

CONSTRAINED GRAPH PARTITIONING  
VIA MATRIX DIFFERENTIAL EQUATIONS\*ELEONORA ANDREOTTI<sup>†</sup>, DOMINIK EDELMANN<sup>‡</sup>, NICOLA GUGLIELMI<sup>§</sup>, AND  
CHRISTIAN LUBICH<sup>‡</sup>

**Abstract.** A novel algorithmic approach is presented for the problem of partitioning a connected undirected weighted graph under constraints such as cardinality or membership requirements or must-link and cannot-link constraints. Such constrained problems can be NP-hard combinatorial optimization problems. They are restated as matrix nearness problems for the weight matrix of the graph, where the objective is to minimize the distance between the given weight matrix and perturbed weight matrices for which a functional of an eigenvalue and eigenvector of the graph Laplacian takes its minimal value. A key element in the numerical solution of these matrix nearness problems is the use of a gradient system of matrix differential equations for the functional.

**Key words.** constrained minimum cut, constrained clustering, matrix nearness problem, gradient flow, matrix differential equation

**AMS subject classifications.** 15A18, 65K05

**DOI.** 10.1137/17M1160987

**1. Introduction.** In this paper we present a novel approach to partitioning a connected weighted undirected graph under constraints such as prescribing the minimum cardinality of connected components or the membership of selected vertices to the components, or under must-link and cannot-link constraints for specified pairs of vertices. These are important problems in the wider area of constrained clustering, which is an active field of research; see, e.g., [30, 2, 3, 4, 5, 6, 7, 20, 23, 24, 31, 32]. There are no standard methods for these problems, and existing algorithms are often heuristic and tailored to special applications. One main motivation for our approach is to obtain a versatile methodology with a clear theoretical foundation that can easily incorporate various kinds of constraints, or combinations of such constraints, in a unified way. The computational complexity of the proposed algorithm is determined by available numerical linear algebra routines for computing a few smallest eigenvalues and their associated eigenvectors of a symmetric positive semidefinite matrix and solving related linear systems of equations. The algorithm can therefore exploit the sparsity of large graphs by using powerful iterative methods that can have linear complexity in the number of vertices of the graph. With these properties, our approach appears to have the potential to significantly enrich the toolbox of methods for these challenging problems.

The starting point of our approach is spectral graph theory as pioneered by Fiedler [10]; see also the monograph by Chung [8] and the introductory articles [27, 29]. While constrained spectral clustering is an active and varied research area [7, 20, 23,

---

\*Received by the editors December 12, 2017; accepted for publication (in revised form) by D. J. Higham October 29, 2018; published electronically January 3, 2019.

<http://www.siam.org/journals/simax/40-1/M116098.html>

**Funding:** The work of the third author was supported by the Italian M.I.U.R., the INdAM GNCS, and the Center of Excellence DEWS.

<sup>†</sup>Dipartimento di Ingegneria Scienze Informatiche e Matematica, Università degli Studi di L’Aquila, I-67010 L’Aquila, Italy ([eleonora.andreotti@graduate.univaq.it](mailto:eleonora.andreotti@graduate.univaq.it)).

<sup>‡</sup>Mathematisches Institut, Universität Tübingen, D-72076 Tübingen, Germany ([dominik.edelmann@na.uni-tuebingen.de](mailto:dominik.edelmann@na.uni-tuebingen.de), [lubich@na.uni-tuebingen.de](mailto:lubich@na.uni-tuebingen.de)).

<sup>§</sup>Gran Sasso Science Institute, I-67100 L’Aquila, Italy ([nicola.guglielmi@gssi.it](mailto:nicola.guglielmi@gssi.it)).

31, 32], we here propose a conceptually different approach that does not use spectral information of the *given* graph Laplacian, but of the Laplacians of graphs with suitably *perturbed* weights.

We reformulate constrained partitioning problems as matrix nearness problems, where we minimize the distance between the given weight matrix of the graph and perturbed weight matrices for which a functional of an eigenvalue and eigenvector of the graph Laplacian takes its minimal value. We formulate and use a gradient system of matrix differential equations to drive the smallest nonzero eigenvalue of the graph Laplacian to zero and to satisfy the imposed constraints. Once this eigenvalue becomes zero, the graph is disconnected and the corresponding eigenvector indicates the membership of vertices to the connected components. The constraints are not enforced in the iteration steps of the algorithm, but eventually they become satisfied in the final result.

As opposed to combinatorial algorithms, the algorithm presented here modifies all weights of the graph as it proceeds, and only in the end arrives at the cut and the unchanged remaining weights. In an analogy with linear optimization, our algorithm is in the spirit of interior-point methods as opposed to the simplex method that hops from corner to corner on the admissible polytope. Our approach can be extended to other partitioning problems beyond the constrained minimum cut problems considered here by just modifying the considered functional and its gradient.

The proposed algorithm is an iterative algorithm, where in each step the second eigenvalue and the associated eigenvector of the Laplacian of a graph with perturbed weights are computed. In the constrained cases considered, additionally a linear system with an extended shifted Laplacian is solved in each step. For a large sparse graph (where the number of edges leaving any vertex is moderately bounded), these computations can typically be done with a complexity that is linear in the number of vertices.

A feature of this algorithm in common with recent algorithms for eigenvalue optimization via differential equations, such as given in [16, 15, 18, 17], is a two-level procedure for matrix nearness problems, where in an inner iteration a gradient flow drives perturbations to the original matrix of a *fixed* size into a (local) minimum of a nonnegative functional that depends on eigenvalues and eigenvectors, and in an outer iteration the perturbation size is optimized such that the functional becomes zero. As with the previous algorithms cited above, the algorithm presented here cannot guarantee to find the *global* minimum of a nonsmooth, nonconvex optimization problem, or of an NP-hard combinatorial optimization problem. There are cases where our algorithm could get stuck in a local minimum, and we will present a contrived example where this happens. Even with this caveat, the presented algorithm performs remarkably well in the examples from the literature on which we have tested it.

The main objective of this paper is to present a conceptually new approach for a wide class of problems, which are usually addressed by completely different and widely varying techniques. We provide a proof of concept for this approach and illustrate its feasibility by a few numerical experiments. However, it is beyond the scope of this paper to present an experimental evaluation in comparison with other, conceptually and algorithmically fundamentally different approaches.

In section 2 we formulate the Frobenius-norm minimum cut problem with membership, cardinality, must-link, and cannot-link constraints. This is stated as a matrix nearness problem where it is asked how far, with respect to the Frobenius norm, the weight matrix of the given graph is from that of some disconnected graph that should satisfy the imposed constraints. We give basic notation and recall Fiedler's theorem on graph connectivity.

In section 3 we first describe the two-level approach to the *unconstrained* Frobenius-norm minimum cut problem. This is the conceptually central section of the paper, where the basic approach is developed that is extended to constrained cases in the following.

In section 4 we extend the approach to minimum cut problems with membership, cardinality, must-link, and cannot-link constraints.

In section 5 we describe algorithmic aspects such as the discretization of the norm- and inequality-constrained gradient flow, the choice of initial values, and stopping criteria. In particular, since it is known beforehand that the weights of the cut graph are either zero or those of the original graph, the iteration need not be carried out to full convergence.

Section 6 illustrates the behavior of the proposed algorithm by numerical results for some graphs taken from the literature.

**2. Preparations and problem formulation.** We begin by giving basic notions and notation and describe the problems addressed in this paper.

**2.1. The minimum cut problem as a matrix nearness problem.** Consider a graph with vertex set  $\mathcal{V} = \{1, \dots, n\}$  and edge set  $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ . We assume that the graph is *undirected*: with  $(i, j) \in \mathcal{E}$ , also  $(j, i) \in \mathcal{E}$ . With the undirected graph we associate *weights*  $w_{ij}$  for  $(i, j) \in \mathcal{E}$ , such that

$$w_{ij} = w_{ji} \geq 0 \quad \text{for all } (i, j) \in \mathcal{E}.$$

The graph is *connected* if for all  $i, j \in \mathcal{V}$ , there is a path  $(i_0, i_1), (i_1, i_2), \dots, (i_{\ell-1}, i_\ell) \in \mathcal{E}$  of arbitrary length  $\ell$ , such that  $i = i_0$  and  $j = i_\ell$  and  $w_{i_{k-1}, i_k} > 0$  for all  $k = 1, \dots, \ell$ .

Given a connected weighted undirected graph with weights  $w_{ij}$ , we aim to find a *disconnected* weighted undirected graph with the same edge set  $\mathcal{E}$  and modified weights  $\widehat{w}_{ij}$  such that

$$(2.1) \quad \sum_{(i,j) \in \mathcal{E}} (\widehat{w}_{ij} - w_{ij})^2 \quad \text{is minimized.}$$

The solution to this matrix nearness problem is the same as that of finding a cut  $\mathcal{C}$ , i.e., a set of edges that yield a disconnected graph when they are removed from  $\mathcal{E}$ , where

$$\text{the cut } \mathcal{C} \text{ is such that } \sum_{(i,j) \in \mathcal{C}} w_{ij}^2 \quad \text{is minimized.}$$

When the weights are replaced by their square roots, so that  $w_{ij}$  instead of  $w_{ij}^2$  appears in the above sum, this becomes the classical minimum cut problem, for which algorithms with complexity  $O(|\mathcal{V}|^2 \log |\mathcal{V}| + |\mathcal{V}| \cdot |\mathcal{E}|)$  exist; see [28] and references therein.

**2.2. Constrained minimum cut problems.** The above problem will be considered with additional constraints. In particular, we consider the following cases:

- *Membership constraint.* It is required that a given set of vertices  $\mathcal{V}^+ \subset \mathcal{V}$  is in one connected component and another given set of vertices  $\mathcal{V}^- \subset \mathcal{V}$  is in the other connected component.
- *Cardinality constraint.* It is required that each of the connected components has a prescribed minimum number  $\bar{n}$  of vertices.
- *Must-link constraint.* This is similar to the membership constraint, but now it is required that pairs of vertices in a given set  $\mathcal{P} \subset \mathcal{V} \times \mathcal{V}$  are pairwise in the

same connected component. This kind of constraints includes the problem when subgraphs are specified that must not be cut.

- *Cannot-link constraint.* It is required that pairs of vertices in a given set  $\mathcal{P} \subset \mathcal{V} \times \mathcal{V}$  are pairwise in different connected components.

It is known that cardinality constraints make the problem NP-hard [5, 6]. A heuristic combinatorial algorithm for the cardinality-constrained minimum cut problem is proposed in [6]. Must-link and cannot-link constraints were considered in [30] and in many works thereafter; see, e.g., [1, 2, 3].

**2.3. Graph Laplacian and algebraic connectivity.** Setting  $w_{ij} = 0$  for  $(i, j) \notin \mathcal{E}$ , we have the symmetric weight matrix

$$W = (w_{ij}) \in \mathbb{R}^{n \times n}.$$

The degrees  $d_i = \sum_{j=1}^n w_{ij}$  are collected in the diagonal matrix

$$D = \text{diag}(d_i) = \text{diag}(W\mathbf{1}), \quad \text{where } \mathbf{1} := (1, \dots, 1)^T \in \mathbb{R}^n.$$

The *Laplacian matrix*  $L = \text{Lap}(W)$  is defined by

$$L = D - W, \quad \text{i.e.,} \quad \text{Lap}(W) = \text{diag}(W\mathbf{1}) - W.$$

We note that by the Gershgorin circle theorem, all eigenvalues of  $L$  are nonnegative, and  $L\mathbf{1} = 0$ , so that  $\lambda_1 = 0$  is the smallest eigenvalue of  $L$ . Remarkably, the connectivity of the graph is characterized by the second-smallest eigenvalue of  $L$ .

**THEOREM 2.1** (see Fiedler [10]). *Let  $W \in \mathbb{R}^{n \times n}$  be the weight matrix of an undirected graph and  $L$  the corresponding Laplacian matrix. Let  $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$  be the eigenvalues of  $L$ . Then, the graph is disconnected if and only if  $\lambda_2 = 0$ . Moreover, if  $0 = \lambda_2 < \lambda_3$ , then the entries of the corresponding eigenvector orthogonal to  $\mathbf{1}$  assume only two different values, of different sign, which mark the membership to the two connected components.*

Because of this result, the second smallest eigenvalue  $\lambda_2$  of  $L$  is called *algebraic connectivity* of  $W$ . If  $\lambda_2$  is a simple eigenvalue, then the corresponding eigenvector is known as the *Fiedler vector*.

**2.4. Normalized Laplacian.** Instead of the above Laplacian  $L$ , often a normalized Laplacian is considered in the literature (e.g., [19, 29] and references therein), which is given as

$$(2.2) \quad \widehat{L} = \widehat{\text{Lap}}(W) = I - D^{-1/2}WD^{-1/2} \quad \text{or} \quad \tilde{L} = I - D^{-1}W.$$

The eigenvalues  $\lambda$  of these normalized Laplacians are generalized eigenvalues  $Lv = \lambda Dv$  of the Laplacian  $L$ , and Fiedler's theorem is equally valid for them.

Our approach extends in a straightforward way to these normalized cases. We just obtain different expressions for the gradient of the functional used in our method; see section 3.4.

### 3. Two-level method for the Frobenius-norm minimum cut problem.

For ease of presentation of the basic ideas, we first describe our approach for the minimum cut problem (2.1) without including constraints. The extension to problems with constraints, which is the actual case of interest in this paper, is presented in the next section.

**3.1. Two-level formulation.** Our approach can be summarized as follows:

1. (Inner iteration) Given  $\varepsilon > 0$ , we look for a symmetric matrix  $E = (e_{ij}) \in \mathbb{R}^{n \times n}$  with the same sparsity pattern as  $W$  (i.e.,  $e_{ij} = 0$  if  $w_{ij} = 0$ ), of unit Frobenius norm, with  $W + \varepsilon E \geq 0$  (with componentwise inequality) such that the second smallest eigenvalue of  $\text{Lap}(W + \varepsilon E)$  is minimal. The obtained minimizer is denoted by  $E(\varepsilon)$ .
2. (Outer iteration) We look for the smallest value of  $\varepsilon$  such that the second smallest eigenvalue of  $\text{Lap}(W + \varepsilon E(\varepsilon))$  equals 0.

In order to compute  $E(\varepsilon)$  for a given  $\varepsilon > 0$ , we make use of a constrained gradient system for the functional

$$(3.1) \quad F_\varepsilon(E) = \lambda_2(\text{Lap}(W + \varepsilon E)),$$

under the constraints of unit Frobenius norm and  $W + \varepsilon E \geq 0$  and the symmetry and the sparsity pattern of  $E$ .

In the outer iteration we compute the optimal  $\varepsilon$  by a combined Newton-bisection method. This optimal  $\varepsilon$ , denoted  $\varepsilon^*$ , is the smallest  $\varepsilon$  such that  $\lambda_2(\text{Lap}(W + \varepsilon E)) = 0$  for some admissible  $E$  of unit norm.

The algorithm computes a partition of the graph as provided by the Fiedler vector corresponding to the weight matrix  $W + \varepsilon^* E(\varepsilon^*)$ , which here is understood as the eigenvector orthogonal to  $\mathbb{1}$  for the double eigenvalue 0.

This is not guaranteed to yield a global optimum for the Frobenius-norm minimum cut problem, since the gradient flow might converge only to a local minimum. In any case, it provides an upper bound for the distance problem (2.1).

### 3.2. Constrained gradient flow for the functional $F_\varepsilon$ .

**3.2.1. Eigenvalue derivatives.** We will use the following standard perturbation result for eigenvalues; see, e.g., [21, section II.1.1]. Here and in the following, we denote  $\dot{\cdot} = d/dt$ .

LEMMA 3.1. *Consider the differentiable symmetric  $n \times n$  matrix valued function  $C(t)$  for  $t$  in a neighborhood of a certain  $t_0$ . Let  $\lambda(t)$  be an eigenvalue of  $C(t)$  converging to a simple eigenvalue  $\lambda_0$  of  $C_0 = C(t_0)$  as  $t \rightarrow t_0$ . Let  $x_0$  be the associated eigenvector, with  $\|x_0\|_2 = 1$ . Then  $\lambda(t)$  is differentiable near  $t = t_0$  with*

$$\dot{\lambda}(t_0) = x_0^T \dot{C}(t_0) x_0.$$

**3.2.2. Gradient of  $F_\varepsilon$ .** We denote by  $\|\cdot\| = \|\cdot\|_F$  the Frobenius norm on  $\mathbb{R}^{n \times n}$  and by  $\langle X, Y \rangle = \text{trace}(X^T Y)$  the corresponding inner product.

We return to the situation of the previous section. For a set of edges  $\mathcal{E}$ , we define  $P_{\mathcal{E}}$  as the orthogonal projection from  $\mathbb{R}^{n \times n}$  onto the sparsity pattern determined by  $\mathcal{E}$ : for  $A = (a_{ij})$ ,

$$P_{\mathcal{E}}(A)|_{ij} := \begin{cases} a_{ij} & \text{if } (i, j) \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases}$$

For a fixed given weight matrix  $W$  and for  $\varepsilon > 0$ , we call a matrix  $E = (e_{ij}) \in \mathbb{R}^{n \times n}$   $\varepsilon$ -feasible if the following conditions are satisfied:

- (i)  $E$  is of unit Frobenius norm.
- (ii)  $E$  is symmetric.

- (iii)  $E = P_{\mathcal{E}}(E)$ .
- (iv)  $W + \varepsilon E \geq 0$ .

Consider now a smooth path  $E(t)$  of  $\varepsilon$ -feasible matrices, and denote the corresponding Laplacian matrix by  $L(t) = \text{Lap}(W + \varepsilon E(t))$  and by  $\lambda_2(t)$  the second smallest eigenvalue of  $L(t)$ . Lemma 3.1 applied to the Laplacian matrix  $L(t)$  yields (omitting the argument  $t$ )

$$(3.2) \quad \dot{\lambda}_2 = x^T \dot{L} x = \langle x x^T, \dot{L} \rangle,$$

where  $x(t)$  is a corresponding eigenvector of unit Euclidean norm. Next we rearrange (3.2) to an equation  $\dot{\lambda}_2 = \varepsilon \langle G_\varepsilon(E), \dot{E} \rangle$  with an appropriate matrix-valued function  $G_\varepsilon$ , which is the gradient of  $F_\varepsilon$  in the space of symmetric matrices with sparsity pattern  $\mathcal{E}$ . In the following,  $\text{Sym}(A) = \frac{1}{2}(A + A^T)$  denotes the symmetric part of a quadratic matrix  $A$ , and we write  $x \bullet x = (x_i^2) \in \mathbb{R}^n$  for the vector of squares of the entries of  $x = (x_i) \in \mathbb{R}^n$ .

LEMMA 3.2. *In the above situation we have*

$$(3.3) \quad \dot{\lambda}_2(t) = \varepsilon \langle G_\varepsilon(E(t)), \dot{E}(t) \rangle, \quad \text{where} \quad G_\varepsilon(E) = P_{\mathcal{E}}(\text{Sym}((x \bullet x) \mathbb{1}^T) - x x^T)$$

*with  $x$  denoting the normalized eigenvector to the second smallest eigenvalue of  $\text{Lap}(W + \varepsilon E)$ . The matrix  $G_\varepsilon(E)$  is symmetric and has the sparsity pattern determined by the set of edges  $\mathcal{E}$ .*

*Proof.* We note that by linearity

$$(3.4) \quad \begin{aligned} \frac{d}{dt} \text{Lap}((W + \varepsilon E(t))) &= \text{Lap}\left(\frac{d}{dt}(W + \varepsilon E(t))\right) \\ &= \varepsilon \text{Lap}(\dot{E}(t)) = \varepsilon(\text{diag}(\dot{E}(t) \mathbb{1}) - \dot{E}(t)). \end{aligned}$$

Combining (3.2) and (3.4), we obtain (omitting the ubiquitous argument  $t$ )

$$(3.5) \quad \dot{\lambda}_2 = \varepsilon \left( \langle x x^T, \text{diag}(\dot{E} \mathbb{1}) \rangle - \langle x x^T, \dot{E} \rangle \right).$$

The second term is already in the desired form. We obtain for the first term

$$(3.6) \quad \langle x x^T, \text{diag}(\dot{E} \mathbb{1}) \rangle = \sum_{i=1}^n x_i^2 (\dot{E} \mathbb{1})_i = \sum_{i=1}^n \sum_{j=1}^n x_i^2 \mathbb{1}_j \dot{e}_{ij} = \langle (x \bullet x) \mathbb{1}^T, \dot{E} \rangle.$$

This yields

$$\dot{\lambda}_2 = \varepsilon \langle (x \bullet x) \mathbb{1}^T - x x^T, \dot{E} \rangle.$$

Since  $\dot{E}$  and  $x x^T$  are symmetric, this can be rewritten as

$$\dot{\lambda}_2 = \varepsilon \langle \text{Sym}((x \bullet x) \mathbb{1}^T) - x x^T, \dot{E} \rangle.$$

We then have

$$\dot{\lambda}_2 = \varepsilon \langle G_\varepsilon(E), \dot{E} \rangle \quad \text{with} \quad G_\varepsilon(E) = P_{\mathcal{E}}(\text{Sym}((x \bullet x) \mathbb{1}^T) - x x^T).$$

This is in the desired form:  $G_\varepsilon(E)$  is symmetric and has the sparsity pattern  $\mathcal{E}$ .  $\square$

**3.2.3. Admissible directions.** Since  $E(t)$  is of unit Frobenius norm by condition (i), we have

$$0 = \frac{1}{2} \frac{d}{dt} \|E(t)\|^2 = \langle E(t), \dot{E}(t) \rangle.$$

To fulfill the nonnegativity condition (iv) of the weight matrix  $W + \varepsilon E(t)$ , we need that  $\dot{e}_{ij}(t) \geq 0$  for all  $(i, j)$  with  $w_{ij} + \varepsilon e_{ij}(t) = 0$ . Conditions (ii) and (iii) are satisfied if the same holds for  $\dot{E}$ . These four conditions are in fact also sufficient for a matrix to be the time derivative of a path of  $\varepsilon$ -feasible matrices. Hence, for every  $\varepsilon$ -feasible matrix  $E$ , a matrix  $Z = (z_{ij}) \in \mathbb{R}^{n \times n}$  is the derivative of some path of  $\varepsilon$ -feasible matrices passing through  $E$  if and only if the following four conditions are satisfied:

- (i')  $\langle E, Z \rangle = 0$ .
- (ii')  $Z$  is symmetric.
- (iii')  $Z = P_{\mathcal{E}}(Z)$ .
- (iv')  $P_{\mathcal{E}_0}(Z) \geq 0$ .

Here  $\mathcal{E}_0 = \mathcal{E}_0(\varepsilon E)$  is the set of cut edges defined by

$$\mathcal{E}_0 := \{(i, j) \in \mathcal{E} : w_{ij} + \varepsilon e_{ij} = 0\}.$$

Condition (iv') says that  $z_{ij} \geq 0$  for all  $(i, j) \in \mathcal{E}_0$ .

**3.2.4. Admissible direction of steepest descent.** To determine the admissible direction  $\dot{E}$  of steepest descent from  $E$ , we therefore consider the following optimization problem for  $G = G_\varepsilon(E)$ :

$$(3.7) \quad \min_Z \langle G, Z \rangle \quad \text{subject to (i')–(iv') and } \langle Z, Z \rangle = 1.$$

The additional constraint  $\|Z\| = 1$  just normalizes the descent direction. Problem (3.7) has a quadratic constraint. We now formulate a quadratic optimization problem with linear constraints, which is equivalent in the sense that it yields the same descent direction, provided that a strict descent direction exists, i.e., satisfying  $\langle G, Z \rangle < 0$  and the constraints (i')–(iv'). This is based on the fact that when  $\langle G, Z \rangle < 0$ , there exists a scaling factor  $\alpha > 0$  such that  $\langle G, \alpha Z \rangle = -1$ . Consider the following problem:

$$(3.8) \quad \min_Z \langle Z, Z \rangle \quad \text{subject to (i')–(iv') and } \langle G, Z \rangle = -1.$$

Both optimization problems yield the same Karush–Kuhn–Tucker (KKT) conditions (apart from the normalization). Since the objective function  $\langle Z, Z \rangle$  of problem (3.8) is convex and all constraints are linear, the KKT conditions are not only necessary but also sufficient conditions [12, Theorem 9.4.1], that is, a KKT point is already a solution of the optimization problem.

The solution of (3.8) satisfies the KKT conditions

$$(3.9a) \quad Z = -G - \kappa E + \sum_{(i,j) \in \mathcal{E}_0} \mu_{ij} e_i e_j^T,$$

$$(3.9b) \quad \mu_{ij} z_{ij} = 0 \text{ for all } (i, j) \in \mathcal{E}_0,$$

$$(3.9c) \quad \mu_{ij} \geq 0 \text{ for all } (i, j) \in \mathcal{E}_0.$$

In addition, there are conditions (i')–(iv').

**3.2.5. Constrained gradient flow.** The gradient flow of  $F_\varepsilon$  under the constraints (i)–(iv) is the system of differential equations

$$(3.10) \quad \dot{E}(t) = Z(t),$$

where  $Z(t)$  solves the KKT system (3.9) with  $G = G_\varepsilon(E(t))$  under the constraints (i')–(iv') with the set of edges  $\mathcal{E}_0(t) = \mathcal{E}_0(\varepsilon E(t))$ .

LEMMA 3.3. *On an interval where  $\mathcal{E}_0(t)$  does not change, the gradient system becomes, with  $P^+ = P_{\mathcal{E} \setminus \mathcal{E}_0}$  and omitting the ubiquitous argument  $t$ ,*

$$(3.11) \quad \dot{E} = -P^+G_\varepsilon(E) - \kappa P^+E \quad \text{with} \quad \kappa = \frac{\langle -G_\varepsilon(E), P^+E \rangle}{\|P^+E\|^2}.$$

*Proof.* The positive Lagrange multipliers  $\mu_{ij} > 0$  just have the role to ensure that  $\dot{e}_{ij} = 0$ . With  $G = G_\varepsilon(E)$ , the gradient system therefore reads

$$(3.12) \quad \dot{E} = P^+(-G - \kappa E),$$

where  $\kappa$  is determined from the constraint  $\langle E, \dot{E} \rangle = 0$ . We then have

$$0 = \langle E, \dot{E} \rangle = \langle E, P^+(-G - \kappa E) \rangle = -\langle P^+E, G \rangle - \kappa \langle P^+E, P^+E \rangle,$$

and the result follows.  $\square$

In a numerical solution of the gradient system, we thus have to monitor the sets of edges where  $w_{ij} + \varepsilon e_{ij} = 0$  and among them further those edges where the sign of  $-g_{ij} - \kappa e_{ij}$  changes. When the active set  $\{(i, j) : w_{ij} + \varepsilon e_{ij} = 0\}$  is changed, then also  $\kappa$  changes in a discontinuous way. Let  $\kappa_-$  and  $\kappa_+$  be the values of  $\kappa$  before and after the event of discontinuity, respectively. Then one has generically  $g_{ij} + \kappa_- e_{ij} > 0$  after the event for the critical edge  $(i, j)$ , but the sign of  $g_{ij} + \kappa_+ e_{ij}$  may be positive or negative. In the first case,  $(i, j)$  leaves  $\mathcal{E}_0$ . In the latter case, only a generalized solution exists [11], which keeps  $(i, j) \in \mathcal{E}_0$ , i.e.,  $w_{ij} + \varepsilon e_{ij} = 0$ . This is enforced until  $g_{ij} + \kappa_+ e_{ij}$  changes sign.

**3.2.6. Monotonicity and stationary points.** The following monotonicity result follows directly from the construction of the gradient system.

THEOREM 3.4. *Let  $E(t)$  of unit Frobenius norm satisfy the differential equation (3.12) with  $G_\varepsilon(E)$  of (3.3). Then, the second smallest eigenvalue  $\lambda_2(t)$  of the Laplacian matrix  $\text{Lap}(W + \varepsilon E(t))$  decreases monotonically with  $t$ :  $\dot{\lambda}_2(t) \leq 0$ .*

Equilibrium points of (3.12) are characterized as follows.

THEOREM 3.5. *The following statements are equivalent along the solutions of (3.12):*

1.  $\dot{\lambda}_2 = 0$ .
2.  $\dot{E} = 0$ .
3.  $P^+E$  is a real multiple of  $P^+G_\varepsilon(E)$ .

*Proof.* Using Lemma 3.2 and (3.12) we obtain, with  $G = G_\varepsilon(E)$ ,

$$\frac{1}{\varepsilon} \dot{\lambda}_2 = \langle G, \dot{E} \rangle = \langle G, -P^+G - \kappa P^+E \rangle = -\|P^+G\|^2 + \frac{\langle P^+G, P^+E \rangle^2}{\|P^+E\|^2}.$$

With the strong form of the Cauchy–Schwarz inequality, the result follows.  $\square$

**3.3. Newton-bisection outer iteration.** Let  $E(\varepsilon)$  denote the minimizer of the functional  $F_\varepsilon$ . In general we expect that for  $\varepsilon < \varepsilon^*$ , the eigenvalue  $\lambda_2(\text{Lap}(W + \varepsilon E(\varepsilon))) > 0$  is simple. If so, then  $f(\varepsilon) = F_\varepsilon(E(\varepsilon))$  is a piecewise smooth function of  $\varepsilon$  and we can exploit its regularity to obtain a fast iterative method to converge to  $\varepsilon^*$  from the left. Otherwise we can use a bisection technique to approach  $\varepsilon^*$ .

The following result provides an inexpensive formula for the computation of the derivative of  $f(\varepsilon) = F_\varepsilon(E(\varepsilon))$ , which will be useful in the construction of the outer iteration of the method.

We denote again  $P^+ = P_{\mathcal{E} \setminus \mathcal{E}_0}$ . We then have the following result.

**LEMMA 3.6.** *We assume that the second smallest eigenvalue of  $\text{Lap}(W + \varepsilon E(\varepsilon))$  is simple. Moreover,  $E(\varepsilon)$  is assumed to be a smooth function of  $\varepsilon$  in some interval, and the set of zero-weight edges  $\mathcal{E}_0$  related to  $E(\varepsilon)$  is independent of  $\varepsilon$  in the interval.*

*Then, the function  $f(\varepsilon) = F_\varepsilon(E(\varepsilon))$  is differentiable and its derivative equals (with  $' = d/d\varepsilon$ )*

$$(3.13) \quad f'(\varepsilon) = -\|P^+G_\varepsilon(E(\varepsilon))\| \|P^+E(\varepsilon)\| - \frac{1}{\varepsilon^2} \frac{\|P^+G_\varepsilon(E(\varepsilon))\|}{\|P^+E(\varepsilon)\|} \|P_{\mathcal{E}_0}W\|^2.$$

*Proof.* By assumption, the projection  $P^+$  remains constant near  $\varepsilon$ . This means that we are effectively working on a reduced set  $\hat{\mathcal{E}} = \mathcal{E} \setminus \mathcal{E}_0$  of edges. We set  $G^+(\varepsilon) = P^+G_\varepsilon(E(\varepsilon))$  and decompose

$$\varepsilon E(\varepsilon) = \varepsilon P^+E(\varepsilon) + R, \quad \text{where } R = (I - P^+)\varepsilon E(\varepsilon) = -(I - P^+)W = -P_{\mathcal{E}_0}W,$$

since  $w_{ij} + \varepsilon e_{ij}(\varepsilon) = 0$  for all  $(i, j) \in \mathcal{E}_0$ . In particular,  $R$  is independent of  $\varepsilon$ . Differentiating  $f(\varepsilon) = F_\varepsilon(E(\varepsilon))$  with respect to  $\varepsilon$  we obtain

$$(3.14) \quad f'(\varepsilon) = \langle G_\varepsilon(E(\varepsilon)), P^+E(\varepsilon) + \varepsilon P^+E'(\varepsilon) \rangle = \langle G^+(\varepsilon), P^+E(\varepsilon) + \varepsilon P^+E'(\varepsilon) \rangle.$$

The conservation of  $\|E(\varepsilon)\| = 1$  and of  $\|R\|$  for all  $\varepsilon$  implies

$$\langle P^+E(\varepsilon), P^+E'(\varepsilon) \rangle = \frac{d}{d\varepsilon} \frac{1}{2} \|P^+E(\varepsilon)\|^2 = \frac{d}{d\varepsilon} \frac{1}{2} (1 - \varepsilon^{-2}\|R\|^2) = \varepsilon^{-3}\|R\|^2.$$

Now we use the property of minimizers as stated by Theorem 3.5,

$$\frac{G^+(\varepsilon)}{\|G^+(\varepsilon)\|} = \pm \frac{P^+E(\varepsilon)}{\|P^+E(\varepsilon)\|},$$

which gives us

$$\langle G^+(\varepsilon), P^+E(\varepsilon) \rangle = \pm \|G^+(\varepsilon)\| \|P^+E(\varepsilon)\|$$

and

$$\begin{aligned} \langle G^+(\varepsilon), \varepsilon P^+E'(\varepsilon) \rangle &= \pm \varepsilon \frac{\|G^+(\varepsilon)\|}{\|P^+E(\varepsilon)\|} \langle P^+E(\varepsilon), P^+E'(\varepsilon) \rangle \\ &= \pm \varepsilon^{-2} \frac{\|G^+(\varepsilon)\|}{\|P^+E(\varepsilon)\|} \|R\|^2. \end{aligned}$$

From (3.14) we thus obtain the stated formula, since  $f'(\varepsilon) \leq 0$ .  $\square$

For  $\varepsilon = \varepsilon_k < \varepsilon^*$ , we make use of the standard Newton iteration

$$(3.15) \quad \varepsilon_{k+1} = \varepsilon_k - \frac{f(\varepsilon_k)}{f'(\varepsilon_k)}.$$

In a practical algorithm it is useful to couple the Newton iteration (3.15) with a bisection technique. To do this we adopt a tolerance tol which allows us to distinguish whether  $\varepsilon < \varepsilon^*$ , in which case we may use the derivative formula and perform the Newton step, or  $\varepsilon > \varepsilon^*$ , so that we have to make use of bisection. The method is formulated in Algorithm 1.

---

**Algorithm 1:** Newton-bisection method for distance approximation.

---

**Data:** Matrix  $W$  is given,  $k_{\max}$  (max number of iterations), tol (tolerance)  $\varepsilon_0$ ,  $\varepsilon_{lb}$ , and  $\varepsilon_{ub}$  (starting values for the lower and upper bounds for  $\varepsilon^*$ )

**Result:**  $\varepsilon^*$  (upper bound for the distance),  $E(\varepsilon^*)$

**begin**

```

1   Compute  $E(\varepsilon_0)$  by the inner iteration
2   Set  $k = 0$ 
3   while  $k \leq k_{\max}$  do
4     if  $f(\varepsilon_k) < \text{tol}$  then
5       Set  $\varepsilon_{ub} = \min(\varepsilon_{ub}, \varepsilon_k)$ 
6       Set  $\varepsilon_{k+1} = (\varepsilon_{lb} + \varepsilon_{ub})/2$  (bisection step)
7     else
8       Set  $\varepsilon_{lb} = \max(\varepsilon_{lb}, \varepsilon_k)$ 
9       Compute  $f(\varepsilon_k)$  and  $f'(\varepsilon_k)$ 
10      Compute  $\varepsilon_{k+1} = \varepsilon_k - \frac{f(\varepsilon_k)}{f'(\varepsilon_k)}$  (Newton step)
11    if  $\varepsilon_{k+1} \notin (\varepsilon_{lb}, \varepsilon_{ub})$  then
12      Set  $\varepsilon_{k+1} = (\varepsilon_{lb} + \varepsilon_{ub})/2$ 
13    if  $k = k_{\max}$  or  $\varepsilon_{ub} - \varepsilon_{lb} < \text{tol}$  then
14      Return  $\varepsilon_{k+1}$  and the interval  $[\varepsilon_{lb}, \varepsilon_{ub}]$ 
15      Stop
16    else
17      Set  $k = k + 1$ 
18  Compute  $E(\varepsilon_k)$  by the inner iteration
19
20  Return  $\varepsilon^* = \varepsilon_k$ 

```

---

**3.4. Modification for the normalized Laplacian.** If instead of (3.1) we consider the analogous functional for the normalized Laplacian,

$$(3.16) \quad \widehat{F}_\varepsilon(E) = \lambda_2(\widehat{\text{Lap}}(W + \varepsilon E)),$$

under the constraints of unit Frobenius norm and  $W + \varepsilon E \geq 0$  and the symmetry and the sparsity pattern of  $E$ , then the only difference concerns the expression for the free gradient, and the following lemma replaces Lemma 3.2. Here  $x$  denotes the normalized eigenvector corresponding to  $\lambda_2$ .

LEMMA 3.7. *In the above situation we have*

$$\dot{\lambda}_2 = \varepsilon \langle \widehat{G}_\varepsilon(E), \dot{E} \rangle,$$

where, with  $D = \text{diag}((W + \varepsilon E)\mathbb{1})$  and  $z = D^{-1/2}x$ ,

$$(3.17) \quad \widehat{G}_\varepsilon(E) = P_{\mathcal{E}}(\text{Sym}(\text{diagvec}(D^{-1}zz^T(W + \varepsilon E))\mathbb{1}^T)) - P_{\mathcal{E}}(zz^T)$$

is symmetric and has the sparsity pattern determined by the set of edges  $\mathcal{E}$ . Here  $\text{diagvec}$  of a matrix is the column vector of its diagonal elements.

*Proof.* We note that

$$\begin{aligned}\frac{d}{dt} \hat{L} &= \frac{d}{dt} (I - D^{-1/2}(W + \varepsilon E)D^{-1/2}) \\ &= -\left(\frac{d}{dt} D^{-1/2}\right)(W + \varepsilon E)D^{-1/2} - D^{-1/2}(W + \varepsilon E)\left(\frac{d}{dt} D^{-1/2}\right) - \varepsilon D^{-1/2} \dot{E} D^{-1/2},\end{aligned}$$

where we note

$$\frac{d}{dt} D^{-1/2} = -\frac{1}{2} D^{-3/2} \dot{D} \quad \text{with} \quad \dot{D} = \varepsilon \operatorname{diag}(\dot{E} \mathbb{1}).$$

We then obtain

$$\begin{aligned}\dot{\lambda}_2 &= \left\langle xx^T, \frac{d}{dt} \hat{L} \right\rangle \\ &= \langle zz^T, \operatorname{Sym}(D^{-1} \dot{D}(W + \varepsilon E)) - \varepsilon \dot{E} \rangle \\ &= \langle D^{-1}zz^T(W + \varepsilon E), \varepsilon \operatorname{diag}(\dot{E} \mathbb{1}) \rangle - \varepsilon \langle zz^T, \dot{E} \rangle \\ &= \varepsilon \langle \operatorname{diagvec}(D^{-1}zz^T(W + \varepsilon E)) \mathbb{1}^T, \dot{E} \rangle - \varepsilon \langle zz^T, \dot{E} \rangle.\end{aligned}$$

In view of the symmetry and the sparsity pattern of  $\mathcal{E}$  this gives the result.  $\square$

**4. The two-level method for constrained minimum cut problems.** Our approach to constrained problems is the same two-level procedure as for the unconstrained minimum cut problem, except that extra terms are added to the functional (3.1) such that the augmented functional takes the minimum value zero if and only if the graph is disconnected and the constraints are satisfied.

**4.1. Functionals for constrained minimum cut problems.** We formulate augmented functionals for the various constraints considered in section 2.2. We first collect some notation.

For  $\varepsilon > 0$  and a matrix  $E$  of unit Frobenius norm, let  $x = (x_i) \in \mathbb{R}^n$  be the eigenvector to the second smallest eigenvalue  $\lambda_2$  of  $\operatorname{Lap}(W + \varepsilon E)$  (or the normalized Laplacian). Let  $x^- = (x_i^-)$  with  $x_i^- = \min(x_i, 0)$  and  $x^+ = (x_i^+)$  with  $x_i^+ = \max(x_i, 0)$  collect the negative and positive components of  $x$ , respectively. Let  $n^-$  and  $n^+$  be the numbers of negative and nonnegative components of  $x$ , respectively. We denote the averages of  $x^-$  and  $x^+$  by

$$\langle x^- \rangle = \frac{1}{n^-} \sum_{i=1}^n x_i^-, \quad \langle x^+ \rangle = \frac{1}{n^+} \sum_{i=1}^n x_i^+.$$

**4.1.1. Functional for the membership-constrained minimum cut problem.** It is required that a given set of vertices  $\mathcal{V}^+ \subset \mathcal{V}$  is in one connected component and another given set of vertices  $\mathcal{V}^- \subset \mathcal{V}$  is in the other connected component.

Motivated by the special form of the eigenvectors as given in the Fiedler theorem (Theorem 2.1), we consider the functional

$$(4.1) \quad F_\varepsilon(E) = \lambda_2(\operatorname{Lap}(W + \varepsilon E)) + \frac{\alpha}{2} \sum_{i \in \mathcal{V}^-} (x_i - \langle x^- \rangle)^2 + \frac{\alpha}{2} \sum_{i \in \mathcal{V}^+} (x_i - \langle x^+ \rangle)^2,$$

where  $\alpha > 0$  is a weight to be chosen. The choice of the sign of the eigenvector  $x$  is such that  $F_\varepsilon(E)$  takes the smaller of the two possible values. This functional is to be minimized under the inequality constraints  $W + \varepsilon E \geq 0$ , the norm constraint  $\|E\| = 1$ , and the symmetry and the sparsity pattern of  $E$ .

#### 4.1.2. Functional for the cardinality-constrained minimum cut problem.

It is required that each of the connected components has a prescribed minimum number  $\bar{n}$  of vertices. We use the same functional  $F_\varepsilon$ , except that the sets  $\mathcal{V}^-$  and  $\mathcal{V}^+$  are not given a priori, but are chosen depending on  $E$  in the following way:  $\mathcal{V}^-$  and  $\mathcal{V}^+$  collect the indices of the smallest and largest  $\bar{n}$  components of the eigenvector  $x$ , respectively, augmented by those indices for which the components of  $x$  do not differ by more than a threshold  $\delta$  from the average of the smallest and largest  $\bar{n}$  components, respectively.

**4.1.3. Functional for the must-link minimum cut problem.** It is required that pairs of vertices in a given set  $\mathcal{P} \subset \mathcal{V} \times \mathcal{V}$  are pairwise in the same connected component. We now consider the functional

$$(4.2) \quad F_\varepsilon(E) = \lambda_2(\text{Lap}(W + \varepsilon E)) + \frac{\alpha}{2} \sum_{(i,j) \in \mathcal{P}} (x_i - x_j)^2.$$

**4.1.4. Functional for the cannot-link minimum cut problem.** It is required that pairs of vertices in a given set  $\mathcal{P} \subset \mathcal{V} \times \mathcal{V}$  are pairwise in different connected components. Here we consider the functional

$$(4.3) \quad F_\varepsilon(E) = \lambda_2(\text{Lap}(W + \varepsilon E)) + \frac{\alpha}{2} \sum_{(i,j) \in \mathcal{P}} \left( \frac{x_i}{\langle x^- \rangle} + \frac{x_j}{\langle x^+ \rangle} \right)^2 \left( \frac{x_i}{\langle x^+ \rangle} + \frac{x_j}{\langle x^- \rangle} \right)^2.$$

All these functionals are such that they take the value 0 when the graph is separated and the constraints are satisfied. The constraints can be combined by suitably adding the terms introduced in the previous functionals.

## 4.2. Constrained gradient flow for the augmented functional.

### 4.2.1. Eigenvector derivatives.

We use the following lemma.

**LEMMA 4.1** (see [25, Corollary 4]). *Consider the differentiable  $n \times n$  symmetric-matrix valued function  $C(t)$  for  $t$  in a neighborhood of 0, let  $\lambda(t)$  be a simple eigenvalue of  $C(t)$ , and let  $x(t)$  be the associated eigenvector normalized such that  $\|x(t)\|_2 = 1$ . Moreover, let  $M(t) = C(t) - \lambda(t)I$  and let  $M(t)^\dagger$  be the Moore–Penrose pseudoinverse of  $M(t)$ . Then, the derivative of the eigenvector is given by*

$$(4.4) \quad \dot{x}(t) = -M(t)^\dagger \dot{M}(t)x(t).$$

We remark that in [25] this is formulated with the group inverse, which in the symmetric case is the same as the Moore–Penrose pseudoinverse.

### 4.2.2. Gradient of $F_\varepsilon$ for the membership and cardinality constraints.

Consider a differentiable path  $E(t)$  of  $\varepsilon$ -feasible matrices, and denote the corresponding Laplacian matrix by  $L(t) = \text{Lap}(W + \varepsilon E(t))$ , by  $\lambda_2(t)$  the second smallest eigenvalue of  $L(t)$ , and by  $x(t)$  the associated eigenvector. We set

$$\mathbb{1}^- = (\mathbb{1}_i^-) \in \mathbb{R}^n \quad \text{with} \quad \mathbb{1}_i^- = \begin{cases} 1 & \text{if } x_i < 0 \\ 0 & \text{else,} \end{cases}$$

$$\mathbb{1}^+ = (\mathbb{1}_i^+) \in \mathbb{R}^n \quad \text{with} \quad \mathbb{1}_i^+ = \begin{cases} 1 & \text{if } x_i \geq 0 \\ 0 & \text{else,} \end{cases}$$

and, with  $e_i$  denoting the  $i$ th standard unit vector,

$$v = v^+ + v^- \quad \text{with} \quad v^\pm = - \sum_{i \in \mathcal{V}^\pm} (x_i - \langle x^\pm \rangle) \left( e_i - \frac{1}{n^\pm} \mathbb{1}^\pm \right).$$

We define

$$z = (L - \lambda_2 I)^\dagger v,$$

which is computed as the solution of the linear system

$$(4.5) \quad \begin{pmatrix} L - \lambda_2 I & x \\ x^T & 0 \end{pmatrix} \begin{pmatrix} z \\ \mu \end{pmatrix} = \begin{pmatrix} v \\ 0 \end{pmatrix}.$$

We denote by  $x \bullet y = (x_i y_i)$  the vector obtained by componentwise multiplication of the entries of  $x$  and  $y$ . We then have the following result.

**LEMMA 4.2.** *In the above situation we have*

$$(4.6) \quad \frac{d}{dt} F_\varepsilon(E) = \varepsilon \langle G_\varepsilon(E), \dot{E} \rangle, \quad \text{where}$$

$$(4.7) \quad G_\varepsilon(E) = P_{\mathcal{E}} \left( \text{Sym}((x \bullet (x + \alpha z) \mathbb{1}^T - x(x + \alpha z)^T) \right)$$

is symmetric and has the sparsity pattern determined by the set of edges  $\mathcal{E}$ .

*Proof.* We have

$$\frac{d}{dt} \frac{1}{2} \sum_{i \in \mathcal{V}^-} (x_i - \langle x^- \rangle)^2 = \sum_{i \in \mathcal{V}^-} (x_i - \langle x^- \rangle) \left( \dot{x}_i - \frac{d}{dt} \langle x^- \rangle \right)$$

and similarly for the sum over  $\mathcal{V}^+$ . With  $K = (L - \lambda_2 I)^\dagger$  we obtain from Lemma 4.1 and  $Kx = 0$  that

$$\dot{x}_i = -e_i^T K \dot{L} x, \quad \frac{d}{dt} \langle x^\pm \rangle = \frac{1}{n^\pm} \mathbb{1}^{\pm T} K \dot{L} x,$$

so that

$$\frac{d}{dt} \left( \frac{1}{2} \sum_{i \in \mathcal{V}^-} (x_i - \langle x^- \rangle)^2 + \frac{1}{2} \sum_{i \in \mathcal{V}^+} (x_i - \langle x^+ \rangle)^2 \right) = v^T K \dot{L} x = \langle K v x^T, \dot{L} \rangle = \langle z x^T, \dot{L} \rangle.$$

Using the expression for  $\dot{L}$  given in (3.4) and proceeding as in the proof of Lemma 3.2 gives the result.  $\square$

The computational cost of computing  $G_\varepsilon(E)$  lies in computing the second eigenvalue and its eigenvector and in solving the linear system (4.5). For a sparse weight matrix, these computations have a complexity that is linear in the number of vertices. With this gradient  $G_\varepsilon(E)$ , the further procedure is now exactly the same as in section 3.

**4.2.3. Gradient of  $F_\varepsilon$  for the must-link constraint.** For the given set of pairs  $\mathcal{P} = \{(i_1, j_1), \dots, (i_d, j_d)\}$  we define the rectangular  $d \times n$  matrix  $B$  such that  $B_{k,i_k} = 1$ ,  $B_{k,j_k} = -1$ , and zero otherwise. The functional  $F_\varepsilon$  for the must-link constraint is  $F_\varepsilon = \lambda_2 + \alpha \Phi_\varepsilon$  with

$$\Phi_\varepsilon(E) = \frac{1}{2} \sum_{(i,j) \in \mathcal{P}} (x_i - x_j)^2 = \frac{1}{2} \|Bx\|_2^2,$$

where  $x = (x_i)$  is the normalized eigenvector to the second-smallest eigenvalue of  $\text{Lap}(W + \varepsilon E)$ .

LEMMA 4.3. *For the functional  $\Phi_\varepsilon$  we have, with  $K = (L - \lambda_2 I)^\dagger$ ,*

$$(4.8) \quad \frac{d}{dt} \Phi_\varepsilon(E) = \varepsilon \langle \Gamma_\varepsilon(E), \dot{E} \rangle, \quad \text{where}$$

$$(4.9) \quad \Gamma_\varepsilon(E) = P_\varepsilon \text{Sym}(\text{diagvec}(KB^T Bxx^T) \mathbb{1}^T - KB^T Bxx^T)$$

*is symmetric and has the sparsity pattern determined by the set of edges  $\mathcal{E}$ .*

*Proof.* We use  $\dot{x} = -K\dot{L}x$  and  $\dot{L} = \text{diag}(\dot{E}\mathbb{1}) - \dot{E}$  to obtain

$$\begin{aligned} \dot{\Phi}_\varepsilon(E) &= \langle Bx, B\dot{x} \rangle = -\langle Bx, BK\dot{L}x \rangle = -\langle KB^T Bxx^T, \dot{L} \rangle \\ &= -\varepsilon \langle KB^T Bxx^T, \text{diag}(\dot{E}\mathbb{1}) - \dot{E} \rangle \\ &= \varepsilon \langle -\text{diagvec}(KB^T Bxx^T) \mathbb{1}^T + KB^T Bxx^T, \dot{E} \rangle \\ &= \varepsilon \langle -P_\varepsilon \text{Sym}(\text{diagvec}(KB^T Bxx^T) \mathbb{1}^T + KB^T Bxx^T), \dot{E} \rangle, \end{aligned}$$

which is the stated result.  $\square$

The gradient for the cannot-link constraint can be calculated in a similar way, but for brevity we do not work out the lengthy formula here.

## 5. Algorithmic aspects.

**5.1. Discretizing the constrained gradient flow.** We use a modified explicit Euler method for the approximate integration of the differential equation (3.10). For a given  $\varepsilon > 0$  and a step-size  $h > 0$  and from the  $\varepsilon$ -feasible perturbation matrix  $E^n$  of the  $n$ th time step, we compute  $E^{n+1}$  as follows. We compute  $G_\varepsilon(E^n) = (g_{ij}^n)$  and define  $\tilde{E}^{n+1} = (\tilde{e}_{ij}^{n+1})$  by setting

$$\tilde{e}_{ij}^{n+1} = e_{ij}^n - hg_{ij}^n \quad \text{if } w_{ij} + \varepsilon(e_{ij}^n - hg_{ij}^n) \geq 0$$

and else

$$\tilde{e}_{ij}^{n+1} = e_{ij}^n - \theta hg_{ij}^n \quad \text{with } \theta \in [0, 1] \text{ such that } w_{ij} + \varepsilon(e_{ij}^n - \theta hg_{ij}^n) = 0,$$

that is, with  $\theta = (w_{ij} + \varepsilon e_{ij}^n) / (\varepsilon hg_{ij}^n)$ . We would ideally take the new perturbation matrix  $E^{n+1}$  such that

$$\|E^{n+1} - \tilde{E}^{n+1}\| \rightarrow \min \quad \text{subject to} \quad \|E^{n+1}\| = 1 \quad \text{and} \quad W + \varepsilon E^{n+1} \geq 0.$$

We approximate this optimization problem by treating the two constraints one after the other in an alternating way. Let  $\mathcal{E}_0$  be the set of edges  $(i, j) \in \mathcal{E}$  for which  $w_{ij} + \varepsilon \tilde{e}_{ij}^{n+1} = 0$  (cut edges), and let  $P^0 = P_{\mathcal{E}_0}$  and  $P^+ = P_{\mathcal{E} \setminus \mathcal{E}_0}$  be the complementary projections as defined in section 3.2.2. We first normalize by choosing  $\rho > 0$  such that

$$\hat{E}^{n+1} = P^0 \tilde{E}^{n+1} + \rho P^+ \tilde{E}^{n+1}$$

has unit Frobenius norm, i.e.,

$$\rho = \frac{\sqrt{1 - \|P^0 \tilde{E}^{n+1}\|^2}}{\|P^+ \tilde{E}^{n+1}\|^2}.$$

(If  $\|P^0 \tilde{E}^{n+1}\|$  is larger than 1 or very close to 1, we replace  $P^0 \tilde{E}^{n+1}$  by  $P^0 E^n$  in the two lines above.) We denote by  $\mathcal{E}_-$  the set of inadmissible edges  $(i, j)$  for which  $w_{ij} + \varepsilon \hat{e}_{ij}^{n+1} < 0$ . We then reset  $\hat{e}_{ij}^{n+1}$  to

$$e_{ij}^{n+1} = -\frac{w_{ij}}{\varepsilon} \quad \text{for } (i, j) \in \mathcal{E}_-,$$

augment  $\mathcal{E}_0 := \mathcal{E}_0 \cup \mathcal{E}_-$ , and consider the updated projection  $P^0 = P_{\mathcal{E}_0}$ . We then normalize the so obtained matrix  $E^{n+1}$  in the same way as above by leaving the entries of  $P^0 E^{n+1}$  unchanged, reset the entries for inadmissible edges, normalize, and so on. As there are only finitely many edges, this iteration terminates after finitely many steps (typically after the first step). Finally, we have obtained an  $\varepsilon$ -feasible perturbation matrix  $E^{n+1}$ .

**5.2. Computational complexity of a time step.** The only computationally expensive part in the above algorithm is the evaluation of the gradient  $G_\varepsilon(E)$  as given in Lemma 4.2 for the problem with membership and cardinality constraints and in Lemma 4.3 together with Lemma 3.2 for the must-link constraint. This requires the computation of the second smallest eigenvalue of the Laplacian and the corresponding eigenvector, and in addition the solution of the linear system (4.5). These are core problems of numerical linear algebra. For large sparse graphs (where every vertex is only connected to a few other vertices), there are a variety of algorithms to perform these tasks by iterative methods based on matrix-vector multiplication which in this case is computed with linear complexity; see, e.g., [14, 22, 26].

**5.3. Choice of step-size.** The step-size  $h$  can, for example, be selected by the following adaptive algorithm. Here the objective is to reduce the function  $F_\varepsilon$ , not to follow accurately a trajectory of the constrained gradient differential equation.

---

**Algorithm 2:** Step-size selection.

---

**Data:** Matrix  $E_n$  and step-size  $h_{n-1}$  are given  
**Result:** Matrix  $E_{n+1}$  and step-size  $h_n$   
**begin**  
1    Initialize the step-size by the previous step-size,  $h = h_{n-1}$   
2    Compute  $E_{n+1}(h)$  and its function value  $F_\varepsilon(E_{n+1}(h))$  with the  
      step-size  $h$   
   **if**  $F_\varepsilon(E_{n+1}(h)) \geq F_\varepsilon(E_n)$  **then**  
      halve the step-size,  $h := h/2$  and repeat from 2  
   **else**  
      **if**  $h = h_{n-1}$  **then**  
        compute  $E_{n+1}(2h)$  and its function value  $F_\varepsilon(E_{n+1}(2h))$  with  
          the step-size  $2h$   
        **if**  $F_\varepsilon(E_{n+1}(2h)) \leq F_\varepsilon(E_{n+1}(h))$  **then**  
          double the step-size,  $h := 2h$   
        Set  $h_n = h$  and  $E_{n+1} = E_{n+1}(h)$   
3    Return  $E_{n+1}$  and  $h_n$

---

**5.4. Stopping criterion.** Let  $F^n = F_\varepsilon(E^n)$ . In order to stop the integration when  $F^n$  has approximately reached a stationary value, we use a criterion of the following type:

$$\text{Integrate until } F^n - F^{n+1} \leq \beta h F^n + \delta \quad \text{or} \quad F^n \leq \text{tol},$$

where tol is a tolerance parameter (e.g., tol =  $10^{-6}$ ) and  $\beta$  and  $\delta$  are further parameters. We had good experience with the choice  $\beta = 10 \cdot \text{tol}$  and  $\delta = \text{tol}/100$ ; see further [9], where also the sensitivity of the algorithm to the chosen parameters is discussed.

**5.5. Initial value of the constrained gradient flow for a new  $\varepsilon$ .** When we change to a new value of  $\varepsilon$  in the outer iteration, we need an initial value for the constrained gradient flow. A first idea might be to take the terminal perturbation matrix  $\tilde{E}^0 = E(\varepsilon_{\text{old}})$  as the initial value, but usually this does not satisfy the nonnegativity constraints  $W + \varepsilon E^0 \geq 0$  if  $\varepsilon \geq \varepsilon_{\text{old}}$ . We therefore modify  $\tilde{E}^0$  to  $E^0$  by solving approximately

$$\|E^0 - \tilde{E}^0\| \rightarrow \min \quad \text{subject to} \quad \|E^0\| = 1 \quad \text{and} \quad W + \varepsilon E^0 \geq 0,$$

alternating between normalization and enforcing the nonnegativity constraints as in section 5.1, but this time beginning with the empty set  $\mathcal{E}_0 = \emptyset$ .

**5.6. Choice of the initial perturbation size  $\varepsilon_0$  and the initial perturbation matrix.** While one might just start with a random perturbation, a more educated guess starts from the normalized free gradient  $E^0 = -G_\varepsilon(0)/\|G_\varepsilon(0)\|$  and determines  $\varepsilon_0$  as the largest number  $\varepsilon$  such that  $W + \varepsilon E^0 \geq 0$ .

**5.7. Stopping the outer iteration before convergence.** In the exact solution  $W^* = W + \varepsilon^* E^*$  to the constrained minimum cut problem, the entries of  $W^*$  are either zero or those of  $W$ . To decide about the cut, it is therefore not necessary to iterate toward  $W^*$  with very high accuracy, but instead the cut can be inferred earlier from a moderately accurate approximation  $W + \varepsilon E(\varepsilon)$ , for example, using the following criterion, with a small threshold parameter  $\vartheta > 0$ :

Stop if for every edge  $(i, j) \in \mathcal{E}$ , either  $w_{ij} + \varepsilon e_{ij} \leq \vartheta w_{ij}$  or  $|\varepsilon e_{ij}| \leq \vartheta w_{ij}$ .

In the first case one would then cut to  $w_{ij}^* = 0$  and in the second case one would leave the weight unchanged:  $w_{ij}^* = w_{ij}$ . It can finally be checked if the so obtained graph with cut edges is indeed disconnected and satisfies the constraints, by evaluating the functional at the obtained weight matrix  $W^*$ , where it should assume its minimal value 0.

**6. Numerical examples.** Without aiming at any detailed experimental evaluation, we here illustrate the convergence behavior of the proposed algorithm on some small-size examples.

*Example 1* (Zachary's karate club). This weighted graph consisting of 34 vertices describes the relationship between 34 members of a karate club. (For a detailed description see [33].)

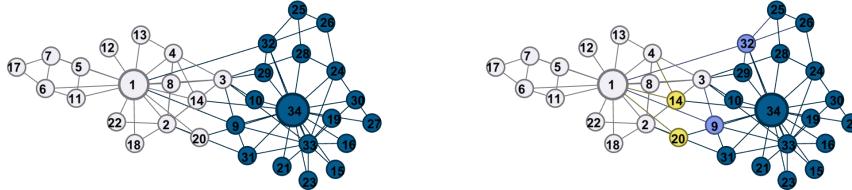
Using the unconstrained minimum cut, we obtain two connected components of 16 and 18 vertices as shown in Figure 6.1(a). The vertex labeled as 1 is the most connected vertex (in the sense that it has the highest degree  $d_i$ ), while the vertex labeled as 34 has the highest degree in the second component.

We next consider the following constraints:

- (i) cardinality constraint with threshold equal to  $\bar{n} = 17$  vertices;
- (ii) membership constraint;
- (iii) both constraints.

More detail follows:

- (i) By asking for a cardinality constraint with  $\bar{n} = 17$  vertices in each component, the approximate computed distance is  $\varepsilon^* \approx 14.9453$ . The results in Table 6.1 are obtained by setting a tolerance  $\text{tol} = 10^{-5}$  and the weight  $\alpha = 3$  in (4.1). With respect to the unconstrained situation, the vertex that changes partition is the vertex labeled as 9.



(a) Zachary's karate club colored by unconstrained partitioning.

(b) Zachary's karate club-membership constraint: highlighted the vertices required to change partition with respect to the clustering shown in (a).

FIG. 6.1. Example 1: Zachary's karate club.

TABLE 6.1  
Computed values of  $\varepsilon$ ,  $f(\varepsilon) = F_\varepsilon(E(\varepsilon))$  for Example 1(i).

k	$\varepsilon_k$	$f(\varepsilon_k)$
1	7.756152057843624	0.142509908100418
2	9.787100575121233	0.050183453498424
3	12.223780666610379	0.068209947467170
4	13.752942450391501	0.006705303461694
5	13.825885190965204	0.005584680775602
6	13.883253932728373	0.005164404072440
:	:	:

TABLE 6.2  
Computed values of  $\varepsilon^*$  for Example 1(ii).

Node	$\varepsilon^*$
9	16.947756820436005
14	19.816423934360159
20	26.394452875575567
32	19.849724386431539

- (ii) In the second case we consider the membership constraint: we ask for the vertices 1 and 34 to be in different partitions. Moreover we consider four different cases, that is, vertex 9 to be in the same connected component as vertex 1, vertex 32 to be in the same component as vertex 1, vertex 14 to be in the same component as vertex 34, or vertex 20 to be in the same component as vertex 34. For these four examples, Table 6.2 reports the values of  $\varepsilon^*$  for the functional  $F_\varepsilon$  of (4.1) with  $\alpha = 3$ , as computed by setting the tolerance  $\text{tol} = 10^{-5}$ . We can see from Table 6.2 that vertex 9 is the easiest to be required for changing the connected component; on the other hand, vertex 20 turns out to be the most difficult to change component.
- (iii) Finally we consider both constraints, that is, we ask for a cardinality constraint with threshold  $\bar{n} = 17$  vertices, and we require that vertex 9 is in the same connected component as vertex 34. Table 6.3 shows values of  $\varepsilon_k$ ,

TABLE 6.3  
Computed values of  $\varepsilon_k$ ,  $f(\varepsilon_k)$  for Example 1(iii).

k	$\varepsilon_k$	$f(\varepsilon_k)$
1	7.756152057843624	0.131754562008851
2	9.787100575121233	0.029535570936096
3	12.223780666610379	0.059967869767512
4	13.752942450391501	0.006343073447649
:	:	:

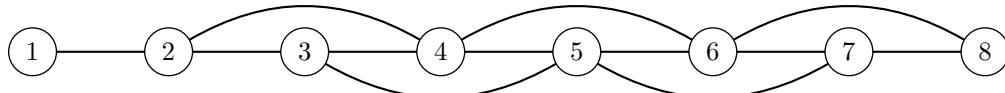


FIG. 6.2. Graph with eight vertices.

$f(\varepsilon_k) = F_{\varepsilon_k}(E(\varepsilon_k))$  computed with the tolerance  $\text{tol} = 10^{-5}$  and the weights  $\alpha_c = 3$ ,  $\alpha_m = 1$  in the functional  $F_\varepsilon$  that combines the cardinality and membership functionals. We obtain that vertex 10 further changes the connected component in order to satisfy the cardinality constraint.

*Example 2.* We present an example where the algorithm does not provide a globally optimal solution. Consider an unweighted graph with  $N$  vertices, such that each vertex  $2, \dots, N$  is connected to the following two vertices, i.e.,

$$w_{i,i+1} = w_{i,i+2} = 1 \text{ for all } 2 \leq i \leq N-2.$$

The first vertex is connected to the second one, i.e.,  $w_{1,2} = 1$ , but not to the third one; see Figure 6.2.

It is clear that the minimum cut in the sense of (2.1) is obtained by removing the edge  $(1, 2)$ , since this is the only possibility to obtain a disconnected graph when only one edge is removed. We tried to solve this example for different values of  $N$ . The algorithm works correctly when  $N = 8$  but fails when  $N \geq 12$ . In the latter case we obtain a disconnected graph, but with the wrong edges removed. For  $N = 20$  the resulting partition is  $\{1, 2, \dots, 10\} \cup \{11, 12, \dots, 20\}$  instead of the correct partition  $\{1\} \cup \{2, 3, \dots, 20\}$ . If we impose a cardinality constraint with  $\bar{n} = 10$ , we get the same solution.

*Example 3* (college football). Figure 6.3 shows the graph drawn from the schedule of games played between 115 American college football teams in the year 2000 [13]. This graph consists of 115 vertices (teams) and 615 edges (games between teams). The teams are divided into 12 conferences (Figure 6.3(a)). On the one hand, the games are more frequent between members of the same conference. On the other hand, teams that are geographically close to one another but belong to different conferences are more likely to play against each other.

According to the *unconstrained* partitioning, 45 of these teams belong to one part and the remaining 70 belong to the other part (Figure 6.3(b)). The results reported in Table 6.4 are obtained for the unconstrained partitioning by setting the tolerance  $\text{tol} = 10^{-4}$  (see section 5.4), the stopping parameter  $\vartheta = 0.2$  for the outer iteration (see section 5.7), and the initial perturbation distance  $\varepsilon_0 = 8$ . The obtained partition is shown in Figure 6.3(b).

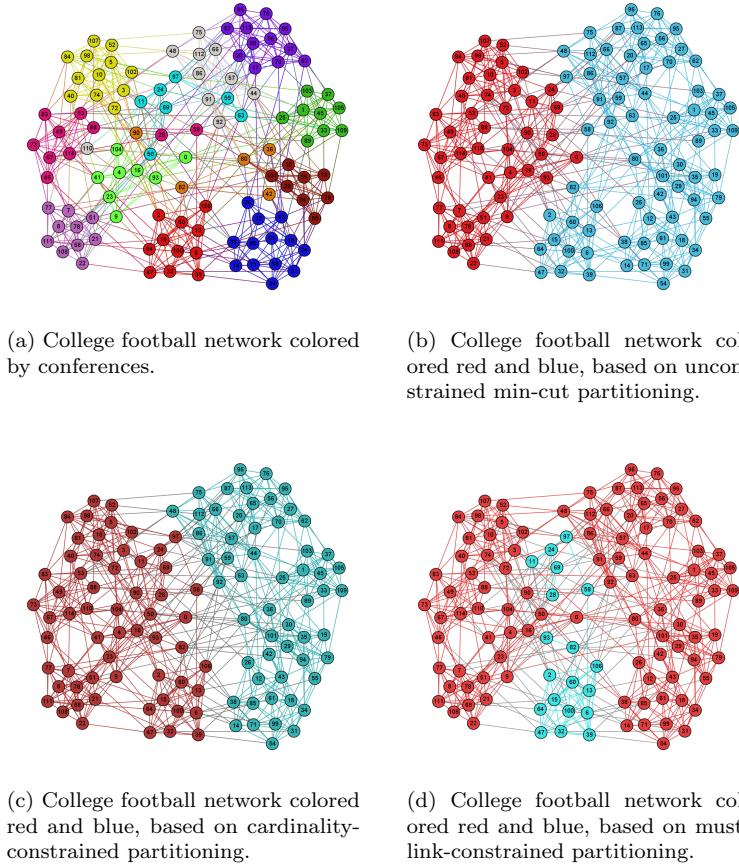


FIG. 6.3. Example 3: College football.

TABLE 6.4  
Computed values of  $\varepsilon_k$ ,  $f(\varepsilon_k)$  for Example 3, unconstrained partitioning.

$k$	$\varepsilon_k$	$f(\varepsilon_k)$
0	8.000000000000000	0.519001152871946
1	11.609559614086045	0.149775020987257
2	12.525977667358259	0.084179160890914
3	13.023731892410803	0.058424570397563
4	13.363225479974574	0.045754609637383
5	13.626288146096934	0

By asking for the partitioning of the graph with the *cardinality constraint* with threshold  $\bar{n} = 57$ , we obtain the result shown in Figure 6.3(c) and the approximate computed distance is  $\varepsilon^* \approx 17.8878$ ; see Table 6.5.

We then consider, for each conference, the partitioning under the *must-link constraint* that the teams of the specified conference remain linked. Table 6.6 shows the optimum distance  $\varepsilon^*$  and the number of outer iteration steps needed to reach the solution for each conference. The starting value was chosen as  $\varepsilon_0 = 13.6263$ , which is the optimum distance for the unconstrained partitioning as obtained in Table 6.4. Figure 6.3(d) shows the partition obtained by setting the fourth conference as a must-link

TABLE 6.5  
*Computed values of  $\varepsilon_k$ ,  $f(\varepsilon_k)$  for Example 3, cardinality-constrained partitioning.*

k	$\varepsilon_k$	$f(\varepsilon_k)$
0	10.00000000000000	0.399007309733819
1	12.433927928288854	0.138560614984389
2	13.157766624574450	0.095744465979244
3	13.637180229424644	0.081050066809253
4	14.032159200775542	0.071948127638700
5	14.373613615967034	0.067948606969860
6	14.686269681045403	0.065709259144731
:	:	:
33	17.785305714757254	0.020468018370244
34	17.820380682754184	0.019930623404160
35	17.854534620255983	0.019389536460448
36	17.887789619278561	0

TABLE 6.6  
*Computed values of  $\varepsilon^*$ ,  $k_{max}$  for Example 3, must-link constrained partitioning for each conference must-link.*

Conference must-link	$\varepsilon^*$	$k_{max}$
0-dark green	16.922701531308217	11
1-dark red	15.340082832306690	8
2-red	16.171039702592900	21
3-yellow	15.896379621686075	5
4-grey	19.871057129132542	5
5-orange	16.071136499285132	15
6-blue	14.308362810097600	3
7-light green	16.477663184752384	8
8-lilac	16.716976884368211	7
9-purple	17.200239963045021	9
10-light blue	18.707907862145131	18
11-pink	17.872577662019761	19

TABLE 6.7  
*Computed values of  $\varepsilon_k$ ,  $f(\varepsilon_k)$  for Example 3, must-link constrained partitioning for conference 4 (the grey one in Figure 6.3(a)).*

k	$\varepsilon_k$	$f(\varepsilon_k)$
0	13.626288146096934	0.375194439330142
1	15.774509755130810	0.359428976155161
2	17.822440712142924	0.343120253265900
3	19.025966023571375	0.332920628660836
4	19.679304384840886	0.160714285714286
5	19.871057129132542	0

set (the grey one in Figure 6.3(a) around team 112, including the outlying team 110 far on the left-hand side), and Table 6.7 gives the iterates  $\varepsilon_k$ .

**7. Conclusion.** We presented a novel algorithmic approach to solving constrained graph partitioning problems. We formulate constrained minimum cut problems as matrix nearness problems for the weight matrix of the graph under the constraint that a functional of the weight matrix takes its minimal value. We consider functionals that can be expressed in terms of an eigenvalue and the associated eigenvector of the graph Laplacian (or possibly several of these). Such functionals are

shown to arise for graph constraints that include membership, cardinality, must-link, and cannot-link constraints, which appear in important applications.

Our algorithmic approach is an iterative procedure with an inner and an outer iteration, where the inner iteration is based on the discretization of a gradient system of matrix differential equations for the functional, and where the outer iteration is just a one-dimensional optimization of the distance to the original weight matrix. The main computational cost is in the solution of eigenvalue problems and linear systems for modified graph Laplacians by suitable numerical linear algebra routines.

The proposed approach is very versatile and can easily be extended to other graph partitioning problems, as, for example, the following:

- While here we considered only bipartitioning, for which the second eigenvalue of the graph Laplacian is driven to zero, a corresponding algorithm for  $k$ -partitioning differs only in driving instead the sum of the first  $k$  eigenvalues to zero.
- The approach can be used with normalized Laplacians as well as with unnormalized Laplacians.
- Instead of the unnormalized minimum cut considered here, which is reformulated as a matrix nearness problem  $\|\widehat{W} - W\|_F \rightarrow \min$  under constraints on  $\widehat{W}$ , the approach could be extended to a normalized minimum cut, which can be reformulated as a weighted matrix nearness problem  $\varphi(\widehat{x}) \|\widehat{W} - W\|_F \rightarrow \min$ , where  $\varphi$  is an appropriately chosen function of the Fiedler eigenvector  $\widehat{x}$  corresponding to the modified weight matrix  $\widehat{W}$ .

We report promising numerical results, but a detailed evaluation and comparison with other, fundamentally different approaches from the literature is beyond the scope of this article. This paper can rather be viewed as a proof of concept for a potentially useful algorithmic idea for constrained clustering, a wide class of problems for which at present no standard methodology exists.

**Acknowledgments.** The authors thank Daniel Kressner (EPFL, Lausanne) for interesting discussions during an Oberwolfach meeting and Armando Bazzani (University of Bologna, Italy) for stimulating discussions. We thank the editor, Des Higham, and two anonymous referees for their constructive comments. Part of this work was developed during some visits to Gran Sasso Science Institute in L'Aquila and to the University of Tübingen. The authors thank both institutions for the very kind hospitality.

## REFERENCES

- [1] S. BASU, A. BANERJEE, AND R. J. MOONEY, *Semi-supervised clustering by seeding*, in Proceedings of the Nineteenth International Conference on Machine Learning (ICML '02), Morgan Kaufmann, San Francisco, CA, 2002, pp. 27–34.
- [2] S. BASU, M. BILENKO, AND R. J. MOONEY, *A probabilistic framework for semi-supervised clustering*, in Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04), New York, 2004, pp. 59–68.
- [3] S. BASU, I. DAVIDSON, AND K. WAGSTAFF, *Constrained Clustering: Advances in Algorithms, Theory, and Applications*, Chapman and Hall/CRC, Boca Raton, FL, 2008.
- [4] M. BILENKO, S. BASU, AND R. J. MOONEY, *Integrating constraints and metric learning in semi-supervised clustering*, in Proceedings of the Twenty-First International Conference on Machine Learning (ICML '04), ACM, New York, 2004, pp. 11–18.
- [5] M. BRUGLIERI, M. EHRGOTT, H. W. HAMACHER, AND F. MAFFIOLI, *An annotated bibliography of combinatorial optimization problems with fixed cardinality constraints*, Discrete Appl. Math., 154 (2006), pp. 1344–1357.
- [6] M. BRUGLIERI, F. MAFFIOLI, AND M. EHRGOTT, *Cardinality constrained minimum cut problems: Complexity and algorithms*, Discrete Appl. Math., 137 (2004), pp. 311–341.

- [7] T. BÜHLER, S. S. RANGAPURAM, S. SETZER, AND M. HEIN, *Constrained fractional set programs and their application in local clustering and community detection*, in Proceedings of the 30th International Conference on Machine Learning (ICML 2013), Atlanta, GA, 2013, pp. 624–632.
- [8] F. R. K. CHUNG, *Spectral Graph Theory*, AMS, Providence, RI, 1997.
- [9] D. EDELMANN, *Graph Partitioning Using Differential Equations*, Master's thesis, University of Tübingen, 2017.
- [10] M. FIEDLER, *Algebraic connectivity of graphs*, Czechoslovak Math. J., 23 (1973), pp. 298–305.
- [11] A. F. FILIPPOV, *Existence and general properties of solutions of discontinuous systems*, in Differential Equations with Discontinuous Righthand Sides, Springer, New York, 1988, pp. 48–122.
- [12] R. FLETCHER, *Practical Methods of Optimization*, John Wiley & Sons, New York, 2013.
- [13] M. GIRVAN AND M. E. J. NEWMAN, *Community structure in social and biological networks*, Proc. Natl. Acad. Sci. USA, 99 (2002), pp. 7821–7826.
- [14] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd edition, The Johns Hopkins University Press, Baltimore, MD, 1996.
- [15] N. GUGLIELMI, D. KRESSNER, AND C. LUBICH, *Low rank differential equations for Hamiltonian matrix nearness problems*, Numer. Math., 129 (2015), pp. 279–319.
- [16] N. GUGLIELMI AND C. LUBICH, *Differential equations for roaming pseudospectra: Paths to extremal points and boundary tracking*, SIAM J. Numer. Anal., 49 (2011), pp. 1194–1209.
- [17] N. GUGLIELMI AND C. LUBICH, *Matrix stabilization using differential equations*, SIAM J. Numer. Anal., 55 (2017), pp. 3097–3119.
- [18] N. GUGLIELMI, C. LUBICH, AND V. MEHRMANN, *On the nearest singular matrix pencil*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 776–806.
- [19] D. J. HIGHAM, G. KALNA, AND M. KIBBLE, *Spectral clustering and its use in bioinformatics*, J. Comput. Appl. Math., 204 (2007), pp. 25–37.
- [20] S. D. KAMVAR, D. KLEIN, AND C. D. MANNING, *Spectral learning*, in Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI '03), Morgan Kaufmann, San Francisco, CA, 2003, pp. 561–566.
- [21] T. KATO, *Perturbation Theory for Linear Operators*, Springer, New York, 1995.
- [22] R. B. LEHOUcq, D. C. SORENSEN, AND C. YANG, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, 1998.
- [23] Z. LI, J. LIU, AND X. TANG, *Constrained clustering via spectral regularization*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, 2009, pp. 421–428.
- [24] M. W. MAHONEY, L. ORECCHIA, AND N. K. VISHNOI, *A local spectral method for graphs: With applications to improving graph partitions and exploring data graphs locally*, J. Mach. Learn. Res., 13 (2012), pp. 2339–2365.
- [25] C. D. MEYER AND G. W. STEWART, *Derivatives and perturbations of eigenvectors*, SIAM J. Numer. Anal., 25 (1988), pp. 679–691.
- [26] Y. SAAD, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK, 1992.
- [27] D. SPIELMAN, *Spectral graph theory and its application*, in Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, 2007, pp. 29–38.
- [28] M. STOER AND F. WAGNER, *A simple min-cut algorithm*, J. ACM, 44 (1997), pp. 585–591.
- [29] U. VON LUXBURG, *A tutorial on spectral clustering*, Statist. Comput., 17 (2007), pp. 395–416.
- [30] K. WAGSTAFF, C. CARDIE, S. ROGERS, AND S. SCHRÖDL, *Constrained k-means clustering with background knowledge*, in Proceedings of the 18th International Conference on Machine Learning (ICML '01), Morgan Kaufmann, San Francisco, CA, 2001, pp. 577–584.
- [31] X. WANG AND I. DAVIDSON, *Flexible constrained spectral clustering*, in Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10), ACM, New York, 2010, pp. 563–572.
- [32] L. XU, W. LI, AND D. SCHUURMANS, *Fast normalized cut with linear constraints*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2866–2873.
- [33] W. W. ZACHARY, *An information flow model for conflict and fission in small groups*, J. Anthropological Res., 33 (1977), pp. 452–473.