

# Sharing Daily Travel Time across Multiple-periods Dial-a-Ride Problem

Timothée Chane-Hai<sup>1</sup>, Samuel Vercaene<sup>1</sup> and Thibaud Monteiro<sup>1</sup>

**Abstract**—This paper introduces the commuting dial-a-ride problem (Com-DARP). In this variant of the dial-a-ride problem (DARP), the goal is to find a routing plan that minimizes the transportation cost while serving one morning request and one afternoon request for each user. In addition to the classic constraints of the DARP, there is a daily ride time limit for serving a user’s morning request and evening request. The main difference with other multi-trip DARP variants is therefore the relation between morning and afternoon requests, due to the users’ daily maximum ride times. We solve the Com-DARP using a large neighborhood search heuristic coupled with a new precedence filter. We test our solution method on a new benchmark of Com-DARP instances. The results show that the precedence filter speeds up the computation and improves the quality of the solutions. To obtain managerial insights, we compare the results from the Com-DARP with those from the combination of the equivalent morning and afternoon DARPs (2-DARP). For the same problem, the 2-DARP achieves a cost increase of 6.9% on average and up to 9.1%.

## I. INTRODUCTION

In a dial-a-ride system, a fleet of vehicles transports users from their pickup to drop-off locations, sharing vehicles to serve them. Users can be schoolchildren, patients, the elderly, disabled people, or workers. This system offers more flexibility than traditional public transport, especially for vulnerable groups or low-density regions. With an aging population and a need to reduce carbon emissions, these systems are increasingly beneficial. Vulnerable users can access services more easily, while commuters save money and reduce their carbon footprint by sharing rides.

The Dial-A-Ride Problem (DARP) involves creating a routing plan that meets user demands, ensures service quality, and minimizes objectives like cost or pollution. As an NP-hard problem, solving DARP for large-scale instances is challenging. Exact methods handle small instances, and metaheuristics extend this limit to a few hundred requests. However, real-life scenarios with thousands of requests and complexities like heterogeneous vehicles and multiple depots are harder to solve.

During a previous research project dedicated to disabled people transportation, the initial analysis revealed that trip durations, particularly their equitable distribution, were a major concern for passengers [1]. Often, the person closest to the depot was the first to board in the morning and the last to disembark in the evening. This article presents our research to find a more equitable yet logistically efficient solution.

This paper introduces a generalization: the Commuting Dial-A-Ride Problem (Com-DARP), where each user has

two requests, one for the morning and one for the afternoon. The novelty lies in having a daily maximum ride time that can be distributed unevenly between morning and afternoon rides. As illustrated by Figure 1, this flexibility expands the solution space, allowing for unbalanced ride times while still respecting the daily time limit, offering new feasible solutions.

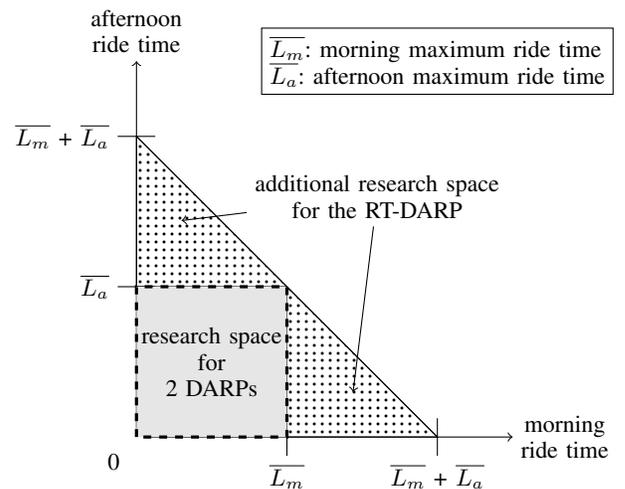


Fig. 1. Daily maximum ride time decomposition of the Commuting Dial-A-Ride Problem (Com-DARP).

The article is structured as follows: Section II presents a literature review, Section III introduces the Com-DARP mathematical formulation, Section IV details the solution method, Section V covers computational results, and Section VI provides final conclusions.

## II. LITERATURE REVIEW

### A. The Dial-A-Ride Problem

The DARP is a classic problem in operations research, first introduced by [2], with the formulation by [3] being the standard. This section summarizes key contributions to the DARP. For a more detailed review on classical DARP, we refer readers to a recent surveys by [4].

a) *Multi-trip problems*: The Com-DARP belongs to the multi-trip problem category. Multi-trip problem is a problem, where vehicles can return to the depot multiple times during the planning period, a feature introduced in [5] to model more realistic scenarios with high demand and short travel times.

In the DARP context, the multi-trip component accounts for real constraints and improves service quality, especially in sanitary transportation. Studies like [6] consider drivers’

<sup>1</sup> DISP, UR4570, Univ Lyon, INSA Lyon, Univ Lyon 1, Univ Lyon 2, INSA de Lyon, Bât Léonard de Vinci, 21 avenue Jean Capelle, Villeurbanne, 69621, France `firstname.name@insa-lyon.fr`

lunch breaks, while add coffee breaks. In these studies, users have one request, making the multi-trip aspect related only to vehicles. However, in cases like [7], [8], and [9], where users have multiple requests over several days, the multi-trip component is also linked to users. These studies ensure consistency in drivers' schedules, service times, and groups to maintain service quality.

Our problem's novelty lies in the daily maximum ride time, which creates a dependency between users' requests.

*b) Solution method:* Given the limitations of exact methods, most research focuses on heuristics. One popular approach is Large Neighborhood Search (LNS), first introduced in [10] and later refined in [11] under the "ruin and recreate" method. The LNS improves the current solution by disturbing and repairing it, stopping when a criterion is met.

In this paper, we use the LNS approach to solve the Com-DARP.

*c) Feasibility testing:* A major challenge in solving the DARP is schedule evaluation. Unlike Pickup and Delivery Problems (PDP), testing route feasibility in DARP is more complex due to users' maximum ride times [12]. To address this, several studies have developed a testing procedure based on forward time slack [13], introduced for DARP by [3], with a complexity of  $O(n^2)$ . A  $O(n)$  three-pass algorithm presented in [14] can incorrectly declare infeasibility due to ride time constraints. A revised version with  $O(n^2)$  complexity, shown to perform better than [15]'s linear-time algorithm, was proposed in [16]. A constant-time feasibility check with  $O(n^3)$  preprocessing was later introduced in [17].

To further speed up the algorithms, additional tests, summarized in [18], have been developed. A recent precedence test in [19], called feasible search range, saves feasible insertion positions for each node in each route, avoiding unfeasible ones. Most of the computation occurs during preprocessing with a  $O(n^2)$  complexity, with updates in  $O(n^2)$  and retrieval in  $O(1)$ .

In this work, we adapt this precedence test for the Com-DARP, as the dependencies between requests dynamically affect the precedence constraints between nodes.

## B. Contributions

This work makes several contributions to the literature. First, we introduce the commuting dial-a-ride problem (Com-DARP) with its mathematical model, where users have multiple requests and a global maximum ride time. Second, we develop a new route feasibility check that leverages precedence constraints between nodes and adapts dynamically during computation. Third, we compare the performance of the new LNS framework with the classic LNS framework. Finally, we draw managerial insights by comparing the Com-DARP with the 2-DARP, where the morning and evening DARPs are solved independently.

## III. PROBLEM DESCRIPTION

This section introduces the mathematical formulation of the Com-DARP.

### A. General problem setting

The Com-DARP is modeled on a complete directed graph  $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ .  $\mathcal{A}$  is the set of arcs and  $\mathcal{N}$  is the set of nodes.  $\mathcal{P} \subset \mathcal{N}$  is the set of pickup nodes,  $\mathcal{D} \subset \mathcal{N}$  is the set of drop-off nodes,  $\mathcal{O}^+ \subset \mathcal{N}$  is the set of starting depot nodes, and  $\mathcal{O}^- \subset \mathcal{N}$  is the set of ending depot nodes. In practice, a starting depot and an ending depot can correspond to the same location. Each starting depot has a corresponding ending depot such that  $\mathcal{E} \subset \mathcal{O}^+ \times \mathcal{O}^-$  is the set of pairs starting and ending depots. Finally,  $\mathcal{U}$  is the set of users,  $\mathcal{R}$  is the set of requests, and  $\mathcal{K}$  is the set of vehicles.

Each arc  $(i, j) \in \mathcal{A}$  has a travel duration  $T_{ij} \geq 0$  and a travel distance  $D_{ij} \geq 0$ . All travel times and distances are non-negative and travel times satisfy the triangle inequality.

Each user  $u \in \mathcal{U}$  has a set of requests  $\mathcal{R}_u \subset \mathcal{R}$ . Within this set, each request  $r \in \mathcal{R}$  corresponds to a pickup node  $P_r \in \mathcal{P}$  and a drop-off node  $D_r \in \mathcal{D}$ . The transportation respects the maximum ride time of the request  $\bar{L}_r \geq 0$ . Additionally, among the time horizon  $H$ , for each user  $u$  there is a maximum ride time associated with this set of requests, which is  $\bar{L}_u \geq 0$ .

At each node, the arrival time is in its time window  $[A_i, B_i]$  with  $B_i \geq A_i \geq 0$ . The service duration at the node is  $S_n \geq 0$  and the load variation is  $Q_n$ . For a request  $r \in \mathcal{R}$  with  $n^+$  the pickup node and  $n^-$  the drop-off node,  $Q_{n^+} = -Q_{n^-}$ . For a depot node  $o \in \mathcal{O}^+ \cup \mathcal{O}^-$ , the service duration is null:  $S_o = 0$  as is the demand:  $Q_o = 0$ .

Each vehicle  $k \in \mathcal{K}$  has a capacity  $\bar{Q}_k \geq 0$ , a fixed cost  $C_k^f \geq 0$ , a cost related to the distance  $C_k^d \geq 0$  and a cost related to the ride time  $C_k^t \geq 0$ . As a vehicle can do multiple trips, it has a set of starting depots  $\mathcal{O}_k^+ \in \mathcal{O}^+$  and a set of ending depots  $\mathcal{O}_k^- \in \mathcal{O}^-$ .  $\mathcal{E}_k \in \mathcal{E}$  is the set of pairs of starting and ending depots for this vehicle. Each individual trip is limited by a maximum trip ride time  $\bar{L}_{o_k^+} \geq 0$ ,  $o_k^+ \in \mathcal{O}_k^+$  and the vehicle's overall ride time is limited by the maximum total ride time  $\bar{L}_k \geq 0$ .

The mathematical model presented here is a generalization of the Com-DARP. This formulation corresponds exactly to a multi-request multi-trip multi-depot heterogeneous fleet dial-a-ride problem. In the case of the Com-DARP, the time horizon  $H$  can be used to define a morning period  $[0, H/2]$  and an afternoon period  $[H/2, H]$ . As a result, a vehicle  $k \in \mathcal{K}$  has only two pairs of starting and ending depot nodes and a user  $u \in \mathcal{U}$  has only two requests. Time windows for the first pairs of depot nodes and the morning requests close during the morning period. Similarly, time windows for the second pairs of depot nodes and the afternoon requests open during the afternoon period.

### B. Mathematical model

The following variables are used in the mathematical model.

- $x_{ij}^k \in \{0, 1\}$  a variable that is equal to 1 if the vehicle  $k \in \mathcal{K}$  uses the arc  $(i, j) \in \mathcal{A}$ , and 0 otherwise;
- $y^k \in \{0, 1\}$  a variable that is equal to 1 if the vehicle  $k \in \mathcal{K}$  is used;
- $l_i^k \in \mathbb{R}^+$  a variable that corresponds to the number of users in the vehicle  $k \in \mathcal{K}$  after leaving the node  $i \in \mathcal{N}$ ;

- $w_i^k \in \mathbb{R}^+$  a variable which corresponds to the service time of the vehicle  $k \in \mathcal{K}$  at node  $i \in \mathcal{N}$ ;
- $w_i \in \mathbb{R}^+$  a variable that corresponds to the service time at node  $i \in \mathcal{N}$ ;
- $z_r \in \mathbb{R}^+$  a variable that corresponds to the ride time of request  $r \in \mathcal{R}$ .

The Com-DARP can be formulated with the following mixed integer program:

Objective

$$\min \sum_{k \in \mathcal{K}} C_k^f y^k + \sum_{k \in \mathcal{K}} \left( \sum_{(o_k^-, o_k^+) \in \mathcal{E}_k} C_k^h (w_{o_k^-} - w_{o_k^+}) + \sum_{(i,j) \in \mathcal{A}} C_k^d D_{ij} x_{ij}^k \right) \quad (1)$$

Flow constraints

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

$$\sum_{k \in \mathcal{K}} \sum_{(i, P_r) \in \mathcal{A}} x_{i P_r}^k = 1 \quad \forall r \in \mathcal{R} \quad (3)$$

$$\sum_{(P_r, j) \in \mathcal{A}} x_{P_r j}^k = \sum_{(i, D_r) \in \mathcal{A}} x_{i D_r}^k \quad \forall r \in \mathcal{R}, k \in \mathcal{K} \quad (4)$$

$$\sum_{j \in \mathcal{N}} x_{o_k^+ j}^k = \sum_{j \in \mathcal{N}} x_{j o_k^-}^k = 1 \quad \forall k \in \mathcal{K}, (o_k^+, o_k^-) \in \mathcal{E}_k \quad (5)$$

$$x_{j o_k^+}^k = x_{o_k^- j}^k = x_{jj}^k = 0 \quad \forall k \in \mathcal{K}, (o_k^+, o_k^-) \in \mathcal{E}_k \quad (6)$$

Time constraints

$$w_i = \sum_{k \in \mathcal{K}} w_i^k \quad \forall i \in \mathcal{N} \quad (7)$$

$$w_i^k + S_i + T_{ij} \leq w_j^k + M_{ij}(1 - x_{ij}^k) \quad \forall (i, j) \in \mathcal{A}, k \in \mathcal{K} \quad (8)$$

$$w_{P_r} + S_{P_r} + T_{P_r, D_r} \leq w_{D_r} \quad \forall r \in \mathcal{R} \quad (9)$$

$$A_j \leq w_j \leq B_j \quad \forall j \in \mathcal{N} \quad (10)$$

$$w_i^k \leq B_i \sum_{j \in \mathcal{N}} x_{ij}^k \quad \forall i \in P \cup D \cup O_k^+ \quad (11)$$

$$w_i^k \leq B_i \sum_{j \in \mathcal{N}} x_{ji}^k \quad \forall i \in O_k^- \quad (12)$$

Vehicles constraints

$$y^k \geq \sum_{i \in \mathcal{P}} x_{o_k^+ i}^k \quad \forall k \in \mathcal{K}, o_k^+ \in O_k^+ \quad (13)$$

$$l_j^k \geq l_i^k + Q_j - M_j^k(1 - x_{ij}^k) \quad \forall (i, j) \in \mathcal{A}, k \in \mathcal{K} \quad (14)$$

$$l_i^k \leq \overline{Q}_k \sum_{j \in \mathcal{N}} x_{ij}^k \quad \forall i \in \mathcal{P}, k \in \mathcal{K} \quad (15)$$

$$l_{o_k^-}^k = 0 \quad \forall k \in \mathcal{K}, o_k^- \in O_k^- \quad (16)$$

Ride time constraints

$$w_{D_r} - (w_{P_r} + S_{P_r}) = z_r \quad \forall r \in \mathcal{R} \quad (17)$$

$$z_r \leq \overline{L}_r \quad \forall r \in \mathcal{R} \quad (18)$$

$$\sum_{r \in \overline{\mathcal{R}}_u} z_r \leq \overline{L}_u \quad \forall u \in \mathcal{U} \quad (19)$$

$$w_{o_k^+} - w_{o_k^-} \leq \overline{L}_{o_k^+} \quad \forall k \in \mathcal{K}, (o_k^+, o_k^-) \in \mathcal{E}_k \quad (20)$$

$$\sum_{(o_k^+, o_k^-) \in \mathcal{E}_k} w_{o_k^+} - w_{o_k^-} \leq \overline{L}_k \quad \forall k \in \mathcal{K} \quad (21)$$

Decision variables

$$x_{ij}^k, y^k \in \{0, 1\} \quad \forall i \in \mathcal{N}, j \in \mathcal{N}, k \in \mathcal{K} \quad (22)$$

$$l_i^k, w_i^k, w_i, z_r \in \mathbb{R}^+ \quad \forall i \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R} \quad (23)$$

With  $M_{ij} = B_i + S_i + T_{ij}$  and  $M_j^k = \overline{Q}_k + Q_j$ .

The objective (1) is to minimize the total transportation cost. The transportation cost considers the duration, the distance, and the use of vehicles. Constraints (2) ensure that the flow is conserved. Constraints (3) and (4) ensure that each demand is served exactly once and by exactly one vehicle. Constraints (5) ensure one and only one vehicles per depots and Constraints (6) remove unnecessary arcs. The service time at a node is computed according to the service time at the previous node, the service duration, and the arc duration as described in Constraints (7) and (8). Constraints (9) ensure that a pickup is served before its corresponding drop-off. The service time at a node respects its time windows as defined in Constraints (10). Constraints (11) and (12) set service time of a node to 0 when the node is not visited by vehicle  $k$ . Constraints (13) correspond to the vehicle use. The load variation during a route is defined by Constraint (14). Constraint (15) ensures that the load does not exceed the vehicle capacity. Constraints (16) are set to reduce the space and ensure that vehicles return empty to their depots. Ride time constraints are expressed by Constraints (17) to (21). Constraints (17) compute the ride time value associated with a request. Constraints (18) ensure that the ride time does not exceed the maximum ride time. Constraints (19) are specific to our problem. These constraints create the dependency between all the requests of a user. It ensures that the cumulative ride times from the requests do not exceed the user's maximum ride time for the entire set of requests. For the Com-DARP, they ensure that the sum of the morning and evening ride time does not exceed the daily max ride time. The same principle is applied to the vehicles in Constraints (20) and (21). Finally, Constraints (22) and (23) are used to define the variables.

Note that user preferences (e.g., prioritizing shorter morning trips) may be taken into account by changing values of  $\bar{L}_r$  in Constraints (17).

#### IV. SOLUTION METHOD

In this section, we detail our solution method for solving the Com-DARP. This method is based on a large neighborhood search (LNS) metaheuristic associated with a speed-up mechanism using a precedence filter (PF). We name the complete metaheuristic LNS-PF. The rest of this section is organized as follows: the LNS framework and the insertion and removal operators are described in Section IV-A, the mechanisms behind the precedence filter are explained in Section IV-B and the feasibility tests are listed in Section IV-C.

##### A. Large Neighborhood Search (LNS) framework

The LNS, based on [20], is presented in Algorithm 1.

---

##### Algorithm 1 LNS framework

---

```

1:  $S$ : initial solution
2: while stopping criterion is not reached do
3:    $S' \leftarrow S$ 
4:   Apply a removal operator selected randomly
5:     on  $\phi \in [\delta, \Delta]$  % of the requests
6:   Apply a insertion operator selected randomly
7:     to re-insert the removed requests
8:   if  $f(S') < \chi \cdot f(S^*)$  then
9:      $S \leftarrow S'$ 
10:  end if
11:  if  $f(S') < f(S^*)$  then
12:     $S^* \leftarrow S'$ 
13:  end if
14: end while
15: return  $S^*$ 

```

---

Three solutions are used in the LNS.  $S^*$  is the best solution found during the resolution.  $S'$  is the solution on which the insertions and removals are performed.  $S$  is the current solution, used to reset  $S'$  at each iteration. The algorithm is initialized at lines 1: the initial solution is loaded. The core of the algorithm corresponds to the while loop that starts at line 2. Until a stopping criterion is reached (in our case a time limit), the LNS will perform removals and insertions. The removal is done at line 4. The solution is repaired at line 6. Once a new solution is obtained, the acceptance criterion is applied at lines 8-13. The acceptance is performed in two steps. In the first step, the current solution is updated if the new solution is better than the best multiplied by the "Record2Record" parameter  $\chi$  [21]. In the second step, the best solution is updated if the new solution is better.

The LNS operators are used to remove and insert users in the solution. They are key components of any LNS heuristic. A good set of operators results in a good search intensification while allowing sufficient diversification. Many operators have been developed but as demonstrated in [22], it is possible to obtain very good results with only a few carefully selected

operators. We use the same operators as in [23]. Two removal operators: the random removal and the history removal. Four insertion operators: K-regret insertion with  $K \in \{1, 2, 3, 4\}$ . We refer the reader to [24] for a more in-depth explanation of these operators.

During the resolution, the insertion operators test the feasibility of millions of insertions. As the feasibility testing consumes most of the computation time, operators must be carefully implemented for best performance.

##### B. Precedence filter

In this section, a node  $n$  is represented by a tuple  $(uhs)$ , with a user  $u \in \mathcal{U}$ , a period (half-day)  $h \in (AM, PM)$ , and a site for pickup or drop-off  $s \in (P, D)$ . Similarly, a request  $r \in \mathcal{R}$  is represented by a tuple  $(uh)$ . If  $h$  denotes the morning period, then  $h'$  denotes the afternoon period, and vice-versa. Additionally, a dynamic time window is defined as  $[a_{uhs}, b_{uhs}]$  and a dynamic maximum ride time for a request as  $\bar{l}_{uh}$ .

1) *Definitions for the precedence filter:* The new notations allow us to formulate the precedence relation between two nodes, the dynamic maximum ride time, and the time windows tightening.

a) *Precedence between two nodes:* Let us consider two users  $u$  and  $w$  at sites  $s$  and  $r$  respectively. For a given period  $h$  these users are represented by the nodes  $(uhs)$  and  $(whr)$ . The node  $(whr)$  is a predecessor of  $(uhs)$  if it is not possible to leave  $(uhs)$  at its earliest and arrive in time at  $(whr)$ . This property is used in [19] to limit the number of tested insertions. Mathematically, the precedence between two nodes is defined by the following equation:

$$a_{uhs} + S_{uhs} + T_{uhs,whr} > b_{whr} \quad (24)$$

b) *Maximum ride time of a request:* The maximum ride time of a request depends on its pair request ride time. It can take its initial value or the user's remaining maximum ride time. This is mathematically defined by the following inequality:

$$\bar{l}_{uh} = \min\{\bar{L}_{uh}; \bar{L}_u - l_{uh'}\} \quad (25)$$

c) *Time window tightening:* The time windows are tightened according to the tightening procedure introduced in [25]. The time window tightening results in the following inequalities:

$$A_{uhP} \geq A_{uhD} - S_{uhP} - \bar{L}_{uh} \quad (26)$$

$$B_{uhP} \leq B_{uhD} - S_{uhP} - T_{uhP,uhD} \quad (27)$$

$$A_{uhD} \geq A_{uhP} + S_{uhP} + T_{uhP,uhD} \quad (28)$$

$$B_{uhD} \leq B_{uhP} + S_{uhP} + \bar{L}_{uh} \quad (29)$$

In the Com-DARP, another tightening can be performed according to the dynamic maximum ride time of a request. As defined by equation (25),  $\bar{l}_{uh} = \min\{\bar{L}_{uh}; \bar{L}_u - l_{uh'}\}$ . This dynamic tightening impacts the start of the time window

at pickup and the end of the time window at delivery. This is defined by the following equations:

$$a_{uhP} = \max\{A_{uhP}; A_{uhD} - S_{uhP} - \overline{l_{uh}}\}, \quad (30)$$

$$b_{uhP} = B_{uhP}, \quad (31)$$

$$a_{uhD} = A_{uhD}, \quad (32)$$

$$b_{uhD} = \min\{B_{uhD}; B_{uhP} + S_{uhP} + \overline{l_{uh}}\}, \quad (33)$$

2) *Precedences for the Com-DARP*: Given the four different types of time window tightening, the precedence relations  $a_{uhs} + S_{uhs} + T_{uhs,whr} > b_{whr}$  can be developed for four pairs of nodes: pickup/pickup, drop-off/drop-off, pickup/drop-off, drop-off/pick-up. The results are summarized here.

a) *Pickup/pickup*:

- the node  $(whP)$  is always a predecessor of  $(uhP)$  if:

$$0 < A_{uhP} + S_{uhP} + T_{uhP,whP} - B_{whP}$$

- the node  $(whP)$  is never a predecessor of  $(uhP)$  if:

$$0 \geq A_{uhD} - T_{uhP,uhD} + T_{uhP,whP} - B_{whP}$$

- the node  $(whP)$  becomes a predecessor of  $(uhP)$  when:

$$l_{uh'} > B_{whP} - A_{uhD} + \overline{L_u} - T_{uhP,whP}$$

b) *Drop-off/drop-off*:

- the node  $(whD)$  is always a predecessor of  $(uhD)$  if:

$$0 < A_{uhD} + S_{uhD} + T_{uhD,whD} - B_{whD}$$

- the node  $(whD)$  is never a predecessor of  $(uhD)$  if:

$$0 \geq A_{uhD} + S_{uhD} + T_{uhD,whD} - B_{whP} - S_{whP} - T_{whP,whD}$$

- the node  $(whD)$  becomes a predecessor of  $(uhD)$  if:

$$l_{wh'} > \overline{L_w} + B_{whP} + S_{whP} - A_{uhD} - S_{uhD} - T_{uhD,whD}$$

c) *Pickup/drop-off*:

- the node  $(whD)$  is always a predecessor of  $(uhP)$  if:

$$0 < A_{uhP} + S_{uhP} + T_{uhP,whD} - B_{whD}$$

- the node  $(whD)$  is never a predecessor of  $(uhP)$  if:

$$0 \geq A_{uhD} - T_{uhP,uhD} + T_{uhP,whD} - B_{whP} - S_{whP} - T_{whP,whD}$$

- the node  $(whD)$  becomes a predecessor of  $(uhP)$  if one of the three cases is valid:

$$\left\{ \begin{array}{l} l_{uh'} > \overline{L_u} - A_{uhD} - T_{uhP,whD} \\ \quad \quad \quad + B_{whD} \\ l_{wh'} > \overline{L_w} + B_{whP} + S_{whP} - A_{uhP} \\ \quad \quad \quad - S_{uhP} - T_{uhP,whD} \\ l_{uh'} + l_{wh'} > \overline{L_u} + \overline{L_w} - A_{uhD} \\ \quad \quad \quad - T_{uhP,whD} + B_{whP} + S_{whP} \end{array} \right.$$

d) *Drop-off/pickup*:

- the node  $(whP)$  is always a predecessor of  $(uhD)$  if:

$$0 < A_{uhD} + S_{uhD} + T_{uhD,whP} - B_{whP}$$

- the node  $(whP)$  is never a predecessor of  $(uhD)$  if:

$$0 \geq A_{uhD} + S_{uhD} + T_{uhD,whP} - B_{whP}$$

### C. Feasibility testing

The feasibility of each candidate insertion must be tested. As a large number of insertions are performed while solving the problem, the feasibility test efficiency directly impacts the speed of the search. The tests must therefore be as efficient as possible. The order in which the different tests are performed is paramount. When testing the insertion of a request in a route, we first use the precedence filter to eliminate as many unfeasible positions as possible. Then, for each position we first execute the preliminary test and finally the schedule test.

## V. COMPUTATIONAL EXPERIMENTS

In the following experiments, we compare our metaheuristic to other metaheuristics for solving the Com-DARP problem. Our metaheuristic is called Large Neighborhood Search with Precedence Filter (LNS-PF). The other metaheuristic is the classic Large Neighborhood Search (LNS) without precedence filter. These acronyms are used throughout the rest of this section. The algorithms are coded in C++ and compiled into a 64-bits single-thread code using g++ 11.3.0. The algorithm has been adapted to solve both the classic DARP and the Com-DARP. The experiments are performed on a 2.0 GHz AMD EPYC 7702 CPU under a Linux Ubuntu 22.04.1 LTS operating system. For each instance, 5 independent runs are performed. In total, we generated 16 instances: 8 instances with 100 users (instances A) and 8 instances with 200 users (instances B). The computational time limit is set to 5 minutes for instances with 100 users and 20 minutes for instances with 200 users. These time limits include pre-computed operations and the optimization part.

The LNS-PF obtains the best results across all the instances. The precedence filter component further improves this gap by an additional 0.47%. Secondly, adding the precedence filter to the LNS allows us to perform on average 73% more iterations. These additional iterations give an edge to the LNS-PF for finding better solutions faster than the other metaheuristics. Thirdly, while the precedence filter does not provide as much gap improvement as the LNS does, it allows good solutions to be obtained more quickly. In other words, solutions of similar quality are reached but the LNS-PF reaches them faster than the LNS. This is the consequence of performing more iterations when the precedence filter is activated.

Then, under 2% gap solutions are obtained far sooner (25% to 40% of the solving time budget).

From a managerial point of view, solving the two corresponding morning and afternoon DARPs instead of the Com-DARP integrated problem presents a significant cost increases of 6.9% on average and up to 9.1%. However, it should also be noted that this is accompanied by an average increase of 5.5 minutes in travel time for users.

## VI. CONCLUSION

In this paper we introduced the commuting dial-a-ride problem (Com-DARP) and a specific metaheuristic to solve it. In this new variant of the dial-a-ride problem (DARP), a user has two requests, one in the morning and the other in the afternoon, each with its respective maximum ride time. In addition, each user has a maximum daily ride time. This feature was extracted from the door-to-door transportation of disabled children in France. However, the model and the solution method can be applied to any transportation system with commuting. Using a framework of LNS, we developed a new feasibility testing procedure that takes advantage of the precedence constraints and adapts dynamically during the search. The complete algorithm is called LNS-PF. The experiments on generated Com-DARP instances of 100 and 200 users showed a 0.47% gap improvement from the LNS-PF on average compared to classical LNS. In addition, the introduced precedence test increases the computation speed. From a managerial perspective, addressing the two separate morning and afternoon DARPs, rather than tackling the integrated Com-DARP problem, leads to a notable average cost increase of 6.9%, with peaks reaching up to 9.1%. It is important to highlight that this approach also results in an average travel time increase of 5.5 minutes for users.

This work opens two main perspectives for future research. First, more features could be integrated into this model such as heterogeneous users and reconfigurable vehicles, in order to get closer to real transportation systems. It would be interesting to study the multi-request multi-trip multi-depot heterogeneous fleet DARP introduced in this work as a generalization of the Com-DARP. It would also be interesting to study the adaptation of the precedence filter for other DARP variants like time-dependent travel times or stochastic demands and the impact of the varying user densities on the numerical results. Then it would be interesting to study the possibility to generate solutions with equivalent costs but lower ride times.

## REFERENCES

- [1] O. Tellez, L. Daguet, F. Lehuédé, T. Monteiro, G. Osorio Montoya, O. Péton, and S. Vercraene, "A stakeholder oriented approach to the optimization of transports of people with disabilities," *Supply Chain Forum: An International Journal*, vol. 21, no. 2, pp. 93–102, Apr. 2020, publisher: Taylor & Francis .eprint: <https://doi.org/10.1080/16258312.2020.1768435>.
- [2] H. Psarafitis, "A Dynamic Programming Solution to the Single Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem," *Transportation Science*, vol. 14, pp. 130–154, May 1980.
- [3] J.-F. Cordeau and G. Laporte, "A tabu search heuristic for the static multi-vehicle dial-a-ride problem," *Transportation Research Part B: Methodological*, vol. 37, no. 6, pp. 579–594, Jul. 2003.
- [4] S. C. Ho, W. Y. Szeto, Y.-H. Kuo, J. M. Y. Leung, M. Petering, and T. W. H. Tou, "A survey of dial-a-ride problems: Literature review and recent developments," *Transportation Research Part B: Methodological*, vol. 111, pp. 395–421, May 2018.
- [5] B. Fleischmann, "The vehicle routing problem with multiple use of the vehicles," *Fachbereich Wirtschaftswissenschaften, Universität Hamburg*, 1990.
- [6] S. N. Parragh, J.-F. Cordeau, K. F. Doerner, and R. F. Hartl, "Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints," *OR Spectrum*, vol. 34, no. 3, pp. 593–633, Jul. 2012.
- [7] K. Braekers and A. A. Kovacs, "A multi-period dial-a-ride problem with driver consistency," *Transportation Research Part B: Methodological*, vol. 94, pp. 355–377, Dec. 2016.
- [8] O. Tellez, S. Vercraene, F. Lehuédé, O. Péton, and T. Monteiro, "The time-consistent dial-a-ride problem," *Networks*, vol. 79, no. 4, pp. 452–478, 2022, .eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/net.22063>.
- [9] C. Lindstrøm and S. Røpke, "Group Consistent Dial-A-Ride using Adaptive Large Neighborhood Search: 13th International Conference on Computational Logistics," 2022.
- [10] P. Shaw, "Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems," in *Principles and Practice of Constraint Programming — CP98*, G. Goos, J. Hartmanis, J. van Leeuwen, M. Maher, and J.-F. Puget, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, vol. 1520, pp. 417–431, series Title: Lecture Notes in Computer Science.
- [11] G. Schrimpf, J. Schneider, H. Stamm-Wilbrandt, and G. Dueck, "Record Breaking Optimization Results Using the Ruin and Recreate Principle," *Journal of Computational Physics*, vol. 159, no. 2, pp. 139–171, Apr. 2000.
- [12] J.-F. Cordeau and G. Laporte, "The dial-a-ride problem: models and algorithms," *Annals of Operations Research*, vol. 153, no. 1, pp. 29–46, Sep. 2007.
- [13] M. W. P. Savelsbergh, "The Vehicle Routing Problem with Time Windows: Minimizing Route Duration," *ORSA Journal on Computing*, vol. 4, no. 2, pp. 146–154, May 1992, publisher: ORSA.
- [14] B. Hunsaker and M. Savelsbergh, "Efficient feasibility testing for dial-a-ride problems," *Operations Research Letters*, vol. 30, no. 3, pp. 169–173, Jun. 2002.
- [15] M. Firat and G. J. Woeginger, "Analysis of the dial-a-ride problem of Hunsaker and Savelsbergh," *Operations Research Letters*, vol. 39, no. 1, pp. 32–35, Jan. 2011.
- [16] T. Gschwind, "Route feasibility testing and forward time slack for the Synchronized Pickup and Delivery Problem," *OR Spectrum*, vol. 41, no. 2, pp. 491–512, Jun. 2019.
- [17] T. Gschwind and M. Drexl, "Adaptive Large Neighborhood Search with a Constant-Time Feasibility Test for the Dial-a-Ride Problem," *Transportation Science*, vol. 53, no. 2, pp. 480–491, Mar. 2019, publisher: INFORMS.
- [18] K. Braekers, A. Caris, and G. K. Janssens, "Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots," *Transportation Research Part B: Methodological*, vol. 67, pp. 166–186, Sep. 2014.
- [19] I. Malheiros, R. Ramalho, B. Passeti, T. Bulhões, and A. Subramanian, "A hybrid algorithm for the multi-depot heterogeneous dial-a-ride problem," *Computers & Operations Research*, vol. 129, p. 105196, May 2021.
- [20] S. Ropke and D. Pisinger, "An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows," *Transportation Science*, vol. 40, no. 4, pp. 455–472, Nov. 2006, publisher: INFORMS.
- [21] G. Dueck, "New optimization heuristics: The great deluge algorithm and the record-to-record travel," vol. 104, no. 1, pp. 86–92.
- [22] J. Christiaens and G. Vanden Berghe, "Slack Induction by String Removals for Vehicle Routing Problems," *Transportation Science*, vol. 54, no. 2, pp. 417–433, Mar. 2020.
- [23] O. Tellez, S. Vercraene, F. Lehuédé, O. Péton, and T. Monteiro, "The fleet size and mix dial-a-ride problem with reconfigurable vehicle capacity," *Transportation Research Part C: Emerging Technologies*, vol. 91, pp. 99–123, Jun. 2018.
- [24] D. Pisinger and S. Ropke, "A general heuristic for vehicle routing problems," *Computers & Operations Research*, vol. 34, no. 8, pp. 2403–2435, Aug. 2007.
- [25] J.-F. Cordeau, "A Branch-and-Cut Algorithm for the Dial-a-Ride Problem," *Operations Research*, vol. 54, no. 3, pp. 573–586, Jun. 2006, publisher: INFORMS.