# The Homogeneous Set Sandwich Problem*

Márcia R. Cerioli† Hazel Everett‡ Celina M. H. de Figueiredo†
Sulamita Klein†

**Abstract**

The graph sandwich problem for property $\Phi$ is defined as follows: Given two graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ such that $E_1 \subseteq E_2$, is there a graph $G = (V, E)$ such that $E_1 \subseteq E \subseteq E_2$ which satisfies property $\Phi$? We present a polynomial-time algorithm for solving the graph sandwich problem, when property $\Phi$ is "to contain a homogeneous set". The algorithm presented also provides the graph $G$ and a homogeneous set in $G$ in case it exists.

KEYWORDS: design of algorithms, sandwich problems, perfect graphs.

## 1 Introduction

We say that a graph $G_1 = (V, E_1)$ is a *spanning subgraph* of $G_2 = (V, E_2)$ when $E_1 \subseteq E_2$; we say that a graph $G = (V, E)$ is a *sandwich graph* for the pair $G_1$, $G_2$ when $E_1 \subseteq E \subseteq E_2$. According to [6], the GRAPH SANDWICH PROBLEM FOR PROPERTY $\Phi$ is defined as follows:

Problem: GRAPH SANDWICH PROBLEM FOR PROPERTY $\Phi$
Instance: Two graphs, $G_1$ and $G_2$, such that $G_1$ is a spanning subgraph of $G_2$.
Question: Does there exist a sandwich graph for the pair $G_1$, $G_2$ that satisfies property $\Phi$?

*This research was partially supported by NSERC and CNPq/ProTeM-CC-I, CNPq/ProTeM-CC-II, Project ProComb - Proc. 680065/94-6.

†Universidade Federal do Rio de Janeiro, Instituto de Matemática, Caixa Postal 68530, 21945-970, Rio de Janeiro, RJ, Brasil. {cerioli,celina,sula}@cos.ufrj.br.

‡Université du Québec à Montréal, Dep. d'informatique, Montréal, Québec, H3C 3P8, Canada. everett@math.uqam.ca.

Note that graph sandwich problems arise as a natural generalization of recognition problems. The recognition problem for a class of graphs $\mathcal{C}$ is equivalent to the graph sandwich problem in which $G_1 = G_2 = G$, where $G$ is the graph we want to recognize and property $\Phi$ is "to belong to class $\mathcal{C}$".

In [6], several subclasses of perfect graphs are considered with respect to sandwich problems. We note that many graph sandwich problems even for quite simple properties turn out to be $NP$-complete. In particular, it is proved that the GRAPH SANDWICH PROBLEM FOR BEING A PERMUTATION GRAPH turns out to be $NP$-complete. On the other hand, it is also proved that the GRAPH SANDWICH PROBLEM FOR BEING A SPLIT GRAPH remains in $P$. In this paper, we consider a new graph sandwich problem that is in $P$.

We are interested in graph sandwich problems for properties $\Phi$ which are related to compositions that arise in perfect graph theory. Several compositions of graphs are known to preserve perfection [2]: graph substitution, join composition, clique identification.

In this paper we present a polynomial-time algorithm for solving the graph sandwich-problem, when property $\Phi$ is "to contain a homogeneous set", which is related to graph substitution. The algorithm presented also provides the graph $G$ and a homogeneous set in $G$ in case it exists.

## 2 Homogeneous set

By a *homogeneous set* in a graph $G = (V, E)$, we mean a set $H$ of vertices of $G$ such that each vertex of $V \setminus H$ is either adjacent to all vertices of $H$ or to none of the vertices of $H$. In order not to have a trivial definition, we also ask that $|H| \geq 2$ and $|V \setminus H| \geq 1$.

We give in Figure 1 below two examples: a graph with a homogeneous set and a graph with no homogeneous set. We note that, by definition, every complete bipartite graph with at least three vertices admits a homogeneous set. On the other hand, every chordless cycle with at least five vertices does not admit a homogeneous set.

We observe that the existence of a homogeneous set $H$ in a graph $G$ implies that the set of vertices $V \setminus H$ can be partitioned into two sets $A$ and $N$ such that:

$$\begin{aligned} A &= \{v \in V : \mathcal{N}(v) \cap H = H\}; \\ N &= \{v \in V : \mathcal{N}(v) \cap H = \emptyset\}, \end{aligned}$$

Figure 1: Complete bipartite graph and chordless cycle.

where $\mathcal{N}(v)$ denotes the neighbourhood of $v$, i.e., the set of vertices of $V$ that are adjacent to $v$. We also say that $x$ *sees* $v$ when $x \in \mathcal{N}(v)$. We extend this terminology to sets, by saying that vertex $v$ sees the set $\mathcal{N}(v)$. In addition, vertex $v$ *sees partially* set $S$ when $v$ sees at least a vertex of $S$ but not all vertices of $S$. For simplicity, we shall denote by $S$ both the vertex set $S \subset V$ and $G[S]$, the subgraph of $G$ induced by $S$. As usual, we denote by $n$ and $m$ the sizes of the vertex set and of the edge set of $G$, respectively.

We can represent a graph $G$ with a homogeneous set $H$ by the diagram in Figure 2, where a continuous line between two sets represents the property that each vertex of one set is adjacent to each vertex of the other set. A broken line represents the property that no vertex of a set is adjacent to any vertex of the other set.

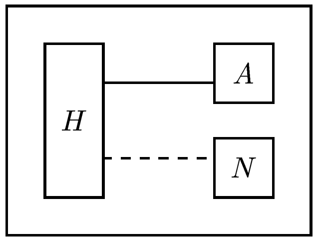


Figure 2: A diagram for a graph with homogeneous set $H$.

The property of containing a homogeneous set is a self-complementary property. A central result in perfect graph theory is the perfect graph theorem: the class of perfect graphs is a self-complementary class, i.e., a graph is perfect if and only if its complement is perfect [8]. In his proof of the perfect graph theorem, Lovász used the concept of homogeneous set to define a decomposition procedure that preserves perfection.

Given two disjoint graphs $G$ and $H$, and a vertex $x$ of $G$, consider the following graph $G'$: Take the disjoint union of $G \setminus x$ and $H$, and for each pair of vertices $y, z$ with $y \in G \setminus x$

and $z \in H$, add the edge $(y,z) \in E(G')$ if and only if $(x,y) \in E(G)$. In this case, we say that $G'$ was obtained by *substitution* of $x$ by $H$ in $G$. Figure 3 represents a graph obtained by substitution.

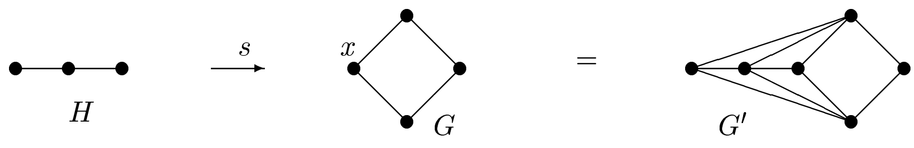


Figure 3: Substitution of $x$ by $H$ in $G$.

We note that a graph admits a homogeneous set if and only if this graph can be obtained by substitution. We also note that, by using substitution, we have a natural decomposition of a given root graph $G$ with homogeneous set $H$ into two subgraphs: $H$ itself and $G \backslash (H \backslash x)$. This decomposition can be used to compute both the chromatic number and the clique number of $G$, given those parameters for its children [3]. Moreover, this decomposition has been used to obtain recognition algorithms for well known classes of perfect graphs such as comparability graphs [5] and bull-free perfect graphs [12].

Polynomial-time algorithms for finding homogeneous sets are given in [11, 10, 13, 1, 7]. The fastest is an $O(m)$-algorithm by McConnell and Spinrad [9]. See [13] for an $O(m\alpha(m,n))$-algorithm.

We note that the sandwich problem for homogeneous set is not trivial, in the sense that it is neither a hereditary property (in this case the graph $G_1$ is always a solution) nor an ancestral property (in this case the graph $G_2$ is always a solution). Figure 4 shows an instance where neither $G_1$ nor $G_2$ admit a homogeneous set but there exists a sandwich graph $G$ for this property with $H = \{3,4,5\}$, $A = \{2,6\}$ and $N = \{1\}$.

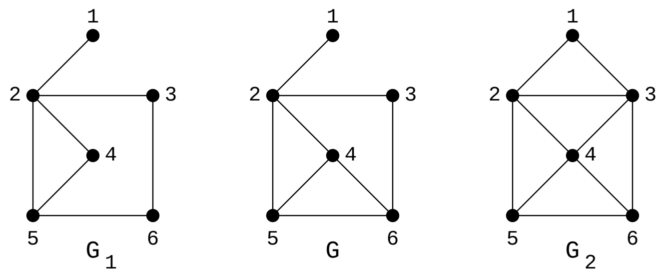


Figure 4: Sandwich graph $G$ that admits a homogeneous set.

## 3 The Algorithm

Let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ be two graphs such that $G_1$ is a spanning subgraph of $G_2$. Our algorithm tests whether there exists a sandwich graph $G$ that admits a homogeneous set. It follows closely and extends the algorithm for testing the existence of a homogeneous set in a graph $G$ given by Reed [11].

The idea is to test, for any two vertices $x$ and $y$ in $V$, if there exists a sandwich graph $G$ that admits a homogeneous set $H$ with $x$ and $y$ in $H$.

Our method starts with a candidate set $H$ consisting of the vertices $x$ and $y$. At each step, we add new vertices to the current set $H$ by considering adjacency with respect to both $G_1$ and $G_2$. For the current set $H$, we partition $V \setminus H$ into four sets: $A, N, Q$ and $C$. This partition is based in two properties: a vertex that is adjacent to at least a vertex of $H$ in $G_1$ is also adjacent to at least a vertex of $H$ in $G_2$ and therefore cannot be placed in set $N$. Analogously, a vertex that is not adjacent to at least one vertex of $H$ in $G_2$ is not adjacent to at least one vertex of $H$ in $G_1$ either and therefore cannot be placed in set $A$.

Thus, we consider first a vertex that is adjacent to at least a vertex of $H$ in $G_1$ and it is adjacent to all vertices of $H$ in $G_2$. We place this vertex in set $A$. This means that this vertex cannot be in set $N$, i.e., in case there is a solution, this vertex is in $H \cup A$. A vertex that is adjacent to no vertex of $H$ in $G_1$ and it is not adjacent to at least one vertex of $H$ in $G_2$ is placed in set $N$. This means that this vertex cannot be in set $A$, i.e., in case there is a solution, this vertex is in $H \cup N$. On the other hand, a vertex that is adjacent to no vertex of $H$ in $G_1$ but it is adjacent to all vertices of $H$ in $G_2$ is placed in set $Q$. This set contains the question mark vertices as we have no information about those vertices yet. We put all the remaining vertices into the last set $C$. More precisely, a vertex that sees $H$ partially both in $G_1$ and in $G_2$, cannot be in $A$ because it is not adjacent to at least one vertex of $H$ in $G_2$ and cannot be in $N$ because it is adjacent to at least a vertex of $H$ in $G_1$. Thus, such a vertex is placed in $C$. The next set $H$ is obtained by adding $C$ to the current set $H$.

We have below the complete algorithm and the corresponding proof of correctness. We shall denote by $\mathcal{N}_i(v)$ the neighbourhood of $v$ respectively in $G_i$, $i \in \{1, 2\}$.

**Algorithm 1** *Test for the existence of a sandwich graph $G$ that admits a homogeneous set $H$ containing two given vertices.*

Input: $G_1 = (V, E_1)$, $G_2 = (V, E_2)$ such that $E_1 \subseteq E_2$ and two vertices $x, y \in V$.

Output: YES or NO.

If the answer is YES, the algorithm also gives the edge set $E$ of the sandwich graph $G$ and the partition of the vertex set $V$ into $H, A$, and $N$ such that:

$$\begin{aligned} H &= \text{homogeneous set of } G \text{ containing } x, y; \\ A &= \{v \in V : \mathcal{N}(v) \cap H = H\}; \\ N &= \{v \in V : \mathcal{N}(v) \cap H = \emptyset\}. \end{aligned}$$

1. Create five vertex sets: $H, A, N, Q$ and $C$.
2. $H := \{x, y\}$;
   Partition the vertex set $V^\star = V \setminus \{x, y\}$ into the following four sets:
   $A := [\mathcal{N}_1(x) \cup \mathcal{N}_1(y)] \cap [\mathcal{N}_2(x) \cap \mathcal{N}_2(y)]$;
   $N := [V^\star \setminus (\mathcal{N}_1(x) \cup \mathcal{N}_1(y))] \cap [(V^\star \setminus \mathcal{N}_2(x)) \cup (V^\star \setminus \mathcal{N}_2(y))]$;
   $Q := [V^\star \setminus (\mathcal{N}_1(x) \cup \mathcal{N}_1(y))] \cap [\mathcal{N}_2(x) \cap \mathcal{N}_2(y)]$;
   $C := V^\star \setminus (A \cup N \cup Q)$.
3. If $H = V$ return NO and stop.
4. If $C = \emptyset$
   $N := N \cup Q$;
   $E := E_1 \cup \{(a, h) : a \in A \text{ and } h \in H\}$;
   return (YES, $E, H, A, N$) and stop.
5. Consider three auxiliary vertex sets: $C', A'$ and $N'$.
   $C' := \emptyset$; $A' := \emptyset$; $N' := \emptyset$;
   For each $z \in C$
   $C' := C' \cup (\mathcal{N}_1(z) \cap N) \cup (A \setminus \mathcal{N}_2(z))$;
   $A' := A' \cup (\mathcal{N}_1(z) \cap Q)$;
   $N' := N' \cup (Q \setminus \mathcal{N}_2(z))$.
6. $H := H \cup C$;
   $A := (A \setminus C') \cup (A' \setminus (A' \cap N'))$;

$N := (N \setminus C') \cup (N' \setminus (A' \cap N'))$;

$Q := Q \setminus (A' \cup N')$;

$C := C' \cup (A' \cap N')$;

Go to 3.

**Theorem 1** *Algorithm 1 correctly tests for the existence of a sandwich graph $G$ that admits a homogeneous set $H$ containing two given vertices.*

**Proof**: The algorithm constructs a sequence of sets

$$\{x, y\} = H_0 \subset H_1 \subset \ldots \subset H_t$$

by adding to the current set $H$ at each step the vertices that must be in $H$, i.e., the vertices that see partially the current set $H$, both in $G_1$ and $G_2$.

Suppose first that the algorithm stops with $C = \emptyset$. We want to show that the sets $E, H, A, N$ are such that $H$ is a homogeneous set of $G$. For we shall prove that Step 6 keeps as invariant the following relation of set $H$ with respect to sets $A, N, Q$ and $C$: set $A$ contains the vertices that cannot be in $N$; set $N$ contains the vertices that cannot be in $A$; set $Q$ contains the question mark vertices; set $C$ contains the vertices that cannot be in $A$ and that cannot be in $N$ either.

This is done by introducing those three auxiliary vertex sets: $C', A'$ and $N'$ in Step 5. Indeed, set $A'$ contains those vertices that are no longer in $Q$ as they now see in $G_1$ a vertex of $H$. In Step 6, part of set $A'$ is put in set $A$ and part of set $A'$ is put in set $C$. Analogously, set $N'$ contains those vertices that are no longer in $Q$ as they now fail to see in $G_2$ a vertex of $H$. Finally, set $C'$ contains those vertices that are no longer in $N$ as they now see in $G_1$ a vertex of $H$ and contains those vertices that are no longer in $A$ as they now fail to see in $G_2$ a vertex of $H$.

On the other hand, suppose that the algorithm stops with $H_t = V$ but there exists a sandwich graph $G$ with a homogeneous set $F$ containing $\{x, y\}$. Because $F$ contains $\{x, y\}$ and $F \subset H_t = V$, we must have an index $i > 0$ such that $H_{i-1} \subseteq F$ but $H_i \not\subseteq F$. By construction, $H_i \setminus H_{i-1}$ is the set of vertices that see $H_{i-1}$ partially, both in $G_1$ and $G_2$. Now, any vertex in $H_i \setminus F$ is in $H_i \setminus H_{i-1}$. Hence there exists a vertex in $V \setminus F$ that sees $F$ partially, both in $G_1$ and $G_2$, a contradiction. ■

Using adjacency lists for the graphs and ordered sets, a straightforward implementation of Algorithm 1 uses $O(n^2)$ time and $O(n^2)$ space. The time consuming part of each iteration is Step 5. For each $z$ in $C$, we have to compute a constant number os unions and intersections of finite sets. By traversing the adjacency list for $z$, we can do this in $O(n)$ time. Since a vertex passes through $C$ at most once, it follows that the algorithm takes $O(n^2)$ time.

We run Algorithm 1 once for any two vertices $x$ and $y$ in $V$ until we find a sandwich graph with homogeneous set containing a pair $x$ and $y$ or we have tested all pairs of vertices. So Algorithm 1 is executed at most $n^2$ times giving a total bound of $O(n^4)$ time to solve the sandwich problem for homogeneous set.

# 4 Conclusion

We have presented a polynomial-time algorithm for the HOMOGENEOUS SET SANDWICH PROBLEM.

One other type of relaxation for recognition problems which has been well-studied is the EDGE-COMPLETION PROBLEM: given a graph and an integer $k$, it asks whether one can add to the original graph at most $k$ edges in order to obtain a graph with a certain property. Such problems have been studied for properties such as: to be a hamiltonian graph, to be an interval graph, to be a path graph [4] and to be a chordal graph [14].

The corresponding edge-completion problem for homogeneous set asks for the minimum number of edges needed to be added to a given graph $G_1$ in order to give a graph with a homogeneous set. our algorithm does not solve this problem. Actually, if $G_2$ is the complete graph, then our algorithm always returns $H = \{x, y\}$. If $G_1$ contains a homogeneous set, but contains no homogeneous set with just two vertices, then our algorithm does not give a solution with the minimum number of edges.

Our algorithm does satisfy a minimality property with respect to the number of vertices in $H$, i.e., our method looks for a sandwich graph $G$ with homogeneous set $H$ (that contains a given pair of vertices) as small as possible.

## Acknowledgements

We are grateful to João Meidanis for introducing us to graph sandwich problems and to Jayme L. Szwarcfiter for pointing out the homogeneous set sandwich problem.

## References


[1] A. Cournier, *Sur Quelques Algorithmes de Décomposition de Graphes*, Thése, Université Montpellier II, (1993).

[2] G. Cornuéjols and W. H. Cunningham, Compositions for perfect graphs, *Discrete Math.* **55** (1985) 245–254.

[3] C. M. H. de Figueiredo, *A Study of Combinatorial Problems in Perfect Graphs*, Tese de Doutorado, Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ, In Portuguese, (1991).

[4] M. R. Garey, D. S. Johnson, Computers and Intractability, a guide to the theory of NP-completeness, Freeman, San Francisco, (1979) 198–199.

[5] M.C. Golumbic, Comparability graphs and a new matroid, *J. Comb. Theory. (B)* **22** (1977) 68–90.

[6] M. C. Golumbic, H. Kaplan and R. Shamir, Graph sandwich problems, *J. of Algorithms* **19** (1995) 449–473.

[7] S. Klein, *Algorithms and Complexity of Graph Decomposition*, Tese de Doutorado, Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ, In Portuguese, (1994).

[8] L. Lovász, Normal hypergraphs and the perfect graph conjecture, *Discrete Math.* **2** (1972) 253–267.

[9] R. M. McConnell and J. Spinrad, Linear-time modular decomposition and efficient transitive orientation of comparability graphs, In Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms. (1994) 536–545.

[10] J.H. Muller and J. Spinrad, Incremental modular decomposition, *Journal of the ACM* **36** (1989) 1–19.

[11] B. Reed, *A Semi-Strong Perfect Graph Theorem*, Ph.D. Thesis, School of Computer Science, McGill University, Montreal, (1986).

[12] B. Reed, N. Sbihi, Recognizing bull-free perfect graphs, *Graphs and Combinatorics* **11** (1995) 171–178.

[13] J. Spinrad, $P_4$-trees and substitution decomposition, *Discrete Appl. Math.* **39** (1992) 263–291.

[14] M. Yannakakis, Computing the minimum fill-in is NP-complete, *SIAM J. Alg. Disc. Meth.* **2** (1981) 77–79.