Minimizing capacity of Electric Vehicle Battery using Bottleneck Traveling Salesman Problem

Rio Aurachman\textsuperscript{1*}, Dyah Putri Saraswari\textsuperscript{2}
\textsuperscript{1,2}Telkom University, Bandung, Indonesia

rioaurachman@telkomuniversity.ac.id

Abstract. Electrical vehicle technology has now developed. Various public transportation can start using electric power sources. However, there are battery capacity constraints. When the distance is too far, the battery capacity is not sufficient to provide the required power. In this paper, we will explain how the bottleneck travel salesman problem is applied to minimize battery capacity requirements. So that vehicles can choose the right route by minimizing the maximum distance between destinations. The optimization process is carried out with the Brute Force algorithm and python programming language.

1. Introduction

The international community has begun to switch to using electric vehicles. This is considered as one of the solutions for environmental sustainability. However, unlike vehicles in general, electrical vehicles store energy to run the engine on a battery that needs to be recharged periodically.

Conventional vehicles store energy to drive the engine on fuel. If a vehicle has a large volume of fuel capacity, it will be able to travel long distances. Whereas in electrical vehicles, the farther the distance that must be traveled, the greater the capacity of the battery needed and that causes a larger battery volume.

The decision maker may want a route design that is not too far away so that it does not need to prepare a large battery capacity, and can reduce the total cost of producing the vehicle. One thing that can be done is to minimize the distance between destinations, assuming that each destination can be recharged.

This requirement is different from the route determination pattern which generally uses the traveling salesman problem to minimize the total route taken [1]. Given this need, what is prioritized is precisely the minimum distance between destinations. This avoids vehicles that stop not at their destination due to exhausted batteries. So we need a mathematical model and an optimization algorithm to minimize the maximum distance between destinations on a route.

Several studies have tried investigating the electrical vehicle route. One of the researches tries to apply blockchain technology in an autonomous selection of charging stations [2]. Another study is trying to propose a system that can provide information on the distance that an electrical vehicle [3] can travel. Other research examines how the system can adjust dynamically to changes in traffic congestion on the highway [4]. In addition, the optimal route for a hybrid vehicle has also been studied [5]. The energy in a hybrid vehicle is one of the things that have been paid attention to, one of which can be seen from the research on optimal energy regulation for each process [6]. There is research that tries to optimize the electric vehicle route by considering the charging station [7]. However, so far we have not succeeded in...
finding research that tries to find the optimal route with the assumption that charging stations are present at each destination. Therefore, this research tries to apply the Bottleneck Traveling Salesman Problem in determining a route that minimizes the maximum distance between destinations on a route.

2. Method
This study uses a mathematical modeling approach [8] and systems modeling [9]. The problems that exist in the electrical vehicle system, especially the scheduling system, are studied and observed in order to obtain a model of the system. Then from this conceptual model, it is transformed into a mathematical model. From the mathematical model obtained, it is then optimized using an algorithm and programming language.

The mathematical model is developed based on the commonly used Traveling salesman problem mathematical model. The model developed is called the Bottleneck traveling salesman problem. The optimization process uses the Brute Force algorithm. The programming language used is the python programming language. The number of optimized destinations is 7 destinations. This was chosen so that the Brute Force Algorithm can still be applied. The Brute Force Algorithm is chosen so that each iteration can evaluate the comparison of the total distance and the maximum distance between destinations on a route [10][11].

3. Result
Traveling salesman problem is a model that tries to find the minimum distance when a salesman must pass through a destination and then return to the starting point of departure. The traveling salesman problem uses the form of a basic mathematical model as shown in the following equation.

$$\text{Min} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \text{ for all } j$$

$$\sum_{j=1}^{n} x_{ij} = 1 \text{ for all } i$$

$$x_{ij} \in \{0, 1\} \text{ for all } i, j$$

The objective function, which was originally intended to minimize the total distance traveled from the traveler in forming a Hamiltonian circuit, in this case, uses different mathematical equations and logic. It is assumed that the electrical vehicle requires a short distance due to battery limitations. Then the minimum total distance is not preferable to the distance between destinations on the selected route. This can be seen in the illustration in Figure 1

![Figure 1. Comparison of two alternative route](image)

In the figure in Figure 1, it can be seen that the total distance traveled is shorter than the second image. However, the distance between the depot to destination 1 is very far. While the second image
has a total distance that is greater than the first image. However, the second image has a shorter total maximum distance between destinations than the first image. In the context of this study, the situation in the second image is preferable to the situation in the first image. When analogous to the case of an electrical vehicle, the vehicle does not need to invest in a large volume of battery because the stop, which also functions as a location for charging battery power sources, is affordable at a short distance.

Then the equation in the objective function which initially takes the form of minimizing the total distance can be converted into other forms to accommodate these needs. The form chosen is the minimization of the maximum distance between destinations. The equation can be seen in the following line.

$$\text{Min} \left( \text{Max}(c_{ij}x_{ij}) \right) \text{ for all } i, j$$

The mathematical model is then optimized using a brute force algorithm in python programming. The biggest difference in the objective function and comparison, can be seen in Figure 2.

```
def maxintour(tour):
    d=0
    da=[]
    for i in range(1,len(tour)-1):
        x1=cities[tour[i-1]][0]
        y1=cities[tour[i-1]][1]
        x2=cities[tour[i]][0]
        y2=cities[tour[i]][1]
        dis=distance(x1,y1,x2,y2)
        da.append(dis)
    x1=cities[tour[len(tour)-1]][0]
    y1=cities[tour[len(tour)-1]][1]
    x2=cities[tour[0]][0]
    y2=cities[tour[0]][1]
    dis=distance(x1,y1,x2,y2)
    da.append(dis)
    damax=numpy.max(da)
    return damax
```

**Figure 2.** Comparison of TSP Optimization Function

In the image to the left is a function to evaluate alternative solutions using a common TSP model. While the picture on the right is a model for TSP Bottlenecks used in this study. It can be seen at the end of the function, the value issued is the maximum value using the numpy.max method.

**Figure 3.** Plot of maximum distance minimization
Figure 3 shows the plot of the evaluation value of each alternative decision using the objective function designed in Figure 2. The left image shows the value of all iterations without exception. Meanwhile, the image on the right shows the optimal value plot. So that if in the next iteration no better evaluation results are obtained, the optimal value in that iteration is taken from the best previous iteration.

4. Discussion
Figure 4 shows the movement of the optimal value when the total distance is optimized, as is usually done in the TSP model. The left image shows the evaluation results for each iteration. Meanwhile, the right image shows the evaluation results for the best solution as far as the iteration goes.

Figure 5 shows the results of combining the plots of Figure 3 and Figure 4. The calculation of the total distance value has a wider range of values than the calculation of the maximum distance value. Meanwhile, the image on the right shows that the calculation of the total distance is faster to find the optimal solution than the maximum distance value.
Figure 6 shows the comparison between the movement of the total distance evaluation results for each iteration (green line), the movement of the optimum maximum distance value (red line), and the movement of the total distance when following the selection of the best alternative based on the maximum distance (yellow line). The yellow line indicates that the total distance cannot get the optimum value. The yellow line is locked at a certain value because according to the model used, there is a minimum optional solution based on the maximum distance, not based on the total distance. So that in some iteration processes a condition occurs where the green line has a lower value than the yellow line, but it is not chosen as the optimal solution because it produces a worse maximum distance.

5. Conclusion
After the research process is carried out, a mathematical model and programming are obtained to minimize the maximum distance between destinations on a route. The minimization results obtained have resulted in the inability to obtain the minimum global total distance. Even so, an acceptable and quite low solution was found. Minimizing the maximum distance between destinations on a route also causes the total distance to be lower, although it is not a global minimum. These results can be used for electrical vehicle route designers so as to minimize the need for battery capacity.

Some further research that can be done is to enlarge the scale of the problem into more destinations that need to be visited. This causes the need for another algorithm that is able to find better solutions in less iterations. Other research can also be carried out such as designing the optimal location for battery charging stations.

6. References
[1] S. Näher, “The travelling salesman problem,” in Algorithms Unplugged, 2011.
[2] M. Pustišek, A. Kos, and U. Sedlar, “Blockchain based autonomous selection of electric vehicle charging station,” 2018, doi: 10.1109/IHKI.2016.60.
[3] K. Sarrafan, D. Sutanto, K. M. Muttaqi, and G. Town, “Accurate range estimation for an electric vehicle including changing environmental conditions and traction system efficiency,” IET Electr. Syst. Transp., 2017, doi: 10.1049/iet-est.2015.0052.
[4] M. de L. Braga, A. de J. dos Santos, A. C.P. Pedroza, L. H. Luís, M. Dias de Amorim, and Y. Ghamri-Doudane, “Anytime route planning with constrained devices,” Comput. Electr. Eng., 2016, doi: 10.1016/j.compeleceng.2016.07.011.
[5] T. Jurik et al., “Energy optimal real-time navigation system,” IEEE Intell. Transp. Syst. Mag., 2014, doi: 10.1109/MITS.2014.232400.
[6] F. J. Jiménez-Espadafor, D. P. Guerrero, E. C. Trujillo, M. T. Garcia, and J. Wideberg, “Fully optimized energy management for propulsion, thermal cooling and auxiliaries of a serial hybrid electric vehicle,” Appl. Therm. Eng., 2015, doi: 10.1016/j.applthermaleng.2015.08.020.
[7] F. Alesiani and N. Maslekar, “Optimization of charging stops for fleet of electric vehicles: A genetic approach,” IEEE Intell. Transp. Syst. Mag., 2014, doi: 10.1109/MITS.2014.231419.
[8] C. Leake, D. N. P. Murthy, N. W. Page, and E. Y. Rodin, “Mathematical Modelling: A Tool for Problem Solving in Engineering, Physical, Biological and Social Sciences.,” J. Oper. Res. Soc., 1993, doi: 10.2307/2583902.
[9] H. Daellenbach, D. McNickle, and S. Dye, Management Science. 2012.
[10] H. Pratiwi et al., “Implementasi Algoritma Brute Force dalam Aplikasi Kamus Istilah Kesehatan,” urnal Ilm. Teknol. Inf. Terap., 2016.
[11] J. R. Ullmann, “An Algorithm for Subgraph Isomorphism,” J. ACM, 1976, doi: 10.1145/321921.321925.

Acknowledgement
We would like to thank Telkom University for providing resources and funds to carry out this research.