Determination of efficient meeting points in ride-sharing scenarios

Paul Czioska
Leibniz University of Hannover
Institute of Cartography and Geoinformatics
Hannover, Germany
Paul.czioska@ikg.uni-hannover.de

Monika Sester
Leibniz University of Hannover
Institute of Cartography and Geoinformatics
Hannover, Germany
Monika.Sester@ikg.uni-hannover.de

Abstract

Ride-sharing is an efficient way to increase vehicle occupancy rates and hence to reduce the number of necessary vehicles and traffic congestion in urban areas. Since ride-sharing usually do not include predefined boarding locations, it is necessary to determine a suitable meeting point between the driver and the rider. Ideally, such a point is located in a way that the travel time and distance is reduced. In this work, five simple optimization methods to identify reasonable meeting point locations on a real street network are compared. Results show that the intersection of Space-Time prisms deliver good results in terms of performance and computing capacity.

Keywords: Dynamic Ride-Sharing, Optimization, Transportation, Routing.

1 Introduction

Many urban areas suffer from traffic congestion and air pollution, which is a growing problem that arises from individual motor car traffic. However, private car occupancy rates are still very low. In Germany, car occupancies range from 1.9 for leisure trips down to only 1.1 for daily commuting trips [4]. Nearly two third of all private car trips are made alone.

An effective way to use empty seat capacities and share the travel costs is ride-sharing. In the static case, travellers with similar itineraries and time schedules are matched in advance or on a regularly basis (e.g. for commuting), whereas the dynamic case considers an ad-hoc matching [1]. There exist already various commercial systems which realize such a ride sharing, e.g. Flinc, BlaBlaCar or Matchrider.

In most cases, the driver with the vehicle does not start at the same place as the rider. This assumption reveals the need for a meeting point. A trivial solution is that the driver picks up the rider at its origin, e.g. in front of his home. However, an efficient meeting point could reduce driving time and distance when they meet halfway.

Ride-sharing is predominantly related to several topics in the Operations Research domain, such as the Vehicle Routing Problem (VRP) or Pickup and Delivery Problems, especially when considering (dynamic) Dial-a-Ride Problems (DARP) [1, 2, 3]. A recent work gives an extensive view into the benefits of meeting points in shared-ride scenarios [9]. The authors use a maximum weight bipartite matching algorithm to assign riders to drivers. Other approaches focus on the trade-off between profitability for the hosts and accessibility for the clients [6] and the determination of dynamic *Bus Boarding* and *Alighting Points* [7]. A related work investigates *Rendezvous* and *Leave Points* based on the rider's position and the planned route of the driver [8]. However, most approaches are based on the Euclidean distance for walking accessibility, so that a further investigation of methods to find efficient meeting points on a graph structure is required.

2 Problem definition

In order to create representative scenarios, the following conditions are applied:

- Exactly one rider is assigned to exactly one driver
- Driver and rider have different origins and a common destination. A related scenario could be the joint commuting to a train station
- The driver uses a vehicle for the whole route
- The rider walks to the meeting point and boards the vehicle there.

The speed difference between driver and rider is denoted in the following with:

$$z = \frac{VehicleSpeed}{WalkingSpeed} \tag{1}$$

The objective function f used for the optimization procedure is defined as the maximum travel time for the driver or the rider. Further, a benefit value g is introduced which compares this value to the time needed without a meeting point:

$$f = \max(t_D, t_R) \tag{2}$$

$$g = \frac{t_P}{f} \tag{3}$$

subject to

$$t_D = t_{DOMP} + t_{MPD}$$
$$t_R = t_{ROMP} + t_{MPD}$$
$$t_P = t_{DORO} + t_{ROD}$$

t_{DOMP}: time(DriverOrigin → MeetingPoint, VehicleSpeed)
t_{ROMP}: time(RiderOrigin → MeetingPoint, WalkingSpeed)
t_{DORO}: time(DriverOrigin → RiderOrigin, VehicleSpeed)
t_{MPD}: time(MeetingPoint → Destination, VehicleSpeed)
t_{ROD}: time(RiderOrigin → Destination, VehicleSpeed)

This formulation leads to a balancing of travel times between driver and rider, so that they ideally meet simultaneously at the meeting point. An implicit condition is thus that both have overlapping time windows.

3 Compared Algorithms

In the following, five simple approaches to find appropriate meeting point nodes are described and compared.

Geo-Radius (GR)
Like most existing approaches, this method uses a simple distance threshold to retrieve possible meeting points in the surrounding of the rider. All detected nodes are evaluated with the goal function and the one with the lowest value is chosen. Here, a threshold of 1000m was used.

Single Gradient Descent (SGD)
The algorithm starts at the origin node of the rider. First, the objective function is calculated of all neighbouring nodes. If an adjacent node has an equal or lower value than the current node, it is chosen to be processed next. This procedure is repeated until no adjacent node with a lower value exists.

Triple Gradient Descent (TGD)
This method is equal to SGD, but with three initial start positions: at the riders' origin, the drivers' origin and the destination. The node with the lowest value is chosen.

Catch-Up Zone (CUZ)
Starting from the origin node of the rider, all nodes in the surrounding which can be reached earlier by the rider than by the driver are exploited. In addition, all first nodes outside of this "Catch-Up front" are also visited since the node with the optimal solution may be reached earlier by the driver than the rider. When driver and rider use the same network, this method delivers always the global optimum.

Intersection of Space-Time-Prisms (STP)
Based on the principles of time geography, space-time prisms are a feasible way to model accessibility [5, 7]. The intersection of space-time prisms can subsequently be used to indentify nodes where driver and rider can possibly meet at a given time. The algorithm discovers these nodes and stops when the accessibility front of the driver has reached the riders origin.

4 Results

The presented results are based on an experiment on the street network of Braunschweig, obtained from OpenStreetMap (~ 7500 edges). The edge weight is dependent only on the edge length; no speed limits have been taken into account. Figure 1 illustrates exemplarily a situation in an inner-city shared-ride trip. The performance of the five algorithms, represented as a percentage value corresponding to 100% as the global optimum, is shown in figure 2. As can be seen, the more different the speeds of driver and rider gets, the better the algorithms perform because of the reduced potential target area. Figure 3 demonstrates how many nodes (in relation to the total amount of nodes) have to be visited on average by the different algorithms. Obviously, this value is dependent on the speed difference factor z at the methods CUZ and STP, in contrast to the other methods. Finally, figure 4 portrays the benefit ratio g, which decreases with an increasing speed difference factor.

Figure 1: Meeting point situation on a street network

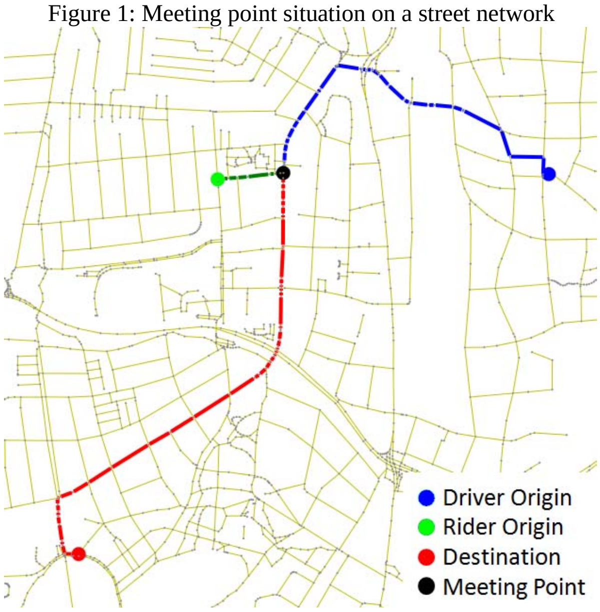


Figure 2: Mean performance depending on speed difference

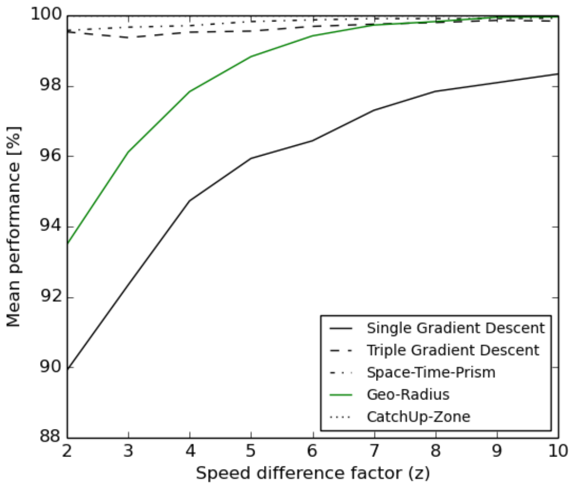


Figure 3: Mean number of objective function calls

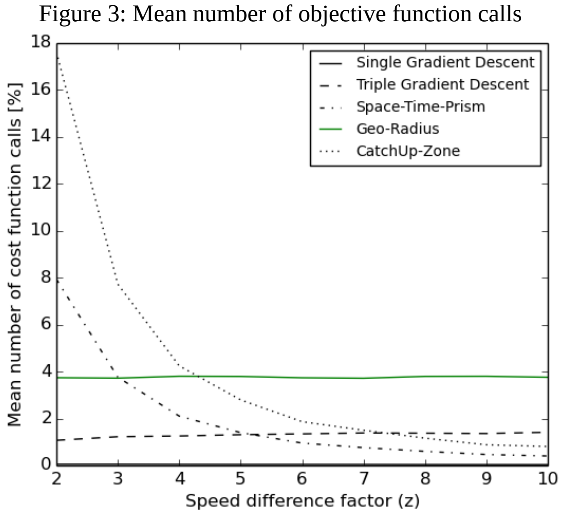
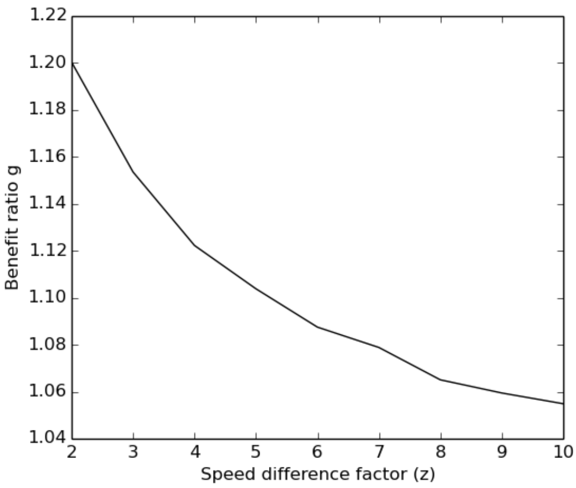


Figure 4: Benefit ratio g depending on the speed difference



5 Conclusion and Outlook

Five optimization algorithms have been evaluated on a real street network. Results show a travel time benefit between 20 % (Driver is twice as fast as the rider) and 5 % (Driver moves 10 times faster than the rider). With a reasonable speed difference factor of 6 for inner city traffic, the benefit is approximately 9 %. Catch-Up Zone and Space-Time-Prism methods perform on average better than Gradient Descent or Geo-Radius approaches to find an efficient meeting point, but especially at lower speed differences more nodes have to be visited.

For future work, all constraints in section 2 can theoretically be relaxed and combined with the mentioned methods (Drop-off points, 1:n or n:m relations between driver and rider etc.). Furthermore, time windows should be introduced. Another future goal is a better determination of the meeting point itself with the help of various map data.

References

[1] N. Agatz and A. Erera and M. Savelsbergh and X. Wang. Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223: 295-303, 2012.

[2] G. Berbeglia and J.-F. Cordeau and G. Laporte. Dynamic pickup and delivery problems. *European Journal of Operational Research*, 202: 8-15, 2010.

[3] J.-F. Cordeau and G. Laporte. The dial-a-ride problem: models and algorithms. *Annals of Operations Research* 153.1: 29-46, 2007.

[4] Infas (Institute for Applied Social Science Research) and DLR (German Aerospace Center). Mobility in Germany. http://www.mobilitaet-in-deutschland.de/engl/ (15.01.15)

[5] H. J. Miller. Modelling accessibility using space-time prism concepts within geographical information systems. *International Journal of Geographical Information System* 5.3 : 287-301, 1991.

[6] N. Mukai and T. Watanabe. Dynamic transport services using flexible positioning of bus stations. *Autonomous Decentralized Systems, 2005. ISADS 2005. Proceedings*, 259-266, 2005.

[7] D. O'Sullivan and A. Morrison and J. Shearer. Using desktop GIS for the investigation of accessibility by public transport: An isochrone approach. *International Journal of Geographical Information Science*, 14.1: 85-104, 2000.

[8] R. Rudnicki and K.-H. Anders and M. Sester. Rendezvous-Problem in Local Shared-Ride Trip Planning. *International archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38, 2008.

[9] M. Stiglic and N. Agatz and M. Savelsbergh and M. Gradisar. The Benefits of Meeting Points in Ride-sharing Systems. *ERIM Report Series Reference No. ERS-2015-003-LIS*, 2015.