Types of Agents in Peer-to-Peer Shared Ride Systems

Yun Hui Wu  
Dept. of Geomatics  
The University of Melbourne  
Victoria 3010, Australia  
y.wu21@pgrad.unimelb.edu.au

Lin Jie Guan  
Dept. of Geomatics  
The University of Melbourne  
Victoria 3010, Australia

Stephan Winter  
Dept. of Geomatics  
The University of Melbourne  
Victoria 3010, Australia

ABSTRACT
Shared ride systems match the travel demand of transport clients with the supply by vehicles, or hosts, such that the clients find rides to their destinations. A peer-to-peer shared ride system allows clients to find rides in an ad-hoc manner, by negotiating directly with nearby hosts via radio-based communication. Such a peer-to-peer shared ride system has to deal with various types of hosts, such as private cars and mass transport vehicles. Their different behaviors affect the negotiation process, and consequently the travel choices.

In this paper, we present and discuss a model of a peer-to-peer shared ride system with different types of agents. The behavior of the model is investigated in a simulation of different communication and way-finding strategies. We demonstrate that different types of agents enrich the choices of the clients, and lead to local solutions that are nearly optimal.

Categories and Subject Descriptors
I.6.4 [Simulation and Modelling]: Model Validation and Analysis; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms
Performance

1. INTRODUCTION
Movement of people in a city forms a complex system. It includes the street network and other ways of travelling, traffic rules, traffic infrastructure (e.g., traffic lights, signs) as well as the cognition, decisions and actions of intelligent, autonomous agents such as pedestrians and vehicle drivers. This complex system is burdened by more and more traffic and expanding cities. In this situation a peer-to-peer shared ride system can contribute relief to the critical situation: it enables people to negotiate in an ad-hoc manner for ride sharing, and thus, helps reducing the traffic, increases urban access, and improves the integration of different modes of transport. In such a system, pedestrian are the agents with transport demand, called clients, and vehicles, or hosts, provide the transport supply. Finding rides in an ad-hoc manner is accomplished by local negotiation between these agents via radio-based communication.

A peer-to-peer shared ride system has to deal with various types of agents, such as private cars and mass transport vehicles, or mobile and immobile clients, to cope adequately with the complexity of urban movements. The agents' different interests, capacities and behaviors affect the negotiation process, and consequently, the trips made. For example, hosts can be distinguished by their travel speed, their passenger capacity and their fare structure, and clients can be distinguished by their mobility.

In this situation a client cannot stay with a simple preference for one mode of travelling, i.e., one type of hosts. For example, in general a rushed client would prefer hosts can deliver a quick and direct trip: taxis. On the other hand, taxis can be in high demand during peak travel times and catching trams, trains or buses can be an alternative: they may travel slower but might reach the destination earlier depending on traffic. Hence, in this paper we present and discuss a model of a peer-to-peer shared ride system with different types of agents.

Agents, i.e., clients and hosts in peer-to-peer shared-ride systems have knowledge of their environment. They can collect and transmit information from/to their neighbors. Frequently agents have choices. They have preferences, various optimization criteria, such as money or time, and are able to make current optimal decisions based on their knowledge. However, for practical reasons agents have only local and current knowledge of their environment. Previous research [18] investigates the ability to make trip plans from different levels of local knowledge. It shows that a mid-range communication depth is both efficient (leading to less communication messages than for complete current knowledge) and effective (leading to travel time comparable to complete current knowledge). This investigation was based on a simulation with homogeneous hosts and an immobile client. The hypothesis of this paper is that involving other types of agents, the trips will change significantly, but mid-range communication is still both efficient and effective compared to other communication strategies.

This hypothesis will be approached by simulation. The
simulation is realized as a multi-agent system, allowing us to model and understand individual behavior of different agents. The approach requires identifying and specifying the essential aspects of an urban shared ride system, implementing them in a multi-agent system, and then running large numbers of random experiments to generate the required evidence. The model can be investigated by systematically varying the design parameters and studying the peer-to-peer shared ride system behavior.

The structure of this paper is follows. Section 2 reviews previous and related research. Section 3 discusses the types of agents in shared ride systems. Section 4 presents the design of a multi-agent simulation, and the simulation results are provided in Section 5. These results are discussed in Section 6, and Section 7 concludes the research.

2. LITERATURE REVIEW

This review consists of an overview of shared ride systems in general, specifically previous research on peer-to-peer shared ride systems in particular, this is followed by an overview of agent-based transportation simulation, the approach used in this paper. Algorithms for trip planning in dynamic environments, are also reviewed.

2.1 Shared ride systems

Shared ride systems exist in many forms and names, such as dial-a-ride, car pooling, van pooling, or find-a-ride. Shared ride systems also have various levels of technological support, such as being based simply on social convention, or using a centralized database with pre-registration and/or pre-booking via a Web interface. Of interest in our context are the types of agents involved in these services, and a particular extension of the ride sharing idea to peer-to-peer services for ad-hoc ride sharing.

Van pooling can be seen as a prearranged shared ride service between home and workplace to save up parking spaces [11]. Traditional van pooling services are organized by private companies and are not door-to-door. People with regular commuting schedules usually appoint together at a place. Vans run on prearranged times and routes according to requests. The drivers do not receive a fee while users are charged subscription cost directly to the companies. Van pooling is limited by the provider’s service area and not viable for areas or individual origins or destinations that do not have the critical mass of people using the service. New users can only participate in existing van pooling routes, or they can create a new van pooling group together with others.

Mass transportation planning systems provide trip planning based on predefined schedules and routes of means like underground, trains, buses and trams. Being government funded or subsidized, the fares are typically lower than the costs of private means of transportation. In addition to guaranteeing mobility and access for everybody, this should also encourage people to mitigate individual car traffic. However, such a shared ride is restricted to fixed time schedules and routes, which is less comfortable than many private transportation alternatives. To better satisfy users, dial-a-ride systems have been initiated. Dial-a-ride systems can offer more flexible and comfortable door-to-door rides, chiefly by commercial vehicles and taxis [5]. To utilize the vehicles’ passenger capacity, drivers can pick up other passengers before reaching the destination of the first customer. The authors implement a dynamic dial-a-ride system, which can re-optimize routes after picking up new customers during services. Therefore, this dynamic dial-a-ride system supports a many-to-many service—customers have different departures and destinations—and does not need booking in advance.

Google Ridefinder provides a real-time approach for individual users to find a ride in local areas. Users have wide choices from taxis, limousines and shuttles, which are contract companies of Google. The locations of vehicles in this service, observed by GPS and collected in a central database, are said to be less than 5 minutes old, which practically means the locations are correct within 2 – 3km. Currently, this service only works in a few metropolitan areas in the United States. Using Google Map, users can view the potential host vehicles by entering city names or addresses and call selected service providers to request a ride. But because only locations of these vehicles are provided in this interface, users do not know whether the shown host vehicles have free passenger capacity for them unless calling. What is more, the dispatching of vehicles varies from city to city and company to company, therefore the users might get alternative rides by dispatchers. This is the common disadvantage of central management shared ride services, because communication does not happen between users and drivers. Communication is duplicated between users and dispatchers, and drivers and dispatchers.

Other shared ride applications provide textual Web interfaces to attract registrations of shared ride clients and hosts, such as Ride Now!, RidePro, eRideShare, or Mitfahrzentrale. The applications are intended to provide shared ride services to the public, and are maintained by local and regional agencies with central databases. Mediated trips are usually regional or national travels, but lesser so inner-urban travels. To request or offer a ride, users (clients and hosts) need to provide their home addresses, cell phone number, email addresses and requested trip details. Then the databases match requests and offers immediately and feedback a contact list of potential shared ride hosts or clients. The choice is left to the users who can email or call their selections. Agencies need high-powered workstations, database servers and internet connectivity to run such an application. Personal computers or mobile devices with Internet connectivity are necessary as data terminals for the users. Such centralized services are restricted to pre-trip registration. High-volume real-time data updating—such as current vehicle locations, current travel plans, and current seat capacities of large numbers of vehicles in urban traffic—is not possible through a centralized service.

In contrast, a peer-to-peer shared ride system [18] enables people to negotiate in an ad-hoc manner for ride sharing.

1http://labs.google.com/ridefinder.
2http://www.ridenow.org
3http://www.ridepro.net
4http://www.e-rideshare.com
5http://www.mitfahrzentrale.de
which is suited for inner-urban rides due to the instantaneous provision. Finding rides in an ad-hoc manner is accomplished by local negotiation between these agents via radio-based communication, as applied in mobile geosensor networks. Users of this service—clients as well as hosts—have to be equipped with mobile devices that form the nodes in such a network. Since such devices are already popular for other purposes, the potential user base is much larger than for traditional shared ride services. Fortunately, the local negotiation process allows for fully scalable services, in contrast to centralized shared ride services.

2.2 Agent-based transportation simulation

Simulation is an accepted approach to investigate the behaviour of complex systems before implementation. As traffic congestion becomes a world-wide problem, effective ways of modelling and predicting traffic flow have become a research focus [1]. Burmeister et al. [4] analyze the potential of multi-agent modelling in traffic and transportation systems, which naturally characterize “geographically and functionally distributed” (p. 52) subsystems, such as traffic management, traffic guide and control, and capacity and resource management. Russel and Norvig [16] define an agent as “anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators”, and describe agent behavior mathematically by functions.

Cellular automata (CA) are popularly applied in agent-based modelling. CA arrange individual automata in a cellular space, where each cell has its state. Automata can collect information from their neighbors, and change their states according to their neighbors’ states and transition rules. Due to its simple structure, CA have proven their success in land use and urban planning. However, CA are inefficient in representing mobile agents, because cells themselves can not move [3]. Additionally, state transition is too simple to implement the negotiation processes in shared ride systems.

Benenson and Torrens [3] combine CA and multi-agent system concepts and extend them as geographic automata systems. Geographic automata systems are multi-agent systems in which agents are distributed in space, allow autonomous behavior, in particular to re-locate, and interact with each other. States of geographic automata, including their location, are influenced by their neighbors, and the behavior of automata is specified by state transition rules.

In transportation systems, pedestrians, vehicle drivers, traffic controllers, traffic lights and toll agents are all collectible. The main challenge of modelling and predicting in transport systems is human behavior, which can be unpredictable.

Nagel discusses that travelers always know where they are heading, and the strategies in simulation decide in which mode they move (walk, bus, car, bicycle and so on) and how [12]. Nagel and Marchal introduce computational techniques for multi-agent simulations in this domain [13]. Modelling issues include strategy generation, adaptation, learning, and feedback. Each agent can perform multiple strategies and memorize them. After comparing the outcome of different strategies, agents can choose a previously tried strategy. The authors also address that the advantages of agent-based modelling could be unclear in practice, due to the complexity of real world processes.

Object-oriented languages, such as C++ and Java, are suggested for such agent-based modelling. Several established agent-based simulation libraries exist that simplify modelling. Swarm is one of the popular libraries based on Objective C and has a Java wrapper. RePast is a newer Swarm-like conceptual toolkit [14]. Repast is a freely open source toolkit core in Java, while it has three implementations in Java, .Net and Python. Both approaches support to program multi-agent systems that are composed of larger numbers of agents with functions describing their behavior. RePast was used successfully for a large-scale peer-to-peer shared ride system simulation [17]. However, installing and using libraries is in itself a larger effort, and we decided to develop our system from scratch.

Object-Based Environment for Urban Simulation, OBEUS has been developed as a simplest implementation of geographic automata systems in .Net [2, 3]. It is designed for urban processes and built in a cellular automata model with transition rules in form of functions. Entities in OBEUS can be one of two types, mobile and immobile entities. In OBEUS no direct relationship is allowed between non-fixed objects. That means that OBEUS is not suitable for our simulation of locally communicating mobile agents.

2.3 Wayfinding algorithms

Finding shortest paths, in terms of some cost criteria, is the key to shared ride planning. Clients are assumed to minimize travel costs in terms of criteria such as distance, time or money. The most widely known one-to-one shortest path search is A*[9, 16]. The basic idea of A* is calculating the costs f for each intermediate node by the sum of g, the costs from start to the node, and h, an estimate of the costs from the node to the destination. The estimate h can be determined by various heuristics, which are called admissible as along as h ≤ c.

In a peer-to-peer shared ride system the shortest route is not determined once, but regularly revised, based on the actual local knowledge at different times and locations in the dynamic transportation network [18]. Hence, A* can be applied in each trip planning process, and can be admissible in each process for the actual local knowledge, but admissibility cannot be preserved throughout consecutive trip plans with their different knowledge of the transportation network. In these dynamic environments life-long planning can be applied [10, 20]. This is an adaptive A* shortest path search with a dynamic start point. Compared to static A*, lifelong planning A* achieves less visited nodes and reduces the computational cost when weights are updated within network. Their approach deals with updates that increase previously stored edge costs, whereas in our peer-to-peer shared ride system updates can decrease costs as well. Consider the worst case in shared ride systems is that clients have to walk from origin to destination, finding no ride. In this case, any

6 http://www.swarm.org
7 http://repa.st.sourceforge.net
8 OBEUS can be downloaded from http://www.geosimulationbook.com
ride from a host found during a trip will reduce their trip time.

Time geography provides tools to improve the efficiency of communication in the negotiation processes of the peer-to-peer shared ride system [19, 17]. Time geography is built on the space-time paths representing the movement of individuals in geographic space over time [7]. The space-time prism is an extension, defined by the reachable areas between start and destination of a trip. The interior of the prism is called potential path space. Dynamic space-time prisms can be achieved, demonstrating that travel times between locations vary by different spaces and times with dynamic traffic flow [21].

3. AGENTS IN SHARED RIDE SYSTEMS
For a peer-to-peer shared ride system it is essential to study the nature of its peer users, both clients and hosts, before their intentions, desires and beliefs can be modelled in a simulation. For that purpose, only the factors that are critical for the aim of the simulation need to be identified. In this research, the critical factors are the constraints on mobility and the passenger capacity.

In order to reflect better the properties of realistic shared ride systems, we identify three typical kinds of hosts and discuss their distinct economic and operational characteristics in this section. They are mass transport, taxi cabs and private cars. We start by characterizing different types of clients.

3.1 Clients
Clients have a desire to travel to their destinations and depend on rides from hosts. We distinguish immobile and mobile clients. Immobile clients rely completely on rides. Mobile clients can alternatively walk forward by their own, but far slower than taking rides. Mobility of clients can depend on their preference, their luggage, or their company (e.g., children).

Some clients might stick to preselected routes (e.g., the shortest) and only look for rides along their route. Other cost functions of clients might be travel time, number of transfers, or trip fares. Clients with a desire to optimize these cost functions will accept detours, as long as they promise to reach the destination for lower cost. For some clients, shorter travel times are more important than trip fares, while budget clients favor cheaper rides. Fewer transfers are more attractive to clients who appreciate comfortable trips, while scenic views would be a cost function (to maximize) for tourist clients. Frequently clients balance these factors with some subjective weighting. Furthermore, clients can have preferences, such as for types of hosts, or for specific profiles of vehicle drivers.

Another factor to consider is the environmental knowledge of the client. While we generally assume that the client has knowledge of the street network, it makes a difference whether the client knows the network and time tables of mass transport, or typical traffic patterns in the city.

3.2 Mass transport
Mass transport in a city includes buses, trains and trams, subway, and ferries. Generally, mass transport vehicles supply a larger passenger capacity compared to other means of transport, although with less comfort and privacy. Travel fares are relatively cheap, especially with flat fare structures on longer distances, or with tickets that are interchangeably valid on various modes of mass transport. Frequently fares are charged by time regardless how far to travel, but other payment systems exist as well.

Mass transport follows fixed timetables, typically with larger gaps between midnight and early morning and varying frequency over the day. They run on predefined routes back and forth, and passengers are only allowed to get on or off at stops. This means that mass transport does not provide door-to-door transport, nor does it reach some areas in the city at all. Some mass transport modes run on their own line network, e.g., trains, trams and subway, or on reserved lanes, and are less affected by other traffic. This means that mass transport vehicles can be faster than street traffic bound vehicles.

3.3 Taxi cabs
Taxi cabs are another popular means of transport. Taxis are more comfortable and convenient compared to mass transport. Taxis can reach every location in a city’s street network, and can be called at any time of the day. Passengers can head directly to their destinations without compulsive intermediate stops or transfers. Detouring, change of destination, and stopovers are also possible during travel.

The main disadvantages of taxis are a limited passenger capacity, and correspondingly, a high trip fare. Normally, taxis have about four seats for passengers, but these are only shared for a group having the same trip. Taxis are charged by a combination of travel distance and time; sometimes a flag fall is added. This means that taxis are more suitable for individual travellers or small groups travelling together, either for short trips, or when time or convenience is more valued than money.

3.4 Private cars
Functionally, private cars in their function as hosts of shared rides are similar to taxis: they share the advantage in comfort, and the disadvantage in low passenger capacity. The difference is that private cars are owned by their drivers, and hence, are considered as private space, or proxemics [8].

Private car drivers, if willing to offer a ride, are unlikely to serve clients off their route. They can pick up clients anywhere along their own trip, but may provide only parts of the travel of a client. Private car drivers may have more rigid interests and preferences in selecting clients, such as non-smoking clients, or clients of a specific gender.

Compared to taxi fares, a ride in a private car could be free, if the incentives for the car drivers may be nonmonetary, such as being allowed to use high-occupancy vehicle lanes. Alternatively, they can charge proportional to the travelled distance, but to lower rates as taxis because their interest is mostly in sharing costs.
4. FORMALIZATION IN A MULTI-AGENT SIMULATION

This section presents a specification of a peer-to-peer shared ride simulation, with the types of agents and their behavior as discussed above. The simulation is implemented in an object-oriented architecture using Java. Various transportation agents are designed inheriting from a base class (transportation) agent.

In our peer-to-peer shared ride system, the agents have knowledge of their locations within the street network, negotiate with their neighbors for shared rides, make decisions according to their desires and intentions, and travel until the next negotiation takes place. Therefore, this system can be seen as a geographic automata system: it has states, and state transitions, in particular finding a ride, depend on the neighbors.

To implement geographic automata systems, Benenson and Torrens [3] suggest to establish a spatially restricted network with immobile and mobile agents, neighborhood relationships and behavior rules. Due to their interest on urban objects, such as buildings or residential addresses, they use a cellular network. In contrast, agents in shared ride systems move in street networks, and hence, we use a grid network to model a real street network, with nodes representing street intersections and edges the street segments.

4.1 Environmental parameters

Agents travel along edges, but are only at nodes allowed to take or change a ride. In the grid network, nodes have coordinates \((x, y)\), which represent index numbers of grid columns and rows. Since coordinates form already a primary key, we do not design an additional identifier field for nodes. With identifiers, an extra reference list between identifiers and coordinates would be needed, which results in additional computations. Also, the length of message is of no concern in a simulation, and hence a key of two fields is as appropriate as a key of one field. Nodes are specified in Table 1.

Edges, the connections between neighboring nodes in this grid, have a length of unit size. The dimension of the grid network is scalable by setting its width and length in terms of numbers of edges. Additionally, an internal clock is employed to synchronize the behavior of the agents. With respect to the experiments in the simulation, several environmental parameters are designed to control the communication range and communication mode between agents, including a counter for the negotiation messages. Furthermore, the agent behavior is relative to their type, and the number of clients and diverse hosts are specified by parameters. The class simWorld is specified in Table 2.

4.2 Communication protocol and strategies

In a peer-to-peer shared ride system, clients depend on transportation information from all hosts to plan optimal trips. However, in the dynamic traffic, an individual client may not reach or may not want to reach all hosts in the street network. This means that clients have to plan a trip with local knowledge only. Nagel suggests that trip plans always include a start time, a start position, and a destination. In shared ride planning, agents are additionally interested in the agents involved in the trip, and arrival times. To enable negotiations between agents for trip plans, a communication protocol is designed for messages of the structure specified in Table 3. The details of the communication model and protocol are specified by Winter and Nittel [18].

In a peer-to-peer system agents radio broadcast messages to their neighbors. Their radio range is limited according to the broadcasting technologies, such as Bluetooth and WiFi, and the broadcasting power. Distant agents can be reached by forwarding messages (multi-hop broadcasting). For a peer-to-peer shared ride system the communication window—the synchronized time all agents listen and broadcast—requires to be long enough to accomplish a complete negotiation process, consisting of a request, offers, and a booking. This means that from the previously investigated three communication strategies—unconstrained, short-range and mid-range [18]—the unconstrained communication strategy is not feasible in reality. Unconstrained communication means that messages flood to the deepest agents in network, as long as agents are connected \((comRange = \infty)\). The other two are local communication strategies. In short-range communication, agents only communicate to agents within their radio range (single-hop, \(comRange = 1\)). In mid-range communication, agents forward messages within several hops (\(comRange > 1\)). The negotiation process will be simulated for different communication ranges to investigate trip planning with different levels of transportation network knowledge.

---

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(x)</td>
<td>(x) coordinate in a grid</td>
</tr>
<tr>
<td>2</td>
<td>(y)</td>
<td>(y) coordinate in a grid</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(length)</td>
<td>length of simulation grid system</td>
</tr>
<tr>
<td>2</td>
<td>(width)</td>
<td>width of simulation grid system</td>
</tr>
<tr>
<td>3</td>
<td>(unit)</td>
<td>length of a grid edge (=1)</td>
</tr>
<tr>
<td>4</td>
<td>(comRange)</td>
<td>communication range (number of hops of messages)</td>
</tr>
<tr>
<td>5</td>
<td>(clientNum)</td>
<td>number of clients</td>
</tr>
<tr>
<td>6</td>
<td>(hostNum)</td>
<td>number of hosts</td>
</tr>
<tr>
<td>7</td>
<td>(msgNum)</td>
<td>count of the total number of broadcasted messages</td>
</tr>
<tr>
<td>8</td>
<td>(time)</td>
<td>current simulation time (starts at 0)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(type)</td>
<td>(request), (offer), (booking)</td>
</tr>
<tr>
<td>2</td>
<td>(route)</td>
<td>(r), (a), (b)</td>
</tr>
<tr>
<td>3</td>
<td>(time)</td>
<td>start time of the route</td>
</tr>
<tr>
<td>4</td>
<td>(agents)</td>
<td>identifiers of agents</td>
</tr>
<tr>
<td>5</td>
<td>(speed)</td>
<td>speed of the original sender</td>
</tr>
</tbody>
</table>

---

Table 1: Node features

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(x)</td>
<td>(x) coordinate in a grid</td>
</tr>
<tr>
<td>2</td>
<td>(y)</td>
<td>(y) coordinate in a grid</td>
</tr>
</tbody>
</table>

Table 2: Simulation world features

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(length)</td>
<td>length of simulation grid system</td>
</tr>
<tr>
<td>2</td>
<td>(width)</td>
<td>width of simulation grid system</td>
</tr>
<tr>
<td>3</td>
<td>(unit)</td>
<td>length of a grid edge (=1)</td>
</tr>
<tr>
<td>4</td>
<td>(comRange)</td>
<td>communication range (number of hops of messages)</td>
</tr>
<tr>
<td>5</td>
<td>(clientNum)</td>
<td>number of clients</td>
</tr>
<tr>
<td>6</td>
<td>(hostNum)</td>
<td>number of hosts</td>
</tr>
<tr>
<td>7</td>
<td>(msgNum)</td>
<td>count of the total number of broadcasted messages</td>
</tr>
<tr>
<td>8</td>
<td>(time)</td>
<td>current simulation time (starts at 0)</td>
</tr>
</tbody>
</table>

Table 3: Message features
4.3 The negotiation mechanism
We need a mechanism to process the negotiations (Figure 1). Clients initiate a negotiation by sending a request. Hosts respond with offers, and the negotiation finishes with a booking made by the client. These three phases happen sequentially. All requests, offers and booking messages are in the format of message, and are identified by type and the original sender in agents. After each negotiation the simulation clock increments. Because agent travel changes dynamically, agents do not need to keep previous negotiations in memory. Therefore, there is no cancellation phase integrated, because booked rides are regarded as being cancelled when no rebooking/confirmation happens in the following negotiation, or no client/host show up for an appointment.

So far, only one client is generated in an individual simulation (clientNum=1). All hosts serve for this client. In this case, hosts do not need to decide which client to contribute, and there is no competition among clients.

4.4 Agent parameters and behavior
Agents are designed in a class hierarchy (Figure 2), because they all have some common features and behavior. These common features and behavior are identified and encapsulated in the base class \textit{agent}.

Common features include the agent’s identifier, its speed, its type, its state, a reference to its current simulation environment, and some information on its travel plan, such as current position and destination, a temporary container of negotiation messages. The travel route contains departure and destination, and for some agents the nodes in between. For investigation purposes, a second container stores details of booked shared rides. All agent features are listed in Table 4. Common behavior includes how to move to the next node, how to listen to neighbors and how to obtain knowledge about current position and state. The agent behavior is specified in Table 5.

![Figure 1: The cycle of negotiations and movements within one time unit.](image1)

![Figure 2: Class hierarchy of agents.](image2)

### Table 4: Agent features

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int</td>
<td>unique identifier</td>
</tr>
<tr>
<td>speed</td>
<td>float</td>
<td>speed of agent, in units of edges</td>
</tr>
<tr>
<td>type</td>
<td>char</td>
<td>agent’s type: client c; host h</td>
</tr>
<tr>
<td>time</td>
<td>int</td>
<td>the time receiving an offer/booking</td>
</tr>
<tr>
<td>state</td>
<td>char</td>
<td>agent’s current state: moving m; on a ride t; waiting w; stopped e</td>
</tr>
<tr>
<td>position</td>
<td>int</td>
<td>index of current position in route array</td>
</tr>
<tr>
<td>route</td>
<td>[node]</td>
<td>route array, the first is start point, the last is destination</td>
</tr>
<tr>
<td>messages</td>
<td>[message]</td>
<td>container of received messages at each negotiation cycle. This list is updated when a message is received, and it is cleared when a new negotiation cycle starts</td>
</tr>
<tr>
<td>services</td>
<td>[message]</td>
<td>(not accessible for agents). For clients: details of all offers so far. For hosts: details of all booking messages received so far</td>
</tr>
<tr>
<td>world</td>
<td>simWorld</td>
<td>reference to the current simulation environment</td>
</tr>
</tbody>
</table>

### Table 5: Agent behavior

<table>
<thead>
<tr>
<th>Name</th>
<th>Input</th>
<th>Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>move</td>
<td>null</td>
<td>null</td>
<td>move to the next node in route (abstract class)</td>
</tr>
<tr>
<td>listen</td>
<td>message</td>
<td>null</td>
<td>update current message container</td>
</tr>
<tr>
<td>getPos</td>
<td>null</td>
<td>node</td>
<td>get agent’s current position</td>
</tr>
<tr>
<td>setState</td>
<td>char</td>
<td>null</td>
<td>set agent’s state (input is a state)</td>
</tr>
</tbody>
</table>
Table 6: Client features

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>search</td>
<td>LPA</td>
<td>instance of shortest path search</td>
</tr>
<tr>
<td>sharedT</td>
<td>int</td>
<td>start time of the most recent</td>
</tr>
<tr>
<td>mobile</td>
<td>Bool</td>
<td>client can walk (TRUE) or not</td>
</tr>
</tbody>
</table>

Table 7: Client behavior

<table>
<thead>
<tr>
<th>Name</th>
<th>Input</th>
<th>Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>request</td>
<td>message</td>
<td>null</td>
<td>broadcast request message</td>
</tr>
<tr>
<td>book</td>
<td>null</td>
<td>message</td>
<td>book an offer</td>
</tr>
<tr>
<td>addNode</td>
<td>int,</td>
<td>node</td>
<td>add a new node after a specified position in</td>
</tr>
<tr>
<td></td>
<td>node</td>
<td></td>
<td>client’s route to an intermediate or final</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>destination</td>
</tr>
<tr>
<td>getNext</td>
<td>null</td>
<td>node</td>
<td>get the position of next node (if at some</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>position in between nodes)</td>
</tr>
<tr>
<td>move</td>
<td>null</td>
<td>null</td>
<td>decide whether and where to move</td>
</tr>
</tbody>
</table>

The classes client and host are derived from agent, and have additional properties and characteristic behavior. Their states, travel routes and current position can change over time, but type and speed are constant within a simulation.

4.5 Client agents

In our simulation, there are two types of clients: immobile clients, taking rides only, and mobile clients that are also able to walk. The first type of client needs to be picked up from their location. They cannot walk and must wait until a vehicle will pick them up for a ride. The second type of client is able to walk and can travel to a public transport line or other location if they can get a ride sooner. For them, a (time-dependent) shortest path algorithm is needed for trip planning. The algorithm applied by these clients is the heuristic lifelong planning A* algorithm. This algorithm is adaptive to a dynamic network. Given various cost functions (e.g., travel time or trip fare), this algorithm allows clients achieving different goals such as the quickest or the cheapest trip. Client features and their behavior are specified in Tables 6 and 7.

4.6 Host agents

Hosts have limited passenger capacity. Hosts need to decide how to contribute to requested routes: they can offer to share sections of their own travel plans that match with requests, or they can leave their predefined travel route and make a detour for clients. Alternatively, hosts can leave the decision in the hands of the clients, by offering their travel route ahead no matter how relevant this is to the request. Host features are shown in Table 8, and their behavior in Table 9.

5. SIMULATING SHARED RIDES WITH DIVERSE AGENTS

The specified peer-to-peer shared ride system simulation is tested for different types of agents. For the purpose of the test, the optimization criterion of travel time was chosen, looking for the quickest trip. The simulation produces output in the form of text, which can be stored or visualized. Here the stored tables of 1000 simulation runs each are summarized in diagrams.

Particularly, we have classified hosts into three groups: private cars, taxis, and mass transport. Table 10 shows the comparison in terms of values of features.

5.1 Types of clients

Figure 3a shows the average time of shared rides by various clients, and Figure 3b shows the corresponding numbers of broadcasted messages. The types of clients compared are: an immobile client who sticks to the geodesic route, a mobile client who sticks to the geodesic route, and a mobile client who is willing to make detours. Each client departs at (0, 5) and heads to the destination at (10, 5), which is a trip of ten
Figure 3: Trip times of immobile clients, mobile clients that stick to the geodesic route, and mobile clients making detours (a), and the corresponding numbers of messages (b).

In this experiment, hosts are homogenous: all are private cars. The numbers of hosts vary from 24 to 144. Each host travels over twelve time units along a travel route that is generated randomly. The simulation was run on a world of $11 \times 11$ nodes for a communication range of three.

In our experiment, the two types of mobility do not make much difference. This is partly the case because of the choice of hosts. For example, mass transport routes make more detours, and reward this effort. It is also correlated with the random routing of cars—they rarely provide longer rides—, and with the communication range: to pick up a ride in the next parallel street, a client has first to walk over one edge or four time units. What comes out clearly, though, is the advantage of being mobile. In terms of communication costs, clients willing to make detours get more offers. This increase in broadcasted messages remains in limits.

5.2 Shared rides by cars

Private cars are specified by a speed of one edge per time unit, a route that is determined by random at their departure and from which they will not deviate, and a low passenger capacity (Table 10). The latter is not relevant in a simulation with one client only. The simulation that tests a peer-to-peer shared ride system consisting of one immobile client. This type and specification of hosts has been investigated in previous work [18]. In that work it was shown that mid-range communication delivered trips nearly as quickly as with unconstrained communication, for all densities of hosts. For details we refer to that work.

However, with the introduction of different client types (Figure 3), it turns out that mobile clients, due to their increased choices, have advantages over immobile clients. If travel time is the optimization criterion, a shared ride system for private cars would lead to combined trips of rides and walks.

5.3 Shared rides by taxis

Taxis are specified by a speed of one edge per time unit, a route that is determined by random at their departure, but from which they are willing to deviate any time (if not occupied), and a lower passenger capacity (Table 10). With this specification, a peer-to-peer shared ride system for taxis leads to client trip times close to the theoretical optimum, which is defined by the distance of the client’s departure and destination, and the host speed. A taxi comes as soon as possible and heads directly to the client’s destination without detours. Only the density of taxis determines the (average) waiting time of the client. The higher the density, the shorter becomes the waiting time. Since the simulation result is predictable, we abstain from a diagram.

Communication range has a minor impact in a peer-to-peer shared ride system for taxis. Since always the nearest (free) taxi will be chosen, and no other agents are present in this system, the nearest taxi has either to be in direct communication range, or currently occupied taxis can bridge by message forwarding. Only if taxis are employed in a shared ride system together with other hosts of larger numbers, the communication connectivity established by the other hosts will make a significant difference.

5.4 Shared rides by mass transport

Means of mass transport are specified by a speed of two edges per time unit (twice the speed of other hosts), by predefined and fixed routes, and by scheduled frequencies. In this experiment the frequency of a bus is set to every ten time units, and the world is of size $21 \times 21$. We distinguish two cases (that reappear in the mixed simulation of Section 5.5). Case 1 is a simulation with a bus line through the parallel street of the client’s geodesic route. Case 2 is a simulation with a bus line overlapping with a major part of the geodesic route. Both cases are investigated for (a) buses being the only hosts in the simulation, (b) buses being hosts among 480 occupied private cars, such that private cars establish communication connectivity but do not offer rides, and (c) buses and bus stops being the only hosts in the simulation, such that missing communication connectivity to the buses is balanced by the presence of the bus stops within the direct communication range of the client.
Figure 5: Two bus lines, a client's departure in the center and destination at the far right end.

Figure 4 demonstrates the results, both in terms of average trip times as well as numbers of broadcasted messages. It turns out that the presence of bus stops is of advantage compared to both other scenarios. The numbers of messages are within limits larger than for buses only, but the average travel times are significantly shorter. Bus stops also help to reduce the communication effort in the presence of other hosts that are willing to establish connectivity within comRange.

5.5 Shared rides deploying all types of agents
In the final experiment we deploy all types of hosts for a mobile client willing to make detours. Results in this case depend completely on factors such as host densities and composition of the hosts. However, this experiment can demonstrate two properties of the system: that (i) mid-range communication is consistently nearly as effective as unconstrained communication, and that (ii) mid-range communication is consistently more efficient. The latter property increases radically with increasing host density, i.e., with increasing connectivity in the agents’ network. Other properties can be studied as well, for example, finding a balance in the composition of hosts.

Figure 5 shows a world of $21 \times 21$ nodes, two bus lines, and a client’s geodesic route from the center to the right. Note that this client is willing to make detours, hence, it does not have to follow the geodesic route. Within this world the shared ride system consists of a constant number of 120 agents in three different compositions (Figure 6): (a) 24 taxis, 24 buses (with a frequency of two time units), and 72 private cars, (b) 24 taxis, 24 buses, 24 bus stops and 48 private cars, and (c) 48 taxis, 24 buses and 48 private cars.

For these three compositions we observe a decreasing travel time from (a) to (c). The effect of bus stops, documented before already, can be recognized again (b), but a larger number of taxis exceeds their effect (c). The decrease is moderate, though, which means that (on average) rides involving taxis and buses are taken. These rides reduce trip times significantly compared to the ‘cars only’ scenario for

Figure 6: A world of 120 host agents of different composition, and a mobile client.
Figure 4: Means of mass transport, although they are bound by their routing and schedules, have an influence on trip times. The two investigated cases are explained in the text.

Figure 7: Different compositions of 120 hosts.

mobile clients in Figure 3. This result is generally confirmed by Figure 7, which compares different compositions of 120 hosts for mid-range communication only. Note that the worst trip time is 40 time units (the client walks), and the best trip time in the present scenario is 7.5 time units (taking the bus, and finding other rides for the rest without delay).

At the same time, the number of broadcasted messages increases from (a) to (b). This is triggered by the large number of offers made by the bus stops. Bus stops currently do not filter their responses by relevance, but offer each approaching bus. This means that this figure for (b) can be reduced by future modifications of the bus stops. For the two compositions in (a) and (c), the broadcasted messages in each negotiation should be the same, but since (c) leads to shorter travel times, or less negotiations, the total sum of messages is smaller in (c) than in (a). The current host density (120 host on 441 nodes) is relatively low, which means that communication connectivity is low. Further experiments with higher densities of hosts shows that the numbers of broadcasted messages are more rapidly increasing for unconstrained communication than for the local communication strategies.

The interesting result, though, is a comparison with the immobile client in the 'cars only' scenario, at the same host density (Figure 3a, upper curve). The results for the immobile client are repeated here from [18]. Compared to their scenario, we can observe that with the introduction of other types of agents (i) the (average) trip times change significantly, effectively being reduced by more than 50%, and (b) mid-range communication is still nearly as efficient than unconstrained communication, with the number of broadcasted messages by mid-range communication being roughly the same.

5.6 Multi-criteria optimization

The previous investigations assume clients pursuing shortest travel time trip planning. In practice, other criteria are also used, such as trip fare and numbers of transfers. In this case, multiple criteria are employed and optimized by clients. That allows clients, for example, to look at a relative quick and cheap trip. Imagine that travel time, trip fare and numbers of transfers are chosen as three criteria, and linear programming is used to balance the importance of them. Table 11 demonstrates the multi-criteria optimization.

In this case, trip 2 is the optimal one; it is the trip of shortest travel time, of no transfer, but is the most expensive one. With other weighting of the criteria other trips could be
preferred. Assume, for example, that a client has a stronger preference for a cheaper trip. This client might choose a relative weighting of (0.2, 0.6, 0.2), and find the trips 1 and 3 equally optimal.

6. DISCUSSION OF THE RESULTS
In the experiments, the characteristic parameter is the density of agents, not their absolute number. Various sizes of grid networks with the same agent density will influence the number of messages for the unlimited strategy only—the one that is not realistic and used only for control purposes. Larger sizes of grid networks allow longer trips, but even this is not a critical change of the concept. Hence we expect that our results hold for longer trips, and also for other forms of street networks.

Walking clients have the risk of missing potential rides during walk. For example, a walking client can see a bus passing along if this bus did not exist at departure time of the client, or if the bus was still out of the client’s communication range. This risk can be reduced by choosing communication ranges large enough to provide the client with all relevant offers for this period. But extra communication costs energy.

Another arbitrary design of the simulation is that buses are running on parts of the geodesic route. As discussed before, the restrictions of buses, such as pick-up at stops only and fixed timetables, limit their occupation.

Figure 4 shows that under some conditions buses, if travelling along parallel streets, can even not contribute to client trips at all. This happens with the communication range being not large enough to inform the client in time to start walking to the parallel street. Globally adapting the communication range to the speed of the hosts helps in this situation (if other hosts establish multi-hop connectivity). This way, the communication effort is increased significantly, which contradicts our intentions and hypothesis. Alternatively, the agency of bus stops helps. In our simulation, bus stops currently offer each approaching bus, in any distance, in an individual offer message. This increases the number of messages in Figure 4, but can be reduced by offering only relevant buses.

So far, all trips are designed to achieve locally the quickest trip. In practice, clients probably look at other criteria as well, such as fares and transfers. The employment of other criteria is expected to change the results. The employed lifelong planning $A^*$ algorithm is adaptive to any criteria, such as fares or transfers. However, only one optimal path is returned by this algorithm each time, and alternatives—for example for ranking with multiple criteria—are not provided. To achieve multi-criteria optimization, all possible trips need to be calculated under these criteria separately. Then linear programming can be used to balance factors and find an optimum.

7. CONCLUSIONS AND OUTLOOK
Previous research studies the behavior of a peer-to-peer shared ride system with an immobile client following a geodesic route, and private cars. It is found that for a peer-to-peer shared ride system mid-range communication is both efficient and effective [18]. In this paper, we extend the previous research with mobile clients, various types of hosts, and other agents, applying mid-range communication in our experiments. We went out to show that employing other types of agents change the trips significantly, but mid-range communication is still the preferable range.

Reviewing the results from our simulation, we can see that multiple types of agents enrich the choices of clients and as a result lead to trips of generally lower costs. The largest impact has a peer-to-peer shared ride system with mobile clients and all types of host agents, since it provides the largest choice for a constant communication range. Mid-range communication still delivers trips of durations close to a (fictional) unconstrained communication range, but has much lower communication costs. Hence, the hypothesis has been proven.

Besides all the improvements in reducing trip times, one problem remains: trips derived from local knowledge (of any communication range) may not be optimal from a global view. Better rides provided by distant hosts and hosts entering the traffic after the client has made a booking are always possible, and can be documented from a subsequent analysis of the simulation protocol. This problem can be approached by more intelligent wayfinding heuristics of the clients. Clients could, for example, learn from experience and use this knowledge in predicting chances of being picked up at specific nodes. For this purpose, a client could, for example, exploit a hierarchy in the street network, or known traffic counts at particular intersections, to assess potential transfer points in the trip planning process. This idea is investigated elsewhere [6].

Related to more intelligent wayfinding behavior is the request for multi-criteria optimization. For example, clients may be interested to reduce their number of transfers and their trip time. The introduction of different fare models, and the choice of the cheapest trip (or of a balanced cheap trip in a multi-criteria optimization), will further allow to test economic concepts of a peer-to-peer shared ride system. This requires a change in the mobility model of the agents in the simulation. Although a random walker mobility model is sufficient for our interests in this paper (except for mass transport vehicles), it is no longer sufficient when numbers of transfers in a realistic traffic scenario shall be investigated. For private cars for example, regular traffic patterns can be introduced, such as back and forth between home and work, as it is done in traffic micro-simulation [15].

Another future extension of this system comes with admitting other clients in the simulation ($\text{clientNum} > 1$). Then the passenger capacity of the hosts becomes a critical re-

<table>
<thead>
<tr>
<th>Trip</th>
<th>Time</th>
<th>Fare</th>
<th>Transfers</th>
<th>Total costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>linear weights:</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>∑</td>
</tr>
<tr>
<td>1 a → b, b → c</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>3.9</td>
</tr>
<tr>
<td>2 a → c</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td>2.7</td>
</tr>
<tr>
<td>3 a → d, d → e, e → c</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>3.7</td>
</tr>
</tbody>
</table>
source. Clients would compete with each other, which might recommend more booking ahead. But more aggressive booking strategies conflict with the hosts’ interests of travelling with occupied vehicles, since travel plans are highly dynamic. Balancing these interests need to be investigated.

Some steps to exploit properties of dynamic transport networks for trip planning from local knowledge are made, but many other questions lie ahead.

8. REFERENCES


