Planar Graph Coloring is Not Self-Reducible, Assuming $P \neq NP$

Samir Khuller *
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742

Vijay V. Vazirani †
Computer Science Department
Indian Institute of Technology
New Delhi 110016, India

Abstract

We show that obtaining the lexicographically first four coloring of a planar graph is NP–hard. This shows that planar graph four-coloring is not self-reducible, assuming $P \neq NP$. One consequence of our result is that the schema of [JVV 86] cannot be used for approximately counting the number of four colorings of a planar graph. These results extend to planar graph k-coloring, for $k \geq 4$.

*Research done while this author was supported by an IBM Graduate Fellowship at Cornell University.
†Work done while at Cornell University. Supported by NSF grant DCR 85-52938 and PYI matching funds from AT&T Bell Labs and Sun Microsystems, Inc.

1. Introduction

Most known problems in NP are *self-reducible.* It is because of this property that the search version of an NP problem is Turing-reducible to its decision version. By suitably carrying out this reduction, the lexicographically first solution to the search problem can also be obtained. This property also plays a crucial role in reducing the problem of approximately counting the number of solutions of an NP problem to generating a random solution to a given instance of this problem [JVV 86].

In this paper we show that planar graph four-colorability is not self-reducible unless $P = NP$. To our knowledge, this is the first such result. The usual manner of carrying out self-reducibility of the general graph k-colorability problem does not work for our problem since the reduction destroys planarity (see Section 2). Is there some other way of achieving self-reducibility ? We provide a negative answer as follows: the decision version of planar graph four-colorability is in P [AH 77]. On the other hand we show that obtaining the lexicographically first such solution is NP-hard, thereby proving that the problem is not self-reducible, if $P \neq NP$. The NP-hardness of obtaining the lexicographically first solution contrasts with the fact that there is a polynomial time algorithm for obtaining an arbitrary solution [AH 77].

One consequence of our result is that the schema of [JVV 86] cannot be used for approximately counting the number of four-colorings of a planar graph. We extend these results to planar graph k-colorability for any fixed $k \geq 4$.

Our proofs are quite straightforward; the main interest lies in the peculiar situation that obtaining the lexicographically first solution is NP-hard even though the decision version is in P, which enables us to get the result. Clearly, this proof method will not work for showing the lack of self-reducibility in problems whose decision version is NP-hard. It will be interesting to find other such natural problems (possibly on restricted families) that exhibit this situation, as well as discover other situations that yield proofs of lack of self-reducibility.

2. Self-Reducibility

In this section we introduce the formal definition of self-reducibility given by [Sc 76]. A different notion of self-reducibility was given by [Se 88]; however the first notion appears to be more useful for relating the complexities of search and decision problems, and random generation and approximate counting. Let Σ be a fixed finite alphabet in which we are going to encode both our problem instance and the solution. Let $R \subseteq \Sigma^* \times \Sigma^*$ be a binary relation over Σ. For each string (problem instance) $x \in \Sigma^*$, we denote by $R(x)$ the corresponding solution set.

$$R(x) = \{y \in \Sigma^* : (x, y) \in R\}$$

By an example we illustrate what R is: Let x encode a boolean formula B, and y encode a satisfying assignment. Then we define

$$R = \{(x, y) : x, y \in \Sigma^* \text{ and } y \text{ is a satisfying assignment to instance } x\}$$

We call R *self-reducible* if

- There exists a polynomial time computable length function $\ell_R : \Sigma^* \to N$ such that $\ell_R(x) = O(|\, x\,|^{k_R})$ for some constant $k_R > 0$, and

$$y \in R(x) \Rightarrow |\, y\,| = \ell_R(x) \;\forall x, y \in \Sigma^*.$$

- For all $x \in \Sigma^*$ with $\ell_R(x) = 0$, the predicate $\Lambda \in R(x)$ can be tested in polynomial time. (Λ denotes the empty string over Σ.)

- There exist polynomial time computable functions $\psi : \Sigma^* \times \Sigma^* \to \Sigma^*$ and $\sigma : \Sigma^* \to N$ satisfying

$$\sigma(x) = O(\log |\, x\,|)$$

$$\forall x \in \Sigma^* \; [\ell_R(x) > 0 \Leftrightarrow \sigma(x) > 0 \,]$$

$$\forall x, w \in \Sigma^* \; [|\, \psi(x, w)\,| \leq \; |\, x\,| \;]$$

$$\forall x, w \in \Sigma^* \; [\ell_R(\psi(x, w)) = \max\{\ell_R(x) - |\, w\,|, 0\} \,]$$

  and such that each solution set can be expressed in the form

$$R(x) = \bigcup_{w \in \Sigma^{\sigma(x)}} \{wy : y \in R(\psi(x, w))\}.$$

The first condition simply states that the length of the solution is bounded by some polynomial function of the problem instance. The third condition provides an inductive construction of the solution sets as follows: if the solution length is greater than 0, then $R(x)$ is partitioned into classes according to the initial segment w of length $\sigma(x)$, and each class can then be expressed as the solution set of another instance $\psi(x, w)$ of the same problem, concatenated with w. For the set of all possible solutions $R(x)$, to a problem x, we can define the usual lexicographic ordering between the strings.

The following proposition is well known:

Proposition 2.1: *For a problem P that is self-reducible, given a polynomial time decision oracle for the problem we can construct the lexicographically first solution in polynomial time.*

In this formal setting, we now illustrate that general graph k-coloring is self-reducible.

Problem : Given a graph G and a clique K of size k, is $G \cup K$ k-colorable ?

Clique K has been introduced in the problem instance since it yields an easy self-reducibility. We use R to encode all possible solutions to the problem. Note that $y \in R(x)$ if y denotes a valid color assignment to the vertices (represented by a set of pairs (v_i, c_i) where v_i has color c_i). Clearly $G \cup K$ is k-colorable if and only if G is k-colorable. Assume that u_i gets color i (vertices u_i belong to K).

Suppose we wish to color vertex v_i with color $j \in \{1, \ldots, k\}$. Here $w = (v_i, j)$. We now produce the graph G_i as follows: delete v_i from G, and add edges from u_j (recall that u_j has color j) to $N(v_i)$ (neighbours of v_i). It is easy to see that a coloring for G_i can be used to obtain a coloring for G by coloring v_i with the color j (same as u_j). Moreover the size of the graph G_i (measured in the number of vertices) is smaller than the size of G.

3. Lexicographic Colorings

Every legal k-coloring of a graph may be represented as a string $C = c_1c_2c_3...c_n$ where c_i is the color of vertex v_i. Assume that $c_i \in \{1, 2, .., k\}$. Note that all strings are of length n (where n is the number of vertices in the graph) and the LF-k coloring is the 'smallest' (in the usual lexical ordering on strings) legal coloring which uses at most k colors.

We show that computing the LF-k coloring for a planar graph is NP-hard for any fixed k ($k \geq 4$) (even though the graph is k colorable) by a reduction from planar graph three colorability which is known to be NP-hard [GJ 78].

Before illustrating the proof for arbitrary k we show a simple proof for the case $k = 4$.

Theorem 3.1: *Obtaining the LF-4 coloring for a planar graph is NP-hard.*

Proof: We prove the problem to be NP-hard by exhibiting a simple reduction from the graph three colorability problem. Given $G(V, E)$ (a planar graph) construct the following graph $G'(V', E')$. Assuming G has n vertices, the graph G' has $2n$ vertices.
Define $V' = V \cup \{u_i \mid v_i \in V\}$.
Define $E' = E \cup \{(v_i, u_i) \mid v_i \in V\}$.
The vertices of G' are numbered as follows: Label each u_i as i and each v_i as $n + i$ (in G each v_i was numbered i). Note that G' is planar, since each vertex u_i can be embedded in a face adjacent to v_i.

Now we obtain a LF-4 coloring for G'. If G was three colorable then the LF-4 coloring of G' has the property that all the u_i vertices are colored with color 1. This coloring is valid since no u_i is adjacent to a u_j and since

G is three colorable the rest of the graph can be assigned a legal coloring (without using the color 1). If the LF-4 coloring has the property that all the u_i vertices are colored with color 1, then it is easy to see that all the vertices v_i use only the colors from the set $\{2,3,4\}$ and hence G must be three colorable. Thus from the LF-4 coloring it is easy to check whether the original graph G is three colorable or not. □

The proof for $k=5,6$ is very similar to the proof shown above, where instead of attaching a single vertex to each node of the graph we attach a K_2 and K_3 respectively to each node of the graph by adding edges from each node of the complete graph. The new vertices have to be numbered carefully so that each K_2 (K_3) is colored with the colors 1 and 2 (1, 2 and 3) thus making these two (three) colors 'forbidden' colors for the vertices in G. Note that the graph G' formed in each case will be planar. We cannot attach a K_4 (for $k=7$) since that would make G' non-planar. We develop a general 'gadget' which is planar, and which can be attached to each vertex of the original graph, achieving the effect of introducing 'forbidden' colors at each vertex.

Theorem 3.2: *Obtaining the LF-k coloring (for any fixed $k \geq 3$) for a planar graph is NP-hard.*

Proof: Obtaining a LF-3 coloring is obviously NP-hard so we concentrate our attention on the case $k>3$. The idea is similar to the one used in the previous theorem. We prove the problem NP-hard by exhibiting a reduction from the graph three colorability problem. Given $G(V,E)$ (a planar graph) we construct a graph $G'(V',E')$ as follows: We first show the construction of the subgraph $H_{k'}(V_H,E_H)$ (where $k'=k-3$) which is used in the construction of the graph G'.

The subgraphs $H_{k'}$ are defined recursively as follows:

If $k' \leq 2$ then $H_{k'} = K_{k'}$ (complete graph on k' vertices).

If $k'>2$ then the subgraphs $H_{k'}$ are defined recursively as follows: Each $H_{k'}$ consists of a *spine*, which is a set of k' vertices $\{u^1_{k'}, u^2_{k'}, ..., u^{k'}_{k'}\}$ with $u^\ell_{k'}$ adjacent to $u^{\ell+1}_{k'}$ $(1 \leq \ell < k')$. On each vertex $u^\ell_{k'}(2<\ell\leq k')$ of the spine we 'attach' the subgraph $H_{\ell-2}$ by adding edges from $u^\ell_{k'}$ to each vertex $u^j_{\ell-2}$ on the spine of $H_{\ell-2}$ $(1 \leq j \leq \ell-2)$. More formally, we introduce the edges $\{(u^\ell_{k'}, u^j_{\ell-2}) \mid 2<\ell\leq k', 1\leq j\leq \ell-2\}$. We refer to $u^m_{k'}$ as the m^{th} vertex on the spine of $H_{k'}$. The *sub-spines* of $H_{k'}$ are the spine and sub-spines of H_ℓ $(1\leq \ell\leq k'-2)$ if H_ℓ is attached to $u^{\ell+2}_{k'}$. Similarly, we can refer to a vertex as the m^{th} vertex on a sub-spine of $H_{k'}$ if it is the

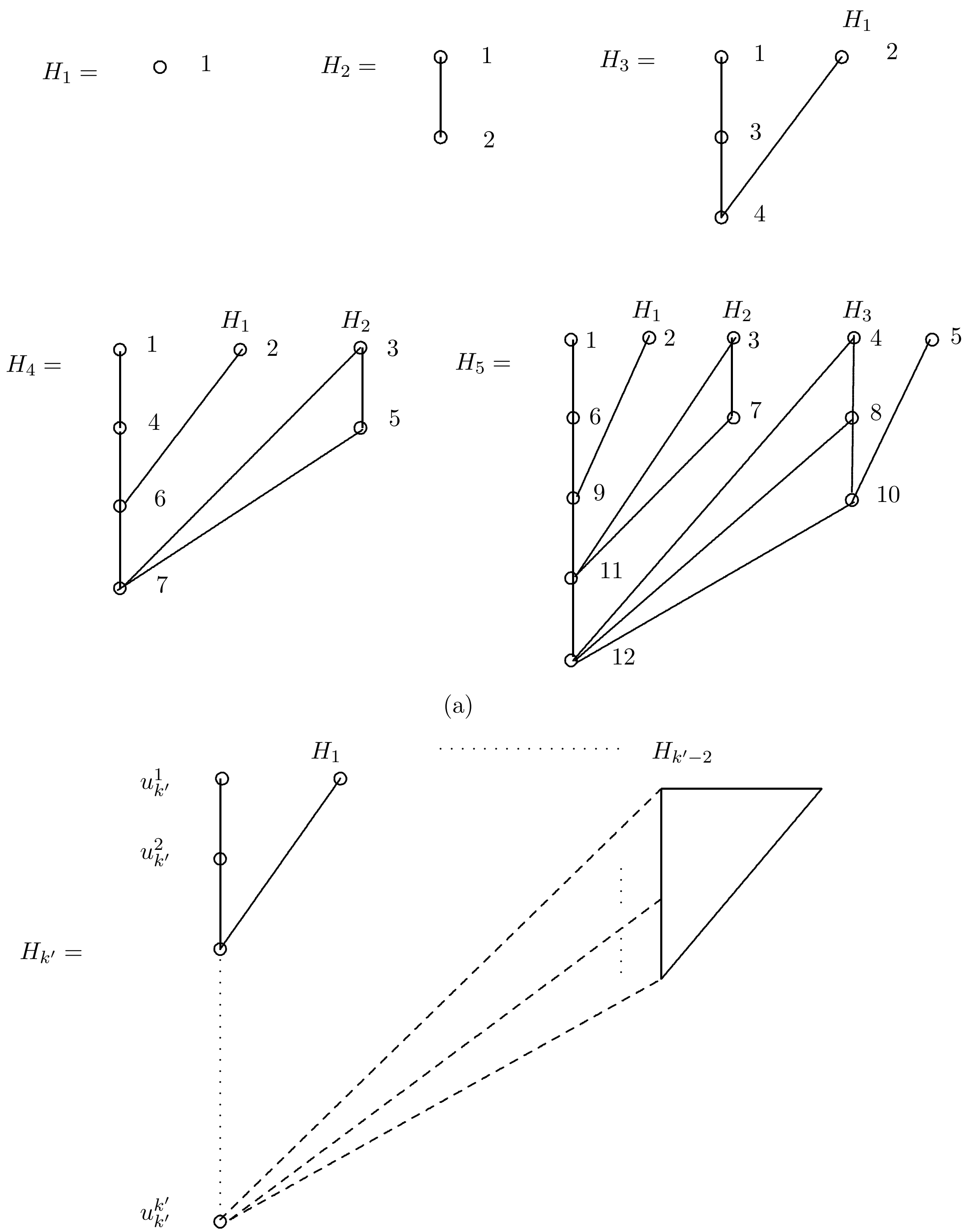

Figure 1: Graphs H_k used in Theorem 3.2

m^{th} vertex on the spine of some H_ℓ that was used in forming $H_{k'}$. The spine of $H_{k'}$ is also a sub-spine of $H_{k'}$.

The graphs H_i ($i \leq 5$) are shown in Fig 1(a). The general structure of $H_{k'}$ is shown in Fig 1(b). It is easy to prove by induction on k' that $H_{k'}$ is planar.

The graph G' is constructed by 'attaching' copies of $H_{k'}$ (call them $H^i_{k'}(V^i_H,\ E^i_H)$) to each vertex v_i of G. We add edges from v_i to each vertex $u^j_{k'}$ ($1 \leq j \leq k'$) on the spine of $H^i_{k'}$. More formally, we have $V' = V \cup V^i_H(1 \leq i \leq n)$. Also we have $E' = E \cup E^i_H \cup \{(v_i, u^j_{k'}) \mid u^j_{k'} \in spine\ of H^i_{k'}(1 \leq j \leq k')\}(1 \leq i \leq n)$.

Each vertex $v_i \in V$ is given the number $i + nf_{k'}$ where $f_{k'}$ is the number of vertices in $H_{k'}$. The graphs $H_{k'}$ may be embedded with all the sub-spines aligned vertically as shown in Fig 1(b). Vertices belonging to $H_{k'}$ are assigned numbers from the set $\{1, 2, , .., f_{k'}\}$, with each vertex being assigned a distinct number. We assign the m^{th} vertex of a sub-spine belonging to $H_{k'}$ a smaller number than the p^{th} vertex of a sub-spine belonging to $H_{k'}$ if $m < p$, regardless of them belonging to the same sub-spine or different sub-spines. One scheme to obtain the numbering, is to number the rows left to right starting from the topmost row (see Fig 1(a)). Each vertex in V^i_H is given the number $(i-1)f_{k'} + j$ where j is the number of the vertex in the numbering of $H_{k'}$.

Now we obtain a *LF-k* coloring for G'. Assume G is three colorable. The *LF-k* coloring of G' has the property that the m^{th} vertices in sub-spine's are colored with color m. The coloring is valid since each $H_{k'}$ subgraph can be legally colored with k' colors. This coloring can now be extended to a complete k-coloring for the graph since all the original vertices of G can be colored using only three colors. In fact, it is easy to see that the *LF-k* coloring will have exactly this property and will color the original vertices in a lexicographically first manner using only three colors.

If the *LF-k* coloring has the property that all the vertices in $H^i_{k'}$ (for $1 \leq i \leq n$) in row j are colored with color j then the graph G is three colorable since every vertex of the graph has vertices of the k' colors {1,2,3,...,k'} adjacent to it (these colors are 'forbidden' colors for the original vertices of the graph and since the graph is k colorable all the original nodes use only 3 colors).

The size of the graphs we generate in our reductions are easily seen to be exponential in k. The reduction however is still a polynomial time reduction since k is a constant. □

From the proof of the previous theorem and the proposition of Section 2, we have:

Corollary 3.3: *Planar Graph k-coloring is not self reducible, assuming* $NP \neq P$.

4. Open Problems

As mentioned in the introduction, it will be interesting to identify other problems that do not possess self-reducibility. Another interesting problem is to determine the complexity of (a) exactly, (b) approximately computing the number of four colorings of a planar graph. The former appears to be $\#P$-complete. The latter also appears to be intractable – perhaps in the sense that if it were doable in random polynomial time, then $NP = RP$. Finally, notice that our proof technique breaks down for three-colorability of planar graphs – is this problem self-reducible?

Acknowledgments: We would like to thank Howard Karloff and Mihalis Yannakakis for useful discussions.

References

[AH 77] K.Appel and W.Haken, "Every planar map is four-colorable", *Illinois Journal of Maths*, 21 (1977), pp. 429–567.

[GJ 78] M.R.Garey and D.S.Johnson, "Computers and Intractibility: A guide to the theory of NP-completeness", *Freeman, San Francisco.*

[JVV 86] M.R.Jerrum, L.G.Valiant and V.V.Vazirani, "Random generation of combinatorial structures from a uniform distribution", *Theoretical Computer Science*, 43 (1986), pp. 169–188.

[Sc 76] C.P.Schnorr, "Optimal algorithms for self-reducible problems", *Proc. of the 3rd ICALP*, (1976), pp. 322–337.

[Se 88] A. Selman, "Natural self–reducible sets", *SIAM Journal on Computing*, vol 17, no 5, (1988), pp. 989–996.