A Spectral Approach to Graphical Representation of Data

Drago Bokal
Department of Mathematics,
IMFM, Jadranska 19, Ljubljana, Slovenia
drago.bokal@imfm.uni-lj.si

Martin Juvan
Department of Mathematics,
University of Ljubljana, Jadranska 19, Ljubljana, Slovenia
martin.juvan@fmf.uni-lj.si

Bojan Mohar
Department of Mathematics,
University of Ljubljana, Jadranska 19, Ljubljana, Slovenia
bojan.mohar@uni-lj.si

Keywords: Data presentation, Clustering, Graph drawing, Spectral method, Gradient method

Received: February 6, 2004

Graphical representation of relationship data is useful in several applications. Relationships among objects are modeled as a graph and the strength of relationship as weights on graph's edges. In the paper we demonstrate how the spectral method can be applied to visualize such data. Application of gradient method is suggested to fine tune the solution obtained by the spectral method.

Povzetek: članek opisuje spektralno metodo za vizualizacijo podatkov.

1 Introduction

Many applications require graphical representation of (non numerical) data. A general approach for such a task is presented. Given a data base, pairs of objects from the given data set are classified as being "close" or "far apart" by specifying a numerical value $S(x, y)$ for each such pair x, y. This value measures *similarity* of objects. This means that the value $S(x, y)$ is large for closely related objects x and y and small for very different objects. For the purpose of this paper we shall assume that S is symmetric, i.e.

$$S(x, y) = S(y, x).$$

Similarity measures can be produced in a number of ways, for example by applying the factor analysis. The goal is to represent the objects graphically (e.g. on a computer screen) in such a way that objects which are similar with respect to the similarity measure S are represented close to each other, i.e., the distance between their graphical representations is in agreement with the "similarity distance" $S(x, y)$.

Two main points of our approach rely on the spectral method where calculations are based on the eigenvalues and eigenvectors of related Laplacian matrices. General setting for this approach is surveyed by Mohar and Poljak [27]. Usually, the spectral approach can be expressed, via the Rayleigh quotient expressions for eigenvalues, as a quadratic optimization and, more generally, via semidefinite programming [1, 18, 30, 32].

Applications of eigenvalue methods in combinatorics, graph theory and in combinatorial optimization have a long history. For example, eigenvalue bounds on the chromatic number were formulated by Wilf [31] and Hoffman [23] in the 1960's. Another early application, in the area of graph partition, is due to Fiedler [15] and Donath and Hoffman [13].

An important result was the use of eigenvalues in the construction of superconcentrators and expanders by Alon and Milman [2, 3]. Isoperimetric properties of graphs and their eigenvalues play a crucial role in the design of various randomized algorithms. These applications are based on the so-called rapidly mixing Markov chains.

There is an increasing interest in the application of eigenvalues to combinatorial optimization problems. For example, Burkard, Finke, Rendl and Wolkowicz [14, 29] used an eigenvalue approach in the study of the quadratic assignment problem and general graph partition problems, Delorme and Poljak [10, 11] and Goemans and Williamson [19] in the max-cut problem, and Juvan and Mohar [24, 25] in labelling problems. Spectral partitioning, which is based on eigenvectors of Laplace eigenvalues of graphs, has proved to be a successful heuristic approach in the design of partition algorithms [7, 22, 21], in solving sparse linear systems [28], clustering [20, 6], ranking [25, 21], in graph drawing [16], automated finding of large components [12], image and video segmentation [17], etc. We refer to [27] for additional applications.

For further results, the reader may consult existing books

and survey papers, such as [8, 9, 27].

2 Problem description

Formally, the above problem can be stated as follows. Given a data set containing n objects and the similarity measure $S(x, y)$ between pairs of objects, find an embedding of the n points in the plane such that the distances between the points representing similar objects are small, while the points corresponding to objects with small similarity are far from each other. In order to formally describe the "agreement" of the similarity measure with the distances in the plane, one has to introduce a function $f : \mathbb{R} \to \mathbb{R}$ which transforms every value s of similarity into the desired distance between points representing two objects whose similarity is $S(x, y) = s$. The transformation f must satisfy the following requirements:

(F1) Monotonicity: if $s \leq s'$, then $f(s) \geq f(s')$.

(F2) Validity: $\lim_{s\to\infty} f(s) = 0$.

It can be proved easily that **(F1)** and **(F2)** together imply:

(F3) Non-negativity: $f(s) \geq 0$ for every $s \in \mathbb{R}$.

Exact choice of f depends on the data to be represented and on the properties of their similarity measure S. If similarities of distinct objects are always positive, then one may take, for example, $f(s) = 1/s$.

We consider the objects as vertices of the (complete) weighted graph G whose edge-weights are determined by S and f: the weight of the edge xy is equal to $f(S(x, y))$. This setting has an advantage that in case when similarity measure of certain pairs of objects is not defined, then the edges corresponding to such pairs can be removed from the graph.

We consider the following problem. Given is a graph $G = (V, E)$ and a weight function $w : E \to \mathbb{R}^+$ (the edge-weights). The goal is to find a mapping $\phi : V \to \mathbb{R}^2$, which assigns to every vertex of G a point in the Euclidean plane, such that the distance between the points $\phi(x)$ and $\phi(y)$ is as close as possible to the prescribed weight $w(xy)$. We refer to this problem as the *vertex placement problem*.

The vertex placement problem is an optimization problem and there are several possible choices for the energy function which is to be minimized. Our choice is described in Subsection 3.2.

3 Solving the problem

We propose the following general algorithm to find (an approximate) solution to the vertex placement problem.

Algorithm 1: Basic algorithm for solving the vertex placement problem

Input: Graph $G = (V, E)$, similarity measure $S : V \times V \to \mathbb{R}$.
Output: Placement of the vertices of G into $\mathbb{R}^2$.

Description:
Compute the edge-weights of G, $w(xy) = f(S(x, y))$, $xy \in E$.
Obtain the initial placement by the spectral method.
Run the gradient method to obtain an improved placement.
Correct the final solution.

3.1 Initial placement

To find the initial placement in Algorithm 1, the spectral method is proposed. Its formal description is given as Algorithm 2. Let us observe that this algorithm is only heuristic, and there are no theoretical guarantees that it will return a solution close to an optimum. However, as mentioned in the introduction, it behaves quite well in practice. We refer to [27] for more information.

Algorithm 2: Obtaining the initial placement of vertices

Input: Weighted graph $G = (V, E)$ with edge-weights w.
Output: Initial placement of the vertices of G in $\mathbb{R}^2$.

Description:
Compute the auxiliary matrix A from the edge-weights w by setting
$(A)_{ij} = 0$, if $i = j$ or $ij \notin E$
$(A)_{ij} = 1/w(ij)$, if $ij \in E$.
Determine the Laplace matrix L_A from A:
$(L_A)_{ii} = \sum_{k=1}^{n} (A)_{ik}$,
$(L_A)_{ij} = -(A)_{ij}$, $i \neq j$.
Compute the eigenvectors e, f of L_A corresponding to the two smallest nontrivial eigenvalues of L_A.
Set $x_i := e_i, y_i := f_i$ as the coordinates of the vertex i.

To obtain the auxiliary matrix A from the edge-weights w, one can also use the following formula:

$$\begin{aligned} (A)_{ij} &= 0, \quad \text{if } i = j \text{ or } ij \notin E \\ (A)_{ij} &= 1/(w(ij))^2, \quad \text{if } ij \in E. \end{aligned}$$

Both alternatives seem to yield good results.

If the number of objects to be represented is too large, we first apply the same spectral method to cluster the data set

into smaller cluster sets whose size fits the requirements. Particular clusters that are small enough can then be graphically represented as described above. On the other hand, the relations among clusters themselves can also be represented by the same method by defining the distance w between two clusters X, Y as

$$w(X,Y) = \frac{1}{|X||Y|} \sum_{x \in X} \sum_{y \in Y} f(S(x,y)).$$

Similar spectral approach has already been applied to clustering problem, see [4, 5].

3.2 The gradient method

The energy function we choose to minimize in solving the vertex placement problem is the sum of the squares of the relative differences between the distances implied by the current placement of the vertices, and the desired distances:

$$\mathcal{E}(\mathbf{x},\mathbf{y}) = \sum_{ij \in E} \left(\frac{w(ij) - \|(x_i,y_i) - (x_j,y_j)\|}{w(ij)} \right)^2,$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, $\mathbf{y} = (y_1, y_2, \ldots, y_n)$, (x_i, y_i) is the point in $\mathbb{R}^2$ corresponding to the vertex i, $w(ij)$ is the desired distance between vertices i and j, and E is the set of edges of the graph G on vertices $\{1, \ldots, n\}$.

The gradient method is a well-known iterative algorithm for solving optimization problems whose objective function is differentiable (see, e.g., [26]).

At each step, first the direction in which the placement will change is determined: usually, the negative gradient is taken as the direction, but in every third step the average of the last two gradients is used instead (this significantly reduces the "zig-zag" behavior which otherwise often occurs). Then the length of the step in the chosen direction is calculated. As the first approximation a step of the Newton's root finding method for the direction derivative of the energy function in the chosen direction is used (the goal is a local minimum and the derivative evaluates to zero in the minimum; the Newton's method is used to find this root). Then step of the calculated length is made in the direction of decreasing energy function. If the value of the energy function in the obtained point is higher than the current value, the length of the step is corrected: it is repeatedly multiplied by an appropriately chosen constant factor from the unit interval until the value of the energy function is lower than the current value. Additionally, if two subsequent directions differ too much (the measure is the angle between the two), the next step is to be shorter. This method is a variant of an inexact line search using the Armijo Rule as its stopping condition (cf. [26, sec. 3.2]) and combined with some heuristics.

The method stops when any of the following cases occurs:

- the norm of the direction vector is small enough,
- the number of steps exceeds maximum number allowed, or
- a certain number of the quotients of subsequent energy values are small enough.

These criteria can be tuned to achieve either higher accuracy or faster performance.

Note that the problem is invariant under translations and rotations of the plane. Thus we may fix the position of one vertex and additionally one coordinate of another vertex.

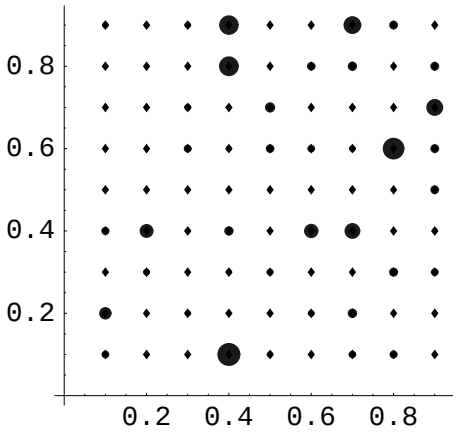

Figure 1: Performance test for random graphs with parameters $p = 0.1$, q on the x-axis and r on the y-axis ranging from 0.1 to 0.9.

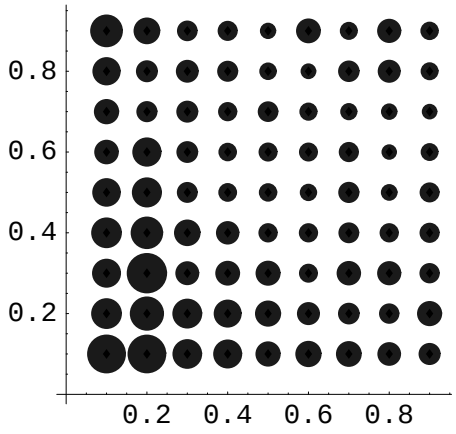

Figure 2: Performance test for random graphs ($p = 0.8$).

4 Practical considerations

The behavior of proposed algorithms was tested on several distance matrices of various sizes. For these tests we used random graphs constructed as follows: let $G = G(n,p)$ be a random graph on n vertices, where each edge is added to G with probability p. Choose randomly n points $x_1, \ldots, x_n$ in the plane, and for each edge $ij \in E_G$ let

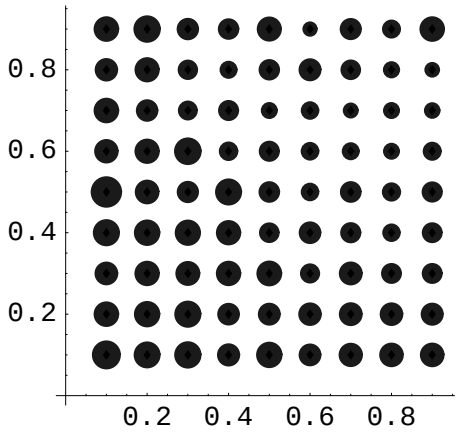

Figure 3: Performance test for random graphs ($p = 1$).

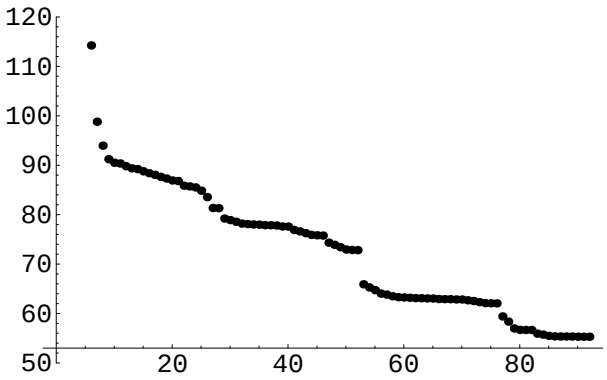

Figure 4: Decrease of the energy during iteration of the gradient method.

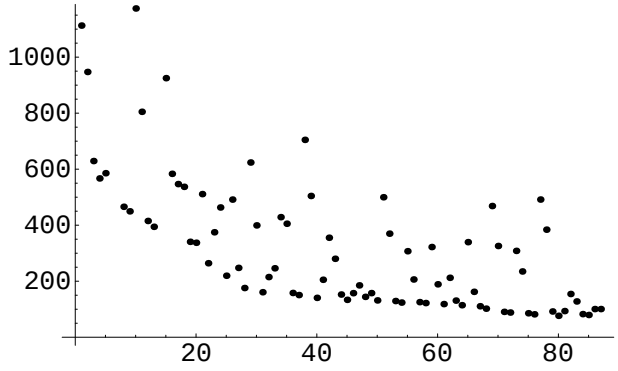

Figure 5: Changes in gradient norm during iteration of the gradient method.

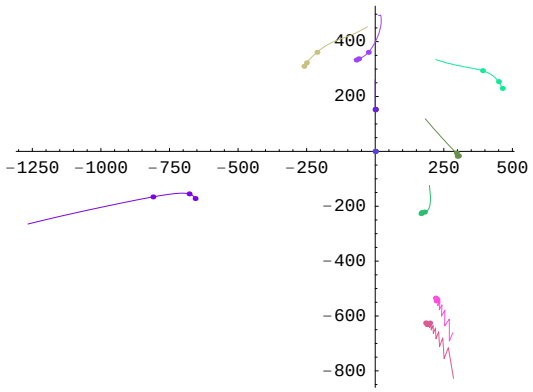

Figure 6: Traces of positions of vertices in the plane during the execution of the gradient method, initial placement using eigenvectors of smallest eigenvalues

$(D)_{ij}$ be the distance between the points x_i, x_j. With probability q the distances $(D)_{ij}$ are perturbed for a factor r (half of the distances are increased, half are decreased). Additional tests were performed using random bipartite graphs, obtained in a similar way. Such graphs appear often in the real world applications, where objects considered naturally fall into two disjoint classes.

Figures 1–3 demonstrate the results of the tests for parameters $n = 10$, q and r ranging from 0 to 0.9 by steps 0.1. The thicker the point, the more steps the gradient method required. The points were obtained averaging the results of ten independently chosen random graphs with the same parameters.

The performance analysis showed that the gradient method requires the largest number of steps when p is large with q and r being small (compare the aforementioned figures). The interpretation could be that then the optimum solution is nearly exact and hard to find, as the graph is dense. With large perturbations, the search space tends to have more local optima that are close to the initial state and the algorithm is more easily stopped in one of them. When the graph is sparse, the problem usually splits into several smaller subproblems and in general requires fewer steps of the gradient method.

Figure 4 presents the energy of the solution as a function of the number of iterations performed. It is decreasing in steps, which demonstrates that the solution may be jumping from one local optimum to another at certain points in time.

Norm of the gradient calculated using the gradient method is displayed in Figure 5. It is demonstrated that the norm of the gradient is not decreasing monotonously, however towards the local optimum its value settles and approaches 0. The large jumps in the gradient norm correspond to the bigger decreases in the value of the energy function.

The progress of the algorithm was monitored using the traces of the points in the plane. After every step of the gradient method the graph was output and its position in the plane was displayed. The positions of the same vertex in two consecutive drawings were connected by a line segment. Thus for each point we reconstructed its trace during the optimization process. Figures 6–8 display one such picture for a graph on ten vertices. These traces were displayed for various initial placements of vertices of considered graphs. Results demonstrate that the traces are shortest if we choose as the initial placement the one proposed

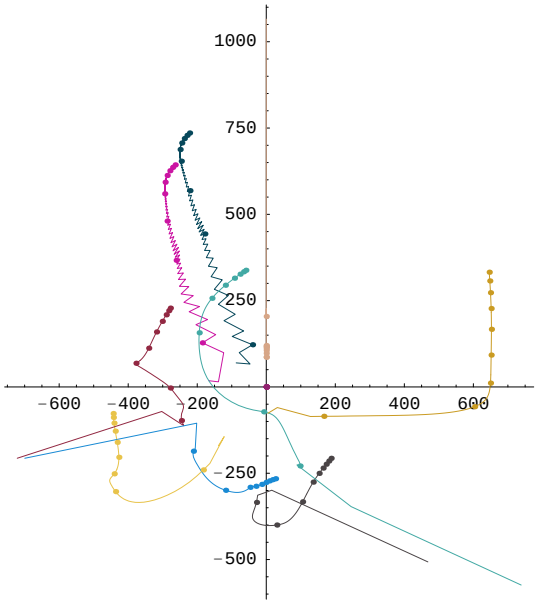

Figure 7: Traces of positions of vertices, initial placement using eigenvectors of larger eigenvalues

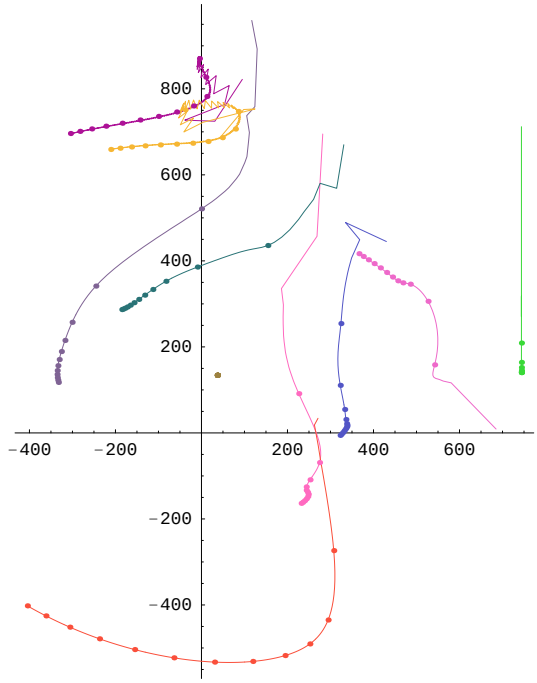

Figure 8: Traces of positions of vertices, random initial placement

by Algorithm 2 (Figure 6). This was tested both against choosing higher eigenvalues (Figure 7) and against choosing a random initial placement (Figure 8). Note that the position of the first vertex and one coordinate of the second vertex are fixed, reducing the unnecessary computational costs due to isometric transformations of the plane.

In Figure 7 certain “zig-zag" behavior of two points can be observed. It hindered the gradient method from finding the optimal solution fast. To reduce the effect of such a behavior, we slightly modified the gradient method such that in every third step, the position of vertices is updated according to the average direction of the last two gradients (see the description of the gradient method).

5 Conclusion

As demonstrated, the spectral method turns out to be well applicable to the problem of graphical representation of relations among objects. In the paper we suggested using gradient method to fine tune the solution obtained by the spectral method, however, other local optimization algorithms could be applied as well, cf. [26]. A comparison of their suitability to the described problem, as well as some rigorous analysis of convergence could be the subject of further research. One could also investigate theoretical bounds of optimality of the solutions proposed by the spectral method.

6 Acknowledgments

The authors acknowledge fruitful discussions about this subject with Robert Reinhardt. Also, programming help of Matija Mazi and Martin Milanič is greatly appreciated. We also thank the referees for their suggestions.

References

[1] F. Alizadeh, Interior point methods in semidefinite programming with applications to combinatorial optimization, SIAM J. Optimiz. 5 (1995) 13–51.

[2] N. Alon, Eigenvalues and expanders, Combinatorica 6 (1986) 83–96.

[3] N. Alon, V. D. Milman, λ_1, isoperimetric inequalities for graphs and superconcentrators, J. Combin. Theory, Ser. B 38 (1985) 73–88.

[4] M. Bolla, Spectra, Euclidean representations and clusterings of hypergraphs, Discrete Math. 117 (1993), 19–39.

[5] M. Bolla, G. Tusnády, Spectra and optimal partitions of weighted graphs, Discrete Math. 128 (1994), 1–20.

[6] P. K. Chan, M. Schlag, J. Zien, Spectral k-way ratio cut partitioning and clustering, Symp. on Integrated Systems, 1993.

[7] T.F. Chan, W.K. Szeto, On the optimality of the median cut spectral bisection graph partitioning method, SIAM J. Scientific Comput. 18 (1997) 943–948.

[8] F. R. K. Chung, Spectral graph theory, American Math. Soc., Providence, RI, 1997.

[9] D. M. Cvetković, M. Doob, H. Sachs, Spectra of graphs, Academic Press, New York, 1979; 3rd edition, Johann Ambrosius Barth Verlag, Heidelberg, 1995.

[10] C. Delorme, S. Poljak, Laplacian eigenvalues and the maximum cut problem, Math. Programming 62 (1993) 557–574.

[11] C. Delorme, S. Poljak, Combinatorial properties and the complexity of a max-cut approximation, Europ. J. Combin. 14 (1993) 313–333.

[12] C. Ding, X. He, and H. Zha, A spectral method to separate disconnected and nearly disconnected web graph components, Proc. 7th ACM Int'l Conf Knowledge Discovery and Data Mining (KDD 2001), 2001, pp. 275–280.

[13] W. E. Donath, A. J. Hoffman, Lower bounds for the partitioning of graphs, IBM J. Res. Develop. 17 (1973) 420–425.

[14] G. Finke, R. E. Burkard and F. Rendl, Quadratic assignment problem, Ann. Discrete Math. 31 (1987) 61-82.

[15] M. Fiedler, Algebraic connectivity of graphs, Czech. Math. J. 23 (98) (1973) 298–305.

[16] P. W. Fowler, T. Pisanski, J. Shawe-Taylor, Molecular graph eigenvectors for molecular coordinates, in "Graph drawing: GD'94," (R. Tamassia, ed.), Springer-Verlag, Berlin, 1995, pp. 282–285.

[17] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, Spectral grouping using the Nyström method, preprint, 2002.

[18] M. Goemans, Semidefinite programming in combinatorial optimization, Math. Program. 79 (1997) 143-161.

[19] M. X. Goemans, D. P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, J. ACM 42 (1995) 1115–1145.

[20] L. Hagen, A. B. Kahng, New spectral methods for ratio cut partitioning and clustering, IEEE Trans. Computer-Aided Design 11 (1992) 1074–1085.

[21] C. Helmberg, B. Mohar, S. Poljak, F. Rendl, A spectral approach to bandwidth and separator problems in graphs, Linear and Multilinear Algebra 39 (1995) 73–90.

[22] B. Hendrickson, R. Leland, An improved spectral graph partitioning algorithm for mapping parallel computations, SIAM J. Sci. Comput. 16 (1995) 452–469.

[23] A. J. Hoffman, On eigenvalues and colorings of graphs, in "Graph Theory and Its Applications" (B. Harris, ed.), Acad. Press, 1970, pp. 79–91.

[24] M. Juvan, B. Mohar, Optimal linear labelings and eigenvalues of graphs, Discrete Appl. Math. 36 (1992) 153–168.

[25] M. Juvan, B. Mohar, Laplace eigenvalues and bandwidth-type invariants of graphs, J. Graph Theory 17 (1993) 393-407.

[26] C. T. Kelley, Iterative Methods for Optimization, SIAM, Philadelphia, 1999.

[27] B. Mohar, S. Poljak, Eigenvalues in combinatorial optimization, in "Combinatorial and Graph-Theoretical Problems in Linear Algebra," R. A. Brualdi, S. Friedland, V. Klee, Eds., IMA Volumes in Mathematics and Its Applications, Vol. 50, Springer-Verlag, 1993, pp. 107–151.

[28] A. Pothen, H. D. Simon, K.P. Liu, Partitioning sparse matrices with eigenvectors of graph, SIAM J. Matrix Anal. Appl. 11 (1990) 430–452.

[29] F. Rendl, H. Wolkowicz, Applications of parametric programming and eigenvalue maximization to the quadratic assignment problem, Math. Progr. 53 (1992) 63-78.

[30] L. Vandenberghe and S. Boyd, Semidefinite programming, SIAM Review 38 (1996) 49-95.

[31] H. S. Wilf, The eigenvalues of a graph and its chromatic number, J. London Math. Soc. 42 (1967) 330–332.

[32] H. Wolkowicz, R. Saigal and L. Vandenberghe (editors), Handbook of semidefinite programming, Theory, algorithms, and applications, International Series in Operations Research & Management Science 27, Kluwer Academic Publishers, Boston, MA, 2000.