A BIOSEQUENCE BASED DYNAMIC RIDE-MATCHING ALGORITHM THAT TAKES INTO ACCOUNT SOCIAL FACTORS

by

Ömer Faruk Aydın

B.S., Civil Engineering, Boğaziçi University, 2011

M.S., Civil Engineering, Boğaziçi University, 2013

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Civil Engineering
Boğaziçi University
2019

A BIOSEQUENCE BASED DYNAMIC RIDE-MATCHING ALGORITHM THAT TAKES INTO ACCOUNT SOCIAL FACTORS

ACKNOWLEDGEMENTS

I would like to express my gratitude to Assoc. Prof. Ilgın Gökaşar for her supervision and support throughout the Ph.D. program, during both pre-qualification process and the dissertation preparation.

I would also like express my sincere appreciation to Assist. Prof. Mehtap Işık for her advices for the preparation of this dissertation. I would also like to thank Assist. Prof. Mustafa Gökçe Baydoğan, Assist. Prof. Gürkan Günay and Assist. Prof. Ahmet Yıldız for their final comments.

I would also like to express my special thanks to my friends, who supported me throughout the Ph.D. program.

Most of all, I would like to express my special thanks to my dearest wife for supporting me spiritually throughout writing this dissertation and my whole life.

ABSTRACT

A BIOSEQUENCE BASED DYNAMIC RIDE-MATCHING ALGORITHM THAT TAKES INTO ACCOUNT SOCIAL FACTORS

Increasing traffic congestion and advancements in technology have fostered the growth of alternative transportation modes such as dynamic ride-sharing. Smartphone technologies enable dynamic ride-sharing, which aims to establish ride matches between people with similar routes and schedules at short notice. Many automated matching methods are designed to improve system performance, such as minimizing process time, minimizing total system cost or maximizing total distance savings; however, the results may not provide the maximum benefits for the participants. In this dissertation, an attempt is made to develop an algorithm to optimize matches when considering participants' gender, age, employment status and social tendencies. A biosequence algorithm, namely the Needleman-Wunsch algorithm, is used to quantify the similarity of participants' itineraries. A stated preference survey was conducted among 604 students and members of staff at Turkish-German University in 2018. An extensive simulation study was then performed by utilizing the survey data to compare the performance of the proposed algorithm with that of traditional bipartite and optimization algorithms. The simulation results indicate that when compared to the traditional bipartite and optimization algorithms, the proposed algorithm significantly increases performance in terms of computation times and the potential success rate of the matches. A sensitivity analysis for the proposed algorithm is also performed.

ÖZET

SOSYAL FAKTÖRLERİ DİKKATE ALAN BİYOLOJİK DİZİLİM HİZALAMASINA DAYALI BİR DİNAMİK YOLCULUK EŞLEŞTİRME ALGORİTMASI

Artan trafik sıkışıklığı ve teknolojideki gelişmeler dinamik yolculuk paylaşımı gibi alternatif yöntemlerin gelişmesine yol açmıştır. Akıllı telefon teknolojileri benzer rota ve zamanda seyahat edecek insanları kısa zaman zarfında eşleştirmeyi amaçlayan dinamik yolculuk paylaşımını mümkün kılmaktadır. Birçok otomatik eşleştirme algoritması işlem süresini, sistem maliyetini veya toplam mesafeyi en aza indirgemek gibi sistem performansını geliştirmek üzere tasarlanmıştır, ancak sonuçlar katılımcılar için en iyi faydayı sağlamayabilir. Bu tezde, katılımcıların cinsiyet, yaş, çalışma durumu ve sosyalleşme isteklerini baz alan bir algoritma geliştirilmiştir. Katılımcıların rotaları arasında mevcut benzerlikleri bulmak için Needleman-Wunsch isimli bir biyo-dizilim algoritması kullanılmıştır. 2018 yılında Türk-Alman Üniversitesi'nde toplam 604 öğrenci ve personelle belirli tercih anketi yapılmıştır. Önerilen algoritmanın performansını geleneksel bipartit ve optimizasyon algoritmaları ile karşılaştırmak için anket verileri kullanılarak kapsamlı bir benzetim çalışması yapılmıştır. Simülasyon çalışmasının sonuçları, önerilen algoritmanın geleneksel bipartit model ve optimizasyon algoritmalarına kıyasla işlem sürelerinde ve eşleşmelerin potansiyel başarı oranlarında önemli oranda artış sağladığını göstermiştir. Önerilen algoritmanın hassasiyet analizi de yapılmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a_s	Age of the participant s
c_d	Capacity (number of empty seats) of the vehicle of driver d
d	Driver
d_{kl}	Distance from location k to l
d_{ld_i}	Distance from location l to d
$d_{o_id_i}$	Distance from location o to d
d_{o_ik}	Distance from location o to k
d_s	Destination location of trip announcement s
D	Set of drivers
e_s	Earliest departure time of trip announcement s
f	Flexibility in time
g_s	Gender of the participant s
k_r	Rider Count of r i.e. number of riders who want to travel with r including r
l_s	Latest departure time of trip announcement s
M	Set of meeting points
n_d	Route of driver d
o_s	Origin location of trip announcement s
P	Set of participants
r	Rider
R	Set of riders
s	Trip announcement
S	Set of trip announcements
t_{od}	Travel time between Location o and d
T_{od}^{ED}	Earliest departure time with origin o and destination d
T_{od}^{LA}	Latest arrival time with origin o and destination d
T^{LD}	Latest departure time
w_s	Employment status of the participant s
x_a^{rd}	Variable i.e. 1 if ages of r and d are same, 0 otherwise

x_g^{rd}	Variable i.e. 1 if genders of r and d are same, 0 otherwise
x_w^{rd}	Variable i.e. 1 if employment status of r and d are same, 0 otherwise
x_σ^{rd}	Variable i.e. 1 if social tendency of r and d are same, 0 otherwise
X_{cod}	Capacity of each driver's vehicle
X_{jss}	Joint social score limit
X_{nor}	Number of riders
X_{nod}	Number of drivers
X_{nrj}	Number of riders who reject first match
X_{rc}	Rider count
Y_{com}	Computation time
Y_{jss}	Average joint social scores of matches
Y_{romr}	Ratio of matched riders
σ_s	Socialness i.e. willingness to meet new people of participant s
$\sigma_{(i,j,k,l)}$	Travel distance saving by rider j is served by driver i from location k to l
γ_a	Weight of the age
γ_g	Weight of the gender
γ_w	Weight of the employment status
γ_σ	Weight of the socialness
γ^{rd}	Joint socialness score of rider r and driver d

LIST OF ACRONYMS/ABBREVIATIONS

ANOVA	Analysis of Variance
JSS	Joint Socialness Score
LCS	Longest Common Subsequence
TAU	Turkish-German University

1. INTRODUCTION AND BACKGROUND

1.1. Motivation

Every day traffic congestion worsens and the rate of global warming accelerates. These factors have negative impacts on economics and social life of the regions which suffer from them. Therefore, policy makers seek certain strategies for these factors in congestion management and reduction perspective.

There are some existing strategies in congestion management perspective in the literature, such as constructing new highways, maintaining current transportation infrastructure, utilizing current highway capacities, reducing the private car use by developing some methods, implementing Intelligent Transportation Systems (ITS) and integrating bicycle into existing transportation network and so forth [3–8]. However, each strategy comes with both benefits and costs. To evaluate the impact of each given strategy and decision, benefit/cost analysis is commonly used in the literature. In this methodology, the benefits and costs of these strategies are calculated by considering some evaluation models, such as subjective scorings, life cycle cost analysis and so forth [9, 10]. Hence, the results can be used for the most appropriate decisions for benefits of the society. This dissertation mainly focuses on a methodology for the reduction in private car use.

Studies show that there is a significant shift from public transport towards the private vehicles despite rising fuel prices in EU-15 countries [11]. In Figure 1.1, it can be seen that especially after the year 2000, the crude oil prices has increased significantly.

While most vehicles can transport up to four passengers, the average passenger per vehicle ratio or private car occupancy rate in Europe was approximately 1.45 in 2015 (in Germany, 1.42; in the Netherlands, 1.38; and, in the UK, 1.58) [13]. Despite of low occupancy rates per vehicles, this value has continued decreasing as seen in Figure 1.2. Ride-sharing may have great potential in reducing traffic congestion.

Figure 1.1. Crude oil prices in last 30 years [12].

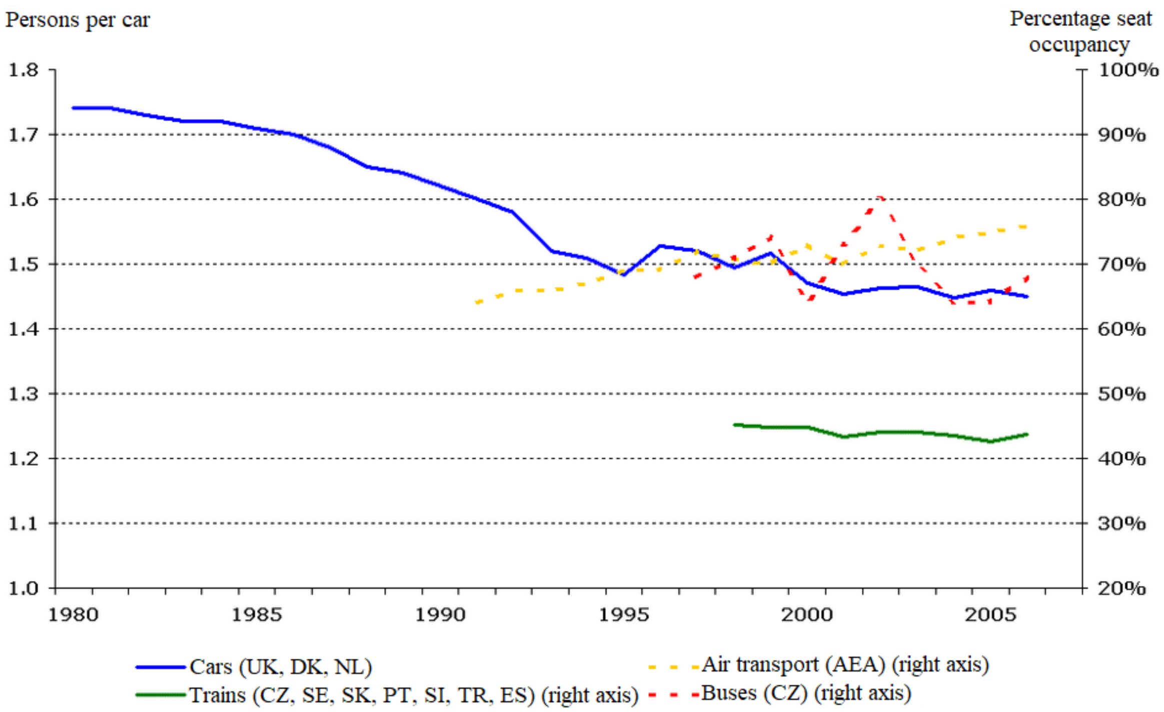

Figure 1.2. Occupancy rates in passenger transport [13].

Increase in number of private vehicles and decrease in occupancy rates of private vehicles led to increase in number of studies aiming to increase occupancy rates of private vehicles such as ride-sharing studies. Ride-sharing can be defined as matching riders, who have no vehicle, with the drivers, who have empty seats on their vehicles and have similar routes and time schedules. The applications of ride-sharing began during World War II and have continued to date.

1.2. History of Ride-Sharing

The history of ride-sharing goes back to the 1940s but the increasing use of smartphone devices and mobile applications has made ride-sharing more appealing [14–16]. The evolution of ride-sharing can be examined in five eras:

- The appearance of ride-sharing can be traced back to the 1940s, when it was done to conserve resources during World War II. At the time, the U.S. Office of Civilian Defense created a program called the "Car Sharing Club Exchange and Self-Dispatching System" to match riders and drivers via a bulletin board at their workplaces.
- The second era of ride-sharing occurred between the late 1960s and 1980, during the energy crisis.
- The third era of ride-sharing occurred in the 1980s and 1990s, when attempts at meeting transport demand focused on mitigating traffic congestion and improving air quality. However, there still existed a need for methods by which to match commuters at short notice.
- Later, the fourth era had come, in which ride-sharing systems focused on encouraging ride-sharing among commuters with the most reliable trip records. This era utilized the Internet for online matching and traveler information services.
- The fifth, current, era of ride-sharing includes the use of software packages, real-time services, financial incentives and social networking platforms [17]. This has resulted in a dramatic increase in dynamic ride-sharing studies in recent decades [3, 18].

Although advancement in technology in recent decades made building an advanced ride-sharing system possible, to-date critical mass for ride-sharing has not been achieved. There are many reasons for this fact, but automated ride-matching, which optimally matches riders and drivers in real-time, plays a key role to achieve critical mass in ride-sharing [3].

1.3. Studies on Ride-Matching Algorithms

Dynamic ride-sharing requires an automated matching system. This automated matching system brings riders and drivers with similar travel patterns and schedules together at very short notice. Dynamic ride-sharing systems are very complicated and require a great deal of attention of the researchers. Therefore, the success of a ride-sharing system depends on successful implementation of ride-matching [18]. In the literature, a number of studies on ride-sharing systems have identified the characteristics of ride-matching problems and proposed solution methodologies [3, 19]. In this section, several dynamic ride-matching algorithms in the literature are outlined and their advantages, disadvantages and importance are discussed.

1.3.1. Solution Approaches

There are many ride-matching algorithms in the literature, which are created based on optimization approach. To overcome present challenges, operation research community addressed this problem by building different ride-sharing models. Dynamic ride-matching includes many parameters, rendering the problem non-deterministic polynomial-time hard (NP-hard) [20–22]. Therefore, many solutions to the ride-matching problem that have been proposed in the literature use heuristics or meta-heuristics [14, 20–28]. Although heuristic and meta-heuristic methods offer feasible processing times, they may not find the best possible matches. To be able to offer feasible solution approaches, some parameters such as social characteristics of the users 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. As a result, there is still need for ride-matching algorithms, which are compatible with the real-life situations and that can be solved at reasonably short notice.

To maximize system benefits, 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 [15]. 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. Rolling horizon approach is introduced to force the matching algorithm to postpone finalization of the previously found matches until a deadline specified by the users. This technique aims to increase the number of matches [15]. The rolling horizon 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 [29]. This algorithm is extended by adding meeting points to increase the number of matches [30]. The algorithm allows multiple riders-single driver matches if the riders are waiting at the same location.

1.3.2. Objective Functions

Ride-sharing offers many advantages to the participants such as decreasing their travel time and cost. Ride-sharing also causes important system-wide benefits such as increase in occupancy rates in private vehicles and decrease in use of private vehicles. Many studies in the literature have employed one of the following objectives to solve ride-matching problems:

- *Maximizing total distance savings [15, 31–33].* Total distance savings represent the difference in vehicle-miles driven by all participants when they drive alone and when they share ride. This objective is important because it is directly related to the objective of reducing air pollution. It is also directly proportional to the objective function of minimizing travel cost.

- *Minimizing total travel time [34].* Total travel time defines the sum of travel times of the ride-sharing participants while travelling from their origin to destination locations. This is also related to the air pollution because vehicle emission is not only related to travelled distance but also vehicle speeds.
- *Minimizing total travel cost [35,36].* Total travel cost is the sum of the travel costs of all ride-sharing participants. Ride-sharing allows its participants to share their travel costs. This is an important parameter that can be an incentive for riders and drivers to be included in a ride-sharing system. Additionally, this would encourage private ride-share providers to build ride-sharing systems because they can make profit by taking a commission from each match they create.
- *Maximizing number of matches [35–37].* This objective is used to increase number of satisfied participants. More satisfied participants may attract even more participants into a ride-sharing system, so this is an important indicator to achieve critical mass.
- *Minimizing total waiting and delay time [28].* Total waiting and delay time is one of the reason people don't want to be included in a ride-sharing system. Therefore, this objective function aims to increase participation rate by minimizing participant inconvenience.

Most of the objective functions used in the literature focused mainly on system-wide benefits, whereas, potential participants in real-life wish to maximize individual benefits. This is an important challenge to overcome in order to apply the ride-matching algorithms in real-life. For example, a driver may want to be matched with a rider at the same gender even if another rider is better for maximizing distance savings or minimizing system-wide travel cost. To the best of writer's knowledge, social parameters are used in ride-matching algorithm as constraints [38], but they are not used in objective functions. In Table 1.1, objective functions used in some ride-matching algorithms in the literature are presented.

Table 1.1. Objective functions used in the literature.

Reference	Objective function
Agatz *et al.* (2011) [15]	Maximizing total travel distance savings
Cheikh and Hammadi (2016) [28]	Minimizing total waiting and delay times
Ghoseiri *et al.* (2011) [38]	Maximizing total number of matches
Masoud and Jayakrishnan (2017) [14]	Maximizing total number of matches
Najmi *et al.* (2017) [23]	Maximizing total number of matches
Stiglic *et al.* (2015) [30]	Maximizing total number of matches
Wang *et al.* (2017) [39]	Maximizing total travel distance savings

1.3.3. Parameters

Many parameters were used in ride-matching algorithms to present real-life instances. There were some parameters naturally used in all algorithms. Announcement time of a ride-sharing request, origin and destination locations, earliest departure time, latest arrival time or latest departure time are some examples to these parameters. On the other hand, there are parameters, which are only used in some of the algorithms. A literature review on use of these parameters are demonstrated in Table 1.2. Some examples to these parameters are as follows:

- *Joint rider.* In many ride-matching algorithms, it is assumed that a driver is matched with a rider at a time, while, in some other algorithms, multiple riders who wish to travel together can be matched with a driver. This parameter may decrease safety concerns of many participants [3]. This parameter can be used in different ways.
  - (i) Stiglic *et al.* [30] used joint rider parameter in their algorithm to present two different riders waiting at the same meeting point. These multiple riders can be matched with a driver together only if it is the most feasible match. Otherwise, these riders can be matched with different drivers as well.

(ii) Cheikh and Hammadi [28] assumed that joint rider can be used for riders, who want to travel together. In their algorithm, this parameter is determined by riders and these riders cannot be separated by the algorithm.

- *Time flexibility.* This parameter is used in many algorithms to represent flexibility of participants in their schedule. In ride-matching algorithms, it is calculated as follows:

$$f = (T_{od}^{LA} - t_{od}) - T_{od}^{ED}, \tag{1.1}$$

where T_{od}^{LA}, t_{od} and T_{od}^{ED} represent latest arrival time, travel time from origin to destination location and earliest departure time, respectively. To use this formula, earliest departure time from origin location and latest arrival time to the destination location are specified by the participants. Travel time is calculated by dividing travel distance to an average speed. In other words, calculation of travel time requires an average speed assumption that may be not feasible because of ever-changing traffic conditions.

- *Allowable waiting time.* Waiting time is one of the crucial parameters that can be used in a ride-matching algorithm. Ghoseiri *et al.* [38] used allowable waiting time parameter, which is specified by participants, instead of calculating time flexibility. Masoud and Hammadi [14] also used allowable waiting time as a constraint in addition to time flexibility.

- *Social parameters.* These parameters can be used in ride-matching algorithms to represent characteristics and choices of the participants. While most of the ride-matching algorithms focus on the system-wide benefits, participants consider their own benefits to overcome security concerns, to enjoy ride-sharing or any other social reasons. For example, a participant may want to be matched with another participant at the same gender or close to his/her age. These social parameters can represent age, gender, pet restrictions, smoking restrictions, employment status or willingness to meet new people, i.e. socialize. Social parameters are very important in real-life; however, these parameters are omitted in most of the ride-matching algorithms. This is mostly because, considering these parameters brings an important computational burden to the algorithms. Ghoseiri *et al.* [38] utilizes social parameters, namely gender, age, pet restrictions and smoking, in their optimization model as constraints, but they did not offer a solution approach to their model.

- *Acceptable walking distance.* This parameter was used for riders, who agree to walk to some predetermined meeting points. This parameter causes increase in participation rate, because this gives a chance to the drivers, who are not willing to change their routes, to be included in a ride-sharing system. Additionally, it is sometimes hard for drivers to find a specific location given by riders, so meeting points can be helpful to overcome this challenge. On the other hand, meeting points are very advantageous for riders, who do not want to reveal their home addresses for security reasons.
- *Acceptable detour distance.* This parameter is similar to acceptable walking distance. In this case, a driver submits this parameter to specify how much they are willing to change their route to pick a rider up.
- *Number of allowable transfers.* This parameter was used in ride-matching algorithms, in which multi-hop ride-matching was allowed. This parameter is specified by riders to represent how many times they are willing to change their transport vehicle. This parameter is included in ride-matching algorithms as a constraint.

Some ride-matching algorithms and parameters used in these algorithms are summarized in Table 1.2. In the algorithms presented in Table 1.2, joint rider parameter did not get enough attention. Time flexibility was widely used, yet, travel times were assumed to be known or calculated by dividing travel distance to an average speed. Nevertheless, assumption of known travel time or average speed would not reflect situations in real-life. Therefore, time flexibility should be considered carefully. Ghoseiri *et al.* [38] used allowable waiting time parameter specified by the participants instead of time flexibility. Masoud and Jayakrishnan [14] used time flexibility and allowable waiting time together as constraints. Social parameters were only used by Ghoseiri *et al.* [38] as constraints, despite the fact that social parameters are seen as very important to achieve critical mass [3]. Concept of meeting points, presented in algorithms of

Ghoseiri *et al.* [38] and Stiglic *et al.* [30], requires additional parameters such as acceptable walking distance and acceptable detour distance. When multi-hop ride-sharing is allowed, number of allowable transfer can be limited as seen in the algorithms of Ghoseiri *et al.* [38] and Masoud and Jayakrishnan [14].

Table 1.2. Parameters used in some ride-matching algorithms in the literature.

Reference	Joint rider	Time flexibility	Allowable waiting time	Social parameters	Acceptable walking distance	Acceptable detour distance	Number of allowable transfers
Agatz *et al.* (2011) [15]	x	Yes	x	x	x	x	x
Ghoseiri *et al.* (2011) [38]	x	x	Yes	(*)	Yes	Yes	Yes
Masoud and Jayakrishnan (2017) [14]	x	Yes	Yes	x	x	(**)	Yes
Cheikh and Hammadi (2016) [28]	Yes	Yes	x	x	x	x	x
Najmi *et al.* (2017) [23]	x	Yes	x	x	x	x	x
Wang *et al.* (2017) [39]	x	Yes	x	x	x	x	x
Stiglic *et al.* (2015) [30]	Yes	Yes	x	x	Yes	Yes	x
(*) Used as constraints.							
(**) Used but not limited.							

1.3.4. Assumptions

Real-life is very complicated to fully represent it in an algorithm. Therefore, many assumptions should be made while constructing an algorithm to be able to solve them in feasible time. Some assumptions made in the investigated ride-matching algorithms are as follows:

- *Rolling horizon approach.* Rolling horizon approach forces the matching algorithm to postpone finalization of the previously found matches until a deadline specified by the users. This technique aims to increase the number of matches by including future ride requests in the matching algorithm.

- *Flexible rider–driver role.* This approach assumes that a driver can accept to be a rider instead of a driver if a tempting request is offered. This is possible in real-life no matter the possibility is low or not. Many algorithms omitted this assumption and assumed fixed rider–driver role, because the flexibility in roles brings significantly large burden to computations.
- *Rider relocation.* Some algorithms assumed that riders walk to some meeting points for ride-sharing. This way, they can meet with a driver on the driver's routes or at least closer to driver's route that is acceptable by the driver. Thus, participation rate can be increased.
- *Driver detour.* Many algorithms assumed that drivers change their route at an acceptable level to meet with riders.
- *Known travel time.* This assumption was made by some researchers to calculate a time flexibility for the participants. As mentioned in the preceding section, this assumption can be considered as weak due to ever-changing traffic conditions.
- *Matching rule.* This is an assumption to describe single rider–single driver, multiple riders–single driver or single rider–multiple drivers matches. Single rider–single driver match is preferred to ease computational burden. In this scenario, a driver is assumed to be matched with single rider. Some researchers achieved to match a driver with multiple riders. In this case, capacity of the driver, i.e. number of empty seats in the vehicle, should be more than the number of drivers. In a single rider–multiple drivers match, namely multi-hop match, riders are assumed to transfer between vehicles to reach their destinations.
- *Multi-modal.* This assumption refers to the case, where ride-sharing is combined with public transportation. In other words, riders use ride-sharing option to reach a point, where they can transfer to public transportation. This may lead to increase in matching rate. There is still a need for studies that combine ride-sharing and public transportation.

Some assumptions made in the literature is summarized in Table 1.3. In Table 1.3, it is seen that rolling horizon approach, which forces participants to wait until a deadline specified by them, was widely used; however, the participants would not want to wait a long time to be matched even if they give a deadline [29]. Ghoesiri *et al.* [38] did not use rolling horizon approach but they did not offer any solution approach to their model either. Cheikh and Hammadi [28] and Stiglic *et al.* [30] did not use rolling horizon approach either but they also did not offer an alternative approach. Flexible rider-driver role was proposed by Agatz *et al.* [15] but in their study computation times were ignored. Rider relocation was used in the studies, in which meeting points were assigned. In all investigated algorithms drivers detour to meet riders at riders' origin locations or meeting locations. Most of the studies assumed travel times are known or can be calculated with an average speed, whereas, Ghoseri *et al.* [38] used travel distance and waiting times specified by participants. Most of the algorithms allowed multiple riders–single driver match, while some of them also allowed multi-hop matching for riders. On the other hand, none of these algorithms constructed a multi-modal matching system.

Table 1.3. Assumptions made in the literature.

Reference	Rolling horizon	Flexible rider-driver role	Rider relocation	Driver detour	Known travel time	Matching rule	Multi-modal
Agatz *et al.* (2011) [15]	Yes	Yes	x	Yes	Yes	SS	x
Ghoseiri *et al.* (2011) [38]	x	x	Yes	Yes	x	MS, SM	x
Masoud and Jayakrishnan (2017) [14]	Yes	x	Yes	Yes	Yes	MS, SM	x
Cheikh and Hammadi (2016) [28]	x	x	x	Yes	Yes	MS, SM	x
Najmi *et al.* (2017) [23]	Yes	x	x	Yes	Yes	SS	x
Wang *et al.* (2017) [39]	Yes	x	x	Yes	Yes	MS	x
Stiglic *et al.* (2015) [30]	x	x	Yes	Yes	Yes	MS	x
SS: Single rider-single driver, MS: Multiple riders-single driver, SM: Single rider-Multiple drivers.							

1.3.5. Summary and Discussion on Ride-Matching Algorithms

Ride-matching algorithms, which match drivers, who owns a vehicle with empty seats, with riders, who look for a vehicle to ride, are at the center of ride-sharing systems. In this section, some ride-matching algorithms in the literature were investigated. Objective functions and parameters used in these algorithms were examined, as well as, their solution approaches and assumptions made to construct these algorithms. The aim was to present challenges and opportunities to construct a successful ride-matching algorithm.

It was concluded that there are some broad areas for researches:

- Objective functions, which should be defined to reflect real-life concerns.
- Parameters, which consider variables faced in the real-world while maintaining feasible computation times.
- Assumptions, which offer solvable ride-matching algorithms while considering facts, which are important for participants.

Objective functions are the main components of a ride-matching algorithms that help decide which rider will be matched with a driver or vice versa. Past attempts to make ride-sharing popular among people have failed to achieve critical mass [17]. Therefore, objective functions should be determined such that more people would be willing to involve in a ride-sharing system. Many of the investigated ride-matching algorithms focused on system-wide benefits such as maximizing total travel distance savings or number of matched participants. There is an algorithm among the investigated studies, which used objective function of minimizing total waiting and delay time for the convenience of the participants but this objective alone may not be sufficient to attract enough people into a ride-sharing system. It is concluded that new objective functions, which consider primarily benefits of users, are needed. There is no point of maximizing system benefits if there are not enough participants in a ride-sharing system to be consistent.

Parameters used in ride-matching algorithms play a key role for algorithms to be successful. More parameters will lead to increase in computation times, but eliminating more than enough parameters will cause algorithms to be deficient. It is concluded that there is a need to determine the importance of parameters used in ride-matching algorithms. For example, joint rider parameter, which represents riders, who are willing to travel together, presents a real-life request because many riders may want to travel with their friends for social reasons and security reasons. Including this parameter in a ride-matching algorithm may increase participation rate; however, this parameter may also bring computational burden. Therefore, the advantages and disadvantages of such parameters should be investigated and supported with numerical studies. Furthermore, social parameters were not included in most of the investigated studies. Social parameters are used in one study among them [38], but they are only used in the constraints. In this study, participants' choices are asked and matches are finalized based on their answers. For example, if a rider asks to be matched with a driver at the same gender, the algorithm eliminates the drivers who do not satisfy this condition. This can be seen as a positive outcome, but this also causes decrease in matching ratio. As a result, alternative methods can be developed such that an algorithm can consider social parameters and when there are no better options, the algorithm can offer matches to the participants even if these constraints are not satisfied. There is also a need for studies, which assess the effects of including social parameters on performance of such algorithms.

Each ride-matching algorithm should make some assumptions to solve the matching problem at feasibly short notice. Rolling horizon approach was proposed to increase matching rate by making finalization of the matching until a deadline to include new-arriving requests in the system. This approach causes increase in waiting times for the participants, which may cause decrease in satisfaction ratio. Alternative approaches can be studied in future studies.

As a conclusion, there are still many remained challenges for the ride-matching algorithms that will provide great research opportunity. A successful ride-matching algorithm may play a key role to achieve critical mass in ride-sharing. Especially, demands of participants, who are the main reasons to construct ride-sharing systems, should be investigated carefully. Advanced ride-matching algorithms should be developed to satisfy the needs of the participants.

1.4. Motivation behind Utilizing Needleman-Wunsch Algorithm

To be able to match drivers with riders, drivers are required to go to riders' origin location or suitable meeting points. For this reason, either drivers should change their routes to pick up riders or riders should wait at a point that is located on drivers' routes. When drivers change their route to pick up riders, only the single rider–single driver match option is available unless dynamic routing for drivers is utilized. Dynamic routing brings significant computational burden that may result in infeasible computation times. Consequently, to allow multiple riders–single driver match, drivers' routes should be fixed to maintain feasible computation times. This can be achieved by asking drivers which routes they choose before the beginning of their travels or assigning the shortest path to their destination for them.

In this dissertation, due to lack of data regarding routes of potential drivers, drivers are assumed to choose the routes assigned for them, which are the shortest path to the destination locations. Riders are assumed to agree on going to the meeting locations, which are located on drivers' routes and available for vehicles to pick up riders. Meeting points are represented by letters. Thus, routes of drivers are represented by letter arrays. If the origin and destination points of a rider are located on a driver's route, then route feasibility for the rider and the driver are satisfied. To create an automated ride-matching, sequence analysis should be done for aligning routes and meeting locations represented by letter arrays.

Recent decades, bioinformatics community has addressed longest common subsequence (LCS) problem. They use sequence analysis to find and score the similarities between a sample amino acid chain and amino acid chains of known proteins [40]. The LCS problem is studied under alignment algorithms created by Needleman-Wunsch [41]. They presented the first systematic tool to consider the insertion and deletion of letters from a letter array that naturally occurs in biological sequences [42]. Needleman-Wunsch algorithm is one of first examples of dynamic programming and still widely used. It is an exact matching algorithm, so it is widely preferred especially when the quality of alignment is of the utmost importance [2].

When sequence alignment algorithms are considered, performance and scalability become more critical as input sizes increase. Therefore, computational complexity of sequence alignment algorithms should be investigated in terms of time and space. This can be measured by examining computation time and memory consumption, which are referred as time complexity and space complexity, respectively [1]. To determine the efficiency of an algorithm, an upper bound on the asymptotic growth rate of the algorithm, "O", is used. Let X[1. . . n] and Y[1. . . m] be two letter arrays with the length of m and n, respectively. If it is known that a sequence alignment algorithm has an upper bound of mxn in its worst case, it is referred as O(mxn). O(mxn) means that as input size increases, the worst case running time of the algorithm will increase with a rate proportional to mxn. Time and space complexity of some sequence alignment algorithms and their main characteristics are given in Table 1.4. In this table, m and n are lengths of sequences and m is bigger than n.

Table 1.4. Some sequence matching algorithms and their characteristics [1, 2].

Algorithm name	Preprocess time complexity	Search time complexity	Space Complexity	Characteristics
Needleman Wunsch	-	O(mn)	O(mn)	Global alignment. An exact matching algorithm. One of the first and basic examples of dynamic programming.
Smith and Waterman	-	O(mn)	O(mn)	Local sequence alignment. It is used only to find best matched part in a sequence. It is developed from Needleman Wunsch algorithm.
Affine gap penalty	-	2 x O(mn)	2 x O(mn)	Increase in accuracy resulted in loss of efficiency. Two matrices stored in the memory, so space complexity is doubled compared to Needleman-Wunsch algorithm.
Hirschberg		O(mn)	O(m)	Splits the longer sequence into two, then calculates for each half. Then final rows are used to find optimal crossing-point.
Boyer-Moore	O(m+n)	O(mn)	O(m+sigma)	It uses good suffix shift and bad character shift. It doesn't check all characters.

In this dissertation, sequence alignment algorithm is used to check if the origin and destination locations of a rider are covered by route of a driver. Therefore, length of sequence for a rider is always two and length of sequence for a driver's route equals to or bigger than two. Some characteristics of route alignment are as follows: (a) A letter occurs only once in a route sequence, because a driver would visit the same location only once, (b) to verify a driver's route covers both origin and destination locations of a rider, all letters should be compared, so an exact matching algorithm should be utilized, (c) Trace-back process is not needed, because to check origin and destination are covered by a route, only the score of the algorithm is needed, (d) gaps and mismatchings should not be penalized. Based on these characteristics of route alignment used in this dissertation, algorithms that utilizes heuristics, such as Boyer-Moore, are not applicable for route alignment problem. Among the rest of the algorithms presented in Table 1.4, affine gap penalty is less efficient than Needleman-Wunsch, Smith and Waterman and Hirschberg by means of time and space complexity. The remaining three algorithms have the same time complexity. Since, trace-back process is not required for route alignment, these three algorithms can be considered. In the literature, a simulation study was conducted to show time and space complexity of these algorithms [1]. Figure 1.3 depicts the time and space complexities of these algorithms under different size of inputs.Figure 1.3 shows that the Hirschberg resulted in good performance of space complexity, whereas, the Needleman-Wunsch algorithm resulted in the best time complexity. As a result, the Needleman-Wunsch algorithm is selected for route alignment for the following reasons:

- It is one of the most used and basic algorithms in bioinformatics over decades.
- It is an exact sequence alignment algorithm that is a required characteristic for route alignment.
- It resulted in the best time performance compared to its competitors, namely Smith and Waterman and Hirschberg, which satisfy other conditions for route alignment.

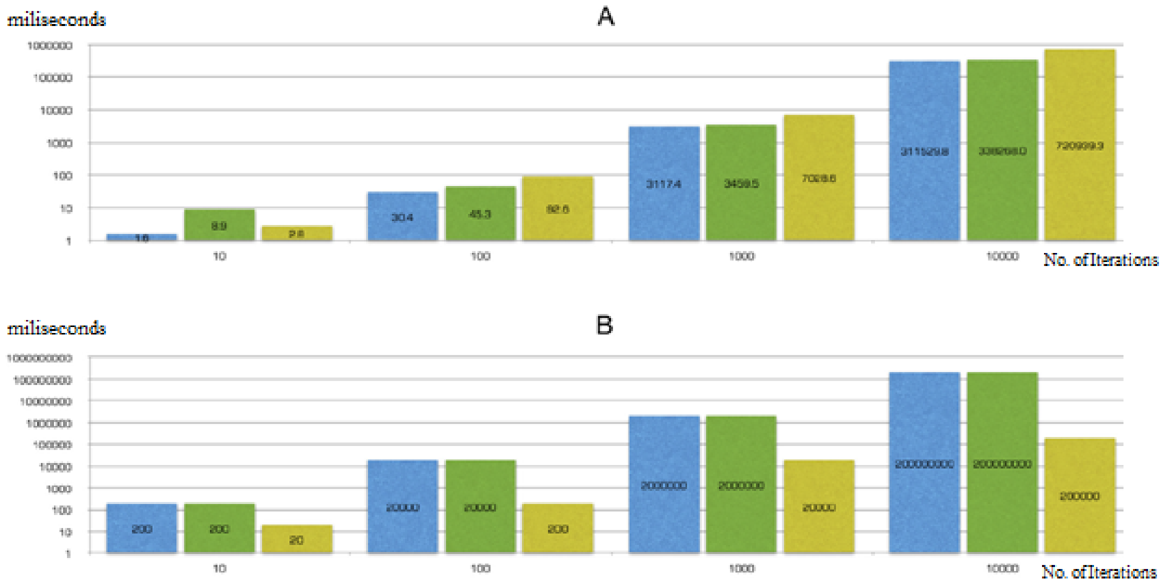

Figure 1.3. Time and space complexities of the Needleman-Wunsch (blue), Smith and Waterman (green) and Hirschberg (yellow). (A) Time complexities in miliseconds, (B) space complexities in bytes [1].

1.5. Contributions

In this dissertation, a novel ride-matching algorithm is proposed to overcome the aforementioned challenges. In other words, a ride-matching algorithm is developed that optimizes matches between drivers and riders by considering their characteristics and choices at a reasonably short notice. The main contributions of this dissertation to the literature can be summarized as follows:

- In the proposed algorithm, to identify similarities among the travel patterns of users, the routes of the drivers are assumed to be fixed. Suitable riders are identified using a sequence alignment algorithm, namely the Needleman-Wunsch algorithm. The Needleman-Wunsch algorithm is widely used in the bioinformatics field to identify similarities between a sample amino acid chain with amino acid chains recorded in a database [43]. The basic function of the Needleman-Wunsch algorithm is to align arrays of letters and rate their similarity [41]. In recent years, this algorithm has been used in social and geographical studies. In such studies, travel or activity patterns are presented as arrays of letters comparable to amino acid chains [44, 45]. In the proposed algorithm, the routes of the users are presented as arrays of letters, and the similarity between these arrays is scored to find feasible matches.

- The characteristics and choices of users, such as gender, age, employment and tendency to meet new people, are included in the objective function of the proposed algorithm. Similar parameters have been presented in the literature as constraints [38]. In the proposed algorithm, the similarities between these parameters are scored by multiplying their weights assigned by participants. Using this approach, a rider can be matched with a driver even should some of the passengers' choices are not completely satisfied, as long as the match is still acceptable.

1.6. Organization of This Dissertation

This dissertation is structured as follows. In Chapter 2, defines the problem and introduces the ride-sharing model. In Chapter 3, the solution approach for this ride-sharing model is outlined, and the application of the Needleman-Wunsch algorithm in this study is also described. Chapter 4 presents the evaluation of the survey outcomes. In Chapter 5, a simulation study is performed using the data acquired from the survey. Furthermore, simulation studies using a traditional weighted bipartite matching algorithm and an optimization algorithm that includes social factors are conducted, and performance of the proposed algorithm is compared with that of these algorithms. In Chapter 6, details of the sensitivity analysis performed for the proposed algorithm are described and results of the analysis are evaluated. Chapter 7 concludes the dissertation by summarizing the results of this study.

2. PROBLEM DEFINITION

The main objective in attempting to solve the ride-matching problem is to find the most feasible matches between riders and drivers. Figure 2.1 depicts an example of a ride-sharing schema. The letters, namely A, B, C, D, E, F and G, represent the locations. In this example, driver $d1$ has origin and destination locations "A" and "E", respectively. Rider $r1$ wishes to travel from "C" to "E", rider $r2$ from "B" to "D", rider $r3$ from "C" to "E" and rider $r4$ from "F" to "E." The driver may choose to be matched with some of the riders based on their route and characteristics.

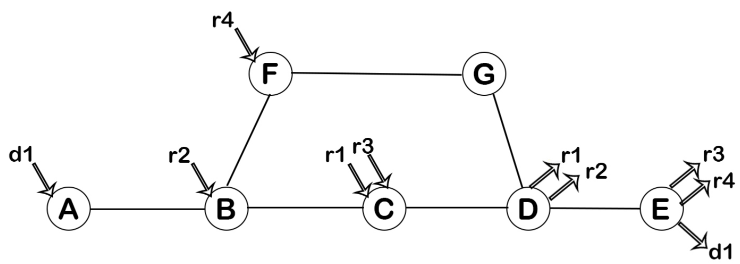

Figure 2.1. Ride-sharing schema for a driver and several riders.

The ride-sharing system contains a set of participants P. These participants are divided into two groups: a set of drivers D and a set of riders R. Each rider and driver make a trip announcement which defines their travel requests. A set of trip announcements S is defined such that $R \subset S$ and $D \subset S$. Each trip announcement $s \subset S$ is associated with origin and destination locations o_s and d_s.

It is assumed that drivers do not change their prespecified routes. Thus, routes are assigned for the drivers based on their origin and destination locations. To check time feasibility, it is assumed that riders will wait past their latest departure time as long as they know a driver is coming for them. Therefore, the earliest departure time T^{ED} and the latest departure time T^{LD} are assigned for each announcement. The latest arrival time and travel time, which are used in traditional optimization algorithms, are ignored, because time suitability is checked with announcement and allowable waiting times.

In this algorithm, each rider $r \in R$ specifies the rider count k_r, that is, the number of riders willing to travel together as a rider group. For example, a single rider's rider count value is one, whereas two friends, who are willing to travel together in the same vehicle, have a rider count value of two. Each driver $d \in D$ specifies his or her capacity, that is, the number of empty seats c_d. A novel aspect of this algorithm is the objective function, which maximizes participants' benefits by considering their characteristics and choices. Shaheen *et al.* [46] suggest that gender, age and employment status are key drivers of ride-sharing. The proposed algorithm uses the following four parameters and their respective weights to define the benefits of the participants: gender g_s; age a_s; employment status w_s; socialness or willingness to meet new people, σ_s; gender weight γ_g; age weight γ_a; employment status weight γ_w; and socialness weight γ_σ. In previous studies, the trip preferences of both drivers and riders, including age, gender, smoking preference and pet restrictions, were incorporated as constraints; however, these are not used in the objective functions [23, 38, 47].

2.1. Feasible Match

A match between a rider and a driver can be considered feasible if their routes and schedules are similar. These similarities are defined as spatial and temporal constraints, respectively. Additional constraints, such as distance savings that are prioritized in traditional weighted bipartite matches can be defined for feasible matches, as long as spatial and temporal constraints are satisfied.

The proposed algorithm assumes that a driver can pick up riders at meeting points located on the driver's route. In the example given in Figure 2.1, driver $d1$ plans to travel from point "A" to "E", and rider $r1$ wishes to travel from "C" to "D". The best route for the driver is "ABCDE." The match between $d1$ and $r1$ is defined as spatially feasible, as the route of driver $d1$ contains both the origin and the destination of rider $r1$.

To verify time feasibility, drivers and riders specify their latest departure times. It is assumed that when a rider specifies a latest notification time of 15 minutes and a driver responds to this call within this time range, the match can be defined as temporally feasible. This is true even if a rider has to wait more than 15 minutes, as riders can wait more than 15 minutes should they know that a driver is coming to pick them up.

2.2. Matching Algorithm

To match riders with the most feasible drivers, arcs are created between each rider and spatially and temporally feasible drivers. The illustrative graph in Figure 2.2 represents the sample case presented in Figure 2.1. In Figure 2.2, the numbers on the edges denote the joint socialness score (JSS). The JSSs are calculated using social parameters and parameter weights. It is assumed that driver $d1$ is taking the "ABCDE" route, which includes the origins and destinations of riders $r1$, $r2$ and $r3$. Therefore, arcs are only created for these pairs and not for $r4$, whose origin and destination are not on the driver's route.

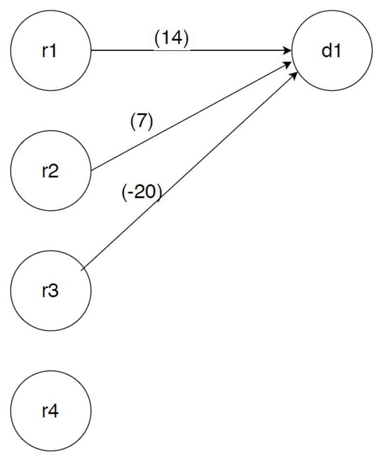

Figure 2.2. An illustrative graph with a driver and four riders.

The objective of the proposed algorithm is to maximize the benefits of both the riders and the drivers. The benefits are measured with JSSs.

The objective function for each rider r becomes

$$maximize \quad \gamma^{rd} = x_g^{rd}\gamma_g^r\gamma_g^d + x_a^{rd}\gamma_a^r\gamma_a^d + x_w^{rd}\gamma_w^r\gamma_w^d + x_\sigma^{rd}\gamma_\sigma^r\gamma_\sigma^d. \tag{2.1}$$

In Equation 2.1, weights of the social factors of the rider r and his or her feasible driver d, specifically gender weight γ_g, age weight γ_a and employment status weight γ_w, are multiplied to calculate the JSS, γ_{rd}. The variable x is positive one if the social characteristics are the same and negative one if they are different. The objective function is 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. A sample calculation of the JSS is presented in Table 2.1, which presents the characteristics of driver $d1$ and rider $r1$. It is assumed that all users want to be matched with a user with similar characteristics. The weights are obtained from the users, who are asked to rate the weights of each social factor from zero to five. A rating of zero indicates that it is not important to be matched with a user with the same social characteristic; a rating of five indicates that being matched with a similar user is very important.

Table 2.1. An illustrative example of the computation of the JSS.

	Driver $d1$		Rider $r1$			
	Characteristics	Weight	Characteristics	Weight	x_{rd}	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 the example given in Table 2.1, driver $d1$ is a male driver with an age of between 18–25 who works at the TAU. Driver $d1$ states 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 states that he is willing to meet new people with an weight factor of five. In contrast, rider $r1$ is a female with an age of between 25–40 who also works at TAU. Her weight factor for willingness to meet new people is three. As mentioned previously, the variable x_{rd} is assigned a value of positive one if characteristics are the same and negative one otherwise. In this situation, the value of x_{rd} is negative one for gender and age because the driver and the rider's gender and age range are different. The value of x_{rd} is positive one for employment and socialness because they are working at the same location and they are both willing to meet with new people. The score for gender becomes $1x5x(-1) = (-5)$. When the scores of the other social characteristics are calculated in this way, the JSS can be calculated by simply adding all of these scores.

In order to find the feasible matches for a rider, a capacity constraint should be checked. For the proposed algorithm, the number of empty seats available in the driver's vehicle, c_d, should be greater than the rider count, k_r. For example, if riders $r1$ and $r2$ are a married couple and want to travel together, only one of them request a ride. The rider count of this couple is two, and they cannot be matched with a driver with only one empty seat. To satisfy this constraint, the following equation is included in the proposed algorithm:

$$c_d \geq k_r. \tag{2.2}$$

3. SOLUTION APPROACH

One of the most significant barriers in ride-matching problems is dealing with large number of participants within a feasible time period [3]. In this section, the approach adopted to solving the defined ride-matching problem is discussed.

3.1. Time Feasibility

In the algorithm proposed in this dissertation, the earliest and latest departure times specified by the participants are used to analyze time feasibility. Therefore, traditional time constraints are not used in the matching algorithm. Instead, a status factor is defined for each participant to determine whether the request for a ride that he or she makes is active or passive. As mentioned previously, it is assumed that even if a driver arrives at the meeting location after a rider's latest departure time, the rider will still wait for the driver if they are matched between the specified earliest departure time and the latest departure time.

Many traditional optimization algorithms [15,39] calculate time flexibility f using travel times t_{od}, earliest departure time T^{ED} and latest arrival times T^{LA} as shown in Equation 1.1, where, the value of t_{od} is calculated using the average speed of the vehicles.

Instead of travel and latest arrival times, the earliest and latest departure times are used in the proposed algorithm. This approach is adopted because travel times vary greatly in metropolitan cities, especially during peak hours, and these time calculations impose a significant computational burden on the computer software. Therefore, an algorithm (Figure 3.1) was created to check the announcement status based on the given earliest and latest departure times of a participant. This algorithm is run every minute and updates the values of announcement activeness.

T^{ED}=*earliest departure time specified by the participant*

T^{LD}=*latest departure time specified by the participant*

if $T^{ED} < current\ time\ \&\ current\ time < T^{LD}$ **then**

Participant Status = "*active*"

else

Participant Status = "*passive*"

end if

Figure 3.1. Announcement status updating algorithm.

3.2. Route Feasibility and the Needleman-Wunsch Algorithm

In the proposed algorithm, drivers' routes are assumed to be prespecified and fixed because they are not willing to change their prespecified routes [48]. Therefore, to satisfy the route feasibility constraint, a rider's origin and destination locations should be on the driver's route. The route of each driver n_d is determined based on the driver's origin and destination locations, and a set of meeting locations M are defined.

In the example given in Figure 2.1, it is clear that driver $d1$, who wants to travel from point "A" to point "E," will use the route "ABCDE." Driver $d1$ can be matched with riders $r1$, $r2$ and $r3$ because their origins and destinations are located on the driver's route. However, an algorithm is required to find the similarities between the routes of the driver and the riders. To analyze route similarity the Needleman-Wunsch algorithm, one of the first examples of dynamic programming, is used.

The Needleman-Wunsch algorithm scores the alignment of two groups of letters. A matrix (M[i, j]) is created, and scores of matching, mismatching and gap are assigned. These scores are assigned to the cells such that if the letters are same, matching score is assigned; if they are different, mismatching score is assigned; if one of the letters is missing, gap score is assigned. The missing letter in a letter array is defined as indel value. The algorithm has various solving methods, but all of them give the same result.

The steps involved in solving the problem are as follows [49]:

(i) A matrix, S, is defined, where i and j denoting the row and column numbers. Let m and n denote the lengths of the first and second letter arrays, then

$$0 \leq i \leq m \text{ and } 0 \leq j \leq n.$$

(ii) The values of S are set to one if there is a match and to zero if there is no match (assuming the matching score is one and the mismatching score is zero). If there is a gap, that is to say an indel value in the letter groups, a gap score is assigned. When the gap score is zero,

$$S[i,0] = 0 \text{ for } i = 1,2,...,m \text{ and } S[0,j] = 0 \text{ for } j = 1,2,...,n.$$

(iii) Compute scores starting from the top-left cell using following equation:

$$M[i,j] = S[i,j] + max(M[i-1:x], M[j-1:y]). \tag{3.1}$$

(iv) Start the traceback process from the bottom-right cell and continue by selecting the cell with the lowest value from the adjacent columns and rows.

In example given in Figure 2.1, to check the route feasibility between driver $d1$ and rider $r1$, the letter arrays “ABCDE” and “CE” should be aligned. The sequence alignment for this pair will result in a score of two, and an alignment is found such that

$$\begin{array}{ccc} AB & CD & E \\ -- & C- & E. \end{array}$$

For the algorithm proposed in this dissertation, only the score of the matrix, not the alignment of the letters, is needed. Therefore, the Needleman-Wunsch algorithm is modified by eliminating the traceback process. In this algorithm, calculation of the matrix begins at the top-left cell and finishes at the bottom-right cell. The S and

M matrices are calculated as depicted in Figures 3.2 and 3.3. In Figure 3.2, the S matrix is created as follows: If the letters are the same, then a matching score of one is written; otherwise, a mismatching score of zero is written. In Figure 3.3, the M matrix is created using equation 3.1. Since S[1,1] = 0, M[0,1] = 0 and M[1,0] = 0, M[1,1] is calculated as 0. The bottom-right cell M[2,5] is calculated as 2 because s[2,5] = 1, M[1,5] = 1 and M[2,4] = 1, thus max(M[1,5], M[2,4]) = 1 and M[2,5] = S[2,5] + 1 = 2.

		A	B	C	D	E
		0	0	0	0	0
C	0	0	0	1	0	0
E	0	0	0	0	0	1

Figure 3.2. The Needleman-Wunsch algorithm after the generation of the S matrix.

		A	B	C	D	E
		0	0	0	0	0
C	0	0	0	1	1	1
E	0	0	0	1	1	2

Figure 3.3. The Needleman-Wunsch algorithm after the generation of the M matrix.

When using the Needleman-Wunsch algorithm for route checking, if the letters representing the origin and the destination of the rider are along the route of the driver, the score (the value of the cell mxn) equals two. Thus, it is concluded that when the Needleman-Wunsch algorithm is used to compare the letter arrays representing the route of driver d and the origin and destination of rider r and the matching score is one, the mismatching score and gap penalty are zero, driver d and rider r are said to be spatially feasible, if the score is two. The proposed algorithm for checking route feasibility is given in Figure 3.4.

m = number of letters in rider's route

n = number of letters in driver's route

$matchscore = 1$

$mismatchscore, gappenalty = 0$

for `i` *in* `m+1` **do**

  for `j` *in* `n+1` **do**

    $match = score[i-1][j-1] + matchscore(seq1[i-1], seq2[j-1])$

    $delete = score[i-1][j] + gappenalty$

    $insert = score[i][j-1] + gappenalty$

    $score[i][j] = max(match, delete, insert)$

  end for

end for

return $score[m][n]$

Figure 3.4. Needleman-Wunsch algorithm to check route feasibility.

3.3. Joint Socialness Score (JSS)

The JSS is used to score the similarity of two participants' characteristics. The JSS of driver d and rider r is calculated using Equation 2.1.

In this equation, γ_g, γ_a, γ_w and γ_σ represent gender weight, age weight, employment status weight and socialness weight, respectively. The value x is a variable defined such that $x \in \{1, -1\}$. Its value is positive one if the characteristics are the same and negative one if the characteristics are different. x can be defined as follows:

$$x = 1, \qquad \text{if characteristics are similar}$$
$$x = -1, \qquad \text{otherwise.}$$

In order to solve for the values of x variables, the similarities of the characteristics of participants are checked. In the example given in Table 2.1, characteristics of driver $d1$ are male, aged 18–25, attending TAU and positive tendency to socialness. Similarly, characteristics of rider $r1$ are female, aged 25–40, attending TAU and positive tendency to socialness. Then, characteristics of the driver and the rider are compared. For example, x for genders of the driver and the rider is assigned as -1, because their genders are different. An example calculation of JSS is demonstrated in Table 2.1.

3.4. Matching Process

In this section, the matching process is outlined. The matching process is carefully constructed to ease computational burden it imposes on the systems used. The "first come, first served (FCFS)" method is applied. When a rider enters the system, the capacity constraint for all available drivers is first checked. Next the JSSs for all feasible drivers are calculated. The rider is matched with the driver whose corresponding JSS is the highest. The proposed algorithm follows these steps:

(i) If there is a new announcement, update the database.
(ii) Select the unmatched rider whose announcement time is the earliest.
(iii) Select the temporally feasible driver with the earliest announcement time.
(iv) Check whether the rider's origin and destination locations are on the driver's route.
(v) If driver's route is feasible for the rider, calculate the JSS between the driver and the rider and add this pair to the matchable pair list.
(vi) If there is an unchecked driver, go to step three and repeat the process.

(vii) Select the driver with the best JSS from the matchable pair list and match him or her with the rider.

(viii) Eliminate the rider from the system and subtract the rider count from the capacity of the matched driver.

(ix) Update the database and repeat the process, starting from step one.

Note that if a rider is matched with a driver even for small part of the driver's route, the capacity of the driver's vehicle is reduced for the entire route. However, the route can be divided into sections using the Needleman-Wunsch algorithm. This may increase the number of matches but also increases computation times. For example, when a driver following the route "ABCDE" picks up a rider with origin "C" and destination "E," the capacity of the driver is decreased for the entire route. In reality, the driver can pick up another rider whose origin and destination locations are on the route "ABC." In the algorithm proposed in this dissertation, the option of separating the route into sections is omitted to reduce the length of computation times. The matching algorithm is described in Algorithm 3.5. The flowchart of the algorithm is presented in Figure 3.6.

m =number of riders

n =number of drivers

for `i` *in* `range(0,m)` **do**

  if rider count $>$ 0 **then**

    for `j` *in* `range(0,n)` **do**

      if driver capacity $>$ rider count **then**

        Route feasibility is checked as in Figure 3.4

        if route is feasible **then**

          $calculate\, socialness\, score$ {See Eqn.2.1}

        end if

      end if

    end for

  end if

  $Match\, the\, rider\, with\, the\, best\, driver$

  $Eliminate\, the\, rider\, from\, the\, system$

  $capacity\, of\, the\, driver = capacity\, of\, the\, driver - rider\, count$

end for

Figure 3.5. The proposed matching process.

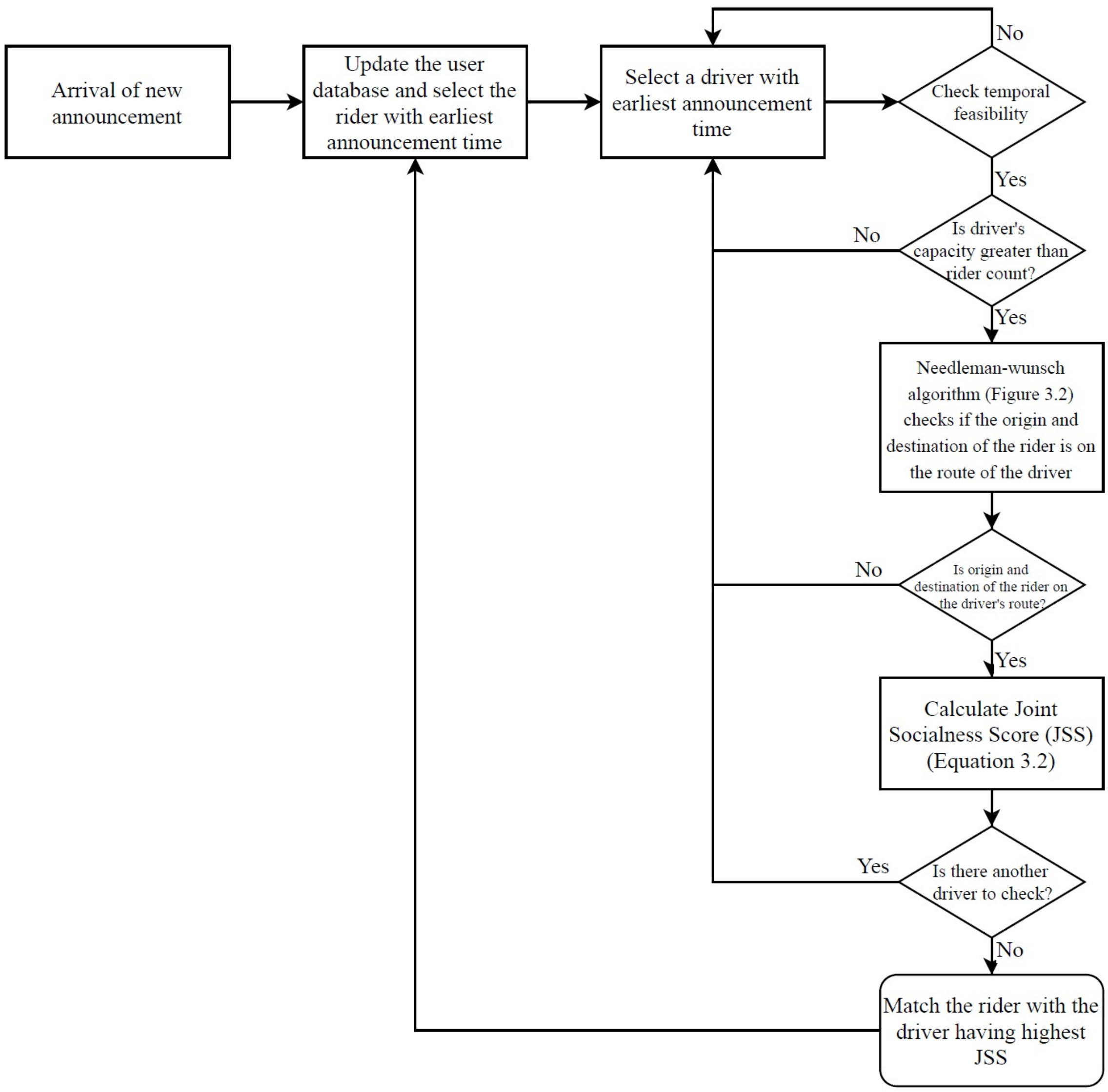

Figure 3.6. Flowchart of the proposed matching process.

4. DATA COLLECTION AND ANALYSIS

4.1. Turkish-German University

The survey was conducted at the TAU, which is located in the Beykoz region, 10 km away from the nearest center of Istanbul, namely Kavacik. Istanbul has a very wide public transportation web, yet TAU has few transportation options because it is located in a district that is relatively sparsely inhabited. As depicted in Figure 4.1, there are only four bus stations located at a walkable distance from the campus entrances. Bus station one is located at Gate B, while bus stations two, three and four are approximately a six-minute walk away. Furthermore, the frequency of bus arrivals at these stations is quite low. A student may wait for approximately 20–30 minutes for a bus during the day. All of the buses take long routes to campus. This makes the travel duration from the city center (Kavacik) to campus at least 30 minutes, whereas the trip takes 10 minutes by private vehicle. As a result, many students prefer to hitchhike at the main enterance of the campus shown in Figure 4.1.

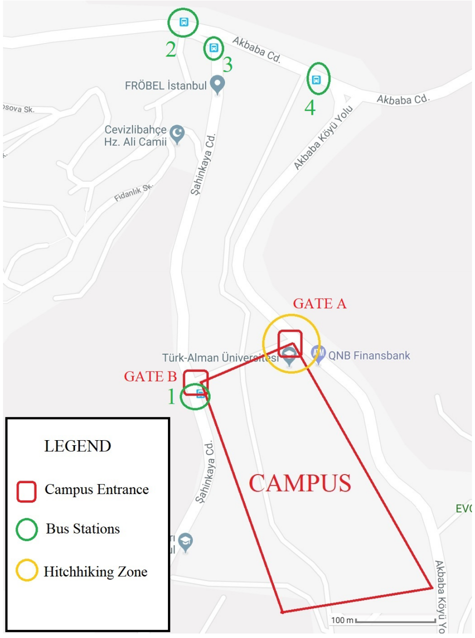

Figure 4.1. Location of the TAU Campus in Beykoz, Istanbul.

4.2. Survey Data

In 2018, there were nearly 2,000 students and staff at the TAU. A stated preference survey with 604 participants was performed at the university. The survey was conducted with faculty members and other staff members (e.g. janitors, tea makers, etc.), in addition to students, in order to understand the behaviors of all potential participants. The survey investigated participants' travel behavior and thoughts on ride-sharing systems. The survey investigated a variety of socioeconomic characteristics, such as gender, age, occupation, education and car ownership. Participants were also questioned on their travel characteristics, including frequency of using private cars and public transportation, trip time and cost, most preferred transportation mode, trip mode alternative and tendency to hitchhike. Technological characteristics, such as tendency to use mobile applications for transport and attitude toward new alternative transportation modes (such as ride-sharing), were also investigated. The survey questions are presented in Appendix A. The description of the survey is presented in Table 4.1.

Table 4.1. Descriptive results of the survey.

	Students	Faculty Staff	Administrative Staff	Other Staff	Total
Total no. of participants	497	32	2	17	548
No. of participants who own a private or family car	44	16	0	9	479
Proportion of participants who own a private or family car to total	8.85%	50%	0%	52.94%	87.41%
No. of participants who do not own a private or family car	453	16	2	8	69
Proportion of participants who do not own a private or family car	91.15%	50%	100%	47.06%	12.59%
Average travel time from home to the campus	58.23	37.72	17.5	19.76	55.69

In Table 4.1, it can be seen that average travel times of students and staff greatly differ. This is because most of the university staff live near the university, while most of the students live with their family in homes located far from the university.

The questions posed to the participants to gain insight into their ride-sharing preferences are presented in Table 4.2. The participants were asked to assign each of these questions a value ranging from zero to five to represent the weights of sharing a ride with a similar participant. These weight factors are utilized in the algorithm to find joint weight factors. The results of the answers to the questions given in Table 4.2 are given for the students, faculty staff, administrative staff and other staff in Table 4.3, 4.4, 4.5 and 4.6, respectively and summarized in Table 4.7. In these tables, drivers are participants who own a private or family car; the rest of the participants are classified as riders.

Table 4.2. Social factors and related survey questions.

Parameter	Relevant Question
Gender	Would you agree sharing a ride with people of the same gender?
Age	Would you agree sharing a ride with people of a similar age?
Employment	Would you agree sharing a ride with people from the same university?
Socialness	Would you agree sharing a ride with strangers?

Table 4.3. The average weights of the social parameters for the students.

	Students		
	driver	rider	total
Importance of gender (0-5)	1.61	2.70	2.61
Importance of age (0-5)	2.59	2.59	2.59
Importance of employment (0-5)	2.61	1.86	1.92
Importance of socialness (0-5)	1.84	1.10	1.16

Table 4.4. The average weights of the social parameters for the faculty staff.

	Faculty Staff		
	driver	rider	total
Importance of gender (0-5)	2.50	2.12	2.31
Importance of age (0-5)	1.75	1.69	1.72
Importance of employment (0-5)	2.19	1.69	1.94
Importance of socialness (0-5)	1.37	0.25	0.81

Table 4.5. The average weights of the social parameters for the administrative staff.

	Administrative Staff		
	driver	rider	total
Importance of gender (0-5)	0	0	0
Importance of age (0-5)	0	0	0
Importance of employment (0-5)	0	0	0
Importance of socialness (0-5)	0	0	0

Table 4.6. The average weights of the social parameters for the other staff.

	Other Staff		
	driver	rider	total
Importance of gender (0-5)	4.78	3.25	4.06
Importance of age (0-5)	0.78	0.50	0.65
Importance of employment (0-5)	0.89	0.62	0.76
Importance of socialness (0-5)	0	1.50	0.71

Table 4.7. The average weights of the social parameters for all participants.

	Total		
	driver	rider	total
Importance of gender (0-5)	2.23	2.68	2.63
Importance of age (0-5)	2.16	2.51	2.47
Importance of employment (0-5)	2.29	1.82	1.88
Importance of socialness (0-5)	1.49	1.07	1.13

Table 4.3, 4.4, 4.5, 4.6 and 4.7 indicate that gender, age and employment play key roles for ride sharing as Shaheen *et al.* suggested in their study [46]. The socialness factors in a ride-matching algorithm can also be used to encourage participation in a ride-sharing system. The average results indicate that the weights of age, gender, employment and socialness are 2.47, 2.63, 1.88 and 1.13 out of 5, respectively. However, these values differ significantly among different groups. For example, the weight of gender for ride-sharing is 4.06 among other staff, which means that gender is very important to them when sharing a ride. In contrast, gender is not very important to the students, who assigned this factor an average rating of 2.61 points out of 5. Additionally, students may be more interested than university staff in using a ride-sharing system to meet new people.

The residential locations of the participants in the survey were used to identify suitable meeting locations. The home addresses of the participants were not asked to maintain their privacy; instead, participants were asked to name the bus stations closest to their home. It is assumed that drivers will use highways because of the traffic problems in Istanbul. The cost of traffic congestion in Istanbul was calculated to be \$ 3.12 billion in 2005 and congestion has continued growing [50, 51]. To define meeting locations suitable for both drivers and riders, the route origin locations are assigned to the nearest meeting locations, which are the bus stations located on the highways. A map of Istanbul and the defined meeting points is presented in Figure 4.2.

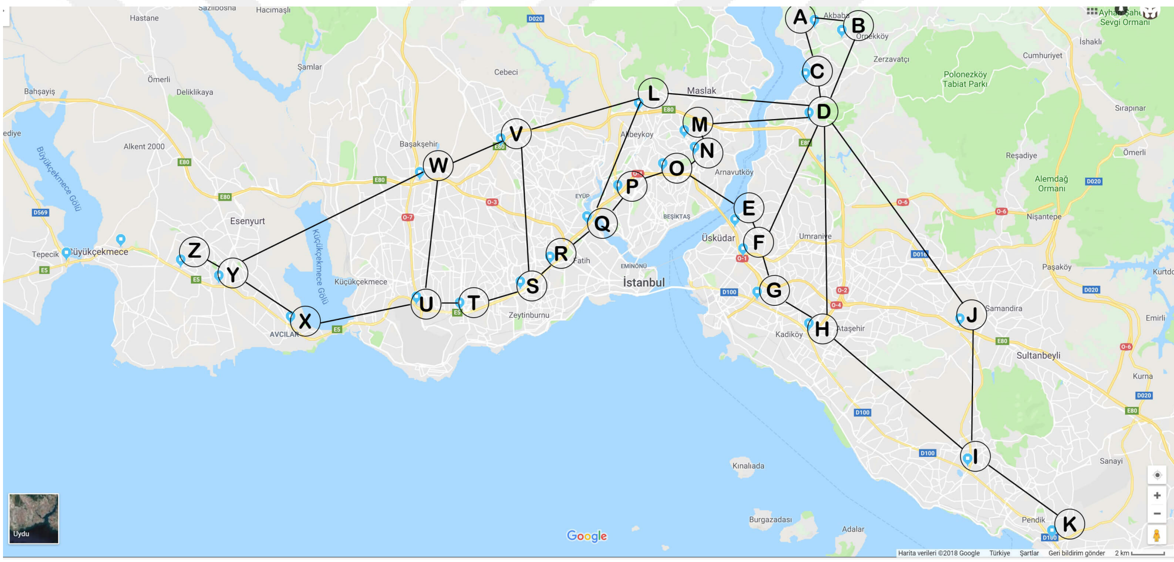

Figure 4.2. Map of Istanbul and meeting points for ride-Sharing.

5. SIMULATION STUDY

In this chapter, the proposed ride-matching algorithm is tested using data set obtained from the survey performed at Turkish-German University and the results are discussed. Performance and matching qualities of the proposed algorithm are compared with other algorithms, namely the traditional bipartite matching algorithm and an optimization algorithm.

5.1. Application of Other Matching Algorithms

In this section, the two other ride-matching algorithm, which are compared with the proposed matching algorithm, namely the traditional bipartite matching algorithm and an optimization algorithm, which includes social parameters, are presented and their applications are described.

5.1.1. Traditional Weighted Bipartite Matching Algorithm

The traditional bipartite matching algorithm was constructed with the objective function of maximizing system-wide distance savings. This algorithm allows single rider-single driver matches and ignores the choices of participants. 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:

$$\sigma_{(i,j,k,l)} = d_{o_i d_i} - (d_{o_i k} + d_{kl} + d_{l d_i}) + \sum_{j \in R} (d_{o_j d_j} - (d_{o_j k} + d_{l d_j})). \tag{5.1}$$

In Equation 5.1, 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. In order 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. The approach used to match drivers and passengers in the weighted bipartite algorithm is presented in Figure 5.1. Flowchart of this algorithm is shown in Figure 5.2.

```
m = number of riders
n = number of drivers
for j in range(0,n) do
   if driver capacity > 0 then
      for i in range(0,m) do
         Calculate distance savings as in Eqn. 5.1
         if distance savings ≥ 0 then
            add to feasible matches
         end if
      end for
   end if
end for
for k in range(0,feasible_matches_count) do
   Select the match with max distance saving
   Eliminate the matched rider and driver from the list
end for
```

Figure 5.1. Weighted bipartite matching algorithm.

Figure 5.2. Flowchart of the weighted bipartite ride-matching algorithm.

5.1.2. The Optimization Algorithm with Social Parameters

An optimization algorithm that includes social parameters was also constructed. This algorithm's objective function is to maximize distance savings. This algorithm allows multiple riders-single driver matches. Distance savings are calculated using Equation 5.1. The route of the driver is assumed to be fixed.

Unlike the proposed algorithm, which uses social parameters in an objective function, this optimization algorithm uses social parameters as constraints [38]. As a result, in the approach used to solve the optimization problem, a driver is matched with a rider only if all of their choices and characteristics match. In contrast, the algorithm proposed in this dissertation can match a female rider with a male driver if there is no better option. As mentioned previously, participants are asked to assign values ranging from zero to five to the parameters. To simulate this algorithm using the survey outcomes, the choices and characteristics of the participants can be transformed accordingly. When a male participant assigns zero, one or two points to "same gender choice," he can travel with a male or female driver. Similarly, if he assigns values of three, four or five, he will be assigned to the same gender.

The optimization algorithm checks the compatibility of social parameters not only between a rider and a driver but also between a rider and other riders. When multiple riders are allowed, the match among riders is also checked. Two versions of this algorithm are constructed, one for single rider-single driver matches (Optimization A) and one for multiple riders-single driver matches (Optimization B). The matching algorithm developed based on these considerations, is depicted in Figure 5.3. Since the only difference between Optimization A and Optimization B is the number of riders to be matched with a driver, the same algorithm is used for both of them.

```
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 5.3. Optimization with social factors algorithm.

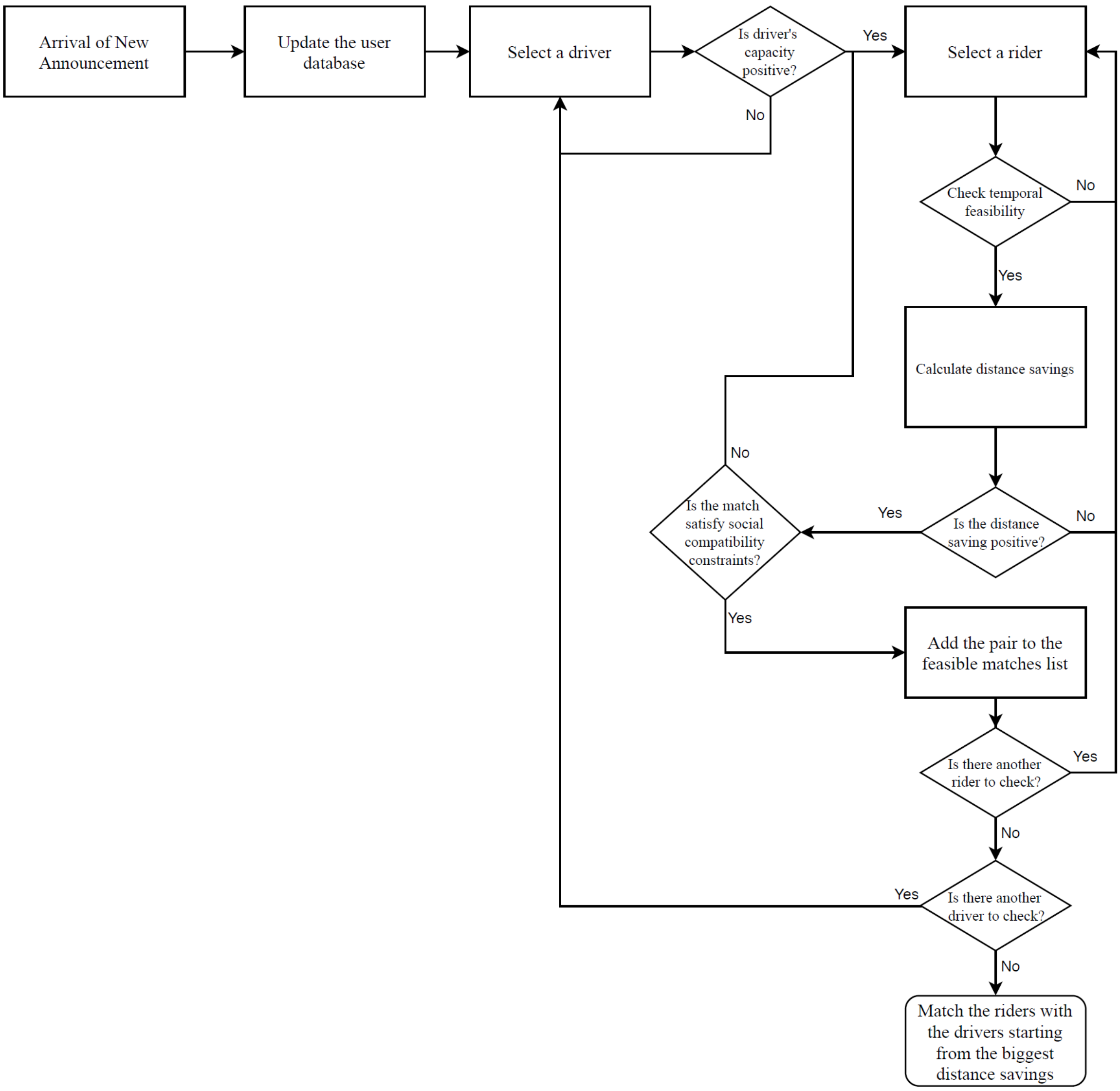

Figure 5.4. Flowchart of the optimization ride-matching algorithm.

5.2. Performance of the Ride-matching Algorithms

The proposed algorithm was modeled in Python 2.7, and its performance was measured on a computer with an i5 2.7 GHz processor and 8 GB of RAM. All of the ride-matching algorithms described in this dissertation were modeled in Python 2.7 using the same data and the same computer.

To conduct a computational study, the survey data were used such that each participant was assumed to be included in the ride-sharing system. Travel demands were created based on survey answers such that:

- The origins of the participants were assumed to be the meeting locations closest to their homes, and the destination was assumed to be the TAU.
- In order to test the algorithms with the highest possible number of participants, two assumptions were made:
  - (i) All trip announcements were known prior to the beginning of the day.
  - (ii) All of the participants wished to arrive to the university at the same time.
  - (iii) All riders can wait for a driver as long as it takes if they know there is a vehicle coming to pick them up.
- Since there are very few students who own a car, all participants with a private car or a family car were counted as drivers.
- Participants' choices and characteristics were determined using questions from the survey presented in Table 4.2.
  - (i) The ages of the participants were categorized into four age groups: under 18, 18–24, 25–40 and over 40. These groups present children, students, young faculty and senior faculty, respectively.
  - (ii) Employment status was categorized as student, faculty staff, administrative staff or other staff.
  - (iii) All participants were assumed to be willing to meet new people, but the weight factor differed from person to person.

To fairly compare the performance of the proposed algorithm with the traditional weighted bipartite matching algorithm and the optimization algorithm, the algorithms were tested using different scenarios. Performance results are given in Table 5.1. In this table, Algorithms A, B, C and D represent the proposed algorithms given in Algorithm 3.5. In Algorithm A, three rider-single driver matches were allowed, and social parameters were included. In Algorithm B, single rider-single driver matches were allowed, and social parameters were included. In Algorithms C and D, social parameters were excluded. Algorithm C allowed three rider-single driver matches while Algorithm D allowed single rider-single driver matches. The traditional weighted bipartite matching algorithm, which is presented in Algorithm 5.1, allows only single rider-single driver matches. This is because the algorithm assumes that a driver will change his or her route to pick up a rider. Social parameters are excluded in this algorithm. Since the route of the driver is fixed in the optimization algorithm, multiple riders-single driver matches are allowed. Two versions of the optimization algorithm that included social parameters, which are given in Figure 5.3, were tested. Optimization A represented the optimization model that included social parameters and allowed single rider-single driver matches. Optimization B used the same algorithm and allowed three rider-single driver matches. Each algorithm was run 10 times to find average values. The computation times presented in the table are the averages of these iterations.

Table 5.1. Performance of the ride-matching algorithms.

	Alg. A	Alg. B	Alg. C	Alg. D	Bipartite	Optimization A	Optimization B
Socialness factor	Included	Included	Excluded	Excluded	Excluded	Included	Included
Driver Capacity	3	1	3	1	1	1	3
Rider Count	1	1	1	1	1	1	1
Number of matches	191	67	191	67	67	61	80
Average total computation time (sec)	17.65	11.63	9.36	8.07	101.64	171.52	214.09
Average computation time per match (sec)	0.09	0.17	0.05	0.12	1.52	2.81	2.68

Table 5.1 indicates that the computation time of the traditional weighted bipartite algorithm was 12.6 times higher than the equivalent version of the proposed algorithm (Algorithm D). However, the number of matches did not change. This is likely because there were many riders to be matched. When social characteristics and choices were included in the single rider-single driver match, the proposed algorithm (Algorithm B) performed 14.7 times faster than the optimization algorithm (Optimization A). When a driver could be matched with three riders and social concerns were considered, the computation time of the proposed algorithm (Algorithm A) was found to be 12.1 times faster than that of the optimization algorithm (Optimization B). The number of matches found by the optimization algorithm was considerably lower than that of the proposed algorithm. This is because no lower bound was set for social parameter scoring in the proposed algorithm, so no match could be defined as unfeasible due to differences in social parameters. Overall, the proposed algorithm found a match for a rider within one second, even when including social characteristics and choices in its calculations. These results suggest that the proposed algorithm presents not only more qualitative matches but also feasible computation times for real-life applications.

5.3. Quality of Matches

The importance of presenting choices to ride-sharing participants has been discussed in the literature. This is seen as one of the key factors in achieving critical mass in ride-sharing [3,46]. Therefore, the matches found by the algorithms presented in the preceding section (Section 5.2) were analyzed to measure the impacts of the choices. The quality of the matches was measured by finding the average JSSs of the matches found by the algorithms. The results are summarized in Table 5.2.

When social factors were excluded for single rider-single driver matches, the proposed algorithm (Algorithm D) yielded higher JSS values compared to the weighted bipartite algorithm (Table 5.2). Note that when the proposed algorithm included social parameters (Algorithms A and B), it yielded greater JSSs compared to those algorithms that did not include social parameters (Algorithms C and D and the bipartite algorithm). When social parameters are included, the optimization algorithms

Table 5.2. Quality of the matches.

	Alg. A	Alg. B	Alg. C	Alg. D	Bipartite	Optimization A	Optimization B
Total number of riders	479	479	479	479	479	479	479
Number of matched riders	191	67	191	67	67	61	80
Average JSS value of the matched pairs	8.25	7.87	1.07	4.48	3.57	11.42	16.35
Number of matched pairs having a JSS greater than zero	154	51	127	48	46	52	70

(Optimization A and Optimization B) had greater JSS values compared to the proposed algorithms (Algorithms A and B). This is because the optimization algorithms used social parameters as constraints and accepted matches only if all of the social parameters of the participants were compatible. The proposed algorithm will match incompatible matches if there is no better option.

When a lower bound of zero was set for JSS values, 48 out of 67 possible matches were found using Algorithm D. In Algorithm D, social parameters were not included. The number of matches found by the weighted bipartite algorithm was 46 out of 67. When social parameters were included, this value increased to 51 for the proposed algorithm (Algorithm B). The number of matches found by the optimization algorithm for single rider-single driver matches (Optimization A) was 52 out of 67. The minor difference between Algorithm B and Optimization A was caused by the FCFS approach used by the proposed algorithm. To clarify, the proposed algorithm matches a rider with the earliest announcement time with a driver, even if some of their social parameters are incompatible, whereas the optimization algorithm skips matching an incompatible rider and instead finds another rider who is compatible with the driver. When the capacities of the vehicles were set to three and social factors were considered, the proposed algorithm (Algorithm A) found 154 matches, while the optimization algorithm (Optimization B) found 70 matches. Thus, it is concluded that if a lower bound is set, Algorithm A finds 2.2 times more matches when compared to Optimization B (when social factors are concerned and multiple riders-single driver matches are allowed).

6. SENSITIVITY ANALYSIS

In this chapter, influences of the model parameters on the results of the proposed ride-matching algorithm were identified. To understand which inputs affect the output variability of the proposed algorithm, regression analysis was used. To conduct regression analysis, 615 iterations with different combinations of numbers of riders, numbers of drivers, capacities of drivers, rider counts, numbers of riders who reject first match and JSS limits were analyzed. The results, namely ratio of matched riders, computation times and average JSSs of the matches, were measured on a computer with an i5 2.7 GHz processor and 8 GB of RAM. The computational results are shown in Appendix B.

6.1. Regression Analysis

Regression Analysis is widely used statistical method to understand the relationship between a dependent variable with one or more independent variables [52]. Using regression analysis, an equation, which defines the relationship between a dependent and one or more independent variables, are created, and coefficients of the independent variables and constant value are calculated [53, 54]. When there are more than one independent variables, multiple regression analysis is utilized. Using multiple regression, contributions of the independent variables on the variability of dependent variable are calculated; however, in some cases the contribution of some independent variables may not be significantly important [55].

There are two types of regression analysis based on the number of independent variables: Simple linear regression and multiple linear regression. In simple linear regression, the relationship between a dependent variable Y and an independent variable X is shown in Equation 6.1.

$$Y = \beta_0 + \beta_1 X + \varepsilon \tag{6.1}$$

In Equation 6.1, β_0, β_1 and define regression coefficients, and ε defines an error value that shows the difference between calculated and observed values.

When there are more that two independent variables, the relationship between a dependent variable and n independent variables $X_1, X_2, ..., X_n$ are shown in Equation 6.2.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \varepsilon \tag{6.2}$$

In Equation 6.2, $X_1, X_2, ..., X_n$ represent n independent variables, $\beta_1, \beta_2, ..., \beta_n$ represent regression coefficients and ε represents the error value.

6.2. Regression Analysis of Parameters

Multiple linear regression analysis were made for dependent variables, namely ratio of matched riders, computation time and average JSSs of the matches. There are six independent variables as follows:

- Number of riders.
- Number of drivers.
- Capacity of each driver's vehicle.
- Rider count.
- Number of riders rejecting first match.
- JSS limit.

Number of riders and number of drivers show the available participants in a given ride-sharing system. Capacity represents the number of available seats in each available driver's vehicle. Rider count means that how many riders want to travel together in the same vehicle. One ride-sharing request are made by riders, even if rider count is specified as more than one. Riders, who would reject the match assigned by a ride-matching algorithm, are also considered. The parameter "number of riders rejecting first match" is used for riders, who reject their first match assigned by an algorithm but accept the second match, if there are any. In the proposed algorithm, JSS of the matches were not limited. To measure the effect of such limit, the parameter "JSS limit" is used. This parameter sets a limit for JSS of the matches, and reject the matches giving lower JSS than the JSS limit and force algorithm to search for other matches that satisfy this constraint.

615 numerical calculations were performed with different combinations of the independent variables. The algorithm was calculated five times for each combination due to stochastic behavior of the problem. Regression analysis were made using a commercially available software SPSS Version 25. The computation times were measured on a computer with an i5 2.7 GHz processor and 8 GB of RAM.

6.2.1. Analysis for Ratio of Matched Riders

In this section, the effects of numbers of riders, numbers of drivers the ratio of matched riders, capacities of drivers, rider counts, numbers of riders who reject first match and JSS limits on the ratio of the matched riders are analyzed using multiple regression analysis and the results are evaluated. Summary output of multiple linear regression model is shown in Table 6.1.

The R^2 value, which is the proportion of variance in the dependent variable that can be explained by the independent variables. In Table 6.1, the R^2 value of 0.913 means that the independent variables, namely numbers of riders, numbers of drivers, capacities of drivers, rider counts, numbers of riders who reject first match and JSS limits, explain 91.3% of the variability of the dependent variable, ratio of matched

Table 6.1. Model summary of the regression for ratio of matched riders.

Model Summary$_b$				
Model	**R**	**R Square**	**Adjusted R Square**	**Std. Error of the Estimate**
1	0.956$_a$	0.913	0.913	7.82703%
a Predictors: (Constant), JSS Limit, Number of riders rejecting first match, Rider count, Number of riders, Capacity of each driver's vehicle, Number of drivers				
b Dependent Variable: Ratio of matched riders				

riders. In other words, high R^2 value indicates a good level of prediction.

Table 6.2. ANOVA of the regression for ratio of matched riders.

ANOVA $_a$					
Model	**Sum of Squares**	**df**	**Mean Square**	**F**	**Sig.**
Regression	393161.630	6	65526.938	1069.612	0.000$_b$
Residual	37247.510	608	61.262		
Total	430409.139	614			
a Dependent Variable: Ratio of matched riders					
b Predictors: (Constant), JSS Limit, Number of riders accepting first match, Rider count, Number of riders, Capacity of each driver's vehicle, Number of drivers					

The analysis of variance (ANOVA) results are shown in Table 6.2. F value defines the ratio of two mean squares. When the F value is large and significance is lower than 0.01, then it is concluded that the independent variables statistically significant to predict the dependent variable. In Table 6.2, F value and significance states that the regression model is good fit of the data.

In Table 6.3, unstandardized coefficients indicate how much the dependent variable varies with an independent variable when other independent variables are held

Table 6.3. Coefficients of the regression for ratio of matched riders.

Coefficients $_a$					
	Unstandardized Coefficients		**Standardized Coefficients**		
Model	**B**	**Std. Error**	**Beta**	**t**	**Sig.**
(Constant)	55.563	2.437		22.804	0.000
Number of riders	-0.139	0.003	-0.758	-52.140	0.000
Number of drivers	0.280	0.017	0.240	16.155	0.000
Capacity of each driver's vehicle	11.033	0.480	0.335	23.005	0.000
Rider count	-22.347	0.914	-0.345	-24.443	0.000
Number of riders rejecting first match	0.015	0.004	0.045	3.320	0.001
JSS Limit	-0.157	0.010	-0.227	-16.199	0.000
a Dependent Variable: Ratio of matched riders					

constant. The general form of the equation to predict ratio of matched riders is shown in Equation 6.2. Based on the coefficients shown in Table 6.3, this equation is rewritten as:

$$Y_{romr} = 55.563 - 0.139X_{nor} + 0.280X_{nod} + 11.033X_{cod} - 22.347X_{rc} + 0.015X_{nrj} - 0.157X_{jss}. \tag{6.3}$$

In Equation 6.3, $Y_{romr}, X_{nor}, X_{nod}, X_{cod}, X_{rc}, X_{nrj}, X_{jss}$ represent ratio of matched riders, number of riders, number of drivers, capacity of drivers, rider count, number of riders rejecting first match and JSS limit, respectively. The statistical significance of these independent variables are also presented in Table 6.3 under the column "Sig.". Based on regression analysis, the following results can be made:

- Number of riders are statistically significant to explain ratio of matched riders. Figure 6.1 and Figure 6.2 show that when number of riders increases, ratio of matched riders decreases. In Figure 6.1, high driver capacity causes increase in ratio of matched riders. When there are 50 riders and driver capacities are set to 3, ratio of matched riders approaches to 100%. In Figure 6.2, increase in rider count led to decrease in ratio of matched riders. These results can be explained with limited number of drivers.
- Number of drivers are statistically significant to explain ratio of matched riders. Figure 6.3 and Figure 6.4 indicate that high number of drivers results in increase in ratio of matched riders. Figure 6.3 shows that capacity of drivers are also found to be directly proportional with ratio of matched riders. Figure 6.4 shows that increase in rider count causes decrease in ratio of matched riders. This can be explained with limited number of drivers.
- Figure 6.5 indicates that ratio of matched riders slightly increases with increasing number of riders rejecting first match. Table 6.3 shows that this change is statistically significant. Since ratio of number of riders to drivers is large, riders may be matched with drivers, even if they reject their first match or different riders may be matched because some riders may not be matched after rejecting their first match.
- Figure 6.6 shows that setting higher JSS limit causes decrease in ratio of matched riders. This is because, JSSs of some matches are smaller than JSS limits.

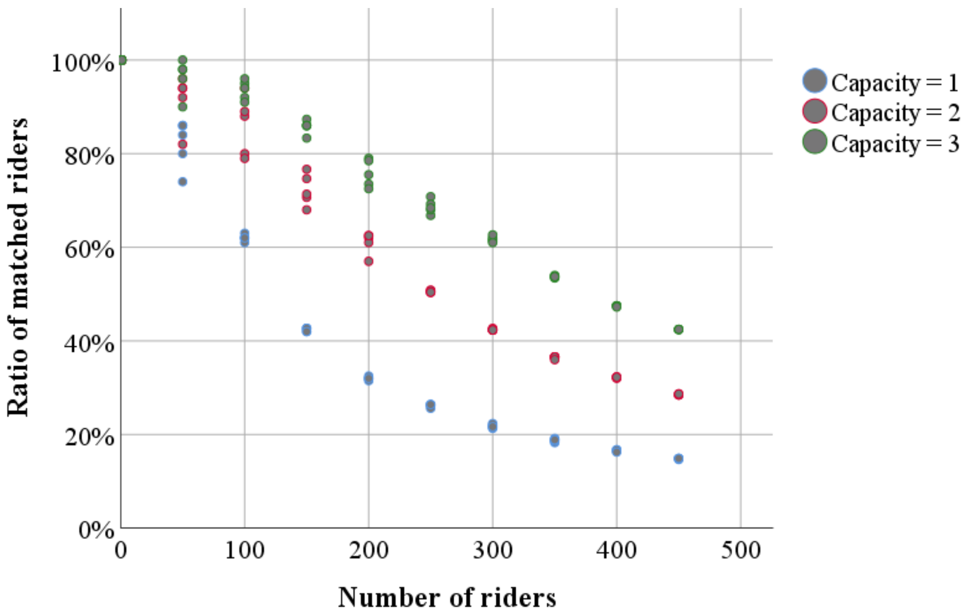

Figure 6.1. Ratio of matched riders versus number of riders by capacities of drivers.

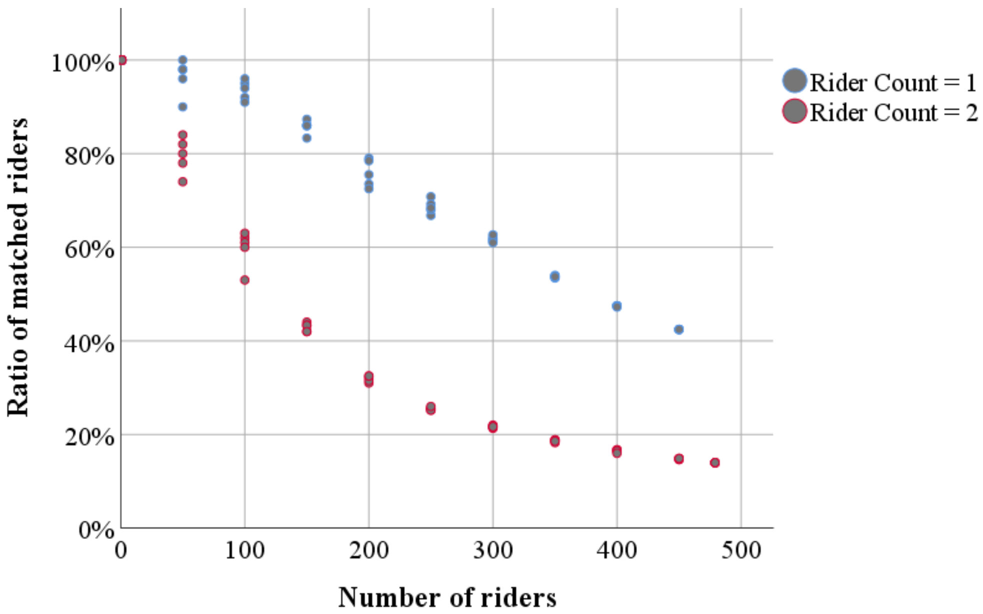

Figure 6.2. Ratio of matched riders versus number of riders by rider count.

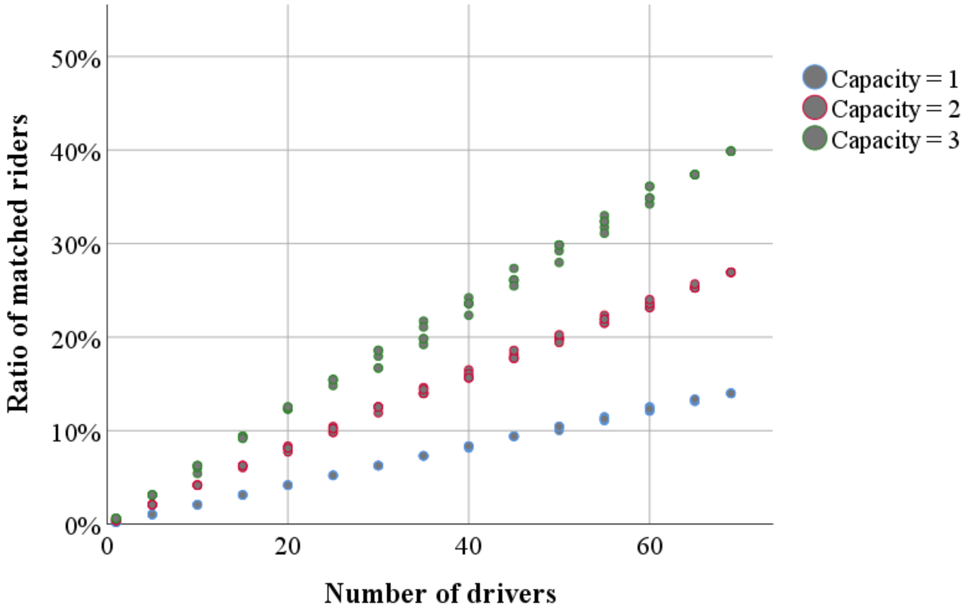

Figure 6.3. Ratio of matched riders versus number of drivers by different capacities.

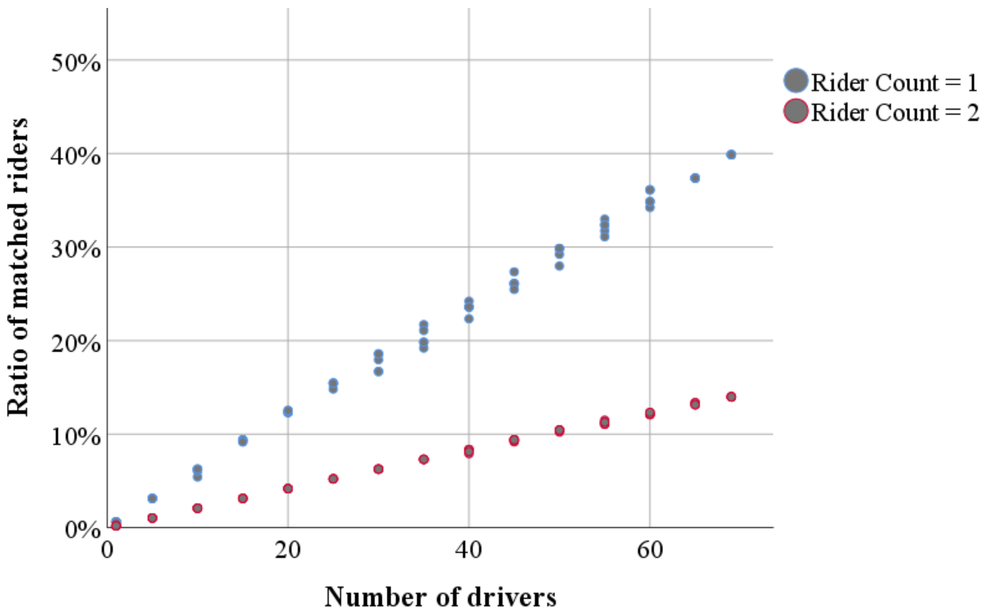

Figure 6.4. Ratio of matched riders versus number of matched drivers by rider count.

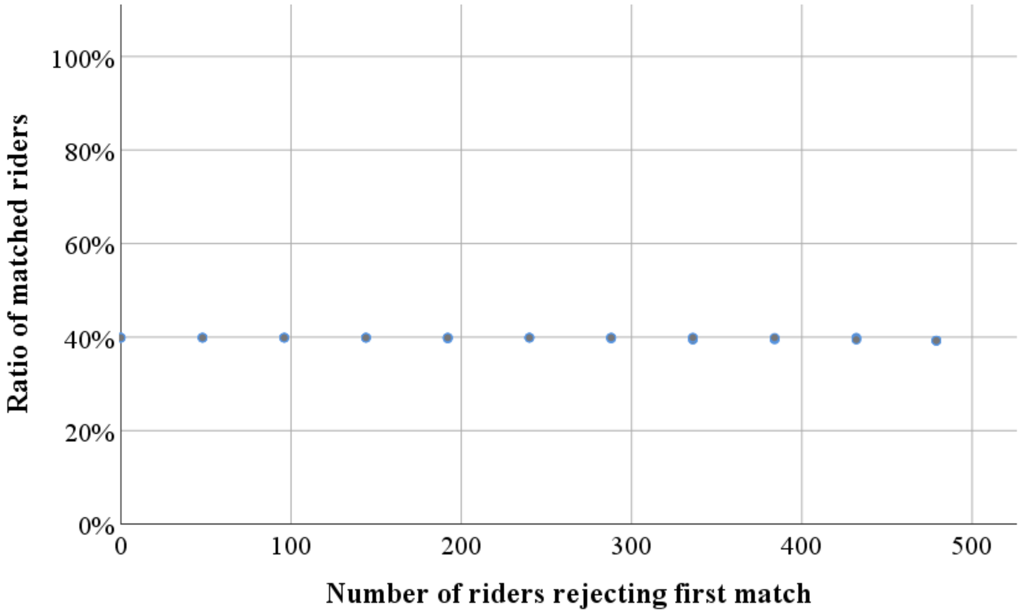

Figure 6.5. Ratio of matched riders versus number of matched riders rejecting first match.

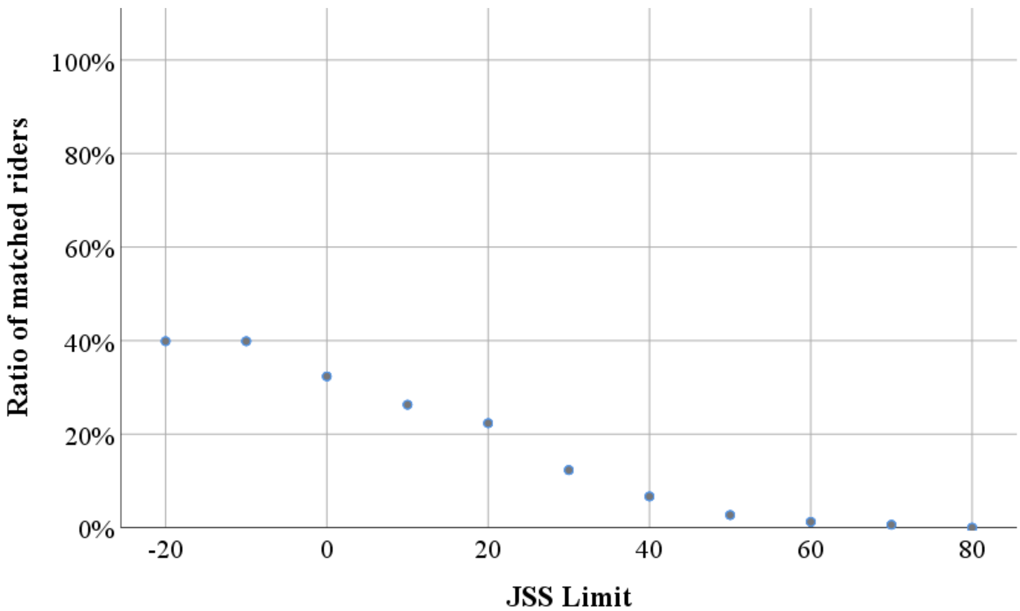

Figure 6.6. Ratio of matched riders versus JSS limit.

6.2.2. Analysis for Computation Time

In this section, using multiple regression analysis, computation times of the proposed algorithm with different combinations of independent variables are examined. Summary output of multiple linear regression model is presented in Table 6.4.

Table 6.4. Model summary of the regression for computation times.

Model Summary $_b$				
Model	**R**	**R Square**	**Adjusted R Square**	**Std. Error of the Estimate**
1	0.980_a	0.961	0.961	1.775814
a Predictors: (Constant), JSS Limit, Number of riders accepting first match, Rider count, Number of riders, Capacity of each driver's vehicle, Number of drivers				
b Dependent Variable: Computation time				

The high R^2 value states that the regression model explains well the relationship between the dependent variable, computation time, and independent variables, numbers of riders, numbers of drivers, capacities of drivers, rider counts, numbers of riders who reject first match and JSS limits.

Table 6.5. ANOVA of the regression for computation times.

ANOVA$_a$					
Model	**Sum of Squares**	**df**	**Mean Square**	**F**	**Sig.**
Regression	47242.229	6	7873.705	2496.803	0.000_b
Residual	1917.337	608	3.154		
Total	49159.565	614			
a Dependent Variable: Computation time					
b Predictors: (Constant), JSS Limit, Number of riders accepting first match, Rider count, Number of riders, Capacity of each driver's vehicle, Number of drivers					

The analysis of variance (ANOVA) results for computation times are shown in Table 6.5. Large F and 0.000 significance values indicate that the independent variables statistically significant to predict the dependent variable. In other words, the regression model for computation times is good fit of the data.

Table 6.6. Coefficients of the regression for computation times.

Coefficients $_a$					
	Unstandardized Coefficients		**Standardized Coefficients**		
Model	**B**	**Std. Error**	**Beta**	**t**	**Sig.**
(Constant)	-2.334	0.553		-4.222	0.000
Number of riders	0.031	0.001	0.494	50.549	0.000
Number of drivers	0.242	0.004	0.613	61.504	0.000
Capacity of each driver's vehicle	1.922	0.109	0.173	17.665	0.000
Rider count	-3.955	0.207	-0.181	-19.068	0.000
Number of riders rejecting first match	0.006	0.001	0.051	5.589	0.000
JSS Limit	0.105	0.002	0.450	47.819	0.000
a Dependent Variable: Computation time					

Based on the results depicted in Table 6.6, the equation for computation time is as follows:

$$Y_{com} = -2.334 + 0.031X_{nor} + 0.242X_{nod} + 1.922X_{cod} - 3.955X_{rc} + 0.006X_{nrj} + 0.105X_{jss}. \tag{6.4}$$

In Equation 6.4, $Y_{com}, X_{nor}, X_{nod}, X_{cod}, X_{rc}, X_{nrj}, X_{jss}$ represent computation time, number of riders, number of drivers, capacity of drivers, rider count, number of riders rejecting first match and JSS limit, respectively. The statistical significance of these independent variables presented in Table 6.6 shows that all independent variables are statistically significant to explain computation time. Based on regression analysis, the following results can be made:

- Number of riders are statistically significant to explain computation times. Figure 6.7 and Figure 6.8 show that when number of riders increases, computation times increase because of increase in number of computation. Similarly, in Figure 6.7, high driver capacity causes more increase in computation times. In Figure 6.8, increase in rider count led to decrease in computation times. This is because, number of computations remains same when rider count is increased; however, since all drivers are matched before computing for all riders, number of computations decreases.
- Number of drivers are statistically significant to explain computation times. Figure 6.9 and Figure 6.10 indicate that high number of drivers results in increase in computation times. Figure 6.9 shows that capacity of drivers are also found to be directly proportional with computation times. These are because the number of computation increases with increasing number of drivers and capacities. Figure 6.10 shows that increase in rider count causes decrease in computation times because smaller number of computations is needed.
- Figure 6.11 indicates that computation times slightly increase with increasing number of riders rejecting first match. This is because, more computations are needed to match riders when the riders reject their first matches, new matches are searched.
- Figure 6.12 shows that setting higher JSS limit causes increase in computation times. More computations are needed to find a new match for a participant, when JSS of the assigned match is lower than the JSS limit.

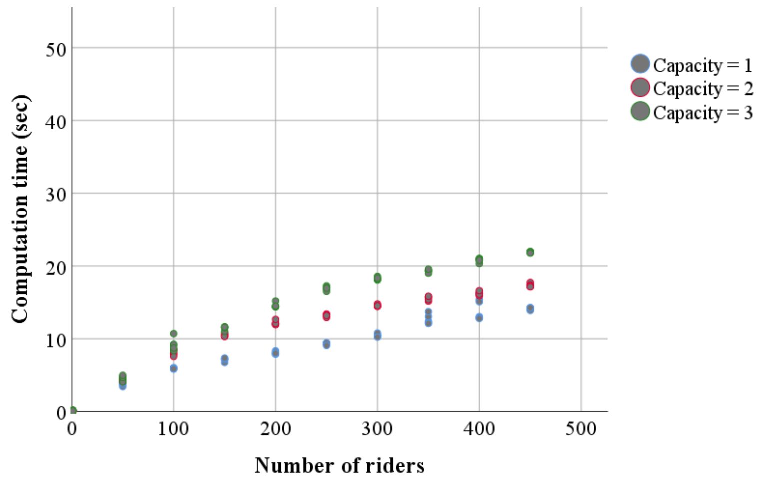

Figure 6.7. Computation time versus number of riders by capacities of drivers.

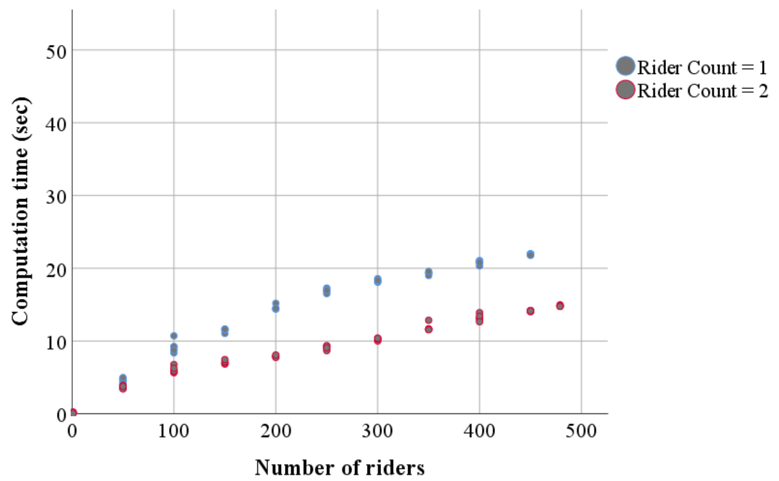

Figure 6.8. Computation time versus number of riders by rider count.

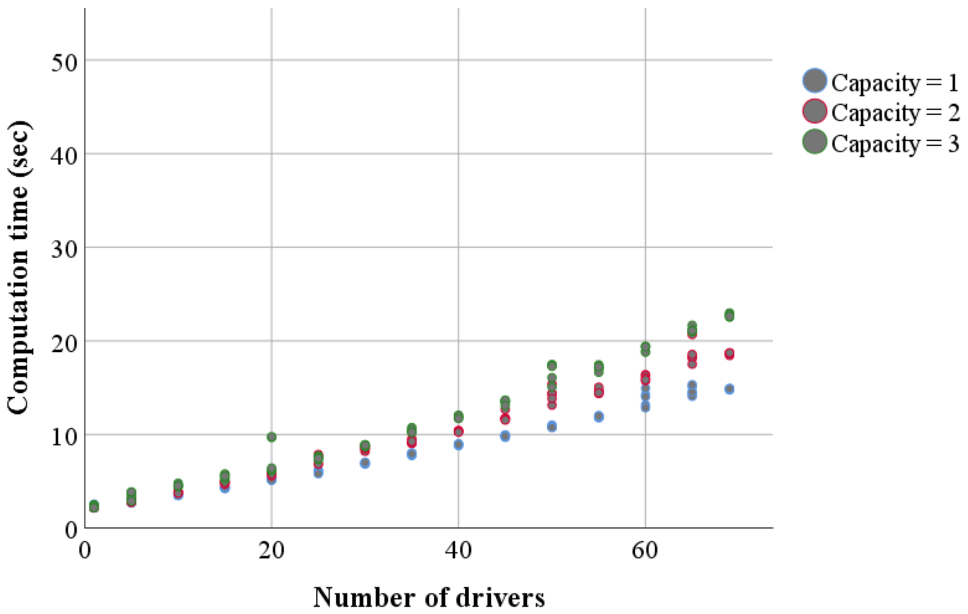

Figure 6.9. Computation time versus number of drivers by different capacities.

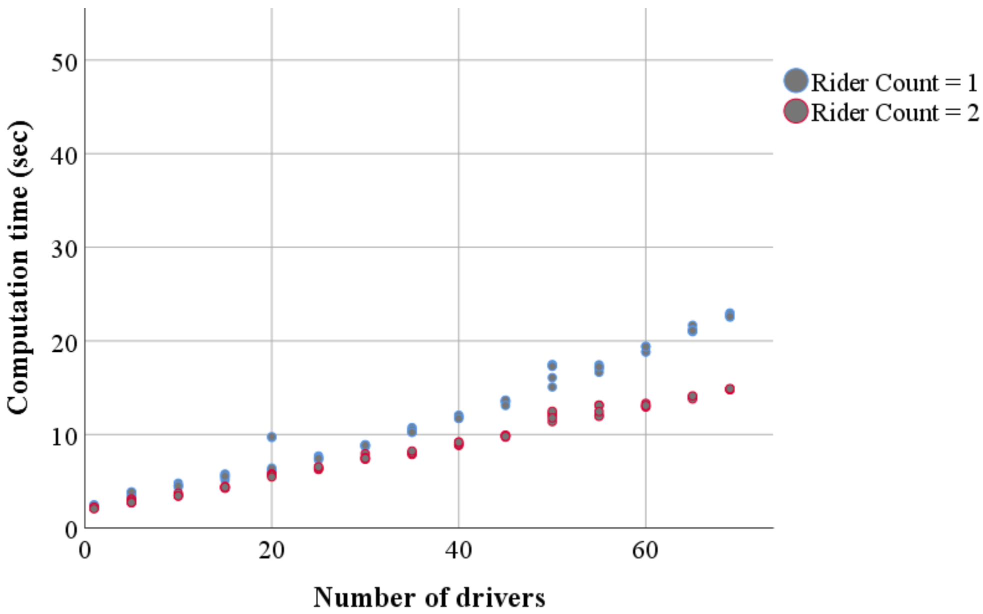

Figure 6.10. Computation time versus number of matched drivers by rider count.

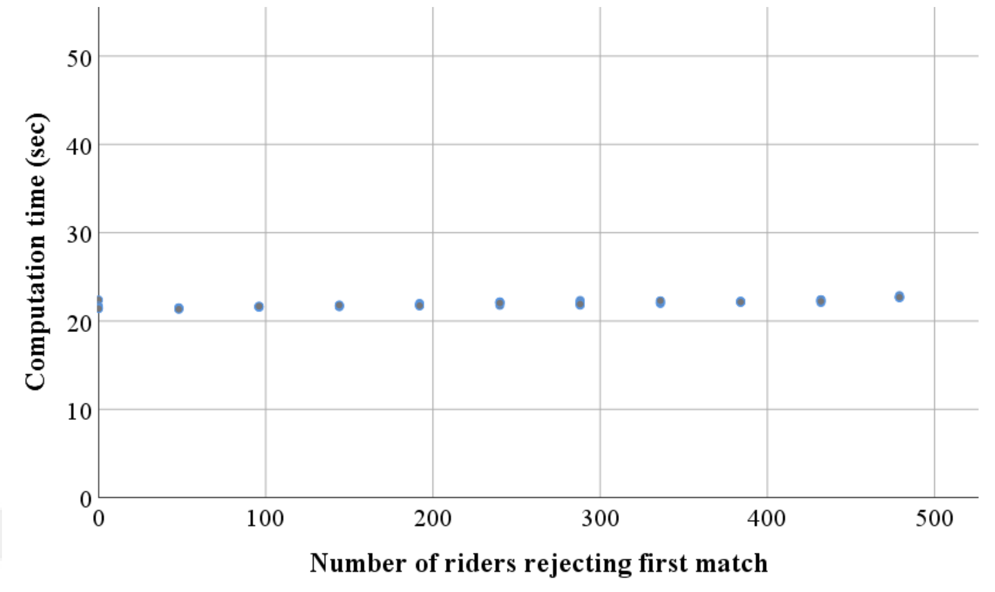

Figure 6.11. Computation time versus number of matched riders rejecting first match.

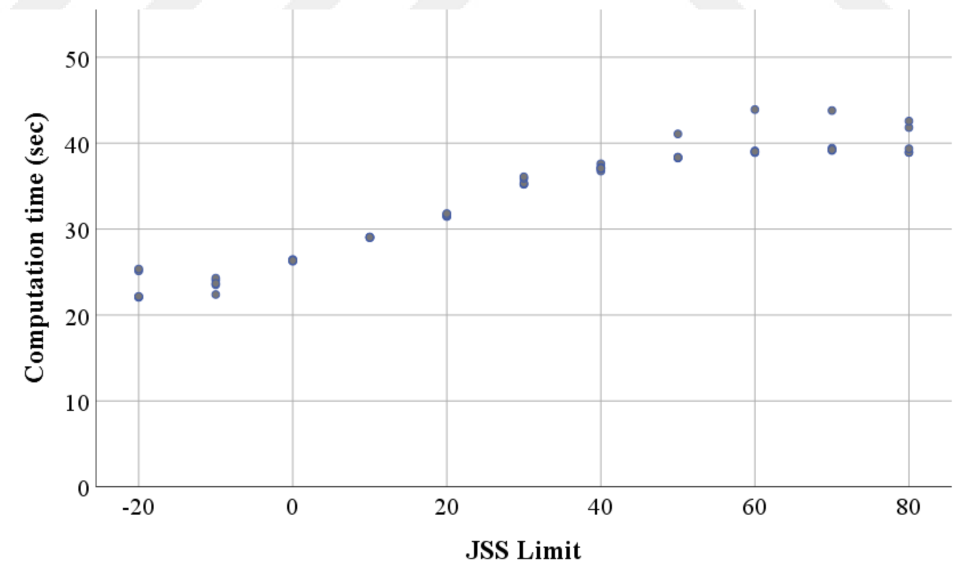

Figure 6.12. Computation time versus JSS limit.

6.2.3. Analysis for Average JSSs of Matched Pairs

In this section, average JSSs of Matched Pairs with variable independent parameters are investigated using multiple regression analysis. Summary output of multiple linear regression model is presented in Table 6.7.

Table 6.7. Model summary of the regression for average JSSs of matched pairs.

Model Summary$_b$				
Model	**R**	**R Square**	**Adjusted R Square**	**Std. Error of the Estimate**
1	0.826$_a$	0.683	0.680	6.950573994
a Predictors: (Constant), JSS Limit, Number of riders accepting first match, Rider count, Number of riders, Capacity of each driver's vehicle, Number of drivers				
b Dependent Variable: Average JSS				

The R^2 value states that the regression model can explain 68.3 % of the relationship between the dependent variable, average JSSs of matched pairs, and independent variables, numbers of riders, numbers of drivers, capacities of drivers, rider counts, numbers of riders who reject first match and JSS limits. Therefore, it can concluded that the regression model is good fit for the given data.

The analysis of variance (ANOVA) results for computation times are shown in Table 6.8. Large F and 0.000 significance value indicates that the independent variables statistically significant to predict the dependent variable. In other words, the regression model for computation times is good fit of the data.

Based on the results depicted in Table 6.9, the equation for computation time is as follows:

Table 6.8. ANOVA of the regression for average JSSs of matched pairs.

ANOVA $_a$					
Model	**Sum of Squares**	**df**	**Mean Square**	**F**	**Sig.**
Regression	62716.225	6	10452.704	216.365	0.000_b
Residual	29131.219	603	48.310		
Total	91847.443	609			
a Dependent Variable: Average JSS					
b Predictors: (Constant), JSS Limit, Number of riders accepting first match, Rider count, Number of riders, Capacity of each driver's vehicle, Number of drivers					

$$Y_{jss} = 45.800 - 0.020X_{nor} - 0.013X_{nod} + 0.204X_{cod} - 0.775X_{rc} + 0.001X_{nrj} + 0.290X_{jss}. \tag{6.5}$$

In Equation 6.5, $Y_{jss}, X_{nor}, X_{nod}, X_{cod}, X_{rc}, X_{nrj}, X_{jss}$ represent JSSs of matched pairs, number of riders, number of drivers, capacity of drivers, rider count, number of riders rejecting first match and JSS limit, respectively. The statistical significance of these independent variables presented in Table 6.9. This results state that number of riders and JSS limit are statistically significant parameters to explain JSSs of matched pairs. Based on regression analysis, the following results can be made:

- Figure 6.13 and Figure 6.14 show that when number of riders increases, average JSSs are slightly decreased. Although, number of riders are found to be statistically significant, these slight changes may be explained by the randomness of the data. In Figure 6.13, high driver capacity causes increase in average JSSs. In Figure 6.14, increase in rider count led to decrease in average JSSs. However, capacity of drivers and rider count are not statistically significant to explain average JSSs.

Table 6.9. Coefficients of the regression for average JSSs of matched pairs.

Coefficients $_a$					
	Unstandardized Coefficients		**Standardized Coefficients**		
Model	**B**	**Std. Error**	**Beta**	**t**	**Sig.**
(Constant)	45.800	2.202		20.798	0.000
Number of riders	-0.020	0.002	-0.235	-8.411	0.000
Number of drivers	-0.013	0.015	-0.024	-0.849	0.396
Capacity of each driver's vehicle	0.204	0.426	0.013	0.478	0.633
Rider count	-0.775	0.812	-0.026	-0.954	0.341
Number of riders rejecting first match	0.001	0.004	0.007	0.249	0.803
JSS Limit	0.290	0.009	0.835	31.276	0.000
a Dependent Variable: Average JSS					

- Figure 6.15 and Figure 6.16 show that number of drivers, capacities and rider count do not affect average JSSs, significantly. Table 6.9 indicates that these parameters are not statistically significant to explain average JSSs.
- Figure 6.17 shows that number of riders rejecting first match does not affect average JSSs. Results presented in Table 6.9 state that this parameter is not statistically significant for average JSSs.
- Figure 6.18 shows that setting higher JSS limit causes significant increase in average JSSs. The matching algorithm rejects matches having lower JSS than a JSS limit. Therefore, average JSSs of the matched pairs increases when higher JSS limits are set.

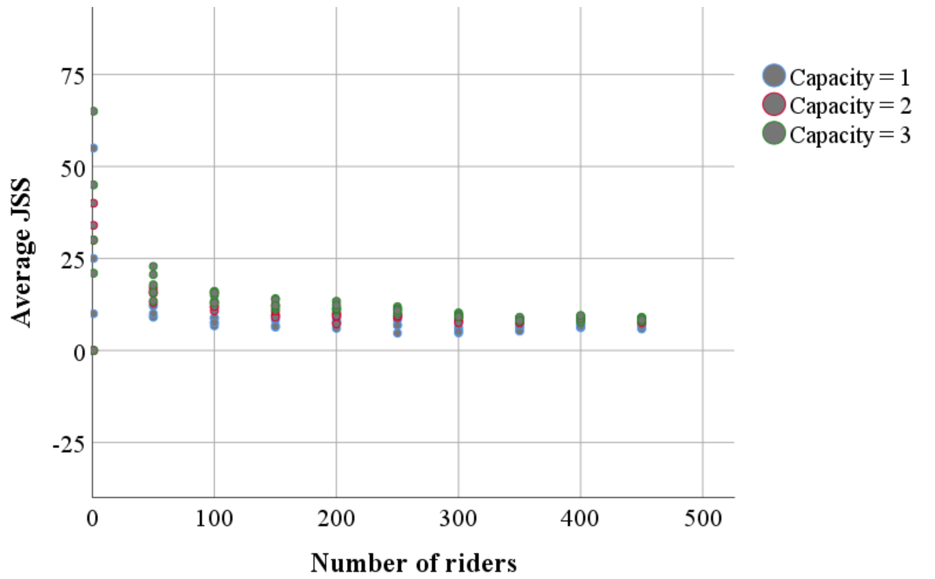

Figure 6.13. Average JSSs of matched pairs versus number of riders by capacities of drivers.

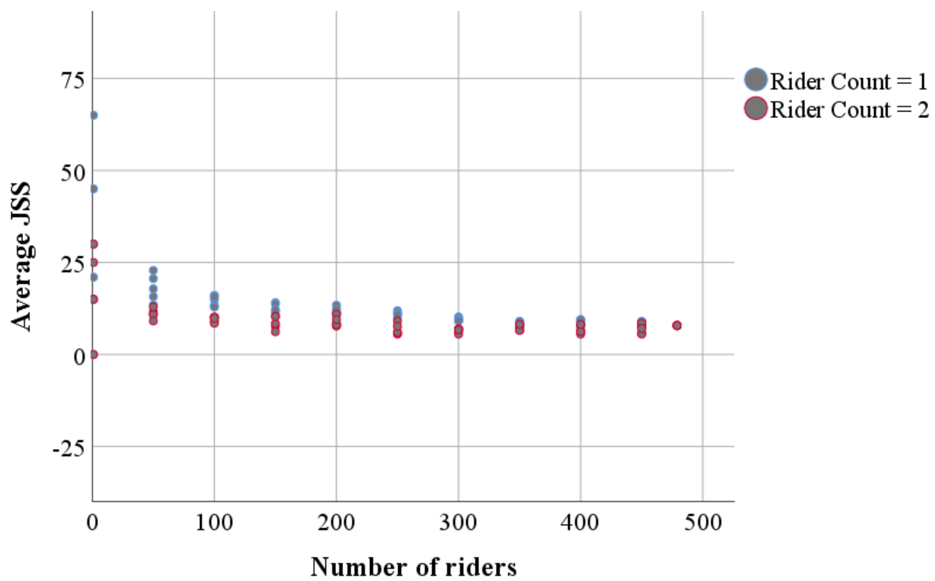

Figure 6.14. Average JSSs of matched pairs versus number of riders by rider count.

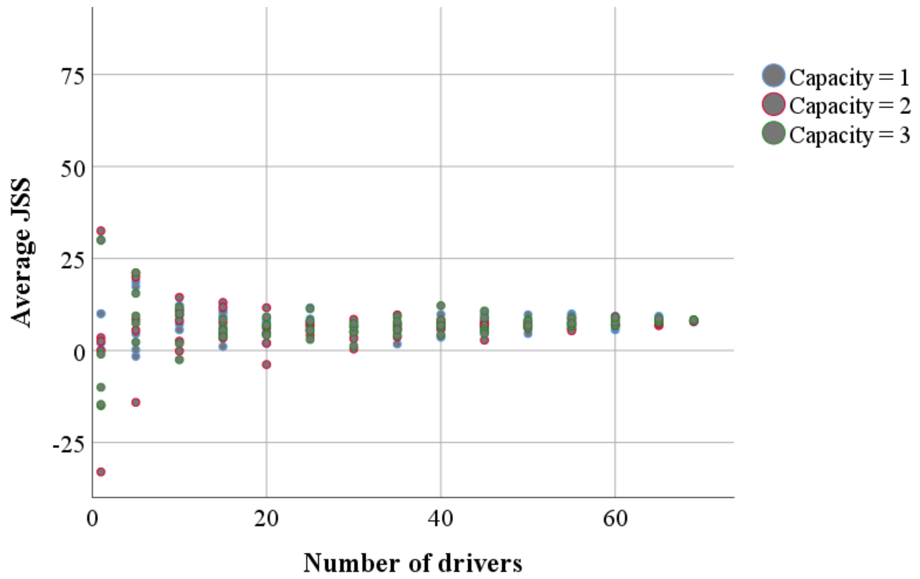

Figure 6.15. Average JSSs of matched pairs versus number of drivers by different capacities.

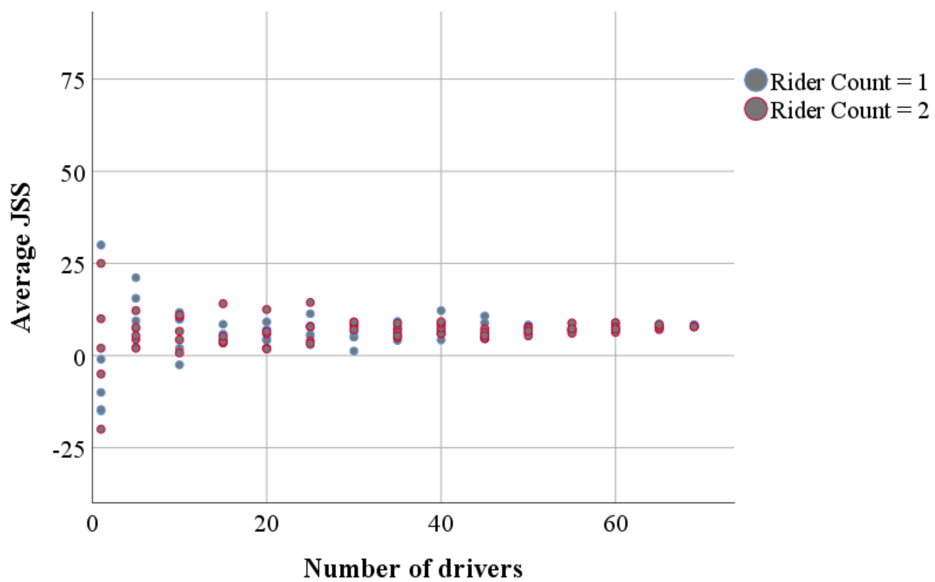

Figure 6.16. Average JSSs of matched pairs versus number of matched drivers by rider count.

Figure 6.17. Average JSSs of matched pairs versus number of matched riders rejecting first match.

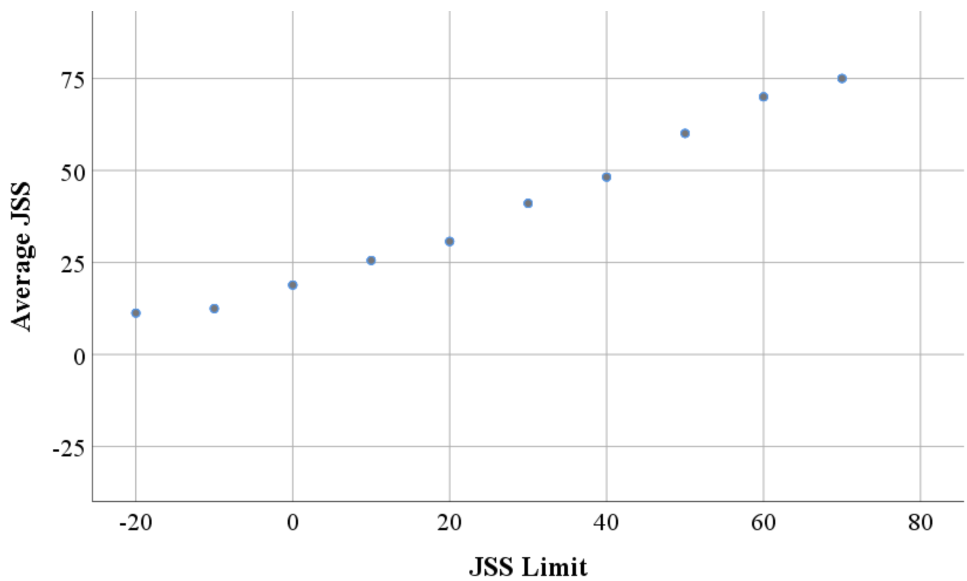

Figure 6.18. Average JSSs of matched pairs versus JSS limit.

7. CONCLUSION & DISCUSSIONS

In this chapter, contributions of the dissertation, discussion of the results, the limitations of the study and future recommendations are presented.

7.1. Summary of the Contributions

- A literature review on ride-sharing is conducted. Several ride-matching algorithms in the literature are investigated based on their objective functions, parameters and assumptions.
- A sequence alignment algorithm used in the bioinformatics field, namely the Needleman-Wunsch algorithm, is utilized to check route alignment. In the proposed algorithm, drivers' routes are assumed to be prespecified and fixed. The Needleman-Wunsch algorithm checks if the origin and the destination of a rider are located on the routes of drivers.
- The proposed ride-matching algorithm includes the social parameters of age, gender, employment and willingness to meet new people. To the best of the author's knowledge, social parameters are included in the objective function of a ride-matching algorithm for the first time. Similar parameters have been presented in the literature as constraints [38]. In the proposed algorithm, the similarities between these parameters are scored. A new parameter, namely JSS, is defined to represent social compatibility between participants.
- The importance of social characteristics and choices of the TAU students and staff were revealed based on a stated preference survey, conducted among 604 students and members of staff at the TAU in 2018.
- The effects of parameters of the proposed algorithm on ratio of matched participants, computation times and average JSSs of the matched pairs were analyzed.

7.2. Conclusions and Discussions

- In this dissertation, a review on several ride-matching algorithms is conducted based on their objective functions, parameters and assumptions and a new ride-matching is proposed. Most of these ride-matching algorithms focused on system wide benefits. Some objective functions are maximizing total distance savings, minimizing total travel time, minimizing total travel cost, maximizing number of matches and maximizing total waiting and delay time. These approaches assumed that participants would want to be matched to maximize system benefits; however, they would want to maximize their own benefits. Furthermore, these algorithms assumed that participants would accept the matches found by these algorithms.
- FCFS approach is utilized in the proposed algorithm to decrease waiting times of participants to be matched. Traditional optimization approaches calculate all possible matches to find the best matches. Rolling horizon approach used in some algorithms forces participants to wait until a deadline in order to include participants, who make ride requests after the initialization of the matching process. Although, this approach improves the results of the objective functions, participants would not want to wait a long period of time. Utilizing FCFS approach, the proposed algorithm finds a match for the rider with the earliest announcement time and eliminates the rider from the database before searching a match for other participants. In this way, waiting time of a participant decreases. Additionally, number of computation decreases, because the matched riders and drivers are eliminated from the database before searching a match for the remaining participants. Consequently, utilizing FCFS approach decreases computation times, but the best match to maximize system-wide JSSs may not be found.
- The outcomes of the survey conducted for this dissertation state that social parameters, such as gender, age, employment and willingness to meet new people, are significant for participants to be included in a ride-sharing system. A new parameter, JSS, is presented to score social compatibility of the participants. The objective function of the proposed algorithm is set to be maximizing total JSS. In the literature, social parameters were used as constraints. This approach

significantly decreases number of matches. In the proposed algorithm, number of matches remains same with number of matches found by the algorithms excluding social parameters; because the proposed algorithm can match participants with low JSSs if there is no better option.

- The results of the simulation study indicate that the computation times of the proposed algorithm are significantly lower than the traditional weighted bipartite algorithm and the optimization algorithm with social parameters. When subjected to the same constraints, the proposed algorithm's computation time was 12.6 times lower than that of the weighted bipartite algorithm and 14.7 times lower than that of the optimization algorithm. The quality of matches was analyzed using JSS values. When social concerns were omitted, the proposed algorithm yielded a 25% higher average JSS value compared to the weighted bipartite algorithm. When social factors were included, the optimization algorithm presented better results; however, the number of matches decreased dramatically compared to the proposed algorithm.
- Sensitivity analysis for ratio of matched riders, computation times and average JSSs of matched pairs are conducted to analyze the effects of numbers of riders, numbers of drivers the ratio of matched riders, capacities of drivers, rider counts, numbers of riders who reject first match and JSS limits. Ratio of matched riders increases with number of drivers and capacities of drivers, whereas it decreases with increasing number of riders and rider count. Number of riders rejecting first match causes a slight increase in ratio of matched riders. This is because, there are limited number of drivers and large number of riders. Higher JSS limit led to decrease in ratio of matched riders, because some matches are rejected due to low JSSs.
- Computation times are directly related to number of computations. Therefore, it increases with increasing number of number of riders, number of drivers and capacities of drivers and it decreases with increasing number of rider count. Similarly, setting JSS limit increases computation times.

- Average JSSs of matched pairs are mainly affected by JSS limit. Setting a JSS limit forces the algorithm to reject matches with JSSs lower than JSS limit. Therefore, increasing JSS limit results in increase in average JSS.

7.3. Limitations of the Study

- In this dissertation, the general limitation is insufficient data set to simulate the ride-matching algorithms. Since the real travel demands with travel times and routes were needed, outcomes of the stated preference survey conducted at the TAU were transformed into travel demands by assuming all participants travel from their home to the TAU and they wanted to be at the TAU at the same time. These assumptions are made to test the algorithms with the highest possible number; however, if the participants do not travel at the same time, number of available participants to be matched may be very small to find successful matches.
- In this dissertation, it is assumed that all drivers use the same predefined route, but drivers may choose different routes in real life. Route flexibility is ignored in the proposed algorithm, but this can be included into the proposed algorithm by offering a route choice to drivers.
- The riders were assumed to be willing to go to the nearest meeting points; however, the distances between homes of the riders and meeting points are not taken into account.
- The survey is conducted in the TAU, which has limited public transportation options. The importance of factors affecting participants' travel behavior may vary for other campuses.

7.4. Recommendations

- To achieve critical mass in ride-sharing, demands of potential participants, such as willingness to be matched from the same gender or age, should be defined carefully and they should be included in ride-matching algorithms. The importance of the objective functions, parameters and assumptions utilized in ride-matching algorithm should be examined.

- To achieve critical mass in ride-sharing, number of drivers in the system is very crucial. In future studies, implementations to increase number of drivers should be discussed and incentives for drivers should be investigated.
- The variable, x, is defined as positive one or negative one depending on the similarity of social characteristics and choices of drivers and riders. The algorithm can be extended by utilizing fuzzy logic for the x variable.
- Using the Needleman-Wunsch algorithm, the proposed algorithm can be developed. The Needleman-Wunsch algorithm allows the route of a driver to be split into matched parts and unmatched parts so that the driver can be matched with other riders if origin and destinations of the riders are located on the unmatched part of the driver's route. Similarly, multi-hop ride-sharing can be added. Consequently, ratio of matched riders and drivers can be increased.
- The effects of adding new parameters into the proposed algorithm should be investigated. Furthermore, meeting locations are represented by letters, but it is possible to using more than one letters to describe a location or even a time stamp. This may bring advantages, such as building dynamic routing, but computational burden may increase. A trade-off analysis for such upgrades for the proposed algorithm can be discussed in the future.
- The importance of social parameters and travel choices of potential participants should be investigated by conducting surveys in other campuses. Similarly, the survey is conducted among the students and staff of a university, but this can be extended by conducting surveys among other types of participants. Thus, better understanding of participants' travel behaviour can be achieved.
- The proposed biosequence based ride-sharing algorithm can be extended by utilizing different algorithms, such as Fast Optimal Global Sequence Alignment Algorithm, to increase computation time performance.

REFERENCES

1. Baichoo, S. and C. A Ouzounis, "Computational complexity of algorithms for sequence comparison, short-read assembly and genome alignment", *Bio Systems*, Vol. 156, 04 2017.

2. Singla, N. and D. Garg, "String Matching Algorithms and their Applicability in various Applications", *International Journal of Soft Computing and Engineering (IJSCE)*, Vol. 1, 01 2012.

3. Agatz, N., A. Erera, M. Savelsbergh and X. Wang, "Optimization for dynamic ride-sharing: A review", *European Journal of Operational Research*, Vol. 223, No. 2, pp. 295 – 303, 2012, `http://www.sciencedirect.com/science/article/pii/S0377221712003864`.

4. Kalan, O., "Application of Region Specific Depreciation Formulas in Highway Capital Stocks: Evidence from New York and New Jersey", *Transportation Research Board, 98th Annual Meeting*, 2019.

5. Litman, T., *Evaluating Transportation Economic Development Impacts*, Victoria Transport Policy Institute, 2010.

6. Kalan, O., *The Relationship Between Economic Growth and Road Infrastructure: Evidence from Turkey*, Ph.D. Thesis, Bogazici University, 2017.

7. Gokasar, I., Bayrak, M. and O. Kalan, "Bogazici Universitesi'nde Bisiklet Kullaniminin Yayginlastirilmasi", *11. Ulastirma Kongresi*, 2015.

8. Gokasar, I. and O. Kalan, "Dunyada ve Turkiye'de AUS icin Yapilan Yatirimlar", *Karayolu 3. Ulasim Kongresi*, 2014.

9. Gao, J., K. Ozbay, H. Nassif and O. Kalan, "Stochastic Multi-Objective

Optimization-Based Life Cycle Cost Analysis for New Construction Materials and Technologies", *Transportation Research Record*, 2019.

10. Kalan, O., A. Kurkcu and K. Ozbay, "Is Additive Utility Function Always a Sufficient Method in the Project Prioritization Process? Bridge Management Perspective", *Transportation Research Record*, 2019.

11. European Environment Agency EEA, *Household consumption and the environment*, 11, European Environment Agency, 2005, `https://www.eea.europa.eu/publications/eea_report_2005_11`, accessed in April 2019.

12. Macrotrends, *Crude Oil Prices - 70 Year Historical Chart*, 2018, `https://www.macrotrends.net/1369/crude-oil-price-history-chart`, accessed in December 2018.

13. European Environment Agency EEA, *Occupancy rates of passenger vehicles*, European Environment Agency, 2015, `https://www.eea.europa.eu/data-and-maps/indicators/occupancy-rates-of-passenger-vehicles/occupancy-rates-of-passenger-vehicles`, accessed in April 2019.

14. Masoud, N. and R. Jayakrishnan, "A decomposition algorithm to solve the multi-hop Peer-to-Peer ride-matching problem", *Transportation Research Part B: Methodological*, Vol. 99, pp. 1 – 29, 2017.

15. Agatz, N. A., A. L. Erera, M. W. Savelsbergh and X. Wang, "Dynamic ride-sharing: A simulation study in metro Atlanta", *Transportation Research Part B: Methodological*, Vol. 45, No. 9, pp. 1450 – 1464, 2011, `http://www.sciencedirect.com/science/article/pii/S0191261511000671`.

16. Jiang, W., C. R. Dominguez, P. Zhang, M. Shen and L. Zhang, "Large-scale nationwide ridesharing system: A case study of Chunyun", *International Journal of Transportation Science and Technology*, Vol. 7, No. 1, pp. 45 – 59, 2018, `http://www.sciencedirect.com/science/article/pii/S2046043017300552`.

17. Chan, N. D. and S. A. Shaheen, “Ridesharing in North America: Past, Present, and Future”, *Transport Reviews*, Vol. 32, No. 1, pp. 93–112, 2012, `https://doi.org/10.1080/01441647.2011.621557`.

18. Masoud, N. and R. Jayakrishnan, “A real-time algorithm to solve the peer-to-peer ride-matching problem in a flexible ridesharing system”, *Transportation Research Part B: Methodological*, Vol. 106, pp. 218 – 236, 2017, `http://www.sciencedirect.com/science/article/pii/S0191261517301169`.

19. Furuhata, M., M. Dessouky, F. Ordóñez, M.-E. Brunet, X. Wang and S. Koenig, “Ridesharing: The state-of-the-art and future directions”, *Transportation Research Part B: Methodological*, Vol. 57, pp. 28 – 46, 2013, `http://www.sciencedirect.com/science/article/pii/S0191261513001483`.

20. Gu, Q.-P., J. L. Liang and G. Zhang, “Algorithmic analysis for ridesharing of personal vehicles”, *Theoretical Computer Science*, Vol. 749, pp. 36 – 46, 2018, `http://www.sciencedirect.com/science/article/pii/S030439751730628X`, combinatorial Optimization and Applications.

21. Qian, X., W. Zhang, S. V. Ukkusuri and C. Yang, “Optimal assignment and incentive design in the taxi group ride problem”, *Transportation Research Part B: Methodological*, Vol. 103, pp. 208 – 226, 2017, `http://www.sciencedirect.com/science/article/pii/S0191261516306166`, green Urban Transportation.

22. Herbawi, W. and Weber, M., “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, 2012.

23. Najmi, A., D. Rey and T. H. Rashidi, “Novel dynamic formulations for real-time ride-sharing systems”, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 108, pp. 122 – 140, 2017, `http://www.sciencedirect.com/science/article/pii/S1366554517300017`.

24. Filcek, G., M. Hojda and J. Żak, "A heuristic algorithm for solving a Multiple Criteria Carpooling Optimization (MCCO) problem", *Transportation Research Procedia*, Vol. 27, pp. 656 – 663, 2017, http://www.sciencedirect.com/science/article/pii/S2352146517310050, 20th EURO Working Group on Transportation Meeting, EWGT 2017, 4-6 September 2017, Budapest, Hungary.

25. Edrissi, A., H. Poorzahedy, H. Nassiri and M. Nourinejad, "A multi-agent optimization formulation of earthquake disaster prevention and management", *European Journal of Operational Research*, Vol. 229, No. 1, pp. 261 – 275, 2013, http://www.sciencedirect.com/science/article/pii/S0377221713002166.

26. Yan, S., C. Chen and Y. Lin, "A Model With a Heuristic Algorithm for Solving the Long-Term Many-to-Many Car Pooling Problem", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12, No. 4, pp. 1362–1373, Dec 2011.

27. Maniezzo, V., Carbonaro, A., Vigo, D. and H. Hildmann, "An ANTS Heuristic for the Long – Term Car Pooling Problem", *Proceedings of ANTS'2000, 2nd International Conference Ant Colony Optimization*, pp. –, 2000.

28. Cheikh, S. B. and S. Hammadi, "Multi-Hop Ridematching optimization problem: Intelligent chromosome agent-driven approach", *Expert Systems with Applications*, Vol. 62, pp. 161 – 176, 2016, http://www.sciencedirect.com/science/article/pii/S0957417416302822.

29. Nielsen, J. R., H. Hovmøller, P.-L. Blyth and B. K. Sovacool, "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*, Vol. 78, pp. 113 – 123, 2015, http://www.sciencedirect.com/science/article/pii/S0965856415001196.

30. Stiglic, M., N. Agatz, M. Savelsbergh and M. Gradisar, "The benefits of meeting points in ride-sharing systems", *Transportation Research Part B: Methodological*, Vol. 82, pp. 36 – 53, 2015, http://www.sciencedirect.com/science/article/

pii/S0191261515002088.

31. Nourinejad, M. and M. J. Roorda, "Agent based model for dynamic ridesharing", *Transportation Research Part C: Emerging Technologies*, Vol. 64, pp. 117 – 132, 2016, http://www.sciencedirect.com/science/article/pii/S0968090X15002661.

32. Ma, S. , Zheng, Y. and Wolfson, O., "T-share: a large-scale dynamic taxi ridesharing service", *Data Engineering (ICDE), 2013 IEEE 29th International Conference on. IEEE*, pp. 410–421, 2013.

33. Chen, P.-Y. , Liu, J.-W. , Chen, W.-T., "A fuel-saving and pollution-reducing dynamic taxi-sharing protocol in vanets", *Vehicular Technology Conference Fall (VTC 2010-Fall)*, pp. 1–5, 2010.

34. Febbraro, A. D., E. Gattorna and N. Sacco, "Optimization of Dynamic Ridesharing Systems", *Transportation Research Record*, Vol. 2359, No. 1, pp. 44–50, 2013, https://doi.org/10.3141/2359-06.

35. Santos, D.O. and Xavier, E.C., "Dynamic taxi and ridesharing: a framework and heuristics for the optimization problem", *IJCAI*, Vol. 13, p. 2885–2891, 2013.

36. Santos, D. O. and E. C. Xavier, "Taxi and Ride Sharing: A Dynamic Dial-a-Ride Problem with Money as an Incentive", *Expert Systems with Applications*, Vol. 42, No. 19, pp. 6728 – 6737, 2015, http://www.sciencedirect.com/science/article/pii/S0957417415003024.

37. Santi, P., G. Resta, M. Szell, S. Sobolevsky, S. H Strogatz and C. Ratti, "Quantifying the benefits of vehicle pooling with shareability networks", *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 111, 09 2014.

38. Ghoseiri, K., Haghani, A. and Hamedi, M., *Real-Time Rideshare Matching Prob-*

lem, Ph.D. Thesis, University of Maryland, Department of Civil and Environmental Engineering, 2011.

39. Wang, X., N. Agatz and A. Erera, "Stable Matching for Dynamic Ride-Sharing Systems", *ERIM Report Series Research in Management*, 07 2014.

40. Keshk, A., M. Ossman and L. Fathi Hussein, "Fast Longest Common Subsequences for Bioinformatics Dynamic Programming", *International Journal of Computer Applications*, Vol. 57, pp. 12–18, 11 2012.

41. Needleman, S. B. and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins", *Journal of Molecular Biology*, Vol. 48, No. 3, pp. 443 – 453, 1970, `http://www.sciencedirect.com/science/article/pii/0022283670900574`.

42. Boutet de Monvel, J., "Extensive simulations for longest common subsequences", *The European Physical Journal B - Condensed Matter and Complex Systems*, Vol. 7, No. 2, pp. 293–308, Jan 1999, `https://doi.org/10.1007/s100510050616`.

43. Watson, J. D., Hopkins, N. H., Roberts, J. W., Steitz, J. A., Weiner, A. M., *Molecular Biology of the Gene*, Vol. 5, Pearson Education, Inc, 2004.

44. Shoval, N. and M. Isaacson, "Sequence Alignment As A Method for Human Activity Analysis in Space and Time", *Annals of the Association of American Geographers*, Vol. 97, pp. 282 – 297, 06 2007.

45. Crawford, F., D. Watling and R. Connors, "Identifying road user classes based on repeated trip behaviour using Bluetooth data", *Transportation Research Part A: Policy and Practice*, Vol. 113, pp. 55 – 74, 2018, `http://www.sciencedirect.com/science/article/pii/S096585641730352X`.

46. Shaheen, S. A., N. D. Chan and T. Gaynor, "Casual carpooling in the San Francisco Bay Area: Understanding user characteristics, behaviors, and motivations",

Transport Policy, Vol. 51, pp. 165 – 173, 2016, http://www.sciencedirect.com/science/article/pii/S0967070X16300038.

47. Xing, X., T. Warden, T. Nicolai and O. Herzog, "SMIZE: A Spontaneous Ride-Sharing System for Individual Urban Transit", *Proceedings of the 7th German Conference on Multiagent System Technologies, MATES'09. Springer-Verlag, Berlin Heidelberg*, pp. 165–176, 09 2009.

48. Heinrich, S., *Implementing Real-time Ridesharing in the San Francisco Bay Area*, Ph.D. Thesis, San Jose State University, 2010.

49. Gunturu, S., *Load scheduling for bioinformatics: Applications in large scale networks*, Ph.D. Thesis, Jawaharlal Nehru Technological University, 2005.

50. Gokasar, I. and O. F. Aydın, "Trafik Yonetim Stratejileri: FSM Koprusu Bakim Calismalari Ornegi", *IMO Teknik Dergi*, Vol. 25, pp. 6655–6678, 2014, http://www.imo.org.tr/resimler/dosya_ekler/146de07f4cbc7dc_ek.pdf?dergi=542.

51. Ergun, G. and N. Sahin, "Development of Traffic Congestion Management Strategies: Strategic Plan Studies", *Final Report prepared for a grant received from Istanbul Metropolitan Planning Bureau*, p. 1, 2005.

52. Schroeder, L., D. Sjoquist and P. Stephan, *Understanding Regression Analysis: An Introductory Guide*, SAGE Publications, 2016.

53. Douglas C. Montgomery, E. A. P. and G. G. Vining, *Introduction to Linear Regression Analysis*, Wiley, 2012.

54. Freund, R. J., W. J. Wilson and P. Sa, *Regression Analysis*, Elsevier, 2006.

55. Fumo, N. and M. R. Biswas, "Regression analysis for prediction of residential energy consumption", *Renewable and Sustainable Energy Reviews*, Vol. 47, pp. 332 – 343, 2015.

APPENDIX A: CAMPUS TRANSPORTATION SURVEY

Tarih:

TÜRK - ALMAN ÜNİVERSİTESİ ULAŞIM TERCİHLERİ ANKETİ

Bu anket Türk - Alman Üniversitesi, Mühendislik Fakültesi Yenilik ve Teknoloji Yönetimi Projesi kapsamında yapılmaktadır. Anketin amacı ulaşım ve park sorunlarını araştırmaktır.
*Araştırma bilimsel bir nitelik taşıdığından derlenen kişi ve aile bilgileri gizli tutulacaktır.
*Lütfen soruları tam olarak okuduktan sonra kendinize en uygun olan cevabı işaretleyiniz.

Öğrenci	Akademik Personel	İdari Personel	Çalışan
( )	( )	( )	( )

Yaşınız : Cinsiyetiniz: E ( ) K( )
Bölümünüz :
Kaçıncı döneminiz :
İkamet Ettiğiniz İlçe :
Size en yakın otobüs durağının ismi :
Nerede ikamet ediyorsunuz? Aile ile ( ) Yurt ( ) Öğrenci evi ( )
İkamet yerinizin okula uzaklığı kaç dakikadır? .... dk
Kendi aracınız var mı ? Evet ( ) Hayır ( )
Okula gelmek için kullandığınız ailenize ait araç var mı ? Evet ( ) Hayır ( )

		Asla		(Haftada Kaç Gün)			Her Zaman
1.	Okula özel araç ile **gelmediğinizde;**	0	1	2	3	4	5
1.1.	En çok hangi ulaşım aracını tercih ediyorsunuz ?						
1.1.1.	Otobüs	( )	( )	( )	( )	( )	( )
1.1.2.	Taksi	( )	( )	( )	( )	( )	( )
1.1.3.	Vapur	( )	( )	( )	( )	( )	( )
1.1.4.	Shuttle	( )	( )	( )	( )	( )	( )
1.1.5.	Okul Personel Servisi	( )	( )	( )	( )	( )	( )
1.1.6.	Aracı olan arkadaşımla gelirim.	( )	( )	( )	( )	( )	( )
1.1.7.	Diğer:	( )	( )	( )	( )	( )	( )
1.2.	Otobüs tercihinizde en önemli faktör nedir?						
1.2.1.	Erken gelmesi	( )	( )	( )	( )	( )	( )
1.2.2.	Seyahat süresinin kısa olması	( )	( )	( )	( )	( )	( )
1.2.3.	Aktarmasız okula gitmesi	( )	( )	( )	( )	( )	( )
1.2.4.	Okulun önüne kadar gitmesi	( )	( )	( )	( )	( )	( )
1.2.5.	Konforlu olması	( )	( )	( )	( )	( )	( )
1.2.6.	Diğer:	( )	( )	( )	( )	( )	( )
1.3.	Okula toplu ulaşımla geldiğimde aktarma sayısı az olan rotayı tercih ederim.	( )	( )	( )	( )	( )	( )
1.4.	Okula toplu ulaşımla geldiğimde ulaşım süresi kısa olan rotayı tercih ederim.	( )	( )	( )	( )	( )	( )
1.5.	Okula gelirken en çok hangi otobüsü/otobüsleri tercih ediyorsunuz ?	1.......	2.......	3.......	4.......	5.......	6.......
1.6.	Herhangi bir toplu ulaşım aracına varmak için şu kadar yürüyebilirim.	1 dk	5 dk	10 dk	20 dk	30 dk	30+ dk
		( )	( )	( )	( )	( )	( )

		Asla		(Haftada Kaç Gün)			Her Zaman
2.	Okula gelmek için **özel aracınız varsa;**	0	1	2	3	4	5
2.1.	Aracımla okula kadar gelirim.	( )	( )	( )	( )	( )	( )
2.2.	Aracımı uygun bir yere park eder ve toplu ulaşımla devam ederim.	( )	( )	( )	( )	( )	( )
2.3.	Masrafı fazla olduğu için belli bir yere kadar arabayla gelirim.	( )	( )	( )	( )	( )	( )
2.4.	Yalnızca vaktim dar olduğunda araç ile kullanırım.	( )	( )	( )	( )	( )	( )
2.5.	Daha konforlu olduğu için aracım ile gelirim.	( )	( )	( )	( )	( )	( )
2.6.	Masrafı fazla olduğu için kullanmam.	( )	( )	( )	( )	( )	( )
2.7.	Trafik sebebiyle araç ile gelmem.	( )	( )	( )	( )	( )	( )
2.8.	Okula gelirken en çok hangi yolları tercih ediyorsunuz ?						
2.8.1.	Riva Yolu	( )	( )	( )	( )	( )	( )
2.8.2.	Sahil yolu	( )	( )	( )	( )	( )	( )
2.8.3.	Diğer:	( )	( )	( )	( )	( )	( )
2.9.	Okula km olarak en kısa yolu tercih ederim.	( )	( )	( )	( )	( )	( )
2.10.	Okula süre olarak en kısa yolu tercih ederim.	( )	( )	( )	( )	( )	( )

Figure A.1. Campus transportation survey, page 1.

		Asla					Her Zaman
3.	**Araç paylaşımına genel olarak bakışınız:** (Otostop, Car-Pooling, Ride-Sharing vb.)	**0**	**1**	**2**	**3**	**4**	**5**
3.1.	Bakışım olumludur.	( )	( )	( )	( )	( )	( )
3.2.	Yabancılara güvenmediğim için başkasının aracına binmem veya başkasını aracıma almam.	( )	( )	( )	( )	( )	( )
3.3.	Araç paylaşacağım kişi/lerin TAÜ'den olması önemlidir.	( )	( )	( )	( )	( )	( )
3.4.	Otostop çekmemde/otostopçu almamda;						
3.4.1.	Hava durumu etkilidir.	( )	( )	( )	( )	( )	( )
3.4.2.	Saat etkilidir.	( )	( )	( )	( )	( )	( )
3.4.3.	Karşı tarafın cinsiyeti etkilidir.	( )	( )	( )	( )	( )	( )
3.4.4.	Karşı tarafın tanıdık olması önemlidir.	( )	( )	( )	( )	( )	( )
3.4.5.	Karşı tarafın üniversiteden olması önemlidir.	( )	( )	( )	( )	( )	( )
3.5.	Bir APP üzerinden ;						
3.5.1.	Okuldan biri ile araç paylaşırım.	( )	( )	( )	( )	( )	( )
3.5.2.	Herhangi biri ile araç paylaşırım.	( )	( )	( )	( )	( )	( )
3.5.3.	Hakkında olumlu yorumlar olan biriyle araç paylaşırım.	( )	( )	( )	( )	( )	( )
3.6.	Araç paylaşımının;						
3.6.1.	Trafik üzerindeki etkilerini önemserim.	( )	( )	( )	( )	( )	( )
3.6.2.	Çevre üzerindeki etkilerini (karbondioksit salınımı azaltması vb.) önemserim.	( )	( )	( )	( )	( )	( )
		Asla					**Her Zaman**
4.	Kampüse <u>**özel araç ile gelmiyorsanız**</u> araç paylaşımı için;	**0**	**1**	**2**	**3**	**4**	**5**
4.1.	Para veririm.	( )	( )	( )	( )	( )	( )
4.2.	Vereceğim ücret toplu ulaşım ile aynı ise kabul ederim.	( )	( )	( )	( )	( )	( )
4.3.	Vereceğim ücret toplu ulaşımın 2 katı kadar olur ise kabul edebilirim.	( )	( )	( )	( )	( )	( )
4.4.	Okula gelirken otostop çekerim.	( )	( )	( )	( )	( )	( )
4.5.	Okuldan eve giderken otostop çekerim.	( )	( )	( )	( )	( )	( )
4.6.	Araç sahibine göre her arabaya binmem.	( )	( )	( )	( )	( )	( )
4.7.	Araç modeline göre her arabaya binmem.	( )	( )	( )	( )	( )	( )
		1 dk	5 dk.	10 dk.	15 dk.	30 dk.	1+ saat
4.8.	Aracın kalkışı için maksimum ne kadar süre bekleyebilirsiniz.	( )	( )	( )	( )	( )	( )
4.9.	Araç paylaşımı için şu kadar mesafe yürüyebilirim.	( )	( )	( )	( )	( )	( )
4.10.	Sürücüsüz (otonom) araçlar olursa daha çok tercih ederim.	( )	( )	( )	( )	( )	( )
4.11.	<u>Okul dışından</u> bir/birkaç yabancıyla taksi paylaşabilirim.	( )	( )	( )	( )	( )	( )
4.12.	<u>Okuldan</u> bir/birkaç yabancıyla taksi paylaşabilirim.	( )	( )	( )	( )	( )	( )
4.13.	<u>Tanıdık</u> bir/birkaç kişiyle taksi paylaşabilirim.	( )	( )	( )	( )	( )	( )
4.14.	Daha önce Über gibi uygulamaları kullandım.	( )	( )	( )	( )	( )	( )
4.15.	Neden otostop çekerim ?						
4.15.1.	Toplu ulaşım olmadığında otostop çekerim.	( )	( )	( )	( )	( )	( )
4.15.2.	Ücretsiz olduğu için otostop çekerim.	( )	( )	( )	( )	( )	( )
4.15.3.	Yeni insanlar tanımak için otostop çekerim.	( )	( )	( )	( )	( )	( )
4.15.4.	Daha konforlu seyahat için otostop çekerim.	( )	( )	( )	( )	( )	( )

		Asla					Her Zaman
5.	Kampüse <u>**özel araç ile geliyorsanız**</u> araç paylaşımı için;	**0**	**1**	**2**	**3**	**4**	**5**
5.1.	Ücretini alarak paylaşabilirim.	( )	( )	( )	( )	( )	( )
5.2.	Ücretsiz yolcu ile paylaşmam.	( )	( )	( )	( )	( )	( )
5.3.	Okula gelirken otostopçu alırım.	( )	( )	( )	( )	( )	( )
5.4.	Okuldan çıkarken otostopçu alırım.	( )	( )	( )	( )	( )	( )
5.5.	Sadece üniversiteden otostopçu alırım.	( )	( )	( )	( )	( )	( )
5.6.	Üniversiteden de olsa her otostopçuyu kabul etmem.	( )	( )	( )	( )	( )	( )
5.7.	Araç paylaşımı için rotamı bir miktar değiştirebilirim.	( )	( )	( )	( )	( )	( )
5.8.	Araç paylaşımı için şu kadar süre beklerim.	( )	( )	( )	( )	( )	( )
5.9.	Otopark imkanı sadece aracını paylaşanlara verilirse aracımı paylaşabilirim.	( )	( )	( )	( )	( )	( )
5.10.	Masraflar azalırsa aracımla okula gelirim.	( )	( )	( )	( )	( )	( )
		1 dk	5 dk.	10 dk.	15 dk.	30 dk.	1+ saat
5.11.	Aracımı ücreti karşılığında paylaşırsam yolcuyu en fazla şu kadar süre beklerim:	( )	( )	( )	( )	( )	( )
5.12.	Daha önce Uber benzeri uygulamalarla aracıma yolcu aldım.	( )	( )	( )	( )	( )	( )
5.13.	Aracıma otostopçu almamda şunlar etkilidir:						
5.13.1.	Yeni insanlar tanımak için otostopçu alırım.	( )	( )	( )	( )	( )	( )
5.13.2.	İnsanlara yardımcı olmak için otostopçu alırım.	( )	( )	( )	( )	( )	( )
5.13.3.	Diğer:	( )	( )	( )	( )	( )	( )

Figure A.2. Campus transportation survey, page 2.

APPENDIX B: COMPUTATIONAL RESULTS

The following tables show the computational results of the proposed algorithm using different values.

Table B.1. Computational results of the proposed algorithm.

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
1	479	69	3	1	0	-100	22.7080	8.2513	191	64	39.87%	92.75%	100.00%	1
2	479	69	3	1	0	-100	22.9750	8.2513	191	64	39.87%	92.75%	100.00%	1
3	479	69	3	1	0	-100	22.5390	8.2513	191	64	39.87%	92.75%	100.00%	1
4	479	69	3	1	0	-100	22.8450	8.2513	191	64	39.87%	92.75%	100.00%	1
5	479	69	3	1	0	-100	22.6100	8.2513	191	64	39.87%	92.75%	100.00%	1
6	479	65	3	1	0	-100	21.6720	7.8548	179	60	37.37%	92.31%	100.00%	1
7	479	65	3	1	0	-100	20.9820	8.1844	179	60	37.37%	92.31%	100.00%	1
8	479	65	3	1	0	-100	21.0600	8.0279	179	60	37.37%	92.31%	100.00%	1
9	479	65	3	1	0	-100	21.1690	7.9553	179	60	37.37%	92.31%	100.00%	1
10	479	65	3	1	0	-100	21.0960	8.6313	179	60	37.37%	92.31%	100.00%	1
11	479	60	3	1	0	-100	19.4090	8.7746	173	58	36.12%	96.67%	100.00%	1
12	479	60	3	1	0	-100	19.4030	6.7866	164	55	34.24%	91.67%	100.00%	1
13	479	60	3	1	0	-100	18.8220	6.8264	167	56	34.86%	93.33%	100.00%	1
14	479	60	3	1	0	-100	18.8230	7.6886	167	57	34.86%	95.00%	100.00%	1
15	479	60	3	1	0	-100	19.3740	8.9364	173	58	36.12%	96.67%	100.00%	1
16	479	55	3	1	0	-100	17.4410	8.8224	152	51	31.73%	92.73%	100.00%	1
17	479	55	3	1	0	-100	17.0700	6.9557	158	54	32.99%	98.18%	100.00%	1
18	479	55	3	1	0	-100	17.1550	6.6000	155	52	32.36%	94.55%	100.00%	1
19	479	55	3	1	0	-100	16.6500	7.2081	149	50	31.11%	90.91%	100.00%	1
20	479	55	3	1	0	-100	17.2320	7.3484	155	53	32.36%	96.36%	100.00%	1
21	479	50	3	1	0	-100	16.0770	7.1571	140	48	29.23%	96.00%	100.00%	1
22	479	50	3	1	0	-100	17.4830	6.0979	143	48	29.85%	96.00%	100.00%	1
23	479	50	3	1	0	-100	17.3690	8.2937	143	48	29.85%	96.00%	100.00%	1

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
24	479	50	3	1	0	-100	17.2890	8.0699	143	48	29.85%	96.00%	100.00%	1
25	479	50	3	1	0	-100	15.0770	6.7687	134	45	27.97%	90.00%	100.00%	1
26	479	45	3	1	0	-100	13.5230	10.7200	125	42	26.10%	93.33%	100.00%	1
27	479	45	3	1	0	-100	13.5840	5.7680	125	42	26.10%	93.33%	100.00%	1
28	479	45	3	1	0	-100	13.4840	5.1985	131	45	27.35%	100.00%	100.00%	1
29	479	45	3	1	0	-100	13.6790	8.8640	125	43	26.10%	95.56%	100.00%	1
30	479	45	3	1	0	-100	13.1030	4.5820	122	41	25.47%	91.11%	100.00%	1
31	479	40	3	1	0	-100	11.8410	4.2241	116	40	24.22%	100.00%	100.00%	1
32	479	40	3	1	0	-100	11.7730	6.7876	113	39	23.59%	97.50%	100.00%	1
33	479	40	3	1	0	-100	12.0530	8.2389	113	39	23.59%	97.50%	100.00%	1
34	479	40	3	1	0	-100	12.0210	12.1776	107	36	22.34%	90.00%	100.00%	1
35	479	40	3	1	0	-100	11.7040	6.8053	113	38	23.59%	95.00%	100.00%	1
36	479	35	3	1	0	-100	10.7370	7.3579	95	32	19.83%	91.43%	100.00%	1
37	479	35	3	1	0	-100	10.2560	9.2174	92	31	19.21%	88.57%	100.00%	1
38	479	35	3	1	0	-100	10.4190	7.9053	95	32	19.83%	91.43%	100.00%	1
39	479	35	3	1	0	-100	10.5010	4.0865	104	35	21.71%	100.00%	100.00%	1
40	479	35	3	1	0	-100	10.2270	5.6733	101	35	21.09%	100.00%	100.00%	1
41	479	30	3	1	0	-100	8.7830	1.2247	89	30	18.58%	100.00%	100.00%	1
42	479	30	3	1	0	-100	8.9150	5.0698	86	30	17.95%	100.00%	100.00%	1
43	479	30	3	1	0	-100	8.7130	5.0225	89	30	18.58%	100.00%	100.00%	1
44	479	30	3	1	0	-100	8.8270	6.4625	80	28	16.70%	93.33%	100.00%	1
45	479	30	3	1	0	-100	8.8540	7.2875	80	27	16.70%	90.00%	100.00%	1
46	479	25	3	1	0	-100	7.7100	3.0000	74	25	15.45%	100.00%	100.00%	1

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
47	479	25	3	1	0	-100	7.3510	5.6351	74	25	15.45%	100.00%	100.00%	1
48	479	25	3	1	0	-100	7.5480	8.0270	74	25	15.45%	100.00%	100.00%	1
49	479	25	3	1	0	-100	7.3750	11.3380	71	24	14.82%	96.00%	100.00%	1
50	479	25	3	1	0	-100	7.4500	5.4460	74	25	15.45%	100.00%	100.00%	1
51	479	20	3	1	0	-100	6.1250	9.1356	59	20	12.32%	100.00%	100.00%	1
52	479	20	3	1	0	-100	6.2410	4.4407	59	20	12.32%	100.00%	100.00%	1
53	479	20	3	1	0	-100	9.6810	6.8305	59	20	12.32%	100.00%	100.00%	1
54	479	20	3	1	0	-100	9.7660	4.1500	60	20	12.53%	100.00%	100.00%	1
55	479	20	3	1	0	-100	6.4130	6.9333	60	20	12.53%	100.00%	100.00%	1
56	479	15	3	1	0	-100	5.1790	4.1364	44	15	9.19%	100.00%	100.00%	1
57	479	15	3	1	0	-100	5.6760	5.8444	45	15	9.39%	100.00%	100.00%	1
58	479	15	3	1	0	-100	5.4250	5.4000	45	15	9.39%	100.00%	100.00%	1
59	479	15	3	1	0	-100	5.7840	8.4667	45	15	9.39%	100.00%	100.00%	1
60	479	15	3	1	0	-100	5.5470	3.7273	44	15	9.19%	100.00%	100.00%	1
61	479	10	3	1	0	-100	4.8090	1.9333	30	10	6.26%	100.00%	100.00%	1
62	479	10	3	1	0	-100	4.5020	-2.5000	26	10	5.43%	100.00%	100.00%	1
63	479	10	3	1	0	-100	4.5480	9.8276	29	10	6.05%	100.00%	100.00%	1
64	479	10	3	1	0	-100	4.4380	11.7000	30	10	6.26%	100.00%	100.00%	1
65	479	10	3	1	0	-100	4.5270	9.9667	30	10	6.26%	100.00%	100.00%	1
66	479	5	3	1	0	-100	3.8210	2.2667	15	5	3.13%	100.00%	100.00%	1
67	479	5	3	1	0	-100	3.4980	15.5333	15	5	3.13%	100.00%	100.00%	1
68	479	5	3	1	0	-100	3.1140	7.4667	15	5	3.13%	100.00%	100.00%	1
69	479	5	3	1	0	-100	3.8730	9.4000	15	5	3.13%	100.00%	100.00%	1

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
70	479	5	3	1	0	-100	2.8770	21.1333	15	5	3.13%	100.00%	100.00%	1
71	479	1	3	1	0	-100	2.4890	-15.0000	3	1	0.63%	100.00%	100.00%	1
72	479	1	3	1	0	-100	2.2310	-14.6667	3	1	0.63%	100.00%	100.00%	1
73	479	1	3	1	0	-100	2.3490	-10.0000	3	1	0.63%	100.00%	100.00%	1
74	479	1	3	1	0	-100	2.1820	-1.0000	3	1	0.63%	100.00%	100.00%	1
75	479	1	3	1	0	-100	2.1860	30.0000	3	1	0.63%	100.00%	100.00%	1
76	479	69	2	1	0	-100	18.5500	8.0000	129	65	26.93%	94.20%	100.00%	2
77	479	69	2	1	0	-100	18.4420	8.0000	129	65	26.93%	94.20%	100.00%	2
78	479	69	2	1	0	-100	18.7150	8.0000	129	65	26.93%	94.20%	100.00%	2
79	479	69	2	1	0	-100	18.6560	8.0000	129	65	26.93%	94.20%	100.00%	2
80	479	69	2	1	0	-100	18.7240	8.0000	129	65	26.93%	94.20%	100.00%	2
81	479	65	2	1	0	-100	18.1410	6.7603	121	61	25.26%	93.85%	100.00%	2
82	479	65	2	1	0	-100	17.5220	7.9504	121	61	25.26%	93.85%	100.00%	2
83	479	65	2	1	0	-100	18.3050	7.2479	121	61	25.26%	93.85%	100.00%	2
84	479	65	2	1	0	-100	18.5590	7.9669	121	61	25.26%	93.85%	100.00%	2
85	479	65	2	1	0	-100	20.6780	7.7724	123	62	25.68%	95.38%	100.00%	2
86	479	60	2	1	0	-100	16.3970	6.9099	111	56	23.17%	93.33%	100.00%	2
87	479	60	2	1	0	-100	16.3050	8.3423	111	56	23.17%	93.33%	100.00%	2
88	479	60	2	1	0	-100	16.0390	6.8850	113	57	23.59%	95.00%	100.00%	2
89	479	60	2	1	0	-100	15.7630	6.9735	113	57	23.59%	95.00%	100.00%	2
90	479	60	2	1	0	-100	15.8440	9.1565	115	58	24.01%	96.67%	100.00%	2
91	479	55	2	1	0	-100	14.6390	5.8131	107	54	22.34%	98.18%	100.00%	2
92	479	55	2	1	0	-100	14.6480	8.6571	105	53	21.92%	96.36%	100.00%	2

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
93	479	55	2	1	0	-100	14.3710	6.0777	103	52	21.50%	94.55%	100.00%	2
94	479	55	2	1	0	-100	15.0490	7.7087	103	52	21.50%	94.55%	100.00%	2
95	479	55	2	1	0	-100	14.4620	5.3429	105	53	21.92%	96.36%	100.00%	2
96	479	50	2	1	0	-100	13.1410	6.0632	95	48	19.83%	96.00%	100.00%	2
97	479	50	2	1	0	-100	14.3930	6.6947	95	48	19.83%	96.00%	100.00%	2
98	479	50	2	1	0	-100	14.3330	7.8632	95	48	19.83%	96.00%	100.00%	2
99	479	50	2	1	0	-100	13.8440	6.3763	93	47	19.42%	94.00%	100.00%	2
100	479	50	2	1	0	-100	15.3920	7.1031	97	49	20.25%	98.00%	100.00%	2
101	479	45	2	1	0	-100	12.6790	2.7882	85	43	17.75%	95.56%	100.00%	2
102	479	45	2	1	0	-100	11.7010	6.8941	85	43	17.75%	95.56%	100.00%	2
103	479	45	2	1	0	-100	11.7700	7.5517	87	44	18.16%	97.78%	100.00%	2
104	479	45	2	1	0	-100	11.6770	6.3882	85	43	17.75%	95.56%	100.00%	2
105	479	45	2	1	0	-100	11.5410	6.8315	89	45	18.58%	100.00%	100.00%	2
106	479	40	2	1	0	-100	10.3800	7.0000	75	38	15.66%	95.00%	100.00%	2
107	479	40	2	1	0	-100	10.4260	5.7600	75	38	15.66%	95.00%	100.00%	2
108	479	40	2	1	0	-100	10.3310	8.1646	79	40	16.49%	100.00%	100.00%	2
109	479	40	2	1	0	-100	10.3430	6.7662	77	39	16.08%	97.50%	100.00%	2
110	479	40	2	1	0	-100	10.2490	7.1067	75	38	15.66%	95.00%	100.00%	2
111	479	35	2	1	0	-100	9.0460	3.6571	70	35	14.61%	100.00%	100.00%	2
112	479	35	2	1	0	-100	9.0750	9.6716	67	34	13.99%	97.14%	100.00%	2
113	479	35	2	1	0	-100	9.5590	5.8060	67	34	13.99%	97.14%	100.00%	2
114	479	35	2	1	0	-100	9.2360	7.2985	67	34	13.99%	97.14%	100.00%	2
115	479	35	2	1	0	-100	9.2590	6.6957	69	35	14.41%	100.00%	100.00%	2

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
116	479	30	2	1	0	-100	8.2000	5.0333	60	30	12.53%	100.00%	100.00%	2
117	479	30	2	1	0	-100	8.7960	3.2000	60	30	12.53%	100.00%	100.00%	2
118	479	30	2	1	0	-100	8.5010	6.4667	60	30	12.53%	100.00%	100.00%	2
119	479	30	2	1	0	-100	8.5110	8.4737	57	29	11.90%	96.67%	100.00%	2
120	479	30	2	1	0	-100	8.5820	0.4000	60	30	12.53%	100.00%	100.00%	2
121	479	25	2	1	0	-100	7.6480	6.8085	47	24	9.81%	96.00%	100.00%	2
122	479	25	2	1	0	-100	7.8750	4.1600	50	25	10.44%	100.00%	100.00%	2
123	479	25	2	1	0	-100	7.6520	6.5319	47	24	9.81%	96.00%	100.00%	2
124	479	25	2	1	0	-100	6.8510	5.4082	49	25	10.23%	100.00%	100.00%	2
125	479	25	2	1	0	-100	6.8500	7.7143	49	25	10.23%	100.00%	100.00%	2
126	479	20	2	1	0	-100	5.8890	6.4615	39	20	8.14%	100.00%	100.00%	2
127	479	20	2	1	0	-100	5.8230	-3.8378	37	19	7.72%	95.00%	100.00%	2
128	479	20	2	1	0	-100	5.8340	2.0000	39	20	8.14%	100.00%	100.00%	2
129	479	20	2	1	0	-100	5.5930	4.6500	40	20	8.35%	100.00%	100.00%	2
130	479	20	2	1	0	-100	5.7810	11.6154	39	20	8.14%	100.00%	100.00%	2
131	479	15	2	1	0	-100	4.9730	13.0333	30	15	6.26%	100.00%	100.00%	2
132	479	15	2	1	0	-100	4.8630	11.7667	30	15	6.26%	100.00%	100.00%	2
133	479	15	2	1	0	-100	4.6890	4.5667	30	15	6.26%	100.00%	100.00%	2
134	479	15	2	1	0	-100	4.9390	7.5517	29	15	6.05%	100.00%	100.00%	2
135	479	15	2	1	0	-100	4.8740	3.3333	30	15	6.26%	100.00%	100.00%	2
136	479	10	2	1	0	-100	3.8700	10.7500	20	10	4.18%	100.00%	100.00%	2
137	479	10	2	1	0	-100	3.7960	-0.1500	20	10	4.18%	100.00%	100.00%	2
138	479	10	2	1	0	-100	3.8130	2.5500	20	10	4.18%	100.00%	100.00%	2

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	**Number of Riders**	**Number of Drivers**	**Capacity**	**Rider Count**	**Number of Riders Rejecting First Match**	**JSS Limit**	**Computation Time**	**Average JSS of Matched Pairs**	**Number of Matched Riders**	**Number of Matched Drivers**	**Ratio of Matched Riders**	**Ratio of Matched Drivers**	**Ratio of Riders Accepting First Match**	**Case #**
139	479	10	2	1	0	-100	3.8000	14.4500	20	10	4.18%	100.00%	100.00%	2
140	479	10	2	1	0	-100	3.7270	8.1500	20	10	4.18%	100.00%	100.00%	2
141	479	5	2	1	0	-100	2.7550	5.5000	10	5	2.09%	100.00%	100.00%	2
142	479	5	2	1	0	-100	2.8820	8.4000	10	5	2.09%	100.00%	100.00%	2
143	479	5	2	1	0	-100	2.8460	-14.1000	10	5	2.09%	100.00%	100.00%	2
144	479	5	2	1	0	-100	2.7950	21.0000	10	5	2.09%	100.00%	100.00%	2
145	479	5	2	1	0	-100	3.0480	20.0000	10	5	2.09%	100.00%	100.00%	2
146	479	1	2	1	0	-100	2.1790	0.0000	2	1	0.42%	100.00%	100.00%	2
147	479	1	2	1	0	-100	2.1760	3.5000	2	1	0.42%	100.00%	100.00%	2
148	479	1	2	1	0	-100	2.1720	-33.0000	2	1	0.42%	100.00%	100.00%	2
149	479	1	2	1	0	-100	2.1730	32.5000	2	1	0.42%	100.00%	100.00%	2
150	479	1	2	1	0	-100	2.3000	2.5000	2	1	0.42%	100.00%	100.00%	2
151	479	69	1	1	0	-100	14.7540	7.8657	67	67	13.99%	97.10%	100.00%	3
152	479	69	1	1	0	-100	14.8700	7.8657	67	67	13.99%	97.10%	100.00%	3
153	479	69	1	1	0	-100	14.8410	7.8657	67	67	13.99%	97.10%	100.00%	3
154	479	69	1	1	0	-100	14.9250	7.8657	67	67	13.99%	97.10%	100.00%	3
155	479	69	1	1	0	-100	14.8820	7.8657	67	67	13.99%	97.10%	100.00%	3
156	479	65	1	1	0	-100	14.3840	8.8571	63	63	13.15%	96.92%	100.00%	3
157	479	65	1	1	0	-100	14.0810	8.2656	64	64	13.36%	98.46%	100.00%	3
158	479	65	1	1	0	-100	15.1190	6.8413	63	63	13.15%	96.92%	100.00%	3
159	479	65	1	1	0	-100	14.5620	9.3016	63	63	13.15%	96.92%	100.00%	3
160	479	65	1	1	0	-100	15.3260	7.4063	64	64	13.36%	98.46%	100.00%	3
161	479	60	1	1	0	-100	13.1430	5.6000	60	60	12.53%	100.00%	100.00%	3

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
162	479	60	1	1	0	-100	14.2280	6.1667	60	60	12.53%	100.00%	100.00%	3
163	479	60	1	1	0	-100	14.9370	9.1035	58	58	12.11%	96.67%	100.00%	3
164	479	60	1	1	0	-100	14.0400	9.3103	58	58	12.11%	96.67%	100.00%	3
165	479	60	1	1	0	-100	12.8460	7.6949	59	59	12.32%	98.33%	100.00%	3
166	479	55	1	1	0	-100	11.8950	9.9259	54	54	11.27%	98.18%	100.00%	3
167	479	55	1	1	0	-100	11.8120	9.0727	55	55	11.48%	100.00%	100.00%	3
168	479	55	1	1	0	-100	12.0190	7.9245	53	53	11.06%	96.36%	100.00%	3
169	479	55	1	1	0	-100	11.8020	7.2037	54	54	11.27%	98.18%	100.00%	3
170	479	55	1	1	0	-100	11.9720	9.6482	54	54	11.27%	98.18%	100.00%	3
171	479	50	1	1	0	-100	10.7850	7.7400	50	50	10.44%	100.00%	100.00%	3
172	479	50	1	1	0	-100	10.9990	9.7000	50	50	10.44%	100.00%	100.00%	3
173	479	50	1	1	0	-100	10.9100	7.1667	48	48	10.02%	96.00%	100.00%	3
174	479	50	1	1	0	-100	10.8120	4.5800	50	50	10.44%	100.00%	100.00%	3
175	479	50	1	1	0	-100	10.7510	6.4400	50	50	10.44%	100.00%	100.00%	3
176	479	45	1	1	0	-100	9.7880	7.3556	45	45	9.39%	100.00%	100.00%	3
177	479	45	1	1	0	-100	9.9010	7.8000	45	45	9.39%	100.00%	100.00%	3
178	479	45	1	1	0	-100	9.9680	9.8222	45	45	9.39%	100.00%	100.00%	3
179	479	45	1	1	0	-100	9.7030	6.8000	45	45	9.39%	100.00%	100.00%	3
180	479	45	1	1	0	-100	9.8160	8.1778	45	45	9.39%	100.00%	100.00%	3
181	479	40	1	1	0	-100	9.0110	6.0513	39	39	8.14%	97.50%	100.00%	3
182	479	40	1	1	0	-100	8.8420	6.2750	40	40	8.35%	100.00%	100.00%	3
183	479	40	1	1	0	-100	8.8440	9.8000	40	40	8.35%	100.00%	100.00%	3
184	479	40	1	1	0	-100	8.9490	3.6000	40	40	8.35%	100.00%	100.00%	3

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
185	479	40	1	1	0	-100	8.9500	7.7000	40	40	8.35%	100.00%	100.00%	3
186	479	35	1	1	0	-100	7.7610	1.7429	35	35	7.31%	100.00%	100.00%	3
187	479	35	1	1	0	-100	8.0420	5.7714	35	35	7.31%	100.00%	100.00%	3
188	479	35	1	1	0	-100	7.8420	5.0571	35	35	7.31%	100.00%	100.00%	3
189	479	35	1	1	0	-100	7.9390	5.4571	35	35	7.31%	100.00%	100.00%	3
190	479	35	1	1	0	-100	7.9490	6.4000	35	35	7.31%	100.00%	100.00%	3
191	479	30	1	1	0	-100	7.0090	6.8333	30	30	6.26%	100.00%	100.00%	3
192	479	30	1	1	0	-100	6.8840	6.2000	30	30	6.26%	100.00%	100.00%	3
193	479	30	1	1	0	-100	6.8760	3.8000	30	30	6.26%	100.00%	100.00%	3
194	479	30	1	1	0	-100	6.8770	7.6000	30	30	6.26%	100.00%	100.00%	3
195	479	30	1	1	0	-100	7.0270	7.8333	30	30	6.26%	100.00%	100.00%	3
196	479	25	1	1	0	-100	6.1200	8.6000	25	25	5.22%	100.00%	100.00%	3
197	479	25	1	1	0	-100	6.1620	7.7200	25	25	5.22%	100.00%	100.00%	3
198	479	25	1	1	0	-100	6.0270	7.0400	25	25	5.22%	100.00%	100.00%	3
199	479	25	1	1	0	-100	5.8840	11.6000	25	25	5.22%	100.00%	100.00%	3
200	479	25	1	1	0	-100	5.8140	7.5200	25	25	5.22%	100.00%	100.00%	3
201	479	20	1	1	0	-100	5.1550	4.4500	20	20	4.18%	100.00%	100.00%	3
202	479	20	1	1	0	-100	5.3330	7.8500	20	20	4.18%	100.00%	100.00%	3
203	479	20	1	1	0	-100	5.1440	8.4000	20	20	4.18%	100.00%	100.00%	3
204	479	20	1	1	0	-100	5.1640	5.8500	20	20	4.18%	100.00%	100.00%	3
205	479	20	1	1	0	-100	5.1370	1.9500	20	20	4.18%	100.00%	100.00%	3
206	479	15	1	1	0	-100	4.2610	10.6000	15	15	3.13%	100.00%	100.00%	3
207	479	15	1	1	0	-100	4.3010	9.3333	15	15	3.13%	100.00%	100.00%	3

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
208	479	15	1	1	0	-100	4.2790	10.6000	15	15	3.13%	100.00%	100.00%	3
209	479	15	1	1	0	-100	4.2470	6.4000	15	15	3.13%	100.00%	100.00%	3
210	479	15	1	1	0	-100	4.4540	1.0667	15	15	3.13%	100.00%	100.00%	3
211	479	10	1	1	0	-100	3.5130	7.2000	10	10	2.09%	100.00%	100.00%	3
212	479	10	1	1	0	-100	3.5640	5.6000	10	10	2.09%	100.00%	100.00%	3
213	479	10	1	1	0	-100	3.4910	12.2000	10	10	2.09%	100.00%	100.00%	3
214	479	10	1	1	0	-100	3.4870	9.5000	10	10	2.09%	100.00%	100.00%	3
215	479	10	1	1	0	-100	3.4970	11.1000	10	10	2.09%	100.00%	100.00%	3
216	479	5	1	1	0	-100	2.9330	17.4000	5	5	1.04%	100.00%	100.00%	3
217	479	5	1	1	0	-100	2.8360	18.8000	5	5	1.04%	100.00%	100.00%	3
218	479	5	1	1	0	-100	2.7010	-1.6000	5	5	1.04%	100.00%	100.00%	3
219	479	5	1	1	0	-100	2.7020	0.2000	5	5	1.04%	100.00%	100.00%	3
220	479	5	1	1	0	-100	2.7820	4.6000	5	5	1.04%	100.00%	100.00%	3
221	479	1	1	1	0	-100	2.1690	10.0000	1	1	0.21%	100.00%	100.00%	3
222	479	1	1	1	0	-100	2.1700	0.0000	1	1	0.21%	100.00%	100.00%	3
223	479	1	1	1	0	-100	2.4050	10.0000	1	1	0.21%	100.00%	100.00%	3
224	479	1	1	1	0	-100	2.5580	30.0000	1	1	0.21%	100.00%	100.00%	3
225	479	1	1	1	0	-100	2.1660	2.0000	1	1	0.21%	100.00%	100.00%	3
226	450	69	3	1	0	-100	21.9600	8.2670	191	64	42.44%	92.75%	100.00%	4
227	450	69	3	1	0	-100	21.9920	8.0209	191	64	42.44%	92.75%	100.00%	4
228	450	69	3	1	0	-100	21.9510	9.0262	191	64	42.44%	92.75%	100.00%	4
229	450	69	3	1	0	-100	21.8250	8.8011	191	64	42.44%	92.75%	100.00%	4
230	450	69	3	1	0	-100	21.7760	8.2827	191	64	42.44%	92.75%	100.00%	4

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
231	400	69	3	1	0	-100	20.8280	9.1474	190	64	47.50%	92.75%	100.00%	4
232	400	69	3	1	0	-100	21.0580	7.7790	190	64	47.50%	92.75%	100.00%	4
233	400	69	3	1	0	-100	20.9480	9.3895	190	64	47.50%	92.75%	100.00%	4
234	400	69	3	1	0	-100	20.3260	8.6421	190	64	47.50%	92.75%	100.00%	4
235	400	69	3	1	0	-100	20.7900	9.5079	189	64	47.25%	92.75%	100.00%	4
236	350	69	3	1	0	-100	19.3230	8.5479	188	64	53.71%	92.75%	100.00%	4
237	350	69	3	1	0	-100	19.3730	8.6667	189	64	54.00%	92.75%	100.00%	4
238	350	69	3	1	0	-100	19.3430	9.0267	187	63	53.43%	91.30%	100.00%	4
239	350	69	3	1	0	-100	19.0010	8.1543	188	64	53.71%	92.75%	100.00%	4
240	350	69	3	1	0	-100	19.5660	8.1170	188	64	53.71%	92.75%	100.00%	4
241	300	69	3	1	0	-100	18.0780	9.1405	185	63	61.67%	91.30%	100.00%	4
242	300	69	3	1	0	-100	18.5720	10.2120	184	63	61.33%	91.30%	100.00%	4
243	300	69	3	1	0	-100	18.2050	9.7796	186	63	62.00%	91.30%	100.00%	4
244	300	69	3	1	0	-100	18.2150	9.2842	183	63	61.00%	91.30%	100.00%	4
245	300	69	3	1	0	-100	18.3500	9.1862	188	63	62.67%	91.30%	100.00%	4
246	250	69	3	1	0	-100	17.2790	11.8802	167	58	66.80%	84.06%	100.00%	4
247	250	69	3	1	0	-100	17.0290	11.2429	177	62	70.80%	89.86%	100.00%	4
248	250	69	3	1	0	-100	16.8360	11.2428	173	60	69.20%	86.96%	100.00%	4
249	250	69	3	1	0	-100	16.5130	10.0824	170	59	68.00%	85.51%	100.00%	4
250	250	69	3	1	0	-100	16.9530	10.8187	171	61	68.40%	88.41%	100.00%	4
251	200	69	3	1	0	-100	14.4100	10.9932	147	53	73.50%	76.81%	100.00%	4
252	200	69	3	1	0	-100	15.1930	11.2089	158	56	79.00%	81.16%	100.00%	4
253	200	69	3	1	0	-100	14.4040	13.3862	145	53	72.50%	76.81%	100.00%	4

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
254	200	69	3	1	0	-100	14.4330	11.4713	157	57	78.50%	82.61%	100.00%	4
255	200	69	3	1	0	-100	14.5010	12.3113	151	53	75.50%	76.81%	100.00%	4
256	150	69	3	1	0	-100	11.0490	14.0840	131	49	87.33%	71.01%	100.00%	4
257	150	69	3	1	0	-100	11.5460	11.3566	129	50	86.00%	72.46%	100.00%	4
258	150	69	3	1	0	-100	11.5550	11.9225	129	50	86.00%	72.46%	100.00%	4
259	150	69	3	1	0	-100	11.6560	13.8240	125	47	83.33%	68.12%	100.00%	4
260	150	69	3	1	0	-100	11.6080	12.1860	129	47	86.00%	68.12%	100.00%	4
261	100	69	3	1	0	-100	9.1530	13.3478	92	38	92.00%	55.07%	100.00%	4
262	100	69	3	1	0	-100	8.3800	12.9158	95	41	95.00%	59.42%	100.00%	4
263	100	69	3	1	0	-100	10.7130	15.0000	91	39	91.00%	56.52%	100.00%	4
264	100	69	3	1	0	-100	9.2490	16.0851	94	38	94.00%	55.07%	100.00%	4
265	100	69	3	1	0	-100	8.7390	15.6250	96	39	96.00%	56.52%	100.00%	4
266	50	69	3	1	0	-100	4.4130	20.6122	49	21	98.00%	30.43%	100.00%	4
267	50	69	3	1	0	-100	4.7740	15.7400	50	23	100.00%	33.33%	100.00%	4
268	50	69	3	1	0	-100	4.1040	13.4444	45	20	90.00%	28.99%	100.00%	4
269	50	69	3	1	0	-100	4.6920	22.8542	48	22	96.00%	31.88%	100.00%	4
270	50	69	3	1	0	-100	4.9860	17.8571	49	22	98.00%	31.88%	100.00%	4
271	1	69	3	1	0	-100	0.0850	30.0000	1	1	100.00%	1.45%	100.00%	4
272	1	69	3	1	0	-100	0.1690	45.0000	1	1	100.00%	1.45%	100.00%	4
273	1	69	3	1	0	-100	0.1890	65.0000	1	1	100.00%	1.45%	100.00%	4
274	1	69	3	1	0	-100	0.0680	21.0000	1	1	100.00%	1.45%	100.00%	4
275	1	69	3	1	0	-100	0.0880	0.0000	1	1	100.00%	1.45%	100.00%	4
276	450	69	2	1	0	-100	17.7350	8.5078	128	64	28.44%	92.75%	100.00%	5

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
277	450	69	2	1	0	-100	17.3830	7.9147	129	65	28.67%	94.20%	100.00%	5
278	450	69	2	1	0	-100	17.1780	7.6094	128	65	28.44%	94.20%	100.00%	5
279	450	69	2	1	0	-100	17.2770	8.1563	128	65	28.44%	94.20%	100.00%	5
280	450	69	2	1	0	-100	17.1420	8.4961	129	65	28.67%	94.20%	100.00%	5
281	400	69	2	1	0	-100	16.3300	8.8217	129	65	32.25%	94.20%	100.00%	5
282	400	69	2	1	0	-100	15.9190	8.4341	129	65	32.25%	94.20%	100.00%	5
283	400	69	2	1	0	-100	16.2130	8.8062	129	65	32.25%	94.20%	100.00%	5
284	400	69	2	1	0	-100	16.1450	8.9531	128	64	32.00%	92.75%	100.00%	5
285	400	69	2	1	0	-100	16.6190	8.4109	129	65	32.25%	94.20%	100.00%	5
286	350	69	2	1	0	-100	15.1680	7.6328	128	65	36.57%	94.20%	100.00%	5
287	350	69	2	1	0	-100	15.7430	8.7266	128	65	36.57%	94.20%	100.00%	5
288	350	69	2	1	0	-100	15.7300	8.3047	128	65	36.57%	94.20%	100.00%	5
289	350	69	2	1	0	-100	15.3910	8.3594	128	64	36.57%	92.75%	100.00%	5
290	350	69	2	1	0	-100	15.8440	7.7222	126	64	36.00%	92.75%	100.00%	5
291	300	69	2	1	0	-100	14.4830	7.8583	127	64	42.33%	92.75%	100.00%	5
292	300	69	2	1	0	-100	14.4660	8.2813	128	64	42.67%	92.75%	100.00%	5
293	300	69	2	1	0	-100	14.7830	9.2598	127	64	42.33%	92.75%	100.00%	5
294	300	69	2	1	0	-100	14.5340	7.5276	127	64	42.33%	92.75%	100.00%	5
295	300	69	2	1	0	-100	14.5330	9.2835	127	64	42.33%	92.75%	100.00%	5
296	250	69	2	1	0	-100	13.4040	10.0635	126	64	50.40%	92.75%	100.00%	5
297	250	69	2	1	0	-100	13.2020	9.4206	126	64	50.40%	92.75%	100.00%	5
298	250	69	2	1	0	-100	12.9400	9.5276	127	64	50.80%	92.75%	100.00%	5
299	250	69	2	1	0	-100	13.3010	8.9921	127	64	50.80%	92.75%	100.00%	5

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
300	250	69	2	1	0	-100	13.1980	9.4206	126	64	50.40%	92.75%	100.00%	5
301	200	69	2	1	0	-100	12.0140	7.2581	124	62	62.00%	89.86%	100.00%	5
302	200	69	2	1	0	-100	12.1000	9.3360	125	63	62.50%	91.30%	100.00%	5
303	200	69	2	1	0	-100	11.9460	10.0439	114	58	57.00%	84.06%	100.00%	5
304	200	69	2	1	0	-100	12.1120	9.9508	122	62	61.00%	89.86%	100.00%	5
305	200	69	2	1	0	-100	12.6720	9.3600	125	63	62.50%	91.30%	100.00%	5
306	150	69	2	1	0	-100	10.5390	12.3529	102	54	68.00%	78.26%	100.00%	5
307	150	69	2	1	0	-100	10.3440	9.6132	106	54	70.67%	78.26%	100.00%	5
308	150	69	2	1	0	-100	10.6530	9.5913	115	59	76.67%	85.51%	100.00%	5
309	150	69	2	1	0	-100	10.8130	9.0357	112	59	74.67%	85.51%	100.00%	5
310	150	69	2	1	0	-100	10.3110	10.7383	107	55	71.33%	79.71%	100.00%	5
311	100	69	2	1	0	-100	8.0250	11.7841	88	48	88.00%	69.57%	100.00%	5
312	100	69	2	1	0	-100	7.8760	15.7000	80	43	80.00%	62.32%	100.00%	5
313	100	69	2	1	0	-100	7.5740	12.8861	79	43	79.00%	62.32%	100.00%	5
314	100	69	2	1	0	-100	8.4900	10.7872	94	51	94.00%	73.91%	100.00%	5
315	100	69	2	1	0	-100	8.4800	11.9775	89	49	89.00%	71.01%	100.00%	5
316	50	69	2	1	0	-100	4.3460	16.3913	46	29	92.00%	42.03%	100.00%	5
317	50	69	2	1	0	-100	4.1560	13.0732	41	26	82.00%	37.68%	100.00%	5
318	50	69	2	1	0	-100	4.4920	15.7234	47	28	94.00%	40.58%	100.00%	5
319	50	69	2	1	0	-100	4.4040	15.5319	47	27	94.00%	39.13%	100.00%	5
320	50	69	2	1	0	-100	4.1650	17.1458	48	28	96.00%	40.58%	100.00%	5
321	1	69	2	1	0	-100	0.1790	40.0000	1	1	100.00%	1.45%	100.00%	5
322	1	69	2	1	0	-100	0.0570	34.0000	1	1	100.00%	0.00%	100.00%	5

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
323	1	69	2	1	0	-100	0.0920	0.0000	1	1	100.00%	1.45%	100.00%	5
324	1	69	2	1	0	-100	0.0880	0.0000	1	1	100.00%	1.45%	100.00%	5
325	1	69	2	1	0	-100	0.0880	30.0000	1	1	100.00%	1.45%	100.00%	5
326	450	69	1	1	0	-100	14.1530	8.1194	67	67	14.89%	97.10%	100.00%	6
327	450	69	1	1	0	-100	14.0540	8.0746	67	67	14.89%	97.10%	100.00%	6
328	450	69	1	1	0	-100	14.2890	7.6212	66	66	14.67%	95.65%	100.00%	6
329	450	69	1	1	0	-100	13.8980	5.8508	67	67	14.89%	97.10%	100.00%	6
330	450	69	1	1	0	-100	14.2600	6.6866	67	67	14.89%	97.10%	100.00%	6
331	400	69	1	1	0	-100	13.0380	6.1940	67	67	16.75%	97.10%	100.00%	6
332	400	69	1	1	0	-100	12.8860	7.2424	66	66	16.50%	95.65%	100.00%	6
333	400	69	1	1	0	-100	12.8020	6.2576	66	66	16.50%	95.65%	100.00%	6
334	400	69	1	1	0	-100	15.3820	7.0597	67	67	16.75%	97.10%	100.00%	6
335	400	69	1	1	0	-100	15.0920	6.5385	65	65	16.25%	94.20%	100.00%	6
336	350	69	1	1	0	-100	12.3980	6.1250	64	64	18.29%	92.75%	100.00%	6
337	350	69	1	1	0	-100	12.2170	7.6769	65	65	18.57%	94.20%	100.00%	6
338	350	69	1	1	0	-100	13.0650	5.2388	67	67	19.14%	97.10%	100.00%	6
339	350	69	1	1	0	-100	13.7210	7.9242	66	66	18.86%	95.65%	100.00%	6
340	350	69	1	1	0	-100	12.1100	6.7576	66	66	18.86%	95.65%	100.00%	6
341	300	69	1	1	0	-100	10.4170	4.8000	65	65	21.67%	94.20%	100.00%	6
342	300	69	1	1	0	-100	10.3620	6.3030	66	66	22.00%	95.65%	100.00%	6
343	300	69	1	1	0	-100	10.4340	9.6094	64	64	21.33%	92.75%	100.00%	6
344	300	69	1	1	0	-100	10.7850	5.7463	67	67	22.33%	97.10%	100.00%	6
345	300	69	1	1	0	-100	10.2320	5.1385	65	65	21.67%	94.20%	100.00%	6

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
346	250	69	1	1	0	-100	9.3850	6.9231	65	65	26.00%	94.20%	100.00%	6
347	250	69	1	1	0	-100	9.3660	8.8333	66	66	26.40%	95.65%	100.00%	6
348	250	69	1	1	0	-100	9.3200	8.2727	66	66	26.40%	95.65%	100.00%	6
349	250	69	1	1	0	-100	9.4680	6.8125	64	64	25.60%	92.75%	100.00%	6
350	250	69	1	1	0	-100	9.0840	4.7121	66	66	26.40%	95.65%	100.00%	6
351	200	69	1	1	0	-100	8.3470	7.5938	64	64	32.00%	92.75%	100.00%	6
352	200	69	1	1	0	-100	8.1660	8.4844	64	64	32.00%	92.75%	100.00%	6
353	200	69	1	1	0	-100	7.8970	7.2923	65	65	32.50%	94.20%	100.00%	6
354	200	69	1	1	0	-100	7.9200	7.8254	63	63	31.50%	91.30%	100.00%	6
355	200	69	1	1	0	-100	8.0090	6.0156	64	64	32.00%	92.75%	100.00%	6
356	150	69	1	1	0	-100	7.2910	8.2188	64	64	42.67%	92.75%	100.00%	6
357	150	69	1	1	0	-100	7.1510	8.8254	63	63	42.00%	91.30%	100.00%	6
358	150	69	1	1	0	-100	6.7410	6.3438	64	64	42.67%	92.75%	100.00%	6
359	150	69	1	1	0	-100	7.1490	6.7344	64	64	42.67%	92.75%	100.00%	6
360	150	69	1	1	0	-100	7.3490	6.5238	63	63	42.00%	91.30%	100.00%	6
361	100	69	1	1	0	-100	5.8550	7.4677	62	62	62.00%	89.86%	100.00%	6
362	100	69	1	1	0	-100	6.0050	8.8889	63	63	63.00%	91.30%	100.00%	6
363	100	69	1	1	0	-100	6.0180	8.7742	62	62	62.00%	89.86%	100.00%	6
364	100	69	1	1	0	-100	5.9810	6.6721	61	61	61.00%	88.41%	100.00%	6
365	100	69	1	1	0	-100	5.8770	7.6129	62	62	62.00%	89.86%	100.00%	6
366	50	69	1	1	0	-100	3.4340	12.2381	42	42	84.00%	60.87%	100.00%	6
367	50	69	1	1	0	-100	4.0420	12.0930	43	43	86.00%	62.32%	100.00%	6
368	50	69	1	1	0	-100	3.8220	9.2750	40	40	80.00%	57.97%	100.00%	6

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
369	50	69	1	1	0	-100	3.7310	9.0270	37	37	74.00%	53.62%	100.00%	6
370	50	69	1	1	0	-100	3.9600	10.0233	43	43	86.00%	62.32%	100.00%	6
371	1	69	1	1	0	-100	0.1030	25.0000	1	1	100.00%	1.45%	100.00%	6
372	1	69	1	1	0	-100	0.0770	10.0000	1	1	100.00%	1.45%	100.00%	6
373	1	69	1	1	0	-100	0.1770	55.0000	1	1	100.00%	1.45%	100.00%	6
374	1	69	1	1	0	-100	0.0840	0.0000	1	1	100.00%	1.45%	100.00%	6
375	1	69	1	1	0	-100	0.0860	0.0000	1	1	100.00%	1.45%	100.00%	6
376	479	69	3	2	0	-100	14.7960	7.8657	67	67	13.99%	97.10%	100.00%	7
377	479	69	3	2	0	-100	14.8030	7.8657	67	67	13.99%	97.10%	100.00%	7
378	479	69	3	2	0	-100	14.8490	7.8657	67	67	13.99%	97.10%	100.00%	7
379	479	69	3	2	0	-100	14.8490	7.8657	67	67	13.99%	97.10%	100.00%	7
380	479	69	3	2	0	-100	14.8960	7.8657	67	67	13.99%	97.10%	100.00%	7
381	479	65	3	2	0	-100	13.9660	7.0318	63	63	13.15%	96.92%	100.00%	7
382	479	65	3	2	0	-100	13.9560	7.8413	63	63	13.15%	96.92%	100.00%	7
383	479	65	3	2	0	-100	14.0360	7.8906	64	64	13.36%	98.46%	100.00%	7
384	479	65	3	2	0	-100	13.8060	7.6719	64	64	13.36%	98.46%	100.00%	7
385	479	65	3	2	0	-100	14.1300	8.3651	63	63	13.15%	96.92%	100.00%	7
386	479	60	3	2	0	-100	12.9300	7.0690	58	58	12.11%	96.67%	100.00%	7
387	479	60	3	2	0	-100	13.1460	6.3104	58	58	12.11%	96.67%	100.00%	7
388	479	60	3	2	0	-100	12.9960	8.8644	59	59	12.32%	98.33%	100.00%	7
389	479	60	3	2	0	-100	13.3350	7.7241	58	58	12.11%	96.67%	100.00%	7
390	479	60	3	2	0	-100	13.0910	7.1525	59	59	12.32%	98.33%	100.00%	7
391	479	55	3	2	0	-100	12.0460	7.0741	54	54	11.27%	98.18%	100.00%	7

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
392	479	55	3	2	0	-100	11.9470	6.6182	55	55	11.48%	100.00%	100.00%	7
393	479	55	3	2	0	-100	13.1470	8.7925	53	53	11.06%	96.36%	100.00%	7
394	479	55	3	2	0	-100	13.1160	6.0377	53	53	11.06%	96.36%	100.00%	7
395	479	55	3	2	0	-100	12.4520	7.2593	54	54	11.27%	98.18%	100.00%	7
396	479	50	3	2	0	-100	12.0560	6.1000	50	50	10.44%	100.00%	100.00%	7
397	479	50	3	2	0	-100	12.2900	7.7959	49	49	10.23%	98.00%	100.00%	7
398	479	50	3	2	0	-100	11.3720	5.3400	50	50	10.44%	100.00%	100.00%	7
399	479	50	3	2	0	-100	12.4870	7.4400	50	50	10.44%	100.00%	100.00%	7
400	479	50	3	2	0	-100	11.7710	6.6600	50	50	10.44%	100.00%	100.00%	7
401	479	45	3	2	0	-100	9.7230	4.6000	45	45	9.39%	100.00%	100.00%	7
402	479	45	3	2	0	-100	9.9350	7.2500	44	44	9.19%	97.78%	100.00%	7
403	479	45	3	2	0	-100	9.8720	4.7556	45	45	9.39%	100.00%	100.00%	7
404	479	45	3	2	0	-100	9.8710	6.2667	45	45	9.39%	100.00%	100.00%	7
405	479	45	3	2	0	-100	9.7990	5.4000	45	45	9.39%	100.00%	100.00%	7
406	479	40	3	2	0	-100	8.8360	5.9250	40	40	8.35%	100.00%	100.00%	7
407	479	40	3	2	0	-100	8.9120	6.0250	40	40	8.35%	100.00%	100.00%	7
408	479	40	3	2	0	-100	8.9820	7.4750	40	40	8.35%	100.00%	100.00%	7
409	479	40	3	2	0	-100	9.1940	8.6053	38	38	7.93%	95.00%	100.00%	7
410	479	40	3	2	0	-100	9.1750	9.2051	39	39	8.14%	97.50%	100.00%	7
411	479	35	3	2	0	-100	8.0320	7.1429	35	35	7.31%	100.00%	100.00%	7
412	479	35	3	2	0	-100	7.8660	8.7429	35	35	7.31%	100.00%	100.00%	7
413	479	35	3	2	0	-100	7.9890	6.2571	35	35	7.31%	100.00%	100.00%	7
414	479	35	3	2	0	-100	8.0240	4.8571	35	35	7.31%	100.00%	100.00%	7

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
415	479	35	3	2	0	-100	8.2350	5.3714	35	35	7.31%	100.00%	100.00%	7
416	479	30	3	2	0	-100	7.9430	7.9000	30	30	6.26%	100.00%	100.00%	7
417	479	30	3	2	0	-100	7.5630	8.4000	30	30	6.26%	100.00%	100.00%	7
418	479	30	3	2	0	-100	7.3660	9.0667	30	30	6.26%	100.00%	100.00%	7
419	479	30	3	2	0	-100	7.4140	7.0000	30	30	6.26%	100.00%	100.00%	7
420	479	30	3	2	0	-100	7.4570	9.1000	30	30	6.26%	100.00%	100.00%	7
421	479	25	3	2	0	-100	6.2670	14.4000	25	25	5.22%	100.00%	100.00%	7
422	479	25	3	2	0	-100	6.3320	7.8000	25	25	5.22%	100.00%	100.00%	7
423	479	25	3	2	0	-100	6.4980	7.8400	25	25	5.22%	100.00%	100.00%	7
424	479	25	3	2	0	-100	6.3770	3.8400	25	25	5.22%	100.00%	100.00%	7
425	479	25	3	2	0	-100	6.5470	3.2400	25	25	5.22%	100.00%	100.00%	7
426	479	20	3	2	0	-100	5.7850	5.9500	20	20	4.18%	100.00%	100.00%	7
427	479	20	3	2	0	-100	5.7650	6.5000	20	20	4.18%	100.00%	100.00%	7
428	479	20	3	2	0	-100	5.6930	12.5000	20	20	4.18%	100.00%	100.00%	7
429	479	20	3	2	0	-100	5.6330	2.0000	20	20	4.18%	100.00%	100.00%	7
430	479	20	3	2	0	-100	5.4630	1.8000	20	20	4.18%	100.00%	100.00%	7
431	479	15	3	2	0	-100	4.4240	3.4000	15	15	3.13%	100.00%	100.00%	7
432	479	15	3	2	0	-100	4.4380	3.8667	15	15	3.13%	100.00%	100.00%	7
433	479	15	3	2	0	-100	4.3210	3.8000	15	15	3.13%	100.00%	100.00%	7
434	479	15	3	2	0	-100	4.2560	5.2000	15	15	3.13%	100.00%	100.00%	7
435	479	15	3	2	0	-100	4.3720	14.0667	15	15	3.13%	100.00%	100.00%	7
436	479	10	3	2	0	-100	3.4960	4.3000	10	10	2.09%	100.00%	100.00%	7
437	479	10	3	2	0	-100	3.5580	6.6000	10	10	2.09%	100.00%	100.00%	7

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
438	479	10	3	2	0	-100	3.7420	10.5000	10	10	2.09%	100.00%	100.00%	7
439	479	10	3	2	0	-100	3.4180	0.7000	10	10	2.09%	100.00%	100.00%	7
440	479	10	3	2	0	-100	3.4460	11.0000	10	10	2.09%	100.00%	100.00%	7
441	479	5	3	2	0	-100	3.1060	7.6000	5	5	1.04%	100.00%	100.00%	7
442	479	5	3	2	0	-100	2.9580	2.0000	5	5	1.04%	100.00%	100.00%	7
443	479	5	3	2	0	-100	2.7070	12.2000	5	5	1.04%	100.00%	100.00%	7
444	479	5	3	2	0	-100	2.7840	4.4000	5	5	1.04%	100.00%	100.00%	7
445	479	5	3	2	0	-100	2.7520	5.4000	5	5	1.04%	100.00%	100.00%	7
446	479	1	3	2	0	-100	2.1650	25.0000	1	1	0.21%	100.00%	100.00%	7
447	479	1	3	2	0	-100	2.1800	10.0000	1	1	0.21%	100.00%	100.00%	7
448	479	1	3	2	0	-100	2.1750	-5.0000	1	1	0.21%	100.00%	100.00%	7
449	479	1	3	2	0	-100	2.2530	-20.0000	1	1	0.21%	100.00%	100.00%	7
450	479	1	3	2	0	-100	2.0710	2.0000	1	1	0.21%	100.00%	100.00%	7
451	479	69	3	2	0	-100	14.7980	7.8657	67	67	13.99%	97.10%	100.00%	8
452	479	69	3	2	0	-100	14.9710	7.8657	67	67	13.99%	97.10%	100.00%	8
453	479	69	3	2	0	-100	14.8170	7.8657	67	67	13.99%	97.10%	100.00%	8
454	479	69	3	2	0	-100	14.8130	7.8657	67	67	13.99%	97.10%	100.00%	8
455	479	69	3	2	0	-100	14.7620	7.8657	67	67	13.99%	97.10%	100.00%	8
456	450	69	3	2	0	-100	14.1550	5.7424	66	66	14.67%	95.65%	100.00%	8
457	450	69	3	2	0	-100	14.1660	7.6667	66	66	14.67%	95.65%	100.00%	8
458	450	69	3	2	0	-100	14.0230	5.5522	67	67	14.89%	97.10%	100.00%	8
459	450	69	3	2	0	-100	14.1840	8.5672	67	67	14.89%	97.10%	100.00%	8
460	450	69	3	2	0	-100	14.1410	7.0597	67	67	14.89%	97.10%	100.00%	8

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
461	400	69	3	2	0	-100	13.2040	7.9105	67	67	16.75%	97.10%	100.00%	8
462	400	69	3	2	0	-100	13.8880	5.5455	66	66	16.50%	95.65%	100.00%	8
463	400	69	3	2	0	-100	12.9930	6.5075	67	67	16.75%	97.10%	100.00%	8
464	400	69	3	2	0	-100	13.4020	6.2424	66	66	16.50%	95.65%	100.00%	8
465	400	69	3	2	0	-100	12.6560	8.1406	64	64	16.00%	92.75%	100.00%	8
466	350	69	3	2	0	-100	12.8500	8.0769	65	65	18.57%	94.20%	100.00%	8
467	350	69	3	2	0	-100	11.6190	6.6061	66	66	18.86%	95.65%	100.00%	8
468	350	69	3	2	0	-100	11.6660	7.6406	64	64	18.29%	92.75%	100.00%	8
469	350	69	3	2	0	-100	11.6090	6.5152	66	66	18.86%	95.65%	100.00%	8
470	350	69	3	2	0	-100	11.5610	8.2000	65	65	18.57%	94.20%	100.00%	8
471	300	69	3	2	0	-100	10.2460	5.5455	66	66	22.00%	95.65%	100.00%	8
472	300	69	3	2	0	-100	10.0120	6.8906	64	64	21.33%	92.75%	100.00%	8
473	300	69	3	2	0	-100	10.2720	6.9539	65	65	21.67%	94.20%	100.00%	8
474	300	69	3	2	0	-100	10.3900	6.6769	65	65	21.67%	94.20%	100.00%	8
475	300	69	3	2	0	-100	10.3770	6.6923	65	65	21.67%	94.20%	100.00%	8
476	250	69	3	2	0	-100	9.1580	9.0938	64	64	25.60%	92.75%	100.00%	8
477	250	69	3	2	0	-100	9.2400	5.5556	63	63	25.20%	91.30%	100.00%	8
478	250	69	3	2	0	-100	8.6880	5.7656	64	64	25.60%	92.75%	100.00%	8
479	250	69	3	2	0	-100	9.3650	6.0635	63	63	25.20%	91.30%	100.00%	8
480	250	69	3	2	0	-100	9.0770	7.6615	65	65	26.00%	94.20%	100.00%	8
481	200	69	3	2	0	-100	7.8090	10.9516	62	62	31.00%	89.86%	100.00%	8
482	200	69	3	2	0	-100	7.8590	7.7344	64	64	32.00%	92.75%	100.00%	8
483	200	69	3	2	0	-100	8.0190	8.2857	63	63	31.50%	91.30%	100.00%	8

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
484	200	69	3	2	0	-100	7.7630	8.0923	65	65	32.50%	94.20%	100.00%	8
485	200	69	3	2	0	-100	8.0700	9.4923	65	65	32.50%	94.20%	100.00%	8
486	150	69	3	2	0	-100	6.8580	7.7424	66	66	44.00%	95.65%	100.00%	8
487	150	69	3	2	0	-100	6.8860	8.3692	65	65	43.33%	94.20%	100.00%	8
488	150	69	3	2	0	-100	6.9920	10.3968	63	63	42.00%	91.30%	100.00%	8
489	150	69	3	2	0	-100	7.2000	6.1846	65	65	43.33%	94.20%	100.00%	8
490	150	69	3	2	0	-100	7.4600	10.3333	63	63	42.00%	91.30%	100.00%	8
491	100	69	3	2	0	-100	5.6600	10.0645	62	62	62.00%	89.86%	100.00%	8
492	100	69	3	2	0	-100	5.7700	10.1905	63	63	63.00%	91.30%	100.00%	8
493	100	69	3	2	0	-100	5.9150	9.0164	61	61	61.00%	88.41%	100.00%	8
494	100	69	3	2	0	-100	6.7670	8.4906	53	53	53.00%	76.81%	100.00%	8
495	100	69	3	2	0	-100	6.2620	9.8500	60	60	60.00%	86.96%	100.00%	8
496	50	69	3	2	0	-100	3.6530	10.8537	41	41	82.00%	59.42%	100.00%	8
497	50	69	3	2	0	-100	3.5030	11.5385	39	39	78.00%	56.52%	100.00%	8
498	50	69	3	2	0	-100	3.8900	10.8500	40	40	80.00%	57.97%	100.00%	8
499	50	69	3	2	0	-100	3.4380	9.1081	37	37	74.00%	53.62%	100.00%	8
500	50	69	3	2	0	-100	3.7700	12.9524	42	42	84.00%	60.87%	100.00%	8
501	1	69	3	2	0	-100	0.1720	15.0000	1	1	100.00%	1.45%	100.00%	8
502	1	69	3	2	0	-100	0.1910	30.0000	1	1	100.00%	1.45%	100.00%	8
503	1	69	3	2	0	-100	0.1870	0.0000	1	1	100.00%	1.45%	100.00%	8
504	1	69	3	2	0	-100	0.2020	25.0000	1	1	100.00%	1.45%	100.00%	8
505	1	69	3	2	0	-100	0.0780	15.0000	1	1	100.00%	1.45%	100.00%	8
506	479	69	3	1	0	-100	22.3780	8.2513	191	64	39.87%	92.75%	100.00%	9

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
507	479	69	3	1	0	-100	21.4350	8.2513	191	64	39.87%	92.75%	100.00%	9
508	479	69	3	1	0	-100	21.6880	8.2513	191	64	39.87%	92.75%	100.00%	9
509	479	69	3	1	0	-100	21.4870	8.2513	191	64	39.87%	92.75%	100.00%	9
510	479	69	3	1	0	-100	21.4120	8.2513	191	64	39.87%	92.75%	100.00%	9
511	479	69	3	1	48	-100	21.4720	8.0052	191	64	39.87%	92.75%	89.98%	9
512	479	69	3	1	48	-100	21.3430	7.8115	191	65	39.87%	94.20%	89.98%	9
513	479	69	3	1	48	-100	21.4730	8.1309	191	64	39.87%	92.75%	89.98%	9
514	479	69	3	1	48	-100	21.4990	8.0209	191	64	39.87%	92.75%	89.98%	9
515	479	69	3	1	48	-100	21.3400	7.4974	191	65	39.87%	94.20%	89.98%	9
516	479	69	3	1	96	-100	21.6120	7.5079	191	66	39.87%	95.65%	79.96%	9
517	479	69	3	1	96	-100	21.6060	7.7330	191	64	39.87%	92.75%	79.96%	9
518	479	69	3	1	96	-100	21.5580	7.4136	191	65	39.87%	94.20%	79.96%	9
519	479	69	3	1	96	-100	21.6680	8.1675	191	64	39.87%	92.75%	79.96%	9
520	479	69	3	1	96	-100	21.6650	7.3665	191	65	39.87%	94.20%	79.96%	9
521	479	69	3	1	144	-100	21.6410	7.8796	191	64	39.87%	92.75%	69.94%	9
522	479	69	3	1	144	-100	21.7970	7.9476	191	65	39.87%	94.20%	69.94%	9
523	479	69	3	1	144	-100	21.7410	7.5812	191	66	39.87%	95.65%	69.94%	9
524	479	69	3	1	144	-100	21.6210	7.1885	191	65	39.87%	94.20%	69.94%	9
525	479	69	3	1	144	-100	21.7950	7.5916	191	65	39.87%	94.20%	69.94%	9
526	479	69	3	1	192	-100	21.8210	7.2737	190	65	39.67%	94.20%	59.92%	9
527	479	69	3	1	192	-100	21.9960	7.2461	191	67	39.87%	97.10%	59.92%	9
528	479	69	3	1	192	-100	21.8520	7.4136	191	66	39.87%	95.65%	59.92%	9
529	479	69	3	1	192	-100	21.7130	6.7853	191	64	39.87%	92.75%	59.92%	9

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
530	479	69	3	1	192	-100	21.7490	6.8534	191	66	39.87%	95.65%	59.92%	9
531	479	69	3	1	240	-100	21.7830	6.2880	191	66	39.87%	95.65%	49.90%	9
532	479	69	3	1	240	-100	22.1550	6.3508	191	65	39.87%	94.20%	49.90%	9
533	479	69	3	1	240	-100	22.1640	6.9948	191	65	39.87%	94.20%	49.90%	9
534	479	69	3	1	240	-100	22.0570	7.4712	191	65	39.87%	94.20%	49.90%	9
535	479	69	3	1	240	-100	22.0290	6.9372	191	66	39.87%	95.65%	49.90%	9
536	479	69	3	1	288	-100	22.3350	6.3158	190	66	39.67%	95.65%	39.87%	9
537	479	69	3	1	288	-100	22.1540	7.4084	191	65	39.87%	94.20%	39.87%	9
538	479	69	3	1	288	-100	21.7870	6.4398	191	67	39.87%	97.10%	39.87%	9
539	479	69	3	1	288	-100	21.9750	6.7644	191	65	39.87%	94.20%	39.87%	9
540	479	69	3	1	288	-100	21.9150	6.5497	191	64	39.87%	92.75%	39.87%	9
541	479	69	3	1	336	-100	21.9830	6.9841	189	65	39.46%	94.20%	29.85%	9
542	479	69	3	1	336	-100	22.0730	6.4922	191	66	39.87%	95.65%	29.85%	9
543	479	69	3	1	336	-100	22.0710	6.3122	189	64	39.46%	92.75%	29.85%	9
544	479	69	3	1	336	-100	22.2850	6.7539	191	65	39.87%	94.20%	29.85%	9
545	479	69	3	1	336	-100	22.2930	6.5445	191	67	39.87%	97.10%	29.85%	9
546	479	69	3	1	384	-100	22.1560	5.7579	190	66	39.67%	95.65%	19.83%	9
547	479	69	3	1	384	-100	22.1100	5.4021	189	64	39.46%	92.75%	19.83%	9
548	479	69	3	1	384	-100	22.2110	6.2421	190	66	39.67%	95.65%	19.83%	9
549	479	69	3	1	384	-100	22.2450	5.9632	190	65	39.67%	94.20%	19.83%	9
550	479	69	3	1	384	-100	22.1540	5.7487	191	65	39.87%	94.20%	19.83%	9
551	479	69	3	1	432	-100	22.2590	6.3403	191	67	39.87%	97.10%	9.81%	9
552	479	69	3	1	432	-100	22.4230	6.3316	190	65	39.67%	94.20%	9.81%	9

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
553	479	69	3	1	432	-100	22.4170	6.6754	191	65	39.87%	94.20%	9.81%	9
554	479	69	3	1	432	-100	22.0970	6.3122	189	64	39.46%	92.75%	9.81%	9
555	479	69	3	1	432	-100	22.2390	6.8889	189	65	39.46%	94.20%	9.81%	9
556	479	69	3	1	479	-100	22.6420	5.5585	188	64	39.25%	92.75%	0.00%	9
557	479	69	3	1	479	-100	22.6820	5.5585	188	64	39.25%	92.75%	0.00%	9
558	479	69	3	1	479	-100	22.6710	5.5585	188	64	39.25%	92.75%	0.00%	9
559	479	69	3	1	479	-100	22.8820	5.5585	188	64	39.25%	92.75%	0.00%	9
560	479	69	3	1	479	-100	22.7080	5.5585	188	64	39.25%	92.75%	0.00%	9
561	479	69	3	1	0	-20	22.0870	11.2618	191	64	39.87%	92.75%	100.00%	10
562	479	69	3	1	0	-20	22.0670	11.2618	191	64	39.87%	92.75%	100.00%	10
563	479	69	3	1	0	-20	22.1750	11.2618	191	64	39.87%	92.75%	100.00%	10
564	479	69	3	1	0	-20	25.1330	11.2618	191	64	39.87%	92.75%	100.00%	10
565	479	69	3	1	0	-20	25.3580	11.2618	191	64	39.87%	92.75%	100.00%	10
566	479	69	3	1	0	-10	23.5180	12.4817	191	64	39.87%	92.75%	100.00%	10
567	479	69	3	1	0	-10	22.3850	12.4817	191	64	39.87%	92.75%	100.00%	10
568	479	69	3	1	0	-10	24.0520	12.4817	191	64	39.87%	92.75%	100.00%	10
569	479	69	3	1	0	-10	24.2880	12.4817	191	64	39.87%	92.75%	100.00%	10
570	479	69	3	1	0	-10	23.6710	12.4817	191	64	39.87%	92.75%	100.00%	10
571	479	69	3	1	0	0	26.3320	18.8710	155	55	32.36%	79.71%	100.00%	10
572	479	69	3	1	0	0	26.2600	18.8710	155	55	32.36%	79.71%	100.00%	10
573	479	69	3	1	0	0	26.4610	18.8710	155	55	32.36%	79.71%	100.00%	10
574	479	69	3	1	0	0	26.4000	18.8710	155	55	32.36%	79.71%	100.00%	10
575	479	69	3	1	0	0	26.3450	18.8710	155	55	32.36%	79.71%	100.00%	10

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
576	479	69	3	1	0	10	29.0190	25.5556	126	45	26.30%	65.22%	100.00%	10
577	479	69	3	1	0	10	29.0090	25.5556	126	45	26.30%	65.22%	100.00%	10
578	479	69	3	1	0	10	29.0260	25.5556	126	45	26.30%	65.22%	100.00%	10
579	479	69	3	1	0	10	29.0800	25.5556	126	45	26.30%	65.22%	100.00%	10
580	479	69	3	1	0	10	29.0480	25.5556	126	45	26.30%	65.22%	100.00%	10
581	479	69	3	1	0	20	31.4970	30.7103	107	41	22.34%	59.42%	100.00%	10
582	479	69	3	1	0	20	31.4600	30.7103	107	41	22.34%	59.42%	100.00%	10
583	479	69	3	1	0	20	31.6910	30.7103	107	41	22.34%	59.42%	100.00%	10
584	479	69	3	1	0	20	31.6200	30.7103	107	41	22.34%	59.42%	100.00%	10
585	479	69	3	1	0	20	31.8080	30.7103	107	41	22.34%	59.42%	100.00%	10
586	479	69	3	1	0	30	35.2160	41.0678	59	23	12.32%	33.33%	100.00%	10
587	479	69	3	1	0	30	35.2250	41.0678	59	23	12.32%	33.33%	100.00%	10
588	479	69	3	1	0	30	35.3510	41.0678	59	23	12.32%	33.33%	100.00%	10
589	479	69	3	1	0	30	35.9040	41.0678	59	23	12.32%	33.33%	100.00%	10
590	479	69	3	1	0	30	36.0840	41.0678	59	23	12.32%	33.33%	100.00%	10
591	479	69	3	1	0	40	36.7570	48.2188	32	14	6.68%	20.29%	100.00%	10
592	479	69	3	1	0	40	37.0310	48.2188	32	14	6.68%	20.29%	100.00%	10
593	479	69	3	1	0	40	37.6000	48.2188	32	14	6.68%	20.29%	100.00%	10
594	479	69	3	1	0	40	37.1850	48.2188	32	14	6.68%	20.29%	100.00%	10
595	479	69	3	1	0	40	37.0720	48.2188	32	14	6.68%	20.29%	100.00%	10
596	479	69	3	1	0	50	38.2730	60.0769	13	5	2.71%	7.25%	100.00%	10
597	479	69	3	1	0	50	38.3010	60.0769	13	5	2.71%	7.25%	100.00%	10
598	479	69	3	1	0	50	38.3720	60.0769	13	5	2.71%	7.25%	100.00%	10

Table B.1. Computational results of the proposed algorithm. (cont.)

Iteration #	Number of Riders	Number of Drivers	Capacity	Rider Count	Number of Riders Rejecting First Match	JSS Limit	Computation Time	Average JSS of Matched Pairs	Number of Matched Riders	Number of Matched Drivers	Ratio of Matched Riders	Ratio of Matched Drivers	Ratio of Riders Accepting First Match	Case #
599	479	69	3	1	0	50	38.3410	60.0769	13	5	2.71%	7.25%	100.00%	10
600	479	69	3	1	0	50	41.0740	60.0769	13	5	2.71%	7.25%	100.00%	10
601	479	69	3	1	0	60	43.9120	70.0000	6	2	1.25%	2.90%	100.00%	10
602	479	69	3	1	0	60	39.1040	70.0000	6	2	1.25%	2.90%	100.00%	10
603	479	69	3	1	0	60	38.8900	70.0000	6	2	1.25%	2.90%	100.00%	10
604	479	69	3	1	0	60	38.9620	70.0000	6	2	1.25%	2.90%	100.00%	10
605	479	69	3	1	0	60	39.0180	70.0000	6	2	1.25%	2.90%	100.00%	10
606	479	69	3	1	0	70	39.1480	75.0000	3	2	0.63%	2.90%	100.00%	10
607	479	69	3	1	0	70	39.2240	75.0000	3	2	0.63%	2.90%	100.00%	10
608	479	69	3	1	0	70	39.4180	75.0000	3	2	0.63%	2.90%	100.00%	10
609	479	69	3	1	0	70	39.2350	75.0000	3	2	0.63%	2.90%	100.00%	10
610	479	69	3	1	0	70	43.7970	75.0000	3	2	0.63%	2.90%	100.00%	10
611	479	69	3	1	0	80	41.8190	-	0	0	0.00%	0.00%	100.00%	10
612	479	69	3	1	0	80	42.5720	-	0	0	0.00%	0.00%	100.00%	10
613	479	69	3	1	0	80	38.9270	-	0	0	0.00%	0.00%	100.00%	10
614	479	69	3	1	0	80	38.9110	-	0	0	0.00%	0.00%	100.00%	10
615	479	69	3	1	0	80	39.3470	-	0	0	0.00%	0.00%	100.00%	10