New Mathematical Model for Finding Minimum Vertex Cut Set

Tina Beseri Sevim[1], Hakan Kutucu[2], Murat Ersen Berberler[3]
[1,2]Izmir Institute of Technology, Izmir, Turkey
[3]Dokuz Eylul University, Izmir, Turkey
[1]*tinabeseri@iyte.edu.tr*, [2]*hakankutucu@iyte.edu.tr*, [3]*murat.berberler@deu.edu.tr*

***Abstract*— In this paper, we consider the vertex separator problem. Given an undirected graph *G*, the vertex separator problem consists in identifying a minimum number of vertex set whose removal disconnects *G*. We present a new mathematical model for solving this problem and also present computational results on graphs with various density.**

Keywords— connectivity; minimum vertex cut set; maximum flow

I. INTRODUCTION

Connectivity of graphs is one of the most famous and fundamental notions for analyzing various types of graph problems. It is also useful in network design problems. The connectivity of a graph is an important measure of its robustness and reliability as a network. Connectivity is closely related to minimum cut maximum flow theorem discovered by P. Elias, A. Feinstein and C. F. Shannon [1] and L. R. Ford and D. R. Fulkerson [2]. Therefore, many graph algorithms have been proposed for solving the graph connectivity problems are based on the minimum cut maximum flow theorem. One can use a maximum flow problem to find a local minimum cut between any pair of vertices. Hao and Orlin [5] proposed an algorithm to compute the maximum flow problem running in O(mnlog n2/m) time in either a directed or an undirected network. Afterwards, Karger [6] presented an algorithm for finding the minimum cut of an undirected graph in O(n2log3n) time. The algorithm does more than find a single minimum cut; it finds all of them.

In a connected graph G, a separator or a vertex-cut S is a subset of vertices whose removal separates G into distinct connected components. S is called a (a− b) separator if and only if it disconnects non-adjacent vertices a and b. A minimal (a,b) separator is an (a,b) separator such that no subset of it is an (a,b) separator. A separator is called a minimal separator if it is a minimal (a,b) separator for some pair of vertices a,b. Kloks and Kratsch [8] give an algorithm to compute all minimal separator of a graph G in polynomial time O(n5) per separator. In [7], Berry et al. present an efficient algorithm which computes the set of minimal separators of G in O(n3) per separator. Their process is based on a new structural result, derived from the work of Kloks and Kratsch on listing all the minimal separators of a graph.

The vertex separator problem (VSP) in an undirected graph asks for a partition of its vertices into nonempty three subsets A, B, C such that there is no edge between A and B, and |C| is minimized subject to a bound on max{|A|, |B|}. The VSP is NP-hard [11]. The problem we study in this paper is a generalization of VSP in which max {|A|, |B|} ≤ |N| − 1. Souza and Balas [9] discuss a polyhedral approach for the VSP and devise a branch and cut algorithm. This is the first work that addresses a polyhedral analysis of the VSP. Then, Biha and Meurs [10] study the VSP from a polyhedral point of view, and give a complete description of the associated polytope.

The graphs we consider are simple, finite, undirected, loopless. A graph is denoted by G = (V, E), where V is the vertex set and E is the edge set. If u, v ∈ V, we will denote by uv an edge between u and v. If W ⊆ V is a vertex subset of G, then the set of edges that have only one vertex in W is called a cut and is denoted by δG (W). When it is clear that the cut is taken with respect to G, we will simply denote it by δ (W). We will write δ (v) for δ ({v}). A cut δ (W) such that s ∈ W and t ∈ V \ W will be called an st − cut. Let v and w be two non-adjacent vertices in a graph G. A set S of vertices is a v – w separating set if v and w lie in different components of G− S; that is, if every v− w path contains a vertex in S. The minimum order of a v− w separating set is called the v− w connectivity and is denoted by κ (v, w). The connectivity κ (G) of a graph G = (V, E) is the smallest number of vertices whose deletion from G produces a disconnected or trivial graph. Clearly, a complete graph cannot be disconnected by deleting vertices, but all other graphs can. It is not hard to see that in any case, κ (G) = min {κ (v, w) | v, w ∈ V}.

This article is organized as follows. In the next section, we propose an integer programming formulation for the vertex cut problem. In Section 3, we give some concluding remarks.

II. MATHEMATICAL FORMULATION FOR THE MINIMUMVERTEX CUT SET

The following mathematical model gives the maximum cardinality set of vertex-disjoint paths between a source vertex and a sink vertex.

$$\max \left[\sum_{(s,j)\epsilon E} x_{sj} - \sum_{(j,s)\epsilon E} x_{js} + \sum_{(j,t)\epsilon E} x_{jt} - \sum_{(t,j)\epsilon E} x_{tj} \right] / 2 \qquad (1)$$

subject to

$$\sum_{j\in\delta^-(i)} x_{ji} - \sum_{j\in\delta^+(i)} x_{ij} = \begin{cases} -1 & \text{if } i = s, \\ 0 & \text{otherwise, } \forall\, i \in N \\ 1 & \text{if } i = t, \end{cases} \quad (2)$$

$$\sum_{(i,j)\in E} x_{ij} \le \begin{cases} |N| - 1 & \text{if } j = s,t \\ 1 & \text{otherwise, } \forall\, j \in N \end{cases} \quad (3)$$

$$0 \le x_{ij} \le 1, \ \forall\, (i,j) \in E \quad (4)$$

$$x_{ij} = 0 \vee 1, \ \forall\, (i,j) \in E \quad (5)$$

where $\delta_M^+(i)$ is the set of arcs directed out of vertex *i* and $\delta_M^-(i)$ is the set of arcs directed into vertex *i* in the graph $G = (V, E)$. The objective function of the integer programming problem (1)-(3) and (5) yields the maximum number of vertex-disjoint paths between a pair of vertices.

It is well known that the dual of the maximum flow problem is the minimum-cut problem. By the similar idea, the dual of the maximum cardinality set of vertex-disjoint paths gives a minimum vertex separator (minimum vertex cut). The dual problem of (1)-(4) (without the constraints (5)) can be formulated as follows:

$$min \quad \sum_{(i,j)\in E} z_{ij} + \sum_{(i\in N)} w_i + (|N| - 1)(w_s + w_t) \quad (6)$$

subject to

$$u_j - u_i + w_j + z_{ij} \ge 0, \ \forall\, (i,j) \in E, \ i \ne s,t \ \wedge \ j \ne s,t, \quad (7)$$

$$u_j + w_j + z_{sj} \ge \tfrac{1}{2}, \ \forall\, (s,j) \in E, \ j \ne t, \quad (8)$$

$$-u_j + w_s + z_{js} \ge -\tfrac{1}{2}, \ \forall\, (j,s) \in E, \ j \ne t, \quad (9)$$

$$-u_j + w_t + z_{jt} \ge \tfrac{1}{2}, \ \forall\, (j,t) \in E, \ j \ne s, \quad (10)$$

$$u_j + w_j + z_{tj} \ge -\tfrac{1}{2}, \ \forall\, (t,j) \in E, \ j \ne s, \quad (11)$$

$$w_i \ge 0, \quad z_{ij} \ge 0. \quad (12)$$

In this dual linear program, u_i is the dual variable for the flow balance constraint (2) at vertex *i*. w_i and z_{ij} are the dual variables for constraints (3) and (4), respectively.

III. Computational Results

In this section, we present numerical results obtained for the minimum vertex cut set. LP model was implemented using CPLEX 11.0 and tested on a workstation with a 2.8 Ghz processor with 8 cores and 3GB RAM.

Our random problem generator creates test problems. It uses the following two input parameters: the number of vertices $|V|$ and the density of the number of edges $|E|$ in the graph. Random problems with 10 to 100 vertices were generated, and we tested five instances of each size. Table 1 reports the average results obtained for randomly generated problems. Abbreviations used in the table are:

n : the number of vertices of the problem
κ : the number of vertex-disjoint paths
Cpu : the time in seconds
d : edge density of the graph $\left(d = \frac{2\,|E|}{|V|\,|V-1|}\right)$

TABLE I. Computational Results

	d=0.25		*d=0.50*		*d=0.75*		*d=0.95*		*d=1*	
n	κ	Cpu	κ	Cpu	K	Cpu	κ	Cpu	κ	Cpu
10	1	1,047	3	0,578	6	0,359	7	0,235	9	0,00
20	1	1,953	6	1,343	8	1,000	15	0,454	19	0,00
30	4	1,844	9	1,985	16	1,282	24	0,532	29	0,00
40	5	4,141	14	2,875	20	1,797	34	0,656	39	0,00
50	7	4,656	18	3,641	30	1,828	43	0,703	49	0,00
60	7	6,047	21	4,375	37	2,828	53	0,844	59	0,00
70	9	7,063	22	6,000	41	3,828	62	1,032	69	0,00
80	12	8,250	29	7,063	48	5,359	71	1,406	79	0,00
90	13	9,782	34	9,188	59	6,047	79	2,125	89	0,00
100	15	12,297	39	13,688	63	10,250	89	2,688	99	0,00

IV. Conclusion

In this article, we have introduced a new mathematical model for finding all vertex-disjoint paths between one pair of vertices in an undirected graph. Then, by the dual of this model we solved the vertex cut set problem between any pair of vertices. The latter model requires $|V| - \delta(G)$ (minimum degree of the graph G) runs to find the minimum vertex cut set in a graph.

References

[1] P. Elias, A. Feinstein, and C. F. Shannon, "A note on the maximum flow through a network, "IRE Transaction on Information Theory, IT-2, 1956, pp. 117–119.

[2] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," Canadian Journal of Mathematics, 1956, pp. 399-404.

[3] M. R. Henzinger, S. Rao, and H. N. Gabow, 2000, "Computing vertex connectivity: New bounds from old techniques,". Journal of Algorithms, vol. 34, pp. 222-250.

[4] M. Sharir, "A strong-connectivity algorithm and its applications in data flow analysis," Computers and Mathematics with Applications, 1981,vol.7, pp. 67-72.

[5] J. Hao, J.B. Orlin, "A faster algorithm for finding the minimum cut in a directed graph," J. Algorithms, 1994, vol.17, pp. 424-446.

[6] D. R. Karger and C. Stein, "A New Approach to the Minimum Cut Problem," Journal of the ACM, 1996, vol. 43, pp. 601–640.

[7] A. Berry, J.P. Bordat, and O. Cogis, "Generating All the Minimal Separators of a Graph," Proc. WG, 1999, pp. 167-172.

[8] T. Kloks and D. Kratsch, "Listing All Minimal Separators of a Graph," SIAM Journal on Computing, 1998, pp. 605–613.

[9] E. Balas and C.C. de Souza, "The vertex separator problem: a polyhedral investigation, " Mathematical Programming, 2005, pp. 583–608.

[10] M. Didi Biha and M. Marie-Jean, "An exact algorithm for solving the vertex separator problem," Journal of Global Optimization, 2011, pp. 425–434.

[11] T.N. Bui and C. Jones, "Finding good approximate vertex and edge partitions is NP-hard", Inf. Process. Lett. 42, 1992, pp.153-159.