# The Electric Dial-a-Ride Problem on a Fixed Circuit



DOI: 10.1287/trsc.2023.1208
Handle: http://hdl.handle.net/1942/40136

# The Electric Dial-a-Ride Problem on a Fixed Circuit[1]

Yves Molenbruch[ab], Kris Braekers[c], Ohad Eisenhandler[d], Mor Kaspi[e]

[a] Research Foundation Flanders (FWO), Brussels, Belgium

[b] Mobility, Logistics and Automotive Technology Research Centre (MOBI), Vrije Universiteit Brussel (VUB), Brussels, Belgium

[c] Research Group Logistics, UHasselt – Hasselt University, Hasselt, Belgium

[d] Department of Industrial Engineering, Afeka College of Engineering, Tel Aviv, Israel

[e] Department of Industrial Engineering, Iby and Aladar Fleischman Faculty of Engineering, Tel Aviv University, Tel Aviv, Israel

**Abstract**:

Shared mobility services involving electric autonomous shuttles have increasingly been implemented in recent years. Due to various restrictions, these services are currently offered on fixed circuits and operated with fixed schedules. This study introduces a service variant with flexible stopping patterns and schedules. Specifically, in the *Electric Dial-a-Ride Problem on a Fixed Circuit* (eDARP-FC), a fleet of capacitated electric shuttles operates on a given circuit, consisting of a recharging depot and a sequence of stations where passengers can be picked-up and dropped-off. The shuttles may perform multiple laps between which they may need to recharge. The goal of the problem is to determine the vehicles' stopping sequences and schedules, including recharging plans, so as to minimize a weighted sum of the total passenger excess time and the total number of laps. The eDARP-FC is formulated as a non-standard lap-based MILP and is shown to be NP-Hard. Efficient polynomial time algorithms are devised for two special scheduling sub-problems. These algorithms and several heuristics are then applied as sub-routines within a *Large Neighborhood Search* metaheuristic. Experiments on instances derived from a real-life system demonstrate that the flexible service results in a 32%-75% decrease in the excess time at the same operational costs.

**Keywords:**

Electric Autonomous Vehicles, Pickup and Delivery, Dial-a-Ride, Dynamic Programming

[1] This is an author accepted manuscript (AAM) version of the following paper: Molenbruch, Y., Braekers, K., Eisenhandler, O., Kaspi, M. (2023). The Electric Dial-a-Ride Problem on a Fixed Circuit. Transportation Science, available at https://doi.org/10.1287/trsc.2023.1208.

## 1. Introduction

Autonomous vehicles are a technological advancement that is expected to completely change urban mobility as we know it today. In particular, they promote the concept of mobility as a service, as opposed to private vehicle ownership, by enabling their shared utilization by multiple passengers simultaneously or at different time periods along the day. Recognizing this inevitable trend and the underlying opportunities it entails, many significant stakeholders have invested in recent years a considerable amount of resources into related research and development, such as technology companies (Google, Intel, Apple), car corporations (Daimler, GM, Toyota) and transit service providers (Uber). In parallel, this phenomenon has drawn the attention of the scientific community as it gives rise to new and challenging operational problems, in addition to technological and legal issues. See Naranayan et al. (2020) for a recent survey on shared autonomous vehicle services.

The prospect of shared mobility transit services based on autonomous vehicles introduces opportunities for reduction in cost, mileage and vehicle ownership (Shaheen and Chan, 2015), representing a potential for positive environmental and social benefit and for a major impact on transportation. In addition, most of the developed autonomous vehicles are expected to be electric, further positioning them as a sustainable transit alternative. Above all, recent studies have claimed that a high penetration of autonomous vehicles is expected to significantly improve road safety and reduce the incidence of road casualties by up to 90% (Bertoncello and Wee, 2015). Nevertheless, while fully autonomous vehicles (SAE Level 5) are expected to become commercially available in the coming decade, the market is expected to completely mature only in three to four decades (Bansal and Kockelman, 2017; Litman, 2018). Recent evidence supporting these projections include, Waymo's shift towards autonomous logistics (Alamalhodaei, 2021), Apple's setbacks in its self-driving car division (Gurman, 2021), and Uber's withdrawal from its self-driving vehicle project (Metz and Conger, 2020). The main barriers to the evolution include cost, regulatory, security, privacy and moral concerns (Fagnant and Kockelman, 2015; Bagloee et al., 2016; Bonnefon et al., 2016).

Preliminary attempts to provide shared transit services via autonomous shuttles have been carried out in multiple locations around the world during the last couple of years (Iclodean et al., 2020). Service in such systems is typically based on small fleets and in many cases is deployed for experimental purposes or due to publicity considerations. Because of the barriers mentioned above, many of these services have been designed to be provided in highly controlled environments, such as factories, resorts or campuses. Furthermore, despite the flexibility that autonomous vehicles could potentially offer, all such systems are currently reduced to providing scheduled services over circular fixed routes, see as examples the autonomous shuttle services provided in Renmark, Australia (Sage, 2020) and Sion, Switzerland (Postauto, 2020). Considering current regulations, it is reasonable to assume that in the coming years autonomous shuttle services will continue to be provided on limited segments of the road network, mainly on fixed

routes. This prospect motivates specific focus on fixed circuits and other relatively simple network structures. At the same time, existing services can already be upgraded by providing flexible services over the existing fixed routes, i.e., services that are not based on fixed stopping regimes and schedules.

In this work, we study an on-demand transit service provided by a fleet of electric autonomous shuttles that operate on a fixed unidirectional circular route. This route connects a set of stations, including a single recharging depot and several passenger stations. Figure 1 presents an illustration of a fixed unidirectional circular route.

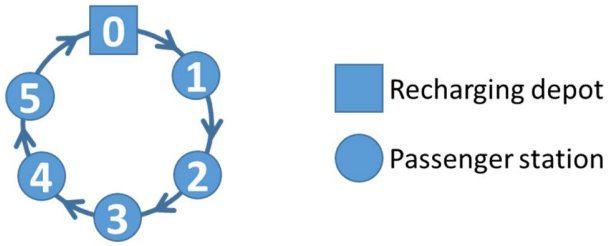


**Figure 1: Fixed unidirectional circular route with a recharging depot and five passenger stations**

During operation, the shuttles carry out multiple laps, as part of which they may perform time consuming stops at stations to allow passengers to embark and disembark. In addition, if no passengers are aboard, the shuttles may stop at the depot in order to recharge their batteries. Each shuttle is characterized by a passenger capacity and battery features, including a capacity, a discharging rate and a recharging rate. Passengers provide in advance information regarding the earliest time they would be available to be picked up at their pick-up station, i.e., their arrival time to the system, and an upper bound on the drop-off time at their destination station.

From the service provider's perspective, the underlying operational optimization problem consists of determining the number of laps each shuttle performs, the stations in which it stops in each lap, including a potential stop at the recharging depot, and the corresponding stopping schedule of each shuttle. In particular, the stopping schedule prescribes the amount of time spent recharging (waiting) during each stop at the recharging depot (stations). We refer to this problem as *The Electric Dial-a-Ride Problem on a Fixed Circuit* (eDARP-FC). The goal of the problem is to simultaneously improve the quality of service provided to the passengers and reduce the operational costs. Specifically, the quality of service is represented by the total excess time of passengers in the system compared to a direct ride at the earliest possible time, and the operational costs are represented by the total number of laps traveled by all of the shuttles.

The eDARP-FC is in its essence a special variant of the static Dial-a-Ride Problem (DARP), as it includes a set of service requests, each characterized by pickup locations, drop-off locations and corresponding time windows, to be served by designated vehicles. Molenbruch et al. (2017) and Ho et al. (2018) review the recent DARP literature, classify problem variants and survey the wide range of solution methodologies. However, it should be noted that the eDARP-FC is inherently defined by the limited

structure of the network included in it. In particular, while in the classic DARP, symmetry in the solution space is typically attributed to the possibility to swap complete routes between homogeneous vehicles, the eDARP-FC contains significantly more symmetries that result from the fact that all vehicles share the same route and traverse it multiple times. Additionally, this setting gives rise to a stronger pooling effect compared to the general DARP. Lastly, battery management considerations further add a decision type which is not prevalent in typical DARPs. These special elements of the eDARP-FC justify the consideration of new modeling and solution approaches compared to those introduced in the existing literature.

Considering the limited number of locations in which passengers can be picked up or dropped off and the pooling effect it induces, the eDARP-FC shares some resemblances to the check-point DARP (Daganzo 1984; Quadrifoglio et al., 2006). In this problem, vehicles are assumed to travel on a fixed route from which they are allowed to occasionally detour in order to pick-up or drop-off passengers at a limited set of stations given in advance. A notable difference to the setting of the eDARP-FC is that most stations will be visited at most once during the planning horizon and that multiple vehicles are not likely to repeatedly travel on the same network links.

With respect to the network structure, the work of Pimenta et al. (2017) is the only one to have studied Dial-a-Ride services on fixed circuits. The model presented in this paper considers a set of travel requests that should be served by a homogeneous fleet cycling along the fixed circuit. Each request is characterized by a pick-up station, a drop-off station, and the number of passengers to be served. The objective is to minimize the total number of stops performed by the vehicles, while respecting vehicle capacity restrictions. In comparison, the eDARP-FC considers the timing of the requests both as part of the input (passengers' time windows, stopping time at the stations) and in the objective function (passengers' excess times). As a direct consequence, the eDARP-FC presents a more complicated objective function that considers the quality of service to passengers in addition to the operational aspects. Lastly, the eDARP-FC introduces battery management aspects which are not included in the model of Pimenta et al. (2017). Very recently, Trotta et al. (2022) studied the closely-related pick-up and delivery problems on rings. They provide a classification scheme, analyze the complexity of several problem variants and present a solution approach that is based on a greedy heuristic and on mathematical programming. The objective function of the problem they investigated considers only the total number of laps. The authors highlighted the incorporation of battery management considerations as a direction for future research.

The implications of using electric vehicles in Dial-a-Ride services have recently been studied in several papers. Masmoudi et al. (2018) assume a non-linear battery consumption model and a battery swapping operating scheme, that is, vehicles can visit swapping stations to replace their depleted batteries with full ones. They formulate the problem as a non-linear Mixed Integer Linear Programming (MILP) and devise evolutionary variable neighborhood search algorithms to solve it. Bongiovanni et al. (2018) introduce the

electric Autonomous Dial-A-Ride Problem, which considers multiple origin depot locations and integrates selection of the destination depots and decisions regarding battery management within the classic DARP. They devise a branch-and-cut algorithm with new valid inequalities that arise from the electric and autonomous aspects of the problem. A dynamic variant of the problem is studied in Bongiovanni et al. (2021), in which passenger requests appear online while the system is in operation. The two stage solution approach consists of a single passenger insertion phase and a polishing phase designed to re-optimize the plans for the already accepted requests. For this phase, the authors develop a machine learning-based metaheuristic that makes use of historical information from previously solved dynamic instances to efficiently guide the search. In a wider context, for a survey of electric vehicle routing problems, see Asghari et al. (2020).

In conclusion, the eDARP-FC arises in an on-demand shared transit setting that is defined by the combination of the following three fundamental characteristics: (1) a special network structure, namely, a fixed circuit; (2) battery management decisions; and (3) time windows for the pick-up and drop-off passengers. This combination has not been studied before within a single model. By identifying, formulating and analyzing this novel operational problem, which has strong applicative grounds, we wish to close this gap in the literature.

The contribution of this paper is fivefold: first, we introduce and formulate the eDARP-FC as a non-standard, lap-based MILP. Second, we analyze the complexity of the problem and prove that it belongs to the class of NP-Hard problems. Third, we focus on two special scheduling sub-problems which assume that the sequence of pick-up/drop-off nodes is predetermined. We devise efficient polynomial time algorithms and several faster heuristics to solve them, which are applicable in general settings as the sub-problems are independent of the network structure. Fourth, we apply these solution algorithms as sub-routines within a Large Neighborhood Search (LNS) metaheuristic (Pisinger and Ropke, 2010) tailored to the structure of the eDARP-FC. We perform a numerical experiment that is based on randomly generated problem instances, as well as real-life instances derived from an autonomous shuttle service in Renmark, Australia. The results demonstrate the capability of the proposed solution approach to efficiently handle real-world scenarios. Fifth, we show that in systems that follow the setting we consider, a flexible regime has the ability to significantly improve the quality of service provided to the passengers compared to the current mode of operation, at no additional costs.

The remainder of this paper is organized as follows: in Section 2, we formally define the eDARP-FC and present a corresponding MILP formulation. In Section 3, we analyze the complexity of eDARP-FC as well as its scheduling sub-problems, and provide efficient algorithms for the latter ones. Following this, in Section 4, we present an LNS based solution approach, whose numerical analysis is then discussed in Section 5. Finally, in Section 6 we provide our conclusions and suggest directions for further research.

## 2. Problem Formulation

In this section, we formally introduce the eDARP-FC and several important sub-problems. In Section 2.1, we present a MILP formulation for the problem, which we also use later in the numerical analysis to benchmark the proposed solution approach against an exact state-of-the-art MILP solver. In Section 2.2, we define two sub-problems of the eDARP-FC. Specifically, the first sub-problem is a single-vehicle lap assignment and scheduling problem assuming that the routing, i.e., the order of requests to be served, is given. The second sub-problem further assumes that all lap assignment decisions have been made and focuses on the remaining single-vehicle scheduling problem. In Section 3 we present an efficient solution approach for the two sub-problems. This solution approach is a key component in our framework, as it enables us to evaluate exactly many routes and schedules within the LNS algorithm to be discussed in Section 4. For the convenience of the reader, all the notations introduced in these sections are summarized in Electronic Companion A.

### 2.1. A MILP formulation for the eDARP-FC

The formulation we present relies on the circular structure of the network. In particular, the operational routing decisions reduce to deciding upon the number of laps each vehicle will perform, the stations at which it will stop during each lap, and the passengers who will be served during each stop. In addition, it is assumed that passengers are dropped off at the first time the vehicle they boarded passes by their drop-off station. In other words, once a passenger is assigned to be picked up by a certain vehicle at a certain lap, the same vehicle is constrained to stop at the drop-off station. Additional sets of constraints are used to enforce sufficient battery charging levels, battery capacity, vehicle passenger capacity and time windows for serving the passengers.

**Indices:**

| | |
|---|---|
| $s$ | stations |
| $l$ | laps |
| $v$ | shuttle vehicles |
| $i$ | pick-up and drop-off nodes |

**Parameters:**

| | |
|---|---|
| $S$ | the number of stations ($s = 0$ denotes the depot, and $s = 1, \dots, S-1$ denote passenger stations, and $\mathcal{S} = \{0, .., S-1\}$) |
| $L$ | an upper bound on the number of laps per vehicle ($\mathcal{L} = \{0, \dots, L-1\}$) |
| $V$ | the number of vehicles ($\mathcal{V} = \{0, \dots, V-1\}$) |
| $t_s$ | the traveling time to station $s \in \mathcal{S}$ from the one preceding it on the circuit |
| $\theta$ | the traveling time required to complete a full lap, i.e., $\theta = \sum_{s=0}^{S-1} t_s$ |
| $n$ | the number of passenger requests |
| $\mathcal{P}$ | set of pick-up nodes ($i = 0, \dots, n-1$) |
| $\mathcal{D}$ | set of drop-off nodes ($i = n, \dots, 2n-1$), the drop-off node of pick-up node $i$ is $n+i$ |

| | |
|---|---|
| $s(i)$ | the station of pick-up/drop-off node $i$ (we assume that $s(i) \neq s(n+i)$ and $s(i) \neq 0$) |
| $a_i$ | the passenger earliest pick-up time at node $i \in \mathcal{P}$ |
| $d_i$ | the latest time the passenger(s) can be dropped off at drop-off node $i \in \mathcal{D}$ |
| $\delta(i,j)$ | the direct travel time between nodes $i, j \in \mathcal{P} \cup \mathcal{D}$ |
| $\varepsilon_i$ | the earliest arrival time to drop-off node $i \in \mathcal{D}$ |
| $q_i$ | the number of passengers embarking (disembarking) at pick-up (drop-off) node $i \in \mathcal{P}$ ($i \in \mathcal{D}$) |
| $I(i)$ | an indicator that equals 0 if $s(i+n) > s(i)$ for node $i \in \mathcal{P}$ and 1 otherwise |
| $C$ | a shuttle's passenger capacity |
| $\sigma$ | the service duration for each stop at a passenger station (regardless of the number of requests or passengers served) |
| $H$ | the horizon time |
| $\beta^c$ | battery charging rate (unit of charge per time unit) |
| $\beta^d$ | battery discharge per lap |
| $\beta^m$ | the battery's capacity in units of charge |
| $\alpha_1$ | the weight in the objective function of the total number of laps traveled by all vehicles |
| $\alpha_2$ | the weight in the objective function of the passengers' total excess times |
| $\Theta(s)$ | the set of pick-up nodes located at station $s$ ($\Theta(s) = \{i \in \mathcal{P} \mid s(i) = s\}$) |
| $\Phi^+(s)$ | the set of drop-off nodes located at station $s$ whose pick-up nodes are located at stations that precede station $s$ on the circuit ($\Phi^+(s) = \{i \in \mathcal{D} \mid s(i) = s \wedge s(i-n) < s(i)\}$) |
| $\Phi^-(s)$ | the set of drop-off nodes located at station $s$ whose pick-up nodes are located at stations that follow station $s$ on the circuit ($\Phi^-(s) = \{i \in \mathcal{D} \mid s(i) = s \wedge s(i-n) > s(i)\}$) |

The direct travel time between nodes $i, j \in \mathcal{P} \cup \mathcal{D}$ is defined as follows:

$$\delta(i,j) = \begin{cases} \sum_{k=s(i)+1}^{s(j)} t_k, & s(i) \leq s(j) \\ \sum_{k=s(i)+1}^{S-1} t_k + \sum_{k=0}^{s(j)} t_k, & s(i) > s(j) \end{cases}$$

The earliest arrival time to a drop-off node is calculated by the sum of the request earliest pick-up time, the service time at the pick-up node and the direct travel time from the pick-up node to the drop-off node, which can be precomputed using the following equation:

$$\varepsilon_i = a_{i-n} + \sigma + \delta(i-n, i) \qquad \forall i \in \mathcal{D}$$

Note that a simple upper bound on the number of laps of a single vehicle can be obtained by the following expression:

$L = \min\left(\left\lceil \frac{\max_{i \in \mathcal{D}} d_i}{\left(\theta + \sigma + \frac{\beta^d}{\beta^c}\right)} \right\rceil, 2n\right)$, in which $\theta + \sigma + \frac{\beta^d}{\beta^c}$ represents the minimal time consumed when performing a non-empty vehicle lap. The horizon time, i.e., the latest time a vehicle may return to the depot on its final lap can be bounded by $H = \theta + \max_{i \in \mathcal{D}} d_i$.

**Decision Variables:**

| | |
|---|---|
| $x_{slv}$ | equals 1 if vehicle $v \in \mathcal{V}$ stops at station $s \in \mathcal{S}$ on lap $l \in \mathcal{L}$, otherwise 0 |
| $u_{ilv}$ | equals 1 if passengers at pick-up node $i \in \mathcal{P}$ board vehicle $v \in \mathcal{V}$ on lap $l \in \mathcal{L}$, otherwise 0 |
| $w_{lv}$ | equals 1 if vehicle $v \in \mathcal{V}$ performs lap $l \in \mathcal{L}$, otherwise 0 |
| $b_{lv}$ | the amount of time that vehicle $v \in \mathcal{V}$ recharges at the beginning of lap $l \in \mathcal{L}$ |
| $B_{lv}$ | the state of charge of the battery of vehicle $v \in \mathcal{V}$ at the beginning of lap $l \in \mathcal{L}$ (before charging) |
| $T_{slv}$ | the time vehicle $v \in \mathcal{V}$ starts service (pick-up or charging) at station $s \in \mathcal{S}$ on lap $l \in \mathcal{L}$ (if no service is performed it represents the arrival time to the station) |
| $Q_{slv}$ | the amount of passengers aboard vehicle $v \in \mathcal{V}$ when it arrives at station $s \in \mathcal{S}$ on lap $l \in \mathcal{L}$ (before any service is performed) |
| $\tau_i$ | the arrival time to drop-off node $i \in \mathcal{D}$ |

Note that decision variables $T_{slv}$ do not necessarily represent the drop-off times. In cases where pick-ups and drop-offs occurs at the same stop, $T_{slv}$ represent the pick-up time while the drop-off times may occur earlier. That is, as soon as the vehicle arrives at the stop.

**Formulation:**

$$Minimize\ \alpha_1 \sum_{l=0}^{L-1} \sum_{v=0}^{V-1} w_{lv} + \alpha_2 \sum_{i \in D} q_i(\tau_i - \varepsilon_i) \tag{1}$$

$s.t.$

$$\sum_{l=0}^{L-1-I(i)} \sum_{v=0}^{V-1} u_{ilv} = 1 \qquad \forall i \in \mathcal{P} \tag{2}$$

$$u_{ilv} \leq x_{s(i),l,v} \qquad \forall i \in \mathcal{P}; \forall l \in \mathcal{L}; \forall v \in \mathcal{V} \tag{3a}$$

$$u_{ilv} \leq x_{s(i+n),l+I(i),v} \qquad \forall i \in \mathcal{P}; \quad \forall l \in \{0, \dots, L-1-I(i)\}; \forall v \in \mathcal{V} \tag{3b}$$

$$x_{slv} \leq w_{lv} \qquad \forall s \in \mathcal{S}; \forall l \in \mathcal{L}; \forall v \in \mathcal{V} \tag{4}$$

$$w_{lv} \leq w_{l-1,v} \qquad \forall l \in \mathcal{L}\backslash\{0\}; \forall v \in \mathcal{V} \tag{5}$$

$$B_{lv} = B_{l-1,v} + \beta^c b_{l-1,v} - \beta^d w_{l-1,v} \qquad \forall l \in \mathcal{L}\backslash\{0\}; \forall v \in \mathcal{V} \tag{6a}$$

$$B_{L-1,v} + \beta^c b_{L-1,v} - \beta^d w_{L-1,v} \geq 0 \qquad \forall v \in \mathcal{V} \tag{6b}$$

$$B_{0v} = 0 \qquad \forall v \in \mathcal{V} \tag{7}$$

$$\beta^c b_{lv} \leq \beta^m x_{0lv} \qquad \forall l \in \mathcal{L}; \forall v \in \mathcal{V} \tag{8}$$

$$B_{lv} + \beta^c b_{lv} \leq \beta^m \qquad \forall l \in \mathcal{L}; \forall v \in \mathcal{V} \tag{9}$$

$$Q_{s0v} = Q_{s-1,0,v} + \sum_{i \in \Theta(s-1)} q_i u_{i0v} - \sum_{i \in \Phi^+(s-1)} q_i u_{i0v} \qquad \forall s \in \mathcal{S}\backslash\{0\}; \forall v \in \mathcal{V} \tag{10a}$$

$$Q_{slv} = Q_{s-1,l,v} + \sum_{i \in \Theta(s-1)} q_i u_{ilv} - \sum_{i \in \Phi^+(s-1)} q_i u_{ilv} - \sum_{i \in \Phi^-(s-1)} q_i u_{i,l-1,v} \qquad \forall s \in \mathcal{S}\backslash\{0\}; \forall l \in \mathcal{L}\backslash\{0\}; \forall v \in \mathcal{V} \tag{10b}$$

$$Q_{1lv} = Q_{0,l,v} \qquad \forall l \in \mathcal{L}; \forall v \in \mathcal{V} \tag{10c}$$

| | | |
|---|---|---|
| $Q_{0lv} = Q_{S-1,l-1,v} + \sum_{i \in \Theta(S-1)} q_i u_{i,l-1,v} - \sum_{i \in \Phi^+(S-1)} q_i u_{i,l-1,v}$ | $\forall l \in \mathcal{L}; \forall v \in \mathcal{V}$ | (10d) |
| $Q_{00v} = 0$ | $\forall v \in \mathcal{V}$ | (11) |
| $Q_{slv} \leq C$ | $\forall s \in \mathcal{S}; \forall l \in \mathcal{L}; \forall v \in \mathcal{V}$ | (12) |
| $Q_{0lv} \leq C(1 - x_{0lv})$ | $\forall l \in \mathcal{L}; \forall v \in \mathcal{V}$ | (13) |
| $T_{0lv} \geq T_{S-1,l-1,v} + \sigma x_{S-1,l-1,v} + t_0$ | $\forall l \in \mathcal{L} \backslash \{0\}; \forall v \in \mathcal{V}$ | (14a) |
| $T_{1lv} \geq T_{0,l,v} + b_{lv} + t_1$ | $\forall l \in \mathcal{L}; \forall v \in \mathcal{V}$ | (14b) |
| $T_{slv} \geq T_{s-1,l,v} + \sigma x_{s-1,l,v} + t_s$ | $\forall s \in \mathcal{S} \backslash \{0\}; \forall l \in \mathcal{L};\ \forall v \in \mathcal{V}$ | (14c) |
| $T_{00v} = 0$ | $\forall v \in \mathcal{V}$ | (15) |
| $T_{s(i),l,v} \geq a_i \sum_{l'=0}^{l} u_{i,l',v}$ | $\forall i \in \mathcal{P}; \forall l \in \mathcal{L}; \forall v \in \mathcal{V}$ | (16) |
| $\tau_i \geq T_{s(i)-1,l+I(i-n),v} + \sigma x_{s(i)-1,l+I(i-n),v} + t_{s(i)} - H(1 - u_{i-n,l,v})$ | $\forall i \in \mathcal{D}: s(i) \neq 1; \forall v \in \mathcal{V};$ $\forall l \in \{0, \dots, L - 1 - I(i)\}$ | (17a) |
| $\tau_i \geq T_{0,l+1,v} + t_1 - H(1 - u_{i-n,l,v})$ | $\forall i \in \mathcal{D}: s(i) = 1;$ $\forall l \in \mathcal{L} \backslash \{L-1\}; \forall v \in \mathcal{V}$ | (17b) |
| $\tau_i \leq d_i$ | $\forall i \in \mathcal{D}$ | (18) |
| $w_{lv} \in \{0,1\}$ | $\forall l \in \mathcal{L}, \forall v \in \mathcal{V}$ | (19) |
| $x_{slv} \in \{0,1\}$ | $\forall s \in \mathcal{S}; \forall l \in \mathcal{L}; \forall v \in \mathcal{V}$ | (20) |
| $u_{ilv} \in \{0,1\}$ | $\forall i \in \mathcal{P};\ \forall l \in \mathcal{L};\ \forall v \in \mathcal{V}$ | (21) |
| $\tau_i \geq 0$ | $\forall i \in \mathcal{D}$ | (22) |
| $B_{lv}, b_{lv} \geq 0$ | $\forall l \in \mathcal{L};\ \forall v \in \mathcal{V}$ | (23) |
| $T_{slv}, Q_{slv} \geq 0$ | $\forall s \in \mathcal{S};\ \forall l \in \mathcal{L}; \forall v \in \mathcal{V}$ | (24) |

The objective function (1) minimizes a weighted sum of the total number of laps traveled by the shuttles and the total passenger excess time. Constraints (2)-(5) are assignment constraints guaranteeing that each passenger request is assigned to exactly one vehicle lap, that each vehicle stops at the stations and laps that correspond to the requests it serves, and that the vehicles only stop on laps that they perform. Constraints (6)-(9) are battery management constraints representing charge conservation, guaranteeing that charging can only take place if the vehicle stops at the depot, and asserting that the level of charge remains between zero and the battery capacity. Constraints (10)-(13) are vehicle capacity constraints including passenger flow conservation and vehicle capacity enforcement. In particular, Constraints (13) state that the vehicles can stop at the depot, and consequently that their batteries can be charged, only if there are no passengers on board. Constraints (14)-(18) are timing constraints enforcing the time windows of the requests and determining the drop-off time of each request, according to the arrival time at the drop-off station. Variable integrality and non-negativity are set in constraints (19)-(24). Finally, note that the above formulation can

be strengthened by adding a number of valid but redundant constraints. Several of such constraints are proposed in Electronic Companion B and applied in the numerical experiments we present in Section 5.

### 2.2. Lap assignment and scheduling for a given node sequence

In this section, we focus on a sub-problem, in which the assignment of passengers to vehicles and the order in which they will be served is known, while stops at the recharging depot still need to be determined. The remaining decisions concern determining the vehicle lap in which each request node will be served and the vehicle schedules, including the recharging stops and amounts. It is assumed and can be efficiently validated that the sequence satisfies all other conditions asserted by the above model (pairing, precedence, capacity). Noting that this sub-problem decomposes to vehicle-specific sub-problems, we next present the resulting *single-vehicle lap assignment and scheduling sub-problem*. Recall that we focus on this sub-problem because an efficient solution approach for it will be used many times as a subroutine within our LNS algorithm. Next, we provide a simplified notation of this sub-problem that is based on the position of the items in the given sequence:

| | |
|---|---|
| $n'$ | the number of passenger requests served by the vehicle |
| $\Psi$ | the given sequence of pickup and drop-off nodes ($j = -1$ and $j = 2n'$, representing vehicle start and end depot) |
| $\mathcal{P}_\Psi$ | the subset of items in the sequence representing pickup nodes |
| $\mathcal{D}_\Psi$ | the subset of items in the sequence representing drop-off nodes |
| $\hat{s}_j$ | the station associated with item $j$ |
| $a_j$ | the lower time window associated with the node represented by item $j \in \mathcal{P}_\Psi$ |
| $d_j$ | the upper time window associated with the node represented by item $j \in \mathcal{D}_\Psi$ |
| $\varepsilon_j$ | the earliest arrival time to the drop-off node represented by item $j \in \mathcal{D}_\Psi$ |
| $e_j$ | equals 1 if item $j$ is served in the same station as item $j-1$, 0 otherwise |
| $\lambda_j$ | the cumulative number of laps completed since travelling from the depot up to the station associated with item $j$ |
| $t_j$ | the traveling time to item $j$ from the one preceding it in the sequence |
| $q_j$ | the number of passengers embarking/disembarking at item $j$ |

Lastly, we set the following values: $\hat{s}_{-1} = S, \lambda_{-1} = 0, e_{-1} = 1, t_0 = 0, a_j = 0 \; \forall j \in \mathcal{D}_\Psi$, $d_j = H \; \forall \, j \in \mathcal{P}_\Psi$ .

We denote by $\mathcal{R} \subseteq \mathcal{P}_\Psi$ the subset of pickup nodes to which the vehicle arrives with no passengers on board, thus potentially allowing for recharging to take place prior to serving the pickup node. We refer to these pickup nodes as recharging opportunities, and distinguish between two types: type (0), in which the vehicle passes anyway by the recharging depot on its way to the pickup node; and type (+), in which the vehicle bypasses the pickup node and travels specially to the recharging depot, thus adding an extra lap to the route. Formally, let $\mathcal{R}^0 = \left\{j \in \mathcal{R}: \hat{s}_{j-1} > \hat{s}_j\right\}$ and $\mathcal{R}^+ = \left\{j \in \mathcal{R}: \hat{s}_{j-1} \leq \hat{s}_j\right\}$ be the subset of recharging opportunities of types (0) and (+), respectively. Note that $\mathcal{R}^0 \cap R^+ = \emptyset$ and $\mathcal{R}^0 \cup \mathcal{R}^+ = \mathcal{R}$. Furthermore, note that as recharging opportunities of type (0) do not require additional traveling, we only need to make

sure that if recharging takes place, the recharging time is taken into account in the schedule. However, recharging opportunities of type (+) come with extra traveling distance (i.e., a full lap), extra battery discharge because of the added lap and extra travel time that should be directly accounted for while formulating the sub-problem. The lap assignment and scheduling sub-problem can then be formulated as a MILP, using the following decision variables:

**Decision Variables:**

$y_j$ equals 1 if a type (+) recharging opportunity is taken in item $j = 0, \dots, 2n' - 1$, otherwise 0
$G_j$ the accumulative recharging time up to item $j = 0, \dots, 2n' - 1$ in the sequence ($G_{-1} = 0$)
$T_j$ the time the vehicle arrives to item $j = 0, \dots, 2n' - 1$ in the sequence ($T_{-1} = 0$)

This leads to the following formulation:

$$Minimize\ \alpha_1 \sum_{j \in \mathcal{R}^+} y_j + \alpha_2 \sum_{j \in \mathcal{D}_\Psi} q_j (T_j - \varepsilon_j) \tag{25}$$

$s.t.$

$$a_j \leq T_j \qquad \forall j \in \mathcal{P}_\Psi \tag{26}$$

$$d_j \geq T_j \qquad \forall j \in \mathcal{D}_\Psi \tag{27}$$

$$T_j \geq T_{j-1} + \sigma\left(1 - e_{j-1}\right) + \sigma e_{j-1} y_{j-1} + t_j \qquad \forall j \in (\mathcal{P}_\Psi \backslash \mathcal{R}) \cup \mathcal{D}_\Psi \tag{28}$$

$$T_j \geq T_{j-1} + \sigma\left(1 - e_{j-1}\right) + G_j - G_{j-1} + \sigma e_{j-1} y_{j-1} + t_j \qquad \forall j \in \mathcal{R}^0 \tag{29}$$

$$T_j \geq T_{j-1} + \sigma\left(1 - e_{j-1}\right) + G_j - G_{j-1} + \sigma e_{j-1} y_{j-1} + t_j + \theta y_j \qquad \forall j \in \mathcal{R}^+ \tag{30}$$

$$G_j - G_{j-1} \leq \frac{\beta^m}{\beta^c} y_j \qquad \forall j \in \mathcal{R}^+ \tag{31}$$

$$G_j - G_{j-1} = 0 \qquad \forall j \in (\mathcal{P}_\Psi \backslash \mathcal{R}) \cup \mathcal{D}_\Psi \tag{32}$$

$$G_{j-1} \geq \frac{\beta^d}{\beta^c} \left( \lambda_j + \sum_{k \in R^+ : k \leq j} y_k \right) \qquad \forall j \in \mathcal{R} \cup \{2n'\} \tag{33}$$

$$G_j \leq \frac{\beta^m + \beta^d \left(\lambda_j + \sum_{k \in \mathcal{R}^+ : k \leq j} y_k\right)}{\beta^c} \qquad \forall j \in \mathcal{R} \tag{34}$$

$$y_j = 0 \qquad \forall j \in (\mathcal{P}_\Psi \cup \mathcal{D}_\Psi) \backslash \mathcal{R}^+ \tag{35}$$

$$G_j, T_j \geq 0 \qquad \forall j \in \mathcal{P}_\Psi \cup \mathcal{D}_\Psi \tag{36}$$

$$y_j \in \{0,1\} \qquad \forall j = -1, \dots, 2n' \tag{37}$$

The objective function (25) is the equivalent representation of (1) to the nodes included in the given sequence. Constraints (26)-(27) guarantee that the service times at pick-up/drop-off nodes satisfy the lower/upper bounds. Constraints (28)-(30) make sure that arrival times to successive items in the sequence respect travel durations, service durations, and recharging times. Note that the term $G_j - G_{j-1}$ represents the recharging duration in recharging opportunity $j$, and that the addition of a full lap is also accounted for in Constraints (30). Constraints (31) state that if recharging opportunity of type (+) $j$ is not used, then there is no additional recharging. Constraints (32) stipulate that the accumulative recharging time at any item in the sequence that does not allow recharging will be equal to the recharging time at the item that precedes it. Constraints (33) enforce lower bounds on the total amount recharged up to recharging opportunity $j$.

This amount should be sufficient to travel the total number of laps up to this recharging opportunity, including extra laps. Constraints (34) provide an upper bound on the total amount that can be recharged up to and including recharging opportunity $j$. Constraints (35) set the value of the $y_j$ variables to equal zero at any item that does not belong to $R^+$. Constraints (36)-(37) are non-negativity and integrality constraints.

Once decisions regarding the recharging opportunities (+) are taken (represented by the $y_j$ variables), the route of the vehicle is fully known. In particular, the first component of the objective function becomes a constant. We refer to the remaining sub-problem as the *single-vehicle scheduling sub-problem*, under which the service start time and recharging times need to be determined in a manner that minimizes the total excess time of the passengers. A straightforward LP formulation of the scheduling sub-problem is obtained by fixing the values of the $y_j$ variables $\forall j \in \mathcal{R}^+$ in the MILP (25)-(37).

## 3. Complexity Analysis

This section analyses the complexity of the eDARP-FC and its sub-problems. First, we prove the problem is NP-Hard considering separately each of the components of the objective function. Second, we characterize the settings under which the single-vehicle lap assignment and scheduling sub-problem can be solved in pseudo-polynomial time using a Dynamic Programming (DP) algorithm. Lastly, we note that the single-vehicle scheduling sub-problem can be solved efficiently under any setting.

The assignment of requests with multiple passengers into vehicles has been shown to be a complicating factor (Pimenta et al., 2017). Similar to that study, a straightforward reduction from the *Bin Packing Problem* can be used to show that the eDARP-FC is also NP-Hard. Thus, we next focus our discussion on cases with single-passenger requests.

**Claim 1:** Minimizing the number of laps for the eDARP-FC with single-passenger requests is NP-Hard.

**Claim 2:** Minimizing the passengers' total excess time for the eDARP-FC with single-passenger requests is NP-Hard.

**Claim 3:** the eDARP-FC is NP-Hard.

**Proof 1:** Consider an instance of the eDARP-FC with the following characteristics: (1) all requests are unit size, that is, $q_i = 1 \ \forall i = 1, \dots, n$; (2) request time windows are non-binding, that is, $a_i = 0$ and $d_i = \infty$ $\forall i = 1, \dots, n$; (3) a single capacitated vehicle is available; (4) service does not require any time, that is, $\sigma = 0$; (5) the battery recharging is non-binding, that is, $\beta^m = \infty, \beta^d = 0$ and $\beta^c = 0$; and (6) the objective is to minimize the total number of laps, that is, $\alpha_1 = 1, \alpha_2 = 0$. The setting of this special case was shown to be NP-Hard (Theorem 3 in Trotta et al., 2022) when minimizing the time the vehicle returns to the depot after serving all requests. Note that under this setting minimizing this objective is fully equivalent to minimizing the total number of laps performed by the vehicle. ■

**Proof 2:** We claim that under the setting of the special case described in Proof 1, minimizing the passengers' total excess time also minimizes the total number of laps, therefore establishing the NP-hardness of this setting as well. To see this, suppose, by contradiction, that these two measures have different optimal solutions, denoted as Solution E (for excess) and Solution L (for laps). In particular, in Solution E more laps are performed as compared to Solution L. We will demonstrate that if this is the case, Solution E does not minimize the total excess time, leading to a contradiction. For this purpose, we take Solution E and apply the following procedure: first (Step 1), we select a request that is served in the last lap and shift it to the lap to which it is assigned under Solution L (denoted as X), which is necessarily an earlier lap. If this move results with a feasible solution, this solution has a lower total excess time than Solution E, as a single request was shifted earlier in time. Otherwise (Step 2), we select a request from lap X that is not assigned to it in Solution L and shift this request to the last lap of Solution E. Note that the combination of Step 1 and Step 2 does not change the total excess time. We repeat these two steps until a feasible solution is found at the end of Step 1. Such a solution is guaranteed to be found due to the existence of Solution L. This shows that minimizing the total excess time is at least as hard as minimizing the number of laps. ■

**Proof 3:** follows directly from the combination of Claim 1 and Claim 2.

Next, we analyze the complexity of the lap assignment and scheduling sub-problem. In particular, we will characterize settings under which this sub-problem can be solved with a polynomial time DP algorithm. In order to demonstrate how the sub-problem can be solved with a DP algorithm, we first wish to show that under certain input characteristics, it has an integral optimal solution.

**Definition 1:** in *integral instances* of the lap assignment and scheduling sub-problem or the scheduling sub-problem the following parameters have integral values: $a_j$ $(\forall j \in \mathcal{P}_\Psi)$; $d_j$ $(\forall j \in \mathcal{D}_\Psi)$; $t_j$ $(\forall j \in \mathcal{P}_\Psi \cup \mathcal{D}_\Psi)$; $\sigma_j$ $(\forall j \in \mathcal{P}_\Psi \cup \mathcal{D}_\Psi)$, $\theta$ and the ratios $\frac{\beta^d}{\beta^c}$ and $\frac{\beta^m}{\beta^c}$.

**Definition 2:** in an *integral solution* of the lap assignment and scheduling sub-problem or the scheduling sub-problem, all arrival times to the nodes in the sequence and all recharging times are integral.

Note that in many realistic scenarios, time windows, travel times, service times and the lap completion time can be rescaled to a time unit for which all of these values are integral (e.g. seconds or minutes). However, while the ratios $\frac{\beta^d}{\beta^c}$ and $\frac{\beta^m}{\beta^c}$ are not necessarily integral, when they are, we can claim the following:

**Claim 4:** The coefficient matrix of the scheduling sub-problem is totally unimodular (TU).

**Claim 5:** An integral instance of the scheduling sub-problem has an integral optimal solution.

**Claim 6:** An integral lap assignment and scheduling sub-problem has an integral optimal solution.

**Proof 4:** By carefully examining the structure of the scheduling sub-problem formulation (i.e., the MILP (25)-(36) with the $y_j$ variables fixed to an appropriate value), we show that it is characterized by a constraint

coefficient matrix which is totally unimodular (TU). The proof of this property relies on the sufficient condition due to Ghouila-Houri (see Padberg, 1976), by which a matrix A is TU if each collection of its columns can be split into two parts so that the sum of the columns in one part minus the sum of the columns in the other part is a vector with entries only 0, +1 or -1. For the sake of brevity, we provide the details of this proof in Electronic Companion C.

**Proof 5**: Consider a formulation for the scheduling sub-problem obtained by fixing the values of the $y_j$ variables $\forall j \in \mathcal{R}^+$ in the MILP (25)-(36). This formulation can be standardized to having only smaller-equal inequalities with all the decision variables on the left hand side. In particular, by Definition 1, in any integral instance of the problem, the constant coefficients on the right hand side are all strictly integral. In addition, since by Claim 4 the coefficient matrix of the resulting formulation is TU, it follows that an integral instance of the scheduling sub-problem has an integral optimal solution. ■

**Proof 6**: Follows straightforwardly from Claim 5. For any feasible assignment of binary values to the $y_j$ variables $\forall j \in \mathcal{R}^+$, we obtain an integral instance of the scheduling sub-problem, which has an integral optimal solution by Claim 5. In particular, the scheduling sub-problem resulting from the optimal assignment of binary values to the $y_j$ variables $\forall j \in \mathcal{R}^+$, has an integral optimal solution as well. ■

A direct consequence of Claim 6 is that the search for an optimal solution of an integral instance of the lap assignment and scheduling sub-problem can be restricted to the set of integral solutions. This allows us to devise a pseudo-polynomial time DP algorithm that while considering only integral states and integral decision values, is still guaranteed to obtain an optimal solution.

**Observation 1**: as the earliest pick-up time of the passengers is fixed, in order to minimize their total excess times, the service start times at the drop-off nodes should be scheduled as early as possible. Therefore, once recharging is completed, service at the following nodes (up to the next recharging opportunity) should start as early as possible.

As a result, the only decisions that impact the quality of the schedule concern the recharging durations at each recharging opportunity. We denote a state of the vehicle by the quartet $(j, T, B, h)$ representing the current item $j$ the vehicle is visiting, the time $T$ at which service begins at that item, the current state of charge $B$, and whether a recharging opportunity of type (+) was used ($h = 1$) or not ($h = 0$) before item $j$. The decision associated with each such state refers to the amount of time spent recharging the vehicle after service in item $j$ (or put differently, before arriving to item $j + 1$), and therefore depends on the node type of item $j + 1$. In this context, we distinguish between two cases. First, at states $(j, T, B, h)$ having item $j + 1 \in (\mathcal{P}_\Psi \backslash \mathcal{R}) \cup \mathcal{D}_\Psi$, no recharging can take place, and therefore no decision needs to be made. In this case, given Observation 1, the state of the vehicle at the following item is straightforwardly implied. Second, at states $(j, T, B, h)$ having item $j + 1 \in \mathcal{R}$, we may decide the recharging duration, which must take an integral value according to Claim 6. Therefore, we denote by $\mathbb{Z}(j, B) = z^L(j, B), \dots, z^U(j, B)$ the range of

possible positive integral recharging durations, when the vehicle is in state $(j, T, B, h)$ such that $j+1 \in \mathcal{R}$. The boundaries $z^L(j,B), z^U(j,B)$ are defined as follows:

$$z^L(j,B) = \max\left\{1, \frac{1}{\beta^c} \cdot \left(\beta^d \left(\lambda_{next(j+1)} - \lambda_{j+1} + 1\right) - B\right)\right\}$$

$$z^U(j,B) = \min\left\{\max\left\{0, \frac{1}{\beta^c} \cdot \left(\beta^d\left(\lambda_{2n'} - \lambda_{j+1} + 1\right) - B\right)\right\}, \frac{1}{\beta^c} \cdot \left(\beta^m - B + \beta^d\right)\right\}$$

$$z^U(-1,B) = \min\left\{\max\left\{0, \frac{1}{\beta^c} \cdot \left(\beta^d \lambda_{2n'} - B\right)\right\}, \frac{1}{\beta^c} \cdot \left(\beta^m - B\right)\right\}$$

where $next(j) = \min_{k \in R}\{k | k > j\}$ denotes the recharging opportunity that follows recharging opportunity $j$. The lower bound $z^L(j,B)$ is the additional recharging duration that guarantees sufficient battery to arrive at the following recharging opportunity. If the vehicle can reach the following recharging opportunity without additional recharge, the lower bound is set to a single time unit. The upper bound $z^U(j,B)$ is the minimum of the recharging duration that guarantees sufficient battery to arrive at the end depot and the recharging duration up to the battery capacity. In all other cases, i.e. $j+1 \in (\mathcal{P}_\Psi \backslash \mathcal{R}) \cup \mathcal{D}_\Psi$, we set $\mathbb{Z}(j,B) = \emptyset$.

Next we introduce the cost function $F(j,T,B,h)$, representing the minimal weighted sum of the total excess times of passengers served in items $j+1, \ldots, 2n'$ and the number of laps traveled in order to serve them, given that the vehicle is in state $(j,T,B,h)$. This function can be defined recursively using the following Bellman equation:

$$F(j,T,B,h) = \min_{z \in \{0\} \cup \mathbb{Z}(j,B)} F\left(j+1, \max\{a_{j+1}, \Pi(j,T,z,h)\}, \Omega(j,B,z), \mathbb{I}(j,z)\right) + \alpha_1 \mathbb{I}(j,z) + \alpha_2 \mathbb{D}(j,h,T)$$

Where $\mathbb{I}(j,z)$ defines whether a recharging opportunity of type (+) is to be used before item $j+1$, $\Pi(j,T,z,h)$ defines the arrival time to item $j+1$, $\Omega(j,B,z)$ defines the state of charge at item $j+1$, and $\mathbb{D}(j,h,T)$ calculates the excess time incurred by a request dropped-off at item $j+1$. These functions are mathematically defined as follows:

$$\mathbb{I}(j,z) = \begin{cases} 1, & j+1 \in \mathcal{R}^+ \; and \; z > 0 \\ 0, & Otherwise \end{cases}$$

$$\Pi(j,T,z,h) = T + \sigma\left(1 - e_j\right) + \sigma h e_j + z + t_{j+1} + \theta \mathbb{I}(j,z)$$

$$\Omega(j,B,z) = B + z\beta^c - \beta^d \left(\lambda_{j+1} - \lambda_j + \mathbb{I}(j,z)\right)$$

$$\mathbb{D}(j,h,T) = \begin{cases} q_{j+1} \cdot \left(T + \sigma\left(1 - e_j\right) + \sigma h e_j + t_{j+1} - \varepsilon_{j+1}\right), & j+1 \in D_\Psi \\ 0 & , \quad Otherwise \end{cases}$$

The structure of the Bellman equation is explained next by referring to the decision space, the state transition and the instantaneous cost function. Positive recharging decisions are enabled only at items representing recharging opportunities, and the recharging amount can be selected from the range $\mathbb{Z}(j,B)$ as defined

above, or zero. The arrival time to item $j+1$ is set to be the latest between its lower time window and the arrival time of the vehicle to that item, as defined by $\Pi(j,T,z,h)$ which accounts for the arrival time to item $j$, and adds the travel time, service time, recharging time and potentially an additional lap time. The state of charge at item $j+1$ as defined by $\Omega(j,B,z)$ accounts for the state of charge at item $j$, adds the chosen amount to be recharged and subtracts the amount discharged since item $j$ which may include the discharge incurred due to an additional lap performed using recharging opportunity of type (+). The updated value of state variable $h$ is by definition the value of $\mathbb{I}(j,z)$. Lastly, the instantaneous cost consists of two components: the first represent an additional lap to be accumulated in case a recharging opportunity of type (+) is used; the second component represents the excess time accumulated in items representing drop-offs.

An optimal solution of the sub-problem is obtained by calculating the cost $F(-1,0,0,0)$, which represents the minimal total weighted cost of a vehicle located at the start depot at the beginning of the planning period with an empty battery. An end condition is applied, representing the cost of the last lap:

$$F\left(2n'-1,T,\beta^d,0\right)=\alpha_1 \ \forall T \ \leq d_{2n'-1}$$

In addition, the following boundary conditions are applied to prevent exploration of infeasible solutions:

- $F\left(2n'-1,T,\beta^d,0\right)=\infty \ \forall T > d_{2n'-1}$
- $F(j,T,B,0)=\infty \quad \forall T > d_j \ \forall B \ \ \forall j \in D_\Psi$
- $F(j,T,B,h)=\infty \quad \forall j+1 \in \mathcal{R}, \forall B < \beta^d \ \ \forall T, \forall h \in \{0,1\}$

The first condition refers to states in which the maximal drop-off time in the last item of the sequence is violated; the second conditions enforces the same restriction with respect to any preceding item; the third condition forbids states without sufficient charge to reach the following recharging opportunity; and the fourth condition excludes states in which the minimal and maximal recharging amount contradict.

A pseudocode summarizing the DP algorithm for the integral lap assignment and scheduling sub-problem is presented in Electronic Companion D. Note that the algorithm traverses at most $O(n')$ items, the service at each item may begin during at most $O\left(\frac{\beta^m}{\beta^c}\cdot n'\right)$ time units, the state of charge may reach at most $O\left(\frac{\beta^m}{\beta^c}\right)$ different levels, and the state variable $h$ can take $O(1)$ different values. Therefore, at most $O\left(\left(\frac{\beta^m}{\beta^c}\right)^2\cdot n'^2\right)$ states may be reached. At each of these states, at most $O\left(\frac{\beta^m}{\beta^c}\right)$ recharging decisions may be considered. Given a recharging decision, the following state and the contribution to the cost function can be both calculated in constant time. To conclude, the worst-case time complexity of the proposed DP algorithm is $O\left(\left(\frac{\beta^m}{\beta^c}\right)^3\cdot n'^2\right)$, which is pseudo-polynomial as it depends not only on the number of requests $n'$, but also on the time to fully recharge an empty battery, $\frac{\beta^m}{\beta^c}$.

As a final remark, we note that we were not able to demonstrate that in the general case, the lap assignment and scheduling sub-problem can be solved in pseudo-polynomial time. In contrast, as the scheduling sub-problem can be formulated as an LP, it can be solved efficiently in the general case as well.

## 4. A Large Neighborhood Search Heuristic for the eDARP-FC

As demonstrated in the previous section, the e-DARP-FC in its general definition is NP-Hard. Therefore, to generate good solutions for large instances of the problem within short running times, a heuristic approach is needed. Notably, the structure of the problem suggests that good solutions to the problem may be similar in terms of the vehicle and lap assignments. In particular, the sequence of requests that a vehicle serves is likely to correspond to the served requests' earliest pick-up times as well as the physical locations of their origin and destination stations on the circuit. For example, requests that have early earliest pick-up times are not likely to appear at the ends of the vehicles' visit sequences. Therefore, a heuristic approach that is based on neighborhood search principles seems to be adequate.

The analysis of the single-vehicle lap assignment and scheduling sub-problem highlights that the impact of local modifications to a given solution cannot be estimated in advance without fully evaluating the resulting solution. That is, a minor change in a vehicle's sequence of visits may result with a new optimal schedule that greatly differs from the original one. In other words, one cannot benefit from the so called "delta calculations" that are often utilized in local search heuristics. Nevertheless, the DP algorithm and additional scheduling heuristics developed for this sub-problem (Section 4.2) permit fast evaluations of complete sequences, which allows us to incorporate them within an LNS framework. In what follows, we present an overview of the LNS framework for the eDARP-FC. The main elements of the framework are then further described in the following subsections.

Our encoding represents a solution for the eDARP-FC by a sequence of passenger nodes for each vehicle. Each of these sequences can then be evaluated independently by solving the single-vehicle lap assignment and scheduling sub-problem. The overall evaluation of a solution is the sum of the single-vehicle evaluations. In the LNS, the sequences of a given solution are modified in search for an improving solution, by removing some requests and reinserting them in new positions of the solution. Each newly obtained single-vehicle sequence is reevaluated using either the DP algorithm or the scheduling heuristics, and then aggregated with the evaluations of the other vehicles to obtain the value of the new solution. In Table 1, we present the main steps of the LNS framework.

**Table 1: LNS framework**

1. Initialize a solution $s^{curr}$
2. $s^{best} = s^{curr}$
3. Repeat:
    4. $s^{new} = repair\big(destroy(s^{curr})\big)$
    5. **If** $accept(s^{new}, s^{curr})$, **then** $s^{curr} = s^{new}$

6. **If** $new\_best(s^{new}, s^{curr})$ **then** $s^{best} = s^{new}$
7. **Until** running time is exhausted
8. **Return** $s^{best}$

The algorithm applies an initialization procedure to construct a solution (**step 1**) and sets it as the initial incumbent solution (**step 2**). Such a solution may be incomplete in the sense that not all requests are included in it. In this case, we refer to the set of non-included requests as the *request bank*. The main loop of the LNS framework iteratively destroys and repairs solutions by removing some requests from the solution, moving them to the bank, and then trying to reinsert all of the requests that are in the bank. At each iteration, the algorithm selects a single destroy operator and a single repair operator to be applied, out of a predefined set of operators (**step 4**). If the newly obtained solution passes some acceptance criterion the algorithm shifts to the new solution (**step 5**). If the newly obtained solution is better than the incumbent solution, it is stored as the new incumbent solution (**step 6**). The algorithm terminates after a predefined calculation time is exhausted (**step 7**) and returns the best solution found (**step 8**).

### 4.1. Solution destruction

At each iteration of the LNS, the proportion of requests to be removed from the current solution is drawn from a uniform distribution $\pi \sim U[0,0.4]$. The framework uniformly selects a destroy operator from the following set of operators and applies it until $\hat{n} = \lceil \pi(n - n_b) \rceil$ requests are removed, where $n_b$ denotes the number of requests currently in the bank.

- **Random request removal**: randomly selects a request to be removed.
- **Shaw removal**: first removes one random request. Next, it continuously removes from the solution the request which is most similar to the initially removed request. In this context, the similarity between a pair of requests is defined with respect to the difference between request earliest pick-up times, as well as the travel time between the stations associated with the requests. In particular, the similarity $\eta(i,j)$ between requests $i, j \in P$ is defined as follows:

$$\eta(i,j) = 1 - 0.5\left(\frac{|a_i - a_j|}{\max_{k,l \in \mathcal{P}} |a_k - a_l|} + \frac{\psi(i,j)}{\max_{k,l \in \mathcal{P}} \psi(k,l)}\right)$$

  Where $\psi(i,j)$ is defined as the minimal direct travel time between the requests' $i, j \in P$ pick-up nodes or drop-off nodes, Namely:

$$\psi(i,j) = \min\big(\delta(i,j), \delta(j,i), \delta(i+n, j+n), \delta(j+n, i+n)\big)$$

  Note that the similarity measure is independent of the solution and therefore can be calculated for all pairs of requests as a preprocessing step.
- **Latest request removal**: removes the request which is, in the current schedule, delivered closest to the upper TW of the drop-off node.

- **Random lap removal**: removes all requests in a randomly selected vehicle lap.
- **Related lap removal**: first selects a random vehicle lap and removes all requests included in it. Then, it continuously removes all requests in the lap (in any vehicle) whose service end time at the recharging depot is closest in time to the initially removed lap.
- **Worst lap removal**: removes all requests in the vehicle lap serving the lowest number of requests.

### 4.2. Solution reconstruction

The reconstruction of a solution follows several steps. First, we determine the order according to which the requests in the bank will be reinserted. Second, we iterate over the requests in the bank in that order and attempt to reinsert them. For each request, we screen out positions in the vehicle sequences to which it cannot be inserted due to capacity and time-window considerations. Third, the remaining candidate positions are exhaustively explored by evaluating their resulting schedules. Finally, the best insertion position is selected. If there are no candidate positions for a request, it remains in the bank.

For the first step, we uniformly select and apply one of the following repair strategies:

- **Random order best insertion**: sort the requests in random order.
- **Time-based best insertion**: sort the requests according to their earliest pick-up times (ascending).

At the second step we screen out infeasible insertion positions using three types of considerations, namely, simple time-window considerations, vehicle load considerations and the requirement that no passenger completes a full lap on board a vehicle (the screening-out procedure is described in details in Electronic Companion E).

Next, at the third step, for each of the remaining candidate positions, we solve the single-vehicle lap assignment and scheduling sub-problem to determine the impact of inserting the request in this position on the overall objective value. Specifically, we apply one of the following solution approaches:

- **DP**: the dynamic programming algorithm presented in Section 3, which obtains the optimal lap assignment and schedule simultaneously.
- **2S**: a two-stage heuristic procedure, in which first the lap assignments are determined, and then the schedule of each such assignment is obtained. Specifically, in the first stage, all recharging opportunities of type (+) are identified and all subsets whose sizes do not exceed a predefined value are enumerated. In the second stage, the schedules are obtained via an LP (Section 2) or a scheduling heuristic (Electronic Companion F). These heuristics are based on two simple rules, namely “recharge as early as possible” and “recharge as late as possible”.

### 4.3. Initialization procedure and acceptance criterion

We construct initial vehicle sequences by starting from empty sequences and iteratively applying the random order best insertion operator, as defined in Section 4.2. Any request that is not successfully inserted into the solution (e.g., if all candidate insertion positions violate the request's time window), is inserted instead into the bank.

We use a hierarchical evaluation function with a primary component that represents the number of requests in the bank and a secondary component representing the objective value of the eDARP-FC. During the search, a new solution with fewer requests in the bank compared to the current solution is always accepted. Vice versa, a new solution with more requests in the bank is never accepted. In cases where the new solution has the same number of requests in the bank we apply a deterministic annealing acceptance criterion with respect to the objective value. Specifically, let $cost(s^{curr})$ and $cost(s^{new})$ denote the evaluations of the current and new solutions, respectively. In addition, let $currtime$, $maxtime$ and $init$ denote, the elapsed computation time, the maximum computation time and the initial accepted percentage increase in the objective value. A new solution having the same number of requests in the bank is accepted if the following condition is met:

$$cost(s^{new}) \leq cost(s^{curr}) \cdot \left(1 + init \cdot \frac{maxtime - currtime}{maxtime}\right)$$

Note that this condition implies that strictly improving solutions are always accepted. Finally, the incumbent solution is updated in case that $s^{new}$ is strictly improving compared to $s^{curr}$ according to the same hierarchical evaluation function.

## 5. Numerical Results

In this section, we describe a series of numerical experiments designed to evaluate the performance of the methods presented in the previous sections. We base these experiments on two sets of problem instances whose construction approach will be detailed in Section 5.1. Following this, we analyze in Section 5.2 the impact of various lap assignment and scheduling approaches on the quality of the obtained solutions and on the required computation times. In addition, we provide in this section a comparison to fixed schedule services and highlight scenarios under which the on-demand service is shown to be beneficial.

### 5.1. Problem instances

Two sets of problem instances are used in the numerical experiments. The first set is based on an artificial network with a simple structure as illustrated in Figure 1. The second set consists of a real-life network structure derived from the driverless bus service launched during 2019 in Renmark, Australia, as illustrated in Figure 2. Together, these two sets provide a variety of input sizes and demand characteristics. A summary of the main parameters with respect to the network and the fleet is presented in Table 2 below, and following

this we discuss the generation of requests. All of the instances are available at the following website: https://www.autolab.sites.tau.ac.il/publications.

**Table 2: Instance parameters**

| | $S$ | $t^s$ | $\theta$ | $\sigma$ | $V$ | $C$ | $\beta^m$ | $\beta^c$ | $\beta^d$ |
|---|---|---|---|---|---|---|---|---|---|
| Artificial | 6 | Varying 1-8 | 24 | 1 | $3-4$ | 6 | [9,11,13,17] | 1 | 4 |
| Renmark | 10 | Varying 2-6 | 30 | 1 | $4-9$ | 10 | 40 | 1 | 16 |

For the artificial instances, note that with the given parameters, a vehicle can complete up to four laps with a full battery, and up to 17 minutes would be required to fully recharge an empty battery. We note that for the artificial instances, we have randomized 10 different spatial distributions of the stations along the circuit, for the sake of diversity. For the Renmark instances, we note that though the existing service is based on a single vehicle, in this experiment we tested fleet sizes that vary between five to eight vehicles, so as to analyze more complex service characteristics. The travel times between the stations were obtained from the published fixed schedule, assuming a one-minute stop at each station, and are presented over their corresponding links in Figure 2. We note that the actual battery characteristics for this network are not publicly available and were approximated using the published schedule. With the used parameters, 2.5 laps can be completed with a full battery, and 40 minutes are required to fully recharge an empty battery.

Next, we describe the random mechanism applied to generate requests. In particular, we generate instances that include $[25, 50, 75]$ single-passenger requests for the artificial network and $[100,300]$ multi-passenger requests for the Renmark network, having one, two, or three passengers per requests, with equal probabilities. We generate 10 instances for each number of requests and denote them by the convention "xy_z", where x represents the set type (artificial/Renmark), y represents the number of requests and z represents the instance number. For each request we first draw the pick-up and drop-off stations. The pick-up station was selected at random from all passenger stations. In the artificial instances the drop-off station was randomly selected from the set of passenger stations, excluding the pick-up station. In the Renmark instances, the drop-off station was randomly selected from the five stations that follow the pick-up station to reflect realistic demand patterns, namely, to avoid trips that are considerably longer than the walking time between the stations. We draw the earliest pick-up time from a uniform distribution between predefined values ($U(30,200)$ for the artificial instances and $U(60,420)$ for the Renmark instances. The latest drop-off time at node $i$ is calculated using the following equation:

$$d_i = a_{i-n} + \gamma_1 \cdot \delta(i-n, i) + \gamma_2$$

where $\gamma_1$ and $\gamma_2$ are instance specific parameters representing a direct time multiplier and a constant time addition, respectively. In the experiment we have tested the following values: $\gamma_1 \in \{1.5,2\}$ and $\gamma_2 \in \{15,20\}$. Finally, recall that the objective function is a weighted sum of the total number of laps traveled by

the shuttles and the total passenger excess time, multiplied by weight factors $\alpha_1$ and $\alpha_2$, respectively. Throughout this experiment we use the values $\alpha_1 = 0.85$, $\alpha_2 = 0.15$.


**Figure 2: The eastern route of the driverless bus service in Renmark, Australia. Travel times in minutes are depicted on each link.**

### 5.2. Results and analysis

The LNS framework and all its components, including the DP algorithm and the scheduling heuristics were implemented in C++ and were executed on a Xeon Gold 6140 CPUs@2.3 GHz desktop with 192GB of RAM. The MILP formulation was implemented in GUROBI v9.5.2 and run on an Intel Core 6230 CPU@ 2.1GHz desktop with 128GB of RAM. In what follows we address several research questions: 1) is there value in using the 2-stage lap assignment and scheduling heuristics over the exact DP approach; 2) how does the LNS heuristic compare to a state-of-the-art MILP solver in solving the eDARP-FC; 3) how sensitive are the results to changes in several parameter settings, including: the battery capacity, the time window widths, and the vehicle fleet size; and 4) can flexible services constitute a superior alternative to fixed schedule services? These questions are addressed in Sections 5.2.1 to 5.2.4, respectively.

#### 5.2.1. Comparison of lap assignment and scheduling approaches

We have tested the performance of the LNS framework using the following lap assignment and scheduling approaches: the dynamic programming algorithm (DP), the 2-stage procedure with the recharge as early as possible scheduling heuristic (2S_Earliest), the 2-stage procedure with the recharge as late as possible scheduling heuristic (2S_Latest), the 2-stage procedure using both scheduling heuristics (2S_Both) and the 2-stage procedure using the LP for scheduling (2S_LP). In Table 3, we present the results obtained for the artificial instances by applying the LNS framework with a 60 seconds time limit using the following baseline parameter settings: $\gamma_1 = 1.5, \gamma_2 = 15, \beta^m = 11$. Specifically, each row represents the average values over 100 runs, i.e. 10 instances × 10 algorithm repetitions. The first three columns in the table

represent the number of requests per demand scenario, the number of vehicles used and the solution approach. The following five columns present the number of requests in the bank, the objective value, the first component of the objective function (number of laps), the second component of the objective function (total passenger excess time) and the number of LNS iterations performed within the time limit. As can be observed, the DP approach produces superior results in all three instance sizes. The DP results are obtained with fewer iterations, as each iteration applying the DP comprises of higher computational burden. Nevertheless, this highlights the impact of determining the optimal lap assignment and schedules of given sequences on the evolution of the search process. In addition, note that the linear programming based approach consistently yields poor results. This is due to running time of each schedule evaluation, which overall does not permit executing a sufficient number of LNS iterations.

**Table 3: Comparison of Lap Assignment and Scheduling Approaches based on the Artificial Instances (60 seconds, $\gamma_1 = 1.5, \gamma_2 = 15, \beta^m = 11$)**

| $n$ | $V$ | Approach | $n_b$ | Obj. | Laps | Excess | Itr. |
|---|---|---|---|---|---|---|---|
| 25 | 3 | DP | 0.00 | 23.31 | 14.80 | 71.50 | 25469 |
| 25 | 3 | 2S_Earliest | 0.00 | 23.67 | 14.80 | 73.90 | 80248 |
| 25 | 3 | 2S_Latest | 0.00 | 23.98 | 14.50 | 77.70 | 64509 |
| 25 | 3 | 2S _Both | 0.00 | 23.63 | 14.90 | 73.10 | 32687 |
| 25 | 3 | 2S_LP | 0.00 | 23.72 | 14.83 | 74.11 | 195 |
| 50 | 3 | DP | 0.00 | 53.29 | 16.83 | 259.90 | 12238 |
| 50 | 3 | 2S_Earliest | 0.00 | 53.76 | 16.30 | 266.00 | 27746 |
| 50 | 3 | 2S_Latest | 0.00 | 55.11 | 16.29 | 275.10 | 21378 |
| 50 | 3 | 2S _Both | 0.00 | 53.77 | 16.30 | 266.08 | 10751 |
| 50 | 3 | 2S_LP | 0.02 | 56.18 | 16.84 | 279.09 | 63 |
| 75 | 4 | DP | 0.00 | 65.82 | 23.20 | 307.34 | 6416 |
| 75 | 4 | 2S_Earliest | 0.00 | 66.72 | 22.95 | 314.73 | 11960 |
| 75 | 4 | 2S_Latest | 0.00 | 67.83 | 22.57 | 324.31 | 9673 |
| 75 | 4 | 2S _Both | 0.00 | 66.94 | 23.09 | 315.45 | 4678 |
| 75 | 4 | 2S_LP | 0.09 | 80.20 | 22.28 | 408.42 | 16 |

Table 4 reports the results obtained for the Renmark instances, with a 600 seconds time limit. It follows the same structure of Table 3. The values presented in this table are averaged over 100 runs, i.e., 10 demand scenarios × 10 algorithm repetitions, for a representative parameter setting: $\gamma_1 = 1.5, \gamma_2 = 15, \beta^m = 40$. As can be seen, for these instances the DP approach is no longer superior. This can be attributed to the considerably lower number of LNS iterations performed within 600 seconds. The linear programming based approach seems to produce even poorer results as the number of requests grows. Specifically, for the 300-request instances, the framework could not even complete the first iteration. Comparing the four 2-stage heuristics, it is evident that the 2S_Earliest approach yields the best results for both the 100- and 300-request instances. Consequently, in the following sections we focus on results obtained using this approach, unless stated otherwise.

**Table 4: Comparison of Lap Assignment and Scheduling Approaches based on the Renmark Instances (600 seconds, $\gamma_1 = 1.5, \gamma_2 = 15, \beta^m = 40$)**

| $n$ | $V$ | Approach | $n_b$ | Obj. | Laps | Excess | Itr. |
|---|---|---|---|---|---|---|---|
| 100 | 5 | DP | 0.00 | 100.72 | 34.33 | 476.94 | 1447 |
| 100 | 5 | 2S_Earliest | 0.00 | 101.16 | 32.81 | 488.45 | 25306 |
| 100 | 5 | 2S_Latest | 0.00 | 107.96 | 31.95 | 538.66 | 20725 |
| 100 | 5 | 2S _Both | 0.00 | 101.24 | 32.78 | 489.18 | 12753 |
| 100 | 5 | 2S_LP | 0.01 | 122.51 | 33.99 | 624.14 | 64 |
| 300 | 8 | DP | 0.00 | 379.99 | 57.18 | 2209.22 | 421 |
| 300 | 8 | 2S_Earliest | 0.00 | 361.05 | 55.60 | 2091.93 | 3786 |
| 300 | 8 | 2S_Latest | 0.00 | 385.08 | 54.53 | 2258.18 | 3054 |
| 300 | 8 | 2S _Both | 0.00 | 374.84 | 55.62 | 2183.75 | 1661 |
| 300 | 8 | 2S_LP | - | - | - | - | - |

To test these insights statistically, we have compared the leading scheduling approach to each of the other approaches, at the instance level. This resulted with a total of 190 comparisons (4 comparisons times 50 instances minus 10 300-requests instances in which the LP is inapplicable). Out of 190 t-tests that were conducted, 147 were significant, i.e. with p-value smaller than 0.05. In 8 of the remaining cases, the difference was not significant since both approaches were able to find the optimal solution in all runs (i.e. obtained the exact same average). Thus, it is safe to assume that whenever the averages are not equal, the differences are statistically significant.

Next, we zoom in on the performance of the scheduling heuristics by fixing the lap assignment decisions. This allows us to further analyze the capability of the scheduling heuristics to find optimal schedules. For this purpose, for each of the Renmark instances, we have randomly generated 100,000 single vehicle sequences. The optimal lap assignment and scheduling of each sequence was first determined using the DP algorithm. Then, we have fixed the lap assignment decisions and solved each resulting scheduling sub-problem with the scheduling heuristics (Earliest, Latest and Both). In Table 5, we present the results of this experiment. The first column presents the instance name, the following four columns present the average values of the schedules for the four scheduling approaches, and the last three columns represent the percentage of cases in which each heuristic succeeded in finding the optimal schedule. The average performance of the heuristics is between 5% to 29% worse than the optimal schedules. Earliest consistently performs better than Latest, and as can be expected, considering Both benefits from the advantages of both and therefore leads to better schedules. The same trend is observed when viewing the percentage of cases that the heuristics were able to find the optimal schedules. Once again, these results emphasize the need to carefully tradeoff between the times required to find schedules and their quality within the overall solution framework.

**Table 5: Comparison of Scheduling Approaches based on the Renmark Instances (600 seconds, $\gamma_1 = 1.5, \gamma_2 = 15, \beta^m = 40$)**

| Instance Name | Average Objective Values (per vehicle) | | | | % Optimal Schedules | | |
|---|---|---|---|---|---|---|---|
| | DP | Earliest | Latest | Both | Earliest | Latest | Both |
| r100_0 | 21.37 | 24.23 | 26.63 | 23.53 | 41.85% | 18.36% | 43.74% |
| r100_1 | 26.73 | 29.90 | 31.08 | 28.82 | 39.87% | 23.83% | 43.52% |
| r100_2 | 25.09 | 28.24 | 30.57 | 27.10 | 46.33% | 22.15% | 48.62% |
| r100_3 | 23.20 | 26.52 | 27.61 | 25.25 | 44.40% | 23.49% | 47.29% |
| r100_4 | 24.94 | 28.40 | 32.09 | 27.23 | 42.12% | 23.21% | 45.60% |
| r100_5 | 28.00 | 32.56 | 33.93 | 30.62 | 36.16% | 20.34% | 41.67% |
| r100_6 | 26.91 | 30.76 | 33.08 | 29.41 | 41.51% | 20.42% | 44.40% |
| r100_7 | 28.44 | 32.69 | 33.51 | 30.94 | 31.86% | 20.94% | 37.84% |
| r100_8 | 29.77 | 34.82 | 35.34 | 32.62 | 30.67% | 21.09% | 36.48% |
| r100_9 | 27.22 | 29.93 | 32.26 | 29.12 | 46.76% | 22.06% | 50.45% |
| r300_0 | 57.43 | 61.72 | 64.00 | 60.33 | 37.95% | 17.58% | 41.73% |
| r300_1 | 61.16 | 66.33 | 67.97 | 64.60 | 29.14% | 15.61% | 33.72% |
| r300_2 | 60.83 | 65.49 | 68.65 | 64.06 | 34.26% | 20.00% | 38.69% |
| r300_3 | 61.15 | 67.81 | 70.51 | 65.07 | 28.65% | 13.40% | 34.73% |
| r300_4 | 61.52 | 67.61 | 71.09 | 65.53 | 24.16% | 13.86% | 29.17% |
| r300_5 | 60.26 | 65.86 | 65.47 | 63.43 | 27.99% | 17.12% | 31.66% |
| r300_6 | 59.43 | 64.77 | 68.10 | 62.84 | 31.80% | 13.80% | 35.03% |
| r300_7 | 56.55 | 61.40 | 63.94 | 59.62 | 36.01% | 15.13% | 40.08% |
| r300_8 | 59.59 | 65.63 | 67.25 | 63.42 | 32.75% | 15.77% | 36.69% |
| r300_9 | 63.06 | 68.82 | 69.30 | 66.10 | 34.74% | 21.96% | 40.88% |

### 5.2.2. Comparison to a state-of-the-art commercial solver

In Table 6, we compare the average results obtained for the baseline parameter configuration by 10 repetitions of the LNS framework, each within 60 seconds, to those obtained using Gurobi to solve the MILP model presented in Section 2, enhanced by the valid inequalities presented in Electronic Companion B. In particular, we present the results obtained by Gurobi with a running time limitations of 120 seconds and 10 hours, used for heuristic and exact comparison, respectively. The first two columns of the table present the instance name and the fleet size. The third and fourth columns present the objective value and actual CPU time when a 10-hour limitation is set. Note that all of the 25-request and 50-request instances (except one) converge to the optimal values within this time (denoted by an asterisk). The fifth and sixth columns present analogous values for the 120 seconds time limitation. The remaining five columns refer to the LNS framework and present the objective value, the number of laps, the total passenger excess time, the number of LNS iterations and the optimality gap. As can be seen, for most of the 25- and 50-request instances, the LNS framework was able to find the optimal solutions within 60 seconds. In contrast, after 120 seconds the optimality gaps obtained by Gurobi range from a few percent for the 25-request instances up to 30.67% for the 50-request instances. For the 75-requests instances, no incumbent solution could be obtained within 120 seconds, and the optimality gaps could not be closed within a time limit of ten hours (with remaining optimality gaps of 77% to 88%). Furthermore, the solutions obtained be the LNS framework within 60 seconds are superior to the ones obtained by Gurobi in 10 hours. This highlights that

one cannot expect to solve the MILP formulation for realistic settings using a state-of-the-art solver within a reasonable computational time, further emphasizing the practical importance of the LNS framework.

**Table 6: Comparison of heuristic and optimal solutions for the artificial instances ($\gamma_1 = 1.5, \gamma_2 = 15, \beta^m = 11$)**

| Instance Name | $V$ | Gurobi (10 hours) | | Gurobi (120 seconds) | | LNS Framework - DP (60 seconds) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Obj. | CPU (s) | Inc. | Opt. Gap | Obj. | Laps | Excess | Itr. | Opt. Gap |
| a25_0 | 3 | 17.45* | 305 | 17.45 | 0.00% | 17.45 | 14.00 | 37.00 | 31094 | 0.00% |
| a25_1 | 3 | 24.00* | 2279 | 26.10 | 8.75% | 24.00 | 15.00 | 75.00 | 29005 | 0.00% |
| a25_2 | 3 | 23.20* | 2301 | 25.20 | 8.62% | 23.20 | 16.00 | 64.00 | 23265 | 0.00% |
| a25_3 | 3 | 26.35* | 562 | 27.25 | 3.42% | 26.35 | 16.00 | 85.00 | 22478 | 0.00% |
| a25_4 | 3 | 21.45* | 217 | 21.85 | 1.86% | 21.45 | 12.00 | 75.00 | 26435 | 0.00% |
| a25_5 | 3 | 26.50* | 239 | 27.20 | 2.64% | 26.50 | 16.00 | 86.00 | 35351 | 0.00% |
| a25_6 | 3 | 23.55* | 490 | 24.30 | 3.18% | 23.55 | 12.00 | 89.00 | 29171 | 0.00% |
| a25_7 | 3 | 20.85* | 5517 | 22.95 | 10.07% | 20.85 | 15.00 | 54.00 | 13318 | 0.00% |
| a25_8 | 3 | 24.75* | 530 | 25.55 | 3.23% | 24.75 | 15.00 | 80.00 | 25565 | 0.00% |
| a25_9 | 3 | 24.95* | 98 | 24.95 | 0.00% | 24.95 | 17.00 | 70.00 | 19008 | 0.00% |
| a50_0 | 3 | 56.75 | 36000 | 65.55 | - | 56.75 | 17.00 | 282.00 | 11378 | - |
| a50_1 | 3 | 53.20* | 11184 | 61.75 | 16.07% | 53.20 | 16.00 | 264.00 | 16439 | 0.00% |
| a50_2 | 3 | 52.00* | 23500 | 67.95 | 30.67% | 52.18 | 16.10 | 256.60 | 11839 | 0.34% |
| a50_3 | 3 | 53.05* | 2561 | 53.75 | 1.32% | 53.13 | 16.20 | 262.40 | 15678 | 0.15% |
| a50_4 | 3 | 57.10* | 16165 | - | - | 57.10 | 16.00 | 290.00 | 13208 | 0.00% |
| a50_5 | 3 | 53.70* | 13917 | 58.20 | 8.38% | 53.70 | 18.00 | 256.00 | 8072 | 0.00% |
| a50_6 | 3 | 45.95* | 19626 | - | - | 45.95 | 17.00 | 210.00 | 13188 | 0.00% |
| a50_7 | 3 | 56.20* | 4634 | 68.35 | 21.62% | 56.20 | 19.00 | 267.00 | 11112 | 0.00% |
| a50_8 | 3 | 52.30* | 24559 | - | - | 52.30 | 16.00 | 258.00 | 10817 | 0.00% |
| a50_9 | 3 | 52.40* | 20467 | - | - | 52.40 | 17.00 | 253.00 | 10645 | 0.00% |
| a75_0 | 4 | 66.10 | 36000 | - | - | 63.88 | 23.80 | 291.00 | 5911 | - |
| a75_1 | 4 | 60.25 | 36000 | - | - | 60.36 | 22.90 | 272.60 | 5400 | - |
| a75_2 | 4 | 79.00 | 36000 | - | - | 68.80 | 25.00 | 317.00 | 5669 | - |
| a75_3 | 4 | 72.65 | 36000 | - | - | 70.54 | 23.00 | 339.90 | 7620 | - |
| a75_4 | 4 | 78.00 | 36000 | - | - | 67.57 | 22.30 | 324.10 | 6813 | - |
| a75_5 | 4 | 73.30 | 36000 | - | - | 68.50 | 22.70 | 328.00 | 6432 | - |
| a75_6 | 4 | 69.35 | 36000 | - | - | 63.52 | 22.00 | 298.80 | 5321 | - |
| a75_7 | 4 | 76.90 | 36000 | - | - | 68.25 | 24.00 | 319.00 | 8019 | - |
| a75_8 | 4 | 72.20 | 36000 | - | - | 62.76 | 22.90 | 288.60 | 5539 | - |
| a75_9 | 4 | 65.15 | 36000 | - | - | 64.05 | 23.40 | 294.40 | 7436 | - |

In Figure 3, we present the evolution of the solution values for the 25, 50 and 75-requests instances, using the LNS framework (in grey) and Gurobi (in orange), compared to the best known solution found by either approach. As can be observed, in all cases the LNS framework fully converges in a few minutes and in particular almost all 25 and 50-request instances converge to the optimal solution in less than a minute. In comparison, using Gurobi, a considerably longer calculation time is needed to obtain the same incumbent solution. Furthermore, for the 75-requests instances the remaining relative gap is 9%, on average, after one hour of CPU time.

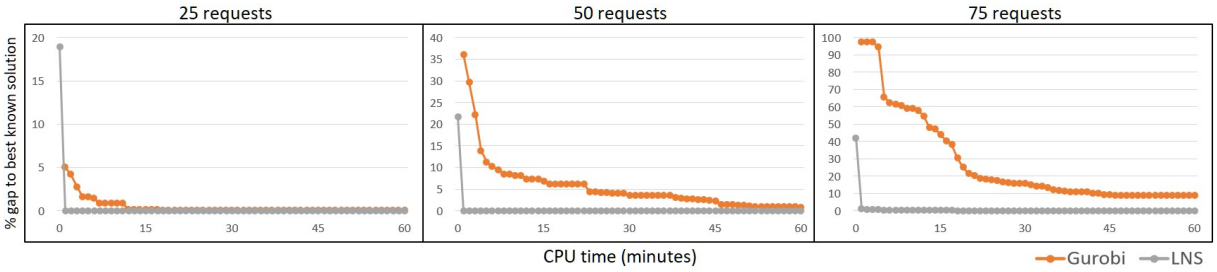


**Figure 3: Evolution of the solution values over time - Artificial Instances ($\gamma_1 = 1.5, \gamma_2 = 15, \beta^m = 11$)**

### 5.2.3. Sensitivity analysis

Next, we perform a sensitivity analysis on our results. First, we test the impact of the battery capacity. Table 7 presents the results on the artificial instances obtained using the LNS framework with the DP approach for different battery capacity values. Each row in the table presents the average values obtained over 100 runs, namely 10 repetitions of 10 instances per setting. Noticeably, as the battery capacity increases the DP requires more computational effort and therefore less LNS iterations can be completed within the 60 seconds time limit. At the same time, increasing the battery capacity provides added flexibility to the system, and hence, the overall solution value improves. The eighth column in the table displays the average number of recharging visits per setting. Noticeably, when the battery capacity increases from 9 to 11 slightly more visits to the recharging station are performed, which can be explained by the added flexibility, whereas further increasing the battery capacity leads to an expected drop in the number of recharging visits. The last four columns in the table consider the extent to which the added capacity is used during recharging at the first four laps (when increasing capacity from 9 to 11, 11 to 13, and 13 to 17). Observe that the added capacity is mainly used in the first lap, and to a lesser extent during the following laps. Indeed, as the vehicles advance in their routes, they are less likely to require more recharging in order to complete their tasks. Nevertheless, the impact of the added capacity is non-negligible.

**Table 7: The impact of the battery capacity – Artificial instances (60 seconds, $\gamma_1 = 1.5, \gamma_2 = 15$)**

| $n$ | $V$ | $\beta^m$ | Obj. | Laps | Excess | Itr. | Avg. recharging visits | % cases added capacity is used | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | Lap 1 | Lap 2 | Lap 3 | Lap 4 |
| 25 | 3 | 9 | 21.26 | 14.48 | 59.72 | 26815 | 2.63 | - | - | - | - |
| 25 | 3 | 11 | 23.73 | 14.70 | 74.90 | 51419 | 2.67 | 23.00% | 3.67% | 13.00% | 0.00% |
| 25 | 3 | 13 | 23.31 | 14.80 | 71.50 | 25719 | 1.93 | 72.00% | 12.00% | 20.33% | 14.67% |
| 25 | 3 | 17 | 19.64 | 14.10 | 51.06 | 19531 | 1.67 | 64.33% | 12.00% | 6.67% | 3.33% |
| 50 | 3 | 9 | 18.37 | 14.30 | 41.40 | 10589 | 2.87 | - | - | - | - |
| 50 | 3 | 11 | 48.65 | 16.78 | 229.23 | 13635 | 2.93 | 50.00% | 3.33% | 26.67% | 3.33% |
| 50 | 3 | 13 | 54.27 | 16.40 | 268.88 | 26663 | 2.21 | 73.33% | 8.00% | 29.00% | 41.33% |
| 50 | 3 | 17 | 53.27 | 16.80 | 259.90 | 12534 | 1.94 | 73.67% | 7.33% | 19.67% | 5.33% |
| 75 | 4 | 9 | 45.23 | 16.77 | 206.47 | 9944 | 3.05 | - | - | - | - |
| 75 | 4 | 11 | 41.84 | 17.16 | 181.67 | 5397 | 3.09 | 45.25% | 6.50% | 29.75% | 6.25% |
| 75 | 4 | 13 | 61.35 | 23.19 | 277.55 | 7238 | 2.32 | 85.00% | 5.75% | 26.25% | 41.50% |
| 75 | 4 | 17 | 66.93 | 22.95 | 316.14 | 14204 | 2.12 | 74.75% | 7.75% | 19.50% | 5.00% |

In Table 8, we analyze the sensitivity of the solution values to the tightness of the requests' time windows. This analysis was carried out by varying the values of the parameters $\gamma_1, \gamma_2$ (representing the direct travel time multiplier and constant time factor, respectively) in the artificial instances and looking into the solutions obtained using the LNS framework with the DP approach. As before, each row in the table presents the average values obtained over 100 runs. In terms of the number of iterations that can be completed within the 60 second running time, a consistent decrease is observed as $\gamma_1, \gamma_2$ increase. This can be explained by the fact that as the requests' time windows become more loose, there are less opportunities

to screen-out possible insertions at each LNS iterations. On the other hand, loosening the time windows provides an opportunity for better schedules, leading to a mixed trend with respect to the objective value.

**Table 8: The impact of the time-window width – Artificial instances (60 seconds, $\beta^m = 11$)**

| $n$ | $V$ | $\gamma_1$ | $\gamma_2$ | Obj. | Laps | Excess | Itr. |
|---|---|---|---|---|---|---|---|
| 25 | 3 | 1.5 | 15 | 23.31 | 14.80 | 71.50 | 25469 |
| 25 | 3 | 2.0 | 15 | 23.31 | 14.80 | 71.50 | 18966 |
| 25 | 3 | 1.5 | 20 | 23.31 | 14.80 | 71.50 | 19527 |
| 25 | 3 | 2.0 | 20 | 23.31 | 14.80 | 71.50 | 14402 |
| 50 | 3 | 1.5 | 15 | 53.29 | 16.83 | 259.90 | 12238 |
| 50 | 3 | 2.0 | 15 | 53.27 | 16.80 | 259.90 | 8755 |
| 50 | 3 | 1.5 | 20 | 53.27 | 16.78 | 260.05 | 8763 |
| 50 | 3 | 2.0 | 20 | 53.27 | 16.80 | 259.93 | 6105 |
| 75 | 4 | 1.5 | 15 | 65.82 | 23.20 | 307.34 | 6416 |
| 75 | 4 | 2.0 | 15 | 65.91 | 23.07 | 308.64 | 4807 |
| 75 | 4 | 1.5 | 20 | 65.92 | 23.17 | 308.18 | 4692 |
| 75 | 4 | 2.0 | 20 | 65.96 | 23.07 | 308.98 | 3384 |

For the Renmark instances, we have also tested the impact of the fleet sizes. Table 9 presents the results for the 100- and 300-request instances, with fleet sizes of {4,5,6} and {7,8,9}, respectively. As can be observed, the quality of the solutions greatly improves with the increase in the fleet size. With more vehicles, the offered load increases and so does the ability to coordinate the vehicles. The total number of laps performed increases, allowing to better facilitate the passenger requests, thus, the total excess time significantly decreases. Further on, to examine the importance of the number of vehicles as compared to the vehicle capacity, we evaluate various configurations in which the total number of seats is constant. Specifically, in Table 10 we report the results for 50 seat settings for the 100-request instances, considering the following configurations: (2 vehicles, 25 seats), (5 vehicles, 10 seats) and (10 vehicles, 5 seats). Similarly, we test for the 300-request instances a setting with 80 seats, considering the following configurations: (2 vehicles, 40 seats), (4 vehicles, 20 seats), (8 vehicles, 10 seats), (16 vehicles, 5 seats), and (20 vehicles, 4 seats). As can be observed, for the weight factors used in this experiment ($\alpha_1 = 0.85$, $\alpha_2 = 0.15$) it appears that it is beneficial to operate a larger fleet with low capacity vehicles rather than using a small fleet with a high capacity. This results in a considerable decrease in the weighted total excess time and a more moderate increase in the weighted total number of laps performed. As the weight of latter component in the objective function increases we expect to see an opposite trend.

**Table 9: The impact of the fleet size – Renmark instances (600 seconds, 2S_Earliest, $\gamma_1 = 1.5, \gamma_2 = 15, \beta^m = 40$)**

| $n$ | $V$ | Obj. | Laps | Excess | Itr. |
|---|---|---|---|---|---|
| 100 | 4 | 145.824 | 27.820 | 814.510 | 33658 |
| 100 | 5 | 101.156 | 32.810 | 488.450 | 25306 |
| 100 | 6 | 77.128 | 37.640 | 300.890 | 17601 |
| 300 | 7 | 433.831 | 49.430 | 2612.100 | 4409 |
| 300 | 8 | 361.050 | 55.600 | 2091.930 | 3786 |
| 300 | 9 | 314.357 | 61.620 | 1746.530 | 2800 |

**Table 10: The impact of capacity distribution over vehicles - Renmark instances (600 seconds, 2S_Earliest, $\gamma_1 = 1.5, \gamma_2 = 15, \beta^m = 40$ )**

| $n$ | $V$ | C | Obj. | Laps | Excess | Itr. |
|---|---|---|---|---|---|---|
| 100 | 2 | 25 | 231.23 | 15.08 | 1456.05 | 91679 |
| 100 | 5 | 10 | 101.16 | 32.81 | 488.45 | 25306 |
| 100 | 10 | 5 | 48.73 | 47.59 | 55.19 | 9758 |
| 300 | 2 | 40 | 555.80 | 14.73 | 3621.83 | 18978 |
| 300 | 4 | 20 | 724.09 | 29.56 | 4659.75 | 6834 |
| 300 | 5 | 16 | 606.13 | 36.36 | 3834.80 | 5478 |
| 300 | 8 | 10 | 361.05 | 55.60 | 2091.93 | 3786 |
| 300 | 10 | 8 | 288.80 | 67.85 | 1540.85 | 2332 |
| 300 | 16 | 5 | 192.36 | 101.26 | 708.59 | 1248 |
| 300 | 20 | 4 | 166.95 | 120.74 | 428.83 | 950 |

### 5.2.4. Comparison to fixed-schedule services

Finally, to measure the potential benefit of the proposed on-demand service, we benchmark it against the existing fixed-stop fixed-schedules service, based on the Renmark case study. We start by describing the fixed service and then compare it to the proposed flexible service. In the fixed service, vehicles are required to stop at all stations. In addition, the vehicles are distributed with uniform margins along the fixed-circuit. Thus, the lap travel time divided by the number of vehicles, determines the headway between consecutive vehicles. When a vehicle stops at a station, it serves the requests waiting there according to a first-come-first-served regime, up to the capacity of the vehicle. The remaining requests are required to wait for the following vehicles. As in the flexible service, passengers disembark at the first visit to their destination following their embarking. The fixed service is simulated by applying the following assumptions: 1) the time required to complete a lap includes the direct travel time between the stations, the service times at all stations and the time required to recharge the battery for a single lap. 2) The recharging time is deducted from the excess times of passengers who are bound to be on board the vehicle while the vehicle enters the recharging station. This is done in order to adhere to the requirement that no passengers spend time at the recharging depot.

For a meaningful comparison, we evaluate both services with the same system settings (network structure, travel and service times, fleet size, battery characteristics, vehicle capacity) and the same demand realizations. With respect to the demand distribution, as demonstrated in multiple public transit studies in the past (e.g., Ingvardson et al. 2018), approximately uniformly distributed arrival times are observed when the vehicle headway is relatively small. Hence, considering the same arrival time realizations as in the on demand service seems appropriate for 12-minute and 7-minute headways used for the 100-request and 300-request instances, respectively.

In Table 11, we present the results obtained by the LNS framework and the described above simulation for the Remark instances. Two types of scenarios are considered: five vehicles and 100-request instances,

as well as eight vehicles and 300-request instances. In addition to the performance measures presented in previous tables, we introduce an additional column representing the average excess time per passenger in the system. Recall that these instances include multi-passenger requests. We note that the results of the LNS framework are the averages of ten repetitions, while given the demand realizations, the fixed service simulation is fully deterministic and therefore was run a single time for each instance.

The results demonstrate that for all tested demand realizations, while completing approximately the same number of laps, considerably shorter excess times can be obtained using a flexible service. This is also reflected in the average excess times, which decrease by 32% to 75%. The improvement in the quality of the solutions is a direct consequence of the flexible service's ability to adapt itself to the actual requests. Note that often the vehicles in the flexible service perform slightly fewer laps though they have the ability to skip stops and thus complete laps faster. This occurs as in many cases vehicles do not directly continue to the following stations, but instead wait at the current station for passengers that are about to arrive soon. Interestingly, it can be observed that as the number of requests and vehicles increase, the gap between the average excess times in the flexible and fixed schedule services decreases. This suggests that as the frequency of service and the demand increases, the advantage of a flexible service might decrease.

**Table 11: Comparison of flexible and fixed schedule services – Renmark instances ($\gamma_1 = 1.5, \gamma_2 = 15, C = 10, \beta^m = 40$)**

| Instance Name | $V$ | LNS Framework - 2S_Earliest (600 seconds) | | | | | Fixed Schedule Service | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Obj. | Laps | Excess | Avg. Excess | Itr. | Obj. | Laps | Excess | Avg. Excess |
| r100_0 | 5 | 79.02 | 31.50 | 348.30 | 1.81 | 21352 | 237.05 | 32.00 | 1399.00 | 7.29 |
| r100_1 | 5 | 113.22 | 34.00 | 562.10 | 2.88 | 25461 | 266.45 | 35.00 | 1578.00 | 8.09 |
| r100_2 | 5 | 95.97 | 32.40 | 456.20 | 2.51 | 25260 | 259.20 | 33.00 | 1541.00 | 8.47 |
| r100_3 | 5 | 99.69 | 33.40 | 475.30 | 2.44 | 24614 | 260.30 | 35.00 | 1537.00 | 7.88 |
| r100_4 | 5 | 93.48 | 32.50 | 439.00 | 2.13 | 23394 | 251.95 | 34.00 | 1487.00 | 7.22 |
| r100_5 | 5 | 98.27 | 34.50 | 459.60 | 2.36 | 28943 | 245.30 | 32.00 | 1454.00 | 7.46 |
| r100_6 | 5 | 101.03 | 31.20 | 496.70 | 2.26 | 22486 | 251.85 | 33.00 | 1492.00 | 6.78 |
| r100_7 | 5 | 117.18 | 31.60 | 602.10 | 3.00 | 28026 | 265.45 | 34.00 | 1577.00 | 7.85 |
| r100_8 | 5 | 105.29 | 33.30 | 513.20 | 2.52 | 28031 | 257.25 | 33.00 | 1528.00 | 7.49 |
| r100_9 | 5 | 108.45 | 33.70 | 532.00 | 2.76 | 25497 | 252.70 | 34.00 | 1492.00 | 7.73 |
| r300_0 | 8 | 357.69 | 54.30 | 2076.90 | 3.57 | 3722 | 533.70 | 57.00 | 3235.00 | 5.56 |
| r300_1 | 8 | 374.85 | 56.20 | 2180.50 | 3.60 | 3817 | 525.70 | 58.00 | 3176.00 | 5.26 |
| r300_2 | 8 | 357.73 | 56.50 | 2064.70 | 3.39 | 3991 | 568.35 | 60.00 | 3449.00 | 5.66 |
| r300_3 | 8 | 366.02 | 54.60 | 2130.70 | 3.42 | 3487 | 566.45 | 59.00 | 3442.00 | 5.52 |
| r300_4 | 8 | 376.74 | 55.30 | 2198.20 | 3.65 | 3596 | 578.20 | 58.00 | 3526.00 | 5.84 |
| r300_5 | 8 | 358.17 | 56.70 | 2066.50 | 3.44 | 3911 | 576.20 | 59.00 | 3507.00 | 5.85 |
| r300_6 | 8 | 371.39 | 56.30 | 2156.90 | 3.71 | 4043 | 534.65 | 59.00 | 3230.00 | 5.56 |
| r300_7 | 8 | 330.26 | 55.40 | 1887.80 | 3.29 | 3501 | 507.05 | 59.00 | 3046.00 | 5.32 |
| r300_8 | 8 | 335.49 | 55.60 | 1921.50 | 3.30 | 3637 | 531.15 | 60.00 | 3201.00 | 5.49 |
| r300_9 | 8 | 382.18 | 55.10 | 2235.60 | 3.70 | 4157 | 591.15 | 57.00 | 3618.00 | 5.98 |

## 6. Conclusion

Autonomous shuttle services have already been brought into operation in multiple locations globally. Because of various restrictions, these services are typically provided by small fleets of electric autonomous

shuttles that travel on predefined circuits according to fixed-stop-fixed-schedule plans. The goal of this study was to develop a decision support framework that will enable operators to provide on-demand services over the same circuits, while taking to account battery management considerations.

In this paper, we have investigated the operational problem faced by the service providers of such systems, denoted as eDARP-FC, and formulated it as an MILP that differs from standard Dial-a-Ride formulations because of the circular route. We have shown that the eDARP-FC belongs to the class of NP-Hard problems and analyzed the complexity of two important sub-problems. Namely, given the sequence of requests to be served by a single vehicle, the first sub-problem simultaneously determines the optimal lap assignment and schedule, whereas the second sub-problem determines the optimal schedule given a predetermined lap assignment as well. We have shown that under realistic conditions, the first sub-problem can be solved exactly via dynamic programming, while the second one can be formulated as a linear program and can thus be solved efficiently even with no particular assumptions regarding the input. In addition, we have developed scheduling heuristics for the second sub-problem. All of these solution procedures have been incorporated as subroutines within an LNS framework, which was tested through an extensive numerical study of an artificial circuit as well as a real-world case study of the autonomous shuttle service in Renmark, Australia. Numerical results for 25- and 50-request problem instances demonstrate that in many cases, the LNS framework is capable of finding the optimal solution within one minute. The impacts of various problem parameters are examined, including: battery capacity, vehicle fleet size and request time-window widths. This leads to several methodological and managerial insights: there is a trade-off between the effort spent on the scheduling sub-problems (exact vs. heuristic) and the number of LNS iterations that can be performed in a given time limit, with different approaches leading to better results for different instance characteristics and time limits; looser time constraints allow for better solutions but require more computational effort; and for the considered setting, a larger fleet of small vehicles is to be preferred over a small fleet of large vehicles. Finally, a comparison to a fixed-stop-fixed-schedule regime has demonstrated that an on-demand service can lead to a significant reduction in passenger excess time, ranging between 32% to 75%. This improvement is achieved using the same fleet and at a comparable (and even smaller) operational cost.

To conclude, we highlight several directions for future research. First, it may be interesting to explore the impact of different network structures on the properties of the resulting optimization problem and the ensuing solution strategies. Such structures can either be simpler special cases, such as a line, or more complicated ones, such as bi-directional or star-shaped circuits. Focusing on the sub-problems, we note that once the sequence of nodes to be visited is determined, the resulting scheduling sub-problem is independent of the complete network structure. Thus, the scheduling procedures developed in this study may well be used in more general network structures. Second, a dynamic variant of the setting discussed in this paper

should be considered, in which information regarding new requests may be arriving after the system has started its operation. We believe that the modeling approaches and methodology presented in this paper may be useful for these cases as well. Third, in this study we assumed that the network includes a single recharging depot. Future studies should examine generalizations in this direction, including the incorporation of multiple recharging depots and the possibility to recharge at passenger stations. The implications of battery swapping approaches constitute an additional interesting direction for consideration.

## Acknowledgments

This work was supported by the projects Data-driven logistics (FWO Strategic Basic Research, S007318N), Optimising the design of a hybrid urban mobility system (FWO junior research project, G020222N) and OR4Logistics (FWO-WOG), all funded by the Research Foundation – Flanders. Yves Molenbruch is partially funded by the Research Foundation Flanders (FWO-1202719N). The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation Flanders (FWO) and the Flemish Government.

## References

Alamalhodaei, A. (2021). Waymo and J.B. Hunt partner to bring autonomous trucks to Texas in new pilot. *TechCrunch.* Retrieved from: https://techcrunch.com/2021/06/10/waymo-and-jb-hunt-partner-to-bring-autonomous-trucks-to-texas-in-new-pilot/.

Asghari, M., & Al-e, S. M. J. M. (2020). Green vehicle routing problem: A state-of-the-art review. International Journal of Production Economics, 107899.

Bagloee, S. A., Tavana, M., Asadi, M. and Oliver, T. (2016). Autonomous vehicles: challenges, opportunities, and future implications for transportation policies. Journal of Modern Transportation, 24(4), 284-303.

Bansal, P., & Kockelman, K. M. (2017). Forecasting Americans' long-term adoption of connected and autonomous vehicle technologies. Transportation Research Part A: Policy and Practice, 95, 49-63.

Bertoncello, M., & Wee, D. (2015). Ten ways autonomous driving could redefine the automotive world. *McKinsey & Company*, 6. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/ten-ways-autonomous-driving-could-redefine-the-automotive-world.

Bongiovanni, C., Kaspi, M., & Geroliminis, N. (2019). The electric autonomous dial-a-ride problem. Transportation Research Part B: Methodological, 122, 436-456.

Bongiovanni, C., Kaspi, M., Cordeau, J-F., & Geroliminis, N. (2021). A learning large neighborhood search for the dynamic electric autonomous dial-a-ride problem. Working paper.

Bonnefon, J. F., Shariff, A., & Rahwan, I. (2016). The social dilemma of autonomous vehicles. Science, 352(6293), 1573-1576.

Daganzo, C. F. (1984). Checkpoint dial-a-ride systems. *Transportation Research Part B: Methodological*, *18*(4-5), 315-327.

Fagnant, D. J. and Kockelman, K. (2015). Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. Transportation Research Part A: Policy and Practice, 77, 167-181.

Gurman, M. (June 2, 2021). Apple Loses Multiple Top Managers from Self-Driving Car Division. *Bloomberg.* Retrieved from: https://www.bloomberg.com/news/articles/2021-06-02/apple-loses-several-top-managers-from-self-driving-car-division

Ho, S. C., Szeto, W. Y., Kuo, Y. H., Leung, J. M., Petering, M., & Tou, T. W. (2018). A survey of dial-a-ride problems: Literature review and recent developments. Transportation Research Part B: Methodological, 111, 395-421.

Iclodean, C., Cordos, N., & Varga, B. O. (2020). Autonomous shuttle bus for public transportation: A review. Energies, 13(11), 2917.

Ingvardson, J. B., Nielsen, O. A., Raveau, S., & Nielsen, B. F. (2018). Passenger arrival and waiting time distributions dependent on train service frequency and station characteristics: A smart card data analysis. *Transportation Research Part C: Emerging Technologies*, *90*, 292-306.

Litman, T. (2018). Autonomous Vehicle Implementation Predictions. Victoria, Canada: Victoria Transport Policy Institute.

Masmoudi, M. A., Hosny, M., Demir, E., Genikomsakis, K. N., & Cheikhrouhou, N. (2018). The dial-a-ride problem with electric vehicles and battery swapping stations. Transportation research part E: logistics and transportation review, 118, 392-420.

Metz, C., & Conger, K. (December 7, 2020). Uber, After Years of Trying, Is Handing Off Its Self-Driving Car Project. *The New York Times*. Retrieved from: https://www-nytimes-com.cdn.ampproject.org/c/s/www.nytimes.com/2020/12/07/technology/uber-self-driving-car-project.amp.html

Molenbruch, Y., Braekers, K. and Caris, A. (2017). Typology and literature review for dial-a-ride problems. Annals of Operations Research, 259(1-2), 295-325.

Narayanan, S., Chaniotakis, E., & Antoniou, C. (2020). Shared autonomous vehicle services: A comprehensive review. *Transportation Research Part C: Emerging Technologies*, *111*, 255-293.

Padberg, M. W. (1976). A note on the total unimodularity of matrices. Discrete Mathematics, 14(3), 273-278.

Pimenta, V., Quilliot, A., Toussaint, H., & Vigo, D. (2017). Models and algorithms for reliability-oriented dial-a-ride with autonomous electric vehicles. European Journal of Operational Research, 257(2), 601-613.

Pisinger, D., & Ropke, S. (2010). Large neighborhood search. In Handbook of metaheuristics (pp. 399-419). Springer, Boston, MA.

Postauto, 2020. Project SmartShuttle – Testing in Sion. Retrieved from: https://www.postauto.ch/en/testing-uvrier

Quadrifoglio, L., Hall, R. W., & Dessouky, M. M. (2006). Performance and design of mobility allowance shuttle transit services: bounds on the maximum longitudinal velocity. *Transportation science*, *40*(3), 351-363.

Sage, 2020. Renmark Autonomous Vehicle Trial. Retrieved from: http://renmark.rideswithsage.com/

Shaheen, S., & Chan, N. (2016). Mobility and the sharing economy: Potential to facilitate the first-and last-mile public transit connections. Built Environment, 42(4), 573-588.

Trotta, M., Archetti, C., Feillet, D., & Quilliot, A. (2022). Pickup and delivery problems with autonomous vehicles on rings. *European Journal of Operational Research*, *300*(1), 221-236.