



Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem

Sophie N. Parragh

Department of Business Administration, University of Vienna, Bruenner Strasse 72, 1210 Vienna, Austria

ARTICLE INFO

Article history:

Received 23 November 2009

Received in revised form 7 June 2010

Accepted 16 June 2010

Keywords:

Dial-a-ride

Heterogeneous passengers

Heterogeneous fleet

Branch-and-cut

Variable neighborhood search

ABSTRACT

Dial-a-ride problems deal with the transportation of people between pickup and delivery locations. Given the fact that people are subject to transportation, constraints related to quality of service are usually present, such as time windows and maximum user ride time limits. In many real world applications, different types of users exist. In the field of patient and disabled people transportation, up to four different transportation modes can be distinguished. In this article we consider staff seats, patient seats, stretchers and wheelchair places. Furthermore, most companies involved in the transportation of the disabled or ill dispose of different types of vehicles. We introduce both aspects into state-of-the-art formulations and branch-and-cut algorithms for the standard dial-a-ride problem. Also a recent metaheuristic method is adapted to this new problem. In addition, a further service quality related issue is analyzed: vehicle waiting time with passengers aboard. Instances with up to 40 requests are solved to optimality. High quality solutions are obtained with the heuristic method.

© 2010 Elsevier Ltd. Open access under [CC BY-NC-ND license](https://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Dial-a-ride problems deal with the transportation of people between pre-specified pickup and delivery locations. Usually, users specify time windows for either the pickup or the drop-off location, depending on the type of request. In the case of an outbound request (e.g., from home to a hospital), a time window is defined for the delivery location (destination). In the case of an inbound request (e.g., from the hospital back home), a time window for the pickup location (origin) is given. In addition, user ride time as well as maximum route duration limits have to be respected. The objective is to generate a transportation plan serving all patient transportation requests at minimum routing costs. In this article we introduce heterogeneous users and vehicles into the standard dial-a-ride problem (DARP), as defined, e.g., by [Cordeau \(2006\)](#). The resulting problem will be denoted as heterogeneous DARP (HDARP).

The introduction of heterogeneous users and vehicles is motivated by observations made at the Austrian Red Cross (ARC) in the field of patient transportation. The ARC distinguishes three patient types. A patient may demand to be transported seated, on a stretcher, or in a wheelchair. In addition, an accompanying person may be present. The ARC disposes of two different types of vehicles. Each type provides different capacities for four modes of transportation (staff seat, patient seat, stretcher, wheelchair place). In the following, these are referred to as resources. Resource 0 refers to staff seats, resource 1 to patient seats, resource 2 to stretchers, and resource 3 to wheelchair places. Each passenger can only be transported by a vehicle that provides the appropriate resource. Usually, accompanying persons use staff seats; seated patients take patient seats; a patient demanding a stretcher is transported on a stretcher; and a patient in a wheelchair occupies a wheelchair place. However, certain so-called upgrading conditions apply: an accompanying person may use the patient seat, in

E-mail address: sophie.parragh@univie.ac.at

case there is no vacant staff seat available, or, in case neither of the two is currently available, he/she can sit on the stretcher. According to Austrian law, a seated patient cannot use a staff seat, he/she may, however, sit on the stretcher, in case there is no additional patient seat available. Patients that demand a stretcher can only be transported on a stretcher; patients in wheelchairs can only be transported by vehicles providing space for a wheelchair.

Another aspect that has only rarely been considered in the literature is user waiting time while aboard a vehicle. In general, one can assume that some waiting time within the defined time window is acceptable for users of dial-a-ride systems. However, when already aboard a vehicle, waiting times should be avoided in order to keep user inconvenience at a reasonably low level. Psychologically, waiting for a vehicle is perceived in a different way than waiting time on board a vehicle (except for loading or unloading operations). The latter decreases the perceived service quality more than the former. Therefore, in the following, we will also integrate a term into the objective function that penalizes vehicle waiting time with passengers aboard.

Previous publications, considering heterogeneous versions of the DARP, mostly employ metaheuristic methods in order to solve the problem at hand. Toth and Vigo (1997), e.g., devise a parallel insertion heuristic and a tabu thresholding algorithm for a DARP considering two modes of transportation (seated and in a wheelchair) and several types of vehicles. Another heterogeneous version of the DARP is described in Melachrinoudis et al. (2007). Different types of vehicles in terms of capacity limits but only one mode of transportation are considered. The solution method developed is a tabu search algorithm. Heterogeneous vehicles in terms of different capacity limits for one mode of transportation in the context of the DARP are also considered in the work of Rekiek et al. (2006). The proposed problem is solved by a grouping genetic algorithm.

In a dynamic environment, Beaudry et al. (2009) adapt the tabu search heuristic developed by Cordeau and Laporte (2003) to solve a heterogeneous DARP that arises in large hospitals. It involves transportation requests demanding three different modes of transportation (seated, on a bed, or in a wheelchair) and several different types of vehicles. Hanne et al. (2009) develop a computer based planning system for a dynamic problem in a large German hospital, considering hospital specific constraints such as multi-dimensional capacities.

In terms of exact algorithms for DARPs, early publications such as Psaraftis (1980, 1983) and Desrosiers et al. (1986) focus on dynamic programming algorithms. More recently, Cordeau, 2006 proposed a branch-and-cut algorithm for the DARP. The algorithm is based on a 3-index mixed-integer programming formulation. New valid inequalities as well as previously developed ones for the pickup and delivery problem and the vehicle routing problem are employed. The largest instance solved to optimality comprises 36 requests. Two new branch-and-cut algorithms followed in Ropke et al. (2007). Instead of a 3-index formulation, more efficient 2-index problem formulations and additional valid inequalities are proposed. In this case, a 96-request instance is the largest instance solved to optimality. The algorithms of Cordeau (2006) and Ropke et al. (2007) are thus state-of-the-art in terms of exact algorithms for the DARP. The work presented in this article is based on their findings.

In a related paper, Parragh et al. (2009a) incorporate driver related constraints as well as heterogeneous users and vehicles into a 3-index and a set partitioning formulation of the DARP. These involve, e.g., the assignment of drivers and additional personnel to vehicles and the scheduling of lunch breaks. In these formulations, maximum ride times are only implicitly taken into account by constructing time windows at both the pickup and the drop-off location. This modification makes the application of column generation possible. The authors introduce a collaborative framework combining variable neighborhood search and column generation. They report results on instances with up to 54 requests.

For further information on the DARP, we refer the reader to Cordeau and Laporte (2007). For overviews of the more general pickup and delivery problem see Berbeglia et al. (2007), Parragh et al. (2008a,b).

The contributions of this article are threefold. First, two problem formulations, a 3-index and a 2-index formulation, for the HDARP are introduced (Section 2). A major new aspect in the 2-index formulation refers to how the differing capacity limits, depending on which vehicle travels along which arc, are integrated. Second, all valid inequalities proposed for the standard DARP are adapted to the new problem (Section 3) and employed in a 3-index and in a 2-index-based branch-and-cut algorithm (Section 4). Third, a state-of-the-art variable neighborhood search (VNS) heuristic (Mladenovic and Hansen, 1997), initially designed for the standard DARP (Parragh, 2009, Chapter 3), is adapted and applied to the HDARP (Section 4.3). Computational results for three sets of instances are reported and the impact of penalizing user waiting time while aboard a vehicle is analyzed. The article ends with a conclusion.

2. Problem definition

The HDARP is modeled on a complete directed graph $G = (V, A)$ where V is the set of all vertices and A the set of all arcs. A set K of m heterogeneous vehicles has to serve all n transportation requests. Each vehicle $k \in K$ is associated with a vector $C^{r,k}$ that gives the amount of resource r available on vehicle k . The ARC disposes of two basic vehicle types. Type 1 (T1) provides 1 staff seat, 6 patient seats, and 1 wheelchair place. Type 2 (T2) provides two staff seats, one patient seat, one stretcher, and one wheelchair place (see also Fig. 1). Each vehicle k has to start its route at the origin depot 0 and end at the destination depot $2n + 1$, respecting a route duration limit T^k . For each arc (i, j) and each vehicle $k \in K$ a non-negative travel cost c_{ij}^k and a non-negative travel time t_{ij}^k is considered. Each transportation request consists of a pickup and delivery vertex pair $\{i, n + i\}$. The set of pickup vertices is given by $P = \{1, \dots, n\}$, the set of delivery vertices by $D = \{n + 1, \dots, 2n\}$. At every pickup vertex one patient waits to be transported. This patient may demand one of three different modes of transportation. Passengers may have to be transported seated ($q_i^1 = 1$), on a stretcher ($q_i^2 = 1$), or in a wheelchair ($q_i^3 = 1$); each patient may be accompanied by a friend, relative or nurse ($q_i^0 = 1$). The demand at every delivery vertex is equal to $q_{n+i}^r = -q_i^r$ for all $r \in R = \{0, 1, 2, 3\}$. Every user either specifies a time window $[e_i, l_i]$ for the pickup (origin) or the drop-off (destination) location

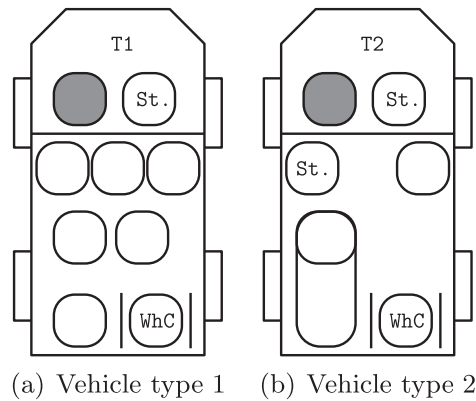


Fig. 1. Vehicle types at the ARC.

and beginning of service has to start within this time window. In case a vehicle arrives too early, it has to wait until service is possible. A maximum passenger ride time \bar{L} is also considered, in order to keep quality of service at a reasonably high level. At each vertex loading or unloading operations last for a given service time d_i . Thus, the set of all vertices is given by $V = P \cup D \cup \{0, 2n + 1\}$, and the set of all arcs by $A = \{(i, j) | i \in V \setminus \{2n + 1\}, j \in V \setminus \{0\}, i \neq j\}$.

As mentioned above, so-called upgrading conditions apply. Patients demanding to be transported seated may use a patient seat or the stretcher. Patients demanding a stretcher can only be transported on a stretcher. The same applies to wheelchair passengers. Accompanying persons, however, may use a staff seat, a patient seat, or the stretcher, if no other transportation mode is available. Table 1 gives an overview of the different upgrading options.

The model can be formulated with the following decision variables:

- $x_{ij}^k = \begin{cases} 1, & \text{if arc } (i, j) \text{ is traversed by vehicle } k \\ 0, & \text{else,} \end{cases}$
- $y_i^k = \begin{cases} 1, & \text{if vehicle } k \text{ arrives with passengers at vertex } i \\ 0, & \text{else,} \end{cases}$
- $B_i^k \dots$ beginning of service of vehicle k at vertex i ,
- $A_i^k \dots$ arrival time of vehicle k at vertex i ,
- $\widehat{W}_i^k \dots$ waiting time of vehicle k with passengers aboard at vertex i ,
- $Q_i^{r,k} \dots$ load of vehicle k of resource r when leaving vertex i ,
- $L_i^k \dots$ ride time of user i on vehicle k .

2.1. A 3-index formulation

The mathematical program, covering all of the above described real world conditions, based on the DARP formulation by Cordeau (2006), is as follows:

$$\min \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij}^k x_{ij}^k + \rho \sum_{k \in K} \sum_{i \in P \cup D} \widehat{W}_i^k, \quad (1)$$

s.t.:

$$\sum_{k \in K} \sum_{j \in P \cup D} x_{ij}^k = 1 \quad \forall i \in P, \quad (2)$$

$$\sum_{j \in V} x_{ij}^k - \sum_{j \in V} x_{n+i,j}^k = 0 \quad \forall i \in P, \quad k \in K, \quad (3)$$

Table 1

Transportation mode upgrading options.

Passenger type	Staff seat	Patient seat	Stretcher	Wheelchair place
Accompanying person	x	x	x	
Seated patient		x	x	
Patient on stretcher			x	
Patient in wheelchair				x

$$\sum_{j \in V} x_{0j}^k = 1 \quad \forall k \in K, \quad (4)$$

$$\sum_{i \in V} x_{i,2n+1}^k = 1 \quad \forall k \in K, \quad (5)$$

$$\sum_{i \in V} x_{ij}^k - \sum_{i \in V} x_{ji}^k = 0 \quad \forall j \in P \cup D, \quad k \in K, \quad (6)$$

$$x_{ij}^k = 1 \Rightarrow Q_j^{r,k} \geq Q_i^{r,k} + q_j^r \quad \forall i, j \in V, \quad k \in K, \quad r \in R, \quad (7)$$

$$\sum_{r' \in R} Q_i^{r',k} \leq \sum_{r' \in R} C^{r',k} \quad \forall i \in V, \quad k \in K, \quad r \in R \setminus \{3\}, \quad (8)$$

$$Q_i^{3,k} \leq C^{3,k} \quad \forall i \in V, \quad k \in K, \quad (9)$$

$$Q_i^{r,k} \geq 0 \quad \forall i \in V, \quad k \in K, \quad r \in R, \quad (10)$$

$$My_i^k \geq \sum_{r \in R} (Q_i^{r,k} - q_i^r) \quad \forall i \in P \cup D, \quad k \in K, \quad (11)$$

$$x_{ij}^k = 1 \Rightarrow A_j^k = B_i^k + d_i + t_{ij}^k \quad \forall i, j \in V, \quad k \in K, \quad (12)$$

$$B_i^k \geq A_i^k \quad \forall i \in V, \quad k \in K, \quad (13)$$

$$y_i^k = 1 \Rightarrow \widehat{W}_i^k \geq B_i^k - A_i^k \quad \forall i \in P \cup D, \quad k \in K, \quad (14)$$

$$L_i^k = B_{n+i}^k - (B_i^k + d_i) \quad \forall i \in P, \quad k \in K, \quad (15)$$

$$B_{2n+1}^k - B_0^k \leq T^k \quad \forall k \in K, \quad (16)$$

$$e_i \leq B_i^k \leq l_i \quad \forall i \in V, \quad k \in K, \quad (17)$$

$$t_{i,n+i} \leq L_i^k \leq \bar{L} \quad \forall i \in P, \quad k \in K, \quad (18)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j \in V, \quad k \in K, \quad (19)$$

$$y_i^k \in \{0, 1\} \quad \forall i \in P \cup D, \quad k \in K, \quad (20)$$

$$\widehat{W}_i^k \geq 0 \quad \forall i \in P \cup D, \quad k \in K. \quad (21)$$

The objective function (1) minimizes total routing costs and penalizes waiting time when passengers are aboard the vehicle. The penalty term ρ will be defined in such a way that waiting time while on board a vehicle is avoided. Constraints (2) and (3) guarantee that each request is served exactly once and that each origin-destination pair is visited by the same vehicle. Equalities (4)–(6) ensure that each vehicle starts at and returns to the depot at the end of its route.

Consistency with respect to resource and load variables is guaranteed by constraints (7)–(9). Upgrading constraints for resources 0, 1, and 2 are given in (8). They guarantee that capacity restrictions regarding these resources are not violated and that each patient demanding resource 0, 1 or 2 can only be loaded if there is either enough capacity of the resource demanded or another one with a higher number (0 = staff seat, 1 = patient seat, 2 = stretcher). Constraints (9) guarantee that patients demanding resource 3 (wheelchair place) can only be transported if there is enough capacity of resource 3.

Equalities (12) define the arrival times for each vertex. Constraints (13) guarantee that beginning of service only starts after having arrived at vertex i . Waiting times are set in (14). Waiting is only penalized if there is someone aboard the vehicle when arriving at vertex i . This is ensured by the way y_i^k are set in (11). Note that constraints (12) and (13) also take care of subtour elimination given that $(t_{ij} + d_i) > 0$ for all $i, j \in V, i \neq j$.

Equalities (15) define the ride time of each user. Total route duration is limited by (16), time window and maximum ride time compliance is ensured by (17) and (18).

Note that the capacity restrictions given by (8) and (9) are tailored to the problem situation of the ARC, they can, however, be rewritten in a more generic way as follows:

$$\sum_{r' \in R_r} Q_i^{r',k} \leq \sum_{r' \in R_r} C^{r',k} \quad \forall i \in V, \quad k \in K, \quad r \in R,$$

where the set R_r simply contains the different resources that can be used for the transportation of passengers demanding resource r . In our case, these would be $R_0 = \{0, 1, 2\}$, $R_1 = \{1, 2\}$, $R_2 = \{2\}$, and $R_3 = \{3\}$. Using this generic formulation, all different types of up-or down-grading could easily be accommodated with any number of resources.

2.1.1. Non-linear constraints

The above problem formulation contains a number of non-linear constraints, (7), (12) and (14). These can be reformulated in a linear way using so-called “big M” terms. All transformations described below are based on Cordeau (2006). The first set of non-linear constraints are the load propagation inequalities given in (7). Using “big M” terms tailored to each constraint, they can be rewritten as follows:

$$Q_j^{r,k} \geq (Q_i^{r,k} + q_j^r) - \mathcal{M}_{ij}^{r,k} (1 - x_{ij}^k) \quad \forall i, j \in V, \quad k \in K, \quad r \in R, \quad (22)$$

where $\mathcal{W}_{ij}^{r,k} \geq \max \{ \widehat{C}_{ij}^{r,k}, \widehat{C}_{ij}^{r,k} + q_j^r \}$ and $\widehat{C}_{ij}^{r,k} = C_{ij}^{r,k} + \sum_{r'=r+1}^2 C_{ij}^{r',k}$. The “big M” terms employed and denoted by $\mathcal{W}_{ij}^{r,k}$ represent the largest value the term $(Q_i^{r,k} + q_j^r)$ may take. Furthermore, as shown by Desrochers and Laporte (1991) they can then be lifted into

$$Q_j^{r,k} \geq (Q_i^{r,k} + q_j^r) - \mathcal{W}_{ij}^{r,k}(1 - x_{ij}^k) + (\mathcal{W}_{ij}^{r,k} - q_i^r - q_j^r)x_{ji}^k, \quad (23)$$

for the DARP by taking the reverse arc into account. Time consistency constraints given in (12) are transformed as follows:

$$A_j^k \leq B_i^k + d_i + t_{ij}^k + l_{2n+1}(1 - x_{ij}^k) \quad \forall i, j \in V, \quad k \in K, \quad (24)$$

$$A_j^k \geq B_i^k + d_i + t_{ij}^k - (l_i + d_i + t_{ij}^k)(1 - x_{ij}^k) \quad \forall i, j \in V, \quad k \in K. \quad (25)$$

Here, the original equality constraints are replaced by two sets of constraints, providing an upper and a lower bound on A_i . In the case where arc (i, j) is not used by vehicle k , these bounds should not be binding. Therefore, if arc (i, k) is not used by vehicle k , the upper bound is increased by l_{2n+1} (the later time window at the end depot) and the lower bound is decreased by $(l_i + d_i + t_{ij}^k) \geq (B_i^k + d_i + t_{ij}^k)$. Following a similar idea, inequalities (14), responsible for defining vehicle waiting time at each vertex i , are replaced by the following linear constraints,

$$\widehat{W}_i^k \geq (B_i^k - A_i^k) - l_i(1 - y_i^k) \quad \forall i \in P \cup D, k \in K. \quad (26)$$

2.1.2. Variable aggregation

The complexity of a problem formulation can be substantially decreased if all those variables that are homogeneous across the different vehicles are aggregated. In our case this is possible for all time related variables as these do not differ from one vehicle to the other. This implies that each vehicle drives at the same speed which is a reasonable assumption especially in rural areas. Therefore, t_{ij}^k can be replaced by t_{ij} and aggregate time variables A_i, B_i, L_i , and \widehat{W}_i can be used at all vertices except the two depots 0 and $2n + 1$. Furthermore, the binary variables y_i^k indicating whether a vehicle arrives full or empty at a vertex i , can be substituted by y_i . Here, the reasoning is that every vertex (except the depots) can only be visited exactly once. Therefore, some vehicle will at most once arrive at each vertex $i \in P \cup D$ either empty or with passengers aboard. All constraints on these variables have to be reformulated as follows. Equalities (12), defining the arrival time at each vertex, are now split into three separate sets of equalities:

$$x_{0j}^k = 1 \Rightarrow A_j = B_0^k + d_0 + t_{0j} \quad \forall j \in V, \quad k \in K, \quad (27)$$

$$\sum_{k \in K} x_{ij}^k = 1 \Rightarrow A_j = B_i + d_i + t_{ij} \quad \forall i \in P \cup D, \quad j \in P \cup D, \quad (28)$$

$$x_{i,2n+1}^k = 1 \Rightarrow A_{2n+1}^k = B_i + d_i + t_{i,2n+1} \quad \forall i \in V, \quad k \in K. \quad (29)$$

Inequalities (13), ensuring that the beginning of service variables are correctly set, are also transformed into three separate sets of constraints:

$$B_0^k \geq A_0^k \quad \forall k \in K, \quad (30)$$

$$B_i \geq A_i \quad \forall i \in P \cup D, \quad (31)$$

$$B_{2n+1}^k \geq A_{2n+1}^k \quad \forall k \in K. \quad (32)$$

Constraints (14) and (11), responsible for defining a correct lower bound on waiting time with passengers aboard, are replaced by

$$y_i = 1 \Rightarrow \widehat{W}_i \geq B_i - A_i \quad \forall i \in P \cup D, \quad (33)$$

$$My_i \geq \sum_{k \in K} \sum_{r \in R} (Q_i^{r,k} - q_i^r \sum_{j \in V} x_{ij}^k) \quad \forall i \in P \cup D, \quad (34)$$

and time window and ride time constraints given in (15), (17) and (18) are replaced by

$$L_i = B_{n+i} - (B_i + d_i) \quad \forall i \in P, \quad (35)$$

$$e_0 \leq B_0^k \leq l_0 \quad \forall k \in K, \quad (36)$$

$$e_i \leq B_i \leq l_i \quad \forall i \in P \cup D, \quad (37)$$

$$e_{2n+1} \leq B_{2n+1}^k \leq l_{2n+1} \quad \forall k \in K, \quad (38)$$

$$t_{i,n+i} \leq L_i \leq \bar{L} \quad \forall i \in P. \quad (39)$$

Let us further assume that the per vehicle arc costs are homogeneous across the different vehicle types, i.e. c_{ij}^k can be replaced by c_{ij} . Eventually, the objective function can be rewritten as follows:

$$\min \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij}^k + \rho \sum_{i \in P \cup D} \widehat{W}_i. \quad (40)$$

2.2. A 2-index formulation

When comparing the 3-index-based branch-and-cut algorithm for the DARP of Cordeau (2006) to the 2-index-based branch-and-cut algorithms proposed in Ropke et al. (2007), the 2-index versions obviously outperform the earlier 3-index-based method. Given the additional complexity of our problem, it can be assumed that a branch-and-cut algorithm based on a 2-index formulation will also perform better than a branch-and-cut algorithm based on a 3-index program. We note that, in order to reformulate the HDARP as a 2-index program, we have to adhere to the basic assumption that per vehicle arc costs c_{ij} and per vehicle arc travel times t_{ij} are homogeneous across the different vehicle types considered. In the transportation of the ill or disabled this assumption is reasonable; usually vehicles do not differ much in their dimensions or their weight but in terms of their on board equipment, expressed in the different transportation mode options. However, the reformulation of the HDARP as a 2-index program is still not as straightforward as one may assume, given the heterogeneous vehicle fleet and the resulting differing capacity limits, depending on which vehicle travels along which arc. How this issue can be resolved is described in the following.

In order to account for the heterogeneous nature of the vehicle fleet, artificial origin and destination depots for each vehicle are introduced. A similar approach is followed, e.g., in Baldacci et al. (2009) in the context of the heterogeneous fleet size and mix vehicle routing problem. Let $D_o = \{2n+1, \dots, 2n+m\}$ denote the set of vehicle origin depots and $D_d = \{2n+m+1, \dots, 2n+2m\}$ the set of vehicle destination depots, the vertex set V is redefined as $V = P \cup D \cup D_o \cup D_d$ and the arc set as $A = \{(i,j): i \in V \setminus D_d, j \in V \setminus D_o, i \neq j\}$. The demand/supply at the artificial depots is set to zero, $q_{2n+k}^r = q_{2n+m+k}^r = 0$ for all $k \in K, r \in R$; and the total maximum vehicle capacity shall be defined as $C = \sum_{r \in R} [\max_{k \in K} (C^{r,k})]$.

To impose request wise precedence and pairing constraints, let \mathcal{S} denote the set of all vertex subsets $S \subseteq V$, such that for all $k \in K$ the respective origin depot $2n+k \in S$ and the destination depot $2n+m+k \notin S$, and there is at least one request i for which $i \notin S$ and $n+i \in S$. For the correct pairing of the artificial depots, define \mathcal{U} as the set of all vertex subsets $U \subseteq V$, such that for exactly one $k \in K$ the origin depot $2n+k \in U$ and the destination depot $2n+m+k \notin U$, while for all other $l \in K \setminus \{k\}$ origin depots $2n+l \notin U$ and destination depots $2n+m+l \in U$.

Furthermore, to incorporate the corresponding vehicle capacity constraints, let \mathcal{H} denote the set of infeasible paths with respect to load violations $H = \{j_1, \dots, j_h\}$, such that $j_1 \in D_o$ and let $A(H)$ denote the arc set of H . Omitting superscript k from all decision variables, defining Q_i as the aggregated load when leaving vertex i , and $q_i = \sum_{r \in R} (q_i^r)$ as the aggregated demand/supply at vertex i , the HDARP can be formulated as the following 2-index program (inspired by Ropke et al., 2007):

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} + \rho \sum_{i \in P \cup D} \widehat{W}_i, \quad (41)$$

s.t.:

$$\sum_{i \in V \setminus D_d} x_{ij} = 1 \quad \forall j \in P \cup D \cup D_d, \quad (42)$$

$$\sum_{j \in V \setminus D_o} x_{ij} = 1 \quad \forall i \in P \cup D \cup D_o, \quad (43)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - (m+1) \quad \forall S \in \mathcal{S}, \quad (44)$$

$$\sum_{i \in U} \sum_{j \notin U} x_{ij} \geq 1 \quad \forall U \in \mathcal{U}, \quad (45)$$

$$x_{ij} = 1 \Rightarrow Q_j \geq Q_i + q_j \quad \forall i, j \in V, \quad (46)$$

$$\max\{0, q_i\} \leq Q_i \leq \min\{C, C + q_i\} \quad \forall i \in V, \quad (47)$$

$$My_i \geq Q_i - q_i \quad \forall i \in P \cup D, \quad (48)$$

$$\sum_{i=1}^{h-1} \sum_{l=i+1}^h x_{i,j_l} \leq |A(H)| - 1 \quad \forall H \in \mathcal{H}, \quad (49)$$

$$x_{ij} = 1 \Rightarrow A_j = B_i + d_i + t_{ij} \quad \forall i, j \in V, \quad (50)$$

$$B_i \geq A_i \quad \forall i \in V, \quad (51)$$

$$y_i = 1 \Rightarrow \widehat{W}_i \geq B_i - A_i \quad \forall i \in P \cup D, \quad (52)$$

$$L_i = B_{n+i} - (B_i + d_i) \quad \forall i \in P, \quad (53)$$

$$B_{2n+m+k} - B_{2n+k} \leq T^k \quad \forall k \in K, \quad (54)$$

$$e_i \leq B_i \leq l_i \quad \forall i \in V, \quad (55)$$

$$t_{i,n+i} \leq L_i \leq \bar{L} \quad \forall i \in P, \quad (56)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V, \quad (57)$$

$$y_i \in \{0, 1\} \quad \forall i \in P \cup D, \quad (58)$$

$$\widehat{W}_i \geq 0 \quad \forall i \in P \cup D. \quad (59)$$

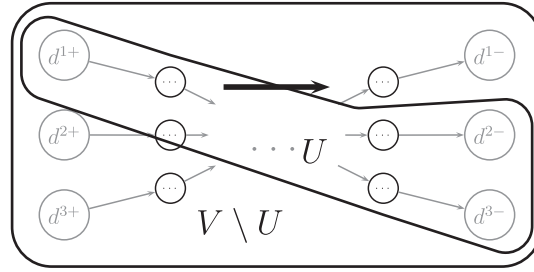


Fig. 2. Pairing of artificial origin and destination depots.

The objective function (41) minimizes total routing costs and penalizes waiting time with passengers aboard a vehicle. Constraints (42) and (43) guarantee that every vertex $i \in P \cup D$ is entered and left and that exactly one arc leaves (enters) every origin (destination) depot, i.e. $i \in D_o$ ($i \in D_d$). Precedence and pairing restrictions are guaranteed by (44) and (45). As shown by Ropke et al. (2007) inequalities (44), initially designed for the single vehicle case, also apply in a multi-vehicle context. Since one depot per vehicle is considered in our mathematical model, the original right hand side $|S| - 2$ has to be replaced by $|S| - (m + 1)$. Suppose the set $S \in \mathcal{S}$ is $S = \{2n + 1, \dots, 2n + m, n + i\}$. From each start depot one arc has to leave S (i.e. m arcs). One of these arcs will be part of a path leading to $i \notin S$. If pairing and precedence constraints are respected, this path has to contain one arc leading back to $n + i \in S$. Finally, it has to arrive at one of the end depots, demanding another arc to be traversed leaving S and connecting to some end depot $2n + m + k \notin S$. Thus, $m + 1$ arcs have to leave S , reducing the number of arcs that can be used within S to $|S| - (m + 1)$. Constraints (45) are depicted in Fig. 2. Here, d^{1+}, d^{2+} , and d^{3+} denote the origin depots of vehicles 1, 2, and 3, respectively, while d^{1-}, d^{2-} , and d^{3-} denote the corresponding destination depots. At least one arc (represented as an arrow) has to leave set U (here U consists of d^{1+}, d^{2+} , and d^{3+} and some other vertices) and enter set $V \setminus U$, in order to ensure the correct pairing of the depots (here d^{1+} and d^{1-}). Aggregate loading variables are introduced and set by constraints (46). These are needed for the definition of y_i , defined in (48). The actual loading restrictions are taken care of by inequalities (49); on every path $H \in \mathcal{H}$, each starting at one of the depots, a loading restriction is violated. Constraints of this type are known as tournament inequalities (Ascheuer et al., 2000). Arrival times are set in (50), beginning of service and waiting times in (51) and (52). Equalities (53) determine each user's ride time. Compliance with maximum route duration restrictions, time windows, and user ride time limits is guaranteed by (54)–(56).

2.2.1. Non-linear constraints

All non-linear inequalities are reformulated as shown for the 3-index model. Thus, load propagation constraints (46) are substituted by

$$Q_j \geq (Q_i + q_j) - \mathcal{W}_{ij}(1 - x_{ij}), \quad (60)$$

with $\mathcal{W}_{ij} \geq \max\{C, C + q_j\}$ and lifted into

$$Q_j \geq (Q_i + q_j) - \mathcal{W}_{ij}(1 - x_{ij}) + (\mathcal{W}_{ij} - q_i - q_j)x_{ji}, \quad (61)$$

by taking the reverse arc into account. Finally, arrival time inequalities (50) are replaced by,

$$A_j \leq (B_i + d_i + t_{ij}) + \max_{k \in K} \{l_{2n+m+k}\}(1 - x_{ij}), \quad (62)$$

$$A_j \geq (B_i + d_i + t_{ij}) - (l_i + d_i + t_{ij})(1 - x_{ij}), \quad (63)$$

and vehicle waiting time inequalities (52) by

$$W_i \geq (B_i - A_i) - l_i(1 - y_i). \quad (64)$$

3. Valid inequalities

Cordeau (2006) and Ropke et al. (2007) introduced new and adapted a number of existing valid inequalities for the DARP. In the following section we will review and adapt all these families of inequalities to the HDARP. Most valid inequalities can be used to strengthen both formulations, some are only valid for either of the two. If not stated otherwise, $x_{ij} = \sum_{k \in K} (x_{ij}^k)$ in the 3-index formulation. Furthermore, for ease of exposition, let $\kappa(S) = \sum_{i,j \in S} (x_{ij})$ denote the number of arcs traversed in a set of vertices S .

3.1. Strengthened bounds on time and load variables

In both the 3-index and the 2-index formulation, bounds on time variables, also denoted as time windows, can be strengthened as follows (Cordeau, 2006; Desrochers and Laporte, 1991):

$$B_i \geq e_i + \sum_{j \in V \setminus \{i\}} \max\{0, e_j - e_i + d_j + t_{ji}\} x_{ji}, \quad (65)$$

$$B_i \leq l_i - \sum_{j \in V \setminus \{i\}} \max\{0, l_i - l_j + d_i + t_{ij}\} x_{ij}. \quad (66)$$

In the case of the 3-index formulation bounds on load variables can also be strengthened (based on Cordeau, 2006). In order to do so, let $\text{orig}(i)$ be the set that contains the origin of i , if i is a destination, and otherwise the empty set. Thus, lower bounds can be strengthened as follows:

$$Q_i^{r,k} \geq \max\{0, q_i^r\} + \sum_{j \in V \setminus \{\text{orig}(i)\}} \max\{0, q_j^r\} x_{ji}^k. \quad (67)$$

The intuition is that if arc (j, i) is used and $q_j^r > 0$, given that j is not the origin of i , the load at i has to be greater or equal than q_j^r (plus q_i^r , if i is an origin).

In case of upper bounds, we have to distinguish between origins and destinations. If i is an origin, i.e. $i \in P = \{1, \dots, n\}$,

$$Q_i^{r,k} + \sum_{r'=r+1}^2 Q_i^{r',k} \leq \widehat{C}^{r,k} - \left(\widehat{C}^{r,k} - \max_{j \in V \setminus \{i\}} \{\hat{q}_j^r\} - \hat{q}_i^r \right) x_{0i}^k - \sum_{j \in V \setminus \{i\}} \max\{0, \hat{q}_j^r\} x_{ij}^k, \quad (68)$$

where $\widehat{C}^{r,k} = C^{r,k} + \sum_{r'=r+1}^2 C^{r',k}$ and $\hat{q}_i^r = q_i^r + \sum_{r'=r+1}^2 q_i^{r'}$, applies. The intuition behind this strengthening is that if i is visited directly after the depot 0, the load can be at most \hat{q}_i^r . In addition, if the vertex visited directly after i is an origin and $\hat{q}_j^r > 0$, then the load at i can be at most $\widehat{C}^{r,k} - \hat{q}_j^r$. If i is a destination, i.e. $i \in D = \{n+1, \dots, 2n\}$,

$$Q_i^{r,k} + \sum_{r'=r+1}^2 Q_i^{r',k} \leq \min\{\widehat{C}^{r,k}, \widehat{C}^{r,k} + \hat{q}_i^r\} - \left(\widehat{C}^{r,k} - \max_{j \in V \setminus \{i\}} \{\hat{q}_j^r\} - \hat{q}_i^r \right) x_{0i}^k, \quad (69)$$

is used. The last term used in (68) cannot be used to strengthen (69). Consider $\widehat{C}^{r,k} = 1$, if $\hat{q}_i^r = -1$ and $\hat{q}_j^r = 1$ then $(Q_i^{r,k} + \sum_{r'=r+1}^2 Q_i^{r',k})$ would have to be ≤ -1 which is clearly not valid, although visiting j after i is feasible with respect to capacity limits.

3.2. Subtour elimination constraints

Standard subtour elimination constraints are given by $x(S) \leq |S| - 1$ for $S \subset P \cup D$. They ensure that a given subset of vertices S cannot contain a cycle through all the vertices forming S . As shown by Cordeau (2006) these constraints can be lifted in several ways, exploiting the fact that an origin i must be visited before its destination $n+i$. Let $S = \{i_1, i_2, \dots, i_h\} \subseteq P \cup D$ and $\bar{S} = \{i \in P \cup D \mid i \notin S\}$. The predecessors of S are denoted as $\pi(S) = \{i \in P \mid n+i \in S\}$, its successors as $\sigma(S) = \{n+i \in D \mid i \in S\}$. The following four sets of inequalities are valid for the DARP (Cordeau, 2006) and are also applicable in the context of the HDARP:

$$x(S) + \sum_{i \in \bar{S} \cap \sigma(S)} \sum_{j \in S} x_{ij} + \sum_{i \in \bar{S} \cap \pi(S)} \sum_{j \in S \cap \sigma(S)} x_{ij} \leq |S| - 1, \quad (70)$$

$$x(S) + \sum_{i \in S} \sum_{j \in \bar{S} \cap \pi(S)} x_{ij} + \sum_{i \in S \cap \pi(S)} \sum_{j \in \bar{S} \cap \pi(S)} x_{ij} \leq |S| - 1, \quad (71)$$

$$\sum_{j=1}^{h-1} x_{i_j, i_{j+1}} + x_{i_h, i_1} + 2 \sum_{j=2}^{h-1} x_{i_j, i_1} + \sum_{j=3}^{h-1} \sum_{l=2}^{j-1} x_{i_j, i_l} + \sum_{n+i_p \in \bar{S} \cap \sigma(S)} x_{n+i_p, i_1} \leq h - 1, \quad (72)$$

$$\sum_{j=1}^{h-1} x_{i_j, i_{j+1}} + x_{i_h, i_1} + 2 \sum_{j=3}^h x_{i_1, i_j} + \sum_{j=4}^h \sum_{l=3}^{j-1} x_{i_j, i_l} + \sum_{i_p \in \bar{S} \cap \pi(S)} x_{i_1, i_p} \leq h - 1. \quad (73)$$

All four sets of inequalities can be used to strengthen both the 3-index and the 2-index formulation. The first two sets of inequalities were originally introduced for the precedence constrained asymmetric TSP by Balas et al. (1995). They are also referred to as successor and predecessor inequalities, respectively. The ordering of the vertices in S does not play a role here. In the second two sets of inequalities the ordering of the vertices is important. They are based on those of Grötschel and Padberg (1985) for the asymmetric TSP.

These four sets of inequalities are further illustrated in Fig. 3. Dotted and dashed arcs represent liftings. In Fig. 3a an example for successor inequality (70) is given. The set $S = \{i, j\}$ consists of two origin vertices. The successors of these two are their destinations $\sigma(S) = \{n+i, n+j\}$. Clearly, at most one ($|S| - 1 = 1$) of the four arcs shown in the figure can be used in a feasible solution. The same is true for the example of (71) in Fig. 3b. For inequalities (72), (73) the reasoning is a bit

different. Instead of sets, sequences of vertices are considered. Such that a sequence does not result in a cycle, at most $h - 1$ arcs can be used, given that the sequence consists of h arcs. If a reverse arc is used between two vertices, the vertices connected by this arc should no longer be connected to the rest of the sequence. This reduces the number of arcs that can be used in the sequence by two. For this reason a reverse arc (dashed in the figure) is counted twice. Furthermore, in case of inequality (72), illustrated in Fig. 3c, if a destination (successor) $\notin S$ of some origin $\in S$ is connected to the first vertex $\in S$, the first vertex cannot be connected to the origin of this destination anymore, reducing the number of vertices that can be used in the sequence by one. The same reasoning, considering predecessors instead of successors of the vertices $\in S$, leads to inequalities (73). An according example is given in Fig. 3d.

3.3. Generalized order constraints

Generalized order constraints form another set of inequalities that can be used to strengthen the above formulations. They were originally proposed by [Ruland and Rodin \(1997\)](#). In order to properly define them, let U_1, \dots, U_h be mutually disjoint subsets and let $i_1, \dots, i_h \in P$ be users such that either $0, 2n + 1 \notin U_l$ (3-index formulation) or $2n + k, 2n + m + k \notin U_l$ for all $k \in K$ (2-index formulation) and $i_l, n + i_{l+1} \in U_l$ for $l = 1, \dots, h$ ($i_h + 1 = i_1$). Then, the following inequalities are valid for the (H)DARP,

$$\sum_{l=1}^h x(U_l) \leq \sum_{l=1}^h |U_l| - h - 1. \quad (74)$$

For the (H)DARP these can be lifted in two ways, as shown by [Cordeau \(2006\)](#),

$$\sum_{l=1}^h x(U_l) + \sum_{l=2}^{h-1} x_{i_1, i_l} + \sum_{l=3}^h x_{i_l, n+i_l} \leq \sum_{l=1}^h |U_l| - h - 1, \quad (75)$$

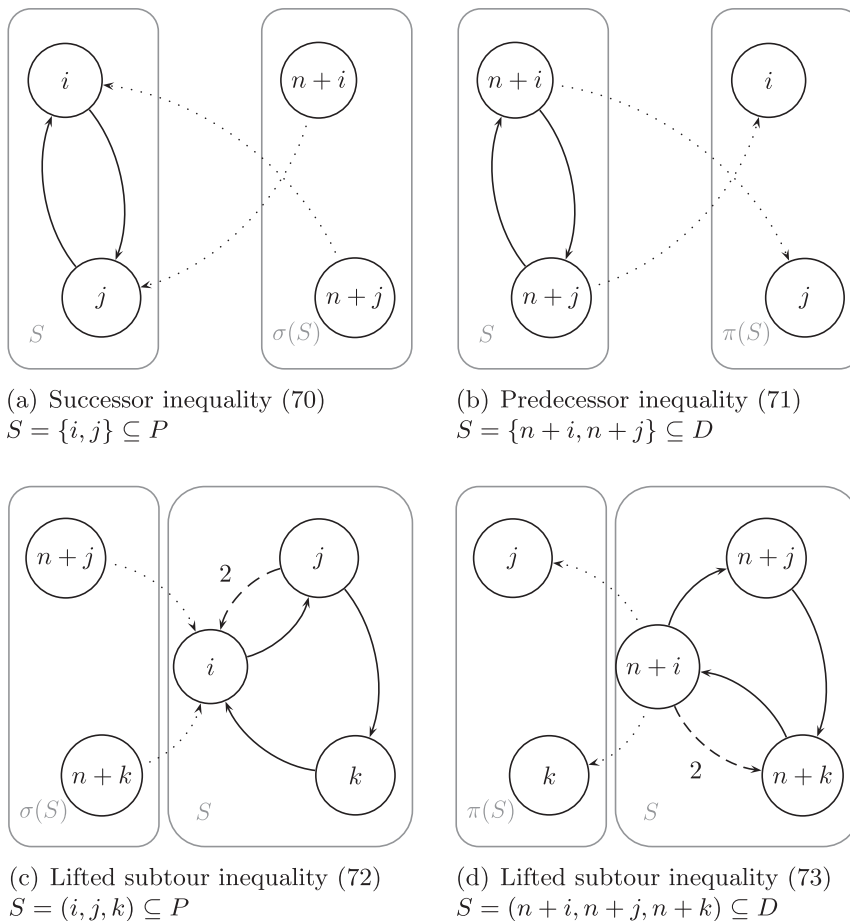


Fig. 3. Lifted subtour elimination constraints (adapted from [Cordeau, 2006](#)).

$$\sum_{l=1}^h x(U_l) + \sum_{l=2}^{h-2} x_{n+i_1, l} + \sum_{l=2}^h x_{n+i_1, n+i_l} \leq \sum_{l=1}^h |U_l| - h - 1. \quad (76)$$

3.4. Infeasible path constraints

Another popular way to further strengthen problem formulations in the field of vehicle routing, are constraints based on infeasible paths. We consider two families of valid inequalities that are based on the notion of infeasible paths. The first are referred to as strengthened infeasible path constraints, the second have been denoted as fork constraints.

In order to define strengthened infeasible path constraints, let \mathcal{F} denote the set of infeasible paths and for each set $F \in \mathcal{F}$, let $A(F)$ be the arc set of F and $V(F)$ the vertex set. Then the following inequalities are valid for the (H)DARP (Ropke et al., 2007),

$$\sum_{(ij) \in A(F)} x_{ij}^k \leq |A(F)| - 1. \quad (77)$$

If $F = (j_1, \dots, j_h)$ denotes an infeasible path, they can be strengthened into so-called tournament constraints (see Ascheuer et al., 2000; Ropke et al., 2007),

$$\sum_{i=1}^{h-1} \sum_{l=i+1}^h x_{j_i j_l}^k \leq |A(F)| - 1. \quad (78)$$

If F links a vertex pair $\{i, n+i\}$, and $F = (i, j_1, \dots, j_h, n+i)$ is infeasible due to time window or ride time constraints, (77) can be lifted in the following way (Cordeau, 2006),

$$x_{ij_1}^k + \sum_{l=1}^{h-1} x_{j_l j_{l+1}}^k + x_{j_h, n+i}^k \leq |A(F)| - 2. \quad (79)$$

If both path $F = (j_1, \dots, j_h)$ and the reverse path $F' = (j_h, \dots, j_1)$ are infeasible, the following inequality can be applied (Ropke et al., 2007),

$$\sum_{i=1}^{h-1} (x_{j_i j_{i+1}}^k + x_{j_{i+1} j_i}^k) \leq h - 1. \quad (80)$$

These inequalities are generated for each vehicle in turn in case of the 3-index formulation. In case of the 2-index formulation x_{ij}^k has to be replaced by x_{ij} and only time related infeasibilities are considered since these restrictions are homogeneous across all vehicles.

The second family of inequalities that is based on the notion of infeasible paths has been proposed by Ropke et al. (2007) in the context of the PDPTW and the DARP. In this case we only consider paths that are infeasible regarding time infeasibilities. They are eliminated by taking a whole bundle of such infeasible paths into account, sharing a number of common arcs. Let $F = (j_1, \dots, j_h)$ be a feasible path. By adding a vertex at the beginning and at the end of this path an infeasible path is generated. A bundle of infeasible paths is defined by (i, F, l) being infeasible for every $i \in S$ and $l \in T, S, T \subset V$, resulting in the so-called fork inequality,

$$\sum_{i \in S} x_{ij_1} + \sum_{i=1}^{h-1} x_{j_i j_{i+1}} + \sum_{l \in T} x_{j_h, l} \leq h. \quad (81)$$

These inequalities can be lifted in two ways. In the first, an additional set of vertices T_i is appended to each vertex $j_i \in \{j_1, \dots, j_{h-1}\}$ resulting in sequences (k, j_1, \dots, j_i, l) that are infeasible for every $k \in S, l \in T_i, 1 \leq i \leq h$. This so-called outfork inequality is given by

$$\sum_{i \in S} x_{ij_1} + \sum_{i=1}^{h-1} x_{j_i j_{i+1}} + \sum_{i=1}^h \sum_{l \in T_i} x_{j_i, l} \leq h. \quad (82)$$

In the second, a set of vertices S_i is added before each vertex $j_i \in \{j_2, \dots, j_h\}$, leading to sequences (k, j_i, \dots, j_h, l) that are infeasible for every $k \in S_i, l \in T, 1 \leq i \leq h$. This so-called infork inequality is given by

$$\sum_{k \in S_i} \sum_{i=1}^h x_{kj_i} + \sum_{i=1}^{h-1} x_{j_i j_{i+1}} + \sum_{l \in T} x_{j_h, l} \leq h. \quad (83)$$

3.5. Capacity inequalities

For the 3-index formulation additional valid inequalities that are based on capacity issues can be appended to the model. We distinguish two families of inequalities, the adapted rounded capacity inequalities and the strengthened capacity

inequalities. In order to define the former, let $\widehat{R}(S)$ denote the minimum number of vehicles necessary to visit all nodes in $S \subseteq P \cup D$, then the constraint $x(\Delta(S)) \geq 2\widehat{R}(S)$ is a valid inequality for the DARP (Cordeau, 2006), $\Delta(S) = \sum_k \sum_{i \in S} \sum_{j \in S} (x_{ij}^k) + \sum_k \sum_{i \in S} \sum_{j \notin S} (x_{ij}^k)$. A lower approximation of $\widehat{R}(S)$ is given by $\lceil q(S)/C \rceil$, where $q(S) = \sum_{i \in S} (q_i)$.

This so-called rounded capacity inequality can be adapted to the HDARP as follows. Let $\widehat{C}^{r,k}$ denote the cumulative capacity limit for resource r on vehicle k and be $q^r(S) = \sum_{i \in S} (q_i^r + \sum_{r'=r+1}^2 q_i^{r'})$, then the constraint

$$x(\Delta(S)) \geq 2 \left\lceil \frac{q^r(S)}{\max_{k \in K} (\widehat{C}^{r,k})} \right\rceil, \quad (84)$$

is a valid rounded capacity inequality for the HDARP.

Introduced in Ropke et al. (2007), the strengthened capacity cuts for the DARP are defined as follows. Let $S, T \subseteq P \cup D$ denote two disjoint subsets such that $q(S) > 0$. Furthermore, let $\widehat{U} = \pi(T) \setminus (S \cup T)$, the strengthened capacity inequality for the DARP is given by

$$x(S) + x(T) + x(S : T) \leq |S| + |T| - \left\lceil \frac{q(S) - q(\widehat{U})}{C} \right\rceil, \quad (85)$$

with $x(S : T) = \sum_{i \in S} \sum_{j \in T} (x_{ij})$. In case of the HDARP $q(\cdot)$ has to be replaced by $q^r(\cdot)$ as defined above, and instead of C , $\max_{k \in K} (\widehat{C}^{r,k})$ has to be used.

3.6. Reachability constraints

In case of the 2-index formulation another group of inequalities, introduced in Lysgaard (2006) and used in the context of the DARP by Ropke et al. (2007), can be used to strengthen the model. It is based on the notion of conflicting vertices. Vertices are said to be conflicting if they cannot be served by the same vehicle. For every vertex $i \in P \cup D$ let $A_i^- \subset A$ define the minimum arc set such that any feasible sequence from an origin depot $2n + k$ ($k \in K$) to vertex i can be constructed only using arcs $\in A_i^-$. Furthermore, let $A_i^+ \subset A$ denote the minimum arc set such that any feasible sequence from i to a destination depot $2n + m + k$ ($k \in K$) only traverses arcs $\in A_i^+$. Finally, let \widehat{T} be a set of conflicting vertices. $A_T^- = \bigcup_{i \in T} A_i^-$ defines the reaching arc set of \widehat{T} and $A_T^+ = \bigcup_{i \in T} A_i^+$ the reachable arc set of \widehat{T} . The following two inequalities can be defined,

$$x(\Delta^-(\widehat{S}) \cap A_T^-) \geq |\widehat{T}|, \quad (86)$$

$$x(\Delta^+(\widehat{S}) \cap A_T^+) \geq |\widehat{T}|, \quad (87)$$

with $\Delta^-(\widehat{S}) = \sum_{i \in \widehat{S}} \sum_{j \in \widehat{S}} (x_{ij})$ and $\Delta^+(\widehat{S}) = \sum_{i \in \widehat{S}} \sum_{j \notin \widehat{S}} (x_{ij})$.

4. Branch-and-cut algorithms

Based on each of the above introduced formulations, we have implemented a branch-and-cut algorithm. The algorithm based on the 3-index program will be denoted as *3indexBC* in the following, the algorithm based on the 2-index formulation as *2indexBC*. Branch-and-cut algorithms combine the branch-and-bound and the cutting-plane idea. Every mixed integer program (MIP) can be reformulated as a linear program (LP) by dropping all integrality constraints. In our case, these are (19) and (20) in the 3-index, and (57) and (58) in the 2-index formulation. The optimal solution to the LP relaxation will yield a lower bound for the original MIP.

Branch-and-cut algorithms depart from the solution of the LP relaxation, considering only a reasonable subset of the original constraints. Typically, all constraint families of exponential size are not included. In case of *2indexBC*, these are the pairing constraints (44) and (45), and the infeasible path constraints (49). Also all families of valid inequalities that are only used to strengthen the model are added in a cutting-plane fashion. Separation algorithms will check the current solution for violations of the omitted constraints and the above defined valid inequalities. In case of omitted constraints, the separation procedures have to correspond to exact methods such that, if one of the omitted constraints is violated, we can guarantee that it will be identified. In case of additional valid inequalities, which are not needed to ensure feasibility, heuristic separation procedures can be employed.

In the case where at least one violated constraint is detected by the separation procedures, it is added to the current LP in the form of a cut and the updated LP is solved again. This process is repeated until the separation procedures fail to detect additional violated constraints. In this case, the optimal solution to the LP relaxation of the original MIP has been found. If this solution is integer, the optimal solution to the original MIP has been identified. Otherwise, the problem is decomposed into two new problems. As in branch-and-bound, by branching on a variable that is associated with a fractional value in the current solution. The following branching rules are applied. In case of *3indexBC*, as in Cordeau (2006), additional artificial variables are introduced that handle the assignment of requests to vehicles. Branching priority is given to these variables. Both *3indexBC* and *2indexBC* branch on the variable furthest away from the nearest integer. Let us assume that this variable is $x_{1,3}$ (e.g., $x_{1,3} = 0.4$) in *2indexBC*. Branching refers to the generation of two child nodes. At one of the child nodes an LP is built with an additional constraint setting a lower bound on the chosen variable. This lower bound is equal to the fractional

value of the variable ceiled to the next integer, e.g. $x_{1,3} \geq 1$. At the second child node another LP is built with an additional constraint setting an upper bound on the chosen variable. This upper bound is equal to the fractional value of the chosen variable floored to the next integer, e.g. $x_{1,3} \leq 0$. Then, each of the two LPs is solved in the same way as the first LP relaxation; cuts are added until no more violated inequalities can be detected. In case the optimal solution is not integer, the child node serves as a new parent node for two new child nodes in the tree. The optimal solution to the original MIP is the best of the first two thus recursively solved problems. For further details and additional references on the branch-and-cut method in general we refer the interested reader to [Naddef and Rinaldi \(2002\)](#).

To accelerate the solution process several preprocessing steps can be performed prior to starting the optimization procedure. These steps refer to graph pruning and time window tightening techniques. Furthermore, some “easy” cuts can be generated in advance to strengthen the formulation. In our case these are all strengthened bounds on time and load variables; and, in order to break symmetry, in case of the 3-index formulation, variable fixing techniques can be employed; e.g. in the homogeneous fleet case, those requests that cannot be served by the same vehicle are fixed to different vehicles. Finally, before applying the branch-and-cut algorithms upper bounds are computed by means of an adapted version of a state-of-the-art heuristic method for the standard DARP. These bounds serve to prune the branch-and-bound tree; whenever the solution value at some node exceeds that of the upper bound, this node and all its child nodes can be excluded from the search. Our branch-and-cut implementations are widely based on the work of [Cordeau \(2006\)](#) and [Ropke et al. \(2007\)](#). The employed preprocessing steps, the different separation procedures, and the computation of upper bounds are briefly described in the following.

4.1. Initialization

In a first step, preprocessing procedures, using graph pruning and time window tightening techniques as described by [Cordeau \(2006\)](#), adapted to the HDARP, are applied. Then an initial pool of inequalities is generated. All inequalities part of this pool are enumerated exhaustively at every node of the branch-and-bound tree. It contains all cuts part of the initial pool of inequalities described in [Cordeau \(2006\)](#): subtour elimination constraints for $|S| = 2$ (see Section 3.2), the following four particular cases of precedence constraints,

$$\begin{aligned} x_{0i} + x_{i,n+j} + x_{n+j,i} &\leq 1, \\ x_{i,n+j} + x_{n+j,i} + x_{n+j,2n+1} &\leq 1, \\ x_{0i} + x_{i,n+i} + x_{i,n+j} + x_{n+j,i} + x_{n+i,n+j} + x_{n+j,n+i} &\leq 2, \\ x_{ij} + x_{ji} + x_{i,n+j} + x_{n+j,i} + x_{j,n+j} + x_{n+j,2n+1} &\leq 2, \end{aligned}$$

generalized order constraints for $h = 2$ and $|U_1| = |U_2| = 2$ (see Section 3.3), and infeasible path constraints of the form $x_{ij} + x_{j,n+j} + x_{n+j,n+i} \leq 1$ if this path violates the ride time constraint of request i . Furthermore, four inequalities for each pair of incompatible users $ij \in P$ and every node $l \in P \cup D$ are added to the pool:

$$\begin{aligned} x_{il} + x_{li} + x_{lj} + x_{jl} &\leq 1, \\ x_{il} + x_{li} + x_{l,n+j} + x_{n+j,l} &\leq 1, \\ x_{n+i,l} + x_{l,n+i} + x_{lj} + x_{jl} &\leq 1, \\ x_{n+i,l} + x_{l,n+i} + x_{l,n+j} + x_{n+j,l} &\leq 1. \end{aligned}$$

Finally, in case of *3indexBC*, variable fixing methods are used. If the vehicle fleet of an instance is homogeneous, the same variable fixing procedures as in [Cordeau \(2006\)](#) are applied. If the fleet is heterogeneous, variables can be fixed as follows. If only one vehicle with resource 2 or 3 (stretcher or wheelchair) is available, all users demanding these transportation modes can be fixed to the corresponding vehicle. If some vehicle does not provide these resources all users demanding them can be forbidden on this vehicle. Furthermore, if two users are identified as incompatible for one vehicle, a constraint guaranteeing that only one of the two can use this vehicle is appended to the model.

4.2. Separation heuristics

In both algorithms, violated subtour elimination (see Section 3.2) and generalized order constraints (see Section 3.3) are separated by means of several (meta)heuristics as described in [Cordeau \(2006\)](#). For the separation of strengthened infeasible path inequalities (see Section 3.4), an enumerative procedure as described in [Ropke et al. \(2007\)](#) is used. Because of the heterogeneous fleet requirements, these constraints are checked and generated for each vehicle in turn in case of the 3-index formulation. The same applies to violated fork constraints which are also determined by means of enumeration procedures ([Ropke et al., 2007](#)). In case of the 2-index formulation, only time related infeasibilities are considered since these are homogeneous across all vehicles.

In *3indexBC*, as in [Cordeau \(2006\)](#), for the separation of rounded capacity inequalities (see Section 3.5), a tabu search algorithm is employed. To detect violated strengthened capacity cuts, the heuristics as described by [Ropke et al. \(2007\)](#) are used, i.e. a randomized construction heuristic and another tabu search algorithm. Note that, in case of *3indexBC*, the separation

procedures briefly described above are only evoked if all assignment variables are integer, i.e. no vertex is partially assigned to more than one vehicle.

For *2indexBC* additional separation procedures need to be devised. Precedence and pairing constraints, given in (44) are separated similar to Ropke et al. (2007) (the implementation of Goldberg's algorithm for solving the individual max-flow problems is used). The pairing constraints in the 2-index model for the introduced artificial depots are separated by a similar procedure. Every origin depot together with all other destination depots are in turn used as the super source node and its corresponding destination depot together with all other origin depots as the super sink node. If the max flow on the network (arc capacities are defined by the current values of the x_{ij} variables) is smaller than one, the according depot pairing constraint, given in (45), is generated.

For the detection of infeasible paths regarding heterogeneous vehicles and resources, violating a capacity restriction in the 2-index formulation, another enumerative procedure is employed: every path starting at one of the start depots is extended on all arcs with some flow ($x_{ij} > 0$) and checked for capacity violations. If the capacity is exceeded at some vertex i , the corresponding inequality (49) is generated. Path extension ends as soon as one of the destination depots has been reached.

As in Ropke et al. (2007), to separate reachability cuts in *2indexBC*, in a first step, for each vertex $i \in P \cup D$ the arc sets A_i^+ and A_i^- are computed. Based on these arc sets conflicting vertex sets are identified (if there is no path that contains a certain vertex j leading from one of the start depots to i , or from i to one of the destination depots, i and j are conflicting vertices). After having determined all conflicting pairs, conflicting sets of larger cardinality are generated on the basis of conflicting pairs. Obviously, at most sets with a cardinality of m are considered. If sets of higher cardinality existed the respective problem instance would be infeasible. At each fractional solution encountered at some node of the branch and bound tree, each conflicting set \hat{T} is considered in turn. In order to identify violated inequalities (86), a maximum flow problem between D_o and \hat{T} is solved using only arcs $\in A_{\hat{T}}^-$. In order to do so, an artificial source node that is connected to each vertex $\in D_o$ with arcs of infinite capacity, and an artificial sink node connecting all vertices $\in \hat{T}$ with arcs of infinite capacity to this sink have to be introduced. If the minimum cut's capacity obtained is smaller than $|\hat{T}|$, the according reachability cut is generated. Similarly, to identify violated inequalities (87), a max-flow problem between \hat{T} and D_d is solved using only arcs $\in A_{\hat{T}}^+$.

4.3. Heuristic upper bounds

In order to accelerate the optimization process initial upper bounds are calculated by means of an adapted version of a state-of-the-art VNS (Mladenovic and Hansen, 1997) designed for the standard DARP in Parragh (2009, Chapter 3). The VNS developed in Parragh (2009, Chapter 3) departs from a possibly infeasible *initial solution* which is constructed taking time window and spatial closeness considerations into account. The thus constructed initial solution constitutes the first incumbent solution s . Then, in every iteration a random solution s' is generated in the current neighborhood of s (*shaking*). In the shaking phase a neighborhood operator belonging to one of four different neighborhood classes is applied. The neighborhood classes are denoted as swap (S), move (M), chain (C), and zero split (Z). Throughout the search they are traversed in alternating order and increasing size. Given that s is a *promising solution*, s undergoes *local search*-based improvement (see Parragh, 2009; Parragh et al., 2010). If s' meets the acceptance criteria, it replaces s and the search continues with the first (smallest) neighborhood. If s' does not meet the acceptance criteria, s is not replaced and the next (larger) neighborhood is used in the subsequent iteration (*move or not*). The acceptance criteria employed refer to simulated annealing type (Kirkpatrick et al., 1983) rules. This means that also deteriorating solutions may become incumbent solutions with a certain probability. Whenever the last (largest) neighborhood is attained, the search continues with the first neighborhood. This is repeated until some *stopping criterion* is met. Furthermore, intermediate infeasible solutions are allowed and infeasibilities are penalized. At the end, the best feasible solution s_{best} encountered during the search is returned. Finally, in order to reduce the search space, the graph pruning and time window tightening techniques, described by Cordeau, 2006, are applied prior to starting the optimization procedure.

We adapted this state-of-the-art method in the following ways. First, a new evaluation function $\hat{f}(s)$ that accommodates the characteristics of the HDARP has been devised. Besides routing costs and constraint violations, also the total waiting time with passenger aboard has to be considered:

$$\hat{f}(s) = \hat{c}(s) + \rho \hat{v}(s) + \sum_{r \in R} \hat{\alpha}_r \hat{q}_r(s) + \hat{\beta} \hat{d}(s) + \hat{\gamma} \hat{w}(s) + \hat{\tau} \hat{t}(s). \quad (88)$$

The routing costs of a solution s are given by $\hat{c}(s)$, the total vehicle waiting time with passengers aboard by $\hat{v}(s)$, weighted by ρ . The terms $\hat{q}_r(s)$ refer to the different resource violations $r \in R$. Each of these is penalized by a separate penalization parameter $\hat{\alpha}_r$. Furthermore, duration violation $\hat{d}(s) = \sum_{k=1}^m (B_{2n+1}^k - B_0^k - T^k)^+$, time window violation $\hat{w}(s) = \sum_{i=1}^{2n} (B_i - l_i)^+$, and ride time violation $\hat{t}(s) = \sum_{i=1}^n (L_i - \bar{L})^+$ are penalized by the self-adjusting terms $\hat{\beta}$, $\hat{\gamma}$, and $\hat{\tau}$, respectively (for details see Parragh et al., 2010). A solution s can only become a new best solution s_{best} if $\hat{q}_r(s) = \hat{d}(s) = \hat{w}(s) = \hat{t}(s) = 0$ for all $r \in R$.

Second, a regret insertion procedure for the construction of the initial solution has been designed. The first m requests are each assigned to a different vehicle. All subsequent requests are inserted as follows. A regret value for every request is calculated: the best insertion positions for each request on each route are determined and the corresponding evaluation function values are calculated; the regret value for a given request corresponds to the difference in terms of evaluation function

value between its two best insertion positions on two different routes. In every iteration the request associated with the largest regret value is inserted at its best position.

Third, only 13 different neighborhoods are considered (S1–M1–C1–S2–M2–C2–S3–M3–C3–S4–M4–C4–Z), where the number in addition to the neighborhood abbreviation indicates the neighborhood size, see also Parragh (2009, Chapter 3)). Initial tests with 19 neighborhoods showed that, as in the context of the multi-objective DARP (Parragh et al., 2009b), 13 neighborhoods are sufficient.

Further, the above mentioned graph pruning techniques are only applied after the first feasible solution has been found. And last but not least, as stopping criterion a limit of 5×10^5 iterations is employed.

5. Computational experiments

All programs are implemented in C++. In the branch-and-cut algorithms CPLEX 11.0 together with Concert Technology 2.5 are employed. All experiments are carried out on a 3.2 GHz Pentium D computer with a memory of 4 GB. All solution procedures are tested on three artificial data sets. These are based on an existing data set from the literature, enriched with the different real world characteristics described above. In the following, first, the characteristics of the generated test instances are described. Then, the obtained results are discussed. First the results obtained by means of the two branch-and-cut algorithms are summarized; then, the solutions obtained by means of the adapted VNS are presented. Both procedures are first tested with $\rho = 0$ (waiting is not penalized) and then with $\rho = 100$, as suggested by the ARC, such that waiting on board a vehicle is avoided.

5.1. Test instances

The test instances used are based on the 12 “A” instances proposed by Cordeau (2006) for the standard DARP, containing 12 instances with 2–4 vehicles and 16–48 requests. In all instances time window length $l_i - e_i = 15$ min, maximum user ride time $\bar{L} = 30$ min, and service time $d_i = 3$ min for all users. For each instance three instances with different degrees of heterogeneity are generated.

Heterogeneous users are introduced based on the probabilities given in Table 2. Every transportation request consists of at most one patient. In data set “U” only seated passengers and no accompanying persons are considered. In data set “E” half of the patients are considered to be seated, 25% to be on a stretcher, and 25% to be in a wheelchair; 10% are assumed to be accompanied by someone. Eventually, in data set “I” users are transformed into seated patients, patients on a stretcher, and wheelchair passengers according to the true distribution across all static transports carried out by the ARC in the city of Graz. Furthermore, half of them are assumed to be accompanied by someone.

In each data set a different fleet configuration is used. In data set “U” a homogeneous fleet setting with only vehicles of type T0 is considered. Vehicle type T0 provides space for three seated passengers. In data set “E” a homogeneous vehicle fleet consisting of T2 vehicles, providing 2 staff seats, 1 patient seat, 1 stretcher, and 1 wheelchair place, is used; for data set “I” a heterogeneous vehicle fleet is generated. Here the original number of vehicles is randomly divided into T1 and T2 vehicles such that at least one vehicle of each type is available. Vehicle types T1 and T2 are derived from data provided by the ARC regarding their vehicle fleet.

5.2. Branch-and-cut results

Table 3 provides the results obtained by 3indexBC and 2indexBC, ignoring the penalization option regarding waiting with passengers aboard ($\rho = 0$). It contains the following information. Columns “z” give the proved optimal objective value (marked by an asterisk) or the best feasible solution found. Furthermore, the best lower bound identified within the time limit (bestLB), run times in seconds (CPU), the number of nodes visited as well as the number of cuts added are provided. The number of cuts refers to the number of user cuts generated during the optimization process. Cuts generated by CPLEX are not counted. In the last column (heurUB) the employed initial upper bounds, computed by means of one run of the adapted VNS, can be found. For all experiments reported in this section a run time limit of 4 h was used.

Table 2
Test instances.

Data set	Probability for patient to be			Probability	
	Seated	On stretcher	In wheelchair	For AP	Fleet
U	1.00	0.00	0.00	0.00	hom. (T0)
E	0.50	0.25	0.25	0.10	hom. (T2)
I	0.83	0.11	0.06	0.50	het. (T1, T2)

AP = accompanying person, hom. = homogeneous, het. = heterogeneous.

T0: 3 patient seats.

T1: 1 staff seat, 6 patient seats, 1 wheelchair place.

T2: 2 staff seats, 1 patient seat, 1 stretcher, 1 wheelchair place.

Table 33indexBC vs. 2indexBC ($\rho = 0$).

	3indexBC					2indexBC					heurUB
	z	bestLB	CPU ^a	Nodes	Cuts	z	bestLB	CPU ^a	Nodes	Cuts	
<i>U</i>											
a2-16	294.25 [*]	294.25	44.05	58	152	294.25 [*]	294.25	1.13	0	44	294.25
a2-20	344.83 [*]	344.80	1674.30	6751	2531	344.83 [*]	344.83	2.59	0	112	344.83
a2-24		415.17	14492.70	13632	2391	431.12 [*]	431.12	8.54	0	161	431.12
a3-18	300.48 [*]	300.45	4376.70	16350	933	300.48 [*]	300.48	4.55	0	249	300.48
a3-24		334	14516.40	7380	595	344.83 [*]	344.83	7.62	0	422	347.42
a3-30		467.25	14490	3953	433	494.85 [*]	494.85	9.83	0	575	494.85
a3-36		553.05	14570.80	3917	351	583.19 [*]	583.19	105.05	0	1099	584.44
a4-16	282.68 [*]	282.67	319.42	284	241	282.68 [*]	282.68	5.61	0	430	282.68
a4-24	375.02 [*]	375.02	709.89	187	240	375.02 [*]	375.02	5.60	0	347	378.13
a4-32		432.78	14520.60	2084	291	485.50 [*]	485.50	30.67	0	1056	487.81
a4-40		502.74	14468.60	88	339	557.69 [*]	557.63	8328.46	9723	6293	582.26
a4-48		571.27	14523.20	48	340	668.82	664.64	14542.60	5076	5164	709.47
\bar{U}		406.12	9058.89	4561	736		429.92	1921.02	1233	1329	436.48
<i>E</i>											
a2-16	331.16 [*]	331.16	37.54	14	101	331.16 [*]	331.13	284.17	4908	1461	331.16
a2-20	347.03 [*]	347.03	441.26	1523	817	347.03 [*]	347.03	8.06	0	150	347.03
a2-24		421.63	14537.70	7198	1255	450.25 [*]	450.21	891.24	5743	1298	450.25
a3-18	300.63 [*]	300.63	4412.52	13053	612	300.63 [*]	300.63	4.28	0	304	300.63
a3-24		340.65	14528.70	6721	505	344.91 [*]	344.91	10.15	0	505	346.22
a3-30		484.26	14514.80	3199	445	500.58 [*]	500.53	1608.63	4898	2600	500.58
a3-36		565.55	14554.10	2165	312	583.19 [*]	583.19	101.40	0	1133	585.94
a4-16	285.99 [*]	285.99	194.69	31	155	285.99 [*]	285.99	759.27	7664	1832	285.99
a4-24	383.84 [*]	383.84	1321.24	112	383	383.84	380.48	14471.60	19808	22083	390.87
a4-32		459.87	14499.90	870	562		488.14	14494.10	4733	12610	508.51
a4-40		534.35	14716.30	43	516		558.09	14518.50	2390	17918	609.08
a4-48		593.36	14625.30	0	635		665.02	14539.50	1476	11919	704.07
\bar{E}		420.69	9032.00	2911	525		436.28	5140.08	4302	6151	446.69
<i>I</i>											
a2-16	294.25 [*]	294.25	92.17	59	148	294.25 [*]	294.25	0.88	0	25	294.25
a2-20	355.74 [*]	355.74	10142.80	42432	3728	355.74 [*]	355.71	115.44	1103	428	355.74
a2-24		421.75	14553.30	13825	1324	431.12 [*]	431.12	7.06	0	236	431.12
a3-18		297.02	14538.70	17172	651	302.17 [*]	302.15	30.78	187	515	302.17
a3-24		332.49	14528.50	5570	354	344.83 [*]	344.83	6.78	0	436	344.83
a3-30		472.35	14490.80	2896	481	494.85 [*]	494.85	9.50	0	574	494.85
a3-36		566.23	14503.70	1173	263	618.63	592.69	14500.10	8742	17455	625.90
a4-16	299.05 [*]	299.05	360.75	804	296	299.05 [*]	299.05	36.51	35	606	299.05
a4-24	375.07	372.07	14499.20	4221	244	375.02 [*]	375.02	5.23	0	391	375.07
a4-32		419.60	14497.50	1073	225	486.93 [*]	486.93	40.63	0	1093	498.48
a4-40		499.03	14525.60	23	188	557.69 [*]	557.63	2600.02	2093	4692	583.53
a4-48		567.51	14630.20	32	276	678.59	663.30	14498.00	1820	9526	708.68
\bar{I}		408.09	11780.27	7440	682		433.13	2654.25	1165	2998	442.81
\bar{UEI}		411.64	9957.05	4971	648		433.11	3238.45	2233	3493	441.99

^a Run times in seconds.

When comparing the results obtained by the two branch-and-cut algorithms, 2indexBC clearly outperforms 3indexBC. 2indexBC finds the optimal solution to 29 out of 36 test instances within the 4-h time limit, while 3indexBC only finds 13. Only in one case (instance a4-24 of data set “E”), 3indexBC finds the optimal solution and proves it to be optimal, while 2indexBC does not. In 2indexBC, on average five times more cuts are generated but only half of the number of nodes are explored. Taking a closer look at the number of nodes explored in the enumeration tree, in 2indexBC, in many cases the optimal solution is already found at the root node; this is never the case for 3indexBC.

The number of cuts added during the optimization process is further analyzed. In Table 4 the percentage share of each class of cuts, as presented above, is given. While the 3-index formulation could be solved by a commercial MIP solver such as CPLEX directly, in case of the 2-index formulation constraint sets (44), (45) and (49) have to be added in a cutting plane fashion. Their percentage share is given in italic letters in Table 4. Almost 70% of the cuts generated in case of 2indexBC correspond to precedence and pairing cuts (44) and (45). Also parts of the infeasible path cuts are needed to ensure feasibility, i.e. those related to loading restrictions (49). These amount to about 2% of the total number of cuts added. Hence, only approximately 30% of the number of cuts given in Table 3 are based on additional valid inequalities.

Comparing those families of inequalities that are used in both 3indexBC and 2indexBC, infeasible path cuts make up the largest portion of cuts. In case of 3indexBC also a considerable number of subtour elimination constraints and capacity

Table 4Percentage share of each class of cuts on the total number of cuts ($\rho = 0$).

	<i>3indexBC</i>				<i>2indexBC</i>					
	Subtour elim.	Gen. order	Infeasible path	Capacity inequ.	subtour elim.	Gen. order	Infeasible path	Reachability	Pairing/ precedence	Infeasible path (load)
\bar{U}	11.40	6.12	57.18	17.64	0.93	0.19	6.14	18.27	73.61	0.03
\bar{E}	15.64	8.57	31.64	35.27	0.28	0.06	28.83	7.43	59.73	3.40
\bar{I}	13.18	6.40	68.70	3.84	0.27	0.10	18.71	15.48	62.97	2.03
\overline{UEI}	13.4	7.03	52.51	18.91	0.49	0.12	17.89	13.73	65.44	1.82

Elim. = elimination, gen. = generalized, inequ. = inequalities.

Table 5*2indexBC* ($\rho = 100$).

	<i>2indexBC</i>					
	<i>z</i>	bestLB	CPU ^a	Nodes	Cuts	heurUB
<i>U</i>						
a2-16	300.17 [*]	300.17	0.92	0	63	300.17
a2-20	345.74 [*]	345.74	2.43	0	117	345.74
a2-24	448.07 [*]	448.07	15.67	0	147	448.07
a3-18	316.78 [*]	316.78	3.59	0	204	316.78
a3-24	346.97 [*]	346.97	11.55	0	516	347.37
a3-30	501.68 [*]	501.68	23.46	0	549	502.39
a3-36	598.53 [*]	598.53	21.55	0	646	599.55
a4-16	282.68 [*]	282.68	5.54	0	357	282.68
a4-24	386.38 [*]	386.38	5.73	0	398	386.38
a4-32	493.15 [*]	493.15	83.48	0	1200	493.15
a4-40	557.94 [*]	557.90	98.35	0	1204	560.31
a4-48	707.89	697.88	14467.20	4514	4132	711.12
\bar{U}		439.66	1228.29	376	794	441.14
<i>E</i>						
a2-16	331.16 [*]	331.13	92.46	1406	657	331.16
a2-20	347.89 [*]	347.89	4.76	0	77	347.89
a2-24	461.93 [*]	461.89	231.05	1310	582	462.82
a3-18	316.78 [*]	316.78	5.99	0	224	316.78
a3-24	347.05 [*]	347.05	17.32	0	637	347.45
a3-30	501.68 [*]	501.68	27.00	0	617	504.15
a3-36	604.35 [*]	604.30	390.94	447	1820	606.08
a4-16	291.55	288.70	14479.30	76702	10520	291.55
a4-24	386.38 [*]	386.38	5.66	0	336	386.38
a4-32		491.66	14490.20	4836	9864	507.73
a4-40		560.00	14507.70	2387	21330	590.19
a4-48		697.28	14498.10	2355	7316	717.50
\bar{E}		444.56	4895.87	7454	4498	450.81
<i>I</i>						
a2-16	300.17 [*]	300.17	0.99	0	39	300.17
a2-20	356.64 [*]	356.61	28.06	42	205	356.64
a2-24	450.37 [*]	450.37	30.04	21	251	450.37
a3-18	318.47 [*]	318.47	17.17	0	311	318.47
a3-24	346.97 [*]	346.97	16.57	0	644	347.37
a3-30	501.68 [*]	501.68	10.45	0	478	501.68
a3-36	631.12	612.35	14511.20	9386	18371	633.29
a4-16	301.81 [*]	301.78	198.41	1564	1244	302.23
a4-24	386.38 [*]	386.38	7.35	0	396	386.38
a4-32	494.59	493.58	14488.90	18973	9994	499.06
a4-40	559.45 [*]	559.42	223.67	51	1589	561.35
a4-48		698.39	14516.60	2125	8138	721.40
\bar{I}		443.85	3670.78	2680	3472	448.20
\overline{UEI}		442.69	3264.98	3503	2921	446.72

^a Run times in seconds.

inequalities are generated. In *2indexBC*, excluding those cuts needed to ensure feasibility, reachability cuts form the second largest portion of cuts.

Analyzing the number of cuts added on a data set level, in *3indexBC*, the percentage share of the capacity cuts varies considerably. Data set “U” corresponds to the original homogeneous data set; only one type of passenger and only one type of

Table 6
VNS (five runs).

	$\rho = 0$						$\rho = 100$					
	Avg.	%	Best	%	CPU ^a	Waiting	Avg.	%	Best	%	CPU ^a	Waiting
<i>U</i>												
a2-16	294.25	0.00	294.25	0.00	68.20	0.57	300.17	0.00	300.17	0.00	70.20	0.00
a2-20	344.83	0.00	344.83	0.00	133.80	13.25	345.74	0.00	345.74	0.00	128.80	0.00
a2-24	431.12	0.00	431.12	0.00	187.80	22.21	448.07	0.00	448.07	0.00	202.40	0.00
a3-18	300.48	0.00	300.48	0.00	45.40	12.19	316.78	0.00	316.78	0.00	47.40	0.00
a3-24	345.26	0.12	344.83	0.00	86.80	14.47	347.37	0.11	347.37	0.11	86.60	0.00
a3-30	495.11	0.05	494.85	0.00	105.60	15.67	502.46	0.16	501.68	0.00	111.20	0.00
a3-36	583.89	0.12	583.30	0.02	162.60	55.06	602.23	0.62	599.02	0.08	175.20	0.00
a4-16	282.68	0.00	282.68	0.00	26.00	0.00	282.68	0.00	282.68	0.00	26.80	0.00
a4-24	375.04	0.00	375.02	0.00	50.80	30.90	386.92	0.14	386.38	0.00	51.20	0.00
a4-32	488.27	0.57	486.88	0.28	86.00	11.79	496.05	0.59	493.15	0.00	84.60	0.00
a4-40	565.58	1.41	561.80	0.74	130.60	0.75	563.87	1.06	559.45	0.27	130.00	0.00
a4-48	680.98		673.64		253.80	50.57	717.58		711.12		0.00	0.00
\bar{U}	432.29		431.14		111.45	18.95	442.49		440.97		92.87	0.00
<i>E</i>												
a2-16	331.16	0.00	331.16	0.00	65.60	0.00	331.16	0.00	331.16	0.00	67.20	0.00
a2-20	347.03	0.00	347.03	0.00	120.00	13.25	347.89	0.00	347.89	0.00	111.80	0.00
a2-24	450.25	0.00	450.25	0.00	160.40	16.03	462.57	0.14	461.93	0.00	172.60	0.00
a3-18	300.63	0.00	300.63	0.00	47.60	9.27	316.78	0.00	316.78	0.00	49.00	0.00
a3-24	345.59	0.20	344.91	0.00	76.20	14.47	347.37	0.09	347.05	0.00	80.60	0.00
a3-30	501.41	0.17	500.58	0.00	107.60	7.24	503.70	0.40	501.68	0.00	106.40	0.00
a3-36	583.79	0.10	583.19	0.00	161.60	58.43	607.10	0.46	605.46	0.18	161.80	0.00
a4-16	285.99	0.00	285.99	0.00	25.00	20.00	291.55	0.00	291.55	0.00	24.40	0.00
a4-24	384.03	0.05	383.84	0.00	52.60	10.88	386.64	0.07	386.38	0.00	51.00	0.00
a4-32	504.79		502.52		83.00	15.88	509.76		507.72		81.00	0.00
a4-40	588.40		585.64		121.00	12.94	595.81		590.19		121.00	0.00
a4-48	681.80		675.37		252.20	49.48	717.95		715.62		285.00	0.00
\bar{E}	442.07		440.93		106.07	18.99	451.52		450.28		109.32	0.00
<i>I</i>												
a2-16	294.25	0.00	294.25	0.00	68.40	0.57	300.36	0.06	300.17	0.00	71.40	0.00
a2-20	355.74	0.00	355.74	0.00	141.80	13.25	356.64	0.00	356.64	0.00	137.40	0.00
a2-24	431.12	0.00	431.12	0.00	211.00	22.21	450.37	0.00	450.37	0.00	207.20	0.00
a3-18	302.17	0.00	302.17	0.00	47.20	12.19	318.62	0.05	318.47	0.00	50.20	0.00
a3-24	344.99	0.05	344.83	0.00	83.60	14.47	347.55	0.17	347.37	0.11	87.20	0.00
a3-30	495.13	0.06	494.85	0.00	106.80	15.67	501.68	0.00	501.68	0.00	111.60	0.00
a3-36	619.64		618.58		170.60	43.11	630.61		627.39		207.80	0.00
a4-16	299.05	0.00	299.05	0.00	27.00	20.00	302.06	0.08	301.81	0.00	26.40	0.00
a4-24	376.19	0.31	375.07	0.01	51.60	26.81	386.89	0.13	386.38	0.00	51.40	0.00
a4-32	488.64	0.35	486.93	0.00	88.00	10.08	498.99		497.07		84.80	0.02
a4-40	563.34	1.01	561.35	0.66	132.20	2.36	563.07	0.65	561.35	0.34	132.20	0.00
a4-48	687.44		680.43		262.40	47.17	717.98		713.21		303.60	0.00
\bar{I}	438.14		437.03		115.88	18.99	447.90		446.83		122.60	0.00
\overline{UEI}	437.50		436.37		111.13	18.98	447.31		446.03		108.26	0.00

^a Run times in seconds.

vehicle are considered. Here quite a lot of capacity cuts are identified. In case of data set “E”, even more capacity cuts are separated; heterogeneous users but only one type of vehicle are considered. Fewer capacity cuts are generated in case of data set “I”. The reason is that, in addition to heterogeneous users, also heterogeneous vehicles are considered and thus the per resource capacity upper bound across all vehicles deteriorates.

The conclusion that can be drawn from these results is that, as expected, the 2-index-based formulation leads to a more efficient branch-and-cut algorithm. Therefore, in the following experiment only *2indexBC* is considered.

Table 5 provides the results obtained with *2indexBC* and $\rho = 100$. This means that waiting with passengers aboard is penalized rather severely. All results thus obtained do not contain any waiting time with passengers aboard. Penalizing waiting with passengers aboard makes the problem slightly more difficult to solve; instead of 29 out of 36 instances, 28 out of 36 instances can be solved to optimality within the 4-h time limit.

5.3. Heuristic results

Table 6 gives the results obtained by means of the adapted VNS for five random runs with $\rho = 0$ and $\rho = 100$. For each data set the average value (avg.), the best value (best), the deviation from the optimal solution (%), where known, the run time in

seconds (CPU), and the total waiting time with passengers aboard (waiting), averaged over five runs, are given. The adapted VNS is able to cope with both settings. The largest deviation from the optimal solution in case of $\rho = 0$ is 1.41%, taking average values over five runs. When considering the best values out of these five runs, the maximum gap is equal to 0.74%. In most cases the optimal solution is found. The largest deviation in case of $\rho = 100$, taking average values over five runs, is 1.06%. The largest gap, taking the best solution values out of these five runs, is equal to 0.27%. This indicates that for the adapted VNS, penalizing waiting with passengers aboard makes the problem slightly easier to solve. In both versions run times are low, less than 2 min on average. When comparing the amount of waiting time contained in the average solutions, setting $\rho = 0$ results in, on average, 18.98 min of waiting, while setting $\rho = 100$ results in no waiting for the passengers. Only for one instance (a4-32 data set “I”) one of the five random runs yielded a solution with passenger waiting time (0.08 min). Only slight adaptations were thus necessary in order to adapt the previously developed VNS for the standard DARP to the new problem version and to yield high quality solutions.

When comparing average routing costs for $\rho = 0$ and $\rho = 100$ a cost increase of about 2.5% can be observed. When comparing best values out of five runs, the largest cost increase amounts to almost 6% (instance a4-48 of data set “E”). The implications from a company perspective are that only moderate cost increases will lead to improved solutions from a customer perspective. What remains to be seen is how this relationship translates into larger problem instances. Possibly, the difference between the two extremes, only minimizing costs and avoiding user waiting time when aboard a vehicle, will increase.

6. Conclusions

In this article heterogeneous vehicles and users have been introduced into state-of-the-art models and algorithms for the standard dial-a-ride problem. Two problem formulations have been proposed: a 3-index and a more sophisticated 2-index formulation that integrates the given heterogeneous fleet requirements in a clever way. When used in a branch-and-cut framework, the 2-index formulation clearly outperforms the 3-index-based one in all three data sets considered. A variable neighborhood search heuristic, originally developed for the standard dial-a-ride problem, has also been adapted to this new problem version. High quality solutions are computed within short computation times. These results suggest that the application of the proposed method is also suitable for larger (real world) instances. Furthermore, the impact of penalizing vehicle waiting time with passengers aboard has been investigated. Heuristic results for the two extreme settings have been compared: on the one hand, only considering the cost part in the objective function, on the other hand, setting the penalization term to a rather high value. The latter setting results in solutions without user waiting time when aboard a vehicle. When comparing the results, the latter setting yields average routing costs that are only 2.5% higher than in the minimum cost setting. The maximum increase amounts to about 6%. This indicates that from a company perspective, avoiding waiting time with users aboard a vehicle does not lead to a significant cost increase.

Acknowledgments

We wish to thank Jean-François Cordeau, Karl F. Doerner, Fabien Tricoire, and two anonymous referees for their valuable comments. This work was supported by the Austrian Science Fund (FWF) under Grants #L286-N04 and #L362-N15. This support is gratefully acknowledged.

References

- Ascheuer, N., Fischetti, M., Grötschel, M., 2000. A polyhedral study of the asymmetric traveling salesman problem with time windows. *Networks* 36, 69–79.
- Balas, E., Fischetti, M., Pulleyblank, W.R., 1995. The precedence-constraint asymmetric traveling salesman polytope. *Math. Program.* 68, 241–265.
- Baldacci, R., Battarra, M., Vigo, D., 2009. Valid inequalities for the fleet size and mix vehicle routing problem with fixed costs. *Networks* 54, 178–189.
- Beaudry, A., Laporte, G., Melo, T., Nickel, S., 2009. Dynamic transportation of patients to hospitals. *OR Spectrum* 32, 77–107.
- Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., Laporte, G., 2007. Static pickup and delivery problems: a classification scheme and survey. *TOP* 15, 1–31.
- Cordeau, J.-F., 2006. A branch-and-cut algorithm for the dial-a-ride problem. *Oper. Res.* 54, 573–586.
- Cordeau, J.-F., Laporte, G., 2003. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transport. Res. Part B – Meth.* 37, 579–594.
- Cordeau, J.-F., Laporte, G., 2007. The dial-a-ride problem: models and algorithms. *Ann. Oper. Res.* 153, 29–46.
- Desrochers, M., Laporte, G., 1991. Improvements and extensions to the Miller–Tucker–Zemlin subtour elimination constraints. *Oper. Res. Lett.* 10, 27–36.
- Desrosiers, J., Dumas, Y., Soumis, F., 1986. A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. *Am. J. Math. Manage. Sci.* 6, 301–325.
- Grötschel, M., Padberg, M.W., 1985. Polyhedral theory. In: *The Traveling Salesman Problem*. Wiley, New York, pp. 251–305.
- Hanne, T., Melo, T., Nickel, S., 2009. Bringing robustness to patient flow management through optimized patient transports in hospitals. *Interfaces* 39, 241–255.
- Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. *Science* 220, 671–680.
- Lysgaard, J., 2006. Reachability cuts for the vehicle routing problem with time windows. *Eur. J. Oper. Res.* 175, 210–223.
- Melachrinoudis, E., Ilhan, A.B., Min, H., 2007. A dial-a-ride problem for client transportation in a health-care organization. *Comput. Oper. Res.* 34, 742–759.
- Mladenovic, N., Hansen, P., 1997. Variable neighborhood search. *Comput. Oper. Res.* 24, 1097–1100.
- Naddef, D., Rinaldi, G., 2002. Branch-and-cut algorithms for the capacitated VRP. In: Toth, P., Vigo, D. (Eds.), *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, vol. 9. SIAM, Philadelphia, pp. 53–84.
- Parragh, S.N., 2009. Ambulance Routing Problems with Rich Constraints and Multiple Objectives. Ph.D. Thesis, University of Vienna, Faculty of Business, Economics and Statistics.
- Parragh, S.N., Cordeau, J.-F., Doerner, K.F., Hartl, R.F., 2009a. Models and Algorithms for the Heterogeneous Dial-a-ride Problem with Driver Related Constraints. Technical Report, University of Vienna, Faculty of Business, Economics and Statistics.
- Parragh, S.N., Doerner, K.F., Gandibleux, X., Hartl, R.F., 2009b. A heuristic two-phase solution method for the multi-objective dial-a-ride problem. *Networks* 54, 227–242.

- Parragh, S.N., Doerner, K.F., Hartl, R.F., 2008a. A survey on pickup and delivery problems. Part I: transportation between customers and depot. *J. Betriebs-wirt.* 58, 21–51.
- Parragh, S.N., Doerner, K.F., Hartl, R.F., 2008b. A survey on pickup and delivery problems. Part II: transportation between pickup and delivery locations. *J. Betriebs wirt.* 58, 81–117.
- Parragh, S.N., Doerner, K.F., Hartl, R.F., 2010. Variable neighborhood search for the dial-a-ride problem. *Comput. Oper. Res.* 37, 1129–1138.
- Psaraftis, H.N., 1980. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transport. Sci.* 14, 130–154.
- Psaraftis, H.N., 1983. An exact algorithm for the single vehicle many to many dial-a-ride problem with time windows. *Transport. Sci.* 17, 351–357.
- Rekiek, B., Delchambre, A., Saleh, H.A., 2006. Handicapped person transportation: an application of the grouping genetic algorithm. *Eng. Appl. Artif. Intel.* 19, 511–520.
- Ropke, S., Cordeau, J.-F., Laporte, G., 2007. Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks* 49, 258–272.
- Ruland, K.S., Rodin, E.Y., 1997. The pickup and delivery problem: faces and branch-and-cut algorithm. *Comput. Math. Appl.* 33, 1–13.
- Toth, P., Vigo, D., 1997. Heuristic algorithms for the handicapped persons transportation problem. *Transport. Sci.* 31, 60–71.