A Combinatorial Algorithm for Minimizing the Maximum Laplacian Eigenvalue of Weighted Bipartite Graphs

C. Helmberg, I. Rocha, U. Schwerdtfeger

Preprint 2015-12

Preprintreihe der Fakultät für Mathematik
ISSN 1614-8835

A Combinatorial Algorithm for Minimizing the Maximum Laplacian Eigenvalue of Weighted Bipartite Graphs

Christoph Helmberg[1], Israel Rocha[2] and Uwe Schwerdtfeger[1]

[1]Fakultät für Mathematik, Technische Universität Chemnitz
D-09107 Chemnitz, Germany
{*helmberg,uwe.schwerdtfeger*}*@mathematik.tu-chemnitz.de*
[2]Instituto de Matemática, Universidade Federal do Rio Grande do Sul
CEP 91509-900, Porto Alegre, RS, Brazil
israel.rocha@ufrgs.br

July 16, 2015

Abstract

We give a strongly polynomial time combinatorial algorithm to minimise the largest eigenvalue of the weighted Laplacian of a bipartite graph $G = (W \cup B, E)$. This is accomplished by solving the dual graph embedding problem which arises from a semidefinite programming formulation. In particular, the problem for trees can be solved in time $O(|W \cup B|^3)$.
Keywords: bipartite graph, tree, weighted Laplacian matrix, graph embedding
MSC 2010: 05C05, 05C10, 05C50, 05C85, 90C22, 90C35

1 Introduction

For a simple graph (no loops or multiple edges) $G = (N = \{1, \ldots, n\}, E)$ and a vector of non-negative edge weights $w \in \mathbb{R}^E_+$ we define the weighted Laplacian as the $N \times N$ matrix

$$L_w(G) = \sum_{ij \in E} w_{ij}(e_i - e_j)(e_i - e_j)^T = D_w - A_w$$

where e_i is the i-th canonical basis vector, $D_w = \operatorname{Diag}\left(\sum_{j\colon ij \in E} w_{ij},\, i \in N\right)$ and A_w is the weighted adjacency matrix. The first representation shows in particular that $L_w(G)$ is symmetric and positive semidefinite ($L_w(G) \succeq 0$) and hence has eigenvalues

$$0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n.$$

We refer to the surveys [7, 8] for numerous applications of the Laplacian spectrum. For instance, in [2] Fiedler studied λ_2 subject to the constraint $\sum_{ij \in E} w_{ij} = |E|$. The maximum value attained is known as the absolute algebraic connectivity of G and a corresponding eigenvector is strongly related to partitions of the graph. The optimal λ_2 was also found to be related to mixing rates of Markov chains on G [1, 11]. In [5, 6] the authors studied a related graph embedding problem and found a new minor monotone graph property referred to as rotational dimension. The embedding problem arises as the Lagrangian dual of maximising λ_2, formulated as a semidefinite program (SDP).

Fiedler [3] also studied the problem of minimising the maximal eigenvalue λ_n which we take as a starting point of our investigations. We slightly deviate from the original formulation of the problem in [3] by assuming the edge weights to sum up to 1 rather than $|E|$. It then reads as an SDP

$$\text{minimise } \lambda_n \quad \text{s.t.} \begin{cases} \lambda_n I - \sum_{ij \in E} w_{ij} E_{ij} \succeq 0, \\ \sum_{ij \in E} w_{ij} = 1, \\ w_{ij} \geq 0 \ (ij \in E), \ \lambda_n \in \mathbb{R}. \end{cases} \tag{1.1}$$

Fiedler gives bounds for the optimal λ_n and described properties of the corresponding eigenvectors of the optimally weighted Laplacian. The largest share of [3] is dedicated to bipartite graphs and especially trees. Some of those results were rediscovered in our investigation of the dual problem (1.3) below and new proofs "from the dual point of view" are given.

Since $\lambda_n > 0$ for any feasible solution we can set $\widetilde{w}_{ij} = \frac{w_{ij}}{\lambda_n}$ and obtain an equivalent program

$$\text{maximise } \sum_{ij \in E} \widetilde{w}_{ij} \quad \text{s.t.} \begin{cases} I - \sum_{ij \in E} \widetilde{w}_{ij} E_{ij} \succeq 0, \\ \widetilde{w}_{ij} \geq 0 \ (ij \in E) \end{cases} \tag{1.2}$$

whose Lagrangian dual is the embedding problem

$$\text{minimise } \sum_{i=1}^{n} \|v_i\|^2 \quad \text{s.t.} \begin{cases} \|v_i - v_j\|^2 \geq 1 \ (ij \in E) \\ v_i \in \mathbb{R}^n \ (i \in N). \end{cases} \tag{1.3}$$

we shall mainly work with. In [4, 10] strong duality is shown to hold for (1.2) and (1.3), *i. e.* both are solvable and attain the same optimal value. Geometric properties of optimal embeddings are related to the separator structure of the graph.

Both [3] and [4] are interested in structural properties rather than computational issues. With semidefinite programming (1.2) and (1.3) can be solved efficiently. Moreover, the work [3] and Section 2 below show that for bipartite graphs optimal solutions to (1.1) and (1.3) have a strong combinatorial flavour. This served as our main motivation to devise a *combinatorial* algorithm which solves the problem in polynomial time. Its main ingredient is an efficient method to solve the following problem:

> For a given bipartite graph $G = (W \dot{\cup} B, E)$ compute $S(G)$ which is defined as the (unique) subset $X \subseteq B$ of largest cardinality that minimises the ratio $|N_G(X)|/|X|$. Here $N_G(X)$ denotes the complete neighbourhood of X in G.

In our approach we first give a strongly polynomial time algorithm for trees and extend this to provide a strongly polynomial time algorithm for bipartite graphs. After completing this work it turned out that earlier work of Poljak [9] for minimising the spectral radius of weighted adjacency matrices builds upon solving the same subproblem via network flows; for the bipartite case this is in fact more elegant and faster. Still, our approach relies on new structural properties, it is self contained and faster for trees, so we believe it is worth to include the extension to the bipartite case, as well.

In Section 2 we give an optimality criterion for embeddings and Section 3 provides an algorithm which computes the optimal embedding (viz. decomposition) of a tree in time $O(n^3)$. Section 4 is devoted to computing $S(\cdot)$ with the help of the tree algorithm. Thus, our approach has the additional feature that it outputs a spanning tree of the input graph with the same optimal embedding as G. Our findings are summarised in

Theorem 1.1. *For a bipartite graph $G = (W\dot{\cup}B, E)$ there is an algorithm which solves problems* (1.1)*,* (1.2) *and* (1.3) *in strongly polynomial time. In particular, if G is a tree these problems can be solved in time $O(|W \cup B|^3)$.*

2 Properties of optimal embeddings of bipartite graphs

A well written exposition of the basic properties of optimal embeddings is given in the thesis [10] and propositions 2.1 and 2.3 are mostly special cases of results in [10, Sec. 4.2]. However, some of the proofs simplify for bipartite graphs and for a selfcontained exposition we include them here.

Notation: When we consider bipartite graphs $G = (N(G), E(G))$ we write the vertex set as $N(G) = W(G)\dot{\cup}B(G)$ and consider the edges to be directed from $W(G)$ to $B(G)$, *i. e.* $E(G) \subseteq W(G) \times B(G)$. We occasionally denote the vertices in $W(G)$ as "white" and those in $B(G)$ rather unsurprisingly as "black" vertices. If there is no danger of ambiguity we set $n = |N(G)|$. If $C = (W(C)\dot{\cup}B(C), E(C) \subseteq W(C) \times B(C))$ is another bipartite graph then $C \subseteq G$ means $W(C) \subseteq W(G)$, $B(C) \subseteq B(G)$ and $E(C) \subseteq E(G)$ hold. Accordingly, when we write $G \setminus C$ we mean the subgraph of G obtained by removing all vertices of C from the vertex set of G together with all incident edges. In the same vein, for subgraphs $C, D \subset G$ with disjoint vertex sets we write $C \cup D$ for the induced subgraph $G\,[W(C) \cup B(C) \cup W(D) \cup B(D)]$.

Proposition 2.1. *The embedding problem* (1.3) *is solvable for arbitrary G and if $G = (N = W\dot{\cup}B, E \subseteq W \times B)$ is bipartite, there is a one-dimensional optimal solution $V = (v_1, \ldots, v_n) \in \mathbb{R}^{1\times n}$ with $v_i \le 0$ for $i \in W$ and $v_i \ge 0$ for $i \in B$. Furthermore, any optimal embedding of a bipartite graph lies within the unit ball.*

Proof. The embedding $v_i = e_i$, $i \in N$, is feasible and has objective value n. Therefore, by continuity of the objective function, all optimal solutions are inside the compact region

$$\left\{(v_1, \ldots v_n) \in \mathbb{R}^{n\times n} \colon \|v_i\| \le \sqrt{n}, i \in N \|v_i - v_j\| \ge 1,\, ij \in E\right\}.$$

Outside this region the objective takes values greater than n. For a bipartite graph $G = (N = W\dot{\cup}B, E \subseteq W \times B)$ a one-dimensional optimal embedding can be chosen, namely if $(v_i^*, i \in N)$ is an optimal embedding into $\mathbb{R}^n$ then $W \ni i \mapsto v_i := -\|v_i^*\|$, $B \ni i \mapsto v_i := \|v_i^*\|$ is an optimal embedding in the line: indeed, for $ij \in E$ feasibility is implied by $1 \le \|v_i^* - v_j^*\| \le |\|v_i^*\| + \|v_j^*\|| = |v_i - v_j|$ and the objective values coincide. As for the last assertion, if there is a solution with, say, a black vertex i embedded at v_i^* with $\|v_{i_0}^*\| > 1$ then construct the corresponding one-dimensional embedding as above with all white embedding points ≤ 0. Then $v_{i_0} > 1$ can be replaced by, say $v_{i_0} = 1$ without violating any constraint and thereby improve the objective, a contradiction. □

Definition 2.2. *For $G = (N, E)$ let $V = (v_1, \ldots, v_n)$ be an optimal solution of (1.3) and w_{ij} a corresponding optimal solution of (1.2). The* active subgraph *$G_V = (N, E_V)$ has edge set $E(G_V) = \{ij \in E \colon \|v_i - v_j\| = 1\}$. The* strictly active subgraph *$G_w = (N, E_w)$ of G with respect to w has edge set $E_w = \{ij \in E : w_{ij} > 0\}$.*

By the complementarity conditions of both problems there holds $E_w \subseteq E_V$; the inclusion is in general strict.

Proposition 2.3. *Let $V \in \mathbb{R}^{1\times N}$ be an optimal one-dimensional embedding with $v_i \le 0$ for $i \in W$ and $v_i \ge 0$ for $i \in B$, cf. Prop. 2.1. Let G_V be the active subgraph of G w.r.t. V and $C = (W(C) \cup\ B(C), E(C))$ be a connected component of G_V.*

1. *Then C is embedded as follows*

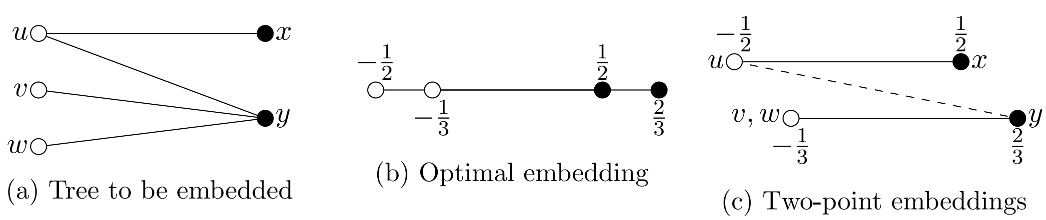
$$\begin{aligned} W(C) \ni i \mapsto & \frac{-|B(C)|}{|B(C)| + |W(C)|} =: y_C \\ B(C) \ni i \mapsto & \frac{|W(C)|}{|B(C)| + |W(C)|} = 1 + y_C. \end{aligned} \tag{2.1}$$

2. *i_0 is isolated in G_V if and only if it is isolated in G. In that case in any optimal embedding $v_{i_0} = 0$.*

3. *All components of V lie in the open interval $]-1,1[$.*

In order to visualise Prop. 2.3 consider the minimal meaningful example of a tree on five nodes (Fig. 1a). Its optimal embedding is displayed in Fig. 1b, the two components of the resulting active subgraph embedded according to (2.1) are displayed in Fig. 1c.

Figure 1



Remarks: By 2 of Prop. 2.3 we can restrict our attention to graphs without isolated vertices. Notice, however, that 2 is in accordance with (2.1). Moreover, for an optimal embedding of a *connected* graph there is by 3 no choice in a one-dimensional embedding but sending the white nodes to one side and the black nodes to the opposite side of the origin. Lastly, we point out that the optimal values of (1.1),(1.2) and (1.3) are rational, cf. [3, Coro. 3.10].

Proof. 1. We first consider a connected component C of G_V with $W(C) \neq \emptyset \neq B(C)$. By definition of G_V the vertices in $W(C)$ are embedded at some nonpositive number y and those in $B(C)$ at $1 + y$. Their contribution to the objective is the quadratic polynomial $|W(C)| \cdot y^2 + |B(C)| \cdot (1+y)^2$ which has its unique minimum at $y_C = \frac{-|B(C)|}{|B(C)|+|W(C)|}$. If $y \neq y_C$, a small perturbation of C's embedding towards y_C would decrease the objective value without changing the active subgraph. This would contradict minimality.
2. Let i_0 be an isolated node in G_V, w.l.o.g. $i_0 \in W$. Then $v_{i_0} = 0$ because otherwise it could be moved towards 0 to obtain a strictly smaller objective value. Now if i_0 had a neighbour $j_0 \in B$ with $i_0 j_0 \in E$ then j_0 would have to be embedded at $v_{j_0} = 1$, where $v_{j_0} \le 1$ is granted by Prop. 2.1. But then i_0 and j_0 are connected in G_V as well.
3. By 2. a vertex i is either isolated ($v_i = 0$) or belongs to a connected component C of G_V which by 1. is embedded at $y_C, 1 + y_C \in]-1,1[$. $\square$

Embedding the entire graph in the two points determined by (2.1) always yields a feasible solution for (1.3), albeit in general not an optimal one, and this two-point embedding will be a convenient starting point.

Definition 2.4. *Let $G = (W(G)\dot{\cup}B(G), E \subseteq W(G) \times B(G))$ be a bipartite graph. Then the embedding of G in y_G and $1 + y_G$ as in* (2.1) *is called the* two-point embedding *of G.*

Remark: G does not have to be connected, neither is it required that both $B(G)$ and $W(G)$ are non-empty ($B(G) = \emptyset \Rightarrow y_G = 0$, $W(G) = \emptyset \Rightarrow y_G = -1$).

Lemma 2.5. *For bipartite graphs C, C' we have*

$$y_C < y_{C'} \Leftrightarrow \frac{|W(C)|}{|B(C)|} < \frac{|W(C')|}{|B(C')|}. \tag{2.2}$$

Here we use the convention that $\frac{a}{0} = \infty$ for $a > 0$ if necessary. Furthermore, if the two-point embeddings of C, C' belong to a feasible embedding of a graph G then

$$y_C < y_{C'} \Rightarrow E(G) \cap \big(W(C') \times B(C)\big) = \emptyset. \tag{2.3}$$

Proof. (2.2) is an easy computation. As for (2.3) assume there is an edge $ij \in W(C') \times B(C)$. Then i would be embedded at $y_{C'}$, j at $1 + y_C$, and $1 + y_C - y_{C'} < 1$ which is not feasible. □

Next we consider partitioning a graph according to a given optimal one-dimensional embedding into maximal subgraphs C_i so that w.r.t. the embedding each subgraph's nodes form a two-point embedding of the subgraph. Such a C_i may comprise several connected components of the active subgraph G_V if each of them has the same two-point embedding. It is convenient to imagine the C_i as being sorted from "left" to "right" in increasing order of their y_{c_i} values. In characterising the leftmost and the rightmost components the neighbourhood structure in G plays an important role.

Definition 2.6. *For a graph $G = (N(G), E(G))$ and a subset $A \subseteq N(G)$ we define by $N_G(A) = \{i \in N(G) \colon \exists\, j \in A \colon ij \in E(G)\}$ the (complete) neighbourhood of A in G.*

Lemma 2.7. *Consider an optimal one-dimensional embedding of a bipartite graph $G = (W(G)\dot{\cup}B(G), E(G) \subseteq W(G) \times B(G))$ as in Prop.2.1. Then the vertex set of the subgraph C of G with the leftmost two-point embedding is of the form $N_G(P) \cup P$ where $P \subseteq B(G)$. For the rightmost two-point-embedding it is of the form $Q \cup N_G(Q)$ where $Q \subseteq W(G)$.*

Proof. For this leftmost C a vertex in $B(C)$ is embedded in $1 + y_C$ and all its neighbours must be embedded in $y \leq y_C$ by feasibility, hence in $y = y_C$. The discussion of the second assertion is analogous. □

Remark: This is in particular true if G contains isolated vertices in which case one point of the corresponding two-point embedding has no vertex mapped to it.

The following simple observation is used throughout the text.

Lemma 2.8. *Let $m \geq 2$ and $a_1, a_2, \ldots, a_m$, $b_1, \ldots, b_m > 0$. Then*

$$\frac{a_1}{b_1} \leq \frac{a_2}{b_2} \leq \ldots \leq \frac{a_m}{b_m} \Rightarrow \frac{a_1}{b_1} \leq \frac{a_1 + a_2 + \ldots + a_m}{b_1 + b_2 + \ldots + b_m} \leq \frac{a_m}{b_m},$$

where the inequalities in the conclusion are strict if and only if at least one inequality in the premise is strict.

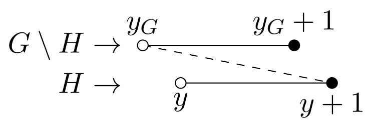
When is a two-point embedding of a graph optimal? We provide an optimality criterion.

Lemma 2.9. *For a bipartite graph $G = (W\dot{\cup}B, E \subseteq W \times B)$ the following are equivalent:*

1. *The two-point embedding $W \to \{y_G\}$ $B \to \{1+y_G\}$ of a bipartite graph $G = (W\dot{\cup}B, E)$ is optimal.*
2. $\forall Q \subseteq W,\ Q \neq \emptyset$: $\frac{|Q|}{|N_G(Q)|} \leq \frac{|W|}{|B|}$.
3. $\forall P \subseteq B,\ P \neq \emptyset$: $\frac{|N_G(P)|}{|P|} \geq \frac{|W|}{|B|}$.

Here we use the convention that $\frac{a}{0} = \infty$ for $a > 0$ if necessary.

Proof. 1. $\Rightarrow$ 2.: Assume there is $Q \subseteq W$ with $\frac{|Q|}{|N_G(Q)|} > \frac{|W|}{|B|}$ and let $H = G\left[N_G(Q) \cup Q\right]$. Due to this particular form of H, for every $y > y_G$ the embedding of G where $Q \to \{y\}$, $N_G(Q) \to \{1 + y\}$, $W \setminus Q \to \{y_G\}$ and $B \setminus N_G(Q) \to \{1 + y_G\}$ is also feasible.



H's contribution to the objective value is $|Q| \cdot y^2 + |N_G(Q)| \cdot (1 + y)^2$ which is minimal at $y = y_H$ and $y_H > y_G$ by assumption and (2.2). Therefore the objective value associated with the two-point embedding of G can be decreased by perturbing the embedding points of H slightly to the right and hence the two-point embedding of G is not optimal.
2. $\Rightarrow$ 1.: If the two-point embedding of G is not optimal then consider an embedding according to Proposition 2.1. It consists by Prop. 2.3 of a collection of at least two two-point embeddings of induced subgraphs of G. Consider the rightmost such two-point embedding. It is by Lemma 2.7 of the form $Q \cup N_G(Q)$ and with (2.2) and an application of Lemma 2.8 $\frac{|Q|}{|N_G(Q)|} > \frac{|W|}{|B|} > \frac{|W \setminus Q|}{|B \setminus N_G(Q)|}$.
2. $\Leftrightarrow$ 3.: Take complements. $\square$

Remark. The two-point embedding of a bipartite graph containing an isolated vertex is optimal $\Leftrightarrow W = \emptyset \vee B = \emptyset$.

Definition 2.10. *A bipartite graph G satisfying one of the conditions in Lemma 2.9 is called* balanced.

Remark. For disconnected graphs the assignment of the colours is important for balancedness: the disjoint union G of two paths P_1, P_2 with 3 vertices is balanced if and only if the two central vertices have the same colour.
Remark. This notion of balancedness has also been used by Fiedler[1]. His [3, Theorem 3.7] states that the balancedness of $G = (W\dot{\cup}B, E)$ is equivalent to the optimal λ_n of $L_w(G)$ being equal to the optimal λ_n of $L_w\left(K_{|W|,|B|}\right)$ where $K_{|W|,|B|} = (W\dot{\cup}B, W \times B)$ is the complete bipartite graph. The latter λ_n he obtains by symmetry argument [3, Theo. 2.5, Coro. 2.6]. Our criterion shows that the two-point embedding of $K_{|W|,|B|}$ is optimal and therefore our Lemma 2.9 can be viewed as the dual analogue of Fiedler's result.

The following lemma is turned in Section 3 into an algorithm for solving problem (1.3) for trees.

[1] Amusingly, we introduced the term "balanced" independently!

Lemma 2.11. *Let $G = (W\dot{\cup}B, E \subseteq W \times B)$ be a bipartite graph.*

1. *If V is an optimal one-dimensional solution to* (1.3) *and C is a connected component of G_V then C is balanced.*

2. *Conversely, let $(W_i)_{i\in\mathcal{J}}$ be a partition of W and $(B_i)_{i\in\mathcal{J}}$ of B and $C_i = G[W_i \cup B_i]$ (induced subgraphs). Let furthermore the following two conditions be satisfied.*

   (a) $\forall\, wb \in E\colon w \in W_i,\, b \in B_j \Rightarrow y_{C_i} \leq y_{C_j}$.

   (b) $\forall\, i \in \mathcal{J}: C_i$ *is balanced.*

   Then the embedding V of G which sends each C_i, $i \in \mathcal{J}$, to its two-point embedding is an optimal solution to (1.3).

Proof. 1: Assume C is not balanced. We can then find $Q \subseteq W(C)$ with $\frac{|Q|}{|N_C(Q)|} > \frac{|W(C)|}{B(C)}$ and $H = C[Q \cup N_C(Q)]$ as in the proof of Lemma 2.9. In the embedding V the white vertices of H have distance > 1 to any of their neighbours in $B(G) \setminus B(C)$. Therefore H can be perturbed to the right by a small positive amount without interfering with other components. Thus the objective value strictly decreases.
2: The first condition 2a ensures feasibility, cf. (2.3). Because C_i is optimally embedded by its two-point embedding there is a vector $w^{(i)} = (w_{ab},\, ab \in E(C_i))$ of optimal edge weights for the primal problem (1.2) for C_i with the same objective value. Concatenating the $w^{(i)}$, $i \in \mathcal{J}$ and setting $w_{ab} = 0$ for $ab \in E \setminus \bigcup_{i\in J} E(C_i)$ yields a feasible vector $w \in \mathbb{R}^E$ of edge weights for G. The objective value of G in problem (1.2) equals the objective value of V in problem (1.3) which proves optimality of both.

$\square$

For extracting the leftmost subgraph from the adjacency structure of G, Lem. 2.7 and (2.2) suggest to consider the following candidate.

Proposition 2.12. *Let $G = (W(G)\dot{\cup}B(G), E(G) \subseteq W(G) \times B(G))$ be a bipartite graph without isolated vertices. Then the set*

$$M = \operatorname*{Argmin}_{\emptyset \neq X \subseteq B(G)} \frac{|N_G(X)|}{|X|}$$

contains a unique element $S(G) \subseteq B(G)$ of maximal cardinality.

Proof. Let $S \in M$ be a set of maximal cardinality. If there was another set $T \in M$ with $|S| = |T|$, then $\frac{|N_G(S)|}{|S|} = \frac{|N_G(S)|+|N_G(T)|}{|S|+|T|} \geq \frac{|N_G(S\cup T)|+|N_G(S\cap T)|}{|S\cup T|+|S\cap T|}$. Since S is a minimiser, we have $\frac{|N_G(S\cap T)|}{|S\cap T|} \geq \frac{|N_G(S)|}{|S|}$. Because S has maximal cardinality and $S \neq T$ we have $\frac{|N_G(S\cup T)|}{|S\cup T|} > \frac{|N_G(S)|}{|S|}$. Lemma 2.8 immediately gives $\frac{|N_G(S)|}{|S|} \leq \min\left\{\frac{|N_G(S\cup T)|}{|S\cup T|}, \frac{|N_G(S\cap T)|}{|S\cap T|}\right\} < \frac{|N_G(S\cup T)|+|N_G(S\cap T)|}{|S\cup T|+|S\cap T|}$, a contradiction. $\square$

Proposition 2.13. *Let G be a bipartite graph without isolated vertices and consider the set of vertices $S(G)$ as in 2.12. Form the graph H from G by removing the set of vertices $N_G(S(G)) \cup S(G)$.*

1. *If the vertex set of the graph H is non-empty then it holds $\frac{|N_G(S(G))|}{|S(G)|} < \frac{|N_H(S(H))|}{|S(H)|}$.*

2. The subgraph of G induced by $N_G(S(G)) \cup S(G)$ is balanced.

Proof. First notice that $B \backslash S(G)$ and $W \backslash N_G(S(G))$ are either both empty or both non-empty. For if $S(G) = B$ then $N_G(S(G)) = W$ since there are no isolated vertices. If $W = N_G(S(G))$ then that minimum ratio is obtained by taking $S(G) = B$. Now let H be non-empty. Since $|S(G)|$ is maximal it holds

$$\begin{aligned}
\frac{|N_G(S(G))|}{|S(G)|} &< \frac{|N_G(S(G)) \cup (N_G(S(H)) \backslash N_G(S(G)))|}{|S(G) \cup S(H)|} \\
&= \frac{|N_G(S(G))| + |(N_G(S(H)) \backslash N_G(S(G)))|}{|S(G)| + |S(H)|} \\
&= \frac{|N_G(S(G))| + |N_H(S(H))|}{|S(G)| + |S(H)|},
\end{aligned}$$

With Lemma 2.8 we conclude that $\frac{|N_G(S(G))|}{|S(G)|} < \frac{|N_H(S(H))|}{|S(H)|}$.

Balancedness of the subgraph induced by $N_G(S(G)) \cup S(G)$ follows from Lemma 2.9. $\square$

The previous proposition directly implies the correctness of the following

Algorithm 2.14.
Input: *A bipartite graph $G = (W \dot\cup B, E \subseteq W \times B)$ without isolated vertices.*
Output: *Partitions $(W_1, \ldots, W_K)$ of W and $(B_1, \ldots, B_K)$ of B such that the induced subgraphs $C_i = G[W_i \cup B_i]$ satisfy conditions (2a) and (2b) of Lemma 2.11. The collection of their two-point embeddings then forms an optimal embedding of G.*
Initialisation: $i \leftarrow 0$.
while $B \neq \emptyset$ **do**

1. $i \leftarrow i + 1$.
2. $B_i \leftarrow S(G)$.
3. $W_i \leftarrow N_G(S(G))$.
4. $G \leftarrow G[(W \setminus W_i) \cup (B \setminus B_i)]$.

end while
return $(W_j \cup B_j, j = 1, \ldots, i)$.

Remarks: The assumption "no isolated vertices" in the previous algorithm is only for notational convenience. Simply remove isolated vertices, embed them in 0 and invoke Algorithm 2.14 for the remaining graph. Furthermore, Algorithm 2.14 needs polynomial time if the function $S(\cdot)$ does as the number of calls of $S(\cdot)$ is bounded by $|B|$.

Proposition 2.15. *Let $G = (W \dot\cup B, E \subseteq W \times B)$ without isolated vertices. Let $S(G)$ be defined as in Prop. 2.12 and let $C = G[N_G(S(G)) \cup S(G)]$. Then in any optimal embedding C is embedded in its two-point embedding $\{y_C, 1 + y_C\}$. The remaining vertices are embedded in points strictly greater than y_C.*

Proof. Let V be an optimal one-dimensional embedding and let $(C_i, i = 1, \ldots, K)$ be the connected components of G_V (cf. Prop. 2.3 and Lemma 2.11). Assume w.l.o.g. that $y_{C_1} <$

$\ldots < y_{C_K}$ (otherwise unify those C_i with identical two-point embeddings). By Lemma 2.7 and the minimality property of $S(G)$

$$\frac{|N_G(S(G))|}{|S(G)|} \leq \frac{|W(C_1)|}{|B(C_1)|} < \frac{|W(C_2)|}{|B(C_2)|} < \ldots < \frac{|W(C_K)|}{|B(C_K)|}.$$

Let L be the largest index i such that $(N_G(S(G)) \cup S(G)) \cap (W(C_i) \cup B(C_i)) \neq \emptyset$.
First case: $L = 1$, *i. e.* all of $N_G(S(G)) \cup S(G)$ is embedded in $\{y_{C_1}, 1 + y_{C_1}\}$. If $y_C = y_{C_1}$ then $C_1 = C$ as $S(G)$ is maximal w.r.t. inclusion. If $y_C < y_{C_1}$ $\left(\Leftrightarrow \frac{|N_G(S(G))|}{|S(G)|} < \frac{|W(C_1)|}{|B(C_1)|}\right)$ then $W(C) = N_G(S(G)) \subsetneq W(C_1)$ and $S(G) = B(C) \subseteq B(C_1)$. Then C_1 is not balanced by Lemma 2.9 and V not optimal by Lemma 2.11.
Second case: $L \geq 2$. $W(C_L) \cap W(C) \neq \emptyset$ because otherwise a vertex $i \in B(C_L) \cap B(C)$ would be isolated in G_V and therefore in G (Prop. 2.3), a contradiction. $B(C_L) \cap B(C) = \emptyset$ is also impossible because then all of $S(G)$ is embedded strictly to the left of $1 + y_{C_L}$ and by relation (2.3) there could not be an edge from $S(G)$ to $W(C_L)$ in G, again a contradiction.

Now let $D = W(C) \cap W(C_L) = N_G(S(G)) \cap W(C_L)$. Then $N_C(D) \subseteq B(C_L)$ by definition of C_L and the relation (2.3). Denote by H the subgraph of G induced by $D \cup N_C(D)$. Notice that we can move the embedding points of H to the left by a small amount without loosing feasibility. H's contribution to the objective in the current embedding is $f(y_{C_L})$ where $f(y) = |D| \cdot y^2 + |N_C(D)| \cdot (1 + y)^2$. This $f(y)$ is minimal for $y = y_H$. We now show that $y_H < y_{C_L}$ and thus a slight perturbation to the left decreases the objective value which yields the desired contradiction.

To that end observe that $N_G(S(G)) \setminus D \supseteq N_G(S(G) \setminus N_C(D)) \supseteq N_G(S(G) \setminus N_G(D))$ and therefore by the minimality property of $S(G)$

$$\frac{|N_G(S(G))|}{|S(G)|} \leq \frac{|N_G(S(G)) \setminus D|}{|S(G) \setminus N_G(D)|} = \frac{|N_G(S(G))| - |D|}{|S(G)| - |N_C(D)|}.$$

It follows with Lemma 2.8 that $\frac{|D|}{|N_C(D)|} \leq \frac{|N_G(S(G))|}{|S(G)|} < \frac{|W(C_L)|}{|B(C_L)|} \Leftrightarrow y_H < y_{C_L}$. $\square$

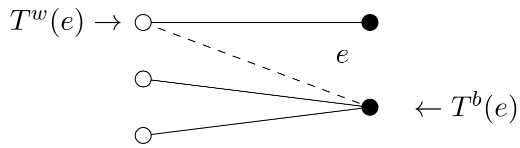
The previous proposition and the uniqueness of $S(G)$ show in particular that the decomposition into balanced subgraphs produced by the previous algorithm is the only one possible.

Corollary 2.16. *Let $G = (W \dot\cup B, E \subseteq W \times B)$ be a bipartite graph without isolated vertices. Then there is a unique one-dimensional optimal solution to* (1.3) *which maps $i \in W$ to $v_i < 0$ and $j \in B$ to $v_j > 0$.*

3 Trees

Lemma 2.11 suggests to find a feasible decomposition into balanced subgraphs. However, in a graph with many edges balancedness might be hard to be checked. For trees it turns out to be simple. We first introduce some further

Notation: Let $T = (V = W \dot\cup B, E \subseteq W \times B)$ be a tree. For any subgraph $G = (V(G), E(G)) \subseteq T$ define $W(G) := V(G) \cap W$, $B(G) := V(G) \cap B$ and $r(G) = \frac{|W(G)|}{|B(G)|}$. The removal of an edge e decomposes T into two subtrees namely the *black subtree* $T^b(e)$ and the *white subtree* $T^w(e)$ where $T^b(e)$ contains the black vertex of e and $T^w(e)$ the white vertex of e.



Using this notation Lemma 2.8 may be reformulated as follows for trees.

Lemma 3.1. *Let $T_1, \dots, T_m$ be node disjoint subtrees of some tree $T = (W\dot{\cup}B, E)$, then*

$$r(T_1) \leq \cdots \leq r(T_m) \quad \Rightarrow \quad r(T_1) \leq r(T_1 \cup \cdots \cup T_m) \leq r(T_m),$$

where the inequalities in the conclusion are strict if and only if at least one inequality in the premise is strict.

We state the tree version of Lemma 2.9.

Lemma 3.2. *A tree $T = (W\dot{\cup}B, E)$ is balanced (*i. e. *its two-point embedding is optimal) if and only if $r(T^b(e)) \leq r(T)$ for all $e \in E$.*

Proof. Notice that $T^b(e) = W(T^b(e)) \cup N_T(W(T^b(e)))$. So, if there is $e \in E$ with $r(T^b(e)) > r(T)$, then by Lemma 2.9 T is not balanced. Conversely, if T's two-point embedding is not optimal then consider the vertex set $S(T) \cup N_T(S(T))$ (cf. Prop. 2.12) and a connected component C of the induced subgraph $T\left[S(T) \cup N_T(S(T))\right]$. Then $\frac{|N_T(S(T))|}{|S(T)|} = r(C) < r(T)$. Because C is embedded leftmost, any edge connecting C to $T \setminus C$ must have its white vertex in C. Hence C is obtained from T by removing suitable subtrees $T^b(e_i), i \in \mathcal{J}$, and we can write T as a disjoint union $T = C \cup \bigcup_i T^b(e_i)$. Now $r(C) < r(T)$ together with Lemma 3.1 imply that $r(T^b(e_i)) > r(T)$ for some $i \in \mathcal{J}$. $\square$

Lemma 2.11 suggests to construct a decomposition into balanced subtrees. This can be done quite efficiently with the following algorithm.

Algorithm 3.3.
Input: *a tree $G = (B(G)\dot{\cup}W(G), \emptyset \neq E(G) \subseteq B(G) \times W(G))$.*
Output: *Decomposition of G into balanced trees as suggested in Lemma 2.11, hence an optimal embedding.*
Initialisation: *$\mathcal{P} \leftarrow \emptyset$ and $\mathcal{Q} \leftarrow \{G\}$.*
while $\mathcal{Q} \neq \emptyset$ **do:**

Choose a tree $\hat{T} \in \mathcal{Q}$, set $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{\hat{T}\}$, $T_0 \leftarrow \hat{T}$, $k \leftarrow 0$.

while $m \leftarrow \max\{r(T_k^b(e)) \colon e \in E(T_k)\} > r(T_k)$ **do**

1. *Choose an edge e_k with $r(T_k^b(e_k)) = m$.*
2. *Set $T_{k+1} \leftarrow T_k^w(e_k)$.*
3. *Set $k \leftarrow k + 1$.*

end while

$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{C \colon C$ *connected component of* $\hat{T} \setminus T_k\}$.

$\mathcal{P} \leftarrow \mathcal{P} \cup \{T_k\}$.

end while
return $\mathcal{P}$.

Remark. The algorithm produces one new component (each final T_k) of the optimal embedding (viz. one new two-point embedding) in every round of the outer loop.

The correctness proof of the algorithm relies on the following

Lemma 3.4. *The inner while loop processes a tree $\hat{T}$ with $E(\hat{T}) \neq \emptyset$ in $k < |B(\hat{T})|$ iterations. Let $D_j = T_j^b(e_j)$, $j = 0, \ldots, k-1$ denote the subtree cut off by edge e_i, then*

1. $\hat{T} = T_k \dot{\cup} D_{k-1} \dot{\cup} \ldots \dot{\cup} D_0$, $E(T_k) \neq \emptyset \neq E(D_i)$, $i = 1, \ldots, k$, *and T_k is balanced,*
2. $r(D_0) \geq r(D_1) \geq \cdots \geq r(D_{k-1}) > r(T_k)$,
3. *for $E' \subseteq E(\hat{T})$ and some $i \in \{0, \ldots, k-1\}$ with $(T_k \cup e_i) \subseteq \bigcap_{e \in E'} \hat{T}^w(e)$ there holds $r(D_i \setminus \bigcup_{e \in E'} T^b(e)) \geq r(D_i)$,*
4. *assuming $k > 0$, for $E' \subseteq E(\hat{T})$ with at least one $i \in \{0, \ldots, k-1\}$ so that $(T_k \cup e_i) \subseteq \bigcap_{e \in E'} \hat{T}^w(e)$ there holds $r(\hat{T} \setminus (T_k \cup \bigcup_{e \in E'} T^b(e))) > r(T_k)$,*
5. *any connected component C of $\hat{T} \setminus T_k$ satisfies $r(C) > r(T_k)$.*

Proof. Starting with $r(\hat{T} = T_0) > 0$ each further iteration $0 \leq i < k$ chooses an e_i with $r(T_i^b(e_i)) > r(T_i)$ so that $T_i = T_i^w(e_i) \cup T_i^b(e_i) = T_{i+1} \cup D_i$ satisfies $r(T_i^w(e_i)) < r(T_i) < r(T_i^b(e_i))$ by Lemma 3.1. The tree D_i contains a black node, so $r(D_i) > 0$ requires edges and thus $|B(T_{i+1})| < |B(T_i)|$. Likewise the tree T_{i+1} contains a white node, so by $r(T_{i+1}) < r(D_i)$ it contains an edge. This proves the iteration bound and part of 1.

1: The expression for $\hat{T}$ is a direct consequence of $T_i = T_{i+1} \cup D_i$ for $i = 0, \ldots, k-1$. By the loop's stopping criterion and Lemma 3.2 T_k is balanced.

2: Proceeding inductively for $i = 0, \ldots, k-1$ we obtain $r(T_{i+1}) = r(T_i^w(e_i)) < r(T_i) < r(T_i^b(e_i)) = r(D_i)$. For $i < k-1$, maximality of $r(T_i^b(e_i))$ and $e_{i+1} \in E(T_{i+1} = T_i^w(e_i))$ imply $r(D_{i+1}) = r(T_{i+1}^b(e_{i+1})) = r(T_i^b(e_{i+1}) \setminus T_i^b(e_i)) \leq r(T_i^b(e_{i+1})) \leq r(T_i^b(e_i)) = r(D_i)$ by Lemma 3.1.

3: Any relevant i has $\bar{D} = D_i \setminus \bigcup_{e \in E'} T^b(e)) \neq \emptyset$. W.l.o.g. we may assume $E' \subset E(D_i)$ and that the $T_i^b(e)$ are disjoint, then $D_i = \bar{D} \dot{\cup} \dot{\bigcup}_{e \in E'} T_i^b(e)$. The choice of e_i guarantees $r(T_i^b(e_i)) = r(D_i) \geq r(T_i^b(e))$ for $e \in E'$. Thus $r(\bar{D}) \geq r(D_i) \geq r(\dot{\bigcup}_{e \in E'} T_i^b(e))$ by Lemma 3.1.

4: Let $J = \{i = 0, \ldots, k-1 \colon e_i \in \bigcap_{e \in E'} \hat{T}^w(e)\}$, then $\hat{T} \setminus (T_k \cup \bigcup_{e \in E'} T^b(e)) = \bigcup_{i \in J} \left(D_i \setminus \bigcup_{e \in E'} T^b(e)\right)$, so the claim follows from 3 and 2.

5: This is a direct consequence of 2 and Lemma 3.1, because every connected component of $\hat{T} \setminus T_k$ is a disjoint union of some D_i. □

Proof of correctness of the algorithm. The algorithm terminates because by 1 of Lemma 3.4 each iteration $j = 1, 2, \ldots$ of the outer while loop stores a balanced final tree F_j (the final T_k of the inner loop) in $\mathcal{P}$ with $E(F_j) \neq 0$ and reduces the number of nodes in $\mathcal{Q}$ by at least two. The balancedness of the F_j of $\mathcal{P}$ ensures (2b) of the optimality conditions in Lemma 2.11, so it remains to show (2a). For this it suffices by (2.2) to prove that the following invariant conditions are satisfied at the test of the outer while loop:

1. No edge has its endpoints in distinct trees of $\mathcal{Q}$ and for each $F \in \mathcal{P}$ any outgoing edge $e = (w, b) \in E(G) \setminus E(F)$ with $w \in W(F)$ ends in $b \in B(T)$ for some $T \in \mathcal{P} \cup \mathcal{Q}$. with $r(F) < r(T)$ and Any incoming edge $e = (w, b) \in E(G) \setminus E(F)$ with $b \in B(F)$ originates in $w \in W(F')$ for some $F' \in \mathcal{P}$. with $r(F') < r(F)$.

2. For $F \in \mathcal{P}$ and $T \in \mathcal{P} \cup \mathcal{Q}$ and an edge $e = (w, b) \in E(G)$ with $w \in W(F)$ and $b \in B(T)$ we have $r(F) < r(T)$.

The proof is by induction. Initially $\mathcal{P} = \emptyset$, $\mathcal{Q} = \{G\}$, and the claim holds. So consider iteration $\hat{\jmath}$ of the outer loop, where a tree $\hat{T}_{\hat{\jmath}} \in \mathcal{Q}$ has been chosen and the inner while loop partitions this into a final subtree $F_{\hat{\jmath}}$ to be added to $\mathcal{P}$ and (possibly zero) connected components $S_1, \ldots, S_h$ of $\hat{T}_{\hat{\jmath}} \setminus F_{\hat{\jmath}}$ to be added to $\mathcal{Q}$; because the S_i are connected components, by induction no edge is incident to two distinct trees in $\mathcal{Q}$. For $S \in \{S_1, \ldots, S_h\}$ there is exactly one edge $e = (w, b) \in E(\hat{T}_{\hat{\jmath}})$ with $w \in W(F_{\hat{\jmath}})$ and $b \in B(S)$, and $r(F_{\hat{\jmath}}) < r(S)$ holds by 5 of Lemma 3.4. No other edges of $E(\hat{T}_{\hat{\jmath}})$ are involved in the edge cut that removes $F_{\hat{\jmath}}$ from $\hat{T}_{\hat{\jmath}}$. Hence, by induction, no edge has its white vertex in a tree in $\mathcal{Q}$ and its black vertex in some tree in $\mathcal{P}$. This shows part 1 and the case $T \in \mathcal{Q}$ of part 2 of the invariant conditions.

For $\hat{\jmath} \geq 2$ there may be a tree $F_{\bar{\jmath}} \in \mathcal{P}$ with $\bar{\jmath} < \hat{\jmath}$ having an outgoing edge $f_{\bar{\jmath}} = (w_{\bar{\jmath}}, b_{\bar{\jmath}}) \in E(G)$ with $w_{\bar{\jmath}} \in W(F_{\bar{\jmath}})$ and $b_{\bar{\jmath}} \in B(\hat{T}_{\hat{\jmath}})$. Let $S' \in \{F_{\hat{\jmath}}, S_1, \ldots, S_h\}$ with $b \in B(S')$, then the proof is complete once $r(F_{\bar{\jmath}}) < r(S')$ is established.

For this put $j_0 = \hat{\jmath}$, define the subtree $H_0 = S' \cup F_{\hat{\jmath}}$ of $\hat{T}_{j_0}$, and recursively for $i \geq 1$ put $j_i = \max\{j < j_{i-1} \colon \exists f_j = (w, b) \in E(G), w \in W(F_j), b \in B(H_{i-1})\}$ and define the subtree $H_i = F_{j_i} \cup H_{i-1}$ of $\hat{T}_{j_i}$ until $j_i = \bar{\jmath}$ for some $i = \bar{\imath}$. Indeed, the recursion stops for some $\bar{\imath} \geq 1$, because $F_{\bar{\jmath}}$ satisfies the requirements by $S' \subseteq H_i$. Note that $r(S') = r(H_0)$ if $S' = F_{\hat{\jmath}}$ and otherwise $r(S') > r(H_0) > r(F_{\hat{\jmath}})$ by virtue of 5 of Lemma 3.4 and Lemma 3.1. Thus it suffices to show $r(H_{i-1}) = r(H_i \setminus F_{j_i}) > r(F_{j_i})$ for $i = 1, \ldots, \bar{\imath}$, then Lemma 3.1 implies $r(S') \geq r(H_0) > \cdots > r(H_{\bar{\imath}}) > r(F_{\bar{\jmath}})$.

For proving $r(H_{i-1}) > r(F_{j_i})$, $i = 1, \ldots, \bar{\imath}$, observe that any edge $e = (w, b) \in E(\hat{T}_{j_i})$ with $b \in B(H_i)$ also satisfies $w \in V(H_i)$. Indeed, let $\bar{l} = \min\{l = 0, \ldots, i \colon b \in H_l = \bigcup_{k=0}^{l} F_{j_k} \cup S'\}$. If $w \in W(H_{\bar{l}})$ then $w \in W(H_i)$. Otherwise $b \in F_{j_{\bar{l}}} \cup S'$ ($b \in S'$ only for $\bar{l} = 0$) and, by part 1 of the invariant conditions, there must be some iteration $j < j_{\bar{l}}$ with $w \in W(F_j)$. Furthermore $j \geq j_i$, because $e \in E(\hat{T}_{j_i}) \cap E(\hat{T}_j)$. Therefore $j = j_{i'}$ for some $\bar{l} \leq i' \leq i$ giving $w \in W(H_i)$. Hence, for $E_i = \{(w, b) \in E(\hat{T}_{j_i}) \colon w \in W(H_i), b \notin B(H_i)\}$ and the f_{j_i} used in the definition of j_i we obtain $(F_{j_i} \cup f_{j_i}) \subseteq H_i = \bigcap_{e \in E_i} \hat{T}^b_{j_i}(e)$. Therefore 4 of Lemma 3.4 applied to $\hat{T} = \hat{T}_{j_i}$, $T_k = F_{j_i}$, $E' = E_i$ and $e_i = f_{j_i}$ asserts $r(H_{i-1} = H_i \setminus F_{j_i}) > r(F_{j_i})$, completing the proof. □

The proof also establishes the following observation.

Corollary 3.5. *Let $\mathcal{P}$ be the output of Algorithm 3.3 for a tree $G = (B(G) \dot{\cup} W(G), \emptyset \neq E(G) \subseteq B(G) \times W(G))$. If an edge $(w, b) \in E(G)$ links distinct trees $T_1, T_2 \in \mathcal{P}$ with $w \in W(T_1)$ and $b \in B(T_2)$ then $r(T_1) < r(T_2)$.*

Proposition 3.6. *The runtime of Algorithm 3.3 is $O\left(\min\{|W|, |B|\}^2 \cdot |W \cup B|\right)$.*

Proof. The outer while loop is traversed at most $\min\{|W|, |B|\}$ times, because in every iteration at least one black and one white vertex is moved to $\mathcal{P}$ and then no longer considered.

The inner while loop is also traversed at most $\min\{|W|, |B|\}$ times (with analogous reasoning). It can be worked out that a maximising edge can be found in $O(|W \cup B|)$ along with its value by one depth first traversal of the tree. This gives the bound. □

4 Bipartite graphs

We now use the tree Algorithm 3.3 to find the minimum ratio pair set and thereby implement Algorithm 2.14.

Algorithm 4.1.
Input: *Connected bipartite graph* $G = (N(G), E(G))$, $N(G) = W(G)\dot{\cup}B(G)$, $E(G) \subseteq W(G) \times B(G)$.
Output: $A \in$ *Argmin* $\left(\frac{|N_G(X)|}{|X|} : X \subseteq B(G)\right)$, $|A|$ *maximal.*
Initialisation: $T \leftarrow$ *spanning tree of* G.
repeat

1. *Compute an optimal decomposition of* T *into balanced subtrees* $(T_1, \ldots, T_K)$ *with Algorithm 3.3.*
2. *Determine all indices* i_l, $l = 1, \ldots, L \leq K$ *for which* $r(T_{i_l}) = \min\{r(T_i),\ i = 1, \ldots, K\}$ *holds and set* $A = \bigcup_{l=1}^{L} B(T_{i_l})$. *(By Prop. 2.15* $A = S(T)$*, the unique set of maximal cardinality in Argmin* $\left(\frac{|N_T(X)|}{|X|} : X \subseteq B(G)\right)$*)*
3. $E' = \{((w, b) \in E(G) \setminus E(T) \colon b \in A$ *and* $w \notin N_T(A)\}$.
4. **if** $E' \neq \emptyset$ **then**
   (a) *Choose an edge* $f = (w_f, b_f) \in E'$.
   (b) $E(T) \leftarrow E(T) \cup \{f\}$. *(This creates a unique cycle.)*
   (c) *Find the component* $T_{i_0} \in \{T_{i_1}, \ldots, T_{i_L}\}$ *with* $b_f \in T_{i_0}$ *and the (unique) edge* $e_0 = (w_0, b_0)$ *on the cycle in* T *with* $w_0 \in T_{i_0}$ *and* $b_0 \notin A$.
   (d) $E(T) \leftarrow E(T) \setminus \{e_0\}$

   end if

until $E' = \emptyset$.
return $(A, N_G(A))$.

Proof of correctness. Observe that the minimum ratio sets of any spanning subgraph T of G provide a lower bound because $\frac{|N_T(X)|}{|X|} \leq \frac{|N_G(X)|}{|X|}$ for every nonempty $X \subseteq B(G)$. If the algorithm produces a tree T for which E' is empty then $N_T(A) = N_G(A)$ and A is the sought for set. Therefore it remains to show that the algorithm indeed terminates. Let T' be a tree that is obtained from T in one exchange step ($T' = (T \setminus e_0) \cup f$) and let A' and A be their respective minimum ratio sets of maximum cardinality. We prove the following
Assertion: Either $\frac{|N_{T'}(A')|}{|A'|} > \frac{|N_T(A)|}{|A|}$ or $\frac{|N_{T'}(A')|}{|A'|} = \frac{|N_T(A)|}{|A|}$ and $A' \subsetneq A$.
Once this is shown the proof is complete. W.l.o.g. assume for the decomposition of T that

$$r(T_1) = \ldots = r(T_L) < r(T_i),\ i = L+1, \ldots, K$$

and T_1 and T_{L+1} are the trees incident with the edge $f \in E(G) \setminus E(T)$ of 4a. Then $A = \bigcup_{i=1}^{L} B(T_i)$, edge e_0 of 4c leads out of T_1 and all T_i remain balanced for the forest $F = (V(T), E(T) \setminus \{e_0\})$. In particular they are still balanced subtrees (though infeasible ones) of the next tree T' ($= F \cup f$). For relating these to the next set A' let T_1' be some connected component of $T'\left[A' \cup N_{T'}(A')\right]$. Because $r(T'_1)$ is minimal in T' and by Corollary 3.5, splitting T' along the edge set $H = \{(w,b) \in E(T')\colon w \in W(T'_1), b \in B(T \setminus T'_1)\}$ yields T'_1, *i. e.* $T'_1 = T' \setminus \bigcup_{e \in H} (T')^b(e)$. This H decides, which part of each T_i belongs to T'_1, thus partitioning the node set of T'_1 into subsets of the node sets of T_i by

$$V(T'_1) = \bigcup_{i=1}^{K} V(T_i \setminus \widetilde{T}_i) \tag{4.1}$$

where $\widetilde{T}_i$ is either empty ($T_i \subseteq T'_1$), or a collection of black subtrees $\widetilde{T}_i = \bigcup_{e \in H \cap E(T_i)} T_i^b(e)$, or $\widetilde{T}_i = T_i$ (*i. e.* no vertex of T_i in T'_1). The last case occurs in particular if $f \in H$ and $T_i \subseteq (T')^b(f)$, because f is contained in no T_i.

Whenever $T_i \setminus \widetilde{T}_i$ is nonempty, the fact that $\widetilde{T}_i$ is a collection of black subtrees ensures $N_{T_i}(B(T_i \setminus \widetilde{T}_i)) \subseteq W(T_i)$. For these i the balancedness of T_i and Lemma 2.9 yield

$$r(T_i \setminus \widetilde{T}_i) \geq r(T_i) \geq r(T_1).$$

We can now proceed to prove the assertion.
Case 1: For some $i \geq L+1$ we have a nonempty $T_i \setminus \widetilde{T}_i$. Then $r(T_i \setminus \widetilde{T}_i) \geq r(T_i) > r(T_1)$ and therefore $r(T'_1) > r(T_1)$ by Lemma 3.1 because every set in the union in (4.1) has ratio $\geq r(T_1)$ with the inequality strict for at least one set.
Case 2: Otherwise we have $T'_1 = \bigcup_{i=1}^{L} (T_i \setminus \widetilde{T}_i)$. Observe that there are no edges between any of $T_1, \ldots, T_L$ in T' because by Corollary 3.5 this is true in T and hence in F and the edge f links T_1 and T_{L+1}. Therefore, as T'_1 is connected, we actually have $T'_1 = T_i \setminus \widetilde{T}_i$ for some $i \in \{1, \ldots, L\}$. Assuming that $r(T'_1) = r(T_1)$ this shows in particular that $A' \subseteq A$ and also $N_{T'}(A') \subseteq N_T(A)$. Equality cannot hold: recall that $f = (b_f, w_f)$ with $b_f \in B(T_1)$ and $w_f \in W(T_{L+1})$ and therefore $N_{T'}(B(T_1)) = N_T(B(T_1)) \cup \{w_f\}$. This completes the proof of the assertion. □

Proposition 4.2. *The runtime of Algorithm 4.1 is* $O\left(|B|^2 \cdot |W| \cdot \min\{|W|, |B|\}^2 \cdot |W \cup B|\right)$.

Proof. Notice that the function $2^B \setminus \{\emptyset\} \to \mathbb{Q}$, $X \mapsto \frac{|N_T(X)|}{|X|}$ takes at most $|B| \cdot |W|$ different values. The correctness proof of the previous algorithm shows that either the cardinality of A decreases (at most $|B|$ times before a change in ratio occurs) or the ratio increases (at most $|B| \cdot |W|$ times) so there are no more than $|B|^2 \cdot |W|$ iterations of the loop. The dominant cost within the loop is the call of Algorithm 3.3. Therefore the running time is $O\left(|B|^2 \cdot |W| \cdot \min\{|W|, |B|\}^2 \cdot |W \cup B|\right)$. □

Poljak's method [9] to compute $S(G)$ requires $O(n|E|\log(n)^2)$ and clearly outperforms Algorithm 4.1. Observe, however, that our approach actually constructs a spanning tree T of G with $S(T) = S(G)$ and $N_T(S(T)) = N_G(S(G))$ and therefore $T[S(T) \cup N_T(S(T))]$ is a spanning forest of $G[S(G) \cup N_G(S(G))]$. This observation applied successively to the subgraphs produced by Algorithm 2.14 yield therefore the following corollary, cf. [3, Theo. 3.9].

Corollary 4.3. *If* $G = (W \cup B, E)$ *is bipartite then there is a spanning forest of* G *all of whose connected components are balanced trees and which has the same optimal one-dimensional solution to* (1.3) *as* G.

5 Conclusion

Let $G = (N, E)$ and $(\lambda_n, w = (w_{ij}, ij \in E))$ be an optimal solution to (1.1) and $V = (v_1, \ldots, v_n)$ be an optimal solution to (1.3). Then the Karush-Kuhn-Tucker conditions (cf. [10, 4]) assert

$$\lambda_n v_i = \sum_{ij \in E} w_{ij}(v_i - v_j) \quad (i \in N) \quad \left(\Leftrightarrow L_w(G)V^T = \lambda_n V^T\right) \tag{5.1}$$

$$w_{ij}(\|v_i - v_j\|^2 - 1) = 0 \quad (ij \in E). \tag{5.2}$$

Given V, we immediately get $\lambda_n = \left(\sum_i \|v_i\|^2\right)^{-1}$. For the spanning tree output by our algorithm (cf. Coro. 4.3) the optimal edge weights are uniquely determined by (5.1), simply solve the equations starting from the leaves. All edges of G not contained in that forest receive $w_{ij} = 0$.

Those weights can also be made explicit: For a balanced tree $(W \cup B, F)$ and its two-point embedding we have in particular $\lambda_n = \frac{|W|+|B|}{|W||B|}$ and (5.1) reads

$$\sum_{j:\, ij \in F} w_{ij} = \begin{cases} \frac{1}{|W|}, & i \in W, \\ \frac{1}{|B|}, & i \in B. \end{cases} \tag{5.3}$$

For edge $e \in F$ consider assigning the weight (cf. [3, Theorem 3.7])

$$w_e = \frac{|B(T^b(e))|}{|B|} - \frac{|W(T^b(e))|}{|W|} \tag{5.4}$$

which is non-negative by the balancedness of T. These weights satisfy (5.3): Fix $i \in B$ and let $d(i)$ be its degree. In the sums $\sum_{j:\, ij \in F} |W(T^b(ij))|$ resp. $\sum_{j:\, ij \in F} |B(T^b(ij))|$ every $w \in W$ and $b \in B \setminus \{i\}$ is counted $d(i) - 1$ times while i is counted $d(i)$ times, thus

$$\sum_{j:\, ij \in F} w_{ij} = \frac{(|B| - 1)(d(i) - 1) + d(i)}{|B|} - \frac{|W|(d(i) - 1)}{|W|} = \frac{1}{|B|}.$$

The argument for $i \in W$ is similar. Summing the previous equation over $i \in B$ proves $\sum_{e \in F} w_e = 1$, so the weights (5.4) are feasible and satisfy the optimality conditions. For a non-balanced tree one can compute the edge weights for every balanced subtree in an optimal decomposition individually by (5.4) and rescale them suitably. To summarise:

Corollary 5.1. *For a bipartite graph $G = (N, E)$, in particular a tree, an optimal solution $(\lambda_n, w = (w_{ij}, ij \in E))$ to (1.1) and an eigenvector of $L_w(G)$ corresponding to λ_n can be determined by a strongly polynomial time algorithm.*

The above weights yield also the multiplicity of the weight optimised λ_n for a tree.

Corollary 5.2. *If $T = (W \cup B, E)$ is a balanced tree and $k = |\{e \in E \colon r(T^b(e)) = r(T)\}|$ then the multiplicity of the weight optimised λ_n is equal to $k + 1$.*

Proof. By (5.4) an edge e receives weight 0 if and only if $r(T^b(e)) = r(T)$. In this case the strictly active subgraph T_w is not connected and $L_w(T)$ can be considered block diagonal with $k + 1$ blocks. For each block λ_n is a simple eigenvalue by a Perron-Frobenius argument [3]. □

In [3, Theo. 2.8] Fiedler shows under the assumption that λ_n is a simple eigenvalue of $L_w(G)$ with eigenvector $V^T = (v_1, \ldots, v_n)^T$, that conditions (5.1) and (5.2) are equivalent to the optimality of (λ_n, w), and in particular that G_w is bipartite. That assumption of simplicity can only be satisfied by bipartite graphs as (5.1) together with the following result show.

Proposition 5.3. *Let V be an optimal solution to* (1.3) *and let C be a connected component of G_V such that V restricted to C is one-dimensional. Then $G[N(C)]$ is bipartite. In particular, G is bipartite if and only if it admits a one-dimensional optimal solution to* (1.3)*.*

Proof. In [4] it is shown that for all graphs any optimal solution to (1.3) satisfies $\|v_i\| < 1$. Thus, for a one-dimensional optimal embedding there is $h \in \mathbb{R}^n$, $\|h\| = 1$ and $c_i \in]-1, 1[$ such that $i \in C$ is embedded at $v_i = c_i h$. Then $W = \{i : c_i < 0\}$ and $B = \{i : c_i \geq 0\}$ yield the desired bipartition of C, because edges within W or B would violate the distance constraints. The converse is Prop. 2.1. □

Acknowledgments This work was partially supported by CAPES Grant PROBRAL 408/13 - Brazil and DAAD PROBRAL Grant 56267227 - Germany.

References

[1] S. Boyd, P. Diaconis and L. Xiao. Fastest mixing Markov chain on a graph, *SIAM Rev.* **46** (2004), No. 4, 667–689.

[2] M. Fiedler. Algebraic connectivity of graphs, *Czechoslovak Mathematical Journal* **23** (1973), 298–305.

[3] M. Fiedler. A Minimax Problem for Graphs and Its Relation to Generalized Doubly Stochastic Matrices, *Linear and Multilinear Algebra* **27** (1990), 1–23.

[4] F. Göring, C. Helmberg and S. Reiss, Graph Realizations Associated with Minimizing the Maximum Eigenvalue of the Laplacian. *Mathematical Programming* **131** (2012), No. 1, 95–111.

[5] F. Göring, C. Helmberg and M. Wappler. Embedded in the Shadow of the Separator, *SIAM J. Optim.* **19** (2008), No. 1, 472–501.

[6] F. Göring, C. Helmberg and M. Wappler. The Rotational Dimension of a Graph, *Journal of Graph Theory* **66** (2011), No. 4, 283–302.

[7] B. Mohar, The Laplacian spectrum of graphs, in *Graph Theory, Combinatorics, and Applications*, Vol. 2, Y. Alavi, G. Chartrand, O. R. Oellermann, A. J. Schwenk (Eds.), Wiley, 1991, pp. 871-898.

[8] B. Mohar, Graph Laplacians, Chapter 4 in *Topics in Algebraic Graph Theory*, L.W. Beineke and R.J. Wilson (Eds.), Encyclopedia of Mathematics and Its Applications, Vol. 102, Cambridge University Press, Cambridge, 2004, pp. 113-136.

[9] S. Poljak, Minimum spectral radius of a weighted graph, *Linear Algebra and its Applications* **171** (1992), 53–63.

[10] S. Reiß. Optimizing Extremal Eigenvalues of Weighted Graph Laplacians and Associated Graph Realizations, *Dissertation*, Chemnitz University of Technology, Department of Mathematics (2012).

[11] J. Sun, S. Boyd, L. Xiao and P. Diaconis. The Fastest Mixing Markov Process on a Graph and a Connection to a Maximum Variance Unfolding Problem, *SIAM Rev.* **48** (2006), No. 4, 681–699.