A fast Monte Carlo algorithm for the Homogeneous Set Sandwich Problem

Celina M. H. de Figueiredo* Guilherme D. da Fonseca†
Vinícius G. P. de Sá†

October 7, 2003

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 $G_s(V, E_S)$, $E_1 \subseteq E_S \subseteq E_2$, which contains a homogeneous set, in case such a graph exists. This is called the Homogeneous Set Sandwich Problem (HSSP). We give a simple $O(n^3)$ Monte Carlo algorithm which is asymptotically more efficient than all deterministic HSSP algorithms known so far.

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* $G_s(V, E_S)$, $E_1 \subseteq E_S \subseteq E_2$, which has a desired property Π [3]. In this paper, the property Π we are interested in is the exhibition of a homogeneous set. A *homogeneous set* H, in a graph $G(V, E)$, is a subset of V such that *(i)* $1 < |H| < |V|$ 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 vertices in H 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 McConnel and Spinrad [4] and has $O(|E|)$ time complexity.

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

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 $n = |V|$). 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* and currently sets the HSSP's upper bounds at its time complexity $O(m_1\overline{m_2})$, 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. All these algorithms are nevertheless deterministic.

In this paper, though, we present a *probabilistic* Monte Carlo algorithm which solves this problem in $O(n^3)$ time. This time complexity is clearly better than $O(n^4)$, which is the complexity of Sá *et al.*'s algorithm expressed only as a function of n. When our algorithm answers *yes*, it presents a valid sandwich homogeneous set. When our algorithm answers *no*, it is possible that there is no sandwich homogeneous set for that input, but it is also possible that the algorithm simply did not find any. To bound the probability of giving an incorrect answer, we prove that our algorithm finds a sandwich homogeneous set, in case there exists one, with probability greater than p, for any given constant p, $0 < p < 1$. This kind of algorithm is called a yes-biased Monte Carlo algorithm.

2 The Witness Test

As our algorithm is strongly based on the EBE algorithm [1], we describe it briefly. We define the *bias set* $B(H_k)$ of a vertex subset H_k as the set of vertices $v \notin H_k$ such that $(v, v_i) \in E_1$ and $(v, v_j) \notin E_2$, for some $v_i, v_j \in H_k$. These such vertices v are called *bias vertices* for the set H_k [6]. It is easy to see that H_k, $1 < |H_k| < n$, is a sandwich homogeneous set if and only if $B(H_k) = \varnothing$. It is proved in [1] that any homogeneous set sandwich containing the set of vertices H_k should also contain $B(H_k)$.

Let us suppose we are given a vertex set $H_1 = \{v_1, v_2\}$ and want to know whether there is a sandwich homogeneous set which contains H_1. The EBE algorithm successively computes $H_{k+1} = H_k \cup B(H_k)$ until either $B(H_k) = \varnothing$, whereby H_k is a sandwich homogeneous set and it answers *yes*, or $|H_k| + |B(H_k)| = n$, when it states that there is no sandwich homogeneous set containing $\{v_1, v_2\}$. This procedure, in which bias vertices are successively added to a sandwich homogeneous set candidate, is called *bias envelopment*.

If there is a sandwich homogeneous set H which contains the pair of ver-

tices $\{v_1, v_2\}$, we call $\{v_1, v_2\}$ a *witness*. We refer to the algorithm described in the previous paragraph as *Witness Test*. Using some appropriate data structures, as described in [1], the Witness Test runs in $O(n^2)$ time. The aim of those structures is to partition the set of vertices V into five auxiliary sets H, B, A, N, D, whose consistency is dynamically maintained. The set H is the set we referred to as H_k, i.e. the current sandwich homogeneous set candidate. The set B is simply our already defined $B(H)$. The set A is the set of vertices $v \notin H \cup B$ such that $(v, v') \in E_1$ for some $v' \in H$ and also there is no $v'' \in H$ such that $(v, v'') \notin E_2$. Analogously, the set N is the set of vertices $v \notin H \cup B$ such that $(v, v') \notin E_2$ for some $v' \in H$ and also there is no $v'' \in H$ such that $(v, v'') \in E_1$. Finally, the set $D = V \setminus (H \cup B \cup A \cup N)$. At each step of the witness test, $H \leftarrow H \cup B$ and all other sets are updated accordingly in $O(n|B|)$ time.

The EBE algorithm tries to find a witness exhaustively. It runs the Witness Test 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 $O(n^4)$.

Our algorithm is based on a variation of the Witness Test, which we call *Incomplete Witness Test*. The input of the Incomplete Witness Test is a pair of vertices $\{v_1, v_2\}$ and a parameter $h' < n$. The only change in the incomplete version of the witness test is that, when $|H_k| + |B(H_k)| > h'$, the test stops prematurely and answers *no*. Notice that a *no* answer from the Incomplete Witness Test with parameter h' means that $\{v_1, v_2\}$ is not contained in any homogeneous set of size at most h'. Using the same data structures as in [1], the Incomplete Witness Test runs in $O(nh')$ time.

3 The Monte Carlo Algorithm

In order to gather some intuition, let us suppose the input has a sandwich homogeneous set H with h vertices or more. If we choose a random pair of distinct vertices $\{v_1, v_2\}$, the probability that $\{v_1, v_2\}$ is a witness is at least $q = h(h-1)/n(n-1)$. If somehow we know there is a "rather big" sandwich homogeneous set, we can expect it to be quite easy to find a witness, for many pairs of vertices are witnesses in this case. By applying the Witness Test on a pair of vertices that happens to be a witness, the proof of completeness of the bias envelopment procedure [1] assures we will find a sandwich homogeneous set that contains it. If we run the Witness Test on t independent random pairs of vertices, the probability that we find a homogeneous set is at least $1-(1-q)^t$. This approach leads to an efficient algorithm when there is a "rather big" homogeneous set sandwich.

On the other hand, suppose the input has a "rather small" homogeneous set sandwich H, with h vertices, where h is now small compared to the size of the input. We can run the Incomplete Witness Test with parameter $h' = h$ on every pair of vertices. The total time to find a sandwich homogeneous

set, in case one exists, will be $O(n^3h)$, so that this deterministic algorithm is efficient when h is "small".

Now we can describe an efficient algorithm which is based neither on assumptions about the size of any sandwich homogeneous sets nor even on their existence. This algorithm is a Monte Carlo algorithm which gives the correct answer with probability p.

Our algorithm's idea is to run several Incomplete Witness Tests on random witness candidates (pairs of vertices), in such a way that any desired probability of finding a sandwich homogeneous set, in case there exists one, can be achieved. At each iteration we run the Incomplete Witness Test with parameter h' on a random pair of vertices and either the algorithm succeeds in finding a sandwich homogeneous set or else it aborts the current test whenever the number of vertices in our sandwich homogeneous set candidate exceeds (due to bias envelopment) the parameter h'. The parameter h' is initially set to $h'_1 = n - 1$, as the first iteration corresponds to a complete Witness Test, and *is progressively decreased* along the iterations until it becomes less than 2 (the minimum size allowed for a homogeneous set). This approach clearly saves time compared to running *complete* Witness Tests on all possible witnesses, as in the EBE algorithm.

This continuous decrease in the parameter h' has to be well controlled, though, in order to maintain, after each iteration t, the probability of having found a sandwich homogeneous set (in case there exists one with $h'_{t+1} \leqslant h'_t$ vertices or more) not less than the predefined p. We accomplish this by carefully calculating each h'_{t+1}, bearing in mind that all t previous candidates $\{v_1, v_2\}$, which have *not* led to the discovery of any sandwich homogeneous sets (by the time they were the starting set), also count for the number of witness candidates that are known not to be contained in any sandwich homogeneous sets with up to h'_t vertices. In other words, by the end of the t-th iteration the algorithm *has already tested* t random pairs, none of them contained in any sandwich homogeneous sets with up to h'_t vertices. This number (t) of already tested random pairs is crucial for us to establish the correct relationship between p, the probability of having found a sandwich homogeneous set in case there exists any with at least $h'_{t+1} \leq h'_t$ vertices, and the next parameter h'_{t+1} itself. By fixing p, we calculate h'_{t+1}.

The probability that a random pair is *not* contained in any sandwich homogeneous sets with at least h vertices is at most $\overline{q} = 1 - h(h-1)/n(n-1)$. If we choose t independent random pairs of vertices, the probability that none of these pairs is contained in any sandwich homogeneous set of size h is at most $\overline{q}^t$. Thus, we calculate the value of h'_{t+1} after each iteration t as follows:

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

With some simple manipulations, we have:

$$h^2 - h - (n^2 - n)(1 - (1 - p)^{1/t}) = 0. \tag{2}$$

The value of h'_{t+1} after each iteration t must be an integer value corresponding to the maximum size of a sandwich homogeneous set that needs to be searched for at iteration $t + 1$. Consequently we can set $h'_{t+1} = \lfloor h \rfloor$. Solving the second-degree equation 2, we can set h'_{t+1} to

$$h'_{t+1} = \left\lfloor \frac{1 + \sqrt{1 + 4(n^2 - n)(1 - (1 - p)^{1/t})}}{2} \right\rfloor. \tag{3}$$

In order to illustrate the algorithm by means of a step-by-step appreciation, let us take a pair (G_1, G_2) as our HSSP input instance. The algorithm starts by choosing a random pair of vertices $\{v_1, v_2\}$ for the first iteration. Then, it executes the Incomplete Witness Test with parameter $h'_1 = n - 1$, which is equivalent to the (complete) Witness Test. Suppose it stopped without finding any sandwich homogeneous set. What we do want now is that, after we have set the value for the next parameter h'_2, we are able to say that the previous iteration sufficed for the algorithm to have found a sandwich homogeneous set, in case there is one with h'_2 vertices or more, with probability at least p. So far we know that *one* random pair of vertices is not contained in any sandwich homogeneous sets (with up to $h'_1 = n - 1$ vertices). The following relation allows us to determine the minimum integer h'_2 that achieves it, had some predefined p been given:

$$p \geqslant \frac{h'_2(h'_2 - 1)}{n(n - 1)}.$$

If there is a sandwich homogeneous set with at least h_2 vertices, the first iteration is enough to find one with probability at least p, and here lies the algorithm's main point: further iterations will *not* have to look for sandwich homogeneous sets with more than h_2 vertices, for it *has already been granted* that sandwich homogeneous sets with as many vertices (h_2 or more) would already have been found by the algorithm with probability not less than p. Indeed, this idea applies to every iteration t, when no sandwich homogeneous set with more than h'_t needs to be searched for. In other words, this is what makes the *complete* Witness Tests redundant for our purposes, allowing the *incomplete* version to be used instead.

The second iteration begins with a new, randomly chosen witness candidate. This iteration, as we have seen, does not need to expand the initial set, by means of bias envelopment, until it holds all n vertices, for it has been granted that a sandwich homogeneous set with h'_2 or more vertices would have already been found with probability at least p, so that this (second) iteration is permitted to abort whenever the candidate set has more than

HomogeneousSetSandwich (V, E_1, E_2, p)

1. $h' \leftarrow |V| - 1$
2. $t \leftarrow 0$
3. while $h' \geq 2$
3.1. $(v_1, v_2) \leftarrow$ random pair of distinct vertices of V
3.2. if IncompleteWitnessTest $(V, E_1, E_2, v_1, v_2, h') =$ *yes*
3.2.1. return *yes*
3.3. $t \leftarrow t + 1$
3.4. $h' \leftarrow \lfloor (1 + \sqrt{1 + 4(|V|^2 - |V|)(1 - (1-p)^{1/t})})/2 \rfloor$
4. return *no*

Figure 1: Monte Carlo algorithm for the homogeneous set sandwich.

h'_2 vertices. If no sandwich homogeneous set is found during this iteration, we obtain h'_3 from the relation below:

$$p \geq 1 - (1 - \frac{h'_3(h'_3 - 1)}{n(n-1)})^2.$$

On further iterations, we just have to calculate each h' the same way, as equation 3 allows us to.

If no homogeneous set sandwich has yet been found whenever h'_t becomes less than 2, the algorithm stops and returns *no*. The pseudo-code for this algorithm is in figure 1.

4 Complexity Analysis

The first iteration of the algorithm runs the complete Witness Test 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 Witness Tests 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 $O(nh')$ time each. To analyze the time complexity of the algorithm, we have to calculate

$$\sum_{t=1}^{t'-1} O(nh'_{t+1}) = \sum_{t=1}^{t'-1} O(nh),$$

where t' is the maximum number of iterations.

The parameter $h'_{t+1} = \lfloor h \rfloor$ of the incomplete witness test in iteration $t + 1$ is defined by equation 1. To calculate $t = t'$, we replace h 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,

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

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

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

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

Since $h \geq 2$,

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

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

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

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

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

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

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

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 a simple $O(n^3)$ yes-biased Monte Carlo algorithm for the Homogeneous Set Sandwich Problem. Considering that Tang *et al.*'s $O(n^3)$ algorithm was proved incorrect, the best deterministic algorithms for this problem are $O(n^4)$, if we express time complexity only as a function of n.

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.

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] C. M. H. Figueiredo and V. G. P. Sá, *A new upper bound for the homogeneous set sandwich problem.*

[3] M. C. Golumbic, H. Kaplan, and R. Shamir, *Graph sandwich problems*, Journal o fAlgorithms **19** (1995), 449–473.

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

[5] V. G. P. Sá, *O problema-sanduíche para conjuntos homogêneos em grafos*, 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** (2001), 17–22.