Pós-Graduação em Ciência da Computação

Diogo B. Henriques

An $O(|E|)$-linear Model for the MaxCut Problem

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

Recife

2019

Diogo B. Henriques

An $O(|E|)$-linear Model for the MaxCut Problem

This work was presented to the Graduate Program in Computer Sciences of Centro de Informática da Universidade Federal de Pernambuco as partial requirement for the Doctorate degree in Computer Sciences

Concentration Area: Combinatorial Optimization

Supervisor: Sóstenes Luiz Soares Lins

Recife

2019

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

H519o Henriques, Diogo Brandão Borborema
An O (|E|)-linear model for the max cut problem / Diogo Brandão Borborema Henriques. – 2019.
58 f.: il., fig., tab.

Orientador: Sóstenes Luiz Soares Lins.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2019.
Inclui referências e apêndices.

1. Ciência da computação. 2. Otimização combinatória. I. Lins, Sóstenes Luiz Soares (orientador). II. Título.

004 CDD (23. ed.) UFPE- MEI 2019-014

Diogo Brandão Borborema Henriques

An O(|E|)-linear Model for the MaxCut Problem

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

Aprovado em: 22/01/2019.

_____

Orientador: Prof. Dr. Sóstenes Luiz Soares Lins

BANCA EXAMINADORA

_____

Prof. Dr. Silvio de Barros Melo
Centro de Informática/ UFPE

_____

Prof. Dr. Ricardo Martins de Abreu Silva
Centro de Informática / UFPE

_____

Prof. Dr. Emerson Alexandre de Oliveira Lima
Escola Politécnica de Pernambuco/UPE

_____

Prof. Dr. Jones Oliveira de Albuquerque
Departamento de Estatística e Informática/ UFRPE

_____

Prof. Dr. Manoel José Machado Soares Lemos
Departamento de Matemática/ UFPE

_____

Prof. Dr. Sérgio de Carvalho Bezerra
Centro de Informática/UFPB

ABSTRACT

A polytope P is a *model* for a combinatorial problem on finite graphs G whose variables are indexed by the edge set E of G if the points of P with (0,1)-coordinates are precisely the characteristic vectors of the subset of edges inducing the feasible configurations for the problem. In the case of the (simple) Max Cut problem, which is the one that concern us here, the feasible subsets of edges are the ones inducing the bipartite subgraphs of G. This work we introduce a new polytope $\mathbb{P}_{12} \subset \mathbb{R}^{|E|}$ given by at most $11|E|$ inequalities, which is a model for the Max Cut problem on G. Moreover, the left side of each inequality is the sum of at most 4 edge variables with coefficients ± 1 and right side 0, 1 or 2. We restrict our analysis to the case of $G = K_z$, the complete graph in z vertices, where z is an even positive integer $z \geq 4$. This case is sufficient to study because the simple Max Cut problem for general graphs G can be reduced to the complete graph K_z by considering the objective function of the associated integer programming as the characteristic vector of the edges in $G \subseteq K_z$. This is a polynomial algorithmic transformation. An extension to the linear model into a more complete symmetric model which contains all the permutations for triangular and quadrilateral inequalities, equivalent to other formulations present in the literature is presented as well as the 01-cliques.

Keywords: Combinatorial Optimization. Integer Programming. Max Cut Problem.

RESUMO

Um politopo P é um *modelo* para um problema combinatorial em um grafo finito G cujas variáveis são indexadas pelo conjunto de arestas E de G se os pontos de P com coordenadas (0,1) são precisamente o vetor característico do subconjunto de arestas induzindo um configuração viável do problema. No caso do Corte Máximo simples, que é o problema abordado neste trabalho, o subconjunto de arestas viáveis é aquele que induz uma bipartição dos vértices de G. Neste trabalho é apresentado um novo politopo $\mathbb{P}_{12} \subset \mathbb{R}^{|E|}$ contendo no máximo $11|E|$ desigualdades, que é um modelo para o problema do Corte Máximo em G. O lado esquerdo de cada inequação é a soma de no máximo quatro variáveis de aresta com coeficientes ± 1 e o lado direito é 0, 1 ou 2. A análise é restrita para o caso $G = K_z$, o grafo completo com z vértices, onde z é um inteiro positivo com $z \geq 4$. Este caso é suficiente pois o problema do Corte Máximo simples para grafos gerais G pode ser reduzido ao grafo completo K_z.

Palavras-chave: Otimização Combinatória. Programação Inteira. Corte Máximo.

LIST OF FIGURES

Figure 3 – Small instances of Pog_h, $h = 2$, integer and $h = 2.5$, half integer: $Pog_2 \hookrightarrow \mathbb{RP}^2$ and $Pog_{2.5} \hookrightarrow \mathbb{RP}^2$. Cellular embeddings of the graphs into the *real projective plane* or a disk with antipodal identification, $\mathbb{RP}^2$. The two thick closed paths are instances of *zigzag paths*. There is a total of $z = 4h$ zigzag paths in Pog_h. Closely related to a zigzag path is a *closed straight line*, which is depicted as a thin line which goes parallel to an edge crossing it at the middle and following close the second half of the edge, turning at the angle to the next edge, where the process is repeated for all edges of the zigzag path which are crossed once by the closed straight line. The graph induced by the closed straight lines of a map is called the line embedding of the *phial map*. The *graph of this embedding* is the one whose vertices are the closed straight lines of the map and whose edges are the intersection points of two of such lines (which may coincide). The line embeddings are in 1-1 correspondence with the usual cellular embeddings which occur in another surface. This surface is determined, but not really relevant here for our current purposes. As a crucial property, we have that the graphs of the line embeddings induced by the Pog_h's are the complete graphs K_z, with z even. This is straightforward by the circular symmetry of these projective graphs: every pair of closed lines cross exactly once. To obtain Pog_h and its dual as labeled graphs consistent with the labels of $G_3 = K_z$ it is convenient to embed it into S^{23}. This can be done combinatorially by the *shaded rozigs*, see Figure 4. 25

Figure 4 – Example of labeled shaded rooted oriented zigzags or *labeled shaded rozigs* $G_1 = Pog_4$. We also display a strong O-join, denoted by T, given in thick edges: the parity of the number of edges of T in the coboundary (boundary) of a vertex (a face) and the parity of the degree of the vertex (the face) of $G_1 = Pog_4$ coincide. The labels of the vertices of K_{16} are the digits in $1, 2 \ldots, 9, A, B, C, D, E, F, G$ (base 17). An edge of $G_3 = K_z$ is labeled by an unordered pair of vertices. Note that the face boundaries in clockwise order and the vertices coboundaries in counter-clockwise order correspond to directed polygons in $G_3 = K_z$. Rozig 1 is displayed. 26

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

BFS	Breadth-first Search
CNF	Conjunctive Normal Form
DNF	Disjunctive Normal Form
DFS	Depth-first Search
ILP	Integer Linear Programming
LP	Linear Programming
SAT	Boolean Satisfiability Problem

CONTENTS

1 INTRODUCTION

The *Max Cut problem* is one of the first NP-complete problems according to Garey e Johnson (1979). This problem can be stated as follows. Given a graph G does it has a bipartite subgraph with n edges? It is a very special problem which has been acting as a paradigm for great theoretical developments. See, for instance Goemans e Williamson (1995), where an algorithm with a rather peculiar worse case performance (greater than 87%) can be established as a fraction of type (solution found/optimum solution). This result constitutes a landmark in the theory of approximation algorithms.

Under computational complexity theory, a *decision problem* can be simply put as a problem with **yes** or **no** answer. Complexity classes can be defined as follows:

Polynomial or P: the set of all decision problems that can be solved in polynomial time by a deterministic Turing machine such as linear programming (i.e.: the time it takes is a polynomial function on the size of the problem).

Non-deterministic Polynomial or NP: the set of all decision problems that can be solved in polynomial time by a non-deterministic Turing machine such as integer factorization. In other words, all problems whose instances where the answer is "yes" have proofs that can be verified in polynomial time.

NP-Complete: the set of all decision problems in NP for which exists a polynomial transformation from any other NP problem into it. A significant consequence of this is that if a polynomial solution exists for any given problem in NP-Complete then all problems within this class also posses polynomial solution. 3-SAT is such an example.

NP-Hard: the set of all problems which are at least as hard as the NP-complete problems but are not necessarily decision problems. If a problem is NP-Hard then an NP-Complete problem exists such that it is reducible to the NP-Hard problem in polynomial time. Consider the halting problem, given a program and an input, will it halt? This decision problem is clearly not in NP but any NP-Complete problem can be trivially reduced to it.

This work investigates the Max Cut problem along with cut polyhedra associated with it. In a formulation found in Barahona, Jünger e Reinelt (1989), the authors express odd subsets of circuits as exponential restrictions in a 0-1 linear program. A short formulation for the Max Cut in Lancia e Serafini (2011) expresses the problem as $O(|V| \cdot |E|)$ inequalities and $O(|V|^2 + |E|)$ variables. Yet another formulation by Nguyen, Minoux e Nguyen (2016) proposed a compact model only containing short inequalities in the order of $O(|V| \cdot |E|)$.

Our contribution is two fold. First we present a formulation that contains exactly E variables and $11|E|$ short inequalities (each involving 3 edge variables with coefficients ± 1) so that the polytope in $\mathbb{R}^{|E|}$ formed by these inequalities has its all integer coordinate points in 1-1 correspondence with the characteristic vectors of the complete bipartite subgraphs of K_z. Second we provide a self-contained proof that signed odd polygons are valid inequalities for the Max Cut problem.

In the first part we discuss the theoretical background involving the Max Cut problem and related works which propose new and efficient formulations. In this section we also discuss several works which provide means to find new classes of valid inequalities, some of which are employed in our computational efforts whilst investigating the problem.

The second part deals exclusively with the theoretical models proposed by this work. We first define the necessary combinatorial elements upon which our formulation is based and thus provide the formulation itself. An important theorem contained in this part demonstrates how it is feasible to decide if an integer maximum cut exists given a relaxed dual optimum with that same integer objective value via a 2-SAT. This reduces the original problem significantly given the necessary conditions stated in the theorem.

All computational efforts and investigations regarding this work, including experiments with both random and well-known graphs such as Coxeter and Johnson's graph family. We also demonstrate the effectiveness of working with relaxed larger formulations containing all cycles of the graph and more compact formulations which append violating inequalities as they appear.

2 MAX CUT PROBLEM

2.1 FORMULATIONS FOR THE CUT POLYTOPE

The Max Cut problem is equivalent to finding the maximum bipartite graph given that no edge has a negative weight associated with it (Barahona, Grötschel e Mahjoub (1985)). It has also been shown that the Max Cut problem is NP-Hard in the general case as stated, but can be solved polynomially for planar graphs (Hadlock (1975)), graphs with no long odd cycles (Grötschel e Nemhauser (1984)) and graphs not contractible to K_5 (Barahona (1983)). An important survey on the Max Cut problem is available at Poljak e Tuza (1995) where several works related to cut polytopes are discussed in detail.

Several formulations for cut polyhedra have been proposed in the revised literature. These polyhedra are frequently associated to classical combinatorial optimization problems such as the separation or maximum cut problem, object of focus in this work. Consider a graph $G = (V, E)$ finite, undirected and with no multiple edges, where V is the *node set* and E is the *edge set*. Given $C \in V$, $\delta(C)$ is the set of edges with one extremity incident to C and the other to $C \setminus V$. This is called a *cut* and the convex hull of incidence vectors of cuts is known as a *cut polytope*. The problem thus resides in finding a maximum cut for a given graph.

Such a formulation proposed by Barahona e Mahjoub (1986) includes all odd *chordless cycles* in a undirected graph G. A chordless cycle is any cycle within G such that the induced subgraph is the cycle itself. The authors demonstrate that any triangular face within the complete graph K_z can be written as a valid *cycle inequality* and the set of all of such inequalities build a cut polytope for a 0-1 linear program.

Let C be a chordless cycle and F a facet in G. Let κ be an incident vector of a cut. Then for each cycle C, $\kappa(C)$ is even such that

$$\kappa(F) - \kappa(C \backslash F) \leq |F| - 1, \quad F \in C, |F| \quad odd.$$

so for every distinct vertices i, j, k in K_z,

$$\begin{aligned} x_{ij} + x_{jk} + x_{ki} &\leq 2 \\ -x_{ij} + x_{jk} + x_{ki} &\leq 0 \\ x_{ij} - x_{jk} + x_{ki} &\leq 0 \\ x_{ij} + x_{jk} - x_{ki} &\leq 0 \end{aligned}$$

where x_{ij} is the edge connecting nodes i to j are the cycle inequalities for the polygon i, j, k. Such valid inequalities of triangles are called *triangle inequalities*. A polytope containing

triangle inequalities covering all possible cycles (and their sign permutations) is known as *triangular polytope*.

In Lancia e Serafini (2011) the authors present a shorter formulation for the Max Cut by defining new valid inequalities on the already known formulation of the problem ((BARAHONA; JÜNGER; REINELT, 1989), (SIMONE; RINALDI, 1994)) and demonstrating these valid inequalities can be used to solve the problem through a simple preparation procedure. The authors define a new graph which contains a duplicated vertex set and new arc cycles whose length depends on the current optimal solution. The formulation presented posses $O(|V| \cdot |E|)$ inequalities and $O(|V|^2 + |E|)$ variables which the authors claim have potential to be efficient for sparse graphs.

An even more compact formulation proposed by Nguyen, Minoux e Nguyen (2016) contains $O(|E| \cdot |V|)$ inequalities. The authors call attention to the particular case of sparse graphs where, on the best case scenario, $|E| = |V|$ and the problem is reduced to $O(|V|^2)$ constraints. The basis for their work is the *metric polytope*, upon which they apply an elimination procedure to remove redundant inequalities.

2.2 VALID INEQUALITIES FOR THE MAX CUT MODEL

Some authors in the revised literature propose new classes of valid inequalities since none of the proposed formulations are the integer hull of the desired polytope. As part of this work, we have adapted these approaches in order to find valid inequalities on demand. Tests with random graphs indicate that including constraints induced by all possible circuits within the complete graph significantly reduces the number of problems whose relaxed optima is an integer vertex on the polytope. Unfortunately this ratio of integer to fraction relaxed optima quickly deteriorates as the problem size increases.

In Barahona, Jünger e Reinelt (1989) the authors present a cutting plane algorithm embedded in a branch and bound framework (also known as branch and cut) to address the problem of finding a maximum cut. The strategy adopted to generate cutting planes uses odd cycle inequalities. Since this strategy is widely used by others, including this work, it is briefly described here. Let the graph H be defined as $(V' \cup V'', E' \cup E'' \cup E''')$ which consists of two copies of a graph G, $G' = (V', E')$ and $G'' = (V'', E'')$ with some additional edges E'''. For every edge e_{ij} in E four edges are created in H, $e_{i'j'} \in E'$ and $e_{i''j''} \in E''$ with weight y^*_{ij} and two edges $e_{i'j''}$ and $e_{i''j'}$ with weight $1 - y^*_{ij}$ both in E''' for $i', j' \in V'$ and $i'', j'' \in V''$ and where x^*_{ij} is the value for the optimization variable corresponding to the edge x_{ij} at the current optimum solution. It is shown that any path from i' to i'' has an odd number of edges in E''' and if the shortest path from i' to i'' is smaller than one then there is a corresponding odd cycle which is violated at x^*.

A polyhedral cut and price approach is investigated by Krishnan e Mitchell (2006)

where the authors formulate the semidefinite programming (SDP) relaxation of Max Cut as a semi-infinite linear programming problem, which is solved within an interior point cutting plane algorithm at the pricing phase. In a second phase cutting planes are added to the original problem in order to improve the SDP relaxation.

Another valid class of constraints known as *gap inequalities* which appear in Laurent e Poljak (1996) and further studied by Galli, Kaparis e Letchford (2012) are constructed from a finite sequence of integers related to a positive semidefinite relaxation of the max-cut problem. This class of inequalities is quite general, including hypermetric (see Tylkin (1960) and Kelly (1975)) and negative type (Schoenberg (1938)) inequalities. Another important type of valid inequality which can be seen as a particular case of gap inequalities are those originated from cliques in the graph. The authors express how to obtain the integer sequences of the corresponding gap inequalities that define facets of the cut polytope. This technique is discussed in further detail at section 3.4 at the appropriate context of this work.

3 THEORETICAL MODELS

3.1 THICK GRAPHS, GEMS AND Q-DUALITIES

Thick graphs into closed surfaces. A surface is *closed* if it is compact and has no boundary. A closed surface is characterized by its Euler characteristic and the information whether or not is orientable. We use the following combinatorial counterpart for a graph G cellularly embedded into a closed surface S, here called a *map*.

Cellularly embedded means that $S \backslash G$ is a finite set of open disks each one named a *face* of the embedding, whence a *surface dual graph* is well defined. Each edge is replaced in the surface by an ϵ-thick version of it, named ϵ-rectangle. Each vertex v is replaced by a δ-disk, where δ is the radius of the disk whose center is v. The ϵ-rectangles and the δ-disks form the *thick graph of* G, denoted by $T(G)$. By choosing an adequate pair ($\epsilon < \delta$), the boundary of $T(G)$ is a *cubic graph* (i.e., regular graph of degree 3), denoted by $C(G)$. The edges of $C(G)$ can be properly colored with 3 colors: we have *short*, *long*, and *angular* colored edges so that at each vertex of $C(G)$ the three colors appear. The long (resp. short) colored edges are the edges which induced by the long (resp. short) sides of the ϵ-rectangles. The angular edges are the other edges (Figure 1).

Gems or hollow thick graphs. A cubic 3-edge colored graph H in colors $(0, 1, 2)$ is called a *gem* (for *g*raph-*e*ncoded *m*ap) if the connected components induced by edges of colors 0 and 2 are *polygons* with 4 edges. A *polygon* in a graph is a non-empty subgraph which is connected and has each vertex of degree 2. A *bigon* in H is a connected component of the subgraph induced by all the edges of any two chosen chosen among the three colors. An ij-gon is a bigon in colors i and j. From H we can easily produce the surface S and $G \hookrightarrow S$: attach disks to the bigons of H thus obtaining $T(G) \hookrightarrow S$ up to isotopy. To get G embedded into S just contract the δ-disks to points. Each rectangle becomes a digon and contracting these digons to their medial lines we get $G \hookrightarrow S$. The Euler characteristic of S is $v(H) + f(H) - r(H)$, where $v(H)$ is the number of 01-gons of G (or the number of vertices of G), $f(H)$ is the number of 12-gons of H (or the number of faces of $G \hookrightarrow S$) and $r(H)$ is the number of rectangles of H (or the number of edges of H). Moreover, S is an orientable surface iff and only H is a bipartite graph, see (LINS, 1980). Note that in each gem any edge appear exactly in two bigons: indeed, if the edge is of color i it will appear once in a ij-gon and once in a ik-gon, where $\{i, j, k\} = \{0, 1, 2\}$. The surface of a map is obtainable from the gem by attaching disks to the bigons and identifying the boundaries along the two occurrences of each edges.

Q-graphs and their dualities. A *perfect matching* in a graph with an even number, v,

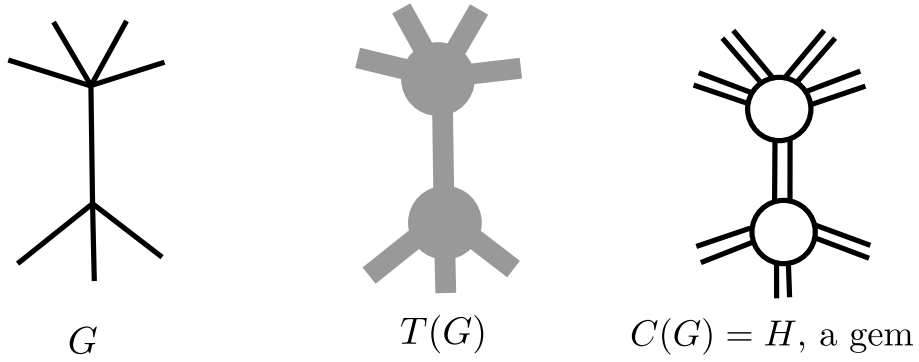


Figure 1 – Neighborhood of an edge in $G \hookrightarrow S$ its thickened version and a hollow counterpart: the gem H. To get a Q-graph from H, let μ_0 be the short edges, μ_1 be the angular edges (they correspond to angles in G), let μ_2 be the long edges, and finally add the crossing edges μ_3 as the diagonals of the 02-rectangles of H. Note that the colors of the edges of Q are implicit 0 is the color of the short edges of the 02-rectangles, 2 of their long edges, 1 is the color of the edges of H not in the rectangles and 3 is the color of the crossing edges.

of vertices is a set of $v/2$ pairwise disjoint edges. A *Q-graph* $Q(\mu_0, \mu_1, \mu_2, \mu_3)$ is the disjoint union of 4 ordered of its perfect perfect matchings $\mu_i,\ i = 0, 1, 2, 3$, so that each component of $\mu_0 \cup \mu_2 \cup \mu_3$ is a complete graph K_4. Each such K_4 is called a *hyperedge* of the Q-graph. The edges in μ_1 are called *angular edges* of the q-graph. The edges in μ_0 are called *short edges*, the ones in μ_2, *long edges*, the ones in μ_3 are called the *crossing edges*. The graphs $Q(\mu_0, \mu_1, \mu_2, \mu_3)$ and $Q(\mu_2, \mu_1, \mu_0, \mu_3)$ are *dual Q-graphs*. The graphs $Q(\mu_0, \mu_1, \mu_2, \mu_3)$ and $Q(\mu_3, \mu_1, \mu_2, \mu_0)$ are *phial Q-graphs*. The graphs $Q(\mu_0, \mu_1, \mu_2, \mu_3)$ and $Q(\mu_0, \mu_1, \mu_3, \mu_2)$ are *skew Q-graphs*. To obtain a gem H, whence G from a Q-graph, just remove its last perfect matching. Note that dual Q-graphs induce the same surface S and the same zigzag paths while interchanging boundary of faces and coboundaries of vertices. Skew Q-graphs induce the same graph G and interchange coboundary of faces and zigzag paths. Phial Q-graphs interchange coboundary of vertices and zigzag paths while maintaining the boundaries of the faces (as cyclic set of edges) in the respective surfaces, see Figure 2. Note that the embedding $G \hookrightarrow S$ defines the Q-graph. This enable us to identify

$$
\begin{aligned}
Q_1 = Q(\mu_0, \mu_1, \mu_2, \mu_3) \equiv G_1 \hookrightarrow S^{12} \equiv G_1,\\
Q_2 = Q(\mu_2, \mu_1, \mu_0, \mu_3) \equiv G_2 \hookrightarrow S^{12} \equiv G_2,\\
Q_3 = Q(\mu_3, \mu_1, \mu_0, \mu_2) \equiv G_2^{\sim} \hookrightarrow S^{23} \equiv G_2^{\sim},\\
Q_4 = Q(\mu_3, \mu_1, \mu_2, \mu_0) \equiv G_3 \hookrightarrow S^{23} \equiv G_3,\\
Q_5 = Q(\mu_2, \mu_1, \mu_3, \mu_0) \equiv G_3^{\sim} \hookrightarrow S^{31} \equiv G_3^{\sim},\\
Q_6 = Q(\mu_0, \mu_1, \mu_3, \mu_2) \equiv G_1^{\sim} \hookrightarrow S^{31} \equiv G_1^{\sim}.
\end{aligned}
$$

G_1 the graph of the dual map, G_2 and graph of the phial map G_3. To get the *phial* of a map, we interchange the short edges of the rectangles by their diagonals. There are

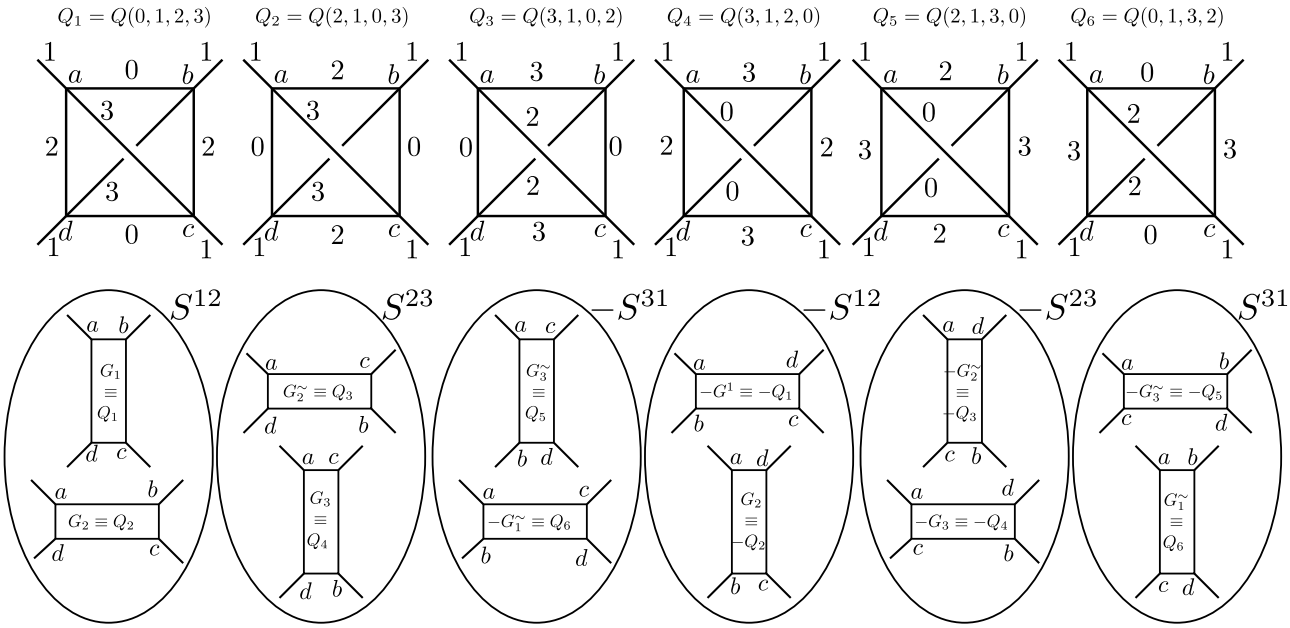


Figure 2 – Q-graph $Q(h,i,j,k)$ is a short form of $Q(\mu_h, \mu_i, \mu_j, \mu_k)$. We depict the Q-dualities of a Q-graph (usual surface duality, skew duality and phial duality) which induce 3 graphs G_1, G_2, G_3 and 3 surfaces: S^{12}, S^{23}, S^{31}. The minus signs mean a local reversal of orientation given by the cyclic order of the rectangle corners (a,b,c,d). Graphs G_i and $G_i^{\sim}$ $(i = 1,2,3)$ are the same: they are just embedded into distinct surfaces in such a way that the faces of one are the zigzags paths (left-right paths) of the other. The zigzags paths are closed and well defined — they correspond to the 13-gons of Q — even if the surface is non-orientable, where is impossible to define left or right globally. Taking the dual $(\mathcal{DU})$ corresponds in the gem to switch the vertical rectangles to horizontal ones (and vice-versa) while maintaining the cyclic order of the corners of the rectangle (so the surface does not change). Taking the skew $(\mathcal{SK})$ corresponds to exchange corners linked by one of the two short sides of the rectangles. Starting with $Q(0,1,2,3)$ and applying iteratively the composition $\mathcal{SK} \circ \mathcal{DU}$ we get the six Q-graphs which appear in the top of each one of the six surfaces. Taking the phial $(\mathcal{PH})$ is defined as $\mathcal{PH} = \mathcal{DU} \circ \mathcal{SK} \circ \mathcal{DU}$, or directly by exchanging a pair of corners linked by one of the 2 long sides of the rectangle. If we care for orientation and all the 3 surfaces are orientable, then there are in fact 12 Q-graphs and 6 oriented surfaces. Note that $\mathcal{PH} = \mathcal{SK} \circ \mathcal{DU} \circ \mathcal{SK} \circ \mathcal{DU} \circ \mathcal{SK} \circ \mathcal{DU} \circ \mathcal{SK} \circ \mathcal{DU} \circ \mathcal{SK}$. But orientation does not concern us here. Therefore there are only 3 pairs of skew maps, each pair inducing the same graph $\{G_1 :\hookrightarrow S^{12}, G_1^{\sim} :\hookrightarrow S^{31}\}$, $\{G_2 :\hookrightarrow S^{12}, G_2^{\sim} :\hookrightarrow S^{23}\}$, $\{G_3 :\hookrightarrow S^{23}, G_3^{\sim} :\hookrightarrow S^{31}\}$, and 3 surfaces S^{12}, S^{23}, S^{31}.

also the twisted maps $G_1^{\sim}$, $G_2^{\sim}$ and $G_3^{\sim}$. There are three closed surfaces S^{12} where G_1 and G_2 embed as duals, S^{23} where $G_2^{\sim}$ and G_3 embed as duals and S^{31} where $G_3^{\sim}$ and $G_1^{\sim}$ embed as duals. For the case that concerns us, G_3 is K_z with line embedding in S^{12}, G_1 is Pog_h and G_2 is the $\mathbb{RP}^2$-dual of Pog_h, since S^{12} is $\mathbb{RP}^2$. These dualities were introduced first in (LINS, 1980) and then in (LINS, 1982).

3.2 THEORY OF GEMS AND Q-DUALITIES FOR CELLULAR EMBEDDINGS OF GRAPHS INTO SURFACES

Let G_1 be an arbitrary map of a graph into a surface, orientable or not, and G_2, G_3, denote respectively the dual and phial of G_1. Let E denote the common set of edges for graphs G_1, G_2, G_3: they are identified via the hyperedges of the associated Q-graph.

3.2.1 Reformulation of the Max Cut Problem

Vector spaces from graphs. For subset of edges A and B let $A + B$ denote their symmetric difference. This is closely related with the sum in $GF(2)$ via the characteristic vectors. Thus an element is in

$$A_1 + A_2 + \ldots + A_p$$

if it belongs to an odd number of A_i's. This sum on subsets of edges becomes an associative binary operation and 2^E, the set of all subsets of E, becomes a vector space via $+$ on subsets, or, what amounts to be the same, the mod 2 sum of characteristic vectors of the subsets of edges. There is a distinguished basis given by the characteristic vectors of the singletons. We say that subset of edges A is orthogonal to subset of edges B if $|A \cap B|$ is even. If $W \subseteq 2^E$ is a subspace, then $W^\perp = \{u \in 2^E : u \perp w, \forall w \in W\}$ is also a subspace and $dim W + dim\ W^\perp = |E|$. Let $\mathcal{V}_i$ $(i = 1, 2, 3)$ be the subspace of 2^E generated by the coboundary of the vertices of G_i, or *coboundary space* of G_i. The *cycle space* of G_i is $\mathcal{V}_i^\perp$. The *face space* of G_i, denoted by $\mathcal{F}_i$, is the subspace of $\mathcal{V}_i^\perp$ generated by the face boundaries of G_i. The *zigzag space* of G_i, denoted by $\mathcal{Z}_i$, is the subspace of $\mathcal{V}_i^\perp$ generated by the zigzag paths of G_i. Note that G_i is rich iff $\mathcal{V}_i^\perp = \mathcal{F}_i + \mathcal{Z}_i$. In particular, $\mathcal{F}_1 = \mathcal{V}_2$ and $\mathcal{Z}_1 = \mathcal{V}_3$.

Theorem 1 (Absorption property)**.** *Let (i, j, k) denote a permutation of $\{1, 2, 3\}$. Then $\mathcal{V}_i \cap \mathcal{V}_j \subseteq \mathcal{V}_k$.*

Proof. For a proof we refer to Theorem 2.5 of (LINS, 1980). The proof is long and we do not know a short one. This is a basic property which opens the way for a perfect abstract symmetry among vertices, faces and zigzags. A useful consequence of this property is that $\mathcal{V}_1 \cap \mathcal{V}_2 = \mathcal{V}_1 \cap \mathcal{V}_3 = \mathcal{V}_2 \cup \mathcal{V}_3 = \mathcal{V}_1 \cap \mathcal{V}_2 \cup \mathcal{V}_3$. □

The *cycle deficiency* of G_i is $cdef(G_i) = dim((\mathcal{V}_i^\perp)/(\mathcal{V}_j + \mathcal{V}_k))$. Map G_i is *rich* if its cycle deficiency is 0, implying, in particular, $\mathcal{V}_i = \mathcal{V}_j^\perp \cap \mathcal{V}_k^\perp$, for all permutations (i, j, k) of $(1, 2, 3)$.

Corollary 2. *Maps G_1, G_2, G_3 have the same cycle deficiency.*

Proof. Assume G_1 has e edges, v vertices, f faces and z zigzags. Then

$$cdef(G_1) = (e - v + 1) - ((f - 1) + (g - 1) - \gamma) = e - (v + f + g) + (3 + \gamma),$$

where $\gamma = dim(\mathcal{V}_1 \cap \mathcal{V}_2 \cup \mathcal{V}_3)$. The Corollary follows because $v + f + z$ is invariant under permutations of (v, f, z). □

Thus richness is a symmetric property on the maps G_1, G_2, G_3: we have G_1 is rich $\Leftrightarrow G_2$ is rich $\Leftrightarrow G_3$ is rich. A subgraph is *even* if each of its vertices has even degree.

Corollary 3. *If $F \subseteq E$ induces an even subgraph of G_i, then $F \in \mathcal{V}_i^{\perp}$.*

Proof. Any polygon of G_i is in $\mathcal{V}_i^{\perp}$. Note that F is a sum of polygons and so, $F \in \mathcal{V}_i^{\perp}$. □

A subset $F \subseteq E$ is a *strong O-join in G_1* if it induces a subgraph so that at each vertex v and each face f the parity of the number of F-edges in the coboundary of v and in the boundary of f coincides with the parity of the degrees of v and f, respectively. Note that F is a strong O-join iff $\overline{F} = E \backslash F \in \mathcal{V}_1^{\perp} \cap \mathcal{V}_2^{\perp}$. See Figure 4, where we depict a strong O-join T given by the thick edges in $G_1 = Pog_4$. In the case of a rich G_3, F is a strong O-join of G_1 iff $\overline{F} \in \mathcal{V}_3$.

The *coboundary of a set of vertices W* is the set of edges which has one end W and the other in $V \backslash W$. A subset of edges is a *coboundary in a graph* iff it induces a bipartite subgraph: the edges of this graph constitutes the coboundary of the set of vertices in the same class of the bipartition. A *cut* in combinatorics is frequently defined as a minimal coboundary. Thus it is preferable to talk about maximum coboundary instead of talking about maximum cut to avoid misunderstanding.

Theorem 4 (Reformulation of Max Cut Problem)**.** *Let G_3 be a rich map. The maximum cardinality of a coboundary in G_3 is equal to the cardinality of $|E|$ minus the minimum cardinality of a strong O-join in G_1.*

Proof. The result follows because the complement of a strong O-join F is an even subgraph in graphs G_1 and G_2. Thus, $\overline{F} \in \mathcal{V}_1^{\perp} \cap \mathcal{V}_2^{\perp} = \mathcal{V}_3$. The last equality follows because G_3 is rich. Note that the elements of $\mathcal{V}_3$ are precisely the coboundaries of G_3. □

3.2.2 Projective Orbital Graphs

Motivation to restrict to $G_3 = K_z, z$ even. In order to use the Q-dualities and rich maps we must start with a rich map G_3. Our universal choice for G_3 is the complete graph K_z with z even. There are various reasons for this choice. (a) Every graph is a subgraph of some K_z. (b) It is very easy to embed $G_3 = K_z$ in some surface so that its phial G_1 and dual of the phial G_2 are embedded into the real projective plane, $\mathbb{PR}^2$:

the simplest closed surface after the sphere. (c) There is a combinatorial well structured generator subset of the cycle space of K_z, $\mathcal{V}_3$, given by all but one coboundaries of the vertices of G_1 and the all but one coboundaries of the vertices of G_2 (faces of G_1). Moreover each one of these generators correspond to a polygon in K_z having either 3 or 4 edges. Finally, (d) the maximum cardinality of a bipartite subgraph of an arbitrary graph G with z vertices can be obtained by solving the integer 0-1 programming problem using the characteristic vector of the edge set of G relative to the complete graph $G_3 = K_z$ as objective function. If z is odd attach a pendant edge e to G, and solve the problem for $G + e \subset K_{z+1}$. All these properties justify the restriction to complete graphs with an even number of vertices.

Our formulation will use inequalities induced signed forms of these generating polygons in all possible ways. So it is paramount to have short polygons as generators, otherwise an exponential number of inequalities arises from the beginning. Our approach starts by constructing graph $G_1 = Pog_h$ embedded into $\mathbb{RP}_2$, and its description follows.

The projective orbital graphs. Let $h \in \{ 1, \frac{3}{2}, 2, \frac{5}{2}, 3, \frac{7}{2}, 4, \frac{9}{2}, \ldots\}$. The *Projective orbital graph or* $Pog(h)$ is defined as follows.

Case h integer. If h is an integer, then $Pog(h)$ consists of h concentric circles (orbits) having each $z = 4h$ vertices equally spaced. In the complex plane the hz vertices of $Pog(h)$ are $\{k \exp(2\pi ij/z) : k = 1, 2, \ldots, h,\ \ j = 0, \ldots, z-1\}$. Each one of the h orbits of $Pog(h)$ induces z edges as closed line segments in the complex plane:

$$\{[k \exp(2\pi ij/z, k \exp(2\pi i(j+1)/z)]) : j = 1, \ldots, z\}.$$

These edges are called *orbital edges*. There are also zh *radial segments* being $z(h-1)$ *radial edges* and z *pre-edges*: $\{[k \exp(2\pi ij/z), (k+1) \exp(2\pi ij/z]) : k = 1, \ldots, h,\ \ j = 1, \ldots, z\}$. Note that the z points $\{[(h+1) \exp(2\pi ij/z]) : j = 1, \ldots, z\}$ are not vertices of $Pog(h)$ and are called *auxiliary points*. Each one of the radial segments incident to an auxiliary point is a *pre-edge*. The graph whose vertices are the vertices of $Pog(h)$ plus the auxiliary points and whose edges are the edges plus pre-edges of $Pog(h)$ is named a *pre-*$Pog(h)$. Take a pre-$Pog(h)$ and embed it in the planar disk with center at the origin and radius $h+1$, denoted D, of the usual plane so that the auxiliary points are in the boundary of D. The antipodal points of ∂D are identified, forming *real projective plane* $\mathbb{RP}^2$. In particular pairs of antipodal auxiliary points become a single bivalent vertex which is removed and the result is the graph $Pog(h)$ embedded into $\mathbb{RP}^2$. (see left side of FigureFigure 3) This completes the definition of $Pog(h)$, in the case of integer h.

Case h is half integer. If h is a half integer then $Pog(h)$ has $\lfloor h \rfloor$ orbits each with $z = 4h$ vertices and a degenerated orbit corresponding to the extra $\frac{1}{2}$ and inducing a single central vertex. In the complex plane the $hz + 1$ vertices of $Pog(h)$ are $\{k \exp(2\pi ij/z) : k = 1, 2, \ldots, \lfloor h \rfloor,\ \ j = 0, \ldots, z-1\} \cup \{0\}$. The orbital and radial edges as well as the

identifications are defined similarly as in the case h integer. The extra ingredient is that there are z edges linking 0 to the vertices in the innermost non-degenerated orbit (see right side of FigureFigure 3).

The shapes of the Pog_h's are tailored in such a way that it has z *zigzag paths*: such a path is exemplified in thick edges in FigureFigure 3). These paths alternates choosing the rightmost and leftmost edges at each vertex. Since $\mathbb{RP}^2$ is non-orientable, in traversing an edge crossing the boundary of D we must repeat the direction (left-left or right-right, instead of changing it). Note that a zigzag path is closed since it links two antipodal auxiliary points in D before they are identified in $\mathbb{RP}^2$.

3.2.3 Combinatorially Constructed Labeled $\mathbf{Pog_h}$

By using a combinatorial construction for Pog_h we get the triad of graphs $G_1 = Pog_h$ its dual in $\mathbb{RP}^2$, G_2 and its phial $G_3 = K_z$. The construction is based on a table named *shaded rozigs* which amounts to an embedding of K_z into some higher genus surface. We refer to Figure 4.

The rozig table has z rows and $z-1$ columns. Each entry of the table is an ordered distinct pair of labels in $\{1, 2, \ldots, z$ and each such pair appears twice (maybe with the symbols switched). These symbols label the vertices of the complete graph and the pair is an oriented form an edge of $G_3 = K_z$. The filling of the table depends on a simple function $suc2\{1, 2, \ldots z\} \times \{z\} \longrightarrow \{1, 2, \ldots z\} \times \{z\}$, where $suc2(\ell, z) = \ell + 2$, if $\ell \leq z - 2$, $suc2(z, z) = 1$, $suc2(z - 1, z) = 2$.

The rozig table has 3 types of columns: the *projective column*, formed by the 0-column, the *left columns*, formed by columns 1 to $z/2$ and the *right columns* formed by columns $z/2 + 1$ to $z - 1$.

Defining the first row of the rozig table. The entries in the first row start with (2,1) in the projective column, followed by

$$(1, 4), (6, 1), \ldots, (z, 1) \text{ or by } (1, 4), (6, 1), \ldots, (1, z),$$

according to $z \equiv 2 \mod 4$ or $z \equiv 0 \mod 4$ filling the left columns. Finally we have, if $z \equiv 2 \mod 4$,

$$(1, 3), (5, 1), \ldots, (z - 1, 1) \text{ or by } (3, 1), (1, 5), \ldots, (1, z - 1),$$

filling the right columns. This completes the filling of the first row of the rozig table. This row corresponds to the cyclic order of the oriented edges of the coboundary of vertex 1 of $G_3 = K_z$. It corresponds also to a rooted oriented zigzag (rozig) path labeled 1 in $G_1 = Pog_h, z = 4h$. See Figure 4.

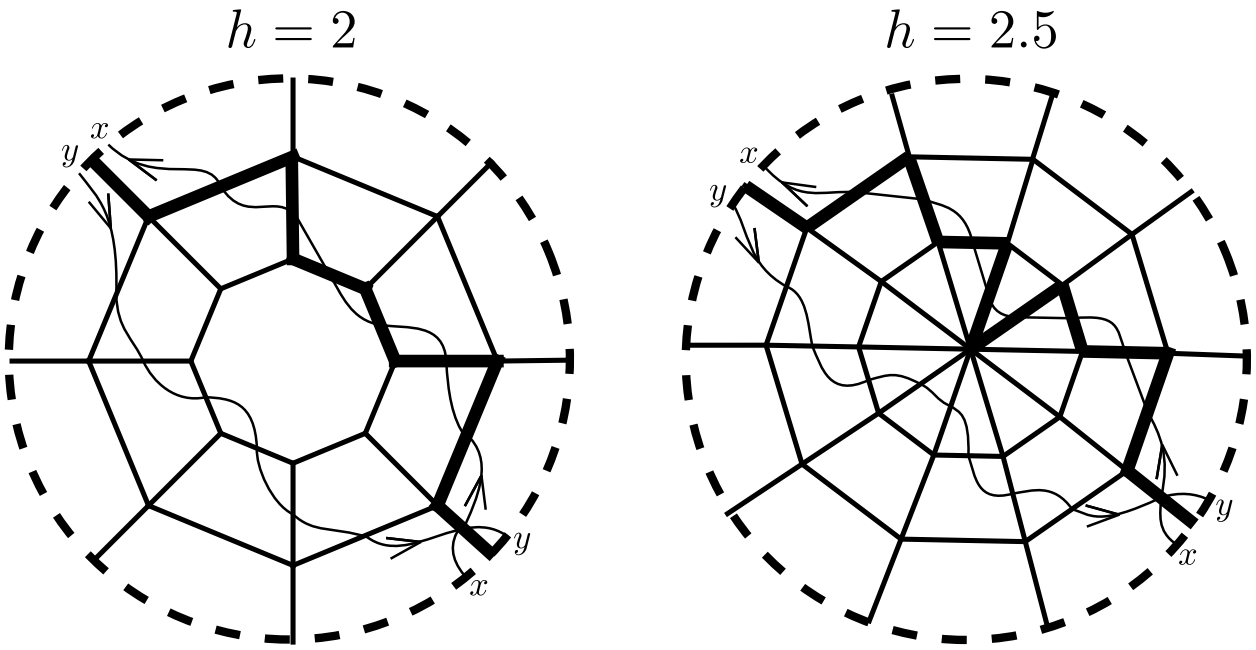


Figure 3 – Small instances of Pog_h, $h = 2$, integer and $h = 2.5$, half integer: $Pog_2 \hookrightarrow \mathbb{RP}^2$ and $Pog_{2.5} \hookrightarrow \mathbb{RP}^2$. Cellular embeddings of the graphs into the *real projective plane* or a disk with antipodal identification, $\mathbb{RP}^2$. The two thick closed paths are instances of *zigzag paths*. There is a total of $z = 4h$ zigzag paths in Pog_h. Closely related to a zigzag path is a *closed straight line*, which is depicted as a thin line which goes parallel to an edge crossing it at the middle and following close the second half of the edge, turning at the angle to the next edge, where the process is repeated for all edges of the zigzag path which are crossed once by the closed straight line. The graph induced by the closed straight lines of a map is called the line embedding of the *phial map*. The *graph of this embedding* is the one whose vertices are the closed straight lines of the map and whose edges are the intersection points of two of such lines (which may coincide). The line embeddings are in 1-1 correspondence with the usual cellular embeddings which occur in another surface. This surface is determined, but not really relevant here for our current purposes. As a crucial property, we have that the graphs of the line embeddings induced by the Pog_h's are the complete graphs K_z, with z even. This is straightforward by the circular symmetry of these projective graphs: every pair of closed lines cross exactly once. To obtain Pog_h and its dual as labeled graphs consistent with the labels of $G_3 = K_z$ it is convenient to embed it into S^{23}. This can be done combinatorially by the *shaded rozigs*, see Figure 4.

21	14	61	18	A1	1C	E1	1G	31	15	71	19	B1	1D	F1
43	36	83	3A	C3	3E	G3	31	53	37	93	3B	D3	3F	23
65	58	A5	5C	E5	5G	15	53	75	59	B5	5D	F5	52	45
87	7A	C7	7E	G7	71	37	75	97	7B	D7	7F	27	74	67
A9	9C	E9	9G	19	93	59	97	B9	9D	F9	92	49	96	89
CB	BE	GB	B1	3B	B5	7B	B9	DB	BF	2B	B4	6B	B8	AB
ED	DG	1D	D3	5D	D7	9D	DB	FD	D2	4D	D6	8D	DA	CD
GF	F1	3F	F5	7F	F9	BF	FD	2F	F4	6F	F8	AF	FC	EF
12	23	52	27	92	2B	D2	2F	42	26	82	2A	C2	2E	G2
34	45	74	49	B4	4D	F4	42	64	48	A4	4C	E4	4G	14
56	67	96	6B	D6	6F	26	64	86	6A	C6	6E	G6	61	36
78	89	B8	8D	F8	82	48	86	A8	8C	E8	8G	18	83	58
9A	AB	DA	AF	2A	A4	6A	A8	CA	AE	GA	A1	3A	A5	7A
BC	CD	FC	C2	4C	C6	8C	CA	EC	CG	1C	C3	5C	C7	9C
DE	EF	2E	E4	6E	E8	AE	EC	GE	E1	3E	E5	7E	E9	BE
FG	G2	4G	G6	8G	GA	CG	GE	1G	G3	5G	G7	9G	GB	DG

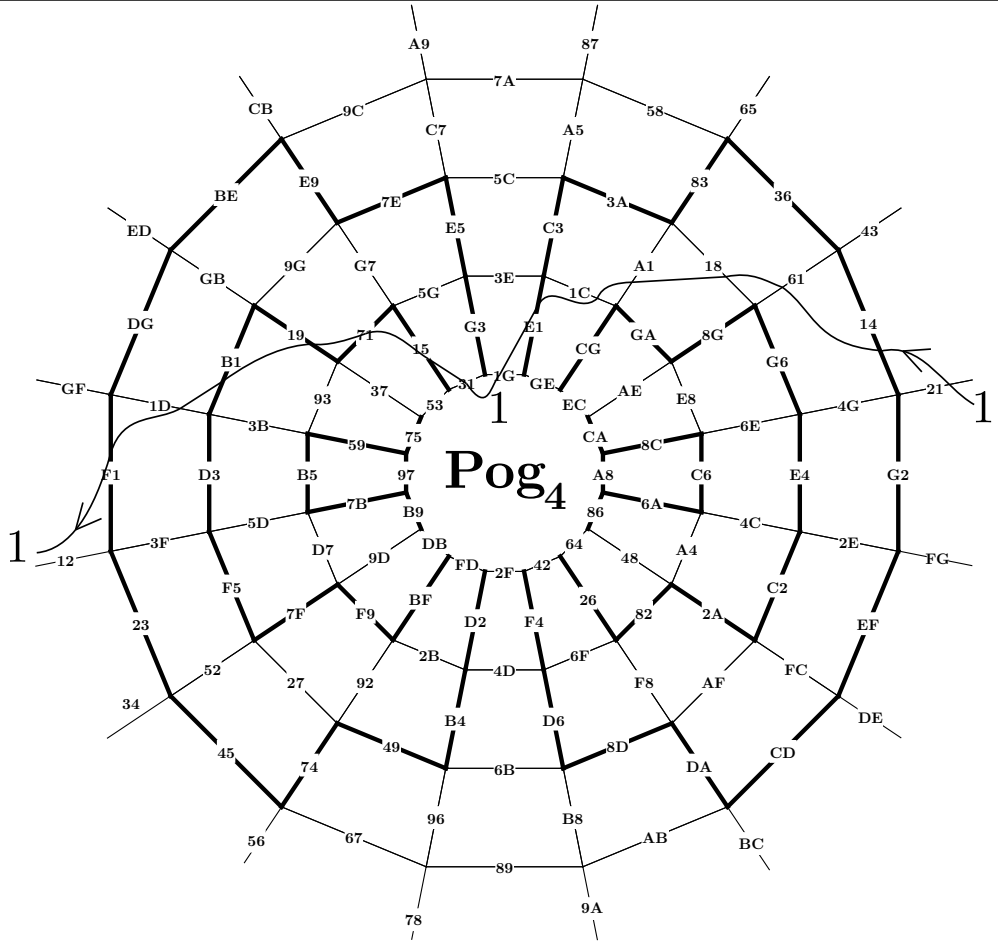


Figure 4 – Example of labeled shaded rooted oriented zigzags or *labeled shaded rozigs* $G_1 = Pog_4$. We also display a strong O-join, denoted by T, given in thick edges: the parity of the number of edges of T in the coboundary (boundary) of a vertex (a face) and the parity of the degree of the vertex (the face) of $G_1 = Pog_4$ coincide. The labels of the vertices of K_{16} are the digits in $1, 2 \ldots, 9, A, B, C, D, E, F, G$ (base 17). An edge of $G_3 = K_z$ is labeled by an unordered pair of vertices. Note that the face boundaries in clockwise order and the vertices coboundaries in counter-clockwise order correspond to directed polygons in $G_3 = K_z$. Rozig 1 is displayed.

Defining the other rows of the rozig table. To get row $i+1$ from row i in the rozig table just apply $suc2$ to the individual symbols of the pairs. This completes the definition of rozig table. From its rows we get a *rotation* for K_z, namely a cyclic ordering for the edges incident to each vertex i of K_z.

Yet another combinatorial counterpart for graphs embedded into surfaces. To obtain a combinatorial counterpart for an embedding of a graph we need a rotation (which we have: the rows of the rozig) together with the corresponding *twist* which is the subset of edges that are twisted for the fixed rotation. In our case, the twisted edges are the ones which correspond to the radial edges of Pog_h. The non-twisted ones correspond to the orbital edges of Pog_h. In terms of rozigs, a twisted edge is one traversed in opposite directions by the two zigzags that traverse the edge. The pair **(rotation,twist)** is sufficient to describe the embedding because from it we can recover the entire Q-graph: given an immersion respecting the rotation of Q (with crossings between the 1-colored edges) in the plane, given a twisted edge e the pair of edges of color 2 in the hyperedge of Q corresponding to e is replaced by the crossing edges.

The relevance of the shading. All the edges in a column of the rozig table are radial or all are orbital. We can shade the columns so that an edge is twisted in the rotation iff it is in a shaded column. In this way, shading defines the twist of the map and complements the rozigs completing its combinatorial presentation.

Defining the shading. The projective column is shaded, the left columns alternate (non-shaded, shaded) starting with non-shaded. The right columns are shaded or not according to the reflection of the left columns in the vertical line separating the left and right columns. See Figure 4.

3.3 LINEAR FORMULATION FOR MINSTRONGOJOIN AND MAX CUT

Suppose that $G_1 = Pog_h$ and G_2) are duals in $\mathbb{RP}^2$ and $G_3 = K_z$ ($z = 4h$) embedded in some surface as the phial of G_1. The common set of edges is denoted E. In order to prove that G_3 is rich is enough to prove that G_1 is rich. We have that $\dim(\mathcal{V}_1^{\perp}/\mathcal{F}_1) = (|E| - v + 1) - (f - 1) = |E| - v - f + 2 = -\chi + 2 = 1$, since $\chi(\mathbb{RP}^2) = 1$. Any zigzag in $\mathcal{Z}_1$ can be adjoined to $\mathcal{F}_1$ to generate the cycle space of G_1. Note that each zigzag is an orientation reversing polygon, so it is not in the span of the boundaries of the faces. Thus G_1 is rich, whence G_3 is rich.

Triangles and quadrangles in V_{12} spanning the cycle space of K_z. Denote by V_{12} the set of polygons p of length 3 and 4 of graph $G_3 = K_z$ which corresponds to the coboundary of the vertices of G_1 and G_2. We have $\langle V_{12}\rangle = \mathcal{V}_3^{\perp}$, because at most one polygon

(corresponding to the central face if $z \equiv 0 \mod 4$ or the central vertex if $z \equiv 2 \mod 4$) has number of sides distinct from 3 and 4. Note that this polygon is equal to the sum of all the other polygons (3- and 4-gons) in the same G_i.

We can now define the first of our polytopal formulations. It has a variable $x'_e \in \mathbb{R}^{|E|}$ for each $e \in E$ and a variable $s_p \in \mathbb{R}^{|V_{12}|}$ for each $p \in V_{12}$.

$$P'_0 = \begin{cases} p \in V_{12}: & 2s_p + \sum\{x'_e : e \subseteq p\} = |p| \\ \text{bounds:} & 0 \leq x'_e \leq 1 \ \forall e' \in E, s_p \geq 0, \forall p \in V_{12}. \end{cases}$$

Proposition 5. *P'_0 is a linear formulation for the MinStrongOJoin problem.*

Proof. Any characteristic vector of a strong O-join satisfies the linear restrictions of P'_0. Reciprocally, if (x'_e, s_P) is all integer and satisfy these restrictions it is the characteristic vector of a strong O-join. □

Double slack variables. Observe that each s_p appears once with coefficient 2. Therefore $\frac{s}{2}$ is a slack variable and s is called a *double slack variable*.

Valid inequalities. A *valid inequality* for a polytope is one which does not remove any of its points with all integer coordinates. It is straightforward to show that a linear formulation for a combinatorial problem remains so if we add valid inequalities. A class of valid inequalities will be added to P'_0 which permits the elimination of the double slack variables s_p and of the unitary upper bounds $x'_e \leq 1$.

Let $p \in V_{12}$ and $q \subset p$ so that $|p| + |q|$ is odd. The *pq-inequality* is

$$s_p + \sum\{x_e : e \subset q \subset p\} \leq \frac{|p| + |q| - 1}{2}.$$

The following theorem is central in this work.

Theorem 6. *The pq-inequalities eliminate fractional double slack variables s_p in the sense that after including them, integer $x'_e, e' \in E$ imply integer $s_p, p \in V_{12}$.*

Proof. Let $S^p_q = \sum\{x'_e : e \subset q \subset p\}$. The analysis for (0-1)-integers x'_e given in Figure 5 shows that the vertices with a fractional s_p are precisely the ones that violate some restriction. The neighborhood of a vertex in V_{12}. The thick edges have $x'_e = 1$ and the dashed edges have $x'_e = 0$. The whole coboundary of the vertex is the edge set of a polygon $p \in G_3$. □

By simply adding the *pq*-inequalities provides another linear formulation for the MinStrongOjoin problem:

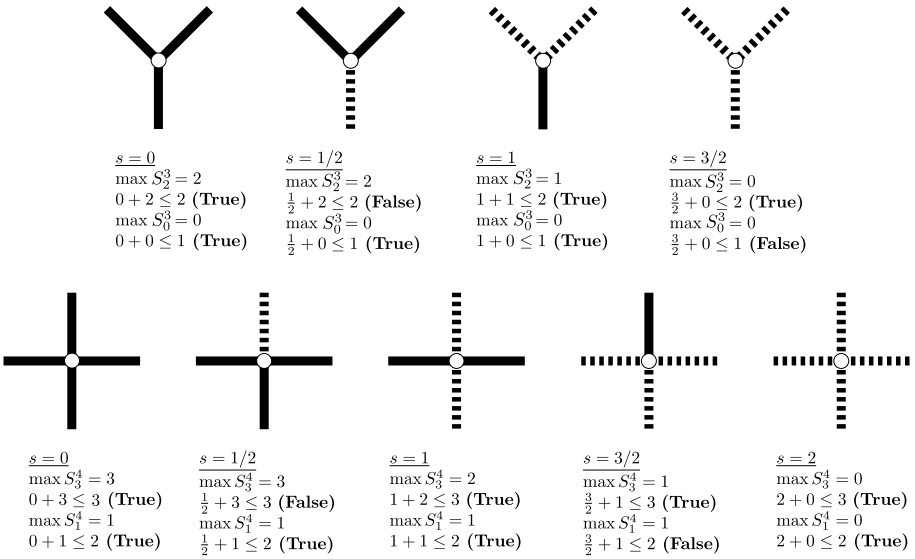


Figure 5 – Valid/violating pq-inequalities: $s + \max S_q^p \leq \frac{|p|+|q|-1}{2}$, with $|p|$ and $|q|$ of distinct parities where $q \subset p$. We impose $x'_e \in \{0, 1\}$. The condition characterizing a strong O-join can be localized to a neighborhood of a generic 3-vertex in V_{12}. The thick full edges are in q and their set constitutes a strong O-join. The coboundary of v is p, which is a polygon in G_3. The dashed edges are the ones in $p \backslash q$. For each half integer s one of its pq-inequalities is violated. If s is integer all its induced pq-inequalities are valid. Thus, integer x'_e, $e \in E$, plus pq-inequalities imply integer double slackness variables s_p, $p \in V_{12}$.

$$P'_1 = \begin{cases} p \in V_{12}: & 2s_p + \sum\{x'_e : e \subseteq p\} = |p| \\ q \subset p \in V_{12}: & s_p + \sum\{x'_e : e \subset q \subset p\} \leq \frac{|p|+|q|-1}{2} \\ \text{bounds:} & 0 \leq x'_e \leq 1, s_p \geq 0. \end{cases}$$

Since integrality of x'_e imply integrality of the s_p and each of these appears once in an equation, we can dispose of these double slackness variables variables by considering its implicit definition,

$$s_p(integer) := \frac{|p| - \sum\{x'_e : e \subseteq p\}}{2} \quad \text{we obtain,}$$

$$P'_2 =$$

$$\begin{cases} q \subset p \in V_{12}, |p|-|q| \text{ odd}: & |p| - \sum\{x'_e : e \subseteq p\} + \sum\{2x'_e : e \subset q \subset p\} \leq |p| + |q| - 1 \\ \text{bounds:} & 0 \leq x'_e \leq 1 \quad \forall e \in E. \end{cases}$$

Consider $q \subset p \in V_{12}$, $|p| - |q|$ odd:

$$|p| - \sum\{x'_e : e \subseteq p\} + \sum\{2x'_e : e \subset q \subset p\} \leq |p| + |q| - 1$$

$$\iff |p| - \sum\{x'_e : e \subset p\backslash q\} - \sum\{x'_e : e \subseteq q \subset p\} + \sum\{2x'_e : e \subset q \subset p\} \leq |p| + |q| - 1$$

$$\iff \sum\{x'_e : e \subseteq q \subset p\} - \sum\{x'_e : e \subset p\backslash q\} \leq |q| - 1.$$

Theorem 7. *The polytope*

$$P_2' = \begin{cases} q \subset p \in V_{12}, |p| - |q| \quad odd: & \sum\{x'_e : e \subset q \subset p\} - \sum\{x'_e : e \subset p \backslash q\} \leq |q| - 1 \\ bounds: & 0 \leq x'_e \leq 1 \quad \forall e \in E. \end{cases}$$

is a linear formulation for the MinStrongOjoin problem.

Proof. It is straightforward from the equivalences above. □

We want to get a linear formulation for the Max Cut problem. Given P_2' it is enough to replace each variable x'_e by $x_e = 1 - x'_e$. This has the effect of complementing the characteristic vectors and the minimization problem becomes a maximization one. We get

$$\begin{gathered} \sum\{x'_e : e \subseteq q \subset p\} - \sum\{x'_e : e \subset p \backslash q\} \leq |q| - 1 \\ \iff \sum\{1 - x_e : e \subseteq q \subset p\} - \sum\{1 - x_e : e \subset p \backslash q\} \leq |q| - 1 \\ \iff |q| - \sum\{x_e : e \subseteq q \subset p\} - (|p| - |q|) + \sum\{x_e : e \subset p \backslash q\} \leq |q| - 1 \\ \iff \sum\{x_e : e \subset p \backslash q\} - \sum\{x_e : e \subseteq q \subset p\} \leq |p| - |q| - 1 \\ \iff \sum\{x_e : e \subset p \backslash q\} - \sum\{x_e : e \subseteq q \subset p\} \leq |p| - p^- - 1 \\ \iff \sum\{x_e : e \subset p \backslash q\} - \sum\{x_e : e \subseteq q \subset p\} \leq p^+ - 1. \end{gathered}$$

Sign of an edge in a polygon. Edges in $p \backslash q$ have sign +1 and edges in q have sign -1.

Let σ_p^e be the sign of edge e in polygon p. Let p^+ and p^- denote respectively the number of $+1$ signs and -1 signs on the edge variables of the polygon p. Note that $|p| - |q|$ odd $\Rightarrow |p| - p^-$ odd $\Rightarrow p^+$ odd. Then the last equivalence can be rewritten as

$$\sum\{\sigma_p^e x_e : e \subset p\} \leq p^+ - 1.$$

Let $S_{12}^{\pm}$ denote the polygons in V_{12} arbitrarily signed except for the fact that p^+ is odd. Note that we have disposed the q's by using signed polygons. In the Theorem below the linear restrictions forming the polytope are induced by signed forms of the coboundaries of the vertices and the signed forms of the boundary of the faces of map $G_1 = Pog_h \hookrightarrow \mathbb{RP}^2$. The phial graph of G_1 is $G_3 = K_z, z = 4h$, embedded into some higher genus surface S^{23}, which does not concern us except for the practical fact that $G_3 = K_z \hookrightarrow S^{23}$ via shaded rozigs is the easier way to obtain combinatorially the graphs $G_1 = Pog_h$ and its dual G_2 in $\mathbb{RP}^2$ so that G_1, G_2, G_3 have the same edge set E.

Theorem 8. *The polytope*

$$\mathbb{P}_{12} = \begin{cases} p \in S_{12}^{\pm}: & \sum\{\sigma_p^e x_e : e \subset p\} \leq p^+ - 1 \\ \textit{bounds:} & x_e \geq 0 \quad \forall e \in E. \end{cases}$$

is a linear formulation for the Max Cut problem on the complete graph K_z, z even.

Proof. It is straightforward from the equivalences above, except for the unitary upper bounds. Given any $ij \in E$ there is in G_1 in a coboundary of a vertex or the boundary of a face of degree 3 or 4 containing ij. The variables correspond to unoriented edges. So we have $x_{ij} = x_{ji}$ for every pair of distinct vertices of $G_3 = K_z$.

Case 3. In the first case there is a k so that $x_{ij} + x_{jk} + x_{ki} \leq 2$ and $x_{ij} - x_{jk} - x_{ki} \leq 0$. Adding these, $2x_{ij} \leq 2$ or $x_{ij} \leq 1$.

Case 4. If ij is in the coboundary of a vertex or in the boundary of a face of degree 4, there are k and l so that $x_{ij} - x_{jk} - x_{kl} - x_{li} \leq 0$, $x_{ij} + x_{jk} + x_{kl} - x_{li} \leq 2$. Adding we get $2x_{ij} - 2x_{li} \leq 2$ or $x_{ij} - x_{li} \leq 1$. We also have $x_{ij} + x_{jk} - x_{kl} + x_{li} \leq 2$, $x_{ij} - x_{jk} + x_{kl} + x_{li} \leq 2$. Adding we get $2x_{ij} + 2x_{li} \leq 4$ or $x_{ij} + x_{li} \leq 2$. The inequalities imply that $2x_{ij} \leq 3$ and, since x_{ij} is an integer, $x_{ij} \leq 1$. □

Estimating $|S_{12}^{\pm}|$. For this estimation we count the number of 3-vertices, 4-vertices, 3-faces and 4-faces of G_1. If h is an integer, then the number of 3-vertices is $z = 4h$ and the number of 3-faces is 0. The number of 4-vertices of G_1 is $z(h-1)$. The number of 4-faces is $z(h-1) + z/2$. If h is a half integer, then the number of 3-vertices is 0, the number of 3-faces is $z = 4h$. The number of 4-vertices is $\lfloor h \rfloor z$. The number of 4-faces is $(\lfloor h \rfloor - 1)z + z/2$.

Unifier of vertices and faces. Let a *unifier* be either a vertex or a face or G_1. If h is integer the number of 3-unifiers is z and the number of 4-unifiers is $z(h-1) + z(h-1) + z/2 = 2z(h-1) + z/2$. If h is a half integer, then the number of 3-unifiers is z and he number of 4-unifiers is $\lfloor h \rfloor z + (\lfloor h \rfloor - 1)z + z/2 = 2\lfloor h \rfloor z - z/2$.

Cardinality of $S_{12}^{\pm}$ in terms of unifiers. This cardinality is 4 times the number of 3-unifiers plus 8 times the number of 4-unifiers of G_1. Thus, if h is an integer, the $|S_{12}^{\pm}|$ is $4z + 8(2z(h-1) + z/2) = 8z + 16zh - 16z = 16zh - 8z \leq 16zh = 4z^2$. If h is a half integer, then $|S_{12}^{\pm}|$ is $4z + 8(2\lfloor h \rfloor z - z/2) = 16\lfloor h \rfloor z \leq 16zh = 4z^2$. Thus, in every case, $|S_{12}^{\pm}| \leq 4z^2 = O(|E|)$. In fact we have $4z^2 \leq 10|E| = 5z^2 - 5z \iff 5z \leq z^2 \iff z \geq 5$, which is clearly true, since there is no use in working with K_4.

Theorem 9. *The number of linear inequalities defining P_2 is at most $11|E|$. Each of them involves the sum of no more than 4 edge variables with ± 1 coefficients. The right hand side of them is either 0, 1 or 2.*

Proof. We have established in the above discussion that $|S_{12}^{\pm}| \leq 10|E|$. There are $|E|$ inequalities corresponding to the non-negativity of the variables. The bounds on each inequality are directly seen to hold. So the result follows. □

3.4 CONSTRUCTION OF TOROIDAL ORBITAL GRAPHS AND THEIR MEDIALS

In this section we develop an algorithm to construct phials of K_z with z odd, complementing the even case. We define the medial of a map as an embedding of a 4-regular graph formed by placing a vertex in the middle of each edge of the map and defining an edge for each angle (pair of adjacent edges) of the original map (Figure 6). The surface induced by the combinatorial embedding of the phial of K_z is the orientable surface of genus $\frac{z-1}{2}$. Recall that in the even case the surface is always the real projective plane. However, we show the that in the medial the non-planar elements are redundant.

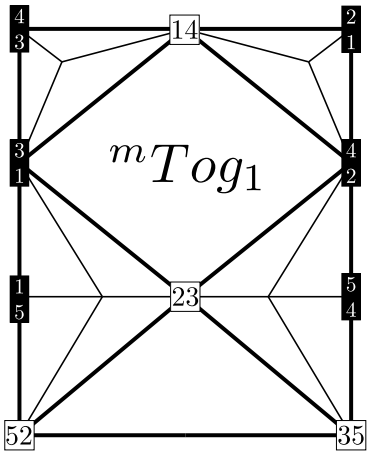
The construction of the medial map is easily accomplished from the toroidal orbital graph. For simplicity, consider the case of $z = 5$ (Figure 6a). Vertex $4 - 3$ is now placed where the midpoint of edge $4 - 3$ was at Tog_1. The same goes for every other vertex. The vertex labeling is done threefold: the name indicating the correspondent edge, the color to indicate where the edge is crossing (black) or on the same side of the cut (white) and finally the shape to indicate if the value of the variable associated with that edge is present on the current objective function (i.e: the edge is present on the subgraph). Rectangle indicated the edge is present whereas a square indicate it is not. The extrapolation for the general case follows the same rules (see cases $z = 7$ and $z = 9$ on Figures Figure 6b, Figure 6c).

The ${}^{m}Tog_h$ is the medial of Tog_h where $4h + 1 = z$. Tog_h is the map induced by the shaded, rooted and oriented zigzags (vertices of K_z). The description of Tog_h is more complicated than the even case z. We present examples which permit an easy generalization.

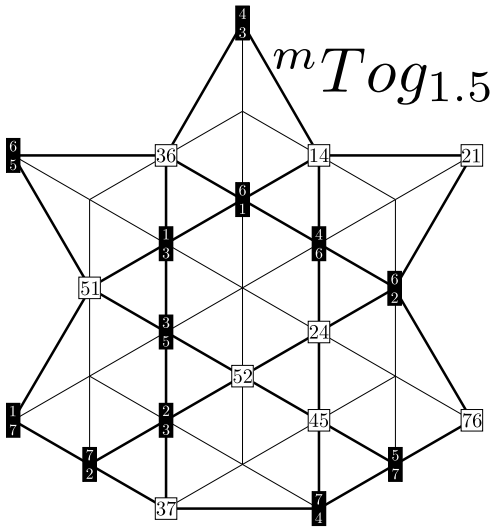
The construction of the rozigs table goes as follows: the first row is the zigzag for the first vertex of the phial (1). Similarly, the last row is the zigzag for the last vertex of the phial (z). The first and last columns have a special rule and all other entries are computed via their *2-successor*. The 2-successor of a vertex v is defined as follows:

$$succ_2^z(v) = \begin{cases} z, & \text{if v=z} \\ 1, & \text{if v=z-1} \\ 2, & \text{if v=z-2} \\ v+2, & \text{otherwise} \end{cases}$$

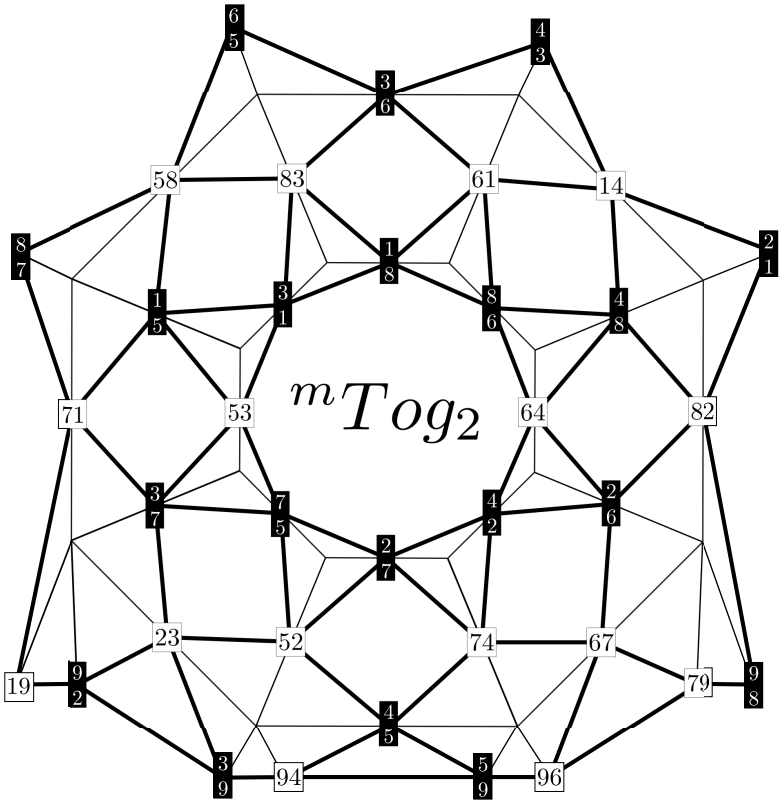
with $v \in \{1, 2, \ldots, z\}$. Take for instance the case $z = 5$ (Figure 7), the first half of the first column as well as the second half of the second column (orange) are computed as the



(a) Medial for $z = 5$



(b) Medial for $z = 7$



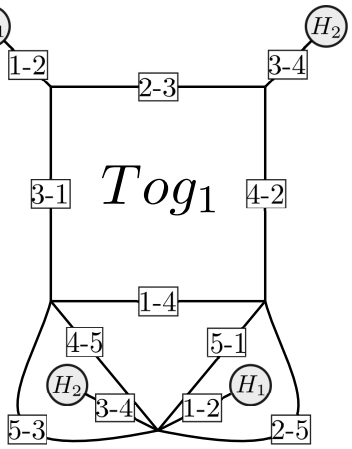
(c) Medial for $z = 9$

Figure 6 – Medial for cases $z = \{5, 7, 9\}$. The construction of the medial is done by connecting the angles between neighboring edges at the edges midpoints.

successors of the first pair $(2-1)$. The second half of the first column is computed through the even vertices of the zth zigzag while the first half of the last column is computed through the odd vertices of the zth zigzag (light blue). Examples for $z = 7$ (Figure 8) and $z = 9$ (Figure 9) are included to facilitate the extrapolation for any odd z.

21	14	31	15
43	31	23	35
52	23	42	21
54	42	14	43
15	52	35	54

(a) Shaded, rooted and oriented zigzags for K_5.

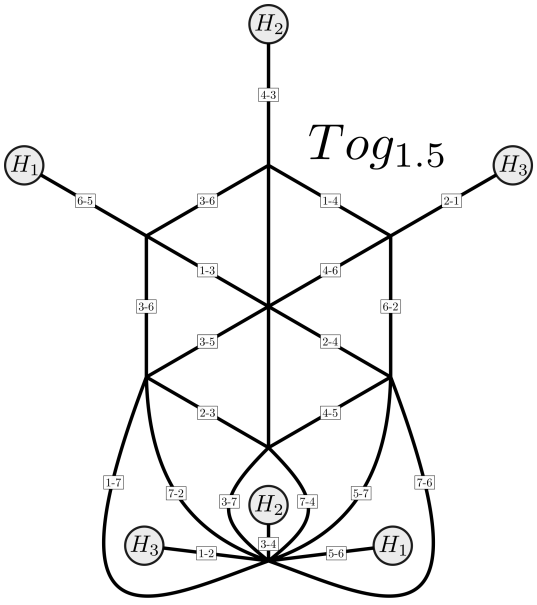


(b) Toroidal Orbital Graph for $z = 5$.

Figure 7 – Rozigs table and its associated Tog_1 for the case $z = 5$.

21	14	61	13	51	17
43	36	13	35	23	37
65	51	35	52	45	57
72	23	52	24	62	21
74	45	24	46	14	43
76	62	46	61	36	65
17	72	37	74	57	76

(a) Shaded, rooted and oriented zigzags for K_5.



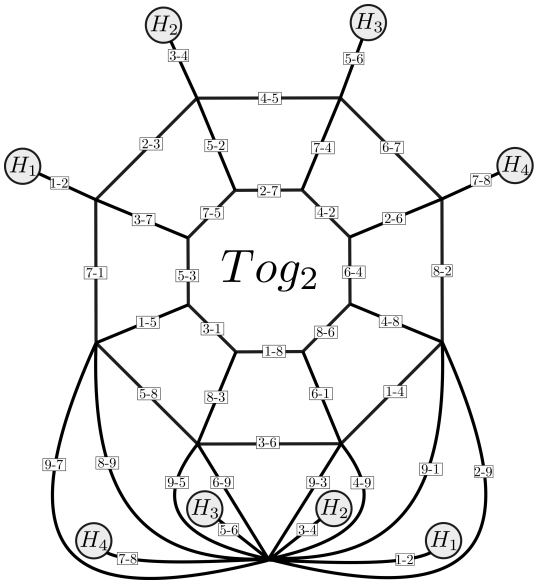
(b) Toroidal Orbital Graph for $z = 7$.

Figure 8 – Rozigs table and its associated $Tog_{1.5}$ for the case $z = 7$.

Unlike the projective orbital graphs described on section 3.2, the naive formulation of the linear program by including the cycle inequalities for every face and vertex coboundary for the odd case is exponential due to the middle face/vertex. The largest vertex and face are redundant but the central face (for cases integer h) or central vertex (for cases half-integer h) remains. A simple solution is to formulate the problem by inequalities from three distinct origins, all of which are cycle inequalities of triangles dubbed *triangle inequalities*.

21	14	61	18	31	15	71	19
43	36	83	31	53	37	23	39
65	58	15	53	75	52	45	59
87	71	37	75	27	74	67	79
92	23	52	27	42	26	82	21
94	45	74	42	64	48	14	43
96	67	26	64	86	61	36	65
98	82	48	86	18	83	58	87
19	92	39	94	59	96	79	98

(a) Shaded, rooted and oriented zigzags for K_9.



(b) Toroidal Orbital Graph for $z = 9$.

Figure 9 – Rozigs table and its associated Tog_2 for the case $z = 9$.

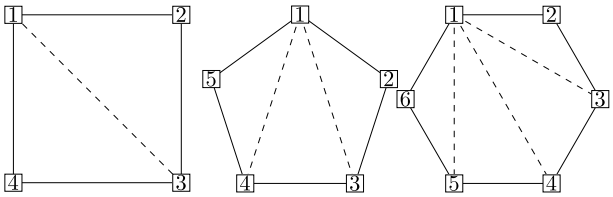


Figure 10 – Polygons with more than three vertices are broken into triangles by the addition of chords from the lexicographically least vertex to all other non-adjacent vertices. For example, square 1234 is broken into two triangles 123 and 134. An analogous procedure is done of larger polygons.

First, add all inequalities originated from triangles of Tog. Then break every other cycle by adding a chord from the lexicographically smallest vertex to the next non-adjacent such that only triangles remain (Figure 10).

Finally, triangle inequalities are added too guarantee full coverage of all edges of Tog. The significance of these inequalities are to ensure that every primal variable of the linear problem will assume value within the $[0, 1]$ interval. Recall that given three distinct vertices $\{i, j, k\}$,

$$\begin{aligned} & x_{ij} + x_{jk} + x_{ki} \leq 2 \\ + \quad & x_{ij} - x_{jk} - x_{ki} \leq 0 \\ \hline & 2x_{ij} \leq 2. \end{aligned}$$

$$i + j + k_{ij}^{l} = l \tag{3.1}$$

$$k_{ij}^{l} = \{l - i - j\} \tag{3.2}$$

$$k_{ij} = \min_{l \in 0,1,2} \{k_{ij}^{l}\}. \tag{3.3}$$

Therefore, given i, j, k must be found such that every possible edge in the original subgraph is covered. Equation Equation 3.3 presents a method of finding such values of k given a pair i, j. Such an example of a formulation for the Max Cut problem for the case $z = 5$ is available at Appendix Appendix 5.

This formulation however is quite extensive. It is possible to extract directly from ${}^{m}Tog_{h}$ a compact relaxed formulation where the faces and coboundaries of vertices are the dual variables and the edges are the primal variables.

$$Primal: \begin{cases} \max \sum \{c_v x_v : v \in V\} \\ \text{Subject to} \\ \quad \forall f \in F,\ y_f : 2x_f - \sum \{x_v : v < f\} = 0 \\ \quad \forall v \in V,\ y_v : x_v \leq 1 \\ \text{Bounds:} \\ \quad x_v \geq 0,\ x_f \geq 0 \end{cases}$$

$$Dual: \begin{cases} \min \sum \{y_v : v \in V\} \\ \text{Subject to} \\ \quad \forall f \in F,\ x_f : 2y_f \geq 0 \\ \quad \forall v \in V,\ x_v : y_v - \sum \{y_f : f > v\} \geq c_v \\ \text{Bounds:} \\ \quad y_v \geq 0,\ y_f \text{ free} \end{cases}$$

Since there exists no known integer hull for the Max Cut problem, the above polyhedron is relaxed. In order to guarantee an actual cut it is necessary to enforce x_v binary and x_f integer at the primal. A formulation for the case $z = 5$ is as follows.

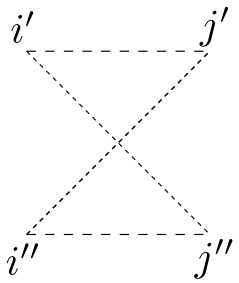


Figure 11 – The graph $H(x^*)$ can be found by duplicating each vertex v in the graph G into v' and v'' and adding new edges from v' to all other vertices except v''. For any two distinct vertices i, j an edge has weight x^*_{ij} if it goes from i' to j' or from i'' to j'' and weight $1 - x^*_{ij}$ otherwise where x^*_{ij} is the value of the variable x_{ij} on the current solution x^*.

$$
\begin{aligned}
y134 : &\quad 2x_{134} - x_{13} - x_{34} - x_{14} = 0 \\
y124 : &\quad 2x_{124} - x_{12} - x_{24} - x_{14} = 0 \\
y235 : &\quad 2x_{235} - x_{23} - x_{35} - x_{25} = 0 \\
y1324 : &\quad 2x_{1324} - x_{13} - x_{23} - x_{24} - x_{14} = 0 \\
y1325 : &\quad 2x_{1325} - x_{13} - x_{23} - x_{25} - x_{15} = 0 \\
y2354 : &\quad 2x_{2354} - x_{23} - x_{35} - x_{45} - x_{24} = 0
\end{aligned}
$$

The compact formulation acts as a starting point. Theron after it is possible to include triangular inequalities to enforce the bounds and to find violating polygons in order to prune fraction vertices of the polytope. There are several examples on the current literature proposing methods and techniques to find valid inequalities for the cut polyhedra.

A straightforward method for finding violating polygons as proposed by Barahona, Jünger e Reinelt (1989) and followed by Lancia e Serafini (2011) takes on from the graph $H(x^*)$. Every vertex in the original graph G appears double as v' and v''. There is an edge connecting all vertices except from v' to v'' for all v and the edge has weight x^*_{ij} if the edge goes from i' to j'' or from i'' to j' and $1 - x^*_{ij}$ otherwise where x^*_{ij} is the value for the variable x_{ij} at the current primal optimum.

Let s^i_p be the length of the shortest path $i' \Rightarrow i''$ inducing a polygon p which uses vertex i. Therefore $s^i_p = p^+ - x^*_p$.

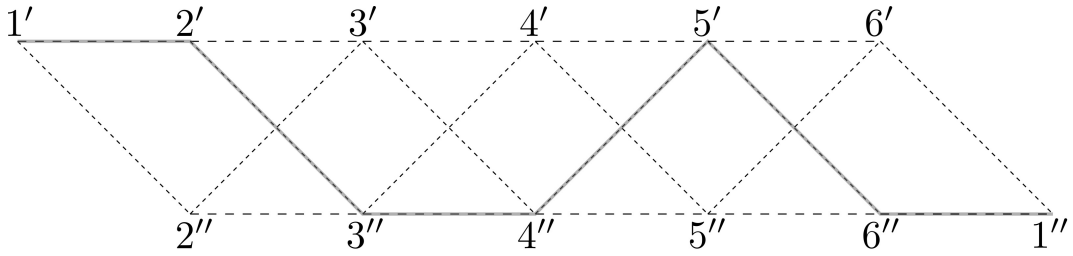


Figure 12 – A violating signalization for the huge polygon at the center can be found by the same means as normal violating polygons are found. Construct the path on $H(x^*)$ from $1'$ to $1''$ going through all other vertices. If the shortest path is smaller than 1 then the edges corresponding to segments of the shortest path going from v' to $(v+1)''$ has coefficient -1 and all others have coefficient 1. This particular path induces the polygon $x_{12} - x_{23} + x_{34} - x_{45} - x_{56} + x_{16} \leq 2$

We have:

$$\begin{aligned} s_p^i < 1 &\iff p^+ - x_p^* < 1 \\ &\iff -p^+ + x_p^* > -1 \\ &\iff x_p^* > -1 \\ &\iff x_p^* > p^+ - 1 \\ &\iff p \text{ is a violating polygon} \end{aligned}$$

where $x_p^* = \sum\{\sigma_p^e x_e^* : e \in p\}$ and p^+ is the number of positive signs in p. A Floyd-Warshall (CORMEN, 2009) algorithm can be employed to find all violating polygons given a primal optimum x^*.

This method can be used to find a signalization for a violating *Tog* central polygon. Since this polygon includes all vertices of the graph, if such a violating polygon is found, a cut can be immediately found. This is done by constructing a graph with all double vertices but only linking edges between lexicographically consecutive vertices and applying Dijkstra's algorithm and determining if a path whose sum of weights of edges is smaller than 1 (Figure 12).

In addition to violating polygons, it is also possible to add gap inequalities to the polytope based on Laurent e Poljak (1996). Given a maximal primal x^*, a symmetric matrix M is constructed by attributing value $1-2x_{ij}^*$ to M_{ij} and M_{ji} where x_{ij}^* is the value for variable x_{ij} at x^* and M_{ij} is the corresponding cell at row i and column j. The next step is to extract the eigenvectors and eigenvalues from M. For every negative eigenvalue the coefficients of the new constraint is calculated from the corresponding eigenvector as follows:

Let b^* be an eigenvector of M whose eigenvalue is strictly negative. Let u^* be the sum of all elements in b^*. Let $b_i = \lfloor b_i^* \frac{z}{u^*} \rfloor$ where i is the element index on b^* and z is the

number of vertices on the subgraph. Then the positive semi-definite cut of the eigenvector b^* is given by

$$\sum_{i=1}^{z-1} \sum_{j=i+1}^{z} (b_i b_j x_{ij}) \leq \frac{\sum(b)^2}{4}.$$

It is important to note that cliques are a special case of these inequalities where all coefficients are equal to one.

3.5 2-SAT SOLVER

It is possible to extract a set of boolean clauses from each restriction. As seen, every triangle inequality possesses four separate forms of attributing the signalization such that the number of positive terms is odd: either all the terms have a positive coefficient or only one term does. Consider the first case containing only positive terms: $x_{ij} + x_{jk} + x_{ki} = 2$. Since all variables assume $0 - 1$ value, there are only three ways in which the equation is satisfied: one of the terms must be equal to 0 and the other two equal to 1 such that

$$x_{ij} + x_{jk} + x_{ki} = 2 \leftrightarrow 011 \vee 101 \vee 110.$$

Let b_{ij} be a boolean variable which is true if $x_{ij} = 0$ and false otherwise. In this manner it is possible to formulate the following clauses

$$(\overline{b_{ij}} \wedge b_{jk} \wedge b_{ki}) \vee (b_{ij} \wedge \overline{b_{jk}} \wedge b_{ki}) \vee (b_{ij} \wedge b_{jk} \wedge \overline{b_{ki}})$$

in disjunctive normal form (DNF). In conjunctive normal form (CNF), the above is written as

$$(\overline{b_{ij}} \vee \overline{b_{jk}} \vee \overline{b_{ki}}) \wedge (b_{ij} \vee b_{jk}) \wedge (b_{jk} \vee b_{ki}) \wedge (b_{ki} \vee b_{ij}).$$

Developing the all triangle inequalities in the same manner yields the following set of clauses

$$
\begin{aligned}
x_{ij} + x_{jk} + x_{ki} = 2 &\leftrightarrow 011 \vee 101 \vee 110 \\
&\equiv (\overline{b_{ij}} \wedge b_{jk} \wedge b_{ki}) \vee (b_{ij} \wedge \overline{b_{jk}} \wedge b_{ki}) \vee (b_{ij} \wedge b_{jk} \wedge \overline{b_{ki}}) \\
&\equiv (\overline{b_{ij}} \vee \overline{b_{jk}} \vee \overline{b_{ki}}) \wedge (b_{ij} \vee b_{jk}) \wedge (b_{jk} \vee b_{ki}) \wedge (b_{ki} \vee b_{ij}) \\
x_{ij} - x_{jk} - x_{ki} = 0 &\leftrightarrow 000 \vee 101 \vee 110 \\
&\equiv (\overline{b_{ij}} \wedge \overline{b_{jk}} \wedge \overline{b_{ki}}) \vee (b_{ij} \wedge \overline{b_{jk}} \wedge b_{ki}) \vee (b_{ij} \wedge b_{jk} \wedge \overline{b_{ki}}) \\
&\equiv (\overline{b_{ij}} \vee \overline{b_{jk}} \vee \overline{b_{ki}}) \wedge (b_{ij} \vee \overline{b_{jk}}) \wedge (\overline{b_{jk}} \vee \overline{b_{ki}}) \wedge (\overline{b_{ki}} \vee b_{ij}) \\
-x_{ij} + x_{jk} - x_{ki} = 0 &\leftrightarrow 000 \vee 011 \vee 110 \\
&\equiv (\overline{b_{ij}} \wedge \overline{b_{jk}} \wedge \overline{b_{ki}}) \vee (\overline{b_{ij}} \wedge b_{jk} \wedge b_{ki}) \vee (b_{ij} \wedge b_{jk} \wedge \overline{b_{ki}}) \\
&\equiv (\overline{b_{ij}} \vee \overline{b_{jk}} \vee \overline{b_{ki}}) \wedge (\overline{b_{ij}} \vee b_{jk}) \wedge (b_{jk} \vee \overline{b_{ki}}) \wedge (\overline{b_{ki}} \vee \overline{b_{ij}}) \\
-x_{ij} - x_{jk} + x_{ki} = 0 &\leftrightarrow 000 \vee 011 \vee 101 \\
&\equiv (\overline{b_{ij}} \wedge \overline{b_{jk}} \wedge \overline{b_{ki}}) \vee (\overline{b_{ij}} \wedge b_{jk} \wedge b_{ki}) \vee (b_{ij} \wedge \overline{b_{jk}} \wedge b_{ki}) \\
&\equiv (\overline{b_{ij}} \vee \overline{b_{jk}} \vee \overline{b_{ki}}) \wedge (\overline{b_{ij}} \vee \overline{b_{jk}}) \wedge (\overline{b_{jk}} \vee b_{ki}) \wedge (b_{ki} \vee \overline{b_{ij}}).
\end{aligned}
$$

Note that for every case in CNF the expression is written as one large clause containing three terms and three other clauses each with two terms. The corresponding truth table is for the boolean variables is

s_i	s_j	s_k	$\overline{b_{ij}}$	$\overline{b_{jk}}$	$\overline{b_{ki}}$	$\overline{b_{ij}} \vee \overline{b_{jk}} \vee \overline{b_{ki}}$
0	0	0	1	1	1	1
0	0	1	1	0	0	1
0	1	0	0	0	1	1
0	1	1	0	1	0	1
1	0	0	0	1	0	1
1	0	1	0	0	1	1
1	1	0	1	0	0	1
1	1	1	1	1	1	1

Table 1 – Truth table for the sidevars and boolean variables corresponding to the equation $x_{ij} + x_{jk} + x_{ki} = 2$

where s_i is a *sidevar* indicating which side of the cut's bipartition the vertex i of the original subgraph belongs to. A canonical way of interpreting this is placing vertex i on the left side if s_i is true. Two significant pieces of information can be immediately extracted from Table Table 1: first the lower half of the table is a mirrored image of the top half, locally this indicates that the partitioning can be mirrored by swapping all vertices; second the last column, which corresponds to the large clause is a tautology. This means that in CNF form the problem reduces to a 2-SAT. Tables Table 2,Table 3,Table 4 correspond to the remaining set of clauses.

Definition: Let x^* be a primal and y_* a dual optimal solution. An edge e_{ij} is in the *equality subgraph* if the dual restriction corresponding to x_{ij} holds as an equality.

s_i	s_j	s_k	$\overline{b_{ij}}$	b_{jk}	b_{ki}	$\overline{b_{ij}} \vee b_{jk} \vee b_{ki}$
0	0	0	1	0	0	1
0	0	1	1	1	1	1
0	1	0	0	1	0	1
0	1	1	0	0	1	1

Table 2 – Truth table for the sidevars and boolean variables corresponding to the equation $x_{ij} - x_{jk} - x_{ki} = 0$

s_i	s_j	s_k	b_{ij}	$\overline{b_{jk}}$	b_{ki}	$b_{ij} \vee \overline{b_{jk}} \vee b_{ki}$
0	0	0	0	1	0	1
0	0	1	0	0	1	1
0	1	0	1	0	0	1
0	1	1	1	1	1	1

Table 3 – Truth table for the sidevars and boolean variables corresponding to the equation $-x_{ij} + x_{jk} - x_{ki} = 0$

s_i	s_j	s_k	b_{ij}	b_{jk}	$\overline{b_{ki}}$	$b_{ij} \vee b_{jk} \vee \overline{b_{ki}}$
0	0	0	0	0	1	1
0	0	1	0	1	0	1
0	1	0	1	1	1	1
0	1	1	1	0	0	1

Table 4 – Truth table for the sidevars and boolean variables corresponding to the equation $-x_{ij} - x_{jk} + x_{ki} = 0$

Theorem 10. *Suppose we have a formulation for the Max Cut composed only of triangular restrictions. Note that these restrictions imply* $0 \leq x_{ij} \leq 1$. *As a consequence we can declare that* x_{ij} *is free for all* i, j. *Then, all edges* e_{ij} *are in the equality subgraph.*

Proof. Since x_{ij} is free, the dual restriction y_{ij} corresponding to x_{ij} holds as an equality. □

Definition: Let y_* be a dual minimum solution for a formulation of the Max Cut and R be the set of restrictions such that $(y_*)_r > 0$. Then y_* is *rich* if for each sidevar s, there is a $r \in R$ with $s \in r$.

Theorem 11. *Suppose we have a Max Cut formulation composed only of triangular inequalities and a dual viable solution for it with integer minimum* y_* *with value* $\alpha \in \mathbb{Z}$. *Let* T *be the set of triangles in* K_z *such that* $(y_*)_t > 0$ *so that the restrictions in* T *hold as equalities. If* T *is rich and connected, then there is a 2-SAT to decide whether there is an integer primal solution with value* α.

Proof. An integer solution x^* with value α exists if and only if x^* is feasible and the pair x^*, y_* satisfies the complementary slackness. Since all edges are in the equality subgraph, complementary slackness is reduced to imposing that the restrictions in T hold. Since each

such equality corresponds to a set of boolean clauses composed of two literals, solving for the set of restrictions in T is equivalent to solving the 2-SAT. Since satisfying the restrictions in T induces a cut by formulation, then the solution for the 2-SAT will set all sidevars s such that the cardinality of the corresponding cut has value α. If there is no solution then the decision is immediate and if there are any solutions it is trivial to determine feasibility given connectivity. □

4 METHODOLOGY

In this chapter, three new different approaches are presented, capable of providing a solution for the models present in chapter 3 along with a proof of optimality. Integer linear programming may involve modifications of the relaxed problem's polytope through cutting planes (as proposed by Gomory (1958)). Such operations are analogous to a change of basis on the current matrix and thus can hardly be verified unless one keeps all basis as well as all pivots involved in each change. So the verification problem is simply overlooked by the solvers. One contribution of this thesis is to address this issue by providing an effective verification to the optimality of the solution in the particular case of the Max Cut for small graphs.

The first approach was to create a connected-graph random generator to work as backbone input. Preliminary experiments indicated that number of vertices, edges and symmetry had the most significant impact on the results. Therefore, the experiments were run based foremost the number of vertices, z, considered the size of the problem. For each such category, the experiments were this based on either random graphs with specific density (ratio of edges in the subgraph by edges in the complete graph) or specific known symmetrical graphs detailed in Appendix 5.

The algorithms run by expanding binary trees as they progress on finding the Max Cut. The trees are always rooted with a single right child which indicates the canonical

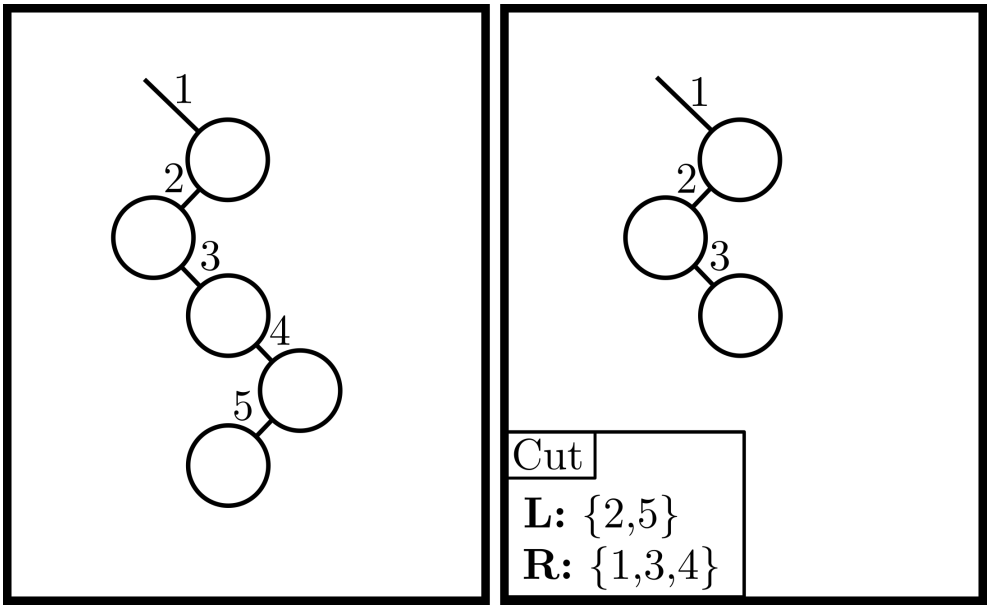


Figure 13 – Two binary trees for a problem of 5 vertices. Both trees induce $\{2, 5\}$ and $\{1, 3, 4\}$ bipartitions for the cut, however, only the tree on the left is complete. Therefore, the tree on the right requires additional information of the bipartition.

placement of 1 on the right partition. In order to indicate this, the label 1 is placed on the edge between the root and its only child. Thereon after, every node on the tree will have either two children or none. Every new edge expanding into a child will contain a label to indicate its corresponding vertex on the subgraph. By following a left label, the vertex is placed on the left partition and analogous to the right. Therefore, every node on the tree partially defines an partition for the cut, unique for each node. Although conceptually indispensable, as data structure, the root node is treated as implicit (pre-root) and its right child as root.

4.1 COMPACT TREE

Consider the randomly generated graph in Figure 14a, in order to provide sufficient proof of correctness, the algorithm builds a single binary tree T_c, which provides on each node the relaxed optimum for a given integer partial variable configuration. The algorithm targets the greatest integer smaller than the largest leaf node. This mark is called α. The algorithm attempts to find an integer solution to the problem with the same value as α with a DFS-like approach, placing vertices on lexicographical order. Each time the algorithm goes down a level on the tree, the correspondent vertex is placed in the adequate bipartition and a relaxed LP is solved. The optimum is attributed to that node and the algorithm checks to see if the value is lower than the target. If that is so, it backtracks to the next open node. Otherwise the algorithm will go deeper on the tree until solved or bursts the target. An illustration of the algorithm for the given example can be seen in Figure 14b. The vertex 1 is canonically placed on the right but as there are no other set vertices, the relaxed LP is solved and its value placed as a label for the root. In order to simplify the algorithm and improve code efficiency, the left node is always expanded first and given a choice, the algorithm will prefer expanding the closest node. Vertex 2 is placed left (across from 1). The LP is modified to reflect this and solved with the appropriate variables now set as integer. The corresponding optimum is placed on the child node and the α is recomputed. In this particular case the α remains the same since the relaxed optimum at least as large as the previous α. Since the current α (71) is still potentially attainable, the algorithm continues expanding the left child. On the next level, vertex 3 is placed left. As it's optimum bursts the current α the node is marked as closed until the α is lowered and its sibling is expanded. Since it also bursts the current α, the right child is also marked as closed and the algorithm looks for the next open node which may still present a solution for the current α. At this point, for this example, it backtracks to the root and expands to the right placing vertex 2 on the same partition as 1. Once again the α is burst and must be lowered to the next integer (69). The algorithm continues from the node its on. It first goes left but once again bursts the target then turns right. The same occurs on the next level and again on the next where an integer solution is found for the current alpha. Notice how both siblings present integer solutions to the problem. However,

as the current α is at 69, the solution for 68 could not be accepted until all open nodes had less than 69 as relaxed optimum.

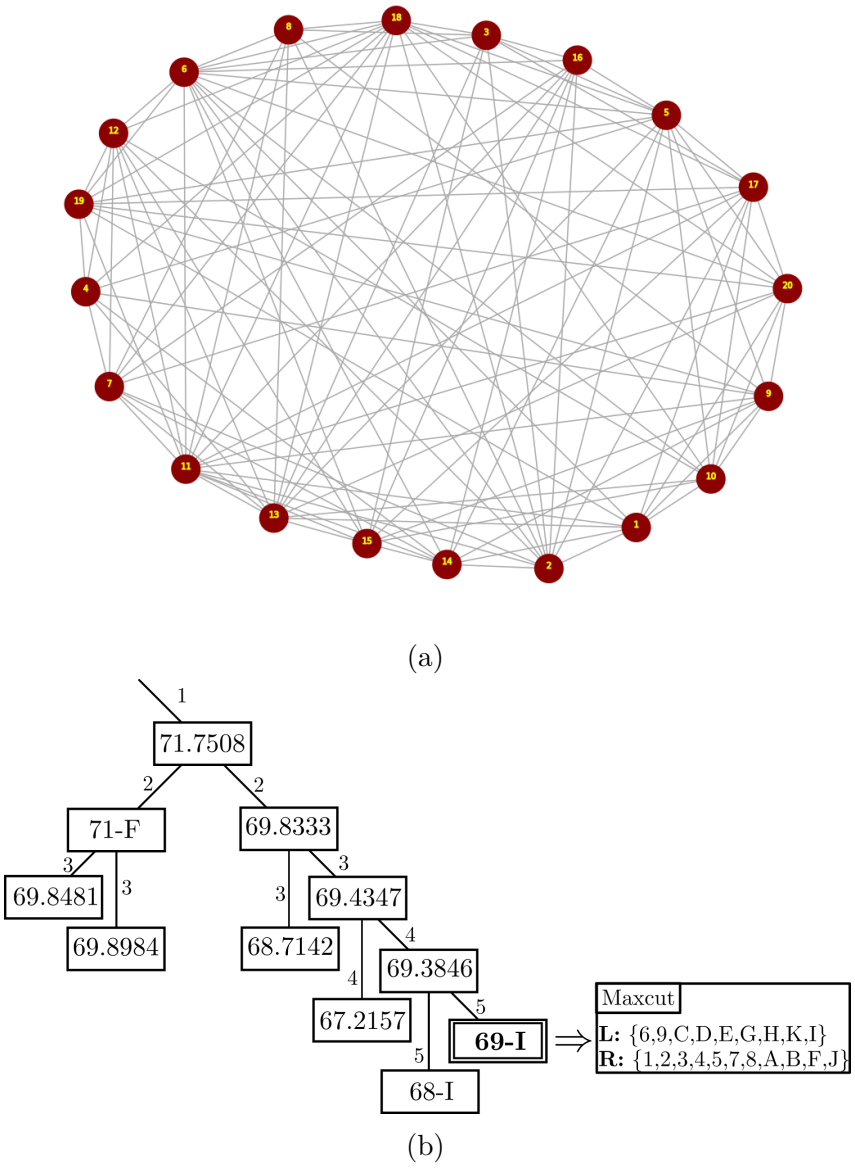


Figure 14 – A compact tree with a simple Max Cut and proof of optimality for a random graph with 20 vertices, *integrallity gap 2.705 and index 125 (a). The number in each node represents the value for that cut. Some cuts have integer value while having at least one fractional variable (indicated by F) while others are completely integer (I). At each level, more variables on the original problem are set by positioning new nodes of the original subgraph on either left or right partitions depending on the path.*

4.2 PROOFING TREES

While the compact trees presented in section 4.1 provide proof of optimality in a single structure, they can grow considerably if the relaxed optimum is distant from the Max Cut. This is especially true for both P_{12} and the triangular polytope for large enough problems. The concept of proofing trees was extended to represent the solution with one or more trees in order to improve the efficiency of the previous algorithm.

Let O_r be the relaxed optimum for a given subgraph and O_m be the value of its corresponding Max Cut. Then, $\delta = \lfloor O_r \rfloor - O_m$ and $O_r \leq \alpha \leq O_m$. The new algorithm builds δ trees, one for each possible value of α with sufficient proof that either α is the value of the Max Cut or there is no possible cut for that value. At first the triangular polytope was used to decrease the number of nodes in the tree. However, since the triangular polytope distances from the integer hull as the problem size increases, a more viable strategy is to start with the most compact formulation and introduce new violating restrictions along the way. Since clique-0 contains the complete graph, it inherently sets the value for the Max Cut to be lesser or equal than the Max Cut for the complete graph (intuitively the largest cut for its size), significantly lowering the gap δ. This approach uses P_{12} for more efficiency since solving it is much cheaper even though more LPs need to be solved for each tree.

Two important integer parameters are associated with each tree. The first is the value of α described above. The second is the *fundamental width* Ω. The fundamental width is calculated by pruning all leafs and counting the resulting leafs. This gives a sense of how many different paths the algorithm had to follow before solving the problem. Our algorithms are polynomial in Ω so it is paramount to bound it. We define $\Omega^\star$ as $\max \Omega$ and $\alpha^\star$ as $\min \alpha$.

The algorithm begins similarly by placing an implicit node 0 on the tree and a node 1 as right child, indicating vertex 1 is on the right partition. We associate a set of constraints and an objective function to each vertex of the tree. We solve an LP for each such vertex, proceeding in a branch and bound manner. By using the so called triangular polytope (our most expensive yet polynomial model), it is possible to solve and verify Max Cut problems for random graphs up to 34 vertices in a reasonable amount of time. For larger problems the computational experiments begin to become unwieldy for the available hardware.

Proposition 12. *All signed binomial inequalities imply the binomial equalities.*

Proof. Assume all signed binomial inequalities involving vertices i, j, k of kz, namely

$$\begin{aligned} x_{ij} &+ x_{jk} &+ x_{ki} &\leq 2 \\ x_{ij} &- x_{jk} &- x_{ki} &\leq 0 \\ -x_{ij} &+ x_{jk} &- x_{ki} &\leq 0 \\ -x_{ij} &- x_{jk} &+ x_{ki} &\leq 0 \end{aligned}$$

$$\begin{aligned} x_{ij} = 1 \;\Rightarrow\; & x_{jk} + x_{ki} \leq 1 \;\equiv\; x_{jk} + x_{ki} \leq 1 \\ & -x_{jk} - x_{ki} \leq 1 \;\Rightarrow\; x_{jk} + x_{ij} \geq 1 \;\Rightarrow\; x_{ki} + x_{jk} = 1 \end{aligned}$$

$$\begin{aligned} x_{ij} = 0 \;\Rightarrow\; & x_{jk} - x_{ki} \leq 0 \;\equiv\; x_{jk} - x_{ki} \leq 0 \\ & -x_{jk} + x_{ki} \leq 0 \;\Rightarrow\; x_{jk} - x_{ij} \geq 0 \;\Rightarrow\; x_{ki} - x_{jk} = 0 \end{aligned}$$

□

For every node on the tree we obtain the corresponding bipartition induced by the path from the root to it. Every edge crossing from one partition to the other has a positive variable on the current objective function while edges linking vertices on the same partition has negative variables. This is done for the complete graph for the positioned vertices up to that point. Similarly, binomials can be added for each edge as well. These are introduced as new constraints on the LP for that node. The algorithm considers successful if the LP returns an optimum exactly the number of positive variables on the objective function.

The intuition of a proof tree is very simple: each tree will correspond to a possible value of maximum cut. The tree will either present the cut or prove why there is no possible cut with that value. Arbitrarily, the first node is always placed on the right partition. If a node appears as a left child, then that node is on the opposite partition. A right child denotes otherwise. If a tree does not contain a maximum cut, then every possible partition configuration is contained within the tree, along with the point where it becomes unfeasible.

4.3 PROBLEM MODELING

Models P_{01} and the triangular polytope were prototyped in the Wolfram programming language and implemented in Mathematica (Wolfram Research, Inc., 2016). The prototype included the drawing and generation of random graphs with parameterized density and Johnson family graphs, text output for both models in CPLEX LP format, external system calls to solve the models as either LP or ILP using the SCIP optimization suite (GAMRATH et al., 2016), automatic routines to run batch tests with random graphs, functions to generate a cutting plane and append to the current problem and functions to generate equivalent CNF clauses in order to verify the possibility of solving the problem as a SAT.

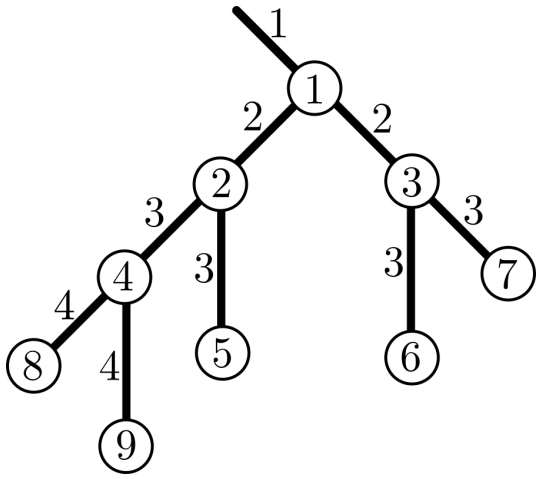


Figure 15 – Example of a naive binary tree. Every node has an unique number which completely identifies a path from the root to it. Each edge contains a label identifying a vertex in the original subgraph while the direction of the edge implies a placement on the cut's partitions (left child implies placement in left partition and right child, right partition). In this particular example, the node number 9 implies the partitions $\{1, 4\}$ and $\{2, 3\}$.

To every vertex in the subgraph was attributed a base-36 element varying from 1 to Z (A being 10). In this manner, every edge was modeled as a variable with the string prefix xe to identify it as a primal structured variable. Implicit to the models, all variables must be positive, however, P_{01} has no implicit upper bound $xe_{ij} \leq 1$ and thus must be explicitly set when defining the inputs for the solver. This is not necessary for the triangular polytope.

In order to take advantage of SCIP's open source SOPLEX callable library written in the C programming language, the whole project was ported to C++. So far, the ported project is capable of generating random graphs with defined density, generate the models in SCIP's internal data structure, export the model as LP file, change the model's description by adding, removing or modifying any constraint or objective set, solve the model as an LP, solve the Max Cut problem using the algorithms described in this chapter providing sufficient proof of it. The program can be run as a command line tool which allows for the user to input a file with the problem to be solved, redirect the output to any file desired, give an initial value to the Max Cut (if desired), input a subgraph directly via the console, generate any number of random graphs to be solved, choose between polytopes P_{01} or the triangular polytope, set the desired number of edges or density of the random subgraph and count the number of random subgraphs at which the fundamental width of the trees is zero (Table 5). These results clearly indicate the lack of efficiency of the triangular polytope to solve the relaxed problem as the problem size increases.

	P_{12}		$P_{34}^{\overline{01}}$	
z	#	%	#	%
10	99	(00.99)	9878	(98.78)
12	13	(00.13)	9565	(95.65)
14	1	(00.11)	9550	(95.50)
16	1	(00.01)	8997	(89.97)
18	0	(00.00)	8468	(84.68)
20	0	(00.00)	7761	(77.61)
22	0	(00.00)	7048	(70.48)
24	0	(00.00)	6390	(63.90)

Table 5 – Amount of trees with $\Omega = 0$ in 10^4 experiments with random graphs. The same problems were run for both Polytope P_{12} and the triangular polytope for Kz with even $z \in 10...24$.

4.4 NATURAL TREE ALGORITHM

This approach builds the set of proofing trees in a naive manner, by assigning vertices to partitions in lexicographical order. As seen in section 4.2, the algorithms will build one tree for each δ starting with $\alpha = \lfloor O_r \rfloor$. The algorithm assumes α is the correct value for the Max Cut and creates a new constraint for that. Each node in the tree is expanded first left and the algorithm keeps going until it finds a cut or it fails to solve the relaxed problem for a particular node. Every time the algorithm exhausts every possible partition configuration, the α is lowered by one and a new tree is build.

Consider the well-known Petersen Graph present in Appendix 5 whose relaxed optimum for P_{12} is less than 15, setting the starting α to 14. As can be seen in Figure 16, the algorithm will build T_{14}^N. It will place 1 in the right partition and attempt to position 2 across than beside 1.

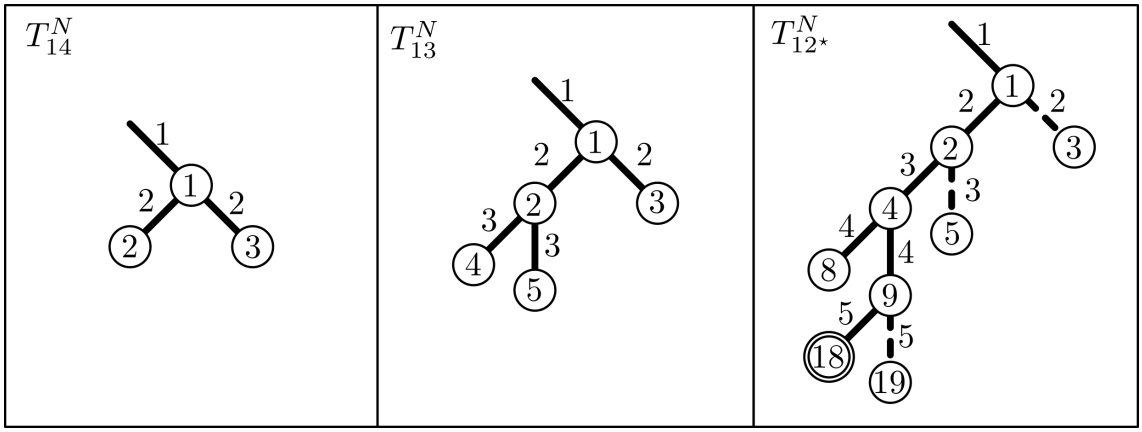


Figure 16 – Example of proof for the Max Cut of the Petersen Graph. The algorithm build tree proofing trees for $\alpha = 14$, $\alpha = 13$ and $\alpha = 12$. Every leaf either indicates that there is no feasible solution with integer variables on that node or that a solution was found on that node (represented by concentric circles.

In this particular case both fails indicating that there is no feasible solution for

$xe_{12} = 1$ or $xe_{12} = 0$. Since a valid cut requires all variables to be integer we conclude that there is no cut for $\alpha = 14$. The next step is to build T_{13}^N. Once again the algorithm places 1 on the right partition and attempts to place 2 across. Empirical evidences suggest that as α tends to $\alpha^\star$, there are more feasible solutions that comprehends the partial placement of variables. In this example the algorithm is successful in placing 2 across, however, there are no feasible solutions that place 3 on either partition given that 2 is placed across from 1. This forces the algorithm to give up on this branch and backtrack to place 2 beside 1. As this fails, the algorithm concludes there is no possible solution for $\alpha = 13$ and once again lowers the α to build T_{12}^N.

The algorithm begins once again placing 1 on the right partition, is able to position both 2 and 3 on the left, fails to place 4 left but is able to place it right. When placing 5 on the left, the algorithm is able to not only find a feasible solution to it but also guess the placement of all other vertices of the subgraph on either partition by handing an integer solution to the LP. We emphasize that the actual number of correctly placed variables that yields a complete solution may vary from solver to solver, but since our emphasis is on verifiability, the returned partition is sufficient proof of optimality since it will result into a feasible solution with value equal to $\alpha^\star$ if inputed in any solver with values for all variables set (or equivalently transformed into a SAT).

4.5 ADAPTIVE TREE ALGORITHM

For each value of α, the algorithm will create a new constraint, demanding that the objective function equals the current α. It will, then, create the tree, having the first node as root. The algorithm will sprout the second node as children of the first node, using a depth-first approach. The main reasoning behind the depth-first in lieu of a width-first approach is that a deeper level on the binary tree implies in more set variables and a greater chance of determining either success or failure.

The algorithm then selects the first vertex that has not been positioned at the bipartition and attempts to build a tree with it on the child node. It then builds a corresponding LP and attempts to solve it. If the LP yields an integer solution then its done. If not then either the problem will have a fractional solution or no solution at all (infeasible). At this point this strategy is a clear improvement since it will be known that if any viable partition with a corresponding cut of value α exists, it will surely not have that partial placement, therefore we can discard all further attempts down the tree (node i will never be on the same side as j for instance).

The major advance in using the Adaptive Tree instead of the Natural Tree is that the problem is polynomial on the number of nodes of the tree. Since the Adaptive Tree is able to prematurely decide some unfeasible cut configurations (i.e: i cannot belong to the same side of the bipartition as j) the adaptive tree is capable of significantly reducing the

complexity of the problem. The clear down side is that it must run multiple LPs on the current node until the problem is solved, an incompatible vertex was found to correspond to that node or all vertices who where not positioned yet at the bipartition were tested.

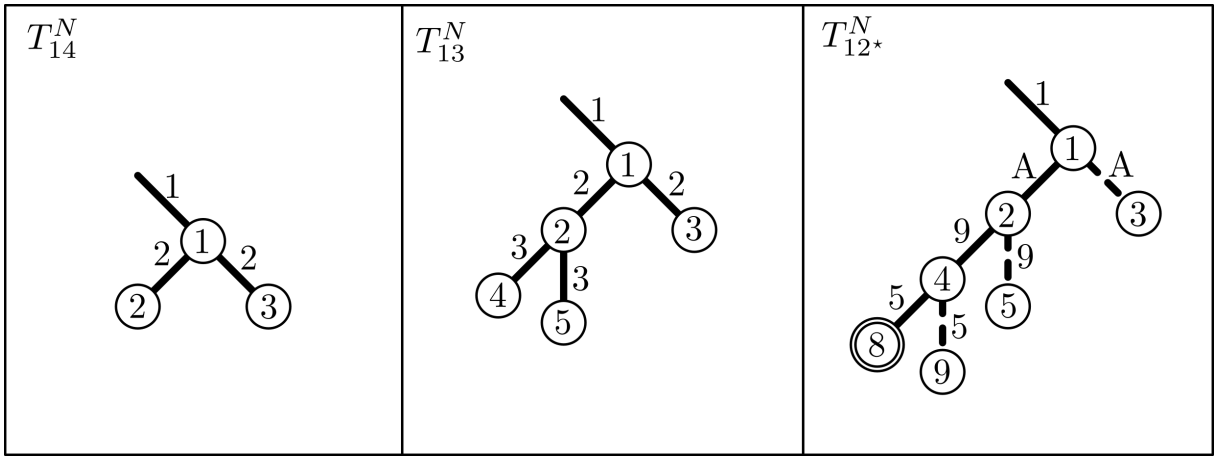


Figure 17 – Adaptive Tree for the Petersen graph. In this particular example, the Adaptive Tree had one less depth level than the Natural Tree. The node where the problem was solved has distance 4 to the root where the Natural Tree had distance 5.

An example for the Adaptive Tree computed for the Petersen Graph (Figure 17) details how it is different from the Natural Tree. Unlike the Natural Tree which orderly placed vertices in lexicographical order as the tree branched out, this algorithm placed the lexicographically last vertices on the first two attempts (excluding the canonical 1 at the right side of the partition) because no vertex could be discarded at this point. When this happens the last tested vertex is placed at the node. Deeper at the tree, the algorithm decided that the vertex 5 could only be placed at the left side of the partition. Coincidentally this yielded an integer solution for the relaxed problem.

Our computational experiments demonstrated that, on average, very dense or very sparse subgraphs will have few possible partition configurations with cuts of the same cardinality of the Max Cut for that problem. In practice this means the Adaptive Tree will fail very often (effectively shrinking the search space) and quickly find the value of the Max Cut. The exact same opposite happens with very symmetrical graphs, where many solutions exists and the tree will succeed in most attempts.

5 CONCLUSION

This work investigated cut polyhedra related to the Max Cut problem. The revised literature presented different formulations for the Max Cut, some extensive others more compact. Contained within the current literature, the smallest formulation prior to this work had $O(|V| \cdot |E|)$ constraints while this work presents new formulations containing at most exactly $11|E|$ constraints for the even case (in the number of vertices) or less for the odd case.

This work also presented a theorem demonstrating how an integer solution for the relaxed problem can be obtained from a fractional dual optimum with the same objective value given that certain conditions are met. Given that both solutions must have the same objective value the dual solution must be of integer optimum objective value. This is also significant to determine when there is no cut of that cardinality. Since this is done by a 2-SAT, all possible integer solutions can be obtained done efficiently.

On the last part we performed computational experiments to evaluate the relaxed problem for different classes of graphs as well as random subgraphs. We found that either sparse or very dense subgraphs were more likely to present integer solutions for the relaxed problem and that symmetric graphs were significantly harder to solve computationally by solvers.

REFERENCES

BARAHONA, F. The max-cut problem on graphs not contractible to k5. *Operations Research Letters*, Elsevier, v. 2, n. 3, p. 107–111, 1983. Cited on page 15.

BARAHONA, F.; GRÖTSCHEL, M.; MAHJOUB, A. R. Facets of the bipartite subgraph polytope. *Mathematics of Operations Research*, INFORMS, v. 10, n. 2, p. 340–358, 1985. Cited on page 15.

BARAHONA, F.; JÜNGER, M.; REINELT, G. Experiments in quadratic 0–1 programming. *Mathematical Programming*, Springer, v. 44, n. 1-3, p. 127–137, 1989. Cited 3 times on pages 13, 16, and 37.

BARAHONA, F.; MAHJOUB, A. R. On the cut polytope. *Mathematical programming*, Springer, v. 36, n. 2, p. 157–173, 1986. Cited on page 15.

CORMEN, T. H. *Introduction to algorithms*. [S.l.]: MIT press, 2009. Cited on page 38.

GALLI, L.; KAPARIS, K.; LETCHFORD, A. N. Complexity results for the gap inequalities for the max-cut problem. *Operations Research Letters*, Elsevier, v. 40, n. 3, p. 149–152, 2012. Cited on page 17.

GAMRATH, G. et al. *The SCIP Optimization Suite 3.2*. Takustr.7, 14195 Berlin, 2016. Cited on page 47.

GAREY, M.; JOHNSON, D. *Computers and intractability*. [S.l.]: Freeman San Francisco, 1979. Cited on page 13.

GOEMANS, M.; WILLIAMSON, D. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *Journal of the Association for Computing Machinery*, [New York] Association for Computing Machinery., 1995. Cited on page 13.

GOMORY, R. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, v. 64, p. 275–278, 1958. Cited on page 43.

GRÖTSCHEL, M.; NEMHAUSER, G. L. A polynomial algorithm for the max-cut problem on graphs without long odd cycles. *Mathematical Programming*, Springer, v. 29, n. 1, p. 28–40, 1984. Cited on page 15.

HADLOCK, F. Finding a maximum cut of a planar graph in polynomial time. *SIAM Journal on Computing*, SIAM, v. 4, n. 3, p. 221–225, 1975. Cited on page 15.

KELLY, J. B. Hypermetric spaces. In: *The geometry of metric and linear spaces*. [S.l.]: Springer, 1975. p. 17–31. Cited on page 17.

KRISHNAN, K.; MITCHELL, J. E. A semidefinite programming based polyhedral cut and price approach for the maxcut problem. *Computational Optimization and Applications*, Springer, v. 33, n. 1, p. 51–71, 2006. Cited on page 16.

LANCIA, G.; SERAFINI, P. An effective compact formulation of the max cut problem on sparse graphs. *Electronic Notes in Discrete Mathematics*, Elsevier, v. 37, p. 111–116, 2011. Cited 3 times on pages 13, 16, and 37.

LAURENT, M.; POLJAK, S. Gap inequalities for the cut polytope. *European Journal of Combinatorics*, Elsevier, v. 17, n. 2, p. 233–254, 1996. Cited 2 times on pages 17 and 38.

LINS, S. *Graphs of maps, Available in the arXiv as math.CO/0305058*. Thesis (Doctorate) — University of Waterloo, 1980. Cited 3 times on pages 18, 20, and 21.

LINS, S. Graph encoded maps. *Journal of Combinatorial Theory, Series B*, v. 32, p. 171–181, 1982. Cited on page 20.

NGUYEN, V. H.; MINOUX, M.; NGUYEN, D. P. Improved compact formulations for metric and cut polyhedra. *Electronic Notes in Discrete Mathematics*, Elsevier, v. 52, p. 125–132, 2016. Cited 2 times on pages 13 and 16.

POLJAK, S.; TUZA, Z. Maximum cuts and large bipartite subgraphs. *DIMACS Series*, v. 20, p. 181–244, 1995. Cited on page 15.

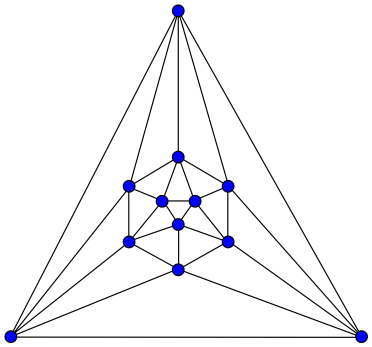
SCHOENBERG, I. J. Metric spaces and positive definite functions. *Transactions of the American Mathematical Society*, JSTOR, v. 44, n. 3, p. 522–536, 1938. Cited on page 17.

SIMONE, C. D.; RINALDI, G. A cutting plane algorithm for the max-cut problem. *Optimization Methods and Software*, Taylor & Francis, v. 3, n. 1-3, p. 195–214, 1994. Cited on page 16.

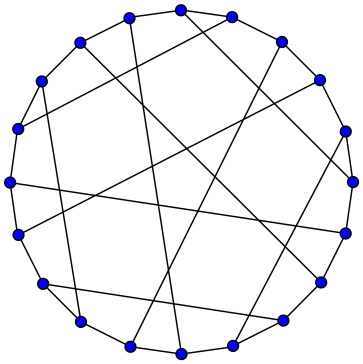
TYLKIN, M. Hamming geometry of unit cubes. *Doklady Akademii Nauk SSSR*, MEZHDUNARODNAYA KNIGA 39 DIMITROVA UL., 113095 MOSCOW, RUSSIA, v. 134, n. 5, p. 1037–1040, 1960. Cited on page 17.

Wolfram Research, Inc. *Mathematica*. Version 10.4. Champaign, Illinois, 2016. Cited on page 47.

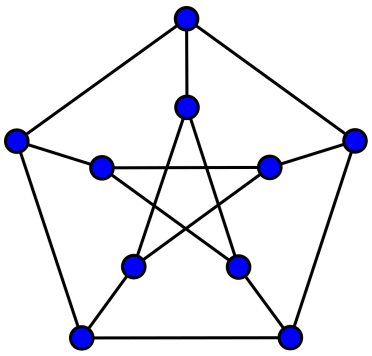
APPENDIX A - GRAPH CATALOG



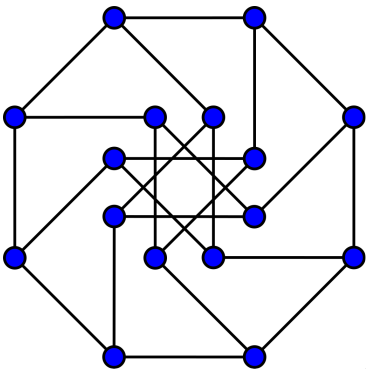
(a) Icosahedral Graph: Hamiltonian, planar. 12 vertices and 30 edges.



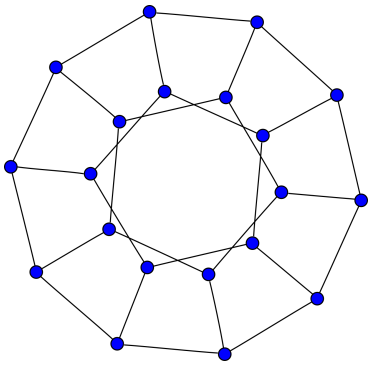
(b) Desargues Graph: Non-planar cubic partial cube. 20 vertices and 30 edges.



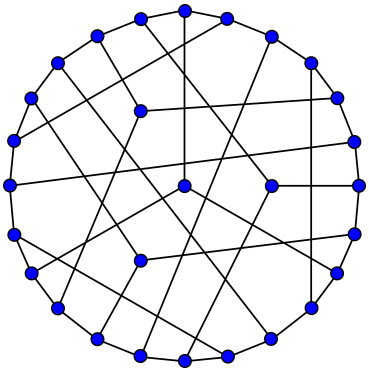
(c) Petersen Graph: Bridgeless cubic. 10 vertices and 15 edges.



(d) Moebius-Kantor Graph: Bipartite. 16 vertices and 24 edges.

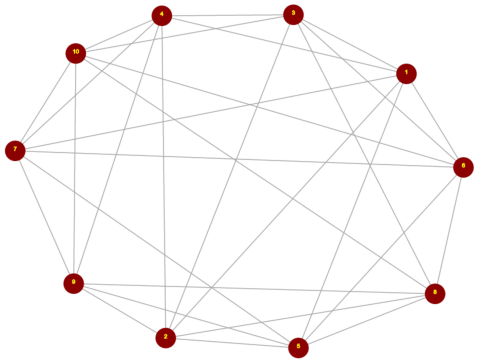


(e) Dodecahedral Graph: Hamiltonian. 20 vertices and 30 edges.

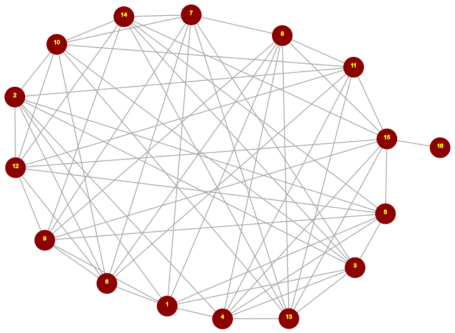


(f) Coxeter Graph: 3-regular. 28 vertices and 42 edges.

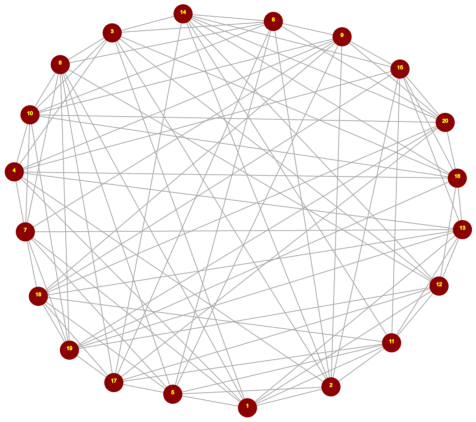
Figure 18 – Some interesting symmetrical graphs



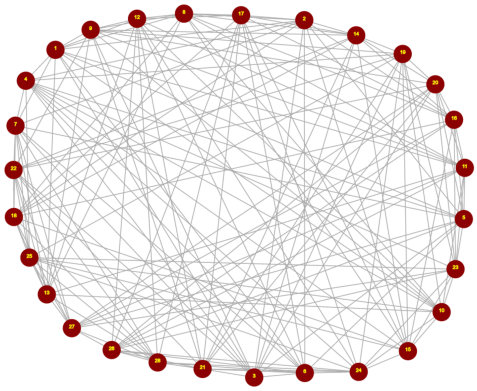
(a) Johnson (5,2)



(b) Johnson (6,2)



(c) Johnson (6,3)



(d) Johnson (8,2)

Figure 19 – Graphs from the Johnson family. J(8,2) is so far the graph that took the longest to solve (approximately 49 hours as an ILP via SCIP).

APPENDIX B - CONSTRAINTS OF A LINEAR FORMULATION FOR THE MAXCUT

$$
\begin{aligned}
y_{123n0} &: x_{12} + x_{13} + x_{23} && \leq 2 \\
y_{123p1} &: x_{12} - x_{13} - x_{23} && \leq 0 \\
y_{123p2} &: -x_{12} - x_{13} + x_{23} && \leq 0 \\
y_{123p3} &: -x_{12} + x_{13} - x_{23} && \leq 0 \\
y_{124n0} &: x_{12} + x_{14} + x_{24} && \leq 2 \\
y_{124p1} &: x_{12} - x_{14} - x_{24} && \leq 0 \\
y_{124p2} &: -x_{12} - x_{14} + x_{24} && \leq 0 \\
y_{124p3} &: -x_{12} + x_{14} - x_{24} && \leq 0 \\
y_{125n0} &: x_{12} + x_{15} + x_{25} && \leq 2 \\
y_{125p1} &: x_{12} - x_{15} - x_{25} && \leq 0 \\
y_{125p2} &: -x_{12} - x_{15} + x_{25} && \leq 0 \\
y_{125p3} &: -x_{12} + x_{15} - x_{25} && \leq 0 \\
y_{134n0} &: x_{13} + x_{14} + x_{34} && \leq 2 \\
y_{134p1} &: x_{13} - x_{14} - x_{34} && \leq 0 \\
y_{134p2} &: -x_{13} - x_{14} + x_{34} && \leq 0 \\
y_{134p3} &: -x_{13} + x_{14} - x_{34} && \leq 0 \\
y_{145n0} &: x_{14} + x_{15} + x_{45} && \leq 2 \\
y_{145p1} &: x_{14} - x_{15} - x_{45} && \leq 0 \\
y_{145p2} &: -x_{14} - x_{15} + x_{45} && \leq 0 \\
y_{145p3} &: -x_{14} + x_{15} - x_{45} && \leq 0
\end{aligned}
$$

$$
\begin{aligned}
y_{235n0} &: x_{23} + x_{25} + x_{35} && \leq 2 \\
y_{235p1} &: x_{23} - x_{25} - x_{35} && \leq 0 \\
y_{235p2} &: -x_{23} - x_{25} + x_{35} && \leq 0 \\
y_{235p3} &: -x_{23} + x_{25} - x_{35} && \leq 0 \\
y_{245n0} &: x_{24} + x_{25} + x_{45} && \leq 2 \\
y_{245p1} &: x_{24} - x_{25} - x_{45} && \leq 0 \\
y_{245p2} &: -x_{24} - x_{25} + x_{45} && \leq 0 \\
y_{245p3} &: -x_{24} + x_{25} - x_{45} && \leq 0 \\
y_{345n0} &: x_{34} + x_{35} + x_{45} && \leq 2 \\
y_{345p1} &: x_{34} - x_{35} - x_{45} && \leq 0 \\
y_{345p2} &: -x_{34} - x_{35} + x_{45} && \leq 0 \\
y_{345p3} &: -x_{34} + x_{35} - x_{45} && \leq 0
\end{aligned}
$$