City-TSP with Objective Minimizing Distance and Noise

Rio Aurachman
Telkom University, Bandung, Indonesia
rioaurachman@telkomuniversity.ac.id

Abstract. Urbanization has encouraged people to have a home in the city. The process of transportation and logistics is also forced to pass through the city area. There is a need to formulate solutions to transportation, supply chain, and logistics problems for cities. One of the commonly used transportation models is the TSP, the Traveling Salesman Problem. Through this paper, we propose a City-TSP, which is a TSP that considers objectives, parameters, and constraints in urban areas. One of the parameters considered is the noise generated from a truck trip. This paper presents the City-TSP model that simultaneously minimizes distance and noise.

1. Introduction
The transportation system supports the economic processes carried out by humans. Human needs are different. Humans also have different abilities from one another. Based on these different abilities humans provide and provide different benefits. These differences give rise to the need to barter or exchange goods and services. Trade processes and economic transactions occur. The transaction is assisted by the transportation system.

Transportation itself is a waste. Waste must be eliminated or at least reduced. The transportation system is one of the objects of study in the field of Supply Chain Management and Logistics. One of the models used is the VRP (Vehicle Routing Problem)[1][2]. The Vehicle Routing Problem has a simple version, namely the TSP (Traveling Salesman Problem)[3]

With increasing urbanization, people shift their activities to urban areas. Production and consumption activities are carried out in the city area. This causes the transportation process, which initially took place outside the city, also took place in the city center. This situation raises the need for knowledge of urban logistics [4]. The urban logistics system has different stakeholders than other systems [5][6]. The existence of these different stakeholders raises different needs and issues. These differences in needs and issues cause the Objective and Constraints of the model to be different. One of the issues of concern is the noise generated by the transportation process. There is one paper that reviews all literature that researching about city VRP [7]

Each route has different noise implications. A route can produce the closest distance. However, these routes may produce noise that is too high. A model and algorithm are needed to help the decision-making process. The model and algorithm are expected to be able to simultaneously minimize mileage and at the same time minimize noise produced

2. Method and Algorithm
In addition to the distance matrix that is commonly used in the Traveling Salesman Problem, this research also involves the Noise matrix. Matrix noise has the same principle as the distance matrix. This matrix has dimensions according to the destinations involved in the Traveling Salesman Problem. For this case study using 7 destinations the matrix noise also has a dimension of 7 x 7. Each row
shows the origin and each column shows the destination. The number that becomes the intersection between the row and column is the noise number for when traveling from the row point to the column point. Based on this matrix, the system must find which route is chosen so as to produce the minimum distance and noise. Following in a row are distance matrix and noise matrix.

\[
\text{distance} = ([1000, 6, 4, 4, 20, 12, 8],
[6, 1000, 4, 6, 7, 12, 8],
[4, 4, 1000, 12, 8, 9, 4],
[4, 6, 12, 1000, 3, 5, 12],
[20, 7, 8, 3, 1000, 8, 8],
[12, 4, 9, 5, 8, 1000, 5],
[8, 12, 4, 12, 8, 5, 1000])
\]

\[
\text{noise} = ([1000, 26, 41, 42, 20, 85, 18],
[60, 1000, 41, 16, 56, 41, 12],
[14, 41, 1000, 12, 18, 91, 41],
[41, 15, 12, 1000, 31, 51, 12],
[20, 72, 16, 13, 1000, 18, 18],
[12, 24, 92, 35, 18, 1000, 15],
[81, 12, 42, 12, 38, 25, 1000])
\]

The system will work based on an algorithm to get the optimal solution. The algorithm can be illustrated as a process. The process can be described using a flowchart. Figure 1 shows the flowchart of the main process of this algorithm. As for each process described can be detailed to explain the sub-process. Each sub-process can be modeled in the form of a separate flowchart.

![Figure 1. Flowchart of Algorithm](attachment:image.png)

The process in the flowchart is divided into approximately three major groups. The first group is responsible for generating all possible tours. There are many algorithm alternatives to solve VRP or TSP [8][9][10], but for this research, we used the Brute Force algorithm. All tour possibilities are considered because the algorithm used is brute force. The brute force evaluates all possible alternatives to choose the one that produces the best performance.

The first group is the iteration process to find all the results of route permutations. This process requires the longest time compared to other processes. For example, a route with position 1,2,3,4,5,6,7
can be built permutations 3,6,1,5,2,4,7. All possibilities are counted and registered in an array. The route that has been obtained, then evaluated and found the best route in accordance with the objectives that have been determined.

The second group in charge is the process that functions to get the greatest value from the distance and matrix noise. The distance matrix and noise matrix have values with different scales. However, at the end of the evaluation process, distance and noise are summed. Then it is necessary to normalize the distance and noise based on the maximum value. Without normalization, it will cause one of the variables of high value to be prioritized. Though both of these objectives need to be achieved. Decision-makers can have a preference to prioritize one of the objectives to be optimized, whether it's distance or noise. However, the system must standardize and normalize the two objective variables. Here is the programming code for calculating maximum noise and maximum distance from the matrix.

```python
def maxvalue(array):
    jar=np.array(array)
    jar2=np.where(jar==1000, 0, jar)
    max=np.amax(jar2)
    #min=np.amin(distance)
    return max
```

In getting the maximum value from a matrix, you can use the numpy package, namely numpy.amax. However, in the distance matrix and noise matrix, there is a Big M parameter that functions as a constraint. So that the route does not depart and go to the same destination, the diagonal value of the matrix needs to be set as a large number. The number will automatically be selected when using numpy.amax to get the maximum value. Then it is necessary to eliminate 1000 values. This is done specifically for the process of calculating the maximum value. The process is carried out with the numpy.where command which temporarily changes the number 1000 to 0.

![Figure 2. Flowchart of Optimum Solution Algorithm](image_url)
The third group is a process that functions to evaluate all routes, as shown in figure 2. Route evaluation is based on the measurement of the total distance and total noise of a route. After that, the two distance and noise calculations are combined into the final score. After that, the system will evaluate whether the resulting score is better than the previous solution score. If yes, then the score and the route being processed will be used as the best route and best temporary score. The next step is to continue the route evaluation process if there are still routes that have not been evaluated.

```python
for i in range(len(allpossibletour)):
    l=a*totaldistancetour2(allpossibletour[i])/mj+b*totalnoisetour2(allpossibletour[i])/mn
    if l<lbest:
        dbest=copy.copy(totaldistancetour2(allpossibletour[i]))
        nbest=copy.copy(totalnoisetour2(allpossibletour[i]))
        lbest=copy.copy(l)
        ibest=copy.copy(i)

def totaldistancetour2(tour):
    d=0
    for i in range(0,len(tour)-1):
        d=d+distance[tour[i]][tour[i+1]]
    d=d+distance[tour[len(tour)-1]][tour[0]]
    return d
```

(2)

3. Result and Discussion

After the development of algorithms and mathematical models, the model of the algorithm and model is tested. To produce the final score, a combination of distance score and noise score is combined. The two scores are summed by giving the weighting of the two scores first. Weights are determined based on decision-maker preferences.

![Comparison of Distance and Noise](image)

**Figure 3. Comparison of Distance and Noise based on Weight**

As seen in Figure 3, if we run the model with the minimum coefficient of noise, the optimum solution is obtained, the noise generated is high and the distance calculated is low. However, if the chosen noise coefficient is high, and of course results in the distance coefficient being low, the model will recommend an optimum solution with a high distance, but low noise.
### Table 1. Total Score Calculation

| a  | b  | Distance | Noise | Normalized Distance | Normalized Noise | Score   |
|----|----|----------|-------|---------------------|------------------|---------|
| 1  | 0  | 31       | 257   | 0.221428571         | 0.39906832      | 0.221428571 |
| 0.9| 0.1| 32       | 195   | 0.228571429         | 0.30279503      | 0.235993789 |
| 0.8| 0.2| 32       | 195   | 0.228571429         | 0.30279503      | 0.243416149 |
| 0.7| 0.3| 32       | 195   | 0.228571429         | 0.30279503      | 0.250838509 |
| 0.6| 0.4| 38       | 144   | 0.271428571         | 0.22360248      | 0.252298137 |
| 0.5| 0.5| 38       | 144   | 0.271428571         | 0.22360248      | 0.247515528 |
| 0.4| 0.6| 38       | 144   | 0.271428571         | 0.22360248      | 0.242732919 |
| 0.3| 0.7| 50       | 120   | 0.357142857         | 0.1863354       | 0.23757764  |
| 0.2| 0.8| 50       | 120   | 0.357142857         | 0.1863354       | 0.220496894 |
| 0.1| 0.9| 66       | 106   | 0.471428571         | 0.16459627      | 0.195279503 |
| 0  | 1  | 66       | 106   | 0.471428571         | 0.16459627      | 0.164596273 |

As shown in Table 1, distance and noise have different values, so that if the two scores are added directly, the system will prioritize the variables with the highest maximum value. To prevent this, normalization is done first for the value of distance and noise. Each distance and noise value is made to be a maximum of 1 and a minimum of zero. After the normalization, the next step is running the code to get the optimum solution. Can be seen in figure 4, the movement of the distance, noise, and total score by changing the noise weight.

![Comparison of Normalized Distance and Noise](image)

**Figure 4.** Comparison of Normalized Distance and Noise based on Weight

### 4. Conclusion

Based on research that has been done, models and algorithms have been found to get the optimum traveling salesman problem solution by considering objective distance minimization and noise minimization. Both objectives are achieved simultaneously.

Before the model and system work, the decision-maker first needs to determine how much weight is set for the two variables. If according to the decision-maker, noise priority is higher than distance, the noise weight is set higher. If the distance priority is higher than noise, the distance weight is set higher.
To make the resulting solution not biased, then the value of the distance and noise needs to be normalized first. The two parameters have different maximum and minimum values so that when calculated together it will cause an imbalance from the optimum solution.

Some further research can be done by considering other constraints on TSPs with an Urban context. City TSP can continue to grow with increasing urbanization. Some constraints that can be used are to consider congestion in the city and consider the comfort of the city dwellers.

References

[1] Eksioglu B, Vural A V, Reisman A 2009 The vehicle routing problem: A taxonomic review Computers & Industrial Engineering 57(4) pp. 1472-1483.

[2] Laporte G 2009 Fifty years of vehicle routing Transportation science 43(4) pp.408-416.

[3] Näher S 2011 The travelling salesman problem In Algorithms unplugged Springer Berlin Heidelberg pp. 383-391

[4] Jiang G, Lam S K, Ning F, He P and Xie J 2019 Peak-Hour Vehicle Routing for First-Mile Transportation: Problem Formulation and Algorithms IEEE Transactions on Intelligent Transportation Systems.

[5] Kim G 2018 Optimal Vehicle Routing for Parcel Delivery with Considering Two Time Periods In 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) pp. 918-922.

[6] Erdoğan S and Miller-Hooks E 2012 A green vehicle routing problem Transportation research part E logistics and transportation review 48(1) pp.100-114.

[7] Kim G, Ong Y S, Heng C K, Tan P S, and Zhang N A 2015 City vehicle routing problem (city VRP): A review IEEE Transactions on Intelligent Transportation Systems 16(4) pp.1654-1666.

[8] Kaewman S and Akararungruangkul R 2018 Heuristics Algorithms for a Heterogeneous Fleets VRP with Excessive Demand for the Vehicle at the Pickup Points, and the Longest Traveling Time Constraint: A Case Study in Prasitsuksa Songkloe Ubonratchathani Thailand Logistics 2(3) p.15.

[9] GONG J L and CHENG Z M 2006 The Realization of Genetic Algorithm of VRP Based on the Matlab [J] Logistics Sci-Tech, 7.

[10] James J Q, Yu W and Gu J 2019 Online vehicle routing with neural combinatorial optimization and deep reinforcement learning IEEE Transactions on Intelligent Transportation Systems 20(10) pp.3806-3817.