Journal of Engineering and Applied Sciences 14 (23): 8891-8898, 2019
ISSN: 1816-949X

Modified k-Medoids based Driver Dispatch Model in City Courier Service

[1]Purba Daru Kusuma and [2]Helmy Widyantara
[1]Department of Computer Engineering, Faculty of Electrical Engineering,
Telkom University, Bandung, Indonesia
[2]Department of Computer Engineering, Faculty of Electrical Engineering,
Institute of Teknologi Telkom Surabaya, Surabaya, Indonesia
purbodaru@telkomuniversity.ac.id, +6282225283186

Abstract: Now a days, city courier service grows fast, especially, in Indonesia. Unfortunately, current city courier service is still inefficient, especially, that is held by online motorcycle taxi companies. Single transaction still means only for single destination. So, delivering many packets to many destinations from single customer is still difficult. Based on this problem in this research, we develop dispatch model that accomodates multi destinations delivery order. By using this model, one transaction may contain more than one destination. In the other side, a driver can deliver many packets simultaneuosly. In this research, we use k-medoids clustering model as a basis model. We improve this k-medoids model to create harmonization among drivers. This model is then implemented into web based simulation application and it is developed by using PHP language. Based on the simulation result, it is proven that this model can give advantages in user's cost reduction and average driver's revenue improvement. The user's cost is reduced to approximately 30%. In the other side, the average driver's revenue increases to approximately 250%.

Key words: k-medoids, city courier service, dispatch system, least cost, telecommunication, PHP language

INTRODUCTION

Now a days, online motorcycle taxi becomes popular in Indonesia. In this country, there are two main players: Go-Jek and Grab. With more than 100 million transactions from 20-25 million monthly users (Potkin, 2018), Go-Jek becomes the market leader. Meanwhile, the Grab application has been downloaded to 139 million devices and it processes more than 6 million ride orders a day (Chandler, 2019). This business has been disrupting conventional taxi industry. Besides claim that million of lower income people have got better income as online taxi driver, the conventional taxi industries become the losers of this new competition (Tjia, 2018).

Besides taxi service, these companies have diversified their business by creating courier service. This diversification has made more business players connect to this environment: from restaurant to e-Commerce merchant. Go-Jek claimed that there are 125,000 food merchants all over Indonesia have been joining its environment (Freischlad, 2018). This new business has increased the driver's utilization in securing transaction and driver's revenue. This business is developed in responding the uncertainty and volatility in taxi service which it meets resistance from conventional players.

Unfortunately, the business model in city courier service that is held by online taxi company is still conventional. Single transaction means a customer creates delivery order for one packet to single destination. This order then is executed by a driver. If customer needs to send more than one packet, they must create more than one order. Then, each order will be executed by single driver.

This existing business model creates inneficiency. Customer spends and wastes time by creating many transactions. Many orders need many drivers. In the other side, drivers in large amount are not always available. So, more delivery orders may increase rejection rate. Large number of transactions also increases transaction cost.

Based on this problem in this research, we develop the combined shipping model for dispatch process between delivery orders and drivers. The purpose is increasing the effieciency which it means reducing customer's cost and in the other side, increasing driver's revenue. This research is also the continuation of our previous researches (Kusuma and Osmond, 2018; Kusuma, 2019).

In this research, we use k-medoids clustering model as our basis model. The reason is that k-medoids method is one popular method in clustering work besides k-means

Corresponding Author: Purba Daru Kusuma, Department of Computer Engineering, Faculty of Electrical Engineering, Telkom University, Bandung, Indonesia, purbodaru@telkomuniversity.ac.id, +6282225283186

method. Some researches have compared the performance between k-means and k-medoids. Zhang *et al*. (2014) have proven that in their research, the k-medoids method outperformed the k-means method. Meanwhile, Aryuni *et al*. (2018) showed that in their research, k-means worked better than k-medoids method.

The k-medoids can also be combined with other methods. The k-medoids which it is basically is non-hierarchical clustering method was also combined with the agglomerative method which it is hierarchical method to improve its effectiveness (Tamura and Miyamoto, 2014). Sabzi *et al*. (2011) have improved the accuracy of the k-medoids clustering method by combining it with fuzzy method, especially, in determining the number of clusters and the initial value of the centroids.

The k-medoids has been implemented in many works. Zhang *et al*. (2014) used k-medoids method in processing traffic flow as part of intelligent transportation system. Aryuni *et al*. (2018) used k-medoids method in segmenting the internet banking customers. The k-medoids method also has been used in data mining in parallel computing environment, especially, that uses Hadoop and MapReduce (Rajendran *et al*., 2018; Zhu *et al*., 2014).

Rather than adopting full basic k-medoids clustering, we improve this clustering method for hamonization. The harmonization is important, so that, variation in number of packets that are executed in single trip among drivers is narrow.

The study organization is as follows. In the introduction, we explain the background, research purpose, method overview and the paper organization. In materials and methods, we explain the method that includes model, formulae and algorithms. In the result and discussion, we explain the implementation of the model into the simulation application, the simulation result and research findings. In the conclusion and future research, we conclude the research and propose future research potentials.

MATERIALS AND METHODS

In this research, we propose driver dispatch model based on clustering approach. We choose k-medoids method as basis method for the clustering process. Meanwhile, we modified the basic k-medoids method to improve equality in packets is distribution among drivers. So, our proposed model is devided into two steps: clustering and equalizing. The main algorithm of this model is shown in algorithm 1.

Algorithm 1; Dispatch model main algorithm:

```
begin
 set(P_user)
 set(P_dest)
 L_clus • clustering (P_user, P_dest)
 L_eq • equalizing (L_clus)
end
```

In algorithm 1, there are some variables and functions that are used. P_{user} is set of users. User is person that has packets that must be delivered. So, user creates delivery order. P_{dest} is set of packet destinations. So, n_{user} is variable that presents the number of users in the system and $n_{totdest}$ is variable that presents the number of total packets.

Every user has several attributes that can be presented as $\{p_{user,x}, p_{user,y}, n_{userpack}\}$. Variable $p_{user,x}$ is the user location in x coordinate. Variable $p_{user,y}$ is user location in y coordinate. Variable $n_{userpack}$ is the number of packets owned by the user. So, the number of total packets that will be delivered in the system can be formalized by using Eq. 1. In Eq. 1, variable i represents the user index:

$$n_{totdest} = \sum_{i=1}^{n_{user}} n_{userpack,i} \quad (1)$$

Variable total number of packets then is used to determine number of drivers that is needed to deliver these packets. Besides the total number of packets, maximum number of packets that can be delivered by a driver ($n_{maxpack}$) is also needed to determine the number of drivers (n_{driver}). The number of drivers is determined by using Eq. 2:

$$n_{driver} = \text{int}\left(\frac{n_{totdest}}{n_{max\,pack}}\right) \quad (2)$$

After this initial variables value is determined, the next step is the clustering process. In this step, k-medoids method is used. The k-medoids clustering algorithm is shown in algorithm 2. In algorithm 2, variable d_{tot} is the total distance between packet destination location and its cluster. Initial d_{tot} is determined by using Eq. 3. In algorithm 2, L_{clust} represents set of links that connect the destination location of the packet to its cluster or central.

Algorithm 2; k-medoids clustering algorithm:

```
begin
 d_tot • set_max_distance (n_totdest, r_city)
 L_clust • set_initial_cluster (P_user, P_dest)
 iterate_cluster (L_clust, P_user, P_dest)
end
```

$$r_{max(0)} = \sqrt[2]{2.\, r_{city}^{\;2}.n_{totdest}} \quad (3)$$

The number of clusters or medoids is equal the number of drivers. In initial clustering process, the medoids position is determined randomly. In k-medoids clustering, medoids are at the nodes or destination position. In this research, packets destination location represents the node. To avoid more than one medoids occupies same nodes, the medoids initial location is determined by using Eq. 4-6. In Eq. 4, variable j represents the node index. In Eq. 5 and 6, variables $p_{x, med, I}$ and $p_{y, med, i}$ represent the location of medoids i.

$$j = \text{int}\left(\frac{n_{totdest}}{n_{driver}}\right) + \text{rand}\left(0, n_{driver}\right) \tag{4}$$

$$p_{x, med, i} = p_{x, dest, j} \tag{5}$$

$$p_{y, med, i} = p_{y, dest, j} \tag{6}$$

The iteration process runs after the initial clustering finishes. During the iteration process, there four processes inside the loop: finding the nearest medoid, relinking nodes and medoids, moving the medoids location and calculating current total distance. The iteration process is shown in algorithm 3.

Algorithm 3; Iteration process algorithm:

```
begin
 run • 1
 while run = 1 do
 begin
  find_nearest_medoid ()
  relinking ()
  moving_medoid ()
  calculate_current_total_distance ()
 end
end
```

The nearest medoids finding process basically is process to find the nearest medoids among medoids in the system to its location. This process is determined by using Eq. 7 and 8. In Eq. 7, it is shown that selected medoids for node i is medoids in the system where its location is the nearest to the destination i. In Eq. 8, the distance between medoids and node uses Euclidean distance:

$$m_{(sel,i)} = m | m \in M \wedge \min\left(d_{(p_{dest,i})}, m\right) \tag{7}$$

$$d\left(p_{dest}, m_j\right) = \left\| p_{dest,i}, m_j \right\| \tag{8}$$

After the linking process finishes, the next step is moving the medoid. The new position of the medoids is determined randomly among nodes that linked to the medoid. After the medoids are moved to their new location, the next step is calculating the total distance. In the iteration, the total distance is calculated by using Eq. 9. In Eq. 9, variable $p_{medsel, i}$ is the location of the selected medoid that is linked to the node I:

$$d_{tot} = \sum_{i=1}^{n_{totdest}} d\left(p_{dest,i}, p_{med_{sel},i}\right) \tag{9}$$

After the total distance is calculated, then this distance is compared with the previous total distance. If the current total distance is still lower than the previous total distance, the iteration continues. Otherwise, the iteration stops. This process is formalized by using Eq. 10:

$$run = \begin{cases} 0, d_{tot}(t) < d_{tot}(t\text{-}1) \\ 1, \text{else} \end{cases} \tag{10}$$

After the clustering process is done, the next process is equalizing the packets distribution. In basic clustering process, number of packets that must be delivered by a driver may be different to other drivers. A driver may handle lots of packets because the destination locations are more concentrated rather than other packets.

The concept of equalization is distributing packets from drivers that handle more packets to drivers that handle fewer packets. The threshold is the maximum number of packets that are handled by single driver ($n_{maxpack}$) plus tolerance (•). Besides this variable, some tolerance is added.

Driver that must transfered his packets to other driver is driver that handles packets more than the threshold. This process is formalized by using Eq. 11. In Eq. 11, c_{out} is driver that must transferred some of his packets to other drivers. Variable n_{cpack} is number of packets that are handled by the driver:

$$c_{out} = c | n_{cpack} > n_{maxpack} + \Delta \tag{11}$$

Meanwhile the packet that must be transferred from this driver is packet that its destination location is the farthest from the driver's medoid. This packet determination is formalized by using Eq. 12 and 13. The iteration process will run until the number of packets that are handled by this driver is lower than or equal to the threshold. In Eq. 12, g_{out} is packet that must be transferred from driver c_{out}. Meanwhile, $p_{med, cout}$ is the medoids location of the driver c_{out}. Variable d_{out} is the distance between packet that must be transferred and the medoids of the driver that handle this packet.

$$g_{out} = g|g = c_{out} \Lambda \max(d_{out}) \quad (12)$$

$$d_{out} = \left\| p_{dest}(g_{out}) \text{-} p_{med, cout} \right\| \quad (13)$$

In this process, driver that his medoids is the nearest to the destination location of the packet that is transferred will be prioritized. This process is formalized in Eq. 14 and 15. Variable c_{in} represents the driver that receives handover. Variable d_{in} is the distance between destination location of the transferred packet and the medoids of the driver that receive this handover:

$$c_{in} = c|\min(d_{in}) \Lambda n_{cpack} < n_{maxpack} \quad (14)$$

$$d_{in} = \left\| p_{dest}(g_{out}) \text{-} p_{med, cin} \right\| \quad (15)$$

RESULTS AND DISCUSSION

This model then is implemented into scheduled shipping simulation application. This application is developed by using PHP language, so that, it is web based application. The simulation result then is stored in MySQL database.

In this application, the world is a virtual city. The courier warehouse is in the center of the city. The city shape is circle with specific radius. The city radius is presented in kilometer. The courier warehouse is in the center of the city. In the beginning of simulation, some users are generated. The user's position is generated randomly and it follows uniform distribution. The user's position in the city is determined by using Eq. 16-19:

$$r_{user,i} = \frac{\text{random}(0, r_{city}.1000)}{1000} \quad (16)$$

$$\alpha_{user,i} = \text{random}(0.360) \quad (17)$$

$$p_{x,\ user,\ i} = r_{user,\ i}.\cos(\alpha_{user,\ i}) \quad (18)$$

$$p_{y,\ user,\ i} = r_{user,\ i}.\sin(\alpha_{user,\ i}) \quad (19)$$

After the users are generated, the next step is generating the packet destination. Similar to user generation, packet destination is also generated randomly and it follows uniform distribution. This process is formalized by using Eq. 20-23. Variable $r_{dest,\ j}$ is the distance of the packet j destination location from the central warehouse and it is presented in kilometer.

Table 1: Default value of adjusted variables

Variables	Default values
City radius (r_{city})	10 (km)
$n_{peruser}$	50 (units/user)
n_{user}	5 (users)
$n_{maxpack}$	5 (units/driver)
•	0 (units/driver)

Variable $•_{dest,\ j}$ is the angle of the packet j destination location related to the central warehouse and it is presented in degree:

$$r_{dest,j} = \frac{\text{random}(0, r_{city}.1000)}{1000} \quad (20)$$

$$\alpha_{dest,j} = \text{random}(0.360) \quad (21)$$

$$p_{dest,j} = r_{dest,j}.\cos(\alpha_{dest,j}) \quad (22)$$

$$p_{dest,j} = r_{dest,j}.\sin(\alpha_{dest,j}) \quad (23)$$

After the users and packets are generated, the next process is running the dispatch process. In this process, some variables are set manually, so that, it can be adjusted to analyze the system response. These adjusted variables include: city radius (r_{city}), number of packets per user ($n_{peruser}$), number of users (n_{user}), maximum number of packets per driver ($n_{maxpack}$) and tolerance (•). In this simulation, all users have same number of packets (Table 1).

Simulation and result analysis: In this research, we analyze the system behavior by running the courier service simulation. The adjusted variables become the input. There are four simulation groups based on the changing value of the input: city radius, number of packets per user, number of users and tolerance. While the value of one variable changes, the value of the other variables is set as their default value. In this simulation, we also compare the system performance with the performance of the courier service when it uses conventional method. Conventional method means one driver only delivers one packet.

The system response is analyzed by observing its outputs. In this research, the outputs include: total conventional driver travel distance ($d_{totdrivercon}$), total k-medoids based driver travel distance ($d_{totdrivermed}$), average conventional driver travel distance ($d_{avdrivercon}$), average k-medoids based driver travel distance ($d_{avdrivermed}$), minimum k-medoids based driver travel distance ($d_{mindriver}$), maximum k-medoids based driver travel distance ($d_{maxdriver}$),

Table 2: Simulation result based on various city radius

r (km)	$d_{totdrivercon}$ (km)	$d_{totdrivermed}$ (km)	$d_{avdrivercon}$ (km)	$d_{avdrivermed}$ (km)	$d_{mindriver}$ (km)	$d_{maxdriver}$ (km)	r_{tot}	r_{av}	r_{minmax}
5	809.80	601.80	3.240	12.04	0.38	30.26	0.743	3.716	0.013
6	958.80	685.80	3.840	13.72	0.53	34.89	0.715	3.576	0.015
7	1,122.80	818.80	4.490	16.38	0.89	40.19	0.729	3.646	0.022
8	1,348.80	971.40	5.400	19.43	1.60	46.18	0.720	3.601	0.035
9	1,399.60	1,061.60	5.600	21.23	0.80	53.02	0.759	3.793	0.015
10	1,669.00	1,185.60	6.680	23.71	0.76	56.66	0.710	3.552	0.013
11	1,853.40	1,376.60	7.410	27.53	1.15	62.52	0.743	3.714	0.018
12	1,997.40	1,414.60	7.990	28.29	1.58	68.79	0.708	3.541	0.023
13	2,103.20	1,587.40	8.410	31.75	1.34	78.49	0.755	3.774	0.017
14	2,338.80	1,630.20	9.360	32.60	1.23	82.74	0.697	3.485	0.015
15	2,550.00	1,810.83	10.200	36.22	1.91	88.20	0.710	3.551	0.022

Table 3: Simulation result based on various number of packets per user

$n_{peruser}$ (units)	$d_{totdrivercon}$ (km)	$d_{totdrivermed}$ (km)	$d_{avdrivercon}$ (km)	$d_{avdrivermed}$ (km)	$d_{mindriver}$ (km)	$d_{maxdriver}$ (km)	r_{tot}	r_{av}	r_{minmax}
10	330.80	239.00	6.62	23.90	3.92	43.55	0.722	3.612	0.090
20	681.40	502.40	6.81	25.12	2.91	51.70	0.737	3.687	0.056
30	1,036.20	722.80	6.91	24.09	1.97	54.63	0.698	3.488	0.036
40	1,350.00	897.00	6.75	22.43	1.35	51.93	0.664	3.322	0.026
50	1,761.20	1,208.20	7.04	24.16	0.89	60.85	0.686	3.430	0.015
60	1,892.80	1,434.80	6.31	23.91	1.00	61.03	0.758	3.790	0.016
70	2,231.00	1,637.60	6.37	23.39	0.98	58.34	0.734	3.670	0.017
80	2,739.40	1,909.00	6.85	23.86	0.62	60.30	0.697	3.484	0.010
90	3,098.50	2,170.17	6.89	24.11	0.48	62.41	0.700	3.502	0.008
100	3,257.60	2,363.60	6.52	23.64	0.50	61.88	0.726	3.628	0.008

total travel distance ratio (r_{tot}), average travel distance ratio (r_{av}), minimum-maximum travel distance ratio (r_{minmax}).

The first simulation group is analyzing the relation between city radius and the simulation outputs. In this simulation, the city radius ranges from 5-15 km. The city radius step size is 1 km. There are five simulations for every city radius value. The simulation result is shown in Table 2.

In Table 2, it is shown that the the city radius has positive correlation with some outputs: total drivers travel distance and average drivers travel distance. This condition occurs when system implements both conventional method or modified k-medoids based method. Especially, when system implements modified k-medoids method, city radius also has positive correlation with the maximum driver travel distance and ratio between minimum and maximum driver travel distance. Meanwhile, there is not any correlation between city radius and minimum driver travel distance or ratio between minimum and maximum driver travel distance.

Comparing between modified k-medoids based method and conventional method, the analysis is as follows. In any city radius, when system implements modified k-medoids based method, the total drivers travel distance is lower than when system implements conventional method. Meanwhile, the average driver travel distance is higher when system implements modified k-medoids based method rather than when system implements conventional method.

As the city radius does not affect the r_{tot} and r_{av}, the analysis is as follows. Modified k-medoids based method can reduce the total drivers travel distance approximately 30% rather than the conventional method. In the other side in any city radius, the average driver travel distance is 3.5 times when system implements modified k-medoids based method rather than conventional method. So, it can be said that in any city radius, users pay lower delivery cost and in the other side, driver gets higher revenue.

The second simulation group is analyzing the relation between number of packets per user and the simulation outputs. In this simulation, the number of packets per user ranges from 10-100 units. The step size of the number of packets per user is 10 packets. There are five simulations for every number of packets per user value. The simulation result is shown in Table 3.

Table 3 shows that the increasing of the number of packets per user has positive correlation with some outputs: total driver travel distance both in modified k-medoids based method and conventional method and in maximum driver travel distance when system implements modified k-medoids based method. The number of packets does not affect the average driver travel distance, the total drivers distance ratio and the average driver distance ratio. Meanwhile, the number of packets has negative correlation with the minimum driver travel distance and the ratio between minimum and maximum value of the driver travel distance.

Comparing between the modified k-medoids based method and the conventional method, the result is as follows. The modified k-medoids based method creates

Table 4: Simulation result based on various number of users

n_{user} (persons)	$d_{totdrivercon}$ (km)	$d_{totdrivermed}$ (km)	$d_{avdrivercon}$ (km)	$d_{avdrivermed}$ (km)	$d_{mindriver}$ (km)	$d_{maxdriver}$ (km)	r_{tot}	r_{av}	r_{minmax}
1	337.60	247.80	6.75	24.78	5.02	44.89	0.734	3.670	0.112
2	666.60	484.60	6.67	24.23	2.51	53.75	0.727	3.635	0.047
3	1,134.00	703.60	7.56	23.45	1.98	55.10	0.620	3.102	0.036
4	1,380.80	969.40	6.90	24.23	1.20	56.99	0.702	3.510	0.021
5	1,729.60	1,275.80	6.92	25.52	1.30	61.07	0.738	3.688	0.021
6	2,029.60	1,452.00	6.77	24.20	0.56	59.65	0.715	3.577	0.009
7	2,272.80	1,613.40	6.49	23.05	0.70	60.97	0.710	3.549	0.011
8	2,742.40	1,949.20	6.86	24.37	0.93	61.45	0.711	3.554	0.015
9	3,158.20	2,089.00	7.02	23.21	0.80	62.33	0.661	3.307	0.013
10	3,505.60	2,439.40	7.01	24.39	0.30	61.62	0.696	3.479	0.005

Table 5: Simulation result based on various maximum packet tolerance

• (unit)	$d_{totdrivercon}$ (km)	$d_{totdrivermed}$ (km)	$d_{avdrivercon}$ (km)	$d_{avdrivermed}$ (km)	$d_{mindriver}$ (km)	$d_{maxdriver}$ (km)	r_{tot}	r_{av}	r_{minmax}
0	1,620.60	1,238.00	6.48	24.76	0.85	56.94	0.764	3.820	0.015
1	1,929.60	1,203.80	7.72	24.08	0.00	67.46	0.624	3.119	0.000
2	1,685.80	1,186.60	6.74	23.73	0.00	76.48	0.704	3.519	0.000
3	1,788.00	1,173.60	7.15	23.47	0.00	87.81	0.656	3.282	0.000
4	1,710.40	1,170.60	6.84	23.41	0.00	92.44	0.684	3.422	0.000
5	1,905.60	1,137.40	7.62	22.75	0.00	86.33	0.597	2.984	0.000

lower total driver travel distance rather than the conventional method. In the other side, the modified k-medoids based method creates higher average driver travel distance rather than conventional method. These advantages remain stable due to the increasing of the number of packets per user. Similar to the city radius, the reduction in total driver travel distance is approximately 30% and the increasing of the average driver travel distance is approximately 250%.

The ratio of the minimum driver travel distance to the maximum driver travel distance increases due to the increasing of the number of packets per users. The reason is as follows. Due to the increasing of the number of packets per user, the minimum driver travel distance decreases while the maximum driver travel distance increases. It means that the revenue disparity among drivers increases due to the increasing of the number of packets per user.

The third simulation group is analyzing the relation between number of users and the simulation outputs. In this simulation, the number of users ranges from 1-10 persons. The step size of the number of users is 1 person. There are five simulations for every number of users. The simulation result is shown in Table 4.

Table 4 shows that due to the increasing of the number of users, the total driver travel distance increases too. This condition occurs both when the system implements the modified k-medoids based method or the conventional method. Besides, the maximum driver travel distance also increases. Meanwhile, the average driver travel distance remains fluctuating. This condition occurs both when system implements modified k-medoids based method or conventional method.

The modified k-medoids based method performs better than the conventional one. Based on the r_{tot}, the modified k-medoids based method can reduces the total driver travel distance approximately 30%. Meanwhile, based on the r_{av}, the average driver travel distance is approximately 250% higher when system implements modified k-medoids based method. These values remain stable due to the increasing of the number of users.

Disparity between minimum and maximum driver travel distance grows due to the increasing of the number of users. When the number of users increases, the minimum driver travel distance reduces. Meanwhile, the maximum driver travel distance increases.

The fourth simulation group is analyzing the relation between tolerance and the simulation outputs. In this simulation, the tolerance range from 0-5 persons. The step size of tolerance is 1 unit. There are five simulations for every tolerance. The simulation result is shown in Table 5.

Table 5 shows that due to the increasing of the tolerance, there is dynamics in total driver travel distance and average driver travel distance. When system implements modified k-medoids based method, the total driver distance reduces. Meanwhile, the total driver distance remains stable when system implements conventional method. The average driver travel distance remains stable when system implements modified k-medoids method or conventional method.

Due to the increasing of the tolerance, modified k-medoids based method performs better than conventional method both in total driver travel distance and average driver travel distance. Modified k-medoids based method reduces the total driver travel distance approximately 30%. Meanwhile, modified k-medoids based method creates 250% higher in average driver travel distance.

Comparing to the research purpose in developing dispatch model in city courier service that can be more efficient rather than the existing system, modified k-medoids based method has met it. System is more efficient, if it can reduce the users delivery cost and or it can increase the driver's revenue. User's delivery cost can be related to the total drivers travel distance. Lower total driver travel distance means lower user's delivery cost. In the other side, driver's revenue can be related to the average driver's travel distance. Higher average driver's travel distance means higher driver's revenue. The modified k-medoids based method has met both these criteria.

The problem that rises is that there is disparity among drivers. Although, harmonization has been implemented in the number of packets, disparity in travel distance between driver with the lowest travel distance and driver with highest travel distance is wide. It is because some drivers may deliver packets in a group that their location is close to each other. Meanwhile, other drivers may deliver packets in a group that their location is not close to each others.

Compared with our previous works by Kusuma and Osmond (2018) and Kusuma (2019), the result is as follows. This modified k-medoids based model performs better than our previous model (Kusuma and Osmond, 2018). In this current research, the user's cost is reduced to approximately 30% and this value tends to stable in any adjusted parameters value. In the other side by using our previous research, the user's cost is reduced from approximately 6-40%. it is depended on the model that is chosen and the number of packets.

In the other side when this current researech is compared with our previous research Kusuma (2019) in user's cost reduction, the result is as follows. When the number of packets is low, this current research performs better. Meanwhile during the increasing of the number of packets, our previous model tends to be more efficient. When the number of packets is high, our previous research performed far better than this current model.

CONCLUSION

In this research, we have developed the modified k-medoids based dispatch model for city courier service. This model has also been implemented in city courier service simulation. Based on simulation result, it is shown that this model performs better than the existing conventional model both in user's total cost and average driver's revenue. By using modified k-medoids based model, user's cost is reduced to approximately 30% and average driver's revenue increases to approximately 250%.

Research in city courier service is still challenging due to the rise up of this industry. Meanwhile, the business model is still in the making and it is far from stable. Many improvements are still needed, so that, sustainable environment can be achieved. The keyword is giving benefits for the stakeholders: customers, drivers and company.

ACKNOWLEDGEMENT

We thank to Telkom University for funding this research and Institut Teknologi Telkom Surabaya for the collaboration in this research.

REFERENCES

Aryuni, M., E.D. Madyatmadja and E. Miranda, 2018. Customer segmentation in XYZ bank using k-means and k-medoids clustering. Proceedings of the 2018 International Conference on Information Management and Technology (ICIMTech), September 3-5, 2018, IEEE, Jakarta, Indonesia, ISBN: 978-1-5386-5822-2, pp: 412-416.

Chandler, C., 2019. Grab vs. Go-Jek: Inside Asia's battle of the super apps. Fortune, New York, USA. https://fortune.com/longform/grab-gojek-super-apps/

Freischlad, N., 2018. Frontiers in food delivery: Go-food versus grab food in Indonesia. Krasia Plus PTE Ltd., Singapore. https://kr-asia.com/frontiers-in-food-delivery-go-food-versus-grabfood-in-indonesia

Kusuma, P.D. and A.B. Osmond, 2018. Round robin-least cost combined model in city courier scheduled shipping service. J. Eng. Appl. Sci., 13: 6181-6187.

Kusuma, P.D., 2019. Comparison between least effort and random walk in combined shipping service. J. Eng. Appl. Sci., 14: 130-138.

Potkin, F., 2018. Indonesia's go-jek close to profits in all segments, except transport: CEO. Reuters, Canary Wharf, London, UK. https://www.reuters.com/article/us-indonesia-gojek-interview/indonesias-go-jek-close-to-profits-in-all-segments-except-transport-ceo-idUSKBN1L20SI

Rajendran, D., S. Jangiti, S. Muralidharan and M. Thendral, 2018. Incremental mapreduce for k-medoids clustering of big time-series data. Proceedings of the 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), May 11-12, 2018, IEEE, Tirunelveli, India, ISBN: 978-1-5386-3571-1, pp: 1143-1146.

Sabzi, A., Y. Farjami and M. ZiHayat, 2011. An improved fuzzy k-medoids clustering algorithm with optimized number of clusters. Proceedings of the 2011 11th International Conference on Hybrid Intelligent Systems (HIS), December 5-8, 2011, IEEE, Melacca, Malaysia, ISBN:978-1-4577-2151-9, pp: 206-210.

Tamura, Y. and S. Miyamoto, 2014. Two-stage clustering using one-pass k-medoids and medoid-based agglomerative hierarchical algorithms. Proceedings of the 2014 Joint 7th and 15th International Conference on Soft Computing and Intelligent Systems (SCIS) and Advanced Intelligent Systems (ISIS and SCIS), December 3-6, 2014, IEEE, Kitakyushu, Japan, pp: 484-488.

Tjia, B., 2018. The damage of ride-hailing disruption to Indonesia's Taxi Industry. A Medium Corporation, https://medium.com/@bennytjia/ride-hailing-disruption-of-indonesias-taxi-industry-how-big-is-the-damage-ed1afab71299

Zhang, T., Y. Xia, Q. Zhu, Y. Liu and J. Shen, 2014. Mining related information of traffic flows on lanes by k-medoids. Proceedings of the 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), August 19-21, 2014, IEEE, Xiamen, China, pp: 390-396.

Zhu, Y.T., F.Z. Wang, X.H. Shan and X.Y. Lv, 2014. k-medoids clustering based on MapReduce and optimal search of medoids. Proceedings of the 2014 9th International Conference on Computer Science & Education, August 22-24, 2014, IEEE, Vancouver, British Columbia, Canada, ISBN:978-1-4799-2949-8, pp: 573-577.