

MINIMIZE TRAFFIC CONGESTION: AN APPLICATION OF MAXIMUM FLOW IN DYNAMIC NETWORKS

K.K. KANODIYA AND MOHD. RIZWANULLAH

Abstract

An important characteristic of a network is its capacity to carry flow. What, given capacities on the arcs, is the maximum flow that can be sent between any two nodes? The dynamic version of the maximum flow problem on networks that generalizes the well-known static one. This basic combinatorial optimization problem has a large implementation for many practical problems. Traffic congestion is a consequence of the nature of supply and demand: capacity is time consuming and costly to build and is fixed for long time periods, demand fluctuates over time, and transport services cannot be stored to smooth imbalances between capacity and demand. In this paper, I tried to solve the traffic congestion problem i.e. Maximum flow of goods in a dynamic network with the help of a Lingo Model. The same can be generalized for the large product if the software supports the systems.

Additional Key Words and Phrases: Dynamic networks, network flow, dynamic flows, flows over time, maximum flows, LINGO, Traffic congestion.

Section - I

1. INTRODUCTION

An important characteristic of a network is its capacity to carry flow. What, given capacities on the arcs, is the maximum flow that can be sent between any two nodes? The resolution of this question determines the "best" use of arc capacities and establishes a reference point against which to compare other ways of using the network. Moreover, the solution of the maximum flow problem with capacity data chosen judiciously establishes other performance measures for a network. For example, what is the minimum number of nodes whose removal from the network destroys all paths joining a particular pair of nodes? Or what is the maximum number of node disjoint paths that join this pair of nodes? These and similar reliability measures indicate the robustness of the network to failure of its components.

In this section, we discuss an algorithm for computing the maximum flow between the nodes in a network. We begin by introducing a basic labeling algorithm for solving the maximum flow problem [Maria A. Fonoberova, Dmitrii D. Lozovanu, 2004]. The validity of these algorithms rests upon the celebrated *max-flow min-cut theorem* of network flows. This remarkable theorem has a number of surprising implications in

machine and vehicle scheduling, communication systems planning and several other application domains. We then consider improved versions of the basic labeling algorithm with better theoretical performance guarantees. In particular, we describe *pre-flow push* algorithms that have recently emerged as the most powerful techniques for solving the maximum flow problem, both theoretically and computationally.

We consider a capacitated network $G = (N, A)$ with a *non-negative integer* capacity u_{ij} for any arc $(i, j) \in A$. The source s and sink t are two distinguished nodes of the network. We assume that for every arc (i, j) in A , (j, i) is also in A . There is no loss of generality in making this assumption since we allow zero capacity arcs. We also assume without any loss of generality that all arc capacities are finite (since we can set the capacity of any un-capacitated arc equal to the sum of the capacities of all capacitated arcs).

Let $U = \max \{u_{ij}; (i, j) \in A\}$. In the maximum flow problem, we wish to find the maximum flow from the source node s to the sink node t that satisfies the arc capacities. Formally, the problem is to

Maximize v
 Subject to :

$$\sum_{\{j:(i,j) \in A\}} x_{ij} - \sum_{\{j:(j,i) \in A\}} x_{ji} = \begin{cases} v, & \text{if } i = s, \\ 0, & \text{if } i \neq s, t, \text{ for all } i \in N, \\ -v, & \text{if } i = t, \end{cases}$$

$$0 \leq x_{ij} \leq u_{ij}, \text{ for each } (i, j) \in A.$$

It is possible to relax the integrality assumption on arc capacities for some algorithms, though this assumption is necessary for others. Algorithms whose complexity bounds involve U assume integrality of data. Note, however, that rational arc capacities can always be transformed to integer arc capacities by appropriately scaling the data. Thus, the integrality assumption is not a restrictive assumption in practice.

The concept of *residual network* is crucial to the algorithms to be considered. Given a flow x , the *residual capacity*, r_{ij} , of any arc $(i, j) \in A$ represents the maximum additional flow that can be sent from node i to node j using the arcs (i, j) and (j, i) . The residual capacity has two components: (i) $u_{ij} - x_{ij}$, the unused capacity of arc (i, j) , and (ii) the current flow x_{ji} on arc (j, i) which can be cancelled to increase flow to node j . Consequently, $r_{ij} = u_{ij} - x_{ij} + x_{ji}$. The network consisting of the arcs with positive residual

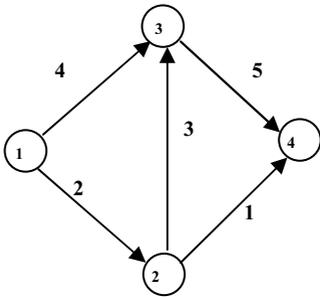
AN APPLICATION OF MAXIMUM FLOW IN DYNAMIC NETWORKS

capacities is called the *residual network* (with respect to the flow x). We represent the residual network by $G(x)$.

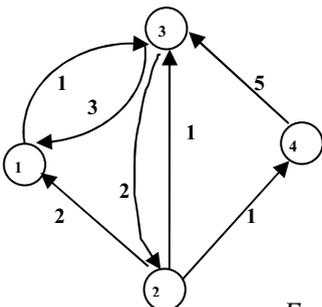
Labeling Algorithm and the Max-Flow Min-Cut Theorem

One of the simplest and most intuitive algorithms for solving the maximum flow problem is the *augmenting path algorithm* due to Ford and Fulkerson [1962]. The algorithm proceeds by identifying directed paths from the source to the sink in the residual network and sending flows on these paths, until the residual network contains no such path.

The labeling algorithm performs a search of the residual network to find a directed path from s to t . It does so by *fanning out* from the source node s to find a directed tree containing nodes that are reachable from the source along a directed path in the residual network. At any step, we refer to the nodes in the tree as *labeled* and those not in the tree as *unlabeled*. The algorithm selects a labeled node and scans its arc adjacency list (in the residual network) to label more unlabeled nodes. Eventually, the sink becomes labeled and the algorithm sends the maximum possible flow on the path from s to t . It then erases the labels and repeats this process. The algorithm terminates when it has scanned all labeled nodes and the sink remains unlabeled.



a. Network with arc.
Node 1 is the source and node 4 is the sink. (Arcs not shown have zero capacities)



b. The residual network with residual arc capacities.

Examples of Residual Networks

(Max-Flow Min-Cut Theorem) *The maximum value of flow from s to t equals the minimum capacity of all s-t cuts.*

Proof. Consider the solution obtained at the end of the labeling algorithm. Let x denote the arc flow vector and v denote the flow value. Let S be the set of labeled nodes in the residual network and $\bar{S} = N - S$. Clearly $s \in S$ and $t \in \bar{S}$. Adding the flow conservation equations for nodes in S , we can obtain

$$v = \sum_{i \in S} \sum_{j \in \bar{S}} x_{ij} - \sum_{i \in \bar{S}} \sum_{j \in S} x_{ij} = F_x(S, \bar{S}) \quad (i)$$

Note that nodes in \bar{S} cannot be labeled from nodes in S , hence $r_{ij} = 0$ for each forward arc (i, j) in the cut (S, \bar{S}) . Since $r_{ij} = u_{ij} - x_{ij} + x_{ji}$, the capacity constraints imply that $x_{ij} = u_{ij}$ and $x_{ji} = 0$. Hence $x_{ij} = u_{ij}$ for each forward arc in the cut (S, \bar{S}) and $x_{ij} = 0$ for each backward arc in the cut. Making these substitutions in (i) yields

$$v = F_x(S, \bar{S}) = \sum_{i \in S} \sum_{j \in \bar{S}} u_{ij} = C(S, \bar{S}) \quad (ii)$$

But we have observed earlier that v is a lower bound on the capacity of any s-t cut. Consequently, v is a maximum flow and the cut (S, \bar{S}) is the minimum capacity s-t cut. We thus have established the theorem and, simultaneously, proved the correctness of the labeling algorithm. There are many drawbacks of this theorem. Does it terminate finitely? second drawback of the labeling algorithm is its *forgetfulness*".

The Maximum Flow in Dynamic Networks:

The dynamic version of the maximum flow problem on networks that generalizes the well-known static one [Ford, L., Fulkerson, D, Ahuja, R., Magnati, T., Orlin, J, 1963]. This basic combinatorial optimization problem has a large implementation for many practical problems. Its solution is needed in order to solve other more complex problems, like the minimum cost circulation problem and the parametric maximum flow problem. Our dynamic model is based on the classical maximum flow problem on static networks and some generalization form. Despite being closer to reality, dynamic flow models have not been investigated in such a detailed form as classical flow models because of the complexity of the dynamic network flow models in comparison

AN APPLICATION OF MAXIMUM FLOW IN DYNAMIC NETWORKS

with the static ones. Dynamic flows [Minieka, E, 1974] are widely used to model network structured, decision-making problems over time: problems in electronic communication, production and distribution, economic planning, cash flow, job scheduling, and transportation. In the considered dynamic models the flow passes an arc with time and it can be delayed at nodes. The flow values on arcs and the network parameters in this problem can change with time. While very efficient solution methods exist for static flow problems, dynamic flow problems have proved to be more difficult to solve. The object of the maximum flow problem is to send a maximum amount of flow within a given time from supply nodes to demand nodes in such a way that link capacities are not exceeded. This problem has been studied extensively in the context of static networks [Kumar, S., Gupta, P, 2003].

Section - II

Minimize Traffic Congestion:

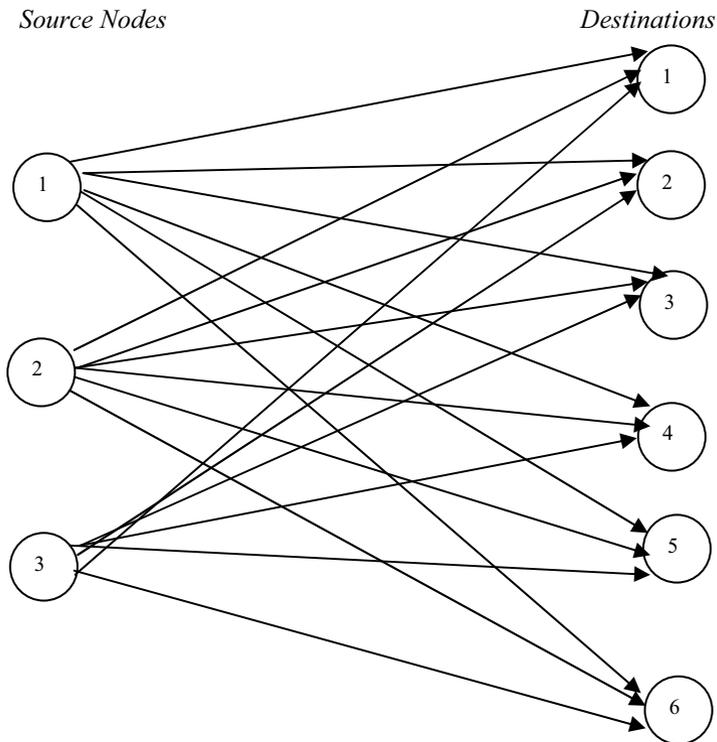
Traffic congestion is a consequence of the nature of supply and demand [Lindsey, R. and E.T. Verhoef (2000)]: capacity is time consuming and costly to build and is fixed for long time periods, demand fluctuates over time, and transport services cannot be stored to smooth imbalances between capacity and demand. Various policies to curb traffic congestion have been adopted or proposed over the years through network design. The traditional response is to expand capacity by building new network paths or upgrading existing ones. A second method is to reduce demand by discouraging peak-period travel, limiting access to congested areas by using permit systems and inventory restrictions, and so on. A third approach is to improve the efficiency of the flow of network system, so that the same demand can be accommodated at a lower cost.

Problem Formulation:

In this paper, we study the maximum flow problem in dynamic networks [Aronson, J, 1989]. We assume that capacities and flow rate depends on time. We propose a LINGO based Model for finding the maximum dynamic flow, which is based on reducing the dynamic problem to the classical maximum flow problem on a time-expanded network. Lingo allocated the maximum amount of goods to the different destination even the capacities are limited and varies according to the time and capacity of carrying the product. The complexity of this problem depends on the complexity of the supply and demand at the nodes used for the maximum static flow problem.

In a University campus, there are 6 residential hostels for the students which are situated in a distant area and there are 3 Gas Ware houses with limited capacity to supply. The hall administration providing two times meal and breakfast to the students in their respective dining hall. A dining hall of a residential hall has a kitchen with good cooking facilities including bio-gas stoves. The problem is what is the maximum unit of cylinder should be provided so that the total transportation time/cost is minimum. The demand at each destination is fulfill to maximum extent and the supply will get exhausted.

The above problem is converted into a network problem of minimum traffic congestion with maximum dynamic flow. In this model, the six hostels become the nodes of a network (Destination nodes) and three of the nodes are source nodes. Each of the source nodes can ship to any of the destination nodes. The problem is the more product you send down an arc, the longer it takes for it to arrive. This might be the case if the underlying network were a railroad, for instance.



We assume shipping times obey the following relationship:

AN APPLICATION OF MAXIMUM FLOW IN DYNAMIC NETWORKS

$$\text{Time} = \text{Rate} / (1 - \text{Flow} / \text{Limit})$$

where,

Time: time to ship one unit of product down a route,

Rate: time required to transport one unit down a route if there is no congestion along the route,

Flow: amount of product shipped down a route, and

Limit: is the maximum limit of product flow down a route.

Based on this relationship, we see shipping times go to infinity as the capacity of an arc is approached. The goal of the model is to determine how much product to ship from each source to each destination, so as to minimize total shipping times.

Source node with supply capacity (in units)

AMUGC		SS		HH	
1000		600		400	
Destination nodes with Demand (in Units)					
MH	SN	SUL	AI	MM	NT
300	200	500	400	350	250

Rate and Limit Matrix

Source Nodes	Destination with Rate of Flow and Limit (Capacity) in Units					
	MH	SN	SUL	AI	MM	NT
AMUGC	23	22	20	21	23	19
	100	100	150	300	250	400
SS Hall	25	27	30	28	25	22
	250	350	300	400	200	100
HH Hall	20	24	27	26	20	17
	400	500	600	650	700	850

Lingo Programme:

! Minimum Traffic congestion transportation problem for Maximum flow.

K.K. KANODIYA, MOHD. RIZWANULLAH

Cost/unit increases to infinity as traffic on link approaches its link capacity.;

```
SETS:
  ORIG/ AMUGC SS HH/: SUPPLY;
  DEST / MH SN SUL AI MM NT/: DEMAND;
  OXD( ORIG, DEST): RATE, LIMIT, TRAF;
ENDSETS
DATA:
  SUPPLY = 1000 600 400 ;

DEMAND = 300 200 500 400 350 250;
  RATE =   23 22 20 21 23 19
          25 27 30 28 25 22
          20 24 27 26 20 17;
  LIMIT = 100 100 150 300 250 400
          250 350 300 400 200 100
          400 500 600 650 700 850;

ENDDATA
[TOTCOST] MIN =
  @SUM( OXD: RATE * TRAF/( 1 - TRAF/ LIMIT));
@FOR( ORIG( I):
  @SUM( OXD( I, J): TRAF( I, J)) = SUPPLY( I));
@FOR( DEST( J):
  @SUM( OXD( I, J): TRAF( I, J)) = DEMAND( J));
@FOR( OXD: @BND( 0, TRAF, LIMIT););
END
```

Lingo Solution of the Model:

In a LINGO solution report, we'll find a reduced cost figure for each variable. There are two valid, equivalent interpretations of a reduced cost. First, we may interpret a variable's reduced cost as the amount that the objective coefficient of the variable would have to improve before it would become profitable to give the variable in question a positive value in the optimal solution. A variable in the optimal solution, automatically has a reduced cost of zero. Second, the reduced cost of a variable may be interpreted as the amount of penalty we would have to pay to introduce one unit of that variable into the solution.

The Slack or Surplus column in a LINGO solution report tells us how close we are to satisfying a constraint as equality. If a constraint is exactly satisfied as equality, the slack or surplus value will be zero. If a constraint is violated, as in an infeasible solution, the slack or surplus value will be negative. Knowing this can help us to find the violated constraints in an infeasible model—a model for which there doesn't exist a set of variable

AN APPLICATION OF MAXIMUM FLOW IN DYNAMIC NETWORKS

values that simultaneously satisfies all constraints. Non-binding constraints, constraints with a slack or surplus value greater than zero, will have positive, nonzero values in this column.

Result:

Local optimal solution found at iteration: 82
 Objective value: 152640.1

Variable	Value	Reduced Cost
SUPPLY(AMUGC)	1000.000	0.000000
SUPPLY(SS)	600.0000	0.000000
SUPPLY(HH)	400.0000	0.000000
DEMAND(MH)	300.0000	0.000000
DEMAND(SN)	200.0000	0.000000
DEMAND(SUL)	500.0000	0.000000
DEMAND(AI)	400.0000	0.000000
DEMAND(MM)	350.0000	0.000000
DEMAND(NT)	250.0000	0.000000
RATE(AMUGC, MH)	23.00000	0.000000
RATE(AMUGC, SN)	22.00000	0.000000
RATE(AMUGC, SUL)	20.00000	0.000000
RATE(AMUGC, AI)	21.00000	0.000000
RATE(AMUGC, MM)	23.00000	0.000000
RATE(AMUGC, NT)	19.00000	0.000000
RATE(SS, MH)	25.00000	0.000000
RATE(SS, SN)	27.00000	0.000000
RATE(SS, SUL)	30.00000	0.000000
RATE(SS, AI)	28.00000	0.000000
RATE(SS, MM)	25.00000	0.000000
RATE(SS, NT)	22.00000	0.000000
RATE(HH, MH)	20.00000	0.000000
RATE(HH, SN)	24.00000	0.000000
RATE(HH, SUL)	27.00000	0.000000
RATE(HH, AI)	26.00000	0.000000
RATE(HH, MM)	20.00000	0.000000
RATE(HH, NT)	17.00000	0.000000
LIMIT(AMUGC, MH)	100.0000	0.000000
LIMIT(AMUGC, SN)	100.0000	0.000000
LIMIT(AMUGC, SUL)	150.0000	0.000000
LIMIT(AMUGC, AI)	300.0000	0.000000
LIMIT(AMUGC, MM)	250.0000	0.000000
LIMIT(AMUGC, NT)	400.0000	0.000000
LIMIT(SS, MH)	250.0000	0.000000
LIMIT(SS, SN)	350.0000	0.000000
LIMIT(SS, SUL)	300.0000	0.000000
LIMIT(SS, AI)	400.0000	0.000000
LIMIT(SS, MM)	200.0000	0.000000

K.K. KANODIYA, MOHD. RIZWANULLAH

LIMIT(SS, NT)	100.0000	0.000000
LIMIT(HH, MH)	400.0000	0.000000
LIMIT(HH, SN)	500.0000	0.000000
LIMIT(HH, SUL)	600.0000	0.000000
LIMIT(HH, AI)	650.0000	0.000000
LIMIT(HH, MM)	700.0000	0.000000
LIMIT(HH, NT)	850.0000	0.000000
TRAF(AMUGC, MH)	82.89277	-0.6721403E-06
TRAF(AMUGC, SN)	83.051	-0.1045279E-05
TRAF(AMUGC, SUL)	126.55	-0.3530884E-06
TRAF(AMUGC, AI)	250.6182	-0.3307146E-06
TRAF(AMUGC, MM)	206.8796	0.000000
TRAF(AMUGC, NT)	250.0000	0.000000
TRAF(SS, MH)	111.0414	-0.1963008E-06
TRAF(SS, SN)	116.9485	0.000000
TRAF(SS, SUL)	146.0337	0.000000
TRAF(SS, AI)	147.1293	0.000000
TRAF(SS, MM)	78.84713	0.000000
TRAF(SS, NT)	0.000000	591.8745
TRAF(HH, MH)	106.0658	0.000000
TRAF(HH, SN)	0.000000	6.983759
TRAF(HH, SUL)	227.4084	0.000000
TRAF(HH, AI)	2.252570	-0.2224461E-04
TRAF(HH, MM)	64.27324	0.5669908E-06
TRAF(HH, NT)	0.000000	630.7551

Row	Slack or Surplus	Dual Price
TOTCOST	152640.1	-1.000000
2	0.000000	-704.9856
3	0.000000	0.000000
4	0.000000	43.88060
5	0.000000	-80.91880
6	0.000000	-60.89703
7	0.000000	-113.8971
8	0.000000	-70.06176
9	0.000000	-68.12912
10	0.000000	569.8743

ACKNOWLEDGMENTS

Many thanks to my wife who supported in terms of motivation, dedication and supports.

REFERENCES

1. Ahuja, R., Magnati, T., Orlin, J., *Network Flows*. Prentice-Hall, Englewood Cliffs, 1993.
2. Aronson, J., *A survey of dynamic network flows*. Ann. Oper. Res., vol. 20, 1989, pp. 1–66. 395 M.A. Fonoberova, D.D. Lozovanu.
3. Aumann, Y. And Rabani, Y. 1998. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM J. Comput.* 27, 1, 291–301.
4. Aronson, J.E., A survey of dynamic network flows, *Annals of Operations Research* 20 (1989) 1–66.
5. Ford, L., Fulkerson, D., *Flows in Networks*. Princeton University Press, Princeton, NJ, 1962.
6. Ben-Akiva, M., A. de Palma and I. Kaysi (1991). Dynamic network models and driver information systems. *Transportation Research A*, 25A(5), 251–266.
7. Dahlgren Lab of U.S. Navy, *Personal communications* 2001.
8. Ford, L., Fulkerson, D., *Constructing maximal dynamic flows from static flows*. Operation Res., vol. 6, 1958, p. 419–433.
9. Fleisher, L., Skutella, M., *The quickest multi-commodity flow problem. Integer programming and combinatorial optimization*, Springer, Berlin, 2002, p. 36–53.
10. Fleischer, L., *Universally Maximum Flow with Piecewise-Constant Capacities. Networks*, vol. 38, no. 3, 2001, pp. 115–125.
11. G. Bretti, R. Natalini, and B. Piccoli, “Numerical algorithms for simulations of a traffic model on road networks”, *J. Comput. Appl. Math.*, 2007, 210, (1-2), pp. 71–77.
12. Goldberg, A., Tarjan, R., *A New Approach to the Maximum-Flow Problem*. Journal of the Association for Computing Machinery, vol. 35, no. 4, 1988, pp. 921–940.
13. Garg, N., Vazarani, V., And Yannakakis, M. 1996. Approximate max-flow min-(multi) cut theorems and their applications. *SIAM J. Comput.* 25, 235–251.
14. Hoppe, B. and Tardos, E., The quickest transshipment problem, *Mathematics of Operations Research* 25 (2000) 36–62.
15. Hoppe, B., Tardos, E., *The quickest transshipment problem. Mathematics of Operations Research*, vol. 25, 2000, p. 36–62.
16. Kohler, E. and Skutella, M., Flows over time with load-dependent transit times, *Proceedings of SODA '02* (2002) 174–183.
17. Kumar, S., Gupta, P., *An Incremental Algorithm for the Maximum Flow Problem*. Journal of Mathematical Modeling and Algorithms. vol. 2, 2003, pp. 1-16.
18. Leighton, F. T., Makedon, F., Plotkin, S., Stein, C., Tardos, E., And Tragoudas, S. 1992. Fast approximation algorithms for multi-commodity flow problems. *J. Comput. Syst. Sci.* 50, 228–243.
19. Lindsey, R. and E.T. Verhoef (2000). Congestion modeling. Forthcoming in: *Handbook of Transport Modeling, Vol. 1*. (D.A. Hensher and K.J. Button, eds.), Elsevier Science, Oxford.
20. Lozovanu, D., Stratila, D., *Optimal flow in dynamic networks with nonlinear cost functions on edges. Analysis and optimization of differential systems*. (Edited by V.Barbu, I.Lesiencko), ISBN 1–4020–7439–5. Klnwer Academic Publissers, 2003, p. 247–258.
21. Lozovanu, D., Stratila, D., *The minimum-cost flow problem on dynamic networks and algorithm for its solving*. Bul. Acad. S,tiint,e Repub. Mold., Mat., vol. 3, 2001, p. 38–56.

22. Laih, C-H. (1994). Queuing at a bottleneck with single- and multi-step tolls. *Transportation Research A*, 28A(3), 197-208.
23. Maria A. Fonoberova, Dmitrii D. Lozovanu, “The maximum flow in dynamic networks”, *Computer Science Journal of Moldova*, vol.12, no.3 (36), 2004.
24. Mazzoni, G., Pallottino, S., Scutella, M., *The maximum flow problem: A max-preflow approach*. European Journal of Operational Research, vol. 53, 1991, pp. 257–278.
25. Minieka, E., Dynamic network flows with arc changes, *Networks* 4 (1974) 255–265.
26. Powell, W.B., Jaillet, P. and Odoni, A., Stochastic and dynamic networks and routing, In: *Network*.
27. Routing, Vol. 8 (1995) of *Handbooks in Operations Research and Management Science*, Chapter 3 141–295.
28. Shahrokhi, F. And Matula, D. W. 1990. The maximum concurrent flow problem. *J. ACM* 37, 318 –334.
29. Wikipedia, “Traffic Congestion”, Available: http://en.wikipedia.org/wiki/Traffic_congestion. [Accessed March 28, 2008]

Dr. K.K. Kaanodiya,
Head of Department of Mathematics,
BSA (P.G.) College,
Mathura-281004,
(U.P.), India.

Mohd. Rizwanullah,
Research Scholar,
Department of Mathematics,
BSA (P.G.) College,
Mathura- 281004, (U.P.), India.