Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Discrete Applied Mathematics 143 (2004) 155–165
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

On decision and optimization (k, l)-graph sandwich problems

Simone Dantas[a], Celina M.H. de Figueiredo[a,b], Luerbio Faria[c]

[a] *COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil*
[b] *Instituto de Matematica, Universidade Federal do Rio de Janeiro, Caixa Postal 68530, Rio de Janeiro 21945-970, RJ, Brazil*
[c] *Departamento de Matemática, Faculdade de Formação de Professores, UERJ, Brazil*

Received 13 September 2002; received in revised form 4 February 2004; accepted 19 February 2004

Abstract

A graph G is (k, l) if its vertex set can be partitioned into at most k independent sets and l cliques. The (k, l)-Graph Sandwich Problem asks, given two graphs $G^1 = (V, E^1)$ and $G^2 = (V, E^2)$, whether there exists a graph $G = (V, E)$ such that $E^1 \subseteq E \subseteq E^2$ and G is (k, l). In this paper, we prove that the (k, l)-Graph Sandwich Problem is NP-complete for the cases $k = 1$ and $l = 2$; $k = 2$ and $l = 1$; or $k = l = 2$. This completely classifies the complexity of the (k, l)-Graph Sandwich Problem as follows: the problem is NP-complete, if $k + l > 2$; the problem is polynomial otherwise. We consider the degree Δ constraint subproblem and completely classify the problem as follows: the problem is polynomial, for $k \leqslant 2$ or $\Delta \leqslant 3$; the problem is NP-complete otherwise. In addition, we propose two optimization versions of graph sandwich problem for a property Π: MAX-Π-GSP and MIN-Π-GSP. We prove that MIN-$(2, 1)$-GSP is a Max-SNP-hard problem, i.e., there is a positive constant ε, such that the existence of an ε-approximative algorithm for MIN-$(2, 1)$-GSP implies $P = \text{NP}$.

Keywords: Partition problems; Sandwich problems; NP-complete problems; Max-SNP-hard problems

1. Introduction

We say that a graph $G^1 = (V, E^1)$ is a *spanning* subgraph of $G^2 = (V, E^2)$ if $E^1 \subseteq E^2$; and that a graph $G = (V, E)$ is a *sandwich* graph for the pair G^1, G^2 if $E^1 \subseteq E \subseteq E^2$. For notational simplicity in the sequel, we let E^3 be the set of all edges in the complete graph with vertex set V which are not in E^2. Thus every sandwich graph for the pair G^1, G^2 satisfies $E^1 \subseteq E$ and $E \cap E^3 = \emptyset$. We call E^1 the *forced edge set*, $E^2 \setminus E^1$ the *optional edge set*, E^3 the *forbidden edge set*. The GRAPH SANDWICH PROBLEM FOR PROPERTY Π is defined as follows [11]:

GRAPH SANDWICH PROBLEM FOR PROPERTY Π
Instance: Vertex set V, forced edge set E^1, forbidden edge set E^3.
Question: Is there a graph $G = (V, E)$ such that $E^1 \subseteq E$ and $E \cap E^3 = \emptyset$ that satisfies property Π?
We shall use both forms (V, E^1, E^2) and (V, E^1, E^3) to refer to an instance of a graph sandwich problem.

Graph sandwich problems have attracted much attention lately arising from many applications and as a natural generalization of recognition problems [6,8,10–12,14,15]. The recognition problem for a class of graphs $\mathscr{C}$ is equivalent to the graph sandwich problem in which the forced edge set $E^1 = E$, the optional edge set $E^2 \setminus E^1 = \emptyset$, $G = (V, E)$ is the graph we want to recognize, and property Π is "to belong to class $\mathscr{C}$".

E-mail addresses: sdantas@cos.ufrj.br (S. Dantas), celina@cos.ufrj.br (C.M.H. de Figueiredo), luerbio@cos.ufrj.br (L. Faria).

0166-218X/$ - see front matter © 2004 Published by Elsevier B.V.
doi:10.1016/j.dam.2004.02.008

Golumbic et al. [11] have considered sandwich problems with respect to several subclasses of perfect graphs, and proved that the GRAPH SANDWICH PROBLEM FOR SPLIT GRAPHS remains in P. On the other hand, they proved that the GRAPH SANDWICH PROBLEM FOR PERMUTATION GRAPHS turns out to be NP-complete.

We are interested in graph sandwich problems for properties Π related to decompositions arising in perfect graph theory: homogeneous set [6], join composition [8]. In this paper, we consider the decomposition of a graph into independent sets and cliques.

Let G be an undirected, finite, simple graph. A (k,l) partition of a graph G is a partition of its vertex set into at most k independent sets and l cliques. A graph is (k,l) if it admits a (k,l) partition. The complexity of (k,l) graph recognition has been completely classified as follows: if $k=3$ and $l=0$ then the corresponding problem is 3-coloring, which implies [2,3] that the recognition of (k,l) graphs is NP-complete, whenever $k \geqslant 3$ or $l \geqslant 3$. For the remaining values of k and l, the problem is polynomial: $(1,1)$ graphs are split graphs; $(2,0)$ graphs are the bipartite graphs; the polynomial-time recognition of $(2,1)$ graphs and consequently of graphs $(1,2)$ was established in [2–4]; the polynomial time recognition of $(2,2)$ graphs was established in [2,3] and independently in [7].

The studies on sandwich problems focus on those problems which are interesting in terms of their complexity, i.e., neither trivially NP-complete nor trivially polynomial.

Fact 1. *If the recognition problem for a class of graphs $\mathscr{C}$ is* NP-*complete*, *then its corresponding sandwich problem is also* NP-*complete*.

Fact 2. *If the property Π is hereditary then there exists a sandwich graph for (V,E^1,E^2) with the property Π if and only if $G^1=(V,E^1)$ has the property Π.*

Fact 3. *If the property Π is ancestral then there exists a sandwich graph for (V,E^1,E^2) with the property Π if and only if $G^2=(V,E^2)$ has the property Π.*

Thus, Fact 1 says that the sandwich problem for (k,l) graphs is NP-complete, whenever $k \geqslant 3$ or $l \geqslant 3$. In addition, Fact 2 (respectively, Fact 3) says that for each property which is hereditary (respectively, ancestral), the graph sandwich problem reduces to the recognition problem for this property on the single graph G^1 (respectively, G^2). Therefore, the hereditary properties defining $(1,0)$ and $(2,0)$ graphs, and the ancestral properties defining $(0,1)$ and $(0,2)$ graphs reduces these graph sandwich problems to recognition problems that are polynomial. Given a property Π, we define its *complementary property* $\bar{\Pi}$ as follows: for every graph G, say G satisfies $\bar{\Pi}$ if and only if $\bar{G}$ satisfies Π.

Fact 4. *There is a sandwich graph with property Π for the instance (V,E^1,E^3) if and only if there is a sandwich graph with property $\bar{\Pi}$ for the instance (V,E^3,E^1).*

Thus, our proof of the NP-completeness of the sandwich problem for $(2,1)$ graphs implies the NP-completeness of the sandwich problem for $(1,2)$ graphs.

Given $V' \subset V$ and E a set of edges over the set V, the set $E[V']=\{uv\colon u \in V' \text{ and } v \in V'\}$ is the subset of edges $e=uv$ of E such that e has its endpoints in V'. In this paper we propose two optimization versions of graph sandwich problem for a property Π. These versions are: one maximization problem, called MAX-Π-GSP, and one minimization problem, called MIN-Π-GSP. Let (V,E^1,E^2) be an instance of graph sandwich problem for a property Π. The problem MAX-Π-GSP consists in finding the maximum subset V' of V such that $(V',E^1[V'],E^2[V'])$ has answer YES for the decision version of the graph sandwich problem for the property Π. The problem MIN-Π-GSP consists in finding the minimum subset V' of V such that $(V \setminus V',E^1[V \setminus V'],E^2[V \setminus V'])$ has answer YES for the decision version of the graph sandwich problem for the property Π. In this paper we give a first application for these definitions. We prove that MIN-$(2,1)$-GSP, MIN-$(1,2)$-GSP and MIN-$(2,2)$-GSP are Max SNP-hard problems [1,16], which means that there is a positive constant ε, where the existence of an ε-approximative algorithm implies $\mathrm{P}=\mathrm{NP}$.

This paper is organized as follows: in Section 2 we prove that the $(2,1)$-Graph Sandwich Problem is NP-complete. Section 3 contains the proof that the $(2,2)$-Graph Sandwich Problem is NP-complete. These results together with the facts above completely classify the complexity of the (k,l)-Graph Sandwich Problem as follows: the problem is NP-complete, if $k+l>2$; and polynomial otherwise. Section 4 defines and classifies the degree constraint subproblems obtained by bounding the maximum degree in G^2. Sections 5 and 6 consider our proposed optimization versions. We present in Section 7 our concluding remarks.

2. (2, 1)-Graph Sandwich Problem

In this section we prove that the $(2,1)$-GRAPH SANDWICH PROBLEM is NP-complete by reducing the NP-complete problem 3-SATISFIABILITY to $(2,1)$-GRAPH SANDWICH PROBLEM. These two decision problems are defined as follows.

3-SATISFIABILITY (3SAT)

Instance: Set $X=\{x_1,\ldots,x_n\}$ of variables, collection $C=\{c_1,\ldots,c_m\}$ of clauses over X such that each clause $c\in C$ has $|c|=3$ literals.

Question: Is there a truth assignment for X such that each clause in C has at least one true literal?

$(2,1)$-GRAPH SANDWICH PROBLEM

Instance: Vertex set V, forced edge set E^1, forbidden edge set E^3.

Question: Is there a graph $G=(V,E)$, such that $E^1\subseteq E$ and $E\cap E^3=\emptyset$, and G is $(2,1)$?

Theorem 5. *The* $(2,1)$-GRAPH SANDWICH PROBLEM *is* NP-*complete*.

Proof. In order to reduce 3SAT to $(2,1)$-GRAPH SANDWICH PROBLEM we need to construct a particular instance (V,E^1,E^3) of $(2,1)$-GRAPH SANDWICH PROBLEM from a generic instance (X,C) of 3SAT, such that C is satisfiable if and only if (V,E^1,E^3) admits a sandwich graph $G=(V,E)$ which is $(2,1)$. First we describe the construction of a particular instance (V,E^1,E^3) of $(2,1)$-GRAPH SANDWICH PROBLEM; second we prove in Lemma 6 that every graph $G=(V,E)$ satisfying $E^1\subseteq E$ and $E\cap E^3=\emptyset$ and such that G is $(2,1)$, defines a truth assignment for (X,C); third we prove in Lemma 10 that every truth assignment for (X,C) defines a graph $G=(V,E)$ which is $(2,1)$ satisfying $E^1\subseteq E$ and $E\cap E^3=\emptyset$. These steps are explained in detail below. □

2.1. *Construction of particular instance of* $(2,1)$-GRAPH SANDWICH PROBLEM

The vertex set V contains: an auxiliary set of vertices: $\{k_1,k_2,s_{11},s_{12},s_{21},s_{22}\}$; for each variable x_i, $1\leqslant i\leqslant n$, two vertices x_i, $\overline{x_i}$, corresponding to its literals and a vertex p_i; for each clause $c_j=(l_1^j\vee l_2^j\vee l_3^j)$, $1\leqslant j\leqslant m$, three corresponding vertices t_1^j, t_2^j, t_3^j. In Fig. 1, solid edges are forced E^1-edges and dashed edges are forbidden E^3-edges.

The Forced Edge Set E^1 contains: edges between auxiliary vertices $\{k_1k_2,\ k_1s_{11},\ k_1s_{12},\ s_{11}s_{12},\ k_2s_{21},\ k_2s_{22},\ s_{21}s_{22}\}$; for each variable x_i, $1\leqslant i\leqslant n$, the set $\{x_is_{11},\ \overline{x_i}s_{12},\ x_ip_i,\ \overline{x_i}p_i\}$; for each clause c_j, $1\leqslant j\leqslant m$, the set $\{t_1^jt_2^j, t_1^jt_3^j, t_2^jt_3^j\}$.

The Forbidden Edge Set E^3 contains: edges between auxiliary vertices: $\{k_1s_{21},\ k_1s_{22},\ k_2s_{11},\ k_2s_{12},\ s_{11}s_{21},\ s_{11}s_{22},\ s_{12}s_{21},\ s_{12}s_{22}\}$; for each variable x_i, $1\leqslant i\leqslant n$, the set $\{x_i\overline{x_i},\ p_ik_2\}$; for each clause $c_j=(l_1^j\vee l_2^j\vee l_3^j)$, $1\leqslant j\leqslant m$, $\{t_1^jl_1^j,\ t_2^jl_2^j,\ t_3^jl_3^j\}$.

We call $(2,1)$ *base graph* the subgraph of $G^2=(V,E^2)$ induced by $\{k_1,k_2,s_{11},s_{12},s_{21},s_{22}\}$ (see Fig. 1(BG)). For each $i\in\{1,\ldots,n\}$, we call *variable gadget* the subgraph of $G^2=(V,E^2)$ induced by $\{x_i,\ \overline{x_i},\ p_i\}$ (see Fig. 1(VG)). For each $j\in\{1,\ldots,m\}$, we call *clause gadget* the subgraph of $G^2=(V,E^2)$ induced by $\{t_1^j,t_2^j,t_3^j\}$ (see Fig. 1(CG)). Lemmas 6 and 10 prove the required equivalence for establishing Theorem 5.

Lemma 6. *If the particular instance* (V,E^1,E^3) *of* $(2,1)$-GRAPH SANDWICH PROBLEM *constructed above admits a graph* $G=(V,E)$ *such that* $E^1\subseteq E$ *and* $E\cap E^3=\emptyset$ *and* G *is* $(2,1)$, *then there exists a truth assignment that satisfies* (X,C).

Proof. Suppose there exists a $(2,1)$ sandwich graph $G=(V,E)$ with $(2,1)$ partition (S_1,S_2,K) where S_1, S_2 are independent sets and K is a clique.

Claim 7. $k_1,k_2\in K$ *and* $s_{11},s_{12},s_{21},s_{22}\in S_1\cup S_2$.

Proof. Since $S_1\cup S_2$ induce a bipartite subgraph in G, any triangle induced in G^1 must have at least one of its vertices in K. Hence, at least one vertex of the triangle induced by k_1,s_{11} and s_{12}; and at least one vertex of the triangle induced

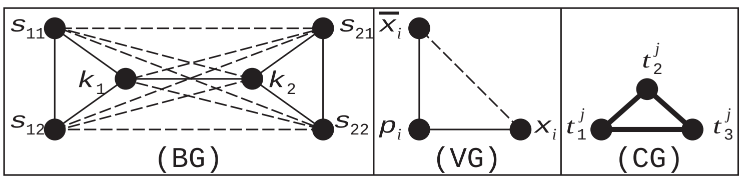

Fig. 1. Base graph (BG), Variable gadget (VG) and Clause gadget (CG).

by k_2, s_{21} and s_{22} belong to K. Now, each vertex in $\{s_{11}, s_{12}, s_{21}, s_{22}\}$ is joined by E^3-edges to three vertices that induce a triangle in G^1. If one of the vertices of $\{s_{11}, s_{12}, s_{21}, s_{22}\}$ belonged to K, then this would force at least one triangle to have no vertices in K, a contradiction. Thus, we must have $\{s_{11}, s_{12}, s_{21}, s_{22}\} \subseteq S_1 \cup S_2$, and $\{k_1, k_2\} \subseteq K$. □

Both $\{s_{11}, s_{12}\}$ and $\{s_{21}, s_{22}\}$ induce edges in G^1, which force $\{s_{11}, s_{12}\} \cap S_i \neq \emptyset$, $\{s_{21}, s_{22}\} \cap S_i \neq \emptyset$, $i = 1, 2$. We assume with no loss of generality that $s_{11}, s_{21} \in S_1$, which implies $s_{12}, s_{22} \in S_2$. In case the particular instance (V, E^1, E^3) admits a $(2,1)$ sandwich graph $G = (V, E)$ any $(2,1)$ partition (S_1, S_2, K) for G satisfies $S_1, S_2, K \neq \emptyset$.

Claim 8. *For each* $i \in \{1, \dots, n\}$, $p_i \in S_1 \cup S_2$, $x_i \in K \cup S_2$ *and* $\overline{x_i} \in K \cup S_1$.

Proof. Since $p_i k_2 \in E^3$ and $k_2 \in K$, we have that p_i cannot be in K. In addition, $x_i s_{11}, \overline{x_i} s_{12} \in E^1$ and $s_{11} \in S_1$, $s_{12} \in S_2$, we have respectively $x_i \in K \cup S_2$ and $\overline{x_i} \in K \cup S_1$, $i \in \{1, \dots, n\}$. □

Observe that since $x_i p_i \in E^1$ and $x_i \overline{x_i} \in E^3$, we have that if $x_i \in K$, then $\overline{x_i} \in S_1$, which implies $p_i \in S_2$; if $x_i \in S_2$, then $p_i \in S_1$, which implies $\overline{x_i} \in K$. Therefore, for each $i \in \{1, \dots, n\}$, exactly one vertex of $\{x_i, \overline{x_i}\}$ belongs to K.

Claim 9. *For each* $j \in \{1, \dots, m\}$, *at least one of the vertices* $\{t_1^j, t_2^j, t_3^j\}$ *must be in* K.

Proof. Since $S_1 \cup S_2$ induce a bipartite subgraph in G, for each $j \in \{1, \dots, m\}$, at least one of the vertices of the triangle induced in G^1 by $\{t_1^j, t_2^j, t_3^j\}$ must be in K. □

We now define the truth assignment for (X, C): for $i \in \{1, \dots, n\}$, variable x_i is false if and only if the vertex $x_i \in K$. Suppose that for some $j \in \{1, \dots, m\}$, the clause $c_j = (l_1^j \vee l_2^j \vee l_3^j)$ is false. By the construction of (V, E^1, E^3), there is an edge of E^3 between the vertex assigned to the literal l_k^j and the vertex t_k^j, $k \in \{1, 2, 3\}$. Hence, if the literal l_k^j is false, then its corresponding vertex is in K which implies that t_k^j cannot be in K. So, all vertices of the triangle induced in G^1 by $\{t_1^j, t_2^j, t_3^j\}$ must be in $S_1 \cup S_2$. By Claim 9, this is a contradiction to the hypothesis that S_1, S_2 and K is a $(2,1)$ partition of the set of vertices of G. Hence, the above defined truth assignment satisfies (X, C). This ends the proof of Lemma 6. □

The converse of Lemma 6 is given next by Lemma 10.

Lemma 10. *If there exists a truth assignment that satisfies* (X, C), *then the particular instance* (V, E^1, E^3) *of* $(2,1)$-GRAPH SANDWICH PROBLEM *constructed above admits a graph* $G = (V, E)$ *such that* $E^1 \subseteq E$ *and* $E \cap E^3 = \emptyset$ *and* G *is* $(2,1)$.

Proof. Suppose there is a truth assignment that satisfies (X, C). We shall define a partition of V into sets S_1, S_2 and K that in turn defines a solution G for the particular instance (V, E^1, E^3) of $(2,1)$-GRAPH SANDWICH PROBLEM associated with the 3SAT instance (X, C).

Place vertices $k_1, k_2 \in K$ and $s_{11}, s_{21} \in S_1$ and $s_{12}, s_{22} \in S_2$. For $i \in \{1, \dots, n\}$ if variable x_i is false then place vertices x_i in K, $\overline{x_i}$ in S_1 and p_i in S_2. Otherwise, if variable x_i is true, then place vertices x_i in S_2, $\overline{x_i}$ in K and p_i in S_1.

For $j \in \{1, \dots, m\}$ and $c_j = (l_1^j \vee l_2^j \vee l_3^j)$, place the corresponding vertices t_1^j, t_2^j, t_3^j as follows. For $k \in \{1, 2, 3\}$, if the literal l_k^j is false then place t_k^j in $S_1 \cup S_2$; otherwise, place t_k^j in K. Since the truth assignment satisfies (X, C), for each j, we have at most two vertices t_k^j in $S_1 \cup S_2$. In addition, in case two vertices t_k^j and t_p^j are placed in $S_1 \cup S_2$, place one in S_1 and the other one in S_2.

To show that (S_1, S_2, K) is a (2,1) partition for a sandwich graph $G = (V, E)$ we need to prove that there is no E^1 edge with both endnodes in S_1, there is no E^1 edge with both endnodes in S_2 and there is no E^3 edge with both endnodes in K.

By the above placement, s_{11}, s_{21} are in S_1, and $\overline{x_i}$, t_k^j and p_i can be in S_1, $i \in \{1, \dots, n\}$, $j \in \{1, \dots, m\}$, $k \in \{1, 2, 3\}$. The only possible forced edges between these vertices are: the edge $\overline{x_i} p_i$ which does not have both endnodes in S_1, because $\overline{x_i} \in S_1$ if the variable x_i is false and $p_i \in S_1$ if x_i is true; and the edge $t_k^j t_q^j$ which does not have both endnodes in S_1, $k \neq q, k, q \in \{1, 2, 3\}$. Hence, there is no E^1 edge with both endnodes in S_1.

In the same way, s_{12}, s_{22} are in S_2, and the vertices x_i, t_k^j, p_i can be in S_2, $i \in \{1, \dots, n\}$, $j \in \{1, \dots, m\}$, $k \in \{1, 2, 3\}$. The only possible forced edges between these vertices are: the edge $x_i p_i$ which does not have both endnodes in S_2, because $x_i \in S_2$ if the variable x_i is true and $p_i \in S_1$ if x_i is false; and the edge $t_k^j t_q^j$ which does not have both endnodes in S_1, $k \neq q$, $k, q \in \{1, 2, 3\}$. Hence, there is no E^1 edge with both endnodes in S_2.

For the set K we have that k_1, k_2 are in K, and the vertices x_i, $\overline{x_i}$, t_k^j can be in K, $i \in \{1, \dots, n\}$, $j \in \{1, \dots, m\}$, $k \in \{1, 2, 3\}$. The only possible forbidden edges between these vertices are: the edge $\overline{x_i} x_i$ which does not have both endnodes in K, because $\overline{x_i} \in K$ if and only if the variable x_i is true and $x_i \in K$ if and only if x_i is false; and the edges

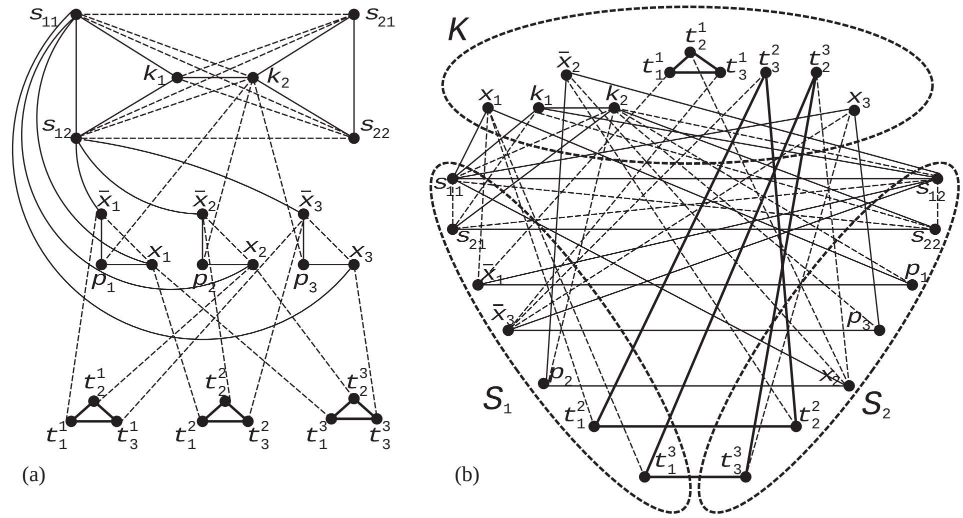

Fig. 2. (a) Instance (V,E^1,E^3) obtained from the satisfiable instance of 3SAT: $I=(U,C)=(\{x_1,x_2,x_3\},\{(\overline{x_1}\vee x_2\vee\overline{x_3}),(x_1\vee\overline{x_2}\vee\overline{x_3}),(x_1\vee x_2\vee x_3)\})$ and (b) respective partition for the $(2,1)$ graph G defined from the satisfying truth assignment $x_1=F$, $x_2=T$, $x_3=F$.

$x_it_k^j, \overline{x_i}t_k^j$, by the above placement, we never have both vertices in K. Hence, there is no E^3 edge with both endnodes in K. And this ends the proof of Lemma 10. In Fig. 2 we give an example of the constructed instance (V,E^1,E^3) for (2,1)-GRAPH SANDWICH PROBLEM, and of the (2,1) graph G given by the proof of Lemma 10. □

3. (2, 2)-Graph Sandwich Problem

In this section we prove that the $(2,2)$-GRAPH SANDWICH PROBLEM is NP-complete by reducing the NP-complete problem 3SAT to $(2,2)$-GRAPH SANDWICH PROBLEM.

$(2,2)$-GRAPH SANDWICH PROBLEM
Instance: Vertex set V, forced edge set E^1, forbidden edge set E^3.
Question: Is there a graph $G=(V,E)$, such that $E^1\subseteq E$ and $E\cap E^3=\emptyset$, and G is $(2,2)$?

Theorem 11. *The* $(2,2)$-GRAPH SANDWICH PROBLEM *is* NP-*complete*.

Proof. In order to reduce 3SAT to $(2,2)$-GRAPH SANDWICH PROBLEM we need to construct a particular instance (V,E^1,E^3) of $(2,2)$-GRAPH SANDWICH PROBLEM from a generic instance (X,C) of 3SAT, such that C is satisfiable if and only if (V,E^1,E^3) admits a sandwich graph $G=(V,E)$ which is $(2,2)$. □

3.1. Construction of particular instance of $(2,2)$-GRAPH SANDWICH PROBLEM

The vertex set V contains: auxiliary sets of vertices $B_1=\{k_1,k_2,s_{11},s_{12},s_{21},s_{22}\}$ and $B_2=\{k_3,k_4,s_{31},s_{32},s_{41},s_{42}\}$; for each variable x_i, $1\leqslant i\leqslant n$, two vertices x_i, $\overline{x_i}$, corresponding to its literals and a vertex p_i; for each clause $c_j=(l_1^j\vee l_2^j\vee l_3^j)$, $1\leqslant j\leqslant m$, three corresponding vertices t_1^j, t_2^j, t_3^j. See Fig. 1, where solid edges denote forced E^1-edges and dashed edges denote forbidden E^3-edges.

The Forced Edge Set E^1 contains: sets of edges between auxiliary vertices $F_1=\{k_1k_2, k_1s_{11}, k_1s_{12}, s_{11}s_{12}, k_2s_{21}, k_2s_{22}, s_{21}s_{22}\}$ and $F_2=\{k_3k_4, k_3s_{31}, k_3s_{32}, s_{31}s_{32}, k_4s_{41}, k_4s_{42}, s_{41}s_{42}\}$; for each variable x_i, $1\leqslant i\leqslant n$, the set $\{x_is_{11}, \overline{x_i}s_{12}, x_ip_i, \overline{x_i}p_i\}$; for each clause c_j, $1\leqslant j\leqslant m$, the set $\{t_1^jt_2^j, t_1^jt_3^j, t_2^jt_3^j\}$.

The Forbidden Edge Set E^3 contains: sets of edges incident to auxiliary vertices $F_3=\{k_1s_{21}, k_1s_{22}, k_2s_{11}, k_2s_{12}, s_{11}s_{21}, s_{11}s_{22}, s_{12}s_{21}, s_{12}s_{22}\}$, $F_4=\{k_3s_{41}, k_3s_{42}, k_4s_{31}, k_4s_{32}, s_{31}s_{41}, s_{31}s_{42}, s_{32}s_{41}, s_{32}s_{42}\}$, $F_5=\{uv\colon u\in B_1 \textit{ and } v\in B_2\}$, $F_6=\{uv\colon u\in B_2 \textit{ and } v\in V\setminus(B_1\cup B_2)\}$; for each variable x_i, $1\leqslant i\leqslant n$, the set $\{x_i\overline{x_i}, p_ik_2\}$; for each clause c_j, $1\leqslant j\leqslant m$, $\{t_1^jl_1^j, t_2^jl_2^j, t_3^jl_3^j\}$.

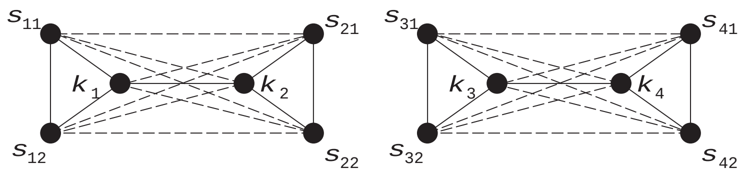

Fig. 3. (2,2) Base graph—all non-represented edges are E^3 edges.

Call (2,2) *Base graph* the subgraph of $G^2=(V,E^2)$ induced by $\{k_1, k_2, s_{11}, s_{12}, s_{21}, s_{22}, k_3, k_4, s_{31}, s_{32}, s_{41}, s_{42}\}$ (see Fig. 3). As in the previous problem, we have two kinds of gadgets: *Variable gadget* (Fig. 1(VG)) and *clause gadget* (Fig. 1(CG)). The special instance has a property similar to Theorem 5: if the particular instance (V,E^1,E^3) admits a (2,2) sandwich graph $G=(V,E)$, then any (2,2) partition (S_1,S_2,K_1,K_2) for G satisfies $S_1,S_2,K_1,K_2 \neq \emptyset$. Without loss of generality assume $k_1,k_2 \in K_1$, $k_3,k_4 \in K_2$, $s_{11},s_{21},s_{31},s_{41} \in S_1$, $s_{12},s_{22},s_{32},s_{42} \in S_2$. This implies $K_2=\{k_3,k_4\}$.

The proof now follows from Theorem 5 and the equivalence: the particular instance (V,E^1,E^3) admits a (2,2) sandwich graph $G=(V,E)$ with (2,2) partition (S_1,S_2,K_1,K_2) if and only if the particular instance $(V-B_2,E^1-F_2,E^3-(F_4\cup F_5\cup F_6))$ admits a (2,1) sandwich graph with (2,1) partition (S_1,S_2,K).

4. (*k*, *l*)-bounded *Δ* Graph Sandwich Problem

In this section, we consider the complexity of the (k,l)-Graph Sandwich problem when restricted to inputs having G^2 with bounded maximum degree.

(k,l)-Bounded Δ Graph Sandwich Problem ($(k,l)-B\Delta GSP$)

Instance: Vertex set V, forced edge set E^1, forbidden edge set E^3, where G^2 is a graph with no vertex degree exceeding Δ.

Question: Is there a graph $G=(V,E)$ such that $E^1\subseteq E$ and $E\cap E^3=\emptyset$ which is a (k,l) graph?

We completely classify the $(k,l)-B\Delta GSP$ as follows: $(k,l)-B\Delta GSP$ is polynomial for $k\leqslant 2$ or $\Delta\leqslant 3$, and NP-complete otherwise.

Lemma 12. *If $(k,l)-B\Delta GSP$ is solvable in polynomial time then the $(k,l+1)-B\Delta GSP$ is solvable in polynomial time.*

Proof. Let (V,E^1,E^3) be an instance for $(k,l+1)-B\Delta GSP$. Suppose that there exists a polynomial time algorithm A to solve the $(k,l)-B\Delta GSP$. We observe that if there exists a sandwich graph for (V,E^1,E^3) which is $(k,l+1)$ then a clique in G is also a clique in G^2. Thus, in order to define a polynomial time algorithm for $(k,l+1)-B\Delta GSP$ we proceed as follows: for each subset S with less than or equal to $\Delta+1$ vertices we verify if S induces a clique in G^2. In the affirmative case we apply the algorithm A to test if there exists a sandwich graph for the instance $(V\setminus S,E^1,E^3)$ of $(k,l)-B\Delta GSP$. Hence, we have designed an algorithm for $(k,l+1)-B\Delta GSP$ which runs in time $O(n^{\Delta+1}P)$, where P is the order of the algorithm $\mathscr{A}$. □

Lemma 13. *If $k\leqslant 2$, then $(k,l)-B\Delta GSP$ is solvable in polynomial time.*

Proof. We argue by induction on l. As we said in the Introduction the (1,0) and (2,0)-Graph Sandwich Problems are solvable in polynomial time, so are the corresponding problems $B\Delta GSP$. Suppose that for $k\leqslant 2$ and $l\geqslant 0$ the $(k,l)-B\Delta GSP$ is solvable in polynomial time. By Lemma 12 we have that the corresponding $(k,l+1)-B\Delta GSP$ is a polynomial time problem. □

Now, consider $k\geqslant 3$. Note that, as a consequence of Brook's theorem [5], $(k,0)$ graph recognition is polynomial when restricted to inputs having $\Delta\leqslant 3$. This implies by Fact 2 that $(k,0)-B3GSP$ is solvable in polynomial time, and by Lemma 12, $(k,l)-B3GSP$ is also polynomial. However, by Garey et al. [9], $(k,0)$ graph recognition is NP-complete, even when restricted to inputs having $\Delta\leqslant 4$, which implies by Fact 2 that $(k,0)-B\Delta GSP$ is NP-complete, and as remarked in [2,3], $(k,l)-B\Delta GSP$ is NP-complete, for $\Delta\geqslant 4$.

5. Optimization versions of Graph Sandwich Problem for a Property Π

In this section we consider two natural extensions for the decision version of Graph Sandwich Problem for a Property Π.

Let (V, E^1, E^2) be an instance of Graph Sandwich Decision Problem for a Property Π. Let V' be a subset of V; and $E^1[V']$ and $E^2[V']$ denote, respectively, the subsets of E^1 and E^2 satisfying $E^1[V'] = \{uv \in E^1\colon u \in V' \text{ and } v \in V'\}$ and $E^2[V'] = \{uv \in E^2\colon u \in V' \text{ and } v \in V'\}$.

The proposed Optimization Versions consist in considering the maximum subset $V \setminus V'$ of V, such that the graph sandwich instance $(V \setminus V', E^1[V \setminus V'], E^2[V \setminus V'])$ for property Π has answer YES.

Note that with respect to ε-approximative algorithms for optimization problems a same decision problem can yield different optimization problems according to whether we are considering maximization or minimization versions. Next we formally define both a minimization and a maximization version for Graph Sandwich Problem for a Property Π.

Version 1. MINIMIZATION VERSION OF GRAPH SANDWICH PROBLEM FOR A PROPERTY Π (MIN-Π-GSP).
Instance: (V, E^1, E^2).
Goal: Minimize $|V'|, V' \subseteq V$, such that $(V \setminus V', E^1[V \setminus V'], E^2[V \setminus V'])$ has a sandwich graph satisfying property Π.

Version 2. MAXIMIZATION VERSION OF GRAPH SANDWICH PROBLEM FOR A PROPERTY Π (MAX-Π-GSP).
Instance: (V, E^1, E^2).
Goal: Maximize $|V'|, V' \subseteq V$, such that $(V', E^1[V'], E^2[V'])$ has a sandwich graph satisfying property Π.

We denote the optimum value for the problems MIN-Π-GSP and MAX-Π-GSP by writing, respectively, $Opt_{\text{min-}\Pi\text{-gsp}}(V, E^1, E^2)$ and $Opt_{\text{max-}\Pi\text{-gsp}}(V, E^1, E^2)$.

We observe that both optimization problems are not constrained to the property (k, l) graphs.

Next, we prove that the minimization problem MIN-$(2,1)$-GSP corresponding to the Version 1 of graph sandwich problem for the property $(2,1)$-graphs is Max SNP-hard, i.e., there is an $\varepsilon > 0$, such that the existence of a ε-approximative algorithm for this problem implies P=NP [1,16].

6. The Max SNP-hardness of MIN-(2, 1)-GSP

In order to establish that MIN-$(2,1)$-GSP is Max SNP-hard, we use the concept of L-*reductions* of Papadimitriou and Yannakakis [16], a special kind of reduction that preserves approximability. Let A and B be two optimization problems. We say that A L-*reduces* to B if there are two polynomial-time algorithms f and g and positive constants α and β, such that for each instance I of A,

(1) Algorithm f produces an instance $I' = f(I)$ of B such that the optima of I and I' satisfy $Opt_B(I') \leqslant \alpha . Opt_A(I)$;
(2) Given any feasible solution of I' with cost c', algorithm g produces a feasible solution of I with cost c such that $|c - Opt_A(I)| \leqslant \beta . |c' - Opt_B(I')|$.

To prove that an optimization problem is Max SNP-hard, we need to L-reduce a suitable Max SNP-hard problem to it. We prove that MIN-$(2,1)$-GSP is Max SNP-hard by L-reducing to it the optimization version of a suitable restriction of MAX3SAT. We have to strengthen the NP-completeness proof presented in Section 2, by considering the Max SNP-complete problem $MAX3SAT_{\bar{3}}$ defined as follows [16]:

$MAX3SAT_{\bar{3}}$
Instance: Set U of variables, collection C of clauses over U such that each clause $c \in C$ has $|c| = 3$ literals, and each variable appears at most three times in the set of clauses.
Goal: Find a truth assignment for U which maximizes the number of clauses in C having at least one true literal.

6.1. The algorithm f

Let $I = (U, C)$ be an instance of $MAX3SAT_{\bar{3}}$, where U has n variables and C has m clauses. Algorithm f produces, in polynomial time on the size of I, an instance (V, E^1, E^3) of MIN-$(2,1)$-GSP. The instance (V, E^1, E^3) is quite similar to the NP-complete instance of $(2,1)$ decision version presented in Section 2. Roughly speaking, the Clause graphs are the same and the Base graph and the Variable graphs have their vertices replaced either by complete graphs or by independent sets. In addition, an edge between two vertices is replaced by the set of all edges between the corresponding sets of vertices.

Next, we precisely define the instance (V, E^1, E^3).

6.1.1. The set V

We have six auxiliary sets of vertices k_1,k_2,s_{11},s_{12}, s_{21} and s_{22}, each containing $3n$ vertices; for each variable x_i, $1 \leqslant i \leqslant n$, we have three sets of vertices x_i, $\overline{x_i}$ and p_i, each containing $3n$ vertices; for each clause $c_i = (l_1^j \vee l_2^j \vee l_3^j)$, $1 \leqslant j \leqslant m$, we have three vertices t_1^j, t_2^j and t_3^j.

For each $j \in \{1,\dots,m\}$, we call *Clause gadget* the subgraph of $G^2 = (V, E^2)$ induced by $\{t_1^j, t_2^j, t_3^j\}$.

6.1.2. The sets E^1 and E^3

The Forced Edge Set E^1 contains the following edges:

(1) If $u, v \in V(k_1)$, then $uv \in E^1$,
(2) If $u, v \in V(k_2)$, then $uv \in E^1$,
(3) If $u \in V(k_1)$ and $v \in V(k_2)$, then $uv \in E^1$,
(4) If $u \in V(k_1)$ and $v \in V(s_{11})$ or $v \in V(s_{12})$, then $uv \in E^1$,
(5) If $u \in V(s_{11})$ and $v \in V(s_{12})$, then $uv \in E^1$,
(6) If $u \in V(k_2)$ and $v \in V(s_{21})$ or $v \in V(s_{22})$, then $uv \in E^1$,
(7) If $u \in V(s_{21})$ and $v \in V(s_{22})$, then $uv \in E^1$,
(8) For all $i \in \{1,2,3,\dots,n\}$, if $u \in V(x_i)$ and $v \in V(p_i)$, then $uv \in E^1$,
(9) For all $i \in \{1,2,3,\dots,n\}$, if $u \in V(\overline{x_i})$ and $v \in V(p_i)$, then $uv \in E^1$,
(10) If u, v are vertices of a Clause Gadget, then $uv \in E^1$.

The Forbidden Edge Set E^3 contains the following edges:

(1) If $u, v \in V(s_{11})$, then $uv \in E^3$,
(2) If $u, v \in V(s_{12})$, then $uv \in E^3$,
(3) If $u, v \in V(s_{21})$, then $uv \in E^3$,
(4) If $u, v \in V(s_{22})$, then $uv \in E^3$,
(5) For all $i \in \{1,2,3,\dots,n\}$, if $u, v \in V(p_i)$, then $uv \in E^3$,
(6) For all $i \in \{1,2,3,\dots,n\}$, if $u \in V(k_1)$ and $v \in V(p_i)$, then $uv \in E^3$,
(7) If $u \in V(s_{11})$ or $u \in V(s_{12})$ and $v \in V(s_{21})$ or $v \in V(s_{22})$, then $uv \in E^3$,
(8) For all $i \in \{1,2,3,\dots,n\}$, if $u \in V(x_i)$ and $v \in V(\overline{x_i})$, then $uv \in E^3$,
(9) For all $j \in \{1,2,3,\dots,m\}$, if clause $c_j = (l_1^j \vee l_2^j \vee l_3^j) \in C$, then for all $x \in V(l_1^j)$ $t_1^j x \in E^3$, for all $x \in V(l_2^j)$ $t_2^j x \in E^3$ and for all $x \in V(l_3^j)$ $t_3^j x \in E^3$.

Note that, we have now as base graph the subgraph of G^2 induced by the vertices in the sets $k_1, k_2, s_{11}, s_{12}, s_{21}$ and s_{22}. Note that, k_1 and k_2 induce complete graphs in G^1 and s_{11}, s_{12}, s_{21} and s_{22} are independent sets in G^3. For each $i \in \{1,2,3,\dots,n\}$, the variable gadget is now a subgraph of G^2 induced by the vertices in the sets x_i, $\overline{x_i}$ and p_i, where x_i and $\overline{x_i}$ are independent sets in G^1 but complete subgraphs of G^2, whereas p_i is an independent set in G^3. For each $j \in \{1,2,3,\dots,m\}$, the Clause gadget, a triangle of G^1 induced by t_1^j, t_2^j, t_3^j, remains unchanged.

Lemma 14. *If $I = (U, C)$ is an instance of $MAX3SAT_{\bar{3}}$ with n variables and m clauses, then*

$$\left\lceil \frac{n}{3} \right\rceil \leqslant Opt_{\mathrm{max3sat}_{\bar{3}}}(I) \leqslant m \leqslant 3n.$$

Proof. Consider $I = (U, C)$ an instance of $MAX3SAT_{\bar{3}}$ with $|U| = n$ and $|C| = m$. Since each variable must have at least one of its two literals occurring in the set of clauses, and since each clause has size 3, the number m of clauses satisfies $3m \geqslant n$. Hence m is limited below by the least integer greater or equal to $n/3$. On the other hand, since each variable occurs at most 3 times in the set of clauses, the number m of clauses satisfies $3m \leqslant 3n$. Therefore we have the inequalities $\lceil n/3 \rceil \leqslant m \leqslant n$, as required.

Now in order to establish the claimed bounds for $Opt_{\mathrm{max3sat}_{\bar{3}}}(I)$, note first that $Opt_{\mathrm{max3sat}_{\bar{3}}}(I) \leqslant m$. Now to establish the claimed lower bound, it is enough to exhibit a truth assignment for I with $\lceil n/3 \rceil$ satisfied clauses. For each variable $u_i \in U$, $i \in \{1,2,\dots,n\}$, set $u_i = T$, if and only if its positive literal occurs in C. Note that this truth assignment for U can be defined in time polynomial in the size of I. Now to each variable u_i we have a corresponding literal x_i occurring in C with value true. Let k be the minimum number of clauses that fit those n literals with value true. Since each clause has size 3, integer k is the least integer satisfying $3k \geqslant n$, i.e., $k = \lceil n/3 \rceil$ is the least integer greater than or equal to $n/3$. Hence, we have at least $\lceil n/3 \rceil$ satisfied clauses, and we have the inequalities $\lceil n/3 \rceil \leqslant Opt_{\text{max3sat}_{\bar{3}}}(I) \leqslant m$, as required. □

Lemma 15. *Let $I = (U, C)$ be a $MAX3SAT_{\bar{3}}$ instance, (V, E^1, E^2) be the MIN-$(2,1)$-GSP instance yielded by algorithm f from $I = (U, C)$ and η_I be a feasible solution of $MAX3SAT_{\bar{3}}$ for $I(U, C)$. Then, there is a feasible solution ϕ_V of MIN-$(2,1)$-GSP for (V, E^1, E^2) produced from η_I in polynomial time in the size of I, such that, $|\phi_V| = m - |\eta_I|$.*

Proof. Consider a truth assignment η_I for $I = (U, C)$. Let $C' \subseteq C$ the subset of satisfied clauses of C. Then, the truth assignment η_I is a satisfiable truth assignment for the instance $I' = (U, C')$, with $|\eta_I| = |C'|$. Hence, we can construct $V' = \phi_V$, a feasible solution of MIN-$(2,1)$-GSP for (V, E^1, E^2), by selecting one vertex of each clause graph corresponding to each one of the $m - |\eta_I|$ non satisfied clauses. This solution is feasible: since η_I is a satisfiable truth assignment for the instance $I' = (U, C')$, we can place all the vertices of $V(k_1)$ and $V(k_2)$ in K, place all the vertices of $V(s_{11})$ and $V(s_{21})$ in S_1 and place all the vertices of $V(s_{12})$ and $V(s_{22})$ in S_2. In addition, for $i \in \{1, \ldots, n\}$, if x_i is false then place all the vertices of $V(x_i)$ in K, all the vertices of $V(\overline{x_i})$ in S_1, and all the vertices of $V(p_i)$ in S_2. Otherwise, if x_i is true, then place all the vertices of $V(x_i)$ in S_2, all the vertices of $V(\overline{x_i})$ in K, and all the vertices of $V(p_i)$ in S_1. For every true clause of C the vertices of the corresponding Clause graph can be placed one in K and the other 2 vertices one into each of the 2 independent sets. As for each non satisfied clause of C we remove one vertex of the corresponding Clause gadget, we can place the remaining 2 vertices one in S_1 and the other in S_2. Thus, ϕ_V is feasible and $|\phi_V| = m - |\eta_I|$. □

Claim 16. *Let $I = (U, C)$ be a $MAX3SAT_{\bar{3}}$ instance, (V, E^1, E^2) be the MIN-$(2,1)$-GSP instance yielded by algorithm f from $I = (U, C)$ and ϕ_V be a feasible solution of MIN-$(2,1)$-GSP for (V, E^1, E^2), where $|\phi_V| < 3n$ and S_1, S_2, K be the $(2,1)$-partition for the sandwich graph defined by the removal of ϕ_V from (V, E^1, E^2). Then the truth assignment η_I, where the variable $x_i = F$ if and only if there is a vertex of the set x_i in K is well defined.*

Proof. First of all, note that as $|\phi_V| < 3n$, there are vertices of $x_i, \overline{x_i}$ and p_i that are not removed by ϕ_V. It is enough to prove that if a non-removed vertex of the set x_i is placed in K, then all non-removed vertices of the set are also placed in K. Suppose there are two non-removed vertices u, v of the set x_i, such that u is placed in K and v is not placed in K. By Claim 8, vertex v is placed in S_2, and then all vertices of the set p_i are placed in S_1, because each one of them is adjacent to v. Now Claim 8 says all non-removed vertices of the set $\overline{x_i}$ must be placed in $K \cup S_1$, a contradiction because each one of them is non adjacent to u and each one of them is adjacent to the vertices of p_i. □

Lemma 17. *Let $I = (U, C)$ be a $MAX3SAT_{\bar{3}}$ instance, (V, E^1, E^2) be the MIN-$(2,1)$-GSP instance yielded by algorithm f from $I = (U, C)$ and ϕ_V be a feasible solution of MIN-$(2,1)$-GSP for (V, E^1, E^2). Then, there is a feasible solution η_I of $MAX3SAT_{\bar{3}}$ for $I(U, C)$ produced from ϕ_V in polynomial time in the size of I, such that, $|\eta_I| \geqslant m - |\phi_V|$.*

Proof. By Lemma 14, we have that $m \leqslant 3n$. Consider a feasible solution ϕ_V of MIN-$(2,1)$-GSP for (V, E^1, E^2). We consider two cases, according to $|\phi_V| \geqslant 3n$ or $|\phi_V| < 3n$.

In the first case, when $|\phi_V| \geqslant 3n$, which implies $m - |\phi_V| \leqslant 0$. Hence, for any truth assignment η_I for $I(U, C)$ the inequality $|\eta_I| \geqslant m - |\phi_V|$ clearly holds.

In the second case, when $|\phi_V| < 3n$, as ϕ_V is feasible, and each one of the graphs $k_1, k_2, s_{11}, s_{12}, s_{21}$ and s_{22}, and $x_i, \overline{x_i}$ and p_i has $3n$ vertices, there is at least one vertex of each of them out of ϕ_V. So the non-removed vertices of the literal graphs x_i and $\overline{x_i}$, according Claim 16 define a truth assignment η_I where $c = |\eta_I|$ clauses are satisfied. This truth assignment forces that each one of the $m - c$ non satisfiable clauses requires one additional vertex of the corresponding Clause graph in ϕ_V. Hence, $|\phi_V| \geqslant m - c = m - |\eta_I|$. □

Theorem 18 (fundamental property). $Opt_{\text{min-}(2,1)\text{-gsp}}(V, E^1, E^2) = m - Opt_{\text{max3sat}_{\bar{3}}}(I)$.

Proof. By Lemma 17, there is a truth assignment η_I to $I = (U, C)$, such that we have $Opt_{\text{min-}(2,1)\text{-gsp}}(V, E^1, E^2) \geqslant m - |\eta_I|$. Since, $MAX3SAT_{\bar{3}}$ is a maximization problem, $m - |\eta_I| \geqslant m - Opt_{\text{max3sat}_{\bar{3}}}(I)$, which gives the inequality $Opt_{\text{min-}(2,1)\text{-gsp}}(V, E^1, E^2) \geqslant m - Opt_{\text{max3sat}_{\bar{3}}}(I)$. By Lemma 15, there is a feasible solution ϕ_V of MIN-$(2,1)$-GSP for (V, E^1, E^2) such that, $Opt_{\text{max3sat}_{\bar{3}}}(I) = m - |\phi_V|$. As MIN-$(2,1)$-GSP is a minimization problem, we have that $m - \phi_V \leqslant Opt_{\text{min-}(2,1)\text{-gsp}}(V, E^1, E^2)$, which gives the inequality $Opt_{\text{max3sat}_{\bar{3}}}(I) \leqslant m - Opt_{\text{min-}(2,1)\text{-gsp}}(V, E^1, E^2)$. Thus, we conclude that $Opt_{\text{min-}(2,1)\text{-gsp}}(V, E^1, E^2) = m - Opt_{\text{max3sat}_{\bar{3}}}(I)$. □

Corollary 19. *MIN-$(2,1)$-GSP is Max* SNP-*hard*.

Proof. We L-reduce $MAX3SAT_{\bar{3}}$ to MIN-$(2,1)$-GSP. We start the L-reduction by setting algorithm f defined in Section 6.1 to be the algorithm f of the L-reduction. By Lemma 14 and Fundamental Theorem 18, we have that $Opt_{\text{min-}(2,1)\text{-gsp}}(V,E^1,E^2) = m - Opt_{\text{max3sat}_{\bar{3}}}(I) \leqslant 3n - \frac{n}{3} = \frac{8n}{3} \leqslant 8Opt_{\text{max3sat}_{\bar{3}}}(I)$, Hence, we can set $\alpha = 8$.

Given a feasible solution ϕ_V of MIN-$(2,1)$-GSP for (V,E^1,E^2), Lemma 17 says that a feasible solution η_I of $I(U,C)$ can be produced from ϕ_V in polynomial time in the size of I, such that, $|\eta_I| \geqslant m - |\phi_V|$. Hence, $|Opt_{\text{max3sat}_{\bar{3}}}(I) - |\eta_I|| \leqslant |m - Opt_{\text{min-}(2,1)\text{-gsp}}(V,E^1,E^2) - (m - |\phi_V|)| = ||\phi_V| - Opt_{\text{min-}(2,1)\text{-gsp}}(V,E^1,E^2)|$, showing that $\beta = 1$ suffices. □

7. Conclusion

We proved that the (k,l)-Graph Sandwich Problem is NP-complete for the cases $k=1$ and $l=2$; $k=2$ and $l=1$; or $k=l=2$. We note that the basic idea of the construction of the particular instance of these problems is a simple necessary condition: if a graph is (k,l) then it does not contain $l+1$ independent cliques of size $k+1$. Recently, this condition was established sufficient for the class of the Chordal graphs, as proved by Hell et al. [13]. In addition, we considered the degree Δ constraint subproblem $(k,l)-B\Delta GSP$ and completely classified the problem as follows: $(k,l)-B\Delta GSP$ is a polynomial problem for $k \leqslant 2$ or $\Delta \leqslant 3$; and NP-complete otherwise.

We observe that it is not possible to use our L-reduction transformation to prove that the maximization version for Graph Sandwich Problem for the property (k,l) graphs (MAX-$(2,1)$-GSP) is a Max SNP-hard problem. Note that, in our transformation we have an instance $I=(U,C)$ with n variables and each variable corresponds to $9n$ additional vertices in the corresponding instance of (V,E^1,E^2). The structure of (V,E^1,E^2) implies the existence of a feasible solution with at least $9n^2$ vertices. Hence, we have an optimum value of MAX-$(2,1)$-GSP for (V,E^1,E^2) of order $\Omega(n^2)$. As $Opt_{\text{max3sat}_{\bar{3}}}(I)=O(n)$, it is impossible to define a positive constant α for the first L-reduction inequality: $Opt_{\text{min-}(2,1)\text{-gsp}}(V,E^1,E^2) \leqslant \alpha Opt_{\text{max3sat}_{\bar{3}}}(I)$. We conclude by noting that analogous arguments show that MIN-(1,2)-GSP and MIN-(2,2)-GSP are Max SNP-hard problems.

Acknowledgements

This research was partially supported by CNPq, MCT/FINEP PRONEX Project 107/97, CAPES (Brazil)/COFECUB (France) Project number 213/97, FAPERJ and PROCIÊNCIA-UERJ Project. An extended abstract of this paper was published in the Proceedings of the *28th International Workshop on Graph Theoretical Concepts in Computer Science* —WG'2002—Cesky Krumlov—Czech Republic.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy, Proof verification and hardness of approximation problems, in: Proceedings of the IEEE Symposium on Foundations of Computer Science—FOCS'92, 1992 pp. 14–23.
[2] A. Brandstädt, Partitions of graphs into one or two independent sets and cliques, Report, Vol. 105, Informatik Berichte, Fern Universität, Hagen, Germany, 1991.
[3] A. Brandstädt, Partitions of graphs into one or two independent sets and cliques, Discrete Math. 152 (1996) 47–54 (see also Corrigendum 186 (1998) 295).
[4] A. Brandstädt, V.B. Le, T. Szymczak, The complexity of some problems related to graph 3-colorability, Discrete Appl. Math. 89 (1998) 59–73.
[5] R.L. Brooks, On coloring the nodes of a network, Proc. Cambridge Philos. Soc. 37 (1941) 194–197.
[6] M. Cerioli, H. Everett, C.M.H. de Figueiredo, S. Klein, The homogeneous set sandwich problem, Inform. Process. Lett. 67 (1998) 31–35.
[7] T. Feder, P. Hell, S. Klein, R. Motwani, Complexity of graph partition problems, in: Proceedings of the 31st Annual ACM Symposium on Theory of Computing—STOC'99, 1999, pp. 464–472.
[8] C.M.H. de Figueiredo, S. Klein, K. Vušković, The graph sandwich problem for 1-join composition is NP-complete, Discrete Appl. Math. 121 (2002) 73–82.
[9] M.R. Garey, D.S. Johnson, L. Stockmeyer, Some simplified NP-complete graph problems, Theoret. Comput. Sci. 1 (1976) 237–267.
[10] M.C. Golumbic, Matrix sandwich problems, Linear Algebra Appl. 277 (1998) 239–251.
[11] M.C. Golumbic, H. Kaplan, R. Shamir, Graph sandwich problems, J. Algorithms 19 (1995) 449–473.
[12] M.C. Golumbic, A. Wassermann, Complexity and algorithms for graph and hypergraph sandwich problems, Graphs Combin. 14 (1998) 223–239.

[13] P. Hell, S. Klein, F. Protti, L. Tito, On generalized split graphs, in: Brazilian Symposium on Graphs, Algorithms and Combinatorics, Electronic Notes in Discrete Mathematics, Vol. 7, 2001.
[14] H. Kaplan, R. Shamir, Bounded degree interval sandwich problems, Algorithmica 24 (1999) 96–104.
[15] H. Kaplan, R. Shamir, Pathwidth, bandwidth, and completion problems to proper interval graphs with small cliques, SIAM J. Comput. 25 (1996) 540–561.
[16] C.H. Papadimitriou, M. Yannakakis, Optimization, approximation, and complexity classes, J. Comput. System Sci. 43 (1991) 425–440.