Computers & Operations Research 168 (2024) 106705

Contents lists available at ScienceDirect

Computers and Operations Research

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

A branch-and-cut algorithm for a skip pick-up and delivery problem

José M. Belenguer [a], Maximiliano Cubillos [b], Sanne Wøhlk [c,*]

[a] Department of Statistics and Operations Research, University of Valencia, 46100 Burjassot, Valencia, Spain
[b] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci 32, Milano 20133, Italy
[c] Department of Economics and Business Economics, Aarhus University, Fuglesangs Allé 4, DK-8210 Aarhus V, Denmark

ARTICLE INFO

Keywords:
Transport
Pick-up and delivery problems
Skip transport
Waste
Branch-and-cut

ABSTRACT

In this study we present a branch-and-cut algorithm for a skip pick-up and delivery problem. The study is motivated by a real-life problem in which full skips are transported from waste drop-off stations to treatment facilities where they are emptied, and then brought back to the original drop-off station. The transportation of the skips is done by trucks with the capacity of carrying two containers at a time. The planning problem is to design the routes of the trucks that perform the collection to satisfy a number of requests in a planning period. A truck route starts at the first pick-up, the truck then performs a sequence of pick-ups, treatments, and deliveries, and the route ends at the last delivery. From the truck perspective, the three actions of pick-up, treatment, and delivery can be performed in any order that respects the vehicle capacity of two and the route duration constraint, but for the single request, the three actions must be performed in the stated order. The problem is formulated as a mixed integer linear problem and several classes of valid inequalities are proposed and integrated into a branch-and-cut algorithm.

1. Introduction

In this paper, we present a branch-and-cut algorithm for the skip pick-up and delivery problem with fixed return (SPDP-R) first presented by Wøhlk and Laporte (2022). In the SPDP-R, a fleet of vehicles must satisfy a set of transportation requests. Each request is characterized by a pick-up location and a corresponding treatment location. The vehicle's operation consists of retrieving a large container (skip) at the pick-up location, transport it to the treatment location where the skip is emptied, and then return the empty skip to the original pick-up location. Given the size of the skips, each vehicle can carry up to two skips at a time. A sufficiently large fleet of vehicles is available to perform the service of the requests. Routes are limited to a maximum time duration. Each vehicle start its route at the first pick-up and end at the last delivery, and the objective is to minimize the travel costs and the fixed costs of using each vehicle to serve all the requests.

The SPDP-R originates from the transportation of skips between recycling centers and recycling treatment facilities. Recycling centers are at the disposal of citizens to drop materials for recycling, which accumulate in the skips. When skips are full, they are transported to a treatment facility which depends on the content of the skip. There, the skip is emptied and then returned to its origin. In addition to waste collection, similar situations arise in other contexts, such as within the building industry (Rabbani et al., 2016).

From the vehicle's perspective, there are three actions: (1) pick-up a full skip, (2) empty a skip at the treatment facility, and (3) return an empty skip to its delivery location, which coincides with the original pick-up location of that skip. For a given request, those actions must be performed in the stated order, but not necessarily in sequence. This means that empty skips must be returned by the same vehicle, but other actions can take place before its return. Since each vehicle has a capacity of carrying two skips, the possible load of a vehicle is a combination of empty and full skips with none, one, or two skips. The complexity of the problem stem from the combinatorial nature of the possible actions combined with the fact that skips must return to their original location. In Fig. 1, we illustrate how actions depend on the current load of a vehicle.

The SPDP-R falls into a more general classification of problems addressed in the literature as pick-up and delivery problems (PDPs). The class of PDPs is large and rich (see Berbeglia et al. (2007), Parragh et al. (2008), and Battarra et al. (2014) for comprehensive reviews). In the PDP, objects or people have to be transported from an origin to a destination. These problems are classified into three main groups. Firstly, many-to-many problems allow any point to act as a source or destination for any commodity. Secondly, one-to-many-to-one problems involve commodities initially at the depot and destined for customer vertices, and vice versa. Lastly, in one-to-one problems,

* Corresponding author.
E-mail address: sanw@econ.au.dk (S. Wøhlk).

https://doi.org/10.1016/j.cor.2024.106705
Received 31 December 2023; Received in revised form 24 April 2024; Accepted 15 May 2024
Available online 18 May 2024

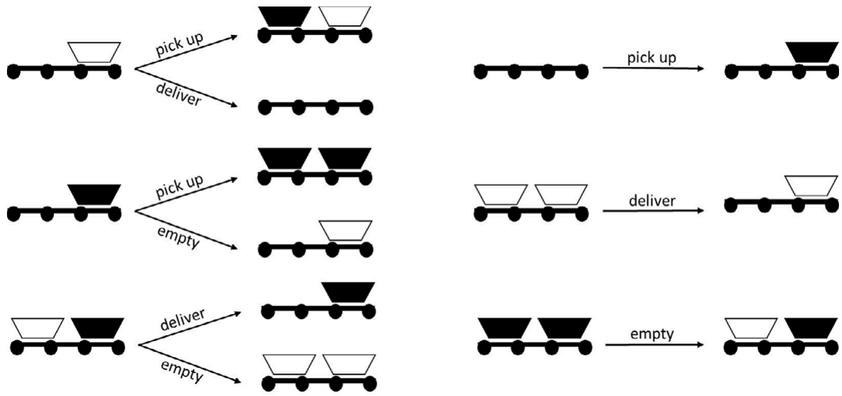

Fig. 1. Illustration of how the possible actions depend on the current load of the vehicle. Full skips are shown in black, while empty skips are shown in white. For example, the lower right illustration shows that for a vehicle carrying two full skips, the only possible next action is to empty one of them.
Source: Wøhlk and Laporte (2022).

each commodity has a specific origin and destination, typical in courier, door-to-door transportation services. While the SPDP-R falls into the last category, our problem is double-paired: one-to-one assignment with delivery at the origin by the same vehicle.

Table 1 summarizes the literature on PDPs that is most relevant to our work. We report assumptions regarding the capacity of the vehicles (number of skips), number of disposal sites, the inclusion of return trips, and solution method. De Meulemeester et al. (1997) was the first to present a PDP as a problem of transporting full and empty skips. The authors consider the capacity of vehicles to be one skip, one disposal site, and two type of customers. Similar assumptions are considered by Bodin et al. (2000), Archetti and Speranza (2005), and Rabbani et al. (2016). Several variants of skip PDPs have been studied, and their main assumptions include more than one disposal site (Aringhieri et al., 2004; Archetti et al., 2005; Baldacci et al., 2006; Benjamin and Beasley, 2010; Wy et al., 2013; Li et al., 2018; Raucq et al., 2019; Wøhlk and Laporte, 2022), time windows (Benjamin and Beasley, 2010; Wy et al., 2013; Li et al., 2018; Raucq et al., 2019), driver's rest time (Benjamin and Beasley, 2010; Wy et al., 2013), precedence constraints (Li et al., 2018), and heterogeneous fleet (Raucq et al., 2019). While many papers approach the skip PDP variations with heuristics (Bodin et al., 2000; Aringhieri et al., 2004; Archetti and Speranza, 2005; Benjamin and Beasley, 2010; Wy et al., 2013; Rabbani et al., 2016; Wøhlk and Laporte, 2022), several studies also take a mathematical programming approach to optimizing the problem. De Meulemeester et al. (1997) approaches the problem using branch-and-bound, Baldacci et al. (2006) utilizes branch-and-cut, Li et al. (2018) employs Benders decomposition, and Raucq et al. (2019) applies column generation.

Few studies consider the case where vehicles can carry two skips at a time. Archetti et al. (2005) study the problem of delivering skips from a joint depot to customers, and show that if the vehicle capacity is two, then the problem can be solved in polynomial time, but if it is larger than two, then the problem is NP-hard. However, it should be noted that the problem has a less complex structure than our problem, as they do not consider the return of skips. The problem studied in Raucq et al. (2019) origins from transport of full and empty skips in the context of industrial waste. Two types of vehicles that can carry either one or two skips are used, and the problem is approached by a heuristic columns generation algorithm, as opposed to our problem, they consider these transports independently, whereas in our problem, the skips stay on the vehicle after being emptied. Also originating from a waste application, Wøhlk and Laporte (2022) study the SPDP-R as well as a version of the problem which allows for more flexibility regarding the return of skips. Mathematical models are presented for both versions, and they are solved by a meta-heuristic inspired by a variable neighborhood search. It is the version of their problem with fixed return that we consider in this paper. As presented in Table 1, our paper focuses on an exact solution method for the SPDP-R with capacity of two skips.

The contribution of this paper is twofold. First, taking the model of Wøhlk and Laporte (2022) as a starting point, we derive several classes of valid inequalities for the SPDP-R and propose an improved mathematical formulation. Second, we integrate them into and branch-and-cut algorithm, and, through extensive computational experiments, show that our valid inequalities yield a stronger model.

The rest of this paper is structured as follows. In Section 2, we describe the problem and present the mathematical model. Section 3 presents a mathematical model for the SPDP-R. In Section 4, we present the valid inequalities we use to strengthen our mathematical model. We divide this section into two parts: first, presenting the valid inequalities presented in Wøhlk and Laporte (2022), and, second, the new inequalities we propose for the problem. Section 5 describes the branch-and-cut algorithm. In Section 6 we present the results of our computational experiments. Finally, Section 7 presents conclusions and future work.

2. Problem description

Let R be a set of n requests to satisfy. Each request consists of the transportation of a full skip from a pick-up location to a treatment facility, where the skip is emptied; and the delivery of the empty skip to the original pick-up location. Each request is assigned to a vehicle which must perform services of the requests within a fixed time limit T. We assume that there is a sufficiently large set K of vehicles to meet the requests. The vehicles have the capacity of carrying two skips which can be moved in any order. Each vehicle performs an open route which starts at its first pick-up and ends at its final delivery. Each of the three operations – pick-up, treatment, and delivery – has a fixed operation time.

The objective of the problem is to identify the set of routes that satisfy all requests that minimizes the total costs and respect the time limits and the capacity of the vehicles. The total cost includes both the transportation costs and the fixed cost for each vehicle used in the solution.

2.1. Mathematical model

In this section, we provide the mathematical model for the problem as it is stated in Wøhlk and Laporte (2022). However, to ease the description, we formulate the model based on a complete graph and define a set of arcs A_0 which contains the arcs that are not included in the graph of that paper. Below, we merely fix the corresponding variables to zero.

We define the problem on a complete directed graph $G(N, A)$, with a node set $N = \{0, \dots, 3n+1\}$ and an arc set A. Nodes 0 and $d = 3n+1$ are

Table 1
Skip pick-up and delivery problems related to our work. MC: multi-commodity, TW: time windows, DR: driver rest time, PC: precedence constraints, HF: heterogeneous fleet, E: exact, H: heuristic.

Reference	Vehicle capacity	Disposal sites	Return	Constraints	Solution method
De Meulemeester et al. (1997)	1	1	No	MC	E, H
Bodin et al. (2000)	1	1	No		H
Aringhieri et al. (2004)	1	Many	No		H
Archetti and Speranza (2005)	1	1	No		H
Archetti et al. (2005)	2	Many	No		H
Baldacci et al. (2006)	1	Many	No		E
Benjamin and Beasley (2010)	1	Many	No	TW, DR	H
Wy et al. (2013)	1	Many	No	TW, DR, MC	H
Rabbani et al. (2016)	1	1	No		H
Li et al. (2018)	1	Many	No	TW, PC	E, H
Raucq et al. (2019)	2	Many	No	TW, MC, HF	E, H
Wøhlk and Laporte (2022)	2	Many	Yes		H
Our paper	2	Many	Yes		E

dummy nodes which denote the origin and destination dummy nodes, respectively. The remaining nodes of N correspond to three nodes for each request, one for each type of operation. We define three node subsets: $N_P = \{1,\dots,n\}$ for pick-up nodes, $N_T = \{n+1,\dots,2n\}$ for treatment nodes, and $N_D = \{2n+1,\dots,3n\}$ for delivery nodes. With this notation, each request i is associated with a pick-up node $i \in N_P$, a treatment node $n+i \in N_T$, and a delivery node $2n+i \in N_D$. Each node $i \in N$ is associated with a service time s_i. Each arc $(i,j) \in A$, is associated with a travel cost c_{ij}, a travel time t_{ij}, and a fixed cost $\bar{c}$ is associated with each vehicle used in the solution. We do not assume that the triangle inequality is respected since it is not always the case in the real life data used in Wøhlk and Laporte (2022), but we do assume that both travel costs and travel times are symmetric. Because the vehicles perform open routes, we have $c_{0j} = t_{0j} = c_{jd} = t_{jd} = 0$ for all nodes $j \in N$ and $s_0 = s_d = 0$. Define $\overline{s}_i = s_i + s_{n+i} + s_{2n+i}$ $\forall i \in N_P$, $\overline{s}_i = s_i + s_{n+i}$ $\forall i \in N_T$, and $\overline{s}_i = s_i$ $\forall i \in N_D$.

Each node is associated with a physical location, namely the location of the corresponding recycling center or treatment facility. We define L as the set of locations, and for each request i, we use $\alpha(i)$ to denote the pick-up and delivery location, while $\beta(i)$ denotes the location of the associated treatment facility.

Let $H = \{1,\dots,2n\}$ be a set of indices in which the first n values represent full skips and the remaining values represent empty skips. For each node $i \in N$ and each $h \in H$, we define the load parameter q_i^h as follows:

$$q_i^h = \begin{cases} 1 & i \in N_P \cup N_T, i = h, \\ -1 & i \in N_T \cup N_D, i = h+n, \\ 0 & \text{otherwise.} \end{cases}$$

In addition, we define auxiliary decision variables Q_i^h for all nodes $i \in N$ and for $h \in H$ as follows:

$$Q_i^h = \text{fraction of skip } h \text{ (full or empty) on the vehicle when it leaves node } i \quad \forall i \in N, \forall h \in H.$$

The constraints in the model force all Q_i^h variables to take binary values. We define Q_i^h for the dummy nodes 0 and d, and fix $Q_0^h = Q_d^h = 0$ for all $h \in H$. We fix $Q_i^h = 1$ for nodes $i \in N_P \cup N_T$ for the cases where $i = h$, since it represents particular pick-up and empty operations. Furthermore, we fix $Q_i^h = 0$ for nodes $i \in N_T \cup N_D$ in the cases where $i = n+h$ since it represents particular empty, or delivery operations. We define Q_0' and Q_1 as the subset of Q_i^h-variables thereby fixed to zero and one, respectively.

For all arcs $(i,j) \in A \setminus \{(0,d)\}$ we define the following decision variables:

$$x_{ij} = \begin{cases} 1 & \text{if arc } (i,j) \text{ is used,} \\ 0 & \text{otherwise,} \end{cases}$$

The integer variable $x_{0,d}$ represents the number of unused vehicles. Due to infeasibility of sequences of actions, some arcs in the graph are not allowed. We denote this set of arcs as A_0, and define it a follows:

$$A_0 = \{(0,j), j \in N_T \cup N_D\} \cup \{(i,d), i \in N_P \cup N_T\} \cup \{(i,2n+i), i \in N_P\} \cup \{(n+i,i), i \in N_P \cup N_T\} \cup \{(2n+i,i), i \in N_P\} \cup \{(d,0)\}$$

In the model, we fix the variables corresponding to the arcs in A_0 to zero.

For all nodes $i \in N \setminus \{d\}$ we define the following time variables:

B_i = time when a vehicle arrives at node $i \in N$.

The problem can be formulated as:

$$\text{minimize} \quad \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{j \in N_P} \bar{c}x_{0j} \tag{1}$$

$$\text{subject to} \quad \sum_{j \in N_P \cup \{d\}} x_{0j} = |K| \tag{2}$$

$$\sum_{i \in N_D \cup \{0\}} x_{id} = |K| \tag{3}$$

$$\sum_{j:(i,j)\in A} x_{ij} = 1 \quad i \in N \setminus \{0,d\} \tag{4}$$

$$\sum_{j:(j,i)\in A} x_{ji} = 1 \quad i \in N \setminus \{0,d\} \tag{5}$$

$$x_{ij} + x_{ji} \le 1 \quad i,j \in N : (i,j),(j,i) \in A \setminus A_0 \tag{6}$$

$$B_0 = 0 \tag{7}$$

$$B_i \le B_{n+i} \quad i \in N_P \cup N_T \tag{8}$$

$$B_i + s_i + t_{ij} - B_j \le (1 - x_{ij})M \quad (i,j) \in A, j \ne d \tag{9}$$

$$Q_i^h = 0 \quad Q_i^h \in Q_0' \tag{10}$$

$$Q_i^h = 1 \quad Q_i^h \in Q_1 \tag{11}$$

$$Q_j^h \ge Q_i^h + q_j^h - (1 - x_{ij}) \quad (i,j) \in A \setminus \{(0,d)\}, h \in H \tag{12}$$

$$Q_j^h \le Q_i^h + q_j^h + (1 - x_{ij}) \quad (i,j) \in A \setminus \{(0,d)\}, h \in H \tag{13}$$

$$0 \le \sum_{h \in H} Q_i^h \le 2 \quad i \in N \setminus \{0,d\} \tag{14}$$

$$0 \le Q_i^h \le 1 \quad i \in N, h \in H \tag{15}$$

$$0 \le B_i \le T - \overline{s}_i \quad i \in N \setminus \{0,d\} \tag{16}$$

$$x_{ij} = 0 \quad (i,j) \in A_0 \tag{17}$$

$$x_{ij} \in \{0,1\} \quad (i,j) \in A \setminus \{(0,d)\} \tag{18}$$

$$x_{0d} \ge 0 \text{ and integer.} \tag{19}$$

The objective function (1) minimizes the sum of both the travel costs and fixed costs of using each of the vehicles. Constraints (2) ensures that vehicles leave the dummy origin, and constraint (2) ensures vehicles enter the dummy destination. For non-dummy nodes, constraints (4) and (5) ensure that nodes are visited exactly once. This enforces the continuity of the routes. Constraints (6) prevents two-nodes cycles. The time of the visit to the nodes is controlled by constraints (7), (8), and (9). In constraint (9), it is sufficient to use $M = T + \max_{i\in N}\{s_i\} + \max_{(i,j)\in A}\{t_{ij}\}$. The flow of skips is controlled by constraints (10)–(13). Constraints (14) define the capacity of the vehicles. Finally, constraints (15)–(19) define the domain of the variables Q_i^h, B_i, and x_{ij}.

3. Improved mathematical model

We have made a number of improvements to the model presented in Section 2. In this section, we present our model, which we will refer to in the following as the BASE model.

Because constraints (6) are not necessary for the feasibility of the model, we do not include them in our BASE model, and as they are further dominated by constraints described in Section 4, we do not add them in our algorithm either. Furthermore, we note that constraints (8) can be tightened as in constraints (26) below, where we note that the multiplication by $x_{i,n+i}$ is necessary because the triangle inequality is not necessarily satisfied.

By the definition of $\mathcal{Q}_0'$ and $\mathcal{Q}_1$, some Q_i^h-variables are fixed to zero or one. However, it can be observed that further variables can be fixed to zero as follows. When we pick up a full skip, the corresponding empty skip cannot be on the vehicle, thus we fix $Q_i^h = 0$ for nodes $i \in N_P$ in the cases where $h = n+i$. Using a similar argument, we fix $Q_i^h = 0$ for nodes $i \in N_T$ in the cases where $i = n+h$ and for nodes $i \in N_D$ in the cases where $i = 2n+h$. We define $\mathcal{Q}_0''$ as the subset of Q_i^h-variables thereby fixed to zero, and set $\mathcal{Q}_0 = \mathcal{Q}_0' \cup \mathcal{Q}_0''$. This affects constraints (10) (constraints (28) in the model below).

Furthermore, we have tightened the 'Big-M' value in constraints (9) (constraints (27) in our model below) to $M_{ij} = T - \bar{s}_i + s_i + t_{ij}$, and tighten the lower bounds b_i for B_i from zero to

$$b_{n+i} = \min\{s_i + t_{i,n+i}, m^1_{n+i}, m^2_{n+i}, m^3_{n+i}\}, n+i \in N_T$$
$$b_{2n+i} = \min\{s_i + s_{n+i} + t_{n+i,2n+i}, m^4_{2n+i}, m^5_{2n+i}, m^6_{2n+i}\}, 2n+i \in N_D$$

where

$$m^1_{n+i} = \min_{j\in N_P, j\neq i}\{s_i + s_j + t_{j,n+i}\}$$
$$m^2_{n+i} = \min_{n+j\in N_T, j\neq i}\{s_i + s_j + s_{n+j} + t_{n+j,n+i}\}$$
$$m^3_{n+i} = \min_{2n+j\in N_D, j\neq i}\{s_i + s_j + s_{n+j} + s_{2n+j} + t_{2n+j,n+i}\}$$
$$m^4_{2n+i} = \min_{j\in N_P, j\neq i}\{s_i + s_{n+i} + s_j + t_{j,2n+i}\}$$
$$m^5_{2n+i} = \min_{n+j\in N_T, j\neq i}\{s_i + s_{n+i} + s_j + s_{n+j} + t_{n+j,2n+i}\}$$
$$m^6_{2n+i} = \min_{2n+j\in N_D, j\neq i}\{s_i + s_{n+i} + s_j + s_{n+j} + s_{2n+j} + t_{2n+j,2n+i}\}$$

while we keep $b_i = 0$ for $i \in N_P$, $b_0 = 0$, and set $B_0 = 0$.

Finally, we have performed an extensive analysis of constraints (12)–(14), and show that they can be replaced by constraints (30)–(62) below. Note, that even though it seems to be more constraints, there are actually significantly fewer. Because the argument for this replacement is quite elaborate, we have devoted Appendix A to this analysis.

This leaves us with the following BASE model, where (21)–(29) and (63)–(67) are identical to constraints (2)–(11) and (15)–(19) in the original model, except from the above mentioned changed to constraints (8) and (10) (corresponding to constraints (26) and (28) in the model below), and the exclusion of constraints (6). As stated above, constraints (30)–(62) replace constraints (12)–(14).

(BASE)

$$\text{minimize} \quad \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{j\in N_P} \bar{c}x_{0j} \tag{20}$$

$$\text{subject to} \quad \sum_{j\in N_P\cup\{d\}} x_{0j} = |K| \tag{21}$$

$$\sum_{i\in N_D\cup\{0\}} x_{id} = |K| \tag{22}$$

$$\sum_{j:(i,j)\in A} x_{ij} = 1 \quad i \in N\setminus\{0,d\} \tag{23}$$

$$\sum_{j:(j,i)\in A} x_{ji} = 1 \quad i \in N\setminus\{0,d\} \tag{24}$$

$$B_0 = 0 \tag{25}$$

$$B_i + s_i + t_{i,n+i}x_{i,n+i} \le B_{n+i} \quad \forall i \in N_P \cup N_T \tag{26}$$

$$B_i + s_i + t_{ij} - B_j \le (1 - x_{ij})M_{ij} \quad (i,j) \in A\setminus A_0, i \neq 0, j \neq d \tag{27}$$

$$Q_i^h = 0 \quad Q_i^h \in \mathcal{Q}_0 \tag{28}$$

$$Q_i^h = 1 \quad Q_i^h \in \mathcal{Q}_1 \tag{29}$$

$$x_{i,n+i} - 1 \le -Q_i^h + Q_{n+i}^h \le 1 - x_{i,n+i} \quad i \in N_P, h \in H\setminus\{i,n+i\} \tag{30}$$

$$x_{ij} \le Q_j^i \quad i,j \in N_P, i \neq j \tag{31}$$

$$x_{ij} - 1 \le -Q_i^h + Q_j^h \le 1 - x_{ij} \quad i,j \in N_P, i\neq j, h \in H\setminus\{i,n+i,j,n+j\} \tag{32}$$

$$x_{i,n+j} \le Q_{n+j}^i \quad i,j \in N_P, i \neq j \tag{33}$$

$$x_{i,n+j} \le Q_i^j \quad i,j \in N_P, i \neq j \tag{34}$$

$$x_{i,n+j} - 1 \le -Q_i^h + Q_{n+j}^h \le 1 - x_{i,n+j} \quad i,j \in N_P, i\neq j, h \in H\setminus\{i,n+i,j,n+j\} \tag{35}$$

$$x_{i,2n+j} \le Q_{2n+j}^i \quad i,j \in N_P, i \neq j \tag{36}$$

$$x_{i,2n+j} \le Q_i^{n+j} \quad i,j \in N_P, i \neq j \tag{37}$$

$$x_{i,2n+j} - 1 \le -Q_i^h + Q_{2n+j}^h \le 1 - x_{i,2n+j} \quad i,j \in N_P, i\neq j, h \in H\setminus\{i,n+i,j,n+j\} \tag{38}$$

$$x_{n+i,2n+i} - 1 \le -Q_{n+i}^h + Q_{2n+i}^h \le 1 - x_{n+i,2n+i} \quad i \in N_P, h \in H\setminus\{i,n+i\} \tag{39}$$

$$x_{n+i,j} \le Q_j^{n+i} \quad i,j \in N_P, i \neq j \tag{40}$$

$$x_{n+i,j} - 1 \le -Q_{n+i}^h + Q_j^h \le 1 - x_{n+i,j} \quad i,j \in N_P, i\neq j, h \in H\setminus\{i,n+i,j,n+j\} \tag{41}$$

$$x_{n+i,n+j} \le Q_{n+i}^j \quad i,j \in N_P, i \neq j \tag{42}$$

$$x_{n+i,n+j} \le Q_{n+j}^{n+i} \quad i,j \in N_P, i \neq j \tag{43}$$

$$x_{n+i,n+j} - 1 \le -Q_{n+i}^h + Q_{n+j}^h \le 1 - x_{n+i,n+j} \quad i,j \in N_P, i\neq j, h \in H\setminus\{i,n+i,j,n+j\} \tag{44}$$

$$x_{n+i,2n+j} \le Q_{2n+j}^{n+i} \quad i,j \in N_P, i \neq j \tag{45}$$

$$x_{n+i,2n+j} \le Q_{n+i}^{n+j} \quad i,j \in N_P, i \neq j \tag{46}$$

$$x_{n+i,2n+j} - 1 \le -Q_{n+i}^h + Q_{2n+j}^h \le 1 - x_{n+i,2n+j} \quad i,j \in N_P, i\neq j, h \in H\setminus\{i,n+i,j,n+j\} \tag{47}$$

$$x_{2n+i,j} + Q_j^i \le 1 \quad i,j \in N_P, i \neq j \tag{48}$$

$$x_{2n+i,j} + Q_{2n+i}^j \le 1 \quad i,j \in N_P, i \neq j \tag{49}$$

$$x_{2n+i,j} + Q_j^{n+i} \le 1 \quad i,j \in N_P, i \neq j \tag{50}$$

$$x_{2n+i,j} + Q_{2n+i}^{n+j} \le 1 \quad i,j \in N_P, i \neq j \tag{51}$$

$$x_{2n+i,j} - 1 \le -Q_{2n+i}^h + Q_j^h \le 1 - x_{2n+i,j} \quad i,j \in N_P, i\neq j, h \in H\setminus\{i,n+i,j,n+j\} \tag{52}$$

$$x_{2n+i,n+j} \le Q_{2n+i}^j \quad i,j \in N_P, i \neq j \tag{53}$$

$$x_{2n+i,n+j} - 1 \le -Q_{2n+i}^h + Q_{n+j}^h \le 1 - x_{2n+i,n+j} \quad i,j \in N_P, i\neq j, h \in H\setminus\{i,n+i,j,n+j\} \tag{54}$$

$$x_{2n+i,2n+j} \le Q_{2n+i}^{n+j} \quad i,j \in N_P, i \neq j \tag{55}$$

$$x_{2n+i,2n+j} - 1 \le -Q_{2n+i}^h + Q_{2n+j}^h \le 1 - x_{2n+i,2n+j} \quad i,j \in N_P, i\neq j, h \in H\setminus\{i,n+i,j,n+j\} \tag{56}$$

$$x_{2n+i,d} + \sum_{h\in H} Q_{2n+i}^h \le 1 \quad 2n+i \in N_D \tag{57}$$

$$x_{0,i} + \sum_{h\in H} Q_i^h \le 2 \quad i \in N_P \tag{58}$$

$$\sum_{2n+i\in N_D, i\neq j} x_{2n+i,n+j} + \sum_{h\in H} Q_{n+j}^h \le 2 \quad n+j \in N_T \tag{59}$$

$$\sum_{2n+i\in N_D, i\neq j} x_{2n+i,2n+j} + \sum_{h\in H} Q_{2n+j}^h \le 1 \quad 2n+j \in N_D \tag{60}$$

$$\sum_{j\in N_P, j\neq i} x_{i,j} + \sum_{h\in H} Q_i^h \leq 2 \quad i \in N_P \tag{61}$$

$$\sum_{j\in N_P, j\neq i} x_{n+i,j} + \sum_{h\in H} Q_{n+i}^h \leq 2 \quad n+i \in N_T \tag{62}$$

$$0 \leq Q_i^h \leq 1 \quad i \in N, h \in H \tag{63}$$

$$b_i \leq B_i \leq T - \bar{s}_i \quad i \in N \setminus \{0, d\} \tag{64}$$

$$x_{ij} = 0 \quad (i,j) \in A_0 \tag{65}$$

$$x_{ij} \in \{0,1\} \quad (i,j) \in A \setminus \{(0,d)\} \tag{66}$$

$$x_{0d} \geq 0 \text{ and integer.} \tag{67}$$

4. Valid inequalities

In this section we present several families of valid inequalities that can be included into the model presented in Section 3 in order to strengthen its linear relaxation. First, in Section 4.1 we present the valid inequalities proposed by Wøhlk and Laporte (2022) for the skip pick-and-delivery problem. Second, in Section 4.2 we show the new valid inequalities we have proposed for this problem. More emphasis in the description of the inequalities is put in the new inequalities rather than the ones proposed in the original paper. In Appendix B, we state two further classes of valid inequalities which were originally presented in Wøhlk and Laporte (2022) and Cubillos (2022), respectively. Our branch-and-cut does not find these inequalities to be violated for any of the tested instances, and we conjecture that they are redundant for our improved model. Indeed, we prove this for one of them in the appendix.

4.1. Known valid inequalities

In capacitated routing problems, rounded capacity inequalities are usually included as valid inequalities to establish capacity lower bounds. The same logic can be used in our problem based on the fact that a vehicle can carry at most two skips. In (68) we present a family of inequalities with the lower bound of the number of vehicles that can enter or leave specific subset of nodes. Even though these inequalities are pairwise equivalent, we choose to state them as they are presented in the source.

$$\begin{aligned}
&\textstyle\sum_{i\in N_P} \sum_{j\in N_T\cup N_D\cup\{0\}} x_{ji} \geq \lceil n/2 \rceil \\
&\textstyle\sum_{i\in N_P} \sum_{j\in N_T\cup N_D} x_{ij} \geq \lceil n/2 \rceil \\
&\textstyle\sum_{i\in N_T} \sum_{j\in N_P\cup N_D} x_{ji} \geq \lceil n/2 \rceil \\
&\textstyle\sum_{i\in N_T} \sum_{j\in N_P\cup N_D} x_{ij} \geq \lceil n/2 \rceil \\
&\textstyle\sum_{i\in N_D} \sum_{j\in N_P\cup N_T} x_{ji} \geq \lceil n/2 \rceil \\
&\textstyle\sum_{i\in N_D} \sum_{j\in N_P\cup N_T\cup\{d\}} x_{ij} \geq \lceil n/2 \rceil .
\end{aligned} \tag{68}$$

Similar rounded capacity constraints can also be added when we consider a specific action at the same location. Let $N_{P_{|l}}$ be the set of nodes associated with a pick-up of a container at location l. Similarly, let $N_{T_{|l}}$ and $N_{D_{|l}}$ be the sets of nodes corresponding to emptying and delivery at location l, respectively. The family of valid inequalities for specific location and action are presented in (69):

$$\begin{aligned}
&\textstyle\sum_{i\in N_{P_{|l}}} \sum_{j\in\{0\}\cup(N_P\setminus N_{P_{|l}})\cup N_T\cup N_D} x_{ji} \geq \lceil |N_{P_{|l}}|/2 \rceil \quad l \in L \\
&\textstyle\sum_{i\in N_{T_{|l}}} \sum_{j\in N_P\cup(N_T\setminus N_{T_{|l}})\cup N_D} x_{ji} \geq \lceil |N_{T_{|l}}|/2 \rceil \quad l \in L \\
&\textstyle\sum_{i\in N_{D_{|l}}} \sum_{j\in N_P\cup N_T\cup(N_D\setminus N_{D_{|l}})} x_{ji} \geq \lceil |N_{D_{|l}}|/2 \rceil \quad l \in L.
\end{aligned} \tag{69}$$

Traditional subtour elimination constraints can be lifted to pick-up and delivery problems as shown in Cordeau (2006). For each pair $n+i, n+j \in N_T$ and triplets $n+i, n+j, n+h \in N_T$ of nodes, we have added the lifted subtour elimination constraints of Cordeau (2006) for pairs and triplets. In (70) we show the case for pairs of nodes. The extension to the case of triplets of nodes is straightforward and results from a change of the right-hand side to 2 instead of 1.

$$\begin{aligned}
&x_{n+i,n+j} + x_{n+i,j} + x_{n+j,n+i} + x_{n+j,i} \leq 1 \quad i,j \in N_P, i \neq j. \\
&x_{n+i,n+j} + x_{2n+i,n+j} + x_{n+j,n+i} + x_{2n+j,n+i} \leq 1 \quad i,j \in N_P, i \neq j.
\end{aligned} \tag{70}$$

Wøhlk and Laporte (2022) shows that the constraints lifted by Cordeau (2006) can be lifted even further for pairs of triplets in N_P and N_D based on the specific structure of the problem for pairs and triplets in N_P and N_D. This provides the two sets of constraints in (71) for pairs of nodes. The extension to triplets is straightforward and results from a change of the right-hand side to 2 instead of 1.

$$\begin{aligned}
&x_{ij} + x_{n+i,j} + x_{2n+i,j} + x_{ji} + x_{n+j,i} + x_{2n+j,i} \leq 1 \quad i,j \in N_P, i \neq j \\
&x_{2n+i,2n+j} + x_{2n+i,n+j} + x_{2n+i,j} + x_{2n+j,2n+i} \\
&\qquad + x_{2n+j,n+i} + x_{2n+j,i} \leq 1 \quad i,j \in N_P, i \neq j.
\end{aligned} \tag{71}$$

We included families of symmetry breaking constraints. Let $\alpha(i)$ and $\beta(i)$ be the pick-up and treatment locations for the request i, respectively. The first set of inequalities (72) eliminates the symmetries when two equal actions are performed sequentially. The family of equations in (73) eliminates the symmetry of different actions performed in sequence at the same location.

$$\begin{aligned}
&x_{ij} = 0 \quad i,j \in N_P, \alpha(i) = \alpha(j), i > j \\
&x_{ij} = 0 \quad i,j \in N_T, \beta(i) = \beta(j), i > j \\
&x_{ij} = 0 \quad i,j \in N_D, \alpha(i) = \alpha(j), i > j;
\end{aligned} \tag{72}$$

$$\begin{aligned}
&x_{ij} = 0 \quad i \in N_P, j \in N_D, \alpha(i) = \alpha(j) \\
&x_{ij} = 0 \quad i \in N_T, j \in N_P, \beta(i) = \alpha(j) \\
&x_{ij} = 0 \quad i \in N_D, j \in N_T, \alpha(i) = \beta(j).
\end{aligned} \tag{73}$$

In (74) we consider the case of two requests that are exactly the same – same pick-up and same treatment –, in which case the request with the smallest ID should be picked up first, or at least not later than the other.

$$B_i \leq B_j \quad \forall i,j \in N_P, i < j, \alpha(i) = \alpha(j), \beta(i) = \beta(j); \tag{74}$$

Finally, we consider a lower bound of the number of vehicles needed to service all requests to be the sum of the travel and service times divided by the maximum time available for one vehicle. We add this lower bound using (75):

$$\sum_{j\in N_P} x_{0j} \geq \left\lceil \Big(\sum_{r\in R} \tilde{t}_{\alpha(r)\beta(r)} + \sum_{i\in N} s_i\Big)/T \right\rceil, \tag{75}$$

where $\tilde{t}_{ij}$ is the fastest travel time from i to j.[1] Note that due to the vehicle capacity of 2, we only consider the traveling from pick-up to treatment, and not the return trip.

4.2. New inequalities

In this section, we propose six new families of valid inequalities for the problem. Some families consist of a significant number of inequalities and in those cases, we only argue for validity of the first one. Besides these new inequalities, the standard sub tour elimination constraints are also valid for this problem

$$\sum_{i\in S} \sum_{j\in S, j\neq i} x_{ij} \leq |S| - 1 \quad \forall i,j \in S \subseteq N \setminus \{0,d\}, |S| > 2 \tag{76}$$

[1] Note that the same lower bound was used in Wøhlk and Laporte (2022) except that they used t_{ij} instead of $\tilde{t}_{ij}$. However, because the triangle inequality might not be respected, we use $\tilde{t}_{ij}$. For completeness, we have checked all 80 instances used in that paper with respect to changes caused by this. Three instances are affected by this change: The bound changes from 4 to 3 for instance C6, from 10 to 9 for instance C18, and from 14 to 13 for instance D12. We confirm that all numbers reported in that paper are still valid.

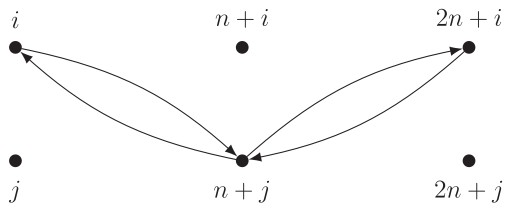

Fig. 2. Illustration of constraint (77).

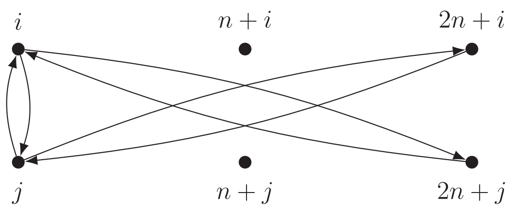

Fig. 3. Illustration of constraint (78).

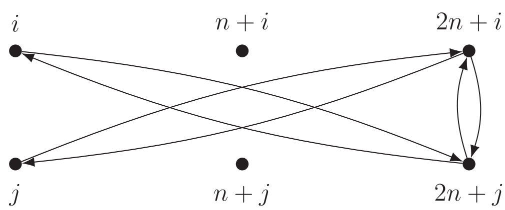

Fig. 4. Illustration of constraint (79).

4.2.1. *Cross inequalities*

We consider a family of valid inequalities that arise from the order of actions that can be performed for two requests. Let $i \in N_P$ and consider first $n+j \in N_T$. In this case, it can easily be argued that (77) holds. This is illustrated in Fig. 2. The figure represents two requests and its respective pick-ups (i,j), treatments $(n+i, n+j)$ and delivery $(2n+i, 2n+j)$ nodes. The arcs represent travels between nodes.

$$x_{i,n+j} + x_{n+j,i} + x_{2n+i,n+j} + x_{n+j,2n+i} \leq 1 \quad \forall i,j \in N_P, j \neq i. \tag{77}$$

In fact, this constraint is valid for any $j \in N \setminus \{0, d, i, n+i, 2n+i\}$, but we can lift it by considering two possible sets in which j can belong: pick-up or delivery. In the first case, where $j \in N_P$, we can lift the inequality by adding the two terms shown in (78). This is illustrated in Fig. 3. The argument for (78) is based on the fact that if we use one of the two arcs between nodes i and $2n+j$, we cannot use an arc in the opposite direction, between nodes $2n+i$ and j. This is because an arc between $2n+i$ and j involves the delivery of i, which conflicts with the fact that the other arc involves pick-up of i which has to happen before. Furthermore, using one of the arcs between i and j together with any of the arcs involving a delivery node, would result in a conflict regarding the corresponding treatment node.

$$x_{ij} + x_{ji} + x_{2n+i,j} + x_{j,2n+i} + x_{i,2n+j} + x_{2n+j,i} \leq 1 \quad \forall i,j \in N_P, j \neq i. \tag{78}$$

The second case, when $2n+j \in N_D$, follows a similar argument and results in the inequality shown in (79). This case is illustrated in Fig. 4.

$$x_{i,2n+j} + x_{2n+j,i} + x_{2n+i,2n+j} + x_{2n+j,2n+i} + x_{2n+i,j} + x_{j,2n+i} \leq 1 \qquad i,j \in N_P, j \neq i. \tag{79}$$

Two additional families of inequalities can be added, with a similar argument, when considering crossing arcs between pick-up and treatment, and crossing arcs between treatment and deliveries. The first case is illustrated in the left side of Fig. 5 and given in (80). In this case, the argument of the constraint is that if we use one of the two arcs from a pick-up i to a treatment $n+j$, we cannot use one of the arcs from or to a pick-up j to a treatment $n+i$. The same is true for a crossing arcs between treatments and deliveries, which is illustrated in the right side of Fig. 5 and given as a constraint in (81).

$$x_{i,n+j} + x_{n+j,i} + x_{j,n+i} + x_{n+i,j} \leq 1 \quad \forall i,j \in N_P, j \neq i. \tag{80}$$

$$x_{n+i,2n+j} + x_{2n+j,n+i} + x_{n+j,2n+i} + x_{2n+i,n+j} \leq 1 \quad \forall i,j \in N_P, j \neq i. \tag{81}$$

4.2.2. *Asymmetric cross inequalities*

We consider the asymmetric case of cross inequalities as described in Section 4.2.1, when one of the symmetric arcs travels to a different node – and not follows the opposite direction of the arc, adding a different variable to the inequality. For all pairs of nodes $i, j \in N_P, i \neq j$ we can add the following valid inequality:

$$x_{i,j} + x_{i,n+j} + x_{n+j,i} + x_{n+i,j} + x_{2n+j,i} \leq 1 \quad \forall i,j \in N_P, j \neq i. \tag{82}$$

Inequality (82) is illustrated in the left side of Fig. 6. The inequality comes from removing the arc from j to $n+i$ from the left side of Fig. 5 and adding arcs from i to j and from $2n+2$ to i. Following the same logic, we can add inequality (83), which is illustrated in the right side of Fig. 6.

$$x_{n+i,n+j} + x_{n+i,2n+j} + x_{2n+j,n+i} + x_{2n+i,n+j} \leq 1 \quad \forall i,j \in N_P, j \neq i. \tag{83}$$

Finally, we can flip inequalities (82) and (83) as illustrated in Fig. 7. These two cases are presented in (84) and (85), respectively.

$$x_{2n+j,2n+i} + x_{2n+j,n+i} + x_{2n+i,n+j} + x_{n+j,2n+i} + x_{2n+i,j} \leq 1 \quad \forall i,j \in N_P, j \neq i. \tag{84}$$

$$x_{n+j,n+i} + x_{n+j,i} + x_{n+i,j} + x_{j,n+i} \leq 1 \quad \forall i,j \in N_P, j \neq i. \tag{85}$$

4.2.3. *Q-Inequalities*

Based on a similar kind of logic as we applied for the Cross inequalities in Section 4.2.1, but now with focus on the load variables Q_i^h, we obtain the following valid inequalities

$$Q_i^j + Q_i^{n+j} + Q_j^i + Q_j^{n+i} \leq 1, \quad i,j \in N_P, i \neq j \tag{86}$$

$$Q_{2n+i}^j + Q_{2n+i}^{n+j} + Q_{2n+j}^i + Q_{2n+j}^{n+i} \leq 1, \quad i,j \in N_P, i \neq j \tag{87}$$

4.2.4. *Capacity inequalities for triples*

As shown in Ropke et al. (2007), the classical capacity inequalities can be lifted for the pick-up and delivery problem with pairs of pick-up and delivery locations. In this section, we show how the capacity inequalities can be lifted for our problem, when considering triples of nodes. More specifically, we consider the capacity inequality of the form

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq \rho, \qquad S \subset N \setminus \{0, d\}, |S| = 3, \tag{88}$$

and determine appropriate values for ρ based on the types of the three nodes in S. For example, is S consists of three pick-up nodes, then only a single arc among the nodes can be used due to the capacity of two. However, if S consists of two pick-up nodes and a treatment node, then two arcs among them can be used if the treatment node corresponds to the same skip as one of the two pick-up nodes, but only one arc can be used if the treatment node corresponds to a different skip.

We consider all possible subsets $S \subset N \setminus \{0, d\}$ with $|S| = 3$. For this, we define the following sets: $S_P = S \cap N_P$, $S_T = S \cap N_T$, and $S_D = S \cap N_D$. In Table 2, we have listed all possible combinations of three nodes in S, their associated sizes of their subsets ($|S_P|$, $|S_T|$, $|S_D|$), and the appropriate right hand side (ρ) for inequality (88). Note that, in cases 1 to 10, the three nodes are associated with three different request, while cases 11 to 19 with nodes associated with 2 requests, and case 20 with only one request.

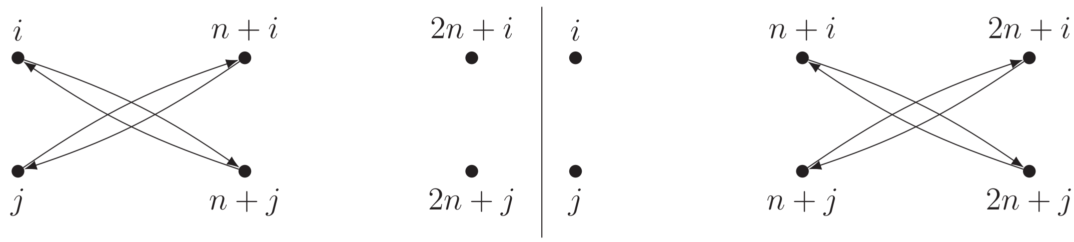

Fig. 5. Illustration of constraints (80) and (81).

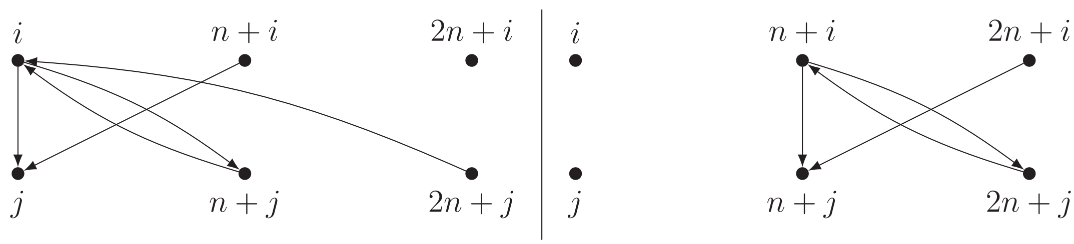

Fig. 6. Illustration of inequalities (82) and (83).

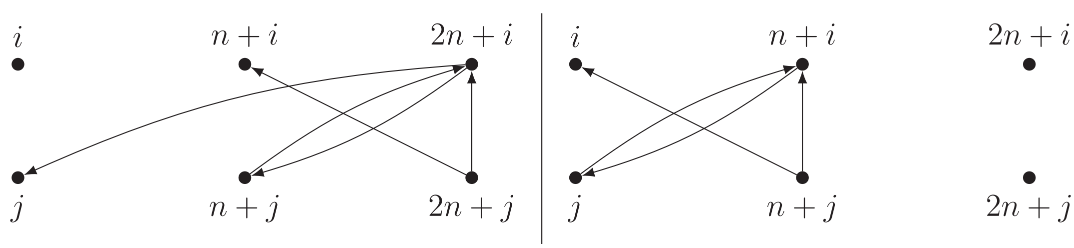

Fig. 7. Illustration of inequalities (84) and (85).

Table 2
Capacity constraint with $|S| = 3$.

Case	Nodes in S	$(\|S_P\|, \|S_T\|, \|S_D\|)$	ρ
1	$i, j, k \in N_P$	(3,0,0)	1
2	$n+i, n+j, n+k \in N_T$	(0,3,0)	1
3	$2n+i, 2n+j, 2n+k \in N_D$	(0,0,3)	1
4	$i, j \in N_P$, $n+k \in N_T$	(2,1,0)	1
5	$i, j \in N_P$, $2n+k \in N_D$	(2,0,1)	2
6	$i \in N_P$, $n+j, n+k \in N_T$	(1,2,0)	1
7	$n+i, n+j \in N_T$, $2n+k \in N_D$	(0,2,1)	1
8	$i \in N_P$, $2n+j, 2n+k \in N_D$	(1,0,2)	2
9	$n+i \in N_T$, $2n+j, 2n+k \in N_D$	(0,1,2)	1
10	$i \in N_P$, $n+j \in N_T$, $2n+k \in N_D$	(1,1,1)	2
11	$i, j \in N_P$, $n+i \in N_T$	(2,1,0)	2
12	$i, j \in N_P$, $2n+i \in N_D$	(2,0,1)	1
13	$i \in N_P$, $n+i, n+j \in N_T$	(1,2,0)	2
14	$n+i, n+j \in N_T$, $2n+i \in N_D$	(0,2,1)	2
15	$i \in N_P$, $2n+i, 2n+j \in N_D$	(1,0,2)	1
16	$n+i \in N_T$, $2n+i, 2n+j \in N_D$	(0,1,2)	2
17	$i \in N_P$, $n+i \in N_T$, $2n+j \in N_D$	(1,1,1)	2
18	$i \in N_P$, $n+j \in N_T$, $2n+i \in N_D$	(1,1,1)	1
19	$j \in N_P$, $n+i \in N_T$, $2n+i \in N_D$	(1,1,1)	2
20	$i \in N_P$, $n+i \in N_T$, $2n+i \in N_D$	(1,1,1)	2

4.2.5. Lifted capacity inequalities

In this section, we present lifted capacity inequalities for our problem. For this, let $S \subset N \setminus \{0, d\}$ be a set of nodes such that if a pick-up node $i \in N_P$ is in S, then the corresponding delivery node $2n+i \in N_D$ is also in S, and vice versa.

The set S can naturally be the set of nodes corresponding to one or several geographically close locations (for example one recycling center and two treatment facilities). This is illustrated in Fig. 8, with pick-up, treatment, and delivery nodes in each their column, and with arrows indicating which nodes correspond to the same skip. There are three cases of possible skip situations that can occur in S:

- Case A: The skip is picked up in S, emptied outside S, and delivered in S. There are $n_A = 3$ such requests in the example.
- Case B: All actions related to the skip (pick-up, treatment, and delivery) are inside S. There is $n_B = 1$ such request in the example.
- Case C: The skip is treated in S, but picked up and delivered outside S. There are $n_C = 2$ such requests in the example.

We refer to n_A, n_B, and n_C to the number of requests of case A, B, and C, respectively. There are seven cases if we consider the different combinations of cases A, B, and C, and we analyze each case below. We derive capacity constraints in the form of (89) to determine the number of times a vehicle must enter S by finding the proper right hand side (γ) for the following inequality:

$$\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq \gamma \qquad S \subset N \setminus \{0, d\} : i \in S \Leftrightarrow 2n+i \in S, |S| \geq 2 \quad (89)$$

(1) Only A: $n_A > 0, n_B = n_C = 0$. The vehicle will have to enter empty the first time to pick up the first two requests. After emptying them out of S, it will then enter again to deliver these, and can immediately pick up the next two. It can continue like this and will thus have to enter once for the first pick-up, and thereafter $\lceil n_A/2 \rceil$ times for the deliveries. Hence, $\gamma = 1 + \lceil n_A/2 \rceil$.

(2) Only B: $n_B > 0, n_A = n_C = 0$. All B-requests can be handled sequentially by a single vehicle (assuming that time allows). The vehicle only has to enter once. Hence $\gamma = 1$.

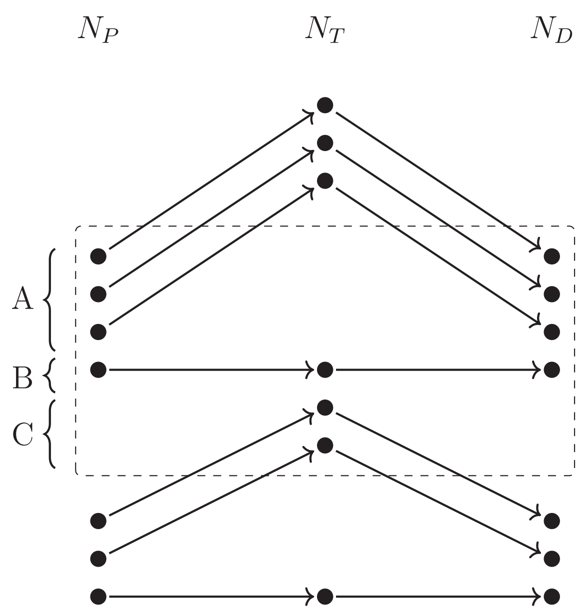

Fig. 8. Illustration the three cases of skips in relation to the set S.

(3) Only C, $n_C > 0, n_A = n_B = 0$. For the C-requests, vehicles enter full, with two full skips, and leave full, with two empty skips, so $\gamma = \lceil n_C/2 \rceil$.

(4) A and B: $n_A > 0, n_B > 0, n_C = 0$. This case is similar to the case with only A, because when the vehicle enters the first time, it can handle the B-requests before picking up the first A-requests. On the other hand, the B-requests cannot decrease the number of entrances needed to handle the A-requests. Hence, $\gamma = 1 + \lceil n_A/2 \rceil$.

(5) A and C: $n_A > 0, n_B = 0, n_C > 0$. We first note, that the trivial lower bound on the number of times we need to enter S with load is $\lceil \frac{n_A+n_C}{2} \rceil$. We are interested in identifying situations in which we can lift this lower bound by one unit.

(i) Consider first the situation where $n_A + n_C$ is odd and $n_C > n_A$. Suppose that the vehicle enters the first time with a full skip of a C-request and pick up an A-request. After leaving S, empty the A-request, leave the C-request and take another C-request. Enter again and deliver the A-request, pick up another and empty the C-request. By repeating this process, after $n_A + 1$ entries in S, all A-requests and $n_A + 1$ C-requests have been handled. The remaining number of C-requests, $n_C - (n_A + 1)$, are even because $n_A + n_C$ is odd. Thus, the total number of entries in S is $n_A + 1 + \frac{n_C-(n_A+1)}{2} = \frac{n_A+n_C+1}{2} = \lceil \frac{n_A+n_C}{2} \rceil$ and it is not possible to lift the trivial lower bound.

(ii) Let us see that one unit is added to the trivial lower bound in the remaining situations.

(ii.1) Consider now the situation where $n_A + n_C$ is odd and $n_A > n_C$. Following an argument similar to (i), after having been in S n_C times, all C-requests and $n_C - 1$ A-requests have been handled. The number of the remaining A-requests is $n_A - (n_C - 1)$, an even number because $n_A + n_C$ is odd. Thus, the total number of entries in S is $n_C + 1 + \frac{n_A-(n_C-1)}{2} = 1 + \frac{n_A+n_C+1}{2} = 1 + \lceil \frac{n_A+n_C}{2} \rceil$. Suppose now that in the last entry in S of the sequence described in the previous paragraph the vehicle does not carry the last C-request, i.e. it only enters with an empty A-request. Then, by delivering this A-request, it can pick up two full A-requests. Thus, as in (1), we can handle all the remaining A-requests, but we need an additional entry for the last C-request, so the total number of entries is again equal to the lower bound plus one.

(ii.2) Finally, consider the situation where both $n_A + n_C$ are even. In this case, we can repeat the arguments of (i) or (ii.1), and the only difference is that the number of remaining A-requests or C-requests are odd, which adds one extra unit to the lower bound.

Summing everything up, we obtain

$$\gamma = \begin{cases} \lceil \frac{n_A+n_C}{2} \rceil & \text{if } n_A + n_C \text{ is odd and } n_C > n_A \\ \lceil \frac{n_A+n_C}{2} \rceil + 1 & \text{otherwise} \end{cases}$$

(6) B and C: $n_A = 0, n_B > 0, n_C > 0$. If n_C is even, there is no benefit in combining C-requests with B-requests, and the vehicle will therefore have to enter once for the B-requests and $\lceil n_C/2 \rceil$ for the C-requests. If n_C is odd, we need $\lfloor n_C/2 \rfloor$ entrances to service all but one C-requests. We then need to enter once more for the final C-request and upon that entrance, the vehicle has an empty slot which can be used to service all B-requests sequentially. Hence, in this case, we need $\lceil n_C/2 \rceil$. Putting this together, we have:

$$\gamma = \begin{cases} 1 + \lceil n_C/2 \rceil & \text{if } n_C \text{ is even} \\ \lceil n_C/2 \rceil & \text{if } n_C \text{ is odd} \end{cases}$$

(7) A, B, and C: $n_A > 0, n_B > 0, n_C > 0$. Because $n_A > 0$, the vehicle will have at least one empty slot upon the first entrance towards servicing A-requests. This slot can be used to service all B-requests without leaving S before picking up the first A-request. Thereby, this case reduces to case 5, and we have:

$$\gamma = \begin{cases} \lceil \frac{n_A+n_C}{2} \rceil & \text{if } n_A + n_C \text{ is odd and } n_C > n_A \\ \lceil \frac{n_A+n_C}{2} \rceil + 1 & \text{otherwise} \end{cases}$$

We summarize the values of γ in Table 3.

Table 3
Summary of the values of γ for the lifted capacity constraint.

Case	In S	γ	
1	$n_A > 0, n_B = n_C = 0$	$1 + \lceil n_A/2 \rceil$	
2	$n_B > 0, n_A = n_C = 0$	1	
3	$n_C > 0, n_A = n_B = 0$	$\lceil n_C/2 \rceil$	
4	$n_A > 0, n_B > 0, n_C = 0$	$1 + \lceil n_A/2 \rceil$	
5	$n_A > 0, n_B = 0, n_C > 0$	$\lceil (n_A + n_C)/2 \rceil$	if $n_A + n_C$ is odd and $n_C > n_A$
		$1 + \lceil (n_A + n_C)/2 \rceil$	otherwise
6	$n_A = 0, n_B > 0, n_C > 0$	$1 + \lceil n_C/2 \rceil$	if n_C is even
		$\lceil n_C/2 \rceil$	if n_C is odd
7	$n_A > 0, n_B > 0, n_C > 0$	$\lceil (n_A + n_C)/2 \rceil$	if $n_A + n_C$ is odd and $n_C > n_A$
		$1 + \lceil (n_A + n_C)/2 \rceil$	otherwise

5. Branch-and-cut algorithm

In this section, we describe our branch-and-cut algorithm and the separation of the valid inequalities. We use the model (20)–(67) presented in Section 3 as our BASE model.

Based on tuning experiments, we initialize our branch-and-cut algorithm by adding the valid inequalities (68)–(71) as well as the symmetry breaking constraints (72)–(74) and the bound on the number of vehicles (75) to our BASE model in the root of the branching tree. In each node of the branching tree, we dynamically add inequalities (76)–(89) to the model when they are violated.

The majority of the valid inequalities can be separated exact in time $O(|N|^2)$ or $O(|N|^3)$ using total enumeration. There are two exceptions to this: The subtour elimination constraints (76) stated in the beginning of Section 4.2, and the lifted capacity inequalities (89) presented in Section 4.2.5 with Table 3 providing the values of γ.

Table 4
Results obtained for the two BASE models using default CPLEX settings and no CPLEX cut generation.

Instance	WL-BASE$_{nc}$				BASE$_{nc}$			
	LB$_{root}$	Time$_{root}$	LB	GAP	LB$_{root}$	Time$_{root}$	LB	GAP
A1	128.5	0.1	661.0	0.0	150.7	0.1	661.0	0.0
A2	104.0	0.2	159.0	75.3	118.0	0.1	168.0	74.0
A3	106.0	0.2	120.0	80.9	106.0	0.1	135.5	78.4
A4	92.0	1.0	96.0	86.7	117.0	0.8	209.0	71.0
A5	126.0	1.0	162.5	77.6	155.0	1.1	216.7	70.1
A6	150.0	2.5	156.5	87.4	156.7	0.9	256.7	79.3
A7	141.0	2.7	161.5	87.1	152.7	0.5	191.3	84.7
A8	214.0	8.1	214.0	84.2	263.5	2.7	313.9	76.9
A9	183.5	7.8	184.0	85.3	208.4	4.5	225.2	82.0
A10	123.0	36.4	131.0	90.0	204.5	22.0	213.5	83.8
A11	148.0	36.8	148.0	89.0	165.0	8.2	176.7	86.8
B1	184.5	20.4	229.0	82.7	217.0	3.2	262.6	80.1
B2	58.0	21.5	61.0	94.7	83.2	9.7	98.3	91.4
C1	224.5	1.3	270.6	80.1	251.9	0.6	332.3	75.6
C2	310.0	1.2	384.0	74.1	357.1	0.4	493.5	66.7
C3	454.0	10.9	553.0	76.3	589.8	4.5	663.4	71.5
C4	228.0	21.0	244.0	88.0	218.3	4.3	275.6	86.5

The subtour elimination (76) are separated heuristically as follows. Based on the current values of the x_{ij}, variables, we first identify connected components. For each of these connected components S, we then check if the solution satisfies inequality (76) and add it to the model if it is not satisfied by the current solution. The separation runs in time $O(n^2)$.

The lifted capacity inequalities (89) are separated heuristically as follows. Remember that multiple requests share the physical location for their pick-up or delivery, and let $\underline{p}$ and $\overline{p}$ be the smallest and largest distance between any two physical locations, respectively. For each physical location $l \in L$, we let $S_\delta(l) \subset N \setminus \{0, d\}$ be the set of nodes for which the associated location is within a radius of δ from l. For each of the three values $\delta \in \{0.001, \frac{9}{10}\underline{p} + \frac{1}{10}\overline{p}, \frac{7}{10}\underline{p} + \frac{3}{10}\overline{p}\}$ and for each sets $S_\delta(l)$, we then check if the solution satisfies inequality (89) and add it to the model if it is not satisfied by the current solution. The separation runs in time $O(n^3)$.

6. Computational experiments

In this section, we present the main results of our computational experiments with the purpose of investigating the strength of the valid inequalities presented in Section 4.2. The implementation is done in C++ in MS Visual Studio Professional 2015 and executed on an Intel Xeon CPU with 12 cores running at 3.5 GHz and 64 GBs RAM, using CPLEX 12.8.

All experiments are performed on a single thread, and we have used one hour computation limit. We initialize all experiments by an upper bound (UB) obtained from Table C.13 in the appendix of Wøhlk and Laporte (2022). No better upper bounds are identified by the models or by our branch-and-cut. All optimality gaps are computed as $GAP = 100\frac{UB-LB}{UB}$, where LB is the best global lower bound obtained after the allowed computation time.

In our experiments, we use the 17 instances from Wøhlk and Laporte (2022) that were used in the exact optimization in that paper. These instances have 5 to 20 requests and include a fixed cost of $\bar{c} = 500$ for each vehicle used. The service time is the same for all pick-ups, i.e. $s_i = s_j\ \forall i, j \in N_P$, and similarly $s_i = s_j\ \forall i, j \in N_T$ and $s_i = s_j\ \forall i, j \in N_D$.

In the first experiment, we focus on the pure effect of the tightening the model presented in Section 3. For this purpose, we run the two models (1)–(19) (WL-BASE$_{nc}$) and (20)–(67) (BASE$_{nc}$) using default CPLEX settings and with CPLEX's automatic cut generation disabled (the nc subscripts indicates this CPLEX setting).

Our results are shown in Table 4, where we show the value of the LP-relaxation (LB_{root}), the time in seconds to obtain it ($Time_{root}$), the best lower bound obtained within one hour of computation (LB), and the optimality gap after one hour (GAP), for each of the two models. For the first instance, both models solve the problem to optimality, with a computation time of 377 and 84.5 s, respectively.

We observe that, except from instance A3, where the root relaxation is the same, our BASE$_{nc}$ model consistently yields better relaxations in roughly 1% of the computation time. Similarly, our model yields dominating results after one hour of computation. Both models solve instance A1 to optimality, and our model does so more than four times faster.

However, knowing that the fixed cost for use of vehicles is 500, it is clear that both models need constraint (75) to provide reasonable lower bounds. We therefore add this constraint to the BASE models in all following experiments (indicated by a + superscript).

In the second experiment, we focus on the effect of adding the valid inequalities presented in this paper. For this, we still use default CPLEX settings and disable CPLEX's automatic cut generation. We compare four models:

1. WL-BASE$^+_{nc}$: Our rerun of the BASE model (1)–(19) of Wøhlk and Laporte (2022) with the inclusion of the vehicle lower bound (75).
2. BASE$^+_{nc}$: Our BASE model (20)–(67) with the inclusion of the vehicle lower bound (75).
3. BASE$^+_{nc}$ wl-cuts: A simplified version of our B&C, which in initiated by (20)–(67) and (68)–(75), and where (B.1) are dynamically added when they are violated. Inequalities (B.1) appear in Appendix B.
4. B&C-nc: Our complete B&C as outlined in Section 5.

Our results are shown in Tables 5 and 6. In Table 5, we first show the instance name and the initial upper bound. Next, the table shows the lower bound value of the root node as well as the time to solve the root node, for each of the four models. In Table 6, we provide the value of the best lower bound as well as the optimality gap for each of the four models after one hour of computation time. In the cases where optimality was proved, we state the computation time in seconds instead. We also provide the average gap over all instances.

We first note, by comparing Tables 5 and 6 to Table 4, that the inclusion of constraint (75) significantly increases the lower bounds both at the root node and after one hour of computation, thereby confirming the importance of this constraint. In fact, our BASE model is now able to solve four instances to optimality within one hour, compared to one instance without this constraint.

The addition of valid inequalities has multiple effects, which are observed by moving towards the right in the two tables: (1) more instances can be solved to optimality. The number is now up to eight. (2) The instances that are solved to optimality are solved faster when all inequalities are used rather than merely the previously known

Table 5
Results for the root node regarding for the four models using default CPLEX settings and no CPLEX cut generation.

Instance	Initial	WL-BASE$^{+}_{nc}$		BASE$^{+}_{nc}$		BASE$^{+}_{nc}$ wl-cuts		B&C-nc	
	UB	LB$_{root}$	Time	LB$_{root}$	Time	LB$_{root}$	Time	LB$_{root}$	Time
A1	661	617.0	0.1	648.5	0.1	653.5	0.1	653.5	0.1
A2	645	579.5	0.2	591.0	0.1	619.6	0.3	619.6	0.3
A3	628	606.0	0.2	606.0	0.1	619.0	0.2	625.3	0.3
A4	720	592.0	1.0	617.0	1.2	679.5	1.1	705.0	2.6
A5	724	623.5	1.1	652.0	0.9	696.0	1.1	710.2	1.9
A6	1240	1150.0	2.7	1155.0	0.9	1224.5	1.7	1228.7	2.6
A7	1248	624.0	2.5	645.9	0.5	672.6	2.0	732.3	4.5
A8	1356	1201.0	7.7	1257.0	5.2	1301.9	4.8	1312.9	6.6
A9	1253	1169.0	7.6	1208.0	9.5	1244.4	5.2	1245.8	6.5
A10	1316	1139.0	37.6	1202.1	14.2	1280.0	26.3	1286.2	43.3
A11	1342	1148.0	35.4	1165.0	6.2	1282.4	16.1	1290.0	40.1
B1	1323	1175.4	21.1	1206.0	3.7	1278.2	8.2	1281.0	11.9
B2	1148	1052.0	21.7	1074.0	4.5	1120.3	10.0	1121.7	15.9
C1	1362	693.5	1.1	719.3	0.5	798.3	1.1	869.3	2.4
C2	1481	1280.8	1.2	1322.7	0.4	1397.8	0.7	1416.1	1.0
C3	2331	1955.0	11.0	2064.0	3.4	2223.6	5.9	2228.3	9.6
C4	2034	1212.7	20.5	1209.0	4.5	1390.6	10.4	1477.8	31.2
Average			10.2		3.3		5.6		10.6

Table 6
Results after one hour of computation time for the four models using default CPLEX settings and no CPLEX cut generation.

Instance	WL-BASE$^{+}_{nc}$			BASE$^{+}_{nc}$			BASE$^{+}_{nc}$ wl-cuts			B&C-nc		
	LB	GAP	Time	LB	GAP	Time	LB	GAP	Time	LB	GAP	Time
A1	661.0	–	2.8	661.0	–	0.4	661.0	–	0.4	661.0	–	0.1
A2	645.0	–	2965.5	645.0	–	99.4	645.0	–	19.4	645.0	–	17.6
A3	628.0	–	1632.5	628.0	–	259.8	628.0	–	41.7	628.0	–	8.6
A4	596.0	17.2	–	707.3	1.8	–	720.0	–	770.1	720.0	–	621.6
A5	650.0	10.2	–	715.0	1.2	–	724.0	–	3407.4	724.0	–	48.9
A6	1152.0	7.1	–	1234.2	0.5	–	1240.0	–	243.2	1240.0	–	73.1
A7	659.0	47.2	–	678.0	45.7	–	717.2	42.5	–	737.0	40.9	–
A8	1214.0	10.5	–	1306.1	3.7	–	1331.5	1.8	–	1332.7	1.7	–
A9	1184.6	5.5	–	1213.0	3.2	–	1253.0	–	3286.1	1253.0	–	930.4
A10	1149.8	12.6	–	1227.0	6.8	–	1284.0	2.4	–	1288.0	2.1	–
A11	1148.0	14.5	–	1186.0	11.6	–	1284.5	4.3	–	1294.5	3.5	–
B1	1225.7	7.4	–	1256.1	5.1	–	1286.7	2.7	–	1287.9	2.7	–
B2	1053.0	8.3	–	1093.0	4.8	–	1122.8	2.2	–	1125.3	2.0	–
C1	739.5	45.7	–	825.8	39.4	–	852.3	37.4	–	894.3	34.3	–
C2	1356.0	8.4	–	1481.0	–	1711.9	1481.0	–	192.4	1481.0	–	121.4
C3	2063.7	11.5	–	2181.8	6.4	–	2274.9	2.4	–	2271.4	2.6	–
C4	1244.0	38.8	–	1292.8	36.4	–	1420.2	30.2	–	1485.5	27.0	–
		14.4			9.8			7.4			6.9	

ones. (3) For all instances that are not solved to optimality, the lower bounds strictly increase (hence, the gap strictly decreases) as more valid inequalities are used, both in the root node and after one hour. The only exceptions are instance A1, where the root-relaxation is not improved[2] and instance B1 after one hour of computation. (4) The run time for solving the root node neither consistently decreases or increases with the addition of the valid inequalities, but the average time increases.

In the third experiment, we compare our B&C algorithm as stated in Section 5 to two previous works:

1. The model from Wøhlk and Laporte (2022) (WL), which corresponds to (1)–(19), (68)–(75), and (B.1) with all inequalities given to CPLEX directly.
2. A preliminary version of our B&C (B&C-2022), which was published in Cubillos (2022), and which roughly corresponds to our algorithm except from the following.
    (a) In that version, the substitution of constraints (12)–(14) with (30)–(62) as described in Appendix A had not been performed, but constraints (A.6)–(A.17) were included as cutting planes.
    (b) In that version, constraints (86), (87), and (89), were not included.
    (c) In that version, constraints (B.1)–(B.19) were included.

Note that we have rerun the results of both studies due to the corrected constraint (75).

In all three cases, we allow CPLEX to generate cutting planes based on default settings. The algorithms are run with a time limit of one hour using a single thread. Bases on tuning using four instance, we use the following CPLEX parameter settings: strong branching (*VarSel=3*), emphasize optimality over feasibility (*MIPEmphasis=2*), and use a moderate probing level (*probe=2*) in our B&C. All other CPLEX parameters are at their default values. For the other two algorithms, we use the CPLEX settings used in those studies.

Our results are shown in Table 7. For each of the three algorithms, we provide the lower bound value of the root node and the lower bound and optimality gap after one hour of computation time. Again, we provide computation time instead of optimality gap when optimality is proved. We also provide the average gap.

We observe that the five instances that are solved to optimality by the other two algorithms are solved to optimality faster by our B&C, in some cases with significant speedup, and our B&C can solve three further instances to optimality.

[2] For instance A2, the improvement is not seen due to rounding. With three digits, we have LB$_{root}$(BASE$^{+}_{nc}$ wl-cuts) = 619.568 and LB$_{root}$(B&C-nc) = 619.621.

Table 7
Comparison our B&C to previous studies.

Instance	WL rerun				B&C-2022 rerun				B&C			
	LB_{root}	LB	GAP	Time	LB_{root}	LB	GAP	Time	LB_{root}	LB	GAP	Time
A1	654.4	661.0	–	1.2	657.7	661.0	–	1.1	653.5	661.0	–	0.1
A2	619.1	645.0	–	100.6	605.0	645.0	–	76.7	623.1	645.0	–	41.9
A3	619.0	628.0	–	87.9	613.3	628.0	–	205.7	625.4	628.0	–	2.1
A4	681.4	712.9	1.0	–	670.9	706.4	1.9	–	705.0	720.0	–	1085.9
A5	694.0	712.1	1.6	–	672.8	717.0	1.0	–	712.6	724.0	–	134.4
A6	1227.0	1240.0	–	536.1	1212.0	1240.0	–	1177.7	1229.1	1240.0	–	79.6
A7	673.8	702.0	43.7	–	657.6	676.8	45.8	–	733.3	743.5	40.4	–
A8	1289.5	1307.9	3.5	–	1269.8	1298.8	4.2	–	1313.7	1340.9	1.1	–
A9	1231.7	1245.8	0.6	–	1211.1	1214.4	3.1	–	1249.4	1253.0	–	298.9
A10	1283.0	1283.0	2.5	–	1224.0	1231.5	6.4	–	1287.7	1290.0	2.0	–
A11	1258.0	1260.8	6.1	–	1196.3	1197.0	10.8	–	1291.1	1305.1	2.7	–
B1	1263.7	1270.1	4.0	–	1248.0	1256.9	5.0	–	1283.5	1295.5	2.1	–
B2	1108.6	1113.8	3.0	–	1098.7	1105.8	3.7	–	1122.2	1129.5	1.6	–
C1	802.8	842.4	38.1	–	778.7	816.0	40.1	–	869.8	899.6	33.9	–
C2	1413.3	1481.0	–	1712.1	1422.4	1481.0	–	318.5	1421.2	1481.0	–	291.6
C3	2185.6	2222.3	4.7	–	2180.0	2214.9	5.0	–	2241.1	2293.8	1.6	–
C4	1364.9	1382.4	32.0	–	1330.4	1358.8	33.2	–	1482.6	1493.0	26.6	–
Average			8.3				9.4				6.6	

Table 8
Number of valid inequalities added by our B&C.

Inst.	(76)	(77)–(79)	(80)–(81)	(82)–(85)	(86)–(87)	(88)	(89)
A1	0	3	1	2	2	1	1
A2	1	3	0	1	6	64	4
A3	0	0	0	2	0	7	3
A4	4	21	0	6	12	212	4
A5	1	6	1	1	11	43	9
A6	1	5	0	0	4	10	4
A7	16	51	2	19	43	180	12
A8	0	17	1	6	11	212	10
A9	1	3	0	2	4	76	4
A10	0	0	0	2	0	249	5
A11	1	4	0	6	8	142	8
B1	0	17	0	1	14	121	3
B2	2	14	3	6	5	151	2
C1	7	48	0	13	41	298	7
C2	1	20	0	0	14	38	11
C3	0	19	1	2	16	80	7
C4	1	39	0	0	11	67	6
Sum	36	270	9	69	202	1951	100
Time per cut	140 393	133	687	602	23	1160	11 858

Of the 17 instances, there were nine that our B&C could not solve to optimality within the allowed computation time. For these instances, we obtain an average gap of 12.4%. The corresponding average gaps of the other two algorithms for these instances are 15.3% and 17.1%, respectively, showing the superiority of our algorithm. Except from two instances, the value of to root relaxation is also higher for our algorithm.

Finally, Table 8 presents the number of valid inequalities from each class added by our B&C. The last line of the table shows time per cut, measured as the total time (in milliseconds) spend searching for each class of cuts divided by the number of violated constraints found. We observe that, as in other capacity routing problems, most of the violated constraints encountered are capacity constraints (88) and (89). Moreover, they are the only ones found in all instances, although for constraints (77)–(79), (82)–(85) and (86)–(87) they are not found in two or three instances. Note that very few subtour elimination constraints (76) are found, indicating that the graph induced by the solution is connected.

7. Conclusions

In this work, we present a branch-and-cut algorithm for a skip pick-up and delivery problem, based in a real-life problem in which full skips are transported from waste drop-off stations to treatment facilities where they are emptied, and then brought back to the original drop-off station. We analyze and strengthen the restrictions of the integer model for the problem, proposed by Wøhlk and Laporte (2022), and propose new valid inequalities for it. Computational results show that our model provides better lower bounds with lower CPU time. We are able to optimally solve 3 more instances with significant run time improvements and reduce the gap of unsolved instances from 15.3% to 12.4%.

Future research directions include the search for new valid inequalities that allow more instances to be solved optimally and also the study of the case where flexible return locations, depending of the kind of skip, are considered.

CRediT authorship contribution statement

José M. Belenguer: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Conceptualization. **Maximiliano Cubillos:** Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Conceptualization. **Sanne Wøhlk:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Conceptualization.

Data availability

The data is freely available.

Acknowledgments

This work is partially supported by the Spanish Ministry of Science and Innovation in the framework of the project "OPRES-Realistic Optimisation in Problems in Public Health" (No. PID2021-124975OB-I00), partially financed by FEDER, Spain funds. Thanks are due to the referee for the valuable comments.

Appendix A. Derivation of the base model

In this section, we explain how the model in Section 3 is derived from the model in Section 2.1, In particular, we justify the replacement of constraints (12)–(14) by constraints (30)–(62) and show that the new model is tighter. For ease of reading, we repeat constraints (12)–(14) here:

$$Q_j^h \geq Q_i^h + q_j^h - (1 - x_{ij}) \quad (i,j) \in A \setminus \{(0,d)\}, h \in H \tag{A.1}$$

$$Q_j^h \leq Q_i^h + q_j^h + (1 - x_{ij}) \quad (i,j) \in A \setminus \{(0,d)\}, h \in H \tag{A.2}$$

$$0 \leq \sum_{h \in H} Q_i^h \leq 2 \quad i \in N \setminus \{0,d\} \tag{A.3}$$

First, we lift constraints (A.3) by improving the upper or lower bound on the number of skips on the vehicle by considering two cases. If node $i \in N_P \cup N_T$, there must be at least one skip on the vehicle since $Q_i^i = 1$, i.e. we have that the lower bound of 1. If $i \in N_D$, the upper bound is 1 because after unloading a skip at $i \in N_D$, the vehicle can carry at most one. Thereby, (A.3) can be replaced by the combination of the following tighter constraints:

$$1 \leq \sum_{h \in H} Q_i^h \leq 2 \qquad i \in N_P \cup N_T \tag{A.4}$$

$$0 \leq \sum_{h \in H} Q_i^h \leq 1 \qquad i \in N_D \tag{A.5}$$

Second, in (A.6)–(A.17), we present valid inequalities for model in Section 2 that arise from how the maximum value of the sum of the Q_i^h variables must be adapted to the capacity of the vehicle in cases when x_{ij} equals 1. For example, in (A.9) we consider an arc from a delivery node to the dummy depot. If $x_{2n+i,d} = 1$ the vehicle must be empty when it leaves node $2n+i$, i.e. $\sum_{h \in H} Q_{2n+i}^h = 0$. On the other hand, if $x_{2n+i,d} = 0$, it follows directly from (A.5) that $\sum_{h \in H} Q_{2n+i}^h$ can be at most 1, and constraints (A.9) thus follows. The validity of the remaining of constraints (A.6)–(A.17) can be argued similarly.

$$\sum_{2n+i \in N_D, i \neq j} x_{2n+i,n+j} + \sum_{h \in H} Q_{n+j}^h \leq 2 \qquad n+j \in N_T \tag{A.6}$$

$$\sum_{2n+i \in N_D, i \neq j} x_{2n+i,2n+j} + \sum_{h \in H} Q_{2n+j}^h \leq 1 \qquad 2n+j \in N_D \tag{A.7}$$

$$\sum_{j \in N_P, j \neq i} x_{i,j} + \sum_{h \in H} Q_i^h \leq 2 \qquad i \in N_P \tag{A.8}$$

$$x_{2n+i,d} + \sum_{h \in H} Q_{2n+i}^h \leq 1 \qquad 2n+i \in N_D \tag{A.9}$$

$$x_{0,i} + \sum_{h \in H} Q_i^h \leq 2 \qquad i \in N_P \tag{A.10}$$

$$\sum_{j \in N_P, j \neq i} x_{n+i,j} + \sum_{h \in H} Q_{n+i}^h \leq 2 \qquad n+i \in N_T \tag{A.11}$$

$$1 + \sum_{i \in N_P, i \neq j} x_{i,j} + \sum_{n+i \in N_T, i \neq j} x_{n+i,j} \leq \sum_{h \in H} Q_j^h \qquad j \in N_P \tag{A.12}$$

$$1 + \sum_{i \in N_P, i \neq j} x_{i,n+j} + \sum_{n+i \in N_T, i \neq j} x_{n+i,n+j} \leq \sum_{h \in H} Q_{n+j}^h \qquad n+j \in N_T \tag{A.13}$$

$$\sum_{i \in N_P, i \neq j} x_{i,2n+j} + \sum_{n+i \in N_T, i \neq j} x_{n+i,2n+j} \leq \sum_{h \in H} Q_{2n+j}^h \qquad 2n+j \in N_D \tag{A.14}$$

$$1 + \sum_{n+j \in N_T, j \neq i} x_{i,n+j} + \sum_{2n+j \in N_D, j \neq i} x_{i,2n+j} \leq \sum_{h \in H} Q_i^h \qquad i \in N_P \tag{A.15}$$

$$1 + \sum_{n+j \in N_T, j \neq i} x_{n+i,n+j} + \sum_{2n+j \in N_D, j \neq i} x_{n+i,2n+j} \leq \sum_{h \in H} Q_{n+i}^h \quad n+i \in N_T \tag{A.16}$$

$$\sum_{n+j \in N_T, i \neq j} x_{2n+i,n+j} + \sum_{2n+j \in N_D, i \neq j} x_{2n+i,2n+j} \leq \sum_{h \in H} Q_{2n+i}^h \quad 2n+i \in N_D \tag{A.17}$$

Note that constraints (A.6)–(A.8) jointly dominate (A.4)–(A.5), and thereby (A.3).

Third, we perform a case analysis on constraints (A.1) and (A.2), where we consider all combinations of i and j in terms of the sets $\{0\}, N_P, N_T, N_D$, and $\{d\}$ and in terms of five situations for the value of $h : i, j, n+i, n+j$, and $h \in H \setminus \{i, n+i, j, n+j\}$. In each case, we consider what happens when $x_{ij} = 1$, which is the situation when the constraints are relevant. Based on the values of the q_i^h-parameters in each case, and based on the bounds on the Q_i^h-variables, it can be shown that either (A.1) or (A.2) (or both) are trivially satisfied in many cases. Furthermore, some cases are not relevant because x_{ij} is fixed at zero. Thereby constraints (A.1)–(A.2) can be replaced by constraints ((A.18))–((A.30)), which constitute fewer constraints in total. For sake of compactness, we define the following notation: For any $i, j \in N_P \cup N_T$, we define $H_{ij}^- = H \setminus \{i, n+i, j, n+j\}$.

$$x_{0j} + Q_j^h \leq 1 \quad j \in N_P, h \in H \setminus \{j, n+j\} \tag{A.18a}$$

$$x_{i,n+i} - 1 \leq Q_{n+i}^h - Q_i^h \leq 1 - x_{i,n+i} i \quad \in N_P, h \in H \setminus \{i, n+i\} \tag{A.19a}$$

$$x_{ij} + Q_i^j \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.20a}$$

$$x_{ij} \leq Q_j^i \qquad i,j \in N_P, i \neq j \tag{A.20b}$$

$$x_{ij} + Q_i^{n+j} \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.20c}$$

$$x_{ij} + Q_j^{n+i} \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.20d}$$

$$x_{ij} - 1 \leq Q_j^h - Q_i^h \leq 1 - x_{ij} \qquad i,j \in N_P, i \neq j, h \in H_{ij}^- \tag{A.20e}$$

$$x_{i,n+j} \leq Q_{n+j}^i \qquad i,j \in N_P, i \neq j \tag{A.21a}$$

$$x_{i,n+j} \leq Q_i^j \qquad i,j \in N_P, i \neq j \tag{A.21b}$$

$$x_{i,n+j} + Q_{n+j}^{n+i} \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.21c}$$

$$x_{i,n+j} + Q_i^{n+j} \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.21d}$$

$$x_{i,n+j} - 1 \leq Q_{n+j}^h - Q_i^h \leq 1 - x_{i,n+j} \qquad i,j \in N_P, i \neq j, h \in H_{ij}^- \tag{A.21e}$$

$$x_{i,2n+j} \leq Q_{2n+j}^i \qquad i,j \in N_P, i \neq j \tag{A.22a}$$

$$x_{i,2n+j} + Q_i^j \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.22b}$$

$$x_{i,2n+j} + Q_{2n+j}^{n+i} \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.22c}$$

$$x_{i,2n+j} \leq Q_i^{n+j} \qquad i,j \in N_P, i \neq j \tag{A.22d}$$

$$x_{i,2n+j} - 1 \leq Q_{2n+j}^h - Q_i^h \leq 1 - x_{i,2n+j} \qquad i,j \in N_P, i \neq j, h \in H_{ij}^- \tag{A.22e}$$

$$x_{n+i,2n+i} - 1 \leq +Q_{2n+i}^h - Q_{n+i}^h \leq 1 - x_{n+i,2n+i} i \in N_P, h \in H \setminus \{i, n+i\} \tag{A.23a}$$

$$x_{n+i,j} + Q_j^i \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.24a}$$

$$x_{n+i,j} + Q_{n+i}^j \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.24b}$$

$$x_{n+i,j} \leq Q_j^{n+i} \qquad i,j \in N_P, i \neq j \tag{A.24c}$$

$$x_{n+i,j} + Q_{n+i}^{n+j} \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.24d}$$

$$x_{n+i,j} - 1 \leq Q_j^h - Q_{n+i}^h \leq 1 - x_{n+i,j} \qquad i,j \in N_P, i \neq j, h \in H_{ij}^- \tag{A.24e}$$

$$x_{n+i,n+j} + Q_{n+j}^i \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.25a}$$

$$x_{n+i,n+j} \leq Q_{n+i}^j \qquad i,j \in N_P, i \neq j \tag{A.25b}$$

$$x_{n+i,n+j} \leq Q_{n+j}^{n+i} \qquad i,j \in N_P, i \neq j \tag{A.25c}$$

$$x_{n+i,n+j} + Q_{n+i}^{n+j} \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.25d}$$

$$x_{n+i,n+j} - 1 \leq Q_{n+j}^h - Q_{n+i}^h \leq 1 - x_{n+i,n+j} \qquad i,j \in N_P, i \neq j, h \in H_{ij}^- \tag{A.25e}$$

$$x_{n+i,2n+j} + Q_{2n+j}^i \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.26a}$$

$$x_{n+i,2n+j} + Q^j_{n+i} \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.26b}$$

$$x_{n+i,2n+j} \leq Q^{n+i}_{2n+j} \qquad i,j \in N_P, i \neq j \tag{A.26c}$$

$$x_{n+i,2n+j} \leq Q^{n+j}_{n+i} \qquad i,j \in N_P, i \neq j \tag{A.26d}$$

$$x_{n+i,2n+j} - 1 \leq Q^h_{2n+j} - Q^h_{n+i} \leq 1 - x_{n+i,2n+j} \qquad i,j \in N_P, i \neq j, h \in H^-_{ij} \tag{A.26e}$$

$$x_{2n+i,j} + Q^i_j \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.27a}$$

$$x_{2n+i,j} + Q^j_{2n+i} \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.27b}$$

$$x_{2n+i,j} + Q^{n+i}_j \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.27c}$$

$$x_{2n+i,j} + Q^{n+j}_{2n+i} \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.27d}$$

$$x_{2n+i,j} - 1 \leq Q^h_j - Q^h_{2n+i} \leq 1 - x_{2n+i,j} \qquad i,j \in N_P, i \neq j, h \in H^-_{ij} \tag{A.27e}$$

$$x_{2n+i,n+j} + Q^i_{n+j} \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.28a}$$

$$x_{2n+i,n+j} \leq Q^j_{2n+i} \qquad i,j \in N_P, i \neq j \tag{A.28b}$$

$$x_{2n+i,n+j} + Q^{n+i}_{n+j} \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.28c}$$

$$x_{2n+i,n+j} + Q^{n+j}_{2n+i} \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.28d}$$

$$x_{2n+i,n+j} - 1 \leq Q^h_{n+j} - Q^h_{2n+i} \leq 1 - x_{2n+i,n+j} \qquad i,j \in N_P, i \neq j, h \in H^-_{ij} \tag{A.28e}$$

$$x_{2n+i,2n+j} + Q^i_{2n+j} \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.29a}$$

$$x_{2n+i,2n+j} + Q^j_{2n+i} \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.29b}$$

$$x_{2n+i,2n+j} + Q^{n+i}_{2n+j} \leq 1 \qquad i,j \in N_P, i \neq j \tag{A.29c}$$

$$x_{2n+i,2n+j} \leq Q^{n+j}_{2n+i} \qquad i,j \in N_P, i \neq j \tag{A.29d}$$

$$x_{2n+i,2n+j} - 1 \leq Q^h_{2n+j} - Q^h_{2n+i} \leq 1 - x_{2n+i,2n+j} \qquad i,j \in N_P, i \neq j, h \in H^-_{ij} \tag{A.29e}$$

$$x_{2n+i,d} + Q^h_{2n+i} \leq 1 \qquad i \in N_P, h \in H \setminus \{i, n+i\} \tag{A.30a}$$

At this point, constraints (A.1)–(A.3) are replaced by (A.6)–(A.8) and ((A.18))–((A.30)), while constraints (A.9)–(A.17) serve as valid inequalities.

Fourth, we can show that some of the constraints ((A.18))–((A.30)) are trivially satisfied under the condition that certain others of these constraints are preserved in the model. We find that

- Constraints (A.20d) are satisfied if constraints (A.20b) are in the model.
- Constraints (A.21c) are satisfied if constraints (A.21a) are in the model.
- Constraints (A.21d) are satisfied if constraints (A.21b) are in the model.
- Constraints (A.22b) are satisfied if constraints (A.22d) are in the model.
- Constraints (A.22c) are satisfied if constraints (A.22a) are in the model.
- Constraints (A.24a) are satisfied if constraints (A.24c) are in the model.
- Constraints (A.25a) are satisfied if constraints (A.25c) are in the model.
- Constraints (A.25d) are satisfied if constraints (A.25b) are in the model.
- Constraints (A.26a) are satisfied if constraints (A.26c) are in the model.
- Constraints (A.26b) are satisfied if constraints (A.26d) are in the model.
- Constraints (A.28d) are satisfied if constraints (A.28b) are in the model.
- Constraints (A.29b) are satisfied if constraints (A.29d) are in the model.

We can therefore remove the first-mentioned constraints from the model provided that we keep the last-mentioned constraints.

Fifth, we investigate which of the constraints ((A.18))–((A.30)) are dominated by some of the constraints (A.6)–(A.8) or (A.9)–(A.17), and can thus be discarded. We find that

- Constraints (A.18a) are dominated by constraints (A.10).
- Constraints (A.20a) are dominated by constraints (A.8).
- Constraints (A.20c) are dominated by constraints (A.8).
- Constraints (A.24b) are dominated by constraints (A.11).
- Constraints (A.24d) are dominated by constraints (A.11).
- Constraints (A.28a) are dominated by constraints (A.6).
- Constraints (A.28c) are dominated by constraints (A.6).
- Constraints (A.29a) are dominated by constraints (A.7).
- Constraints (A.29c) are dominated by constraints (A.7).
- Constraints (A.30a) are dominated by constraints (A.9).

We can therefore remove these constraints provided that we add (A.9)–(A.11) to the model. Note that (A.6)–(A.8) are already in the model.

Sixth, we reverse the analysis, and investigate which of the constraints (A.6)–(A.17) are redundant in the presence of some of the remaining constraints from ((A.18))–((A.30)). To illustrate this, note that by adding constraints (A.20b) for all $i \in N_P$ and (A.24c) for all $n+i \in N_T$, and by noting that $Q^j_j = 1$ and $Q^{n+j}_j = 0$, we obtain constraints (A.12). Therefore, constraints (A.12) are redundant in the presence of (A.20b) and (A.24c). Using this procedure, we find that

- Constraints (A.12) are redundant in the presence of (A.20b) and (A.24c).
- Constraints (A.13) are redundant in the presence of (A.21a) and (A.25c).
- Constraints (A.14) are redundant in the presence of (A.22a) and (A.26c).
- Constraints (A.15) are redundant in the presence of (A.21b) and (A.22d).
- Constraints (A.16) are redundant in the presence of (A.25b) and (A.26d).
- Constraints (A.17) are redundant in the presence of (A.28b) and (A.29d).

To conclude, constraints (A.1)–(A.3), which corresponds to constraints (12)–(14) in the model in Section 2.1 can be substituted by the following stronger constraints:

- (A.6)–(A.8)
- (A.9)–(A.11)
- (A.19a), (A.20b), (A.20e), (A.21a), (A.21b), and (A.21e)
- (A.22a), (A.22d), (A.22e), and (A.23a)

- (A.24c), (A.24e), (A.25b), (A.25c), and (A.25e)
- (A.26c), (A.26d), (A.26e), and (A.27a)–(A.27e)
- (A.28b), (A.28e), (A.29d) and (A.29e)

These constraints constitute constraints (30)–(62) in the model in Section 3.

Appendix B. Proof that some constraints are redundant

For completeness, we state in this appendix two classes of valid inequalities previously proposed for this problem, which we conjecture to be redundant with our improved model, and we prove redundancy for one of the inequalities.

The following class of inequalities are proposed by Wøhlk and Laporte (2022) for the original model. They are based on the possible actions of a vehicle before and after visiting a given node $i \in N$. For example, the first family states that it is only possible to pick-up two skips in a sequence, considering that the capacity of the vehicles is two skips, as well as the fact that it is not possible to deliver an empty skip immediately after picking up two skips. The remaining families of inequalities follow the a similar logic considering the cases in which $i \in N_T$ and $i \in N_D$.

$$\begin{array}{ll} \sum_{l\in N_P, l\neq i} x_{li} + \sum_{j\in N_P\cup N_D, j\neq i} x_{ij} \leq 1 & i \in N_P \\ \sum_{l\in N_P\cup N_T, l\neq i} x_{li} + \sum_{j\in N_P, j\neq i} x_{ij} \leq 1 & i \in N_P \\ \sum_{l\in N_T, l\neq i} x_{li} + \sum_{j\in N_P\cup N_T, j\neq i} x_{ij} \leq 1 & i \in N_T \\ \sum_{l\in N_T\cup N_D, l\neq i} x_{li} + \sum_{j\in N_T, j\neq i} x_{ij} \leq 1 & i \in N_T \\ \sum_{l\in N_D, l\neq i} x_{li} + \sum_{j\in N_T\cup N_D, j\neq i} x_{ij} \leq 1 & i \in N_D \\ \sum_{l\in N_D\cup N_P, l\neq i} x_{li} + \sum_{j\in N_D, j\neq i} x_{ij} \leq 1 & i \in N_D. \end{array} \tag{B.1}$$

The following class of inequalities are proposed by Cubillos (2022) for the original model. They are referred to as *Precedence and successor inequalities* and are based on investigating which arcs can be used to leave a node j given that a specific arc is used to enter node j, i.e., they answer the question: given that $x_{ij} = 1$, which other variables x_{jk} must then necessarily also be 1? In (B.2)–(B.10) we present the arcs that can occur after the arc on the left hand side of the inequality if the left hand side is equals 1 in the solution. For instance, in (B.2) we consider the case in which the variable $x_{0j} = 1, j \in N_P$, which corresponds to a first pick-up in a route. Right after a first pick-up, a vehicle can either travel to its treatment location at $n + j$, or travel to any other pick-up $k \in N_P\backslash\{j\}$. The same logic follows for the remaining inequalities.

However, in some cases, the constraints can be tightened by replacing the x_{ij}-variable on the left hand side by Q^i_j. Consider for instance a version of constraints (B.3) with the left hand side replaced by x_{ij}. That constraint states that if a vehicle picks up skip i and immediately after picks up skip j (after which the two full skips are on the vehicle), then the only possible next action is to empty one of them, which is expressed in the right hand side of the constraint. By replacing x_{ij} by Q^i_j as in (B.3), the constraint states that if a vehicle is in the pick-up node for skip j and carries the full skip i, then these are the next two possible actions. However, the vehicle can reach this state by coming directly from node i as above or by having performed other tasks while carrying i, and eventually arrive at node j. Therefore the constraint becomes stronger.

$$x_{0,j} \leq x_{j,n+j} + \sum_{k\in N_P\backslash\{j\}} x_{j,k} \qquad \forall j \in N_P, \tag{B.2}$$

$$Q^i_j \leq x_{j,n+i} + x_{j,n+j} \qquad \forall i, j \in N_P, i \neq j, \tag{B.3}$$

$$Q^i_{n+j} \leq x_{n+j,n+i} + x_{n+j,2n+j} \qquad \forall i \in N_P, \forall n+j \in N_T, i \neq j, \tag{B.4}$$

$$Q^i_{2n+j} \leq x_{2n+j,n+i} + \sum_{k\in N_P\backslash\{i,j\}} x_{2n+j,k} \qquad \forall i \in N_P, \forall 2n+j \in N_D, i \neq j, \tag{B.5}$$

$$Q^{n+i}_j \leq x_{j,2n+i} + x_{j,n+j} \qquad \forall j \in N_P, \forall n+i \in N_T, i \neq j, \tag{B.6}$$

$$Q^{n+i}_{n+j} \leq x_{n+j,2n+i} + x_{n+j,2n+j} \qquad \forall n+i, n+j \in N_T, i \neq j, \tag{B.7}$$

$$Q^{n+i}_{2n+j} \leq x_{2n+j,2n+i} + \sum_{k\in N_P\backslash\{i,j\}} x_{2n+j,k} \qquad \forall n+i \in N_T, \forall 2n+j \in N_D, i \neq j, \tag{B.8}$$

$$x_{2n+i,n+j} \leq x_{n+j,2n+j} + \sum_{k\in N_P\backslash\{i,j\}} x_{n+j,k} \qquad \forall 2n+i \in N_D, \forall n+j \in N_T, i \neq j, \tag{B.9}$$

$$x_{2n+i,2n+j} \leq x_{2n+j,d} + \sum_{k\in N_P\backslash\{i,j\}} x_{2n+j,k} \qquad \forall 2n+i, 2n+j \in N_D, i \neq j. \tag{B.10}$$

There are three additional cases, namely $x_{i,n+i}$, $x_{n+i,2n+i}$, and $x_{2n+i,j}$. However, in these cases, the flexibility of the problem means that we cannot derive similar constraints. Using the same argument, we can find valid inequalities for travels that can occur right before a specific travel. (B.11)–(B.19) present valid inequalities for the possible travels that can occur before the arc or state on the left hand side if the left hand side of the inequality equals 1.

$$x_{i,j} \leq x_{0,i} + \sum_{k\in N_P\backslash\{i,j\}} x_{2n+k,i} \qquad \forall i, j \in N_P, i \neq j, \tag{B.11}$$

$$Q^j_i \leq x_{j,i} + \sum_{k\in N_P\backslash\{i,j\}} x_{2n+k,i} \qquad \forall i \in N_P, \forall n+j \in N_T, i \neq j, \tag{B.12}$$

$$Q^{n+j}_i \leq x_{n+j,i} + \sum_{k\in N_P\backslash\{i,j\}} x_{2n+k,i} \qquad \forall i \in N_P, \forall 2n+j \in N_D, i \neq j, \tag{B.13}$$

$$x_{n+i,j} \leq x_{i,n+i} + \sum_{k\in N_P\backslash\{i,j\}} x_{2n+k,n+i} \qquad \forall j \in N_P, \forall n+i \in N_T, i \neq j, \tag{B.14}$$

$$Q^j_{n+i} \leq x_{i,n+i} + x_{j,n+i} \qquad \forall n+i, n+j \in N_T, i \neq j, \tag{B.15}$$

$$Q^{n+j}_{n+i} \leq x_{i,n+i} + x_{n+j,n+i} \qquad \forall n+i \in N_T, \forall 2n+j \in N_D, i \neq j, \tag{B.16}$$

$$Q^j_{2n+i} \leq x_{n+i,2n+i} + x_{j,2n+i} \qquad \forall 2n+i \in N_D, \forall n+j \in N_T, i \neq j, \tag{B.17}$$

$$Q^{n+j}_{2n+i} \leq x_{n+i,2n+i} + x_{n+j,2n+i} \qquad \forall 2n+i, 2n+j \in N_D, i \neq j, \tag{B.18}$$

$$x_{2n+i,d} \leq x_{n+i,2n+i} + \sum_{k\in N_P\backslash\{i\}} x_{2n+k,2n+i} \qquad \forall 2n+i \in N_D. \tag{B.19}$$

We conjecture that with the stronger BASE model presented in Section 3, all above inequalities of this appendix are redundant. We now prove that this is the case for constraints (B.2).

For $i \in N_P$, we have the following, which we aim to prove is $\leq$ zero:

$$\begin{array}{lll} & x_{0,i} - x_{i,n+i} - \sum_{j\in N_P\backslash\{i\}} x_{i,j} & \\ \leq & 2 - \sum_{h\in H} Q^h_i - x_{i,n+i} - \sum_{j\in N_P\backslash\{i\}} x_{i,j} & \text{by (58)} \\ = & 2 - \sum_{h\in H} Q^h_i + \sum_{n+j\in N_T\backslash\{n+i\}} x_{i,n+j} + \sum_{2n+j\in N_D\backslash\{2n+i\}} x_{i,2n+j} - 1 & \text{by (4)} \\ \leq & 0 & \text{by (A.15)} \end{array}$$

Thereby, we have shown that constraints (B.2) are redundant.

References

Archetti, C., Mansini, R., Speranza, M.G., 2005. Complexity and reducibility of the skip delivery problem. Transp. Sci. 39 (2), 182–187. http://dx.doi.org/10.1287/trsc.1030.0084.

Archetti, C., Speranza, M.G., 2005. Collection of waste with single load trucks: A real case. In: Fleischmann, B., Klose, A. (Eds.), Distribution Logistics. Springer-Verlag, Berlin Heidelberg, pp. 105–119. http://dx.doi.org/10.1007/978-3-642-17020-1_6.

Aringhieri, R., Bruglieri, M., Malucelli, F., Nonato, M., 2004. An asymmetric vehicle routing problem arising in the collection and disposal of special waste. Electron. Notes Discrete Math. 17, 41–47. http://dx.doi.org/10.1016/j.endm.2004.03.011.

Baldacci, R., Bodin, L.D., Mingozzi, A., 2006. The multiple disposal facilities and multiple inventory locations rollon-rolloff vehicle routing problem. Comput. Oper. Res. 33 (9), 2667–2702. http://dx.doi.org/10.1016/j.cor.2005.02.023.

Battarra, M., Cordeau, J.-F., Iori, M., 2014. Pickup-and-delivery problems for goods transportation. In: Toth, P., Vigo, D. (Eds.), Vehicle routing: Problems, Methods, and Applications, Second ed. In: MOS-SIAM Series on Optimization, pp. 166–191. http://dx.doi.org/10.1137/1.9781611973594.ch6.

Benjamin, A.M., Beasley, J.E., 2010. Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities. Comput. Oper. Res. 37 (12), 2270–2280. http://dx.doi.org/10.1016/j.cor.2010.03.019.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., Laporte, G., 2007. Static pickup and delivery problems: A classification scheme and survey. TOP 15 (1), 1–31. http://dx.doi.org/10.1007/s11750-007-0009-0.

Bodin, L.D., Mingozzi, A., Baldacci, R., Ball, M.O., 2000. The rollon-rolloff vehicle routing problem. Transp. Sci. 34 (3), 271–288. http://dx.doi.org/10.1287/trsc.34.3.271.12301.

Cordeau, J.-F., 2006. A branch-and-cut algorithm for the dial-a-ride problem. Oper. Res. 54 (3), 573–586. http://dx.doi.org/10.1287/opre.1060.0283.

Cubillos, M., 2022. Optimization and Data Analytics in Waste Management (Ph.D. thesis). School of Business and Social Sciences, Aarhus University, Aarhus, Denmark, URL: https://pure.au.dk/ws/portalfiles/portal/263804916/Maximiliano_Cubillos_PhD_dissertation.pdf.

De Meulemeester, G., Louveaux, F.V., Semet, F.J., 1997. Optimal sequencing of skip collections and deliveries. J. Oper. Res. Soc. 48 (1), 57–64. http://dx.doi.org/10.1057/palgrave.jors.2600325.

Li, H., Jian, X., Chang, X., Lu, Y., 2018. The generalized rollon-rolloff vehicle routing problem and savings-based algorithm. Transp. Res. B 113, 1–23. http://dx.doi.org/10.1016/j.trb.2018.05.005.

Parragh, S.N., Doerner, K.F., Hartl, R.F., 2008. A survey on pickup and delivery problems, part II: Transportation between pickup and delivery locations. J. für Betriebswirtschaft 58, 81–117. http://dx.doi.org/10.1007/s11301-008-0036-4.

Rabbani, M., Tabrizi, F.Haeri., Farrokhi-Asi, H., 2016. A hybrid algorithm for solving a roll-on roll-off waste collection vehicle routing problem considering waste separation and recycling center. Int. J. Appl. Oper. Res. 6 (2), 19–31, URL: https://ijorlu.liau.ac.ir/article-1-514-en.pdf.

Raucq, J., Sörensen, K., Cattrysse, D., 2019. Solving a real-life roll-on–roll-off waste collection problem with column generation. J. Veh. Routing Algorithms 2, 41–54. http://dx.doi.org/10.1007/s41604-019-00013-6.

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. http://dx.doi.org/10.1002/net.20177.

Wøhlk, S., Laporte, G., 2022. Transport of skips between recycling centers and treatment facilities. Comput. Oper. Res. 145, 105879. http://dx.doi.org/10.1016/j.cor.2022.105879.

Wy, J., Kim, B.I., Kim, S., 2013. The rollon-rolloff waste collection problem with time windows. European J. Oper. Res. 224 (3), 466–476. http://dx.doi.org/10.1016/j.ejor.2012.09.001.