A Multilevel Vertex Separator Algorithm Based on the Solution of Bilinear Programs

William W. Hager* James T. Hungerford† Ilya Safro‡

August 22, 2016

Abstract

We consider the vertex separator problem on a graph: find a set of vertices of minimum cost whose removal disconnects the graph into two roughly equal sized components. In this talk, we present a multilevel algorithm for the VSP, whose refinement phase is based on the solution of a bilinear program which is shown to approximate the VSP at each level in the multilevel graph hierarchy. We investigate the correspondence between local optimizers in the bilinear program and locally optimal separators in the VSP. In addition, two techniques are developed for escaping from either stationary points or local optima. These techniques are based on making small perturbations in the problem parameters in order to deliberately violate the first-order optimality conditions of the bilinear program. Finally, we investigate the effectiveness of the refinement procedure both in isolation and in the context of the multilevel algorithm, providing computational results on a set of medium to large scale graphs.

Let $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ be a simple, undirected graph on vertex set $\mathcal{V} = \{1, 2, \ldots, n\}$ and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Let $c_i \in \mathbb{R}$ denote the cost of vertex i and let $w_i > 0$ denote its weight. If $\mathcal{X} \subseteq \mathcal{V}$, let $\mathcal{W}(\mathcal{X}) = \sum_{i \in \mathcal{X}} w_i$ and $\mathcal{C}(\mathcal{X}) = \sum_{i \in \mathcal{X}} c_i$.

The Vertex Separator Problem (VSP) on $\mathcal{G}$ is to find a partition of $\mathcal{V}$ into three sets $\mathcal{A}$, $\mathcal{S}$, and $\mathcal{B}$ such that there are no edges between $\mathcal{A}$ and $\mathcal{B}$ (that is, $(\mathcal{A} \cap \mathcal{B}) \cap \mathcal{E} = \emptyset$), $\mathcal{W}(\mathcal{A})$ and $\mathcal{W}(\mathcal{B})$ lie within specified ranges, and $\mathcal{C}(\mathcal{S})$ is as small as possible. Identifying low-cost vertex separators is important in several applications, including sparse matrix factorizations (see [4, Sect. 7.6], [7, Chapter 8], and [13]), hypergraph partitioning [9], and network security [3, 10, 12].

*hager@ufl.edu, http://people.clas.ufl.edu/hager/ PO Box 118105, Department of Mathematics, University of Florida, Gainesville, FL 32611-8105. Phone (352) 294-2308. Fax (352) 392-8357.

†jamesthungerford@gmail.com, M.A.I.O.R. – Management, Artificial Intelligence, and Operations Research, Srl., Lucca, ITALY

‡isafro@clemson.edu, http://www.cs.clemson.edu/∼isafro, 228 McAdams Hall, School of Computing, Clemson University, Clemson, SC 29634. Phone (864) 656-0637.

A combinatorial formulation of the VSP is as follows:

$$\begin{aligned} \min_{\mathcal{A},\mathcal{S},\mathcal{B}\subseteq\mathcal{V}} \quad & \mathcal{C}(\mathcal{S}) \\ \text{subject to} \quad & \mathcal{S} = \mathcal{V} \setminus (\mathcal{A}\cup\mathcal{B}), \quad \mathcal{A}\cap\mathcal{B} = \emptyset, \quad (\mathcal{A}\times\mathcal{B})\cap\mathcal{E} = \emptyset, \\ & \ell_a \le \mathcal{W}(\mathcal{A}) \le u_a, \ \text{ and } \ \ell_b \le \mathcal{W}(\mathcal{B}) \le u_b, \end{aligned} \tag{1}$$

where ℓ_a, ℓ_b, u_a, and u_b are given non-negative real numbers less than or equal to $\mathcal{W}(\mathcal{V})$. The VSP (1) is NP-hard [2, 6]. However, due to its practical importance, many heuristics have been developed for finding low-cost separators, including node-swapping heuristics [11], spectral methods [13], semidefinite programming methods [5], and recently a breakout local search algorithm [1].

In the case where the vertex weights w_i are identically one, the authors in [8] found conditions under which (1) is equivalent to the following continuous bilinear program:

$$\begin{aligned} \max_{\mathbf{x},\mathbf{y}\in\mathbb{R}^n} \quad & \mathbf{c}^{\mathsf{T}}(\mathbf{x}+\mathbf{y}) - \gamma\mathbf{x}^{\mathsf{T}}(\mathbf{A}+\mathbf{I})\mathbf{y} \\ \text{subject to} \quad & \ell_a \le \mathbf{w}^{\mathsf{T}}\mathbf{x} \le u_a, \ \ \ell_b \le \mathbf{w}^{\mathsf{T}}\mathbf{y} \le u_b, \\ & \mathbf{0} \le \mathbf{x} \le \mathbf{1}, \ \text{ and } \ \mathbf{0} \le \mathbf{y} \le \mathbf{1}. \end{aligned} \tag{2}$$

Here, the decision variables $\mathbf{x}$ and $\mathbf{y}$ represent continuous extensions of the incidence vectors for the disconnected components $\mathcal{A}$ and $\mathcal{B}$, $\mathbf{A}$ is the binary adjacency matrix for $\mathcal{G}$, and γ is a penalty parameter satisfying $\gamma \ge \max\{c_i : i \in \mathcal{V}\}$ which enforces the separation condition $(\mathcal{A}\times\mathcal{B})\cap\mathcal{E} = \emptyset$.

Preliminary numerical experiments in [8] showed that applying a continuous optimization algorithm to the bilinear program can be an effective means of making local improvements to vertex separators in multilevel algorithms for large scale graphs. However, at the coarser levels of a multilevel algorithm, vertices typically have non unit weights, since they represent aggregates of vertices from the original graph; moreover, the aggregates often vary widely in size. Hence, it is necessary to understand the relationship between the bilinear program of [8] and (1) in the more general case where vertex weights are arbitrary positive real numbers.

In this talk, we show that in the general case where $w_i > 0$, the bilinear program (2), while not equivalent to the VSP, still approximates it in some sense. In addition, two techniques are presented for escaping from either stationary points or local optima encountered during the refinement phase in order to encourage a wide exploration of the solution space before stopping. These techniques are based on perturbing the problem parameters $\mathbf{c}$ and γ in order to deliberately violate the first-order optimality conditions. At the end of the talk, we conduct a numerical investigation, comparing the performance of our refinement method with refinements based on node swaps. We look at the performance of the refinement method both in isolation and in the context of a multilevel algorithm. Experiments are made on a large benchmark set of medium to large scale graphs from the University of Florida Sparse Matrix Library, the Konect Database, and the Stanford SNAP collection.

References

[1] U. Benlic and J. Hao, *Breakout local search for the vertex separator problem*, in Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, 2013.

[2] T. Bui and C. Jones, *Finding good approximate vertex and edge partitions is NP-hard*, Information Processing Letters, 42 (1992), pp. 153–159.

[3] M. Burmester, Y. Desmedt, and Y. Wang, *Using approximation hardness to achieve dependable computation*, in Randomization and Approximation Techniques in Computer Science, M. Luby, J. Rolim, and M. Serna, eds., vol. 1518 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1998, pp. 172–186.

[4] T. A. Davis, *Direct Methods for Sparse Linear Systems*, SIAM, Philadelphia, PA, 2006.

[5] U. Feige, M. Hajiaghayi, and J. Lee, *Improved approximation algorithms for vertex separators*, SIAM J. Comput., 38 (2008), pp. 629–657.

[6] J. Fukuyama, *NP-completeness of the planar separator problems*, Journal of Graph Algorithms and Applications, 10, No. 2 (2006), pp. 317–328.

[7] A. George and J. W. H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[8] W. W. Hager and J. T. Hungerford, *Continuous quadratic programming formulations of optimization problems on graphs*, European J. Oper. Res., 240 (2015), pp. 328–337.

[9] E. Kayaaslan, A. Pinar, U. Catalyürek, and C. Aykanat, *Partitioning hypergraphs in scientific computing applications through vertex separators*, SIAM J. Sci. Comput., 34(2) (2012), pp. A970–A992.

[10] D. Kim, G. R. Frye, S.-S. Kwon, H. J. Chang, and A. O. Tokuta, *On combinatoric approach to circumvent internet censorship using decoy routers*, in Military Communications Conference, 2013. MILCOM 2013., IEEE, 2013, pp. 1–6.

[11] C. Leiserson and J. Lewis, *Orderings for parallel sparse symmetric factorization*, in Third SIAM Conference on Parallel Processing for Scientific Computing, SIAM Publications, 1987, pp. 27–31.

[12] C. Litsas, A. Pagourtzis, G. Panagiotakos, and D. Sakavalas, *On the resilience and uniqueness of CPA for secure broadcast*, IACR Cryptology ePrint Archive, (2013).

[13] A. Pothen, H. D. Simon, and K. Liou, *Partitioning sparse matrices with eigenvectors of graphs*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 430–452.