Advanced Traveler Information Systems: Itinerary Optimisation Using Orienteering Problem Model and Genetic Algorithm

I Wayan A K Yoga
Departement of Information Systems
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
aka.yogaa@gmail.com

Arif Djunaidy
Departement of Information Systems
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
adjunaidy@is.its.ac.id

Wiwik Anggraeni
Departement of Information Systems
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
wiwik@is.its.ac.id

Ahmad Muklason
Departement of Information Systems
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
mukhlason@is.its.ac.id

Faizal Mahananto
Department of Information Systems
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
faizal@is.its.ac.id

Edwin Riksakomara
Department of Information Systems
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
erk@is.its.ac.id

Nisa D Angresti
Department of Information Systems
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
nisa.dwi.angresti16@mhs.is.its.ac.id

Sasmi Hidayatul Y T
Department of Information Systems
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
sasmi16@mhs.is.its.ac.id

Vicha Azthanty Supoyo
Department of Information Systems
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
vicha.supoyo16@mhs.is.its.ac.id

***Abstract*— Traffic congestion is one of common urban problems. One of the main factors of congestion is the significant increase in the number motor vehicle growth that is not proportional with the growth of roads. To reduce traffic congestion, the government of Surabaya has made various efforts, one of which is the plan to encourage more people to use the public transportation system. To support the system a reliable transportation management system namely Surabaya Intelligent Transport Systems (SITS) is under development.**

As a part of SITS, this paper presents an advanced traveler information system which helps visitors planning their itinerary using public transport during their visit in the city. The problem is modeled as an orienteering problem (OP) and solved using genetic algorithm (GA). The model was developed upon six routes of 'Angkot' which has been chosen. Travel time data is obtained through Google Maps.

The contributions of this paper are two folds. First, we provide a public two datasets that is the dataset of travel time and score where each form of matrix. Upon the public datasets, future research, especially in the area of orienteering and optimisation problems could be encouraged. Second, initial algorithm based on the Genetic Algorithm to solve the problem. The computational experiments showed that the number of generations, the number of populations, the cross-over probability, and the mutations probability play important role in the performance of the proposed algorithm. It is expected that this research can help the development of Surabaya Intelligent Transport System and the implementation of public transport revitalization in Surabaya.

Keywords*— *optimization, public transportation, orienteering problem model, genetic algorithm

I. INTRODUCTION

Traffic congestion in large cities in Indonesia such as Jakarta and Surabaya is a serious problem that must be addressed by Government. Traffic congestion affect economic growth in large cities [1] [2]. In Surabaya, losses could reach 1 trillion per day in the year 2010 [3]. One of the main factors of the severe congestion is the increase in growth vehicles that are not comparable to the growth of route. To reduce the number of vehicles, the Government should be able to attract the interest of the public to use public transport. Public transport that is provided should be convenient, cheap, fast, and timely.

The rapid advancement of information technology is already utilized by Surabaya's Governments to support the development of good public transportation system i.e. Surabaya Intelligent Transport Systems (Surabaya ITS or SITS) [4]. In the developed countries like the United States, Japan, and Singapore, ITS have been well-implemented for traffic and transportation management [5]. However, in Surabaya, the implementation of ITS is still in its infancy stage, e.g. monitoring traffic congestion in real time using CCTV [4]. To support the implementation of ITS, this paper presents one of applications within and Advanced Traveler Information Systems (ATIS) which is under the bigger umbrella term, ITS. The final goal of the ATIS is an intelligent system that can provide information and advice on to the user of the optimum route for public transport route to commute from one place to the another.

The public transportation mode studied in this research is a van-based transport system, which is in Surabaya commonly called Mikrolet or Angkot, according to plan revitalization using public transportation in Surabaya. Angkot is a favorite public transportation in Surabaya because of its cheap fare and the number of fleets that cover the most nodes in the city [6]. This research aim at supporting the implementation of ATIS as a part of ITS in Surabaya, by providing model and solution in the form of recommendation of travel route to people who want to use public transportation.

The problem is modelled as an Orienteering Problem (OP). This model is used for the case of optimization with condition of available time limitation [7]. The first model of OP is used for travelling salesman problem that do not have enough time to visit all the cities [14]. Therefore, the model aims at maximizing total sales in cities that are considered to have high profits during such limited time.

Other problems that could be solved as OP are travel planning as mentioned in [14][15][16], where travelers may not be able to visit all the spots they like due to the limited time. The use of OP has also been carried out in [8]. This research has the same problem characteristic that is maximizing the value obtained by user, the time limit/cost, and not all nodes (i.e. places) need to be visited.

The search for solutions for route recommendations in this study was proposed using Genetic Algorithm (GA). This algorithm has been widely used in solving optimization problems, and there have been several studies using this algorithm to find solutions of the OP model as in [18] and [19]. Based on the advantages of OP and GA from previous research, this study employed the OP as model and GA to solve the model. The contribution of this research lies on new open public dataset and the modification of the algorithm.

II. Literature Review

A. Orientaring Problem

The Orienteering Problem (OP) is derived from an orienteering sport game where each player visits the provided posts (node) and returns to the game's starting post within a specified time. Each post has a different score and the goal of the game is to collect score as many as possible. OP method aims at finding the best route based on the selected nodes (place, locations) in order to maximize the number scores, while getting the minimum time. A score in this sense, could represent the advantage or satisfaction level of of user when visiting the node. Therefore, OP could be seen as a combination of Knapsack Problem (KP) and Traveling Salesperson Problem (TSP). In OP, each node is not necessary to be visited, but to visit the number of nodes as many as possible within the provided time limit[7].

OP can be formulated as integer programming. Given a decision variable $x_{ij} = 1$ if the visit to node *i* is followed by visit to the node *j*, and $x_{ij} = 0$ if it is not and variable u_i as the position of node *i* in the route, the objective function the constraints can be formulated as follow:

$$Max \sum_{i=2}^{N-1} \sum_{j=2}^{N} S_i x_{ij}, \quad (1)$$

$$\sum_{j=2}^{N} x_{1j} = \sum_{i=1}^{N-1} x_{iN} = 1, \quad (2)$$

$$\sum_{i=1}^{N-1} x_{ik} = \sum_{j=2}^{N} x_{kj} \leq 1; \forall k = 2, \ldots, N-1, \quad (3)$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^{N} t_{ij} x_{ij} \leq T_{max}, \quad (4)$$

$$2 \leq u_i \leq N; \forall i = 2, \ldots, N, \quad (5)$$

$$u_i - u_j + 1 \leq (N-1)\left(1 - x_{ij}\right); \forall i, j = 2, \ldots, N, \quad (6)$$

The objective function of the OP as given in Equation (1) is to maximize the total score, subject to hard constraints: to ensure the path starts at node 1 and ends at node as in Equation (2); to ensure that all nodes in the path are connected and each node is visited at most once as in Equation (3); and to ensure that the amount of travel time does not exceed the time limit (Tmax) as in Equation (4). The constraints in Equations (5) and (6) are to prevent the existence of subtours, i.e. conditions where the path starts and ends at the same point (forming the circuit) [7].

B. Genetic Algorithm

Genetic Algorithm (GA) is a meta-heuristics algorithm used to solve search problem in optimization problem. This algorithm has principles imitating the biological processes of natural selection and evolution. This algorithm works with a population of individuals in which each individual represents an alternative solution to the problem. In this algorithm, the quality of individual is represented by a fitness value where that value will be used to find the best solution of the problem.

In a population, individuals have the probability to reproduce through cross-over mechanism with other individuals. From this cross-over mechanism a new individual is generated. The new individual carries some of its parent traits. The unselected individuals in the process of reproduction will die by itself. Therefore, individual with good characteristics will fit in the population and in turn will be cross-overed with other good individuals. In GA, the typical processes include initial population generation, fitness evaluation, individual selection, cross-over, and mutation. Generally, the process in Genetic Algorithm can be seen in Fig. 1 [9].

```
Initialize population P of size λ
Evaluate λ individuals in P
While not termination do {
        Select 2*μ individuals from P
        Crossover individuals to produce
         μ offspring
        Mutate some individuals in μ
        Add μ offspring to λ individuals in P
         Evaluate (λ + μ) individuals in P
        Select λ individuals from (λ + μ) individuals in P }
End While
End Algorithm
```

Fig. 1 Genetic Algorithm Pseudocode

C. Dijkstra's Algorithm

Dijkstra's algorithm was discovered by E. W. Dijkstra in 1959 to solve minimal spanning tree problem and a shortest path problem [10]. This algorithm is used to solve the problem of finding the shortest route from the starting node to all destination node in a weighted graph (directed or undirected). The condition in using this algorithm is that the graph used should not have negative weights (e.g. negative travel time) in each interconnecting line between node [11].

In general, the steps of finding the minimum travel time using the Dijkstra algorithm [12] are as follow: (1) Create a set (e.g. sptSet) that stores the path of the nodes that belong to the shortest path tree, i.e. the path with the minimum travel time calculated from the source node. Initially, sptSet is an empty set. (2) Set the initial trip time value to all nodes in the Graph. The destination node will be infinite while the source node will be 0. (3) Select a node that does not exist in sptSet and has a minimum travel time value (e.g. node u) and enter the u node into sptSet. (4) Update the travel time value of all nodes adjacent to node u. To update the travel time value, iterate to all adjacent nodes. For each adjacent node (e.g. node v), if the sum of the travel time of node u (starting node) with the travel time of node u to v, less than the the prior travel time of v, the value of travel time will be updated. (5) If the minimum travel time value of node v has been found, then it is inserted into sptSet. (6) Repeat steps 3 to 5 until all nodes have been entered the set. The value of the source node is taken to each destination node.

D. Related Works

In [20], GA was used to solve Team Orienteering Problem of collecting rubbish bin problem in a waste managament company. The objective is to minimise the total travel time and in turn to minimise cost. In [21], GA was employed to solve orienteering problem of a large transportation network with 900 nodes. The result showed that the proposed algorithm is effective. On the other hand, in [8], orienteering problem model was used to optimise tourist itinerary planning, using iterative local search combined with hill climbing algorithm. Compared to [8], this study is different from [8] in terms of the problem instance datasets and the algorithm to solve the problem.

III. METHODOLOGY

The methodology of the research shown in Fig. 2.

A. Problem Identification

This stage is carried out conducting deep analysis on the existing problem of the public transport system in the city of Surabaya. The problem is then modelled as an orienteering problem. The output of this is model that further will be solved using the proposed algorithm.

B. Literature Review

The literature study was conducted by collecting references such as bibliography, previous research, and related documents supporting this research.

C. Data Collection

In this stage, the existing real-world route using Angkot are collected. The data used in this research are six routes of Angkot in Surabaya, namely Kalimas Barat/Petekan-Manukan Kulon with route code of DP, Dukuh Kupang-Benowo with route code of I, Kalimas Barat-Bratang with route code of Q, Joyoboyo-Tubanan-Manukan with route code of TV2, Kalimas Barat - Benowo with route code of Z, Benowo-Ujung Baru with route code of Z1.

Based on these six routes of the Angkot collected from the Department of Transportation of Surabaya and Google Maps application. The score data is counted based on the number of Angkot passing through each node and also the travel time to each stopping point of the Angkot.

D. Pre-processing Data

Pre-processing data or data transformation aims to convert the raw data into qualified data. The data is transformed into the format that could be easily read by system. The result of this data transformation (score and search result of travel time data) will be a matrix.

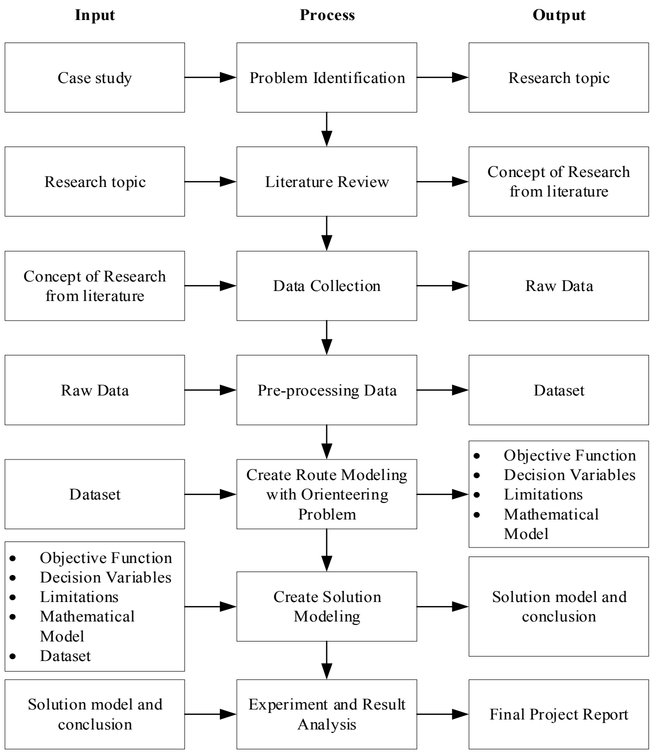

Fig. 1 Reseaarch Methodology

E. Route Modeling with Orienteering Problem

In this stage, the problem is converted into mathematical models. It also includes determining the initial and final nodes, as well as the arc value of the OP model. The nodes represent each of the Angkot stops and the Arc values represent the travel time between one node and the other. Since in the existing system, the Angkot could stop anywhere, in this study artificial nodes are created. Assuming the value of travel time is proportional to the distance, which is adapted from the Decision of the Director General of Land Transportation no. 271 / HK.105 / DJRD / 96, the distance between the nodes (arc values) should not be less than two hundred meters (> 200 m). The distance between nodes (arc values) should also not be more than one kilometer (<1 km) so that roads that have lengths longer than one kilometer will have two or more nodes. It is also assumed that each Angkot only stops at the specified node. The score for each node is the number of Angkot passing through each node. The more Angkot passes the more chance of getting an Angkot. This is adapted to the users of the Surabaya society and the visitors especially who want to get around visiting many places in Surabaya using public transportation. By maximizing the chances of getting an Angkot, it can indirectly reduce the waiting time.

F. Solution Modeling

At this stage the solution is made from the previous OP model. This stage is divided into two main stages, namely: (1) Update the search results between nodes of the process of pre-processing data by finding the shortest route. (2) Prepare the modeling solution using Genetic Algorithms according to the pseudocode in Fig 1. The output of the algorithm is the best route.

IV. IMPLEMENTATION

A. *Travelling Time and Score Matrix*

Travelling time and scores matrices are the input of the algorithm. The travel time matrix from one node to another node consists of 191 columns and 191 rows representing the position of each node according to the network model to the six angkot routes in this study. Each cell contains the travel time from row to column. To be more easily understood, TABLE 1 is an example of the travel time matrix in the form of an adjacent matrix. Score matrix consists of 191 rows of number of nodes, each is the score of each node.

B. *Updates of Travelling Time Matrix Using Dijkstra's Algorithm*

As shown in TABLE 1, the travel time matrix has many null values because many nodes are not directly connected. This result still can not be used as input for Genetic Algorithm because if executed will not produce a decent solution (individual) in accordance with the population. In fact, although not directly connected, nodes have passable paths so that the travel time value is not null. To fill in the zero value of the initial trip time matrix, Dijkstra's algorithm is employed to find the travel time. The zero value in the initial matrix will be filled with the fastest travel time between the nodes generated by the algorithm.

C. *Parameter and Data Initiation*

At this stage parameters are defined as inputs into the algorithm: initial node, end node, number of nodes, time constraints (tMax), population number, number of individuals selected, number of generations, crossover probabilities, and mutation probabilities.

D. *Initial Population Generation*

In this stage, initial individuals representing feasible solutions based on the tMax variable (time limit) are randomly generated. In addition, in searching for feasible (individual) solutions, iterations are carried out with a pre-determined amount and should not exceed the number of populations.

E. *Fitness Evaluation*

At this stage, the fitness function of each individual is calculated based on the objective function define in Equation (1). The fitness value represents the quality of each individual.

TABLE 1 The Initial Traveling Time Matrix

node	1	2	3	4	5	6	7	
1	0	1	0	0	0	0	0	
2	1	0	1	0	0	0	0	
3	0	1	0	1	2	0	0	
5	0	0	1	0	0	0	1	
6	0	0	0	0	0	0	2	
7	0	0	0	1	0	0	0	

.

F. *Selection*

Within selection stage, a set of individuals are selected from the population that will serve as the parent for the cross-over process. The selection probability is obtained by dividing the individual fitness value by the total fitness of the population. If the individual probability is greater than a generated random number then the individual with the k index will be selected, but if it is not, then the search will continue to the next index.

G. *Crossover*

Selected individuals from the selection stage in turn become parent in this stage. Upon the chosen parent cross-over operator is carried out. The probability of cross-over probability used in this study is 0.9.

The new generation is formed by merging the two sub-lists as result from cross over operator and the final node is added to the last index. After the iteration for the cross-over process is completed, then the generation of the cross-over will be considered as offspring. The generation that appear more than once and exceed the time travel limit (tMax) will be excluded.

H. *Mutation*

The offspring as result of the cross-over operation, then will be mutated using a local search with two operators: the 'add operator' to add genes and 'omit operator' to remove the gene. The local search is performed sequentially and repeatedly until it reaches the pre-defined maximum loop. Each generation result has a chance to be mutated provided The mutation probability used in this study is 0.1.

Operation of local search on add and omit operators is called 1 mutation iteration. The number iteration is equivalent with the number of generation produced in the cross-over process.

I. *Generate New Population*

The new population is derived from the combining the initial individual population and the the new individual from the mutation stage. All such individuals will be temporarily put into storage. Furthermore, from the combined individuals in the deposit will be selected the best individual based on fitness value as much as the predefined population number (50 individuals).

J. *Selection of The Best Solution*

Once the Genetic Algorithm reaches the final iteration, the individual with the highest fitness value is chosen as the best solution.

V. RESULT AND DISCUSSION

A. *Search Results Solution with Genetic Algorithm*

During the experiment, the algorithm is run to find the solution of the problem of finding the optimum route from node 1 ((Jl. Kalimas Barat No.63, Pangkalan Petekan) to node 166 (Jl. Wonorejo No.18, Pangkalan Manukon). In this case study the maximum time limit is 70 minutes.

1) Validation of Initial Solution

Before finding the optimal solution, the application will find for an initial solution that will be the output of the initial population generation process and become the input for the selection process until the mutation. Some Results of Initial Population Generation can be seen in the TABLE 2. The resulting solution as in TABLE 2 is considered feasible if it does not violate the hard constrains predefined in the OP model. From the table we can observe that the first constraint is fulfilled because the route has fixed start and end nod. The is second constraint is also fulfilled because all nodes are connected, and each node is visited once. The third constraint is fulfilled because the total travel time does not exceed the total allocated time limit (tMax). Finally, the fourth and fifth constraints are also met because there are no subtours, seen from the initial and final genes only appear once in each solution. At this stage, all limitation are met, so the initial solutions could be validated as feasible solutions.

2) Validation of Final Solution

After the initial population generation stage, the selection, cross-over, and mutation stages as explained in the previous section are carried out sequentially in several generations to get the final solution. The final solution in the population can be seen in TABLE 3.

From TABLE 3, it can be observed that the final solution of population there are some solutions with decreased travel time (compared to the initial solution), however and almost all solutions have increased score. All of these solutions have also met the hard constraints of the OP model so that the final solution has been validated as feasible solutions. From these solutions the best solution is taken by looking at the highest fitness value that is on the first line solution with a fitness value of 7095 and has a travel time of 62 minutes. The best solution is considered to be the most optimum route recommendation for this case study.

TABLE 2 Some Results of Initial Population Generation

Individuals	[Solution] (Traveling Time, Score)
1	[1, 13, 166] (58, 496)
2	[1, 83, 166] (58, 512)
3	[1, 75, 166] (70, 428)
4	[1, 107, 17, 166] (63, 828)
5	[1, 12, 105, 166] (58, 699)

B. Comparison of Eac h Genetic Algorithm Parameter

In this experiment, several experimental scenarios were made to compare Genetic Algorithm (GA) results based on the problems in the previous case study, i.e. searching the optimum route from node 1 to node 166. The number of generations and Tmax used were 300 generations and 70 minutes. This experiment compares the size of initial population parameter (50, 100, and 150 individuals), cross-over probability (0.8 and 0.9), and mutation probabilities (0.05 and 0.01) as suggested in [13]. Therefore, there are 4 scenarios in which each scenario will be run 10 times and the average of its fitness value is recorded. The best result of the 4 scenarios can be seen in TABLE 4.

Based on the average fitness value of the four scenarios, the best solution is found from running algorithm with parameters: the cross-over probability parameter is 0.9, the mutation probability is 0.1, and the population size is 150. The result is the average of ten experiments with a fitness value of 8045.1.

C. Comparison of Number Of Generations

The experiment is carried out by using different number of iterations i.e. 300, 500, and 1000. In this case study we used the population number, Tmax, cross-over probability, and mutation probability like the previous case study, i.e. 50 individuals, 70 minutes, 0.9, and 0.1. The results of the comparison can be seen in TABLE 5.

TABLE 3 Final Solution of Population

Individuals	[Solution] (Traveling Time, Score)
1	[1, 2, 3, 4, 7, 8, 9, 12, 13, 15, 18, 107, 106, 105, 104, 103, 102, 101, 100, 99, 98, 97, 92, 91, 96, 90, 89, 88, 87, 86, 85, 84, 83, 81, 79, 82, 155, 166] (62, 7095)
2	[1, 2, 3, 4, 7, 8, 9, 12, 13, 15, 18, 107, 106, 105, 104, 103, 102, 101, 100, 99, 98, 97, 92, 91, 90, 89, 96, 88, 87, 86, 85, 84, 83, 81, 79, 82, 166] (66, 7011)
3	[1, 2, 3, 4, 7, 8, 9, 12, 13, 15, 18, 107, 106, 105, 104, 103, 102, 101, 100, 99, 98, 97, 92, 91, 90, 89, 87, 96, 88, 86, 85, 84, 83, 81, 79, 82, 166] (70, 7011)
4	[1, 2, 3, 4, 7, 8, 9, 12, 13, 15, 18, 107, 106, 105, 104, 103, 101, 100, 99, 102, 98, 97, 92, 91, 90, 96, 89, 88, 87, 86, 85, 84, 83, 81, 79, 82, 166] (70, 7011)
5	[1, 2, 3, 4, 7, 8, 9, 12, 13, 15, 18, 107, 106, 105, 104, 103, 101, 100, 99, 102, 98, 97, 96, 91, 92, 90, 89, 88, 87, 86, 85, 84, 83, 81, 79, 82, 166] (66, 7011)

TABLE 4 Best Test Results for Each Scenario

Prob CO	Prob Mut	Num Pop	Fitness	Traveling Time
0.9	0.1	150	8045.1	67
0.9	0.05	150	7991.7	67.6
0.8	0.1	150	7912.2	66.9
0.8	0.05	150	7894.4	66.6

From TABLE 5 it can be observed that improvements in fitness values occur in all scenarios with differing number of iterations. However, unsurprisingly, the scenario with the greatest number of iterations, i.e. 1000 outperforms the other scenarios. This result shows that the number of generations is one of important factors in finding the optimum solution. In addition, as shown in Fig. 2, the actual fitness value improvement has stopped in the 758th generation (as shown by the red box) with a fitness value of 7775. The fitness value does not change until 1000th generation (maximum number of generations specified in this experiment). For other case studies, the best number of generations are varied.

TABLE 5 The Result of the Comparison of the Number Generations

Gene ration	Optimum Solution	Travel Time	Fit ness
300	[1, 2, 3, 7, 8, 9, 10, 11, 6, 4, 12, 13, 15, 18, 107, 106, 105, 104, 103, 100, 99, 98, 97, 96, 92, 91, 90, 89, 88, 86, 85, 84, 83, 81, 79, 78, 77, 166]	69	7523
500	[1, 2, 3, 7, 8, 9, 10, 11, 6, 4, 12, 13, 15, 18, 107, 106, 105, 104, 103, 102, 101, 100, 99, 98, 97, 96, 92, 91, 90, 89, 88, 86, 85, 84, 83, 81, 79, 78, 77, 166]	69	7691
1000	[1, 2, 3, 7, 8, 9, 10, 11, 6, 4, 12, 13, 15, 18, 107, 106, 105, 104, 103, 102, 101, 100, 99, 98, 97, 96, 92, 91, 90, 89, 88, 87, 86, 85, 84, 83, 81, 79, 78, 77, 166]	69	7775

VI. Conclusion And Discussion

Based on the results of research that has been done, can be concluded: (1) Orienteering Problem (OP) method can be used to model the determination of the optimum travel route using angkot in Surabaya. (2) Dijkstra's algorithm can be used to help search the fastest travel time between nodes so as to assist in the search for viable solutions. The results of this algorithm can be used as a dataset for further research. (3) Genetic Algorithm (GA) can be used to find the optimum route based on the predefined OP model. (4) Selection of the number of generations, the number population, the cross-over probability, and the mutations probabilityhave an important role in optimizing using the Genetic Algorithm for the search for the optimum solution.

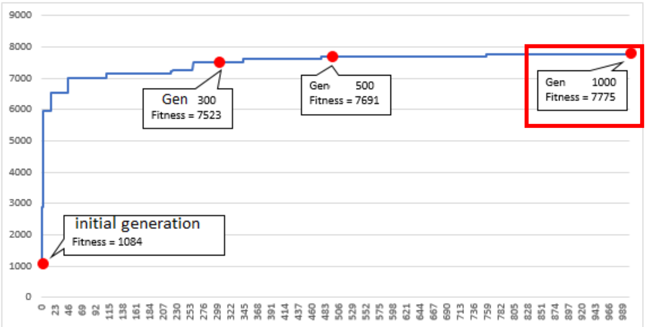

Fig. 2 Change the Fitness Value Each Generation

In the future work, it is recommended that more number of route, i.e. the number of Angkot route should be considered. The representation of scores from the Orienteering Problem model can be enhanced by considering several other factors such as the level of congestion at each node, the number visitors passing through each node, or the level of popularity node or place around the node. Regarding the travel time, to be more realistic, instead of fixed travel time, in the future dynamic travel time could be used, for example based on the hours from historical records of real-time data in the field. In addition, it can be further developed by adding Angkot recommendations in addition to only the routes.

References

[1] "Kemacetan Jakarta: Nilai Kerugian Masyarakat Rp150 Triliun Per Tahun." [Online]. Available: http://jakarta.bisnis.com/read/20160425/77/541368/kemacetan-jakarta-nilai-kerugian-masyarakat-rp150-triliun-per-tahun. [Accessed: 29-May-2018].

[2] M. Agustinus, "Ini Dampak Kemacetan Terhadap Perekonomian Jakarta," *detikfinance*. [Online]. Available: https://finance.detik.com/berita-ekonomi-bisnis/d-3243447/ini-dampak-kemacetan-terhadap-perekonomian-jakarta. [Accessed: 29-May-2018].

[3] P. V. M. B.- VIVA, "Surabaya Macet, Rp1 Triliun Menguap – VIVA," 23-Sep-2010. [Online]. Available: https://www.viva.co.id/jatim/179159-surabaya-macet-rp1-triliun-per-hari-menguap. [Accessed: 29-May-2018].

[4] "SITS - DISHUB SURABAYA." [Online]. Available: http://sits.dishub.surabaya.go.id/ver2/tentang-sits. [Accessed: 29-May-2018].

[5] "Siapkah Indonesia Implementasikan ITS? oleh Dian Kurniasari - Kompasiana.com." [Online]. Available: https://www.kompasiana.com/dian_kur/siapkah-indonesia-implementasikan-its_54f7c3d3a33311181d8b4980. [Accessed: 29-May-2018].

[6] "Perihal Angkot dan Tips Aman Menaikinya.

[7] P. Vansteenwegen, W. Souffriau, and D. V. Oudheusden, "The orienteering problem: A survey," *Eur. J. Oper. Res.*, vol. 209, no. 1, pp. 1–10, Feb. 2011.

[8] J. S. Wijaya, W. Anggraeni, A. Muklason, F. Mahananto, E. Riksakomara, and A. Djunaidy, "Advanced Traveler Information System: Itinerary Optimization as an Orienteering Problem Using Iterative Local Search-Hill Climbing Algorithm," *Procedia Comput. Sci.*, vol. 124, pp. 173–180, 2017.

[9] M. F. Tasgetiren and A. E. Smith, "A genetic algorithm for the orienteering problem," 2000, vol. 2, pp. 910–915.

[10] B. W. Taylor, *Introduction to Management Science*, 11th ed. Pearson, 2013.

[11] "Dijkstra's Shortest Path Algorithm | Brilliant Math & Science Wiki." [Online]. Available: https://brilliant.org/wiki/dijkstras-short-path-finder/. [Accessed: 29-May-2018].

[12] "Dijsktra's algorithm," *GeeksforGeeks*, 25-Nov-2012.

[13] "Parameters of GA." [Online]. Available: http://www.obitko.com/tutorials/genetic-algorithms/recommendations.php. [Accessed: 29-May-2018].

[14] T. Tsiligirides, "Heuristic Methods Applied to Orienteering," *Journal of the Operational Research Society,* pp. 797-809, 1984.

[15] W. e. a. Souffriau, "A Personalized Tourist Trip Design Algorithm for Mobile Tourist Guides," Applied Artificial Intelligence, vol. 22, pp. 964-985, 2008.

[16] X. Wang, B. L. Golden dan E. A. Wasil, "Using a Genetic Algorithm to Solve the Generalized Orienteering Problem," The Vehicle Routing Problem: Latest Advances and New Challenges, vol. 43, pp. 263-274, 2008.

[17] M. Schilde, K. Doerner, R. Hartl dan G. Kiechle, "Metaheuristics for the Bi-Objective Orienteering Problem," Swarm Intelligence, vol. 3, pp. 179-201, 2009.

[18] M. F. Tasgetiren, "A Genetic Algorithm with an Adaptive Penalty Function for the Orienteering Problem," Journal of Economic and Social Research, pp. 1-26, 2001.

[19] M. F. Tasgetiren dan A. E. Smith, "A Genetic Algorithm for the Orienteering Problem," Evolutionary Computation, vol. 2, pp. 910-915, 2000.

[20] J. Ferreira, A. Quintas, J. A. Oliveira, G. A. Pereira dan L. & Dias, "Solving the Team Orienteering Problem: Developing a Solution Tool Using a Genetic Algorithm Approach," Soft Computing in Industrial Applications, pp. 365-375, 2014.

[21] J. Karbowska-Chilinska, J. Koszelew, K. Ostrowski dan P. Zabielski, "Genetic Algorithm Solving Orienteering Problem in Large Networks," KES, pp. 28-38, 2012.