2021 IEEE Intelligent Vehicles Symposium (IV)
July 11-17, 2021. Nagoya, Japan

Fleet Fairness and Fleet Efficiency in Capacitated Pickup and Delivery Problems

Martin Damyanov Aleksandrov
Technical University Berlin, Germany
martin.aleksandrov@tu-berlin.de

***Abstract*— In 2016, the German Ministry for Traffic and Digital Infrastructure has granted 100 million euros on autonomous and connected vehicles. Soon after, in 2017, the German Ministry appointed Ethics Commission to regulate the use of such vehicles for social good. They identified *transparency*, *trust*, and *safety* as vital features for this purpose. Transparency requires that information about vehicles is available to customers, e.g. locations, fees, etc. Trust requires that vehicles are used in a fair and efficient manner. Safety requires that vehicles follow all road regulations and customers have top priority in cases of potential accidents. In this paper, we give scientific guarantees that respond to these requirements by providing preliminary fairness and efficiency analyses of the fleet of vehicles.**

I. Introduction

Autonomous and connected driving is likely to make our roads much safer. For this reason, we focus on the associated fairness and efficiency. In particular, we consider a fleet of *vehicles* and a number of *customer requests*. Each vehicle has some *current depot locations* and *available capacity*, as well as it moves with some *velocity*. Each request requires a service for some *demand* between some *pick-up* and *drop-off* locations. Each request can be serviced by at least one vehicle. The *distances* in our model are specified by means of any metric. These could be the straight-line or shortest distances. We focus on the case of a 2-dimensional plane because this is the most practical one.

An appealing feature of our model is that it makes use of location *delays*. For example, in the context of package pickup and delivery, the delay could include the time for collecting, delivering, loading, and/or unloading packages. As a second example, in the context of people pickup and dropoff, the delay could include the time for assisting senior citizens. However, location delays may not necessarily relate to the specifics of the customer requests. They may also relate to traffic conditions. For example, the delay at a crossing could include an estimation of the time spent at the traffic light or the congestion in terms of the number of cars, time of the day, number of lanes, etc.

Thus, for this model, we consider the notion of *feasible routing plans*. These plans induce a constrained form of routing where each request is serviced by exactly one vehicle, the pickup of each request is serviced before the dropoff, and the vehicle capacities are never exceeded. In this context, we consider incentives such as the *waiting time*, *tour time*, and *arrival time* for customer requests.

The waiting time is the time customers wait for a vehicle to pick them up. The tour time is the time customers travel between their pick-up and drop-off locations. Their arrival time is the waiting time plus the tour time. We thus formulate three natural objectives for fleet *fairness*/*efficiency*. More specifically, these are the total/maximum waiting time (totWAIT/maxWAIT), tour time (totTOUR/maxTOUR), and arrival time (totARR/maxARR). We then use social choice concepts such as *impossibility*, *the price of fairness*, and *computability* in order to analyse these objectives.

Our outline is as follows: We first show that it might be impossible to have feasible plans that minimise each pair of objectives in combination. As a result, fairness and efficiency might be incompatible. For this reason, we measure the loss in efficiency due to obtaining fairness by using the price of fairness. Also, we study how difficult might be to compute feasible plans that satisfy fairness or efficiency. This leads us to a number of computational problems, most of which are in NP-hard (i.e. intractable) and the others are in P (i.e. tractable). Table I summarises our theoretical results.

TABLE I
Key: $\times$ - impossibility result; POF - the price of fairness; $n \geq 2$ vehicles; NP - time complexity; P - time complexity.

totARR/maxARR	totTOUR/maxTOUR	totWAIT/maxWAIT
n/a	$\times$ (Thm 1)	
$\times$ (Thm 1)	n/a	$\times$ (Thm 1)
$\times$ (Thm 2)		n/a
POF$\geq (n-1)$ (Thm 3)	POF$\geq (n-1)$ (Thm 4)	POF$\geq n$ (Thm 5)
NP-hard (Thm 6)	P (Thm 7)/NP-hard (Thm 8)	NP-hard (Thm 9)

Our novel fairness and efficiency analyses apply to other extremely popular settings as well. For example, in the job-shop scheduling problem [1], the goal is to minimise the total scheduling time for completing a number of jobs by using a number of different machines. This is a case of our model where each machine is a vehicle and each job is a request. Thus, the time for starting a job corresponds to the waiting time; for processing a job corresponds to the tour time; for completing a job corresponds to the arrival time. From this perspective, our results relate also to such problems.

We present related work and formal preliminaries in Sections II and III, respectively. Section IV contains the impossibility results. We then report on the price of fairness in Section V and the computability results in Section VI. In the end, we consider a case study in Section VII and conclude in Section VIII.

II. Related Work

The Capacitated Pickup and Delivery Problem (CPDP) [2] generalises the PDP from [3] and the Capacitated Vehicle Routing Problem (CVRP) from [4]. They both generalise the 62-year-old VRP from [5]. Although the VRP literature is vast, two good surveys of PDPs are [6] and [7]. Some works considered varying drivers' incomes. We consider constant such incomes, e.g. autonomous, subsidised, or contracted drivers. Axiomatic and computational analyses are two cornerstones in economics [8], game theory [9], and social choice [10]. Recently, Vidal, Laporte, and Matl [11] discussed fairness challenges in VRPs. Nucamendi, Cardona-Valdes, and Angel-Bello Acosta [12] considered the total waiting time in VRPs with one vehicle and visit requests. In contrast, we study fleet fairness and efficiency in terms of the waiting, tour, and arrival times of customers in CPDPs with multiple vehicles of possibly unequal capacities, velocities, and location delays, as well as PD requests of possibly unequal demands and locations. From this perspective, all objectives and analyses in our work are new.

III. Preliminary Work

For $t \in \mathbb{N}_{>0}$, we let $[t]$ denote $\{1,\ldots,t\}$. We also let $L \subset \mathbb{R}^2$ denote a finite set of locations. We write $d(l,l') \in \mathbb{R}^{<\infty}_{\geq 0}$ for the metric *distance* between $l,l' \in L$, and D for the matrix $[d(l,l')]_{|L|\times|L|}$. We consider a set of *vehicles* $V = \{v_i | i \in [n]\}$, where v_i *begins/ends* at location $b_i \in L/e_i \in L$, has *available capacity* $q_i \in \mathbb{N}_{>0}$, moves with *velocity* $\nu_i \in \mathbb{R}_{>0}$, and has *delay* $\delta_{il} \in \mathbb{R}_{\geq 0}$ at location $l \in L$. We also consider a set of *customer requests* $R = \{r_j = (p_j, d_j, m_j) | j \in [m]\}$, where r_j has *pick-up* location $p_j \in L$, *drop-off* location $d_j \in L$, and *demand* $m_j \in \mathbb{N}_{>0}$. We let $\max_{r_j \in R} m_j \leq \max_{v_i \in V} q_i$ hold. That is, each r_j can be serviced by at least one v_i.

We let $\mathscr{I} = (L,D,V,R)$ denote an instance. We next define routes and plans for it. For $R' \subseteq R$, a route $\mathscr{R}' = (s_1,\ldots,s_{2|R'|})$ is a sequence of the service (i.e. pick-up/drop-off) locations of the requests from R'. We associate each location s_l in $\mathscr{R}'$ with some request $r_j \in R'$ and weight $cap_l = +m_j$ if $s_l = p_j$ and $cap_l = -m_j$ if $s_l = d_j$. Thus, a plan $\mathscr{P} = \{\mathscr{R}_1,\ldots,\mathscr{R}_n\}$ is a set of routes, where $\mathscr{R}_i$ is assigned to $v_i \in V$. We let $R_i \subseteq R$ denote the set of requests associated with v_i. We let L_i denote the set of locations of requests from R_i. We write $\mathscr{R}_i = (s^i_1,\ldots,s^i_{2|R_i|})$ and cap^i_l for each s^i_l. We also write $p_j = s^i_{k_j}$ and $d_j = s^i_{h_j}$ for $r_j \in R_i$.

A. Feasible plans

Let us consider plan $\mathscr{P}$. We consider four types of constraints. *Completeness constraints* ensure that each request is serviced once by some vehicle. Thus, all demands are transported between their pick-up and drop-off locations.

$$\forall r_j \in R, \exists v_i : m_j \leq q_i \wedge r_j \in R_i \tag{a}$$

Disjointness constraints require that the requests cannot be split across multiple vehicles. That is, no request is serviced by more than one vehicle. However, a given vehicle can still service multiple requests.

$$\forall r_j \in R, \forall v_i : r_j \in R_i \Rightarrow \forall v_k, k \neq i : r_j \notin R_k \tag{b}$$

Ordering constraints ask that the pickup of a given request is serviced before its corresponding dropoff. This still allows that the drop-off location is visited before the pick-up location, say as being a location of some other request.

$$\forall v_i, \forall r_j \in R_i : k_j < h_j \text{ if } s^i_{k_j} \neq s^i_{h_j}, \text{ else } k_j = h_j - 1 \tag{c}$$

Capacity constraints enforce that the capacity of a vehicle cannot be exceeded while servicing any request. Nevertheless, a vehicle can service multiple (pick-up/drop-off) locations while servicing a given request.

$$\forall v_i, \forall r_j \in R_i : \textstyle\sum_{l=k_j}^{h_j-1} cap^i_l \leq q_i \tag{d}$$

Constraints (a-d) prompt customers to participate in the system. For example, if the plan is not complete and some request is not serviced, then the customers may not use the system in the future.

We say that $\mathscr{P}$ is *feasible* if (a-d) hold. We note that a feasible plan always exists. To see this, pick a vehicle of the greatest capacity. This can accommodate any request. Thus, servicing the requests one by one with it is a feasible plan.

B. Fleet objectives

Let us consider feasible plan $\mathscr{P}$. The *waiting time* for r_j is the time needed for the service to start: $w_{ij} = (1/\nu_i)\cdot[d(b_i,s^i_1) + \sum_{l=1}^{(k_j-1)} d(s^i_l,s^i_{l+1})] + \delta_{ib_i} + \sum_{l=1}^{(k_j-1)} \delta_{is^i_l}$. The *tour time* for r_j is the time the service lasts: $t_{ij} = (1/\nu_i)\cdot[\sum_{l=k_j}^{(h_j-1)} d(s^i_l,s^i_{l+1})] + \sum_{l=k_j}^{h_j} \delta_{is^i_l}$.

We multiply each time by m_j as if r_j were decomposed into m_j unit-demand requests. Thus, $m_j \cdot w_{ij}$ is interpreted as the overall waiting time whereas $m_j \cdot t_{ij}$ is interpreted as the overall tour time for all m_j customers. This motivates four new fleet objectives: (1-4).

We also consider two additional fleet objectives that use sums of waiting and tour times: (5-6). More specifically, the *arrival time* for r_j is the time needed for the service to finish: $a_{ij} = w_{ij} + t_{ij}$. Thus, $m_j \cdot a_{ij}$ is interpreted as the overall arrival time for all m_j customers.

$$\text{(totWAIT): } \arg\min_{\mathscr{P}\text{: feasible}} \sum_{v_i \in V} \sum_{r_j \in R_i} m_j \cdot w_{ij} \tag{1}$$

$$\text{(maxWAIT): } \arg\min_{\mathscr{P}\text{: feasible}} \max_{v_i \in V} \sum_{r_j \in R_i} m_j \cdot w_{ij} \tag{2}$$

$$\text{(totTOUR): } \arg\min_{\mathscr{P}\text{: feasible}} \sum_{v_i \in V} \sum_{r_j \in R_i} m_j \cdot t_{ij} \tag{3}$$

$$\text{(maxTOUR): } \arg\min_{\mathscr{P}\text{: feasible}} \max_{v_i \in V} \sum_{r_j \in R_i} m_j \cdot t_{ij} \tag{4}$$

$$\text{(totARR): } \arg\min_{\mathscr{P}\text{: feasible}} \sum_{v_i \in V} \sum_{r_j \in R_i} m_j \cdot a_{ij} \tag{5}$$

$$\text{(maxARR): } \arg\min_{\mathscr{P}\text{: feasible}} \max_{v_i \in V} \sum_{r_j \in R_i} m_j \cdot a_{ij} \tag{6}$$

We highlight that totWAIT/totTOUR/totARR measures the *efficiency* of the fleet because it minimises the total workload for all vehicles whereas maxWAIT/maxTOUR/maxARR measures the *fairness* of the fleet because it aims at achieving a similar workload per vehicle.

C. Other objectives

We might wish to focus on other customer objectives such as $\arg\min_{\mathscr{P}} \max_{v_i} \max_{r_j} m_j \cdot w_{ij}$, $\arg\min_{\mathscr{P}} \max_{v_i} \max_{r_j} m_j \cdot t_{ij}$, and $\arg\min_{\mathscr{P}} \max_{v_i} \max_{r_j} m_j \cdot a_{ij}$. They iterate over $v_i \in V$ and $r_j \in R_i$. However, we do not do this in this work. The rationale behind this choice is simple.

Objectives (1-6) engage vehicles in the system with work. By comparison, plans that minimise the maximum customer times may force a single vehicle to service all requests, thus leaving all other vehicles with no work. We show this by means of Example 1.

Example 1: Consider vehicles $v_1, \ldots, v_n$ at $A(0,0)$ and v_{n+1} at $B(0,1)$. We let v_j have $q_i = 1$, $\nu_i = 1$ for $j \in [n]$ and v_{n+1} have $q_{n+1} = (n+1)$, $\nu_{n+1} = 3$. Suppose zero delays. Further, consider n requests of unit demands at $C(1,0)$ for $A(0,0)$ and one request of unit demand at $D(1,1)$ for $B(0,1)$. Dispatching v_{n+1} from B to D, then to C, A, B minimises the maximum customer waiting time to $\frac{2}{3}$ and the maximum customer arrival time to $\frac{4}{3}$.

However, the waiting time of v_{n+1} is $\frac{1}{3} + n \cdot \frac{2}{3}$ and the arrival time of v_{n+1} is $\frac{4}{3} + n$. These times go to ∞ as n goes to ∞. By comparison, dispatching v_j for $j \in [n]$ from A along (C,A), each v_j servicing a different request at C, and v_{n+1} from B along (D,B), servicing the request at D, minimises the maximum vehicle waiting (arrival) time to 1 (2). The maximum customer waiting (arrival) time is also 1 (2), being slightly greater than $\frac{2}{3}$ ($\frac{4}{3}$), but all vehicles work.

IV. Impossibility Results

Our motivation to consider all fleet objectives stems from the fact that they are generally quite different in nature. For example, minimising totTOUR/maxTOUR is *not* the same as minimising totWAIT/maxWAIT. We demonstrate this in our first impossibility result.

Theorem 1: There are instances where each feasible plan that minimises totWAIT/maxWAIT does not minimise totTOUT/maxTOUR.

Proof: Consider $A(0,0)$, $C(0,1)$, $D(1,1)$ and the straight-line metric. Further, consider v_1 with $b_1 = D$, $e_1 = D$, $q_1 = 2$, $\nu_1 = 1$ and $r_1 = (D,C,1)$, $r_2 = (A,C,1)$. Suppose zero delays. Minimising the tour times requires that the demands do not share the vehicle.

Indeed, let us suppose that the vehicle transports first the demand at D to C, then heads towards A, and finally transports the demand there to C. This plan minimises the tour times but generates waiting times: $t_{11} = 1$, $t_{12} = 1$ and $w_{11} = 0$, $w_{12} = 2$.

By comparison, let us suppose that the vehicle picks up the demand at D, drives then to A where it picks up the other demand, and heads towards C. This plan minimises the waiting times but gives greater tour times: $t_{11} = (1+\sqrt{2}) > 1$, $t_{12} = 1$ and $w_{11} = 0$, $w_{12} = \sqrt{2} < 2$. ∎

In this proof, minimising totARR/maxARR is also *not* the same as minimising totWAIT/maxWAIT. By comparison, there are also instances where minimising totARR/maxARR is *not* the same as minimising totTOUR/maxTOUR. We illustrate this in our second impossibility result.

Theorem 2: There are instances where each feasible plan that minimises totARR/maxARR does not minimise totTOUR/maxTOUR.

Proof: Consider $A(0,0)$, $B(0,1)$, $C(1,0)$ and the straight-line metric. Further, consider v_1 with $b_1 = C$, $e_1 = C$, $q_1 = 2$, $\nu_1 = 1$ and $r_1 = (A,B,1)$, $r_2 = (A,C,1)$. Suppose zero delays. Minimising the arrival times requires to pick up both demands at once.

Indeed, suppose that the vehicle picks up both demands. In this case, route (A,A,B,C) minimises the waiting times bu generates tour times: $t_{11} + t_{12} = 1 + (1+\sqrt{2}) \approx 3.4$ and $w_{11} + w_{12} = 1 + 1 = 2$. The sum is 5.4. The maximum request waiting/tour/arrival time is 1/2.4/3.4.

By comparison, if the vehicle services r_1 before r_2, then route (A,B,A,C) minimises the tour times but generates longer waiting times: $t_{11} + t_{12} = 1 + 1 < 3.4$ and $w_{11} + w_{12} = 1 + 3 = 4$. The sum is $6 > 5.4$. The maximum request waiting/tour/arrival time is 3/1/4. ∎

V. Fairness and Efficiency

Feasible plans that satisfy both fairness and efficiency might not exist in some instances. For this reason, we next measure the loss in efficiency due to fairly distributing the time workload to vehicles. To measure this loss, we adapt a popular concept from computational social choice theory to our domain: the price of fairness [13].

For measure O $\in$ {WAIT, TOUR, ARR} and $c \in \mathbb{R}_{>0}$, the *price of fairness* (POF) is the ratio between the minimum value of totO in a feasible plan that minimises maxO plus c and the minimum value of totO in any feasible plan plus c. The price is well-defined due to the choice of c. For the corresponding time $o_{ij} \in \{w_{ij}, t_{ij}, a_{ij}\}$, it is given by:

$$\mathrm{POF}(O,c) = \frac{\left[\underset{\mathscr{P}:\ \mathrm{maxO}}{\arg\min} \sum_{v_i \in V} \sum_{r_j \in R_i} m_j \cdot o_{ij}\right] + c}{\left[\underset{\mathscr{P}:\ \mathrm{feasible}}{\arg\min} \sum_{v_i \in V} \sum_{r_j \in R_i} m_j \cdot o_{ij}\right] + c}.$$

If $n = 1$, $\mathrm{POF}(O,c) = 1$. Otherwise, $\mathrm{POF}(O,c) \geq 1$. For the arrival time, using one fast vehicle might be minimising totARR. However, using all other vehicles might be minimising maxARR but thus giving at least $(n-1)$ greater value of totARR.

Theorem 3: For $n \geq 2$ vehicles, the price of fairness for the arrival time is at least $(n-1)$.

Proof: Pick $\varepsilon \in (0, \frac{2}{3(n-1)})$. We let $A_j(0, j \cdot \varepsilon)$ and $B_j(1, j \cdot \varepsilon)$ for each $j \in [n]$. Consider $V = \{v_1, \ldots, v_n\}$ and $R = \{r_1, \ldots, r_n\}$. Suppose zero delays. We let $e_j = b_j = A_j$ and $q_j = n$ for each vehicle $v_j \in V$. Further, we let $v_j \in V \setminus \{v_n\}$ move with $\nu_j = 1$ and v_n move with $\nu_n = n$. Also, we let $r_j = (B_j, A_j, 1)$ for each request $r_j \in R$.

Minimising maxARR requires us to dispatch each vehicle v_j from A_j to B_j and back again to A_j. This plan induces maximum vehicle tour/waiting time of one unit and, therefore, the maximum vehicle arrival time in it is 2, achieved by any vehicle v_j but v_n. However, the total arrival time in this plan is $2 \cdot (n-1) + \frac{2}{n}$, which is at least 3 as $n \geq 2$.

Minimising totARR requires us to dispatch vehicle v_n from A_n to $B_n, B_{n-1}, \dots, B_1$ and then from B_1 to $A_1, A_2, \dots, A_n$. The total waiting/tour time in this plan is $1+\varepsilon\frac{(n-1)}{2}/1+\varepsilon(n-1)$. Hence, their sum is $2+\varepsilon\frac{3(n-1)}{2}$. This value is in $(2,3)$ for the choice of ε. Consequently, the plan does not minimise the maximum vehicle arrival time. The price goes to $(n-1)+\frac{1}{n} \geq (n-1)$ as ε and c go to 0. ■

For the tour time, there are instances where minimising totTOUR may produce a plan where a single vehicle services all requests whereas minimising maxTOUR may produce a plan where all vehicles are engaged but the value of totTOUR is at least $(n-1)$ times greater.

Theorem 4: For $n \geq 2$ vehicles, the price of fairness for the tour time is at least $(n-1)$.

Proof: Consider the problem from Theorem 3 but for $\varepsilon \in (0, \frac{1}{(n-1)})$. Minimising maxTOUR requires us to dispatch each vehicle v_j from A_j to B_j and back again to A_j. This plan induces maximum vehicle tour time of one unit, achieved by any vehicle v_j but v_n. The tour time for r_n is equal to $\frac{1}{n}$ because v_n moves with velocity n. Consequently, the total tour time in this plan is $(n-1)+\frac{1}{n}$.

Minimising totTOUR requires us to dispatch vehicle v_n from A_n to $B_n, B_{n-1}, \dots, B_1$ and then from B_1 to $A_1, A_2, \dots, A_n$. The total tour time in this plan is $1+\varepsilon(n-1)$. By the choice of ε, it follows that this time lies in $(1,2)$. Hence, the plan does not minimise the maximum vehicle tour time. Finally, the price approaches $(n-1)+\frac{1}{n} \geq (n-1)$ as both ε and c go to 0. ■

For the waiting time, minimising totWAIT may return a plan that uses only one vehicle whereas minimising maxWAIT may return a plan that uses all vehicles. For this reason, the value of totWAIT in the latter plan is at least n times greater than in the former plan.

Theorem 5: For $n \geq 2$ vehicles, the price of fairness for the waiting time is at least n.

Proof: Pick $\varepsilon \in (0, \frac{2}{(n+2)(n-1)})$. We let $A_j(0, j \cdot \varepsilon)$ and $B_j(1, j \cdot \varepsilon)$ for $j \in [n]$, and $C_n(2, n \cdot \varepsilon)$. Consider $V = \{v_1, \dots, v_n\}$ and $R = \{r_1, \dots, r_n\}$. Suppose zero delays. Let $q_j = n$, $\nu_j = 1$ for each $v_j \in V$, $b_j = e_j = A_j$ for each $v_j \in V \setminus \{v_n\}$, and $b_n = e_n = B_n$. Further, let $r_j = (B_j, A_j, 1)$ for each $r_j \in R \setminus \{r_n\}$ and $r_n = (C_n, B_n, 1)$.

Dispatching each vehicle $v_j \in V \setminus \{v_n\}$ from A_j along (B_j, A_j), and vehicle v_n from B_n along (C_n, B_n) minimises maxWAIT. This further minimises the maximum request waiting time. Indeed, doing so induces one unit of waiting time for each r_j. However, the total waiting time of the fleet of vehicles is equal to n simply because each customer waits for a different vehicle.

Dispatching v_n from B_n to each of $B_{n-1}, \dots, B_1$ and then from B_1 to C_n, B_n minimises totWAIT. The value of it is $\varepsilon\frac{n(n-1)}{2} + \sqrt{1+\varepsilon^2(n-1)^2}$. By the choice of ε, this value belongs to $(1,2)$. Hence, this plan does not minimise maxWAIT. However, it produces almost n times shorter total time. The price goes to n as ε and c go to 0. ■

VI. Fairness or Efficiency

In response to the general incompatibility between fairness and efficiency, we might prefer feasible plans that satisfy fairness or efficiency. Such plans always exist because we consider optimisation problems related to these concepts. For this reason, we study the computability of such plans.

We say that an algorithm is NP-*hard* whenever it solves an intractable problem. For example, deciding whether the value of totARR/maxARR does not exceed a given threshold in at least one plan is intractable. We show this via a reduction from the *travelling repairman problem (TRP)* [14], [15].

Theorem 6: Unless $\mathsf{NP} = \mathsf{P}$, there are instances where each exact algorithm for totARR/maxARR is NP-hard.

Proof: Let us consider a set of locations $L = \{l_{i_1}, \dots, l_{i_m}\}$. Given a sequence $(l_{i_1}, \dots, l_{i_m})$, the *delay* for reaching the jth location is given by the term $del_j = \sum_{k=1}^{j-1} d(l_{i_k}, l_{i_{k+1}})$. Given $L = \{l_1, \dots, l_m\}$, $D = [d(l,l')]_{m \times m}$, location l_1, and $k \in \mathbb{R}_{>0}$, the TRP asks whether there is a sequence $(l_{i_1} = l_1, l_{i_2}, \dots, l_{i_m})$ such that $\sum_{j=1}^{m} del_j \leq k$ holds. We next construct an instance of our model and ask whether the value of totARR/maxARR in a feasible plan is at most k. For this purpose, we let $V = \{v_1\}$ and $R = \{r_1, \dots, r_m\}$. Further, we let v_1 have $b_1 = l_1$, $e_1 = l_1$, $q_1 = 1$, $\nu_1 = 1$ and each r_j be $(l_j, l_j, 1)$. Suppose zero delays.

In each feasible plan of our instance, we note that the value of totARR is equal to the value of maxARR because there is just one vehicle, i.e. v_1. In each such plan, we also note that each route that services the pick-up location of a given request also services directly its drop-off location. This follows by the ordering constraints. Consequently, each feasible plan is such that v_1 services each r_j, and the tour time of each r_j is 0. This immediately gives us a bijection between the solutions to the TRP and the feasible plans of the related decision problem for totARR/maxARR. Indeed, there is a sequence for the TRP of delay at most k iff the value of totARR/maxARR in a feasible plan is at most k. ■

By comparison, minimising totTOUR is tractable for any instance. More precisely, we can do it in $O(m \cdot n)$ time by using a greedy algorithm (Algorithm 1) that assigns each request to a vehicle of sufficient capacity (i.e. a feasible vehicle), minimising further the tour time for it.

Theorem 7: For each instance, there is an exact algorithm for totTOUR that runs in $O(m \cdot n)$ time.

Algorithm 1 A feasible plan for totTOUR.

procedure GREEDY(L, D, V, R) ▷ $|V| = n, |R| = m$
  $\forall v_i \in V : \mathscr{R}_i \leftarrow ()$
  $\forall v_i \in V : R_i \leftarrow \emptyset$
  $\mathscr{P} \leftarrow \{\mathscr{R}_1, \dots, \mathscr{R}_n\}$
  for $r_j \in R$ **do**
    $F = \{v_i \in V | m_j \leq q_i\}$ ▷ feasible vehicles
    pick $v_i \in F$ that minimises $\frac{d(p_j,d_j)}{\nu_i} + \delta_{ip_j} + \delta_{id_j}$
    $\mathscr{R}_i \leftarrow (s_1^i, \dots, s_{2|R_i|}^i, p_j, d_j)$
    $R_i \leftarrow R_i \cup \{r_j\}$
  return $\mathscr{P}$

Proof: Let us consider the plan returned by Algorithm 1. Pick any $r_j \in R$. Note that $m_j > 0$ holds. Also, let r_j be assigned to v_i in this plan. Hence, the tour time for r_j is $m_j \cdot [\frac{d(p_j,d_j)}{v_i} + \delta_{ip_j} + \delta_{id_j}]$. By the choice of the algorithm for r_j, v_i minimises this time among the vehicles feasible for r_j. We next argue that this time is the shortest possible in any feasible plan. For the sake of contradiction, let us *assume* that there is another feasible plan where this time is strictly shorter. In this plan, let r_j be assigned to v_k and have tour time $m_j \cdot t_{kj}$. Hence, v_k is feasible for r_j. We can write $t_{kj} = (1/\nu_k) \cdot [\sum_{l=k_j:(h_j-1)} d(s^k_l, s^k_{l+1})] + \sum_{l=k_j:h_j} \delta_{ks^k_l}$. Under our assumption, we have that $t_{kj} < (1/\nu_i) \cdot d(p_j,d_j) + \delta_{ip_j} + \delta_{id_j}$ holds. If $k \neq i$ holds, we observe that $t_{kj} \geq (1/\nu_k) \cdot d(p_j,d_j) + \delta_{kp_j} + \delta_{kd_j}$ holds by the fact that d is a metric. This implies $(1/\nu_k) \cdot d(p_j,d_j) + \delta_{kp_j} + \delta_{kd_j} < (1/\nu_i) \cdot d(p_j,d_j) + \delta_{ip_j} + \delta_{id_j}$. This contradicts the choice of v_i for r_j made by the algorithm. If $k = i$ holds, we observe $\sum_{l=k_j:h_j} \delta_{is^i_l} \geq \delta_{ip_j} + \delta_{id_j}$ and derive $\sum_{l=k_j:(h_j-1)} d(s^i_l, s^i_{l+1}) < d(p_j,d_j)$ under our assumption. This is, however, not possible because d is a metric. We conclude that the algorithm minimises the tour time of r_j in any feasible plan. As r_j was arbitrarily chosen, the algorithm minimises the tour time of any request. Therefore, it must also minimise totTOUR. Otherwise, there would be a request whose tour time is not minimised by the algorithm and this would lead to a contradiction. ∎

Algorithm 1 minimises the tour time for each request in a feasible plan. Hence, the plan returned by this algorithm minimises further the objective $\arg\min_{\mathscr{P}} \max_{v_i} \max_{r_j} m_j \cdot t_{ij}$. It follows that the price of fairness between this objective and totTOUR is equal to one.

We believe that this corollary makes an objective such as $\arg\min_{\mathscr{P}} \max_{v_i} \max_{r_j} m_j \cdot t_{ij}$ much less appealing from a research perspective. This motivates us to focus on maxTOUR instead. Indeed, computing plans that minimise maxTOUR is much more challenging.

This result holds in instances with two or more vehicles. With just one vehicle, maxTOUR is tractable because it coincides with totTOUR and we can use Algorithm 1 for it. We show the next intractable result via a reduction from the *perfect partition problem (PPP)* [16].

Theorem 8: Unless $\mathsf{NP} = \mathsf{P}$, there are instances where each exact algorithm for maxTOUR is NP-hard.

Proof: Let us give a reduction from the PPP. For given multiset $S = \{n_1, \ldots, n_m\}$ of numbers, this problem asks whether there is a *perfect* partition of S into S_1 and S_2 such that the sum in S_1 is equal to the sum in S_2. We next construct an instance of our model and prove that there is a bijection between the feasible plans of this instance and the solutions to the PPP. For this purpose, we first let $V = \{v_1, v_2\}$ and $R = \{r_1, \ldots, r_m\}$. We consider a line with endpoints $A(0,0)$ and $B(\sum_{j\in[m]} n_j, 0)$. For $i \in [2]$, we set b_i to A, e_i to B, q_i to m, and ν_i to 1. For $j \in [m]$, we set p_j to $A_j(\sum_{l\leq j-1} n_l, 0)$, d_j to $B_j(\sum_{l\leq j} n_l, 0)$, and m_j to 1. Suppose zero delays. We claim that there is a solution to the PPP iff there is a feasible plan of the instance where the maximum vehicle tour time is at most $k = (\sum_{j\in[m]} n_j)/2$.

If there is a feasible plan where the maximum vehicle tour time is at most k, then construct $S_1 = \{n_j = d(A_j,B_j) | r_j \in R_1\}$ and $S_2 = \{n_j = d(A_j,B_j) | r_j \in R_2\}$. As the plan is feasible, it follows $S_1 \cup S_2 = S$ and $S_1 \cap S_2 = \emptyset$. As the vehicle tour time is at most k, it follows $\sum_{n_j\in S_1} n_j \leq k$ and $\sum_{n_j\in S_2} n_j \leq k$. If $\sum_{n_j\in S_1} n_j < k$, then $\sum_{n_j\in S_1} n_j + \sum_{n_j\in S_2} n_j < 2\cdot k$. However, this would contradict $\sum_{n_j\in S_1} n_j + \sum_{n_j\in S_2} n_j = 2\cdot k$. Hence, $\sum_{n_j\in S_1} n_j = k$ must hold. Similarly, we derive $\sum_{n_j\in S_2} n_j = k$. We can now state that S_1 and S_2 form a perfect partition of S because $\sum_{n_j\in S_1} n_j = \sum_{n_j\in S_2} n_j$ holds.

If there is a perfect partition (S_1, S_2) of S such that $\sum_{n_j\in S_1} n_j = \sum_{n_j\in S_2} n_j$ holds, then construct a plan where v_1 services r_j directly via the distance between A_j and B_j for each $n_j \in S_1$ and v_2 services r_j directly via the distance between A_j and B_j for each $n_j \in S_2$. By construction, either v_1 or v_2 services each given request r_j, all requests are serviced, each vehicle could accommodate all requests, and each d_j is serviced immediately after p_j. Hence, this plan is feasible. Furthermore, by construction, v_i services any given r_j before it services any other r_l with $l > j$. Therefore, the total tour time of v_1/v_2 is $\sum_{n_j\in S_1} n_j / \sum_{n_j\in S_2} n_j$. This means that the maximum vehicle tour time is at most k. ∎

This result amplifies further the difference between efficiency and fairness in our setting with respect to totTOUR and maxTOUR, respectively. These two concepts are not only incompatible in general but also efficiency is tractable whereas fairness is intractable.

Minimising totWAIT/maxWAIT is also intractable. In fact, deciding whether each of the corresponding objectives does not exceed a given threshold in at least one plan is in NP-hard, even if there is just one vehicle. To show this result, we give yet another reduction from the TRP.

Theorem 9: Unless $\mathsf{NP} = \mathsf{P}$, there are instances where each exact algorithm for totWAIT/maxWAIT is NP-hard.

Proof: Let us examine the reduction in Theorem 6. In it, the tour time for each request is zero and the delay at each location is zero. As a result, the arrival time for each request is equal to the waiting time for it. Hence, the value of totWAIT/maxWAIT is equal to the value of totARR/maxARR. Consequently, the value of totWAIT/maxWAIT in a feasible plan is at most k iff the value of totARR/maxARR in such a plan is at most k. ∎

VII. A Case Study: Shared Commuting Vehicles

In Berlin, there are nearly 300 000 daily in-commuters [17]. 90 000 of these use private cars. For this reason, we consider 70 000 vehicles, each having capacity 3 and velocity 1 (or the maximum city speed). As each vehicle can accommodate 3 in-commuters, the entire fleet can accommodate all 210 000 in-commuters who use public transport. In contrast to the problems in the previous sections, the requests in this case study arrive over time simply because different people commute at different times of the day. In response, we propose another greedy algorithm (Algorithm 2) that matches in an online fashion each next request that arrives at the system to some feasible and locally optimal vehicle.

Algorithm 2 A feasible plan for shared commuting vehicles.

```
procedure OnlineGreedy(L, D, V)                    ▷ |V| = n
    ∀v_i ∈ V : 𝓡_i ← ()
    ∀v_i ∈ V : R_i ← ∅
    𝓟 ← {𝓡_1, ..., 𝓡_n}
    while the system is not shut down do
        if request r_j arrives at the system then
            L ← L ∪ {p_j, d_j}
            D ← update the distance matrix
            v_i ← feasible and locally optimal for r_j
            𝓡_i ← (s^i_1, ..., s^i_{2|R_i|}, p_j, d_j)
            R_i ← R_i ∪ {r_j}
            send a notification to v_i about r_j
    return 𝓟
```

We refer to Algorithm 2 as ALG TOTWAIT, ALG MAXWAIT, ALG TOTTOUR, ALG MAXTOUR, ALG TOTARR, and ALG MAXARR whenever the feasible and locally optimal decision concerns the minimisation of the corresponding objective. For this task, Algorithm 2 runs a thought experiment where each feasible vehicle services a copy of the new request at the end of its route and, among all such thought routes, it picks a vehicle that minimises the objective. This decision is calculated in $O(n)$ time. For m requests, each algorithm needs $O(m \cdot n)$ time.

We simulated 1 000 runs with all of these algorithms. The 71 000 vehicles were fixed. The 210 000 requests arrived one after another. Their locations were sampled on the fly and uniformly at random from grid $[1\,000] \times [1\,000]$. The pick-up/drop-off times are negligible compared to the other times. For this reason, we set each delay to zero. For each given run, we measured the fairness of ALG MAXWAIT, ALG MAXTOUR, ALG MAXARR and the efficiency of ALG TOTWAIT, ALG TOTTOUR, ALG TOTARR. Figure 1 depicts the average results across all runs.

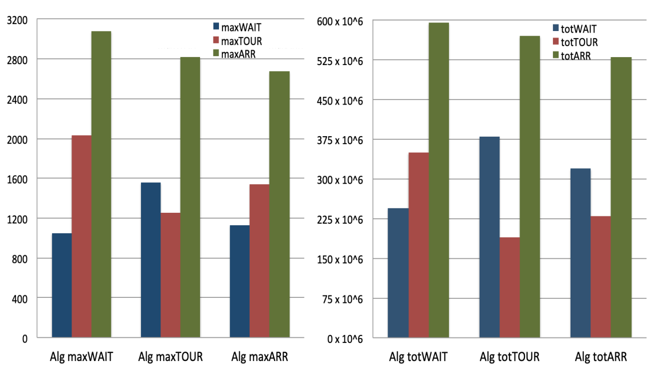

Fig. 1. Key: (left plot) fleet fairness, (right plot) fleet efficiency.

Lastly, we note that Algorithm 2 provides some nice customer properties as well. For example, let us consider the returned plan $\mathscr{P}$ and pick some $v_i \in V$. If $r_j, r_k \in R_i$ and r_j arrived before r_k, then v_i serviced r_j before r_k. This is fair for customers. Also, if $r_j \in R_i$, then v_i serviced r_j from p_j directly to d_j. This is efficient for customers.

VIII. CONCLUSIONS

We considered pickup and delivery problems where a fleet of vehicles services a number of customer requests. For this model, we considered fleet fairness and fleet efficiency. We analysed these two concepts with respect to the waiting, tour, and arrival times of customers. We showed that both concepts might be incompatible in general. For this reason, we measured the loss in efficiency due to obtaining fairness and analysed the complexity of guaranteeing just fleet fairness or just fleet efficiency. We further applied these concepts to a case study. We gave two polynomial-time algorithms. Finally, we showed our results in Table I and Figure 1.

REFERENCES

[1] R. L. Graham, "Bounds for certain multiprocessing anomalies," *Bell System Technical Journal*, vol. 45, no. 9, pp. 1563–1581, 1966.

[2] S. Irnich, "Multi-depot pickup and delivery problem with a single hub and heterogeneous vehicles," *European Journal of Operational Research*, vol. 122, pp. 310–328, April 2000.

[3] M. W. P. Savelsbergh and M. Sol, "The general pickup and delivery problem," *Transportation Science*, vol. 29, no. 1, pp. 17–29, 1995.

[4] T. K. Ralphs, L. Kopman, W. R. Pulleyblank, and L. E. Trotter, "On the capacitated vehicle routing problem," *Mathematical Programming*, vol. 94, no. 2-3, pp. 343–359, 2003.

[5] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, pp. 80–91, October 1959.

[6] S. Parragh, K. Doerner, and R. Hartl, "A survey on pickup and delivery problems: Part i: Transportation between customers and depot," *Journal für Betriebswirtschaft*, vol. 58, pp. 21–51, April 2008.

[7] ——, "A survey on pickup and delivery problems: Part ii: Transportation between pickup and delivery locations," *Journal für Betriebswirtschaft*, vol. 58, pp. 81–117, June 2008.

[8] A. Gibbard, "Manipulation of voting schemes: A general result," *Econometrica*, vol. 41, no. 4, pp. 587–601, 1973.

[9] J. C. Harsanyi, "Games with incomplete information played by "Bayesian" players, i-iii. part i. the basic model," *Management Science*, vol. 14, no. 3, pp. 159–182, 1967.

[10] M. A. Satterthwaite, "Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions," *Journal of Economic Theory*, vol. 10, no. 2, pp. 187–217, 1975.

[11] T. Vidal, G. Laporte, and P. Matl, "A concise guide to existing and emerging vehicle routing problem variants," *European Journal of Operational Research*, vol. 286, no. 2, pp. 401–416, 2020.

[12] S. Nucamendi, Y. Cardona-Valdes, and F. Angel-Bello Acosta, "Minimizing customers waiting time in a vehicle routing problem with unit demands," *Journal of Computer and Systems Sciences International*, vol. 54, no. 6, pp. 866–881, 2015.

[13] D. Bertsimas, V. F. Farias, and N. Trichakis, "The price of fairness," *Operations Research*, vol. 59, no. 1, pp. 17–31, 2011.

[14] F. Afrati, S. Cosmadakis, C. H. Papadimitriou, G. Papageorgiou, and N. Papakostantinou, "The complexity of the travelling repairman problem," *RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications*, vol. 20, no. 1, pp. 79–87, 1986.

[15] J. Fakcharoenphol, C. Harrelson, and S. Rao, "The k-traveling repairman problem," in *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA 03, 2003, pp. 655–664.

[16] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[17] "Broschüre Mobilität der Stadt - Berliner Verkehr in Zahlen 2017," 2017, visited on 31.05.2021. [Online]. Available: https://www.berlin.de/sen/uvk/verkehr/verkehrsdaten/zahlen-und-fakten/mobilitaet-der-stadt-berliner-verkehr-in-zahlen-2017