Stable Matching for Dynamic Ride-sharing Systems

Xing Wang Niels Agatz Alan Erera

June 23, 2014

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 the most 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.

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, like Lyft, UberX, Carticipate, Avego, Sidecar and Flinc have emerged in both the U.S. 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 U.S. 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 as it allows them to share trip related expenses. The cost to own and operate a vehicle have risen sharply over the last years, with fuel prices in the US rising 52% between 2009 and 2013 (American Automobile Association, 2009, 2013).

In this paper, we consider a dynamic ride-share system setting where 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 that specifies 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 as participants are always better off in a match than driving alone.

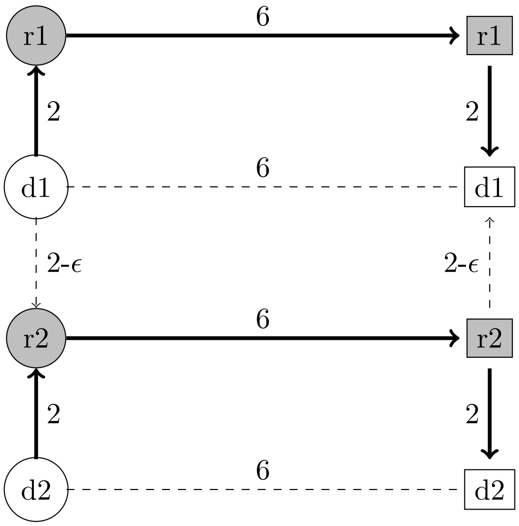
An automated process for establishing shared rides should be fast and require minimal effort from the participants. Centralized planning and coordination typically leads to better system-wide performance than uncoordinated decentralized approaches. In Agatz et al. (2011), we have shown the potential benefits of using optimization-based solution approaches to match riders and drivers compared to 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, a driver and/or rider may reject a match if they believe they can establish a better match on their own. This is related to the notation 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 system-wide vehicle miles savings of 4 and individual savings of 1 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 4 to $2 + 2\epsilon$. In the worst-case, when epsilon approaches 0, the price of anarchy could be as large as 50% degradation in vehicle-miles 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 will 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 introduce the concept of stability in the context of dynamic ride-sharing and to provide optimization 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.

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 the Atlanta region. 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

In this section, we review two streams of existing 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 et al. (2004); 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); Ghoseiri et al. (2011) propose similar approaches for a centralized dynamic ride matching service to match up drivers and riders.

There is a large stream of literature on methods to establish stable matches in two-sided matching markets (Roth and Sotomayor, 1992). Typical applications include the centralized matching of college/public high school admissions and between medical students and residency programmes. 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 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 which can be found in $O(n^2)$ time. They introduce an algorithm that involves several rounds which 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 where a driver (rider) i may receive identical benefit for a match with rider (driver) j or j', and thus does not strictly prefer one match to the other. Manlove et al. (2002) show interestingly that in such bipartite matching problems with both incomplete preference lists and also non-strict 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 non-strict 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 will be alternative stable matchings in our ride-share matching application, it will be useful to consider matching problems that include a utility objective. **?** considers the stable marriage problem where 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 where the bipartite graph is not complete. A simpler proof of the primary result in Vande Vate (1989) is provided by Roth et al. (1993). Mathematical programming formulations for stable matching problem variants will be used in our study.

The degradation of a system-optimal objective due to 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 a 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-miles 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 et al. (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 on systems where at most one pickup and delivery can take place during the trip and where riders do not transfer from one driver's vehicle to another. Moreover, 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 $l(s)$,

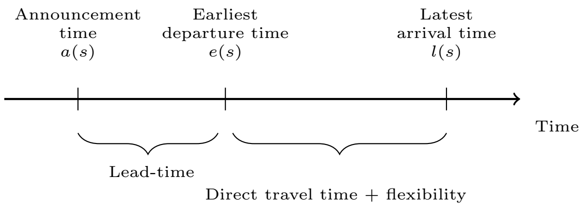

Announcement time $a(s)$
Earliest departure time $e(s)$
Latest arrival time $l(s)$
Time
Lead-time
Direct travel time + flexibility

Figure 2: Time line of a trip announcement.

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 (see Figure 2).

To asses the benefits of a particular match for an individual participant, we calculate the cost savings compared to traveling alone. For simplicity, we focus on a setting where riders have a car available which 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-miles savings from ride-sharing as compared to 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 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 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-miles 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 its 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-miles 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*, a well-known cost-allocation method in cooperative game theory.

We assume that the preferences of participants for different ride-share matches depends solely on the potential financial benefits, *i.e.*, the cost savings as compared to 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 that 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 $\mu(d) \neq r$, $r >_d \mu(d)$, and $d >_r \mu(r)$; 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 will 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 will consider single-stage optimization problems where 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, non-strict 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-miles savings, we now describe a greedy matching heuristic given a set S of announcements that consist 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.

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 is 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. □

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 where 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 et al. (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.

$$\begin{aligned}
\text{Maximize} \quad & \sum_{i,j \in A} c_{i,j} x_{i,j} \\
\text{subject to} \quad & \sum_{j \in D} x_{i,j} \leq 1 \quad \forall i \in R && (1) \\
& \sum_{i \in R} x_{i,j} \leq 1 \quad \forall j \in D && (2) \\
& \sum_{j' \succeq_i j} x_{i,j'} + \sum_{i' \succeq_j i} x_{i',j} + x_{i,j} \geq 1 \quad \forall (i, j) \in A && (3) \\
& x_{i,j} \in \{0, 1\}, \quad \forall (i, j) \in A && (4)
\end{aligned}$$

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 she prefers at least as well as rider i, or rider i is matched with another driver j' that she 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

$$\sum_{j' \succ_i j} x_{i,j'} + \sum_{i' \succ_j i} x_{i',j} + x_{i,j} \geq 1 \quad \forall (i,j) \in A \tag{5}$$

Roth et al. (1993) show that the linear relaxation of the system (1), (2), (5), and (4) has a special form:

Theorem 2; Roth, et al. 1993. Let C be the convex polytope corresponding to all feasible solutions to the linear relaxation of constraint system (1), (2), (5), and (4). 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 $c_{i,j} = 1$ 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 where 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) where 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:

$$\sum_{j' \succeq_i^\epsilon j} x_{i,j'} + \sum_{i' \succeq_j^\epsilon i} x_{i',j} + x_{i,j} \geq 1 \quad \forall (i, j) \in A \tag{6}$$

where the notation $i' \succeq_j^\epsilon i$ 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:

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{i \in R, j \in D} y_{i,j} \\
\text{subject to} \quad & \sum_{j \in D} x_{i,j} \leq 1 \quad \forall i \in R && (7) \\
& \sum_{i \in R} x_{i,j} \leq 1 \quad \forall j \in D && (8) \\
& y_{i,j} \geq 1 - \left(\sum_{j' \succeq_i j} x_{i,j'} + \sum_{i' \succeq_j i} x_{i',j} + x_{i,j} \right) \quad \forall (i,j) \in A && (9) \\
& \sum_{i \in R, j \in D} c_{i,j} x_{i,j} \geq \beta O && (10) \\
& x_{i,j}, y_{i,j} \in \{0,1\}, \quad \forall (i,j) \in (R \times D), \beta \in [0,1] && (11)
\end{aligned}
$$

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 the 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.

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 excluding 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 are 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, where drivers 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-miles 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

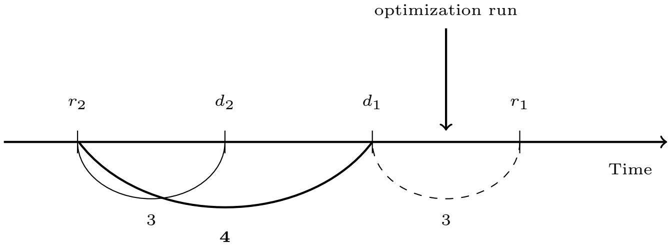


Figure 3: Prevented blocking pair d_1-r_2

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 3 where the numbers below the arcs represent the savings for the match between that particular rider and driver. The time line shows the announcement times of the different trip announcements. The *a posteriori* optimal solution would match rider 1 ($r1$) to driver 1 ($d1$) and rider 2($r2$) to driver 2 ($d2$) with total savings of 6. This would create a blocking pair between $d1$ and $r2$ because they would prefer being matched to each other than to their matching partners. Now consider what occurs in the dynamic problem where $d1$, $r2$ and $d2$ place their trip announcements before $r1$, and the rolling horizon framework initiates an optimization run at some time between the announcement time of $d2$ and $r1$. Without $r1$ present, the optimal matching includes only the *stable* match between $d1$ and $r2$. If the implied latest commitment time of $d1$ and $r2$ is earlier than the announcement time of $r1$, then the match $d1-r2$ is finalized before $r1$ and $d1$ 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 small example in Figure 4. Since the match between rider 1 ($r1$) and driver 1 ($d1$) has the largest savings, $d1$ prefers being matched to $r1$, *i.e.*, $r1 >_{d1} r2$. This matching also represents the solution that maximizes the savings. Now consider a situation in which rider 2 ($r2$) and driver 1 ($d1$) announce their trips before rider 1 ($r1$). Without $r1$ present as an option, we would match $r2$ and $d1$. If the next optimization run after the announcement time of request $r1$ occurs after the implied latest departure time of match $r2 - d1$, then we miss

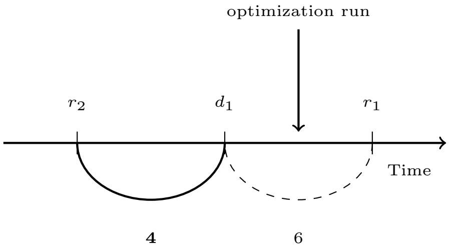


Figure 4: Created blocking pair d_1-r_1

the opportunity to match $r1$ and $d1$ 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 where at least one of the participants is currently not in a match. This cannot occur in an *a posteriori* optimal solution, since by definition such a solution cannot be optimal given the objective of maximizing total system vehicle-miles 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 of 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 2.66GHz Xeon E5430 Linux machine with 32GB 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 5 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 generate 5 random streams of trips for use within our simulations as follows. Each travel analysis zone is considered to be a possible origin and a possible 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 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.

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 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 time flexibility value from the latest departure time. Furthermore, the announcement time is calculated by subtracting an announcement lead time value from the earliest departure time.

6.2 Evaluation Criteria

To evaluate the performance of a specific ride-share matching solution, we compute the following statistics for each solution:

1. Success rate (S): the number of matched trips divided by the total number of trip announcements

2. Total system-wide vehicle-mile savings (M): the sum of the vehicle-mile savings of all ride-share matches divided by the total vehicle-miles that would be incurred had all announced

participants driven alone

3. Average individual savings per trip (C): half of the vehicle-mile savings of a ride-share match divided by the distance the participant would have driven alone, averaged over all matched participants

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

To asses the stability of a matching solution, we propose the following measures:

- Participant rate in blocking pairs (B): the number of unique participants in at least one blocking pair divided by the number of matched participants, separated for riders B_r and drivers B_d.

- Average number of blocking pairs per participant (R): the total number of blocking pairs divided by the number of unique participants in at least one blocking pair, separated for the riders R_r and drivers R_d.

- Average unrealized potential savings (Q): 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.

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

6.3 Stability of Solutions that Maximize System-wide Savings

In this section, we asses the stability of different matching solutions that do not explicitly constrain solutions to be stable. We focus initially on *a posteriori* matching optimization where an optimal matching for a complete, known set of daily announcements S is determined in hindsight. The impact of dynamics on the results will be discussed in the following section.

While the maximum weight stable matching problem with incomplete preference list and ties is generally difficult, we quickly find optimal stable solutions for all our practical instances. To assess the difference between the best and the worst stable solution, 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 minimizing 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.

Table 1: Stable matching solution quality comparison

	S(%)	M (%)	C (%)	D (%)	B_r (%)	B_d (%)	R_r	R_d	Q (%)	δ (%)
1%										
MWM	65.5	21.9	28.7	19.3	19.4	18.2	1.3	1.3	19.6	
MWSM	62.0	20.9	29.1	16.3	-	-	-	-	-	4.6
MinWSM	61.9	20.9	29.1	16.3	-	-	-	-	-	4.6
2%										
MWM	72.4	25.8	30.7	19.0	21.3	20.0	1.3	1.4	17.3	
MWSM	68.6	24.6	31.0	15.6	-	-	-	-	-	4.7
MinWSM	68.6	24.6	31.0	15.6	-	-	-	-	-	4.7
4%										
MWM	78.0	29.2	32.9	18.7	23.9	22.2	1.3	1.4	15.8	
MWSM	73.9	27.9	32.9	14.8	-	-	-	-	-	4.5
MinWSM	73.8	27.9	32.9	14.8	-	-	-	-	-	4.5

Table 1 presents the results for different participant rates averaged over 5 simulation runs, *i.e.*, 1% (11,563 announcements), 2% (23,155 announcements) and 4% (46,156 announcements). The results show that the solutions that maximize the total system-wide vehicle mile savings are relatively unstable. The number of matched participants in at least one blocking pair ranges from 18.2% to 23.9%, which indicates that there is 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 at least have one feasible match option to be in a blocking pair.

From the table, 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 5 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, in order 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.

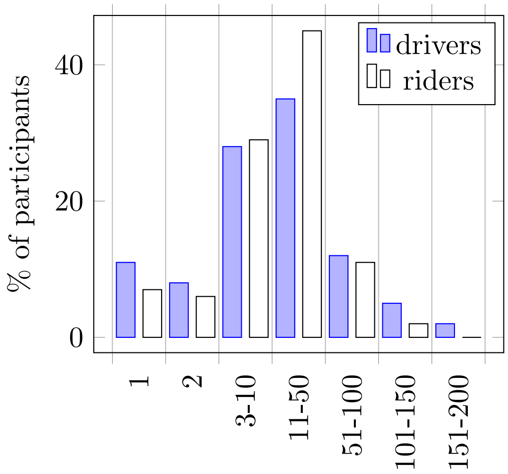


Figure 5: Number of feasible matches per participant

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 to 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-miles savings of blocking pairs for the 2% participant rate in Figure 6. The figure shows that approximately 30% of the participants have more than 1 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 6.6 we study the impact of relaxing the value at which participants perceive blocking pairs.

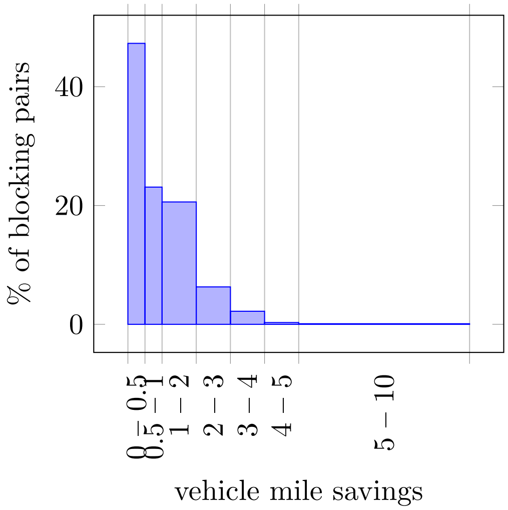


Figure 6: Unrealized savings in blocking pairs

Table 1 also shows that enforcing stability by solving the stable ride-share problem ($MWSM$) comes at the cost of deteriorating system-wide solution quality. However, the price of stability is relatively small and represents a 5% approximate reduction 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 not too many 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-miles savings may explain the relative small price of stability. These results are in line with the findings of Anshelevich et al. (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 weight stable matching 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 methods presented in Section 4.1 may suffice to find good stable matching solutions.

6.4 Stability of Maximum Savings Solutions in the Dynamic Setting

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 re-optimization 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 to an *a posteriori* benchmark in which we solve the off-line problem that considers all trips for that day simultaneously and enforces *a posteriori* stability.

As was 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* (B^-). 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 B_r^- and drivers B_d^-. If both people in a blocking pair are unmatched, Echenique et al. (2009) refer to this as an *available* blocking pair.

In Table 2, 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 also see that these blocking pairs represent higher unrealized savings. This suggest 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 due to the fact that 30.7% of the participants in these *a posteriori* blocking pairs are currently not in a match. We also see that these blocking pair participants are part of a remarkably large number of blocking pairs. Again, this results from currently unmatched participants that may create blocking pairs with many potential ride-share partners.

Table 2: Stability measures for dynamic matching solutions

	S(%)	M(%)	B_r (%)	B_d (%)	R_r	R_d	B_r^- (%)	B_d^- (%)	Q (%)
—a-posteriori—									
MWM	72.4	25.8	21.3	20.0	1.3	1.4	0	0	17.3
MWSM	68.6	24.6	-	-	-	-	-	-	-
—dynamic—									
MWM	72.2	25.4	25.0	22.4	1.3	1.5	3.6	0.8	20.3
MWSM	68.6	24.4	4.5	4.1	114.9	124.9	31.7	12.4	44.8

6.5 Maximizing the Number of Matches

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 3 we report simulation results when we maximize the total number of matched participants in the system by solving *a posteriori* maximum cardinality matching (MCM) problems. The different stability measures suggest that these matching solutions are less stable than those that maximize system vehicle-miles savings (Table 1). 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 of 12%. 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.

Table 3: Stability measures for maximum cardinality matching solutions

	S (%)	M (%)	C (%)	D (%)	B (%)	Q (%)	δ (%)
MWM	72.4	25.8	30.7	19.1	20.6	17.3	
MCM	77.2	25.2	30.7	22.2	31.7	34.5	
—stable—							
MCSM	68.7	24.6	34.3	15.7	-	-	12
MWSM	68.6	24.6	31.0	15.6	-	-	4.7

6.6 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 as described in Section 4.3. We vary the perceived instability threshold ϵ between 0 and 5 miles. Recall that threshold ϵ defines the minimum amount of difference in vehicle-miles savings between a participant's current match and a potential blocking pair match for which the participant can perceive the blocking pair; smaller savings are assumed to be imperceptible. A threshold of $\epsilon = 0.5$ miles, for instance, denotes that participants only perceive a blocking pair if the difference in vehicle-miles savings is greater than 0.5 miles.

Figure 7 presents the price of stability when using different perceived instability thresholds. A participant rate of 2% is used and the results are again averaged over 5 announcement streams. As expected, we see that the price of stability decreases with the perceived instability threshold. The results show that the price of stability is negligible if participants cannot perceive blocking pairs with a value of less than 1 mile of additional savings beyond the system match. The figure also shows the number of non-perceived blocking pairs in the relaxed stable solutions for the different perceived instability thresholds. We observe that an increase in the perceived instability threshold corresponds to a substantial increase in the number of imperceptible blocking pairs.

6.7 Relaxing System Optimality

In this section, 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 8 presents the results when allowing 1% or 5% deviation from unstable maximum vehicle-miles savings objective function value. Again, we use a participant rate of 2% and average the results over 5 announcement streams. 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 from 20% to 8% of the total number of matched participants. The results suggest that

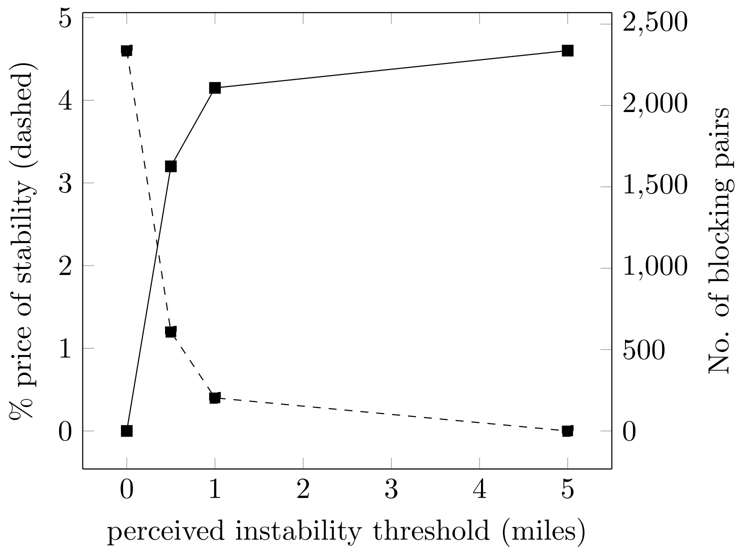


Figure 7: The impact of relaxing perceived instability

allowing a small degradation in the system's total savings allows the creation of significantly more stable solutions.

7 Modeling Ride-share System Participant Behavioral Response to Unstable Matchings

To investigate the potential impact of instability on the performance of a ride-share matching system over time, we attempt to provide a reasonable model for the participants' response to proposed matches that are unstable. 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.

7.1 Experimental Setup

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-

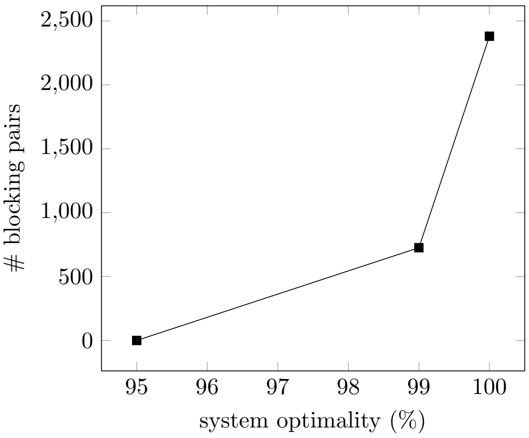


Figure 8: The impact of relaxing system optimality

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 1 hour. We now assume that this set of potential participants place trip announcements each day. However, the exact timing of their trip may vary a bit 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, each participant is 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 given a matching. In most real systems, it will likely be nearly impossible for participants to know complete announcement information for all other participants. Thus, we also consider a limited visibility scenario in which each participant only knows 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.

7.2 System Degradation Under Match Rejections

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 her current system match to each possible feasible other match with another known participant, in order of highest to lowest cost savings. If she finds a match with 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 be now be unavailable for future proposals. This procedure is similar to the Gale–Shapley algorithm 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 et al., 2009).

Figure 9 depicts simulated results for 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 due to *a posteriori* stability, no participants reject their system matches. The benchmark is compared to the full and limited visibility scenarios which establish a system matching each day that maximizes total system vehicle-miles 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-miles savings produced by the ride-share matches established (both system and extra-system matches), measured as a fraction of the optimal unstable vehicle-miles savings.

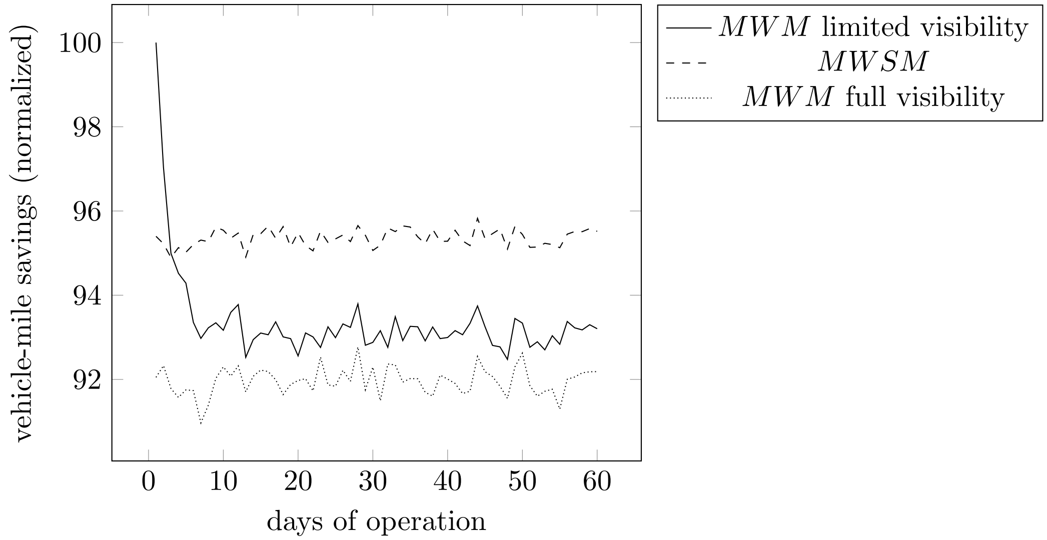
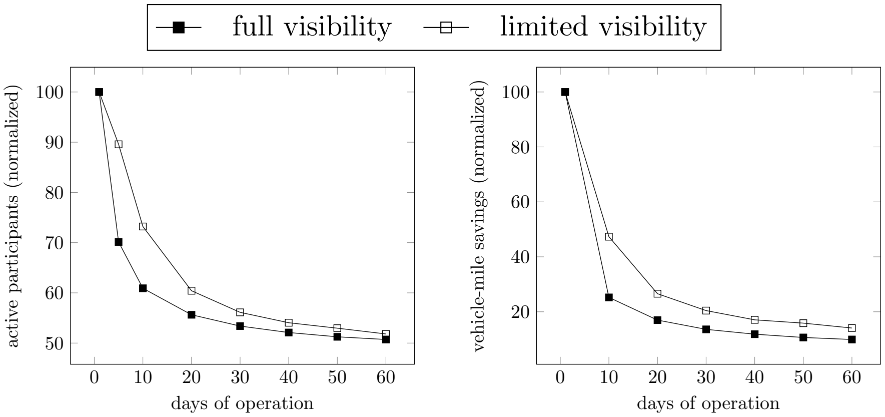


Figure 9: System Performance Degradation when Participants Reject Unstable Matches

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 about two weeks, the system appears to reach a steady state in which the system-wide vehicle miles savings are approximately 93% of the hypothetical unstable maximum. We see that this is close to the degradation that results in the scenario where all participants have full visibility of all other participants' announcements, which appears to reach a steady state of 92% of the hypothetical unstable maximum.

7.3 System Degradation under Participant Loss

In the most pessimistic scenario, participants may stop posting announcements to the ride-share matching system if they discover that it does not always provide them with stable matches. Suppose for illustration that a participant will never post another trip announcement to the ride-share system provider if they discover that they are in a blocking pair in the proposed system matching solution. Figure 10 provides an overview of the performance of the system over time in this scenario, where again each day a system optimal matching is generated that ignores stability constraints. We observe that the number of remaining participants in the system falls quickly and dramatically



(a) % active participants.

(b) % vehicle-mile.

Figure 10: Participants in non-stable matches stop participating

over time, and that a disproportionately large fraction of the vehicle-miles savings is lost. After 30 days, we lose almost 50% of the original participants but see a reduction in the corresponding system vehicle-miles savings of approximately 80%. Such results suggest that it is often the participants who have feasible matches that generate large savings who also tend to be in blocking pairs when stability constraints are ignored. Note that in this section, we do not account for the fact that some participants who decide to no longer announce their trips to the ride-share system provider may be able to establish extra-system matches that generate savings. Nonetheless, it seems clear that ignoring stability constraints is a risky strategy that has the potential to seriously degrade system performance and potentially lead to ride-share systems that cannot be sustained with participants.

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 likely 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 reduction of possible total system vehicle-miles savings, and is likely to lead to more sustainable ride-share systems in the long run.

References

N. A. H. Agatz, A. Erera, M. P. W. Savelsbergh, and X. Wang. Dynamic ride-sharing: A simulation study in metro atlanta. *Transportation Research Part B*, 45(9):1450–1464, 2011.

N. A. H. Agatz, A. Erera, M. P. W. Savelsbergh, and X. Wang. Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2):295–303, 2012.

American Automobile Association. Your driving costs: how much are you really paying to drive? Technical report, American Automobile Association, 2009.

American Automobile Association. Your driving costs: how much are you really paying to drive? Technical report, American Automobile Association, 2013.

A. Amey. Proposed Methodology for Estimating Rideshare Viability Within an Organization: Application to the MIT Community. *Transportation Research Board Annual Meeting 2011*, Paper 11-2585, 2011.

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

E. Anshelevich, S. Das, and Y. Naamad. Anarchy, stability, and utopia: Creating better matchings. In *Algorithmic Game Theory*, pages 159–170. Springer, 2009.

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

R. W. Calvo, F. de Luigi, P. Haastrup, and V. Maniezzo. A distributed geographic information system for the daily car pooling problem. *Computers & Operations Research*, 31(13):2263–2278, 2004.

S. Clark. The uniqueness of stable matchings. *Contributions in Theoretical Economics*, 6(1):1–28, 2006.

F. Echenique, A.J. Wilson, and L. Yariv. Clearinghouses for two-sided matching: an experimental study. *Social Science Working Paper, 1315*, 2009.

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

M. Furuhata, M. Dessouky, F. Ordez, M.E. Brunet, X. Wang, and S. Koenig. Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, 57(0):28 – 46, 2013. ISSN 0191-2615.

D. Gale and L.S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69(1):9–15, 1962.

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

K. Ghoseiri, A. Haghani, and M Hamedi. Real-time rideshare matching problem. *University of Maryland, Department of Civil and Environmental Engineering*, UMD-2009-05, 2011.

R. W. Irving. Stable marriage and indifference. *Discrete Applied Mathematics*, 48(3):261–272, 1994.

K. Iwama and S. Miyazaki. A survey of the stable marriage problem and its variants. In *Informatics Education and Research for Knowledge-Circulating Society, 2008. ICKS 2008. International Conference on*, pages 131–136. IEEE, 2008.

E. Koutsoupias and C. Papadimitriou. *Worst-Case Equilibria*, pages 404–413. Springer-Verlag, 1999.

D.F. Manlove, R.W. Irving, K. Iwama, S. Miyazaki, and Y. Morita. Hard variants of stable marriage. *Theoretical Computer Science*, 276(1):261–279, 2002.

N. McGuckin and N. Srinivasan. Journey to work trends in the united states and its major metropolitan areas 1960-2000. Technical report, US Department of Transportation Federal Highway Administration, 2003.

A.E. Roth. The evolution of the labor market for medical interns and residents: a case study in game theory. *The Journal of Political Economy*, pages 991–1016, 1984.

A.E. Roth. A natural experiment in the organization of entry-level labor markets: regional markets for new physicians and surgeons in the united kingdom. *The American economic review*, pages 415–440, 1991.

A.E. Roth and M.A.O. Sotomayor. *Two-sided matching: A study in game-theoretic modeling and analysis*, volume 18. Cambridge University Press, 1992.

A.E. Roth, U.G. Rothblum, and J.H. Vande Vate. Stable matchings, optimal assignments and linear programming. *Mathematics of Operations Research*, 18(4):803–828, 1993.

U.G. Rothblum. Characterization of stable matchings as extreme points of a polytope. *Mathematical Programming*, 54:57–67, 1992.

M. Sivak. 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, The University of Michigan Transportation Research Institute, 2013.

J.H. Vande Vate. Linear programming brings marital bliss. *Operations Research Letters*, 8:147–153, 1989.

ERIM Report Series *Research in Management*	
ERIM Report Series reference number	ERS–2015–006–LIS
Date of publication	2015–05–27
Version	27–05–2015
Number of pages	34
Persistent URL for paper	`http://hdl.handle.net/1765/78193`
Email address corresponding author	nagatz@rsm.nl
Address	Erasmus Research Institute of Management (ERIM) RSM Erasmus University / Erasmus School of Economics Erasmus University Rotterdam PO Box 1738 3000 DR Rotterdam, The Netherlands Phone: +31104081182 Fax: +31104089640 Email: info@erim.eur.nl Internet: `http://www.erim.eur.nl`
Availability	The ERIM Report Series is distributed through the following platforms: RePub, the EUR institutional repository Social Science Research Network (SSRN) Research Papers in Economics (RePEc)
Classifications	The electronic versions of the papers in the ERIM Report Series contain bibliographic metadata from the following classification systems: Library of Congress Classification (LCC) Journal of Economic Literature (JEL) ACM Computing Classification System Inspec Classification Scheme (ICS)