Journal of Combinatorial Optimization manuscript No.
(will be inserted by the editor)

A Completely Positive formulation of the Graph Isomorphism problem and its Positive Semidefinite relaxation

Pawan Aurora · **Shashank K. Mehta**

Received: date / Accepted: date

Abstract In [1] the authors show that two graphs G_1, G_2, on n vertices each, are isomorphic if and only if the feasible region of a certain linear program, LP-GI, intersects with the Quadratic Assignment Problem (QAP)-polytope in $\mathbb{R}^{(n^4+n^2)/2}$. The linear program LP-GI in [1] is obtained by relaxing an integer linear program whose feasible points correspond to the isomorphisms between G_1, G_2. In this paper we take an analogous approach with the linear programs replaced with conic programs. A completely positive description of the QAP-polytope was obtained in [13]. By adding the graph conditions to this description we get a completely positive formulation of the graph isomorphism problem. However, analogous to integer linear programs, it is NP-hard to optimize over the cone of completely positive matrices. So we relax this formulation by replacing the cone of completely positive matrices with the cone of positive semidefinite matrices. We observe that the resulting SDP is the Lovász Theta function [10] of a graph product of G_1, G_2 and can be efficiently computed. We provide a natural heuristic that uses the SDP to solve the graph isomorphism problem. We run our heuristic on several pairs of non-isomorphic strongly regular graphs and find the results to be encouraging. Further, by adding the non-negativity constraints to the SDP, we obtain a doubly non-negative formulation, DNN-GI. We show that if the set of optimal points in DNN-GI contains a point of rank at most 3, then the given pair of graphs must be isomorphic.

Keywords Graph Isomorphism · Quadratic Assignment Problem · Lovász Theta function

P. Aurora
IISER Bhopal
Tel.: +91-755-269-2644
E-mail: paurora@iiserb.ac.in

S.K. Mehta
IIT Kanpur
E-mail: skmehta@cse.iitk.ac.in

1 Introduction

The graph isomorphism problem (GI) is a well-studied computational problem; listed as an open problem in [6]. Formally, given two graphs G_1 and G_2 on n vertices each, GI is a decision problem that asks if there exists a bijection $\sigma : V(G_1) \rightarrow V(G_2)$ such that $\{u,v\} \in E(G_1)$ iff $\{\sigma(u),\sigma(v)\} \in E(G_2)$. Each such bijection is called an isomorphism between G_1, G_2. Without loss of generality, we assume that the vertices in both the graphs are labeled by integers $1,\ldots,n$. Hence $V(G_1) = V(G_2) = [n]$ and each bijection is a permutation of $1,\ldots,n$. The most successful approach in tackling the GI problem has been a group-theoretic approach. The fastest known algorithm for GI for general graphs uses this approach and runs in quasipolynomial time [2]. A polyhedral approach, like the one used in [1], is not expected to solve the problem for general graphs in sub-exponential time [11]. ODonnell, Wright, Wu, and Zhou [12] and Codenotti, Schoenbeck, and Snook [16] studied the Lasserre hierarchy [9] of semidefinite relaxations of the integer linear program for GI. They proved that a class of graphs known as the CFI graphs [3] cannot be distinguished by $o(n)$ levels of the Lasserre hierarchy. However, none of the known results rule out a sub-exponential time algorithm for strongly regular graphs using either a polyhedral or a semidefinite approach. In this paper we present a semidefinite approach to tackle the graph isomorphism problem. We present an iterative rounding procedure and show experimentally that it is effective in differentiating between non-isomorphic strongly regular graphs. Further, we prove that if the feasible region of a certain positive semidefinite program has points of rank at most 3, then the given pair of graphs must be isomorphic.

The rest of the paper is organized as follows. In Section 2 we present a completely positive (CP) formulation of the graph isomorphism (GI) problem. In this section we first present a CP formulation of the QAP-polytope and then arrive at the CP formulation for GI by adding the graph constraints. In Section 3 we relax an optimization version of the CP formulation to obtain a SDP relaxation that we show is equivalent to a formulation of the Lovász Theta function of a graph product of the input graphs G_1, G_2. Further in this section we present our heuristic algorithm and give a geometric interpretation of the same. We also show that the SDP formulation implies several of the linear constraints of the LP formulation given in [1]. We end the section by presenting the results of some experimentation with strongly regular graphs and show that our SDP formulation is superior to the LP formulation given in [1]. In Section 4 we prove a technical result on the rank structure of the feasible region of the SDP formulation when the feasible region is restricted to the set of points that have the maximum possible value of the objective function. We prove that if this region has a point of rank at most 3, then the given pair of graphs must be isomorphic. Finally we conclude in Section 5 with some open problems.

2 CP formulation of GI

An $m \times m$ symmetric real matrix M is said to be *positive semidefinite* if it can be expressed as QQ^T for some $m \times k$ real matrix Q. If the row vectors of Q are $\mathbf{v_1}^T, \ldots, \mathbf{v_m}^T$, then we will call this set a vector-realization of M in k-dimensional space. It is easy to see that there is always a vector realization in $k = rank(M)$ dimensional space.

If matrix M has a vector realization in which each $\mathbf{v_i}$ belongs to the non-negative orthant of $\mathbb{R}^k$, then it is called a *completely positive*(CP) matrix. *cp-rank* of a completely positive matrix M is the smallest integer k such that M has a non-negative vector realization in k dimensional space.

A CP program is a program with linear constraints and a linear objective function, where the variable matrix is confined to the cone of CP matrices. In this section we give a CP formulation of GI and study its feasible region.

In [13] a CP formulation of the *Quadratic Assignment Problem* (QAP) is given. It is established there that the feasible region of a certain completely positive program QAP_{CP} is equal to the convex hull of rank-1 $n^2 \times n^2$ matrices given by $xx^T, x = vec(X), X \in \Pi$, where Π is the set of all $n \times n$ permutation matrices and $vec(X)$ is the n^2-dimensional vector obtained from X columnwise. We observe that these rank-1 matrices are nothing but the $P_\sigma^{[2]}$ matrices defined in [1]. Moreover, the feasible region of QAP_{CP} is the QAP-polytope, defined as $\mathscr{B}^{[2]}$ in [1]. Further, in [13], the authors present another CP formulation of the QAP, denoted as QAP_{CP1} with feasible region equal to the convex hull of rank-1 $(n^2+1) \times (n^2+1)$ matrices given by $yy^T, y = \begin{bmatrix} 1 \\ vec(X) \end{bmatrix}, X \in \Pi$. Note that the feasible regions of QAP_{CP} and QAP_{CP1} are in bijective correspondence. Here, we will present a CP program that is equivalent to QAP_{CP1}, in the sense that their feasible regions are identical. Then we will add the graph conditions to our CP program to get a CP formulation whose feasible region is the convex hull of those rank-1 matrices yy^T that correspond to the permutation matrices X with the underlying permutation as an isomorphism between the input pair of graphs. Clearly, this CP formulation has a non-empty feasible region if and only if the input graphs are isomorphic.

Let $\mathscr{C}^*$ denote the cone of $(n^2+1) \times (n^2+1)$ completely positive matrices. Consider the set of all $(n^2+1) \times (n^2+1)$ matrices Y with index set $(([n] \times [n]) \cup \{\omega\}) \times (([n] \times [n]) \cup \{\omega\})$, that satisfy the following set of constraints:

$$
\begin{aligned}
& Y \in \mathscr{C}^* && \text{(1a)} \\
& Y_{ij,ik} = 0 \quad , 1 \le i,j,k \le n,\ j \ne k && \text{(1b)} \\
& Y_{ji,ki} = 0 \quad , 1 \le i,j,k \le n,\ j \ne k && \text{(1c)} \\
& Y_{\omega,\omega} = 1 && \text{(1d)} \\
& Y_{ij,\omega} = Y_{ij,ij} \quad , 1 \le i,j \le n && \text{(1e)} \\
& \sum_{i,j \in [n]} Y_{ij,ij} = n && \text{(1f)}
\end{aligned}
$$

Let $\mathscr{F}$ denote the set of all $(n^2+1) \times (n^2+1)$ matrices Y that satisfy the constraints (1a)-(1f). Let Z be an $N \times N$ completely positive matrix or, more generally, a positive semidefinite matrix. Then there exists a set of vectors $\{\mathbf{u_i} | 1 \le i \le N\}$ such that $Z_{ij} = \mathbf{u_i} \cdot \mathbf{u_j}$ for all i, j. We will refer to $\{\mathbf{u_i} | 1 \le i \le N\}$ as a *vector realization* of Z.

2.1 CP description of the QAP-polytope

Let $\mathbf{w}$ be any fixed unit vector. Then for every unit vector $\mathbf{v}$, we call $\mathbf{u} = (\mathbf{w}+\mathbf{v})/2$ a *united vector with respect to* $\mathbf{w}$. Note that $\mathbf{u}\cdot\mathbf{u} \le 1$.

Observation 1 *With respect to a fixed unit vector* $\mathbf{w}$,

- *a vector* $\mathbf{u}$ *is united if and only if* $\mathbf{u}\cdot\mathbf{w} = \|\mathbf{u}\|^2$.
- *if* $\mathbf{u_1}$ *and* $\mathbf{u_2}$ *are mutually orthogonal united vectors, then* $\mathbf{u_1}+\mathbf{u_2}$ *is also a united vector.*
- *let* $\mathbf{u_1},\dots,\mathbf{u_k}$ *be a set of pairwise orthogonal united vectors. This set is maximal (i.e., no new united vector can be added to it while preserving pairwise orthogonality) if and only if* $\mathbf{w}$ *belongs to the subspace spanned by these vectors.*
- *let* $\mathbf{u_1},\dots,\mathbf{u_k}$ *be a set of pairwise orthogonal united vectors.* $\mathbf{w}$ *belongs to the subspace spanned by these vectors if and only if* $\sum_i \mathbf{u_i} = \mathbf{w}$.
- *let* $\mathbf{u_1},\dots,\mathbf{u_k}$ *be a set of pairwise orthogonal united vectors.* $\sum_i \mathbf{u_i} = \mathbf{w}$ *if and only if* $\sum_i \|\mathbf{u_i}\|^2 = 1$.

Let Y be a point in $\mathscr{F}$. Since it is a completely positive matrix, there exist vectors $\mathbf{u_{ij}}$ for $1 \le i,j \le n$ and a unit vector $\mathbf{w}$ such that $Y_{ij,kl} = \mathbf{u_{ij}}\cdot\mathbf{u_{kl}}$ and $Y_{ij,\omega} = Y_{\omega,ij} = \mathbf{u_{ij}}\cdot\mathbf{w}$. Note that the same would be true if Y was any positive semidefinite matrix. From conditions (1d) and (1e) we see that $\mathbf{u_{ij}}$ are united vectors with respect to $\mathbf{w}$. From condition (1b) we see that $\{\mathbf{u_{i1}},\dots,\mathbf{u_{in}}\}$ is a set of mutually orthogonal united vectors, for each i. Same is true for $\{\mathbf{u_{1i}},\dots,\mathbf{u_{ni}}\}$ from (1c). Further, condition (1f) enforces that each of the sets, $\{\mathbf{u_{i1}},\dots,\mathbf{u_{in}}\}$, (equivalently, $\{\mathbf{u_{1i}},\dots,\mathbf{u_{ni}}\}$) is a maximal pairwise orthogonal set of united vectors. Note that $\sum_j Y_{ij,ij} = (\sum_j \mathbf{u_{ij}})\cdot\mathbf{w}$. Since, $\sum_j \mathbf{u_{ij}}$ being the sum of pairwise orthogonal united vectors is also a united vector, $(\sum_j \mathbf{u_{ij}})\cdot\mathbf{w} = (\sum_j \mathbf{u_{ij}})\cdot(\sum_j \mathbf{u_{ij}}) \le 1$, implying that $\sum_j Y_{ij,ij} \le 1\ \forall\ i$. So condition (1f) enforces that $\sum_j Y_{ij,ij} = 1\ \forall\ i$.

Observe that Y is a $(n^2+1)\times(n^2+1)$ matrix having its last row and the last column equal to its diagonal, see condition (1e). Hence the $n^2\times n^2$ principal submatrix contains complete information of Y. We will refer to it as $\tilde{Y}$. So $\tilde{Y}_{ij,kl} = \mathbf{u_{ij}}\cdot\mathbf{u_{kl}}$ for all i,j,k,l. Clearly Y can be obtained from $\tilde{Y}$ by setting $Y_{ij,n^2+1} = Y_{n^2+1,ij} = \tilde{Y}_{ij,ij}$ and $Y_{n^2+1,n^2+1} = 1$. The remaining entries of Y are same as those of $\tilde{Y}$. So, $\tilde{Y}$ is an orthogonal projection of Y onto a $(n^2\times n^2)$-dimensional subspace. Since the projection of a convex set onto some of its coordinates is convex, the set formed by all $\tilde{Y}$ is also convex. Let us denote this set as $\mathscr{F}_p$. The following Lemma shows that $\mathscr{F}_p$ is in fact the QAP-polytope in $\mathbb{R}^{n^2\times n^2}$, henceforth denoted as $\mathscr{B}^{[2]}$.

Lemma 1 $\mathscr{F}_p = \mathscr{B}^{[2]}$.

Proof Let $P_\sigma^{[2]}$ denote a rank-1 matrix yy^T, where $y = vec(P_\sigma), P_\sigma \in \Pi$. Recall that the QAP-polytope $\mathscr{B}^{[2]}$, is defined as the convex hull of $P_\sigma^{[2]}$ for all $\sigma \in S_n$. Observe that $P_\sigma^{[2]}$ is the $n^2\times n^2$ principal submatrix of the $(n^2+1)\times(n^2+1)$ rank-1 matrix $Z = zz^T$ where $z = \begin{bmatrix} vec(P_\sigma) \\ 1 \end{bmatrix}$. Clearly, matrix Z is a completely positive matrix since it has a non-negative vector realization in 1-dimensional space. Also, $Z_{n^2+1,n^2+1} = 1$ and

$Z_{ij,n^2+1} = z_{ij} \cdot z_{n^2+1} = z_{ij} = Z_{ij,ij}$. Similarly, $Z_{n^2+1,ij} = Z_{ij,ij}$. Moreover, $(P_\sigma^{[2]})_{ij,ik} = (P_\sigma)_{ij} \cdot (P_\sigma)_{ik} = 0$ for all i and $j \neq k$. Similarly, $(P_\sigma^{[2]})_{ji,ki} = (P_\sigma)_{ji} \cdot (P_\sigma)_{ki} = 0$ for all i and $j \neq k$. Also, $\sum_{i,j\in[n]} (P_\sigma^{[2]})_{ij,ij} = \sum_{i,j\in[n]} (P_\sigma)_{ij} \cdot (P_\sigma)_{ij} = \sum_{i,j\in[n]} (P_\sigma)_{ij} = n$. Thus, $Z \in \mathscr{F}$ and $P_\sigma^{[2]} \in \mathscr{F}_p$. This is true for all $\sigma \in S_n$. Hence, $\mathscr{B}^{[2]} \subseteq \mathscr{F}_p$.

Consider a non-negative vector realization $\{\mathbf{u_{ij}} | i,j \in [n]\} \cup \{\mathbf{w}\}$ of $Y \in \mathscr{F}$. We arrange the vectors $\{\mathbf{u_{ij}} | i,j \in [n]\}$ as the entries of a $n \times n$ matrix W with (i,j)-th entry being $\mathbf{u_{ij}}$, as shown below

$$W = \begin{bmatrix} \mathbf{u_{11}} & \mathbf{u_{12}} & \mathbf{u_{13}} & \dots & \mathbf{u_{1n}} \\ \mathbf{u_{21}} & \mathbf{u_{22}} & \mathbf{u_{23}} & \dots & \mathbf{u_{2n}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{u_{n1}} & \mathbf{u_{n2}} & \mathbf{u_{n3}} & \dots & \mathbf{u_{nn}} \end{bmatrix}$$

Conditions (1b) and (1c) ensure that the vectors in any row or any column of W are pairwise orthogonal. From Observation 1 the vectors in each row/column form a maximal set of pairwise orthogonal united vectors. Also from the same observation, the vectors in each row/column add up to the vector $\mathbf{w}$. Assume that the vector realization is in an N-dimensional space, where N is the cp-rank of Y. Let W_r denote the $n \times n$ matrix formed by the r-th coordinate of each vector $\mathbf{u_{ij}}$, as shown below

$$W_r = \begin{bmatrix} \mathbf{u_{11}}(r) & \mathbf{u_{12}}(r) & \mathbf{u_{13}}(r) & \dots & \mathbf{u_{1n}}(r) \\ \mathbf{u_{21}}(r) & \mathbf{u_{22}}(r) & \mathbf{u_{23}}(r) & \dots & \mathbf{u_{2n}}(r) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{u_{n1}}(r) & \mathbf{u_{n2}}(r) & \mathbf{u_{n3}}(r) & \dots & \mathbf{u_{nn}}(r) \end{bmatrix}$$

Each element of W_r is non-negative and each row and each column adds up to $\mathbf{w}_r$, the r-th coordinate of the vector $\mathbf{w}$. Hence W_r is $\mathbf{w}_r$ times a doubly-stochastic matrix. But the vectors in the same row (resp. column) are pairwise orthogonal, implying that exactly one entry is non-zero in each row (resp. column) of W_r if $\mathbf{w}_r > 0$. So $W_r = \mathbf{w}_r P_{\sigma_r}$ for some permutation σ_r. We can express W by $\sum_r \mathbf{w}_r P_{\sigma_r} \mathbf{e_r}$ where $\mathbf{e_r}$ denotes the unit vector along the r-th axis. $\tilde{Y}_{ij,kl}$ is the inner product of the vectors $\mathbf{u_{ij}}$ and $\mathbf{u_{kl}}$ which is $(\sum_r \mathbf{w}_r (P_{\sigma_r})_{ij} \mathbf{e_r}) \cdot (\sum_s \mathbf{w}_s (P_{\sigma_s})_{kl} \mathbf{e_s}) = \sum_r \mathbf{w}_r^2 (P_{\sigma_r})_{ij} . (P_{\sigma_r})_{kl} = \sum_r \mathbf{w}_r^2 (P_{\sigma_r}^{[2]})_{ij,kl}$. Thus $\tilde{Y} = \sum_r \mathbf{w}_r^2 P_{\sigma_r}^{[2]}$. Since $\sum_r \mathbf{w}_r^2 = \|\mathbf{w}\|^2 = 1$, $\tilde{Y}$ is a convex combination of some of the $P_\sigma^{[2]}$s or $\tilde{Y} \in \mathscr{B}^{[2]}$. Hence, $\mathscr{F}_p \subseteq \mathscr{B}^{[2]}$. $\square$

2.2 GI as a CP feasibility problem

Let $G_1 = ([n], E_1)$ and $G_2 = ([n], E_2)$ be simple graphs on n vertices each. Define a graph $G = (V, E)$, where $V = [n] \times [n]$ and $\{ij, kl\} \in E$ if either $\{i,k\} \in E_1$ and $\{j,l\} \notin E_2$ or $\{i,k\} \notin E_1$ and $\{j,l\} \in E_2$, provided $i \neq k$ and $j \neq l$. Let $\mathscr{F}_p^G$ be the subset of $\mathscr{F}_p$ satisfying $\tilde{Y}_{ij,kl} = 0 \; \forall \; \{ij, kl\} \in E$.

Theorem 1 *$G_1 \cong G_2$ if and only if $\mathscr{F}_p^G \neq \emptyset$.*

Proof We will establish that $\mathscr{F}_p^G = \mathscr{B}_{G_1G_2}^{[2]}$, where $\mathscr{B}_{G_1G_2}^{[2]}$ is the convex hull of those $P_\sigma^{[2]}$ that correspond to an isomorphism between G_1, G_2. Clearly, $\mathscr{B}_{G_1G_2}^{[2]} = \emptyset$ implies that there is no $P_\sigma^{[2]}$ that corresponds to an isomorphism between G_1, G_2 or that $G_1 \ncong G_2$. On the other hand, if $\mathscr{B}_{G_1G_2}^{[2]} \neq \emptyset$, there must exist at least one $P_\sigma^{[2]}$ that corresponds to an isomorphism between G_1, G_2, or $G_1 \cong G_2$. From the definition of G above, $\tilde{Y}_{ij,kl} = 0$ implies that either $\{i,k\}$ is an edge in G_1 and $\{j,l\}$ is not an edge in G_2 or $\{i,k\}$ is not an edge in G_1 and $\{j,l\}$ is an edge in G_2. Clearly, no edge preserving bijection between G_1, G_2 can map i to k and j to l or at least one of $(P_\sigma)_{ik}$ and $(P_\sigma)_{jl}$ must be zero for all the bijections σ that give an isomorphism between G_1, G_2 or $(P_\sigma^{[2]})_{ij,kl} = 0$ for each such bijection for all $\{ij,kl\} \in E$. So a point $Z \in \mathscr{B}_{G_1G_2}^{[2]}$ must satisfy $Z_{ij,kl} = 0 \ \forall \ \{ij,kl\} \in E$. Since $Z \in \mathscr{F}_p$, we have $Z \in \mathscr{F}_p^G$. Thus, $\mathscr{B}_{G_1G_2}^{[2]} \subseteq \mathscr{F}_p^G$. Next, consider a point $\tilde{Y} \in \mathscr{F}_p^G$. From the proof of Lemma 1 we know that $\tilde{Y}_{ij,kl} = \sum_r \mathbf{w}_r^2 (P_{\sigma_r}^{[2]})_{ij,kl}$ where $\sum_r \mathbf{w}_r^2 = 1$. Since $\tilde{Y}_{ij,kl} = 0 \ \forall \ \{ij,kl\} \in E$, we have $\sum_r \mathbf{w}_r^2 (P_{\sigma_r}^{[2]})_{ij,kl} = 0 \ \forall \ \{ij,kl\} \in E$ implying that $(P_{\sigma_r}^{[2]})_{ij,kl} = 0 \ \forall \ \{ij,kl\} \in E$ for every σ_r corresponding to $\mathbf{w}_r > 0$. Thus, σ_r for every $\mathbf{w}_r > 0$ gives an edge preserving bijection between G_1, G_2 or $\tilde{Y} \in \mathscr{B}_{G_1G_2}^{[2]}$ or $\mathscr{F}_p^G \subseteq \mathscr{B}_{G_1G_2}^{[2]}$. $\square$

2.3 GI as a CP optimization problem

Consider the following optimization program:

$$
\begin{aligned}
\text{CP-GI:} \quad & \text{maximize} && \sum_{i,j \in [n]} Y_{ij,ij} \\
& \text{subject to} && \text{(1a)-(1e)} \\
& && Y_{ij,kl} = 0 \ \ , \forall \ \{ij,kl\} \in E \qquad \text{(2a)}
\end{aligned}
$$

The following observation follows from Theorem 1.

Observation 2 *Graphs G_1, G_2 are isomorphic if and only if the objective function of CP-GI attains its maximum possible value of n.*

We next investigate the maximum possible value that the objective function of CP-GI can attain when G_1, G_2 are not isomorphic. Let H denote the $n \times n$ rook's graph. The set of vertices of H is $[n] \times [n]$ and the set of edges consists of all $\{ij,kl\}$ such that $i = k, j \neq l$ and $i \neq k, j = l$. Define H_G as the graph obtained by taking the union of the edges of the graphs G and H. CP-GI can equivalently be defined as follows:

$$
\begin{aligned}
\text{CP-GI:} \quad & \text{maximize} && \sum_{i,j \in [n]} Y_{ij,ij} \\
& \text{subject to} && Y \in \mathscr{C}^* && \text{(3a)} \\
& && Y_{\omega,\omega} = 1 && \text{(3b)} \\
& && Y_{ij,\omega} = Y_{ij,ij} \ , \ 1 \leq i,j \leq n && \text{(3c)} \\
& && Y_{ij,kl} = 0 \qquad , \forall \ \{ij,kl\} \in H_G && \text{(3d)}
\end{aligned}
$$

Note that in the above formulation of CP-GI, the constraints (1b)-(1c) are subsumed by the graph constraints (3d). If we relax the above formulation by replacing the cone of completely positive matrices with the cone of positive semidefinite matrices (let us call the resulting formulation SDP-GI), then we get a formulation of the Lovász theta function [10] of the graph H_G. There are several equivalent formulations of the Lovász theta function of a graph G. It is easy to see that the following formulation is equivalent to SDP-GI. Here $\mathbf{e}$ is the vector of all ones and $\mathscr{S}^n$ denotes the space of $n \times n$ real symmetric matrices. Further, the weak optimization problem for $TH(G)$ is solvable in polynomial time [5].

$$\vartheta(G) = \max_x \{e^T x : x \in TH(G)\}, \text{where}$$

$$TH(G) = \left\{ x \in \mathbb{R}^n : \exists\, W = \begin{bmatrix} U & x \\ x^T & 1 \end{bmatrix} \in \mathscr{S}^{n+1}, \right.$$
$$\left. diag(U) = x, U_{ij} = 0, (i,j) \in E, W \succeq 0 \right\}.$$

It is shown in [7] that by replacing the positive semidefinite condition in a SDP formulation of the Lovász theta number of a graph by the completely positive condition, the optimum value of the resulting program is the stability number (independence number) of that graph. So the optimum value of CP-GI is the stability number of H_G. Note that the subset of vertices of H given by $I_\sigma = \{(ij) | i \in [n], j = \sigma(i)\}$ is an independent set of size n for every $\sigma \in S_n$.

Lemma 2 *Graphs G_1, G_2 are isomorphic if and only if the graph H_G contains an independent set of size n. Moreover, every $\sigma \in S_n$ for which I_σ is an independent set in H_G, gives an isomorphism between G_1, G_2.*

Proof Let H_G contain an independent set of size n. Clearly, this independent set corresponds to some permutation σ since any independent set in H_G of size n must contain exactly one vertex from each row and exactly one vertex from each column, viewing the vertices as the squares of a $n \times n$ chessboard. Also, there is no edge between the vertices $(i\sigma(i))$ and $(j\sigma(j))$ for all $i \neq j$. This implies that either $\{i, j\}$ is an edge in G_1 and $\{\sigma(i), \sigma(j)\}$ is an edge in G_2, or $\{i, j\}$ is not an edge in G_1 and $\{\sigma(i), \sigma(j)\}$ is not an edge in G_2. Thus, σ gives an edge preserving bijection between G_1, G_2 or σ is an isomorphism between G_1, G_2. For the other direction, let σ be an isomorphism between G_1, G_2. This implies that for every edge $\{i, j\} \in G_1$, $\{\sigma(i), \sigma(j)\} \in G_2$. So there cannot be an edge in H_G between $(i\sigma(i))$ and $(j\sigma(j))$ for all $i \neq j$ or $(i\sigma(i))$ for all $i \in [n]$ forms an independent set of size n. □

Corollary 1 *If G_1, G_2 are non-isomorphic, then the maximum independent set in H_G has size at most $n-1$. Hence, for non-isomorphic G_1, G_2, the maximum value that the objective function of CP-GI can take is $n-1$.*

It follows from a result in [8] that graphs G_1, G_2 are isomorphic if and only if $\overline{H_G}$ contains a clique of size n. The same also follows from Lemma 2.

It follows from Lemma 2 and Corollary 1, that a PTAS for the maximum independent set (MIS) problem suffices to decide graph isomorphism. However, our graph G_H has genus $\Omega(n^2)$ since it contains the complete graph on n vertices, K_n, as a subgraph. It is shown in [4] that the MIS problem cannot have a PTAS if the input graph has genus $\Omega(N)$ where N is the number of vertices in the input graph.

In this section we have seen that the graph isomorphism problem can be solved via a completely positive formulation. However, it is NP-hard to optimize over the cone of completely positive matrices. But the same is not true of a positive semidefinite formulation. Although in general semidefinite programs may not be solvable in polynomial time, they can be solved in polynomial time to within an additive error of 2^{-n^c} if the feasible region is nonempty and is contained in a polynomially-sized ball centred at the origin. This is true for the SDP we are interested in. The natural step now is to relax the cp formulation to a positive semidefinite formulation and design a rounding algorithm to obtain a solution to the former from the latter.

3 SDP relaxation - Lovász Theta function

In this section we consider the positive semidefinite relaxation of the completely positive formulation of the graph isomorphism problem. In the previous section we named the resulting semidefinite program as SDP-GI and showed that it is nothing but the Lovász theta function of the graph H_G. As mentioned above, we can solve SDP-GI to arbitrary precision in polynomial time. However, it is not guaranteed that each vector in a vector realization of a solution Y to SDP-GI, must lie in the non-negative orthant of $\mathbb{R}^k$ for any value of $k \geq rank(Y)$. It is expected that SDP-GI being a relaxation of CP-GI, must obtain a larger value for its objective function than the one obtained by CP-GI. This is not always true as the following observation shows.

Observation 3 *The maximum possible value that the objective function of SDP-GI can attain is n.*

Recall that when we arrived at the maximum possible value of the objective function of CP-GI to be n, we did not require the vectors to have non-negative co-ordinates. So the above observation also follows from the same argument. Clearly, when the given pair of graphs are isomorphic, the objective function of SDP-GI must attain a value of n, similar to the case with CP-GI. However, unlike the case with CP-GI, the objective function of SDP-GI can attain a value of n, even when the given pair of graphs are not isomorphic. So using SDP-GI to decide graph isomorphism can have only false positives but no false negatives. An optimum value strictly less than n for SDP-GI also indicates an optimum value strictly less than n for the corresponding CP-GI, implying that the given graphs are non-isomorphic. So we need to only consider the case when $\vartheta(G_H) = n$.

Let $\mathscr{T}^G$ denote the set of points at which $\vartheta(G_H) = n$ and let $\mathscr{T}_p^G$ be the orthogonal projection of $\mathscr{T}^G$ onto the $(n^2 \times n^2)$-dimensional subspace spanned by the $n^2 \times n^2$ principal submatrix of points in $\mathscr{T}^G$. Note that $\mathscr{T}_p^G$ is $\mathscr{F}_p^G$ with the CP cone replaced with the PSD cone. Clearly, $\mathscr{F}_p^G \subseteq \mathscr{T}_p^G$. Further, note that all points in $\mathscr{F}_p^G$ are $n^2 \times n^2$ matrices with entries in $\mathbb{R}_{\geq 0}$. However, the same is not true for points in $\mathscr{T}_p^G$. But we can enforce this by adding $Y_{ij,kl} \geq 0$ as an additional constraint to SDP-GI. Let the resulting program be denoted as DNN-GI, where DNN denotes the cone of doubly non-negative matrices. Also, let $\mathscr{D}_p^G$ denote $\mathscr{F}_p^G \cap \{Y_{ij,kl} \geq 0 \; \forall \; i,j,k,l\}$. Further, let $\mathscr{D}^G$ denote the feasible region of DNN-GI restricted to the points at which $\vartheta(G_H) = n$. One of our goals in the rest of this paper would be to prove certain properties of points in $\mathscr{D}_p^G$ when G_1, G_2 are isomorphic.

In Theorem 1 we established that $\mathscr{F}_p^G = \mathscr{B}_{G_1G_2}^{[2]}$, where $\mathscr{B}_{G_1G_2}^{[2]}$ is the convex hull of those $P_\sigma^{[2]}$ that correspond to an isomorphism between G_1, G_2. It follows from this theorem that $\mathscr{B}_{G_1G_2}^{[2]} \subseteq \mathscr{D}_p^G$. Since $\mathscr{B}_{G_1G_2}^{[2]} \neq \emptyset$ if and only if G_1, G_2 are isomorphic, $P_\sigma^{[2]} \in \mathscr{D}_p^G$ if and only if σ is an isomorphism between G_1 and G_2. Recall that $P_\sigma^{[2]}$ denotes a rank-1 matrix yy^T, where $y = vec(P_\sigma), P_\sigma \in \Pi$. Clearly, $P_\sigma^{[2]}$ are the only rank-1 points in $\mathscr{F}_p^G$. The following lemma shows that the same is true even for $\mathscr{D}_p^G$, which implies that when G_1, G_2 are non-isomorphic, then for any point $Y \in \mathscr{D}_p^G$, we have $rank(Y) \geq 2$. We will later improve this lower bound to 4.

Lemma 3 $P_\sigma^{[2]}$ *are the only rank-*1 *points in* $\mathscr{D}_p^G$*. Also, these constitute some of the extreme points of* $\mathscr{D}_p^G$*.*

Proof If Y is a rank-1 point in $\mathscr{D}_p^G$, then there exists a vector $\mathbf{v} = \{\mathbf{v}_{ij} | 1 \leq i,j \leq n\} \in \mathbb{R}^{n^2}$ such that $Y_{ij,kl} = \mathbf{v}_{ij} \cdot \mathbf{v}_{kl}$ or $Y = vv^T$. Since $Y_{ij,il} = 0$ for $j \neq l$, for any given i, $\mathbf{v}_{ij}$ must be zero for at least $n-1$ values of j. Similarly for a given j, $\mathbf{v}_{ij}$ is zero for at least $n-1$ values of i.

If $\mathbf{v}_{ij} = 0$ for all j, then for any arbitrary k,l, $Y_{kl,kl} = \sum_j Y_{ij,kl} = 0$. Hence $\sum_{kl} Y_{kl,kl} = 0$. This is absurd because $\sum_{kl} Y_{kl,kl}$ must be n as Y is a point in $\mathscr{D}_p^G$. So we see that for each i there exists a unique j_i such that $\mathbf{v}_{ij_i} \neq 0$ and $\mathbf{v}_{ij} = 0$ for all $j \neq j_i$. Since $Y \in \mathscr{D}_p^G$, $\sum_j Y_{ij,ij} = 1$ for each i. So $1 = \sum_j Y_{ij,ij} = \sum_j \mathbf{v}_{ij}^2 = \mathbf{v}_{ij_i}^2$. Hence $\mathbf{v}_{ij_i}$ is either 1 or -1. But $\mathbf{v}_{ij_i} \cdot \mathbf{v}_{kj_k} \geq 0$ for all i,k. So either all $\mathbf{v}_{ij_i}$ are 1 or all are -1. Let V denote the $n \times n$ matrix with $V_{ij} = \mathbf{v}_{ij}$ for all i,j. We see that each row of V has one 1 (or -1) and the rest of the entries are 0. Similarly we can show that each column has one 1 (respectively, -1). So V is a permutation matrix, say P_σ or its negation, and $Y = P_\sigma^{[2]}$. Since rank-1 points lie on extreme rays of the PSD cone they form some of the extreme points of $\mathscr{D}_p^G$. □

From the above we have a vector realization of a rank-1 point in $\mathscr{D}^G$ as $\{\mathbf{u}_{i\sigma(i)} | 1 \leq i \leq n\} \cup \{\mathbf{w}\}$ for some $\sigma \in S_n$ that gives an isomorphism between G_1, G_2. Further, the vectors $\{\mathbf{u}_{i\sigma(i)} | 1 \leq i \leq n\}$ are unit vectors aligned with the unit vector $\mathbf{w}$ or the vector realization of a rank-1 point in $\mathscr{D}^G$ is nothing but $n+1$ copies of the vector $\mathbf{w}$ or written as a set is simply the vector $\mathbf{w}$. This gives us the following iterative procedure for the graph isomorphism problem. Starting with an arbitrary point $Y \in \mathscr{D}^G$ having a vector realization $\{\mathbf{u}_{ij} | 1 \leq i,j \leq n\} \cup \{\mathbf{w}\}$, the procedure attempts to iteratively construct a rank-1 solution. Actually a rank-3 solution suffices (refer to Theorem 2). In each iteration, a vector $\mathbf{u}_{ij}$ that has the maximum inner product with the vector $\mathbf{w}$, is forced to align with the vector $\mathbf{w}$ (unless it is already aligned, in which case it is ignored). From (3c), we can easily find such a vector by examining the diagonal entries of Y and selecting the $\mathbf{u}_{ij}$ that corresponds to the maximum value of $Y_{ij,ij}$ less than 1. In case of multiple diagonal entries having the maximum value, we iterate over all of them. Further, $\mathbf{u}_{ij}$ can be aligned with $\mathbf{w}$ by adding the constraint $Y_{ij,ij} = 1$

to DNN-GI. If the procedure successfully finds a solution of value n and rank at most 3, then the given pair of graphs are isomorphic, else after forcing less than n diagonal entries to 1 the optimum value of DNN-GI drops below n, implying either that the given graphs are non-isomorphic or that the set of vectors forced to align with $\mathbf{w}$ do not coincide with any isomorphism of G_1, G_2. So if A is the set of all isomorphisms between G_1, G_2, then Algorithm 1 returns 0 when either $A = \emptyset$ i.e., the given pair of graphs are non-isomorphic, or for every $\sigma \in A$ there exists (i, j) such that $Y_{ij,ij} = 1$ is included in DNN-GI but $\sigma(i) \neq j$. Note that this can happen only when a solution of value n exists such that none of the maximum valued (< 1) diagonal entries corresponds to any $\sigma \in A$.

Algorithm 1 Algorithm for testing if G_1, G_2 are isomorphic

function GRAPHISOTEST(G_H,map,itr) ▷ map is a n-dimensional vector initialized to $map(i) = -1 \; \forall \, i \in [n]$
    $Y \leftarrow$SDP-SOLVER($DNN\text{-}GI$,G_H,map) ▷ For every $map(i) = j$ an additional constraint of the form $Y_{ij,ij} = 1$ is added to $DNN\text{-}GI$
    $itr \leftarrow itr + 1$
    if $trace(Y) < n + 1$ **then**
        return 0 ▷ G_1, G_2 are not isomorphic
    else if $rank(Y) \leq 3$ **then** ▷ Refer to Theorem 2
        return 1 ▷ G_1, G_2 are isomorphic
    else
        $val \leftarrow \max\{Y_{ij,ij} : Y_{ij,ij} < 1, i, j \in [n]\}$ ▷ The largest value less than 1 along the diagonal of Y
        for all $i, j \in [n]$ **do**
            if $Y_{ij,ij} == val$ **then**
                $map(i) \leftarrow j$ ▷ $Y_{ij,ij} = 1$ added to DNN-GI
                $flag \leftarrow$GRAPHISOTEST(G_H,map,itr)
                if $flag == 1$ **then**
                    return 1
                else
                    $map(i) \leftarrow -1$ ▷ $Y_{ij,ij} = 1$ failed to generate a solution of value n
                end if
            end if
        end for
        return 0
    end if
end function

3.1 Geometric interpretation of Algorithm 1

Consider SDP-GI without the graph conditions. We refer to a set of n pairwise orthogonal united vectors as a n-frame. Here united vectors are as defined in Section 2.1. We could think of the objective of the SDP as that of finding a set of n, n-frames, so as to maximize the total inner product of these n^2 vectors with a unit vector $\mathbf{w}$. The vectors forming a n-frame also form a basis of the n-dimensional subspace of the ambient space spanned by the n^2 vectors. Clearly, the objective function attains its maximum value when the vector $\mathbf{w}$ lies in the span of each of the n, n-frames. In the absence of any graph conditions, these n n-frames could be aligned with each other in

$n!$ different ways, to together span a n-dimensional space that contains the vector $\mathbf{w}$. Each graph condition could be interpreted as one that forces two vectors from different n-frames to be orthogonal to each other, thereby ruling out certain ways in which they could be aligned with each other. A value of less than n for the objective function implies that the graph conditions have not only ruled out all possible alignments of the n-frames with each other but also forced them far enough to contain the vector $\mathbf{w}$ in their spans. Since the objective function is the sum of the projections of the vector $\mathbf{w}$ in the n n-frames, vector $\mathbf{w}$ has a projection of value less than 1 in a n-frame in whose span it is not contained, leading to a total value of less than n. A rank-1 solution corresponds to the situation when one vector from each of the n n-frames is aligned with the vector $\mathbf{w}$, and the remaining n^2-n vectors all become zero (or are orthogonal to the vector $\mathbf{w}$). So it is natural to take the vector that the SDP solution has placed closest to the vector $\mathbf{w}$ as the one that could possibly be aligned with $\mathbf{w}$ in a rank-1 solution.

3.2 Analysis of Algorithm 1

Note that when the input pair of graphs are isomorphic, the algorithm terminates as soon as it has found a solution of value n and rank at most 3. On the other hand, for non-isomorphic graphs, the algorithm terminates only when an exhaustive search fails to find a solution of value n and rank at most 3. Clearly, in the worst-case the algorithm takes exponential time. Also, the algorithm can return 0 even when the input pair of graphs are isomorphic, since the search is not exhaustive in the sense that it is not guaranteed to try all the $n!$ possible mappings between the vertices of G_1, G_2. The search is limited to the mappings that correspond to the maximum valued diagonal entries having a value of less than 1, in the first solution obtained in each recursive call. Since in the subsequent iterations of the algorithm, each of these maximum valued entries are forced to 1, one at a time, a value of 1 is not considered when deciding the maximum valued diagonal entries of a solution Y. Recall that the maximum value that any entry in a solution matrix Y can take is 1. So Algorithm 1 can output false negatives but certainly not false positives.

In Section 3.3 we run Algorithm 1 on some pairs of isomorphic and non- isomorphic strongly regular graphs and find that in all the cases the algorithm terminates with the correct solution and in polynomial time. Note that DNN-GI can be converted to its feasibility version DNN-GI-OPT by removing the objective function and including it as the constraint $\sum_{i,j\in[n]} Y_{ij,ij} = n$. So now a feasible solution to DNN-GI-OPT is one that achieves the maximum possible value of n in DNN-GI. In the following we show that several of the constraints in [1, LP-GI] are implicit in DNN-GI-OPT, thus offering some explanation to the unexpected performance of Algorithm 1. Note that we can replace DNN-GI with DNN-GI-OPT in Algorithm 1 with minimal changes that do not affect its performance.

Consider a solution Y to DNN-GI-OPT having a vector realization $\{\mathbf{u_{ij}} | 1 \le i, j \le n\}$. Since the objective function achieves its maximum value, namely, n for Y, each set $\{\mathbf{u_{i1}}, \ldots, \mathbf{u_{in}}\}$ is a maximal orthogonal set. Similarly each set $\{\mathbf{u_{1i}}, \ldots, \mathbf{u_{ni}}\}$ is also a maximal orthogonal set. So from the united vector property we have $\sum_i \mathbf{u_{ij}} = \sum_j \mathbf{u_{ij}} =$

$\mathbf{w}$. We then have $1 = w^T w = (\sum_i \mathbf{u_{ij}}) \cdot \mathbf{w} = \sum_i Y_{ij,\omega} = \sum_i Y_{ij,ij}$. Similarly $\sum_j Y_{ij,ij} = 1$. We also have $\sum_k Y_{ij,kl} = \mathbf{u_{ij}} \cdot (\sum_k \mathbf{u_{kl}}) = \mathbf{u_{ij}} \cdot \mathbf{w} = Y_{ij,ij}$. Similarly $\sum_l Y_{ij,kl} = Y_{ij,ij}$. Note that these conditions are the same as the conditions (1c), (1d) in [1, LP-GI]. Also, note that (1a) in [1, LP-GI] is implicit in the PSD constraint while (1b) in [1, LP-GI] forms a subset of (3d) in DNN-GI-OPT. Let X_{ij} denote the $n \times n$ matrix formed by reshaping ($reshape(Y(n*(i-1)+j,:)', n, n)$ in Matlab) the (ij)th row of Y. The constraint (1f) in [1, LP-GI] can be written as $AX_{ij} = X_{ij}B$, where A, B are the adjacency matrices of G_1, G_2, respectively. The constraint (1e) in [1, LP-GI] can be written as $AX = XB$ where X is the $n \times n$ matrix formed by reshaping the diagonal of Y. We will show that (1e) is implied by (1f) and the PSD constraint, thus making it redundant for DNN-GI as long as (1f) holds. We can view the matrix X_{ij} for a PSD matrix Y as $\mathbf{u_{ij}} \cdot V$ where V is a $n \times n$ matrix such that $V(k,l) = \mathbf{u_{kl}}$. Note that the matrix V is similar to the matrix W defined in the proof of Lemma 1. So we have $A(\mathbf{u_{ij}} \cdot V) = (\mathbf{u_{ij}} \cdot V)B$ for all i, j assuming that (1f) holds. In particular, we have $A(\mathbf{u_{i1}} \cdot V) = (\mathbf{u_{i1}} \cdot V)B$ for all i. Adding these equations, we get $A(\mathbf{u_{11}} \cdot V + \mathbf{u_{21}} \cdot V + \ldots + \mathbf{u_{n1}} \cdot V) = (\mathbf{u_{11}} \cdot V + \mathbf{u_{21}} \cdot V + \ldots + \mathbf{u_{n1}} \cdot V)B$ or $A((\mathbf{u_{11}} + \mathbf{u_{21}} + \ldots + \mathbf{u_{n1}}) \cdot V) = ((\mathbf{u_{11}} + \mathbf{u_{21}} + \ldots + \mathbf{u_{n1}}) \cdot V)B$, which is the same as $A(\mathbf{w} \cdot V) = (\mathbf{w} \cdot V)B$ or $AX = XB$. Finally, consider (1f). From [1, Observation 2], a subset of the conditions given by (1f) imply the remaining conditions (those not already included in (1b)) given by (3d) in DNN-GI-OPT. This leaves the only interesting conditions in (1f) as those that correspond to $i \neq k$ and $j \neq l$. Consider the special case when the input graphs are isomorphic and Y belongs to the convex hull of the feasible $P_\sigma^{[2]}$s. Clearly, in this case $AX_{ij} = X_{ij}B$ holds for all i, j. However, we cannot claim the same for the situation when Y does not belong to the convex hull of the feasible $P_\sigma^{[2]}$s. We could add the constraint (1f) to DNN-GI-OPT without affecting its polynomial time solvability to obtain a SDP with feasible region strictly contained within the feasible region of [1, LP-GI] and hence stronger than [1, LP-GI]. In the following section however, we show via experimentation that even without (1f) the semidefinite formulation performs better than the LP formulation from [1] as far as strongly regular graphs are concerned.

3.3 Experiments with the Lovász Theta function

3.3.1 Experimental setup

We use a public domain software [14, 15] based on Matlab to solve the semidefinite program for the Lovász Theta function. Small changes are made to the software to take advantage of the fact that the matrix AA^T is a diagonal matrix, which allows for an easy solution to the system $AA^T x = b$. Also there is no need to store the matrix A separately which saves an enormous amount of space. Separate functions are written to directly compute and store only the diagonal of the square matrix AA^T without first creating matrix A, and to directly compute the matrix-vector products Ay and $A^T y$ without first creating and storing matix A. Constraints of the form $Y_{ij,ij} = 1$ are added dynamically to the SDP and these can cause the matrix AA^T to have off-diagonal entries, since $Y_{ij,ij} = Y_{ij,\omega}$ constraints are already present. To circumvent this, the constraint $Y_{ij,ij} = Y_{ij,\omega}$ is replaced with the constraint $Y_{ij,\omega} = 1$ every time a constraint of the form $Y_{ij,ij} = 1$ is added to the SDP, since $Y_{ij,\omega} = 1$ along with $Y_{ij,ij} = 1$ implies $Y_{ij,ij} = Y_{ij,\omega}$. The experiments are run on a desktop computer equipped with the Intel® Core$^{\mathrm{TM}}$ i5-6400 CPU @ 2.70$GHz \times 4$ running Matlab R2017b (9.3.0.713579) on Ubuntu 18.04.2 LTS.

We run Algorithm 1 with SDP-GI instead of DNN-GI. This is so because the SDP solver does not allow the linear conditions to be specified as inequalities. However, since the diagonal vector of a PSD matrix is always non-negative, we can create a larger matrix with the original matrix as a principal sub-matrix, and with condition that all the diagonal entries outside the principal sub-matrix are equal to the off-diagonal entries of the original matrix. This would result in a variable matrix of dimension $(N + \binom{N}{2}) \times (N + \binom{N}{2})$ if the original matrix was of dimension $N \times N$. This is not practical since our original matrix is $n^2 \times n^2$ where n is the number of vertices in the input graphs. So for even ten vertex graphs, the variable matrix of the SDP would blow up to 5050×5050. Hence, we run the solver on the original variable matrix without the non-negativity conditions and it turns out that we can still differentiate non-isomorphic strongly regular graphs in polynomial time as summarized in Table 1. However, the same is not true for CFI graphs. Here, we do get an optimal solution but with the solution matrix having negative entries. Thus our experiments with the Lovász theta function on CFI graphs are not interesting.

3.3.2 Results

Column one of Table 1 identifies the strongly regular graph family by giving its parameters whereas column two of Table 1 identifies the two graphs from this family on which the experiment is run. For example a table entry with column one as $(25, 12, 5, 6)$ implies a strongly regular graph having $n = 25$ vertices, degree of each vertex as 12, the number of common neighbors shared by each pair of adjacent vertices as 5 and the number of common neighbors shared by each pair of non-adjacent vertices as 6. A value of 2-3 under column two of this table entry means that the second and the third graphs in the order of listing on the website [17], are used as the pair of non-isomorphic graphs belonging to the class $(25, 12, 5, 6)$. A value of *i*-*i* under column two suggests that two copies of the same graph are used and hence in this case the input graphs are isomorphic. Column three of Table 1 lists the number of times that SDP-GI is solved for each input. We observe that for isomorphic graphs, the algorithm converges quickly to a solution of value n and rank at most 3. For non-isomorphic graphs, every time exactly $n + 1$ iterations of SDP-GI suffice. These iterations correspond to setting $Y_{1j,1j} = 1$ for $j = 1 \dots n$ after the first iteration where the original SDP is solved. Since in each of these n-iterations the value of the objective function drops below n, we can safely conclude that the given pair of graphs are non-isomorphic, and there is no need to run the algorithm on the remaining diagonal entries. We observe that irrespective of the input graphs being isomorphic or non-isomorphic, the solution obtained during the first iteration of SDP-GI has each diagonal entry equal to $1/n$. Since the maximum number of times that SDP-GI gets solved for any input is at most $n + 1$, Algorithm 1 runs in time that is bounded by a polynomial in n. Further, in comparison to the results with the same set of graphs in [1], the number of times that SDP-GI is solved is several orders of magnitude less than the number of times that LP-GI is solved in [1] (see [1, Table 1]). For example consider the graph pair 4-5 from the family $(50,21,8,9)$. From [1, Table 1], LP-GI is solved 1050 times before the feasible region becomes empty, as compared to only 51 times of SDP-GI.

Table 1

Class	Pair	No. of iterations of SDP-GI	Class	Pair	No. of iterations of SDP-GI	Class	Pair	No. of iterations of SDP-GI
(16,6,2,2)	1-1	6	(29,14,6,7)	4-4	2	(40,12,2,4)	2-3	41
(16,6,2,2)	1-2	17	(29,14,6,7)	4-5	30	(40,12,2,4)	2-4	41
(16,6,2,2)	2-2	5	(29,14,6,7)	5-5	2	(40,12,2,4)	2-5	41
(25,12,5,6)	1-1	3	(35,18,9,9)	1-1	2	(40,12,2,4)	3-3	3
(25,12,5,6)	1-2	26	(35,18,9,9)	1-2	36	(40,12,2,4)	3-4	41
(25,12,5,6)	1-3	26	(35,18,9,9)	1-3	36	(40,12,2,4)	3-5	41
(25,12,5,6)	1-4	26	(35,18,9,9)	1-4	36	(40,12,2,4)	4-4	2
(25,12,5,6)	1-5	26	(35,18,9,9)	1-5	36	(40,12,2,4)	4-5	41
(25,12,5,6)	2-2	2	(35,18,9,9)	2-2	2	(40,12,2,4)	5-5	4
(25,12,5,6)	2-3	26	(35,18,9,9)	2-3	36	(45,12,3,3)	1-1	4
(25,12,5,6)	2-4	26	(35,18,9,9)	2-4	36	(45,12,3,3)	1-2	46
(25,12,5,6)	2-5	26	(35,18,9,9)	2-5	36	(45,12,3,3)	1-3	46
(25,12,5,6)	3-3	2	(35,18,9,9)	3-3	2	(45,12,3,3)	1-4	46
(25,12,5,6)	3-4	26	(35,18,9,9)	3-4	36	(45,12,3,3)	1-5	46
(25,12,5,6)	3-5	26	(35,18,9,9)	3-5	36	(45,12,3,3)	2-2	6
(25,12,5,6)	4-4	2	(35,18,9,9)	4-4	2	(45,12,3,3)	2-3	46
(25,12,5,6)	4-5	26	(35,18,9,9)	4-5	36	(45,12,3,3)	2-4	46
(25,12,5,6)	5-5	2	(35,18,9,9)	5-5	2	(45,12,3,3)	2-5	46
(26,10,3,4)	1-1	2	(36,14,4,6)	1-1	2	(45,12,3,3)	3-3	4
(26,10,3,4)	1-2	27	(36,14,4,6)	1-2	37	(45,12,3,3)	3-4	46
(26,10,3,4)	1-3	27	(36,14,4,6)	1-3	37	(45,12,3,3)	3-5	46
(26,10,3,4)	1-4	27	(36,14,4,6)	1-4	37	(45,12,3,3)	4-4	6
(26,10,3,4)	1-5	27	(36,14,4,6)	1-5	37	(45,12,3,3)	4-5	46
(26,10,3,4)	2-2	2	(36,14,4,6)	2-2	2	(45,12,3,3)	5-5	6
(26,10,3,4)	2-3	27	(36,14,4,6)	2-3	37	(50,21,8,9)	1-1	2
(26,10,3,4)	2-4	27	(36,14,4,6)	2-4	37	(50,21,8,9)	1-2	51
(26,10,3,4)	2-5	27	(36,14,4,6)	2-5	37	(50,21,8,9)	1-3	51
(26,10,3,4)	3-3	2	(36,14,4,6)	3-3	2	(50,21,8,9)	1-4	51
(26,10,3,4)	3-4	27	(36,14,4,6)	3-4	37	(50,21,8,9)	1-5	51
(26,10,3,4)	3-5	27	(36,14,4,6)	3-5	37	(50,21,8,9)	2-2	2
(26,10,3,4)	4-4	3	(36,14,4,6)	4-4	2	(50,21,8,9)	2-3	51
(26,10,3,4)	4-5	27	(36,14,4,6)	4-5	37	(50,21,8,9)	2-4	51
(26,10,3,4)	5-5	2	(36,14,4,6)	5-5	2	(50,21,8,9)	2-5	51
(28,12,6,4)	1-1	4	(37,18,8,9)	1-1	3	(50,21,8,9)	3-3	2
(28,12,6,4)	1-2	29	(37,18,8,9)	1-2	38	(50,21,8,9)	3-4	51
(28,12,6,4)	1-3	29	(37,18,8,9)	1-3	38	(50,21,8,9)	3-5	51
(28,12,6,4)	1-4	29	(37,18,8,9)	1-4	38	(50,21,8,9)	4-4	2
(28,12,6,4)	2-2	6	(37,18,8,9)	1-5	38	(50,21,8,9)	4-5	51
(28,12,6,4)	2-3	29	(37,18,8,9)	2-2	2	(50,21,8,9)	5-5	2
(28,12,6,4)	2-4	29	(37,18,8,9)	2-3	38	(64,18,2,6)	1-1	6
(28,12,6,4)	3-3	5	(37,18,8,9)	2-4	38	(64,18,2,6)	1-2	65
(28,12,6,4)	3-4	29	(37,18,8,9)	2-5	38	(64,18,2,6)	1-3	65
(28,12,6,4)	4-4	5	(37,18,8,9)	3-3	2	(64,18,2,6)	1-4	65
(29,14,6,7)	1-1	2	(37,18,8,9)	3-4	38	(64,18,2,6)	1-5	65
(29,14,6,7)	1-2	30	(37,18,8,9)	3-5	38	(64,18,2,6)	2-2	6
(29,14,6,7)	1-3	30	(37,18,8,9)	4-4	2	(64,18,2,6)	2-3	65
(29,14,6,7)	1-4	30	(37,18,8,9)	4-5	38	(64,18,2,6)	2-4	65
(29,14,6,7)	1-5	30	(37,18,8,9)	5-5	2	(64,18,2,6)	2-5	65
(29,14,6,7)	2-2	2	(40,12,2,4)	1-1	4	(64,18,2,6)	3-3	6
(29,14,6,7)	2-3	30	(40,12,2,4)	1-2	41	(64,18,2,6)	3-4	65
(29,14,6,7)	2-4	30	(40,12,2,4)	1-3	41	(64,18,2,6)	3-5	65
(29,14,6,7)	2-5	30	(40,12,2,4)	1-4	41	(64,18,2,6)	4-4	6
(29,14,6,7)	3-3	2	(40,12,2,4)	1-5	41	(64,18,2,6)	4-5	65
(29,14,6,7)	3-4	30	(40,12,2,4)	2-2	5	(64,18,2,6)	5-5	5
(29,14,6,7)	3-5	30						

Continued on next page

Table 1 – continued from previous page

Class	Pair	No. of iterations of SDP-GI	Class	Pair	No. of iterations of SDP-GI	Class	Pair	No. of iterations of SDP-GI

Table 1: Results of experiments with strongly regular graphs from [17]

4 Rank structure of $\mathscr{D}_p^G$

In the previous section we showed that if the optimal feasible region of DNN-GI contains a point of rank-1, then the given pair of graphs must be isomorphic. In this section we extend this claim to all points of rank at most 3. This would allow us to terminate Algorithm 1 as soon as we have obtained a point of rank 3, rather than going all the way to construct a rank-1 solution.

Lemma 4 *Consider a point Y in $\mathscr{D}_p^G$ of rank $r \leq n$. If it has a vector realization $\{\mathbf{u_{ij}} | 1 \leq i, j \leq n\}$ such that there exist $i, j_1, \ldots, j_r$ with $\left\|\mathbf{u_{ij_1}}\right\|^2 > 0$, $\left\|\mathbf{u_{ij_2}}\right\|^2 > 0$, $\ldots$, $\left\|\mathbf{u_{ij_r}}\right\|^2 > 0$, then it belongs to the CP-feasible region, $\mathscr{B}^{[2]}_{G_1G_2}$.*

Proof Vectors $\mathbf{u_{ij_1}}, \mathbf{u_{ij_2}}, \ldots, \mathbf{u_{ij_r}}$ are mutually orthogonal vectors so they can be taken as a basis of the r-dimensional space in which all the vectors lie. Now since the remaining vectors make non-negative dot products with these r vectors, all the vectors lie in the positive orthant of this basis. Thus the given matrix Y is completely positive and hence belongs to the CP-feasible region. □

Lemma 5 *All rank-2 points of $\mathscr{D}_p^G$ belong to the CP-feasible region, $\mathscr{B}^{[2]}_{G_1G_2}$.*

Proof Let Y be a rank-2 point in $\mathscr{D}_p^G$ with vector realization $\{\mathbf{u_{ij}} | 1 \leq i, j \leq n\}$. The vectors must be in 2-dimensional space so for each i there are at most two values of j such that $\mathbf{u_{ij}}$ is non-zero. If for some i there is only one such index, $j = j_1$, such that $\mathbf{u_{ij}}$ is non-zero, then $\sum_{j=1}^{n} \mathbf{u_{ij}} = \mathbf{w}$ (this must be true since $\sum_{ij} Y_{ij,ij} = n$) implies that $\mathbf{u_{ij_1}} = \mathbf{w}$. If $\mathbf{u_{ij}}$ is non-zero for only one value of j for all i, then Y will be a rank-1 matrix, i.e., $P_\sigma^{[2]}$ for some σ. But Y is a rank-2 matrix so there exists an i such that $\mathbf{u_{ij_1}}$ and $\mathbf{u_{ij_2}}$ are non-zero for some $j_1 \neq j_2$. From Lemma 4 Y is a completely positive matrix. □

Given a vector realization of a solution of DNN-GI, $\{\mathbf{u_{ij}} | 1 \leq i, j \leq n\}$, a subset $\{\mathbf{u_{ij_i}} | i \in I\}$ of non-zero vectors will be called a *consistent set* if $\mathbf{u_{pj_p}} \cdot \mathbf{u_{qj_q}} > 0$ for all $p, q \in I$. If $I = \{1, \ldots, n\}$, then it will be called a *complete consistent set*. Let $\{\mathbf{u_{ij_i}} | 1 \leq i \leq n\}$ be a complete consistent set. Define a function f as $f(i) = j_i$ for $1 \leq i \leq n$. From (1b) and (1c) we see that f is a permutation. From the graph conditions (2a) we see that f is an isomorphism between G_1 and G_2.

Observation 4 *If $\{\mathbf{u_{ij}} | 1 \leq i, j \leq n\}$ is the vector realization of a solution Y of DNN-GI which contains a complete consistent set $\{\mathbf{u_{1\sigma(1)}}, \ldots, \mathbf{u_{n\sigma(n)}}\}$, then σ is an isomorphism between G_1 and G_2.*

Lemma 6 *Let $\{\mathbf{u_{j1}}, \mathbf{u_{j2}}\}$ be a pair of orthogonal united vectors with $\mathbf{u_{j1}} + \mathbf{u_{j2}} = \mathbf{w}$, for $j = 1, 2, \ldots, r$. Then there exists a consistent set of vectors $\{\mathbf{v_j} \in \{\mathbf{u_{j1}}, \mathbf{u_{j2}}\} | j = 1, \ldots, r\}$.*

Proof Since $\mathbf{u_{j1}} \cdot \mathbf{u_{j2}} = 0$ and $\mathbf{u_{j1}} + \mathbf{u_{j2}} = \mathbf{w}$, $\|\mathbf{u_{j1}}\|^2 + \|\mathbf{u_{j2}}\|^2 = 1$ for each j. Without loss of generality assume that for each j, $\|\mathbf{u_{j1}}\|^2 \geq \|\mathbf{u_{j2}}\|^2$. Let $\mathbf{v_j} = \mathbf{u_{j1}}$. So $\|\mathbf{v_j}\|^2 \geq 1/2\ \forall j$. For arbitrary $p \neq q$, we will show that $\mathbf{u_{p1}} \cdot \mathbf{u_{q1}} > 0$. Contrary to the claim, assume that $\mathbf{u_{p1}} \cdot \mathbf{u_{q1}} = 0$. From the above, we have $\|\mathbf{u_{p1}}\|^2 = \mathbf{u_{p1}} \cdot \mathbf{w} = \mathbf{u_{p1}} \cdot \mathbf{u_{q1}} + \mathbf{u_{p1}} \cdot \mathbf{u_{q2}} = \mathbf{u_{p1}} \cdot \mathbf{u_{q2}}$. Similarly $\|\mathbf{u_{q1}}\|^2 = \mathbf{u_{p2}} \cdot \mathbf{u_{q1}}$. Also, we have $1 = \|\mathbf{w}\|^2 = (\mathbf{u_{p1}} + \mathbf{u_{p2}}) \cdot (\mathbf{u_{q1}} + \mathbf{u_{q2}}) = \|\mathbf{u_{p1}}\|^2 + \|\mathbf{u_{q1}}\|^2 + \mathbf{u_{p2}} \cdot \mathbf{u_{q2}}$. First consider the case that $\|\mathbf{u_{p1}}\|^2 > 1/2$ or $\|\mathbf{u_{q1}}\|^2 > 1/2$. In this case $1 = \|\mathbf{u_{p1}}\|^2 + \|\mathbf{u_{q1}}\|^2 + \mathbf{u_{p2}} \cdot \mathbf{u_{q2}} > 1$, which is absurd. Next we consider the remaining case that $\|\mathbf{u_{p1}}\|^2 = \|\mathbf{u_{q1}}\|^2 = 1/2$. From the above, $\mathbf{u_{p1}} \cdot \mathbf{u_{q2}} = 1/2$ and $\mathbf{u_{p2}} \cdot \mathbf{u_{q1}} = 1/2$. So $\|\mathbf{u_{p1}} - \mathbf{u_{q2}}\|^2 = 1/2 + 1/2 - 2\mathbf{u_{p1}} \cdot \mathbf{u_{q2}} = 0$, giving $\mathbf{u_{p1}} = \mathbf{u_{q2}}$. Similarly we can show that $\mathbf{u_{p2}} = \mathbf{u_{q1}}$. This gives $\{\mathbf{u_{p1}}, \mathbf{u_{p2}}\} = \{\mathbf{u_{q1}}, \mathbf{u_{q2}}\}$, which violates our assumption that $p \neq q$. □

Lemma 7 *The vector realization $\{\mathbf{u_{ij}} | 1 \leq i, j \leq n\}$ of any rank-3 point in $\mathscr{D}_p^G$ contains at least one complete consistent set.*

Proof If there exist i, j_1, j_2, j_3 such that $\|\mathbf{u_{ij_1}}\|^2 > 0$, $\|\mathbf{u_{ij_2}}\|^2 > 0$, $\|\mathbf{u_{ij_3}}\|^2 > 0$, then the claim holds from Lemma 4.

Next suppose the only non-zero vectors are $\{\mathbf{u_{ij_i}}, \mathbf{u_{ik_i}} | 1 \leq i \leq r\} \cup \{\mathbf{u_{ij_i}} | i > r\}$. So $\mathbf{u_{ij_i}} = \mathbf{w}$ for $i > r$. From Lemma 6 there exist $\mathbf{v_i} \in \{\mathbf{u_{ij_i}}, \mathbf{u_{ik_i}}\}$ for $1 \leq i \leq r$ such that $\mathbf{v_{i'}} \cdot \mathbf{v_{i''}} > 0$ for all $1 \leq i', i'' \leq r$. The desired complete consistent set is $\{\mathbf{x_i} | 1 \leq i \leq n\}$ where $\mathbf{x_i} = \mathbf{v_i}$ for $i \leq r$ and $\mathbf{x_i} = \mathbf{w}$ for $i > r$. Observe that $\mathbf{v_i} \cdot \mathbf{w} = \|\mathbf{v_i}\|^2 > 0$. □

Theorem 2 *If the set of optimal points in DNN-GI contains a point of rank at most 3, then G_1, G_2 are isomorphic.*

5 Conclusions

In this paper we presented a semidefinite approach to graph isomorphism and showed its relation to the well known Lovász Theta function of a graph product of the input graphs. We presented an iterative rounding procedure to construct a rank-1 solution to establish that the given pair of graphs are isomorphic. So far this procedure is simply a heuristic and can possibly give false negatives. We leave it as an open problem to prove the correctness of this procedure or to show that it is not guaranteed to always return a rank-1 solution if one exists. Our experimental results with strongly regular graphs are encouraging and it would be a worthwhile exercise to try these experiments with larger graphs to see if the same holds true even for them. In any case it is desirable to find an explanation for the experimental results reported in this paper. Further, it would be interesting to see if the results of Section 4 can be extended to points of rank-4 and higher.

References

1. Aurora, P., Mehta, S.K.: The qap-polytope and the graph isomorphism problem. J. Comb. Optim. **36**(3), 965–1006 (2018)
2. Babai, L.: Graph isomorphism in quasipolynomial time [extended abstract]. In: Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016, pp. 684–697 (2016)
3. Cai, J., Fürer, M., Immerman, N.: An optimal lower bound on the number of variables for graph identifications. Combinatorica **12**(4), 389–410 (1992)
4. Chen, J., Kanj, I.A., Perkovic, L., Sedgwick, E., Xia, G.: Genus characterizes the complexity of certain graph problems: Some tight results. J. Comput. Syst. Sci. **73**(6), 892–907 (2007)
5. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization, *Algorithms and Combinatorics*, vol. 2. Springer (1988). DOI 10.1007/978-3-642-97881-4
6. Karp, R.M.: Reducibility among combinatorial problems. In: Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York., pp. 85–103 (1972)
7. de Klerk, E., Pasechnik, D.V.: Approximation of the stability number of a graph via copositive programming. SIAM Journal on Optimization **12**(4), 875–892 (2002)
8. Kozen, D.: A clique problem equivalent to graph isomorphism. SIGACT News **10**(2), 50–52 (1978)
9. Lasserre, J.B.: Global optimization with polynomials and the problem of moments. SIAM Journal on Optimization **11**(3), 796–817 (2001)
10. Lovász, L.: On the shannon capacity of a graph. Information Theory, IEEE Transactions on **25**(1), 1–7 (Jan 1979)
11. Malkin, P.N.: Sherali-adams relaxations of graph isomorphism polytopes. Discrete Optimization **12**, 73–97 (2014)
12. O'Donnell, R., Wright, J., Wu, C., Zhou, Y.: Hardness of robust graph isomorphism, lasserre gaps, and asymmetry of random graphs. In: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014, pp. 1659–1677 (2014)
13. Povh, J., Rendl, F.: Copositive and semidefinite relaxations of the quadratic assignment problem. Discrete Optimization **6**(3), 231–241 (2009)
14. Santis, M.D., Rendl, F., Wiegele, A.: Dadal – using a factored dual in augmented lagrangian methods for semidefinite programming. https://www.math.aau.at/or/Software/ (2017)
15. Santis, M.D., Rendl, F., Wiegele, A.: Using a factored dual in augmented lagrangian methods for semidefinite programming. Oper. Res. Lett. **46**(5), 523–528 (2018)
16. Snook, A., Schoenebeck, G., Codenotti, P.: Graph isomorphism and the lasserre hierarchy. CoRR **abs/1401.0758** (2014)
17. Spence, E.: Strongly regular graphs. http://www.maths.gla.ac.uk/ es/srgraphs.php. Accessed: 17/04/2019