



From graph orientation to the unweighted maximum cut

Walid Ben-Ameur, Antoine Glorieux, José Neto

► To cite this version:

Walid Ben-Ameur, Antoine Glorieux, José Neto. From graph orientation to the unweighted maximum cut. COCOON 2016: 22nd International Conference on Computing and Combinatorics, Aug 2016, Ho Chi Minh Ville, Vietnam. pp.370-384, 10.1007/978-3-319-42634-1_30 . hal-01497887

HAL Id: hal-01497887

<https://hal.science/hal-01497887>

Submitted on 29 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

From graph orientation to unweighted maximum cut

Walid Ben-Ameur, Antoine Glorieux, and José Neto

Samovar UMR 5157, Télécom SudParis, CNRS, Université Paris-Saclay
9 Rue Charles Fourier, 91011 Evry Cedex, France
`walid.benameur@telecom-sudparis.eu, antoine.glorieux@telecom-sudparis.eu,`
`jose.neto@telecom-sudparis.eu`

Abstract. In this paper, starting from graph orientation problems, we introduce some new mixed integer linear programming formulations for the unweighted maximum cut problem. Then a new semidefinite relaxation is proposed and shown to be tighter than the Goemans and Williamson’s semidefinite relaxation. Preliminary computational results are also reported.

1 Introduction

Let $G = (V, E)$ be a simple undirected graph, with node set V , edge set E , and let $(w_e)_{e \in E}$ denote nonnegative edge-weights. Given a node subset $S \subseteq V$, the *cut* defined by S , denoted by $\delta(S)$, is the subset of edges in E having exactly one endpoint in S , i.e. $\delta(S) = \{ij \in E : |S \cap \{i, j\}| = 1\}$. The *weight of the cut* defined by S , denoted by $w(\delta(S))$ is the sum of the weights of the edges belonging to the cut, i.e., $w(\delta(S)) = \sum_{e \in \delta(S)} w_e$. The maximum cut problem consists in finding a cut of maximum weight, denoted by w^* , in the graph G : $\max\{w(\delta(S)) : S \subset V\}$. The cardinality of V will be denoted by n .

The maximum cut problem is a fundamental combinatorial optimization problem that emerges in several scientific disciplines: VLSI design [4], sparse matrix computation [2], parallel programming [9], statistical physics [4], quadratic programming [17], etc. A less known application is given by frequency assignment in networks where the weight of each edge represents the interference level between two nodes. Assuming that only two frequencies (resources) are available, assigning a frequency to each node such that the whole interference between nodes using the same frequency is minimized is a maximum cut problem.

The maximum cut problem is known to be NP-hard [20] in general, and not approximable within a ratio $\frac{16}{17} + \epsilon$ for any $\epsilon > 0$ unless $P = NP$ [18]. However, the problem may be polynomial for some instances. We know, for example, that the problem becomes easy when the underlying graph is weakly bipartite and the weights are nonnegative [16]. Other polynomial cases are reviewed, e.g., in [5].

One line of research to solve this problem has consisted in the development of (meta)heuristics, see, e.g. [6, 12, 27]. Another important line of research relies on linear programming formulations of the problem. This has namely led to deep investigations on the polyhedral structure of the cut polytope: the convex hull of the incidence vectors of all the cuts of the graph, which has then been extensively

used, e.g., in Branch and Cut algorithms [3,5,10]. More recently, essentially since the mid-1990's and the breakthrough paper by Goemans and Williamson [15], there has been a growing interest in semidefinite programming based algorithms. Goemans and Williamson's work presents a 0.87856-approximation algorithm for the maximum cut problem when the edge weights are nonnegative. Their method relies on the following semidefinite relaxation of the problem for a complete graph

$$(\text{SDP0}) \begin{cases} \max \frac{1}{2} \sum_{i=1}^n \sum_{j=i+1}^n w_{ij}(1 - y_{ij}) \\ \text{s.t.} \\ y_{ii} = 1, \forall i \in \llbracket 1, n \rrbracket, \\ Y \succeq 0, \\ Y \in \mathbb{S}^n, \end{cases}$$

where Y represents the matrix with entry y_{ij} in the i th row and j th column, $Y \succeq 0$ is the constraint that the matrix Y is positive semidefinite, and, for any given integer n , $\mathbb{S}^n$ denotes the set of symmetric matrices with order n .

To improve the quality of the bound Z_{SDP0}^* given by the formulation (SDP0) [15], different approaches have been proposed in the literature: namely by making use of polyhedral knowledge on the cut polytope and adding linear inequalities [13,19], or by means of lift-and-project techniques [1,22]. Another way to improve the upper bound given by the semidefinite relaxation is described in [7,8] where some spectral techniques are used leading to polynomial-time algorithms for some low rank weight matrices. This semidefinite approach of the problem also led to efficient solvers such as BiqMac [25] and BiqCrunch [21].

The reader can find in [5,10] and the references therein further results about the maximum cut problem including applications, polynomial cases, approximation algorithms, relationships with other combinatorial problems, polyhedral studies etc.

The paper is organized as follows. We introduce some new mixed integer linear programming formulations for the maximum cut problem based on graph orientations (Section 2). A semidefinite programming relaxation is then proposed. We show that the bound provided by this new SDP relaxation is stronger than the bound given by the relaxation (SDP0) introduced by Goemans and Williamson (Section 3). We also prove that the new bound is tight for complete graphs. We then introduce further Mixed Integer Programming formulations (Section 4). Several numerical experiments have been conducted showing the relevance of the SDP formulation and the performances of the new Mixed Integer Programming formulations (Section 6). A conclusion follows.

2 Mixed integer linear programming formulations

In this section, we gradually introduce our new formulation for the maximum cardinality cut problem, i.e. the maximum cut problem for the case when all the edge-weights are equal to 1: $w_e = 1, \forall e \in E$. For our purposes, we shall think of the original graph $G = (V, E)$ as directed (consider any arbitrary orientation) and let $B \in \{-1, 0, 1\}^{|V| \times |E|}$ stand for its incidence matrix, i.e., the column corresponding to the arc uv (or, equivalently, to the edge uv directed from node u to node v), has only nonzero entries in the rows corresponding to the nodes u and v : $B_{u,uv} = 1$ and $B_{v,uv} = -1$, respectively.

For clarity, we start introducing an auxiliary formulation. It involves two types of variables. The first type of variables $x \in \{-1, 1\}^{|E|}$ describes an orientation of the graph G and may be interpreted as follows. For each edge $ij \in E$ which is originally directed from node i to node j : if $x_{ij} = 1$ then ij is directed from i to j (i.e., the orientation is the same as the original one) and is directed from j to i otherwise (i.e., the edge is “reversed” with respect to the original orientation). The other variables are binary and denoted y_k^v , with $v \in V$ and $k \in \llbracket -d_v, d_v \rrbracket$, where d_v denotes the degree of the node v in G . They have the following interpretation: $y_k^v = 1$ if and only if $B_v x = k$, where B_v denotes the row of B corresponding to vertex v , so that the following equation trivially holds

$$\sum_{k=-d_v}^{d_v} k y_k^v = B_v x, \forall v \in V. \quad (1)$$

Notice that $B_v x = d_v^{(x,+)} - d_v^{(x,-)}$, where $d_v^{(x,+)}$ (resp. $d_v^{(x,-)}$) denotes the outdegree (resp. indegree) of the node v w.r.t. the orientation described by x .

Also, given the interpretation for the variables y , among those of the form y_k^v , for some fixed node $v \in V$, exactly one of them has value 1. Thus, the following constraints are satisfied

$$\sum_{k=-d_v}^{d_v} y_k^v = 1, \forall v \in V. \quad (2)$$

Then we can show the maximum cardinality cut problem may be formulated as the mixed-integer program

$$(\text{MIP1}) \begin{cases} Z_{\text{MIP1}}^* = \frac{1}{2} \max \sum_{v \in V} \sum_{k=-d_v}^{d_v} |k| y_k^v \\ \text{s.t. (1), (2),} \\ x \in [-1; 1]^{|E|}, \\ y_k^v \in \{0, 1\}, \forall v \in V, \forall k \in \llbracket -d_v, d_v \rrbracket. \end{cases}$$

Proposition 1. *The optimal objective value of (MIP1) equals the maximum cardinality of a cut in the graph G : $Z_{\text{MIP1}}^* = w^*$.*

Proof. First, note that it is equivalent to take $x \in [-1; 1]^{|E|}$ or $x \in \{-1, 1\}^{|E|}$ since B is totally unimodular, the integrality of y implies the integrality of x . Observe that introducing sign variables $z \in \{-1, 1\}^{|V|}$ with the following interpretation: z_v is the sign of the only k for which $y_k^v \neq 0$, (MIP1) is equivalent to the following

$$\begin{aligned} & \begin{cases} \max \frac{1}{2} \sum_{v \in V} z_v \sum_{k=-d_v}^{d_v} k y_k^v \\ \text{s.t. (1), (2),} \\ y_k^v \in \{0, 1\}, \forall v \in V, \forall k \in \llbracket -d_v, d_v \rrbracket, \\ z \in \{-1, 1\}^{|V|}, x \in [-1; 1]^{|E|}. \end{cases} \Leftrightarrow \begin{cases} \max \frac{1}{2} \sum_{v \in V} z_v B_v x = \frac{1}{2} z^t B x \\ \text{s.t.} \\ z \in \{-1, 1\}^{|V|}, x \in \{-1, 0, 1\}^{|E|}. \end{cases} \\ & \Leftrightarrow \begin{cases} \max \frac{1}{2} \sum_{uv \in E} x_{uv} (z_v - z_u) = \frac{1}{2} x^t B^t z \\ \text{s.t.} \\ z \in \{-1, 1\}^{|V|}, x \in \{-1, 1\}^{|E|}. \end{cases} \Leftrightarrow \begin{cases} \max \frac{1}{2} \sum_{uv \in E} |z_v - z_u| \\ \text{s.t.} \\ z \in \{-1, 1\}^{|V|}. \end{cases} \end{aligned}$$

Noting that, for any $z \in \{-1, 1\}^{|V|}$, the quantity $\sum_{uv \in E} |z_v - z_u|$ equals twice the number of edges in $\delta(S)$, with $S = \{v \in V : z_v = 1\}$, the proposition follows. $\square$

Remark 1. Given a cut $\delta(S)$ with maximum cardinality, we can associate to it the following feasible solution of (MIP1): the vector x corresponds to an orientation of all edges in the cut from S to $V \setminus S$, all other edges are not oriented ($x_{uv} = 0, \forall uv \in E \setminus \delta(S)$), and $y_k^v = 1$ if and only if k is equal to the outdegree minus the indegree w.r.t. the orientation given by x (ignoring edges which are not oriented). Then, for any edge $uv \in E$ which has the original orientation from u to v (i.e. the one given by the matrix B), the following equation holds: $x_{uv} = \sum_{k=1}^{d_u} y_k^u - \sum_{k=1}^{d_v} y_k^v$. From the latter we deduce (developing the expression $B_v x$):

$$\sum_{k=-d_v}^{d_v} k y_k^v = B_v x = \sum_{uv \in E} \left(\sum_{k=1}^{d_v} y_k^v - \sum_{k=1}^{d_u} y_k^u \right). \quad (3)$$

Observe also that each vertex v is incident to at least $\lceil \frac{d_v}{2} \rceil$ edges in the maximum cut. This implies that the variables y_k^v with $k \in \{1 - \lceil \frac{d_v}{2} \rceil, \dots, \lceil \frac{d_v}{2} \rceil - 1\}$ may be removed from formulation (MIP1), while Proposition 1 remains valid.

It follows that, in place of (MIP1), we may consider a formulation involving variables of the form y_k^v only: replace equations (1) by (3). For each vertex v , we only consider y_k^v variables for $k \in \llbracket -d_v, -\lceil \frac{d_v}{2} \rceil \rrbracket \cup \llbracket \lceil \frac{d_v}{2} \rceil, d_v \rrbracket$. We therefore denote for all $v \in V$, $I_v^- = \llbracket -d_v, -\lceil \frac{d_v}{2} \rceil \rrbracket$, $I_v^+ = \llbracket \lceil \frac{d_v}{2} \rceil, d_v \rrbracket$ and $I_v = I_v^- \cup I_v^+$.

$$(MIP2) \begin{cases} Z_{MIP2}^* = \frac{1}{2} \max \sum_{v \in V} \sum_{k \in I_v} |k| y_k^v \\ \text{s.t.} \\ \sum_{k \in I_v} y_k^v = 1, \forall v \in V, \\ \sum_{k \in I_v} k y_k^v = \sum_{uv \in E} (\sum_{k \in I_v^+} y_k^v - \sum_{k \in I_u^+} y_k^u), \forall v \in V, \\ y_k^v \in \{0, 1\}, \forall v \in V, \forall k \in I_v. \end{cases}$$

Formulation (MIP2) involves about $O(|E|)$ variables and $O(|V|)$ constraints. It will be studied in a forthcoming paper. We will rather consider a strengthening of the linear relaxation of (MIP2) through reformulation-linearization techniques. The latter is obtained by multiplying constraints of (MIP2) and then linearizing. Some other constraints follow from the afore mentioned interpretation of the variables in (MIP2). So, let Y_{kl}^{uv} , with $(u, v) \in V^2, (k, l) \in I_u \times I_v$, denote a binary variable representing the product $y_k^u y_l^v$. Then, given that the variables y_k^v are binary and satisfy (2) we have $Y_{kl}^{uu} = 0, \forall u \in V, \forall k \neq l$. Considering then the product of the left side of (2) with itself we deduce $\sum_{k \in I_v} Y_{kk}^{vv} = 1$. Using other equations obtained by multiplying variables of the form y_k^v with equations (2) and others obtained from (3), we can deduce from (MIP2) the following exact formulation for the maximum cardinality cut problem.

$$\begin{aligned}
& \left\{ \begin{aligned}
Z_{\text{MIP3}}^* &= \frac{1}{2} \max \sum_{v \in V} \sum_{k \in I_v} |k| Y_{kk}^{vv} \\
&\text{s.t.} \\
&\sum_{k \in I_v} Y_{kk}^{vv} = 1, \forall v \in V, \\
&\sum_{k \in I_v} k Y_{kk}^{vv} = \sum_{uv \in E} \left(\sum_{k \in I_v^+} Y_{kk}^{vv} - \sum_{k \in I_u^+} Y_{kk}^{uu} \right), \forall v \in V, \\
&Y_{kk}^{vv} = \sum_{l \in I_u} Y_{kl}^{vu}, \forall u, v \in V, \forall k \in I_v, \\
&(d_v - k) Y_{kk}^{vv} = \sum_{uv \in E} \sum_{l \in I_u^+} Y_{kl}^{vu}, \forall v \in V, \forall k \in I_v^+, \\
&-k Y_{kk}^{vv} = \sum_{uv \in E} \sum_{l \in I_u^+} Y_{kl}^{vu}, \forall v \in V, \forall k \in I_v^-, \\
&\sum_{l \in I_u} l Y_{kl}^{vu} = \sum_{uw \in E} \left(\sum_{l \in I_u^+} Y_{kl}^{vw} - \sum_{l \in I_u^+} Y_{kl}^{vu} \right), \forall v \neq u \in V, \forall k \in I_v, \\
&Y_{kl}^{vu} = Y_{lk}^{uv}, \forall u, v \in V, \forall (k, l) \in I_v \times I_u, \\
&Y_{kl}^{vu} \in \{0, 1\}, \forall u, v \in V, \forall (k, l) \in I_v \times I_u,
\end{aligned} \right. \quad \begin{aligned} (4a) \\ (4b) \\ (4c) \\ (4d) \\ (4e) \\ (4f) \\ (4g) \\ (4h) \end{aligned}
\end{aligned}$$

Proposition 2. *The optimal objective value of (MIP3) equals the maximum cardinality of a cut in the graph G : $Z_{\text{MIP3}}^* = w^*$. $\square$*

Observe that (MIP3) was derived from (MIP2) using lifting. Other formulations could be obtained using some other well-known lifting techniques such as the one of Lassere, the Sherali-Adams technique or the lifting of Lovász-Schrijver [26,23,28]. Since the aim of this paper is not to compare lifting techniques, we do not elaborate more on this topic.

3 A semidefinite programming bound

Let Y denote a symmetric matrix with rows and columns indexed by all pairs (u, k) with $u \in V$ and $k \in I_u$. The entry in the row indexed by (u, k) and column indexed by (v, l) corresponds to the variable Y_{kl}^{uv} . Then, let (SDP3) denote the relaxation obtained from (MIP3) replacing the symmetry (4g) and binary constraints (4h) by the following ones

$$Y - \text{Diag}(Y) \text{Diag}(Y)^t \succeq 0 \quad (5)$$

$$Y_{kl}^{vu} \geq 0, \forall u, v \in V, \forall k \in I_v^- \cup I_v^+, l \in I_u^- \cup I_u^+, \quad (6)$$

where $\text{Diag}(Y)$ denotes the vector corresponding to the diagonal of Y .

We are going to prove that the semidefinite relaxation (SDP3) provides a generally better upper bound for the maximum cardinality cut problem than that from Goemans & Williamson's relaxation. For, consider the following semidefinite relaxation of a 0/1 formulation of the maximum cardinality cut problem whose optimal objective value coincides with Z_{SDP0}^* ,

$$(\text{SDP4}) \left\{ \begin{aligned}
Z_{\text{SDP4}}^* &= \max \frac{1}{2} \sum_{u \in V} \sum_{v: uv \in E} (x_u + x_v - 2X_{uv}) \\
&\text{s.t.} \\
&\text{Diag}(X) = x, \\
&X - xx^t \succeq 0, \\
&x \in \mathbb{R}^{|V|}, X \in \mathbb{S}^{|V|},
\end{aligned} \right.$$

where X_{uv} stands for the entry of the matrix X in the row corresponding to node u , and column corresponding to node v .

Proposition 3. *The following inequality holds $Z_{\text{SDP3}}^* \leq Z_{\text{SDP4}}^* (= Z_{\text{SDP0}}^*)$.*

Proof. Let Y denote a feasible solution for the formulation (SDP3) and let $(X, x) \in \mathbb{R}^{|V| \times |V|} \times \mathbb{R}^{|V|}$ be defined as follows: $X_{uv} := \sum_{k=-d_u}^{-1} \sum_{l=-d_v}^{-1} Y_{kl}^{uv}$, $\forall u, v \in V$ with $u \neq v$ and $X_{vv} = x_v := \sum_{k=-d_v}^{-1} Y_{kk}^{vv}$, $\forall v \in V$.

We now show $X - xx^t \succeq 0$. For, let $z \in \mathbb{R}^{|V|}$ and define $\bar{z}$ of convenient dimension as follows: $\bar{z}_k^v = z_v$ if $k < 0$ and 0 otherwise. Then, we have

$$\begin{aligned} \bar{z}^t Y \bar{z} &= \sum_{u \in V} \sum_{k=-d_u}^{-1} \sum_{v \in V} \sum_{l=-d_v}^{-1} \bar{z}_k^u \bar{z}_l^v Y_{kl}^{uv} \\ &= \sum_{u \in V} \sum_{v \in V} z_u z_v \left(\sum_{k=-d_u}^{-1} \sum_{l=-d_v}^{-1} Y_{kl}^{uv} \right) \\ &= \sum_{u \in V} \sum_{v \in V} z_u z_v X_{uv} = z^t X z. \end{aligned}$$

Also,

$$\begin{aligned} \bar{z}^t \text{Diag}(Y) &= \sum_{v \in V} \sum_{k=-d_v}^{-1} \bar{z}_k^v Y_{kk}^{vv} \\ &= \sum_{v \in V} (z_v \sum_{k=-d_v}^{-1} Y_{kk}^{vv}) \\ &= \sum_{v \in V} z_v x_v = z^t x. \end{aligned}$$

It follows that $z^t (X - xx^t) z = \bar{z}^t (Y - \text{Diag}(Y) \text{Diag}(Y)^t) \bar{z} \geq 0$, where the last inequality follows from the feasibility of Y w.r.t. (SDP3), and thus $X - xx^t \succeq 0$. So, we have shown (X, x) is a feasible solution for (SDP4).

Since Y is symmetric, the same holds for X . We now show the objective value of (X, x) w.r.t. (SDP4), denoted Z_X equals that of Y w.r.t. (SDP3).

$$\begin{aligned} Z_X &= \frac{1}{2} \left(\sum_{u \in V} \sum_{v: uv \in E} (x_u + x_v - 2X_{uv}) \right) \\ &= \sum_{v \in V} d_v x_v - \sum_{u \in V} \sum_{uv \in E} X_{uv} \\ &= \sum_{v \in V} d_v \sum_{k=-d_v}^{-1} Y_{kk}^{vv} - \sum_{u \in V} \sum_{uv \in E} \sum_{k=-d_u}^{-1} \sum_{l=-d_v}^{-1} Y_{kl}^{uv} \\ &= \sum_{v \in V} d_v \sum_{k=-d_v}^{-1} Y_{kk}^{vv} - \sum_{u \in V} \sum_{k=-d_u}^{-1} (k + d_u) Y_{kk}^{uu} \\ &= - \sum_{u \in V} \sum_{k=-d_u}^{-1} k Y_{kk}^{uu} \\ &= \sum_{u \in V} \sum_{k=1}^{d_u} k Y_{kk}^{uu}. \end{aligned} \quad \square$$

By Proposition 3, it follows that a randomized algorithm similar to the one by Goemans and Williamson [15] but applied to an optimal solution Y of (SDP3) has the same approximation ratio. To be more precise, let $Z \in \mathbb{R}^{|V| \times |V|}$ denote the matrix with entries $Z_{uv} = 4(X_{uv} - x_u x_v) + (2x_u - 1)(2x_v - 1)$, where $X_{uv} := \sum_{k=-d_u}^{-1} \sum_{l=-d_v}^{-1} Y_{kl}^{uv}$, $\forall u, v \in V$ with $u \neq v$ and $X_{vv} = x_v := \sum_{k=-d_v}^{-1} Y_{kk}^{vv}$, $\forall v \in V$. It can be checked that Z is a feasible solution for the formulation (SDP0) with the same objective value as Y . Then, let $H \in \mathbb{R}^{m \times |V|}$ (for some $m \leq |V|$) denote a matrix such that $Z = H^t H$ and let r denote a vector which is randomly generated according to a uniform distribution on the unit sphere in $\mathbb{R}^m$. The cut returned by the algorithm is then $\delta(S)$ with $S := \{v \in V : r^t h_v \geq 0\}$, where h_v stands for the column of H corresponding to node $v \in V$.

The new bound is exact (i.e., equal to the maximum cardinality of a cut) for some graph classes. For space limitation reasons, we will only consider the case of complete graphs. We already know that the bound provided by relaxation

(SDP0) is exact for even complete graphs (K_n with n even). This does not hold for odd complete graphs. We prove that the bound given by (SDP3) is exact for all complete graphs.

Proposition 4. *For a complete graph, the optimal objective value of (SDP3) is exact: $Z_{\text{SDP3}}^* = w^*$.*

Proof. See Appendix A.

In fact the proof of Proposition 4 implies that the linear relaxation of (MIP3) is exact. Details about this linear relaxation will follow in an extended version of the paper.

4 Further mixed integer linear programming formulations

In this section, we present three new exact formulations for the unweighted maximum cardinality cut problem with interesting computational performances. The first one stems from (MIP2) using the fact that for all $v \in V$, $\sum_{k \in I_v^-} y_k^v = 1 - \sum_{k \in I_v^+} y_k^v$ deleting all the variables y_k^v where k is negative. We obtain the following exact formulation.

$$(MIP5) \begin{cases} Z_{\text{MIP5}}^* = \max \sum_{v \in V} \sum_{k \in I_v^+} k y_k^v \\ \text{s.t.} \\ \sum_{k \in I_v^+} y_k^v \leq 1, \quad \forall v \in V, \\ \sum_{uv \in E} (\sum_{k \in I_v^+} y_k^v - \sum_{k \in I_u^+} y_k^u) \leq \sum_{k \in I_v^+} k y_k^v - \lceil \frac{d_v}{2} \rceil (1 - \sum_{k \in I_v^+} y_k^v), \quad \forall v \in V, \\ \sum_{uv \in E} (\sum_{k \in I_v^+} y_k^v - \sum_{k \in I_u^+} y_k^u) \geq \sum_{k \in I_v^+} k y_k^v - d_v (1 - \sum_{k \in I_v^+} y_k^v), \quad \forall v \in V, \\ y_k^v \in \{0, 1\}, \quad \forall v \in V, \forall k \in I_v^+. \end{cases}$$

It involves about half as many variables as (MIP2) and has generally better performance, detailed results can be found in section 6. In order to further reduce the number of variables for an exact formulation, we can aggregate the variables y_k^v with $k \in I_v^+$ for a vertex v to form a variable x^v equal to $\sum_{k \in I_v^+} y_k^v$. For doing so, we need another variable z^v equal to $\sum_{k \in I_v^+} k y_k^v$ in order to keep the information about the difference between the outdegree and indegree of v important for the objective function. We thus obtain the following exact formulation.

$$(MIP6) \begin{cases} Z_{\text{MIP6}}^* = \max \sum_{v \in V} z^v \\ \text{s.t.} \\ \lceil \frac{d_v}{2} \rceil + \lfloor \frac{d_v}{2} \rfloor x^v - z^v \leq \sum_{uv \in E} x^u \leq d_v - z^v, \quad \forall v \in V, \\ \lceil \frac{d_v}{2} \rceil x^v \leq z^v \leq d_v x^v, \quad \forall v \in V, \\ x \in \{0, 1\}^V, \quad z \in \mathbb{R}^V. \end{cases}$$

(MIP6) involves $2|V|$ variables, half of which are integer variables and its performance is better than that of (MIP5) for many instances (see Section 6). This formulation can also be obtained using the linearization technique of Glover [14] applied to the standard quadratic program modeling the maximum cut problem. One can also propose a third formulation somewhat in between (MIP5), i.e. no aggregation of variables, and (MIP6), i.e. total aggregation of the variables for each vertex. To do so, we partition the interval I_v^+ for each vertex $v \in V$. Let

$\alpha > 1$, we parametrize such a partition with α defining the following sequences for each $v \in V$

$$\begin{cases} a_1^v = \lceil \frac{d_v}{2} \rceil, \\ a_i^v = 1 + b_{i-1}^v, \text{ for } i > 1, \\ b_i^v = \min(\lfloor \alpha * a_i^v \rfloor, d_v), \end{cases}$$

and compute k_v , the smallest integer such that $b_{k_v}^v = d_v$. Then similarly to the formulations (MIP1), (MIP2) and (MIP3), we take a variable y_k^v for each vertex $v \in V$ and each $k \in \llbracket 1, k_v \rrbracket$ whose interpretation is the following: $y_k^v = 1$ if and only if $z^v \in \llbracket a_k^v, b_k^v \rrbracket$. We therefore obtain the following exact formulation for all $\alpha > 1$.

$$(MIP7[\alpha]) \begin{cases} Z_{MIP7}^* = \max \sum_{v \in V} z^v \\ \text{s.t.} \\ \lceil \frac{d_v}{2} \rceil + \lfloor \frac{d_v}{2} \rfloor x^v - z^v \leq \sum_{uv \in E} x^u \leq d_v - z^v, \quad \forall v \in V, \\ \sum_{k=1}^{k_v} y_k^v = x^v, \quad \forall v \in V, \\ \sum_{k=1}^{k_v} a_k^v y_k^v \leq z^v \leq \sum_{k=1}^{k_v} b_k^v y_k^v, \quad \forall v \in V, \\ x \in [0, 1]^V, \quad y^v \in \{0, 1\}^{k_v}, \quad \forall v \in V, \quad z \in \mathbb{R}^V. \end{cases}$$

For the purpose of comparing performances, we now give a basic exact formulation for the unweighted maximum cut problem based on the triangle inequalities. It involves one variable $x_{i,j}$ for each unordered pair of vertices $\{i, j\} \subset V$ ($i \neq j$). Hence $O(n^2)$ variables and $O(n^3)$ constraints.

$$(MIP8) \begin{cases} \max \sum_{i,j \in E} x_{i,j} \\ \text{s.t.} \\ x_{i,j} + x_{j,k} + x_{i,k} \leq 2, \quad \forall \{i, j, k\} \subset V, \quad |\{i, j, k\}| = 3, \\ x_{i,j} + x_{j,k} - x_{i,k} \geq 0, \quad \forall (i, j, k) \in V^3, \quad |\{i, j, k\}| = 3, \\ x_{i,j} \in \{0, 1\}, \quad \forall \{i, j\} \subset V, \quad i \neq j. \end{cases}$$

5 Preliminary computational experiments

Some numerical experiments have been conducted to evaluate the quality of the new SDP bound. For each problem instance, we report w^* (the maximum cardinality of a cut), Z_{SDP0}^* (the optimal objective value of (SDP0)) and Z_{SDP3}^* (the optimal objective value of (SDP3)). We also mention some results related to the Mixed integer Programming formulations of the previous Section reporting the running time of (MIP2), (MIP5), (MIP6), (MIP7[1.5]), (MIP7[1.3]), (MIP7[1.1]) and (MIP8) for each instance.

The algorithms used for these computations were written in C/C++ calling COIN-OR's CSDP library to solve the semidefinite programs and IBM's ILOG CPLEX optimizer[©] for the linear and mixed integer programs; all have been performed with a processor 1.9GHzx4, 15.6GB RAM. In order to further the relevance of our comparison, we also give for each instance the running time (BC) of the semi-definite based solver BiqCrunch [21] compiled in Python and run on the same machine as the mixed integer programs.

The graph instances used for the computations are denoted as follows:

- K_n : the complete graph with n vertices,

- W_n : the wheel graph with n vertices (i.e. $n - 1$ spokes),
- Pe, Co, Oc, Do and Ic: the Petersen graph, the Coxeter graph, the octahedron, the dodecahedron and the icosahedron respectively. (Information about the platonic graphs can be found in [24]),
- C_n : the cycle graph with n vertices,
- $G_{n_1, n_2}^{t_2}$: the 2-dimensional toroidal grid graph, i.e. the cartesian product of two cycles $C_{n_1} \bullet C_{n_2}$,
- $G_n^{t_k}$: the n -dimensional toroidal grid graph of length k , i.e. the cartesian product $\bullet_{i=1}^n C_k$,
- $R_{n,d}$: a randomly generated graph with n vertices and density d : $d = \frac{200|E|}{n(n-1)}$,
- $P_{n,D}$: a randomly generated planar graph with n vertices and proportion of edges with respect to a maximum planar graph D : $D = \frac{100|E|}{3(n-2)}$.

The random graphs were generated using rudy, a machine-independent graph generator by Giovanni Rinaldi.

Let us start with the results related to the new SDP bound. The first set of instances considered consists of two basic graph classes : odd complete graphs and wheel graphs. One can see that $Z_{\text{SDP3}}^* = w^*$ for complete graphs as shown in Proposition 4. The bound seems to be exact for wheels (according to numerical experiments). We report in Table 2 results obtained on some well-known graphs: the Petersen graph, the Coxeter graph, the octahedron, the dodecahedron and the icosahedron, along with some toroidal grid graphs. The results reported in Table 3 were computed from randomly generated graphs.

The preliminary computational results from Tables 1-3 not only confirm the inequality proved in Proposition 3, but clearly point out that the quality of the new bound presented in the previous Sections is sometimes significantly better than that of Goemans and Williamsons relaxation.

Let us now look at the running times of the Formulations (MIP8), (MIP2), (MIP5), (MIP6), (MIP7[1.5]), (MIP7[1.3]) and (MIP7[1.1]) and of the solver (BC) on several bigger instances found in Table 4. For the entries marked “>900”, the running time exceeded 900s and the process was therefore interrupted and for the entries marked “-”, the memory of the machine was full and the process was therefore interrupted. First, one can see that the new formulations introduced in Section 4 perform much better than the classical triangular formulation (MIP8) for all the studied graph families except the general random graphs, and on all of these instances, there is one of our formulation that performs better than BiqCrunch does. More specifically, we can see that (MIP5) has generally better performance than (MIP2) and that for some instances, (MIP5) is drastically better than (MIP6), and for others, it is the other way around. Interestingly, we observe that being (MIP7) somewhat in between (MIP5) and (MIP6), there exists for almost each graph instance a value of α for which (MIP7[α]) has the shortest computing time. Practically, (MIP7[1.1]) seems to be the most robust of them.

6 Conclusion

Starting from graph orientation, we have seen that the maximum cut problem can be modeled in several new ways. By lifting, one can get some bounds that

are stronger than the standard semidefinite bound of [15]. The bounds are even exact for some graph classes. Several new mixed integer programming formulations have been obtained using discretization and aggregation techniques. The performance of these formulations compares to and is often better than the performance of the BiqCrunch solver on many graph families, it can even be improved if we strengthen the formulations either by adding valid inequalities or using other lifting techniques. Also the new formulations we introduced here for the unweighted maximum cut problem may lead to similar formulations for the weighted case. This will be part of a forthcoming paper.

Table 1. Computational results of (SDP3) for complete graphs and wheel graphs

Instance	K_5	K_7	K_{11}	W_5	W_8	W_{10}	W_{12}	W_{15}	W_{17}	W_{20}	W_{22}	W_{25}
w^*	6	12	30	6	10	13	16	21	24	28	31	36
Z_{SDP3}^*	6	12	30	6	10	13	16	21	24	28	31	36
Z_{SDP0}^*	6.25	12.25	30.25	6.25	10.614	13.809	16.979	21.875	25	29.566	32.703	37.5

Table 2. Computational results of (SDP3) for special graph classes

Instance	Pe	Co	Oc	Do	Ic	C_3	C_5	C_7	C_9	C_{11}	C_{13}	C_{15}
w^*	12	36	8	24	20	2	4	6	8	10	12	14
Z_{SDP3}^*	12	36.167	9	25	21	2	4	6.125	8.25	10.383	12.463	14.523
Z_{SDP0}^*	12.5	37.9	9	26.18	21.708	2.25	4.523	6.653	8.729	10.777	12.811	14.836

Instance	C_{17}	C_{19}	C_{21}	C_{23}	C_{25}	$G_{3,3}^{t_2}$	$G_{3,4}^{t_2}$	$G_{3,5}^{t_2}$	$G_{4,5}^{t_2}$	$G_{5,5}^{t_2}$	$G_3^{t_3}$
w^*	16	18	20	22	24	12	20	22	36	40	54
Z_{SDP3}^*	16.58	18.621	20.653	22.685	24.709	13.5	20	23.639	36	44.168	60
Z_{SDP0}^*	16.855	18.87	20.883	22.893	24.901	13.5	21	24.818	38.09	45.225	60.75

Table 3. Computational results of (SDP3) for randomly generated graphs

Instance	$R_{5,8}$	$R_{10,9}$	$R_{10,14}$	$R_{10,18}$	$R_{10,23}$	$R_{10,27}$	$R_{10,34}$	$R_{10,36}$	$R_{15,21}$	$R_{15,32}$	$R_{15,42}$
w^*	6	8	12	14	17	19	22	23	17	24	30
Z_{SDP3}^*	6	8	12	14	17	19	22	23	17	24.236	30.381
Z_{SDP0}^*	6.25	8.25	12.585	14.399	17.603	19.962	22.676	23.346	18.006	25.357	31.569

Instance	$R_{15,53}$	$R_{20,19}$	$R_{20,38}$	$R_{20,57}$	$R_{30,44}$	$P_{5,7}$	$P_{5,9}$	$P_{10,10}$	$P_{10,12}$	$P_{10,18}$	$P_{10,24}$
w^*	36	16	29	43	37	5	6	8	10	13	16
Z_{SDP3}^*	36.567	16	29.202	43	37.31	5	6	8	10	13	16
Z_{SDP0}^*	37.39	16.679	30.682	44.757	39.005	5.432	6.25	8.409	10.715	13.932	16.992

Instance	$P_{20,11}$	$P_{20,16}$	$P_{20,27}$	$P_{20,41}$	$P_{20,54}$	$P_{25,35}$	$P_{25,52}$	$P_{25,69}$	$P_{30,8}$	$P_{30,17}$	$P_{30,42}$
w^*	9	15	21	30	36	28	39	46	7	15	33
Z_{SDP3}^*	9	15	21	30	36.207	28.091	39	46.446	7	15	33.037
Z_{SDP0}^*	9.25	15.25	22.495	31.289	38.131	29.705	40.614	48.468	7.25	15.25	34.412

References

1. Anjos, M. & Wolkowicz, H.: ‘Strengthened semidefinite relaxations via a second lifting for the max-cut problem’, *Discrete Applied Mathematics* 119, pp.79–106 (2002)
2. Ashcraft, C.C., Liu, J.W.H.: ‘Using domain decomposition to find graph bisectors’, *Technical report CS-95-08, York University*, (1995)
3. Barahona, F. & Mahjoub, A.R.: ‘On the cut polytope’, *Mathematical Programming* 36, pp.157–173 (1986)
4. Barahona, F., Grötschel, M., Jünger, M., Reinelt, G.: ‘An application of combinatorial optimization to statistical physics and circuit layout design’, *Operations Research* 36, pp.493–513 (1998)
5. Ben-Ameur, W., Mahjoub, A.R. & Neto, J.: ‘The Maximum Cut Problem, in Paradigms of Combinatorial Optimization’, *V. Paschos (Ed), Wiley-ISTE* pp.131–172 (2010)

Table 4. Running time for the MIP formulations (in seconds)

Instance	w^*	MIP8	MIP2	MIP5	MIP6	MIP7 [1.5]	MIP7 [1.3]	MIP7 [1.1]	BC	V	E
$R_{20,90}$	99	3	1	1	3	1	1	1	0	20	171
$R_{25,50}$	97	7	63	9	1	2	5	5	1	25	150
$R_{25,90}$	152	71	12	3	11	20	16	4	0	25	270
$R_{30,50}$	141	30	341	71	13	21	26	39	1	30	217
$R_{30,90}$	219	>900	44	21	144	111	99	23	0	30	391
$R_{40,25}$	136	29	>900	748	33	49	47	114	1	40	195
$P_{50,50}$	373	2	1	0	0	0	0	0	3	46	72
$P_{50,90}$	598	3	17	10	2	2	1	1	4	50	129
$P_{75,40}$	75	89	0	0	1	0	1	0	5	66	87
$P_{75,50}$	90	77	2	1	1	1	1	0	6	70	109
$P_{75,70}$	120	22	7	5	1	1	1	2	2	75	153
$P_{75,100}$	146	>900	>900	>900	172	>900	62	38	47	75	219
$P_{100,40}$	100	73	0	1	0	0	1	0	8	87	117
$P_{100,90}$	190	168	>900	>900	47	102	35	34	40	100	264
$P_{200,50}$	246	>900	>900	272	75	47	31	26	304	188	297
$P_{300,50}$	376	-	>900	>900	340	203	64	53	>900	282	447
K_{50}	625	>900	2	1	0	0	0	0	0	50	1225
K_{100}	2500	>900	17	7	4	1	1	1	64	100	4950
K_{150}	5625	>900	179	42	29	4	3	3	166	150	11175
K_{175}	7656	>900	322	77	>900	>900	>900	3	5	175	15225
K_{200}	10000	>900	704	145	4	7	4	6	368	200	19900
K_{225}	12656	>900	>900	223	>900	>900	>900	8	11	225	25200
K_{300}	22500	-	>900	875	>900	>900	106	208	177	300	44850
G_4^{t4}	1024	>900	0	1	0	0	0	1	19	256	1024
$G_{10,10}^{t2}$	200	121	0	0	0	0	0	0	2	100	200
$G_{8,15}^{t2}$	232	>900	2	1	1	0	1	0	6	120	240
$G_{11,12}^{t2}$	252	>900	40	8	2	2	1	2	5	132	264
$G_{15,20}^{t2}$	580	>900	>900	>900	86	75	32	32	150	300	600
$G_{20,20}^{t2}$	800	>900	0	0	0	0	1	0	109	400	800
$G_{100,100}^{t2}$	20000	-	7	29	43	45	17	17	>900	10000	20000
W_{75}	111	26	2	3	1	1	0	1	4	75	148
W_{100}	148	58	2	7	5	3	0	1	11	100	198
W_{175}	261	>900	9	333	22	14	1	1	44	175	348
W_{250}	373	>900	61	>900	26	26	2	2	301	250	498
W_{400}	598	>900	62	>900	>900	>900	4	3	>900	400	798
W_{550}	823	-	>900	>900	>900	>900	5	5	>900	550	1098
W_{775}	1161	-	>900	>900	>900	>900	11	10	-	775	1548
W_{925}	1386	-	>900	>900	>900	626	15	15	-	925	1848
W_{1250}	1873	-	>900	>900	>900	>900	34	39	-	1250	2548
C_{100}	100	105	0	0	0	0	0	0	5	100	100
C_{175}	174	889	0	0	0	0	0	0	31	175	175
C_{250}	250	>900	0	0	0	0	0	0	71	250	250
C_{550}	550	-	1	0	0	0	0	0	>900	550	550
C_{1250}	1250	-	0	0	0	0	1	0	-	1250	1250
C_{1750}	1750	-	1	0	1	1	1	0	-	1750	1750
C_{3000}	3000	-	1	1	2	1	1	2	-	3000	3000
C_{5725}	5725	-	13	12	37	13	11	13	-	5725	5725
C_{9000}	9000	-	2	7	10	8	7	7	-	9000	9000

6. Ben-Ameur, W. & Neto, J.: 'On a gradient-based randomized heuristic for the maximum cut problem', *International journal of Mathematics in Operational Research* 4(3), pp. 276–293 (2012)
7. Ben-Ameur, W. & Neto, J.: 'Spectral bounds for the maximum cut problem', *Networks* 52(1), pp.8–13 (2008)
8. Ben-Ameur, W. & Neto, J.: 'Spectral bounds for unconstrained $(-1, 1)$ -quadratic optimization problems', *European Journal of Operational Research* 207(1), pp.15–24 (2010)
9. Calzarossa, M., G. Serazzi, G.: 'Workload characterization: a survey', *Proceedings of the IEEE* 81, pp.1136–1150 (1993)
10. Deza, M. & Laurent, M.: 'Geometry of cuts and metrics', *Springer, Berlin* (1997)
11. Dolezal, O., Hofmeister, T. and Lefmann, H.: 'A comparison of approximation algorithms for the MaxCut problem', *Report CI-/99, Universität Dortmund* (1999)
12. Festa, P., Pardalos, P., Resende, M. & Ribeiro, C.: 'Randomized heuristics for the MAX-CUT problem', *Optimization Methods and Software* 7, pp.1033–1058 (2002)
13. Fischer, I., Gruber, G., Rendl, F. & Sotirov, R.: 'Computational experience with a bundle approach for semidefinite cutting plane relaxations of Max-Cut and Equipartition', *Mathematical Programming* 105, pp.451–469 (2006)
14. Glover, F.: 'Improved linear integer programming formulations of nonlinear integer problems', *Management Science* 22, pp. 455–460 (1975)
15. Goemans, M. & Williamson, D.: 'Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming', *Journal of the ACM* 42, pp.1115–1145 (1995)
16. Grötschel, M. & Pulleyblank, W.R.: 'Weakly bipartite graphs and the max-cut problem', *Operations Research Letters* 1, pp.23–27 (1981)
17. Hammer, P.: 'Some network flow problems solved with pseudo-boolean programming', *Operations Research* 32, 388–399 (1965)
18. Hastad, J.: 'Some optimal inapproximability results', *Journal of the ACM* 48, pp.798–859 (2001)
19. Helmberg, C.: 'A cutting plane algorithm for large scale semidefinite relaxations, Technical report ZR-01-26', *Konrad-Zuse-Zentrum Berlin* (2001)
20. Karp, R.M.: 'Reducibility among combinatorial problems', *Complexity of computer computation, R.E. Miller and J.W. Thatcher (eds.), Plenum Press, New York*, pp.85–103 (1972)
21. Krislock, N., Malick, J. & Roupin, F.: 'Improved semidefinite bounding procedure for solving Max-Cut problems to optimality', *Mathematical Programming* 143 (1-2), pp.61–86 (2014)
22. Laurent, M.: 'Tighter linear and semidefinite relaxations for max-cut based on the Lovász-Schrijver lift-and-project procedure', *SIAM Journal on Optimization* 2, pp.345–375 (2001)
23. Laurent, M.: 'A Comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre Relaxations for 0-1 Programming', *Mathematics of Operations Research Vol. 28, No. 3* pp. 470–496 (2003)
24. Read, R.C. & Wilson, R.J.: 'An atlas of graphs', *Clarendon press* (1998)
25. Rendl, F., Rinaldi, G., Wiegele, A.: 'Solving Max-Cut to optimality by intersecting semidefinite and polyhedral relaxations', *IASI research report 08-11* (2008)
26. Rothvoß, T.: 'The Lasserre hierarchy in Approximation algorithms', *Lecture notes for the MAPSP Tutorial preliminary version* (2013)
27. Sahni, S. & Gonzalez, T.: 'P-complete approximation algorithms', *Journal of the Association for Computing Machinery* 23:3, pp.555–565 (1976)
28. Wiegele, A.: 'Nonlinear Optimization Techniques Applied to Combinatorial Optimization Problems', *PhD thesis, Alpen-Adria-Universität Klagenfurt* (2006).

Appendix A Proof of Proposition 4

Since Z_{SDP0}^* is exact for even complete graphs, Z_{SDP3}^* is also exact by Proposition 3.

Let us now consider odd complete graphs. Let Y be an optimal solution of (SDP3). Let π be any permutation of the set of vertices. By symmetry of the complete graph, the solution defined by $Y_{kl}^{\pi(u)\pi(v)} = Y_{kl}^{uv}$ is obviously still an optimal solution of (SDP3). By considering the set of all permutations Π_n (the symmetric group), and combining all solutions, we still get an optimal solution (by linearity) Z where $Z_{kl}^{uv} = \frac{1}{|\Pi_n|} \sum_{\pi \in \Pi_n} Y_{kl}^{\pi(u)\pi(v)}$. Since we consider all permutations, Z_{kl}^{uv} does not depend on u and v . In other words, there are numbers $f(k, l)$ and $g(k)$ such that $Z_{kl}^{uv} = f(k, l)$ if $u \neq v$ and $Z_{kk}^{uu} = g(k)$, $k, l = \frac{n-1}{2}, \dots, n-1$.

Let us build another solution Z' of (SDP3) as follows: $Z'_{kl}{}^{uv} = Z_{(-k)(-l)}^{uv}$. By symmetry of complete graphs, Z' is also an optimal solution of (SDP3). Then, $Z'' = \frac{1}{2}(Z + Z')$ is also optimal. Observe that $Z''_{kl}{}^{uv} = Z''_{(-k)(-l)}{}^{uv}$. This implies that we can assume that $g(k) = g(-k)$ and $f(k, l) = f(-k, -l)$.

Moreover, constraints (4a) lead to

$$\sum_{k \in \llbracket \frac{n-1}{2}, n-1 \rrbracket \cup \llbracket 1-n, \frac{1-n}{2} \rrbracket} g(k) = 1 = 2 \sum_{k \in \llbracket \frac{n-1}{2}, n-1 \rrbracket} g(k). \quad (7)$$

From (4c), we deduce that

$$\sum_{l \in \llbracket \frac{n-1}{2}, n-1 \rrbracket \cup \llbracket 1-n, \frac{1-n}{2} \rrbracket} f(k, l) = g(k), \quad \forall k \in \llbracket \frac{n-1}{2}, n-1 \rrbracket \cup \llbracket 1-n, \frac{1-n}{2} \rrbracket. \quad (8)$$

Using equalities (4d), we can also write that

$$kg(k) = (n-1) \sum_{l \in \llbracket 1-n, \frac{1-n}{2} \rrbracket} f(k, l), \quad \forall k \in \llbracket \frac{n-1}{2}, n-1 \rrbracket. \quad (9)$$

Considering (4f) for a positive k , we obtain $\sum_{l \in \llbracket \frac{n-1}{2}, n-1 \rrbracket \cup \llbracket 1-n, \frac{1-n}{2} \rrbracket} lf(k, l) = -\sum_{l \in \llbracket 1-n, \frac{1-n}{2} \rrbracket} f(k, l)$, which is equivalent to

$$\sum_{l \in \llbracket 1-n, \frac{1-n}{2} \rrbracket} -(l+1)f(k, l) = \sum_{l \in \llbracket \frac{n-1}{2}, n-1 \rrbracket} lf(k, l), \quad \forall k \in \llbracket \frac{n-1}{2}, n-1 \rrbracket. \quad (10)$$

Combining (8) and (9), we deduce that

$$(n-1-k) \sum_{l \in \llbracket 1-n, \frac{1-n}{2} \rrbracket} f(k, l) = k \sum_{l \in \llbracket \frac{n-1}{2}, n-1 \rrbracket} f(k, l), \quad \forall k \in \llbracket \frac{n-1}{2}, n-1 \rrbracket. \quad (11)$$

Observe that (11) implies that when $k = n-1$ then $f(n-1, l) = 0$ for $l \in \llbracket \frac{n-1}{2}, n-1 \rrbracket$. By (10), we get that $f(n-1, l) = 0$ for $l \in \llbracket 1-n, \frac{1-n}{2} \rrbracket$. We can then assume in the rest of the proof that $k < n-1$.

Observe that the left side of (10) satisfies

$$\begin{aligned} \sum_{l \in \llbracket 1-n, \frac{1-n}{2} \rrbracket} -(l+1)f(k, l) &\geq \frac{n-3}{2} \sum_{l \in \llbracket 1-n, \frac{1-n}{2} \rrbracket} f(k, l) \\ &= \frac{n-3}{2} \frac{k}{n-1-k} \sum_{l \in \llbracket \frac{n-1}{2}, n-1 \rrbracket} f(k, l) \end{aligned}$$

where the last equality is induced by (11). Let k^{max} be the largest k such that $f(k, l) \neq 0$ for some l . We already know that $k^{max} \leq n-2$. The right side of (10) necessarily satisfies $\sum_{l \in \llbracket \frac{n-1}{2}, n-1 \rrbracket} lf(k, l) \leq k^{max} \sum_{l \in \llbracket \frac{n-1}{2}, n-1 \rrbracket} f(k, l)$. Combining the two previous inequalities together with (10), we obtain

$$(k^{max} - \frac{n-3}{2} \frac{k}{n-1-k}) \sum_{l \in \llbracket \frac{n-1}{2}, n-1 \rrbracket} f(k, l) \geq 0, \quad \forall k \in \llbracket \frac{n-1}{2}, k^{max} \rrbracket. \quad (12)$$

By considering the case $k = k^{max}$ in (12), the sum $\sum_{l \in \llbracket \frac{n-1}{2}, n-1 \rrbracket} f(k^{max}, l)$ is strictly positive, leading to $k^{max} - \frac{n-3}{2} \frac{k^{max}}{n-1-k^{max}} \geq 0$. In other words, we necessarily have $k^{max} \leq \frac{n+1}{2}$. This implies that $g(k) = 0$ and $f(k, l) = 0$ if either $k > \frac{n+1}{2}$ or $k < -\frac{n+1}{2}$ (we use here the fact that $g(k) = g(-k)$ and (8)).

Writing (10) and (11) for $k = \frac{n-1}{2}$ and $k = \frac{n+1}{2}$, we get the next 4 equations.

$$(n-3)[f(\frac{n+1}{2}, \frac{-1-n}{2}) + f(\frac{n+1}{2}, \frac{1-n}{2})] = (n+1)[f(\frac{n+1}{2}, \frac{n+1}{2}) + f(\frac{n+1}{2}, \frac{n-1}{2})] \quad (13)$$

$$(n-1)f(\frac{n+1}{2}, \frac{-1-n}{2}) + (n-3)f(\frac{n+1}{2}, \frac{1-n}{2}) = (n+1)f(\frac{n+1}{2}, \frac{n+1}{2}) + (n-1)f(\frac{n+1}{2}, \frac{n-1}{2}) \quad (14)$$

$$f(\frac{n-1}{2}, \frac{-1-n}{2}) + f(\frac{n-1}{2}, \frac{1-n}{2}) = f(\frac{n-1}{2}, \frac{n+1}{2}) + f(\frac{n-1}{2}, \frac{n-1}{2}) \quad (15)$$

$$(n-1)f(\frac{n-1}{2}, \frac{-1-n}{2}) + (n-3)f(\frac{n-1}{2}, \frac{1-n}{2}) = (n+1)f(\frac{n-1}{2}, \frac{n+1}{2}) + (n-1)f(\frac{n-1}{2}, \frac{n-1}{2}) \quad (16)$$

Subtracting (13) from (14) leads to $f(\frac{n+1}{2}, \frac{n-1}{2}) = -f(\frac{n+1}{2}, \frac{-1-n}{2})$. By non-negativity of the f values, we deduce that $f(\frac{n+1}{2}, \frac{n-1}{2}) = f(\frac{n+1}{2}, \frac{-1-n}{2}) = 0$ and $f(\frac{n+1}{2}, \frac{1-n}{2}) = \frac{n+1}{n-3} f(\frac{n+1}{2}, \frac{n+1}{2})$. Subtracting $(n-1) \times (15)$ from (16) leads in a similar way to $f(\frac{n-1}{2}, \frac{1-n}{2}) = f(\frac{n-1}{2}, \frac{n+1}{2}) = 0$ and $f(\frac{n-1}{2}, \frac{-1-n}{2}) = f(\frac{n-1}{2}, \frac{n-1}{2})$.

Using (9) and the previous observations we get that:

$$g(\frac{n+1}{2}) = 2 \frac{n-1}{n+1} f(\frac{n+1}{2}, \frac{1-n}{2}) \text{ and } g(\frac{n-1}{2}) = 2f(\frac{n-1}{2}, \frac{-n-1}{2}).$$

Using the fact that $f(\frac{n+1}{2}, \frac{1-n}{2}) = f(\frac{n-1}{2}, \frac{-n-1}{2})$ and $g(\frac{n+1}{2}) + g(\frac{n-1}{2}) = \frac{1}{2}$, one can deduce that $f(\frac{n-1}{2}, \frac{-n-1}{2}) = \frac{n+1}{8n}$. Consequently, $g(\frac{n+1}{2}) = \frac{n-1}{4n}$ and $g(\frac{n-1}{2}) = \frac{n+1}{4n}$.

Remember that $Z_{SDP3}^* = \frac{1}{2} \sum_{v \in V} \sum_k |k| Y_{kk}^{vv} = \sum_{v \in V} \sum_{k > 0} k Y_{kk}^{vv}$, leading to $Z_{SDP3}^* = n (\frac{n+1}{2} g(\frac{n+1}{2}) + \frac{n-1}{2} g(\frac{n-1}{2})) = \frac{n^2-1}{4}$, and ending the proof. $\square$