2017 IEEE 18th International Conference on Mobile Data Management

Fairness versus Optimality in Ridesharing

Ouri Wolfson
Department of Computer Science,
University of Illinois at Chicago &
Pirouette Software, Inc.
Chicago, IL 60607, USA
Tel: +1-312-996-6770
owolfson@gmail.com

Jane Lin
Department of Civil and Materials Engineering &
Institute for Environmental Science and Policy
University of Illinois at Chicago
Chicago, IL 60607, USA
Tel: +1-312-996-3068
janelin@uic.edu

Abstract—*As a spatio-temporal data-management problem, taxi ridesharing has received a lot of attention recently in the database literature. The broader scientific community, and the commercial world have also addressed the issue through services such as UberPool and Lyftline. The issues addressed have been efficient matching of passengers and taxis, fares, and savings from ridesharing. However, ridesharing fairness has not been addressed so far. Ridesharing fairness is a new problem that we formally define in this paper. We also propose a method of combining the benefits of fair and optimal ridesharing, and of efficiently executing fair and optimal ridesharing queries.*

Keywords—*Optimum, stable-roommates problem*

I. BACKGROUND

Taxi ridesharing[1] has received increasing attention recently both in the database literature ([6,7,8,9,10,12]) and in the broader scientific community [5]. These references discuss the questions of how to match passengers and taxis efficiently (i.e. quickly), effectively (i.e. maximizing the benefits), and what savings (e.g., in terms of mileage, number of trips, or dollar values) can be obtained by this matching in various practical situations.

Generally speaking the scientific literature is divided between papers that aim at optimizing some criteria such as travel time or distance (e.g. [5, 13]) and those that propose heuristics to find ridesharing partners quickly (e.g. [7]).

Increasingly popular commercial ridesharing systems such as UberPool and LyftLine are also studied extensively. However, these systems offer limited information and choice to passengers, in most cases merely a price comparison between riding alone and ridesharing. As the ridesharing market develops passengers will undoubtedly demand greater transparency and choice.

One problem with current commercial ridesharing systems is the lack of bounds on inconvenience due to ridesharing. Specifically, a passenger does not know in advance how long her trip will be delayed due to ridesharing, and cannot specify constraints such as tolerable delay. Furthermore, to save money she may be willing to walk or bike to a designated ridesharing pickup location, or from a drop-off location to the final destination; however, current ridesharing services do not allow her to specify this willingness, or bounds on the length of walking time. The scientific literature has proposed modeling a trip using such bounds (e.g. [5, 6, 10, 12]), however these have not been adopted by industry yet.

Two other problems with the taxi ridesharing are the lack of *transparency* and *fairness*. These problems are common to both commercial systems and the scientific literature, and we discuss them next. Consider first the lack of transparency. A passenger does not know the pool of available ridesharing candidates, and consequently, she does not have a say in her matching with a ridesharing partner[2]. For example, assume that passenger C can save \$3.5 when ridesharing with passenger B (compared to riding alone), and \$4 if ridesharing with passenger D. This will be the case if the routes of C and D are more similar than those of C and B. Then C will be better off ridesharing with passenger D than with B. Furthermore, as we illustrate in Example 2 of sec. II, a ridesharing service such as Uber may be motivated to pair C with B rather than with D. Intuitively, this is due to the fact that Uber optimizes globally rather than locally, i.e. for a specific passenger.

The solution to the lack of transparency is to show passengers (and their software agents) their potential ridesharing partners[3]. This transparency option of letting each passenger P see, and possibly automatically rank its potential ridesharing partners, e.g. in terms of dollar-savings[4], introduces the fairness problem. Intuitively, fairness can be summarized as follows. If P is not matched with its top choice partner it is because the top choice preferred a different ridesharing partner; and recursively, if P is not matched with its second choice partner Q it is because Q preferred a different ridesharing partner, etc. As we demonstrate in this paper, the fair

[1] We use the term "taxi ridesharing" because it is common in the literature, but the results of this paper apply more broadly, i.e. also to the Mobility-As-A-Service model of Uber and Lyft.

[2] In this paper we initially focus on the case where at most 2 trips can be merged. This limitation may be due to vehicle capacity, or individual traveler constraints, or both. This limitation will be relaxed in sec. 5.

[3] We believe that a transparent ridesharing service, that enables travelers to indicate their preferences in terms of partners and considers these preferences in forming the ridesharing plan, is more appealing than opaque services such as Uber and Lyft. The choice will not be done manually for each ride. Instead, when installing the ridesharing app (e.g. Uber) the user will specify criteria that enables automatic ranking of potential partners.

[4] Other criteria such as pollution-savings are possible. Furthermore, criteria can be combined; social, gender, safety preferences, and wait-time, can be valued for each partnership. A weight and a value given to each criterion will result in a utility number for each potential partner. This in turn will result in a total ranking of the potential partners.

2375-0324/17 \$31.00 © 2017 IEEE
DOI 10.1109/MDM.2017.25

ridesharing plan - a set of combined trips to be executed - may be different than the optimal one.

Example 1 (fairness): In order to demonstrate fairness in ridesharing, consider for example the ridesharing options represented by the graph in Fig. 1. The nodes represent single trips A, B, C, D; the edges represent shareability of trips, with each edge (X,Y) labeled by a pair of numbers: the number closer to X is the saving ($) of X from sharing the trip with Y, and the number closer to Y is the saving ($) of Y from sharing the trip with X. So the graph indicates that passenger A can rideshare with passenger B or D (but not both, due for example to vehicle capacity or constraints on the delay). If A rideshares with B, then A saves $0.5 compared to riding alone. If A rideshares with D, then A saves $3 compared to riding alone. Similarly, B can rideshare with A (saving $0.5), or rideshare with C saving $3.5. And similarly for passengers C and D.

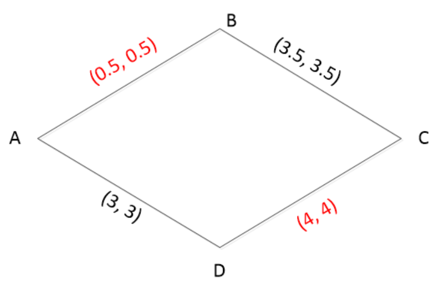

Figure 1: A Ridesharing-Graph

Now assume that for each passenger, the saving ($) is the single criterion that dictates its ranking of the potential ridesharing partners. C can rideshare with B or D, but C prefers to rideshare with D because its saving is higher. Similarly, D can rideshare with A or C, but D prefers to rideshare with C because its saving is higher. In other words, C and D prefer each other over any other partner. Thus it would be unfair to pair C with a partner other than D; and similarly, it would be unfair to pair D with a partner other than C.

Observe that fairness requires the pairing of C and D only in the case of reciprocity. This means that if C prefers D, but D prefers another partner, then it may be fair to pair C with another partner, e.g. B. But since in Fig. 1 both C and D prefer each other over other partners, it is unfair to pair either of them with other partners.

Furthermore, even though A pays more when paired with B, the scheme is still fair to him. This is because D, A's only other potential partner, prefers C over A. This is analogous to A making a partnership offer that is turned down. So the scheme is fair to A and B since they do not have a choice of ridesharing with other partners.

Practically, even though transparency is more conducive to fairness, the two concepts are independent. Even an opaque but fair ridesharing system could commit to a passenger C that if it is matched with C's *i*'th choice, then choices 1,…,*i*-1 preferred other partners. Moreover, even in an opaque system unfairness can be detected; for example, if C and D stand in a taxi line, by talking to each other they may discover that the system paired them unfairly. []

In this paper we introduce and formalize the notion of ridesharing fairness, and discuss algorithms to compute a fair ridesharing plan under various assumptions. We also introduce a payment scheme called *Guaranteed-Ridesharing-Fairness (GRF)* that enables a fair execution of an optimal plan. This means that the executed ridesharing plan is the optimal one, but the payments of the passengers are according to the fair plan, and often even lower. Furthermore, the GRF payment scheme is self-sustaining in the sense that it does not need to be externally subsidized.

The rest of the paper is organized as follows. In Sec. 2 we define the model and the concept of fairness. In Sec. 3 we introduce the GRF payment scheme and prove its properties. GRF requires the computation of both the optimal ridesharing plan, and the fair one, thus in Sec. 4 we discuss the efficient computation of these plans. In Sec. 5 we extend the discussion to ridesharing that involves more than two passengers, and in Sec. 6 we conclude.

II. THE MODEL

In this section we first introduce the Transparent & Fair Ridesharing (TFR) system that produces ridesharing plans. Each ridesharing plan (*rsp*) is a set of combined trips to be executed, where a combined trip consists of at most two single trips (recall that the first four sections of this paper focus on the case where at most two single trips share a ride). Then we define the concept of a Ridesharing Graph (RSG) which models the ridesharing options and benefits. Then we formally define the concepts of an optimal *rsp*, and a fair *rsp*. We also demonstrate that an optimal *rsp* is not necessarily fair, and vice versa.

The TFR system is cloud based. It receives from mobile clients information about individual trips, which form a ridesharing pool, and combination benefits. Specifically, a mobile client submits to the TFR system: 1) a single trip A, and 2) the benefit of combining A with each one of the other trips in the pool. A single trip is a triplet (origin-address, destination-address, and constraints). Constraints may include bounds on arrival time, pickup time, and wait time. Further details are found in [5, 10]. However, the model is applicable regardless of a specific precise format of a trip. Furthermore, the trip may be dynamic, i.e. currently being executed by a taxi (this corresponds to dynamic ridesharing [6,7]); or it may be static, i.e. just requested ([5, 10]) but not started. Indeed, a recent paper ([13]) uses pools that consist of both static and dynamic trips.

Assuming that a dynamic passenger does not transfer from one taxi to another before completing her trip, two dynamic trips cannot be combined. This will be indicated by the absence of an edge between two dynamic trips. A static and a dynamic trips can be combined by the static passenger joining the taxi used by the dynamic trip. In the case that two dynamic trips are combined, a separate problem is allocating a taxi to such a combined trip. However, we regard this allocation as orthogonal to the problem of constructing a ride-sharing plan. An existing scheme (see e.g. [7]) can be used for this purpose.

The benefit of combining two trips is represented by a weighted average of multiple individual criteria. The criteria of two passengers may be different. In this paper we equate the benefit to individual savings ($). For example in Fig. 1, the benefit of A in ridesharing with D is $3 of individual saving. The computation of the benefit of combining two trips can be done in the cloud (e.g. in the TFR system), even though conceptually the benefit is submitted by the client.

In TFR the problem of finding a ridesharing plan (e.g. the pairs of combined trips) is modeled as a weighted undirected *Ridesharing Graph* (RSG) (see e.g. Fig. 1). In an RSG(**V**,**E**), each node $v_i \in \mathbf{V}$ is a single *trip*, and each edge $e_{ij} \in \mathbf{E}$, if exists, connects trip v_i to a potential ridesharing partner v_j[5]. The weight of edge e_{ij}, denoted S_{ij}, represents the benefit obtained by combining the trips v_i and v_j , compared to the two separate single trips. S_{ij} must be positive for the ridesharing between v_i and v_j to take place. In other words, if $S_{ij} <0$ then v_i and v_j should not be combined and the edge e_{ij} does not exist, i.e., $e_{ij} \notin \mathbf{E}$. For example, in Fig. 1, edge e_{AB} exists and its weight S_{AB} equals 1 (sum of the individual savings). It means that A and B are willing to rideshare, and if the ridesharing plan combines them, then the total benefit (cost-saving) of the combined trip is $1. For simplicity we assume that the weights are distinct, i.e., any two edges have different weights.

In addition to the weight, each edge is also labeled by a pair of the individual savings ($S_{i(j)}$, $S_{j(i)}$), in which $S_{i(j)}$ is the saving (benefit) of v_i when ridesharing with v_j, and vice versa for $S_{j(i)}$. The sum of $S_{i(j)}$ and $S_{j(i)}$ equals S_{ij}, the weight of the edge. So if the individual-savings label of edge e_{AB} is (.25, .75) then the benefit (total saving) in combining trips A and B, S_{AB}, is $1, and the savings of A, $S_{A(B)}$, is $0.25 and that of B, $S_{B(A)}$, is $0.75. In Fig. 1. each edge is labeled by the individual savings pair of the two trips that it connects. If a trip $v_k \in \mathbf{V}$ is not connected by an edge to any other trips in **V**, then v_k rides alone and the individual saving of v_k, denoted S_k, is zero.

The savings may be evenly or unevenly split. An *Evenly-split* RSG is one in which the savings represented by the weight of each edge is evenly split between the two ridesharing partners. The RSG in Fig. 1 is evenly split.

However, if the distance or time saved is uneven, then even splitting of the savings is unreasonable. For example, assume that two passengers A and B are picked up at the airport, and the taxi drives straight to A's destination, drops her off, and then drives to B's destination. Then A traveled along her shortest path, whereas B hasn't. In this case, a better approach is to divide the "total saving" according to the increase in the distance (or time) traveled, compared to the shortest path. For instance, let's assume that A's destination is "n1" miles away along the shortest path, and in the joint path she travels "m1" miles. For B, assume that the corresponding figures are "n2" and "m2". Let x=m1/n1 and y=m2/n2. As a result, x/(x+y) of the saving is assigned to A, whereas the rest is assigned to B.

An *Unevenly-split* RSG is one which is not evenly split.

We postulate that any ridesharing algorithm that optimizes some criteria (e.g. $-savings, or distance, or pollution), rather than providing a heuristic, must implicitly or explicitly use a variant of the RSG. The reason is that the RSG simply encodes the ridesharing options and benefits, and each optimization algorithm must consider these. Indeed existing optimization papers do so (e.g. [5, 10, 13]).

In order to transparently construct the RSG, each passenger's software agent pairs her trip with all the potential partner trips; pairs of trips that are connected to each other become an undirected edge in the RSG. Directed edges, indicating that only one of the partners is willing to rideshare with the other, are dropped from the RSG.

Observe that the RSG models the ridesharing potential, i.e. which trips can be shared. However, not all the trips that can potentially be shared, will actually be shared in the set of executed combined trips. For example, in Fig. 1, passenger A cannot rideshare with both B and D because at most two trips can be combined; and even if they could, B and D cannot be combined. Thus, a ridesharing plan has to be computed. A *ridesharing plan* **R** is a subset of RSG(**V**,**E**) that contains only node-disjoint edges from the RSG. Node-disjointness means that a trip can be shared with at most one other trip. **R** may not contain all the nodes (trips) in **V**. A trip that is unpaired in **R** rides alone and incurs zero individual saving.

The *total saving* of a ridesharing plan (rsp) **R** is the sum of the weights of edges in **R**. Now we formally define the fair and the optimal *rsp*.

Definition 1: given an RSG(**V**,**E**), an *rsp* **O** is *optimum* if no other *rsp* **B** has a total saving greater than that of **O**.

Definition 2: given an RSG(**V**,**E**), an *rsp* **R** is *unfair* to a trip $v_i \in \mathbf{V}$ if: 1) there exists another trip $v_j \in \mathbf{V}$ and the edge $e_{ij} \in \mathbf{E}$ is not in **R**; and 2) v_i and v_j both incur a higher individual saving if ridesharing with each other than with their partners in **R**.

For example, consider the evenly split RSG in Fig. 2, and an **R** ={(A,D), (B,C)}. **R** is unfair to C (and to D) since C's saving in **R** is $3.5, and D's saving in **R** is $3. However, if C rideshares with D then each saves $4, i.e. more than that in **R**.

An *rsp* **F** is *fair* if it is not unfair to any trip. The concept of fairness is related to the Nash Equilibrium (NE).

Example 2 (difference between fair and optimal ridesharing plans): Consider the evenly split RSG in Fig. 2. This is simply the RSG of Fig. 1, with the weights, rather than the individual savings, of the edges displayed. An *optimum rsp* **O** is to

[5] See [5, 6, 10, 12] for efficient spatio-temporal data management algorithms that construct the RSG under various assumptions, e.g. whether or not passengers are willing to walk. The RSG can be constructed periodically, e.g. every 5 minutes. A large number of trip-requests can be pooled in short periods of time at hubs such as airports or train stations (see [10]).

combine trips B and C, and trips A and D. This leads to a total saving of $13. However, this plan would be unfair to C and D. Observe that both C and D prefer to be combined with each other rather than their partners in the optimum plan, since in this case the saving of each would be $4, compared to $3 and $3.5 respectively, in the optimal plan. And if C and D are combined, it leaves A and B to be combined. Thus, the *fair rsp* **F**, is the red one, having a total saving of $9.

From the societal good perspective, an optimal *rsp* is preferred because it maximizes the social welfare in aggregate. However, it may not be fair to some individuals. To remedy this gap, it means that if an optimal *rsp* **O** is executed, then a compensation scheme should be according to a fair *rsp* **F**. In other words, fairness requires that after a passenger pays the cost of her trip, she is compensated according to a fair *rsp*, regardless of the *rsp* actually executed. For the example in Fig. 1, this means that after ridesharing C is "compensated" at least $4 compared to riding alone. Likewise for D. This compensation scheme is discussed in the next section. []

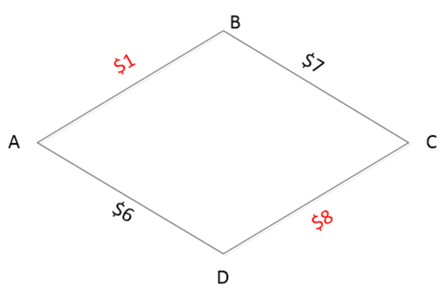

Figure 2: A Ridesharing Graph

III. GUARANTEED-RIDESHARING-FAIRNESS (GRF) PAYMENT SCHEME

In this section we introduce the GRF payment scheme. It executes the optimum *rsp*, while providing a fairness guarantee to each passenger. We show in this section that GRF is always feasible without external subsidies. GRF is an adaptation to ridesharing of a payment scheme for resources (e.g. parking slots), that was introduced in [2].

Given an RSG(**V**,**E**), let **O** be an optimum rsp, and let **F** be a fair rsp. GRF is a payment scheme that guarantees to each passenger/trip $v_i \in \mathbf{V}$ that its saving in **O**, denoted $S(i,\mathbf{O})$, will be no less than its saving in **F**, denoted $S(i,\mathbf{F})$. We denote the difference between the two savings by D_i, i.e.,

$$D_i = S(i,\mathbf{O}) - S(i,\mathbf{F}) \quad (1)$$

This is how GRF works:

i. If D_i is negative, then the TFR system pays passenger v_i an amount equal to $|D_i|$ to compensate the decrease in v_i's saving by moving from **F** to **O**.
ii. If D_i is positive - this means that vi benefits from some other passengers "sacrificing" what they could have saved in F - then v_i pays to the TFR system the amount of D_i; and its overall saving in **O** is still no less than that in **F**.

Thus, the GRF payment scheme guarantees that each passenger v_i incurs an adjusted saving, i.e. $S(i,\mathbf{O})$-D_i, which is not less than v_i's fair saving, $S(i,\mathbf{F})$, according to Eq.(1). Hence, GRF is Pareto-improving meaning no one is worse off in O with the GRF payment scheme, than in F.

<u>*Example 3 (the GRF payment scheme):*</u> For the example in Fig. 1, each passenger expects to be fairly compensated (i.e., the individual saving) according to the red color-coded *rsp* even when an optimum *rsp* (black color-coded) is executed. This can be done without any subsidy by implementing GRF as follows. In the optimum plan, the individual savings are: S(A,**O**)=3, S(B,**O**)=3.5, S(C,**O**)=3.5, and S(D,**O**)=3, and the total saving is $13. In the fair plan, the individual savings are: S(A,**F**)=0.5, S(B,**F**)=0.5, S(C,**F**)=4, S(D,**F**)=4, and the total saving is $9. So according to fairness, A and B are over compensated, and C and D are under compensated in the optimum plan **O**. Thus TFR with GRF collects $2.5 from A and $3 from B (i.e., a total of $5.5 collected by TFR), and compensates C $0.5 and D $1. After that, $4 is left over to distribute among the passengers and ridesharing platform in any mutually agreeable form. In particular, the leftover $4 can be distributed evenly among the 4 passengers to make each passenger better off than its compensation in a fair plan. [] [][]

Now we prove that the properties demonstrated in Table 1 hold in general. Assume that the TFR system selects the optimum *rsp* **O** and the GRF payment scheme. Then the *total income* (I), is the sum of the D_i's received from the passengers in (ii). The *total outcome* (U), is the sum of $|D_i|$'s paid out to passengers in (i). The GRF payment scheme is *revenue neutral* if and only if the total income is no less than the total outcome, i.e., $I \geq U$.

Theorem 1: For every RSG(**V**,**E**), the GRF payment scheme combined with the optimum *rsp* **O** is revenue neutral.

Proof is based on the fact that the total system saving of assignment **O** is no less than the total system saving of any other assignment, including **F**. That is, the following inequality always holds:

$$\sum_{\forall i \in \mathbf{V}} S(i,\mathbf{O}) \geq \sum_{\forall i \in \mathbf{V}} S(i,\mathbf{F}) \quad (2).$$

Then

$$\sum_{\forall i \in \mathbf{V}} [S(i,\mathbf{O}) - S(i,\mathbf{F})] \geq 0 \quad (3),$$

i.e.,

$$\sum_{\forall i \in \mathbf{V}} D_i \geq 0 \quad (4).$$

We rewrite (4) into the following:

$$\sum\nolimits_{+} D_i + \sum\nolimits_{-} D_i \geq 0 \quad (5),$$

where $\sum_{+} D_i$ represents the summation of all positive D_i's and $\sum_{-} D_i$ all negative D_i's. As denoted, $\sum_{+} D_i$ is the total income I of the TFR system from the passengers, and $\sum_{-} D_i$ is the total outcome U from the TFR system to the passengers. Thus Eq. (5) indicates that $I \geq U$. []

Now we will briefly discuss the practical implications of the GRF scheme. Generally, ridesharing represents a tradeoff between time and cost. If a passenger rideshares, she saves cost versus riding alone, but wastes time since her trajectory is not direct. Furthermore, a higher cost-savings indicates higher similarity in the individual trajectories of the two trips, and thus a lower amount of time wasted. Then the optimal plan implies that overall, a minimum amount of time wasted.

So, in these terms, what does the GRF scheme practically mean for each one of the passengers in Fig. 1? A and B waste less time in the optimal plan actually executed (than in the fair plan), but are reimbursed less than the executed plan warrants (0.5 instead of 3). On the other hand, C and D waste more time in the executed plan than in the fair plan, but they are compensated more than the executed plan warrants.

In case the weight of an edge represents pollution savings and the system is cap-and-trade, the GRF scheme's currency may be pollution credits. If the benefit of ridesharing consists of a combination of criteria, then the GRF scheme can be applied by mapping the benefit to some $-cost.

IV. DATA MANAGEMENT AND COMPUTATIONAL ISSUES

The *GRF* payment scheme requires that the Transparent and Fair Ridesharing (TFR) system construct the RSG of a given pool of candidate trips and compute two ridesharing plans, the fair and the optimum. In this section we address the problem of computing these *rsp*'s.

Consider first construction of the RSG. If we assume that a pool consists of all the taxi-trips originating from an airport during a 5 minutes interval, then the number of trips is less than 50 on average (see [10, 4]). The computation of the RSG requires multiple shortest path computations over the road network (which in NYC has about 500,000 edges). This computation, including optimizations such as Euclidean filtering, was addressed and can be done efficiently (see [10]).

Now consider the computation of the optimum *rsp*. Observe that any *rsp* is simply a matching, i.e. a subset **R** of the RSG edges, such that no two edges in **R** share a node (due to the fact that at most two trips can be shared). And an optimum *rsp* is a maximum-weight matching. Computing the maximum matching can be done in $O(n^{2.5})$, where n is the number of nodes of the RSG, i.e. trips in the ridesharing pool (see [3]).

Now consider the computation of the fair *rsp*. This computation differs depending on whether or not the Ridesharing Graph is evenly or unevenly split. These two cases are addressed in IV.A and IV.B, respectively.

A. Evenly-split Ridesharing Graphs

A fair *rsp* is computed iteratively by combining trips connected by the heaviest edge in the RSG, and removing them from the RSG. More specifically, the fair *rsp* **F** is computed by the following algorithm (NE4.1):

a. Let **F** consist of the empty set.

b. While there are edges in the remaining RSG do:

b.1) find the heaviest edge e_{XY} in the remaining RSG,

b.2) put the edge e_{XY} in **F** (i.e. combine trips X and Y), and remove X,Y and their adjacent edges from the remaining RSG.[]

Theorem 2: For an evenly split RSG, the *rsp* **F** computed by Algorithm NE4.1 is fair.

Proof: Assume by contradiction that there exist two trips A and B that are not paired with each other in **F**, but both A and B have a higher individual-saving if they rideshare with each other, rather than with their assigned partners in **F**. Then the edge e_{AB} must have been removed at step b.2 in some iteration of NE4.1. Furthermore, at that iteration another edge, say e_{AC}, must have been put in **F**. The fact that NE4.1 selected e_{AC} rather than say e_{AB} means that the weight of e_{AC} is higher than that of e_{AB}, and therefore, since the RSG is evenly split and the weights are distinct, the saving of A in ridesharing with C is higher than in ridesharing with B. This is a contradiction to the assumption that A's saving is higher when ridesharing with B.[]

NE4.1 can be made to run in O(n log n) (see [11]).

B. Unevenly-split Ridesharing Graphs

In the case of unevenly-split RSG's the NE4.1 algorithm does not work anymore. The reason is that NE4.1 uses the fact that at any iteration, if e_{XY} is the maximum-weight edge, then trip X is the preferred ridesharing partner for trip Y, and Y is the preferred partner for X. However, if the split is uneven, this not necessarily true. So in the example of Fig. 2 assume that trip D's individual-saving in ridesharing with C is 7 and in ridesharing with A is 2. So trip D prefers ridesharing partner C over partner A. However, if C's savings when riding with B is 3, then C prefers B over D.

Thus we introduce the NE4.2 algorithm, which computes the fair *rsp* of the RSG, if such an *rsp* exists, using a solution to the stable-roommates problem (SRP). In a given instance of the stable-roommates problem (SRP), each of *2n* participants ranks the others in strict order of preference. A matching is a set of *n* disjoint pairs of participants. A matching *M* in an instance of SRP is stable if there are no two participants *x* and *y*, each of whom prefers the other to their partner in *M* [16].

This is done as follows. Each trip X sorts its neighbors (potential partners in the ridesharing plan) in increasing order of X's cost in the partnership. The a solution to the SRP can be found in $O(n^2)$, where n is the number of trips(see [1]). And it is easy to see that this solution constitutes a fair *rsp*.

However, a solution to an SRP problem may not exist, and similarly a fair *rsp* may not exist for an unevenly split RSG.

<u>*Example 4 (nonexistence of a fair rsp)*</u>*:* Consider the RSG of Fig. 3 that uses the same notation as Fig. 1 in example 1. Consider the *rsp* **R**={(A,D), (B,C)}. This *rsp* is unfair to A, since both A and C save more when pairing with each other than with their *rsp* partners (D and B respectively). Also, if B or C pairs with D instead of A, it can be similarly shown that

the plan would be unfair to D's partner. Since the benefit in partnering with D is > 0, any *rsp* **R** will partner a passenger with D, and that **R** will be unfair to that passenger.

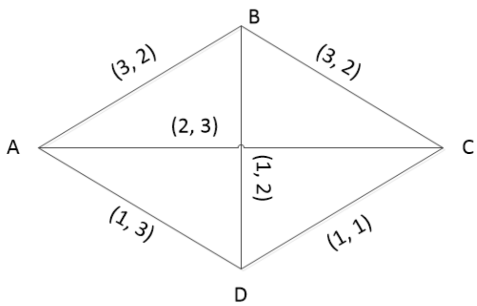

Figure 3: A Ridesharing Graph for which a fair rsp does not exist. []

If there is no fair *rsp*, there are several options. The first one is to declare that in this case, GRF is inapplicable, and each passenger is reimbursed according to the executed, optimum *rsp*. The second option is to invoke an algorithm that finds the minimum number of trips to be deleted from the RSG, such that the remaining set of trips have a solution to the SRP problem. Such an algorithm exists and has time complexity $O(n^2)$ ([14]).

V. COMBINING MORE THAN TWO TRIPS

In this case the problem of finding a ridesharing plan (i.e. the subsets of combined trips) is represented as a weighted undirected Ridesharing-Hypergraph (RSH). In a hypergraph each (hyper)edge is a set of nodes; the cardinality of each edge is c or lower, where c is the maximum capacity of a vehicle. The weight of each hyperedge is the total savings obtained if the set of trips is combined into a single one. The individual-savings in an RSH is defined analogously to the RSG. Observe that all the subsets of a hyperedge are also hyperedges in the RSH. The reason is that if trips A, B, C can be combined such that the constraints of each one of them is satisfied, then clearly they can be pairwise combined such that the same set of constraints is satisfied.

A *ridesharing plan* is a set of node-disjoint hyperedges in the RSH. An *optimum rsp* is an *rsp* of maximum weight, and finding it becomes NP-complete in hypergraphs; but a polynomial (2c+1)/3-approximation exists (see [5]).

An *rsp* **F** is *fair* if there is no hyperedge in RSH in which all the trips have a higher individual savings than in **F**. Define an *evenly-split* RSH to be one in which savings represented by the weight of each hyperedge is evenly split among its nodes (i.e. trips). Then an equivalent of the algorithm NE4.1, i.e. the selection of hyperedges in decreasing order of their i-s, works to find a fair rsp. Observe that here, in contrast to NE4.1, it is possible that at some iteration of step b.1 of NE4.1, a selected hyperedge H may have a lower weight than that of another hyperedge K in the remaining RSH, even though H's i-s is higher (due to the fact that H has a lower cardinality).

The problem of finding a fair rsp (if it exists) for an unevenly split RSH is NP-complete (can be shown by reduction from 3D-SR [15]), so approximations should be developed.

VI. CONCLUSION

In this paper we introduced the concept of ridesharing fairness, and contrasted it with the optimum currently used. We also introduced the Guaranteed Ridesharing Fairness (GRF) payment scheme, which executes an optimum ridesharing plan, but compensates the passengers fairly without external subsidies. GRF requires the computation of a Ridesharing Graph (RSG), and the fair and optimum ridesharing plans, so we also discussed the data management and complexity issues involved in these computations. These issues differ for ridesharing that involves at most two trips, and for ridesharing that involves more than 2 trips. They also differ depending on whether or not the savings from ridesharing are evenly split among the passengers.

Acknowledgement: This work was supported in part by the NSF under grants IIS-1213013 and IIP- 1534138.

VII. REFERENCES

[1] Gusfield, D., R. W. Irving. 1989. The Stable Marriage Problem: Structure and Algorithms, MIT Press.

[2] Wolfson, O., and Lin, J., 2014. A Marketplace for Spatio-temporal Resources and Truthfulness of its Users, Proc. of the 7th ACM SIGSPATIAL International Workshop on CTS.

[3] Micali, S.; Vazirani, V. V. (1980), "An (√VE) algorithm for finding maximum matching in general graphs", Proc. 21st IEEE Symp. Foundations of Computer Science,

[4] B. Donovan and D. B. Work, "New York City Taxi Trip Data (2010-2013)," 2014.

[5] P. Santi, G. Resta, M. Szell, S. Sobolevsky, S. H. Strogatz, and C. Ratti, "Quantifying the benefits of vehicle pooling with shareability networks," Proc. Natl. Acad. Sci., vol. 111, no. 37, 2014.

[6] Y Huang, F Bastani, R Jin, XS Wang, "Large scale real-time ridesharing with service guarantee on road networks", Proceedings of the VLDB Endowment 7 (14), 2017-2028.

[7] S. Ma, Y. Zheng, and O. Wolfson, "Real-Time City-Scale Taxi Ridesharing," IEEE Trans. Knowledge Data Eng., vol. 27, pp. 1782–1795, 2015.

[8] Tian, C., Huang, Y., Liu, Z., Bastani, F. and Jin, R., 2013. Noah: a dynamic ridesharing system, Proc. 2013 ACM SIGMOD Int. Conf.

[9] S. Ma and O. Wolfson, "Analysis and evaluation of the slugging form of ridesharing," Proc. 21st ACM SIGSPATIAL, pp. 64–73, 2013.

[10] Lin, J., Sasidharan, S., Ma, S. and Wolfson, O., 2016. "A Model for Multimodal Ridesharing and Its Analysis", Proc. of the 17th IEEE International Conference on Mobile Data Management, pp.164-173.

[11] S. N. Bespamyatnikh. An optimal algorithm for closest pair maintenance. Proc. of 11th Symposium on Comp. Geometry, 1995.

[12] M. Asghari, D. Deng, C. Shahabi, U. Demiryurek, Y. Li "Price-aware Real-time Ride-sharing at Scale - An Auction-based Approach", ACM SIGSPATIAL, 2016.

[13] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment", Proc. Natl. Acad. Sci., www.pnas.org/lookup/suppl/doi:10.1073/pnas.1611675114/-/DCSupplemental., 2016.

[14] J.J.M. Tan, Stable matchings and stable partitions, International Journal of Computer Mathematics 39 (1991) 11–20.

[15] K. Iwama, S. Miyazaki, and K. Okamoto. Stable roommates problem with triple rooms, 2007.

[16] Wikipedia. Stable Roommates Problem. https://en.wikipedia.org/wiki/Stable_roommates_problem.