Taxi Recommendation System Using Big Data Analysis and Dijkstra Algorithm

Aahna Kansal [1], Er. Inderjeet Singh[2]
[1, 2] Dept of Computer Science Engineering
[1, 2] Asra College of Engineering and Technology, Bhawanigarh, Punjab

Abstract- *With the rapid development of urbanization, the imbalance between supply and demand of taxi is becoming more and more serious. For the taxis, when they are cruising on the street the drivers looking for passengers, most drivers rely on their experience and intuition for the guideline to optimize their cruise routes and increase profit. Therefore, in this paper, the method and application of large data mining of taxi moving trajectory were presented.The system is based on multimodal annotation of geo-data. Users can rank their choices and then, based on the multi-criteria cost associated with each route, the best route can be used by them. Taxi GPS data has been used for Beijing city in which time and latitude and longitude co-ordinates has been stored for each taxi after every five minutes of time during their travel. At first, all text files have been read in the mat lab and collected dataset has been obtained for a number of taxi data in a given day. Then pre-processing is carried out in order to filter out the parking candidates based on algorithm designed for parking candidate selection. Essentially, the candidate detection algorithm finds out the locations where the GPS points of a taxi are densely clustered, with spatial and temporal constraints. After that Dijkstra algorithm has been applied based on starting and destination co-ordinates in which at first nearest parking slot is find out using the Euclidian distance and then shortest feasible route is find out between the starting and destination co-ordinates. Geographical show has been provided in the end to show the route between starting and destination nodes by Geoshow command. Proposed system effectively finds out the route along with intermediate nodes through which taxi can travel along with approximate distance of the journey in kilometers as well.*

Keywords- Taxi Route recommender system, Parking candidate selection, Dijkstra algorithm, data mining, etc.

I. INTRODUCTION

Traveling plays an important role in our lives and more and more people choose to use vehicles for traveling. To facilitate route selection, a variety of navigation services become available and are able to recommend routes when source, destination, and sometimes, departure time, are given. However, the routes recommended by existing navigation services are not always preferred by all drivers. For example, a recent study suggests that the routes provided by a leading navigation service often fail to agree with the routes chosen by local drivers [1]. The reason of the disagreement may be two-fold. First, most of the existing navigation services only consider a limited number of travel costs, e.g., distance or travel time, and return routes that minimize a single travel cost, e.g., shortest routes or fastest routes. In contrast, drivers may consider a multitude of different travel costs. For instance, due to an increasing public awareness of environmental protection and high fuel pricing, many drivers increasingly consider fuel consumption [2], in addition to travel times and travel distances. Second, existing navigation services provide all drivers with the same routes (e.g., shortest routes or fastest routes) and they do not take into account individual drivers' driving preferences (e.g., time-efficient driving, fuel-efficient driving, or some trade-off between them). These motivate us to study how to model drivers' driving preferences and to provide personalized routes to different drivers, which can better satisfy drivers' needs. Figure 1 shows two different driver's choices of routes from source s to destination d. Both routes have similar distances; however, $route_A$ take less travel time and $route_B$ takes less fuel1.

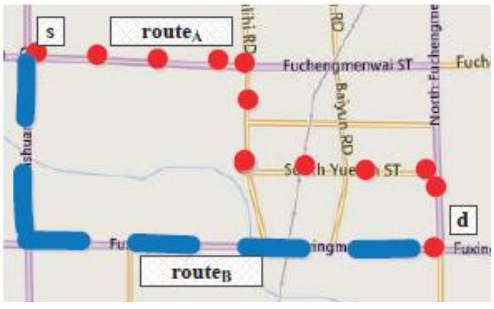

Figure 1: Routes Used by Two Different Drivers

This clearly demonstrates that the two drivers have different driving preferences—one tries to save time and the other aims to save fuel. In many cases, drivers also choose routes according to trade-offs among multiple travel costs of interest. Since different drivers may have different trade-offs, a single, recommended route cannot be preferred by all

drivers.With the rapid development and continuing use of vehicle tracking technologies (e.g., GPS), big trajectory data becomes available [3]. The big trajectory data provides opportunities to enable better navigation services that consider multiple travel costs and individual drivers' driving preferences. In particular, it is possible to learn and update individual drivers' driving preferences according to their trajectories. Further, when a driver plans a route, the trajectories used by those drivers who have similar driving preferences can be utilized to suggest personalized route to the driver.

II. LITERATURE SURVEY

Huimin Lv *et al.* [4] proposed a novel model for evaluating the candidate. Based on it, they design a recommendation system for taxi drivers to minimize their cruising driving distance before taking passengers regarding the time and location of the taxi. To be specific, they first put forward the temporal probabilistic recommending pick-up points by exploring the historical trajectory data of taxi drivers. Then they introduce the novel evaluation model, and based on it, they provide an algorithm to get the optimal route of different time and location for taxi drivers.

Fan Yang *et al.* [5] proposed a new framework of recommending cruising routes based on Urban Traffic Coulomb's Law where taxis and passengers are viewed as different types of charges. Traffic charges and attractions are calculated for each region at different time slots according to Urban Traffic Coulomb's law, and then the cruising routes for drivers are computed by comparing the differences between attractions and the headings of adjacent road segments.

Xiaola Lin *et al.* [6] proposed the method and application of big data mining for moving track of taxi based on MapReduce. TheMapReduce distributed computing framework in Hadoop was combined with the mining algorithm to extract the feature points of the taxi moving trajectory and perform hotspot analysis.

Bin Yu *et al.* [7] presented the overall search-delivery process and define five factors that may explain the income difference, as well as develop a GMOL model to find a quantitative association between incomes and service strategies. Then, they finally conduct an elasticity analysis of the significant factors using the GMOL model and estimate contributions of each significant factors.

Yi Mei *et al.* [8] developed new accessibility measures for routes and have designed a multi-objective A ∗routing algorithm. They have demonstrated the use of CAPRA in four different hilly environments where the path elevation could be very steep and problematic for a person in a wheelchair.

Mei-Po Kwan *et al.* [9] paper puts forward a constraint-based model instead of using local frequency. Through taking into account the space-time characteristics of the specific origins and destinations of taxi trips, the routes identified with this constraint-based model better reflect the experience and actual route choice of taxi drivers.

C. Guo *et al.* [10] propose route recommendation problem using big trajectory data. They provide techniques for modeling and updating driver's driving preferences. They also provide efficient and effective methods to recommend personalized routes in two different settings: local route recommendation and global route recommendation.

Ting Liu *et al.* [11] presented a novel approach to discover spatiotemporal patterns of household travel from the taxi trajectory dataset with a large number of point locations. The approach involves three critical steps: spatial clustering of taxi OD based on urban traffic grids to discover potentially meaningful places, identifying threshold values from statistics of the OD clusters to extract urban jobs-housing structures, and visualization of analytic results to understand the spatial distribution and temporal trends of the revealed urban structures and implied household commuting behavior.

X. Xie *et al.* [12] evaluated the system by extensive experiments including a series of in the- field studies. As a result, the taxi recommender accurately predicts the time-varying queue length at parking places and effectively provides the high-profit parking places; the passenger recommender successfully suggests the road segments where users can easily find vacant taxis, for example, the top-1 road segment recommended by their system considering day of the week and weather conditions matches the ground truth for all of the tested areas.

Renaud Lambiotte *et al.* [13] developed new approaches that identify the evolution of surge dynamically over time. In this context, the development of algorithms and models that realize the spatio-temporal dynamics of complex urban systems using modern datasets from multiple location-based services or transport systems could be an interesting future direction to consider.

Jian-Pan Li *et al.* [14] proposed a taxi-sharing recommendation mechanism for both taxis and passengers, which combines a non-cooperative game model to solve the competition among taxis in need of route recommendations. Based on the historical information of taxis and passengers, they built time-dependent R-Trees to discover popular

locations so that a server can suggest routes and locations using these R-Trees.

III. SYSTEM MODULE

Below is the figure of system overview.

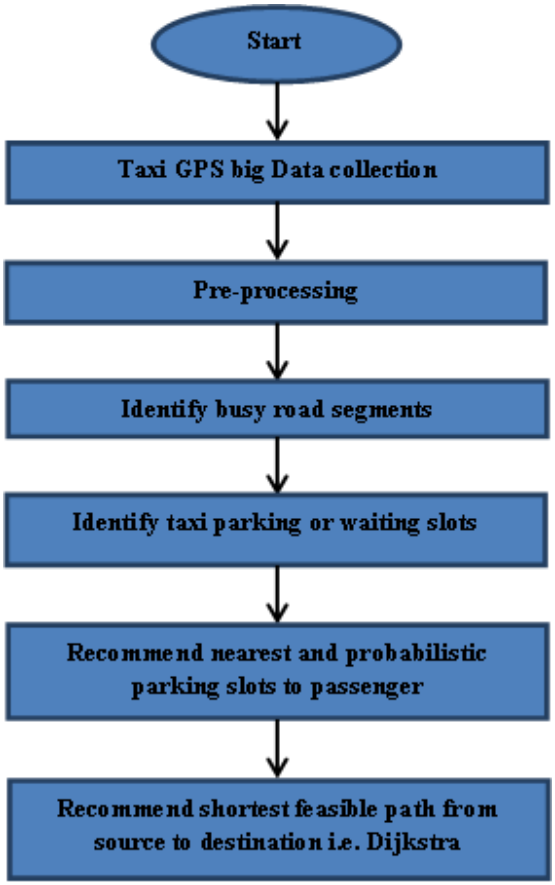

Figure 2: System Overview

OFF-LINE MINING

- **Parking Places Detection**

This section details the process for detecting parking status from a non-occupied trip and accordingly finding out the parking places in the urban area of a city based on a collection of taxi trajectories.

- **Candidates Detection**

Figure 3 demonstrates the parking candidate detection approach, given a non-occupied trip p1 → p2 → ·· ·→p7.

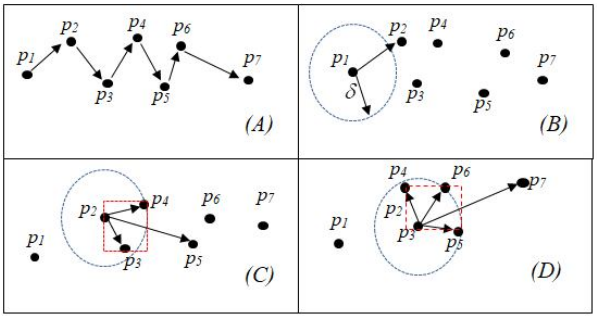

Figure 3: Parking candidates' detection

Algorithm 1: **ParkingCandidateDetection**

Input : *A road network U*, *atrajectoryTr*, *distance threshold* 5, *timethresholdT*

Output : *Asetofparking candidatesP* = {*P*}

*e*0, *M* *P*0, *Ilpe*0;

whilei < (*M* —1) *do*

j *i*1 – **1**;**flag** < —**false**;

whilej < *Mdo*

dist + –*Distance*(**pi ppj**);

if dist < **45then***j* *j***1**;**flag** = **true**;

else*break*;

ifp$_{i\ \_i}$*.t* — p_i*.t* > *Tand flag* = **truethen***foveae*!' *pointpETr*[*i*, .0*andp*0*Pdo*

P.Add (*p*); /**builda candidate*

if = *j* —1*then*

IP.Add(*MB*(*P*)); *P*0;

/**addthe minimumbounding boxofPintoP*

returnP

- **Filtering**

To reduce false selections, we design a supervised model for picking out the true parking status from the candidate sets, using the following features:

- **Spatial-Temporal features including**

1) Minimum Bounding Ratio (MBR). MBR is the area ratio between the bounding box of the road segment (MBRr) and the bounding box of the GPS points (MBRc) in the candidate set.
2) Average Distance: The average distance dc between points in the candidate set and their nearest road segments
3) Center Distance: The distance between center point in MBRc of the candidate set and the road segments.
4) Duration: The parking duration of a candidate.
5) Last Speed: The speed of the last point leaving a parking candidate.

- **POI feature**

.

As we know, a parking place is highly relevant with the points of interests (POI) around it, e.g., subway exits, theaters, shopping malls within 50 meters, shown in Figure 4 C). We employ the term frequency-

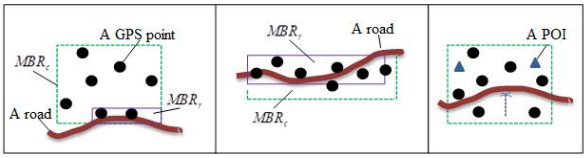

Figure 4: A) Real parking place B) Traffic jam C) Features

- **Dijkstra's Algorithm for shortest path selection**

To solve the single-pair shortest path problem, Dijkstra's Algorithm is the most commonly used algorithm. For a given source node, Dijkstra's Algorithm finds the path with the lowest net cost (sum of edge weights) to get to a particular destination node. The algorithm requires that all edge weights in the matrix be non-negative. By letting it run fully instead of stopping the algorithm once the desired destination is reached, Dijkstra's Algorithm can also be used to solve the single-source shortest path problem, giving the shortest path from a defined start node to all other nodes in the matrix. Dijkstra's Algorithm starts by labeling the value for the starting point as zero and temporarily labeling the value for every other node as infinity. It then recalculates the value for each node directly connected to the starting node as the starting node's value (zero) plus the distance from the starting node to that node. The starting node is then labeled as visited and will not enter the calculations again. Meanwhile, of all the remaining nodes, the one with the smallest value (which is the shortest distance from the starting point) becomes the current node of interest. The values are recalculated for all nodes directly connected to the current node. The values for each of these connected nodes equal the value of the current node plus the distance from the current node to each of them. The current node is then labeled as visited and does not enter into future calculations. As before, the remaining node with the smallest value becomes the new current node. The values of all nodes directly connected to it are recalculated, leading to selection of the minimum as the next current node. This process continues until every node in the matrix has been labeled as visited. At this point, the value of each node equals the length of the shortest path from the starting node to it. The worst case performance of Dijkstra's original algorithm, first conceived in 1956, requires on the order of O (N2) calculations, where N is the number of nodes in the matrix. However, by implementing a Min-Priority Queue with a Fibonacci Heap along with Dijkstra's Algorithm, the solution can be reached on the order of O(E + N log N) calculations, where N is the number of nodes in the matrix and E is the number of edges between nodes. In the Min-Priority Queue with Fibonacci Heap method, the matrix is typically stored in the form of adjacency lists telling which nodes are directly connected to which. Certain values in this node heaps can be pre-calculated. Then, instead of recalculating values for almost every node in the matrix every time a new current node is considered, certain heaps are given priority as the most likely sources of the next current node. Dijkstra's Algorithm implementing Min-Priority Queue with a Fibonacci Heap is especially efficient in dispersed matrices, where all nodes are not connected to each other . This is considered the fastest single-source shortest path algorithm for matrices with non-negative edge weights. Because of its speed and versatility, Dijkstra's Algorithm with Min-Priority Queue and Fibonacci Heap is widely used in network routing protocols, especially OSPF (Open Shortest Path First) and IS-IS (Intermediate System to Intermediate System), as well as most transportation routing applications.

IV. RESULTS AND DISCUSSIONS

In first step all the files are read into matlab and data, time and latitude-longitude parameters are concatenated into single matrices. Further these matrices are further processed to get the taxi spots. It removes the traffic jam stoppages etc. and provide the parking taxi spots.

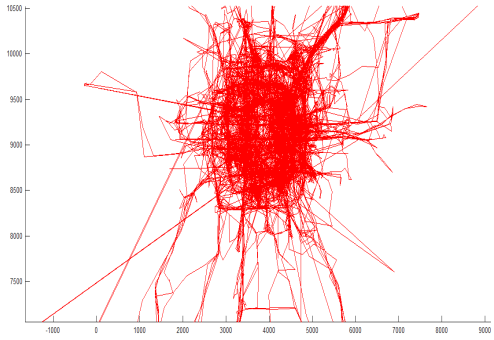

Figure 5: Trajectories of all the taxis in a given day

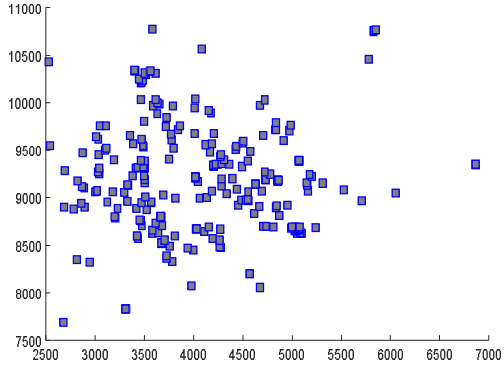

Figure 6: Parking candidates' detection

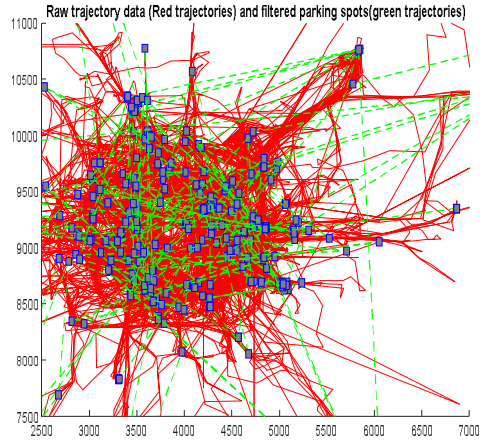

Figure 7: Raw trajectory data (Red trajectories) and filtered parking candidates (green trajectories)

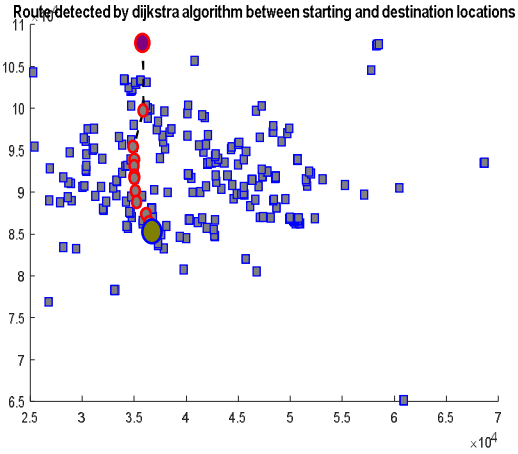

Figure 8: Route detected by dijkstra algorithm between starting and destination locations with SID and FID location

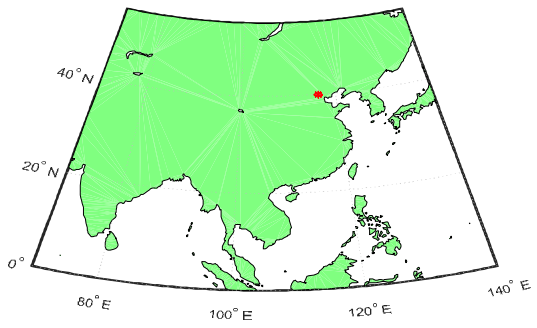

Figure 9: Route trajectory shown on Geoshow graph for SID and FID

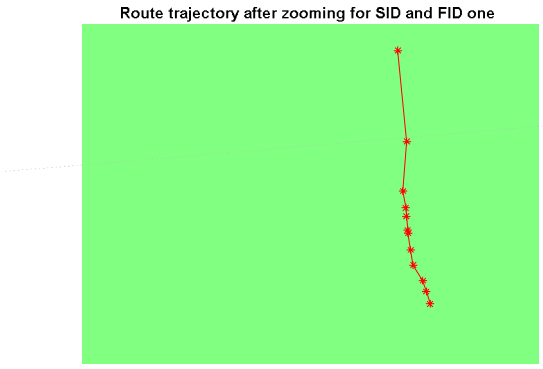

Figure 10: Route trajectory after zooming for SID and FID

Table 1: Performance evaluation using Distance to nearest parking place, Distance bw starting and destination point and Distance b/w starting and destination point for difference SID and FID coordinates

SID (Latitude, Longitude)	FID (Latitude, Longitude)	Distance to nearest parking place (in kms)	Distance bw starting and destination point (in degrees)	Distance bw starting and destination point (in Kms)
[39.65133,116.16822]	[39.86549,116.50373]	0.0069466	0.22980	25.5820
[39.99736,116.46728]	[39.85747,116.34369]	0.0140232	0.17070	19.0026
[39.88959,116.32478]	[39.86248,116.50631]	0.2039004	0.21702	24.1597
[39.87647,116.54499]	[39.99192,116.79514]	2.00133028181075	0.18254	20.3205

V. CONCLUSION

From the above study, it is concluded that a novel approach which is used to discover spatiotemporal patterns of household travel from the taxi trajectory dataset with a large number of point locations involves three critical steps: spatial clustering of taxi OD based on urban traffic grids to discover potentially meaningful places, identifying threshold values from statistics of the OD clusters to extract urban jobs-housing structures, and visualization of analytic results to understand the spatial distribution and temporal trends of the revealed urban structures and implied household commuting behavior. Based on the historical information of taxis and passengers, we propose parking candidate selection algorithm to discover popular locations so that a server can suggest routes and locations using these nodes. The proposed recommender system can assist both taxi drivers and passengers using a huge number of historical GPS trajectories of taxis. Specifically, on the one hand, given the geo-position and time of a taxicab looking for passengers, we suggest the taxi driver with a location, towards which he/she is most likely to pick up a passenger as soon as possible and maximize the profit of the next trip. This recommendation helps reduce the cruising (without a fare) time of a taxi thus saves energy consumption and eases the exhaust pollution as well as helps the drivers to make more profit. On the other hand, we provide people expecting to take a taxi with the locations (within a walking distance) where they are most likely to find a vacant taxicab. Using our recommender system, a taxi will find passengers more quickly and people will take a taxi more easily thereby reducing the supply/demand disequilibrium problem to some

extent. In the future, proposed taxi recommender can be deployed the in the real world so as to further validate and improve the effectiveness and robustness of this system.

REFERENCES

[1] V. Ceikute and C. S. Jensen "Routing service quality - local driver behavior versus routing services" In MDM (1), pages 97–106, 2013

[2] O. Andersen, C. S. Jensen, K. Torp, and B. Yang "Ecotour: Reducing the environmental footprint of vehicles using eco-routes" In MDM (1), pages 338–340, 2013

[3] Y. Zheng, Y. Liu, J. Yuan, and X. Xie "Urban computing with taxicabs" In Ubicomp, pages 89–98, 2011

[4] Huimin Lv, Fang Fang, Yishi Zhao, Yuanyuan Liu, and Zhongwen Luo, "A Performance Evaluation Model for Taxi Cruising Path Recommendation System" Published in: Pacific-Asia Conference on Knowledge Discovery and Data Mining PAKDD 2017: Advances in Knowledge Discovery and Data Mining pp 156-167

[5] Zheng Lyu, Yongxuan Lai, Kuan-Ching Li, Fan Yang, Minghong Liao, Xing Gao, "Taxi Route Recommendation Based on Urban Traffic Coulomb's Law" Published in: International Conference on Web Information Systems Engineering WISE 2017: Web Information Systems Engineering – WISE 2017 pp 376-390

[6] Fansheng Kong, Xiaola Lin, "Themethod and application of big data mining formobile trajectory of taxi based on MapReduce" Published in: Cluster Computing, 2018, pp 1–8

[7] Guoyang Qin, Tienan Li, Bin Yu, Yunpeng Wang, Zhenhua Huang, Jian Sun, "Mining factors affecting taxi drivers' incomes using GPS trajectories" Published in: Transportation Research Part C: Emerging Technologies Volume 79, June 2017, Pages 103-118

[8] Mohammad Saiedur Rahaman, Yi Mei, Margaret Hamilton, FloraD. Salim, "CAPRA: A contour-based accessible path routing algorithm" Published in: Information Sciences 385–386 (2017) 157–173

[9] Lin Yang, Mei-Po Kwan, Xiaofang Pan, Bo Wan, Shunping Zhou, "Scalable space-time trajectory cube for path-finding: A study using big taxi trajectory data" Published in: Transportation Research Part B: Methodological Volume 101, July 2017, Pages 1-27

[10] J. Dai, B. Yang, C. Guo and Z. Ding, "Personalized route recommendation using big trajectory data," *2015 IEEE 31st International Conference on Data Engineering*, Seoul, 2015, pp. 543-554.

[11] Feng Mao, Minhe Ji, Ting Liu, "Mining spatiotemporal patterns of urban dwellers from taxi trajectory data" Published in: Frontiers of Earth Science June 2016, Volume 10, Issue 2, pp 205–221

[12] N. J. Yuan, Y. Zheng, L. Zhang and X. Xie, "T-Finder: A Recommender System for Finding Passengers and Vacant Taxis," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 10, pp. 2390-2403, Oct. 2013

[13] Anastasios Noulas, Vsevolod Salnikov, Renaud Lambiotte, Cecilia Mascolo, "Mining open datasets for transparency in taxi transport in metropolitan environments" Published in: Noulas et al. EPJ Data Science (2015) 4:23

[14] Jian-Pan Li, Gwo-Jiun Horng, Yin-Jun Chen, Sheng-Tzong Cheng, "Using Non-cooperative Game Theory for Taxi-Sharing Recommendation Systems" Published in: Wireless Personal Communications June 2016, Volume 88, Issue 4, pp 761–786