INSTANT SOCIAL RIDE–SHARING

Győző Gidófalvi[1,2], Gergely Herényi[3], and Torben Bach Pedersen[2]

[1]Uppsala University, Department of Information Technology, gyozo.gidofalvi@it.uu.se

[2]Aalborg University, Department of Computer Science, {gyg,tbp}@cs.aau.dk

[3]www.motoros.hu: Online Hungarian Forum for Mobility and Transport, gergely.herenyi@motoros.hu

ABSTRACT

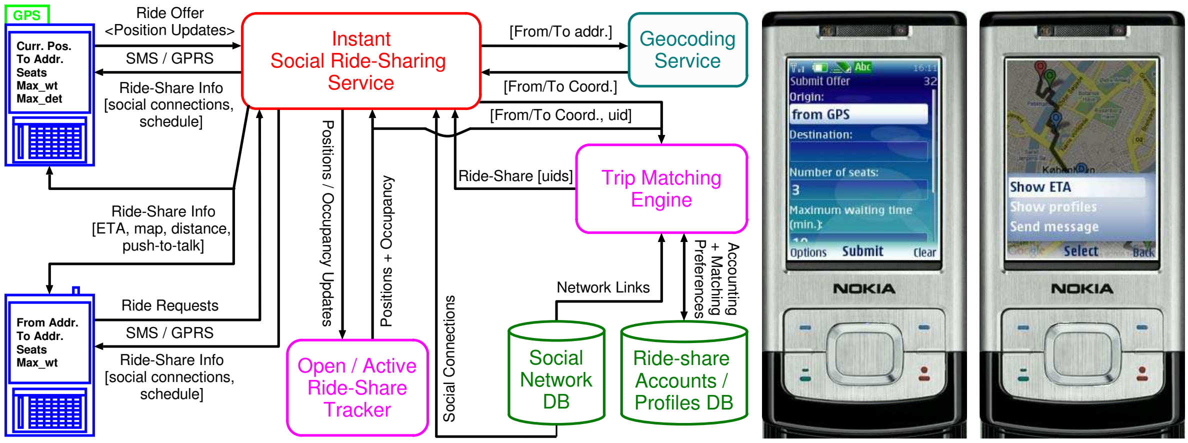
Ride–sharing is a resource efficient mode of personal transportation. While the perceived benefits of ride–sharing include reduced travel times, transportation costs, congestion, and carbon emissions, its wide–spread adoption is hindered by a number of barriers. These include the scheduling and coordination of routes, safety risks, social discomfort in sharing private spaces, and an imbalance of costs and benefits among parties. To this extent, this paper describes implementation details of a system for ride–sharing that eliminates these barriers, in particular the one relating to social aspects, by utilizing the concepts of social networks and social interest groups. Realistic, city–wide simulations show that the proposed social ride–sharing system is viable and effective.

KEYWORDS

Ride–sharing, carpool, social aspects, social network, trust, location–based services

INTRODUCTION

Transportation–related problems, like congestion, parking, and pollution, are increasing in most cities. The magnitude of these problems and the potential for improvements via ride–sharing has been recognized by both industry [6] and academia [7]. To ease the problems, ride–sharing or carpooling has also been encouraged in metropolitan areas by public policies such as the establishment of exclusive lanes, parking places, and reduced road tolls for carpools. Despite the potential for improvements and the public policies, ride–sharing is not widely adopted by the public. The major barriers that hinder wide–spread adoption has been identified as: the lack of effective mechanisms for scheduling and coordinating ride–shares, safety risks, social discomfort in sharing private spaces, and/or an imbalance of costs and benefits among parties [8]. Current commercial products/system for ride–sharing (nuRide, Carpoolworld, liftShare, eRideShare) allow users to post and search trips, and manually construct/negotiate regular ride–shares. Trust is commonly managed through a self–regulatory user–rating mechanism. In comparison, this paper describes a ride–sharing system that 1) by using mobile communication, computing, and positioning technologies allows instant ride–sharing, and 2) by exploiting the exponentially growing popularity of *social networking* and the *voluntary sharing of personal information online* eliminates the social barrier to ride–sharing and the tedious task of manual ride-share scheduling and coordination. An extensive set of experiments on realistically simulated transport and social data evaluates the transport effectiveness of the proposed system and shows it to be a viable solution to our increasing transportation problems.



(a) Process and System Components (b) Submit Offer (c) Ride–Share Info

Figure 1 – The SRSS Process, Components and Mobile Application.

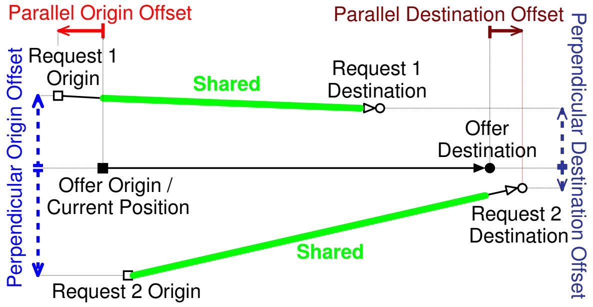
AN INSTANT SOCIAL RIDE–SHARING SERVICE

The Social Ride–Sharing Service / System (SRSS), as depicted in Figure 1(a), is primarily accessible to the users via a mobile phone through a mobile application, but limited functionality can also be provided via an SMS interface. After logging in, a user can submit a *ride offer* or a *ride request*. As shown in Figure 1(b), both an offer and a request specifies 1) a ride/trip *origin* and a *destination* location, 2) a *maximum wait time*, `max_wt`, until the user is willing to wait for a ride–share, and 3) the number of *seats offered* or *requested*, respectively. In addition, an *offer* also specifies 1) a *maximum detour*, `max_det`, that user is willing to make to facilitate ride–sharing, and 2) whether or not the user is willing to participate as a passenger in a ride–share without his/her vehicle. A location can be specified by entering a new or selecting a previously entered/visited address/location, or by using the current location of the user, if a GPS unit is available. In the absence of a GPS unit, an offer is *valid* until either the maximum wait time has passed or all of the offered seats have been filled. If a GPS unit is available, the “origin” of the offer is constantly updated according to the current position, and the offer remains *open/active* as long as the vehicle has seats available and is more than a *minimum distance* away from its destination. A request is *valid* until it is assigned to a ride–share, or it is implicitly or explicitly deleted by the user. As described in the next section, the matching of offers and requests is *automatically* performed by the system, taking into account the spatial and temporal constraints (`max_det`, `max_wt`) of offers and requests, the vehicle capacities, and the social connections between the users. Once a set of offers[1] and requests are matched, i.e., grouped into a ride–share, as shown in Figure 1(c), through the mobile application the SRSS delivers scheduling and social information to the participants in the form of distances, estimated arrival times, maps, and user profiles. Due to the inherent, community–based nature of the proposed SRSS, currently, only a demo version of the SRSS mobile application exists, which shows a subset of the functionality of the application for a fixed input. It can be downloaded from: `http://www.motoros.hu/SRSS/`.

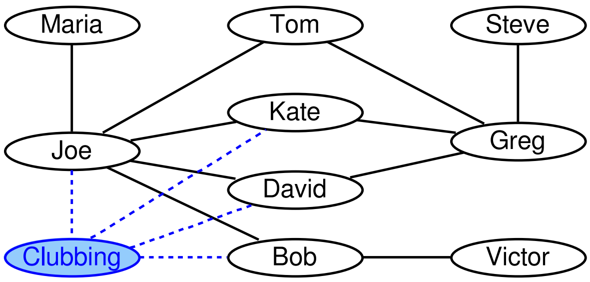
GROUPING TRIPS INTO RIDE–SHARES

The SRSS groups trips into ride–shares according to two objectives: 1) to minimize the extent of the “detour” that offering parties must make in order to serve requests, and 2) to maximize

[1] A ride–share can contain more than one offer if at most one of the offering users requires to be a driver.



(a) Route–Based



(b) Social Network Based

Figure 2 – Route–Based and Social Network Based Trip Matching Criteria.

the amount of social connections amongst participants of ride–shares. Minimizing detours and maximizing social connections in ride–shares is expected to increase the social comfort level and trust among ride-share participants, ultimately leading to increased user acceptance and adoption of ride–sharing. The following subsections give details about two measures that need to be simultaneously optimized to achieve the two objectives, a greedy method that performs the optimization, and four modes of operation.

Detour Measure Between an Offer and a Request

To meet the first objective, as in [4], a simple application–specific *detour* measure is devised between an offer o and a request r. As shown in Figure 2(a), first, the displacement from the origin location / current position of o to the origin location of r and the similar displacement for the destination locations are broken into perpendicular detour offset components (relative to o). Then, the *detour* measure for o and r is calculated as the fraction of the weighted average of the perpendicular detour offset components and the *shared* distance. Weights are adjusted such that the *detour* measure favors requests that can be more naturally accommodated by an offer, i.e., parallel offsets are weighted more. Unlike the measure in [4], the *detour* measure allows the identification of sharing possibilities, in which the request can be accommodated in the middle part of an offer. The *detour* measure is implemented in the SRSS as an SQL function.

Social Connection Between Two Users

To meet the second objective, as illustrated in Figure 1, the SRSS assumes the existence of an online social network data source in which users link to each other and possibly interest groups. In such a network, the number of relatively short paths between two users indicates the *strength of the social connection*, *ssc*, between the two users. In the example in Figure 2(b), Joe and Greg are strongly connected, while Joe and Victor are weakly connected. Given that the offer and the requests 1 and 2 in Figure 2(a), belong to users Joe, Victor, and Greg, respectively, the SRSS is more likely to group Joe and Greg in a ride–share, giving Joe and Greg a lot to talk about. In the SRSS the mutual friendship relation is stored in a database table `F` $= \langle$`pid, fid`$\rangle$. The social connections and associated *ssc*–values between users are stored in a database table `P` and are pre–calculated using the following simple SQL statement[2]:

```
SELECT f1.pid pid1, f2.pid pid2, count(*) ssc FROM F f1, F f2
WHERE f1.fid = f2.fid GROUP BY f1.pid, f2.pid
```

[2]`P` also includes the direct paths between users, which are stored in `F`. To emphasize the importance of such direct paths, such paths are appropriately weighted higher when calculating the strength of social connections.

Calculation of a Set of Candidate Matches

To simultaneously minimize the detours and maximize social connections, the two measures are combined into a single *match score*, ms. To obtain ms, first, both the ssc–values between users and the *detour*–values between offers and requests are scaled to the $[0,1]$–interval. Then, $ms(o,r,u_1,u_2)$, for offer o and request r, for respective users u_1 and u_2, is calculated as $w_d * detour_{[0,1]}(o,r) + w_s / ssc_{[0,1]}(u_1,u_2)$, where w_d and w_s weigh the relative importance of the detour vs the social connection strength in the match score. Given a *maximum detour* threshold, `max_det`[3], the SRSS calculates a set of candidate matches for a set of active offers and valid requests using a single SQL statement as follows:

```
SELECT o.pid, o.rid, r.pid, r.rid, ms(o.rid, r.rid, o.pid, r.pid) ms
FROM ActiveOffer o, ValidRequest r, P
WHERE o.pid = P.pid1 AND r.pid = P.pid2 AND o.seats >= r.seats
  AND detour(o.rid,r.rid) <= max_det
```

Calculation of a Ride–Share Assignment

In a set of candidate matches an offer can be matched with more requests than it can accommodate. At the same time, some requests can potentially be accommodated by only one or a few offers. Given the match scores that are associated with candidate matches, finding an "optimal" assignment of requests to offers, one that has minimal score and ensures that as many requests as possible are accommodated, is not trivial. The following paragraphs formalize the problem and give details of a greedy method that effectively finds a "near–optimal" assignment.

The Ride–Share Assignment Problem: Let G be an undirected bipartite graph $G = (O, R, E)$, where O and R are vertices and represent offers and request, respectively. E is a set of edges, each of which has a weight, $e.w$, the *match score*, and connects a vertex $o \in O$ with a vertex $r \in R$, indicating that o can accommodate r. G is a set of candidate matches. To represent capacity constraints, each vertex in $o \in O$ and $r \in R$ also has an associated measure, $o.s$ and $r.s$, the number of seats that are offered or requested, respectively. Then, an assignment A is a subset of the edges in E, i.e., $A \subseteq E$. Request r is said to be *assigned* to offer o according to A, if there exists an edge $e \in A$ such that $e = (o, r)$. A is *valid* if 1) any request $r \in R$ is assigned to at most one offer $o \in O$, and 2) for any offer $o \in O$ the sum of the requested seats of the requests that are assigned to o is less than or equal to the seats offered by o. The *cover* of an assignment A is the number of requests that are assigned in it, i.e. $|A|$. The *weight* of an assignment A is the sum of the weights of the edges in A, i.e. $\sum_{e \in A} e.w$. Then, the *Ride–Share Assignment (R-SA) Problem* is to find a maximum–cover minimum–weight assignment $A^* \in E$ for G[4].

There are a number of candidate methods that could give potential solutions to the R-SA problem. First, as proposed in [5], the trip grouping algorithm could be used to calculate a minimum weight assignment, but this algorithm would fail to maximize the cover. Second, since the R-SA problem calls for multi–objective or Pareto optimization, which is related the "skyline" query in the database literature [1], one could view the solution to the R-SA problem as a skyline query of two objective functions $cover(G, A) \to \mathbb{R}$ and $weight(G, A) \to \mathbb{R}$ over the space of assignments $\mathbb{A}_G$, i.e., the objective space. However, since the objective space is potentially large and is not enumerated, such an approach is likely to be infeasible. Finally, a simplified

[3]For clarity, a common `max_det` value is used, but user–defined threshold values are trivial to implement.

[4]Given a problem instance there can be many solutions that achieve the same cover and weight.

version of the problem, in which the second condition of assignment validity is not required, is known as the weighted bipartite matching problem [2]. The necessary modifications to a bipartite matching based solution are unknown and likely to be complex.

Instead, to solve the R-SA problem, a simple but effective, greedy method is devised that starting from an initially empty partial assignment A', at every iteration selects and adds a new edge e to A' such that e minimizes a *single*, *local* objective function $assign\_score_{G,A'}(e) \rightarrow \mathbb{R}$. To treat the multi–objective nature of the R-SA problem, the single, local objective function $assign\_score_{G,A'}(\cdot)$ is constructed as a weighted sum of the partial derivative of $weight(G, A) \rightarrow \mathbb{R}$ and a derived function $demand_{G,A'}(o) \rightarrow \mathbb{R}$, which is inversely proportional to the partial derivative of $cover(G, A) \rightarrow \mathbb{R}$. Figure 3 lists the pseudo code of the greedy assignment algorithm and shows how the single, local objective function is calculated and used in the optimization. After initializing an empty assignment A (line 2), the weights are scaled to the $[0, 1]$–interval (line 3). Then, while there are offers and requests (line 4), lines 5 through 12 are iterated. On lines 5, 6, and (7), for each offer $o_j \in O$ a (scaled) *demand* measure, $o_j.d$, is calculated as the fraction of the number of seats requested (from o_j) and the number of seats offered (by o_j). Then, on lines 8 and 9, for each edge $e_i \in E$ an *assign score*, $e_i.as$, is calculated as a weighted sum of the scaled demand measure and weight. Then, on lines 10 and 11, edge e_{i^*} with the lowest assign score is greedily selected and added to A. Finally, on line 12, G is accordingly updated to reflect the new addition to the assignment. The proposed greedy method is implemented in the SRSS as a small set of simple SQL statements.

(1) **procedure** *greedyAssign* $(G = (O, R, E))$
(2) $\quad A \leftarrow \emptyset$
(3) $\quad$ **foreach** $e_i \in E \quad e_i.w_{[0,1]} = e_i.w / \max_{e \in E} e.w$
(4) $\quad$ **while** $(|O| > 0$ and $|R| > 0)$
(5) $\quad\quad$ **foreach** $o_j \in O$
(6) $\quad\quad\quad o_j.d = \sum_{r_k \in R : \exists e_i = (o_j, r_k) \in E} r_k.s / o_j.s$
(7) $\quad\quad$ **foreach** $o_j \in O \quad o_j.d_{[0,1]} = o_j.d / \max_{o \in O} o.d$
(8) $\quad\quad$ **foreach** $e_i = (o_j, r_k) \in E$
(9) $\quad\quad\quad e_i.as = w_c * o_j.d_{[0,1]} + w_w * e_i.w_{[0,1]}$
(10) $\quad\quad i^* = \mathrm{argmin}_i \, e_i.as$
(11) $\quad\quad A \leftarrow A \cup e_{i^*}$
(12) $\quad\quad G \leftarrow$ *updateG* (G, e_{i^*})
(13) $\quad$ **return** A

Figure 3 – Greedy Assignment Algorithm.

Four Modes of Operation

Depending on when requests are assigned to offers and when offers depart, the SRSS has four possible modes of operations. First, requests can be assigned to offers that have not yet departed, i.e., are *stationary*, or to offers that have already departed, i.e., are *mobile*. Obeying the wait time constraint, offers can depart, in an *eager* fashion, immediately after the first assignment, or can wait, in a *lazy* fashion, for more assignments. The advantages and disadvantages of the modes of assignment and modes of departures are listed in the table below:

Mode	Advantages	Disadvantages
Stationary	no GPS or online routing is needed; ride–share information is available prior to departure → less distraction to the driver	misses possible ride–share opportunities after departure → less effective
Mobile	allows for more ride–sharing → more effective	information relating to ride–shares can distract the driver
Eager	shorter offer and request wait times	shorter ride–share opportunities → less effective
Lazy	longer ride–share opportunities → more effective	longer offer and request wait times

GENERATION OF SIMULATED DATA SETS

Since real–world, *related / interlinked*, large–scale transport and social data sets are not available, in an attempt to evaluate the effectiveness and feasibility of the proposed SRSS, simulated data sets are derived and used. Realistic, city–wide transportation data is derived from a Spatio–Temporal ACTivity Simulator (ST–ACTS) [3] for a population of prospective users. Using ST–ACTS, its underlying real–world data sources, and a simple assumption about how people build social connections in the real world, a social network of the prospective user population is also derived. Finally, using realistic assumptions about how users adopt new services, a subset of the prospective users is selected as SRSS users. The following paragraphs give details about how the three simulated data sets are derived, respectively.

Transportation Data Set: The transportation data set is derived from ST–ACTS [3]. ST–ACTS is a probabilistic, parameterizable, realistic spatio–temporal activity simulator. ST–ACTS is realistic in the sense, that it is based on a number of real–world (geo–demographic, transportation, business, consumer survey) data sources and a set of principles that model the *social* and some of the *physical* aspects of mobility. The output of ST–ACTS is a population of simulated persons, each uniquely identified by an ID, `pid`, described by a set of geo–demographic variables and associated with a sequence of time–stamped activities performed at particular physical locations, i.e., coordinates. Treating the coordinates that are associated with two consecutive activities of a simulated person as the origin and destination locations of a trip, realistic trips of approximately 590,000 individuals in the city of Copenhagen, Denmark is simulated for the course of a workday. Since a short trip to a local coffee shop does not represent a great opportunity for ride–sharing, only trips that are longer than 1 km are retained as candidates for ride–sharing. The resulting data set is considered as the transportation data set and represents the transportation needs of a prospective user population. The transportation data set contains 1.74 million trips of approximately 548,000 prospective users. The length of an average trip is 2.9 ± 1.8 km and an average prospective user performs 3.2 ± 1.6 trips during the day.

Social Network Data Set: A social network of the prospective user population is derived based on a simple assumption about how people build real–world social connections: People tend to make social connections with other people that they frequently interact with in physical space. More specifically, for a given prospective user u, based on the location of the home and work place of u, and the (1 out of 29) geo–demographic group that u belongs, three types of friends are selected: home, work, interest. The selection of the home and work friends is spatially constrained, i.e, the home (work) locations of home (work) friends are required to be within 500 (250) meters of each other. The selection of interest friends is constrained by the geo–demographic groups of the prospective users, i.e., two interest friends must belong to the same geo–demographic group. Using the constraints, for each prospective user approximately 12.5 friends are selected for each type of friend[5], resulting in a social network data set that contains approximately 11 million mutual connections between the prospective users.

SRSS Users Data Set: The two main forces that drive the adoption process of the proposed SRSS is the “herd mentality” of users and the service utility that users foresee from joining the service. These forces are materialized in the simulation as follows. First, according to the herd mentality the more invitations a given prospective user receives from his/her friends, the more likely he/she will join the SRSS. Second, the more a given prospective user travels and the

[5]Since friendship is mutual, to get a close to even distribution of the number of friends that prospective users have, the number (and the identities) of already selected friends of a prospective user are maintained during the selection. Candidate friends with a lower number of friends are prioritized in the selection.

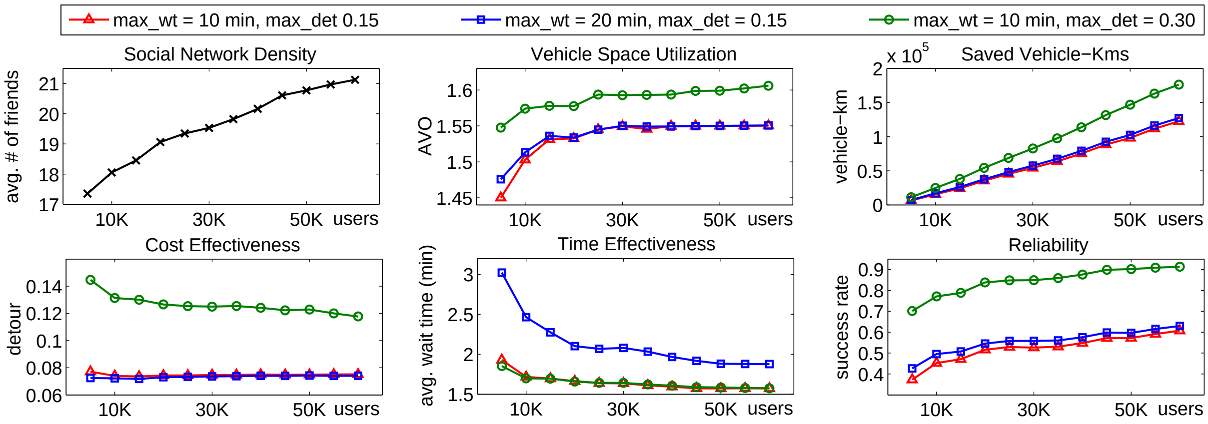


Figure 4 – Experimental Results for Varying Parameter Settings and Number of Users.

more friends he/she has, who have already joined the SRSS, the more likely he/she will find the service useful. Accordingly, given the transportation and social network data sets, the joining / adoption process is simulated iteratively according to six steps as follows. In Step 1, to assess travel needs of users, for each prospective user $u \in U_p$, a *service utility*, $u.su$, is calculated, which is linearly proportional to the number of, and the sum of the length of, the trips that u performs. In Step 2, as a seed, the first SRSS user u_1, with the highest travel needs / service utility, i.e., $u_1 = \mathrm{argmax}_{u \in U_p}\, u.su$, is selected. In Step 3, simulating the invitation process, the `pid`s of u_1's friends are recorded. In Step 4, for each so far invited user $u \in U_i \subseteq U_p$ (currently the friends of u_1 only) a *join score*, $u.js$, is calculated, which is linearly proportional to $u.su$ and the number of invitations that u has received so far. In Step 5, the next SRSS user, u_n is selected as the invited user $u \in U_i$ with the highest $u.js$, i.e., $u_n = \mathrm{argmax}_{u \in U_i}\, u.js$. Finally, in Step 6, the friends of u_n get invited[6]. Subsequent SRSS users are selected from the invited users by iterating Steps 4 through 6. The resulting SRSS users data set contains 60,000 users: 60% of which are offering, and 40% of which are requesting rides.

EVALUATION OF TRANSPORT EFFECTIVENESS AND FEASIBILITY

Using the simulated data sets, for each mode of operation a set of experiments were conducted to measure the transport effectiveness and feasibility of the proposed SRSS. Transport effectiveness for varying number of users and under various parameter settings (`max_det` and `max_wt`) are evaluated according to three measures: resource–effectiveness, time–effectiveness, and reliability. These measures give answers to the following three groups of questions, respectively: "What is the Average Vehicle Occupancy (AVO)? How many vehicle–kilometers are saved? What is the average amount of detour in ride–shares?", "How long must a user wait for a ride–share on average?", and "With what probability can a request be served?"

The four sets of experiments verified the transport effectiveness related advantages and disadvantages of the four modes of operation. In particular, stationary assignment consistently resulted in lower resource–effectiveness, longer wait times, and lower reliability than its mobile alternative. Similarly, eager departure consistently resulted in moderately lower resource–effectiveness, substantially shorter wait times, and moderately lower reliability than its lazy alternative. Due to space constraints, Figure 4 only presents detailed results for the arguably most effective mode of operation: mobile assignment with eager departure. The trends in the transport effectiveness measures w.r.t. the number of users are clear: as the number of users increases, the number of social connections between users increases, which allows more ride–

[6] To avoid reinvitation (and duplicate users) only the friends of u_n get invited who are not already SRSS users.

share opportunities and more effective transport. At a reasonable level of participation, 60,000 users, i.e., 10% of the population: the AVO–level is raised from 1 to 1.61; users wait 2.4 minutes for a ride–share; offers have to make a detour that is 8% of the shared distance; and 91% of the requests can be served. The corresponding savings, during the course of a single workday in Copenhagen, are very substantial, amounting to 32% of the normal transport cost, specifically 176,000 vehicle–kilometers, 14,000 liters of fuel, and 32.7 tonnes of CO_2–emissions. This makes the proposed SRSS clearly viable and effective.

CONCLUSIONS AND FUTURE WORK

The paper described a Social Ride–Sharing System / Service (SRSS), which reduces 1) the burden of manual scheduling and coordination of trips, and 2) the safety risks and social discomfort that is commonly associated with ride–sharing. To achieve these improvements the SRSS automatically groups ride offers and requests into ride–shares according to two objectives: 1) to minimize ride–share detours, and 2) to maximize the amount of social connections amongst participants of ride–shares. A mobile application that places the SRSS at the fingertips of mobile users (anyplace and anytime) was also developed and illustrated. Realistic simulations for a large European city showed that the proposed SRSS is highly viable and effective, resulting in very large savings of approximately one third of the original transport costs. As service adoption grows even further, the number of possible user connections will grow even more, leading to even higher savings. On top of this, the novel consideration of the social aspects not only leads to increased service adoption, but more significantly makes ride–sharing much more enjoyable and potentially rewarding, both in a personal and a business related sense.

Future work is planned along several directions. First, efficient implementations of a road network distance based detour measure will be investigated. Second, similarly to previous research [5], using a data stream management system, a highly scalable parallel implementation of the SRSS will be devised. Finally, alternative system architecture possibilities will be considered. In particular, the possibility of distributing much of the matching task among the computationally powerful mobile devices will be explored.

REFERENCES

[1] S. Börzsönyi, D. Kossmann, and K. Stocker. "The Skyline Operator." In *Proc. of ICDE*, pp. 421–430, 2001.

[2] T. H. Cormen et al. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 2001.

[3] G. Gidófalvi and T. B. Pedersen. "ST–ACTS: A Spatio-Temporal Activity Simulator." In *Proc. of ACM-GIS*, pp. 155–162, 2006.

[4] G. Gidófalvi and T. B. Pedersen. "Cab–Sharing: An Effective, Door–to–Door, On–Demand Transportation Service." In *Proc. of ITS Europe*, 2007.

[5] G. Gidófalvi, T. B. Pedersen, T. Risch, and E. Zeitler. "Highly Scalable Trip Grouping for Large–Scale Collective Transportation Systems." In *Proc. of EDBT*, pp. 678–689, 2008.

[6] S. Hartwig and M. Buchmann. "Empty Seats Travelling: Next Generation Ridesharing and its Potential to Mitigate Traffic– and Emission Problems in the 21st Century." Nokia Research Center Technical Report: NRC-TR-2007-0003, 2007.

[7] *Online Workshop on Ridesharing Supported by Location Reporting Phones*. Proceedings can be obtained by contacting Jim Longbottom at CEETI.org, 2005.

[8] P. Resnick. "SocioTechnical Support for Ride Sharing": `http://www.si.umich.edu/~presnick/papers/rideshare/draftscenario.pdf` (last accessed: 17/08/2008)