MAXIMAL FLOW MODEL AND ITS APPLICATIONS

Mobin Ahmad

Department of Mathematics, Faculty of Science, Jazan University, Jazan 45142, Saudi Arabia

Email: msyed@jazanu.edu.sa, profmobin@yahoo.com

ABSTRACT

Most extreme stream issue. In the conventional most extreme stream issue, there is a capacitated arrange and the objective is to send however much of a solitary item as could be expected between two recognized hubs, without surpassing the circular segment limit limits. The issue has many applications including: shipping cargo in transportation arrange and directing liquid through a pressure driven system.

Keyword- Network, Maximal-Flow Model, Residual network, Source-Sink

1. INTRODUCTION

A stream arranges is characterized as a coordinated chart including a source(S) and a sink (T) and a few different hubs associated with edges. Each edge has an individual limit which is the most extreme utmost of stream that edge could permit. Stream in the system ought to take after the accompanying conditions:

For any non-source and non-sink hub, the info stream is equivalent to yield stream.

•For any edge (Ei) in the system, $0 \leq flow(Ei) \leq Capacity(Ei)$.

•Total stream out of the source hub is equivalent aggregate to stream in to the sink hub.

•Net stream in the edges takes after skew symmetry i.e. F(u,v)=−F(v,u) where F(u,v) is spill out of hub u to hub v.

This prompts a conclusion where you need to whole up every one of the streams between two nodes (either bearing) to discover net stream between the hubs at first. Greatest Flow:

It is characterized as the greatest measure of stream that the system would permit to spill out of source to sink. Different calculations exist in tackling the most extreme stream issue.

The most extreme stream issue is again organized on a system; however here the circular segment limits, or upper limits, are the main pertinent parameters. The issue is to locate the most extreme stream conceivable from some given source hub to a given sink hub. A system display is in Fig. All curve costs are zero, yet the cost on the circular segment leaving the sink is set to - 1. Since the objective of the enhancement is to limit cost, the most extreme stream conceivable is conveyed to the sink hub.

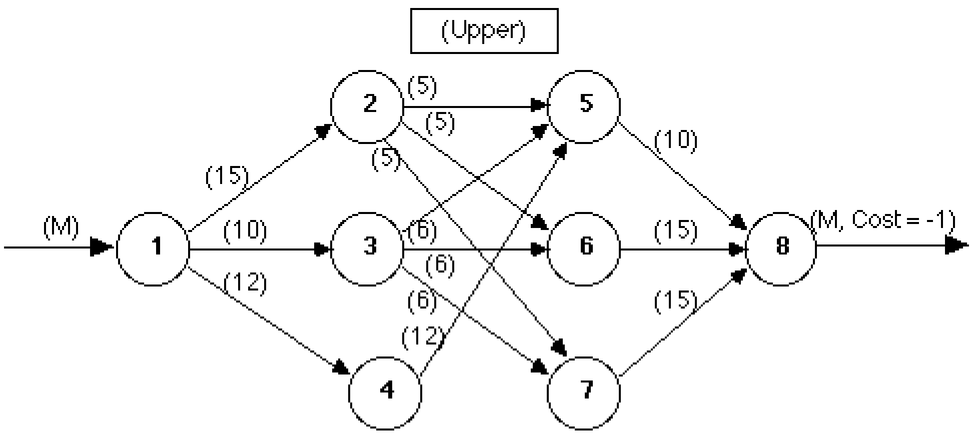


Figure: Network model for the maximum flow problem. [36]

The solution to the example is in Fig. 18. The maximum flow from node 1 to node 8 is 30 and the flows that yield this flow are shown on the figure. The heavy arcs on the figure are called the minimal cut. These arcs are the bottlenecks that are restricting the maximum flow. The fact that the sum of the capacities of the arcs on the minimal cut equals the maximum flow is a famous theorem of network theory called the max flow min cut theorem. The arcs on the minimum cut can be identified using sensitivity analysis.

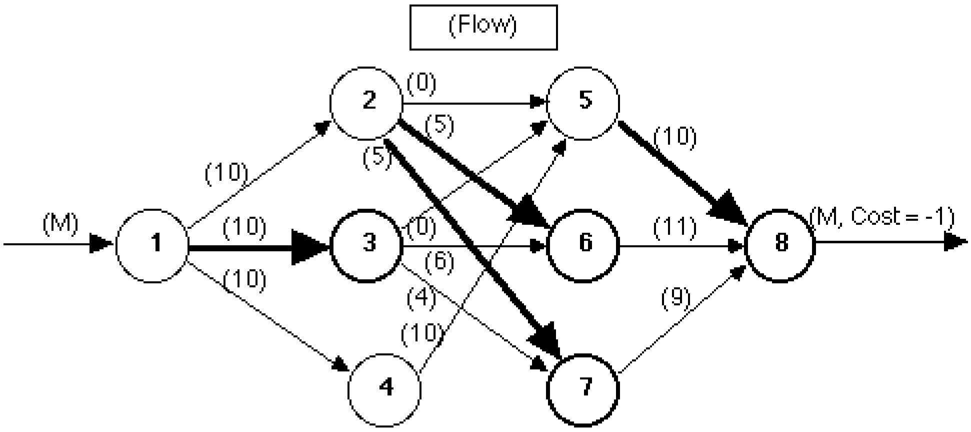


Figure: Solution. Maximum flow = 30[36]

2. REVIEW OF LITERATURE

The primary calculation intended for arrange stream issues was the system simplex strategy for Dantzig [1]. It is a variation of the direct programming simplex strategy intended to exploit the combinatorial structure of system stream issues. Variations of the simplex strategy that abstain from cycling give an exponential bound on the unpredictability of all the system stream issues. (Cunningham [2] gives a rich hostile to cycling procedure for the system simplex technique in light of diagram theoretic properties of the base cost flow issue). As of late, Goldfarb and Hao [3] have planned a variation of the primal system simplex strategy for the most extreme stream issue that keeps running in firmly polynomial time. Orlin [4] composed a variation of the double system simplex strategy for the base cost course issue that keeps running in firmly polynomial time: For quite a while, the system simplex technique has been the strategy for decision practically speaking, specifically for the base cost dissemination issue; for huge occurrences of difficult issues, the new scaling calculations are presumably better, notwithstanding. The primary pseudo polynomial calculation for the most extreme stream issue is the enlarging way calculation of Ford and Fulkerson [5]

Dinic [6] and Edmonds and Karp [7] autonomously got polynomial variants of the enlarging way calculation. From that point forward, a few progressively effective calculations have been created. Section 2 introduces the push/re label strategy, as of late proposed by Goldberg [8] and Goldberg and Tarjan [9], alongside a portion of its more effective variations. The main pseudo polynomial calculation for the base cost dissemination issue is the out-of-kilter strategy, which was created freely by Yakovleva [10], Minty [11], and Fulkerson [12]. The primary polynomial calculation for the base cost course issue is because of Edmonds and Karp [13]. To build up this calculation Edmonds and Karp presented the system of scaling, which has ended up being a valuable device in the plan and examination of calculations for an assortment of combinatorial improvement issues. The greatest stream calculations of Dinic [14] and Edmonds and Karp [15] are unequivocally polynomial, yet the base cost dissemination calculation of Edmonds 1 All logarithms in this paper without an express base will be base two. 2 For a more formal meaning of polynomial and unequivocally polynomial calculations, see [16]. System Flow Algorithms 103 and Karp [17] isn't. The primary unequivocally polynomial calculation for the minimum cost dissemination issue was planned by Tardos [18]. Part 4 and Section 5.3 are given to ongoing firmly polynomial calculations for the base cost dissemination issue. The primary expanding way calculations for the summed up stream issue were created autonomously by Jewell [19] and Onaga [20]. Numerous pseudo opolynomial least cost course calculations have been adjusted for the summed up stream issue (see [21] for a review). The principal polynomial-time calculation for the summed up stream issue was the ellipsoid technique [22]. Kapoor and Vaidya [23] have demonstrated to accelerate Karmarkar [24] — or Renegar [25] — type inside point calculations on arrange stream issues by exploiting the exceptional structure of the lattices utilized as a part of the straight programming definitions of these issues. Vaidya's calculation [26] is the quickest at present known calculation for the summed up stream issue. The main polynomial calculations for the summed up stream issue that are not founded on broadly useful straight programming strategies are because of Goldberg, Plotkin, and Tardos [27]. These calculations are talked about in Chapter 6. The presence of an emphatically polynomial calculation for the summed up stream issue is an intriguing open inquiry. Essential unique instances of system stream issues that won't be canvassed in this overview are the bipartite coordinating issue and its weighted form, the task issue. These issues can be expressed as greatest stream and least cost dissemination issues, individually, on systems with unit limits and an uncommon structure Some of the productive calculations for the more broad issues have advanced from effective calculations grew before for these less difficult issues. Konig's [28] evidence of a decent portrayal of the most extreme size of a coordinating in a bipartite chart gives an O(ww)- time calculation for finding a greatest coordinating. The Ford-Fulkerson greatest stream calculation can be seen as an expansion of this calculation. Hopcroft and Karp [29] gave a 0(^/nm) calculation for the bipartite coordinating issue. Indeed, even and Tarjan watched [30] that Dinic's greatest stream calculation, when connected to the bipartite coordinating issue, acts also to the Hopcroft-Karp calculation and keeps running in 0(^/nm) time too. A variety of the Goldberg-Tarjan most extreme stream calculation (which can be seen as a speculation of Dinic's calculation) can be effectively appeared to prompt a similar bound [31] notwithstanding late advance on related issues, the Ofv/nm) bound has not been made strides. The main calculation for the task issue is the Hungarian strategy for Kuhn [32], The out-of-kilter calculation is an augmentation of this calculation to the base cost flow issue. The Hungarian strategy tackles the task issue in 0(n) most limited way calculations. Edmonds and Karp [33] and Tomizawa [34] have watched that the double factors can be kept up so these most brief way calculations are on charts with non-negative curve costs. Joined with the most brief way calculation of [35], this perception gives an O(n(m + nlogn)) destined for the issue. Gabow

3. APPLICATIONS OF MAXIMUM FLOW

3.1 Edges-Disjoint Paths

One of the most effortless uses of greatest streams is figuring the most extreme number of edge-disjoint ways between two indicated vertices s and t in a coordinated chart G utilizing greatest streams. An arrangement of ways in G is edge-disjoint if each edge in G shows up in at most one of the ways; a few edge-disjoint ways may go through a similar vertex, in any case. On the off chance that we give each edge limit 1, at that point the maxflow from s to t appoints a stream of either 0 or 1 to each edge. Since any vertex of G lies on at most two immersed edges (one in and one out, or none by any stretch of the imagination), the sub chart S of soaked edges is the association of a few edge-disjoint ways and cycles. In addition, the quantity of ways is precisely equivalent to the estimation of the stream. Extricating the real ways from S is simple—simply take after any guided way in S from s to t, expel that way from S, and recurse. On the other hand, we can change any gathering of k edge-disjoint ways into a stream by pushing one unit of stream along every way from s to t; the estimation of the subsequent stream is precisely k. It takes after that the maxflow calculation really figures the biggest conceivable arrangement of edge-disjoint ways. The general running time is O(V E), simply like for greatest bipartite matchings. A similar calculation can likewise be utilized to discover edge-disjoint ways in undirected diagrams. We essentially supplant each undirected edge in G with a couple of coordinated edges, each with unit limit, and process a most extreme spill out of s to t in the subsequent coordinated diagram G 0 utilizing the Ford-Fulkerson calculation. For any edge uv in G, if our maximum stream immerses both coordinated edges uv and vu in G 0 , we can expel the two edges from the stream without changing its esteem. In this manner, without loss of all inclusive statement, the greatest stream doles out a course to each soaked edge, and we can separate the edge-disjoint ways via looking through the diagram of coordinated immersed edges.

3.2 Vertex Capacities And Vertex-Disjoint Paths

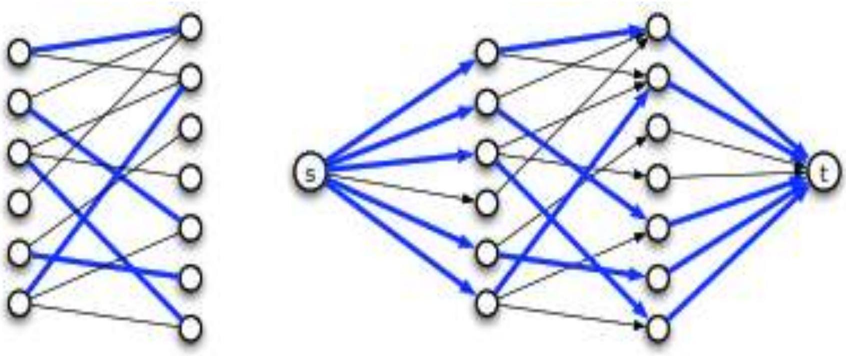
Assume we have limits on the vertices and in addition the edges. Here, notwithstanding our different limitations, we require that for any vertex v other than s and t, the aggregate stream into v (and hence the aggregate stream out of v) is at most some non-negative esteem c(v). How might we register a most extreme stream with these new limitations? One plausibility is to adjust our current calculations to consider these vertex limits. Given a stream f , we can characterize the lingering limit of a vertex v to be its unique limit less the aggregate stream into v:

$$c_f(v) = c(v) - \sum_u f(u \to v).$$

Since we can't send any more stream into a vertex with leftover limit 0 we expel from the remaining diagram Gf each edge uv that shows up in G whose head vertex v is immersed. Something else, the increasing way calculation is unaltered. In any case, a significantly less difficult strategy is to change the contribution to a customary stream organize, with just edge limits. In particular, we supplant each vertex v with two vertices vin and vout, associated byan edge vinvout with limit c(v), and afterward supplant each coordinated edge uv with the edge uoutvin (keeping a similar limit). At last, we process the most extreme spill out of sout to tin in this altered stream arrange. It is presently simple to figure the most extreme number of vertex-disjoint ways from s to t in any coordinated chart. Basically give each vertex limit 1, and register a most extreme stream!

3.3 Maximum Machings In Bipartite Graphs

Another characteristic utilization of most extreme streams is discovering huge matchings in bipartite charts. A coordinating is a sub graph in which each vertex has degree at most one, or equally, an accumulation of edges to such an extent that no two offer a vertex. The issue is to locate the coordinating with the most extreme number of edges in a given bipartite diagram. We can tackle this issue by diminishing it to a most extreme stream issue as takes after. Give G a chance to be the given bipartite chart with vertex set U ∪ W, to such an extent that each edge joins a vertex in U to a vertex in W. We make another coordinated chart G 0 by (1) arranging each edge from U to W, (2) including two new vertices s and t, (3) adding edges from s to each vertex in U, and (4) including edges from every vertex in W to t. At long last, we appoint each edge in G 0 a limit of 1. Any coordinating M in G can be changed into a stream f M in G 0 as takes after: For each edge uw in M, push one unit of stream along the way suwt. These ways are disjoint aside from at s and t, so the subsequent stream fulfills the limit imperatives. In addition, the estimation of the subsequent stream is equivalent to the quantity of edges in M. Then again, think about any (s, t)- stream f in G 0 figured utilizing the Ford-Fulkerson enlarging way calculation. Since the edge limits are whole numbers, the Ford-Fulkerson calculation doles out a whole number stream to each edge. (This is anything but difficult to check by enlistment, imply, indicate.) Moreover, since each edge has unit limit, the figured stream either soaks (f (e) = 1) or evades (f (e) = 0) each edge in G 0 . At long last, since at most one unit of stream can enter any vertex in U or leave any vertex in W, the soaked edges from U to W shape a coordinating in G. The span of this coordinating is precisely |f |. In this way, the span of the greatest coordinating in G is equivalent to the estimation of the most extreme stream in G 0 , and gave we register the max flow utilizing increasing ways, we can change over the real max flow into a greatest coordinating. The greatest stream has an incentive at most min{|U|,|W|} = O(V), so the Ford-Fulkerson calculation keeps running in O(VE) time.



Binary Assignment Problems

Greatest cardinality matching are an exceptional instance of a general group of supposed task problems.1 An un weighted paired task issue includes two disjoint limited sets X and Y , which normally speak to two various types of assets, for example, website pages and servers, occupations and machines, lines and sections of a network, healing centers and assistants, or clients and pints of dessert. Our undertaking is to pick the biggest conceivable accumulation of sets (x, y) as could be allowed, where $x \in X$ and $y \in Y$, subject to a few requirements of the accompanying structure:

- Each component $x \in X$ can show up in at most c(x) sets.
- Each component $y \in Y$ can show up in at most c(y) sets
- Each match $(x, y) \in X \times Y$ can show up in the yield at most c(x, y) times.

Every upper bound c(x), c(y), and c(x, y) is either a (normally little) non-negative number or ∞. Instinctively, we make each combine in our yield by relegating a component of X to a component of y. The most extreme coordinating issue is an uncommon case, where c(z) = 1 for all $z \in X \cup Y$, and each c(x, y) is either 0 or 1, contingent upon whether the match x y characterizes an edge in the hidden bipartite diagram. Here is a marginally additionally intriguing illustration. An adjacent school, renowned for its burdensome authoritative obstacles, chooses to arrange a move. Each match of understudies (one kid, one young lady) who needs to move must enroll ahead of time. School controls constrain every kid young lady match to at most three moves together, and restrains every understudy to at most ten moves generally speaking. How might we expand the quantity of moves? This is a paired task issue for the set X of young ladies and the set Y of young men. For every young lady x and kid y, we have c(x) = 10, c(y) = 10, and either c(x, y) = 3 (if x and y enrolled to move) or c(x, y) = 0 (in the event that they didn't). This twofold task issue can be diminishes to a standard greatest stream issue as takes after. We build a stream organize G = (V, E) with vertices $X \cup Y \cup \{s, t\}$ and the accompanying edges:

- an edge sx with limit c(x) for every $x \in X$,
- an edge yt with limit c(y) for every $y \in Y$.
- an edge xy with limit c(x, y) for every $x \in X$ and $y \in Y$, and

Since every one of the edges have number limits, the Ford-Fulkerson calculation builds a whole number most extreme stream $f*$. This stream can be disintegrated into the whole of $|f*|$ ways of the shape sxyt for some $x \in X$ and $y \in Y$. For each such way, we report the match (x, y). (Identically, the combine (x, y) shows up in our yield accumulation f (xy) times.) It is anything but difficult to confirm (imply, imply) that this gathering of sets fulfills all the vital imperatives. On the other hand, any legitimate gathering of r sets can be changed into a plausible whole number stream with esteem r in G. In this manner, the biggest lawful accumulation of sets relates to a greatest stream in G. So our calculation is right.

CONCLUSION

An expanding way is a basic way from source to sink which do exclude any cycles and that go just through positive weighted edges. A lingering system diagram shows the amount more stream is permitted in each edge in the system chart. On the off chance that there are no enlarging ways conceivable from S to T, at that point the stream is greatest. The outcome i.e. the greatest stream will be the aggregate stream out of source hub which is likewise equivalent to add up to stream in to the sink hub.

REFERENCES

1. Cunningham, W.H. (1976): A network simplex method. Math. Program. //, 105-116
2. Goldfarb, D., Hao, J. (1988): A primal simplex algorithm that solves the maximum flow problem in at most nm pivots and 0(t?m) time. Technical Report, Department of Industrial Engineering and Operations Research, Columbia University
3. Grigoriadis, M.D. (1986): An efficient implementation of the network simplex method. Math. Program. Study 26, 83-111
4. Orlin, J.B. (1984): Genuinely polynomial simplex and non-simplex algorithms for the minimum cost flow problem. Technical Report No. 1615-84, Sloan School of Management, M.I.T.
5. Ford jr., L.R., Fulkerson, D.R. (1962): Flows in networks. Princeton University Press, Princeton, NJ
6. Ford jr., L.R., Fulkerson, D.R. (1956): Maximal flow through a network. Can. J. Math. 8, 399^104
7. Goldberg, A.V. (1985): A new max-flow algorithm. Technical Report MIT/LCS/ TM-291, Laboratory for Computer Science, M.I.T.
8. Dinic, E.A. (1970): Algorithm for solution of a problem of maximum flow in networks with power estimation. Sov. Math. Dokl. 11, 1277-1280
9. Edmonds, J., Karp, R.M. (1972): Theoretical improvements in algorithmic efficiency for network flow problems. J. Assoc. Comput. Mach. 19, 248-264
10. Goldberg, A.V., Tarjan, R.E. (1986): A new approach to the maximum flow problem. Proc. 18th ACM STOC, pp. 136-146

11. Fulkerson, D.R. (1961): An out-of-Kilter method for minimal cost flow problems. SIAM J. Appl. Math. 9, 18-27
12. Minty, G.J. (1960): Monotone networks. Proc. R. Soc. Lend., Ser. A 257, 194-212
13. [105] Yakovleva, M.A. (1959): A problem on minimum transportation cost. In: Nemchinov, V.S. (ed.): Applications of mathematics in economic research. Izdat. Social'noEkon. Lit., Moscow, pp. 390-399
14. Renegar, J. (1988): A polynomial time algorithm, based on Newton's method, for linear programming. Math. Program. 40, 59-94
15. Tardos, E. (1985): A strongly polynomial minimum cost circulation algorithm. Combinatorica 5, 247-255
16. Jewell, W.S. (1962): Optimal flow through networks with gains. Oper. Res. 70, 476-499
17. Truemper, K. (1977): On max flows with gains and pure min-cost flows. SIAM J. Appl. Math. 32, 450-456
18. [70] Khachiyan, L.G. (1980): P
19. Kapoor, S., Vaidya, P.M. (1986): Fast algorithms for convex quadratic programming and multicommodity flows. Proc. 18th ACM STOC, pp. 147-159
20. Karmarkar, N. (1984): A new polynomial-time algorithm for linear programming. Combinatorica 4, 373-395
21. Goldberg, A.V., Plotkin, S.A., Tardos, E. (1988): Combinatorial algorithms for the generalized circulation problem Technical Report STAN-CS-88-1209, Stanford University (Also available as Technical Memorandum MIT/LCS/TM-358, Laboratory for Computer Science, M.I.T.)

22. Konig, D. (1950): Theorie der endlichen und unendlichenGraphen. Chelsea Publishing Co., New York, NY

23. Hopcroft, J.E., Karp, R.M. (1973): An n i/2 algorithm for maximum matching in bipartite graphs. SIAM J. Comput. 2, 225-231

24. Even, S., Tarjan, R.E. (1975): Network flow and testing graph connectivity. SIAM J. Comput. 4, 507-518

25. Kuhn, H.W. (1955): The Hungarian method for the assignment problem. Nav. Res. Logist. 2, 83-97

26. Orlin, J.B., Ahuja, R.K. (1987): New distance-directed algorithms for maximumflow and parametric maximum-flow problems. Sloan Working Paper 1908-87, Sloan School of Management, M.I.T

27. Gabow, H.N. (1985): Scaling algorithms for network problems. J. Comput. Syst. Sci. 31, 148-168

28. Tomizawa, N. (1972): On some techniques useful for solution of transportation networks problems. Networks 1, 173-194

29. [106] Zadeh, N. (1973): A bad network problem for the simplex method and other minimum cost flow problems. Math. Program. 5, 255-266

30 Fredrickson, G. (1987): Fast algorithms for shortest paths in planar graphs with applications. SIAM J. Comput. 16, 1004-1022

32 Fujishige, S. (1986): A capacity-rounding algorithm for the minimum-cost circulation problem: A dual framework of the Tardos algorithm. Math.Program. 35, 298-308

33. Fulkerson, D.R. (1961): An out-of-Kilter method for minimal cost flow problems. SIAM J. Appl. Math. 9, 18-27

34. Gabow, H.N., Tarjan, R.E. (1988): Almost-optimal speed-ups of algorithms for matching and related problems. Proc. 20th ACM STOC, pp. 514-527

35. Gabow, H.N., Tarjan, R.E.: Faster scaling algorithms for network problems. SIAM J. Comput., to appear