Appendix

Appendix A: Proofs

A.1. Proposition 4

For convenience, let $\tau = \tau_{\mathsf{f}}^{\text{late}}$. The proof first requires the following fact.

PROPOSITION A.1. *For $1 \le k \le K$, if there is waiting time in $\tau_{\mathsf{f}}^{\text{late}}(t)$ between i_1 and i_k then there exists $1 \le k^* < k$ such that $\tau_{k^*}(t) = \tau_{k^*}(l_{i_K})$.*

Proof. *We proceed by induction, assuming the statement is true for $k = 1, \dots, k'$. Suppose there is waiting time between 1 and $k'+1$. Without loss of generality assume that there is waiting time between k' and $k'+1$, else we may apply the inductive hypothesis.*

For the sake of contradiction, suppose $\tau_k(t) < \tau_k(l_{i_K})$ for all $k = 1, \dots, k'+1$. Define

$$\epsilon = \min\Big\{\big(\tau_1(l_{i_K}) - \tau_1(t)\big), \dots, \big(\tau_{k'+1}(l_{i_K}) - \tau_{k'+1}(t)\big), \big(\tau_{k'+1}(t) - \tau_{k'}(t) - t_{i_{k'}, i_{k'+1}}\big)\Big\}$$

noting that $\epsilon > 0$ since there is waiting time between k' and $k'+1$. Let

$$\bar{\tau}(t) = \begin{cases} \tau_k(t) + \epsilon & \text{if } 1 \le k \le k' \\ \tau_k(t) & \text{otherwise} \end{cases}$$

which is a valid schedule by construction and services i_1 later than $\tau(t)$. This contradicts the definition of $\tau(t)$ as the latest schedule, and therefore there is some $1 \le k^* \le k'$ where $\tau_{k^*}(t) = \tau_{k^*}(l_{i_K})$.

For the base case, $k = 1$, there can never be waiting time between i_1 and i_1, so the statement is vacuously true. □

Let k^* be the smallest k such that $\tau_k(t) = \tau_k(l_{i_K})$. Since k^* is the smallest such index, there cannot be any waiting time between 1 and k^*, by the above proposition. Thus,

$$T_{\mathsf{f}}(1, k^*) = \tau_{k^*}(t) - \tau_1(t) = \tau_{k^*}(l_{i_K}) - \tau_1(t) \ge \tau_{k^*}(l_{i_K}) - \tau_1(l_{i_K})$$

and for any $t \ge E_{\mathsf{f}}$ it is also true that $\tau_{k^*}(t) - \tau_1(t) \ge T_{\mathsf{f}}(1, k^*)$. Therefore,

$$\tau_{k^*}(l_{i_K}) - \tau_1(t) = \tau_{k^*}(l_{i_K}) - \tau_1(l_{i_K}) = T_{\mathsf{f}}(1, k^*)$$

which implies $\tau_1(t) = \tau_1(l_{i_K}) =: L_{\mathsf{f}}$.

A.2. Proposition 6

If $f(x)$ is constant on $[a, b]$, then $f(x) = f(b) \ge g(b) \ge g(x)$. Otherwise, let $x_f \in (a, b)$ be the point at which the gradient of f changes from 0 to 1, so that

$$f(x) = \max\{x - x_f + f(x_f), f(x_f)\}$$

Likewise define x_g. Suppose $f(x_f) < g(x_f)$, then $g(a) \le f(a) = f(x_f) < g(x_f)$, which implies that $g(x)$ must be increasing on $[x_f, b]$. Consequently, for $x \in [x_f, b]$,

$$g(x) = x - x_f + g(x_f) > x - x_f - f(x_f) = f(x)$$

and in particular $g(b) > f(b)$, a contradiction. Therefore $g(x_f) \le f(x_f)$. Fix $x \in [a, b]$, there are three cases to cover:

Case (i): $x \in [a, x_f]$. Since $x \le x_f$, $g(x) \le g(x_f) \le f(x_f) = f(x)$.

Case (ii): $x \in (x_f, b]$ and $x_g \le x_f$. As $x_g \le x_f$, $g(x)$ must be increasing on $(x_f, b]$, so

$$g(x) = x - x_f + g(x_f) \le x - x_f - f(x_f) = f(x)$$

Case (iii): $x \in (x_f, b]$ and $x_g > x_f$. In this case $g(x)$ is constant at x_f, so $g(x_f) = g(x_g)$ and

$$g(x) = x - x_g + g(x_g) = x - x_g + g(x_f) < x - x_f + f(x_f) = f(x)$$

In all cases, $f(x) \ge g(x)$.

A.3. Proposition 7

If $\tau_{\mathsf{f}}^{\text{late}}(t)$ contains no waiting time, then

$$\tau_{\mathsf{f},1}^{\text{late}}(t) = t - T_{\mathsf{f}} \ge t - T_{\mathsf{g}} \ge \tau_{\mathsf{g},1}^{\text{late}}(t)$$

otherwise by Proposition 4, $\tau_{\mathsf{f},1}^{\text{late}}(t) = L_{\mathsf{f}}$ and we arrive at the same result.

$$\tau_{\mathsf{f},1}^{\text{late}}(t) = L_{\mathsf{f}} \ge L_{\mathsf{g}} \ge \tau_{\mathsf{g},1}^{\text{late}}(t)$$

A.4. Proposition 9

Let (x, y) satisfy (33). Using flow and cover constraints again, the following inequalities are true for any point in the LP hull

$$y_{\mathsf{a}_{l-1}} \le \sum_{\mathsf{f} \in \mathsf{F}^-_{\mathsf{a}_{l-1}}} x_{\mathsf{f}} \le 1$$

Setting $S = \{\mathsf{f}_1, \ldots, \mathsf{f}_{l-1}\} \cup \{\mathsf{a}_1, \ldots, \mathsf{a}_{l-1}\}$ and $Z = \mathsf{F}^-_{\mathsf{a}_{l-1}}$, Lemma 1 gives part 1 and the implication of part 2. To see that the converse is not true, let $(\mathsf{f}_1, \mathsf{f}_2)$ be an illegal chain in a fractional route with value $2/3$, so that $x_{\mathsf{f}_1} = x_{\mathsf{f}_2} = y_{\mathsf{a}_1} = 2/3$. In addition, suppose $x_{\mathsf{f}} = 0$ for all $\mathsf{f} \in \mathsf{F}^-_{\mathsf{a}_1} \setminus \{\mathsf{f}_2\}$. The left-hand side of (32) reads $x_{\mathsf{f}_1} + y_{\mathsf{a}_1} + x_{\mathsf{f}_2} = 2$, so (32) is not violated. On the other hand, the left-hand side of (33) evaluates to $4/3 > 1$, whilst the summation on the right-hand side evaluates to 0, so the inequality is violated.

A.5. Proposition 10

Suppose (38) is violated at an integer solution (x, y). Since there are $2l - 3$ terms on the left-hand side, the inequality may only be violated when the left-hand side is equal to $2l - 3$ and the right-hand side is equal to $2l - 4$. In other words, the chain $(\mathsf{f}_1, \ldots, \mathsf{f}_{l-1})$ is in the solution, and no arcs from $\mathsf{A}^-_{\mathsf{c}}$ are in the solution. Therefore, by flow conservation, an arc from $\mathsf{A}^-_{\mathsf{f}_{l-1}} \setminus \mathsf{A}^-_{\mathsf{c}}$ is in the solution, and so too is some fragment g which starts at $\text{End}(\mathsf{a})$. However, the chain $(\mathsf{f}_1, \ldots, \mathsf{f}_{l-1}, \mathsf{g})$ is illegal by definition of $\mathsf{A}^-_{\mathsf{c}}$, so the solution must be illegal as well. Obviously, c is cut off, since $\mathsf{a}_{l-1} \notin \mathsf{A}^-_{\mathsf{c}}$ by assumption, so the first part of the proposition is true.

The flow and cover constraints (5)–(7) imply that for any fractional (x, y),

$$x_{\mathsf{f}_{l-1}} \le \sum_{\mathsf{a} \in \mathsf{A}^-_{\mathsf{f}_{l-1}}} y_{\mathsf{a}} \le 1$$

Consider the inequality

$$\sum_{k=1}^{l-1} x_{\mathsf{f}_k} + \sum_{k=1}^{l-2} y_{\mathsf{a}_k} + \sum_{\mathsf{a} \in \mathsf{A}^-_{\mathsf{f}_{l-1}} \setminus \mathsf{A}^-_{\mathsf{c}}} y_{\mathsf{a}} \leq 2l - 3$$

and note that since $\mathsf{a}_{l-1} \in \mathsf{A}^-_{\mathsf{f}_{l-1}} \setminus \mathsf{A}^-_{\mathsf{c}}$, this inequality implies (33) when $\mathsf{F}^-_{\mathsf{c}} = \emptyset$. Defining $S = \{\mathsf{f}_1, \ldots \mathsf{f}_{l-1}\} \cup \{\mathsf{a}_1, \ldots, \mathsf{a}_{l-2}\}$ and $Z = \mathsf{A}_{\mathsf{f}_{l-1}}$ in lemma 1 proves that in the LP hull, (38) implies the above inequality, and therefore implies (33) as well.

As a counterexample to the converse, consider $\mathsf{c} = (\mathsf{f}_1, \mathsf{f}_2)$ and suppose $x_{\mathsf{f}_1} = x_{\mathsf{f}_2} = y_{\mathsf{a}_1} = 1/2$, and $y_{\mathsf{a}} = 0$ for all $\mathsf{a} \in A^-_{\mathsf{c}}$. Then the left-hand side of (33) is $x_{\mathsf{f}_1} + y_{\mathsf{a}_1} = 1$ so (33) is not violated, but (38) reads

$$x_{\mathsf{f}_1} \leq \sum_{\mathsf{a} \in A^-_{\mathsf{c}}} y_{\mathsf{a}}$$

which is clearly violated since the right-hand side is 0.

A.6. Proposition 12

Let (x, y) be a legal solution such that $x_{\mathsf{f}} = 1$. If f is the last fragment in a route then $y_{\mathsf{a}} = 1$ where $\mathsf{a} = \big(\mathrm{End}(\mathsf{f}), \mathsf{o}^-\big) \in \mathsf{A}_{(\mathsf{f},\cdot)}$ by conservation of flow. Otherwise, f is followed by a fragment g where (f, g) is legal, since the route is legal. In this case $y_{\mathsf{a}} = 1$ where $\mathsf{a} = \big(\mathrm{End}(\mathsf{f}), \mathrm{Start}(\mathsf{g})\big) \in \mathsf{A}^-_{\mathsf{f}}$, and therefore $\mathsf{g} \in \mathsf{F}_{(\mathsf{f},\cdot)} \cap \mathsf{F}^-_{\mathsf{a}}$ and $\mathsf{a} \in \mathsf{A}_{(\mathsf{f},\cdot)}$.

An almost identical proof may be given for the validity of inequality (44), which is the analogue to (43) considering compatible fragments that may be placed before an arc. Similar analogues exists for the Fragment-Arc and Fragment-Fragment inequalities, and their proofs are likewise omitted.

A.7. Proposition 13

Suppose (x, y) is a legal solution and $x_{\mathsf{f}} = 1$. Since f does not end empty, f cannot be the last fragment in a route, and the result follows from the proof of Proposition 12.

A.8. Proposition 14

Suppose $y_{\mathsf{a}} = 1$ in a legal solution (x, y). Using conservation of flow, there must be $\mathsf{f} \in \mathsf{F}^+_{\mathsf{a}}, \mathsf{g} \in \mathsf{F}^-_{\mathsf{a}}$ such that $x_{\mathsf{f}} = x_{\mathsf{g}} = 1$. This implies the chain (f, g) forms part of a route in the solution, and is therefore legal. Hence, $\mathsf{g} \in \mathsf{F}_{(\mathsf{f},\cdot)} \cap \mathsf{F}^-_{\mathsf{a}} \neq \emptyset$ so $\mathsf{f} \in \mathsf{F}_{(\mathsf{a},\cdot)}$ and the inequality is satisfied.

Appendix B: Enumeration algorithm details

During Step 1 of Algorithm 1 we iteratively extend paths which start on some pickup location $p \in P$. At every extension step we store three resources: the time t, which is the earliest we may arrive at the last node of the path, neglecting ride time constraints, the current load q and a hashmap $\mathcal{T}$. $\mathcal{T}$ contains as keys those customers currently onboard, as well as the origin depot location. The values of $\mathcal{T}$ are underestimates of the ride time and the total route duration.

Algorithm 2 details the path generation and trimming steps of Algorithm 1. The function GetSchedule(ρ) attempts to compute a DARP schedule for the path ρ, using the method of Tang et al. (2010). The function TrimPath(ρ) enumerates all the possible restricted fragments resulting from the route-path ρ (as described in Section 4).

Algorithm 1 The Update step of algorithm 2

procedure Update(i, j, t, q, $\mathcal{T}$)
  $t \leftarrow \max\{e_j, t + t_{ij}\}$
  $q \leftarrow q + q_j$
  for $k \in \mathcal{T}$ **do**
    $\mathcal{T}(k) \leftarrow \mathcal{T}(k) + t_{ij}$
  end for
  if $j \in D$ **then**
    remove key p from $\mathcal{T}$ where $d(p) = j$
  else
    insert $\mathcal{T}(j) \leftarrow 0$
  end if
  return t, q, $\mathcal{T}$
end procedure

Algorithm 2 Generating F' by extending and trimming paths.

Initialise set of potential restricted fragments: $\mathsf{F}' \leftarrow \emptyset$

procedure EXTEND(ρ, t, q, $\mathcal{T}$)
  $i \leftarrow$ last location of path ρ
  if $t > l_i$ or $q > Q$ or Any($\mathcal{T}(j) > r_j$ for $j \in \mathcal{T}$) or $\mathcal{T}(o^+) + t_{i,o^-} > r_{o^+}$ **then**
    return `false` ▷ Some constraint has been violated
  end if
  if the only key in $\mathcal{T}$ is o^+ **then**
    if GETSCHEDULE(ρ) finds a legal schedule **then**
      $\mathsf{F}' \leftarrow \mathsf{F}' \cup$ TRIMPATH(ρ).
      return `true`
    else
      return `false`
    end if
  end if
  Initialise `can_complete` $\leftarrow$ `false`
  for $j \in \mathcal{T}$ such that $j \neq o^+$ **do** ▷ Loop through onboard customers
    t', q', $\mathcal{T}' \leftarrow$ UPDATE(i,$j+n$,t, q, $\mathcal{T}$)
    `can_complete` $\leftarrow$ `can_complete` $\vee$ EXTEND($(\rho, d(j))$, t', q', $\mathcal{T}'$)
  end for
  if $i \in P$ and `can_complete` **then** ▷ Extend to pickups only if a completion exists
    for $j \in P$ such that $j \notin \rho$ **do**
      t', q', $\mathcal{T}' \leftarrow$ UPDATE(i,j,t, q, $\mathcal{T}$)
      `can_complete` $\leftarrow$ `can_complete` $\vee$ EXTEND((ρ, j), t', q', $\mathcal{T}'$)
    end for
  end if
  return `can_complete`
end procedure

for $p \in P$ **do** ▷ This could be done in parallel
  EXTEND($(p,)$, e_p, q_p, $\{(o^+ \rightarrow t_{o^+,p}), (p \rightarrow 0)\}$)
end for

Appendix C: Branch-and-Cut Procedure

Algorithm 3 The Branch-Cut procedure of Algorithm 2

procedure BRANCH-CUT(`USE_VI`, `USE_HEUR`)
  while new node solved **do**
    if node gives new integer solution **then**
      if cyclic route **or** schedule violation **then**
        Add IPEC and cycle constraints (33) and (38)–(41)
      end if
      if `USE_HEUR` **and** OptGap > 0.01 **then**
        run LNS heuristic: solve with BRANCH-CUT(`false`, `false`)
        if LNS failed **and** IPEC/cycle constraint added **and** (50) satisfied **then**
          run FR heuristic: solve with BRANCH-CUT(`false`, `false`)
        end if
      end if
    else if `USE_VI` **then**
      if OptGap > 0.001 **and** (explored nodes ≤ 10 **or** (49) satisfied) **then**
        Separate and add valid inequalities (42)–(48) with violation > 0.01
      end if
    end if
  end while
end procedure

Appendix D: Notation

Table 1 Summary of notation.

Symbol	Description
n	Number of customers.
P	Pickup locations.
D	Delivery locations.
o^+, o^-	Origin and destination depot locations, respectively.
N	All locations ($N = P \cup D \cup \{o^+, o^-\}$).
$d(p)$	Delivery location corresponding to $p \in P$.
t_{ij}	Travel time between $i,\ j \in N$.
c_{ij}	Travel cost between $i,\ j \in N$.
q_i	Demand of location $i \in N$.
Q	Capacity of a vehicle.
r_p	Maximum ride time of customer $p \in P$.
r_{o^+}	Maximum route duration.
$[e_i, l_i]$	Time window of location $i \in N$.
N	Set of abstract nodes.
P	Set of nodes with pickup locations.
D	Set of nodes with delivery locations.
A	Set of arcs between nodes.
F	Set of restricted fragments.
$\mathsf{F}^-_{\mathsf{n}}$	Restricted fragments starting at node n.
$\mathsf{F}^+_{\mathsf{n}}$	Restricted fragments ending at node n.
$\mathsf{A}^-_{\mathsf{n}}$	Arcs starting at node n.
$\mathsf{A}^+_{\mathsf{n}}$	Arcs ending at node n.
F_i	Fragments which visit location $i \in P \cup D$.
$\text{Load}(\mathsf{n})$	The partial load at node $\mathsf{n} \in \mathsf{N}$.
$\text{Loc}(\mathsf{n})$	The physical location, $\text{Loc}(\mathsf{n}) \in N$, for node $\mathsf{n} \in \mathsf{N}$.
$\text{Start}(\cdot)$	The start node for a fragment or arc.
$\text{End}(\cdot)$	The end node for a fragment or arc.
$\text{Locs}(\mathsf{f})$	The set of physical locations, $\text{Locs}(\mathsf{f}) \subset N$ that f visits.
T_{f}	The sum of travel within f.
$\tau^{\text{early}}_{\mathsf{f}}(t)$	The earliest schedule of fragment f starting no earlier than t.
$\tau^{\text{late}}_{\mathsf{f}}(t)$	The latest assignment of fragment f finishing no later than t.
E_{f}	The earliest time service may start at the last location in f.
L_{f}	The latest time service may start at the first location in f.
$\mathsf{F}^+_{\mathsf{a}}$	Shorthand for $\mathsf{F}^+_{\text{Start}(\mathsf{a})}$.
$\mathsf{F}^-_{\mathsf{a}}$	Shorthand for $\mathsf{F}^-_{\text{End}(\mathsf{a})}$.
$\mathsf{A}^+_{\mathsf{f}}$	Shorthand for $\mathsf{A}^+_{\text{Start}(\mathsf{f})}$.
$\mathsf{A}^-_{\mathsf{f}}$	Shorthand for $\mathsf{A}^-_{\text{End}(\mathsf{f})}$.
c	A chain of fragments; $\mathsf{c} = (\mathsf{f}_1, \ldots, \mathsf{f}_l)$.
$\mathsf{F}^-_{\mathsf{c}}$	The fragments $\mathsf{g} \in \mathsf{F}$ which make the chain $(\mathsf{f}_1, \ldots, \mathsf{f}_{l-1}, \mathsf{g})$ legal.
$\mathsf{F}^+_{\mathsf{c}}$	The fragments $\mathsf{g} \in \mathsf{F}$ which make the chain $(\mathsf{g}, \mathsf{f}_2, \ldots, \mathsf{f}_l)$ legal.
$\mathsf{A}^-_{\mathsf{c}}$	The arcs $\mathsf{a} \in \mathsf{A}^-_{\mathsf{f}_{l-1}}$ for which there exists a $\mathsf{g} \in \mathsf{F}^-_{\mathsf{a}}$ that makes the chain $(\mathsf{f}_1, \ldots, \mathsf{f}_{l-1}, \mathsf{g})$ legal.
$\mathsf{A}^+_{\mathsf{c}}$	The arcs $\mathsf{a} \in \mathsf{A}^+_{\mathsf{f}_1}$ for which there exists a $\mathsf{g} \in \mathsf{F}^+_{\mathsf{a}}$ that makes the chain $(\mathsf{g}, \mathsf{f}_2, \ldots, \mathsf{f}_l)$ legal.
$\mathsf{F}_{(\mathsf{f},\cdot)}$	Fragments which can be placed after f in a route.
$\mathsf{F}_{(\cdot,\mathsf{f})}$	Fragments which can be placed before f in a route.
$\mathsf{A}_{(\mathsf{f},\cdot)}$	Arcs leaving $\text{End}(\mathsf{f})$ which lead to the depot or to a compatible fragment.
$\mathsf{A}_{(\cdot,\mathsf{f})}$	Arcs ending at $\text{Start}(\mathsf{f})$ which lead from the depot or from a compatible fragment.
$\mathsf{F}_{(\mathsf{a},\cdot)}$	Fragments which leave $\text{End}(\mathsf{a})$ for which there exists a compatible fragment arriving at $\text{Start}(\mathsf{a})$.
$\mathsf{F}_{(\cdot,\mathsf{a})}$	Fragments which arrive at $\text{Start}(\mathsf{a})$ for which there exists a compatible fragment leaving $\text{End}(\mathsf{a})$.

Appendix E: Single-Thread Computational Results

The results shown here are aggregates of 5 runs with different solver seeds, running on a 2.6GHz CPU with a single thread, 50GB of memory and a 1-hour time limit. As in Table 3, the UB column is the best UB founds across the 5 runs and the other columns are averages.

Table 2 **Single-threaded results for extended A and B instances. Columns have the same meaning as in Table 3**

Instance	B&C Tree					Valid Inequalities				Solve Time (s)		
	Nodes	Cuts	LB	UB	Gap	AF	FA	FF	SR	Network	Tree	Total
a2-16-X	1.0	0.0	*	278.235	*	3.0	3.0	0.0	0.0	0.0	0.1	0.1
a2-20-X	0.0	2.0	*	330.691	*	8.0	8.0	0.0	0.0	0.0	0.1	0.1
a2-24-X	1.0	13.6	*	389.095	*	10.0	10.2	0.0	8.0	0.1	0.3	0.4
a3-18-X	0.0	7.0	*	272.705	*	7.0	7.0	0.0	2.0	0.1	0.1	0.2
a3-24-X	1.0	7.0	*	289.570	*	8.0	8.0	4.0	0.0	0.1	0.1	0.3
a3-30-X	0.0	8.0	*	452.829	*	13.0	10.0	0.0	8.0	0.1	0.2	0.3
a3-36-X	1.0	0.0	*	501.022	*	1.0	1.0	0.0	0.0	0.1	0.1	0.3
a4-16-X	1.0	16.0	*	235.174	*	5.6	5.6	0.0	16.0	0.1	0.2	0.2
a4-24-X	0.0	4.0	*	359.340	*	3.0	3.0	0.0	4.0	0.1	0.1	0.2
a4-32-X	1.0	13.2	*	447.278	*	3.0	3.0	0.0	13.6	0.2	0.3	0.5
a4-40-X	0.0	0.0	*	509.018	*	10.0	10.0	4.0	4.0	0.3	0.3	0.6
a4-48-X	7.2	16.4	*	618.958	*	20.8	23.2	6.4	27.0	0.4	1.4	1.8
a5-40-X	0.0	7.0	*	464.066	*	10.0	10.0	4.0	36.0	0.3	0.3	0.6
a5-50-X	1.0	5.6	*	621.957	*	8.2	8.2	4.0	26.4	0.4	0.7	1.1
a5-60-X	1.0	14.8	*	745.398	*	13.2	14.0	0.4	8.8	0.6	0.9	1.5
a6-48-X	1.0	10.4	*	572.513	*	21.8	22.6	17.4	11.2	0.8	0.8	1.6
a6-60-X	1.0	15.6	*	757.876	*	19.2	22.6	9.4	59.6	1.1	1.4	2.4
a6-72-X	1.0	15.0	*	868.288	*	26.0	22.6	9.8	15.6	1.0	1.7	2.6
a7-56-X	1.0	8.8	*	663.428	*	33.8	36.2	19.8	23.8	0.8	1.6	2.4
a7-70-X	1.0	10.0	*	824.428	*	33.4	36.4	21.2	35.0	1.7	1.9	3.6
a7-84-X	61.0	19.8	*	950.577	*	62.0	60.8	39.4	38.2	1.8	6.6	8.5
a8-64-X	1.0	10.0	*	701.151	*	33.6	35.6	27.6	38.0	1.7	2.1	3.8
a8-80-X	1.0	29.2	*	880.288	*	42.6	42.6	33.8	56.8	1.6	3.7	5.3
a8-96-X	324.6	44.2	*	1115.093	*	58.4	59.0	46.2	74.8	2.9	16.2	19.2
b2-16-X	0.0	4.0	*	244.319	*	5.0	7.0	8.0	6.0	0.2	0.1	0.4
b2-20-X	1.0	11.0	*	280.211	*	1.0	1.0	0.0	0.0	0.2	0.1	0.3
b2-24-X	1.0	8.2	*	372.589	*	9.0	7.0	9.0	5.0	0.1	0.2	0.4
b3-18-X	1.0	13.2	*	251.949	*	12.0	16.0	1.0	15.0	0.7	0.5	1.1
b3-24-X	1.0	22.0	*	321.209	*	8.6	8.6	4.0	2.0	0.5	0.5	1.0
b3-30-X	1.0	12.4	*	433.282	*	15.0	12.2	5.2	19.8	0.5	0.5	0.9
b3-36-X	1.0	10.0	*	486.014	*	10.0	9.0	2.0	8.0	0.6	0.4	1.0
b4-16-X	10.4	16.8	*	233.172	*	14.6	17.0	7.2	9.6	0.3	0.7	1.0
b4-24-X	1.0	17.6	*	292.969	*	27.2	24.6	12.8	20.6	1.0	1.0	2.0
b4-32-X	1.0	15.4	*	410.142	*	8.8	7.8	7.0	5.0	0.6	0.4	1.0
b4-40-X	0.0	5.6	*	478.675	*	7.0	7.0	10.0	8.0	2.5	0.9	3.4
b4-48-X	1.0	19.8	*	543.252	*	25.2	25.0	15.8	37.2	7.1	3.9	11.0
b5-40-X	1.0	8.0	*	494.781	*	33.6	41.0	35.8	39.4	3.2	3.2	6.4
b5-50-X	56.0	16.4	*	606.820	*	74.0	75.4	88.4	86.4	6.4	15.8	22.2
b5-60-X	1.0	5.6	*	704.518	*	40.2	43.2	54.2	67.4	6.9	6.8	13.6
b6-48-X	412.6	32.8	*	577.822	*	69.6	66.6	84.8	57.4	6.1	19.5	25.7
b6-60-X	199.0	48.2	*	703.240	*	99.8	98.4	124.2	75.2	12.6	41.3	53.9
b6-72-X	1.0	42.0	*	789.145	*	110.6	110.4	164.4	52.6	16.6	40.4	56.9
b7-56-X	0.8	2.4	*	603.941	*	81.6	80.8	148.8	89.6	69.4	51.4	120.8
b7-70-X	2169.6	78.0	*	740.946	*	213.4	226.4	294.6	116.0	21.3	139.9	161.2
b7-84-X	3372.0	40.8	*	961.569	*	250.6	269.2	373.6	195.0	22.6	323.2	345.8
b8-64-X	1624.6	42.0	*	638.461	*	230.8	239.8	372.6	142.0	55.4	241.0	296.4
b8-80-X	6607.0	131.4	*	822.222	*	262.4	259.2	312.0	135.0	21.3	399.1	420.4
b8-96-X	13082.8	126.2	947.543	951.952	0.463%	530.6	574.4	980.0	280.6	91.3	3508.7	1h

Table 3 Single-threaded results for RS and selected R instances. Columns have the same meaning as in Table 3

Instance	B&C Tree					Valid Inequalities				Solve Time (s)		
	Nodes	Cuts	LB	UB	Gap	AF	FA	FF	SR	Network	Tree	Total
R1a	0.0	0.0	*	190.019	*	27.0	27.0	37.6	44.0	121.9	18.6	140.5
R7a	460.8	145.8	*	291.711	*	315.8	327.4	604.6	107.0	802.3	1444.2	2246.5
R1b	1.0	11.0	*	164.460	*	81.6	82.8	210.4	70.6	1718.0	516.4	2234.4
R1a-S	0.0	0.0	*	206.725	*	0.0	0.0	0.0	0.0	0.3	0.1	0.4
R2a-S	135.0	100.4	*	333.549	*	130.4	111.8	78.0	96.6	8.8	29.5	38.3
R3a-S	3992.2	141.6	*	618.363	*	159.2	164.4	80.4	308.8	7.0	102.8	109.9
R4a-S	345.0	31.6	*	675.724	*	174.0	168.4	172.0	171.2	40.2	101.3	141.5
R5a-S	612.6	273.6	728.625	735.736	0.966%	412.2	440.6	477.0	402.2	221.7	3378.3	1h
R6a-S	492.0	272.6	916.480	930.638	1.521%	521.2	539.0	598.4	435.6	208.7	3391.3	1h
R7a-S	987.6	144.2	*	329.364	*	62.2	62.6	25.6	47.4	0.8	35.2	36.0
R8a-S	39369.6	547.0	*	585.619	*	438.4	522.4	319.4	339.6	12.5	1605.6	1618.0
R9a-S	2124.0	2862.8	764.893	-	-	584.6	532.8	422.6	380.8	60.9	3539.1	1h
R10a-S	531.2	1955.4	980.640	1074.537	8.738%	651.4	664.2	551.2	414.4	192.3	3407.7	1h
R1b-S	1.0	6.0	*	185.165	*	4.0	4.0	2.0	4.0	1.4	0.5	1.8
R2b-S	5784.2	199.0	*	335.791	*	217.6	268.0	230.8	154.2	28.2	277.4	305.6
R3b-S	5220.2	668.2	571.576	586.423	2.532%	504.4	519.2	451.2	313.6	83.9	3516.1	1h
R4b-S	215.8	530.0	611.543	636.184	3.873%	477.2	511.6	644.0	265.2	616.9	2983.1	1h
R5b-S	0.0	0.0	-	-	-	0.0	0.0	0.0	0.0	2592.9	1007.1	1h
R6b-S	0.0	0.0	-	-	-	0.0	0.0	0.0	0.0	2347.4	1252.6	1h
R7b-S	988.2	133.0	*	290.341	*	131.0	119.2	68.8	68.4	7.1	41.4	48.5
R8b-S	12555.8	349.4	*	532.284	*	373.0	408.2	324.8	393.4	54.0	1560.9	1614.8
R9b-S	368.2	1225.6	686.201	706.405	2.860%	593.2	564.8	575.8	352.0	408.4	3191.6	1h
R10b-S	0.0	3.2	193.037	-	-	408.0	440.0	385.6	212.8	1864.4	1735.6	1h

References

Tang J, Kong Y, Lau H, Ip AW, 2010 *A note on "efficient feasibility testing for dial-a-ride problems"*. *Operations Research Letters* 38(5):405–407, URL http://dx.doi.org/10.1016/j.orl.2010.05.002.