

A resource sharing (sharing platform) scheme on online taxi services

Helen Burhan^{1,*}, Sutanto Soehodho², and Nahry Yusuf²

¹Department of Mathematics, Faculty of Mathematics and Natural Science, University of Indonesia, Depok, Indonesia

²Department of Civil Engineering, Faculty of Engineering, University of Indonesia, Depok, Indonesia

Abstract. This paper will review the match between single driver and single rider in online taxi services through a resource sharing (sharing platform) for the operators with the objectives to maximize the profit for drivers (operators) and minimize waiting time for passengers so that the matching rate is higher. A low matching rate between rider and driver can cause the consumer to drop the services. The matching between single driver and single rider in online taxi services through a sharing platform scheme is formulated in maximum weighted bipartite matching problem. To solve the proposed model, we use Kuhn Munkres Algorithm, while to solve the shortest path for the driver to pick up the passenger and the shortest path of passenger's origin destination, modified Dijkstra with adaptive algorithm based on Wei Peng et.al (2012) is used. Based on illustrative example with several cases, we found a resource sharing scenario can optimize the matching between driver and rider and moreover can solve the surge pricing problem which is deemed as less transparent to customer

1 Introduction

Ride sourcing services, more widely known as online taxi, have become a new transportation alternative to people in big cities all over the world, including in Jakarta, Indonesia. The ride sourcing services use private vehicles as a means of public transportation for passengers similar to taxi services, which are regulated through an application utilizing technology of smartphone, internet and GPS [1]. The inadequacy of public transportation services, while the use of technology is very fast growing in the society for the past few years, has boosted the use of online taxi services.

In Indonesia, specifically in Jakarta, previously there were 3 big companies operating in online taxi services, which were Uber, Grab and GoCar [2], whereas in 2018 Uber was acquired by Grab. Each company has its own specific characteristics in their online taxi services for the consumers. These companies providing the online transportation services are not interlinked with one another, so that the route setting to match drivers and riders is also not interlinked, while a lot of the drivers are actually joining or working under 2 or 3 operators at the same time. On the other hand, from the operational point of view, the online taxi services are not yet optimal, since the waiting time for customers is still quite long or is not as per estimated waiting time stated in the app, and the estimated arrival time is not correct either. In addition, in certain hours and conditions, the fare charged to customers can drastically increase or fluctuate, which is

called a surge pricing, which depends on the available vehicles and customer demand levels in the specific areas.

Based on operational issue of ride sourcing services, in this paper, we will evaluate the matching between single drivers and single riders in online taxi services, using a resource sharing (sharing platform/operator) schemes, in order to optimize the profit for operators by minimizing the rider's waiting time so that the matching rate can be higher. A low matching rate between drivers and riders can cause the customers to drop the service. The resource-sharing (sharing platform) scheme is interlinking drivers from different operators (companies) in providing services to the customers, where a driver from company X can serve customers from company Y and vice versa, with a predetermined profit sharing. In addition, the resource sharing scheme is expected to solve the surge pricing problem which is deemed as less transparent for the customers [3]. The matching between single driver and single rider in online taxi services through a sharing platform scheme is formulated in maximum weighted bipartite matching problem. To solve the proposed model, we use Kuhn Munkres Algorithm [4], while to solve the shortest path for the driver to pick up the passenger and the shortest path of passenger's origin destination, modified Dijkstra with adaptive algorithm based on Wei Peng et.al (2012) [5] is used.

The rest of the paper is organized as follows. Section 2 presents a literature review, while Section 3 describes the problem formulation. Section 4 on the other hand presents the solution method using modified Dijkstra with

* Corresponding author: helen.burhan@sci.ui.ac.id

adaptive algorithm [5] and Kuhn Munkres Algorithm [4], and Section 5 presents illustrative example which describes a possible scenario that can occur. Finally, Section 6 concludes and illustrates future research directions.

2 Literature review

Ride-sourcing service was started in San Fransisco by Uber in 2009 [6], and it was fast spreading to other big cities all over the world, as other companies providing similar services were also popping up. The ride-sourcing service itself was adopting the previously existing ride-sharing concept, so that the ride-sourcing service was often considered as ride-sharing services, but with a profit-oriented concept [1,6].

Different from the common public transportation services, a ride-sourcing service is using a dynamic pricing system, which is taking into consideration the real supply and demand condition in each specific area [7]. For example, when the demand for service is high, while the vehicle availability (supply) is low, the fare will be higher than the regular fare, known as surge pricing. But the formulation for the surge pricing is not transparent to the consumers so that they can feel cheated and drop the service [3]. On the other hand, drivers are taking advantage during the surge pricing, since they will earn more money, so that more drivers opt to drive during the surge pricing time [8]. If this situation continues, in the long run customers will leave the service.

The application of surge pricing is not always harmful, because a flat fare can be unfair for a specific zone [9]. However, since this ride-sourcing service is relatively new, the research reviewing the services, specifically on service fares is still very limited. Galichon and Hsieh [10] developed revised surge pricing algorithm to minimize the total inefficiency resulting from time waited in line considering demand and supply uncertainties, while Zha et.al [9] developed the commission rate cap regulation that reaps the flexibility of spatial pricing to solve the surge pricing problem.

In this paper, we will develop a resource-sharing (sharing platform) scheme, which is interlinking drivers from different operators (companies) in providing services to the customers, where a driver from company X can serve customers from company Y, and vice versa, with a predetermined profit sharing. This scheme is expected to benefit the customers with shorter waiting time, while the drivers can optimize their income, and in addition will make the surge pricing calculation, which in this case is a predetermined profit, more transparent so that no parties will feel cheated or harmed.

According to Agatz [11], the problem in optimizing the ride-sharing services for single driver-single rider case can be represented as bipartite matching problem. In this study, the ride-sourcing services for the single driver-single rider will be formulated in maximum weighted bipartite matching.

3 Problem formulation

The whole process of the assignment of a request to a vehicle with resource sharing (sharing platform) scenario can be described in figure (1)-(3), assuming there are two different operators (platforms) providing ride sourcing services, where it can be enlarged to n platforms.



Fig. 1. Customer announces request

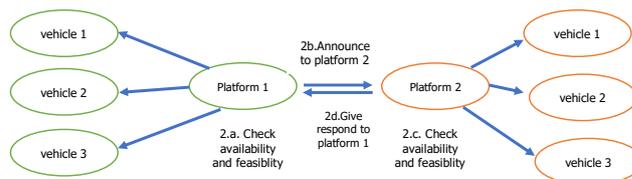


Fig. 2. Platform 1 checks the availability and feasibility of the vehicle while announcing to other platform about the request

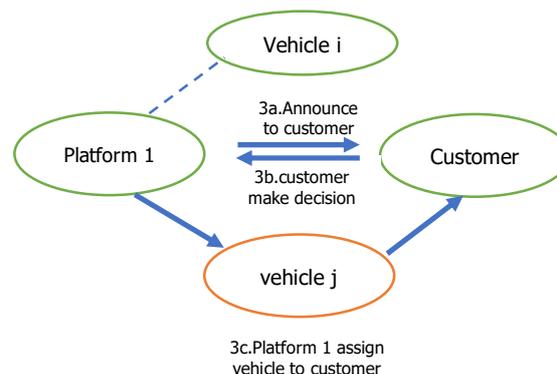


Fig. 3. Platform announces to customer, customer makes decision and platform assign request

Initially, customer will announce the request to platform 1 (Figure 1), and then platform 1 will check the availability and feasibility of the vehicle that can serve the customer (2a), while announcing to platform 2 about the request (2b) as described in Figure 2. When the platform 1 announces the request to platform 2, it also offers the trip fare that should be paid by the customer and profit that can be shared if a vehicle on platform 2 serve the request. Next, platform 2 will check the availability and feasibility of its vehicle (2c) and then responds to the platform 1, which available vehicle on it that feasible (Figure 2).

At the final stage (Figure 3), platform 1 will announce to customer (3a) which vehicles are available, from platform 1 and platform 2, each having different waiting time and trip fare that should be paid by the customer. Then the customer will choose which vehicle will serve him/her (3b). Next, platform 1 will assign the chosen vehicle to customer (3c).

Based on the assignment process as described above, the resource sharing scheme to optimize ride sourcing services for single driver-single rider problem considered in this paper is as follow. Let the number of operators

(platforms) providing ride sourcing services is 2. Let S be the set of all vehicles, where $S = S_1 \cup S_2$; S_n be the set of all vehicles of operator n ; $n = 1,2$. Let T be the set of all passengers, where $T = T_1 \cup T_2$; T_k be the set of all passengers of platform k ; $k = 1,2$.

Given a complete bipartite graph $G = (V, E)$, where $V = S \cup T$ be the set of nodes, S is associated with vehicle nodes and T is associated with passenger nodes and E be the set of edges, where $E = \{(i, j) | i \in S, j \in T\}$. Let F be the set of feasible matches, the binary decision variable x_{ij} indicate whether the edge is in an optimal matching ($x_{ij} = 1$) or not ($x_{ij} = 0$), then the single driver (vehicle) - single rider matching problem that maximizes the number of matches in order to maximize the profit for drivers (operators) and minimize waiting time for passengers with sharing platform scenario can be formulated as follows:

$$\text{Max } \sum_{(i,j) \in F} w_{ij} x_{ij} \quad (1)$$

Subject to :

$$\sum_{i \in S, (i,j) \in F} x_{ij} \leq 1 \quad \forall j \in T \quad (2)$$

$$\sum_{i \in T, (i,j) \in F} x_{ij} \leq 1 \quad \forall i \in S \quad (3)$$

$$x_{ij} \in \{0,1\}, \quad \forall (i, j) \in F \quad (4)$$

The objective function (1) maximizes the number of matches. Constraints (2) and (3) assure that each vehicle and each rider is only included in at most one match in an optimal matching. To determine the set of feasible matches F , we impose time feasibility constraints, as detailed below.

3.1. Time feasibility

Time feasibility refers to the time of vehicle to pick up and drop off passenger should relate to passenger's time interval preferences, for a match to be made.

Each customer/passenger $p \in T$ is associated with request time t_{req} ; an earliest pick up time e_p^{dept} and latest pick up time l_p^{dept} , from pick up (origin) location o_p ; an earliest drop off time e_p^{arr} and latest drop off time l_p^{arr} at drop off (destination) location d_p ; origin-destination (OD) travel distance $Distance_{od}^p$; and OD travel time TT_{od}^p .

Each vehicle $v \in S$ has a maximum vehicle capacity Cap_v and associated with t_v , time at which it leaves from its origin o_v .

If vehicle $v \in S$ serves passenger $p \in T$, let $Distance_v^p$ be the distance from vehicle v 's current location (its origin) o_v to passenger's p pick up location o_p , TT_v^p be travel time of vehicle v to reach passenger's p pick up location o_p from vehicle v 's current location (its origin) o_v and WT_v^p be waiting time of passenger p to be picked up by vehicle v .

For a given passenger $p \in T$ and vehicle $v \in S$, we can determine the time flexibility that is required to make a feasible match between them, as follows.

The vehicle v leaves its current location o_v at time t_v , and should arrive at passenger's pick up location o_p between

passenger's earliest pick up time e_p^{dept} and latest pick up time l_p^{dept} , as described in condition (5).

$$t_{req} \leq e_p^{dept} \leq t_v + TT_v^p \leq l_p^{dept} \quad (5)$$

Next, the vehicle v which serves passenger p , should arrive at passenger's drop off location d_p between passenger's earliest drop off time e_p^{arr} and latest drop off time l_p^{arr} , as described in condition (6).

$$e_p^{arr} \leq t_v + TT_v^p + TT_{od}^p \leq l_p^{arr} \quad (6)$$

After time feasibility constraints checked, then we determine trip fare and operational cost for each feasible match, as detailed in the next subsection.

3.2. Fare determination and operational cost

Let p_{km} be m -th passenger of platform k and v_{nl} be l -th vehicle of platform n .

For a given passenger p_{km} and vehicle v_{nl} , the trip fare which should be paid by passenger p_{km} to vehicle v_{nl} is:

1. If $k = n$, that is p_{km} and v_{nl} from the same platform, then

$$F(v_{nl}, p_{km}) = \left(\begin{array}{l} \text{trip fare of platform } n \\ \text{per unit of distance} \end{array} \right) \times \text{Distance}_{od}^p \quad (7)$$

2. If $k \neq n$, that is p_{km} and v_{nl} from different platform (sharing platform), then

$$F'(v_{nl}, p_{km}) = \gamma \times \left(\begin{array}{l} \text{trip fare of platform } n \\ \text{per unit of distance} \end{array} \right) \times \text{Distance}_{od}^p \quad (8)$$

where $1 < \gamma \leq 1 + \Delta$, γ is the "Sharing" multiplier factor, and the value of Δ is determined based on the converting money value of waiting time per unit time, usually the value of Δ less than 50% of initial trip fare.

The operational cost to be incurred by vehicle v_{nl} to serve passenger p_{km} :

$$\text{op. cost}(v_{nl}, p_{km}) = \text{cost per unit of distance} \times (\text{Distance}_v^p + \text{Distance}_{od}^p) \quad (9)$$

Then the revenue (profit) which could be gained by vehicle v_{nl} to serve passenger p_{km} is:

1. If $k = n$, that is p_{km} and v_{nl} from the same platform, then

$$R(v_{nl}, p_{km}) = F(v_{nl}, p_{km}) - \text{op. cost}(v_{nl}, p_{km}) \quad (10)$$

2. If $k \neq n$, that is p_{km} and v_{nl} from different platform (sharing platform), then

$$R(v_{nl}, p_{km}) = \beta \times \left(\begin{array}{l} F'(v_{nl}, p_{km}) \\ \text{op. cost}(v_{nl}, p_{km}) \end{array} \right) \quad (11)$$

where $0.05 \leq (1 - \beta) \leq 0.1$, with $(1 - \beta)$ is the sharing revenue factor taken by initial platform

Next, weight of feasible match is determined based on waiting time of passenger and revenue gained by driver (vehicle), as detailed in subsection 3.3.

3.3. Weight of feasible match

Let $Value_{WT_{v_{nl}}^{p_{km}}}$ be the converting money value of per unit waiting time $WT_{v_{nl}}^{p_{km}}$ of passenger p_{km} to be picked up by vehicle v_{nl} ,

The weight of each feasible match (i, j) , where $v_{nl} = i \in S$ and $p_{km} = j \in T$, is

1. If $k = n$, that is p_{km} and v_{nl} from the same platform, then

$$w_{ij} = w(v_{nl}, p_{km}) = \frac{R(v_{nl}, p_{km})}{Value_{WT_{v_{nl}}^{p_{km}}}} \quad (12)$$

2. If there are two nearest possible vehicle that available to serve passenger p_{km} , where vehicle v_{nl} from the platform n , and $v_{n'l}$ from the platform n' , then

$$w_{ij} = \begin{cases} \frac{\omega \times R(v_{nl}, p_{km})}{Value_{WT_{v_{nl}}^{p_{km}}}} & \text{if } k = n \\ \frac{R(v_{n'l}, p_{km})}{\omega \times Value_{WT_{v_{n'l}}^{p_{km}}}} & \text{if } k \neq n' \end{cases} \quad (13.a)$$

$$w_{ij} = \begin{cases} \frac{\omega \times R(v_{nl}, p_{km})}{Value_{WT_{v_{nl}}^{p_{km}}}} & \text{if } k = n \\ \frac{R(v_{n'l}, p_{km})}{\omega \times Value_{WT_{v_{n'l}}^{p_{km}}}} & \text{if } k \neq n' \end{cases} \quad (13.b)$$

with the parameter value $\omega > 1$.

Here we have waiting time $WT_{v_{nl}}^{p_{km}}$ of passenger p_{km} to be picked up by vehicle v_{nl} is equal to waiting time $WT_{v_{n'l}}^{p_{km}}$ of passenger p_{km} to be picked up by vehicle $v_{n'l}$, that is $WT_{v_{nl}}^{p_{km}} = WT_{v_{n'l}}^{p_{km}}$

But the trip fare $F'(v_{n'l}, p_{km})$ which should be paid by passenger p_{km} to vehicle $v_{n'l}$ is greater than the trip fare $F(v_{nl}, p_{km})$ which should be paid by passenger p_{km} to vehicle v_{nl} , that is $F'(v_{n'l}, p_{km}) > F(v_{nl}, p_{km})$.

Therefore we add parameter ω to the weight so the vehicle will prioritize choosing passenger from the same platform, as in (13.a).

4 Solution method

To determine the shortest path, which is the distance and the travel time from vehicle current location (origin) to passenger's pick up location, and passenger's OD, we use modified Dijkstra with adaptive algorithm based on Wei peng et.al (2012) [5] as described in figure 4 and figure 5. The following data structured which will be used in modified Dijkstra with adaptive algorithm [5] :

- L : the matrix containing the edge weights, where $L[u, v]$ is the weight of edge (u, v) . If the edge does not exist, then $L[u, v] = \infty$;
- D : the distance matrix, where $D[u, v]$ is the distance from the vertex u to v edge (u, v) . If the edge does not exist, then $L[u, v] = \infty$;
- $flag$: the vector to indicate whether the shortest paths from a vertex to other vertices have been calculated. All elements of the vector are set to

zero initially. After the shortest paths for vertex u are calculated, $flag[u]$ is set to 1.

Q : the min-priority queue containing the vertices to be visited. It is the same queue as that used in the classic Dijkstra's algorithm.

Deg : the vector containing the degree of vertices, $deg[i]$ is the degree of the i -th vertex;

$Order$: the vector containing the indices of vertices to be used as sources. $Order[i]$ is the index of i -th source vertex.

Procedure 1: Modified Dijkstra's Procedure

Input: Graph $G = (V, E)$, source s , weight matrix L , distance matrix D , vector $flag$

Output: updated distance matrix D , updated vector $flag$

```

1:  $D[s, s] = 0$ 
2:  $Q = \{s\}$ 
3: while  $Q$  is not empty do
4:    $t = DeQueue(Q)$ 
5:   if  $flag[t] = 1$ , then
6:     for each vertex  $v \in V$  do
7:       if  $D[s, t] + D[t, v] < D[s, v]$  then
8:          $D[s, v] = D[s, t] + D[t, v]$ 
9:       end if
10:    end for
11:  else
12:    for each edge  $(t, v)$  outgoing from  $t$  do
13:      if  $D[s, t] + L[t, v] < D[s, v]$  then
14:         $D[s, v] = D[s, t] + L[t, v]$ 
15:         $Enqueue(Q, v)$ 
16:      end if
17:    end for
18:  end if
19: end while
20:  $flag[s] = 1$ 

```

Fig. 4. Modified Dijkstra's procedure based on Wei peng et.al (2012)

The procedure $Enqueue(Q, v)$ adds a vertex v in the min-priority queue Q . The procedure $DeQueue(Q)$ gets a vertex from the queue Q which has the smallest shortest-path starting from s .

Input: Graph $G = (V, E)$, weight matrix L

Output: distance matrix D

```

1: for each vertex  $v \in V$  do
2:    $flag[v] = 0$ 
3: for each vertex pair  $(u, v)$  do
4:    $D[u, v] = \infty$ 
5: for  $i = 1$  to  $n$  do
6:   Calculate the degree of the  $i$ -th vertex
7: for  $i = 1$  to  $n$  do
8:    $order[i] = i$ 
9: for  $i = 1$  to  $n$  do
10:  for  $j = i + 1$  to  $n$  do
11:    if  $deg[order[j]] > deg[order[i]]$  then
12:       $swap(order[j], order[i])$ 
13:    end if
14:  end for
15:  call the modified Dijkstra's procedure using the source of index  $order[i]$ 

```

Fig. 5. Adaptive algorithm based on Wei peng et.al (2012).

To solve the maximum weighted bipartite matching problem as in (1)-(4), we will use Kuhn Munkres Algorithm (Hungarian Algorithm) [4].

Initial step, let the initial label for each $v_{nl} = i \in S$ and $p_{km} = j \in T$, is

$$L(v_{nl}) = \max\{w(v_{nl}, p_{km}) | p_{km} \in T\} \forall v_{nl} \in S \quad (14)$$

$$L(p_{km}) = 0 \quad \forall p_{km} \in T \quad (15)$$

Then we have $L(v_{nl}) + L(p_{km}) \leq w(v_{nl}, p_{km})$.

Let $E_L = \{(i, j) | L(i) + L(j) = w_{ij}\}$ and

define neighbour of $i \in V$ set for $S \subseteq V$ to be

$$N_i(i) = \{j : (i, j) \in E_L\}, \quad N_i(S) = \cup_{i \in S} N_i(i)$$

Next step, run the procedure of Kuhn Munkres Algorithm as follow:

1. Generate initial labelling L and matching M in E_L
2. If M is perfect, terminate
 Otherwise pick free vertex $i \in S$,
 set $S' = \{i\}, T' = \emptyset$
3. If $N_i(S') = T'$ update labels (forcing $N_i(S') \neq T'$)
 $\alpha_L = \min\{L(i) + L(j) - w_{ij}\}, i \in S', j \notin T'$
 If $u \in S' : L'(u) = L(u) - \alpha_L$
 If $u \in T' : L'(u) = L(u) + \alpha_L$
 Otherwise $L'(u) = L(u)$
4. If $N_i(S') \neq T'$ pick $j \in N_i(S') - T'$
 If j is free, augmenting $i - j$ and go to step 2.
 If j is matched to $j', S' = S' \cup \{j'\}, T' = T' \cup \{j\}$ and go to step 3.

5 Illustrative example

As an illustrative example, which describes a possible scenario that can occur, suppose there are 4 passengers and 4 vehicles in total, where each operator having 2 passenger's request and 2 available vehicle as described in figure 6.

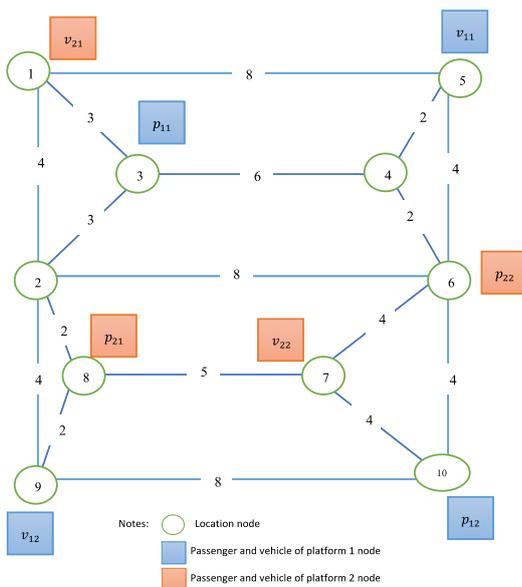


Fig. 6. Illustration of location of passenger and vehicle of platform 1 and 2.

Let p_{11} and p_{12} be two passengers who request ridesourcing services on operator 1, and p_{21} and p_{22} be two passengers who request ride sourcing on

operator 2. Let v_{11} and v_{12} be two available vehicles on operator 1, and v_{21} and v_{22} be two available vehicles on operator 2.

Assume all passengers request vehicle at the same time, let p_{11} pick up location at node 3, and drop off location at node 9; p_{12} pick up location at node 10, and drop off location at node 2; p_{21} pick up location at node 8, and drop off location at node 10; and p_{22} pick up location at node 6, and drop off location at node 1.

By using Modified Dijkstra and Adaptive Algorithm [10] as described on section 4, then we have the following passenger's waiting time for vehicle described in table 1.

Table 1. Passenger's waiting time for vehicle

waiting time				
	v_{11}	v_{12}	v_{21}	v_{22}
p_{11}	8	7	3	10
p_{12}	8	8	16	4
p_{21}	13	2	6	5
p_{22}	4	11	11	4

From the table 1 above, for example, passenger p_{11} has to wait for 8 unit of time if he/she is picked up by vehicle v_{11} , or 7 unit of time if picked up by v_{12} , or 3 unit of time if picked up by v_{21} and 10 unit of time if picked up by v_{22} .

As we can see from table 1, p_{22} 's minimum waiting time is 4 unit of time, which picked up by v_{11} (different platform) or v_{22} (same platform). The minimum waiting for passenger p_{12} is also 4 unit of time if he/she picked up by vehicle v_{22} .

Let F_1 and F_1' be the trip fare of platform 1 per unit distance of distance, where $F_1 = 3/\text{unit}$ of distance for the same platform and $F_1' = 3,25/\text{unit}$ of distance for different platform. Here we have the value of sharing multiplier factor is $\gamma_1 = 1,08$.

Let F_2 and F_2' be the trip fare of platform 2 per unit distance of distance, where $F_2 = 3,5/\text{unit}$ of distance for the same platform and $F_2' = 3,6/\text{unit}$ of distance for different platform, with sharing multiplier factor $\gamma_2 = 1,029$.

Then we have the following trip fare should be paid by the passenger described in table 2.

Table 2. The trip fare should be paid by passenger

Trip Fare				
	v_{11}	v_{12}	v_{21}	v_{22}
p_{11}	21	21	22,75	22,75
p_{12}	33	33	35,75	35,75
p_{21}	32,4	32,4	31,5	31,5
p_{22}	39,6	39,6	38,5	38,5

From the table 2, for example, passenger p_{22} has to paid 39,6 if he/she is picked up by vehicle v_{11} , or 39,6 if picked up by v_{12} , or 38,5 if picked up by v_{21} and 38,5 if picked up by v_{22} . By using condition (7)-(11), with

sharing revenue factor $(1 - \beta) = 0.05$, the revenue gained by the vehicle can be described on table 3 below.

Table 3. Revenue gained by the vehicle

revenue				
	v_{11}	v_{12}	v_{21}	v_{22}
p_{11}	6,00	7,00	12,11	5,46
p_{12}	14,00	14,00	8,31	19,71
p_{21}	9,88	20,33	16,50	17,50
p_{22}	23,37	16,72	16,50	23,50

From the table 3, we have revenue gained by the vehicle v_{22} is 23,5 if it serves passenger p_{22} (same platform), greater than if it serves passenger p_{12} (different platform), which is 19,71.

The weight of feasible match for the illustrative example problem, can be described on table 4 below.

Table 4. Weight of feasible match

weight				
	v_{11}	v_{12}	v_{21}	v_{22}
p_{11}	0,75	1,00	4,04	0,55
p_{12}	1,75	1,75	0,52	4,93
p_{21}	0,76	10,17	2,75	3,50
p_{22}	2,92	1,52	1,50	11,75

By using Kuhn Munkres Algorithm as described on section 4, then we have passenger p_{11} served by vehicle v_{21} , p_{12} served by v_{11} , p_{21} served by v_{12} and p_{22} served by v_{22} , with the maximum weight objective value is 27,71.

6 Conclusion and future research

Based on our study it can be concluded resource sharing (sharing platform) scheme can optimize the matching between driver and rider, which will give benefit for the customers in shorter waiting time, while the benefit for the drivers is optimum income. Furthermore, the weight defined in this study, makes the vehicle (operator) to continue to prioritize providing services to passengers from the same platform.

We hope that the insights generated by our study can be used by ride-sourcing system providers to solve the surge pricing problem which is deemed as less transparent for the customers.

In this paper, the proposed model is only tested using a small example to illustrate a possible scenario that can occur. In future research work, we aim to use larger data to test the proposed model and using appropriate heuristic method to solve the problem.

The authors would like to thank to DRPM UI (Directorate of Research and Community Service Universitas Indonesia) for the research grants through Hibah PITTA 2018.

References

1. L. Rayle, S. Shaheen, N. Chan, D. Dai, and R. Cervero. *App-Based, On-Demand Ride Services: Comparing Taxi and Ridesourcing Trips and User Characteristics in San Francisco*. University of California Transportation Center (UCTC) Working Paper, UCTC- FR-2014-08 (2014)
2. H.Burhan, S. Soehodho and N.Yusuf. *Review on the Development of Ride Sharing System using Online Transportation Service in Jakarta*. Proceeding of 6th IEEE International Conference on Advanced Logistics and Transport (2017)
3. L. Chen, A. Mislove, and C. Wilson. *Peeking Beneath the Hood of Uber*. In Proc. of the ACM Conference on Internet Measurement Conference. ACM (2015)
4. <https://www.cse.ust.hk/~golin/COMP572/Notes/Matching.pdf>
5. W.Peng, X. Hu, F. Zhao, and J. Su. A Fast Algorithm to Find All-Pairs Shortest Paths in Complex Networks. *Procedia Computer Science* **9**, 557 – 566 (2012)
6. D.N. Anderson. “Not just a taxi”? For-profit ridesharing, driver strategies, and VMT. *Transportation* **41**, 1099–1117. <http://dx.doi.org/10.1007/s11116-014-95318>. (2014)
7. M.K. Chen and M.Sheldon. *Dynamic Pricing in a Labor Market: Surge Pricing and Flexible Work On the Uber Platform*. UCLA Anderson School of Management (2016)
8. L. Zha, Y. Yin and Y.Du. *Surge pricing and labor supply in the ride-sourcing market*. *Transportation Research Part B*, <http://dx.doi.org/10.1016/j.trb.2017.09.010> (2017)
9. L. Zha, Y. Yin and Z. Xu. *Geometric matching and spatial pricing in ride-sourcing markets*. *Transportation Research Part C: Emerging Technologies*, Vol. **92**, pages 58-75 (2018)
10. Galichon, A., Hsieh, Y.-W. *A Theory of Decentralized Matching Markets without Transfers, with An Application to Surge Pricing*. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2908532. Accessed Aug., 28, 2017
11. N.A.H. Agatz, A.Erera, M.P.W. Savelsbergh, M.P & X. Wang. Optimization for dynamic ride-sharing: a review. *Eur. J. Oper. Res.* **223**, 295–303 (2012)