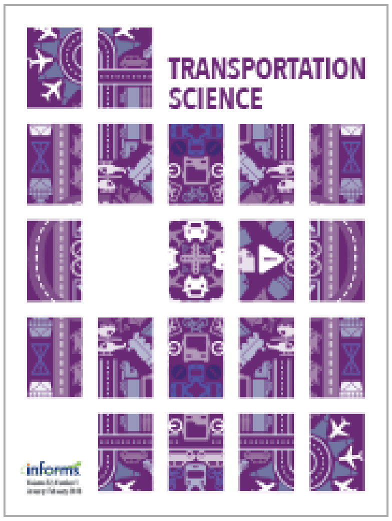

Transportation Science

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

A Two-Phase Branch-and-Price-and-Cut for a Dial-a-Ride Problem in Patient Transportation

Zhixing Luo, Mengyang Liu, Andrew Lim

informs
http://pubsonline.informs.org/journal/trsc/

TRANSPORTATION SCIENCE
Vol. 53, No. 1, January–February 2019, pp. 113–130
ISSN 0041-1655 (print), ISSN 1526-5447 (online)

A Two-Phase Branch-and-Price-and-Cut for a Dial-a-Ride Problem in Patient Transportation

Zhixing Luo,[a,b] Mengyang Liu,[c] Andrew Lim[a,b]

[a] School of Management and Engineering, Nanjing University, 210093 Nanjing, China; [b] Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore 119077; [c] Department of Management Sciences, City University of Hong Kong, Kowloon Tong, Hong Kong

Contact: luozx.hkphd@gmail.com, http://orcid.org/0000-0002-0491-1501 (ZL); liu.mengyang.hust@gmail.com (ML); alim.china@gmail.com, http://orcid.org/0000-0003-0510-8080 (AL)

Received: November 5, 2015
Revised: September 9, 2016; March 6, 2017
Accepted: March 15, 2017
Published Online in Articles in Advance: August 1, 2018

https://doi.org/10.1287/trsc.2017.0772

Abstract. In this paper, we investigate an extension of the R-DARP recently proposed by [Liu M, Luo Z, Lim A (2015) A branch-and-cut algorithm for a realistic dial-a-ride problem. *Transporation Res. Part B: Methodological* 81(1):267–288.]. The R-DARP, as a variant of the classic dial-a-ride problem (DARP), consists of determining a set of minimum-distance trips for vehicles to transport a set of clients from their origins to their destinations, subject to side constraints, such as capacity constraints, time window constraints, maximum riding time constraints, and manpower planning constraints. Our problem extends the R-DARP by (1) changing its objective to first maximizing the number of requests satisfied and then minimizing the total travel distance of the vehicles, and (2) generalizing the lunch break constraints of staff members. To solve this problem, we propose a two-phase branch-and-price-and-cut algorithm based on a strong trip-based model. The trip-based model is built on a set of nondominated trips, which are enumerated by an ad hoc label-setting algorithm in the first phase. Then we decompose the trip-based model by Benders decomposition and propose a branch-and-price-and-cut algorithm to solve the decomposed model in the second phase. Our two-phase exact algorithm is tested on the R-DARP benchmark instances and a set of new instances generated according to the same real-world data set as the R-DARP instances. Our algorithm quickly solves all of the R-DARP instances to optimality and outperforms the branch-and-cut proposed by Liu, Luo, and Lim. On the 42 new test instances, our algorithm solves 27 instances to optimality in four hours with the largest instance consisting of 36 requests.

History: This paper has been accepted for the *Transportation Science* Special Issue on Recent Advances in Urban Transportation Through Optimization and Analytics.
Funding: This research was partially supported by the National Natural Science Foundation of China [Grants 71501091, 71571077, and 71531009] and the MOE Academic Research Fund of Singapore [Grants R266000096133, R266000096731, and MOE2017-T2-2-153].
Supplemental Material: The online appendix is available at https://doi.org/10.1287/trsc.2017.0772.

Keywords: dial-a-ride problem • manpower planning • branch and price and cut • patient transportation

1. Introduction

The *dial-a-ride problem* (DARP), generalizes from the classic *pickup and delivery problem* (PDP). It consists of deciding a set of minimum-cost routes for a fleet of vehicles to satisfy a set of transportation requests, each of which involves transporting a client from an origin point to a destination point, subject to side constraints, such as the capacity constraint of the vehicles, the time window, and the maximum riding time constraints of the clients. One of the major applications of the DARP is the patient transportation services provided by hospitals and other medical institutions (Madsen, Ravn, and Rygaard 1995; Toth and Vigo 1996, 1997; Melachrinoudis, Ilhan, and Min 2007). This application involves scheduling a fleet of ambulances to transport elderly people or patients from their home to the hospital and vice versa.

In the past decade, more and more practical constraints and features motivated by real-world applications have been considered in patient transportation problems. Parragh (2011) and Parragh et al. (2012), motivated by observations made by the Austrian Red Cross in the field of patient transportation, studied a variant of the PDP in which clients make requests for multiple types of facilities, such as seats, wheelchairs, and stretchers, and the vehicles have different capacities for different types of facilities. Qu and Bard (2013, 2014) investigated a PDP variant in which the vehicles have multiple capacity configurations and can adjust the configurations according to the facility requests of the destined clients. The problem arises from a real application associated with daily route planning for the All-Inclusive Care Program for the Elderly. Lim, Zhang, and Qin (2017) studied a multi-trip variant of the PDP in which a team of assistants is deciding on vehicles to accomplish the transportation tasks. This problem originates from the Non-Emergency Ambulance Transfer Service (NEATS) in Hong Kong, which is a free public service offered by the local government. In this problem, clients require different kinds of facilities, such as seats, wheelchairs, and stretchers. Because some of the clients live in old buildings without elevators, they can also require a certain number of assistants to help them to get on or off the vehicle. Thus, a sufficient condition for a vehicle to serve a client is that the number of assistants on board the vehicle is greater than or equal to that required by the client. Recently, Liu, Luo, and Lim (2015) introduced a more practical variant of the DARP, referred to as *R-DARP*, which simultaneously considers all of the new features mentioned here, namely multiple trips, heterogeneous vehicles, multiple request types, configurable vehicle capacity, and manpower planning. They proposed a branch-and-cut algorithm to solve the problem exactly. Compared with other PDP and DARP variants, the R-DARP is more applicable to patient transportation.

Although the R-DARP addresses many new features and constraints that have arisen in patient transportation in the past decade, there are still some limitations to the R-DARP. First, all clients must be served, and the objective of the R-DARP is to minimize the total travel distance of the vehicles. However, in practice, resources such as vehicles, drivers, and assistants are usually limited, and it can be difficult to satisfy all of the clients (Lim, Zhang, and Qin 2017). Therefore, to make the model more applicable, we relax the constraint that all of the clients must be satisfied and change the objective to first maximizing the number of clients satisfied and then minimizing the total travel distance of the vehicles. Second, the lunch time window of the staff is defined as the period in which the staff must stay in the depot to have lunch. In this way, the planning horizon of each staff member is divided into two separate parts by the lunch time window. However, in the literature, the lunch time window of a staff, similar to the ordinary service time windows of customers, is usually defined as the time interval in which the staff can start to have lunch (Savelsbergh and Sol 1998). In our problem, we also follow such lunch time windows. This extension of the R-DARP is referred to as the *ER-DARP*.

In this paper, we propose a two-phase exact algorithm to solve the ER-DARP. The exact algorithm relies on a strong trip-based model built on a graph, in which each node represents a nondominated trip. A trip is a nondominated trip if no other trip can replace it in any feasible solution, resulting in a better solution. Therefore, there must exist an optimal solution consisting of nondominated trips. To enumerate the set of nondominated trips, we propose a label-setting algorithm to handle the configurable capacity constraint and the maximum riding time constraint. After the nondominated trips are enumerated in the first phase, a branch-and-price-and-cut algorithm is proposed to solve the trip-based model in the second phase. In the branch-and-price-and-cut algorithm, a pricing problem is solved to find a workday with minimum reduced cost for each vehicle and assistant. We show that the pricing problem is indeed an elementary shortest path problem with resource constraints (ESSPRC) and, thus, can be solved by a label-setting algorithm. The branch-and-price-and-cut algorithm involves two families of valid inequalities. The first family of valid inequalities are infeasible path elimination inequalities, which are similar to the classic infeasible path elimination inequalities for routing problems (Ascheuer, Fischetti, and Grötschel 2000). The second family of valid inequalities are Benders cuts derived from the manpower planning problem. That is, for a vehicle scheduling solution, if there exists no feasible scheduling for the assistants with respect to this vehicle scheduling solution, a Benders cut can be derived to cut off this vehicle scheduling solution. The two-phase algorithm is tested on the R-DARP benchmark instances proposed by Liu, Luo, and Lim (2015) and a set of new instances generated according to the real-world data of the NEATS. Computational experiments show that the two-phase algorithm outperforms the branch-and-cut algorithm proposed by Liu, Luo, and Lim (2015) on the R-DARP instances and can optimally solve new instances with up to 36 requests in four hours.

The remainder of the paper is organized as follows. In Section 2, we briefly review the exact algorithms for the DARP. In Section 3, we present a trip-based model for the ER-DARP. The solution approaches, including Benders decomposition, the label-setting algorithm to enumerate the set of nondominated trips, and the branch-and-price-and-cut algorithm to solve the trip-based model, are presented in Section 4. Section 5 is devoted to the computational experiments. Finally, Section 6 concludes the paper with some closing remarks.

2. Literature Review

A survey on the DARP can be found in Cordeau and Laporte (2007). Early studies of exact algorithms for the DARP concentrated on dynamic programming to solve the single vehicle DARP. Psaraftis (1980, 1983) proposed two dynamic programming approaches, which resembled the dynamic programming used to solve the traveling salesman problem (TSP). Desrosiers, Dumas, and Soumis (1986) proposed another dynamic programming approach to solve the single-vehicle DARP with precedence, capacity, and time window constraints. This approach was able to solve instances with up to 40 requests but required tight capacity and time window constraints.

The multi-vehicle DARP was first exactly solved by a branch-and-cut algorithm proposed by Cordeau (2006), which relied on a three-index mixed integer programming (MIP) model and several families of valid inequalities derived from the well-known inequalities for the TSP and the vehicle routing problem (VRP). Then, based on a two-index MIP model and some well-known valid inequalities, Ropke, Cordeau, and Laporte (2007) proposed another branch-and-cut algorithm to solve the pickup and delivery problem with time windows (PDPTW) as well as the DARP. After that, Parragh (2011) developed a branch-and-cut algorithm to solve a variant of the DARP that considers heterogeneous clients and vehicles. Later, Parragh et al. (2012) proposed a branch-and-price algorithm to solve a variant of the DARP in which the clients and the vehicles are heterogeneous and the maximum riding time constraints of the clients are implicitly enforced by artificially constructing time windows for certain nodes. However, this implementation cannot truly enforce the maximum riding constraints because the riding time of a client may still exceed the given limit. Recently, Braekers, Caris, and Janssens (2014) devised a branch-and-cut algorithm to solve a DARP variant with multiple depots and heterogeneous vehicles, and Gschwind and Irnich (2015) proposed a branch-and-price-and-cut algorithm that can handle dynamic time window constraints for routing problems and successfully applied it to solve the DARP.

3. Problem Formulation

3.1. Notation

The ER-DARP is defined on a complete directed graph $G=(V,A)$, where $V=\{0,\ldots,2n\}=\{0\}\cup P\cup D$ is the node set and $A=\{(i,j)\mid i,j\in V, i\neq j\}$ is the arc set. Node 0 is the depot. Node sets $P=\{1,\ldots,n\}$ and $D=\{n+1,\ldots,2n\}$ are the set of pickup nodes and the set of delivery nodes, respectively. A request consists of transporting a client from a pickup node $i\in P$ to a delivery node $n+i\in D$. For simplicity, we also use P to represent the set of clients. Each arc $(i,j)\in A$ has a travel time $t_{i,j}$ (in minutes). Each node $i\in P\cup D$ has demands q_i^c, where $c=1,2,3$ represents seats, wheelchairs, and stretchers, respectively. Here, we set $q_i^c=-q_{n+i}^c\geq 0$ for all $i\in P$ and $c=1,2,3$. Each node $i\in P\cup D$ also has a service time s_i and a service time window $[e_i,l_i]$. The time window of the nodes in $P\cup D$ is the time interval in which the service should start. If a vehicle arrives at a node $i\in P\cup D$ before e_i, the vehicle must wait until e_i to start the service. For each client $i\in P$, let f_i be the number of assistants that client i requires to help her to get on the vehicle, and g_i be the maximum riding time of client i. Each assistant on the vehicle requires a seat. Let Ψ_V be the set of vehicles, Ψ_A be the set of assistants, and $\Psi=\Psi_V\cup\Psi_A$. We assume each vehicle is associated with a fixed driver. Thus, we also use Ψ_V to represent the set of drivers. In each vehicle ψ ($\psi\in\Psi_V$), there are Q_ψ^f fixed seats and Q_ψ^d configurable divisions. Each division can be configured to hold three seats, two wheelchairs, or one stretcher at the start of each trip. At the end of each trip, the staff on the vehicle must have a break for t^B minutes. Moreover, each staff member ψ ($\psi\in\Psi$) has a work time window $[e_\psi,l_\psi]$. Staff member ψ cannot start work before e_ψ and cannot work after l_ψ. Each staff member ψ also has a lunch time window $[e_\psi^{\text{lunch}},l_\psi^{\text{lunch}}]$ within which the staff member must start to have lunch at the depot. The lunch break lasts for t^{lunch} minutes. We assume $t^{\text{lunch}}>t^B$. The maximum duration of each trip is T. Online Appendix D gives a list of the notations used in this paper.

Let α ($\alpha>0$) be the profit collected by a vehicle after a request is satisfied. The objective of the ER-DARP is to design a set of workdays that minimizes the total travel distance of the vehicles minus the total profits collected by the vehicles. When α is sufficiently large, the problem aims to first maximize the number of requests satisfied and then minimize the total travel distance of vehicles. However, if α is too large, the problem may suffer from numerical difficulty. Online Appendix B presents an approach to determine a suitable value for α.

3.2. Trip-Based Model

In this section, we propose a trip-based model for the ER-DARP. The backbone of the trip-based model is the set of nondominated feasible trips, which start from and end at the depot; serve some clients; and satisfy the pairing, the precedence, the assistant requirement, the maximum riding time constraints of the clients, and the time window constraints of the visited nodes as well as the maximum duration constraint. We refer the reader to Online Appendix A for the compact arc-based model. Before we present the model, we first introduce some necessary definitions. Let $K=\{1,\ldots,m\}$ be a set of consecutive positive numbers that are used to index the trips of the staff. Let $e_0=\min_{\psi\in\Psi}e_\psi$ and $l_0=\max_{\psi\in\Psi}l_\psi$. We assign time window $[e_0,l_0]$ to depot 0. Furthermore, we assign a maximum riding time, that is, g_0, to the staff and set $g_0=T$.

Definition 1. A trip $r=(i_0=0,i_1,\ldots,i_m=0)$ is feasible for vehicle $\psi\in\Psi_V$ if

1. It visits a node at most once and satisfies the pairing and precedence constraints.
2. There exists a configuration b^c ($c=1,2,3$) and o assistants on the trip such that

$$\sum_{k=1}^{j} q_{i_k}^1 + o \leq Q_\psi^f + 3b^1,\quad \forall j=1,\ldots,m-1,$$

$$\sum_{k=1}^{j} q_{i_k}^2 \leq 2b^2,\quad \forall j=1,\ldots,m-1,$$

$$\sum_{k=1}^{j} q^3_{i_k} \le b^3, \quad \forall\, j = 1, \ldots, m-1,$$
$$o \ge f_{i_j}, \quad \forall\, j = 1, \ldots, m-1,$$
$$b^1 + b^2 + b^3 \le Q^d_\psi.$$

3. There exists a service start time a_{i_j} for each node i_j $(j = 0, \ldots, m)$ such that

$$e_{i_j} \le a_{i_j} \le l_{i_j}, \quad \forall\, j = 0, \ldots, m,$$
$$a_{i_m} - a_{i_0} \le T,$$
$$a_{i_j} \le a_{i_j+n} \le a_{i_j} + s_{i_j} + g_{i_j}, \quad \forall\, i_j \in P.$$

For a feasible trip r, let V_r denote the set of clients served in r and C_r, L_r, and E_r denote the cost, the latest start time, and the earliest completion time of r, respectively. The latest start time is the latest departure time from the depot to ensure the feasibility of r. Similarly, the earliest completion time is the earliest time to return to the depot for a vehicle after finishing the requests. Let $X_r(t)$ be the earliest completion time of r given that the start time of r is t $(t \le L_r)$ and $Y_r(t)$ be the latest start time of r given that the completion time of r is t $(t \ge E_r)$.

Definition 2. Given two feasible trips r_1 and r_2 for the same vehicle, r_1 dominates r_2 if

$$V_{r_1} = V_{r_2}, \tag{1}$$
$$C_{r_1} \le C_{r_2}, \tag{2}$$
$$L_{r_1} \ge L_{r_2}, \tag{3}$$
$$X_{r_1}(t) \le X_{r_2}(t), \quad \forall\, t \le L_{r_2}. \tag{4}$$

If trip r_2 is dominated by trip r_1, we can replace r_2 with r_1 in any feasible solution that contains r_2. Another equivalent definition for the domination relationship between two trips is as follows:

Definition 3. Given two feasible trips r_1 and r_2 for the same vehicle, r_1 dominates r_2 if

$$V_{r_1} = V_{r_2}, \tag{5}$$
$$C_{r_1} \le C_{r_2}, \tag{6}$$
$$E_{r_1} \le E_{r_2}, \tag{7}$$
$$Y_{r_1}(t) \ge Y_{r_2}(t), \quad \forall\, t \ge E_{r_2}. \tag{8}$$

According to the definition of nondominated trips, there must exist an optimal solution consisting of nondominated trips. Therefore, it is sufficient to consider the set of nondominated trips, denoted as R_N, when modeling the ER-DARP.

Definition 4. A path $\phi = (r_1, \ldots, r_m)$, $r_i \in R_N$, $i = 1, \ldots, m$ consisting of feasible trips is feasible if there exists a start time a_{r_i} $(i = 1, \ldots, m)$ for each trip r_i, such that

$$X_{r_i}(a_{r_i}) + t^B \le a_{r_{i+1}} \le L_{r_{i+1}}, \quad \forall\, i = 1, \ldots, m-1.$$

Let Φ denote the set of infeasible paths consisting of feasible nondominated trips. A feasible solution to the ER-DARP cannot contain an infeasible path. Therefore, we can implement the synchronization between the vehicles and the assistants by eliminating the infeasible paths.

To deal with the heterogenous lunch time windows and the work periods of the staff, we create pseudo-trips r^l_ψ, r^s_ψ, and r^e_ψ for each staff member $\psi \in \Psi$ to represent the lunch break, the first trip, and the last trip of ψ, respectively. Let $R_\Psi = \{r^l_\psi, r^s_\psi, r^e_\psi \mid \psi \in \Psi\}$, and $R = R_N \cup R_\Psi$. For each $\psi \in \Psi$, the attributes of the trips in R_Ψ are listed in Table 1.

Note that, for simplicity, we assume that a staff must take a break for t^B minutes after a trip, including the pseudo-trips. Therefore, for r^l_ψ, $E_{r^l_\psi} + t^B$ indicates the earliest time to finish lunch, which is equal to $e^{\text{lunch}}_\psi + t^{\text{lunch}}$, and $L_{r^l_\psi} - t^B$ indicates the latest start time to have lunch, which is equal to l^{lunch}_ψ. Furthermore, given the start time of r^l_ψ is t, the actual time to start having lunch is $t - t^B$ because before r^l_ψ, the staff already has had a break for t^B minutes. So $X_{r^l_\psi}(t) + t^B$, which indicates the time to finish lunch given the start time of r^l_ψ is t, is equal to $\max\{t - t^B + t^{\text{lunch}}, e^{\text{lunch}}_\psi + t^{\text{lunch}}\}$. Similarly, given the completion time of r^l_ψ is t, the actual time to finish lunch is $t + t^B$ because after r^l_ψ, the staff must take a break for t^B minutes. So $Y_{r^l_\psi}(t) - t^B$, which gives the time to start having lunch given the completion time of r^l_ψ is t, is equal to $\min\{t + t^B - t^{\text{lunch}}, l^{\text{lunch}}_\psi\}$. Using the same rationales, we can easily derive the attributes of r^s_ψ and r^e_ψ.

Definition 5. A workday $\omega = (r_1, \ldots, r_m)$, $r_i \in R$, $i = 1, \ldots, m$ consisting of feasible trips for $\psi \in \Psi$ is a feasible workday for staff member $\psi \in \Psi$ if $r_1 = r^s_\psi$, $r_m = r^e_\psi$, $r^l_\psi \in \omega$, and ω is a feasible path.

Next, we present the trip-based model. For a trip $r \in R$, let F_r denote the minimum number of assistants required to satisfy the requests in r, that is, $F_r = \max_{i \in V_r} f_i$. For each staff member $\psi \in \Psi$, let Ω_ψ be the

Table 1. Attributes of Trip $r \in R_\Psi$

	V_r	C_r	E_r	L_r	$X_r(t)$	$Y_r(t)$
r^l_ψ	$\varnothing$	0	$e^{\text{lunch}}_\psi + t^{\text{lunch}} - t^B$	$l^{\text{lunch}}_\psi + t^B$	$\max\{t - t^B + t^{\text{lunch}}, e^{\text{lunch}}_\psi + t^{\text{lunch}}\} - t^B$	$\min\{t + t^B - t^{\text{lunch}}, l^{\text{lunch}}_\psi\} + t^B$
r^s_ψ	$\varnothing$	0	$e_\psi - t^B$	$+\infty$	$\max\{t, e_\psi\} - t^B$	$t + t^B$
r^e_ψ	$\varnothing$	0	$-t^B$	$l_\psi + t^B$	$t - t^B$	$\min\{t, l_\psi\} + t^B$

set of feasible workdays, which consist of trips in R. Let $\Omega = \bigcup_{\psi\in\Psi}\Omega_\psi$. For a workday $\omega\in\Omega$, let c_ω denote the cost of ω, $\beta_{i,\omega}$ $(i\in P)$ be a binary number equal to one if and only if client i is served in ω, $\gamma_{r,\omega}$ $(r\in R)$ be a binary number equal to one if and only if trip r is used in ω, and $\rho_{r_1,r_2,\omega}$ $(r_1,r_2\in R)$ be a binary number equal to one if and only if two consecutive trips r_1 and r_2 are included in ω. The trip-based model involves two types of decision variables. Let θ_ω $(\omega\in\Omega)$ be a binary decision variable equal to one if and only if workday ω is selected, and λ_{r_1,r_2} $(r_1,r_2\in R)$ be a binary variable equal to one if and only if trip r_1 is used immediately after trip r_2 by the same staff member. The trip-based model is formulated as follows:

$$(\mathbf{I})\quad \min \sum_{\psi\in\Psi_V}\sum_{\omega\in\Omega_\psi} c_\omega\theta_\omega \tag{9}$$

s.t.

$$\sum_{\psi\in\Psi_V}\sum_{\omega\in\Omega_\psi}\beta_{i,\omega}\theta_\omega \le 1, \quad \forall\, i\in P, \tag{10}$$

$$\sum_{\psi\in\Psi_V}\sum_{\omega\in\Omega_\psi}F_r\gamma_{r,\omega}\theta_\omega \le \sum_{\psi\in\Psi_A}\sum_{\omega\in\Omega_\psi}\gamma_{r,\omega}\theta_\omega, \forall\, r\in R_N, \tag{11}$$

$$\sum_{(r_1,r_2)\in\phi}\lambda_{r_1,r_2} \le |\phi|-1, \quad \forall\,\phi\in\Phi, \tag{12}$$

$$\sum_{\psi\in\Psi_V}\sum_{\omega\in\Omega_\psi}\rho_{r_1,r_2,\omega}\theta_\omega \le \lambda_{r_1,r_2}, \quad \forall\, r_1,r_2\in R, \tag{13}$$

$$\sum_{\omega\in\Omega_\psi}\rho_{r_1,r_2,\omega}\theta_\omega \le \lambda_{r_1,r_2}, \quad \forall\,\psi\in\Psi_A,\, r_1,r_2\in R, \tag{14}$$

$$\sum_{\omega\in\Omega_\psi}\theta_\omega \le 1, \quad \forall\,\psi\in\Psi_V, \tag{15}$$

$$\sum_{\omega\in\Omega_\psi}\theta_\omega \le 1, \quad \forall\,\psi\in\Psi_A, \tag{16}$$

$$\theta_\omega\in\{0,1\}, \quad \forall\,\omega\in\Omega_\psi,\,\psi\in\Psi_V, \tag{17}$$

$$\theta_\omega\in\{0,1\}, \quad \forall\,\omega\in\Omega_\psi,\,\psi\in\Psi_A, \tag{18}$$

$$\lambda_{r_1,r_2}\in\{0,1\}, \quad \forall\, r_1,r_2\in R. \tag{19}$$

The objective function (9) minimizes the solution cost. Constraints (10) ensure that each request can be satisfied at most once. Constraints (11) ensure that the number of assistants for each trip is sufficient to satisfy the requests in the trip. Constraints (12) are the infeasible path elimination constraint, which together with constraints (13) and (14), ensure that the synchronization between the vehicle workdays and the assistant workdays is feasible. Constraints (15) and (16) are the convexity constraints. If $\min\{F_{r_1},F_{r_2}\}=1$, constraints (14) can be aggregated as follows:

$$\sum_{\psi\in\Psi_A}\sum_{\omega\in\Omega_\psi}\rho_{r_1,r_2,\omega}\theta_\omega \le \lambda_{r_1,r_2}, \quad \forall\, r_1,r_2\in R. \tag{20}$$

Therefore, if $\min\{F_{r_1},F_{r_2}\}=1$, we use constraints (20) in the trip-based model; otherwise, we use constraints (14).

4. Methodology

In this section, we propose a two-phase approach to solve the ER-DARP. In the first phase, we apply a label-setting algorithm to enumerate the set of nondominated trips, and in the second phase, we use a branch-and-price-and-cut algorithm to achieve an optimal solution. The branch-and-price-and-cut algorithm relies on the lower bound provided by the linear programming (LP) relaxation of the trip-based model. Therefore, we first relax the integral constraints (17)–(19) to $\theta_\omega\ge 0$ $(\forall\,\omega\in\Omega_\psi)$, $\psi\in\Psi_V$, $\theta_\omega\ge 0$ $(\forall\,\omega\in\Omega_\psi)$, $\psi\in\Psi_A$, and $\lambda_{r_1,r_2}\ge 0$ $(\forall\, r_1,\, r_2\in R)$, respectively. Because the number of infeasible paths is exponential, they cannot be enumerated beforehand. Therefore, constraints (12)–(14) and (20) are ignored at first and then added to the model dynamically. The LP relaxation of Model (I) then becomes an LP involving an exponential number of columns, which can be solved by *column generation*. However, the LP relaxation is still difficult to solve provided that the number of nondominated trips is huge, resulting in a huge number of constraints (11). Thus, we apply Benders decomposition to transform the decision variables associated with the workdays of the assistants, that is, θ_ω $(\omega\in\Omega_\psi)$, $\psi\in\Psi_A$, into a set of Benders cuts so that these variables and constraints (11) can be removed from the LP relaxation. Similar to the infeasible path elimination constraints, the Benders cuts can also be added into the model dynamically. In summary, the core of the branch-and-price-and-cut algorithm is a column- and cut-generation procedure to solve the LP relaxation of the trip-based model. This technique has been successfully applied to solve the simultaneous aircraft routing and crew scheduling problem (Cordeau et al. 2001; Mercier, Cordeau, and Soumis 2005).

In the remainder of this section, we first introduce the Benders decomposition applied to the trip-based model. Then we generally describe the column- and cut-generation procedure to solve the LP-relaxed trip-based model. Finally, we present the details of the two-phase algorithm, including the label-setting algorithms used in both the first and second phase, and the branching strategies.

4.1. Benders Decomposition

Let $A_1=\{(r_1,r_2)\,|\,r_1,r_2\in R,\ r_1\ne r_2,\ \min\{F_{r_1},F_{r_2}\}=1\}$ and $A_2=\{(r_1,r_2)\,|\,r_1,\, r_2\in R,\ r_1\ne r_2,\ \min\{F_{r_1},F_{r_2}\}>1\}$. Given nonnegative values $\bar\theta_\omega$ $(\omega\in\Omega_\psi,\psi\in\Psi_V)$ and $\bar\lambda_{r_1,r_2}$ $(r_1,r_2\in R)$ satisfying constraints (10), (12), (13), and (15), the LP relaxation of the trip-based model can be reduced to the following *primal model* involving only θ_ω associated with the assistant workdays:

$$(\mathbf{II})\quad \min\ 0 \tag{21}$$

$$\text{s.t.}\ \sum_{\psi\in\Psi_A}\sum_{\omega\in\Omega_\psi}\gamma_{r,\omega}\theta_\omega \ge \sum_{\psi\in\Psi_V}\sum_{\omega\in\Omega_\psi}F_r\gamma_{r,\omega}\bar\theta_\omega, \quad \forall\, r\in R_N, \tag{22}$$

$$\sum_{\psi\in\Psi_A}\sum_{\omega\in\Omega_\psi}\rho_{r_1,r_2,\omega}\theta_\omega \le \bar{\lambda}_{r_1,r_2}, \quad \forall (r_1,r_2)\in A_1, \quad (23)$$

$$\sum_{\omega\in\Omega_\psi}\rho_{r_1,r_2,\omega}\theta_\omega \le \bar{\lambda}_{r_1,r_2}, \quad \forall \psi\in\Psi_A,(r_1,r_2)\in A_2, \quad (24)$$

(16), (18).

Let $\mu_r(r\in R_N)$, $\nu^1_{r_1,r_2}((r_1,r_2)\in A_1)$, $\nu^2_{\psi,r_1,r_2}(\psi\in\Psi_A,(r_1,r_2)\in A_2)$, and $\pi_\psi(\psi\in\Psi_A)$ denote the dual values of constraints (22)–(24) and (16), respectively. The dual problem of Model (II) is as follows:

$$\textbf{(III)}\quad \max\left\{\sum_{r\in R_N}\mu_r\sum_{\psi\in\Psi_V}\sum_{\omega\in\Omega_\psi}F_r\gamma_{r,\omega}\bar{\theta}_\omega + \sum_{(r_1,r_2)\in A_1}\nu^1_{r_1,r_2}\bar{\lambda}_{r_1,r_2} + \sum_{(r_1,r_2)\in A_2}\sum_{\psi\in\Psi_A}\nu^2_{\psi,r_1,r_2}\bar{\lambda}_{r_1,r_2} + \sum_{\psi\in\Psi_A}\pi_\psi\right\} \quad (25)$$

$$\text{s.t.}\quad \sum_{r\in R_N}\gamma_{r,\omega}\mu_r + \sum_{r_1\in R}\sum_{r_2\in R}\rho_{r_1,r_2,\omega}(\nu^1_{r_1,r_2}+\nu^2_{\psi,r_1,r_2}) + \pi_\psi \le 0, \quad \forall\omega\in\Omega_\psi,\psi\in\Psi_A, \quad (26)$$

$$\mu_r\ge 0, \quad \forall r\in R_N, \quad (27)$$

$$\nu^1_{r_1,r_2}\le 0, \quad \forall r_1,r_2\in A_1, \quad (28)$$

$$\nu^2_{\psi,r_1,r_2}\le 0, \quad \forall\psi\in\Psi_A, r_1,r_2\in A_2, \quad (29)$$

$$\pi_\psi\le 0, \quad \forall\psi\in\Psi_A. \quad (30)$$

Let Δ denote the polyhedron defined by constraints (26)–(30) and $\mathbb{R}_\Delta$ be the set of extreme rays of Δ. Then, the LP relaxation of the trip-based model (Model I) can be reformulated as the following *Benders master problem*:

$$\textbf{(IV)}\quad \min \sum_{\psi\in\Psi_V}\sum_{\omega\in\Omega_\psi}c_\omega\theta_\omega \quad (31)$$

$$\text{s.t.}\quad \sum_{r\in R_N}\sum_{\psi\in\Psi_V}\sum_{\omega\in\Omega_\psi}\mu_rF_r\gamma_{r,\omega}\theta_\omega + \sum_{(r_1,r_2)\in A_1}\nu^1_{r_1,r_2}\lambda_{r_1,r_2} + \sum_{(r_1,r_2)\in A_2}\sum_{\psi\in\Psi_A}\nu^2_{\psi,r_1,r_2}\lambda_{r_1,r_2} + \sum_{\psi\in\Psi_A}\pi_\psi \le 0, \quad \forall(\mu,\nu^1,\nu^2,\pi)\in\mathbb{R}_\Delta, \quad (32)$$

(10), (12), (13), (15),

$$0\le\theta_\omega\le 1, \quad \forall\omega\in\Omega_\psi,\psi\in\Psi_V, \quad (33)$$

$$0\le\lambda_{r_1,r_2}\le 1, \quad \forall r_1,r_2\in R. \quad (34)$$

The set of extreme rays $\mathbb{R}_\Delta$ can be identified by solving the primal model (Model II) with column generation. If the primal model is infeasible, then we identify an extreme ray; otherwise, the feasible solution provided by the primal model is the set of feasible workdays for the assistants.

4.1.1. Refinements of the Benders Decomposition.

As the efficiency of the Benders decomposition is heavily influenced by the qualities of the Benders cuts, to improve the efficiency of the Benders decomposition, we use the strong Benders cut (Magnanti and Wong 1981, Mercier, Cordeau, and Soumis 2005, de Sá, de Camargo, and de Miranda 2013).

Let M denote a sufficiently large number. We reformulate Model (II) as follows:

$$\textbf{(V)}\quad \min \sum_{r\in R_N}Mx_r \quad (35)$$

$$\text{s.t.}\quad \sum_{\psi\in\Psi_A}\sum_{\omega\in\Omega_\psi}\gamma_{r,\omega}\theta_\omega + x_r \ge \sum_{\psi\in\Psi_V}\sum_{\omega\in\Omega_\psi}F_r\gamma_{r,\omega}\bar{\theta}_\omega, \quad \forall r\in R_N, \quad (36)$$

$$x_r\ge 0, \quad \forall r\in R_N, \quad (37)$$

(16), (18), (23), (24).

Let P_v be the polyhedron defined by the Benders master problem (Model (IV)) and let $\text{ri}(P_v)$ denote the relative interior of P_v. Let $(\bar{\theta}^0_\omega,\bar{\lambda}^0_{r_1,r_2})$ represent a feasible solution in $\text{ri}(P_v)$. Let $\bar{V}$ denote the optimal value of Model (V). According to de Sá, de Camargo, and de Miranda (2013) and Mercier, Cordeau, and Soumis (2005), a strong Benders cut can be identified by solving the following model:

$$\textbf{(VI)}\quad \max \sum_{r\in R_N}\sum_{\psi\in\Psi_V}\sum_{\omega\in\Omega_\psi}\mu_rF_r\gamma_{r,\omega}\bar{\theta}^0_\omega + \sum_{(r_1,r_2)\in A_1}v^1_{r_1,r_2}\bar{\lambda}^0_{r_1,r_2} + \sum_{(r_1,r_2)\in A_2}\sum_{\psi\in\Psi_A}v^2_{\psi,r_1,r_2}\bar{\lambda}^0_{r_1,r_2} + \sum_{\psi\in\Psi_A}\pi_\psi \quad (38)$$

$$\text{s.t.}\quad \sum_{r\in R_N}\sum_{\psi\in\Psi_V}\sum_{\omega\in\Omega_\psi}\mu_rF_r\gamma_{r,\omega}\bar{\theta}_\omega + \sum_{(r_1,r_2)\in A_1}v^1_{r_1,r_2}\bar{\lambda}_{r_1,r_2} + \sum_{(r_1,r_2)\in A_2}\sum_{\psi\in\Psi_A}v^2_{\psi,r_1,r_2}\bar{\lambda}_{r_1,r_2} + \sum_{\psi\in\Psi_A}\pi_\psi = \bar{V}, \quad (39)$$

(26)–(30),

$$0\le\mu_r\le M, \quad \forall r\in R_N. \quad (40)$$

Instead of solving Model (VI) directly, we solve its dual form. Let y denote the dual variable of constraints (39). The dual problem of Model (VI) is as follows:

$$\textbf{(VII)}\quad \min \sum_{r\in R_N}Mz_r + \bar{V}y \quad (41)$$

$$\text{s.t.}\quad \sum_{\psi\in\Psi_A}\sum_{\omega\in\Omega_\psi}\gamma_{r,\omega}\theta_\omega + x_r + \sum_{\psi\in\Psi_V}\sum_{\omega\in\Omega_\psi}F_r\gamma_{r,\omega}\bar{\theta}_\omega y \ge \sum_{\psi\in\Psi_V}\sum_{\omega\in\Omega_\psi}F_r\gamma_{r,\omega}\bar{\theta}^0_\omega, \quad \forall r\in R_N, \quad (42)$$

$$\sum_{\psi\in\Psi_A}\sum_{\omega\in\Omega_\psi}\rho_{r_1,r_2,\omega}\theta_\omega + \bar{\lambda}_{r_1,r_2}y \le \bar{\lambda}^0_{r_1,r_2}, \quad \forall(r_1,r_2)\in A_1, \quad (43)$$

$$\sum_{\omega\in\Omega_\psi}\rho_{r_1,r_2,\omega}\theta_\omega + \bar{\lambda}_{r_1,r_2}y \le \bar{\lambda}^0_{r_1,r_2}, \quad \forall\psi\in\Psi_A,(r_1,r_2)\in A_2, \quad (44)$$

$$\sum_{\omega\in\Omega_\psi}\theta_\omega + y\le 1, \quad \forall\psi\in\Psi_A, \quad (45)$$

$$\theta_\omega\ge 0, \quad \forall\omega\in\Omega_\psi,\psi\in\Psi_A, \quad (46)$$

$$z_r\ge 0, \quad \forall r\in R_N. \quad (47)$$

It is difficult to find a feasible solution $(\bar{\theta}^0_\omega, \bar{\lambda}^0_{r_1,r_2})$ in $\mathrm{ri}(P_v)$ because the set of variables in Model (V) is not fully known. However, choosing a solution out of $\mathrm{ri}(P_v)$ does not prevent Model (VII) from generating a valid Benders cut for the Benders master problem since $(\bar{\theta}^0_\omega, \bar{\lambda}^0_{r_1,r_2})$ only modifies the objective function of Model (VII) although it might reduce the strength of the Benders cut. In our implementation, we follow the same method proposed by Mercier, Cordeau, and Soumis (2005) and fix $\sum_{\psi\in\Psi_V}\sum_{\omega\in\Omega_\psi} F_r\gamma_{r,\omega}\bar{\theta}^0_\omega$ and $\bar{\lambda}^0_{r_1,r_2}$ in Model (VII) to 0.1 in our experiments.

4.2. Column and Cut Generation Procedure

As the LP-relaxed Model (I) and Model (IV) are equivalent, we solve Model (IV) by a column- and cut-generation procedure instead of solving LP-relaxed Model (I) directly. Note that when the value of the term $\sum_{\psi\in\Psi_V}\sum_{\omega\in\Omega_\psi} F_r\gamma_{r,\omega}\bar{\theta}_\omega$ for trip r is zero, there is no need for constraints (22), (36), and (42) for trip r to be included in the subproblem. Similarly, when the value of $\bar{\lambda}_{r_1,r_2}$ is one, constraints (23), (24), (43), and (44) for (r_1,r_2) become redundant. Let $R^1_N = R_N \cap \{r \mid \sum_{\psi\in\Psi_V}\sum_{\omega\in\Omega_\psi} F_r\gamma_{r,\omega}\bar{\theta}_\omega > 0\}$, $A^0_1 = A_1 \cap \{(r_1,r_2) \mid \bar{\lambda}_{r_1,r_2} < 1\}$ and $A^0_2 = A_2 \cap \{(r_1,r_2) \mid \bar{\lambda}_{r_1,r_2} < 1\}$. In our implementation, we use R^1_N, A^0_1, and A^0_2 in place of R_N, A_1, and A_2 in Models (II)–(VII), respectively.

The details of the column- and cut-generation procedure are presented in Algorithm 1. The algorithm consists of a two-level nested while loop. In the inner loop (Lines 3–15), the algorithm iteratively solves Model (IV) to optimality by column generation and dynamical inclusion of the strong Benders cuts, which are separated through solving Model (V) and Model (VII) (Lines 6–10). Until no strong Benders cuts are found (Line 13), the algorithm jumps out of the inner loop and then searches for the violated infeasible path elimination constraints (Line 17). If any violated infeasible path elimination constraints are identified (Line 19), they are added to Model (VI), which is resolved again; otherwise, the algorithm terminates (Line 21). The core of the column generation in Algorithm 1 is the label-setting algorithms to solve the pricing problems, which are detailed in Section 4.3.

4.3. The Label-Setting Algorithms

In this section, we introduce two label-setting algorithms: one for enumerating the nondominated trips in the first phase and the other for solving the pricing problems in the column generation of the second phase. Label-setting algorithms are a widely used dynamic programming approach for solving shortest path problems. In label-setting algorithms, each label (or state) represents a partial path from the origin to a certain node. All of the feasible paths can be enumerated by a label-setting algorithm through propagating labels along the arcs of a graph. A label-setting algorithm consists of three key components: label definition, extension functions, and dominance rules. In the remainder of this section, we mainly focus on these three key components when we describe a label-setting algorithm. For the framework of a label-setting algorithm, please see Boland, Dethridge, and Dumitrescu (2006) and Righini and Salani (2008).

4.3.1. The Label-Setting Algorithm for Enumerating the Nondominated Trips.

The label-setting algorithm for enumerating the nondominated trips is defined on the graph $G=(V,A)$. Some arcs in A cannot exist in any feasible solution because of various side constraints and thus can be safely removed from the graph. We follow the arc elimination step in Cordeau (2006) to remove the infeasible arcs in A.

Label Definitions. Let $\mathbb{L}_p = (\mathscr{C}_p, \mathscr{A}_p, \mathscr{V}_p, \{\mathscr{Q}^c_p, \mathscr{U}^c_p\}_{c=1,2,3}, \mathscr{Q}^{1,\max}_p, \mathscr{O}^{\min}_p, \{ld^h_p(t), \mathscr{B}^h_p\}_{h\in\{0\}\cup\mathscr{V}_p\cap P \text{ and } h+n\notin\mathscr{V}_p})$ denote a label representing a partial trip $p=(i_1,\ldots,i_m)$ $(i_1=0)$, where

- $\mathscr{C}_p$ is the cost of p;
- $\mathscr{A}_p$ is the earliest time to arrive at node i_m;
- $\mathscr{V}_p$ is the set of nodes that have been visited in p;
- $\mathscr{Q}^c_p$ is the load of facility c at node i_m;
- $\mathscr{U}^c_p$ is the number of divisions configured to hold facility c;
- $\mathscr{Q}^{1,\max}_p$ is the maximum number of seats occupied by patients;
- $\mathscr{O}^{\min}_p$ is the minimum number of assistants needed by requests in p;
- $ld^h_p(t)$ $(h\in\{0\}\cup\mathscr{V}_p\cap P$ and $h+n\notin\mathscr{V}_p)$ is the latest time to deliver client h $(h>0)$ or the latest time to return to the depot $(h=0)$ given that the service start time at node i_m is t;
- $\mathscr{B}^h_p$ $(h\in\{0\}\cup\mathscr{V}_p\cap P$ and $h+n\notin\mathscr{V}_p)$ is the break point when function $ld^h_p(t)$ stops increasing.

Resources $\mathscr{Q}^c_p$ and $\mathscr{U}^c_p$ are designed to deal with the configurable capacity constraints. Resources $\mathscr{Q}^{1,\max}_p$ and $\mathscr{O}^{\min}_p$ are proposed to handle the assistant requirements of the clients. Resources $ld^h_p(t)$ and $\mathscr{B}^h_p$ were first proposed by Gschwind and Irnich (2015) to cope with the maximum riding time constraints of the clients in the label-setting algorithm for solving the pricing problem of the DARP. Note that $ld^h_p(t)$ is a linear piecewise function consisting of two pieces, one with positive slope and the other with zero slope. Therefore, $ld^h_p(t)$ can be represented by two end points of the first piece. For more detailed information about $ld^h_p(t)$ and $\mathscr{B}^h_p$, please see Gschwind and Irnich (2015).

Extensions. Initially, we create a label $\mathbb{L}_0 = (0, e_0, \varnothing, \{0,0,0,0,0,0\}, 0, 0, \{ld^0_0(t), \mathscr{B}^0_0\})$ associated with the depot where $ld^0_0(t) = \min\{l_0, t+T\}$ and $\mathscr{B}^0_0 = l_0 - T$. Given an arc $(i,j)\in A$ and a label $\mathbb{L}_p$ associated with node i,

Algorithm 1 (Column- and cut-generation procedure)

```
initialize Model (IV) with constraints (12)–(19) (if they exist).
while TRUE do
    while TRUE do
        solve Model (IV) by column generation, and obtain an optimal solution θ̄_ω (ω ∈ Ω_ψ, ψ ∈ Ψ_V)
            and λ̄_{r1,r2} (r1, r2 ∈ R) (if they exist).
        use θ̄_ω (ω ∈ Ω_ψ, ψ ∈ Ψ_V) and λ̄_{r1,r2} (r1, r2 ∈ R) to initialize Model (V).
        solve Model (V) by column generation.
        if the optimal value of Model (V) (V̄) is positive then
            use V̄ to construct Model (VII).
            solve Model (VII) by column generation.
            obtain an extreme ray (λ̄, μ̄, ν̄) and add constraints (32) to Model (IV).
        else
            obtain an optimal solution θ̄_ω (ω ∈ Ω_ψ, ψ ∈ Ψ_A) of Model (V).
            BREAK.
        end if
    end while
    restore λ̄ by θ̄_ω (ω ∈ Ω): (see below)
    search for violated infeasible path elimination constraints according to λ̄.
    if violated infeasible path elimination constraints are found then
        add corresponding constraints (12) to Model (IV).
    else
        BREAK.
    end if
end while.
```

$$\bar{\lambda}_{r_1,r_2} = \begin{cases} \max\left\{\sum_{\psi\in\Psi_V}\sum_{\omega\in\Omega_\psi}\rho_{r_1,r_2,\omega}\bar{\theta}_\omega, \sum_{\psi\in\Psi_A}\sum_{\omega\in\Omega_\psi}\rho_{r_1,r_2,\omega}\bar{\theta}_\omega\right\}, & \min\{F_{r_1},F_{r_2}\}=1,\\ \max\left\{\sum_{\psi\in\Psi_V}\sum_{\omega\in\Omega_\psi}\rho_{r_1,r_2,\omega}\bar{\theta}_\omega, \max_{\psi\in\Psi_A}\sum_{\omega\in\Omega_\psi}\rho_{r_1,r_2,\omega}\bar{\theta}_\omega\right\}, & \min\{F_{r_1},F_{r_2}\}>1.\end{cases}$$

$\mathbb{L}_p$ is extended to node j if it satisfies the following constraints:

$$j \notin \mathcal{V}_p, \tag{48}$$

$$j \in P \vee j \in D \wedge j-n \in \mathcal{V}_p \vee j = 0 \wedge \sum_{c=1}^{3} \mathcal{Q}_p^c = 0, \tag{49}$$

$$\mathcal{A}_p + s_i + t_{i,j} \le l_j. \tag{50}$$

Constraint (48) ensures that node j has not been visited by p. Constraint (49) ensures that node j is either a pickup node, a delivery node with its pickup node visited, or the depot with zero clients on the vehicle. Constraint (50) ensures that the time window of node j is satisfied. The capacity constraints, the maximum duration constraints, and the maximum riding time constraints are checked when the new label $\mathbb{L}_q = (\mathcal{C}_q, \mathcal{A}_q, \mathcal{V}_q, \{\mathcal{Q}_q^c, \mathcal{U}_q^c\}_{c=1,2,3}, \mathcal{Q}_q^{1,\max}, \mathcal{O}_q^{\min}, \{ld_q^h(t), \mathcal{B}_q^h\}_{h\in\{0\}\cup\mathcal{V}_q\cap P \text{ and } h+n\notin\mathcal{V}_q})$ is created. The resources of $\mathbb{L}_q$ are set as follows:

$$\mathcal{C}_q = \mathcal{C}_p + t_{i,j}, \tag{51}$$

$$\mathcal{A}_q = \max\{e_j, \mathcal{A}_p + s_i + t_{i,j}\}, \tag{52}$$

$$\mathcal{V}_q = \mathcal{V}_p \cup \{j\}, \tag{53}$$

$$\mathcal{O}_q^{\min} = \max\{\mathcal{O}_p^{\min}, f_j\}, \tag{54}$$

$$\mathcal{Q}_q^c = \mathcal{Q}_p^c + q_j^c, \quad c = 1,2,3, \tag{55}$$

$$\mathcal{Q}_q^{1,\max} = \max\{\mathcal{Q}_q^1, \mathcal{Q}_p^{1,\max}\}, \tag{56}$$

$$\mathcal{U}_q^1 = \max\{\mathcal{U}_p^1, \lceil(\mathcal{Q}_q^{1,\max} + \mathcal{O}_q^{\min} - Q_\psi^f)/3\rceil\}, \tag{57}$$

$$\mathcal{U}_q^2 = \max\{\mathcal{U}_p^2, \lceil\mathcal{Q}_q^2/2\rceil\}, \tag{58}$$

$$\mathcal{U}_q^3 = \max\{\mathcal{U}_p^3, \mathcal{Q}_q^3\}, \tag{59}$$

$$\mathcal{B}_q^h = \begin{cases} \min\{b_j, l_{j+n} - s_j - g_j\}, & h = j,\\ \max\{\mathcal{A}_q, \min\{\mathcal{B}_p^h + s_i + t_{i,j}, b_j\}\}, & \text{otherwise,}\end{cases} \tag{60}$$

$$ld_q^h(\mathcal{A}_q) = \begin{cases} \min\{\mathcal{A}_q + s_j + g_j, l_{j+n}\}, & h = j,\\ ld_p^h(\mathcal{A}_q) + \min\{\mathcal{A}_q - s_i - t_{i,j}, \mathcal{B}_p^h\} - \mathcal{A}_p, & \text{otherwise,}\end{cases} \tag{61}$$

$$ld_q^h(\mathcal{B}_q^h) = \begin{cases} \mathcal{B}_q^j + s_j + g_j, & h = j,\\ ld_p^j(\mathcal{B}_p^h) - \max\{0, \mathcal{B}_p^h + s_i + t_{i,j} - b_j\}, & \text{otherwise,}\end{cases} \tag{62}$$

where

$$b_j = \begin{cases} \min\left\{l_j, \min_{h\in\mathcal{V}_q\cap P, h+n\notin\mathcal{V}_q}(ld_p^h(\mathcal{B}_p^h) - s_j - t_{j,h+n})\right\}, & j \in P,\\ \min\left\{l_j, ld_p^{j-n}(\mathcal{B}_p^{j-n}), \min_{h\in\mathcal{V}_q\cap P, h+n\notin\mathcal{V}_q}(ld_p^h(\mathcal{B}_p^h) - s_j - t_{j,h+n})\right\}, & \text{otherwise.}\end{cases} \tag{63}$$

The extension functions for $\mathcal{B}_q^h$, $ld_q^h(\mathcal{A}_q)$, $ld_q^h(\mathcal{B}_q^h)$, and b_j are introduced by Gschwind and Irnich (2015). Then,

$\mathbb{L}_q$ is feasible if the following constraints are satisfied:

$$\mathcal{U}_q^1 + \mathcal{U}_q^2 + \mathcal{U}_q^3 \le Q_\psi^d, \tag{64}$$
$$\mathcal{A}_q + s_j + t_{j,0} \le ld_q^0(\mathcal{A}_q), \tag{65}$$
$$\mathcal{A}_q \le ld_p^{j-n}(\mathcal{A}_p) \wedge j \in D, \tag{66}$$
$$\mathcal{A}_q + s_j + t_{j,h} \le ld_q^h(\mathcal{A}_q), \quad \forall h \in \mathcal{V}_q \cap P \wedge h + n \notin \mathcal{V}_q. \tag{67}$$

Constraint (64) not only ensures that the capacity constraint of Q_ψ^d is satisfied, but it also implies that the capacity constraint of Q_ψ^f is satisfied since the exceeding part of the required seats is transformed into configurable divisions. Constraint (65) guarantees that the maximum duration constraint can be satisfied. At first glance, constraint (66) seems to only guarantee that when the service start time at node i is $\mathcal{A}_p$, the maximum riding time constraint of client $j-n$ is satisfied. However, in fact, constraint (66) also ensures that the maximum riding time constraint of client $j-n$ is satisfied when the service start time at node i is later than $\mathcal{A}_p$. We show this by two cases. First, suppose the service start time at node j is $\mathcal{A}_q$. Then $\mathcal{A}_q = e_j$ according to (52); otherwise, the service start time at node i is exactly $\mathcal{A}_p$. As $e_j \le ld_p^{j-n}(t)$ for any $t \ge \mathcal{A}_p$, we conclude that the maximum riding time constraint of client $j-n$ is satisfied. Second, suppose the service start time at node j is postponed to $\mathcal{A}_q + t_\delta$. According to constraint (66), we have $\mathcal{A}_q + t_\delta \le ld_p^{j-n}(\mathcal{A}_p) + t_\delta$. From (63) and (60), we have $\mathcal{A}_q + t_\delta \le b_j \le ld_p^{j-n}(\mathcal{B}_p^{j-n})$. Hence, $\mathcal{A}_q + t_\delta \le \min\{ld_p^{j-n}(\mathcal{A}_p) + t_\delta, ld_p^{j-n}(\mathcal{B}_p^{j-n})\} = ld_p^{j-n}(\mathcal{A}_p + t_\delta)$ and the maximum riding time constraint of client $j-n$ is satisfied. Constraint (67) ensures that the maximum riding time constraints of unfulfilled clients in q are not violated.

Dominance. During the partial trip extension, if a partial trip p is dominated by another trip p', p can be safely discarded. By dropping the dominated partial trips, a label-setting algorithm can significantly reduce the number of extensions and, hence, improve efficiency. In our problem, a partial trip is considered to be dominated by another if the following conditions hold:

Definition 6. Given a partial trip p and a feasible extension r from p to the depot, let (p,r) be the complete trip created by joining p and r. Then, for any two forward partial trips p and q, p dominates q if for any feasible extension r of q, there exists a feasible extension r' of p such that trip (p,r') dominates trip (q,r) by Definition 2 or 3.

Obviously, the dominated partial trips can be safely discarded in the label-setting algorithm without losing any nondominated trips. Now, we introduce the dominance rules used in the label-setting algorithm. Let $\mathbb{L}_p$ and $\mathbb{L}_q$ be two labels associated with the same node. Then $\mathbb{L}_p$ dominates $\mathbb{L}_q$ if

$$\mathcal{C}_p \le \mathcal{C}_q, \tag{68}$$
$$\mathcal{V}_p \cap P = \mathcal{V}_q \cap P, \tag{69}$$
$$\mathcal{V}_p \cap D \supset \mathcal{V}_q \cap D, \tag{70}$$
$$\mathcal{Q}_p^{1,\max} \le \mathcal{Q}_q^{1,\max}, \tag{71}$$
$$\mathcal{U}_p^c \le \mathcal{U}_q^c, \quad c = 1,2,3, \tag{72}$$
$$\mathcal{A}_p \le \mathcal{A}_q, \tag{73}$$
$$ld_p^h(\mathcal{A}_p) + (\mathcal{A}_p - \mathcal{A}_q) \ge ld_q^h(\mathcal{A}_q), \quad h \in \{0\} \cup \mathcal{V}_q \cap P \quad \text{and} \quad h + n \notin \mathcal{V}_q, \tag{74}$$
$$ld_p^h(\mathcal{B}_p^h) \ge ld_q^h(\mathcal{B}_p^h), \quad h \in \{0\} \cup \mathcal{V}_q \cap P, \text{ and } h + n \notin \mathcal{V}_q. \tag{75}$$

Theorem 1. *If the travel time matrix $\{t_{i,j}\}_{(i,j)\in A}$ satisfies the triangle inequality, dominance rules (68)–(75) are valid.*

Proof. See Online Appendix C. □

Label Eliminations. We follow the label elimination approach proposed by Dumas, Desrosiers, and Soumis (1991) to eliminate infeasible labels. Given a label $\mathbb{L}_p$, if no feasible extension from p to the depot can be found to visit the nodes in $\{i \mid i-n \in \mathcal{V}_p \cap P \wedge i \notin \mathcal{V}_p \cap D\}$, $\mathbb{L}_p$ is proven infeasible. However, to prove that no extension can cover all of the nodes in $\{i \mid i-n \in \mathcal{V}_p \cap P \wedge i \notin \mathcal{V}_p \cap D\}$ is NP-hard in general. Thus, as in Dumas, Desrosiers, and Soumis (1991), the problem is restricted to consider at most two nodes in $\{i \mid i-n \in \mathcal{V}_p \cap P \wedge i \notin \mathcal{V}_p \cap D\}$.

Trip Generation. Given a nondominated label $\mathbb{L}_p$ at the depot, the attributes associated with the trip r represented by label $\mathbb{L}_p$ are as follows:

$$C_r = \mathcal{C}_p, \tag{76}$$
$$V_r = \mathcal{V}_p \cap P, \tag{77}$$
$$E_r = \mathcal{A}_p, \tag{78}$$
$$L_r = ld_p^0(\mathcal{B}_p^0) - T, \tag{79}$$
$$Y_r(t) = \min\{L_r, t - d_r\}, \ t \ge E_r, \tag{80}$$
$$X_r(t) = \max\{E_r, t + d_r\}, \ t \le L_r, \tag{81}$$

where

$$d_r = T - (ld_p^0(\mathcal{A}_p) - \mathcal{A}_p). \tag{82}$$

Equations (76)–(78) are obvious. Given the completion time of the route r is t, the latest time to depart from the depot is $ld_p^0(t) - T$. According to Proposition 4 in Gschwind and Irnich (2015)

$$\begin{aligned} ld_p^0(t) &= \min\{ld_p^0(\mathcal{A}_p) + (t - \mathcal{A}_p), ld_p^0(\mathcal{B}_p^0)\} \\ &= \min\{t + ld_p^0(\mathcal{A}_p) - \mathcal{A}_p, ld_p^0(\mathcal{B}_p^0)\}. \end{aligned} \tag{83}$$

Therefore,

$$\begin{aligned} Y_r(t) &= \min\{t + ld_p^0(\mathscr{A}_p) - \mathscr{A}_p, ld_p^0(\mathscr{B}_p^0)\} - T \\ &= \min\{t - (T - (ld_p^0(\mathscr{A}_p) - \mathscr{A}_p)), ld_p^0(\mathscr{B}_p^0) - T\} \\ &= \min\{L_r, t - d_r\}. \end{aligned} \tag{84}$$

Equation (81) is easily obtained from Equation (80).

4.3.2. The Label-Setting Algorithm for Column Generation. Let μ_r $(r \in R_N)$, $\nu^1_{r_1,r_2}((r_1,r_2)\in A_1)$, $\nu^2_{\psi,r_1,r_2}(\psi\in\Psi_A, (r_1,r_2)\in A_2)$, and $\pi_\psi(\psi\in\Psi_A)$ denote the dual values of constraints (36), (23), (24), and (16) in Model (V) or constraints (42)–(45) in Model (VII). The pricing problems of Model (V) and Model (VII) are the same and can be formulated as follows:

$$\min_{\omega\in\Omega_\psi,\,\psi\in\Psi_A}\left\{0 - \sum_{r\in\omega}\mu_r - \sum_{(r_1,r_2)\in A_1}\sum_{(r_1,r_2)\in\omega}\nu^1_{r_1,r_2} - \sum_{(r_1,r_2)\in A_2}\sum_{(r_1,r_2)\in\omega}\nu^2_{\psi,r_1,r_2} - \pi_\psi\right\}. \tag{85}$$

Let $\varphi_{\mu,\nu^1,\nu^2,\pi}$ $((\mu,\nu^1,\nu^2,\pi)\in\mathbb{R}_\Delta)$, τ_i $(i\in P)$, σ_{r_1,r_2} $(r_1,r_2\in R)$, and χ_ψ $(\psi\in\Psi_V)$ denote the dual values of constraints (32), (10), (13), and (15), respectively. The pricing problem of Model (IV) can be formulated as follows:

$$\min_{\omega\in\Omega_\psi,\,\psi\in\Psi_V}\left\{c_\omega - \sum_{r\in\omega}\sum_{(\mu,\nu^1,\nu^2,\pi)\in\mathbb{R}}\mu_r F_r\varphi_{\mu,\nu^1,\nu^2,\pi} - \sum_{(r_1,r_2)\in\omega}\sigma_{r_1,r_2} - \sum_{i\in\omega\cap P}\tau_i - \chi_\psi\right\}. \tag{86}$$

Let $G'_\psi = (V'_\psi, A'_\psi)$ $(\psi\in\Psi)$ be a directed graph corresponding to staff member ψ, where V'_ψ is the node set and A'_ψ is the arc set. Each nondominated trip $r\in R$ corresponds to a node in V'_ψ. Therefore, each node $r\in V'_\psi$ is associated with attributes $C_r, F_r, V_r, E_r, L_r, X_r(t)$, and $Y_r(t)$, as defined in Section 3.2. For each node $r\in V'_\psi$ and arc $(r_1,r_2)\in A'_\psi$, we assign them a cost, namely c'_r and c'_{r_1,r_2}, respectively. For $\psi\in\Psi_V$, c'_r and c'_{r_1,r_2} are set as follows:

$$c'_r = \begin{cases} -\chi_\psi, & \text{if } r = r^s_\psi, \\ C_r - \displaystyle\sum_{(\mu,\nu^1,\nu^2,\pi)\in\mathbb{R}_\Delta}\mu_r F_r\varphi_{\mu,\nu^1,\nu^2,\pi} - \sum_{i\in V_r}\tau_i, & \\ \qquad\text{otherwise}, & \end{cases} \tag{87}$$

$$c'_{r_1,r_2} = \begin{cases} -\sigma_{r_1,r_2}, & \text{if } \sigma_{r_1,r_2} \text{ exists}, \\ 0, & \text{otherwise}. \end{cases} \tag{88}$$

For $\psi\in\Psi_A$, c'_r and c'_{r_1,r_2} are set as follows:

$$c'_r = \begin{cases} -\pi_\psi, & \text{if } r = r^s_\psi, \\ -\mu_r, & \text{otherwise}, \end{cases} \tag{89}$$

$$c'_{r_1,r_2} = \begin{cases} -\nu^1_{r_1,r_2}, & \text{if } (r_1,r_2)\in A_1, \\ -\nu^2_{\psi,r_1,r_2}, & \text{if } (r_1,r_2)\in A_2, \\ 0, & \text{otherwise}. \end{cases} \tag{90}$$

Obviously, for a trip r, if $c'_r = 0$, $c'_{r_1,r} = 0$, $c'_{r,r_2} = 0$, for all $r_1, r_2 \in V_\psi$, and there exists no arc (r_1,r_2) such that $c'_{r_1,r_2}\neq 0$ and path (r_1,r,r_2) is feasible, r can be removed from any feasible workday without affecting the cost of the workday. The pricing problems of Models (V)–(VII) all correspond to finding a feasible route with minimum cost in G'_ψ $(\psi\in\Psi)$. A route in G'_ψ is feasible if it is a feasible workday according to Definition 5. Next, we describe the bounded bidirectional label-setting algorithm (Righini and Salani 2006, 2008) for finding a feasible route with minimum cost in G'_ψ.

Label Definitions. Let $\mathbb{L}^f_p = (\mathscr{C}^f_p, \mathscr{B}^f_p, \mathscr{A}^f_p, \mathscr{U}^f_p, \mathscr{V}^f_p)$ be a forward label associated with node $p\in V'_\psi$, where

- $\mathscr{C}^f_p$ is the cost;
- $\mathscr{B}^f_p$ is a binary resource equal to one if and only if the lunch break node r^l_ψ has been visited;
- $\mathscr{A}^f_p$ is the earliest completion time at node p;
- $\mathscr{U}^f_p$ is the set of clients to which the label cannot be extended;
- $\mathscr{V}^f_p$ is the set of clients which the label has visited.

Let $\mathbb{L}^b_p = (\mathscr{C}^b_p, \mathscr{B}^b_p, \mathscr{L}^b_p, \mathscr{U}^b_p, \mathscr{V}^b_p)$ be a backward label associated with node $p\in V'_\psi$, where

- $\mathscr{C}^b_p$ is the cost;
- $\mathscr{B}^b_p$ is a binary resource equal to one if and only if the lunch break node r^l_ψ has been visited;
- $\mathscr{L}^b_p$ is the latest start time at node p;
- $\mathscr{U}^b_p$ is the set of clients to which the label cannot be extended;
- $\mathscr{V}^b_p$ is the set of clients that the label has visited.

Extensions. In the forward extension, given a label $\mathbb{L}^f_{r_1}$ and an arc $(r_1,r_2)\in A'_\psi$, label $\mathbb{L}^f_{r_1}$ can be extended to node r_2 if

$$\mathscr{A}^f_{r_1} + t^B \le L_{r_2}, \tag{91}$$

$$\mathscr{U}^f_{r_1} \cap V_{r_2} = \varnothing. \tag{92}$$

If these constraints are satisfied, a new label $\mathbb{L}^f_{r_2} = (\mathscr{C}^f_{r_2}, \mathscr{B}^f_{r_2}, \mathscr{A}^f_{r_2}, \mathscr{U}^f_{r_2}, \mathscr{V}^f_{r_2})$ is created. The resources are updated according to the following resource extension functions:

$$\mathscr{C}^f_{r_2} = \mathscr{C}^f_{r_1} + c'_{r_1,r_2} + c'_{r_2}, \tag{93}$$

$$\mathscr{A}^f_{r_2} = X_{r_2}(\mathscr{A}^f_{r_1} + t^B), \tag{94}$$

$$\mathscr{U}^f_{r_2} = \mathscr{U}^f_{r_1} \cup V_{r_2} \cup \{i \mid i\in P, \mathscr{A}^f_{r_2} > \mathfrak{L}^1_i\}, \tag{95}$$

$$\mathscr{V}^f_{r_2} = \mathscr{V}^f_{r_1} \cup V_{r_2}, \tag{96}$$

$$\mathscr{B}^f_{r_2} = \begin{cases} 1, & \text{if } r_2 = r_\phi, \\ \mathscr{B}^f_{r_1}, & \text{otherwise}, \end{cases} \tag{97}$$

where $\mathfrak{L}^1_i$ is the latest time to start a trip that serves client i among all of the trips. The forward extension of label $\mathbb{L}^f_r$ stops if $\mathscr{A}^f_r \ge e_\psi + (l_\psi - e_\psi)/2$.

In the backward extension, given a label $\mathbb{L}^b_{r_1}$ and an arc $(r_2, r_1) \in A'_\psi$, label $\mathbb{L}^b_{r_1}$ can be extended to node r_2 if

$$\mathscr{L}^b_{r_1} \geq E_{r_2} + t^B, \tag{98}$$
$$\mathscr{U}^b_{r_1} \cap V_{r_2} = \varnothing. \tag{99}$$

If these constraints are satisfied, a new label $\mathbb{L}^b_{r_2} = (\mathscr{C}^b_{r_2}, \mathscr{B}^b_{r_2}, \mathscr{L}^b_{r_2}, \mathscr{U}^b_{r_2}, \mathscr{V}^b_{r_2})$ is created. The resources are updated according to the following resource extension functions:

$$\mathscr{C}^b_{r_2} = \mathscr{C}^b_{r_1} + c'_{r_2,r_1} + c'_{r_2}, \tag{100}$$
$$\mathscr{L}^b_{r_2} = Y_{r_2}(\mathscr{L}^b_{r_1} - t^B), \tag{101}$$
$$\mathscr{U}^b_{r_2} = \mathscr{U}^b_{r_1} \cup V_{r_2} \cup \{i \mid i \in P, \mathscr{L}^b_{r_2} < \mathfrak{L}^2_i\}, \tag{102}$$
$$\mathscr{V}^b_{r_2} = \mathscr{V}^b_{r_1} \cup V_{r_2}, \tag{103}$$
$$\mathscr{B}^b_{r_2} = \begin{cases} 1, & \text{if } r_2 = r_\phi, \\ \mathscr{B}^b_{r_1}, & \text{otherwise,} \end{cases} \tag{104}$$

where $\mathfrak{L}^2_i$ is the earliest time to finish a trip that serves client i among all of the trips. The backward extension of label $\mathbb{L}^b_r$ stops if

$$\mathscr{L}^b_r - \min_{r' \in V'_\psi \text{ and } L_{r'} + d_{r'} \geq (e_\psi + (l_\psi - e_\psi)/2)} (d_{r'} + t^B) < e_\psi + \frac{l_\psi - e_\psi}{2}$$

where d_r is defined in Equation (82).

Dominance. In the forward extension, given two labels $\mathbb{L}^f_{r_1}$ and $\mathbb{L}^f_{r_2}$, $\mathbb{L}^f_{r_1}$ dominates $\mathbb{L}^f_{r_2}$ if

$$r_1 = r_2 \wedge \sum_{r \in V'_\psi} c'_{r_1,r} > 0, \tag{105}$$
$$\mathscr{C}^f_{r_1} \leq \mathscr{C}^f_{r_2}, \tag{106}$$
$$\mathscr{A}^f_{r_1} \leq \mathscr{A}^f_{r_2}, \tag{107}$$
$$\mathscr{U}^f_{r_1} \subseteq \mathscr{U}^f_{r_2}, \tag{108}$$
$$\mathscr{B}^f_{r_1} = \mathscr{B}^f_{r_2}. \tag{109}$$

Here, we only require $\mathbb{L}^f_{r_1}$ and $\mathbb{L}^f_{r_2}$ to associate with the same node when there exists a positive cost $c'_{r_1,r} (r \in V'_\psi)$. Note that according to Equations (88) and (90), the arc cost is always nonnegative.

In the backward extension, given two labels $\mathbb{L}^b_{r_1}$ and $\mathbb{L}^b_{r_2}$, $\mathbb{L}^b_{r_1}$ dominates $\mathbb{L}^b_{r_2}$ if

$$r_1 = r_2 \wedge \sum_{r \in V'_\psi} c'_{r,r_1} > 0, \tag{110}$$
$$\mathscr{C}^b_{r_1} \leq \mathscr{C}^b_{r_2}, \tag{111}$$
$$\mathscr{L}^b_{r_1} \geq \mathscr{L}^b_{r_2}, \tag{112}$$
$$\mathscr{U}^b_{r_1} \subseteq \mathscr{U}^b_{r_2}, \tag{113}$$
$$\mathscr{B}^b_{r_1} = \mathscr{B}^b_{r_2}. \tag{114}$$

Label Joining. A forward label $\mathbb{L}^f_{r_1}$ and a backward label $\mathbb{L}^b_{r_2}$ can be joined to form a complete workday if the following constraints are satisfied:

$$\mathscr{B}^f_{r_1} + \mathscr{B}^b_{r_2} = 1, \tag{115}$$
$$\mathscr{A}^f_{r_1} + t^B \leq \mathscr{L}^b_{r_2}, \tag{116}$$
$$\mathscr{V}^f_{r_1} \cap \mathscr{V}^b_{r_2} = \varnothing. \tag{117}$$

The cost of the resultant workday is equal to $\mathscr{C}^f_{r_1} + \mathscr{C}^b_{r_2} + c'_{r_1,r_2}$.

In many routing problems, pricing out elementary routes is usually very time-consuming so that it can possibly slow down a branch-and-price algorithm for solving the corresponding problems dramatically. Therefore, pricing out nonelementary routes is usually a possible way of speeding up a branch-and-price algorithm since pricing out nonelementary routes is more efficient. However, using nonelementary routes in the master problem will deteriorate the lower bound yielded by the master problem so that a branch-and-price algorithm has to explore more nodes to achieve an optimal solution. Therefore, we conduct an experiment to study whether the ng-path relaxation proposed by Baldacci, Mingozzi, and Roberti (2011), an efficient way for pricing out nonelementary routes, can accelerate our branch-and-price-and-cut algorithm. According to the experiment, no clear evidence shows that the ng-path relaxation can speed up the overall algorithm. The details of the experiment are presented in Online Appendix E.

4.4. Primal Heuristic

We propose a simple primal heuristic to find a feasible solution according to the optimal LP solution at each node. Let $\bar{\Omega}_V$ and $\bar{\Omega}_A$ be the set of workdays with nonzero values in the optimal LP solution for the vehicles and the assistants, respectively. The primal heuristic consists of three stages. In the first stage, a set of workdays for the vehicles is recovered according to the optimal LP solution. Let $\bar{S}_V$ be the set of workdays for the vehicles recovered, which is initialized by the workdays with integral values and maximum fractional value in $\bar{\Omega}_V$. In each iteration, a workday $\omega' \in \bar{\Omega}_V$ with minimum cost is selected and added to $\bar{S}_V$. Then, the workdays in $\bar{\Omega}_V$ that share the same clients with ω' are removed from $\bar{\Omega}_V$. This process is repeated until $\bar{\Omega}_V$ becomes empty. In the second stage, a set of workdays for the assistants, denoted by $\bar{S}_A$, are recovered. First, trips that are not in $\bar{S}_V$ are removed from the workdays in $\bar{\Omega}_A$. Then, the heuristic iteratively selects a workday in $\bar{\Omega}_A$ that covers the most uncovered trips in $\bar{S}_V$ and adds it to $\bar{S}_A$ until all of the assistants have been assigned a workday. After that, the workdays in $\bar{S}_V$ that contain trips that cannot be covered by the workdays in $\bar{S}_A$ are removed. In the last stage, an LP problem is solved to determine feasible start times for trips in $\bar{S}_V$

and $\bar{S}_A$. Note that the primal heuristic is not guaranteed to find a feasible solution, because the LP problem is not necessarily feasible.

4.5. Branching Strategies

The branch-and-price-and-cut algorithm uses six branching rules in its hierarchy. Let $\bar{\theta}_\omega$ $(\omega \in \Omega_\psi, \psi \in \Psi_V)$ be an optimal solution of Model (IV) and $\bar{\theta}_\omega$ $(\omega \in \Omega_\psi, \psi \in \Psi_A)$ be an optimal solution of Model (V).

The first branching rule is to branch on vehicles. Let $\bar{K}_\psi = \sum_{\omega \in \Omega_\psi} \bar{\theta}_\omega$ $(\psi \in \Psi)$ be the number of vehicles ψ used in the current fractional solution. If there exists a fractional $\bar{K}_\psi$ $(\psi \in \Psi_V)$, we select a ψ with $\bar{K}_\psi$ closest to 0.5 to branch the current node into two child nodes. In one child node, vehicle ψ is forbidden to be used, which is equivalent to dropping all of the workdays in Ω_ψ while in the other child node, vehicle ψ is forced to be used, which is equivalent to changing constraint (15) with respect to ψ to equality.

The second branching rule branches on clients. Let $\bar{\beta}_i = \sum_{\psi \in \Psi_V} \sum_{\omega \in \Omega_\psi} \beta_{i,\omega} \bar{\theta}_\omega$. We select a client i with $\bar{\beta}_i$ closest to 0.5 to branch on. In one child node, client i is forbidden to be served, which is equivalent to dropping all of the trips containing i as well as the affected workdays while, in the other child node, client i must be served, which is equivalent to changing the corresponding constraint (10) to equality.

The third branching rule branches on trips that visit a certain pair of clients. For a pair of clients (i, j) $(i, j \in P)$ and a workday $\omega \in \Omega$, let $\eta_{i,j,\omega}$ be a binary number equal to one if and only if ω has a trip serving both clients i and j. Let $\bar{\eta}_{i,j} = \sum_{\psi \in \Psi_V} \sum_{\omega \in \Omega_\psi} \eta_{i,j,\omega} \bar{\theta}_\omega$. We select a pair (i, j) $(i, j \in P)$ with $\bar{\eta}_{i,j}$ closest to 0.5 to branch on. In one child node, clients (i, j) are forbidden to be served by the same trip, which is equivalent to dropping all of the trips containing clients (i, j) as well as the affected workdays. In the other child node, clients (i, j) are forced to be served on the same trip, which can be implemented by changing constraints (10) with respect to nodes i and j to equalities and dropping trips that contain either node i or node j, but not both, as well as the affected workdays.

The fourth branching rule branches on arcs in graph G. For an arc $(i, j) \in A$ and a workday $\omega \in \Omega$, let $\kappa_{i,j,\omega}$ be a binary number that is equal to one if and only if ω has a trip covering arc (i, j). Let $\bar{\kappa}_{i,j} = \sum_{\psi \in \Psi_V} \sum_{\omega \in \Omega_\psi} \kappa_{i,j,\omega} \bar{\theta}_\omega$. We select an arc (i, j) $((i, j) \in A)$ with $\bar{\kappa}_{i,j}$ closest to 0.5 to branch on. In one child node, arc (i, j) is forbidden to be used by any vehicle, which is equivalent to dropping all of the trips containing arc (i, j) as well as the affected workdays. In the other child node, arc (i, j) is forced to be covered by one vehicle, which can be implemented by changing constraints (10) with respect to nodes i and j to equalities and dropping trips that contain either node i or node j but not arc (i, j) as well as the affected workdays.

The fifth branching rule branches on whether two trips are used by a vehicle consecutively. Let $\bar{\rho}^1_{r_1,r_2} = \sum_{\psi \in \Psi_V} \sum_{\omega \in \Omega_\psi} \rho_{r_1,r_2,\omega} \bar{\theta}_\omega$. Then we select two trips r_1 and r_2 that have $\bar{\rho}^1_{r_1,r_2}$ closest to 0.5 and branch the current node into two child nodes. In one child node, trips r_1 and r_2 are forbidden to be used by a vehicle consecutively, which is equivalent to forcing the right-hand side value of the corresponding constraint (13) to zero. In the other child node, a vehicle is forced to use trips r_1 and r_2 consecutively. This can be achieved by changing constraints (13) with respect to trips r_1 and r_2 to equality and forcing $\lambda_{r_1,r_2} = 1$.

The last branching rule branches on whether two trips are taken by an assistant consecutively. Let $\bar{\rho}^2_{r_1,r_2} = \sum_{\psi \in \Psi_A} \sum_{\omega \in \Omega_\psi} \rho_{r_1,r_2,\omega} \bar{\theta}_\omega$ if $(r_1, r_2) \in A_1$ and $\bar{\rho}^3_{\psi,r_1,r_2} = \sum_{\omega \in \Omega_\psi} \rho_{r_1,r_2,\omega} \bar{\theta}_\omega$ if $(r_1, r_2) \in A_2$. If there exists a fractional $\bar{\rho}^2$, we select two trips $(r_1, r_2) \in A_1$ with $\bar{\rho}^2_{r_1,r_2}$ closest to 0.5; otherwise, we select two trips $(r_1, r_2) \in A_2$ and assistant $\psi \in \Psi_A$ with $\bar{\rho}^3_{\psi,r_1,r_2}$ closest to 0.5. In one child node, trips r_1 and r_2 are forbidden to be consecutively used by an assistant, which can be enforced by changing the right-hand side values of constraints (20) $((r_1, r_2) \in A_1)$ or constraints (14) $((r_1, r_2) \in A_2)$ with respect to trips r_1 and r_2 to zero. In the other child node, trips r_1 and r_2 must be consecutively used by an assistant. This can be enforced by changing constraints (20) $((r_1, r_2) \in A_1)$ or constraints (14) $((r_1, r_2) \in A_2)$ with respect to trips r_1 and r_2 to equality and forcing $\lambda_{r_1,r_2} = 1$. According to our computational experience, the last two branching rules are seldom used in the algorithm. They are implemented to ensure the correctness of our algorithm. Meanwhile, additional experiments are conducted to analyze the impacts of the first four branching rules on the branch and price and cut. The computational results are summarized in Online Appendix F.

5. Computational Experiments

In this section, we test our two-phase, branch-and-price-and-cut algorithm on the R-DARP benchmark instances introduced by Liu, Luo, and Lim (2015) and a set of new ER-DARP instances that are generated according to the same real-world data set as the R-DARP benchmark instances but have more requests compared with the R-DARP instances. The algorithm was implemented in Java 1.7.0 with ILOG Cplex 12.5.1 to solve the restricted master problem. All of the experiments were conducted on a Dell personal computer with an Intel i7-4790 3.6 GHz CPU, 16 GB RAM, and a Windows 7 operating system.

5.1. Results on the R-DARP Instances

In this section, we compare our two-phase, branch-and-price-and-cut algorithm with the branch-and-cut algorithm proposed by Liu, Luo, and Lim (2015) on

52 R-DARP benchmark instances. The R-DARP benchmark instances are generated according to the operational data of NEATS. The NEATS data set contains detailed information about client requests, vehicles, and staff in 2009, including the pickup and delivery locations of each client; the travel time between any two locations; the time window on each pickup and delivery node; the number of seats, wheelchairs, stretchers, and assistants required by each client; the work period and the lunch period of each staff member; and the capacity configuration of each vehicle, that is, the number of fixed seats and configurable divisions. Other information required by the R-DARP but not available in the NEATS data set, such as the maximum riding time of client i (r_i) and the maximum duration of a trip (T), is determined as follows: $r_i = \min\{60, 3\max\{10, t_{i,i+n}\}\}$ and $T = 150$ (minutes). Because the branch-and-cut algorithm is only expected to optimally solve small instances as a result of the complexity of the R-DARP, the R-DARP instances consist of only a limited number of requests, varying from 16 to 23. These requests are randomly sampled from all of the requests in a work day in the NEATS data set, where there can be more than 100 requests in a work day. For detailed information about the R-DARP instances, please see Liu, Luo, and Lim (2015).

Both the proposed algorithm and the branch-and-cut algorithm were implemented in Java and run on the same machine. We limited the running time of the proposed algorithm to four hours, which is the same as for the branch-and-cut algorithm. Table 2 presents the details of the computational results. Column *Instance* gives the name of an instance. The first and second numbers in the name of an instance are the number of requests and vehicles in the instance, respectively. For example, in the first instance *16-2-a*, there are 16 requests and two vehicles. The *LB* columns give the lower bounds obtained by the algorithms at the root node of the branch-and-bound tree. Column *LB time* reports the computational time to solve the LP relaxation of Model (V) at the root node in the branch-and-price-and-cut algorithm. The *Best LB* columns give the best lower bounds obtained by the algorithms during the branch-and-bound process. The *Total time* columns report the total time in seconds for the algorithms either to achieve an optimal solution or to terminate as a result of the time limit. The *Nodes* columns give the number of nodes explored by the algorithms. If an algorithm achieves an optimal solution for an instance, the best lower bound obtained by this algorithm for this instance is marked in bold.

From Table 2, it can be seen that the proposed algorithm can solve all of the instances to optimality very quickly while the branch-and-cut algorithm can only solve 22 instances to optimality. On average, the proposed algorithm takes 4.5 seconds to optimally solve these instances while the branch-and-cut algorithm takes about 8,562 seconds. Obviously, the proposed algorithm performs much better than the branch-and-cut on the R-DARP benchmark instances. The success of our algorithm is mainly a result of the strong lower bound yielded by the LP relaxation of Model (V) as we can see that the lower bounds at the root node of the branch-and-price-and-cut algorithm are much better than those of the branch-and-cut algorithm.

5.2. Results for the ER-DARP Instances

Because the R-DARP benchmark instances can be easily solved to optimality by the proposed algorithm, we generate a set of larger instances using the same data-generation method as the R-DARP instances. These new instances are referred to as the *ER-DARP instances*. Compared with the R-DARP instances, the ER-DARP instances have more requests and may not have enough vehicles or manpower to satisfy all of the requests. Table 3 shows the detailed characteristics of all of the instances in the ER-DARP data set. The columns *Instance*, n, $|K|$, and $|H|$ give the name, number of requests, number of vehicles, and number of assistants in each instance, respectively. Column $p_{o>1}$ reports the proportion of requests that require more than one assistant in each instance. This proportion is consistent with that in the NEATS data set. The next three columns provide the statistical information about the width of the clients' time windows (in minutes) that are defined as $\min\{l_i - e_i, l_{i+n} - e_{i+n}\}$ for each client i. Column r_i gives the average maximum riding time of the clients. The next three columns present statistical information about the minimum traveling times of the clients. Here, the minimum traveling time of a client is the direct traveling time from her origin to her destination. The next eight columns present the statistical information about the work time windows of the staff in which the *var* columns represent the variance. The last column α gives the coefficient α in the objective computed by the method in Online Appendix B. From the table, we can see that the average time window width of the clients is more than two hours, and the average maximum riding time of the clients is about half an hour while the average minimum traveling time of the clients is about 10 minutes. This suggests that the clients in the instances have loose time windows and maximum riding time constraints compared with the traveling times.

Table 4 summarizes the computational results of our algorithm for the ER-DARP instances. The first column gives the name of the instances. The second and third columns show the number of nondominated trips generated by the label-setting algorithm in the first phase and the computational time in seconds to generate them, respectively. Columns 4–10 show the information about the LP relaxation of the Benders master

Table 2. Computational Results for the R-DARP Benchmark Instances

		Two-phase, branch-and-price-and-cut					Branch-and-cut			
Instance	UB	LB	LB time	Best LB	Total time	Nodes	LB	Best LB	Total time	Nodes
16-2-a	238	238.00	0	**238.00**	0	1	206.50	**238.00**	5,822	66,916
16-2-b	221	220.71	0	**221.00**	0	5	175.51	**221.00**	4,178	40,568
16-2-c	241	241.00	0	**241.00**	0	1	206.46	**241.00**	1,889	13,407
16-2-d	366	366.00	0	**366.00**	0	1	315.10	**366.00**	562	10,900
16-2-e	293	293.00	0	**293.00**	0	1	254.02	**293.00**	926	11,275
16-2-f	287	286.50	0	**287.00**	0	3	231.19	**287.00**	1,721	27,777
17-2-a	254	254.00	0	**254.00**	0	1	216.87	**254.00**	5,472	36,092
17-2-b	224	224.00	0	**224.00**	0	1	197.52	**224.00**	1,560	10,788
17-2-c	176	176.00	0	**176.00**	0	1	146.62	**176.00**	2,768	22,341
17-2-d	249	249.00	0	**249.00**	0	1	203.18	**249.00**	1,566	26,507
17-2-e	207	202.33	0	**207.00**	0	3	171.54	**207.00**	370	4,853
17-2-f	282	282.00	0	**282.00**	0	1	206.20	**282.00**	2,636	36,411
18-2-a	220	220.00	0	**220.00**	0	1	176.87	**220.00**	3,718	22,159
18-2-b	222	222.00	0	**222.00**	0	1	176.86	**222.00**	9,497	35,094
18-2-c	236	236.00	0	**236.00**	0	1	182.56	**236.00**	2,063	15,660
18-2-d	274	272.00	0	**274.00**	0	7	228.13	**274.00**	3,757	50,118
18-2-e	330	330.00	0	**330.00**	0	1	259.95	**330.00**	1,329	10,201
18-2-f	287	287.00	0	**287.00**	0	1	226.42	**287.00**	9,953	104,660
19-2-a	236	234.50	0	**236.00**	0	5	205.13	231.71	14,400	115,031
19-2-b	224	222.00	0	**224.00**	0	5	167.85	191.59	14,400	18,126
19-2-c	228	221.44	0	**228.00**	8	253	186.36	197.19	14,400	17,462
19-2-d	284	284.00	0	**284.00**	0	1	224.64	**284.00**	4,270	26,016
19-2-e	285	285.00	0	**285.00**	0	1	234.30	**285.00**	1,290	12,950
19-2-f	304	304.00	0	**304.00**	0	1	245.41	301.05	14,400	155,292
20-2-a	283	282.80	0	**283.00**	0	3	244.64	258.19	14,400	19,390
20-2-b	266	264.50	0	**266.00**	0	7	216.19	228.50	14,400	36,256
20-2-c	300	297.33	0	**300.00**	0	7	251.25	274.63	14,400	70,565
20-2-d	328	321.40	1	**328.00**	5	15	262.36	308.36	14,400	64,200
20-2-e	285	285.00	0	**285.00**	0	1	249.16	**285.00**	1,952	6,510
20-2-f	272	272.00	0	**272.00**	0	1	235.33	**272.00**	4,285	31,682
21-2-a	289	289.00	0	**289.00**	0	3	227.86	255.65	14,400	22,509
21-2-b	237	237.00	0	**237.00**	0	3	205.13	229.78	14,400	59,805
21-2-c	176	176.00	7	**176.00**	7	1	142.40	154.82	14,400	53,505
21-2-d	319	319.00	0	**319.00**	0	1	233.43	276.05	14,400	53,359
21-2-e	319	319.00	0	**319.00**	0	1	251.16	289.55	14,400	21,079
21-2-f	317	317.00	0	**317.00**	0	1	235.32	294.67	14,400	95,200
22-2-a	285	281.60	0	**285.00**	1	15	207.51	226.17	14,400	11,044
22-2-b	215	215.00	8	**215.00**	10	3	167.01	175.76	14,400	11,394
22-2-c	219	216.33	5	**219.00**	33	23	145.19	152.78	14,400	31,938
23-2-a	243	242.67	4	**243.00**	12	19	171.94	192.65	14,400	11,502
23-2-b	273	272.73	0	**273.00**	1	5	210.11	227.90	14,400	23,436
23-2-c	262	256.75	2	**262.00**	110	19	202.71	223.10	14,400	24,555
Average	263	262.28	0.6	263.24	4.5	10.1	212.00	248.17	8,562	36,632

problem (Model (IV)) at the root node of the branch-and-bound tree. Specifically, column *Objective* gives the objective value of the optimal LP solution; the percentage of requests satisfied and the total travel distance of the vehicles are reported in columns *Requests* and *Distance*, respectively; column *Gap* gives the percentage gap between the objective values of the optimal LP solution and the optimal integral solution; column *LS time* shows the computational time in seconds taken by the label-setting algorithm; column *BMP solved* gives the frequency that the LP relaxation of the Benders master problem is optimally solved, which is equal to the number of iterations taken by Algorithm 1; and column *Time* gives the computational time in seconds to solve the root node. Columns 11–13 show the best upper bounds obtained by the algorithm. If the algorithm successfully achieves an optimum of an instance, the values in the *Objective* column are marked in bold. The last four columns, *LS time*, *BMP solved*, *Total time*, and *Nodes*, give the total label-setting time in seconds, the total frequency that the LP relaxation of the Benders master problem is solved, the total computational time in seconds of the second phase, and the number of nodes explored by the algorithm, respectively. A dash "—" in the *Gap* column indicates that the optimal integral solution of an instance cannot be obtained by the algorithm in the time limit of four hours, and a dash "—" in *Best UB* the column indicates that a feasible solution cannot be found by the algorithm.

Table 3. Characteristics of Instances in the ER-DARP Data Set

					$l_i - e_i,\ i \in P$				$t_{i,i+n},\ i \in P$			$l_\psi - e_\psi,\ \psi \in \Psi_V$				$l_\psi - e_\psi,\ \psi \in \Psi_A$				
Instance	n	$\lvert K\rvert$	$\lvert H\rvert$	$p_{o>1}$	Min	Avg.	Max	Avg. r_i	Min	Avg.	Max	Min	Avg.	Max	Var	Min	Avg.	Max	Var	α
26-2-a	26	2	3	0.12	60	226.15	720	34.50	6	10.42	24	510	510.00	510	0.00	510	540.00	600	42.43	1,117
26-2-b	26	2	3	0.15	60	212.31	720	34.04	5	10.19	17	510	510.00	510	0.00	510	540.00	600	42.43	998
27-2-a	27	2	3	0.19	20	257.41	720	33.67	5	10.37	21	510	510.00	510	0.00	510	540.00	600	42.43	978
27-2-b	27	2	3	0.15	60	264.44	720	34.00	7	10.85	22	510	510.00	510	0.00	510	540.00	600	42.43	1,250
28-2-a	28	2	3	0.11	20	250.36	720	34.18	3	9.86	24	510	510.00	510	0.00	510	540.00	600	42.43	1,281
28-2-b	28	2	3	0.11	60	229.29	720	31.39	6	9.25	15	510	510.00	510	0.00	510	540.00	600	42.43	881
29-2-a	29	2	3	0.14	60	281.38	720	32.28	4	9.45	15	510	510.00	510	0.00	510	540.00	600	42.43	944
29-2-b	29	2	3	0.14	30	262.76	720	32.90	4	9.90	14	510	510.00	510	0.00	510	540.00	600	42.43	996
30-2-a	30	2	3	0.17	60	211.00	720	35.10	3	10.20	16	510	510.00	510	0.00	510	540.00	600	42.43	1,124
30-2-b	30	2	3	0.13	20	218.67	720	33.80	3	9.83	16	510	510.00	510	0.00	510	540.00	600	42.43	1,064
31-2-a	31	2	3	0.16	60	271.94	720	33.48	4	10.06	21	510	510.00	510	0.00	510	540.00	600	42.43	1,224
31-2-b	31	2	3	0.13	60	229.35	720	35.03	4	9.74	23	510	510.00	510	0.00	510	540.00	600	42.43	1,336
32-2-a	32	2	3	0.13	20	247.19	720	33.94	5	10.56	15	510	510.00	510	0.00	510	540.00	600	42.43	1,175
32-2-b	32	2	3	0.13	60	241.88	720	34.22	4	10.53	15	510	510.00	510	0.00	510	540.00	600	42.43	1,069
33-2-a	33	2	3	0.12	60	266.36	720	33.55	5	10.27	15	510	510.00	510	0.00	510	540.00	600	42.43	1,207
33-2-b	33	2	3	0.12	20	206.06	720	34.18	4	10.70	31	510	510.00	510	0.00	510	540.00	600	42.43	1,641
34-2-a	34	2	3	0.12	60	204.71	720	34.15	4	10.53	17	510	510.00	510	0.00	510	540.00	600	42.43	1,191
34-2-b	34	2	3	0.15	60	180.00	720	35.65	5	11.12	23	510	510.00	510	0.00	510	540.00	600	42.43	1,370
35-2-a	35	2	3	0.11	60	250.29	720	33.77	4	10.66	15	510	510.00	510	0.00	510	540.00	600	42.43	1,246
35-2-b	35	2	3	0.11	20	191.71	720	34.80	3	10.77	16	510	510.00	510	0.00	510	540.00	600	42.43	1,204
31-3-a	31	3	4	0.19	30	273.87	720	33.97	4	10.26	16	510	540.00	600	42.43	450	517.50	600	53.56	1,220
31-3-b	31	3	4	0.13	30	243.87	720	33.00	4	10.00	14	510	540.00	600	42.43	450	517.50	600	53.56	1,108
31-3-c	31	3	4	0.23	30	238.06	720	34.35	5	10.23	29	510	540.00	600	42.43	450	517.50	600	53.56	1,238
32-3-a	32	3	4	0.19	60	231.56	720	35.63	5	11.34	20	510	540.00	600	42.43	450	517.50	600	53.56	1,553
32-3-b	32	3	4	0.16	20	237.50	720	33.66	3	9.56	19	450	520.00	600	61.64	450	517.50	600	53.56	1,412
32-3-c	32	3	4	0.13	30	224.06	720	32.91	4	9.78	15	510	540.00	600	42.43	450	517.50	600	53.56	1,167
33-3-a	33	3	4	0.15	60	260.91	720	36.18	4	11.30	20	510	540.00	600	42.43	450	517.50	600	53.56	1,410
33-3-b	33	3	4	0.18	30	271.82	720	35.82	4	11.48	31	510	540.00	600	42.43	450	517.50	600	53.56	2,028
33-3-c	33	3	4	0.15	20	235.15	720	33.55	4	10.42	14	510	540.00	600	42.43	450	517.50	600	53.56	1,261
34-3-a	34	3	4	0.18	60	233.82	720	35.29	5	11.15	20	510	540.00	600	42.43	450	517.50	600	53.56	1,658
34-3-b	34	3	4	0.12	20	204.12	720	36.35	3	11.21	22	510	540.00	600	42.43	450	517.50	600	53.56	1,669
34-3-c	34	3	4	0.15	20	171.76	720	33.88	3	10.38	35	450	520.00	600	61.64	450	517.50	600	53.56	2,115
35-3-a	35	3	4	0.14	30	244.29	720	34.54	4	10.40	20	510	540.00	600	42.43	450	517.50	600	53.56	1,467
35-3-b	35	3	4	0.11	20	202.00	720	34.20	5	10.66	15	510	540.00	600	42.43	450	517.50	600	53.56	1,354
35-3-c	35	3	4	0.14	60	208.29	720	32.74	3	9.31	14	510	540.00	600	42.43	450	517.50	600	53.56	1,287
36-3-a	36	3	4	0.14	30	192.50	720	36.92	2	12.03	36	510	540.00	600	42.43	450	517.50	600	53.56	2,613
36-3-b	36	3	4	0.11	30	225.83	720	36.00	3	11.11	39	450	520.00	600	61.64	450	517.50	600	53.56	1,777
36-3-c	36	3	4	0.14	30	224.17	720	33.25	4	10.00	25	510	540.00	600	42.43	450	517.50	600	53.56	1,990
37-3-a	37	3	4	0.11	60	231.89	720	33.49	3	9.81	24	510	540.00	600	42.43	450	517.50	600	53.56	1,688
37-3-b	37	3	4	0.16	20	184.59	720	33.41	5	9.89	15	510	540.00	600	42.43	450	517.50	600	53.56	1,347
37-3-c	37	3	4	0.14	20	191.84	720	34.05	3	10.68	39	450	520.00	600	61.64	450	517.50	600	53.56	1,607

From Table 4, we can make the following observations. First, out of the 41 test instances, the proposed algorithm successfully solved 27 instances to optimality. Among the 27 instances optimally solved, 17 instances have more than 30 requests, and the largest instance has as many as 36 requests. Second, the first phase of the algorithm accounts for only a small proportion of the computational time, which demonstrates the efficiency of the label-setting algorithm in the first phase. Third, the algorithm solves the LP relaxation of the Benders master problem to optimality very frequently. The number of times that the Benders master problem is solved is 2,237 on average while the average number of nodes explored is only 39. This suggests that the column and cut generation procedure (Algorithm 1) converges rather slowly to optimality and is the bottleneck of the overall algorithm. Finally, the number of requests satisfied in the optimal solutions of the instances with three vehicles is much higher than those in the instances with two vehicles. Moreover, in the optimal solutions of the instances with three vehicles, almost 100% of the requests are satisfied except for instance *36-3-a* with 94.44%. This indicates that two vehicles are not enough to satisfy all of the requests in most of the ER-DARP instances, but three vehicles is sufficient.

5.3. Effects of Manpower Resources

In this section, we analyze the effects of manpower on the performance of the algorithm and the optimal

Table 4. Results for the ER-DARP Data Set

	First phase		Second phase													
			LB_0							Best UB						
Instance	Trips	Time	Objective	Requests (%)	Distance	Gap (%)	LS time	BMP solved	Time	Objective	Requests (%)	Distance	LS time	BMP solved	Total time	Nodes
26-2-a	38,601	1	−27,571.5	96.15	353.5	0.01	1	2	1	**−27,570**	96.15	355	2	4	2	3
26-2-b	56,718	1	−25,599.0	100.00	349.0	0.00	1	1	1	**−25,599**	100.00	349	1	1	1	1
27-2-a	180,749	8	−25,140.0	96.30	288.0	0.00	110	1	111	**−25,140**	96.30	288	111	1	111	1
27-2-b	219,652	10	−32,212.0	96.30	288.0	0.02	20	51	36	**−32,205**	96.30	295	288	1,816	616	59
28-2-a	102,361	3	−35,472.1	100.00	395.9	0.00	394	100	468	**−35,471**	100.00	397	410	102	484	3
28-2-b	69,159	1	−22,569.0	92.86	337.0	0.00	5	3	5	**−22,569**	92.86	337	5	3	5	1
29-2-a	332,447	22	−27,083.7	100.00	292.3	0.01	84	1	86	**−27,081**	100.00	295	1,409	1,135	2,972	37
29-2-b	186,861	8	−26,630.6	93.10	261.4	0.01	88	113	168	**−26,627**	93.10	265	123	161	214	5
30-2-a	189,182	9	−33,375.9	100.00	344.1	0.01	299	253	465	**−33,373**	100.00	347	1,189	728	1,567	37
30-2-b	171,946	5	−30,513.5	96.67	342.5	0.00	486	61	530	**−30,513**	96.67	343	549	97	607	5
31-2-a	471,338	50	−36,395.0	96.77	325.0	0.00	6,341	7	6,345	**−36,395**	96.77	325	6,342	7	6,345	1
31-2-b	263,059	11	−34,431.5	83.87	304.5	0.01	342	180	488	**−34,428**	83.87	308	1,170	675	2,076	5
32-2-a	335,027	47	−34,906.0	93.75	344.0	0.02	555	1	557	**−34,900**	93.75	350	6,457	1,934	7,166	491
32-2-b	330,980	24	−32,799.0	96.88	340.0	0.00	156	1	158	**−32,799**	96.88	340	157	1	158	1
33-2-a	339,226	20	−34,664.3	87.88	338.7	0.02	1,342	4	1,343	**−34,657**	87.88	346	6,092	12	6,099	9
33-2-b	351,895	20	−44,023.5	81.82	283.5	0.01	7	1	8	**−44,019**	81.82	288	92	38	98	33
34-2-a	231,911	10	−35,817.4	89.32	351.4	1.20	42	1	43	**−35,394**	88.24	336	1,772	630	2,229	43
34-2-b	100,887	3	−37,727.6	81.81	378.7	—	307	662	433	−32,555	70.59	325	1,764	2,420	14,418	6
35-2-a	528,103	38	−38,301.0	88.57	325.0	0.02	447	1	450	**−38,294**	88.57	332	4,529	32	4,557	29
35-2-b	164,657	5	−34,190.5	82.02	374.4	—	9	1	10	−33,338	80.00	374	7,098	2,965	14,402	44
31-3-a	400,910	18	−37,470.5	100.00	349.5	0.01	221	95	306	**−37,468**	100.00	352	626	672	1,247	15
31-3-b	375,352	31	−34,050.1	100.00	297.9	0.02	189	107	306	**−34,044**	100.00	304	948	3,123	2,729	85
31-3-c	191,386	10	−37,970.5	100.00	407.5	0.01	34	33	39	**−37,966**	100.00	412	596	342	711	27
32-3-a	251,468	19	−49,316.1	100.00	379.9	0.00	124	25	135	**−49,314**	100.00	382	894	1,069	1,176	57
32-3-b	435,581	51	−44,847.7	100.00	336.3	0.01	86	7	89	**−44,841**	100.00	343	4,365	1,549	5,421	141
32-3-c	229,467	11	−37,036.1	100.00	307.9	0.01	43	28	54	**−37,033**	100.00	311	863	1,445	2,178	21
33-3-a	353,068	57	−46,201.4	100.00	328.6	—	798	133	963	−41,993	90.91	307	10,474	3,161	14,401	36
33-3-b	739,697	106	−66,516.0	100.00	408.0	0.00	404	25	423	**−66,516**	100.00	408	405	25	423	1
33-3-c	387,637	31	−41,279.6	100.00	333.4	—	118	1,071	413	−41,258	100.00	355	1,767	12,247	14,401	13
34-3-a	365,444	32	−55,979.6	100.00	392.4	—	458	123	528	−44,426	79.41	340	1,056	5,244	14,402	11
34-3-b	312,912	19	−56,337.6	100.00	408.4	0.01	11	1	13	**−56,330**	100.00	416	700	9,769	6,297	261
34-3-c	158,550	7	−71,479.7	100.00	430.3	—	10	154	128	−58,899	82.35	321	182	3,427	14,402	11
35-3-a	750,803	67	−50,994.9	100.00	350.1	—	1,161	1,829	9,134	−46,634	91.43	310	1,924	3,780	14,401	3
35-3-b	287,676	23	−47,034.4	100.00	355.6	—	72	763	871	−34,939	74.29	265	280	3,238	14,405	3
35-3-c	313,668	20	−44,643.5	100.00	401.5	—	51	286	280	−38,276	85.71	334	747	3,811	14,407	8
36-3-a	150,141	5	−88,662.7	94.77	486.7	0.33	54	3	58	**−88,371**	94.44	471	765	3,144	9,021	67
36-3-b	265,309	17	−63,467.2	100.00	504.8	—	82	248	242	−51,129	80.56	404	1,337	6,014	14,407	19
36-3-c	513,096	33	−71,287.3	100.00	352.7	—	421	796	3,239	−51,485	72.22	255	1,957	4,053	14,403	5
37-3-a	324,514	20	−62,042.4	100.00	413.6	—	253	647	1,459	—	—	—	2,053	5,106	14,401	9
37-3-b	317,165	20	−49,461.6	100.00	377.4	—	758	483	1,431	−29,384	59.46	250	3,184	4,615	14,401	5
37-3-c	393,040	45	−58,967.6	100.00	491.4	—	1,185	1,182	4,217	−35,094	59.46	260	7,339	3,135	14,402	4
Average	297,113	23	−43,035.8	96.31	359.3	0.06	429	231	879	−39,208	90.25	335	2,001	2,237	6,492	39

Table 5. Effects of Manpower

$\Delta\|K\|$	$\Delta\|H\|$	Avg. $\|K\|$	Avg. $\|H\|$	Avg. $\|H\|/\|K\|$	Opt solved	Objective	Requests (%)	Distance	LS time	Total time	Nodes	BMP solved
0	0	2.51	3.51	1.41	27	−38,108.0	95.69	344.3	2,001	6,492	39	2,237
−1	−1	1.51	2.51	1.74	17	−14,029.8	52.76	150.1	7,219	8,797	2	347
+1	0	3.50	3.50	1.00	3	−38,595.3	100.00	330.3	2,758	13,760	15	3,205
0	+1	2.51	4.51	1.83	38	−42,597.4	96.19	354.8	2,532	2,872	244	762
+1	+1	3.51	4.51	1.29	18	−50,011.9	99.69	372.2	2,240	8,827	43	2,046
+2	+2	4.51	5.51	1.22	27	−46,156.6	100.00	367.1	2,666	6,511	84	1,878

solutions of the instances. We conduct another five experiments with different adjustments for the numbers of vehicles and the assistants in the instances. The computational results are summarized in Table 5. The fist two columns give the settings of each experiment. To be more specific, columns $\Delta|K|$ and $\Delta|H|$ give the deviations of the numbers of vehicles and assistants from the default values in each experiment, respectively. For example, the first row in the table gives the results of the default setting, and the second row gives the results when the numbers of vehicles and assistants in each instance are both reduced by one. Columns *Avg.* $|K|$ and *Avg.* $|H|$ give the average number of vehicles and the average number of assistants, respectively. Column *Avg.* $|H|/|K|$ gives the average ratio between the number of assistants and the number of vehicles. Column *Opt solved* gives the number of instances that are optimally solved by the algorithm. Columns *Objective*, *Requests*, and *Distance* give the average information of the optimal solutions. Columns *LS time* and *Total time* give the average computational times taken by the label-setting algorithm and the overall algorithm in the second phase of all of the instances, respectively. Columns *Nodes* and *BMP solved* give the average number of nodes explored and the average number of Benders master problem (Model (IV)) solved by the algorithm of all of the instances, respectively. The detailed results of the optimal solutions are presented in Online Appendix G.

From these results, we make the following observations. First, the number of instances optimally solved increases only if the number of assistants increases. This indicates that the instances become much easier to solve when the number of available assistants increases. To be more general, the ratio between the number of assistants and the number of vehicles, that is, $|H|/|K|$, is a very important factor of influence to the problem complexity. Except for the two cases in which $\Delta|K| = -1$, $\Delta|H| = -1$ and $\Delta|K| = +2$, $\Delta|H| = +2$, the number of instances optimally solved increases with the increase of this ratio. This can be easily explained because the restrictions of the synchronization constraints brought by the assistants become progressively weaker with the increase of the ratio, and thus Algorithm 1 takes fewer iterations to finally solve the Benders master problem (Model (IV)). Second, in the case in which $\Delta|K| = -1$ and $\Delta|H| = -1$, the percentage of requests satisfied in the optimal solutions dramatically decreases to about 52%, which suggests that both the vehicles and the assistants are in serious shortage. Moreover, the algorithm spends a large proportion of computational time on the label-setting algorithm in this case: 7,219 seconds on average. Therefore, despite the increase in the ratio $|H|/|K|$, the number of instances optimally still decreases in this case. Third, after the number of vehicles is increased by at least one, the percentage of clients satisfied in the optimal solutions is very close to 100%. This indicates that the number of vehicles becomes sufficient for most of the instances after one more vehicle is added. Therefore, in the case in which $\Delta|K| = 2$ and $\Delta|H| = 2$, the vehicles can be in surplus for most of the instances, which makes the ratio $|H|/|K|$ smaller than its real value. This provides a possible explanation as to why the number of instances optimally solved increases despite the decrease in the ratio $|H|/|K|$ in this case. From these observations, we can conclude that the performance of the proposed algorithm is heavily influenced by the complexity of the manpower planning part of the problem.

6. Conclusion

In this paper, we investigate a practical DARP variant arising in patient transportation. The problem, referred to as *ER-DARP*, is an extension of the R-DARP recently proposed by Liu, Luo, and Lim (2015). We formulate the problem into a compact arc-flow and strong trip-based model and propose a two-phase, branch-and-price-and-cut algorithm to exactly solve the problem. In the first phase, we use a label-setting algorithm to enumerate the set of nondominated trips to initialize the trip-based model. In the second phase, we propose a branch-and-price-and-cut algorithm based on the Benders decomposition to solve the trip-based model exactly. The two-phase algorithm is tested on the R-DARP benchmark instances and a set of test instances is also generated according to the NEATS data set. The proposed algorithm quickly solved all of the R-DARP instances to optimality and outperformed the branch-and-cut algorithm proposed by Liu, Luo, and Lim (2015) for the R-DARP. For the new test instances, our algorithm successfully solved instances with up to 36 requests to optimality in four hours. Finally, we conduct additional experiments to analyze the effects of manpower resources on the proposed algorithm and the structure of the optimal solutions.

Our two-phase exact algorithm performs better than the existing exact algorithm on the R-DARP and successfully solves medium-sized instances. However, there is still room for improvement, especially in the column- and cut-generation procedure. It still takes many iterations to finally solve the Benders master problem. In the future, we will strive to propose a more powerful column- and cut-generation procedure to reduce the number of iterations to solve the Benders master problem in the hope that the overall algorithm can solve practical-sized instances.

References

Ascheuer N, Fischetti M, Grötschel M (2000) A polyhedral study of the asymmetric traveling salesman problem with time windows. *Networks* 36(2):69–79.

Baldacci R, Mingozzi A, Roberti R (2011) New route relaxation and pricing strategies for the vehicle routing problem. *Oper. Res.* 59(5):1269–1283.

Boland N, Dethridge J, Dumitrescu I (2006) Accelerated label setting algorithms for the elementary resource constrained shortest path problem. *Oper. Res. Lett.* 34(1):58–68.

Braekers K, Caris A, Janssens GK (2014) Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Res. Part B: Methodological* 67: 166–186.

Cordeau J-F (2006) A branch-and-cut algorithm for the dial-a-ride problem. *Oper. Res.* 54(3):573–586.

Cordeau J-F, Laporte G (2007) The dial-a-ride problem: Models and algorithms. *Ann. Oper. Res.* 153(1):29–46.

Cordeau J-F, Stojković G, Soumis F, Desrosiers J (2001) Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Sci.* 35(4):375–388.

de Sá EM, de Camargo RS, de Miranda G (2013) An improved Benders decomposition algorithm for the tree of hubs location problem. *Eur. J. Oper. Res.* 226(2):185–202.

Desrosiers J, Dumas Y, Soumis F (1986) A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. *Amer. J. Math. Management Sci.* 6(3–4): 301–325.

Dumas Y, Desrosiers J, Soumis F (1991) The pickup and delivery problem with time windows. *Eur. J. Oper. Res.* 54(1):7–22.

Gschwind T, Irnich S (2015) Effective handling of dynamic time windows and its application to solving the dial-a-ride problem. *Transportation Sci.* 49(2):335–354.

Lim A, Zhang Z, Qin H (2017) Pickup and delivery service with manpower planning in Hong Kong public hospitals. *Transportation Sci.* 51(2):688–705.

Liu M, Luo Z, Lim A (2015) A branch-and-cut algorithm for a realistic dial-a-ride problem. *Transporation Res. Part B: Methodological* 81(1):267–288.

Madsen OBG, Ravn HF, Rygaard JM (1995) A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Ann. Oper. Res.* 60(1):193–208.

Magnanti TL, Wong RT (1981) Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Oper. Res.* 29(3):464–484.

Melachrinoudis E, Ilhan AB, Min H (2007) A dial-a-ride problem for client transportation in a health-care organization. *Comput. Oper. Res.* 34(3):742–759.

Mercier A, Cordeau J-F, Soumis F (2005) A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. *Comput. Oper. Res.* 32(6):1451–1476.

Parragh SN (2011) Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. *Transportation Res. Part C: Emerging Tech.* 19(5):912–930.

Parragh SN, Cordeau JF, Doerner KF, Hartl RF (2012) Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints. *OR Spectrum* 34(3):593–633.

Psaraftis HN (1980) A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Sci.* 14(2):130–154.

Psaraftis HN (1983) An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. *Transportation Sci.* 17(3):351–357.

Qu Y, Bard JF (2013) The heterogeneous pickup and delivery problem with configurable vehicle capacity. *Transportation Res. Part C: Emerging Tech.* 32:1–20.

Qu Y, Bard JF (2014) A branch-and-price-and-cut algorithm for heterogeneous pickup and delivery problems with configurable vehicle capacity. *Transportation Sci.* 49(2):254–270.

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.

Righini G, Salani M (2008) New dynamic programming algorithms for the resource constrained elementary shortest path problem. *Networks* 51(3):155–170.

Ropke S, Cordeau J-F, Laporte G (2007) Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks* 49(4):258–272.

Savelsbergh M, Sol M (1998) Drive: Dynamic routing of independent vehicles. *Oper. Res.* 46(4):474–490.

Toth P, Vigo D (1996) Fast local search algorithms for the handicapped persons transportation problem. Osman IH, Kelly JP, eds. *Meta-Heuristics* (Springer, Boston), 677–690.

Toth P, Vigo D (1997) Heuristic algorithms for the handicapped persons transportation problem. *Transportation Sci.* 31(1):60–71.