Contents

1 Introduction

Thorughout this text we consider the maximum cut problem, which is shown to be NP-hard by [GJS74] To tackle this problem approximation algorithms and heurisitc algorithms have been proposed.

The text divides into conceptual parts: The first part can be considered the theoretical part, in which we try to give to develop just enough We begin by introducing a semidefinite relaxation for the maximum cut problem.

In section 2, we give an overview of the approximation algorithm introduced by Goemans and Williamson. That includes introducing a semidefinite programming problem relaxation for the maximum cut problemand showing how to derive a cut from a set of points on the unit sphere. We will discuss the performance guarantee in detail. However, the proof of polynomial runtime will not be treated here, we refer to [KV18].

In section 3, we lay the theoretical foundation for the Burer heuristic(described in section 4). In particular we study the special case of the SDP relaxation introduced in section 2 for dimension two in polar coordinates. This leads to an unconstrained nonconvex optimisation problem, which has advantages and disadvantages. On the upside by using this relaxation the number of variables is not increased, i.e. the number of variables remains the number of vertices of the graph. This implies good scalability to large instances. However as the function is nonconvex, we can not expect to solve this relaxation optimally. Therefore we face a trade-off between computational runtime and a theoretical guarantee, see [BMZ02, p. 506]. We will conduct experiments on Mallach instance library to investigate this trade-off. This yields data demonstrating the effectiveness.

In section 4 we give a detailed exposition of the Burer heuristic. The heuristic combines the rank-two relaxation and Goemans-Williamson-type cuts. Given a set of points on the unit circle, we will provide a complete description of how to generate a best possible Goemans-Williamson-type cut in a deterministic manner. Furthermore we will describe an algorithm to improve a cut value by means of trial and error, where the termination criterion is given by the number of consecutive unsuccesful attempts to improve the cut value. Lastly we will end the section with algorithm of the Burer heuristic. This is the algorithm on which the experimental data is based upon.

In section 5 we show that instances of the binary quadratic programming problem can be transformed to instances of the maximum cut problem.

In the second part we will go into computational results. We implemented the heuristic (distilation from ..) and ran it on the instance library provided by Mallach. The results described in section, gather more evidence, that the rank-two relaxation proposed by Burer et al. is highly effective. In the results part we will

2 A semidefinite relaxation for the Max cut problem

Throughout this text we consider undirected graphs G=(V,E) with $V = [1:n] := \{1,\ldots,n\}$ and $E \subseteq \{(i,j) \in V \times V \mid i < j\}$. Note that we adopt this notation to stay coherent with [BMZ02] on which this text is based. By the condition $i < j$ there can only be one edge between the vertices i and j, thus we could have just as well considered $E \subseteq \{\{i,j\} \mid i,j \in V : i \neq j\}$. Let the edge weights $w_{ij} = w_{ji}$ be given such that $w_{ij} = 0$ if $(i,j) \notin E$. We then have $w_{ii} = 0$ for all $i \in V$. We are interested in studying the weights of cuts in the graph G. Given a bipartition of $\left(S, \overline{S}\right)$ of V, a cut is the set $\left\{e \in E \mid |S \cap e| = 1 \text{ and } \left|\overline{S} \cap e\right| = 1\right\}$. Clearly, a cut is not uniquely defined by $\left(S, \overline{S}\right)$, as $\left(\overline{S}, S\right)$ generates the same cut. Furthermore a cut is uniquely defined by a set X, as the bipartion is uniquely given by $(X, V \setminus X)$. The corresponding weight is obtained by summing the weights of the edges in the cut. The task of finding a cut of maximum weight is called the maximum cut problem, abbreviated by MCP, see [KV18]:

Maximum Weight Cut Problem

Instance:	An undirected weighted graph G.
Task:	Find a cut in G with maximum total weight.

It is well known that the MCP is NP-hard, see [GJS74]. Therefore there has been a lot of work on approximation algorithms and heuristics tackling the MCP. A groundbreaking contribution is the approximation algorithm described by Goemans and Williamson in [GW95]. In fact the Burer heuristicis based on this approximation algorithm. Therefore we will spend the rest of this section on developing the theory to understand the algorithm by Goemans Williamson. We will focus on the main ideas and not go into, for example proving the polynomial runtime.
Following the description given by [Vaz03, p. 268 ff], we can motivate the formulation of the maximum cut problemas a binary quadratic program:
We give a for our purposes sufficient definition of binary quadratic program.

Definition 2.1 (Binary quadratic program) Let B be either $\{0,1\}$ or $\{-1,1\}$, and $Q \in \mathbb{R}^{n\times n}$. A (unrestricted) binary quadratic program is of the form:

$$\begin{aligned} &\text{minimize} && x^TQx \\ &\text{subject to} && x_i \in B \quad \forall i \in [1:n] \end{aligned} \tag{1}$$

To every vertex i we assign a indicator variable x_i which is constrained to be in $\{-1,1\}$. This allows us to define the cut-defining partition $\left(S, \overline{S}\right)$ by $S := \{i \mid x_i = 1\}$ and $\overline{S} = \{i \mid x_i = -1\}$. For any edge $\{i,j\}$ in the cut we have $i \in S$ and $j \in \overline{S}$ or vice versa. Thus we have $x_ix_j = -1$ for every edge $\{i,j\}$ in the cut. On the other hand, for every edge $\{i,j\}$ that is not in the cut we have $i,j \in S$ or $i,j \in \overline{S}$, implying $x_ix_j = 1$.

$$\frac{1}{2}\left(1 - x_ix_j\right) = \begin{cases} 1 & \{i,j\} \text{ in the cut defined by } S \\ 0 & \text{otherwise} \end{cases}$$

This explains, why we can write the Maximum Cut Problem, abbreviated with MCP, as the following quadratic program.

$$\begin{aligned} &\text{maximize} && \frac{1}{2}\sum_{1\le i<j\le n} w_{ij}(1-x_ix_j) \\ &\text{subject to} && x_i\in\{-1,1\} \quad \forall i\in[1:n] \end{aligned} \tag{2}$$

Alternatively, we can solve the following binary quadratic program as Lemma 2.2. This will prove useful as we progress in the section.

$$\begin{aligned} &\text{minimize} && \sum_{1\le i<j\le n} w_{ij}x_ix_j \\ &\text{subject to} && |x_i|=1 \quad \forall i\in[1:n] \end{aligned} \tag{3}$$

For the sake of convenience we will introduce some notation to denote sums, see [Aig07, p. 41]. Say we want to sum the even numbers between 0 and 100, we can write $\sum_{k=0}^{100} k\,[k \text{ even}]$. The expression in brackets, which is to be multiplied, means that

$$[k \text{ has property E}] = \begin{cases} 1 & \text{if } k \text{ satisfies property E} \\ 0 & \text{otherwise} \end{cases}$$

This expresses the same sum as

$$\sum_{\substack{k=0 \\ k \text{ even}}}^{100} k \quad \text{or} \quad \sum_{k=1}^{50} 2k$$

Lemma 2.2 The solutions of 2 and 3 coincide.

Proof. We first observe that the constraints of 2 and 3 coincide so it suffices to show that the target function of the BQP is minimised if and only if the target funciton of the MCP is maximised. The following computation uses that $x_ix_j\in\{-1,1\}$ for all $i,j\in[1:n]$:

$$\sum_{1\le i<j\le n} w_{ij}x_ix_j = \sum_{1\le i<j\le n} w_{ij}x_ix_j\,[x_ix_j=1] + \sum_{1\le i<j\le n} w_{ij}x_ix_j\,[x_ix_j=-1]$$
$$= \sum_{1\le i<j\le n} w_{ij}\,[x_ix_j=1] - \sum_{1\le i<j\le n} w_{ij}\,[x_ix_j=-1] = \sum_{1\le i<j\le n} w_{ij} - 2\sum_{1\le i<j\le n} w_{ij}\,[x_ix_j=-1]$$

Using $\sum_{1\le i<j\le n} w_{ij}$ is constant and $\sum_{1\le i<j\le n} w_{ij}(1-x_ix_j) = 2\sum_{1\le i<j\le n} w_{ij}\,[x_ix_j=-1]$. We can thus conclude, that maximising $\frac{1}{2}\sum_{1\le i<j\le n} w_{ij}(1-x_ix_j)$ (and thus $\sum_{1\le i<j\le n} w_{ij}(1-x_ix_j)$) minimises $\sum_{1\le i<j\le n} w_{ij}x_ix_j$ and vice versa. □

Let us introduce some notation allowing for more concise formulation. Consider $W = (w_{ij})_{1\le i,j\le n}, X = (x_{ij})_{1\le i,j\le n}\in\mathbb{R}^{n\times n}$. We denote the sum of their entrywise product by $W\bullet X = \sum_{i=1}^n\sum_{j=1}^n w_{ij}x_{ij}$ and vector of diagonal elements by $\operatorname{diag}(X) = (x_{11},\dots,x_{nn})^T\in\mathbb{R}^n$. Further we denote the vector of all ones by e and denote a symmetric positive semidefinite matrix X by $X\succeq 0$.

The following Lemma shows a useful reformulation of 3, which will form the basis of our construction of the Goemans Williamson algorithm.

Lemma 2.3 The BQP 3 can be rewritten into the following matrix optimisation program

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} W \bullet X \\ \text{subject to} \quad & \operatorname{diag}(X) = e, \\ & \operatorname{rank}(X) = 1, \\ & X \succeq 0 \end{aligned} \tag{4}$$

Proof. First of all, we are going to show that a matrix X satisfies the conditions of 4 if and only if there exists a $x \in \{-1, 1\}^n$ such that $X = xx^T$:
As $\operatorname{rank}(X) = 1$ implies that the columns of the matrix are scalar multiples of each other, there exist $x, y \in \mathbb{R}^n \setminus 0$ such that $X = xy^T$.
Claim: The vectors x and y are necessarily linearly dependent.
Assume otherwise. Then the Cauchy-Schwarz inequality (CS) is strict, i.e. $|\langle x, y \rangle| < \|x\| \, \|y\|$. Furthermore the point $\frac{x+y}{2}$ is linearly independent from x and y. By symmetry, we only show this for y: Given x, y linearly independent. Assume there exists $\lambda \in \mathbb{R}$ such that $\frac{x+y}{2} = \lambda y \Leftrightarrow x = (2\lambda - 1)y$ contradicting x, y linearly independent. Thus $\left|\left\langle x, \frac{x+y}{2} \right\rangle\right| < \|x\| \left\|\frac{x+y}{2}\right\|$ and $\left|\left\langle y, \frac{x+y}{2} \right\rangle\right| < \|y\| \left\|\frac{x+y}{2}\right\|$. Now we can show the claim using the following preparatory calculations. We consider the vector $v := y - \frac{\left\langle y, \frac{x+y}{2} \right\rangle}{\left\|\frac{x+y}{2}\right\|^2} \frac{x+y}{2}$ and calculate:

$$\langle y, v \rangle = \|y\|^2 - \frac{\left\langle y, \frac{x+y}{2} \right\rangle}{\left\|\frac{x+y}{2}\right\|^2} \left\langle y, \frac{x+y}{2} \right\rangle \overset{\text{CS}}{>} \|y\|^2 - \frac{1}{\left\|\frac{x+y}{2}\right\|^2} \|y\|^2 \left\|\frac{x+y}{2}\right\|^2 = 0$$

Furthermore we have:

$$\begin{aligned} \frac{1}{2} \left(\langle x, v \rangle + \langle y, v \rangle \right) = \left\langle v, \frac{x+y}{2} \right\rangle &= \left\langle y, \frac{x+y}{2} \right\rangle - \frac{\left\langle y, \frac{x+y}{2} \right\rangle}{\left\|\frac{x+y}{2}\right\|^2} \left\langle \frac{x+y}{2}, \frac{x+y}{2} \right\rangle \\ &= \left\langle y, \frac{x+y}{2} \right\rangle - \left\langle y, \frac{x+y}{2} \right\rangle = 0 \end{aligned}$$

Thus we have $\langle x, v \rangle + \langle y, v \rangle = 0$ and using the first calculation we get $\langle x, v \rangle < 0$ and $\langle y, v \rangle > 0$. The claim follows by observing that $v^T X v = v^T x y^T v < 0$ contradicting X postive semidefinite.

As x and y are linearly dependent, there exist an $\lambda \in \mathbb{R}$ such that $X = \lambda x x^T$ and $X_{ii} = \lambda x_i^2$ for all $i \in [1:n]$. Using the assumption $\operatorname{diag}(X) = e$ yields $1 = \lambda x_i^2$ for all $i \in [1:n]$. This implies $\lambda \neq 0$ and we can rewrite $x_i^2 = \frac{1}{\lambda}$ for all $i \in [1:n]$. In other words we have $x_i \in \left\{-\frac{1}{\sqrt{\lambda}}, \frac{1}{\sqrt{\lambda}}\right\}$ for all $i \in [1:n]$ and setting $\widetilde{x} := \sqrt{\lambda} x$ yields $|\widetilde{x}_i| = 1$ for all $i \in [1:n]$ and $X = \widetilde{x}\widetilde{x}^T$.

For the other direction: Let $x \in \{-1, 1\}^n$ and set $X = xx^T$. We clearly have $\operatorname{rank}(X) = 1$, also $x_i \in \{-1, 1\}$ for all $i \in [1:n]$ implies $x_i x_i = 1$ for all $i \in [1:n]$, which shows $\operatorname{diag}(X) = e$. That X is positive semidefinite follows from:

$$y^T X y = y^T x x^T y = |(x, y)|^2 \geq 0 \quad \forall y \in \mathbb{R}^n$$

Finally, we conclude the proof by showing that the objective functions coincide. Let $X = xx^T$, with $x \in \{-1, 1\}$:

$$\frac{1}{2} W \bullet X = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} x_i x_j \overset{\substack{w_{ii}=0 \\ w_{ij}=w_{ji}}}{=} \frac{1}{2} \sum_{1 \leq i < j \leq n} w_{ij} x_i x_j$$

$\square$

Remark In the previous proof, we showed that for linearly independent $x, y \in \mathbb{R}^n$ the matrix xy^T is not positive semidefinite, by an explicit construction. We will explain the intuition behind the construction: As x and y are different, by the geometric Hahn-Banach theorem, there exists a hyperplane separating the points. However the points being different is not to sufficient to show that there exists a separating hyperplane containing the origin. This is where the linear independence is necessary. As the construction shows, there exists a hyperplane, generated by v, that contains the origin. Containing the origin is critical as this ensures that $x^T v, y^T v \neq 0$ and that $x^T v$ and $y^T v$ have different signs. Only then we can deduce $v^T xy^T v < 0$.

In order to describe the algorithm by Goemans and Williamson [GW95] we need to define what a semidefinite program is.

Definition 2.4 (Semidefinite program [Vaz03, p.258]) Let $C, D_1, \ldots, D_k \in \mathbb{R}^n$ symmetric and $d_1, \ldots, d_k \in \mathbb{R}$. The general semidefinite programming problem, abbreviated to SDP, is given by

$$\begin{aligned} \text{maximize} \quad & C \bullet X \\ \text{subject to} \quad & D_i \bullet X = d_i \quad \forall i \in [1:k], \\ & X \succeq 0 \end{aligned} \tag{5}$$

The following Theorem guarantees that SDP can be approximately solved in polynomial time. Showing this is a key point in showing that the algorithm presented by Goemans and Williamson has polynomial runtime. As we will not use SDP for the heuristic refer to [Vaz03, p. 258 ff] or [KV18, Theorem 16.10] for a proof.

Theorem 2.5 ([Vaz03, p. 259]) For any $\varepsilon > 0$ semidefinite programs can be solved up within an additive error of ε, in time polynomial in n and $\log(1/\varepsilon)$

By setting $C = -\frac{1}{2}W$ and $D_i = e_i e_i^T, d_i = 1$ for all $i \in [1:n]$ we see that we can relax the matrix optimisation program 4 into a SDP by simply dropping the rank constraint:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} W \bullet X \\ \text{subject to} \quad & \operatorname{diag}(X) = e, \\ & X \succeq 0 \end{aligned} \tag{6}$$

The first big idea is to relax BQP 3 in this ingenious way. We replace the unit scalars x_i by unit vectors $v_i \in \mathbb{R}^d$ (where the dimension $d > 1$ can be chosen freely). Furthermore the product $x_i x_j$ can be interpreted as the one dimensional scalar product, we therefore replace it by $v_i^T v_j$, which is the standard scalar product in $\mathbb{R}^d$. Noting $v_i \in \mathbb{R}^d$ the relaxation writes as follows:

$$\begin{aligned} \text{minimize} \quad & \sum_{1 \leq i < j \leq n} w_{ij} v_i^T v_j \\ \text{subject to} \quad & \|v_i\|_d = 1 \quad \forall i \in [1:n] \end{aligned} \tag{7}$$

Remark 1. We are going to show that 7 is a relaxation of 3:
Let $x \in \{-1, 1\}^n$ be a solution to 3. Let $e_1 \in \mathbb{R}^d$ be the first standard basis vector and set $v_i = x_i e_1 \in \mathbb{R}^d$ for all $i \in [1:n]$. Then the v_i's form a feasible solution as for every $i \in [1:n]$ we have $\|v_i\| = |x_i| \, \|e_1\|_d = 1$. Furthermore values of the objective functions coincide by $v_i^T v_j = x_i x_j e_1^T e_1 = x_i^T x_j$ for all $i, j \in [1:n]$.

2. Observe that for a feasible (and optimal) solution $v_1, \dots, v_n$ each point v_i is located on the unit sphere $\mathbb{S}^{d-1}$ representing the vertex i in G.

Aiming to apply Lemma 2.3, we can change the variables $x_i x_j$ to $v_i^T v_j$. Therefore X_{ij} is given by $v_i^T v_j$ instead of $x_i^T x_j$.

Lemma 2.6 ([Vaz03, p. 259f]) We can rewrite 7 into the following SDP

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} W \bullet X \\ \text{subject to} \quad & \operatorname{diag}(X) = e, \\ & X \succeq 0 \end{aligned} \tag{8}$$

Proof. Throughout this proof we denote the i-th column of a matrix L by L_i.

Let X be a feasible solution of 7. As X is positive semidefinite there exists a $L \in \mathbb{R}^{n \times n}$ such that $X = L^T L$. Thus we have $X_{ij} = L_i^T L_j$ for all $i, j \in [1:n]$. For $i \in [1:n]$ we set $v_i := L_i$. As X is a feasible solution to 8 we have $1 = X_{ii} = L_i^T L_i = \|v_i\|_n^2$, which shows $\|v_i\|_n = 1$ for all $i \in [1:n]$. Thus the v_i's constitute a feasible solution to 7 of the same objective function value:

$$\frac{1}{2} \sum_{i,j=1}^{n} w_{ij} v_i^T v_j = \frac{1}{2} W \bullet L^T L = \frac{1}{2} W \bullet X.$$

Given a feasible solution $v_1, \dots, v_n$ to 7 we define the matrix $L \in \mathbb{R}^{n \times n}$ by $L_i := v_i$. Now set $X = L^T L$, we then have $X_{ij} = v_i^T v_j$. As the v_i's are a feasible solution we have $X_{ii} = \|v_i\|_n^2 = 1$ for $i \in [1:n]$. Furthermore X is positive semidefinite since $y^T X y = \|Ly\|_n^2 \geq 0$ for all $y \in \mathbb{R}^n$, which shows that X is a feasible solution of 8 . By construction we have that the two feasible solutions have the same value:

$$\frac{1}{2} W \bullet X = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} X_{ij} = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} v_i^T v_j$$

□

Remark As this is essential, we reiterate how to retrieve a solution of 7 from a solution X of 8: As X is a positive semidefinite matrix, using the Cholesky decomposition we compute a matrix $L \in \mathbb{R}^{n \times n}$ sucht that $X = L^T L$. The columns of L are a solution to 7.

The algorithm presented by Goemans and Williamson can be stated as follows [KV18, p.424]: We emphasise that nonnegative edge weights are required. This requirement is essential in the proof of Lemma 2.10.

As we will not solve semidefinite programs in the heuristic presented later in the text, we will for simplicity's sake, assume that we can find an optimal solution to 8. That means that we have an optimal solution to 7. This inaccuracy can be absorbed into the approximation factor; as stated in [Vaz03, p. 260]. A detailed account of how to handle nearly optimal solutions is given in [KV18]. Until the end of the section we will closely follow the description provided in [Vaz03, p. 260 ff]. In order to approximate the MCP we need to relate this optimum solution to 7 to a cut. As the matrix whose columns are an optimum solution to 7 is in general not of rank 1 we are faced with a rounding problem. The question boils down to the following: How can we derive a good cut from an optimum solution to 7?

Algorithm 1 Goemans Williamson algorithm

Input: A graph with nonnegative edge weights

Output: A cut given by the set S

1: Find an approximately optimal solution X to 3

2: Find vectors $y_1, ..., y_n \in S^d$ such that $y_i^T y_j = x_{ij}$ for all $i, j \in [1:n]$ (using Cholesky decomposition)

3: Choose a random point $r \in \mathbb{S}^d$

4: Set $S := \left\{ i \in [1:n] \mid r^T y_i \geq 0 \right\}$

The second big idea by Goemans and Williamson, answers the question. As the heuristic presented later on in the text is based on this idea we will describe it in detail.

Let $v_1, \ldots, v_n$ be an optimal solution and denote the optimum value by OPT. We have $v_i \in \mathbb{S}^d$ for all $i \in [1:n]$ and for all $i, j \in [1:n]$ we have $\cos(\theta_{ij}) = v_i^T v_j$. Here θ_{ij} denotes the angle between v_i and v_j. The contribution of v_i and v_j to the value of OPT is, as we simply read off from the definition of the objective function:

$$\frac{1}{2} w_{ij} \left(1 - \cos(\theta_{ij})\right)$$

Therefore the contribution of v_i and v_j is the greater, the closer θ_{ij} is to π (, while considering the same w_{ij}). Geometrically this means, that diametrially opposed points have the highest contribution. This means that roughly speaking we want to seperate diametrially opposed points. The cuts introduced in the next Definition achieve just that.

Definition 2.7 (Goemans-Williamson-type cut) A Goemans-Williamson-type cut for a direction $r \in \mathbb{S}^n$ is the cut given by $\left(S, \overline{S}\right)$, where we set $S := \left\{ i \in [1:n] \mid v_i^T r \geq 0 \right\}$ and $\overline{S} := \left\{ i \in [1:n] \mid v_i^T r < 0 \right\}$.

As the algorithm is based on rounding the values according to a uniformly chosen Goemans-Williamson-type cut we are naturally interested in knowing what the odds of separating two points are.

Lemma 2.8 Let $n \geq 2$ and $v_i, v_j \in \mathbb{S}^n$ different. We divide the unit sphere into two hemispheres by a hyperplane through the origin. The probability that v_i and v_j are seperated is given by $P\left[v_i \text{ and } v_j \text{ are separated}\right] = \frac{\theta_{ij}}{\pi}$

Proof. Consider the plane that contains $0, v_i$ and v_j, given by $\operatorname{span}(v_i, v_j)$. A hyperplane (through the origin) seperates the two points if and only if its projection onto the plane is on intersect the arc connecting v_i, v_j on the sphere. As the hyperplane is chosen uniformly and the angle between v_i and v_j is given by θ_{ij} the probability, that the hyperplane separates the two points is $\frac{\theta_{ij}}{\pi}$. □

Lemma 2.9 The expression $\alpha := \min_{0<\theta\leq\pi} \frac{2}{\pi} \frac{\theta}{1-\cos(\theta)}$ is well defined and $\alpha > 0.87856$ holds.

Proof. The enumerator has a simple zero and the denominator a zero of order two. As the terms are nonnegative we have $\lim\limits_{\theta\to 0} = \frac{2}{\pi} \frac{\theta}{1-\cos(\theta)} = \infty$. Thus we have a contnuous function on an interval and the As the denominator and the numerator have simple zeroes in zero, the function can be continously extended in 0. Thus we have a contnuous function on an interval and the minimum will be attained. We take the derivative and get

$$\frac{\partial}{\partial \theta}\frac{2}{\pi}\frac{\theta}{1-\cos(\theta)} = \frac{2}{\pi}\frac{1-\cos(\theta)-\theta\sin(\theta)}{(1-\cos(\theta))^2}$$

Thus we have a critical point if and only if $1 = \cos(\theta) + \theta\sin(\theta)$ In $(0, \pi]$ there is only one solutions, namely 2.33112237041442. By the first part this has to be a local minimum with value 0.878567205784851604. By $\frac{2}{\pi}\frac{\pi}{1-\cos(\pi)} = 2$ the local minimum is a global minimum and $\alpha > 0.87856$ □

Lemma 2.10 The expected value of the weight of a Goemans-Williamson-type cut is greater or equal to $\alpha \cdot \mathrm{OPT}$, where OPT denotes the value of an optimal solution to 7.

Proof. Using Lemma 2.9 we have for all $\theta \in (0, \pi]$:

$$\frac{\theta}{\pi} \geq \alpha\frac{1-\cos(\theta)}{2}$$

We denote the weight corresponding to the edge (i, j) by w_{ij} and set $w_{ij} = 0$ otherwise. The expected value of a Goemans-Williamson-type cut is given by
$\sum_{1\leq i<j\leq n} w_{ij}P\left[v_i \text{ and } v_j \text{ are separated}\right]$, together with Lemma 2.8 we can compute:

$$\begin{aligned}
\sum_{1\leq i<j\leq n} w_{ij}P\left[v_i \text{ and } v_j \text{ are separated}\right] &\geq \sum_{1\leq i<j\leq n} w_{ij}\frac{\theta_{ij}}{\pi} \\
&\geq \frac{\alpha}{2}\sum_{1\leq i<j\leq n} w_{ij}\left(1-\cos(\theta_{ij})\right) \\
&= \alpha \cdot \mathrm{OPT}
\end{aligned}$$

□

Remark In particular theorem 2.10 implies that there exists a Goemans-Williamson-type cut with value $0.8785 \cdot \mathrm{OPT}$.

Therefore we have discussed all the, for our purposes, important aspects and can state, as in [KV18, Theorem 16.12]:

Theorem 2.11 The Goemans-Williamson Max-Cut-Algorithm returns a set S for which the expected value of the associated cut is at least 0.878 times the maximum possible value.

3 A rank-two relaxation

Write outline of first paragraph

Type out motivation from Goemans Williamson

In this section we will investigate relaxation 7 in dimension two. Instead of using cartesian coordinates we will use polar coordinates. We will see that this change carries big changes. We will introduce the notion of angular representation of cuts, which are certain vectors that correspond to cuts. The section will culminate in giving a classification of cuts as extremal points of the objective function of the relaxation.

Using polar coordinates we can represent any point on the unit sphere by their angle. In other words for every $v \in S^1$, there exists a $\theta \in \mathbb{R}$ such that

$$v = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix}$$

By this idea we can represent n points $v_1, \ldots, v_n$ on the unit sphere, by a vector $\theta = (\theta_1, \ldots, \theta_n) \in \mathbb{R}^n$, such that the i-th vector corresponds to the i-th coordinate:

$$v_i = \begin{pmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{pmatrix} \qquad \text{for all } i \in [1:n] \tag{9}$$

Polar coordinates have the additional benefit, that using the addition theorem the inner product simplifies to:

$$v_i^T v_j = \cos(\theta_i)\cos(\theta_j) + \sin(\theta_i)\sin(\theta_j) = \cos(\theta_i - \theta_j) \tag{10}$$

Next we define the following map:

$$T : \mathbb{R}^n \to \mathbb{R}^{n\times n},\ \theta \mapsto (\theta_i - \theta_j)_{\substack{i\in[1:n] \\ j\in[1:n]}}$$

For every $\theta \in \mathbb{R}^n$ the resulting matrix $T(\theta)$ is skew-symmetric, since for all $i, j \in [1:n]$

$$T(\theta)_{ij} = \theta_i - \theta_j = -(\theta_j - \theta_i) = -T(\theta)_{ji}$$

Throughout the text application of a scalar function onto a matrix, is to be understood as entrywise application of the scalar function. We define the following function, which will turn out to be of central importance:

$$f : \mathbb{R}^n \to \mathbb{R},\ \theta \mapsto \frac{1}{2} W \bullet \cos(T(\theta)) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\cos(\theta_i - \theta_j)$$

Read if this makes sense

Lemma 3.1 The function f is

1. invariant with respect to simultaneous, uniform rotation on every component. That means for every $\theta \in \mathbb{R}^n$ and every $\tau \in \mathbb{R}$ we have $f(\theta) = f(\theta + \tau e)$.
2. 2π-periodic with respect to each variable, i.e., for all $\theta \in \mathbb{R}^n$ we have $f(\theta) = f(\theta + 2\pi e_i)$, where e_i denotes the i-th standard basis vector.

Proof. We beginn by proving the first property. Let $\theta \in \mathbb{R}^n$ and $\tau \in \mathbb{R}$, it is sufficient to show $T(\theta + \tau e) = T(\theta)$, as this is the only term in f that depends on θ. This holds as for any $i, j \in [1:n]$ we have $T(\theta + \tau e)_{ij} = \theta_i + \tau - (\theta_j + \tau) = \theta_i - \theta_j = T(\theta)_{ij}$. To show the second property let $\theta \in \mathbb{R}^n$ and $i \in [1:n]$ and compute using 2π-periodicity of the cosine:

$$\cos(T(\theta + 2\pi e_i))_{ij} = \cos(\theta_i + 2\pi - \theta_j) = \cos(\theta_i - \theta_j) = \cos(T(\theta))_{ij} \text{ and}$$
$$\cos(T(\theta + 2\pi e_i))_{ji} = \cos(\theta_j - \theta_i - 2\pi) = \cos(\theta_j - \theta_i) = \cos(T(\theta))_{ji}$$

As all other entries of $\cos(T(\theta))$ remain unchanged, $f(\theta) = f(\theta + 2\pi e_i)$ is established. □

Recalling 10 and 9 we see that minimising f is nothing but solving 7 in polar coordinates. This observation is of central importance and we denote the relaxation of the MCP:

$$\min_{\theta \in \mathbb{R}^n} f(\theta) \tag{11}$$

This point of view has favorable and unfavorable features. As f is nonconvex we have no general tool to find a global minimum or even to decide if a local minimum is a global minimum. In general, there can be multiple local but nonglobal minima. However, formulation 11 is an unconstrained minimisation problem, with a fairly easy analytical description of f. Making use of its simplicity we compute its partial derivatives by hand.

This is an unconstrained optimization problem with a nonconvex objective function.

Lemma 3.2 The partial derivatives of f for any $j \in [1:n]$ is given by

$$\frac{\partial f}{\partial \theta_j}(\theta) = \sum_{k=1}^{n} w_{kj} \sin\left(\theta_k - \theta_j\right)$$

and the gradient can be written as:

$$\nabla f(\theta) = e^T \left(W \circ \sin(T(\theta)\right) \tag{12}$$

Here the notation $\circ$ stands for the Hadamard product, i.e. the entrywise product of W and $\sin(T(\theta)$

Proof. We fix $j \in [1:n]$ as the index of the variable for which we compute the partial derivative. To avoid ambiguity we rename the indices of f yielding:

$$f(\theta) = \frac{1}{2} \sum_{k=1}^{n} \sum_{i=1}^{n} w_{ki} \cos(\theta_k - \theta_i)$$

From this expression we see that the term $k = i = j$ is equal to zero and we only need to distinguish between following two cases:
In case of $k = j$ and $i \neq j$ we have:

$$\frac{\partial}{\partial j} w_{ji} \cos(\theta_j - \theta_i) = -w_{ji} \sin(\theta_j - \theta_i) \overset{\substack{w_{ij}=w_{ji} \\ \sin \text{ odd}}}{=} w_{ij} \sin(\theta_i - \theta_j)$$

In case of $i = j$ and $k \neq j$ we have:

$$\frac{\partial}{\partial j} w_{kj} \cos(\theta_k - \theta_j) = w_{kj} \sin(\theta_k - \theta_j)$$

In the case of $k \neq j \neq i$ the derivative is zero as the term is constant with respect to j. By linearity we therefore conclude:

$$\frac{\partial}{\partial j} f(\theta) = \frac{1}{2} \left(\sum_{k=1}^{n} w_{kj} \sin(\theta_k - \theta_j) + \sum_{i=1}^{n} w_{ij} \sin(\theta_k - \theta_j) \right) = \sum_{k=1}^{n} w_{kj} \sin(\theta_k - \theta_j)$$

To justify the gradient, we only need to observe that $\frac{\partial}{\partial j} f(\theta)$ is the sum of the entries of the j-th column of the matrix $W \circ \sin(T(\theta))$. This coincides with the j-th entry of $\left(W \circ \sin(T(\theta))\right)^T e = \left(e^T \left(W \circ \sin(T(\theta)\right)\right)^T$. $\square$

To be able to classify extremal points we now compute the Hessian matrix, i.e. the second derivative of f.

Lemma 3.3 The Hessian matrix H of f is given by

$$H(\theta) = W \circ \cos(T(\theta)) - \operatorname{diag}\left((W \circ \cos(T(\theta)))\, e\right) \tag{13}$$

and the partial derivatives of second order are explicitly give by:

$$\frac{\partial^2}{\partial\theta_i \partial\theta_j} f(\theta) = \begin{cases} w_{ij}\cos(\theta_i - \theta_j) & \text{if } i \neq j \\ -\sum_{k \neq j} w_{kj}\cos(\theta_k - \theta_j) & \text{if } i = j \end{cases}$$

Proof. Case $i \neq j$:
By $i \neq j$ we have that $w_{kj}\sin(\theta_k - \theta_j)$ is not constant with respect to θ_i if and only if $k = i$. Noting that $\frac{\partial}{\partial\theta_i} w_{ij}\sin(\theta_i - \theta_j) = w_{ij}\cos(\theta_i - \theta_j)$ we get

$$\frac{\partial^2}{\partial\theta_i \partial\theta_j} f(\theta) = w_{ij}\cos(\theta_i - \theta_j)$$

Case $i = j$:
In this case we have $w_{jj} = 0$ and $\frac{\partial}{\partial\theta_j}\sin(\theta_k - \theta_j) = -\cos(\theta_k - \theta_j)$. By linearity we thus get

$$\frac{\partial^2}{\partial\theta_j \partial\theta_j} f(\theta) = \sum_{k \neq j} -w_{kj}\cos(\theta_k - \theta_j)$$

□

Next we want to investigate the relationship between the function f and the cuts of a graph.

Definition 3.4 (Angular representation of a cut) A vector $\theta \in \mathbb{R}^n$ is called an angular representation of a cut, or simply a cut, if there exist integers k_{ij} such that $\theta_i - \theta_j = k_{ij}\pi$ for all $i, j \in [1:n]$

Let $\overline{\theta}$ be an angular representation of a cut. Using that $\cos(\overline{\theta}_i - \overline{\theta}_j)$ evaluates to 1 if k_{ij} is even and to -1 if k_{ij} is odd, there exists a binary vector $\overline{x} \in \{-1, 1\}^n$ such that

$$\cos(\overline{\theta}_i - \overline{\theta}_j) = \overline{x}_i \overline{x}_j = \pm 1 \quad \text{for all } i, j \in [1:n]\,. \tag{14}$$

This can be seen by setting $x_1 = 1$ and

$$\overline{x}_i = \begin{cases} 1 & \text{if } \overline{\theta}_i - \overline{\theta}_1 \in 2\pi\mathbb{Z} \\ -1 & \text{otherwise} \end{cases} \quad \text{for } i \in [2:n]$$

This construction is correct, as for any $i, j \in [1:n]$ we have k_{ij} is even if and only if k_{i1} and k_{j1} are both even or both odd if and only if $\overline{x}_i \overline{x}_j = 1$. The equivalence holds by

$$k_{ij}\pi = \overline{\theta}_i - \overline{\theta}_j = \overline{\theta}_i - \overline{\theta}_1 - (\overline{\theta}_j - \overline{\theta}_1) = (k_{i1} - k_{j1})\pi$$

Then $\overline{x}$ can be viewed as the cut corresponding to $\overline{\theta}$. Moreover, the cut value corresponding to $\overline{\theta}$ is

$$\psi(\overline{\theta}) = \frac{1}{2} \sum_{1 \le i < j \le n} w_{ij}\left(1 - \cos(\overline{\theta}_i - \overline{\theta}_j)\right) \tag{15}$$

In the next Proposition we give a detailed acount of this correspondence.

Proposition 3.5 Modulo the uniform rotation and the periodicity for each variable, there is an one-to-one correspondence between binary cuts and angular representations of a cut, given by

$$\overline{\theta}_i = \begin{cases} 0 & \text{if } \overline{x}_i = 1 \\ \pi & \text{if } \overline{x}_i = -1 \end{cases} \tag{16}$$

In the same manner we can set

$$\overline{x}_i = \begin{cases} 1 & \text{if } \overline{\theta}_i = 0 \\ -1 & \text{if } \overline{\theta}_i = \pi \end{cases}$$

Proof. It is clear that assignment 16 assigns exactly one angular representation to a binary cut. On the other hand every angular representation of a cut has a representant, where every variable is either 0 or π. Let $\overline{\theta}$ be an angular representation of a cut. By the second property, we can assume $\overline{\theta}_i \in [0, 2\pi)$ for all $i \in [1:n]$. We only need to show that there exists a $\tau \in \mathbb{R}$ such that $\overline{\theta}_i - \tau \in \{0, \pi\}$ for all $i \in [1:n]$. If $\overline{\theta}_i \in \{0, \pi\}$ for all $i \in [1:n]$ we simply pick $\tau = 0$ and are done. Otherwise there exists an $i \in [1:n]$, such that $\overline{\theta}_i \notin \{0, \pi\}$. If $\overline{\theta}_i \in (0, \pi)$ we choose $\tau := \overline{\theta}_i$, else we have $\overline{\theta}_i \in (\pi, 2\pi)$ and we choose $\tau := \overline{\theta}_i - \pi$. Now let $j \in [1:n]$. With this choice we just have to show that

$$\overline{\theta}_j = \begin{cases} \tau & \text{if } \overline{\theta}_j \in (0, \pi) \\ \pi + \tau & \text{if } \overline{\theta}_j \in (\pi, 2\pi) \end{cases} \tag{17}$$

holds. As $\overline{\theta}$ is an angular representation of a cut there exists a $k \in \mathbb{Z}$ such that $\overline{\theta}_i - \overline{\theta}_j = k\pi$. By rearranging the terms there exists a $\widetilde{k} \in \mathbb{Z}$ such that $\overline{\theta}_j = \overline{\theta}_i - k\pi = \widetilde{k}\pi + \tau$. This shows equation 17. This shows $\left(\overline{\theta} - \tau e\right)_i \in \{0, \pi\}$ for all $i \in [1:n]$. Now let $\overline{\theta}$ be any angular representaion of a cut and $\overline{\vartheta} \in \{0, \pi\}^n$ its representant. In the proof of Proposition 3.1 we have seen that $\cos(\overline{\theta}_i - \overline{\theta}_j) = \cos(\overline{\omega}_i - \overline{\omega}_j)$ for all □

Using this correspondence we can represent a cut by both $\overline{\theta}$ and $\overline{x}$. Given a binary representation of a cut $\overline{x}$ (or an angular representation $\overline{\theta}$), we denote the angular representation by $\theta(\overline{x})$ (or the binary representation by $x(\overline{\theta})$ of that same cut. The next Proposition states an import property of angular representations of a cut.

Proposition 3.6 Every angular representation of a cut $\overline{\theta} \in \mathbb{R}^n$ is a stationary point of the function f.

Proof. As $\overline{\theta}$ is a cut we have $\overline{\theta}_i - \overline{\theta}_j \in \mathbb{Z}\pi$ for all $i, j \in [1:n]$. This directly implies that every entry of $\sin(T(\overline{\theta}))$ is zero. Plugging this into 12 yields $\nabla f(\overline{\theta}) = 0$. This shows that $\overline{\theta}$ is a stationary point. □

Definition 3.7 (Nonnegatively summable matrix) A matrix $M \in \mathbb{R}^{n \times n}$ is called nonnegatively summable if the sum of the entries in every principal submatrix of M is nonnegative, or equivalently, if $u^T M u \geq 0$ for every binary vector $u \in \{0, 1\}^n$

Put in words a principal submatrix is a matrix that we get by crossing out rows and corresponding columns, i.e. crossing out the i-th row implies crossing out the i-th coloum as well and vice versa. By definition, every semidefinite matrix is nonnegatively summable. However the other implication does not hold as the following example shows.

Example Let $n > 1$ and denote the identity matrix as $I \in \mathbb{R}^{n\times n}$. The matrix $ee^T - I$ is nonnegatively summable but not positive semidefinite. As all the entries of $ee^T - I$ are nonnegative the matrix is clearly nonnegatively summable. To show that $ee^T - I$ is not positive semidefinite we consider the vector $u := e_1 - e_n$. By $n > 1$ we have $1 \neq n$ and we get $u^T \left(ee^T - I\right) u = u^T (-1, 0, \cdots, 0, 1) = -2 < 0$.

Before we can provide a characterisation for maximum (and minimum) cuts in Lemma 3.9 we prove a little calculation rule:

Lemma 3.8 Let $W \in \mathbb{R}^{n\times n}$ be a symmetric matrix, $x \in \mathbb{R}^n$ and $\delta, \delta' \in \{0,1\}^n$. Then we have

$$(\delta \circ x)^T W (\delta' \circ x) = \delta^T \left(W \circ xx^T\right) \delta' = \delta'^T \left(W \circ xx^T\right) \delta$$

Proof. First we observe that for any $\delta \in \{0,1\}^n$ we have

$$(\delta \circ x)_i = \begin{cases} 0 & \text{if } \delta_i = 0 \\ x_i & \text{if } \delta_i = 1 \end{cases} = \delta_i x_i$$

Using this we can compute:

$$\begin{aligned} (\delta \circ x)^T W (\delta' \circ x) &= \sum_{i=1}^n (\delta \circ x)_i \sum_{j=1}^n w_{ij} (\delta' \circ x)_j = \sum_{i=1}^n \delta_i x_i \sum_{j=1}^n w_{ij} \delta'_j x_j \\ &= \sum_{i=1}^n \delta_i \sum_{j=1}^n w_{ij} x_i x_j \delta'_j = \delta^T A \delta' \end{aligned}$$

where $A_{ij} = w_{ij} x_i x_j$ for all $i, j \in [1:n]$. Thus we have $A = W \circ xx^T$ and the first equality holds. Having established the first equality the second follows by noting that W is symmetric

$$(\delta \circ x)^T W (\delta' \circ x) = (\delta' \circ x)^T W^T (\delta \circ x) = \delta'^T \left(W \circ xx^T\right) \delta$$

□

Lemma 3.9 Let $\overline{x} \in \{-1,1\}^n$ be given and consider the matrix $M(\overline{x}) \in \mathbb{R}^{n\times n}$ defined as

$$M(\overline{x}) = W \circ \left(\overline{x}\overline{x}^T\right) - \operatorname{diag}\left(\left(W \circ \left(\overline{x}\overline{x}^T\right)\right) e\right) \tag{18}$$

Then, $\overline{x}$ is a maximum (respectively, minimum) cut if and only if $M(\overline{x})$ (respectively, $-M(\overline{x})$) is nonnegatively summable.

Proof. Consider the quadratic function $q : \mathbb{R}^n \to \mathbb{R},\ x \mapsto x^T W x/2$. By Lemma 2.2 we know that $\overline{x} \in \{-1,1\}^n$ is a maximum cut if and only if $\overline{x}$ satisfies $q(\overline{x}) \leq q(x)$ for all $x \in \{-1,1\}^n$. Consider a maximum cut $\overline{x} \in \{-1,1\}^n$ and an arbitrary $x \in \{-1,1\}^n$. As W is symmetric $\overline{x}^T W x = x^T W \overline{x}$ holds. Using this and the previous equivalence we have:

$$\begin{aligned} 0 &\leq q(x) - q(\overline{x}) = \frac{1}{2}\left(x^T W x - \overline{x}^T W \overline{x}\right) = \frac{1}{2}\left(x^T W x + \overline{x}^T W x - x^T W \overline{x} - \overline{x}^T W \overline{x}\right) \\ &= \frac{1}{2}(x + \overline{x})^T W (x - \overline{x}) = \left(\overline{x} + \frac{1}{2}(x - \overline{x}\right)^T W (x - \overline{x}) \qquad (19) \\ &= \overline{x}^T W (x - \overline{x}) + \frac{1}{2}(x - \overline{x})^T W (x - \overline{x}) \end{aligned}$$

As we have $\overline{x}, x \in \{-1,1\}^n$ we know that either $x_i = \overline{x}_i$ or $x_i = -\overline{x}_i$ for all $i \in [1:n]$. With this observation we see that

$$(x - \overline{x})_i = \begin{cases} 0 & \text{if } x_i = \overline{x}_i \\ -2\overline{x}_i & \text{if } x_i \neq \overline{x}_i \end{cases} \text{ for all } i \in [1:n]$$

and by defining $\delta \in \mathbb{R}^n$ as

$$\delta_i = \begin{cases} 0 & \text{if } x_i = \overline{x_i} \\ 1 & \text{if } x_i \neq \overline{x_i} \end{cases}$$

we get the identity

$$x - \overline{x} = -2\delta \circ \overline{x}$$

With this we continue computation 19:

$$\begin{aligned} 0 &\leq \overline{x}^T W (x - \overline{x}) + \frac{1}{2} (x - \overline{x})^T W (x - \overline{x}) \\ &= -2\overline{x}^T W (\delta \circ \overline{x}) + 2 (\delta \circ \overline{x})^T W (\delta \circ \overline{x}) \\ &= -2 (e \circ \overline{x})^T W (\delta \circ \overline{x}) + 2 (\delta \circ \overline{x})^T W (\delta \circ \overline{x}) \\ &\overset{3.8}{=} -2\delta^T \left(W \circ \overline{x}\overline{x}^T\right) e + 2\delta^T \left(W \circ \overline{x}\overline{x}^T\right) \delta \\ &= -2\delta^T \text{diag}\left((W \circ \overline{x}\overline{x}^T)e\right) \delta + 2\delta^T \left(W \circ \overline{x}\overline{x}^T\right) \delta \overset{\text{def } M(\overline{x})}{=} 2\delta M(\overline{x})\delta \end{aligned}$$

In the second to last equality we used $\delta^T v = \delta^T \text{diag}(v)\, \delta$, which follows immediately from

$$(\text{diag}(v)\delta)_i = \begin{cases} 0 & \text{if } \delta_i = 0 \\ v_i & \text{if } \delta_i = 1 \end{cases} \quad \text{for all } i \in [1:n]\,.$$

So far we have shown $0 \leq \delta^T M(\overline{x})\delta$, however for nonnegative summability we need to argue $0 \leq y^T M(\overline{x}) y$ for all $y \in \{0,1\}^n$.
Claim: For any $y \in \{0,1\}^n$ there exists a $x \in \{-1,1\}^n$ such that $\delta = y$.
First recal that $\delta \in \{0,1\}^n$ depends on $\overline{x}$, which is fixed, and x, which is a variable. To clarify this dependence we denote δ as $\delta_{\overline{x}}(x)$. As $x \in \{-1,1\}^n$ can be chosen arbitrarily, we choose

$$x_i = \begin{cases} \overline{x_i} & \text{if } y_i = 0 \\ -\overline{x_i} & \text{if } y_i = 1 \end{cases}$$

This choice directly implies $\delta_{\overline{x}}(x) = y$. The claim shows that $M(\overline{x})$ is nonnegatively summable. Now we want to show that $\overline{x}$ is minimum cut if and only if $-M(\overline{x})$ is nonnegatively summable. We do this in the same way as above only outlining the slight changes. Let q be the quadratic function we defined above, $\overline{x} \in \{-1,1\}^n$ a minimum cut and $x \in \{-1,1\}^n$ arbitrary. Thus we have $0 \geq q(x) - q(\overline{x})$ and the same compuations as above yield $0 \geq \delta^T M(\overline{x})\delta \Leftrightarrow 0 \leq \delta^T \left(-M(\overline{x})\right)\delta$. The nonnegative summability of $-M(\overline{x})$ follows in the same manner as above. $\square$

Remark 1. Let $\overline{x} \in \{-1,1\}^n$. Then the map

$$\delta_{\overline{x}} : \{-1,1\}^n \to \{0,1\}^n\,,\; x \mapsto \delta_{\overline{x}}(x),\text{ where } \delta_{\overline{x}}(x)_i = \begin{cases} 0 & \text{if } x_i = \overline{x_i} \\ 1 & \text{if } x_i \neq \overline{x_i} \end{cases} \text{ for all } i \in [1:n]$$

is a bijection.
In the previous proof we have shown that $\delta_{\overline{x}}$ is surjective and as $\{-1,1\}^n, \{0,1\}^n$ are finite sets of the same cardinality we have bijectivity.

2. The matrix $M(\overline{x})$ is symmetric for every $\overline{x} \in \{0,1\}^n$:
The matrix $\operatorname{diag}\left((W \circ \left(\overline{x}\overline{x}^T\right) e\right)$ is symmetric as it is a diagonal matrix. As W is symmetric, we have for every $i,j \in [1:n]$:

$$\left(W \circ \overline{x}\overline{x}^T\right)_{ij} = w_{ij}\overline{x}_i\overline{x}_j = w_{ji}\overline{x}_j\overline{x}_i = \left(W \circ \overline{x}\overline{x}^T\right)_{ji}$$

This shows that $\left(W \circ \overline{x}\overline{x}^T\right)$ is symmetric and, as a difference of symmetric matrices, $M(\overline{x})$ is a symmetric matrix.

Comparing the Hessian of f as given in 13 to 18, we see that the two look similar. The difference is that $\cos(T(\theta))$ in 13 is replaced by xx^T in 18. Comparing the entries we see that $\cos(\theta_i - \theta_j) = x_i x_j$ for all $i,j \in [1:n]$ is a sufficent condition for $H(\theta) = M(x)$. This observation leads to the next theorem providing a classification of angular representations of cuts as stationary points of the function f.

Theorem 3.10 Let $\overline{\theta}$ be a angular representation of a cut, or simply a cut, and let $\overline{x} \equiv x(\overline{\theta})$ be the associated binary cut. If $\overline{\theta}$ is a local minimum (respectively, local maximum) of $f(\theta)$, then $\overline{x}$ is a maximum (respectively, minimum) cut). Consequently, if $\overline{x}$ is neither a maximum cut nor a minimum cut, then $\overline{\theta}$ must be a saddle point of f.

Proof. First recall that every angular representation of a cut is a critical point of f and that $\cos(\overline{\theta_i} - \overline{\theta_j}) = \overline{x}_i\overline{x}_j$ for all $i,j \in [1:n]$. Therefore $H(\overline{\theta}) = M(\overline{x})$ holds. Let $\overline{\theta}$ be a local minimum of f. As f is smooth $H(\overline{\theta})$ is positive semidefinite and in particular nonnegatively summable. By $H(\overline{x}) = M(\overline{x})$ we have that $M(\overline{x})$ is nonnegatively summable. Now let $\overline{\theta}$ be a local maximum of f. Then $H(\overline{\theta})$ is negative semidefinite, that means that $-H(\overline{\theta})$ is nonnegatively summbale. Therefore $-H(\overline{\theta}) = -M(\overline{x})$ implies that $-M(\overline{x})$ is nonnegatively summable. Lastly let $\overline{x}$ be neither a maximum cut nor a minimum cut. By Lemma 3.9 we know that neither $M(\overline{x})$ nor $-M(\overline{x})$ is nonnegatively summable. By $M(\overline{x}) = H(\overline{\theta})$ we have that $H(\overline{\theta})$ is neither positive semidefinite nor negative semidefinite. By the necessary condition for a local extremum we conclude that $\overline{\theta}$ is neither a local minimum nor a local maximum. As $\overline{\theta}$ is a critical point of f, it must be a sattle point. □

Although all cuts are stationary points of f (by 3.6) Theorem 3.10 shows that only the maximum points can be local minima of f, see [BMZ02, p. 508].

Example The converse of the two implications in the above theorem do not hold. We will give an example showing that the angular representation of a maximum cut is not necessarily a local minimum of f.

Consider the complete graph with three vertices, where every edge has weight 1. Then the weight of a maximum cut is 2 and is achieved by $\overline{x} = \begin{pmatrix} 1 & -1 & -1 \end{pmatrix}$. We then have

$$W \circ \overline{x}\overline{x}^T = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{pmatrix} \text{ and } \left(W \circ \overline{x}\overline{x}^T\right) e = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}$$

Using 18 we get

$$M(\overline{x}) = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{pmatrix} - \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{pmatrix}$$

Straight forward computation shows nonnegative summability, namely $u^T M(\overline{x}) u \geq 0$ for all $u \in \{0,1\}^n$. However, $M(\overline{x})$ is not positive semidefinite, since

$$\begin{pmatrix} 1 & 0 & 2 \end{pmatrix} M(\overline{x}) \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} = -2 < 0$$

Using $H(\theta(\overline{x})) = M(\overline{x})$, we know that $H(\theta(\overline{x}))$ is not postive semidefinite. Thus $\theta(\overline{x})$ is not a local minimum of f.

However there special classes of instances, where for every maximum cut $\overline{x}$, the angular representation $\theta(\overline{x})$ is a local minimum. The following Proposition gives a class for which this holds:

Proposition 3.11 For a bipartite graph with nonnegative edges, the global minimum value of f is attained by a maximum cut.

Proof. We denote the bipartite graph $G = (A, B, E)$. As all the edge weights are nonnegative, the cut (A, B), that cuts through all edges is a maximum cut. For that cut we have $\cos(\theta_i - \theta_j) = -1$ for all $\{i, j\} \in E$. For that cut f evaluates to $-\frac{1}{2} e^T W e$. This value must be a global minimum of f as all the entries of W are nonnegative. $\square$

For instances, where a maximum cut $\overline{x}$ corresponds to a local minimum $\theta(\overline{x})$ of f, the optimality of x can be checked in polynomial time. That is because $\theta(\overline{x})$ local minimum implies $H(\theta(\overline{x}) = M(\overline{x})$ postive semidefinite. This in turn implies $M(\overline{x})$ nonnegatively summable, which shows $\overline{x}$ maximum cut by 3.9. A detailed proof showing that we can decide whether a symmetric matrix (- $M(\overline{x})$ is symmetric-) is positive semidefinite in polynomial time is provided in [KV18] Theorem 16.8.

Theorem 3.10 directly implies the following Corollary, which plays an importrole for the heuristic we will develop in the next section.

Corollary 3.12 Let $x \in \{0,1\}^n$ be a nonmaximum cut. Then $\theta(x)$ can not be a local minimum.

Proof. The statement is the contraposition of the first statement in theorem 3.10 $\square$

Therefore, as written in [BMZ02, p.509], a good minimisation algorithm would not be attracted to stationary points which are not local minima. We construct such an alogrithm in section 4.

4 A heuristic algorithm for MCP

In this section we will describe the heuristic described by [BMZ02]. Again we assume non-negative edge weights, in order to mirror the analysis done in 2.

We begin by minimising f, i.e. finding a solution θ to 11. Note that θ is not necessarily an angular representation of a cut. Therefore we need to develop a method associating a cut $x \in \{-1,1\}^n$ to θ. Using polar coordinates we can associate the entries of θ to points on the unit circle. Furthermore we can assume $\theta_i \in [0, 2\pi)$ for all $i \in [1:n]$. As in section 2 we can generate a cut by cutting the unit circle into two halves. For any angle $\alpha \in [0, \pi)$ we can set

$$x_i = \begin{cases} 1 & \text{if } \theta_i \in [\alpha, \alpha+\pi) \\ -1 & \text{otherwise} \end{cases} \tag{20}$$

generating a cut. The weight of the obtained cut is given by

$$\gamma(x) = \frac{1}{2} \sum_{1 \leq i < j \leq n} w_{ij} \left(1 - x_i x_j\right)$$

Remark 1. Let $\theta \in [0, 2\pi)^n$ and $\alpha \in [0, \pi)$. Let x^α denote the cut generated by α and $x^{\alpha+\pi}$ the cut generated by $\alpha + \pi$. We then have $x^{\alpha+\pi} = -x^\alpha$. For $\alpha + \pi$ assignement 20 is to be understood as:

$$x_i = \begin{cases} 1 & \text{if } \theta_i \in [0, \alpha) \cup [\alpha+\pi, 2\pi) \\ -1 & \text{otherwise} \end{cases}$$

By observing $[0, 2\pi) = [\alpha, \alpha+\pi) \,\dot\cup\, ([0, \alpha) \cup [\alpha+\pi, 2\pi))$ we deduce $-x^\alpha = x^{\alpha+\pi}$.

2. Choosing the halfopen interval $[\alpha, \alpha+\pi)$ ensures, that diametrially opposed points are seperated, i.e. the corresponding vertices are on diferent sides of the cut. This is a positive feature: We denote x^{GW} .... and x^{B} the assignement obtained by 20
We have $x_i^{\mathrm{GW}} \neq x_i^{\mathrm{B}}$ if and only if $\theta_i = \alpha + \pi$ Because of this we will allow ourselves the slight inaccuracy and claim that 20 finds all the cuts. It can be understood as finds all the best possible Goemans-Williamson-type cuts.

By increasing α we have a simple and computationally inexpensive way to examine all cuts obtained by assignement 20. As we will describe in algorithm 2 we have a deterministic way of finding a best possible Goemans-Williamson-type cut.

As we are given a set of points on the unit circle, it is easy to examine all the possible cuts obtained by assignement 20. Thus a big difference to the algorithm in 2 is that we have a deterministic way of obtaining the best possible Goemans-Williamson-type cut associated to a given θ.

Actually we do not necessarily traverse all possible Goemans-Williamson-type cut as, diametrially opposed points are always separated.
However we should be able to argue that, we still find the best one, reasoning something like this:
The vectors are diametrially opposed for a good reason, as the points solve the vector program, and they could have been chosen differently.
This argument does not feel watertight anymore, as we do not solve the vector program optimally, but only obtain a local min.
Maybe this is good enough, as we may still be able to argue, that for a given set of points the diamatral opposition is optimal, if it weren't slightly moving one point could improve the cut. Contradiction to local minimum of f.

Before we describe the algorithm in detail, we will make two assumptions without loss of generality. First of all we note that by the 2π-periodicity in each variable of f, see 3.1 , we can assume $\theta_i \in [0, 2\pi)$ for all $i \in [1:n]$. Furthermore we assume $\theta_1 \leq \theta_2 \leq \cdots \leq \theta_n$.

Algorithm 2 Procedure-Cut

Input: Undirected weighted graph G
Output: A cut x^*

```
 1: Let x* trivial cut, i.e. all vertices on one side
 2: Let α = 0, Γ = −∞, i = 1. Let j be the smallest index such that θ_j > π if there is one;
    otherwise set j = n + 1. Set θ_{n+1} = 2π.
 3: while α < π do
 4:     Generate cut x by 20 and compute γ(x)
 5:     If γ(x) > Γ, then let Γ = γ(x) and x* = x.
 6:     if θ_i ≤ θ_j − π then
 7:         Let α = θ_i and increment i by 1.
 8:     else
 9:         Let α = θ_j − π and increment j by 1.
10:     end if
11: end while
12: return x*
```

Remark The assumption $\theta_1 \leq \theta_2 \leq \cdots \leq \theta_n$ is not a real constraint, we just have to keep track the original indices, as they determine the cut. Let σ be a permutation such that $\theta_{\sigma(1)} \leq \theta_{\sigma(2)} \leq \cdots \leq \theta_{\sigma(n)}$. We can now apply algorithm 2, with the only slight change that the cut in line 2.4 needs to be generated in the following manner:

$$x_i = \begin{cases} 1 & \text{if } \theta_{\sigma(i)} \in [\alpha, \alpha + \pi) \\ -1 & \text{otherwise} \end{cases}$$

Lemma 4.1 For a given set of points $\theta_1 \leq \cdots \leq \theta_n$ algorithm 2 returns a Goemans-Williamson-type cut of maximum weight.

Remark Intuitively, this can be argued by slightly rotating the dividing line (through the origin) in the mathematically negative sense. Just enough so that the points with angle $\alpha + \pi$ are separated from the points with angle α. But not so far, that any other points change their assignment. In the following we will make this precise.

Proof of Lemma 4.1. To show that the returned cut is a Goemans-Williamson-type cut, it suffices to show that for every $\alpha \in [0, \pi)$ assignement 20 can be achieved by a Goemans-Williamson-type cut. We begin by showing that th Assignement 20 states: $x_i = 1$ if $\theta_i \in [\alpha, \alpha + \pi)$. As this is a half open interval, we need to argue that assignment 20 can be achieved by a Goemans-Williamson-type cut:
Let $P = \{i \in [1 : n] \mid \theta_i \in [\alpha, \alpha + \pi)\}$, denote the indices that are assigned to $x_i = 1$. The indices that are assigned to $x_i = -1$ are denoted by $N := [1 : n] \setminus P$. We denote the smallest angle of points with indices in N to the point $\begin{pmatrix}\cos(\alpha) & \sin(\alpha)\end{pmatrix}^T$, by

$$\varepsilon_N := \min_{i \in N} \arccos\left(\begin{pmatrix}\cos(\theta_i) \\ \sin(\theta_i)\end{pmatrix}^T \begin{pmatrix}\cos(\alpha) \\ \sin(\alpha)\end{pmatrix}\right)$$

Likewise, we denote the smallest angle of points with indices in P to the other point

$\begin{pmatrix}\cos(\alpha+\pi) & \sin(\alpha+\pi)\end{pmatrix}^T$, by

$$\varepsilon_P := \min_{i \in P} \arccos\left(\begin{pmatrix}\cos(\theta_i)\\ \sin(\theta_i)\end{pmatrix}^T \begin{pmatrix}\cos(\alpha+\pi)\\ \sin(\alpha+\pi)\end{pmatrix}\right)$$

We have $\varepsilon_N, \varepsilon_P > 0$, since $\alpha \notin \{\theta_i \mid i \in N\}$ and $\alpha + \pi \notin \{\theta_i \mid i \in P\}$. Now we can choose $\varepsilon \in (0, \min\{\varepsilon_P, \varepsilon_N\})$ and consider the Goemans-Williamson-type cut generated by $\begin{pmatrix}\cos(\alpha + \frac{\pi}{2} - \varepsilon) & \sin(\alpha + \frac{\pi}{2} - \varepsilon)\end{pmatrix}$ This leads to the cut

$$x_i^{\mathrm{GW}} = \begin{cases} 1 & \text{if } \theta_i \in [\alpha - \varepsilon, \alpha + \pi - \varepsilon] \\ -1 & \text{otherwise} \end{cases}$$

In case of $\alpha - \varepsilon < 0$, we have $\alpha - \varepsilon > -\pi$, the interval $[\alpha - \varepsilon, \alpha + \pi - \varepsilon]$ is to be understood as $[0, \alpha - \varepsilon + \pi] \cup [2\pi + \alpha - \varepsilon, 2\pi)$. We need to verify $x = x^{\mathrm{GW}}$. As the variable is binary it suffices to show $x_i = 1$ implies $x_i^{\mathrm{GW}} = 1$. Let $x_i = 1$, then by construction of ε_P we have $\theta_i \in [\alpha, \alpha + \pi - \varepsilon_P]$. By construction of ε, we have $\theta_i \in [\alpha - \varepsilon, \alpha + \pi - \varepsilon]$ and thus $x_i^{\mathrm{GW}} = 1$. Similarly, for $x_i = -1$, we have $\theta_i \in [\alpha + \pi, 2\pi) \cup [0, \alpha - \varepsilon_N]$. By $0 < \varepsilon < \varepsilon_N$ we in particular have $\theta_i \in (\alpha + \pi - \varepsilon, 2\pi) \cup [0, \alpha - \varepsilon)$, which shows $x_i^{\mathrm{GW}} = -1$.

This interval may not make sense , i.e. if $\alpha - \varepsilon_N < 0$, in that case the interval is to be understood as $[\alpha + \pi, 2\pi +$

So far we have shown that the Goemans-Williamson-type cut between two .. cuts eval to the same
This doesnt show that all Goemans-Williamson-type cuts can be gen by 20. The ones missing are the ones where $\theta_i = \alpha, \theta_j = \alpha + \pi$ are in the same set. This is not a problem if all weigths are nonnegative, also the two being diametrially opposed is a better cut (as the other assignements remain unchanged). However if that happens to be a negative weight the cut we don't find might be better. This is import with regard to the results as we have instances, with neg weights. Also remember that GW algo assumes nonnegative weights.

□

2do: We are going to go into the changes:

1. The cut we obtain is not probabilistic but optimal, for the distribution of points

2. We did not solve the SDP so we arent at the global optimum

Lemma 4.2 Let $\theta \in \mathbb{R}^n$. Then the cut value generated by 2 is at least 0.878 times the relaxed cut value $\psi(\theta)$ as defined in 15 , written in a formula:

$$\gamma(x^*) \geq 0.878\psi(\theta) \tag{21}$$

Proof. As our rank-two relaxation has the form of the Goemans and Williamson relaxation with $d = 2$, essentially the same analysis applies. Let us quickly go over the changes:
As f is nonconvex, we can not expect to find the global minimum, only a local minimum. This destroys the performance guarantee of the algorithm. Let us show how to appropriately change this:
The local minium of f, say $\theta \in \mathbb{R}^n$ represents the points $v_i = \begin{pmatrix}\cos(\theta_i) & \sin(\theta_i)\end{pmatrix}^T$ for $i \in [1:n]$. As all these points are on the unit circle, $v_1, \ldots, v_n$ are a feasible solution to 7 and their value of the objective function is given by ...

CONTINUE

. This is closely linked with the fact that we do not solve the SDP. On the other hand, since we fixed $n = 2$, we have a methodical and efficient way to find the cut values of all possible Goemans-Williamson-type cut, see 4.1. □

This is not a performance guarantee, as we can not guarantee that $\psi(\theta)$ is an upper bound on the maximum cut value. Recall, that in Goemans-Williamson algorithm, the guarantee comes from solving the vector program The next Lemma can be interpreted as a weak performance guarantee

Vector Program not yet introduced

Lemma 4.3 Let x_a^* and x_b^* be two cuts generated by algorithm 2 from θ_a and θ_b respectively. If $\gamma(x_a^*) \leq \psi(\theta)$ and $\psi(\theta_b) > \psi(\theta_a)/0.878$, then x_b^* has a higher cut value than x_a^*.

Proof. Using the assumptions we confirm

$$\gamma(x_b^*) \overset{21}{\geq} 0.878 \cdot \psi(\theta_b) > \psi(\theta_a) \geq \gamma(x_a^*)$$

□

The next paragraph, describes the idea of algorithm 3. We minimise the function f and obtain a minimiser called θ^1. We can now start 2 and obtain a best possible cut x^1 associated with θ^1. At this point we can return the cut x^1. If θ^1 happens to be an angular representation of a cut we know that x^1 is a maximum cut, by theorem 3.10. Otherwise, we can continue to try and improve the cut value by spending more computational resources. By theorem 3.10 we know that the angular representation of x^1, denoted as $\theta(x^1)$, is a stationary point of f, most likely a sattle point. Therefore we can hope to find a deepr local minimum close to $\theta(x^1)$. As we use a gradient descend, restarting the minimisation from a stationary point is of no use. Thus we restart the minimisation of f from a slight perturbation of $\theta(x^1)$, in hopes of finding another local minimum $\theta^2 \neq \theta^1$ from which we hope to obtain a cut x^2 with a higher cut value than x^1, i.e. $\gamma(x^2) > \gamma(x^1)$. In case $\gamma(x^2) > \gamma(x^1)$ is achieved we deem our attempt succesful, and unsuccesful otherwise. We can set the termination criterion to be N consecutive unsuccesful attempts of improving the value of the cut.

Algorithm 3 ImprovedCut

Find a better name

Input: Undirected weighted graph G
Output: A cut x^*

procedure ImprovedCut(input N, θ^0)
    Given $\theta^0 \in \mathbb{R}^n$ and integer $N \geq 0$, let $k = 0$ and $\Gamma = -\infty$
    while $k \leq N$ **do**
        Starting from θ^0, minimize f to get θ
        Compute a best cut x associated with θ by 2
        if $\gamma(x) > \Gamma$ **then**
            Let $\Gamma = \gamma(x), x^* = x$ and $k = 0$
        else
            $k = k + 1$
        end if
        Set θ^0 to a random perturbation of the angular representation of x.
    end while
    return x^*
end procedure

In 3 we have θ^0 as input, which is the initial seed for the minimisatio of f. We can try and improve our chances of finding a high quality heuristic solution by increasing running 3 multiple times, in other words increase the number of seeds. The number of seeds, will be given by the input M.

The algorithm 3, can also be interpreted in the following way. From a seed we start minimising f. By means of Goemans-Williamson-type cuts, we look for nearby sattlepoints.

Write how increasing M and N usually leads to better solutions but the runtime can nnot be vorhergesagt. Also there can be lucky punches say start at the angular representation of a cut.

We assume that we do

Here it is important to note, that nearby is to be understood by the cuts that we can generate from the minimum. The so found sattlepoint is not necessarily the closest sattlepoint in the euclidean metric. The sattlepoint is chosen by the associated cut with the highest weight. From a slight perturbation of that point we search for a local minimum, which has a sattlepoint with lower f-value in its neighborhood.

Algorithm 4 BurerStub

Find better name?

Input: Undirected weighted graph G
Output: A cut x^*

```
Let x* trivial cut, i.e. all vertices on one side
for i ← 0, n do
    Generate random θ^0 ∈ ℝ^n
    x ← ImprovedCut()
    if γ(x) > γ(x*) then
        x* ← x
    end if
end for
return x*
```

reference of algorithms doesnt work! fix in whole file!

Generally speaking the bigger M and N are the longer the algorithm will run and the better the returned cut will be. However as the termination criterion in IMPROVEDCUT relies on the number of unsuccesful attempts there can always be exceptions. We will investigate the performance of the heuristic for a set of different choices for M and N.

5 Solving BQP with this heuristic

Closely following [JM21,] we are going to give a cursory explanation, that binary quadratic problems of the form: Let $Q \in \mathbb{R}^{n \times n}$

$$\begin{aligned} \text{minimize} \quad & x^T Q x \\ \text{subject to} \quad & x_i \in \{0,1\} \forall i \in [1:n] \end{aligned} \tag{22}$$

can be solved by solving the MCP on a suitable graph with appropriately chosen edge weights.

This should orient itself to Mallachs Paper. Something along the lines of This is a BQP we are interested in solving, this is the way to transform it to a MCP

Maybe begin by lemma showing that if there are no diagonal entries it is fairly simple

The goal of this section is to lay out the relationship between the BQP and MC.

choose right word if not section, will see later

To this end we formulate the MCP as an integer linear programming problem:

def opti

We have done a special case of this in 2.2, where Q is a strict upper diagonal matrix. However the following construction is much more general.

$$\begin{aligned} \text{minimize} \quad & \sum_{e \in E} w_e z_e \\ \text{subject to} \quad & \sum_{e \in F} z_e - \sum_{e \in C \setminus F} \leq |F| - 1 && F \subseteq C \subseteq E \text{ subject to } C \text{ cycle and } |F| \text{ odd}, \\ & 0 \leq z_e \leq 1 && \forall e \in E, \\ & z_e \in \mathbb{Z} && \forall e \in E \end{aligned} \tag{23}$$

give the intuition for odd cycle constraints

An optimal solution to 23 gives us the following maximum cut $\{\{i,j\} \mid z_{ij} = 1\}$

The cut polytope of a graph G can thus be denoted as

$$P^G_{\text{CUT}} := \left\{ z \in [0,1]^E \mid z \text{ is a feasible solution to } 23 \right\}$$

Without proof we state the following theorem proven in [De 90].

Theorem 5.1 Let $Q \in \mathbb{R}^{n \times n}$ and consider $G = (V, E)$ with $V := [1:n]$ and $E := \{\{i,j\} \mid i,j \in [1:n] : q_{ij} + q_{ji} \neq 0\}$. Define $H = (W, F)$ where $W := \{v_0, \dots, v_n\}$ and $F := \{\{v_i, v_j\} \mid \{i,j\} \in E\} \cup \{\{v_0, v_i\} \mid i \in [1:n]\}$. Let $f : \mathbb{R}^F \to \mathbb{R}^{V \cup E}$ be the bijective linear map defined by

$$\begin{aligned} x_v &= z_{0v} \text{ for all } v \in V \text{ , and} \\ y_{vw} &= x_v x_w = \frac{1}{2}\left(z_{0v} + z_{0w} - z_{vw}\right) \text{ for all } \{v,w\} \in E. \end{aligned}$$

Then $P^{\text{G}}_{\text{BQP}} = f\left(P^{\text{H}}_{\text{CUT}}\right)$.

This statement is not proven in Mallachs lecture notes, but the paper is: C. De Simone, “The cut polytope and the boolean quadric polytope”, Discrete Math. 79 (1989), 71–75

Using this Theorem we can show a fundamental result allowing us to solve BQP as MC. More specifically the BQP is of the form:

Corollary 5.2 Let $Q \in \mathbb{R}^{n\times n}$ and $E = \{\{i,j\} \mid i,j \in [1:n] \text{ such that } q_{ij}+q_{ji} \neq 0\}$. Then the corresponding BQP with the objective function $x^T Q x$ to be minimised can be solved as a MCP on $H = (W,F)$ where $W := \{v_0,\dots,v_n\}$ and $F := \{\{v_i,v_j\} \mid \{i,j\} \in E\} \cup \{\{v_0,v_i\} \mid i \in [1:n]\}$, with the objective function:

$$\max \sum_{\substack{\{v_i,v_j\}\in F\\ v_i\neq v_0\neq v_j}} \frac{1}{2}(q_{ij}+q_{ji})\, z_{ij} - \sum_{i\in W\setminus\{v_0\}} \widetilde{c}_i z_{0i}$$

where $\widetilde{c}_i := q_{ii} + \sum_{j=1}^{n} \frac{1}{2}(q_{ij}+q_{ji})\,[\{v_i,v_j\}\in F, v_i \neq v_0 \neq v_j]$ for all $i \in W \setminus \{v_0\}$

Proof. We have the following equality

$$\begin{aligned}
&xQx^T\\
&= \sum_{i,j=1}^{n} q_{ij}x_ix_j + \sum_{i=1}^{n} q_{ii}x_ix_i\\
&\overset{\substack{x_ix_i=x_i\\ \text{def } E}}{=} \sum_{\{i,j\}\in E} (q_{ij}+q_{ji})\, x_ix_j + \sum_{i=1}^{n} q_{ii}x_i\\
&\overset{5.1}{=} \sum_{\{i,j\}\in E} (q_{ij}+q_{ji}) \left(\frac{1}{2}(z_{0i}+z_{0j}-z_{ij})\right) + \sum_{i\in W\setminus\{v_0\}} q_{ii}z_{0i}\\
&= -\sum_{\{i,j\}\in E} \frac{1}{2}(q_{ij}+q_{ji})\, z_{ij} + \frac{1}{2}\sum_{\{i,j\}\in E} (q_{ij}+q_{ji})(z_{0i}+z_{0j}) + \sum_{i\in W\setminus\{v_0\}} q_{ii}z_{0i}\\
&= -\sum_{\{i,j\}\in E} \frac{1}{2}(q_{ij}+q_{ji})\, z_{ij} + \sum_{i\in W\setminus\{v_0\}} \underbrace{\left(q_{ii} + \frac{1}{2}\sum_{\substack{j\neq 0\\ \{v_i,v_j\}\in F}} (q_{ij}+q_{ji})\right)}_{=:\widetilde{c}_i} z_{0i}
\end{aligned}$$

Why does * hold?

The previous computation together with $P^{\mathrm{G}}_{\mathrm{BQP}} = f\left(P^{\mathrm{H}}_{\mathrm{CUT}}\right)$, by Theorem 5.1.

Finish this proof:
minimising xQx^T is the same as maximising ...

□

6 Experimental Setup

The goal of our experimental study is to compare the performance of the Burer heuristicfor a range of different parameters. We recall that

7 Results

The experiments gather further data that the Burer heuristicis highly effective in returning high quality approximations. Out of the 421 test instances with known optimum value, the instances for which the heuristic returns a cut with ratio less than 0.878 (the performance guarantee in Goemans Williamson) are so few we can name them here. They are name the instances In particular the heuristic return high quality cuts for instances with mixed sign weight, these are

find out number of instances with mixed sign

This is very impressive, as the performance guarantee of the Goemans Williamson algorithm only holds for graphs with nonnegative weights. Therefore there is no theoretical reason for the approximation algorithm to perform well, let alone the Burer heuristic.

We point out that the standard deviation of the returned cut values of the Burer heuristicwith different parameters is very low in general.

The data indicates, that if a run with big parameters, i.e. expecting longer runtime finds a very good solution, then a high quality solution can already be found with small values.

What does very good mean?

Looking at this from the perspective that the optimal value is not known, the following procedure might be an informed choice:
Run the Burer heuristicfor very small parameters, i.e $M = 1$ and $N = 5$

References

[Aig07] Martin Aigner. *Discrete mathematics*. American Mathematical Society, 2007.

[BMZ02] Samuel Burer, Renato D. C. Monteiro, and Yin Zhang. Rank-two relaxation heuristics for max-cut and other binary quadratic programs. *SIAM Journal on Optimization*, 12(2):503–521, 2002.

[De 90] Caterina De Simone. The cut polytope and the boolean quadric polytope. *Discrete Mathematics*, 79(1):71–75, 1990.

[GJS74] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified np-complete problems. In *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*, STOC '74, page 47–63, New York, NY, USA, 1974. Association for Computing Machinery.

[GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, nov 1995.

[JM21] Michael Jünger and Sven Mallach. Exact facetial odd-cycle separation for maximum cut and binary quadratic optimization. *INFORMS Journal on Computing*, 33(4):1419–1430, 2021.

[KV18] Bernhard Korte and Jens Vygen. *Approximation Algorithms*, pages 423–469. Springer Berlin Heidelberg, Berlin, Heidelberg, 2018.

[Vaz03] Vijay V. Vazirani. *Semidefinite Programming*, pages 255–269. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

Notes