Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Stable Matching for Dynamic Ride-Sharing Systems

Xing Wang, Niels Agatz, Alan Erera

To cite this article:
Xing Wang, Niels Agatz, Alan Erera (2018) Stable Matching for Dynamic Ride-Sharing Systems. Transportation Science 52(4):850-867. https://doi.org/10.1287/trsc.2017.0768

http://pubsonline.informs.org/journal/trsc/

TRANSPORTATION SCIENCE
Vol. 52, No. 4, July–August 2018, pp. 850–867
ISSN 0041-1655 (print), ISSN 1526-5447 (online)

Stable Matching for Dynamic Ride-Sharing Systems

Xing Wang,[a] **Niels Agatz,**[b] **Alan Erera**[c]

[a] General Electronic Global Research, Niskayuna, New York 12309; [b] Rotterdam School of Management, Erasmus University, 3062 PA Rotterdam, Netherlands; [c] School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30318

Contact: xwstella@gmail.com (XW); nagatz@rsm.nl, http://orcid.org/0000-0003-3514-201X (NG); alan.erera@isye.gatech.edu (AE)

Received: June 23, 2014
Revised: January 18, 2016; September 30, 2016; March 30, 2017
Accepted: April 5, 2017
Published Online in Articles in Advance: August 16, 2017

https://doi.org/10.1287/trsc.2017.0768

Abstract. Dynamic ride-sharing systems enable people to share rides and increase the efficiency of urban transportation by connecting riders and drivers on short notice. Automated systems that establish ride-share matches with minimal input from participants provide convenience and the most potential for system-wide performance improvement, such as reduction in total vehicle-miles traveled. Indeed, such systems may be designed to match riders and drivers to maximize system performance improvement. However, system-optimal matches may not provide the maximum benefit to each individual participant. In this paper, we consider a notion of stability for ride-share matches and present several mathematical programming methods to establish stable or nearly stable matches, where we note that ride-share matching optimization is performed over time with incomplete information. Our numerical experiments using travel demand data for the metropolitan Atlanta region show that we can significantly increase the stability of ride-share matching solutions at the cost of only a small degradation in system-wide performance.

Keywords: dynamic ride-sharing • stable matching • sustainable transportation

1. Introduction

Rising gasoline prices, traffic congestion, and environmental concerns have increased the appeal of services that allow drivers with spare seats to connect to people wanting to share a ride. The ubiquity of Internet-enabled cell phones provides new opportunities to enable *dynamic* ride-sharing, where rides are established on very short notice or even en route. Recently, many new companies, such as Lyft, UberX, Carticipate, Avego, Sidecar, and Flinc have emerged in both the United States and Europe that offer mobile phone applications that help match up drivers and riders.

While most cars can transport up to four passengers, the average occupancy rate in the United States is 1.4 persons (Sivak 2013). Similar occupancy rates are also found in Europe (European Environment Agency 2010). These empty seats represent a supply of unused transportation capacity. Effectively using empty seats by ride sharing represents an important opportunity to increase the efficiency of the urban transportation system, potentially reducing traffic congestion, fuel consumption, and emissions (Agatz et al. 2012, Furuhata et al. 2013). For the individual participants, there is also a clear financial incentive to share rides because it allows them to share trip-related expenses. The costs to own and operate a vehicle have risen sharply over the past decade with fuel prices in the United States rising 52% between 2009 and 2013 (American Automobile Association 2009, 2013).

In this paper, we consider a dynamic ride-share system setting in which potential participants place trip announcements as a rider or a driver at a time close to their desired trip departure times. A trip announcement specifies an origin and a destination location and additional information specify its potential timing. With this information, the ride-share system provider automatically establishes shared rides over time, matching potential drivers and riders. Some potential participants may remain unmatched.

In this research, we suppose that the ride-share system provider attempts to minimize a system-wide objective when determining shared rides. Specifically, we consider an objective to minimize *total system-wide vehicle-miles*, the total vehicle-miles driven by all potential participants traveling to their destinations, either in a ride-share or driving alone if unmatched. This objective aligns with societal objectives, such as reducing traffic congestion and emissions. Since this objective seeks to maximize the total travel distance savings of all participants, it also coincides with *minimizing total travel costs*, an important consideration for the participating drivers and riders.

Variable trip costs are largely proportional to distance (or time) traveled, which implies that cost reduction is only possible when the length of a ride-share trip is shorter than the sum of the lengths of the separate trips. If the cost of a shared trip is less than the sum of the costs of individual trips of its participants, it is always possible to allocate the cost savings among the participants such that each individual benefits. In this case, every match is *individually rational* with respect to travel cost because participants are always better off in a match than driving alone.

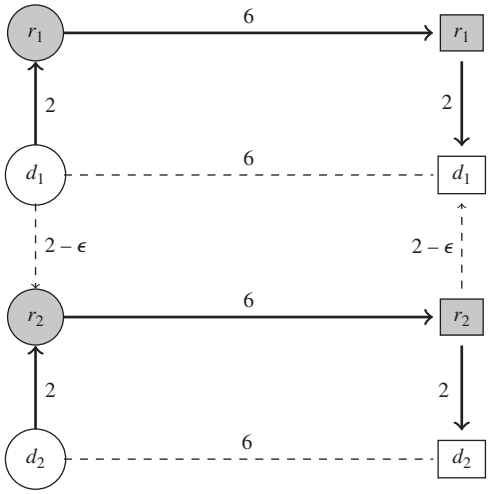
An automated process for establishing shared rides should be fast and require minimal effort from the participants. Centralized planning and coordination typically leads to better system-wide performance than uncoordinated decentralized approaches. In Agatz et al. (2011), we have shown the potential benefits of using optimization-based solution approaches to match riders and drivers compared with simple greedy matching rules.

However, drivers and riders do not necessarily have to accept a match as proposed by the ride-share provider. Even if a proposed match provides individual cost savings and satisfies a participant's time preferences, drivers and/or riders may reject a match if they believe they can establish a better match on their own. This is related to the notion of stability in cooperative game theory. In this terminology, a set of matches between riders and drivers is defined as *stable* if no rider and driver, currently matched to others or unmatched, would prefer to be matched together. If a driver and a rider form such a pair, they are called a *blocking pair*. If there are no blocking pairs in the matching solution, we call it a stable ride-share matching.

A ride-share matching solution aimed at minimizing total system-wide vehicle-miles (and corresponding external societal costs) or total number of matches may not necessarily maximize the cost savings of each individual participant. The degradation of system performance in terms of total vehicle-miles when participants reject proposed matches to form their own matches is an example of the price of anarchy (Koutsoupias and Papadimitriou 1999). The price of anarchy could potentially be large. Consider the simple example in Figure 1. Let us assume that the savings of each match are divided equally between the driver and rider that share a ride together. If $0 < \epsilon < 1$, the system-optimal solution is to assign rider 1 (r_1) to driver 1 (d_1) and rider 2 (r_2) to driver 2 (d_2). This would result in a system-wide vehicle-mile savings of four and individual savings of one for each participant. However, r_2 and d_1 would both prefer to be matched together instead of to their current partners. This would increase their individual savings by ϵ and reduce the system-wide savings from four to $2 + 2\epsilon$. In the worst case, when epsilon approaches zero, the price of anarchy could approach a 50% degradation in vehicle-mile savings from the system optimum. On the other hand, a system-optimal solution may be stable; for example, when $1 < \epsilon \leqslant 2$, the system-optimal assignment is to assign r_2 to d_1, and this assignment is also stable.

In this paper, we investigate how large the price of anarchy might be in a practical ride-sharing setting and develop optimization approaches that use stability considerations when generating ride-share matches. The main contributions of this paper can be summarized as follows:

Figure 1. Riders (Grey) and Drivers (White) Traveling from Origin (Circle) to Destination (Square)



- We are the first to our knowledge to introduce the concept of stability in the context of dynamic ride sharing and to provide mathematical programming approaches to solve stable and nearly stable ride-share matching problems.
- We experimentally quantify the impact of enforcing stability in a dynamic ride-sharing setting using simulations of representative work-based trips for a major U.S. metropolitan area.
- We evaluate the performance of a ride-sharing system that matches drivers to riders without taking stability considerations into account under various assumptions about participant information and behavioral response.

The remainder of the paper is structured as follows. In Section 2, we discuss literature that is relevant to our work. In Section 3, we describe the problem setting and assumptions. In Section 4, we introduce several optimization models to find stable and nearly stable matching solutions for the single-rider, single-driver dynamic ride-share problem. In Section 5, we discuss the dynamics of the system. In Section 6, we present numerical experiments based on work-based travel demand data for a large U.S. metropolitan area. In Section 7, we model participants' responses to unstable ride-share matches and study the system performance over time. Finally, in Section 8, we make some concluding remarks.

2. Related Literature

We review two streams of literature that are relevant to our work: literature on ride-sharing and literature on stability in two-sided matching markets.

Earlier work on ride sharing typically considers traditional carpooling that requires a long-term commitment among two or more people to travel together on

recurring trips for a particular purpose, often for traveling to work (see Baldacci, Maniezzo, and Mingozzi 2004, Wolfler Calvo et al. 2004). Dynamic ride-sharing systems that establish single matches on short notice have only recently started to receive some attention (see Furuhata et al. 2013 and Agatz et al. 2012 for a recent overview). In Agatz et al. (2011), we consider different optimization approaches to support matching of drivers and riders in real time. Amey (2011) and Ghoseiri, Haghani, and Hamedi (2011) propose similar approaches for a centralized dynamic ride-matching service to match up drivers and riders. More recently, Lee and Savelsbergh (2015) considered the use of dedicated drivers to serve riders that would otherwise remain unmatched in a dynamic ride-share system. Stiglic et al. (2015) consider the use of meeting points to allow a driver to be matched with multiple riders without increasing the number of stops the driver needs to make. Stiglic et al. (2016) study the impact of different types of participants' flexibility on the performance of a ride-sharing system.

There is a large stream of literature on methods to establish stable matches in two-sided matching markets (Roth and Sotomayor 1990). Typical applications include the centralized matching of college/public high school admissions and between medical students and residency programs. Empirical evidence suggests that the outcomes of centralized clearinghouses must be *stable* to ensure that they are accepted by the market (Roth 1984, 1991).

A fundamental problem in stable matching is the well-known stable marriage problem. This problem considers two finite sets of equal size: a set of men and a set of women. Each person has a strict preference ranking of all members of the opposite sex as a marriage partner. The objective is to create a matching of men and women such that there does not exist any pair of man and woman who prefer each other to their current partners. In their seminal paper, Gale and Shapley (1962) have shown that every instance of the stable marriage problem has a stable matching that can be found in $O(n^2)$ time. They introduce an algorithm that involves several rounds that can be expressed as a sequence of "proposals" from the men to the women. Since its introduction, many aspects of the stable marriage problem and related stable matching problems have been studied in depth by a large number of researchers; Iwama and Miyazaki (2008) provides a useful recent review of this literature.

In this paper, we consider a problem of matching riders to drivers for shared rides, and in our context, there are often riders who cannot be feasibly matched with certain drivers; thus, participants may have incomplete preferences. Furthermore, there are also cases in which a driver (rider) i may receive an identical benefit for a match with rider (driver) j or j' and thus does not strictly prefer one match to another. Manlove et al. (2002) and Iwama et al. (1999) show interestingly that, in such bipartite matching problems with both incomplete preference lists and also nonstrict preferences, finding a stable matching of maximum cardinality is NP-hard. This is true despite the fact that bipartite stable matching with incomplete lists and bipartite matching with nonstrict preferences are both problems in which polynomial approaches exist to find maximum cardinality matchings; in the former case, all stable matchings have identical cardinality and can be found by a Gale–Shapley algorithm extension, and in the latter case, all stable matchings are complete.

Since there may be alternative stable matchings in our ride-share matching application, it will be useful to consider matching problems that include a utility objective. Irving, Leather, and Gusfield (1987) considers the stable marriage problem in which the objective is to find a matching that is not only stable, but that maximizes a specific measure of total satisfaction for all participants, and provides a polynomial algorithm for its solution. Vande Vate (1989) initiated the study of the stable marriage problem using mathematical programming and showed that stable marriage solutions are extreme points of a polytope; thus, for any linear objective function, an optimal stable marriage solution can be identified via linear programming. Rothblum (1992) extended the polyhedral description to the case in which the bipartite graph is not complete. A simpler proof of the primary result in Vande Vate (1989) is provided by Roth, Rothblum, and Vande Vate (1993). Mathematical programming formulations for stable matching problem variants are used in our study.

The degradation of a system-optimal objective as a result of selfish behavior of system participants is known as the price of anarchy. Such a price is typically determined by comparing the objective function value of a system-optimal solution with the worst-case objective function value for a stable (or user) equilibrium (Koutsoupias and Papadimitriou 1999). A closely related concept is the price of stability, which compares a system-optimal objective function with the best-case stable equilibrium objective (Anshelevich et al. 2008). This concept is relevant to our study because we model a centralized ride-share provider that tries to achieve a *good* stable outcome.

The quality of a matching solution is typically defined as the sum of the utilities for all individuals in the system. The reason why there has not been a lot of attention for measuring the quality of a matching is because in many settings it is difficult to come up with an appropriate measure of utility. In the ride-share context, it seems likely that there are many potentially useful ways to measure the utility of individual matches, including those proportional to vehicle-mile savings (or other transportation cost measure savings).

We are only aware of one paper that studies the price of stability in two-sided matching markets. Based on experiments in which they randomly assigned utilities for each match, Anshelevich, Das, and Naamad (2009) demonstrate that the loss of social welfare from stability is often substantially lower than the theoretical worst case.

3. Problem Description and Assumptions

We consider a ride-share setting in which a ride-share provider for a particular metropolitan area receives a sequence S of trip announcements over time from potential participants. Each announced trip specifies whether the participant wants to act as a driver or a rider. This process generates two disjoint sets of trip announcements: a set of rider trips $R \subset S$ and a set of driver trips $D \subset S$. Each trip announcement specifies an origin and a destination location and additional information that reflects its potential timing. Suppose for simplicity that each origin and destination location is a member of a set P of locations and that the travel time t_{ij} and travel distance d_{ij} between each pair of locations $i, j \in P$ are known and constant. Let $v(s)$ and $w(s)$ represent, respectively, the origin and destination of trip announcement $s \in S$.

To minimize inconvenience for participants, we focus primarily on systems in which at most one pickup and delivery can take place during the trip and in which riders do not transfer from one driver's vehicle to another. For completeness, an extension of our basic system to the case in which drivers may be matched to multiple riders is provided in Section 4.4. We suppose that the ride-share service provider automatically generates match proposals between riders and drivers. An important consideration for a centralized system that creates matches between autonomous independent entities is whether the participants are satisfied with the matches. If they are not satisfied, they may not accept the proposed matches and eventually may stop using the system. Besides the interpersonal aspects, the timing of the ride and the financial benefits of sharing this ride together are important for the satisfaction of the participants with a particular ride-share match.

Drivers and riders provide information on their time schedule preferences in their announcements by specifying their earliest departure time and, indirectly, their latest arrival time using the notion of departure time flexibility. Specifically, we suppose that each announcement $s \in R \cup D$ provides an earliest time $e(s)$ at which the participant can depart from the origin $v(s)$ and a time flexibility $f(s)$ that specifies the difference between $e(s)$ and the latest time he would like to depart by if he were driving alone. For example, if a driver wished to arrive at his destination no later than $l(s)$, then we have time flexibility $f(s) = l(s) - e(s) - t_{v(s),w(s)}$.

The ride-sharing system provider automatically establishes ride-shares over time, matching drivers and riders who have announced trips. In this paper, we assume that a ride-share match is acceptable to its two participants only if it satisfies both of their time requirements; only acceptable matches are considered feasible. This implies that the participant for announcement s departs his origin no earlier than $e(s)$ and arrives at his destination no later than $l(s)$. A participant announces his trip at time $a(s)$ shortly before or at his earliest departure time. The announcement lead time $a^l(s) \geq 0$ denotes the difference between the participant's earliest departure time and his announcement time.

To assess the benefits of a particular match for an individual participant, we calculate the cost savings compared with traveling alone. For simplicity, we focus on a setting in which riders have a car available that they could use to drive to their destination alone if no ride-share can be identified. This means that for each rider–driver match we can calculate the vehicle-mile savings from ride sharing as compared with both driving alone as follows: $d_{v(d),w(d)} + d_{v(r),w(r)} - (d_{v(d),v(r)} + d_{v(r),w(r)} + d_{w(r),w(d)})$. Since variable trip costs, such as fuel expense, are typically proportional to the travel distance, the vehicle-mile savings also represent cost savings. We assume that the usual objective of the system provider is to maximize the total system vehicle-mile savings achieved by all driver–rider matches in a matching solution, which is the sum of the savings generated by each individual match.

If the cost of a ride-share trip is less than the sum of the costs of individual trips of its participants, it is always possible to allocate the cost savings among the participants such that each individual receives cost savings. This means that, as long as we have positive savings, a ride-share matching is *individually rational* since none of the participants prefers driving alone (i.e., remaining unmatched) to their match. Often, there may be time-feasible ride-share matches that do not generate positive vehicle-mile savings. These matches are not considered acceptable and are also treated as infeasible during the matching optimization.

The variable costs of each ride-share trip should be divided between the two participants in a way that is *budget-balanced* or *efficient*. This means that the total cost allocated to the participants is equal to the total cost incurred in the ride-share. As such, a driver can never receive a compensation that is greater than the cost of the complete trip to accommodate the rider. This is a relevant requirement in practice to distinguish ride sharing from commercial taxi services in legal terms (see Geron 2013 for a recent discussion on the legal issues). In this paper, we assume that the vehicle-mile savings (cost savings) of sharing a ride are equally divided between the two matched participants. It is

not difficult to see that such an allocation satisfies the properties of the *Shapley value* (1953), a well-known cost-allocation method in cooperative game theory.

We assume that the preferences of participants for different ride-share matches depend solely on the potential financial benefits, that is, the cost savings as compared with driving alone. The preference list of each participant consists of the set of feasible matches ranked based on the corresponding savings. For example, if a particular rider can be matched with drivers 1, 2, or 3 with respective savings of 3, 1, and 5, then the preference list for this rider is $(3, 1, 2)$. We denote the preferences with the following notation: $a \succ_c b$ denotes that person c prefers person a to b, and $a \succeq_c b$ denotes that either $a \succ_c b$ or person c is indifferent between a and b. Note that these preference lists are only defined over the set of feasible matches and are therefore incomplete since some matches are not feasible.

In the ride-share setting, it may often occur that an individual participant may have multiple ride-share match options that yield the same savings. This may occur, for example, when multiple riders announce around the same time and travel from the same origin to the same destination. We assume that participants are indifferent between matches if they generate the same financial benefits; thus, some preference lists may contain ties. In this context, we define a *stable* matching as a matching in which no driver–rider pair who are not matched together both strictly prefer each other to their corresponding partners in the matching. More formally, let $\mu(s)$ denote the matched ride-share partner of participant s, where $\mu(s) = s$ implies that participant s is unmatched. Define a *blocking pair* as a pair $(d, r) \in D \times R$ where $\mu(d) \neq r$, $r \succ_d \mu(d)$, and $d \succ_r \mu(r)$; note that d and/or r may be unmatched. This stability criterion is referred to as *weak* stability (see Irving 1994 for two other stability criteria in settings with preference indifference). We focus exclusively on *weak* stability in this paper and use the term *stability* to indicate weak stability.

4. Enforcing Stability in Ride-Sharing

In this section, we consider different approaches to establish *stable* ride-share matches. Dynamic ride-share matching optimization typically requires solving problems sequentially over time as new driver and rider requests are announced. To begin, however, we consider single-stage optimization problems in which the goal is to create a matching given a known set of announcements S. We introduce a simple heuristic and several mathematical programming formulations. Since we have incomplete, nonstrict preference lists in our setting, there may exist multiple stable solutions (Clark 2006 presents conditions for the uniqueness of stable matchings). While the heuristic simply finds a stable matching, the mathematical programming approaches aim to find a *best* stable solution given a system objective.

4.1. Greedy Matching Method

Given the objective of creating a matching solution that maximizes total system vehicle-mile savings, we now describe a greedy matching heuristic given a set S of announcements that consists of a set of rider trips $R \in S$ and a set of driver trips $D \in S$.

1. Determine for each rider announcement $r \in R$ the driver announcement $d \in D$ (if any) that represents the feasible match with the largest savings per participant in the match.
2. Among these matches, we select (r_m, d_m) with the largest savings and add this match to the matching.
3. Requests r_m and d_m are removed from S, and the process is repeated until no feasible matches remain.

Theorem 1. *The greedy algorithm generates a* stable *matching.*

Proof. Let s' be a greedy matching solution that is not stable. By definition, then there must exist a blocking pair (d, r) for which $r \succ_d \mu(d)$ and $d \succ_r \mu(r)$. Since we divide the savings equally between the two participants in a given ride-share match, this means that one half of the savings generated by match (d, r) is greater than the savings from match $(d, \mu(d))$ and also greater than the savings for match $(r, \mu(r))$. This is a contradiction because if this were true the match (d, r) would have been selected before the other two pairs were considered for matching. Q.E.D.

4.2. Basic Stable Formulation

As illustrated in Agatz et al. (2011), we can represent the ride-share problem using a maximum-weight bipartite matching model and then solve the problem using standard optimization software. Since we consider a setting in which the ride-share provider seeks to maximize the total distance savings produced for all participants, we can create a maximum-weight stable matching model by enhancing the earlier approach with blocking pair constraints similar to those in Roth, Rothblum, and Vande Vate (1993).

We create a node for each announcement in $R \cup D$ and an arc connecting a node $i \in R$ on one side of the bipartition with a node $j \in D$ on the other side if it is feasible to establish a ride-share match with driver j and rider i; recall that a match must be both time-feasible and produce positive travel distance savings. The weight c_{ij} assigned to feasible match arc (i, j) is simply the travel distance savings. The set A represents the set of feasible arcs in our bipartite graph. To complete the specification, let x_{ij} be a binary decision variable equal to 1 if ride-share match (i, j) is established and 0 if not. This gives the following maximum

weight bipartite matching optimization problem that maximizes system travel distance savings:

$$\begin{aligned}
\text{maximize} \quad & \sum_{i,j\in A} c_{ij}x_{ij} \\
\text{s.t.} \quad & \sum_{j\in D} x_{ij} \le 1, \quad \forall i \in R, && (1)\\
& \sum_{i\in R} x_{ij} \le 1, \quad \forall j \in D, && (2)\\
& \sum_{j'\succeq_i j} x_{i,j'} + \sum_{i'\succeq_j i} x_{i',j} + x_{ij} \ge 1, \quad \forall (i,j)\in A, && (3)\\
& x_{ij} \in \{0,1\}, \quad \forall (i,j)\in A. && (4)
\end{aligned}$$

Constraints (1) and (2) represent standard matching constraints. Each stability constraint (3) prevents blocking pair (i,j) by ensuring that either driver j is matched with rider i or driver j is matched with another rider i' who he prefers at least as well as rider i or rider i is matched with another driver j' that he prefers at least as well as driver j. This stability constraint ensures that the matching is weakly stable.

Note that if the preference lists of each participant contained no ties, constraint (3) becomes

$$\sum_{j'\succ_i j} x_{i,j'} + \sum_{i'\succ_j i} x_{i',j} + x_{ij} \ge 1, \quad \forall (i,j)\in A. \qquad (5)$$

Roth, Rothblum, and Vande Vate (1993) show that the linear relaxation of the system (1), (2), (4), and (5) has a special form.

Theorem 2 (Roth, Rothblum, and Vande Vate 1993). *Let C be the convex polytope corresponding to all feasible solutions to the linear relaxation of constraint system (1), (2), (4), and (5). Then the extreme points of C are all integer-valued and correspond to the set of stable matchings.*

Theorem 2 is quite useful for bipartite stable matching problems with strict preference lists since linear programming can be used to find an optimal matching for any linear objective function. For example, if $c_{ij} = 1$ for all $(i,j) \in A$, a maximum cardinality stable matching given incomplete preference lists can be identified. Since our ride-share matching problems will often contain ties in the preference lists, it will be necessary to use constraint (3) and integer programming methodology to identify maximum-weight stable matchings.

4.3. Nearly Stable Problem Formulations

To further explore the trade-off between system optimality and stability, we formulate two different additional models in which the stability constraints are relaxed. In the first model, we propose a relaxation that uses a modified definition of a blocking pair that accounts for the fact that participants may only perceive one match to be better than another if it creates significantly more savings. In the second model, we retain the original blocking pair definition but move stability considerations from the constraints to the objective function.

First, consider a model with a stronger definition of a blocking pair. Recall that a blocking pair (i,j) is originally defined as a rider–driver pair for which the savings generated by matching rider i with driver j exceeds both the savings of matching rider i with $\mu(i)$ and the savings of matching driver j with rider $\mu(j)$. In this research, blocking pairs are assumed to be undesirable since they create an incentive for i and j to match together rather than with the partners proposed in a system-optimal solution; there is an incentive for i and j to reject the system-proposed match. However, a matched participant may not be able to perceive small differences in cost savings or may decide that small additional savings provided by a blocking pair match may not justify rejecting a match proposed by the system. Therefore, for a given savings threshold ϵ, we can define a stronger blocking pair as rider–driver pair (i,j) for which the vehicle-mile savings S_{ij} exceeds the matched savings by at least ϵ. That is, $S_{ij} > S_{i,\mu(i)} + \epsilon$ and $S_{ij} > S_{\mu(j),j} + \epsilon$ if (i,j) is a blocking pair. Given this stronger blocking pair definition, we can create a relaxed form of the original stable matching model by replacing constraints (3) with

$$\sum_{j'\succeq^{\epsilon}_i j} x_{i,j'} + \sum_{i'\succeq^{\epsilon}_j i} x_{i',j} + x_{ij} \ge 1, \quad \forall (i,j)\in A, \qquad (6)$$

where the notation $i' \succeq^{\epsilon}_j i$ indicates that the savings S_{ij} from matching i with j does not exceed $S_{i'j}$ by more than ϵ.

Second, suppose that instead of introducing constraints to guarantee that a matching solution is stable that we instead build a formulation that attempts to maximize a measure of stability. To do so, we propose an approach with two phases. For the first stage, consider the basic stable ride-share matching optimization problem and remove constraints (3). Optimizing this system provides an upper bound on total vehicle-mile savings (objective function value O) and will typically be unstable (since it includes blocking pairs). In the second stage, we solve a matching problem that aims to minimize instability (measured here as the total number of blocking pairs) given a lower bound on the total system vehicle-mile savings, which we define as βO for some parameter $\beta \in [0,1]$. The second stage formulation is as follows:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i\in R,\, j\in D} y_{ij} \\
\text{s.t.} \quad & \sum_{j\in D} x_{ij} \le 1, \quad \forall i \in R, && (7)\\
& \sum_{i\in R} x_{ij} \le 1, \quad \forall j \in D, && (8)\\
& y_{ij} \ge 1 - \left(\sum_{j'\succeq_i j} x_{i,j'} + \sum_{i'\succeq_j i} x_{i',j} + x_{ij} \right), \quad \forall (i,j)\in A, && (9)
\end{aligned}$$

$$\sum_{i\in R,\, j\in D} c_{ij}x_{ij} \geq \beta O, \tag{10}$$

$$x_{ij}, y_{ij} \in \{0,1\}, \quad \forall (i,j) \in (R \times D), \beta \in [0,1]. \tag{11}$$

The objective function minimizes the total number of blocking pairs with variable y_{ij} taking the value 1 for a rider–driver pair (i, j) if it is a blocking pair. Constraints (7) and (8) are standard matching constraints. Constraints (9) ensure that y_{ij} takes the value 1 if (i, j) is a blocking pair. Constraint (10) enforces a lower bound on the total vehicle-mile savings generated by the matching solution x, where the bound limits the maximum degradation from the unstable objective function to $(1 - \beta)O$. Note that when $\beta = 1$, this formulation finds an alternative optimal solution with the fewest blocking pairs, and when $\beta = 0$, the problem reduces to the basic stable formulation.

4.4. Extension to Multiple Riders per Trip

If drivers and riders are relatively flexible in terms of their time schedules, it may be possible to simultaneously share rides with multiple riders with different origins and destinations on a single trip. To model this ride-share variant, we can extend our matching framework by introducing combined rider nodes. Such a combined node represents a set of riders that share a ride together in a single match. Here, we only create an arc between a driver and a combined node if the driver can pick up and drop off the riders while satisfying all time schedule restrictions. If multiple route sequences are feasible, the weight of an arc is equal to the savings generated by the shortest pickup and drop-off route, which is maximum among all feasible route sequences. In practice, we do not expect that drivers are willing to make more than four additional stops (pickups and drop-offs) in a single trip, which means that we have to evaluate at most six possible routes ($\{o_1, o_2, d_1, d_2\}$, $\{o_2, o_1, d_2, d_1\}$, etc.). This means we can simply find the shortest route by enumeration (Arslan et al. 2018).

In the extended formulation, we introduce the set of nodes $\mathcal{R}$ that contains single rider nodes ($R \subseteq \mathcal{R}$) and combined rider nodes. As such, a particular rider i may be associated with multiple nodes in $\mathcal{R}$. To ensure that each rider is matched at most once, we need to modify the matching constraints in the basic formulation. Let I refer to a node in set $\mathcal{R}$, and let $\mathcal{R}_i \subseteq \mathcal{R}$ be all nodes associated with rider i. We then replace constraints (1) and (2) in the basic stable matching formulation by the following constraints: $\sum_{j\in D} \sum_{I\in\mathcal{R}_i} x_{Ij} \leq 1 \; \forall i \in R$ and $\sum_{I\in\mathcal{R}} x_{Ij} \leq 1 \; \forall j \in D$, where the notation $i \in I$ refers to all riders i grouped together for a shared ride by node $I \in \mathcal{R}$.

To model the stability considerations, we can generalize the concept of a blocking pair to a blocking tuple. Similar to a blocking pair, we can define a blocking n-tuple as a set of $n - 1$ riders and one driver that all prefer being in a ride-share match together over their current match. In this setting, we replace constraints (3) in the basic stable matching formulation by the following (more general) stability constraints to ensure a weakly stable matching: $\sum_{i\subseteq I} \sum_{j'\geq_i j} x_{i,j'} + \sum_{I'\geq_j I} x_{I',j} + x_{Ij} \geq 1 \; \forall (I, j) \in A$. To prevent a blocking tuple (I, j), this constraint makes sure that we either establish ride-share match (I, j) or at least one of the riders or the driver is assigned to a match that he prefers at least as well.

If we assume that the cost savings of the match are divided equally among the driver and rider(s) in the match, then the greedy procedure described in Section 4.1 also provides a stable solution to the problem that allows matches between one driver and multiple riders.

5. Rolling Horizon Framework

Since new driver and rider trip announcements continuously arrive each day, it seems likely that any dynamic ride-share service provider must determine potential matches at many time points during the day. Each time the provider executes a procedure for planning matches, there are likely to be future ride request announcements that are not yet known.

A common mechanism for handling uncertainty of this type when planning is to use a deterministic rolling horizon solution approach, in which plans are made using all known information within a planning horizon but decisions are not finalized until necessitated by a deadline. After each execution of the algorithm, certain decisions are fixed, the planning horizon is rolled forward to include more known information, and the process is repeated. Our proposed approach uses a planning horizon that extends forward in time from the current time and captures all currently known announcements regardless of the announcement lead time. We assume a framework that reoptimizes at specific, regularly spaced time points.

In our solution framework, optimization run q at time $t(q)$ during an operational day considers all trip announcements s announced at times $a(s) \leq t(q)$ but excludes expired announcements (where $e(s) + f(s) < t(q)$) and those that have been matched within finalized ride-share arrangements. For run q, we set the earliest departure time $e(s)$ of each remaining announcement s to $\max(t(q), e(s))$. The optimization procedure then determines a best set of proposed ride-share matches as its output. Although matches may be found throughout the planning horizon, only a subset is finalized. We assume that the ride-share provider may notify participants about a ride-share arrangement as late as possible and that drivers (and riders) are indifferent to departure times as long as they arrive on time to their respective destinations. Thus, a ride-share arrangement is finalized only if the latest implied

departure time of the driver must occur before the next scheduled optimization run. For a ride-share match with driver d sharing a ride with rider r, the implied latest departure time is given by $\min(l(r) - t_{v(r),w(r)} - t_{v(d),v(r)}, l(d) - t_{w(r),w(d)}, t_{v(r),w(r)} - t_{v(d),v(r)})$.

5.1. Stability in a Dynamic Setting

In a *dynamic* ride-share setting, in which driver and rider trip announcements continuously arrive over time, we may consider stability in two different contexts. In the first context, we define *transient stability* as stability with respect only to information known at time $t(q)$. In the second context, we define the stronger notion of *a posteriori stability* as stability given the complete set of trip announcements that are received during an operating day. It is straightforward to enforce transient stability by simply solving a stable optimization model each time matches are optimized within the rolling horizon framework. Not surprisingly, however, doing so does not lead to a set of matches that are a posteriori stable.

Now suppose that matching solutions are generated to maximize total system vehicle-mile savings without including constraints to enforce (transient or a posteriori) stability. It is interesting to compare the a posteriori stability of matching solutions established via a rolling horizon framework versus an a posteriori optimization that considers all trip requests for a day. The dynamic arrival of trip announcements may either lead to more or less a posteriori stability in terms of the number of blocking pairs.

The rolling horizon framework may create fewer blocking pairs than found in the a posteriori solution if more stable matches are finalized before new requests arrive that lead to larger savings but less stability. Consider the example in Figure 2 in which the numbers below the arcs represent the savings for the match between that particular rider and driver. The timeline shows the announcement times of the different trip announcements. The a posteriori optimal solution would match rider 1 (r_1) to driver 1 (d_1) and rider 2 (r_2) to driver 2 (d_2) with a total savings of six. This would create a blocking pair between d_1 and r_2 because they would prefer being matched to each other than to their matching partners. Now consider what occurs in the dynamic problem in which d_1, r_2, and d_2 place

Figure 2. Prevented Blocking Pair $d_1 - r_2$

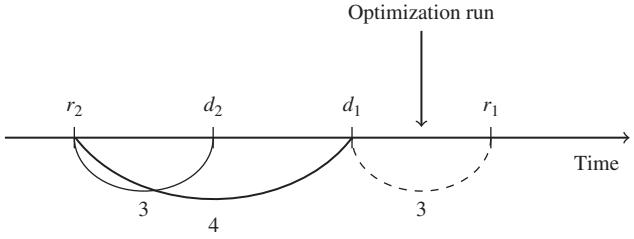
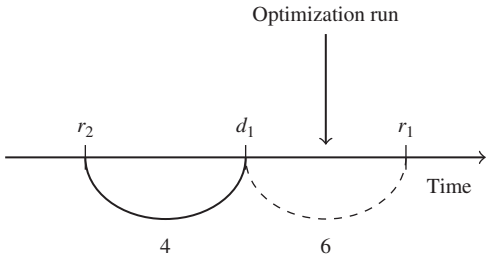


Figure 3. Created Blocking Pair $d_1 - r_1$



their trip announcements before r_1 and the rolling horizon framework initiates an optimization run at some time between the announcement time of d_2 and r_1. Without r_1 present, the optimal matching includes only the *stable* match between d_1 and r_2. If the implied latest commitment time of d_1 and r_2 is earlier than the announcement time of r_1, then the match $d_1 - r_2$ is finalized before r_1 and d_1 could be considered simultaneously in a subsequent optimization run.

It is also possible for the rolling horizon framework to generate a less stable matching than one established via an a posteriori optimization. Clearly, this would occur if matches that imply more blocking pairs are finalized before better alternative match opportunities are available to be considered. To illustrate this idea, consider the example in Figure 3. Since the match between rider 1 (r_1) and driver 1 (d_1) has the largest savings, d_1 prefers being matched to r_1, that is, $r_1 >_{d_1} r_2$. This matching also represents the solution that maximizes the savings. Now consider a situation in which rider 2 (r_2) and driver 1 (d_1) announce their trips before rider 1 (r_1). Without r_1 present as an option, we would match r_2 and d_1. If the next optimization run after the announcement time of request r_1 occurs after the implied latest departure time of match $r_2 - d_1$, then we miss the opportunity to match r_1 and d_1 and thus create a blocking pair in hindsight.

The previous example illustrates that, in a matching generated by the rolling horizon framework, there may exist blocking pairs for which at least one of the participants is currently not in a match. This cannot occur in an a posteriori optimal solution for the single-rider setting since, by definition, such a solution cannot be optimal given the objective of maximizing total system vehicle-mile savings. An *unmatched* participant i in a blocking pair with participant j implies that the savings from matching i and j is higher than the savings created by j's current match. Therefore, we would be able to strictly increase total system savings by matching i and j.

6. Computational Experiments

We now present the results of a set of computational experiments designed to generate insights about the role that stability and stable matchings might play

in dynamic ride-share matching systems. We implemented the stable and nearly stable matching solution approaches outlined in Section 4 and a simulation environment using the C++ programming language and CPLEX 11.1 as the linear and integer programming solver and performed all experiments on a quad-core 2.66 GHz Xeon E5430 Linux machine with 32 GB RAM. We now detail the study and its results.

6.1. Simulation Setup

In Agatz et al. (2011), we developed a ride-sharing simulation environment based on the 2009 travel demand model for the metropolitan Atlanta region, developed by the Atlanta Regional Commission (ARC). The ARC is the regional planning and intergovernmental coordination agency for the 10-county Atlanta area, a sprawling region with a population of approximately five million people occupying 6,500 square miles. The travel demand model for the region is used in this study to generate daily vehicle trips by purpose between all pairs of travel analysis zones within the region.

We generated five random streams of trips for use within our simulations as follows. Each travel analysis zone is considered to be a possible origin and destination for trips. For each origin–destination pair, we calculate an expected number of daily trip announcements by multiplying the average number of single-occupancy, home-based work vehicle trips with a fixed percentage of vehicle trips (i.e., the participation rate) that we assume might consider participating in a dynamic ride-sharing system. Then, for each pair and stream, we determine the number of actual trip announcements using a Poisson random variable with an expected value equal to the computed expected number of trips. Each trip announcement is equally likely to be a rider announcement or a driver announcement. The average trip distance in an announcement is approximately 11.6 miles. We assume a constant travel speed of 30 miles per hour.

Trip timing information is not available in the travel demand model data set. Therefore, we construct the time windows for each announcement as follows. For each trip from home to work, we draw the latest departure time from a normal distribution with mean 7:30 A.M. and a standard deviation of one hour to model a typical morning peak (McGuckin and Srinivasan 2003) and calculate the latest arrival time by adding the direct travel time to the latest departure time. Subsequently, we calculate the earliest departure time by subtracting a fixed system-wide time flexibility value of 20 minutes from the latest departure time. Such a fixed, system-wide time flexibility represents a system in which the ride-share provider sets a default time flexibility for all participants. In Section 6.4, we investigate the impact of this time flexibility and also study the effect of different flexibility levels for each of the participants. Furthermore, the announcement time is calculated by subtracting an announcement lead-time value from the earliest departure time.

In the experiments, unless stated otherwise, we assume that a driver can make at most one pickup and one drop-off during the trip. We believe that such one-to-one matches are likely to be accepted by the participants as they minimize inconvenience for both drivers and riders. However, since settings in which participants may agree to matches in which a single driver serves multiple riders (with different trip origins and/or destinations) are also interesting, we explore the impact of allowing two riders in a ride-share match in Section 6.8.

6.2. Stability of System-Optimal Solutions

In this section, we assess the stability of different matching solutions that do not explicitly constrain solutions to be stable. We focus initially on a posteriori matching optimization with which an optimal matching for a complete, known set of daily announcements S is determined in hindsight. The impact of dynamics on the results is discussed in Section 6.3. To assess the stability of a matching solution, we report the following statistics:

Participants in blocking pairs: the number of unique participants in at least one blocking pair divided by the number of matched participants, separated for riders and drivers

Blocking pairs per participant: the total number of blocking pairs divided by the number of unique participants in at least one blocking pair, separated for the riders and drivers

Unrealized savings: the potential savings of a blocking pair match minus the savings of the current match divided by the potential savings of a blocking pair match, averaged over all blocking pair participants

Table 1 presents the results for different participant rates, that is, 1% (11,563 announcements), 2% (23,155 announcements), and 4% (46,156 announcements), averaged over five simulation runs. The results show that the solutions that maximize the total system-wide vehicle-mile savings are relatively unstable. The

Table 1. Stability Measures of System-Optimal Maximum Weight Matching Solutions, $f = 20$ min, Averages Over Five Instances

Participation rate (%)	1	2	4
Participants in blocking pairs (%)			
Riders	19.4	21.3	23.9
Drivers	18.2	20.0	22.2
Blocking pairs per participant			
Riders	1.3	1.3	1.3
Drivers	1.3	1.4	1.4
Unrealized savings (%)	19.6	17.3	15.8

number of matched participants in at least one blocking pair ranges from 18.2% to 23.9%, which indicates that there are a significant number of participants who could potentially improve their savings by establishing a match with an alternative partner. The results also show that the instability, in terms of the number of blocking pairs, increases with the participation rate. An intuitive explanation for this is that the number of feasible matches per participant is likely to increase with the number of trip announcements in the system. As the number of feasible matches increases, there are also more opportunities for blocking pairs; note that a participant must have at least one feasible match option to be in a blocking pair.

From Table 1, we observe that for each of the different participation rates, the percentage of unique riders in blocking pairs is higher than the percentage of unique drivers. A potential explanation for this is that riders typically have more feasible match options than drivers. To provide more insights into the underlying structure of our ride-share instances, Figure 4 presents the distribution of the number of feasible matches per participant for both the drivers and riders. To understand this histogram, we must realize that to establish a feasible match, we need to satisfy two conditions: time feasibility and positive cost savings. Thus, a feasible match is only possible between trip announcements that have similar departure times and similar itineraries. Also, to create positive savings, the driver's original trip distance must be greater than the sum of the distance between driver and rider origins and the distance between driver and rider destinations. These factors clearly limit the number of feasible matches for both drivers and riders. On the other hand, the number of feasible match options appears to be sufficient to provide alternative matches for those who lost their proposed ride-share partner due to the stability constraints.

Figure 4. Number of Feasible Matches per Participant, 2% Participation Rate, $f = 20$ min, Averages Over Five Instances

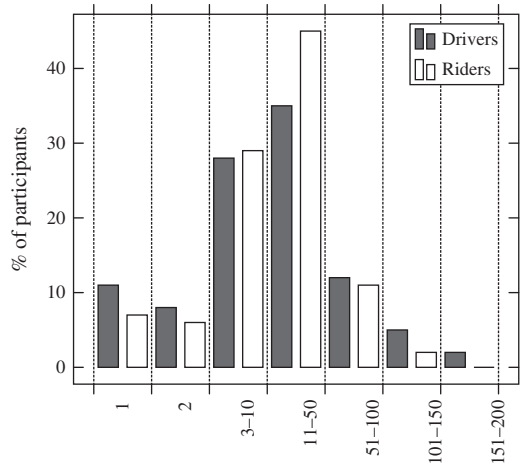
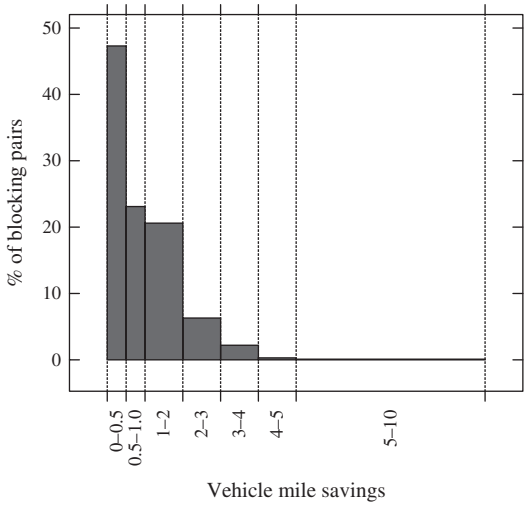


Figure 5. Unrealized Savings in Blocking Pairs, 2% Participation Rate, $f = 20$ min, Averages Over Five Instances



The potential, currently unrealized, savings of forming a blocking pair are relatively high. For the 2% participation rate case, we see a 17.3% potential increase in individual savings as compared with the savings offered by the system-proposed match. Interestingly, the unrealized potential savings per blocking pair seem to decrease with the participation rate.

To further investigate the potential savings, we plot the distribution of the potential extra vehicle-mile savings of blocking pairs for the 2% participant rate in Figure 5. The figure shows that approximately 30% of the participants have more than one mile of unrealized additional savings, which represents more than \$0.61 of savings per trip (American Automobile Association 2013). However, the potential savings are small for most participants in a blocking pair. Almost 50% of the participants have potential savings of less than 0.5 miles. This may be so small that participants may not perceive this value as significant. In Section 6.7, we study the impact of relaxing the value at which participants perceive blocking pairs.

6.3. Price of Stability

In this section, we compare the performance of the stable solution with the system-optimal solution. To evaluate the performance of a specific ride-share matching solution, we compute the following statistics for each solution:

Success rate: the number of matched trips divided by the total number of trip announcements.

Total system-wide vehicle-mile savings: the sum of the vehicle-mile savings of all ride-share matches divided by the total vehicle-miles that would be incurred if all announced participants would drive alone.

Average individual savings per trip: half of the vehicle-mile savings of a ride-share match divided by the

Table 2. Comparison of System-Optimal and Stable Matching Solutions, $f = 20$ min, Averages Over Five Instances

	System optimal			Best stable			Worst stable		
Participation rate (%)	1	2	4	1	2	4	1	2	4
Total vehicle-mile savings (%)[a]	21.9	25.8	29.2	20.9	24.6	27.9	20.9	24.6	27.9
Success rate (%)	65.5	72.4	78.0	62.0	68.6	73.9	61.9	68.6	73.8
Individual savings (%)	28.7	30.7	32.9	29.1	31.0	32.9	29.1	31.0	32.9
Detour distance (%)	19.3	19.0	18.7	16.3	15.6	14.8	16.3	15.6	14.8
$\delta = (Z - Z^s)/Z$				4.6	4.7	4.5	4.6	4.7	4.5

[a]Objective value.

distance the participant would have driven alone, averaged over all matched participants.

Average detour distance of drivers: the additional trip distance for the matched drivers as compared with their direct trip divided by the distance when driving alone, averaged over all drivers in a match.

To evaluate the impact of stability on the performance of the system, we compare the optimal objective function value for the unconstrained model (Z) with that for the *stable* matching model (Z^s). We define the price of stability δ as the relative gap as follows: $\delta = (Z - Z^s)/Z$.

In Table 2, we present the solutions when enforcing complete stability and compare these with the system-optimal solutions. To assess the difference between the best and the worst stable solutions, we also try to find the *worst* possible stable outcome. Therefore, we solve a *minimum-weight* stable matching problem by changing the objective of the model presented in Section 4.2 to minimize total vehicle-mile savings. The relative difference between the objective value of the *worst* stable solution and the objective value of the system-wide optimal solution represents the price of anarchy. While the maximum-weight stable matching problem with incomplete preference list and ties is theoretically difficult, we quickly find optimal stable solutions for all our practical instances. Table 3 shows that we find a solution for the largest maximum-weight stable matching problem in 241 seconds (more than 46,000 trips). A potential reason for this is that the number of ties in our instances is relatively small (see Table 4), and stable matching problems without ties can be solved in polynomial time.

The results show that enforcing stability by solving the stable ride-share problem comes at the cost of deteriorating system-wide solution quality. However, the price of stability is relatively small and represents an approximate reduction of 5% of the objective value. The fact that we only observe a small decrease in the total number of matches (i.e., the success rate) suggests that there are very few additional participants that remain unmatched due to stability requirements. This, together with the fact that all feasible matches have positive savings and thus contribute to the system vehicle-mile savings may explain the relative small price of stability. These results are in line with the findings of Anshelevich, Das, and Naamad (2009) that were based on several artificial matching instances.

We see that taking into account stability improves the individual cost savings of the matched participants. However, the average individual improvements are relatively small. The average relative detour of matched drivers decreases by about 18% when stability constraints are enforced.

Surprisingly, we observe that the difference between the maximum- and the minimum-weight stable matching optimal objective function values is very small. This means that the different stable solutions are very similar in terms of total savings for our specific instances. This implies that the simple greedy method presented in Section 4.1 may suffice to find good stable matching solutions. Note that this also indicates that any stable matching solution provides a very good lower bound on the optimal system-wide savings for these instances. This helps to quickly find optimal stable solutions.

6.4. Impact of Time Flexibility

We now examine the impact of time flexibility f on the price of stability by varying the flexibility level of

Table 3. CPU Times of System-Optimal and Stable Matching Models in Seconds

Participation rate (%)	1	2	4
Max-weight	2.6	16	118
Max-weight stable	2.8	18	241

Table 4. Ties in Preference Lists, $f = 20$ min, Averages Over Five Instances

Participation rate (%)	1	2	4
Maximum number of ties per participant	2	3	4
Avg. number of ties per participant	0.03	0.09	0.3
Avg. number of ties per participant with ties	1	1	1
Number of participants with ties (%)	3	9.5	23

Table 5. Comparison of System-Optimal and Stable Matching Solutions for Different Time Flexibilities, 2% Participation Rate, Averages Over Five Instances

	System optimal				Best stable			
Time flexibility (min.)	10	20	30	$U(10,30)$	10	20	30	$U(10,30)$
Total vehicle-mile savings (%)[a]	18.8	25.8	28.7	25.2	17.9	24.6	27.5	24.0
Success rate (%)	58.6	72.4	77.3	71.2	55.1	68.6	73.6	67.2
Individual savings (%)	29.9	32.3	33.4	32.1	29.9	32.5	33.6	32.3
Detour distance (%)	11.3	19.1	21.9	19.0	9.6	15.6	17.9	15.4
Participants in blocking pairs (%)	17.3	20.6	22.0	20.9	0	0	0	0
Blocking pairs per participant	1.3	1.4	1.4	1.4	0	0	0	0
Unrealized savings (%)	19.1	17.3	16.5	17.6	0	0	0	0
$\delta = (Z - Z^s)/Z$					4.8	4.6	4.2	4.8

[a]Objective value.

all participants between 10, 20, and 30 minutes. Furthermore, we study a setting in which each participant specifies his own time flexibility. In this experiment, we draw the individual flexibility of each participant from a uniform distribution between 10 and 30 minutes. On average, this gives the same flexibility as in the base case.

Table 5 provides the results for the system-optimal and stable matching for different time flexibilities. As expected, we see that the performance of the system increases with the time flexibility. Moreover, we also see that the rate of participants in a blocking pair increases with the time flexibility. The reason for this is that there may be more opportunities to form blocking pairs when the number of time-feasible matches increases with the time flexibility. Interestingly, the price of stability decreases with the time flexibility. The increasing number of feasible matches also enables *stable* matching solutions of higher quality. We see that the results for the $U(10,30)$ case are comparable to the 20-minute base case results.

6.5. Impact of the Objective

Instead of maximizing the system vehicle-mile savings, we may also consider other system objectives. The number of matched participants may be an important performance indicator for a practical, dynamic ride-share system because it indicates how likely it is that a participant may find a match. In Table 6, we report simulation results when we maximize the total number of matched participants in the system by solving a posteriori maximum cardinality matching problems. The different stability measures suggest that these matching solutions are less stable than those that maximize system vehicle-mile savings (Table 1). We see that more participants are in blocking pairs, and the unrealized potential savings are higher. Thus, the objective of maximizing the total number of matches appears to be less aligned with preserving stability. Such a result is not surprising since this objective values the number of matches over the savings (weight) of matches.

Similarly, we see that the costs of enforcing stability are significantly higher in this situation with a price of stability between 10% and 12%. As before, we see that the price of stability decreases with the instance sizes. The results also show that the difference between the maximum cardinality and maximum-weight stable matching solutions is very small. This is consistent with the earlier observation that different stable solutions do not differ much in terms of their objective values.

6.6. Impact of Dynamics

When new trip announcements continuously arrive over time, we use the rolling horizon strategy to enforce transient stability during each optimization run. We then assess the a posteriori stability of the complete matching solution. We use a reoptimization frequency of 10 minutes, commencing the first optimization run 10 minutes after the first announcement arrival each day. We compare the results from this dynamic setting with an a posteriori benchmark in which we solve the

Table 6. Stability Measures of Maximum Cardinality Matching Solutions, $f = 20$ min, Averages Over Five Instances

	System optimal			Best stable		
Participation rate (%)	1	2	4	1	2	4
Total vehicle-mile savings (%)	21.4	25.2	28.8	20.9	24.6	27.9
Success rate (%)[a]	70.1	77.2	82.3	62.0	68.6	74.0
Individual savings (%)	28.2	30.7	33.2	30.0	31.0	34.7
Detour distance (%)	22.2	22.2	21.5	16.3	15.6	14.8
Participants in blocking pairs (%)	29.4	31.7	32.2	0	0	0
Blocking pairs per participant	1.6	1.6	1.6	0	0	0
Unrealized savings (%)	36.5	34.5	30.4	0	0	0
$\delta = (Z - Z^s)/Z$				12	12	10

[a]Objective value.

Table 7. A Posteriori Stability Measures of Dynamic Matching Solutions, $f = 20$ min, 2% Participation Rate, Averages Over Five Instances

	Transient optimal	Transient *stable*
Total vehicle-mile savings (%)[a]	25.4	24.4
Success rate (%)	72.2	68.6
Individual savings (%)	32	32.3
Detour distance (%)	19.4	15.9
Participants in blocking pairs (%)	23.7	4.3
Unmatched participants in blocking pairs (%)	2.2	22.0
Blocking pairs per participant	1.4	1.2
Unrealized savings (%)	20.3	44.8
$\delta = (Z - Z^s)/Z$		4.0

[a]Objective value.

off-line problem that considers all trips for that day simultaneously and enforces a posteriori stability.

As discussed in Section 5, the rolling horizon framework can create blocking pairs that involve participants that are not matched in the final matching solution; recall that this is not possible in any optimal solution found a posteriori. In addition to the stability measures that were presented earlier, we therefore also determine the *rate of unmatched unique participants in a blocking pair*. This measure represents the number of *unmatched* unique participants in a blocking pair divided by the number of unique participants in at least one blocking pair, separated for the riders and drivers. If both people in a blocking pair are unmatched, Echenique, Wilson, and Yariv (2016) refer to this as an *available* blocking pair.

In Table 7, we see that dynamic matching solutions are less stable than the a posteriori benchmarks across all of the different stability measures. In particular, we not only see more participants in blocking pairs, but we also see that these blocking pairs represent higher unrealized savings. This suggests that the dynamics have a stronger destabilizing than stabilizing effect.

When enforcing transient stability during each optimization run, we see a significant reduction in the number of participants in blocking pairs, from 23.7% to 4.3% of the participants. The remaining a posteriori blocking pairs have relatively high potential savings. This is partially a result of the fact that 22% of the participants in these a posteriori blocking pairs are currently not in a match.

6.7. Impact of Relaxing Stability Constraints

Up to now, we enforced stability by not allowing any blocking pairs in the matching solution. In this experiment, we establish a *nearly stable* matching that maximizes the system-wide savings by modifying the stability constraints to only prevent perceptible blocking pairs with costs savings of at least ϵ as described in Section 4.3. Recall that threshold ϵ defines the minimum cost savings over the participant's current match for which a potential blocking pair match is perceived by the participant; smaller savings are assumed to be imperceptible. Since we assume the cost savings to be proportional to the distance savings, we can also define ϵ in terms of vehicle-mile savings. A perceived instability threshold of $\epsilon = 0.5$ miles, for instance, denotes that participants only perceive a blocking pair if the difference in vehicle-mile savings is greater than 0.5 miles.

Figure 6. The Impact of Relaxing Perceived Instability for Different Participation Rates, Averages Over Five Instances

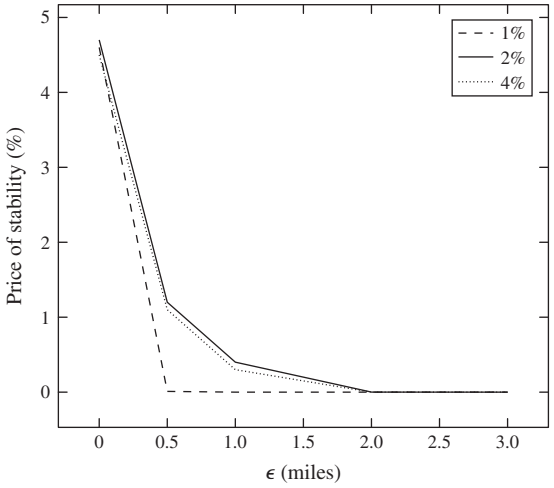
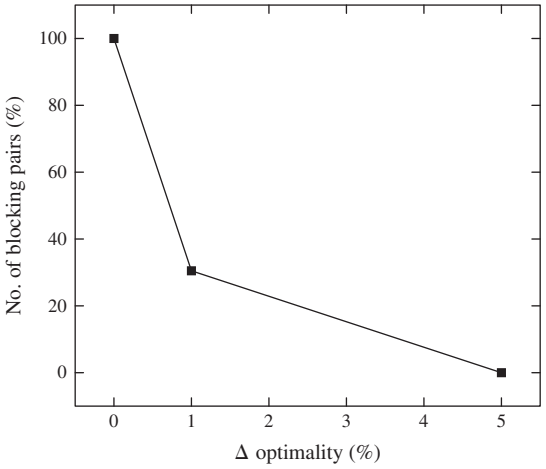


Figure 6 presents the price of stability for different perceived instability thresholds ϵ, ranging from full stability ($\epsilon = 0$) to a setting in which stability considerations do not play a role ($\epsilon = 3$). As expected, we see that the price of stability decreases with the perceived instability threshold. The decline is quite steep, and the results show that the price of stability is negligible if participants cannot perceive blocking pairs with a value of less than one mile of additional savings beyond the system match. Note that for the smaller (1%) instance the price of stability already goes down to zero at $\epsilon = 0.5$.

Next, we create nearly stable solutions by allowing a certain maximum degradation of the system's total savings while minimizing the number of blocking pairs. To do so, we apply the two-stage optimization approach as described in Section 4.3. Figure 7 presents the results when allowing a 1% or 5% deviation from the unstable maximum vehicle-mile savings objective function value. We can see that by allowing a 1% reduction in system savings from the unconstrained optimal value, the number of participants in blocking pairs decreases by more than 60%, and the total number of blocking pairs decreases by about 70%. The relative number of participants in blocking pairs also decreases

Figure 7. The Impact of Relaxing System Optimality, 2% Participation Rate, Averages Over Five Instances



from 20% to 8% of the total number of matched participants. The results suggest that allowing a small degradation in the system's total savings allows the creation of significantly more stable solutions.

6.8. Impact of Allowing Matches with Multiple Riders

In this section, we study the impact of allowing matches that involve multiple riders by using the model extension described in Section 4.4. In particular, we consider a setting in which a driver can serve at most two riders along his route.

When the number of riders and drivers is large, it is time-consuming to find all feasible matches with one or two riders. Moreover, since the number of required stability constraints is equal to the number of feasible matches, it becomes computationally prohibitive to solve very large stable matching problems using our approach. Table 8 presents the solution times for different instance sizes. Note that these are *theoretical* hindsight benchmarks in which we determine an optimal stable matching for a complete, known set of daily announcements. Within a dynamic setting, the problems that we would have to solve in each run of the rolling horizon approach would likely be much smaller.

Table 8. System-Optimal and Stable Matching Models with at most Two Riders, $f = 20$ min, Averages Over Five Instances

Participation rate (%)	0.3	0.5	0.8	1
Number of riders	1,711	2,909	4,634	5,758
Number of drivers	1,741	2,914	4,668	5,805
Number of feasible matches (thousands)	49	233	868	1,696
CPU times max-weight (sec.)	18	187	997	2,238
CPU times max-weight stable (sec.)	27	203	1,814	6,034

Figure 8. Bipartite Graph with Two Drivers (d_1 and d_2) and Two Riders (r_1 and r_2)

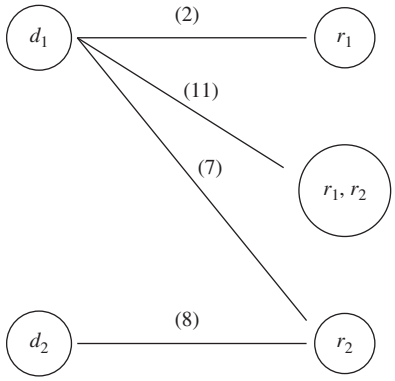


Table 9 presents the system-optimal and stable solutions for two-rider matching problems for a participation rate of 1%. For convenience, we also display the results for the single-rider setting. As before, we provide the solution of the worst possible stable solution in terms of vehicle-mile savings. The results show that we can improve the performance of the ride-share system by allowing multiple riders per match. In particular, we see an increase in the total vehicle miles savings of 6.8 percentage points as compared with the single-rider case. Moreover, we are able to match more riders with fewer drivers, that is, more than 62% of the matches in the optimal solution involve two riders.

We also observe that the two-rider system-optimal solutions are less stable than the single-rider solutions. No less than 52.5% of the participants are in at least one blocking pair. To understand this significant increase in the number of participants in blocking pairs, note that there are many more feasible matches and thus many more options to create a blocking pair in the two-rider solutions. The relative unrealized savings of these blocking pairs are lower, on average, than the unrealized savings in the single-rider solutions. The relative instability of two-rider system-optimal solutions suggests that it is more important to find stable solutions in ride-sharing matching problems when drivers can serve multiple riders since participants may be less satisfied with proposed matches when stability constraints are ignored.

While not possible in the a posteriori single-rider setting, we see that there are *unmatched* participants (mostly drivers) in blocking pairs in the two-rider solutions. The reason for this is that dividing the cost savings equally among all participants in a two-rider match is relatively unattractive for the rider who is responsible for most of the savings in the match. Those riders may be better off with an alternative driver that is currently not matched. Figure 8 illustrates this for

Table 9. Stability Measures of Maximum-Weight Matching Solutions with at most Two Riders per Match, $f = 20$ min, 1% Participation Rate, Averages Over Five Instances

	System optimal		Best stable		Worst stable	
Max. number of riders per match	1	2	1	2	1	2
Total vehicle-mile savings (%)[a]	21.9	28.7	20.9	26.5	20.9	26.5
Success rate (%)						
Riders	65.5	81.0	62.0	75.6	61.9	75.6
Drivers	65.5	49.8	62.0	48.3	61.9	48.3
Two rider matches (% of matches)	0	62.7	0	56.7	0	56.7
Individual savings (%)	28.7	39.1	29.1	37.9	29.1	37.9
Detour distance (%)	19.3	19.3	16.3	19.2	16.3	19.2
Unmatched participants in blocking pairs (%)						
Riders	0	0.3	0	0	0	0
Drivers	0	5.0	0	0	0	0
Participants in blocking pairs (%)	18.8	52.5	0	0	0	0
Blocking pairs per participant						
Riders	1.3	2.5	0	0	0	0
Drivers	1.3	4.2	0	0	0	0
Unrealized savings (%)	19.6	19.3	0	0	0	0
$\delta = (Z - Z^s)/Z$ (%)			4.6	7.7	4.6	7.7

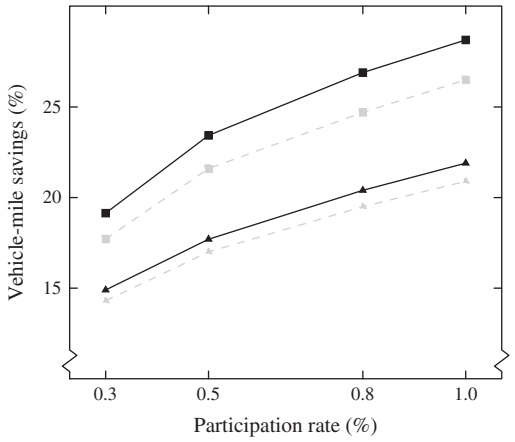
[a]Objective value.

an example with two drivers and two riders. The numbers above the edges denote the cost savings associated with the match. In this example, the optimal solution is to match driver d_1 with rider r_1 and rider r_2 for a total distance savings of 11. The match between driver d_2 and rider r_2 forms a blocking pair because the individual savings for rider r_2 are larger with driver d_2 ($\frac{8}{2}$) than in the optimal matching ($\frac{11}{3}$).

As in the single-rider case, the results suggest that the difference between the minimum- and maximum-weight matchings is very small. This means that for the two-rider setting we are able to find very good stable solutions by using the greedy heuristic as presented in Section 4.1. Note, however, that since the number of matches with multiple riders grows exponentially with the number of riders and drivers, the greedy algorithm is no longer solvable in polynomial time.

Finally, we see that the price of stability is higher for the two-rider case than for the single-rider case. Moreover, Figure 9 suggests that this is true for different participation rates. One reason for this is that the number of two-rider matches increases with the density and that, while such matches are beneficial in terms of system-wide savings, they are also associated with less stable solutions.

Figure 9. The Vehicle-Mile Savings for Optimal (Black Solid) and Stable (Grey Dashed) Solutions for Different Densities for the Single-Rider Case (Triangle) and Two-Rider Case (Square), Averages Over Five Instances



7. Participant Behavior in Response to Unstable Matchings

Empirical evidence in various real-life market settings (most notably in the medical labor market) suggest that stable matching procedures allow for more successful markets than unstable procedures (Roth 1990, 2008). Unstable procedures give people incentives to form matches outside of the marketplace, which leads to various types of inefficiencies. To investigate the potential impact of instability on the performance of a ride-share matching system, we run a simulation of the system over time based on participants' behavior over multiple trips. To develop a reasonable model for the participants' responses to proposed matches that are unstable, we draw on the economics literature on two-sided matching markets (Roth and Sotomayor 1990). A fundamental assumption is that a participant makes decisions that maximize his expected payoff. This means that if participants discover that a ride-share matching

system has matched them in such a way that they are in a blocking pair, they may not accept the proposed match and instead may try to establish a match by themselves outside of the system. Furthermore, they may be dissatisfied with the service provided by the system and may decide to stop announcing new trips in the future.

To evaluate the performance of a system over an extended period of time, we generate trip announcements for multiple consecutive days as follows. First, we generate a set of initial participants as described in Section 6.1 (we use a participation rate of 1%). Each participant i has a given home–work trip and an expected latest departure time x_i that is drawn from a normal distribution with mean 7:30 A.M. and a standard deviation of one hour. We now assume that this set of potential participants places trip announcements each day. However, the exact timing of their trip may vary from day to day. In particular, we draw their actual daily latest departure time from a normal distribution with a mean x_i and a standard deviation of 10 minutes. For each day, we determine a system-optimal matching solution by solving an a posteriori matching problem using a complete day's worth of announcements at a time.

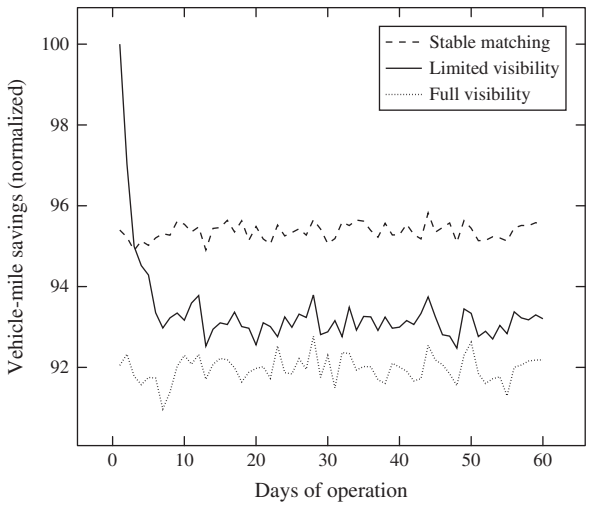
We distinguish between two information scenarios: full visibility or limited visibility. In the scenario with full visibility, the participants are assumed to be aware of all trip announcements in the system at any point in time. This allows them to easily perceive whether or not they are in a blocking pair. In most real systems, it will be nearly impossible for participants to know the complete announcement information for all other participants. Thus, we also consider a limited visibility scenario in which participants only know the ride-share announcement information for a relatively small set of other participants and thus only perceive blocking pairs that may exist with other participants that they know. In this limited-visibility case, we do assume that participants get to know other participants over time by sharing rides with them. Since a dynamic ride-share system is characterized by varying participant time schedules, participants are likely to share rides with different people on different days. For each of the participants, we keep track of which other participants they get to know over time. If participants d and r ever share a ride together, we assume that from that point forward they are aware of each other's detailed announcement information.

We now study how the performance of a dynamic ride-share system may degrade over time if participants only accept system matches they perceive to be *stable* and otherwise establish an alternative match outside of the optimal matching.

First, we detail the approach we assume participants use to find alternative matches. Consider a set of participants S matched in a system-optimal matching. Let participant $i \in S$ be the remaining one with the smallest, earliest departure time. We assume that participant i then compares his current system match to each possible feasible other match with another known participant in order of highest to lowest cost savings. If he finds a match j with higher cost savings that also forms a blocking pair, we assume that i and j mutually decide to reject their system matches and ride together. Once matched in this way, i and j are assumed to now be unavailable for future proposals. This procedure is similar to the Gale–Shapley algorithm (1953) for the stable marriage problem except that there are no tentative engagements; therefore, this procedure is not guaranteed to yield a stable matching in the end. We use this simpler procedure to model participants who may be willing to investigate some alternative match options but who must finalize matches quickly and begin traveling. Such an approach is consistent with empirically observed behavior in decentralized, two-sided matching markets (Echenique, Wilson, and Yariv 2016).

Figure 10 depicts simulated results for the long-run system performance, assuming such a system match rejection scheme. We compare three different scenarios. In the benchmark scenario, we generate a *stable* system matching; since no blocking pairs exist because of a posteriori stability, no participants reject their system matches. The benchmark is compared with the full- and limited-visibility scenarios that establish a system matching each day that maximizes total system vehicle-mile savings while ignoring stability; in the former, participants who reject their system match may establish another match with any other (feasible) participant while in the latter they are limited only to participants that they know at that point in time. The figure depicts the total vehicle-mile savings produced

Figure 10. System Performance Degradation When Participants Reject Unstable Matches



by the ride-share matches established (both system and extra-system matches), measured as a fraction of the optimal unstable vehicle-mile savings.

We see that the performance of the ride-share system under the limited-information scenario deteriorates rapidly over time as more people get to know each other and start breaking up system matches to create extra-system matches out of blocking pairs. After a couple of days of operation, the system that relies on a stable matching already outperforms the system that proposes unstable matchings under the limited information scenario. After less than two weeks, the system appears to reach a steady state in which the system-wide vehicle-mile savings are approximately 93% of the hypothetical unstable maximum. We see that this is close to the degradation that results in the scenario in which all participants have full visibility of all other participants' announcements, which appears to reach a steady state of 92% of the hypothetical unstable maximum.

8. Concluding Remarks

In an automated, dynamic ride-share setting, drivers and riders do not necessarily have to accept match proposals by the ride-share provider. Therefore, it is important to consider the stability of the proposed ride-share matches. We provide several methods for generating stable or nearly stable ride-share matching solutions. We empirically show that not taking stability considerations into account can create relatively unstable matching solutions. We also show that enforcing complete stability requires only a small price measured by the reduction of the possible total system vehicle-mile savings and is likely to lead to more sustainable ride-share systems in the long run.

This paper presents mathematical programming approaches to quickly find good stable or nearly stable matchings for single-rider, single-driver dynamic ride-sharing. Since this is the first paper to our knowledge that considers the notion of stability in the context of ride sharing, we see many opportunities for further research. One opportunity is the development of fast methods to determine (nearly) stable matchings in a setting in which the participants agree to matches in which drivers can serve multiple riders along their route. The results of our experiments with the rolling horizon suggest that a temporal decomposition of the problem may provide a good starting point for further research in this direction.

References

Agatz NAH, Erera A, Savelsbergh MWP, Wang X (2011) Dynamic ride-sharing: A simulation study in metro Atlanta. *Transportation Res. Part B* 45(9):1450–1464.

Agatz NAH, Erera A, Savelsbergh MWP, Wang X (2012) Optimization for dynamic ride-sharing: A review. *Eur. J. Oper. Res.* 223(2): 295–303.

American Automobile Association (2009) Your driving costs: How much are you really paying to drive? Technical report, American Automobile Association, Heathrow, FL.

American Automobile Association (2013) Your driving costs: How much are you really paying to drive? Technical report, American Automobile Association, Heathrow, FL.

Amey A (2011) Proposed methodology for estimating rideshare viability within an organization: Application to the MIT community. Transportation Res. Board Annual Meeting 2011, Paper 11-2585.

Anshelevich E, Das S, Naamad Y (2009) Anarchy, stability, and utopia: Creating better matchings. Mavronicolas M, Papadopoulou VG, eds. *Algorithmic Game Theory*, Lecture Notes Comput. Sci., Vol. 5814 (Springer, Berlin Heidelberg), 159–170.

Anshelevich E, Dasgupta A, Kleinberg J, Tardos E, Wexler T, Roughgarden T (2008) The price of stability for network design with fair cost allocation. *SIAM J. Comput.* 38(4):1602–1623.

Arslan A, Agatz N, Kroon L, Zuidwijk R (2018) Crowdsourced delivery: A dynamic pickup and delivery problem with ad-hoc drivers. *Transportation Sci.*, ePub ahead of print July 9, https://doi.org/10.1287/trsc.2017.0803.

Baldacci R, Maniezzo V, Mingozzi A (2004) An exact method for the car pooling problem based on Lagrangean column generation. *Oper. Res.* 52(3):422–439.

Clark S (2006) The uniqueness of stable matchings. *Contributions Theor. Econom.* 6(1):1–28.

Echenique F, Wilson AJ, Yariv L (2016) Clearinghouses for two-sided matching: An experimental study. *Quant. Econom.* 7(2): 449–482.

European Environment Agency (2010) Occupancy rates of passenger vehicles. Technical report, European Environment Agency, Copenhagen.

Furuhata M, Dessouky M, Ordónez F, Brunet ME, Wang X, Koenig S (2013) Ridesharing: The state-of-the-art and future directions. *Transportation Res. Part B* 57:28–46.

Gale D, Shapley LS (1962) College admissions and the stability of marriage. *Amer. Math. Monthly* 69(1):9–15.

Geron T (2013) Tickengo's Willie Brown wants revenue cap for ride-sharing drivers. *Forbes* (January 28), http://www.forbes.com/sites/tomiogeron/2013/01/28/tickengos-willie-brown-wants-revenue-cap-for-ride-sharing-drivers/.

Ghoseiri K, Haghani A, Hamedi M (2011) *Real-time Rideshare Matching Problem* (Mid-Atlantic Universities Transportation Center, University Park, PA).

Irving RW (1994) Stable marriage and indifference. *Discrete Appl. Math.* 48(3):261–272.

Irving RW, Leather P, Gusfield D (1987) An efficient algorithm for the "optimal" stable marriage. *J. ACM* 34(3):532–543.

Iwama K, Miyazaki S (2008) A survey of the stable marriage problem and its variants. *Proc. Internat. Conf. Informatics Ed. Res. Knowledge-Circulating Soc.* (IEEE Computer Society, Washington, DC), 131–136.

Iwama K, Manlove D, Miyazaki S, Morita Y (1999) Stable marriage with incomplete lists and ties. Wiedermann J, van Emde Boas P, Nielsen M, eds. *Automata, Languages and Programming*, Lecture Notes Comput. Sci., Vol. 1644 (Springer-Verlag, Berlin Heidelberg), 443–452.

Koutsoupias E, Papadimitriou C (1999) Worst-case equilibria. Meinel C, Tison S, eds. *Proc. 16th Annual Conf. Theoret. Aspects Comput. Sci.*, Lecture Notes Comput Sci., Vol. 1563 (Springer-Verlag, Berlin Heidelberg), 404–413.

Lee A, Savelsbergh M (2015) Dynamic ridesharing: Is there a role for dedicated drivers? *Transportation Res. Part B* 81:483–497.

Manlove DF, Irving RW, Iwama K, Miyazaki S, Morita Y (2002) Hard variants of stable marriage. *Theor. Comput. Sci.* 276(1): 261–279.

McGuckin N, Srinivasan N (2003) Journey to work trends in the United States and its major metropolitan areas 1960–2000. Technical report, U.S. Department of Transportation Federal Highway Administration, Washington, DC.

Roth AE (1984) The evolution of the labor market for medical interns and residents: A case study in game theory. *J. Political Econom.* 92(6):991–1016.

Roth AE (1990) New physicians: A natural experiment in market organization. *Science* 250(4987):1524–1528.

Roth AE (1991) A natural experiment in the organization of entry-level labor markets: Regional markets for new physicians and surgeons in the United Kingdom. *Amer. Econom. Rev.* 81(3):415–440.

Roth AE (2008) What have we learned from market design? *Econom. J.* 118(527):285–310.

Roth AE, Sotomayor MAO (1990) *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*, Econometric Society Monographs, Vol. 18 (Cambridge University Press, Cambridge, UK).

Roth AE, Rothblum UG, Vande Vate JH (1993) Stable matchings, optimal assignments and linear programming. *Math. Oper. Res.* 18(4):803–828.

Rothblum UG (1992) Characterization of stable matchings as extreme points of a polytope. *Math. Programming* 54(1–3):57–67.

Shapley LS (1953) A value for n-person games. Kuhn HW, Tucker AW, eds. *Contributions to the Theory of Games*, Ann. Math. Stud., 28 (Princeton University Press, Princeton, NJ), 307–317.

Sivak M (2013) Effects of vehicle fuel economy, distance travelled, and vehicle load on the amount of fuel used for personal transportation in the U.S: 1970–2010. Technical report, University of Michigan Transportation Research Institute, Ann Arbor.

Stiglic M, Agatz N, Savelsbergh M, Gradisar M (2015) The benefits of meeting points in ride-sharing systems. *Transportation Res. Part B* 82:36–53.

Stiglic M, Agatz N, Savelsbergh M, Gradisar M (2016) Making dynamic ride-sharing work: The impact of driver and rider flexibility. *Transportation Res. Part E* 91:190–207.

Vande Vate JH (1989) Linear programming brings marital bliss. *Oper. Res. Lett.* 8(3):147–153.

Wolfler Calvo R, de Luigi F, Haastrup P, Maniezzo V (2004) A distributed geographic information system for the daily car pooling problem. *Comput. Oper. Res.* 31(13):2263–2278.