

Exploiting Graph-theoretic Tools for Matching in Carpooling Applications

Luk Knapen · Ansar Yasar · Sungjin Cho · Daniel Keren · Abed Abu
Dbai · Tom Bellemans · Davy Janssens · Geert Wets · Assaf Schuster ·
Izchak Sharfman · Kanishka Bhaduri

Received: date / Accepted: date

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 270833

L. Knapen
Hasselt University, Transportation Research Institute IMOB,
Wetenschapspark 5 bus 6 3590 Diepenbeek, Belgium
Tel.: +32-11-269111
Fax: +32-11-269199 E-mail: luk.knapen@uhasselt.be

A. Yasar
Hasselt University, Transportation Research Institute IMOB

S. Cho
Hasselt University, Transportation Research Institute IMOB

D. Keren
Department of Computer Science
University of Haifa, Haifa 31905, Israel
Tel.: +972-4-824-9730
Fax: +972-4-824-9331 E-mail: dkeren@cs.haifa.ac.il

A. Abu Dbai
Department of Computer Science
University of Haifa, Haifa 31905, Israel
E-mail: roz.blaban@gmail.com

T. Bellemans
Hasselt University, Transportation Research Institute IMOB

D. Janssens
Hasselt University, Transportation Research Institute IMOB

G. Wets
Hasselt University, Transportation Research Institute IMOB

A. Schuster
Faculty of Computer Science, Technion Israel Institute of
Technology, Haifa 32000 (Israel)

I. Sharfman
Faculty of Computer Science, Technion

Kanishka Bhaduri
Netflix Inc.
100 Winchester Cir. Los Gatos, CA, 95032 United States
Tel.: 408 540 3331 (office)
E-mail: kanishka.bh@gmail.com

Abstract An automatic service to match commuting trips has been designed. Candidate carpoolers register their personal profile and a set of periodically recurring trips. The *Global CarPooling Matching Service (GCPMS)* shall advise registered candidates how to combine their commuting trips by carpooling. Planned periodic trips correspond to nodes in a graph; the edges are labeled with the probability for success while negotiating to merge two planned trips by carpooling. The probability values are calculated by a learning mechanism using on one hand the registered person and trip characteristics and on the other hand the negotiation feedback. The probability values vary over time due to repetitive execution of the learning mechanism. As a consequence, the matcher needs to cope with a dynamically changing graph both with respect to topology and edge weights. In order to evaluate the matcher performance before deployment in the real world, it will be exercised using a large scale agent based model. This paper describes both the exercising model and the matcher.

Keywords Graph theory · Binary matching · Agent-based modeling · Scalability · Dynamic networks · Learning

1 Introduction

An advisory service for carpooling while commuting is to be built. People will register their periodic commuting trips: the base period typically is one week i.e. a specific pattern valid for working days is repeated after every seventh day. Considering one week periods accommodates for most situations (including part-time workers).

People who are able to fulfill all their carpooling needs within their own social network of acquaintances (*local exploration*), are assumed not to need the service.

Others will need to explore the set of carpooling candidates yet unknown to them and managed by a web based *GCPMS* (Global Car Pooling Matching Service): this is called *global exploration*. The matching service integrated in the *GCPMS* shall determine which trips are best suited to be merged for carpooling and shall provide advice by suggesting people to start a negotiation with respect to a specific periodically executed trip.

Testing software that implements a *GCPMS* is essential because providing inaccurate or wrong advice initiates negotiations having a large probability to fail, which can expel customers. Deployment of a *GCPMS* shall go flawlessly because lost customers will be reluctant to return. However, the integrated matching mechanism can only be verified for operational fitness by testing under real world conditions. This complicates testing since (i) the *GCPMS* requires a critical mass of registered users in order to operate effectively and efficiently, (ii) performance and effectiveness need to be evaluated on a running system because they are very difficult to predict from design data only and finally (iii) the behaviour of the advisor (with respect to accuracy) during the initial phase is difficult to predict and observations made are difficult to interpret; this phase corresponds the startup transient phenomenon where stable operation has not yet been reached.

Therefore we propose the use of an agent-based model simulating the customer community in order to exercise the matching service for testing and system validation. This paper describes aspects of the combined setup of the *MAS* exerciser and the *GCPMS* under test and focuses on the required matching component.

The remainder of the paper is organized as follows. Section 2 presents related work. Section 3 explains the principle of operation for the *GCPMS* and shows the test environment setup using a multi-agent system (*MAS*). Section 4 discusses several functions used to model domain specific concepts and shows how different functions can be required to implement a particular concept in the *GCPMS* and the *MAS* respectively. Section 5 explains how the *GCPMS* determines the success probability for the negotiations to succeed. Section 6 describes the problem of matching along with some proposed solutions. It also presents an early experiment to estimate computational performance. Finally sections 7 and 8 present future research directions and conclusions respectively.

2 Related Work

In recent years, agent-based simulation has entered the field of transportation science because of its capability to analyse aggregated consequences of individual specific behaviour changes.

[14] investigates the effect of environmental conditions and plans to incorporate the agent interactions required when carpooling.

[10] describes an agent-based model aiming to optimally combine demand and supply in an advisory system for *repeated ride-sharing*. The authors focus on the mechanisms required to model users cooperating on joint plans and focuses on the economic value of the shared plans; this research focuses on the fairness of the payment system but does not consider the rideshare demand and supply change in time.

[1] focuses on dynamic non-recurring trips which is related to commuting carpooling but requires different solution concepts. Both maximal individual advantage and system wide optimum are considered.

[4] describes a group negotiation protocol for agreement on agenda schedules. A group can consist of two or more agents. The negotiation mechanism is based on ideas drawn from the A* shortest path algorithm. Each agent is assumed to specify its most preferred option first and to specify consecutive new proposals in non-increasing order of preference. Each one uses a private (i.e. not published) utility function. The protocol initiator makes use of a proposal evaluation function that is based on the assumption that agents behave as mentioned before. Versions using preference feedback by agents and conflict resolution by initiator are reported to result in nearly optimal solutions using a quite small number of negotiation rounds.

[11] studies the problem of finding an optimal route for co-traveling. The origin (home) and destination (work) locations are given for each individual as well as a set of carpool parkings. Each of those home, work and parking locations are possible transferia (locations where to change travel mode or to change vehicle) where one can join or leave a carpool. Each individual declares the maximal time and/or distance that is acceptable to move from origin to destination. The combined route (co-route) that solves the problem consists of a join part and a fork part. In the join tree, carpoolers enter the main drivers car at several locations and times. In the fork tree they successively leave the car and, if not at their destination, continue their trip by other means. The paper proposes an algorithm to find the optimal solution for the join tree.

[21] provides an informal and formal problem statement for the LCPP (Long term Car Pooling Problem).

Then the soundness of the problem formulation is argued and some properties of the LCPP are proved. Finally the problem is proved to be NP-complete. This paper assumes that pools are stable in time and that every member in turn acts as the driver (round robin concept).

[15] describes an interactive system to support the mobility manager (officer) operating on the LTCPP (long term car pool problem). The proposed methods and models make use of clustering analysis (CA). The basic hypothesis is that in a group the driver of the shared car turns among the participants (similar to [21]). Clustering procedures using methods available in standard DSS (decision support system) are proposed. After clustering, for each driver a TSP (traveling salesman problem) is to be solved. Similarity measures are used but not discussed in the paper. The result is a GUI based interactive system that can be applied to company employees. A case study for a public service in the city of Bologna is presented. Experiments show that the overall relative saving in distance and time increases with the number of participants.

[7] describe mining algorithms to discover *user link* and *location link* patterns respectively. User link patterns focus on similarity between the sequences of locations visited by individuals. User link patterns apply to sequences of locations. Both are relevant when trying to estimate the probability for people to be able to carpool.

[20] derives *travel routine* from sets of GPS traces. Similar trips are extracted (based on space and time-of-day). A routine is defined as a sufficiently large set of similar traces belonging to an individual. A profile is a set of routines. Based on the assumption that the passenger walks for a given maximal distance to (from) a location where (s)he is picked up (dropped off) by a driver, an upper bound for carpooling is determined using travel profiles.

Person traces can provide more information than car traces. Carpooling induces mutual dependency and hence additional uncertainty about the driver's and passengers timeliness. Exchanging location information is used to help solving this problem. This however requires energy efficient localization techniques and ubiquitous coverage like the one presented in [18].

[22] infers social ties between people from semantic location histories (SLH). GPS trajectories first are annotated by assigning a meaning to each visited location. Then user's movements are modeled as sequences of semantically annotated locations. Finally, similarity between users is calculated by comparing their semantic location sequences. The method is demonstrated using the public GeoLife GPS trajectories dataset.

[19] presents a multi-agent system that models joint social activity execution. Although not focusing carpooling directly, it is important in this context because it focuses on cooperative activity execution while executing daily agendas which is related to the concept of co-traveling (cooperative trip execution).

Finally, a large body of literature ([17], [6], etc) has been published about the concept of rescheduling activities in a daily agenda. This however, considers agenda adaptation to unexpected events as opposed to rescheduling in the context of negotiation to cooperate. [3] presents an overview of the Aurora activity-based model for schedule generation and adaptation. People are simulated as individual agents. A comprehensive model has been specified describing the insertion, repositioning, deletion and substitution of activities as well as changing locations, trip chaining options and transport modes. Models of this level of detail are required to integrate carpooling concepts in a simulator. The paper describes the use of *Aurora* in an experimental setup to study schedules consisting of work activities and green activities in several scenarios.

3 Carpooling Model

3.1 Problem Context

This paper focuses on carpooling for commuting i.e. planned periodic cooperative traveling, not on ad-hoc ridesharing where people try to find companions for a single ride in the very near future (usually within the same day). In order to find carpooling companions, people who did not find a suitable partner by exploring their private network, register themselves with the *GCPMS* service. Registration implies first posting some characteristics describing the individual like age, gender, education level, special interests (like music style preferences), job category, driver license availability, etc. Those qualifiers are used because it is known that continued successful cooperation between people requires a minimal level of similarity [16].

Secondly, people post information about each trip they periodically plan to execute: those data consist of source and destination locations, earliest and latest departure and arrival times, the maximal detour distance and delay that are acceptable, as well as the availability of a car (possibility to drive). Note that a particular driver license owner can be unavailable for driving on a specific day of the week because the family car on that day is in use by her/his partner.

Periodic trip executions need to be matched, not people. A periodic trip on Wednesday from A to B leaving at about 08:30h needs to be matched with another

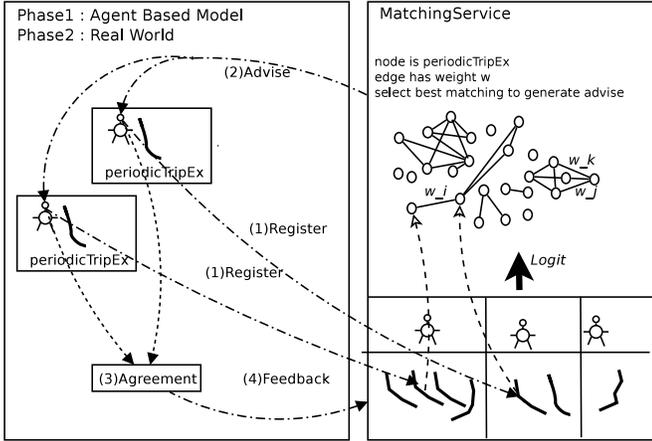


Fig. 1 Application context: the right hand side shows the *matcher service*. People register some descriptive data about themselves and trips to be executed periodically (*periodicTripEx*). Those constitute a graph : the edges are labeled with the probability that negotiation will succeed when the trip owners are advised to carpool. Negotiation result is fed back to train the *logit* predictor. The left hand side shows the entities exercising the *matcher service* in consecutive phases.

one having similar characteristics. Of course, the people involved shall be mutually compatible but they are not the primary subject of matching. A particular individual can periodically carpool with several people for different trips in the week (on Monday with colleague A, on Tuesday with neighbour B who differs from A). Periodic trip execution is abbreviated by *periodicTripEx* in the remainder of the text.

After having found a good match (details on how to do so will be explained below) the matcher conveys its advise to the candidates involved (the owners of the matched *periodicTripEx*); they evaluate the proposal, negotiate about carpooling and possibly agree to cooperate. Note that this negotiation is not guaranteed to succeed. One of the reasons is that the individuals dispose of more information during the *negotiation process* than the service does during the *matching process*. Therefore, the candidates convey the negotiation result back to the matcher service. This paper assumes that sufficient (financial) incentives are in place in order to make this happen. The feedback is used by a learning mechanism incorporated in the matching service. After receiving the feedback, the matching service disposes of the *periodicTripEx* and the individuals characteristics as well as of the negotiation result; those are used to train a predictor. Please refer to Fig. 1 for a high level overview of dataflows, relations and method activation.

It is important to note that the first implementation focuses on *pairs* of commuters carpooling. This is essen-

tial to the problem of edge weight determination and it allows the advice to be based on *binary matching*.

The model used for matching consists of a directed graph; by convention, each edge points to the *periodicTripEx* whose owner will be the driver. Each vertex corresponds to a *periodicTripEx*. A vertex for which the owner is unable to become the driver, never can be a target edge (its *indegree* equals zero). Every edge is labeled with the estimated probability for the the negotiation to succeed. Two vertices are connected by an edge if and only if it is worth to advise the *periodicTripEx* owners to start negotiating: in general the graph is incomplete. The need to determine the probability threshold to include an edge in the graph is one of the main reasons to use a MAS to exercise the *GCPMS*. Note that

1. the set of vertices evolves over time because people register and withdraw *periodicTripEx* as time evolves and because people join and leave the carpooling candidates society (removing all their *periodicTripEx* in the latter case).
2. edges emerge as soon as the negotiation success probability exceeds a given threshold; this can be caused by changes in the *periodicTripEx* (e.g. by relaxing the time constraints) and people characteristics respectively (e.g. by reputation changes (see below)).
3. probability estimates can change over time by re-training the predictor. Note that this can cause threshold crossing and hence edge creation or deletion.

Finally the problem size can grow large when a nationwide service is considered. Large scale deployment probably is a necessary condition for both effective operation (delivery of advice that has a high success probability) and economic viability. The matcher needs to cope with large networks whose topology and edge weights evolve in time. This represents a complex problem and hence thorough evaluation before deployment.

3.2 Basic Concepts - Definitions

Some definitions for concepts in the application domain are required to explain the functions used to calculate the edge weights for the *periodicTripEx* graph. Refer to Fig. 2 for an overview of the datasets and relations involved. Fig. 3 summarizes the essential application domain functions.

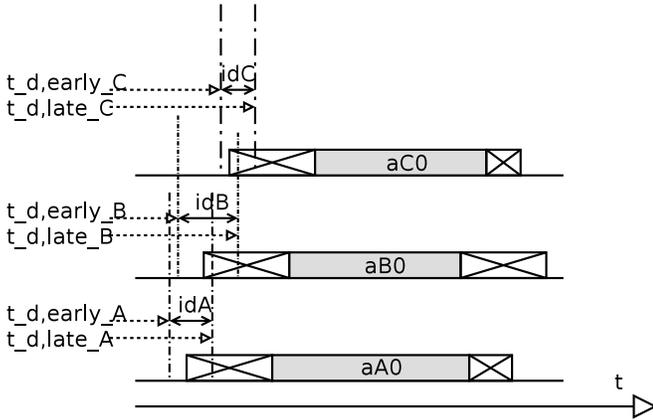


Fig. 4 Activities ($a_{A,0}, a_{B,0}, a_{C,0}$) for individuals A, B and C and the associated trips. The valid departure intervals $i_{d,A}, i_{d,B}, i_{d,C}$ are shown. Note that B can choose to co-travel with A or C but A and C cannot co-travel.

3.2.1 Symbols Used

\mathcal{A}	: the set of all <i>agreements</i> (see definition 2)
\mathcal{I}	: the set of all individuals
\mathcal{P}	: the set of all pools (see definition 3)
$range(TOD)$: $24 * 60 * 60$ (time-of-day)
$range(TOW)$: $7 * range(TOD)$ (time-of-week)
\mathcal{T}	: The set of all <i>periodicTripEx</i> 's (see definition 1)
TOD	: Time of day ; if expressed in seconds, cardinal $\in [0, range(TOD) - 1]$
TOW	: Time of week ; if expressed in seconds, cardinal $\in [0, range(TOW) - 1]$
$t_{.,early}, t_{.,late}$: Earliest resp. latest time
$t_{d.,}, t_{a.,}$: Departure resp. arrival time

3.2.2 Definitions

Definition 1 (periodicTripEx) A *periodicTripEx* is a tuple $(i, O, D, w, t_{d,early}, t_{d,late}, t_{a,early}, t_{a,late})$ where $i \in \mathcal{I}$, O and D denote the origin and destination locations respectively, $t_{.,} \in TOW$ and w denotes a start-of-week moment in time so that the first trip execution for the given *periodicTripEx* starts in $w, w + 1$.

Notes:

1. A *periodicTripEx* denotes the weekly execution of a trip with given characteristics by a specific individual. Individual i is called the owner of the *periodicTripEx*.
2. Examples:
 - (a) $t_{d,late}$ is: *Wednesday at 08:20h*
 - (b) $w = 2012\text{-jun-04 } 00:00:00$
 - (c) Refer to Fig. 4 to see intervals overlap.

Definition 2 (agreement) An *agreement* specifies operational details about the collaborative execution of all elements in the list of *periodicTripEx* to which the *agreement* applies. An *agreement* specifies the moment in time at which it starts to hold.

Notes:

1. An *agreement* has no termination time
2. A *periodicTripEx* is referred to by (belongs to) at most one *agreement*
3. *Agreement* details cover: timing, routing and driver selection.

Definition 3 (pool) A *pool* is a tuple $(\{a\}, T)$ where $\{a\}$ denotes a set of *agreements* negotiated by the cooperating partners and T is a nonempty set of *periodicTripEx* so that $(\{a\} = \emptyset \wedge \|T\| = 1) \vee (\|\{a\}\| = 1 \wedge \|T\| > 1)$

Notes:

1. The condition states that there is either a single individual without any *agreement* or multiple cooperating individuals sharing a single *agreement*.
2. Each individual occurs in at most one *periodicTripEx* in a specific *pool*:

$$\forall t_0, t_1 \in T : (t_0 \neq t_1) \Rightarrow (t_0.owner \neq t_1.owner) \quad (1)$$

where $t.owner$ denotes the individual owning the *periodicTripEx*.

Definition 4 (profile similarity) *Profile similarity* is a value in $[0, 1]$ assigned to a pair of individuals that indicates to what extent the individuals are compatible for carpooling (homophily concept described in [16]).

Definition 5 (pooled trip execution) A *pooled trip execution* (abbreviated by *pooledTripEx*) is the cooperative execution of a set of trips using a single car and a single driver.

Note: In this context, it is assumed that each *pooledTripEx* is driven by exactly one driver. More complex cases are covered by [11].

Definition 6 (path similarity) *Path similarity* is a value in $[0, 1]$ assigned to an *ordered* pair (pte_0, pte_1) of *periodicTripEx* that indicates to what extent the OD (Origin, Destination) pairs involved in the respective trips, are compatible for carpooling in case the owner of pte_0 is assigned to be the driver.

Notes:

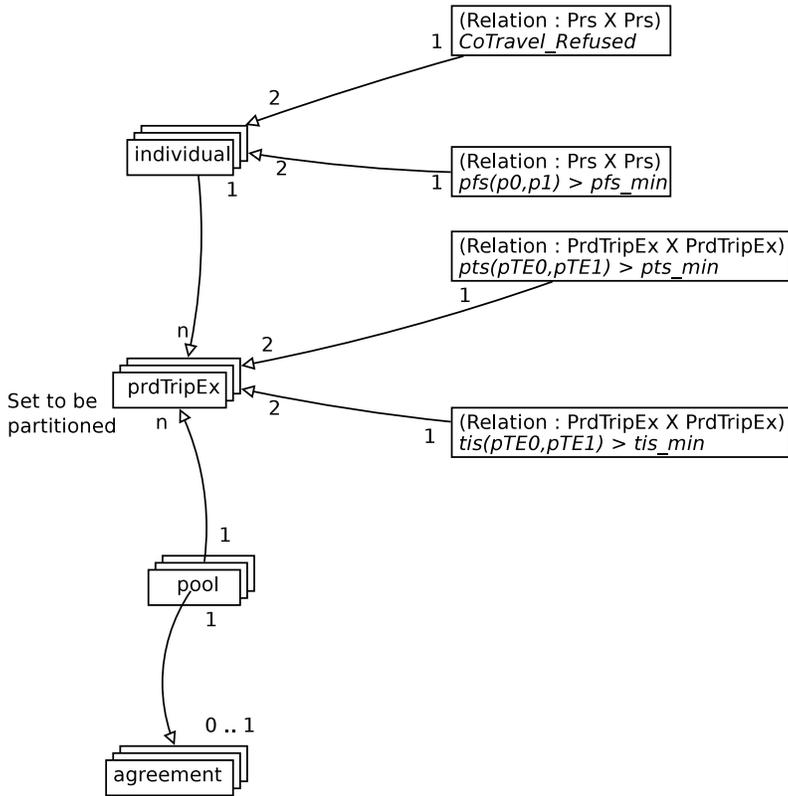


Fig. 2 Overview of sets (*individual*, *periodicTripEx* and *agreement*) used in the model and relations those sets are involved in.

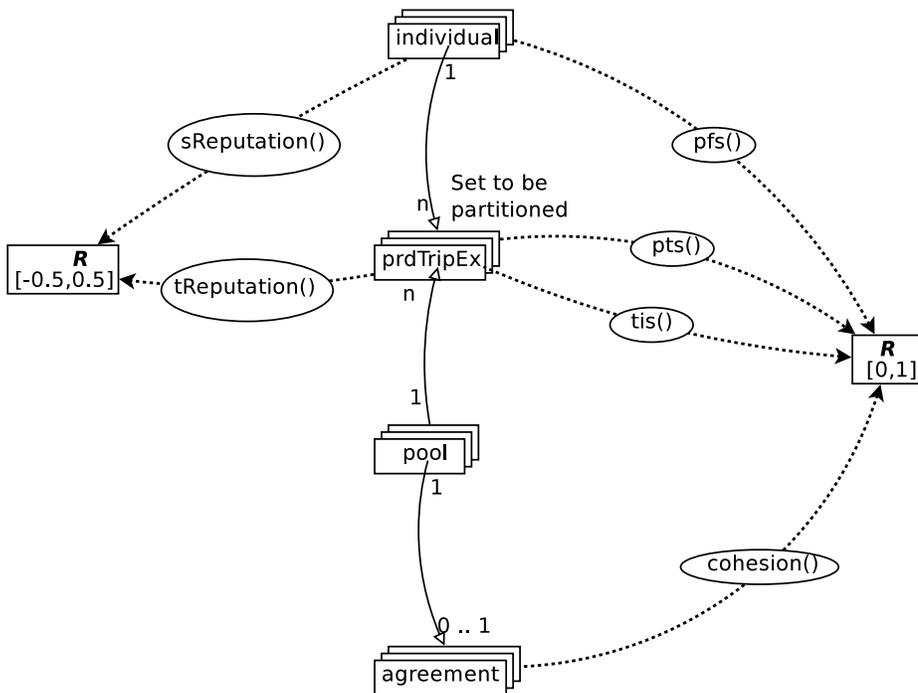


Fig. 3 Overview of functions defined on the sets (*individual*, *periodicTripEx* and *agreement*) used in the model and functions those sets are involved in. Continuous lines represent references, dashed lines represent functions.

1. Path similarity defines a function of *periodicTripEx* that is not symmetric in its arguments. This is easily seen because the distance driven depends on the driver selection; the driver needs a detour to pick up passengers.
2. When the *single driver* constraint in *pooledTripEx* is dropped, paths driven no longer are strings of path segments but consist of *join* and *fork* trees which requires a more advanced concept of path similarity which has been discussed in [11].

Definition 7 (tiSim) Time interval similarity (*tiSim*) is a value in the range $[0,1]$ assigned to a pair of time intervals specified by different people, that indicates to what measure the intersection of the intervals can be used for a specific act of cooperation.

Definition 8 (depArr_tiSim) Departure/arrival time interval similarity (*depArr_tiSim*) is a value in $[0, 1]$ assigned to an ordered pair (pte_0, pte_1) of *periodicTripEx* having identical origins and identical destinations; it indicates to what extent the time intervals involved are compatible for carpooling.

Notes:

1. Compatibility for car pooling requires a minimal amount of intervals overlap (see Fig. 4).
2. Due to the *single driver* constraint (see definition 5, the route for each passenger's trip shall be included in the route for the *pooledTripEx* which is the route for the driver.
3. Time interval similarity can be calculated only for a pair consisting of the passenger trip and the part of the driver's trip for which the route coincides with the passenger trip route (because identical origins and destinations are required).

Definition 9 (sReputation) Safety reputation is a value in the range $[-0.5, 0.5]$ assigned to an *individual* (by the passengers) to qualify the individual as a safe driver.

Notes:

1. *sReputation* is a characteristic of an individual because it is assumed that safe driving does not depend on the periodic trip driven.
2. The initial value for individual *i* is *i.sReputation* = 0.0 (which means *neutral*).

Definition 10 (tReputation) Timeliness reputation (or accuracy reputation) is a value in the range $[-0.5, 0.5]$ assigned (by the co-travellers) to a *periodicTripEx* in an *agreement*: it indicates to what measure the owning individual respects the timing when executing the periodic trip in the *agreement*.

Notes:

1. *tReputation* is defined for both drivers and passengers.
2. *tReputation* has been defined as a characteristic of a tuple $(periodicTripEx, agreement)$ and not as a characteristic of an *individual* or of a *periodicTripEx* because an individual can behave differently on a specific *periodicTripEx* *pte₀* in different agreement contexts (pools). Example: the interval between a pick-drop activity to be executed by individual *i₀* and the start of the *periodicTripEx* in a given agreement *a₀* is too short so that it is difficult for *i₀* to meet the timing requirements of *a₀*. Within a different agreement *a₁* timing constraints for *pte₀* can be less severe so that the owner can meet them easily.
3. The initial value for *periodicTripEx* *pte* in *a* is given by $(pte, a).tReputation = 0.0$ (which means *neutral*)

Definition 11 (cohesion) Cohesion qualifies the strength of an *agreement* using a value in $[0, 1]$ that is a function of attributes of the *agreement* only.

Notes:

1. An *agreement* with a high *cohesion* value is less likely to be broken whenever some of its *periodic-TripEx* get a proposal to set up a new cooperation. Cohesion determines the resistance to breakdown in case opportunities for recombination come available.
2. *Cohesion* does not depend on *sReputation* since that is an attribute of an *individual* and not of an *agreement*.
3. The matcher shall derive *cohesion* from individual's negotiation feedback since individuals are assumed not to be prepared to specify and maintain *cohesion* values; furthermore, they are unable to do so since no universally valid scale or method is available.

Note: In order for carpooling (co-travelling) to be suitable, both the collective departure and arrival time intervals shall be suitable for each participant.

3.3 Exploration/advisory and Negotiation Phases

Matching is applied in both *local* and *global* exploration phases. In both cases, matching precedes the negotiation phase where final decisions to carpool are taken. Mechanisms used in the exploration/advisory phases shall be consistent with mechanisms in the negotiation phase. It is not possible to predict the negotiation phase with certainty; reasons are:

1. Negotiation covers driver selection, co-route determination and re-scheduling (daily planning adaptation) for the cooperators. Schedule adaptation makes use of *VOT* (individual specific *Value Of Time*). An advisory mechanism does not have all required data available nor has any knowledge of the private goals (and in general the *BDI* (Beliefs, Desires, Intentions)) of the individuals (agents) involved in a negotiation. Individuals need to adapt their daily agenda because the decision to carpool introduces mutual dependency and hence additional constraints. On one hand, those constraints induce computational complexity at two levels: (i) agenda adaptation for a particular individual, includes re-timing, re-location, re-sequencing (combinatorial optimization) of activities along with activity dropping and replacement [8], [9],[13],[3] and (ii) networked individuals need to take care of the agreements they are involved in. On the other hand, constraints induced by mutual agreements cannot be considered to be fixed; under certain circumstances they evolve over time. This induces the need for replanning. Relevant literature has been mentioned in 2. Since planning is involved, agents need to predict the near future. [13] predicts future travel times using perception filters that actually implement part of the belief of an agent.
2. The total distance driven cannot be predicted by the *matcher* when carpool parkings are involved because in such cases the co-route can be tree structured: see note 2 for definition 6. Hence the path similarity function delivers only an approximation of the one involved in negotiation.
3. People are assumed to be prepared posting a minimal amount of data about the time intervals that suit them for departure and arrival respectively; candidates are supposed to specify just the interval boundaries. However, during negotiation, they can make use of *preferences* to state that one of a set of proposed intervals suits better than another one. Hence, the *trip times interval similarity* function available to the matcher is only an approximation for the one used during negotiation (see Fig. 5).
4. The *Cotrip_Refused* relation shown in Fig. 2 allows individuals to unconditionally avoid any advise to carpool with specific people. For privacy reasons, it is not possible for a refused individual to know the refusing party.

3.4 Principle of Operation of the Carpooling Model

1. An individual looks for other individuals to cooperate while executing *periodicTripEx*'s: this is called *exploration*.
2. *Local* exploration within the private social network (*PrivNet*) is applied before *global* exploration. If *carpool candidates* can be found within an individual's *PrivNet*, they will be contacted first (as preferred candidates).
3. *Global* exploration is applied only in a second stage when no suitable *pool* was found in the *PrivNet*. In the *Global* exploration phase, the *matcher* provides advice about which *pools* an individual should negotiate with. This corresponds to the use of an online service by a candidate exploring the set of formerly unknown carpooling candidates.
4. If an individual joins a *pool*, (s)he is added to the *PrivNet* for all other participants in the *pool* (if still required) so that if i_0 and i_1 cooperate in a *pool*, (i_0, i_1) and (i_1, i_0) belong to each others *PrivNet*. Because links never are removed from the *PrivNet*, if i_0 and i_1 ever carpooled, $(i_1) \in PrivNet(i_0, 1) \wedge (i_0) \in PrivNet(i_1, 1)$.
5. Candidates register, join and leave *pools* at random moments in time. As a consequence the main data structures dynamically change due to events external to the matching process.

4 Functions related to Domain Concepts

A specific concept can be implemented by different functions in *MAS* and *GCPMS* matcher.

4.1 Time Interval Based Functions

Table 1 contains a summary of the functions presented in this section. Two kinds of function are used. The *tiSim* and *tiSuitFunc* functions apply to a set of two intervals of the same kind (i.e. both are departure or both are arrival intervals) that correspond to two individuals considering to cooperate. The *depArr_tiSim* and *depArr_tiSuitFunc* are defined over 4 time intervals (i.e. both departure and arrival intervals for two people considering to cooperate).

For the reason mentioned in Section 3.3 item 3, different time interval similarity functions are used in respectively the agent-based exerciser and the matching operational service.

Remember that the trips considered shall have identical origins and destinations respectively (hence the time

Function type	Context	
	MAS (function of time)	GCPMS (constant)
2 time intervals of same kind (either <i>departure</i> or <i>arrival</i> time intervals for 2 agents)	tiSuitFunc	tiSim
2 pairs (<i>departure</i> and <i>arrival</i>) of time intervals for 2 agents are considered	depArr_tiSuitFunc	depArr_tiSim

Table 1 Summary of the time interval functions used.

intervals stated by the participants shall apply to a single origin-destination pair which implies that the passenger trip embedded in the driver's trip is to be considered: see note 2 with definition 8).

4.1.1 Time Interval Based Functions for Negotiation

1. This section define the *tiSuitFunc* (time interval suitability) and *depArr_tiSuitFunc* (departure/arrival time interval suitability) functions.
2. The departure (arrival) interval for a trip (*periodic-TripEx*) is the time interval that suits the traveller to start (end) the trip. Let $pte.i_d()$ and $pte.i_a()$ denote respectively the departure and arrival intervals of the *periodicTripEx* pte .
3. Individual p_0 's *preference* for a given moment in time is given by the function $f_{p_0} : \mathbb{R} \Rightarrow \mathbb{R} : t \mapsto f_{p_0}(t) \in [0, 1]$. The function is not required to be differentiable or continuous but the product of two such functions shall be integrable. For each moment in time belonging to the departure and arrival intervals, the *preference* value needs to be specified.
4. The *combined preference* function is the product of the preference functions associated with two *periodic-TripEx*'s. It is essential to the negotiation process.
5. The *time interval suitability* is the integral of the combined preference over a fixed time interval. The length of the interval has a pre-specified constant C value; a suitable choice is the expected duration of the trip interruption to get someone on/off board of the vehicle. Let $t_{i_X,0}$ and $t_{i_X,1}$ denote the begin en end times for a time interval specified by agent X . The *time interval suitability* is denoted by $S(C, i_A, f_A, i_B, f_B)$, where $i_A = [t_{i_A,0}, t_{i_A,1}]$ and $i_B = [t_{i_B,0}, t_{i_B,1}]$ are intervals specified by individuals A and B ; f_A and f_B are the associated preference functions. The suitability function is given by

$$t_0 = \max(t_{i_A,0}, t_{i_B,0}) \quad (2)$$

$$t_1 = \min(t_{i_A,1}, t_{i_B,1}) \quad (3)$$

$$S(t; C, i_A, i_B, f_A, f_B) = \begin{cases} \frac{1}{C} \int_t^{t+C} f_A(x) \cdot f_B(x) dx & \text{if } t \in [t_0, t_1 - C] \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where t denotes the start of the boarding/alighting operation. The dimension of the *time interval suitability* value is $[prefUnit^2]$. In this context, preference is assumed to be dimensionless, hence the suitability is dimensionless. During negotiation, $S(t; C, i_A, i_B, f_A, f_B)$ is used to find a suitable time to board/alight.

6. Piecewise linear functions are used because they are flexible, they can easily be specified by the user (in charge for the configuration of the agent-based model) and integration is computationally cheap. An example is shown in Fig. 5. The left hand part shows piecewise linear *preference* functions, their product and the associated time interval suitability (proportional to the crosshatched area under the product function).
7. Time interval *suitability* is a value in $[0, 1]$.
8. The departure/arrival time interval suitability *depArr_tiSuitFunc* is defined as

$$f(t) = S_{dep}(t; C, i_{A,dep}, i_{B,dep}, f_{A,dep}, f_{B,dep}) \cdot S_{arr}(t + d; C, i_{A,arr}, i_{B,arr}, f_{A,arr}, f_{B,arr}) \quad (5)$$

where d is the expected trip duration.

4.2 Time Interval Similarity Evaluation for Matching

1. It is not feasible to ask the individuals to register the piecewise linear preference function mentioned in Section 4.1.1. People are assumed to be prepared to register simply a time interval only. Hence the *preference* value is assumed to be a constant f over the time interval specified. The right hand part in Fig. 5 shows case for the same intervals where the preference function is assumed to equal one everywhere: this is the assumption made by the matching service due to lack of information: the user only specifies the boundaries for the departure and arrival intervals.
2. The negotiation outcome is assumed to be positively correlated with the length of the intersection of the intervals associated with the *periodic-TripEx*'s to compare. The value is not compared to the constant C mentioned above because this comparison would only imply a linear scaling of an independent

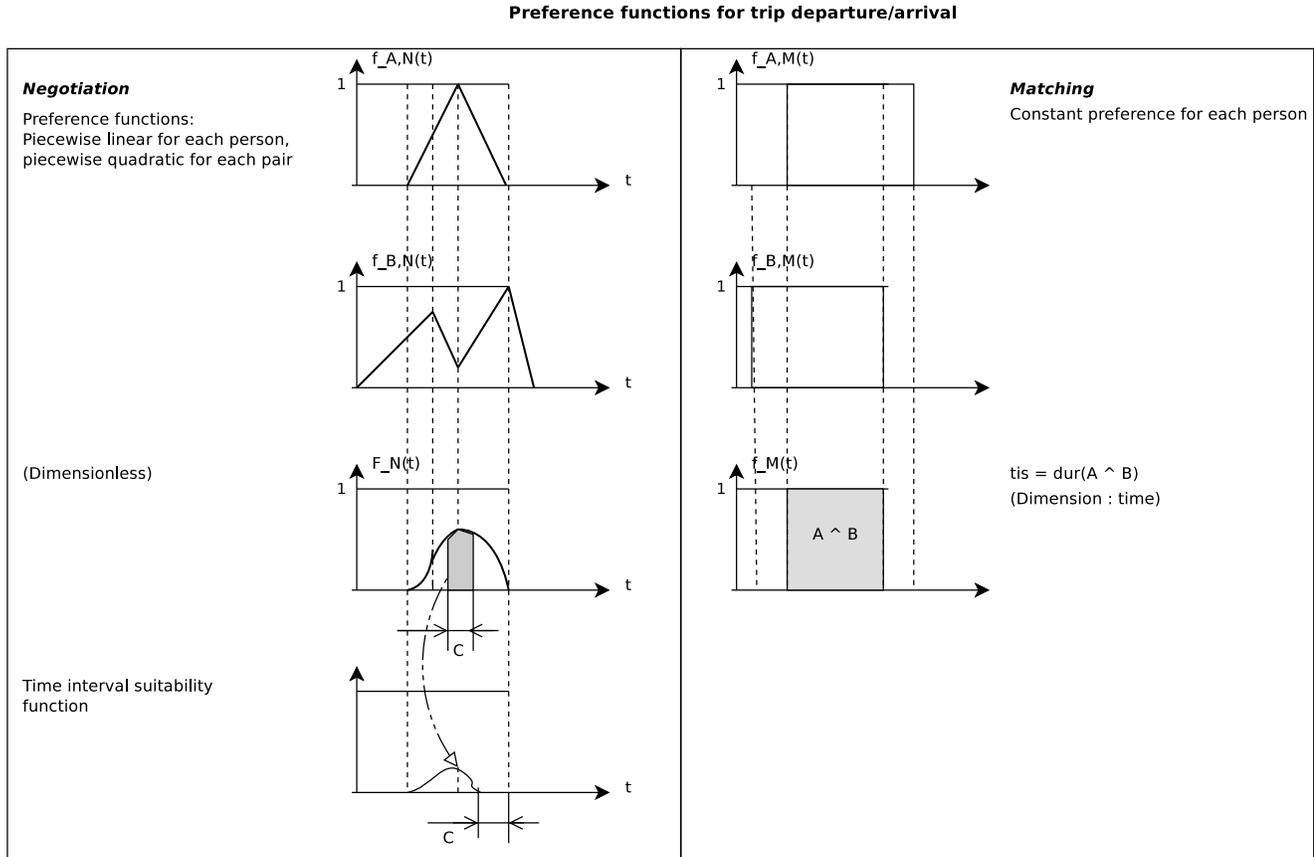


Fig. 5 (Left) Time similarity used while negotiating: $f_{A,N}(t)$ and $f_{B,N}(t)$ are time preference functions for specific intervals. $F_N(t)$ is the *combined preference* and the size of the cross-hatched area is the resulting *time interval suitability function*. (Right) Time similarity used by the matcher: all preference functions equal 1 because users are expected to only submit feasible time intervals.

variable which has no effect on the *logit* estimator.
The time interval similarity $tiSim$ is given by

$$t_0 = \max(t_{i_A,0}, t_{i_B,0}) \quad (6)$$

$$t_1 = \min(t_{i_A,1}, t_{i_B,1}) \quad (7)$$

$$tiSim(i_A, i_B) = t_1 - t_0 \quad (8)$$

- For a given pair of *periodicTripEx*'s, $tiSim$ values are fed into the *logit* estimator as two independent variables; combining them into a single value would cause a loss of information.

4.3 Path Similarity

4.3.1 Path Similarity in MAS

The first version of the agent-base exerciser takes does not take carpool parkings into account. As a consequence, path similarity is calculated in the same way as for the *GCPMS*.

4.3.2 Path Similarity in GCPMS

- The *GCPMS* per hypothesis has no information about carpool parkings potentially being used (because that is not specified by the candidates). Therefore, it is assumed that people board and alight at home and work locations only.
- The owner of the first *periodicTripEx* is the driver.

Let
 O_i, D_i

denote respectively the *origin* and *destination* locations for individual i (e.g. home and work locations)

$r(a, b, t)$

denote the route from a to b when starting at time t that is optimal with respect to some cost function $c(r)$ based on distance and travel time

$d(r, t)$

denote the duration to travel the route r starting at time t

$l(r, t)$

denote the duration to travel the route r starting at time t

$c(r)$	denote a cost function based on route length $l(r, t)$ and route travel duration $d(r, t)$
$p_{i,solo}(O_i, D_i), t$	denote the optimal path from O_i to D_i when individual i drives alone (solo) and <u>starts</u> at time t
$\bar{p}_{i,solo}(O_i, D_i), t$	denote the optimal path from O_i to D_i when individual i drives alone (solo) and <u>ends</u> at time t
$p_{i,carpool}(O_i, D_i), t$	denote the optimal path from O_i to D_i when individual i drives the carpool trip via O_j and D_j for $i \neq j$ and <u>starts</u> at time t
$\bar{p}_{i,carpool}(O_i, D_i), t$	denote the optimal path from O_i to D_i when individual i drives the carpool trip via O_j and D_j for $i \neq j$ and <u>ends</u> at time t
$pathSim_d()$	denote the path similarity function for the case where the earliest departure is given
$pathSim_a()$	denote the path similarity function for the case where the latest arrival is given

- Note that the minimal cost depends on the start moment in time. Also note that the optimal path can be determined either by fixing the time for the first departure or by fixing the time for the latest arrival.
- The ratio between the lengths of the optimal routes for the driver is used as a *path similarity function*. For the *given earliest departure* case (starting at t_0) where A is the driver and the trip is $O_A \rightarrow O_B \rightarrow D_B \rightarrow D_A$, this leads to

$$t_1 = t_0 + d(r(O_A, O_B, t_0)) \quad (9)$$

$$t_2 = t_1 + d(r(D_B, D_A, t_1)) \quad (10)$$

$$pathSim_d(pte_A, pte_B, c()) = \frac{c(O_A, D_A, t_0)}{c(O_A, O_B, t_0) + c(O_B, D_B, t_1) + c(D_B, D_A, t_2)} \quad (11)$$

Note that t_1 denotes the time at which the carpool trip leaves O_B and t_2 denotes the time at which the carpool trip leaves D_B . Also note that in general

$$pathSim_d(pte_A, pte_B, c()) \neq pathSim_a(pte_A, pte_B, c()) \quad (12)$$

since the departure times can differ. Finally note that in general

$$pathSim_a(pte_A, pte_B, c()) \neq pathSim_a(pte_B, pte_A, c()) \quad (13)$$

since the routes differ.

- The departure time can have a large effect on the trip duration. In the first GCPMS this dependency is ignored due to lack of data. Because of the availability of speed profiles registered using GPS navigators, it will become feasible to take the time dependency into account (which will lead to more accurate negotiation outcome prediction) in the near future although that will require a large amount of data pre-processing and data storage. By ignoring time dependency, the equation 11 is reduced to

$$pathSim_d(pte_A, pte_B, c()) = \frac{c(O_A, D_A)}{c(O_A, O_B) + c(O_B, D_B) + c(D_B, D_A)} \quad (14)$$

4.4 Profile Similarity

The candidate carpooler specifies the value for a set of N_A attribute values: those constitute the candidate's *profile*.

- The attributes can be: (a) continuous variables limited to a finite interval (b) discrete quantities for which a total order relation exists, also limited to a finite interval and (c) enumerations (discrete quantities without an intrinsic order relation).
- Attributes that are ordinal values (cases (a) and (b)) are handled in the same way; case (a) is expected not to occur in practice. The domain is mapped onto $[0, 1]$. The distance between two attribute tuples a_0 and a_1 having N_{OA} ordinal attributes, is the Euclidean distance divided by a scale factor to normalize the distance (map to interval $[0, 1]$).

$$d(a_0, a_1) = \sqrt{\frac{\sum_{i \in [1, N_{OA}]} (a_0[i] - a_1[i])^2}{N_{OA}}} \quad (15)$$

Continuous variables are combined into a single distance value d_C and discrete ordinal values are combined into another one d_D . The range of d_D is a finite subset of $[0, 1]$.

- All attributes have an equal weight.
- The distance d_E between two vectors constituting of enumeration variables, is the number of differing attributes divided by the total number of attributes for normalisation.

- Both *MAS* and *GCPMS* use the same mechanisms to calculate profile similarity. However, the *MAS* can make use of particular attributes that have not been registered in the *GCPMS*.

4.4.1 Profile Similarity in *GCPMS*

- The first model uses the *similarity* between two profiles as a predictor (one independent variable) for the *logit* model. It is to be investigated under what conditions it is more efficient (in terms of prediction accuracy) to feed N_A variables into the *logit* independently, each one of which is the difference between the values for a given attribute in the respective profiles.
- The similarity values $s_C = (1 - d_C)$, $s_D = (1 - d_D)$ and $s_E = (1 - d_E)$ are used as independent variables for the *logit* estimator.

4.5 Reputation

Both *sReputation* and *tReputation* are handled in the same way.

4.5.1 Reputation in *MAS*

- In the agent based exerciser a gossip based mechanism is modeled. In this case the social network is considered; the set of carpooling based connections (links) is a small subset of the social network for each individual. People keep a qualification for everyone they ever carpoled with. Furthermore, an individual can keep a qualification for another one that has been derived from *gossip*; this is a transitive mechanism. The credibility of the qualification decreases with each intermediate step involved.
- Every agent keeps list containing a perceived safety reputation value for a limited set of other agents. In the exerciser, a specific agent can be qualified by zero or more safety reputation values (each one of which is owned by a peers). Every agent can determine the reputation of another agent using a method that is not specified in this section and overriding the value already in place (if any). Furthermore, everyone can adjust the reputation of peers based on *gossip* as follows. At a random moment in time, an emitter agent a_e can multicast its reputation value $R^e(q)$ to qualify agent a_q to a subset agents directly connected to it in the social network. The receiver ar
 - retransmits the reputation message with a given probability p_r (hence simply drops it with probability $(1 - p_r)$)

- adjusts its own perception of a_q with a given probability p_a (hence simply ignores it with probability $(1 - p_a)$)

Consider agents a_e, a_q, ar, a_v that are all pairwise different. a_e emitted a qualification about a_q that reaches ar via its neighbour a_v . If ar did not yet have registered an opinion about a_q , the value for $R^r(q) = 0$. Reputation update by receiver a_a is done by

$$\alpha = 2^{-d(e,r)} \quad (16)$$

$$\beta_{r,v} \in [0, 1] \quad (17)$$

$$R_q^{r'} \leftarrow \frac{R^r(q) + \alpha \cdot \beta \cdot R^e(q)}{1 + \alpha \cdot \beta_{r,v}} \quad (18)$$

where $d(e,r)$ is the distance between emitter and receiver in the network and $\beta_{r,v}$ is the strength of the link between a_r and a_v .

4.5.2 Reputation in *GCPMS*

- The *GCPMS* allows for controlled mutual evaluation of individuals with respect to timeliness and safety. Only individuals cooperating in an agreement can qualify each other; this means that the reputation mechanism in the matching service is not transitive.
- Similar mechanisms are for *sReputation* and *tReputation*. Qualifications received are registered in a dedicated *qualifications list* assigned to the entity they apply to; for each issuer, only the most recent qualification is kept. *sReputation* qualifications are registered with individuals and *tReputation* qualifications are registered with *periodicTripEx*.
- Each individual has reputation values that evolves over time due to qualification by cooperators (i.e. individuals who participated in an agreement with the person being evaluated). Passengers can qualify *sReputation* for drivers. Every cooperator can qualify every other cooperator's *periodicTripEx*'s with respect to *tReputation*.
- The attributes for a qualification are: (i) the generation timestamp $q.ts()$, (ii) the issuer $q.iss()$ and (iii) the specified value $q.rep()$.
- In both cases, the reputation is calculated as a weighted average of the values posted in the qualification list: the weight decreases with age of the qualification and increases with the duration of the cooperation (the agreement lifetime) $a.dur(q.ts())$ up to the moment of qualification. Note that the cooperation duration in the case of *sReputation* is to be summed over a set of agreements. Let \mathcal{Q}_{i_0} be the qualifications list for individual i_0 . Let $a_j^{i_0}$ denote an agreement in which individuals i_0 and j cooperated. Let

and $a_y^x.dur(t)$ denotes the lifetime of agreement between the qualified agent x and the qualifying agent y at time t . Let A_x^y denote the set of all agreements in which individual x and y cooperate. Then following equations determine the *sReputation*:

$$age = now - q.ts() \quad (19)$$

$$coopDur = \sum_{a \in A_{q,iss()}} a.dur(q.ts()) \quad (20)$$

$$w_{q,iss()}^{i_0} = \exp(-\alpha \cdot age) \cdot (1 - \exp(-\beta \cdot coopDur)) \quad (21)$$

$$sReputation_{i_0} = \frac{\sum_{n \in \mathcal{Q}_{i_0}} q.sRep() \cdot w_{q,iss()}^{i_0}}{\sum_{n \in \mathcal{Q}_{i_0}} w_{q,iss()}^{i_0}} \quad (22)$$

4.6 Cohesion of an Agreement

1. *Cohesion* is supposed to be a monotonically decreasing function of the time t elapsed since the creation of the *agreement*. *Cohesion* is a monotonically decreasing function of *pool* size s (large *pools* are more likely to disintegrate). Note that cohesion does not depend on mutual evaluation of carpoolers; cohesion and reputation shall be independent concepts because all of them are fed into a probability estimator. The *cohesion value* is given by:

$$c = e^{\alpha \cdot t} \cdot e^{\beta \cdot (s-1)} \quad (23)$$

2. In the pairwise case, when considering a specific edge, exactly two *cohesion values* apply (one for each of the vertices (*periodicTripEx*'s)). Each of the cohesion values possibly applies to an *agreement*. For a given *periodicTripEx* pair (pte_0, pte_1), the c_0 and c_1 can relate to either different *agreements* or to a single one. The meaning of the tuple (c_0, c_1) depends on number of *agreements* involved. Therefore, *cohesion values* are combined into a single *cohesion based indicator* using the function given in equation 26; the second case in equation 26 corresponds to the case where both *periodicTripEx* are members of an *agreement* (but not necessarily to the same one). Let $pte_0, pte_1 \in \mathcal{T}$ the *periodicTripEx*s involved. Let c_0 and c_1 denote the respective corresponding *cohesion values* and $p.T()$ denote the list of *periodicTripEx* involved in pool p . Let $pte.a()$ denote the *agreement* covering pte when it belongs to a *pool*. The *cohesion indicator* $\bar{c}(pte_0, pte_1)$ is a measure for the cohesion between two *periodicTripEx*'s when they already form a pair and for the feasibility

to get them released when they are bound in pairs with others.

$$pte_x.a() = \begin{cases} nil & \text{if } \nexists p \in \mathcal{P} | pte_x \in p.T() \\ p.a() & \text{if } \exists p \in \mathcal{P} | pte_x \in p.T() \end{cases} \quad (24)$$

$$c_x = \begin{cases} 0 & \text{if } pte_x.a() = nil \\ pte_x.a().c() & \text{else} \end{cases} \quad (25)$$

$$\bar{c} = \begin{cases} (1 - c_0) * (1 - c_1) & \text{if } pte_0.a() \neq pte_1.a() \\ c_0 * c_1 & \text{if } pte_0.a() = pte_1.a() \neq nil \end{cases} \quad (26)$$

In case both *periodicTripEx* belong to the same *agreement*, the cohesion values are taken from that *agreement* and in fact $c_0 = c_1$. In the other case (which also covers the case where at least one of the *periodicTripEx* is not covered by an *agreement*), the complement of the *cohesion* values is used. When neither of the *periodicTripEx* belongs to an *agreement*, $\bar{c} = 1$.

5 Weights Determination

The weights used to label the edges in the graph, are probability values associated with the success of the negotiation process between individuals. Those probabilities are calculated by means of *logistic regression* (logit) fed by results of negotiations who have been advised by the *carpoolMatcher*.

Edges are not removed from the graph when they become member of a pool (*agreement*). They are labeled with a weight in exactly the same way as *free* edges. As soon as their weight becomes lower than the weight of other edges involving the same vertices, some of the carpool members will get an advice to negotiate with non-pool-members to setup a new *agreement*.

5.1 Negotiation Outcome Prediction

Fig. 6 summarizes the data dependencies relevant to edge weight determination. From the point of view of the matcher service, the outcome of a negotiation process is a discrete variable with values : *success* (yes) and *failure* (no). Independent variables influencing the negotiation are continuous : *profSim*, *pathSim*, *tiSim*, *cohesion* and *sReputation*. A *logit* model will be used to predict the negotiation outcome. Negotiation results fed back to the *Global CarPooling Matching Service (GCPMS)* are used to determine the coefficients for the *logit* model by linear regression.

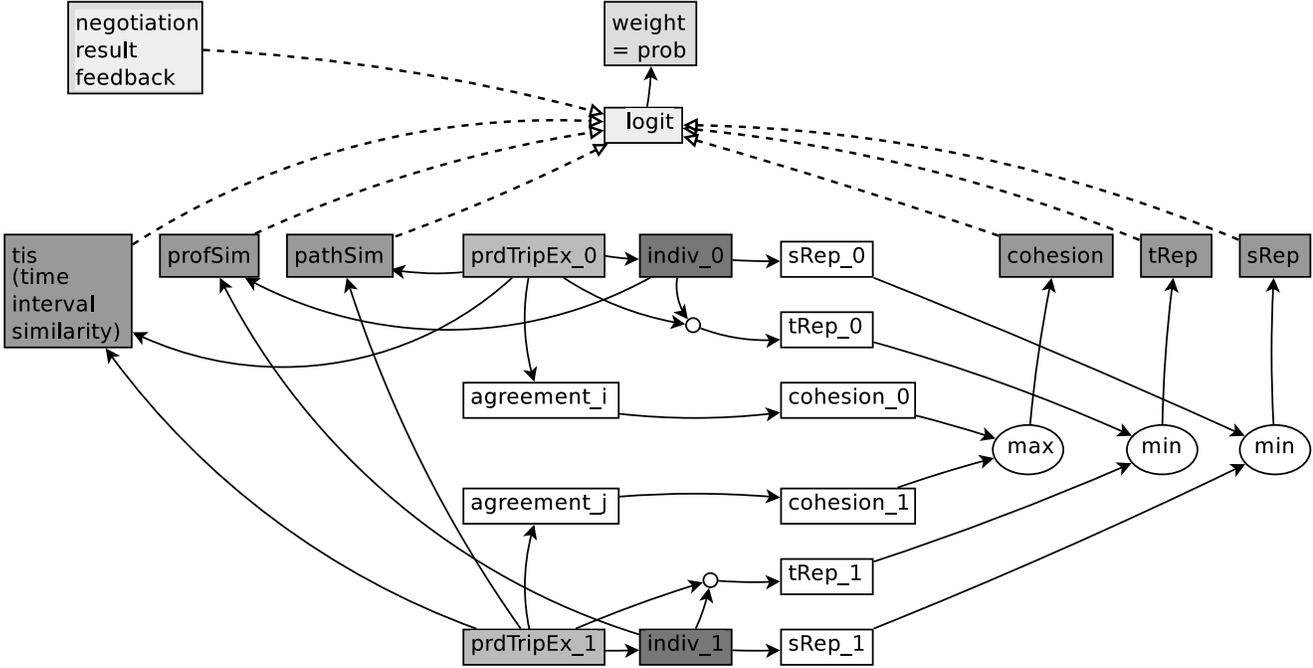


Fig. 6 Dependencies between concepts used to calculate the weight for an edge connecting two *periodicTripEx*'s

5.2 Dynamic Actor and Agreement Attributes

While evaluating the success probability for a *pool*, exactly one *sReputation* value applies since only the *sReputation* for the driver is relevant.

The *tReputation* of an individual i_0 applies to an existing *agreement* and only exists as long as the agreement holds. It can only be affected by the partners in the agreement different from i_0 (in the pairwise case, there is only one such partner). The *tReputation* is an evaluation score assigned by the partners; the default value equals the neutral value : see definition 10.

6 Matching using dynamically updated Edge Weights

We now show how to apply graph matching techniques to find optimal carpooling (see also [2]). We are also interested in quickly updating the result, when the weights of the edges (which correspond to the result of negotiation between the users) slightly change (as opposed to running the entire matching from scratch whenever such change occurs). We proceed to introduce the graph structure. In this paper, we deal with the case in which the *periodicTripEx*'s belong to either *drivers* or *non-driving passengers*, and we try to optimally match each driver trip with a passenger trip; in the future we plan to tackle more general problems. Note that this case requires the set of candidate carpoolers to be partitioned a priori.

We quickly introduce some notations: a graph G consists of a set of nodes, V , and a set of edges, E , such that each edge is associated with a pair of nodes. Denote $G = (V, E)$. A directed graph is the same as above, but where each edge is associated with an ordered pair of nodes. In a weighted graph, each edge has an associated non-negative real number with it, defined as its weight.

Given a graph $G = (V, E)$, a matching M in G is a set of pairwise non-adjacent (disjoint) edges. That is, no two edges share a common node. The weight of a matching M is the sum of the weights of the edges in M . A maximum (optimal) matching is a matching such that it obtains the maximum weight of all matchings. It does not have to be unique.

A special case of the matching problem is when the graph is *bipartite*, that is, its nodes can be partitioned into two disjoint sets, L and R , such that all edges are between a node in L and a node in R . For the carpooling problem, this may correspond to the case in which the *periodicTripEx* set is composed of trips owned by drivers (L) and trips owned by non-drivers (R). For a schematic example, see Fig. 6.

In order to solve the carpooling problem when represented by a bipartite graph, denote the weight of the edge between $i \in L$ and $j \in R$ by c_{ij} , and define variables x_{ij} . To find an optimal matching, solve the following optimization problem: maximize $\sum_{ij} c_{ij} x_{ij}$, subject to the constraints $x_{ij} \geq 0$, $\forall j \sum_i x_{ij} \leq 1$, $\forall i \sum_j x_{ij} \leq 1$. According to Section 5, the c_{ij} values are the proba-

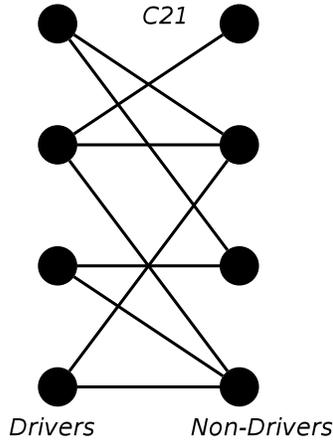


Fig. 7 A schematic example of a graph used for the carpooling problem. c_{21} stands for the weight of the match between trip owned by driver number 2 and trip owned by passenger number 1.

bilities for the negotiation to succeed when i and j are advised to carpool.

To test the algorithm, data was simulated as follows. Weights were chosen as the absolute values drawn at random from a normal distribution with mean $\mu = 0$ and given standard deviation σ , and then those smaller than a certain threshold (0.2 for the experiments reported here) were thresholded to 0. Fig. 8 shows run times for different numbers of nodes and values of σ .

As discussed in the introduction, the input to the carpooling problem is highly dynamic. This necessitates developing, in addition to the well-studied batch solution described above, algorithms which are *incremental*. An incremental algorithm assumes that the optimal solution of the carpooling for some input was computed already, and then it attempts to solve (either accurately or approximately) the problem for the same input but under a small perturbation.

Here we present an analysis which, given a solution to the optimization problem, allows to determine how far this solution is from the optimal one to the perturbed problem. Assume that the solution for the original (unperturbed) problem, with weights c_{ij} , is x_{ij}^0 , and that x_{ij}^1 is the solution for the perturbed weights $c_{ij} + \epsilon_{ij}$. Then the following holds:

$$\sum_{i,j} (c_{ij} + \epsilon_{ij}) x_{ij}^1 = \sum_{i,j} c_{ij} x_{ij}^1 + \sum_{i,j} \epsilon_{ij} x_{ij}^1 \leq \quad (27)$$

$$\sum_{i,j} c_{ij} x_{ij}^0 + \sum_{i,j} \epsilon_{ij} x_{ij}^1 \quad (28)$$

where the inequality follows from the optimality of x_{ij}^0 for the unperturbed weights c_{ij} . Hence, the solution for the perturbed weights is better than the old by an

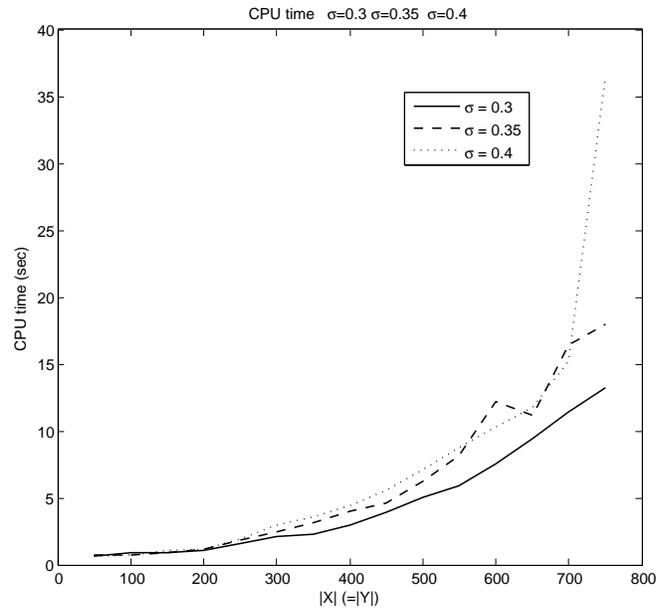


Fig. 8 Running times [sec] for various values of σ and numbers of nodes.

error term of at most $\sum_{i,j} \epsilon_{ij} x_{ij}^1$. Since $\forall j \sum_i x_{ij}^1 \leq 1$ and $\forall i \sum_j x_{ij}^1 \leq 1$, this term can be bounded by $\sum_i \max_j |\epsilon_{ij}|$.

7 Future Research Directions

7.1 Ongoing research

Network characteristics are studied for the graph connecting the trips that have been posted for carpooling in order to find out how partitioning (independent components) depends on the probability threshold used.

The carpooling problem has been formulated in a general way. This leads to an *assignment* problem in a directed weighted graph and no longer requires a priori drivers-passengers partitioning. More than two *periodicTripEx* can be part of an agreement. Research now focuses on the complexity of the assignment problem. First results have been presented in [12] (accepted conference paper, to be presented). Although a straightforward formulation as an integer programming problem has been given, graph theoretical methods are focused in order to find out whether results from graph theory can provide solutions to mitigate the problem size explosion.

An Janus [5] based experiment using 10000 agents. It includes negotiation and trip execution simulation in order to generate reputation feedback.

7.2 Research for the longer term

Negotiation for agreements involving more than two people (some of who can already be involved in an existing agreement) requires cooperative adjustment of more than two schedules and needs additional attention.

Determination of agent profiles needs improvement; in particular semantic-based models will be evaluated to replace the basic method now used in the model.

The cohesion function used in the MAS needs to account explicitly for *private social network membership* of participants in an agreement.

Finally, in order to achieve the ultimate goal of replacing a real testers community by a community of agents in a MAS, several coefficients used in the behavioural models need to be quantified by means of surveys and appropriate statistical methods.

8 Conclusions

In order to evaluate carpooling advisory service, MAS is used as an exerciser. Large scale simulation is required in order to analyse the advisor behaviour. Agents are required to accurately mimic real individuals. Thereto the process has been analysed in depth and model components have been proposed. Preliminary experiments show that computational problems are to be expected due to combinatorial explosion. Several research paths to explore possible solutions to handle the problem have been identified.

References

1. Agatz, N., Erera, A., Savelsbergh, M., Wang, X.: Sustainable passenger transportation: Dynamic Ride-Sharing. Research Paper ERIM Report Series Reference No. ERS-2010-010-LIS, Erasmus University of Rotterdam Erasmus Research Institute of Management (2010)
2. Agatz, N., Erera, A.L., Savelsbergh, M.W., Wang, X.: Dynamic ride-sharing: a simulation study in metro atlanta. In: *Procedia - Social and Behavioral Sciences*, vol. 17, pp. 532–550 (2011)
3. Arentze, T., Pelizaro, C., Timmermans, H.: Implementation of a model of dynamic activity-travel rescheduling decisions: an agent-based micro-simulation framework. In: *Proceedings of CUPUM 05, Computers in Urban Planning and Urban Management*, London (2005)
4. Chun, H.W., Wong, R.Y.: N* - an agent-based negotiation algorithm for dynamic scheduling and rescheduling. *Advanced Engineering Informatics* (17), 1–22 (2003). URL www.elsevier.com/locate/aei
5. Cossentino, M., Gaud, N., Galland, S., Koukam, A.: ASPECS: an agent-oriented software process for engineering complex systems how to design agent societies under a holonic perspective. *Auton Agent Multi-Agent Syst* (Springer Science+Business Media) **20**, 260304 (2010). DOI 10.1007/s10458-009-9099-4
6. Guo, J., Nandam, S., Adams, T.: A data collection framework for exploring the dynamic adaptation of Activity-Travel decisions. TRB (Transportation Research Board), Tampa, Florida (2012)
7. Iwan, L., Safar, M.: Pattern mining from movement of mobile users. *Journal of Ambient Intelligence and Humanized Computing* **1**(4), 295–308 (2010). DOI 10.1007/s12652-010-0024-0. URL <http://dx.doi.org/10.1007/s12652-010-0024-0>
8. Joh, C.: Modeling individuals ActivityTravel rescheduling heuristics : Theory and numerical experiments. Transportation Research Board of the National Academies (1807 Paper 02-2173), 16 – 25 (2002)
9. Joh, C.: Measuring and predicting adaptation in multi-dimensional activity-travel patterns. PhD thesis, TUE, Eindhoven (2004)
10. Kamar, E., Horvitz, E.: Collaboration and shared plans in the open world: Studies of ridesharing. In: *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence* (2009)
11. Knapen, L., Keren, D., Yasar, A., Cho, S., Bellemans, T., Janssens, D., Wets, G.: Analysis of the co-routing problem in agent-based carpooling simulation. In: *Procedia Computer Science. Niagara Falls* (2012)
12. Knapen, L., Keren, D., Yasar, A., Cho, S., Bellemans, T., Janssens, D., Wets, G.: Estimating scalability issues while finding an optimal assignment for carpooling. In: *Procedia Computer Science. Procedia Computer Science, Elsevier, Halifax, Nova Scotia, Canada* (2013)
13. Knapen, L., Muhammad, U., Bellemans, T., Janssens, D., Wets, G.: Framework to evaluate rescheduling due to unexpected events in an Activity-Based model. In: *TRB 2013 Annual Meeting. Washington, D.C.* (2012)
14. Luetzenberger, M., Masuch, N., Hirsch, B., Ahrndt, S., Albayrak, S.: Strategic behaviour in dynamic cities. In: D. Weed (ed.) *Proceedings of the 43-rd Summer Computer Simulation Conference, The Hague, The Netherlands*, pp. 148–155 (2011)
15. Manzini, R., Pareschi, A.: A Decision-Support system for the car pooling problem. *Journal of Transportation Technologies* (2), 85–101 (2012). DOI doi:10.4236/jtts.2012.22011
16. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: Homophily in social networks. *Annual*

- Review of Sociology **27**(1), 415–444 (2001). DOI 10.1146/annurev.soc.27.1.415
17. Nijland, L., Arentze, T., Borgers, A., Timmermans, H.J.: Individuals' activity-travel rescheduling behaviour: experiment and model-based analysis. *Environment and Planning A* **41**, 1511–1522 (2009). DOI doi:10.1068/a4134
 18. Papandrea, M., Giordano, S.: Location prediction and mobility modelling for enhanced localization solution. *Journal of Ambient Intelligence and Humanized Computing* pp. 1–17 (2013). DOI 10.1007/s12652-013-0175-x. URL <http://dx.doi.org/10.1007/s12652-013-0175-x>
 19. Ronald, N.: Modelling the effects of social networks on activity and travel behaviour. PhD thesis, TUE, Eindhoven (2012)
 20. Trasarti, R., Pinelli, F., Nanni, M., Giannotti, F.: Mining mobility user profiles for car pooling. In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '11*, pp. 1190–1198. ACM, New York, NY, USA (2011). DOI <http://doi.acm.org/10.1145/2020408.2020591>. URL <http://doi.acm.org/10.1145/2020408.2020591>
 21. Varrentrapp, K., Maniezzo, V., Sttze, T.: The long term car pooling problem on the soundness of the problem formulation and proof of NP-completeness. Technical Report AIDA-02-03, Fachgebiet Intellektik, Fachbereich Informatik, TU Darmstadt, Darmstadt, Germany (2002)
 22. Xiao, X., Zheng, Y., Luo, Q., Xie, X.: Inferring social ties between users with human location history. *Journal of Ambient Intelligence and Humanized Computing* pp. 1–17 (2012). DOI 10.1007/s12652-012-0117-z. URL <http://dx.doi.org/10.1007/s12652-012-0117-z>