International
Journal
Of Advanced Research in Engineering & Management (IJAREM)

Generalized Trapezoidal Fuzzy Numbers with Rank and Divergence Provide the Optimal Outcome of Fuzzy Maximal Flow Network

M. K. Alam[1*], M. K. Hasan[2], Maheswara Rao Valluri[3], Nirmal Kanti Mitra[4] and K. Ramanuja Rao[5]

[1,2] *Department of Mathematics and Statistics, Bangladesh University of Business and Technology, Dhaka, Bangladesh.*

[3,4,5] *School of Mathematical and Computing Sciences, Fiji National University, Fiji Island.*

Abstract: A flow network is a directed graph where each edge receives a flow and has a capacity.One of the applications to network-flow is labeling algorithm. Labeling techniques algorithms are used to solve wide variety of network problems, such as shortest-path problems, maximal-flow problems, general minimal-cost network-flow problems and minimal spanning tree problems. In 2010,Kumar et al., proposed an algorithm to find the fuzzy maximal flow network between source and sink with generalized trapezoidal fuzzy numbers with only rank. But sometimes ranking function fails for choosing the path of the flow. To avoid this drawback, divergence functions are to be used. This paper proposes an algorithm for getting fuzzy maximal flow network between source and sink for generalized trapezoidal fuzzy number with rank and divergence functionsand also illustrates the algorithm by solving a numerical example.

Index Terms— Normal Trapezoidal Fuzzy Numbers; Generalized Trapezoidal Fuzzy Numbers; Fuzzy Flow, Fuzzy Residue

1. Introduction

Arithmetic operations on fuzzy numbers had been developed by Moore [15] according to the extension principle based on interval arithmetic. In [19], Abbasband has introduced a new approach for ranking of trapezoidal fuzzy numbers based on the rank, mode, left and right spreads at some-levels of trapezoidal fuzzy numbers.

The maximum flow problem is one of basic problems for combinatorial optimization in weighted directed graphs. In the real life situation very useful models in a number of practical contexts including communication networks, oil pipeline systems, power systems, costs, capacities and demands are constructed by the base of maximal flow network problem. In [7], Fulkersonhad provided the maximal flow problem and solved by the simplex method for the linear programming. The maximal flow problem has been solved by Ford [11] using augmenting path algorithm. This algorithm had been used to solve the crisp maximal flow problems [9], [13], [16]. Fuzzy numbers represent the parameters of maximal flow problems . It is noted that, K. Kim and F. Roush are the first introducer on this subject [10]. Chanas [20-22] had approached this problem using minimum costs technique. An algorithm for a network with crisp stricter was presented by Chanas [20]. In [21],itis discussed that the flow is a real number and the capacities have upper and lower bounds. In [23], the integer flow had been studied and an algorithm had been proposed. Interval-valued versions of the max-flow min cut theorem and Karp-Edmonds algorithm was developed by Diamond [2]. Some times, it arises uncertain environment. The network flow problems using fuzzy numbers were investigated by Liu [25]. Generalized fuzzy versions of maximum flow problem were considered by Ji [27] with respect to arc capacity as fuzzy variables. An algorithm to find fuzzy maximal flow between source and sink is proposed by Kumar et al. [3] with the help of ranking function.

The objectives of this paper to stimulate the inclusion of trapezoidal fuzzy numbers in applied engineering and scientific problems by extending the concept of traditional algebra into fuzzy set theory, which is described by Bansal [1]. In this paper, we propose an algorithm by modifying an existing algorithm [3] to find fuzzy maximal flow between sources and sinks by representing all the parameters considered as generalized trapezoidal fuzzy numbers. To illustrate the algorithm, a numerical example is solved. If there is no uncertainty about the flow between source and sink then the proposed algorithm gives the same result as in crisp maximal flow problems. But when we face same rank, we apply divergence function for selected maximal flow path. In section 2, some basic definitions, rank and divergence function and arithmetic operations for interval and generalized trapezoidal fuzzy numbers are discussed. In section 3, we propose an algorithm for solving fuzzy maximal flow problems. In section 4, a numerical example and results are given. Finally, conclusion remarks are made in section 5.

2. Preliminaries

In this section some basic definitions, ranking function, divergence function and arithmetic operations are presented.

Definition 2.1 [14]:Fuzzy set defined on the set $\mathbb{R}$ of real numberswith membership function of the form $\mu: \mathbb{R} \to [0,1]$ is called fuzzy number if the following axioms are satisfies:

- μ must be normal fuzzy set i.e. there exist $x \in \mathbb{R}$; $\mu(x) = 1$
- $\alpha -$cut, μ^{α} must be closed interval of real number, for every $\alpha \in]0,1]$
- The support of μ must be bounded and compact, that is $\{x \in \mathbb{R};\ \mu(x) > 0\}$ is bounded and compact.

Throughout this paper, we write fuzzy number FN.

Definition 2.2 [24]: A fuzzy number $A = [[a, b, c, d]]$ is said to be a trapezoidal fuzzy numbers if its membership function is given by

$$\mu(x) = \begin{cases} 0 & ; \quad -\infty < x \le a \\ \dfrac{x-a}{b-a} & ; \quad a \le x < b \\ 1 & ; \quad b \le x \le c \\ \dfrac{x-d}{c-d} & ; \quad c < x \le d \\ 0 & ; \quad d \le x < \infty \end{cases}, \qquad \text{where } a, b, c, d \in \mathbb{R}$$

Throughout this paper, we denote trapezoidal fuzzy numberby TFN.

Definition 2.3[24]: A fuzzy number $A = [[a, b, c, d; w]]$ is said to be a generalized trapezoidal fuzzy number if its membership function is given by

$$\mu(x) = \begin{cases} 0 & ; \quad -\infty < x \le a \\ \dfrac{w(x-a)}{b-a} & ; \quad a \le x < b \\ w & ; \quad b \le x \le c \\ \dfrac{w(x-d)}{c-d} & ; \quad c < x \le d \\ 0 & ; \quad d \le x < \infty \end{cases} \qquad \text{where } a, b, c, d \in \mathbb{R} \text{ and } w \in]0, 1]$$

Throughout this paper, we denote generalized trapezoidal fuzzy number by $GTFN$

.

2.4 Arithmetic Operations

In this subsection addition and subtraction of two GFTN are defined.

Definition 2.4.1[10]: Let $A = [[a_1, b_1, c_1, d_1;\ w_1]]$ and $B = [[a_2, b_2, c_2, d_2;\ w_2]]$ be two GFTNs, then

- $A + B = [[a_1 + a_2, b_1 + b_2, c_1 + c_2, d_1 + d_2; \min\{w_1, w_2\}]]$
- $A - B = [[a_1 - d_2, b_1 - c_2, c_1 - b_2, d_1 - a_2; \min\{w_1, w_2\}]]$

2.5 Ranking function

A convenient method for comparing fuzzy numbers is to use ranking function. A ranking function $\Re: F(\mathbb{R}) \to \mathbb{R}$ maps each fuzzy number in to a real number, here $F(\mathbb{R})$ is set of all fuzzy numbers defined on $\mathbb{R}$.

Definition 2.5.1[26]**:** Let $A = [[a_1, b_1, c_1, d_1;\ w_1]]$ and $B = [[a_2, b_2, c_2, d_2;\ w_2]]$ be two $GTFNs$, then

$\Re(A) = \frac{w_1(a_1+b_1+c_1+d_1)}{4}$ and $\Re(B) = \frac{w_2(a_2+b_2+c_2+d_2)}{4}$

If $\Re(A) > R(B)$ then we say $A \succ B$

If $\Re(A) < R(B)$ then we say $A \prec B$

If $\Re(A) = \Re(B)$ then we say $A \approx B$

2.6Divergence function

When two GFTNs A and B are such that $A \approx B$ with respect to ranking and mode function, we apply divergence function for maximum flow position. A divergence function $D: F(\mathbb{R}) \to \mathbb{R}$ maps each fuzzy number into a real number, here $F(\mathbb{R})$ is set of all fuzzy numbers defined on $\mathbb{R}$.

Definition 2.6.1[4]**:** Let $A = [[a_1, b_1, c_1, d_1;\ w_1]]$ and $B = [[a_2, b_2, c_2, d_2;\ w_2]]$ be two GFTNs, then

$$D(A) = w_1(d_1 - a_1) \text{and } D(B) = w_2(d_2 - a_2).$$

3 Proposed Algorithm

The normal form of trapezoidal fuzzy numbers had been used by various papers for solving real life problems [12-15]. In this section, we propose an algorithm for solving the fuzzy maximal flow problems. In order to propose the algorithm, we first modify the maximal flow network problem by using GFTN. For network flow, to find fuzzy maximal flow between sources and sink for GFTN, we follow [8, 9, 28] . The fuzzy maximal flow algorithm is based on finding breakthrough paths with net positive flow between the source and sink nodes. Consider arc (i, j) with initial fuzzy capacities $\left(\overline{\mu c_{ij}}\ ,\ \overline{\mu c_{ji}}\right)$ and fuzzy residuals capacities$\left(\mu c_{ij}\ , \mu c_{ji}\right)$. For a node j that receives flow from node i, we consider a label$\left[\mu a_j\ , i\right]$, where μa_j is the fuzzy flow from node $i$to j. The steps of algorithm for GFTN are summarized as follow:

Step 1: For all arcs (i, j), set the residual fuzzy capacity is equal to initial fuzzy capacity, that is,$\left(\mu c_{ij}\ , \mu c_{ji}\right) = \left(\overline{\mu c_{ij}}\ ,\ \overline{\mu c_{ji}}\right)$. Let $\mu a_1 = [[\infty, \infty, \infty, \infty; 1]]$ and label the source node 1 with$\left([[\infty, \infty, \infty, \infty; 1]], _\right)$. Set $i = 1$, and go to step 2.

Step 2: Determine S_i, the set of unlabeled nodes j that can be reached directly from node i by arcs with positive residuals capacity, that is, μc_{ij} is non-negative fuzzy number for each $j \in S_i$. If$S_i = \emptyset$, then go to step 4, otherwise go to step 3.

Step 3: Determine$k \in S_i$ such that$\max_{j \in S_i}\{\Re(\mu c_{ij})\} = \Re(\mu c_{ik})$.Set $\mu a_k = \mu c_{ik}$ and label node k with $[\mu a_k, i]$. If $k = n$, the sink node has been labeled, and a breakthrough path is found, then go to step 5. Otherwise go to step 2. Again, if$\max_{j \in S_i}\{\Re(\mu c_{ij})\}$ is more than one fuzzy flow, then we apply divergence test for maximal flow according to maximal flow arc tested.

Seep 4: If$i = 1$, no breakthrough possible, then go to step 6. Otherwise, let r be the node, that is labeled immediately before current node i and remove i from the set of nodes adjacent to r. Set $i = r$ and go to step 2.

Step 5: Let $N_p = \{1, k_1, k_2, \dots \dots \dots, n\}$. Define the nodes of thep^{th}breakthrough path from source node 1 to sink node n. Then the maximal flow along the path is completed as $\mu_p = \min\{\mu_1, \mu_{k_1}, \mu_{k_2}, \dots \dots \dots, \mu_n\}$. The residual capacity of each along the breakthrough path is decreased by μ_p in the direction of the flow and increased by μ_p in the reverse direction, that is for nodes i and j on the path, the residual flow changes from the current $\left(\mu c_{ij}\ , \mu c_{ji}\right)$ to

Case 1:$\left(\mu c_{ij} - \mu_p, \mu c_{ji} + \mu_p\right)$ if the flow is from i to j or else

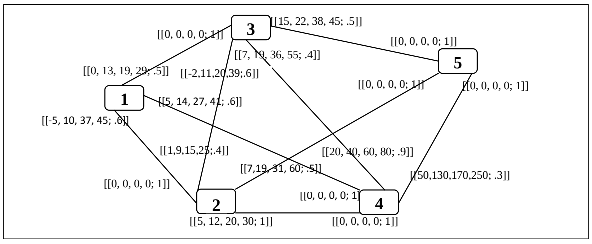
Case 2 :$\left(\mu c_{ij} + \mu_p, \mu c_{ji} - \mu_p\right)$ if the flow is from j to i.

3.1 Step 6: In the step we will determine flow and residue.

3.1.1 Given that total number of breakthrough paths are m. Then we get total flow of a network by determining: $F = \mu_1 + \mu_2 + \mu_3 + \cdots + \mu_m$, where m is the number of iterations.

4 Illustrative Example

In this section the proposed algorithm is illustrated by solving a numerical example.



Example: Consider the network shown in the figure 1. We will find out the fuzzy maximal flow between source node 1 and destination node 5.

Iteration 1 and 2 :(Residue Network**)**

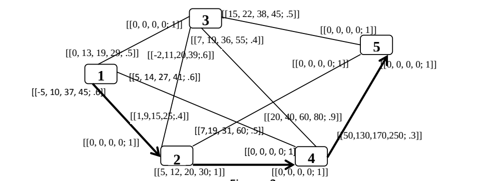


Figure: 2

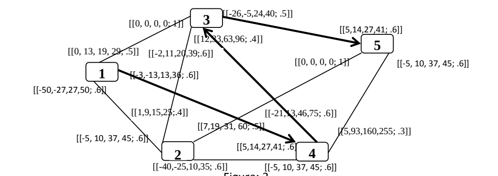


Figure: 3

Iteration 3 and 4: (Residue Network**)**

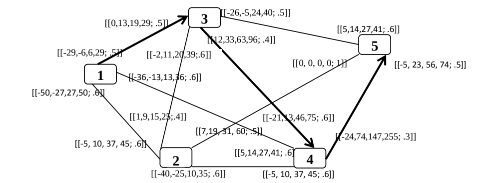


Figure: 4

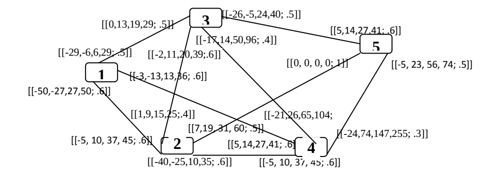


Figure:5

Flow direction after end of iteration:

Using the initial and final fuzzy residuals of arc (i, j) as $\left(\overline{\mu c}_{ij}, \overline{\mu c}_{ji}\right)$ and $\left(\mu c_{ij}, \mu c_{ji}\right)$ respectively, the fuzzy optimal flow in arc (i, j) is computed as follows:

For $(\alpha, \beta) = \left(\overline{\mu c}_{ij} - \mu c_{ij}, \overline{\mu c}_{ji} - \mu c_{ji}\right)$, if$\Re(\alpha) > 0$, then fuzzy optimal flow from i to j is α. Otherwise, if $\Re(\beta) > 0$, then the fuzzy optimal flow from j to $i$is β.

Remark 1: Set the initial fuzzy residual $\left(\mu c_{ij}, \mu c_{ji}\right)$ equal to the initial fuzzy capacity $\left(\overline{\mu c}_{ij}, \overline{\mu c}_{ji}\right)$. Input of all flow from the given network according to Mathematica:

Remark 2: First four entries of each vector represents trapezoidal fuzzy number and fifth entry is value of w for GFTN. We have used "**,**" in replace of ";" for calculation in Mathematica. Also we have used "**{}**" in replace of "[[ ]]" for fuzzy number to calculation in Mathematica. Any bold Mathematica texts are input and other texts are output.

Remark 3: Mathematicaprogram scripts are not shown in the text(shown in appendix).

Initiallythe mathematicaprogromnagpath[F_] is used forupdating to arc, direction of path selection of augmented path according to rank and divergent, maximal flow along the path which is breakthrough path and update residual flow to each path.Secondly the mathematicaprogromfdir[OF_,NF] is used for calculating total flow and selecting flow direction.

Remark 4: Outputs are as follows**:**

Initial flow [[-5, 10, 37, 45; .6]] from node 1 to node 2, [[0, 13, 19, 29; .5]] from node 1 to node 3 [[5, 14, 27, 41; .6]] from node 1 to node 4 and other nodes are not directly connected with node 1 which is mentioned by [[0,0,0,0,0]] .Finally it is found the following:

```
{{0, 0, 0, 0, 0}, {-50, -27, 27, 50, 0.6},
 {-29, -6, 6, 29, 0.5}, {-36, -13, 13, 36, 0.6}, {0, 0, 0, 0, 0}}
```

Sum of any flow are zero for the source node 1.i,e the further improvement is not possible.

5 Results and Discussion:

In this section, we will shortly explanation our result for the above numerical example. We have found the fuzzy maximal (optimal) flow $F = [[0, 37, 83, 115; 0.5]]$. From this result we can take the decision that the amount of flow between source and sink is 0 and 105 unit. We also can take the decision about the statement that the maximal flow will be 37 to 83 unit is 50%. The decision making for the reaming amount of flow can be obtained by

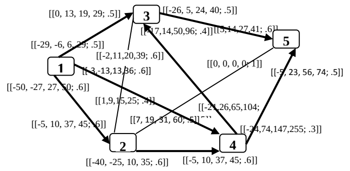


Figure: 5

$$\mu F(x) = \begin{cases} 0 & ; \quad -\infty < x \leq 0 \\ \dfrac{x}{74} & ; \quad 0 \leq x < 37 \\ 0.5 & ; \quad 37 \leq x \leq 83 \\ \dfrac{115 - x}{64} & ; \quad 83 < x \leq 115 \\ 0 & : \quad 115 \leq x < \infty \end{cases}$$

```
sum of total flow
 {{-5, 10, 37, 45, 0.6}, {5, 14, 27, 41, 0.6}, {0, 13, 19, 29, 0.5}}
  is = {0, 37, 83, 115, 0.5}
```

5. Conclusion

In this paper, we have proposed an algorithm for solving the fuzzy maximal flow problems occurring in real life situation and have shown that the flows are represented by using generalized trapezoidal fuzzy numbers. In [5], Kumar and Kaur have solved fuzzy maximal flow problems using generalized trapezoidal fuzzy numbers but they have applied only ranking function for maximal flow path. In the our proposed algorithm of the paper, we have used ranking function and divergence function when ranking function fails for choosing the path of the flow. In future, we can solve the other network problems such as one way traffic flow, gas flow and liquid flow etc. by extending the proposed algorithm.

References

[1] A. Bansal, "Trapezoidal Fuzzy Numbers (a,b,c,d): Arithmetic Behavior", *International Journal of Physical and Mathematical Science*, (2011), pp. 39-44.
[2] A. Diamond, "A Fuzzy Max-Flow Min-Cut Theorem", *Fuzzy Sets and Systems*, 119: (2001), pp. 139-148.
[3] A. Kumar, N. Bhatia and M. Kaur, "A new approach for solving fuzzy maximal flow problems", *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Heidelberg, 5908 (2009), pp. 278-286.
[4] A. Kumar, P. Singh, A. Kaur and P. Kaur, "Ranking of generalized Trapezoidal Fuzzy Numbers Based on Rank, Mode, Fivergence and Spread", *Turkish Journal of Fuzzy Systems*, Vol. 1, No. 2 (2010), pp. 141-152.
[5] A. Kumar and M. Kaur, "An Algorithm for Solving Fuzzy Maximal Flow Problems Using Generalized Trapezoidal Fuzzy Numbers", *International Journal of Applied Science and Engineering*, 8, 2 (2010), pp. 109-118.
[6] D. Dubois and H. Prade, "Fuzzy real algebra Some results", Fuzzy Sets Syatems, 2 (1979), pp. 327-349
[7] D. R. Fulkerson and G. B. Dantzig, "Computation of Maximum Flow in Network", *Naval Research Logistics Quarterly*, vol. 2, (1955), pp. 277-283.
[8] H. A. Taha, *Operational Research: An Introduction*, Prentice-Hall, New Jersey, (2003).
[9] H. A. Taha, *Operation Research: An Introduction*; Eighth Edition, New Dhelhi-110 001, (2006).
[10] K. Kim and F. Roush, "Fuzzy Flows on Network", *Fuzzy Sets and Systems*, 8: (1982), pp 35-38
[11] L. R. Ford and D. R. Fulkerson, "Maximal Flow Through a network", *Canadian Journal of Mathematics*, vol. 8, (1956), pp. 399-401.
[12] L. A. Zadeh, "Fuzzy sets", *Information and Control*, 8, (1965), pp. 338–353.
[13] M. S. Bazarra, J. J. Jarvis and H. D. Sherali, *Linear Programming and Network Flows*, 2nd Edition, Wiley, New York, (1990).
[14] P. A. Thaker, D. S. Shelar and S. P. Thaker, "Solving Fuzzy Linear Programming Problem", *Proceedings of the World Corld Congress on Engineering*, London, U.K., Vol II, (2009) pp. 978-988.
[15] R. E. Moore, *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, 1979.
[16] R. K. Ahuja, T. L. Magnanti and J. B.Orlin, *Network Flows, Theory and Algorithm*, Prentice Hall, New Jersey (1993).
[17] R. Yager, "On Solving Fuzzy Mathematical Relationships", *Information Control*, 41 (1979) 29-55.

[18] S. Wolfram, *TheMathematica Book*: 5th Edition, Cambridge (2003).
[19] S. Abbasband, T. Hajjari, “A new approach for ranking of trapezoidal fuzzy numbers”, *Computers and Mathematics with Applications*, 57 (2009), pp. 413-419.
[20] S. Chanas and W. Kolodziejczyk, “Maximum Flow in a Network with fuzzy arc capacities”, *Fuzzy Sets and Systems*, 8: (1982), pp. 165-173.
[21] S. Chanas and W. Kolodziejczyk, “Real-valued Flow in a Network with fuzzy arc capacities”, *Fuzzy Sets and Systems*, 13: (1984), pp. 139-151.
[22] S. Chanas and W. Kolodziejczyk, “Integer Flow in Network with Fuzzy Capacity Constrains”, *Networks*, 16: (1986), pp. 17-31.
[23] S. H. Chen “Operations of Fuzzy Numbers with function principal”, *Tamkang Journal of Management Sciences*, 6 (1985), pp. 13-25.
[24] S. M. Chen and J. H. Chen, “Fuzzy risk analysis based on ranking generalized fuzzy numbers with different heights and different spreads”, *Expert Systems with Applications*, vol. 36, (2009), pp. 6833-6842.
[25] S. T. Liu and C. Kao, “Network Flow Problems with Fuzzy Arc Lengths”, *IEEE Transactions on Systems, Man and Cybernetics*, 34: (2004), pp. 765-769.
[26] T. S. Liou, and M. J. Wang, “Ranking fuzzy numbers with integral value”, *Fuzzy Sets and Systems*, 50(1): (1992), pp. 247-255.
[27] X. Ji, L. Yang and Z. Shao, “Chance Constrained Maximum Flow Problem with Arc Capacities”, *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Heidelberg, 4114: (2006), pp. 11-19.
[28] M. K. Alam, M. K. Hasan, “Solution of Fuzzy Maximal Flow Network Problem Based on Generalized Trapezoidal Fuzzy Numbers with Rank and Mode”, International Journal of Engineering Research and Development, Volume 9, Issue 7 (January 2014), PP. 40-49.

Appendix:

Mathematica program for breakthrough path and total flow calculation according to *rank* and *divergence;* also *residual capacity* calculation:

```
nagpath[F_] := Module[{f = F, n, dv, vd, pr, ddv}, n = Dimensions[f];
  tt = {}; uu = {}; dd = {}; path = {};
  For[i = 1, i ≤ n[[1]], i ++, t = {}; dv = {};
   For[j = 1, j ≤ n[[2]], j ++,
    r = ra[f[[i, j]]]; vd = div[f[[i, j]]]; t = Append[t, r]; dv = Append[dv, vd]];
   tt = Append[tt, t]; dd = Append[dd, dv]; Print["node ", i, ": ", t, "  ", dv];
   pr = Position[tt, Max[tt[[i]]]]; ddv = Position[dd, Max[dd[[i]]]];
   If[Max[t] > 0, Print["       ", pr, "              ", ddv], Break[]];
   l = Length[pr]; If[l == 1, pr = pr, pr = ddv]; path = Append[path, pr];
   If[Max[dv] > 0,
    If[l > 1, Print["     update according to arc ", ddv, " with divergence ",
      Max[dv]]]]]; If[Max[tt[[1]]] ≠ 0, npath = {};
   For[i = 1, i ≤ n[[1]] - 1, i ++, s = path[[i]][[1]]; npath = Append[npath, s]];
   Print[" "]; Print["connected path is ", npath]; augpath = npath[[1]];
   For[i = 2, i ≤ n[[1]] - 1, i ++,
    If[augpath[[-1]] == npath[[i, 1]],
     augpath = Append[augpath, npath[[i, 2]]]]];
   For[i = 2, i ≤ n[[1]] - 1, i ++,
    If[augpath[[-1]] == npath[[i, 1]],
     augpath = Append[augpath, npath[[i, 2]]]]];
```

```
Print["the augmented path is ", augpath, " withe the arc capacity"];
pp = Partition[augpath, 2, 1]; tf = {};
For[i = 1, i ≤ Length[pp], i ++, ff = f[[pp[[i]][[1]], pp[[i]][[2]]]];
 tf = Append[tf, ff]]; Print[tf]; fp = {};
For[i = 1, i ≤ Length[tf], i ++, pf = ra[tf[[i]]]; fp = Append[fp, pf]];
mfp = tf[[Position[fp, Min[fp]][[1, 1]]]]; Print[" "];
Print["maximal flow pass of this iteration along the path ",
 augpath, " is ", mfp];
qqn = {}; For[i = 1, i ≤ Length[pp], i ++, qq = Reverse[pp[[i]]];
 qqn = Append[qqn, qq]]; Print["update residual flow along the path",
 pp, " and ", qqn]; Print[" "]; gn = {};
For[i = 1, i ≤ Length[qqn], i ++, hn = f[[qqn[[i]][[1]], qqn[[i]][[2]]]];
 gn = Append[gn, hn]];
en = Length[tf]; ln = mfp[[-1]]; an = Delete[mfp, -1];
bn = Transpose[Delete[Transpose[tf], -1]]; dn = Transpose[tf][[-1]];
in = Transpose[Delete[Transpose[gn], -1]]; ki = Transpose[gn][[-1]];
nm = {}; pnn = {};
For[i = 1, i ≤ en, i ++, mn = Min[dn[[i]], ln]; qnn = Min[ki[[i]], ln];
 pnn = Append[pnn, qnn]; nm = Append[nm, mn]]; ncn = {}; njn = {};
For[i = 1, i ≤ en, i ++, cn = Append[bn[[i]] - Reverse[an], nm[[i]]];
 ncn = Append[ncn, cn]; jn = Append[in[[i]] + an, pnn[[i]]];
 njn = Append[njn, jn]; Print[pp[[i]], "→", cn, "  and  ", qqn[[i]],
  " → ", jn]];
For[i = 1, i ≤ Length[pp], i ++, If[i == pp[[i, 1]],
  f[[pp[[i, 1]], pp[[i, 2]]]] = ncn[[i]],
  f[[pp[[i, 1]], pp[[i, 2]]]] = ncn[[i]]];];
For[i = 1, i ≤ Length[qqn], i ++, If[i == qqn[[i, 1]],
  f[[qqn[[i, 1]], qqn[[i, 2]]]] = njn[[i]],
  f[[qqn[[i, 1]], qqn[[i, 2]]]] = njn[[i]]];];
{Join[pp, ncn], Join[qqn, njn]}; af = Append[af, mfp];,
{Print["end the iteration and  "];
 Module[{a, b, c, d, e, p, r, s}, p = af; a = Transpose[p];
  b = Transpose[Delete[a, -1]]; c = Sum[b[[i]], {i, 1, Length[b]}];
```

Mathematica function for finding flow direction:

```
fdir[OF_, NF_] :=
 Module[{a, b, c, d, e, f, g, h, m, n, p, q, r, s, t, u, v, w, x, y, z, of,
   nf, ss, rr}, of = OF; nf = NF; n = Length[nf]; a = {}; b = {}; y = {};
  z = {};
  For[i = 1, i ≤ n, i++, For[j = 1, j ≤ n, j++, c = Delete[of[[i, j]], -1];
    a = Append[a, c];
    d = Last[of[[i, j]]]; b = Append[b, d]; x = Delete[nf[[i, j]], -1];
    y = Append[y, x];
    w = Last[nf[[i, j]]]; z = Append[z, w]]]; e = Partition[a, 5, 5];
  v = Partition[y, 5, 5]; h = Partition[Partition[b, 1, 1], 5, 5];
  u = Partition[Partition[z, 1, 1], 5, 5]; m = {};
  For[i = 1, i ≤ n, i++, For[j = 1, j ≤ n, j++, f = e[[i, j]] - Reverse[v[[i, j]]];
    g = Append[f, Min[h[[i, j]], u[[i, j]]]]; m = Append[m, g];]];
  p = Partition[m, 5, 5]; r = {}; ss = {};
  For[i = 1, i ≤ n, i++, For[j = i + 1, j ≤ n, j++,
    If[Max[ra[p[[i, j]]], ra[p[[j, i]]]] > 0,
     If[ra[p[[i, j]]] > ra[p[[j, i]]],
      s = Append[{{ra[p[[i, j]]], ra[p[[j, i]]]}}, {"flow pass ", i, "⟹", j}],
      s = Append[{{ra[p[[i, j]]], ra[p[[j, i]]]}}, {i, "⟸", j, " flow pass"}]],
     s = Append[{{ra[p[[i, j]]], ra[p[[j, i]]]}},
       {"no flow pass", i, " ⟺", j}]]; ss = Append[ss, s];
    q = Join[{p[[i, j]]}, {p[[j, i]]}]; r = Join[r, {q, s}]; rr = Join[r, {ss}];]];
  l = Partition[r, 4, 4]]
```

```
fmfnp = {{{0, 0, 0, 0, 0}, {-5, 10, 37, 45, .6}, {0, 13, 19, 29, .5},
   {5, 14, 27, 41, .6}, {0, 0, 0, 0, 0}},
  {{0, 0, 0, 0, 1}, {0, 0, 0, 0, 0}, {1, 9, 15, 25, .4}, {5, 12, 20, 30, 1.},
   {7, 19, 31, 60, .5}}, {{0, 0, 0, 0, 1}, {-2, 11, 20, 39, .6},
   {0, 0, 0, 0, 0}, {7, 19, 36, 55, 0.4}, {15, 22, 38, 45, 0.5}},
  {{0, 0, 0, 0, 1}, {0, 0, 0, 0, 1}, {20, 40, 60, 80, .9}, {0, 0, 0, 0, 0},
   {50, 130, 170, 250, .3}},
  {{0, 0, 0, 0, 0}, {0, 0, 0, 0, 1}, {0, 0, 0, 0, 1}, {0, 0, 0, 0, 1},
   {0, 0, 0, 0, 0}}};
```

Using Mathematica program have calculated step 1 to 6 and network updating:

Input:

```
nagpath[fmfnp]
```

Output:

```
node 1: {0, 13.05, 7.625, 13.05, 0}  {0, 30., 14.5, 21.6, 0}
      {{1, 2}, {1, 4}}          {{1, 2}}
     update according to arc {{1, 2}} with divergence 30.
node 2: {0, 0, 5., 16.75, 14.625}  {0, 0, 9.6, 25., 26.5}
      {{2, 4}}          {{2, 5}}
node 3: {0, 10.2, 0, 11.7, 15.}  {0, 24.6, 0, 19.2, 15.}
      {{3, 5}}          {{3, 2}}
node 4: {0, 0, 45., 0, 45.}  {0, 0, 54., 0, 60.}
      {{4, 3}, {4, 5}}          {{4, 5}}
     update according to arc {{4, 5}} with divergence 60.
node 5: {0, 0, 0, 0, 0}  {0, 0, 0, 0, 0}


connected path is {{1, 2}, {2, 4}, {3, 5}, {4, 5}}
the augmented path is {1, 2, 4, 5} withe the arc capacity
{{-5, 10, 37, 45, 0.6}, {5, 12, 20, 30, 1.}, {50, 130, 170, 250, 0.3}}
```

```
maximal flow pass of this iteration along the path
 {1, 2, 4, 5} is {-5, 10, 37, 45, 0.6}
update residual flow along the path
 {{1, 2}, {2, 4}, {4, 5}} and {{2, 1}, {4, 2}, {5, 4}}
{1, 2}→{-50, -27, 27, 50, 0.6}  and  {2, 1} → {-5, 10, 37, 45, 0.6}
{2, 4}→{-40, -25, 10, 35, 0.6}  and  {4, 2} → {-5, 10, 37, 45, 0.6}
{4, 5}→{5, 93, 160, 255, 0.3}  and  {5, 4} → {-5, 10, 37, 45, 0.6}
network updated with
{{{0, 0, 0, 0, 0}, {-50, -27, 27, 50, 0.6},
  {0, 13, 19, 29, 0.5}, {5, 14, 27, 41, 0.6}, {0, 0, 0, 0, 0}},
 {{-5, 10, 37, 45, 0.6}, {0, 0, 0, 0, 0}, {1, 9, 15, 25, 0.4},
  {-40, -25, 10, 35, 0.6}, {7, 19, 31, 60, 0.5}},
 {{0, 0, 0, 0, 1}, {-2, 11, 20, 39, 0.6}, {0, 0, 0, 0, 0},
  {7, 19, 36, 55, 0.4}, {15, 22, 38, 45, 0.5}},
 {{0, 0, 0, 0, 1}, {-5, 10, 37, 45, 0.6}, {20, 40, 60, 80, 0.9},
  {0, 0, 0, 0, 0}, {5, 93, 160, 255, 0.3}}, {{0, 0, 0, 0, 0},
  {0, 0, 0, 0, 1}, {0, 0, 0, 0, 1}, {-5, 10, 37, 45, 0.6}, {0, 0, 0, 0, 0}}}
```

Input:

```
nagpath[ftf]
```

Only short outputs have been shown.

```
{1, 4}→{-36, -13, 13, 36, 0.6}  and  {4, 1} → {5, 14, 27, 41, 0.6}
{4, 3}→{-21, 13, 46, 75, 0.6}  and  {3, 4} → {12, 33, 63, 96, 0.4}
{3, 5}→{-26, -5, 24, 40, 0.5}  and  {5, 3} → {5, 14, 27, 41, 0.6}
```

Input:

```
nagpath[ftf]
```

Only short outputs have been shown

```
{1, 3}→{-29, -6, 6, 29, 0.5}  and  {3, 1} → {0, 13, 19, 29, 0.5}
{3, 4}→{-17, 14, 50, 96, 0.4}  and  {4, 3} → {-21, 26, 65, 104, 0.5}
{4, 5}→{-24, 74, 147, 255, 0.3}  and  {5, 4} → {-5, 23, 56, 74, 0.5}
```

Input:

```
nagpath[ftf]
```

Output:

```
node 1: {0, 0., 0., 0., 0}  {0, 60., 29., 43.2, 0}
end the iteration and
sum of total flow
 {{-5, 10, 37, 45, 0.6}, {5, 14, 27, 41, 0.6}, {0, 13, 19, 29, 0.5}}
  is = {0, 37, 83, 115, 0.5}
```

```
network updated with

{{{0, 0, 0, 0, 0}, {-50, -27, 27, 50, 0.6},
  {-29, -6, 6, 29, 0.5}, {-36, -13, 13, 36, 0.6}, {0, 0, 0, 0, 0}},
 {{-5, 10, 37, 45, 0.6}, {0, 0, 0, 0, 0}, {1, 9, 15, 25, 0.4},
  {-40, -25, 10, 35, 0.6}, {7, 19, 31, 60, 0.5}},
 {{0, 13, 19, 29, 0.5}, {-2, 11, 20, 39, 0.6}, {0, 0, 0, 0, 0},
  {-17, 14, 50, 96, 0.4}, {-26, -5, 24, 40, 0.5}},
 {{5, 14, 27, 41, 0.6}, {-5, 10, 37, 45, 0.6}, {-21, 26, 65, 104, 0.5},
  {0, 0, 0, 0, 0}, {-24, 74, 147, 255, 0.3}},
 {{0, 0, 0, 0, 0}, {0, 0, 0, 0, 1}, {5, 14, 27, 41, 0.6},
  {-5, 23, 56, 74, 0.5}, {0, 0, 0, 0, 0}}}
```

Using Mathematica program we have found flow direction:

Input:

```
fdir[fmfnp, ftf]
```

Output:

```
{{{{-55, -17, 64, 95, 0.6}, {-45, -37, -10, 5, 0.6}}, {{13.05, -13.05}, {flow pass , 1, ⟹, 2}},
  {{-29, 7, 25, 58, 0.5}, {-29, -19, -13, 0, 0.5}}, {{7.625, -7.625}, {flow pass , 1, ⟹, 3}}},
 {{{-31, 1, 40, 77, 0.6}, {-41, -27, -14, -5, 0.6}}, {{13.05, -13.05}, {flow pass , 1, ⟹, 4}},
  {{0, 0, 0, 0, 0}, {0, 0, 0, 0, 0}}, {{0, 0}, {no flow pass, 1, ⟺, 5}}},
 {{{-24, -6, 6, 24, 0.4}, {-41, -9, 9, 41, 0.6}}, {{0., 0.}, {no flow pass, 2, ⟺, 3}},
  {{-30, 2, 45, 70, 0.6}, {-45, -37, -10, 5, 0.6}}, {{13.05, -13.05}, {flow pass , 2, ⟹, 4}}},
 {{{-53, -12, 12, 53, 0.5}, {0, 0, 0, 0, 1}}, {{0., 0}, {no flow pass, 2, ⟺, 5}},
  {{-89, -31, 22, 72, 0.4}, {-84, -25, 34, 101, 0.5}}, {{-2.6, 3.25}, {3, ⟸, 4, flow pass}}},
 {{{-25, -2, 43, 71, 0.5}, {-41, -27, -14, -5, 0.6}}, {{10.875, -13.05}, {flow pass , 3, ⟹, 5}},
  {{-205, -17, 96, 274, 0.3}, {-74, -56, -23, 5, 0.5}}, {{11.1, -18.5}, {flow pass , 4, ⟹, 5}}}}
```