International Journal of Innovations in Engineering and Technology (IJIET)
http://dx.doi.org/10.21172/ijiet.163.02

Use of Linear Programming for solving Maximum Flow and Minimum Cut for Traffic Control Problems

NajibTanish[1], Vladimir Saso[1], Borivoje Milosevic[2]
[1]University ALFA BK Belgrade,
11000 Belgrade, Serbia
[2]Faculty of Business and Law, 11000 Belgrade, Serbia

Abstract- The main objective of this paper is to solve the problem of maximum traffic flow in a single congested road network using the Ford-Fulkerson algorithm. The problem of traffic flow maximization can be abstracted by the use of linear programming and graph theory where vertices are source and destination locations and edges of directed graphs are elements that connect these traffic nodes with their parameters which can be: capacities, flows, distances, etc. In this scenario, the capacity of vertices can be mapped as the capacity of roads, which can be defined as the maximum number of vehicles that can simultaneously use the road without stopping due to congestion. Therefore, reducing traffic congestion by optimizing maximum vehicle flow is mainly the main objective of this paper. To optimize this problem, we implement an algorithm called the "Ford-Fulkerson algorithm" that provides key parameters that can help reduce traffic congestion in homogeneous traffic and find the optimal route. The basic idea of this paper is also to introduce methodology and simulation of traffic scenarios as well as to look at their impact on traffic congestion using MATLAB application.

Keywords –Traffic congestion, Capacity, Distance, Maximum flow,Minimum cut, Bottleneck, Shortest path.

I. INTRODUCTION

Most major cities suffer from congestion problems, one of the main causes of congestion being a sudden increase in traffic during certain hours, mainly in areas where bottlenecks occur. Current solutions in the literature are based on spotting traffic and diverting vehicles to avoid congested areas. However, they do not take into account the impact of a traffic change in the near future. Thus, these approaches cannot provide a long-term solution to the congestion problem, because when alternative paths are suggested, they create new bottlenecks on paths closer to congestion, so that the problem shifts from one point to another. Recently, with the implementation of standards for traffic networks and advances in wireless communication technologies, the implementation of Intelligent Transport Systems-ITS [1] has become possible. ITS is a comprehensive real-time traffic management solution that relies on collecting data from vehicles, Road Sides Units-RSUs and other sensors that as entities can interact and interact, creating a reliable traffic network [2]. ITS can create patterns based on online data collection, such as traffic density, speed, travel time, etc., and then using patterns that the ITS network detects, controls and minimizes traffic congestion [3], [4] [5], [6].

However, the main challenge is to predict congestion and divert vehicles without causing new congestion elsewhere. Traffic jams have been a problem over the years, as most major cities suffer from this problem, and the bandwidth of roads at different levels, highways, etc., is experiencing major bottlenecks and congestion. According to reports, there are two major sources of congestion in the urban environment. The first is the event that affects traffic flow, such as incident cases, work zones and weather. Others are traffic needs related to fluctuations in normal traffic and special events. Finally, reports show that bottleneck is the main cause of congestion with 40%, followed by incident events such as car accidents with 25%, bad weather with 15%, work zones 10%, poor timing of traffic signals and special events with 5% each, Fig. 1.

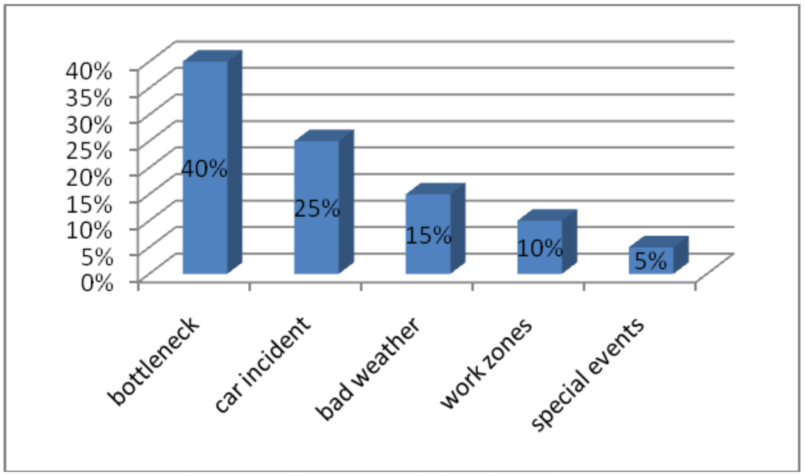

Figure 1. Traffic congestion in the urban environment

Detecting and classifying traffic congestion is not an easy task because of the vast amount of information provided by vehicles and road conditions such as speed, location, time and events (eg accidents, signaling problems, bottlenecks at critical hours). In this scenario, there are approaches that focus on detecting traffic congestion [7], [8] and others that focus on classifying traffic congestion using the concept of fuzzy logic, linear programming, etc.

The pilot project will analyze only one shaded part of the displayed set of elements that can affect traffic congestion, Fig. 2, with parameters: current congestion, capacity, direct graph, unit source and destination. Ford Fulkerston algorithm will be applied, maximum flow and minimum cross section calculated and labeled congestion sites.

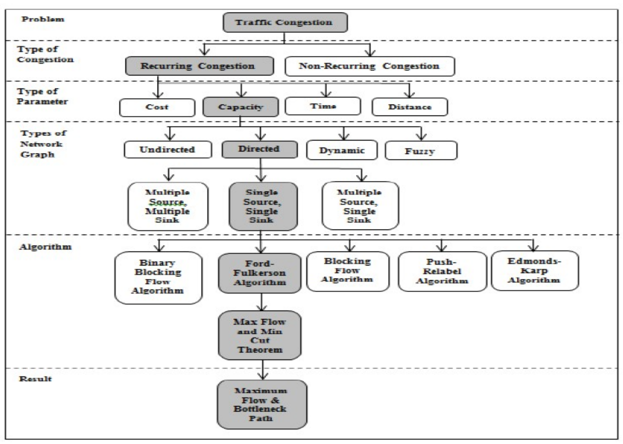

Figure 2. Traffic congestion elements

II.LINEAR PROGRAMMING AND FORD-FULKERSON ALGORITHM

Linear Programming (LP) is a widely used mathematical technique designed to help managers plan operations and make the decisions necessary to allocate resources. Linear programming is a simple technique that describes complex relationships and which, through linear functions, simplify and find optimal points for specific conditions. Linear programming is used to obtain the most optimal solution to problems with given constraints. In linear programming, we formulate our real life problem into a mathematical model. This involves objective function and linear inequalities with some restrictions.

The Max-Flow problem and Min-Cut problem can be formulated as two primal-dual linear programs.The equality in the Max-Flow Min-Dut theoremfollows from the strong duality theorem in linear programming, which states that if the primalprogram has an optimal solution, x^*, then the dual program also has an optimal solution, y^*, such that the optimal values formed by the twosolutions are equal.

The max-flow LP is straightforward. The dual LP is obtained using the algorithm described in dual linear program. The resulting LP requires someexplanation. The interpretation of the variables in the min-cut LP is:

Common terminologies used in linear programming can be defined as:

- Variable Decisions: These are the variables that will decide the output. They are the final solution to the problem. To solve any problem, we must first identify the decision variables.
- Objective function: defined as a target function for making decision.
- Constraints: Constraints are restriction or limitation on a decision variable. They usually limit the value of a decision variable.
- Nonnegative restriction: for all linear programs, decision variables should always take non negative values. This means that values for decision variables should be greater than or equal to 0.

A linear program can provide a solution by choosing multiple methods. In this paper, we will not use a graphical method to solve a linear program. We will use MATLAB script elements to solve our problem and apply the Ford Fulkerston Max Flow Min Cut algorithm. The Ford-Fulkerson algorithm is an algorithm that tackles the Max Flow Min Cut problem. That is, given a network with vertices and edges between those vertices that have certain weights, how much "flow" can the network process at a time? Flow can mean anything, but typically it means data through a computer network. The Ford-Fulkerson algorithm assumes that the input will be a graph, G, along with a source vertex, s, and a sink vertex, t. The graph is any representation of a weighted graph where vertices are connected by edges of specified weights. There must also be a source vertex and sink vertex to understand the beginning and end of the flow network.

A common question in network theory is "what is the largest flow between a particular node and another node in the network"? When the system is mapped to the grid, the pull-arches represent flow channels with limited capacity. To find the Max Flow, we assign the flow to each arc in the network so that the total simultaneous flow between the endpoint of two nodes is as large as possible.Logically speaking, we can define flow as [20]:

flow(u,v) ∀ edge(u,v) – flow for each edge with some restrictions

- *Capacity constraints:*∀ edge (u,v)

$$0 \le flow(u,v) \le capacity(u,v)$$

$\forall (u, v) \varepsilon E \ f(u, v) \le c(u,v)$

(The flow along an edge cannot exceed its capacity.)

- *Skew symmetry*: ∀ vertex v ≠ s, t

$$\sum_{u \varepsilon in(v)} flow(u,v) = \sum_{w \varepsilon out(v)} flow(v,w)$$

$$\forall (u, v) \varepsilon E \ \ f(u, v) = - f(u, v)$$

(The net flow from u to v must be opposite of the net flow from u to v.)

- *Flow conservation:* $|f| = \sum_{w \varepsilon out(s)} flow(s,w) = \sum_{u \varepsilon in(t)} flow(u,t)$

$\forall u \varepsilon V: \ u \ne s$ *and* $u \ne t \Rightarrow \Sigma f(u, w) = 0.$

*Lemma 1.*Let G = (V; E), and f be the flow in G. LetG_f be residualgraph induced by S_fLet f*be the flow inG_f . Then f + f*is the flow in G, $| f + f^* | = | j | + | f^* |$.
Rezidual path is now any path between s and t in Gf.
Residual path capacity p is:

$$p: \ c_f(p) = \min\{ c_f(u;v) \,|\, (u;v) \varepsilon p\}$$

*Lemma 2.*Let G = (V;E), and f be the flow in G, p path between s and t in Gf .
Definef_p : V x V → R $_s$

$$f_p(u,v) = \begin{cases} c_f(p) & (u;v) \varepsilon p \\ -c_f(p) & (v;u) \varepsilon p \\ 0 & \text{in other cases} \end{cases}$$

Then fp is the flow in Gf.
Using these lemmas we prove an important theorem:

Theorem 1. Let G = (V; E), and f be the flow in G, p is the path between s and t in Gf.
Let fp be defined as in Lemma 2.We define f *: V x V → R with f'* = f + fp.
Then f *is the flow in G, │f*│ = │ f│ + │ fp │> f.

III. Max Flow Min Cut

Ford-Fulkerson algorithm can be applied in other fields like liquid flow through pipe, information in communications network, current flow in electrical circuit, production factory with various tools and others. We presented applied minimum cut maximum flow using cut set of a weighted graph to the traffic flow. A weighted graph was a resulting graph with a real number which served as a structural model in transportation. The traffic control strategy of minimal cut and maximum flow was to minimize the number of edges in a network and maximum capacity of vehicles which moved through these edges. With a minimal cut in the traffic network, it allowed minimum waiting time of traffic participants for a smooth and uncongested traffic flow. The main objective was to localize the pipeline bottleneck of a transport network. The algorithm was used to find the minimum cut and maximum flow of the network. Dual problem involved to find bottlenecks and the flow value could be less than or equal to maximum flow. The finding was the sum of capacity constraints of edges for such a cut could be larger than the minimum of the dual problem. The proposed algorithm could effectively find out the bottlenecks in constant inflows and outflows condition.

The traffic networks must be formed into the map of graph theory before identification of bottleneck. We applied the Max-flow Min-cut Theorem[22] to find out the bottleneck of the network. This theorem stated that the minimum cut was the smallest capacity of the road section. Therefore, it determined the maximum capacity of the whole network. The identified weak parts of the road allowed traffic planar to know that which parts of the road needed to be widened. They further proposed a way to solve traffic bottleneck which was to increase road lines. Simulation software, MATLAB was applied to perform simulation on the new network with which road line was added to test the efficiency of the solution. The results showed that identification of bottleneck based on Max-flow Min-cut Theorem could thus find out the bottleneck effectively. Max flow and min cut is used to solve:

- Two very rich algorithmic problems.
- Cornerstone problems in combinatorial optimization.
- Beautiful mathematical duality.

And nontrivial applications / reductions.

Data mining.	Network reliability.
Open-pit mining.	Network connectivity.
Project selection.	Distributed computing.
Airline scheduling.	Security of statistical data.
Bipartite matching.	Egalitarian stable matching.
Baseball elimination.	Image segmentation.
Network intrusion detection.	Multi-camera scene reconstruction.

IV.Practical Results

In this particular scenario, the edge elements can be mapped as road capacities (number of vehicles per minute), which in turn can be defined as the largest number of vehicles that can simultaneously use the route without stopping due to congestion. So, with all these mappings, we are able to apply the Ford-Fulkerson algorithm to determine a set of optimized vertex that suggest the selection of paths where traffic congestion sites should be eliminated. In all this, we can use them as traffic alerts that result in maximum flow and less traffic congestion. This gives rise to the main values of these parameters and it is clear that using these parameters, traffic jams can be reduced to some extent.

In this study, traffic volume and distances as two types of data were collected. Traffic volume can be collected by two main methods. It could be either manual traffic count or automatic traffic count. Manual traffic count would take the least cost, but required more manpower. Manual traffic count was selected in this study because it was considered as one of the user friendly methods.

The collected traffic volumes are defined in total, for all traffic vehicles. Besides, distances were needed in this study to form the weighted network graph. The distances data were collected from the Google Map. The distances can be obtained by entering the address of a starting point and a destination using the link on Google Map.

Data preparation for traffic flow in units of vehicle/min was performed from the database of the Public Enterprise Roads of Serbia - http://www.putevi-srbije.rs/index.php, Table 1. The database was downloaded for 2018 and contains data on the average annual daily traffic on public roads of the 1st and 2nd order in the Republic of Serbia. In our specific case, some characteristic sections were selected from the road map of Serbia.

Table 1. Traffic Flow (vehicle / min)

	vehicle/min	Nis	Pojate	Razanj	Krusevac	Paracin	Kraljevo	Batocina	Kragujevac	Velika Plana	Mladenovac	Smederevo	Mali Pozarevac	Kovin	Pancevo	Beograd
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Nis	-	11	11	-	-	-	-	-	-	-	-	-	-	-	-
2	Pojate	-	-	-	5	-	-	-	-	-	-	-	-	-	-	-
3	Razanj	-	-	-	-	12	-	-	-	-	-	-	-	-	-	-
4	Krusevac	-	-	-	-	-	5	-	-	-	-	-	-	-	-	-
5	Paracin	-	-	-	-	-	-	14	-	-	-	-	-	-	-	-
6	Kraljevo	-	-	-	-	-	-	-	31	-	-	-	-	-	-	-
7	Batocina	-	-	-	-	-	-	-	32	16	-	-	-	-	-	-
8	Kragujevac	-	-	-	-	-	-	-	-	-	45	-	-	-	-	-
9	Velika Plana	-	-	-	-	-	-	-	-	-	-	18	-	-	-	-
10	Mladenovac	-	-	-	-	-	-	-	-	-	-	-	12	-	-	-
11	Smederevo	-	-	-	-	-	-	-	-	-	-	-	12	11	-	-
12	Mali Pozarevac	-	-	-	-	-	-	-	-	-	-	-	-	-	-	24
13	Kovin	-	-	-	-	-	-	-	-	-	-	-	-	-	11	-
14	Pancevo	-	-	-	-	-	-	-	-	-	-	-	-	-	-	11
15	Beograd	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Max Flow																	
Source	1	1	2	3	4	5	6	7	7	8	9	10	11	11	12	13	14
Destination	2	3	4	5	6	7	8	8	9	10	11	12	12	13	15	14	15
Capacity	11	11	5	12	5	14	31	32	16	45	18	12	21	11	24	11	11

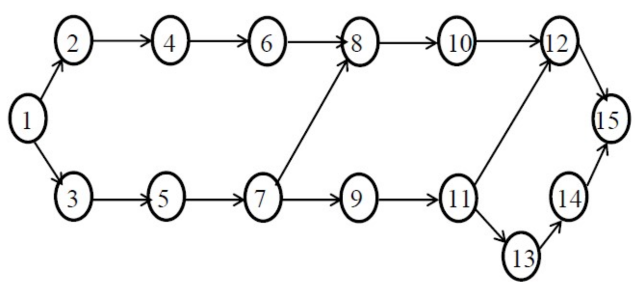

Figure3. Incidence graph

This matrix is always symmetric. That is, M^T is always equal to M for an incidence matrix. This is because whenever there is an edge from vertex i to vertex j, there is also an edge (the same one!) going in the other direction.

So far we have just talked about simple graphs. We can direct the edges of a simple graph to get a directed graph, or digraph for short. Usually we refer to directed edges as arcs. Figure 5 shows a directed version of the graph from Figure 3. The incidence matrices of digraphs are defined essentially the same as the incidence matrices of graphs, except that now the orientation of the edges matters, so we have notice that this matrix is not symmetric, since now there can be an arc from vertex i to vertex j without there being an arc in the other direction, from j to i.

A. Creating a scattered SPARSE matrix

The sparse function generates matrices in the MATLAB sparse storage organization. S = sparse(A) converts a full matrix to sparse form by squeezing out any zero elements. If S is already sparse, sparse(S) returns S. S = sparse(i,j,s,m,n) uses vectors i, j, and s to generate an m-by-n sparse matrix such that S(i(k),j(k)) = s(k).

http://dx.doi.org/10.21172/ijiet.163.02

With the following script in MATLAB, we form a sparse matrix:

cm = sparse([1 1 2 3 4 5 6 7 7 8 9 10 11 11 12 13 14],[2 3 4 5 6 7 8 8 9 10 11 12 12 13 15 14 15],[11 12 5 12 5 14 31 32 16 45 18 12 21 11 24 11 11],15,15)

Then use the following command to draw a graph, Fig. 4:

h = view(biograph(cm,[],'ShowWeights','on'))

System respond:

Biograph object with 15 nodes and 14 edges.

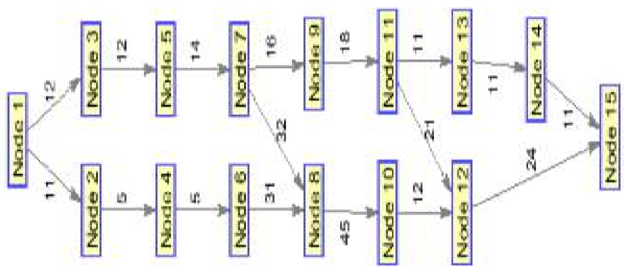

Fig. 4. Incidence graph

With parameter values:

M = The Maximum Flow
F = Flow on each link
K = Minimum cut (result displayed in matrix)
cm = Refers to the collection of data we inputted early, a N x N matrix
1 = The Source node
15 = The Destination node
We can apply the Maximum Flow / Minimum Cut algorithm by using the command:

[M,F,K] = graphmaxflow(cm,1,15)

System respond:
M = 17 (The Maximum Flow)
F =(Flow on each link)

And when we use the following command, the results for Max flow, flow for each link and Minimum Cut will be displayed. For MaxFlow, Fig. 5:

view(biograph(F,[],'ShowWeights','on'))

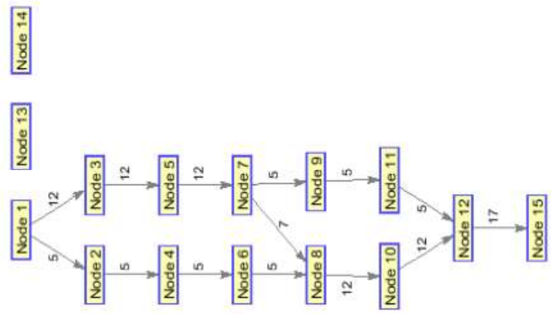

Fig.5. Viewing the Maximum Flow Network Diagram

For MinCut, Fig. 6 and Shortest path graph Fig. 7

set(h.Nodes(K(1,:)),'Color',[1 0 0]) //View Minimum Cut

NOTE: The basic network graph diagram must be open to display the MinCut **tree:**

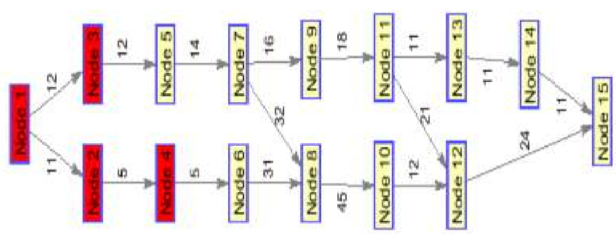

Fig. 6. Graph Minimum Cut

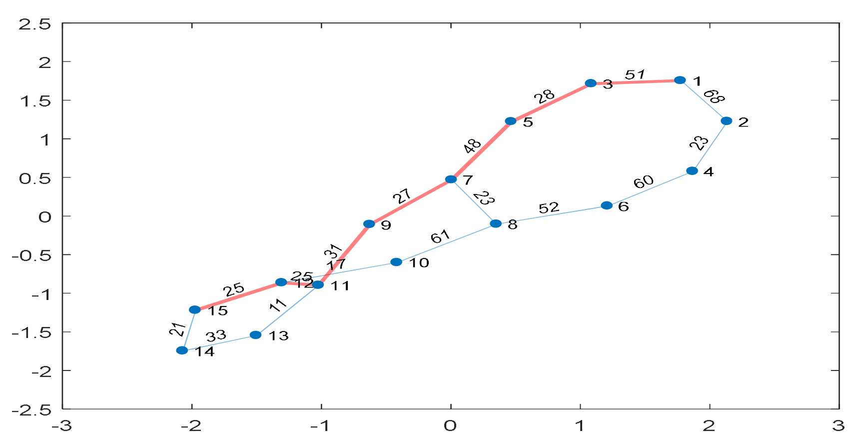

Fig.7. Shortest path graph

V. Conclusion

In this paper firstly theoretical model has been developed using graph theory to reduce traffic scenario as a capacitated network and later this concept defines nodes, edges and network flow in terms of roads, traffic Junction and traffic flow.It has been tested practically by implementing a MATLAB based simulation. The particular value of this approach is that different parameters can be applied in the simulation environment that can clearly detect changes in the newly created traffic flow. Using the Ford-Fulkerson algorithm, we can calculate the optimum value of these parameters and experimentally confirm that these optimum values result in maximum traffic and congestion during the congestion.In the simulation environment various different parameters can be applied which impact is clearly observed in traffic flow. Thus using Ford-Fulkerson's algorithm we can compute optimal value for these parameters and can experimentally confirm that these optimal values results in maximum traffic flow during congested traffic. Using this algorithm various traffic scenario such as traffic light timings and traffic speed is observed and computed the optional solution of maximize traffic flow in congested traffic as well as reduce traffic congestion. With these outputs, traffic planar can make decisions on the bottlenecks, and traffic drivers can avoid this respective traffic congestion routes, and reach their respective selected destinations in a shorter time.

REFERENCES

[1] M. Wang, H. Shan, R. Lu, R. Zhang, X. Shen, and F. Bai.: Real-time path planning based on hybrid-vanet-enhanced transportation system. Vehicular Technology, IEEE Transactions on, May 2015.
[2] G. Maia, C. Rezende, L. A. Villas, A. Boukerche, A. C. Viana, A. L. Aquino, and A. A. Loureiro: Traffic aware video dissemination over vehicular ad hoc networks. In Proceedings of the 16th ACM International Conference on Modeling, Analysis, Simulation of Wireless and Mobile Systems, MSWiM '13, pages 419–426, New York, NY, USA, 2013. ACM.
[3] R. Bauza and J. Gozalvez: Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications. Journal of Network and Computer Applications, 36(5):1295 – 1307, 2013.
[4] C. Brennad, A. Souza, G. Maia, A. Boukerche, A. Loureiro, and L. Villas: An intelligent transportation system for detection and control of congested roads in urban centers. In Computers and Communications (ISCC), 2015 IEEE Symposium on, July 2015.
[5] R. Doolan and G.-M. Muntean: Vanet-enabled eco-friendly road characteristics-aware routing for vehicular traffic. In Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th, pages 1–5, June 2013.
[6] J. Pan, M. Khan, I. SanduPopa, K. Zeitouni, and C. Borcea: Proactive vehicle re-routing strategies for congestion avoidance. In Distributed Computing in Sensor Systems (DCOSS), 2012 IEEE 8th International Conference on, pages 265–272, May 2012.
[7] A. M. d. Souza, A. Boukerche, G. Maia, R. I. Meneguette, A. A. Loureiro, and L. A. Villas: Decreasing greenhouse emissions through an intelligent traffic information system based on inter-vehicle communication. In Proceedings of the 12th ACM International Symposium on Mobility Management and Wireless Access, MobiWac '14, pages 91–98, New York, NY, USA, 2014. ACM.
[8] T. Zhong, B. Xu, and O. Wolfson: Disseminating real-time traffic information in vehicular ad-hoc networks. In Intelligent Vehicles Symposium, 2008 IEEE, pages 1056–1061, June 2008.
[9] Allan Mariano de Souza, Rodolfo Meneguette, Leonardo Castro Botega, Leandro A. Villas, SCORPION: A Solution using Cooperative Rerouting to Prevent Congestion and Improve traffic Condition, 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, ttps://www.researchgate.net/publication/292983611
[10] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein: Introduction to Algorithms, By. (Chapter 26 Flow Networks).
[11] http://en.wikipedia.org/wiki/Ford%E2%80%93Fulkerson_algorithm.
[12] MirceaParpalea: Min-Max Algorithm For The Parametric Flow Problem Bulletin of the Transilvania University of Brasov, Vol 3(52) , 2010.
[13] Steven H. Low, David E. Lapsley: Optimization flow control I: basic algorithm and convergence, IEEE/ACM Pansactions on Networking, vol. 7, no. 6, pp. 861-874, December 1999.
[14] Dorit S. Hochbaum: The Pseudoflow algorithm: A New Algorithm for the Maximum Flow Problem, Operations research, 2008
[15] D. Spasov, S. Ristov, M. Gusev: Enhancing and simulating the RIP routing protocol in cloud, Telecommunications Forum Telfor (TELFOR) 2014 22nd, pp. 372-375, Nov. 2014.
[16] A. Dwivedi, Xinghuo Yu: "A Maximum-Flow-Based Complex Network Approach for Power System Vulnerability Analysis", Industrial Informatics IEEE Transactions on, vol. 9, no. 1, pp. 81, Feb. 2013.
[17] Robert Wójcik, Jerzy Domzał, and ZbigniewDulinski: Flow-Aware Multi-Topology Adaptive Routing, Article in IEEE Communications Letters · September 2014, DOI: 10.1109/LCOMM.2014.2334314, https://www. researchgate.net /publication/265471062
[18] Marin Gunja: Maksimalniprotokkrozmrezu - Ford-Fulkerson, Edmonds-Karp, PMF-MO, Seminar izkolegijaOblikovanjeianalizaalgoritama, 22.1.2019.
[19] S.Aravinthkumar, Chandrashekar T, R.Viswanathan: An Application Designed To Obtain Maximum Flow during Homogenous Congested Traffic, International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 5, ISSN (Online) 2394-2320, Issue 4, April 2018
[20] Borivoje Milosevic, Slobodan Obradovic, Zoran Mijovic: MATLAB SOLUTIONS FOR MAXFLOW/MINCUT AND FORD FULKERSTON ALGORITHM, UNITECH 2011, INTERNATIONAL SCIENTIFIC CONFERENCE 18 – 19 November 2011, GABROVO, Bulgaria.
[21] Alex Chumbley, Karleigh Moore, Tim Hor: Ford-Fulkerson Algorithm, https://brilliant.org/wiki/ford-fulkerson-algorithm/
[22] Noraini Abdullah, Ting Kien Hua: USING FORD-FULKERSON ALGORITHM AND MAX FLOW-MIN CUT THEOREM TO MINIMIZE TRAFFIC CONGESTION IN KOTA KINABALU, SABAH, Journal of Information System and Technology Management (JISTM), 2(4), 18-34. Journal homepage: www.jistm.com.