


# Triangles Improve 0.878 Approximation for Maxcut

Fredie George 

Indian Institute of Science, Bangalore, India

Anand Louis 

Indian Institute of Science, Bangalore, India

Rameesh Paul 

Indian Institute of Science, Bangalore, India

---

## Abstract

MAXCUT is a fundamental problem in graph algorithms, extensively studied for its theoretical and practical significance. The goal is to partition the vertex set of a graph  $G = (V, E)$  into disjoint subsets  $S$  and  $V \setminus S$  so as to maximize the number of edges crossing the cut  $(S, V \setminus S)$ . The seminal work of Goemans and Williamson [39] introduced a semidefinite programming (SDP) based algorithm achieving a  $\alpha_{\text{GW}} \approx 0.87856$ -approximation for general graphs, guaranteed to be optimal under the Unique Games Conjecture [56, 57].

We revisit the Goemans–Williamson SDP and prove that the standard MAXCUT SDP achieves a  $(\alpha_{\text{GW}} + \Omega(1))$ -approximation whenever the input graph contains  $\Omega(|E|)$  edge-disjoint triangles. Our analysis builds on classical rounding techniques studied in [39, 76] and introduces a refined understanding of the SDP solution structure in regimes where the previous guarantees are tight. Our result identifies a simple combinatorial property that may be satisfied by many natural graph classes.

As applications, we show that unit ball graphs and graphs satisfying a spectral transitivity condition (as studied in [40, 12]) meet our structural criterion, and therefore we get better than  $\alpha_{\text{GW}}$  approximation guarantees for them. Our algorithm runs in nearly linear time  $\tilde{O}(|E|)$ , offering a more practical alternative to the PTAS of [48] for unit ball graphs, which has exponential dependence on the approximation parameter.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Approximation algorithms analysis; Theory of computation  $\rightarrow$  Graph algorithms analysis

**Keywords and phrases** Approximation Algorithms, Maxcut, Semidefinite Programming, Edge-disjoint Triangles, Unit Ball Graphs, Spectral Triadic Graphs

**Digital Object Identifier** 10.4230/LIPIcs.APPROX/RANDOM.2025.27

**Category** APPROX

**Funding** *Anand Louis*: Supported by the Walmart Center for Tech Excellence at IISc, and SERB award CRG/2023/002896.

*Rameesh Paul*: Supported by Prime Minister’s Research Fellowship, India.

**Acknowledgements** We thank Rahul Saladi for valuable discussions. We thank anonymous reviewers for their insightful comments.

## 1 Introduction

We study the MAXCUT problem, a fundamental problem in combinatorial optimization. We are given a graph  $G = (V, E)$ , and the goal is to partition the vertex set  $V$  into disjoint subsets  $S$  and  $V \setminus S$  such that the number of edges crossing the cut  $(S, V \setminus S)$  is maximized. MAXCUT is one of Karp’s original NP-hard problems [53] and has been extensively studied both for its theoretical depth and its wide range of applications, including areas such as VLSI design [23], statistical physics [9], and image segmentation [15].



© Fredie George, Anand Louis, and Rameesh Paul;  
licensed under Creative Commons License CC-BY 4.0

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2025).

Editors: Alina Ene and Eshan Chattopadhyay; Article No. 27; pp. 27:1–27:25



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A simple algorithm that outputs a random cut in the graph achieves a 0.5-approximation. For a long time, this remained the best known algorithm for the problem, until the seminal work of Goemans and Williamson [39] introduced a semidefinite programming (SDP) based algorithm that achieves roughly 0.87856-approximation factor. The exact value of the approximation ratio is known as the Goemans-Williamson constant  $\alpha_{\text{GW}} \approx 0.87856$ . The guarantee is also tight in the worst-case, due to an integrality gap construction [36], and is guaranteed to be optimal assuming the Unique Games Conjecture [56, 57].

While the SDP achieves  $\alpha_{\text{GW}}$ -approximation ratio in the worst-case instances, it is natural to ask whether stronger guarantees can be obtained on more structured inputs. In this work, we identify a simple combinatorial property, namely the presence of many edge-disjoint triangles, that suffices to improve the approximation ratio achieved by the Goemans-Williamson SDP. We show that this property holds for several natural classes of graphs, including unit ball graphs and certain real-world networks. Our results thus provide provable guarantees explaining improved SDP performance on structured instances, offering a step towards a broader understanding of approximation algorithms beyond the worst-case.

## 1.1 Our Results and Techniques

We start by presenting our main result about the MAXCUT SDP,

► **Theorem 1** (Informal Version of Theorem 18). *Given a graph  $G = (V, E)$ , having optimal MAXCUT value  $\text{opt}(G)$ , there exists an algorithm  $\text{Alg}(G)$  running in  $\tilde{O}(|E|)$  time such that,*

$$\mathbb{E}[\text{Alg}(G)] \geq \left( \alpha_{\text{GW}} + \Omega \left( \frac{\Delta(G)}{|E|} \right)^2 \right) \text{opt}(G),$$

where  $\Delta(G)$  denotes the number of edge-disjoint triangles in  $G$ .

Our approach involves a refined analysis of the standard MAXCUT SDP. The original analysis of Goemans and Williamson [39], and subsequent follow-up works [76, 35], already show an  $\Omega(1)$  improvement (over  $\alpha_{\text{GW}}$ ) in certain regimes of  $\text{sdopt}$ , the SDP optimum value. We leverage these results as black-box components. Our main technical contribution is to show that for the remaining regimes of  $\text{sdopt}$ , those where there are no known guarantees, the approximation ratio can still be improved by an additive factor of  $\Omega(\Delta(G)/|E|)^2$ . This yields a uniformly improved  $(\alpha_{\text{GW}} + \Omega(1))$ -approximation ratio in graphs whenever the number of edge-disjoint triangles  $\Delta(G) = \Omega(|E|)$ .

Although SDP-based algorithms are often considered computationally expensive, recent developments [3, 65, 68] have enabled fast implementations for these on a broad class of constraint satisfaction problems (CSP), including MAXCUT.

We then proceed to identify natural classes of graphs where this structural condition holds. Combined with Theorem 1, this suffices to show an improved  $(\alpha_{\text{GW}} + \Omega(1))$ -approximation guarantees. Specifically, we give a constructive proof showing that unit ball graphs, including unit interval graphs, unit disk graphs, and their higher-dimensional generalizations contain  $\Omega(|E|)$  edge-disjoint triangles.

► **Proposition 2** (Informal Version of Theorem 31). *For any geometric intersection graph of unit balls in  $\mathbb{S}^{k-1}$  with degree  $d \geq 6(2)^{3k}$ , it has  $\Omega_k(|E|)$  many edge-disjoint triangles.*

► **Corollary 3.** *For any geometric intersection graph of unit balls in  $\mathbb{S}^{k-1}$  with average degree at least  $6(2)^{3k}$ , there exists an algorithm achieving a  $(\alpha_{\text{GW}} + \Omega(1))$ -approximation for the MAXCUT problem on  $G$ .*

We note that [48] previously gave a PTAS (Polynomial Time Approximation Schemes) for MAXCUT on unit disk graphs via a dynamic programming approach that partitions the graph into dense components, which are then solved using the dense graph algorithm of [4]. They also note that their techniques extend to general unit ball graphs.

► **Remark 4.** We contrast our running time with that of the algorithm by the work [48], which achieves a PTAS for MAXCUT on unit ball graphs. When combined with the fast dense graph algorithm of [60] as a subroutine, their overall running time becomes  $\mathcal{O}(|E| + 2^{\mathcal{O}(1/v^2)})$ . The exponential dependence on  $v$  makes the running time of their algorithm impractically large even for values of  $v$  as large as  $v = 0.1$ .

Our work also considers MAXCUT SDP on models inspired by real world networks considered by [12] which satisfy a spectral criterion of  $\tau(G) = \Omega(1)$ . We elaborate on the spectral condition  $\tau(G)$  in Section 4.2; intuitively, it captures a form of triadic closure commonly observed in social networks and community-structured networks.

► **Lemma 5** (Section 3 in [12]). *For any regular graph  $G$  satisfying a spectral condition  $\tau(G) = \Omega(1)$ , the number of edge-disjoint triangles in graph is  $\Omega(|E|)$ .*

► **Remark 6.** Some degenerate unit ball graphs, such as path graphs contain no triangles. However, if they satisfy a mild constant lower bound on the average degree, they indeed have high edge-disjoint triangle density. We also show that graphs satisfying the spectral criterion in the work [12] have large number of edge-disjoint triangles. For graphs with low average degree ( $|E| = \mathcal{O}(|V|)$ ), we can alternatively use the SDP-based algorithm of [44], which achieves a  $(\alpha_{\text{GW}} + \Omega(1))$ -approximation, albeit with higher running time due to additional  $l_2^2$ -triangle inequality constraints in their SDP

## 1.2 Related Work

### Maxcut Problem

The problem has been extensively studied across various settings. As noted earlier, in the worst-case graphs, a simple 0.5-approximation was known from [32, 66] which was best until [39] gave an  $\alpha_{\text{GW}}$ -approximation algorithm. The SDP is also known to be tight due to an integrality gap example of [36]. On the hardness front, MAXCUT is APX-hard in general graphs, and has a 16/17-hardness of approximation due to [46, 71]. Further,  $\alpha_{\text{GW}}$  is also conjectured to be the optimal approximation ratio achievable assuming UGC [56, 57].

The work of [70] showed how to achieve a better than 0.5-approximation without using SDP. They show that the eigenvector corresponding to the smallest eigenvalue of the graph's adjacency matrix could be used to obtain at least 0.531-approximation algorithm, the bound was later improved to 0.614 by [67].

The problem has also been considered in various special graph classes. On the algorithmic front, [43] shows that the problem can be solved for planar graphs. The work [41] shows that the problem is polynomial time solvable for line graphs. The work [15] shows how to solve the problem efficiently on bounded treewidth graphs. The work [19] gives an efficient algorithm for co-bipartite chain graphs. The approximation guarantees also have been studied extensively in works of [25, 52, 70]. The work [4] showed how LP relaxations can yield a PTAS for the problem on dense graphs. The work [60] later showed that a greedy randomized algorithm can give a PTAS for the problem, with a runtime of  $\mathcal{O}(|E| + 2^{\mathcal{O}(1/v^2)})$ .

The work [13] shows that the problem remains NP-hard on cubic graphs. The work [41] showed it is NP-hard for total graphs. The work [15] then showed that it remains NP-hard for chordal graphs, split graphs, co-bipartite graphs, and tripartite graphs. Lower bounds on  $\text{opt}(G)$  for the problem were also studied in the works of [32, 31, 22].

## Maxcut SDP

SDP algorithms are central to state-of-the-art solutions for a range of problems, such as graph coloring [54], sparsest cut [5], correlation clustering (maximizing agreements version) [69], maximizing cut norm of matrix [2], max bisection [6], maxcut in general graphs [39], among others. While SDP algorithms are traditionally viewed as “slow”, recent developments [3, 68] enable fast implementation of these algorithms for many Constraint Satisfaction Problems (CSPs) such as the MAXCUT problem.

The MAXCUT SDP itself has received a lot of attention over the years. The SDP formulation itself was originally proposed in an earlier work of [29], and [39] gave a randomized rounding algorithm which produced a cut having (in expectation) at least  $\alpha_{\text{GW}}$  fraction of the edges in the optimal cut. The work [76] gave a rounding algorithm based on “outward rotations”, which we review in Section 2.2. Later [35] further generalized this with the RPR<sup>2</sup> rounding procedure; which they claim is at least as good as algorithm by [76] and potentially better than [76] (Theorem 5.2 in [35]), though this improvement was formally demonstrated only for particular values of  $\text{opt}(G)$ . This culminated with the work of [64] that characterized the entire approximability curve of the MAXCUT SDP on general graphs.

For bounded degree graphs, the work of [34] showed that the MAXCUT SDP combined with a local algorithm achieves a  $\alpha_{\text{GW}} + \Omega(1/d^4)$ -approximation algorithm, which was later improved to  $\alpha_{\text{GW}} + \Omega(1/d^3)$  in the work [37], and recently improved to  $\alpha_{\text{GW}} + \Omega(1/d^2 \log d)$ -approximation by the work [44], where  $d$  is the maximum degree of the graph. For dense graphs and their spectral generalizations, the works of [10, 42] showed that one can obtain a PTAS for this problem in time  $n^{\mathcal{O}(1/\varepsilon)}$  by using a higher-order SDP (degree  $\mathcal{O}(1/\varepsilon)$  sum of squares hierarchy). Combining these results with ideas from [48], one can also obtain a PTAS for unit ball graphs (albeit with higher running time compared to [60]).

## Fast SDP Solvers

The work [3] developed the framework of approximately solving SDPs using the Matrix Multiplicative Weight Update method. The work [68] builds on this to develop a fast SDP solver by designing an efficient separation oracle. When paired with the general rounding algorithm of [65], it yields an algorithm for approximately solving a canonical SDP relaxation for a CSP with  $n$  variables and  $m$  constraints (up to a  $(1 - v)$  factor for some  $v > 0$ ) in running time  $\mathcal{O}(m \text{poly}(\log n) \text{poly}(1/v))$ .

## Geometric Intersection Graphs

Geometric intersection graphs are graphs where vertices represent geometric objects, such as intervals, disks, rectangles, polygons etc., and there is an edge between vertices if the corresponding objects overlap. These graphs have applications in wireless networks [45], resource allocation [7], protein sequencing [50], VLSI design [23], statistical physics [9], and image segmentation [15]. An important class of geometric intersection graphs are interval graphs. It is well-known that many NP-hard problems become easy on interval graphs, such as graph isomorphism [17], maximum clique/independent set problem [47], hamiltonian cycle [55], maximum dominating set [24], graph coloring [75, 21], minimum vertex cover [59].

## Maxcut in Geometric Intersection Graphs

One may conjecture that MAXCUT problem follows a similar narrative, and this was posed as an open question in [49]. Earlier works due to [16] and then [18] claimed a polynomial time algorithm for the problem, however both proofs were subsequently reported to be

incorrect due to the works of [14] and [58] respectively. The problem thus remained open for a very long time, and the mystery was resolved only recently in [1] where they show that the problem is NP-hard on general interval graphs.

An interval count of an interval graph is defined as the number of distinct interval lengths in an interval graph, and thus unit interval graphs have interval count 1. The work of [28] showed that the problem is NP-hard even if interval count is 4. More recently, [11] improved upon this to show that MAXCUT remains NP-hard even when interval count is 2. However, whether the problem is NP-hard for unit interval graphs (interval count 1) remains an open question. The work of [30] established that the problem is NP-hard on unit disk graphs.

On the algorithmic front, [48] proposed a PTAS (polynomial time approximation scheme) for the problem on unit ball graphs. They give a graph partitioning algorithm that employs a Dynamic Programming approach to partition the graph into dense components such that  $(1 - v)$ -fraction of edges lie inside these components. For dense components they apply the dense graph MAXCUT algorithm of [4] and obtain an  $(1 - \mathcal{O}(v))$ -approximation with running time  $\mathcal{O}(|E| + 2^{\mathcal{O}(1/v^2)})$ , which is impractically large for even large values of  $v$  (say  $v = 0.1$ ).

## Real World Graphs

While the performance of MAXCUT SDP on random Erdős–Rényi  $G(n, p)$  graphs has received considerable attention [27, 62, 38, 61], such models fail to capture many key features of real-world networks, such as community structure, heavy-tailed degree distributions, and high clustering coefficients. There are various models proposed to mimic such real-world features as: the Preferential Attachment Model due to [8], the small-world network model due to [74], and more recently frameworks such as the combinatorial triangle-dense decompositions by [40] and the “spectral triadic decomposition” by [12]. These capture the role of triangles and higher-order motifs in capturing the nuanced community structure of real-world networks. We will discuss real-world graphs in detail in Section 4.2.

## 2 Overview and Discussion

We start by setting up some notation in Section 2.1. We review the proof of the algorithm due to [39] and the critical values where their analysis is tight. Then we discuss the improvement by [76], followed by our algorithm. In Section 2.4, we present an overview of the proof and our main result. The proof is in two parts, first we show (formally in Section 3) that proving a certain structural property about the SDP can lead to an improved approximation algorithm. Later, in Section 4 we show that the desired property holds for some natural class of graphs.

### 2.1 Notation

Let  $E(U, V)$  denote the set of edges from set  $U$  to set  $V$ , and  $K_n$  denote the complete graph on  $n$  vertices. We let  $T(G)$  denote the maximum number of triangles in a graph, and  $\Delta(G)$  denote the maximum number of edge-disjoint triangles in a graph  $G$ . We let  $\mathcal{N}(0, 1)^n$  denote a multivariate normal distribution in  $\mathbb{R}^n$  where each of the  $n$  components is an independent standard normal random variable with mean 0 and variance 1.

### 2.2 Preliminaries

We recall the MAXCUT SDP relaxation due to [29], typically presented in vector formulation,

► **SDP 7** (MAXCUT SDP).

$$\begin{aligned} & \max \sum_{\{i,j\} \in E} \frac{1}{2} (1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle) \\ & \text{subject to} \\ & \quad \|\mathbf{v}_i\|^2 = 1 \quad \forall i \in V \end{aligned}$$

The SDP was analyzed using a novel randomized rounding procedure by [39] where they show how to obtain an  $\alpha_{\text{GW}}$  approximation algorithm (denoted  $\text{Alg}_{\text{GW}}$ ) for the MAXCUT problem. Their rounding algorithm samples a random hyperplane by sampling the normal (a random  $n$  dimensional gaussian vector  $\mathbf{g} \sim \mathcal{N}(0, 1)^n$ ) and computes the set  $S_g = \{i : \langle \mathbf{v}_i, \mathbf{g} \rangle \geq 0\}$ . They output the edges cut by the algorithm as  $E(S_g, V \setminus S_g)$ . We reproduce their analysis below. We let  $\text{opt}_{\text{MC}} = \text{opt}(G)$  denote the optimal value for the MAXCUT problem in  $G$  and  $\text{sdpopt} = \text{opt}(\text{SDP 7})$ .

► **Definition 8** (GW Critical Constants). *We consider the functions  $h_r(t)$  and  $g_r(t)$  below from the analysis in works of [39] and [76],*

$$h_r(t) = \frac{\arccos(r(1-2t))}{\pi} \text{ and let } g_r(t) = \frac{h_r(t)}{t} \text{ where } t \in (0, 1] \text{ and } r \in [0, 1].$$

We define the unique minimum value of  $g_1(t)$  as  $\alpha_{\text{GW}}$  ( $\approx 0.87856$ ) which is obtained for  $t = t_c \approx 0.84458$ . Also, we let  $\theta_c = \pi h_1(t_c) = \arccos(1 - 2t_c) \approx 133.56345^\circ$ .

► **Theorem 9** (Theorem 3.3 in [39]). *For a graph  $G = (V, E)$  the following holds,*

$$\mathbb{E}_{\mathbf{g} \sim \mathcal{N}(0, 1)^n} [ |E(S_g, V \setminus S_g)| ] \geq \alpha_{\text{GW}} \text{opt}_{\text{MC}}$$

**Proof.** Their proof is a term by term analysis of the edges cut by the rounding algorithm and it's contribution to the SDP objective value. Let  $\theta_{ij} \stackrel{\text{def}}{=} \arccos(\langle \mathbf{v}_i, \mathbf{v}_j \rangle)$  and we note,

$$\begin{aligned} \mathbb{E}_{\mathbf{g} \sim \mathcal{N}(0, 1)^n} [ |E(S_g, V \setminus S_g)| ] &= \sum_{\{i,j\} \in E} \mathbb{E}_{\mathbf{g} \sim \mathcal{N}(0, 1)^n} [ \mathbf{1}_{\{i,j\} \in E(S_g, V \setminus S_g)} ] \\ &= \sum_{\{i,j\} \in E} \mathbb{P}_{\mathbf{g} \sim \mathcal{N}(0, 1)^n} [ \{i, j\} \in E(S_g, V \setminus S_g) ] \\ &= \sum_{\{i,j\} \in E} \frac{\theta_{ij}}{\pi} \quad (\text{Look at the plane spanned by } \mathbf{v}_i \text{ and } \mathbf{v}_j) \\ &= \sum_{\{i,j\} \in E} \left( \frac{2\theta_{ij}}{\pi(1 - \cos(\theta_{ij}))} \right) \left( \frac{1}{2} (1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle) \right) \\ &= \sum_{\{i,j\} \in E} \left( \frac{\arccos(1 - 2t_{ij})}{\pi t_{ij}} \right) \left( \frac{1}{2} (1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle) \right) \quad (\text{Let } t_{ij} = (1 - \cos(\theta_{ij}))/2) \\ &\geq \sum_{\{i,j\} \in E} g_1(t_{ij}) \left( \frac{1}{2} (1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle) \right) \geq \alpha_{\text{GW}} \sum_{\{i,j\} \in E} \left( \frac{1}{2} (1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle) \right) \\ &= \alpha_{\text{GW}} \text{sdpopt} \geq \alpha_{\text{GW}} \text{opt}_{\text{MC}} \quad \blacktriangleleft \end{aligned}$$

Since the above is an edge by edge analysis, it is tight when the function  $g_1(t)$  has the same value i.e.,  $g_1(t_e) = g_1(t_c)$  for all edges  $e \in E$ , which occurs when  $t_{ij} = t_c, \forall \{i, j\} \in E$  and  $\theta_{ij} = \theta_c, \forall \{i, j\} \in E$ . We note that when this happens,  $\text{sdpopt} = t_c |E|$ .

When the  $\text{sdpopt}$  value is strictly larger than  $t_c |E|$ , there are edges for which  $t_e > t_c$ . In this case, for such edges we have that  $g_1(t_e) > \alpha_{\text{GW}}$ . Then the work [39] gives an improved approximation guarantee presented below,

► **Theorem 10** (Theorem 3.1.1 in [39]). *For a graph  $G = (V, E)$  and some  $\varepsilon_u > 0$  such that  $\text{sdpopt} \geq (t_c + \varepsilon_u) |E|$  we have that,* 
$$\mathbb{E}_{\mathbf{g} \sim \mathcal{N}(0,1)^n} [|E(S_g, V \setminus S_g)|] \geq g_1(t_c + \varepsilon_u) \text{opt}_{\text{MC}}.$$

► **Remark 11** (Lemma 3.5 in [39]). For a choice of  $\varepsilon_u > 0$ , we have that  $g_1(t_c + \varepsilon_u) > g_1(t_c)$  since  $t_c$  is the unique minimizer of  $g_1(t)$ .

However, a similar argument can't be made for the light maxcut setting, i.e., where  $\text{sdpopt} = (t_c - \varepsilon_l) |E|$  for some  $\varepsilon_l > 0$ . This is since, it may be possible that for a  $(1 - \varepsilon_l)$  fraction of the edges  $t_e = t_c$  while it is 0 for the other  $\varepsilon_l |E|$  edges. We note that if the  $\text{sdpopt}$  value is very close to  $1/2$ , then indeed a random cut gives improved approximation guarantees. However for  $\text{sdpopt} \geq |E|/2t_c \stackrel{\text{def}}{=} \alpha_{\text{random}} |E| \approx 0.56911 |E|$ , the approximation factor of a random cut is also strictly smaller than  $\alpha_{\text{GW}}$ . Therefore, for a large range of values for  $\text{sdpopt}$  in the interval  $[\alpha_{\text{random}} |E|, t_c |E|]$ , [39] can only guarantee a  $\alpha_{\text{GW}}$ -approximation.

This is where the work [76] introduced a novel rounding technique and analysis that gives an algorithm (denoted  $\text{Alg}_{\text{rot}}$ ) which improves the approximation guarantees in the light maxcut setting. Their rounding algorithm is based on “outward rotations”, an idea proposed by [63]. We present their algorithm next.

### Zwick's Outward Rotation Algorithm [76]

**Input.** The algorithm takes a graph  $G = (V, E)$  and a parameter  $\tau$  as input.

**Algorithm.** We present their algorithm from [76] denoted  $\text{Alg}_{\text{rot}}(G, \tau)$  below,

- Solve the MAXCUT SDP to obtain a collection of vectors  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ .
- They embed the SDP vectors in  $\mathbb{R}^{2n}$  (by padding them with  $n$  zeroes).
- An angle  $\phi$  is obtained by solving the below system of equations in variables  $r, t$ ,

$$\frac{\arccos(r(1-2t)) - \arccos(r)}{t} = \frac{2r}{\sqrt{1-r^2(1-2t)^2}} \quad (1)$$

$$\frac{1 - \frac{t|E|}{\text{sdpopt}}}{\sqrt{1-r^2}} = \frac{1-2t}{\sqrt{1-r^2(1-2t)^2}} \quad (2)$$

and letting  $\phi = \arccos(\sqrt{r})$ . If  $\text{sdpopt} \geq \tau |E|$ , they set  $\phi = 0^\circ$ .

- Consider an orthogonal subspace of vectors  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$  and for each  $i \in [n]$  they rotate the vector  $\mathbf{v}_i$  by an angle  $\phi$  towards the vector  $\mathbf{u}_i$  in the plane spanned by  $\mathbf{v}_i$  and  $\mathbf{u}_i$  to obtain vectors  $\{\mathbf{w}_i\}_{i \in [n]}$  where  $\langle \mathbf{w}_i, \mathbf{w}_j \rangle = \cos^2(\phi) \langle \mathbf{v}_i, \mathbf{v}_j \rangle$ .
- Finally, they run the rounding algorithm of [39] on the  $\{\mathbf{w}_i\}_{i \in [n]}$  vectors.

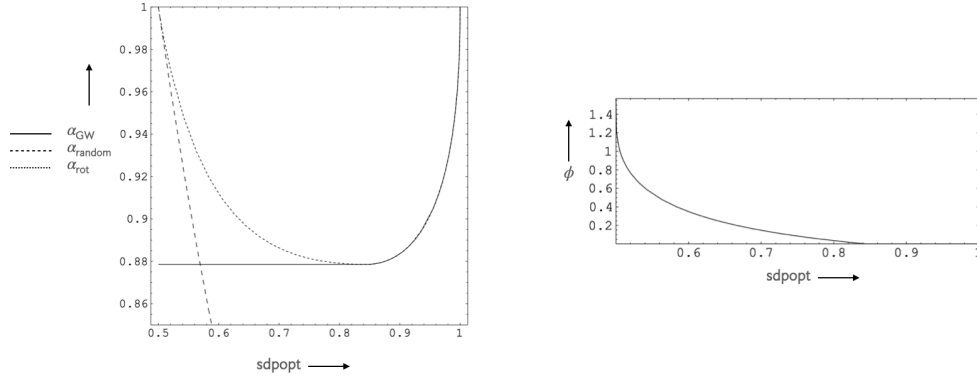
The work [76] then show the following guarantees about their algorithm.

► **Theorem 12** (Theorem 3.3 in [76]). *For a graph  $G = (V, E)$  where  $\text{sdpval} \leq (t_c - \varepsilon_l) |E|$  for  $\varepsilon_l > 0$ ,*

$$\mathbb{E}[\text{Alg}_{\text{rot}}(G, t_c)] \geq \left( \left( \frac{1}{t_c - \varepsilon_l} - \frac{1}{t^*} \right) h_{r^*}(0) + g_{r^*}(t^*) \right) \text{opt}_{\text{MC}}$$

where  $r^*$  and  $t^*$  are the unique solutions to Equation (1) and Equation (2), obtained by setting  $\text{sdpopt} = (t_c - \varepsilon_l) |E|$ .





■ **Figure 1** Improvement by Outward Rotation (Figure 2, Figure 4 in [76]).

► **Remark 13** (Lemma 3.2 in [76]). For a choice of  $\tau = t_c$  and  $\varepsilon_l > 0$ , we have that

$$\left( \frac{1}{t_c - \varepsilon_l} - \frac{1}{t^*} \right) h_{r^*}(0) + g_{r^*}(t^*) > \alpha_{\text{GW}}.$$

### 2.3 Our SDP Based Algorithm

From Theorem 10 and Theorem 12, we observe that if  $\text{sdopt}$  value is bounded sufficiently away from  $t_c$ , it is possible to round the MAXCUT SDP and obtain a strictly better than  $\alpha_{\text{GW}}$ -approximation algorithm. However, we note from Theorem 12 (see also  $\alpha_{\text{rot}}$  in Figure 1) that when  $\text{sdopt} = t_c |E|$ , the algorithm of [76] reduces to algorithm of [39], and as before it offers no improvement beyond  $\alpha_{\text{GW}}$ -approximation. This highlights the need for new ideas to handle the case when  $\text{sdopt} \approx t_c$ . Towards this we make a definition first,

► **Definition 14** (Edge-Disjoint Triangle Density). For a graph  $G = (V, E)$ , we define its edge-disjoint triangle density denoted  $\mu(G)$  as,

$$\mu(G) = \frac{3\Delta(G)}{|E|}.$$

► **Remark 15.** Clearly,  $0 \leq \mu(G) \leq 1$ . The quantity  $\mu(G)$  is similar to the global clustering coefficient of the graph considered in [73, 40]. However, in  $\mu(G)$  we consider edge-disjoint triangles as opposed to total number of triangles.

We now present our algorithm, which we later show achieves an improved approximation ratio regardless of the value of  $\text{sdopt}$ .

**Algorithm.** Our SDP algorithm (denoted  $\mathbf{Alg}(G)$ ) is the following,

■ Fix  $\varepsilon$  to be,

$$\varepsilon \stackrel{\text{def}}{=} \frac{(t_c - 3/4)\mu(G)}{9} \quad (\text{note that clearly } \varepsilon \leq (t_c - 3/4)/9 < t_c) \quad (3)$$

■ Run  $\mathbf{Alg}_{\text{rot}}(G, t_c - \varepsilon)$

Our algorithm essentially runs the Zwick's Outward Rotation Algorithm  $\mathbf{Alg}_{\text{rot}}$  as a subroutine, with the key difference being that we use a smaller threshold value of  $\tau = t_c - \varepsilon$ , as compared to  $\tau = t_c$  in [76]. Thus, for regimes of  $\text{sdopt} \leq (t_c - \varepsilon) |E|$ , our algorithm



reduces to  $\text{Alg}_{\text{rot}}$ , and retains the improvement over  $\alpha_{\text{GW}}$  from Theorem 12. Similarly for  $\text{sdpopt} \geq (t_c + \varepsilon) |E|$ , our algorithm reduces to the standard [39] analysis and we have an improvement due to Theorem 10. We formalize this discussion below.

► **Definition 16** (Minimum Improvement). *We define the minimum improvement (over  $\alpha_{\text{GW}}$ ) as the minimum of the improvement obtained from Theorem 10 by setting  $\varepsilon_l = \varepsilon$  and from Theorem 12 by setting  $\varepsilon_u = \varepsilon$  (for choice of  $\varepsilon$  in Equation (3)) as,*

$$\delta = \min \left\{ g_1(t_c + \varepsilon) - \alpha_{\text{GW}}, \left( \left( \frac{1}{t_c - \varepsilon} - \frac{1}{t^*} \right) h_{r^*}(0) + g_{r^*}(t^*) \right) - \alpha_{\text{GW}} \right\}.$$

Note that  $\delta > 0$  (from Remark 11 and Remark 13) and it follows from the discussion above,

► **Corollary 17.** *For a graph  $G = (V, E)$  with  $\text{sdpopt} \notin [t_c - \varepsilon, t_c + \varepsilon] |E|$ , our algorithm gives,*

$$\mathbb{E}[\text{Alg}(G)] \geq (\alpha_{\text{GW}} + \delta) \text{sdpopt} \geq (\alpha_{\text{GW}} + \delta) \text{opt}_{\text{MC}}.$$

## 2.4 Our SDP Analysis

Building on the discussion above, it suffices to show that our algorithm obtains  $(\alpha_{\text{GW}} + \Omega(1))$ -approximation guarantees when  $\text{sdpopt} \approx t_c |E|$ . Our analysis centers on the contribution of individual edges to the SDP solution, denoted by  $t_e$ . We define  $t_{\text{avg}} = \text{sdpopt} / |E|$ , and we will focus on the values of  $t_{\text{avg}}$  in the band  $[t_c - \varepsilon, t_c + \varepsilon]$ . We note that in these regimes our algorithm sets  $\phi = 0^\circ$  and reduces to the algorithm of [39].

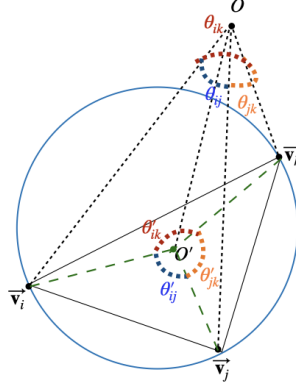
### Refined SDP Analysis

Previously, we mentioned that if  $t_{\text{avg}} = t_c - \varepsilon$ , it's possible that a  $(1 - \varepsilon)$ -fraction of edges have  $t_e = t_c$  while the remaining edges have  $t_e = 0$  yielding no improvement over  $\alpha_{\text{GW}}$ . Here, we consider a more refined analysis of this argument by defining a set of good edges as  $F = \{e \in E : t_e \geq t_c + \varepsilon\}$ . We leverage the edge by edge analysis of [39] to show that,

$$\begin{aligned} \mathbb{E}_{\mathbf{g} \sim \mathcal{N}(0,1)^n} [ |E(S_g, V \setminus S_g)| ] &= \sum_{e \in F} g_1(t_e) t_e + \sum_{e \in E \setminus F} g_1(t_e) t_e && \text{(See Theorem 9)} \\ &\geq \sum_{e \in F} g_1(t_c + \varepsilon) t_e + \alpha_{\text{GW}} \sum_{e \in E \setminus F} t_e && \text{(Definition of } F \text{ and } g) \\ &\geq (\alpha_{\text{GW}} + \delta) \sum_{e \in F} t_e + \alpha_{\text{GW}} \sum_{e \in E \setminus F} t_e && \text{(By Theorem 10)} \\ &= \alpha_{\text{GW}} \sum_{e \in E} t_e + \delta \sum_{e \in F} t_e \\ &\geq \alpha_{\text{GW}} t_{\text{avg}} |E| + \delta |F| (t_c + \varepsilon) && \text{(By definition of } F) \\ &\geq \alpha_{\text{GW}} \text{opt}_{\text{MC}} + \delta |F| (t_c + \varepsilon) && \text{(By definition of } t_{\text{avg}}). \end{aligned}$$

### Improved Approximation Guarantees

The main takeaway from the above analysis is that for  $t_{\text{avg}} \in [t_c - \varepsilon, t_c + \varepsilon]$ , the condition  $|F| = \Omega(1) |E|$  ensures an  $(\alpha_{\text{GW}} + \Omega(1))$ -approximation. We will next identify a sufficient criterion for graphs where this condition holds.



■ **Figure 2** Orientation of SDP vectors for a triangle  $\{i, j, k\}$  in  $G$ .

### Triangles and Good Edges

We consider another set of good edges  $D = \{e \in E : t_e \ll t_c\}$  (say smaller than  $t_c - 100\varepsilon$ ). Now, given a triangle  $T = \{i, j, k\}$  in the graph, we examine the geometry of SDP vectors for  $T$  and note (see Figure 2) that there always exists an edge  $\{i, j\}$  having

$$\theta_{ij} \leq \left( \frac{\theta_{ij} + \theta_{jk} + \theta_{ik}}{3} \right) \leq \left( \frac{\theta'_{ij} + \theta'_{jk} + \theta'_{ik}}{3} \right) \leq \frac{2\pi}{3} \text{ (also see proof of Claim 24).}$$

For such an edge  $e = \{i, j\}$  having  $\theta_{ij} \leq 2\pi/3$  we have its contribution to the SDP value as,

$$t_e = \frac{1}{2} (1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle) = \frac{1}{2} (1 - \cos(\theta_{ij})) \leq \frac{1}{2} (1 - \cos(2\pi/3)) = 0.75.$$

Now  $t_c - 100\varepsilon > 0.75$  (for  $\varepsilon \leq 1/1000$ ) and hence the edge  $e \in D$ . Thus if the graph has “large” number of edge-disjoint triangles, such that  $\Delta(G) = \Omega(1)|E|$  (equivalently  $\mu(G) = \Omega(1)$ ), then each triangle will contribute a unique edge to the set  $D$  and we can infer that  $|D| = \Omega(1)|E|$ .

### Averaging argument saves the day

Recall we earlier show that an improvement over  $\alpha_{\text{GW}}$  is possible when a certain set of good edges  $F$  are large  $|F| = \Omega(1)|E|$ . Then we showed that if the graph has  $\Omega(1)|E|$  many edge-disjoint triangles, a different set of good edges  $D$  such that  $|D| = \Omega(1)|E|$ . One may wonder where are we headed and the connection between these set of good edges?

Here we use the fact that we only care about  $t_{\text{avg}}$  in the band  $[t_c - \varepsilon, t_c + \varepsilon]$ . This allows us to use a simple averaging argument (see Lemma 25 for details) to argue that  $|D| = \Omega(1)|E|$  implies that  $|F| = \Omega(1)|E|$ . Thus, we identify a structural criterion, namely a large number ( $\Omega(1)|E|$  many) of edge-disjoint triangles in a graph that is sufficient for obtaining a  $(\alpha_{\text{GW}} + \Omega(1))$ -approximation.

### Fast SDP Solvers

As we pointed out earlier, for CSPs such as the MAXCUT problem, it is possible to use the result of [68] which builds upon framework of [3, 65] to obtain a feasible SDP solution with  $\text{sdpval} \geq \text{sdpopt} - v|E|$  for some  $v > 0$  in time  $O(|E| \log^2 |V| \text{poly}(1/v))$ . We show later that it is possible to choose  $v$  appropriately and prove Theorem 1.

## 2.5 Applications of Improved SDP Analysis

In this section, we apply our main result (Theorem 1) which shows that the MAXCUT SDP achieves a strictly better-than- $\alpha_{\text{GW}}$  approximation ratio whenever the graph  $G$  contains a constant fraction of edge-disjoint triangles, i.e.,  $\mu(G) = \Delta(G)/|E| = \Omega(1)$ . We show this property for two broad classes of graphs: (1) unit ball graphs (for simplicity we use unit disk graphs in the overview), and (2) real-world networks with strong community structure.

### Unit Disk Graphs

A unit disk graph is defined by mapping vertices to points in  $\mathbb{R}^2$ , where an edge is present between two vertices if their corresponding unit-radius disks intersect. Our goal is to show unit disks graph with sufficiently large average degree contains  $\Omega(|E|)$  edge-disjoint triangles.

We begin by fixing a vertex  $v$  and analyzing its neighborhood  $N(v)$  in the geometric embedding. The vectors from  $v$  to its neighbors are projected onto the unit circle, and this space is partitioned into six small angular sectors (using spherical caps on  $\mathbb{S}^1$ ). By the pigeonhole principle, one such sector contains  $\Omega(\deg(v)/6)$  neighbors, all within an angular range of  $\pi/6$ . This is small enough to ensure that any pair of vertices here have an edge due to the geometry.

Thus, the set of neighbors, along with  $v$ , induces a dense subgraph which contains a clique of size  $\Omega(\deg(v))$ . Using an algorithm that greedily packs triangles in cliques, we extract  $\Omega(\deg(v)^2)$  edge-disjoint triangles from this neighborhood. Repeating this process over all high-degree vertices, we construct a global set of  $\Omega(|E|)$  edge-disjoint triangles. Therefore, any unit disk graph with average degree at least 18 satisfies  $\mu(G) = \Omega(1)$ , and Theorem 1 implies a  $(\alpha_{\text{GW}} + \Omega(1))$ -approximation in  $\tilde{O}(|E|)$  time.

### Real-World Network Graphs

Real-world networks often display features like high clustering, community structure, and triangle density, which are not captured by classical Erdős–Rényi models. To formalize this, we use the “spectral triadic decomposition” framework of [12], which introduces a measure of triangle density called *spectral transitivity*:

$$\tau(G) = \left( \sum_{i=1}^n \lambda_i^3 \right) / \left( \sum_{i=1}^n \lambda_i^2 \right),$$

where  $\lambda_i$  are the eigenvalues of the normalized adjacency matrix.

For  $d$ -regular graphs, they show that  $\tau(G) = \frac{3T(G)}{|E|d}$ , where  $T(G)$  is the total number of triangles in the graph. Using a simple greedy packing argument and upper bounding the triangle load on any edge, we show that such graphs must contain at least  $\Omega(\tau(G)|E|)$  edge-disjoint triangles.

Thus, for such graphs where  $\tau(G) = \Omega(1)$ , we again conclude  $\mu(G) = \Omega(1)$ , and hence the MAXCUT SDP achieves a  $(\alpha_{\text{GW}} + \Omega(1))$ -approximation guarantee.

## 3 Maxcut SDP Analysis

The main result in this section is presenting a sufficient condition under which the MAXCUT SDP achieves better than  $\alpha_{\text{GW}}$  approximation. We begin by recalling that in Equation (3) we fix our value for  $\varepsilon = ((t_c - 3/4)\mu(G))/9$ , where  $\mu(G)$  denotes the edge-disjoint triangle density of a graph. For this choice of  $\varepsilon$ , we also refer to the definition of  $\delta$  from Definition 16.

► **Theorem 18.** *Given a graph  $G = (V, E)$ , there exists an algorithm  $\text{Alg}+(G)$  such that,*

$$\mathbb{E}[\text{Alg}+(G)] \geq \left( \alpha_{\text{GW}} + \frac{\delta}{2} \left( t_c + \frac{(t_c - 3/4)\mu(G)}{9} \right) \left( \frac{\frac{(t_c - 3/4)\mu(G)}{9}}{1 - t_c - \frac{(t_c - 3/4)\mu(G)}{9}} \right) \right) \text{opt}_{\text{MC}}.$$

where the running time of  $\text{Alg}+(G)$  is  $O(|E| \log^2 |V| \text{poly}(1/(\delta\mu^2(G))))$ .

### 3.1 Improved Approximation Ratio Analysis

Towards proving Theorem 18, we start with a few useful definitions,

► **Definition 19** (Good and Bad Edges). *For a given feasible solution to the MAXCUT SDP and  $\varepsilon', \zeta' > 0$ , we say that an edge  $e = \{i, j\}$  is an  $(\varepsilon', \zeta')$ -bad edge if  $t_e \in [t_c - \zeta'\varepsilon', t_c + \varepsilon']$ . We say an edge  $e$  is an  $(\varepsilon', \zeta')$ -good edge if it is not  $(\varepsilon', \zeta')$ -bad. We let  $E_g(\varepsilon', \zeta')$  denote the set of  $(\varepsilon', \zeta')$ -good edges and  $E_b(\varepsilon', \zeta')$  denote the set of  $(\varepsilon', \zeta')$ -bad edges. Further we let,  $E_g^{(l)}(\varepsilon', \zeta') = \{e \in E : t_e \leq t_c - \zeta'\varepsilon'\}$  and  $E_g^{(u)}(\varepsilon', \zeta') = E_g(\varepsilon', \zeta') \setminus E_g^{(l)}(\varepsilon', \zeta')$ .*

► **Definition 20** (Density Constants). *We let  $\zeta = ((t_c - 3/4)/\varepsilon) - 1/2$  (any arbitrary small constant  $c > 0$  instead of  $1/2$  would also suffice). For choice of  $\varepsilon' = \varepsilon, \zeta' = \zeta$  we let  $\kappa = |E_g^{(l)}(\varepsilon, \zeta)| / |E|$ , and we let  $\gamma = |E_g^{(u)}(\varepsilon, \zeta)| / |E|$ . We note that  $0 \leq \kappa, \gamma \leq 1$ .*

Recall that for  $\text{sdpopt} \geq (t_c - \varepsilon)|E|$ , our algorithm  $\text{Alg}(G)$  sets  $\phi = 0^\circ$  and consequently reduces to the [39] algorithm. As noted earlier, if  $\text{sdpopt} = t_c|E|$ , the analysis of [39] gives exactly  $\alpha_{\text{GW}}$ -approximation ratio (no improvement). We now present a more refined analysis of [39] that leverages the notion of good edges we defined above to show that,

► **Lemma 21.** *For a graph  $G = (V, E)$  and a feasible SDP solution to the MAXCUT SDP such that  $\text{sdpval}(G) \geq (t_c - \varepsilon)|E|$  for some  $\varepsilon > 0$ , there exists an algorithm  $\text{Alg}+(G)$  such that,*

$$\mathbb{E}_{\mathbf{g} \sim \mathcal{N}(0,1)^n}[\text{Alg}+(G)] \geq \alpha_{\text{GW}} \text{sdpval}(G) + \delta \sum_{e \in E_g^{(u)}(\varepsilon, \zeta)} \left( \frac{1}{2} (1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle) \right).$$

**Proof.** The proof follows along lines of proof of Theorem 9. For  $\text{sdpval}(G) \geq (t_c - \varepsilon)|E|$  for some  $\varepsilon > 0$  the algorithm sets  $\phi = 0^\circ$  and hence one obtains that,

$$\begin{aligned} \mathbb{E}_{\mathbf{g} \sim \mathcal{N}(0,1)^n}[\text{Alg}+(G)] &= \mathbb{E}_{\mathbf{g} \sim \mathcal{N}(0,1)^n}[|E(S_g, V \setminus S_g)|] = \mathbb{E}_{\mathbf{g} \sim \mathcal{N}(0,1)^n} \left[ \sum_{e \in E} \mathbf{1}_{e \in E(S_g, V \setminus S_g)} \right] \\ &= \sum_{e \in E} \mathbb{E}_{\mathbf{g} \sim \mathcal{N}(0,1)^n} [\mathbf{1}_{e \in E(S_g, V \setminus S_g)}] = \sum_{e \in E} \mathbb{P}_{\mathbf{g} \sim \mathcal{N}(0,1)^n} [e \in E(S_g, V \setminus S_g)] \\ &= \sum_{e \in E_g^{(u)}(\varepsilon, \zeta)} \mathbb{P}_{\mathbf{g} \sim \mathcal{N}(0,1)^n} [e \in E(S_g, V \setminus S_g)] + \sum_{e \in E \setminus E_g^{(u)}(\varepsilon, \zeta)} \mathbb{P}_{\mathbf{g} \sim \mathcal{N}(0,1)^n} [e \in E(S_g, V \setminus S_g)] \\ &\geq (\alpha_{\text{GW}} + \delta) \sum_{e \in E_g^{(u)}(\varepsilon, \zeta)} \left( \frac{1}{2} (1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle) \right) + (\alpha_{\text{GW}}) \sum_{e \in E \setminus E_g^{(u)}(\varepsilon, \zeta)} \left( \frac{1}{2} (1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle) \right) \\ &= \alpha_{\text{GW}} \sum_{e \in E} \left( \frac{1}{2} (1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle) \right) + \delta \sum_{e \in E_g^{(u)}(\varepsilon, \zeta)} \left( \frac{1}{2} (1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle) \right) \\ &= \alpha_{\text{GW}} \text{sdpval}(G) + \delta \sum_{e \in E_g^{(u)}(\varepsilon, \zeta)} \left( \frac{1}{2} (1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle) \right). \end{aligned}$$

where the inequality follows from Definition 16 and proof of Theorem 10. ◀

Above, we show that we obtain an improvement over  $\alpha_{\text{GW}}$  when the “SDP mass” on the good edges,  $E_g^{(u)}(\varepsilon, \zeta)$  is large. Next, we argue that rather than requiring a large “SDP mass”, it suffices to show that the fraction of good edges,  $|E_g^{(u)}(\varepsilon, \zeta)| / |E|$  is large.

► **Lemma 22.** *Given a graph  $G = (V, E)$ , and a feasible solution to the MAXCUT SDP one obtains that,*

$$\sum_{e \in E_g^{(u)}(\varepsilon, \zeta)} \left( \frac{1}{2} (1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle) \right) \geq \gamma (t_c + \varepsilon) \text{sdpval}(G) \text{ where } \gamma = |E_g^{(u)}(\varepsilon, \zeta)| / |E|.$$

**Proof.** Considering the contribution in the SDP objective from  $E_g^{(u)}(\varepsilon, \zeta)$  it follows that,

$$\begin{aligned} \sum_{e \in E_g^{(u)}(\varepsilon, \zeta)} \left( \frac{1}{2} (1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle) \right) &= \sum_{\{i, j\} \in E_g^{(u)}(\varepsilon, \zeta)} t_{ij} \\ &\geq |E_g^{(u)}(\varepsilon, \zeta)| \min_{\{i, j\} \in E_g^{(u)}(\varepsilon, \zeta)} \{t_{ij}\} \\ &\geq |E_g^{(u)}(\varepsilon, \zeta)| (t_c + \varepsilon) && \text{(Definition of } E_g^{(u)}(\varepsilon, \zeta) \text{)} \\ &\geq (t_c + \varepsilon) \gamma |E| && \text{(By Definition 20)} \\ &\geq (t_c + \varepsilon) \gamma \text{sdpval}(G) && \text{(sdpval}(G) \leq |E| \text{)}. \end{aligned}$$

◀

Now we present a crucial result, where we show that the other type of good edges  $E_g^{(l)}(\varepsilon, \zeta)$  can be lower bounded by  $\mu(G)/3$ , the edge-disjoint triangle density of our graph  $G$ . We examine the SDP vectors corresponding to the vertices of triangles, and use their geometry to infer a useful bound on the SDP contribution of the edges within each triangle.

► **Lemma 23.** *Given a graph  $G = (V, E)$ , and a feasible solution to the MAXCUT SDP one obtains that  $\kappa \geq \mu(G)/3$  where  $\kappa = |E_g^{(l)}(\varepsilon, \zeta)| / |E|$ .*

**Proof.** Let  $T = \{i, j, k\}$  be a triangle in the graph  $G$  and let  $\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k$  be the corresponding vectors for a solution to the MAXCUT SDP. Now, consider the plane spanned by  $\mathbf{v}_i, \mathbf{v}_j$  and  $\mathbf{v}_k$  and let  $\theta_{ij}, \theta_{jk}, \theta_{ik}$  be the corresponding angles between the respective vectors. Then we have that,

▷ **Claim 24.**  $\min \{\theta_{ij}, \theta_{ik}, \theta_{jk}\} \leq 2\pi/3$ .

**Proof.** The analysis proceeds by examining  $\|\mathbf{v}_i + \mathbf{v}_j + \mathbf{v}_k\|^2$  and noting that,

$$\begin{aligned} 0 &\leq \|\mathbf{v}_i + \mathbf{v}_j + \mathbf{v}_k\|^2 \\ &= \|\mathbf{v}_i\|^2 + \|\mathbf{v}_j\|^2 + \|\mathbf{v}_k\|^2 + 2(\langle \mathbf{v}_i, \mathbf{v}_j \rangle + \langle \mathbf{v}_j, \mathbf{v}_k \rangle + \langle \mathbf{v}_i, \mathbf{v}_k \rangle) \\ &= 3 + 2(\cos(\theta_{ij}) + \cos(\theta_{jk}) + \cos(\theta_{ik})) && \text{(By constraints of MAXCUT SDP)}. \end{aligned}$$

From the above it follows that  $\cos(\theta_{ij}) + \cos(\theta_{jk}) + \cos(\theta_{ik}) \geq -3/2$ . Since the maximum of three real numbers at least their average, one obtains that,

$$\max \{\cos(\theta_{ij}), \cos(\theta_{ik}), \cos(\theta_{jk})\} \geq -1/2 \text{ and hence } \min \{\theta_{ij}, \theta_{ik}, \theta_{jk}\} \leq 2\pi/3 \quad \triangleleft$$

## 27:14 Triangles Improve 0.878 Approximation for Maxcut

Now, at least one of the edges in the triangles (without loss of generality, say the edge  $\{i, j\}$ ) is such that  $\theta_{ij} \leq 2\pi/3 = 120^\circ$  and hence for this edge it follows that,

$$t_{ij} \stackrel{\text{def}}{=} \frac{1}{2} (1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle) = \frac{1}{2} (1 - \cos \theta_{ij}) \leq \frac{1}{2} (1 - \cos(2\pi/3)) = \frac{3}{4}.$$

Now for the given choice of  $\zeta = (t_c - 3/4)/\varepsilon - 1/2$  it is easy to see that for choice of  $\varepsilon$  in Equation (3),

$$\zeta = \frac{t_c - 3/4}{\varepsilon} - \frac{1}{2} = \frac{9}{\mu(G)} - \frac{1}{2} \geq 9 - \frac{1}{2} > 0$$

where above uses  $\mu(G) \leq 1$  and further one obtains that,

$$t_c - \varepsilon\zeta = t_c - \varepsilon \left( \frac{t_c - 3/4}{\varepsilon} - \frac{1}{2} \right) = t_c - (t_c - 3/4) + \frac{\varepsilon}{2} > \frac{3}{4}$$

Therefore one concludes that  $\{i, j\} \in E_g^{(l)}(\varepsilon, \zeta)$  and hence,

$$\kappa = \frac{|E_g^{(l)}(\varepsilon, \zeta)|}{|E|} \geq \frac{\Delta(G)}{|E|} = \frac{\mu(G)}{3}. \quad \blacktriangleleft$$

We now revisit the object of our interest, the type of good edges  $E_g^{(u)}(\varepsilon, \zeta)$ . Next, we show through an averaging argument that for  $\text{sdpopt} \geq (t_c - \varepsilon)|E|$ , a large number of good edges of other type  $E_g^{(l)}(\varepsilon, \zeta)$  imply a large number of edges of type  $E_g^{(u)}(\varepsilon, \zeta)$ .

► **Lemma 25.** *Given a graph  $G = (V, E)$ , a feasible solution to the MAXCUT SDP such that  $\text{sdpval}(G) \geq (t_c - \varepsilon)|E|$  one obtains that,*

$$\gamma \geq \frac{\varepsilon}{(1 - t_c - \varepsilon)}, \text{ where recall that } \gamma = |E_g^{(u)}(\varepsilon, \zeta)| / |E|.$$

**Proof.** Let  $T$  be a uniform random variable that takes values  $(t_e)_{e \in E}$ , then it is given that,

$$\begin{aligned} \mathbb{E}[T] &= \frac{1}{|E|} \sum_{e \in E} t_e \geq \frac{1}{|E|} (t_c - \varepsilon) |E| \geq t_c - \varepsilon \text{ and} \\ \mathbb{P}[T \leq t_c - \zeta\varepsilon] &= \frac{|E_g^{(l)}(\varepsilon, \zeta)|}{|E|} = \frac{\kappa |E|}{|E|} = \kappa \end{aligned}$$

Note that  $T \in [0, 1]$  and it follows that  $\mathbb{P}[T \geq (t_c + \varepsilon)] = \gamma$  and hence one obtains that,

$$\begin{aligned} \mathbb{E}[T] &= \frac{1}{|E|} \left( \sum_{\substack{e \in E \\ t_e \leq t_c - \zeta\varepsilon}} t_e + \sum_{\substack{e \in E \\ t_c - \zeta\varepsilon \leq t_e \leq t_c + \varepsilon}} t_e + \sum_{\substack{e \in E \\ t_e \geq t_c + \varepsilon}} t_e \right) \\ &\leq (t_c - \zeta\varepsilon) \mathbb{P}[T \leq t_c - \zeta\varepsilon] + (t_c + \varepsilon) \mathbb{P}[t_c - \zeta\varepsilon \leq T \leq t_c + \varepsilon] + 1 \cdot \mathbb{P}[T \geq t_c + \varepsilon] \\ &\leq (t_c - \zeta\varepsilon) \kappa + (t_c + \varepsilon) (1 - \kappa - \gamma) + \gamma \end{aligned}$$

Now using  $\mathbb{E}[T] \geq t_c - \varepsilon$  and by expanding the above expression it follows that,

$$t_c - \varepsilon \leq t_c \kappa - \zeta\varepsilon \kappa + t_c + \varepsilon - t_c \kappa - t_c \gamma - \varepsilon \kappa - \varepsilon \gamma + \gamma$$

and rearranging terms in the above expression one obtains that,

$$(1 - t_c - \varepsilon) \gamma \geq \zeta \varepsilon \kappa + \varepsilon \kappa - 2\varepsilon = ((\zeta + 1) \kappa - 2) \varepsilon$$

and we one can then solve for  $\gamma$  by noting that for choice of  $\zeta$  from Definition 20,

$$\zeta + 1 = \left( \frac{t_c - 3/4}{\varepsilon} - \frac{1}{2} \right) + 1 \geq \frac{t_c - 3/4}{\varepsilon} \geq (t_c - 3/4) \frac{9}{(t_c - 3/4)\mu(G)} = \frac{9}{\mu(G)}$$

where the second inequality follows by choice of  $\varepsilon$  in Equation (3). Now solving for  $\gamma$ ,

$$\gamma \geq \frac{((\zeta + 1) \kappa - 2) \varepsilon}{1 - t_c - \varepsilon} \geq \frac{((9/\mu(G)) \kappa - 2) \varepsilon}{1 - t_c - \varepsilon} \geq \frac{(3 - 2) \varepsilon}{1 - t_c - \varepsilon} \geq \frac{\varepsilon}{1 - t_c - \varepsilon}$$

where the last inequality follows from Lemma 23 since  $\kappa \geq \mu(G)/3$ . ◀

### 3.2 Running Time Analysis

We recall our discussion of fast SDP solvers for MAXCUT SDP. The work of [68] (building on the works of [3, 65]) showed that for CSPs such as MAXCUT, we can obtain an approximate solution to their canonical SDP relaxation in near-linear time. We state the main result in [68] in the context of solving the MAXCUT SDP.

► **Theorem 26** (Theorem 1.1 in [68]). *Given a graph  $G = (V, E)$  and a constant  $v > 0$ , there exists an algorithm that computes in time  $O(|E| \log^2 |V| \text{poly}(1/v))$ , a subgraph  $G' = (V, E')$  and a feasible solution to MAXCUT SDP with value  $\text{sdpval}(G')$  such that,*

$$\blacksquare \quad \text{sdpval}(G') \geq \text{sdpopt}(G) - v |E| \quad \blacksquare \quad |E \setminus E'| \leq v |E|.$$

The fast SDP solver of [68] finds an approximately optimal solution using the Matrix Multiplicative Weights Update framework developed in [3] that does not construct a full  $n \times n$  Gram matrix but a low-rank approximation where instead it produces the solution vectors  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\} \in \mathbb{R}^k$  for  $k = \mathcal{O}(\text{poly}(\log n))$ . Next the angle computation for the outward rotation angle  $\phi$  can be done in  $\mathcal{O}(n)$  time. The vectors after rotation lie in space  $\mathbb{R}^{n+k}$ , but we do not need to explicitly compute these vectors. Instead we proceed to show that the [39] rounding can be more directly applied. We recall first that in [76] the rotated vectors  $\mathbf{w}_i$  can be written as,

$$\mathbf{w}_i = (\cos \phi) \mathbf{v}_i + (\sin \phi) \mathbf{u}_i$$

where  $\mathbf{u}_i$  is simply the  $i^{\text{th}}$  standard basis vector of  $\mathbb{R}^n$ . Finally we see that the last step is a plain [39] rounding which requires computing an inner product with a random Gaussian vector  $\mathbf{g} \sim \mathcal{N}(0, 1)^{n+k}$ . Now we can decompose  $\mathbf{g}$  as  $\mathbf{g} = \mathbf{g}_v + \mathbf{g}_u$  where  $\mathbf{g}_v \in \mathbb{R}^k$  and  $\mathbf{g}_u \in \mathbb{R}^n$  and  $\mathbf{g}_v \perp \mathbf{g}_u$ . Then we note that the inner product computation is simply,

$$\begin{aligned} \langle \mathbf{w}_i, \mathbf{g} \rangle &= \langle (\cos \phi) \mathbf{v}_i + (\sin \phi) \mathbf{u}_i, \mathbf{g}_v + \mathbf{g}_u \rangle \\ &= (\cos \phi) \langle \mathbf{v}_i, \mathbf{g}_v \rangle + (\sin \phi) \langle \mathbf{u}_i, \mathbf{g}_u \rangle \\ &= (\cos \phi) \langle \mathbf{v}_i, \mathbf{g}_v \rangle + (\sin \phi) g_u(i) \end{aligned} \tag{4}$$

Since  $\mathbf{v}_i$  and  $\mathbf{g}_v$  are vectors in  $\mathbb{R}^k$ , we can compute  $\langle \mathbf{w}_i, \mathbf{g} \rangle$  in  $\mathcal{O}(k)$  time. Thus the overall running time is dominated by the time required by the fast SDP solver. We restate the algorithm for clarity.



## 27:16 Triangles Improve 0.878 Approximation for Maxcut

**Input.** The algorithm takes a graph  $G = (V, E)$  as input.

**Algorithm.** We present the final algorithm denoted  $\text{Alg}+(G)$  below,

- Solve the MAXCUT SDP using fast SDP solver of [68] to obtain a subgraph  $G' = (V, E')$  and a collection of vectors  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$  that are feasible for it.
- An angle  $\phi$  is obtained by solving the below system of equations in variables  $r, t$ ,

$$\frac{\arccos(r(1-2t)) - \arccos(r)}{t} = \frac{2r}{\sqrt{1-r^2(1-2t)^2}},$$

$$\frac{1 - \frac{t|E|}{\text{sdpval}(G')}}{\sqrt{1-r^2}} = \frac{1-2t}{\sqrt{1-r^2(1-2t)^2}}$$

and letting  $\phi = \arccos(\sqrt{r})$ . If  $\text{sdpval}(G') \geq (t_c - \varepsilon)|E|$ , set  $\phi = 0^\circ$ .

- Run the rounding algorithm of [39] on the  $\{\mathbf{w}_i\}_{i \in [n]}$  vectors using computation as in Equation (4).

Now we have all the ingredients to prove our main result in this section where we show formal guarantees on the performance of  $\text{Alg}+(G)$ .

**Proof of Theorem 18.** For the choice of  $\varepsilon$  in Equation (3) observe when  $\text{sdpval}(G') \geq (t_c - \varepsilon)|E|$  it follows that,

$$\text{sdpopt}(G) \geq \text{sdpval}(G') \geq (t_c - \varepsilon)|E|.$$

If  $\text{sdpval}(G') \leq (t_c - \varepsilon)|E|$ , then from Theorem 26 with choice of  $v = 2\varepsilon$  one obtains that,

$$\text{sdpopt}(G) \leq \text{sdpval}(G') + v|E| \leq (t_c - \varepsilon)|E| + v|E| = (t_c + \varepsilon)|E|$$

Therefore, it follows that  $\text{sdpopt}(G) \notin [(t_c - \varepsilon)|E|, (t_c + \varepsilon)|E|]$  and using Corollary 17 one obtains the following,

$$\mathbb{E}[\text{Alg}+(G)] \geq (\alpha_{\text{GW}} + \delta) \text{sdpval}(G) \geq (\alpha_{\text{GW}} + \delta) \text{opt}_{\text{MC}}.$$

Next, consider the regimes where  $\text{sdpval}(G') \in [(t_c - \varepsilon)|E|, (t_c + \varepsilon)|E|]$ . For choice of  $\zeta = ((t_c - 3/4)/\varepsilon) - 1/2$  and using Lemma 25 one obtains that,

$$\gamma \geq \frac{\varepsilon}{(1 - t_c - \varepsilon)}$$

Using this in Lemma 22 and the bound obtained there further in Lemma 21 one obtains that,

$$\begin{aligned} \mathbb{E}[\text{Alg}+(G)] &\geq \alpha_{\text{GW}} \text{sdpval}(G') + \delta\gamma(t_c + \varepsilon) \text{sdpval}(G') \\ &\geq \alpha_{\text{GW}} \text{sdpval}(G') + \delta \left( \frac{\varepsilon}{(1 - t_c - \varepsilon)} \right) (t_c + \varepsilon) \text{sdpval}(G') \\ &= \left( \alpha_{\text{GW}} + \delta(t_c + \varepsilon) \left( \frac{\varepsilon}{1 - t_c - \varepsilon} \right) \right) \text{sdpval}(G') \end{aligned}$$

Putting everything together we have that regardless of the value of  $\text{sdpval}(G')$ ,

$$\mathbb{E}[\text{Alg}+(G)] \geq \left( \alpha_{\text{GW}} + \min \left\{ \delta, \delta(t_c + \varepsilon) \left( \frac{\varepsilon}{1 - t_c - \varepsilon} \right) \right\} \right) \text{sdpval}(G').$$

Note that for choice of  $\varepsilon$  in Equation (3) and using  $\mu(G) \leq 1$  it follows that for  $t_c \approx 0.84458$ ,

$$t_c + \varepsilon \leq t_c + \frac{t_c - 3/4}{9} = \frac{10t_c}{9} - \frac{1}{12} < 1, \quad \frac{\varepsilon}{1 - t_c - \varepsilon} \leq \frac{\frac{t_c}{9} - \frac{1}{12}}{1 - t_c - (\frac{t_c}{9} - \frac{1}{12})} = \frac{4t_c - 3}{39 - 40t_c} < 1$$

and hence one obtains that,

$$\min \left\{ \delta, \delta(t_c + \varepsilon) \left( \frac{\varepsilon}{1 - t_c - \varepsilon} \right) \right\} = \delta(t_c + \varepsilon) \left( \frac{\varepsilon}{1 - t_c - \varepsilon} \right)$$

Setting the value of  $\varepsilon$  from Equation (3) it follows that,

$$\mathbb{E}[\text{Alg}+(G)] \geq \left( \alpha_{\text{GW}} + \delta \left( t_c + \frac{(t_c - 3/4)\mu(G)}{9} \right) \left( \frac{\frac{(t_c - 3/4)\mu(G)}{9}}{1 - t_c - \frac{(t_c - 3/4)\mu(G)}{9}} \right) \right) \text{sdpval}(G').$$

Now to simplify the notation in rest of the analysis consider a definition,

► **Definition 27** (Final Improvement). Denote the final improvement (over  $\alpha_{\text{GW}}$ ) as  $\tau$  by letting,

$$\tau \stackrel{\text{def}}{=} \frac{\delta}{2} \left( t_c + \frac{(t_c - 3/4)\mu(G)}{9} \right) \left( \frac{\frac{(t_c - 3/4)\mu(G)}{9}}{1 - t_c - \frac{(t_c - 3/4)\mu(G)}{9}} \right).$$

Thus one notes that  $\mathbb{E}[\text{Alg}+(G)] \geq (\alpha_{\text{GW}} + 2\tau) \text{sdpval}(G')$ . Now using Theorem 26 for choice of  $v = \tau / (4\alpha_{\text{GW}} + 8\tau)$  it follows that,

$$\begin{aligned} \mathbb{E}[\text{Alg}+(G)] &\geq (\alpha_{\text{GW}} + 2\tau) \text{sdpval}(G') \\ &\geq (\alpha_{\text{GW}} + 2\tau) (\text{sdpopt}(G) - 2v |E|) \quad (\text{Using Theorem 26}) \\ &\geq (\alpha_{\text{GW}} + 2\tau) (1 - 4v) \text{sdpopt}(G) \quad (\text{Using } \text{sdpopt}(G) \geq \text{opt}(G) \geq |E|/2) \\ &= (\alpha_{\text{GW}} + \tau) \text{sdpopt}(G) \quad (\text{Using choice of } v) \\ &\geq (\alpha_{\text{GW}} + \tau) \text{opt}(G) \quad (\text{Since } \text{sdpopt}(G) \geq \text{opt}(G)). \end{aligned}$$

The running time is  $O(|E| \log^2 |V| \text{poly}(1/\tau))$  and it is easy to note from Definition 27 that  $\tau = \Omega(\delta\mu^2(G))$  and hence the running time is  $O(|E| \log^2 |V| \text{poly}(1/(\delta\mu^2(G))))$ . ◀

## 4 Applications of Improved Maxcut SDP Analysis

In this section, we apply Theorem 18 to derive improved approximation guarantees for MAX-CUT problem on some natural class of graphs. We show that  $(\alpha_{\text{GW}} + \Omega(1))$ -approximation is achievable by showing that  $\mu(G) = \Omega(1)$  indeed holds for these classes of graphs.

### 4.1 Geometric Intersection Graphs of Unit Balls

Given a set of geometric objects, a geometric intersection graph is constructed by representing each object as a vertex and placing an edge between two vertices if and only if the corresponding objects intersect.

► **Definition 28** (Unit Ball Graph in  $\mathbb{R}^k$ ). An undirected graph  $G = (V, E)$  is said to be a unit ball graph if there exists an embedding  $\varphi : V \rightarrow \mathbb{R}^k$  such that for every vertex  $u \in V$ , we associate a unit radius ball centered at  $\varphi(u)$ , and for all  $u, v \in V$  we have,

$$\{u, v\} \in E \iff \|\varphi(u) - \varphi(v)\|_2 \leq 2 \quad \text{i.e., an edge exists iff the vertex balls intersect.}$$

► **Remark 29.** Since recognizing whether a graph (given in standard representation such as adjacency list or matrix), is a unit ball graph is NP-hard (even for  $k = 2$ , see [20]), one assumes that the mapping  $\varphi$  is given explicitly as part of the input.

## 27:18 Triangles Improve 0.878 Approximation for Maxcut

Our goal in this section is to show that any unit ball graph with sufficiently high average degree satisfies  $\mu(G) = \Omega(1)$  by exhibiting that it has  $\Delta(G) = \Omega(|E|)$  many edge-disjoint triangles.

► **Definition 30** (Covering Number). *The covering number  $C(l, \theta)$  is the minimum number of spherical caps of angular radius  $\theta$  required to cover the entire sphere  $\mathbb{S}^{l-1}$ . We let  $N_l = C(l, \pi/6)$ , be the covering number with spherical caps of angular radius<sup>1</sup>  $\pi/6$ . Equivalently, in each such cap, the angle between any two points is at most  $\pi/3$ .*

► **Theorem 31.** *Let  $G = (V, E)$  be a unit ball graph in  $\mathbb{R}^k$  with average degree  $d_{\text{avg}} \geq 6N_k$ . Then  $G$  contains  $|E|/216N_k^2$  edge-disjoint triangles.*

► **Construction 32** (Spherical Cap Partitioning). *Fix a vertex  $v \in V$  of degree  $r = \deg(v)$ , and let the set  $S(v) = v \cup N(v)$ . Without loss of generality, translate the embedding so that  $\varphi(v) = O$  (the origin). For each neighbor  $u \in N(v)$  define:*

$$p_u = \varphi(u) \text{ and } \hat{p}_u = \frac{p_u}{\|p_u\|} \in \mathbb{S}^{k-1} \text{ that represent unit direction from } v \text{ to } u.$$

*Let  $\{C_1, C_2, \dots, C_{N_k}\}$  be a covering of  $\mathbb{S}^{k-1}$  by spherical caps of angular radius  $\pi/6$  where  $N_k = C(k, \pi/6)$ . We then group the neighbors of  $v$  according to which cap the direction falls into as,*

$$R_i(v) = \{u \in N(v) : \hat{p}_u \in C_i\}, \quad i \in \{1, 2, \dots, N_k\}$$

► **Fact 33.** *For  $l \in \mathbb{N}$  we define  $N_l \stackrel{\text{def}}{=} C(l, \pi/6)$ . Then for all dimensions  $l$ , the covering number  $N_l$  satisfies  $N_l \leq 2^{3l}$ .*

**Proof.** We consider a proof by the connection to packing number. We define the packing number as,

► **Definition 34** (Packing Number). *The packing number, denoted  $P(l, \theta)$  is the maximum number of points that can be placed on the unit sphere  $\mathbb{S}^{l-1}$  such that the angular distance between any pair of points is at least  $\theta$ .*

► **Remark 35.** The kissing number  $\kappa_l$  in  $\mathbb{R}^l$  is defined as the maximum number of non-overlapping unit balls that can simultaneously touch a central unit ball. Equivalently  $\kappa_l = P(l, \pi/3)$  since the angle between any two such directions must be at least  $\pi/3$ .

► **Fact 36** (Covering-Packing Duality, Lemma 5.5 in [72]). *For all dimensions  $l$ , the packing number and covering number satisfy the following relation,*

$$P(l, \theta) \leq C(l, \theta) \leq P(l, \theta/2).$$

Now using the general bounds in the works [51, 26] and using the Fact 36 one notes that,

$$N_l = C(l, \pi/6) \leq P(l, \pi/12) \leq (7.67)^{l-2} \leq 2^{3l}$$

where the last inequality is easy to check, and this finishes the proof. ◀

▷ **Claim 37.** Let the vertices  $x, y, z \in R_i(v)$  for some  $i \in \{1, 2, \dots, N_k\}$ . Then it follows that  $\{x, y, z\}$  forms a triangle.

---

<sup>1</sup> A spherical cap of angular radius  $\theta$  is set of points on surface of sphere with angular distance from center at most  $\theta$ .

Proof. By construction, all three vertices  $x, y, z$  lie in the same spherical cap  $\mathcal{C}_i$  of angular radius  $\pi/6$ . Therefore the angular distance between any two direction vectors is at most  $\pi/3$ . In particular,

$$\begin{aligned}\|p_x - p_y\|^2 &= \|p_x\|_2^2 + \|p_y\|_2^2 - 2p_x^T p_y = \|p_x\|_2^2 + \|p_y\|_2^2 - 2\|p_x\|_2 \|p_y\|_2 \langle \hat{p}_x, \hat{p}_y \rangle \\ &\leq \|p_x\|_2^2 + \|p_y\|_2^2 - \|p_x\|_2 \|p_y\|_2 \leq \|p_x\|_2^2 + \|p_y\|_2^2 - \min \left\{ \|p_x\|_2^2, \|p_y\|_2^2 \right\} \\ &= \max \left\{ \|p_x\|_2^2, \|p_y\|_2^2 \right\} = 4\end{aligned}$$

and hence  $\|p_x - p_y\| \leq 2$  and  $\{x, y\} \in E$ . The same argument applies to pairs  $\{y, z\}$  and  $\{x, z\}$ , and hence all three edges  $\{x, y\}, \{y, z\}, \{x, z\}$  are present. Therefore,  $\{x, y, z\}$  forms a triangle.  $\triangleleft$

If  $n = 6x + 1$  or  $n = 6x + 3$ , one gets a decomposition of complete graph  $K_n$  into edge-disjoint triangles, the number of edge-disjoint triangles is exactly  $\binom{n}{2}/3$ , a perfect packing of edge-disjoint triangles. We consider the following result that tells the maximum number of edge-disjoint triangles that can be packed into  $K_n$  when  $n$  is not of the form  $6x + 1$  or  $6x + 3$ ,

► **Fact 38** (Theorem 2, [33]). *Given a complete graph  $K_n$  where  $|V| = n = 6x + i, 0 \leq i \leq 5$ ,*

$$\Delta(K_n) = \begin{cases} \frac{\binom{n}{2} - \frac{n}{2}}{3} & \text{if } i = 0, 2 \\ \frac{\binom{n}{2}}{3} & \text{if } i = 1, 3 \\ \frac{\binom{n}{2} - \frac{n}{2} - 1}{3} & \text{if } i = 4 \\ \frac{\binom{n}{2} - 4}{3} & \text{if } i = 5, \end{cases}, \text{ thus a uniform lower bound, } \Delta(K_n) \geq \frac{\binom{n}{2} - \frac{n}{2} - 1}{3}.$$

► **Lemma 39.** *Let  $G = (V, E)$  be a unit ball graph in  $\mathbb{R}^k$ . Fix a vertex  $v \in V$  of degree  $d \geq 3N_k$  and let  $S(v) = \{v\} \cup N(v)$ . Then the induced subgraph  $G[S(v)]$  has at least  $d^2/54N_k^2$  edge-disjoint triangles.*

**Proof.** Recall Construction 32 for vertex  $v$  where we partition  $\mathbb{S}^{k-1}$  into  $N_k$  spherical caps of angular radius  $\pi/6$ . By a simple averaging argument, some cap (say  $\mathcal{C}_i$ ) contains at least  $\lfloor d/N_k \rfloor$  neighbors of  $v$ . Including  $v$  this gives us a set of  $n' = \lfloor d/N_k \rfloor + 1$  vertices all lying in the same region  $R_i(v)$ . By Claim 37, the subgraph induced by this set forms a clique of size  $n' = \lfloor d/N_k \rfloor + 1$ . Using Fact 38 the number of edge disjoint triangles in a clique of size  $n'$  is at least,

$$\frac{\binom{n'}{2} - (n'/2 + 1)}{3} = \frac{n'^2 - 2n' - 2}{6}$$

Now let  $C$  be a constant such that  $d \geq CN_k$ . Since  $n' = \lfloor d/N_k \rfloor + 1 \geq d/N_k$  it follows that,

$$\frac{n'^2 - 2n' - 2}{6} \geq \left( \frac{1 - 2/C - 2/C^2}{6} \right) \frac{d^2}{N_k^2}$$

The coefficient above is non-negative when  $1 - 2/C - 2/C^2 \geq 0$  which holds for  $C \geq 1 + \sqrt{3} \approx 2.732$ . For simplicity taking  $C = 3$  gives a valid lower bound whenever  $d \geq 3N_k$  and one obtains that the number of edge-disjoint triangles in  $G[S(v)]$  is at least,

$$\left( \frac{1 - 2/3 - 2/9}{6} \right) \frac{d^2}{N_k^2} = \frac{d^2}{54N_k^2}. \quad \blacktriangleleft$$

**Proof of Theorem 31.** We will give a constructive proof for this by extracting edge-disjoint triangles greedily as below,

**Algorithm.** Initialize  $G_0 = G$ ,  $m_0 = m = |E|$ , and initially empty set of edge disjoint triangles  $\mathcal{T} \leftarrow \emptyset$ . For rounds  $i = 1, 2, \dots$ , do the following:

1. If  $\max_{v \in V(G_{i-1})} \deg_{G_{i-1}}(v) \leq 3N_k$ , then terminate.
2. Otherwise choose a vertex  $v_i$  in  $G_{i-1}$  with maximum degree where  $d_i \geq 3N_k$ .
3. Apply Lemma 39 to extract edge disjoint triangles  $\mathcal{T}_i$  from  $G_{i-1}[S(v_i)]$  and add to  $\mathcal{T}$ .
4. Delete all vertices in  $S(v_i)$  and their incident edges to form a graph  $G_i$ . Let  $m_i = |E(G_i)|$ .

It is easy to see that when procedure terminates correctly. Each triangle in  $\mathcal{T}$  is edge-disjoint by construction. Since all vertices and their incident edges in  $S(v_i)$  are removed in each round, no triangle in any subsequent round can share an edge with any triangle previously added. Thus,  $\mathcal{T}$  is a collection of edge-disjoint triangles.

In each round we remove  $m_{i-1} - m_i$  edges, which is at most  $(d_i + 1)d_i \leq 2d_i^2$  (since each vertex in  $S(v_i)$  has degree at most  $d_i$  in  $G_{i-1}$ ). By Lemma 39, we add at least  $d_i^2/(54N_k^2)$  triangles. Therefore, the number of triangles added satisfies:

$$|\mathcal{T}_i| \geq \frac{d_i^2}{54N_k^2} \geq \frac{m_{i-1} - m_i}{108N_k^2} \quad (5)$$

Let the procedure end after  $r$  rounds where one notes that  $r \geq 1$  since to start with the graph has average degree at least  $6N_k$  and hence,

$$m_0 = |E| \geq \frac{6N_k |V|}{2} = 3N_k |V| \quad (6)$$

Each round removes at least one vertex and hence  $|V(G_r)| \leq |V| - r$  and in  $G_r$  we have all degrees are smaller than  $3N_k$  and hence,

$$m_r \leq \frac{1}{2} (3N_k - 1) |V(G_r)| < \frac{3N_k}{2} (|V| - r) < \frac{3N_k |V|}{2} \quad (7)$$

and using Equation (6) one concludes that  $m_r \leq m_0/2$ .

Finally, summing over all the rounds and using Equation (5) the number of edge-disjoint triangles,

$$|\mathcal{T}| = \sum_{i=1}^r |\mathcal{T}_i| \geq \sum_{i=1}^r \left( \frac{m_{i-1} - m_i}{108N_k^2} \right) = \frac{m_0 - m_r}{108N_k^2} \geq \frac{m_0}{216N_k^2} = \frac{|E|}{216N_k^2}. \quad \blacktriangleleft$$

## 4.2 Real-World Network Graphs

While the performance of MAXCUT SDP on random Erdős–Rényi  $G(n, p)$  graphs has received considerable attention, with several works [27, 62, 38, 61] showing that  $\alpha_{\text{GW}}$ -approximation ratio can be significantly improved. However, Erdős–Rényi graphs are often too simplified of model that fail to capture many key features of real-world networks. This includes the presence of community structure, heavy-tailed degree distributions, small diameter, and high clustering coefficients. There are various models proposed that address these shortcomings, the most popular ones being the Preferential Attachment Model due to [8] and the small-world network model due to [74]. More recently frameworks such as the combinatorial triangle-dense decompositions by [40] and the “spectral triadic decomposition” by [12], emphasize the role of triangles and higher-order motifs in capturing the nuanced community structure of real-world networks.

While one might expect that tailored algorithms could yield much better than  $\alpha_{\text{GW}}$ -approximation in such models by exploiting the structure in such graphs. However, we show here that one can use the result from our SDP analysis in a black box fashion to show an  $(\alpha_{\text{GW}} + \Omega(1))$ -approximation on such real-world network graphs.

#### 4.2.1 Spectral Triadic Decomposition Framework

The spectral triadic decomposition proposed by [12] provides an analytical model for studying the community structure of real-world networks. They define a spectral measure called “spectral transitivity” denoted  $\tau(G)$ , which quantifies the pervulence of triangles in real-world graphs. They show that a large value for this measure allows for a graph decomposition into multiple dense “clique like” structures capturing a stylized form of community structure observed in real-world networks.

► **Definition 40** (Spectral Transitivity, Definition 1.1 in [12]). *For an undirected graph  $G$  with a normalized adjacency matrix  $\mathcal{A}$ , the spectral transitivity denoted  $\tau(G)$  is defined as,*

$$\tau(G) = \frac{\sum_{i=1}^n \lambda_i^3}{\sum_{i=1}^n \lambda_i^2}.$$

where  $\lambda_1, \lambda_2, \dots, \lambda_n$  are eigenvalues of  $\mathcal{A}$ .

► **Remark 41.** This measure is a reweighted version of transitivity notion from [40]. For simplicity we restrict to  $d$ -regular graphs where  $\mathcal{A} = A/d$ . For an edge  $\{u, v\}$  the weight is defined as  $1/d_u d_v = 1/d^2$  and for a triangle  $\{u, v, w\}$  it is defined to be  $1/d_u d_v d_w = 1/d^3$

► **Lemma 42** (Lemma 3.5 in [12]). *For a  $d$ -regular graph  $G = (V, E)$  having  $T(G)$  many triangles, the spectral transitivity is,*

$$\tau(G) = \frac{3 \sum_{t \in T} w(t)}{\sum_{e \in E} w(e)} = \frac{3T(G)/d^3}{|E|/d^2} = \frac{3T(G)}{|E|d}.$$

► **Lemma 43.** *Let  $G$  be a  $d$ -regular graphs with  $d \geq 3$ , and  $\tau(G) \geq c$  for a constant  $c > 0$ , then  $\tau(G) \geq c/3$ .*

**Proof.** Consider the quantity  $t_e = |t \in T(G) : e \in t|$  which calculates the number of triangles containing edge  $e$ , and let  $t^* = \max_{e \in E} t_e$ . In a  $d$ -regular graph it is easy to see that  $t_e \leq d-1$  and hence  $t^* \leq d-1$ . Now, we greedily pick a triangle and remove its three edges from the graph. The total number of triangles in the graph can go down by at most  $3t^*$ . Thus we have that total number of edge-disjoint triangles is at least,

$$\Delta(G) \geq \frac{T(G)}{1 + 3t^*} \geq \frac{T(G)}{1 + 3(d-1)} \geq \frac{T(G)}{3d-2}$$

and using  $d \geq 3$  and using Lemma 42 we have that,

$$\Delta(G) \geq \frac{\tau(G)|E|d}{3(3d-2)} \geq \frac{c|E|}{9} \text{ and hence } \tau(G) = \frac{3\Delta(G)}{|E|} \geq \frac{c}{3}. \quad \blacktriangleleft$$

## References

- 1 Ranendu Adhikary, Kaustav Bose, Satwik Mukherjee, and Bodhayan Roy. Complexity of maximum cut on interval graphs. In *37th International Symposium on Computational Geometry (SoCG)*, volume 189, pages 7:1–7:11, 2021. doi:10.4230/LIPICS.SOCG.2021.7.
- 2 Noga Alon and Assaf Naor. Approximating the cut-norm via Grothendieck’s inequality. *SIAM J. Comput.*, 35(4):787–803, 2006. doi:10.1137/S0097539704441629.
- 3 Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. *J. ACM*, 63(2):Art. 12, 35, 2016. doi:10.1145/2837020.
- 4 Sanjeev Arora, David Karger, and Marek Karpinski. Polynomial time approximation schemes for dense instances of NP-hard problems. *J. Comput. System Sci.*, 58(1):193–210, 1999. doi:10.1006/jcss.1998.1605.
- 5 Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56(2):Art. 5, 37, 2009. doi:10.1145/1502793.1502794.
- 6 Per Austrin, Siavosh Benabbas, and Konstantinos Georgiou. Better balance by being biased: a 0.8776-approximation for Max Bisection. *ACM Trans. Algorithms*, 13(1):Art. 2, 27, 2016. doi:10.1145/2907052.
- 7 R. Bar-Yehuda, M. M. Halldórsson, J. Naor, H. Shachnai, and I. Shapira. Scheduling split intervals. *SIAM J. Comput.*, 36(1):1–15, 2006. doi:10.1137/S0097539703437843.
- 8 Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999. doi:10.1126/science.286.5439.509.
- 9 Francisco Barahona, Martin Grötschel, Michael Jünger, and Gerhard Reinelt. An application of combinatorial optimization to statistical physics and circuit layout design. *Oper. Res.*, 36(3):493–513, June 1988. doi:10.1287/OPRE.36.3.493.
- 10 Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science—FOCS 2011*, pages 472–481. IEEE Computer Soc., Los Alamitos, CA, 2011. doi:10.1109/FOCS.2011.95.
- 11 Alexey Barsukov and Bodhayan Roy. Maximum cut on interval graphs of interval count two is np-complete, 2024. arXiv:2203.06630.
- 12 Sabyasachi Basu, Suman Kalyan Bera, and C. Seshadhri. Spectral triadic decompositions of real-world networks. *SIAM J. Math. Data Sci.*, 6(3):703–730, 2024. doi:10.1137/23M1586926.
- 13 Piotr Berman and Marek Karpinski. On some tighter inapproximability results (extended abstract). In *icalp*, pages 200–209, 1999. doi:10.1007/3-540-48523-6\_17.
- 14 Hans L. Bodlaender, Celina M. H. de Figueiredo, Marisa Gutierrez, Ton Kloks, and Rolf Niedermeier. Simple max-cut for split-indifference graphs and graphs with few p4’s. In *Experimental and Efficient Algorithms*, pages 87–99, 2004. doi:10.1007/978-3-540-24838-5\_7.
- 15 Hans L. Bodlaender and Klaus Jansen. On the complexity of the maximum cut problem. *Nordic Journal of Computing*, 7(1):14–31, 2000.
- 16 Hans L. Bodlaender, Ton Kloks, and Rolf Niedermeier. Simple max-cut for unit interval graphs and graphs with few p4s. *Electronic Notes in Discrete Mathematics*, 3:19–26, 1999. doi:10.1016/S1571-0653(05)80014-9.
- 17 Kellogg S. Booth and George S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *J. Comput. System Sci.*, 13(3):335–379, 1976. doi:10.1016/S0022-0000(76)80045-1.
- 18 Arman Boyacı, Tinaz Ekim, and Mordechai Shalom. A polynomial-time algorithm for the maximum cardinality cut problem in proper interval graphs. *Information Processing Letters*, 121:29–33, 2017. doi:10.1016/j.ipl.2017.01.007.
- 19 Arman Boyacı, Tinaz Ekim, and Mordechai Shalom. The maximum cardinality cut problem in co-bipartite chain graphs. *Journal of Combinatorial Optimization*, 35(1):250–265, 2018. doi:10.1007/s10878-015-9963-x.
- 20 Heinz Breu and David G. Kirkpatrick. Unit disk graph recognition is NP-hard. *Comput. Geom.*, 9(1-2):3–24, 1998. doi:10.1016/S0925-7721(97)00014-X.



- 21 Martin C. Carlisle and Errol L. Lloyd. On the  $k$ -coloring of intervals. *Discrete Appl. Math.*, 59(3):225–235, 1995. doi:10.1016/0166-218X(93)E0174-W.
- 22 Charles Carlson, Alexandra Kolla, Ray Li, Nitya Mani, Benny Sudakov, and Luca Trevisan. Lower bounds for max-cut in  $H$ -free graphs via semidefinite programming. *SIAM J. Discrete Math.*, 35(3):1557–1568, 2021. doi:10.1137/20M1333985.
- 23 K.C. Chang and D.H.-C. Du. Efficient algorithms for layer assignment problem. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 6(1):67–78, 1987. doi:10.1109/TCAD.1987.1270247.
- 24 Maw-Shang Chang. Efficient algorithms for the domination problems on interval and circular-arc graphs. *SIAM J. Comput.*, 27(6):1671–1694, December 1998. doi:10.1137/S0097539792238431.
- 25 Moses Charikar and Anthony Wirth. Maximizing quadratic programs: Extending grothendieck’s inequality. In *focs*, pages 54–60, 2004. doi:10.1109/F0CS.2004.39.
- 26 John H. Conway and Neil J. A. Sloane. *Sphere Packings, Lattices and Groups*, volume 290 of *Grundlehren der mathematischen Wissenschaften*. Springer, 3rd edition, 1999.
- 27 Hervé Daudé, Conrado Martínez, Vojtya Rasendrasasina, and Vlady Ravelomanana. The MAX-CUT of sparse random graphs. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 265–271. ACM, New York, 2012. doi:10.1137/1.9781611973099.24.
- 28 Celina MH de Figueiredo, Alexsander A de Melo, Fabiano S Oliveira, and Ana Silva. Maximum cut on interval graphs of interval count four is np-complete. *arXiv preprint*, 2020. arXiv:2012.09804.
- 29 Charles Delorme and Svatopluk Poljak. Combinatorial properties and the complexity of a max-cut approximation. *European J. Combin.*, 14(4):313–333, 1993. doi:10.1006/eujc.1993.1035.
- 30 Josep Díaz and Marcin Kamiński. Max-cut and max-bisection are NP-hard on unit disk graphs. *Theoretical Computer Science*, 377(1):271–276, 2007. doi:10.1016/j.tcs.2007.02.013.
- 31 C. S. Edwards. An improved lower bound for the number of edges in a largest bipartite subgraph. In *Recent advances in graph theory (Proc. Second Czechoslovak Sympos., Prague, 1974)*, pages 167–181. Academia, Prague, 1975.
- 32 Pál Erdős. On even subgraphs of graphs. *Mat. Lapok*, 18:283–288, 1967.
- 33 Tomás Feder and Carlos S. Subi. Packing edge-disjoint triangles in given graphs. *Electron. Colloquium Comput. Complex.*, TR12-013, 2012. URL: <https://eccc.weizmann.ac.il/report/2012/013>.
- 34 Uriel Feige, Marek Karpinski, and Michael Langberg. Improved approximation of Max-Cut on graphs of bounded degree. *J. Algorithms*, 43(2):201–219, 2002. doi:10.1016/S0196-6774(02)00005-6.
- 35 Uriel Feige and Michael Langberg. The RPR<sup>2</sup> rounding technique for semidefinite programs. *J. Algorithms*, 60(1):1–23, 2006. doi:10.1016/j.jalgor.2004.11.003.
- 36 Uriel Feige and Gideon Schechtman. On the optimality of the random hyperplane rounding technique for MAX CUT. *Random Structures Algorithms*, 20(3):403–440, 2002. Probabilistic methods in combinatorial optimization. doi:10.1002/rsa.10036.
- 37 Mikael Florén. Approximation of max-cut on graphs of bounded degree, 2016.
- 38 David Gamarnik and Quan Li. On the max-cut of sparse random graphs. *Random Structures Algorithms*, 52(2):219–262, 2018. doi:10.1002/rsa.20738.
- 39 Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach.*, 42(6):1115–1145, 1995. doi:10.1145/227683.227684.
- 40 Rishi Gupta, Tim Roughgarden, and C. Seshadhri. Decompositions of triangle-dense graphs. *SIAM J. Comput.*, 45(2):197–215, 2016. doi:10.1137/140955331.
- 41 Venkatesan Guruswami. Maximum cut on line and total graphs. *Discrete Applied Mathematics*, 92(2):217–221, 1999. doi:10.1016/S0166-218X(99)00056-6.

- 42 Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graphs partitioning and quadratic integer programming with PSD objectives (extended abstract). In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science – FOCS 2011*, pages 482–491. IEEE Computer Soc., Los Alamitos, CA, 2011. doi:10.1109/FOCS.2011.36.
- 43 F. Hadlock. Finding a maximum cut of a planar graph in polynomial time. *SIAM Journal on Computing*, 4(3):221–225, 1975. doi:10.1137/0204019.
- 44 Jun-Ting Hsieh and Pravesh K. Kothari. Approximating max-cut on bounded degree graphs: tighter analysis of the FKL algorithm. In *50th International Colloquium on Automata, Languages, and Programming*, volume 261 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 77, 7. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ICALP.2023.77.
- 45 M.L. Huson and A. Sen. Broadcast scheduling algorithms for radio networks. In *Proceedings of MILCOM '95*, volume 2, pages 647–651 vol.2, 1995. doi:10.1109/MILCOM.1995.483546.
- 46 Johan Håstad. Some optimal inapproximability results. In *STOC '97 (El Paso, TX)*, pages 1–10. ACM, New York, 1999.
- 47 Hiroshi Imai and Takao Asano. Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane. *J. Algorithms*, 4(4):310–323, 1983. doi:10.1016/0196-6774(83)90012-3.
- 48 Klaus Jansen, Marek Karpinski, Andrzej Lingas, and Eike Seidel. Polynomial time approximation schemes for max-bisection on planar and geometric graphs. *SIAM J. Comput.*, 35(1):110–119, 2005. doi:10.1137/S009753970139567X.
- 49 David S Johnson. The NP-completeness column: an ongoing guide. *Journal of Algorithms*, 6(3):434–451, 1985. doi:10.1016/0196-6774(85)90012-4.
- 50 John R. Jungck, Gregg Dick, and Amy Gleason Dick. Computer-assisted sequencing, interval graphs, and molecular evolution. *Biosystems*, 15(3):259–273, 1982. doi:10.1016/0303-2647(82)90010-7.
- 51 G. A. Kabatiansky and V. I. Levenshtein. On bounds for packings on a sphere and in space. *Problems of Information Transmission*, 14(1):1–17, 1978. Originally in Russian: *Probl. Peredachi Inf.* 14(1):3–25.
- 52 Satyen Kale and C. Seshadhri. Combinatorial approximation algorithms for maxcut using random walks. In *Proceedings of 2nd Symposium on Innovations in Computer Science (ICS)*, pages 367–388, 2011. URL: <http://conference.iiis.tsinghua.edu.cn/ICS2011/content/papers/20.html>.
- 53 Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972)*, The IBM Research Symposia Series, pages 85–103. Plenum, New York-London, 1972. doi:10.1007/978-1-4684-2001-2\_9.
- 54 Ken-ichi Kawarabayashi, Mikkel Thorup, and Hirotaka Yoneda. Better coloring of 3-colorable graphs. In *STOC'24—Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 331–339. ACM, New York, 2024. doi:10.1145/3618260.3649768.
- 55 J. Mark Keil. Finding Hamiltonian circuits in interval graphs. *Inform. Process. Lett.*, 20(4):201–206, 1985. doi:10.1016/0020-0190(85)90050-X.
- 56 Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, pages 767–775. ACM, New York, 2002. doi:10.1145/509907.510017.
- 57 Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.*, 37(1):319–357, 2007. doi:10.1137/S0097539705447372.
- 58 Jan Kratochvíl, Tomáš Masařík, and Jana Novotná. U-Bubble Model for Mixed Unit Interval Graphs and Its Applications: The MaxCut Problem Revisited. In *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*, volume 170, pages 57:1–57:14, 2020. doi:10.4230/LIPIcs.MFCS.2020.57.

- 59 Madhav V. Marathe, R. Ravi, and C. Pandu Rangan. Generalized vertex covering in interval graphs. *Discrete Appl. Math.*, 39(1):87–93, 1992. doi:10.1016/0166-218X(92)90116-R.
- 60 Claire Mathieu and Warren Schudy. Yet another algorithm for dense max cut: go greedy. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 176–182. ACM, New York, 2008. URL: <http://dl.acm.org/citation.cfm?id=1347082.1347102>.
- 61 Renee Mirka and David P. Williamson. An experimental evaluation of semidefinite programming and spectral algorithms for max cut. *ACM J. Exp. Algorithmics*, 28, August 2023. doi:10.1145/3609426.
- 62 Andrea Montanari and Subhabrata Sen. Semidefinite programs on sparse random graphs and their application to community detection. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 814–827, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2897518.2897548.
- 63 Yurii Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Methods & Software*, 9:141–160, 1998. URL: <https://api.semanticscholar.org/CorpusID:121309892>.
- 64 Ryan O'Donnell and Yi Wu. An optimal SDP algorithm for max-cut, and equally optimal long code tests. In *STOC'08*, pages 335–344. ACM, New York, 2008. doi:10.1145/1374376.1374425.
- 65 Prasad Raghavendra and David Steurer. How to round any CSP. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2009*, pages 586–594. IEEE Computer Soc., Los Alamitos, CA, 2009. doi:10.1109/FOCS.2009.74.
- 66 Sartaj Sahni and Teofilo Gonzalez.  $P$ -complete approximation problems. *J. Assoc. Comput. Mach.*, 23(3):555–565, 1976. doi:10.1145/321958.321975.
- 67 José A. Soto. Improved analysis of a max-cut algorithm based on spectral partitioning. *SIAM J. Discrete Math.*, 29(1):259–268, 2015. doi:10.1137/14099098X.
- 68 David Steurer. Fast SDP algorithms for constraint satisfaction problems. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 684–697. SIAM, Philadelphia, PA, 2010. doi:10.1137/1.9781611973075.56.
- 69 Chaitanya Swamy. Correlation clustering: maximizing agreements via semidefinite programming. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 526–527. ACM, New York, 2004. URL: <http://dl.acm.org/citation.cfm?id=982792.982866>.
- 70 Luca Trevisan. Max cut and the smallest eigenvalue. *siamj*, 41(6):1769–1786, 2012. doi:10.1137/090773714.
- 71 Luca Trevisan, Gregory B. Sorkin, Madhu Sudan, and David P. Williamson. Gadgets, approximation, and linear programming. *SIAM J. Comput.*, 29(6):2074–2097, 2000. doi:10.1137/S0097539797328847.
- 72 Martin J. Wainwright. *High-dimensional statistics*, volume 48 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 2019. A non-asymptotic viewpoint. doi:10.1017/9781108627771.
- 73 Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press, 1994. doi:10.1017/CB09780511815478.
- 74 Duncan J. Watts and Steven H. Strogatz. Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684):440–442, 1998. doi:10.1038/30918.
- 75 Mihalis Yannakakis and Fănică Gavril. The maximum  $k$ -colorable subgraph problem for chordal graphs. *Inform. Process. Lett.*, 24(2):133–137, 1987. doi:10.1016/0020-0190(87)90107-4.
- 76 Uri Zwick. Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems. In *Annual ACM Symposium on Theory of Computing (Atlanta, GA, 1999)*, pages 679–687. ACM, New York, 1999. doi:10.1145/301250.301431.