Faster deterministic and randomized algorithms on the Homogeneous Set Sandwich Problem

Celina M. H. de Figueiredo[1], Guilherme D. da Fonseca[1], Vinícius G. P. de Sá[1], and Jeremy Spinrad[2]

[1] Instituto de Matemática and COPPE, Universidade Federal do Rio de Janeiro, Brazil.
[2] Vanderbilt University, USA.

Abstract. A homogeneous set is a non-trivial, proper subset of a graph's vertices such that all its elements present exactly the same outer neighborhood. Given two graphs, $G_1(V, E_1)$, $G_2(V, E_2)$, we consider the problem of finding a sandwich graph $G_s(V, E_S)$, with $E_1 \subseteq E_S \subseteq E_2$, which contains a homogeneous set, in case such a graph exists. This is called the Homogeneous Set Sandwich Problem (HSSP). We give an $O(n^{3.5})$ deterministic algorithm, which updates the known upper bounds for this problem, and an $O(n^3)$ Monte Carlo algorithm as well. Both algorithms, which share the same underlying idea, are quite easy to be implemented on the computer.

1 Introduction

Given two graphs $G_1(V, E_1)$, $G_2(V, E_2)$ such that $E_1 \subseteq E_2$, a sandwich problem with input pair (G_1, G_2) consists in finding a *sandwich graph* $G_s(V, E_S)$, with $E_1 \subseteq E_S \subseteq E_2$, which has a desired property Π [3]. In this paper, the property we are interested in is the ownership of a homogeneous set. A *homogeneous set* H, in a graph $G(V, E)$, is a subset of V such that *(i)* $2 \leq |H| \leq |V| - 1$ and *(ii)* for all $v \in V \setminus H$, either $(v, v') \in E$ is true for all $v' \in H$ or $(v, v') \notin E$ is true for all $v' \in H$. In other words, a homogeneous set H is a subset of V such that the outside-H neighborhood of all its vertices is the same and which also satisfies the necessary, above mentioned size constraints. A *sandwich homogeneous set* of a pair (G_1, G_2) is a homogeneous set for at least one among all possible sandwich graphs for (G_1, G_2).

There are many algorithms which find homogeneous sets quickly in a single graph. The most efficient one is due to McConnell and Spinrad [4] and has $O(|E|)$ time complexity.

On the other hand, the known algorithms for the homogeneous set sandwich problem are far less efficient. The first polynomial time algorithm was presented by Cerioli *et al.* [1] and has $O(n^4)$ time complexity (where $n = |V|$, as throughout the whole text). We refer to it as the *Exhaustive Bias Envelopment algorithm* (EBE algorithm, for short), as in [2]. An $O(\Delta n^2)$ algorithm (where Δ stands for the maximum vertex degree in G_1) has been found by Tang *et al.* [6], but in [5, 2] it is proved incorrect. Although all efforts to correct Tang *et al.*'s algorithm (referred to as the *Bias Graph Components algorithm*, in [2]) have been in vain, some of its ideas were used, in [5, 2], to build a hybrid algorithm, inspired by both [1] and [6]. This one has been called the *Two-Phase algorithm* (2-P algorithm, for short) and currently sets the HSSP's upper bounds at its $O(m_1\overline{m_2})$ time complexity, where m_1 and $\overline{m_2}$ respectively refer to the number of edges in G_1 and the number of edges *not* in G_2.

After defining some concepts and auxiliary procedures in Section 2, we present, in Section 3, a new $O(n^{3.5})$ deterministic algorithm for the HSSP. It offers a good alternative to the *2-P* algorithm (whose time complexity is not better than $O(n^4)$ if we express it only as a function of n) when dealing with dense input graphs, whereas the *2-P* would remain the best choice when sparse graphs are dealt with. Besides, Section 4 is devoted to a fast, randomized Monte Carlo algorithm, which solves this problem in $O(n^3)$ time with whatever desired error ratio.

2 Bias Envelopment

We define the *bias set* $B(H)$ of a vertex subset H as the set of vertices $b \notin H$ such that $(b, v_i) \in E_1$ and $(b, v_j) \notin E_2$, for some $v_i, v_j \in H$. Such vertices b are called *bias vertices* of the set H [6]. It is easy to see that H, with $2 \leq |H| \leq n-1$, is a sandwich homogeneous set if and only if $B(H) = \varnothing$.

It is proved in [1] that any sandwich homogeneous set containing the set of vertices H should also contain $B(H)$. This result, along with the fact that no homogeneous sets are allowed with less than two vertices, gave birth in that same paper to a procedure which we call *Bias Envelopment* [2]. The Bias Envelopment procedure answers whether a given pair of vertices is contained in any sandwich homogeneous sets of the input instance. The procedure starts from an initial sandwich homogeneous set candidate $H_1 = \{v_1, v_2\}$ and successively computes $H_{t+1} = H_t \cup B(H_t)$ until either *(i)* $B(H_t) = \varnothing$, whereby H_t is a sandwich homogeneous set and it answers *yes*; or *(ii)* $|H_t| + |B(H_t)| = n$, when it answers *no*, meaning that there is no sandwich homogeneous set containing $\{v_1, v_2\}$. The Bias Envelopment procedure runs in $O(n^2)$ time, granted some appropriate data structures are used, as described in [1].

The EBE algorithm, presented in [1], tries to find a sandwich homogeneous set exhaustively, running the Bias Envelopment procedure on all $n(n-1)/2$ pairs of the input graphs' vertices, in the worst case. Thus, the time complexity of the EBE algorithm is $O(n^4)$.

Both algorithms we introduce in this paper are based on a variation of the Bias Envelopment procedure, which we call the *Incomplete Bias Envelopment*. The input of the Incomplete Bias Envelopment is a pair of vertices $\{v_1, v_2\}$ and a *stop parameter* $k < n$. The only change in this incomplete version is that, whenever $|H_t| > k$, the envelopment stops prematurely, answering *no* and rejecting $\{v_1, v_2\}$. Notice that a *no* answer from the Incomplete Bias Envelopment with parameter k means that $\{v_1, v_2\}$ is not contained in any homogeneous sets

```
IncompleteBiasEnvelopment $(G_1(V, E_1), G_2(V, E_2), v_1, v_2, k)$

1.       $H \leftarrow \{v_1, v_2\}$
2.       while $|H| \leq k$
2.1.         if $B(H) = \varnothing$ and $|H| < |V|$
2.1.1.           return $H$ and yes   // a sandwich homogeneous set was found.
2.2.         else
2.2.1.           $H \leftarrow H \cup B(H)$
3.       return no   // there are no sandwich homogeneous sets with $k$ vertices
                     // or less which contain $\{v_1, v_2\}$.
```

Fig. 1. The Incomplete Bias Envelopment procedure.

of size at most k. Using the same data structures as in [1], the Incomplete Bias Envelopment runs in $O(nk)$ time.

The Incomplete Bias Envelopment generalizes its complete version, as a normal Bias Envelopment is equivalent to an Incomplete Bias Envelopment with parameter $k = n$.

The pseudo-code for the Incomplete Bias Envelopment is in Figure 1.

3 The Balanced Subsets Algorithm

The algorithm we propose in this section is quite similar to the EBE algorithm, in the sense that it submits each of the input vertices' pairs to the process of Bias Envelopment. The only difference is that this algorithm establishes a particular order in which the vertex pairs are chosen, in such a way that it can benefit, at a certain point, from unsuccessful envelopments that have already taken place. After some unsuccessful envelopments, a number of vertex pairs have been found not to be contained in any sandwich homogeneous sets. This knowledge is then made useful by the algorithm, which will stop further envelopments earlier by means of calling *Incomplete* Bias Envelopments instead of complete ones, saving relevant time without loss of completeness.

When the algorithm starts, it partitions all n vertices of the input graphs into $O(\sqrt{n})$ disjoint subsets C_i of size $O(\sqrt{n})$, each. Then all pairs of vertices will be submitted to Bias Envelopment in two distinct phases: in the first phase, all pairs consisting of vertices from the same subset C_i are bias enveloped (and only those); in the second phase, all remaining pairs (i.e. those comprising vertices that are not from the same subset C_i) are then bias-enveloped. In the end, all possible vertex pairs will have been checked out as to belong or not to some sandwich homogeneous set from the input instance, just like in the EBE algorithm. The point is: if all Bias Envelopments in the first phase fail to find a sandwich homogeneous set, then the input instance does not admit any sandwich homogeneous sets which contain two vertices from the same subset C_i. Thence, the

BalancedSubsetsHSSP $(G_1(V, E_1), G_2(V, E_2))$

```
1.       label all vertices in $V$ from $v_1$ to $v_n$.
2.       create $\lceil \sqrt{n} \rceil$ empty sets $C_i$.
3.       for each vertex $v_j \in V$, do $C_{j \textbf{ modulo} \lceil \sqrt{n} \rceil} = C_{j \textbf{ modulo} \lceil \sqrt{n} \rceil} \cup \{v_j\}$.
4.       for each pair of vertices $\{x, y\}$ in the same subset $C_i$
4.1.         if BiasEnvelopment($G_1, G_2, x, y$) = yes
4.1.1.           return yes.
5.       for each pair of vertices $\{x, y\}$ not in the same subset $C_i$
5.1.         if IncompleteBiasEnvelopment($G_1, G_2, x, y, \lceil \sqrt{n} \rceil$) = yes
5.1.1.           return yes.
6.       return no.
```

Fig. 2. The Balanced Subsets algorithm for the HSSP.

maximum size of any possibly existing homogeneous set is $O(\sqrt{n})$ (the number of subsets into which the vertices had been dispersed), which grants that all Bias Envelopments of the second phase need not search for large homogeneous sets! That is why an Incomplete Bias Envelopment with stop parameter $k = O(\sqrt{n})$ can be used instead.

Figure 2 illustrates the pseudo-code for the Balanced Subsets algorithm.

Theorem 1 *The Balanced Subsets algorithm correctly answers whether there exists a sandwich homogeneous set in the input graphs.*

Proof. If the algorithm returns *yes*, then it has successfully found a set $H \subset V$, with $2 \leq |H| \leq |V| - 1$, such that the bias set of H is empty. Thus, H is indeed a valid sandwich homogeneous set.

Now, suppose the input has a sandwich homogeneous set H. If $|H| > \lceil \sqrt{n} \rceil$ then there are more vertices in H than subsets into which all input vertices were spread, in the beginning of the algorithm (line 3). Thus, by the pigeon hole principle, there must be two vertices $x, y \in H$ which were assigned to the same subset C_i. So, whenever $\{x, y\}$ is submitted to Bias Envelopment (line 4), the algorithm is doomed to find a sandwich homogeneous set. On the other hand, if $|H| \leq \lceil \sqrt{n} \rceil$, then it is possible that H does not contain any two vertices from the same subset C_i, which would cause all Bias Envelopments of the first phase (line 4.1) to fail. In this case, however, when a pair $\{x, y\} \subseteq H$ happens to be bias enveloped in line 5, the Incomplete Bias Envelopment is meant to be successful, for the size of H is, by hypothesis, less than or equal its stop parameter $k = \lceil \sqrt{n} \rceil$. □

3.1 Complexity Analysis

As each subset C_i has $O(n)$ pairs of vertices and there are $O(\sqrt{n})$ such subsets, the number of pairs that are bias enveloped in the first phase of the algorithm (line 4) is $O(n\sqrt{n})$. All Bias Envelopments, in this phase, are complete and take $O(n^2)$ time to be executed, which yields a subtotal of $O(n^{3.5})$ time in the whole first phase.

The number of pairs that are only submitted to Bias Envelopment in the second phase (line 5) is $O(n^2) - O(n\sqrt{n}) = O(n^2)$ pairs. Each Bias Envelopment is, now, an incomplete one with parameter $k = \lceil \sqrt{n} \rceil = O(\sqrt{n})$. Because the time complexity of each Incomplete Bias Envelopment with parameter k is $O(nk)$, then the total time complexity of the whole second phase of the algorithm is $O(n^3 k) = O(n^{3.5})$.

Thus, the overall time complexity of the Balanced Subsets algorithm is $O(n^{3.5})$.

4 The Monte Carlo Algorithm

An yes-biased Monte Carlo algorithm for a decision problem is one which always answers *no* when the correct answer is *no* and which answers *yes* with probability p whenever the correct answer is *yes*.

In order to gather some intuition, let us suppose the input has a sandwich homogeneous set H with h vertices or more.

What would be, in this case, the probability $\overline{p_1}$ that a random pair of vertices $\{x, y\} \in V$ is *not* contained in H? Clearly,

$$\overline{p_1} \leq 1 - \frac{h(h-1)}{n(n-1)}.$$

What about the probability $\overline{p_t}$ that t random pairs of vertices fail to be contained in H? It is easy to see that

$$\overline{p_t} \leq \left(1 - \frac{h(h-1)}{n(n-1)}\right)^t.$$

Now, what is the probability p_t that, after t Bias Envelopment procedures have been run (starting from t randomly chosen pairs of vertices), a sandwich homogeneous set have been found? Again, it is quite simple to reach the following expression, which will be vital to the forthcoming reasoning.

$$p_t \geq 1 - \left(1 - \frac{h(h-1)}{n(n-1)}\right)^t. \tag{1}$$

If, instead of obtaining the probability p_t from the expression above, we fix p_t at some desired value $p = 1 - \epsilon$, we will be able to calculate the minimum integer value of h_t (which will denote h as a function of t) that satisfies the inequality 1. This value h_t is such that the execution of t independent Bias Envelopment procedures (on t random pairs) *is sufficient to find a sandwich homogeneous set of the input instance with probability at least p, in case there exists any with h_t vertices or more* (see equation 2):

$$h_t = \left\lfloor \frac{1 + \sqrt{1 + 4(n^2 - n)(1 - (1 - p)^{1/t})}}{2} \right\rfloor . \tag{2}$$

However, we want an algorithm that finds a sandwich homogeneous set with some fixed probability p *in case there exists any, no matter its size.* But as h_t decreases with the growth of t, the following question arises: how many random pairs do we need to submit to Bias Envelopment in order to achieve that? The answer is rather simple: the minimum integer t' such that $h_{t'} = 2$, for 2 is the shortest possible size of a sandwich homogeneous set!

Determining t' comes straightforwardly from equation 2 (please refer to Section 4.1 for the detailed calculations):

$$t' = \frac{\ln(1 - p)}{\ln\left(1 - \frac{2}{n(n-1)}\right)} = \Theta(n^2). \tag{3}$$

Once the number t' of Bias Envelopment procedures that need to be undertaken on randomly chosen pairs of vertices is $\Theta(n^2)$ and the time complexity of each Bias Envelopment is $O(n^2)$, so far we seem to have been lead to an $O(n^4)$ randomized algorithm, which is totally undesirable, for we could already solve the problem *deterministically* with less asymptotical effort (see Section 3)!

Now we have come to a point where the *incomplete* version of the Bias Envelopment procedure will play an essential role as far as time saving goes. We show that, by the time the t-th Bias Envelopment is run, its incomplete version with stop parameter $k = h_{t-1}$ serves exactly the same purpose as its complete version would do.

Lemma 2 *In order to find a sandwich homogeneous set, with probability p, in case there exists any with h_t vertices or more, the t-th Bias Envelopment need not go further when the size of the candidate set has exceeded h_{t-1}.*

Proof. Two are the possibilities regarding the input: *(i)* there is a sandwich homogeneous set with more than h_{t-1} vertices; or *(ii)* there are no sandwich homogeneous sets with more than h_{t-1} vertices.

If *(i)* is true, then no more than $t - 1$ Bias Envelopments would even be required to achieve that. Hence the t-th Bias Envelopment can stop as early as it pleases.

If *(ii)* is the case, then an Incomplete Bias Envelopment with stop parameter $k = h_{t-1}$ is meant to give the exact same answer as the complete Bias Envelopment would, for there are no sandwich homogeneous sets with more than h_{t-1} vertices to be found.

Whichever the case, thus, such an Incomplete Bias Envelopment is perfectly sufficient. $\square$

Now we can describe an efficient Monte Carlo algorithm which gives the correct answer to the HSSP with probability at least p.

```
MonteCarloHSSP $(G_1(V, E_1), G_2(V, E_2), p)$

1.      $h \leftarrow |V|$
2.      $t \leftarrow 1$
3.      while $h \geq 2$
3.1.        $(v_1, v_2) \leftarrow$ random pair of distinct vertices of $V$
3.2.        if IncompleteBiasEnvelopment$(G_1, G_2, v_1, v_2, h)$ = yes
3.2.1.          return yes
3.3.        $h \leftarrow \lfloor (1 + \sqrt{1 + 4(|V|^2 - |V|)(1 - (1-p)^{1/t})})/2 \rfloor$
3.4.        $t \leftarrow t + 1$
4.      return no
```

Fig. 3. The Monte Carlo algorithm for the HSSP.

The algorithm's idea is to run several Incomplete Bias Envelopment procedures on randomly chosen initial candidate sets (pairs of vertices). At each iteration t of the algorithm we run an Incomplete Bias Envelopment with stop parameter $k = h_{t-1}$ and either it succeeds in finding a sandwich homogeneous set (and the algorithm stops with an *yes* answer) or else it aborts the current envelopment whenever the number of vertices in the sandwich homogeneous set candidate exceeds the h_{t-1} threshold. (In this case, Lemma 2 grants its applicability.) For the first iteration, the stop parameter k is set to $h_0 = n$, as the first iteration corresponds to a complete Bias Envelopment. At the end of each iteration, the value of h_t is then updated (see equation 2), which makes it progressively decrease throughout the iterations until it reaches 2 (the minimum size allowed for a homogeneous set), which necessarily happens after $\Theta(n^2)$ iterations (see equation 3).

The pseudo-code for this algorithm is in Figure 3.

Theorem 3 *The Monte Carlo HSSP algorithm correctly answers whether there exists a sandwich homogeneous set in the input graphs with probability at least p.*

Proof. If the algorithm returns *yes*, then it is the consequence of having found a set $H \subset V$, with $2 \leq |H| \leq |V| - 1$, such that the bias set of H is empty, which makes a valid sandwich homogeneous set out of H. In other words, if the correct answer is *no* then the algorithm gives a correct *no* answer with probability 1.

If the correct answer is *yes*, we want to show that it gives a correct *yes* answer with probability p. Let h^* be the size of the largest sandwich homogeneous set of the input instance. As $h_0 = n$ and the algorithm only answers *no* after h_t has lowered down to 2, there must exist an index d such that $h_d \leq h^* < h_{d-1}$. From the definition of h_t we know that, on the hypothesis that the input has a sandwich homogeneous set with h_t vertices or more, t Bias Envelopments are sufficient to find one, with probability at least p. As, by hypothesis, there is a sandwich homogeneous set with $h^* \geq h_d$ vertices, then d independent Bias Envelopments are sufficient to find a sandwich homogeneous set with probability p. So, it is enough to show that this quota of d Bias Envelopments is achieved. It is true that Incomplete Bias Envelopments that stop *before* the candidate set has reached the size of h^* cannot find a sandwich homogeneous set with h^* vertices. Nevertheless, the first d iterations alone perform this minimum quota of Bias Envelopments. Because h^* is the size of the largest sandwich homogeneous set, the fact of being *incomplete* simply does not matter for these first d Bias Envelopments, none among which being allowed to stop before the size of the candidate set has become larger than $h_{d-1} > h^*$. □

4.1 Complexity Analysis

The first iteration of the algorithm runs the complete Bias Envelopment in $O(n^2)$ time [1]. (Actually, a more precise bound is given by $O(m_1 + \overline{m_2})$ [2], but, as the complexities of the Incomplete Bias Envelopment procedures do not benefit at all from having edge quantities in their analysis, we prefer to write time bounds only as functions of n, however.) The remaining iterations take $O(nh_t)$ time each. To analyze the time complexity of the algorithm, we have to calculate

$$\sum_{t=1}^{t'} O\left(nh_{t-1}\right),$$

where t' is the number of iterations in the worst case.

The value of h_t, obtained at the end of iteration t, is defined by equation 2. To calculate t', we replace $h_{t'}$ for 2 and have

$$\left(1 - \frac{2}{n(n-1)}\right)^{t'} = 1 - p, \text{ and finally}$$

$$t' = \frac{\ln(1-p)}{\ln\left(1 - \frac{2}{n(n-1)}\right)}.$$

For $0 < x < 1$, it is known that

$$\ln(1-x) = -x - \frac{x^2}{2} - \frac{x^3}{3} - \cdots.$$

Consequently,

$$t' = \frac{\ln(1-p)}{-\frac{2}{n(n-1)} - \frac{1}{\Theta(n^4)}} = \ln\frac{1}{\epsilon}\Theta(n^2) = \Theta(n^2).$$

Now, we will show that $q = h(h-1)/n(n-1) \geq h^2/2n^2$. This result is useful to simplify some calculations. We have

$$\frac{n}{n-1} \cdot \frac{h-1}{h} \cdot \frac{h^2}{n^2} = \frac{h(h-1)}{n(n-1)}, \text{ and}$$

$$\frac{h-1}{h} \cdot \frac{h^2}{n^2} \le \frac{h(h-1)}{n(n-1)}.$$

Since $h \ge 2$,

$$\frac{h^2}{2n^2} \le \frac{h(h-1)}{n(n-1)} = q.$$

To calculate the total time complexity, we replace $h(h-1)/n(n-1)$ for $h^2/2n^2$ and p_t for the fixed value p in equation 1, and have

$$\left(1 - \frac{h^2}{2n^2}\right)^t \ge 1 - p,$$

$$\frac{h^2}{2n^2} \le 1 - (1-p)^{1/t}, \text{ and}$$

$$h \le \Theta(n)\sqrt{1 - (1-p)^{1/t}}.$$

It is well known that

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots.$$

Consequently, for $x > 1$,

$$e^{1/x} = 1 + 1/\Theta(x).$$

Using this approximation, we have

$$h \le \Theta(n)\sqrt{1 - (1 + 1/\Theta(t))} = \Theta(n)/\Theta(\sqrt{t}).$$

The total time complexity of the algorithm is

$$\sum_{t=1}^{\Theta(n^2)} O\big(nh_t\big) = \sum_{t=1}^{\Theta(n^2)} \frac{O(n^2)}{O(\sqrt{t})} = O(n^2) \sum_{t=1}^{\Theta(n^2)} 1/O(\sqrt{t}).$$

Using elementary calculus, we have

$$\sum_{t=1}^{\Theta(n^2)} 1/O(\sqrt{t}) = O(n).$$

Consequently, the total time complexity of the algorithm is $O(n^3)$.

5 Conclusion

In this article, we presented two efficient algorithms for the Homogeneous Set Sandwich Problem: the first was an $O(n^{3.5})$ deterministic algorithm and the other, an $O(n^3)$ Monte Carlo one. The best results so far had been $O(n^4)$, if only functions of n are used to express time complexities.

A natural step, after having developed such a Monte Carlo algorithm, is often the development of a related Las Vegas algorithm, i.e. an algorithm which *always* gives the right answer in some expected polynomial time. Unfortunately, we do not know of any short certificate for the *non-existence* of sandwich homogeneous sets in some given HSSP instance, which surely complicates matters and suggests a little more research on this issue.

References

1. M. R. Cerioli, M. R. Everett, C. M. H. Figueiredo, and S. Klein. The homogeneous set sandwich problem. *Information Processing Letters*, 67:31–35, 1998.
2. C. M. H. Figueiredo and V. G. P. Sá. A new upper bound for the homogeneous set sandwich problem. Technical report, COPPE/Sistemas, Universidade Federal do Rio de Janeiro, 2004.
3. M. C. Golumbic, H. Kaplan, and R. Shamir. Graph sandwich problems. *Journal of Algorithms*, 19:449–473, 1995.
4. R. M. McConnell and J. Spinrad. Modular decomposition and transitive orientations. *Discrete Math.*, 201:189–241, 1999.
5. V. G. P. Sá. The sandwich problem for homogeneous sets in graphs. Master's thesis, COPPE / Universidade Federal do Rio de Janeiro, May 2003.
6. S. Tang, F. Yeh, and Y. Wang. An efficient algorithm for solving the homogeneous set sandwich problem. *Information Processing Letters*, 77:17–22, 2001.