A new upper bound for the Homogeneous Set Sandwich Problem

Celina M. H. de Figueiredo* Vinícius G. P. de Sá†

January 26, 2004

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. Two years ago, Tang *et al.* [8] proposed an interesting $O(\triangle_1 \cdot n^2)$ algorithm for this problem, which has been considered its most efficient algorithm since. We show the incorrectness of their algorithm and present a new deterministic algorithm for the Homogeneous Set Sandwich Problem, whose $O(m_1\overline{m_2})$ time complexity becomes its current upper bound.

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 $\overline{m_i}$. Additionally, the degree of vertex x in graph G_i is represented by $d_i(x)$, the degree of vertex x in G_i's complement $\overline{G_i}$ is represented by $\overline{d_i}(x)$ and, finally, $\triangle_i$ stands for G_i's maximum vertex degree.

Notwithstanding the existence of linear-time algorithms for solving the problem of finding homogeneous sets in a single graph [2, 3, 4, 5, 6, 7], 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.* [8] 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

*Instituto de Matemática and COPPE, Universidade Federal do Rio de Janeiro, Brazil.
†COPPE, Universidade Federal do Rio de Janeiro, Brazil.

HSSP's upper bound at $O(n^4)$. A careful study of the underlying ideas contained in both [1] and [8], though, has led us to the development of a faster deterministic algorithm, which establishes a new upper bound to the problem.

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. This result will be particularly useful in the analysis of our algorithm. In Section 3, we give a brief description of the BGC algorithm and point out where its basic flaw lies. Finally, Section 4 introduces our new algorithm. It is not altogether original, it must be said, for it bears essentially on the main ideas of both preceding algorithms.

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 [8].

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 $S \subseteq V$ 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 $B(S) \neq \varnothing$. 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 $B(S) = \varnothing$, 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. □

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 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 $B(H) = \varnothing$ 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]

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 $B(H)$ 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 $O(Min\{|S_1|, |S_2|\})$ 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 [8], 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

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 [8]

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 $[v,w]$ 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 [9] 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. [8] *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. [8] *The set H of vertices found in line 2 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 set S of vertices on the left of the dashed line constitutes an ESCC. (The bold edges in S stress the existence of cycles providing a path from each vertex in S to every other vertex in S. Notice, also, that all edges that come across the dashed line *reach* S, which makes an *end* strongly connected component out of it.) The set $H = \{1, 2, \ldots, 7\} \subset V$ that labels the vertices in S, 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 S (among other existing G_B's ESCCs) in line 2, it is likely to answer *yes* along with set $H = \{1, 2, \ldots, 7\}$, 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. In other words, there may exist two vertices $x, y \in V$ which appear in some labels of C's vertices, without vertex $[x, y] \in V_S$ being a necessary element of C. 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 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, 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 $\{1, 2, \ldots, 7\}$ (resp. $\{1', 2', \ldots, 7'\}$). 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. [8] *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 $H = \{1, 2, \ldots, 9, 1', 2', \ldots, 9'\}$ (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 $x, y \in \{1, 2, \ldots, 9, 1', 2', \ldots, 9'\}$. (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 $F = \{1, 2, \ldots, 7\}$ or $F' = \{1', 2', \ldots, 7'\}$, 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 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 A new $O(m_1\overline{m_2})$ algorithm

Theorem 6 that follows gives a correct characterization, based on Tang *et al.*'s bias graphs, of sandwich homogeneous sets. It corrects Lemmas 4 and 5 in a row, supporting the new HSSP algorithm which we will introduce, in this section.

Let $G_B(V_B, E_B)$ be the bias graph of input graphs $G_1(V, E_1), G_2(V, E_2)$. A subset $K \subseteq V_B$ is said to be a *pair-closed* set if and only if there do not exist two vertices $x, y \in V$, among those which label K's vertices, such that vertex $[x, y]$ is not an element of K. The set $A = \{[1, 2], [1, 3], [2, 3]\}$ is a pair-closed set. The set $B = \{[1, 2], [1, 3], [1, 4], [2, 3], [2, 4]\}$ is *not* pair-closed, for vertices 3 and 4 appear in the label of some vertices in B but $[3, 4] \notin B$.

Theorem 6. *A set $H \subset V$ is a sandwich homogeneous set of graphs $G_1(V, E_1), G_2(V, E_2)$ if and only if the pair-closed set $K = \{[x, y] \mid x, y \in H\} \subset V_B$ induce an* end *subgraph in bias graph G_B of (G_1, G_2).*

Proof. Let $K \subset V_B$ be the pair-closed set that holds all vertices $[v, w] \in V_B$ such that $v, w \in H \subset V$. Assume, by hypothesis, that K induces an end subgraph in G_B. Now suppose, by contradiction, that H is not a sandwich homogeneous set of (G_1, G_2). Then, H must have a bias vertex $b \in V \setminus H$, which means that there exists a mandatory edge between b and some vertex $h_1 \in H$ and also a forbidden edge between b and some other vertex $h_2 \in H$. But this implies that vertex $[h_1, h_2] \in K$ has outgoing edges to vertices $[h_1, b]$ and $[h_2, b]$, which cannot be in K. This is an absurd, for K induces an end subgraph. Conversely, if H is a sandwich homogeneous set, then it does not have any bias vertices. Consequently, if a pair of vertices $h_1, h_2 \in H$ has a bias vertex u then u also belongs to H. (Otherwise, u would be a bias vertex of H.) Once K is pair-closed in the vertices of H, vertices $[h_1, u], [h_2, u] \in V_B$ must belong to K, so that the subgraph of G_B induced by K does not have any outgoing edges. □

Theorem 6 does not lead directly to an efficient algorithm for the HSSP, for there is not any quick means of finding pair-closed sets which induce end subgraphs in the bias graph. Corollary 7, however, brings about the central inspiration for the algorithm that follows.

Corollary 7. *If $H \subset V$ is a sandwich homogeneous set of graphs $G_1(V, E_1)$, $G_2(V, E_2)$, then either the subgraph $G_B\langle K\rangle$, induced by the pair-closed set $K = \{[x, y] \mid x, y \in H\} \subset V_B$ in the bias graph $G_B(V_B, E_B)$ of (G_1, G_2), is itself an end strongly connected component or else it contains, properly, some end strongly connected component of G_B.*

Proof. From Theorem 6, we know that $G_B\langle K\rangle$ is an end subgraph of G_B. If it is strongly connected, then the statement holds trivially. If it is not, then there must exist two vertices $x, y \in G_B\langle K\rangle$ such that there is not any path from x to y. The set $R(x)$ of all vertices that are reachable from x certainly induces an end subgraph, for it cannot contain any outgoing edges to vertices $u \notin R(x)$, otherwise x would reach u. Also, $y \notin R(x)$, so that $R(x)$ is a proper subgraph of $G_B\langle K\rangle$. Thus, the fact that $G_B\langle K\rangle$ is not strongly connected implies that it contains some end, proper subgraph $G_B\langle K'\rangle$. This subgraph, in turn, must either be itself strongly connected (which would end the proof) or contain an end, proper subgraph $G_B\langle K''\rangle$, and so on and so forth. As G_B is finite, this ought to stop at some point, whereupon we will finally have found an end subgraph of $G_B\langle K\rangle$ which is strongly connected. □

The new algorithm we propose is rather simple. It can be regarded as either (i) an improved version of the EBE algorithm which just does not run the bias envelopment on all $O(n^2)$ input vertices' pairs, but on a shorter number of initial candidates, instead; or (ii) an improved version of the BGC algorithm, which only does not translate the bias graph's ESCCs directly into alleged sandwich homogeneous sets (for this is just not possible), but instead into subsets of the input vertices such that one of them has to be contained in a sandwich homogeneous set, in case there exists one. We prefer to consider it a hybrid between its two equally important predecessors.

The new algorithm has two distinct phases. We called it the Two-Phase algorithm (2-P algorithm, for short). The first phase of the 2-P algorithm builds the bias graph of the input instance, as in the BGC algorithm, and locates all its ESCCs $G_B\langle C_i\rangle$. Each of these ESCCs is then used to determine a subset H_i of the input vertices such that H_i contains all vertices which appear in the labels of the bias graph's vertices that belong to $G_B\langle C_i\rangle$. The second phase simply runs the bias envelopment procedure of the EBE algorithm on each of those subsets H_i, returning *yes* and a sandwich homogeneous set H that contains H_i, in case there exists one, or *no* in case none of the subsets H_i happen to be contained in any sandwich homogeneous sets of the input instance.

Figure 6 appropriately depicts the mechanics of the Two-Phase algorithm.

4.1 Proof of correctness / completeness

The validity of the 2-P algorithm as a HSSP solver comes directly from Corollary 7 and from the fact that the bias envelopment procedure rightly determines whether there exists a sandwich homogeneous set which contains a given subset of the input vertices [1]. Let $G_1(V, E_1), G_2(V, E_2)$ be the input graphs for the 2-P algorithm, and $G_B(V_B, E_B)$ their bias graph. First, if the algorithm returns *yes* and a sandwich homogeneous set H, then $1 < |H| < n$ and the bias graph of H is the empty set. Thus, by Theorem 1, H is a sandwich homogeneous set. Second, if the pair (G_1, G_2) admits a sandwich homogeneous set H, then, by Corollary 7, the subgraph $G_B\langle K\rangle$ which is induced in G_B by the pair-closed subset $K \subset V_B$ (comprising all vertices $[x, y]$ labelled by $x, y \in H$) contains an ESCC of G_B, say $G_B\langle C_i\rangle$. If all vertices in K belong to $G_B\langle C_i\rangle$, then the set H itself is the initial candidate of some bias envelopment iteration (lines 4 to 4.3.3). As its bias set is empty, the algorithm discovers (line 4.3.1) that it is a sandwich homogeneous set of (G_1, G_2) and stops. If, on the other hand, not all vertices in K belong to ESCC $G_B\langle C_i\rangle$ of $G_B\langle K\rangle$, then a subset $H_i \subseteq H$ is the initial candidate of some bias envelopment iteration. In this case, there will exist at least one sandwich homogeneous set, namely H, which contains H_i. So, the proof of completeness of the EBE algorithm [1] assures that the bias envelopment iteration which starts with candidate H_i can find a sandwich homogeneous set of (G_1, G_2).

4.2 Complexity analysis

The first phase of the 2-P algorithm takes $O(\triangle_1 \cdot n^2)$ time to build the bias graph $G_B(V_B, E_B)$ of input graphs $G_1(V, E_1), G_2(V, E_2)$ and locate its ESCCs [8]. Actually, the number of edges in E_B can be more precisely bounded by $O\left(n \cdot (m_1 + \overline{m_2})\right)$ than by $O(\triangle_1 \cdot n^2)$, as in [8]. It is true that the number of outgoing edges of a vertex $[x, y] \in V_B$ is twice the number of bias vertices of set $\{x, y\} \in V$. The point is, a vertex $b \in V$ is a bias vertex of $\{x, y\}$ only if edge (x, b) is mandatory and edge (y, b) is forbidden, or vice-versa. In other words, either (i) $(x, b) \in E_1$ and $(y, b) \notin E_2$ or (ii) $(y, b) \in E_1$ and $(x, b) \notin E_2$. The number of bias vertices of $\{x, y\}$ is therefore $O(Min\{d_1(x), \overline{d_2}(y)\} + Min\{d_1(y), \overline{d_2}(x)\}) = O(d_1(x) + \overline{d_2}(x) + d_1(y) + \overline{d_2}(y))$, which consequently bounds the number of outgoing edges of vertex $[x, y] \in V_B$. Thus, the total number of edges in E_B is

$$
\begin{aligned}
& O\left(\sum_{[x,y]\in V_B} d_1(x) + \overline{d_2}(x) + d_1(y) + \overline{d_2}(y)\right) = \\
= \; & O\left(n \cdot \sum_{x \in V} d_1(x) + \overline{d_2}(x)\right) = \\
= \; & O\left(n \cdot (m_1 + \overline{m_2})\right).
\end{aligned}
$$

As the time complexity of all steps in the first phase is bounded by the number of edges in the bias graph [8], we arrive at a final complexity of $O\left(n \cdot (m_1 + \overline{m_2})\right)$ for the whole first phase.

The second phase of the 2-P algorithm runs several successive bias envelopments (one for each ESCC of the bias graph, in the worst case). The time demanded by each bias envelopment is, as we saw in the end of Section 2, $O(m_1 + \overline{m_2})$. The question is, how many times, in the worst case, does the bias envelopment procedure have to be run? That is, what is the maximum size of a sequence of ESCC-associated input vertices' subsets that fail to be contained in any sandwich homogeneous sets of (G_1, G_2)?

The answer comes straight from Theorem 6, which grants that all end subgraphs of G_B whose vertices perform a pair-closed set are associated to a sandwich homogeneous set. Consequently, the ESCC $G_B\langle C_i\rangle$, whose associated set $H_i \in V$ fails to be contained in any sandwich homogeneous sets, *has to be* induced by a *non*-pair-closed vertex set $C_i \subset V_B$. The number of executions of the bias envelopment procedure is therefore bounded by the maximum number of ESCCs, in the bias graph, that are induced by non-pair-closed sets of vertices.

Lemma 8 establishes an upper bound to such ESCCs and allows us to determine the total time complexity of the 2-P algorithm.

Lemma 8. *In any bias graph, there are $O(Min\{m_1, \overline{m_2}\})$ end strongly connected components that are induced by non-pair-closed vertex sets.*

Proof. Let $G_B(V_B, E_B)$ be the bias graph of graphs $G_1(V, E_1), G_2(V, E_2)$ and let $C_i \subset V_B$ be a non-pair-closed vertex set that induces ESCC $G_B\langle C_i \rangle$ in G_B. Now let $[x, y]$ be a vertex that belongs to C_i. Vertex $[x, y]$ necessarily presents an outgoing edge, otherwise $[x, y]$ would induce an ESCC and could not be properly contained in C_i. (Clearly, $C_i \setminus \{[x, y]\}$ is nonempty, for $\{[x, y]\}$ *is* pair-closed, *whereas* C_i *is not.*) Let $([x, y], [x, t])$ be an edge in $G_B\langle C_i \rangle$. Vertex $t \in V$ is therefore a bias vertex of set $\{x, y\} \subset V$, so that edge (x, t) is mandatory and edge (y, t) is forbidden (or vice-versa). Without loss of generality, let edge (x, t) be the mandatory one. We define a function $l_M(C)$ that associates ESCC $G_B\langle C \rangle$ with such a mandatory edge (i.e. a mandatory edge, belonging to the input instance, which is necessary for the existence of some bias relationship that appears inside $G_B\langle C \rangle$). This chosen mandatory edge, returned by $l_M(C)$, becomes $G_B\langle C \rangle$'s *label*. In the current example, "(x, t)" is a possible label for $G_B\langle C_i \rangle$. Notice that no other ESCC $G_B\langle C_j \rangle$ can possibly be assigned the same label "(x, t)". Otherwise, because of the way a label is chosen by l_M, there would necessarily have to be an edge $([x, w], [x, t])$ (or $([t, w], [x, t])$) in $G_B\langle C_j \rangle$, for some $w \in V$. Because $G_B\langle C_j \rangle$ is an ESCC, $[x, t]$ must be in $G_B\langle C_j \rangle$. But this is an absurd, for $[x, t]$ already belongs to $G_B\langle C_i \rangle$ and the intersection between two ESCCs has to be empty (recall that a digraph's strongly connected components constitute a partition of its vertex set). Thus, labelling function l_M is bijective, for every ESCC $C_i \subset V_B$ can be designated a label and no two distinct ESCCs can share the same label. But every distinct label generated by l_M depends on the existence of a mandatory edge, which implies that the number of ESCCs that are induced by non-pair-closed vertex sets is bounded by the number of mandatory edges, namely m_1. We reason that a similar labelling bijective function l_F, which only names each non-pair-closed ESCC after some *forbidden* edge (instead of a mandatory one, as in function l_M) implies that the number of such special ESCCs is also bounded by the number of forbidden edges in the input instance, namely $\overline{m_2}$, which ends the proof. □

The time complexity of the second phase of the 2-P algorithm is therefore $O\left(Min\{m_1, \overline{m_2}\} \cdot (m_1 + \overline{m_2})\right)$.

We remark that the time complexity of the first phase — $O(n \cdot (m_1 + \overline{m_2}))$ — has been clearly overtaken by the second's, for $n = O(Min\{m_1, \overline{m_2}\})$. Otherwise, either $m_1 < n$ (implying that G_1 is not connected) or $\overline{m_2} < n$ (implying that G_2's complement $\overline{G_2}$ is not connected) would characterize trivial HSSP instances, for the existence of a vertex $v \in V$ with either no incident edges in G_1 or $n-1$ incident edges in G_2 effortlessly testifies the existence of a trivial $(n-1)$-vertex sandwich homogeneous set of (G_1, G_2), namely $V \setminus \{v\}$.

We finally rewrite the time complexity of the whole 2-P algorithm as follows:

$$\begin{aligned} & O\left(Min\{m_1, \overline{m_2}\} \cdot (m_1 + \overline{m_2})\right) = \\ = \; & O\left(Min\{m_1, \overline{m_2}\} \cdot Max\{m_1, \overline{m_2}\}\right) = \\ = \; & O\left(m_1 \overline{m_2}\right). \end{aligned}$$

5 Conclusion

In this article, we invalidated the current upper bound for the Homogeneous Set Sandwich Problem, which had been set too low by an incorrect, recently published algorithm. Also, we presented a new $O\left(m_1 \overline{m_2}\right)$ algorithm, which is better than all previously known ones.

An open question is whether $O(Min\{m_1, \overline{m_2}\})$ is a good bound for the number of non-pair-closed ESCCs in a bias graph. So far we still have not succeeded in finding any HSSP instance whose bias graph had a number of non-pair-closed ESCCs that was not $O(n \log n)$, clearly a better (albeit unproven) bound. For the time being, either a proof that $O(n \log n)$ is indeed an upper bound or the exhibition of an instance with more ESCCs in its bias graph shall be equally welcome.

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] 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.

[4] R. M. McConnel and J. Spinrad, *Modular decomposition and transitive orientations*, Discrete Math. **201** (1999), 189–241.

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

[6] B. Reed, *A semi-strong perfect graph theorem*, Ph.D. thesis, School of Computer Science, McGill University, Montreal, 1986.

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

[8] S. Tang, F. Yeh, and Y. Wang, *An efficient algorithm for solving the homogeneous set sandwich problem*, Information Processing Letters **77** (2001), 17–22.

[9] R. E. Tarjan, *Depth-first search and linear graph algorithms*, SIAM J. Comput. **1** (1972), no. 2, 146–160.

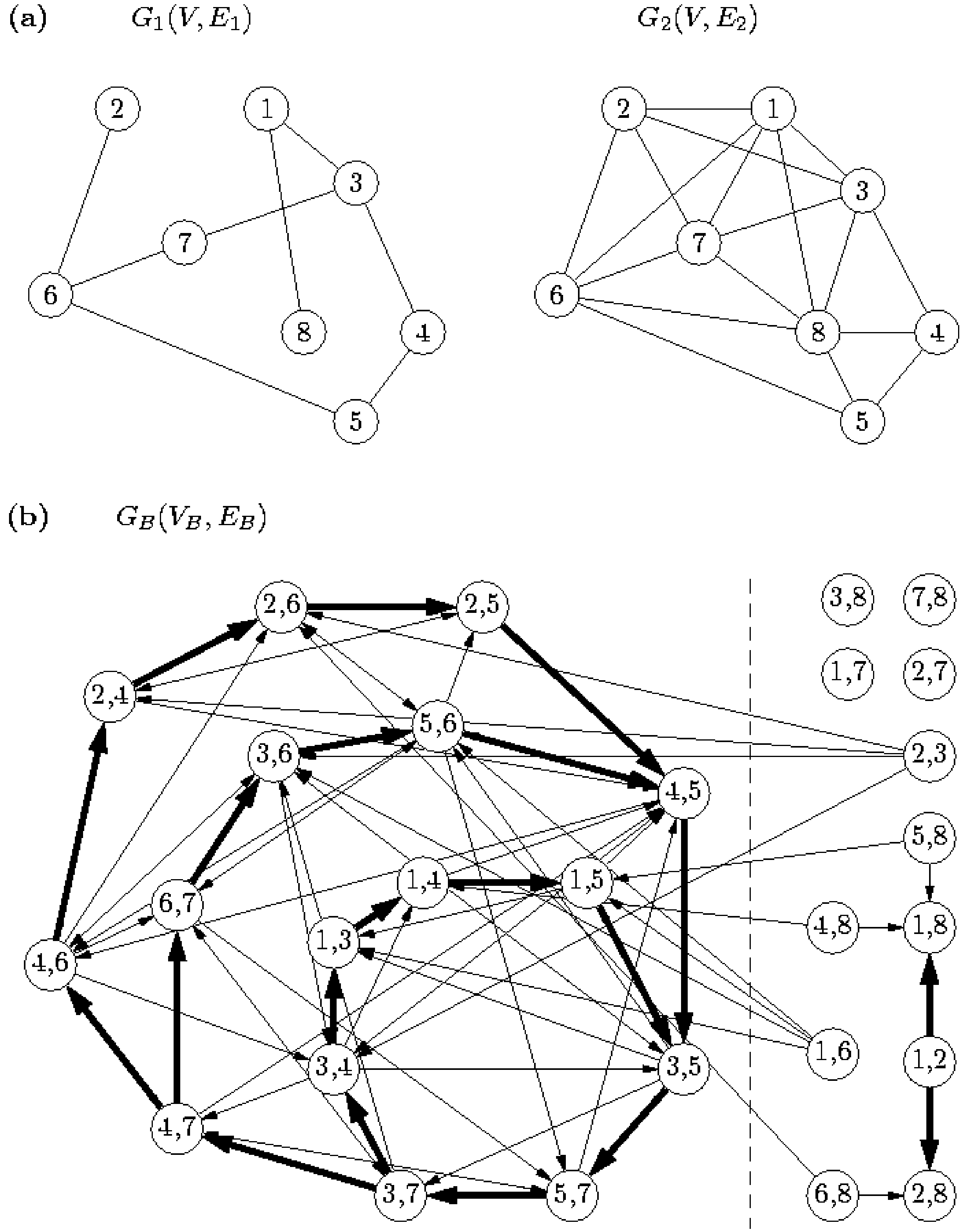

Figure 3: Counterexample 1 (to Lemma 4 [8])

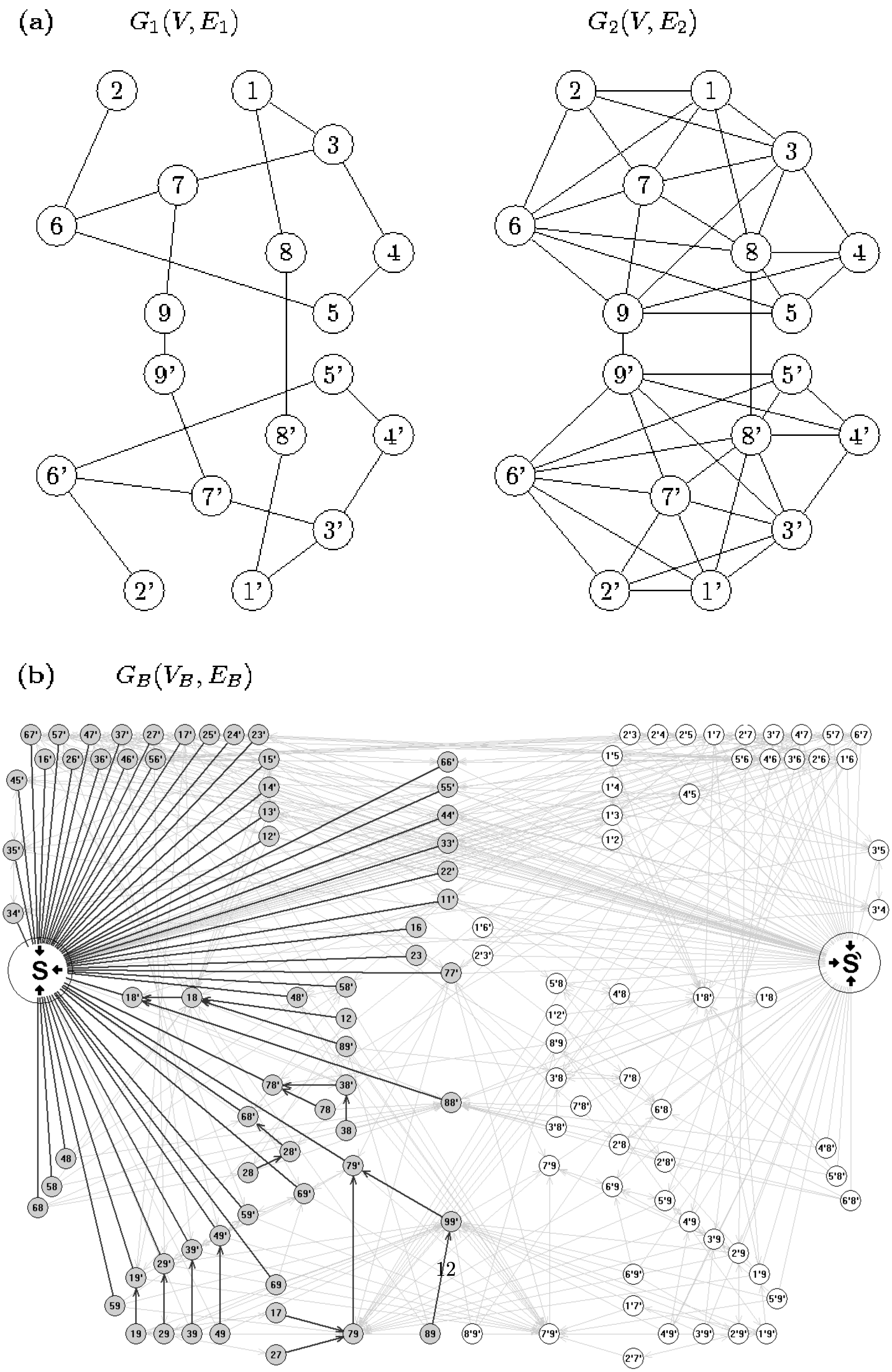

Figure 4: Counterexample 2 (to Lemma 4 [8])

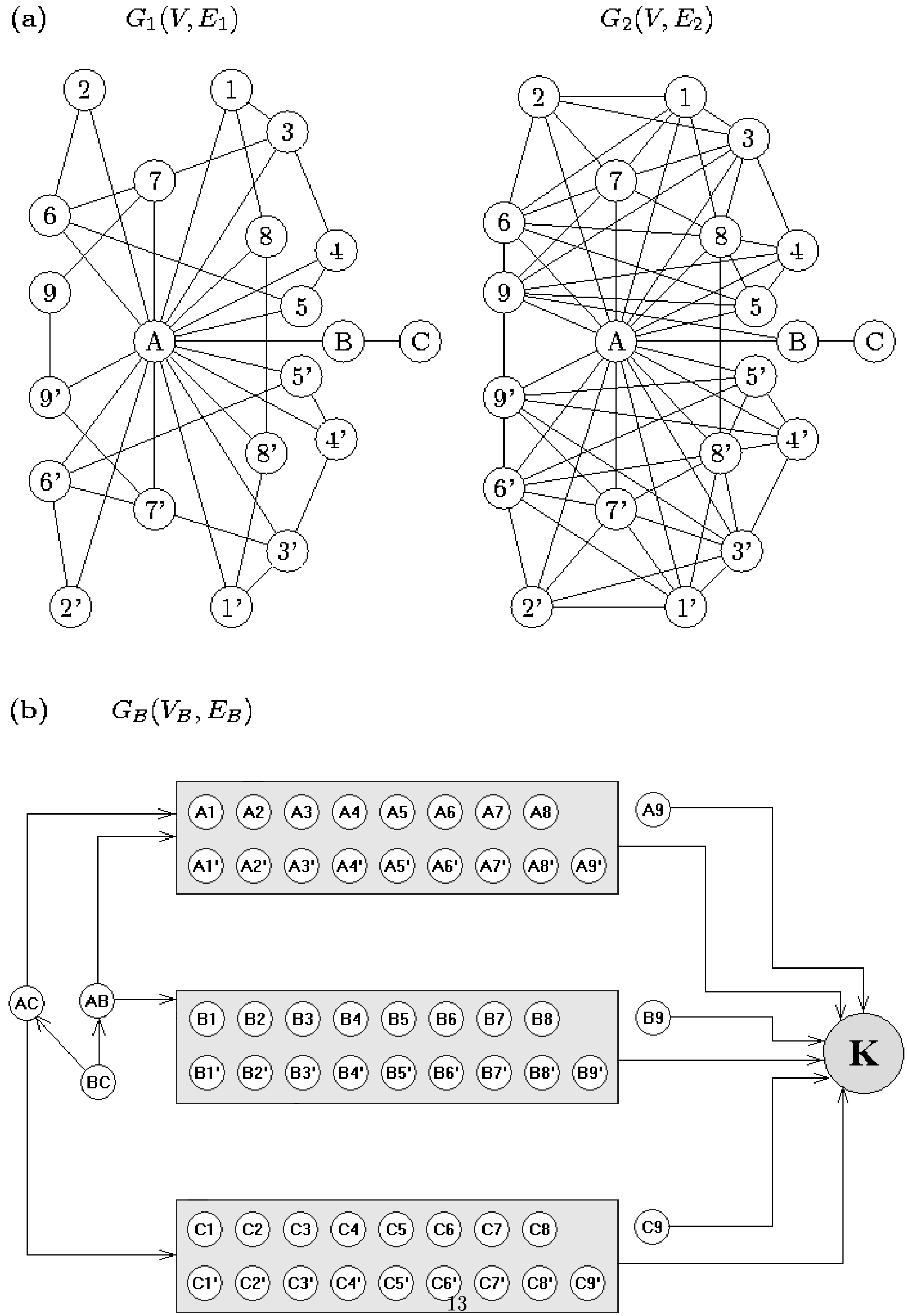

Figure 5: Counterexample 3 (to Lemma 5 [8])

The Two-Phase 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 all end strongly connected component $G_B\langle C_i \rangle$ of G_B.
3. Let H_i denote the set of vertices in V that label the vertices in $G_B\langle C_i \rangle$.
4. For each set $H_i \subset V$ do
4.1. $H \leftarrow H_i$.
4.2. Find the bias set $B(H)$.
4.3. While $H \neq V$ do
4.3.1. If $B(H) = \varnothing$ then return *yes* and H. End.
4.3.2. $H \leftarrow H \cup B(H)$.
4.3.3. Update $B(H)$.
5. Return *no*.

Figure 6: The 2-P algorithm