Chair for Information Systems
Faculty of Informatics
Technische Universität München

A Matching Algorithm for Dynamic Ridesharing

Maximilian Schreieck[a*], Hazem Safetli[b], Sajjad Ali Siddiqui[c], Christoph Pflügler[d], Manuel Wiesche[e], Helmut Krcmar[f]
[a-f]*Chair for Information Systems, Technical University of Munich, Germany*

To be presented at: International Scientific Conference on Mobility and Transport 2016, 6th/7th June, Munich
*Corresponding author: Maximilian.Schreieck@in.tum.de, 089 289 19510

MOTIVATION

Ridesharing is one of the emerging solutions to increase the efficiency of the transportation networks by reducing the empty seats while traveling in private vehicles.

Traditional ridesharing approaches are suitable for long-distance travel, especially inter-city travel, yet they are not flexible enough for short routes within cities. Contrarily, dynamic ridesharing provides a means by which two or more travelers can be joined in real time like taxi service responsiveness (Ying et al., 2008, Arena et al., 2013).

Goal: A dynamic ride matching algorithm that matches drivers and passengers without pre-defining a meeting point

THEORETICAL BACKGROUND

Shortest-Path Algorithms

Dijkstra	A*	Reach-based
ALT	SHARC	Core-ALT
REAL	CHASE	

Ride Matching Algorithms

- Agatz et al. (2011): Rolling horizon optimization strategy
- Tao and Chen (2008): greedy heuristics for taxipooling services

We enhance this literature by developing a simple yet powerful algorithm for dynamic ridesharing

DYNAMIC RIDESHARING APPROACH

Inverted Index Data Structure

Whenever a ride offer is created, the system generates a set of nodes that composes the shortest route between the source and destination points of the ride. In order to get the shortest route nodes N (the set of nodes composing a route), only departure and destination location are required. We apply the inverted index data structure on the mechanism of saving ride's nodes.

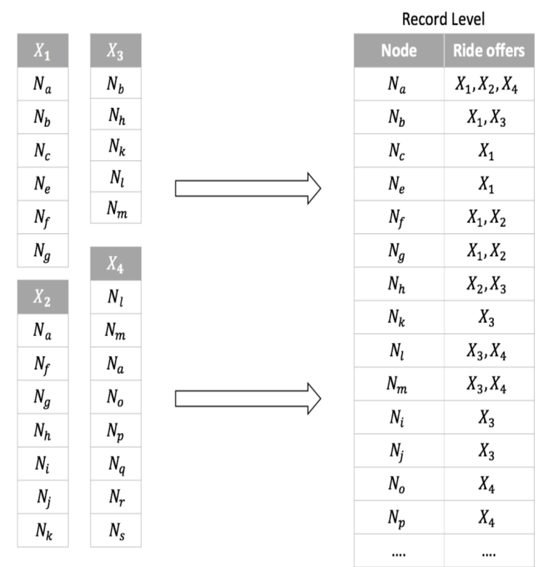

Figure 1: Inverted Index.

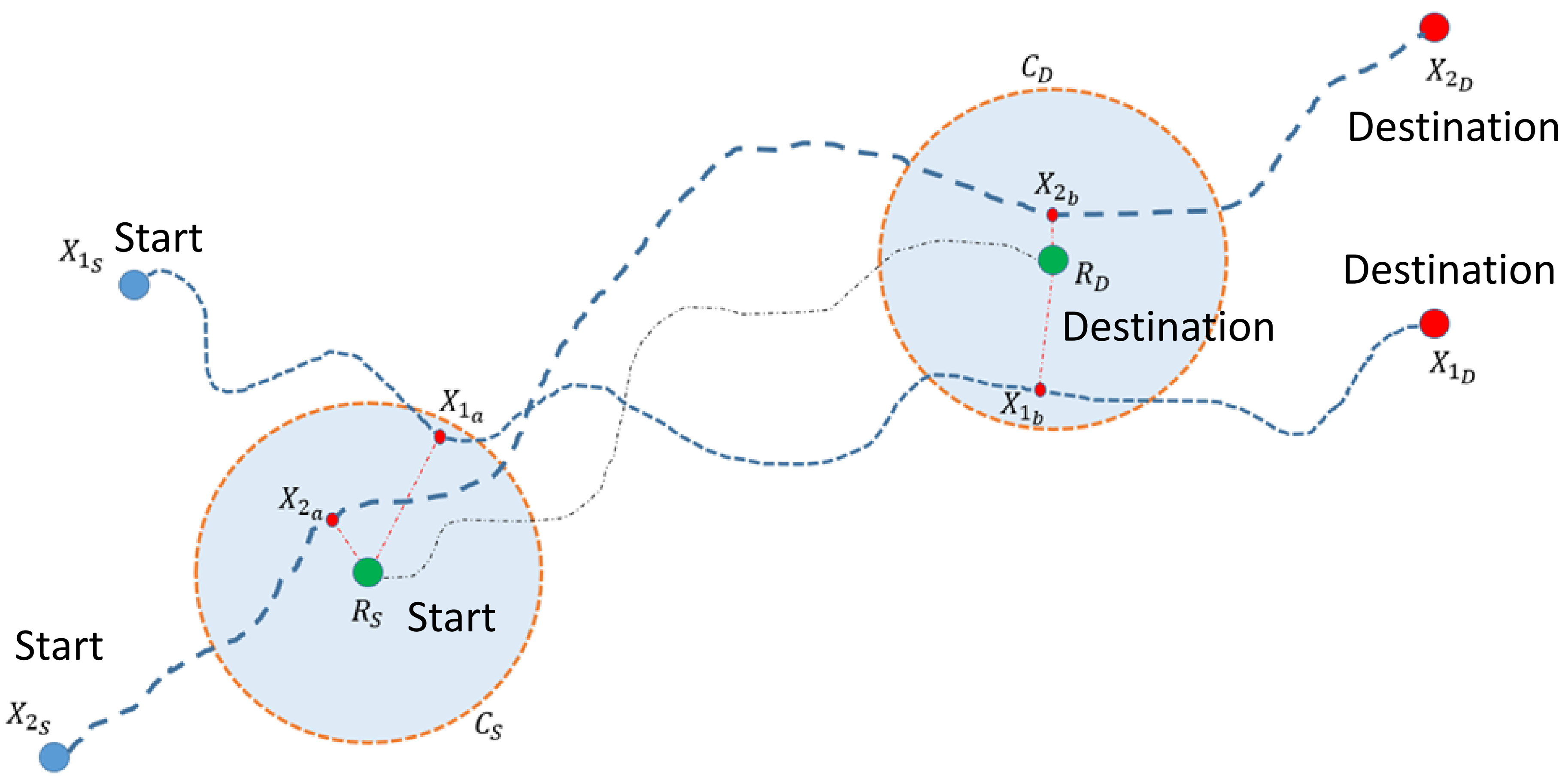

Figure 2: Single ride request, multiple ride offers.

	Matching Steps	50 Rides	500 Rides	10,000 Rides
1	Rides around departure (sec)	0.003	0.023	0.099
2	Rides around destination (sec)	0.002	0.003	0.011
3	Intersection (sec)	0.000	0.000	0.002
4	Check intersection departure time (sec)	0.000	0.000	0.103
5	Check max detour condition (sec)	0.000	0.000	0.179
6	Check ride offers direction (sec)	0.000	0.000	0.003
	Total (sec)	**0.005**	**0.026**	**0.398**

Table 1: Evaluation of the matching algorithm.

Matching Process

Assuming that there is one ride offer *X*, in the system, the set of nodes composing the shortest route between the source (X_S) and destination locations (X_D) is denoted by this equation:

$$SP(X_S,X_D) = (X_1,X_2,X_3,\ldots,X_n) \quad (1)$$

Similarly, the set of nodes for ride request R, that makes the shortest route between the departure location and the destination location of the passenger is represented by the following equation:

$$SP(R_S,R_D) = (R_1,R_2,R_3,\ldots,R_n) \quad (2)$$

The matching process algorithm starts by collecting the geographical nodes around departure location R_S and destination location R_D within radius r given by the passenger.

In order to get the ride offers *X* that are close to ride request *R*, we check which ride offers are close to the departure and destination locations of the ride request. This can be done by applying two intersections as the following:

$$S_S = C_S \cap SP(X_S,X_D) \quad (3)$$

$$S_D = C_D \cap SP(X_S,X_D) \quad (4)$$

If both sets S contain nodes from the same route, a match is found.

EVALUATION

We tested the performance of the algorithm for different numbers of rides in the system. Even with 10.000 rides in the system, the algorithm took only 0.4 seconds to derive the best match (Table 1).

REFERENCES:
AGATZ, N. A. H., ERERA, A. L., SAVELSBERGH, M. W. P. & WANG, X. 2011. Dynamic ride-sharing: A simulation study in metro Atlanta. Transportation Research Part B: Methodological, 45`, 1450-1464.
TAO, C.-C. & CHEN, C.-Y. 2007. Heuristic Algorithms for the Dynamic Taxipooling Problem Based on Intelligent Transportation System Technologies. Fourth International Conference on Fuzzy Systems and Knowledge Discovery, 590-595.

ACCELERATOR

TUMITFAHRER

"With TUMitfahrer, members of TUM can easily share rides. Whether you go from one of TUM's campus to another or spend a day hiking in the Alps – TUMitfahrer helps you to organize a shared ride." www.tumitfahrer.de

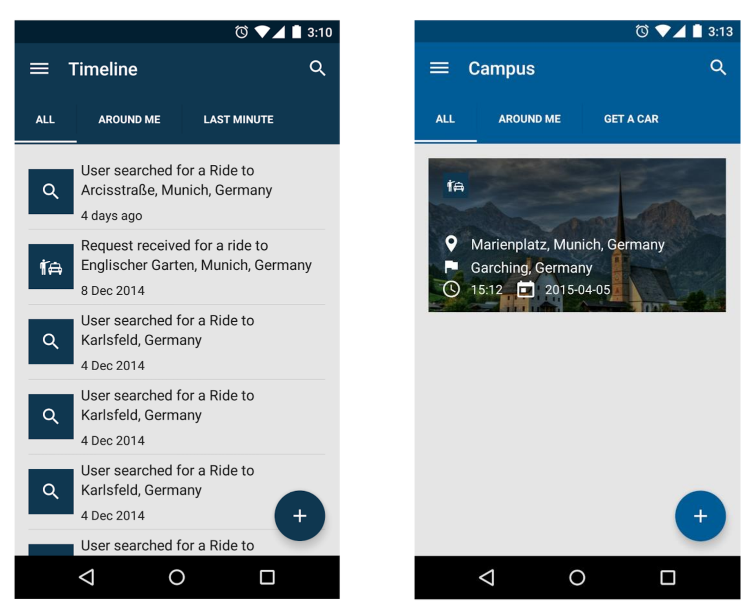

Figure 3: Screenshots of the TUMitfahrer application.

Gefördert durch:
Bundesministerium für Wirtschaft und Energie
aufgrund eines Beschlusses des Deutschen Bundestages

excell

Chair for Information Systems