# The effects of including social factors in ride-matching algorithms on the performance and the quality of matches

**Omer Faruk Aydın** [*1] , **Ilgin Gokasar** [1]

*1Bogaziçi University, Faculty of Engineering, Department of Civil Engineering, İstanbul, Turkey*

**ABSTRACT**

Advancement in communication technologies has fostered alternative transport modes, such as ride-sharing. Ride-sharing aims to increase vehicle occupancy rates by matching riders with the drivers, who have empty seats on their vehicles and have similar routes and time schedules. Regarding to the success of a ride-sharing system, many researchers have been interested in efficient ride-matching algorithms. Ride-matching optimization problem is considered as NP-Hard Problem. In most of the ride-matching algorithms in the literature, to be able find matches at short notice some parameters were omitted. Hence, social characteristics and choices of participants, such as gender, age, employment status and willingness to socialize, were not included in many ride-matching algorithms. In this paper, the effects of including such factors in a ride-matching algorithm on the performance and the quality of the matches are investigated. Several ride-matching algorithms in the literature are simulated with randomly generated data. The simulation results show that when social factors are included the computation times and the quality of the matches increase significantly.

## 1. INTRODUCTION

Traffic congestion is one of the most important problems of modern cities. A significant shift from public transport towards private vehicles has been observed despite of increasing oil prices, parking problem and traffic congestion (European Environment Agency, 2005). On the other hand, while most private vehicles can transport 4 passengers, occupancy rates in private vehicles are very low. Occupancy rates in private vehicles were found to be 1.45 in European Countries in 2015, where this ratio is 1.42 in Germany, 1.3 in Netherlands and 1.58 in the UK (European Environment Agency, 2015). Many alternative transport modes have been studied such as ride-sharing to decrease traffic congestion by increasing occupancy rates.

Ride-sharing can be defined as matching riders who have no vehicle, with the drivers who have empty seats in their vehicles and have similar itineraries and time schedules. The history of ride-sharing can be traced back to the 1940s. At the time it was applied to conserve resources during World War II (Chan and Shaheen, 2012). Rapid growth in smartphone technologies and software packages made building an advanced ride-sharing system possible. Technological changes, such as

app-based shared transportation, growth of cloud computing, advanced navigation services and data sharing, contribute the growth of innovative shared transportation services (Shaheen et al. 2018). In recent years, many leading ride-sourcing companies, such as Uber and Lyft, have increasingly focused on ride-sharing and they have launched smartphone applications (e.g., UberPOOL, ExpressPOOL, Lyft Line) to allow potential users finding matches and lowering their travel costs from 25% to 60% (Shaheen and Cohen 2018). These applications also enable dynamic route changes.

Building an automated ride-sharing system requires ride-matching algorithms, which can optimize matches between drivers and riders at short notice. Automated ride-matching, which optimally matches riders and drivers in real-time, plays a key role in ride-sharing (Agatz et al. 2012).

Dynamic ride-matching algorithms are very complicated systems and require a lot of attention of the researchers to overcome challenges. Ride-matching optimization problem has been considered as non-deterministic polynomial-time hard (NP-hard) problem (Gu et al. 2018; Qian et al., 2017; Herbawi and Weber, 2012). To be able to offer feasible solution approaches, some parameters such as social characteristics and

choices of the participants and/or some transportation modes, such as multiple rider or multi-hop, are omitted in these problems. There are some algorithms in the literature, which considers these parameters and modes, but most of these studies did not consider computation times.

To maximize system benefits in ride-matching algorithms, a previous study has proposed a novel approach to solve the ride-matching problem by modeling it using a traditional maximum-weight bipartite matching algorithm (Agatz et al. 2011). This algorithm is based on a single rider-single driver match. It is demonstrated that the weighted bipartite matching algorithm can be used for ride-matching; however, this algorithm requires long processing times because it calculates distance savings for each rider-driver pair to determine distance savings. The algorithm also omits multiple riders-single driver matches and ignores individual preferences to simplify the problem. This algorithm aims to increase the number of matches. This approach would not encourage people to be included in ride-sharing systems; even should users specify a deadline for their travel request, they do not like to wait long (Nielsen et al. 2015). This study is extended by adding social parameters as constraints, but solution approach for this problem is not offered (Ghoseiri et al. 2011).

A novel biosequence based ride-matching algorithm is created to optimize matches when considering participants' gender, age, employment status and social tendencies (Aydin 2019). This algorithm aims to maximize social compatibility between drivers and riders. A biosequence matching algorithm, namely Needleman-Wunsch algorithm, is used to check the similarity of routes of drivers and riders. First comes first served approach is utilized in this algorithm to solve the problem at short notice.

In this paper, several dynamic ride-matching algorithms in the literature are modelled with different settings, such as including social parameters, using different driver capacities and allowing multiple riders. To evaluate effects of including social parameters, these algorithms are simulated using the same data and performance of the algorithms and quality of matches are evaluated.

The remainder of the paper is structured as follows. In Section 2, the ride-matching algorithms are presented. Section 3 describes the details of the simulation process. In Section 4 and 5, performance of the ride-matching algorithms and quality of matches are analyzed. In Section 6, the findings of the study are summarized and directions for the future studies are discussed.

## 2. RIDE-MATCHING ALGORITHMS

### 2.1. Biosequence Based Ride-Matching Algorithm

The biosequence based ride-matching algorithm utilizes social parameters, namely gender, age, employment status and willingness to socialize. The objective function of this algorithm is to maximize social compatibility between drivers and riders (Aydin 2019).

To score social compatibility, a parameter, Joint Socialness Score (JSS), is defined. The JSS is calculated as follows:

$$\gamma^{rd} = x_g^{rd}\gamma_g^r\gamma_g^d + x_a^{rd}\gamma_a^r\gamma_a^d + x_w^{rd}\gamma_w^r\gamma_w^d + x_\sigma^{rd}\gamma_\sigma^r\gamma_\sigma^d. \qquad (1)$$

In Eq. (1), weights of the social parameters of the rider (r) and feasible driver (d) are multiplied to calculate the JSS, $\gamma^{rd}$, The social parameters are gender weight $\gamma_g$, age weight $\gamma_a$, employment status weight $\gamma_w$ and willingness to socialize weight $\gamma_\sigma$. The variable x equals to positive one if the social characteristics are the same and negative one if they are different. Table 1 shows a sample calculation of the JSS.

**Table 1.** An illustrative example of the computation of JSS

|  | Driver *d1* |  | Rider *r2* |  |  |  |
|---|---|---|---|---|---|---|
|  | Char. | Factor | Char. | Factor | *x* | Scores |
| Gender | male | 1 | female | 5 | -1 | -5 |
| Age | 18-25 | 3 | 25-40 | 4 | -1 | -12 |
| Employment | TAU | 4 | TAU | 4 | 1 | 16 |
| Socialness | Yes | 5 | Yes | 3 | 1 | 15 |
| Total score |  |  |  |  |  | 14 |

In Table 1, driver d1 represents a male driver with an age of between 18-25 who works at the TAU. Driver d1 claims that the weights of a rider's gender, age range and working place are one, three and four out of five, respectively. Driver d1 also claims that he wants to meet a new person on a ride with a weight factor of five. On the other hand, rider r1 represents a female rider with an age of between 25-40 who also works at TAU. Her weight factor for willingness to meet a new person on a ride is three. As mentioned previously, the variable $x^{rd}$ equals to positive one if social characteristics of a driver and a rider are the same and negative one otherwise. In the example given in Table 1, $x^{rd}$ is found to be negative one for gender and age because the driver and the rider's gender and age range are not the same. $x^{rd}$ is found to be positive one for employment and socialness because they are working at the same location and they both want to meet with new people on a ride. In this situation, the score for gender becomes 1x5x(-1) = (5). When the scores for the other characteristics are calculated the same way, the JSS is calculated by simply adding the scores of each characteristic.

This algorithm assumes that routes of drivers are determined before the beginning of a ride and drivers do not change their routes to pick up a rider. This algorithm utilizes first comes first served approach, so that the matches found by the algorithm may not give the optimal result when the overall system benefit is considered. On the other hand, due to first comes first served approach, the algorithm shows good computation time performance. The social parameters are not utilized as constraints. A match is considered as feasible if the routes and time schedules of the participants are compatible and capacity of the vehicle is adequate.

There are some limitations of this algorithm. First of all, this algorithm utilizes greedy-heuristic approach that does not guarantee the optimal matches or maximum number of matches. This would cause decrease in quality

of matches, when the objective function is set to maximizing JSSs. Secondly, there are no constraint for social factors. This assumes that all participants will accept their matches, even if their JSSs are low. This case may not be valid in real-life, so the performance of the algorithm may decrease.

The JSSs are calculated separately for each rider. First, the objection function of the rider with the earliest trip announcement time is calculated. After this rider is matched using the objective function, the next rider is selected, and the process is repeated. The pseudo codes and flowchart of the biosequence based ride-matching algorithm are depicted in Fig. 1 and 2.



**Figure 1.** Pseudo code of the biosequence based ride-matching algorithm (Aydin 2019)
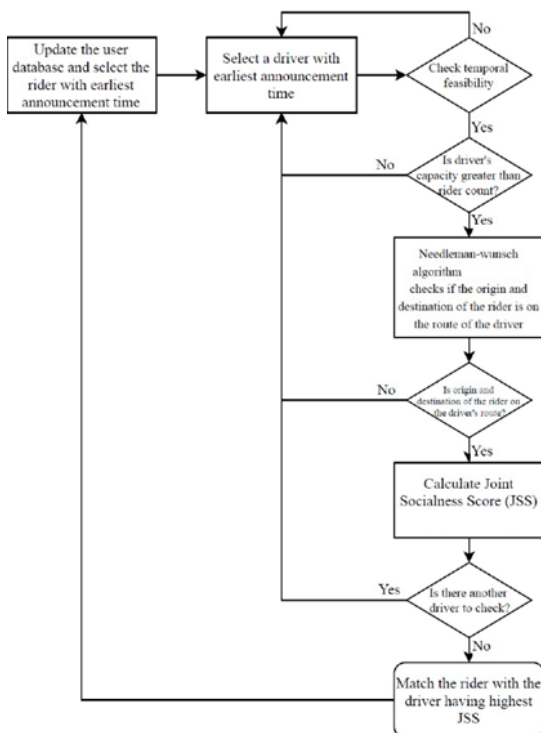


**Figure 2.** Flowchart of the biosequence based ride-matching algorithm (Aydin 2019)

## 2.2. Weighted Bipartite Ride-Matching Algorithm

The weighted bipartite matching algorithm can be constructed using several different objective functions, such as maximizing total distance savings, number of matches or fuel savings. In this paper, one of most preferred objective function, maximizing system-wide distance savings, are selected. This algorithm allows only single rider-single driver matches, because drivers change their routes to pick up riders. It also ignores the choices of participants. This study is extended by adding social parameters as constraints (Ghoseiri et al. 2011).

The algorithm builds arcs between each rider and each driver. These arcs are considered feasible if they create positive distance savings. Distance savings are calculated using the following equation (Agatz et al. 2011):

$$\sigma_{(i,j,k,l)} = d_{o_i d_i} - (d_{o_i k} + d_{kl} + d_{l d_i}) + \sum_{j \in R} (d_{o_j d_j} - (d_{o_j k} + d_{l d_j})). \quad (2)$$

In Eq. (2), distance savings are calculated for the scenario in which driver i picks up rider j from point k and drops him or her off at point l. To maximize system-wide distance savings, the calculations of distance savings are performed for all possible matches before any match is finalized. The matches are then finalized, starting with the match that offers the highest distance savings. Since all participants in the system must wait for the algorithm to make calculations for all possible matches, it takes a relatively long time to find a match for a participant.

There are some limitations of this algorithm. This algorithm utilizes heuristics that may not offer the optimal solutions. Furthermore, number of matches decreases dramatically when social factors are included as constraints; because, many feasible matches are eliminated, even if only one of the social characteristics is not compatible.

The pseudo code and flowchart of the weighted bipartite algorithm including social factors are presented in Fig. 3 and 4.

## 3. SIMULATION PROCESS

All matching algorithms were modeled in Python 2.7. Their performances were measured on a computer with an i5 2.7 GHz processor and 8 GB of RAM. All ride-matching algorithms were simulated using the same data and the same computer. To conduct a computational study, 1000 drivers and 1000 riders were randomly generated including their routes, origin/destination locations and social parameters. Drivers and riders are selected randomly from the generated data pool.

The algorithms were tested using different scenarios that are created with different number of riders, drivers, capacities of drivers and JSS limits. To measure the effects of including social parameters in an algorithm the biosequence based algorithm and weighted bipartite matching algorithm are simulated. Each case is simulated five times to eliminate the effects of stochastic behavior.

## 4. PERFORMANCE OF THE RIDE-MATCHING ALGORITHMS

Computation times of the matching algorithms with different combinations of independent variables are examined using multiple regression analysis. Independent variables are number of riders, number of drivers, capacity of each driver and JSS limits. Model summary and regression analysis for biosequence including social factors model results are given in Table 2 and 3, respectively.

```
m, n = number of riders, number of drivers
for j in range(0,n) do
   if driver capacity > 0 then
      for i in range(0,m) do
         if distance savings (Eqn. 5.1) ≥ 0 then
            if routes are compatible then
               if social choices are compatible then
                  add to feasible matches
               end if
            end if
         end if
      end for
   end if
end for
for k in range(0,feasible_matches_count) do
   Select the match with max distance saving
   if driver_capacity > 0 then
      if driver is unmatched then
         Match the driver with the rider
         Eliminate the matched rider from the list
         driver_capacity = driver_capacity − 1
      else if driver is matched with other riders then
         if social choices are compatible between riders then
            Match the driver with the rider
            Eliminate the matched rider from the list
            driver_capacity = driver_capacity − 1
         end if
      end if
   end if
end for
```

**Figure 3.** Pseudo code of the weighted bipartite matching algorithm (Aydin 2019)

$R^2$ values of 0.88 shown in Table 2 states that the regression model explains 88% of the relationship between coefficients and computation times.

**Table 2.** Model summary of the regression for computation times

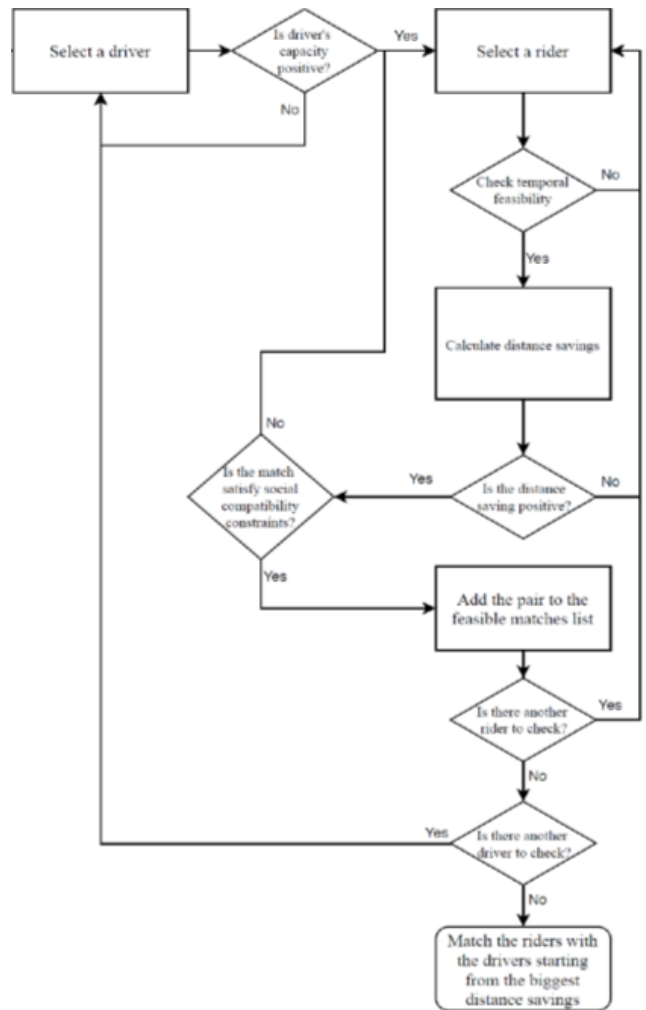| Model | R | R Square | Adjusted R Square | Std. Err. of the Estimate |
|---|---|---|---|---|
| Biosequence incl. social factors | 0.938 | 0.880 | 0.879 | 2.3916 |
| Biosequence excl. social factors | 0.938 | 0.879 | 0.878 | 1.6465 |
| Bipartite incl. social factors | 0.999 | 0.999 | 0.999 | 0.7372 |
| Bipartite excl. social factors | 0.998 | 0.996 | 0.996 | 0.9086 |



**Figure 4.** Flowchart of the weighted bipartite matching algorithm (Aydin 2019)

**Table 3.** Coefficients of the regression for computation times

| Model: Bioseq. Incl. Social | Unstandardized Coefficients B | Std. Error | Standardized Coefficients Beta | t | Sig. |
|---|---|---|---|---|---|
| 1   (Constant) | -22.33 | 0.919 | | -24.52 | 0.00 |
| Number_of _riders | 0.188 | 0.005 | 0.757 | 38.15 | 0.00 |
| Number_of _drivers | 0.216 | 0.005 | 0.873 | 43.96 | 0.00 |
| Capacity | 0.668 | 0.146 | 0.082 | 4.58 | 0.00 |
| JSS_Limit | -0.009 | 0.004 | -0.050 | -2.62 | 0.01 |

Table 3 shows that number of riders and drivers, capacities and JSS limits are significantly important for the computation times. Computation times increase as

the problem sizes and/or JSS limits increase. The relationship between number of riders and computation times are depicted in Fig. 5.

Figure 5 shows that including social parameters increases computation times of the weighted bipartite algorithm and biosequence based algorithm. The descriptive analysis for each algorithm is given in Table 4.
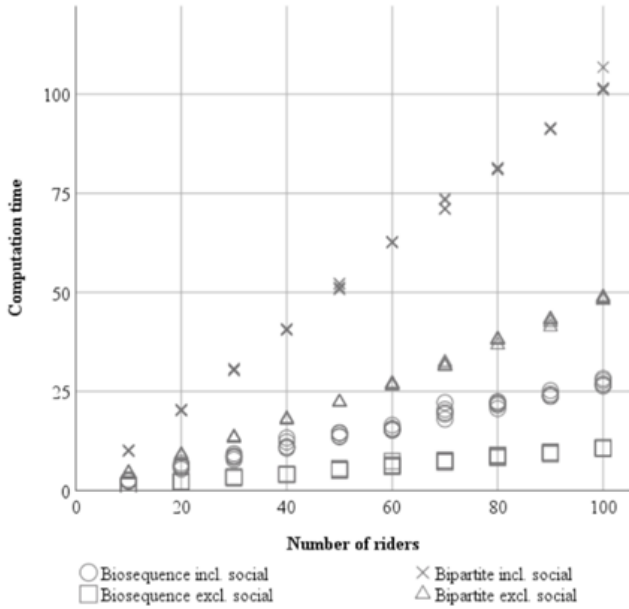


**Figure 5.** Computation times versus number of riders for the algorithms

Mean computation times given in Table 4 states that including social factors causes 71% increase in computation times for biosequence based algorithm and %220 increase for weighted bipartite algorithm. Average computation times of biosequence based algorithm are significantly lower than weighted bipartite algorithm because biosequence based algorithm utilizes first comes first served approach.

**Table 4.** Descriptive analysis of computation times for each algorithm

|  | N | Mean | Std. Deviation | Std. Error |
|---|---|---|---|---|
| Biosequence inc. social | 412 | 13.9022 | 6.8767 | 0.3388 |
| Biosequence exc. social | 413 | 8.1387 | 4.7104 | 0.2318 |
| Bipartite inc. social | 413 | 68.5130 | 31.9897 | 1.5741 |
| Bipartite exc. social | 430 | 31.1825 | 14.6050 | 0.7043 |

## 5. QUALITY OF MATCHES

The importance of social parameters for matching riders and drivers in a ride-sharing system has been discussed in the literature. To achieve critical mass in ride-sharing, considering social parameters can be seen as one of the key factors (Agatz et al. 2012; Shaheen et al. 2018). In this paper, the quality of matches found by the algorithms are measured by analyzing the similarity between their choices and the matches. The similarities

are scored by calculating the average JSSs of the matches found by the algorithms.

The simulation results of the biosequence algorithm and weighted bipartite algorithm including or excluding social parameters are shown in Fig. 6. In this Figure, the average of JSS scores of the matches found by the algorithms under different JSS limits are depicted.
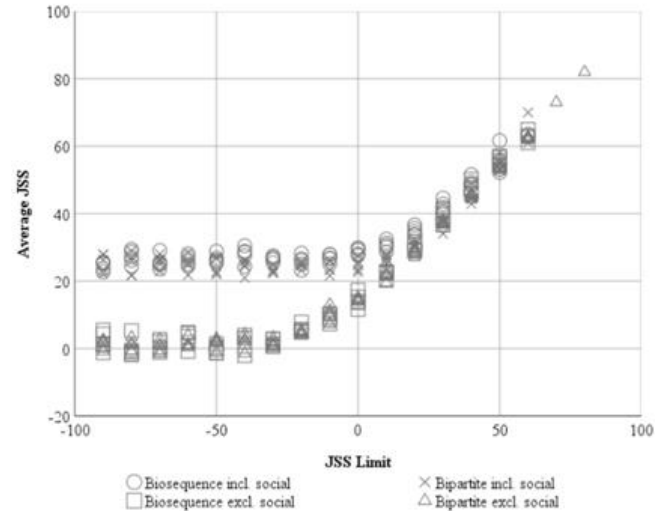


**Figure 6.** Average JSSs versus JSS limit for the algorithms

Figure 6 shows that including social parameters in the algorithms show better JSSs, even should there is no limit for JSS. This is because, the objective function of the biosequence matches riders and drivers, who have the highest JSS. Similarly, social parameters constraints used by the weighted bipartite algorithm eliminate the matches when there are one or more social parameters that are not compatible for drivers and riders. When there are no JSS limit, the average JSSs found by the algorithms are 26.43, -0.18, 23.65 and 1.32 for the biosequence algorithm including social parameters, biosequence based with excluding social parameters, weighted bipartite algorithm including social parameters and weighted bipartite algorithm excluding social parameters, respectively.

When a JSS limit is set, average JSSs of the matches found by the algorithm increases because the matches having lower JSSs than the JSS limit are eliminated. Therefore, the number of matches decreases with increasing JSSs. The relationship between the number of the matched riders and JSS limit is shown in Fig. 7.
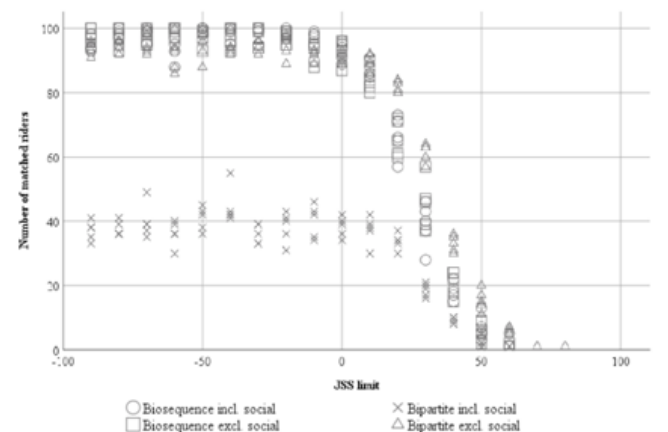


**Figure 7.** Number of matched riders versus JSS limit

Figure 7 shows that number of the matched riders significantly drops, when a JSS limit is set that is higher than the average JSSs found by the algorithms where no limit is set. The relationship between the average JSSs of the matches found the algorithms and number of matched riders are examined using regression analysis. Model summary and regression analysis results are given in Table 5 and 6, respectively.

**Table 5.** Model summary of the regression for average JSSs

| Model | R | R Square | Adjusted R Square | Std. Error of the Estimate |
|---|---|---|---|---|
| 1 | ,800 | 0.640 | 0.639 | 10.91 |

**Table 6.** Coefficients of the regression for average JSSs

| Model | | Unstandardized Coefficients | | Standardized Coefficients | | |
|---|---|---|---|---|---|---|
| | | B | Std. Error | Beta | t | Sig. |
| 1 | (Constant) | 52.18 | 1.314 | | 39.71 | 0.00 |
| | Number_of_matched_riders | -0.419 | 0.018 | -0.800 | -23.56 | 0.00 |

$R^2$ value of 0.640 shown in Table 5 states that the regression model explains 64% of the relationship between average JSSs and number of matched riders. Table 6 shows that number of matched riders are statistically significant to describe average JSSs. The regression results explain that when JSS limit is increased by one, number of matched riders decreases by 0.419. Consequently, it can be concluded that higher quality of the matches can be achieved but it costs reduction of number of matches.

## 6. CONCLUSION and DISCUSSION

Advancement in technology, especially smartphones, brings important advantages for ride-sharing. Dynamic ride-sharing system requires automated ride-matching. Ride-matching algorithms are at the center of ride-sharing systems that optimize matches between people with similar routes and schedules at short notice. Yet, computation times of these algorithms and the quality of the found matches are still studied by many researchers. In this paper, several ride-matching algorithms in the literature are modeled, simulation study is conducted and the results of the simulation study are analyzed. It is aimed (1) to present the effects of including social parameters and/or allowing different options, such as multiple rider-single driver match, on the performance of an algorithm and (2) to analyze the quality of the matches found by the algorithms.

The computation time of an algorithm is directly proportional to the number of computation. Therefore, including social parameters causes decrease in performance of the algorithms for biosequence based algorithm and weighted bipartite matching algorithm. Furthermore, first comes first served approach causes the most significant decrease in computation times but this approach does not promise to find the best possible

matches in the system. As a result, it can be concluded that including social parameters has negative effect on the performance, whereas utilizing first comes first served approach positive effect. Consequently, while creating an algorithm, a trade-off analysis should be done to decide which variables to include or exclude.

Achieving critical mass in ride-sharing has been widely discussed in the literature, yet it is not achieved in real world. To achieve critical mass in ride-sharing, social characteristics and choices of participants should be analyzed. Most ride-matching algorithms omitted choices of participants that may lead participants to reject the matches found by the algorithms. Thus, in this paper, the quality of the matches is measured by scoring the similarity between social characteristics and choices of the matched participants. The results show that including social parameters in the algorithms has significant effect on the quality of matches. When objective function of the algorithm is to maximize social compatibility of riders and drivers, the quality of matches significantly increases while the number of found matches did not change. Similarly, when the social parameters are used as constraints in the weighted bipartite algorithm, the quality of matches are found to be significantly higher than the same algorithm excluding social parameters. On the other hand, number of matches decreases, because if one of the characteristics and choices of the participants are not compatible, the match is eliminated. Thus, many matches having low quality score were eliminated and average quality of the matches increased. As a result, including social parameters in a ride-matching algorithm and/or setting a limit for social compatibility score are very important for the quality of the matches but number of found matches may decrease. In the future, case studies should be conducted to measure the necessity of setting a limit for social compatibility so that number of matches can be maximized while satisfying enough social compatibility quality for potential participants.

As a conclusion, the performance of a ride-matching algorithm and quality of the matches found by an algorithm depends on the variables used in the algorithms. Including social parameters increase the quality of matches significantly while decreasing computation time performance. Furthermore, first comes first served approach cause significant decrease in computation times. In the future, it would be intriguing to analyze trade-off costs of such variables on the performance and quality of the various matching algorithms. The importance of social parameters and travel choices of potential participants in real-life should be investigated by conducting surveys. Thus, better understanding of participants' travel behavior can be achieved.

## ACKNOWLEDGEMENTS

**REFERENCES**

Agatz N, Erera A L, Savelsbergh M W P & Wang X (2011). Dynamic ride-sharing: A simulation study in metro Atlanta. Procedia Social and Behavioral Sciences, 17, 532-550. DOI: 10.1016/j.sbspro.2011.04.530

Agatz N, Erera A, Savelsbergh M & Wang X (2012). Optimization for dynamic ride-sharing: A review, European Journal of Operational Research, 223(2), 295-303. DOI: 10.1016/j.ejor.2012.05.028

Aydin Ö F (2019). A Biosequence Based Ride-Matching Algorithm that Takes into Account Social Factors. PhD Thesis, Boğaziçi University, Department of Civil Engineering, Istanbul, Turkey.

Chan N D & Shaheen S A (2012). Ridesharing in North America: Past, Present, and Future. Transport Reviews, 32(1), 93-112. DOI: 10.1080/01441647.2011.621557

European Environment Agency (2005). Household consumption and the environment, 11, European Environment Agency.

European Environment Agency. (2015). Occupancy rates of passenger vehicles, European Environment Agency.

Ghoseiri K, Haghani A & Hamedi M (2011). Real-Time Rideshare Matching Problem. Final report, University of Maryland, Department of Civil and Environmental Engineering, MD, USA.

Gu Q-P, Liang J L & Zhang G. (2018). Algorithmic analysis for ridesharing of personal vehicles. Theoretical Computer Science, 749, 36-46. DOI: 10.1016/j.tcs.2017.08.019

Herbawi W M & Weber M (2012). A genetic and insertion heuristic algorithm for solving the dynamic ridematching problem with time windows. Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference, pp. 385-392. DOI: 10.1145/2330163.2330219

Nielsen J R, Hovmoller H, Blyth P-L & Sovacool B K. (2015). Of "white crows" and "cash savers:" A qualitative study of travel behavior and perceptions of ridesharing in Denmark. Transportation Research Part A: Policy and Practice, 78, 113-123. DOI: 10.1016/j.tra.2015.04.033

Qian X, Zhang W, Ukkusuri S V & Yang C (2017). Optimal assignment and incentive design in the taxi group ride problem. Transportation Research Part B: Methodological, 103, 208-226. DOI: 10.1016/j.trb.2017.03.001

Shaheen S & Cohen A (2018). Shared ride services in North America: definitions, impacts, and the future of pooling. Transport Reviews, 39(4), 1-16. DOI: 10.1080/01441647.2018.1497728

Shaheen S, Cohen A & Bayen A (2018). The Benefits of Carpooling. UC Berkeley: Transportation Sustainability Research Center. http://dx.doi.org/10.7922/G2DZ06GF.