A LINEAR ALGORITHM FOR FIVE-COLORING A PLANAR GRAPH

N. Chiba, T. Nishizeki and N. Saito
Department of Electrical Communications
Faculty of Engineering, Tohoku University
Sendai, Japan 980

Abstract. A simple linear algorithm is presented for coloring planar graphs with at most five colors. The algorithm employs a recursive reduction of a graph involving the deletion of a vertex of degree 6 or less possibly together with the identification of its several neighbors.

1. Introduction

A coloring of a graph is an assignment of colors to the vertices in such a way that adjacent vertices have distinct colors. Although the problem of coloring a graph with the minimal number of colors has practical applications in some schedulings [1], it is known to be NP-complete even for the class of planar graphs [3].

We present here a linear algorithm for finding a coloring of a planar graph with at most five colors, that is, 5-coloring. We denote by n the number of vertices of a graph throughout this paper. Based on the well-known Kempe-chain argument, one can easily design an $O(n^2)$ time algorithm for the purpose by employing a simple recursive reduction of a graph involving the deletion of a vertex of degree 5 or less possibly together with the interchange of colors in a two-colored subgraph. Lipton and Miller have given an O(nlogn) algorithm for the problem by removing a "batch" of vertices rather than just a single vertex [4]. Their algorithm and its proof are a little complicated. In this paper we give a simple linear algorithm for the purpose. The algorithm does not use the Kempe-chain argument, but uses a recursive reduction of a graph involving the deletion of a vertex of degree 6 or less possibly together with the identification of several neighbors of the vertex. We prove that the algorithm runs in O(n) time. Hence the computational complexity of our algorithm is optimal within a constant factor.

2. Outline of the algorithm

We first define some terms. Let G=(V,E) be a graph with vertex set V and edge set E. We consider only a simple graph G, that is, a graph with no multiple edges or loops. A graph G is planar if it is embedable in the plane without edge crossing. The neighborhood N(v) of a vertex v is the set of all vertices which are adjacent to v. The degree d(v) of a vertex v of G is the number of vertices adjacent to v. The deletion of a vertex v is an operation on G which delete v together with all the edges incident to v, and the resulting graph is denoted by G-v. Let u and v be two vertices of a graph G. A vertex-identification (or simply identification) ⟨u,v⟩ is an operation on G which identifies u and v, that is, removes u and v and adds a new vertex adjacent to those vertices to which u or v was adjacent. Our algorithm frequently uses these operations in recursive reductions of graphs.

The outline of the algorithm is as follows. Suppose that G is a given planar graph. We construct a new planar (simple) graph G' from G by deleting a vertex v of degree 6 or less possibly together with some other modifications, and then color G' with 5 colors by recursively applying the algorithm. We extend the 5-coloring of G' to a 5-coloring of G by assigning to v a color not used to vertices in N(v). In order to guarantee that there remains such a color, we construct G' so that G' contains only four vertices in N(v), as follows. If v is of degree 4 or less, then we simply set G'=G-v. If v is of degree 5, then we construct G' from G-v by identifying a pair of nonadjacent vertices in N(v). Note that there exist such a pair of vertices since G is planar (see Lemma 1), and that the resulting planar graph G' has no loops. The pair of vertices of G will be assigned the same color as the vertex substituting for them in G'. Finally, if v is of degree 6, then we construct a planar graph G' from G-v by identifying either three pairwise nonadjacent vertices in N(v) or two pairs of nonadjacent vertices in N(v). Lemma 2 in Section 3 guarantees that there exist such vertices. Note that we must select two pairs of vertices appropriately so that G' is planar.

We use adjacency lists to represent a graph G. All the operations in the algorithm, other than vertex-deletions or vertex-identifications, require O(n) time in total. Clearly the deletion of a single vertex v requires O(d(v)) time. Therefore all the vertex deletions used in the algorithm require at most O(n) time in total, since $\Sigma_{v \in V} d(v) \leq 6n$. Hence we should implement the algorithm so that all the vertex-identifications require O(n) time in total. One can easily execute the single identification of vertices u and v in O(d(u)+d(v)) time, that is, one can modify the adjacency lists of G in that amount of time so that the resulting lists represent a new graph obtained from G by identifying u and v. However the same vertex may appear in identifications O(n) times, so a direct implementation of the algorithm would require $O(n^2)$ time. As we describe the detailes in the following section, the algorithm runs in several stages, in each of which we do the recursive reductions as far as any vertex would not be involved in more than two of identifications, so that the stage requires at most O(n) time. An argument in Section 4 will show that the resulting graph G' at the end of a stage has a positive fraction of vertices at the beginning of the stage. From these facts it will be shown that the algorithm requires O(n) time in total.

Remark We have given a simple "on-line" algorithm to execute any sequence of vertex-identifications of a graph G=(V,E) in O(|E|log|V|) time, by using adjacency lists together with an adjacency matrix [2]. It yields an alternative simple O(nlogn) 5-coloring algorithm of planar graphs.

3. 5-coloring algorithm

In this section we present the linear algorithm for coloring planar graphs with at most five colors. We first have the following lemmas.

LEMMA 1. Let a planar graph G=(V,E) contain a vertex v of degree 5 with $N(v)=\{v_1,v_2,v_3,v_4,v_5\}$. Then, for any specified $v_i \in N(v)$, there exists a pair of nonadjacent vertices v_j and v_k, $j,k \neq i$. Furthermore one can find such a pair in $O(\mathrm{MIN}_{v \in N(v)-v_i} d(v))$ time if the planar embedding of G is given.

Proof. We can assume without loss of generality that $v_i=v_1$, and that the vertices v_1,v_2,v_3,v_4 and v_5 in N(v) are labeled clockwise about v in the plane embedding of G. Consider the case in which $d(v_2)$ is minimum among $d(v_2)$, $d(v_3)$, $d(v_4)$ and $d(v_5)$. Scanning all the elements in the adjacency list for v_2, one can know whether $(v_2,v_4)\in E$ or not. If $(v_2,v_4)\in E$, then $(v_3,v_5)\notin E$. Thus one can find a pair of nonadjacent vertices in $O(d(v_2))$ time. The proof for all the remaining cases is similar to above. Q.E.D.

Lemma 1 implies that for a vertex v of degree 5 one can always find a pair of nonadjacent vertices v_j and v_k in N(v) both of which have not been involved in vertex-identifications even if N(v) contains a vertex v_i involved in a vertex-identification so far.

LEMMA 2. Let a planar graph G=(V,E) contain a vertex v of degree 6 with $N(v)=\{v_1,v_2,\ldots,v_6\}$. Then N(v) contains either

(i) three pairwise nonadjacent vertices,

or (ii) two pairs of nonadjacent vertices v_i,v_j and v_k,v_ℓ such that the identification $\langle v_i,v_j\rangle$ together with $\langle v_k,v_\ell\rangle$ does not destroy the planarity of G-v.

Furthermore one can find these vertices in $O(\mathrm{MIN}_{1\leq s<t\leq 5}[d(v_s)+d(v_t)])$ time if the planar embedding of G is given.

Proof. Assume that the vertices $v_1,v_2,\ldots,v_6$ in N(v) are labeled clockwise about v in the plane embedding of G. The identifications of two "cross-over" pairs of vertices in N(v), such as v_2,v_5 and v_3,v_6, may destroy the planarity of G-v, since v_3 and v_6 possibly donot lie on the boundary of a common face when v_2 is identified with v_5 in G-v. However the identifications of two "parallel" pairs, such as v_2,v_6 and v_3,v_5, necessarily preserve the planarity of G-v. We establish our claim only for the case in which $d(v_1)+d(v_2)$ is minimum among all the sums of degrees of two vertices in N(v), since the proof for all the remaining cases is similar. Scanning all the elements of the adjacency lists for v_1 and for v_2, one can know whether the edges (v_1,v_5) and (v_2,v_4) exist or not. If exactly one of them, say (v_1,v_5), exists, then v_2,v_4 and v_6 are the required three pairwise nonadjacent vertices. Otherwise, v_2,v_6 and v_3,v_5 (if both (v_1,v_5) and (v_2,v_4) exist) or v_2,v_4 and v_1,v_5 (if neither exists) are the required two "parallel" pairs of nonadjacent vertices in N(v). Thus one can find the required vertices in $O(d(v_1)+d(v_2))$ time. Q.E.D.

As a data structure to represent a graph G, we use an adjacency list L(v) for each $v \in V$. Each adjacency list is doubly linked. The two copies of each edge (u,v), one in L(u) and the other in L(v), are also doubly linked. In addition to L, we use four arrays FLAG, COUNT, DEG and DP together with three queues Q(i), $4 \leq i \leq 6$. An element DEG(v) of array DEG contains the value of d(v), $v \in V$. FLAG(v) has an initial value "false" at the beginning of each stage of the algorithm, and will be set to "true" when v is identified with another vertex. COUNT(v) contains the number of vertices $w \in N(v)$ with FLAG(w)=true, that is, the number of vertices in N(v) involved in vertex-identifications in the current stage so far. The queue Q(i), $4 \leq i \leq 6$, contains all the vertices which are available for the recursive reduction of the stage, defined as follows:

$Q(4)=\{v \mid DEG(v) \leq 4\}$;

$Q(5)=\{v \mid DEG(v)=5, COUNT(v) \leq 1\}$; and

$Q(6)=\{v \mid DEG(v)=6, COUNT(v)=0\}$.

That is, Q(4) is the set of all the vertices of degree 4 or less, Q(5) the set of all the vertices of degree 5 with at most one neighbor involved in an identification in the stage, and Q(6) the set of all the vertices of degree 6 with no neighbors involved in any identification in the stage. DP(v) has a pointer to an element "v" in Q(i) if v is contained in Q(i). We are now ready to present the algorithm.

```
procedure FIVE;
    comment The procedures DELETE and IDENTIFY are for the
    vertex-deletion and the vertex-identification, respectively;
    procedure COLOR(G);
      begin
        if |V|≤5 then assign |V| colors to |V| vertices
        else
         begin
          if Q(4)≠∅
             then begin
                    take a top entry v from Q(4);
                    DELETE(v);
                    let G' be the reduced graph
                  end
             else
               if Q(5)≠∅
                  then begin
```

```
                    take a top entry v from Q(5);
                    choose two nonadjacent vertices
                    x,y∈N(v) such that
                    FLAG(x)=FLAG(y)=false;
                    DELETE(v);
                    IDENTIFY(x,y);
                    let G' be the reduced graph
                  end
              else
               if Q(6)≠∅
                 then
                  begin
                   take a top entry v from Q(6);
                   comment By Lemma 2 either
                   case (i) or case (ii) holds;
                   for case (i) do
                     begin
                       let x,y and z be the three pairwise
                       nonadjacent vertices in N(v);
                       DELETE(v);
                       IDENTIFY(y,x);
                       IDENTIFY(z,x)
                     end;
                   for case (ii) do
                     begin
                       let v_i,v_j and v_k,v_l be the two
                       "parallel" pairs of nonadjacent vertices
                       in N(v);
                       DELETE(v);
                       IDENTIFY(v_i,v_j);
                       IDENTIFY(v_k,v_l)
                     end;
                   let G' be the reduced graph
                  end
                else
                  begin
                    comment Current stage is over. Reset FLAG
                            and COUNT;
                    for v∈V do begin FLAG(v):=false;
                                     COUNT(v):=0 end;
                    COLOR(G)
                  end;
         COLOR(G');
         assign to v a color not used in the coloring of N(v),
         and to each identified vertex of G the color of the
         vertex substituting for it in G';
         comment Note that the number of colors used in the
         coloring of N(v) is at most 4
    end
end

begin
   embed a given planar graph G in the plane;
   for v∈V do
     begin
       calculate DEG(v);
       FLAG(v):=false;
       COUNT(v):=0
     end;
   COLOR(G)
end
```

```
procedure DELETE(v);
  begin
    for w∈L(v) do
         begin
           delete w from L(v);
           delete v from L(w);
           DEG(w):=DEG(w)-1;
           if FLAG(v)=true
              then COUNT(w):=COUNT(w)-1;
         end;
    delete L(v) from the adjacency lists and "v" from Q(i), i=4,5 or
    6, if any, and update appropriately the elements in Q(i)
    according to the modifications of DEG and COUNT above
  end


procedure IDENTIFY(u,v);
    comment This procedure executes the identification <u,v> of two
    nonadjacent vertices u and v such that either FLAG(u) or FLAG(v)
    is "false". We assume FLAG(u)=false without loss of generality.
    The vertex v will act as a new vertex substituting for u and old
    v;
    begin
      if FLAG(v)=false
              then begin
                     FLAG(v):=true;
                     for w∈L(v) do COUNT(w):=COUNT(w)+1
                   end;
      for w∈L(v) do mark w with "v";
      for w∈L(u) do
              begin
                delete w from L(u); delete u from L(w);
                if w has no mark "v"
                   then begin
                          comment w is adjacent to u, but not to
                          v;
                          add w to L(v); add v to L(w);
                          DEG(v):=DEG(v)+1;
                          COUNT(w):=COUNT(w)+1;
                          if FLAG(w)=true
                              then COUNT(v):=COUNT(v)+1
                        end
                   else begin
                          comment w is adjacent to both u and v;
                          DEG(w):=DEG(w)-1
                        end
              end;
         delete L(u) from the adjacency lists and "u" from Q(i),
         i=4,5 or 6, if any, and update appropriately the elements in
         Q(i), i=4,5,6, according to the above modifications of DEG
         and COUNT
    end
```

In the algorithm above we omit the detail of the method for obtaining the planar embedding of G' from that of G, since clearly the time required for the purpose is propotional to that for the vertex-deletions and identifications.

4. Time complexity

In this section, we establish the following theorem.

THEOREM. The procedure FIVE colors a planar graph $G=(V,E)$ with at most five colors in $O(n)$ time, where $n=|V|$.

We first present the following lemma before establishing the Theorem. The lemma implies that at the end of each stage of the algorithm a positive fraction, say 1/12, of the remaining vertices have been involved in vertex-identifications.

LEMMA 3. Let $G=(V,E)$ be a planar graph with minimum degree 5, and let S be a subset of V. If every vertex of degree 5 is adjacent to at least two vertices in S, and every vertex of degree 6 is adjacent to at least one vertex in S, then $|S| \geq n/12$.

Proof. Define $V_5=\{v|d(v)=5, v\in V\}$, $V_6=\{v|d(v)=6, v\in V\}$, and $V_*=\{v|d(v)\geq 7, v\in V\}$ so that $V=V_5\cup V_6\cup V_*$, and let $p_5=|V_5|$, $p_6=|V_6|$, and $p_*=|V_*|$. Define $S_5=S\cap V_5$, $S_6=S\cap V_6$ and $S_*=S\cap V_*$ so that $S=S_5\cup S_6\cup S_*$, and let $r_5=|S_5|$, $r_6=|S_6|$, and $r_*=|S_*|$.

By Euler's formula $|E| \leq 3n$, we have

$$5p_5+6p_6+\Sigma_{v\in V_*}d(v) \leq 6(p_5+p_6+p_*).$$

Hence we have

$$p_5 \geq \Sigma_{v\in V_*}(d(v)-6) \geq p_*. \quad (1)$$

Since $n=p_5+p_6+p_*$, we have from (1)

$$p_5+p_6 \geq n/2. \quad (2)$$

We furthermore have from (1)

$$p_5 \geq \Sigma_{v\in S_*}d(v)-6r_*. \quad (3)$$

Since every vertex of degree 5 is adjacent to at least two vertices in S, and every vertex of degree 6 is adjacent to at least one vertex in S, we have

$$\Sigma_{v\in S}d(v) \geq 2p_5+p_6. \quad (4)$$

On the other hand we have

$$\Sigma_{v\in S}d(v) \leq 6(r_5+r_6)+\Sigma_{v\in S_*}d(v). \quad (5)$$

Combining (4) and (5), we have

$$2p_5+p_6 \le 6(r_5+r_6)+\Sigma_{v \in S_*} d(v). \qquad (6)$$

By (3) and (6),

$$2p_5+p_6 \le 6(r_5+r_6)+p_5+6r_*=6|S|+p_5,$$

and hence

$$|S| \ge (p_5+p_6)/6.$$

Therefore we have $|S| \ge n/12$ by (2), as desired. Q.E.D.

We are now ready to prove the Theorem.

Proof of the Theorem. Noting that the reduced graph G' of a planar graph G is a planar simple graph smaller than G, we can easily prove by induction on the number of vertices of a graph that the algorithm correctly colors a planar graph G with at most 5 colors. Hence we shall show that the algorithm runs in O(n) time.

We first show that the first stage of the algorithm requires at most O(n) time. One can easily verify that the procedure DELETE executes the deletion of a vertex v in O(d(v)) time, and that the procedure IDENTIFY does the identification of two nonadjacent vertices u and w in O(d(u)+d(w)) time since it simply scans the elements of L(u) and L(w). The algorithm calls DELETE for a vertex in each reduction. Since every vertex appears in at most one vertex-deletion, all the vertex-deletions in the stage require O(n) time in total. Consider a reduction around a vertex v of degree 5 or 6, in which IDENTIFY is called in addition to DELETE. If v is in Q(5), the algorithm finds two neighbors v_i and v_j of v with $FLAG(v_i)=FLAG(v_j)=false$, and then calls $IDENTIFY(v_i,v_j)$. The identification requires $O(d(v_i)+d(v_j))$ time. Lemma 1 implies that one can find v_i and v_j in that amount of time. If v is in Q(6), the algorithm finds either three pairwise nonadjacent vertices x,y and z or two pairs of nonadjacent vertices v_i,v_j and v_k,v_ℓ, and then calls IDENTIFY(y,x) and IDENTIFY(z,x) or $IDENTIFY(v_i,v_j)$ and $IDENTIFY(v_k,v_\ell)$, respectively. These two identifications together require O(d(x)+d(y)+d(z)) or $O(d(v_i)+d(v_j)+d(v_k)+d(v_\ell))$ time, respectively. Lemma 2 implies that one can find these vertices in that amount of time. Of course, FLAG's for these vertices are all "false", since COUNT(v)=0. That is, all these vertices have not been involved in any vertex-identification in the stage. Thus every vertex is involved in at most two identifications in the stage. (The vertex x above is possibly involved in two identifications.) Therefore all the identifications in the stage require O(n) time in total. Clearly the book-keeping operations required for the four arrays and three queues need O(n) time in total. Note that one can directly access "v" via a pointer in DP(v). Hence we can conclude that the stage requires O(n) time.

We next show that at the end of the first stage the reduced graph $G'=(V',E')$ contains at most 8n/9 vertices. Suppose that $|V'|=n'\neq 0$. Then the minimum degree of G' is 5, and $COUNT(v) \geq 2$ for every vertex v of degree 5, and $COUNT(v) \geq 1$ for every vertex of degree 6, since Q(4),Q(5) and Q(6) are all empty at the end of the stage. Let $S=\{v|FLAG(v)=true, v\in V'\}$ so that the subset S of V' satisfies the requirement of Lemma 3, then we have $|S| \geq n'/12$. Clearly at least |S| vertices disappear from the graph G by vertex-identifications. Since each reduction produces at most two vertices in S, there must occur at least |S|/2 graph reductions around vertices of degree 5 or 6 in the stage. Therefore at least |S|/2 vertices are deleted from G by vertex-deletions in the stage. Hence at least 3|S|/2 vertices disappear from G in the stage. Therefore we have

$$n-n' \geq 3|S|/2$$

Since $|S| \geq n'/12$, we have

$$n' \leq 8n/9.$$

Using the two facts above, we have the following equations on T(n) the number of steps (or time) needed to 5-color a planar graph G of n vertices:

$$\begin{cases} T(n) \leq c_1 & \text{if } n \leq 5; \\ T(n) \leq T(8n/9)+c_2 n & \text{otherwise,} \end{cases}$$

where c_1 and c_2 are constants. Solving these equations, we have $T(n)=O(n)$. Q.E.D.

Acknowledgement. We wish to thank Dr. T. Asano for his valuable suggestions and discussions on the subjects. This work was partly supported by the Grant in Aid for Scientific Research of the Ministry of Education, Science and Culture of Japan under Grant: Cooperative Research (A) 435013 (1980).

REFERENCES

[1] B. Carre, Graphs and Network, Clarendon Press, Oxford, 1979.

[2] N. Chiba, T. Nishizeki and N. Saito, An Approximation Algorithm for the Maximum Independent Set Problem on Planar Graphs, submitted to SIAM J. on Comput., 1980.

[3] M. R. Garey, D. S. Jhonson and L. Stockmeyer, Some simplified NP-complete graph problems, Theor. Comput. Sci. 1, 237-267, 1976.

[4] R. J. Lipton and R. E. Miller, A batching method for coloring planar graphs, Information Processing Letters, 7,4, 185-188, 1978.