University of Nottingham

Applying partial differential equations on networks to approximate the Max-Cut and Max-K-Cut problems.

Blaine Keetch

A thesis submitted to the University of Nottingham for the degree of

Doctor of Philosophy

August 2020

Abstract

The Max-Cut and Max-K-Cut problems are well known combinatorial optimization problems. In this thesis we produce fast approximation methods for these problems by applying methods from partial differential equations on networks.

Given a graph G, a cut is a partition of the vertex set of G into two disjoint subsets, and a K-cut is a partition of the set into K disjoint subsets. For an unweighted graph, the size of the cut or K-cut is the number of edges which are connected by nodes belonging to distinct subsets in the partition.

We introduce the signless Ginzburg–Landau and multiclass signless Ginzburg–Landau functionals, proving that these functionals Γ-converge to a Max-Cut objective functional and a Max-K-Cut objective functional respectively. We approximately minimize these functionals using graph based signless Merriman–Bence–Osher schemes, which use a signless Laplacian.

We show experimentally that on some classes of graphs, the resulting algorithms produce more accurate maximum cut and maximum K-cut approximations than the current state-of-the-art approximation algorithms.

In this thesis, we also prove that the graph diffusion operators of graphs G_1 and G_2 are isomorphic, if and only if graphs G_1 and G_2 are isomorphic. An isomorphism of graphs G_1 and G_2 is a bijection from the vertex set of G_1 to the vertex set of G_2, such that the bijection from the vertex sets of G_1 and G_2 preserves the edges of G_1 and G_2.

Acknowledgements

I would like to thank my primary supervisor Yves van Gennip and my secondary supervisor Matthias Kurzke for their constant support mathematically and otherwise.

I would also like to thank David Hodge, Kewei Zhang, Daniel Nicks, Braxton Osting, Jeremy Budd, Zach O'Boyd, and Andrea Bertozzi for their useful mathematical discussions.

The Nottingham support staff over my time as a PhD have been invaluable, and I thank them for their support with the non mathematical parts of the PhD and for their friendliness and warmth.

I would like to thank Andrea Bertozzi and the team behind the NOMADs grant for the opportunity to spend two months at UCLA near the end of my PhD. On a personal level and a professional level the two months spent there were superb for me.

Finally, special thanks goes to my fellow PhD students; without your support this experience would not have been as great as it has been.

CONTENTS

1 INTRODUCTION 1

1.1 Overview . . . 1
1.2 Graph Laplacians . . . 1
1.3 Max-Cut . . . 2
1.4 Max-K-Cut . . . 8
1.5 Structure of the thesis . . . 13

2 LITERATURE REVIEW 15

2.1 NP problems . . . 15
2.2 Max-Cut . . . 17
2.3 Max-K-Cut . . . 20
2.4 Spectral graph theory . . . 22
2.5 Diffuse interface models . . . 26

3 PRELIMINARIES 29

3.1 Graph based operators . . . 29
3.2 The Max-Cut and Max-K-Cut problems . . . 36
3.3 Γ-convergence . . . 41
3.4 Graph definitions . . . 43
3.5 Spectral properties of Δ_r^+ . . . 44

4 MAX-CUT 49

4.1 Γ-convergence of f_ε^+ . . . 49
4.2 The signless MBO algorithm . . . 52
4.3 Γ-convergence of the Lyapunov functional for signless MBO evolutions . . . 60

5 MAX-K-CUT 68

5.1 Γ-convergence of $F^+_{\varepsilon,K}$ 68
5.2 Multiclass algorithms 72

6 GRAPH ISOMORPHISM 80
6.1 Notation 80
6.2 Diffusion operator isomorphism 81
6.3 Additional results 84

7 COMPUTATIONAL RESULTS - MAX-CUT 86
7.1 Max-Cut results 86
7.2 Max-Cut - Parameter tests 104
7.3 Minimizing f^+_ε by convex-concave splitting 112
7.4 Reduction of τ 117

8 COMPUTATIONAL RESULTS - MAX-K-CUT 134
8.1 Method 134
8.2 Max-K-Cut results 137
8.3 Max-K-Cut - Parameter tests 162
8.4 Max-Cut results by minimizing $F^+_{\varepsilon,2}$ 166

9 CONCLUSIONS AND FURTHER WORK 170
9.1 Conclusions 170
9.2 Future work 172

A APPENDIX A - MIN-K-CUT 174

1

INTRODUCTION

1.1 OVERVIEW

This thesis introduces algorithms which approximately solve the Max-Cut and Max-K-Cut problems; these algorithms are inspired by the growing work of applying partial differential equations on networks in order to tackle graph based problems. We propose the algorithms (MBO+) for the Max-Cut problem, and the algorithms (KCut) and (KMBO+) for the Max-K-Cut problem. The Max-Cut and Max-K-Cut problems are NP-Hard problems [35].

In Chapter 1.2 we provide a brief overview about graph Laplacians and their use in clustering problems on graphs. In Chapter 1.3 and Chapter 1.4, we introduce the Max-Cut and Max-K-Cut problems respectively, providing introductions to the algorithms proposed to find approximate solutions to these problems. In Chapter 1.5 we explain the structure of the remaining parts of this thesis.

1.2 GRAPH LAPLACIANS

In this thesis we will consider non-empty, finite, simple[1], undirected graphs $G = (V, E, \omega)$ without isolated nodes, with vertex set (or node set) V, edge

1 By 'simple' we mean 'without self-loops and without multiple edges between the same pair of vertices'. Note that removing self-loops from a graph does not change its maximum cut nor its maximum K cut.

set $E \subset V^2$ and non-negative edge weights ω. We denote the set of all such graphs by $\mathcal{G}$. Since G is non-empty and finite, V has finite cardinality, which we denote by $n := |V| \in \mathbb{N}^2$. We assume a node labelling such that $V = \{1, \ldots, n\}$. When $i, j \in V$ are nodes, the undirected edge between i and j, if present, is denoted by (i, j). The edge weight corresponding to this edge is $\omega_{ij} > 0$. Since G is undirected, for all $(i, j) \in E$, $\omega_{ij} = \omega_{ji}$. Within this framework we can also consider unweighted graphs, which correspond to the cases in which, for all $(i, j) \in E$, $\omega_{ij} = 1$. Because $G \in \mathcal{G}$ is simple, for all $i \in V$, $(i, i) \notin E$. The degree of a node i is $d_i := \sum_{j \in V} \omega_{ij}$. Because $G \in \mathcal{G}$ does not contain isolated nodes, we have for all $i \in V, d_i > 0$.

For a graph $G \in \mathcal{G}$, the (weighted)-adjacency matrix A is defined by

$$A_{ij} := \begin{cases} w_{ij}, \text{ if } (i, j) \in E, \\ 0, \text{ if } (i, j) \notin E, \end{cases}$$

and the degree matrix D is defined by

$$D_{ij} := \begin{cases} d_i, \text{ if } i = j, \\ 0, \text{ if } i \neq j. \end{cases}$$

Given the degree and adjacency matrix of a graph G, its unnormalised graph Laplacian L is $L := D - A$, its random walk graph Laplacian is $L_{rw} := I_n - D^{-1}A$ and its symmetric graph Laplacian is $L_s = I_n - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$, where I_n is the $n \times n$ identity matrix [19]. In Chapter 2.4 and Chapter 3.1.1, we discuss graph Laplacians in more detail.

1.3 MAX-CUT

1.3.1 *Maximum cut*

Given an undirected (edge-)weighted graph $G = (V, E, \omega)$, a cut $V_{-1}|V_1$ is a partition of the node set V into two disjoint subsets V_{-1} and V_1. The size of a

2 For definiteness we use the convention $0 \notin \mathbb{N}$.

cut $C = V_{-1}|V_1$, denoted by $s(C)$, is the sum of all the weights corresponding to edges that have one end vertex in V_{-1} and one in V_1. The maximum cut (Max-Cut) problem is the problem of finding a cut C^* such that for all cuts C, $s(C) \leq s(C^*)$. We call such a C^* a maximum cut and say $\mathrm{mc}(G) := s(C^*)$ is the maximum cut value of the graph G. The Max-Cut problem for an unweighted graph is a special case of the Max-Cut problem on a weighted graph which we obtain by assuming all edge weights are 1. Finding an unweighted graph's Max-Cut is equivalent to finding a bipartite subgraph with the largest number of edges possible. In fact, for an unweighted bipartite graph $\mathrm{mc}(G) = |E|$.

The Max-Cut problem is an NP-hard problem [35]; assuming P $\neq$ NP no solution can be acquired in polynomial time. There are a variety of polynomial time approximation algorithms for this problem [36, 83]. Some Max-Cut approximation algorithms have a proven lower bound on their accuracy, which asserts the existence of a $\beta \in [0,1]$ such that, for all output cuts C obtained by the algorithm, $s(C) \geq \beta \mathrm{mc}(G)$. We call such a β a performance guarantee. For algorithms that incorporate stochastic steps, such a lower bound typically takes the form $E[s(C)] \geq \beta \mathrm{mc}(G)$ instead, where $E[s(C)]$ denotes the expected value of the size of the output cut.

In recent years a new type of approach to approximating such graph problems has gained traction. Models from the world of partial differential equations and variational methods that exhibit behaviour of the kind that could be helpful in solving the graph problem are transcribed from their usual continuum formulation to a graph based model. The resulting discrete model can then be solved using techniques from numerical analysis and scientific computing. Examples of problems that have successfully been tackled in this manner include data classification [9], image segmentation [16], and community detection [47]. In this thesis we use a variation on the graph Ginzburg–Landau functional, which was introduced in [9], to construct an algorithm which approximately solves the Max-Cut problem on simple undirected weighted graphs.

We compare our method with the Goemans–Williamson (GW) algorithm [36], which is the current state-of-the-art method for approximately solving the Max-Cut problem. In [36] the authors solve a relaxed Max-Cut objective function and intersect the solution with a random hyperplane through the origin of a n-dimensional sphere. It is proven that if $\mathrm{gw}(C)$ is the size of the cut produced by the Goemans–Williamson algorithm, then its expected value $E[\mathrm{gw}(C)]$ satisfies the inequality $E[\mathrm{gw}(C)] \geq \beta \mathrm{mc}(G)$ where $\beta = 0.878$ (rounded down). We explain the Goemans–Williamson algorithm in more detail in Chapter 2.2.

Finding $\mathrm{mc}(G)$ is equivalent to finding the ground state of the Ising Hamiltonian in Ising spin models [44], and 0/1 linear programming problems can be restated as Max-Cut problems [59].

1.3.2 *Signless Ginzburg–Landau functional*

Spectral graph theory [19] explores the relationships between the spectra of graph operators, such as graph Laplacians (see Chapter 3.1.1), and properties of graphs. For example, the multiplicity of the zero eigenvalue of the (unnormalised, random walk, or symmetrically normalised) graph Laplacian is equal to the number of connected components of the graph. Such properties lie at the basis of the successful usage of the graph Laplacian in graph clustering, such as in spectral clustering [88] and in clustering and classification methods that use the graph Ginzburg–Landau functional [9]

$$f_\varepsilon(u) := \frac{1}{2} \sum_{i,j \in V} \omega_{ij} (u_i - u_j)^2 + \frac{1}{\varepsilon} \sum_{i \in V} W(u_i). \tag{1}$$

Here $u : V \to \mathbb{R}$ is a real-valued function defined on the node set V, with value u_i on node i, ω_{ij} is a positive weight associated with the edge between nodes i and j (and $\omega_{ij} = 0$ if such an edge is absent), and $W(x) := (x^2 - 1)^2$ is a double-well potential with minima at $x = \pm 1$. In Chapter 3.1 we will introduce our setting and notation more precisely.

In this thesis we introduce the *signless Ginzburg–Landau functional*, which is based on a variation of f_ε. The signless Ginzburg–Landau functional is:

$$f_\varepsilon^+(u) := \frac{1}{2}\sum_{i,j\in V}\omega_{ij}(u_i+u_j)^2 + \frac{1}{\varepsilon}\sum_{i\in V}W(u_i). \tag{2}$$

This nomenclature is suggested by the fact that the signless graph Laplacians are related to f_ε^+ in a similar way as the graph Laplacians are related to f_ε, as we will see in Chapter 3.1. Signless graph Laplacians have been studied because of the connections between their spectra and bipartite subgraphs [25]. In [45, 85] the authors derive a graph difference operator and a graph divergence operator to form a graph Laplacian operator. In this chapter we mimic this framework by deriving a signless difference operator and a signless divergence operator to form a signless Laplacian operator. Whereas the graph Laplacian operator is a discretization of the continuum Laplacian operator, the continuum analogue of the signless Laplacian is the subject of current and future research. Preliminary calculations for specific graph structures suggest that the continuum signless Laplacian is an integral operator [52].

The functional f_ε is useful in clustering and classification problems, since minimizers of f_ε (in the presence of some constraint or additional term, to prevent trivial minimizers) will be approximately binary (with values close to ± 1), because of the double-well potential term, and will have similar values on nodes that are connected by highly weighted edges, because of the first term in f_ε. This intuition can be formalised using the language of Γ-convergence [23]. In analogy with the continuum case in [69, 70], it was proven in [85] that if $\varepsilon \downarrow 0$, then f_ε Γ-converges to

$$f_0(u) := \begin{cases} 2\mathrm{TV}(u), & \text{if } u \text{ only takes the values } \pm 1, \\ \infty, & \text{otherwise,} \end{cases}$$

where $\mathrm{TV}(u) := \frac{1}{2}\sum_{i,j\in V}\omega_{ij}|u_i - u_j|$ is the graph total variation[3]. Together with an equi-coercivity property, which we will return to in more detail in

3 The multiplicative factor 2 in f_0 above differs from that in [85] because in this thesis we choose different locations for the wells of W.

Chapter 4.1, this Γ-convergence result guarantees that up to a subsequence, minimizers of f_ε converge to minimizers of f_0 as $\varepsilon \downarrow 0$. If u only takes the values ± 1, we note that $\mathrm{TV}(u) = 2s(C)$, where $C = V_{-1}|V_1$ is the cut given by $V_{\pm 1} := \{i \in V : u_i = \pm 1\}$. Hence minimizers u^ε of f_ε are expected to approximately solve the *minimal* cut problem, if we let $V_\pm \approx \{i \in V : u_i^\varepsilon \approx \pm 1\}$.

In Chapter 4.1 we prove that f_ε^+ Γ-converges to a limit functional whose minimizers solve the Max-Cut problem. Hence, we expect minimizers u^ε of f_ε^+ to approximately solve the Max-Cut problem, if we consider the cut $C = V_{-1}|V_1$, with $V_\pm = \{i \in V : u_i^\varepsilon \approx \pm 1\}$.

1.3.3 *Graph MBO scheme*

There are various ways in which the minimization of f_ε^+ can be attempted. One such way, is to use a gradient flow method. In the case of f_ε the gradient flow is given by an Allen–Cahn type equation on graphs [9, 86],

$$\frac{du_i}{dt} = -(\Delta u)_i - \frac{1}{\varepsilon} d_i^{-r} W'(u_i), \tag{3}$$

where Δu is a graph Laplacian of u [45], d_i the degree of node i, and $r \in [0, 1]$ a parameter (see Chapter 3.1 for further details). This can be solved using a combination of convex splitting and spectral truncation. In the case of f_ε^+ such an approach would lead to a similar equation and scheme, with the main difference being the use of a signless graph Laplacian instead of a graph Laplacian.

In this thesis, however, we have opted for an alternative approach, which is also inspired by similar approaches which have been developed for the f_ε case. The continuum Merriman–Bence–Osher (MBO) scheme [66, 67] involves iteratively solving the diffusion equation over a small time step τ and thresholding the solution to an indicator function. For a short diffusion time τ this scheme approximates motion by mean curvature [8]. This scheme has been adapted to a graph setting [64, 86]. Heuristically it is expected that the outcome of the graph MBO scheme closely approximates minimizers of f_ε, as

the diffusion step involves solving $\frac{du_i}{dt} = -(\Delta u)_i$ and the thresholding step has a similar effect as the nonlinearity $-\frac{1}{\varepsilon}W'(u_i)$ in (3). In [15], the authors prove connections between numerically solving the Allen–Cahn equation and the MBO scheme.

In order to approximately minimize f_ε^+, and consequently approximately solve the Max-Cut problem, we use an MBO type scheme in which we replace the graph Laplacian in the diffusion step by a signless graph Laplacian. We use two methods to compute this step: (1) a spectral method, adapted from the one in [9], which allows us to use a small subset of the eigenfunctions, which correspond to the smallest eigenvalues of the graph Laplacian, and (2) an Euler method. Moreover, we show (Chapter 4.3) that this variant of the graph MBO scheme has a Lyapunov functional associated with it which Γ-converges to a Max-Cut objective functional (i.e. a functional whose minimizers solve the Max-Cut problem) as the signless 'diffusion' time $\tau \downarrow 0$.

The usefulness of (normalised) signless graph Laplacians when attempting to find maximum cuts can be intuitively understood from the fact that their spectra are in a sense (which is made precise in Proposition 3.5.1) the reverse of the spectra of the corresponding (normalised) graph Laplacians. Hence, where a standard graph Laplacian driven diffusion leads to clustering patterns according to the eigenfunctions corresponding to its smallest eigenvalues, 'diffusion' driven by a signless graph Laplacian leads to patterns resembling the eigenfunctions corresponding to the smallest eigenvalues of that signless graph Laplacian and thus the largest eigenvalues of the corresponding standard graph Laplacian.

1.4 MAX-K-CUT

1.4.1 *Max K-cuts*

Given an undirected (edge-)weighted graph $G = (V, E, \omega)$, for $K \in \mathbb{N}$, a K-cut denoted $X = V_1|\ldots|V_K$ is a partition of the node set V into K disjoint

non-empty subsets $V_1, \ldots, V_K$. The size of the cut X, denoted by $S_K(X)$ is the sum of all edge weights connecting nodes from distinct node subsets. The maximum K-cut problem (Max K-Cut) is the problem of finding a K-cut X^* such that for all K-cuts X, $S_K(X^*) \geq S_K(X)$. We refer to this maximum by $m_K(G) := S_K(X^*)$.

The Max K-cut problem is NP-hard [35]; assuming P $\neq$ NP no solution can be acquired in polynomial time. Some Max K-Cut approximation algorithms have a proven lower bound on their accuracy, which asserts the existence of a $\beta_K \in [0,1]$ such that, for all output K-cuts X obtained by the algorithm, $S_K(X) \geq \beta_K m_K(G)$. We call such a β_K a performance guarantee. For algorithms that incorporate stochastic steps, such a lower bound typically takes the form $E[S_K(X)] \geq \beta_K m_K(G)$ instead, where $E[S_K(X)]$ denotes the expected value of the size of the output cut. A partition into K disjoint non-empty subsets chosen uniformly at random will achieve a cut X' with expected value $E[S_K(X')] \geq (1 - \frac{1}{K}) m_K(G)$ [32]. The authors in [32] produce an algorithm with performance guarantees of $\beta_2 = 0.878$, $\beta_3 = 0.832$, $\beta_4 = 0.850$, and $\beta_5 = 0.874$ (all rounded down). However, for larger K the algorithm does not improve significantly on the performance guarantee obtained by choosing partitions uniformly at random. The performance guarantee $\beta_2 = 0.878$ is obtained by using the analysis from [36].

The problem of determining whether a given graph is K-partite for a given K is also of interest; this is one of the 21 original NP-complete problems stated in [53]. A graph is K-partite if there exists a K-cut X' such that $S_K(X') = \frac{1}{2}\sum_{i,j \in V} \omega_{ij}$. K-partite graphs are applicable in scheduling tasks [62] and code compiler optimization [18].

1.4.2 *Multiclass Signless Ginzburg–Landau functional*

Where $K \in \mathbb{N}$, K-clustering is an important problem in machine learning. The aim of K-clustering is to partition a data-set into K meaningful clusters, where data points are clustered with other similar data points. One

well-known method to tackle the K-clustering problem is spectral clustering [26, 88]. Given a data-set, the author constructs a similarity matrix W on the data, and then constructs a graph Laplacian $L = D - W$, where D is the degree matrix of the graph. The author proposes an algorithm which involves applying the K-means algorithm on the first K-eigenvectors of L. The success of this algorithm has inspired the usage of the graph Laplacian and its spectrum in tackling K-clustering problems.

In [33, 65] the authors propose the multiclass Ginzburg–Landau functional,

$$E(U) := \frac{\varepsilon}{2}\langle U, L_s U\rangle + W_{\varepsilon,K}(U) + \sum_{i\in V}\frac{\mu_i}{2}\|U_i - \bar{U}_i\|^2. \tag{4}$$

Here $U \in \mathbb{R}^{n\times K}$, with row $U_i = (U_{i1}, \dots, U_{iK})$ on node i, L_s is the symmetric graph Laplacian (see Chapter 3.1.1), $\langle U, L_s U\rangle = \operatorname{trace}(U^T L_s U)$, $\bar{U} \in \mathbb{R}^{n\times K}$, with row $\bar{U}_i = (\bar{U}_{i1}, \dots, \bar{U}_{iK})$ on node i, $\varepsilon > 0$, and $\mu_i \geq 0$. Here, $W_{\varepsilon,K}(U)$ is a K-well potential, where

$$W_{\varepsilon,K}(U) := \frac{1}{2\varepsilon}\sum_{i\in V}\prod_{k=1}^{K}\frac{1}{4}\|U_i - E_k\|_{L_1}^2 \tag{5}$$

and, for $j \in \{1, \dots, K\}$,

$$(E_k)_j := \begin{cases} 1, & \text{if } j = k, \\ 0, & \text{if } j \neq k. \end{cases}$$

The minima E_k of (5) correspond to the corners of a K-simplex Σ_K, where

$$\Sigma_K := \{(\alpha_1, \dots, \alpha_K) \in [0,1]^K : \sum_{i\in\{1,\dots,K\}}\alpha_i = 1\}. \tag{6}$$

In (4), $\frac{\varepsilon}{2}\langle U, L_s U\rangle = \frac{\varepsilon}{2}\sum_{k\in\{1,\dots,K\}}\sum_{i,j\in V}\omega_{ij}(U_{ik} - U_{jk})^2$. The minimization of this term means nodes which are connected by highly weighted edges are likely to have similar values in the minimization of (4), and that similar nodes are more likely to be placed into the same cluster. The minimization of $W_{\varepsilon,K}$ means that solutions will be matrices whose rows are close to corners of Σ_K. The minimization of these terms means that minimizing $E(U)$ is an effective method for K-clustering problems [33, 65].

In practice, the minimization of (4) is followed by a projection of the solution's rows onto Σ_K; this projection has a probabilistic interpretation. After projection, for $i \in V$, each U_i will satisfy $\sum_{k\in\{1,\dots,K\}} U_{ik} = 1$. Each U_{ik} represents the probability that node i belongs to cluster $k \in \{1,\dots,K\}$. This step is followed by, for all $i \in V$, thresholding U_i to the closest corner of (6). As seen in [22], the closest corner of (6) to U_i will be $E_{k'}$, where $k' \in \operatorname*{argmax}_{k\in\{1,\dots,K\}} U_{ik}$.

For $i \in V$, $\bar{U}_i$ in (4) represents a priori known information about node i. If it is known that node i belongs to cluster $k' \in \{1,\dots,K\}$, then $\bar{U}_i = E_{k'}$. The parameter μ_i takes the value of a positive constant if $\bar{U}_i$ is non-zero, and zero otherwise. This allows the minimization of (4) to retain the a priori known information. Furthermore, the addition of a priori known information prevents trivial minimizers of (4).

In this thesis we introduce a variant of (4) called the multiclass signless Ginzburg–Landau functional:

$$F^{+}_{\varepsilon,K}(U) := \frac{1}{2}\sum_{i,j\in V}\sum_{k\in\{1,2,\dots,K\}} \omega_{ij}(U_{ik}+U_{jk})^2 + W^{\pm}_{\varepsilon,K}(U). \tag{7}$$

The introduction of this functional is inspired by Max-Cut approximations obtained when minimizing the signless Ginzburg–Landau functional f^{+}_{ε} in (2) in Chapter 4, and the effectiveness of minimizing E in (4) for obtaining approximate solutions to K-clustering problems. Here, $W^{\pm}_{\varepsilon,K}$ is a K-well potential, where

$$W^{\pm}_{\varepsilon,K}(U) := \frac{1}{\varepsilon}\sum_{i\in V}\prod_{k\in\{1,\dots,K\}} \frac{1}{16}\|U_i - E'_k\|_1^2 \tag{8}$$

and, for $j \in \{1,\dots,K\}$,

$$(E'_k)_j := \begin{cases} 1, & \text{if } j = k, \\ -1 & \text{if } j \neq k. \end{cases}$$

The minima E'_k of (8) correspond to the corners of a translated simplex $\Sigma^{\pm}_K$, where

$$\Sigma^{\pm}_K := \{(\alpha_1,\dots,\alpha_K) \in [-1,1]^K : \sum_{i\in[K]} \alpha_i = 2-K\}. \tag{9}$$

Similar to the methods used in [65, 33], we follow the minimization of (7) by projecting the solution to (9), and thresholding to the closest corner of (9). As shown in [22], projecting U_i on $\Sigma_K^{\pm}$ and thresholding to the closest corner of $\Sigma_K^{\pm}$ is equivalent to projecting U_i on Σ_K and thresholding to the closest corner of Σ_K in (6), in that the closest corner of (9) to U_i will be $E'_{k'}$, where $k' \in \underset{k\in\{1,\dots,K\}}{\operatorname{argmax}} U_{ik}$[4].

The thresholding of U to the corners of (9) can be related to K-cuts on a graph. For a K-cut $X = V_1|\dots|V_K$, and $k \in \{1,\dots,K\}$, $V_k := \{i \in V : U_i = E'_k\}$. In Chapter 3.2.2 we explain this more concisely.

In Chapter 5.1 we prove that $F^+_{\varepsilon,K}$ Γ-converges to a Max-K-Cut objective functional. By proving a result on the equicoercivity of $F^+_{\varepsilon,K}$ we prove that as $\varepsilon \downarrow 0$, the minimizers of $F^+_{\varepsilon,K}$ converge to solutions of the Max-K-Cut problem.

In Chapter 5.2, we introduce two algorithms to minimize $F^+_{\varepsilon,K}$. The second method uses an MBO scheme as explained in Chapter 1.4.4; the first method involves convex-concave splitting, as explained below.

1.4.3 *Convex-concave splitting*

One method we consider to minimize (7) is to use a gradient flow method. The functional $F^+_{\varepsilon,K}$ has the following gradient flow equation:

$$\frac{dU_i}{dt} = -(\widetilde{\Delta}^+_r U)_i - \frac{1}{\varepsilon} d_i^{-r} {W^{\pm}_{\varepsilon,K}}'(U)_i. \tag{10}$$

Here, $\widetilde{\Delta}^+_r U$ is a signless graph Laplacian of U (see Chapter 3.1.3), d_i is the degree of node i, $r \in [0,1]$, and ${W^{\pm}_{\varepsilon,K}}'$ is the derivative of $W^{\pm}_{\varepsilon,K}$ in (8).

In this chapter we apply a convex-concave splitting scheme to solve (10); the convex-concave splitting scheme is also applied in [65] to find minimizers of $E(U)$ in (4). The convex-concave splitting method is a method of gradient descent for functionals with a bounded Hessian matrix [29, 90]. Using this

4 It is possible for U_i to have multiple closest corners of (6) and (9). In practice, if there are multiple closest corners, we choose one of those corners uniformly at random in the thresholding step.

method we express $F^+_{\varepsilon,K}$ as a sum of a convex functional $F^{+,convex}_{\varepsilon,K}$ and a concave functional $F^{+,concave}_{\varepsilon,K}$. This gradient descent method is done implicitly on the convex term and explicitly on the concave term to ensure stability. Where $F^+_{\varepsilon,K} = F^{+,convex}_{\varepsilon,K} + F^{+,concave}_{\varepsilon,K}$, the convex-concave splitting scheme of $F^+_{\varepsilon,K}$ consists in solving

$$\frac{U^{n+1} - U^n}{dt} = -\frac{\delta F^{+,convex}_{\varepsilon,K}}{\delta U^{n+1}_{ik}} + \frac{\delta F^{+,concave}_{\varepsilon,K}}{\delta U^n_{ik}}. \tag{11}$$

for a fixed time step dt. As elaborated in [90], this scheme will be guaranteed to monotonically decrease to a minimum or a saddle point of $F^+_{\varepsilon,K}$.

In practice, we apply spectral truncation in order to reduce the computational time of computing (11). Explained more concisely in Chapter 5.2.1, we use a small subset of the signless Laplacian's eigenfunctions corresponding to its smallest eigenvalues in order to reduce the computation time, without sacrificing much accuracy in the results.

1.4.4 *Multiclass MBO Scheme*

Another method we apply to find minimizers of $F^+_{\varepsilon,K}$ is to use a similar MBO scheme to the one explained in Chapter 1.3.3. The multiclass MBO scheme applied to minimize $F^+_{\varepsilon,K}$ involves iteratively solving $\frac{dU_i}{dt} = -(\widetilde{\Delta}^+_r U)_i$ for a small time step τ, and thresholding the solution to the corners of $\Sigma^{\pm}_K$. Similarly to how it is expected that the results of the MBO scheme produced in Chapter 4.2 will approximately resemble the minimizers of f^+_ε, the multiclass MBO scheme consisting of a diffusion step $\frac{dU_i}{dt} = -(\widetilde{\Delta}^+_r U)_i$ and a thresholding step should have a similar effect as solving (10), and thus will approximately resemble the minimizers of $F^+_{\varepsilon,K}$. This method is similar to the MBO scheme used in [65] to find minimizers of (4). The key differences are that our diffusion equation is defined using the signless Laplacian instead of the graph Laplacian, and that we threshold the solutions to the corners of $\Sigma^{\pm}_K$ instead of Σ_K due to $F^+_{\varepsilon,K}$ in (7) having $W^{\pm}_{\varepsilon,K}$ in (8) as the K-well potential instead of $W_{\varepsilon,K}$ in (5).

As with the MBO scheme stated in Chapter 1.3.3, we apply spectral truncation on the signless Laplacian when computing the diffusion step of the multiclass MBO scheme in order to reduce the computation time.

1.5 STRUCTURE OF THE THESIS

In Chapter 2 we discuss the relevant literature for this thesis. In Chapter 2.1 we provide a brief overview of NP and NP-hard problems. We discuss the relevant literature on the Max-Cut and Max-K-Cut problems in Chapter 2.2 and Chapter 2.3 respectively. In Chapter 2.4 we discuss the literature on spectral graph theory. Chapter 2.5 discusses the relevant literature for the graph Ginzburg–Landau functional introduced in Chapter 1.3.2, and the relevant literature for the graph MBO scheme introduced in Chapter 1.3.3.

Chapter 3 consists of an introduction to the notation used in this thesis, alongside preliminary results in Chapter 3.1, and definitions of specific graphs used in this thesis in Chapter 3.4. In Chapter 3.2 we discuss the Max-Cut and Max-K-Cut problems. In Chapter 3.3 we define Γ-convergence, and state some previously established Γ-convergence results. In Chapter 3.5 we prove results for the spectrum of the graph Laplacian and signless Laplacian operators used in this thesis.

The algorithm (MBO+) is introduced in Chapter 4 to find approximate solutions to the Max-Cut problem. In Chapter 4.1 we use Γ-convergence techniques to prove that minimizers of f_ε^+ converge to solutions of the Max-Cut problem. We describe (MBO+) in detail in Chapter 4.2. In Chapter 4.3 we derive a Lyapunov functional whose value is non-increasing along (MBO+) iterations, and show that subject to a re-scaling this functional Γ-converges to a total variation objective functional, which is shown to be related to maximum cuts.

The algorithms (KCut) and (KMBO+) are introduced in Chapter 5 to produce approximate solutions to the Max-K-Cut problem. In Chapter 5.1 we use Γ-convergence techniques to prove that minimizers of $F_{\varepsilon,K}^+$ converge to

solutions of the Max-K-Cut problem. We describe the algorithms (KCut) and (KMBO+) in detail in Chapter 5.2.

In Chapter 6, we define the notion of isomorphic operators, and show that the graph diffusion operators of two graphs are isomorphic, if and only if, the two graphs are isomorphic.

In Chapter 7 we present some results produced in practice using (MBO+). In Chapter 7.1.1 we explain the tests in more detail. In Chapters 7.1.2-7.1.6 we display our results on some sets of graphs, comparing the results of (MBO+) to the results of (GW) (see Chapter 2.2. In Chapter 7.2 we provide additional parameter analysis. In Chapter 7.3 we minimize f_ε^+ using a convex-concave splitting scheme. In Chapter 7.4 we introduce (MBO$\downarrow$), which is the algorithm 4.2 with a linearly decreasing time step along iterations. We run tests using (MBO$\downarrow$) and compare the results against tests using (MBO+).

In Chapter 8 we present some results produced in practice using (KCut) and (KMBO+). In Chapter 8.1 we explain the tests in more detail. In Chapters 8.2-8.2.4 we display the results obtained on some graphs for the 3-Cut, 4-Cut, 5-Cut problems, comparing the results of (KCut) and (KMBO+) to the results obtained by (SDPCut) (see Chapter 2.3). In Chapter 8.2.5 we display results obtained on some graphs using (KMBO+) for the K-cut problem, where $K \geq 5$. In Chapter 8.3 we provide additional parameter analysis. In Chapter 8.4 we minimize $F_{\varepsilon,2}^+$ in order to obtain approximate solutions to the Max-Cut problem.

We conclude the thesis and discuss future work in Chapter 9. Furthermore, we show some additional results in Appendix A.

In Appendix A we show that the addition of a barrier function to the multiclass Ginzburg–Landau functional $F_{\varepsilon,K}$ introduced in Chapter 3.1.3 results in a functional which Γ-converges to a Min-K-Cut objective functional.

Most of the work in Chapter 1.3, Chapter 3.1, Chapter 3.1.2, Chapter 3.2.1, Chapter 3.5, Chapter 4, and Chapter 7 has been published in [54].

2

LITERATURE REVIEW

2.1 NP PROBLEMS

The time taken to run an algorithm is calculated by the number of elementary operations the algorithm requires. Elementary operations are defined as operations which do not increase in the time taken when the input increases; examples of elementary operations are the addition and multiplication of two numbers. The time taken to run an algorithm is defined as a function $f(n)$, where n is the size of the algorithm input. The function $f(n)$ states the number of elementary operations the algorithm computes, with respect to the input size n.

We say that $f(n) = \mathcal{O}(g(n))$, if and only if for all sufficiently large values of n, the absolute value of $f(n)$ is at most a positive constant multiple of $g(n)$. We refer to the time taken to run an algorithm as its time complexity, which is expressed by $\mathcal{O}(g(n))$, where g is a function and n an input for the problem. The time complexity indicates the proportionality in the maximal time taken to run an algorithm against the size of the input. If $g(n)$ can be expressed as a polynomial, the time complexity of the algorithm is said to be polynomial time. If $g(n)$ can be expressed as 2^{n^k} for some positive constant k, the time complexity of the algorithm is said to be exponential time.

The complexity class of a problem is based on the time complexity of its most efficient algorithm which guarantees to solve the problem for all possible inputs. This has lead to the creation of the complexity classes P and NP.

P is the class of all problems which are solvable in polynomial time. NP is the class of problems whose solutions are verifiable as correct or incorrect, in polynomial time. The question "is P = NP is true or false?" is one of the six remaining unsolved Millennium Prize Problems [50].

If L and M are problems, we say that L is reducible to M, denoted $L \leq_P M$, if there exists an polynomial time algorithm P such that for all inputs x of L, $P(x)$ is an input of M.

The complexity classes NP-hard and NP-complete are also of interest. A problem M is called NP-hard, and belongs to the NP-hard complexity class, if for all NP problems L, $L \leq_P M$. NP-complete is the class of all problems which are both NP and NP-hard. If there exists a solver for an NP-complete problem which runs in polynomial time, all problems in NP are solvable in polynomial time, thus proving that P = NP.

The theory of NP-completeness was established by Karp in 1972 [53]. Karp proves that 21 specific problems are NP-complete; this is achieved by proving that these problems can be polynomial time transformed from the k-SAT problem[1]. The k-SAT problem was proved to be NP-complete by Cook in 1971 [20], however the nomenclature NP-complete was not established by then. From 1972, more problems have since been proven to belong to the NP-complete class, such as the travelling salesman problem [74]. The key reference work for NP-complete problems is the book by Garey [35] which outlines all the problems which had been categorised as NP-complete up to the year 1979. Some additional problems have since been introduced and classified as NP-complete; the modularity problem [73] is one such example [13].

In recent years some problems which were not in the complexity class P have been reclassified to belong in P. The decision problem "Given a number x, is x prime?" [1] is one such example, having been believed to be in NP in 1975. There is an interest in problems which belong to NP, but do not

1 The k-SAT problem is determining whether a Boolean formula in conjunctive-normal-form with k variables per clause is satisfiable.

belong to the classes P or NP-complete. This has inspired the introduction of the NP-intermediate (NPI) complexity class [58]. In this paper the author constructs an artificial NP problem which cannot be polynomial time solvable and cannot be NP-complete. The author further conjectures that, assuming P $\neq$ NP, the NPI class is non-empty. Two such examples that are conjectured to be NPI are the discrete logarithm problem [63] and the integer factorization problem [72].

The difficulty in finding accurate performance guarantees for approximations to NP problems can be highlighted by the unique games conjecture [55], alongside specific results for the Max-Cut [82] and Max-K-Cut problems [51]. In [56] the author proves that if the unique games conjecture is true, for all $\varepsilon > 0$ it is NP-hard to achieve a performance guarantee higher than $\beta_{GW} + \varepsilon$ for the Max-Cut problem, where β_{GW} is the performance guarantee of the Goemans–Williamson algorithm, which was briefly discussed in Chapter 1.3.1 and is explained in more detail in Chapter 2.2. In [82] the authors show that without the unique games conjecture, for all $\varepsilon > 0$, obtaining a performance guarantee of $\frac{16}{17} + \varepsilon$ for Max-Cut is itself an NP-hard problem. Furthermore, for any K the authors in [51] prove that, unless P = NP, no polynomial time solver for the Max K-Cut has a performance guarantee greater than $(1 - \frac{1}{34K})$.

The exponential time hypothesis was proposed in [48], which states that for $k \geq 3$, k-SAT cannot be solved in subexponential time. In [48], the authors further show that if the exponential time hypothesis is true, for $k \geq 3$, k-SAT increases in computational difficulty as k increases. Therefore, the exponential time hypothesis being true implies that no NP-hard problem can be solved in sub-exponential time.

2.2 MAX-CUT

The Max-Cut problem is NP-hard, which can be proved by a chain of reductions originating from the 3-SAT problem [35].

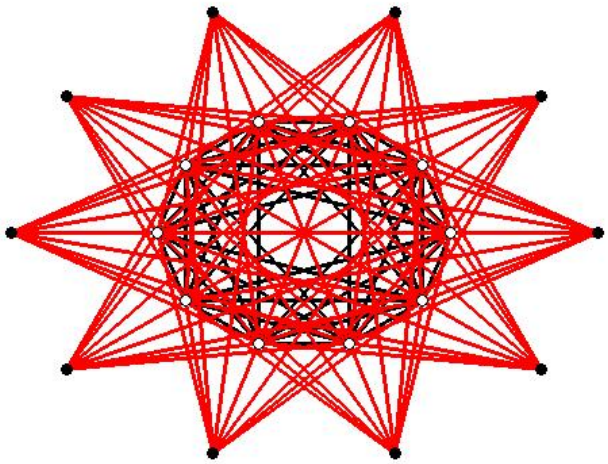


Figure 1.: A maximum cut on a core-periphery graph.

The Max-Cut problem is solvable in polynomial time if restrictions are placed on the graph; finding the maximum cut of a planar graph is solvable in polynomial time [43]. The authors show that the Max-Cut problem on a planar graph is the same as the polynomial time solvable maximum weighted bipartite matching problem on the geometric dual of the graph. Furthermore, if the graph is a line graph [42], or weakly bipartite [40], solutions can also be found in polynomial time. In Figure 1 we show an example of a Max-Cut on a core-periphery graph [77], where the node subsets V_1 and V_{-1} correspond to the black nodes and the white nodes respectively. The cut edges are highlighted in red, and the edges not included in the cut are black.

As explained in Chapter 1.3.1, the current state of the art Max-Cut algorithm with respect to obtaining a polynomial time achievable performance guarantee was introduced by Goemans and Williamson in [36], which has a performance guarantee of 0.878. We present this algorithm in detail below in Algorithm (GW). A problem equivalent to the Max-Cut problem is to find a maximizer which achieves

$$\max_u \frac{1}{2}\sum_{i,j}\omega_{ij}(1-u_iu_j) \quad \text{subject to } \forall i \in V, u_i \in \{-1,1\}.$$

The (GW) algorithm solves a relaxed version of this integer quadratic program, by allowing u to be an n-dimensional vector with unit Euclidean norm. The n-dimensional vector relaxation is an upper bound on the original integer quadratic program. This relaxed problem is equivalent to finding a maximizer which achieves

$$Z_P^* := \max_Y \frac{1}{2} \sum_{i,j \in V, i<j} \omega_{ij}(1 - y_{ij}), \tag{12}$$

where the maximization is over all n by n real positive-semidefinite matrices $Y = (y_{ij})$ with ones on the diagonal. This semidefinite program has an associated dual problem of finding a minimizer which achieves

$$Z_D^* := \frac{1}{2} \sum_{i,j \in V} \omega_{ij} + \frac{1}{4} \min_{\gamma \in \mathbb{R}^n} \sum_{i \in V} \gamma_i, \tag{13}$$

subject to $W + \mathrm{diag}(\gamma)$ being positive-semidefinite, where W is the adjacency matrix of G and $\mathrm{diag}(\gamma)$ is the diagonal matrix with diagonal entries γ_i.

In Algorithm (GW) below, we use the unit sphere $S_n := \{x \in \mathbb{R}^n : \|x\| = 1\}$, where $\|\cdot\|$ denotes the Euclidean norm on $\mathbb{R}^n$. For vectors $w, \tilde{w} \in \mathbb{R}^n$, $w \cdot \tilde{w}$ denotes the Euclidean inner product.

Previously, the best performance guarantee for the Max-Cut problem was $\frac{1}{2}$ [79]. Prior to 2012, this was the highest performance guarantee for a Max-Cut polynomial time algorithm which did not use semi-definite programming. In [83], a non-semi-definite programming Max-Cut algorithm using the spectra of graph operators to partition the graph's nodes achieved a performance guarantee of $\alpha_{TR} = 0.531$ [83]; after further analysis of the algorithm, its guarantee was improved to $\alpha_{TR} = 0.614$ in [81]. The problem of determining if a cut of a given size exists on a graph is NP-complete [53].

Given an unweighted graph, its Max-Cut can be bounded by the number of nodes and edges the graph contains. For the number of nodes n and the number of edges $|E|$ the inequality $\frac{|E|}{2} + \frac{\sqrt{8|E|+1}-1}{8} \leq \mathrm{mc}(G)$ holds for all unweighted graphs; if the graph is connected then $\frac{|E|}{2} + \frac{1}{4}(n-1) \leq \mathrm{mc}(G)$ holds [27].

Algorithm (GW): The Goemans–Williamson algorithm

Data: The weighted adjacency matrix W of a graph $G \in \mathcal{G}$, and a tolerance ν.

Relaxation step: Use semidefinite programming to find approximate solutions $\tilde{Z}_P^*$ and $\tilde{Z}_D^*$ to (12) and (13), respectively, which satisfy $|\tilde{Z}_P^* - \tilde{Z}_D^*| < \nu$. Use an incomplete Cholesky decomposition of the matrix Y that achieves $\tilde{Z}_P^*$ in (12) to find an approximate solution to

$$w^* \in \underset{w \in (S_n)^n}{\operatorname{argmax}} \frac{1}{2} \sum_{\substack{1 \le i,j, \le n \\ i<j}} \omega_{ij}(1 - w_i \cdot w_j).$$

Hyperplane step: Let $r \in S_n$ be a random vector drawn from the uniform distribution on S_n. Define the cut $C := V_1 | V_1^c$, where

$$V_1 := \{i \in V | w_i \cdot r \geq 0\}.$$

2.3 MAX-K-CUT

For a given $K \in \mathbb{N}$, if $\hat{X}$ is a K-cut where each node is assigned to a node subset uniformly at random, and X^* is a maximum K-cut, then $\hat{X}$ will satisfy the inequality

$$\mathbb{E}(S_K(\hat{X})) \geq (1 - \frac{1}{K}) S_K(X^*),$$

where $\mathbb{E}(S_K(\hat{X}))$ denotes the expectation value of $S_K(\hat{X})$ [32]. This can be seen by each node having a $(1 - \frac{1}{K})$ probability of connecting to another node from a different partition.

As explained in Chapter 1.4.1, the algorithm in [32] has a performance guarantee for any K. For $K = 3$ approximation algorithms created by [37] and [24] achieve a performance guarantee of $\beta_3 \geq 0.836$, improving on the 0.832 guarantee obtained in [32]. However, their framework cannot extend to solving the Max K-cut problem for all possible K.

We present the algorithm proposed in [32] in (SDPCut). As justified in [32], the Max-K-Cut problem can be formulated as follows:

Let $b_1, \ldots, b_K$ be the vertices of Σ_K as in (6), let $c_K := \frac{(b_1+\cdots+b_K)}{K}$ be the centroid of Σ_K, and let $a_i := b_i - c_K$ for $1 \leq i \leq K$. Since Σ_K is scaled, $|a_i| = 1$ for $1 \leq i \leq K$. The Max-K-Cut problem can be formulated by:

$$\text{IP}_K : \text{maximize } \frac{K-1}{2K} \sum_{i,j \in V} \omega_{ij}(1 - y_i \cdot y_j)$$

$$\text{subject to: } y_j \in \{a_1, \ldots, a_K\}.$$

In [32] the authors relax the problem IP_K by replacing y_i with v_i, where $v_i \in S_n$. As justified in [32], this requires the additional constraint $v_i \cdot v_j \geq \frac{-1}{(K-1)}$. The semi-definite programming relaxation of IP_K is therefore defined as follows:

$$\text{SDP}_K : \text{maximize } \frac{K-1}{2K} \sum_{i,j \in V} \omega_{ij}(1 - v_i \cdot v_j)$$

$$\text{subject to: } \forall j \in V, v_j \in S_{n-1}; \forall i \neq j, v_i \cdot v_j \geq \frac{-1}{(K-1)}.$$

The algorithm (SDPCut) is expressed as follows:

Algorithm (SDPCut): The K-cut semi-definite programming algorithm

---

Data: Weighted adjacency matrix A of a graph $G \in \mathcal{G}$

SDP step: Solve the problem SDP_K to obtain vectors $v_1, \ldots, v_n \in S_n$.

Random vector step: Choose K vectors $z_1, \ldots, z_K \in S^n$ uniformly at random.

Partitioning step: Partition V according to which of $z_1, \ldots, z_K$ is closest to each v_j, let $X = V_1 | \ldots | V_K$ be defined by

$$V_i = \{j : v_j \cdot z_i \geq v_j \cdot z_{i'} \text{ for } i \neq i'\}, \text{for } 1 \leq i \leq K.$$

As proved in [32], (SDPCut) has a performance guarantee satisfying the following. If $X' \in \mathcal{X}^K$ such that $S_K(X') = m_K(G)$, then the cut X produced by (SDPCut) satisfies $\mathbb{E}(S_K(X)) \geq \alpha_K S_K(X')$, where $\alpha_2 \geq 0.879, \alpha_3 \geq 0.832, \alpha_4 \geq 0.850, \alpha_5 \geq 0.874, \alpha_{10} \geq 0.967,$ and $\alpha_{100} \geq 0.991$.

The graph K-partite problem (or K-colouring problem) is determining whether there exists a partition of the graph's nodes into K disjoint subsets such that no edges connect two nodes from the same subset. To formalise:

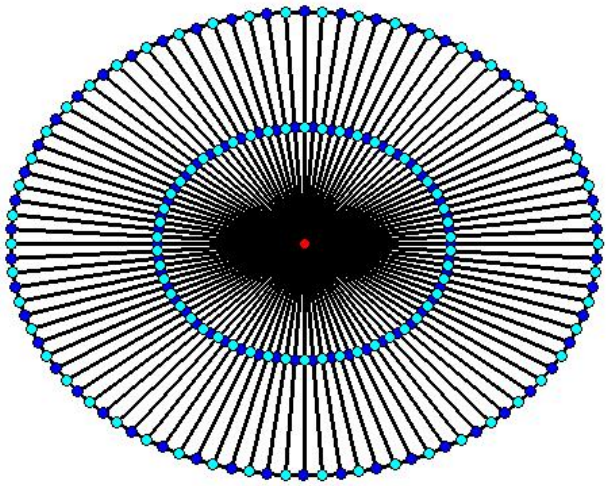


Figure 2.: An example of a graph which is 3-partite.

Definition 2.3.1. (The K-partite problem)

For a given graph G, and an integer $K \geq 2$, is G K-partite?

For $K = 2$, the K-partite problem is equivalent to deducing whether G is bi-partite, which is solvable in linear time [60]. For $K \geq 3$, the K-partite problem is NP-complete [53].

In 1852, the four colour conjecture was proposed. It conjectured that any planar graph is 4-partite; this was finally proved 120 years later in [5]. An example which demonstrates the difficulty of the K-partite problem is in [34]; the authors prove that if a planar graph is 3-partite and has maximum node degree of 4 or greater, then finding the 3-partition is itself a NP-complete problem.

Figure 2 shows a graph which is 3-partite. The partitions are the set of light blue nodes, the set of dark blue nodes, and the set of red nodes. As seen in Figure 2, each edge connects nodes of distinct colours.

2.4 SPECTRAL GRAPH THEORY

The best known graph operator is the graph Laplacian. In [45] the authors construct a node function Hilbert space $\mathcal{V}$, and an edge function Hilbert space

$\mathcal{E}$. From this a graph difference operator $\nabla : \mathcal{V} \to \mathcal{E}$, a graph divergence operator $\text{div} : \mathcal{E} \to \mathcal{V}$, and the graph Laplacian operator $\Delta = \text{div}(\nabla) : \mathcal{V} \to \mathcal{V}$ are derived. We adapt this framework to derive the signless Laplacian operator in Chapter 3.1.

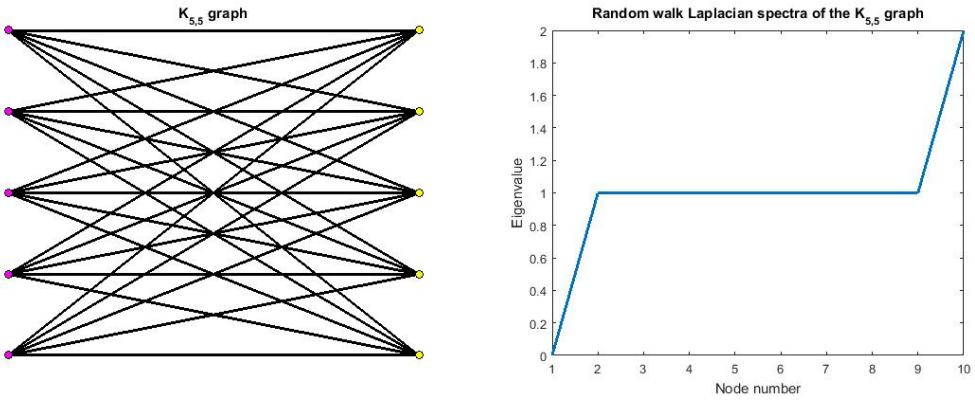
Spectral graph theory [19] focuses on the connections between graphs and the spectra of graph operators. An example of this is Kirchhoff's theorem [57]; this theorem states that for a connected graph G, the number of spanning trees can be calculated as $t(G) = \frac{1}{n}\lambda_1\lambda_2 \cdots \lambda_{n-1}$, where λ_k, $k \in \{1, \ldots, n-1\}$, are the non-zero, non-negative eigenvalues of the graph Laplacian. Another important result in spectral graph theory is that the graph Laplacian has a zero eigenvalue of multiplicity equal to the number of connected components of a graph [19].

The algebraic connectivity of a graph was first introduced by Fiedler in 1973 [30]. Where $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ are the eigenvalues of the graph Laplacian, listing multiple times for multiplicity, the algebraic connectivity is λ_2. This nomenclature is chosen due to the fact that $\lambda_2 = 0$ if and only if the graph is not connected. In [30], the author proves some results for λ_2, relating λ_2 to edge and node connectivity in a graph.

In [19] the author proves some results for the spectra of the graph Laplacian. For example, for a connected graph, the random walk or symmetric normalised Laplacian has its largest eigenvalue equal to 2, if and only if the graph is bipartite.

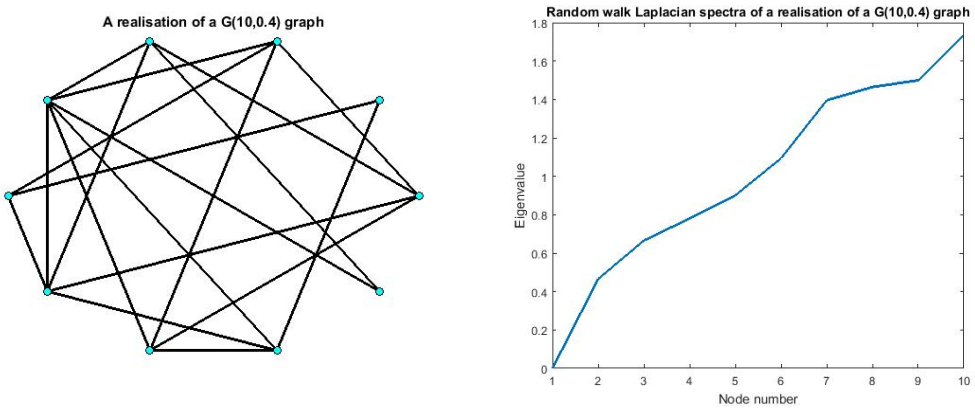
In Figure 3 we show a complete bipartite graph and the spectrum of its random walk graph Laplacian. In Figure 4 we show a realisation of an Erdös-Rényi graph (see Chapter 3.4.3) with $n = 10$ and $p = 0.4$ and the spectrum of its random walk graph Laplacian. As seen in Figure 3b and Figure 4b, both Laplacians have eigenvalue 0, and in Figure 3b the Laplacian has eigenvalue 2, due to the graph being bipartite.

Unlike the graph Laplacian, the signless Laplacian is not derived from a continuum operator; instead, it is derived from the graph Laplacian. The spectra of the random walk and signless random walk Laplacians are con-



(a) $K_{5,5}$ graph (b) Plot of $K_{5,5}$ spectra

Figure 3.: The $K_{5,5}$ graph [11] and its random walk Laplacian spectra.



(a) A realisation of a $G(10.0.4)$ graph (b) Plot of the realisation $G(10, 0.4)$ spectra

Figure 4.: A realisation of a $G(10.0.4)$ graph (see Definition 3.4.3) and its random walk Laplacian spectra.

tained in $[0,2]$. For any graph, the random walk graph Laplacian (L_1) and the random walk signless Laplacian (L_1^+) are related by the fact that $L_1^+ = 2I_n - L_1$. From this, the spectral properties of the random walk signless Laplacian are "reversed" in comparison to the random walk graph Laplacian. Specifically, the signless random walk Laplacian always has a largest eigenvalue 2, of multiplicity equal to the number of connected components of the graph, and the signless random walk Laplacian's smallest eigenvalue is equal to zero, if and only if the graph is bipartite.

The spectral theory of graph operators can be practically applied to some specific problems on graphs; one such problem is the problem of K-clustering. For $K \in \mathbb{N}$, the purpose of K-clustering is to partition n data points into K disjoint subsets, such that data points are clustered with other similar data points. Spectral clustering [26, 88] is a well known K-clustering algorithm. A dataset is represented as a similarity graph with weighted edges. Each data point is represented as a node on the similarity graph, and the weight between two nodes of the graph indicates the similarity of the two data points. One method utilised in [88] to generate a similarity graph is to use a Gaussian similarity measure. Where x_i and x_j are data points in the dataset, the adjacency matrix of the similarity graph S is defined by $S(x_i, x_j) := exp(\frac{-\|x_i - x_j\|^2}{2\sigma^2})$, where $\|\cdot\|$ denotes the Euclidean norm, and $\sigma > 0$. The authors construct a graph Laplacian (unnormalized, random walk, or symmetric) using the adjacency matrix and the degree matrix of a similarity graph. The K-means algorithm is then applied to the K eigenvectors of the graph Laplacian, corresponding to the K smallest eigenvalues; in practice this usually achieves an accurate approximation of the maximal K-clustering. Figure 5 displays applying spectral clustering on a two moon dataset [92].

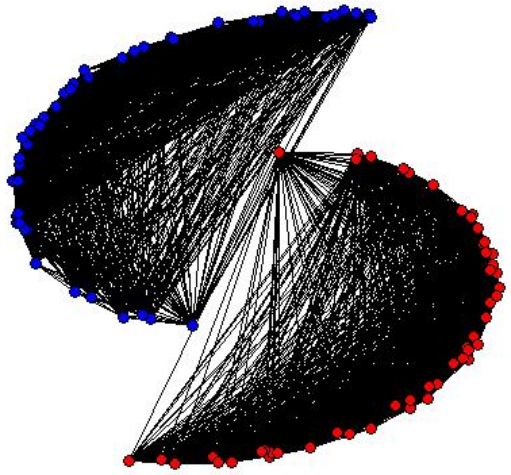


Figure 5.: Example of spectral clustering on a two moons dataset.

2.5 DIFFUSE INTERFACE MODELS

Diffuse interface models are widely used to model phase separation with a free boundary between its phases, such as in fluid mechanics [39], and superconductivity [89]. Diffuse interface models are often built around the Ginzburg–Landau functional:

$$GL(u) := \frac{\varepsilon}{2}\int_\Omega |\nabla u|^2 dx + \frac{1}{\varepsilon}\int_\Omega W(u)dx \tag{14}$$

Here, u is a scalar field representing the state of the phases in the system, ∇ represents the spatial gradient operator, and $\varepsilon > 0$. This functional is of interest in modelling phase separation due to the property of its minimizers. In (14), $\int_\Omega |\nabla u|^2 dx$ is a smoothing term, and W is a quartic double well potential, with minima traditionally chosen to be either $\{-1, 1\}$ or $\{0, 1\}$. The minimization of $\int_\Omega |\nabla u|^2 dx$ penalises variations between adjacent phases, whilst the minimization of the double well potential pushes phases into the minima of the double well potential.

As shown in [69, 70], as $\varepsilon \to 0$, the Ginzburg–Landau functional in (14) Γ-converges (see Chapter 3.3) to the total variation semi norm $\|u\|_{\mathrm{TV}}$, where

$$\|u\|_{\mathrm{TV}} := \int_\Omega |\nabla u|. \tag{15}$$

The space Ω represents all functions with bounded variation, taking on values of the wells of $W(u)$.

The properties of (14) mean that the evolution of its gradient flow can effectively model the dynamics of two phases in material science. One method to model the gradient flow is by the Allen–Cahn equation

$$\frac{du}{dt} = -\nabla GL(u) = \varepsilon \Delta u - \frac{1}{\varepsilon} W'(u), \tag{16}$$

which is the L^2 gradient flow of (14). Here, Δ is the Laplacian operator. This is evolved to a steady state in order to obtain a local minimizer of (14). We note that (14) is not convex and therefore contains multiple energy minima.

The authors in [9] produce a generalization of the Ginzburg–Landau functional (as in (1)) for functions which are defined on networks. For $u : V \to \mathbb{R}$, the gradient term used in the continuum equation is replaced with the discrete gradient term $\|\nabla u\|_{\mathcal{E}}^2$ introduced in [45] (see Chapter 3.1 for more details). As elaborated in Chapter 1.3.2, the graph Ginzburg–Landau functional Γ-converges to total variation defined on graphs [85]. The definition of total variation of graphs is equivalent to the definition of a graph cut.

The original MBO scheme (or threshold dynamics scheme) in the continuum was introduced to approximate motion by mean curvature flow [66, 67]. It consists of iteratively applying (N times) two steps: diffusing a binary initial condition for a time τ and then thresholding the result back to a binary function. In the (suitably scaled) limit $\tau \downarrow 0$, $N \to \infty$, solutions of this process converge to solutions of motion by mean curvature [8]. It is also known that solutions the continuum Allen–Cahn equation (in the limit $\varepsilon \downarrow 0$) converge to solutions of motion by mean curvature [14]. Therefore, this implies that for small values of ε, the MBO scheme is effective at numerically solving the Allen–Cahn equation.

The MBO scheme has been adapted to a graph based setting, where the diffusion step is replaced with solving the graph diffusion equation $\frac{du_i}{dt} = -(\Delta u)_i$ as elaborated in Chapter 1.3.3. Experimentally, the graph MBO scheme has proven to be an effective method of minimizing the graph Ginzburg–

Landau functional in [9]. In [15], the authors prove connections between numerically solving the Allen–Cahn equation and the MBO scheme.

The graph MBO scheme has also been shown to be effective for some additional problems. In [47] the authors show that the MBO scheme can effectively optimize modularity. In [64], the authors use the MBO scheme for image segmentation, image inpainting, and data classification tasks. The MBO scheme has also be adapted to simulate auction dynamics [49], where volume constraints are placed on the resulting clusters generated by the scheme.

3

PRELIMINARIES

3.1 GRAPH BASED OPERATORS

3.1.1 *Graph and Hilbert space set up*

We define $\mathcal{V}$ to be the set consisting of all node functions $u : V \to \mathbb{R}$ and $\mathcal{E}$ to be the set of edge functions $\varphi : E \to \mathbb{R}$ [45]. We will use the notation $u_i := u(i)$ and $\varphi_{ij} := \varphi(i,j)$ for functions $u \in \mathcal{V}$ and $\varphi \in \mathcal{E}$, respectively. For notational convenience, we will typically associate $\varphi \in \mathcal{E}$ with its extension to V^2 obtained by setting $\varphi_{ij} = 0$ if $(i,j) \notin E$. We also extend ω to V^2 in this way: if $(i,j) \notin E$, then $\omega_{ij} = 0$. We recall the definition of node degree d_i from Chapter 1.2.

As shown in [45], it is possible for $\mathcal{V}$ and $\mathcal{E}$ to be defined for directed graphs, but we will not pursue these ideas in this thesis.

To introduce the graph Laplacians and signless graph Laplacians we use and extend the structure that was used in [45, 85, 86]. We define the inner products on $\mathcal{V}$ and $\mathcal{E}$ as

$$\langle u, v\rangle_{\mathcal{V}} := \sum_{i\in V} u_i v_i d_i^r, \qquad \langle \varphi, \phi\rangle_{\mathcal{E}} := \frac{1}{2}\sum_{i,j\in V} \varphi_{ij}\phi_{ij}\omega_{ij}^{2q-1},$$

where $r \in [0,1]$ and $q \in [\frac{1}{2}, 1]$. If $r = 0$ and $d_i = 0$, we interpret d_i^r as 0. Similarly for ω_{ij}^{2q-1} and other such expressions below.

For $K \in \mathbb{N}$ we define $\mathcal{V}^K$ to be the set of all node functions $U : V \to \mathbb{R}^K$ and $\mathcal{E}^K$ to be the set of edge functions $\Phi : E \to \mathbb{R}^K$. We can represent a function

$U \in \mathcal{V}^K$ by a matrix in $\mathbb{R}^{n\times K}$, where we will use the notations $U_i := U(i)$ and $U_{ik} := U(i)_k$ for $U \in \mathcal{V}^K$. We use the notation $(\Psi_{ij})_k := (\Psi(i,j))_k$ for $\Psi \in \mathcal{E}^K$. We define $[K] := \{1, \dots, K\}$ and we define the inner products on $\mathcal{V}^K$ and $\mathcal{E}^K$ by

$$\langle U, V\rangle_{\mathcal{V}^K} := \sum_{\substack{i\in V,\\ k\in[K]}} U_{ik}V_{ik}d_i^r, \quad \langle \Phi, \Psi\rangle_{\mathcal{E}^K} := \sum_{\substack{i,j\in V\\ k\in[K]}} (\Phi_{ij})_k(\Psi_{ij})_k\omega_{ij}^{2q-1},$$

where $r \in [0,1]$ and $q \in [\frac{1}{2}, 1]$.

3.1.2 *Signless Ginzburg–Landau functional*

We define the graph gradient operator $(\nabla : \mathcal{V} \to \mathcal{E})$ by, for all $(i,j) \in E$,

$$(\nabla u)_{ij} := \omega_{ij}^{1-q}(u_j - u_i).$$

We define the graph divergence operator $(\mathrm{div} : \mathcal{E} \to \mathcal{V})$ as the adjoint of the gradient, and a graph Laplacian operator $(\Delta_r : \mathcal{V} \to \mathcal{V})$ as the graph divergence of the graph gradient: for all $i \in V$,

$$(\mathrm{div}\varphi)_i := \frac{1}{2}d_i^{-r}\sum_{j\in V}\omega_{ij}^q(\varphi_{ji} - \varphi_{ij}),\ (\Delta_r u)_i := (\mathrm{div}(\nabla u))_i = d_i^{-r}\sum_{j\in V}\omega_{ij}(u_i - u_j). \tag{17}$$

We note that the choices $r = 0$ and $r = 1$ lead to Δ_r being the unnormalised graph Laplacian and random walk graph Laplacian, respectively [71, 88]. Hence it is useful for us to explicitly incorporate r in the notation Δ_r for the graph Laplacian.

In analogy with the graph gradient, divergence, and Laplacian, we now define their 'signless' counterparts. We define the signless gradient operator $(\nabla^+ : \mathcal{V} \to \mathcal{E})$ by, for all $(i,j) \in E$,

$$(\nabla^+ u)_{ij} := \omega_{ij}^{1-q}(u_j + u_i).$$

Then, we define the signless divergence operator ($\mathrm{div}^+ : \mathcal{E} \to \mathcal{V}$) to be the adjoint of the signless gradient, and the signless Laplacian operator ($\Delta_r^+ : \mathcal{V} \to \mathcal{V}$) as the signless divergence of the signless gradient[1]: for all $i \in V$,

$$(\mathrm{div}^+ \varphi)_i := \frac{1}{2} d_i^{-r} \sum_{j \in V} \omega_{ij}^q (\varphi_{ji} + \varphi_{ij}),$$

$$(\Delta_r^+ u)_i := (\mathrm{div}^+(\nabla^+ u))_i = d_i^{-r} \sum_{j \in V} \omega_{ij} (u_i + u_j).$$

By definition we have

$$\langle \nabla u, \phi \rangle_{\mathcal{E}} = \langle u, \mathrm{div}\, \phi \rangle_{\mathcal{V}}, \qquad \langle \nabla^+ u, \phi \rangle_{\mathcal{E}} = \langle u, \mathrm{div}^+ \phi \rangle_{\mathcal{V}}.$$

Proposition 3.1.1. *The operators $\Delta_r : \mathcal{V} \to \mathcal{V}$ and $\Delta_r^+ : \mathcal{V} \to \mathcal{V}$ are self-adjoint and positive-semidefinite.*

Proof. The original proof showing that Δ_r is self-adjoint can be found in [45]. Let $u, v \in \mathcal{V}$. Since $\langle u, \Delta_r v \rangle_{\mathcal{V}} = \langle \nabla u, \nabla v \rangle_{\mathcal{E}} = \langle \Delta_r u, v \rangle_{\mathcal{V}}$ and $\langle u, \Delta_r^+ v \rangle_{\mathcal{V}} = \langle \nabla^+ u, \nabla^+ v \rangle_{\mathcal{E}} = \langle \Delta_r^+ u, v \rangle_{\mathcal{V}}$, the operators are self-adjoint. Positive-semidefiniteness follows from $\langle u, \Delta_r u \rangle_{\mathcal{V}} = \langle \nabla u, \nabla u \rangle_{\mathcal{E}} \geq 0$ and $\langle u, \Delta_r^+ u \rangle_{\mathcal{V}} = \langle \nabla^+ u, \nabla^+ u \rangle_{\mathcal{E}} \geq 0$. □

In the literature a third graph Laplacian is often used, besides the unnormalised and random walk graph Laplacians. This symmetrically normalised graph Laplacian [19] is defined by, for all $i \in V$,

$$(\Delta_s u)_i := \frac{1}{\sqrt{d_i}} \sum_{j \in V} \omega_{ij} \left(\frac{u_i}{\sqrt{d_i}} - \frac{u_j}{\sqrt{d_j}} \right).$$

This Laplacian cannot be obtained by choosing a suitable r in the framework we introduced above, but will be useful to consider in practical applications. Analogously, we define the signless symmetrically normalised graph Laplacian by, for all $i \in V$,

$$(\Delta_s^+ u)_i := \frac{1}{\sqrt{d_i}} \sum_{j \in V} \omega_{ij} \left(\frac{u_i}{\sqrt{d_i}} + \frac{u_j}{\sqrt{d_j}} \right).$$

1 In some papers the space $\mathcal{E}$ is defined as the space of all *skew-symmetric* edge functions. We do not require the skew-symmetry condition here, hence $\nabla^+ u \in \mathcal{E}$, having div^+ act on $\nabla^+ u$ is consistent with our definitions, and $\mathrm{div}^+ \varphi$ is not identically equal to 0 for all $\varphi \in \mathcal{E}$.

There is a canonical way to represent a function $u \in \mathcal{V}$ by a vector in $\mathbb{R}^n$ with components u_i. The operators Δ_r and Δ_r^+ can then be represented by the $n \times n$ matrices $L_r := D^{1-r} - D^{-r}A$ and $L_r^+ := D^{1-r} + D^{-r}A$, respectively. Here D is the degree matrix, i.e. the diagonal matrix with diagonal entries $D_{ii} := d_i$, and A is the weighted adjacency matrix with entries $A_{ij} := \omega_{ij}$.

Similarly the operators Δ_s and Δ_s^+ are then represented by $L_s := I - D^{-1/2}AD^{-1/2}$ and $L_s^+ := I + D^{-1/2}AD^{-1/2}$, respectively, where I denotes the $n \times n$ identity matrix. Any eigenvalue-eigenvector pair (λ, v) of L_r, L_r^+, L_s, L_s^+ corresponds via the canonical representation to an eigenvalue-eigenfunction pair (λ, ϕ) of Δ_r, Δ_r^+, Δ_s, Δ_s^+, respectively. We refer to the eigenvalue-eigenvector pair (λ, v) as an eigenpair.

For a vertex set $S \subset V$, we define the indicator function (or characteristic function)

$$\chi_S := \begin{cases} 1, & \text{if} \quad i \in S, \\ 0, & \text{if} \quad i \notin S. \end{cases}$$

We define the inner product norms $\|u\|_{\mathcal{V}} := \sqrt{\langle u, u\rangle_{\mathcal{V}}}$, $\|\phi\|_{\mathcal{E}} := \sqrt{\langle \phi, \phi\rangle_{\mathcal{E}}}$ which we use to define the Dirichlet energy and signless Dirichlet energy,

$$\frac{1}{2}\|\nabla u\|_{\mathcal{E}}^2 = \frac{1}{4}\sum_{i,j\in V} \omega_{ij}(u_i - u_j)^2 \quad \text{and} \quad \frac{1}{2}\|\nabla^+ u\|_{\mathcal{E}}^2 = \frac{1}{4}\sum_{i,j\in V} \omega_{ij}(u_i + u_j)^2.$$

In particular we recognise that the graph Ginzburg–Landau functional $f_\varepsilon : \mathcal{V} \to \mathbb{R}$ from (1) and the signless graph Ginzburg–Landau functional $f_\varepsilon^+ : \mathcal{V} \to \mathbb{R}$ from (2) can be written as

$$f_\varepsilon(u) = \|\nabla u\|_{\mathcal{E}}^2 + \frac{1}{\varepsilon}\sum_{i\in V} W(u_i) \quad \text{and} \quad f_\varepsilon^+(u) = \|\nabla^+ u\|_{\mathcal{E}}^2 + \frac{1}{\varepsilon}\sum_{i\in V} W(u_i).$$

It is interesting to note here an important difference between the functionals f_ε and f_ε^+. Most of the results that are derived in the literature for f_ε (such as the Γ-convergence results in [85]) do not crucially depend on the specific locations of the wells of W. For example, in f_ε the wells are often chosen to be at 0 and 1, instead of at -1 and 1. However, for f_ε^+ we have less freedom to choose the wells without drastically altering the properties of the functional. The wells have to be placed symmetrically with respect to 0, because we want

$(u_i + u_j)^2$ to be zero when u_i and u_j are located in different wells. In particular, we see that placing a well at 0 would have the undesired consequence of introducing the trivial minimizer $u = 0$. This points to a second, related, difference. Whereas minimization of f_ε in the absence of any further constraints or additional terms in the functional leads to trivial minimizers of the form $u = c\chi_V$, where $c \in \mathbb{R}$ is one of the values of the wells of W (so $c \in \{-1, 1\}$ for our choice of W), minimizers of f_ε^+ are not constant, if the graph has more than one vertex. The following lemma gives the details.

Lemma 3.1.2. *Let $G \in \mathcal{G}$ with $n \geq 2$, let $\varepsilon > 0$, and let u be a minimizer of $f_\varepsilon^+ : \mathcal{V} \to \mathbb{R}$ as in (2). Then u is not a constant function.*

Proof. Let $c \in \mathbb{R}$ and $i^* \in V$. Define the functions $u, \bar{u} \in \mathcal{V}$ by $u := c\chi_V$ and

$$\bar{u}_i := \begin{cases} c, & \text{if } i \neq i^*, \\ -c, & \text{if } i = i^*. \end{cases}$$

Since W is an even function, we have $\sum_{i \in V} W(\bar{u}_i) = \sum_{i \in V} W(u_i)$. Moreover, since for all $j \in V$, $\omega_{i^* j} = 0$ or $u_j = -u_{i^*}$, we have

$$\|\nabla^+ \bar{u}\|_{\mathcal{E}}^2 = \frac{1}{2} \sum_{\substack{i \in V \\ i \neq i^*}} \sum_{j \in V} \omega_{ij} (2c)^2 < \frac{1}{2} \sum_{i,j \in V} \omega_{ij} (2c)^2 = \|\nabla^+ u\|_{\mathcal{E}}^2.$$

The inequality is strict, because per assumption G has no isolated nodes and thus there is a $j \in V$ such that $\omega_{i^* j} > 0$. We conclude that $f_\varepsilon^+(\bar{u}) < f_\varepsilon^+(u)$, which proves that u is not a minimizer of f_ε^+. □

We define the graph total variation $\mathrm{TV} : \mathcal{V} \to \mathbb{R}$ as

$$\mathrm{TV}(u) := \max\{\langle u, \operatorname{div} \varphi \rangle_{\mathcal{V}} : \varphi \in \mathcal{E}, \forall i, j \in V \ |\varphi_{ij}| \leq 1\} = \frac{1}{2} \sum_{i,j \in V} \omega_{ij}^q |u_i - u_j|. \tag{18}$$

The second expression follows since the maximum in the definition is achieved by $\varphi = \operatorname{sgn}(\nabla u)$ [86]. We can define an analogous (signless total variation) functional $\mathrm{TV}^+ : \mathcal{V} \to \mathbb{R}$, using the signless divergence:

$$\mathrm{TV}^+(u) := \max\{\langle u, \operatorname{div}^+ \varphi \rangle_{\mathcal{V}} : \varphi \in \mathcal{E}, \forall i, j \in V \ |\varphi_{ij}| \leq 1\}.$$

Lemma 3.1.3. *Let $u \in \mathcal{V}$, then* $\mathrm{TV}^+(u) = \frac{1}{2}\sum_{i,j\in V} \omega_{ij}^q |u_i + u_j|$.

Proof. Let $\varphi \in \mathcal{E}$ such that, for all $i, j \in V$, $|\varphi_{ij}| \leq 1$. We compute

$$\begin{aligned}\langle u, \mathrm{div}^+ \varphi\rangle_{\mathcal{V}} &= \frac{1}{2}\sum_{i,j\in V} \omega_{ij}^q u_i(\varphi_{ji} + \varphi_{ij}) = \frac{1}{2}\sum_{i,j\in V} \omega_{ij}^q \varphi_{ij}(u_i + u_j) \\ &\leq \frac{1}{2}\sum_{i,j\in V} \omega_{ij}^q |\varphi_{ij}||u_i + u_j| \leq \frac{1}{2}\sum_{i,j\in V} \omega_{ij}^q |u_i + u_j|.\end{aligned}$$

Moreover, since $\varphi = \mathrm{sgn}(\nabla^+ u)$ is an admissable choice for φ and

$$\langle \mathrm{sgn}\left(\nabla^+ u\right), \mathrm{div}^+ \varphi\rangle_{\mathcal{V}} = \frac{1}{2}\sum_{i,j\in V} \omega_{ij}^q |u_i + u_j|,$$

the result follows. □

Note that the total variation functional that was mentioned in Chapter 1.3.2 corresponds to the choice $q = 1$ in (18). This is the relevant choice for this thesis and hence from now on we will assume that $q = 1$. Note that the choice of q does not have any influence on the form of the graph (signless) Laplacians.

One consequence of the choice $q = 1$ is that TV and TV^+ are now closely connected to cut sizes: If $S \subset V$ and $C = S|S^c$ is the cut induced by S, then

$$\mathrm{TV}\left(\chi_S - \chi_{S^c}\right) = 2\mathrm{TV}\left(\chi_S\right) = 2s(C) \quad \text{and} \quad \mathrm{TV}^+\left(\chi_S - \chi_{S^c}\right) = \sum_{i,j\in V} \omega_{ij} - 2s(C). \tag{19}$$

We will give a precise definition of $s(C)$ in Definition 3.2.1.

3.1.3 *Multiclass signless Ginzburg–Landau functional*

The graph gradient, graph divergence, and graph Laplacian operators in Chapter 3.1.2 can be interpreted as operators on $\mathcal{V}^K$ and $\mathcal{E}^K$. We introduce the difference operator ($\widetilde{\nabla} : \mathcal{V}^K \to \mathcal{E}^K$), divergence operator ($\widetilde{\mathrm{div}} : \mathcal{E}^K \to \mathcal{V}^K$), and the Laplacian operator ($\widetilde{\mathrm{div}}(\widetilde{\nabla} U) : \mathcal{V}^K \to \mathcal{V}^K$) as

- $((\widetilde{\nabla} U)_{ij})_k := \omega_{ij}^{1-q}(U_{ik} - U_{jk})$,
- $(\widetilde{\mathrm{div}}\Phi)_{ik} := \frac{1}{2} d_i^{-r} \sum_{j\in V} \omega_{ij}^q ((\Phi_{ji})_k - (\Phi_{ij})_k)$,

- $(\widetilde{\Delta}_r U)_{ik} := \frac{1}{d_i^r} \sum_{j \in V} \omega_{ij}(U_{ik} - U_{jk})$.

Similarly, we introduce the signless difference operator ($\widetilde{\nabla}^+ : \mathcal{V}^K \to \mathcal{E}^K$), signless divergence operator ($\widetilde{\mathrm{div}}^+ : \mathcal{E}^K \to \mathcal{V}^K$), and the signless Laplacian operator ($\widetilde{\mathrm{div}}^+(\widetilde{\nabla} U) : \mathcal{V}^K \to \mathcal{V}^K$) as

- $((\widetilde{\nabla}^+ U)_{ij})_k := \omega_{ij}^{1-q}(U_{ik} + U_{jk})$,
- $(\widetilde{\mathrm{div}}^+ \Phi)_{ik} := \frac{1}{2} d_i^{-r} \sum_{j \in V} \omega_{ij}^q ((\Phi_{ji})_k + (\Phi_{ij})_k)$,
- $(\widetilde{\Delta}_r^+ U)_{ik} := \frac{1}{d_i^r} \sum_{j \in V} \omega_{ij}(U_{ik} + U_{jk})$.

There is a canonical way to represent a function $U \in \mathcal{V}^K$ by a matrix in $\mathbb{R}^{n \times K}$ with components $U_i = (U_{i1}, \ldots, U_{iK})$. As with the operators Δ_r and Δ_r^+, $\widetilde{\Delta}_r$ and $\widetilde{\Delta}_r^+$ can be represented by the $n \times n$ matrices $L_r = D^{1-r} - D^{-r}A$ and $L_r^+ = D^{1-r} + D^{-r}A$ respectively.

Using the definition above, we see that $F_{\varepsilon,K}^+$ from (7) can be expressed by

$$F_{\varepsilon,K}^+(U) = \langle U, \widetilde{\Delta}_r^+ U \rangle_{\mathcal{V}^K} + W_{\varepsilon,K}^{\pm}(U).$$

We can also express $F_{\varepsilon,K}^+$ in (7) by

$$F_{\varepsilon,K}(U) := \langle U, \widetilde{\Delta}_r U \rangle_{\mathcal{V}^K} + W_{\varepsilon,K}^{\pm}(U). \tag{20}$$

Like in Chapter 3.1.2 where f_ε and f_ε^+ were expressed using the Dirichlet energy and signless Dirichlet energy, we can express (20) and (7) in a similar way. Where $\|U\|_{\mathcal{V}^K} := \sqrt{\langle U, U \rangle_{\mathcal{V}^K}}$ and $\|\Phi\|_{\mathcal{E}^K} := \sqrt{\langle \Phi, \Phi \rangle_{\mathcal{E}^K}}$, we define the K-Dirichlet energy and K-signless Dirichlet energy by

$$\frac{1}{2}\|\widetilde{\nabla} U\|_{\mathcal{E}^K}^2 = \frac{1}{4} \sum_{\substack{i,j \in V \\ k \in [K]}} \omega_{ij}(U_{ik} - U_{jk})^2 \text{ and } \frac{1}{2}\|\widetilde{\nabla}^+ U\|_{\mathcal{E}^K}^2 = \frac{1}{4} \sum_{\substack{i,j \in V \\ k \in [K]}} \omega_{ij}(U_{ik} + U_{jk})^2.$$

Furthermore, we can express $F_{\varepsilon,K}$ in (20) and $F_{\varepsilon,K}^+$ in (7) by

$$F_{\varepsilon,K}(U) = \|\widetilde{\nabla} U\|_{\mathcal{E}^K}^2 + W_{\varepsilon,K}^{\pm}(U) \text{ and } F_{\varepsilon,K}^+(U) = \|\widetilde{\nabla}^+ U\|_{\mathcal{E}^K}^2 + W_{\varepsilon,K}^{\pm}(U)$$

respectively. In Chapter 3.2.2, we show that for appropriately chosen $U \in \mathcal{V}^K$, both $\|\widetilde{\nabla} U\|_{\mathcal{E}^K}^2$ and $\|\widetilde{\nabla}^+ U\|_{\mathcal{E}^K}^2$ are connected to K-cut sizes.

As explained in Chapter 3.1.2, we have more freedom to choose the double well potential for f_ε in (1) than the functional f_ε^+ in (2). Similarly, we have more freedom to choose the K-well potential for $F_{\varepsilon,K}$ in (20) than we do for $F_{\varepsilon,K}^+$ in (7). For $F_{\varepsilon,K}$ in (20), the minima of the K-well potential can be chosen to be the corners of either Σ_K in (6) or $\Sigma_K^\pm$ in (9). For $F_{\varepsilon,K}^+$, we require the minima of the K-well potential to be the corners of $\Sigma_K^\pm$ in (9). As explained more concisely in Chapter 3.2.2, the corners of $\Sigma_K^\pm$ correspond to a K-cuts node subsets, in that for $i \in V$ and $U \in \mathcal{V}^K$, if $U_i = E'_k$, we say that node i belongs to the node subset V_k in the corresponding K-cut. For any $k \in [K]$, we want $(U_{ik} + U_{jk})^2$ to be zero when $U_i = E'_k$ and $U_j = E'_l$, where $k \neq l$. If the minima of the K-well potential of $F_{\varepsilon,K}^+$ in (7) corresponds to the corners of Σ_K in (6), such a property does not hold.

In Lemma 3.2.6 we justify the K-well potential choice $W_{\varepsilon,K}^\pm$ in (8) for $F_{\varepsilon,K}^+$.

3.2 THE MAX-CUT AND MAX-K-CUT PROBLEMS

3.2.1 *The Max-Cut problem*

In order to identify candidate solutions to the Max-Cut problem with node functions in $\mathcal{V}$, we define the subset of binary $\{-1, 1\}$-valued node functions,

$$\mathcal{V}^b := \{u \in \mathcal{V} : \forall\, i \in V, u_i \in \{-1, 1\}\}.$$

For a given function $u \in \mathcal{V}^b$ we define the sets $V_k := \{i \in V, u_i = k\}$ for $k \in \{-1, 1\}$. We say that the partition $C = V_{-1}|V_1$ is the cut induced by u. We define the set of all possible cuts, $\mathcal{C} := \{C : \text{there exists a } u \in \mathcal{V}^b \text{ such that } u \text{ induces the cut } C\}$.

Definition 3.2.1 ([53])**.** Let $G = (V, E, \omega) \in \mathcal{G}$ and let V_1 and V_{-1} be two disjoint subsets of V. The size of the cut $C = V_{-1}|V_1$ is

$$s(C) := \sum_{\substack{i \in V_{-1} \\ j \in V_1}} \omega_{ij}.$$

A maximum cut of G is a cut $C^* \in \mathcal{C}$ such that, for all cuts $C \in \mathcal{C}$, $s(C) \leq s(C^*)$. The size of the maximum cut is

$$\mathrm{mc}(G) := \max_{C \in \mathcal{C}} s(C).$$

Note that if the cut C in Definition 3.2.1 is induced by $u \in \mathcal{V}^b$, then

$$s(C) = \frac{1}{4}\langle u, \Delta_r u \rangle_{\mathcal{V}}. \tag{21}$$

Moreover, if $C = \emptyset | V_1$ or $C = V_{-1} | \emptyset$, then $s(C) = 0$.

Definition 3.2.2 (Max-Cut problem, [35])**.** Given $G \in \mathcal{G}$, find a maximum cut for G.

For a given $G \in \mathcal{G}$ we have $|E| < \infty$, hence a maximum cut for G exists, but note that this maximum cut need not be unique.

The cardinality of the set $\mathcal{V}^b$ is equal to the total number of ways a set of n elements can be partitioned into two disjoint subsets, i.e. $|\mathcal{V}^b| = 2^n$. This highlights the difficulty of finding $\mathrm{mc}(G)$ as n increases.

3.2.2 *The Max-K-Cut problem*

In order to identify candidate solutions to the Min-K-Cut and Max-K-Cut problems we define two subsets of $\mathcal{V}^K$:

$$\Omega_K := \{U \in \mathcal{V}^K : \forall i \in V, \exists\, k \in [K], U_i = E'_k\} \text{ and}$$

$$\Omega_K^* := \{U \in \Omega_K : \forall k \in [K], \exists\, i \in V, U_i = E'_k\}.$$

Definition 3.2.3 ([35])**.** Let $G \in \mathcal{G}$. Let $X = V_1 | \ldots | V_K$ be a node partition of V into K disjoint non-empty subsets. Then we call X a K-cut of G, and the value of the K-cut is defined by

$$S_K(X) := \sum_{\substack{k,l \in [K] \\ k<l}} \sum_{i \in V_k, j \in V_l} \omega_{ij}.$$

We denote $\mathcal{X}^K$ as the space of all possible node partitions of V into K disjoint non-empty subsets.

If $U_X \in \Omega_K$, we say that U_X induces the cut X, where $X = V_1|\dots|V_K$, and for $j \in [K]$, $V_j := \{i \in V : (U_X)_i = E'_j\}$.

The space Ω_K^* corresponds to all functions in $\mathcal{V}^K$ which can induce a K-cut. The space Ω_K corresponds to all functions in $\mathcal{V}^K$ which can induce an L-cut, where $L \leq K$. In this case, if $U_X \in \Omega_K$, and $X = V_1|\dots|V_K$, we say that U_X induces an L-cut, where L is equal to the number of non-empty subsets induced in this way.

Definition 3.2.3 allows for $K = 1$ and $K = 2$. If $K = 1$ and $X \in \mathcal{X}^1$, $S_1(X) = 0$, since all nodes are placed into one subset. If $K = 2$, the definition of a 2-cut is equivalent to the definition of a graph cut given in Chapter 3.2.1.

The main problem of interest in this chapter is the Max-K-Cut problem. A maximum K-cut of $\mathcal{G}$ is a K-cut $X^* \in \mathcal{X}^K$ such that, for all $X \in \mathcal{X}^K$, $S_K(X) \leq S_K(X^*)$. The size of the maximum K-cut is

$$m_K(G) := \max_{X \in \mathcal{X}^K} S_K(X).$$

Definition 3.2.4. (The Max-K-Cut problem, [35])

Given $G \in \mathcal{G}$ and $K \in \mathbb{N}\backslash\{1\}$, find a maximum K-cut of G.

For any $G \in \mathcal{G}$, we have that $|E| < \infty$, and therefore a maximum K-cut for G exists. Similarly to the Max-Cut problem in Chapter 3.2.1, the maximum K-cut of G need not be unique.

A similar problem of interest is the K-partite problem. For $G \in \mathcal{G}$, we say that G is K-partite if there exists an $X' \in \mathcal{X}^K$ such that $S_K(X') = \frac{1}{2}\sum_{i,j \in V} \omega_{ij}$. A K-cut X' with this property is referred to as a K-partition. In some literature a K-partite graph is often referred to as a K-colourable graph; for the remainder of this thesis, we will use the term K-partite graph instead of a K-colourable graph. Any graph which is K-partite will also be L-partite, where $L > K$.

In Figure 6 we show a graph which is 4-partite, and we show a 4-partition, where each subset of the cut is represented by a colour. As seen in Figure 6, all edges connect nodes of different subsets.

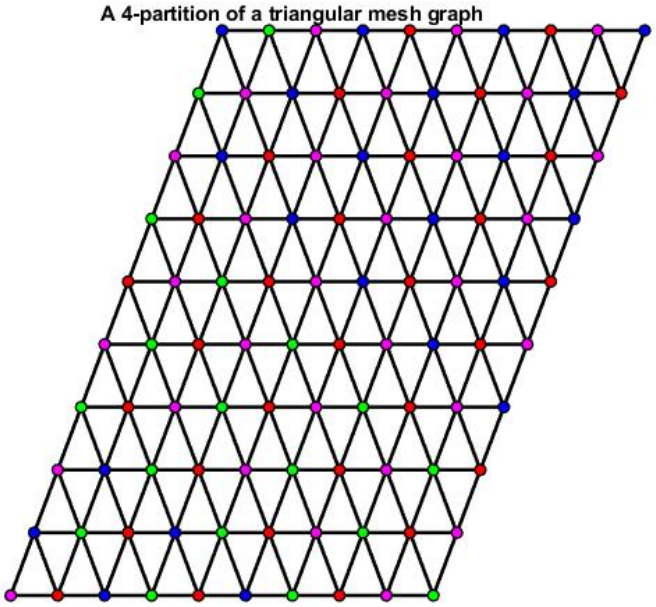


Figure 6.: A 4-partition of a triangular mesh graph [93]

Proposition 3.2.5. *Let $G \in \mathcal{G}$. For all $K \in \mathbb{N}$, there exists $X \in \mathcal{X}^{K+1}$ such that $m_K(G) < S_{K+1}(X)$, if and only if G is not K-partite.*

Proof. First assume that G is not K-partite. For any $X \in \mathcal{X}^K$ such that $S_K(X) = m_K(G)$, we have that $S_K(X) < \frac{1}{2}\sum_{i,j\in V}\omega_{ij}$. If $X = V_1|\dots|V_K$, there exists $(i,j) \in E$, and there exists $k \in [K]$ such that $i,j \in V_k$. A $(K+1)$-cut can be constructed by setting $V_{K+1} := \{i\}$, $V_k' := V_k\backslash\{i\}$, and $X' := V_1|\dots|V_{k-1}|V_k'|V_{k+1}|\dots|V_{K+1}$. Then,

$$\begin{aligned} S_{K+1}(X') &= S_K(X) - \sum_{j\in[K]\backslash\{k\}}\sum_{l\in V_j}\omega_{il} + \sum_{j\in[K]\backslash\{k\}}\sum_{l\in V_j}\omega_{il} + \sum_{l\in V_k'}\omega_{il} \\ &= S_K(X) + \sum_{l\in V_k'}\omega_{il}. \end{aligned}$$

Since $(i,j) \in E$, $i \in V_{K+1}$, $j \in V_k'$, and all edge weights of G are positive, it follows that $\sum_{l\in V_k'}\omega_{il} > 0$. Therefore, $S_K(X) < S_{K+1}(X')$.

Next assume that G is K-partite. There exists a $X \in \mathcal{X}^K$ such that $S_K(X) = m_K(G) = \frac{1}{2}\sum_{i,j\in V}\omega_{ij}$. Since G is K-partite, it will also be L-partite, for all $L > K$. Therefore, for all $L > K$, there exists an $X' \in \mathcal{X}^L$ such that $S_L(X') =$

$m_L(G) = \frac{1}{2}\sum_{i,j\in V}\omega_{ij}$. Consequently, for all $X' \in \mathcal{X}^L$, and for all $X \in \mathcal{X}^K$, $S_K(X) < S_L(X')$ is not possible.

□

Next we show the values of $\|\widetilde{\nabla}U\|^2_{\mathcal{E}^K}$ and $\|\widetilde{\nabla}^+U\|^2_{\mathcal{E}^K}$, where $U \in \Omega_K$.

Lemma 3.2.6. *Let $G \in \mathcal{G}$. If $U_X \in \Omega_K$ induces the cut $X \in \mathcal{X}^N$, where $N \in \mathbb{N}$, and $N \leq K$, then the following holds,*

- $\|\widetilde{\nabla}U_X\|^2_{\mathcal{E}^K} = 8S_N(X)$,
- $\|\widetilde{\nabla}^+U_X\|^2_{\mathcal{E}^K} = 2(K-2)\sum_{i,j\in V}\omega_{ij} + 4\sum_{k\in[N]}\sum_{i,j\in V_k}\omega_{ij}$.

Proof. Assume without loss of generality that if U_X induces an N cut for $N < K$, and $X = V_1|\dots|V_N|V_{N+1}|\dots|V_K$, it follows that $V_1 \cup \cdots \cup V_N = V$, and $V_{N+1} = V_{N+2} = \cdots = V_K = \emptyset$.

First we show that $\|\widetilde{\nabla}U_X\|^2_{\mathcal{E}^K} = 8S_N(X)$.

$$\begin{aligned}
\|\widetilde{\nabla}U_X\|^2_{\mathcal{E}^K} &= \langle U_X, \widetilde{\Delta}_r U_X\rangle_{\mathcal{V}^K} = \sum_{k\in[K]}\langle(\chi_{V_k} - \chi_{V_k^c}), \Delta_r(\chi_{V_k} - \chi_{V_k^c})\rangle_{\mathcal{V}} \\
&= \sum_{k\in[N]}\langle(\chi_{V_k} - \chi_{V_k^c}), \Delta_r(\chi_{V_k} - \chi_{V_k^c})\rangle_{\mathcal{V}} \\
&+ \sum_{k\in[K]\setminus[N]}\langle(\chi_{V_k} - \chi_{V_k^c}), \Delta_r(\chi_{V_k} - \chi_{V_k^c})\rangle_{\mathcal{V}} \\
&= \sum_{k\in[N]}\langle(\chi_{V_k} - \chi_{V_k^c}), \Delta_r(\chi_{V_k} - \chi_{V_k^c})\rangle_{\mathcal{V}} + \sum_{k\in[K]\setminus[N]}\langle(-\chi_V), \Delta_r(-\chi_V)\rangle_{\mathcal{V}}.
\end{aligned}$$

By (21), for any $k \in [K]$, if $C = V_k|V_k^c$,

$$\langle(\chi_{V_k} - \chi_{V_k^c}), \Delta_r(\chi_{V_k} - \chi_{V_k^c})\rangle_{\mathcal{V}} = 4s(C) = 4\sum_{l\in[K]\setminus\{k\}}\sum_{i\in V_k, j\in V_l}\omega_{ij}.$$

If $\bar{u} = \chi_V$, $\langle\bar{u}, \Delta_r\bar{u}\rangle_{\mathcal{V}} = 0$. It follows that

$$\begin{aligned}
\|\widetilde{\nabla}U_X\|^2_{\mathcal{E}^K} &= 4\sum_{k\in[N]}\sum_{l\in[N]\setminus\{k\}}\sum_{i\in V_k, j\in V_l}\omega_{ij} = 4\sum_{\substack{k,l\in[N]\\k<l}}\sum_{i\in V_k, j\in V_l}\omega_{ij} + 4\sum_{\substack{k,l\in[N]\\k>l}}\sum_{i\in V_k, j\in V_l}\omega_{ij} \\
&= 8\sum_{\substack{k,l\in[N]\\k<l}}\sum_{i\in V_k, j\in V_l}\omega_{ij} = 8S_N(X).
\end{aligned}$$

Next we show that $\|\widetilde{\nabla}^+ U_X\|^2_{\mathcal{E}^K} = 2(K-2)\sum_{i,j\in V}\omega_{ij} + 4\sum_{k\in[N]}\sum_{i,j\in V_k}\omega_{ij}$.

$$\begin{aligned}\|\widetilde{\nabla} U_X\|^2_{\mathcal{E}^K} &= \langle U_X, \widetilde{\Delta}_r^+ U_X\rangle_{\mathcal{V}^K} = \sum_{k\in[K]}\langle(\chi_{V_k}-\chi_{V_k^c}), \Delta_r^+(\chi_{V_k}-\chi_{V_k^c})\rangle_{\mathcal{V}} \\ &= \sum_{k\in[N]}\langle(\chi_{V_k}-\chi_{V_k^c}), \Delta_r^+(\chi_{V_k}-\chi_{V_k^c})\rangle_{\mathcal{V}} + \sum_{k\in[K]\setminus[N]}\langle(-\chi_V), \Delta_r^+(-\chi_V)\rangle_{\mathcal{V}}.\end{aligned}$$

For any $k\in[K]$, if $C = V_k|V_k^c$,

$$\begin{aligned}\langle(\chi_{V_k}-\chi_{V_k^c}), \Delta_r^+(\chi_{V_k}-\chi_{V_k^c})\rangle_{\mathcal{V}} &= 2\sum_{i,j\in V}\omega_{ij} - 4s(C) \\ &= 2\sum_{i,j\in V}\omega_{ij} - 4\sum_{l\in[K]\setminus\{k\}}\sum_{i\in V_k, j\in V_l}\omega_{ij}.\end{aligned}$$

If $\bar{u} = \chi_V$, $\langle\bar{u}, \Delta_r^+\bar{u}\rangle_{\mathcal{V}} = 2\sum_{i,j\in V}\omega_{ij}$. It follows that

$$\begin{aligned}\|\widetilde{\nabla} U_X\|^2_{\mathcal{E}^K} &= 2\sum_{k\in[N]}\sum_{i,j\in V}\omega_{ij} - 4\sum_{k\in[N]}\sum_{l\in[K]\setminus\{k\}}\sum_{i\in V_k, j\in V_l}\omega_{ij} + \sum_{k\in[K]\setminus[N]} 2\sum_{i,j\in V}\omega_{ij} \\ &= 2K\sum_{i,j\in V}\omega_{ij} - 4\sum_{\substack{k,l\in[K]\\k<l}}\sum_{i\in V_k, j\in V_l}\omega_{ij} - 4\sum_{\substack{k,l\in[K]\\k>l}}\sum_{i\in V_k, j\in V_l}\omega_{ij} \\ &= 2K\sum_{i,j\in V}\omega_{ij} - 8S_N(X) = 2K\sum_{i,j\in V}\omega_{ij} - 4\sum_{i,j\in V}\omega_{ij} - 4\sum_{k\in[K]}\sum_{i,j\in V_k}\omega_{ij} \\ &= 2(K-2)\sum_{i,j\in V} + 4\sum_{k\in[K]}\sum_{i,j\in V_k}\omega_{ij}.\end{aligned}$$

□

3.3 Γ-convergence

Γ-convergence is designed to express the convergence of minimization problems. In Chapter 3.3 we state the definition of Γ-convergence, alongside some Γ-convergence results, which we apply in our Γ-convergence proofs in Chapter 4.1, Chapter 4.3.2, and Chapter 5.1.

Definition 3.3.1 ([[12, 23]). Let X be a metric space, let $\{F_k\}_{k=1}^{\infty}$ be a sequence of functionals, where $F_k : X \to \mathbb{R}\cup\{\infty\}$, and let $F : X \to \mathbb{R}\cup\{\infty\}$. Then, we say that F_k Γ-converges to F, denoted $F_k \to F$, if for all $u \in X$, it holds that:

(LB) For every sequence $u_k \to u$, $F(u) \le \liminf_{k\to\infty} F_k(u_k)$.

(UB) There exists a sequence $u_k \to u$, such that $\limsup_{k\to\infty} F_k(u_k) \le F(u)$.

Furthermore, we say that F is the Γ-limit of F_k.

If F_k depends on a continuous parameter k, the conditions (LB) and (UB) in Definition 3.3.1 need to be proved with an arbitrary sequence $\{k_n\}_{n=1}^{\infty}$, where $k_n \to 0$ as $n \to \infty$.

Proposition 3.3.2 ([12, 23]). *Let X be a metric space, let $\{F_k\}_{k=1}^{\infty}$ be a sequence of functionals, where $F_k : X \to \mathbb{R} \cup \{\infty\}$, and let $F : X \to \mathbb{R} \cup \{\infty\}$ be the Γ-limit of F_k. Then, if $G : X \to \mathbb{R}$ is a continuous function, $F + G$ is the Γ-limit of $F_k + G$.*

Proof. See [23, Proposition 6.21] and [12, Remark 1.7].

□

Definition 3.3.3 ([12, 23]). Let X be a metric space, and let $\{F_k\}_{k=1}^{\infty}$ be a sequence of functionals, where $F_k : X \to \mathbb{R} \cup \{\infty\}$. The sequence $\{F_k\}_{k=1}^{\infty}$ is equi-coercive if, for all $t \in \mathbb{R}$ there exists a compact set K_t such that $\{F_k \leq t\} \subset K_t$.

Below is the fundamental theorem of Γ-convergence.

Theorem 3.3.4 ([12, 23]). *Let X be a metric space, let $\{F_k\}_{k=1}^{\infty}$ be a equi-coercive sequence of functions on X, and let $F : X \to \mathbb{R} \cup \{\infty\}$ be the Γ-limit of F_k, then*

$$\exists \min_X F = \lim_{k \to 0} \inf_X F_k$$

Moreover, if (u_k) is a pre-compact sequence such that $\lim_{k\to 0} F_k(u_k) = \lim_{k\to 0} \inf_X F_k$, then every limit of a subsequence of $\{F_k\}$ is a minimum point of F.

Proof. See [23, Corollary 7.20] and [12, Theorem 1.21].

□

We require a concept of convergence in $\mathcal{V}$ and $\mathcal{V}^K$. We can identify $\mathcal{V}$ with $\mathbb{R}^n$ and $\mathcal{V}^K$ with $\mathbb{R}^{n\times K}$. Since all norms on $\mathbb{R}^n$ and $\mathbb{R}^{n\times K}$ are topologically equivalent, the choice of a particular norm is not of great importance. For definiteness, however, we say that sequence $\{u_k\}_{k\in\mathbb{N}} \subset \mathcal{V}$ converges to a $u_\infty \in \mathcal{V}$ in $\mathcal{V}$ if and only if $\|\hat{u}_k - \hat{u}_\infty\|_{\mathcal{V}} \to 0$ as $k \to \infty$, where $\hat{u}_k, \hat{u}_\infty \in \mathbb{R}^n$ are the canonical vector representations of u_k, u_∞, respectively. Furthermore, we say that the sequence $\{U_p\}_{p\in\mathbb{N}} \subset \mathcal{V}^K$ converges to a $U_\infty \in \mathcal{V}^K$ in $\mathcal{V}^K$ if and only if $\|\widehat{U_p} - \widehat{U_\infty}\|_{\mathcal{V}^K} \to 0$ as $p \to \infty$, where $\widehat{U_p}$,$\widehat{U_\infty} \in \mathbb{R}^{n\times K}$ are the canonical matrix representations of U_p and U_∞ respectively.

3.4 GRAPH DEFINITIONS

In the thesis we deal with some specific types of graph. Whilst some graph definitions were provided in Chapters 1.3-1.4, for completeness we include their definitions below.

Definition 3.4.1 ([11]). Let $G \in \mathcal{G}$. Then G is bipartite if and only if there exist $V_1 \subset V$, $V_2 \subset V$, such that all the conditions below are satisfied:

- $V_1 \cup V_2 = V$,
- $V_1 \cap V_2 = \emptyset$, and
- for all $(i, j) \in E$, $i \in V_1$ and $j \in V_2$, or $i \in V_2$ and $j \in V_1$.

In that case we say that G has a bipartition (V_1, V_2).

Definition 3.4.2 ([11]). Let $G \in \mathcal{G}$. Then G is K-partite if and only if there exists $V_1, V_2, \ldots, V_K \subset V$, such that all the conditions below are satisfied:

- $V_1 \cup \cdots \cup V_K = V$,
- for all $k, l \in [K]$, where $k \neq l$, $V_k \cap V_l = \emptyset$,
- for all $(i, j) \in E$, $i \in V_k$, $j \in V_l$, where $k, l \in [K]$, and $k \neq l$.

In that case we say that G has a K-partition $(V_1, \ldots, V_K)$.

Definition 3.4.3 ([10]). An Erdös-Rényi graph $G(n, p)$ is a realization of a random graph generated by the Erdös-Rényi model, i.e. it is an unweighted, undirected, simple graph with n nodes, in which, for all unordered pairs $\{i, j\}$ of distinct $i, j \in V$, an edge $(i, j) \in E$ has been generated with probability $p \in [0, 1]$.

3.5 SPECTRAL PROPERTIES OF Δ_r^+

We consider the Rayleigh quotients [46] for Δ_r and Δ_r^+ defined, for $u \in V$, as

$$R(u) := \frac{\langle u, \Delta_r u \rangle_{\mathcal{V}}}{\|u\|_{\mathcal{V}}^2} = \frac{\|\nabla u\|_{\mathcal{E}}^2}{\|u\|_{\mathcal{V}}^2} = \frac{\frac{1}{2}\sum_{i,j\in V} \omega_{ij}(u_i - u_j)^2}{\sum_{i\in V} d_i^r u_i^2},$$
$$R^+(u) := \frac{\langle u, \Delta_r^+ u \rangle_{\mathcal{V}}}{\|u\|_{\mathcal{V}}^2} = \frac{\|\nabla^+ u\|_{\mathcal{E}}^2}{\|u\|_{\mathcal{V}}^2} = \frac{\frac{1}{2}\sum_{i,j\in V} \omega_{ij}(u_i + u_j)^2}{\sum_{i\in V} d_i^r u_i^2},$$

respectively. By Proposition 3.1.1, Δ_r and Δ_r^+ are self-adjoint and positive-semidefinite operators on $\mathcal{V}$, so their eigenvalues will be real and non-negative. The eigenvalues of Δ_r and Δ_r^+ are linked to the extremal values of their Rayleigh quotients by the min-max theorem [21]. In particular, if we denote by $0 \leq \lambda_1 \leq \ldots \leq \lambda_n$ the (possibly repeated) eigenvalues of Δ^+, then $\lambda_1 = \min_{u_1 \in \mathcal{V}\setminus\{0\}} R^+(u_1)$ and $\lambda_n = \max_{u_n \in \mathcal{V}\setminus\{0\}} R^+(u_n)$.

In the following proposition we extend a well-known result for the graph Laplacians [88] to include signless graph Laplacians.

Proposition 3.5.1. *Let $r \in [0, 1]$. The following statements are equivalent:*

1. *λ is an eigenvalue of L_1 with corresponding eigenvector v;*
2. *λ is an eigenvalue of L_s with corresponding eigenvector $D^{1/2}v$;*
3. *$2 - \lambda$ is an eigenvalue of L_1^+ with corresponding eigenvector v;*
4. *$2 - \lambda$ is an eigenvalue of L_s^+ with corresponding eigenvector $D^{1/2}v$;*
5. *λ and v are solutions of the generalized eigenvalue problem $L_r v = \lambda D^{1-r} v$.*

Proof. For $r = 1$ the matrix representations of the graph Laplacian and signless graph Laplacian satisfy $L_1^+ = I + D^{-1}A = 2I - (I - D^{-1}A) = 2I - L_1$. Hence λ is an eigenvalue of L_1 with corresponding eigenvector v if and only if $2 - \lambda$ is an eigenvalue of L_1^+ with the same eigenvector.

Because $L_s = D^{1/2} L_1 D^{-1/2}$, λ is an eigenvalue of L_1 with eigenvector v if and only if λ is an eigenvalue of L_s with eigenvector $D^{1/2}v$. Moreover, since $L_s^+ = 2I - L_s$, we have that $2 - \lambda$ is an eigenvalue of L_s^+ with eigenvector $D^{1/2}v$ if and only if λ is an eigenvalue of L_s with eigenvalue $D^{1/2}v$.

Finally, for $r \in [0,1]$, we have $L_r = D^{1-r}L_1$, hence λ is an eigenvalue of L_1 with corresponding eigenvector v if and only if $L_r v = \lambda D^{1-r} v$. □

Inspired by Proposition 3.5.1, we define, for a given graph $G \in \mathcal{G}$ and node subset $S \subset V$, the rescaled indicator function $\tilde{\chi}_S \in \mathcal{V}$, by, for all $j \in V$,

$$(\tilde{\chi}_S)_j := d_j^{\frac{1}{2}} (\chi_S)_j . \tag{22}$$

Proposition 3.5.2. *The graph $G = (V, E, \omega) \in \mathcal{G}$ has k connected components if and only if $\Delta \in \{\Delta_r, \Delta_s\}$ ($r \in [0,1]$) has eigenvalue 0 with algebraic and geometric multiplicity equal to k. In that case, the eigenspace corresponding to the 0 eigenvalue is spanned by*

- *the indicator functions χ_{S_i}, if $\Delta = \Delta_r$, or*
- *the rescaled indicator functions $\tilde{\chi}_{S_i}$ (as in (22)), if $\Delta = \Delta_s$.*

Here the node subsets $S_i \subset V$, $i \in [k]$, are such that each connected component of G is the subgraph induced by an S_i.

Proof. We follow the proof in [88]. First we consider the case where $\Delta = \Delta_r$, $r \in [0,1]$. We note that Δ_r is diagonizable in the $\mathcal{V}$ inner product and thus the algebraic multiplicity of any of its eigenvalues is equal to its geometric multiplicity. In this proof we will thus refer to both simply as 'multiplicity'.

For any function $u \in \mathcal{V}$ we have that $\langle u, \Delta_r u \rangle_{\mathcal{V}} = \frac{1}{2} \sum_{i,j \in V} \omega_{ij}(u_i - u_j)^2$. We have that 0 is an eigenvalue if and only if there exists a $u \in \mathcal{V} \setminus \{0\}$ such that

$$\langle u, \Delta_r u \rangle_{\mathcal{V}} = 0. \tag{23}$$

This condition is satisfied if and only if, for all $i, j \in V$ for which $\omega_{ij} > 0$, $u_i = u_j$.

Now assume that G is connected (hence G has $k = 1$ connected component), then (23) is satisfied if and only if, for all $i, j \in V$, $u_i = u_j$. Therefore any eigenfunction corresponding to the eigenvalue $\lambda_1 = 0$ has to be constant, e.g. $u = \chi_V$. In particular, the multiplicity of λ_1 is 1.

Now assume that G has $k \geq 2$ connected components, let S_i, $i \in [k]$ be the node sets corresponding to the connected components of the graph. Via

a suitable reordering of nodes G will have a graph Laplacian matrix of the form

$$L_r = \begin{pmatrix} L_r^{(1)} & 0 & \cdots & 0 \\ 0 & L_r^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & L_r^{(k)} \end{pmatrix},$$

where each matrix $L_r^{(i)}$, $i \in \{1, \ldots, k\}$ corresponds to Δ_r restricted to the connected component induced by S_i. This restriction is itself a graph Laplacian for that component. Because each $L_r^{(i)}$ has eigenvalue zero with multiplicity 1, L_r (and thus Δ_r) has eigenvalue 0 with multiplicity k. We can choose the eigenvectors equal to χ_{S_i} for $i \in [k]$ by a similar argument as in the $k = 1$ case.

Conversely, if Δ_r has eigenvalue 0 with multiplicity k, then G has k connected components, because if G has $l \neq k$ connected components, then by the proof above the eigenvalue 0 has multiplicity $l \neq k$.

For Δ_s we use Proposition 3.5.1 to find that the eigenvalues are the same as those of Δ_r, with the corresponding eigenfunctions rescaled as stated in the result. □

Proposition 3.5.3. *Let $G = (V, E, \omega) \in \mathcal{G}$ have k connected components and let the node subsets $S_i \subset V$, $i \in \{1, \ldots, k\}$ be such that each connected component is the subgraph induced by one of the S_i. We denote these subgraphs by G_i. Let $\Delta^+ \in \{\Delta_r^+, \Delta_s^+\}$ ($r \in [0, 1]$) and let $0 \leq k' \leq k$. Then Δ^+ has an eigenvalue equal to 0 with algebraic and geometric multiplicity k' if and only if k' of the subgraphs G_i are bipartite. In that case, assume the labelling is such that G_i, $i \in [k']$ are bipartite with bipartition $(T_i, S_i \setminus T_i)$, where $T_i \subset S_i$. Then the eigenspace corresponding to the 0 eigenvalue is spanned by*

- *the indicator functions $\chi_{T_i} - \chi_{S_i \setminus T_i}$, if $\Delta^+ = \Delta_r^+$, or*
- *the rescaled indicator functions $\tilde{\chi}_{T_i} - \tilde{\chi}_{S_i \setminus T_i}$ (as in (22)), if $\Delta^+ = \Delta_s^+$.*

Proof. First we consider the case where $\Delta^+ = \Delta_r^+, r \in [0,1]$. For any vector $u \in \mathcal{V}$ we have that

$$\langle u, \Delta_r^+ u \rangle_{\mathcal{V}} = \frac{1}{2} \sum_{i,j \in V} \omega_{ij} (u_i + u_j)^2.$$

Let $k = 1$, then $\lambda_1 = 0$ is an eigenvalue if and only if there exists $u \in \mathcal{V} \setminus \{0\}$ such that $\langle u, \Delta_r^+ u \rangle_{\mathcal{V}} = 0$. This condition is satisfied if and only if, for all $i, j \in V$ for which $\omega_{ij} > 0$ we have

$$u_i = -u_j. \tag{24}$$

We claim that this condition in turn is satisfied if and only if G is bipartite. To prove the 'if' part of that claim, assume G is bipartite with bipartition (A, A^c) for some $A \subset V$, and define $u \in \mathcal{V}$ such that $u|_A = -1$ and $u|_{A^c} = 1$. To prove the 'only if' statement, assume G is not bipartite, then there exists an odd cycle in G [11, Theorem 1.4]. Let $i \in V$ be a vertex on this cycle, then by applying condition (24) to all the vertices of the cycle, we find $u_i = -u_i = 0$. Since G is connected, it now follows, by applying condition (24) to all vertices in V, that $u = 0$, which is a contradiction.

The argument above also shows that, if G is bipartite with bipartition (A, A^c), then any eigenfunction corresponding to $\lambda_1 = 0$ is proportional to $u = \chi_A - \chi_{A^c}$. Therefore the eigenvalue 0 has geometric multiplicity 1. Since Δ_r^+ is diagonizable in the $\mathcal{V}$ inner product the algebraic multiplicity of λ_1 is equal to its geometric multiplicity.

Now let $k \geq 2$ and let S_i, $i \in [k]$ be the node sets corresponding to the connected components of the graph. Via a suitable reordering of nodes the graph G will have a signless Laplacian matrix of the form

$$L_r^+ = \begin{pmatrix} L^{(1)+} & 0 & \cdots & 0 \\ 0 & L^{(2)+} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & L^{(k)+} \end{pmatrix},$$

where each matrix $L^{(i)+}$, $i \in [k]$ corresponds to Δ_r^+ restricted to the connected component induced by S_i. This restriction is itself a signless Laplacian for

that connected component. Hence, we can apply the case $k = 1$ to each component separately to find that the (algebraic and geometric) multiplicity of the eigenvalue 0 of Δ_r^+ is equal to the number of connected components which are also bipartite.

If G has $k' \leq k$ connected components which are also bipartite, then, without loss of generality, assume that these components correspond to S_i, $i \in [k']$. Then the corresponding eigenspace is spanned by functions $u^{(i)} = \chi_{T_i} - \chi_{S_i \setminus T_i} \in \mathcal{V}$, $i \in [k']$, where $T_i \subset S_i$ and $(T_i, S_i \setminus T_i)$ is the bipartition of the bipartite component induced by S_i.

For Δ_s^+ we use Proposition 3.5.1 to find the appropriately rescaled eigenfunctions as given in the result. □

Corollary 3.5.4. *The eigenvalues of* Δ_1, Δ_1^+ , Δ_s, *and* Δ_s^+ *are in* $[0, 2]$.

Proof. For Δ_1 the proof can be found in [86, Lemma 2.5]. For completeness we reproduce it here. By Proposition 3.1.1 we know that Δ_1 has non-negative eigenvalues. The upper bound is obtained by maximizing the Rayleigh quotient $R(u)$ over all nonzero $u \in \mathcal{V}$. Since $(u_i - u_j)^2 \leq 2(u_i^2 + u_j^2)$ we have that

$$\max_{u \in \mathcal{V} \setminus \{0\}} R(u) = \max_{u \in \mathcal{V} \setminus \{0\}} \frac{\frac{1}{2} \sum_{i,j \in V} \omega_{ij}(u_i - u_j)^2}{\sum_{i \in V} d_i u_i^2} \leq \max_{u \in \mathcal{V} \setminus \{0\}} \frac{2 \sum_{i \in V} d_i u_i^2}{\sum_{i \in V} d_i u_i^2} = 2.$$

From Proposition 3.5.1 it then follows that the eigenvalues of the other operators are in $[0, 2]$ as well. □

4

MAX-CUT

4.1 Γ-convergence of f_ε^+

In (2) we introduced the signless Ginzburg–Landau functional $f_\varepsilon^+ : \mathcal{V} \to \mathbb{R}$. In this sub-chapter we prove minimizers of f_ε^+ converge to solutions of the Max-Cut problem, using the tools of Γ-convergence [12].

We will prove that f_ε^+ Γ-converges to the functional $f_0^+ : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$, which is defined as

$$f_0^+(u) := \begin{cases} 2\mathrm{TV}^+(u), & \text{if } u \in \mathcal{V}^b, \\ +\infty, & \text{if } u \in \mathcal{V} \setminus \mathcal{V}^b. \end{cases} \tag{25}$$

Lemma 4.1.1. *Let $G \in \mathcal{G}$. For every $\bar{u} \in \mathcal{V}^b$, let $C_{\bar{u}} \in \mathcal{C}$ be the cut induced by $\bar{u}$, then for all $u \in \mathcal{V}$,*

$$f_0^+(u) = \begin{cases} 2\sum_{i,j \in V} \omega_{ij} - 4s(C_u), & \text{if } u \in \mathcal{V}^b, \\ +\infty, & \text{if } u \in \mathcal{V} \setminus \mathcal{V}^b. \end{cases}$$

In particular, if $u^* \in \operatorname{argmin}_{u \in \mathcal{V}} f_\varepsilon^+(u)$, then $u^* \in \mathcal{V}^b$ and C_{u^*} is a maximum cut of G.

Proof. Because, for $u \in \mathcal{V}^b$, $f_0^+(u) = 2\mathrm{TV}^+(u)$ (with $q = 1$), the result follows by (19). □

Lemma 4.1.2. *Let $G \in \mathcal{G}$ and $\varepsilon > 0$. There exist minimizers for the functionals $f_\varepsilon^+ : \mathcal{V} \to \mathbb{R}$ and $f_0^+ : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ from (2) and (25), respectively. Moreover, if $u \in \mathcal{V}$ is a minimizer of f_0^+, then $u \in \mathcal{V}^b$.*

Proof. The potential W satisfies a coercivity condition in the following sense. There exists C_1, $C_2 > 0$ such that, for all $x \in \mathbb{R}$,

$$|x| \geq C_1 \Rightarrow C_2(x^2 - 1) \leq W(x). \tag{26}$$

Combined with the fact that $\|\nabla^+ u\|_{\mathcal{E}} \geq 0$, this shows that f_ε^+ is coercive. Since f_ε^+ is a (multivariate) polynomial, it is continuous. Thus, by the direct method in the calculus of variations [23, Theorem 1.15] f_ε^+ has a minimizer in $\mathcal{V}$.

Since $n \geq 1$, $\mathcal{V}^b \neq \emptyset$ and thus $\inf_{u \in \mathcal{V}} f_0^+(u) < +\infty$. In particular, any minimizer of f_0^+ has to be in $\mathcal{V}^b$. Since $|\mathcal{V}^b| < \infty$ the minimum is achieved. □

Lemma 4.1.3. *Let $G \in \mathcal{G}$ and let f_ε^+ and f_0^+ be as in* (2) *and* (25)*, respectively. Then, as $\varepsilon \downarrow 0$, f_ε^+ Γ-converges to f_0^+.*

Proof. This proof is an adaptation of the proofs in [85, Section 3.1].

Note that

$$f_\varepsilon^+(u) = \frac{1}{2} \sum_{i,j \in V} \omega_{ij}(u_i + u_j)^2 + w_\varepsilon(u),$$

where we define $w_\varepsilon : \mathcal{V} \to \mathbb{R}$ by

$$w_\varepsilon(u) := \frac{1}{\varepsilon} \sum_{i \in V} W(u_i).$$

First we prove that w_ε Γ-converges to w_0 as $\varepsilon \downarrow 0$, where

$$w_0(u) := \begin{cases} 0, & \text{if } u \in \mathcal{V}^b, \\ +\infty, & \text{if } u \in \mathcal{V} \setminus \mathcal{V}^b. \end{cases}$$

Let $\{\varepsilon_k\}_{k \in \mathbb{N}}$ be a sequence of positive real numbers such that $\varepsilon_k \downarrow 0$ as $k \to \infty$ and $u_0 \in \mathcal{V}$.

(LB) Note that, for all $u \in \mathcal{V}$ we have $w_\varepsilon(u) \geq 0$. Let $\{u_k\}_{k=1}^\infty$ be a sequence such that $u_k \to u_0$ as $k \to \infty$. First we assume that $u_0 \in \mathcal{V}^b$, then

$$w_0(u_0) = 0 \leq \liminf_{k \to \infty} w_{\varepsilon_k}(u_k).$$

Next suppose that $u_0 \in \mathcal{V} \setminus \mathcal{V}^b$, then there is an $i \in V$ such that $(u_0)_i \notin \{-1, 1\}$. Since $u_k \to u_0$ as $k \to \infty$, for every $\eta > 0$ there is an $N(\eta) \in \mathbb{N}$ such that for all $k \geq N(\eta)$ we have that $d_i^r |(u_0)_i - (u_k)_i| < \eta$. Define

$$\bar{\eta} := \frac{1}{2} d_i^r \min\{|1 - (u_0)_i|, |-1 - (u_0)_i|\} > 0,$$

then, for all $k \geq N(\bar{\eta})$,

$$|1-(u_k)_i| \geq \big||1-(u_0)_i| - |(u_0)_i-(u_k)_i|\big| \geq \frac{1}{2}|1-(u_0)_i| > 0.$$

Similarly, for all $n \geq N(\bar{\eta})$, $|-1-(u_k)_i| \geq \frac{1}{2}|-1-(u_0)_i| > 0$. Hence, there is a $c > 0$ such that, for all $k \geq N(\bar{\eta})$, $|(u_k)_i| \leq 1-c$. Thus there is a $C > 0$ such that, for all $k \geq N(\bar{\eta})$, $W((u_k)_i) \geq C$. It follows that

$$\liminf_{k\to\infty} w_{\varepsilon_k}(u_k) \geq \liminf_{k\to\infty} \frac{1}{\varepsilon_k} W((u_k)_i) = +\infty = w_0(u_0).$$

(UB) If $u_0 \in \mathcal{V} \setminus \mathcal{V}^b$, then $w_0(u_0)$, and the upper bound condition is trivially satisfied. Now assume $u_0 \in \mathcal{V}^b$. Define the sequence $\{u_k\}_{k=1}^\infty$ by, for all $k \in \mathbb{N}$, $u_k = u_0$. Then, for all $k \in \mathbb{N}$, $w_{\varepsilon_k}(u_k) = 0$ and thus $\limsup_{k\to\infty} w_{\varepsilon_k}(u_k) = 0 = w_0(u_0)$. This concludes the proof that w_ε Γ-converges to w_0 as $\varepsilon \downarrow 0$.

By Proposition 3.3.2, $w_\varepsilon + p$ Γ-converges to $w_0 + p$ for any continuous $p : \mathcal{V} \to \mathbb{R}$. Since $u \mapsto \frac{1}{2}\sum_{i,j\in V} \omega_{ij}(u_i+u_j)^2$ is a polynomial and hence a continuous function on $\mathcal{V}$, we find that, as $\varepsilon \downarrow 0$, f_ε^+ Γ-converges to $g : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$, where

$$g(u) := \frac{1}{2}\sum_{i,j\in V} \omega_{ij}(u_i+u_j)^2 + w_0(u).$$

If $u \in \mathcal{V} \setminus \mathcal{V}^b$, then $g(u) = +\infty$. If $u \in \mathcal{V}^b$, then, for all $i, j \in V$, $(u_i+u_j)^2 = 2|u_i+u_j|$, hence

$$\frac{1}{2}\sum_{i,j\in V} \omega_{ij}(u_i+u_j)^2 = \sum_{i,j\in V} \omega_{ij}|u_i+u_j| = 2\mathrm{TV}^+(u).$$

Thus $g = f_0^+$ and the theorem is proven. □

Lemma 4.1.4. *Let $G \in \mathcal{G}$ and let f_ε^+ be as in (2). Let $\{\varepsilon_k\}_{k\in\mathbb{N}} \subset (0,\infty)$ be a sequence such that $\varepsilon_k \downarrow 0$ as $k \to \infty$, then the sequence $\{f_{\varepsilon_k}^+\}_{k\in\mathbb{N}}$ satisfies the following equi-coerciveness property: If $\{u_k\}_{k\in\mathbb{N}} \subset \mathcal{V}$ is a sequence such that there exists $C > 0$ such that, for all $k \in \mathbb{N}$, $f_{\varepsilon_k}^+(u_k) < C$, then there exists a subsequence $\{u_{k'}\}_{k'\in\mathbb{N}} \subset \{u_k\}_{k\in\mathbb{N}}$ and a $u_0 \in \mathcal{V}^b$ such that $u_{k'} \to u_0$ as $k \to \infty$.*

Proof. This proof closely follows [85, Section 3.1].

From the uniform bound $f_{\varepsilon_k}^+(u_k) < C$, we have that, for all $k \in \mathbb{N}$ and all $i \in V$, $0 \leq W((u_k)_i) \leq C$. Because of the coercivity property (26) of W, the

uniform bound on $W((u_k)_i)$ gives, for all $i \in V$, boundedness of $\{d_i^r (u_k)_i^2\}_{k\in\mathbb{N}}$ and thus $\{u_k\}_{k\in\mathbb{N}}$ is bounded in the $\mathcal{V}$-norm. The result now follows by the Bolzano-Weierstrass theorem. □

With the Γ-convergence and equi-coercivity results from Lemmas 4.1.3 and 4.1.4, respectively, in place, we now prove that minimizers of f_ε^+ converge to solutions of the Max-Cut problem.

Theorem 4.1.5. *Let $G \in \mathcal{G}$. Let $\{\varepsilon_k\}_{k\in\mathbb{N}} \subset (0,\infty)$ be a sequence such that $\varepsilon_k \downarrow 0$ as $k \to \infty$ and, for each $k \in \mathbb{N}$, let $f_{\varepsilon_k}^+$ be as in (2) and let u_{ε_k} be a minimizer of $f_{\varepsilon_k}^+$. Then $\min_{u\in\mathcal{V}} f_{\varepsilon_k}^+(u) \to \min_{u\in\mathcal{V}} f_0^+(u)$ as $k \to \infty$. Furthermore, there exists $u_0 \in \mathcal{V}^b$ and a subsequence $\{u_{\varepsilon_{k'}}\}_{k'\in\mathbb{N}} \subset \{u_{\varepsilon_k}\}_{k\in\mathbb{N}}$, such that $\|u_{\varepsilon_{k'}} - u_0\|_{\mathcal{V}} \to 0$ as $k' \to \infty$. Moreover, $u_0 \in \operatorname{argmin}_{u\in\mathcal{V}} f_0^+(u)$, where f_0^+ is as in (25). In particular, if $C_{u_0} \in \mathcal{C}$ is the cut induced by u_0, then C_{u_0} is a maximum cut of G.*

Proof. By Theorem 3.3.4, the equi-coercivity property of $\{f_{\varepsilon_k}\}_{k\in\mathbb{N}}$ from Lemma 4.1.4 combined with the Γ-convergence property of Lemma 4.1.3 implies that $\min_{u\in\mathcal{V}} f_{\varepsilon_k}^+(u)$ converges to $\min_{u\in\mathcal{V}} f_0^+(u)$ as $k \to \infty$, and, up to taking a subsequence, minimizers of $f_{\varepsilon_k}^+$ converge to a minimizer of f_0^+ as $k \to \infty$. By Lemma 4.1.2, if $u_0 \in \operatorname{argmin}_{u\in\mathcal{V}} f_0^+(u)$, then $u_0 \in \mathcal{V}^b$. By Lemma 4.1.1, the cut C_{u_0} induced by u_0 is a maximum cut of G. □

4.2 THE SIGNLESS MBO ALGORITHM

4.2.1 *Algorithm*

One way of attempting to find minimizers of f_ε^+ is via its gradient flow [4]. This is, for example, the method employed in [9] to find approximate minimizers of f_ε. In that case the gradient flow is given by a graph-based analogue of the Allen–Cahn equation [3]. To find the $\mathcal{V}$-gradient flow of f_ε^+ we compute the first variation of the functional f_ε^+: for $t \in \mathbb{R}$, $u, v \in \mathcal{V}$, we have

$$\frac{d}{dt} f_\varepsilon^+(u + tv)|_{t=0} = \langle \Delta_r^+ u, v\rangle_{\mathcal{V}} + \frac{1}{\varepsilon}\langle D^{-r} W' \circ u, v\rangle_{\mathcal{V}},$$

where we used the notation $(D^{-r}W' \circ u)_i = d_i^{-r}W'(u_i)$. This leads to the following $\mathcal{V}$-gradient flow: for all $i \in V$,

$$\begin{cases} \frac{du_i}{dt} = -(\Delta_r^+ u)_i - \frac{1}{\varepsilon} d_i^{-r} W'(u_i), & \text{for } t > 0, \\ u_i = (u_0)_i, & \text{for } t = 0. \end{cases} \tag{27}$$

Since f_ε^+ is not convex, as $t \to \infty$ the solution of the $\mathcal{V}$-gradient flow is not guaranteed to converge to a global minimum, and can get stuck in local minimizers.

In this thesis we attempt to directly solve the gradient flow equation in Chapter 7.3. However for most tests in this thesis, in order to minimize f_ε^+ we will be focusing on applying a graph MBO type scheme, which we call the signless MBO algorithm. It is given in (MBO+). Despite there currently not being any rigorous results on the matter, the outcome of this scheme is believed to approximate minimizers of f_ε^+ due to the results provided in Chapter 4.1

We see in (MBO+) that in the signless diffusion step the equation that is solved is the gradient flow equation from (27) without the double well potential term. Since we expect the double well potential term in (27) to force the solution to take values close to ± 1, the signless diffusion step in (MBO+) is followed by a thresholding step. Note that, despite our choice of nomenclature, the signless graph 'diffusion' dynamics is expected to be significantly different from standard graph diffusion.

In (MBO+), we choose S_0 at random. The output and run-time of (MBO+) will differ depending on the choice of S_0. However, it is unclear what choices of S_0 will produce the best Max-Cut approximations, and which choices of S_0 will correlate to the fastest possible run time of (MBO+).

In Figures 7 and 8 we show the minimization of f_ε^+ using (MBO+) with the spectral method (which is explained in Chapter 4.2.2) on the AS8 graph and the GNutella09 graph (see Chapter 7.1.3). The (MBO+) iteration numbers j are indicated along the x-axis. The y-axis shows the value of $f_\varepsilon^+(\mu^j)$. The overall tendency we see in both figures is for the (MBO+) algorithm to decrease the value of $f_\varepsilon^+(\mu^j)$, however, in some iterations the value increases.

Algorithm (MBO+): The signless graph MBO algorithm

Data: A signless graph Laplacian $\Delta^+ \in \{\Delta_0^+, \Delta_1^+, \Delta_s^+\}$ corresponding to a graph $G \in \mathcal{G}$, a $\tau > 0$, an initial condition $\mu^0 := \chi_{S_0} - \chi_{S_0^c}$ corresponding to a node subset $S_0 \subset V$, a time step dt, and a stopping criterion tolerance $\eta > 0$.

Output: A sequence of functions $\{\mu^j\}_{j=0}^N \subset \mathcal{V}^b$ giving the signless MBO evolution of μ^0, a sequence of corresponding cuts $\{C^j\}_{j=0}^N \subset \mathcal{C}$ and their sizes $\{s(C^j)\}_{j=0}^N \subset [0, \infty)$, with largest value s^*.

for $j = 1$ **to** stopping criterion is satisfied, **do**

Signless diffusion step: Compute $u^*(\tau)$, where $u^* \in \mathcal{V}$ is the solution of the initial value problem

$$\begin{cases} \frac{du(t)}{dt} = -\Delta^+ u(t), & \text{for } t > 0, \\ u(0) = \mu^{j-1}. \end{cases} \tag{28}$$

Threshold step: Define $\mu^j \in \mathcal{V}^b$ by, for $i \in V$,

$$\mu_i^j := T(u_i^*(\tau)) := \begin{cases} 1, & \text{if } u_i^*(\tau) > 0, \\ -1, & \text{if } u_i^*(\tau) \leq 0. \end{cases} \tag{29}$$

Define the cut $C^j := V_{-1}^j | V_1^j$, where $V_{\pm 1}^j := \{i \in V : \mu_i^j = \pm 1\}$ and compute $s(C^j)$.

Set $N = j$.

if $s(C^j) = \frac{1}{2} \sum_{i,j \in V} \omega_{ij}$ **then**
  Stop

if $\frac{\|\mu^j - \mu^{j-1}\|_2^2}{\|\mu^j\|_2^2} < \eta$ **then**
  Stop

Find the largest cut size: Set $s^* := \max_{1 \leq j \leq N} s(C^j)$.

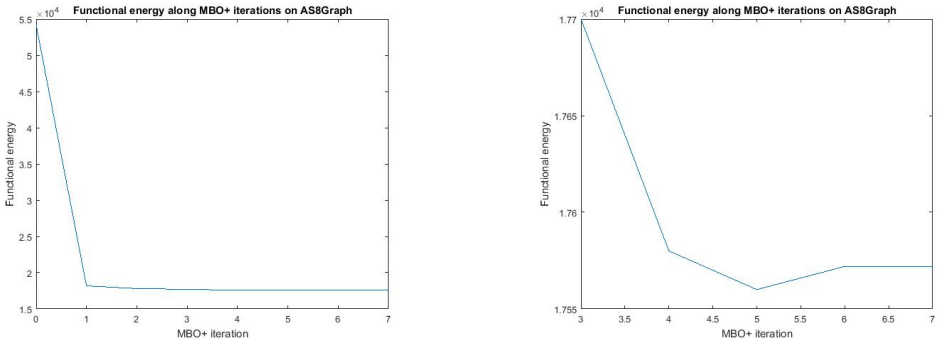


Figure 7.: The value $f_\varepsilon^+(\mu^j)$ as a function of the iteration number j in the (MBO+) scheme on AS8Graph, using the spectral method and Δ_1^+, with $K = 100$, $\tau = 20$, and μ^0 chosen at random. The left hand plot shows the initial condition and all iterations of the (MBO+) scheme on AS8Graph; the right hand plot displays the 3rd to the final iterations of the (MBO+) scheme on AS8Graph.

This is why in (MBO+) we output the cut size which is largest among all iterations computed and use that as the final output, if it outperforms the cut C which (MBO+) returns. Alternatively, in order to save on computing memory, one could also keep track of the largest cut size found so far in each iteration and discard the other cut sizes, or accept the final cut size $s(C^N)$ as approximation to s^* . The results we report in Chapter 7 are all based on the output s^*.

In our experiments we choose the stopping criterion tolerance $\eta = 10^{-8}$.

4.2.2 *Spectral decomposition method*

We will compare two implementations of the (MBO+) algorithm, which differ in the way they solve (28) for $t \in [0, \tau]$. In the next section we consider an explicit Euler method, but first we discuss a spectral decomposition method.

In order to solve (28) we use spectral decomposition of the signless graph Laplacian $\Delta^+ \in \{\Delta_0^+, \Delta_1^+, \Delta_s^+\}$. Let $\lambda_k \geq 0$, $k \in [n]$ be the eigenvalues of Δ^+. We assume $\lambda_1 \leq \lambda_2 \leq \ldots \lambda_n$ and list eigenvalues multiple times according to their multiplicity. Let $\phi^k \in \mathcal{V}$ be an eigenfunction corresponding to λ_k,

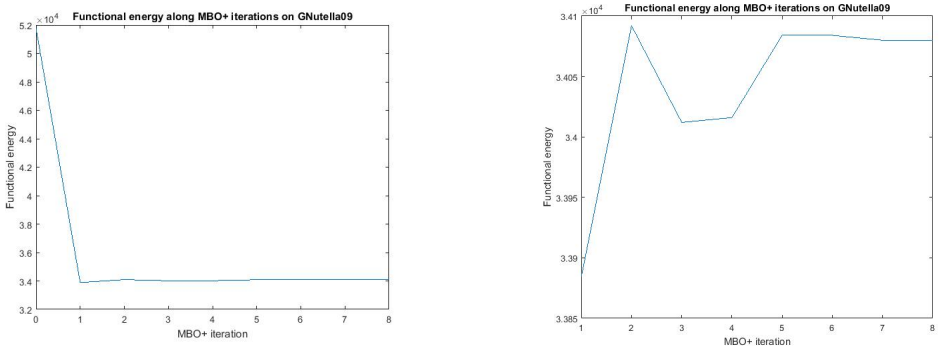


Figure 8.: The value $f_\varepsilon^+(\mu^j)$ as a function of the iteration number j in the (MBO+) scheme on the GNutella09 graph, using the spectral method and Δ_1^+, with $K = 100$, $\tau = 20$, and μ^0 chosen at random. The left hand plot shows the initial condition and all iterations of the (MBO+) scheme on GNutella09; the right hand plot displays all iterations of the (MBO+) scheme on GNutella09, without the initial condition.

chosen such that $\{\phi_k\}_{k=1}^n$ is a set of orthonormal functions in $\mathcal{V}$. We then use the decomposition

$$u^*(\tau) = \sum_{k=1}^{n} e^{-\lambda_k \tau} \langle \phi^k, u(0) \rangle_{\mathcal{V}} \, \phi^k \tag{30}$$

to solve (28).

For Δ_s^+ we use the Euclidean inner product instead of the $\mathcal{V}$ inner product in (30), because the Laplacian Δ_s^+ is not of the form as given in (17).

The optimal choice for τ with respect to the cut size obtained by (MBO+) is a topic for future research. Based on trial and error, we decided to use $\tau = 20$ in the results we present in Chapter 7, when using Δ_1^+ or Δ_s^+ as our operator. When using Δ_0^+ as our operator, we use $\tau = \frac{40}{\lambda_n}$, where λ_n is the largest eigenvalue of Δ_0^+. The division of τ by half of the largest eigenvalue of Δ_0^+ is justified in Chapter 4.2.4. In Chapter 7.2 and Chapter 7.4 we investigate how cut sizes change with varying τ.

A computational advantage of the spectral decomposition method is that we do not necessarily need to use all of the eigenvalues and eigenfunctions of the signless Laplacian. We may use far fewer eigenfunctions, say k^* eigen-

functions corresponding to the smallest eigenvalues in our decomposition (30), with $k^* << n$. To be explicit, doing this replaces n in (30) by k^*. In Chapter 7.2 we show how increasing k^* beyond a certain point has little effect on the size of the cut obtained by (MBO+) for three examples. We refer to using the k^* eigenfunctions corresponding to the smallest eigenvalues in the decomposition as spectral truncation.

By Proposition 3.5.1, we can compute the k^* smallest eigenvalues λ_k ($k \in [k^*]$) of Δ_1^+ and Δ_s^+ by first computing the k^* largest eigenvalues $\hat{\lambda}_l$ ($l \in \{n-K+1,\dots,n\}$) of Δ_1 and Δ_s respectively instead and then setting $\lambda_k = 2 - \hat{\lambda}_{n-k+1}$. There is not a similar property for Δ_0^+ however. Proving upper bounds on the largest eigenvalues of Δ_0 and Δ_0^+ is of interest [41].

We use the MATLAB `eigs` function to calculate the k^* eigenpairs of the signless Laplacian. This function [61] uses the Implicitly Restarted Arnoldi Method (IRAM) [80], which in practice can efficiently compute the largest eigenvalues and corresponding eigenvectors of sparse matrices. The function `eigs` firstly computes the orthogonal projection of the matrix you want eigenpairs from, and a random vector, onto the matrix's K-dimensional Krylov subspace. This projection is represented by a smaller $k^* \times k^*$ matrix. Then `eigs` calculates the eigenvalues of this $k^* \times k^*$ matrix, whose eigenvalues are called Ritz eigenvalues. The Ritz eigenvalues are computed efficiently using a QR method [31]. Computationally these Ritz eigenvalues typically approximate the largest eigenvalues of the original matrix. The time complexity of IRAM is currently unknown, but in practice it produces approximate eigenpairs efficiently.

If the matrix of which the eigenvalues are to be computed is symmetric, the MATLAB `eigs` function simplifies to the Implicitly Restarted Lanczos Method (IRLM) [17], therefore typically in practice `eigs` will usually compute the eigenvalues and eigenfunctions of Δ_s^+ faster than those of Δ_1^+.

Using the IRLM for computing the eigenpairs of Δ_0^+ corresponding to its smallest eigenvalues is inefficient. In our experiments using the MATLAB `eig` function to calculate all eigenpairs of Δ_0^+ and choosing the k^* eigenpairs

corresponding to the smallest eigenvalues for the decomposition (30) was faster than using the IRLM to calculate the k^* eigenpairs of Δ_0^+. Hence, the results discussed in this paper are obtained with `eig` when using Δ_0^+ and `eigs` when using Δ_1^+ or Δ_s^+.

If we use the MATLAB `eigs` function when using our spectral decomposition method we cannot a priori determine the time complexity for (MBO+), because practical experiments have shown the complexity of the IRAM and IRLM methods is heavily dependent on the matrix to which they are applied [75]. If we choose to use the MATLAB `eig` function then the time complexity of (MBO+) is $\mathcal{O}(n^3)$, which is the time complexity of computing all eigenpairs of an $n \times n$ matrix [38]. All other remaining steps of (MBO+) require fewer operations to compute.

4.2.3 *Explicit Euler method*

We also compute the solution of (28) for $t \in [0, \tau]$ using an explicit finite difference scheme,

$$\begin{cases} u^{m+1} = u^m - \Delta^+ u^m dt, & \text{for } m \in \{0, 1, \dots, M\}, \\ u^0 = u(0) \end{cases} \tag{31}$$

for the same choice of τ as in (30). For $M \in \mathbb{N}$, $dt = \frac{\tau}{M}$, and we set $u^*(\tau) = u^M$.

If $G \in \mathcal{G}$ then (MBO+) using the Euler method will have a time complexity of $\mathcal{O}(|E|)$, because of the sparsity of the signless Laplacian matrix. When zero entries are ignored, the multiplication of the vector u^m by Δ^+ takes $4|E| + 2n$ operations to compute. Since $G \in \mathcal{G}$ has no isolated nodes, $|E| \geq n - 1$, therefore, when n is large enough, $4|E| > 2n$ and hence the time complexity of the multiplication is $\mathcal{O}(|E|)$. All other remaining steps in (MBO+) using the Euler method require fewer operations to compute.

In Chapter 7.2.3 we show some results for (MBO+) when solving (28) using an implicit finite difference scheme,

$$\begin{cases} u^{m+1} = u^m - \Delta^+ u^{m+1} dt, & \text{for } m \in \{0, 1, \dots, M\}, \\ u^0 = u(0). \end{cases} \tag{32}$$

We compare the results of (MBO+) obtained using (32) to solve (28) against the results of (MBO+) using (31) to solve (28).

4.2.4 *(MBO+) pinning condition*

For (MBO+), choosing τ too small causes trivial dynamics in the sense that, for any j, $\mu^j = \mu^0$ in (MBO+). In this sub-chapter we prove that such a τ dependent on the largest eigenvalue of the signless Laplacian chosen for (MBO+).

We define $d_- := \min_{i \in V} d_i$, and $d_+ := \max_{i \in V} d_i$. Let $\Delta^+ \in \{\Delta_0^+, \Delta_1^+, \Delta_s^+\}$, then the operator norm $\|\Delta^+\|_{\mathcal{V}}$ is defined by

$$\|\Delta^+\|_{\mathcal{V}} := \sup_{u \in \mathcal{V} \setminus \{0\}} \frac{\|\Delta^+ u\|_{\mathcal{V}}}{\|u\|_{\mathcal{V}}}$$

We define the maximum norm of $\mathcal{V}$ by $\|u\|_{\mathcal{V},\infty} := \max\{|u_i| : i \in V\}$.

Lemma 4.2.1. *Let $\Delta^+ \in \{\Delta_0^+, \Delta_1^+, \Delta_s^+\}$. The operator norm $\|\Delta^+\|_{\mathcal{V}}$ and the largest eigenvalue λ_n of Δ^+ are equal. This implies that, for all $u \in \mathcal{V}$,*

$$\|\Delta^+ u\|_{\mathcal{V}} \leq \lambda_n \|u\|_{\mathcal{V}}.$$

Proof. See [86, Lemma 2.5]. □

Lemma 4.2.2. *The norms $\|\cdot\|_{\mathcal{V}}$ and $\|\cdot\|_{\mathcal{V},\infty}$ are equivalent, with optimal constants given by*

$$d_-^{\frac{r}{2}} \|u\|_{\mathcal{V},\infty} \leq \|u\|_{\mathcal{V}} \leq \|\chi_V\|_{\mathcal{V}} \|u\|_{\mathcal{V},\infty}.$$

Proof. See [86, Lemma 2.2]. □

Theorem 4.2.3. *Let $G \in \mathcal{G}$, and let λ_n be the largest eigenvalue of the signless Laplacian $\Delta^+ \in \{\Delta_0^+, \Delta_1^+, \Delta_s^+\}$. Let $S_0 \subset V$, $\mu^0 := \chi_{S_0} - \chi_{S_0^c}$, and let $\mu^1 \in \mathcal{V}^b$ be*

the result of applying one (MBO+) iteration (i.e one application of (28) *and* (29)*) to* μ^0*. If*

$$\tau < \lambda_n^{-1}\log(1 + d_-^{\frac{r}{2}}\|\chi_V\|_{\mathcal{V}}^{-1}),$$

then $\mu^1 = \mu^0$*.*

Proof. This proof closely follows the proof of a similar result in [86, Section 4.2].

If $\|e^{-\tau\Delta^+}\mu^0 - \mu^0\|_{\mathcal{V},\infty} < 1$, then $\mu^1 = \mu^0$. Using Lemma 4.2.2, we compute

$$\|e^{-\tau\Delta^+}\mu^0 - \mu^0\|_{\mathcal{V},\infty} \leq d_-^{-\frac{r}{2}}\|e^{-\tau\Delta^+}\mu^0 - \mu^0\|_{\mathcal{V}} \leq d_-^{-\frac{r}{2}}\|e^{-\tau\Delta^+} - \mathrm{Id}\|_{\mathcal{V}}\,\|\mu^0\|_{\mathcal{V}}.$$

Moreover, since $\langle \chi_{S_0}, \chi_{S_0^c}\rangle_{\mathcal{V}} = 0$, we have $\|\mu^0\|_{\mathcal{V}}^2 = \|\chi_{S_0}\|_{\mathcal{V}}^2 + \|\chi_{S_0^c}\|_{\mathcal{V}}^2 = \|\chi_{S_0} + \chi_{S_0^c}\|_{\mathcal{V}}^2 = \|\chi_V\|_{\mathcal{V}}^2$.

Using the triangle inequality and the submultiplicative property (see [78] for example) of $\|\cdot\|_{\mathcal{V}}$, we compute $\|e^{-\tau\Delta^+} - \mathrm{Id}\|_{\mathcal{V}} \leq \sum_{k=1}^{\infty}\frac{1}{k!}(\tau\|\Delta^+\|_{\mathcal{V}})^k = e^{\lambda_n\tau} - 1$. Therefore, if $\tau < \lambda_n^{-1}\log(1 + d_-^{\frac{r}{2}}\|\chi_V\|^{-1})$, then $\mu^1 = \mu^0$. □

As stated in Chapter 4.2.2, we choose $\tau = 20$ as diffusion time for (MBO+) using Δ_1^+ or Δ_s^+, and $\tau = \frac{40}{\lambda_n}$ when using (MBO+) with Δ_0^+ as the choice of operator. This is due to $\tau = 20$ often being too large when using (MBO+) with Δ_0^+. Choosing $\tau = 20$ for (MBO+) using Δ_0^+ causes the solution to converge to $u(\tau) = 0$ to machine precision. We therefore choose $\tau = \frac{40}{\lambda_n}$ for Δ_0^+ since Theorem 4.2.3 implies a suitable choice of τ for (MBO+) with respect to obtaining non-trivial output cuts is inversely proportional to the largest eigenvalue of the chosen operator Δ^+. Since $\lambda_n = 2$ for Δ_1^+ and Δ_s^+ we choose to divide τ by $\frac{\lambda_n}{2}$ for Δ_0^+.

4.3 Γ-convergence of the Lyapunov functional for signless MBO evolutions

4.3.1 *Lyapunov functional*

In this sub-chapter, in (33), we introduce a Lyapunov functional for (MBO+). We prove that, when properly rescaled, this functional Γ-converges to the

signless total variation. In our setup and proofs we follow similar work in [86, 87] which in turn was inspired by results in the continuum setting obtained in [28]. We first introduce some needed notation.

Define the space of functions $\mathcal{K} := \{u \in \mathcal{V} : \forall i \in V, u_i \in [-1,1]\}$. In analogy with [86] we use our operators to define a graph curvature using our signless definitions. If $C = S|S^c$ we define the signless normal of a set by

$$\nu_{ij}^S := \operatorname{sgn}((\nabla^+(\chi_S - \chi_{S^c})_{ij}) = \begin{cases} 1, & \text{if } i,j \in S, \\ -1, & \text{if } i,j \in S^c, \\ 0, & \text{otherwise,} \end{cases}$$

and our curvature as

$$(\kappa_S^{r,+})_i := (\operatorname{div}^+ \nu^S)_i = d_i^{-r} \begin{cases} \sum_{j\in S} \omega_{ij}, & \text{if } i \in S, \\ -\sum_{j\in S^c} \omega_{ij}, & \text{if } i \in S^c. \end{cases}$$

By direct computation we note that $\Delta^+(\chi_S - \chi_{S^c})_i = 2\kappa_S^{r,+}$. We define the volume of a node set $S \subseteq V$ by

$$\operatorname{Vol} S := \|\chi_S\|_{\mathcal{V}}^2 = \sum_{i\in S} d_i^r.$$

Proposition 4.3.1. *Let $G \in \mathcal{G}$ and for $r \in [0,1]$, let $\Delta^+ = \Delta_r^+$. Then, for all $u \in \mathcal{V} \setminus \{0\}$ we have that*

$$\langle u, e^{-\tau\Delta^+} u\rangle_{\mathcal{V}} = \sum_{i\in V} e^{-\tau\lambda_i} \langle u, \phi^i\rangle_{\mathcal{V}}^2 > 0.$$

Proof. This has been proved in [87] using the operator Δ instead of Δ^+. Our proof here, which we include for completeness, is similar. We expand u on the $\mathcal{V}$-orthonormal basis formed by the eigenfunctions $\{\phi^i\}_{i=1}^n$:

$$\begin{aligned} \langle u, e^{-\tau\Delta^+} u\rangle_{\mathcal{V}} &= \left\langle \sum_{i\in V} \langle u, \phi^i\rangle_{\mathcal{V}}\, \phi^i, \sum_{j\in V} e^{-\tau\lambda_j} \langle u, \phi^j\rangle_{\mathcal{V}}\, \phi^j \right\rangle_{\mathcal{V}} \\ &= \sum_{i,j\in V} e^{-\tau\lambda_j} \langle u, \phi^i\rangle_{\mathcal{V}}\, \langle u, \phi^j\rangle_{\mathcal{V}}\, \delta_{ij} = \sum_{i\in V} e^{-\tau\lambda_i} \langle u, \phi^i\rangle_{\mathcal{V}}^2, \end{aligned}$$

where δ_{ij} is the Kronecker delta function. This expression is strictly positive since u is not the zero function. □

Proposition 4.3.2. *Let $G \in \mathcal{G}$ and let $C \in \mathcal{C}$. Then for $S \subseteq V$ we have that*

$$\mathrm{TV}^+(\chi_S - \chi_{S^c}) = \langle \kappa_S^{r,+}, \chi_S - \chi_{S^c} \rangle_{\mathcal{V}}.$$

Proof. By direct computation we show that

$$\begin{aligned}\langle \kappa_S^{r,+}, \chi_S - \chi_{S^c} \rangle_{\mathcal{V}} &= d_i^r \sum_{i \in S} (\sum_{j \in S} d_i^{-r} \omega_{ij}) - d_i^r \sum_{i \in S^c} (- \sum_{j \in S^c} d_i^{-r} \omega_{ij}) \\ &= \sum_{i,j \in S} \omega_{ij} + \sum_{i,j \in S^c} \omega_{ij}.\end{aligned}$$

We see that if $C = S|S^c$, then

$$\begin{aligned}\sum_{i,j \in S} \omega_{ij} + \sum_{i,j \in S^c} \omega_{ij} = \sum_{i,j \in V} \omega_{ij} - 2(\sum_{i \in S, j \in S^c} \omega_{ij}) &= \sum_{i,j \in V} \omega_{ij} - 2s(C) \\ &= \mathrm{TV}^+(\chi_S - \chi_{S^c}).\end{aligned}$$

□

Proposition 4.3.3. *Let $G \in \mathcal{G}$ and, for $r \in [0,1]$, let $\Delta^+ = \Delta_r^+$. Let (λ_i, ϕ^i) $(i \in [n])$ denote the eigenpairs of Δ^+, where the eigenfunctions are $\mathcal{V}$-orthonormal. Then for $S \subseteq V$ we have that*

$$\mathrm{TV}^+(\chi_S - \chi_{S^c}) = \frac{1}{2} \sum_{i \in V} \lambda_i \langle \chi_S - \chi_{S^c}, \phi^i \rangle_{\mathcal{V}}^2.$$

Proof. By Proposition 4.3.2 we have

$$\mathrm{TV}^+(\chi_S - \chi_{S^c}) = \langle \kappa_S^{1,r,+}, \chi_S - \chi_{S^c} \rangle_{\mathcal{V}},$$

which can be expressed as

$$\left\langle \kappa_S^{1,r,+}, \sum_{i \in V} \langle \chi_S - \chi_{S^c}, \phi^i \rangle_{\mathcal{V}} \, \phi^i \right\rangle_{\mathcal{V}} = \sum_{i \in V} \langle \kappa_S^{1,r,+}, \phi^i \rangle_{\mathcal{V}} \, \langle \chi_S - \chi_{S^c}, \phi^i \rangle_{\mathcal{V}}.$$

We have that $\Delta^+(\chi_S - \chi_{S^c}) = 2\kappa_S^{1,r,+}$, therefore

$$\mathrm{TV}^+(\chi_S - \chi_{S^c}) = \frac{1}{2} \sum_{i \in V} \langle \Delta^+(\chi_S - \chi_{S^c}), \phi^i \rangle_{\mathcal{V}} \, \langle \chi_S - \chi_{S^c}, \phi^i \rangle_{\mathcal{V}}.$$

Since Δ^+ is self-adjoint in the $\mathcal{V}$ inner product by Proposition 3.1.1, we have that

$$\begin{aligned}&\frac{1}{2} \sum_{i \in V} \langle \Delta^+(\chi_S - \chi_{S^c}), \phi^i \rangle_{\mathcal{V}} \, \langle \chi_S - \chi_{S^c}, \phi^i \rangle_{\mathcal{V}} \\ &= \frac{1}{2} \sum_{i \in V} \langle \chi_S - \chi_{S^c}, \Delta^+ \phi^i \rangle_{\mathcal{V}} \, \langle \chi_S - \chi_{S^c}, \phi^i \rangle_{\mathcal{V}} \\ &= \frac{1}{2} \sum_{i \in V} \langle \chi_S - \chi_{S^c}, \lambda_i \phi^i \rangle_{\mathcal{V}} \, \langle \chi_S - \chi_{S^c}, \phi^i \rangle_{\mathcal{V}} = \frac{1}{2} \sum_{i \in V} \lambda_i \langle \chi_S - \chi_{S^c}, \phi^i \rangle_{\mathcal{V}}^2.\end{aligned}$$

□

Lemma 4.3.4. *Let $G \in \mathcal{G}$, $\tau > 0$, and for $r \in [0,1]$, let $\Delta^+ = \Delta_r^+$. Define $J_\tau : \mathcal{V} \to \mathbb{R}$ by*

$$J_\tau(u) := -\langle u, e^{-\tau\Delta^+} u\rangle_{\mathcal{V}} + \operatorname{Vol} V. \tag{33}$$

Then the following hold.

1. *The functional $J_\tau(u)$ is strictly concave and Fréchet differentiable.*

2. *Its derivative at $u \in \mathcal{V}$ in the direction of $v \in \mathcal{V}$ is given by*

$$dJ_\tau^u(v) = -2\langle e^{-\tau\Delta^+} u, v\rangle_{\mathcal{V}}.$$

3. *Let $\mu^0 := \chi_{S_0} - \chi_{S_0^c}$ for some $S_0 \subset V$ and let $\{\mu^j\}_{j=1}^N$ be a sequence generated by (MBO+). For all $j \in [N]$, μ^j minimizes $dJ_\tau^{\mu^{j-1}}(\cdot)$ over $\mathcal{K}$:*

$$\mu^j \in \operatorname*{argmin}_{v\in\mathcal{K}} dJ_\tau^{\mu^{j-1}}(v). \tag{34}$$

4. *For all $j \in [N]$, $J_\tau(\mu^j) \leq J_\tau(\mu^{j-1})$, with equality attained if and only if $\mu^j = \mu^{j-1}$.*

Remark 4.3.5. The final property in Lemma 4.3.4 shows that J_τ is a Lyapunov functional for (MBO+). All the results in Lemma 4.3.4 remain true if a constant term is added to J_τ. In particular, they remain true if the term $\operatorname{Vol} V$ is removed from (33). We included this term in the definition of J_τ, however, because it is needed in Chapter 4.3.2 for our Γ-convergence results.

Proof of Lemma 4.3.4. In [86, Lemma 4.5, Proposition 4.6] and [87, Lemma 5.5] the authors derive similar Lyapunov functionals for two graph based MBO schemes similar to (MBO+). This proof is similar to those proofs.

We first compute the Gateaux derivative at $u \in \mathcal{V}$ in the direction of $v \in \mathcal{V}$. Let $\varepsilon \in \mathbb{R}$, then

$$dJ_\tau^u(v) := \frac{dJ_\tau(u+\varepsilon v)}{d\varepsilon}|_{\varepsilon=0} = -2\langle e^{-\tau\Delta^+} u, v\rangle_{\mathcal{V}},$$

where we have applied the fact that Δ^+, and thus also $e^{-\tau\Delta^+}$ are self-adjoint. Since $u \mapsto dJ_\tau^u(v)$ is continuous, J_τ is also Fréchet differentiable. Moreover, if $v \in \mathcal{V}\backslash\{0\}$ the second directional derivative is given by

$$\frac{d^2 J_\tau(u+\varepsilon v)}{d\varepsilon^2}|_{\varepsilon=0} = -2\langle v, e^{-\tau\Delta^+} v\rangle_{\mathcal{V}}$$

which, by Proposition 4.3.1, is strictly negative. Therefore J_τ is strictly concave.

To prove the third property, let $j \in [N]$. Then μ^j is generated by (MBO+) and thus it takes values in $\{-1,1\}$. Let $\mu^* \in \mathcal{K} \setminus \{\mu^j\}$ and let $i \in V$ be such that $\mu_i^* \neq \mu_i^j$. Assume first that $\mu_i^j = 1$ and hence $\mu_i^* \in [-1,1)$. Then, by (28), (29), we have $\left(e^{-\tau\Delta^+}\mu^{j-1}\right)_i > 0$ and thus $-2\left(e^{-\tau\Delta^+}\mu^{j-1}\right)_i d_i^r \mu_i^j < -2\left(e^{-\tau\Delta^+}\mu^{j-1}\right)_i d_i^r \mu_i^*$. If, on the other hand $i \in V$ is such that $\mu_i^j = -1$ and hence $\mu_i^* \in (-1,1]$, then $\left(e^{-\tau\Delta^+}\mu^{j-1}\right)_i \leq 0$ and thus $-2\left(e^{-\tau\Delta^+}\mu_i^{j-1} d_i^r \mu^j\right)_i \leq -2\left(e^{-\tau\Delta^+}\mu^{j-1}\right)_i d_i^r \mu_i^*$. Hence, for all $\mu^* \in \mathcal{K} \setminus \{\mu^j\}$, we have $dJ_\tau^{\mu^{j-1}}(\mu^j) \leq dJ_\tau^{\mu^{j-1}}(\mu^*)$ and therefore μ^j is a minimizer.

Since J_τ is strictly concave and $dJ_\tau^{\mu^{j-1}}$ is linear we conclude that if $\mu^j \neq \mu^{j+1}$, then

$$J_\tau(\mu^{j+1}) - J_\tau(\mu^j) < dJ_\tau^{\mu^j}(\mu^{j+1} - \mu^j) = dJ_\tau^{\mu^j}(\mu^{j+1}) - dJ_\tau^{\mu^j}(\mu^j) \leq 0.$$

The inequality follows from (34). If $\mu^{j+1} = \mu^j$ then $J_\tau(\mu^{j+1}) - J_\tau(\mu^j) = 0$. □

Remark 4.3.6. Our proof of (34) above also shows that minimizers in (34) need not be unique: at vertices $i \in V$ where $\left(e^{-\tau\Delta^+}\mu^{j-1}\right)_i \neq 0$ any minimizer has to agree with μ^j obtained by (MBO+), but at vertices i where $\left(e^{-\tau\Delta^+}\mu^{j-1}\right)_i = 0$ minimizers can take any value in $[-1,1]$.

4.3.2 Γ-*convergence proof*

Let $r \in [0,1]$ and $\Delta^+ = \Delta_r^+$. For $\tau > 0$ we define a rescaled version of J_τ, restricted to $\mathcal{K}$: define $\hat{J}_\tau : \mathcal{K} \to \mathbb{R} \cup \{\infty\}$ by $\hat{J}_\tau := \frac{1}{\tau} J_\tau|_{\mathcal{K}}$, i.e.

$$\hat{J}_\tau(u) := -\frac{1}{\tau}\langle u, e^{-\tau\Delta^+} u\rangle_{\mathcal{V}} + \frac{1}{\tau}\mathrm{Vol}V. \tag{35}$$

We prove that for $\tau \downarrow 0$ this functional Γ-converges to $\hat{J}_0 : \mathcal{K} \to \mathbb{R} \cup \{\infty\}$ defined by

$$\hat{J}_0(u) := \begin{cases} 2\mathrm{TV}^+(u), & \text{if } u \in \mathcal{V}^b, \\ +\infty, & \text{if } u \in \mathcal{K} \backslash \mathcal{V}^b, \end{cases} \tag{36}$$

where $q = 1$. Note that $\hat{J}_0 = f_0^+|_{\mathcal{K}}$, where f_0^+ is as in Lemma 4.1.1. We use the same notion of convergence in $\mathcal{V}$ as in Chapter 4.1.

Lemma 4.3.7. *Let $G \in \mathcal{G}$ and let $\hat{J}_\tau$ and $\hat{J}_0$ be as in* (35) *and* (36)*, respectively. Then, as $\tau \downarrow 0$, $\hat{J}_\tau$ Γ-converges to $\hat{J}_0$.*

Proof. (LB) By Proposition 4.3.1 we can express (35) as

$$\begin{aligned} \hat{J}_\tau(u) &= -\frac{1}{\tau} \sum_{i \in V} e^{-\tau\lambda_i} \langle u, \phi^i \rangle_{\mathcal{V}}^2 + \frac{1}{\tau} \mathrm{Vol}\, V \\ &= -\sum_{i \in V} \frac{e^{-\tau\lambda_i} - 1}{\tau} \langle u, \phi^i \rangle_{\mathcal{V}}^2 - \frac{1}{\tau} \sum_{i \in V} \langle u, \phi^i \rangle_{\mathcal{V}}^2 + \frac{1}{\tau} \mathrm{Vol}\, V. \end{aligned}$$

Since $\tau_k \downarrow 0$ as $k \to \infty$, we have $\dfrac{e^{-\tau_k\lambda_i} - 1}{\tau_k} \to \left.\dfrac{d}{dt} e^{-t\lambda_i}\right|_{t=0}$ as $k \to \infty$. Moreover, since $u_k \to u_0$ as $k \to \infty$, we deduce that, for all $i \in V$, $-\lim\limits_{k\to\infty} \dfrac{e^{-\tau_k\lambda_i} - 1}{\tau_k} \langle u_k, \phi^i \rangle_{\mathcal{V}}^2 = \lambda_i \langle u_0, \phi^i \rangle_{\mathcal{V}}^2$. It follows that

$$\liminf_{k\to\infty} \hat{J}_\tau(u_k) = \sum_{i \in V} \lambda_i \langle u_0, \phi^i \rangle_{\mathcal{V}}^2 + \liminf_{k\to\infty} \frac{1}{\tau_k} \left(\mathrm{vol}\, V - \sum_{i \in V} \langle u_k, \phi^i \rangle_{\mathcal{V}}^2 \right). \tag{37}$$

Since $u_k \in \mathcal{K}$, we have, for all $i \in V$, $(u_k)_i^2 \leq 1$ and thus

$$\sum_{i \in V} \langle u_k, \phi^i \rangle_{\mathcal{V}}^2 = \langle u_k^2, \chi_V \rangle_{\mathcal{V}} \leq \langle \chi_V, \chi_V \rangle_{\mathcal{V}} = \mathrm{Vol}\, V. \tag{38}$$

For the first equality we have used [87, Lemma 4.6] (which in its original form is formulated for eigenfunctions of the graph Laplacian, but holds without further change for any functions ϕ^i which form an orthonormal basis of $\mathcal{V}$). Hence $\liminf\limits_{k\to\infty} \dfrac{1}{\tau_k} \left(\mathrm{vol}\, V - \sum\limits_{i \in V} \langle u_k, \phi^i \rangle_{\mathcal{V}}^2 \right) \geq 0$.

Now assume that $u_0 \in \mathcal{V}^b$, then $u_0 = \chi_S - \chi_{S^c}$ for some $S \subseteq V$. It follows from the above, combined with Proposition 4.3.3, that

$$\liminf_{k\to\infty} \hat{J}_\tau(u_k) \geq \sum_{i \in V} \lambda_i \langle u_0, \phi^i \rangle_{\mathcal{V}}^2 = 2\mathrm{TV}^+(u_0) = \hat{J}_0(u_0).$$

Next assume that $u_0 \in \mathcal{K}\backslash\mathcal{V}^b$. Then there is a $j \in V$ such that $(u_0)_j \in (-1,1)$. Since $u_k \to u_0$ as $k \to \infty$, there exists an $\eta \in (0,1)$ such that, for all k large enough, $(u_k)_j^2 \in [0, 1-\eta]$. Using the fact that, for all $i \in V$, we have $d_i^r(1-(u_k)_i^2) \geq 0$, we find that, for large k,

$$\begin{aligned}\operatorname{vol} V - \sum_{i\in V} \langle u_k, \phi^i\rangle_{\mathcal{V}}^2 = \sum_{i\in V}\left(d_i^r - d_i^r (u_k)_i^2\right) &\geq d_j^r(1-(u_k)_j^2) \\ &\geq d_j^r(1-(1-\eta)) = \eta d_j^r > 0.\end{aligned}$$

Therefore, using (37), we have

$$\liminf_{k\to\infty} \hat{J}_{\tau_k}(u_k) \geq \liminf_{k\to\infty} \frac{1}{\tau_k} d_j^r \eta = +\infty = \hat{J}_0(u_0).$$

(UB) If $u_0 \in \mathcal{K}\backslash\mathcal{V}^b$ then the upper bound is trivially satisfied. If $u_0 \in \mathcal{V}^b$, then we define our recovery sequence $\{u_k\}$ such that, for all $k \in \mathbb{N}$, $u_k := u_0$. As in the proof of (LB) above, we have that

$$\begin{aligned}\limsup_{k\to\infty} \hat{J}_\tau(u_0) &= \sum_{i\in V} \lambda_i \langle u_0, \phi^i\rangle_{\mathcal{V}}^2 + \limsup_{k\to\infty} \frac{1}{\tau_k}\left(\operatorname{vol} V - \sum_{i\in V}\langle u_0, \phi^i\rangle_{\mathcal{V}}^2\right) \\ &= 2\mathrm{TV}^+(u_0) = \hat{J}_0(u_0),\end{aligned}$$

where we used Proposition 4.3.3 for the second equality, combined with the fact that, as in (38), $\sum_{i\in V}\langle u_0, \phi^i\rangle_{\mathcal{V}}^2 = \langle u_0^2, \chi_V\rangle_{\mathcal{V}} = \langle \chi_V, \chi_V\rangle_{\mathcal{V}} = \operatorname{Vol} V$. □

Lemma 4.3.8. *Let $G \in \mathcal{G}$ and let $\{\tau_k\}_{k\in\mathbb{N}}$ be a sequence of positive real numbers such that $\tau_k \downarrow 0$ as $k \to \infty$ and let $\{u_k\}_{k\in\mathbb{N}}$ be a sequence in $\mathcal{K}$. Assume that there exists a $C > 0$ such that for all $k \in \mathbb{N}$ we have $\hat{J}_\tau(u_k) \leq C$. Then there exists a subsequence $\{u_{k'}\}_{k'\in\mathbb{N}} \subset \{u_k\}_{k\in\mathbb{N}}$ and a $u_0 \in \mathcal{V}^b$ such that $u_{k'} \to u_0$ as $k' \to \infty$.*

Proof. This proof closely resembles the proof in [87, Theorem 5.23].

As explained in Chapter 3.1.1, to each $u \in \mathcal{V}$ we can assign a canonical vector representation $\hat{u} \in \mathbb{R}^n$. Let $\|u\|_2$ denote the Euclidean norm of $\hat{u}$. Since $\{u_k\}_{k\in\mathbb{N}} \subset \mathcal{K}$ we have that for all $k \in \mathbb{N}$, $0 \leq \|u_k\|_2 \leq \sqrt{n}$. By the Bolzano-Weierstrass theorem it follows that a subsequence $\{u_{k'}\}_{k'\in\mathbb{N}} \subset \{u_k\}_{k\in\mathbb{N}}$ exists which converges to a limit $u_0 \in \mathcal{V}$ with respect to $\|\cdot\|_2$ and hence also with respect to the topologically equivalent norm $\|\cdot\|_{\mathcal{V}}$. Moreover,

by componentwise convergence and preservation of inequalities in the limit, $\mathcal{K}$ is closed (with respect to either norm) and thus $u_0 \in \mathcal{K}$.

Assume that $u_0 \in \mathcal{K} \setminus \mathcal{V}^b$, then there exists a $j \in V$ such that $(u_0)_j \in (-1, 1)$ and thus there exists an $\eta \in (0, 1)$ such that, for all large enough $l' \in \mathbb{N}$, $(u_{l'})_j^2 \in [0, 1 - \eta]$. Since for all $i \in [n]$ we have $\lambda_i \geq 0$, it follows by (38) that

$$\begin{aligned} \operatorname{Vol} V - \sum_{i \in V} e^{-\tau_{l'} \lambda_i} \langle u_{l'}, \phi^i \rangle_{\mathcal{V}}^2 &\geq \operatorname{Vol} V - \sum_{i \in V} \langle u_{l'}, \phi^i \rangle_{\mathcal{V}}^2 \\ &= \operatorname{Vol} V - \langle u_{l'}^2, \chi_V \rangle_{\mathcal{V}}^2 = \langle \chi_V - u_{l'}^2, \chi_V \rangle_{\mathcal{V}} \\ &\geq \eta d_j^r. \end{aligned} \tag{39}$$

Moreover, by assumption and using Proposition 4.3.1 we have, for all $k' \in \mathbb{N}$,

$$\tau_{k'} \hat{J}_{\tau_{k'}}(u) = \operatorname{Vol} V - \sum_{i \in V} e^{-\tau_{k'} \lambda_i} \langle u_{k'}, \phi^i \rangle_{\mathcal{V}}^2 \leq C \tau_{k'}.$$

Choose l' to be large enough such that $C\tau_{l'} < \eta d_j^r$ and large enough so that (39) holds. Then we arrive at a contradiction and thus $u_0 \in \mathcal{V}^b$. □

Theorem 4.3.9. *Let $G \in \mathcal{G}$. Let $\{\tau_k\}_{k \in \mathbb{N}} \subset (0, \infty)$ be a sequence such that $\tau_k \downarrow 0$ as $k \to \infty$ and, for each $k \in \mathbb{N}$, let $\hat{J}_{\tau_k}$ be as in* (35) *and let u_{τ_k} be a minimizer of $\hat{J}_{\tau_k}$. Then $\min_{u \in \mathcal{K}} \hat{J}_{\tau_k}(u) \to \min_{u \in \mathcal{K}} \hat{J}_0(u)$ as $k \to \infty$. Furthermore, there exists $u_0 \in \mathcal{V}^b$ and a subsequence $\{u_{\tau_{k'}}\}_{k' \in \mathbb{N}} \subset \{u_{\tau_k}\}_{k \in \mathbb{N}}$, such that $\|u_{\tau_{k'}} - u_0\|_{\mathcal{V}} \to 0$ as $k' \to \infty$. Moreover, $u_0 \in \operatorname{argmin}_{u \in \mathcal{K}} \hat{J}_0(u)$, where $\hat{J}_0$ is as in* (36). *In particular, if $C_{u_0} \in \mathcal{C}$ is the cut induced by u_0, then C_{u_0} is a maximum cut of G.*

Proof. By Theorem 3.3.4, the equi-coercivity proven in Lemma 4.3.8 and the Γ-convergence of Lemma 4.3.7 together imply that $\min_{u \in \mathcal{K}} \hat{J}_{\tau_k}(u) \to \min_{u \in \mathcal{K}} \hat{J}_0(u)$ as $k \to \infty$. Furthermore, up to a subsequence, minimizers of $\hat{J}_{\tau_k}$ converge to a minimizer of $\hat{J}_0$ as $k \to \infty$. By Lemma 4.3.8, this minimizer, u_0 is in $\mathcal{V}^b$. The final statement follows in the same way as in the proof of Theorem 4.1.5. □

Remark 4.3.10. Even though the results of Theorem 4.3.9 suggest that minimizers of $\hat{J}_\tau$ (or J_τ) for small $\tau > 0$ are good approximations to the solution of the Max-Cut problem, this is not a guarantee that solutions of the (MBO+) scheme are as well. In fact, as we have seen in Chapter 4.2.4, for small $\tau > 0$ the (MBO+) dynamics is trivial and thus will not produce an (approximate) solution to the Max-Cut problem unless the initial condition already is one.

5

MAX-K-CUT

5.1 Γ-convergence of $F^+_{\varepsilon,K}$

In (7) we introduced the multiclass signless Ginzburg–Landau functional $F^+_{\varepsilon,K} : \mathcal{V}^K \to \mathbb{R}$. In Chapter 5.1 we prove that minimizers of $F^+_{\varepsilon,K}$ converge to solutions of the Max-K-Cut problem, using similar Γ-convergence techniques to the ones used in Chapter 4.1. For this chapter we recall the definitions of Ω_K and Ω^*_K from Chapter 3.2.2.

We will prove that $F^+_{\varepsilon,K}$ Γ-converges to the functional $F^+_{0,K} : \mathcal{V}^K \to \mathbb{R} \cup \{\infty\}$ which is defined by

$$F^+_{0,K} := \begin{cases} \|\widetilde{\nabla}^+ U\|^2_{\mathcal{E}^K}, & \text{if } U \in \Omega_K, \\ +\infty, & \text{if } U \in \mathcal{V}^K \backslash \Omega_K, \end{cases} \tag{40}$$

Lemma 5.1.1. *Let $G \in \mathcal{G}$ and $\varepsilon > 0$. There exist minimizers for the functionals $F^+_{\varepsilon,K} : \mathcal{V}^K \to \mathbb{R}$, $F^+_{0,K} : \mathcal{V}^K \to \mathbb{R} \cup \{+\infty\}$ from (7) and (40) respectively. Moreover, if $U \in \mathcal{V}^K$ is a minimizer of $F^+_{0,K}$, then $U \in \Omega_K$.*

Proof. The potential $W^{\pm}_{\varepsilon,K}$ from (8) satisfies a coercivity condition in the following sense. There exist a $C_1 > 0$ and a $C_2 > 0$ such that, for all $U \in \mathcal{V}^K$, and for all $i \in V$,

$$\|U\|_{\mathcal{V}^K} \geq C_1 \Rightarrow C_2\Big(\sum_{k\in[K]} U^2_{ik}\Big) \leq W^{\pm}_{\varepsilon,K}(U). \tag{41}$$

Combined with the fact that we have $\|\widetilde{\nabla}^+ U\|_{\mathcal{E}^K} \geq 0$, this shows that $F^+_{\varepsilon,K}$ is coercive. Since $F^+_{\varepsilon,K}$ is a (multivariate) polynomial, it is continuous. Thus,

by the direct method in the calculus of variations [23, Theorem 1.15] $F^+_{\varepsilon,K}$ has minimizers in $\mathcal{V}^K$.

Since $n \geq 1$, $\Omega_K \neq \emptyset$ and thus $\inf_{u\in\mathcal{V}^K} F^+_{0,K}(u) < +\infty$. In particular, any minimizer of $F^+_{0,K}$ has to be in Ω_K. Since $|\Omega_K| < \infty$ the minimum is achieved.

□

Lemma 5.1.2. *Let $G \in \mathcal{G}$. If $U_X \in \operatorname{argmin}_{U\in\mathcal{V}^K} F^+_{0,K}(U)$, and for $L \in \{2,\dots,K\}$, if U_X induces an L-cut, then X is a maximum L-cut of G.*

Proof. By Lemma 5.1.1, since $U_X \in \operatorname{argmin}_{U\in\mathcal{V}^K} F^+_{0,K}(U)$, it follows that $U_X \in \Omega_K$. Furthermore, for $L \in \{2,\dots,K\}$, if U_X induces an L-cut, then $F^+_{0,K}(U_X) = 2(K-2)\sum_{i,j\in V}\omega_{ij} + 4\sum_{k\in[L]}\sum_{i,j\in V_k}\omega_{ij}$ by Lemma 3.2.6. Since U_X minimizes $F^+_{0,K}$, $\sum_{k\in[L]}\sum_{i,j\in V_k}\omega_{ij}$ will be minimized. Any L-cut value can be calculated by $S_L(X) = \frac{1}{2}\sum_{i,j\in V}\omega_{ij} - \frac{1}{2}\sum_{k\in[L]}\sum_{i,j\in V_k}\omega_{ij}$; as $\sum_{k\in[L]}\sum_{i,j\in V_k}\omega_{ij}$ is minimized, then $S_L(X)$ will be maximized, and X is a maximum L-cut of G.

□

Lemma 5.1.3. *Let $G \in G$ and let $F^+_{\varepsilon,K}$ and $F^+_{0,K}$ be as in* (7) *and* (40) *respectively. Then, as $\varepsilon \downarrow 0$, $F^+_{\varepsilon,K}$ Γ-converges to $F^+_{0,K}$.*

Proof. The ideas in this proof are very similar to [85] and Lemma 4.1.3.

First we prove that $W^\pm_{\varepsilon,K}$ Γ-converges to $W^\pm_{0,K}$ as $\varepsilon \downarrow 0$, where

$$W^\pm_{0,K}(U) := \begin{cases} 0, & \text{if } U \in \Omega_K, \\ +\infty & \text{if } U \in \mathcal{V}^K \backslash \Omega_K. \end{cases}$$

Let $\{\varepsilon_p\}_{p\in\mathbb{N}}$ is a sequence of positive real numbers such that $\varepsilon_p \downarrow 0$ as $p \to \infty$ and $U_0 \in \Omega_K$.

(LB) Note that, for all $U \in \mathcal{V}^K$ we have $W^\pm_{\varepsilon,K}(U) \geq 0$. Let $\{U_p\}_{p=1}^\infty$ be a sequence such that $U_p \to U_0$ as $p \to \infty$. First we assume that $U_0 \in \Omega_K$, then

$$W^\pm_{0,K}(U_0) = 0 \leq \liminf_{p\to\infty} W^\pm_{\varepsilon_p,K}(U_p).$$

Next suppose that $U_0 \in \mathcal{V}^K \setminus \Omega_K$, then there is an $i \in V$ such that for all $k \in [K]$, $(U_0)_i \neq E'_k$. Since $U_p \to U_0$ as $p \to \infty$, for every $\eta > 0$ there is

an $N(\eta) \in \mathbb{N}$ such that for all $p \geq N(\eta)$ we have that $d_i^r|(U_p)_i - (U_0)_i| < \eta$. Define

$$\bar{\eta} = \min_{k\in[K]} d_i^r|E'_k - (U_0)_i| > 0.$$

Then, for all $p \geq N(\bar{\eta})$ and for all $k \in [K]$ we have that

$$|E'_k - (U_p)_i| \geq \big||E'_k - (U_0)_i| - |(U_0)_i - (U_p)_i|\big| \geq \frac{1}{2}\bar{\eta} > 0.$$

Therefore, there is a $C > 0$ such that for all $p \geq N(\bar{\eta})$ and for all $k \in [K]$, we have that $|(U_p)_i - E'_k|_1^2 \geq C$. It follows that

$$\liminf_{p\to\infty} W^{\pm}_{\varepsilon_p,K}(U_p) \geq \liminf_{p\to\infty} \frac{1}{\varepsilon_p} \prod_{k\in[K]} \frac{1}{16}|(U_p)_i - E'_k|_1^2 = \infty = W^{\pm}_{0,K}(U_0).$$

(UB) If $U_0 \in \mathcal{V}^K \setminus \Omega_K$, then $W^{\pm}_{0,K}(U_0) = \infty$ and the upper bound condition is trivially satisfied. Now assume $U_0 \in \Omega_K$. For all $p \in \mathbb{N}$, define the sequence $\{U_p\}_{p=1}^{\infty}$ where $U_p = U_0$. Then, for all $p \in \mathbb{N}$, $W^{\pm}_{\varepsilon_p,K}(U_p) = 0$ and thus $\limsup_{p\to\infty} W^{\pm}_{\varepsilon_p,K}(U_p) = 0 = W^{\pm}_{0,K}(U_0)$.

Therefore, $W^{\pm}_{\varepsilon,K}$ Γ-converges to $W^{\pm}_{0,K}$ as $\varepsilon \downarrow 0$.

By Propostion 3.3.2, Γ-convergence is stable under continuous perturbations; this means that for any continuous $p : \mathcal{V}^K \to \mathbb{R}$, we have that as $\varepsilon \downarrow 0$, $W^{\pm}_{\varepsilon,k} + p$ will Γ-converge to $W^{\pm}_{0,K} + p$. Since $\|\widetilde{\nabla}^+ U\|_{\mathcal{E}^K}$ is continuous on $\mathcal{V}^K$, as $\varepsilon \downarrow 0$, $F^+_{\varepsilon,K}$ Γ-converges to $F^+ : \mathcal{V}^K \to \cup\{\infty\}$, where

$$F^+(U) = \|\widetilde{\nabla}^+ U\|_{\mathcal{E}^K} + W^{\pm}_{0,K}.$$

If $U \in \mathcal{V}^K \setminus \Omega_K$, then $F^+(U) = \infty = F^+_{0,K}(U)$. If $U \in \Omega_K$, then $F^+(U) = \|\widetilde{\nabla}^+ U\|_{\mathcal{E}^K} = F^+_{0,K}(U)$, which concludes the proof.

□

Lemma 5.1.4. *Let $G \in \mathcal{G}$ and let $F^+_{\varepsilon,K}$ be as in (7). Let $\{\varepsilon_p\}_{p\in\mathbb{N}} \subset (0,\infty)$ be a sequence such that $\varepsilon_p \downarrow 0$ as $p \to \infty$, then the sequence $\{F^+_{\varepsilon_p,K}\}_{p\in\mathbb{N}}$ satisfies the following equi-coerciveness property: If $\{U_p\}_{p\in\mathbb{N}} \subset \mathcal{V}^K$ is a sequence such that there exists $C > 0$ such that, for all $p \in \mathbb{N}$, $F^+_{\varepsilon_p,K}(U_p) < C$ holds, then there exists a subsequence $\{U'_p\}_{p'\in\mathbb{N}} \subset \{U_p\}_{p\in\mathbb{N}}$ and a $U_0 \in \Omega_K$ such that $U_{p'} \to U_0$ as $p \to \infty$.*

Proof. This proof closely follows [85] and Lemma 4.1.4. From the uniform bound $F^+_{\varepsilon_p,K}(U_p) < C$, we have that for all $p \in N$ and all $i \in V$, $0 \leq W^{\pm}_{\varepsilon_p,K}((U_p)_i) \leq C$. Since $W^{\pm}_{\varepsilon,K}$ is coercive, the uniform bound on $W^{\pm}_{\varepsilon_p,K}((U_p)_i)$ gives, for all $i \in V$ and for all $k' \in [K]$, boundedness of $\{d_i^r(U_p)^2_{ik'}\}_{p\in\mathbb{N}}$, and thus, $\{U_p\}_{p\in\mathbb{N}}$ is bounded in the $\mathcal{V}^K$ norm. The result now follows by the Bolzano-Weierstrass theorem.

□

With the Γ-convergence result of Lemma 5.1.3 and the equi-coercive result of Lemma 5.1.4, we can now prove that minimizers of $F^+_{\varepsilon,K}$ converge to solutions of the Max-K-Cut problem.

Theorem 5.1.5. *Let $G \in \mathcal{G}$. Let $\{\varepsilon_p\}_{p\in\mathbb{N}}$ be a sequence such that $\varepsilon_p \downarrow 0$ as $p \to \infty$ and, for each $p \in \mathbb{N}$, let $F^+_{\varepsilon_p,K}$ be as in (7). Let U_{ε_p} be a minimizer of $F^+_{\varepsilon_p,K}$. Then $\min_{U\in\mathcal{V}^K} F^+_{\varepsilon_p,K}(U) \to \min_{U\in\mathcal{V}^K} F^+_{0,K}(U)$ as $k \to \infty$. Furthermore, there exists $U_0 \in \Omega_K$ and a subsequence $\{U_{\varepsilon_{p'}}\}_{p'\in\mathbb{N}} \subset \{U_{\varepsilon_p}\}_{p\in\mathbb{N}}$ such that $\|U_{\varepsilon_{p'}} - U_0\|_{\mathcal{V}^K} \to 0$ as $p' \to \infty$. Moreover, $U_0 \in \operatorname{argmin}_{U\in\mathcal{V}^K} F^+_{0,K}(U)$, where $F^+_{0,K}$ is as in (40). In particular, if $X_{U_0} \in \mathcal{X}^K$ is the L-cut induced by U_0, where $L \in [K]$, then X_{U_0} is a maximum L-cut of G.*

Proof. By Theorem 3.3.4, the equi-coercivity property of $\{F^+_{\varepsilon_p,K}\}_{p\in\mathbb{N}}$ from Lemma 5.1.4, combined with the Γ-convergence property of Lemma 5.1.3 implies that as $p \to \infty$, $\min_{U\in\mathcal{V}^K} F^+_{\varepsilon_p,K}(U) \to \min_{U\in\mathcal{V}^K} F^+_{0,K}(U)$. Furthermore, up to taking a subsequence, as $p \to \infty$ the minimizers of $F^+_{\varepsilon_p,K}$ will converge to minimizers of $F^+_{0,K}$. By Lemma 5.1.1, if $U_0 \in \operatorname{argmin}_{U\in\mathcal{V}^K} F^+_{0,K}$, then $U_0 \in \Omega_K$. By Lemma 5.1.2, the cut X_{U_0} will be a maximum L-cut of G.

□

As seen above, when setting $K = 2$, as $\varepsilon \downarrow 0$, the minimizers of the functional $F^+_{\varepsilon,2}$ will converge to solutions to the Max-2-Cut problem, which is better known as the Max-Cut problem. In Chapter 8.4 we minimize $F^+_{\varepsilon,2}$ and compare the results to the ones obtained by minimizing f^+_ε in (2).

As shown in Lemma 5.1.1, for all $G \in \mathcal{G}$, all minimizers of $F^+_{0,K}$ will be contained in Ω_K. However, for many graphs in $G \in \mathcal{G}$, all of the minimizers

of $F^+_{0,K}$ will in fact be contained in Ω^*_K. We formalise this in the following proposition:

Proposition 5.1.6. *Let $G \in \mathcal{G}$. Then all minimizers of $F^+_{0,K}$ will be contained in Ω^*_K, if and only if G is not K-partite.*

Proof. First assume that G is not K-partite. As shown in Proposition 3.2.5, for all $K \in \mathbb{N}\backslash\{1\}$, if G is not $(K-1)$-partite, and $X' \in \mathcal{X}^{K-1}$, such that $S_{K-1}(X') = m_{K-1}(G)$, there exists an $X \in \mathcal{X}^K$ such that $S_{K-1}(X') < S_K(X)$.

Let $X = V_1|\dots|V_K$ and $X' = V'_1|\dots|V'_K$, where $V'_K = \emptyset$. Since $S_{K-1}(X') < S_K(X)$, it follows that

$$\sum_{k\in[K]}\sum_{i,j\in V_k}\omega_{ij} < \sum_{k\in[K]}\sum_{i,j\in V'_k}\omega_{ij},$$

meaning $F^+_{0,K}(U_X) < F^+_{0,K}(U_{X'})$. It follows that if $U_X \in \operatorname*{argmin}_{U\in\mathcal{V}^K} F^+_{0,K}(U)$, then U_X must be in Ω^*_K.

Next assume that G is L-partite for some $L < K$. Let $X = V_1|\dots|V_K$ be an L-partition of G, where without loss of generality, assume that for $k \in [K]\backslash[L]$, $V_k = \emptyset$. Let U_X be the function that induces X. Since X is an L-partition, for all $i, j \in V$, if $(i,j) \in E$, there exists a $k \in [K]$ such that $i, j \in V_k$, meaning $\sum_{k\in[K]}\sum_{i,j\in V_k} = 0$. Therefore, $F^+_0(U_X) = 2(K-2)\sum_{i,j\in V}\omega_{ij}$. Furthermore, $F^+_{0,K}$ is minimized, and there exists a $U_X \in \Omega_K$ such that $U_X \in \operatorname*{argmin}_{U\in\mathcal{V}^K} F^+_{0,K}(U)$. □

5.2 MULTICLASS ALGORITHMS

5.2.1 *Concave-convex splitting*

In this sub-chapter we introduce the (KCut) algorithm. As inspired by the algorithm produced in [65] to find minimizers of (4), our algorithm (KCut) consists of solving the gradient flow equation of $F^+_{\varepsilon,K}$ for a small time step, and thresholding to the closest corner of $\Sigma^\pm_K$. As shown in [22], the closest corner of $\Sigma^\pm_K$ to U_i will be $E'_{k'}$, where $k' \in \operatorname*{argmax}_{k\in[K]} U_{ik}$.

To find the $\mathcal{V}^K$-gradient flow of $F^+_{\varepsilon,K}$ we compute the first variation of the functional $F^+_{\varepsilon,K}$: for $t \in \mathbb{R}$, $U, V \in \mathcal{V}^K$, we have

$$\frac{d}{dt}F^+_{\varepsilon,K}(U+tV)|_{t=0} = \langle \widetilde{\Delta}^+_r U, V\rangle_{\mathcal{V}^K} + \frac{1}{\varepsilon}\langle D^{-r}W^{\pm}_{\varepsilon,K}{}' \circ U, V\rangle_{\mathcal{V}^K},$$

where we have used the notation $(D^{-r}W^{\pm}_{\varepsilon,K}{}' \circ U)_i = d_i^{-r}W^{\pm}_{\varepsilon,K}{}'(U)_i$. This leads to the following $\mathcal{V}^K$-gradient flow: for all $i \in V$,

$$\begin{cases} \frac{dU_i}{dt} = -(\widetilde{\Delta}^+_r U)_i - \frac{1}{\varepsilon}d_i^{-r}W^{\pm}_{\varepsilon,K}{}'(U)_i, & \text{for } t > 0, \\ U_i(0) = (U_0)_i, & \text{for } t = 0. \end{cases} \tag{42}$$

Along iterations of (KCut), it is possible for the output cut to be an L-cut, where $L \leq K$. We define the notation $\mathcal{X}^{[K]}$ to be the set of all possible L-cuts on $\mathcal{G}$, where $L \in [K]$. If $X \in \mathcal{X}^{[K]}$, then we denote the cut size by $S_{[K]}(X)$.

For both (KCut) and (KMBO+) (as introduced in Chapter 5.2.2), the initial condition μ^0 is chosen at random. The outputs and run-times of (KCut) and (KMBO+) will differ depending on the choice of μ^0. However, we are unaware of what choices of μ^0 will be optimal with respect to the run-times and the size of the cuts induced by (KCut) and (KMBO+).

In [90] the authors introduce the Concave-Convex Procedure (CCCP) minimization scheme for functionals with a bounded Hessian. This scheme is monotonically decreasing, unconditionally stable, and guaranteed to converge to a minimum or a saddle point of the functional. In [33] the authors apply this method to solve the gradient flow equation of (4) for a small time step. The method we use to solve (42) is similar:

For $\widetilde{\Delta}^+ \in \{\widetilde{\Delta}^+_0, \widetilde{\Delta}^+_1, \widetilde{\Delta}^+_s\}$, we express the functional (7) as a sum of convex and concave parts:

$$F^+_{\varepsilon,K} = F^{+,convex}_{\varepsilon,K} + F^{+,concave}_{\varepsilon,K}, \text{ where}$$

$$F^{+,convex}_{\varepsilon,K}(U) := \langle U, \widetilde{\Delta}^+ U\rangle_{\mathcal{V}^K} + C\langle U, U\rangle_{\mathcal{V}^K}, \text{ and}$$

$$F^{+,concave}_{\varepsilon,K}(U) := \frac{1}{\varepsilon}\sum_{i\in V}\Big(\prod_{k\in[K]}\frac{1}{16}|U_i - E'_k|_1^2\Big) - C\langle U, U\rangle_{\mathcal{V}^K},$$

where $C \in \mathbb{R}$ is chosen in order to assure that the above terms are convex and concave. If Δ^+_s is chosen as the choice of signless Laplacian, we use

Algorithm (KCut): The spectral K-cut algorithm

Data: A signless graph Laplacian $\widetilde{\Delta}^+ \in \{\widetilde{\Delta}_0^+, \widetilde{\Delta}_1^+, \widetilde{\Delta}_s^+\}$ corresponding to a graph $G \in \mathcal{G}$, a time step $\tau > 0$, an initial condition $\mu^0 \in \Omega_K$, a choice of $\varepsilon > 0$, and a stopping tolerance η.

Output: A sequence of functions $\{\mu^j\}_{j=0}^N \subset \Omega_K$, a sequence of corresponding cuts $\{X^j\}_{j=0}^N \subset \mathcal{X}^{[K]}$, their cut sizes $\{S_{[K]}(X^j)\}_{j=0}^N \subset [0, \infty)$, with largest value S^*.

for $j = 1$ **to** stopping criterion is satisfied, **do**

Gradient flow step: Compute $U^*(\tau)$, where μ^{j-1} is the initial condition of (42), and $U^* \in \mathcal{V}^K$ is the solution of (42).

Projection to Ω_K step: Define $\mu^j \in \Omega_K$ by, for $i \in V$,

$$\mu_i^j := E_{k'}, \text{ where } k' \in \operatorname*{argmax}_{k \in [K]} U_{ik'}^*(\tau)$$

Define the cut $X^j := V_{X_1}^j | \dots | V_{X_K}^j$, where $V_{X_k}^j := \{i \in V : (\mu^j)_{ik} = 1\}$, and compute $S_{[K]}(X^j)$.

Set $N = j$.

if $S_{[K]}(X^j) = \frac{1}{2} \sum_{i,j \in V} \omega_{ij}$ **then**
  Stop

if $\frac{\|\mu^j - \mu^{j-1}\|_2^2}{\|\mu^j\|_2^2} < \eta$ **then**
  Stop

Find the largest cut size: Set $S^* := \max_{1 \le j \le N} S_{[K]}(X^j)$.

the Euclidean inner product over the $\mathcal{V}^K$ inner product in the $F_{\varepsilon,K}^{+,concave}$ and $F_{\varepsilon,K}^{+,convex}$ terms, and we set $r = 0$ for the equations (43) and (44) below.

As stated in [90] in order to guarantee stability the gradient descent is implicit for $F_{\varepsilon,K}^{+,convex}$ and explicit for $F_{\varepsilon,K}^{+,concave}$, meaning for the iteration $n \in \mathbb{N}$ and for $i \in V$, $k \in [K]$, our scheme is

$$U_{ik}^{n+1} + dt\frac{\delta F_{\varepsilon,K}^{+,convex}}{\delta U_{ik}}(U_{ik}^{n+1}) = U_{ik}^{n} - dt\frac{\delta F_{\varepsilon,K}^{+,concave}}{\delta U_{ik}}(U_{ik}^{n}).$$

which is expressed by the equation

$$U^{n+1} + dt(\widetilde{\Delta}^{+}(U^{n+1}) + CU^{n+1}) = U^{n} - dt(\frac{1}{\varepsilon}D^{-r}T(U^{n}) - CU^{n}). \tag{43}$$

Here, D^{-r} is the diagonal matrix where, for $i \in V$, $(D^{-r})_{ii} = d_i^{-r}$, and

$$T(U)_{ik} = \sum_{l=1}^{K}\frac{1}{8}(1-2\delta_{kl})\|U_i - E_l\|_{L^1}\prod_{m=1,m\neq l}^{K}\frac{1}{16}\|U_i - E_m\|_{L_1}^2.$$

We denote L^+ as the matrix canonization of $\widetilde{\Delta}^+$ and I_n is the $n \times n$ identity matrix. From this, (43) has the update equation

$$U^{n+1} = B^{-1}\big((1 + CD^{-r}dt)U^{n} - \frac{dt}{\varepsilon}D^{-r}T(U^{n})\big) \tag{44}$$

where

$$B = (1 + Cdt)I_n + dtL^{+}. \tag{45}$$

As with the spectral decomposition method for (MBO+) in Chapter 4.2.2, and as stated in [33, 65] for their scheme to minimize E in (4), we can apply spectral truncation on the signless graph Laplacian in order to reduce the computation time of the convex-concave scheme. Choosing k^* as the number of eigenpairs we use in the spectral truncation of L^+, if $\bar{\Lambda}$ is an $k^* \times k^*$ matrix containing the k^* smallest eigenvalues of L^+, and $\bar{X}$ denotes their corresponding eigenvectors, we replace (45) in (44) with

$$B = \bar{X}[(1 + Cdt)I_{k^*} + dt\bar{\Lambda}]\bar{X}' \tag{46}$$

The computation of (44) results in the solution of (42), with U^n as the initial condition.

5.2.2 *Multiclass MBO*

In this sub-chapter we discuss the (KMBO+) algorithm, which is similar to the MBO scheme proposed in [65] to find minimizers of (4). The (KMBO+) algorithm differs from (KCut) in that instead of solving (42) for a small time-step and an initial condition $U_0 \in \Omega_K$, we instead solve

$$\begin{cases} \frac{dU(t)}{dt} = -\widetilde{\Delta}_r^+ U(t), \quad \text{for } t > 0, \\ U_i(0) = (U_0)_i. \end{cases} \tag{47}$$

for a small step. As in (KCut), we then threshold the solution to the corners of $\Sigma_K^{\pm}$. For completeness, we propose the following (KMBO+) algorithm:

Solving (47) is very similar to solving (28) in Chapter 4.2.1. We use spectral decomposition of the signless graph Laplacian $\widetilde{\Delta}^+ \in \{\widetilde{\Delta}_0^+, \widetilde{\Delta}_1^+, \widetilde{\Delta}_s^+\}$. Let $\lambda_k \geq 0$, $k \in [n]$ be the eigenvalues of $\widetilde{\Delta}^+$. We assume $\lambda_1 \leq \lambda_2 \leq \ldots \lambda_n$ and list eigenvalues multiple times according to their multiplicity. Let $\phi^k \in \mathcal{V}^K$ be an eigenfunction corresponding to λ_k, chosen such that $\{\phi_k\}_{k=1}^n$ is a set of orthonormal functions in $\mathcal{V}^K$. We then use the decomposition

$$U^*(\tau) = \sum_{k=1}^{n} e^{-\lambda_k \tau} \langle \phi^k, U(0) \rangle_{\mathcal{V}^K} \phi^k \tag{48}$$

to solve (47). If Δ_s^+ is chosen as the signless graph Laplacian, then we choose the Euclidean inner product over the $\mathcal{V}^K$ inner product in (48)

Similarly to the spectral truncation applied in Chapter 4.2.2 to reduce the computation time in solving (30), we can reduce the computation time of (KMBO+) by using the k^* eigenfunctions corresponding to the smallest eigenvalues of $\widetilde{\Delta}^+$ in solving (48).

In Figure 9 we show two examples of the minimization of $F_{\varepsilon,K}^+$ along iterations of (KCut) and in Figure 10 we show two examples of the minimization of $F_{\varepsilon,K}^+$ along iterations of (KMBO+). Figure 9a and Figure 10a plots the value of $F_{\varepsilon,4}^+(\mu^j)$ for a realisation of $G(200, 0.05)$ against the iteration j of (KCut) and (KMBO+) respectively. Figure 9b and Figure 10b plots the value of $F_{\varepsilon,3}^+(\mu^j)$ for a realisation of $R(200, 3, 0.1, 0.9)$ against the iteration j of (KCut) and (KMBO+) respectively. As we can see, both (KCut) and (KMBO+) have

Algorithm (KMBO+): The signless K-dimensional MBO algorithm

Data: A signless graph Laplacian $\widetilde{\Delta}^+ \in \{\widetilde{\Delta}_0^+, \widetilde{\Delta}_1^+, \widetilde{\Delta}_s^+\}$ corresponding to a graph $G \in \mathcal{G}$, a time step $\tau > 0$, an initial condition $\mu^0 \in \Omega_K$, and a stopping tolerance η.

Output: A sequence of functions $\{\mu^j\}_{j=0}^N \subset \Omega_K$, a sequence of corresponding cuts $\{X^j\}_{j=0}^N \subset \mathcal{X}^{[K]}$, their cut sizes $\{S_{[K]}(X^j)\}_{j=0}^N \subset [0,\infty)$, with largest value S^*.

for $j = 1$ **to** stopping criterion is satisfied, **do**

Gradient flow step: Compute $U^*(\tau)$, where μ^{j-1} is the initial condition of (47), and $U^* \in \mathcal{V}^K$ is the solution of (47).

Projection to Ω_K step: Define $\mu^j \in \Omega_K$ by, for $i \in V$,

$$\mu_i^j := E_{k'}, \text{ where } k' \in \operatorname*{argmax}_{k \in [K]} U_{ik'}^*(\tau)$$

Define the cut $X^j := V_{X_1}^j | \dots | V_{X_K}^j$, where $V_{X_k}^j := \{i \in V : (\mu^j)_{ik} = 1\}$ and compute $S_{[K]}(X^j)$.

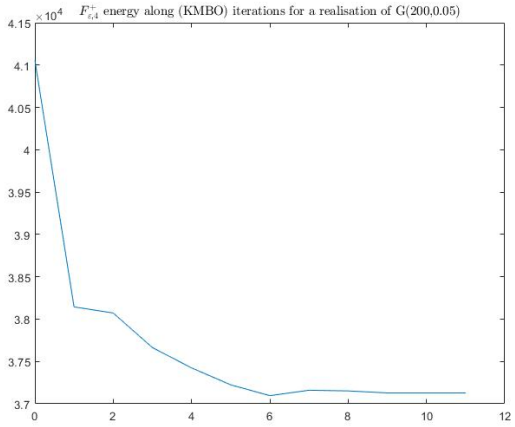
Set $N = j$.

if $S_{[K]}(X^j) = \frac{1}{2}\sum_{i,j \in V} \omega_{ij}$ **then**
  Stop

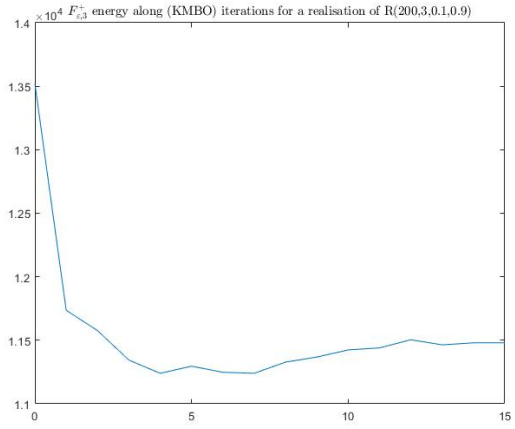
if $\frac{\|\mu^j - \mu^{j-1}\|_2^2}{\|\mu^j\|_2^2} < \eta$ **then**
  Stop

Find the largest cut size: Set $S^* := \max_{1 \le j \le N} S_{[K]}(X^j)$.

an overall tendency to decrease the value of $F^{+}_{\varepsilon,K}(U^j)$, however sometimes the functional value increases along iterations. Hence, we state the largest K-Cut obtained along iterations of both (KCut) and (KMBO+) as their outputs, instead of stating the K-Cut obtained on the final iteration of (KCut) and (KMBO+) as their outputs.

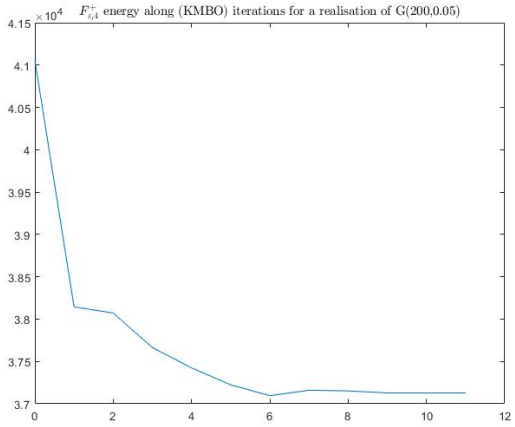


(a) The value $F_{\varepsilon,4}(\mu^j)$ as a function of the iteration number j in (KCut) on a realisation of $G(200, 0.05)$ (see Definition 3.4.3) using Δ_1^+ as the choice of signless Laplacian, with $k^* = 20$ and $\tau = 20$.

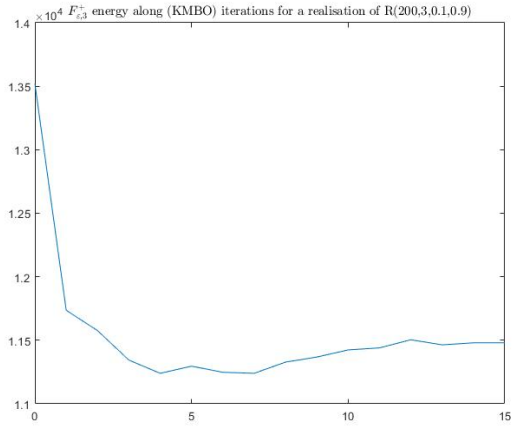


(b) The value $F_{\varepsilon,3}(\mu^j)$ as a function of the iteration number j in (KCut) on a realisation of $R(200, 3, 0.1, 0.9)$ (see Chapter 7.1.4) using Δ_1^+ as the choice of signless Laplacian, with $k^* = 20$ and $\tau = 20$.

Figure 9.: Two examples of the functional energy $F^+_{\varepsilon,K}$ along iterations of (KCut).



(a) The value $F_{\varepsilon,4}(\mu^j)$ as a function of the iteration number j in (KMBO+) on a realisation of $G(200, 0.05)$ (see Definition 3.4.3) using Δ_1^+ as the choice of signless Laplacian, with $k^* = 20$ and $\tau = 20$.



(b) The value $F_{\varepsilon,3}(\mu^j)$ as a function of the iteration number j in (KMBO+) on a realisation of $R(200, 3, 0.1, 0.9)$ (see Chapter 7.1.4) using Δ_1^+ as the choice of signless Laplacian, with $k^* = 20$ and $\tau = 20$.

Figure 10.: Two examples of the functional energy $F^+_{\varepsilon,K}$ along iterations of (KMBO+).

6

GRAPH ISOMORPHISM

6.1 NOTATION

We consider the sets of graphs $\bar{\mathcal{G}} := \{G \in \mathcal{G} : \forall (i,j) \in E,\, \omega_{ij} = 1\}$ and $\bar{\mathcal{G}}^n := \{G \in \bar{\mathcal{G}} : |V| = n\}$. For $G_1,\, G_2 \in \bar{\mathcal{G}}^n$, we denote their node and edge sets by (V_1, E_1) and (V_2, E_2), respectively. For edges $(i,j) \in E_1$ and $(k,l) \in E_2$, we denote their edge weights by $\omega_{ij}^{(1)}$ and $\omega_{kl}^{(2)}$, respectively. We denote the degrees of the nodes $i \in V_1$ and $j \in V_2$ by $d_i^{(1)} := \sum_{j \in V_1} \omega_{ij}^{(1)}$ and $d_j^{(2)} := \sum_{k \in V_2} \omega_{kl}^{(2)}$, respectively. For graphs $G_1,\, G_2 \in \bar{\mathcal{G}}^n$, we define $\mathcal{V}_1$ and $\mathcal{V}_2$ as the set of node functions $u : V_1 \to \mathbb{R}$ and $v : V_2 \to \mathbb{R}$, respectively. Given G_1 and G_2, we denote their respective graph Laplacians by Δ_1^r and Δ_2^r, and we define their diffusion operators $\mathcal{D}_{\tau,1} : \mathcal{V}_1 \to \mathcal{V}_1$ and $\mathcal{D}_{\tau,2} : \mathcal{V}_2 \to \mathcal{V}_2$ by, for $u \in \mathcal{V}_1$ and $v \in \mathcal{V}_2$, $\mathcal{D}_{\tau,1}(u) := e^{-\tau \Delta_1^r} u$ and $\mathcal{D}_{\tau,2}(v) := e^{-\tau \Delta_2^r} v$, respectively. We define the degree mappings $D_1 : V_1 \to \mathbb{N}$ and $D_2 : V_2 \to \mathbb{N}$, where for $i \in V_1$, $(D_1)_i = d_i^{(1)}$ and for $j \in V_2$, $(D_2)_j = d_j^{(2)}$. We denote the degree sequences of G_1 and G_2 by $\mathbf{d_1}$ and $\mathbf{d_2}$ respectively. For $d \in \mathbb{N}$, we define the subsets $W_{1,d} := \{i \in V_1 : d_i^1 = d\}$ and $W_{2,d} := \{j \in V_2 : d_j^2 = d\}$.

For $G_1,\, G_2 \in \mathcal{G}^n$, we next define the notion of node classifications between G_1 and G_2. Let $G_1,\, G_2 \in \mathcal{G}^n$, $u_0 \in \mathcal{V}_1$, and $v_0 \in \mathcal{V}_2$. Then, for $d \in \mathbb{N}$, and $e \in \mathbb{R}$, we define the sets $X_{d,e,\tau_0}^{(1)}$ and $X_{d,e,\tau_0}^{(2)}$ by

$$X_{d,e,\tau_0}^{(1)}(u_0) := D_1^{-1}(d) \cap (D_{\tau_0,1}(u_0))^{-1}(e),$$

and

$$X_{d,e,\tau_0}^{(2)}(v_0) := D_2^{-1}(d) \cap (\mathcal{D}_{\tau_0,2}(v_0))^{-1}(e),$$

respectively. Furthermore, we define $c^{(1)}_{d,e,\tau_0}$ and $c^{(2)}_{d,e,\tau_0}$ by

$$c^{(1)}_{d,e,\tau_0}(u_0) := card\big(X^1_{d,e,\tau_0}(u_0)\big) \text{ and } c^{(2)}_{d,e,\tau_0}(v_0) := card\big(X^2_{d,e,\tau_0}(v_0)\big)$$

respectively, where $card(X)$ denotes the cardinality of the set X.

6.2 DIFFUSION OPERATOR ISOMORPHISM

Definition 6.2.1. For G_1, $G_2 \in \bar{\mathcal{G}}^n$, let $\mathcal{O}_1$ and $\mathcal{O}_2$ be operators on $\mathcal{V}_1$ and $\mathcal{V}_2$ respectively. Then, $\mathcal{O}_1$ and $\mathcal{O}_2$ are isomorphic operators, denoted $\mathcal{O}_1 \simeq \mathcal{O}_2$, if there exists a bijection $\psi : V_1 \to V_2$ such that for all $u \in \mathcal{V}_1$, we have that $\mathcal{O}_1(u) = \mathcal{O}_2(u \circ \psi^{-1}) \circ \psi$.

A graph isomorphism from $G_1 = (V_1, E_1)$ to $G_2 = (V_2, E_2)$ is a bijection $\psi : V_1 \to V_2$, such that for all $i, j \in V_1$, $(i, j) \in E_1$ if and only if $(\psi(i), \psi(j)) \in E_2$. If an isomorphism exists, G_1 and G_2 are said to be isomorphic, denoted $G_1 \simeq G_2$. The graph isomorphism problem is the problem of determining whether any given G_1 and G_2 are isomorphic. The graph isomorphism problem is in the complexity class NP [76]; it is conjectured that if the complexity class NPI is non-empty, then the graph isomorphism problem could belong to the NPI class [58].

We denote the set of bijections from V_1 to V_2 by $V_{1,2} := \{\psi : V_1 \to V_2 : \psi \text{ is a bijection}\}$, and the set of graph isomorphisms of G_1 and G_2 by $Iso(G_1, G_2) := \{\psi \in V_{1,2} : \forall i, j \in V_1, (i, j) \in E_1 \Leftrightarrow (\psi(i), \psi(j)) \in E_2\}$.

Proposition 6.2.2. *Let G_1, $G_2 \in \bar{\mathcal{G}}^n$. Then, $G_1 \simeq G_2$ if and only if, for all $\tau \geq 0$, $\mathcal{D}_{\tau,1} \simeq \mathcal{D}_{\tau,2}$. Furthermore, if ψ is an isomorphism between G_1 and G_2, then ψ is an isomorphism between $\mathcal{D}_{\tau,1}$ and $\mathcal{D}_{\tau,2}$.*

Proof. We begin this proof by first showing that $G_1 \simeq G_2$ if and only if $\Delta^r_1 \simeq \Delta^r_2$. We first show that, for $r \in [0, 1]$, $G_1 \simeq G_2 \Rightarrow \Delta^r_1 \simeq \Delta^r_2$.

Assume that $G_1 \simeq G_2$, and $\psi \in Iso(G_1, G_2)$. For $\Delta^r_1 \simeq \Delta^r_2$, we require that for all $i \in V_1$,

$$(\Delta^r_1 u)_i = (\Delta^r_2(u \circ \psi^{-1}) \circ \psi)_i = (\Delta^r_2(u \circ \psi^{-1}))_{\psi(i)}.$$

Let $v = u \circ \psi^{-1}$, then

$$(\Delta_2^r v)_{\psi(i)} = \frac{1}{(d^{(2)}_{\psi(i)})^r} \sum_{k \in V_2} \omega^{(2)}_{\psi(i)j}(v_{\psi(i)} - v_k).$$

Since $G_1 \simeq G_2$, for all $i, j \in V_1$, $(i,j) \in E_1$, if and only if $(\psi(i), \psi(j)) \in E_2$, and for all $i \in V_1$, $d^{(1)}_i = d^{(2)}_{\psi(i)}$.

Since ψ is a bijection, $\forall k \in V_2$, $\exists! j \in V_1$ such that $\psi(j) = k$. Then,

$$\begin{aligned}
(\Delta_2^r v)_{\psi(i)} &= \frac{1}{(d^{(2)}_{\psi(i)})^r} \sum_{j \in V_1} \omega^{(2)}_{\psi(i)\psi(j)}(v_{\psi(i)} - v_{\psi(j)}) \\
&= \frac{1}{(d^{(1)}_i)^r} \sum_{j \in V_1} \omega^{(1)}_{ij}(v_{\psi(i)} - v_{\psi(j)}) \\
&= \frac{1}{(d^{(1)}_i)^r} \sum_{j \in V_1} \omega^{(1)}_{ij}\Big((u \circ \psi^{-1})_{\psi(i)} - (u \circ \psi^{-1})_{\psi(j)}\Big) \\
&= \frac{1}{(d^{(1)}_i)^r} \sum_{j \in V_1} \omega^{(1)}_{ij}(u_i - u_j) = (\Delta_1^r u)_i
\end{aligned}$$

Next we show that $\Delta_1^r \simeq \Delta_2^r \Rightarrow G_1 \simeq G_2$. For all $u \in \mathcal{V}_1$, there exists $\psi \in V_{1,2}$ such that for all $i \in V_1$, $(\Delta_1^r u)_i = (\Delta_2^r v)_{\psi(i)}$. Setting $v = u \circ \psi^{-1}$, we have that

$$\frac{1}{(d^{(1)}_i)^r} \sum_{j \in V_1} \omega^{(1)}_{ij}(u_i - u_j) = \frac{1}{(d^{(2)}_{\psi(i)})^r} \sum_{l \in V_2} \omega^{(2)}_{\psi(i)l}(v_{\psi(i)} - v_l). \tag{49}$$

Let $k \in V_1$, and let $u = \chi_{\{k\}}$, then, (49) implies that

$$\begin{aligned}
&\frac{1}{(d^{(1)}_i)^r} \sum_{j \in V_1} \omega^{(1)}_{ij}\Big((\chi_{\{k\}})_i - (\chi_{\{k\}})_j\Big) \\
&= \frac{1}{(d^{(2)}_{\psi(i)})^r} \sum_{l \in V_2} \omega^{(2)}_{\psi(i)l}\Big((\chi_{\{k\}} \circ \psi^{-1})_{\psi(i)} - (\chi_{\{k\}} \circ \psi^{-1})_l\Big).
\end{aligned} \tag{50}$$

If $i = k$, then (50) implies

$$\frac{1}{(d^{(1)}_1)^r} \sum_{j \in V_1} \omega^{(1)}_{ij} = \frac{1}{(d^{(2)}_{\psi(i)})^r} \sum_{l \in V_2} \omega^{(2)}_{\psi(i)l},$$

which is equivalent to $(d^{(1)}_i)^{(1-r)} = (d^{(2)}_{\psi*(i)})^{(1-r)}$. For $r \in [0,1)$, this holds if and only if for all $i \in V_1$, $d^{(1)}_i = d^{(2)}_{\psi(i)}$. If $r = 1$, this condition trivially holds.

If $i \neq k$, then (50) implies

$$(d^{(2)}_{\psi(i)})^r \omega^{(1)}_{ik} = \omega^{(2)}_{\psi(i)\psi(k)}(d^{(1)}_i)^r.$$

Since G_1 and G_2 are unweighted, this holds if, for all $i, j \in V_1$, $(i,j) \in E_1$, if and only if $(\psi(i), \psi(j)) \in E_2$. Furthermore, for $r \in (0,1]$, we require that for all $i \in V_1$, $d_i^{(1)} = d_{\psi(i)}^{(2)}$. Therefore, for all $k \in V_1$, and for all $r \in [0,1]$, (50) holds if and only if ψ is the isomorphism of G_1 and G_2.

For all $k \in V_1$, $u = \chi_{\{k\}}$ forms a basis of all node functions in $\mathcal{V}_1$, and the operators Δ_1^r and Δ_2^r are linear operators. It follows that since (50) holds for all $k \in V_1$, and $u = \chi_{\{k\}}$, if and only if $G_1 \simeq G_2$, then (50) holds for all $u \in \mathcal{V}_1$, if and only if $G_1 \simeq G_2$.

Therefore, $\Delta_1^r \simeq \Delta_2^r \Rightarrow G_1 \simeq G_2$, and $G_1 \simeq G_2 \Leftrightarrow \Delta_1^r \simeq \Delta_2^r$.

Next, we prove that $\Delta_1^r \simeq \Delta_2^r$ if and only if for all $\tau \geq 0$, $\mathcal{D}_{\tau,1} \simeq \mathcal{D}_{\tau,2}$, thus proving that $G_1 \simeq G_2$ if and only if for all $\tau \geq 0$, $\mathcal{D}_{\tau,1} \simeq \mathcal{D}_{\tau,2}$. First we prove that $\Delta_1^r \simeq \Delta_2^r \implies \mathcal{D}_{\tau,1} \simeq \mathcal{D}_{\tau,2}$.

For all $u \in \mathcal{V}_1$, if $\Delta_1^r \simeq \Delta_2^r$, with isomorphism ψ, then

$$\mathcal{D}_{\tau,1}(u) = e^{-\tau\Delta_1^r} u = \left(e^{-\tau\Delta_2^r}(u \circ \psi^{-1})\right) \circ \psi = \mathcal{D}_{\tau,2}(u \circ \psi^{-1}) \circ \psi.$$

Therefore, ψ is an isomorphism of $\mathcal{D}_{\tau,1}$ and $\mathcal{D}_{\tau,2}$.

Next we show that $\mathcal{D}_{\tau,1} \simeq \mathcal{D}_{\tau,2} \Rightarrow \Delta_1 \simeq \Delta_2$. Since $\mathcal{D}_{\tau,1} \simeq \mathcal{D}_{\tau,2}$, the derivatives of $\mathcal{D}_{\tau,1}$ and $\mathcal{D}_{\tau,2}$ with respect to τ must also be isomorphic. Therefore,

$$\begin{aligned} -\Delta_1^r(\mathcal{D}_{\tau,1}(u)) &= -\left(\Delta_2^r\big(\mathcal{D}_{\tau,2}(u \circ \psi^{-1})\big)\right) \circ \psi \\ &= -\Delta_2^r(\mathcal{D}_{\tau,1}(u) \circ \psi^{-1}) \circ \psi \end{aligned}$$

must hold for all $u \in \mathcal{V}_1$ and for all $\tau \geq 0$. Setting $v = \mathcal{D}_{\tau,1}(u)$, we have that

$$\Delta_1^r(v) = \Delta_2^r(v \circ \psi^{*-1}) \circ \psi^*. \tag{51}$$

Since (51) holds for all $u \in \mathcal{V}_1$, and the range of $\mathcal{D}_{\tau,1}$ is $\mathcal{V}_1$, it follows that the derivatives of $\mathcal{D}_{\tau,1}$ and $\mathcal{D}_{\tau,2}$ with respect to τ are isomorphic if and only $\Delta_1^r \simeq \Delta_2^r$. Therefore, since $\Delta_1^r \simeq \Delta_2^r \Leftrightarrow \mathcal{D}_{\tau,1} \simeq \mathcal{D}_{\tau,2}$, it holds that $G_1 \simeq G_2$ if and only if, for all $\tau \geq 0$, $\mathcal{D}_{\tau,1} \simeq \mathcal{D}_{\tau,2}$.

□

6.3 ADDITIONAL RESULTS

Corollary 6.3.1. *Let G_1, $G_2 \in \bar{\mathcal{G}}^n$ such that $G_1 \simeq G_2$. Let $\psi \in Iso(G_1, G_2)$. For all $u_0 \in \mathcal{V}_1$ and $v_0 \in \mathcal{V}_2$, such that $u_0 \circ \psi^{-1} = v_0$, then, for all $\tau \geq 0$, $d \in \mathbb{N}$ and $e \in \mathbb{R}$, $c^1_{d,e,\tau_0}(u_0) = c^2_{d,e,\tau_0}(v_0)$.*

Proof. By Proposition 6.2.2, since $G_1 \simeq G_2$, and $\psi \in Iso(G_1, G_2)$, for all $\tau \geq 0$, $\mathcal{D}_{\tau,1} \simeq \mathcal{D}_{\tau,2}$ to the same isomorphism. Since $u_0 \circ \psi^{-1} = v_0$, for all $\tau \geq 0$, $\mathcal{D}_{\tau,1}(u_0) = \mathcal{D}_{\tau,2}(v_0) \circ \psi$. For all $\tau \geq 0$, and for all $i \in V_1$, $(\mathcal{D}_{\tau,1}(u_0))_i = (\mathcal{D}_{\tau,2}(v_0))_{\psi(i)}$.

Since $\psi \in Iso(G_1, G_2)$, for all $i \in V_1$, $d^{(1)}_i = d^{(2)}_{\psi(i)}$. Furthermore, for all $d \in \mathbb{N}$, $card(W_{1,d}) = card(W_{2,d})$. Therefore, since for all $\tau_0 \geq 0$, and for all $i \in V_1$, $(\mathcal{D}_{\tau,1}(u_0))_i = (\mathcal{D}_{\tau,2}(v_0))_{\psi(i)}$, it follows that, for all $d \in \mathbb{N}$, for all $\tau_0 \geq 0$, and for all $k \in W_{1,d}$, $(\mathcal{D}_{\tau,1}(u_0))_k = (\mathcal{D}_{\tau,2}(v_0))_{\psi(k)}$. Therefore, for all $\tau \geq 0$, for all $d \in \mathbb{N}$, and for all $e \in \mathbb{R}$, we have that

$$\begin{aligned} c^{(1)}_{d,e,\tau}(u_0) = card(D_1^{-1}(d) \cup \mathcal{D}_{\tau,1}(u_0)^{-1}(e)) &= card(D_2^{-1}(d) \cup \mathcal{D}_{\tau,2}(v_0)^{-1}(e)) \\ &= c^{(2)}_{d,e,\tau}(v_0). \end{aligned}$$

□

Corollary 6.3.2. *Let G_1, $G_2 \in \bar{\mathcal{G}}^n$. For any $d \in \mathbb{N}$, let $u_0 = \chi_{W_{1,d}}$ and $v_0 = \chi_{W_{2,d}}$. If there exists a $d \in \mathbb{N}$, an $e \in \mathbb{R}$, and a $\tau_0 \geq 0$ such that $c^1_{d,e,\tau_0}(u_0) \neq c^2_{d,e,\tau_0}(v_0)$, G_1 and G_2 are not isomorphic.*

Proof. We prove by contradiction. Assume that $G_1 \simeq G_2$. For any $\psi \in Iso(G_1, G_2)$ and for any $d \in \mathbb{N}$, if $i \in W_{1,d}$, and $\psi(i) = j$, then $j \in W_{2,d}$. Similarly, if $j \in W_{2,d}$, and $\psi^{-1}(j) = i$, then $i \in W_{1,d}$. Therefore, $u_0 \circ \psi^{-1} = v_0$. However, since there exists a $d \in \mathbb{N}$, an $e \in \mathbb{R}$, and a $\tau_0 \geq 0$ such that $c^1_{d,e,\tau_0}(u_0) \neq c^2_{d,e,\tau_0}(v_0)$, by Corollary 6.3.1, G_1 cannot be isomorphic to G_2.

□

Proposition 6.3.3. *Let G_1, $G_2 \in \bar{\mathcal{G}}^n$, such that $G_1 \simeq G_2$. Let $\psi \in Iso(G_1, G_2)$, and let $u_0 \in \mathcal{V}_1$ and $v_0 \in \mathcal{V}_2$, such that $u_0 \circ \psi^{-1} = v_0$. For all $d \in \mathbb{N}$ and for*

all $e \in \mathbb{R}$ *such that for some* $\tau_* \geq 0$, $c^1_{d,e,\tau_*}(u_0) = c^2_{d,e,\tau_*}(v_0) = 1$, *we have that* $\psi\Big(X^1_{d,e,\tau_*}(u_0)\Big) = X^2_{d,e,\tau_*}(v_0)$.

Proof. Since $G_1 \simeq G_2$, by Proposition 6.2.2 with $\psi \in Iso(G_1, G_2)$, for all $\tau \geq 0$, $\mathcal{D}_{\tau,1} \simeq \mathcal{D}_{\tau,2}$, with isomorphism ψ. For all $d \in \mathbb{N}$, if $i \in W_{1,d}$, then $\psi(i) \in W_{2,d}$, and if $j \in W_{2,d}$, then $\psi^{-1}(j) \in W_{1,d}$.

Since for a $d \in \mathbb{N}$, $e \in \mathbb{R}$, and $\tau_* \geq 0$, we have $c^{(1)}_{d,e,\tau_*}(u_0) = c^{(2)}_{d,e,\tau_*}(v_0) = 1$, $\exists! i \in W_{1,d}$ and $\exists! k \in W_{2,d}$, such that $X^{(1)}_{d,e,\tau_*}(u_0) = \{i\}$ and $X^{(2)}_{d,e,\tau_*}(v_0) = \{k\}$. Therefore, $\mathcal{D}_{\tau_*,1}(u_0)_i = \mathcal{D}_{\tau_*,2}(v_0)_k$. Furthermore, for all $j \in W_{1,d}\backslash\{i\}$, $\mathcal{D}_{\tau_*,1}(u_0)_j \neq \mathcal{D}_{\tau_*,2}(v_0)_k$, and for all $l \in W_{2,d}\backslash\{k\}$, $\mathcal{D}_{\tau_*,1}(u_0)_i \neq \mathcal{D}_{\tau_*,2}(v_0)_l$. Since $\mathcal{D}_{\tau,1} \simeq \mathcal{D}_{\tau,2}$, it holds that for all $l \in W_{2,d}\backslash\{k\}$, $\psi(i) \neq l$, and for all $j \in W_{1,d}\backslash\{i\}$, $\psi^{-1}(k) \neq j$. Therefore, $\psi(i) = k$ must hold, and $\psi\Big(X^{(1)}_{d,e,\tau_*}(u_0)\Big) = X^{(2)}_{d,e,\tau_*}(v_0)$.

□

Proposition 6.3.4. *Let* $G_1, G_2 \in \bar{\mathcal{G}}^n$ *such that* $G_1 \simeq G_2$, *and* $|Iso(G_1, G_2)| > 1$. *Let* $u_0 \in \mathcal{V}_1$ *and* $v_0 \in \mathcal{V}_2$ *such that if* $\psi \in Iso(G_1, G_2)$, *then* $u_0 \circ \psi^{-1} = v_0$. *For all* $\tau \geq 0$, *there exists a* $d \in \mathbb{N}$ *and there exists an* $e \in \mathbb{R}$ *such that* $c^{(1)}_{d,e,\tau} = c^{(2)}_{d,e,\tau} > 1$.

Proof. Let $\psi_1, \psi_2 \in Iso(G_1, G_2)$ such that $\psi_1 \neq \psi_2$. There exists $i_1, i_2 \in V_1$, where $i_1 \neq i_2$, and there exists $j_1, j_2 \in V_2$, where $j_1 \neq j_2$, such that $\psi_1(i_1) = j_1$, $\psi_2(i_1) = j_2$, and $\psi_2(i_2) = j_1$. By Proposition 6.2.2, for all $\tau \geq 0$, ψ_1 and ψ_2 will also be isomorphisms between $\mathcal{D}_{\tau,1}$ and $\mathcal{D}_{\tau,2}$. Therefore, we have that $\mathcal{D}_{\tau,1}(u_0) = \mathcal{D}_{\tau,2}(v_0) \circ \psi_1$ and $\mathcal{D}_{\tau,1}(u_0) = \mathcal{D}_{\tau,2}(v_0) \circ \psi_2$ both hold, meaning $\mathcal{D}_{\tau,1}(u_0)_{i_1} = \mathcal{D}_{\tau,2}(v_0)_{\psi_1(i_1)}$, $\mathcal{D}_{\tau_0,1}(u_0)_{i_1} = \mathcal{D}_{\tau_0,2}(v_0)_{\psi_2(i_1)}$, and $\mathcal{D}_{\tau_0,1}(u_0)_{i_2} = \mathcal{D}_{\tau_0,2}(v_0)_{\psi_2(i_2)}$ all hold. As $\psi_1, \psi_2 \in Iso(G_1, G_2)$, $(d_1)_{i_1} = (d_2)_{j_1}$, $(d_1)_{i_1} = (d_2)_{j_2}$, and $(d_1)_{i_2} = (d_2)_{j_1}$ all hold. Therefore, we have that $\exists d \in \mathbb{N}$ and $\exists e \in \mathbb{R}$ such that $i_1, i_2 \in X^{(1)}_{d,e,\tau_0}(u_0)$ and $j_1, j_2 \in X^{(2)}_{d,e,\tau_0}(v_0)$.

By Proposition 6.3.1, since $G_1 \simeq G_2$, and for all $\psi \in Iso(G_1, G_2)$, $u_0 \circ \psi^{-1} = v_0$, we have that for all $d \in \mathbb{N}$, for all $e \in \mathbb{R}$, and for all $\tau \geq 0$, $c^1_{d,e,\tau}(u_0) = c^2_{d,e,\tau}(v_0)$. Since $i_1, i_2 \in X^{(1)}_{d,e,\tau_0}(u_0)$ and $j_1, j_2 \in X^{(2)}_{d,e,\tau_0}(v_0)$, for all $\tau \geq 0$, there exists a $d \in \mathbb{N}$, $e \in \mathbb{R}$ such that $c^{(1)}_{d,e,\tau}(u_0) = c^{(2)}_{d,e,\tau}(v_0) > 1$.

□

7

COMPUTATIONAL RESULTS - MAX-CUT

7.1 MAX-CUT RESULTS

7.1.1 *Method*

In Chapter 7.1 we compare the results of (MBO+) with the results obtained by (GW). In Chapters 7.1.2–7.1.6 we display the results of (MBO+) using both the spectral decomposition method and the explicit Euler method, fixing the parameter τ for both methods. We run all our tests in Chapter 7 on a Windows 7 PC with 16GB RAM and an Intel(R) Core(TM) i5-4590 CPU with clock speed 3.30GHz. For both (MBO+) and (GW) we use MATLAB, which is convenient to use when dealing with large sparse matrices.

For all of our tests using the spectral decomposition method we choose the largest out of $k^* = \lfloor \frac{n}{100} \rfloor$ or $k^* = 20$. In practice, the computation time of (MBO+) using this choice of k^* is reduced in comparison to choosing $k^* = n$, without sacrificing much accuracy in the cut approximations. We further analyse this choice in Chapter 7.2.1. For all of our tests using the Euler method we set $M = 100$, in order to keep dt small so as to ensure stability on our explicit scheme. We compute the (MBO+) evolutions for 50 initial conditions chosen at random from $\mathcal{V}^b$. In the tables which we refer to in this section, we state the greatest (Best), average (Avg), and smallest (Least) sizes of cuts obtained by these 50 runs of (MBO+). We run (MBO+) using Δ_0^+, Δ_1^+ and Δ_s^+,

fixing the initial conditions for each operator, using both the spectral method and the Euler method for each operator, and compare the results.

We compare the results of (MBO+) with those of (GW). To compute the relaxation step of (GW) we use SDPT3 MATLAB software [84] as it exploits the sparse structure of the matrices we work on. According to [68] it is best suited for problems with sparse matrices. The stopping tolerance is set as $\nu < 10^{-6}$. The recommended tolerance for the SDPT3 software is set as 10^{-8}. However, in our experiments increasing this tolerance to 10^{-6} reduced the computation time of (GW), without any change in output cut sizes. After the relaxation step, we perform the hyperplane step 50 times, randomly choosing a vector r each time. Each choice of r leads to a resulting cut; in the tables referred to in this sub-chapter, we list the highest (Best), average (Avg), and lowest (Least) sizes of these cuts. In each of these categories in our tables we highlight the method that obtained the best result, (MBO+) using Δ_0^+, (MBO+) using Δ_1^+, (MBO+) using Δ_s^+, or (GW). We do the same for the run times (Time) of each method.

For both (MBO+) and (GW) only the adjacency matrix and the parameter choice η is initially provided, therefore the reported run times cover all calculations from that starting point. For each graph we remove the isolated nodes by removing all rows and columns of the graph's adjacency matrix which have all zero entries. (This does not affect the size of any cut of the graph.) For the spectral decompostion variant of (MBO+) using Δ_1^+ and Δ_s^+ this includes removing all isolated nodes, computing the matrices L_1 and L_s, finding their k^* eigenpairs corresponding to the leading eigenvalues in order to use Proposition 3.5.1, to compute the eigenpairs corresponding to the trailing eigenvalues of L_1^+ and L_s^+ respectively, generating initial conditions, running the signless diffusion and thresholding steps, and computing the size of the cut from each MBO iteration. For Δ_s^+ the computation time includes calculating L_1 in order to compute the size of the output cuts using (21). The computation time for (MBO+) using Δ_0^+ includes removing all isolated nodes, computing the matrix L_0^+, finding all its eigenpairs, choos-

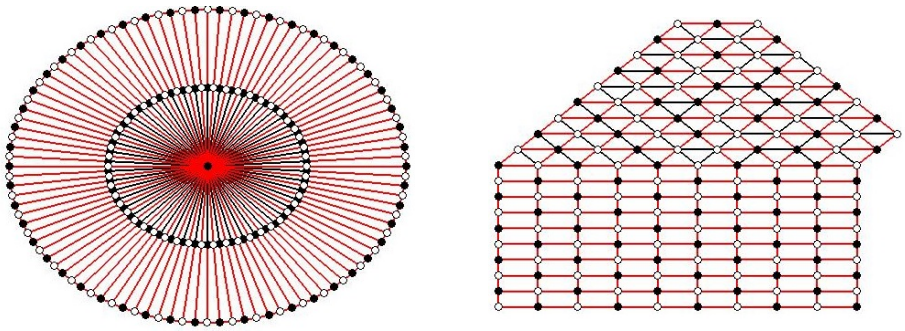
ing the largest eigenvalue for the time step τ, and using the k^* eigenpairs corresponding to the smallest eigenvalues for the remaining steps.

For the explicit Euler method variant of (MBO+) the computation time includes removing all isolated nodes, computing $L^+ \in \{L_0^+, L_1^+, L_s^+\}$, generating initial conditions, running the signless diffusion and thresholding steps, and computing the size of the cut induced by each MBO iteration. For $L = L_s$ we also compute L_1 to obtain the size of the output cut using (21).

For every graph there exists a τ_{max} such that for all $\tau \geq \tau_{max}$ the solution to (28) computed using (MBO+) converges to $u(\tau) = 0$ to machine precision. In practice τ_{max} is dependent on the operator Δ^+. In our experiments we see that choosing a τ which is in between the pinning condition in Theorem 4.2.3 and τ_{max} is difficult due to the difference between them being small when Δ_0^+ is our operator for (MBO+). In Chapter 7.1.3 and Chapter 7.1.5 we run our experiments on graphs with a scale free structure (see Section 7.1.3). If (MBO+) returns a cut which has pinned due to Theorem 4.2.3 or is zero due to the solution of (28) converging to zero to machine precision then we refer to the cut as a trivial cut. In Theorem 7.2.3 we show that it is possible to obtain non-trivial cut sizes using (MBO+) with Δ_0^+ by solving (28) using an implicit Euler scheme.

Figure 11 shows two examples of approximate maximum cuts obtained with the (MBO+) algorithm. The black nodes are in V_1 and the white nodes are in V_{-1}. An edge is coloured red, if it connects two nodes of different colour, i.e. if it contributes to the size of the cut. If it does not, it is black.

Figure 11a shows an unweighted web graph which has 201 nodes and 400 edges. We set $\tau = 20$ in (MBO+) using Δ_1^+ and the Euler method to solve (28). The resulting approximation of the maximum cut value is 350. The run time is 0.09 seconds. Figure 11b shows an unweighted triangle-square graph which has 162 nodes and 355 edges. We set $\tau = 20$ and $k^* = 20$ in (MBO+) using Δ_1^+ and the spectral method to solve (28). The approximation of the maximum cut value is 295 and the run time is 0.14 seconds.



(a) Web graph, maximum cut approximation

(b) Square-triangle mesh maximum cut approximation

Figure 11.: Visualisation of maximum cut approximations (best viewed in colour)

7.1.2 *Random graphs*

In Figures 12, 13, and 14 we list results obtained for Erdös-Rényi graphs.

For each of $G(1000, 0.01)$ (Figure 12), $G(2500, 0.4)$ (Figure 13), and $G(5000, 0.001)$ (Figure 14) we create 100 realisations. We then run (MBO+) with both the spectral method and the Euler method, and we run (GW). For both of the (MBO+) methods we choose either Δ_0^+, Δ_1^+, or Δ_s^+, setting $\tau = 20$ for all tests. The bar chart represents the mean of the best, average, and least cuts over all 100 realisations of the chosen random graph. The error bars are the corrected sample standard deviation[1] of the results obtained over all 100 realisations. Figure 12 shows that (MBO+) using either the spectral method or Euler method for Δ_1^+ and Δ_s^+ produces better mean best, mean average, and mean least cuts than (GW) on this set of graphs. Figure 13 shows that (MBO+) using the spectral method and either Δ_1^+ or Δ_s^+ produces better mean cut approximations than (GW) on this set of graphs. Figure 14 shows the same conclusions as Figure 12 for this set of graphs. Table 1 shows that (MBO+) using the spectral method produces the fastest run times on all three types

1 The corrected sample standard deviation is computed using MATLAB's `std` code in all experiments in this thesis.

of Erdös-Rényi graphs that we test on. We note that (GW) has a superior run time over (MBO+) using the Euler method on the realisations of $G(2500, 0.4)$.

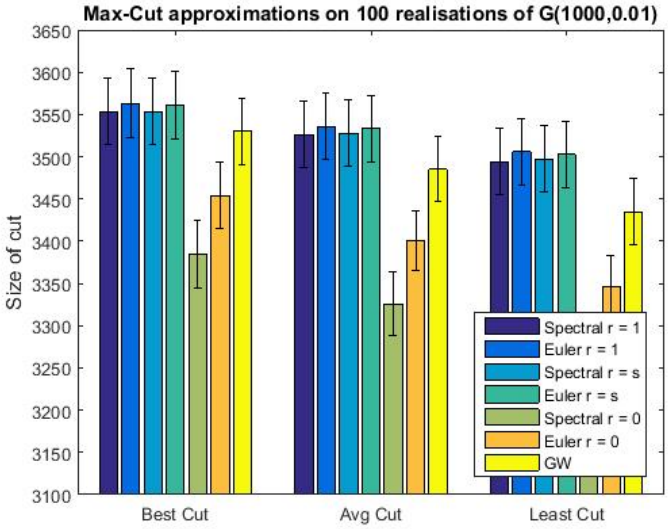


Figure 12.: Bar chart of Max-Cut approximations on 100 realisations of $G(1000, 0.01)$.

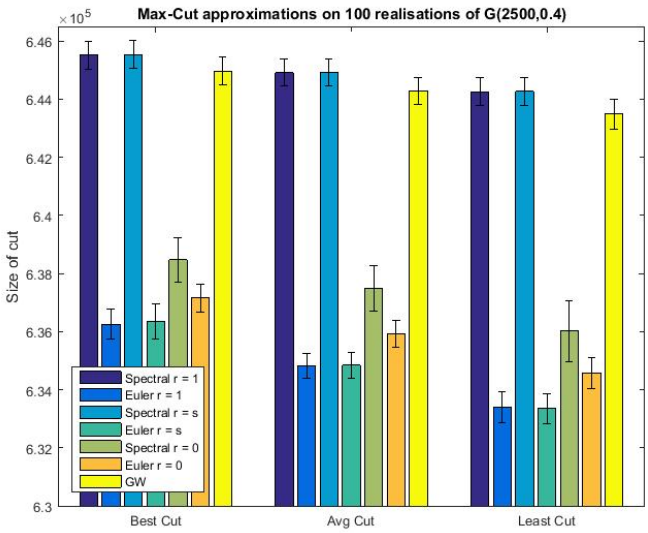


Figure 13.: Bar chart of Max-Cut approximations on 100 realisations of $G(2500, 0.4)$.

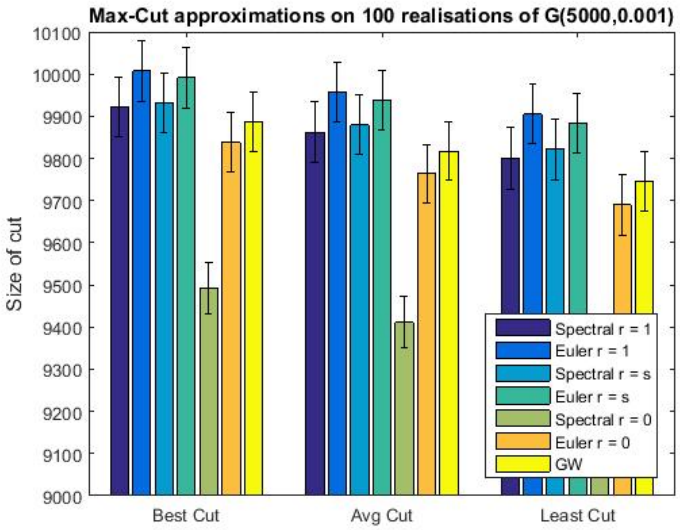


Figure 14.: Bar chart of Max-Cut approximations on 100 realisations of $G(5000, 0.001)$.

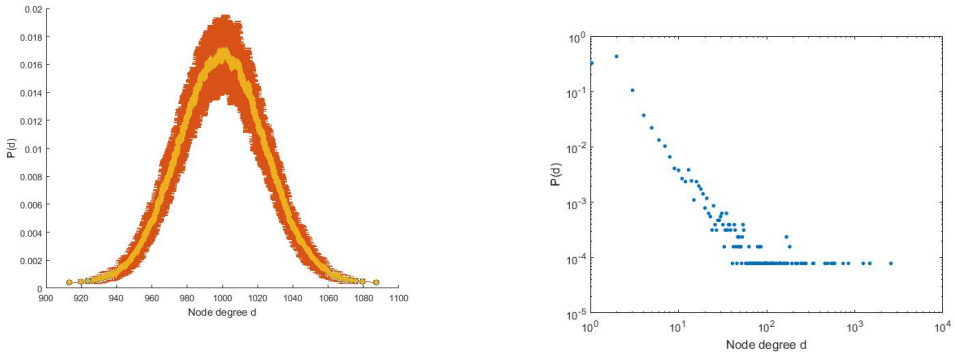
Graph	Δ_1^+ (S)	Δ_1^+ (E)	Δ_s^+ (S)	Δ_s^+ (E)	Δ_0^+ (S)	Δ_0^+ (E)	GW
$G(1000, 0.01)$	**0.20**	1.58	0.34	1.52	0.56	1.06	5.25
$G(2500, 0.4)$	8.04	172.91	13.33	181.40	**6.40**	172.73	55.36
$G(5000, 0.001)$	**4.38**	16.96	6.37	14.95	24.99	6.97	257.09

Table 1.: Average (MBO+) and (GW) run-times for each realisation of $G(n, p)$, time in seconds.

7.1.3 *Scale-free graphs*

The degree distribution $P : \mathbb{N} \to \mathbb{R}$ of an unweighted graph G is given by $P(j) := \frac{|\{i \in V : d_i = j\}|}{n}$. Random graphs such as the ones discussed in Chapter 7.1.2 have a degree distribution which resembles a normal distribution [10]. The graph $G \in \mathcal{G}$ is a scale-free graph if its degree distribution roughly follows a power law, i.e $P(j) \approx j^{-\gamma}$, where often in practice, $\gamma \in (2, 3)$ [6]. Scale-free graphs have become of interest as graphs such as internet networks, collaboration networks, and social networks are conjectured to more closely resemble scale-free graphs instead of random graphs [7].

In Tables 3-6 we list results for some scale free graphs. We test the algorithms on 8 autonomous systems internet graphs, ASi, $i \in \{1, \ldots, 8\}$. These



(a) Degree distribution of a realisation of $G(2500, 0.4)$.

(b) Degree distribution of the AS1 Graph.

Figure 15.: Average degree distribution of 100 realisations of a random graph and the degree distribution of a scale free graph.

graphs represent smaller imitations of an internet network, which were acquired from the website [91]. We also test on the graph Gnutella09 which is a model of a peer to peer file sharing network, and the graph WikiVote, which is a network representing a Wikipedia administrator election, both obtained from [95]. All of the scale free graphs in this section are unweighted and undirected graphs.

Table 2 displays some properties of the random graphs in Chapter 7.1.2 and the scale-free graphs we test on. Figure 15 displays the average degree distribution of 100 realisations of $G(2500, 0.4)$, in Figure 15a, and the degree distribution of the AS1 Graph, in Figure 15b. In Figure 15a the yellow points indicate the degree distribution, and the orange lines indicate the corrected sample standard deviation of the average degree distribution. In Figure 15b the blue dots indicate the degree distribution. As we see, the average degree distribution of the realisations of $G(2500, 0.4)$ is similar to a normal distribution, and the degree distribution of the AS1 graph resembles a power law, as expected.

Tables 3, 4, and 6 show that, choosing either Δ_1^+ or Δ_s^+ as the choice of signless Laplacian, (MBO+) using the Euler method or the spectral method outperforms (GW) with respect to the average and least obtained cut sizes and the run time, but (GW) obtains the best results when considering the

Graph	$\|V\|$	$\|E\|$	d_-	d_+
$G(1000, 0.01)(1)$	1000	4919	1	21
$G(1000, 0.01)(2)$	1000	4939	2	21
$G(2500, 0.4)(1)$	2500	1248937	910	1079
$G(2500, 0.4)(2)$	2500	1251182	904	1081
$G(5000, 0.001)(1)$	4962	12646	1	16
$G(5000, 0.001)(2)$	4969	12642	1	16

Graph	$\|V\|$	$\|E\|$	d_-	d_+
AS1	12694	26559	1	2566
AS2	7690	15413	1	1713
AS3	8689	17709	1	1911
AS4	8904	17653	1	1921
GNutella09	8114	26013	1	102
Wiki-Vote	7115	100762	1	1065

Table 2.: Properties of $G(n, p)$ graph realisations vs scale free graphs.

greatest obtained cuts. For any choice of Δ_1^+ and Δ_s^+ and for any choice of signless diffusion solver the greatest cuts obtained by (MBO+) are all at least 98.1% of the greatest cut size obtained by (GW). The difference in run times is notable though. The time taken by (MBO+) stays below 30 seconds for all graphs in Table 6, irrespective of choice of Laplacian and signless diffusion solver. However, the (GW) algorithm's run times range between 9 and 44 minutes. These results suggest that (MBO+) using Δ_1^+ or Δ_s^+, and using either signless diffusion solver offers a significant decrease in run time at the cost of about 1-2% accuracy in the resulting cut size, in comparison with (GW), when applied to the graphs in Tables 3-6.

7.1.4 *Random modular graphs*

Modular graphs have a community structure. Nodes in a community have many connections with other members of the same community and notice-

Graph	Δ_1^+ (S) Best	Δ_1^+ (S) Avg	Δ_1^+ (S) Least	Δ_1^+ (S) Time
AS1	22744	22542.20	22183	**15.85**
AS2	13249	13153.72	13054	**3.55**
AS3	15118	15027.22	14907	4.73
AS4	15194	15143.44	15042	**5.67**
AS5	14080	13988.90	13928	**4.82**
AS6	18053	17964.74	17876	10.06
AS7	22741	22535.00	22150	17.82
AS8	22990	22720.36	22334	17.22
GNutella09	20280	20143.74	19983	8.16
WikiVote	72981	72856.40	72744	2.46

Graph	Δ_1^+ (E) Best	Δ_1^+ (E) Avg	Δ_1^+ (E) Least	Δ_1^+ (E) Time
AS1	22798	**22670.76**	22268	23.62
AS2	13281	**13199.72**	**13120**	8.76
AS3	15175	**15095.46**	**15007**	9.95
AS4	15270	**15202.70**	**15117**	10.88
AS5	14120	**14020.62**	**13944**	9.50
AS6	18134	**18034.10**	**17933**	16.50
AS7	22826	**22696.42**	**22525**	25.78
AS8	23070	**22951.54**	**22550**	25.38
GNutella09	20437	**20361.92**	**20295**	17.14
WikiVote	73159	**73126.34**	**73086**	9.06

Table 3.: (MBO+) cut approximations using Δ_1^+ on graphs with a scale free structure, time in seconds.

ably fewer connections with members of other communities. In Figure 16 we show what our Max-Cut approximation looks like on a random modular graph. We generate realisations of random unweighted modular graphs $R(n, c, p, r)$ using the code provided at [94]. The variables for the graph are the number of nodes n, the number $c \in \mathbb{N}$ of communities that the graph contains, a probability p such that the graph will have an expected number of $\frac{n^2}{2p}$ edges, and a ratio $r \in [0, 1]$, with $r|E|$ being the expected number of edges connecting nodes in the same community and $(1 - r)|E|$ being the expected number of edges connecting nodes in different communities.

Graph	Δ_s^+ (S) Best	Δ_s^+ (S) Avg	Δ_s^+ (S) Least	Δ_s^+ (S) Time
AS1	22809	22620.8	**22325**	17.83
AS2	13271	13178.86	13103	4.12
AS3	15166	15082.1	14992	**4.66**
AS4	15237	15166.24	15077	5.78
AS5	14075	14011.96	13911	5.47
AS6	18088	17968.04	17859	**9.14**
AS7	22822	22629.66	22218	**15.73**
AS8	23061	22884.8	22547	**15.46**
GNutella09	20282	20186.32	20101	**6.82**
WikiVote	73169	73003.44	72917	**2.25**

Graph	Δ_s^+ (E) Best	Δ_s^+ (E) Avg	Δ_s^+ (E) Least	Δ_s^+ (E) Time
AS1	22789	22629.62	22261	27.63
AS2	13256	13176.64	13094	9.09
AS3	15139	15059.54	14967	10.24
AS4	15234	15159.76	15079	11.57
AS5	14096	14011.9	13930	10.47
AS6	18088	17994.66	17876	16.12
AS7	22823	22639.58	22237	24.5
AS8	23036	22865	22440	25.08
GNutella09	20397	20332.28	20170	18.75
WikiVote	72993	72772.26	72549	9.00

Table 4.: (MBO+) cut approximations using Δ_s^+ on graphs with a scale free structure, time in seconds.

In Figures 17, 18, and 19 we display results obtained for random modular graphs. For each of $R(2500, 2, 0.009, 0.8)$ (Figure 17), $R(4000, 20, 0.01, 0.7)$ (Figure 18), and $R(10000, 10, 0.01, 0.8)$ (Figure 19) we create 100 realisations. We then run (MBO+) with both the spectral method and the Euler method, and we run (GW). For both of the (MBO+) methods we choose either Δ_0^+, Δ_1^+, or Δ_s^+, setting $\tau = 20$ for all tests. The bar chart represents the mean of the best, average, and least cuts over all 100 realisations of the chosen random modular graph. The error bars are the corrected sample standard deviation of the results obtained over all 100 realisations.

Graph	Δ_0^+ (S) Best	Δ_0^+ (S) Avg	Δ_0^+ (S) Least	Δ_0^+ (S) Time
AS1	22578	22303.10	21844	297.79
AS2	13081	12935.80	12763	62.41
AS3	14995	14869.52	14702	90.32
AS4	15097	14994.92	14885	88.53
AS5	13952	13795.24	13561	70.81
AS6	17836	17672.50	17527	149.60
AS7	22571	22328.18	21932	294.26
AS8	22824	22585.88	22075	287.79
GNutella09	19079	18419.36	17951	72.03
WikiVote	65504	60599.74	56917	46.11

Table 5.: (MBO+) cut approximations using Δ_0^+ on graphs with a scale free structure, time in seconds.

In Figures 17, 18, and 19 we see that using either Δ_1^+ or Δ_s^+ (MBO+) with both the spectral method and the Euler method outperforms (GW) with respect to the best, average, and least cuts. In Table 7 we see that for any choice of operator and method, (MBO+) is faster on average than (GW) for our choices for random modular graphs. We note in particular that for our realisations of $R(10000, 10, 0.01, 0.8)$ the average (GW) test took just below 65 minutes, where as the average (MBO+) test using the spectral method and either Δ_1^+ or Δ_s^+ took under a minute, obtaining on average better outcomes.

7.1.5 *Weighted graphs*

In this sub-chapter we assign random weights to the edges of selected graphs from Chapter 7.1.2 and Chapter 7.1.3. To create the graphs W1 and W2 we use two of the realisations of $G(1000, 0.01)$, and multiply its edges by random real numbers drawn uniformly from in the range $[0, 2]$ and $[0, 20]$ respectively. W3 and W4 were created by using two of the realisations of $G(2500, 0.4)$ in Chapter 7.1.2, and multiplying its edges by random real numbers drawn uniformly from in the ranges $[0, 5]$ and $[0, 1]$ respectively. W5, W6, W7 were created by using three of the realisations of $G(5000, 0.001)$ in Chapter 7.1.2,

Graph	GW Best	GW Avg	GW Least	GW Time
AS1	**22864**	22346.26	20546	2324.98
AS2	**13328**	13039.10	12048	594.29
AS3	**15240**	14961.56	14050	826.65
AS4	**15328**	15015.34	14072	832.28
AS5	**14190**	13810.82	12922	721.51
AS6	**18191**	17851.24	16483	1368.35
AS7	**22901**	22421.80	21244	2321.34
AS8	**23170**	22593.10	21110	2613.62
GNutella09	**20658**	20242.02	18815	1095.04
Wiki-Vote	**73363**	71510	62886	1074.98

Table 6.: (GW) cut approximations on graphs with a scale free structure, time in seconds.



Figure 16.: A Max-Cut approximation on a random 4-modular graph.

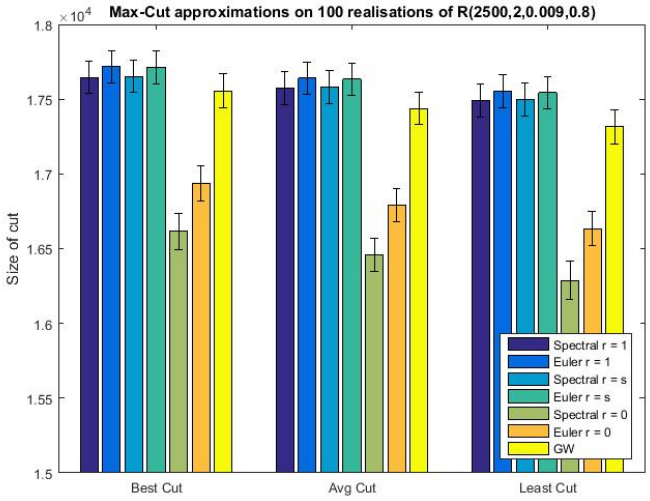


Figure 17.: Bar chart of Max-Cut approximations on 100 realisations of $R(2500, 2, 0.009, 0.8)$.

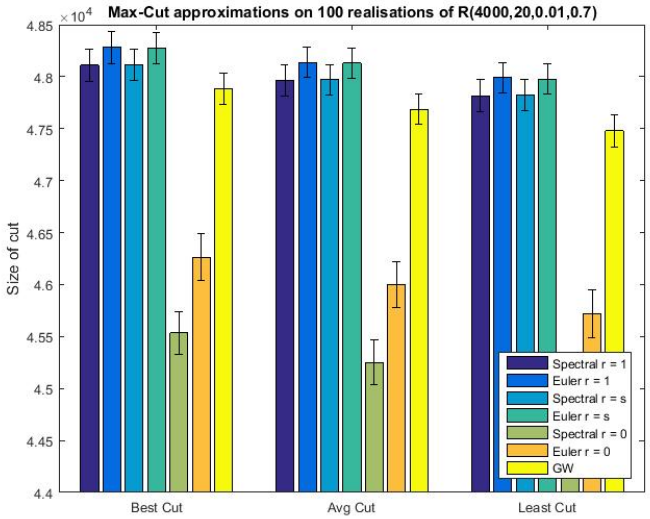


Figure 18.: Bar chart of Max-Cut approximations on 100 realisations of $R(4000, 20, 0.01, 0.7)$.

and multiplying its edges by random real numbers drawn uniformly from in the ranges $[0, 1]$, $[0, 15]$, and $[0, 50]$ respectively. W8 is the AS1 graph, whose edges are multiplied by random real numbers drawn uniformly from in the range $[0, 12]$, W9 is the AS5 graph whose edges are multiplied by random real numbers drawn uniformly from in the range $[0, 4]$ and W10 is the AS8 graph whose edges are multiplied by random real numbers drawn uniformly

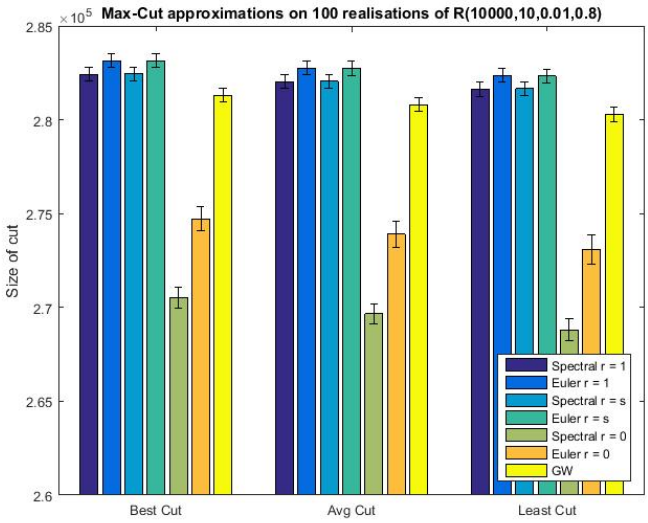


Figure 19.: Bar chart of Max-Cut approximations on 100 realisations of $R(10000, 10, 0.01, 0.8)$.

Graph	Δ_1^+ (S)	Δ_1^+ (E)	Δ_s^+ (S)	Δ_s^+ (E)	Δ_0^+ (S)	Δ_0^+ (E)	GW
$R(2500, 2, 0.009, 0.8)$	0.80	10.43	**0.79**	10.26	4.36	6.13	56.30
$R(4000, 20, 0.01, 0.7)$	**4.05**	30.46	4.49	29.52	16.26	18.19	248.25
$R(10000, 10, 0.01, 0.8)$	**49.98**	266.10	52.85	266.40	210.94	194.52	3893.87

Table 7.: Average (MBO+) and (GW) run-times for each realisation of $R(n, c, p, r)$, time in seconds.

from in the range $[0, 8]$. We run (MBO+) for all three choices of Δ^+, on all of these graphs, and compare against (GW) in Table 10. We set $\tau = 20$ for both the spectral decomposition method and the Euler method.

We saw that (MBO+) using the spectral method produced larger cuts than (GW) on the random graphs considered in Chapter 7.1.2; when assigning random weights to the edges of these random graphs the same conclusion holds. We see in Tables 8, 9, and 10 that for this collection of random graphs (MBO+) using the spectral method (with either Δ_1^+ or Δ_s^+ used) outperforms (GW) with respect to the best, average, and smallest obtained cut sizes, and the run time. In Chapter 7.1.3 we saw that (MBO+) using both the spectral method and the Euler method produced better average and smallest cuts than (GW) on the scale free graphs considered in that section, but the best cut sizes

were produced more often by (GW). These weighted examples support the same conclusions. The blank results in Table 11 for the Euler method using Δ_0^+ are due to (MBO+) producing trivial results for these choices as stated in Chapter 7.1.1.

Graph	Δ_1^+ (S) Best	Δ_1^+ (S) Avg	Δ_1^+ (S) Least	Δ_1^+ (S) Time
W1	3612.00	3569.08	3537.10	0.47
W2	36487.51	36082.58	35687.87	**0.30**
W3	1622125.53	**1620885.77**	**1619371.25**	8.09
W4	323926.34	323639.05	**323321.92**	8.59
W5	5054.26	5033.54	5010.38	**4.00**
W6	74560.24	74218.26	73776.17	**3.90**
W7	252448.52	251045.03	249459.89	4.18
W8	137202.14	135952.94	133480.08	16.17
W9	28351.01	28194.96	28009.15	**3.99**
W10	92376.49	91570.35	90172.90	17.02

Graph	Δ_1^+ (E) Best	Δ_1^+ (E) Avg	Δ_1^+ (E) Least	Δ_1^+ (E) Time
W1	**3622.58**	**3580.53**	3548.82	1.41
W2	**36530.25**	**36191.16**	**35928.56**	1.67
W3	1603390.76	1600505.43	1596558.94	185.03
W4	320347.01	319612.93	318849.26	195.66
W5	**5104.45**	**5081.95**	**5063.64**	15.31
W6	**75499.50**	**75175.73**	**74833.80**	15.70
W7	**255793.23**	**254569.97**	**253091.91**	15.71
W8	137569.32	**136896.1**	**136094.60**	23.83
W9	28545.45	**28369.43**	**28141.76**	9.24
W10	93021.06	**92489.04**	**91626.99**	25.37

Table 8.: (MBO+) cut approximations using Δ_1^+ on randomly weighted graphs, time in seconds.

Graph	Δ_s^+ (S) Best	Δ_s^+ (S) Avg	Δ_s^+ (S) Least	Δ_s^+ (S) Time
W1	3601.29	3569.23	3545.85	**0.33**
W2	36192.09	36059.80	35867.83	0.49
W3	**1622372.91**	1620484	1618809.76	8.40
W4	**323933.40**	**323642.4**	323114.45	7.65
W5	5068.19	5041.94	5015.16	4.50
W6	74844.37	74505.45	73963.79	4.67
W7	253043.96	251668.30	250600.35	**4.12**
W8	137195.52	136360.17	134856.06	**15.38**
W9	28389.38	28227.09	28067.66	4.12
W10	92439.42	91952.98	90488.33	**15.33**

Graph	Δ_s^+ (E) Best	Δ_s^+ (E) Avg	Δ_s^+ (E) Least	Δ_s^+ (E) Time
W1	3614.37	3577.56	3542.19	1.40
W2	36321.80	36150.05	35910.90	1.53
W3	1604257.12	1600145.68	1597577.4	187.88
W4	320691.88	319596.27	318900.13	199.01
W5	5096.55	5072.36	5041.89	15.9
W6	75456.87	75089.73	74745.17	18.09
W7	255316.85	253821.64	252527.13	15.48
W8	137282.02	136475.24	134333.1	24.51
W9	28445.94	28258.64	28101.22	9.18
W10	92731.62	92093.05	90448.61	24.36

Table 9.: (MBO+) cut approximations using Δ_s^+ on randomly weighted graphs, time in seconds.

7.1.6 *Large graphs*

Since the Euler method has a time complexity of $\mathcal{O}(|E|)$, in this section we show that (MBO+) using the Euler method can provide Max-Cut approximations in a respectable time on large sparse datasets. The graphs Amazon0302 and Amazon0601 are networks in which the nodes represent products and an edge exists between two nodes if the corresponding products are frequently co-purchased; both of these networks were constructed in 2003. GNutella31 depicts a peer to peer file sharing network in 2002. PA RoadNet is a road network of Pennsylvania with intersections and endpoints acting as nodes

Graph	GW Best	GW Avg	GW Least	GW Time
W1	3585.17	3535.63	3494.26	5.74
W2	36101.30	35698.47	35151.60	6.07
W3	1620705.80	1618813.52	1616502.33	43.58
W4	323573.40	323275.84	322795.83	44.09
W5	5038.00	5000.74	4953.71	265.27
W6	74372.75	73852.36	73293.27	241.33
W7	251802.56	250316.08	248098.85	263.44
W8	**138159.14**	135899.20	129576.95	2629.60
W9	**28705.35**	28169.25	26422.54	689.16
W10	**93547.26**	91571.68	87487.99	2646.94

Table 10.: (GW) cut approximations on randomly weighted graphs, time in seconds.

and roads connecting them acting as edges. Email-Enron is a network where each edge represents an email being sent between two people. BerkStan-Web is a network of inter-domain and intra-domain hyperlinks between pages on the domains berkeley.edu and stanford.edu in 2002. Stanford is a network of hyperlinks between pages on the domain stanford.edu in 2002. All of these datasets were obtained from the website [95]. The graph WWW1999 is a model of the Internet in 1999 with edges depicting hyperlinks between websites, obtained from [2]. Table 12 displays the properties of these graphs. Table 13 displays the results we obtained on these graphs choosing Δ_1^+ as our operator, the Euler method as our signless diffusion solver, and $\tau = 10$. For these large graphs, we are unable to obtain results for comparison using (GW), because (GW) requires too much memory for it to run on the same computer setup.

Graph	Δ_0^+ (S) Best	Δ_0^+ (S) Avg	Δ_0^+ (S) Least	Δ_0^+ (S) Time
W1	3413.96	3345.32	3276.63	0.61
W2	34784.30	34304.33	33627.16	0.51
W3	1602346.52	1600022.33	1595791.12	**6.97**
W4	320251.52	319940.38	319663.40	**6.25**
W5	4793.44	4761.72	4715.51	18.66
W6	71219.49	70427.83	69643.31	18.93
W7	239991.72	237647.45	235617.15	19.17
W8	134097.55	131088.97	126123.70	272.56
W9	27528.99	26554.77	25501.34	69.63
W10	90271.70	88031.84	83130.60	264.89

Graph	Δ_0^+ (E) Best	Δ_0^+ (E) Avg	Δ_0^+ (E) Least	Δ_0^+ (E) Time
W1	3524.24	3456.55	3406.93	1.03
W2	35664.18	35040.71	34383.57	1.03
W3	1605419.97	1602251.82	1597064.59	203.27
W4	320321.63	319809.73	319237.08	192.51
W5	5017.66	4983.90	4954.63	7.76
W6	74195.87	73688.97	73231.67	7.33
W7	251330.73	249754.88	248091.06	7.51
W8	-	-	-	-
W9	-	-	-	-
W10	-	-	-	-

Table 11.: (MBO+) cut approximations using Δ_0^+ on randomly weighted graphs, time in seconds.

Graph	$\lvert V\rvert$	$\lvert E\rvert$	d_-	d_+
Amazono302	262111	899792	1	420
Amazono601	403394	2443408	1	2752
GNutella31	62586	147892	1	95
PA RoadNet	1088092	1541898	1	9
Email-Enron	36692	183831	1	1383
BerkStan-Web	685230	6649470	1	84290
Stanford	281904	1992636	1	38625
WWW1999	325729	1090108	1	10721

Table 12.: Properties of our large datasets we are testing on.

Graph	Δ_1^+ (E) Best	Δ_1^+ (E) Avg	Δ_1^+ (E) Min	Δ_1^+ (E) Time
Amazon0302	618942	618512.18	618030	0.49
Amazon0601	1580070	1576960.80	1571089	1.90
GNutella31	116552	116213.74	115916	0.06
PA RoadNet	1380131	1379797.90	1379416	0.64
Email-Enron	112665	111680.24	110279	0.02
BerkStan-Web	5335813	5319662.06	5281630	0.83
Stanford	1585802	1580445.14	1570469	0.47
WWW1999	813000	809329.52	806130	0.21

Table 13.: Results of (MBO+) using Δ_1^+ and the Euler method on large datasets, time in hours.

7.2 MAX-CUT - PARAMETER TESTS

7.2.1 *Variable k^**

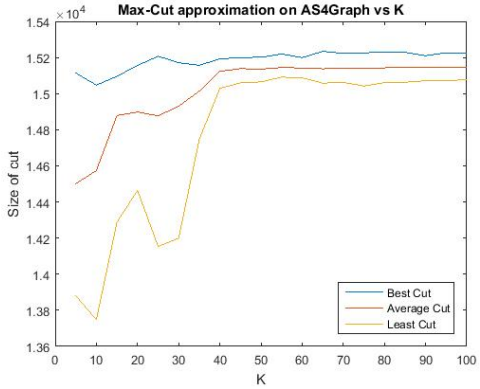
As stated in Chapter 4.2.2, the computational advantage of (MBO+) using the spectral method is that not all the eigenpairs of Δ^+ need to be used. In practice, if k^* is large enough, the cut sizes obtained by (MBO+) using the spectral method does not improve significantly when k^* is increased further. The plots in Figure 20 highlight this. For these three tests we fixed the initial conditions, the choice of operator Δ_1^+, and $\tau = 20$ for each respective graph. For Figure 20a we plot the best, average, and least cuts for each choice of K. For Figure 20b and Figure 20c we plot the mean of the best, average, and least cuts over all 100 graphs for each choice of k^*. The error bars indicate the corrected sample standard deviation of the best, average, and least cuts. We ran (MBO+) using the spectral method on the AS4 graph, increasing the value for k^* in increments of 5 from 5 until 100. The plot in Figure 20a shows that at $k^* = 40$ the best, average, and least cut size changes very little for increasing k^*. For Figure 20b we ran (MBO+) on the 100 realisations of $G(5000, 0.001)$ from Chapter 7.1.2, increasing k^* in increments of 10 from 10 until 200. For Figure 20c we ran (MBO+) on the 100

realisations of $R(2500, 2, 0.009, 0.8)$, increasing k^* in increments of 5 from 5 until 100. The plots in Figure 20b and Figure 20c show that for our choices of Erdös-Rényi and random modular graphs increasing k^* increases the cut sizes. We also note that the best, average, and minimum cut sizes plateau.

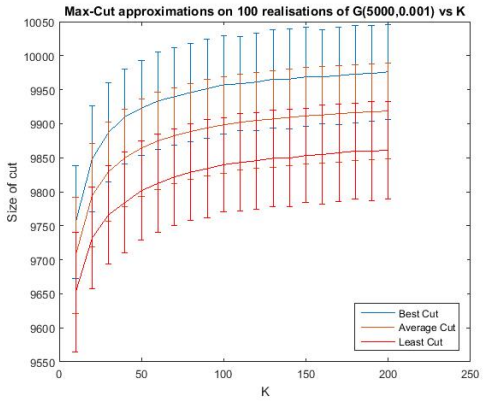
For large graphs, however, finding the value of k^* beyond which the produced cut sizes plateau is problematic. We ran (MBO+) using the spectral method with Δ_1^+ on the Amazon0302 graph, increasing k^* in increments of 100 starting from 100 to 2600. As shown in Figure 21 the best, average, and least outcomes of (MBO+) are still increasing at the end of the range of k^* values we plotted. For $k^* = 200$ and $k^* = 2600$ the run time of (MBO+) was 12 minutes and 26 hours, respectively; this increase in computation time resulted in a 3% increase in cut values. Comparing the cut size obtained for $k^* = 2600$ with the cut sizes obtained on Amazon0302 in Table 13 we see that using the Euler method as the signless diffusion solver is more accurate and significantly faster.

7.2.2 *Variable τ*

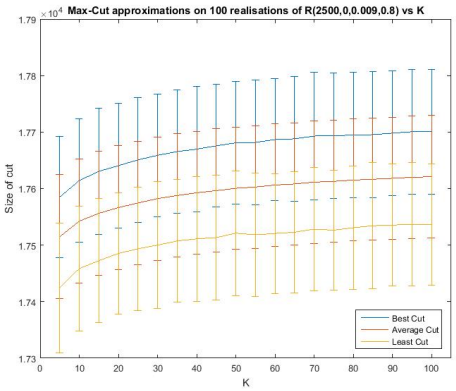
Other than the pinning condition stated in Chapter 4.2.4, currently we have very little information on which to base our choice of τ. In this sub-chapter we compare the cut sizes obtained by (MBO+) against the variable τ. We choose Δ_1^+ as the signless Laplacian operator and the spectral method as the signless diffusion solver. Figure 22 displays the obtained cut sizes from (MBO+) on three (sets of) graphs and compares against τ. For Figure 22a we plot the best, average, and least cuts for each choice of τ. For Figure 22b and Figure 22c we plot the mean of the best, average, and least cuts over all 100 graphs for each choice of τ. The error bars indicate the corrected sample standard deviation of the best, average, and least cuts. We ran (MBO+) using the spectral method on the AS4 graph, increasing the value for τ in increments of 5 starting from 5 until 500. In Figure 22a we see in this experiment that $5 \leq \tau \leq 40$ produces the best results with respect to our cut sizes. We also



(a) AS4: Cut size as function of k^*, $\tau = 20$.



(b) 100 realisations of $G(5000, 0.001)$: Cut size as function of k^*, $\tau = 20$.



(c) 100 realisations of $R(2500, 2, 0.009, 0.8)$: Cut size as function of k^*, $\tau = 20$.

Figure 20.: Cut size as function of K for three graphs (best viewed in colour).

see that for $330 \leq \tau \leq 480$ the best, average, and least cuts are almost identical. For Figure 22b we ran (MBO+) on the 100 realisations of $G(5000, 0.001)$

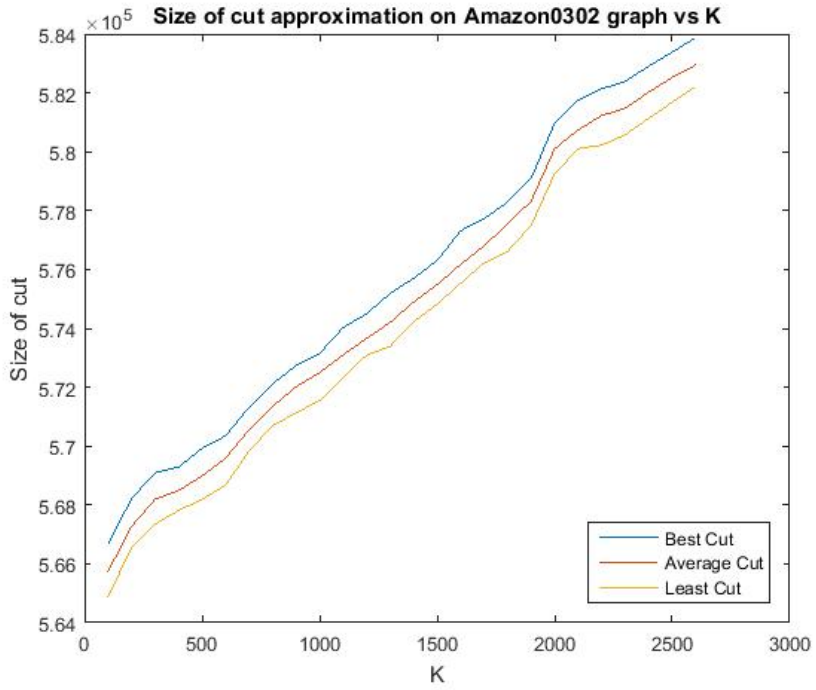
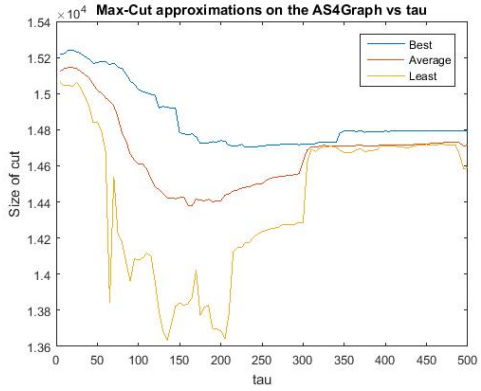


Figure 21.: Comparison of cut size approximation vs k^* on Amazon0302 graph.

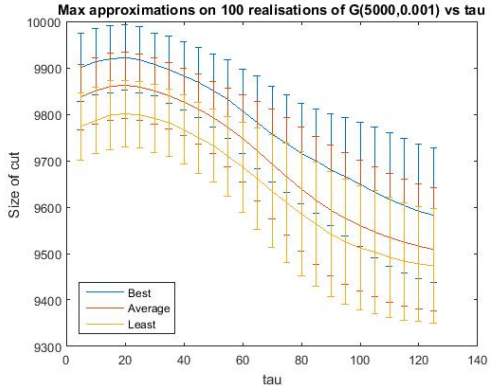
from Chapter 7.1.2, increasing τ in increments of 5 starting from 5 until 125. For Figure 22c we ran (MBO+) on the 100 realisations of $R(4000, 20, 0.01, 0.7)$, increasing τ in increments of 5 starting from 5 until 100. In Figure 22b and Figure 22c we see the general trend that increasing τ beyond 20 decreases the mean over the best, average, and least cuts over all 100 realisations of $G(5000, 0.001)$.

7.2.3 *Implicit Euler scheme*

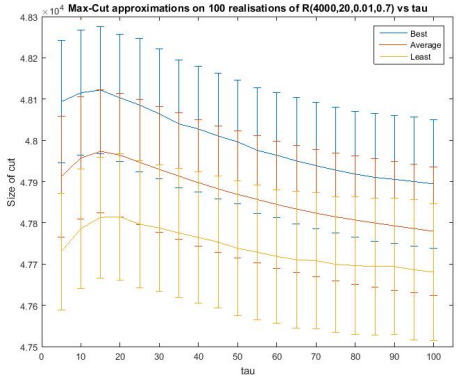
On the random graphs we tested on in Chapter 7.1.2 and Chapter 7.1.4 our explicit Euler scheme using Δ_0^+ produced non-trivial cut sizes. However, for the scale free graphs in Chapter 7.1.3 and Chapter 7.1.5 we did not find a value of τ or dt such that the cuts induced from (MBO+) were non-trivial. In this sub-chapter we show that we can solve the Euler equation implicitly in order to obtain non-trivial cut sizes with the operator Δ_0^+, subject to suitable



(a) AS4: Cut size as function of τ, $k^* = 89$.



(b) 100 realisations of $G(5000, 0.001)$: Cut size as function of τ, $k^* = 49$.



(c) 100 realisations of $R(4000, 20, 0.01, 0.7)$: Cut size as function of τ, $k^* = 40$.

Figure 22.: Cut size as function of τ for three graphs (best viewed in colour).

choices of dt and τ. However, the results are significantly inferior to the operators Δ_1^+ and Δ_s^+ for the implicit Euler scheme. We also compare the

(MBO+) results obtained using the implicit scheme to the results obtained using the explicit scheme for a set of random graphs.

We run (MBO+) using the implicit Euler scheme on the AS4 and AS8 graph from Chapter 7.1.3 and the W9 graph from Chapter 7.1.5. We choose $dt = 0.2$ and $\tau = 20$ when Δ_1^+ or Δ_s^+ is the operator. For Δ_0^+ we set $dt = 0.0005$ and $\tau = 0.05$ for the AS4 graph, and for the AS8 graph and the W9 graph we set $dt = 0.0001$ and $\tau = 0.01$. Table 14 shows that (MBO+) using the implicit Euler scheme with Δ_0^+ and our choice of parameters produces cut sizes, however they are significantly smaller in comparison to using Δ_1^+ or Δ_s^+.

Graph	Δ_1^+ Best	Δ_s^+ Best	Δ_0^+ Best
AS4	15276	**15279**	9259
AS8	**23083**	23033	13725
W9	**28553.66**	28485.28	17146.69

Graph	Δ_1^+ Avg	Δ_s^+ Avg	Δ_0^+ Avg
AS4	**15196.52**	15175.52	9124.68
AS8	**22934.30**	22844.16	13585.56
W9	**28360.46**	28294.40	16847.92

Graph	Δ_1^+ Least	Δ_s^+ Least	Δ_0^+ Least
AS4	**15124**	15056	8964
AS8	**22521**	22454	13477
W9	**28103.28**	28075.62	16521.43

Graph	Δ_1^+ Time	Δ_s^+ Time	Δ_0^+ Time
AS4	47.83	50.94	**7.47**
AS8	105.22	114.74	**11.48**
W9	38.61	42.57	**5.26**

Table 14.: Cut sizes obtained by (MBO+) using the implicit Euler scheme on scale free graphs, time in seconds.

We run (MBO+) on the 100 realisations of $G(1000, 0.01)$ and $R(4000, 20, 0.01, 0.7)$ in Chapter 7.1.2 and Chapter 7.1.4 respectively, using the implicit and explicit Euler method for each operator $\Delta^+ \in \{\Delta_0^+, \Delta_1^+, \Delta_s^+\}$. We choose the same values of τ and dt as chosen in Chapter 7.1.2 and Chapter 7.1.4, fixing the initial conditions for both methods. Figure 23 and Figure 24 show that the average obtained cut sizes using the implicit Euler method are slightly better than the average obtained cut sizes obtained using the explicit method. However, Table 15 shows that (MBO+) using the explicit Euler method produces cut sizes in less time than using the implicit Euler method on these sets of random graphs. This is why we choose the explicit method for the Euler method in Chapter 7.1.

Graph	Δ_1^+ (I)	Δ_1^+ (E)	Δ_s^+ (I)	Δ_s^+ (E)	Δ_0^+ (I)	Δ_0^+ (E)
$G(1000, 0.01)$	3.36	1.82	3.28	1.79	2.20	**1.19**
$R(4000, 20, 0.01, 0.7)$	62.97	44.23	62.16	44.18	41.53	**24.01**

Table 15.: Average (MBO+) run-times for each realisation of $G(1000, 0.01)$ and $R(4000, 20, 0.01, 0.7)$, time in seconds. (I) indicates the implicit Euler method and (E) indicates the explicit Euler method.

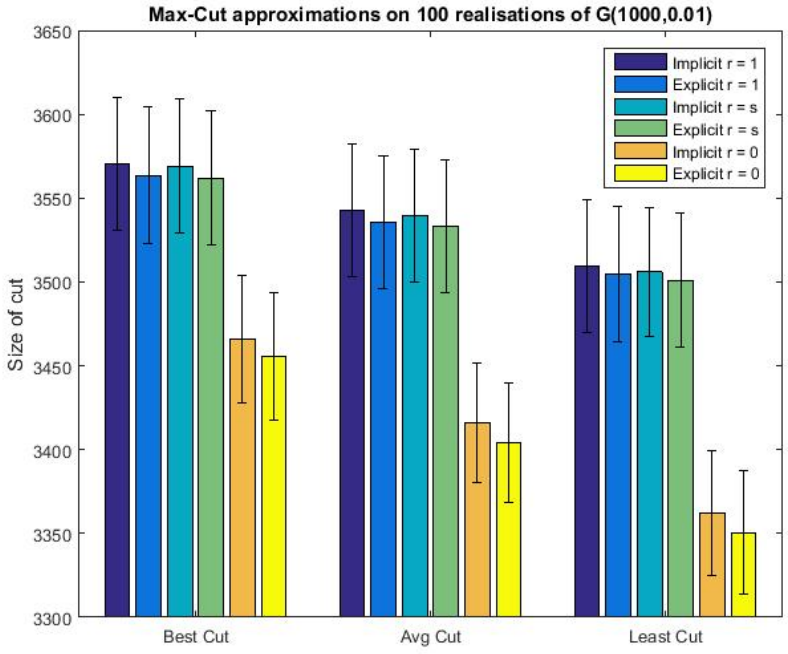


Figure 23.: Bar chart of Max-Cut approximations on 100 realisations of $G(1000, 0.01)$ using the implicit Euler method and the explicit Euler method.

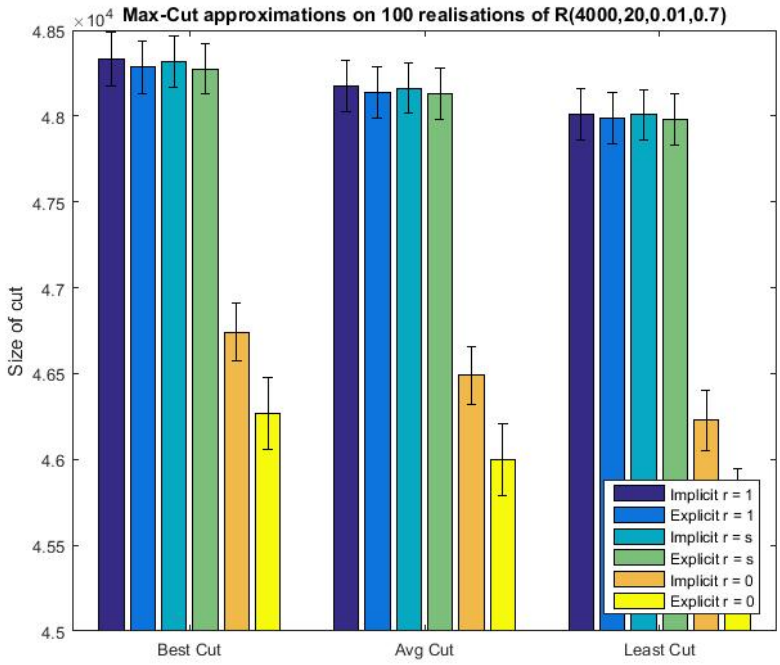


Figure 24.: Bar chart of Max-Cut approximations on 100 realisations of $R(4000, 20, 0.01, 0.7)$.

7.3 MINIMIZING f_ε^+ BY CONVEX-CONCAVE SPLITTING

As elaborated in Chapter 4.2.1, we can minimize f_ε^+ in (2) by solving the gradient flow equation directly, instead of using the MBO scheme. In Chapter 7.3, we introduce the algorithm (Cut), which minimizes f_ε^+ in (2) by solving the gradient flow equation of f_ε^+. We solve (27) by applying a convex-concave scheme similar to the one stated in Chapter 5.2.1. We use the convex-concave scheme used in [9] to minimize f_ε in (1) in order to minimize f_ε^+, since f_ε^+ and f_ε differ only in the spectral truncation of the signless Laplacian in f_ε^+ and the spectral truncation of the graph Laplacian in f_ε.

The algorithm (Cut) differs from (MBO+) only in the choice of solving (27) for a small time step, instead of solving (28). The remaining steps are equivalent; however, for completeness, we present the (Cut) algorithm below.

For all tests in Chapter 7.3, we choose $\tau = 20$, $k^* = \lfloor \frac{n}{100} \rfloor$, and $\eta = 10^{-8}$ for both (Cut) and (MBO+). For both algorithms, we test for both Δ_1^+ and Δ_s^+ as the choice of Laplacian. For all tests using (MBO+) in Chapter 7.3, we choose the spectral decomposition method instead of the Euler method in order for the algorithm to be more similar to (Cut). For all tests, we fix the initial conditions for both (Cut) and (MBO+). For (Cut), we choose $\varepsilon = 0.1$.

In Figure 25 and Figure 26, we run (Cut) and (MBO+) on 100 realisations of $G(1000, 0.01)$ and $R(2500, 2, 0.009, 0.8)$, respectively. We choose the same realisations as the ones tested on in Chapter 7.1.2 and Chapter 7.1.4, respectively.

Figure 25 and Figure 26 show that for both choices of signless Laplacian, (MBO+) produces larger best, average, and least cuts compared to (Cut). We note that in Figure 25 and Figure 26 the least cuts produced by (MBO+) for either choice of signless Laplacian are larger than the best cuts produced by (Cut) for both signless Laplacians.

In Table 16, we see that the run times of (MBO+) for either choice of signless Laplacian are lower than the run times of (Cut), for either choice of signless Laplacian.

Algorithm (Cut): The signless Ginzburg–Landau gradient flow algorithm

Data: A signless graph Laplacian $\Delta^+ \in \{\Delta_0^+, \Delta_1^+, \Delta_s^+\}$ corresponding to a graph $G \in \mathcal{G}$, a signless diffusion time $\tau > 0$, an initial condition $\mu^0 := \chi_{S_0} - \chi_{S_0^c}$ corresponding to a node subset $S_0 \subset V$, a time step dt, an epsilon value $\varepsilon > 0$, and a stopping criterion tolerance η.

Output: A sequence of functions $\{\mu^j\}_{j=0}^N \subset \mathcal{V}^b$ giving the signless MBO evolution of μ^0, a sequence of corresponding cuts $\{C^j\}_{j=0}^N \subset \mathcal{C}$ and their sizes $\{s(C^j)\}_{j=0}^N \subset [0, \infty)$, with largest value s^*.

for $j = 1$ **to** stopping criterion is satisfied, **do**

Signless diffusion step: Compute $u^*(\tau)$, where $u^* \in \mathcal{V}$ is the solution of the initial value problem

$$\begin{cases} \frac{du(t)}{dt} = -\Delta^+ u(t) - \frac{1}{\varepsilon} d_i^{-r} W'(u_i), & \text{for } t > 0, \\ u(0) = \mu^j. \end{cases}$$

Threshold step: Define $\mu^j \in \mathcal{V}^b$ by, for $i \in V$,

$$\mu_i^j := T(u_i^*(\tau)) := \begin{cases} 1, & \text{if } u_i^*(\tau) > 0, \\ -1, & \text{if } u_i^*(\tau) \leq 0. \end{cases}$$

Define the cut $C^j := V_{-1}^j | V_1^j$, where $V_{\pm 1}^j := \{i \in V : \mu_i^j = \pm 1\}$ and compute $s(C^j)$.

Set $N = j$.

if $\frac{\|\mu^j - \mu^{j-1}\|_2^2}{\|\mu^j\|_2^2} < \eta$ **then**

Stop

Find the largest cut size: Set $s^* := \max_{1 \leq j \leq N} s(C^j)$.

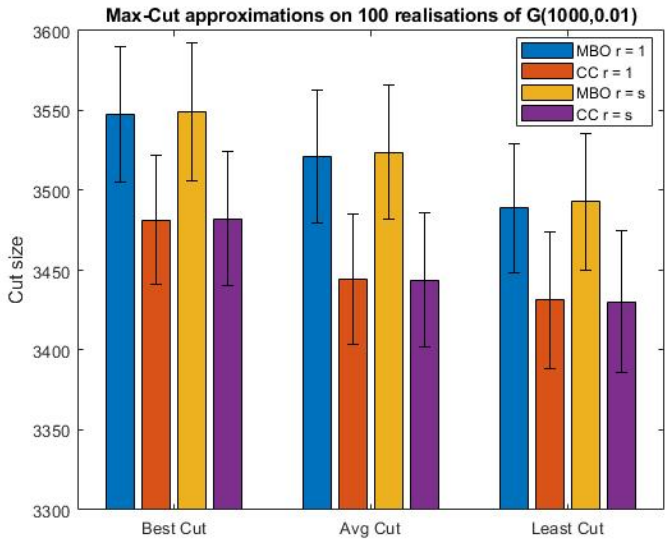


Figure 25.: Bar chart of Max-Cut approximations on 100 realisations of $G(1000, 0.01)$.

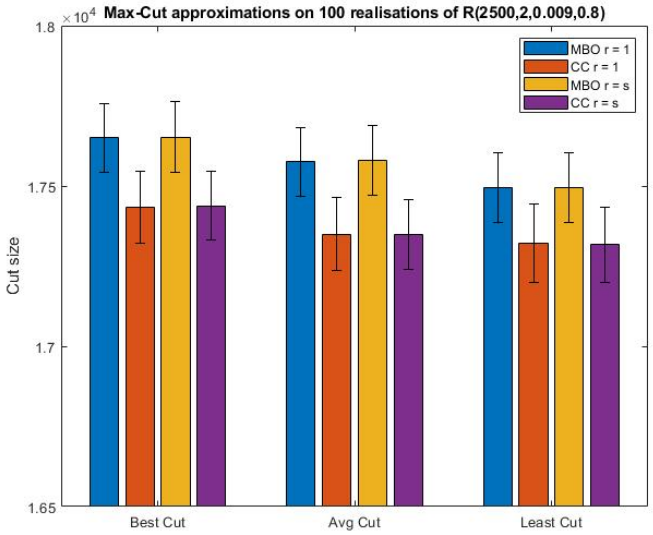


Figure 26.: Bar chart of Max-Cut approximations on 100 realisations of $R(2500, 2, 0.009, 0.8)$.

In Table 18 we list results of (Cut) and (MBO+) on the scale free graphs in Chapter 7.1.3. We see that on all scale free graphs tested on, (MBO+) for either choice of signless Laplacian produced larger cuts than (Cut), for either choice of signless Laplacian.

Graph	(Cut) Δ_1^+	(MBO+) Δ_1^+	(Cut) Δ_s^+	(MBO+) Δ_s^+
$G(1000, 0.01)$	4.45	0.14	4.49	**0.13**
$R(2500, 2, 0.009, 0.8)$	11.62	0.77	11.29	**0.73**

Table 16.: Average (Cut) and (MBO+) run-times for each realisation of $G(n, p)$ and $R(n, c, p, r)$, time in seconds.

In Table 17 we see that on 6 out of 10 scale free graphs we tested on, (MBO+) produced significantly lower run times in comparison to (Cut). However, it is interesting to note that on AS4, the run time of (Cut) using Δ_1^+ was significantly lower than the other scale free graphs tested on, and on AS2, AS6, and GNutella, the run time of (Cut) using Δ_s^+ was significantly lower than the other scale free graphs.

Graph	(Cut) Δ_1^+	(MBO+) Δ_1^+	(Cut) Δ_s^+	(MBO+) Δ_s^+
AS1	376.19	14.71	380.08	**10.87**
AS2	141.63	3.87	**1.38**	3.14
AS3	177.73	5.16	179.72	**3.82**
AS4	**3.15**	5.52	188.22	4.79
AS5	154.72	4.43	153.02	**3.66**
AS6	262.35	9.39	**3.07**	6.70
AS7	377.01	14.74	368.64	**10.40**
AS8	382.91	15.11	378.95	**10.33**
GNutella	158.76	7.99	**1.92**	7.14
WikiVote	130.37	2.33	132.08	**1.86**

Table 17.: Average (Cut) and (MBO+) run-times for each scale free graph, time in seconds.

Graph	(Cut) Δ_1^+ (Best)	(MBO+) Δ_1^+ (Best)	(Cut) Δ_s^+ (Best)	(MBO+) Δ_s^+ (Best)
AS1	22575	22766	22578	**22800**
AS2	13113	13228	13116	**13245**
AS3	15071	15099	14982	**15157**
AS4	15096	**15236**	15096	15233
AS5	13987	**14074**	14012	14066
AS6	17949	18063	17970	**18070**
AS7	22573	22768	22531	**22786**
AS8	22775	22960	22754	**23026**
GNutella09	19996	20259	20067	**20308**
WikiVote	74039	74311	74039	**74432**

Graph	(Cut) Δ_1^+ (Avg)	(MBO+) Δ_1^+ (Avg)	(Cut) Δ_s^+ (Avg)	(MBO+) Δ_s^+ (Avg)
AS1	22315.62	22575.18	22325.14	**22616.72**
AS2	13033.12	13150.70	13015.04	**13177.46**
AS3	14893.62	15028.82	14877.23	**15077.98**
AS4	14985.74	15145.44	15002.12	**15168.08**
AS5	13856.06	**13992.42**	13849.20	13990.76
AS6	17756.38	**17977.84**	17849.14	17970.20
AS7	22306.08	22580.04	22353.08	**22654.02**
AS8	22266.96	22724.78	22480.26	**22819.04**
GNutella09	19769.02	20151.12	19877.68	**20186.84**
WikiVote	70975.42	74117.92	70855.00	**74283.54**

Graph	(Cut) Δ_1^+ (Least)	(MBO+) Δ_1^+ (Least)	(Cut) Δ_s^+ (Least)	(MBO+) Δ_s^+ (Least)
AS1	22203	22190	21825	**22361**
AS2	12943	13079	12914	**13091**
AS3	14807	14929	14722	**14955**
AS4	14832	**15079**	14871	15068
AS5	13516	**13930**	12609	13900
AS6	16923	17845	17629	**17865**
AS7	22203	22233	22057	**22431**
AS8	21232	22344	22180	**22511**
GNutella09	19463	19988	19811	**20048**
WikiVote	69949	73945	69652	**74153**

Table 18.: (Cut) and (MBO+) Max-Cut approximations on graphs with a scale free structure, time in seconds.

7.4 REDUCTION OF τ

7.4.1 *Introduction*

In some gradient descent algorithms there is a notable improvement when the time step is set to linearly decrease along iterations of the algorithm, in comparison to keeping the time step stationary. In Chapter 7.4, we show that linearly reducing τ along iterations of (MBO+) can produce some improved results compared to the results obtained in Chapter 7.1. We introduce the (MBO$\downarrow$) algorithm below, and compare some of the cut sizes obtained by (MBO$\downarrow$) against the cut sizes obtained by (MBO+).

We compute the (MBO+) and (MBO$\downarrow$) evolutions for 50 initial conditions chosen at random from $\mathcal{V}^b$, using both the spectral decomposition method proposed in Chapter 4.2.2 and the explicit Euler method proposed in Chapter 4.2.3 for both algorithms. We use the same 50 initial conditions for each method. We state the greatest (Best), average (Avg), and smallest (Least) sizes of cuts obtained by the 50 initial conditions of both algorithms. We run (MBO+) and (MBO$\downarrow$) using Δ_1^+. For all tests in this appendix, we choose $\tau = \tau_0 = 20$. For all tests using (MBO$\downarrow$), we test for all $N \in \{20, 25 \dots, 100\}$.

For all tests in Chapter 7.4.2 and Chapter 7.4.3, we plot the results obtained for each choice of N for (MBO$\downarrow$) and plot them against the results obtained using (MBO+). The (MBO+) results using the spectral decomposition method and the explicit Euler method are expressed using a straight line parallel to the N axis since (MBO+) does not depend on the parameter N.

7.4.2 *Erdös-Rényi and random modular graphs*

For each of $G(1000, 0.01)$, $G(5000, 0.001)$, $R(2500, 2, 0.009, 0.8)$, and $R(4000, 20, 0.01, 0.7)$ we choose the same 100 realisations as in Chapter 7.1.2 and Chapter 7.1.4. The plots in Figure 27, Figure 28, Figure 29, and Figure 30

Algorithm (MBO↓): The signless graph MBO algorithm with a linearly decreasing time step

Data: A signless graph Laplacian $\Delta^+ \in \{\Delta_0^+, \Delta_1^+, \Delta_s^+\}$ corresponding to a graph $G \in \mathcal{G}$, an initial signless diffusion time $\tau_0 > 0$, an initial condition $\mu^0 := \chi_{S_0} - \chi_{S_0^c}$ corresponding to a node subset $S_0 \subset V$, a time step dt, $N \in \mathbb{N}$, and a stopping criterion tolerance $\eta > 0$.

Output: A sequence of functions $\{\mu^j\}_{j=0}^M \subset \mathcal{V}^b$ giving the signless MBO evolution of μ^0, a sequence of corresponding cuts $\{C^j\}_{j=0}^M \subset \mathcal{C}$ and their sizes $\{s(C^j)\}_{j=0}^M \subset [0, \infty)$, with largest value s^*.

for $j = 1$ **to** stopping criterion is satisfied, **do**

Decrease τ step: Set $\tau := (1 - \frac{j-1}{N})\tau_0$.

Signless diffusion step: Compute $u^*(\tau)$, where $u^* \in \mathcal{V}$ is the solution of the initial value problem

$$\begin{cases} \frac{du(t)}{dt} = -\Delta^+ u(t), & \text{for } t > 0, \\ u(0) = \mu^{j-1}. \end{cases}$$

Threshold step: Define $\mu^j \in \mathcal{V}^b$ by, for $i \in V$,

$$\mu_i^j := T(u_i^*(\tau)) := \begin{cases} 1, & \text{if } u_i^*(\tau) > 0, \\ -1, & \text{if } u_i^*(\tau) \leq 0. \end{cases}$$

Define the cut $C^j := V_{-1}^j | V_1^j$, where $V_{\pm 1}^j := \{i \in V : \mu_i^j = \pm 1\}$ and compute $s(C^j)$.

Set $M = j$.

if $\frac{\|\mu^j - \mu^{j-1}\|_2^2}{\|\mu^j\|_2^2} < \eta$ **then**

Stop

if $s(C^j) = \frac{1}{2}\sum_{i,j \in V} \omega_{ij}$ **then**

Stop

Find the largest cut size: Set $s^* := \max_{1 \leq j \leq M} s(C^j)$.

represent the mean of the best, average, and least cuts over all 100 realisations of the chosen random graph.

Figure 27 shows that for all choices of N, (MBO$\downarrow$) using the Euler method achieves superior results compared against (MBO+) using the Euler method, with respect to best, average, and least cut sizes produced, with the exception being the choice of $N = 20$ with respect to the least cut produced. Figure 27 further shows that for all choices of N, (MBO$\downarrow$) using the spectral decomposition method achieves superior results compared against (MBO+) using the spectral decomposition method, with respect to best and average cut sizes produced, with inferior least cut sizes being produced for some values of N.

Figure 28 shows that for all choices of N, (MBO$\downarrow$) using the Euler method achieves superior results compared against (MBO+) using the Euler method, with respect to best, average, and least cut sizes produced. The figure further shows that for larger N, (MBO$\downarrow$) using the spectral method marginally outperforms (MBO+) using the spectral method with respect to best, average, and least cut sizes produced.

Figure 29 shows that for all choices of N, (MBO$\downarrow$) using the Euler method achieves superior results compared against (MBO+) using the Euler method, with respect to best, average, and least cut sizes produced, with the exception being the choices $N = 20$ and $N = 25$ with respect to the Least cut produced. Figure 29 further shows that for all choices of N, (MBO$\downarrow$) using the spectral decomposition method achieves superior results compared against (MBO+) using the spectral decomposition method, with respect to Best and Avg cut sizes produced, with inferior Least cut sizes being produced for some values of N.

Figure 30 shows that for $N \geq 75$, (MBO$\downarrow$) using the Euler method achieves marginally superior results compared against (MBO+) using the Euler method, with respect to best, and average cut sizes produced, with inferior cut sizes being produced for least cuts for $N \leq 80$. Figure 30 further shows that for all choices of N, (MBO$\downarrow$) using the spectral decomposition method achieves marginally superior results compared against (MBO+) us-

ing the spectral decomposition method, with respect to best and average cut sizes produced, for $N \geq 35$, and superior least cut sizes produced for some values of N.

Figure 31 shows that for all four tests, (MBO↓) using the Euler method has reduced run times over (MBO+) using the Euler method. Figure 31 further shows that for $G(1000, 0.01)$ and $R(2500, 2, 0.009, 0.8)$, (MBO+) using the spectral method and (MBO↓) using the spectral method have similar run times for larger choices of N. However, for $G(5000, 0.001)$ and $R(4000, 20, 0.01, 0.7)$, (MBO↓) using the spectral methods has larger run times in comparison to (MBO+) using the spectral method.

7.4.3 *Scale free graphs*

In Figures 32-36 we list results for the scale free graphs on which we tested on in Chapter 7.1.3. We run (MBO+) with both the spectral method and the Euler method, and then (MBO↓) with both the spectral method and the Euler method. For both (MBO+) and (MBO↓), we choose Δ_1^+, setting $\tau = 20$ for all tests.

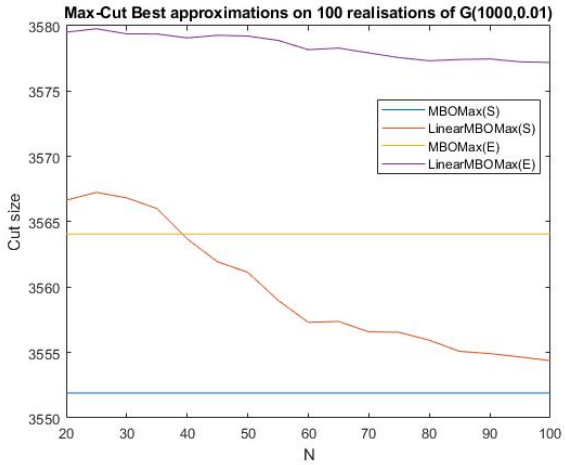
In Chapter 7.1.3, we showed that using either Δ_1^+ or Δ_s^+ as the choice of signless Laplacian in (MBO+) using the Euler method or the spectral method resulted in superior average and least obtained cut sizes over (GW), but (GW) produced superior best cut sizes. Therefore in Figures 32-36, we plot the greatest cuts obtained using (GW) in Chapter 7.1.3 and compare them against the greatest cuts obtained using (MBO+) with the Euler and spectral methods, and (MBO↓) with the Euler and spectral methods.

As shown in Figures 32a, 33a, 33b, 34b, 35a, 35b, and 36b, (MBO↓) using the Euler method produced superior Best cuts over (GW) for all choices of N. Figure 36a shows that for some choices of N, (MBO↓) using the Euler method produces marginally better best cuts over (GW). In Figure 32b and Figure 34a, we see that (GW) produced superior best cut sizes over (MBO↓) using the Euler method. Figures 32-36 show that (GW) still retained its superiority

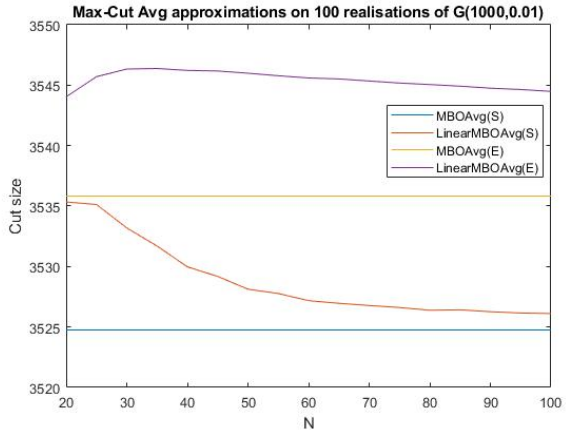
over (MBO+) using the Euler and spectral method with respect to best cuts obtained, and that the best cuts obtained using (GW) were superior to the best cuts obtained using (MBO↓) with the spectral method, for all graphs considered and for all choices of N, with the exception being the results of the WikiVote graph in Figure 36b, where (MBO↓) using the Euler method and spectral method, and (MBO+) using the Euler method and spectral method, are superior over (GW) with respect to best cut sizes produced.

Figures 32-36 show that for all choices of N, (MBO↓) using the Euler method produced superior results when compared against (MBO+) using the Euler method, with respect to best, average, and least cut sizes obtained. Figures 32-36 also show that (MBO↓) using the spectral method mostly obtains superior results over (MBO+) using the spectral method for most values of N.

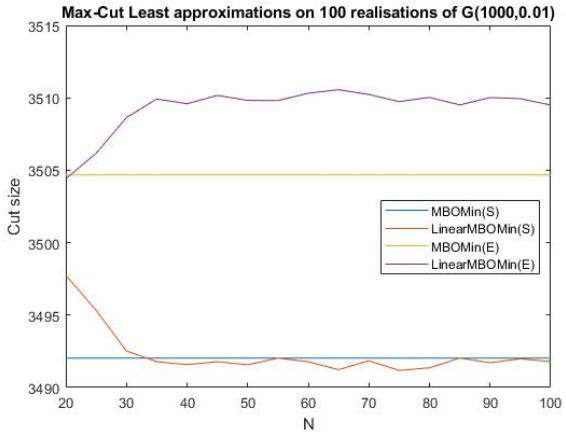
In Figure 37 and Figure 38 we show the computation time of our algorithms on the scale free graphs we tested on. We see that (MBO↓) using the Euler method for any choice of N has increased run times on all tests when compared to (MBO+) using the Euler method, (MBO↓) using the spectral method, and (MBO+) using the spectral method. The run times of (MBO↓) using the spectral method are generally larger than the run times of (MBO+), for most choices of N. However, in some cases we see that for some values of N, (MBO↓) using the spectral method has lower run times than (MBO+) using the spectral method.



(a) 100 realisations of $G(1000, 0.01)$: Best Cut size as function of N.

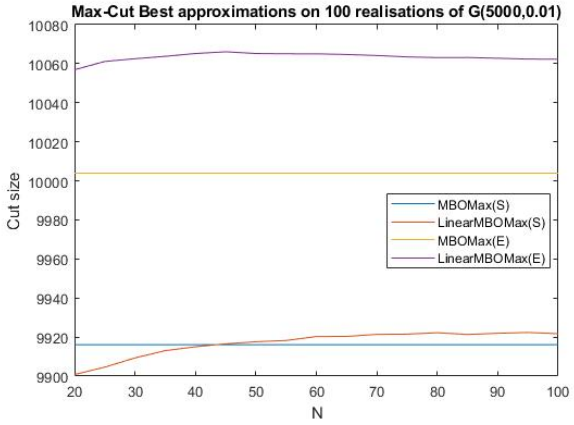


(b) 100 realisations of $G(1000, 0.01)$: Avg Cut size as function of N.

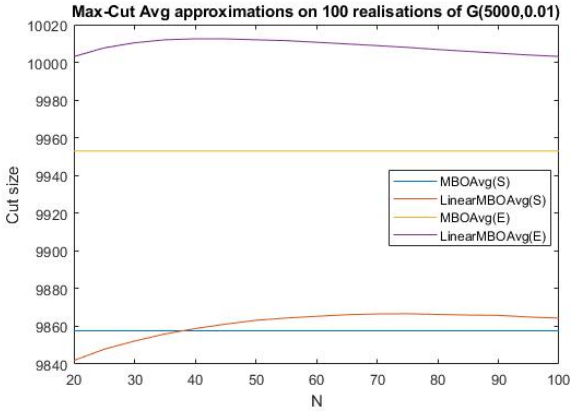


(c) 100 realisations of $G(1000, 0.01)$: Least Cut size as function of N.

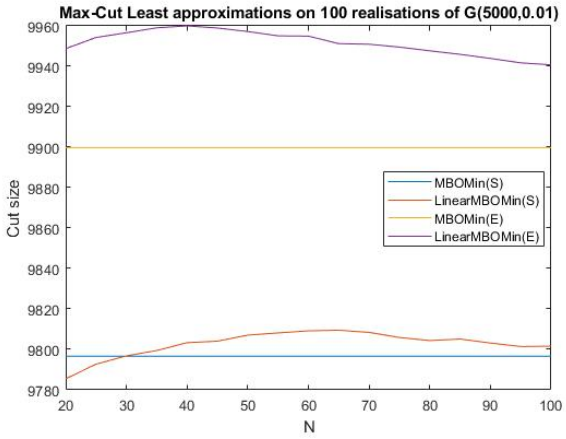
Figure 27.: Best, Avg, and Least Cut size as function of N for 100 realisations of $G(1000, 0.01)$.



(a) 100 realisations of $G(5000, 0.001)$: Best Cut size as function of N.

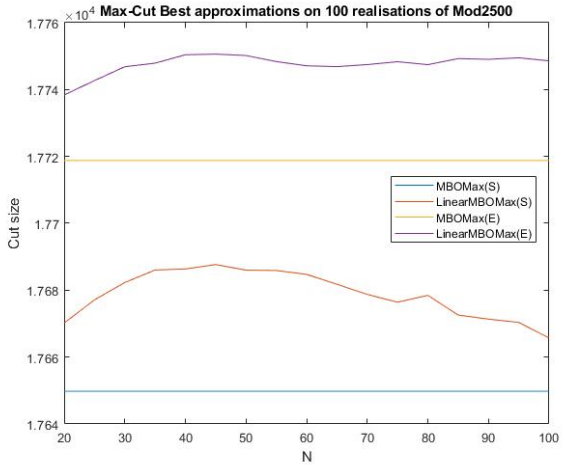


(b) 100 realisations of $G(5000, 0.001)$: Avg Cut size as function of N.

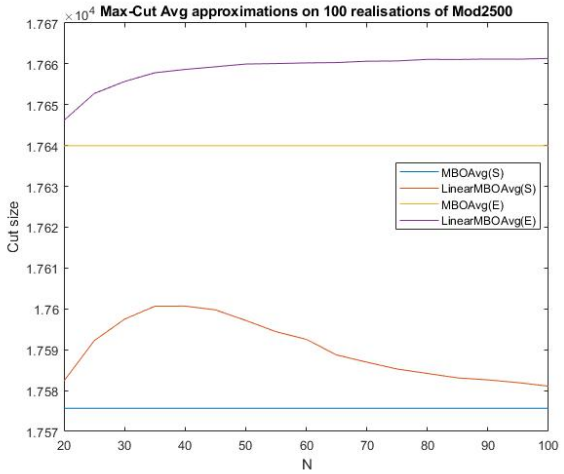


(c) 100 realisations of $G(5000, 0.001)$: Least Cut size as function of N.

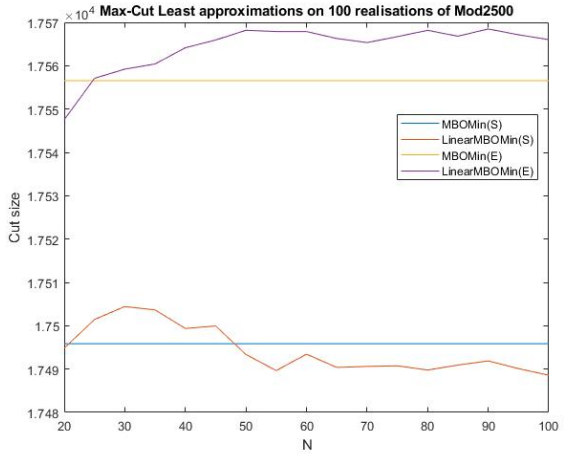
Figure 28.: Best, Avg, and Least Cut size as function of N for 100 realisations of $G(5000, 0.001)$.



(a) 100 realisations of $R(2500, 2, 0.009, 0.8)$: Best Cut size as function of N.

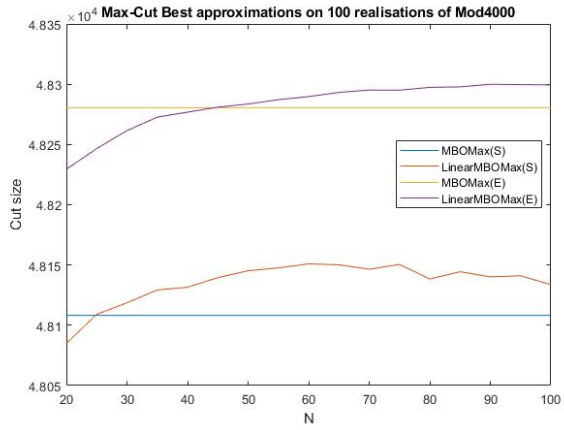


(b) 100 realisations of $R(2500, 2, 0.009, 0.8)$: Avg Cut size as function of N.

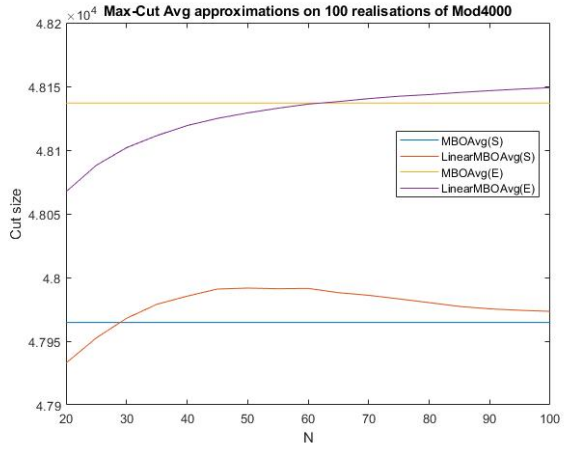


(c) 100 realisations of $R(2500, 2, 0.009, 0.8)$: Least Cut size as function of N.

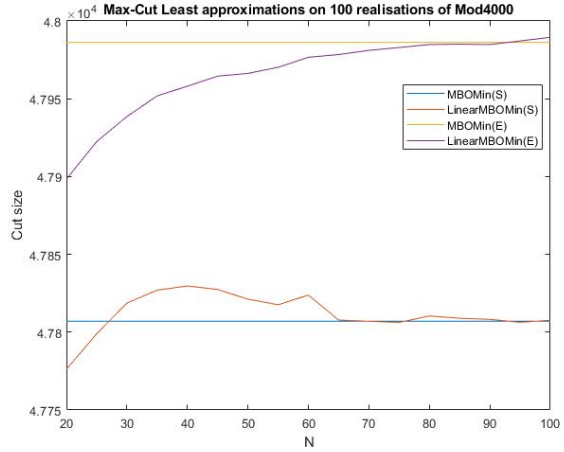
Figure 29.: Best, Avg, and Least Cut size as function of N for 100 realisations of $R(2500, 2, 0.009, 0.8)$.



(a) 100 realisations of $R(4000, 20, 0.01, 0.7)$: Best Cut size as function of N.

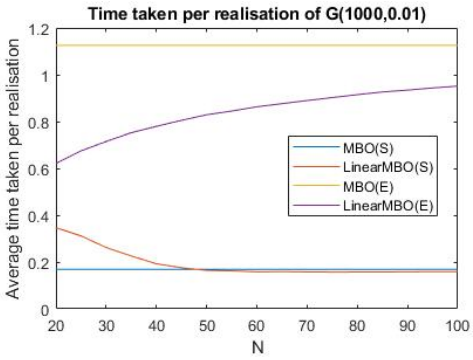


(b) 100 realisations of $R(4000, 20, 0.01, 0.7)$: Avg Cut size as function of N.

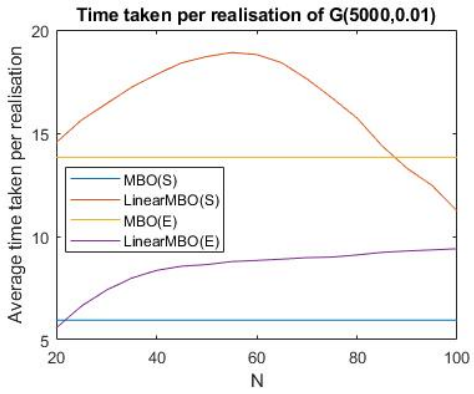


(c) 100 realisations of $R(4000, 20, 0.01, 0.7)$: Least Cut size as function of N.

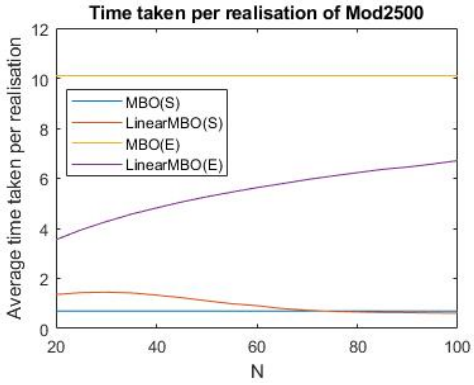
Figure 30.: Best, Avg, and Least Cut size as function of N for 100 realisations of $R(4000, 20, 0.01, 0.7)$.



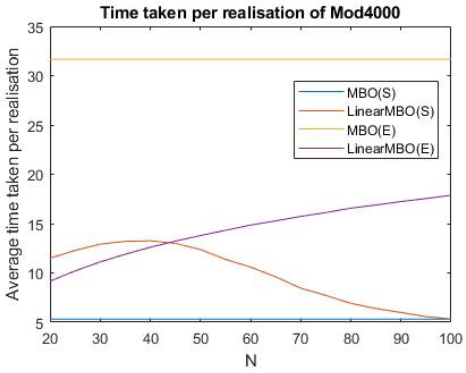
(a) Time taken per realisation of $G(1000, 0.01)$



(b) Time taken per realisation of $G(5000, 0.001)$

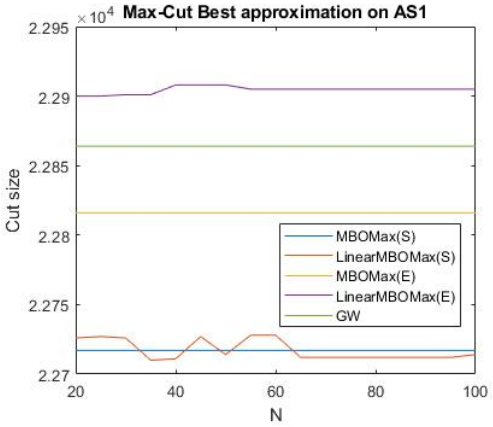


(c) Time taken per realisation of $R(2500, 2, 0.009, 0.8)$

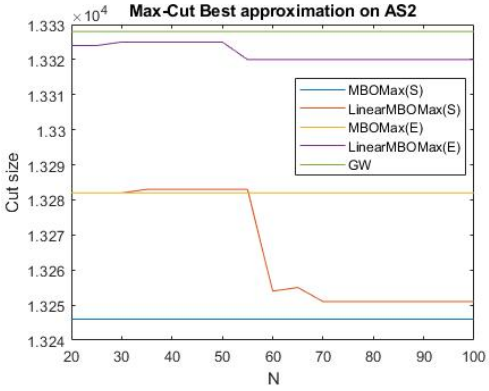


(d) Time taken per realisation of$R(4000, 20, 0.01, 0.7)$

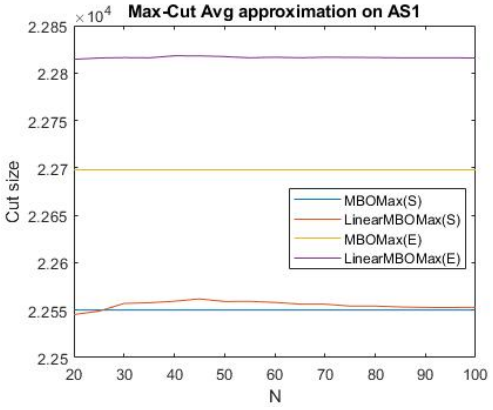
Figure 31.: Time taken per realisation of each random graph.



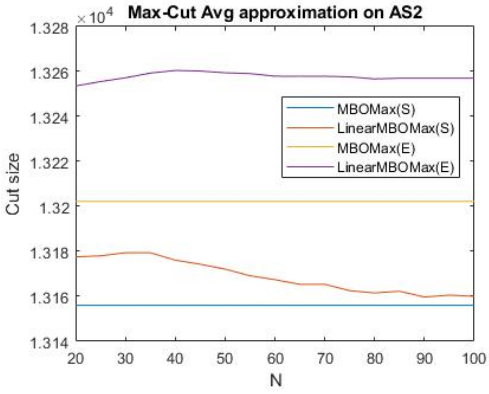
(a) Best cut obtained on the AS1 graph



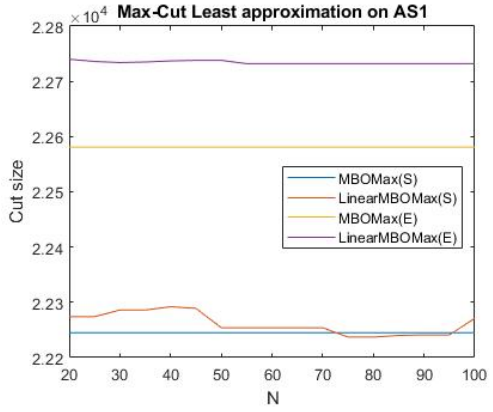
(b) Best cut obtained on the AS2 graph



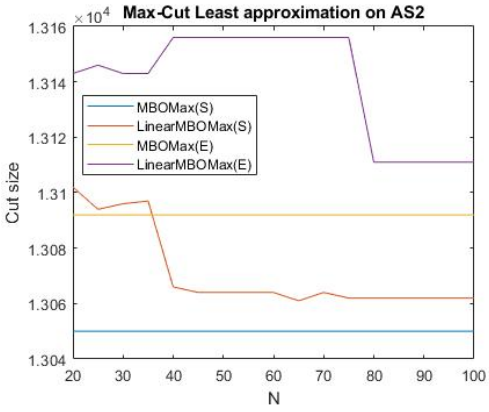
(c) Avg cut obtained on the AS1 graph



(d) Avg cut obtained on the AS2 graph

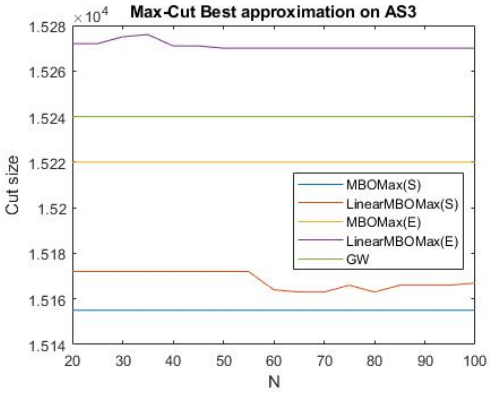


(e) Least cut obtained on the AS1 graph

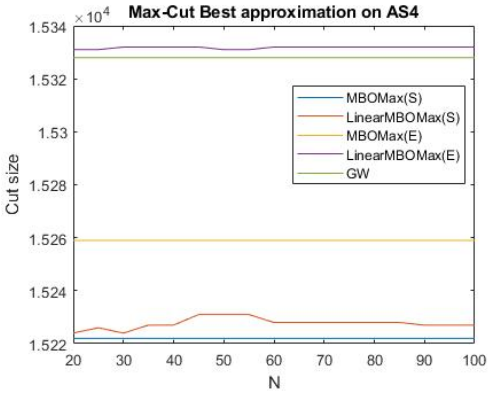


(f) Least cut obtained on the AS2 graph

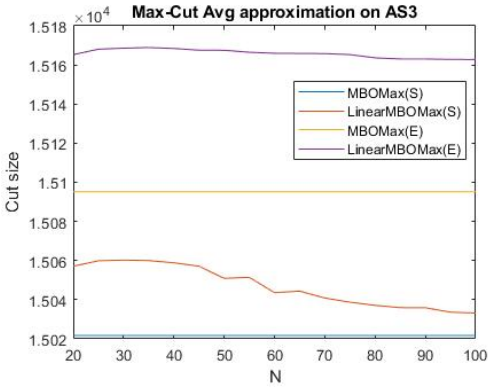
Figure 32.: Max-Cut approximation results obtained on the AS1 and AS2 graphs.



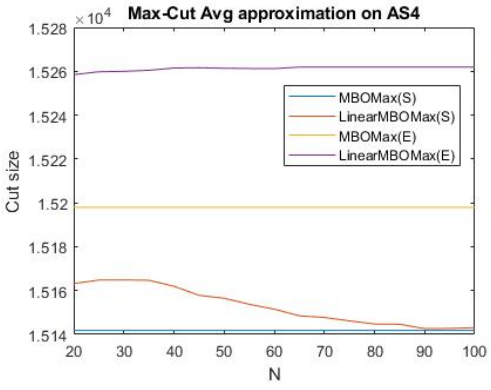
(a) Best cut obtained on the AS3 graph



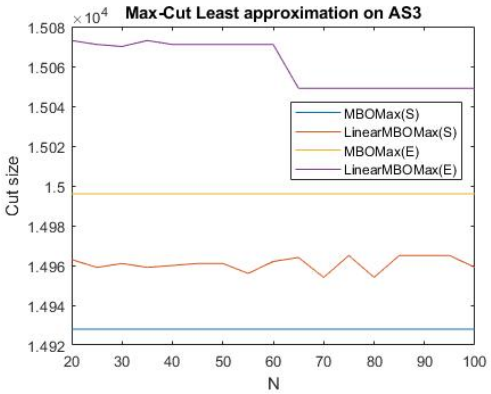
(b) Best cut obtained on the AS4 graph



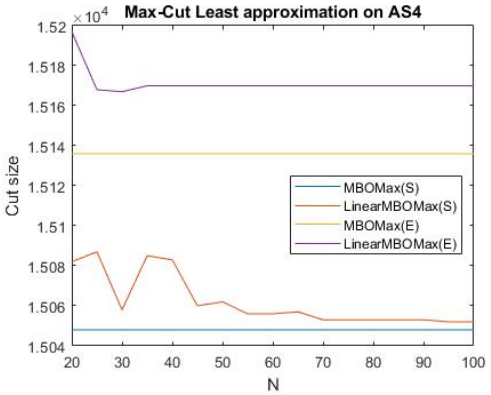
(c) Avg cut obtained on the AS3 graph



(d) Avg cut obtained on the AS4 graph

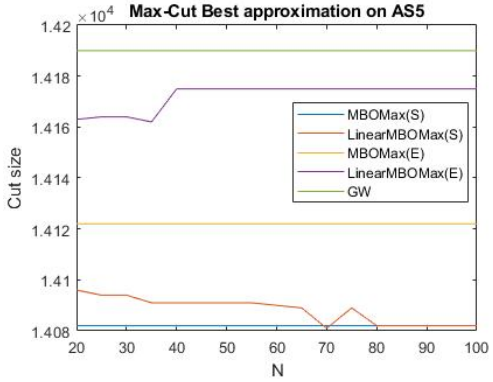


(e) Least cut obtained on the AS3 graph

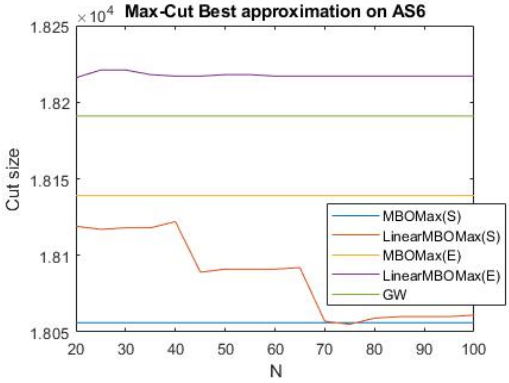


(f) Least cut obtained on the AS4 graph

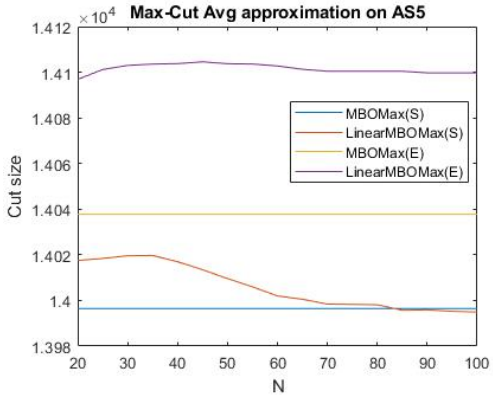
Figure 33.: Max-Cut approximation results obtained on the AS3 and AS4 graphs.



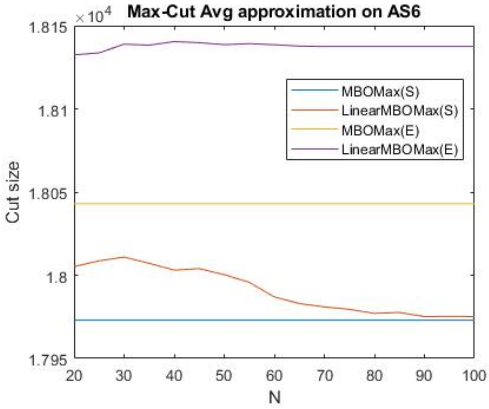
(a) Best cut obtained on the AS5 graph



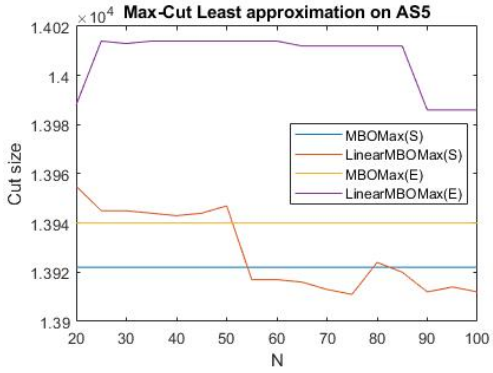
(b) Best cut obtained on the AS6 graph



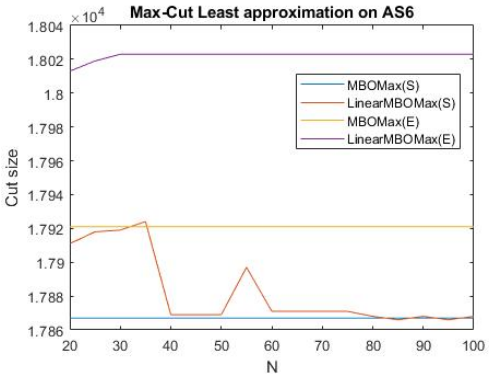
(c) Avg cut obtained on the AS5 graph



(d) Avg cut obtained on the AS6 graph

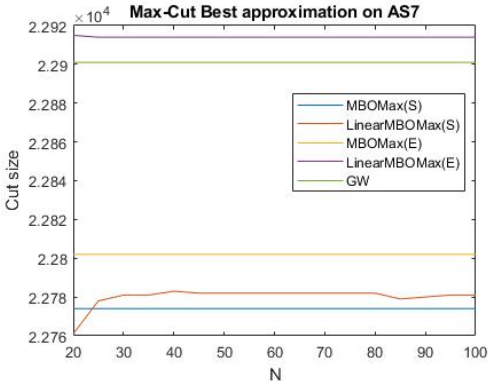


(e) Least cut obtained on the AS5 graph

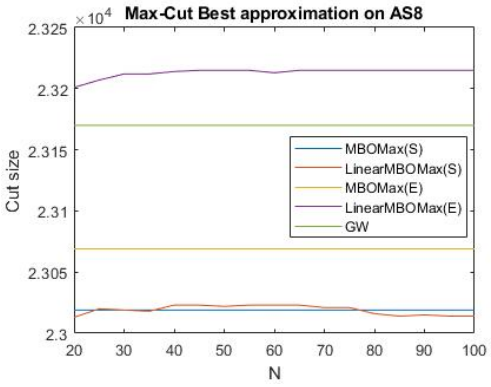


(f) Least cut obtained on the AS6 graph

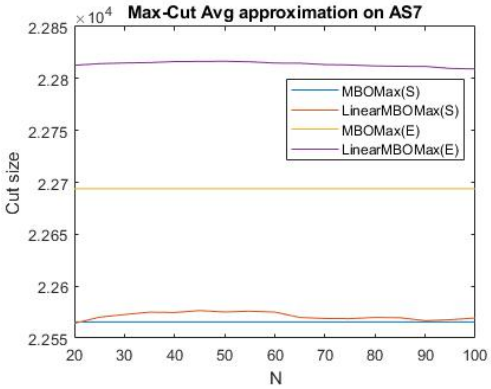
Figure 34.: Max-Cut approximation results obtained on the AS5 and AS6 graphs.



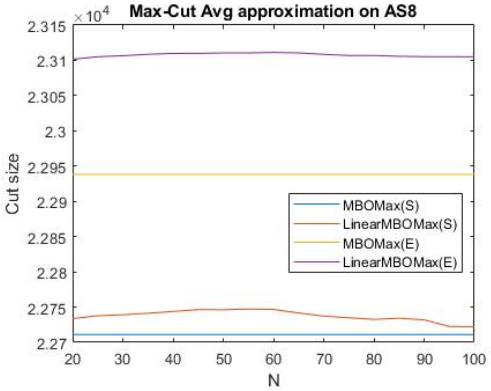
(a) Best cut obtained on the AS7 graph



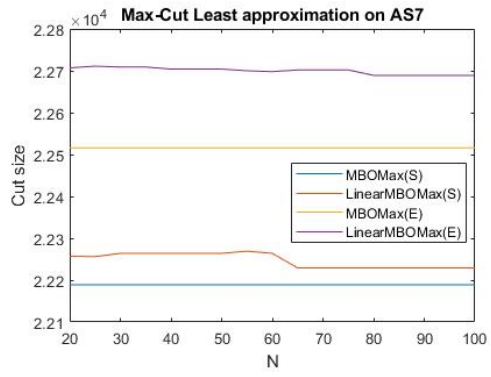
(b) Best cut obtained on the AS8 graph



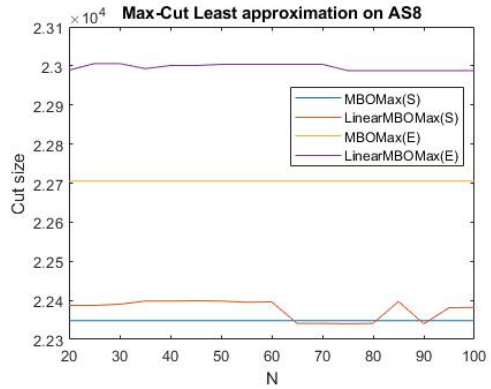
(c) Avg cut obtained on the AS7 graph



(d) Avg cut obtained on the AS8 graph

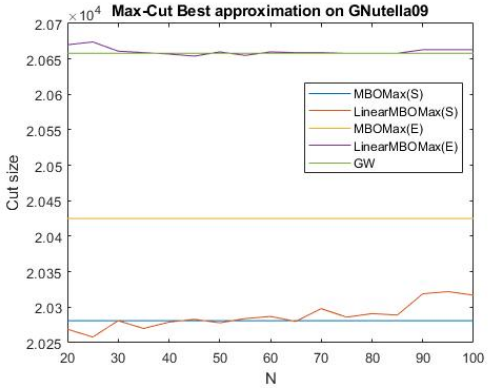


(e) Least cut obtained on the AS7 graph

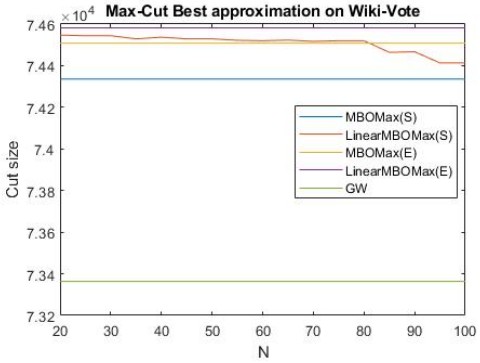


(f) Least cut obtained on the AS8 graph

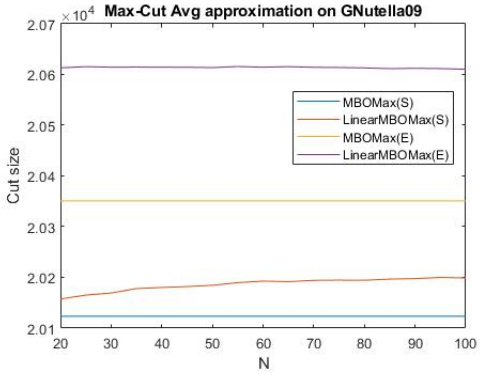
Figure 35.: Max-Cut approximation results obtained on the AS7 and AS8 graphs.



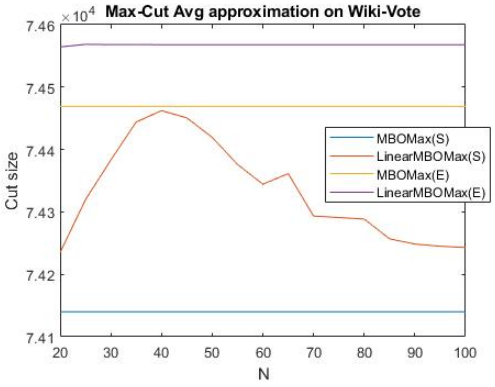
(a) Best cut obtained on the GNutella graph



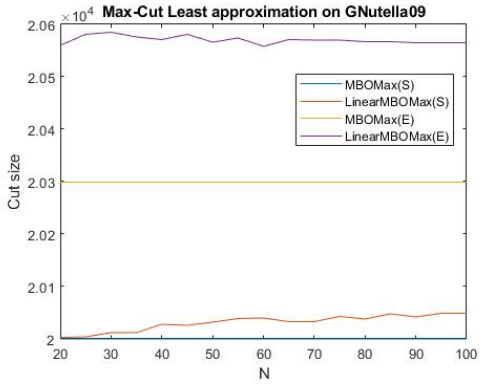
(b) Best cut obtained on the WikiVote graph



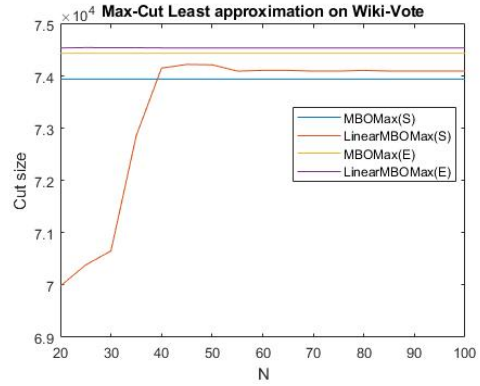
(c) Avg cut obtained on the GNutella graph



(d) Avg cut obtained on the WikiVote graph

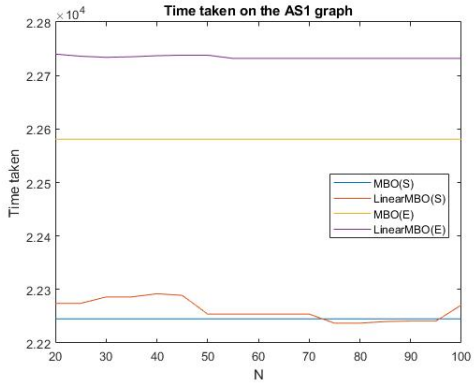


(e) Least cut obtained on the GNutella graph

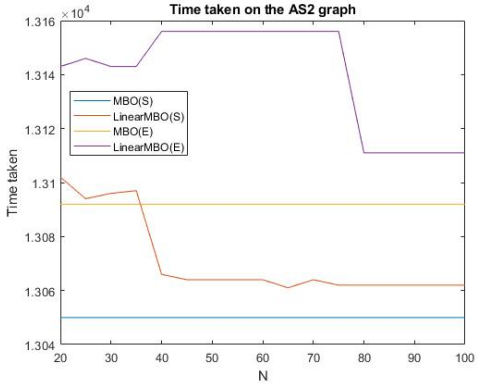


(f) Least cut obtained on the WikiVote graph

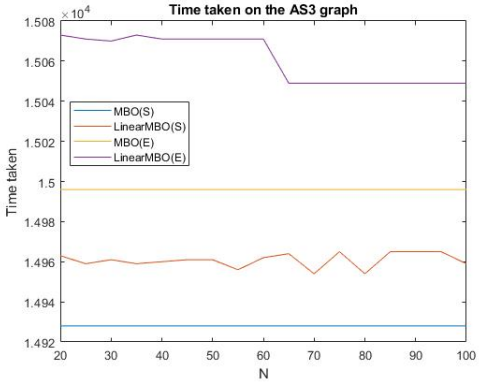
Figure 36.: Max-Cut approximation results obtained on the GNutella and WikiVote graphs.



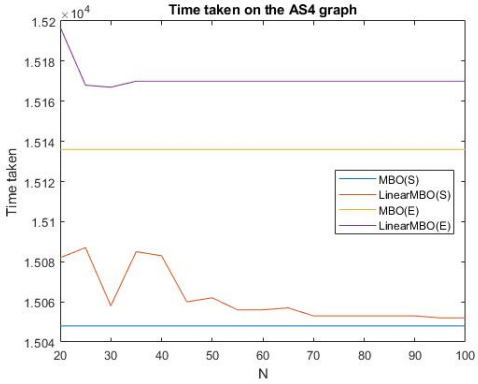
(a) Time taken on the AS1 graph



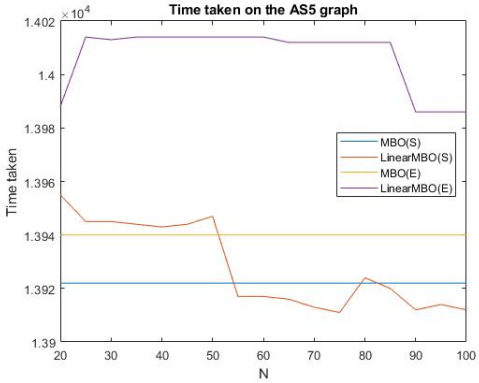
(b) Time taken on the AS2 graph



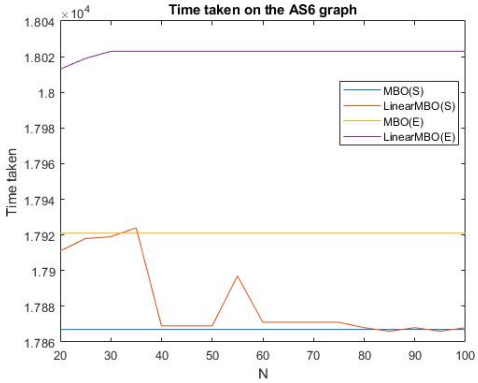
(c) Time taken on the AS3 graph



(d) Time taken on the AS4 graph

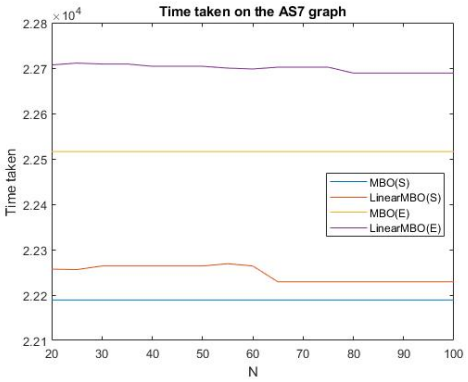


(e) Time taken on the AS5 graph

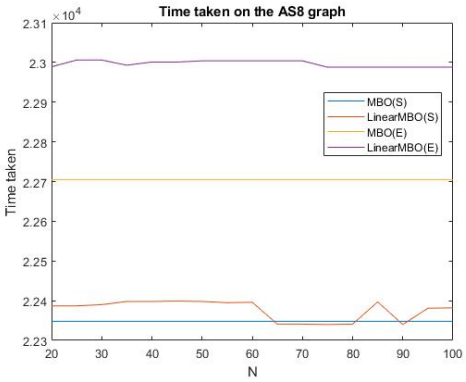


(f) Time taken on the AS6 graph

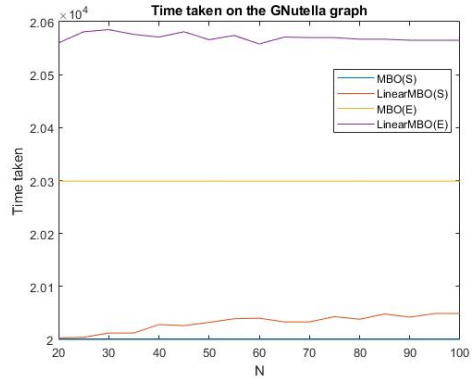
Figure 37.: Time taken to run (KMBO+) and (MBO↓) on scale free graphs, compared against N.



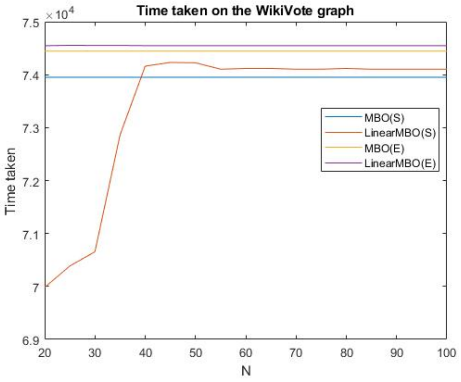
(a) Time taken on the AS7 graph



(b) Time taken on the AS8 graph



(c) Time taken on the GNutella graph



(d) Time taken on the WikiVote graph

Figure 38.: Time taken to run (KMBO+) and (MBO↓) on scale free graphs, compared against N.

8

COMPUTATIONAL RESULTS - MAX-K-CUT

8.1 METHOD

In Chapter 8, we test the results of (KCut) and (KMBO+). We compare the results obtained by (KCut) and (KMBO+) to the results obtained by (SDPCut) in Chapter 8.2.1. In Chapters 8.2.2-8.2.3 we compare (KCut) and (KMBO+) to choosing K-cuts chosen uniformly at random. This is due to (SDPCut) running out of memory on graphs larger than the graphs tested on in Chapter 8.2.1.

We run all our tests on a Windows 7 PC with 16GB RAM and an Intel(R) Core(TM) i5-4590 CPU with clock speed 3.30GHz. For (KCut), (KMBO+), and (SDPCut) we use MATLAB, which is ideal for large sparse matrices.

As shown in Chapters 8.2.1-8.2.3, the run times when using (KMBO+) were significantly lower than the run times when using (KCut). For this reason, in Chapters 8.2.4-8.2.5, we choose (KMBO+) for our tests, due to (KMBO+) running with reasonable run times on larger graphs, and for larger values of K.

For all of our tests we apply spectral truncation on the choice of $\widetilde{\Delta}^{+}$. We choose $k^* = \lfloor \frac{n}{100} \rfloor$ or $k^* = 20$, whichever is largest. This reduces the computation time of (KCut) and (KMBO+) without sacrificing much accuracy in the cuts obtained from both algorithms. We further analyse this parameter choice in Chapter 8.3.1. We compute the (KCut) and (KMBO+) evolutions of 50 initial conditions chosen at random from Ω_K. In the tables and figures

which we refer to in Chapter 8, we state the greatest (Best), average (Avg), and smallest (Least) sizes of cuts obtained by these 50 runs of (KCut) and (KMBO+). We run both (KCut) and (KMBO+) using $\widetilde{\Delta}_1^+$ and $\widetilde{\Delta}_s^+$, fixing the initial conditions for each algorithm, and each operator, and compare the results. For all tests in Chapter 8, we choose to not test with $\widetilde{\Delta}_0^+$ as our choice of signless Laplacian. This is due to the inferior results in testing (MBO+) using Δ_0^+ as the choice of signless Laplacian, in comparison to choosing Δ_1^+ or Δ_s^+, as shown in Chapter 7.1.

In all tests involving either (KCut) or (KMBO+), we set $\tau = 20$ and $\eta = 10^{-4}$. For (KCut), we choose $\varepsilon = 0.01$, and $C = 100$ for all tests.

In the tests comparing the results of (KCut) and (KMBO+) to the results of (SDPCut) in Chapter 8.2.1, we compute the relaxation step of (SDPCut) using SDPT3 MATLAB software [84]. After the relaxation step, we perform the random vector step 50 times, uniformly at random choosing $z_1, \ldots, z_K \in S^n$ each time. After the partitioning step for the K randomly chosen vectors, this results to a K-cut. In the figures referred to in Chapter 8.2.1, we list the greatest (Best), average (Avg), and smallest (Least) sizes of these cuts. In our tables, we highlight the method that obtained the best run times (Time), (KCut) using $\widetilde{\Delta}_1^+$, (KCut) using $\widetilde{\Delta}_s^+$, (KMBO+) using $\widetilde{\Delta}_1^+$, (KMBO+) using $\widetilde{\Delta}_s^+$, or (SDPCut).

In the tests comparing the results of (KCut) and (KMBO+) to the results of choosing K-cuts at random in Chapters 8.2.2-8.2.3, we generate 50 random K-cuts, chosen uniformly at random from $\mathcal{X}^K$. We state the greatest (Best), average (Avg), and smallest (Least) sizes of these cuts. In the tests comparing (KMBO+) to choosing K-cuts at random in Chapters 8.2.4-8.2.5, we follow the same procedure in producing K-cuts at random.

For (KCut), (KMBO+), and (SDPCut), only the adjacency matrix is initially provided, therefore the reported run times cover all calculations from that starting point. Like for the tests using (MBO+) in Chapter 7.1, the run times of (KCut) and (KMBO+) include removing all isolated nodes, computing L_1 and L_s in order to calculate the K eigenpairs corresponding to the smallest

eigenvalues of L_1^+ and L_s^+, and calculating the 50 (KCut) or (KMBO+) evolutions. For tests involving $\widetilde{\Delta}_s^+$ the computation time also includes calculating L_1^+ in order to compute the size of the output cuts produced by the algorithm.

For all tests in Chapter 8.2 involving Erdos-Renyi and random modular graphs, we create 100 realisations. We display the results on these graphs with bar charts, which represent the mean of the best, average, and least cuts over all 100 realisations of the chosen random graph. The error bars are the corrected sample standard deviation of the results obtained over all 100 realisations.

In Figure 39 and Figure 40 we show the results of (KMBO+) on the Buckminster Fuller dome graph with 60 nodes and 90 edges. The node subsets are defined by the colours red, blue, and green. If an edge connects two nodes of differing colour, the edge is included in the cut. Figure 39 shows the initial condition for the (KMBO+) evolution. The cut induced by this initial condition contains 61 edges. Figure 40 shows the resulting cut produced by (KMBO+). As seen in 40, the result produced by (KMBO+) is a 3-partition of the graph, therefore the cut contains 90 edges, the maximal number. This result was obtained in 0.098 seconds.

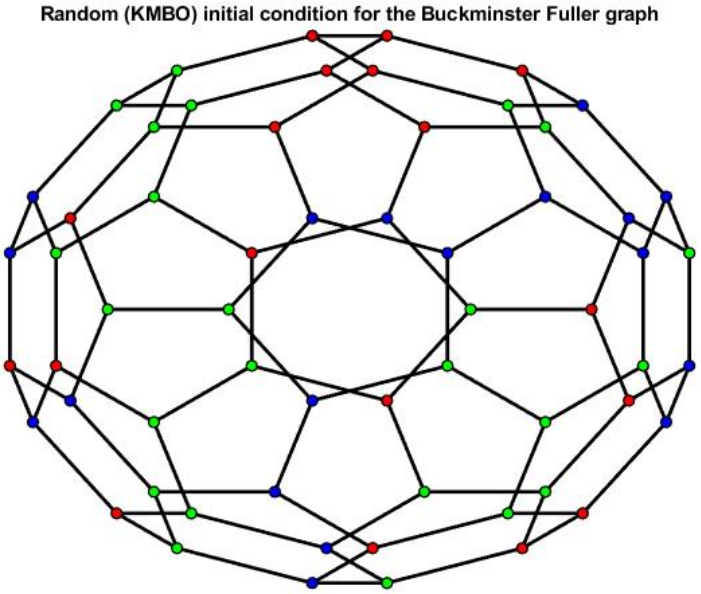


Figure 39.: The (KMBO+) initial condition on a Buckminster Fuller graph

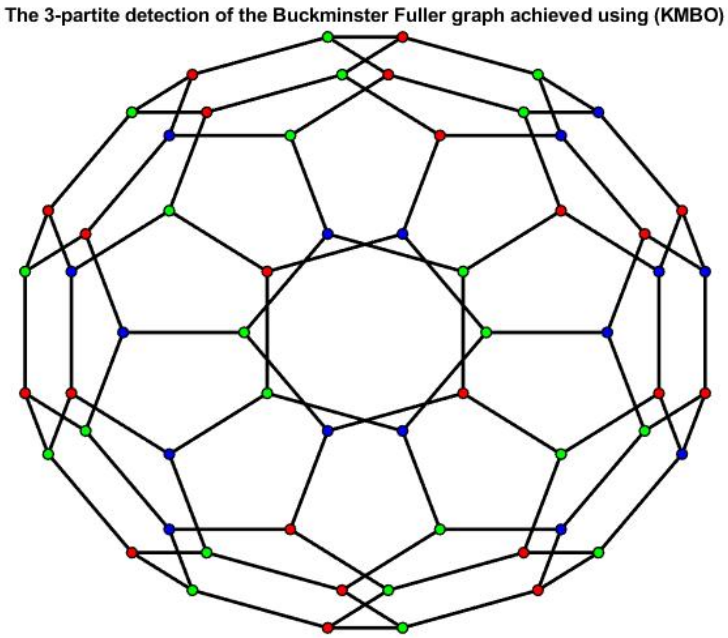


Figure 40.: The (KMBO+) outcome on a Buckminster Fuller graph

8.2 MAX-K-CUT RESULTS

8.2.1 *Tests comparing against (SDPCut)*

In Figures 41-52 we list results obtained on Erdos-Renyi and random modular graphs, comparing the results with the results obtained using (SDPCut).

For each of $G(50, 0.3)$ (Figure 41 for the Max-3-Cut problem, Figure 45 for the Max-4-Cut problem, and Figure 49 for the Max-5-Cut problem), $G(100, 0.1)$ (Figure 42 for the Max-3-Cut problem, Figure 46 for the Max-4-Cut problem, and Figure 50 for the Max-5-Cut problem), $R(100, 4, 0.1, 0.8)$ (Figure 43 for the Max-3-Cut problem, Figure 47 for the Max-4-Cut problem, and Figure 51 for the Max-5-Cut problem), and $R(100, 2, 0.4, 0.9)$ (Figure 44 for the Max-3-Cut problem, Figure 48 for the Max-4-Cut problem, and Figure 52 for the Max-5-Cut problem) we run (KCut), (KMBO+), and (SDPCut).

Figure 41 shows that (KMBO+) using either $\widetilde{\Delta}_1^+$ or $\widetilde{\Delta}_s^+$ as the choice of Laplacian produced similar sized Best 3-cuts to (SDPCut), marginally better average 3-cuts than (SDPCut), and marginally inferior least 3-cuts than (SDPCut). We also see that (KCut) produced superior least 3-cuts than both (KMBO+) and (SDPCut), for either $\widetilde{\Delta}_1^+$ or $\widetilde{\Delta}_s^+$ as the choice of Laplacian.

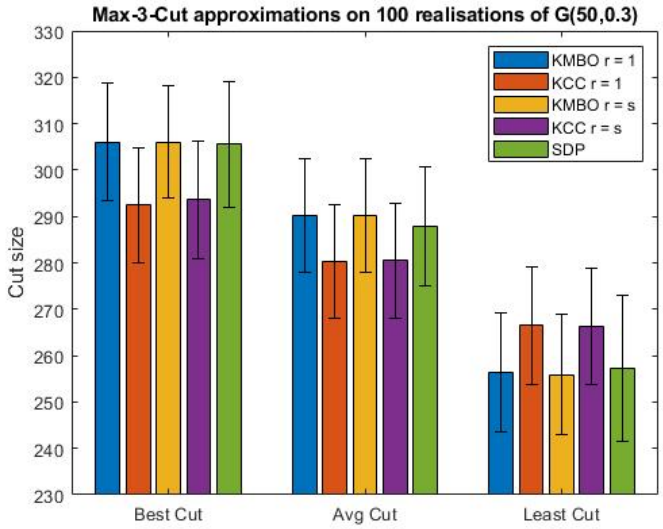


Figure 41.: Bar chart of 3-Cut approximations on 100 realisations of $G(50, 0.3)$.

Figure 42 shows that (KMBO+) using either $\widetilde{\Delta}_1^+$ or $\widetilde{\Delta}_s^+$ as the choice of Laplacian produced larger best and average 3-cuts than (SDPCut). We also see that (KCut) using either $\widetilde{\Delta}_1^+$ or $\widetilde{\Delta}_s^+$ as the choice of Laplacian produced marginally smaller best 3-cuts compared to (SDPCut), and superior average and least 3-cuts compared to (SDPCut).

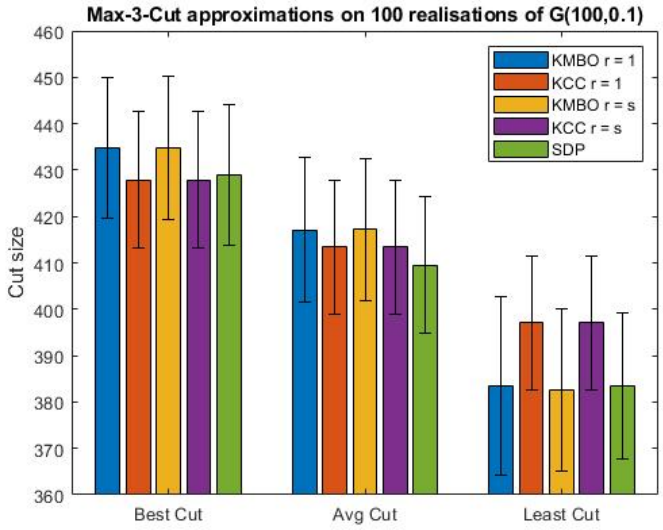


Figure 42.: Bar chart of 3-Cut approximations on 100 realisations of $G(100, 0.1)$.

Figure 43 and Figure 44 show that (KMBO+) using either $\widetilde{\Delta}_1^+$ or $\widetilde{\Delta}_s^+$ as the choice of Laplacian produced larger best, average, and least 3-cuts than (SDPCut). We also see that (KCut) using either $\widetilde{\Delta}_1^+$ or $\widetilde{\Delta}_s^+$ as the choice of Laplacian produced similar best 3-cuts compared to (SDPCut), and superior average and least 3-cuts compared to (SDPCut).

Graph ($K = 3$)	(KMBO+) Δ_1^+	(KCut) Δ_1^+	(KMBO+) Δ_s^+	(KCut) Δ_s^+	(SDPCut)
$G(50, 0.3)$	0.05	1.29	**0.03**	1.23	0.97
$G(100, 0.1)$	0.13	4.45	**0.08**	4.29	26.38
$R(100, 4, 0.1, 0.8)$	0.12	4.02	**0.07**	3.88	19.79
$R(100, 2, 0.4, 0.9)$	0.13	5.06	**0.08**	4.99	15.97

Table 19.: Average (KMBO+), (KCut), and (SDPCut) run-times for each realisation of $G(n, p)$ or $R(n, c, p, r)$, time in seconds.

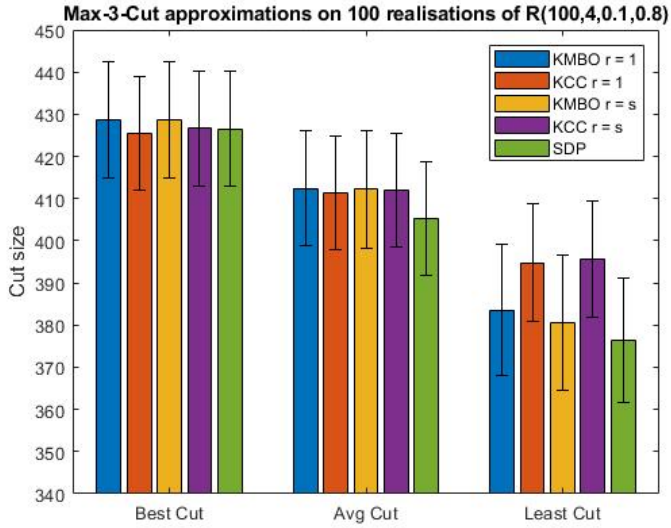


Figure 43.: Bar chart of 3-Cut approximations on 100 realisations of $R(100, 4, 0.1, 0.8)$.

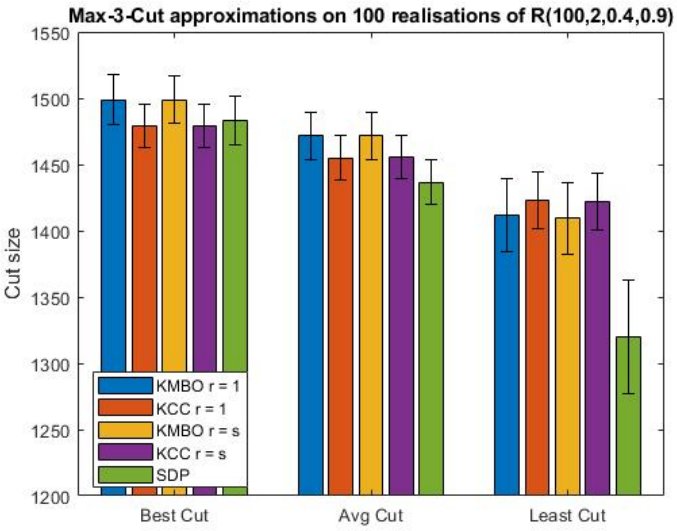


Figure 44.: Bar chart of 3-Cut approximations on 100 realisations of $R(100, 2, 0.4, 0.9)$.

Figure 45 shows that the best 4-cuts produced by (KMBO+) using both $\widetilde{\Delta}_1^+$ and $\widetilde{\Delta}_s^+$ were similar to the best 4-cuts produced by (SDPCut). However, the average 4-cuts produced by (SDPCut) were greater than both (KMBO+) and (KCut), for either choice of signless Laplacian. We also see that the least 4-

cuts produced by (KCut) for either choice of signless Laplacian were greater than the 4-cuts produced by (KMBO+), for either choice of signless Laplacian.

Figures 46-48 show that both (KMBO+) and (KCut) using either $\widetilde{\Delta}_1^+$ or $\widetilde{\Delta}_s^+$ as the choice of Laplacian outperformed (SDPCut) with respect to best, average, and least 4-cuts produced.

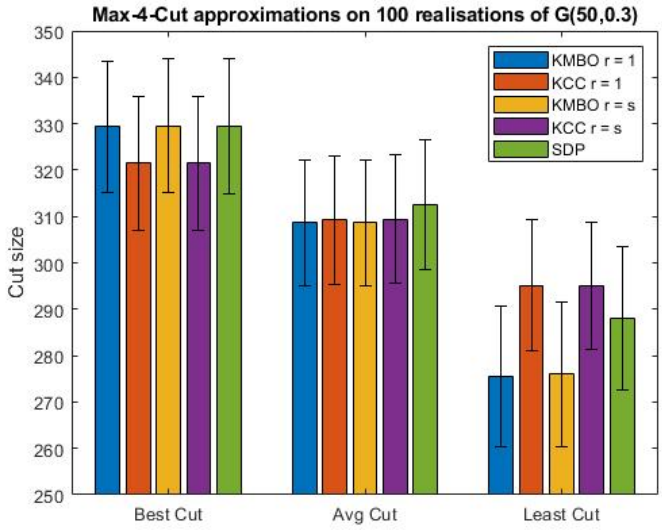


Figure 45.: Bar chart of 4-Cut approximations on 100 realisations of $G(50, 0.3)$.

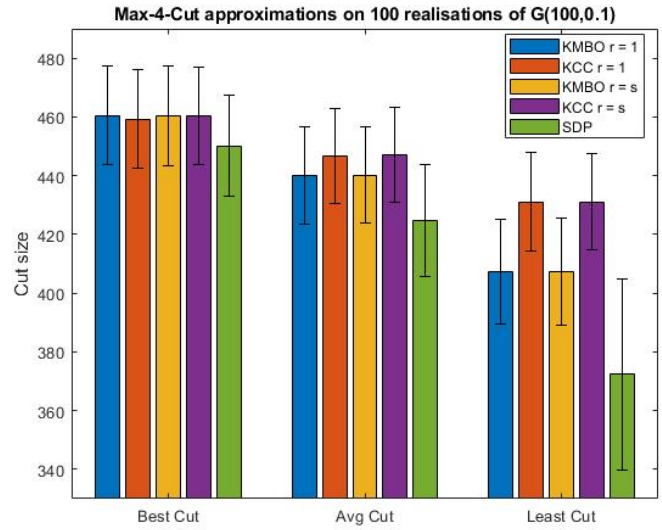


Figure 46.: Bar chart of 4-Cut approximations on 100 realisations of $G(100, 0.1)$.

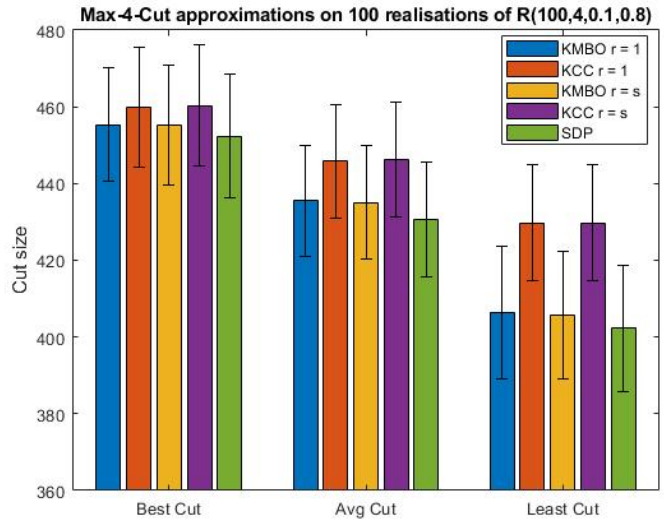


Figure 47.: Bar chart of 4-Cut approximations on 100 realisations of $R(100, 4, 0.1, 0.8)$.

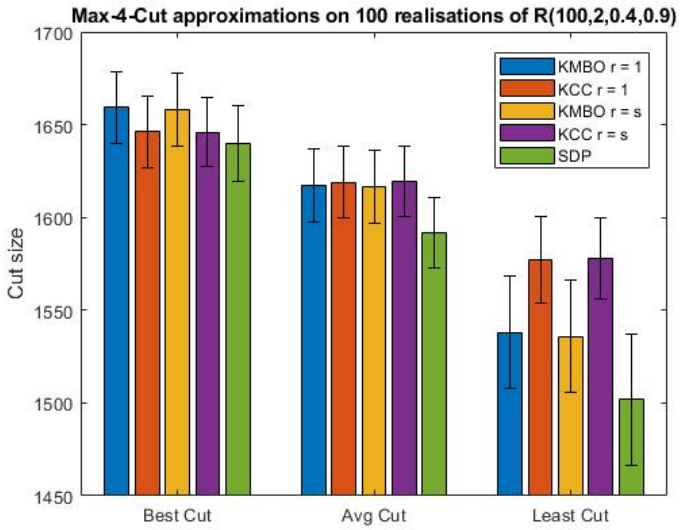


Figure 48.: Bar chart of 4-Cut approximations on 100 realisations of $R(100, 2, 0.4, 0.9)$.

Figure 49 shows that (SDPCut) produced larger best, average, and least 5-cuts than (KMBO+) for either choice of signless Laplacian, larger best 5-cuts than (KCut) for either choice of signless Laplacian, and similar average 5-cuts compared to (KCut) for either choice of signless Laplacian.

Graph ($K = 4$)	(KMBO+) Δ_1^+	(KCut) Δ_1^+	(KMBO+) Δ_s^+	(KCut) Δ_s^+	(SDPCut)
$G(50, 0.3)$	0.07	2.45	**0.04**	2.32	1.07
$G(100, 0.1)$	0.15	6.96	**0.09**	6.60	28.68
$R(100, 4, 0.1, 0.8)$	0.14	6.25	**0.08**	6.04	25.25
$R(100, 2, 0.4, 0.9)$	0.14	8.33	**0.09**	8.33	16.98

Table 20.: Average (KMBO+), (KCut), and (SDPCut) run-times for each realisation of $G(n, p)$ or $R(n, c, p, r)$, time in seconds.

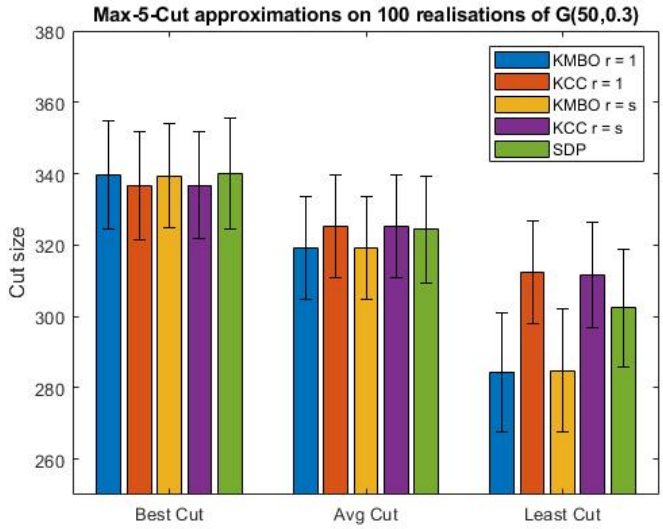


Figure 49.: Bar chart of 5-Cut approximations on 100 realisations of $G(50, 0.3)$.

Figures 50-52 show that both (KMBO+) and (KCut) for either choice of signless Laplacian produces larger best, average, and least 5-cuts compared to the 5-cuts produced by (SDPCut). In Figure 50 and Figure 51 we see that the best, average, and least 5-cuts produced by (KCut) are larger than the 5-cuts produced by (KMBO+), for either choice of signless Laplacian. Figure 52 shows that (KMBO+) produces larger best 5-cuts compared to (KCut), and inferior average and least 5-cuts compared to (KCut), for either choice of signless Laplacian.

Figures 41-52 show that in majority of these tests, (KMBO+) produced superior best and average 3-cuts. 4-cuts, and 5-cuts, compared to (KCut). In all tests, (KMBO+) produced inferior least 3-cuts, 4-cuts, and 5-cuts compared to (KCut). We also see that (KMBO+) using $\widetilde{\Delta}_1^+$ produces marginally better re-

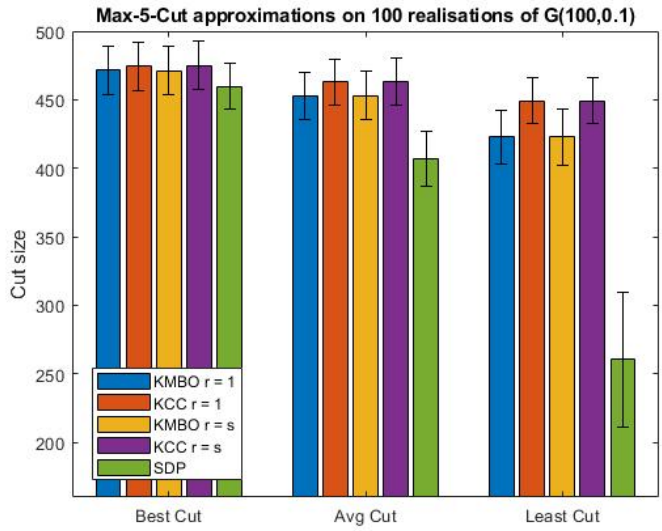


Figure 50.: Bar chart of 5-Cut approximations on 100 realisations of $G(100, 0.1)$.

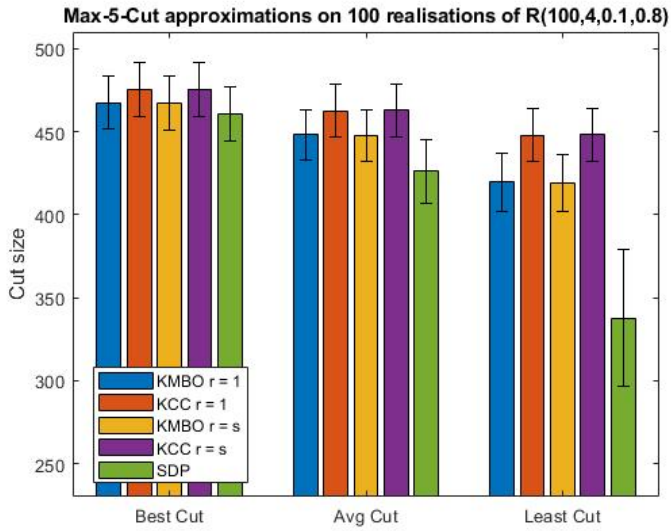


Figure 51.: Bar chart of 5-Cut approximations on 100 realisations of $R(100, 4, 0.1, 0.8)$.

sults than choosing $\widetilde{\Delta}_s^+$, and (KCut) produces using $\widetilde{\Delta}_s^+$ produces marginally better results than choosing $\widetilde{\Delta}_1^+$ in most tests.

Tables 19-21 show that for all tests, (KMBO+) using $\widetilde{\Delta}_s^+$ produced superior run times over (KMBO+) using $\widetilde{\Delta}_1^+$, (KCut) using either signless Laplacian, and (SDPCut). For all tests on 100 realisations of $G(100, 0.1)$, $R(100, 4, 0.1, 0.8)$,

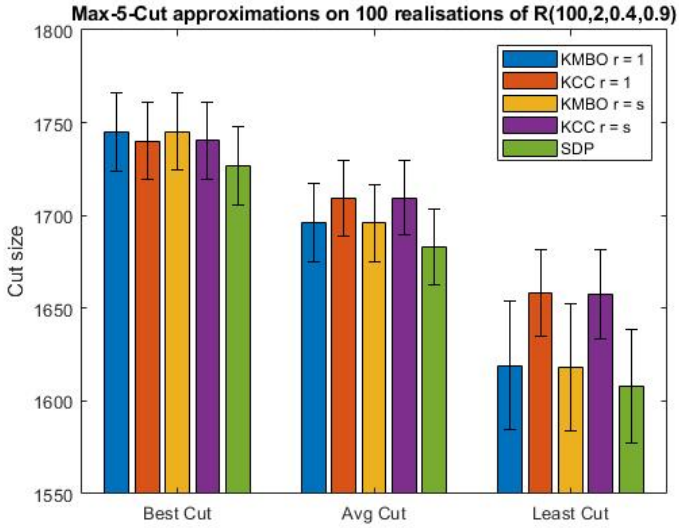


Figure 52.: Bar chart of 5-Cut approximations on 100 realisations of $R(100, 2, 0.4, 0.9)$.

Graph ($K = 5$)	(KMBO+) Δ_1^+	(KCut) Δ_1^+	(KMBO+) Δ_s^+	(KCut) Δ_s^+	(SDPCut)
$G(50, 0.3)$	0.07	2.64	**0.04**	2.45	1.11
$G(100, 0.1)$	0.16	7.47	**0.09**	7.11	13.31
$R(100, 4, 0.1, 0.8)$	0.16	7.56	**0.09**	7.30	19.21
$R(100, 2, 0.4, 0.9)$	0.16	9.38	**0.10**	9.29	17.86

Table 21.: Average (KMBO+), (KCut), and (SDPCut) run-times for each realisation of $G(n, p)$ or $R(n, c, p, r)$, time in seconds.

and $R(100, 2, 0.4, 0.9)$, we see that both (KMBO+) and (KCut) for either choice of signless Laplacian produces superior run times in comparison to (SDP-Cut). For all tests on 100 realisations of $G(50, 0.3)$, (KMBO+) for either choice of signless Laplacian had superior run times when compared to (SDPCut). However, (SDPCut) had superior times over (KCut) for both choices of signless Laplacian.

8.2.2 *Erdos-Renyi graphs*

In Figures 53-58 we list results obtained on Erdos-Renyi graphs using (KCut) and (KMBO+).

For each of $G(500, 0.05)$ (Figure 53 for the Max-3-Cut problem, Figure 55 for the Max-4-Cut problem, and Figure 57 for the Max-5-Cut problem) and $G(1000, 0.2)$ (Figure 54 for the Max-3-Cut problem, Figure 56 for the Max-4-Cut problem, and Figure 58 we run (KCut) and (KMBO+).

Figures 53, 55, and 57 show that both (KMBO+) and (KCut) using either $\widetilde{\Delta}_1^+$ or $\widetilde{\Delta}_s^+$ as the choice of Laplacian outperforms choosing 3-cuts, 4-cuts, and 5-cuts uniformly at random with respect to the best, average, and least cuts produced. We see that the best, average, and least 3-cuts produced by (KCut) and (KMBO+) using both Laplacians are 19-20% larger than choosing 3-cuts at random, their best, average, and least 4-cuts are at least 14-16% larger than choosing 4-cuts at random, and their best, average, and least 5-cuts are at least 11-13% larger than choosing 5-cuts at random.

Figures 54, 56, and 58 show that both (KMBO+) and (KCut) using either $\widetilde{\Delta}_1^+$ or $\widetilde{\Delta}_s^+$ as the choice of Laplacian outperforms choosing 3-cuts, 4-cuts, and 5-cuts uniformly at random with respect to the best, average, and least cuts produced. We see that the best, average, and least 3-cuts produced by (KCut) and (KMBO+) using both Laplacians are 7% larger than choosing 3-cuts at random, their best, average, and least 4-cuts are 5-6% larger than choosing 4-cuts at random, and their best, average, and least 5-cuts are 3-5% larger than choosing 5-cuts at random.

In Tables 22-24 we see that choosing $\widetilde{\Delta}_s^+$ as the choice of signless Laplacian in (KMBO+) and (KCut) results in reduced run times over choosing $\widetilde{\Delta}_1^+$. We also see that (KMBO+) has significantly lower run times than (KCut).

Graph ($K = 3$)	(KMBO+) Δ_1^+	(KCut) Δ_1^+	(KMBO+) Δ_s^+	(KCut) Δ_s^+
$G(500, 0.05)$	1.54	65.40	**0.86**	64.58
$G(1000, 0.2)$	5.80	241.63	**3.20**	234.60

Table 22.: Average (KMBO+) and (KCut) run-times for each realisation of $G(n, p)$, time in seconds.

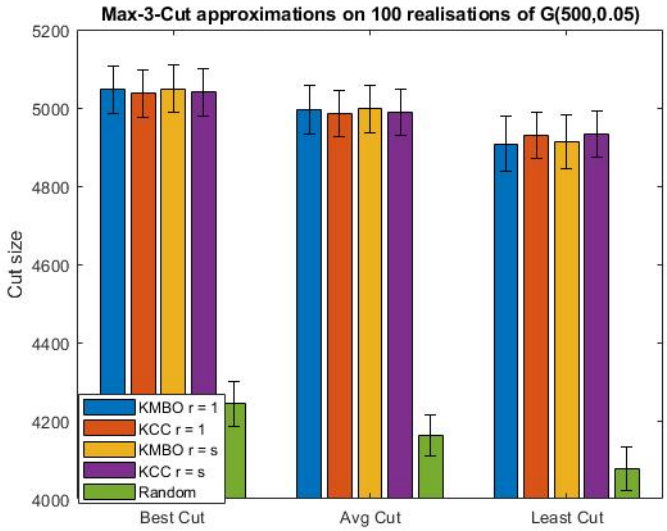


Figure 53.: Bar chart of 3-Cut approximations on 100 realisations of $G(500, 0.05)$.

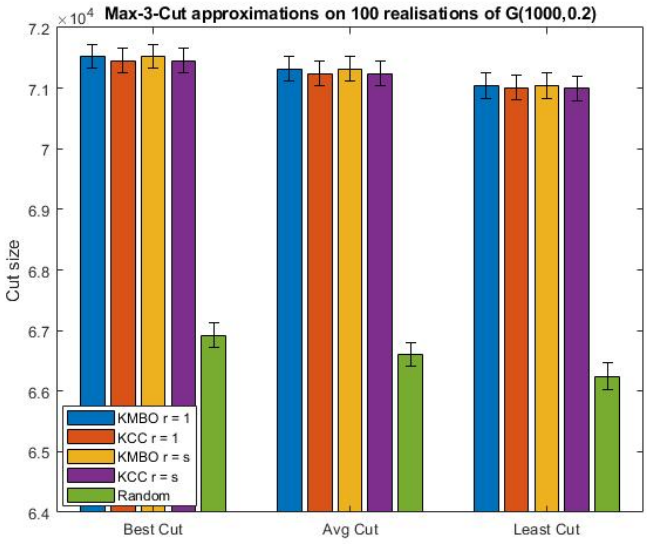


Figure 54.: Bar chart of 3-Cut approximations on 100 realisations of $G(1000, 0.2)$.

8.2.3 *Random modular graphs*

In Figures 59-64 we list results obtained on random modular graphs using (KCut) and (KMBO+).

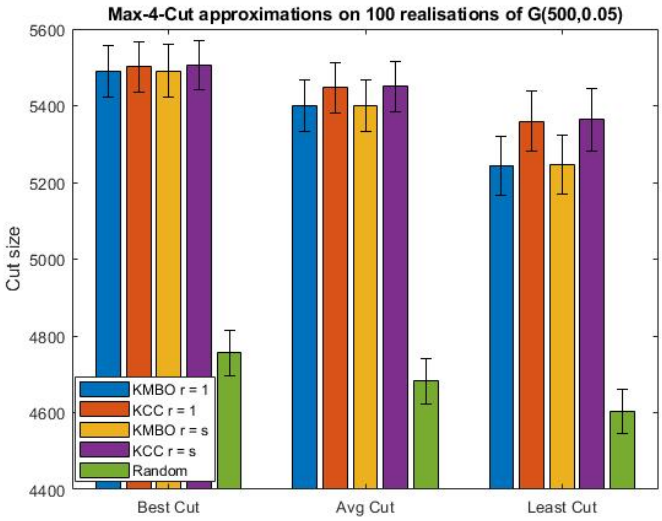


Figure 55.: Bar chart of 4-Cut approximations on 100 realisations of $G(500, 0.05)$.

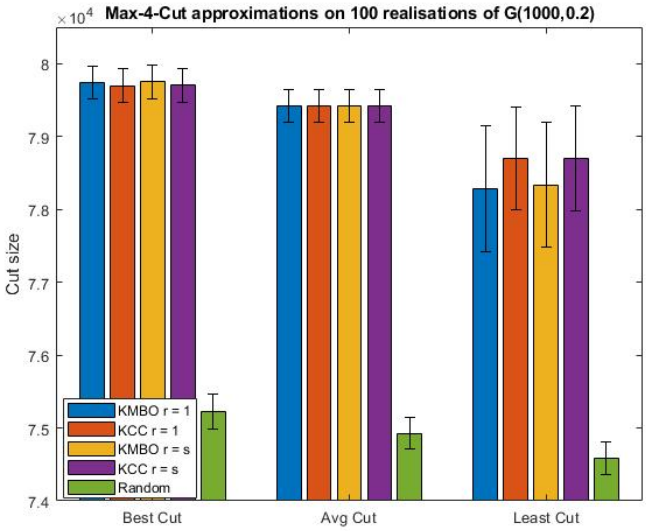


Figure 56.: Bar chart of 4-Cut approximations on 100 realisations of $G(1000, 0.2)$.

For each of $R(250, 4, 0.2, 0.9)$ (Figure 59 for the Max-3-Cut problem, Figure 61 for the Max-4-Cut problem, and Figure 63 for the Max-5-Cut problem) and $R(500, 2, 0.05, 0.8)$ (Figure 60 for the Max-3-Cut problem, Figure 62 for the Max-4-Cut problem, and Figure 64 for the Max-5-Cut problem) we run (KCut) and (KMBO+).

Graph ($K = 4$)	(KMBO+) Δ_1^+	(KCut) Δ_1^+	(KMBO+) Δ_s^+	(KCut) Δ_s^+
$G(500, 0.05)$	1.89	106.89	**1.00**	106.73
$G(1000, 0.2)$	7.13	407.83	**3.76**	402.28

Table 23.: Average (KMBO+) and (KCut) run-times for each realisation of $G(n, p)$, time in seconds.

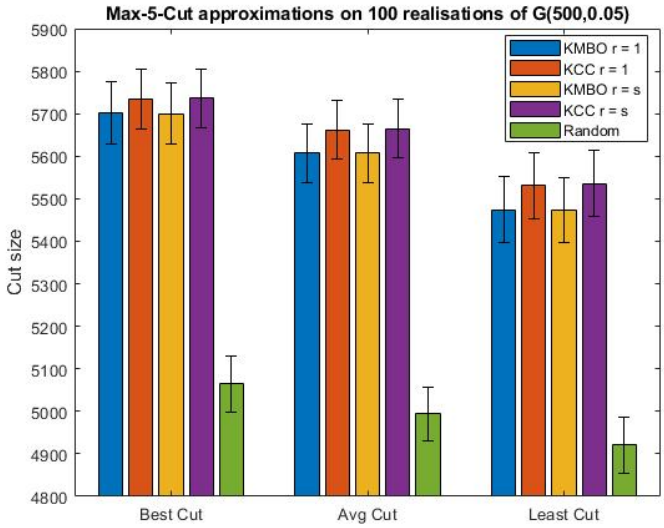


Figure 57.: Bar chart of 5-Cut approximations on 100 realisations of $G(500, 0.05)$.

Graph ($K = 5$)	(KMBO+) Δ_1^+	(KCut) Δ_1^+	(KMBO+) Δ_s^+	(KCut) Δ_s^+
$G(500, 0.05)$	2.25	137.64	**1.16**	136.12
$G(1000, 0.2)$	8.64	531.60	**4.47**	530.39

Table 24.: Average (KMBO+) and (KCut) run-times for each realisation of $G(n, p)$, time in seconds.

Figures 59, 61, and 63 show that both (KMBO+) and (KCut) using either $\widetilde{\Delta}_1^+$ or $\widetilde{\Delta}_s^+$ as the choice of Laplacian outperform choosing 3-cuts, 4-cuts, and 5-cuts uniformly at random with respect to the best, average, and least cuts produced. We see that the best, average, and least 3-cuts produced by (KCut) and (KMBO+) using both Laplacians are 8-10% larger than choosing 3-cuts at random, their 4-cuts are 6-8% larger than choosing 4-cuts at random, and their 5-cuts are 4-5% larger than choosing 5-cuts at random.

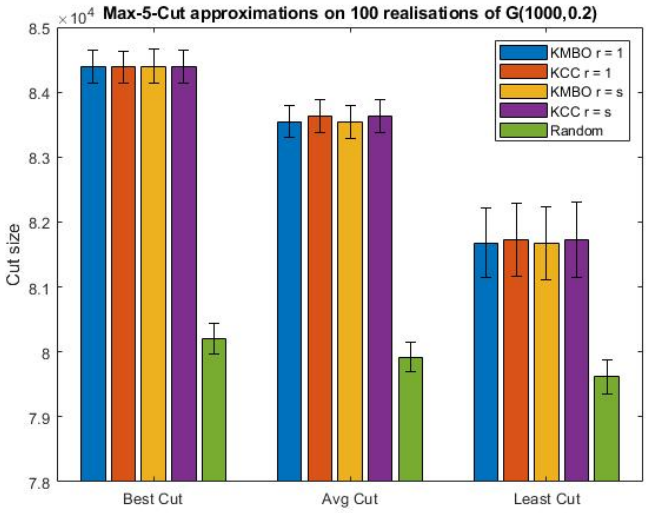


Figure 58.: Bar chart of 5-Cut approximations on 100 realisations of $G(1000, 0.2)$.

Graph $(K = 3)$	(KMBO+) Δ_1^+	(KCut) Δ_1^+	(KMBO+) Δ_s^+	(KCut) Δ_s^+
$R(250, 4, 0.2, 0.9)$	0.51	21.02	**0.30**	20.66
$R(500, 2, 0.05, 0.8)$	1.50	74.62	**0.81**	73.57

Table 25.: Average (KMBO+) and (KCut) run-times for each realisation of $R(n, c, p, r)$, time in seconds.

Figures 60, 62, and 64 show that both (KMBO+) and (KCut) using either $\widetilde{\Delta}_1^+$ or $\widetilde{\Delta}_s^+$ as the choice of Laplacian outperform choosing 3-cuts, 4-cuts, and 5-cuts uniformly at random with respect to the best, average, and least cuts produced. We see that the best, average, and least 3-cuts produced by (KCut) and (KMBO+) using both Laplacians are 19-20% larger than choosing 3-cuts at random, their 4-cuts are 14-16% larger than choosing 4-cuts at random, and their 5-cuts 11-13% larger than choosing 5-cuts at random.

In Tables 25-27 we see that choosing $\widetilde{\Delta}_s^+$ as the choice of signless Laplacian in (KMBO+) and (KCut) results in reduced run times over choosing $\widetilde{\Delta}_1^+$. We also see that (KMBO+) has significantly lower run times than (KCut).

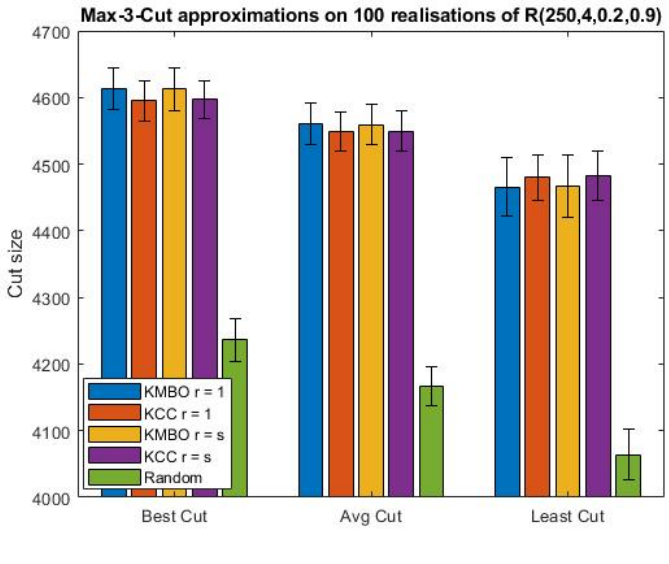


Figure 59.: Bar chart of 3-Cut approximations on 100 realisations of $R(250, 4, 0.2, 0.9)$.

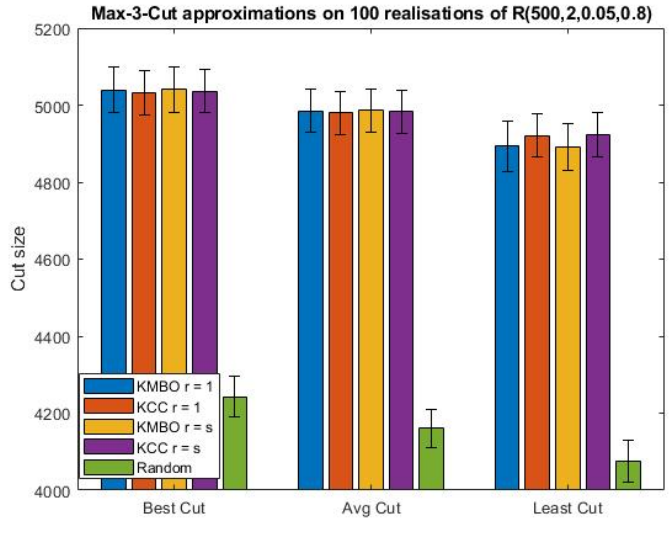


Figure 60.: Bar chart of 3-Cut approximations on 100 realisations of $R(500, 2, 0.05, 0.8)$.

Graph $(K = 4)$	(KMBO+) Δ_1^+	(KCut) Δ_1^+	(KMBO+) Δ_s^+	(KCut) Δ_s^+
$R(250, 4, 0.2, 0.9)$	0.58	34.26	**0.33**	34.08
$R(500, 2, 0.05, 0.8)$	1.88	101.37	**0.98**	100.17

Table 26.: Average (KMBO+) and (KCut) run-times for each realisation of $R(n, c, p, r)$, time in seconds.

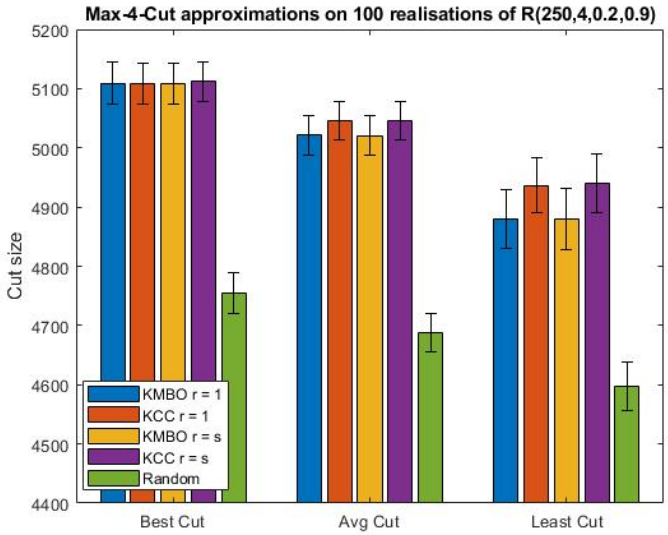


Figure 61.: Bar chart of 4-Cut approximations on 100 realisations of $R(250, 4, 0.2, 0.9)$.

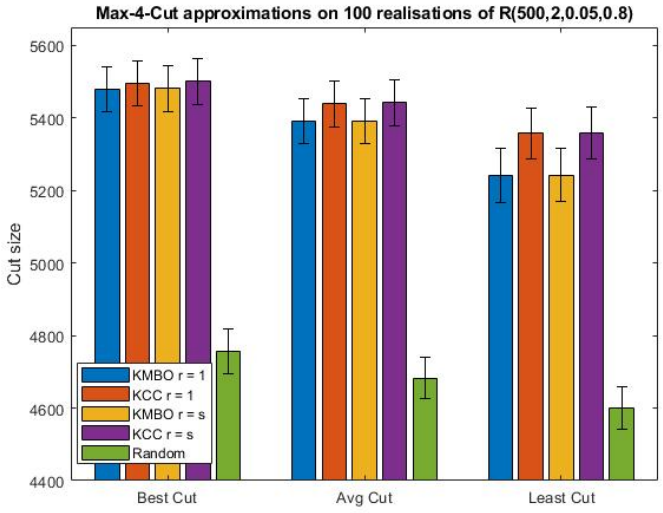


Figure 62.: Bar chart of 4-Cut approximations on 100 realisations of $R(500, 2, 0.05, 0.8)$.

Graph $(K = 5)$	(KMBO+) Δ_1^+	(KCut) Δ_1^+	(KMBO+) Δ_s^+	(KCut) Δ_s^+
$R(250, 4, 0.2, 0.9)$	0.67	41.97	**0.38**	41.68
$R(500, 2, 0.05, 0.8)$	2.22	130.71	**1.13**	130.48

Table 27.: Average (KMBO+) and (KCut) run-times for each realisation of $R(n, c, p, r)$, time in seconds.

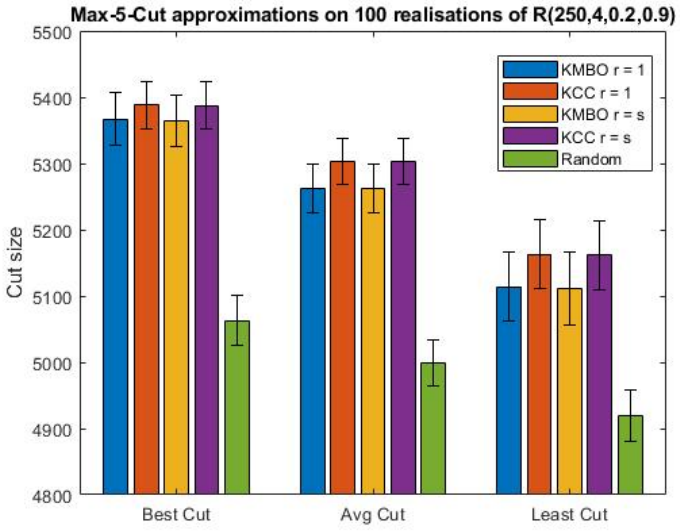


Figure 63.: Bar chart of 5-Cut approximations on 100 realisations of $R(250, 4, 0.2, 0.9)$.

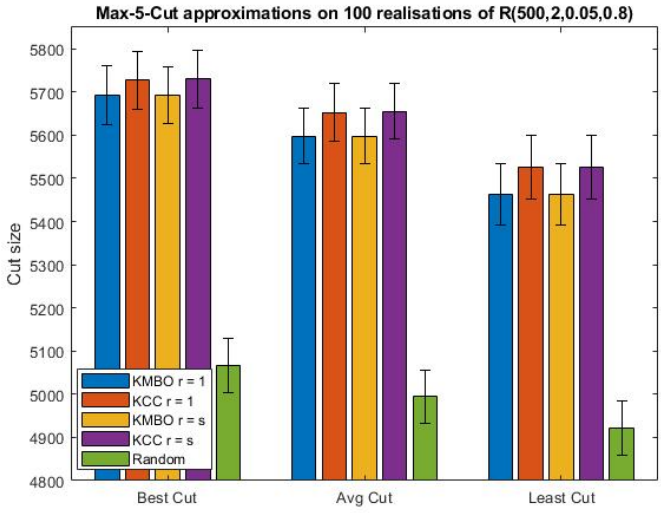


Figure 64.: Bar chart of 5-Cut approximations on 100 realisations of $R(500, 2, 0.05, 0.8)$.

8.2.4 *Scale free graphs*

As shown in all tests in Chapters 8.2.1-8.2.3, (KMBO+) produced superior run times over (KCut). In this sub-chapter we show that (KMBO+) can produce K-cut approximations on larger graphs with reasonable run times.

Tables 29-31 show the results of our test of (KMBO+) on the scale-free graphs tested on in Chapter 7.1.3. Table 29 shows the results where $K = 3$, Table 30 shows the results where $K = 4$, and Table 31 shows the results where $K = 5$. We compare the results against uniformly choosing 50 random K-cuts, comparing the cut sizes induced from the random K-cuts.

In Table 29, we see that for the ASi graphs, where $i \in \{1, \ldots, 8\}$, the best cuts, average cuts, and least cuts produced by (KMBO+) with $\widetilde{\Delta}_1^+$ as the choice of signless Laplacian outperforms 3-cuts chosen at random by 33-36%. On GNutella09 and WikiVote, the best cuts, average cuts, and least cuts produced by (KMBO+) with $\widetilde{\Delta}_1^+$ outperforms 3-cuts chosen at random by 31-32% and 11-20% respectively.

We note that on the WikiVote graph, the best cut obtained by (KMBO+) with $\widetilde{\Delta}_1^+$ is 9% larger than the least cut obtained. For ASi where $i \in \{1, \ldots, 8\}$, the best cuts obtained by (KMBO+) with $\widetilde{\Delta}_1^+$ are 2-4% larger than the least cut produced, and on the GNutella graph, the best cut is 2% greater than the least cut produced.

Graph	(3MBO+) $\widetilde{\Delta}_1^+$	(3MBO+) $\widetilde{\Delta}_s^+$	(4MBO+) $\widetilde{\Delta}_1^+$	(4MBO+) $\widetilde{\Delta}_s^+$	(5MBO+) $\widetilde{\Delta}_1^+$	(5MBO+) $\widetilde{\Delta}_s^+$
AS1	396.88	**74.37**	520.68	**182.26**	640.96	**148.51**
AS2	149.75	**19.09**	195.60	**36.11**	241.65	**25.10**
AS3	188.59	**34.56**	248.70	**41.24**	305.38	**62.48**
AS4	199.18	**35.99**	260.89	**105.91**	320.68	**82.26**
AS5	164.35	**7.78**	215.69	**10.45**	265.17	**11.38**
AS6	273.81	**59.24**	361.01	**126.76**	441.52	**118.36**
AS7	397.67	**96.13**	520.60	**157.56**	640.51	**124.26**
AS8	404.26	**156.14**	526.08	**228.46**	645.56	**150.84**
GNutella	176.04	**18.28**	227.06	**51.61**	278.66	**60.06**
WikiVote	183.29	**4.03**	181.15	**5.10**	253.66	**5.70**

Table 28.: (KMBO+) run-times for each scale free graph, time in seconds.

In Table 30, we see that for the GNutella graph and the ASi graphs, where $i \in \{1, \ldots, 8\}$, the best cuts, average cuts, and least cuts produced by (KMBO+) with $\widetilde{\Delta}_1^+$ outperforms 4-cuts chosen at random by 21-24%. On the WikiVote graph, we see that the best cuts and average cuts produced by (KMBO+) with $\widetilde{\Delta}_1^+$ outperform 4-cuts chosen at random by 11% and 6% respectively. However, the least 4-cut chosen at random outperformed the least cuts induced produced by (KMBO+) with $\widetilde{\Delta}_1^+$ by 0.2%.

In Table 31, we see that for the GNutella graph and the ASi graphs, where $i \in \{1, \ldots, 8\}$, the best cuts, average cuts, and least cuts produced by (KMBO+) with $\widetilde{\Delta}_1^+$ outperforms 5-cuts chosen at random by 14-18%. On the WikiVote graph, we see that the best cuts and average cuts produced by (KMBO+) with $\widetilde{\Delta}_1^+$ outperform 5-cuts chosen at random by 8% and 2% respectively. However, the least 5-cut chosen at random outperformed the least cuts induced produced by (KMBO+) with $\widetilde{\Delta}_1^+$ by 7%.

In Tables 29-31, we see that in most tests, (KMBO+) using $\widetilde{\Delta}_1^+$ produced larger best, average, and least cut sizes over using $\widetilde{\Delta}_s^+$ as the choice of signless Laplacian.

In Table 28, we see that the computation time of (KMBO+) using $\widetilde{\Delta}_s^+$ is lower than (KMBO+) using $\widetilde{\Delta}_1^+$, for all 10 of the scale free graphs we tested on. It is interesting to note that the computation times of (KMBO+) using $\widetilde{\Delta}_s^+$ on the AS5 and the WikiVote graphs are substantially lower than using $\widetilde{\Delta}_1^+$ as the choice of signless Laplacian, for any $K \in \{3, 4, 5\}$.

8.2.5 *Max-K-Cut, where $K \geq 5$*

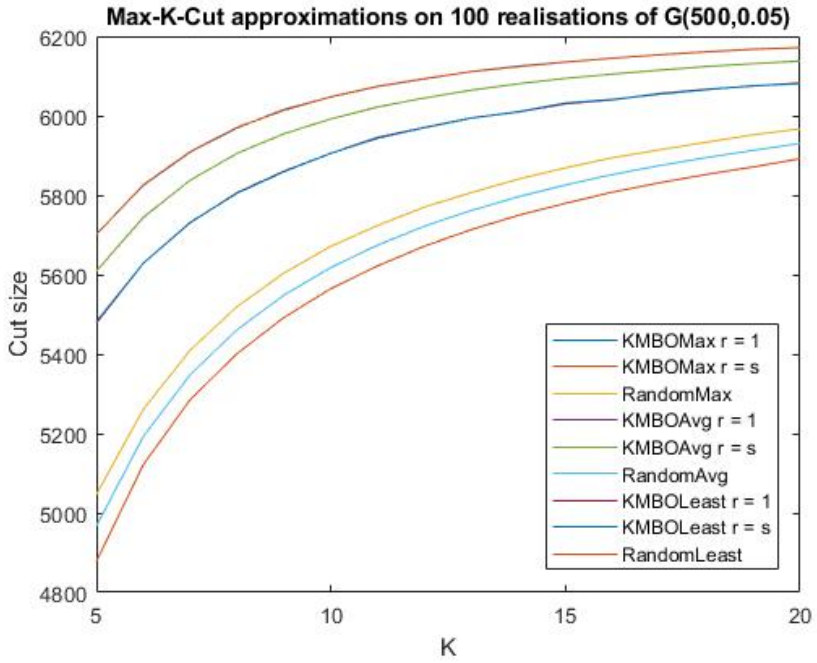
In this sub-chapter we test (KMBO+) on approximately solving the Max-K-Cut problem where $K \geq 5$. In these tests we ran (KMBO+) using $\widetilde{\Delta}_1^+$ and $\widetilde{\Delta}_s^+$, for $K \in \{5, 6, \ldots, 20\}$, and compared them to K-cuts chosen uniformly at random. We display the best, average, and least cuts induced by (KMBO+) and K-cuts chosen at random, and we display the percentage increase or

decrease that the cuts induced by (KMBO+) produced in comparison to K-cuts chosen uniformly at random.

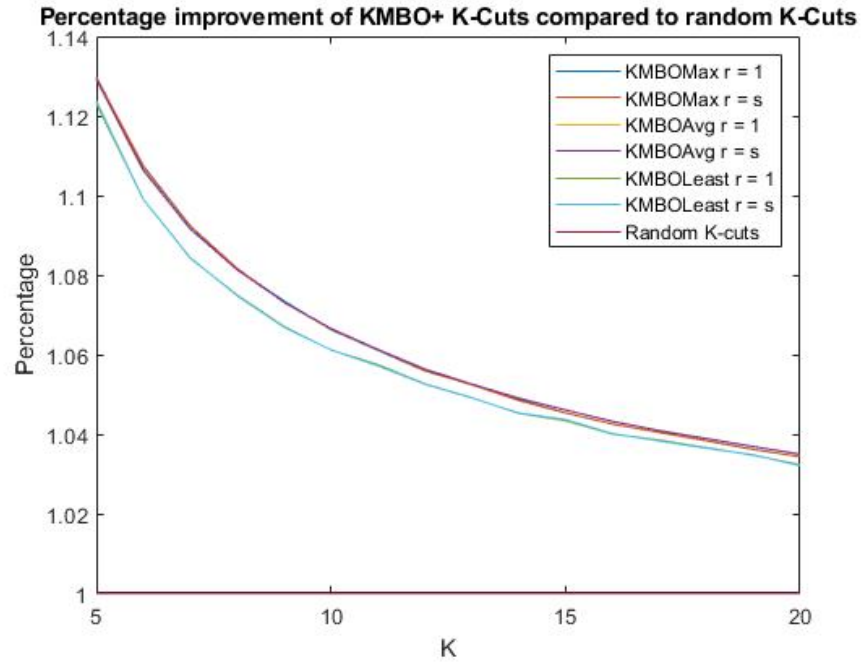
In Figure 65 we test on 100 realisations of $G(500, 0.05)$. In Figure 65a we see that for all $K \in \{5, \dots, 20\}$, the best, average, and least cuts produced by (KMBO+), for either choice of signless Laplacian, are larger than the Best K-cuts chosen uniformly at random for all $K = \{5, \dots, 20\}$. Figure 65b shows that the 20-cuts produced by (KMBO+) are roughly 4% greater than the 20-cuts produced at random.

In Figure 66, we tested (KMBO+) on the AS8 graph. In Figure 66a, we see that for all $K \in \{5, \dots, 20\}$, the best and average cuts produced by (KMBO+) using $\widetilde{\Delta}_1^+$ are larger than the Best K-cuts chosen uniformly at random. For $K \in \{5, \dots, 19\}$, the best cuts produced by (KMBO+) using $\widetilde{\Delta}_s^+$ are larger than K-cuts chosen at random. Figure 66b shows that for $K \in \{5, \dots, 10\}$, the best, average, and least cuts induced by (KMBO+) were all larger than their respective best, average, and least random K-cuts. However, for $K \in \{15, \dots, 20\}$, the average and least cuts produced by (KMBO+) were inferior to the average and least K-cuts chosen at random.

In Figure 67, we tested (KMBO+) on the GNutella graph. In Figure 67a, we see that for all $K \in \{5, \dots, 20\}$, the best, average, and least cuts produced by (KMBO+) are larger than the Best K-cuts chosen uniformly at random for all $K = \{5, \dots, 20\}$, excluding for $K = 19$, where the least 19-cut induced by (KMBO+) was smaller than the best random 19-cut. Figure 67b shows that for all $K \in \{5, \dots, 20\}$, the best, average, and least K-cuts produced by (KMBO+) were larger than the respective best, average, and least cuts chosen at random, using both $\widetilde{\Delta}_1^+$ or $\widetilde{\Delta}_s^+$ as the choice of signless Laplacian.

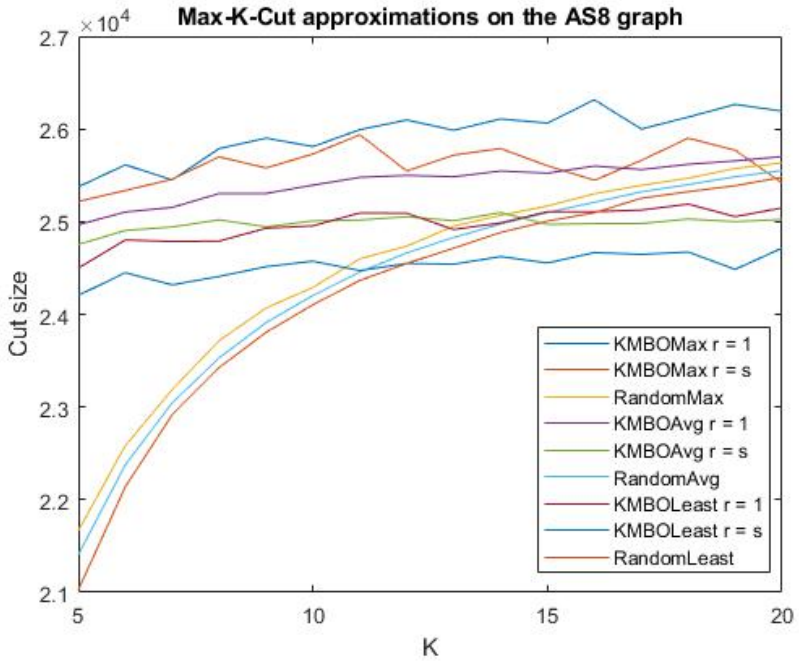


(a) $G(500, 0.05)$: Cut size as function of K, $\tau = 20$.

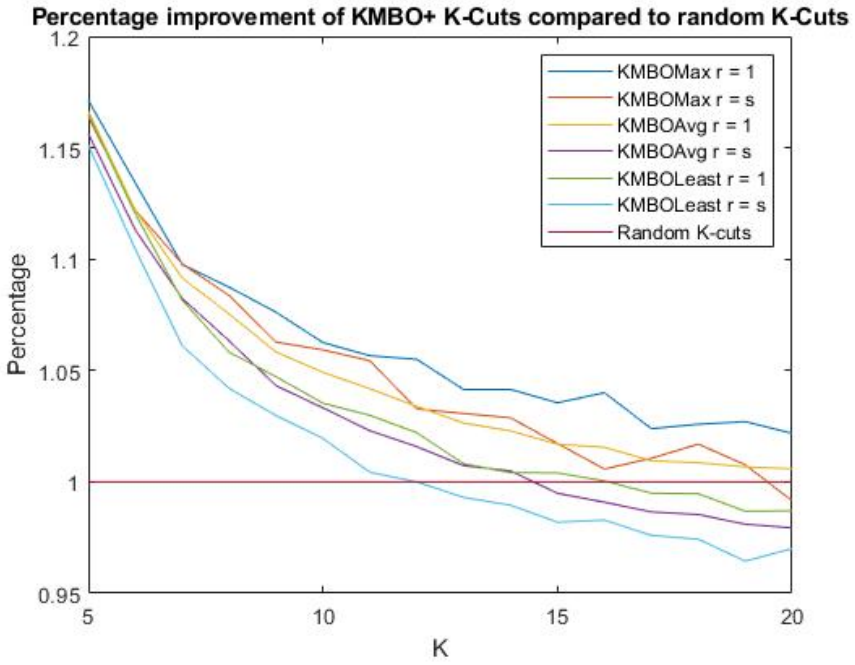


(b) $G(500, 0.05)$: Percentage difference of (KMBO+) cuts compared to random K-cuts as function of K

Figure 65.: Plot of K-cut approximations on 100 realisations of $G(500, 0.05)$, where $K \in \{5, \ldots, 20\}$.

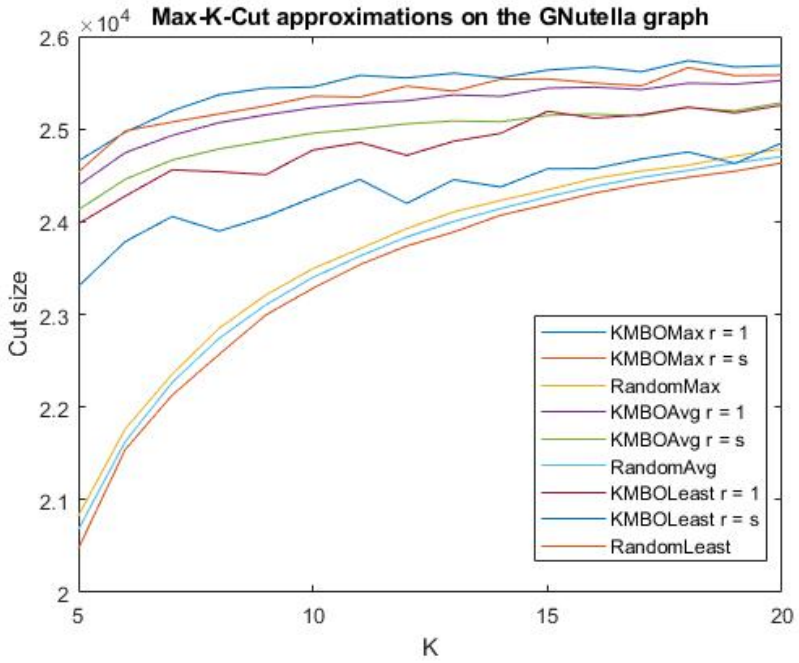


(a) AS8: Cut size as function of K.

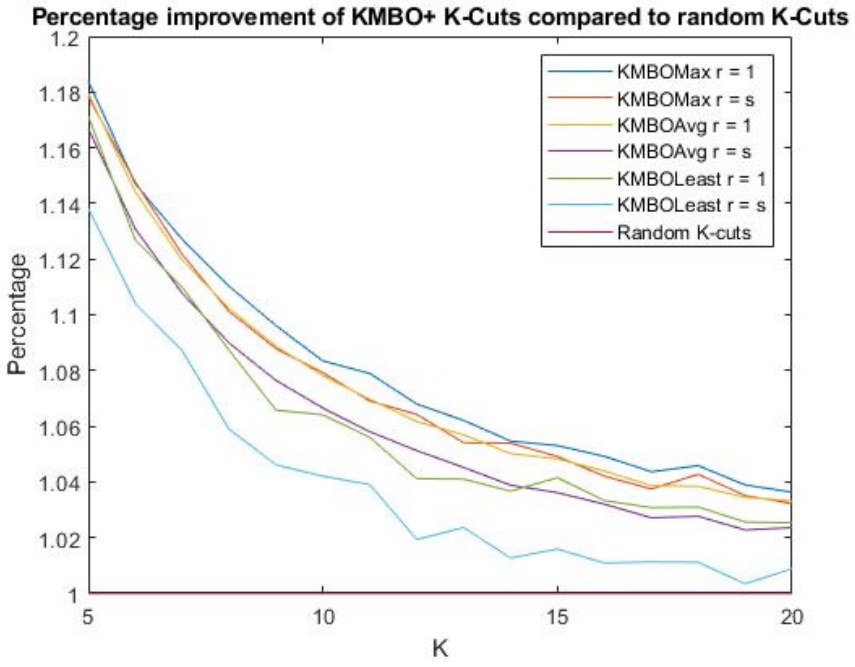


(b) AS8: Percentage difference of (KMBO+) cuts compared to random K-cuts as function of K

Figure 66.: Plot of K-cut approximations on the AS8 graph, where $K \in \{5, \ldots, 20\}$.



(a) GNutella: Cut size as function of K



(b) GNutella: Percentage difference of (KMBO+) cuts compared to random K-cuts as function of K

Figure 67.: Plot of K-cut approximations on the GNutella graph, where $K \in \{5, \ldots, 20\}$.

Graph ($K = 3$)	(KMBO+) Δ_1^+ (Best)	(KMBO+) Δ_s^+ (Best)	Random (Best)
AS1	**24337**	24255	17841
AS2	**14124**	13980	10407
AS3	**16214**	16101	11972
AS4	**16252**	16117	11937
AS5	**15020**	14988	11128
AS6	**19364**	19261	14242
AS7	**24265**	24145	17928
AS8	24553	**24612**	18093
GNutella09	**23034**	22970	17499
WikiVote	78587	**78602**	65095

Graph ($K = 3$)	(KMBO+) Δ_1^+ (Avg)	(KMBO+) Δ_s^+ (Avg)	Random (Avg)
AS1	**23951.46**	23899.12	17717.26
AS2	**13907.06**	13848.36	10269.62
AS3	**15996.30**	15896.72	11818.20
AS4	**15995.82**	15955.26	11771.30
AS5	**14827.28**	14738.84	11014.54
AS6	**19089.60**	19052.34	14073.10
AS7	**23919.82**	23885.78	17714.56
AS8	**24249.22**	24186.36	17930.44
GNutella09	**22834.72**	22674.08	17321.56
WikiVote	**75463.88**	74997.44	64762.08

Graph ($K = 3$)	(KMBO+) Δ_1^+ (Least)	(KMBO+) Δ_s^+ (Least)	Random (Least)
AS1	**23589**	23408	17551
AS2	**13752**	13684	10116
AS3	**15817**	15677	11669
AS4	15807	**15823**	11677
AS5	**14655**	14573	10864
AS6	**18895**	18854	13957
AS7	23554	**23609**	17566
AS8	23711	**23715**	17776
GNutella09	**22507**	22234	17121
WikiVote	**71938**	71337	64469

Table 29.: (KCut), (KMBO+), and random 3-cut approximations on graphs with a scale free structure, time in seconds.

Graph ($K = 4$)	(KMBO+) Δ_1^+ (Best)	(KMBO+) Δ_s^+ (Best)	Random (Best)
AS1	**24756**	24671	20069
AS2	**14335**	14278	11699
AS3	**16529**	16388	13405
AS4	**16558**	16353	13367
AS5	**15380**	15144	12550
AS6	**19629**	19497	15964
AS7	**24887**	24872	20060
AS8	**25291**	24896	20324
GNutella09	**24144**	24059	19694
WikiVote	**80512**	80424	72241

Graph ($K = 4$)	(KMBO+) Δ_1^+ (Avg)	(KMBO+) Δ_s^+ (Avg)	Random (Avg)
AS1	**24413.24**	24299.06	19911.48
AS2	**14141.10**	14015.16	11570.08
AS3	**16286.70**	16145.20	13283.96
AS4	**16263.06**	16163.90	13240.34
AS5	**15081.38**	14912.28	12393.58
AS6	**19422.36**	19301.06	15834.12
AS7	**24410.18**	24246.69	19921.12
AS8	**24709.62**	24501.94	20159.58
GNutella09	**23852.48**	23611.16	19499.72
WikiVote	**76538.18**	76020.12	71918.38

Graph ($K = 4$)	(KMBO+) Δ_1^+ (Least)	(KMBO+) Δ_s^+ (Least)	Random (Least)
AS1	**24017**	23903	19753
AS2	**13966**	13836	11467
AS3	**16003**	15918	13169
AS4	**16075**	16027	13139
AS5	**14859**	14701	12259
AS6	**19139**	19055	15697
AS7	**24062**	23756	19780
AS8	**24280**	24260	19979
GNutella09	**23425**	23026	19349
WikiVote	71495	70301	**71638**

Table 30.: (KMBO+), and random 4-cut approximations on graphs with a scale free structure, time in seconds.

Graph ($K = 5$)	(KMBO+) Δ_1^+ (Best)	(KMBO+) Δ_s^+ (Best)	Random (Best)
AS1	**25038**	24890	21347
AS2	**14584**	14312	12452
AS3	**16827**	16559	14287
AS4	**16739**	16541	14231
AS5	**15636**	15374	13308
AS6	**19991**	19790	17047
AS7	**25119**	24986	21384
AS8	**25353**	25211	21656
GNutella09	**24733**	24593	20936
WikiVote	**81787**	80665	75955

Graph ($K = 5$)	(KMBO+) Δ_1^+ (Avg)	(KMBO+) Δ_s^+ (Avg)	Random (Avg)
AS1	**24600.70**	24487.32	21240.00
AS2	**14273.38**	14097.68	12320.99
AS3	**16436.30**	16223.06	14170.06
AS4	**16410.84**	16238.72	14117.84
AS5	**15192.56**	14991.36	13197.08
AS6	**19589.30**	19457.54	16899.34
AS7	**24617.68**	24465.94	21243.90
AS8	**24922.64**	24771.38	21518.44
GNutella09	**24394.22**	24150.88	20809.34
WikiVote	**77402.76**	75648.28	75642.18

Graph ($K = 5$)	(KMBO+) Δ_1^+ (Least)	(KMBO+) Δ_s^+ (Least)	Random (Least)
AS1	**24346**	24133	21119
AS2	**14089**	13851	12224
AS3	**16160**	16011	14058
AS4	**16214**	16074	13997
AS5	**14980**	14793	13064
AS6	**19291**	19205	16788
AS7	**24130**	24101	21094
AS8	**24508**	24358	21363
GNutella09	**24000**	23546	20635
WikiVote	70292	70442	**75241**

Table 31.: (KCut), (KMBO+), and random 5-cut approximations on graphs with a scale free structure, time in seconds.

8.3 MAX-K-CUT - PARAMETER TESTS

8.3.1 *The parameter k^**

As shown in Chapter 7.2.1, as the number of eigenpairs k^* used in the spectral method of (MBO+) increase, the cut sizes produced by (MBO+) also increase, up to reaching a value of k^* where the cut sizes do not change upon increasing k^*. In Chapter 8.3.1, we test to see if (KMBO+) shows the same behaviour with respect to the choice of k^*.

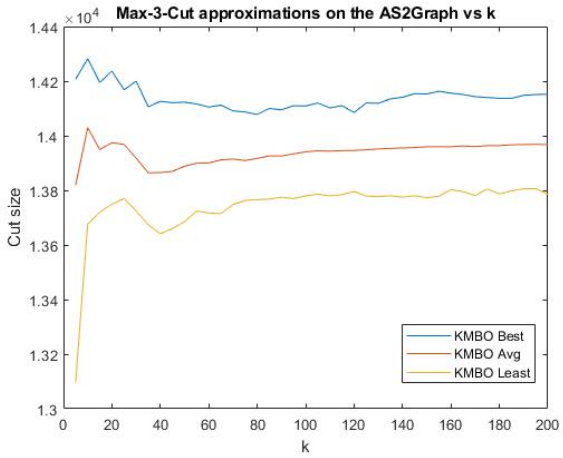
For $K \in \{3, 4, 5\}$, in Figure 68, we ran (KMBO+) on the AS2 graph, increasing the value for k^* in increments of 5 from 5 to 200.

In Figure 68a we see that from $k^* = 40$ to $k^* = 200$ the best cut sizes remain mostly stationary as k^* increases. From $k^* = 40$ to $k^* = 200$, the average cuts slowly increase as k^* increases. We also see that the largest best and average cut was achieved where $k^* = 10$. The least cut size increases sharply from $k^* = 5$ to $k^* = 10$. From $k^* = 10$ onwards, the least cut is mostly stationary.

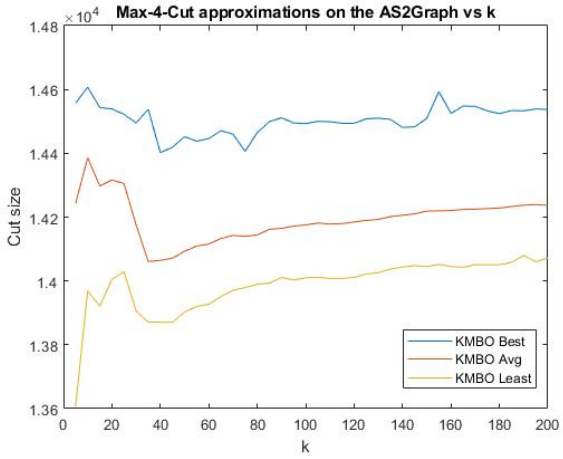
In Figure 68b we see that the best cut and average cut peak where $k^* = 10$. From $k^* = 10$ to $k^* = 40$ the average cut decreases, and from $k^* = 40$ onwards, the average cut size slowly increases as k^* increases. The least cut slowly increases from $k^* = 40$ onwards, with a peak occurring at $k^* = 30$. The best cut does not demonstrate much of a pattern as k^* increases.

In Figure 68c we see that the peak of the best cut and average cut also occurs at $k^* = 10$, as with the Max-3-Cut problem in Figure 68a and Max-4-Cut problem in Figure 68b, and that the trends shown in Figure 68c closely resemble those in Figure 68b.

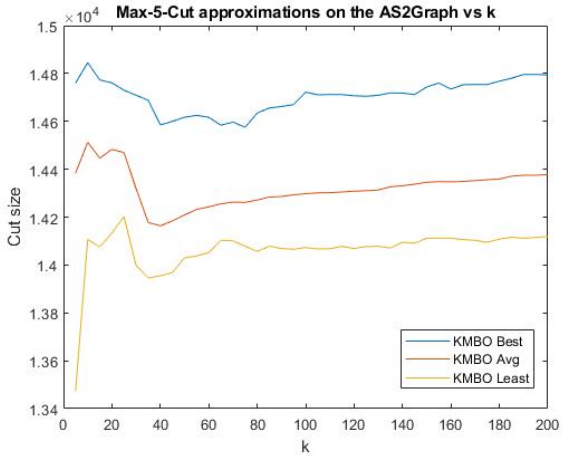
Overall, we see that choosing a value of k^* smaller than $\lfloor \frac{n}{100} \rfloor$ for (KMBO+) may have been beneficial with respect to the cut sizes obtained. Furthermore, the time taken to run (KMBO+) will have decreased for smaller k^*. We see that choosing $k^* = 10$ on these graphs would have mostly likely produced the best results. This is unlike the results in Chapter 7.2.1, where it was preferable to choose larger values of k^* than 10.



(a) AS2: Cut size as function of k^*, $K = 3$, $\tau = 20$.



(b) AS2: Cut size as function of k^*, $K = 4$, $\tau = 20$.



(c) AS2: Cut size as function of k^*, $K = 5$, $\tau = 20$.

Figure 68.: Cut size as function of K for three graphs.

8.3.2 *Variable* τ

In Chapter 8.3.2 we compare the K-cut sizes produced by (KMBO+) against the variable τ.

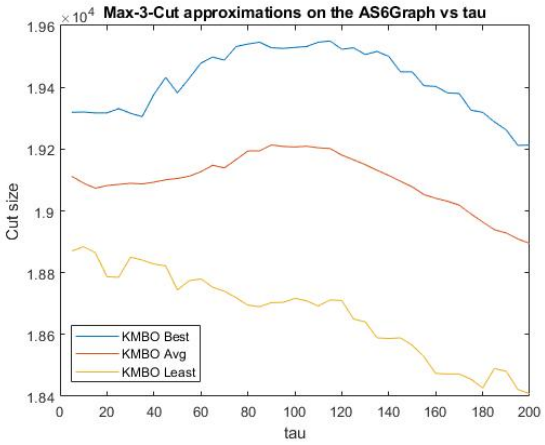
For $K \in \{3, 4, 5\}$, we ran (KMBO+) on the AS6 graph, increasing the value for τ in increments of 5 from 5 to 200; we display the results in Figure 69. In Figures 69a, 69b, and 69c we display the results for the Max-3-Cut, Max-4-Cut, and Max-5-Cut problems respectively.

Figure 69a shows that the least cuts produced by (KMBO+) decrease slightly from $\tau = 5$ to $\tau = 45$, increases from $\tau = 45$ to $\tau = 80$, and decreases from $\tau = 80$ onwards. The average cut tends to increase from $\tau = 10$ to $\tau = 120$, and decreases from $\tau = 120$ onwards. The best cuts have an overall tendency to increase from $\tau = 5$ to $\tau = 90$, remain roughly stable from $\tau = 90$ to $\tau = 140$, and decrease from $\tau = 140$ onwards.

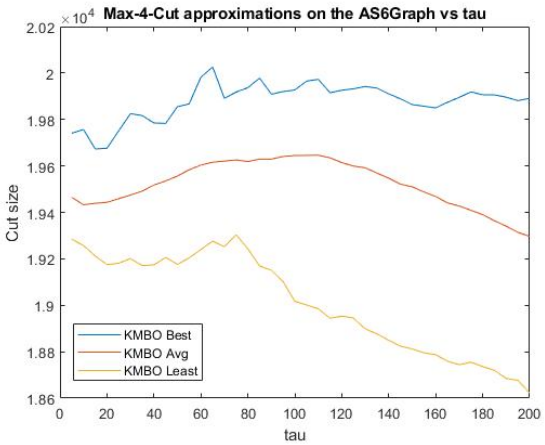
Figure 69b shows that the best cuts produced by (KMBO+) for $\tau = 5$ and $\tau = 200$ are similar, and that the cuts produced between $\tau = 5$ and $\tau = 200$ do not differ greatly from those two values. We see that the average cuts have an overall tendency to increase from $\tau = 10$ up to $\tau = 110$, decreasing for $\tau > 110$. The least cuts decrease from $\tau = 5$ to $\tau = 20$, remaining roughly stationary from $\tau = 20$ to $\tau = 60$, increasing from $\tau = 60$ to $\tau = 80$, and decreasing from that point.

Figure 69c shows that the best cuts produced by (KMBO+) are roughly similar from $\tau = 75$ to $\tau = 200$, with erratic changes from $\tau = 5$ to $\tau = 60$. The average cuts from $\tau = 10$ to $\tau = 120$ increase slowly, and decrease from $\tau = 120$ to $\tau = 200$. The least cut decreases from $\tau = 5$ to $\tau = 25$, roughly increasing from $\tau = 25$ to $\tau = 105$, and decreasing from $\tau = 105$ to $\tau = 200$.

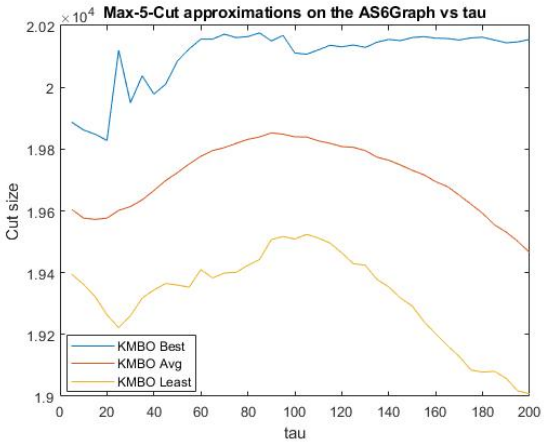
Overall, we see that it is difficult to find a definitive trend for the effect on τ when using (KMBO+) to approximately solve the Max-3-Cut, Max-4-Cut, and Max-5-Cut problem for the AS6 graph. However, these tests suggest that the choice of $\tau = 20$ may not have been optimal for the tests conducted in Chapter 8.2.



(a) AS6: Cut size as function of τ, $K = 3$, $k^* = 89$.



(b) AS6: Cut size as function of τ, $K = 4$, $k^* = 89$.



(c) AS6: Cut size as function of τ, $K = 5$, $k^* = 89$.

Figure 69.: Cut size as function of τ for three graphs.

8.4 MAX-CUT RESULTS BY MINIMIZING $F^+_{\varepsilon,2}$

As shown in Theorem 5.1.5, as $\varepsilon \to 0$, minimizers of $F^+_{\varepsilon,2}$ in (7) will converge to solutions of the Max-Cut problem. In Chapter 8.4 we test (KMBO+), setting $K = 2$, and compare the results to the results obtained by (MBO+). For (MBO+), we choose the spectral method over the Euler method in order to have a more comparable algorithm to (KMBO+).

To ensure that (KMBO+) and (MBO+) are as consistent as possible in the tests in Chapter 8.4, we generate the 50 initial conditions for (MBO+) uniformly at random from $\mathcal{V}^b$; for the 50 initial conditions of (KMBO+), we choose 50 initial conditions from Ω_K, such that those initial conditions induce the same graph cut as the initial conditions for (MBO+). This is to make the results obtained from both algorithms more comparable.

For all tests in Chapter 8.4, we set $\tau = 20$ and $k^* = \lfloor \frac{n}{100} \rfloor$. We test for both $\widetilde{\Delta}^+_1$ and $\widetilde{\Delta}^+_s$ in (KMBO+) and we test for both Δ^+_1 and Δ^+_s in (MBO+).

In Figure 70 and Figure 71, for $K = 2$, we test (KMBO+) and (MBO+) on 100 realisations of $G(1000, 0.01)$ and $G(5000, 0.001)$, respectively, where the 100 realisations are the same realisations as used in the tests in Chapter 7.1.2. In Figure 70 and Figure 71 we see that for either choice of signless Laplacian, (MBO+) produces larger best, average, and least cuts compared to (KMBO+), for either $\widetilde{\Delta}^+_1$ or $\widetilde{\Delta}^+_s$ as the choice of Laplacian.

In Table 32, we see that the run times of (KMBO+) are lower than the run times of (MBO+).

Graph	(KMBO+) Δ^+_1	(MBO+) Δ^+_1	(KMBO+) Δ^+_s	(MBO+) Δ^+_s
$G(1000, 0.01)$	0.14	0.15	**0.12**	0.14
$G(5000, 0.001)$	1.17	6.11	**0.92**	5.89

Table 32.: Average (KMBO+) and (MBO+) run-times for each realisation of $G(n, p)$, time in seconds.

In Table 34 we test (KMBO+) and (MBO+) on the scale free graphs tested on in Chapter 7.1.3. We see in Table 34 that (MBO+) with Δ^+_s produces larger

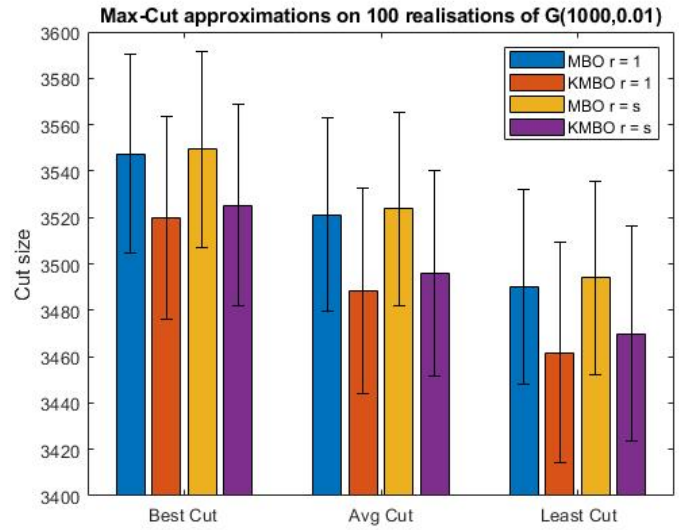


Figure 70.: Bar chart of Max-Cut approximations on 100 realisations of $G(1000, 0.01)$.

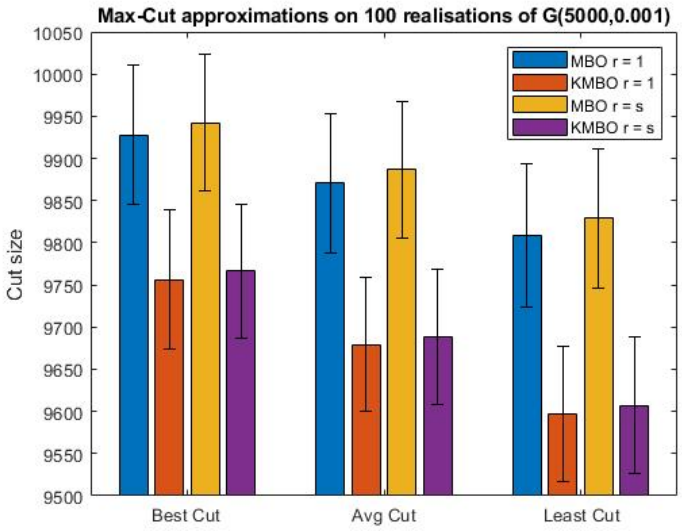


Figure 71.: Bar chart of Max-Cut approximations on 100 realisations of $G(5000, 0.001)$.

best, average, and least cuts compared to (KMBO+) with $\widetilde{\Delta}^+_s$, and (MBO+) with Δ^+_1 produced larger cuts compared to (KMBO+) with $\widetilde{\Delta}^+_1$. However, we note that in some cases, the cuts produced by (KMBO+) with $\widetilde{\Delta}^+_s$ outperformed those produced by (MBO+) with Δ^+_1.

It is also interesting to see in Table 34 that (KMBO+) using either signless Laplacian produced significantly lower results on the WikiVote graph compared to (MBO+). This mirrors the results obtained when testing (KMBO+) on WikiVote in Chapter 8.2.4, where for $K \in \{4,5\}$, (KMBO+) produced K-cuts which were inferior to the least K-cut obtained from 50 K-cuts chosen uniformly at random.

In Table 33 we see that the run times using (KMBO+) with $\widetilde{\Delta}^+_s$ are lower than (MBO+) using Δ^+_s. Similarly, the run times of (KMBO+) using $\widetilde{\Delta}^+_1$ are lower than (MBO+) using Δ^+_1.

Graph	(KMBO+) Δ^+_1	(MBO+) Δ^+_1	(KMBO+) Δ^+_s	(MBO+) Δ^+_s
AS1	11.51	14.13	**8.27**	10.87
AS2	2.54	4.17	**2.08**	3.15
AS3	3.69	5.14	**2.45**	3.88
AS4	5.31	5.71	**4.22**	5.09
AS5	3.22	4.58	**2.68**	3.58
AS6	10.08	8.66	**6.04**	7.09
AS7	12.04	15.59	**8.38**	10.58
AS8	13.01	14.71	**10.12**	10.67
GNutella	2.60	7.58	**1.94**	6.61
WikiVote	1.65	2.13	**1.32**	1.88

Table 33.: Average (KMBO+) and (MBO+) run-times for each scale free graph, time in seconds.

Graph	(KMBO+) Δ_1^+ (Best)	(MBO+) Δ_1^+ (Best)	(KMBO+) Δ_s^+ (Best)	(MBO+) Δ_s^+ (Best)
AS1	22622	22751	22720	**22824**
AS2	13186	**13276**	13179	13255
AS3	15061	15146	15069	**15156**
AS4	15183	15230	15189	**15274**
AS5	13995	14066	14052	**14089**
AS6	18024	18074	18061	**18085**
AS7	22665	22780	22744	**22834**
AS8	22850	22955	22920	**23006**
GNutella09	20015	20260	20009	**20292**
WikiVote	74155	74298	**74472**	**74472**

Graph	(KMBO+) Δ_1^+ (Avg)	(MBO+) Δ_1^+ (Avg)	(KMBO+) Δ_s^+ (Avg)	(MBO+) Δ_s^+ (Avg)
AS1	22302.92	22559.00	22525.82	**22616.16**
AS2	13017.14	13162.98	13095.38	**13179.02**
AS3	14881.42	15023.88	14985.54	**15069.36**
AS4	14993.50	15147.64	15121.66	**15184.10**
AS5	13865.70	13993.00	13954.26	**14003.76**
AS6	17839.96	17959.34	17943.58	**17976.08**
AS7	22347.00	22579.90	22549.12	**22615.12**
AS8	22653.56	22677.40	22794.60	**22826.64**
GNutella09	19727.80	20135.40	19648.64	**20176.46**
WikiVote	70869.56	74144.72	71721.00	**74260.04**

Graph	(KMBO+) Δ_1^+ (Least)	(MBO+) Δ_1^+ (Least)	(KMBO+) Δ_s^+ (Least)	(MBO+) Δ_s^+ (Least)
AS1	21550	**22245**	21822	22220
AS2	12418	13072	12423	**13085**
AS3	14260	14933	14840	**14938**
AS4	14604	15055	15020	**15094**
AS5	13489	13926	13834	**13937**
AS6	17447	17810	17740	**17874**
AS7	21787	22218	22074	**22252**
AS8	22590	22340	22597	**22371**
GNutella09	18831	19986	19084	**20038**
WikiVote	62633	73915	62932	**74128**

Table 34.: (KMBO+) and (MBO+) Max-Cut approximations on graphs with a scale free structure, time in seconds.

9

CONCLUSIONS AND FURTHER WORK

9.1 CONCLUSIONS

9.1.1 *Max-Cut*

We have proven that the signless graph Ginzburg–Landau functional f_ε^+ Γ-converges to a Max-Cut objective functional as $\varepsilon \downarrow 0$ and thus minimizers of f_ε^+ can be used to approximate maximal cuts of a graph. We used an adaptation of the graph MBO scheme involving signless graph Laplacians to approximately minimize f_ε^+. We solved the signless diffusion step of our graph MBO scheme using a spectral truncation method or an Euler method. We also derived a Lyapunov functional for our signless MBO scheme and proved that a rescaled version of this functional Γ-converges to the signless total variation, minimizers of which correspond to maximum cuts.

We tested the resulting (MBO+) algorithm on various graphs using both these signless diffusion solvers, and compared the results and run times with those obtained using the (GW) algorithm stated in Chapter 2.2. In our tests on realisations of random Erdös-Rényi graphs and on realizations of random modular graphs our (MBO+) algorithm using the spectral method outperformed (GW) with reduced run times. On our examples of scale free graphs (GW) usually gave the best maximum cut approximations, but required run times that are two orders of magnitude longer than those of (MBO+), which obtains cut sizes within about 2% of those obtained by (GW). Similar con-

clusions followed from our tests on weighted graphs, that used realisations of randomly generated Erdös-Rényi graphs and modular graphs, and some scale free graphs, all with random edge weights, chosen uniformly at random. We have also shown that our algorithm using the Euler method can be used on large sparse datasets, with reasonable computation times.

We introduced the algorithm (Cut) which minimizes f_ε^+ by using a gradient flow method, where the solution to the gradient flow equation was computed using a convex-concave splitting scheme. On our tests, we showed that the (MBO+) algorithm produced superior results compared to (Cut).

We introduced the algorithm (MBO↓), which is the (MBO+) algorithm with a diffusion time τ that is linearly reduced per iteration. We have shown that for most graphs on which we tested on, (MBO↓) using an Euler method produced superior results over (MBO+) using either an Euler method or the spectral method. On our examples of scale free graphs, (MBO↓) using the Euler method outperformed (GW) on 7 out of 10 examples with respect to the maximum cut sizes obtained.

In our tests (and for our parameter choices) we saw that (MBO+) using both Δ_1^+ and Δ_s^+ produced larger Max-Cut approximations than Δ_0^+ for all of the graphs that we tested on.

9.1.2 *Max-K-Cut*

We have proven that the multiclass signless Ginzburg–Landau functional $F_{\varepsilon,K}^+$ Γ-converges to a Max-K-Cut functional as $\varepsilon \to 0$ and thus minimizers of $F_{\varepsilon,K}^+$ can be used to approximate maximal K-cuts on a graph. We produced two algorithms to find approximate maximal K-cuts, (KCut) and (KMBO+). (KCut) involved solving the gradient flow equation of $F_{\varepsilon,K}^+$ for a small time step, followed by projecting the solution to a simplex whose corners correspond to node subsets within a K-cut. The solutions of the gradient flow equation are obtained using a convex-concave splitting scheme. (KMBO+) involved

solving the K-signless diffusion equation for a small step, and projecting the solution to the same simplex as the simplex used in (KCut).

On smaller graphs (where $n \in \{50, 100\}$), we tested the (KCut) and (KMBO+) algorithms for $K \in \{3, 4, 5\}$, and compared the results and run times with those obtained using the (SDPCut) algorithm. In our tests on realisations of Erdös-Rényi and random modular graphs with 50 and 100 nodes, we showed that in most cases, for all $K \in \{3, 4, 5\}$, both (KCut) and (KMBO+) outperformed (SDPCut) with respect to K-cuts obtained, with reduced run times on most of the graphs we tested on.

We tested (KCut) and (KMBO+) for $K \in \{3, 4, 5\}$ on graphs with more than 100 nodes, and compared the results with K-cuts obtained uniformly at random. In these tests, we showed that on the graphs we tested on, both (KCut) and (KMBO+) produced K-cuts which are consistently better than choosing K-partitions at random, for all $K \in \{3, 4, 5\}$.

We tested (KMBO+) on some graphs, where $K \in \{5, \dots, 20\}$. In most of our tests, we showed that for all $K \in \{5, \dots, 20\}$, (KMBO+) would produce larger K-cuts than K-cuts chosen uniformly at random.

We showed that in our tests, where $K = 2$, (KMBO+) produced inferior results to (MBO+) with respect to the Max-Cut approximations obtained.

In most of our tests, we see that (KMBO+) outperforms (KCut) with respect to the K-cut sizes produced, and for all tests, the computation time of (KMBO+) was lower than the computation time of (KCut). Furthermore, we see that choosing $\widetilde{\Delta}_1^+$ as the choice of signless Laplacian in (KCut) and (KMBO+) produces similar sized K-cuts over choosing $\widetilde{\Delta}_s^+$ for both algorithms.

9.2 FUTURE WORK

9.2.1 *Max-Cut*

There are still many open questions related to the (MBO+) algorithm, for example questions related to a priori parameter choices (such as the time step τ and the number of eigenpairs k^* used for the spectral truncation of Δ_r^+ and Δ_s^+), and performance guarantees. These can be the subject of future research.

9.2.2 *Max-K-Cut*

In Chapter 4.2.4 we show that if the time step τ is chosen to be too small, (MBO+) will have trivial dynamics. We wish to prove a similar result for the dynamics of (KMBO+).

As explained in [32], for an unweighted undirected graph G, if X is a K-cut chosen uniformly at random, and X' is the Max-K-Cut of G, then $\mathbb{E}[S(X)] \geq (1 - \frac{1}{K})S(X')$ will be satisfied. Therefore, as K increases, the Max-K-Cut approximation produced by a K-cut chosen at random will increase in accuracy. We wish to test (KCut) and (KMBO+) for larger values of K on some graphs, and compare against the results obtained by choosing K partitions at random.

As stated in Chapter 8.1, our computer ran out of memory when running (SDPCut) for most graphs tested on. We wish to test (SDPCut) on larger graphs by either using a computer with more memory, or by using an alternative semi-definite programming tool to the one provided in [84].

As with (MBO+), there are questions regarding the parameter choices of (KCut) and (KMBO+). The time step τ and the number of eigenpairs k^* used in the spectral truncation of $\widetilde{\Delta}_1^+$ and $\widetilde{\Delta}_s^+$ are some examples. These could be the subject of future research. We also note that in Chapter 8.3, no parameter

analysis was produced for the choice of ε. In future we would like to see how changing the parameter ε changes the K-cut sizes produced by (KCut).

A

APPENDIX A - MIN-K-CUT

In Appendix A we briefly discuss the Min-K-Cut problem. A minimum K-cut of $\mathcal{G}$ is a K-cut $X^* \in \mathcal{X}^K$ such that, for all cuts $X \in \mathcal{X}^K$, $S_K(X) \geq S_K(X^*)$.

Definition A.0.1. (The Min-K-Cut problem)

Given $G \in \mathcal{G}$ and $K \in \mathbb{N}\backslash\{1\}$, find a minimum K-cut of G.

The Min-K-Cut problem is equivalent to finding the least amount of edges such that their removal from the graph means the graphs now consists of K connected components.

As shown in Chapter 5.1, as $\varepsilon \downarrow 0$, the minimizers of $F^+_{\varepsilon,K}$ will converge to solutions of the Max-K-Cut problem. In this Appendix, we show that a similar result for $F_{\varepsilon,K}$ and the Min-K-Cut problem does not hold. First, we define $F_{0,K} : \mathcal{V}^K \to \mathbb{R} \cup \{\infty\}$ as follows:

$$F_{0,K}(U) := \begin{cases} \|\widetilde{\nabla} U\|^2_{\mathcal{E}^K}, & \text{if } U \in \Omega_K, \\ +\infty, & \text{if } U \in \mathcal{V}^K \backslash \Omega_K. \end{cases} \tag{52}$$

Since $F_{\varepsilon,K}$ and $F_{0,K}$ differ to $F^+_{\varepsilon,K}$ and $F^+_{0,K}$ only by the K-Dirichlet energy terms, we can easily adapt the proofs in Chapter 5.1 in order to prove the following:

Theorem A.0.2. *Let $G \in \mathcal{G}$. Let $\{\varepsilon_k\}_{k\in\mathbb{N}}$ be a sequence such that $\varepsilon_k \downarrow 0$ as $k \to \infty$ and, for each $k \in \mathbb{N}$, let $F_{\varepsilon_k,K}$ be as in* (20)*. Let U_{ε_k} be a minimizer of $F_{\varepsilon_k,K}$. Then $\min_{U\in\mathcal{V}^K} F_{\varepsilon_k,K}(U) \to \min_{U\in\mathcal{V}^K} F_{0,K}(U)$ as $k \to \infty$. Furthermore, there exists $U_0 \in \Omega_K$ and a subsequence $\{U_{\varepsilon_{k'}}\}_{k'\in\mathbb{N}} \subset \{U_{\varepsilon_k}\}_{k'\in\mathbb{N}}$ such that $\|U_{\varepsilon_{k'}} - U_0\|_{\mathcal{V}^K} \to 0$ as $k' \to \infty$. Moreover, $U_0 \in \operatorname{argmin}_{U\in\mathcal{V}^K} F_{0,K}(U)$, where $F_{0,K}$ is as in* (52)*.*

In most cases, the minimizers of $F_{0,K}$ will not correspond to K-cuts, as highlighted below:

Proposition A.0.3. *Let $G \in \mathcal{G}$, and let $F_{0,K}$ be as in (52). Then, all minimizers of $F_{0,K}$ are contained in $\Omega_K \backslash \Omega_K^*$, if and only if G is a graph with L connected components, where $L < K$.*

Proof. First assume that G is a graph with L connected components.

We can express G by $G = G_1 \cup \cdots \cup G_L$, where for each $k \in [L]$, $G_k := (V_k, E_k)$. Let $X' \in \mathcal{X}^L$ such that for $k \in [L]$, $V_k' := \chi_{V_k}$. For all $(i,j) \in E$, we have that, for a $k \in [L]$, $i,j \in V_k'$. Therefore,

$$S_L(X') = \sum_{\substack{k,l \in [L] \\ k<l}} \sum_{i \in V_k, j \in V_l} \omega_{ij} = 0,$$

and if $U_{X'}$ is the matrix that induces the L-cut, then $U_{X'} \in \Omega_K \backslash \Omega_K^*$, and $U_{X'} \in \operatorname*{argmin}_{U \in \mathcal{V}^K} F_{0,K}(U)$.

Let $X \in \mathcal{X}^K$, where $X = V_1 | \ldots | V_K$. Since G is not a graph with K-connected components, for all $X \in \mathcal{X}$, there exists a $k \in [K]$ and there exists $(i,j) \in E_k$ such that, for any $k,l \in [K]$, where $k \neq l$, $i \in V_k$ and $j \in V_l$. Therefore,

$$S_K(X) = \sum_{\substack{k,l \in [L] \\ k<l}} \sum_{m \in V_k, n \in V_l} \omega_{mn} \geq \omega_{ij},$$

meaning $S_L(X') < S_K(X)$. Therefore, since for all $X \in \mathcal{X}^K$, $S_L(X') < S_K(X)$, it follows that if $U_X \in \operatorname*{argmin}_{U \in \mathcal{V}^K} F_{0,K}(U)$, then $U_X \in \Omega_K \backslash \Omega_K^*$.

Next assume that G is a graph with M connected components, where $M \geq K$. We express G by $G = G_1 \cup \cdots \cup G_M$, where for each $k \in [M]$, $G_k := (V_k, E_k)$. Let $X' \in \mathcal{X}^K$, such that for $k \in [K-1]$, $V_k' := \chi_{V_k}$ and $V_K' := \chi_{V_K} + \chi_{V_{K+1}} + \cdots + \chi_{V_M}$. For all $(i,j) \in E$, we have that, for a $k \in [K]$, $i,j \in V_k'$. Therefore, $S_K(X') = 0$, and if $U_{X'}$ is the matrix that induces the K-cut, then $U_{X'} \in \Omega_K^*$, and $U_{X'} \in \operatorname*{argmin}_{U \in \mathcal{V}^K} F_{0,K}(U)$.

□

We propose adding a barrier function to $F_{\varepsilon,K}$ in order to define a functional which Γ-converges to a Min-K-Cut objective functional.

Definition A.0.4. Let $G \in \mathcal{G}$, $K \in \mathbb{N}$, and let $F_{\varepsilon,K} : \mathcal{V}^K \to \mathbb{R}$ be as in (20). The barrier multiclass Ginzburg–Landau functional $\bar{F}_{\varepsilon,K} : \mathcal{V}^K \to \mathbb{R}$ is defined by

$$\bar{F}_{\varepsilon,K}(U) := F_{\varepsilon,K}(U) + G_\varepsilon(U) \tag{53}$$

where

$$G_\varepsilon(U) := \sum_{k\in[K]} X_\varepsilon(|\sum_{i\in V} U_{ik}| - (V-2)),$$

and

$$X_\varepsilon(x) := \begin{cases} \frac{x^2}{\varepsilon}, & \text{if } x \geq 0, \\ 0, & \text{if } x < 0. \end{cases}$$

Next we define a Min-K-Cut objective functional $F_{0,K} : \mathcal{V}^K \to \mathbb{R} \cup \{\infty\}$, where

$$\bar{F}_{0,K}(U) := \begin{cases} \|\widetilde{\nabla} U\|^2_{\mathcal{E}^K}, & \text{if } U \in \Omega_K^*, \\ \infty, & \text{if } U \in \mathcal{V}^K \backslash \Omega_K^* \end{cases} \tag{54}$$

Proposition A.0.5. *Let $G \in \mathcal{G}$, let $\bar{F}_{\varepsilon,K}$ be as in (53), and let $\bar{F}_{0,K}$ be as in (54). Then, as $\varepsilon \to 0$, $\bar{F}_{0,K}$ is the Γ-limit of $\bar{F}_{\varepsilon,K}$.*

Proof. First we prove that $W^{\pm}_{\varepsilon,K} + G_\varepsilon$ Γ-converges to $\bar{W}^{\pm}_{0,K}$ as $\varepsilon \downarrow 0$, where

$$\bar{W}^{\pm}_{0,K}(U) := \begin{cases} 0, & \text{if } U \in \Omega_K^*, \\ +\infty, & \text{if } U \in \mathcal{V}^K \backslash \Omega_K^*. \end{cases}$$

(LB) For all $U \in \mathcal{V}^K$, we have that $W^{\pm}_{\varepsilon,K}(U) + G_\varepsilon(U) \geq 0$. Let $\{U_p\}_{p=1}^{\infty}$ be a sequence such that $U_p \to U_0$ as $p \to \infty$. If $U_0 \in \Omega_K^*$, then

$$W^{\pm}_{0,K}(U_0) = 0 \leq \liminf_{p\to\infty} \left(W^{\pm}_{\varepsilon_p,K}(U_p) + G_{\varepsilon_p}(U_p)\right).$$

Next assume that $U_0 \in \mathcal{V}^K \backslash \Omega_K$. As shown in the proof of Lemma 5.1.3, since $U_0 \in \mathcal{V}^K \backslash \Omega_K$, if $U_p \to U_0$, as $p \to \infty$, then $\liminf_{p\to\infty} W^{\pm}_{\varepsilon_p,K}(U_p) = \infty$.

Since for all $U \in \mathcal{V}^K$, $G_e(U) \geq 0$, it follows that if $U_p \to U_0$, as $p \to \infty$, then

$$\liminf_{p\to\infty} \left(W^{\pm}_{\varepsilon_p,K}(U_p) + G_{\varepsilon_p}(U_p)\right) \geq \liminf_{p\to\infty} W^{\pm}_{\varepsilon_p,K}(U_p) = \infty = \bar{W}^{\pm}_{0,K}(U_0).$$

Now assume that $U_0 \in \Omega_K \backslash \Omega_K^*$. Since $U_0 \in \Omega_K \backslash \Omega_K^*$, then $\exists k' \in [K]$ such that $|\sum_{i\in V} U_{ik'}| = V$.

Since $U_p \to U_0$ as $p \to \infty$, for every $\eta > 0$ there is an $N(\eta) \in \mathbb{N}$ such that for all $p \geq N(\eta)$, we have that $\|U_p - U_0\|_{\mathcal{V}^K} \leq \eta$. Then, for all $p \geq N(\eta)$,

$$\begin{aligned}\Big|\big|\sum_{i\in V}(U_p)_{ik'}\big| - (V-2)\Big| &\geq \Big|\big|\sum_{i\in V}(U_0)_{ik'}\big| - (V-2)\Big| - \Big|\big|\sum_{i\in V}(U_0)_{ik'}\big| - \big|\sum_{i\in V}(U_p)_{ik'}\big|\Big| \\ &\geq 2 - \eta > 1.\end{aligned}$$

It follows that there is a $C > 1$ such that for all $p \geq N(\eta)$, $(|\sum_{i\in V}(U_p)_{ik'} - (V-2)) > C$. Since $W^{\pm}_{\varepsilon,K}(U) \geq 0$ for all $U \in \mathcal{V}^K$, we have that

$$\begin{aligned}\liminf_{p\to\infty}\left(W^{\pm}_{\varepsilon_p,K}(U_p) + G_{\varepsilon_p}(U_p)\right) &\geq \liminf_{p\to\infty} G_{\varepsilon_p}(U_p) \\ &\geq \liminf_{p\to\infty} \frac{1}{\varepsilon_p}\Big(\sum_{i\in V}(U_p)_{ik'} - (V-2)\Big)^2 \\ &= \infty = \bar{W}^{\pm}_{0,K}(U_0)\end{aligned}$$

(UB) If $U_0 \in \mathcal{V}^K \backslash \Omega^*_K$, then $\bar{W}^{\pm}_{0,K}(U_0) = \infty$ and the upper bound condition is trivially satisfied. If $U_0 \in \Omega^*_K$, for all $k \in [K]$, there exists an $i \in V$ such that $(U_0)_{ik} = 1$. Therefore, for all $k \in [K]$, $\sum_{i\in V}(U_0)_{ik} - (V-2) \leq 0$, meaning $G_\varepsilon(U_0) = 0$. For all $p \in \mathbb{N}$, define the sequence $\{U_p\}_{p=1}^{\infty}$ where $U_p = U_0$. Then, for all $p \in \mathbb{N}$, $W^{\pm}_{\varepsilon_p,K}(U_p) + G_{\varepsilon_p}(U_p) = 0$, and thus $\limsup_{p\to\infty}\left(W^{\pm}_{\varepsilon_p,K}(U_p) + G_{\varepsilon_p}(U_p)\right) = 0 = \bar{W}^{\pm}_{0,K}$.

Therefore, $W^{\pm}_{\varepsilon,K} + G_\varepsilon$ Γ-converges to $\bar{W}^{\pm}_{0,K}$ as $\varepsilon \downarrow 0$.

Since $\|\widetilde{\nabla} U\|^2_{\mathcal{E}^K}$ is continuous on $\mathcal{V}^K$, by Proposition 3.3.2, as $\varepsilon \downarrow 0$, $\bar{F}_{\varepsilon,K}$ Γ-converges to $\bar{F} : \mathcal{V}^K \to \mathbb{R} \cup \{\infty\}$, where

$$\bar{F}(U) = \|\widetilde{\nabla} U\|^2_{\mathcal{E}^K} + \bar{W}^{\pm}_{0,K}.$$

If $U \in \mathcal{V}^K \backslash \Omega^*_K$, then $\bar{F}(U) = \infty = \bar{F}_{0,K}$. If $U \in \Omega^*_K$, then $\bar{F}(U) = \|\widetilde{\nabla} U\|^2_{\mathcal{E}^K} = \bar{F}_{0,K}$, which concludes the proof.

□

BIBLIOGRAPHY

[1] M. Agrawal, N. Kayal, and N. Saxena, PRIMES is in P, Annals of mathematics JSTOR, (2004), 781–793

[2] R. Albert, H. Jeong and A.-L. Barabási, Internet: Diameter of the world-wide web, Nature, (1999), 130–131.

[3] S. M. Allen and J. W. Cahn, A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening, Acta Metallurgica, **27** (1979), 1085–1095.

[4] L. Ambrosio, N. Gigli and G. Savaré, Gradient Flows: In Metric Spaces and in the Space of Probability Measures, Springer Science & Business Media, 2008.

[5] K. Appel and W. Haken, Every planar map is four colorable. Part I: Discharging, in Illnois Journal of Mathematics, **21** (1977), 429–490.

[6] A. L. Barabási, Scale-free networks: A decade and beyond, Science, (2002), 412–413.

[7] A. L. Barabási and E. Bonabeau, Scale-free, Scientific American, (2003), 50–59.

[8] G. Barles and C. Georgelin, A simple proof of convergence for an approximation scheme for computing motions by mean curvature, SIAM Journal on Numerical Analysis, **32** (1995), 484–500.

[9] A. L. Bertozzi and A. Flenner, Diffuse interface models on graphs for classification of high dimensional data, Multiscale Modeling & Simulation, **10** (2012), 1090–1118.

[10] B. Bollobás, Degree sequences of random graphs, Discrete Mathematics, **33**, (1981), 1–19.

[11] B. Bollobás, Modern Graph Theory, Graduate Texts in Mathematics, 184. Springer-Verlag, New York, 1998.

[12] A. Braides, Gamma-convergence for Beginners, Clarendon Press, 2002.

[13] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner, Maximizing modularity is hard, in arXiv preprint physics/0608255, (2006).

[14] L. Bronsard and R. V. Kohn, Motion by mean curvature as the singular limit of Ginzburg–Landau dynamics, Journal of Differential Equations, **90** (1991), 211–237.

[15] J. Budd and Y. van Gennip, Graph MBO as a semi-discrete implicit Euler scheme for graph Allen–Cahn, in arXiv preprint arXiv:1907.10774, (2019)

[16] L. Calatroni, Y. van Gennip, C.-B. Schönlieb, H. M. Rowland and A. Flenner, Graph clustering, variational image segmentation methods and Hough transform scale detection for object measurement in images, Journal of Mathematical Imaging and Vision, **57** (2017), 269–291.

[17] D. Calvetti, L. Reichel and D. C. Sorensen, An implicitly restarted Lanczos method for large symmetric eigenvalue problems, Electronic Transactions on Numerical Analysis, **2** (1994), 1–21.

[18] G. J. Chaitin, Register allocation & spilling via graph coloring, in ACM Sigplan Notices, **17**, (1982), 98–105.

[19] F. R. K. Chung, Spectral Graph Theory, American Mathematical Society, 1997.

[20] S. A. Cook, Proceedings of the third annual ACM symposium on Theory of computing, in Proceedings of the third annual ACM symposium on Theory of computing ACM, (1971), 151–158.

[21] R. Courant and D. Hilbert, Methods of Mathematical Physics, CUP Archive, 1965.

[22] M. Cucuringu, A. Pizzoferrato and Y. van Gennip, An MBO scheme for clustering and semi-supervised clustering of signed networks, in arXiv preprint arXiv:1901.03091, (2019).

[23] G. Dal Maso, An Introduction to Γ-convergence, in Progress in Nonlinear Differential Equations and Their Applications, Birkhäuser, 1993.

[24] E. de Klerk, D. V. Pasechnik, and J. P. Warners, On approximate graph colouring and Max-K-Cut algorithms based on the θ-function, in Journal of Combinatorial Optimization, **8** (2004), 267–294.

[25] M. Desai and V. Rao, A characterization of the smallest eigenvalue of a graph, Journal of Graph Theory, **18** (1994), 181–194.

[26] W. E. Donath and A. J. Hoffman, Lower bounds for the partitioning of graphs, Selected Papers Of Alan J Hoffman: With Commentary, (2003), 437–442.

[27] C. S. Edwards, An improved lower bound for the number of edges in a largest bipartite subgraph, in Proc. 2nd Czechoslovak Symposium on Graph Theory, Prague, (1975), 167–181.

[28] S. Esedoğlu and F. Otto, Threshold dynamics for networks with arbitrary surface tensions, Communications on Pure and Applied Mathematics, **68** (2015), 808–864.

[29] D. J. Eyre, An unconditionally stable one-step scheme for gradient systems, in Unpublished article, (1998).

[30] M. Fiedler, Algebraic connectivity of graphs, in Czechoslovak mathematical journal, **23** (1973), 298–305.

[31] J. G. F. Francis, The QR transformation a unitary analogue to the LR transformation—Part 1, The Computer Journal, **4** (1961), 265–271.

[32] A. Frieze and M. Jerrum, Improved approximation algorithms for Max-K-Cut and max bisection, in Algorithmica, **18** (1997), 67–81.

[33] C. Garcia-Cardona, E. Merkurjev, A. L. Bertozzi, A. Flenner, and A. G. Percus, Multiclass data segmentation using diffuse interface methods on graphs, in IEEE transactions on pattern analysis and machine intelligence, **36** (2014), 1600–1613.

[34] M. R. Garey, D. S. Johnson, and L. Stockmeyer, Some simplified NP-complete graph problems, in Theoretical computer science, Elsevier, (1976), 237–267

[35] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness, San Francisco, LA: Freeman, 1979.

[36] M. X. Goemans and D. P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, Journal of the ACM (JACM), **42** (1995), 1115–1145.

[37] M. X. Goemans and D. P. Williamson, Approximation algorithms for MAX-3-CUT and other problems via complex semidefinite programming, in Journal of Computer and System Sciences, **68** (2004), 442–470.

[38] G. H. Golub and H. A. van der Vorst, Eigenvalue computation in the 20th century, in Journal of Computational and Applied Mathematics, **123** (2000), 35–65.

[39] M. Grmela and H. C. Öttinger, Dynamics and thermodynamics of complex fluids. I. Development of a general formalism, in Physical Review E, **56**, (1997), 6620.

[40] M. Grötschel and W. R. Pulleyblank, Weakly bipartite graphs and the Max-Cut problem, Operations Research Letters, **1** (1981), 23–27.

[41] J.-M. Guo, A new upper bound for the Laplacian spectral radius of graphs, Linear Algebra and Its Applications, **400** (2005), 61–66.

[42] V. Guruswami, Maximum cut on line and total graphs, Discrete Applied Mathematics, **92** (1999), 217–221.

[43] F. Hadlock, Finding a maximum cut of a planar graph in polynomial time, SIAM Journal on Computing, **4** (1975), 221–225.

[44] Y. Haribara, S. Utsunomiya and Y. Yamamoto, A coherent Ising machine for MAX-CUT problems: Performance evaluation against semidefinite programming and simulated annealing, in Principles and Methods of Quantum Information Technologies, Springer, **911** (2016), 251–262.

[45] M. Hein, J.-Y. Audibert and U. von Luxburg, Graph Laplacians and their convergence on random neighborhood graphs, Journal of Machine Learning Research, **8** (2007), 1325–1368.

[46] R.A. Horn and C.R. Johnson, Matrix Analysis. Cambridge University Press, (2012)

[47] H. Hu, T. Laurent, M. A. Porter and A. L. Bertozzi, A method based on total variation for network modularity optimization using the MBO scheme, SIAM Journal on Applied Mathematics, **73** (2013), 2224–2246.

[48] R. Impagliazzo and R. Paturi, On the complexity of k-SAT, in Journal of Computer and System Sciences, **62**, (2001), 367–375.

[49] M. Jacobs, E. Merkurjev, and S. Esedoglu, Auction dynamics: A volume constrained MBO scheme, in Journal of Computational Physics, **354**, (2018), 288–310.

[50] A. M. Jaffe, The Millennium Grand Challenge in Mathematics, in Notices of the AMS, **53**, (2006), 652–660.

[51] V. Kann, S. Khanna, J. Sanjeev, and A. Panconesi, On the hardness of approximating max k-cut and its dual, in Chicago Journal of Theoretical Computer Science, **2** (1997), 151–180.

[52] C.-Y. Kao and Y. van Gennip, Private communication.

[53] R. M. Karp, Reducibility among combinatorial problems, in Complexity of Computer Computations, Springer, (1972), 85–103.

[54] B. Keetch and Y. van Gennip, A Max-Cut approximation using a graph based MBO scheme, in Discrete & Continuous Dynamical Systems-B, **24**, (2019), 6091–6139.

[55] S. Khot, On the power of unique 2-prover 1-round games, in Proceedings of the thirty-fourth Annual ACM Symposium on Theory of Computing, ACM, (2002), 767–775.

[56] S. Khot, G. Kindler, E. Mossel and R. O'Donnell, Optimal inapproximability results for MAX-CUT and other 2-variable CSPs?, SIAM Journal on Computing, **37** (2007), 319–357.

[57] G. Kirchhoff, Ueber die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanischer Ströme geführt wird, in Annalen der Physik, **148** (1847), 497–508.

[58] R. E. Ladner, On the structure of polynomial time reducibility, in Journal of the ACM (JACM), **22** (1975), 155–171.

[59] J. B. Lasserre, A MAX-CUT formulation of 0/1 programs, Operations Research Letters, **44** (2016), 158–164.

[60] H. T. Lang, A Java library of graph algorithms and optimization, (2006).

[61] R. B. Lehoucq and D. C. Sorensen, Deflation techniques for an implicitly restarted Arnoldi iteration, SIAM Journal on Matrix Analysis and Applications, **17** (1996), 789–821.

[62] D. Marx, Graph colouring problems and their applications in scheduling, in Periodica Polytechnica Electrical Engineering, **48**, (2004), 11–16.

[63] K. S. McCurley, The discrete logarithm problem, in AMS Proc. Symp. Appl. Math, (1990), 49–74.

[64] E. Merkurjev, T. Kostic and A. L. Bertozzi, An MBO scheme on graphs for classification and image processing, SIAM Journal on Imaging Sciences, **6** (2013), 1903–1930.

[65] E. Merkurjev, C. Garcia-Cardona, A. L. Bertozzi, A. Flenner, and A. G. Percus, Diffuse interface methods for multiclass segmentation of high-dimensional data, in Applied Mathematics Letters, **33**, (2014), 29–34.

[66] B. Merriman, J. Bence and S. Osher, Diffusion Generated Motion by Mean Curvature, UCLA Department of Mathematics CAM report CAM 06–32, 1992.

[67] B. Merriman, J. Bence and S. Osher, Diffusion generated motion by mean curvature, AMS Selected Letters, Crystal Grower's Workshop, (1993), 73–83.

[68] H. D. Mittelmann, The state-of-the-art in conic optimization software, in Handbook on Semidefinite, Conic and Polynomial Optimization, Springer, **166** (2012), 671–686.

[69] L. Modica and S. Mortola, Un esempio di Γ^{-}-convergenza, Boll. Un. Mat. Ital. B (5), **14** (1977), 285–299.

[70] L. Modica, The gradient theory of phase transitions and the minimal interface criterion, Archive for Rational Mechanics and Analysis, **98** (1987), 123–142.

[71] B. Mohar, The Laplacian spectrum of graphs, in Graph Theory, Combinatorics, and Applications, Volume 2, Wiley, (1991), 871–898.

[72] P. L. Montgomery, A survey of modern integer factorization algorithms, in CWI quarterly, **7** (1994), 337–366.

[73] M. E. J. Newman Modularity and community structure in networks, in Proceedings of the national academy of sciences, **103** (2006), 8577–8582.

[74] C. H. Paradimitriou, The Euclidean travelling salesman problem is NP-complete, in Theoretical computer science, Elsevier, (1977), 237–244.

[75] R. J. Radke, A Matlab Implementation of the Implicitly Restarted Arnoldi Method for Solving Large-scale Eigenvalue Problems, Masters thesis, Rice University, 1996.

[76] R. C. Read and D. G. Corneil, The graph isomorphism disease, in *Journal of Graph Theory*, **1** (1977), 339–363.

[77] M.P. Rombach, M. A. Porter, J.H. Fowler and P. J. Mucha, Core-periphery structure in networks, in *SIAM Journal on Applied Mathematics*, **74** (2014), 167–190.

[78] W. Rudin, Principles of Mathematical Analysis, McGraw-hill New York, 1964.

[79] S. Sahni and T. Gonzalez, P-complete approximation problems, in Journal of the ACM (JACM), **23** (1976), 555–565

[80] D. C. Sorensen, Implicitly restarted Arnoldi/Lanczos methods for large scale eigenvalue calculations, in Parallel Numerical Algorithms, Springer, **4** (1997), 119–165.

[81] J. A. Soto, Improved analysis of a Max-Cut algorithm based on spectral partitioning, SIAM Journal on Discrete Mathematics, **29** (2015), 259–268.

[82] L. Trevisan, G. B. Sorkin, M. Sudan and D. P. Williamson, Gadgets, approximation, and linear programming, *SIAM Journal on Computing*, **29** (2000), 2074–2097.

[83] L. Trevisan, Max-Cut and the smallest eigenvalue, SIAM Journal on Computing, **41** (2012), 1769–1786.

[84] R. H. Tütüncü, K.-C. Toh and M. Todd, Solving semidefinite-quadratic-linear programs using SDPT3, Mathematical Programming, **95** (2003), 189–217.

[85] Y. van Gennip and A. L. Bertozzi, Γ-convergence of graph Ginzburg-Landau functionals, Advances in Differential Equations, **17** (2012), 1115–1180.

[86] Y. van Gennip, N. Guillen, B. Osting and A. L. Bertozzi, Mean curvature, threshold dynamics, and phase field theory on finite graphs, Milan Journal of Mathematics, **82** (2014), 3–65.

[87] Y. van Gennip, An MBO scheme for minimizing the graph Ohta–Kawasaki functional, Journal Of Nonlinear Science, (2018), 1–49.

[88] U. von Luxburg, A tutorial on spectral clustering, Statistics and computing, **17** (2007), 395–416.

[89] E. Weinan, Dynamics of vortex liquids in Ginzburg-Landau theories with applications to superconductivity, in Physical Review B, **50**, (1994), 1126–1135.

[90] A. L. Yuille and A. Rangarajan, The concave-convex procedure (CCCP), in Advances in neural information processing systems, (2002), 1033–1040.

[91] University of Ioannina, Department of Computer Science and Engineering Teaching Resources, `http://www.cs.uoi.gr/tsap/teaching/InformationNetworks/data-code.html.`, Accessed: 2017-03-13.

[92] J.Yoo, shallow-DANN-two-moon-dataset (MATLAB), `https://github.com/jaejun-yoo/shallow-DANN-two-moon-dataset`, Accessed: 2018-09-11.

[93] MATLAB, Graph and Network Algorithms, `https://uk.mathworks.com/help/matlab/graph-and-network-algorithms.html`, Accessed: 2015-11-13.

[94] MIT Strategic Engineering Research Group: Matlab Tools for Network Analysis (2006-2011), `http://strategic.mit.edu/docs/matlab_networks/random_modular_graph.m.`, Accessed: 2017-07-20.

[95] SNAP Datasets: Stanford Large Network Dataset Collection, `http://snap.stanford.edu/data`, Accessed: 2017-07-01.