Item Delivery Simulation Using Dijkstra Algorithm for Solving Traveling Salesman Problem

To cite this article: Hagai Nuansa Ginting et al 2019 J. Phys.: Conf. Ser. 1201 012068

View the article online for updates and enhancements.
Item Delivery Simulation Using Dijkstra Algorithm for Solving Traveling Salesman Problem

Hagai Nuansa Ginting\textsuperscript{1,2}, Andrew Brian Osmond\textsuperscript{2} and Annisa Aditsania\textsuperscript{3}

\textsuperscript{1,2} Department of Electrical Engineering, Telkom University, Bandung, Indonesia.
\textsuperscript{3} Department of Informatic Engineering, Telkom University, Bandung, Indonesia.

E-mail: \textsuperscript{1}hagaiginting@student.telkomuniversity.ac.id, \textsuperscript{2}abosmond@telkomuniversity.ac.id, \textsuperscript{3}aaditsania@telkomuniversity.ac.id

Abstract. Companies that contribute to travel have many problems in the process of item delivery. Distances and priorities are considered for a process of item delivery based on the highest priority. A delivery target that can be done one day evidently exceed the expected limit and that is the impact. This is an example of the waste time and operational costs that should be at the same time that two or more addresses can be sent. Traveling Salesman Problem (TSP) was define a classical problem to finding the shortest route that salesman can be passed when visiting several places without visit again in the same place more than once. In this study, TSP requires all calculations of possible routes to be obtained. Then choose one of the shortest routes by prioritizing the things considered, namely distance and priority. Delivery is done quickly through the shortest route according to priority using the Dijkstra algorithm. Simulation shows that the Dijkstra algorithm must be approved by use clustering data for Dijkstra’s priorities and sub-routes to solve TSP problems. The resulting route has an influence between two graphs. Complete graph has a distance efficiency of 47.8\% and execution time of 48.1\% compared to non-complete graphs.

1. Introduction

Item delivery is an important industry as technology develops. In this case, the companies that contribute to delivery have many problems in the process of item delivery, one of example is time and distance that are not optimal as well as the priority of an item that must take a first over several existing items. The high level of delivery from year to year must be considered in the distribution of item. Viewed from large travel companies such as Item Express, Uber and others, they only deliver from one place to one destination and have not implemented a multi-destination delivery system.

In this research, we will try a new feature in the process of item delivery. This feature is a presentation of item delivery system in one order to multiple destinations. Then, the seller can visit several places without having to visit same place more than once \cite{4}. In general, to go to a destination can be reached through several tracks. In this case, we can determine which places or roads have to be passed so that can be find the shortest destination. Thus the shortest path can be interpreted as the minimum weight of all trajectories.

Traveling Salesman Problem (TSP) is a classic problem for finding the shortest route that salesmen can pass when they want to visit several places without having to go to the same city more than once \cite{4} \cite{1}. Completion in this method requires calculation of all possible paths/routes that can be obtained, then selecting one of the shortest paths \cite{5}\cite{1}. In determining the shortest path can be
obtained by mathematical algorithms, including Dijkstra's algorithm, Greedy, Floyd-Warshall, Ant-Colony, Genetics, and others [3]. The algorithm that will be solved in this system is Dijkstra's algorithm. The Dijkstra algorithm is used to determine the shortest path based on the smallest weight from one node to another using the Cartesian diagram [2]. To use the Dijkstra algorithm in the case of the Traveling Salesman Problem can use a complete graph where each node is connected to all other nodes. In this algorithm, it has a disadvantage, there is no visit of all nodes in the graph due to the characteristic of the algorithm is a single source shortest path algorithm. The Traveling Salesman Problem is the most important problem, so that makes modifications in the algorithm. This modification is changing the form of Dijkstra's sub-routing in the algorithm and adds cluster analysis to the priority part of the data [10]. The Dijkstra algorithm will perform calculations on all variables where the smallest weight of each node so that each place will be calculated the distance traveled by each track [2].

2. Dijkstra’s Algorithm

Dijkstra’s algorithm is often known as the single source shortest path algorithm. The Dijkstra algorithm is used to determine the shortest distance on a directed and undirected graph. An example of the application of Dijkstra’s algorithm is the shortest path that connects two locations, different places (single source single-destination shortest path problem). The Dijkstra algorithm requires parameters of place of source and destination. The final result of this algorithm is the shortest distance from the place of source to destination and its route. [11]. Dijkstra's algorithm using to determine the shortest path from a graph, it will find the best path because when determining the path to be selected we will analyze the weights of the vertices that have not been selected, then select the vertex with the smallest weight. The Dijkstra algorithm looks for the shortest distance from the source vertex to the closest vertex, then to the second vertex. [4].

The pseudo code of Dijkstra algorithm is shown below. Two sets of vertices are used in the graph: T is a set of temporarily labeled node; λ is value of node of T, P is the set of permanently labeled vertices. When λ (v) is minimum, among the vertices in T, vertex V is chosen as shown in Line 5. This vertex moves from T to P, and every edge outgoing from v is tested again. If the end-vertex u of e is in T, λ(u) in Line 10 is calculated, if e enable such an improvement. If u is new, it gets a label in Line 12 and joins T in Line 13 [3].

a. Steps for algorithm

Procedure Dijkstra(G,s,l;λ)
1 λ(s) ← 0
2 T ← {s}
3 P ← Ø
4 while T ≠ Ø
5 choose a vertex v ∈ T for which λ(v) is minimum
6 T ← T \ {v}
7 P ← P ∪ {v}
8 for every v → u do
9     If u ∈ T then do
10       λ(u) ← min{λ(u), λ(v) + l(e)}
11     else, if u ∈ P then do
12       λ(u) ← λ(v) + l(e)
13     T ← T ∪ {u}

Proof: Every vertex enters T at most once, and every vertex chosen in Line 5 leaves T. The performance of Lines 5 takes at most O(|V|) time, and the sum total time to perform Lines 8 -13, for all chosen vertices, is O(|E|), since no edge is examined more than once. After performing Line 5|V| times, T must be empty, and the procedure halts. It follows that time complexity of Dijkstra’s algorithm is O(|V|^2 + |E|). We shall return to this issue [3].
3. Problem Modeling with Dijkstra’s Algorithm

Modifications will be made in such a way to resolve the Traveling Salesman Problem. Algorithm modification is additions destination nodes, sub-routing Dijkstra, and clustering for each search to finding the shortest route [8][12].

In addition to the destination node, the program will generate a random numbers from the priority value ranges that have been clustered into 3 parts. Cluster 1 shows that the high priority at that node has a weight of 3, cluster 2 indicates that the priority is medium which is weighted 2 and also the third cluster with priority which is weighted 1. The program will execute the first cluster value or the largest priority value with the destination node and generate random number at each node in that priority. The sub-routing used is a repetition of the Dijkstra algorithm that occurs if all destination node has not been reached. As in Figure 1 which is the flowchart of a solution to the TSP problem using the modified Dijkstra algorithm.

4. Simulation And Analysis

The simulation will carry in twenty (20) nodes. The coordinates of these nodes are input into a Microsoft Office Excel software as well as the node value for each priority. The following Table 1 shows 20 simulated data nodes on the system. The distance between nodes is calculated by using the Euclidean distance [7].

| Nodes | X Coordinate | Y Coordinate | Priority |
|-------|---------------|--------------|----------|
| 1     | 22            | 10           | 2        |
| 2     | 34            | 5            | 3        |
| 3     | 23            | 32           | 1        |
| 4     | 12            | 10           | 3        |
| 5     | 40            | 30           | 3        |
Nodes | X Coordinate | Y Coordinate | Priority
---|---|---|---
6 | 21 | 50 | 3
7 | 24 | 46 | 1
8 | 13 | 45 | 1
9 | 43 | 23 | 2
10 | 34 | 23 | 2
11 | 31 | 20 | 3
12 | 9 | 34 | 3
13 | 2 | 1 | 1
14 | 23 | 2 | 2
15 | 43 | 8 | 1
16 | 35 | 45 | 2
17 | 50 | 29 | 2
18 | 31 | 29 | 2
19 | 40 | 12 | 2
20 | 23 | 45 | 3

**Complete Graph**

Complete graph is a graph in which each node is connected by an edge to each other node.

Figure 2 is a complete graph that is connected between all nodes of the neighbor nodes. The coordinates of nodes and priorities have been input into an Excel file as in Table 1 and represented as a graph.

Running simulations will do three times each starting node and will be compared with the results of different running graphs. The following Figure 2 shows the simulation results with the starting node are 6 to all nodes using complete graphs. There are 3 types of priorities that are in Table 1, the priority that is visited first is a priority with a value of 3. Nodes that have priority weight 3 are 2,4,5,6,11,12,20, nodes with priority 2 are 1,9,10,14,16,17,18,19, the node with priority 1 are 3,7,8,13,15. Initializing the destination is a random number from the priority being run. In this simulation the starting node is 6, then the next destination will generate a random number in priority 3 so that the next node will be the next starting node. The program will look for the highest priority and see all possible shortest routes using the Dijkstra algorithm.
The program results in Matlab show the route to be traveled, that route is 6-20-12-11-5-2-4-1-14-19-10-9-17-16-18-3-7-8-15-13 with the initial start node is 6, distance = 473,8309, and the program execution time is 0.804866 seconds.

The results shown by the program output are recommendations using the Dijkstra algorithm using the type of complete graph and initial start node that have been initialized as node 6. Non-complete graphs, taking routes based on the highest priority to the lowest. Node 20,12,11,5,2,4 is a cluster of priority 3, so this node will be visited first. After all nodes in priority 3 are visited, node 1,14,19,10,9,17,16,18 which is a cluster of priority 2 will be visited. Same with priority node 1 which is 3,7,8,15,13, so the distance will be traveled by the salesman to send an item of all nodes based on priority is 473,8309 with program execution time of 0.80 seconds.

**Non-complete Graph**

The next simulation uses non-complete graphs.

Figure 3 shows a graph that has an edge not connected to several nodes called a non-complete graph. The simulation uses the same coordinates and priorities as Table 1 and has been input into Microsoft Office Excel. The segment value (edge) that is not connected and connected are generated by a program with using connected edge probabilities smaller than 0.7 (p(x) < 0.7) and included in the adjacency matrix.
The following below Figure 4 shows the simulation results with the starting node is node 1 to all node using non-complete graphs. From the explanation on the previous graph, the coordinates and priority nodes the same used. Initialize the destination node as well as random numbers from the priority being run.

In this simulation the starting node is 1, then for the next destination, a random number will be generated in priority 3 so that the next node will be the next starting node. The program will look for the highest priority and see all possible shortest routes using the Dijkstra algorithm.

![Figure 5. Recommendation route in non-complete graph.](image)

The program results in Matlab show the route to be traveled, namely 15-2-19-5-20-6-12-10-11-4-1-9-18-14-16-17-7-3-13-8 with the initial node is 5, distance = 989,2471, and with program execution time 1,660490 seconds.

From the test results, the results of the program output are recommended using the Dijkstra algorithm by using a type of non-complete graph and the initial node that has been initialized is node 15. In non-complete graphs, the route is based on the highest priority to the lowest. Node 2,5,6,12,11 is a cluster of priority 3, so this node will be visited first, but node 19 has priority value 2. Because there are several edges that are not connected, then to be able to visit nodes on priority the highest must visit the node that has the same path to arrive at the destination node. The destination node will generate a random number from the highest priority. For example the result of the first generate is node 5 with the starting node is 2, because 2 with 5 is not connected, the program will see the node path connected to it, there is node 19 connected so that the path to be recommended is 2,19,5 and like that soon so that all nodes are visited. This will give the lowest priority possible to visit earlier. So that the distance taken by the salesman to send an item of all nodes based on priority is 989,2471 with program execution time of 1.660490 seconds.

**Dijkstra Without Modification**

Searching for optimization routes using Dijkstra's algorithm has an influence on algorithm modifications. This is due to the use of clustering data for each priority value used and generate random numbers. The Comparison will be doing by a complete graph without priority and without random numbers for the destination node.
The results in matlab like figure 6 show 2 as the starting node followed by the route that will be passed, that is 2-19-15-9-18-10-11-14-13-12-8-6-7-20-16-3-17, distance = 223.55, and with program execution time 2.101 seconds.

The results of the test are the most optimal results from each node tested in a complete graph. It can be seen that the value without modification is more optimal than modified. This is because the route with the modified dijkstra algorithm does not use priority and randomization nodes so that it is pure with the dijkstra algorithm. This statement also applies to non-complete graph types.

**Effect of Graph Type**

Figure 7 shows the comparison of distances between the two graphs. Non-complete graphs have a higher distance than complete graphs. This is because the edges of non-complete graphs are not connected, so the recommended route will be different. Also marked with the results of the destination node generated randomly [9].
Figure 8 shows a comparison of the time needed for program execution. In the value of non-complete graphs, it takes more time to run the program, this is indicated by modifications to the Dijkstra algorithm and randomly generated numbers so that the estimated time changes.

5. Conclusion
Modeling of item delivery systems for services can be carried out by displaying the results of the route and distance. To get these results, the clustering method is used in priority values. The difference in the graph used in the simulation has an impact on the route, distance of recommendation and execution time. Complete graph with simulation distance results of 473.83 is more optimal than non-complete graphs with a distance of 989.24 with an efficiency of 47.8%. Complete graphs have as many as 190 edges that can be traversed so that many possible optimal routes will be traversed. The execution time is affected by the number of edges, in complete graphs with 190 edges the program has 0.8 seconds execution time while the non-complete graph with 131 edges has an execution time of 1.66 seconds and an efficiency of 48.1%. So that more edges have less execution time.
The route on the program is the optimal route using the Dijkstra algorithm. It is said to be optimal because the search process in the algorithm includes all values in clustering in priority. The use of priority values influences the optimization process. Recommended routes without priority not use clustering for priority distribution. Therefore nodes that are scrambled as end node are not only certain priorities but all nodes.
6. References

[1] M.F.A Syahputra; Riri N.D; Sherly A.S; K.C Kirana “Implementation of Traveling Salesman Problem (TSP) based on Dijkstra's Algorithm in the Information System of Delivery Service”, JAVA, International Journal of Electrical and Electronics Engineering, Vol 14, Number 1, April 2016
[2] “Dijkstra’s algorithm,” Wikipedia, 5 August 2018. [Online]. Available : https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm. [Accessed 5 August 2018].
[3] Shimon Even, Guy Even. 2012. “Graph Algorithms 2nd Edition”. Cambridge : Cambridge University Press.
[4] Applegate, D. L.; Bixby, R. M.; Chvátal, V.; Cook, W. J. (2006), The Traveling Salesman Problem, ISBN 0-691-12993-2
[5] A. R. Amin, M. Ikhsan dan L. Wibisono, “Traveling Salesman Problem”.
[6] Potvin, J. Y. (1996). Genetic algorithms for the traveling salesman problem. Annals of Operations Research, 63, 337–370. https://doi.org/10.1007/BF02125403
[7] Sedgewick, R., & Vitter, J. S. (1986). “Shortest paths in euclidean graphs”. Algorithmica, 1(1–4),31–48. https://doi.org/10.1007/BF01840435
[8] M.M. Mena, R.H. Ucan, V.C. Vetina, and F.M. Ramirez, “Web Services Competition using the Bidirectional Dijkstra Algorithm” IEEE Latin America Transaction, vol 14, issue 5, May 2016.
[9] S.S. Biswas, B. Alam and M.N. Doja, "Generalization of Dijkstra’s Algorithm for Extraction of Shortest Paths In Directed Multigraphs " Journal of Computer Science, vol. 9, no.3, pp. 377-382, 2013.
[10] Liu L., (2011), "Data Model and Algorithms for Multimodal Route Planning with Transportation Networks", Technische Universität München, pp. 9-13.
[11] Nilsson, C. (2003)."Heuristics for the traveling salesman problem”. Linkoping University, 3–8. https://doi.org/10.1016/S0305-0548(98)00085-9
[12] R. Munir, “Matematika Diskrit”, Bandung: Informatika Bandung, 2010.
[13] A Y.Y. Sheng, “The Role of Transportation in Logistics Chain”, Proceedings of the Eastern Asia Society for Transportation Studies, Vol.5, pp. 1657 - 1672, 2005
[14] Kleinberg J., Tardos É., (2006), "Algorithm Design", Pearson Education Inc., pp. 73-78.
[15] Peyer, S., Rautenbach, D., & Vygen, J. (2009)."A generalization of Dijkstra’s shortest path algorithm with applications to VLSI routing". Journal of Discrete Algorithms, 7(4), 377–390. https://doi.org/10.1016/j.jda.2007.08.003
[16] M.M. Mena, R.H. Ucan, V.C. Vetina, and F.M. Ramirez, “Web Services Competition using the Bidirectional Dijkstra Algorithm” IEEE Latin America Transaction, vol 14, issue 5, May 2016
[17] Vaibhavi Patel and Prof. ChitraBaggar, “A Survey Paper of Bellman-Ford Algorithm and Dijkstra Algorithm For Finding Shortest Path in GIS Application”, International Journal of P2P Network Trends Technology(IJPTT), Vol. 5, February 2014.
[18] Huang, Y., Yi, Q., & Shi, M. (2013). An Improved Dijkstra Shortest Path Algorithm. 2nd International Conference on Computer Science and Electronics Engineering, (Iccsee), 226–229
[19] M. Agung Prasetyo, R. Latuconsina, Tito W. Purboyo."Shortest Path Algorithms : State of the Art". Bandung : Research India Publication. International Journal of Applied Engineering Research ISSN 0973-4562 Vol.12, Number 23(2017) pp.13610-13617.
[20] Bott, R. (2014). Using Dijkstra Algorithm in Calculating Alternative Shortest Paths for Public Transportation with Transfers and Walking - Case Study: Ankara. Igarss 2014, (1), 1–5. https://doi.org/10.1007/s13398-014-0173-7.2.