Maximizing Quadratic Programs

Fabian Andres Prada Nino

December 12, 2013

Abstract

In the first half of this report I present a brief review of MAXQP and a description of my greedy and SDP approaches, as well as some other attempted ideas. In the second part, I focus on MAX2CORR. Particularly, **I present a probably good approximation of MAX2CORR in cubic graphs**. This is obtained by adapting Zwicks[8] CSDP+LI approach of MAXCUT in cubic graphs. This approximation factor should be slightly improved using tighter bounds, and it was calculated using computer assisted techniques[1]. The new factor outperforms the 1/3 approximation factor obtained independently from the Greedy and the SDP approach.

1 Formulation

Our formulation is based in **MAXQP**[1]:

Problem 1. *Let A be a real $n \times n$ zero-diagonal matrix.*
Maximize

$$q_A(\boldsymbol{x}) = \boldsymbol{x}^T A \boldsymbol{x}$$

Subject to

$$\boldsymbol{x} \in \{-1, 1\}^n$$

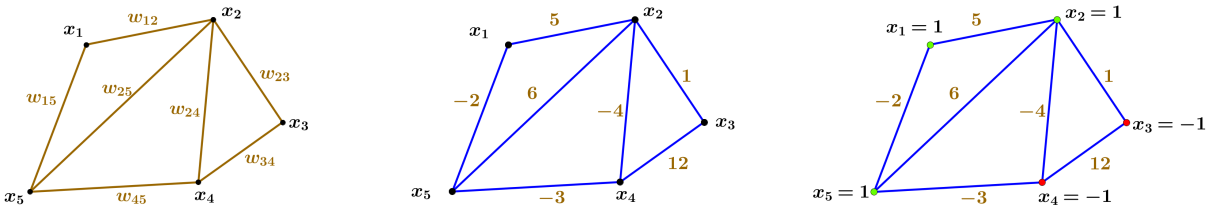
Observations:

- Adding diagonal terms to A increases the value of the objective function by the constant term $\sum_{i=1}^{n} d_i$. Therefore, the optimality of a vector $\boldsymbol{x}^*$ is invariant to the diagonal values. This justifies the formulation of the problem using a zero-diagonal matrix.
- The matrix A is not necessarily symmetric. Since $\boldsymbol{x}^T A \boldsymbol{x} = \frac{1}{2}\boldsymbol{x}^T(A + A^T)\boldsymbol{x}$, we get an equivalent energy function defined for the symmetric matrix $A = \frac{1}{2}(A + A^T)$. WLOG I will assume that A is symmetric.
- The matrix A can be associated to a **real-weighted** loop free graph. Specifically, we will associate matrix A with the undirected graph G_A with weights $w_{ij} = a_{ij} + a_{ji}$ for the edge (ij) (see Figure 1). Then, our objective function can be written as the sum of edge weights multiplied by the vertices signs:

[1]My current approximation factor is 0.8221. This follows an standard approach used by Zwick[10][8] to get a 7/8-approximation of MAX3SAT, and a 0.9326-approximation of MAXCUT in cubic graphs. The approximation factor was obtained from my naive MATLAB implementation, which still does not perform a detailed domain search due to the large computational time. Any way, the real factor can be accurately computed as was done in the previously cited examples. I think the real factor should not be too far from the one I computed. This seems to be a promising result, and may be this could provide a good bound of MAX2CORR on cubic graphs.

$$q_A(\boldsymbol{x}) = \sum_{i,j} a_{ij} x_i x_j = \sum_{(ij)\in E(G_A)} w_{ij} x_i x_j \tag{1}$$

We said that an edge is satisfied whenever $\operatorname{sign}(w_{ij}) = \operatorname{sign}(x_i x_j)$.



(a) Graph G_A (b) Problem Instance (c) Optimal Solution

Figure 1: The value of OPT is 27. Observe that for any optimal solution $\boldsymbol{x}^*$, we have that $-\boldsymbol{x}^*$ is also optimal. In this case the edges $(2, 5)$ and $(2, 3)$ are the only not satisfied.

1.1 Well Defined

In order to prove that MAXQP is a well defined problem we need to check $\mathrm{OPT}_{\mathrm{QP}} \geqslant 0$. This follows from this simple claim

Proposition 1. *Let A be a real $n \times n$ zero-diagonal matrix. Choose $\boldsymbol{x}$ uniformly at random in $\{-1, 1\}^n$. Then $E(q_A(\boldsymbol{x})) = 0$*

Proof. Since the variables are independent, and all satisfy $E(x_i) = 0$, we get,

$$E(q_A(\boldsymbol{x})) = \sum_{i,j} a_{ij} E(x_i x_j) = \sum_{i,j} a_{ij} E(x_i) E(x_j) = 0$$

□

The result above is sufficient to prove that our problem is well defined but does not give a significant lower bound. A larger lower bound of MAXQP can be obtained using matchings[2]

Proposition 2. *Let G_A be the graph associated to A, and $G_{|A|}$ the graph obtained by taking the absolute value at each edge. Let $\mathrm{MaxMatch}(G_{|A|})$ denote the maximum sum of edge weights in a match of $G_{|A|}$. Then we have $\mathrm{OPT}_{\mathrm{QP}} \leqslant \mathrm{MaxMatch}(|G_{|A|})$*

Proof. Let $(i_1, j_1), (i_2, j_2), \ldots, (i_{\lfloor n/2 \rfloor}, j_{\lfloor n/2 \rfloor})$ be a maximum weighted match. For all pairs define the following random assignations:

If $w_{(i_k, j_k)} > 0$ then assign

$$\begin{cases} x_{i_k} = x_{j_k} = 1 & \text{with probability } 1/2 \\ x_{i_k} = x_{j_k} = -1 & \text{with probability } 1/2 \end{cases}$$

If $w_{(i_k, j_k)} < 0$ then assign

$$\begin{cases} x_{i_k} = 1, x_{j_k} = -1 & \text{with probability } 1/2 \\ x_{i_k} = -1, x_{j_k} = 1 & \text{with probability } 1/2 \end{cases}$$

[2]This is just an adaptation of the result presented in [1]

Then,

$$\begin{aligned} E(q_A(\boldsymbol{x})) &= \sum_{(i,j) \text{ match edge}} |w_{ij}| E(x_i x_j) + \sum_{(i,j) \text{ non match edge}} |w_{ij}| E(x_i x_j) \\ &= \sum_{(i,j) \text{ match edge}} |w_{ij}| E(1) + \sum_{(i,j) \text{ non match edge}} |w_{ij}| E(x_i) E(x_j) \\ &= \sum_{(i,j) \text{ match edge}} |w_{ij}| \end{aligned} \tag{2}$$

□

1.2 Relation to Other Problems

Real Weighted MAXCUT: Let's denote $W := \sum_{i,j} a_{ij} = \sum_{(ij) \in E(G_A)} w_{ij}$ the total weight of the graph. Let $\delta_{G_A}(\boldsymbol{x}) := \sum_{x_i \neq x_j} a_{ij} = \sum_{(ij) \in E(G_A): x_i \neq x_j} w_{ij}$ be the value of the cut on G_A induced by $\boldsymbol{x}$. We have the following relation:

$$\begin{aligned} q_A(\boldsymbol{x}) &= \sum_{i,j} a_{ij} x_i x_j \\ &= \sum_{i,j} a_{ij} - 2 \sum_{x_i \neq x_j} a_{ij} \\ &= W - 2\delta_{G_A}(\boldsymbol{x}) \end{aligned}$$

Denote by G_{-A} the graph with identical connectivity of A but edges with opposite sign. Then we get

$$\max_{\boldsymbol{x}} q_A(\boldsymbol{x}) = W + 2 \max_{\boldsymbol{x}} \delta_{G_{-A}}(\boldsymbol{x}) \tag{3}$$

This equation provide a relation between real-weighted MAXCUT and MAXQP

MAXCUT: This is a particular case of the relation provided by equation 3, when all the edges in G_{-A} have positive weight (i.e., if all the w_{ij}'s are negative). As we will see further, the GW[6] 0.878-approximation does not provide any approximation of MAXQP.

2-Clustering Problems:

For the next three problems, we assume that $G_A(V, E)$ is a graph with edge weigths in $\{-1, 1\}$ and $\boldsymbol{x} \in \{-1.1\}^{|V|}$ defines a partition of the vertices set in two clusters. Define $W^+ := \sum_{(ij) \in E(G_A)} |w_{ij}| = |E(G_A)|$.

MAX-2AGREE[3]: Define $Agr_{G_A}(\boldsymbol{x}) = |\{(ij) \in E(G_A) : w_{ij} x_i x_j = 1\}|$. It follows that

$$q_A(\boldsymbol{x}) = Agr_{G_A}(\boldsymbol{x}) - (W^+ - Agr_{G_A}(\boldsymbol{x})) = 2Agr_{G_A}(\boldsymbol{x}) - W^+$$

Thus

$$\max_{\boldsymbol{x}} q_A(\boldsymbol{x}) = 2 \max_{\boldsymbol{x}} Agr_{G_A}(\boldsymbol{x}) - W^+ \tag{4}$$

MIN-2DISAGREE: Define $Dis_{G_A}(\boldsymbol{x}) = |\{(ij) \in E(G_A) : w_{ij} x_i x_j = -1\}|$. It follows that

[3]This is equivalent to MAX-2XOR

$$q_A(\boldsymbol{x}) = (W^+ - Dis_{G_A}) - Dis_{G_A}(\boldsymbol{x}) = W^+ - 2Dis_{G_A}(\boldsymbol{x})$$

Thus

$$\max_{\boldsymbol{x}} q_A(\boldsymbol{x}) = W^+ - 2\min_{\boldsymbol{x}} Dis_{G_A}(\boldsymbol{x}) \tag{5}$$

MAX-2CORR: Define $Cor_{G_A}(\boldsymbol{x}) = Agr_{G_A}(\boldsymbol{x}) - Dis_{G_A}(\boldsymbol{x})$. From the previous definitions it follow that,

$$Cor_{G_A}(\boldsymbol{x}) = \frac{1}{2}(q_A(\boldsymbol{x}) + W^+) - \frac{1}{2}(W^+ - q_A(\boldsymbol{x})) = q_A(\boldsymbol{x})$$

Thus

$$\max_{\boldsymbol{x}} q_A(\boldsymbol{x}) = \max_{\boldsymbol{x}} Cor_{G_A}(\boldsymbol{x}) \tag{6}$$

Proposition 3. *Constant approximations to MAXCUT or MAX-2AGREE does not provide any approximation of MAXQP. In fact, they do not guarantee positivity of MAXQP.*

Proof. I present the analysis for MAXCUT and for MAX-2AGREE is analogous. Let $-K_{2n}$ be the complete graph of $2n$ vertices with all edges of weight -1. From equation 3, we get that $\mathrm{OPT}_{\mathrm{QP}}(-K_{2n}) = W + 2\,\mathrm{OPT}_{\mathrm{MAXCUT}}(K_{2n})$. For this graph we have $W = -\frac{2n(2n-1)}{2} = -2n^2+n$, and it is easy to check $\mathrm{OPT}_{\mathrm{MAXCUT}}(K_{2n}) = n^2$. Suppose we have an algorithm providing a constant $\alpha < 1$ approximation to MAXCUT. Computing the value of the quadratic function from such approximation, we get $q(\boldsymbol{x}) = n - 2(1-\alpha)n^2$, which is negative when $n \to \infty$.

□

1.3 Known Results

The following are the main results known about particular and general instances of MAXQP:

1. For planar graphs OPT of MAXQP can be computed in polynomial time. This was proven by Hadlock [11] in the context of real weighted MAXCUT, so can be adapted for MAXQP by the equivalence described in the previous section. Hadlock reduce the problem to find a maximum weigthed matching in an expanded graph, which can be computed in polynomial time from Edmonds algorithm.

2. Charikar and Wirth [1] provide $\Omega(\frac{1}{\log n})$-approximation for the general instance of MAXQP. This is the best approximation factor known about this problem (to my extent), and was obtained from a standard SDP relaxation (defined below) and a thorough rounding technique. This $\Omega(\frac{1}{\log n})$ is also the best approximation factor known for MAXCORR.
   Standard SDP: Maximize $\sum_{i,j} a_{ij} v_i \cdot v_j$ s.t. $v_i \in \mathbb{R}^n, ||v_i|| = 1$.

3. Alon and Naor [2] present a detailed analysis of the gap between the integer and vector solutions of the standard SDP. They prove that this gap depends on the graph connectivity. Let $K(G)$ be the maximum gap between the optimal integer solution and the optimal vector solution over all possible edge weights assignations on G. The authors prove:

   $$\Omega(\log \omega(G)) = K(G) = O(\log \chi(G))$$

   where $\omega(G)$ = maximum clique, and $\chi(G)$=chromatic number. If G is complete we get $K(G) = \theta(\log n)$. Therefore, the approximation factor of Charikar and Wirth is of the best order that can be obtained from the standard SDP.

4. Charikar and Wirth [1] prove that it is NP-hard to approximate MAXQP within a factor of $\frac{11}{13} + \epsilon$.

2 Personal approaches to MAXQP in general instances

2.1 Greedy

We formulate the greedy approach having in mind the graph interpretation described in the first section. For each vertex i, we define the value of its ring as $R_i = \sum_{j \in N(i)} w_{ij} x_j$. Then $q(\boldsymbol{x}) = \frac{1}{2} \sum_i R_i x_i$ since each edge is counted in two rings.

GreedyMAXQP
$\boldsymbol{x}$ initialized at random
$S = \{i : R_i x_i < 0\}$
while $(S \neq \varnothing)$
Pick $i \in S$
$x_i \leftarrow -x_i$
Update S
end while
return $\boldsymbol{x}$

The following conclusions about general MAXQP can be derived from the greedy algorithm:

- The optimal value for MAXQP in $[-1, 1]^n$ is the same than in $\{-1, 1\}^n$: Let $\boldsymbol{y} \in [-1, 1]^n$ be an optimal solution. If y_i is a non integer coordinate we must have $R_i = 0$ (otherwise, we can improve the solution by taking $y_i = \text{sign}(R_i)$). From this condition it follows that any assignation to $\{-1, 1\}$ of the non integer coordinates of $\boldsymbol{y}$ is a solution of the integer problem with identical optimal value.
- GreedyMAXQP converges to a local optimal solution: The algorithm produces a sequence of monotonically increasing values. The finite number of configurations (2^n) and the monotonicity guarantees termination in finite number of steps. I did not find a prove of polynomial time termination for general instances of the problem. Also, no guarantee was found on the approximation factor of the local optimal to OPT_{QP}.
- $R_i x_i^* \geqslant 0$ for all i in the OPT_{QP} and the local optima of GreedyMAXQP : If $\boldsymbol{x}$ is a solution that does not satisfy this condition for some i, a better result can be obtained by taking $x_i = \text{sign}(R_i)$.
- Using the previous, result and defining $m_i = \min_{\boldsymbol{x} \in \{-1,1\}^n} |\sum_{j \in N(i)} w_{ij} x_j|$, we get

$$OPT_{QP} \geqslant \frac{1}{2} \sum_{i=1}^{n} m_i$$

  This provides other proof of the positivity of MAXQP, and similar bound to Proposition 2.
- For the particular case of correlation cubic graphs (see definition in section 3), each $m_i \geqslant 1$, so we get, $OPT_{QP} \geqslant \frac{1}{2} \sum_{i=1}^{n} m_i \geqslant \frac{n}{2}$, where n is the number of vertices. Observe that the total number of edges in a cubic graph is $\frac{3n}{2}$, so this is also a upper bound of OPT_{QP}. In this case it is easy to check that the local optima of GreedyMAXQP satisfies $R_i x_i^* \geqslant 1$ (by

the same reason $m_i \geqslant 1$), so we conclude that $\frac{1}{3}\,\mathrm{OPT_{QP}} \leqslant \frac{1}{3}(\frac{3n}{2}) = \frac{n}{2} \leqslant GreedyMAXQP$. Therefore, GreedyMAXQP provides a $\frac{1}{3}$-approximation of MAXQP in correlation cubic graphs.

2.2 SDP

In Charikar and Wirth[1], the projection-rounding technique used on the solution of the standard SDP relaxation is based in a generalized approach introduced in [4]. Specifically, this rounding technique provide an approximation:

$$E(q(\boldsymbol{x})) \geqslant \mathrm{OPT_{QP}}\left(\frac{1}{4\log n} - 8\frac{1}{n}\right),$$

which is valid for large n.

For comparison, I compute the expected result using the standard hyperplane rounding technique [6]: Let $v_1, \ldots, v_n$ be the optimal solution of the standard SDP relaxation, and n a random vector chosen uniformly on S^{n-1}. Let $\boldsymbol{x}$ be the rounded solution, defined as $x_i = 1$ if $v_i \cdot n > 0$ and $x_i = -1$ otherwise. Then, it can be checked:

$$E(x_i x_j) = 1(\frac{\pi - \arccos(v_i \cdot v_j)}{\pi}) + (-1)(\frac{\arccos(v_i \cdot v_j)}{\pi}) = \frac{\pi - 2\arccos(v_i \cdot v_j)}{\pi} \tag{7}$$

We can write $\frac{\pi - 2\arccos(\mu)}{\pi} = \mu + \epsilon(\mu)$, where $\epsilon(\mu) \in [-0.22, 0.22]$ as can be observed from Figure 2. Defining $W^+ = \sum_{ij} |a_{ij}|$, we get

$$E(q(\boldsymbol{x})) = \sum_{ij} a_{ij} E(x_i x_j) = \sum_{ij} a_{ij}(v_i \cdot v_j + \epsilon(v_i \cdot v_j)) \geqslant OPT_{QP} - 0.22W^+$$

If $\alpha OPT_{QP} > W^+$ with $1 \leqslant \alpha < 4.5$, we get a constant approximation,

$$E(q(\boldsymbol{x})) \geqslant (1 - 0.22\alpha) OPT_{QP} \tag{8}$$

In general the ratio $\frac{\mathrm{OPT_{QP}}}{W^+}$ can be done arbitrarily small as shown in Proposition 3. So the result of equation 8 only can be applied to limited instances. One of such instances are correlation cubic graphs (see definition in section 3). From the greedy algorithm, we already know that $3\,\mathrm{OPT_{QP}} \geqslant W^+$. Plugging this in equation 8 (using $\alpha = 3$), we obtain that the standard hyperplane rounding of the SDP also provide a $1 - 0.22 * 3 \approx \frac{1}{3}$ approximation of $\mathrm{OPT_{QP}}$.

2.3 Other Approaches

2.3.1 LP

Observe that for any $\boldsymbol{x} \in \{-1, 1\}^n$, the energy function

$$q_A(\boldsymbol{x}) = \sum_{i,j} a_{ij} x_i x_j = \sum_{i,j} a_{ij}(1 - |x_i - x_j|) = \sum_{i,j} a_{ij} - \sum_{i,j} a_{ij}|x_i - x_j|$$

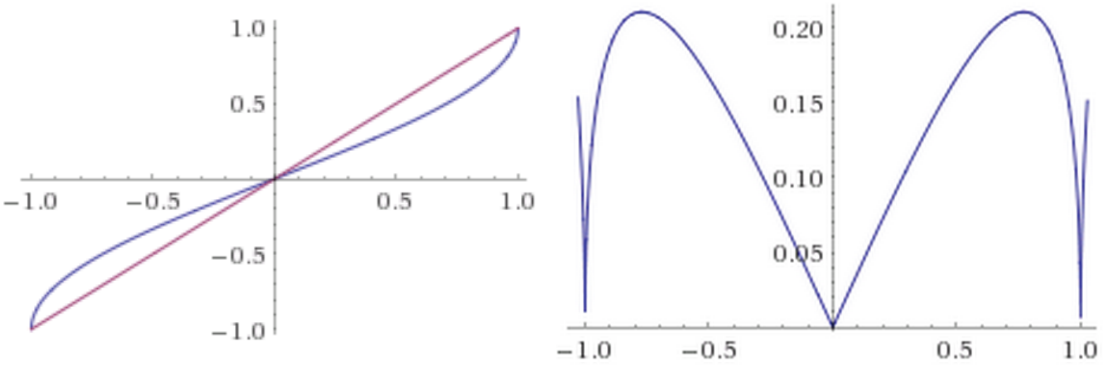
Therefore, in the integer set, computing the optimal of MAXQP is equivalent to solve

Problem 2. *Minimize*

$$\sum_{i,j} a_{ij}|x_i - x_j|$$

Subject to,

$$\boldsymbol{x} \in \{-1, 1\}^n$$



(a) Comparison (b) Absolute Difference

Figure 2: (a) Plots $\frac{\pi - 2\arccos(\mu)}{\pi}$ vs μ. (b) Plots the absolute value of the difference.

The problem above is not even an IP, due to the non-linearity of the objective function. At first glance, this problem could resemblance the approach to facility location presented in class, where an intial nonlinear formulation of problem (using linear combinations of l_1 norms of vectors) is transformed to a LP. However, we will show that in this case such transformation is not possible.

We would like to introduce variables z_{ij} such that $z_{ij} = |x_i - x_j|$. So the problem now is stated as

Minimize

$$\sum_{i,j} a_{ij} z_{ij}$$

Subject to,

$$z_{ij} = |x_i - x_j|$$
$$\boldsymbol{x} \in \{-1, 1\}^n$$

The inequality $z_{ij} \geqslant |x_i - x_j|$ can be done using the trick :

$$z_{ij} \geqslant |x_i - x_j| \iff \begin{cases} z_{ij} \geqslant x_i - x_j \\ z_{ij} \geqslant x_j - x_i \end{cases}$$

On the other hand, a formulation of $z_{ij} \leqslant |x_i - x_j|$ can not be attained in a direct way. Lets decompose the energy function in two terms:

$$\sum_{i,j} a_{ij} z_{ij} = \sum_{a_{ij}>0} a_{ij} z_{ij} + \sum_{a_{ij}<0} a_{ij} z_{ij}$$

For the pairs in $\sum_{a_{ij}>0} a_{ij}|x_i - x_j|$, the condition $z_{ij} \geqslant |x_i - x_j|$ is enough to guarantee $z_{ij} = |x_i - x_j|$ at the optimal solution. Analogously, for the pairs in $\sum_{a_{ij}<0} a_{ij}|x_i - x_j|$, the condition $z_{ij} \leqslant |x_i - x_j|$ would be enough to guarantee $z_{ij} = |x_i - x_j|$ at the optimal solution.

My attempt to express $z_{ij} \leqslant |x_i - x_j|$ as a set of linear inequalities was based in the following idea:

$$z_{ij} \leqslant |x_i - x_j| \iff \begin{cases} z_{ij} \leqslant 2|x_i - x_j| - (x_i - x_j) \\ z_{ij} \leqslant 2|x_i - x_j| - (x_j - x_i) \end{cases}$$

Then, I a introduced new variable u_{ij} for all negative a_{ij}'s. The formulation for the part of the problem associated only to the negatives a_{ij}'s is as follows:

Minimize

$$\sum_{a_{ij}<0} a_{ij} z_{ij} + \sum_{a_{ij}<0} p_{ij} u_{ij}$$

Subject to,

$$u_{ij} \geqslant x_i - x_j \text{ if } a_{ij} < 0$$

$$u_{ij} \geqslant x_j - x_i \text{ if } a_{ij} < 0$$

$$z_{ij} \leqslant 2u_{ij} - (x_i - x_j) \text{ if } a_{ij} < 0$$

$$z_{ij} \leqslant 2u_{ij} - (x_j - x_i) \text{ if } a_{ij} < 0$$

$$\boldsymbol{x} \in \{-1, 1\}^n$$

I associate to each u_{ij} a positive weights p_{ij} in the objective function. The intuition is that the objective function should force u_{ij} to approximate $|x_i - x_j|$. If u_{ij} indeed approximate $|x_i - x_j|$, then the objective function should also force z_{ij} to approximate $|x_i - x_j|$. However, since both terms $\sum_{a_{ij}<0} a_{ij} z_{ij}$ and $\sum_{a_{ij}<0} p_{ij} u_{ij}$ are optimized simultaneously, my two step argument does not hold. In fact, the previous formulation is equivalent to

Minimize

$$\sum_{a_{ij}<0} (2a_{ij} - pij) u_{ij} - \sum_{a_{ij}<0} |x_i - x_j|$$

Subject to

$$u_{ij} > |x_i - x_j|$$

$$\boldsymbol{x} \in \{-1, 1\}^n$$

If $2a_{ij} - pij < 0$ the global problem is unbounded (by taking $u_{ij} \to \infty$). If $2a_{ij} - pij \geqslant 0$ then the optimal solution for the global problem is given by $x_i = x_j = 1$ and $u_{ij} = 0$, what are non relevant values.

2.3.2 Sampling And Consensus

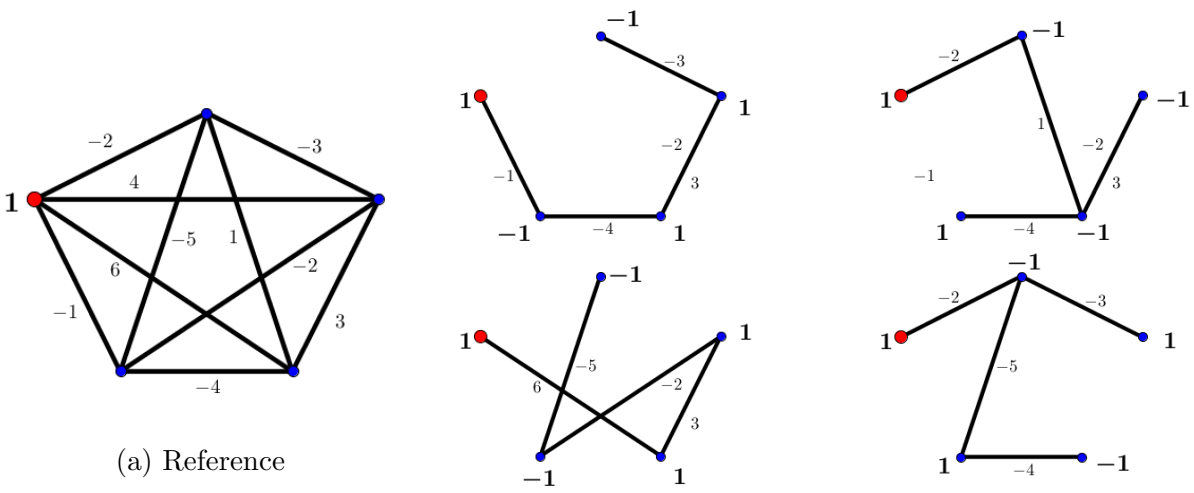
Other approach we attempted for MAXQP was based is sampling and consensus. The idea is that a complete graph[4] can be decomposed in a basis of as chains or trees, and the optima of the problem restriction to this kind of subgraphs is easily computed (see Figure **??**).

The naive algorithm proposed is as follows (and illustrated in Figure 3):

Sampling+Consensus
Fix a vertex v_1 to be positive.
For all vertices define $p_i = n_i = 0$
for $i = 1 : numiterations$
Let S_i be a subgraph of G, where S_i is either a chain or a tree containing v_1.
Compute the optimal solution of the problem restricted to S_i taking $v_i = 1$.
If vertex i is positive in the restricted solution, set $p_i \leftarrow p_i + 1$.
If vertex i is negative in the restricted solution, set $n_i \leftarrow n_i + 1$.

[4]WLOG we can assume the graph complete by taking weight equal to zero is absent edges.

end for
If $p_i > n_i$ set $x_i = 1$
Else set $x_i = -1$
return $\boldsymbol{x}$



(a) Reference

(b) Chains

(c) Trees

Figure 3: By fixing the sign of one of the vertices the optimal solution of a tree or a chain is unique.

The intuition behind this approach is that we can write $q(\boldsymbol{x}) = \frac{1}{\text{Edge Rep.}} \sum_{c_i \in \text{Chains}} c_i(\boldsymbol{x})$ and $q(\boldsymbol{x}) = \frac{1}{\text{Edge Rep.}} \sum_{t_i \in \text{Trees}} t_i(\boldsymbol{x})$. What **Sampling+Consensus** naively try to do is to find a vertex assignation that maximizes the larger number of chains or trees. However since the best assignation is computed for each vertex independently I was not able to provide a global guarantee.

It would be interesting to see how this method works in practice. Some modification may include update p_i and n_i using a value dependent of the particular subgraph (e.g., the absolute value of the minimum edge),and a probabilistic rounding from the sampling results. Cycles and small cliques are other interesting type of subgraphs that may be considered.

3 Probably Good Approximation Factors of MAX-2CORR in cubic graphs

In the previous section we presented a $\frac{1}{3}$-approximation of MAXQP in correlation cubic graphs (now ahead I will refer this problem as MAX-2CORR in cubic graphs). In this section I present a probably better approximation using a constrained SDP problem followed by local improvement steps. For clarity lets restate the basic definitions:

*A **correlation cubic graph** is graph of degree 3 at all its vertices, and edge weights w_{ij} in $\{-1, 1\}$. **MAX-2CORR** is the problem of finding a vertex assignation x_i in $\{-1, 1\}$, such that $\sum_{(ij) \in E(G)} w_{ij} x_i x_j$ is maximized.*

Let introduce the basic ideas that motivated my approach:

Feige et.al. [5] presented an improvement of the GW[6] 0.878-approximation of MAXCUT for general graphs, to a 0.921-approximation in the particular case of cubic graphs. The authors presents two fundamental ideas: (1) the inclusion of a set of constraints in the SDP and (2) a set of local improvement steps. The set of constraints added to the SDP are satisfied by the integer problem, thus reducing the integral gap between the integer solution and the vector solution. The local improvement steps increases the value of the integer solution provided by the rounded vector solution, thus also reduces the integral gap.

A further improvement was attained by Zwick et.al.[10] by formulating the cut problem as a particular instance of MAX-2XOR, and providing a 0.9326-approximation of MAX-2XOR in cubic graphs. The approach of Zwick et.al., follows closely the one of Feige et.al.[5], but there are three notorious differences: (1) It uses a different objective function (the 2-XOR, instead of the CUT function), (2) It introduces new constraints (reducing further the integral gap), and (3) The local improvement step is simpler and may provides larger increases.

Here I present my adaptation of these techniques to the MAX-2CORR problem. By simplicity, I will be following very closely the approach of Zwick et.al. rather than Feige et.al.

The algorithm can be described in three steps:

CSDP+LI MAX-2CORR

1. Solve the constrained SDP.
2. Round the vector solution using a random hyperplane.
3. Improve the solution using local steps.

(1)Solve the constrained SDP.

Problem 3. *Maximize*

$$\sum_{a_{ij} \in E} a_{ij} v_i \cdot v_j$$

Subject to:

$$\begin{aligned}
v_i \cdot v_j + v_i \cdot v_k + v_j \cdot v_k &\geqslant -1, \quad (I1), && 1 \leqslant i,j,k \leqslant n \\
v_i \cdot v_j - v_i \cdot v_k - v_j \cdot v_k &\geqslant -1, \quad (I2), && 1 \leqslant i,j,k \leqslant n \\
v_i \cdot v_j + v_i \cdot v_k + v_j \cdot v_k &= -1, \quad (E1), && \text{if } a_{ij} = -1, a_{ik} = -1, \\
-v_i \cdot v_j - v_i \cdot v_k + v_j \cdot v_k &= -1, \quad (E2), && \text{if } a_{ij} = 1, a_{ik} = 1, \\
v_i \cdot v_j - v_i \cdot v_k - v_j \cdot v_k &= -1, \quad (E3), && \text{if } a_{ij} = -1, a_{ik} = 1, \\
v_i \in \mathbb{R}^n, ||v_i|| &= 1, \quad (V), && 1 \leqslant i \leqslant n
\end{aligned} \tag{9}$$

To understand where these constrains come from lets introduce the function $f : \{-1,1\}^3 \rightarrow \{3,-1\}$, where $f(x_i,x_j,x_k) = x_ix_j + x_ix_k + x_jx_k$. We must notice two properties of f:

1. $f(x_i,x_j,x_k) = 3$ just for the triples $(1,1,1)$ and $(-1,-1,-1)$. $f(x_i,x_j,x_k) = -1$ for the remain 6 triples.
2. $f(-x_i,x_j,x_k) = -x_ix_j - x_ix_k + x_jx_k$ and $f(x_i,x_j,-x_k) = x_ix_j - x_ix_k - x_jx_k$, are the integer equivalent expression for constrains $E2$ and $E3$. From the previous item we also get that there are 6 triples satisfying $f(-x_i,x_j,x_k) = -1$ and 6 triples satisfying $f(-x_i,x_j,x_k) = -1$

The constrains $(I1)$ and $(I2)$ can be called soft constrains and as we have seen, they are satisfied by any triple $(x_i, x_j, x_k) \in \{-1, 1\}^3$. In particular, the optimal integer solution satisfy these constrain, so adding them to the SDP wont increase the integral gap.

Constrains $(E1)$,$(E2)$, and $(E3)$ can be called hard constrains, and capture specific properties of the integer optimal. We say that an *arc* is a pair of edges (ij, ik) with i as a vertex in common. These hard constrains discard the configurations where both edges in the arc are unsatisfied (this is, discarding two of the eight possible signs assignation in each case). The cases discarded are shown in Figure 5. Since any vertex in the integer OPT has at most one unsatisfied adjacent edge (this follows from the greedy algorithm), all these constrains are satisfied by the integer OPT, so adding them to the SDP wont increase the integral gap.

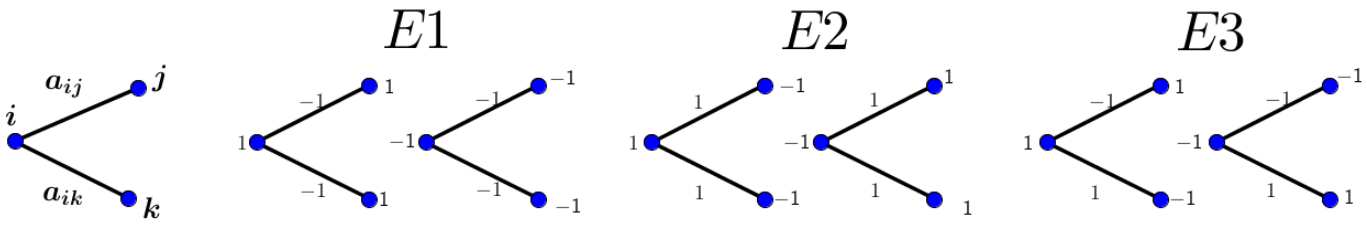


Figure 4: Configurations discarded by the hard constrains.

(2)Round the vector solution using a random hyperplane.

The optimal vector solution of the SDP is rounded using standard random hyperplane techniques. As was shown in section 2, the rounded integer solution satisfies:

$$E(x_i x_j) = \frac{\pi - 2\arccos(v_i \cdot v_j)}{\pi}$$

Where v_i's are the optimal vector solution of the SDP.

(3)Improve the solution using local steps.

Once we obtain an integer approximation from SDP, we proceed to improve this solution using a sequence of transformations. For clarity, I present a simple algorithm of sequential transformations and give some rough (but still significant) guarantees on its improvements[5]. The Zwick analysis provide finer bounds on the guarantees but require thorough work that wont be described here.

Let V_i be the set of vertices with i unsatisfied edges.

Local Improvement Algorithm
s_0 = Rounded integer solution from SDP
while improvements are possible (give priority to (a)):
(a) Change the sign of a vertex in V_3 with minimum number of neighbours in V_3.
(b) For a path of unsatisfied edges through the vertices $x - u_1 - u_2 - \ldots - u_k - y$, with $x, y \notin V_2$,

[5]The improvement algorithm is the same than Zwick's. However, for the analysis of the guarantees I am not assuming an initial reduction of the graph, or separating triangle free and non triangle free cases.

change the sign of $u_i's$ with i odd.
(c) For a cycle of unsatisfied edges through the vertices $u_1 - u_2 - \ldots - u_k - u_1$, change the sign of $u_i's$ with i even.
end while

The following proposition states the guarantees provided by this algorithm:

Proposition 4. *Let s be an initial signed vertex assignation obtained by the rounded SDP. Then the Local Improvement Algorithm increases the solution by at least $\frac{2}{3}|V_2| + \frac{20}{9}|V_3|$.*

Proof. The prove follows Zwick's ideas. If $q(s) = OPT$, then $V_2 = V_3 = 0$, and the condition trivially holds in this case. Now we proceed by decreasing induction on the value of the quadratic function: Let s be an assignation of value $OPT - k$ and assume that the proposition is valid for all assignation of larger value.

Observe that if all vertices in V_3 has 3 neighbours in V_3, then $V = V_3$ (i.e, all vertices are in V_3 assuming V is connected). This configuration provides the smallest value of the quadratic function, so it is highly improbable that such configuration corresponds to a rounded solution of the SDP vector optimum[6]. Since the Local Improvement Algorithm provide a sequence of monotonic increasing values of the function, this worst configuration wont be reached from any other configuration. For simplicity and following the previous arguments, we skip the analysis of this configuration.

Suppose we are in an instance of (a). Let $v \in V_3$ the vertex to be changed of sign. Let n_2 be number of neighbours in V_2 and n_3 the number of neighbours in V_3. We know $n_2 + n_3 \leqslant 3$ and $n_3 \leqslant 2$. When the algorithm changes the sign of a vertex in V_3, the function value increases by 6, and from the induction hypothesis, we can improve the new configuration by at least, $\frac{2}{3}(|V_2| - n_2 + n_3) + \frac{20}{9}(|V_3| - n_3 - 1)$. Therefore the total gain from the initial configuration is at least,

$$\frac{2}{3}|V_2| + \frac{20}{9}|V_3| + \frac{34 - 14n_3 - 6n_2}{9} \geqslant \frac{2}{3}|V_2| + \frac{20}{9}|V_3|$$

Suppose we are in instance of (b) or (c). Observe that all the vertices in $u_1, u_2, \ldots u_k$ belong to V_2 (otherwise we would still be in case (a)), so the *exterior* edge adjacent to each u_i that is not in the path must be satisfied. If we change the sign of the vertices in odd positions in the case (b) (resp. the even position in case (c)), the value of the quadratic function increases by $2\lfloor\frac{k+1}{2}\rfloor$ (resp. $2\lfloor\frac{k}{2}\rfloor$ in (c)) since $2\lfloor\frac{k+1}{2}\rfloor$ edges passes from unsatisfied to satisfied, and $\lfloor\frac{k+1}{2}\rfloor$ edges passes from satisfied to unsatisfied (resp in (c)). Since *exterior* edges are satisfied there is no risk that *exterior* vertices to the cycle be "downgraded" from V_2 to V_1, instead, it may happen that some *exterior* vertices be upgraded from V_2 to V_3 or from V_1 to V_2 what is good for us. Then by the induction hypothesis the new configuration can be improved by at least $\frac{2}{3}(|V_2| - k)$. Therefore the total gain from the initial configuration of instance (b) and (c) is at least,

$$\frac{2}{3}(|V_2| - k) + 2\lfloor\frac{k}{2}\rfloor \geqslant \frac{2}{3}|V_2| = \frac{2}{3}|V_2| + \frac{20}{9}|V_3|$$

The above inequality is satisfied (i.e., the worst case) for a triangle with vertices in V_2.

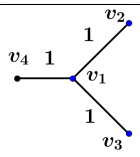
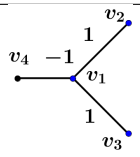
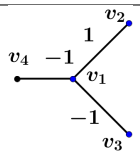
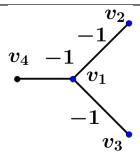
□

[6]I think that it should be possible show from the SDP constraints that the probability of having a worst case initialization is either 0 or much less than any other configuration

Let $p_l(i)$ be the probability that vertex i, obtained from the rounded SDP solution, belong to V_l, where $l = 2, 3$. Combining the results of the rounding and the improvement algorithm we get that for the rounded solution $\boldsymbol{x}$,

$$\begin{aligned} E(\boldsymbol{x}) &= \sum_{i,j \in E(G)} w_{ij} E(x_i x_j) + \frac{2}{3} \sum_{i \in V(G)} p_2(i) + \frac{20}{9} \sum_{i \in V(G)} p_3(i) \\ &= \sum_{i,j \in E(G)} w_{ij} \frac{\pi - 2\arccos(v_i \cdot v_j)}{\pi} + \frac{2}{3} \sum_{i \in V(G)} p_2(i) + \frac{20}{9} \sum_{i \in V(G)} p_3(i) \end{aligned} \tag{10}$$

To compute $p_2(i)$ and $p_3(i)$ we must check the sign of the edges adjacent to vertex i. For each of the 4 possible configurations of edge neighbourhood, the probabilities that the central vertex is in $p_2(i)$ and $p_3(i)$ is given in the next table. $p_H(a, b, c, d)$ denotes the probability that vectors a, b, c, d are in the same side of the random rounding hyperplane.

Configuration	$p_3(1)$	$p_2(1)$
	$p_H(-v_1, v_2, v_3, v_4)$	$p_H(-v_1, -v_2, v_3, v_4) + p_H(-v_1, v_2, -v_3, v_4) + p_H(-v_1, v_2, v_3, -v_4)$
	$p_H(-v_1, v_2, v_3, -v_4)$	$p_H(-v_1, -v_2, v_3, -v_4) + p_H(-v_1, v_2, -v_3, -v_4) + p_H(-v_1, v_2, v_3, v_4)$
	$p_H(v_1, -v_2, v_3, v_4)$	$p_H(v_1, v_2, v_3, v_4) + p_H(v_1, -v_2, -v_3, v_4) + p_H(v_1, -v_2, v_3, -v_4)$
	$p_H(v_1, v_2, v_3, v_4)$	$p_H(v_1, -v_2, v_3, v_4) + p_H(v_1, v_2, -v_3, v_4) + p_H(v_1, v_2, v_3, -v_4)$

Observe that the ratio between the $E(\boldsymbol{x})$ and $\mathrm{OPT}_{\mathrm{SDP}}$ satisfies:

$$\begin{aligned} \frac{E(\boldsymbol{x})}{\mathrm{OPT}_{\mathrm{SDP}}} &= \frac{\sum_{i,j \in E(G)} w_{ij} \frac{\pi - 2\arccos(v_i \cdot v_j)}{\pi} + \frac{2}{3} \sum_{i \in V(G)} p_2(i) + \frac{20}{9} \sum_{i \in V(G)} p_3(i)}{\sum_{i,j \in E(G)} w_{ij} v_i \cdot v_j} \\ &= \frac{\sum_{\in V(G)} \left(\sum_{j \in N(i)} \frac{1}{2} w_{ij} \frac{\pi - 2\arccos(v_i \cdot v_j)}{\pi} + \frac{2}{3} p_2(i) + \frac{20}{9} p_3(i) \right)}{\sum_{i \in V(G)} \left(\sum_{j \in N(i)} \frac{1}{2} w_{ij} v_i \cdot v_j \right)} \\ &\geqslant \min_{i \in V(G)} \frac{\sum_{j \in N(i)} \frac{1}{2} w_{ij} \frac{\pi - 2\arccos(v_i \cdot v_j)}{\pi} + \frac{2}{3} p_2(i) + \frac{20}{9} p_3(i)}{\sum_{j \in N(i)} \frac{1}{2} w_{ij} v_i \cdot v_j} \end{aligned} \tag{11}$$

The previous assumes that $\sum_{j \in N(i)} \frac{1}{2} w_{ij} v_i \cdot v_j > 0$ for all vertices i. I do not have a rigorous proof of this, however there are very strong facts that should make this condition hold: (1) It is true for the integer OPT, (2) here the vectors v's are the solution $\mathrm{OPT}_{\mathrm{SDP}}$, (3) the SDP constraints may not allow negative values for this term (4) numerical results.

For each of the 4 possible configurations of edge adjacency values, define[7],

$$C_k := \min_{v_1,v_2,v_3,v_4} \frac{\frac{1}{2}\left(a_{12}^k \frac{\pi - 2\arccos(v_1\cdot v_2)}{\pi} + a_{13}^k \frac{\pi - 2\arccos(v_1\cdot v_3)}{\pi} + a_{14}^k \frac{\pi - 2\arccos(v_1\cdot v_4)}{\pi}\right) + \frac{2}{3}p_2^k(i) + \frac{20}{9}p_3^k(i)}{\frac{1}{2}(a_{12}^k v_1\cdot v_2 + a_{13}^k v_1\cdot v_3 + a_{14}^k v_1\cdot v_4)} \tag{12}$$

Subject to the SDP constraints of Equation 9.

Let $C := \min_k C_k$. From Equation 11, and the previous definition we get that $\frac{E(\boldsymbol{x})}{\mathrm{OPT}_{\mathrm{SDP}}} \geqslant C$, i.e., **CSDP+LI MAX-2CORR** provide a C-approximation of MAX-2CORR.

The value of C can be calculated from a computer assisted technique. This is the kind of approach presented in [8][10] for MAX-3SAT, and MAX-2XOR in cubic graphs. It consists on doing a discrete search in the hole domain and bounding the derivative of the function.

In our case, since the function only depends of 4 vectors, we can limit the search to S^3. The hard part of the computations is the estimation of p_H, i.e., the probability of 4 vector to belong in the same side of a random hyperplane. As described by [10], this probability is just a function of the angles formed by each pair of vectors, but no closed formula is known.

I did a very basic MATLAB implementation of a discrete search on S^3 of the minimum of equation12 subject to the SDP constraints of equation 9, for each of the 4 possible configuration. My implementation genereta 4-tuples of vectors to evaluate using angle parametrization. This is a very intuitive and easy to implement approach, but does not guarantee uniform distribution of samples in $(S^3)^4$. Computation of p_H is done using random sampling from 10^5 vectors what provides a result with moderate variance.

The minimum of Equation 12 from the 20^6 4-tuples provided by my implementation was 0.8221. Although the exact value (which can be computed by a thorough computation) may be some below the one currently described, this seems to be a promising result. Disregarding the contribution of the local improvements (i.e, assuming $p_2(i) = p_3(i) = 0$) the approximation factor fall to 0.6634, what corroborates the importance of adding a local improvement term.

I attach the code used and the results.

References

[1] M. Charikar and A. Wirth. *Maximizing quadratic programs: extending Grothendieck's inequality.* FOCS 2004, pages 54-60.

[2] N. Alon and K. Makarychev, Y. Makarychev, and A. Naor. *Quadratic forms on graphs.* STOC 2005, pages 486-493.

[3] N. Alon and A. Naor. *Approximating the cut-norm via Grothendieck's inequality.* STOC 2004, pages 72-80.

[4] U. Feige and M. Langberg *The RPR^2 rounding technique for semidefinite programs.* ICALP 2001, pages 213-224.

[7] Here $k = 1, 2, 3, 4$ is just a superindex used to denote the parameters of each configuration.

[5] U. Feige, M. Karpinski and M. Langberg, *Improved Approximation of Max-cut on Graphs of Bounded Degree*, J. Algorithms 2002, volume 43, pages 201-219.

[6] M. Goemansand and D. Williamson. *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming.* JACM, 42:1115-45, 1995.

[7] N. Bansal, A Blum, and S. Chawla. *Correlation Clustering.* Machine Learning, 56:89-113, 2004.

[8] E. Halperin and D. Livnat, and U. Zwick. *MAX CUT in cubic graphs.* SODA 2002, pages 506-513.

[9] W.-K. Shih, S. Wu and Y. S. Kuo, *Unifying Maximum Cut and Minimum Cut of a Planar Graph.* IEEE Trans. Comput. 1990, pages 694-697.

[10] H. Karlo and U. Zwick, *A 7=8-approximation algorithm for MAX 3SAT?* Proceedings IEEE Symposium on Foundations of Computer Science 1997, pages 406415.

[11] F. Hadlock, *Finding a maximum cut of a planar graph in polynomial time.* SIAM J. Comput. 1975, vol 4, pages 221-225.