An application of traveling salesman problem using the improved genetic algorithm on android google maps

Cite as: AIP Conference Proceedings **1818**, 020035 (2017); https://doi.org/10.1063/1.4976899
Published Online: 10 March 2017

Teguh Narwadi and Subiyanto

An Application of Traveling Salesman Problem Using the Improved Genetic Algorithm on Android Google Maps

Teguh Narwadi[1,a)] and Subiyanto[2, b)]

[1]*Electrical Engineering Department, Engineering Faculty Universitas Negeri Semarang*
[2]*Sekaran, Gunungpati, Semarang*

[a)] Corresponding author: teguhnarwadi@gmail.com
[b)] subiyanto@mail.unnes.ac.id

Abstract. *The Travelling Salesman Problem (TSP) is one of the best known NP-hard problems, which means that no exact algorithm to solve it in polynomial time. This paper present a new variant application genetic algorithm approach with a local search technique has been developed to solve the TSP. For the local search technique, an iterative hill climbing method has been used. The system is implemented on the Android OS because android is now widely used around the world and it is mobile system. It is also integrated with Google API that can to get the geographical location and the distance of the cities, and displays the route. Therefore, we do some experimentation to test the behavior of the application. To test the effectiveness of the application of hybrid genetic algorithm (HGA) is compare with the application of simple GA in 5 sample from the cities in Central Java, Indonesia with different numbers of cities. According to the experiment results obtained that in the average solution HGA shows in 5 tests out of 5 (100%) is better than simple GA. The results have shown that the hybrid genetic algorithm outperforms the genetic algorithm especially in the case with the problem higher complexity.*

INTRODUCTION

The Travelling Salesman Problem (TSP) is one of the best known NP-hard problems, which means that there is not guaranteed to get the optimal route and no exact algorithm to solve it in polynomial time [1]. The method which would definitely obtain the optimal solution of TSP is the method of exhaustive enumeration and evaluation. This procedure begins by generating the possibility of all the tours and evaluate according length of the tour. The tour with the smallest length chosen as the best, and guaranteed to be optimal [2]. At this time many companies and institutions are confronted and have to resolve cases of TSP. For example, delivery service companies that every day faced with the problems, it occurs because the route followed to get to the destination is not optimal. Examples of cases in delivery service, a courier needs to deliver goods to some customers with a different destinations. Therefore, time is of considerable concern in the delivery of goods it relates to the reputation of the company. To reach these goals required a system capable of providing an optimal travel route so that the travel time to be minimum. So it needs an application that can optimize the TSP solution, it uses a genetic algorithm to produce the optimal solution in a short time.

A genetic algorithm (GA) is search technique in computer science to find approximate solutions to optimization and search problems such a TSP. GA can be an alternative to solve the TSP in recent years, as has been proved to be more efficient and attractive in finding the optimal or near-optimal solutions [5] [6]. GA is a population-based algorithm, this method a collection of solutions called population and the best populations will be maintained to obtain an optimal solution by using genetic operators [3]. There is a weakness in applying GA to solve TSP, that is GA will get optimal results when converge to the global optimum but when GA converges to a local optimum, it gets a solution that is not optimal. Due GA can‘t do a search on the whole space and there is no way to escape [4]. One of the methods to overcome these weaknesses is hybridization of local search technique. It can improve the performance of GA with inserting the best new individual into looping GA. HGA has proven to be more efficient than GA is applied in optimization problems such as TSP [7] [9].

Engineering International Conference (EIC) 2016
AIP Conf. Proc. 1818, 020035-1–020035-11; doi: 10.1063/1.4976899
Published by AIP Publishing. 978-0-7354-1486-0/$30.00

The application is developed on android because it is a mobile operating system that is easily portable and it has been integrated by Google Maps. Android is now the most widely used around the world because it has the largest installed base of all operating systems in 2015 [10]. Besides, it can be build a system that gave optimal travel route in TSP [1] [17]. The system is integrated with Google maps to get a real trip planning, such as showing the route to the destination cities, and indicate travel distances in set of kilometres. Based on the description above, this paper design and develop the application solve the TSP using hybrid genetic algorithm with Google maps on android. The developed application can become an effective for solving TSP. The remainder of this paper is organized as follows: Sect. 2 overview of problem and method. Section 3 the HGA methodology to solving TSP are described. Section 4 the implementation of HGA to solve TSP on android and the application testing described. Section 5 The computational results are presented and discussed. Finally, the conclusion is followed in Sect. 6.

OVERVIEW TRAVELING SALESMAN PROBLEM AND HYBRID GENETIC ALGORITHM

Traveling salesman problem

The traveling salesman problem (TSP) is a problem in combinatorial optimization and has several applications, such as vehicle routing problems, logistics, planning and scheduling. The main problem of the TSP is to find a tour of a given number (n) of cities with a minimum distance, where each city visited exactly once and returning to the starting city [8]. Mathematically, it can be defined as given a set of n cities, named $\{c_1,c_2,\ldots, c_n\}$, and permutations, $\sigma_1,\ldots, \sigma_n!$, the objective is to choose σi such that the sum of all Euclidean distances between cities in a tour is minimized.

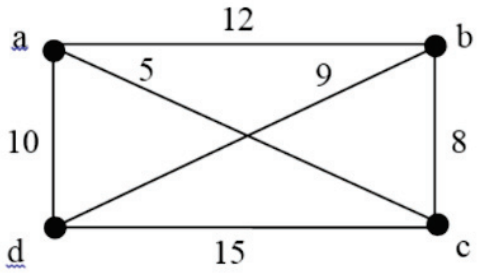

FIGURE 1. Four city TSP [12]

TSP issue can be solved by enumerating $(n-1)!/2$, where n is the number of cities and then select the route with the shortest length. The position of the city with the number 4 city (a, b, c, d) as shown in Figure 1 that have $(4-1)!/2 = 3$, possible routes that can be passed. Finished, the shortest route is (a, c, b, d, a) or (a, d, b, c, a) with a route length of 32.

Genetic Algorithm

John Holland proposed Genetic Algorithm in 1975. In the field of artificial intelligence genetic algorithm is a search heuristic that mimics the process of natural evolution. Genetic Algorithm belongs to class of evolutionary algorithm. GA begin with various problem solution which are encoded into population, a fitness function is applied for evaluating the fitness of each individual, after that a new generation is created through the process of selection, crossover and mutation. After the termination of genetic algorithm, an optimal solution is obtained. If the termination condition is not satisfied then algorithm continues with new population [13]. The overall simple genetic algorithm shown in Fig 2.

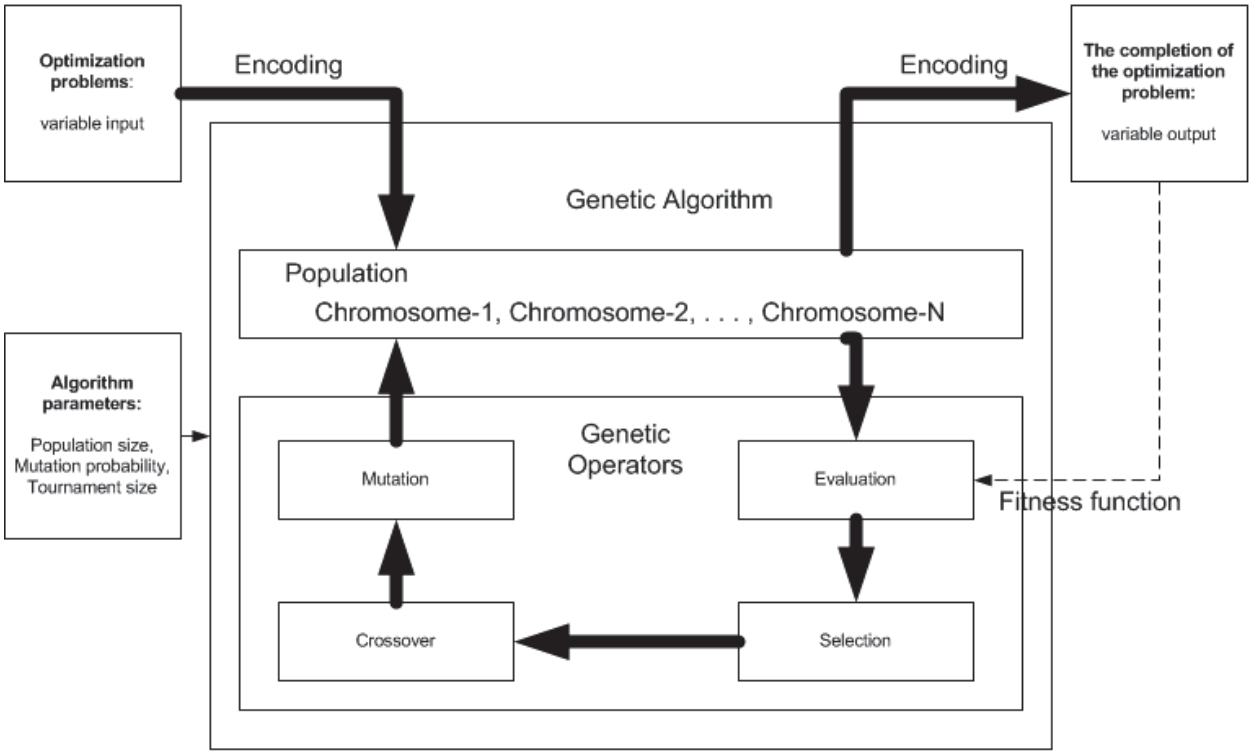

FIGURE 2. The overall structure of genetic algorithm

1. **Representation**
   Several representation methods have been proposed for represent a problem of TSP like a binary, matrix and path representation. The path representation is probably the most obvious representation of a tour on TSP [16].

2. **Initialization**
   In this step, an initial population is generated from a random selection of solutions (chromosomes). Genetic algorithm performance is influenced by the size of the population if the problem becomes more difficult then the size of the population should increase [3].

3. **Evaluation**
   After initial population, to improve the solutions each chromosome in the population is evaluated by using the fitness function. The fitness function we are using inverse of the objective function. In this way, we can find the best chromosome that corresponds to the shortest path.

4. **Selection**
   Process selection is choosing two parents from the population for further crossover process. Tournament selection is used for this section, because of its efficiency and simple implementation [5].

5. **Crossover**
   Crossover is the process that mimics mating between two chromosomes with the goal of producing offspring. A new offspring is generated via the order crossover (OX) of two parent chromosomes. OX is used to perform the path/permutation representation, because OX rather concerns circular permutations [14].

6. **Mutation**
   The function of mutation is to add the diversity of the population and prevent the chromosomes falling into the local minima [9]. The current implementation uses exchange mutation. Which randomly selects two genes of a chromosome and exchanges them. After creating new population by crossover and mutation, there is a big

chance, that we will lose the best chromosome [6]. Therefore elitism method (elite selection) is used. The main objective of elitism is to determine the best individuals and immediately copy them over to the next generation.

7. **Termination condition**
   Termination condition is the criteria used to stop the evolutionary process. When the evolutionary process has reached the maximum time period then stop.

Local Search Technique

A local search method appropriate to help weaknesses of GA because it is generate a local optimal solution by exploring the neighborhood of a given initial solution. The use of local search techniques has been proven to be useful in solving combinatorial problems. For local search technique, we use the iterative hill climbing suggested by Michalewicz [15]. Hill Climbing algorithm searches for a better solution in the neighborhood. If it finds a better solution, it changes the current solution with this new one. If the new solution is not the better one then the algorithm stops and keeps the current local optimum solution. Here are the details procedure of the local search.

Procedure: Iterative hill climbing method
Begin
Select best individual v_c from current GA population according to fitness function;
Generate as many individuals as the population size in the v_c neighborhood randomly;
Select the individual v_n with the best fitness value from newly generated individuals;
If fitness(v_c) < fitness(v_n) **then**
$v_c \rightarrow v_n$
Else If fitness(v_c) ≥ fitness(v_n) **then**
$v_c \rightarrow v_c$
end
End

Hybrid Genetic Algorithm

In the field of TSP, simple GA can't find the optimal solution [4]. Therefore, a hybrid genetic algorithm (HGA) is designed to obtain optimal results of the traveling salesman. In our HGA, combines genetic algorithm described in section 2.2, and local search is described in section 2.3. Here are the details of hybrid genetic algorithm procedure.

Procedure: Hybrid Genetic Algorithm
Begin
Step1: Initial population:
Encode the problem using path representation and generate initial population
Step2: Evaluation:
Evaluate the current population using the fitness function
Step3: Genetic operators:
Selection: Select two parents for crossover using tournament selection
Crossover: Crossover two chromosomes selected using order crossover
Mutation: Mutate the offspring using exchange mutation
Step4: Local Search:
Using hill climbing method to search a new best individual
Step5: Termination condition:
The termination conditions have been set to a time frame of 60 seconds. If this termination condition are reach and stop looping.
End

HYBRID SYSTEM FOR TSP WITH GOOGLE MAPS ANDROID

The hybrid system is implemented to solve TSP to obtain the optimal route on Google Maps in Android OS. The hybrid system using Google Maps API Android for handles access to Google Maps servers, data downloading, map display, and response to map gestures [11]. In this section, the methodology to build the hybrid system described.

Design of Hybrid Algorithm on Android

In the hybrid system function of Google Maps API is to obtain distance between cities and marker of the geographical location (latitude and longitude) on the map as a destination cities of the salesman. The destination cities should be set from database SQLite. Once, the city position is obtained and shown into Google maps android as a marker. Based on the TSP, distance between cities to be visited are gotten from Google Server are stored into the data set matrix. Fig. 3 shows the marker and distance obtained from Google servers.

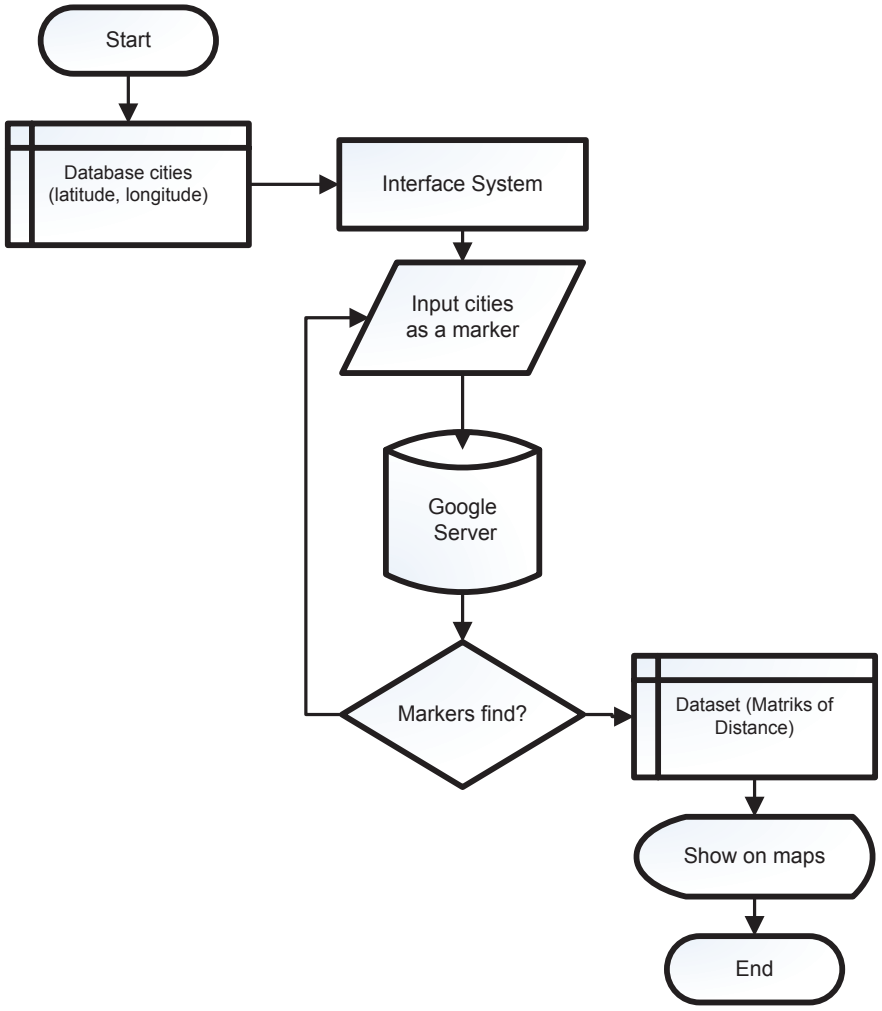

FIGURE 3. Markers and dataset distance obtained from Google server

The hybrid genetic algorithm taking of dataset (Matrix of distance) to get the distance used in the evaluation process. This matrix is temporary dataset so that when a hybrid genetic algorithm has found the optimal route, matrix dataset will be set automatically to null. Hybrid genetic algorithm will stop if the stop condition has been met.

The stop condition in the hybrid system is a time period. This time period is chosen for a reasonable waiting time for users [1]. Fig. 4 show a workflow of the hybrid genetic algorithm.

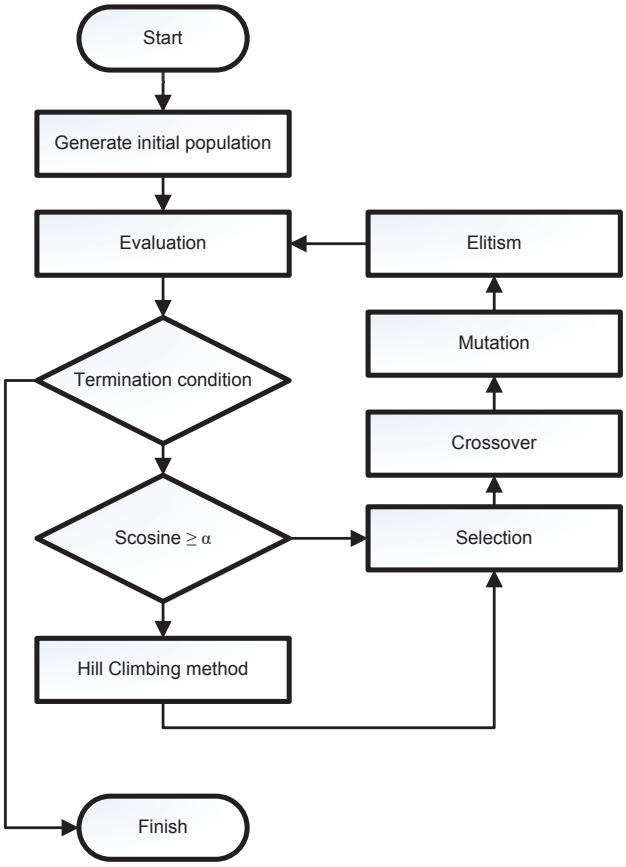

FIGURE 4. Workflow of the hybrid genetic algorithm

Flowchart of the overall hybrid system is described in Figure 4. So the hybrid system, the first step is the input location of the application, then the distance and markers obtained from Google maps server and stored in temporary on the matrix dataset. So that the hybrid genetic algorithm calculations do not need to get distance from Google maps server but the direct taking of datasets matrices. Route with the shortest distance shown on Google maps.

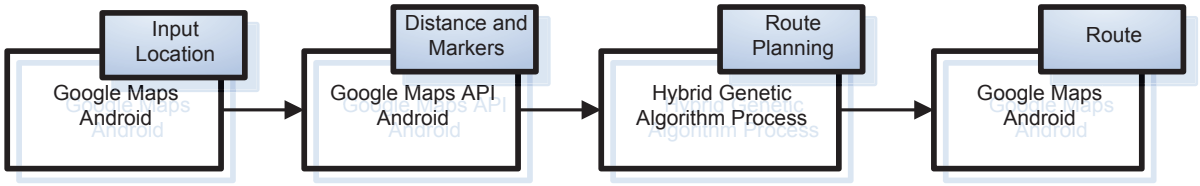

FIGURE 5. Workflow of the hybrid system

Implementation of Hybrid Algorithm on Google Maps in Android

The application developed by Android Developer Tool (ADT) and using the Java programming language. To test the application, smartphone android Quad-core 1.3 GHz Cortex-A7, 1 GB RAM, Android OS v4.4.2 (Kitkat)

was used. To integrating the Google maps into applications must be install Google play services in framework ADT. Then, get access the Google maps server with API key and add to your application. To get the API key must be generated using the SHA1 fingerprint from the developer computer, the following command must be executed:

```
keytool -list -v -keystore [path to debug.keystore] -alias androiddebugkey -storepass android-keypass android
```

Register the SHA1 fingerprint in the Google APIs Console to get API key and can access Google Maps Service V2. In the androidmanifest.xml.

COMPUTATIONAL EXPERIMENTS

Experiments of algorithm

Experiment of the algorithm is used to investigate the parameters on hybrid genetic algorithm. Parameters tested were as follows mutation probability (*pm*), termination condition, and population size. The data used for the experiments of algorithm that is using a well-known benchmark TSPLIB. Tests carried out on 3 sample data from TSPLIB with the number of cities (n) different. Hybrid genetic algorithm in the same conditions in each sample. Tests proved to the best solution (best) and the average solution (avg) of 20 experiments carried. Exposure results of experiments on parameters hybrid genetic algorithm is presented in table 1, table 2, and table 3.

In the first test, testing the mutation probability. In Table 1 shows that the mutation probability on $pm = 0.05$ to get the best results in all experiments. Meanwhile the computing time in seconds, shows the average computation time required to obtain the optimal solution in 20 times the experiment.

TABLE 1. Results over different instances using different mutation probabilities

instance		*pm*=0.01	*pm*=0.03	*pm*=0.05
burma14	best	3668	3650	3210
	avg	4016.2	3743.8	3642.6
	time	0.040	0.054	0.064
ulysses22	best	8530	9233	8386
	avg	9575.8	9701.6	9415.2
	time	0.187	0.215	0.179
bays29	best	14786	13650	12377
	avg	15021	14508.2	14116.2
	time	0.466	0.518	0.412

Table 2 shows the results of testing the termination condition. The termination condition is limited to a maximum period, the criteria have been more appropriate for the application being developed. The termination condition will determine the optimal solution produced, because if the process of evolution increases, the optimal solution can be reached. Therefore, to test the improvement solutions, time period (s) be varied to determine if there is a significant difference in the resulting solution. From Table 2 test results obtained with s=60 obtain optimal results in all tests conducted. This time period is used as a parameter in the application developed. So the user needs to wait for 60 seconds for the process of the calculation algorithm before a solution is found

TABLE 2. The result of the termination condition

instance	n		s=20	s=40	s=60
burma14	14	best	3087	3087	3087
		avg	3087	3087	3087
ulysses22	22	best	7579	7530	7530
		avg	7681	7626	7603
bays29	29	best	9542	9213	9105
		avg	10272	9741	9488

In the third test are shown in Table 3, testing the population size. Tests conducted to determine the appropriate number of the population in developed applications. Population size is made varies depending on the number of cities (n) is selected. Table 3 shows the optimal distance difference resulting from the increase in the population with a limited period of time. The optimal result obtained on the sample ulysses22 and 29 bays on the amount equal to the total population of the cities. While in the multiples of two and three of the population getting worse results.

TABLE 3. The result of the population size

instance	n	pop size	best	avg
burma14	14	14	3087	3087
		28	3087	3087
		42	3087	3087
ulysses22	22	22	7530	7590
		44	7530	7601
		66	7530	7592
bays29	29	29	9105	9468
		58	9203	9582
		87	9203	9653

Experiments data

To test the hybrid system, tested using a sample of several city in Central Java, Indonesia. Data such as geographical location of cities obtained from https://maps.google.co.id/. Used 5 case for testing, with the number of cities different in every case. The purpose of the test is to know the effectiveness of HGA to solve the TSP, in the testing HGA compared with the simple GA. Measurement used in this test is a unit of meters which represents the length of the route that passed. For each sample TSP, the simulation performed 10 times. HGA and GA under the same conditions for experiments and parameters used are as follows: the probability of mutation is 0.05, tournament size is 3 and termination condition is 60 second. For the 5 sample, we set the the population size 5, 8, 12, 16, and 20 respectively. Here is the list of cities in Central Java, Indonesia.

In this experiment we take 5 TSP sample with different number of cities, including:

- Case1 with the number of cities are 5 of them Pati, Purwodadi, Demak, Temanggung, Magelang.
- Case2 with the number of cities are 8 of them Pati, Purwodadi, Demak, Temanggung, Magelang, Sragen, Rembang, Wonosobo.
- Case3 with the number of cities are 12 of them Pati, Purwodadi, Demak, Temanggung, Magelang, Sragen, Rembang, Wonosobo, Kendal, Batang, Banjarnegera, Semarang.
- Case4 with the number of cities are 16 of them Pati, Purwodadi, Demak, Temanggung, Magelang, Sragen, Rembang, Wonosobo, Kendal, Batang, Banjarnegera, Semarang, Purwokerto, Blora, Kudus, Jepara.

- Case5 with the number of cities are 20 of them Pati, Purwodadi, Demak, Temanggung, Magelang, Sragen, Rembang, Wonosobo, Kendal, Batang, Banjarnegera, Semarang, Purwokerto, Blora, Kudus, Jepara, Kebumen, Sukoharjo, Slawi, Wonogiri.

- **TABLE 4.** The several cities in Central Java, Indonesia

No.	City	Latitude	Longitude
1	Pati	-6.753899	111.042784
2	Purwodadi	-7.083219	110.913327
3	Demak	-6.894602	110.638154
4	Temanggung	-7.316753	110.178341
5	Magelang	-7.476794	110.218917
6	Sragen	-7.427645	111.023387
7	Rembang	-6.705611	111.348445
8	Wonosobo	-7.358785	109.902953
9	Kendal	-6.920633	110.20483
10	Batang	-6.906720	109.732468
11	Banjarnegera	-7.399196	109.688866
12	Semarang	-6.990155	110.422515
13	Purwokerto	-7.427879	109.242443
14	Blora	-6.970731	111.421695
15	Kudus	-6.808916	110.842946
16	Jepara	-6.590626	110.667318
17	Kebumen	-7.669664	109.651954
18	Sukoharjo	-7.683752	110.846919
19	Slawi	-6.974607	109.139532
20	Wonogiri	-7.813768	110.926073

RESULT AND DISCUSSION

Comparison of the best distance, distance worst and average distance is calculated using a hybrid genetic algorithm and simple genetic algorithm in case1, case2, case3, case4 and case5 presented in Table 5. Our experiments shows that the hybrid genetic algorithm method is better than simple genetic algorithm.

TABLE 5. The result of the experiments

No	Data	Number of cities	Distance (meters)					
			GA			HGA		
			Best	Worst	Average	Best	Worst	Average
1	Case1	5	333390	333390	333390	333390	333390	333390
2	Case2	8	518053	595128	533484.6	518053	518053	518053
3	Case3	12	667143	872809	750450.6	622172	719781	679885.6
4	Case4	16	999926	1107099	1060378.4	823899	977451	906274
5	Case5	20	1175422	1314313	1260734.8	1001333	1289746	1185912.8

In the table 5 shows that GA getting worst results at all the case, especially on a large number of cities. It is proved that GA is not able to obtain optimal results because it was stuck in a local optimum. Therefore, the HGA method is able to improve the performance of GA in obtaining optimal results. This is evidenced in the average distance and the worst distance. The average distance HGA shows in 5 tests (100%) find the optimal solution that outperform the GA. Then in the worst distance are resulting from each method, in 5 tests (100%) show that HGA is better than GA.

The best distance show that hybrid genetic algorithm found the optimal distance in case3, case4, case5. In case1 and case2 the best solution HGA same with the best solution of GA. In this case does not mean that HGA will not constantly perform better than GA because small data merely reflects simple problem. This method is less effectively used in a small data. In the experiment with the large data especially where the cities is 8, HGA obtain significantly better results than the simple GA.

FIGURE 6. Result of the best solution of HGA for case5

In the Figure 6 shows the best route of the destination cities to be visited by a salesman contains 20 city. The cities are represented by markers and travel route represented with black lines. A salesman started the travel from the city first then visit the city next two salesman visiting the city third, fourth city until the twentieth city and return to first city. The distance covered by the salesman is 1,001,333 meters or 1,001.333 kilometers. The hybrid system able to do the calculations in problems of optimally traveling salesman problem, and displays trip planning with Google maps on android.

CONCLUSIONS

In this paper, the application of hybrid genetic algorithm to optimize the traveling salesman problem have been developed and tested. In the experiments, HGA have been compared with GA to solve traveling salesman problem. In the results and discussion, it was obtained that the HGA yields the minimum distance (optimal tour length) in most cases, especially in the case with the problem higher complexity. The computational results have shown that the HGA outperforms the simple GA.

REFERENCES

1. I. Berninger, "Vehicle Routing Problem on Android/iOS," Bachelor Thesis, Institute of Computer Science Research Group DPS, University Innsbruck, 2014.
2. A. Homaifar, S. Guan, and G. E. Liepins, "Schema Analysis of the Traveling Salesman Problem Using Genetic Algorithms," Complex Systems 6, pp. 533-552, 1992.
3. A. Reese, "Random number generators in genetic algorithms for unconstrained and constrained optimization," J Nonlinear Anal, **71**, 679–692, 2009.

4. Y. Yun, C. Moon, and D. Kim. "Hybrid genetic algorithm with adaptive local search scheme for solving multistage-based supply chain problems," Comput Ind Eng **56**, 821–838, 2009.
5. N. M. Razali and J. Geraghty, "*Genetic Algorithm Performance with Different Selection Strategies in Solving TSP*," Proceedings of the World Congress on Engineering 2011, Vol II WCE 2011, July 6 - 8, 2011, London, U.K.
6. K. Rani and V. Kumar, Int. J. Res.Eng. Tech. **2**, 27-34. (2014)
7. Z. H. Ahmed, "An experimental study of a hybrid genetic algorithm for the maximum traveling salesman problem," Springer open journal, 2013.
8. K. Bryant, "Genetic Algorithms and the Traveling Salesman Problem," Master's thesis, Harvey Mudd College, 2000.
9. Y. Wang, "The hybrid genetic algorithm with two local optimization strategies for traveling salesman problem," Comput. Ind. Eng. **70**, pp. 124–133 (2014).
10. F. Manjoo. (May 2015). A Murky Road Ahead for Android, Despite Market Dominance. The New York Times *[Online]*. ISSN 0362-4331. Available: http://www.nytimes.com/2015/05/28/technology/personaltech/a-murky-road-ahead-for-android-despite-market-dominance.html?_r=0
11. Developers Google. 2015. Google Maps Android API. Developer Google. *[Online]*. Available: https://developers.google.com/maps/documentation/android-api/
12. N. Jian and S. Sangwan, Global J. Inc, **11**, 7-14 (2011).
13. N. Kumar, Karambir, and R. Kumar, Int. J. Latest Res. Sci.Tech. **1**, 98-101, (2012).
14. R. Liu, Z. Jiang, and N. Geng, OR Spectrum **36**, pp. 401-421 (2014).
15. Z. Michalewicz, "Genetic Algorithms + Data Structures= Evolution Program," Springer, Berlin, 1992.
16. P. Larranaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, Artif. Intell, **13**, 129-170 (1992).
17. Anupriya, and M. Saxena, Int. J. Comput. Organ. Trends **3**, 70-73 (2013).