Transportation Research Part B 81 (2015) 267–288

Contents lists available at ScienceDirect

Transportation Research Part B

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

# A branch-and-cut algorithm for a realistic dial-a-ride problem

Mengyang Liu [b], Zhixing Luo [a,*], Andrew Lim [c,a,b,1]

[a] International Center of Management Science and Engineering, School of Management and Engineering, Nanjing University, Nanjing 210093, PR China
[b] Department of Management Sciences, City University of Hong Kong, Tat Chee Ave, Kowloon Tong, Hong Kong
[c] School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371, Singapore

ARTICLE INFO

*Article history:*
Received 13 October 2014
Received in revised form 14 May 2015
Accepted 14 May 2015
Available online 6 June 2015

*Keywords:*
Dial-a-ride problem
Manpower planning
Branch-and-cut
Healthcare transportation

ABSTRACT

In this paper we study a realistic dial-a-ride problem which simultaneously considers multiple trips, heterogeneous vehicles, multiple request types, configurable vehicle capacity and manpower planning. All of these features originate from practical applications in recent years. To formulate the problem, we propose two mathematical models that use different methods to deal with requests associated with the depot. To further strengthen the models, we propose eight families of valid inequalities, and based on them, we propose a branch-and-cut algorithm to solve the problem. The branch-and-cut algorithm was extensively tested on a set of instances generated according to the data of a real world application. The computational results showed that seven families of inequalities can improve the lower bounds substantially and the branch-and-cut algorithm can solve instances with up to 22 requests within 4 h.



## 1. Introduction

The classic dial-a-ride problem (DARP) mainly arises in door-to-door transportation services for elderly or disabled people (Madsen et al., 1995; Toth and Vigo, 1996, 1997; Melachrinoudis et al., 2007). Given a set of transportation requests, each of which involves transporting a client from an origin to a destination, the DARP consists of determining a set of trips for a fleet of vehicles to satisfy the requests, subject to a series of side constraints, like time window constraints and maximum riding time constraints of clients, and capacity constraints and maximum trip duration constraints of vehicles. The time window constraints of a client require the pickup and drop-off of the client to be within given time intervals, while the maximum riding time constraints restrict the riding time of the client. The capacity constraints and the maximum duration constraints of a vehicle ensure that the load and the trip duration of the vehicle cannot exceed its capacity and a given limit, respectively. The objective of the DARP can be maximizing the number of requests satisfied, minimizing the number of vehicles used, or minimizing the total travel distance of the vehicles.

In recent years, a trend of research on the DARP is to take more realistic constraints into consideration to make the problem more practical. Parragh (2011) and Parragh et al. (2012), motivated by observations made at Austrian Red Cross in the field of patient transportation, considered heterogeneous vehicles and clients in the DARP. In their problem, clients have requests for multiple types of facilities like seats, wheelchairs, and stretchers and the vehicles have different capacities for each type of facility. Qu and Bard (2013, 2014) studied a pickup and delivery problem (PDP) derived from a real application associated with daily route planning for the All-Inclusive Care Program for the Elderly organization. It considers not

* Corresponding author.
*E-mail addresses:* mengyliu2-c@my.cityu.edu.hk (M. Liu), luozx@nju.edu.cn (Z. Luo), lim.andrew@cityu.edu.hk (A. Lim).
[1] Currently on no pay leave from City University of Hong Kong.

http://dx.doi.org/10.1016/j.trb.2015.05.009
0191-2615/

only heterogeneous vehicles and clients but also the configurability of the vehicles' capacities. In their problem, the vehicles have multiple configurations for seats and wheelchairs, which can be adjusted according to the destined clients before leaving the depot. Lim et al. (in press) studied another realistic DARP variant originating from the Non-Emergency Ambulance Transfer Service (NEATS) in Hong Kong. In this new variant, some clients may require several assistants to help her/him to get on/off a vehicle. These clients are mainly patients or the disabled who live in old buildings without elevators. Therefore, before a vehicle departs from the depot, there must be sufficient assistants on board the vehicle for the destined requests. An assistant on the vehicle occupies one seat. In addition, due to a strict limit on the trip duration, each vehicle performs several trips per day. The assistants can move freely from a trip of a vehicle to a trip of another vehicle given that the two trips do not cause time conflicts for the assistants. Therefore, the problem faced in the NEATS involves a complicated manpower planning problem (see Fig. 1). To deal with this problem, Lim et al. (in press) proposed an efficient heuristic with an ad-hoc component to handle the manpower planning problem. Zhang et al. (2015) proposed a memetic algorithm to solve a simplified version of the problem proposed by Lim et al. (in press) without manpower planning.

In this paper, we study a realistic variant of the DARP, referred to as the *R-DARP*, which simultaneously considers multiple trips, heterogeneous vehicles, multiple request types, configurable vehicle capacity and manpower planning. The R-DARP differs from the one studied by Lim et al. (in press) in the following major aspects. First, Lim et al. (in press) treats the demands of the requests in term of seats, instead of different types of facilities. For example, a client demanding a wheelchair (or stretcher) will be treated as demanding 1.5 (or 3) seats. This is unrealistic, because a vehicle cannot convert 1.5 (or 3) seats into a wheelchair (or stretcher) during the routing process. The configuration of vehicle capacity should be done before a vehicle departs from the depot. Second, the objective of the problem studied by Lim et al. (in press) is a linear combination of three parts: the number of requests satisfied, the total travel distance of vehicles and the workload balance of the staff, while in the R-DARP we follow the most common objective in the literature, namely minimizing the total travel distance of vehicles. Third, in the R-DARP we consider the maximum riding time constraints of the clients, because we think it is important to maintain a consistent and high service level in the public service sector, while Lim et al. (in press) ignore the maximum riding time constraints.

The contributions of the paper are summarized as follows. First, we introduce a realistic and practical variant of the DARP which simultaneously considers several new constraints appearing in recent years. Second, we formulate the problem into two mixed integer programming (MIP) models that use different methods to handle requests whose pickup or delivery points correspond to the depot. Third, we introduce eight families of valid inequalities to tighten the proposed models, and devise a branch-and-cut algorithm to optimally solve the proposed models. Last, we test the branch-and-cut algorithm on a set of instances generated from the NEATS data. Our computational results show that the eight families of valid inequalities can substantially improve the quality of lower bounds, and the proposed branch-and-cut algorithm can solve instances with up to 22 requests to optimality within 4 h. The proposed algorithm and computational results can serve as reference for future research on this problem.

The remainder of this paper is organized as follows. In Section 2, we conduct a literature review on the solution approaches for the DARP as well as the pickup and delivery problem with time windows (PDPTW), a problem similar to the DARP. In Section 3, we describe the two MIP models in detail. The eight families of inequalities and the details of the branch-and-cut algorithm are presented in Sections 4 and 5, respectively. Section 6 is devoted to the computational experiments, followed by Section 7, which concludes this paper with some closing remarks.

## 2. Literature review

In the past decades, many variants of the DARP have been proposed, like the dynamic DARP (Colorni and Righini, 2001; Attanasio et al., 2004; Coslovich et al., 2006), the single vehicle DARP (Psaraftis, 1980, 1983; Desrosiers et al., 1986), and the multi-vehicle DARP (Jaw et al., 1986; Cordeau and Laporte, 2003b). For a detail survey of the DARP, please see Cordeau and Laporte (2003a, 2007) and Parragh et al. (2008). Among all of these variants, the multi-vehicle DARP is the most relevant to the R-DARP, so we mainly review the literature on the multi-vehicle DARP. Solution approaches for the multi-vehicle DARP

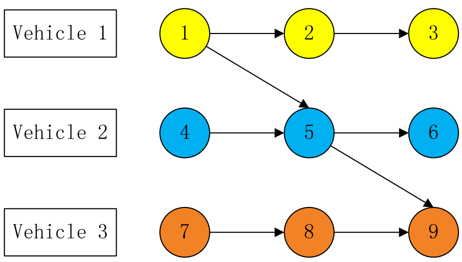


**Fig. 1.** An example of manpower planning. Vehicle 1 performs trip 1–3; vehicle 2 performs trip 4–6; and vehicle 3 performs trip 7–9. When an assistant is assigned to trip 1, 5, and 9, vehicle 1–3 must collaborate to avoid time conflicts for this assistant.

can be basically divided into two classes: heuristics and exact algorithms. According to the purpose of a heuristic, heuristics can be roughly classified into construction heuristics and improvement heuristics. Construction heuristics are able to generate a solution rapidly. They can provide an initial solution (Toth and Vigo, 1997; Aldaihani and Dessouky, 2003) for the improvement heuristics, or applied in the instances when hundreds of requests are involved and a solution is needed in seconds (Jaw et al., 1986; Diana and Dessouky, 2004; Luo and Schonfeld, 2007). Improvement heuristics, on the other hand, usually attempt to improve the quality of a solution by a local search procedure involving relocation or exchange of vertices or clients (Toth and Vigo, 1997; Cordeau and Laporte, 2003b; Xiang et al., 2006; Kirchler and Wolfler Calvo, 2013; Paquette et al., 2013). To date, the deterministic annealing algorithm proposed by Braekers et al. (2014) has been the best heuristic for the homogeneous and heterogeneous DARP, both in terms of solution quality and computation time.

Exact algorithms for the DARP are mainly branch-and-cut algorithms and branch-and-price algorithms. The first branch-and-cut algorithm for the DARP was devised by Cordeau, 2006, who formulated the problem into a three-index MIP model, and further strengthened the model by several families of valid inequalities derived from the well-known inequalities in the travelling salesman problem (TSP) and the vehicle routing problem (VRP). Ropke et al. (2007) proposed a branch-and-cut algorithm for the PDPTW based on a two-index MIP model and six families of valid inequalities, and applied it to solve the DARP successfully. Parragh (2011) developed another branch-and-cut algorithm based on the work of Cordeau (2006) and Ropke et al. (2007) to solve a variant of the DARP that considers heterogeneous clients and vehicles. Recently, a DARP variant with multi-depot and heterogeneous vehicles was solved by a branch-and-cut algorithm proposed by Braekers et al. (2014), which is also based on the work of Cordeau (2006) and Ropke et al. (2007). A branch-and-price algorithm was proposed by Parragh et al. (2012) to solve a variant of DARP where clients and vehicles are heterogeneous and the maximum riding time constraints of clients are implicitly enforced by artificially constructing time windows for certain nodes. Such an implementation cannot truly enforce the maximum riding constraints because the riding time of a client may still exceed the given limit. Recently, Gschwind and Irnich (2014) proposed a branch-and-price-and-cut algorithm that can truly handle the maximum riding constraints. The algorithm outperforms the branch-and-cut algorithm proposed by Cordeau (2006) and Ropke et al. (2007), and has been the best exact algorithm for the DARP to date.

As we mentioned above, the PDPTW (Dumas et al., 1991; Savelsbergh and Sol, 1995; Parragh et al., 2008) is a problem similar to the DARP. Because the targets of the PDPTW is freight instead of human beings, client-centric or service-based constraints, i.e. the maximum riding constraints, are absent in the PDPTW. Heuristics that have been successfully applied to solve the PDPTW include tabu search (Nanry and Wesley Barnes, 2000), genetic algorithms (Jung and Haghani, 2000; Pankratz, 2005), simulated annealing (Li and Lim, 2003), adaptive large neighborhood search (Ropke and Pisinger, 2006), and a hybrid algorithm combining simulated annealing and large neighborhood search (Bent and Hentenryck, 2006). Exact algorithms for the PDPTW include branch-and-price algorithms (Dumas et al., 1991; Savelsbergh and Sol, 1998; Ropke and Cordeau, 2009), branch-and-cut algorithms (Ropke et al., 2007), and a hybrid exact algorithm with two dual ascent heuristics and a cut-and-column generation procedure (Baldacci et al., 2011).

## 3. Mathematical models

### 3.1. Notation

Let $n$ denote the number of transportation requests. The R-DARP is defined on a complete graph $G = (V, A)$, where $V = \{0, \ldots, 2n\}$ is the node set and $A = \{(i,j) | i,j \in V, i \neq j\}$ is the arc set. Node 0 is the depot. $P = \{1, \ldots, n\}$ and $D = \{n+1, \ldots, 2n\}$ are two subsets of $V$ that denote the set of pickup nodes and the set of delivery nodes, respectively. Let $N = P \cup D$ denote the set of nodes except the depot. A transportation request involves transporting a client from his/her pickup node $i \in P$ to the delivery node $n+i \in D$. Each node $i \in N$ requires $q_i^1$ seats, $q_i^2$ wheelchairs and $q_i^3$ stretchers, and has a time window $[e_i, l_i]$ and a service time $s_i$. Here we ensure that $q_{n+i}^m = -q_i^m$ and $q_i^m \geqslant 0$ for all $m = 1, 2, 3$ and $i \in P$. Time window $[e_i, l_i]$ on node $i$ means that if a vehicle arrives at node $i$ before $e_i$, the vehicle must wait until $e_i$ to start service, and the service starting time on node $i$ cannot be later than $l_i$. Each arc $(i,j) \in A$ has a travel time $t_{i,j}$. We assume that all of the vehicles follow constant speed and $t_{i,j}$ satisfies the triangle inequality, i.e., $t_{i,j} \leqslant t_{i,k} + t_{k,j}$ for all $(i,j), (i,k), (k,j) \in A$. Client $i \in P$ has a maximum riding time $f_i$ and requires $o_i$ assistants to help him/her to get on/off a vehicle. The riding time of a client is defined to be the elapsed time between the end of service at his/her pickup node and the start of service at his/her delivery node.

Let $K = \{1, \ldots, |K|\}$ and $H = \{1, \ldots, |H|\}$ denote the set of vehicles (drivers) and assistants. Each staff member $i \in K \cup H$ has a work period $[e_i^s, l_i^s]$ and a lunch period $[e_i^l, l_i^l]$ ($e_i^s \leqslant e_i^l \leqslant l_i^l \leqslant l_i^s$) during which the staff member must stay in the depot. Therefore, the working time of staff member $i$ is divided into two segments, namely $[e_i^s, e_i^l]$ and $[l_i^l, l_i^s]$ (). Let $|R|$ denote the maximum number of trips that a staff member can perform in a workday and $R = \{1, \ldots, |R|\}$ denote the set of indices. In the remainder of this paper, we use notation trip $(k, r)$ $(k \in K, r \in R)$ to represent the $r$-th trip of vehicle $k$, and notation tour $(h, r)$ $(h \in H, r \in R)$ to represent the $r$-th tour of assistant $h$. Each trip has a maximum duration $L$. At the end of each trip, the staff must rest for $B$ hours, where $B$ is equal to 0.5 h. A proportion of the inner space of a vehicle is configurable, while the other space is allocated to seats. The configurable space in a vehicle is organized by divisions: each division can be configured to a place for either one stretcher, two wheelchairs or three seats. For a vehicle $k \in K$, let $Q_k$ denote the number of fixed seats, and $Q'_k$ denote the number of divisions.

To build MIP models for the problem, we first create a set of duplicates of the depot. Let $V_d = \{2n+1, 2n+|K||R|\}$ denote a set of duplicates of the depot. Then we create a new graph $G' = (V', A')$, where $V' = N \cup V_d$ is the node set and $A'$ is the arc set. For vehicle $k \in K$ and trip index $r \in R$, let nodes $\sigma^o_{k,r}$ and $\sigma^d_{k,r}$ denote the starting node and the ending node of trip $(k,r)$, respectively. Here node $\sigma^o_{k,r}$ is defined as node $2n+(k-1)|R|+r$, and node $\sigma^d_{k,r}$ is defined as node $2n+(k-1)|R|+r+1$ if $k<|K|$ or $r<|R|$, and node $2n+1$ otherwise. Arc set $A'$ is the combination of (i) the arcs between nodes in $N$, i.e., $\{(i,j)|i,j \in N, i \neq j\}$; (ii) the arcs from depot duplicates to the nodes in $P$, i.e., $\{(\sigma^o_{k,r}, i)|i \in P, k \in K, r \in R\}$; (iii) the arcs from the nodes in $D$ to the depot duplicates, i.e., $\{(i,\sigma^d_{k,r})|i \in D, k \in K, r \in R\}$; and (iv) the arcs between depot duplicates, i.e., $\{(i,i+1)|i = 2n+1, \ldots, 2n+|K||R|-1\} \bigcup \{(2n+|K||R|, 2n+1)\}$. The travel time between the depot duplicates is set to zero.

### 3.2. Model I

In this section, we introduce the first MIP model for the R-DARP. In this model, although some pickup or delivery points have the same physical location as the depot, these pickup or delivery points are treated as different nodes. The decision variables used in Model I are as follows:

- $x_{i,j}$: 1 if arc $(i,j) \in A'$ is used; 0 otherwise.
- $w_{i,k,r}$: 1 if $i \in P$ and client $i$ is served by trip $(k,r)$ or $i \in D$ and client $i-n$ is served by trip $(k,r)$ $(k \in K, r \in R)$; 0 otherwise.
- $a_i$: arrival time at node $i \in N$.
- $b^m_{k,r}$: the number of divisions that are configured into seats $(m=1)$, wheelchairs $(m=2)$ or stretchers $(m=3)$ in the $r$-th $(r \in R)$ trip of vehicle $k$.
- $d^m_i$: the number of seats $(m=1)$, wheelchairs $(m=2)$ or stretchers $(m=3)$ used in a vehicle after node $i \in N$.
- $u_{k,r}$: starting time of trip $(k,r)$ $(k \in K, r \in R)$.
- $u_{h,r}$: starting time of tour $(h,r)$ $(h \in H, r \in R)$.
- $v_{k,r}$: completion time of trip $(k,r)$ $(k \in K, r \in R)$.
- $v_{h,r}$: completion time of tour $(h,r)$ $(h \in H, r \in R)$.
- $y_{k,r}$: 1 if trip $(k,r)$ $(k \in K, r \in R)$ or tour $(k,r)$ $(k \in H, r \in R)$ is performed in time interval $[e^s_k, e^l_k]$, and 0 in time interval $[l^l_k, l^s_k]$.
- $y_{h,r}$: 1 if tour $(h,r)$ $(h \in H, r \in R)$ is performed in time interval $[e^s_k, e^l_k]$, and 0 in time interval $[l^l_k, l^s_k]$.
- $z_{h,r,k,p}$: 1 if tour $(h,r)$ $(h \in H, r \in R)$ coincides with trip $(k,p)$ $(k \in K, p \in R)$.

According to the definition of $z_{h,r,k,p}$, the number of assistants assigned to vehicle $k$ on its $p$-th trip can be represented as $\sum_{h \in H}\sum_{r \in R} z_{h,r,k,p}$. Model I of the R-DARP is formulated as follows:

$$\textbf{Model I}: \min \sum_{(i,j)\in A'} t_{i,j}x_{i,j} \tag{1}$$

$$\text{s.t.} \sum_{(i,j)\in A'} x_{i,j} = \sum_{(j,i)\in A'} x_{j,i} = 1, \quad \forall i \in V' \tag{2}$$

$$w_{j,k,r} \geqslant w_{i,k,r} + x_{i,j} - 1, \quad \forall i,j \in N, k \in K, r \in R \tag{3}$$

$$w_{i,k,r} \geqslant x_{\sigma^o_{k,r},i}, \quad \forall i \in N, k \in K, r \in R \tag{4}$$

$$w_{i,k,r} \geqslant x_{i,\sigma^d_{k,r}}, \quad \forall i \in N, k \in K, r \in R \tag{5}$$

$$\sum_{k \in K}\sum_{r \in R} w_{i,k,r} = 1, \quad \forall i \in P \tag{6}$$

$$w_{i,k,r} = w_{n+i,k,r}, \quad \forall i \in P, k \in K, r \in R \tag{7}$$

$$a_j \geqslant (a_i + s_i + t_{i,j})x_{i,j}, \quad \forall i,j \in N \tag{8}$$

$$a_i \geqslant (u_{k,r} + t_{\sigma^o_{k,r},i})x_{\sigma^o_{k,r},i}, \quad \forall i \in P, k \in K, r \in R \tag{9}$$

$$v_{k,r} \geqslant (a_i + s_i + t_{i,\sigma^d_{k,r}})x_{i,\sigma^d_{k,r}}, \quad \forall i \in D, k \in K, r \in R \tag{10}$$

$$e_i \leqslant a_i \leqslant l_i, \quad \forall i \in N \tag{11}$$

$$t_{i,i+n} \leqslant a_{n+i} - (a_i + s_i) \leqslant f_i, \quad \forall i \in P \tag{12}$$

$$d^m_j \geqslant (d^m_i + q^m_j)x_{i,j}, \quad \forall i,j \in N, m = 1,2,3 \tag{13}$$

$$\left(\sum_{h \in H}\sum_{r \in R} z_{h,r,k,p} + d^1_i\right) w_{i,k,p} \leqslant Q_k + 3b^1_{k,p}, \quad \forall i \in P, k \in K, p \in R \tag{14}$$

$$d^2_i w_{i,k,p} \leqslant 2b^2_{k,p}, \quad \forall i \in P, k \in K, p \in R \tag{15}$$

$$d^3_i w_{i,k,p} \leqslant b^3_{k,p}, \quad \forall i \in P, k \in K, p \in R \tag{16}$$

$$b^1_{k,p} + b^2_{k,p} + b^3_{k,p} \leqslant Q'_k, \quad \forall k \in K, p \in R \tag{17}$$

$$o_i w_{i,k,p} \leqslant \sum_{h \in H} \sum_{r \in R} z_{h,r,k,p}, \quad \forall i \in P, k \in K, p \in R \tag{18}$$

$$u_{k,r+1} \geqslant v_{k,r} + B\left(1 - x_{\sigma^o_{k,r}, \sigma^d_{k,r}}\right), \quad \forall k \in K, r = 1, \ldots, |R| - 1 \tag{19}$$

$$u_{h,r+1} \geqslant v_{h,r} + Bz_{h,r,k,p}, \quad \forall h \in H, r = 1, \ldots, |R| - 1, k \in K, p \in R \tag{20}$$

$$u_{k,p} \geqslant u_{h,r} z_{h,r,k,p}, \quad \forall h \in H, k \in K, p, r \in R \tag{21}$$

$$v_{h,r} \geqslant v_{k,p} z_{h,r,k,p}, \quad \forall h \in H, k \in K, p, r \in R \tag{22}$$

$$e^s_k y_{k,r} + l^l_k (1 - y_{k,r}) \leqslant u_{k,r} \leqslant v_{k,r} \leqslant e^l_k y_{k,r} + l^s_k (1 - y_{k,r}), \quad \forall k \in K \cup H, r \in R \tag{23}$$

$$v_{k,r} - u_{k,r} \leqslant L, \quad \forall k \in K, r \in R \tag{24}$$

$$\sum_{k \in K} \sum_{p \in R} z_{h,r,k,p} \leqslant 1, \quad \forall h \in H, r \in R \tag{25}$$

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

$$w_{i,k,r} \in \{0, 1\}, \quad \forall i \in N, k \in K, r \in R \tag{27}$$

$$y_{k,r} \in \{0, 1\}, \quad \forall k \in K \cup H, r \in R \tag{28}$$

$$z_{h,r,k,p} \in \{0, 1\}, \quad \forall k \in K, h \in H, r, p \in R \tag{29}$$

$$a_i \geqslant 0, \quad \forall i \in N \tag{30}$$

$$d^m_i \geqslant \max\{q^m_i, 0\}, \quad \forall i \in N, m = 1, 2, 3 \tag{31}$$

$$b^m_{k,r} \geqslant 0 \text{ and integer}, \quad \forall k \in K, r \in R, m = 1, 2, 3 \tag{32}$$

$$u_{k,r}, v_{k,r} \geqslant 0, \quad \forall k \in K \cup H, r \in R \tag{33}$$

The objective function (1) minimizes the total travel time of the vehicles. We have assumed that all of the vehicles follow the same constant speed, hence minimizing their total travel time is equivalent to minimizing their total travel distance. Constraints (2) are flow conservation constraints which ensure that all of the nodes in $V'$ are visited exactly once. Constraints (3) ensure that if an arc is selected, both the starting node and the ending node of this arc are served in the same trip. For instance, if arc $(i,j)$ $(i,j \in N)$ is selected and node $i$ is served in trip $(k,r)$, then $x_{i,j} = 1, w_{i,k,r} = 1$, and $w_{j,k,r}$ is forced to be one by constraints (3), which means node $j$ is also served by trip $(k,r)$. Constraints (4) and (5) ensure that the nodes directly linking to the starting node or the ending node of trip $(k,r)$ are labelled as being served by trip $(k,r)$. Constraints (6) and (7) are the pairing constraints that force the pickup node and the delivery node of a request to be visited in the same trip.

Constraints (8) guarantee the consistency of arrival time at each node in $N$. Similarly, constraints (9) and (10) ensure the consistency of the arrival time at the nodes that are directly linked to the starting or ending node of a trip, with the starting and completion time of the trip, respectively. Constraints (11) are the time window constraints. The precedence constraints and the maximum riding time constraints of a client are imposed by constraints (12).

Constraints (13) define the relationship of a vehicle's load between two consecutively visited nodes. The sufficiency of seats, wheelchairs and stretchers is enforced by constraints (14)–(16), respectively. Note that each assistant occupies one seat, and thus in constraints (14), the number of seats required in total is equal to the number of seats required by the clients plus the number of seats required by the assistants. The capacity constraints for the divisions are enforced by constraints (17). Constraints (18) are the minimum assistant requirement constraints, which ensure that the number of assistants on each trip is enough to satisfy all of the destined requests.

Constraints (19) and (20) enforce the consistency between the completion time of a trip or tour and the starting time of its next trip or tour, respectively. In constraints (19), if $x_{\sigma^o_{k,r}, \sigma^d_{k,r}} = 1$, the $r$-th trip of vehicle $k$ is not really performed and therefore the starting time of trip $(k, r+1)$ can be equal to the completion time of trip $(k,r)$; otherwise the starting time of trip $(k, r+1)$ cannot be earlier than the completion time of trip $(k,r)$ plus the rest break $B$. Similarly, in constraints (20), if $z_{h,r,k,p} = 0$ for all $k \in K$ and $p \in R$, assistant $h$ on his/her $r$-th tour is not really performed, and therefore, the starting time of tour $(k, r+1)$ can be equal to the completion time of tour $(k,r)$; otherwise the starting time of tour $(k, r+1)$ cannot be earlier than the completion time of tour $(k,r)$ plus the rest break $B$.

Synchronization of the vehicles and the assistants is imposed by constraints (21) and (22). If tour $(h,r)$ $(h \in H, r \in R)$ coincides with trip $(k,p)$ $(k \in K, p \in R)$, i.e., $z_{h,r,k,p} = 1$, constraints (21) ensure that the starting time of trip $(k,p)$ cannot be earlier than the starting time of tour $(h,r)$, while constraints (22) ensure that the completion time of tour $(h,r)$ cannot be earlier than the completion time of trip $(k,p)$.

Finally, constraints (23) guarantee that the working time of staff member $k$ $(k \in K \cup H)$ is either in $[e^s_k, e^l_k]$ or in $[l^s_k, l^l_k]$. The maximum duration constraints are enforced by constraints (24). Constraints (25) ensure that each tour of an assistant can coincide with at most one trip of the vehicles.

Constraints (3) could be lifted by introducing the reverse arc $(j,i)$. The tightened constraints are as follows:

$$w_{j,k,r} \geqslant w_{i,k,r} + x_{i,j} + x_{j,i} - 1, \forall i,j \in N_2, k \in K, r \in R \tag{34}$$

To strengthen the model, we add a set of simple but useful redundant constraints:

$$\sum_{r\in R} z_{h,r,k,p} \leqslant 1, \forall h \in H, k \in K, p \in R \tag{35}$$

The basic idea of inequalities (35) is to forbid more than one tour of an assistant to coincide with a trip of a vehicle. These inequalities are redundant for the model, because the time consistency of the tours of an assistant have prevented situation violating inequalities (35) from happening. But an LP solution may violate these constraints.

Note that constraints (8)–(10), (13)–(16), (21), and (22) are non-linear constraints. However, these constraints can be linearized using the big-M method. Let $Q_{max} = \max_{k\in K} Q_k + 3Q'_k$ and $Q'_{max} = \max_{k\in K} Q'_k$. Let $o_{min} = \min_{i\in P} o_i$ and $o_{max} = \max_{i\in P} o_i$. The linearized constraints are listed as follows:

- (8) is replaced by $a_j \geqslant a_i + s_i + t_{i,j} + M^1_{i,j}(x_{i,j} - 1)$, where $M^1_{i,j} = l_i + s_i + t_{i,j} - e_j$,
- (9) is replaced by $a_i \geqslant u_{k,r} + t_{\sigma^o_{k,r},i} + M^2_{i,k,r}(x_{\sigma^o_{k,r},i} - 1)$, where $M^2_{i,k,r} = l^s_k + t_{\sigma^o_{k,r},i} - e_i$,
- (10) is replaced by $v_{k,r} \geqslant a_i + s_i + t_{i,\sigma^d_{k,r}} + M^3_{i,k,r}(x_{i,\sigma^d_{k,r}} - 1)$, where $M^3_{i,k,r} = l_i + t_{i,\sigma^d_{k,r}} + s_i - e^s_k$,
- (13) is replaced by $d^m_j \geqslant d^m_i + q^m_j + M^4_{i,j,m}(x_{i,j} - 1)$ $(m = 1, 2, 3)$, where $M^4_{i,j,1} = Q_{max} + q^1_j, M^4_{i,j,2} = 2Q'_{max} + q^2_j$ and $M^4_{i,j,3} = Q'_{max} + q^3_j$,
- (14) is replaced by $\sum_{h\in H}\sum_{r\in R} z_{h,r,k,p} + d_i + M^5_{i,k}(w_{i,k,p} - 1) \leqslant Q_k + 3b^1_{k,p}$, where $M^5_{i,k} = Q_{max} - o_i + o_{max} - Q_k$,
- (15) is replaced by $d^2_i + 2Q'_{max}(w_{i,k,p} - 1) \leqslant 2b^2_{k,p}$,
- (16) is replaced by $d^3_i + Q'_{max}(w_{i,k,p} - 1) \leqslant b^3_{k,p}$,
- (21) is replaced by $u_{k,p} \geqslant u_{h,r} + M^6_{h,k}(z_{h,r,k,p} - 1)$, where $M^6_{h,k} = l^s_h - e^s_k$,
- (22) is replaced by $v_{h,r} \geqslant v_{k,p} + M^7_{h,k}(z_{h,r,k,p} - 1)$, where $M^7_{h,k} = l^s_k - e^s_h$.

### 3.3. Model II

In many real-world applications of the DARP (Qu and Bard, 2014; Lim et al., in press), many requests have the same pickup or drop-off locations. In the data of the NEATS, according to our observation, a significant proportion of requests have the depot, which is a large local hospital, as their pickup or delivery points. Treating these pickup or delivery points separately requires creating an artificial node for each of them, and hence complicates the model, as the number of decision variables and constraints in a model is usually proportional or even exponential to the number of nodes in the model. In the branch-and-price algorithm proposed for the PDP with configurable vehicle capacities, Qu and Bard (2014) impose the following two additional requirements on the operations on the requests with the same pickup or drop-off locations to speed up the algorithm: (i) clients who live together, who are being dropped off at the same location, and who have the same pickup/drop-off time windows should be routed together and (ii) clients in a vehicle who have the same destination should be dropped together. In our case, we adopt a similar way to simplify the operations on the requests whose pickup or delivery points correspond to the depot. That is, we allow the pickup or drop-off operations on these requests with large time windows to be grouped together. To be more specific, for a set of clients who are picked up from or dropped off at the depot and, at the same time, have large service time windows, if the service starting time of a vehicle is in the time windows of all of the clients, these clients can be grouped and served together, and the service time of these clients is equal to the summation of these clients' service time.

According to the new requirement, we propose another MIP model for the problem. Let $P_1 \subset P$ and $D_1 \subset D$ denote the set of pickup nodes and the set of delivery nodes whose pickup or drop-off operations can be grouped together, respectively. Let $P_2 = P \setminus P_1, D_2 = D \setminus D_1, N_1 = P_1 \cup D_1$ and $N_2 = P_2 \cup D_2$. We first create a new graph $G'' = (V'', A'')$, where $V'' = V' \setminus N_1$ is the node set and $A'' = A' \setminus \{(i,j)|(i,j) \in A', i \text{ or } j \in N_1\}$ is the arc set. Obviously, $G''$ is a sub-graph of $G'$. Let $P_{k,r} \subset P_1$ and $D_{k,r} \subset D_1$ $(k \in K, r \in R)$ denote the set of clients that are picked up or dropped off at the depot in trip $(k,r)$, respectively. The new time window constraints in $P_1$ and $D_1$ require the ready time of trip $(k,r)$ to be within $[e_i, l_i]$ for all $i \in P_{k,r}$ and the time for vehicle $k$ back to the depot in trip $(k,r)$ to be within time windows $[e_i, l_i]$ for all $i \in D_{k,r}$, respectively. For a trip $(k,r)$ $(k \in K, r \in R)$, let.

- $s^u_{k,r}$ denote the total service time of clients in $P_{k,r}$, i.e. $\sum_{i\in P_{k,r}} s_i$;
- $s^v_{k,r}$ denote the total service time of clients in $D_{k,r}$, i.e. $\sum_{i\in D_{k,r}} s_i$;
- $d^m_{k,r}$ denote the number of seats $(m = 1)$, wheelchairs $(m = 2)$ or stretchers $(m = 3)$ used when leaving the depot, i.e. $\sum_{i\in P_{k,r}} d^m_i$.

According to the definitions of $s^u_{k,r}$ and $s^v_{k,r}$, the time when the vehicle leaves the depot is $u_{k,r} + s^u_{k,r}$ and the time when the vehicle returns back to the depot is $v_{k,r} - s^v_{k,r}$. Therefore constraints (9) and (10) should be modified to take $s^u_{k,r}$ and $s^v_{k,r}$ into consideration. Also, several new constraints are added to the model to describe the new features and to enforce the new time constraints in $P_1$ and $D_1$. The new model defined on $G''$, referred to as Model II, is formulated as follows:

$$\textbf{Model II}: \min \sum_{(i,j)\in A''} t_{i,j}x_{i,j} \tag{36}$$

$$\text{s.t. } s^u_{k,r} = \sum_{i\in P_1} w_{i+n,k,r}s_i, \quad \forall k\in K, r\in R \tag{37}$$

$$s^v_{k,r} = \sum_{i\in D_1} w_{i-n,k,r}s_i, \quad \forall k\in K, r\in R \tag{38}$$

$$a_i \geqslant (u_{k,r} + s^u_{k,r} + t_{\sigma^o_{k,r},i})x_{\sigma^o_{k,r},i}, \quad \forall i\in P_2\cup\{i|i\in D_2, i-n\in P_1\}, k\in K, r\in R \tag{39}$$

$$v_{k,r} - s^v_{k,r} \geqslant (a_i + s_i + t_{i,\sigma^d_{k,r}})x_{i,\sigma^d_{k,r}}, \quad \forall i\in D_2\cup\{i|i\in P_2, i+n\in D_1\}, k\in K, r\in R \tag{40}$$

$$e_iw_{i+n,k,r} \leqslant u_{k,r} \leqslant l_i + (1-w_{i+n,k,r})l^s_k, \quad \forall i\in P_1, k\in K, r\in R \tag{41}$$

$$e_iw_{i-n,k,r} \leqslant v_{k,r} - s^v_{k,r} \leqslant l_i + (1-w_{i-n,k,r})l^s_k, \quad \forall i\in D_1, k\in K, r\in R \tag{42}$$

$$(a_{i+n} - u_{k,r} - s^u_{k,r})w_{i+n,k,r} \leqslant f_i, \quad \forall i\in P_1, k\in K, r\in R \tag{43}$$

$$(v_{k,r} - s^v_{k,r} - a_{i-n} - s_{i-n})w_{i-n,k,r} \leqslant f_{i-n}, \quad \forall i\in D_1, k\in K, r\in R \tag{44}$$

$$d^m_{k,r} = \sum_{i\in P_1} q^m_i w_{i,k,r}, \quad \forall k\in K, r\in R, m=1,2,3 \tag{45}$$

$$d^m_i \geqslant q^m_i + d^m_{k,r}x_{\sigma^d_{k,r},i}, \quad \forall i\in P_2\cup\{i|i\in D_2, i-n\in P_1\}, k\in K, r\in R, m=1,2,3 \tag{46}$$

$$\sum_{h\in H}\sum_{r\in R} z_{h,r,k,p} + d^1_{k,p} \leqslant Q_k + 3b^1_{k,p}, \quad \forall k\in K, p\in R \tag{47}$$

$$d^2_{k,p} \leqslant 2b^2_{k,p}, \quad \forall k\in K, p\in R \tag{48}$$

$$d^3_{k,p} \leqslant b^3_{k,p}, \quad \forall k\in K, p\in R \tag{49}$$

$$s^u_{k,r}, s^v_{k,r} \geqslant 0, \quad \forall k\in K, r\in R \tag{50}$$

$$d^m_{k,r} \geqslant 0, \forall k\in K, r\in R, m=1,2,3 \tag{51}$$

$$(2)-(8), (11)-(33)$$

where variables $w_{i,k,r}, a_i$, and $d_i$ only consider nodes $i\in N_2; N, V'$, and $A'$ in constraints (2)–(8), (11)–(33) are replaced by $N_2, V''$, and $A''$, respectively; $P$ in constraints (6) is replaced by nodes set $N_2\setminus\{i|i-n\in P_2, i\in D_2\}; P$ in constraints (7) and (12) is replaced by nodes $\{i|i\in P_2, i+n\in D_2\}; P$ in constraints (14) is replaced by $P_2$; and $w_{i,k,r}$ in constraints (18) when $i\in P_1$ is replaced by $w_{i+n,k,r}$.

The objective function (36) minimizes the total travel time of the vehicles. Constraints (37) compute the time to pick up clients before a vehicle leaves the depot and constraints (38) compute the time to drop off clients after a vehicle returns back to the depot. Constraints (39) and (40) are used to replace constraints (9) and (10) in Model I. Constraints (41) and (42) are the new time window constraints for the nodes in $P_1$ and $D_1$. And Constraints (43) and (44) are the maximum riding time constraints of clients in $P_1$ and $D_1$. The number of seats, wheelchairs and stretchers used in a vehicle when the vehicle leaves the depot is computed by constraints (45). Constraints (46) ensure the consistency of a vehicle's load between the depot and the pickup nodes directly linked to the depot. Constraints (47)–(49) ensure that the capacity constraints of different types of facilities are satisfied when the vehicle sets off from the depot.

Constraints (39), (40), (43), (44), and (46) are not linear constraints, but they can be linearized as follows:

- (39) is replaced by $a_i \geqslant u_{k,r} + s^u_{k,r} + t_{\sigma^o_{k,r},i} + M^8_{i,k,r}\left(x_{\sigma^o_{k,r},i} - 1\right)$ where $M^8_{i,k,r} = l^s_k + t_{\sigma^o_{k,r},i} + \min\left\{\sum_{i\in P_1} s_i, L\right\} - e_i$,
- (40) is replaced by $v_{k,r} - s^v_{k,r} \geqslant a_i + s_i + t_{i,\sigma^d_{k,r}} + M^9_{i,k,r}\left(x_{i,\sigma^d_{k,r}} - 1\right)$ where $M^9_{i,k,r} = l_i + t_{i,\sigma^d_{k,r}} + s_i + \min\left\{\sum_{i\in D_1} s_i, L\right\} - e^s_k$,
- (43) is replaced by $a_i - u_{k,r} - s^u_{k,r} + M^{10}_{i,k}(w_{i,k,r} - 1) \leqslant f_i$ where $M^{10}_{i,k} = l_i - e^s_k - f_i$,
- (44) is replaced by $v_{k,r} - s^v_{k,r} - a_{i-n} - s_{i-n} + M^{11}_{i,k}(w_{i-n,k,r} - 1) \leqslant f_{i-n}$ where $M^{11}_{i,k} = l^s_k - f_{i-n} - e_{i-n} - s_{i-n}$,
- (46) is replaced by $d^m_i \geqslant q^m_i + d^m_{k,r} + M^{12}_{i,k,j}\left(x_{\sigma^o_{k,r},i} - 1\right)$ $(m=1,2,3)$ where $M^{12}_{i,k,1} = \min\left\{\sum_{i\in P_1} q^1_i, Q_{max}\right\}$, $M^{12}_{i,k,2} = \min\left\{\sum_{i\in P_1} q^2_i, 2Q'_{max}\right\}$, and $M^{12}_{i,k,3} = \min\left\{\sum_{i\in P_1} q^3_i, Q'_{max}\right\}$.

Constraints (39) and (40) could be lifted by replacing variable $x_{\sigma^o_{k,r},i}$ and $x_{i,\sigma^d_{k,r}}$ with variable $w_{i,k,r}$. As constraints (4) and (5) have forced the value of variable $w_{i,k,r}$ to be larger or equal to the value of corresponding variable $x$, such lifting is obviously valid.

### 3.4. Relationship between Model I and Model II

Model II simplifies the pickup and drop-off operations of the requests associated with depot by ignoring the pickup or drop-off sequence of these requests, and therefore, has fewer decision variables and constraints than Model I on the same instance. Provided that the involving requests have large time windows, this simplification will not make a big difference between Model I and Model II on the same instance in practice. But in theory, a feasible trip of Model I may be infeasible for Model II and vice versa. On the one hand, a feasible trip of Model I may be infeasible for Model II because the nodes in $N_1$ in Model II must be visited at the beginning or the end of a trip, while these nodes in Model I can be visited at any part of a trip. For instance, let $(i_1, \ldots, i_{k-1}, i_k, i_{k+1}, \ldots, i_n)$ be a feasible trip of Model I with $i_{k-1}, i_{k+1} \in N_2$ and $i_k \in N_1$. Then this trip is infeasible for Model II, because the position of node $i_k$ in this trip is infeasible according to the definition of a trip in Model II. On the other hand, a feasible trip of Model II may be infeasible for Model I, because in Model I, the time windows constraints and the maximum riding time constraints related to nodes in $N_1$ may be violated by a feasible trip of Model II. As the pickup or drop-off sequence of the requests related to the nodes in $N_1$ are ignored in Model I, the time windows constraints and the maximum riding time constraints related to these nodes in Model II are less restrictive than in Model I.

## 4. Valid inequalities

In this section, we introduce eight families of valid inequalities to strengthen Model I as well as Model II. For convenience, we use notations $P, D, N, A'$ and $V'$ with respect to Model I to describe the inequalities. When the inequalities are applied for Model II, notations $P, D, N, A'$ and $V'$ are just replaced by notations $P_2, D_2, N_2, A''$ and $V''$, respectively. Before we describe these inequalities in detail, we first introduce some notations used throughout this section. For a subset $S \subset V'$,

- $\bar{S} = \{i | i \in V', i \notin S\}$,
- $\delta^+(S) = \{(i,j) | (i,j) \in A', i \in \bar{S}, j \in S\}$,
- $\delta^-(S) = \{(i,j) | (i,j) \in A', i \in S, j \in \bar{S}\}$,
- $\delta(S) = \delta^+(S) \cup \delta^-(S)$,
- $x(S) = \sum_{i,j \in S} x_{i,j}$,
- $\pi(S) = \{i \in P | n + i \in S\}$,
- $\sigma(S) = \{n + i \in D | i \in S\}$.

For two disjoint subsets $S_1$ and $S_2$ $(S_1, S_2 \subset V', S_1 \cap S_2 = \emptyset)$, let $x(S_1 : S_2) = \sum_{i \in S_1} \sum_{j \in S_2} x_{i,j}$.

### 4.1. Strengthened subtour elimination inequalities

Subtour elimination inequalities (SEI) are a family of inequalities derived from the TSP and widely used in various VRPs. Balas et al. (1995) first proposed two strong versions of SEI for the precedence-constrained asymmetric TSP by considering the precedence constraints. Later, SEI were successfully applied to solve the DARP in the branch-and-cut algorithm proposed by Cordeau (2006). For a subset $S \subset N$, these two strong versions of SEI are as follows:

$$x(S) + \sum_{i \in \bar{S} \cap \sigma(S)} \sum_{j \in S} x_{i,j} + \sum_{i \in \bar{S} \setminus \sigma(S)} \sum_{j \in S \cup \sigma(S)} x_{i,j} \leqslant |S| - 1 \quad (52)$$

$$x(S) + \sum_{i \in S} \sum_{j \in \bar{S} \cap \pi(S)} x_{i,j} + \sum_{i \in S \cap \pi(S)} \sum_{j \in \bar{S} \setminus \pi(S)} x_{i,j} \leqslant |S| - 1 \quad (53)$$

In the R-DARP, because of various restrictions on a trip, such as the maximum duration constraints, maximum riding time constraints, the capacity constraints, and the time window constrains, trips' length is usually quite short compared with that in the TSP. Hence, given a subset of nodes in $N$, it is very likely that more than one trip is required to cover the subset. Therefore, the effect of SEI (52) and (53) on the R-DARP is quite weak compared to their effect on the TSP. Motivated by the $k$-path inequalities proposed by Kohl et al. (1999) for the VRP with time windows, we tighten SEI (52) and (53) as follows:

**Proposition 1.** *For a subset $S \subset N$, let $r(S)$ denote the minimum number of trips needed to serve all of the clients included in S. By "included in S", it means that either the client's pickup node or delivery node or both belong to S. Then, the following inequalities are valid for the R-DARP:*

$$x(S) + \sum_{i \in S} \sum_{j \in \bar{S} \cap \pi(S)} x_{i,j} + \sum_{i \in S \cap \pi(S)} \sum_{j \in \bar{S} \setminus \pi(S)} x_{i,j} \leqslant |S| - r(S) \quad (54)$$

$$x(S) + \sum_{i \in \bar{S} \cap \sigma(S)} \sum_{j \in S} x_{i,j} + \sum_{i \in \bar{S} \setminus \sigma(S)} \sum_{j \in S \cup \sigma(S)} x_{i,j} \leqslant |S| - r(S) \quad (55)$$

**Proof.** Here we only prove (54), because (55) can be proved in a similar way.

By substituting $2x(S) + x(\delta(S)) = 2|S|$ into (54), we have

$$\frac{1}{2}x(\delta(S)) - \sum_{i \in S} \sum_{j \in \bar{S} \cap \pi(S)} x_{i,j} - \sum_{i \in S \cap \pi(S)} \sum_{j \in \bar{S} \setminus \pi(S)} x_{i,j} \geqslant r(S) \tag{56}$$

which can be re-written as

$$\frac{1}{2}x(\delta(S)) - \sum_{i \in S} \sum_{j \in \bar{S} \cap \pi(S)} x_{i,j} - \sum_{i \in S} \sum_{j \in \bar{S} \setminus \pi(S)} x_{i,j} + \sum_{i \in S \setminus \pi(S)} \sum_{j \in \bar{S} \setminus \pi(S)} x_{i,j} \geqslant r(S) \tag{57}$$

Because

$$x(\delta(S)) = \sum_{i \in S} \sum_{j \in \bar{S}} x_{i,j} + \sum_{i \in \bar{S}} \sum_{j \in S} x_{i,j} \tag{58}$$

and

$$\sum_{i \in S} \sum_{j \in \bar{S}} x_{i,j} = \sum_{i \in \bar{S}} \sum_{j \in S} x_{i,j} \tag{59}$$

we have

$$\sum_{i \in S \setminus \pi(S)} \sum_{j \in \bar{S} \setminus \pi(S)} x_{i,j} \geqslant r(S) \tag{60}$$

Therefore, to prove (54), we only need to prove (60).

For any feasible scheduling plan that has successfully served all of the clients included in $S$, there must exist at least $r(S)$ different and disjoint trips that enter $S$. Let $\phi$ denote one of those trips and $S_\phi$ denote the nodes that are in $S$, and at the same time, are visited by $\phi$. Let node $n_\phi$ be the last node in $S$ visited by $\phi$, and $\bar{n}_\phi$ be the successor of $n_\phi$ in $\phi$. Note that $\bar{n}_\phi \in \bar{S}$. According to the precedence constraints, $n_\phi \notin S \cap \pi(S)$ and $\bar{n}_\phi \notin \pi(S)$; otherwise it could not be the last node in $S$ visited by $\phi$. Therefore, we have $n_\phi \in S \setminus \pi(S)$ and $\bar{n}_\phi \in \bar{S} \setminus \pi(S)$, which means trip $\phi_i$ must travel from $S \setminus \pi(S)$ to $\bar{S} \setminus \pi(S)$. As there are at least $r(S)$ trips, the flow passing from $S \setminus \pi(S)$ to $\bar{S} \setminus \pi(S)$ must be larger than or equal to $r(S)$, which is exactly the meaning of (60). □

### 4.2. Strengthened precedence inequalities

Precedence inequalities were first introduced by Ruland and Rodin (1997) for the PDP and were then applied in the branch-and-cut algorithm for the DARP by Cordeau (2006). The precedence inequalities used by Cordeau (2006) are described as follows. For a subset $S$, there exists $i \in P$, such that $i \notin S, i + n \in S$, start depot $\in S$, and end depot $\notin S$ then

$$x(\delta(S)) \geqslant 3 \tag{61}$$

For a proof of validity of inequalities (61), see Cordeau (2006).

According to our preliminary computational experiments, the effect of precedence inequalities (61) on the R-DARP is weak. Therefore, we propose another version of precedence inequalities. Let $\rho(S) = \{i | i \in S \cap P, i + n \in N \setminus S\}$. For a subset $S \subset N$, where $\rho(S) \neq \emptyset$, the precedence inequalities are stated as follows:

$$x(S : N_2 \setminus S) \geqslant 1 \tag{62}$$

The intuition behind inequalities (62) is that given a subset of nodes that covers the pickup node of a request but not its delivery node, there must be at least one trip out of this subset.

Similar to SEI (52) and (53), precedence inequalities (62) can be strengthened by considering the context where multiple trips are required to serve all of the clients in $\rho(S)$. For a subset $S \subset N$, let $u(S)$ denote the pickup and delivery nodes belonging to the clients that cannot be served in the same trip with any client in $S$. If $\rho(S) \neq \emptyset$, the strengthened precedence inequalities are as follows:

$$x(S \setminus u(\rho(S)) : N \setminus (S \cup u(\rho(S)))) \geqslant r(\rho(S)) \tag{63}$$

The validity proof of inequalities (63) is similar to the proof of Proposition 1. The intuition behind it is that for any feasible routing plan, at least $r(\rho(S))$ trips are needed to serve all of the nodes in $\rho(S)$ and those trips must leave $S$ to serve the corresponding delivery nodes of $\rho(S)$. Trips travelling through $u(\rho(S))$ cannot serve $\rho(S)$, therefore they are excluded in (63).

### 4.3. Capacity inequalities

We propose a set of capacity inequalities considering the configurable capacity. For a subset $S \subset N$, let $q^m(S) = \sum_{i \in S} q_i^m$ $(m = 1, 2, 3)$. Then $q^m = \max\{q^m(\pi(S) \setminus S), -q^m(\sigma(S) \setminus S)\}$ $(m = 1, 2, 3)$ are lower bounds of the numbers of seats $(m = 1)$, wheelchairs $(m = 2)$ and stretchers $(m = 3)$ required to serve all of the clients in $S$. Based on $q^m$ $(m = 1, 2, 3)$, we can determine a lower bound for the number of vehicles required to serve all of the clients in $S$, by solving an optimization problem. Let $\theta_k$ be a binary decision variable which is equal to one if vehicle $k \in K$ serves the clients in $S$, and zero otherwise, and $b_k^m$ be an integral decision variable indicating the number of divisions which are configured into seats $(m = 1)$, wheelchairs $(m = 2)$ and stretchers $(m = 3)$, respectively. Let $o'_{min} = \min_{i \in \pi(S)} o_i$. The optimization problem is formulated as follows:

$$\textbf{CIP} : \min \sum_{k \in K} \theta_k \tag{64}$$

$$\text{s.t.} \sum_{k \in K} \theta_k (Q_k - o'_{min}) + \sum_{k \in K} 3 b_k^1 \geqslant q^1 \tag{65}$$

$$\sum_{k \in K} 2 b_k^2 \geqslant q^2 \tag{66}$$

$$\sum_{k \in K} b_k^3 \geqslant q^3 \tag{67}$$

$$b_k^1 + b_k^2 + b_k^3 \leqslant \theta_k Q'_k, \quad \forall k \in K \tag{68}$$

$$b_k^m \geqslant 0 \text{ and integral}, \quad \forall m = 1, 2, 3, k \in K \tag{69}$$

$$\theta_k \in \{0, 1\}, \forall k \in K \tag{70}$$

Problem CIP can be solved by the following dynamic programming. Let $U(d_1, d_2, d_3)$ denote the minimum number of vehicles to serve clients who in total require $d_1$ seats, $d_2$ wheelchairs and $d_3$ stretchers. Then $U(d_1, d_2, d_3)$ can be computed with the following dynamic recursion:

$$U(d_1, d_2, d_3) = \min_{3b^1 + Q_k \geqslant o'_{min}, b^1 + b^2 + b^3 \leqslant Q'_k, k \in K} 1 + U(\max\{0, d_1 - (Q_k + 3b^1 - o'_{min})\}, \max\{0, d_2 - 2b^2\}, \max\{0, d_3 - b^3\}) \tag{71}$$

with boundary condition:

$$U(0, 0, 0) = 0 \tag{72}$$

The optimal solution of problem CIP is equal to $U(q^1, q^2, q^3)$, denoted by $\kappa(S)$. Then the capacity inequality with respect to $S$ is as follows:

$$x(\delta^+(S)) \geqslant \kappa(S) \tag{73}$$

### 4.4. Infeasible path elimination inequalities

We consider three classes of infeasible path elimination inequalities (IPEI) in our branch-and-cut algorithm. The first class of IPEI is defined on a single infeasible path within a trip. For any infeasible directed path $\phi = \{i_1, i_2, \ldots, i_p\}$ within a trip, the IPEI with respect to $\phi$ is defined as follows:

$$\sum_{j=1}^{p-1} x_{i_j, i_{j+1}} \leqslant p - 2 \tag{74}$$

The second class of IPEI is based on the break between two consecutive trips performed by the same vehicle. For two consecutive trips $(k, r)$ and $(k, r+1)$ of vehicle $k$ $(k \in K, r = 1, \ldots, |R| - 1)$, let $\phi_1 = \{i_1, i_2, \ldots, i_g, \sigma_{k,r}^d\}$ and $\phi_2 = \{\sigma_{k,r+1}^o, j_1, \ldots j_p\}$ denote two sub-paths of trips $(k, r)$ and $(k, r+1)$, respectively, and $e_{\phi_1}$ denote the earliest time for a vehicle to arrive at the depot along $\phi_1$, and $l_{\phi_2}$ denote the latest time for a vehicle to start from the depot along $\phi_2$. If $e_{\phi_1} + B > l_{\phi_2}$, sub-paths $\phi_1$ and $\phi_2$ cannot be taken by vehicle $k$ at the same time and the IPEI with respect to $\phi_1$ and $\phi_2$ is as follows:

$$\sum_{m=1}^{g-1} x_{i_m, i_{m+1}} + x_{i_g, \sigma_{k,r}^d} + x_{\sigma_{k,r+1}^o, j_1} + \sum_{m=1}^{p-1} x_{j_m, j_{m+1}} \leqslant p + g - 1 \tag{75}$$

The third category of IPEI is similar to the second one, but based on the break between two consecutive trips performed by the same assistant. For an assistant $h \in H$ and his/her trip index $u \in R$, two vehicles $k_1, k_2$ $(k_1, k_2 \in K)$ and their trip index $r_1, r_2$ $(r_1, r_2 \in R)$, and two sub-paths $\phi_1 = \left\{i_1, i_2, \ldots, i_g, \sigma_{k_1, r_1}^d\right\}$ and $\phi_2 = \left\{\sigma_{k_2, r_2}^o, j_1, \ldots j_p\right\}$, if $e_{\phi_1} + B > l_{\phi_2}$, the IPEI is described as follows:

$$\sum_{m=1}^{g-1} x_{i_m,i_{m+1}} + x_{i_g,\sigma^d_{k_1,r_1}} + z_{h,u,k_1,r_1} + z_{h,u+1,k_2,r_2} + x_{\sigma^o_{k_2,r_2},j_1} + \sum_{m=1}^{p-1} x_{j_m,j_{m+1}} \leqslant p+g+1 \tag{76}$$

### 4.5. Fork inequalities

The fork inequalities (FI) were proposed by Ropke et al. (2007), and can be viewed as a strengthened version of the infeasible path elimination inequalities. For a feasible path $R = \{p_1, \ldots, p_j\}$, let $S, T_1, \ldots, T_j \subset N$ be subsets such that $p_h \notin T_{h-1}$ for $h = 2, \ldots, j$. If for any integer $i \leqslant j$ and any node pair $p_0 \in S, r \in T_j$, the path $\{p_0, p_1, \ldots, p_i, r\}$ is infeasible, then the *infork inequality* is defined as follows:

$$\sum_{p_0 \in S} x_{p_0,p_1} + \sum_{h=1}^{j-1} x_{p_h,p_{h+1}} + \sum_{h=1}^{j} \sum_{r \in T_h} x_{p_h,r} \leqslant j \tag{77}$$

For a feasible path $R = \{p_1, \ldots, p_j\}$, let $S_1, \ldots, S_j, T \subset N$ be subsets such that $p_h \notin T_{h+1}$ for $h = 1, \ldots, j-1$. If for any integer $i \leqslant j$ and any node pair $r \in S_i, p_j \in T$, the path $\{r, p_1, \ldots, p_i, p_j\}$ is infeasible, then the *outfork inequality* is defined as follows:

$$\sum_{h=1}^{j} \sum_{r \in S_h} x_{r,p_h} + \sum_{h=1}^{j-1} x_{p_h,p_{h+1}} + \sum_{p_j \in T} x_{p_i,p_j} \leqslant j \tag{78}$$

### 4.6. Reachability inequalities

The reachability inequalities (RI) were firstly proposed by Lysgaard (2006) for the vehicle routing problem with time windows (VRPTW), and then were used by Ropke et al. (2007) and Ropke and Cordeau (2009) to solve the DARP and the PDPTW, respectively. For any node $i \in N$, let $A_i^-$ and $A_i^+$ be the reaching arc set and reachable arc set of node $i$, respectively. The reachable set and the reaching set of node $i$ contain all the arcs that are used by any feasible path from the depot to node $i$ and from node $i$ back to the depot, respectively. A node set $T$ is said to be conflicting if any two nodes in $T$ cannot be served by one trip. Let $A_T^- := \cup_{i \in T} A_i^-$ and $A_T^+ := \cup_{i \in T} A_i^+$ be the reaching arc set and reachable arc set for a conflicting set $T$. The two set of valid inequalities introduced by Lysgaard (2006) are as follows:

$$x(\delta^+(S) \cap A_T^+) \geqslant |T| \tag{79}$$

$$x(\delta^-(S) \cap A_T^-) \geqslant |T| \tag{80}$$

### 4.7. Trip number inequality

The trip number inequality (TNI) is based on a lower bound of the number of trips required to satisfy all of the transportation requests. Let $\eta(i) = t_{0,i} + t_{i,i+n} + t_{i+n,0}$ $(i \in P)$ and $\eta_{max} = \max_{i \in P} \eta_i$. The TSI is described as follows:

$$\sum_{i \in V_d} \sum_{j \in N} x_{i,j} \geqslant \left\lceil \left( \left( \eta_{max} + \sum_{i \in P} (s_i + s_{i+n}) \right) \Big/ L \right) \right\rceil \tag{81}$$

### 4.8. Lunch break inequalities

Because of the time windows, some clients can only be served before or after some drivers' lunch. The existence of these customers yields to a family of trivial but useful inequalities. Define a client $i \in P$ to be a morning client with respect to driver $k \in K$ if $l_i < l_k^l + t_{k \cdot r,i}$ $(r \in R)$ or an afternoon client, if $e_{i+n} + s_{i+n} + t_{i+n,k \times r} > e_k^l$ $(r \in R)$. For vehicle $k \in K$, a trip index $r \in R$ and a morning client $i$ with respect to $k$, the lunch break inequalities are defined as follows:

$$y_{k,r} \geqslant w_{i,k,r} \tag{82}$$

Similarly, for vehicle $k \in K$, a trip index $r \in R$, and an afternoon client $i$ with respect to $k$, the lunch break inequalities are defined as follows:

$$y_{k,r} \leqslant 1 - w_{i,k,r} \tag{83}$$

## 5. The branch and cut algorithm

The branch-and-cut algorithm is one of the most successful exact approaches to solve combinatorial optimization problems. It has been applied to solve the travelling salesman problem (Padberg and Rinaldi, 1991), the vehicle routing problems (Lysgaard et al., 2004; Bard et al., 2002), the location routing problem (Belenguer et al., 2011), etc. It is a tree search procedure where for each node, the LP relaxation of Model I (Model II) tightened by the valid inequalities in Section 4 is solved to optimality. If the optimal LP solution happens to be integral, it is also the optimal solution with respect to the current node. Then the solution can be used to update the upper bound if its cost is smaller than the current upper bound. If the optimal LP solution is not integral but its cost is smaller than the upper bound, the current node is branched into two child nodes; otherwise the current node is pruned. The global optimal solution is found after all of the nodes in the branch-and-bound tree are examined.

In this section, we first describe the preprocessing procedures to tighten the time windows of an instance and to eliminate unnecessary variables and constraints. Then we describe a preprocessing procedure to help identify the violated strengthened subtour inequalities (54) and (55) and the strengthened precedence inequalities (63). The details of the separation algorithms for the inequalities in Section 4 are presented at the end of this section. In this section, for convenience we use notations $P_1, P_2, D_1, D_2, N_2, A''$ and $V''$ with respect to Model II to describe the algorithm. To Model I, notations $P_2, D_2, N_2, A''$ and $V''$ are just replaced by $P, D, N, A'$ and $V'$, respectively, and $P_1 = D_1 = \emptyset$.

## 5.1. Preprocessing

### 5.1.1. Time window tightening

We tighten client time windows in two steps. First, let $E^s_{min} = \min_{i \in K} e^s_i$ and $L^s_{max} = \max_{i \in K} l^s_i$. Then each client's time window can be tightened by setting $e_i = \max\{e_i, E^s_{min} + t_{0,i}\}$ and $l_{i+n} = \min\{l_{i+n}, L^s_{max} - t_{i+n,0} - s_{i+n}\}$. Second, for a client $i \in P_2$, the time window on his or her pickup node and delivery node can be further tightened by setting $e_i = \max\{e_i, e_{n+i} - s_i - f_i\}, l_i = \min\{l_i, l_{i+n} - s_i - t_{i,i+n}\}, e_{i+n} = \max\{e_{i+n}, e_i + s_i + t_{i,i+n}\}$ and $l_{i+n} = \min\{l_{i+n}, l_i + s_i + f_i\}$. Similarly, for a client $i \in P_1$, the time window on his or her pickup node and delivery node can be further tightened by setting $e_i = \max\left\{e_i, e_{n+i} - \min\left\{L, \sum_{j \in P_1} s_i\right\} - f_i\right\}, l_i = \min\{l_i, l_{i+n} - s_i - t_{i,i+n}\}, e_{i+n} = \max\{e_{i+n}, e_i + s_i + t_{i,i+n}\}$ and $l_{i+n} = \min\{l_{i+n}, l_i + \min\left\{L, \sum_{j \in P_1} s_i\right\} + f_i\}$.

The work periods of staff members can be tightened as follows. Let $E_{min} = \min_{i \in P}\{e_i - t_{0,i}\}, L^1_{max} = \max_{i \in D_1}\{l_i + s_i + t_{i,0}\}$, and $L^2_{max} = \max_{i \in D_2}\left\{l_i + \min\left\{L, \sum_{j \in D_1} s_i\right\} + t_{i,0}\right\}$. Then the work period for staff $k \in K \cup H$ can be tightened by setting $e^s_k = \max\{E_{min}, e^s_k\}$ and $l^s_k = \min\{\max\{L^1_{max}, L^2_{max}\}, l^s_k\}$.

### 5.1.2. Variables and constraints reduction

If an arc $(i,j) \in A''$ is redundant, which means it cannot be used in any feasible solution, then variables and constraints related to this arc, e.g., $x_{i,j}$ and constraints (3), are also redundant and can be removed from Model II. Therefore, before solving Model II, we try to identify as many redundant arcs as possible, and remove their relative variables and constraints. Obviously, arcs $(n+i, i)$ $(i \in N_2, n+i \in N_2)$ are redundant. We apply approaches proposed by Dumas et al. (1991) and Cordeau (2006) to remove redundant variables and constraints with respect to nodes in $N_2$.

Clients in $P_1$ or $D_1$ cannot be served in the same trip if their time windows do not overlap. Therefore, for client $i, j \in P_1$, if $e_i > l_j$ or $e_j > l_i$, then arcs $(i+n, j+n)$ and $(j+n, i+n)$ are redundant. Similarly, for clients $i, j \in D_1$, if $e_i > l_j$ or $e_j > l_i$, arc $(i-n, j-n)$ and $(j-n, i-n)$ are redundant.

For client $i \in P$ and vehicle $k \in K$, if they satisfy at least one of the following conditions.

- $e_{i+n} + s_i + t_{i+n,0} > e^l_k$ and $l^l_k + t_{0,i} > l_i$,
- $l_i - t_{0,i} < e^s_k$,
- $e_{i+n} + t_{i+n,0} > l^s_k$.

Then $i$ cannot to be served by vehicle $k$. Therefore, relative arcs $(\sigma^o_{k,r}, i), (i+n, \sigma^d_{k,r})$ and variables $w_{i,k,r}$ $(r \in R)$ are redundant if they exist.

## 5.2. Computation of $r(S)$ and $u(S)$

For a client set, if all of the clients it contains can be served in a single trip in a certain order, we call it a feasible client set. Let $\mathcal{T}$ denote the pool of all of these feasible client sets. Then, given any feasible trip $r$ in the R-DARP, there exists a set in $\mathcal{T}$ that contains and only contains the clients served by $r$. And for any client set in $\mathcal{T}$, there must exists at least one feasible trip that can serve all of the clients in the set. Obviously, if a client set is not contained in $\mathcal{T}$, at least two trips are needed to serve all of the clients in the set.

If $\mathcal{T}$ is generated, then the computation of $r(S)$ in the strengthened subtour elimination inequalities (54) and (55), the strengthened precedence inequalities (63), and $u(S)$ in the strengthened precedence inequalities (63) can be generated as follows. For a subset of nodes $S \subset N_2$, if the set of all of the clients included in $S$ (the definition of "included in $S$" is the same as that in Proposition (1)) can be found in $\mathcal{T}$, then $r(S)$ is 1; otherwise, if the set of all of the clients included in $S$ can be divided into two sets in $\mathcal{T}$, then $r(S)$ is 2; otherwise $r(S)$ is set to 3. Desaulniers (2010) used a similar method to set $r(S)$

for the general capacity inequalities, which is a strengthened version of the subtour elimination inequalities. For a node $i \in N_2$, let node $j$ be a node that is unreachable from $i$ if there exists no set in $\mathcal{T}$ that contains both $i^u$ ($i^u = i$ if $i \in P; i^u = i - n$ if $i \in D$) and $j^u$. Let $U_i$ denote the set of unreachable nodes from $i$. Then $u(S)$ it equal to $\cap_{i \in S} U_i$.

To generate $\mathcal{T}$, we first generate all of the feasible client sets with no more than three clients, denoted by $\mathcal{T}_1$, by full enumeration, and then generate all the feasible client sets with more than three clients, denoted by $\mathcal{T}_2$, by a label-setting algorithm. Because the size of client set to generate by enumeration is restricted to 3, the enumeration process can be done fast. However, the speed of the label-setting algorithm depends on the size of an instance. When the size of an instance is large, the label-setting algorithm cannot terminate in reasonable time (e.g., 20 min in our experiments). If the label-setting algorithm is forced to terminate, $r(S)$ is forced to be 1 for $S$ in which more than 3 clients are "included". So by applying a two-stage procedure, we can at least obtain non-trivial $r(S)$ for $S$ in which at most 3 clients are "included". In addition, when computing $u(S)$, using $\mathcal{T}_1$ is enough, because for two nodes, if there exists no feasible client set in $\mathcal{T}_1$ that covers these two nodes, it is also impossible to find such a feasible client set in $\mathcal{T}_2$.

To speed up the enumeration process, we consider relaxed versions of the capacity constraints and the maximum riding time constraints. The original configurable capacity constraints are relaxed to the classic one-dimension capacity constraints where the demands of the clients and the vehicle capacity are counted in terms of seats. Let $q'_i = q^1_i + 1.5q^2_i + 3q^3_i$. Then $Q_{max}$ and $q'_i$ are viewed as the capacity of the vehicles and the demand of client $i$. In the relaxed maximum riding time constraints, we use an easy-to-compute lower bound to replace the true riding time of a client, and a client is considered to violate the maximum riding time constraints if the lower bound is larger than the client's riding time limit. For a path $\{p_0, p_1, \ldots, p_j\}, \bar{r} = t_{p_0,p_1} + \sum_{k=1}^{j-1} (t_{p_k,p_{k+1}} + s_{p_k} + max\{0, e_{p_{k+1}} - l_{p_k} - t_{p_k,p_{k+1}} - s_{p_k})\}$ is a lower bound of the riding time between node $p_0$ and $p_j$. Here, we choose this lower bound not only because it is easy to compute, but also because it is valid for all of clients in both Model I and Model II. Note that the riding time constraints for the clients in $P_1 \cup D_1$ in Model II are slightly different from others.

In the label-setting algorithm, we completely ignore the maximum riding time constraints and consider the relaxed version of the capacity constraints as the enumeration process. Here we drop the maximum riding time constraints and relax the capacity constraints because these constraints complicate the label-setting algorithm and probably slow down the label-setting algorithm dramatically. The design of the label-setting algorithm is based on Model I. As described in Section 3.4, trips in Model I and Model II are different. After the maximum riding time constraints is dropped, the differences between them are time window constraints related to nodes in $P_1 \cup D_1$, which causes a feasible trip of Model II to be infeasible for Model I. To use the label-setting algorithm in Model II, we relax the time windows constraints related to nodes in $P_1 \cup D_1$. For node $i \in P_1$, the latest service starting time of node $i$, i.e. $l_i$, is delayed to $l_i + \min\left\{L, \sum_{j \in P_1} s_j\right\} - s_i$. Similarly, for node $i \in D_1, l_i$ is delayed to $l_i + \min\left\{L, \sum_{j \in D_1} s_j\right\} - s_i$. In this way, any feasible trip of Model II under original time window constraints is also feasible for Model I under the relaxed time window constraints.

We adopt a label-setting algorithm similar to that proposed by Hernandez et al. (2013) for the multi-trips VRP with time windows and limited duration (MTVRPTW-LD). Let $L_r = (q_r, o^{min}_r, o^{max}_r, d_r, \mathcal{A}_r, h_r, \mathcal{B}_r, V^1_r, \cdots, V^{2n}_r)$ be a label representing a partial trip $r$ associated with node $i$ where.

- $q_r$ is the number of seats used in the vehicle at node $i$,
- $o^{min}_r$ is the minimum number of assistants required to serve requests in $r$,
- $o^{max}_r$ is the maximum number of assistants that can get on the vehicle in the depot,
- $h_r$ is the earliest arrival time at node $i$,
- $d_r$ is the minimum duration of $r$,
- $\mathcal{A}_r$ is the earliest departure time from the depot to achieve the minimum duration for $r$,
- $\mathcal{B}_r$ is the latest arrival time at node $i$ when the duration is equal to the minimum duration,
- $V^i_r$ is binary resource which is equal to one if node $i$ is visited in $r$, and zero otherwise.

Resources $h_r, d_r, \mathcal{A}_r$ and $\mathcal{B}_r$ are introduced for handling the time windows constraints and the maximum duration constraints. For more detailed information about these resources, please see Hernandez et al. (2013). Resources $q_r, o^{min}_r$ and $o^{max}_r$ are relative to the relaxed capacity constraints and the minimum assistant requirement constraints. When label $L_r$ is extended to node $i$ so that a new label $L_{r'}$ is created, these resources associated with $L_{r'}$ are set as follows:

$$q_{r'} = q_r + q'_i \tag{84}$$
$$o^{min}_{r'} = \max\{o^{min}_r, o_i\} \tag{85}$$
$$o^{max}_{r'} = \min\{o^{max}_r, Q_{max} - q_{r'}\} \tag{86}$$

We adopt a set of domination rules similar to those in Hernandez et al. (2013) to eliminate dominated labels. If trips $r$ and $r'$ lead to the same node, then $r$ dominates $r'$ if and only if $q_r \leqslant q_{r'}, o^{min}_r \leqslant o^{min}_{r'}, o^{max}_r \geqslant o^{max}_{r'},\ h_r \leqslant h_{r'}, d_r \leqslant d_{r'}, \mathcal{A}_r \leqslant \mathcal{A}_{r'}, \mathcal{B}_r \geqslant \mathcal{B}_{r'}, V^j_r = V^j_{r'}\ (\forall j = 1, \ldots, n)$ and $V^j_r \geqslant V^j_{r'}\ (\forall j = n+1, \ldots, 2n)$. Because constraints $V^j_r = V^j_{r'}\ (\forall j = 1, \ldots, n)$ and $V^j_r \geqslant V^j_{r'}$ $(\forall j = n+1, \ldots, 2n)$ guarantee constraints $q_r \leqslant q_{r'}$ and $o^{min}_r \leqslant o^{min}_{r'}$ to be satisfied, constraints $q_r \leqslant q_{r'}$ and $o^{min}_r \leqslant o^{min}_{r'}$ can be ignored from the domination rules. Here we do not require the cost of $r'$ to be less than or equal to that of $r$ because the cost dost not concerned with the feasibility of a trip.

After $\mathcal{T}_2$ is generated, it can be further refined by $\mathcal{T}_1$. For a feasible client set $S \in \mathcal{T}_2, S$ can be removed from $\mathcal{T}_2$ if there exists a $S' \subset S$ such that $S'$ consists of no more than 3 clients but $S' \notin \mathcal{T}_1$.

### 5.3. Separation heuristics

We implemented two heuristics to simultaneously identify the violated strengthened subtour elimination inequalities (54) and (55), the strengthened precedence inequalities (63), and the capacity inequalities (73). These two heuristics are the shrink heuristic and the connected component heuristic, both of which are classical separation heuristics for the general capacity inequalities (Augerat et al., 1998). Let $\bar{x}$ denote the values of variables $x$ in the current optimal LP solution. Before the heuristic is invoked, we first construct a supported graph $G^* = (N_2, A^*)$, where $N_2$ is the node set and $A^* = \{(i,j) | i,j \in N_2, i \neq j, \bar{x}_{i,j} > 0\}$ is the arc set. The shrink heuristic is an iterative heuristic. In each iteration, the heuristic selects arc $(i,j)$ in $A^*$ with the maximum arc value $\bar{x}_{i,j}$ and shrinks nodes $i,j$ into a super node $k$. By shrinking $i,j$ into $k$, it means $i,j$ are replaced by $k$, and all of the arcs pointing to $i$ and $j$ are set to pointing to $k$, and all of the arcs out from (into) $i$ and $j$ are set to pointing out from (into) $k$. Then the node set represented by $k$ is examined to check whether it violates inequalities (54), (55), (63), and (73). The violated inequalities are added to the current LP problem. The iteration process repeats until all of the arc values in $A^*$ are smaller than the threshold 0.1.

The connected component heuristic works as follows. It begins by computing the set of connected components in $G^*$. Here a connected component is a subgraph of $G^*$ where every two nodes are connected with each other through a path without considering the direction of an arc. For each connected component, the heuristic iteratively searches for the violated inequalities. In each iteration, the component as a whole is checked to determine whether it violates inequalities (54), (55), (63) and (73). If violated inequalities are found, they are saved and added to a pool of violated inequalities. Then the heuristic removes the node which, if removed, will lead to the least outflow from this component. The iteration process repeats until the outflow from the component is smaller than the threshold 0.1.

When separating the capacity inequality (73), the right-hand-side value, i.e. $\kappa(S)$, has to be computed by the dynamic programming approach discussed in Section 4.3. If $\kappa(S)$ needs to be computed for each subset $S$, the separation heuristics would be slowed down dramatically. Fortunately, all of the possible values of $\kappa(S)$ can be enumerated before the separation heuristics are invoked. The detail enumeration process is as follows: for each value $o'_{min}$ in $[o_{min}, o_{max}]$, we invoke the dynamic programming to compute $U(\sum_{i\in P} q_i^1, \sum_{i\in P} q_i^2, \sum_{i\in P} q_i^3)$. Note that during the dynamic programming, states $U(d_1, d_2, d_3)$ with $0 \leqslant d_1 < \sum_{i\in P} q_i^1, 0 \leqslant d_2 < \sum_{i\in P} q_i^2$, and $0 \leqslant d_3 < \sum_{i\in P} q_i^3$ are also generated.

To identify the violated infeasible path elimination inequalities (74), a path-construction heuristic (Cordeau, 2006) is applied to each node $i \in P_2$. The heuristic constructs a path from node $i$ according to the current solution, and either ends at $i+n$, depot, or when a circle is found. Then the constructed path is checked to find out whether it is feasible. Here the maximum riding time constraints are checked using the same method as the enumeration process in Section 5.2. To identify the violated infeasible path elimination inequalities (75) and (76), all of the node pairs that cannot be served by two consecutive trips of a staff member are generated before the branch-and-cut algorithm. To be more specific, at the initialization, we enumerate all of the node pairs $(i,j)$ that satisfy $e_i + s_i + t_{i,0} + B + t_{0,j} > l_j$. Then, for each of these node pairs, if they are served by two consecutive trips of either a vehicle or an assistant in the current optimal LP solution, the node pair and corresponding trips are added to the pool of violated inequalities.

We use the algorithms proposed by Ropke et al. (2007) to separate the fork inequalities (78) and (77). To save running times, only paths containing at most four nodes are considered. In addition, we use the method in the enumeration process in Section 5.2 to check the maximum riding time constraints.

We follow the procedures proposed by Ropke et al. (2007) to identify the reachability inequalities (79) and (80). The conflict client set can be identified with the help of $\mathcal{T}$ in Section 5.2. If two clients are not in any feasible client set in $\mathcal{T}$ simultaneously, they are considered to be conflicted. For each conflict client set, we only consider two types of conflict node sets. The first type of node set consists of the pickup nodes in $P_2$, and the delivery nodes whose pickup nodes are in $P_1$. The second type of node set consists of the delivery nodes whose pickup nodes are in $P_2$. According to our preliminary computational experiments, these two types of node sets yielded the best results.

For the trip number inequalities (81) and the lunch break inequalities (83) and (82), all of them are very simple so that they can be separated by full enumeration.

## 6. Computational experiments

In this section we present the computational results of the branch-and-cut algorithm on a set of instances generated from the real-world data of the NEATS. The algorithm was implemented in Java using callbacks of ILOG CPLEX 12.5.1. Note that CPLEX applies a branch-and-cut algorithm to solve MIP problems. The callbacks allow users to add user-defined inequalities to the model at each node of the branch-and-bound tree, while CPLEX takes over the other parts of the branch-and-cut algorithm, like branching decisions and node exploration strategies. All of the experiments were conducted on a Dell personal computer with an Intel i7-4790 3.60 GHz CPU, 16 GB RAM, and Windows 7 operating system.

### 6.1. Test instances

The NEATS dataset contains detailed information about client's requests, vehicles, and staffs in the working days of 2009. The information includes (1) the pickup and delivery locations of each client; (2) the travel time between any two locations; (3) the time window on each pickup and deliver node; (4) the number of seats, wheelchairs, stretchers and assistants required by each client; (5) the work period and the lunch period of each staff member; and (6) the capacity configuration of each vehicle, i.e. the number of fixed seats and configurable divisions. However, some information necessary for the R-DARP does not exist in the dataset, such as the maximum riding time of a client and the maximum duration of a trip. All of the clients to be served in a single working day and related information constitute one instance in NEATS. However, these instances are too large to be solved exactly by the branch-and-cut algorithm: most of the instances have more than 100 requests, which are required to be served by more than 10 vehicles and 10 staff members. Therefore, we generate a set of small-size instances for the R-DARP by randomly sampling requests and staffs from the NEATS dataset. The non-existent information, e.g., the maximum riding time of a client and the maximum duration of a trip, are generated artificially.

The maximum riding time of client $i$, namely $f_i$, is set to $\min\{60, 3\max\{10, t_{i,i+n}\}\}$. To ensure a good experience for each client, it is reasonable to require the time he/she spends on a vehicle to be less than 60 min. However, if the minimum travel time of a client (i.e., the time needed to travel from his/her origin to the destination directly) only requires, for example, 10 min, then the client may easily get annoyed if he/she spends 60 min on the vehicle. Therefore, another limitation on the maximum riding time of a client is added. That is, a client should not stay on the vehicle for more than three times his/her minimum travelling time. A lower bound of 10 min is needed in case of extreme requests with a very small $t_{i,i+n}$. The maximum duration of a trip, i.e., $L$, is set to 150 min. The rest time after each trip, i.e. $B$, is equal to 30 min.

In the NEATS dataset, there are two major types of vehicles: one with one fixed seat and three configurable divisions, and the other with only three configurable divisions. In the R-DARP instances, both of these two types of vehicles are included. The maximum number of trips each staff member can take during a single working day, i.e. $|R|$, is set to three. In the NEATS dataset, the work periods of many staff members last for about 9 h a day, including one hour's lunch time. When the staff members are allowed to perform three trips one day, with each trip's maximum duration being 2.5 h and the break time being 30 min, they can actually have a tight schedule.

The R-DARP dataset consists of two types of instances. The first type of instances, referred to as *normal instances*, have a similar proportion of requests, whose pickup node or delivery node is the depot, as the NEATS dataset. The second type of instances, referred to as *scattered instances*, have no requests whose pickup node or delivery node is the depot. The R-DARP dataset contains 24 normal instances, and 18 scattered instances in total. These instances are now available in www.computational-logistics.org/orlib/topic/R-DARP.

The characteristics of the normal instances and the scattered instances are detailed in Tables 1 and 2, respectively. We mainly discuss the characteristics of the normal instances, as the scattered instances are similar to the normal instances.

**Table 1**
Characteristics of the normal instances.

| Instance | $n$ | $\|K\|$ | $\|H\|$ | $p_{o>1}$ | $p_0$ | $p_{dp}$ | $i \in P_1 \cup D_1$ | | | $(l_i - e_i), i \in P \cup D$ | | | Avg. $f_i$ | $t_{i,i+n}, i \in P$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | Avg. $s_i$ | Min. $(l_i - e_i)$ | Avg. $(l_i - e_i)$ | Min | Avg. | Max | | Min | Avg. | Max |
| 16-2-a | 16 | 2 | 3 | 0.13 | 0.63 | 0.31 | 7.00 | 120 | 204.00 | 60 | 135.00 | 450 | 36.94 | 4 | 10.88 | 20 |
| 16-2-b | 16 | 2 | 3 | 0.13 | 0.44 | 0.25 | 8.00 | 120 | 352.50 | 30 | 180.00 | 720 | 32.63 | 5 | 10.00 | 13 |
| 16-2-c | 16 | 2 | 3 | 0.13 | 0.50 | 0.38 | 8.00 | 150 | 485.00 | 60 | 271.88 | 720 | 34.13 | 4 | 9.56 | 19 |
| 17-2-a | 17 | 2 | 3 | 0.12 | 0.53 | 0.41 | 8.00 | 120 | 244.29 | 20 | 181.18 | 720 | 36.35 | 5 | 11.76 | 31 |
| 17-2-b | 17 | 2 | 3 | 0.12 | 0.59 | 0.24 | 8.00 | 270 | 540.00 | 60 | 181.76 | 720 | 34.41 | 4 | 10.24 | 16 |
| 17-2-c | 17 | 2 | 3 | 0.18 | 0.53 | 0.41 | 8.00 | 120 | 364.29 | 30 | 213.53 | 720 | 31.06 | 3 | 8.24 | 12 |
| 18-2-a | 18 | 2 | 3 | 0.11 | 0.50 | 0.33 | 8.00 | 120 | 480.00 | 30 | 283.33 | 720 | 33.50 | 3 | 9.06 | 15 |
| 18-2-b | 18 | 2 | 3 | 0.11 | 0.44 | 0.39 | 8.00 | 150 | 471.43 | 20 | 231.11 | 720 | 32.83 | 7 | 9.61 | 14 |
| 18-2-c | 18 | 2 | 3 | 0.11 | 0.56 | 0.33 | 8.00 | 120 | 290.00 | 20 | 196.11 | 720 | 33.33 | 5 | 10.11 | 19 |
| 19-2-a | 19 | 2 | 3 | 0.11 | 0.32 | 0.32 | 8.00 | 120 | 430.00 | 30 | 197.37 | 720 | 32.84 | 4 | 9.26 | 16 |
| 19-2-b | 19 | 2 | 3 | 0.11 | 0.68 | 0.37 | 8.00 | 120 | 450.00 | 30 | 241.53 | 720 | 33.79 | 5 | 10.68 | 18 |
| 19-2-c | 19 | 2 | 3 | 0.11 | 0.53 | 0.53 | 8.00 | 120 | 432.00 | 60 | 287.37 | 720 | 33.63 | 5 | 10.21 | 18 |
| 20-2-a | 20 | 2 | 3 | 0.10 | 0.60 | 0.60 | 8.00 | 120 | 272.50 | 60 | 226.50 | 720 | 35.40 | 4 | 10.60 | 31 |
| 20-2-b | 20 | 2 | 3 | 0.15 | 0.45 | 0.35 | 8.00 | 120 | 424.29 | 60 | 219.00 | 720 | 35.25 | 5 | 10.80 | 21 |
| 20-2-c | 20 | 2 | 3 | 0.10 | 0.50 | 0.40 | 8.00 | 150 | 315.00 | 30 | 192.00 | 450 | 34.50 | 4 | 11.20 | 33 |
| 21-2-a | 21 | 2 | 3 | 0.10 | 0.62 | 0.33 | 8.00 | 120 | 270.00 | 30 | 227.14 | 720 | 36.43 | 5 | 11.86 | 21 |
| 21-2-b | 21 | 2 | 3 | 0.10 | 0.52 | 0.24 | 8.00 | 120 | 366.00 | 60 | 147.14 | 720 | 32.43 | 3 | 9.29 | 16 |
| 21-2-c | 21 | 2 | 3 | 0.10 | 0.57 | 0.52 | 7.55 | 120 | 406.36 | 60 | 261.43 | 720 | 30.71 | 4 | 7.95 | 11 |
| 22-2-a | 22 | 2 | 3 | 0.14 | 0.73 | 0.45 | 7.50 | 120 | 336.00 | 20 | 223.18 | 720 | 34.91 | 1 | 10.77 | 22 |
| 22-2-b | 22 | 2 | 3 | 0.14 | 0.59 | 0.41 | 8.00 | 120 | 466.67 | 60 | 274.09 | 720 | 33.82 | 3 | 9.86 | 18 |
| 22-2-c | 22 | 2 | 3 | 0.09 | 0.64 | 0.45 | 8.00 | 150 | 507.00 | 60 | 346.36 | 720 | 31.91 | 5 | 9.00 | 13 |
| 23-2-a | 23 | 2 | 3 | 0.17 | 0.70 | 0.43 | 8.00 | 120 | 327.00 | 30 | 242.61 | 720 | 34.04 | 5 | 10.35 | 16 |
| 23-2-b | 23 | 2 | 3 | 0.09 | 0.52 | 0.30 | 8.00 | 120 | 368.57 | 60 | 270.00 | 720 | 32.74 | 5 | 9.30 | 14 |
| 23-2-c | 23 | 2 | 3 | 0.09 | 0.57 | 0.35 | 8.00 | 120 | 240.00 | 20 | 167.83 | 720 | 33.39 | 3 | 9.26 | 15 |

**Table 2**
Characteristics of the scattered instances.

| Instance | $n$ | $\lvert K\rvert$ | $\lvert H\rvert$ | $p_{o>1}$ | $p_0$ | $i \in P_1 \cup D_1$ | | | Avg. $f_i$ | $t_{i,i+n}, i \in P$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Min | Avg. | Max | | Min | Avg. | Max |
| 16-2-d | 16 | 2 | 3 | 0.13 | 0 | 60 | 221.25 | 720 | 41.81 | 5 | 13.94 | 38 |
| 16-2-e | 16 | 2 | 3 | 0.13 | 0 | 60 | 210.00 | 720 | 34.69 | 4 | 10.81 | 30 |
| 16-2-f | 16 | 2 | 3 | 0.13 | 0 | 20 | 200.00 | 720 | 34.13 | 4 | 9.56 | 22 |
| 17-2-d | 17 | 2 | 3 | 0.18 | 0 | 20 | 250.00 | 720 | 33.18 | 6 | 9.65 | 16 |
| 17-2-e | 17 | 2 | 3 | 0.12 | 0 | 20 | 187.65 | 720 | 30.18 | 3 | 6.94 | 11 |
| 17-2-f | 17 | 2 | 3 | 0.12 | 0 | 20 | 147.65 | 720 | 32.29 | 4 | 9.00 | 14 |
| 18-2-d | 18 | 2 | 3 | 0.11 | 0 | 60 | 156.67 | 720 | 33.33 | 3 | 8.67 | 21 |
| 18-2-e | 18 | 2 | 3 | 0.11 | 0 | 60 | 156.67 | 720 | 34.33 | 4 | 9.83 | 21 |
| 18-2-f | 18 | 2 | 3 | 0.11 | 0 | 20 | 217.78 | 720 | 33.00 | 5 | 9.00 | 15 |
| 19-2-d | 19 | 2 | 3 | 0.11 | 0 | 30 | 148.42 | 720 | 32.53 | 3 | 8.68 | 16 |
| 19-2-e | 19 | 2 | 3 | 0.11 | 0 | 20 | 146.32 | 720 | 32.53 | 3 | 8.68 | 14 |
| 19-2-f | 19 | 2 | 3 | 0.16 | 0 | 30 | 154.74 | 720 | 33.47 | 3 | 8.79 | 19 |
| 20-2-d | 20 | 2 | 3 | 0.15 | 0 | 60 | 165.00 | 450 | 32.85 | 3 | 8.95 | 22 |
| 20-2-e | 20 | 2 | 3 | 0.10 | 0 | 30 | 174.00 | 450 | 33.15 | 3 | 9.25 | 18 |
| 20-2-f | 20 | 2 | 3 | 0.10 | 0 | 20 | 158.50 | 720 | 31.95 | 3 | 7.90 | 15 |
| 21-2-d | 21 | 2 | 3 | 0.10 | 0 | 20 | 195.24 | 720 | 32.57 | 5 | 9.67 | 16 |
| 21-2-e | 21 | 2 | 3 | 0.10 | 0 | 20 | 183.81 | 720 | 34.29 | 3 | 9.43 | 22 |
| 21-2-f | 21 | 2 | 3 | 0.14 | 0 | 30 | 162.86 | 720 | 35.00 | 6 | 10.71 | 16 |

The first four columns in Table 1, namely *Instance*, $n$, $|K|$, and $|H|$, give the name, the number of requests, the number of vehicles (drivers), and the number of assistants in an instance, respectively. Column $p_{o>1}$ gives the percentage of requests that require more than one assistant, and Column $p_0$ gives the percentage of requests whose pickup node or delivery node is the depot. In most of the normal instances, more than half of the requests have the depot as their pickup node or delivery node, and only a small proportion of clients require more than one assistant. These proportions are consistent with those in the NEATS dataset. The percentage of nodes that belongs to $P_1 \cup D_1$ is shown in Column $p_{dp}$. Because when implementing Model II, only nodes at the depot with large time window are grouped together, the percentage in Column $p_{dp}$ may be smaller than that in Column $p_0$. The average service time and average (minimal) time window width of the nodes that are grouped together, i.e. nodes in $P_1 \cup D_1$, are shown in Columns *avg.* $s_i$ and *avg.* $(l_i - e_i)$ (*min* $(l_i - e_i)$), respectively. From these columns, we can see that the time window's width of the nodes in $P_1 \cup D_1$ is at least 120 min and is quite large with regard to their service time. The next three columns contain the statistical information about width of all clients' time window. Here, the width of a client $i$'s time window is set as $\min\{l_i - e_i, l_{i+n} - e_{i+n}\}$. The average time window's width is more than 60 min, while the largest time window width is 720 min. The average maximum riding time of the clients is presented in Column *avg.* $f_i$. The next three columns present the statistical information about the minimum travel time of the clients, i.e. $t_{i,i+n}$. We can see that in most of the instances, a client's origin is not too far away from his/her destination and is about 10 min driving. Given the fact that one trip can last for as long as 150 min, we can say that the maximum riding time constraints are quite important. If a client's minimum travel time is only 15 min, but he/she spends more than 2 h on a vehicle, he/she may feel quite uncomfortable.

## 6.2. Effects of valid inequalities

Tables 3–5 summarize the lower bounds of Model I and Model II obtained at the root node (after applying the problem reduction step) under different scenarios. Column *LP* reports the lower bounds without any valid inequalities. The next eight columns list the lower bounds after adding one of the following inequalities: the trip number inequalities (81) (*TNI*), the strengthened subtour elimination inequalities (54) and (55) (*SEI*), the strengthened precedence inequalities (63) (*PI*), the capacity inequalities (73) (*CI*), the infeasible path elimination inequalities (74)–(76) (*IPEI*), the lunch break inequalities (83) and (82) (*LBI*), the fork inequalities (77) and (78) (*FI*), and the reachability inequalities (79) and (80) (*RI*). Column *ALL* reports the lower bounds achieved when adding all of the eight families of inequalities. The lower bounds obtained after adding the default inequalities generated by CPLEX are presented in column *CPLEX*. The last column, *FINAL*, gives the lower bounds after adding all of the inequalities, including the eight families of inequalities and the default inequalities generated by CPLEX.

From Tables 3–5, we can observe that except for the lunch break inequalities and the infeasible path elimination inequalities, all of the inequalities can improve the lower bounds at the root node. In terms of average improvement, the strengthened subtour elimination inequalities, the reachability inequalities, and the fork inequalities are among the best three under different scenarios, followed by the capacity inequalities and the precedence inequalities. The trip number inequality is simple but turns out to be effective in the scattered instances. It improved the lower bounds by about 12% on average. Although the lunch break inequalities did not work in the root node, given the fact that they are quite simple to separate and are violated through the branch-and-cut process, we adopt them in the branch-and-cut algorithm. After adding all the eight families of inequalities, the average lower bound of Model I is lifted from 128.61 to 189.35 in the normal instances, and from

**Table 3**
Lower bounds of Model I in the normal instances.

| Instance | LP | TNI | SEI | PI | CI | IPEI | LBI | FI | RI | ALL | CPLEX | FINAL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 16-2-a | 122.35 | 122.38 | 172.00 | 141.50 | 155.34 | 122.35 | 122.35 | 169.00 | 157.62 | 194.70 | 174.17 | 206.50 |
| 16-2-b | 117.17 | 117.17 | 151.31 | 137.59 | 128.26 | 117.17 | 117.17 | 166.26 | 147.20 | 173.84 | 128.24 | 175.51 |
| 16-2-c | 149.13 | 149.13 | 184.87 | 161.50 | 162.44 | 149.13 | 149.13 | 190.75 | 188.06 | 204.29 | 175.03 | 206.46 |
| 17-2-a | 149.85 | 150.82 | 198.84 | 176.02 | 167.45 | 149.85 | 149.85 | 195.07 | 189.55 | 217.03 | 163.49 | 216.87 |
| 17-2-b | 145.92 | 145.92 | 174.36 | 156.33 | 167.00 | 145.92 | 145.92 | 188.00 | 173.01 | 197.56 | 162.59 | 197.52 |
| 17-2-c | 87.69 | 91.69 | 129.00 | 118.39 | 108.00 | 87.69 | 87.69 | 124.19 | 130.80 | 144.97 | 113.55 | 146.62 |
| 18-2-a | 110.82 | 110.82 | 159.04 | 118.62 | 135.83 | 110.84 | 110.82 | 144.24 | 151.60 | 168.03 | 146.05 | 176.87 |
| 18-2-b | 103.05 | 131.00 | 160.20 | 131.00 | 135.54 | 103.05 | 103.05 | 146.70 | 152.59 | 175.54 | 119.15 | 176.86 |
| 18-2-c | 115.16 | 115.16 | 161.56 | 126.00 | 123.00 | 115.16 | 115.16 | 160.17 | 164.67 | 180.09 | 154.53 | 182.56 |
| 19-2-a | 131.12 | 137.40 | 185.15 | 142.06 | 145.75 | 131.12 | 131.12 | 185.19 | 172.27 | 204.40 | 151.03 | 205.13 |
| 19-2-b | 100.00 | 102.00 | 146.00 | 139.00 | 117.00 | 100.00 | 100.00 | 145.59 | 147.42 | 167.48 | 114.00 | 167.85 |
| 19-2-c | 135.19 | 147.86 | 172.33 | 153.50 | 171.33 | 135.19 | 135.19 | 164.81 | 170.00 | 186.34 | 142.63 | 186.36 |
| 20-2-a | 173.17 | 188.61 | 229.00 | 203.00 | 203.50 | 173.17 | 173.17 | 226.26 | 220.65 | 240.74 | 189.45 | 244.64 |
| 20-2-b | 151.00 | 151.00 | 185.00 | 166.44 | 162.00 | 151.00 | 151.00 | 175.22 | 178.32 | 194.06 | 176.00 | 216.19 |
| 20-2-c | 154.97 | 154.97 | 211.87 | 185.55 | 177.43 | 154.97 | 154.97 | 213.24 | 214.72 | 236.92 | 208.21 | 251.25 |
| 21-2-a | 171.18 | 171.18 | 216.00 | 191.26 | 202.33 | 171.18 | 171.18 | 205.56 | 202.19 | 226.21 | 183.66 | 227.86 |
| 21-2-b | 145.11 | 145.11 | 190.64 | 164.19 | 159.28 | 145.11 | 145.11 | 188.38 | 175.18 | 204.50 | 161.24 | 205.13 |
| **21-2-c** | 97.25 | 99.67 | 110.50 | 112.15 | 119.00 | 97.25 | 97.25 | 128.35 | 97.25 | 139.59 | 103.67 | 142.40 |
| 22-2-a | 147.05 | 147.05 | 192.33 | 178.00 | 182.62 | 147.05 | 147.05 | 178.80 | 181.81 | 201.27 | 166.13 | 207.51 |
| 22-2-b | 104.94 | 107.49 | 156.25 | 136.43 | 152.00 | 104.94 | 104.94 | 140.34 | 151.60 | 168.25 | 118.00 | 167.01 |
| **22-2-c** | 91.00 | 100.00 | 101.33 | 103.17 | 139.33 | 91.00 | 91.00 | 129.01 | 91.00 | 145.02 | 96.50 | 145.19 |
| 23-2-a | 103.66 | 103.66 | 153.83 | 140.00 | 144.67 | 103.66 | 103.66 | 141.85 | 145.40 | 167.05 | 126.00 | 171.94 |
| 23-2-b | 138.70 | 141.52 | 186.01 | 163.00 | 170.00 | 138.70 | 138.70 | 186.96 | 180.93 | 206.01 | 158.43 | 210.11 |
| 23-2-c | 141.08 | 141.08 | 192.90 | 161.87 | 176.27 | 141.08 | 141.08 | 172.46 | 175.58 | 200.41 | 153.29 | 202.71 |
| Average | 128.61 | 132.19 | 171.68 | 150.27 | 154.39 | 128.61 | 128.61 | 169.43 | 164.98 | 189.35 | 149.38 | 193.21 |

**Table 4**
Lower bounds of Model II in the normal instances.

| Instance | LP | TNI | SEI | PI | CI | IPEI | LBI | FI | RI | ALL | CPLEX | FINAL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 16-2-a | 139.05 | 139.19 | 177.14 | 146.21 | 151.71 | 139.05 | 139.05 | 182.14 | 181.96 | 205.94 | 190.00 | 210.25 |
| 16-2-b | 118.84 | 118.84 | 153.00 | 130.58 | 130.17 | 118.84 | 118.84 | 169.22 | 156.00 | 177.43 | 135.83 | 182.43 |
| 16-2-c | 169.51 | 169.51 | 200.00 | 178.81 | 177.13 | 169.53 | 169.51 | 206.63 | 209.74 | 219.19 | 191.02 | 218.98 |
| 17-2-a | 167.11 | 170.57 | 205.63 | 171.67 | 177.88 | 167.11 | 167.11 | 212.65 | 208.77 | 223.88 | 192.00 | 224.16 |
| 17-2-b | 156.80 | 156.80 | 176.34 | 165.16 | 167.00 | 156.98 | 156.80 | 195.40 | 176.50 | 207.42 | 181.23 | 209.24 |
| 17-2-c | 100.03 | 102.30 | 133.11 | 114.92 | 111.10 | 101.92 | 100.03 | 136.87 | 151.26 | 155.51 | 132.62 | 155.82 |
| 18-2-a | 121.54 | 126.48 | 160.35 | 123.64 | 137.33 | 121.54 | 121.54 | 156.84 | 162.97 | 175.86 | 168.43 | 192.92 |
| 18-2-b | 118.16 | 118.16 | 162.33 | 136.50 | 136.91 | 119.00 | 118.16 | 164.21 | 166.25 | 182.82 | 141.18 | 194.84 |
| 18-2-c | 119.00 | 119.00 | 164.33 | 135.96 | 125.50 | 125.54 | 119.00 | 166.30 | 180.23 | 198.23 | 183.29 | 204.11 |
| 19-2-a | 142.25 | 143.70 | 190.25 | 146.49 | 152.50 | 142.90 | 142.25 | 197.39 | 184.50 | 217.83 | 166.00 | 218.79 |
| 19-2-b | 119.30 | 119.30 | 157.67 | 144.77 | 126.22 | 123.47 | 119.30 | 158.14 | 157.03 | 186.65 | 145.10 | 191.47 |
| 19-2-c | 139.64 | 142.22 | 174.15 | 147.67 | 178.50 | 139.64 | 139.64 | 158.75 | 174.71 | 186.57 | 156.07 | 184.48 |
| 20-2-a | 183.91 | 190.58 | 231.94 | 189.14 | 210.83 | 183.91 | 183.91 | 238.21 | 241.21 | 250.36 | 205.36 | 249.93 |
| 20-2-b | 171.01 | 171.01 | 191.40 | 181.63 | 178.25 | 171.01 | 171.01 | 190.53 | 196.69 | 204.73 | 205.80 | 225.85 |
| 20-2-c | 167.49 | 170.06 | 223.50 | 177.25 | 174.97 | 167.49 | 167.49 | 226.21 | 224.50 | 248.56 | 226.72 | 260.75 |
| 21-2-a | 177.76 | 177.76 | 220.33 | 201.08 | 203.12 | 179.56 | 177.76 | 226.71 | 221.18 | 245.23 | 215.22 | 257.10 |
| 21-2-b | 152.02 | 152.02 | 194.87 | 164.01 | 164.66 | 152.02 | 152.02 | 192.91 | 186.92 | 209.87 | 178.38 | 212.26 |
| 21-2-c | 105.38 | 109.39 | 148.57 | 114.77 | 124.00 | 105.38 | 105.38 | 138.73 | 156.26 | 161.70 | 109.54 | 161.86 |
| 22-2-a | 148.00 | 148.00 | 193.50 | 168.56 | 180.50 | 152.50 | 148.00 | 198.46 | 205.46 | 227.63 | 185.50 | 240.19 |
| 22-2-b | 105.50 | 105.60 | 162.25 | 132.00 | 153.11 | 105.50 | 105.50 | 145.36 | 160.50 | 175.20 | 124.19 | 178.43 |
| 22-2-c | 95.08 | 95.33 | 152.25 | 123.50 | 139.33 | 98.00 | 95.08 | 145.90 | 147.49 | 169.06 | 112.65 | 169.83 |
| 23-2-a | 107.92 | 107.92 | 158.50 | 134.80 | 146.67 | 107.92 | 107.92 | 153.28 | 163.50 | 187.78 | 137.83 | 189.71 |
| 23-2-b | 143.74 | 144.23 | 191.20 | 163.28 | 173.52 | 143.74 | 143.74 | 198.10 | 192.00 | 219.62 | 165.44 | 220.36 |
| 23-2-c | 146.68 | 146.68 | 189.00 | 152.58 | 177.56 | 146.68 | 146.68 | 182.42 | 184.66 | 205.61 | 161.36 | 207.21 |
| Average | 138.16 | 139.36 | 179.65 | 151.87 | 158.27 | 139.13 | 138.16 | 180.89 | 182.93 | 201.78 | 167.11 | 206.71 |

155.54 to 226.24 in the scattered instances, and the average lower bound of Model II is lifted from 138.16 to 201.78 in the normal instances. All of those improvements are more than 45%, which are much better than the improvement brought by the default inequalities generated by CPLEX. After all of the inequalities are added, the lower bound is further improved, which is almost 50 % higher than the original lower bound without any cut. Note that in Table 3, two instances are marked in bold, because in these instances, the label-setting algorithm in the preprocessing stage cannot terminate within 20 min.

**Table 5**
Lower bounds of Model I in the scattered instances.

| Instance | LP | TNI | SEI | PI | CI | IPEI | LBI | FI | RI | ALL | CPLEX | FINAL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 16-2-d | 229.13 | 248.00 | 289.17 | 242.33 | 240.00 | 229.13 | 229.13 | 285.73 | 294.16 | 306.81 | 272.48 | 315.10 |
| 16-2-e | 169.67 | 193.92 | 226.82 | 182.17 | 178.33 | 169.67 | 169.67 | 220.24 | 224.99 | 247.09 | 193.10 | 254.02 |
| 16-2-f | 165.03 | 191.55 | 207.53 | 177.74 | 176.45 | 165.03 | 165.03 | 207.26 | 210.00 | 226.47 | 187.55 | 231.19 |
| 17-2-d | 116.17 | 135.33 | 173.00 | 147.79 | 143.33 | 116.17 | 116.17 | 178.29 | 164.39 | 197.01 | 156.50 | 203.18 |
| 17-2-e | 88.00 | 112.00 | 144.50 | 103.75 | 106.00 | 88.00 | 88.00 | 125.57 | 140.02 | 159.91 | 123.00 | 171.54 |
| 17-2-f | 142.89 | 144.81 | 174.50 | 155.22 | 149.14 | 142.89 | 142.89 | 179.81 | 180.67 | 194.60 | 182.82 | 206.20 |
| 18-2-d | 128.63 | 167.75 | 195.59 | 148.67 | 139.02 | 128.63 | 128.63 | 190.28 | 196.87 | 219.49 | 160.61 | 228.13 |
| 18-2-e | 158.10 | 178.50 | 223.16 | 180.63 | 177.31 | 158.10 | 158.10 | 217.90 | 212.57 | 248.75 | 206.83 | 259.95 |
| 18-2-f | 154.71 | 169.14 | 195.33 | 161.50 | 165.12 | 154.71 | 154.71 | 194.23 | 201.71 | 211.91 | 181.33 | 226.42 |
| 19-2-d | 147.00 | 167.00 | 199.03 | 155.50 | 164.00 | 147.00 | 147.00 | 179.07 | 190.12 | 213.52 | 174.00 | 224.64 |
| 19-2-e | 178.70 | 193.33 | 208.88 | 182.56 | 182.00 | 178.70 | 178.70 | 208.21 | 211.92 | 224.16 | 211.87 | 234.30 |
| 19-2-f | 166.21 | 183.13 | 222.33 | 179.85 | 174.86 | 166.21 | 166.21 | 216.44 | 219.16 | 243.49 | 206.00 | 245.41 |
| 20-2-d | 174.37 | 188.00 | 227.74 | 185.07 | 194.82 | 174.37 | 174.37 | 209.82 | 219.78 | 245.86 | 211.00 | 262.36 |
| 20-2-e | 170.48 | 183.77 | 218.17 | 183.47 | 181.11 | 170.76 | 170.48 | 215.50 | 207.07 | 241.03 | 193.00 | 249.16 |
| 20-2-f | 165.90 | 178.33 | 202.75 | 178.00 | 179.60 | 165.90 | 165.90 | 199.17 | 204.22 | 216.30 | 200.20 | 235.33 |
| 21-2-d | 122.83 | 144.31 | 178.56 | 152.11 | 147.68 | 122.83 | 122.83 | 189.22 | 173.97 | 211.70 | 172.28 | 233.43 |
| 21-2-e | 194.44 | 213.37 | 234.67 | 203.00 | 198.61 | 194.44 | 194.44 | 229.56 | 224.75 | 246.11 | 221.63 | 251.16 |
| 21-2-f | 127.47 | 144.76 | 189.54 | 151.67 | 173.28 | 128.00 | 127.47 | 176.47 | 191.12 | 218.10 | 172.02 | 235.32 |
| Average | 155.54 | 174.28 | 206.18 | 170.61 | 170.59 | 155.59 | 155.54 | 201.26 | 203.75 | 226.24 | 190.35 | 237.05 |

**Table 6**
Comparisons with CPLEX in the normal instances.

| Instance | Model I | | | | | | | Model II | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CPLEX | | | Branch-and-cut | | | UB | CPLEX | | | Branch-and-cut | | | UB |
| | LB | Time | Nodes | LB | Time | Nodes | | LB | Time | Nodes | LB | Time | Nodes | |
| 16-2-a | 203.21 | 240 | 938,930 | **238.00** | 97 | 66,916 | 238 | **238.00** | 4.88 | 33,159 | **238.00** | 1.87 | 5988 | 238 |
| 16-2-b | 177.83 | 240 | 1,282,160 | **221.00** | 70 | 40,568 | 221 | **221.00** | 151.82 | 605,907 | **221.00** | 3.02 | 5507 | 221 |
| 16-2-c | 196.66 | 240 | 1,024,693 | **241.00** | 31 | 13,407 | 241 | **241.00** | 117.17 | 497,604 | **241.00** | 1.87 | 3969 | 241 |
| 17-2-a | 198.75 | 240 | 816,513 | **254.00** | 91 | 36,092 | 254 | **254.00** | 21.70 | 90,039 | **254.00** | 2.20 | 3981 | 254 |
| 17-2-b | 184.04 | 240 | 1,114,237 | **224.00** | 26 | 10,788 | 224 | **224.00** | 27.45 | 160,854 | **224.00** | 0.85 | 1145 | 224 |
| 17-2-c | 136.11 | 240 | 1,148,229 | **176.00** | 46 | 22,341 | 176 | **174.00** | 8.30 | 67,036 | **174.00** | 0.88 | 1545 | 174 |
| 18-2-a | 163.16 | 174 | 1,214,147 | **220.00** | 62 | 22,159 | 220 | 192.43 | 240.00 | 1,553,791 | **220.00** | 0.92 | 1499 | 220 |
| 18-2-b | 150.18 | 240 | 1,974,441 | **222.00** | 158 | 35,094 | 222 | **222.00** | 75.18 | 637,959 | **222.00** | 0.75 | 774 | 222 |
| 18-2-c | 212.21 | 240 | 2,865,129 | **236.00** | 34 | 15,660 | 236 | **236.00** | 3.15 | 59,582 | **236.00** | 0.58 | 1303 | 236 |
| 19-2-a | 173.23 | 240 | 747,458 | 231.71 | 240 | 115,031 | 236 | 188.50 | 240.00 | 814,651 | **236.00** | 31.45 | 53,749 | 236 |
| 19-2-b | 133.55 | 240 | 563,599 | 191.59 | 240 | 18,126 | 237 | 197.02 | 240.00 | 1,028,248 | **226.00** | 30.03 | 26,892 | 226 |
| 19-2-c | 160.14 | 240 | 831,541 | 197.19 | 240 | 17,462 | 236 | **214.00** | 53.88 | 322,475 | **214.00** | 38.85 | 30,776 | 214 |
| 20-2-a | 213.09 | 240 | 442,479 | 258.19 | 240 | 19,390 | – | 245.18 | 240.00 | 475,167 | **287.00** | 96.33 | 64,977 | 287 |
| 20-2-b | 187.52 | 240 | 985,266 | 228.50 | 240 | 36,256 | – | 251.41 | 240.00 | 921,325 | **266.00** | 21.67 | 17,637 | 266 |
| 20-2-c | 234.28 | 219 | 1,235,395 | 274.63 | 240 | 70,565 | – | **308.00** | 21.08 | 244,896 | **308.00** | 15.55 | 23,675 | 308 |
| 21-2-a | 218.90 | 240 | 389,088 | 255.65 | 240 | 22,509 | – | 253.73 | 240.00 | 1,051,553 | **289.00** | 22.37 | 12,280 | 289 |
| 21-2-b | 172.30 | 240 | 389,380 | 229.78 | 240 | 59,805 | 237 | 194.95 | 240.00 | 505,620 | **237.00** | 14.20 | 13,115 | 237 |
| 21-2-c | 111.01 | 240 | 519,903 | 154.82 | 240 | 53,505 | 194 | 125.57 | 240.00 | 835,632 | **176.00** | 41.40 | 35,023 | 176 |
| 22-2-a | 188.18 | 240 | 431,185 | 226.17 | 240 | 11,044 | – | **291.00** | 83.58 | 886,392 | **291.00** | 34.90 | 23,833 | 291 |
| 22-2-b | 124.27 | 240 | 236,165 | 175.76 | 240 | 11,394 | – | 143.48 | 240.00 | 322,354 | 193.44 | 240.00 | 34,959 | 230 |
| 22-2-c | 100.34 | 240 | 504,600 | 152.78 | 240 | 31,938 | – | 140.68 | 240.00 | 924,675 | 186.94 | 240.00 | 50,937 | 225 |
| 23-2-a | 137.24 | 240 | 517,102 | 192.65 | 240 | 11,502 | – | 172.34 | 240.00 | 897,923 | 229.32 | 240.00 | 41,578 | 255 |
| 23-2-b | 181.30 | 240 | 338,058 | 227.90 | 240 | 23,436 | – | 220.50 | 240.00 | 578,047 | 247.50 | 240.00 | 47,661 | 291 |
| 23-2-c | 162.25 | 240 | 452,766 | 223.10 | 240 | 24,555 | – | 189.06 | 240.00 | 871,738 | 232.90 | 240.00 | 85,757 | 281 |
| Average | 171.66 | 236 | 873,436 | 218.85 | 176 | 32,898 | | 214.08 | 153.68 | 599,443 | 235.42 | 64.99 | 24,523 | |

## 6.3. Integer solution results

Table 6 summarizes the integer solution results obtained by CPLEX and the branch-and-cut algorithm based on Model I and Model II for the normal instances. All of the algorithms were run on a single thread with a time limit of 4 h. In the branch-and-cut algorithm, the valid inequalities implemented include the eight families of inequalities and the default inequalities generated by CPLEX. Columns *LB* report the best lower bounds found in each instance within the 4 h computational time. Columns *Time* (in minutes) give the running time in minutes for an algorithm either to find an optimal solution or to terminate. The number of nodes explored by each algorithm is presented in Columns *Nodes*. Columns *UB* report the best upper bounds obtained by both CPLEX and the branch-and-cut algorithm. Note that if the best upper bound of an instance is equal to the best lower bound of this instance obtained by an algorithm, it means that this algorithm finds an optimal

**Table 7**
Comparisons with CPLEX in the scattered instances.

| Instance | CPLEX | | | Branch-and-Cut | | | Best UB |
|---|---|---|---|---|---|---|---|
| | Best LB | Time (min) | Nodes | Best LB | Time (min) | Nodes | |
| 16-2-d | 314.30 | 240.00 | 1,914,533 | **366.00** | 9.37 | 10,900 | 366 |
| 16-2-e | 249.96 | 240.00 | 1,150,696 | **293.00** | 15.43 | 11,275 | 293 |
| 16-2-f | 233.61 | 240.00 | 4,378,114 | **287.00** | 28.68 | 27,777 | 287 |
| 17-2-d | 201.50 | 200.57 | 1,952,935 | **249.00** | 26.10 | 26,507 | 249 |
| 17-2-e | 146.17 | 231.40 | 1,648,291 | **207.00** | 6.17 | 4853 | 207 |
| 17-2-f | 231.00 | 240.00 | 3,597,255 | **282.00** | 43.93 | 36,411 | 282 |
| 18-2-d | 205.02 | 240.00 | 1,080,849 | **274.00** | 62.62 | 50,118 | 274 |
| 18-2-e | 263.53 | 240.00 | 1,237,929 | **330.00** | 22.15 | 10,201 | 330 |
| 18-2-f | 218.00 | 240.00 | 1,164,000 | **287.00** | 165.88 | 104,660 | 287 |
| 19-2-d | 197.33 | 240.00 | 1,079,694 | **284.00** | 71.17 | 26,016 | 284 |
| 19-2-e | 263.76 | 240.00 | 1,402,231 | **285.00** | 21.50 | 12,950 | 285 |
| 19-2-f | 241.88 | 240.00 | 1,104,415 | 301.05 | 240.00 | 155,292 | 304 |
| 20-2-d | 280.41 | 240.00 | 1,129,962 | 308.36 | 240.00 | 64,200 | 329 |
| 20-2-e | 203.65 | 240.00 | 606,945 | **285.00** | 32.53 | 6510 | 285 |
| 20-2-f | 236.95 | 240.00 | 1,627,844 | **272.00** | 71.42 | 31,682 | 272 |
| 21-2-d | 213.03 | 184.33 | 1,445,678 | 276.05 | 240.00 | 53,359 | 322 |
| 21-2-e | 244.14 | 240.00 | 2,691,222 | 289.55 | 240.00 | 21,079 | 324 |
| 21-2-f | 218.20 | 240.00 | 1,590,943 | 294.67 | 240.00 | 95,200 | 317 |
| Average | 231.25 | 234.24 | 1,711,308 | 287.26 | 98.72 | 41,611 | |

solution of the instance. The bold font indicates that a lower bound is an optimal solution. In some instances, for example instance 17-2-d, the lower bound is less than the upper bound, and at the same time, the running time is less than 4 h. This indicates that the algorithm ran out of memory limit and was forced to terminate before an optimal solution was found or the time limit was reached. Hyphen "–" in Columns *UB* indicates that neither CPLEX nor the branch-and-cut algorithm successfully found a feasible solution in the corresponding instances.

As we can see from Table 6, the branch-and-cut algorithm can solve more instances to optimality than CPLEX. In Model I, the branch-and-cut algorithm can solve 9 instances to optimality, whereas CPLEX cannot solve any instances to optimality. In Model II, the branch-and-cut algorithm can solve 19 instances to optimality, whereas CPLEX can solve 11 instances to optimality. In most of the solvable instances, i.e. the instances that are solved to optimality by the branch-and-cut within 4 h, the branch-and-cut algorithm used less time and explored fewer nodes to get an optimal solution than CPLEX. In the unsolvable instances, the best lower bounds obtained by the branch-and-cut algorithm within the time limit are much better than those obtained by CPLEX. Compared with Model I, the branch-and-cut algorithm based on Model II is more efficient. In addition, in terms of solution cost, the optimal solutions of Model I are similar to those of Model II: in only one instance, the optimal solutions of Model I and Model II are different.

Table 7 summarizes the integer solution results obtained by CPLEX and the branch-and-cut algorithm in the scattered instances. The bold font indicates that a lower bound is an optimal solution. Similar to the results in the normal instances, the branch-and-cut algorithm performs much better than CPLEX. The branch-and-cut algorithm can solve 13 instances to optimality, while CPLEX cannot solve any instances to optimality.

Tables 8–10 present detailed information about the optimal solutions obtained by the branch-and-cut algorithm for the normal instances and the scattered instances. Columns *Empty Trips* present the number of empty trips of drivers and empty tours of assistants in an optimal solution. The remaining columns show the statistics of the number of clients served in a trip, the trip duration, and the maximum vehicle capacity occupancy of a trip in terms of seats.

**Table 8**
Optimal Solutions Obtained by the Branch-and-cut Based on Model I for the Normal Instances.

| Instance | Number of empty trips | | Number of clients in a trip | | | Trip duration (min) | | | Capacity occupancy | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Drivers | Assistants | Min | Avg. | Max | Min | Avg. | Max | Min | Avg. | Max |
| 16-2-a | 1 | 2 | 2 | 3.20 | 5 | 45 | 106.60 | 150 | 0.39 | 0.62 | 0.80 |
| 16-2-b | 1 | 2 | 2 | 3.20 | 4 | 66 | 120.00 | 150 | 0.35 | 0.54 | 0.78 |
| 16-2-c | 1 | 2 | 1 | 3.20 | 5 | 27 | 105.40 | 149 | 0.35 | 0.51 | 0.67 |
| 17-2-a | 1 | 2 | 2 | 3.40 | 4 | 84 | 132.00 | 150 | 0.50 | 0.73 | 1.00 |
| 17-2-b | 2 | 3 | 3 | 4.25 | 6 | 93 | 124.50 | 150 | 0.61 | 0.72 | 0.85 |
| 17-2-c | 1 | 2 | 1 | 3.40 | 5 | 28 | 97.60 | 150 | 0.39 | 0.55 | 0.70 |
| 18-2-a | 1 | 2 | 2 | 3.60 | 6 | 72 | 113.80 | 150 | 0.39 | 0.59 | 0.89 |
| 18-2-b | 1 | 3 | 2 | 3.60 | 5 | 61 | 122.20 | 150 | 0.39 | 0.54 | 0.78 |
| 18-2-c | 1 | 2 | 1 | 3.60 | 5 | 41 | 121.00 | 150 | 0.45 | 0.60 | 0.75 |
| Average | 1.11 | 2.22 | 1.78 | 3.49 | 5.00 | 57.44 | 115.90 | 149.89 | 0.42 | 0.60 | 0.80 |

**Table 9**
Optimal solutions obtained by the branch-and-cut based on Model II for the normal instances.

| Instance | Number of empty trips | | Number of clients in a trip | | | Trip duration (min) | | | Capacity occupancy | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Drivers | Assistants | Min | Avg. | Max | Min | Avg. | Max | Min | Avg. | Max |
| 16-2-a | 1 | 2 | 1 | 2.60 | 4 | 46 | 104.80 | 150 | 0.39 | 0.62 | 0.80 |
| 16-2-b | 1 | 2 | 1 | 2.60 | 4 | 62 | 102.20 | 150 | 0.40 | 0.56 | 0.78 |
| 16-2-c | 1 | 2 | 1 | 2.20 | 3 | 27 | 113.20 | 150 | 0.35 | 0.54 | 0.67 |
| 17-2-a | 1 | 2 | 1 | 2.40 | 4 | 70 | 129.20 | 150 | 0.50 | 0.73 | 1.00 |
| 17-2-b | 2 | 3 | 2 | 3.50 | 5 | 123 | 138.50 | 150 | 0.61 | 0.72 | 0.85 |
| 17-2-c | 1 | 2 | 1 | 2.40 | 4 | 27 | 93.00 | 150 | 0.39 | 0.65 | 0.78 |
| 18-2-a | 1 | 2 | 1 | 2.80 | 5 | 63 | 104.80 | 150 | 0.39 | 0.59 | 0.89 |
| 18-2-b | 1 | 3 | 2 | 2.80 | 4 | 61 | 124.00 | 150 | 0.39 | 0.56 | 0.85 |
| 18-2-c | 1 | 2 | 1 | 3.00 | 4 | 42 | 125.60 | 150 | 0.45 | 0.61 | 0.78 |
| 19-2-a | 1 | 2 | 1 | 3.20 | 6 | 54 | 110.00 | 150 | 0.44 | 0.58 | 0.78 |
| 19-2-b | 0 | 1 | 2 | 2.17 | 3 | 62 | 84.67 | 113 | 0.39 | 0.57 | 0.83 |
| 19-2-c | 1 | 2 | 1 | 2.60 | 4 | 77 | 108.20 | 150 | 0.60 | 0.76 | 1.00 |
| 20-2-a | 1 | 2 | 1 | 2.20 | 4 | 66 | 120.40 | 150 | 0.50 | 0.75 | 0.90 |
| 20-2-b | 0 | 0 | 2 | 2.67 | 3 | 59 | 101.17 | 147 | 0.40 | 0.63 | 0.89 |
| 20-2-c | 0 | 0 | 1 | 2.50 | 5 | 68 | 109.83 | 150 | 0.33 | 0.54 | 0.67 |
| 21-2-a | 0 | 1 | 1 | 2.50 | 4 | 61 | 112.67 | 150 | 0.39 | 0.63 | 0.89 |
| 21-2-b | 1 | 3 | 3 | 3.40 | 4 | 93 | 122.60 | 150 | 0.40 | 0.53 | 0.67 |
| 21-2-c | 1 | 3 | 1 | 2.80 | 4 | 36 | 102.80 | 149 | 0.56 | 0.62 | 0.70 |
| 22-2-a | 0 | 1 | 2 | 2.67 | 3 | 56 | 114.83 | 150 | 0.40 | 0.67 | 0.90 |
| Average | 0.79 | 1.84 | 1.37 | 2.68 | 4.05 | 60.68 | 111.71 | 147.84 | 0.44 | 0.62 | 0.82 |

**Table 10**
Optimal solutions obtained by the branch-and-cut for the scattered instances.

| Instance | Number of empty trips | | Number of clients in a trip | | | Trip duration (min) | | | Capacity occupancy(%) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Drivers | Assistants | Min | Avg. | Max | Min | Avg. | Max | Min | Avg. | max |
| 16-2-d | 1 | 2 | 1 | 3.20 | 5 | 124 | 143.80 | 150 | 0.33 | 0.50 | 0.60 |
| 16-2-e | 2 | 3 | 3 | 4.00 | 5 | 120 | 139.75 | 150 | 0.45 | 0.58 | 0.67 |
| 16-2-f | 1 | 3 | 1 | 3.20 | 4 | 45 | 120.40 | 150 | 0.25 | 0.49 | 0.67 |
| 17-2-d | 1 | 2 | 1 | 3.40 | 5 | 38 | 120.00 | 150 | 0.35 | 0.58 | 0.78 |
| 17-2-e | 1 | 3 | 3 | 3.40 | 5 | 87 | 101.40 | 150 | 0.40 | 0.58 | 0.89 |
| 17-2-f | 1 | 2 | 1 | 3.40 | 5 | 46 | 123.80 | 150 | 0.25 | 0.55 | 0.83 |
| 18-2-d | 1 | 2 | 2 | 3.60 | 6 | 65 | 124.60 | 150 | 0.33 | 0.52 | 0.67 |
| 18-2-e | 1 | 2 | 1 | 3.60 | 5 | 40 | 125.00 | 150 | 0.35 | 0.56 | 1.00 |
| 18-2-f | 1 | 2 | 2 | 3.60 | 4 | 76 | 126.00 | 150 | 0.50 | 0.55 | 0.65 |
| 19-2-d | 1 | 2 | 2 | 3.80 | 5 | 75 | 125.60 | 150 | 0.25 | 0.48 | 0.65 |
| 19-2-e | 1 | 2 | 3 | 3.80 | 5 | 87 | 119.40 | 150 | 0.40 | 0.61 | 0.83 |
| 20-2-e | 1 | 2 | 2 | 4.00 | 5 | 73 | 123.00 | 150 | 0.25 | 0.55 | 0.80 |
| 20-2-f | 1 | 2 | 1 | 4.00 | 5 | 53 | 123.00 | 150 | 0.25 | 0.58 | 0.89 |
| Average | 1.08 | 2.23 | 1.77 | 3.62 | 4.92 | 71.46 | 124.29 | 150.00 | 0.34 | 0.55 | 0.76 |

As we can see from Tables 8–10, in a large proportion of instances, there exist empty trips of drivers and empty tours of assistants. This suggests that setting $|R|$ to three in our experiments is enough, and the manpower resources in the instances are sufficient. When comparing the trip duration and the vehicle capacity occupancy, we can see that the utilization of trip duration is larger than the vehicle capacity. This suggests that the branch-and-cut algorithm may be more influenced by the maximum trip duration constraints than the vehicle capacity constraints.

## 7. Conclusion

In this paper, we introduce a new variant of the DARP which simultaneously considers several practical constraints appearing in recent years in the literature. Compared with the existing DARP variants in the literature, the problem studied in this paper is closer to the actual applications in practice and hence more complicated. To formulate the problem, we propose two mathematical models, which distinguish each other by the operations on the clients who are picked up from or delivered to the depot. We propose eight families of valid inequalities based on the characteristics of the problem to strengthen the models. Based on the models and the valid inequalities, we propose a branch-and-cut algorithm to solve the problem. The branch-and-cut algorithm was extensively tested on a set of instances generated according to the data of a real world application. The computational results show that seven families of the valid inequalities improve the lower bounds yielded by the LP relaxation of the models, and the proposed branch-and-cut algorithm can optimally solve instances with up to 22 requests and outperforms CPLEX in the test instances.

There are two possible directions to continue research on this problem. The first direction is to design more powerful exact algorithms which are capable to solve practical size instances. Branch-and-price is one of the most powerful exact algorithms to solve many routing problems, including the DARP, and seems to be promising in this problem. However, the problem studied in this paper cannot be directly formulated into a stronger set-partitioning (set-covering) model, which is a prerequisite for the success of a branch-and-price algorithm. Another direction is to design efficient heuristics to solve this problem. In this problem, the most difficult parts to handle by a heuristic are the maximum riding time constraints and the manpower planning, because the feasibility of these two parts are difficult to check. Therefore, we expect a successful heuristic for this problem to have special efforts to handle these two parts. For both of the two directions, the computational study in this paper can serve as a reference to evaluate the performance of a solution approach on this problem in the future.

## Acknowledgement

This work was supported in part by the Hong Kong Research Grants Council (Grant No. T32-102/14N).

## References

Aldaihani, M., Dessouky, M.M., 2003. Hybrid scheduling methods for paratransit operations. Computers & Industrial Engineering 45 (1), 75–96.
Attanasio, A., Cordeau, J.-F., Ghiani, G., Laporte, G., 2004. Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. Parallel Computing 30 (3), 377–387.
Augerat, P., Belenguer, J.M., Benavent, E., Corbéran, A., Naddef, D., 1998. Separating capacity constraints in the CVRP using tabu search. European Journal of Operational Research 106 (2), 546–557.
Balas, E., Fischetti, M., Pulleyblank, W.R., 1995. The precedence-constrained asymmetric traveling salesman polytope. Mathematical Programming 68 (1–3), 241–265.
Baldacci, R., Bartolini, E., Mingozzi, A., 2011. An exact algorithm for the pickup and delivery problem with time windows. Operations Research 59 (2), 414–426.
Bard, J.F., Kontoravdis, G., Yu, G., 2002. A branch-and-cut procedure for the vehicle routing problem with time windows. Transportation Science 36 (2), 250–269.
Belenguer, J.-M., Benavent, E., Prins, C., Prodhon, C., Wolfler Calvo, R., 2011. A branch-and-cut method for the capacitated location-routing problem. Computers & Operations Research 38 (6), 931–941.
Bent, R., Hentenryck, P.V., 2006. A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. Computers & Operations Research 33 (4), 875–893.
Braekers, K., Caris, A., Janssens, G.K., 2014. Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. Transportation Research Part B 67, 166–186.
Colorni, A., Righini, G., 2001. Modeling and optimizing dynamic dial-a-ride problems. International Transactions in Operational Research 8 (2), 155–166.
Cordeau, J.-F., 2006. A branch-and-cut algorithm for the dial-a-ride problem. Operations Research 54 (3), 573–586.
Cordeau, J.-F., Laporte, G., 2003a. The dial-a-ride problem (DARP): variants, modeling issues and algorithms. Quarterly Journal of the Belgian, French and Italian Operations Research Societies 1 (2), 89–101.
Cordeau, J.-F., Laporte, G., 2003b. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transportation Research Part B 37 (6), 579–594.
Cordeau, J.-F., Laporte, G., 2007. The dial-a-ride problem: models and algorithms. Annals of Operations Research 153 (1), 29–46.
Coslovich, L., Pesenti, R., Ukovich, W., 2006. A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. European Journal of Operational Research 175 (3), 1605–1615.
Desaulniers, G., 2010. Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. Operations Research 58 (1), 179–192.
Desrosiers, J., Dumas, Y., Soumis, F., 1986. A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. American Journal of Mathematical and Management Sciences 6 (3–4), 301–325.
Diana, M., Dessouky, M.M., 2004. A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. Transportation Research Part B 38 (6), 539–557.
Dumas, Y., Desrosiers, J., Soumis, F., 1991. The pickup and delivery problem with time windows. European Journal of Operational Research 54 (1), 7–22.
Gschwind, T., Irnich, S., 2014. Effective handling of dynamic time windows and its application to solving the dial-a-ride problem. Transportation Science.
Hernandez, F., Feillet, D., Giroudeau, R., Naud, O., 2013. A new exact algorithm to solve the multi-trip vehicle routing problem with time windows and limited duration. 4OR.
Jaw, J.-J., Odoni, A.R., Psaraftis, H.N., Wilson, N.H., 1986. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. Transportation Research Part B 20 (3), 243–257.
Jung, S., Haghani, A., 2000. Genetic algorithm for a pickup and delivery problem with time windows. Transportation Research Record: Journal of the Transportation Research Board 1733 (1), 1–7.
Kirchler, D., Wolfler Calvo, R., 2013. A granular tabu search algorithm for the dial-a-ride problem. Transportation Research Part B 56, 120–135.
Kohl, N., Desrosiers, J., Madsen, O.B., Solomon, M.M., Soumis, F., 1999. 2-Path cuts for the vehicle routing problem with time windows. Transportation Science 33 (1), 101–116.
Li, H., Lim, A., 2003. A metaheuristic for the pickup and delivery problem with time windows. International Journal on Artificial Intelligence Tools 12 (02), 173–186.
Lim, A., Zhang, Z., Qin, H., in press. Pickup and delivery service with manpower planning in Hong Kong public hospitals. Transportation Science.
Luo, Y., Schonfeld, P., 2007. A rejected-reinsertion heuristic for the static dial-a-ride problem. Transportation Research Part B 41 (7), 736–755.
Lysgaard, J., 2006. Reachability cuts for the vehicle routing problem with time windows. European Journal of Operational Research 175 (1), 210–223.
Lysgaard, J., Letchford, A.N., Eglese, R.W., 2004. A new branch-and-cut algorithm for the capacitated vehicle routing problem. Mathematical Programming 100 (2), 423–445.
Madsen, O.B., Ravn, H.F., Rygaard, J.M., 1995. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. Annals of Operations Research 60 (1), 193–208.
Melachrinoudis, E., Ilhan, A.B., Min, H., 2007. A dial-a-ride problem for client transportation in a health-care organization. Computers & Operations Research 34 (3), 742–759.
Nanry, W.P., Wesley Barnes, J., 2000. Solving the pickup and delivery problem with time windows using reactive tabu search. Transportation Research Part B 34 (2), 107–121.
Padberg, M., Rinaldi, G., 1991. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. SIAM Review 33 (1), 60–100.
Pankratz, G., 2005. A grouping genetic algorithm for the pickup and delivery problem with time windows. OR Spectrum 27 (1), 21–41.
Paquette, J., Cordeau, J.-F., Laporte, G., Pascoal, M., 2013. Combining multicriteria analysis and tabu search for dial-a-ride problems. Transportation Research Part B 52, 1–16.

Parragh, S.N., 2011. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. Transportation Research Part C 19 (5), 912–930.
Parragh, S.N., Doerner, K.F., Hartl, R.F., 2008. A survey on pickup and delivery problems. Journal für Betriebswirtschaft 58 (1), 21–51.
Parragh, S.N., Cordeau, J.-F., Doerner, K.F., Hartl, R.F., 2012. Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints. OR Spectrum 34 (3), 593–633.
Psaraftis, H.N., 1980. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. Transportation Science 14 (2), 130–154.
Psaraftis, H.N., 1983. An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. Transportation Science 17 (3), 351–357.
Qu, Y., Bard, J.F., 2013. The heterogeneous pickup and delivery problem with configurable vehicle capacity. Transportation Research Part C: Emerging Technologies 32, 1–20.
Qu, Y., Bard, J.F., 2014. A branch-and-price-and-cut algorithm for heterogeneous pickup and delivery problems with configurable vehicle capacity. Transportation Science.
Ropke, S., Cordeau, J.-F., 2009. Branch and cut and price for the pickup and delivery problem with time windows. Transportation Science 43 (3), 267–286.
Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Science 40 (4), 455–472.
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.
Ruland, K., Rodin, E., 1997. The pickup and delivery problem: faces and branch-and-cut algorithm. Computers & Mathematics with Applications 33 (12), 1–13.
Savelsbergh, M.W., Sol, M., 1995. The general pickup and delivery problem. Transportation Science 29 (1), 17–29.
Savelsbergh, M., Sol, M., 1998. Drive: dynamic routing of independent vehicles. Operations Research 46 (4), 474–490.
Toth, P., Vigo, D., 1996. Fast local search algorithms for the handicapped persons transportation problem. In: Meta-Heuristics. Springer, pp. 677–690.
Toth, P., Vigo, D., 1997. Heuristic algorithms for the handicapped persons transportation problem. Transportation Science 31 (1), 60–71.
Xiang, Z., Chu, C., Chen, H., 2006. A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints. European Journal of Operational Research 174 (2), 1117–1139.
Zhang, Z., Liu, M., Lim, A., 2015. A memetic algorithm for the patient transportation problem. Omega 54 (0), 60–71.