A Study of Graph Bisection by Positive Semidefinite Programming

Bo Yu
Department of Combinatorics and Optimization
University of Waterloo, Waterloo, Ontario
Canada, N2L 3G1

Abstract

The graph bisection problem is to partition the vertex set of a graph into two equal-size pieces S and $\bar{S}$ so that the total weight of edges with exact one end in S is minimum. We present a positive semidefinite programming formulation for the problem and some computational testing results of the formulation.

1 Introduction

Given an edge-weighted undirected graph with an even number of vertices, *the graph bisection problem* is to partition the vertex set of the graph into two equal-size pieces S and $\bar{S}$ so that the total weight of edges with exact one end in S is minimum. The set of edges between the two equal-size pieces S and $\bar{S}$ is called a *bisector*.

The graph bisection problem has many applications in VLSI design [1], sparse matrix computations and solving various graph problems. Unfortunately, the problem is NP-hard [3], and people turn to developing heuristic

methods, or considering special classes of graphs in which the problem can be solved.

This project firstly formulate a positive semidefinite programming (SDP) relaxation for the graph bisection problem. Hence the optimal value of the SDP gives us an lower bound for the problem. Secondly we do some computational testing on a few randomly generated examples. We use the eigenvector corresponding to the maximum eigenvalue of the optimal solution of the SDP to get a feasible bisector, which serves as an upperbound for the minimum bisector. The gap, defined by the difference between the lowerbound and the upperbound divided by the lowerbound, lies between zero and twelve percentage in our examples. We use a matlab program, written by Rendl and Wolkowicz using the interior point method developed by Helmberg et al [4], to do the computation.

2 SDP Formulation

In this section we formulate a positive semidefinite programming for the graph bisection problem using Lagrangian multiplier.

Given an undirected graph $G = (V, E)$ with $V = \{1, \ldots, n\}$, n is an even number, and non-negative weights $w_{ij} = w_{ji}$ on the edges $(i, j) \in E$, the weight of a bisector $(S, \bar{S})$ is denoted by $w(S, \bar{S}) = \sum_{i \in S, j \notin S} w_{ij}$.

We first formulate an integer programming for our problem. Let $y_i = 1$ for $i \in S$ and $y_i = -1$ for $i \in \bar{S}$. So $w(S, \bar{S}) = \frac{1}{2} \sum_{i<j} w_{ij}(1 - y_i y_j) =$

$\frac{1}{4}\sum_{i,j} w_{ij}(1 - y_i y_j)$, where $w_{ii} = 0$. Therefore the weight of the minimum bisector is given by the following integer quadratic program:

$$(IP1)\begin{cases} \text{Minmize } \frac{1}{4}\sum_{i,j} w_{ij}(1 - y_i y_j) \\ \text{subject to} \\ \ \sum_i y_i = 0 \\ y_i \in \{-1, 1\}, \text{ for all } i \in V. \end{cases}$$

Equivalently, we have the following integer quadratic program:

$$(IP2)\begin{cases} \text{Minimize } \frac{1}{4}(\sum_{i,j} w_{ij}) - \ \frac{1}{4}(y^t W y) \\ \text{subject to} \\ (\ \sum_i y_i)^2 = 0 \\ y_i^2 = 1, \text{ for all } i \in V, \end{cases}$$

where $y = (y_1, y_2, \ldots, y_n)^t$ and W is the $n \times n$ matrix with w_{ij} as its (i, j) entry.

Now in order to solve $(IP2)$, we only need to consider solving the following program:

$$(IP3)\begin{cases} \mu^* = \text{Maximize } y^t W y \\ \text{subject to} \\ (\ \sum_i y_i)^2 = 0 \\ y_i^2 = 1, \text{ for all } i \in V. \end{cases}$$

Since solving an integer quadratic program is NP-complete in general, we consider its relaxation. Before giving an equivalent formulation for $(IP3)$, we introduce some standard notations. Let $A = (a_{ij})_{n\times n}$ be a matrix of size $n \times n$ and $u = (u_1, u_2, \ldots, u_n)^t$ be a column vector. Diag(u) denotes the diagonal matrix of size $n \times n$ with diagonal elements $u_1, u_2, \ldots, u_n$ and diag$(A) = (a_{11}, a_{22}, \ldots, a_{nn})^t$. J is the square matrix with all elements ones. e is the column vector with all ones. I_n is the identity matrix of size n.

Let μ^* be the optimal value of $(IP3)$. Then equivalently, using Lagrangian multipliers, we have

$$\mu^* = \max_y \min_{\alpha,\beta} L(y,\alpha,\beta) = \max_y \min_{\alpha,\beta} \{y^t W y - \sum_{i=1}^{n} \alpha_i (y_i^2 - 1) - \beta (\sum_i y_i)^2\}$$

$$= \min_{\alpha,\beta} \max_y \{y^t W y - \sum_{i=1}^{n} \alpha_i (y_i^2 - 1) - \beta (\sum_i y_i)^2\}, \qquad \text{(by strong duality)}$$

$$= \min_{\alpha,\beta} \max_y \{\sum_{i=1}^{n} \alpha_i + y^t (W - \mathrm{Diag}(\alpha) - \beta J) y\}.$$

Therefore μ^* is equal to the optimal value of the following positive semidefinite programming using the hidden constraint of above expression.

$$(DIP3) \begin{cases} \mu^* = \text{ Minimize } \sum_{i=1}^{n} \alpha_i \\ \text{subject to} \\ \mathrm{Diag}(\alpha) + \beta J \succeq W, \end{cases}$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)^t$.

Dual of $(DIP3)$ is

$$(P4) \begin{cases} \mu^* = \text{ Maximize } tr(WX) \\ \text{subject to} \\ \mathrm{diag}(X) = e \\ tr(XJ) = 0 \qquad\qquad (*) \\ X \succeq 0. \end{cases}$$

If the rank of X in $(P4)$ is one, then $X = yy^t$ for some non-zero vector y since X is positive semidefinite. Furthermore if X is substituted by yy^t in $(P4)$, we'll get $(IP3)$. This indicates that $(P4)$ is indeed a positive semidefinite programming relaxation of our problem.

3 Computational study

In this section we run a matlab program, designed by Rendl and Wolkowicz using the interior point method developed by Helmberg et al [4], to several randomly generated examples to find the optimal solution of $(P4)$. Moreover we convert the optimal solution into a bisector, which gives us an upperbound of the minimum bisector. In the end, we compare the upperbound and the lowerbound given by the optimal value of $(P4)$. Namely, we consider the gap, defined by the difference between the lowerbound and the upperbound divided by the lowerbound. The gap is between zero and twelve percentage in our examples. Most of the examples has zero gap.

Due to the constraint $(*)$ in $(P4)$, the rank of X must be less than n. Therefore $(P4)$ has no interior point. In order to use the interior point method to solve $(P4)$, we have to consider the feasible region of $(P4)$ in a lower dimensional space.

We look at the $(n-1)$-dimensional vector space which is orthogonal to e. It is easy to check that the following $n-1$ vectors form a basis of the $(n-1)$-dimensional space.

$v_1 = (-\frac{1}{\sqrt{n}}, 1-\frac{1}{n+\sqrt{n}}, -\frac{1}{n+\sqrt{n}}, \ldots, -\frac{1}{n+\sqrt{n}})^t,$

$v_2 = (-\frac{1}{\sqrt{n}}, -\frac{1}{n+\sqrt{n}}, 1-\frac{1}{n+\sqrt{n}}, -\frac{1}{n+\sqrt{n}}, \ldots, -\frac{1}{n+\sqrt{n}})^t, \ldots\ldots,$

$v_{n-1} = (-\frac{1}{\sqrt{n}}, -\frac{1}{n+\sqrt{n}}, \ldots, -\frac{1}{n+\sqrt{n}}, 1-\frac{1}{n+\sqrt{n}})^t.$

Also v_i's are mutually orthogonal and let $V = (v_1, v_2, \ldots, v_{n-1})$.

Furthermore, any such matrix X in $(P4)$ can be written as VYV^t where Y

is a positive semidefinite matrix of size $n-1$. (Since $X = UDiag(\lambda_1, \lambda_2, \ldots, \lambda_{n-1}, 0)U^t$ where U is formed by eigenvectors of X which are mutually orthogonal. $= (U_1, e)Diag(\lambda_1, \lambda_2, \ldots, \lambda_{n-1}, 0)(U_1, e)^t = U_1 Diag(\lambda_1, \lambda_2, \ldots, \lambda_{n-1})U_1^t$. Since U_1 can be written as VB for some matrix B since columns of U_1 are orthogonal to e, $X = VBDiag(\lambda_1, \lambda_2, \ldots, \lambda_{n-1})B^tV^t$ and $Y = BDiag(\lambda_1, \lambda_2, \ldots, \lambda_{n-1})B^t$.)

Therefore $(P4)$ can be rewritten as follows:

$$(P5) \begin{cases} \text{Maximize } tr(WVYV^t) \\ \text{subject to} \\ \text{diag}(VYV^t) = e \\ Y \succeq 0 \end{cases}$$

The dual of $(P5)$ is

$$(DP5) \begin{cases} \text{Minimize } u^t e \\ \text{subject to} \\ V^t(\text{Diag}(u) - W)V \succeq 0. \end{cases}$$

Now we can use the interior point method developed by Helmberg et al [4]to solve $(P5)$. The initial interior point for (P5) is $Y = \frac{n^2+2n\sqrt{n}}{n^2+2n\sqrt{n}-n-2\sqrt{n}} I_{n-1}$. The initial interior point for (DP5) can be $u = Ne$ for a sufficient large integer N. Here we pick $N = n$.

The following is our algorithm:

Algorithm:

Given an edge weighted graph G, compute a lowerbound and an upperbound of the minimum bisector.

(1) Solve $(P5)$ using interior point method and get an optimal solution Y and the optimal value μ^*. The lowerbound is given by $\frac{1}{4}\sum_{i,j} w_{ij} - \frac{1}{4}\mu^*$ from $(IP2)$.

(2) Let $X = VYV^t$. Solve for the eigenvector v corresponding to the maximum eigenvalue of X. Find the bisector $(S, \bar{S})$ by choosing the indices, corresponding to the $n/2$ largest components of v, as S. The weight of $(S, \bar{S})$ gives an upperbound.

(4) Compute the gap $= \frac{\text{upperbound - lowerbound}}{\text{lowerbound}}$.

Acknowledgement The author would like to thank Dr. Wolkowicz and Qing Zhao for lots of helpful discussions.

References

[1] S. Bhatt and T. Leighton, A framework for solving VLSI graph problems, *Journal of Computer and System Sciences* 28:2 (1984) 300-343.

[2] J. Falkner, F. Rendl and H. Wolkowicz, A computational study of graph partitioning, *Technical Report*, Faculty of Mathematics, University of Waterloo, CORR 92-25, August 1992.

[3] M.R. Garey, D.S. Johnson and L. Stockmeyer, Some simplified NP-complete graph problems, *Theoretical Computer Science* 1 (1976) 237-267.

[4] C. Helmberg, F. Rendl, R.J. Vanderbei and H. Wolkowicz, *Technical Report*, Faculty of Mathematics, University of Waterloo, SOR-93-15.