorithm for IP column generation. *Oper. Res. Lett.* 19(4):151–160.

Wang H, Shen J (2007) Heuristic approaches for solving transit vehicle scheduling problem with route and fueling time constraints. *Appl. Math. Comput.* 190(2):1237–1249.

Wentges P (1997) Weighted Dantzig-Wolfe decomposition for linear mixed-integer programming. *Internat. Trans. Oper. Res.* 4(2):151–162.