Application of Dijkstra’s Algorithm and Hamiltonian Cycle in Third Party Logistics

Bethea Zia Davida 13515084[1]
Informatics Engineering Study Program
School of Electrical Engineering and Informatics
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*betheazdavida@gmail.com*

***Abstract*—3PL companies have to visit a lot of cities to deliver logistics. Each city is connected to the other city by road, railway, or airplanes. To reduce costs, they have to find the most efficient route to go to each city that they need to visit. We develop a solution due to this problem based on Dijkstra’s Algorithm and Hamiltonian Cycle.**

***Keywords*—Dijkstra’s Algorithm, Graph, Hamiltonian Cycle, Hamiltonian Graph, Traveling Salesman Problem**

I. Introduction

Third party logistics (3PL) is a provider of outsourced logistics services that cover anything that involves management of the way resources are moved to the areas where they are required to be. These kind of companies are important part of today’s supply chain. Many 3PL companies offer many kind of services including; inbound freight, freight consolidation, warehousing, distribution, order fulfillment, and outbound freight.

Nowadays, 3PL has become more important in logistic sector. Companies tend to use 3PL’s sevice because they do not want to deal with logistics issues and problems. It is also because they want to reduce the costs.

In the 3PL, we have to consider the supply chain. We have to find the most efficient and shortest way to deliver the logistics, so we can save the costs. We address the routing problem that faced by 3PLs provider in planning day-of-week delivery routes for their customers who need to make deliveries to their destination. The route will be the most important issue which may charge great costs to the company if it is neglected.

The goal here is to minimize the total cost for the transportation and inventory while satisfying the customers. This problem that we get to reach that goal also known as Traveling Salesman Problem. And we offer Hamiltonian Cycle and Dijkstra’s Algorithm as the solution for this problem which can give us the most efficient route as we wish.

We can take an example in Java island. The problem is to find the most efficient way send the logistics to each city in Java island.

Picture 1. Java Island
http://islandoanialisme.blogspot.co.id/2014/12/java.html

II. Theory

A. Graph

1. Definition

Graph is a set of points and lines that connect some of the points. The points in graph are also known as vertices or nodes. And the lines that connect the vertices knwon as edges or arcs. The edges of some graphs may be directed or undirected.

In formal terms, graph is an ordered pair of G = (V,E), where V is a set of points, vertices, or nodes and E is a set of edges or arcs.

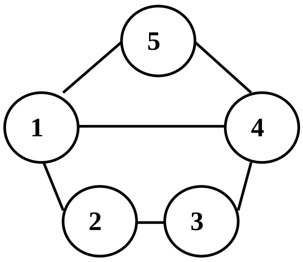
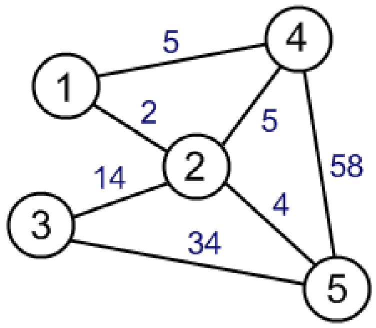


Diagram 1. Example of Graph

2. Weighted Graph

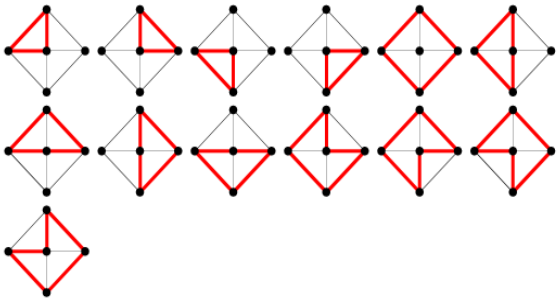
Weighted graph is a graph where the edges is given a numerical weight. A weighted graph is also a special type of labeled graph in which the labels are numbers.



Picture 2. Example of Weighted Graph
http://web.cecs.pdx.edu/~sheard/course/Cs163/Doc/Graphs.html

3. Graph Cycle

A graph cycle is a subset of the edge set in the graph that forms a path and first node of the path corresponds to the last node.



Picture 3. Example of Graph Cycle
http://mathworld.wolfram.com/WheelGraph.html

B. Hamiltonian Cycle

Hamiltonian cycle is a graph cycle that trough the graph and visit each nodes in the graph exactly with the ending point in the one it is started. The example of the Hamiltonian cycle is the cycle graph.

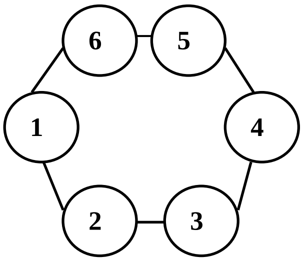
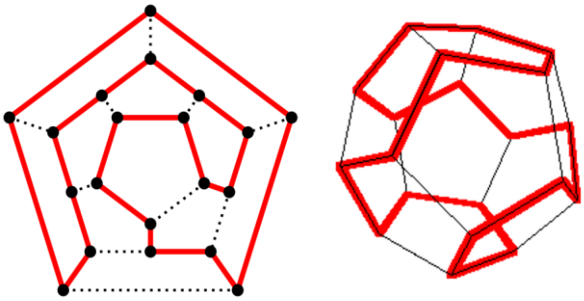


Diagram 2. Cycle Graph

Hamiltonian cycle is named after Sir William Rowan Hamilton who invented mathematical game, Hamilton's puzzle in 1857. The game's goal is to find a Hamiltonian cycle along the polyhedron edges of an dodecahedron so that every vertices, which named with different capital city, is visited single time. The ending point of the game have to be the same with the starting point. This game solved using Icosian calculus, so this game also known as Icosian game.

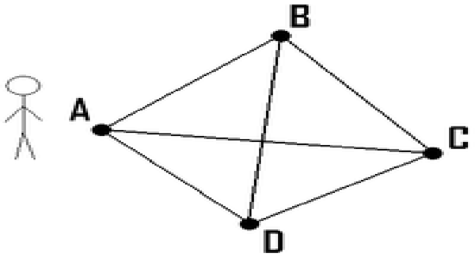
Picture 4. The Icosian Game
http://puzzlemuseum.com/month/picm02/200207icosian.htm



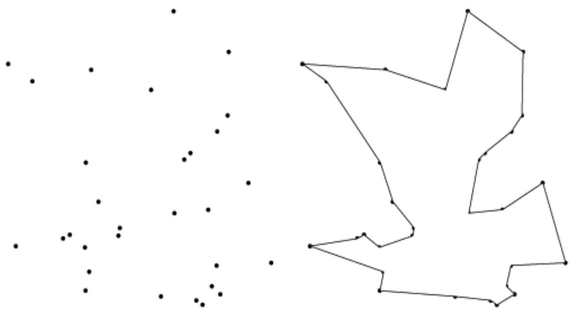
Picture 5. Hamiltonian Cycle in The Icosian Game
https://thatsmaths.com/2012/12/20/santas-tsp-algorithm/icosian-game-2d3d/

C. Travelling Salesman Problem

The travelling salesman problem is one of graph theory problem that requiring the mot efficient Hamiltonian cycle a salesman can take to go to each of *n* cities. The basic idea of Travelling Salesman Problem is to find the shortest route for the salesman so he can visit a number of cities and return to his starting point in the end of the route. The travelling salesman prbolen is typical of hard optimization problem.



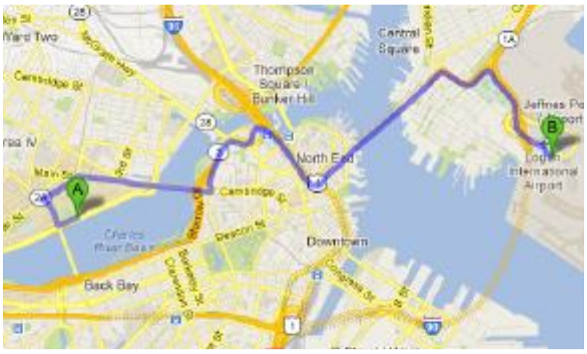
Picture 6. Salesman has to visit each city
https://simple.wikipedia.org/wiki/Travelling_salesman_problem



Picture 7. Travelling Salesman Problem in *N* Cities http://mathworld.wolfram.com/TravelingSalesmanProblem.html

D. Dijkstra's Algorithm

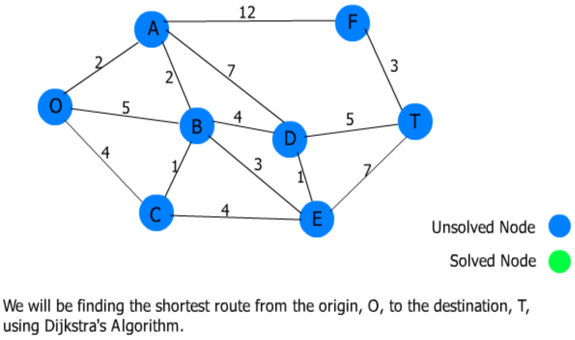
Dijkstra's Algorithm is named after Edsger Wybe Dijkstra. Dijkstra's Algorithm is an algorithm to find the shortest path from a point in a graph to a destination. Application of Dijkstra's algorithm in our daily life is to get to another place with the shortest route. In picture 7, we can find the shortest way from A to B with Dijkstra's Algorithm.



Picture 8. Application of Dijkstra's Algorithm in Our Daily Life *math.mit.edu/~rothvoss/18.304.3PM/Presentations/1-Melissa.pdf*

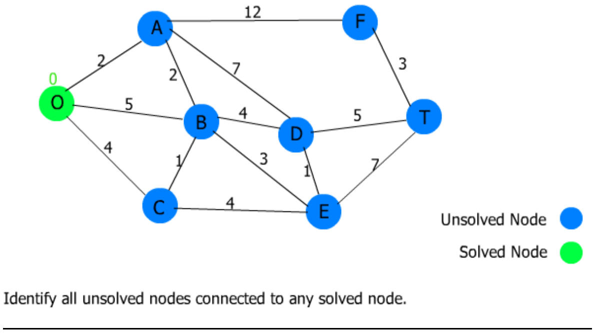
The idea of the algorithm is to continiously calculate the shortest distance beginning from a starting point, and to exclude longer distances when making an update. It consists of the following steps:

1. Initialization of the starting node with 0
2. Marking the distance of the starting node as permanent and other distances as temporarily.
3. Calculate the temporary distances of all neighbour nodes of the starting node by summing up its distance with the weights of the edges.
4. If the calculated distance of a node is smaller than the current one, update the distance and set the current node as antecessor. This step is also called update and is Dijkstra's central idea.
5. Setting of the node with the minimal temporary distance as starting node. Mark its distance as permanent.
6. Repeating of steps 4 to 7 until there aren't any nodes left with a permanent distance, which the neighbour nodes still have temporary distances.



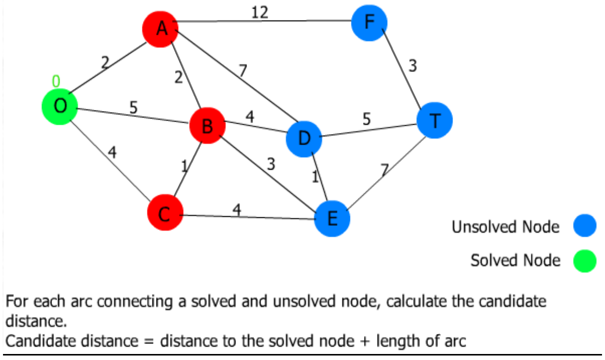
We will be finding the shortest route from the origin, O, to the destination, T, using Dijkstra's Algorithm.

Picture 9. Dijkstra Algorithm Step http://optlab-server.sce.carleton.ca/POAnimations2007/DijkstrasAlgo.html



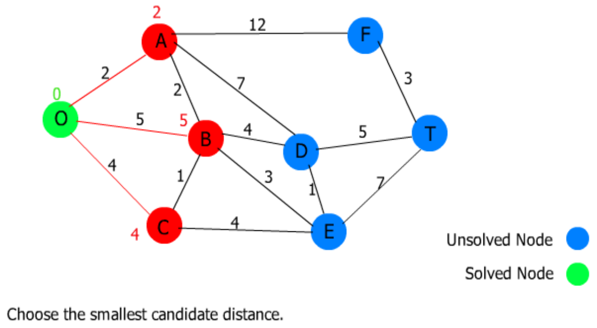
Identify all unsolved nodes connected to any solved node.

Picture 10. Dijkstra Algorithm Step http://optlab-server.sce.carleton.ca/POAnimations2007/DijkstrasAlgo.html



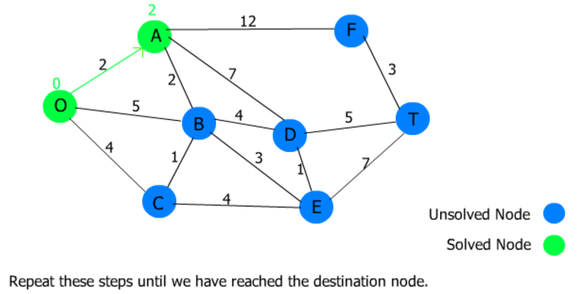
For each arc connecting a solved and unsolved node, calculate the candidate distance.
Candidate distance = distance to the solved node + length of arc

Picture 11. Dijkstra Algorithm Step http://optlab-server.sce.carleton.ca/POAnimations2007/DijkstrasAlgo.html



Choose the smallest candidate distance.

Picture 12. Dijkstra Algorithm Step http://optlab-server.sce.carleton.ca/POAnimations2007/DijkstrasAlgo.html



Repeat these steps until we have reached the destination node.

Picture 13. Dijkstra Algorithm Step http://optlab-server.sce.carleton.ca/POAnimations2007/DijkstrasAlgo.html

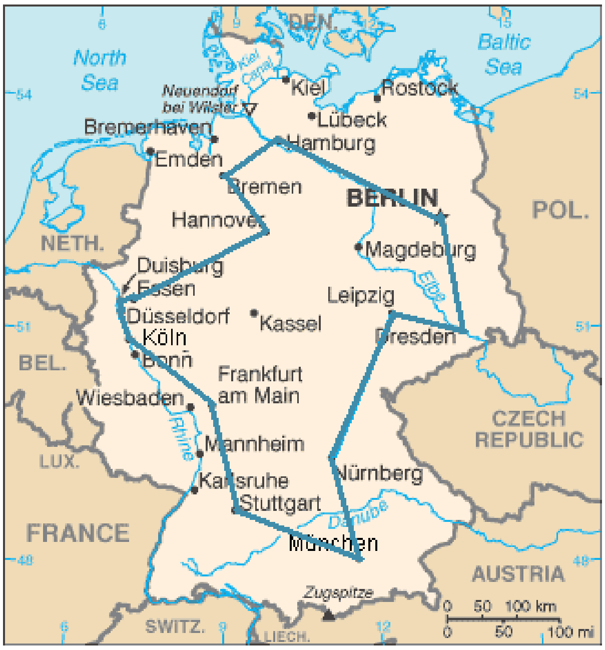
Dijkstra's Algorithm's Pseudocode :

```
function Dijkstra(Graph, source):
    for each vertex v in Graph:
        dist[v] := infinity
        previous[v] := undefined
    dist[source] := 0
    Q := the set of all nodes in Graph
    while Q is not empty:
        u := node in Q with smallest dist[ ]
        remove u from Q
        for each neighbor v of u:
            alt := dist[u] + dist_between(u, v)
            if alt < dist[v]
                dist[v] := alt
                previous[v] := u
    return previous[ ]
```

III. APPLICATION OF DIJKSTRA'S ALGORITHM AND HAMILTONIAN CYCLE

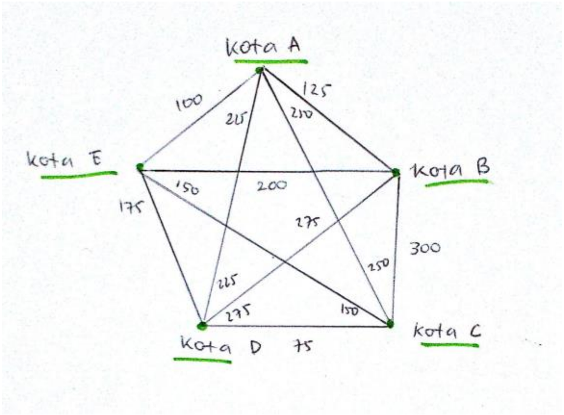
In 3PL, we have to find the most efficient route that go through each city once for sending the logistics and back to the starting city as in the travelling salesman problem. We can implement Djikstra's algorithm and Hamiltoninan cycle to solve it, which is as we know, hamiltonian cycle can fulfill that requirement.

Each city is connected to the other city by road, airplanes, or railway. So we have to make a list of the city that we need to visit and the distance between the cities. From the list, we will use Djikstra's algorithm and Hamiltonian cycle to make the route.



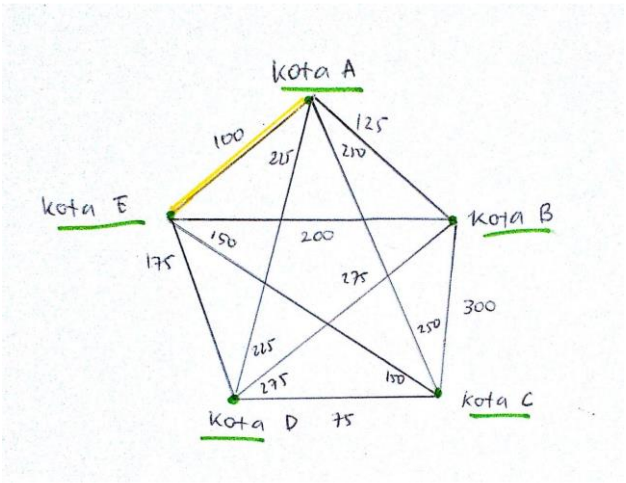
Picture 14. Optimal route of a salesman visits the 15 bigget cities in Germany https://simple.wikipedia.org/wiki/Travelling_salesman_problem

Dijkstra's algorithm is used to find the shortest distance from one city to the another, which when we want to go from a city to another there are many ways, as we have the distance between the city. While the Hamiltonian cycle is used to find the route to visit each city and return to the starting city. Picture 15 is the example of the travelling salesman problem that can be solve with Dijkstra algorithm and Hamiltonian cycle. In that picture, we want to visit Kota E, Kota A, Kota B, Kota C, and Kota D.

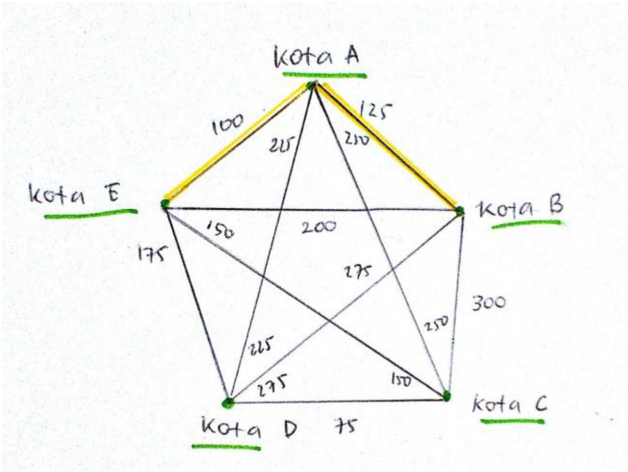


Picture 15. Example of Problem

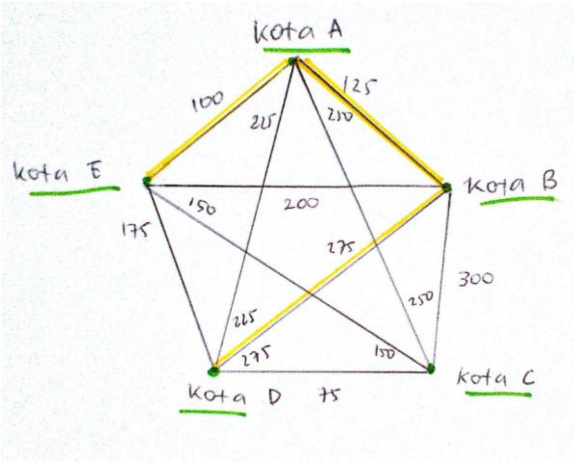
Our starting point is Kota E. From Kota E to Kota A, with Dijkstra's algorithm, we can find the shortest way to go there, so do from Kota A to another city.



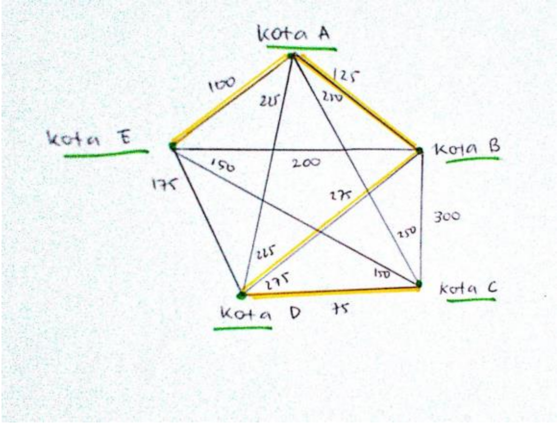
Picture 16. Dijkstra's algorithm and Hamiltonian cycle application



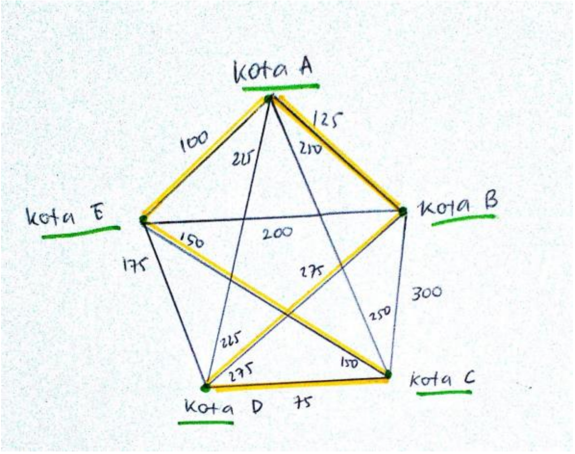
Picture 17. Dijkstra's algorithm and Hamiltonian cycle application



Picture 18. Dijkstra's algorithm and Hamiltonian cycle application



Picture 19. Dijkstra's algorithm and Hamiltonian cycle application



Picture 20. Dijkstra's algorithm and Hamiltonian cycle application

And at the end, it will return to Kota A again as we wish with Hamiltonian cycle.

This example can be implemented in the real cities for 3PL delivery plan before it delivers the logistics.

IV. CONCLUSION

Dijkstra’s algorithm and Hamiltonian cycle can be implemented to find the most efficient route for third party logistics delivery plan.

V. ACKNOWLEDGMENT

First and above all, the writer praises God, the almighty for providing me this opportunity and granting me the capability to proceed successfully this paper. In performing this paper, the writer had to take the help and guideline of some respected persons and resources. The writer would like to show her gratitude to Mr. Rinaldi Munir and Mrs. Harlili, lecturers of IF2120 Matematika Diskrit, Institut Teknologi Bandung, for giving the writer a good guideline for the paper. The writer would also like to expand deepest gratitude to all those who have directly and indirectly guided the writer in writing this paper.

REFERENCES

[1] Rinaldi Munir, *Matematika Diskrit*. Bandung: Penerbit Informatika, 2012, ch. 8.

[2] *Dijkstra, E. W.* (1959). *"A note on two problems in connexion with graphs"* (PDF). *Numerische Mathematik*. **1**: 269–271

[3] https://www.thebalance.com/selecting-a-third-party-logistics-3pl-provider-2221230 accessed Dec. 7, 2016

[4] http://mathworld.wolfram.com/Graph.html accessed Dec 7, 2016

[5] http://mathworld.wolfram.com/TravelingSalesmanProblem.html accesed Dec 7, 2016

[6] http://mathworld.wolfram.com/HamiltonianCycle.html accesed Dec 8,2016

[7] http://www.gitta.info/Accessibiliti/en/html/Dijkstra_learningObject1.html accesed Dec 8, 2016

[8] http://mathworld.wolfram.com/GraphCycle.html accesed Dec 8,2016

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 9 Desember 2016

Bethzia

Bethea Zia Davida 13515084