Community-Based Trip Sharing for Urban Commuting

Mohd. Hafiz Hasan, Pascal Van Hentenryck, Ceren Budak, Jiayu Chen, and **Chhavi Chaudhry**
University of Michigan
Ann Arbor, Michigan 48109

Abstract

This paper explores Community-Based Trip Sharing which uses the structure of communities and commuting patterns to optimize car or ride sharing for urban communities. It introduces the Commuting Trip Sharing Problem (CTSP) and proposes an optimization approach to maximize trip sharing. The optimization method, which exploits trip clustering, shareability graphs, and mixed-integer programming, is applied to a dataset of 9000 daily commuting trips from a mid-size city. Experimental results show that community-based trip sharing reduces daily car usage by up to 44%, thus producing significant environmental and traffic benefits and reducing parking pressure. The results also indicate that daily flexibility in pairing cars and passengers has significant impact on the benefits of the approach, revealing new insights on commuting patterns and trip sharing.

1 Introduction

Carpooling services provide an appealing alternative for urban mobility due to their potential benefits, be it in reducing traffic congestion, energy consumption, greenhouse gas emissions, or parking utilization. For instance, a case study on the CarLink carpooling program of about 50 people revealed up to 43.5% reduction in the number of single occupant vehicle trips, a 23 miles reduction in average commute vehicle travel distance per day, and reduced parking utilization (Shaheen and Rodier 2005). Private cars however have remained as the primary choice for daily commuting due to a number of challenges associated with carpooling. For instance, a survey by (Li et al. 2007) indicated difficulty in finding people with matching schedules and locations as the primary reason for not carpooling. This highlights the potential for matching platforms which alleviate this burden and automatically identify commuting groups based on factors determined to be consequential to individuals' commuting decisions. A meta-analysis of related work reveals the following set of guiding *principles* that should ideally be supported by car-pooling and car-sharing platforms:

1. Spatial proximity of riders (Richardson and Young 1981; Buliung et al. 2009);
2. Temporal proximity of riders (Tsao and Lin 1999; Buliung et al. 2010; Poulenez-Donovan and Ulberg 1994);
3. Guaranteed ride back home (Correia and Viegas 2011);
4. Low coordination costs (Arning, Ziefle, and Muehlhans 2013);
5. Low trust concerns (Arning, Ziefle, and Muehlhans 2013; Correia and Viegas 2011);
6. Clear commuter roles (Buliung et al. 2010; Richardson and Young 1981).

The first two principles reduce the per-trip costs by matching commuters based on their schedules and locations. The third principle highlights the importance of accounting for the commute needs for the *entire* day–individuals who cannot be matched for the return trip should not be matched for the incoming one. Principles (4–6) account for various psychological factors by limiting the perceived coordination costs, by alleviating trust concerns, and by assigning clear commuter roles to individuals.

To address these challenges, this paper explores the concept of *community-based trip sharing* which uses the structure of communities and commuting patterns to optimize trip sharing for urban communities. Community-based trip sharing identifies matches according to the schedules and locations of riders and guarantees a ride home and hence it satisfies guiding Principles (1–3) by construction. The implementation of community-based trip sharing first clusters commuters by communities before applying an optimization model to determine optimal trip-sharing solutions minimizing daily car usage. Community-based trip sharing can be applied both to car pooling, where commuters use their own cars, and to car sharing, where a community has at its disposal a pool of cars for commuting purposes.

This paper also studies the cost of implementing Principles (4–6). The implementation of each of these principles reduces the opportunities for trip sharing and the tradeoff between the effectiveness of a trip-sharing platform and these guiding principles is largely unexplored. To provide new insights on this issue, the paper proposes a series of optimization models for community-based trip sharing that incrementally enforce additional constraints to implement these principles. For instance, Principle (6) forces a given commuter to be either a driver or a passenger in all her trips, which may minimize opportunities for trip sharing as her schedule may vary on different days.

This paper evaluates the potential and limitations of community-based trip sharing on a large case study using a dataset containing trip data from 15,000 commuters working downtown in the city of Ann Arbor (Michigan) over the span of a month. Ann Arbor is facing significant pressure on its downtown parking lots and congestion has been increasing annually. The results indicate that community-based trip sharing may reduce daily car usage by as much as 44%, while implementing Principles (1–3). However, the benefits continuously decrease as Principles (4–6) are implemented, up to a point where they become negligible. This highlights the trade-off between the effectiveness of trip sharing and the (psychological) comfort of commuters.

The main contributions of this paper are as follows:

1. Community-based trip sharing is introduced and applied to both car pooling and car sharing.
2. An effective implementation of community-based trip sharing is proposed, which combines hierarchical clustering and optimization to minimize daily car usage.
3. Community-based trip sharing is evaluated with the first large-scale, high-fidelity study of car pooling and car sharing for commuting purposes.
4. The study provides compelling quantitative evidence for the inherent trade-off between the benefits of trip sharing and the psychological burden imposed on commuters.

2 Additional Related Work

Ride sharing has been widely studied in the literature, and some implemented in the real world. (Alexander and González 2015) found that ride-sharing services would have a noticeable impact on congested travel time, and (Handke and Jonuschat 2013) showed in a survey that 45% of respondents were interested in ride sharing. Many current studies are dedicated to the ride sharing of private vehicles. (He et al. 2012) designed a route-mining algorithm that leverages frequent user routes to provide ride-sharing recommendations. (Trasarti et al. 2011) used GPS traces to build mobility profiles and match users with similar profiles. (Bellemans et al. 2012) designed a multi-agent based model to provide online matching for those living and working in close areas. More recently, (Xia et al. 2015) developed optimal and heuristic approaches for a carpool matching service and applied them on a real-world transportation network combined with randomly generated trip data. Contrary to prior work, this paper provides high-fidelity evaluation of the potential of trip sharing based on a large-scale, real-world dataset and detailed optimization models that impose various matching constraints on the pooling platform.

Other related studies focused on ride sharing of public vehicles. For example, a research on taxis in New York City by (Santi et al. 2014) finds that ride sharing using a shareability graph could reduce trip duration by 40% with low level of discomfort. This research is further developed by (Alonso-Mora et al. 2017) using a reactive anytime optimal method that allows 3000 vehicles to serve 98% of the trip requests originally served by 14,000 taxis with minimal discomfort. In another study, (Zhu et al. 2016) designed a mixed-integer programming (MIP) algorithm for dynamic ride sharing that results in 90% reduction of cars used in the conventional vehicle system and 57% reduction of cars used in Uber Pool. Although the algorithms presented in this paper share some concepts with those for shared public vehicles, they differ in their focus on commuting and the constraints imposed by pooling services, which leads to rather different models.

3 Notation and Preliminaries

A trip $t =< o, st, d, at >$ consists of an origin o, a start time st, a destination d, and an arrival time at. On day δ, commuter c makes two trips: a trip to the workplace (an inbound trip) $t_{c,i,\delta}$ and a trip back home (an outbound trip) $t_{c,o,\delta}$. A roundtrip $t_{c,rt,\delta} = (t_{c,i,\delta}, t_{c,o,\delta})$ is the pair of inbound and outbound trips taken by commuter c on day δ.

A trip-sharing route $r_{\mathcal{T}}$ is a sequence of origin and destination locations from a set of trips $\mathcal{T}$ in which each origin and destination from the set is visited exactly once. For instance, given two trips $t_1 =< o_1, st_1, d_1, at_1 >$ and $t_2 =< o_2, st_2, d_2, at_2 >$, a possible trip-sharing route is $r_{\{t_1,t_2\}} = o_2 \rightarrow o_1 \rightarrow d_1 \rightarrow d_2$. Each route r has a set of commuters $C(r)$ and a designated driver $D(r) \in C(r)$. The driver must be the commuter residing at the start location of the route. For instance, commuter 2 must be the driver for route $r_{\{t_1,t_2\}}$ shown earlier.

Definition 3.1 (Valid Trip-Sharing Route)**.** A valid trip-sharing route r visits o_c before d_c for every commuter $c \in C(r)$ and starts at $o_{D(r)}$ and ends at $d_{D(r)}$.

A feasible trip-sharing route is a valid trip-sharing route that can pickup and drop-off its commuters at their respective origins and destinations within a given time window Δ. The rationale is that commuters may be willing to shift their pickup and drop-off times by at most $\pm\frac{\Delta}{2}$.

Definition 3.2 (Feasible Trip-Sharing Route)**.** A feasible trip-sharing route r is a valid trip-sharing route that picks up and drops off its commuters at their respective origins and destinations such that

$$pt_c \in [st_c - \frac{\Delta}{2}, st_c + \frac{\Delta}{2}] \wedge dt_c \in [at_c - \frac{\Delta}{2}, at_c + \frac{\Delta}{2}]$$

for each commuter $c \in C(r)$.

For example, route $r_{\{t_1,t_2\}}$ shown earlier is feasible if there exist a pickup time pt_c at o_c and a drop-off time dt_c at d_c for $c \in \{1, 2\}$ that satisfy the following constraints:

$$pt_1 \in [st_1 - \frac{\Delta}{2}, st_1 + \frac{\Delta}{2}] \wedge pt_2 \in [st_2 - \frac{\Delta}{2}, st_2 + \frac{\Delta}{2}] \quad (1)$$

$$dt_1 \in [at_1 - \frac{\Delta}{2}, at_1 + \frac{\Delta}{2}] \wedge dt_2 \in [at_2 - \frac{\Delta}{2}, at_2 + \frac{\Delta}{2}] \quad (2)$$

For route $r_{\{t_1,t_2\}}$, pt_1, dt_1, and dt_2 can be represented in terms of pt_2 using the following relations:

$$pt_1 = pt_2 + tt(o_2, o_1) \quad (3)$$

$$dt_1 = pt_2 + tt(o_2, o_1) + tt(o_1, d_1) \quad (4)$$

$$dt_2 = pt_2 + tt(o_2, o_1) + tt(o_1, d_1) + tt(d_1, d_2) \quad (5)$$

where $tt(x, y)$ is the estimated travel time for the shortest path between locations x and y. Therefore, the route is feasible if there exists a pickup time pt_2 that satisfies (1)–(5).

Definition 3.3 (Feasible Roundtrip-Sharing Route)**.** Let $r_{\mathcal{T}_i}$ and $r_{\mathcal{T}_o}$ denote feasible trip-sharing routes for a set of inbound trips $\mathcal{T}_i$ and a set of outbound trips $\mathcal{T}_o$ respectively. A feasible roundtrip-sharing route $r_{\mathcal{T}_{rt}} = (r_{\mathcal{T}_i}, r_{\mathcal{T}_o})$ is a pair of feasible inbound and outbound trip-sharing routes serving the same set of commuters, i.e., $C(r_{\mathcal{T}_i}) = C(r_{\mathcal{T}_o})$, and having the same driver, $D(r_{\mathcal{T}_i}) = D(r_{\mathcal{T}_o})$.

Given a set of commuters $\mathcal{C}$, the sets of all feasible trip-sharing routes for inbound, outbound, and round trips taken by $\mathcal{C}$ on day δ are denoted by $\mathcal{R}_{i,\delta}$, $\mathcal{R}_{o,\delta}$, and $\mathcal{R}_{rt,\delta}$, and the set of days under consideration is denoted by $\mathcal{D}$. The algorithms in this paper solve the Commuting Trip Sharing Problem (CTSP) that minimizes the number of cars needed daily to cover all commuting trips of $\mathcal{C}$ for all days $\delta \in \mathcal{D}$ subject to specific commuter-matching constraints.

4 Community-Based Trip-Sharing

The community-based trip-sharing algorithm solves (versions of the) CTSP using as input a dataset containing daily roundtrips of commuters from an urban population. It proceeds in three major stages: (1) It clusters commuters based on their home locations; (2) it identifies all feasible trip-sharing routes using shareability graphs; and (3) it solves an optimization model to obtain an optimal trip-sharing assignment. This section focuses on steps (1–2). The next two sections present the optimization models.

Clustering Community-based trip sharing clusters commuters residing in close proximity to each other, implementing Principle (1) from the introduction. Trip sharing is only considered intra-cluster to foster intra-community interactions and limit the distance traveled by drivers when picking up or dropping off passengers. As a side-effect, community-based trip sharing keeps the CTSP tractable by breaking it down into many smaller subproblems.

The clustering algorithm imposes a limit on the diameter of each cluster, where the diameter is defined as the maximum distance between any two points in a cluster. The clustering algorithm is *hierarchical* and represents commuters as points in 2D Euclidean space using the Cartesian coordinates of their homes. The algorithm begins by treating each point as its own cluster. A pair of clusters with the shortest inter-cluster distance is then selected and merged. Inter-cluster distance is measured by taking the largest distance between points in the two clusters. Pairwise cluster selection and merging is repeated until further merging causes the diameter of the largest cluster σ_{max} to exceed a limit σ_{limit}, at which point the algorithm is terminated. This algorithm permits distance-based control of the size of all clusters, which do not exceed σ_{limit}.

Shareability Graph After the clustering step, the algorithm computes the sets $\mathcal{R}_{i,\delta}$, $\mathcal{R}_{o,\delta}$, and $\mathcal{R}_{rt,\delta}$ for each day $\delta \in \mathcal{D}$ and each cluster, using the concept of shareability graphs (Santi et al. 2014). Without loss of generality, the presentation focuses on a single cluster. A shareability graph $\mathcal{G} = (\mathcal{T}, \mathcal{E})$ is an undirected graph with nodes $\mathcal{T}$ consisting of trips and edges $\mathcal{E}$ representing pairwise shareable trips.

Algorithm 1 Shareability Graph for Inbound Trips

Require: $\mathcal{T}_{i,\delta} = \{t_{c,i,\delta} \mid c \in \mathcal{C}\}$
1: **for** each $x \in \mathcal{C}$ **do**
2:   **for** each $y \in \mathcal{C}$ **do**
3:     $\mathcal{R}_{\text{temp}} \leftarrow$ {all feasible trip sharing routes for $(t_{x,i,\delta}, t_{y,i,\delta})$}
4:     **for** each $r \in \mathcal{R}_{\text{temp}}$ **do**
5:       **if** $ttr(r) \geq 0$ **then**
6:         $\mathcal{R}_{\{t_{x,i,\delta}, t_{y,i,\delta}\}} \leftarrow \mathcal{R}_{\{t_{x,i,\delta}, t_{y,i,\delta}\}} \cup \{r\}$
7:     **if** $\mathcal{R}_{\{t_{x,i,\delta}, t_{y,i,\delta}\}} \neq \emptyset$ **then**
8:       Store $\mathcal{R}_{\{t_{x,i,\delta}, t_{y,i,\delta}\}}$ in edge $(t_{x,i,\delta}, t_{y,i,\delta})$
9:       $\mathcal{E} \leftarrow \mathcal{E} \cup \{(t_{x,i,\delta}, t_{y,i,\delta})\}$
10: **return** $\mathcal{G}_{i,\delta} = (\mathcal{T}_{i,\delta}, \mathcal{E})$

For a specified time window Δ, each edge $(t_x, t_y) \in \mathcal{E}$ contains a set $\mathcal{R}_{\{t_x, t_y\}}$ of feasible trip-sharing routes between trips t_x and t_y. This work uses three types of shareability graphs: $\mathcal{G}_{i,\delta}$, $\mathcal{G}_{o,\delta}$, and $\mathcal{G}_{rt,\delta}$ respectively denote shareability graphs for inbound, outbound, and round trips. More precisely, let $\mathcal{T}_{i,\delta}$, $\mathcal{T}_{o,\delta}$, and $\mathcal{T}_{rt,\delta}$ denote the set of all inbound, outbound, and round trips taken by $\mathcal{C}$ on day δ, i.e., $\mathcal{T}_{i,\delta} = \{t_{c,i,\delta} \mid c \in \mathcal{C}\}$, $\mathcal{T}_{o,\delta} = \{t_{c,o,\delta} \mid c \in \mathcal{C}\}$, and $\mathcal{T}_{rt,\delta} = \{t_{c,rt,\delta} \mid c \in \mathcal{C}\}$. The graphs $\mathcal{G}_{i,\delta}$, $\mathcal{G}_{o,\delta}$, and $\mathcal{G}_{rt,\delta}$ are then constructed from $\mathcal{T}_{i,\delta}$, $\mathcal{T}_{o,\delta}$, and $\mathcal{T}_{rt,\delta}$ respectively.

For example, $\mathcal{G}_{i,\delta}$ is constructed by first introducing a node to represent each trip in $\mathcal{T}_{i,\delta}$. A shareability check is then performed on every pair of trips in $\mathcal{T}_{i,\delta}$, i.e., for each pair in $\{(t_{x,i,\delta}, t_{y,i,\delta}) \mid x \in \mathcal{C}, y \in \mathcal{C}\}$. The check first searches for all feasible trip-sharing routes for the pair being considered, e.g., by enumerating all valid route permutations and checking if they satisfy constraints (1)–(5). Should feasible routes exist, the algorithm only considers those with a non-negative travel time reduction, where the travel time reduction $ttr(r)$ of a route r is the difference between the time of the trip sharing route and the total duration of the individual (unshared) trips. The rationale behind this restriction is to consider trip sharing only if its route time is at least as good as the total duration of the individual trips. Edges store the feasible routes between pairs of trips satisfying the shareability check. The algorithm for constructing $\mathcal{G}_{i,\delta}$ is summarized in Algorithm 1.

Let $\mathcal{R}_{k,i,\delta}$ denote the set of all feasible inbound trip-sharing routes serving k commuters on day δ. $\mathcal{R}_{1,i,\delta}$ is simply given by the routes of all individual trips represented by the nodes of $\mathcal{G}_{i,\delta}$, while $\mathcal{R}_{2,i,\delta}$ is obtained by taking the union of all feasible routes stored in the edges of $\mathcal{G}_{i,\delta}$. To obtain $\mathcal{R}_{k,i,\delta}$ for $k > 2$, all k-cliques in $\mathcal{G}_{i,\delta}$ are first identified. A k-clique is the subset of k nodes of the graph such that every two nodes in the set are connected. Let $Q_{k,i,\delta}$ denote the set of all k-cliques in $\mathcal{G}_{i,\delta}$. For each clique $q \in Q_{k,i,\delta}$, the algorithm searches for all feasible trip-sharing routes for the set of trips in q with non-negative travel time reduction and stores them in $\mathcal{R}_{k,i,\delta}$. Should there exist multiple feasible routes for q, the algorithm only stores the fastest route for each driver. Since this work focuses on small vehicles, it only considers sharing for up to 4 commuters and hence

Algorithm 2 All Feasible Inbound Trip Sharing Routes

Require: $\mathcal{G}_{i,\delta} = (\mathcal{T}_{i,\delta}, \mathcal{E})$
for each $t_{c,i,\delta} \in \mathcal{T}_{i,\delta}$ **do**
  $\mathcal{R}_{1,i,\delta} \leftarrow \mathcal{R}_{1,i,\delta} \cup \mathcal{R}_{\{t_{c,i,\delta}\}}$
for each $(t_{x,i,\delta}, t_{y,i,\delta}) \in \mathcal{E}$ **do**
  $\mathcal{R}_{2,i,\delta} \leftarrow \mathcal{R}_{2,i,\delta} \cup \mathcal{R}_{\{t_{x,i,\delta}, t_{y,i,\delta}\}}$
for each $k \in \{3, 4\}$ **do**
  $Q_{k,i,\delta} \leftarrow$ {all k-cliques in $\mathcal{G}_{i,\delta}$}
  for each $q \in Q_{k,i,\delta}$ **do**
    $\mathcal{R}_{\text{temp}} \leftarrow$ {all feasible trip sharing routes for q}
    for each $r \in \mathcal{R}_{\text{temp}}$ **do**
      if $ttr(r) \geq 0$ **then**
        $\mathcal{R}_{k,i,\delta} \leftarrow \mathcal{R}_{k,i,\delta} \cup \{r\}$
$\mathcal{R}_{i,\delta} \leftarrow \mathcal{R}_{1,i,\delta} \cup \mathcal{R}_{2,i,\delta} \cup \mathcal{R}_{3,i,\delta} \cup \mathcal{R}_{4,i,\delta}$
return $\mathcal{R}_{i,\delta}$

$\mathcal{R}_{i,\delta}$ is obtained from $\mathcal{R}_{1,i,\delta} \cup \mathcal{R}_{2,i,\delta} \cup \mathcal{R}_{3,i,\delta} \cup \mathcal{R}_{4,i,\delta}$. However, the algorithm can be extended to vehicles with arbitrary capacity. The algorithm for obtaining $\mathcal{R}_{i,\delta}$ is summarized in Algorithm 2. The same procedure of shareability-graph construction and identification of all feasible trip-sharing routes is repeated on $\mathcal{T}_{o,\delta}$ and $\mathcal{T}_{rt,\delta}$ to obtain $\mathcal{R}_{o,\delta}$, and $\mathcal{R}_{rt,\delta}$.

Global Shareability with Travel Distance Constraint While the clustering approach significantly improves the tractability of the approach, it may preclude trip sharing across cluster boundaries for commuters who live within a short distance. An alternative to the clustering approach, that still enforces close-proximity trip sharing, amounts to building global shareability graphs whose edges must have travel distances less than σ_{limit}. More formally, a route is feasible in the global shareability graphs $\mathcal{G}_{i,\delta}$, $\mathcal{G}_{o,\delta}$, and $\mathcal{G}_{rt,\delta}$ if it satisfies (1)–(5) and the additional constraint:

$$td(o_2, o_1) \leq \sigma_{\text{limit}} \wedge td(d_1, d_2) \leq \sigma_{\text{limit}} \tag{6}$$

where $td(x, y)$ is the travel distance for the shortest path between locations x and y. $\mathcal{R}_{i,\delta}$, $\mathcal{R}_{o,\delta}$, and $\mathcal{R}_{rt,\delta}$ can then be obtained from the global shareability graphs using the algorithm described in the previous section.

5 Optimization Models for Ride Sharing

This section presents optimization models for finding optimal ride-sharing assignments for commuters for each cluster and every day $\delta \in \mathcal{D}$. The models utilize the trip-sharing routes from the shareability graphs. The names, high-level constraints, and desirable properties of each model are summarized in Table 1.

MIP-DD MIP-DD is the least-constrained optimization model for ride sharing and satisfies Principles (1–3) from the introduction. It minimizes the number of cars required subject to the constraint that drivers are the same for the inbound and outbound routes. This ensures the cars leaving a cluster returns to the cluster every day. The model optimizes ride-sharing assignments for each day $\delta \in \mathcal{D}$ independently. As a result, drivers selected for different days do not need to be the same. Passengers also do not need to be paired with the same driver for inbound and outbound routes for the same day or for different days.

The model is defined in terms of two binary variables: variable x_r indicates whether trip sharing route $r \in \mathcal{R}_{i,\delta} \cup \mathcal{R}_{o,\delta}$ is selected for the optimal assignment and variable y_c specifies whether commuter $c \in \mathcal{C}$ is selected as the driver for a pair of inbound and outbound routes. In the following formulation, $\mathcal{R} \sim c$ denotes all routes from set $\mathcal{R}$ serving commuter c, i.e., $\mathcal{R} \sim c = \{r \in \mathcal{R} \mid c \in C(r)\}$ and $P(r)$ denotes the passengers of route r, i.e., $P(r) = C(r) \setminus \{D(r)\}$. The model for day δ is specified as follows:

$$\min \sum_{r \in \mathcal{R}_{i,\delta} \cup \mathcal{R}_{o,\delta}} x_r \tag{7}$$

subject to

$$\sum_{r \in (\mathcal{R}_{i,\delta} \sim c)} x_r = 1 \qquad \forall c \in \mathcal{C} \tag{8}$$

$$\sum_{\hat{r} \in (\mathcal{R}_{o,\delta} \sim c)} x_{\hat{r}} = 1 \qquad \forall c \in \mathcal{C} \tag{9}$$

$$y_{D(r)} \geq x_r \qquad \forall r \in \mathcal{R}_{i,\delta} \cup \mathcal{R}_{o,\delta} \tag{10}$$

$$y_c \leq 1 - x_r \qquad \forall r \in \mathcal{R}_{i,\delta} \cup \mathcal{R}_{o,\delta}, \forall c \in P(r) \tag{11}$$

$$x_r \in \{0, 1\} \qquad \forall r \in \mathcal{R}_{i,\delta} \cup \mathcal{R}_{o,\delta} \tag{12}$$

$$y_c \in \{0, 1\} \qquad \forall c \in \mathcal{C} \tag{13}$$

Objective function (7) minimizes the number of cars for inbound and outbound routes. Constraints (8) and (9) indicate that exactly one inbound and one outbound route must be selected for each commuter respectively. Constraints (10) assign drivers of selected routes, while constraints (11) prevent passengers of selected routes from being selected as drivers.

MIP-DD-DIO MIP-DD-DIO contains an additional requirement compared to MIP-DD. It requires that commuters for inbound and outbound routes must be the same. This constraint reduces coordination costs and alleviates trust concerns by reducing the maximum unique matches per commuter from 2 to 1 per day. Hence the model can be considered to satisfy Principles (1–5), although it does so partially. To satisfy this constraint, the model utilizes $\mathcal{R}_{rt,\delta}$ since a roundtrip-sharing route already ensures commuters of its inbound route are the same as those of its outbound route. The model uses a single binary variable x_r to indicate whether roundtrip route $r \in \mathcal{R}_{rt,\delta}$ is selected in the optimal assignment. The model for day δ is specified as follows:

$$\min \sum_{r \in \mathcal{R}_{rt,\delta}} x_r \tag{14}$$

subject to

$$\sum_{r \in (\mathcal{R}_{rt,\delta} \sim c)} x_r = 1 \qquad \forall c \in \mathcal{C} \tag{15}$$

$$x_r \in \{0, 1\} \qquad \forall r \in \mathcal{R}_{rt,\delta} \tag{16}$$

Objective function (14) minimizes the number of cars used for roundtrips and constraints (15) state that exactly one roundtrip route must be selected for each commuter.

Application	Name	Constraints	Principles
Ride sharing	MIP-DD	• Drivers of inbound and outbound routes consistent on any given day	(1–3)
	MIP-DD-DIO	• Commuters of inbound and outbound routes consistent on any given day	(1–5)
	MIP-WD-DIO	• Commuters of inbound and outbound routes the same on any given day • Drivers are consistent every day	(1–6)
	MIP-WD-WIO	• Commuters of inbound and outbound routes consistent on any given day • Drivers are consistent every day • Passenger-driver pairings are consistent every day	(1–6)
Car sharing	MIP-DC	• Total number of inbound and outbound routes consistent on any given day	(1–3)

Table 1: Summary of Optimization Models for Trip Sharing.

MIP-WD-DIO MIP-WD-DIO has the same objective and constraints as MIP-DD-DIO, with an additional constraint that drivers for every day $\delta \in \mathcal{D}$ must be consistent. In other words, a commuter is prohibited from being a driver on some days and a passenger on others. This model satisfies Principles (1–6), since now drivers and passengers have a clearly identified role. The model uses two binary variables: variable x_r is the same as in MIP-DD-DIO and variable y_c to indicate whether commuter $c \in \mathcal{C}$ is selected as the driver for a roundtrip route. The model is specified as follows:

$$\min \sum_{\delta \in \mathcal{D}} \sum_{r \in \mathcal{R}_{rt,\delta}} x_r \tag{17}$$

subject to

$$\sum_{r \in (\mathcal{R}_{rt,\delta} \sim c)} x_r = 1 \qquad \forall \delta \in \mathcal{D}, \forall c \in \mathcal{C} \tag{18}$$

$$y_{D(r)} \geq x_r \qquad \forall \delta \in \mathcal{D}, \forall r \in \mathcal{R}_{rt,\delta} \tag{19}$$

$$y_c \leq 1 - x_r \qquad \forall \delta \in \mathcal{D}, \forall r \in \mathcal{R}_{rt,\delta}, \forall c \in P(r) \tag{20}$$

$$x_r \in \{0,1\} \qquad \forall \delta \in \mathcal{D}, \forall r \in \mathcal{R}_{rt,\delta} \tag{21}$$

$$y_c \in \{0,1\} \qquad \forall c \in \mathcal{C} \tag{22}$$

Objective function (17) globally minimizes the number of cars for every day $\delta \in \mathcal{D}$. Constraints (18) ensure exactly one roundtrip route is selected for each commuter every day, constraints (19) assign drivers of selected roundtrip routes, and constraints (20) ensure passengers of selected routes are never assigned as drivers. The differences with MIP-DD-DIO are quite subtle when formalized: The key is to recognize that the universal quantification in Constraints (19) and (20) forces a driver to drive every day and a passenger to never drive. Model MIP-DD-DIO in contrast is optimized once for each day.

MIP-WD-WIO MIP-WD-WIO adds a final additional constraint that passenger-driver pairings for every day $\delta \in \mathcal{D}$ must be consistent, i.e., a passenger always commutes with the same driver. This is the most desirable model and it strongly obeys all principles. Let $\mathcal{R}_{rt}$ denote the set of all feasible roundtrip routes across all days, i.e. $\mathcal{R}_{rt} = \{r \in \mathcal{R}_{rt,\delta} \mid \delta \in \mathcal{D}\}$, and $\mathcal{W}$ denote the set of all passenger-driver pairs obtained from all feasible roundtrip routes, i.e., $\mathcal{W} = \{(c, D(r)) \mid c \in P(r), r \in \mathcal{R}_{rt}\}$. The model uses three binary variables: x_r and y_c are the same as those used in MIP-WD-DIO, and v_w keeps track of each passenger-driver pair $w \in \mathcal{W}$ selected in the optimal assignment. Let $\Gamma(c)$ denote the set of all routes where c is a passenger, i.e., $\Gamma(c) = \{r \in \mathcal{R}_{rt} \mid c \in P(r)\}$, and $\Lambda(c)$ denote the set of all possible drivers for passenger c, i.e., $\Lambda(c) = \{D(r) \mid r \in \Gamma(c)\}$. The objective function of the model is given by (17), subject to (18), (19), (20), (21), (22), and

$$v_{(c,D(r))} \geq x_r \qquad \forall \delta \in \mathcal{D}, \forall r \in \mathcal{R}_{rt,\delta}, \forall c \in P(r) \tag{23}$$

$$v_{(c,p)} \leq 1 - x_r \qquad \forall \delta \in \mathcal{D}, \forall r \in \mathcal{R}_{rt,\delta}, \forall c \in P(r), \forall p \in \Lambda(c) \setminus \{D(r)\} \tag{24}$$

$$v_w \in \{0,1\} \qquad \forall w \in \mathcal{W} \tag{25}$$

Constraint (23) selects passenger-driver pairs according to selected roundtrip routes, while constraint (24) prohibits selection of passenger-driver pairs other than those from selected roundtrip routes.

6 Optimization Model for Car Sharing

This section studies community-based car sharing and it assumes that each cluster has a pool of cars that can be used by anyone for commuting trips. Model MIP-DC minimizes daily car usage for commuting trips subject to the constraint that the number of inbound routes is equal to the number of outbound routes on any given day. This constraint ensures that the number of cars shared in the cluster remains the same day after day. The model approximates the number of daily cars and routes required for a car-sharing model.[1] Drivers for inbound and outbound trips for any given day do not need to be the same, which makes the model even less restrictive than MIP-DD. The model optimizes trip assignments for each day independently. It uses binary variable x_r like in MIP-DD. Its objective function is given by (7) subject to (8), (9), (12), and

$$\sum_{r \in \mathcal{R}_{i,\delta}} x_r = \sum_{\hat{r} \in \mathcal{R}_{o,\delta}} x_{\hat{r}} \tag{26}$$

Constraint (26) ensures the total number of inbound and outbound routes are the same for any day δ. This model satisfies the same set of principles as MIP-DD.

7 Experimental Results

The Dataset The dataset contains access information of 15 parking structures located in downtown Ann Arbor. Each

[1] For simplicity, we ignore where the cars are parked in the cluster: They can be at a central point or with the drivers. We also ignore how the drivers get to the car, which is not a major issue given the small diameter of the clusters.

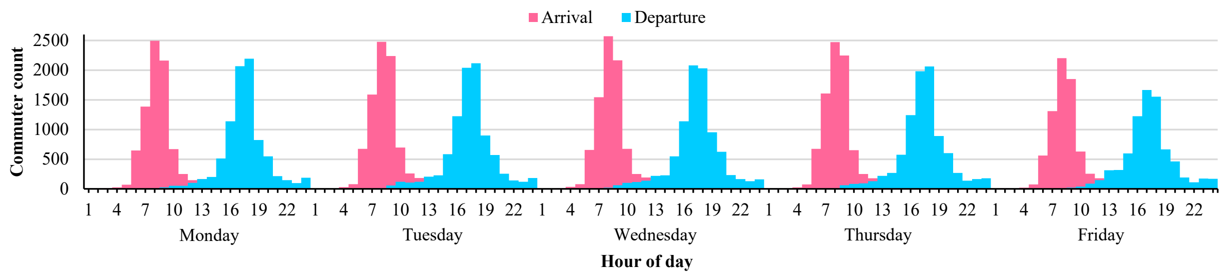


Figure 1: Commuting Patterns on Week 2 (Busiest Week) of April 2017.

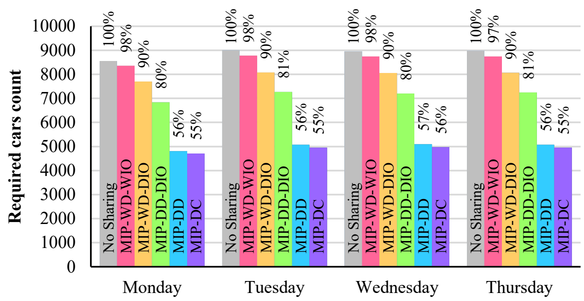


Figure 2: Car Reduction Results for Clustering Approach.

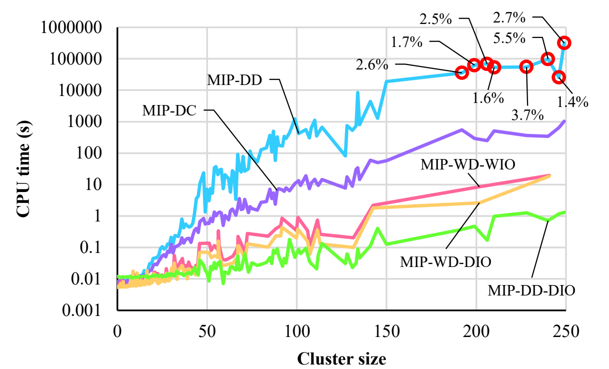


Figure 3: CPU Time as a Function of Cluster Size.

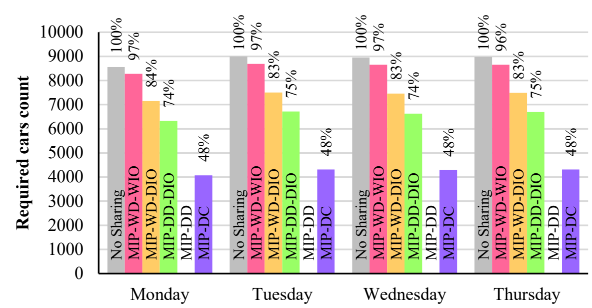


Figure 4: Car Reduction Results for Global Shareability Approach.

entry contains the ID, access time, and direction (in/out) of each customer throughout April 2017. This information was joined with the home address of every customer to reconstruct their daily trips. The dataset provides trip information for 15,000 commuters within an area spanning 13,000 square miles. About 9000 people commute to these parking lots on any given weekday. For more insights, we partition the commuters into two sets; the 4,000 commuters living within city limits (the Ann Arbor region bounded by highways US-23, M-14, and I-94), and the 11,000 commuters living outside that region. Results are given for the busiest week of the month (week 2), and focus on Monday–Thursday, which are the busiest days. Figure 1 depicts the commuting patterns of this population which are remarkably predictable and consistent, a key property for effective carpooling (Buliung et al. 2010).

The Algorithms The values $\sigma_{\text{limit}} = 2$ miles and $\Delta = 20$ mins were used for most experiments. We also include results for $\sigma_{\text{limit}} = 2 \pm 1$ miles and $\Delta = 20 \pm 10$ mins to demonstrate the algorithm's sensitivity to these parameters. The clustering stage converts all GPS coordinates to local Cartesian coordinates before applying MATLAB 2016b's `clusterdata` function. The rest of the algorithm was implemented in C++, using GUROBI 6.5.2 for solving the MIPs. The shortest paths, travel time, and travel distance estimates between any two locations were obtained using GraphHopper's Direction API and OpenStreetMap. All models were executed on a high-performance computing cluster with 8 cores of a 2.5 GHz Intel Xeon E5-2680v3 processor, 64 GB of RAM, and a time limit of 120 hours.

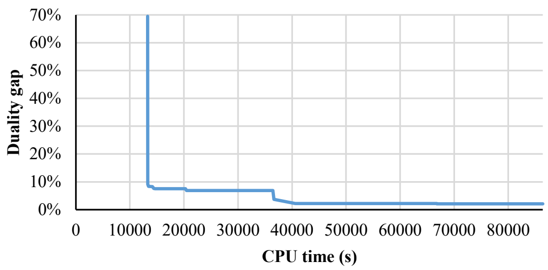
Reduction in Car Usage Figure 2 summarizes results in car reduction during the busiest week of the month with $\sigma_{\text{limit}} = 2$ miles and $\Delta = 20$ mins. It shows the number of cars for the first 4 weekdays under the various optimization models and the clustering approach. It also displays the required number of cars as a percentage of the number of cars in the existing no-sharing conditions.

The first insight is that ride-sharing and car-sharing programs may bring substantial benefits for the city of Ann Arbor. For both programs, the results show a potential reduction of about 44% in car utilization for community-based ride sharing (MIP-DD) and 45% for community-based car sharing (MIP-DC). This would substantially reduce pressure on parking in the city and congestion during the morning and evening commutes.

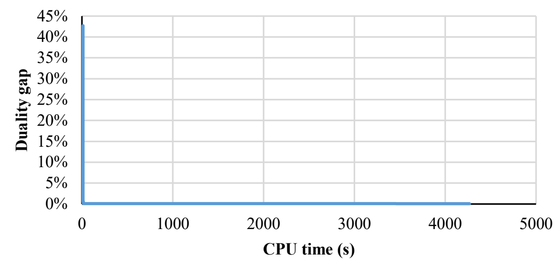
The second insight is that these benefits require flexibil-

Location	Day	MIP-DC		CPU Time (s)		
		CPU Time (s)	Duality Gap	MIP-DD-DIO	MIP-WD-DIO	MIP-WD-WIO
Outside city limits	Monday	13513	0.07%	10	45	66
	Tuesday	38516	0.08%	10		
	Wednesday	10695	0.08%	8		
	Thursday	11288	0.11%	9		
Inside city limits	Monday	28635	0.97%	4275	1904	551
	Tuesday	32235	0.47%	21		
	Wednesday	36848	0.99%	17		
	Thursday	66800	2.10%	21		

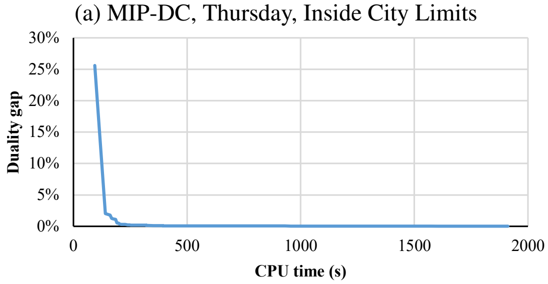
Table 2: CPU Times of Global Shareability Approach with $\sigma_{\text{limit}} = 2$ miles and $\Delta = 20$ mins.



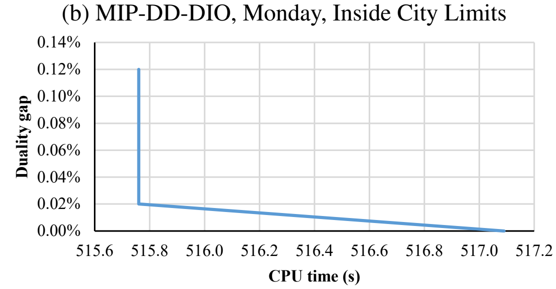
(a) MIP-DC, Thursday, Inside City Limits



(b) MIP-DD-DIO, Monday, Inside City Limits



(c) MIP-WD-DIO, Inside City Limits



(d) MIP-WD-WIO, Inside City Limits

Figure 5: Duality Gap Convergence of Global Shareability Optimization Models.

ity. As the models enforce additional constraints on driver selection and the driver-passenger matching, the results significantly deteriorate. When the matching must be the same inbound and outbound on a given day (MIP-DD-DIO), the potential reduction in car utilization is around 20%. This is still significant but these results also highlight the challenge of matching commuters in roundtrips versus one-way trips. When the drivers and the driver-passenger matching are the same every day (MIP-WD-WIO), the reduction falls to about 2%. It remains around 10% when the drivers are the same every day, but the driver-passenger matching must only be the same inbound and outbound each day (but may differ on different days) (MIP-WD-DIO). It is particularly interesting that desirable properties (4–6) for ride-sharing and carsharing platforms are extremely hard to enforce while reducing car utilization effectively. Any effective platform will require a different sharing pattern for every weekday, although these schedules can be repeated week after week. As a result, these platforms will necessarily impose some psychological burden as commuters need to interact with different people and may have different daily roles.

Figure 3 summarizes CPU times of each optimization

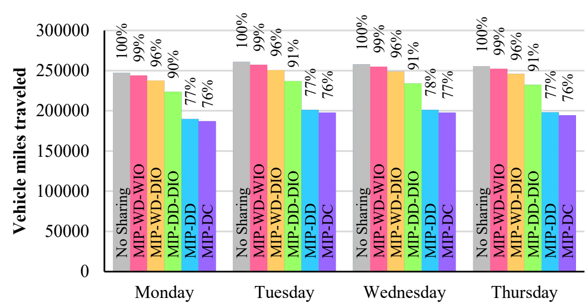


Figure 6: Vehicle Miles Traveled During Trip Sharing.

model for various cluster sizes. CPU times increase exponentially with cluster size for each model, with MIP-DD-DIO consistently being the least expensive and MIP-DD being the most. The MIP-DD model cannot be solved to optimality within the time limit when clusters are of size 150 or more. In these cases, the figure reports the time to achieve the smallest duality gap. The final duality gaps for these clusters are also indicated in the figure.

Figure 4 summarizes car reduction results for each opti-

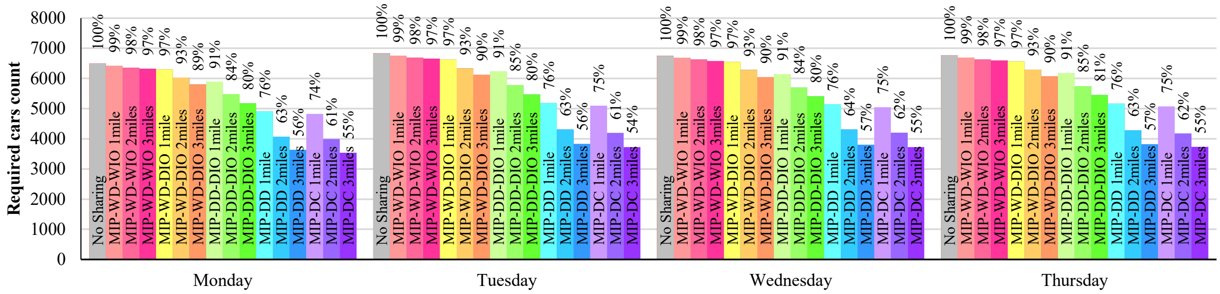


Figure 7: Number of Cars for Trip Sharing of Commuters Living Outside City Limits When $\sigma_{\text{limit}} = \{1, 2, 3\}$ miles.

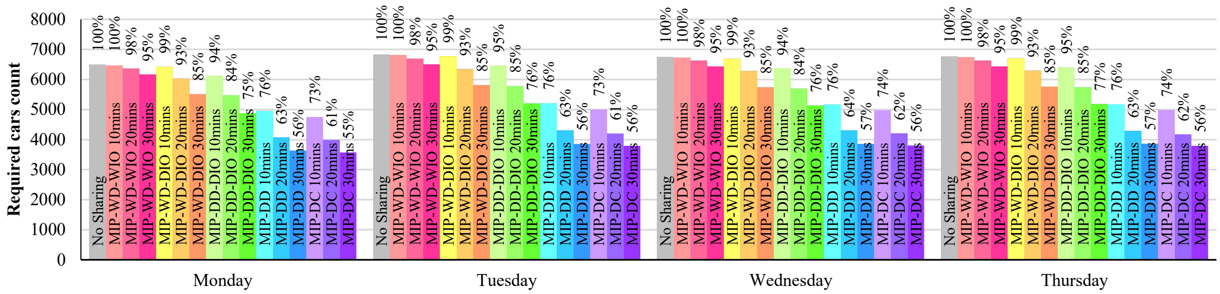


Figure 8: Number of Cars for Trip Sharing of Commuters Living Outside City Limits When $\Delta = \{10, 20, 30\}$ mins.

mization model using the global shareability approach during the same period (week 2) with the same parameters ($\sigma_{\text{limit}} = 2$ miles and $\Delta = 20$ mins), Table 2 shows corresponding CPU times, and Figures 5a–5d show how the solution quality of each model evolves over time in the worst case. Since MIP-DD is intractable in this approach, its results are not shown. MIP-DC cannot be solved to optimality, so the table reports the CPU times to achieve the smallest duality gap (also included in Table 2). Comparison of the results with those from Figure 2 reveals that the global shareability approach consistently produces larger car reduction, with improvements ranging from 1% for MIP-WD-WIO to 7% for MIP-DC. This improvement comes at a price however: The CPU times significantly increase and are 1 to 3 orders of magnitude higher than the clustering approach. It is also worth mentioning that the clustering approach lends itself very well to parallelization, since each cluster can be optimized concurrently, whereas the global shareability approach does not. The trends with respect to flexibility observed in Figure 2 remain present in the global approach, where enforcement of additional selection and matching constraints diminish trip shareability.

Reduction in Miles Traveled Figure 6 shows the total travel distance of all routes for each model from the clustering approach. The results show trends similar to daily car reductions. However, the percentage reduction in daily miles traveled is not as significant as in daily car usage. Commuters living further from the city are less likely to share trips due to the small size of their clusters, while their travel distances account for more in the vehicle miles traveled. Nevertheless, MIP-DD and MIP-DC reduce vehicle miles traveled by an average of 23% and 24% a day which amounts to approximately 58,000 miles a day or 6.5 miles per commuter.

Sensitivity to the Cluster Diameter and Time Windows Figure 7 shows the sensitivity of the algorithm to the cluster diameter σ_{limit} for the commuters outside the city limits. The base value is modified by ± 1 miles while keeping Δ constant at 20 mins. The first observation is that the increase in diameter does not fundamentally change the nature of the prior conclusions: The reductions in car usage for MIP-WD-WIO, MIP-WD-DIO, and MIP-DD-DIO range from 1 to 4% when the diameter is increased. Interestingly, MIP-DC and MIP-DD are most affected by changes in diameters. Increasing (resp. decreasing) the diameter by 1 mile improves (resp. degrades) trip sharing by 6% (resp. 13%) for MIP-DC (MIP-DD is similar). MIP-DD-DIO, which is an intermediate model, have improvements and degradation by about 4%, which is not negligible, but does not bring the model close to MIP-DC and MIP-DD.

Figure 8 shows the performance of each model for commuters living outside city limits as Δ is varied between $\{10, 20, 30\}$ mins while keeping σ_{limit} equal to 2 miles. The results show a stronger sensitivity to the time windows for MIP-WD-WIO, MIP-WD-DIO, and MIP-DD-DIO. When enlarging the time windows, the additional constraints are easier to enforce, showing that the commuting schedule with additional reduction of 3%, 8%, and 9% for these models. MIP-DC and MIP-DD obtain similar benefits when enlarging the time windows and when expanding the diameters.

The Cost of Car Balancing All models ensure that the cars leaving a cluster return to the cluster. Figure 9 shows that the cost of this balancing constraint is relatively small. It

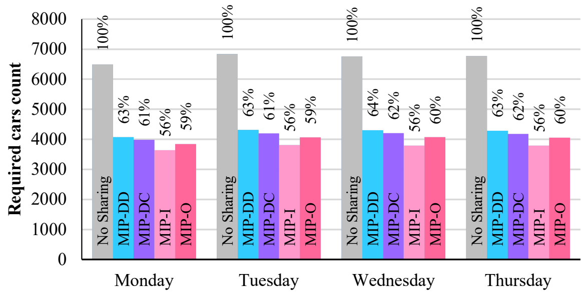


Figure 9: The Cost of Car Balancing.

compares MIP-DD and MIP-DC on the commuters outside the city limits with two models, MIP-I and MIP-O, that minimize the inbound routes and the outbound independently. Balancing the cars induces a cost increase of about 2% for MIP-DC and 4% for MIP-DD over MIP-O. Interestingly, the outbound schedule MIP-O is more challenging due to the less regular outbound patterns of commuters.

8 Conclusion

This paper explored the idea of community-based trip sharing and its application to car pooling and car sharing. It studied the trade-off between the effectiveness of community-based trip sharing in reducing daily car usage and the desirable principles for trip sharing platforms. These ideas were explored on a large case study using a dataset of 15,000 commuters working in downtown Ann Arbor (Michigan).

The paper showed that a platform implementing the core principles for trip sharing can reduce daily car usage by up to 44%, which amounts to approximately 4000 cars, and traveled miles by 58,000 daily. However, as additional principles are integrated, e.g., low coordination costs and clear commuter roles, the benefits progressively reduce and eventually disappear almost entirely. The paper also showed that these results are robust with respect to the cluster sizes and time windows, although more flexibility on both dimensions help alleviate some of the trade-off, with temporal flexibility bringing the most benefits. The study thus indicated that there are trade-offs between the principles themselves.

Future work will be devoted to the maximization of trip sharing opportunities by exploring other clustering techniques, integrating personalized matching constraints based on individual commuter preferences, and scaling the algorithms for applications in large metropolitan areas.

References

Alexander, L. P., and González, M. C. 2015. Assessing the impact of real-time ridesharing on urban traffic using mobile phone data. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, UrbComp '15, 1–9. New York, NY, USA: ACM.

Alonso-Mora, J.; Samaranayake, S.; Wallar, A.; Frazzoli, E.; and Rus, D. 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences* 114(3):462–467.

Arning, K.; Ziefle, M.; and Muehlhans, H. 2013. Join the ride! user requirements and interface design guidelines for a commuter carpooling platform. In *International Conference of Design, User Experience, and Usability*, 10–19. Springer.

Bellemans, T.; Bothe, S.; Cho, S.; Giannotti, F.; Janssens, D.; Knapen, L.; Krner, C.; May, M.; Nanni, M.; Pedreschi, D.; Stange, H.; Trasarti, R.; Yasar, A.-U.-H.; and Wets, G. 2012. An agent-based model to evaluate carpooling at large manufacturing plants. *Procedia Computer Science* 10:1221 – 1227. ANT 2012 and MobiWIS 2012.

Buliung, R.; Soltys, K.; Habel, C.; and Lanyon, R. 2009. Driving factors behind successful carpool formation and use. *Transportation Research Record: Journal of the Transportation Research Board* (2118):31–38.

Buliung, R. N.; Soltys, K.; Bui, R.; Habel, C.; and Lanyon, R. 2010. Catching a ride on the information super-highway: toward an understanding of internet-based carpool formation and use. *Transportation* 37(6):849–873.

Correia, G., and Viegas, J. M. 2011. Carpooling and carpool clubs: Clarifying concepts and assessing value enhancement possibilities through a stated preference web survey in lisbon, portugal. *Transportation Research Part A: Policy and Practice* 45(2):81–90.

Handke, V., and Jonuschat, H. 2013. *Flexible ridesharing*. Berlin, Germany: Springer-Verlag Berlin Heidelberg.

He, W.; Li, D.; Zhang, T.; An, L.; Guo, M.; and Chen, G. 2012. Mining regular routes from gps data for ridesharing recommendations. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, UrbComp '12, 79–86. New York, NY, USA: ACM.

Li, J.; Embry, P.; Mattingly, S.; Sadabadi, K.; Rasmidatta, I.; and Burris, M. 2007. Who chooses to carpool and why?: Examination of texas carpoolers. *Transportation Research Record: Journal of the Transportation Research Board* (2021):110–117.

Poulenez-Donovan, C. J., and Ulberg, C. 1994. Seeing the trees and missing the forest: qualitative versus quantitative research findings in a model transportation demand management program evaluation. *Transportation Research Record* 1–1.

Richardson, A., and Young, W. 1981. Spatial relationship between carpool members'trip ends. *Transportation Research Record* (823).

Santi, P.; Resta, G.; Szell, M.; Sobolevsky, S.; Strogatz, S. H.; and Ratti, C. 2014. Quantifying the benefits of vehicle pooling with shareability networks. *Proceedings of the National Academy of Sciences* 111(37):13290–13294.

Shaheen, S., and Rodier, C. 2005. Travel effects of a suburban commuter carsharing service: Carlink case study. *Transportation research record: Journal of the transportation research board* (1927):182–188.

Trasarti, R.; Pinelli, F.; Nanni, M.; and Giannotti, F. 2011. Mining mobility user profiles for car pooling. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, 1190–1198. New York, NY, USA: ACM.

Tsao, H.-S. J., and Lin, D.-J. 1999. Spatial and temporal factors in estimating the potential of ride-sharing for demand reduction. *California Partners for Advanced Transit and Highways (PATH)*.

Xia, J.; Curtin, K. M.; Li, W.; and Zhao, Y. 2015. A new model for a carpool matching service. *PLOS ONE* 10(6):1–23.

Zhu, M.; Liu, X. Y.; Tang, F.; Qiu, M.; Shen, R.; Shu, W.; and Wu, M. Y. 2016. Public vehicles for future urban transportation. *IEEE Transactions on Intelligent Transportation Systems* 17(12):3344–3353.