Fairness for Drivers with Additive Profits in Emerging Vehicle Routing Problems

Martin Aleksandrov
Technische Universität Berlin
Ernst-Reuter-Platz, 10587 Berlin, Germany
martin.aleksandrov@tu-berlin.de

Abstract

We consider semi-decentralised fair divisions in the context of emerging VRPs, where not just the preferences of drivers play a crucial role, but also the feasibilities of their vehicles. For such settings, we propose three new fairness notions: responsive FEF1, responsive FEQX, and responsive FEFX, which capture the responsiveness of drivers for requests. For such settings, we also give two new algorithms. Our first algorithm returns responsive FEF1 assignments. Our second algorithm returns responsive FEQX assignments, as well as responsive FEFX assignments in a practical case.

1 Introduction

Let us consider the classical Vehicle Routing Problem (VRPs) (Dantzig and Ramser 1959). In this problem, there is a single vehicle and a set of visit requests. Generalisations of the VRP consider a fleet of multiple vehicles and a set of pickup-and-delivery requests (Savelsbergh and Sol 1995). We initiate a study of emerging VRPs. Applications of such problems include autonomous vehicles, connected vehicles, electric vehicles, garbage vehicles, and data-driven logistics. The 2020 EU Strategy for Sustainable and Smart Mobility has formulated public mobility Transport Policy Flagships, according to which the transition to emerging VRPs *must* involve the preferences of individuals. One objective in this strategy is achieving trust. Among explainability and safety, trust requires that vehicles are used fairly.

In this paper, we provide an early qualitative analysis of fairness for drivers. We thus propose a fair division assignment model, where a fleet of *vehicles* and a set of *requests* are available in a fixed time interval, and each driver charge the customer of each given request with some cost. We consider *driver-dependent costs* (i.e. request costs depend on drivers) and *driver-independent costs* for customers. We also consider *additive profits* for drivers (i.e. the profit for some requests is sum of the request costs). Like in many models for fair division (Brams and Taylor 1996) and fair public decision making (Conitzer, Freeman, and Shah 2017), additivity is a common assumption. Unlike many such models, we consider two additional features: *vehicle feasibilities* and *driver responsiveness*.

That is, a given vehicle may be feasible or not for a given request. We model this by using a hard *feasibility* indicator. For instance, suppose that a given vehicle is feasible only for packages that can be loaded inside its trunk, subject to maximising the total number of packages. This is known as the loading problem and it is NP-hard in general (Männel and Bortfeldt 2018). In this context, if a given request can be loaded in the trunk of a given vehicle in some solution to the loading problem, then we set the corresponding feasibility indicator to one, and else we set it to zero. We partly overcome such intractabilities by decentralising some of the feasibilities of vehicles and letting their drivers decide whether they are feasible or not for requests. In addition, we consider the possibility of decentralising not just hard feasibilities of vehicles, but also some of the profit preferences of their drivers. Indeed, many real-world applications are actually semi-decentralised. In such applications, drivers can therefore be *responsive* or not.

Realistic features such as feasibilities and responsiveness require that we modify centralised fairness notions and centralised fairness algorithms so that they reflect the nature of our model. We do such modifications in our work. Our model can thus be simulated in centralised, decentralised, as well as on various Internet and mobile settings, where some request information is known and some missing information is revealed as drivers are prompted for responses. If drivers respond, then they are online. Otherwise, they are offline until the next time they are prompted for responses. For example, the dispatchers of Bonds Express often communicate for work with drivers via SMS messages (Aleksandrov et al. 2013). We next give an outline of our paper.

Outline: We explain our contributions in Section 2 and review related literature in Section 3. We define formally our model in Section 4. For this model, in Sections 5, 6, and 7, we show that existing fairness notions such as EF1, EQX, and EFX need to be modified. In response, we propose three new notions: *responsive FEF1*, *responsive FEQX*, and *responsive FEFX*. With driver-dependent costs, we prove in Section 8 that a responsive FEF1 assignment can be computed in polynomial time and in Section 9 that a responsive FEQX assignment can also be computed in polynomial time. With driver-independent costs, we give in Section 10 a similar tractability result for a responsive FEFX assignment. Finally, we draw our conclusions in Section 11.

2 Contributions

In our assignment model, we let drivers (pre-)submit to the (central) planner some but possibly not all of their profit preferences and vehicle feasibilities. For example, Bonds Express contracts on-demand drivers, and the dispatchers often do not know all of their preferences, or all of their vehicle feasibilities, for requests (Aleksandrov et al. 2013). If the planner have complete such information, then the setting is purely *centralised*. If they have no such information at all, then the setting is purely *decentralised*. Otherwise, the planner need to decentralise some of the assignment decisions. We note that the decentralisation of such decisions is challenging simply because we do not know how drivers might behave in practice. At this point, our model intersects behavioral game theory (Camerer 2003), where *rationality* is a key concept that is suitable for settings in which drivers act in their best interest. In our work, we consider rationality concepts such as truthfulness and selfishness.

Thus, in our *semi-decentralised* setting, whenever some profit preferences or vehicle feasibilities are unknown to the planner, they send some (at most m) requests to drivers, and drivers may respond for such requests with such information within some (pre-)specified time $T \in \Omega(m)$. We assume that drivers are truthful and maximise their profit. More specifically, if drivers are *responsive* to a request they receive, then we assume that their behavior is *truthful* and *profit-maximising*, i.e. supposing that their vehicles are truthfully feasible for such a request, they reveal information for some truthfully most profitable among their truthfully feasible such requests, without changing any (*public*) preferences and feasibilities, which are known to the planner, and accounting for any (*private*) preferences and feasibilities, which are known just to them. Otherwise, we assume that drivers are *unresponsive* and cannot service requests until the next time they respond. Drivers can be unresponsive for various reasons: they depart from and arrive at the market at different times; they have made sufficient profit in the current day and decide to go home earlier; their vehicles are truthfully infeasible. For such settings, we study fairness for drivers with additive profits.

In fair division of goods, three fairness notions are "envy-freeness up to some good" (EF1) (Budish 2011), "equitability up to every good" (EQX) (Freeman et al. 2019), and "envy-freeness up to every good" (EFX) (Caragiannis et al. 2019). In our model, these notions ignore the vehicle feasibilities or driver responsiveness. This motivates us to define three new *responsive* versions of them: *responsive EF1*, *responsive EFX*, and *responsive EQX*, and three new *feasible* versions of them: *FEF1*, *FEFX*, and *FEQX*. We show that there are settings where all feasible assignments do not satisfy responsive EF1/EQX/EFX, as well as settings where all responsive assignments do not satisfy FEF1/FEQX/FEFX. We define therefore *feasible* and *responsive* versions of EF1, EQX, and EFX: *responsive FEF1*, *responsive FEQX*, and *responsive FEFX*. For any two drivers, responsive FEF1/FEFX assignments bound the absolute *envy* (i.e. subjective profit difference) for their feasible requests. For any two drivers, responsive FEQX assignments bound the absolute *jealousy* (i.e. objective profit difference) for their feasible requests.

Furthermore, these feasible and responsive fairness notions account for the fact that drivers can have feasible envy or jealousy for an assigned request only if they are not unresponsive when the request is assigned. We further give algorithms for returning assignments that satisfy these notions. In particular, for settings with driver-dependent costs, we first give a polynomial-time algorithm (Algorithm 1) for returning feasible and responsive FEF1 assignments: Theorem 1. We then give another polynomial-time algorithm (Algorithm 2) for returning feasible and responsive FEQX assignments: Theorem 2. For settings with driver-independent costs, the notion of responsive FEQX coincides with the notion of responsive FEFX. In this practical domain, Algorithm 1 may fail to return such assignments: Lemma 1, whereas Algorithm 2 is guaranteed to return such assignments: Lemma 2. Table 1 summarises our main results.

driver-dependent costs			
result	**rule**	**property**	**run-time**
Thm 1	Alg 1	responsive FEF1	$O(m \max\{nm, T\})$
Thm 2	Alg 2	responsive FEQX	$O(m \max\{nm, T\})$

driver-independent costs			
result	**rule**	**property**	**satisfaction**
Lem 1	Alg 1	responsive FEFX	"no"
Lem 2	Alg 2	responsive FEFX	"yes"

Table 1: A summary of results for truthful and profit-maximising drivers with additive profits: n vehicles, m requests, and deadline $T \in \Omega(m)$.

3 Related work

Our model generalises fair division of goods from (Brams and Taylor 1996) by adding vehicle feasibilities and driver responsiveness. Thus, we consider three layers of profit, feasibility, and responsive preferences. Multi-layer preferences were recently announced to broaden the research agenda of COMSOC (Boehmer and Niedermeier 2021). Unlike existing works for EF1, EQX, and EFX, our work is about feasible and responsive fairness in semi-decentralised settings that intersect behavioral game theory (Camerer 2003). For VRPs, Rheingans-Yoo et al. (2019) analysed matchings with driver location (not profit) preferences. Ma et al. (2019) modelled spatio-temporal settings with a focus on drivers. Xu and Xu (2020) investigated trading the system efficiency for the income equality of drivers. By comparison, we focus on the profit preferences of drivers and the feasibilities of their vehicles. For fair division, Dror, Feldman, and Segal-Halevi (2021) considered a model, where agents have categories and upper quotas. The authors studied F-EF1 assignments for centralised settings. By comparison, we study responsive FEF1, responsive FEQX, and responsive FEFX for semi-decentralised settings. In (Gerding et al. 2019) and (Kash, Procaccia, and Shah 2014), each agent have fixed arrival and departure times. In our setting, drivers can also arrive and depart but we may not know when. In each of these settings, the (private) preferences do not change over time.

4 Preliminaries

Vehicles: We let $V = \{v_1, \ldots, v_n\}$ denote the driver *vehicles*, where each v_i has start/finish location $s_i \in \mathbb{R}^2$/$f_i \in \mathbb{R}^2$ and capacity $q_i \in \mathbb{N}_{>0}$. The locations could respectively denote depot locations, or a request location submitted in the past and a request location predicted in the future.

Requests: We let $R = \{r_1, \ldots, r_m\}$ denote the customer *requests*, where each $r_j = (p_j, d_j, m_j)$ has begin/end location $p_j \in \mathbb{R}^2$/$d_j \in \mathbb{R}^2$ and demand $m_j \in \mathbb{N}_{>0}$. If $p_j \neq d_j$, r_j requires pickup and delivery (e.g. courier services, taxi services). Otherwise, r_j requires a visit (e.g. home services).

Preferences: We consider *driver-dependent costs*. For servicing each r_j, we let the driver of each v_i charge some cost $c_{ij} \in \mathbb{R}_{>0}$. Commonly, costs are proportional to the request service times. We also consider *driver-independent costs*. That is, $c_{ij} = c_j$ for each v_i and each r_j. We suppose that drivers have *additive profits*. That is, for $S \subseteq R$, the profit of the driver of v_i is $c_i(S) = \sum_{r_j \in S} c_{ij}$. Each cost c_{ij} can be public or private. Thus, we let $\tilde{c}_{ij} = c_{ij}$ if c_{ij} is public and $\tilde{c}_{ij} = unk$ if c_{ij} is private (i.e. it is *unk*nown to the planner). The *preferences* are $P = (\tilde{c}_{ij})_{n \times m}$.

Feasibilities: We consider some set of constraints C_{ij} for each v_i and each r_j. Package dimensions and driver shifts are common constraints. Thus, we can define a hard feasibility indicator f_{ij}: $f_{ij} = 1$ if all constraints in C_{ij} can be satisfied; $f_{ij} = 0$ otherwise. We suppose that the capacity feasibility constraint $[(m_j \leq q_i)?1; 0]$ belongs to C_{ij}. That is, if $q_i < m_j$, then $f_{ij} = 0$ holds. However, if $f_{ij} = 0$, then $m_j \leq q_i$ might hold but other constraints in C_{ij} could be violated. As for preferences, each f_{ij} can be public or private. We let $\tilde{f}_{ij} = f_{ij}$ if f_{ij} is public, and else $\tilde{f}_{ij} = unk$. The *feasibilities* are $F = (\tilde{f}_{ij})_{n \times m}$.

Feasible assignments: In practice, Bonds Express cannot service all requests within a single fixed time interval and, for this reason, schedule any remaining requests for the next time interval, even though their vehicles might be feasible for such requests (Aleksandrov et al. 2013). As a result, the current assignment may not give all such requests to drivers.

More formally, assignment is $\mathcal{R} = (R_1, \ldots, R_n)$, where $R_i \subseteq R$ for each v_i and $R_i \cap R_j = \emptyset$ for each (v_i, v_j) such that $i \neq j$. We say that $(R_1, \ldots, R_n)$ is *feasible* if, for each v_i and each $r_j \in R_i$, $f_{ij} > 0$ holds. Feasible assignments are such that no r_j s.t. $f_{ij} = 0$ holds for each v_i is assigned.

Algorithms: We consider *semi-decentralised* algorithms that assign requests to drivers if the algorithms have all the private request information of drivers, or else prompt drivers for some such information, and drivers reveal it.

An algorithm returns some $(R_1, \ldots, R_n)$ and *responsiveness matrix* $U = (\tilde{u}_{ij})_{n \times m}$. We can think of matrix U as a tool for capturing the responsive behavior of drivers. For each r_j, whenever $r_j \in R_i$, $\tilde{u}_{kj} = 1$ holds for each driver k (also i) who is not unresponsive and $\tilde{u}_{hj} = 0$ holds for each driver $h \neq k$ who is unresponsive.

$(R_1, \ldots, R_n)$ is *responsive complete* wrt matrix U if, for every $r_j \in R$ s.t. $f_{ij} > 0$ and $\tilde{u}_{ij} = 1$ hold for some $v_i \in V$, $r_j \in R_k$ for some $v_k \in V$ with $f_{ik} > 0$ and $\tilde{u}_{ik} = 1$. We consider algorithms that return such assignments.

5 Feasibilities and EF1, EQX, EFX

In this section, we compare feasible assignments and EF1, EQX, and EFX assignments. We begin with EF1. As we mentioned, EF1 does not account for any feasibilities.

Definition 1. *$(R_1, \ldots, R_n)$ is* EF1 *if, for each $v_i, v_k \in V$ s.t. $R_k \neq \emptyset$, $c_i(R_i) \geq c_i(R_k \setminus \{r_j\})$ holds for some $r_j \in R_k$.*

There are settings where each EF1 assignment is not feasible and each feasible assignment is not EF1. We illustrate this in Example 1.

Example 1. *In a centralised setting, let there be vehicles v_1, v_2 and requests r_1, r_2. Also, define any strictly positive costs, and $f_{11} = f_{12} = 1$ but $f_{21} = f_{22} = 0$. For example, the costs could be driver-independent and all equal to* 1. *We make two observations for this setting.*

Firstly, to achieve EF1, we observe that we must assign exactly one request to each vehicle. There are two drivers and two requests, so there are two EF1 assignments. Hence, each such assignment gives some infeasible request to v_2. Such an outcome cannot be feasible.

Secondly, the unique feasible assignment gives both requests to v_1. However, this outcome violates EF1 because the costs are strictly positive, and removing any single request from the bundle of the driver of v_1 does not eliminate the envy of the driver of v_2.

In centralised settings, it follows by Example 1 that the classical EF1 round-robin rule from (Caragiannis et al. 2019) may not return a feasible assignment.

We next show a similar result for EQX. For this reason, we define EQX. Again, as we mentioned, as EF1, this notion does not account for any feasibilities.

Definition 2. *$(R_1, \ldots, R_n)$ is* EQX *if, for each $v_i, v_k \in V$ s.t. $R_k \neq \emptyset$, $c_i(R_i) \geq c_k(R_k \setminus \{r_j\})$ holds for every $r_j \in R_k$.*

In Example 1, each EQX assignment gives one request to each vehicle and falsifies feasibility, whereas the unique feasible assignment falsifies EQX due to similar reasons as for EF1. In centralised settings, it also follows by Example 1 that the EQX rule from (Freeman et al. 2019) is not guaranteed to return a feasible assignment.

Lastly, we confirm a similar incompatibility for EFX. We thus define it. As EF1 and EQX, the notion of EFX does not account for any feasibilities.

Definition 3. *$(R_1, \ldots, R_n)$ is* EFX *if, for each $v_i, v_k \in V$ s.t. $R_k \neq \emptyset$, $c_i(R_i) \geq c_i(R_k \setminus \{r_j\})$ holds for every $r_j \in R_k$.*

In Example 1, each EFX assignment gives one request to each vehicle and, moreover, is infeasible, whilst the unique feasible assignment violates EFX. In centralised settings, it also follows by Example 1 that the popular EFX leximin rule from (Plaut and Roughgarden 2018) may not return a feasible assignment.

Finally, we conclude that EF1, EQX, and EFX must be modified in order to account for the feasibilities of vehicles for requests.

6 Responsiveness and EF1, EQX, EFX

In this section, if driver 1 never respond and driver 2 always do, then *no* semi-decentralised algorithm can guarantee to return EF1, EQX, or EFX assignments. In Example 2, we demonstrate this.

Example 2. *In a decentralised setting, let there be v_1, v_2 and r_1, r_2. Pick* unk *public costs and* unit *public feasibilities. As the private costs are unknown, any semi-decentralised algorithm must prompt drivers for their private costs in some fixed sequence of calls (e.g. 12, 21, 22, etc.). Suppose that driver 1 never respond and driver 2 always do.*

If we suppose that the algorithm returns one of the assignments $(\{r_1\}, \{r_2\})$ (EF1, EQX, and EFX), $(\{r_2\}, \{r_1\})$ (EF1, EQX, and EFX), or $(\{r_1, r_2\}, \emptyset)$ (not EF1, EQX, or EFX), then it must have prompted driver 1 for their private costs. But, if it has prompted driver 1 for their private costs and assigned some r_j to driver 1 anyway, then it must have ignored the fact that driver 1 never respond and, hence, it cannot be semi-decentralised. This leads to a contradiction. We conclude that it cannot assign any request r_j to driver 1. Consequently, it must assign every r_j to driver 1.

More specifically, the algorithm assigns r_1, r_2 to driver 2 after they are prompted for responses and reveal their private costs. This assignment violates EF1, EQX, or EFX because driver 1 is still envious and jealous even after the removal of any request from 2's bundle. Hence, the algorithm does not return a fair assignment in this setting. □

Finally, we conclude that EF1, EQX, and EFX must be modified in order to account for the responsive behavior of drivers.

7 Feasibilities or Responsiveness

In this section, we discuss that modifying EF1, EQX, and EFX to account just for the feasibilities of vehicles for requests or just for the responsive behavior of drivers is not sufficient for returning feasible and responsive assignments.

We first show this for three new notions that account just for the responsive behavior of drivers: responsive EF1, responsive EQX, and responsive EFX.

Definition 4. $(R_1, \ldots, R_n)$ *is* responsive EF1 *wrt matrix* $U = (\tilde{u}_{ij})_{n\times m}$ *if, for each* $v_i, v_k \in V$ *where* $R_{ii}^U = \{r_j \in R_i | \tilde{u}_{ij} = 1\}$ *and* $R_{ik}^U = \{r_j \in R_k | \tilde{u}_{ij} = 1\} \neq \emptyset$, $c_i(R_{ii}^U) \geq c_i(R_{ik}^U \setminus \{r_j\})$ *holds for some* $r_j \in R_{ik}^U$.

Definition 5. $(R_1, \ldots, R_n)$ *is* responsive EQX *wrt matrix* $U = (\tilde{u}_{ij})_{n\times m}$ *if, for each* $v_i, v_k \in V$ *where* $R_{ii}^U = \{r_j \in R_i | \tilde{u}_{ij} = 1\}$ *and* $R_{ik}^U = \{r_j \in R_k | \tilde{u}_{ij} = 1\} \neq \emptyset$, $c_i(R_{ii}^U) \geq c_k(R_{ik}^U \setminus \{r_j\})$ *holds for every* $r_j \in R_{ik}^U$.

Definition 6. $(R_1, \ldots, R_n)$ *is* responsive EFX *wrt matrix* $U = (\tilde{u}_{ij})_{n\times m}$ *if, for each* $v_i, v_k \in V$ *where* $R_{ii}^U = \{r_j \in R_i | \tilde{u}_{ij} = 1\}$ *and* $R_{ik}^U = \{r_j \in R_k | \tilde{u}_{ij} = 1\} \neq \emptyset$, $c_i(R_{ii}^U) \geq c_i(R_{ik}^U \setminus \{r_j\})$ *holds for every* $r_j \in R_{ik}^U$.

In centralised settings where each driver is responsive to each request (i.e. $U = (1)_{n\times m}$), an assignment is responsive EF1/EQX/EFX iff it is EF1/EQX/EFX. These follow by definition. By Example 1, it follows that no feasible assignment can ever satisfy responsive EF1/EQX/EFX.

We next show a similar result for three new notions that accound just for the feasibilities of vehicles for requests: FEF1, FEQX, and FEFX.

Definition 7. $(R_1, \ldots, R_n)$ *is* FEF1 *if, for each* $v_i, v_k \in V$ *where* $F_{ii} = \{r_j \in R_i | f_{ij} > 0\}$ *and* $F_{ik} = \{r_j \in R_k | f_{ij} > 0\} \neq \emptyset$, $c_i(F_{ii}) \geq c_i(F_{ik} \setminus \{r_j\})$ *holds for some* $r_j \in F_{ik}$.

Definition 8. $(R_1, \ldots, R_n)$ *is* FEQX *if, for each* $v_i, v_k \in V$ *where* $F_{ii} = \{r_j \in R_i | f_{ij} > 0\}$ *and* $F_{ik} = \{r_j \in R_k | f_{ij} > 0\} \neq \emptyset$, $c_i(F_{ii}) \geq c_k(F_{ik} \setminus \{r_j\})$ *holds for every* $r_j \in F_{ik}$.

Definition 9. $(R_1, \ldots, R_n)$ *is* FEFX *if, for each* $v_i, v_k \in V$ *where* $F_{ii} = \{r_j \in R_i | f_{ij} > 0\}$ *and* $F_{ik} = \{r_j \in R_k | f_{ij} > 0\} \neq \emptyset$, $c_i(F_{ii}) \geq c_i(F_{ik} \setminus \{r_j\})$ *holds for every* $r_j \in F_{ik}$.

In settings where each vehicle is feasible for each request (i.e. unit feasibilities), an assignment is FEF1/FEQX/FEFX iff it is EF1/EQX/EFX. These follow by definition. By Example 2, it follows that no semi-decentralised algorithm can ever return FEF1/FEQX/FEFX assignments.

Finally, we conclude that EF1, EQX, and EFX must be modified, and account simultaneously for both the vehicle feasibilities and driver responsiveness. We do exactly this in the next sections.

8 Responsive FEF1

In our semi-decentralised setting, infeasibility and unresponsiveness seem harmful for guaranteeing EF1. However, these features take a vital part of any real-world system. For this reason, we propose to integrate the vehicle feasibilities and driver responsiveness into the definition of EF1.

Thus, for any pair of drivers i and k, the new subjective responsive FEF1 requires eliminating any feasible envy that i might have of k by removing a request from k's bundle of i's feasible requests, that is assigned to k when driver i is responsive.

Definition 10. $(R_1, \ldots, R_n)$ *is* responsive FEF1 *wrt matrix* $U = (\tilde{u}_{ij})_{n\times m}$ *if, for each* $v_i, v_k \in V$ *where* $F_{ii}^U = \{r_j \in R_i | f_{ij} > 0, \tilde{u}_{ij} = 1\}$ *and* $F_{ik}^U = \{r_j \in R_k | f_{ij} > 0, \tilde{u}_{ij} = 1\} \neq \emptyset$, $c_i(F_{ii}^U) \geq c_i(F_{ik}^U \setminus \{r_j\})$ *holds for some* $r_j \in F_{ik}^U$.

In Example 1, each driver respond, i.e. $U = \left(\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right)$. The unique feasible assignment gives both requests to the first driver. This assignment is also responsive FEF1 because the vehicle of the driver who receive no request is infeasible for every request and, therefore, its driver does not have feasible envy of the driver who receive both requests.

In Example 2, the returned assignment is feasible and responsive FEF1 only if the first driver responds for at most one request, i.e. $U = \left(\begin{smallmatrix} 0 & 0 \\ 1 & 1 \end{smallmatrix}\right)$, $\left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)$, or $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 1 \end{smallmatrix}\right)$. Indeed, if they respond for both requests, i.e. $U = \left(\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right)$, then this assignment would induce feasible envy for the first driver even after removing one request from the second driver's bundle.

Further, we give an algorithm for computing feasible and responsive FEF1 assignments in every setting of our model: Algorithm 1. This simulates a *round-robin* order by using a driver counter at each iteration. However, unlike the classical round-robin, this new algorithm accounts for both the vehicle feasibilities and driver responsiveness.

As the computation of the algorithm evolves, a given driver may depart from (i.e. become "unresponsive") or arrive at (i.e. become "responsive") the market. However, we do not know the iterations at which they may depart from or arrive at the market. From this perspective, we can view the algorithm as a tool for achieving FEF1 between any two drivers over the total numbers of the respective assignment iterations at which they are responsive at the market. Furthermore, the returned assignment is not just feasible and responsive FEF1, but also responsive complete.

Algorithm 1: SEMI-DECENTRALISED ROUND-ROBIN

Input: V, R, P, F, deadline $T \in \Omega(|R|)$
Output: an assignment that is feasible, responsive complete, and responsive FEF1

$\forall v_i \in V : R_i \leftarrow \emptyset$
$\forall v_i \in V$: mark driver i as "responsive"
$U = (\tilde{u}_{ij})_{n\times m} \leftarrow (1)_{n\times m}$ ▷ responsiveness matrix
$i \leftarrow 1$ ▷ a driver counter
while $R \neq \emptyset$ **do**
  if all drivers are "unresponsive" **then** ▷ termination
    exit
  if $i > n$ **then** ▷ end of round
    $i \leftarrow 1$
    continue
  if $\exists r_k \in R : \tilde{c}_{ik} = unk \vee \tilde{f}_{ik} = unk$ **then**
    send R to driver i ▷ decentralisation
    if within T, driver i reply with r_j, c_{ij}, f_{ij} **then**
      mark driver i as "responsive"
      $\tilde{c}_{ij} \leftarrow c_{ij}, \tilde{f}_{ij} \leftarrow f_{ij}$
      $R_i \leftarrow R_i \cup \{r_j\}, R \leftarrow R \setminus \{r_j\}$
      $\forall k \in \{h | h$ is "unresponsive"$\} : \tilde{u}_{kj} \leftarrow 0$
    else
      mark driver i as "unresponsive"
    $i \leftarrow (i + 1)$
  else
    if $\exists r_k \in R : f_{ik} > 0$ **then** ▷ centralisation
      mark driver i as "responsive"
      $r_j \leftarrow \arg\max_{r_k \in R: f_{ik}>0} c_{ik}$
      $R_i \leftarrow R_i \cup \{r_j\}, R \leftarrow R \setminus \{r_j\}$
      $\forall k \in \{h | h$ is "unresponsive"$\} : \tilde{u}_{kj} \leftarrow 0$
    else
      mark driver i as "unresponsive"
    $i \leftarrow (i + 1)$
return $[(R_1, \ldots, R_n), U]$

As we mentioned in the introduction, we consider truthful and profit-maximising behavior of responsive drivers. That is, whenever some driver i respond for $r_j \in R$ with c_{ij} and f_{ij}, we assume that $\{r_h \in R | f_{ih} > 0\} \neq \emptyset$ and $r_j \in \{r_k \in R | c_{ik} = \arg\max_{r_h \in R: f_{ih}>0} c_{ih}\}$ hold.

Theorem 1. *In our semi-decentralised setting, Algorithm 1 returns in $O(m \cdot \max\{n \cdot m, T\})$ time an assignment that is feasible, responsive complete and responsive FEF1 wrt the returned matrix, supposing that drivers are truthful and profit-maximising whenever they respond.*

Proof. The complexity is dominated by at most m iterations. At each of these, the current driver i enters either the centralised phase or the decentralised phase in $O(n \cdot m)$ time. The former phase takes time $O(\max\{n, m\})$. The latter phase takes time $O(\{n, T\})$, during which driver i receive the set R of unassigned requests, and they may reveal some $r_j \in R$. In this case, they are truthful and profit-maximising. Hence, $r_j \in \{r_k \in R | c_{ik} = \max_{r_h \in R: f_{ih}>0} c_{ih}\} \neq \emptyset$ holds. Thus, they may need to consider at most m candidate feasible requests but, as $T \in \Omega(m)$, it follows that the overall time is $O(m \cdot \max\{n \cdot m, T\})$.

Drivers pick feasible requests either in the centralised phase or once they respond in the decetralised phase. This continues until termination. Consequently, it is easy to derive that the returned assignment $(R_1, \ldots, R_n)$ is feasible and responsive complete wrt $U = (\tilde{u}_{ij})_{n\times m}$. We next therefore show that this assignment is also responsive FEF1 wrt $U = (\tilde{u}_{ij})_{n\times m}$. Pick two drivers i and k such that $i \neq k$. We note that $i \leq n, k \leq n$ hold. We let $r_i(1), \ldots, r_i(h_i)$ denote the 1st, …, h_ith requests picked up by driver i, and $r_k(1), \ldots, r_k(h_k)$ denote the 1st, …, h_kth requests picked up by driver k. Here $h_i, h_k \in \mathbb{N}_{\geq 0}$. We consider two cases.

Case 1: We let $i < k$ hold. That is, driver i pick up before driver k. We also let l_i denote the number of requests in $r_k(1), \ldots, r_k(h_k)$ for which v_i is feasible and not unresponsive. Wlog, for $l_i \geq 1$, let these be $r_k(j_1), \ldots, r_k(j_{l_i})$ where $1 \leq j_1 \leq \ldots \leq j_{l_i} \leq h_k$ and be picked up at rounds $s_1, \ldots, s_{l_i}$, respectively.

For any fixed l between 1 and l_i, we have that $f_{ij_l} > 0, \tilde{u}_{ij_l} = 1$ hold. Furthermore, driver i must have picked up a request in round s_l. Otherwise, they would be infeasible or unresponsive for $r_k(j_l)$ and, hence, $f_{ij_l} = 0$ or $\tilde{u}_{ij_l} = 0$ would hold. This would lead to a contradiction because $f_{ij_l} > 0, \tilde{u}_{ij_l} = 1$ hold. It follows that $l_i \leq h_i$ holds.

As drivers are profit-maximising, driver i pick feasible requests in some non-increasing cost sequence. Thus, as $i < k$, $c_i(r_i(l)) \geq c_i(r_k(l))$ follow for each $l = 1 : l_i$. Also, as $j_1 \geq 1, \ldots, j_{l_i} \geq l_i$, $c_i(r_k(l)) \geq c_i(r_k(j_l))$ follow for each $l = 1 : l_i$. These inequalities imply $c_i(r_i(l)) \geq c_i(r_k(j_l))$ for each $l = 1 : l_i$.

We let $F^U_{ii} = \{r_i(1), \ldots, r_i(h_i)\}$ and $F^U_{ik} = \{r_k(j_1), \ldots, r_k(j_{l_i})\}$. We thus derive $c_i(F^U_{ii}) = \sum_{l=1:h_i} c_i(r_i(l)) \geq \sum_{l=1:l_i} c_i(r_i(l)) \geq \sum_{l=1:l_i} c_i(r_k(j_l)) = c_i(F^U_{ik})$. If $l_i = 0$, then $F^U_{ik} = \emptyset$ and $c_i(F^U_{ii}) \geq c_i(F^U_{ik}) = c_i(\emptyset) = 0$ hold. Otherwise, $F^U_{ik} \neq \emptyset$ and $c_i(F^U_{ii}) \geq c_i(F^U_{ik}) \geq c_i(F^U_{ik} \setminus \{r\})$ for each $r \in F^U_{ik}$ hold. Hence, the notion holds for driver i.

We next show that the notion also holds for driver k. We note that driver k pick up after driver i. We let l_k denote the number of requests in $r_i(1), \ldots, r_i(h_i)$ for which v_k is feasible and not unresponsive. Wlog, for $l_k \geq 1$, let these be $r_i(j_1), \ldots, r_i(j_{l_k})$ with $1 \leq j_1 \leq \ldots \leq j_{l_k} \leq h_i$ and be picked up at rounds $s_1, \ldots, s_{l_k}$, respectively.

For any l between 2 and l_k, $f_{kj_l} > 0, \tilde{u}_{kj_l} = 1$ hold. Also, driver k must have picked up a request in round s_{l-1}. Otherwise, they would be infeasible or unresponsive for $r_i(j_l)$ and, hence, $f_{kj_l} = 0$ or $\tilde{u}_{kj_l} = 0$ would hold. This would lead to a contradiction because $f_{kj_l} > 0, \tilde{u}_{kj_l} = 1$ hold. It follows that $(l_k - 1) \leq h_k$ holds.

As $i < k$ and driver k pick feasible requests in some non-increasing cost sequence, $c_k(r_k(l)) \geq c_k(r_i(l+1))$ follow for each $l = 1 : (l_k - 1)$. Also, as $j_1 \geq 1, \ldots, j_{l_k} \geq l_k$, $c_k(r_i(l+1)) \geq c_k(r_i(j_{l+1}))$ follow for each $l = 1 : (l_k - 1)$. These inequalities imply $c_k(r_k(l)) \geq c_k(r_i(j_{l+1}))$ for each $l = 1 : (l_k - 1)$.

We let $F^U_{kk} = \{r_k(1), \ldots, r_k(h_k)\}$ and $F^U_{ki} = \{r_i(j_1), \ldots, r_i(j_{l_k})\}$. We derive $c_k(F^U_{kk}) = \sum_{l=1:h_k} c_k(r_k(l)) \geq \sum_{l=1:(l_k-1)} c_k(r_k(l)) \geq \sum_{l=1:(l_k-1)} c_k(r_i(j_{l+1}))$. The latter sum is $c_k(F^U_{ki} \setminus \{r_i(j_1)\})$. If $l_k = 0$, then $F^U_{ki} = \emptyset$ and $c_k(F^U_{kk}) \geq c_k(\emptyset) = 0$ hold. Otherwise, $F^U_{ki} \neq \emptyset$ and $c_k(F^U_{kk}) \geq c_k(F^U_{ki} \setminus \{r_i(j_1)\})$ for $r_i(j_1) \in F^U_{ki}$ hold.

Case 2: We let $i > k$ hold. We now go to Case 1 with $i = k$ and $k = i$, and follow the proof of Case 1. The result follows. □

9 Responsive FEQX

Similarly as for FEF1, we propose to integrate the feasibilities and responsiveness into the definition of EQX. Thus, for any pair of drivers i and k, the new objective FEQX requires eliminating any feasible jealousy that i might have of k by removing a request from k's bundle of i's feasible requests, that is assigned to k when driver i is responsive.

Definition 11. $(R_1, \ldots, R_n)$ *is* responsive FEQX *wrt matrix* $U = (\tilde{u}_{ij})_{n \times m}$ *if, for each* $v_i, v_k \in V$ *where* $F^U_{ii} = \{r_j \in R_i | f_{ij} > 0, \tilde{u}_{ij} = 1\}$ *and* $F^U_{ik} = \{r_j \in R_k | f_{ij} > 0, \tilde{u}_{ij} = 1\} \neq \emptyset$, $c_i(F^U_{ii}) \geq c_k(F^U_{ik} \setminus \{r_j\})$ *holds for every* $r_j \in F^U_{ik}$.

For responsive FEQX, we can draw similar conclusions as for responsive FEF1 in Examples 1 and 2. In summary, feasible and responsive FEQX assignments may not be EQX wrt private costs of drivers simply because the vehicle feasibilities or driver responsiveness may force such assignments to give all requests to one driver.

We give another algorithm for computing feasible and responsive FEQX assignments in every setting of our model: Algorithm 2. This algorithm simulates a greedy selection of some minimum (*min*) profit driver at each iteration and lets such a driver pick or respond for a most profitable (*max*), remaining, and feasible request. Thus, for two or more drivers, this algorithm and Algorithm 1 may return different assignments, because the former algorithm may select the same driver at two or more consecutive iterations whereas Algorithm 1 selects a driver at a given iteration that is different from the one selected at the previous iteration.

As the computation of the new algorithm evolves, a given driver may depart from the market (i.e. become "unresponsive") when their vehicle is removed from the fleet. This is another difference with Algorithm 1, suggesting that achieving responsive FEQX invites drivers to respond actively, supposing that they want to receive more requests and, thus, higher profit. From this perspective, we can view the new algorithm as a tool for achieving FEQX between any two drivers over the total numbers of their respective assignment iterations at which they are present at the market. Furthermore, in addition to feasibility and responsive FEQX, the returned assignment is also responsive complete.

Algorithm 2: SEMI-DECENTRALISED MIN-MAX

Input: V, R, P, F, deadline $T \in \Omega(|R|)$
Output: an assignment that is feasible, responsive complete, and responsive FEQX

$\forall v_i \in V : R_i \leftarrow \emptyset$
$\forall v_i \in V$: mark driver i as "responsive"
$U = (\tilde{u}_{ij})_{n \times m} \leftarrow (1)_{n \times m}$ ▷ responsiveness matrix
while $R \neq \emptyset$ **do**
  if all drivers are "unresponsive" **then** ▷ termination
    exit
  $v_i \leftarrow \arg\min_{v_h \in V} \tilde{c}_h(R_h)$ ▷ $\tilde{c}_h(R_h) = c_h(R_h)$
  if $\exists r_k \in R : \tilde{c}_{ik} = unk \vee \tilde{f}_{ik} = unk$ **then**
    send R to driver i ▷ decentralisation
    if within T, driver i reply with r_j, c_{ij}, f_{ij} **then**
      mark driver i as "responsive"
      $\tilde{c}_{ij} \leftarrow c_{ij}, \tilde{f}_{ij} \leftarrow f_{ij}$
      $R_i \leftarrow R_i \cup \{r_j\}, R \leftarrow R \setminus \{r_j\}$
      $\forall k \in \{h | h$ is "unresponsive"$\} : \tilde{u}_{kj} \leftarrow 0$
    else
      mark driver i as "unresponsive"
      $V \leftarrow V \setminus \{v_i\}$
  else
    if $\exists r_k \in R : f_{ik} > 0$ **then** ▷ centralisation
      mark driver i as "responsive"
      $r_j \leftarrow \arg\max_{r_k \in R : f_{ik} > 0} c_{ik}$
      $R_i \leftarrow R_i \cup \{r_j\}, R \leftarrow R \setminus \{r_j\}$
      $\forall k \in \{h | h$ is "unresponsive"$\} : \tilde{u}_{kj} \leftarrow 0$
    else
      mark driver i as "unresponsive"
      $V \leftarrow V \setminus \{v_i\}$
return $[(R_1, \ldots, R_n), U]$

We next compare Algorithms 1 and 2 in terms of their returned assignments. On the one hand, Algorithm 1 may return assignments that are not responsive FEQX. On the other hand, Algorithm 2 may return assignments that are not responsive FEF1. To observe these clearly, we give a simple setting in Example 3.

Example 3. *In a centralised setting, let there be* v_1, v_2 *and* r_1, r_2, r_3, r_4*. Define the costs as:* $c_{11} = 3, c_{12} = c_{13} = c_{14} = 2$ *and* $c_{21} = 6, c_{22} = c_{23} = c_{24} = 1$*. Pick unit feasibilities. Algorithm 1 gives priority firstly to driver* 1 *and secondly to driver* 2*. Hence, it may return* $(\{r_1, r_2\}, \{r_3, r_4\})$*. This assignment is not responsive FEQX because of* $c_2(\{r_3, r_4\}) = 2 < 3 = c_1(\{r_1\})$*. In contrast, Algorithm 2 may pick driver* 1 *at the first iteration and then it must pick driver* 2 *at any of the next iterations. Thus, it may return* $(\{r_1\}, \{r_2, r_3, r_4\})$*. This assignment is not responsive FEF1 because of* $c_1(\{r_1\}) = 3 < 4 = c_1(\{r_2, r_3\}) = c_1(\{r_2, r_4\}) = c_1(\{r_3, r_4\})$. □

Theorem 2. *In our semi-decentralised setting, Algorithm 2 returns in* $O(m \cdot \max\{n \cdot m, T\})$ *time an assignment that is feasible, responsive complete and responsive FEQX wrt the returned matrix, supposing that drivers are truthful and profit-maximising whenever they respond.*

Proof. For each iteration, after $O(n \cdot m)$ time, the centralisation/decentralisation phase takes $O(\max\{n, m\})$/$O(\{n, T\})$ time. For each driver k, we note that $\tilde{c}_k(R_k) = c_k(R_k)$ holds because if they are assigned some r_j then $\tilde{c}_{kj} = c_{kj}$ holds. Hence, the choice of the current least bundle profit driver i is well-defined. They may respond within T with some $r_j \in R$. In this case, driver i are truthful and profit-maximising. Hence, $r_j \in \{r_k \in R | c_{ik} = \max_{r_h \in R: f_{ih} > 0} c_{ih}\} \neq \emptyset$ holds. Thus, they may need to consider at most m candidate requests but, as $T \in \Omega(m)$, it follows that the overall time is $O(m \cdot \max\{n \cdot m, T\})$.

It is easy to see that the returned $(R_1, \ldots, R_n)$ is feasible and responsive complete wrt $U = (\tilde{u}_{ij})_{n \times m}$. We next therefore show that it is also responsive FEQX wrt $U = (\tilde{u}_{ij})_{n \times m}$. Wlog, we let $(r_1, \ldots, r_s)$ denote the picking order. We note that $s \leq m$ holds. We also let h denote the iteration number at termination. We note that $h \geq (s+1)$ holds. We next let $M(p) = (R_1(p), \ldots, R_n(p))$ denote the partial assignment at the start of iteration p. We prove by induction on $p \in \{1, \ldots, h\}$ that $M(p)$ is responsive FEQX wrt $U = (\tilde{u}_{ij})_{n \times m}$. Hence, $M(h) = (R_1, \ldots, R_n)$ as well.

In the base case, we let $p = 1$. $M(1) = (\emptyset, \ldots, \emptyset)$ is responsive FEQX wrt U by definition. In the hypothesis, we let $p < h$ and suppose that $M(p)$ is responsive FEQX wrt U. We note that $R \neq \emptyset$ hold and some drivers are not "unresponsive". In the step case, if no request is assigned at iteration p then $M(p+1) = (R_1(p), \ldots, R_n(p))$ is responsive FEQX wrt U by the hypothesis. For this purpose, we suppose that r_j is the request assigned to v_i at iteration p. Let us look at $M(p+1) = (R_1(p), \ldots, R_i(p+1), \ldots, R_n(p))$, where $R_i(p+1) = R_i(p) \cup \{r_j\}$.

For the drivers of $v_k, v_l \in V$ with $k \neq i, l \neq i$, $c_k(F^U_{kk}(p)) \geq c_l(F^U_{kl}(p)) - c_{lh}$ holds for each $r_h \in F^U_{kl}(p)$ by the hypothesis and the fact that the bundles of v_k, v_l do not change between $M(p)$ and $M(p+1)$.

We note that $V \neq \emptyset$ holds because not all drivers are marked as "unresponsive". Otherwise, the algorithm would have terminated. This would imply $p = h$ and lead to a contradiction with $p < h$. We note that $V \subset \{v_1 \ldots, v_n\}$ might hold because some drivers might have been "unresponsive" prior to iteration p and, for this reason, their vehicles might have been removed from $\{v_1 \ldots, v_n\}$ by the algorithm. Wlog, suppose that $V \subset \{v_1 \ldots, v_n\}$ holds. We consider two cases.

Case 1: Pick some $v_k \in \{v_1 \ldots, v_n\} \setminus V$. We note that $v_i \in V$ and, hence, $k \neq i$ holds.

Sub-case 1.1: For the driver of v_i, it follows by the induction hypothesis and the fact that driver i have strictly higher profit in $M(p+1)$ than in $M(p)$ that driver i are responsive FEQX wrt U of driver k. Indeed, $c_i(F^U_{ii}(p+1)) = c_{ij} + c_i(F^U_{ii}(p)) \geq c_k(F^U_{ik}(p)) - c_{kh}$ holds for each $r_h \in F^U_{ik}(p)$ because of $f_{ij} > 0$ and $\tilde{u}_{ij} = 1$.

Sub-case 1.2: For the driver of v_k, it follows by the induction hypothesis and the fact that driver k are "unresponsive" and, therefore, $\tilde{u}_{kj} = 0$ holds at iteration p that driver k are responsive FEQX wrt U of driver i. Indeed, $c_k(F^U_{kk}(p)) \geq c_i(F^U_{ki}(p)) - c_{ih} = c_i(F^U_{ki}(p+1)) - c_{ih}$ holds for each $r_h \in F^U_{ki}(p+1)$ because of $F^U_{ki}(p) = F^U_{ki}(p+1)$.

Case 2: Pick some $v_k \in V$ with $k \neq i$. We again consider two sub-cases.

Sub-case 2.1: For the driver of v_i, we have that $c_i(F_{ii}(p+1)) = c_{ij} + c_i(F^U_{ii}(p))$ holds by $f_{ij} > 0$ and $\tilde{u}_{ij} = 1$, as well as $c_i(F_{ii}(p)) \geq c_k(F_{ik}(p)) - c_{kh}$ holds for each $r_h \in F_{ik}(p)$ by the hypothesis. Hence, driver i are responsive FEQX wrt U of driver k.

Sub-case 2.2: For the driver of v_k, we observe that $c_k(R_k(p)) \geq c_i(R_i(p))$ holds by the fact that v_i is the least bundle profit vehicle within V. Therefore, $c_k(F^U_{kk}(p)) \geq c_i(F^U_{ii}(p))$ also holds because v_i and v_k are assigned only feasible requests when they are "responsive" (i.e. $F^U_{kk}(p) = R_k(p)$ and $F^U_{ii}(p) = R_i(p)$). As v_k might not be feasible for all requests in $F^U_{ii}(p)$, we derive $F^U_{ii}(p) \supseteq F^U_{ki}(p)$. By additivity, it follows that $c_i(F^U_{ii}(p)) \geq c_i(F^U_{ki}(p))$ holds. Consequently, $c_k(F^U_{kk}(p)) \geq c_i(F^U_{ki}(p))$ holds as well.

If $f_{kj} = 0$ or $\tilde{u}_{kj} = 0$, we derive $c_k(F^U_{kk}(p)) \geq c_i(F^U_{ki}(p+1))$ because of $F^U_{ki}(p+1) = F^U_{ki}(p)$. Otherwise, we derive $c_k(F^U_{kk}(p)) \geq c_i(F^U_{ki}(p+1)) - c_{ij}$ because of $F^U_{ki}(p+1) \setminus \{r_j\} = F^U_{ki}(p)$ and additivity. As driver i are profit-maximising, they pick feasible requests in some non-increasing cost sequence. It follows that $c_{ih} \geq c_{ij}$ holds for any $r_h \in F^U_{ki}(p+1)$. This implies $c_i(F^U_{ki}(p+1)) - c_{ij} \geq c_i(F^U_{ki}(p+1)) - c_{ih}$. We thus derive $c_k(F^U_{kk}(p)) \geq c_i(F^U_{ki}(p+1)) - c_{ih}$ for each $r_h \in F^U_{ki}(p+1)$. □

10 Responsive FEFX

In many real-world applications, the planner fix each request cost but semi-decentralise the feasibilities (e.g. DHL, GoAirlink). The cost of a given request thus does not depend on which feasible vehicle services the customer. For such applications, we define responsive FEFX.

Definition 12. $(R_1, \ldots, R_n)$ *is* responsive FEFX *wrt matrix* $U = (\tilde{u}_{ij})_{n \times m}$ *if, for each* $v_i, v_k \in V$ *where* $F^U_{ii} = \{r_j \in R_i | f_{ij} > 0, \tilde{u}_{ij} = 1\}$ *and* $F^U_{ik} = \{r_j \in R_k | f_{ij} > 0, \tilde{u}_{ij} = 1\} \neq \emptyset$, $c_i(F^U_{ii}) \geq c_i(F^U_{ik} \setminus \{r_j\})$ *holds for every* $r_j \in F^U_{ik}$.

With driver-independent costs, Algorithm 1 may not return a responsive FEFX assignment. By comparison, Algorithm 2 is guaranteed to return such an assignment in applications from this practical domain. We next prove these two results.

Lemma 1. *There are centralised settings with driver-independent costs, where Algorithm 1 may not return an assignment that is responsive FEFX wrt the returned matrix.*

Proof. In a centralised setting, let there be v_1, v_2 and r_1, r_2, r_3, r_4. For each v_i, define the costs as: $c_{i1} = 3, c_{i2} = c_{i3} = c_{i4} = 1$. Pick unit feasibilities. We run Algorithm 1 in this setting. The algorithm gives priority firstly to driver 1 and secondly to driver 2. Hence, the algorithm may return assignment $(\{r_1, r_2\}, \{r_3, r_4\})$. This assignment is not responsive FEFX because the envy of driver 2 cannot be eliminated by removing r_2 from the bundle of driver 1. To verify this, we observe that the inequality $c_2(\{r_3, r_4\}) = 2 < 3 = c_2(\{r_1\})$ holds. □

Lemma 2. *In our semi-decentralised setting with driver-independent costs, Algorithm 2 returns an assignment that is responsive FEFX wrt the returned matrix, if drivers are truthful and profit-maximising whenever they respond.*

Proof. With driver-independent costs, the notions of responsive FEFX and responsive FEQX coincide. This follows by the additivity of the profit preferences and the fact that $c_{ij} = c_j$ holds for each $v_i \in V$ and each $r_j \in R$. By Theorem 2, the result follows. □

11 Conclusions

We considered semi-decentralised fair division settings in the context of emerging vehicle routing problems, where not just the preferences of drivers play a crucial role, but also the feasibilities of their vehicles. We showed that popular notions such as EF1, EQX, and EFX, are not able to cope with the vehicle feasibilities or driver responsiveness in such settings. This motivated us to define respectively new notions: responsive FEF1, responsive FEQX, and responsive FEFX. They are applicable to semi-decentralised settings, including the traditional centralised settings.

We also proposed two new algorithms for such settings. For the theoretical domain of driver-dependent costs, our first algorithm returns an assignment that is feasible, responsive complete, and responsive FEF1, and our second algorithm returns an assignment that is feasible, responsive complete, and responsive FEQX. For the practical domain of driver-independent costs, responsive FEQX and responsive FEEX coincide and, therefore, returning responsive FEFX assignments in many real-world VRP applications is possible. Finally, we summarised our main results in Table 1.

References

Aleksandrov, M.; Barahona, P.; Kilby, P.; and Walsh, T. 2013. Heuristics and Policies for Online Pickup and Delivery Problems. In *Proceedings of AAAI 2013, July 14-18, 2013, Bellevue, WA, USA*.

Boehmer, N.; and Niedermeier, R. 2021. Broadening the Research Agenda for Computational Social Choice: Multiple Preference Profiles and Multiple Solutions. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '21, 15. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450383073.

Brams, S. J.; and Taylor, A. D. 1996. *Fair Division - from Cake-cutting to Dispute Resolution*. Cambridge University Press. ISBN 978-0-521-55644-6.

Budish, E. 2011. The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes. *Journal of Political Economy*, 119(6): 1061–1103.

Camerer, C. F. 2003. *Behavioral Game Theory Experiment in Strategic Interaction*, volume 32. Princeton University Press. ISBN 978-0-691-09039-9.

Caragiannis, I.; Kurokawa, D.; Moulin, H.; Procaccia, A. D.; Shah, N.; and Wang, J. 2019. The Unreasonable Fairness of Maximum Nash Welfare. *ACM Transactions on Economics and Computation*, 7(3).

Conitzer, V.; Freeman, R.; and Shah, N. 2017. Fair Public Decision Making. In Daskalakis, C.; Babaioff, M.; and Moulin, H., eds., *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, 629–646. ACM.

Dantzig, G. B.; and Ramser, J. H. 1959. The Truck Dispatching Problem. *Management Science*, 6: 80–91.

Dror, A.; Feldman, M.; and Segal-Halevi, E. 2021. On Fair Division under Heterogeneous Matroid Constraints. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 5312–5320. AAAI Press.

Freeman, R.; Sikdar, S.; Vaish, R.; and Xia, L. 2019. Equitable Allocations of Indivisible Goods. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 280–286. International Joint Conferences on Artificial Intelligence Organization.

Gerding, E. H.; Perez-Diaz, A.; Aziz, H.; Gaspers, S.; Marcu, A.; Mattei, N.; and Walsh, T. 2019. Fair Online Allocation of Perishable Goods and its Application to Electric Vehicle Charging. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 5569–5575. International Joint Conferences on Artificial Intelligence Organization.

Kash, I. A.; Procaccia, A. D.; and Shah, N. 2014. No Agent Left Behind: Dynamic Fair Division of Multiple Resources. *JAIR*, 51: 579–603.

Ma, H.; Fang, F.; and Parkes, D. C. 2019. Spatio-Temporal Pricing for Ridesharing Platforms. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, EC '19, 583. New York, NY, USA: Association for Computing Machinery.

Männel, D.; and Bortfeldt, A. 2018. Solving the pickup and delivery problem with three-dimensional loading constraints and reloading ban. *European Journal of Operational Research*, 264(1): 119–137.

Plaut, B.; and Roughgarden, T. 2018. Almost Envy-Freeness with General Valuations. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, 2584–2603.

Rheingans-Yoo, D.; Kominers, S. D.; Ma, H.; and Parkes, D. C. 2019. Ridesharing with Driver Location Preferences. In *Proceedings of IJCAI-19*, 557–564. International Joint Conferences on Artificial Intelligence Organization.

Savelsbergh, M. W. P.; and Sol, M. 1995. The General Pickup and Delivery Problem. *Transportation Science*, 29(1): 17–29.

Xu, Y.; and Xu, P. 2020. Trade the System Efficiency for the Income Equality of Drivers in Rideshare. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 4199–4205. International Joint Conferences on Artificial Intelligence Organization.