Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.OA-2024-0051

Vol. **18**, No. 1, pp. 199-225
February 2025

Finding Cheeger Cuts via 1-Laplacian of Graphs

Wei Zhu*

Department of Mathematics, University of Alabama, Tuscaloosa, AL 35487, USA

Received 11 May 2024; Accepted (in revised version) 11 November 2024

Abstract. In this paper, we propose a novel algorithm for finding Cheeger cuts via 1-Laplacian of graphs. In [6], Chang introduced the theory of 1-Laplacian of graphs and built the connection between searching for the Cheeger cut of an undirected and unweighted graph and finding the first nonzero eigenvalue of 1-Laplacian, the latter of which is equivalent to solving a constrained non-convex optimization problem. We develop an alternating direction method of multipliers based algorithm to solve the optimization problem. We also prove that the generated sequence is bounded and it thus has a convergent subsequence. To find the goal optimal solution to the problem, we apply the proposed algorithm using different initial guesses and select the cut with the smallest cut value as the desired cut. Experimental results are presented for typical graphs, including Petersen's graph and Cockroach graphs, and the well-known Zachary karate club graph.

AMS subject classifications: 05C85, 65K10

Key words: Cheeger cuts, 1-Laplacian of graphs, alternating direction method of multipliers.

---

1. Introduction

Partitioning a large dataset into a prescribed number of subsets is a fundamental problem in machine learning and it has many applications in the fields ranging from computer sciences, statistics, computational biology, image processing, neural networks, and social sciences, etc. For instance, in the field of image processing, image segmentation can be regarded as a clustering problem that aims to partition a given image domain into several parts in order to capture target objects or extract features depicted in images. The community detection problem is also a clustering problem, aiming to divide a community into two or more smaller communities, each of which shares similar preference or homogeneity. For example, for the well-known Zachary's

*Corresponding author. *Email address:* `wzhu7@ua.edu` (W. Zhu)

karate club network [25], there are 34 members and 78 edges, and each edge indicates the associated two members interacted or were friends. One interesting question is to split this network into two groups such that members in each group have some tight connection and members in different groups are only loosely connected. In chemical engineering, cutting 3D crystals into two separate partitions by severing a minimum number of bonds assumes a large variety of technological applications [23]. Those crystals can also be described as graphs with atoms for nodes and bonds for edges.

All the above mentioned datasets can be expressed as graphs with nodes and edges. The partition of those data sets is indeed a graph cut problem, whose objective is to split a data set into sensible subsets such that points or nodes in each subset share some similarity or homogeneity while points in different subsets are dissimilar. Spectral graph theory [10] is one of the most successful mathematical tools for tackling graph cut problems. It employs the characteristic polynomials, eigenvalues, and eigenvectors of matrices that are associated with a given graph, including the graph Laplacian matrix, adjacency matrix, and so on. Its appealing feature lies in the fact that the properties of a graph, such as connectivity and symmetry, can be determined by the spectrum of those matrices.

To find some appropriate cut, specific energy functions or cut functions are often designed for graphs. In the literature, many different cuts have been proposed, including the Cheeger cut [8], the ratio cut [13], the normalized cut [19], etc. The Cheeger cut is one of the most important cuts and it originates from the well-known Cheeger's inequality from Riemannian geometry, where Cheeger proved an inequality that involves the first nontrivial eigenvalue of the Laplace-Beltrami operator on a compact Riemannian manifold. The Cheeger cut is a discrete analogue that associates with the graph Laplacian matrix.

Solving those cut problems is usually NP-hard and one has to resort to approximate solutions. Spectral graph theory provides the most popular approaches for obtaining such approximate solutions of the original cut problems [10, 22]. By using spectral graph theory, the original cut problem is relaxed to some linear algebra problem that can be handled easily.

In the literature, lots of research works have focused on the development of relaxation of the cut problems [4, 5, 19, 20, 22], especially for the Cheeger cut. Buhler *et al.* [5] considered the spectral clustering based on the graph p-Laplacian with $p > 1$, and showed that the limit cut, as $p \to 1+$, of thresholding the second eigenvector of this p-Laplacian converges to the optimal Cheeger cut. In fact, Kawohl *et al.* [14, 15] proved that the Cheeger constant equals to the limit of the first eigenvalue of the p-Laplacian as $p \to 1+$. In a recent work by Bresson *et al.* [4], they proposed minimizing the original l^1-relaxation of the Cheeger cut by employing the augmented Lagrangian method [11]. Even though these relaxations have produced lots of promising clustering results for a variety of practical problems, they only provide approximations for Cheeger cuts. Recently, Chang [6] developed a novel nonlinear spectral graph theory and systematically studied the 1-Laplacian of graphs, including the associated eigenvalue problem and the structure of its solutions. Most importantly, in this work, for the first time, Chang proved that the Cheeger's constant is equal to the first nonzero eigenvalue of the 1-Laplacian for a connected graph and the seek of the corresponding eigenvector amounts to solving a constrained optimization. This is totally different from the linear spectral theory, which merely provides lower and upper bounds for the Cheeger's constant.

However, solving the above-mentioned constrained optimization problem is also very challenging. As detailed later, the optimization problem consists of a nondifferentiable objective function over a non-convex domain. Especially, the domain consists of more than $3^{[(n+1)/2]} - 1$ simplex cells for a graph with n vertices, where $[x]$ represents the nearest integer that is less than or equal to x. In this paper, we propose using the augmented Lagrangian method (ALM) [11, 17] or alternating direction method of multipliers (ADMM) to solve the above optimization problem.

In the past decade, ALM/ADMM has been successfully applied for nonlinear, non-differentiable, and high-order variational models in image processing [1,12,21,24,26–30]. An appealing feature of using ALM/ADMM lies in the fact that solving the original optimization minimization amounts to the seek of saddle point of some augmented Lagrangian functional, which can be carried out by minimizing several relatively simpler functionals repeatedly and alternatingly. Usually these resulting functionals could have closed-form solutions.

The rest of the paper is organized as follows. In Section 2, we review Cheeger cut and Chang's work on 1-Laplacian of graphs, especially for the constrained optimization problem for finding the Cheeger cut. We then develop an ALM/ADMM based algorithm for solving the optimization problem in Section 3, where we detail how to solve those sub-problems. Numerical experiments are then presented in Section 5 by applying the proposed algorithm for typical graphs including Cockroach graphs, Petersen graph, and Zachary club graph, which is followed by our conclusion in Section 6.

2. Review of Cheeger cut and 1-Laplacian for graphs

In this section, we review a recent work by Chang [6], where the nonlinear spectral theory for graphs was first developed in the literature. Specifically, Chang introduced the 1-Laplacian operator for a graph and studied the spectrum. Most importantly, in this work, for the first time, Chang discovered the marvelous connection between the Cheeger's constant and the second eigenvalue of 1-Laplacian Δ_1 for any connected graph.

Let $G = (V, E)$ be an undirected and unweighted graph with vertices $V = \{1, \ldots, n\}$ and edge set E. Each edge $e \in E$ is a pair of vertices (i, j). For each vertex i, its degree, denoted as d_i, represents the number of edges passing through it. For any two subsets S and T of V, the set $E(S,T) = \{(i,j) \in E : i \in S, j \in T\}$ collects all the edges between S and T. The edge boundary of S is defined as $\partial S = E(S, S^c)$, where $S^c = V \setminus S$, the complement of S. The volume of S is defined as $\text{Vol}(S) = \sum_{i \in S} d_i$. With these notations, one could introduce the well-known Cheeger constant for the graph as follows:

$$h(G) = \min_{S \subset V} \frac{|\partial S|}{\min\{\mathrm{Vol}(S), \mathrm{Vol}(S^c)\}}, \tag{2.1}$$

where $|\partial S|$ is the cardinality of ∂S. A partition (S, S^c) of V is called a Cheeger cut of the graph G if the Cheeger constant is attained for the set S.

As discussed in the introduction, finding the Cheeger cut of a graph is an NP-hard problem. In the literature, many approaches have been proposed to approximate the solution [3–5, 19, 22]. One of the most popular methods is due to the spectral graphy theory [10]. Based on the theory, the standard graph Laplacian is defined for the graph $G = (V, E)$ as $L = D - A$, where $D = \mathrm{diag}(d_1, \ldots, d_n)$ is a diagonal matrix and $A = [a_{ij}]$ is the adjacency matrix with $a_{ij} = 1$ if $(i, j) \in E$ and $a_{ij} = 0$ if $(i, j) \notin E$. The matrix L has eigenvalues $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$ and its second eigenvalue λ_2 can be used to give bounds for the Cheeger's constant, that is, the Cheeger inequality,

$$\frac{\lambda_2}{2} \leq h(G) \leq \sqrt{2\lambda_2}. \tag{2.2}$$

Due to this relation, to approximate the Cheeger cut, one could find the associated eigenvector of λ_2 and use the sign of its elements or apply some threshold to them to obtain a cut. Later on, in [5], to get more tight approximation to the Cheeger cut, the graph p-Laplacian with $p \in (1, 2)$ was introduced as follows:

$$(\Delta_p x)_i = \sum_{j \sim i} |x_i - x_j|^{p-1} \mathrm{sign}(x_i - x_j), \tag{2.3}$$

where $j \sim i$ denotes the edge $(i, j) \in E$ and $\mathrm{sign}(t)$ is the standard sign function, which reads

$$\mathrm{sign}(t) = \begin{cases} 1, & \text{if} \quad t > 0, \\ 0, & \text{if} \quad t = 0, \\ -1, & \text{if} \quad t < 0. \end{cases}$$

In [5], the authors proved that, by thresholding the second eigenvector of this p-Laplacian, the resulting cut converges to the Cheeger cut as $p \to 1+$.

All these methods give only approximations to the Cheeger cut problem. Recently, Chang [6] developed a novel nonlinear spectral graph theory to study properties of graphs via 1-Laplacian operator.

In what follows, we recall the key details of Chang's theory on the spectrum of 1-Laplacian on graphs.

For an undirected and unweighted graph $G = (V, E)$ with a vertex set $V = \{1, \ldots, n\}$ and an edge set E, each edge e is a pair of vertices and one could assign an orientation as follows: if x is the head of e, denote $x = e_h$; if y is the tail of e, denote $y = e_t$. With this notation, an $m \times n$ incidence matrix $B = [b_{ei}]$ can be introduced

$$b_{ei} = \begin{cases} 1, & \text{if} \quad i = e_h, \\ -1, & \text{if} \quad i = e_t, \\ 0, & \text{if} \quad i \notin e, \end{cases} \tag{2.4}$$

where m is the number of edges in E.

In the spectral graph theory [10], the following Dirichlet function was introduced to study the properties of a graph:

$$J(x) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j\sim i}(x_i - x_j)^2, \tag{2.5}$$

while Chang [6] considered the following energy function:

$$I(x) = \sum_{j\sim i}|x_i - x_j|. \tag{2.6}$$

Note that the subdifferential of the convex function $t \to |t|$ is the set valued function $\partial|t| = \mathrm{Sgn}(t)$, which reads

$$\mathrm{Sgn}(t) = \begin{cases} 1, & \text{if} \quad t > 0, \\ -1, & \text{if} \quad t < 0, \\ [-1,1], & \text{if} \quad t = 0. \end{cases} \tag{2.7}$$

Then one can consider the subdifferential of the non-differentiable function $I(x)$. For this, Chang [6] proved the following theorem:

Theorem 2.1 ([6, Theorem 2.1]). *For any $x \in \mathbb{R}^n$, $u \in \partial I(x)$ if and only if there exists a function $z : E \to \mathbb{R}$ such that $u = B^T z$ and $z_e(Bx)_e = |(Bx)_e|$.*

Note that in this theorem, $(Bx)_e$ is the entry associated with the edge $e \in E$. For instance, if e is the first edge in the set E, $(Bx)_e$ denotes the first entry of the vector Bx. Moreover, $z_e = z(e)$ represents a real number in $[-1, 1]$ for the edge e. As shown in this theorem, for the edge $e = (i, j)$, $z_e = \partial|x_i - x_j|/\partial x_i$. Specifically, $z_e = 1$ if $x_i > x_j$, $z_e = -1$ if $x_i < x_j$, and $z_e = c$ for some $c \in [-1, 1]$ if $x_i = x_j$. Therefore, $z_e \in \mathrm{Sgn}(x_i - x_j)$ for $e = (i, j)$.

To introduce the concept of 1-Laplacian on the graph G, the following definition was given in [6].

Definition 2.1 ([6, Definition 2.3]). *For a given graph $G = (V, E)$, the set valued map $\Delta_1(G) : x \to \{B^T z | z : E \to \mathbb{R}$ is an $\mathbb{R}^m$-vector, satisfying $z_e(Bx)_e = |(Bx)_e|\}$ is called the 1-Laplacian on the graph G.*

Then the 1-Laplacian operator on graph can be expressed as

$$\Delta_1(G)x = B^T \mathrm{Sgn}(Bx), \tag{2.8}$$

where $\mathrm{Sgn}(y) = (\mathrm{Sgn}(y_1), \ldots, \mathrm{Sgn}(y_m))$ for $y = (y_1, \ldots, y_m) \in \mathbb{R}^m$. One can see that this definition of 1-Laplacian is independent of the choice of orientation of edges.

With the above definition and the set

$$X = \left\{ x = (x_1, \ldots, x_n) \in \mathbb{R}^n : \sum_{i=1}^{n} d_i|x_i| = 1 \right\}, \tag{2.9}$$

one can define the eigenvalue problem associated with the 1-Laplacian $\Delta_1(G)$ as follows:

Definition 2.2 ([6, Definition 2.4]). *$(\mu, x) \in \mathbb{R} \times X$ is called an eigenpair of the 1-Laplacian $\Delta_1(G)$ on $G = (V, E)$, if*

$$\mu D \mathrm{Sgn}(x) \cap \Delta_1(G)x \neq 0, \tag{2.10}$$

where $D = \mathrm{diag}(d_1, \ldots, d_n)$.

In this eigenpair (μ, x), μ is called an eigenvalue of $\Delta_1(G)$ and x is the associated eigenvector.

For any vector $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$, the vertex set V can be split in three groups:

$$D^0(x) = \{i \in V : x_i = 0\}, \quad D^{\pm}(x) = \{i \in V : \pm x_i > 0\}$$

and one can define $\delta^0(x) = \sum_{i \in D^0} d_i$, $\delta^{\pm}(x) = \sum_{i \in D^{\pm}} d_i$, and hence $\delta^0(x) + \delta^+(x) + \delta^-(x) = d$, where $d = \sum_{i=1}^n d_i$. With these notations, a special subset π of X is introduced

$$\pi = \left\{ x = (x_1, \ldots, x_n) \in X : |\delta^+(x) - \delta^-(x)| \leq \delta^0(x) \right\}. \tag{2.11}$$

Chang [6] proved any eigenvector x of 1-Laplacian $\Delta_1(G)$ with eigenvalue $\mu \neq 0$ must lie in the set π. Let K denote the set of all critical points of $I(x)$ over X. Chang showed that K is the same as the set of all eigenvectors of $\Delta_1(G)$. By studying the critical points of $I(x)$, Chang also proved that the spectrum of $\Delta_1(G)$ is discrete and any eigenvalue lies in the interval $[0, 1]$, and therefore those eigenvalues can be arranged as $0 = \mu_1 \leq \mu_2 \leq \cdots \leq 1$.

Most importantly, for the first time, Chang discovered a direct connection between the Cheeger cut and the second eigenvalue of $\Delta_1(G)$ that can be summarized in the following two theorems:

Theorem 2.2 ([6, Theorem 5.12]). *If $G = (V, E)$ is connected, then $\mu_2 = m$, where $m = \min\{I(x) : x \in \pi\}$.*

Theorem 2.3 ([6, Theorem 5.15]). *If $G = (V, E)$ is connected, then $\mu_2 = h(G)$.*

Therefore, to get the Cheeger cut, we need to find the second eigenvalue of the 1-Laplacian $\Delta_1(G)$, which can be obtained by searching for the minimizer of $I(x)$ over the set $\pi \subset X$. This minimizer or the second eigenvector x determines the Cheeger cut (S, S^c) by setting $S = D^+(x)$ or $S = D^-(x)$.

In a nutshell, finding the Cheeger cut of a given graph $G = (V, E)$ amounts to solving an optimization problem. As the set X is composed of cells of different dimensions, one could search for the minimizer over each of these cells. However, there are as many as $3^{[(n+1)/2]} - 1$ cells in the set π, and therefore this way of searching for the minimizer is NP-hard.

Chang *et al.* [7] developed a numerical algorithm called the cell descend (CD) to find the minimizer of $I(x)$ over the set π. The idea is as follows: as the objective function $I(x)$ is convex over each sub-cell of π, one could easily get the minimizer over one cell and then search for another cell in order to further minimize the function

$I(x)$ until no decreasing direction can be obtained. Numerical experiments for typical graphs of small size, as their Cheeger cuts can be obtained theoretically were reported in [7]. Recently, another interesting work on finding balanced graph cut was proposed by Shao and Yang [18].

In this work, we develop an ALM/ADMM based algorithm to solve the optimization problem in order to find the Cheeger cut for a given graph.

3. ADMM for finding Cheeger cuts

In this section, we discuss the details of our ADMM based algorithm to solve the Cheeger cut problem for a given graph by using the theory developed by Chang [6].

In what follows, we use $\|x\|_p$ to represent the p-norm of a vector $x \in \mathbb{R}^n$. As discussed in [6] and reviewed above, to find the Cheeger cut for a given graph $G = (V, E)$ with $V = \{1, \ldots, n\}$, one needs to solve the following optimization problem:

$$\begin{aligned} \min \quad & I(x) \\ \text{s.t.} \quad & x \in \pi = \left\{ x \in X : |\delta^+(x) - \delta^-(x)| \le \delta^0(x) \right\}, \end{aligned} \tag{3.1}$$

where

$$I(x) = \sum_{j \sim i} |x_i - x_j| = \sum_{e \in E} |(Bx)_e|.$$

The difficulties of tackling this problem lie in two aspects: first, the object function $I(x)$ is non-differentiable; second, besides being non-convex, the set π also imposes a strict constraint on the sign of the elements of the vector x, which raises another challenging issue for the seek of minimizer.

To deal with this tough optimization problem, we develop an ALM/ADMM based algorithm. ALM/ADMM has proven to be very successful in designing fast algorithms for variational imaging models with higher-order terms or/and non-differential terms during the past decade [21, 24, 26–31].

Note that $I(x)$ can be rewritten as $I(x) = \|Bx\|_1$. We recast the optimization problem (3.1) as an equivalent one

$$\begin{aligned} \min \quad & \|y\|_1 + \delta_\pi(z) \\ \text{s.t.} \quad & y = Bx, \quad x = z, \end{aligned} \tag{3.2}$$

where the indicator function δ_π is defined as

$$\delta_\pi(x) = \begin{cases} 0, & \text{if} \quad x \in \pi, \\ \infty, & \text{if} \quad x \notin \pi. \end{cases}$$

Then we consider the following augmented Lagrangian functional:

$$\mathcal{L}(x, y, z; \lambda_1, \lambda_2) = \|y\|_1 + \frac{r_1}{2}\|y - Bx + \lambda_1\|_2^2 + \frac{r_2}{2}\|x - z + \lambda_2\|_2^2 + \delta_\pi(z), \tag{3.3}$$

where $\lambda_1 \in \mathbb{R}^m, \lambda_2 \in \mathbb{R}^n$ are Lagrange multipliers and $r_1, r_2 > 0$ are penalty parameters.

Based on the optimization theory, to find the minimizer of $I(x)$ over the set π, one needs to obtain a saddle point of the augmented Lagrangian functional (3.3). For this, one can propose the ADMM to find the saddle point. Specifically, we minimize the corresponding subproblem for each of the variables x, y, and z respectively by fixing the other variables and then advance the Lagrange multipliers λ_1 and λ_2 accordingly; this process will be repeated until some criterion is met. This iterative method for approximating the saddle point of the functional (3.3) is given in Algorithm 3.1.

Algorithm 3.1 ADMM for the Minimization of (3.1).

1: Initialization: $x^0, y^0, z^0, \lambda_1^0, \lambda_2^0$.
2: **for** $k \geq 1$ **do**
3: Compute the minimizer x^k, y^k, z^k for the associated sub-problems with fixed Lagrange multipliers λ_1^{k-1} and λ_2^{k-1}

$$x^{k+1} = \text{argmin}_x \mathcal{L}\big(x, y^k, z^k; \lambda_1^k, \lambda_2^k\big), \tag{3.4}$$

$$y^{k+1} = \text{argmin}_y \mathcal{L}\big(x^{k+1}, y, z^k; \lambda_1^k, \lambda_2^k\big), \tag{3.5}$$

$$z^{k+1} = \text{argmin}_z \mathcal{L}\big(x^{k+1}, y^{k+1}, z; \lambda_1^k, \lambda_2^k\big). \tag{3.6}$$

4: Update the Lagrange multipliers

$$\lambda_1^{k+1} = \lambda_1^k + y^{k+1} - Bx^{k+1}, \tag{3.7}$$

$$\lambda_2^{k+1} = \lambda_2^k + x^{k+1} - z^{k+1}. \tag{3.8}$$

5: Stop the iteration if some criterion is met.
6: **end for**

In what follows, we discuss how to solve the three sub-problems (3.4)-(3.6) one by one.

3.1. The sub-problems for the variables x and y

For the sub-problem (3.4), one needs to find the minimizer of the function

$$\varepsilon_1(x) = \frac{r_1}{2}\big\|y^k - Bx + \lambda_1^k\big\|_2^2 + \frac{r_2}{2}\big\|x - z^k + \lambda_2^k\big\|_2^2. \tag{3.9}$$

Its minimizer is given by the following equation:

$$\big(r_1 B^T B + r_2 I_n\big)x = r_1 B^T\big(y^k + \lambda_1^k\big) + r_2\big(z^k - \lambda_2^k\big), \tag{3.10}$$

where I_n represents the $n \times n$ identity matrix. As the coefficient matrix $r_1 B^T B + r_2 I_n$ is symmetric positive definite, the system can be readily solved using the preconditioned conjugate gradients method (PCG).

As for the sub-problem (3.5), the associated function to be minimized can be expressed as follows:

$$\varepsilon_2(y) = \|y\|_1 + \frac{r_1}{2}\|y-a\|_2^2, \tag{3.11}$$

where $a = [a_1, \dots, a_m]^T = Bx^{k+1} - \lambda_1^k$. This is a Lasso problem and its minimizer $y = [y_1, \dots, y_m]^T$ is given for the i^{th} component as follows:

$$y_i = \begin{cases} 0, & \text{if} \quad |a_i| \le \dfrac{1}{r_1}, \\ \left(1 - \dfrac{1}{r_1|a_i|}\right) a_i, & \text{if} \quad |a_i| > \dfrac{1}{r_1}. \end{cases} \tag{3.12}$$

3.2. The sub-problem for the variable z

We next discuss how to solve the third sub-problem (3.6), which can re-formulated as an equivalent problem

$$\begin{aligned} \min \quad & \|z-b\|_2 \\ \text{s.t.} \quad & z \in \pi, \end{aligned} \tag{3.13}$$

where $b = x^{k+1} + \lambda_2^k$. Geometrically, this problem seeks the projection of the point $b \in \mathbb{R}^n$ on the set π. Note that the set

$$\pi = \left\{ x \in \mathbb{R}^n : \sum_{i=1}^n d_i|x_i| = 1, |\delta^+(x) - \delta^-(x)| \le \delta^0(x) \right\},$$

the projection is only onto some simplex cells of the set $X = \{x \in \mathbb{R}^n : \sum_{i=1}^n d_i|x_i| = 1\}$. To solve this challenging problem, we propose two steps: 1) we first relax the problem to be the projection onto X; 2) we then adjust the obtained projection to meet the requirement $|\delta^+(x) - \delta^-(x)| \le \delta^0(x)$.

For the first step, it is easy to prove the following lemma.

Lemma 3.1. *Let $b, c \in \mathbb{R}^n$ and c is a projection of b onto the set X, then $b_i c_i \ge 0, i = 1, \dots, n$, where b_i and c_i are the i-th components of b and c, respectively.*

This lemma suggests that we only need to consider the problem of projecting a vector b with $b_i \ge 0, i = 1, \dots, n$ onto the simplex cell of X with all nonnegative variables, which reads as follows:

$$\begin{aligned} \min \quad & \|y-b\|_2 \\ \text{s.t.} \quad & \sum_{i=1}^n d_i y_i = 1, \quad y_i \ge 0, \quad i = 1, \dots, n. \end{aligned} \tag{3.14}$$

For the simplicity of presentation, denote $Y = \{y = (y_1, \dots, y_n) \in \mathbb{R}^n : \sum_{i=1}^n d_i y_i = 1, y_i \ge 0, i = 1, \dots, n\}$ and $T = \{i \in V : y_i > 0\}$.

To solve this projection problem (3.14), let us first see its property.

Lemma 3.2. *Let* $b = (b_1, \ldots, b_n) \in \mathbb{R}^n$ *with* $b_i \geq 0, i = 1, \ldots, n$, $y = (y_1, \ldots, y_n) \in \mathbb{R}^n$ *be its projection onto the set* Y. *Then for any* $i, j \in T$ *with* $i \neq j$,

$$\frac{y_i - b_i}{d_i} = \frac{y_j - b_j}{d_j}. \tag{3.15}$$

Proof. If $i, j \in T$ and $i \neq j$, one considers the perturbation $\widehat{y}_\epsilon = (\widehat{y}_1, \ldots, \widehat{y}_n)$ of the projection $y = (y_1, \ldots, y_n)$ as follows: $\widehat{y}_k = y_k$ for $k \neq i, j$, $\widehat{y}_i = y_i - d_j\epsilon, \widehat{y}_j = y_j + d_i\epsilon$ for $\epsilon \in (-y_j/d_i, y_i/d_j)$. One can see that $\widehat{y}_\epsilon \in Y$ for $\epsilon \in (-y_j/d_i, y_i/d_j)$. As y is the projection of b onto Y, the function $h(\epsilon) = \|\widehat{y} - b\|_2^2$ attains its minimum value at 0, and hence $h'(0) = 2(y_i - b_i)(-d_j) + 2(y_j - b_j)d_i = 0$, which leads to the conclusion. $\square$

This lemma implies an explicit form of this projection of b onto Y, as stated in the following theorem.

Theorem 3.1. *Let* $b = (b_1, \ldots, b_n) \in \mathbb{R}^n$ *with* $b_i \geq 0, i = 1, \ldots, n$, $y = (y_1, \ldots, y_n) \in \mathbb{R}^n$ *be its projection onto the set* Y. *Then for any* $i \in T$,

$$y_i = b_i + d_i \frac{1 - \sum_{j \in T} b_j d_j}{\sum_{j \in T} d_j^2}. \tag{3.16}$$

Proof. Based on Lemma 3.2, there exists a constant c such that $(y_i - b_i)/d_i = c$ for any $i \in T$, and then $y_i = b_i + cd_i$ for $i \in T$. As $\sum_{i=1}^n d_i y_i = 1$, one gets $c = (1 - \sum_{j \in T} b_j d_j)/\sum_{j \in T} d_j^2$ and thus completes the proof. $\square$

The above theorem provides an explicit form of the projection onto the set Y once the index set T is known. However, how to determine this set? For this, we propose the idea of finding a simplex cell, a subset of the set Y such that all the components of the projection are positive. In fact, this goal simplex cell can be found using a recursive method. To this end, let us introduce the following lemmas.

Lemma 3.3. *Let* $b = (b_1, \ldots, b_n) \in \mathbb{R}^n$ *with* $b_i \geq 0, i = 1, \ldots, n$. *Then its projection* $y = (y_1, \ldots, y_n)$ *onto the hyperplane* $H = \{x \in \mathbb{R}^n : \sum_{i=1}^n d_i x_i = 1\}$ *can be expressed as* $y_i = b_i + cd_i, i = 1, \ldots, n$, *where* $c = (1 - \sum_{i=1}^n b_i d_i)/\sum_{i=1}^n d_i^2$.

This lemma can be easily justified using the method of Lagrange multiplier.

Note that in this lemma, the point is projected onto the whole hyperplane, not the simplex cell Y. In this lemma, if $y_i \geq 0$ for $i = 1, \ldots, n$, then this projection is inside the cell Y, which means the projection problem (3.14) is solved. If some component is negative, one needs to further project the above projection point onto some lower-dimensional simplex cell of Y. To elucidate this argument, we introduce the following lemma.

Lemma 3.4. *Let* $b = (b_1, \ldots, b_n) \in \mathbb{R}^n$ *with* $b_i \geq 0, i = 1, \ldots, n$, $y^* = (y_1^*, \ldots, y_n^*) \in \mathbb{R}^n$ *be its projection onto the hyperplane* $H = \{x \in \mathbb{R}^n : \sum_{i=1}^n d_i x_i = 1\}$. *Assume that* $y_1^* < 0$

and $y_i^* \geq 0$ *for* $i = 2, \ldots, n$. *Then the projection problem* (3.14) *is equivalent to the following one:*

$$\begin{aligned} \min \quad & \|y - y^*\|_2 \\ \text{s.t.} \quad & y_1 = 0, \quad \sum_{i=2}^{n} d_i y_i = 1, \quad y_i \geq 0, \quad i = 2, \ldots, n. \end{aligned} \tag{3.17}$$

Proof. For any point $x = (x_1, \ldots, x_n) \in Y \subset H$, by connecting it with y^*, one gets the line which can be represented as $y_i = y_i^* + t(x_i - y_i^*), i = 1, \ldots, n$ with t being a parameter. This line intersects the simplex cell Y at

$$\widehat{y} = \big(0, y_2^* + t^*(x_2 - y_2^*), \ldots, y_n^* + t^*(x_n - y_n^*)\big)$$

with $t^* = y_1^*/(y_1^* - x_1)$. In fact,

$$y_i^* + t^*(x_i - y_i^*) = \frac{(x_i y_1^* - x_1 y_i^*)}{(y_1^* - x_1)} \geq 0, \quad i = 2, \ldots, n.$$

As $y_1^* < 0$, t^* falls in $(0,1)$, that is, this intersection $\widehat{y}$ lies between x and y^*, and by the Pythagorean theorem,

$$\|x - b\|_2^2 = \|x - y^*\|_2^2 + \|y^* - b\|_2^2 \geq \|\widehat{y} - y^*\|_2^2 + \|y^* - b\|_2^2 = \|\widehat{y} - b\|_2^2.$$

Therefore, to find the projection of b onto Y, one only needs to find the projection onto its subset simplex cell $\{y = (y_1, \ldots, y_n) \in \mathbb{R}^n : y_1 = 0, \sum_{i=2}^{n} d_i y_i = 1, y_i \geq 0, i = 2, \ldots, n\}$. $\square$

Based on the same argument, one could also obtain the same conclusion if y_1^* is replaced by any other component while the remaining components are all non-negative. This lemma guarantees that the seek of the projection onto Y amounts to finding the projection onto a subset of Y.

In this lemma, only one component of y^* is assumed to be negative. In fact, based on similar arguments, we can prove the following theorem.

Theorem 3.2. *Let* $b = (b_1, \ldots, b_n) \in \mathbb{R}^n$ *with* $b_i \geq 0, i = 1, \ldots, n$ *and* $y^* = (y_1^*, \ldots, y_n^*) \in \mathbb{R}^n$ *be its projection onto the hyperplane* $H = \{x \in \mathbb{R}^n : \sum_{i=1}^{n} d_i x_i = 1\}$. *Let* A *and* B *be two disjoint subsets of* $V = \{1, \ldots, n\}$ *and* $A \cup B = V$. *Assume that* $y_i^* < 0$ *for any* $i \in A$ *and* $y_j^* \geq 0$ *for any* $j \in B$. *Then the projection problem* (3.14) *is equivalent to the following one:*

$$\begin{aligned} \min \quad & \|y - y^*\|_2 \\ \text{s.t.} \quad & y_i = 0, \quad \textit{for any} \quad i \in A \quad \textit{and} \quad \sum_{j \in B} d_j y_j = 1, \quad y_j \geq 0, \quad j \in B. \end{aligned} \tag{3.18}$$

Proof. Just as the proof in Lemma 3.4, one needs to show that the line connecting any point $x \in Y$ and y^* must pass through the lower-dimensional simplex cell $\{y \in \mathbb{R}^n : y_i = 0 \text{ for } i \in A, y_j \geq 0 \text{ for } j \in B, \sum_{j \in B} d_j y_j = 1\}$.

Suppose that $A = \{i_1, \ldots, i_k\}$ and $B = \{j_1, \ldots, j_{n-k}\}$. Let us first consider any point $x = (x_1, \ldots, x_n) \in Y$ with $x_i > 0, i = 1, \ldots, n$. Without loss of generality, assume that

$$\frac{y^*_{i_1}}{x_{i_1}} \leq \frac{y^*_{i_2}}{x_{i_2}} \leq \cdots \leq \frac{y^*_{i_k}}{x_{i_k}}.$$

As the line connecting x and y^* can be expressed as

$$y_i = y^*_i + t(x_i - y^*_i), \quad i = 1, \ldots, n$$

with t being a parameter, it intersects the hyper-plane $\{(y_1, \ldots, y_n) : y_{i_1} = 0, \sum_{j \neq i_1} d_j y_j = 1\}$ at $z = (z_1, \ldots, z_n)$, where

$$\begin{aligned}
&z_{i_1} = 0, \quad z_{i_p} = y^*_{i_p} + t^*(x_{i_p} - y^*_{i_p}), && p = 2, \ldots, k, \\
&z_{j_q} = y^*_{j_q} + t^*(x_{j_q} - y^*_{j_q}), && q = 1, \ldots, n-k,
\end{aligned}$$

and $t^* = y^*_{i_1}/(y^*_{i_1} - x_{i_1})$. Then

$$\begin{aligned}
z_{i_p} &= \frac{x_{i_1} x_{i_p}}{y^*_{i_1} - x_{i_1}} \left(\frac{y^*_{i_1}}{x_{i_1}} - \frac{y^*_{i_p}}{x_{i_p}} \right), \quad p = 2, \ldots, k, \\
z_{j_q} &= \frac{x_{i_1} x_{j_q}}{y^*_{i_1} - x_{i_1}} \left(\frac{y^*_{i_1}}{x_{i_1}} - \frac{y^*_{j_q}}{x_{j_q}} \right), \quad q = 1, \ldots, n-k.
\end{aligned}$$

By the above assumption, one has $z_{i_p} \geq 0, p = 2, \ldots, k$. Note that $y^*_{j_q} \geq 0$ for $q = 1, \ldots, n-k$ and $y^*_{i_1} < 0$, one also has $z_{j_q} \geq 0, q = 1, \ldots, n-k$. Therefore, this intersection point z lies in the lower-dimensional simplex cell $S_{i_1} := \{(y_1, \ldots, y_n) : y_{i_1} = 0, \sum_{j \neq i_1} d_j y_j = 1, y_j \geq 0\} \subset Y$, which shows that $\|x - y^*\|_2 \geq \|z - y^*\|_2$. Similarly, for any $x \in Y$ with $x_i > 0, i = 1, \ldots, n$, if $i_s = \mathrm{argmin}_{p \in A}(y^*_p / x_p)$, then the line connecting y^* and x intersects at a point in the simplex cell S_{i_s}.

Based on the above argument, to get the projection of y^* onto Y, one only needs to find its projections onto its sub-cells $S_{i_p}, p = 1, \ldots, k$ and compare them to get the desired one. In what follows, we intend to show that the projection onto these sub-cells is equivalent to the projection onto their intersection cell

$$S = \left\{ y = (y_1, \ldots, y_n) : y_i = 0, \text{ for } i \in A, y_j \geq 0, \text{ for } j \in B, \sum_{j \in B} d_j y_j = 1 \right\}.$$

To justify this claim, let us consider the projection of y^* onto the sub-cell S_{i_1}. Assume that $y = (y_1, \ldots, y_n) \in S_{i_1}$ is the projection point, and then $y_{i_1} = 0$ and $y_j \geq 0$ for all j except i_1. We want to show that $y_{i_p} = 0$ for $p = 2, \ldots, k$. If $y_{i_2} > 0$, for any $j_q \in B$, we consider a perturbation point $\widehat{y}^\epsilon = (\widehat{y}^\epsilon_1, \ldots, \widehat{y}^\epsilon_n)$, where $\widehat{y}^\epsilon_{i_2} = y_{i_2} - \epsilon$, $\widehat{y}^\epsilon_{j_q} = y_{j_q} + d_{i_2}/d_{j_q}\epsilon$, and $\widehat{y}^\epsilon_i = y_i$ for all $i \neq i_2, j_q$. It is easy to see that $\widehat{y}^\epsilon \in S_{i_1}$ for each $\epsilon \in [0, y_{i_2})$. Then the function $f(\epsilon) = \|\widehat{y}^\epsilon - y^*\|^2_2$ takes its minimum value at $\epsilon = 0$, therefore $f'(0) = -2(y_{i_2} - y^*_{i_2}) + 2(y_{j_q} - y^*_{j_q})d_{i_2}/d_{j_q} \geq 0$, and then $y_{j_q} - y^*_{j_q}/d_{j_q} \geq y_{i_2} - y^*_{i_2}/d_{i_2}$ for any $j_q \in B$.

Set $\lambda = (y_{i_2} - y^*_{i_2})/d_{i_2}$, then $\lambda > 0$ and $y_{j_q} - y^*_{j_q} \geq \lambda d_{j_q}$ for all $j_q \in B$. Then

$$\begin{aligned}1 &= \sum_{i\in A} d_i y_i + \sum_{j\in B} d_j y_j \\ &\geq \sum_{i\in A} d_i y_i + \sum_{j\in B} d_j(y^*_j + \lambda d_j) \\ &\geq d_{i_2} y_{i_2} + 1 - \sum_{i\in A} d_i y^*_i + \lambda \sum_{j\in B} d_j^2. \end{aligned} \tag{3.19}$$

Therefore, as $y^*_i < 0$ for all $i \in A$, one gets

$$0 > -d_{i_2} y_{i_2} \geq -\sum_{i\in A} d_i y^*_i + \lambda \sum_{j\in B} d_j^2 > 0,$$

which leads to a contradiction. This justifies that $y_{i_2} \leq 0$.

Similarly, one can show that $y_{i_p} = 0$ for $p = 3, \ldots, k$. This means that the projection of y^* onto the simplex cell S_{i_1} is in fact inside an even lower-dimensional cell

$$S = \left\{ y = (y_1, \ldots, y_n) : y_i = 0, \text{ for } i \in A, y_j \geq 0, \text{ for } j \in B, \sum_{j\in B} d_j y_j = 1 \right\},$$

which completes the proof. $\square$

This theorem converts the projection problem (3.14) to be a similar one (3.18) but onto a lower-dimensional simplex cell if some component of y^* are negative. In fact, the above new problem (3.18) might not guarantee that all the components of the projection are non-negative. If this occurs, one may repeatedly use the above theorem until all the components are non-negative. Using this procedure, we can find the exact solution for the problem (3.14), which leads to the solution of the original projection problem that projects any point onto the set X. We summarize it as the following theorem.

Theorem 3.3. *For any point $b = (b_1, \ldots, b_n) \in \mathbb{R}^n$, suppose $y = (y_1, \ldots, y_n) \in \mathbb{R}^n$ is the projection of the point $\widetilde{b} = (|b_1|, \ldots, |b_n|)$ onto the set*

$$Y = \left\{ y = (y_1, \ldots, y_n) \in \mathbb{R}^n : \sum_{i=1}^n d_i y_i = 1, y_i \geq 0, i = 1, \ldots, n \right\},$$

then the projection of b onto

$$X = \left\{ x \in \mathbb{R}^n : \sum_{i=1}^n d_i |x_i| = 1 \right\}$$

is the point $\widetilde{y} = (\mathrm{sign}(b_1) y_1, \ldots, \mathrm{sign}(b_n) y_n)$, where ‘sign’ *is the* sign *function.*

This theorem is a direct implication of Lemma 3.1.

Having solved the problem of projecting any point onto the set X, we then need to find the projection onto $\pi = \{x \in X : |\delta^+(x) - \delta^-(x)| \leq \delta^0(x)\}$, which is a proper subset of X. If the projection is already inside π, then we find the goal projection, in other words, the problem (3.13) and thus the sub-problem (3.6) is completely solved. However, this might not always be the case and we need to re-project the original point onto some appropriate simplex cell. To determine this cell, we consider an equivalent form of the sign constraint as follows.

Lemma 3.5 ([7, Lemma 2]). *$x \in \pi$ if and only if $\delta^+(x) \leq d/2$ and $\delta^-(x) \leq d/2$, where $d = \sum_{i=1}^n d_i$.*

Suppose that $\widetilde{y}$ is the projection of a point b onto X. Based on this lemma, if both $\delta^+(\widetilde{y})$ and $\delta^-(\widetilde{y})$ are not larger than $d/2$, then $\widetilde{y}$ is the goal projection onto π. Otherwise, we need to adjust the projection to meet the sign constraint to do the projection. To be specific, denote

$$\iota_+ = \{i \in V : \widetilde{y}_i > 0\}, \quad \iota_- = \{i \in V : \widetilde{y}_i < 0\}.$$

If $\delta^+(\widetilde{y}) > d/2$, we need to find a subset of $\iota_+^1 \subset \iota_+$ such that the total degree $\sum_{i \in \iota_+^1} d_i \leq d/2$. Then the index set $\omega = \iota_+^1 \cup \iota_-$ forms a sub-simplex cell defined as follows:

$$\Gamma_{\omega,m} = \left\{ y = \sum_{i \in \omega} y_i m_i \mathbf{e}_i : \sum_{i \in \omega} d_i y_i = 1, y_i > 0, m_i = \operatorname{sign}(\widetilde{y}_i), \forall i \in \omega \right\}, \tag{3.20}$$

where $\mathbf{e}_i$ represents the i-th standard basis vector in $\mathbb{R}^n$.

To find the goal projection onto π, we encounter two problems:

1. How to determine all of those subsets of ι_+^1 of ι_+ to form admissible sub-simplex cells?

2. Once all those sub-simplex cells are obtained, how to get the desired one?

The first problem is in fact the subset sum problem [16]. If $\delta^+(\widetilde{y}) > d/2$, we denote $\mathrm{gap} = \delta^+(\widetilde{y}) - d/2$, and find all the subsets of ι_+, each of which has the total degree that lies in $[\mathrm{gap}, \delta^+(\widetilde{y})]$. Suppose that $\iota_+^2 \subset \iota_+$ with $\sum_{i \in \iota_+^2} d_i \geq \mathrm{gap}$, then $\iota_+^1 = \iota_+ \setminus \iota_+^2$ and $\omega = \iota_+^1 \cup \iota_-$ gives an index set for an admissible sub-simplex cell $\Gamma_{\omega,m}$.

As for the second problem, since $\widetilde{y}$ is the projection of b onto X, the projection of b onto any simplex cell of X gives an equal or larger value than $\|\widetilde{y} - b\|^2$. As a result, one only needs to compare those value increments over all those admissible simplex-cells and choose the one with the smallest increment. In fact, to determine the increment for each sub-simplex cell, we have the following theorem.

Theorem 3.4. *Assume that $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ is the projection of the point $b = (b_1, \ldots, b_n) \in \mathbb{R}^n$ with $b_i \geq 0, i = 1, \ldots, n$ onto the set X and $\iota = \{i_1, \ldots, i_k\} = \{i \in V : x_i > 0\}$. If $\omega \subset \iota$ and x_ω is the projection of b onto the simplex cell*

$$\Gamma_\omega = \left\{ y = \sum_{i \in \omega} y_i \mathbf{e}_i : \sum_{i \in \iota} d_i y_i = 1, y_i > 0, \forall i \in \omega \right\},$$

then

$$\|x_\omega - b\|_2^2 - \|x_\iota - b\|_2^2 = \sum_{i\in\iota\setminus\omega} x_i^2 + \frac{(\sum_{i\in\iota\setminus\omega} d_i x_i)^2}{\sum_{i\in\iota} d_i^2 - \sum_{i\in\iota\setminus\omega} d_i^2}. \tag{3.21}$$

Proof. By Theorem 3.1, one has for any $i \in \iota$, $x_i = b_i + td_i$ with $t = (1 - \sum_{i\in\iota} d_i b_i)/\sum_{i\in\iota} d_i^2$.

Similarly, for any $i \in \omega$, $x_{\omega,i} = b_i + \hat{t}d_i$ with $\hat{t} = (1 - \sum_{i\in\omega} d_i b_i)/\sum_{i\in\omega} d_i^2$, and for any $i \in \iota \setminus \omega$, $x_{\omega,i} = 0$.

As

$$\sum_{i\in\iota} d_i(b_i + td_i) = 1, \quad \sum_{i\in\omega} d_i(b_i + \hat{t}d_i) = 1,$$

one gets

$$\sum_{i\in\iota\setminus\omega} d_i x_i + \sum_{i\in\omega} d_i^2(t - \hat{t}) = 0,$$

and then

$$\hat{t} - t = \frac{\sum_{i\in\iota\setminus\omega} d_i x_i}{\sum_{i\in\omega} d_i^2} =: s.$$

Therefore,

$$\begin{aligned}
&\|x_\omega - b\|_2^2 - \|x_\iota - b\|_2^2 \\
&= \sum_{i\in\iota\setminus\omega} b_i^2 + \sum_{i\in\omega} \hat{t}^2 d_i^2 - \sum_{i\in\iota} t^2 d_i^2 \\
&= \sum_{i\in\iota\setminus\omega} b_i^2 + \sum_{i\in\omega} (\hat{t}^2 - t^2) d_i^2 - \sum_{i\in\iota\setminus\omega} t^2 d_i^2 \\
&= \sum_{i\in\iota\setminus\omega} (x_i - td_i)^2 + \sum_{i\in\omega} (s^2 + 2ts) d_i^2 - \sum_{i\in\iota\setminus\omega} t^2 d_i^2 \\
&= \sum_{i\in\iota\setminus\omega} (x_i^2 - 2td_i x_i) + \sum_{i\in\omega} (s^2 + 2ts) d_i^2 \\
&= \sum_{i\in\iota\setminus\omega} x_i^2 + \sum_{i\in\omega} s^2 d_i^2 = \sum_{i\in\iota\setminus\omega} x_i^2 + \frac{(\sum_{i\in\iota\setminus\omega} d_i x_i)^2}{\sum_{i\in\iota} d_i^2 - \sum_{i\in\iota\setminus\omega} d_i^2},
\end{aligned}$$

which gives the conclusion. ◻

In a nutshell, for any given point b, one gets its projection x onto X, and checks whether $\delta^+(x)$ and $\delta^-(x)$ are both equal or less than $d/2$. If $\delta^+(x) > d/2$, one sets $\iota_+ = \{i \in V : x_i > 0\}$ and $\iota_- = \{i \in V : x_i < 0\}$, and finds all the subsets of ι_+ such that each of them has a total degree equal to or greater than $\mathrm{gap} = \delta^+(x) - d/2$. Let $\iota_+^2 \subset \iota_+$ with $\sum_{i\in\iota_+^2} d_i \geq \mathrm{gap}$ be one of these subsets, then one calculates the corresponding increment $\|x_\omega - \hat{b}\|_2^2 - \|x_\iota - \hat{b}\|_2^2$, where $\hat{b} = (|b_1|, \ldots, |b_n|)$, $\omega = \iota_+ \setminus \iota_+^2$, $\iota = \iota_+ \cup \iota_-$, and x_ω and x_ι are the projection of $\hat{b}$ onto X. This is because the above theorem holds for the projection of a point with non-negative entries.

Based on the above arguments, the sub-problem (3.6) of z is solved, that is, one finds the projection of any point in $\mathbb{R}^n$ onto the set π.

As a summary, we have discussed the methods to solve the three sub-problems (3.4)-(3.6). Therefore, once the initial $(x^0, y^0, z^0; \lambda_1^0, \lambda_2^0)$ is given, as shown in (3.4)-(3.8), a sequence $\{(x^k, y^k, z^k; \lambda_1^k, \lambda_2^k)_{k=1}^\infty\}$ can be generated. If this sequence converges to the saddle point of the augmented Lagrangian functional (3.3), then the original optimization problem (3.1) is solved.

Even though the above ADMM algorithm provides a viable tool to find the Cheeger cut for any given undirected and unweighted graph, there still remain two issues. The first issue is whether the algorithm could give the global minimizer of the optimization problem (3.1). Note that the optimization problem is defined over a non-convex set, and the proposed algorithm does not ensure the global minimizer, but merely a local minimizer. The second issue lies in the computational complexity of solving the subset sum problem. In fact, it can also be regarded as an NP-hard problem.

To deal with the first problem, in this paper, we propose applying the developed ADMM with different initial guess (x^0, y^0, z^0) for a given graph and comparing all those obtained cuts to determine the Cheeger cut. Specifically, among all the obtained cuts, the cut with the smallest value of

$$\mathrm{cut}(S, S^c) = \frac{|\partial S|}{\min\{\mathrm{Vol}(S), \mathrm{Vol}(S^c)\}}$$

is the desired cut. As shown in the numerical experiments section, this procedure could help find many optimal solutions to the optimization problem (3.1) or Cheeger cuts for a given graph.

Before discussing how to solve the second problem, we consider a corollary of Theorem 3.4.

Corollary 3.1. *Assume that $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ is the projection of the point $b = (b_1, \ldots, b_n) \in \mathbb{R}^n$ with $b_i \geq 0, i = 1, \ldots, n$ onto the set X and $\iota = \{i_1, \ldots, i_k\} = \{i \in V : x_i > 0\}$. For any $j \in \{1, \ldots, k\}$, let $\omega_j = \iota \setminus \{i_j\}$ and x_{ω_j} be the projection of b onto the simplex cell $\Gamma_{\omega_j, m}$, then*

$$\|x_{\omega_j} - b\|_2^2 - \|x_\iota - b\|_2^2 = \frac{D x_{i_j}^2}{D - d_{i_j}^2}, \tag{3.22}$$

where $D = \sum_{i \in \iota}^n d_i^2$.

To elucidate the method of dealing with the second problem, as discussed before, we assume x is the projection of b onto X and $\delta^+(x) = \{i_1, \ldots, i_k\}$ with $\delta^+(x) > d/2$. Note that b could be any vector in $\mathbb{R}^n$. To apply the above corollary, one needs to consider the vector $\hat{b} = (|b_1|, \ldots, |b_n|)$. It is easy to see that the vector $(|x_1|, \ldots, |x_n|)$ is the projection of $\hat{b}$. Denote $\iota = \{i_1, \ldots, i_k\} \cup \delta^-(x)$ and $\omega_j = \iota \setminus \{i_j\}, j = 1, \ldots, k$. We first calculate all the increments $g_j = D x_{i_j}^2 / (D - d_{i_j}^2), j = 1, \ldots, k$, and then sort $\{g_j\}, j = 1, \ldots, k$. For simplicity, let us assume $g_1 \leq g_2 \leq \cdots \leq g_k$. If $p = \inf\{q \in$

$\mathbb{N} : \sum_{s=1}^{q} d_{i_s} \geq \delta^+(x) - d/2\}$, then we get the index set $\omega = \{i_{p+1}, \ldots, i_k\} \cup \delta^-(x)$ and the goal sub-simplex cell is $\Gamma_{\omega,m}$ as defined in (3.20). The new projection $\hat{x}$ onto this simplex cell surely satisfies the sign constraints $\delta^+(\hat{x}) \leq d/2$ and $\delta^-(\hat{x}) \leq d/2$.

The above procedure does not ensure that the projection of b onto the set π can be found. However, it does give some projection of b onto π. Note that the proposed algorithm is iterative, and one could solve the sub-problem (3.6) approximately. In fact, by using this procedure, one could dramatically reduce the computational cost for solving the sub-problem of z.

4. Convergence analysis

In this section, we discuss the property of the sequence $\{(x^k, y^k, z^k; \lambda_1^k, \lambda_2^k)\}$ generated by Algorithm 3.1. Note that the original optimization problem (3.1) is non-differentiable and non-convex, especially, the domain consisting of more than $3^{[(n+1)/2]} - 1$ simplex cells of different dimensions. This raises a challenging problem for the convergence study. In what follows, we intend to show that the above generated sequence is bounded under some mild condition on the parameters r_1 and r_2, and therefore, there must exist a subsequence of $\{(x^k, y^k, z^k; \lambda_1^k, \lambda_2^k)\}$ that is convergent.

Let $(x^0, y^0, z^0; \lambda_1^0, \lambda_2^0)$ be initially given and $(x^k, y^k, z^k; \lambda_1^k, \lambda_2^k)$ is generated by Algorithm 3.1. To show the boundedness of this sequence, we first introduce several lemmas.

Lemma 4.1. *For any $(x^0, y^0, z^0; \lambda_1^0, \lambda_2^0)$, in the sequence $\{(x^k, y^k, z^k; \lambda_1^k, \lambda_2^k)\}$, the Lagrange multiplier λ_1^k is bounded for any $k \geq 1$, specifically, $\|\lambda_1^k\|_1 \leq m/r_1$. Moreover, $\|y^{k+1}\|_2^2 \leq \|Bx^{k+1}\|_2^2$ for any $k \geq 0$.*

Proof. As discussed in (3.5),

$$y^{k+1} = \text{argmin}_y \mathcal{L}\big(x^{k+1}, y, z^k; \lambda_1^k, \lambda_2^k\big).$$

For this sub-problem of y, as discussed above, it has a closed-form solution. In fact, denote

$$a = Bx^{k+1} - \lambda_1^k = [a_1 \ldots a_m]^T,$$

then if $|a_i| \leq 1/r_1$, $y_i^{k+1} = 0$, and otherwise

$$y_i^{k+1} = a_i - \frac{a_i}{r_1|a_i|}, \quad i = 1, \ldots, m.$$

Note that

$$\lambda_1^{k+1} = y^{k+1} - Bx^{k+1} + \lambda_1^k = y^{k+1} - a,$$

then for $i \in \{1, \ldots, m\}$, the i-th component of λ_1^{k+1} reads either $-a_i$ when $|a_i| \leq 1/r_1$ or $-a_i/(r_1|a_i|)$ when $|a_i| > 1/r_1$. For both cases, one has $|\lambda_{1,i}^{k+1}| \leq 1/r_1$. Therefore, $\|\lambda_1^{k+1}\|_1 \leq m/r_1$.

As $a = Bx^{k+1} - \lambda_1^k$, one has

$$|a_i| \le \left|(Bx^{k+1})_i\right| + \left|\lambda_{1,i}^k\right| \le \left|(Bx^{k+1})_i\right| + \frac{1}{r_1}.$$

By using the expression of y_i^{k+1}, we get

$$\|y^{k+1}\|_2^2 = \sum_{|a_i|>1/r_1} \left(1 - \frac{1}{r_1|a_i|}\right)^2 |a_i|^2 = \sum_{|a_i|>1/r_1} \left(|a_i| - \frac{1}{r_1}\right)^2. \tag{4.1}$$

Note that if $|a_i| > 1/r_1$, we have

$$0 < |a_i| - \frac{1}{r_1} \le \left|(Bx^{k+1})_i\right|,$$

which shows that

$$\|y^{k+1}\|_2^2 \le \sum_{|a_i|>1/r_1} \left|(Bx^{k+1})_i\right|^2 \le \|Bx^{k+1}\|_2^2. \tag{4.2}$$

The proof is complete. □

Lemma 4.2. *For the generated sequence* $\{(x^k, y^k, z^k; \lambda_1^k, \lambda_2^k)\}$, *the following identity holds for any* $k \ge 1$*:*

$$(r_1 B^T B + r_2 I_n)x^{k+1} = r_1 B^T(2y^k - y^{k-1}) + r_2(2z^k - z^{k-1}).$$

Proof. If $(x^k, y^k, z^k; \lambda_1^k, \lambda_2^k)$ is given, one has $x^{k+1} = \operatorname{argmin}_x \mathcal{L}(x, y^k, z^k; \lambda_1^k, \lambda_2^k)$. Then from (3.10), x^{k+1} satisfies the following equation:

$$-r_1 B^T\big(y^k - Bx^{k+1} + \lambda_1^k\big) + r_2\big(x^{k+1} - z^k + \lambda_2^k\big) = 0.$$

Similarly, for x^k, one has

$$-r_1 B^T\big(y^{k-1} - Bx^k + \lambda_1^{k-1}\big) + r_2\big(x^k - z^{k-1} + \lambda_2^{k-1}\big) = 0.$$

One subtracts the second equation from the first one and gets

$$\begin{aligned} &- r_1 B^T \left[y^k - Bx^{k+1} + \lambda_1^k - (y^k - Bx^k + \lambda_1^{k-1}) + y^k - y^{k-1}\right] \\ &\quad + r_2 \left[x^{k+1} - z^k + \lambda_2^k - (x^k - z^k + \lambda_2^{k-1}) + z^{k-1} - z^k\right] = 0. \end{aligned}$$

As

$$y^k - Bx^k + \lambda_1^{k-1} = \lambda_1^k, \quad x^k - z^k + \lambda_2^{k-1} = \lambda_2^k,$$

the above equation leads to the conclusion. □

With the above lemmas, in what follows, we intend to show that the sequence $\{(x^k, y^k, z^k; \lambda_1^k, \lambda_2^k)\}_{k=1}^\infty$ is bounded and must have a convergent subsequence.

Theorem 4.1. *For any initial guess* $(x^0, y^0, z^0; \lambda_1^0, \lambda_2^0)$, *if in Algorithm* 3.1 *the two penalty parameters* $r_1, r_2 > 0$ *satisfy the following inequality:*

$$\frac{\sqrt{16\tau^2 + 4\tau} + 4\tau}{2} < 1, \tag{4.3}$$

where

$$\tau = \frac{4r_1}{(r_1/2 + r_2/(4d_{\max}))}, \quad d_{\max} = \max\{d_1, \ldots, d_n\},$$

then the generated sequence $\{(x^k, y^k, z^k; \lambda_1^k, \lambda_2^k)\}_{k=1}^{\infty}$ *is bounded and there exists a convergent subsequence.*

Proof. From Lemma 4.2,

$$(r_1 B^T B + r_2 I_n) x^{k+1} = r_1 B^T (2y^k - y^{k-1}) + r_2 (2z^k - z^{k-1}).$$

By multiplying $(x^{k+1})^T$ from the left of this equation, one gets

$$\begin{aligned}
&r_1 \|Bx^{k+1}\|_2^2 + r_2 \|x^{k+1}\|_2^2 \\
=\ & r_1 (Bx^{k+1})^T (2y^k - y^{k-1}) + r_2 (x^{k+1})^T (2z^k - z^{k-1}) \\
\le\ & r_1 \left(\frac{1}{2} \|Bx^{k+1}\|_2^2 + \frac{1}{2} \|2y^k - y^{k-1}\|_2^2 \right) + r_2 \left(\frac{1}{2} \|x^{k+1}\|_2^2 + \frac{1}{2} \|2z^k - z^{k-1}\|_2^2 \right),
\end{aligned}$$

and therefore

$$\frac{r_1}{2} \|Bx^{k+1}\|_2^2 + \frac{r_2}{2} \|x^{k+1}\|_2^2 \le \frac{r_1}{2} \|2y^k - y^{k-1}\|_2^2 + \frac{r_2}{2} \|2z^k - z^{k-1}\|_2^2. \tag{4.4}$$

For any $x \in \mathbb{R}^n$,

$$\|Bx\|_2^2 = \sum_{i \sim j} (x_i - x_j)^2 \le \sum_{i \sim j} 2\big(x_i^2 + x_j^2\big) \le 2d_{\max} \|x\|_2^2.$$

This shows that $\|x\|_2^2 \ge \|Bx\|_2^2 / (2d_{\max})$ for any $x \in \mathbb{R}^n$, and therefore

$$\frac{r_1}{2} \|Bx^{k+1}\|_2^2 + \frac{r_2}{2} \|x^{k+1}\|_2^2 \ge \left(\frac{r_1}{2} + \frac{r_2}{4d_{\max}} \right) \|Bx^{k+1}\|_2^2, \tag{4.5}$$

and by Lemma 4.1

$$\begin{aligned}
&\left(\frac{r_1}{2} + \frac{r_2}{4d_{\max}} \right) \|y^{k+1}\|_2^2 \\
\le\ & \left(\frac{r_1}{2} + \frac{r_2}{4d_{\max}} \right) \|Bx^{k+1}\|_2^2 \\
\le\ & \frac{r_1}{2} \|Bx^{k+1}\|_2^2 + \frac{r_2}{2} \|x^{k+1}\|_2^2 \\
\le\ & \frac{r_1}{2} \|2y^k - y^{k-1}\|_2^2 + \frac{r_2}{2} \|2z^k - z^{k-1}\|_2^2 \\
\le\ & \frac{r_1}{2} \left(8\|y^k\|_2^2 + 2\|y^{k-1}\|_2^2 \right) + 5r_2,
\end{aligned} \tag{4.6}$$

where we use the fact that $z^k \in \pi$ and then $\|z^k\|_2 \leq \|z^k\|_1 \leq \sum_{i=1}^n d_i|z_i^k| = 1$ for any $k \geq 1$ in the last inequality. This leads to the following inequality:

$$\|y^{k+1}\|_2^2 \leq 4\tau\|y^k\|_2^2 + \tau\|y^{k-1}\|_2^2 + c, \tag{4.7}$$

where

$$\tau = \frac{4r_1}{(r_1/2 + r_2/(4d_{\max}))}, \quad c = \frac{5r_2}{(r_1/2 + r_2/(4d_{\max}))}.$$

The above inequality can be rewritten for any $k \geq 1$ as follows:

$$\|y^{k+1}\|_2^2 + s\|y^k\|_2^2 \leq t\left(\|y^k\|_2^2 + s\|y^{k-1}\|_2^2\right) + c, \tag{4.8}$$

where

$$s = \frac{\sqrt{16\tau^2 + 4\tau} - 4\tau}{2}, \quad t = \frac{\sqrt{16\tau^2 + 4\tau} + 4\tau}{2}.$$

Denote $a_k := \|y^{k+1}\|_2^2 + s\|y^k\|_2^2$, then one has $a_k \leq ta_{k-1} + c$ for any $k > 2$. It is easy to see that if $t \in (0,1)$

$$a_k \leq t^{k-1}a_1 + c\sum_{i=0}^{k-2} t^i \leq a_1 + \frac{c}{1-t} \tag{4.9}$$

for any $k > 2$.

Therefore, if the two parameters r_1, r_2 are chosen such that $0 < t < 1$, specifically,

$$t = \frac{\sqrt{16\tau^2 + 4\tau} + 4\tau}{2} < 1, \tag{4.10}$$

one then gets

$$\|y^{k+1}\|_2^2 \leq \|y^{k+1}\|_2^2 + s\|y^k\|_2^2 = a_k \leq a_1 + \frac{c}{1-t} \tag{4.11}$$

for any $k > 2$, which shows that the sequence $\{y^k\}_{k=0}^\infty$ is bounded. In fact, this can be easily achieved by setting $r_2 > 0$ large enough so that τ is a small number.

By using (4.4), we can show that $\{x^k\}_{k=0}^\infty$ is bounded. Note from (3.10), one has

$$\lambda_2^k = \frac{r_1}{r_2}B^T\left(y^k - Bx^{k+1} + \lambda_1^k\right) - x^{k+1} + z^k. \tag{4.12}$$

Since each of $x^k, y^k, z^k, \lambda_1^k$ is bounded, λ_2^k is also bounded for all $k \geq 1$.

In summary, if $r_1, r_2 > 0$ are chosen such that (4.10) holds, for any initial guess $(x^0, y^0, z^0; \lambda_1^0, \lambda_2^0)$, the generated sequence $\{(x^k, y^k, z^k; \lambda_1^k, \lambda_2^k)\}_{k=1}^\infty$ by Algorithm 3.1 is bounded so that there exists a convergent subsequence, which finishes the proof. $\square$

The above theorem guarantees that there will not be any blow-up by applying the proposed algorithm.

5. Numerical experiments

In this section, we report our numerical experiments by applying the proposed algorithm for typical undirected and unweighted graphs.

As discussed above, the original optimization problem (3.1) is non-convex, and the result of our algorithm depends on the initial guess $(x^0, y^0, z^0; \lambda_1^0, \lambda_2^0)$. For each test, we fix $\lambda_1^0 = 0$ and $\lambda_2^0 = 0$, and choose x^0 such that each of its entries is from the uniform distribution in the interval $(0, 1)$. Once x^0 is chosen, we set $y^0 = Bx^0$ and $z^0 = x^0$. Hence, the initial guess is fully determined by x^0.

For a given undirected and unweighted graph, using different initial guess x^0 can help find the optimal solution, moreover, it also helps discover different optimal solutions when the uniqueness of optimal solution fails to exist. Therefore, in our experiments, for each given graph, we apply the proposed algorithm with a group of randomized initial vectors x^0 and terminate each of the iterative process with a fixed iteration number. For each test, we determine the cut value $\mathrm{cut}(S, S^c)$ based on the obtained variable z. We utilize the variable z instead of x for determining the cut because the variable z is indeed inside a simplex cell. Specifically, for each test, we set $S = \{i \in V : z_i \geq 0\}$ or $S = \{i \in V : z_i > 0\}$ and calculate the two cut values $\mathrm{cut}(S, S^c)$, the lower of which gives the associated cut for this test. We then compare all the cut values from these tests, and the lowest cut value leads to the Cheeger cut.

Note that for each test, we terminate the iterative process with a fixed iteration number because what we want is the sign of the entries of x, instead of the real minimizer of (3.1), which also helps save lots of computational efforts.

We first apply the proposed algorithm for the Cockroach graphs C_{4k} as shown in Fig. 1. For each Cockroach graph with $k \in \{2, \ldots, 10\}$, we test the algorithm using 40 different initial guess x^0, each test runs 2000 iterations, and then get the cut determined by the obtained z. The smallest cut values $\mathrm{cut}(S, S^c)$ and the associated cuts for these Cockroach graphs are listed in Table 1. In fact, these obtained smallest cut values are exactly the same as the associated Cheeger cut constants, which can be calculated by considering all the possible sets S of V.

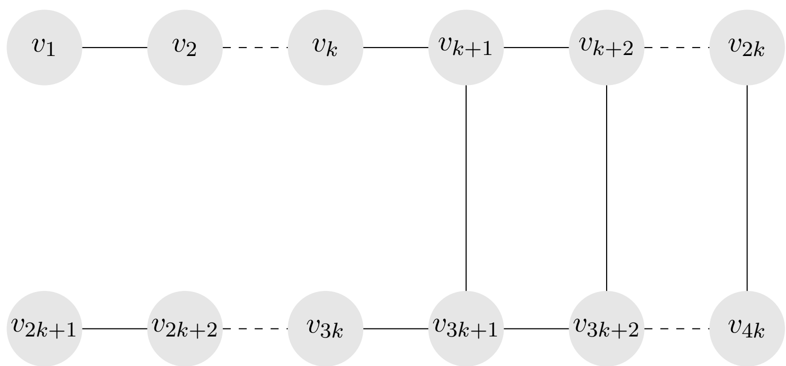

Figure 1: The cockroach graph C_{4k} with $4k$ nodes.

From Table 1, interestingly, one can see that the Cockroach graphs C_{12} and C_{36} have more than one Cheeger cut. This justifies that the proposed procedure of using different initial guess x^0 helps find more optimal solutions. In Figs. 2 and 3, we present the plot of $\mathrm{Cut}(S, S^c)$ versus tests and also the obtained Cheeger cuts. From the plots, even with 40 tests, many of them lead to the desired Cheeger cut. Note that in the plots of Cheeger cuts, the nodes with the same color (red or blue) belong to the same group.

In Fig. 4, we seek the Cheeger cut for the Petersen graph. The plots show the smallest value $\mathrm{cut}(S, S^c) = 1/3$, which corresponds to the Cheeger cut and again there exist different Cheeger cuts. We here only present a few typical Cheeger cuts, and the rotation of those presented ones could introduce more new Cheeger cuts. This fact also demonstrates that there could exist more than one optimal solution to the original optimization problem (3.1).

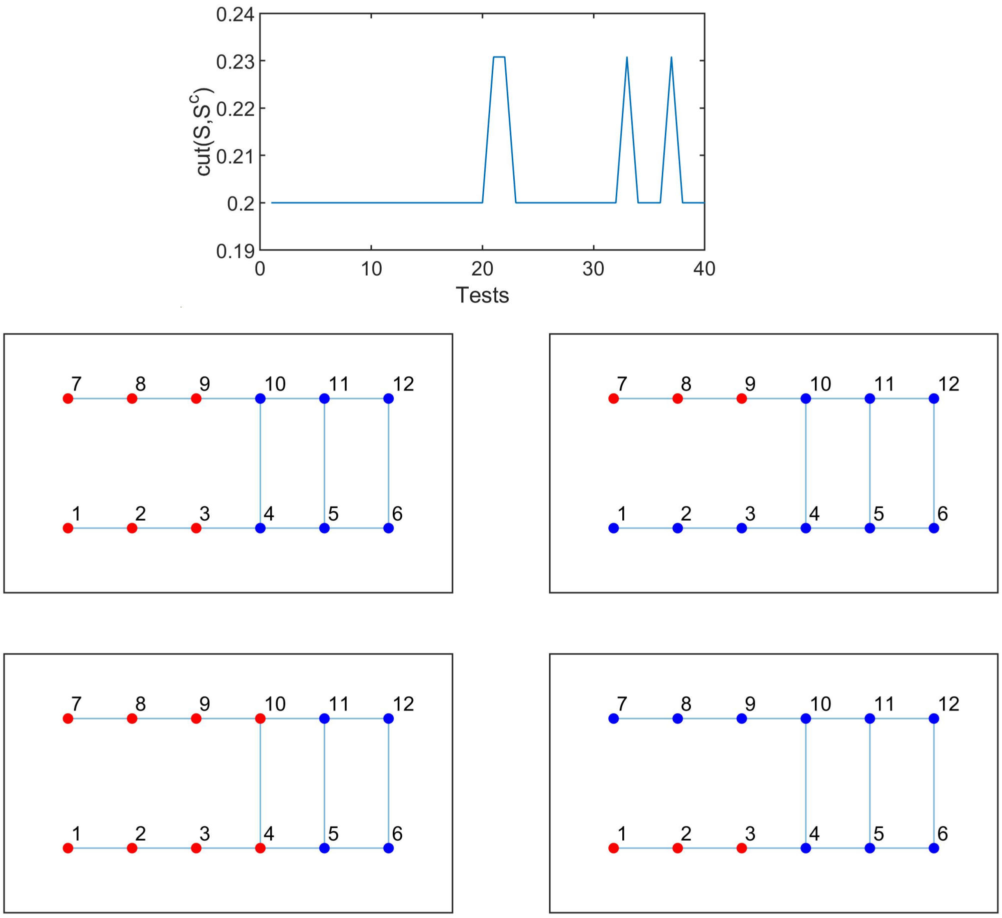

Figure 2: The plot of the cut values $\mathrm{cut}(S, S^c)$ for 40 tests with different initial value for x^0 and the obtained different Cheeger cuts for the Cockroach graph C_{12}. In this experiment, we set the parameters $r_1 = r_2 = 100$.

We then consider the well-known Zachary's karate club network [25]. This club consists of 34 members. To represent this social network, one assigns an edge between two members if they are friends. As a result, there are 34 nodes and 78 edges in the resulting unweighted and undirected graph. As shown in Fig. 5, we obtain the desired

Table 1: The obtained Cheeger constants and cuts for the cockroach graphs C_{4k}.

Cockroach C_{4k}	The smallest $\mathrm{cut}(S,S^c)$ (Cheeger constant)	Cheeger cut (S,S^c)
k=2	$1/4$	$S=(5,6,7,8)$
k=3	$1/5$	$S=(1,2,3,7,8,9)$, $S=(7,8,9)$, $S=(1,2,3,4,7,8,9,10)$, $S=(1,2,3)$
k=4	$1/8$	$S=(1,2,3,4,5,9,10,11,12,13)$
k=5	$1/11$	$S=(1,2,3,4,5,6,11,12,13,14,15,16)$
k=6	$1/14$	$S=(1,2,3,4,5,6,7,13,14,15,16,17,18,19)$
k=7	$1/16$	$S=(9,10,11,12,13,14,23,24,25,26,27,28)$
k=8	$1/18$	$S=(10,11,12,13,14,15,16,26,27,28,29,30,31,32)$
k=9	$1/20$	$S=(11,12,13,14,15,16,17,18,29,30,31,32,33,34,35,36)$, $S=(12,13,14,15,16,17,18,30,31,32,33,34,35,36)$
k=10	$1/23$	$S=(13,14,15,16,17,18,19,20,33,34,35,36,37,38,39,40)$

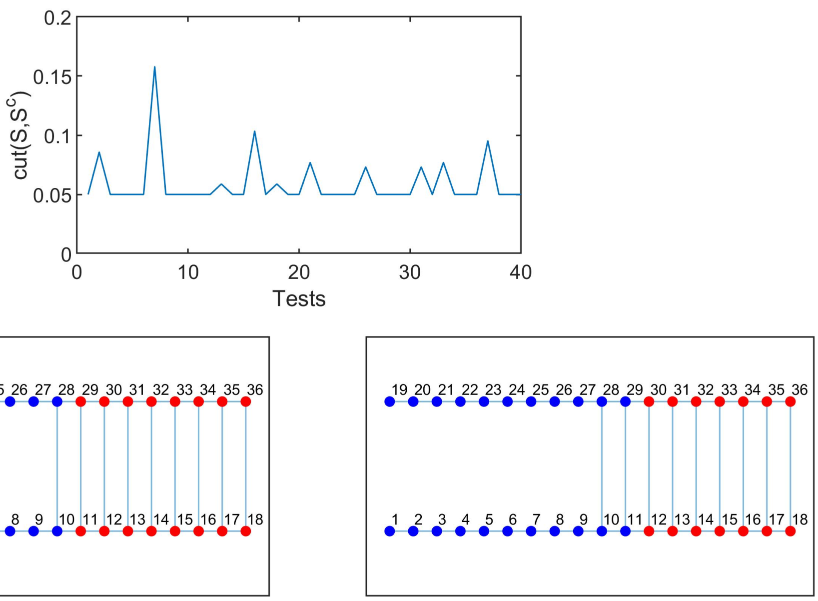

Figure 3: The plot of the cut values $\mathrm{cut}(S,S^c)$ for 40 tests with different initial value for x^0 and the obtained different Cheeger cuts for the Cockroach graph C_{36}. In this experiment, we set the parameters $r_1=r_2=100$.

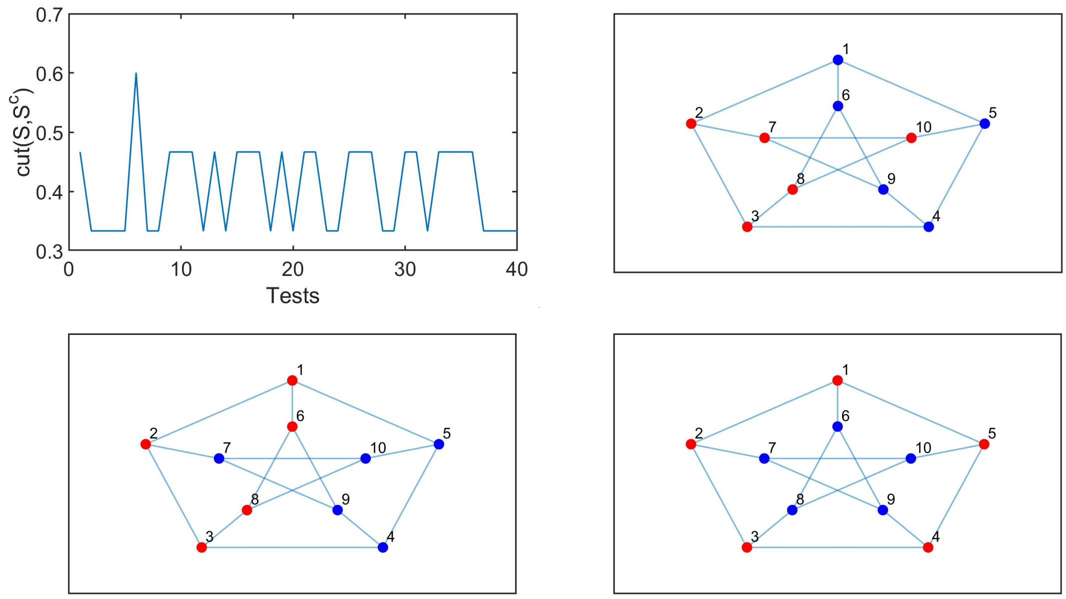

Figure 4: The plot of the cut values $\mathrm{cut}(S,S^c)$ for 40 tests with different initial value for x^0 and the obtained different Cheeger cuts for the Petersen graph. In this experiment, we set the parameters $r_1 = r_2 = 100$.

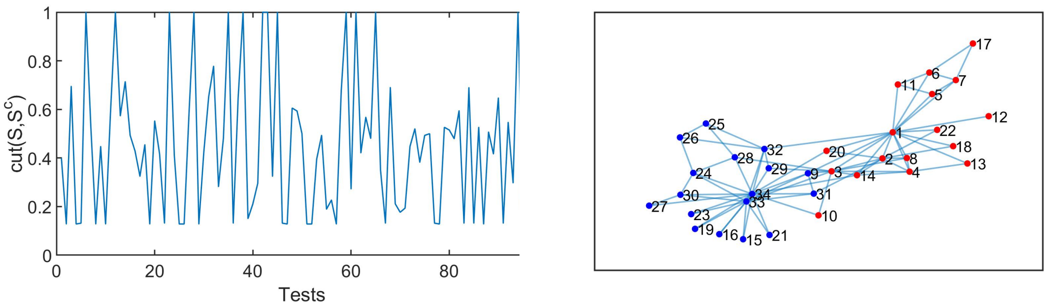

Figure 5: The plot of the cut values $\mathrm{cut}(S,S^c)$ for 40 tests with different initial value for x^0 and the obtained Cheeger cut. In this experiment, we set the parameters $r_1 = 40, r_2 = 400$.

cut

$$S = \{1,2,3,4,5,6,7,8,10,11,12,13,14,17,18,20,22\},$$

and the cut value $\mathrm{cut}(S,S^c) = 0.1282$.

As a comparison, on `https://en.wikipedia.org/wiki/Zachary%27s_karate_club`, the split result of Zachary's club is listed as follows:

$$T = \{1,2,3,4,5,6,7,8,9,11,12,13,14,17,18,20,22\},$$

and the cut value $\mathrm{cut}(T,T^c) = 0.1467$, which is larger than our obtained cut value. In fact, a direct calculation gives

$$\begin{aligned} |\partial S| &= 10, \quad \mathrm{Vol}(S) = 78, \quad \mathrm{Vol}(S^c) = 78, \\ |\partial T| &= 11, \quad \mathrm{Vol}(T) = 81, \quad \mathrm{Vol}(T^c) = 75. \end{aligned}$$

This shows that our obtained cut gives a smaller cut value than the one provided on the wiki.

The above experimental results demonstrate that the proposed algorithm is able to find Cheeger cuts of those typical unweighted and undirected graphs and real graphs. In our future work, we plan to apply the proposed algorithm for dealing with practical problems, especially in studying advanced engineering alloys, such as steels, high-entropy alloys, and nickel-superalloys [2, 9].

To see the effectiveness and efficiency of the proposed algorithm, we conduct a number of tests for those Cockroach graphs $C_{4k}, k = 2, \ldots, 10$. For each graph, we choose 50 different randomized initial guesses of x^0, run the proposed algorithm 200 iterations for each guess, and then calculate the corresponding value $\mathrm{cut}(S, S^c)$ based on the obtained variable z. In Table 2, for each graph, we list the number of those tests that successfully obtain the Cheeger cut and the averaged time spent on each test. From this table, one can see that the proposed algorithm could find Cheeger cuts with many different initial guesses of x^0, which demonstrates the effectiveness of the algorithm.

Moreover, Table 2 shows that the computational cost is very low for each of the listed graphs. This is because the proposed algorithm consists of solving three sub-problems. Specifically, for the sub-problem of the variable x, one can solve (3.10) efficiently using PCG, especially for the coefficient matrix $r_1 B^T B + r_2 I_n$ with B being sparse; while for the sub-problems of the variables y and z, the cost is just $\mathcal{O}(n)$, with n being the number of vertices. These facts explain why the proposed algorithm is efficient. In Table 2, the CPU time was recorded when the code was running under Matlab R2021b on a desktop with Intel(R) Core(TM) i5-10505 CPU @ 3.20GHz.

Table 2: The number of tests that successfully obtained Cheeger cuts among 50 tests with randomized initial guesses of x^0 and the averaged time for each test for the cockroach graphs C_{4k}.

Cockroach C_{4k}	Number of tests for obtaining Cheeger cut	Time spent for each test (seconds)
k=2	29	3.35e-02
k=3	47	3.61e-02
k=4	34	3.80e-02
k=5	38	3.94e-02
k=6	39	3.50e-02
k=7	43	4.22e-02
k=8	44	4.19e-02
k=9	42	4.07e-02
k=10	41	4.25e-02

6. Conclusion

In the paper, we propose a novel ALM/ADMM based algorithm for finding Cheeger cuts for any given unweighted and undirected graph $G = (V, E)$ by solving the corresponding eigenvalue problem of the 1-Laplacian $\Delta_1(G)$. As the related optimization problem involves a non-differentiable function over a non-convex set that consists of more than $3^{[(n+1)/2]} - 1$ simplex cells of different dimensions, there is no guarantee that the proposed algorithm must converge to the optimal solution that leads to the Cheeger cut. However, we show that the sequence generated by the proposed algorithm is bounded and thus has a convergent subsequence. To help obtain the optimal solution, we propose using different initial guesses to find different local minimizers. Among these local minimizers, we choose the one with the smallest cut value to form the goal cut. Numerical experiments demonstrate that the proposed algorithm is capable of finding the Cheeger cuts for typical graphs like the Cockroach graphs, the Petersen graph. Moreover, it is also applicable for real graphs like the Zachary karate Club graph.

Acknowledgments

The author would like to thank the anonymous referee for the valuable comments and suggestions, which have helped very much to improve the presentation of this paper.

References

[1] E. Bae, X. C. Tai, and W. Zhu, *Augmented Lagrangian method for an Euler's elastica based segmentation model that promotes convex contours*, Inverse Probl. Imag. 11 (2017), 1–23.

[2] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. Mailoa, M. Kornbluth, N. Molinari, T. Smidt, and B. Kozinsky, *E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials*, Nat. Commun. 1 (2022), p. 2453.

[3] X. Bresson, T. Laurent, D. Uminsky, and J. H. von Brecht, *Multiclass total variation clustering*, Adv. Neural Inf. Process. Syst. 26 (2013), 1421–1429.

[4] X. Bresson, X. C. Tai, T. Chan, and A. Szlam, *Multi-class transductive learning based on l^1 relaxations of Cheeger cut and Mumford-Shah-Potts model*, J. Math. Imaging Vis. 49 (2014), 191–201.

[5] T. Buhler and M. Hein, *Spectral clustering based on the graph p-Laplacian*, in: International Conference on Machine Learning, (2009), 81–88.

[6] K. C. Chang, *Spectrum of the 1-Laplacian and Cheeger's constant on graphs*, J. Graph Theory (2016), 167–207.

[7] K. C. Chang, S. H. Shao, and D. Zhang, *The 1-Laplacian Cheeger cut: Theory and algorithms*, J. Comput. Math. 33 (2015), 443–467.

[8] J. Cheeger, *A lower bound for the smallest eigenvalue of the Laplacian*, in: Proceedings of the Princeton Conference in Honor of Professor S. Bochner, (1969), 195–199.

[9] C. CHEN AND S. ONG, *A universal graph deep learning interatomic potential for the periodic table*, Nat. Comput. Sci. 11 (2022), 718–728.
[10] F. R. CHUNG, *Spectral Graph Theory*, AMS, 1997.
[11] R. GLOWINSKI AND P. L. TALLEC, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, SIAM, 1989.
[12] T. GOLDSTEIN AND S. OSHER, *The split Bregman method for l1-regularized problems*, SIAM J. Imaging Sci. 2 (2009), 323–343.
[13] L. HAGEN AND A. KAHNG, *Fast spectral methods for ratio cut partitioning and clustering*, in: Proceedings of IEEE International Conference on Compter-Aided Design, (1991), 10–13.
[14] B. KAWOHL AND V. FRIDMAN, *Isoperimetric estimates for the first eigenvalue of the p-Laplace operator and the Cheeger constant*, Comment. Math. Univ. Carol. 44 (2003), 659–667.
[15] B. KAWOHL AND F. SCHURICHT, *Dirichlet problems for the 1-Laplace operator, including the eigenvalue problem*, Commun. Contemp. Math. 9 (2007), 515–543.
[16] J. KLEINBERG AND E. TARDOS, *Algorithm Design*, Pearson, 2006.
[17] R. T. ROCKAFELLAR, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. Oper. Res. 1 (1976), 97–116.
[18] S. SHAO AND C. YANG, *A simple inverse power method for balanced graph cut*, arXiv:2405.18705, 2024.
[19] J. SHI AND J. MALIK, *Normalized cuts and image segmentation*, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000), 888–905.
[20] A. SZLAM AND X. BRESSON, *Total variation and Cheeger cuts*, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, 10 (2010), 1039–1046.
[21] X. C. TAI, J. HAHN, AND G. J. CHUNG, *A fast algorithm for Euler's elastica model using augmented Lagrangian method*, SIAM J. Imaging Sci. 4 (2011), 313–344.
[22] U. VON LUXBURG, *A tutorial on spectral clustering*, Stat. Comput. 17 (2007), 395–416.
[23] M. WITMAN, S. LING, P. BOYD, S. BARTHEL, M. HARANCZYK, B. SLATER, AND B. SMIT, *Cutting materials in half: A graph theory approach for generating crystal surfaces and its prediction of 2d zeolites*, ACS Cent. Sci. 4 (2018), 235–245.
[24] C. WU AND X.-C. TAI, *Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, Vectorial TV, and high order models*, SIAM J. Imaging Sci. 3 (2010), 300–339.
[25] W. W. ZACHARY, *An information flow model for conflict and fission in small groups*, J. Anthropol. Res. 33 (1977), 452–473.
[26] W. ZHU, *A numerical study of a mean curvature denoising model using a novel augmented Lagrangian method*, Inverse Probl. Imaging 11 (2017), 975–996.
[27] W. ZHU, *A first-order image denoising model for staircase reduction*, Adv. Comput. Math. 45 (2019), 3217–3239.
[28] W. ZHU, *Image denoising using lp-norm of mean curvature of image surface*, J. Sci. Comput. 83 (2020), p. 32.
[29] W. ZHU, X.-C. TAI, AND T. CHAN, *Augmented Lagrangian method for a mean curvature based image denoising model*, Inverse Probl. Imaging 7 (2013), 1409–1432.
[30] W. ZHU, X.-C. TAI, AND T. CHAN, *Image segmentation using Euler's elastica as the regularization*, J. Sci. Comput. 57 (2013), 414–438.
[31] W. ZHU, X.-C. TAI, AND T. CHAN, *A fast algorithm for a mean curvature based image denoising model using augmented Lagrangian method*, in: Efficient Algorithms for Global Optimization Methods in Computer Vision. Lecture Notes in Computer Science, Springer, 8293 (2014), 104–118.