ANNALS of Faculty Engineering Hunedoara
– International Journal of Engineering
Tome XV [2017] – Fascicule 4 [November]

ISSN: 1584-2665 [print; online]
ISSN: 1584-2673 [CD-Rom; online]
a free-access multidisciplinary publication
of the Faculty of Engineering Hunedoara

[1.]Ramë LIKAJ, [2.]Ahmet SHALA

APPLICATION OF GRAPH SEARCH ALGORITHM DIJKSTRA TO FIND OPTIMAL SOLUTION FOR THE PROBLEM OF TRANSPORT

[1-2.] University of Prishtina, Faculty of Mechanical Engineering, Prishtina, KOSOVO

Abstract: Graph theory is used for modelling of real life systems and finding optimal solutions in different networks systems such are: transport, water, electricity, internet, work operations schemes in the process of production, construction, etc. Although these systems are different, however they have some common features and are in the relation between each other. In this paper is analyzed one practical problem to find a shortest and maximal path between two or more points by using graph search algorithm Dijkstra. Also, for this case was developed a network model of the transportation problem which is analyzed in detail to minimise shipment costs and the results have been compared by Solver in Excel.
Keywords: Modeling, dynamics, robot, analysis

1. INTRODUCTION TO GRAPH THEORY

Graph theory provides many useful applications in Operations Research. A graph is defined as a finite number of points (known as nodes or vertices) connected by lines (known as edges or arcs).

Our task is for given graph by using graph search algorithm Dijkstra to find the shortest path between two points. There are different path options to reach from node A to node G, but our aim is to find the shortest path with minimum transportation costs, which requires a lot efforts. In order to compare the results the search algorithm Dijkstra is used to find maximal distance between node A and G, too.

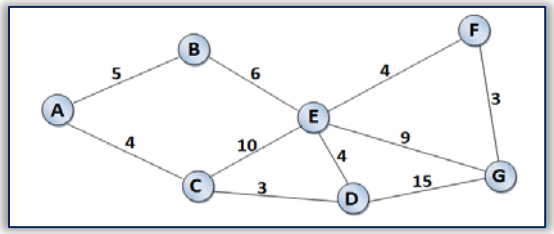

Figure 1. Connected Graph

2. ALGORITHM DIJKSTRA: The shortest path from node A to node G

By using Dijkstra algorithm we are able to find the shortest distances (length of arc) from a node to all other nodes. Some of the options are:

- ACDG 22
- ABEG 20
- ACEG 23

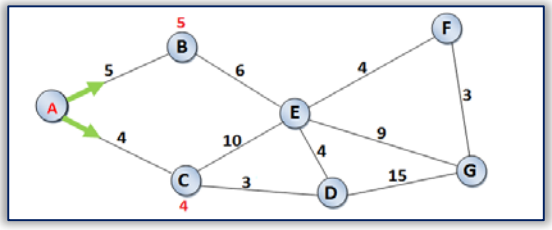

Figure 2. The shortest path from node A (B and C)

Firstly, we start from the node A, which is chosen as permanent node. Analysing the distances of the neighbourhoods' nodes of the node A, we are able to find the shortest path to nodes B and C (their distances are equal with 4 and 5). Afterwards the node C is chosen as permanent node since is the shortest distance, and we have to check after the distances from node C to the neighbour nodes. To the each neighbour

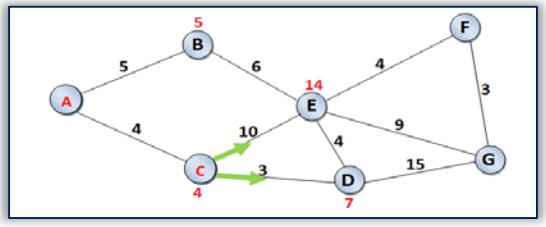

Figure 3. The shortest path from node C (D and E)

node is added the length of the permanent node. Node C with length 4 is chosen as permanent node. We chose the shortest distance from the node A. Node B is chosen as permanent node with the shortest distance 5 and the procedures requires observing the distance of neighbours nodes. To each neighbour nodes (node E) is added the distance of the permanent node, in this case the distance from the nodes ABE is shorter than ACE (11<14), this means that distance ACE is not considered any more.

We chose the shortest distance from the node A. Node D is chosen as permanent node with the shortest distance 7 and the procedures requires observing the distance of neighbours nodes. To each neighbour nodes (node E and G) is added the distance of the permanent node.

We chose the shortest distance from the node A. Node E is chosen as permanent node with the shortest distance 11 and the procedures requires observing the distance of neighbours nodes. To each neighbour nodes (nodes G and F) is added the distance of the permanent node.

We chose the shortest distance from the node A. Node F is chosen as permanent node with the shortest distance 15 and the procedures requires observing the distance of neighbours nodes. To each neighbour node (node G) is added the distance of the permanent node.

From the permanent node F by adding distances 15+3=18, is shown that 18 < 20, this means that distance 20 is not considered any more. The same procedure is followed by subtracting the distance of neighbour nodes from node F.

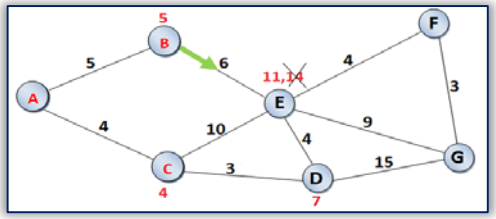

Figure 4. Permanent node B

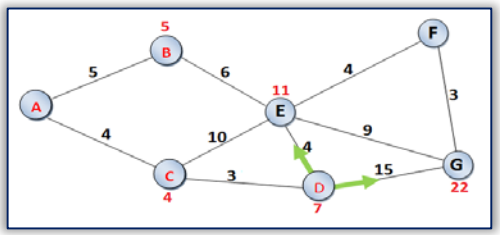

Figure 5. The shortest path from node D (E and G)

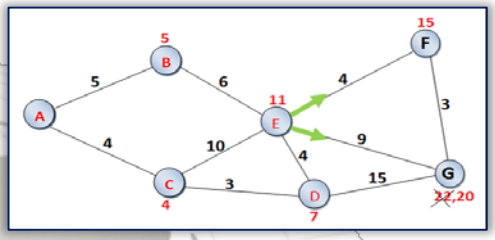

Figure 6. The shortest path from node E (F and G)

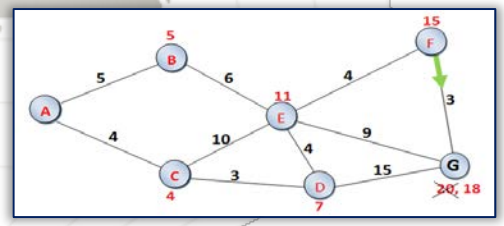

Figure 7. Permanent nodes D, E and F

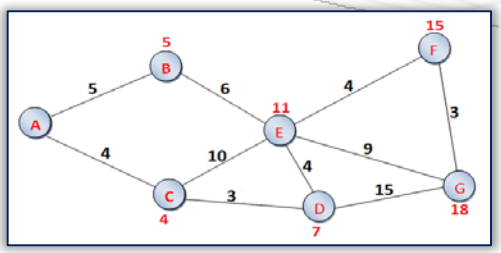

Figure 8. The shortest distance nodes ABEFG

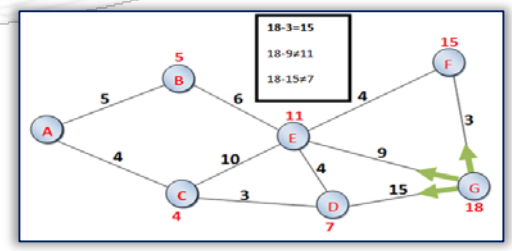

Figure 9. Final destination node G

3. CALCULATION OF MINIMAL COST OF TRANSPORTATION PATH A-G

The starting point is node G by subtracting from this vertex the distance of each neighbour node.

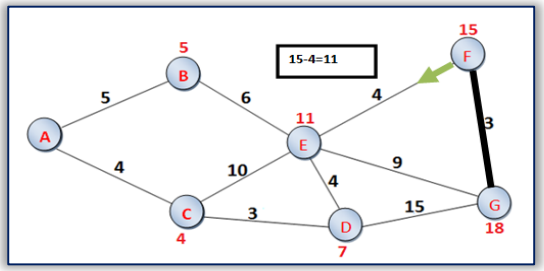

Figure 10. Correct option from node F to node E

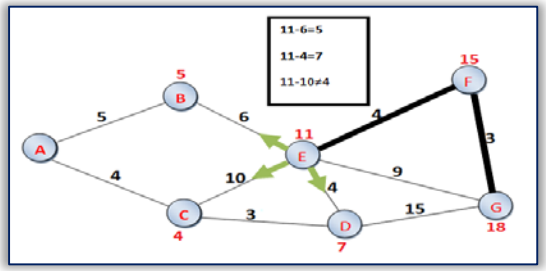

Figure 11. Analysis of options for node E

In this case there are two options available and we have to choose between them (11-4).

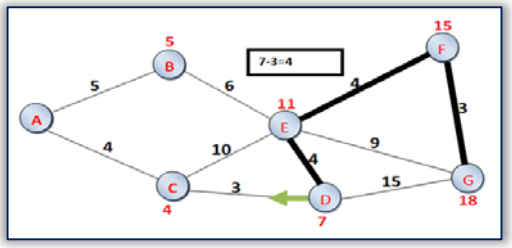

Figure 12. Correct option for node E

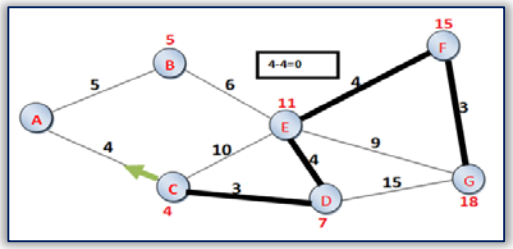

Figure 13. Analysis of options for node D

The shortest path between nodes A and G is 18 (ACDEFG).

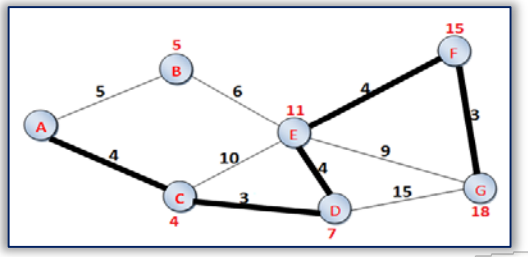

Figure 14. The shortest path from node G to node A

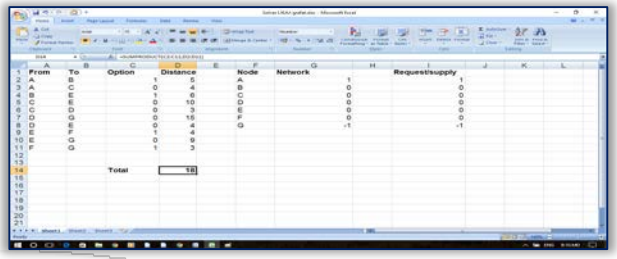

Figure 15. Results obtained in solver/excel

4. MAXIMUM DISTANCE FROM NODE A TO G

There are different options to find the maximal distance between nodes A and G and in this case is used graph search algorithm Dijkstra. Similar procedure is followed to find maximal distance. Initially node A is chosen as permanent node and in this context we analyse distances of neighbour nodes B and C.

The longest distance from node A to neighbour nodes (B and C) is 5. Now the permanent node is chosen node B and the procedure requires observing the distance of the neighbour nodes. To each neighbour node is added the distance of the permanent node.

The longest distance from node A is chosen. Now the permanent node is chosen node E with the longest distance of 11 and the procedure requires observing the distance of the neighbour nodes (nodes F, G and D). To each neighbour node is added the distance of the permanent node.

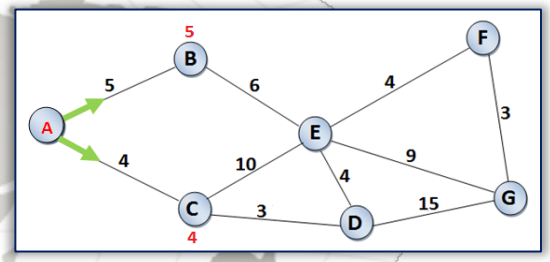

Figure 16. Permanent node A

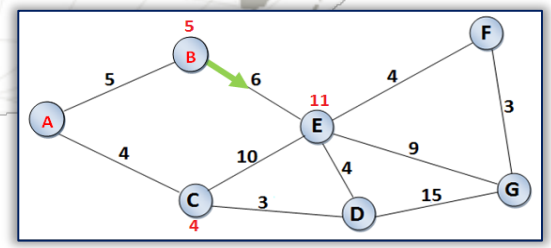

Figure 17. Analysis of options for node A (nodes B and C)

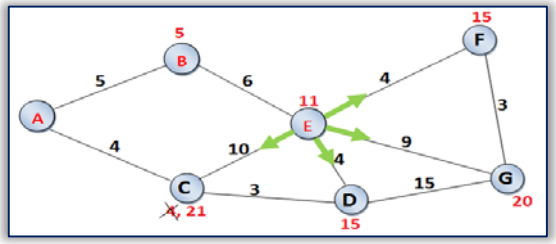

Figure 18. Analysis of options for node E

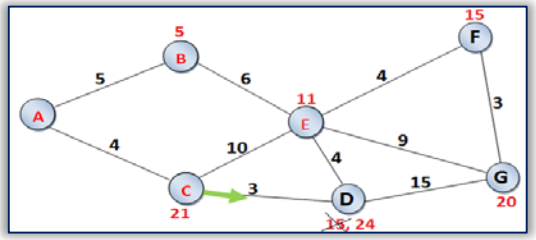

Figure 19. Permanent node C

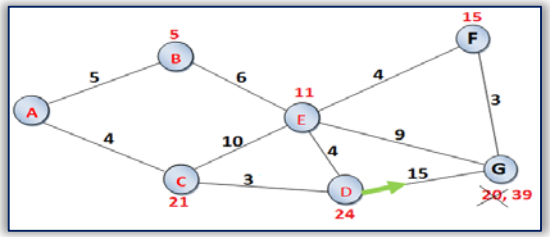

Figure 20. Permanent node D

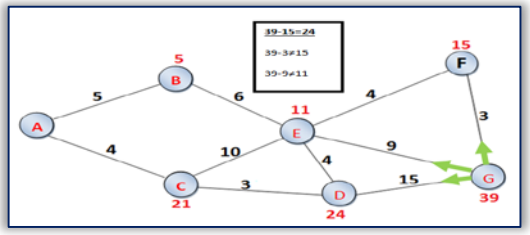

Figure 21. Calculation of longest path A-G

The longest distance from node A is chosen. Now the permanent node is chosen node C with the longest distance of 21 and since the distance of vertices AC<ABEC, then the distance 4 is not considered any more. To each neighbour node is added the distance of the permanent node.
The longest distance from node A is chosen. Now the permanent node is chosen node D with the longest distance of 24 and since the distance of vertices ABEC<ABECD, then the distance 15 is not considered any more. To each neighbour node is added the distance of the permanent node.

5. CALCULATING THE LONGEST DISTANCE OF NODES A-G

The starting point is node G by subtracting from this vertex the distance of each neighbour node.

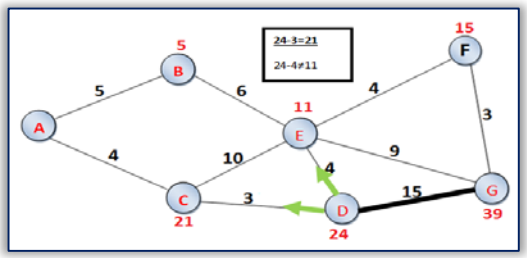

Figure 22. Analysis of options for node D (nodes C & E)

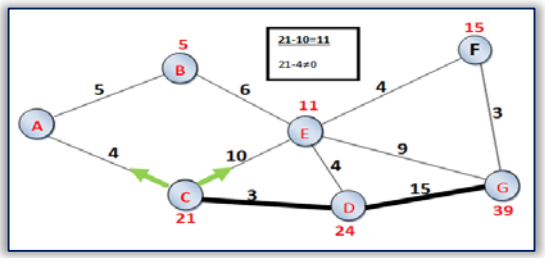

Figure 23. Analysis of options for node C (nodes A & E)

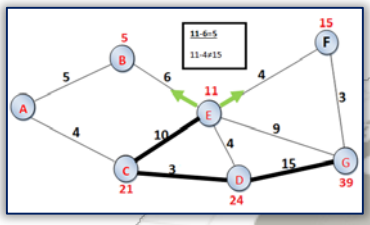

Figure 24. Analysis of options for node E

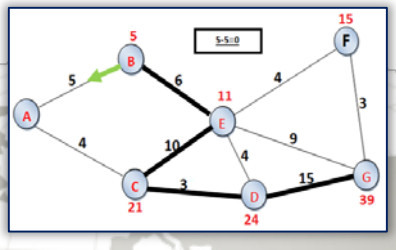

Figure 25. Correct option node E

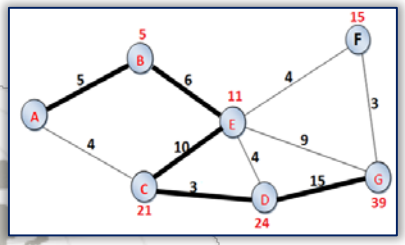

Figure 26. Final destination node A

The longest distance is path ABECDG with length equal to 39.

6. CONCLUSIONS

Dijkstra's Algorithm is used to find the shortest path between two nodes/vertices in our case from node A to node G. Applying the rules of Dijkstra's Algorithm for the shortest path we have obtained two possible solutions with the same result 18; path ABEFG and ACDEFG. The same results have been confirmed by using powerful Excel tool Solver. The results which have been obtained for the given example shows that graph search Dijkstra's Algorithm is very effective optimisation tool for solving practical problems (in our case used to find the path with lowest cost from node A to node G). By using the same procedure which is described in this paper we calculated the longest distance between nodes A and G which is the worst scenario. From the obtained results we can conclude that this option to reach node G from node A is approximately 54% more expensive from the first case.

Acknowledgements: The first author is profoundly thankful to the corresponding author Ahmet Shala which has paid attention to fulfill all requirements about this research work.

References

[1] Likaj, R. (2009), Application of Graph Theory to find Optimal Paths for the Transportation Problem 15th Workshop on International Stability, Technology, and Culture The International Federation of Automatic Control June 6-8, 2013. Pp 235-240.
[2] Likaj, R. (2009), The problem of transport in designing the production systems, pages 175-180, MITIP, Bergamo, Italy.
[3] Likaj, R. (2010), Sensitivity analysis in designing the production systems and economic interpretation, MOTSTP, International Scientific Conference Management of Technology, Zagreb Croatia.
[4] Murthy, P. (2007), Operations Research_ 2nd Edition_2, New age International publishers, USA.
[5] Grotchel, M (2010), Linear optimirung, Technische Universität Berlin Institut für Mathematik.
[6] Alt, H. (2011), Lineare Programmierung, Technische Universität Ulm.
[7] Schonmann, F. (2004),Lineare Programmierung, Technische Universität Bielefeld.
[8] McCarl, B. (2015), Solving Linear Programs in Excel, Texas A&M University, USA.
[9] Reeb, J. and Leavengood, S. (2002), Transportation Problem: A Special Case for Linear Programming Problems .
[10] Shehu, Sh. (2004), Kërkime Operacionale", Polytechnic University of Tirana.
[11] Pula J. (2005), Metodat e transportit të programimit linear, University of Pristina.