Dynamic Ride-Sharing: a Simulation Study in Metro Atlanta

Article *in* Procedia - Social and Behavioral Sciences · November 2011

DOI: 10.1016/j.trb.2011.05.017

Alan Erera
Georgia Institute of Technology

Martin Savelsbergh
Georgia Institute of Technology

Available online at www.sciencedirect.com

ScienceDirect

Procedia Social and Behavioral Sciences 17 (2011) 532–550

Procedia
Social and Behavioral Sciences

19th International Symposium on Transportation and Traffic Theory

Dynamic Ride-Sharing: a Simulation Study in Metro Atlanta

Niels Agatz[a], Alan L. Erera[b], Martin W.P. Savelsbergh[c], Xing Wang[b]

[a]*Rotterdam School of Management, Netherlands*
[b]*School of Industrial and Systems Engineering, Georgia Institute of Technology*
[c]*Commonwealth Scientific and Industrial Research Organisation (CSIRO), Sydney, Australia*

Abstract

Smartphone technology enables dynamic ride-sharing systems that bring together people with similar itineraries and time schedules to share rides on short-notice. This paper considers the problem of matching drivers and riders in this dynamic setting. We develop optimization-based approaches that aim at minimizing the total system-wide vehicle miles incurred by system users, and their individual travel costs. To assess the merits of our methods we present a simulation study based on 2008 travel demand data from metropolitan Atlanta. The simulation results indicate that the use of sophisticated optimization methods instead of simple greedy matching rules substantially improve the performance of ride-sharing systems. Furthermore, even with relatively low participation rates, it appears that sustainable populations of dynamic ride-sharing participants may be possible even in relatively sprawling urban areas with many employment centers.

1. Introduction

The growing ubiquity of mobile Internet technology has created new opportunities to bring together people with similar itineraries and time schedules to share rides on short-notice. Internet-enabled smartphones allow people to offer and request trips whenever they want wherever they are, enabling dynamic, on-demand ride-sharing [1]. Increasing the number of travelers per vehicle trip by effective usage of empty car seats by ride-sharing may of course enhance the efficiency of private transportation, and contribute to reducing traffic congestion, fuel consumption, and pollution. Moreover, ride-sharing allows users to share car-related expenses such as fuel costs.

By dynamic ride-sharing, we refer to a system where an automated process employed by a ride-share provider matches up drivers and riders on very short notice, which can range from a few minutes to a few hours before departure time. We believe ride matching should be largely automated in a dynamic setting to establish ride-shares in a way that requires minimal effort from the participants. Recently, many new companies have emerged that offer dynamic ride-share services. For example, providers like Carticipate, EnergeticX/Zebigo, Avego, and Piggyback have recently started offering mobile phone applications that allow drivers with spare seats to connect to people wanting to share a ride.

The ride-share provider typically lets a user offer a ride as a *driver* or request a ride as a *rider*. To facilitate easy trip specification, applications allow users to store and select pre-defined locations such as home, work, and the

Email addresses: nagatz@rsm.nl (Niels Agatz), alan.erera@isye.gatech.edu (Alan L. Erera), martin.savelsbergh@csiro.au (Martin W.P. Savelsbergh), xwang35@isye.gatech.edu (Xing Wang)

doi:10.1016/j.sbspro.2011.04.530

grocery store. With a GPS-enabled phone, a user can select his current location as the origin of the trip. If a match is established, the ride-share provider proposes the arrangement to the participants. If the driver and the rider agree on the proposed arrangement, the driver picks up the rider at the agreed time and location. The ride-share provider may send the driver the rider's photo and a personal identification number to allow him to verify identity. The ride-share provider also automatically assesses a trip fee to the rider, of which the company receives a fixed percentage and the driver receives the remainder as reimbursement for costs.

Dynamic ride-sharing is distinguished from traditional carpooling, and is focused on single, non-recurring trips which do not require long-term commitments between people to travel together for a particular purpose. Single-trip ride-sharing is more flexible because it does not require rigid time schedules or itineraries over time. The trips are prearranged (but on short notice) which means that the participants agree to share a ride in advance, typically while they are not yet at the same location. This is also different from the spontaneous, so-called casual carpooling (see *e.g.*, [2]) in which riders and drivers establish a ride-share on the spot, similar both to hitch-hiking and also to hailing a taxi on the side of the street.

While dynamic ride-sharing has been considered in recent research efforts (see *e.g.*, [3]), the development of algorithms for optimally matching drivers and riders in real-time has not received attention from the transportation optimization community to date. Since ride-shares are established on-demand, a ride-sharing system is similar to other on-demand forms of passenger transit such as taxis and dial-a-ride services like airport shuttles. The key planning tasks in on-demand transportation are the assignment of passengers to vehicles and the sequencing of stops for pickup and delivery. See [1] for an in-depth literature review in these areas, and a systematic comparison between dynamic ride-sharing and other modes of passenger transit.

In this paper, we present methods for solving dynamic ride-share matching problems, and use computer simulations based on actual travel demand data from Metro Atlanta to test the performance of a practical dynamic ride-sharing system. The main contributions of this paper can be summarized as follows:

- We develop optimization approaches specifically tailored to the dynamics of a practical ride-share environment where new drivers and riders continuously enter and leave the system. The rolling horizon approach provides high quality solutions to practical dynamic ride-share problem instances; and
- We build a simulation environment based on travel demand model data from the Atlanta Regional Commission, and use it to test dynamic ride-sharing concepts. The simulation results suggest that dynamic ride-sharing may represent a useful option to reduce system-wide vehicle miles, reduce trips and save travel costs, even when participation rates are relatively small; and
- We demonstrate the value of more sophisticated optimization techniques over simple greedy matching methods in dynamic ride-sharing systems.

The remainder of the paper is structured as follows. In Section 2, we describe the dynamic ride-sharing setting and explain the planning issues that arise in this context. In Section 3, we explain our approach to solve the dynamic ride-share problem. In Section 4 we present a simulation study based on the travel demand model of the Atlanta Regional Commission. In Section 5 we focus on understanding the performance of a ride-sharing system over time. Finally, in Section 6, we summarize our main insights and discuss directions for future research.

2. The Dynamic Ride-share Setting

We consider a specific dynamic ride-share system setting that we believe is representative of many new and proposed systems. In this setting, 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 intends to be a driver, intends to be a rider, or is flexible to perform either role. A trip announcement also contains an origin and a destination location, and additional information that specifies its potential timing. With this information, the provider automatically establishes ride-shares over time, matching potential drivers and riders.

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 $s \in S$.

We furthermore adopt the following reasonable model of trip timing, assuming that most trips are made with some flexibility in their schedule [4]. For each announcement $s \in S$, the participant provides an earliest time $e(s)$ at which he can depart from his 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 (see Figure 1). For example, if a driver wished to arrive at his destination no later than $l(s)$, then we have time flexibility $f(s) = l(s) - e(s) - t_{v(s),w(s)}$. In this research, one condition for the feasibility of a ride-share match is that the participant for announcement s departs his origin no earlier than $e(s)$ and arrives at his destination no later than $l(s)$. We choose not to model any additional constraints that limit the amount of time participants spend traveling in-vehicle.

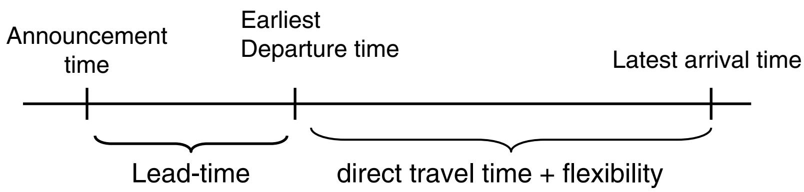

Figure 1: Time schedule information

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.

Although a potential driver may typically have several spare seats available (see *e.g.*, [5]), time considerations will restrict the number of stops he is willing to make in a single trip. To minimize the inconvenience of the participants, in this research we limit our attention to systems where at most one pickup and delivery can take place during the trip and no transfers occur (see Figure 2). This does not imply that a driver cannot accommodate multiple riders if they are traveling from the same origin to the same destination at the same time.

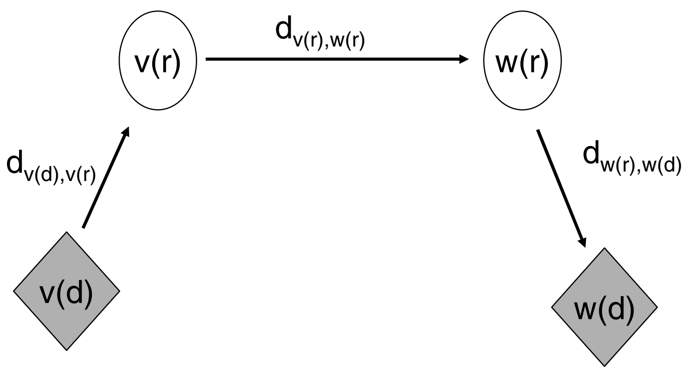

Figure 2: A Shared Trip between Driver d (squares) and Rider r (circles)

People may choose to participate in a ride-sharing to reduce travel costs. In this research, we focus on systems designed to enable users to share variable trip costs. When such costs are roughly proportional to distance traveled, 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 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 benefits. We consider a match feasible only if it provides positive cost savings: a ride-share between driver d and rider r is feasible only if $d_{v(d),w(d)} + d_{v(r),w(r)} - (d_{v(d),v(r)} + d_{v(r),w(r)} + d_{w(r),w(d)}) > 0$.

A trip announcement s is said to expire when the latest possible departure time $e(s)+f(s)$ occurs before a successful ride-share match can be found. Thus, ride-shares cannot be arranged for potential drivers that are already en-route. Furthermore, virtually all trips in practice are likely to be round trips. While a potential rider participant may choose to arrange ride-shares for the trips separately, some may not feel comfortable traveling to certain destinations without having a confirmed ride back. The need for round trip planning may necessitate that systems allow riders to place two trip announcements at the same time, and only agree to participate if both requests are matched in ride-shares. Of course, the return trip need not be with the same driver that provides the outbound trip.

Although ride-sharing systems may provide opportunities to increase the mobility of people that do not have access to public transit or a private vehicle, we focus on ride-sharing as a means to reduce travel costs, congestion and pollution. We therefore limit our attention to a setting where both drivers and riders have a car available which they could use to drive to their destination alone if no ride-share can be identified.

Given this setting, we explore ride-share optimization problems in which the ride-share provider seeks 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 is aligned with societal objectives for reducing emissions and traffic congestion. Furthermore, 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. Finally, if the ride-share provider is compensated with a fraction of the total travel cost savings of all participants, the objective is also consistent with *maximizing the revenues* of the provider.

3. Solving the Dynamic Ride-share Problem

3.1. Rolling Horizon Strategy

Since new driver and rider trip announcements continuously arrive each day, it seems clear that any dynamic ride-sharing 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 requests 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. At each execution of the algorithm, the planning horizon is "rolled" forward to include more known information, and the process continues. Our proposed approach uses a planning horizon that extends forward from the current time and captures all currently known requests, regardless of their timing during the day.

A key decision when implementing a rolling horizon solution approach is how frequently, and specifically when, to execute the planning algorithm. One possibility would be to initiate a matching optimization each time a new request becomes known. This, however, may lead to synchronization issues when a new announcement arrives before the end of the previous optimization run. For simplicity, therefore, we consider strategies that reoptimize at specific, regularly-spaced time points. Even so, in this study we ignore the time required to execute a planning algorithm, and assume that it is negligible.

In our solution approach, optimization run q at time $t(q)$ during an operational day considers all trip announcements s that were announced (at times $a(s)$) prior to $t(q)$, 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 as late as possible. Thus, a ride-share match 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 $\hat{l}(d,r)$ is given by $\min(l(r) - t_{v(r),w(r)} - t_{v(d),v(r)}, l(d) - t_{w(r),w(d)} - t_{v(r),w(r)} - t_{v(d),v(r)})$.

In the case where we determine round trip matches for riders, note that we also finalize the return ride-share match for a rider prior to the latest implied departure time of the driver for his outbound trip. Furthermore, for round-trip announcements in which the participant is willing to serve as a driver or rider, the role of the participant is finalized when his outbound ride-share match is finalized, and his role cannot change between the outbound and return trips; *i.e.*, a rider for an outbound trip cannot be scheduled in a return trip as a driver, and vice versa, since both cases are likely infeasible in practice.

In Section 3.2, we discuss the details of the optimization procedures used to determine matches within this rolling horizon approach.

3.2. Solving the Ride-share Matching Optimization Problem

Suppose that the optimization procedure is seeking to find the best ride-share matches from within the current set of active announcements, $S_A \subset S$. We first discuss the simplest case, where each participant declares whether he intends to be a driver or rider.

3.2.1. Fixed Driver, Rider Roles

There are two disjoint sets of announcements: a set $D \subset S_A$ representing driver trips, and a set $R \subset S_A$ representing rider trips. If the total benefit of a set of ride-share matches can be expressed as the sum of the benefits of individual matches, we can represent the ride-share problem using a maximum-weight bipartite matching model and then solve the problem using any linear programming or network optimization code. Since we consider a setting where the ride-share provider seeks to maximize the total distance savings produced for all participants, we can use this model as follows. 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 propose 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. To complete the specification, let x_{ij} be a binary decision variable equal to 1 if ride-share match (i, j) is proposed, and 0 if not. Then, a formulation of the maximum weight bipartite matching optimization problem to maximize system travel distance savings uses objective function $\sum_{i,j} c_{ij}x_{ij}$, along with a set of constraints to ensure that each driver and rider is included in at most one proposed ride-share match: $\sum_j x_{ij} \leq 1 \ \forall i \in R$ and $\sum_i x_{ij} \leq 1 \ \forall j \in D$.

To solve the problem in our computational study, we use the standard commercial optimization software CPLEX. We transform the bipartite matching into a network flow maximum cost circulation problem by adding a source node s and a sink node t, along with an arc from s to rider node $i \in R$ with zero cost and unit capacity and an identical arc from each driver node $j \in D$ to t. Connecting to t to s with a zero cost and no capacity completes the specification.

It is not difficult to extend the bipartite matching model to the case where some (or all) of the riders wish to schedule round trip matches. To do so, we simply need to ensure that if a rider is matched on his outbound trip, that he is also matched on his return trip. Such riders i will be represented with two separate rider nodes i^1 and i^2, representing the two trip segments respectively. To ensure that these two segments are either both matched or neither are matched, we must add a bundle constraint for each such round-trip rider: $\sum_j x_{i^1 j} - \sum_k x_{i^2 k} = 0$. The addition of constraints of this type, however, does not preserve the total unimodularity of the constraint matrix, and therefore must be solved using optimization software capable of handling binary integer programs.

3.2.2. Driver, Rider Role Assignment

We now consider the more complex case where some ride-share participants announce trips in which they are flexible to serve as drivers or riders. Clearly, ride-share match optimization in this case must not only decide on the assignment of riders to driver but also assign a role to each of the participants. It is therefore no longer possible to model this problem using bipartite matching, but we can instead use a general graph matching model as follows. Consider a directed network with a node for each announcement in S_A. A directed arc (i, j) between announcement i and announcement j is generated if the potential match is time feasible and has positive cost savings c_{ij} when i serves as a rider and j as a driver, and an arc (j, i) with cost savings c_{ji} if it is feasible for j to ride and i to drive. If both arcs are generated, then we retain only the one with larger cost savings c. The matching objective function again seeks to maximize the savings of selected matches over all possibilities: $\sum_{i,j} c_{ij}x_{ij}$. Then, a single matching constraint is used to ensure that each announcement is selected to be included with no more than one proposed match: $\sum_j x_{ij} + \sum_j x_{ji} \leq 1 \quad \forall \ i \in S$. Note that this constraint considers all outbound arcs ("rider" arcs) and inbound arcs ("driver" arcs) for announcement i.

The general graph matching problem can be solved with algorithms of polynomial complexity [see 6]. Again, however, if we need to solve problems with requests for round-trip matching, it is necessary to add bundle constraints that then require binary integer programming software. For this case, the required bundle constraints take the same form: $\sum_j x_{i^1 j} - \sum_k x_{i^2 k} = 0, \forall i^1, i^2 \in S$, where i^1 represents the outbound trip announcement and i^2 the return trip of participant i. Note that since we only bundle outbound arcs from i^1 and i^2, this constraint only matters when participant i is selected as rider. If i is flexible and is used as a driver, he may be matched only on outbound, only on return, or for both trips. Furthermore, note that these constraints also ensure consistent role assignments within a round trip of a rider, so that if a participant is matched as a rider on the outbound he must also be matched as a rider on the return. This is necessary since a participant who shared a ride to work likely does not have access to a vehicle for the return trip home.

It is also necessary when considering round-trip matching in this case to include both arcs (i, j) and (j, i) if they are both feasible, even if one dominates the other in terms of cost savings. For example, consider a problem in which i and k can be feasibly matched for the return trip of i, and greater cost savings are generated with k serving as the rider and i as the driver. If there is another participant j, and the only feasible matches are given by arcs (i^1, j), (i^2, k), and (k, i^2), an optimal solution may be to create matches (i^1, j) and (i^2, k) even if $c_{ki^2} > c_{i^2 k}$.

3.2.3. Greedy Approach

To gain some understanding of the value of optimization-based approaches in ride-share matching, we will compare the matching and integer programming methods described earlier with a strawman greedy algorithm. The greedy matching algorithm that we propose is a straightforward rule-based approach that a ride-share provider could use to match riders and drivers without requiring more sophisticated optimization software.

The greedy algorithm works as follows. First consider the case where all announcements are either rider or driver requests. Given a set S_A of active announcements, we determine for each rider announcement r the driver announcement d (if any) that represents a feasible match with the largest possible savings. Among all of these matches, we then select (r_m, d_m) with the largest savings and fix it. Requests r_m and d_m are then removed from S_A, and the process is repeated until no positive savings matches remain; this involves only recomputing new best feasible matches for any riders whose best potential previous match was d_m. For round trip scheduling, we follow the same procedure but only consider riders if they have feasible drivers for both trips and store the average positive savings of the outbound and return matched trip. Finally, for the flexible role case, we use the same procedure but consider each flexible role announcement twice, once as a rider and once as a driver.

3.3. Benchmarks

To evaluate the performance of our ride-matching solution approaches, we propose two benchmarks that represent upper bounds on solution quality. For both benchmarks, we solve a so-called *off-line problem* that considers simultaneously the complete set S of announcements received on a particular day. Each off-line problem has advantages over reality, since announcements are essentially known in advance, and thus optimal solutions determined using a technique presented in Section 3.2 are upper bounds on the quality of the matches determined sequentially in time using the same technique within the rolling horizon approach.

The two benchmarks are determined as follows. For the *a posteriori* benchmark, a driver-rider match is only considered feasible in the off-line problem if, in addition to the time feasibility and positive cost savings described earlier, the announcements could possibly be considered simultaneously within some set S_A in a rolling horizon approach, *i.e.*, if there is some overlap between the intervals between the announcement times and the implied latest departure times. The *static* benchmark provides a weaker bound, and drops this requirement for overlap; this benchmark essentially emulates a case where all participants announced their trips in advance on the day prior to traveling.

For instances in which riders and drivers announce fixed roles, each of the off-line optimization problems can be solved in reasonable compute times using CPLEX. However, when instances contain large numbers of announcements with flexible roles, it is difficult to solve the off-line problems to optimality and we therefore determine only a very good (but not provably optimal) solution using an iterative rounding procedure. In this procedure, we first solve the linear programming relaxation of the integer program, then fix certain variables x_{ij} to zero, and finally solve this restricted integer program. Specifically, we fix all outgoing arcs from the node representing participant i to zero if in the linear programming relaxation solution $\sum_j x_{ji} - \sum_j x_{ij} > 0$; this restricts this participant from being assigned as a rider, since the relaxed solution prefers him as a driver.

4. Numerical Experiments

We implemented the ride-share matching solution approaches detailed earlier and a simulation environment in *C++*, using CPLEX 11.1 as the linear and binary integer programming solver running on a quad-core 2.66GHz Xeon E5430 with 32GB RAM. We now detail the simulation study and its results.

4.1. Simulation Environment

To test the viability of dynamic ride-sharing and to study the merits of optimization for ride-share matching, we developed a simulation environment that considers work trips made in the Atlanta metropolitan region, in the U.S. state of Georgia. The Atlanta area represents a potentially interesting environment for ride-sharing since it does not have good public transport infrastructure and its freeway traffic congestion is among the most severe in the U.S. Also, many major U.S. metropolitan areas have similar urban forms, with low population density and many commercial employment hubs outside of the downtown core. It also represents a challenging test case due to its large size and the large number of automobile work trips. Dynamic ride-sharing concepts that work in Atlanta should also be likely to work in more densely populated urban environments, and perhaps more effectively.

The simulation environment is based on the 2008 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 (see Figure 3), 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 to generate estimates of the daily home-based work-related vehicle trips between all 2024 travel analysis zones (TAZs) within the region (see Table 1). For travel distances and times, we compute airline distances between TAZ population centroids and assume a constant average vehicle speed of 30 miles per hour. Thus, we approximate the true travel distances and times in the Atlanta region, and ignore any time-dependency in travel time caused by congestion. We also ignore any time expending during pick up or drop off of riders. We do not believe that these simplifications have a major impact on our conclusions.

Figure 3: 10-Country Atlanta Region

We generate 5 random streams of trips for use within our simulations as follows. Each travel analysis zone is considered 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 that we assume might consider participating in dynamic ride-sharing (the *participation rate*). Then, for each pair, we determine the number of trip announcements using a Poisson random variable with expected value equal to the computed expected number of trips. Each trip announcement is equally

Table 1: Home-Based Work Travel Information (ARC, 2008)

daily # round trips	2.96 million
daily vehicle-miles	32 million
avg. trip distance	10.8 mile
low occupancy trips	2.55 million
# o-d pairs	2.9 million
max trips per o-d	881
min trips per o-d	0.01

likely to be a rider announcement or a driver announcement, when roles are not flexible. Once an outbound trip announcement is generated from a to b, we assume that a return trip from b to a will occur and generate it also.

Trip timing information is also not available in the travel demand model data set. Therefore, we construct the time windows for each announcement as follows. For the outbound trip from home to work, we draw the latest departure time from a normal distribution with mean 7:30 a.m. and standard deviation 1 hour to model a typical morning peak [7], and calculate the latest arrival time by adding the direct travel distance 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 latest departure time. For the return trip from work to home, we draw a work day length value from a normal distribution with mean 9 hours and standard deviation 0.5 hour. To construct the time window for the return trip, we add the work day length to the earliest departure time and the latest arrival time of the initial trip.

In all experiments, unless specifically stated otherwise, we generate 5 different random trip announcement streams based on a 2% participation rate, a 30 minute announcement lead-time, and a time-flexibility of 20 minutes. Each stream represents a sample day. As commonly seen in practice (see for example the system operated by zebigo.com), we specify the flexibility as an absolute value rather than a value relative to the duration of the trip; a relative flexibility, *e.g.*, 25% of trip duration, will likely underestimate the flexibility for short trips and overestimate it for longer trips. We will also use a standard re-optimization frequency of 10 minutes within the rolling horizon solution strategy, commencing the first optimization run 10 minutes after the first announcement arrival each day. Importantly, we assume that if participants are notified of a feasible ride-share arrangement, they will always accept it. It would not be too difficult to extend this research to attempt to model the accept/reject behavior of potential participants, but we have chosen to ignore this idea in this initial study.

4.2. Base Case Computational Results

We now provide computational results for a base case in which participants are assumed to announce their intended roles in advance, and in which all announcements are for round trips. We consider three different participation rate levels: 1%, 2%, and 4%. For each scenario, we assess the value of the optimization-based approaches for ride-share matching by comparing the quality of the solutions found by the greedy algorithm (denoted GREEDY) and the bipartite matching with bundle constraints binary integer programming approach (denoted BIPART). Each rolling horizon solution is furthermore compared to the two off-line solution quality benchmarks.

We compute the following statistics to compare the different solution approaches, where the averages are computed over the 5 separate announcement streams:

1. *average success rate* (S): matched trip announcements divided by the number of trip announcements;
2. *average total system-wide vehicle miles savings* (M): miles saved for all announced trips versus if all individual trips were executed unmatched; and
3. *average individual cost savings per trip* (C): costs are assumed to be proportional to vehicle-miles driven, and cost savings are divided proportionally between driver and rider based on the lengths of their original trips.

Note that since we consider single-rider, single-driver ride-share matches only, $S/2$ corresponds to the percentage reduction in the number of vehicle trips among the population of announced trips.

Table 2: Base Case Solution Quality Comparison

	S (%)	M (%)	C (%)
—1%—			
GREEDY	28.2	10.5	26.2
BIPART	58.3	18.3	25.2
a posteriori	60.3	19.9	26.3
static	62.2	20.8	26.8
—2%—			
GREEDY	28.7	11.4	27.4
BIPART	67.0	22.3	27.3
a posteriori	68.7	23.8	28.3
static	70.3	24.6	28.6
—4%—			
GREEDY	28.3	12.2	29.0
BIPART	74.5	26.6	29.6
a posteriori	75.8	28.0	30.5
static	77.1	28.8	31.0

Table 2 demonstrates clearly that BIPART significantly outperforms GREEDY in terms of success rate ($28 - 36\%$) and vehicle-miles savings ($14 - 18\%$) over all three participation rate levels. The greedy algorithm strawman seems reasonable, but it does not yield good results in practice. Not surprisingly, the greedy approach generates good individual cost savings. It seems clear, however, that it is much more important to maximize the number of matches than the quality of the individual matches, and the integer programming technique does a much better job in this regard. Optimization-based approaches clearly appear to have much potential value in ride-share matching application. Both methods are fast and can solve even the very large off-line problems within a minute of compute time; the largest off-line problem with approximately 29,000 announcements required 78 seconds of compute time for BIPART.

Comparison to the *a posteriori* bound suggests that the rolling horizon approach is close to optimal for practical instances. This is not unexpected, since the trips of drivers and riders that can be feasibly and cost-effectively matched often have departure times that are close together and thus are likely to be considered in the same optimization run. The gap between the *a posteriori* bound and the rolling horizon approach decreases with the announcement density. A potential reason for this is that a higher announcement density leads to more feasible and cost-effective matches, thereby making the cost of committing to a less than optimal match smaller. The *static* benchmark demonstrates the further potential improvement possible given advance information from participants. If trips are announced further in advance of departure, this may allow the ride-share provider to establish matches that would otherwise be missed because compatible trips may not have been announced before the expiration time. For example, a compatible return ride may not yet been announced by the latest departure time of the initial trip of the rider. A more rare example would be a rider who has not announced by the implied latest departure time of the driver if they were to be matched, *i.e.*, if the travel time between the driver's origin and the rider's origin is greater than the rider's announcement lead-time.

The results also demonstrate that increasing the participation rate leads to a higher success rate, and also improves the average individual savings. That is, not only does the relative fraction of participants that find a ride-share increase, but also the individual savings from sharing the trip costs. This result quantifies the importance of density for ride-sharing, which of course is well known. Note also, however, that the relative advantage of BIPART increases with the participation rate. Thus, the optimization-based procedure provides additional advantage over simpler strategies when it considers more options during a run.

Since travel cost is assumed proportional to the travel distance, the reported system-wide vehicle-miles savings correspond to cost savings. Assuming an average per-mile direct cost of $0.54 [8], we see daily cost savings in these scenarios that range from approximately $27,000 (1% participation) to $152,000 (4% participation). Even the revenue from a small fraction of these savings may provide an interesting business opportunity for a private ride-share provider. For a participation rate of 2%, the average individual savings for the matched participants is approximately $1.90 per trip ($3.80 per round trip) which may provide sufficient incentive for participants (who may already be motivated by travel time savings in carpool lanes or concerns about the environment). Note also that the average additional in-vehicle travel time for the drivers ranges from 5.8 minutes for the 1% participation rate to 5.2 minutes for the 4% participation rate, which seems to be an acceptably small increase according to the findings of previous ride-sharing surveys [9].

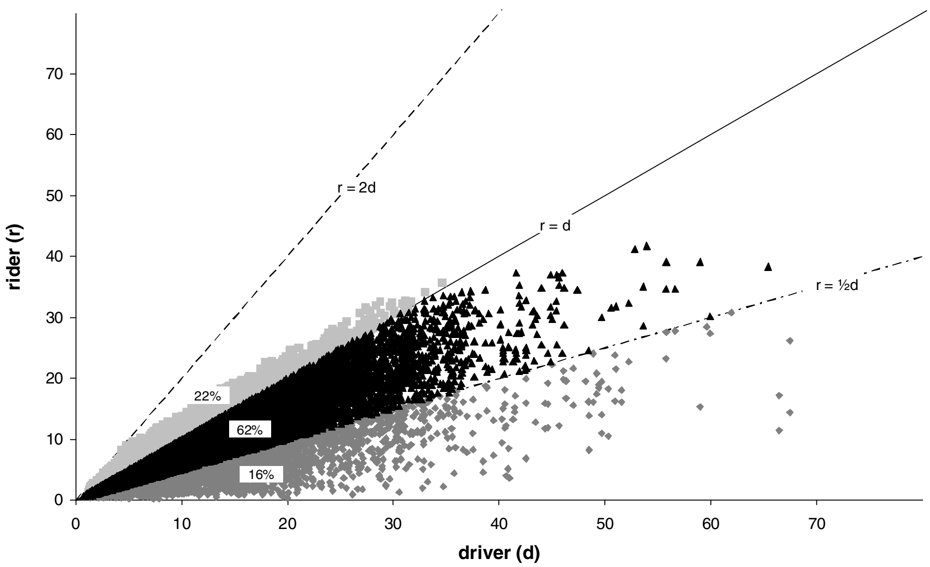

Figure 4: Original Trip Distances for Matched Participants

Next, we consider some additional characteristics of the solutions by examining the individual origin-destination distances of each driver-rider match. In Figure 4, we see that the rider's trip distance is typically smaller than the driver's original trip distance in a match; 78% of the matches lie below the diagonal where the driver's trip and rider's trip have the same length. This is not unexpected, since if the rider's trip is larger than the driver's trip, the additional driving distance required to accommodate the rider reduces potential distance savings for the pair. Recall that a match between rider r and driver d only produces cost savings if $d_{v(d),v(r)} + d_{w(r),w(d)} < d_{v(d),w(d)}$. There is no possibility for cost savings if the length $d_{v(r),w(r)}$ of the rider's trip is more than twice the distance $d_{v(d),w(d)}$ of the driver's, which further implies that the total driving required of a driver in a ride-share match cannot exceed twice $d_{v(d),w(d)}$.

Matches in which the rider has the longer trip distance (above the diagonal in Figure 4) generally involve participants with smaller individual trip distances. The driver's time flexibility makes matches between participants with longer trips less likely. Moreover, we see relatively few matches where the rider trips are significantly shorter than the matched driver trips. To understand this, note that maximizing vehicle-mile savings coincides with maximizing the travel distance when both participants are traveling together. Thus, there is more savings possible if a driver can travel with a rider who is traveling further.

Figure 5 depicts the success rate of for announced trips of different lengths, where each bucket represents roughly 25% of the daily announcements. For the driver trips, we see that the likelihood of a match increases with the length of the trip, again since longer trips correspond to more potential savings and also result in a higher likelihood of finding a compatible rider on the way. For the rider trips, we observe a trade-off between feasibility and savings with respect to trip length. Although smaller trips may easily find compatible drivers, they also represent smaller potential savings.

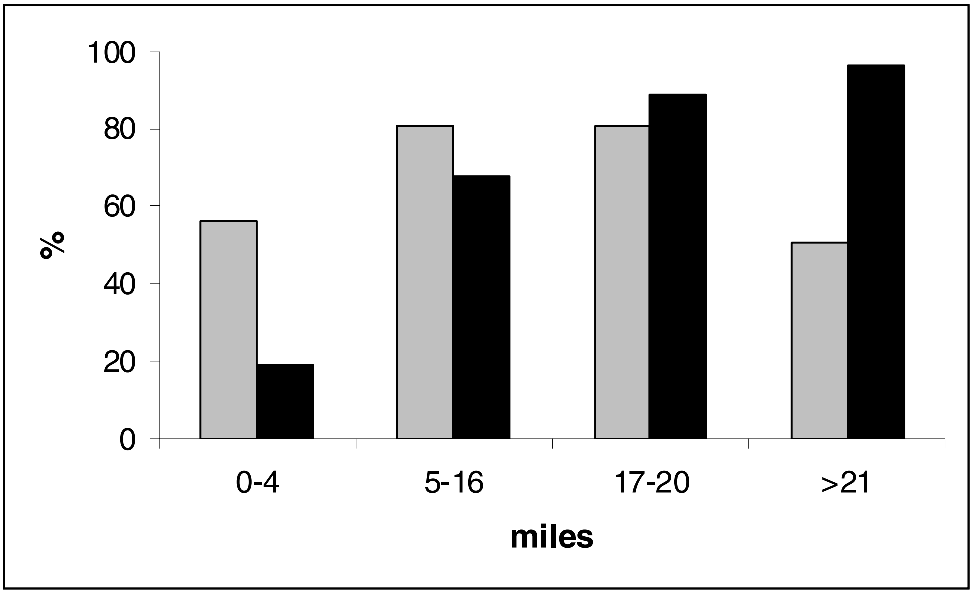

Figure 5: Success rates for riders (gray) and drivers (black) by original trip distance

Longer trips, on the other hand, may represent more savings but are also harder to match.

Next we focus on the likelihood of getting matched for announcements with different earliest departure times. Figure 6 shows that the highest success rates occur during the morning rush period (6 a.m. to 9 a.m.) and the evening rush period (3 p.m. to 6 p.m.). This is intuitive because these times have the highest announcement densities in our scenarios. A nice feature of dynamic ride-sharing, then, is that the high concentration of trips that leads to negative system impacts like congestion also leads to positive impacts on the performance of ride-sharing systems.

Finally, we consider the rolling horizon strategies in more detail by examining the impact on solution quality by changing the re-optimization timing and the commitment strategy. The strategy that re-optimizes after each minute coincides with a strategy that runs an optimization each time a new announcement is made. Recall that our base case assumption is that the potential ride-share matches found via optimization are not finalized until as late as possible. Here, we also examine an alternative strategy where all proposed matches are finalized immediately after the optimization run in which they were identified.

Table 3 presents the results for the 2% participation rate announcement streams. The results demonstrate that for our test scenario assumptions regarding announcement lead time and time flexibility, systems that employ the latest commitment strategy for matches should be optimized more frequently. However, if we commit matches immediately, we observe that there are advantages of optimizing less frequently since it allows the accumulation of more trip announcements between optimization runs. Although not depicted in these results, it should be clear that this benefit of optimizing less frequently will eventually reverse itself. When the time between optimization runs grows too large, missed matching opportunities become more and more prevalent. For a simple example, consider a rider who announces a trip at 8:01 and a driver who expires at 8:07 (but announced before 8:01). This driver-rider match may be missed when the time between re-optimization runs is greater than 6 minutes, *e.g.*, if optimizing at 8:00 and 8:10.

4.3. The Advantages of Flexible Roles

The previous results assume that all participants announce trips with fixed roles, as drivers or riders. Here, we focus on the other extreme where every participant is flexible to serve as a driver or a rider for his announced trip. In this case, the optimization problem considered during each optimization run cannot necessarily be solved to optimality quickly. Therefore, we configure the optimization with two stopping criteria: a maximum solution time limit of 200

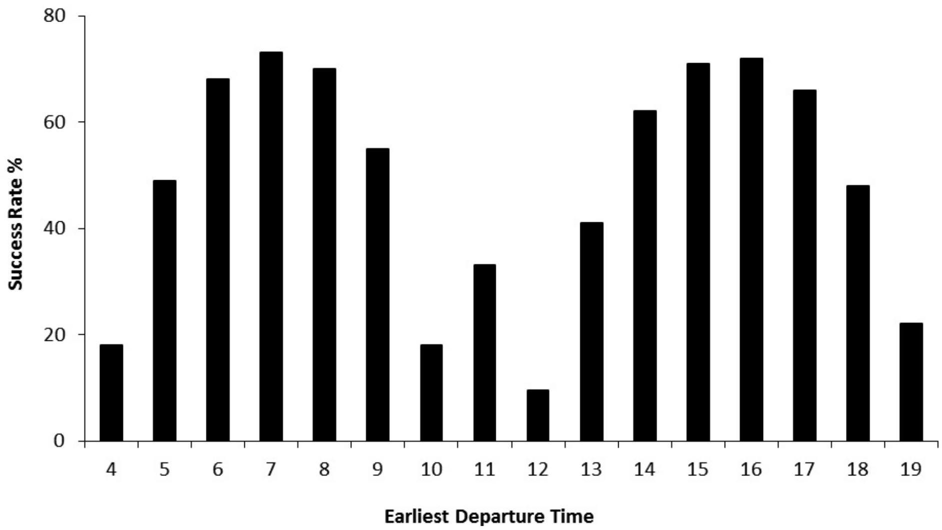

Figure 6: Success rate by time of day

seconds, or a feasible solution found that has an objective function value guaranteed to be no worse than 1% smaller than the optimal value (also known as 1% optimality gap in integer programming). Note then that it is possible that no feasible solution is found within the time limit; in this case, we use as the solution the proposed matches found in the previous optimization run. This time limit is not imposed when computing the *a posteriori* benchmarks, but since the problems are very difficult to solve we apply the iterative rounding procedure described earlier to find a very good feasible solution; the final integer program after variable fixing is solved to a 5% optimality gap. Since the *a posteriori* benchmark problem is not solved to provable optimality in this case, we also record the solution of its linear programming relaxation to provide an upper bound on potential cost savings.

Table 4 summarizes results for the 2% participation rate announcement streams and shows that role flexibility yields substantial improvements: an absolute increase of approximately 15% on the success rate, and 10% on vehicle-miles savings. As in the earlier fixed role case, the optimization-based approach (denoted IP) performs much better than the greedy heuristic. However, the individual optimization problems are much harder and more time-consuming to solve. In our study, the integer programming software finds at least one integer feasible solution for each of the optimization runs for each of the 5 announcement streams within the 200 second time limit. In 15% of the runs, the time limit expires before the 1% optimality gap is attained; for these runs, the maximum gap observed was 2.9%. Note that again the rolling horizon aggregate solution has total quality not much smaller than the best integer solution found for the *a posteriori* benchmark problem. Furthermore, the best integer solutions found for the *a posteriori* problems are quite close to the linear programming upper bound, indicating that the benchmarks are quite good and that the iterative rounding procedure is useful for solving these very large flexible role instances.

Figure 7 demonstrates how flexible role problems solved using the optimization-based approach are able to find matches for most trips with longer distances. The figure breaks out trip announcements in distance buckets into three subsets: matched as rider, matched as driver, and not matched. We see that the longest trips have the highest success rate and the shorter trips have the smallest success rate. This is intuitive since ride-share matches between longer trips lead to greater vehicle-mile savings. As expected, a relatively larger number of the longer trip announcements are matched up as drivers. However, not all long (short) trips are drivers (riders) because in fact the ride-share matches that produce the largest savings involve participants with very similar trip lengths, often traveling from the same origin region to same destination region.

Table 3: Rolling Horizon Strategy Comparison

	S (%)	M (%)	C (%)
Latest commitment			
BIPART 1 min	68.5	22.9	21.9
BIPART 5 min	67.3	22.5	27.3
BIPART 10 min	67.0	22.3	27.3
BIPART 30 min	65.5	21.2	26.6
Immediate commitment			
BIPART 1 min	62.6	14.3	15.7
BIPART 5 min	62.4	15.6	21.2
BIPART 10 min	63.0	16.9	22.5
BIPART 30 min	64.3	19.8	25.4
* base case			

Table 4: Ride-Sharing with Flexible Roles

	S (%)	M (%)	C (%)
GREEDY	45.8	19.3	28.3
IP	85.4	31.4	30.0
a posteriori	85.6	33.6	32.1
LP-relaxation	87.0	34.3	

4.4. Single Trip Ride-Sharing

In the experiments described earlier, we assume that all trip announcements are for round trips, and that both the outbound and return trip timing are known with certainty when announced. However, for certain round trips, it may be difficult for participants to specify the time of their return trip, and they may prefer to announce both trips separately on short notice.

To understand the system impacts that result when participants attempt to arrange their trips separately, we conduct an experiment where we consider the same 5 announcement streams for the 2% participation rate, only now return trips are announced 30 minutes before their earliest departure time instead of together with the outbound trips. Drivers are assumed to always announce two trips, but riders will not announce a return trip if they did not share a ride on their outbound trip. To prevent unmatched trip requests, we also consider using a different objective function for the optimization problems solved here, maximizing the total number of system matches instead of total system travel distance savings.

For this experiment, we compute the success rate (S) by considering the percentage of *riders* that were matched for rides on both their outbound and return trip. Moreover, we compute the percentage of *riders* (S^-) that were matched outbound, but failed to be matched on their return trip. The results are presented in Table 5. Notably, for both the round-trip announcement cases (BIPART-JOINT) and the separate announcement cases (BIPART-SEP), the objective of maximizing the number of matches rather than savings can increase the matching success rate by 4 – 8% with only small degradation of the total vehicle-miles savings (< 1%) and per-match cost savings (3 – 4%).

Separate trip announcements without a return guarantee increase the vehicle-miles savings for both cases and success rate when maximizing matches. However, the additional flexibility creates a risk for each rider of failing to find a return ride-share match. Not surprisingly, maximizing the number of matches seems to mitigate this risk, *i.e.*, 5.3% of the riders without a return ride compared to 10% when savings are maximized. Furthermore, it is also possible to build optimization approaches that attempt to maximize total cost savings while prioritizing matching riders that are completing round trips; of course, the risk of not finding a match for a "stranded" rider still remains. Whether such

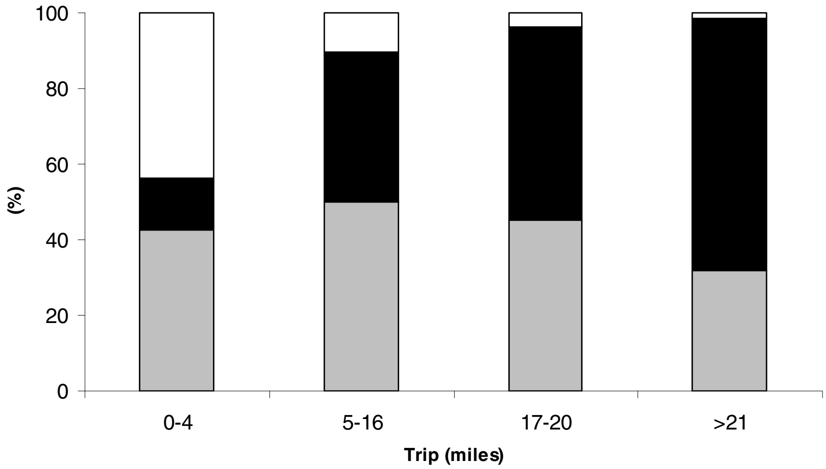

Figure 7: Matching Results for Flexible Roles Scenarios by Original Trip Length: Matched as Rider (gray), Matched as Driver (black), Not Matched (white)

Table 5: Maximize Savings versus Maximizing Matches

	S (%)	S^{-}(%)	M (%)	C (%)
maximize savings				
BIPART-JOINT	67.0	-	22.3	27.3
BIPART-SEP	65.2	10.0	24.5	29.0
maximize matches				
BIPART-JOINT	71.1	-	21.7	25.0
BIPART-SEP	73.0	5.3	23.6	25.4
* base case				

risk is acceptable depends on the situation, in particular on the availability of inexpensive alternatives such as public transport. To allow guaranteed return trips without the corresponding round trip restrictions, the ride-share provider may utilize back-up drivers, *e.g.*, by cooperating with urban commercial taxis.

4.5. Fixing Ride-share Pairs on Round Trips

Traditional carpooling typically involves a long-term commitment among at least two people to share rides to work on some or all of their weekly workdays. The lack of travel flexibility afforded by carpooling is often quoted as one of the major reasons people are hesitant to participate in carpooling [10]. Furthermore, irregular working hours also hinder traditional carpooling, since it may be more difficult to find compatible time schedules [11].

Dynamic ride-sharing is more flexible because daily trips can be arranged separately without requiring the same driver-rider pairs on different trips or on different days. To attempt to quantify some of the flexibility benefit of dynamic ride-sharing versus traditional carpooling, we consider a slightly less flexible ride-share scenario that requires a rider to be matched with the same driver on both his outbound and return trip on a specific day. Note that this scenario is more flexible than traditional carpooling, because it still allows different matches across days. We also choose to conduct this study using the assumptions of the static benchmark problem, where all trip announcements are known prior to the beginning of the day, and assume that announcements have fixed roles.

For this experiment, we will also vary the variability of participant departure times to understand its impact on the value of the flexibility of dynamic ride-sharing. To do so, we consider a set of scenarios in which we increase the standard deviation of morning departure time and the standard deviation of the workday duration both by 50%, and another set of scenarios where both deviations where we decrease these deviations by 50%.

Note that when we only consider ride-share matches in which the driver for each matched rider is the same on the outbound and return trips, we introduce symmetry to the optimization problem since the vehicle-mile savings on the outbound trip are equal to the savings on the inbound trip. This optimization problem can be represented using a maximum weight bipartite matching model with one node for each round-trip announcement, and an arc from a rider announcement i to a driver announcement j if both the outbound and return trip matches are feasible, with weight c_{ij} equal to twice the cost savings generated by the outbound match.

Table 6: Fixed Ride-share Pairs

	S (%)	M (%)	C (%)
fixed pairs	57.6	18.4	25.4
flexible pairs	68.7	23.8	27.3
time variability +50%			
fixed pairs	47.4	14.0	23.0
flexible pairs	65.7	22.1	27.2
time variability -50%			
fixed pairs	71.6	25.4	29.3
flexible pairs	77.1	28.6	30.9

Table 6 summarizes the results of this experiment, where the lines labeled “fixed pairs” assume that riders are matched both on outbound and return trip with the same driver, while the lines labeled “flexible pairs” relax this assumption (as in the earlier results). Flexible pairings substantially increase the solution quality: the success rate is increased by about 10% in absolute terms, and the cost savings by about 4-5%. As expected, the benefit of flexible pairs increases with the variability of the departure times of the participants. This is because one can always keep the same ride-share pairs on both trips if the participants spend roughly the same amount of time at work between the two trips. In the absence of any time variability, of course, the flexible and fixed pairs case would yield the same solution. Since many information economy workers no longer have rigid work schedules, the flexibility benefits provided by dynamic ride-sharing over traditional carpooling are quite important to consider.

4.6. *Varying the Participants’ Flexibility*

Ride-sharing asks for time sacrifices, especially from the drivers. In addition, participants may have to be somewhat flexible in their departure times to find a ride-share match. The individual benefits in terms of travel cost savings provided by ride-sharing should counterbalance these inconveniences. Therefore, financial gains less than a specified threshold may not be acceptable for participants. Moreover, more certainty regarding the potential savings may motivate participants to be more flexible in their departure times. In this experiment, we evaluate the impact of the participant’s time flexibility and a cost savings threshold on the performance of the system. The cost savings threshold (denoted τ) represents the minimum acceptable cost savings per feasible ride-share match.

The results are shown in Table 7 for the base case strategy and a 2% participation rate. As expected, there are system and individual benefits created by additional time flexibility. Furthermore, the marginal benefits decrease time flexibility increases. We observe that an increase in the cost savings threshold has a stronger negative impact on the success rate than on the system-wide vehicle miles savings. Surprisingly, in one scenario with a time flexibility of 10 minutes, setting a small threshold ($1) even leads to an increase in vehicle miles savings. In this case, the system appears less likely to commit to a match with very small cost savings while better matching opportunities are available at a later point in time. Overall, the results suggest that more time flexibility allows participants to set a higher cost savings threshold with limited impact on the performance of the system when measured by savings in vehicle miles.

Table 7: Participants' Time Flexibility and Cost Savings Threshold

	S (%)	M (%)	C (%)
$\tau = \$0$			
10 min	47.9	13.4	23.9
20 min	67.0	22.3	27.3
30 min	73.7	26.3	28.9
$\tau = \$1$			
10 min	39.8	13.5	22.0
20 min	56.7	22.0	24.1
30 min	62.8	25.9	25.3
$\tau = \$2$			
10 min	30.3	12.5	24.2
20 min	45.9	20.8	25.7
30 min	51.6	24.6	26.6

5. How to Achieve Critical Mass?

The experiments presented in this paper have shown the importance of sufficient numbers of announcing participants to enable dynamic ride-share matches to be established on short notice in practice. In the startup phase of a dynamic ride-share system, it may be difficult to attract enough participants to generate good matches, and this will likely lead many potential participants to give up on the system. In this section, we attempt to develop a reasonable model for an adoption pattern of dynamic ride-sharing over time, and to determine whether dynamic ride-sharing systems may be successfully initiated and sustained.

To model the adoption of dynamic ride-sharing, we draw upon the very large body of marketing literature on the diffusion of new products and technology. The most widely accepted diffusion model is the Bass diffusion model [12]. The model assumes that the probability that an initial purchase will be made is a linear function of the number of previous buyers [13]. Due to interpersonal communications (*e.g.*, word-of-mouth), potential adopters are more likely to become aware of a certain product or service if the number of users increases. The probability $\frac{k(t)}{1-K(t)}$ of adoption, *i.e.*, starts placing announcements, at time t is $p + \frac{q}{m}Y(t)$, with $k(t)$ representing the individual probability of adoption at time t and $K(t)$ its cumulative form and $Y(t)$ the total number of adopters up to time t. The constant parameters m, p, and q represent the total number of potential adopters, a coefficient of innovation and a coefficient of imitation respectively. While the coefficient of innovation represents the exogenous likelihood that a new participant joins the system, the coefficient of imitation relates to the increase in this likelihood with the number of participants that are already in the system.

While the diffusion model allows us to forecast how many new participants join the system, we also want to consider the announcement behavior of the existing participants over time. Conceptually, we may assume that participants are satisfied if they are matched in ride-shares, and thus continue to announce trips regularly. Participants that do not receive ride-share matches may become discouraged and stop announcing new trips. To model this behavior, we assume that a participant i receives one additional positive goodwill credit s_i from each successful ride-share match, and one negative credit f_i each time a trip is announced and is not matched. As long as his net credit is positive ($s_i - f_i + g > 0$), we assume that the participant will continue to announce his ride-share trips, where we define g to be the starting goodwill credit of the participant. Once goodwill is depleted to zero, the participant never announces again. We recognize that this set of assumptions creates a system that, if simulated over very long time horizons, will eventually include no possible participants. However, we believe that the model is useful for examining system behavior over relatively short time periods. To examine longer time periods, it would not be difficult to extend the model to allow new potential adopters to enter the system over time, for example, representing new members of the labor force entering the pool of commuters.

In the following experiments, we follow the behavior of a hypothetical system for Atlanta over a two month period after startup for different diffusion parameters. Each day in the study period includes a set of round-trip announcements with fixed roles, and is solved using the rolling horizon optimization approach. Unless stated otherwise, we assume the total number of potential trip announcements (m) to be 4% of the total number of home-based trips and a goodwill g of 5. First, we determine a set of potential participant round-trips using the methods described earlier. For each potential participant, we draw a base latest departure time from a normal distribution with a mean of 7:30 a.m. and a standard deviation of 1 hour (see Section 4). For each subsequent day, we draw the latest departure of each active participant again using a normal distribution with his base departure time as the mean and 15 minutes as the standard deviation.

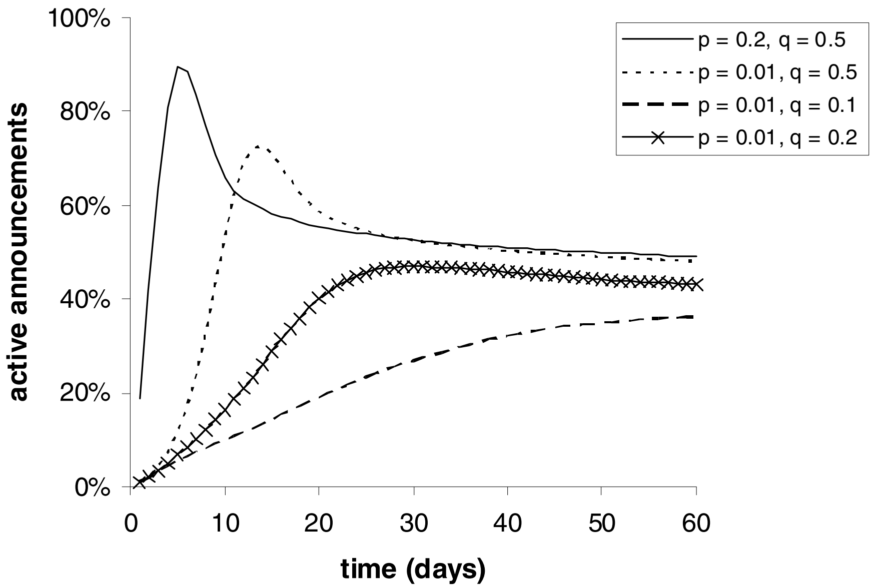

Figure 8: Ride-Sharing System Sustainability for Various Diffusion Patterns

The results of these experiments are summarized in Figure 8, where the fraction of active participant announcements is plotted over time. The plots demonstrate that when the sum of the innovation and imitation rates is sufficiently high (*i.e.*, > 0.5), the system seems to converge to a steady active announcement stream in two to three weeks. Approximately 55% of the total potential trip announcements remain active, and the success rate converges to approximately 85% of announced trips. The results show that even when the total potential pool of participants is limited to a small fraction (4%) of the total home-based work round-trips, dynamic ride-sharing may still be sustainable. Participants in corridors amenable to ride-sharing will likely continue to announce given the high match rate; in this way, the ride-sharing system at least in this experiment has configured itself.

In Figure 9, we see that the initial goodwill possessed by potential participants has a significant impact on the success and sustainability of dynamic ride-sharing systems. It seems particularly important in the startup phase that potential participants continue to place announcements even though they are not matched. It seems highly likely, therefore, that public incentives might be necessary to initiate a dynamic ride-sharing system. If participants are discouraged by not finding matches when the participant density is low, it may be quite difficult to build a sustainable

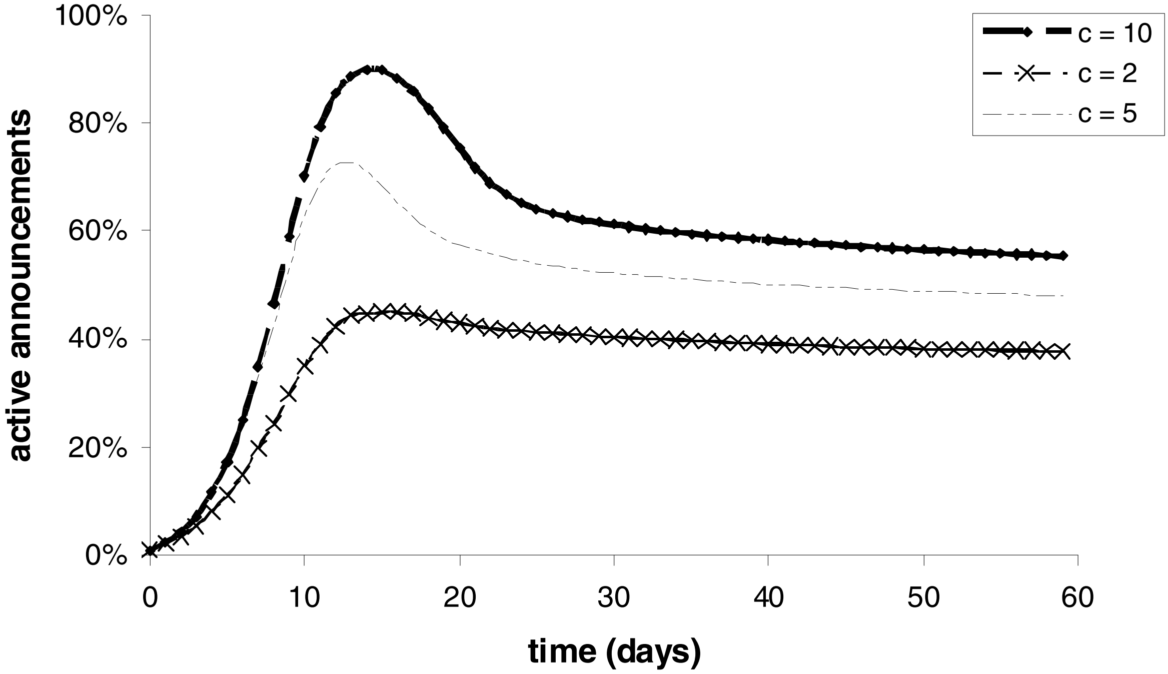

Figure 9: Sustainability of Ride-sharing Systems for Different Levels of Initial Participant Goodwill

participant community.

6. Concluding Remarks

Internet-enabled mobile technology allows car travelers to announce trip requests and ride offers on short-notice. In this study of dynamic ride-sharing, we have seen that the use of sophisticated optimization methods substantially increases the likelihood that ride-share matches can be found for participants, and leads to ride-sharing systems that generate larger overall system travel cost savings. Furthermore, our simulation studies have shown that dynamic ride-sharing may have potential for success in large U.S. metropolitan areas, with sustainable ride-share populations forming over time even with relatively small overall participation rates and when considering only home-based work trips.

Besides travel costs savings, ride-sharing systems may provide travel time savings to participants by providing access to high occupancy lanes. Moreover, ride-sharing may help to decrease traffic congestion and thereby reduce system-wide travel times. We believe that extending ride-sharing simulation models to explicitly consider time-dependent and occupancy-dependent travel times provides a valuable area of future research.

References

[1] N. Agatz, A. Erera, M. Savelsbergh, X. Wang, Sustainable passenger transportation: Dynamic ride-sharing, Tech. rep., Rotterdam School of Management, Erasmus University (2010).

[2] K. Kelley, Casual carpooling enhanced, Journal of Public Transportation 10 (4) (2007) 119–130.

[3] R. Buliung, K. Soltys, C. Habel, R. Lanyon, Driving factors behind successful carpool formation and use, Transportation Research Record 2118 (2009) 31–38.
[4] R. H. M. Emmerink, P. vanBeek, Empirical analysis of work schedule flexibility: Implications for road pricing and driver information systems, Urban Studies 34 (2) (1997) 217–234.
[5] P. Jensen, Indicator: Occupancy rates of passenger vehicles, Tech. rep., European Environmental Agency (2005).
[6] J. Edmonds, E. Johnson, Matching: A well-solved class of integer linear programs, in: M. Jünger, G. Reinelt, G. Rinaldi (Eds.), Combinatorial Optimization - Eureka, You Shrink!, Vol. 2570 of Lecture Notes in Computer Science, Springer, 2003, pp. 27–30.
[7] N. McGuckin, N. Srinivasan, Journey to work trends in the United States and its major metropolitan areas 1960 - 2000, Tech. rep., US Department of Transportation Federal Highway Administration (2003).
[8] American Automobile Association, Your driving costs: how much are you really paying to drive? (2009).
[9] J. Li, P. Embry, S. P. Mattingly, K. F. Sadabadi, I. Rasmidatta, M. W. Burris, Who chooses to carpool and why? Examination of Texas carpoolers, Transportation Research Record 2021 (2007) 110–117.
[10] H. Tsao, D.-J. Lin, Spatial and temporal factors in estimating the potential of ride-sharing for demand reduction, Tech. rep., Institute of Transportation Studies, University of California, Berkeley (1999).
[11] E. Ferguson, The influence of employer ridesharing programs on employee mode choice, Transportation 17 (2) (1990) 179–207.
[12] V. Mahajan, E. Muller, F. Bass, Diffusion of new products: empirical generalizations and managerial uses, Marketing Science 14 (3) (1995) G79–G88.
[13] F. Bass, A new product growth model for consumer durables, Management Science 15 (5) (1969) 215–227.