A Ridesharing Recommendation Framework with Hard and Soft Constraints

Mow Fei Kim and Hamidah Ibrahim

Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia

Keywords: Ridesharing, Hard Constraints, Soft Constraints, Recommendation System.

Abstract: Ridesharing, the act or practice of sharing automobile trips, has now becoming very popular due to many benefits it provides not only to the society, economic but also the environment. There are several ridesharing frameworks and applications being proposed, however in identifying a ridesharing group consisting of a driver and passengers (also known as riders), most of these solutions rely on the hard constraints which include timeslot (departure and arrival), location (departure and arrival), and capacity of the vehicle. Since these people will be sharing a ride together and they are strangers to each other, it is important to consider their preferences in identifying an ideal group. These preferences, called soft constraints in this paper, include among others race, age group, gender, non-smoking, etc. This is for assuring a pleasant, cosy, and most importantly a safe journey. Hence, this paper proposes a ridesharing recommendation framework that aims at identifying an ideal group by considering both the hard and the soft constraints. The framework is then embedded into a mobile application prototype, named SAGE, which aims to provide a safe, available, green, and economical ridesharing service.

1 INTRODUCTION

In this modern era, the rapid growth of technologies has brought significant changes to the society in various aspects of economy including sharing economy. Sharing economy also known as collaborative consumption economy, is a new economic phenomenon, in which the sharing and reusing of the redundant products or services by individuals or organisations are through online platforms (Juho et al., 2015; Georgina, 2018; Steven and Matthias, 2019). The popularity of the mobile phones, especially smart phones which are mainly used for communicating and accessing information on the Internet has facilitates the emergence of sharing economy. One of the focuses in sharing economy in transportation sector is ridesharing service.

Ridesharing, also known as carpooling, is now getting popular worldwide. It can be defined as the sharing of a ride by individuals in a personal vehicle with the same journey and schedules (Mitja et al., 2015; Hajra et al., 2018). Different from ride-hailing which creates new capacity issues, ridesharing which can fully utilise the capacity of the vehicle, brings many benefits, not only to society, but also environment (Biying et al., 2017; Xuan et al., 2017; Wang et al., 2018). To the society, it reduces the expenses of the transportation per individual such as toll fee, parking fee, and fuel fee as the individuals share those expenses. It also enhances the travel convenience as the effective utilisation of capacity can result in fewer trips, less travel time although spent the same amount of expenses in one trip (Ziru et al., 2016; Conner-Simons, 2017; Xuan et al., 2017; Hajra et al., 2018).

Since ridesharing service is a new and growing phenomenon in Malaysia (Indra and Ibrahim, 2017; Muhamad et al., 2019), only limited ridesharing services exist in Malaysia. Most of the ridesharing applications in the worldwide market aim at matching a driver to riders that are having the same journey and schedules. Hence, they only consider hard
constraints that include timeslot (departure and arrival), location (departure and arrival), and number of empty seats in forming the group. Soft constraints such as demographic (age group, gender, race, etc.) and environmental preferences (non-smoking, peace and quiet, music on, etc.) that reflect the users’ preferences are also important and should be taken into consideration in forming an ideal group. This is for assuring a pleasant, cosy and most importantly a safe journey (Benish et al., 2018; Diep et al., 2021) as these people will be sharing a ride together and they are strangers to each other. For instance, a female passenger will feel safer if she shares a ride with the other female passengers and driver. Hence, it is important to consider not only the hard constraints, but also the soft constraints in forming an ideal group.

In this paper, we propose a ridesharing recommendation framework, that provides a platform for the public to share rides with others who own the same journey and travel time. The framework attempts to identify an ideal group which consists of a driver and passengers (also known as riders) by considering both (i) the hard constraints which include the timeslot (departure and arrival), location (departure and arrival), and number of empty seats; and (ii) the soft constraints which include demographic (age, race, gender, etc.) and environmental preferences of the users (non-smoking, peace and quiet, music on, etc.). The proposed framework is then embedded into a mobile application prototype, named SAGE. In general the main contributions of this work are briefly described as follows:

- we have devised a matching mechanism to automatically identify an ideal group which consists of a driver and passengers by considering both the hard constraints and soft constraints,
- we have designed a flexible filtering function for users to express their preferences with regards to both the hard and soft constraints, and
- we have developed a mobile application prototype, SAGE, that incorporates the conceptual design of the proposed framework.

This paper is organized as follows: Section 2 presents the existing ridesharing solutions. In Section 3, the definitions and notations that are used in the rest of the paper are set out. Our proposed framework and its implementation are elaborated in Section 4 and Section 5, respectively. Conclusion and future works are presented in the final section of this paper, Section 6.

2 RELATED WORK

This section presents the reviews that have been conducted on several existing ridesharing solutions with emphasise given on the criteria they used in identifying a ridesharing group. We categorized these works into two main categories, namely: ridesharing frameworks and ridesharing applications.

Ridesharing frameworks: several ridesharing frameworks have been proposed, each having a unique aim. An early work by Douglas and Eduardo (2013) aims at maximizing the number of shared trips by proposing a framework and heuristic-based models. In the case of taxis, people going to close locations can share the costs of the trip, whereas in the case of rides, the driver and passengers can share costs as well. Later, Nusrat et al. (2016) propose a framework for dynamic vehicle pooling and a ridesharing system that are not limited to any particular type of vehicle. Hence, vehicle such as car, bus or even lorry can be pooled using the proposed system. Meanwhile, Na et al. (2017) propose a new ridesharing model, with a requirement that if a driver shares a ride with a rider, the shared route percentage must exceeds an expectation rate of the driver. Hajra et al. (2018) on the other hand introduce the highest aggregated score vehicular recommendation (HASVR) framework that recommends a vehicle with the highest aggregated score to the requesting passenger. They consider five parameters, namely: average time delay, vehicle’s capacity, fare reduction, driving distance, and profit increment in calculating the score. A real-time ride-sharing framework is proposed by Yuhan et al. (2021) with a dynamic timeframe and anticipation-based migration to handle the density variation of commuters in different time periods.

Ridesharing applications: There are several ridesharing applications available in the market which include Uber, Grab, Lyft, WeRide, and Ryde. Uber is a ride-hailing and ridesharing application in over 785 metropolitan areas worldwide. It is available for up to two people per party and provides up to two additional stops when requesting a ride, as well as allowing the users to choose their preferred driver based on their experiences with the driver. Whereas, Grab is a ride-hailing and ridesharing application which dominates the taxi market in Southeast Asia. GrabShare enables passengers to share a ride with another party who have the same destination, and passenger is only allowed to bring along a friend. GrabHitch enables the passenger to schedule ride in advance and get a shared lift at half of the usual price. Lyft is one of the largest ridesharing applications and it offers transport in over 600 U.S. cities including

Lyft is one of the largest ridesharing applications and it offers transport in over 600 U.S. cities including

WEBIST 2021 - 17th International Conference on Web Information Systems and Technologies
New York City, Los Angeles, and San Francisco, as well as 12 cities in Canada. Lyft allows the passengers to share a ride with another party and share fare with up to 60% discount rate. Meanwhile, WeRide is a new ridesharing application that provides carpooling service in Malaysia and Singapore. WeRide operates under the “You Decide Your Ride” concept in which users decide their ride preferences and cost sharing’s details. Another ridesharing application is Ryde, a Singapore-based transport-booking application. Similar to the other ridesharing applications, Ryde provides carpooling service that intends to help the car owners in offsetting petrol and parking costs, as well as to make travelling eco-friendly by reducing carbon emissions. Besides, Ryde provides the feature of setting preferred driver or passenger and sending them private requests.

From the above, most of the ridesharing solutions consider the hard constraints which include timeslot (arrival and departure), location (arrival and departure), and number of empty seats during the matching process as these are the core criteria that must be satisfied. Whereas, none of them take into account the preferences of the driver and passengers in identifying an ideal ridesharing group.

3 PRELIMINARIES

In this section, we present the necessary definitions and introduce the notations and symbols that are used throughout this paper. These symbols and notations are summarized in Table 2.

**Definition 1. Properties of a Driver:** a driver, \( D_i \), is associated with two main elements denoted by \( D_i = (P_{D_i}, R_{D_i}) \) where \( P_{D_i} \) is the profile of the driver, \( D_i \), which include race, age, gender, etc.; whereas \( R_{D_i} \) is the driver’s request.

**Definition 2. Properties of a Driver’s Request:** each request, \( R_{D_i} \), submitted by a driver, \( D_i \), is associated with four elements (\( SA-D_i \), \( TA-D_i \), \( PA-D_i \), \( C \)) where \( SA-D_i \) Spatial Attributes, represent the departure (\( SA_d \)) and arrival (\( SA_a \)) locations of the trip specified by the driver, \( D_i \); \( TA-D_i \) Temporal Attributes, represent the departure (\( TA_d \)) and arrival (\( TA_a \)) time/date of the trip specified by the driver, \( D_i \); \( PA-D_i \) Preference Attributes, represent the preferences of the driver, \( D_i \) like race, age group, gender, etc. of the passenger(s); whereas \( C \) is the capacity of the vehicle. \( SA-D_i \), \( TA-D_i \), and \( C \) are the hard constraints whereas \( PA-D_i \) is the soft constraint. Hard constraints are conditions that must be satisfied.

**Definition 3. Properties of a Passenger:** a passenger, \( P_j \), is associated with two main elements denoted by \( P_j = (P_{P_j}, R_{P_j}) \) where \( P_{P_j} \) is the profile of the passenger, \( P_j \), which include race, age, gender, etc.; whereas \( R_{P_j} \) is the passenger’s request.

<table>
<thead>
<tr>
<th>Symbols/Notations</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>( D_i )</td>
<td>The ( i )th driver</td>
</tr>
<tr>
<td>( P_{D_i} )</td>
<td>The profile of driver ( D_i )</td>
</tr>
<tr>
<td>( R_{D_i} )</td>
<td>The request submitted by ( D_i )</td>
</tr>
<tr>
<td>( SA-D_i = (SA_d, SA_a) )</td>
<td>Spatial attributes associated to ( D_i ) consisting of a departure location, ( SA_d ), and an arrival location, ( SA_a )</td>
</tr>
<tr>
<td>( TA-D_i = (TA_d, TA_a) )</td>
<td>Temporal attributes associated to ( D_i ) consisting of a departure time/date, ( TA_d ), and an arrival time/date, ( TA_a )</td>
</tr>
<tr>
<td>( PA-D_i )</td>
<td>Preference attributes, i.e. the preferences of ( D_i )</td>
</tr>
<tr>
<td>( C )</td>
<td>Capacity of the vehicle owns by the driver, ( D_i )</td>
</tr>
<tr>
<td>( P_{P_j} )</td>
<td>The ( j )th passenger</td>
</tr>
<tr>
<td>( R_{P_j} )</td>
<td>The request submitted by ( P_j )</td>
</tr>
<tr>
<td>( SA-P_j = (SA_d, SA_a) )</td>
<td>Spatial attributes associated to ( P_j ) consisting of a departure location, ( SA_d ), and an arrival location, ( SA_a )</td>
</tr>
<tr>
<td>( TA-P_j = (TA_d, TA_a) )</td>
<td>Temporal attributes associated to ( P_j ) consisting of a departure time/date, ( TA_d ), and an arrival time/date, ( TA_a )</td>
</tr>
<tr>
<td>( PA-P_j )</td>
<td>Preference attributes, i.e. the preferences of ( P_j )</td>
</tr>
<tr>
<td>( h )</td>
<td>The number of hard constraints/attributes</td>
</tr>
<tr>
<td>( s )</td>
<td>The number of soft constraints/attributes</td>
</tr>
<tr>
<td>( n = h + s )</td>
<td>The number of criteria considered in the matching process</td>
</tr>
<tr>
<td>( W_{R_{D_i} \rightarrow R_{P_j}} )</td>
<td>The weight value between ( R_{D_i} ) and ( R_{P_j} )</td>
</tr>
<tr>
<td>( TW-P_j )</td>
<td>The total weight value of the ( j )th group</td>
</tr>
<tr>
<td>( P_{D_i} )</td>
<td>The ideal group</td>
</tr>
<tr>
<td>( c_\alpha )</td>
<td>The ( \alpha )th criterion</td>
</tr>
</tbody>
</table>

**Definition 4. Properties of a Passenger’s Request:** each request, \( R_{P_j} \), submitted by a passenger, \( P_j \), is associated with three elements (\( SA-P_j \), \( TA-P_j \), \( PA-P_j \)) where \( SA-P_j \) Spatial Attributes, represent the departure (\( SA_d \)) and arrival (\( SA_a \)) locations of the
trip specified by the passenger, \( P_j \); \( TA-P_j \), Temporal Attributes, represent the departure \((TA_\omega)\) and arrival \((TA_\omega)\) time/date of the trip specified by the passenger, \( P_j \); \( PA-P_j \), Preference Attributes, represent the preferences of the passenger, \( P_j \), like race, age group, gender, etc. of the group members. Both \( SA-P_j \) and \( TA-P_j \) are the hard constraints, whereas \( PA-P_j \) is the soft constraint.

**Definition 5. Totally Match between \( R_{D_i} \) and \( R_{P_j} \):** the request of a driver, \( R_{D_i} \), is said to totally match with the request of a passenger, \( R_{P_j} \), iff both the hard and soft constraints specified by the driver, \( D_i \), are the same as the hard and soft constraints specified by the passenger, \( P_j \), i.e. \( SA-D_i = SA-P_j \land TA-D_i = TA-P_j \land PA-D_i = PA-P_j \).

**Definition 6. Not Match between \( R_{D_i} \) and \( R_{P_j} \):** the request of a driver, \( R_{D_i} \), is said to not match with the request of a passenger, \( R_{P_j} \), iff at least one of the hard constraints specified by the driver, \( D_i \), does not match with the hard constraints specified by the passenger, \( P_j \), i.e. \( SA-D_i \neq SA-P_j \lor TA-D_i \neq TA-P_j \).

**Definition 7. Partially Match between \( R_{D_i} \) and \( R_{P_j} \):** the request of a driver, \( R_{D_i} \), is said to partially match with the request of a passenger, \( R_{P_j} \), iff the hard constraints specified by the driver, \( D_i \), and the passenger, \( P_j \), are the same, i.e. \( SA-D_i = SA-P_j \land TA-D_i = TA-P_j \); whereas there is at least one preference specified by \( D_i \) that does not match with the soft constraints specified by the passenger, \( P_j \).

For the following definitions, we use the following variables to represent the number of criteria considered in the matching process:

- **Hard constraints:** \( h \) criteria
- **Soft constraints:** \( s \) criteria
- **Total criteria:** \( n = h + s \)

Also, every single criterion that is matched is given a value 1 whereas not matched is given a value 0.

**Definition 8. Weight for Totally Match between \( R_{D_i} \) and \( R_{P_j} \):** for a given \( n \) criteria the weight for totally match between \( R_{D_i} \) and \( R_{P_j} \), denoted by \( W_{R_{D_i} - R_{P_j}} \), is \( h \).

**Definition 9. Weight for Not Match between \( R_{D_i} \) and \( R_{P_j} \):** for a given \( n \) criteria the weight, \( W_{R_{D_i} - R_{P_j}} \), for not match between \( R_{D_i} \) and \( R_{P_j} \), is \( 0 \leq W_{R_{D_i} - R_{P_j}} \leq (h - 1) + s \).

**Definition 10. Weight for Partially Match between \( R_{D_i} \) and \( R_{P_j} \):** for a given \( n \) criteria the weight, \( W_{R_{D_i} - R_{P_j}} \), for partially match between \( R_{D_i} \) and \( R_{P_j} \), is \( h \leq W_{R_{D_i} - R_{P_j}} \leq h + (s - 1) \).

**Definition 11. Possible Group of a \( R_{D_i} \):** given the profile of a driver \( D_i \), \( P_{D_i} \), the driver’ s request, \( R_{D_i} \), and a set of passengers, \( P = \{ P_1, P_2, \ldots, P_l \} \), a possible group is defined as \( P_{D_i} = \{ P_1, P_2, \ldots, P_m \} \), where \( m \leq C \); \( \forall P_j \in P_{D_i}, R_{D_i} \), and \( R_{P_j} \) are either totally match or partially match, and \( q \) is the \( q \)th possible group derived for \( D_i \). If there are \( k \) passengers that meet the above conditions, then the number of possible groups that can be derived is \( \frac{k!}{(k-q)!} \). Hence, given a possible group, \( P_{D_i} \), the total weight for the group, \( TW_{P_{D_i}} \), is given by the formula, \( TW_{P_{D_i}} = \sum_{q=1}^{m} W_{R_{D_i} - R_{P_j}} \).

The problem tackled by this paper is formulated as follows: given the profile of a driver \( D_i \), \( P_{D_i} \), the driver’ s request \( R_{D_i} \), a set of passengers \( P = \{ P_1, P_2, \ldots, P_l \} \), find an ideal group \( P_{D_i} = \{ P_1, P_2, \ldots, P_m \} \), where \( m \leq C \); \( \forall P_j \in P_{D_i}, R_{D_i} \), and \( R_{P_j} \) are either totally match or partially match; and the total weight, \( TW_{P_{D_i}} = \sum_{q=1}^{m} W_{R_{D_i} - R_{P_j}} = \max(TW_{P_{D_i}}-1, TW_{P_{D_i}}-2, \ldots, TW_{P_{D_i}}-r) \), is \( r \) the number of possible groups derived based on \( R_{D_i} \).

Throughout this paper, the samples given in Table 1 are used to clarify the phases of the proposed framework. Table 1(a) and Table 1(b) present samples of drivers’ requests and passengers’ requests, respectively. Here, we assume the following: list of departure locations, \( SA_d = \{ A, B, C, D, E \} \); list of arrival locations, \( SA_a = \{ A, B, C, D, E \} \), the requests are on the same date, and the time is given in range, for instance, \( TA_d = [12:13] \) which means the departure time is between 12 to 13 noon. This can be easily modified to suit other possible ranges of time slots. For example [12:12] means exactly at 12 noon. Similar notation is used for arrival time, \( TA_a \). Meanwhile, for the preferences, we assume there are \( s \) preferences, labelled as \( pr_1, pr_2, \ldots, pr_s \). Examples of possible preferences are race, age group, gender, etc. Since indicating the preferences is optional, hence the symbol ‘−’ is used to indicate that the preference is not important to the driver/passenger.
Table 2: Sample of Requests.

<table>
<thead>
<tr>
<th>$D_1$</th>
<th>$P_{D_1}$</th>
<th>$SA-D_1$</th>
<th>$TA-D_1$</th>
<th>$PA-D_1$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$SA_a$</td>
<td>$TA_a$</td>
<td>$pr_1$</td>
<td>$pr_2$</td>
</tr>
<tr>
<td>$D_1$</td>
<td>$P_{D_1}$</td>
<td>A</td>
<td>B</td>
<td>[12:13]</td>
<td>[14:15]</td>
</tr>
<tr>
<td>$D_2$</td>
<td>$P_{D_2}$</td>
<td>D</td>
<td>E</td>
<td>[9:10]</td>
<td>[12:13]</td>
</tr>
<tr>
<td>$D_3$</td>
<td>$P_{D_3}$</td>
<td>A</td>
<td>C</td>
<td>[7:8]</td>
<td>[10:11]</td>
</tr>
<tr>
<td>$D_4$</td>
<td>$P_{D_4}$</td>
<td>E</td>
<td>D</td>
<td>[16:17]</td>
<td>[19:20]</td>
</tr>
<tr>
<td>$D_5$</td>
<td>$P_{D_5}$</td>
<td>B</td>
<td>E</td>
<td>[9:10]</td>
<td>[12:13]</td>
</tr>
</tbody>
</table>

(a) Drivers’ Requests

<table>
<thead>
<tr>
<th>$P_j$</th>
<th>$P_{P_j}$</th>
<th>$SA-P_j$</th>
<th>$TA-P_j$</th>
<th>$PA-P_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1$</td>
<td>$P_{P_1}$</td>
<td>A</td>
<td>B</td>
<td>[12:13]</td>
</tr>
<tr>
<td>$P_2$</td>
<td>$P_{P_2}$</td>
<td>D</td>
<td>E</td>
<td>[9:10]</td>
</tr>
<tr>
<td>$P_3$</td>
<td>$P_{P_3}$</td>
<td>A</td>
<td>B</td>
<td>[12:13]</td>
</tr>
<tr>
<td>$P_4$</td>
<td>$P_{P_4}$</td>
<td>A</td>
<td>C</td>
<td>[16:17]</td>
</tr>
<tr>
<td>$P_5$</td>
<td>$P_{P_5}$</td>
<td>A</td>
<td>B</td>
<td>[12:13]</td>
</tr>
<tr>
<td>$P_6$</td>
<td>$P_{P_6}$</td>
<td>A</td>
<td>C</td>
<td>[7:8]</td>
</tr>
</tbody>
</table>

(b) Passengers’ Requests

Figure 1: The Proposed Ridesharing Recommendation Framework.
4 THE PROPOSED FRAMEWORK

This section presents our proposed framework which aims at identifying an ideal group, \( P_1 \), given a driver, \( D_i \), with request, \( R_{D_i} \), and a set of passengers, \( P = \{P_1, P_2, \ldots, P_j\} \) with each passenger having their own request denoted by \( R_{P_j} \). The proposed framework consists of five main phases as shown in Figure 1 before an ideal group, \( P_1 \), is identified and recommended. These phases are: (i) match \( R_{D_i} \) and \( R_{P_j} \), (ii) calculate \( W_{R_{D_i} - R_{P_j}} \) for each matched \( P_1 \) and \( R_{P_j} \), (iii) generate possible groups of \( P_1 \), i.e. \( P_{D_1} - 1, P_{D_1} - 2, \ldots, P_{D_1} - r \), (iv) calculate \( TW \cdot P_{D_1} - q = \sum_{j=1}^{m} W_{R_{D_i} - R_{P_j}} \) for each possible group, and (v) find the ideal group, \( P_1 \), where \( TW \cdot P_{D_1} - \alpha = \sum_{j=1}^{m} W_{R_{D_i} - R_{P_j}} = \max (TW \cdot P_{D_1} - 1, \ldots, TW \cdot P_{D_1} - r) \). These phases are further elaborated in the following paragraphs.

Phase 1 Match \( R_{D_i} \) and \( R_{P_j} \) – This phase aims at identifying the passengers that match with a given driver’s request, \( R_{D_i} \). Hence, at this stage the \( R_{D_i} \) and \( R_{P_j} \) of each passenger are compared based on the hard constraints. Those pairs of \( R_{D_i} \) and \( R_{P_j} \) that do not match as defined by Definition 6 are filtered out. Hence, only those pairs of \( R_{D_i} \) and \( R_{P_j} \) that meet the Definition 5 and Definition 7 are considered in the next phases. These pairs are said to satisfy the conditions \( SA_{D_1} = SA_{P_1} \) and \( TA_{D_1} = TA_{P_1} \) where \( SA_{D_1} \) (\( SA_{P_1} \)) represents the departure (arrival) locations of the trip specified by the driver \( D_1 \) (passenger \( P_1 \), respectively) and \( TA_{D_1} \) (\( TA_{P_1} \)) represents the departure (arrival) time/date of the trip specified by the driver \( D_1 \) (passenger \( P_1 \), respectively). The following algorithm details the steps of this phase.

Example: Based on Table 1, given the hard constraints of \( R_{D_1} = < SA_{D_1}, SA_{A_1}, TA_{A_1}, TA_{A_2} > = < A, B, [12:13], [14:15] > \), then \( FP = \{P_1, P_2, P_3\} \). Meanwhile, \( FP = \{P_2\} \) for the request \( R_{D_2} = < D_i, E, [9:10], [12:13] > \).

Phase 2 Calculate \( W_{R_{D_1} - R_{P_j}} \) for each matched \( R_{D_1} \) and \( R_{P_j} \) – Once, the set of passengers that satisfied the hard constraints has been identified, i.e. \( FP \), then the weight \( W_{R_{D_1} - R_{P_j}} \) for each matched pair \( R_{D_1} \) and \( R_{P_j} \) is calculated. The weight value indicates the degree of similarities between \( R_{D_1} \) and \( R_{P_j} \). Here, the Definition 8 and Definition 10 are applied. If every criterion, \( c_u \), that is matched is given a value 1 whereas not matched is given a value 0, and assuming that there are \( h \) criteria for hard constraints and \( s \) criteria for soft constraints with \( n = h + s \), then if \( R_{D_1} \) and \( R_{P_j} \) are totally matched (100% similar), the weight \( W_{R_{D_1} - R_{P_j}} = \sum_{u=1}^{n} w(c_u) = n \) where \( w(c_u) \) is the weight value given based on the criterion, \( c_u \). However, if \( R_{D_1} \) and \( R_{P_j} \) are partially matched (100% similar based on \( h \) criteria whereas not 100% similar based on \( s \) criteria), the weight \( W_{R_{D_1} - R_{P_j}} = h \leq \sum_{u=1}^{n} w(c_u) \leq h + (s - 1) \). Here, we assume every criterion is equally important. This is as given below:

```
Input: \( R_{D_i}, P = \{P_1, P_2, \ldots, P_j\}, R_{P_j} \)
Output: \( FP = \{P_1, P_2, \ldots, P_j\} \)
Step 1: \( FP = \{\} \)
Step 2: For each \( P_i \in P \) do
Step 3: If \( SA_{D_i} \) of \( D_i = SA_{P_1} \) AND \( TA_{D_i} = TA_{P_1} \)
\( SA_{a} \) of \( D_i \) = \( SA_{a} \) of \( P_1 \) AND \( TA_{a} = TA_{a} \) of \( P_1 \) AND \( TA_{A_2} = TA_{A_2} \) of \( P_1 \) AND Then \( FP = FP \cup P_i \)
Step 4: Return \( FP \)
```

Example: Given the \( FP = \{P_1, P_3, P_5\} \) derived in the previous phase, the phase weight \( W_{R_{D_1} - R_{P_j}} \) for each matched pair \( R_{D_1} \) and \( R_{P_j} \) is as follows: \( W_{R_{D_1} - R_{P_3}} = 7, W_{R_{D_1} - R_{P_5}} = 6, \) and \( W_{R_{D_1} - R_{P_2}} = 5 \). Here, we assume \( h = 4 \) (\( SA_{D_1}, SA_{A_1}, TA_{A_1}, TA_{A_2} \)) and \( s = 3 \) (\( pr_1, pr_2, pr_3 \)). This can be easily extended to cater other preferences. Based on the weight and the number of criteria considered, the requests \( R_{D_1} \) and \( R_{P_j} \) reflect totally matched and Definition 8 is applied. Meanwhile, the pairs \( R_{D_1} \) and \( R_{P_3} \) and \( R_{D_1} \) and \( P_{P_2} \) reflect partially matched whereby Definition 10 is applied. Meanwhile, the phase weight \( W_{R_{D_1} - R_{P_2}} = 5 \).

Phase 3 Generate possible groups of \( R_{D_1} \), i.e. \( P_{D_1} - 1, P_{D_1} - 2, \ldots, P_{D_1} - r \) – Given the capacity, \( C \), defined by the driver, \( D_1 \), and based on the \( FP \) identified in the first phase, this phase generates all possible groups by simply performing permutation on the elements of \( FP \). The number of possible groups as defined by Definition 11 is \( \binom{k}{C} \) where \( k \) is the number of
elements (passengers) in \( F_P \). The number of possible groups reflects the number of ways the \( k \) passengers can be grouped based on the capacity, \( C \). For instance, if \( C = 2 \) and \( k = 4 \), then there are \( \frac{4!}{2!(4-2)!} = 6 \) different ways to group the 4 passengers where each group has two passengers to accommodate the capacity, \( C = 2 \). If these four passengers are labelled as \( P_1, P_2, P_3, P_4 \), then the 6 different ways are \( \{P_1, P_2\}, \{P_1, P_3\}, \{P_1, P_4\}, \{P_2, P_3\}, \{P_2, P_4\}, \{P_3, P_4\} \). Although to select 2 passengers from these 4 potential passengers that are having the closest preferences to \( D_j \) can be easily done by picking 2 passengers with the highest \( W_{R_{D_j}-R_F} \) values, it is natural to recommend several solutions in a recommendation system. In other words, these 6 different ways should be recommended in certain ranking order whereas the final decision is to be made by the user. It also caters the possibilities of passengers cancelling their trip.

**Example:** Given the \( F_P = \{P_1, P_3, P_4\} \) derived in the previous phase, the number of possible groups is \( \frac{3!}{2!(3-2)!} = 3 \). These groups are as follows: \( P_{D_1} - 1 = \{P_2, P_3\} \), \( P_{D_1} - 2 = \{P_1, P_3\}, P_{D_1} - 3 = \{P_3, P_4\} \). Since, there is only one passenger that matched the request of \( D_2 \), \( P_{D_2} - 1 = \{P_1\} \).

**Phase 4. Calculate** \( TW-P_{D_1}, q = \sum_{j=1}^{m} W_{R_{D_1}-R_F} \) for each possible group. Once all possible groups have been generated, before any of them is recommended, it is important to determine among all these possible groups which one is the ideal group as highlighted in the problem formulation. Literally, the group with the highest total weight is the ideal group. Hence, this phase calculates the total weight, \( TW-P_{D_1}, q = \sum_{j=1}^{m} W_{R_{D_1}-R_F} \), for each possible group.

**Example:** Based on the possible groups derived in the previous phase, \( P_{D_1} - 1 = \{P_1, P_3\}, P_{D_1} - 2 = \{P_1, P_3\}, P_{D_1} - 3 = \{P_3, P_4\} \), the total weight, \( TW-P_{D_1}, q \) for each group is as follows: \( TW-P_{D_1}, 1 = W_{R_{D_1}-R_F} = 13 \), \( TW-P_{D_1}, 2 = W_{R_{D_1}-R_F} = 12 \), and \( TW-P_{D_1}, 3 = W_{R_{D_1}-R_F} = 11 \).

**Phase 5. Find the ideal group.** \( P_{D_1}, o = \{P_1, P_2, ..., P_m\} \) where \( TW-P_{D_1}, o = \sum_{j=1}^{m} W_{R_{D_1}-R_F} = \max(TW-P_{D_1}, 1, TW-P_{D_1}, 2, ..., TW-P_{D_1}, r) \) – This phase determines the ideal group of a given \( D_1 \) denoted by \( P_{D_1}, o \) by analysing the following optimal function: \( TW-P_{D_1}, o = \sum_{j=1}^{m} W_{R_{D_1}-R_F} = \max(TW-P_{D_1}, 1, TW-P_{D_1}, 2, ..., TW-P_{D_1}, r) \). In other words, the possible group with the maximum total weight value is the ideal group for the given \( R_{D_1} \). It reflects the similarities of the group members. If \( n \) is the total number of criteria considered and \( k \) is the number of selected passengers, then \( n \times k \) is the maximum total weight which implies all members of the group have the same exact preferences. For instance, if we assume \( h = 4 (SA_d, SA_h, TA_d, TA_h) \) and \( s = 3 (pr_1, pr_2, pr_3) \) as given in Table 1, if the maximum total weight achieved is \( 7 \times 2 = 14 \), this implies that the 2 passengers and the driver have the same preferences with 100% similarities.

**Example:** Applying the above optimal function to the total weights derived in the previous phase, \( TW-P_{D_1}, o = \sum_{j=1}^{m} W_{R_{D_1}-R_F} = \max(TW-P_{D_1}, 1, TW-P_{D_1}, 2, TW-P_{D_1}, 3) \) as given in Table 1, if the maximum total weight achieved is \( 13 \), \( 12 \), and \( 11 \), then this means the ideal group is \( P_{D_1} - 1 = \{P_1, P_3\} \).

## 5 FRAMEWORK IMPLEMENTATION

This section presents the results of implementing the proposed framework on a ridesharing mobile application prototype, named SAGE. It provides a platform for the public to share rides with others who own the same journey and travel time. SAGE aims to provide a safe, available, green and economical ridesharing as defined below: (i) safe – among the criteria used by SAGE in identifying an ideal group is the demographic of the group members. This is to ensure that members of the group will have a pleasant, cozy, and most importantly a safe journey; (ii) available – SAGE is available from anywhere and anytime of the day, i.e. it is available 24 hours 7 days a week; (iii) green – SAGE focuses on ridesharing service that aims to utilise the capacity of a vehicle to reduce the emission of air pollutant from vehicle exhaust, hence keeping a green environment; and (iv) economical – it reduces the expenses of the transportation per individual, as these expenses are borne among the members.

SAGE is developed using Android platform. A smartphone, Huawei Mate 20, as well as two virtual devices created from Android Studio are used as the emulators to test SAGE. The minimum SDK version for SAGE is set to API level 24 which is compatible to run on approximately 73.7% of the devices that are active on the Google Play Store. Java programming language is used as the primary programming language in developing SAGE.

Figure 2 presents samples of SAGE interfaces when a driver, \( D_i \), registered into the system.
profile of the driver \( D_i \) is captured at this stage which among others include name, email address, phone number, gender, race, date of birth, etc. It also captured the details of the vehicle owns by the driver, \( D_i \). Similar interfaces are also designed to capture the details of a passenger, \( P_j \), as shown in Figure 3.

Figure 2: Interfaces for registration of a \( D_i \).

Figure 3: Interfaces for registration of a \( P_j \).

Figure 4 presents the interfaces for users to schedule a trip. Here, users either \( D_i \) or \( P_j \) are required to fill in the fields related to hard constraints which include From (\( SA_d \)), To (\( SA_d \)), Date (\( TA_d, TA_a \)), Time Range (\( TA_d \)), and Seater (\( C \), only applicable for a driver). As for the soft constraints, we have included the following: race, gender, age group, language, and preferred environment. In specifying the preferences, users may select a particular value or choose All which implies that the field is not important to the users. Meanwhile, Figure 5 shows a sample of interfaces during a trip.

Once a request has been submitted by a driver, \( D_i \), all requests submitted by passengers that are saved in the system are filtered and only those requests that matched with the hard constraints as specified by the driver, \( D_i \), are listed. This list represents the \( FP \) list described in Section 4. The driver can further filter the list to narrow down the searching.

Figure 4: Interfaces for scheduling a trip.

Figure 5: Example of a trip.

We have tested \( SAGE \) with several cases and the initial results show that \( SAGE \) is functioning well according to the conceptual framework described in Section 4. However, more testing needs to be conducted before \( SAGE \) can be fully utilised.

6 CONCLUSION AND FUTURE WORK

Ridesharing is now becoming one of the popular sharing economy due to the benefits it provides. This paper proposes a ridesharing recommendation framework that aims at identifying an ideal group consisting of a driver and passengers by considering both the hard and soft constraints. There are five phases, namely: (i) match \( R_{D_i} \) and \( R_{P_j} \), (ii) calculate
the weight, $W_{D_i} - R_{P_j}$, for each matched $R_{D_i}$ and $R_{P_j}$,
(iii) generate possible groups of $R_{D_i}$, (iv) calculate the total weight, $TW - D_i - q$, for each possible group, and
(v) find the ideal group, $P_{D_i - q} = \{P_1, P_2, ..., P_m\}$. The framework is embedded into a mobile application prototype, named SAGE. SAGE has been tested in a small-scale environment. Hence, testing SAGE in a large-scale environment will be the next step to be conducted. Moreover, we attempt to further analyse the performance of the proposed framework/SAGE with regard to processing time and accuracy.

REFERENCES


https://www.lyft.com
https://www.rydesharing.com
https://www.uber.com
https://weride.my