

Note on the Homogeneous Set Sandwich Problem *

Celina M. H. de Figueiredo[†]

Vinícius G. P. de Sá*

Abstract

A homogeneous set is a non-trivial module of a graph, i.e. a non-unitary, proper subset H of a graph's vertices such that all vertices in H have the same neighbors outside H . Given two graphs $G_1(V, E_1), G_2(V, E_2)$, the Homogeneous Set Sandwich Problem asks whether there exists a sandwich graph $G_S(V, E_S)$, $E_1 \subseteq E_S \subseteq E_2$, which has a homogeneous set. Recently, Tang *et al.* [9] proposed an interesting $O(\Delta_1 \cdot n^2)$ algorithm for this problem, which has been considered its most efficient algorithm since. We show the incorrectness of their algorithm by presenting three counterexamples.

1 Introduction

A graph $G_S(V, E_S)$ is said to be a *sandwich graph* of graphs $G_1(V, E_1), G_2(V, E_2)$ if and only if $E_1 \subseteq E_S \subseteq E_2$. A homogeneous set H for a graph $G(V, E)$ is a subset of V such that $1 < |H| < |V|$ and for all $v \in V \setminus H$, either $(v, h) \in E$ for all $h \in H$ or $(v, h) \notin E$ for all $h \in H$. Given two graphs $G_1(V, E_1), G_2(V, E_2)$ such that $E_1 \subseteq E_2$, the Homogeneous Set Sandwich Problem (HSSP) comprises the search for a sandwich graph $G_S(V, E_S)$ of (G_1, G_2) which contains a homogeneous set. Such a homogeneous set is called a *sandwich homogeneous set* of pair (G_1, G_2) .

Throughout this paper, we denote the number of vertices in the input graphs by n , the number of edges in graph G_i by m_i and the number of edges *not* in G_i by \overline{m}_i .

Notwithstanding the existence of linear-time algorithms for solving the problem of finding homogeneous sets in a single graph [2, 4, 5, 6, 7, 8], the known HSSP algorithms are considerably less efficient.

The first polynomial-time algorithm for this problem was presented by Cerioli *et al.* [1], which set HSSP's upper bound at their algorithm's $O(n^4)$ time complexity. We refer to this algorithm as the *Exhaustive Bias Envelopment Algorithm (EBE algorithm, for short)*. A few years later, Tang *et al.* [9] tailored a brand new algorithm, based on a quite beautiful idea of theirs, which would have largely diminished HSSP's upper bound. This algorithm is referred to as the *Bias Graph Components Algorithm (BGC algorithm, for short)*. We show, with brief counterexamples, that this algorithm is unfortunately not correct. Consequently, the most efficient algorithm that correctly solves the HSSP would turn back to be former EBE algorithm presented in [1], resetting HSSP's upper bound at $O(n^4)$. A careful study of the underlying ideas contained in both [1] and [9], though, has led us to the development of a faster deterministic algorithm, which establishes [3] a new upper bound to the problem at $O(m_1 \overline{m}_2)$.

*This is a manuscript by the authors. The final article was published by Elsevier with DOI number 10.1016/j.ipl.2004.09.022.

[†]Instituto de Matemática and COPPE, Universidade Federal do Rio de Janeiro, Caixa Postal 68530, 21945-970 Rio de Janeiro, RJ, Brazil. E-mail: {celina,vigusmao}@cos.ufrj.br

We summarize the EBE algorithm, in Section 2, and refine its analysis. Actually, we show that its time complexity can be more precisely bounded by $O(n^2 \cdot (m_1 + \overline{m}_2))$, which is somewhat better. In Section 3, we give a brief description of the BGC algorithm and point out where its basic flaw lies by presenting three counterexamples.

2 The Exhaustive Bias Envelopment algorithm

Before describing the EBE algorithm, presented in [1], we define some notation which will be used henceforth.

Let $G_S(V, E_S)$ be a sandwich graph of graphs $G_1(V, E_1)$, $G_2(V, E_2)$. The edges in E_1 are called *mandatory edges*, once each and every sandwich graph of (G_1, G_2) has to contain them. On the other hand, the edges *not* in E_2 are said to be *forbidden edges*, meaning that no sandwich graph of (G_1, G_2) is allowed to contain them. A vertex $b \in V$ is called a *bias vertex* of a vertex set $S \subseteq V \setminus \{b\}$ if there exists at least one mandatory edge (b, v) and at least one forbidden edge (b, w) , for some $v, w \in S$. The set $B(S)$ contains all bias vertices of S , thereby it is called the *bias set* of S [9].

The following theorem, based on the concept of bias sets, gives a characterization of sandwich homogeneous sets and is implicit in the proof of correctness of the EBE algorithm, presented in [1].

Theorem 1. *The set $S \subset V, |S| \geq 2$, is a sandwich homogeneous set of a pair (G_1, G_2) if and only if its bias set $B(S)$ is the empty set.*

Proof. Suppose $B(S) \neq \emptyset$. Thus, in all possible sandwich graphs of (G_1, G_2) , any vertex $t \in B(S)$ must be adjacent to at least one vertex $v \in S$ and also non-adjacent to at least one vertex $w \in S$. This clearly prevents S from being a sandwich homogeneous set. If we suppose, on the other hand, that $B(S) = \emptyset$, we are able to build a sandwich graph $G_S(V, E_S)$ of (G_1, G_2) in such a way that S is a homogeneous set of G_S . We do this by adding all mandatory edges $(u, v) \in E_1$ to an initially empty E_S . Then, for every vertex $x \in V \setminus S$ such that (x, y) is mandatory for some $y \in S$, we add to E_S the edges (x, z) from x to each and every vertex $z \in S$. Notice that this is always possible, once x is not a bias vertex of S . \square

Given Theorem 1, it is quite simple to understand the EBE algorithm. It starts by choosing a sandwich homogeneous set *candidate* $\{x, y\}$. Then it successively determines the candidate's bias vertices and adds all of them to the current candidate. We refer to this procedure as *bias envelopment*. The bias envelopment continues until either a candidate with an empty bias set has been found, whereby the algorithm stops with an *yes* answer, or else the candidate set has become equal to the input vertex set V , in which case the algorithm restarts the process with another initial pair of vertices. If no sandwich homogeneous set has been found by the time all possible pairs have been investigated, the algorithm answers *no*.

Figure 1 presents the pseudo-code for the EBE algorithm.

Theorem 2. [1] *The EBE algorithm is a complete, correct method for solving the HSSP.*

The time complexity of this algorithm is undoubtedly $O(n^4)$, as in [1]. However, we can tighten this bound a little bit by allowing m to take place in the analysis.

Let $G_1(V, E_1)$, $G_2(V, E_2)$ be an input for the HSSP. At a first glance, *each iteration* of the algorithm's inner loop (lines 1.3.1 to 1.3.3) would take $O(n^2)$ time, for computing a bias set $B(H)$

The Exhaustive Bias Envelopment algorithm ($G_1(V, E_1), G_2(V, E_2)$)

1. For each pair of vertices $\{x, y\} \subset V$ do
 - 1.1. $H \leftarrow \{x, y\}$.
 - 1.2. Find the bias set $B(H)$.
 - 1.3. While $H \neq V$ do
 - 1.3.1. If $B(H) = \emptyset$ then return *yes* and H . End.
 - 1.3.2. $H \leftarrow H \cup B(H)$.
 - 1.3.3. Update $B(H)$.
 2. Return *no*.
-

Figure 1: The EBE algorithm [1]

from scratch demands that all vertices v that are not in H are investigated (in order to check out whether there exists both mandatory and forbidden edges between v and whichever vertices in H). Notice, however, that each bias set (except for the first one, which is outside the inner loop) is *not* computed from scratch, but updated (line 1.3.3), instead, from the bias set of the preceding iteration. This is accomplished with the introduction of three auxiliary, dynamically maintained sets, as described in [1]. Each update in the current bias set is, then, achieved as a result of a constant number of unions, differences and intersections of sets, none of which containing more than n vertices. Along with the fact that no vertex enters the bias set more than once, this allows that *the whole loop* (i.e. all its iterations) can be carried on in $O(n^2)$ time. Thus, the complexity of the EBE algorithm, which runs the bias envelopment on $O(n^2)$ candidates in the worst case, is certainly $O(n^2 \cdot n^2) = O(n^4)$. Nevertheless, this analysis can be slightly improved.

The point is, one of the sets involved in each of those unions, differences and intersections described in [1] is always the set of neighbors, in G_1 (resp. non-neighbors, in G_2), of vertices b in the bias set of the preceding iteration. We remark that any union, difference or intersection of any two subsets S_1, S_2 of some finite set S with pre-ordered elements can be achieved in $O(\text{Min}\{|S_1|, |S_2|\})$ time, granted an adequate data structure is used. Thus, the time complexity of any operation involving the set $N_1(b)$ (resp. $\overline{N}_2(b)$) of neighbors of b in G_1 (resp. non-neighbors of b in G_2), during a bias set update, is correctly bounded by a linear function of the cardinality of $N_1(b)$ (resp. $\overline{N}_2(b)$). On this basis, each iteration of the inner loop (lines 1.3.1 to 1.3.3) can be done in $O(\sum_{b \in B(H)} |N_1(b)| + |\overline{N}_2(b)|)$ time. As each vertex $v \in V$ appears in $B(H)$ only once, the whole bias envelopment loop (line 1.3) takes $O(\sum_{v \in V} |N_1(v)| + |\overline{N}_2(v)|) = O(m_1 + \overline{m}_2)$ time. Therefore, the whole EBE algorithm runs in $O(n^2 \cdot (m_1 + \overline{m}_2))$ time.

3 The Bias Graph Components algorithm

The main idea of the BGC algorithm, presented in [9], is to use the bias relation introduced in Section 2 to construct a directed graph, called *bias graph*. The bias graph exhibits at once these relations, allowing interdependent vertices to be quickly grouped in a number of disjoint sets, some of which likely to be associated with sandwich homogeneous sets.

The bias graph $G_B(V_B, E_B)$ of a pair of graphs $G_1(V, E_1), G_2(V, E_2)$ has vertex set $V_B = \{\{x, y\} \mid x, y \in V, x \neq y\}$ and there are two outgoing edges from vertex $[u, v]$ to vertices $[u, w]$ and

The Bias Graph Components algorithm $(G_1(V, E_1), G_2(V, E_2))$

1. Construct the bias graph G_B of (G_1, G_2) .
2. Find an end strongly connected component C of G_B .
3. Let H denote the set of vertices in V that label the vertices in C .
4. If $H = V$ then return *no*. End.
5. Return *yes* and H .

Figure 2: The BGC algorithm [9]

$[v, w]$ in G_B if and only if vertex w is a bias vertex of vertex set $\{u, v\}$ with respect to the pair (G_1, G_2) . Notice that vertices $[x, y]$ and $[y, x]$ in G_B are the same.

Once the bias graph has been constructed, the algorithm runs Tarjan’s method [10] to find all its strongly connected components and then looks for an *end* strongly connected component (ESCC) among them, i.e. a strongly connected component with no outgoing edges. If only one ESCC is found and it embraces all input vertices (as part of its vertices’ labels), the algorithm returns *no*. Otherwise, the algorithm translates one of the bias graph’s ESCCs, say component C , into the set $H \subset V$ of input vertices that are used to label C ’s vertices. Then it returns *yes* and H , for H is allegedly a sandwich homogeneous set.

The summarized steps of the BGC algorithm are shown in Figure 2.

Claim 3. [9] *The BGC algorithm correctly solves the HSSP.*

Tang *et al.* present Claim 3 as a theorem whose proof is based on the validity of the next two lemmas, one for the algorithm’s correctness and the other for its completeness. We show that both are incorrect.

Lemma 4. [9] *The set H of vertices found in line 3 of the BGC algorithm is a sandwich homogeneous set of the input graphs (G_1, G_2) .*

To begin with, Figure 3(a) shows a very simple refutation. It presents a pair of graphs (G_1, G_2) that produce the bias graph $G_B(V_B, E_B)$ in Figure 3(b). It is easy to see that the subgraph C on the left of the dashed line constitutes an ESCC. (The bold edges in C stress the existence of cycles providing a path from each vertex in C to every other vertex in C . Notice, also, that all edges that come across the dashed line *reach* C , which makes an *end* strongly connected component out of it.) The set $H = \{1, 2, \dots, 7\} \subset V$ that labels the vertices in C , however, is *not* a sandwich homogeneous set of (G_1, G_2) . (Notice that vertex 8 is a bias vertex of H , once the input instance presents mandatory edge $(1, 8) \in E_1$ and forbidden edge $(2, 8) \notin E_2$.) As the BGC algorithm might possibly choose C (among other existing G_B ’s ESCCs) in line 2, it is likely to answer *yes* along with set $H = \{1, 2, \dots, 7\}$, which is definitely *not* a sandwich homogeneous set of (G_1, G_2) .

Tang *et al.* seem to have overlooked the possibility that an ESCC C does not comprise all possible vertices $[x, y]$ such that x and y appear in some of its vertices’ labels. This may cause the set $H \subseteq V$, associated with $C \subset V_B$, to contain both vertices x and y , but not some bias vertex b of $\{x, y\}$ that happened not to label any of C ’s vertices. In such cases, H is *not* a sandwich

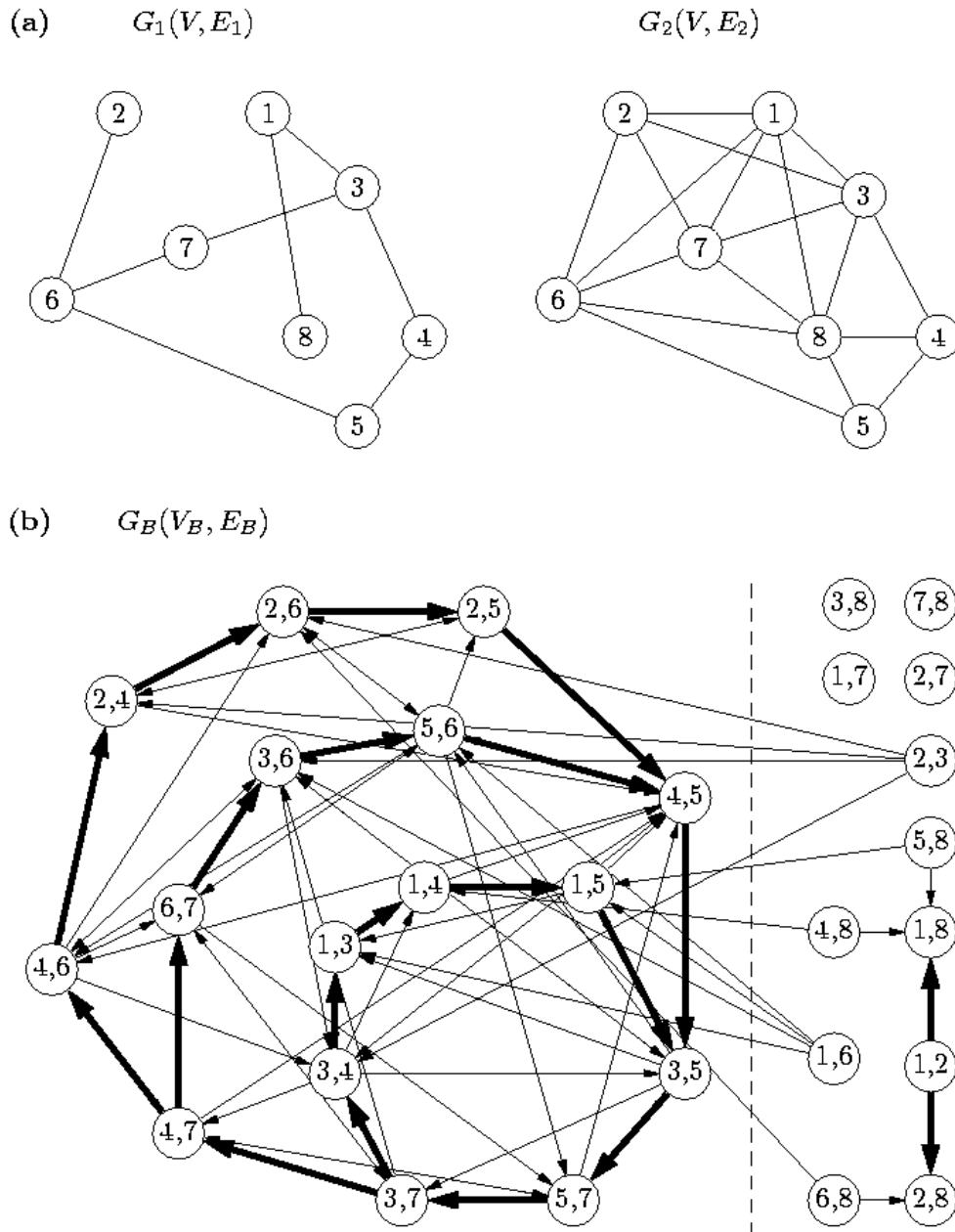


Figure 3: Counterexample 1 (to Lemma 4 [9])

homogeneous set, despite the fact that C is an ESCC. The bias graph in Figure 3(b) illustrates it. Although vertices 1 and 2 do appear in the labels of some vertices in the ESCC (on the left of the dashed line), the very vertex $[1, 2] \in V_B$ is not itself in this ESCC. That is why vertices $[1, 8], [2, 8] \in V_B$, which are respectively incident to edges $([1, 2], [1, 8])$ and $([1, 2], [2, 8])$ are not seen by the ESCC, therefore preventing vertex 8 from taking part in H , contrarily to what Tang *et al.* may have expected it to. (Notice that vertex 8 is a bias vertex of $\{1, 2\}$ and, consequently, of

$H \supset \{1, 2\}$, once $H \not\supseteq \{8\}$).

It is true that the HSSP instance in Figure 3(a) does have some sandwich homogeneous sets, although set $H = \{1, 2, \dots, 7\}$, which might possibly have been returned by the BGC algorithm, is not among them. (E.g. set $\{1, 8\}$ is a homogeneous set of sandwich graph $G_S(V, E_S)$, where $E_S = E_1 \cup \{(3, 8)\}$). Interesting enough, Figure 4(a) shows an instance which does not admit any sandwich homogeneous sets *at all*. Still its bias graph G_B , shown in Figure 4(b), has two proper ESCCs, which causes the BGC algorithm to incorrectly answer *yes*. (In Figure 4(b), we removed the commas from all vertex labels in order to save some space.) Vertex S (resp. S') condensates G_B 's induced subgraph with vertex set $\{1, 2, \dots, 7\}$ (resp. $\{1', 2', \dots, 7'\}$). S and S' are both isomorphic to the ESCC on the left of the dashed line in Figure 3(b), which grants they are still strongly connected. Also, there are not any outgoing edges from neither S nor S' . (This is highlighted, in the figure, by means of three big arrowheads towards both S and S'). The bold edges in the leftmost half of the figure (and their counterparts in the other half, for the graph is noticeably symmetrical) stress the existence of a path from every G_B 's vertex $v \notin S \cup S'$ to one of the ESCCs S or S' . This clearly prevents the existence of ESCCs other than S and S' , in G_B . Thus, being the only ESCCs in G_B , S and S' are the only possible choices in line 2 of the BGC algorithm. However, neither S nor S' can be associated to any sandwich homogeneous sets whatsoever (in fact, there does not exist any!), thence an incorrect answer is inevitable.

Lemma 5. [9] *If graphs (G_1, G_2) admit a sandwich homogeneous set, then the BGC algorithm can find one.*

Unfortunately, this is not correct either. Figure 5(a) illustrates the pair $G_1(V, E_1)$, $G_2(V, E_2)$, which has sandwich homogeneous set $H = \{1, 2, \dots, 9, 1', 2', \dots, 9'\}$ (and no other). However, this sandwich homogeneous set simply cannot be found by the BGC algorithm, for it is not associated with any of the two existing ESCCs in the bias graph of (G_1, G_2) . The point is that it is neither sufficient (as we saw in the refutation of Lemma 4) nor necessary that a set of vertices in G_B constitute an end strongly connected component in order to be associated with a sandwich homogeneous set. Figure 5(b) shows the bias graph $G_B(V_B, E_B)$ of input instance in Figure 5(a), which has 210 vertices and 1684 edges. For obvious reasons, its graphic representation is rather condensed here. The vertex labelled K comprises a 153-vertex induced subgraph of G_B 's that is isomorphic to the whole bias graph in Figure 4(a) and holds all vertices $[x, y] \in V_B$ such that $x, y \in \{1, 2, \dots, 9, 1', 2', \dots, 9'\}$. (To save space, all commas in the vertices' labels were again suppressed.)

We know already that there are two (and only two) ESCCs inside K , namely S and S' , which happen to be the only ESCCs in the whole G_B . This can be easily verified by noticing that (i) there are not any outgoing edges leaving K and (ii) there is a path to K from each and every vertex outside K . Again, because of the huge number of edges in this bias graph, we have wrapped similar groups of vertices in three bounding boxes with 17 vertices each. An edge that leaves (resp. reaches) one of these boxes towards (resp. coming from) a vertex v stands for 17 converging (resp. diverging) edges towards (resp. coming from) v , one from (resp. to) each vertex inside the origin box. Irrelevant edges have not been drawn.

In this case, the BGC algorithm would certainly answer *yes*, giving one of the two fake sandwich homogeneous sets $F = \{1, 2, \dots, 7\}$ or $F' = \{1', 2', \dots, 7'\}$, associated with S and S' , respectively. It is easy to see that vertices 8 and 8' forbid them to be sandwich homogeneous sets, invalidating such answers. More than that, this instance's one and only sandwich homogeneous set H cannot be found by the BGC algorithm. Recall that K stands for the induced subgraph of G_B that holds all vertices $[x, y]$ such that $x, y \in H$. In spite of being an *end subgraph* of G_B (i.e. a subgraph

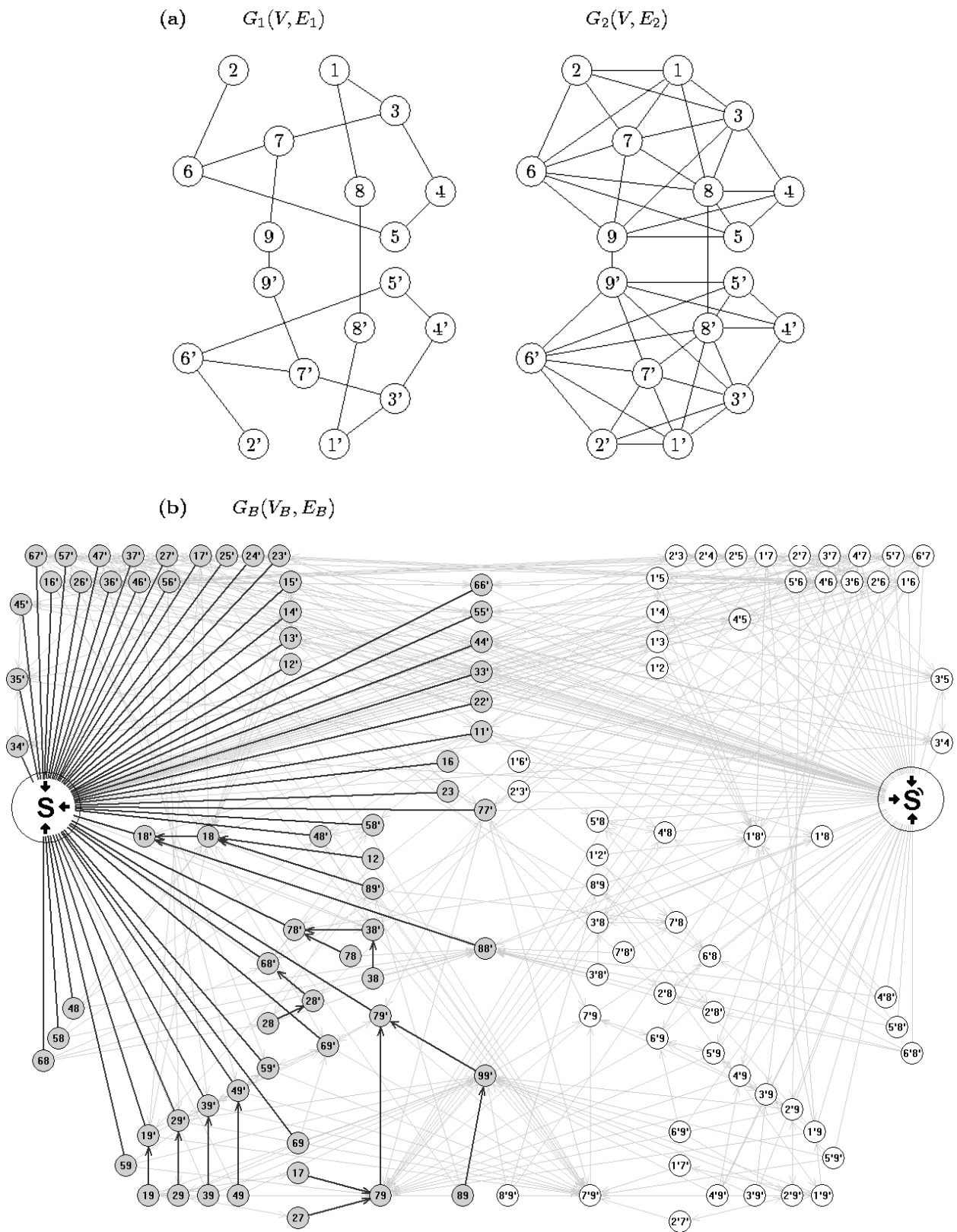


Figure 4: Counterexample 2 (to Lemma 4 [9])

that does not have any outgoing edges), K is not *strongly connected*, hence cannot be found by Tarjan’s SCC-partitioning method. However, it is easy to see that H is indeed a homogeneous set of sandwich graph G_S which contains the (mandatory) edges in E_1 plus edge $(A, 9) \in E_2$.

4 Conclusion

This paper presented some counterexamples which invalidate the (so far) best algorithm for the Homogeneous Set Sandwich Problem. A quite natural step would certainly be to question the minimality of Counterexample 1. The answer is yes. It can be shown that any ESCC which might not be associated with a sandwich homogeneous set has to be labelled by vertices of a set $H \subset V$ with cardinality at least 7. As long as an extra (bias) vertex is required in $V \setminus H$, we have that any suchlike counterexamples to Tang *et al.*’s Claim 3 must have at least 8 input vertices – as does Counterexample 1.

References

- [1] M. R. Cerioli, M. R. Everett, C. M. H. Figueiredo, and S. Klein, *The homogeneous set sandwich problem*, Information Processing Letters **67** (1998), 31–35.
- [2] A. Cournier, *Sur quelques algorithmes de décomposition de graphes*, Ph.D. thesis, Université Montpellier II, 1993.
- [3] C. M. H. Figueiredo and V. G. P. Sá, *A new upper bound for the homogeneous set sandwich problem*, Tech. Report ES-628/04, COPPE/Sistemas, Universidade Federal do Rio de Janeiro, 2004.
- [4] M. R. Garey and D. S. Johnson, *Computers and intractability, a guide to the theory of np-completeness*, pp. 198–199, Freeman, San Francisco, 1979.
- [5] R. M. McConnell and J. Spinrad, *Modular decomposition and transitive orientations*, Discrete Math. **201** (1999), 189–241.
- [6] J. H. Muller and J. Spinrad, *Incremental modular decomposition*, J. ACM **36** (1989), 1–19.
- [7] B. Reed, *A semi-strong perfect graph theorem*, Ph.D. thesis, School of Computer Science, McGill University, Montreal, 1986.
- [8] J. Spinrad, *P_4 -trees and substitution decomposition*, Discrete Appl. Math. **39** (1992), 263–291.
- [9] S. Tang, F. Yeh, and Y. Wang, *An efficient algorithm for solving the homogeneous set sandwich problem*, Information Processing Letters **77** (2001), 17–22.
- [10] R. E. Tarjan, *Depth-first search and linear graph algorithms*, SIAM J. Comput. **1** (1972), no. 2, 146–160.

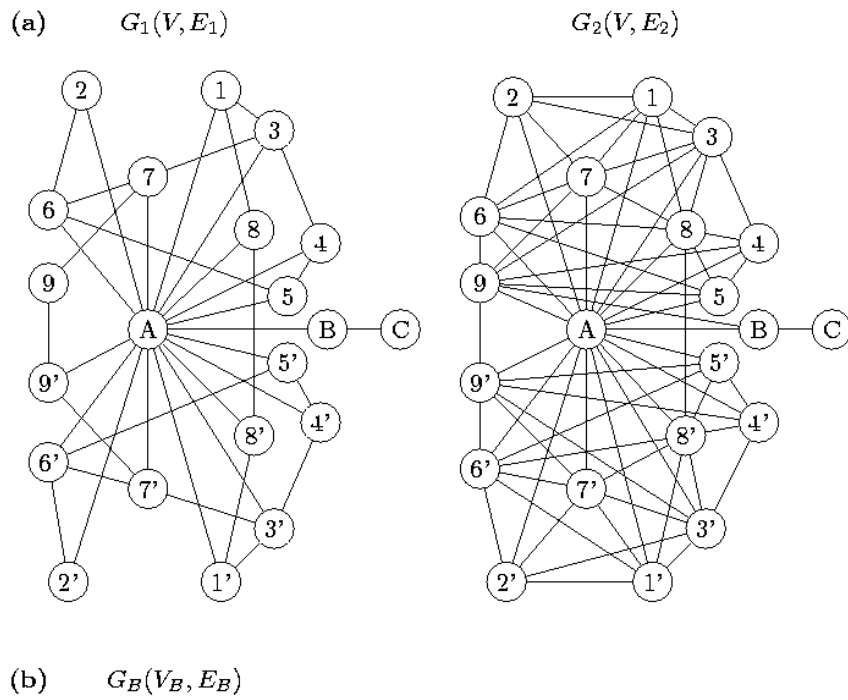


Figure 5: Counterexample 3 (to Lemma 5 [9])