The Graph Sandwich Problem for a **coNP** property

Marcus Schaefer

School of CTI
DePaul University
243 South Wabash Avenue
Chicago, Illinois 60604, USA
schaefer@cs.depaul.edu

July 2, 2006

Abstract

If $F \subseteq G \subseteq F'$ we say that the graph G is sandwiched between graphs F and F'. The graph sandwich problem consists in finding such a graph G in a particular class or with a special property. We show that there is a **coNP**-complete graph property for which the graph sandwich problem is $\mathbf{\Sigma_2^p}$-complete.

1 Introduction

We say a graph G is *sandwiched* between two graphs F and F' if $F \subseteq G \subseteq F'$. Given any graph property Π, Golumbic, Kaplan and Shamir [GKS95] asked how hard it is to find a graph G with property Π sandwiched between two given graphs F and F'.

GRAPH SANDWICH PROBLEM FOR Π

Given: Graphs F and F' such that $F \subseteq F'$

Question: Is there a graph G satisfying Π so that $F \subseteq G \subseteq F'$?

If Π is an **NP**-complete graph property such as being Hamiltonian, then the graph sandwich problem for Π is also **NP**-complete. If Π can be checked in polynomial time, then the corresponding sandwich problem lies in **NP** and can range from being in **P** itself (as in the case of split graphs) to being **NP**-complete (for chordal graphs) [GKS95]. Obviously, if Π is closed downward under taking subgraphs or closed upward under taking supergraphs, then deciding the graph sandwich problem for Π has the same complexity as deciding Π. The original paper by Golumbic, *et al.* settled the complexity for many properties stronger than being a perfect graph. Recent work by Figueiredo, Faria, Klein, Sritharan [FFKS06] settled some of the open problems left by that paper (including the cases of strongly chordal graphs and chordal bipartite graphs), but there are still many open problems, including the case of perfect graphs.

Intriguingly, there seems to have been no study yet of properties not known to be in **NP**, for example **coNP**-complete properties. For a property Π in **coNP**, the graph sandwich problem lies in $\mathbf{\Sigma_2^p} = \mathbf{NP^{NP}}$, the second level of the polynomial time hierarchy (for details, see [GJ79, SU02]). In this paper we show that for a reasonably natural **coNP**-complete property (not containing long induced paths), the sandwich problem is complete for $\mathbf{\Sigma_2^p}$.

2 Hardness of the Graph Sandwich Problem

As the candidate for Π we have selected the property of being P_k-free, where a graph is *H-free* if it does not contain an *induced* copy of H. P_k is the path on k vertices (of length $k-1$). Being P_k-free is **coNP**-complete (something we will show along the way) and our claim is that the graph sandwich problem is $\mathbf{\Sigma_2^p}$-complete for this property.

Before proving the $\mathbf{\Sigma_2^p}$-completeness result we first show that the problem is **coNP**-hard to illustrate the construction.

Suppose we are given a formula $(\forall y)[\varphi(y)]$, where $\varphi(y)$ is in disjunctive normal form: $\varphi(y) = C_1(y) \vee C_2(y) \vee \ldots \vee C_m(y)$ where each clause $C_j(y)$ contains 3 literals (this problem is **coNP**-complete since it is the negation of 3SAT).

The construction is a traditional gadget construction, building a variable and a clause gadget. We first construct the variable gadget V: for every variable y_i in φ $(1 \leq i \leq n)$ take a $K_4 - e$, label the two vertices of degree 3 with y_i and $\overline{y}_i$ and let u_i and v_i be the two vertices of degree 2. Connect the $K_4 - e$s by adding an edge from v_i to u_{i+1} for $1 \leq i < n$. Call the resulting graph C.

To construct the clause gadget C, take a $K_5 - e$ for each clause C_j $(1 \leq j \leq m)$, label the three vertices of degree 4 with the literals in the clause and let the two vertices of degree 3 be called s_j and t_j. Connect the $K_5 - e$s by adding an edge from t_j to s_{j+1} for $1 \leq j < m$. Call the resulting graph C.

From V and C we build a graph H as follows: add an edge between a vertex from V and a vertex from C if they have the same label. Finally, add an edge from v_n to s_1. See Figure 1 for an example.

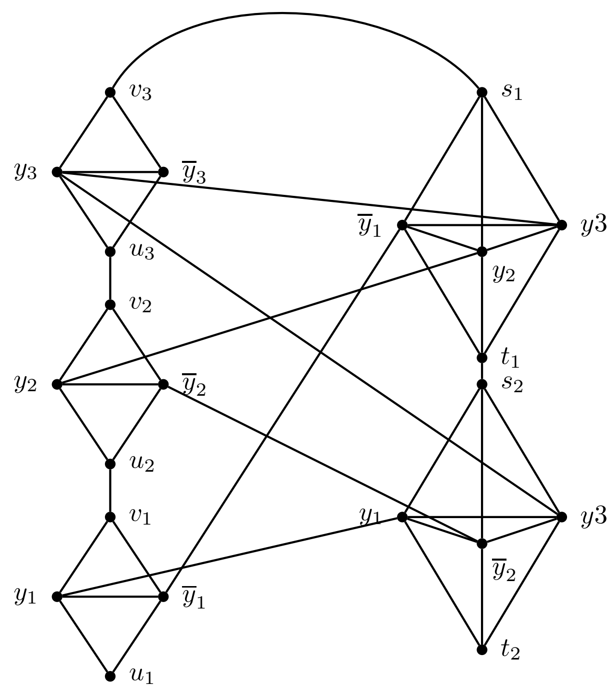

Figure 1: The graph H for $\varphi(y) = (\overline{y}_1 \wedge y_2 \wedge y_3) \vee (y_1 \wedge \overline{y}_2 \wedge y_3)$.

We claim that $(\forall y)[\varphi(y)]$ is true if and only if H is $P_{3(n+m)}$-free. If $(\forall y)[\varphi(y)]$ is false, there is an assignment of the variables y_i which makes $\varphi(y)$ false; fix such an assignment. In particular, for each clause C_j contains a literal which makes that clause false. Hence, there is a path starting at u_1 which picks the true literal in each variable gadget and then continues through C by picking the false literal in each clause gadget. This path is induced by construction and has length $3(n+m)$ which means that H is not $P_{3(n+m)}$-free.

To show that the truth of $(\forall y)[\varphi(y)]$ implies that H is $P_{3(n+m)}$-free, assume that H is not $P_{3(n+m)}$-free; that is H contains an induced path P of length $3(n+m)$. Note that an induced path can contain at most three vertices from each $K_4 - e$ (u_i, v_i and one of the two labeled vertices) and at most three vertices from each $K_5 - e$ (s_i, t_i and one of the three labeled vertices). If an induced path contains two labeled vertices from one of the $K_4 - e$ or $K_5 - e$ its total length of the path is at most $3(n+m-1)+2 < 3(n+m)$. Hence, P contains at most one labeled vertex from each $K_4 - e$ and $K_5 - e$, and, therefore, exactly one labeled vertex from each $K_4 - e$ and $K_5 - e$. However, by construction, such an induced path corresponds to an assignment to the variables in y that makes $\varphi(y)$ false. Consequently, $(\forall y)[\varphi(y)]$ is false.

It is now a short step to lifting this construction to a $\mathbf{\Sigma_2^p}$-hardness result for the graph sandwich problem. Assume we are given a formula $(\exists x)(\forall y)[\varphi(x, y)]$, where $\varphi(x, y)$ is in disjunctive normal form: $\varphi(x, y) = C_1(x, y) \vee C_2(x, y) \vee \ldots \vee C_m(x, y)$ where each clause $C_j(x, y)$ contains 3 literals. Deciding the truth of such a formula is known to be $\mathbf{\Sigma_2^p}$-complete [Sto76, Wra76, SU02]. For this formula, construct the graph H as described above (treating x and y variables the same). We need to modify this graph slightly: for each variable x_i we add a path Q_i of length $3(n + m) - 2$ between the u-vertex in the x_i-gadget and the vertex labeled x_i (that is, we need to add $3(n + m) - 3$ new vertices for each Q_i). After adding all these paths, we add edges from the inner vertices of any Q_i (that is, the vertices that are not endpoints) to *any* vertex not belonging to P_i. Call the resulting graph H'.

For the graph sandwich problem, we have to define two graphs F and F'. For F' we simply choose H' itself. For F we take H' and, for each x_i, remove the two edges connecting the x_i and $\overline{x}_i$ literals in the variable gadget to the u-vertex of the variable gadget (for an example, see Figure 2).

We claim that $(\exists x)(\forall y)[\varphi(x, y)]$ is true if and only if there is a $P_{3(n+m)}$-free G with $F \subseteq G \subseteq F'$.

If $(\exists x)(\forall y)[\varphi(x, y)]$ is false, then for any assignment to the x-variables, there is an assignment to the y variables, which makes $\varphi(x, y)$ false. Assume that there is a $P_{3(n+m)}$-free G with $F \subseteq G \subseteq F'$.

Consider one of the $K_4 - e$ belonging to some x_i. We want to argue that G contains either the edges from the u-vertex to the vertex labeled x_i or to the vertex labeled $\overline{x}_i$. If G does not contain either edge, then the edge between the vertices labeled x_i and $\overline{x}_i$ completes an induced $P_{3(n+m)}$ with Q_i which contradicts the assumption that G is $P_{3(n+m)}$-free. Therefore, G contains at least one of the two edges. If G contains the edge from u to the vertex labeled x_i, we call x_i true, otherwise we call it false (note that we do not exclude the case that both edges are present). For this assignment of truth-values to the x-variables we can find an assignment of truth-value to the y-variables so that $\varphi(x, y)$ is false (since we assumed that $(\exists x)(\forall y)[\varphi(x, y)]$ is false). Since $\varphi(x, y)$ is false, each clause C_j contains a literal which makes that clause false. We can now pick a path starting at u_1 which picks the true literal in each variable gadget (be it an x-variable or a y-variable) and continues through C picking the false literal in each clause gadget. By construction this path is induced and has length $3(n + m)$ contradicting the assumption that G is $P_{3(n+m)}$-free.

For the opposite direction, assume that $(\exists x)(\forall y)[\varphi(x, y)]$ is true. Fix an assignment to the x-variables such that $(\forall y)[\varphi(x, y)]$ is true. Pick G sandwiched between F and F' as follows: start with F; if x_i is true, add the edge from the u-vertex belonging to x_i to the vertex labeled x_i; if x_i is false, add the edge from the u-vertex belonging to x_i to the vertex labeled $\overline{x}_i$. Since F' includes both of those edges, this is a valid choice for G. We need to argue that G thus chosen is $P_{3(n+m)}$-free. Suppose it was not.

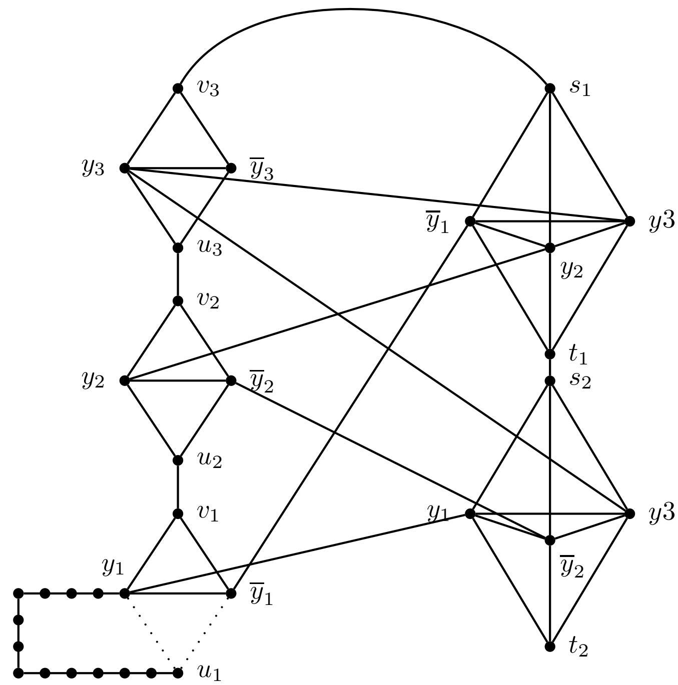

Figure 2: The graph H' for $(\exists y_1)(\forall y_2)(\forall y_3)[(\overline{y}_1 \wedge y_2 \wedge y_3) \vee (y_1 \wedge \overline{y}_2 \wedge y_3)]$. The dotted edges belong to F' but not to F.

First note that none of the edges belonging to any Q_i can be involved in an induced path of length more than $3(n+m)-1$, so we can safely ignore them. At this point we can argue as we did earlier that an induced path can contain at most three vertices from each $K_4 - e$ and $K_5 - e$ and, to have length $3(n+m)$ must indeed contain exactly one labeled vertex from each $K_4 - e$ and $K_5 - e$ together with a u- and a v-vertex (in the case of the variable gadgets) or a s- and a t-vertex (in the case of the clause gadgets). By construction, the induced path now corresponds to an assignment to the x and y variables that makes $\varphi(x, y)$ false. However, the assignment to x was chosen such that $\varphi(x, y)$ is true for all assignments to y, a contradiction.

To summarize, we have shown that given a $\mathbf{\Sigma_2^p}$-formula $(\exists x)(\forall y)[\varphi(x, y)]$ we can construct graphs F and F' so that the formula is true if and only if there is a graph G sandwiched between F and F' which is P_k-free for some k (depending on the formula). To phrase this as a graph sandwich problem, we need a property which does not depend on the input formula. This is easily done as follows: the graphs F, F' and G have at most $4n + 5m + n(3(n+m)) \leq 9(n+m)^2$ vertices. By adding isolated vertices to F and F' we can make sure that G has exactly $9(n+m)^2$ vertices. For this G, the number of vertices on $P_{3(n+m)}$ is exactly the square root of the number of vertices of G. Hence we can phrase the property Π as: the graph $G = (V, E)$ is $P_{|V|^{1/2}}$-free.

Theorem 2.1 GRAPH SANDWICH PROBLEM FOR Π *is* $\mathbf{\Sigma_2^p}$*-complete for the property of being* P_k*-free, where* k *is the square root of the order of the graph.*

3 Conclusion

It should be straightforward to extend the main result to the property of being H-free for other families of graphs, such as cycles. More interesting would be to find a truly natural property Π which is **coNP**-complete and leads to a $\mathbf{\Sigma_2^p}$-complete graph sandwich problem. As a candidate we suggest the class of *well-covered graphs*, that is, graphs in which every maximal independent set is maximum.

References

[FFKS06] Celina Figueiredo, Luerbio Faria, Sulamita Klein, and R. Sritharan. Graph sandwich problems, 2006.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability.* Freeman, San Francisco, 1979.

[GKS95] Martin Charles Golumbic, Haim Kaplan, and Ron Shamir. Graph sandwich problems. *J. Algorithms*, 19(3):449–473, 1995.

[Sto76] Larry J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, October 1976.

[SU02] Marcus Schaefer and Chris Umans. Completeness in the polynomial-time hierarchy: Part I: A compendium. *SIGACTN: SIGACT News (ACM Special Interest Group on Automata and Computability Theory)*, 33, 2002.

[Wra76] C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):23–33, October 1976.