

# Ridesharing Problem with Social Network

## INTRODUCTION

---

Ridesharing, or shared transportation, allows people to use private transport resources in a much more flexible, clever, and effective way. Two characters are often involved in this period. The first is driver, while the second is passenger. By using his or her own car, the driver can pick up orders from passengers, drive them to the destination, and gain benefits, generally considered as revenue. On the other side, the passengers can reach their destinations without purchasing private cars, and can even share a trip with other passengers, which again enlarges the utility of private vehicles.

After 2014, the research on ridesharing becomes much more complex and the aspect of ridesharing has become larger than before. The entropy of keywords has increased, and the quantity of articles in this field has increased exponentially. However, there are still new areas within to be explored. According to [1]'s investigation, dynamic networks, routing, location problem, social network, and collaborative consumption are all hot topics currently. This trend shows the public concern on social relationship and its interaction instead of pure optimization problem. While with the development of those large ridesharing platforms such as DiDi and Uber, social safety and trust between driver and passengers has become a new problem. Some of them focus on the problem definition or platform ([2]), while some of them focus on algorithms and simulation ([3, 4]).

Since the diversity of objective function, there are many topics about ridesharing in literature. Some researches aim to minimize the total travel cost of all vehicles to potentially reduce the pollution and congestion status, while some focus on minimizing the waiting time of the passengers [5]. Also, all these objectives can be combined together to find a better solution for all people [6, 7, 8]. Either way, there are many factors to consider. If certain factors are considered as the objectives, then the others could be considered as the constraints (e.g., if a problem doesn't aim to reduce the passengers' detour, then the detour tolerance of the passenger could be a constraint).

[9] lists several algorithms including the exact method, heuristic method and meta heuristic method presented in several research fields like operation research, database, transportation and artificial intelligence. All these algorithms have a common operation: insert a new request to a vehicle's schedule. [10] concludes that this is a core operation named "insertion operator" in dynamic ridesharing and presenting a more efficient insertion operator helps reduce the computation cost. [11] presents a solution with a new perspective: Make route plan with detour instead of shortest path to maximize the expected number of compatible passengers in a single tour. There are also interesting ideas like [12], which proposes a greedy and a ranking approach for order dispatch and their corresponding pricing strategies, to maximize the overall utility of the auction, while ensuring desirable auction properties such as truthfulness and individual rationality, in the situation of shortage of vehicles. With new technologies advancing, [13] collects a set of researches concerning ridesharing using an Autonomous Vehicle Systems, potentially integrated with electricity power usage.

In our project, we are going to explore how the social network works in ridesharing based on existing methodology to solve a practical problem which is the lack of vehicle in rush hour. We are going to find an algorithm to solve it and develop a simple simulator to evaluate our solution.

## WHY IS SOCIAL NETWORK INVOLVED?

---

[5] mentioned that building trust among unknown passengers in online systems is a major challenge in shared mobility system. A clear trend for commuters in the same building is to form a shared mobility community to reduce transportation cost by sharing vehicles with friends (people close in the social network) [14]. Searching ridesharing groups based on communities would make ridesharing services more convenient, safer and attractive to the users [15]. Ridesharing with friends is a more acceptable solution for people who don't like to share vehicles with strangers and will be potentially more acceptable by more people [16].

[17] illustrates another reason. The relationship between friends can affect ridesharing, because they often share a similar initial position or destination. Meanwhile, this paper shows that the more friends a traveler has, the larger the number of travelers who are willing to use shared transportation is. This may further reduce the travel cost and enhance the advantages of ridesharing.

Basically, we would like to design a ridesharing system which is capable enough of providing high request matching rate. That means, we want the requests to be arranged successfully as many as possible. However, a trip in a car is not just a rush inside the city. We hope that the passengers can fully enjoy the ride so the system might take in more users in the future. As specified in the Utility-Aware paper [18], this value is named as the "Utility" for the passenger. This value is often related to other passengers the passenger travels with. If the passengers share common interests or they are in the same communities, the utility value is supposed to be higher. As a result, applying social network as a constraint helps the passengers achieve high utility values.

## EXISTING APPROACHES OF "RIDESHARING IN SOCIAL NETWORK"

---

In artificial intelligence research field, this problem is formulated as a Graph Constraint Coalition Formation problem and solved by an approximate method using branch and bound [14]. To ensure the information of users are kept secret only to other passengers close to them, [15] develops a CaRG (Community-aware Ridesharing Group) query to enhance the security level of information flowing in the ridesharing procedure, as well as reduce the cost by avoiding unnecessary computations. [18] considers even more to make sure the passengers enjoy the most. In this paper, not only social relationship matters, but also vehicle quality and even sceneries along the trip. All these factors are considered as utilities for passengers, and are categorized into three sections: vehicle-related utility, rider-related utility (rider as passengers) and trajectory-related utility. To maximize the passengers' overall utility, the authors first formulate an NP-hard problem – URR (Utility-aware Ridesharing on Road Networks), then propose assignment method as well as three efficient approximate algorithms to assign passengers to suitable vehicles with a high overall utility, subject to spatial-temporal and capacity constraints.

Among these papers, there are three which share the most similar topics with ours. The AAAI [14], Top-K [20] and the Utility-Aware [18]. However, they all have some limitations that we might explore for solutions.

For the AAAI paper [14], only direct friends are considered and the constraint is too tight such that the matching rate will decrease drastically. According to the KDD paper [21], only 9% of the users are sharing the same check-in locations with their friends. This means if a group of friends are going to the exact same destination, the matching rate will be really low. To deal with this issue, we first

relax the constraint of “same destination” so the problem suits the ridesharing scenario. Then we define certain groups or organizations to allow indirect relationship so the constraints are further relaxed.

For the Top-K paper [20], the number of “hops” in a social network graph is computed to represent the closeness among passengers. However, the computation needs access to the entire social network, which is not safe since we need to fetch other passengers’ data while computing the current two’s hop count. To solve this, we simply maintain a group/organization list as well as a friend list so we only use data from the direct two passengers we are dealing with.

For the Utility-Aware paper [18], a function called “Jaccard Similarity” is used to compute the closeness of two passengers. This is simply measuring the intersection of their friend lists. But the case that the two are direct friends seems to be ignored. To deal with this problem, we consider extend the Jaccard Similarity function to take both direct and indirect relationships into account.

## PROBLEM DEFINITION

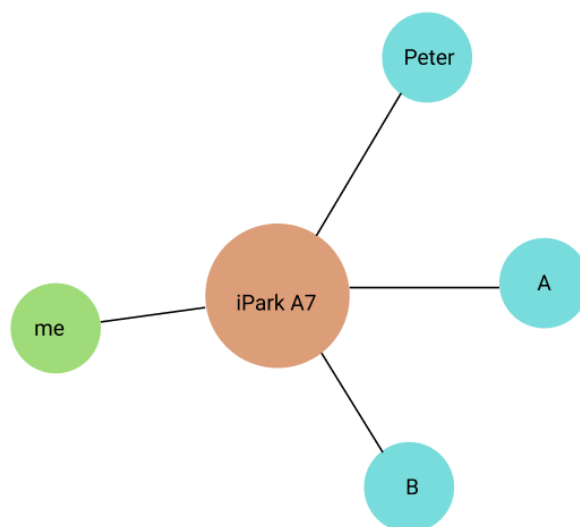
---

### Ridesharing in Social Network

Given a road network, a social network, a set of vehicles and a set of passengers with ridesharing requests, we try to find an arrangement schedule to maximize the overall utility of passengers as well as the request handling rate, which is the number of successful handle requests.

### Social Network

Besides the classical ridesharing problem, another concept taken as a constraint is the social network. The connection through network means that two nodes are “friends”. A passenger can only be matched to his/her direct friends and people in some social group he/she participates in. For example, as shown in the figure below, “me” knows “Peter”, but “me” doesn’t know A and B, but “me”, A and B are in the same social group which is “iPark A7”. Thus “me” can be matched with Peter, A or B.



In the future, we will consider adapting the Jaccard Similarity function to compute the closeness of different passengers.

## Dynamic Ridesharing

A list of requests from passenger will be handled by the ridesharing matching system. If a request is handled, the passenger sending this request will be assigned to a working vehicle nearby. Before the passenger arriving at the destination, the vehicle can also handle other requests, if those requests' sender can be matched with passengers already in the vehicle.

We use a **directed** weighted graph to represent the road network:  $G = (V, E, w)$ , where  $V$  is a set of vetices of the graph representing specific locations in the street network,  $E$  is a set of edges  $e = (u, v)$  connecting vetices in  $V$ , representing a path between two locations  $u$  and  $v$ , and  $w : e \rightarrow c$  is a weight function whose input is a path  $e$  and output is the travel cost  $c$  of the path.

Passenger  $p_k$ 's feasible matching passenger list is represented as:  $l_k = \{p_i, \dots, p_j\}$ .

A request  $r$  contains four information:  $r_i = (p_i, v_{s_i}, v_{d_i}, t_{r_i})$ , where  $p_i$  is passenger  $i$ ,  $v_{s_i}$  is the position where  $p_i$  is waiting for a vehicle,  $v_{d_i}$  is the position of  $p_i$ 's destination, and  $t_{r_i}$  is the requests's sending time. Real time: the request is invisible to the system if current timestamp is smaller than  $t_{r_i}$ . In the future, we also consider building time windows for the passengers which, in details, are the earliest pickup deadline and the latest drop-off deadline.

A vehicle's schedule  $s$  is represented by an ordered sequence of positions:  $s = \{v_{s_i}, \dots, v_{d_i}\}$ , where the elements  $v$  are waiting positions or destinations of its handling requests and the waiting position of a request should be placed in front of the destination of this request.

For each transfer event of a vehicle, there is a maximum flexible/detour time which specifies the largest acceptable extra time cost for the vehicle to handle requests from other passengers along the trip.

## FRAMEWORK DESIGN

---

Our framework uses an event-based traffic simulator, rather than the time-based one in Cargo, which we have been using in the last semester. The framework includes all the structures we need, the road network, social network, vehicles, requests and the optimizers for algorithms. The framework is coded in C++ and use a very fast indexing method called PHL (Pruned Highway Labelling) from [22]. Generally speaking, the framework is designed to run simulations very efficiently.

## BASELINE

---

In our framework, a cost-first greedy approach has been implemented as a baseline algorithm. The algorithm for each time picks out request which leads to the smallest extra cost on arrangement. We also have a random arrangement algorithm for simpler case testing.

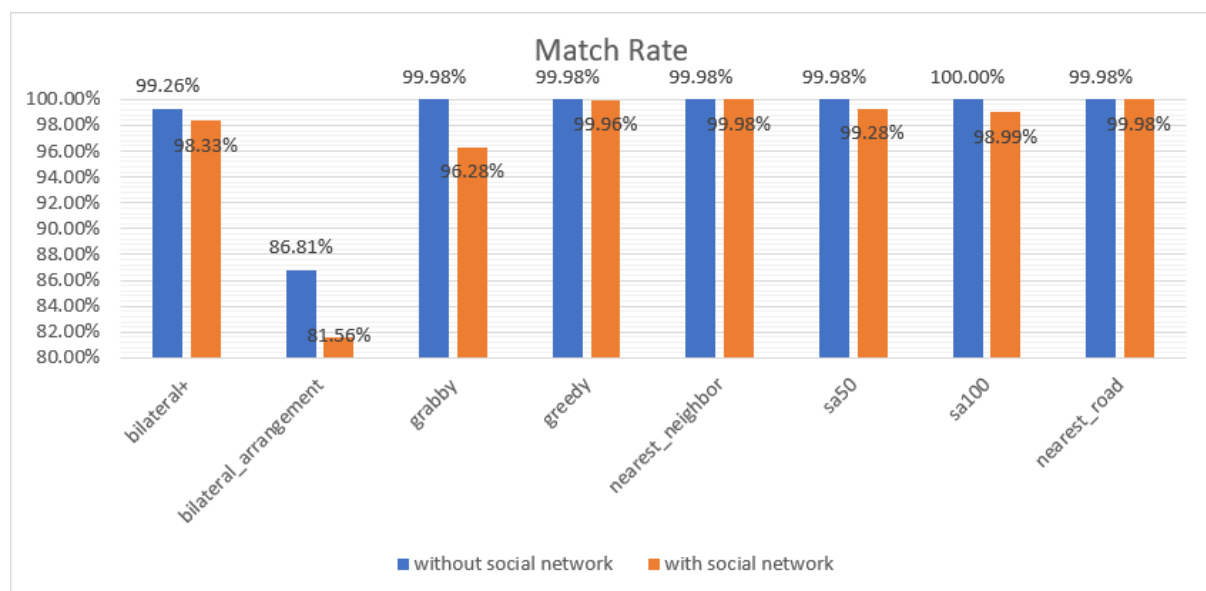
## EXPERIMENT SET

The experiment runs on previous' work, a platform which called Cargo. This simulator can test algorithms under dataset and show results. But the problem definition of Cargo doesn't as same as ours. So some changes should be added to Cargo.

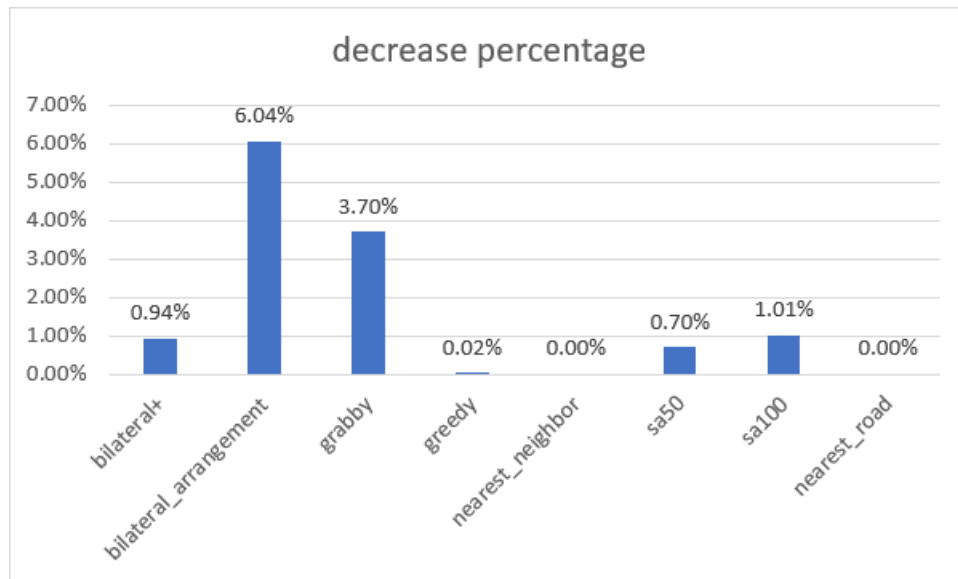
We modified and added a constraint of social network to Cargo. The assumption follows the previous definition. Only passengers in the same social network can be matched into the same vehicle. For new customers who is waiting for a vehicle, only the vehicle that contains passengers in the same social network can be sent to that customer and pick it up.

The test runs under several algorithms and different data. But the social network data is computed in another way. If some passengers have the same start location or destination, they are considered to be in the same social network. One customer can belong to several organizations.

The experiment first tests the result of all algorithms under no social network constraint, and compare them with the result under social-network constraint. Therefore, we can have a clear view of match rate loss after adding a new constraint. The basic result under no constraint can be found in following chart.



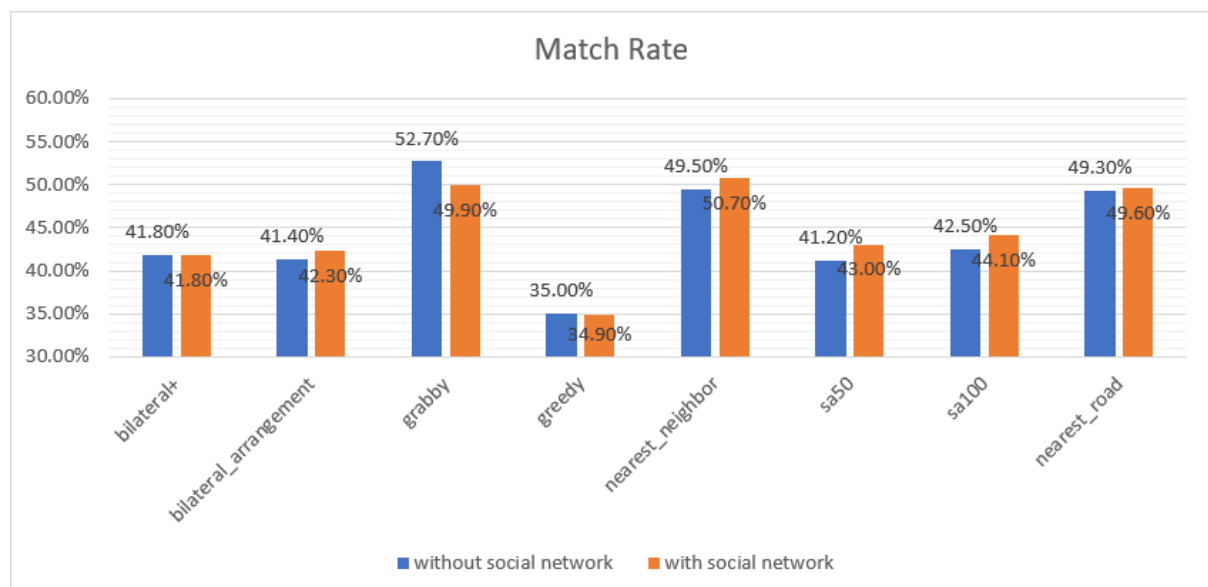
For the lower percent of match rate of social network, can be shown below.



From the result above, we can find that some algorithms even can achieve the same match rate after limiting the social network. The bilateral\_arrangement algorithm is affected the most, which lost 6.04% percent of match rate. And grabby is affect to reduce 3.70% as well. Those 2 algorithms are affected large. Meanwhile, bilateral+, sa50, and sa100 are all reduced match rate by about 1%. In practice, this value is small enough to negligible.

At the same time, we can see there are 3 algorithms that basically aren't affected. Greedy, nearest\_neighbor, and nearest\_road even have no change under social network as a constraint.

But this test is under Manhattan's taxi data, which has more taxies (1000) and less passengers (5033). What if the vehicles are not enough? Another test under SUSTech bus data is then simulated under social network constraint and without constraint. For 1000 students with only 25 vehicles, what would happen to the result? The comparison is below.



Interestingly, in some case, the social network constraint can make better choice than without constraint. This is because all the students' trips are separated into 3 social networks and their start locations and destinations are very close. If with location-based social network constraint, the algorithm can even perform better than before.

## FUTURE PLAN

---

Current simulator doesn't perform fast enough as it used time-based method to simulate events. The first thing we can do is to design another event-based simulator to speed up the simulation and optimize this step.

After comparing those algorithms, none of them are specially designed for social-network involved Ridesharing problem. The second thing we can do is to design a new algorithm which would perform well under social network and reach higher rate in special cases.

The third thing is to gather social network data from real environment which is combined with Ridesharing / taxi information. There isn't any ready-made data. So, we should either find an article about the location relationship of social network and their taxi habit, or gather real world data from social media or Ridesharing platform.

## CONCLUSION

---

The result is amazing, even with social-network constraint, we can see that the match rate doesn't lower much. Consider that using social-network can promote usage rate of the application, so with this function, the user may be more satisfying with the service, as well as the provider can acquire more users.

In the small dataset with less social network groups, after constraint, the match rate may even higher than before. That means the check of social network (physical address relationship) can affect and enhance the percentage of matching.

We can conclude that social network won't reduce user experience, but can enhance it. This isn't only a new field in Ridesharing research topic, but also a light sight of Ridesharing applications.

## SOURCE DATA

---

- <http://snap.stanford.edu/data/#socnets>: Social Networks, Road Networks
- <https://www.openstreetmap.org/>: Road Networks
- [https://chriswhong.com/open-data/foil\\_nyc\\_taxi/](https://chriswhong.com/open-data/foil_nyc_taxi/): NYC's Taxi Trip Data
- <https://socnetv.org/>: Social Network Visualizer - generate small dataset of social network
- Data from our school bus system with a Digital Twin simulator is also used

## REFERENCES

---

[1] Z. Sun, Y. Wang, H. Zhou, J. Jiao and R. Overstreet, "Travel behaviors, user characteristics, and social-economic impacts of shared transportation: a comprehensive review", *International Journal of Logistics Research and Applications*, pp. 1-28, 2019. Available: 10.1080/13675567.2019.1663162.

[2] H. Wang and H. Yang, "Ridesourcing systems: A framework and review", *Transportation Research Part B: Methodological*, vol. 129, pp. 122-155, 2019. Available: 10.1016/j.trb.2019.07.009.

- [3] D. Bertsimas, P. Jaillet and S. Martin, "Online Vehicle Routing: The Edge of Optimization in Large-Scale Applications", *Operations Research*, vol. 67, no. 1, pp. 143-162, 2019. Available: 10.1287/opre.2018.1763.
- [4] Held, Maximilian & Küng, Lukas & Çabukoglu, Emir & Georges, Gil & Pareschi, Giacomo & Konstantinos, Boulouchos. (2018). Future mobility demand estimation based on socio-demographic information: A data-driven approach using machine learning algorithms. 10.3929/ethz-b-000266653.
- [5] 2018 trb - Mourad, Abood & Puchinger, Jakob & Chu, Chengbin. (2019). A survey of models and algorithms for optimizing shared mobility. *Transportation Research Part B: Methodological*. 10.1016/j.trb.2019.02.003.
- [6] Y. Tong, Y. Zeng, Z. Zhou, L. Chen, J. Ye and K. Xu, "A unified approach to route planning for shared mobility", *Proceedings of the VLDB Endowment*, vol. 11, no. 11, pp. 1633-1646, 2018. Available: 10.14778/3236187.3236211.
- [7] P. Cheng, C. Feng, L. Chen and Z. Wang, "A Queueing-Theoretic Framework for Vehicle Dispatching in Dynamic Car-Hailing," 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macao, Macao, 2019, pp. 1622-1625.
- [8] L. Chen, Q. Zhong, X. Xiao, Y. Gao, P. Jin and C. S. Jensen, "Price-and-Time-Aware Dynamic Ridesharing," 2018 IEEE 34th International Conference on Data Engineering (ICDE), Paris, 2018, pp. 1061-1072.
- [9] James J. Pan, Guoliang Li, and Juntao Hu. 2019. Ridesharing: simulator, benchmark, and evaluation. *Proc. VLDB Endow.* 12, 10 (June 2019), 1085–1098. DOI:<https://doi.org/10.14778/3339490.3339493>
- [10] Y. Xu, Y. Tong, Y. Shi, Q. Tao, K. Xu and W. Li, "An Efficient Insertion Operator in Dynamic Ridesharing Services," 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macao, Macao, 2019, pp. 1022-1033.
- [11] Chak Fai Yuen, Abhishek Pratap Singh, Sagar Goyal, Sayan Ranu, and Amitabha Bagchi. 2019. Beyond Shortest Paths: Route Recommendations for Ride-sharing. In *The World Wide Web Conference (WWW '19)*. Association for Computing Machinery, New York, NY, USA, 2258–2269. DOI:<https://doi.org/10.1145/3308558.3313465>
- [12] L. Zheng, P. Cheng and L. Chen, "Auction-Based Order Dispatch and Pricing in Ridesharing," 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macao, Macao, 2019, pp. 1034-1045.
- [13] L. Zhao and A. Malikopoulos, "Enhanced Mobility With Connectivity and Automation: A Review of Shared Autonomous Vehicle Systems", *IEEE Intelligent Transportation Systems Magazine*, pp. 0-0, 2020. Available: 10.1109/mits.2019.2953526.
- [14] Filippo Bistaffa, Alessandro Farinelli, and Sarvapali D. Ramchurn. 2015. Sharing rides with friends: a coalition formation algorithm for ridesharing. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15)*. AAAI Press, 608–614.
- [15] Samiul Anwar, Shuha Nabila, and Tanzima Hashem. 2017. A Novel Approach for Efficient Computation of Community Aware Ridesharing Groups. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17)*. Association for Computing Machinery, New York, NY, USA, 1971–1974. DOI:<https://doi.org/10.1145/3132847.3133117>



- [16] Y. Wang, S. Winter and N. Ronald, "How much is trust: The cost and benefit of ridesharing with friends", *Computers, Environment and Urban Systems*, vol. 65, pp. 103-112, 2017. Available: 10.1016/j.compenvurbsys.2017.06.002.
- [17] P. Thaithatkul, T. Seo, T. Kusakabe and Y. Asakura, "Adoption of dynamic ridesharing system under influence of information on social network", *Transportation Research Procedia*, vol. 37, pp. 401-408, 2019. Available: 10.1016/j.trpro.2018.12.209.
- [18] Peng Cheng, Hao Xin, and Lei Chen. 2017. Utility-Aware Ridesharing on Road Networks. In *Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD '17)*. Association for Computing Machinery, New York, NY, USA, 1197–1210. DOI:<https://doi.org/10.1145/3035918.3064008>
- [19] Choromanski, Krzysztof & Matuszak, Michał & Miekisz, Jacek. (2013). Scale-Free Graph with Preferential Attachment and Evolving Internal Vertex Structure. *Journal of Statistical Physics*. 151. 10.1007/s10955-013-0749-1.
- [20] Y. Li et al., "Top-k Vehicle Matching in Social Ridesharing: A Price-aware Approach," in *IEEE Transactions on Knowledge and Data Engineering*.
- [21] Cho, Eunjoon & Myers, Seth & Leskovec, Jure. (2011). Friendship and Mobility: User Movement in Location-Based Social Networks. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1082-1090. 10.1145/2020408.2020579.
- [22] R. Zhong, G. Li, K.-L. Tan, L. Zhou, and Z. Gong. G-tree: an efficient and scalable index for spatial search on road networks. *TKDE*, 27(8):2175{2189,2015.