数理解析研究所講究録 1375 巻 2004 年 35-41

A Linear-Time Algorithm for 7-colorirng 1-planar Graphs

東京電機大学理工学部数理科学科 陳 致中 (Zhi-Zhong Chen)
河野 光治 (Mitsuharu Kouno)
Department of Mathematical Sciences,
Tokyo Denki University

Abstract

A graph G is 1-planar if it can be embedded in the plane in such a way that each edge crosses at most one other edge. Borodin showed that 1-planar graphs are 6-colorable, but his proof does not lead to an efficient algorithm. This paper presents a linear-time algorithm for 7-coloring 1-planar graphs. The main difficulty in the design of our algorithm comes from the fact that the class of 1-planar graphs is not closed under the operation of edge contraction. This difficulty is overcome by a structure lemma that may find useful in other problems on 1-planar graphs.

1 Introduction

The problem of coloring the vertices of a graph using few colors has been a central problem in graph theory. It has also been intensively studied in algorithm theory due to its applications in many practical fields such as scheduling, resource allocation, and VLSI design. Of special interest is the case where the graph is planar. Appel and Haken [1, 2] showed that every planar graph is 4-colorable, but their proof does not lead to an efficient algorithm. Since then, a number of linear-time algorithms for 5-coloring planar graphs have appeared [8, 11, 10, 15, 9].

An interesting generalization of planar graphs is the class of 1-planar graphs. The problem of coloring the vertices of a 1-planar graph using few colors has also attracted very much attention [13, 12, 3, 4, 5]. Indeed, the problem has been formulated in another different way: It is equivalent to the problem of coloring the vertices of a plane graph so that the boundary vertices of every face of size at most 4 receive different colors [12]. Ringel [13] proved that every 1-planar graph is 7-colorable and conjectured that every 1-planar graph is 6-colorable. Ringel [13] and Archdeacon [3] confirmed the conjecture for two special cases. Borodin [4] settled the conjecture in the affirmative with a lengthy proof. He [5] later came up with a relatively shorter proof. However, his proof does not lead to an efficient algorithm for 6-coloring 1-planar graphs.

Chen, Grigni, and Papadimitriou [6] studied a modified notion of planarity, in which two nations of a (political) map are considered adjacent when they share any *point* of their boundaries (not necessarily an *edge*, as planarity requires). Such adjacencies define a *map graph* (see [7] for a comprehensive survey of known results on map graphs). The map graph is called a *k-map graph* if no more than k nations on the map meet at a point. As observed in [6], the adjacency graph of the United States is nonplanar but is a 4-map graph . Obviously, every 4-map graph is 1-planar. In Section 4, we will observe that every 1-planar graph can be modified to a 4-map graph by adding some edges (see Corollary 4.3 below). By these facts, the problem of coloring 1-planar graphs is essentially equivalent to the problem of coloring 4-map graphs.

Recall that in the case of planar graphs, an efficient 4-coloring algorithm seems to be difficult to design and hence it is of interest to look for an efficient 5-coloring algorithm. Similarly, in the case of 1-planar graphs, an efficient 6-coloring algorithm seems to be difficult to design and hence it is of interest to look for an efficient 7-coloring algorithm. In this paper, we present the first linear-time algorithm for 7-coloring 1-planar graphs. Our algorithm is much more complicated than all 5-coloring algorithms for planar graphs. The main reason is that unlike planar graphs, the class of 1-planar graphs is not closed under the operation of *edge contraction* (recall that *contracting an edge* $\{u, v\}$ in a graph G is made by replacing u and v by a single new vertex z and adding an edge between z and each original neighbor of u and/or v). It is worth noting that many coloring algorithms (e.g., those for planar graphs) are crucially based on the property that the class of their input graphs is closed under the operation of edge contraction. In the case of 1-planar graphs, this property is not available and it becomes difficult to find suitable vertices to merge so that the resulting graph is still 1-planar. We overcome this difficulty with a structure lemma which essentially says that every 1-planar graph either has a constant fraction of vertices of degree at most 7, or has a constant fraction of vertices each of which is of degree 8 and has at least 5 neighbors of degree at most 8. We believe that this lemma will find useful in the design of algorithms for other problems on 1-planar graphs.

2 Preliminaries

Throughout this paper, a graph is always simple (i.e., has neither multiple edges nor self-loops) unless stated explicitly otherwise.

Let $G = (V, E)$ be a graph. The *neighborhood* of a vertex v in G, denoted $N_G(v)$, is the set of vertices in G adjacent to v; $d_G(v) = |N_G(v)|$ is the *degree* of v in G. For $U \subseteq V$, let $N_G(U) = \cup_{u \in U} N_G(u)$. For $U \subseteq V$, the *subgraph of G induced by U* is the graph (U, F) with $F = \{\{u, v\} \in E : u, v \in U\}$ and is denoted by $G[U]$. For $U \subseteq V$, we denote by $G - U$ the subgraph induced by $V - U$. If $u \in V$, we write $G - u$ instead of $G - \{u\}$. A *cut set* of G is a subset U of V such that $G - U$ is disconnected. A *k-cut set* is a cut set consisting of k vertices. G is *k-connected* if it has at least k vertices but has no i-cut set with $i \leq k - 1$. An *independent set* in G is a set of pairwise nonadjacent vertices in G. A *maximal independent set* in G is an independent set in G that is not a proper subset of another independent set in G.

A *1-plane embedding* of G is an embedding of G in the plane in such a way that each edge crosses at most one other edge. G has a 1-plane embedding only when G is a 1-planar graph. An *edge-crossing list* of G is a list L of disjoint (unordered) pairs of edges of G such that G has a 1-plane embedding in which the two edges in each pair in L cross while no two other edges of G cross.

For a sequence $\langle u_1, \ldots, u_k \rangle$ of two or more distinct pairwise nonadjacent vertices in G, *merging* $\langle u_1, \ldots, u_k \rangle$ is the operation of modifying G by adding an edge between u_k and every vertex in $\cup_{1 \leq i \leq k-1} N_G(u_i) - N_G(u_k)$ and further removing vertices $u_1, \ldots, u_{k-1}$. Note that the sequence is ordered and u_k is the last vertex in the sequence.

Let k be a natural number. A *k-coloring* of G is a coloring of the vertices of G with at most k colors such that no two adjacent vertices get the same color. The *color classes* of a coloring $\mathcal{C}$ of the vertices of G are the sets V_1, V_2, ..., V_k, where k is the number of colors used by $\mathcal{C}$ and V_i, $1 \leq i \leq k$, is the set of all vertices with the ith color.

3 Simple Reductions

In this section, we show how to reduce the problem to its 3-connected case. Throughout this section, G denotes the input 1-planar graph.

Since it is still unknown whether 1-planar graphs can be recognized in polynomial time, we assume that G is given by its adjacency list together with an edge-crossing list L of G. We may further assume that for each pair $(e, e') \in L$, e and e' share no endpoint (otherwise, the pair (e, e') can be removed from L, and L remains an edge-crossing list of G after the removal).

In the initialization step of the algorithm, we augment G as follows: For each pair $(e, e') \in L$, each endpoint u of e, and each endpoint v of e', if (u, v) is not an edge of G, then add a new edge between u and v in G. Obviously, this augmentation can be done in linear time. Moreover, after the augmentation, it is clear that (1) G remains 1-planar and (2) for each pair $(e, e') \in L$, the endpoints of e and e' induce a clique of size 4 in G.

We may assume that G is connected, since otherwise the problem of 7-coloring G is easily reduced to the problems of 7-coloring the connected components of G. A *block* of G is a maximal 2-connected subgraph of G. The following fact is widely known.

Fact 3.1 *Suppose that G is connected but not 2-connected. Let B be the set of all blocks of G. Let C be the class of all 1-cut sets of G. Consider the bipartite graph $T = (B \cup C, E_T)$ where E_T consists of all edges $\{b, c\}$ such that $b \in B$, $c \in C$, and the vertex in c is a vertex in B. Then, T is a tree.*

Corollary 3.2 *Suppose that G is 1-planar and connected but not 2-connected. Then, given the blocks of G and a 7-coloring of each block of G, we can compute a 7-coloring of G in linear time.*

Next, suppose that G is 2-connected but not 3-connected. Let U be a 2-cut set of G, and V_1, ..., V_p be the vertex sets of the connected components of $G - U$. For $1 \leq i \leq p$, let G_i be the graph $G[V_i \cup U]$ if the two vertices in U are adjacent in G, while let G_i be the graph obtained from $G[V_i \cup U]$ by adding an edge between the two vertices in U otherwise. The graphs G_1, G_2, ..., G_p are called the *augmented components* of G induced by U. Obviously, all the augmented components of G induced by U are also 2-connected.

Lemma 3.3 *Suppose that G is 1-planar. Then, each augmented component of G induced by a 2-cut set U is also 1-planar. Moreover, for each augmented component G_i of G, we can obtain an edge-crossing list of G_i from L by removing all pairs (e, e') such that e or e' is not an edge of G_i.*

Replacing G by the augmented components induced by a 2-cut set is called *splitting* G. Suppose G is split, the augmented components are split, and so on, until no more splits are possible. The graphs constructed in this way are 3-connected and the set of the graphs are called a *2-decomposition* of G. for an example.) Each element of a 2-decomposition of G is called a *split component* of G. It is possible for G to have two or more 2-decompositions. A split component of G must be either a triangle or a 3-connected graph with at least four vertices.

The following fact is widely known [14].

Fact 3.4 *Suppose that G is 2-connected but not 3-connected. Let D be a 2-decomposition of G. Let C be the family of all 2-cut sets of G used to split G into the split components in D. Consider the bipartite graph $T = (D \cup C, E_T)$ where E_T consists of all edges $\{d, c\}$ such that $d \in D$, $c \in C$, and the two vertices in c are vertices in d. Then, T is a tree.*

Corollary 3.5 *Suppose that G is 1-planar and connected but not 2-connected. Then, given a 2-decomposition D and a 7-coloring of each split component in D, we can compute a 7-coloring of G in linear time.*

4 The Algorithm for the 3-Connected Case

Throughout this section, G denotes the input 3-connected 1-planar graph and L denotes the input edge-crossing list of G. By our discussion in Section 3, we may assume that for each pair $(e, e') \in L$, (1) e and e' share no endpoint, (2) the endpoints of e and e' induce a clique C of size 4 in G, and (3) no edge of C other than e and e' is contained in a pair in L.

Given G and L, we first construct a graph H as follows. H contains all vertices of G and all those edges of G that are contained in no pair in L. Moreover, for each (unordered) pair $\{e, e'\} \in L$, H contains a new vertex $v_{e,e'}$, and contains an edge between $v_{e,e'}$ and each endpoint of e and/or e'. H does not contain other vertices or edges. Since L is an edge-crossing list of G, H is a planar graph. We then compute a plane embedding of H in linear time. For convenience, we identify H with its plane embedding. Hereafter, for a vertex v of H, we say that two neighbors u and w of v in H are *consecutive* if u and w appear around $v_{e,e'}$ consecutively (clockwise or counterclockwise) in H.

Fact 4.1 *Let $v_{e,e'}$ be a vertex in H but not in G. Suppose that x and y are two consecutive neighbors of $v_{e,e'}$ in H (note that $\{x, y\}$ is an edge in H by our assumptions on L). Then, the cycle C formed by the three edges $\{v_{e,e'}, x\}$, $\{x, y\}$, and $\{y, v_{e,e'}\}$ together is the boundary of some face of H.*

Lemma 4.2 *We can modify H in linear time so that H satisfies the following conditions:*

1. *H has the same vertices as G, and contains all edges of G.*
2. *For each vertex v of H and for every two consecutive neighbors u and w of v in H, $\{u,w\}$ is an edge in H and crosses no edge in H.*
3. *For each pair of edges of H that cross in H, the endpoints of the two edges induce a clique of size 4 in H.*

Corollary 4.3 *Every 1-planar graph has a supergraph that is a 4-map graph.*

Assumption 4.4 *H satisfies the conditions in Lemma 4.2.*

Corollary 4.5 *The following two statements hold:*

1. *Let v be a vertex in H. Suppose that u is a neighbor of v in H such that edge $\{v,u\}$ crosses another edge $\{x,y\}$ in H. Then, $\{x,y\} \subseteq N_H(v)$, and x,u,y appear around v consecutively in H in this order (clockwise or counterclockwise).*
2. *Let u and w be two consecutive neighbors of v in H. Then, at least one of edges $\{v,u\}$ and $\{v,w\}$ crosses no edge in H.*

4.1 A Structure Lemma

Fix two constants α and K with $1 < \alpha < 2$ and $K > 7 + 9/(\alpha - 1)$. Let v be a vertex of H. If $|d_H(v)| \leq K$, we say that v is *small*; otherwise, we say that v is *large*. We say that v is *reducible* if one of the following holds:

1. $d_H(v) \leq 6$.
2. $d_H(v) = 7$ and $N_H(v)$ contains at most one large vertex.
3. $d_H(v) = 8$, $N_H(v)$ contains no large vertex, and one of the following holds:
   (a) There are at most two vertices $u \in N_H(v)$ with $d_H(u) \geq 9$.
   (b) There are exactly three vertices $u \in N_H(v)$ with $d_H(u) \geq 9$ and there are distinct vertices u_1, u_2, u_3 in $N_H(v)$ such that $d_H(u_1) \geq 9$, $d_H(u_2) \geq 9$, $d_H(u_3) \leq 8$, and $\{v, u_2\}$ and $\{u_1, u_3\}$ are edges of H and they cross in H.

Lemma 4.6 *Let R be the set of reducible vertices in H. Then, R contains a constant fraction of vertices of H.*

Corollary 4.7 *We can compute a set I of reducible vertices of H in linear time such that the following conditions are satisfied:*

1. *I contains a constant fraction of vertices of H.*
2. *For every two vertices u and v in I, there is no path P between u and v in H such that P has at most three edges and has no large vertex.*

4.2 Outline of the Algorithm

We first give an outline of the algorithm. It first computes a set I of reducible vertices of H satisfying the conditions in Corollary 4.7. It then uses I and H to construct a new 1-planar graph G' in linear time such that the number of vertices in G' is a constant fraction of the number of vertices in H and a 7-coloring of H can be constructed in linear time from an arbitrarily given 7-coloring of G'. It further recurses on G' to obtain a 7-coloring of G' which is then used to obtain a 7-coloring of H in linear time. Since each recursion takes linear time and reduces the size of the graph by a constant fraction, the overall time is linear. The core of the algorithm is in the construction of G'.

4.3 Constructing Graph G' for Recursion

To construct G', we may simply remove all $v \in I$ with $d_H(v) \leq 6$ from H because each 7-coloring of $H - v$ extends to a 7-coloring of H. Similarly, for each $v \in I$ such that $N_H(v)$ contains a vertex u with $d_H(u) \leq 6$, we may remove u from H. However, these are not enough because I may contain very few such vertices v. So, we need to do something about those vertices $v \in I$ such that $7 \leq d_H(v) \leq 8$ and $N_H(v)$ contains no vertex u with $d_H(u) \leq 6$. We call such vertices v *critical* vertices. The idea is to explore the neighborhood structure of critical vertices. First, we need the following definitions:

Definition 4.1 *A vertex x in H is* dangerous *for a critical vertex v if one of the following holds:*

- *$d_H(v) = 7$ and x is a large neighbor of v in H.*
- *$d_H(v) = 8$, $x \notin N_H(v) \cup \{v\}$, and x is adjacent to some vertex $u \in N_H(v)$ in H.*

Definition 4.2 *Let v be a critical vertex with $d_H(v) = 7$. A* mergable pair *for v is a pair (u, w) of two nonadjacent neighbors of v in H such that u is small and the graph G_1 obtained from $H - v$ by merging $\langle u, w\rangle$ is a 1-planar graph.*

Definition 4.3 *Let v be a critical vertex with $d_H(v) = 8$.*

1. *A* mergable triple *for v is a set $\{u_1, u_2, u_3\}$ of three pairwise nonadjacent neighbors of v in H such that the graph G_2 obtained from $H - v$ by merging $\langle u_1, u_2, u_3\rangle$ is a 1-planar graph.*
2. *Two* simultaneously mergable pairs *for v are two pairs (u_1, u_2) and (w_1, w_2) such that u_1, u_2, w_1, and w_2 are distinct neighbors of v in H, neither $\{u_1, u_2\}$ nor $\{w_1, w_2\}$ is an edge of H, and the graph G_3 obtained from $H - v$ by merging $\langle u_1, u_2\rangle$ and merging $\langle w_1, w_2\rangle$ is a 1-planar graph.*
3. *A* desired quadruple *for v is an ordered list (u, w_1, w_2, w_3) of four distinct neighbors in H such that*
   - *$d_H(u) \leq 8$,*
   - *$\{w_1, w_2\} \subseteq N_H(u)$, and*
   - *$\{w_1, w_2, w_3\}$ is an independent set in H, and the graph G_4 obtained from $H - \{v, u\}$ by merging $\langle w_1, w_2, w_3\rangle$ is a 1-planar graph.*
4. *A* favorite quintuple *for v is an ordered list (u, w_1, w_2, w_3, w_4) of five distinct neighbors of v in H such that*
   - *$d_H(u) \leq 8$,*
   - *$\{w_1, w_2\} \subseteq N_H(u)$, and*
   - *neither $\{w_1, w_2\}$ nor $\{w_3, w_4\}$ is an edge in H, and the graph G_5 obtained from $H - \{v, u\}$ by merging $\langle w_1, w_2\rangle$ and merging $\langle w_3, w_4\rangle$ is a 1-planar graph.*
5. *A* desired quintuple *for v is an ordered list $(u_1, u_2, w_1, w_2, w_3)$ of five distinct neighbors of v in H such that*
   - *$d_H(u_1) \leq 8$ and $d_H(u_2) \leq 8$,*
   - *$\{w_1, w_2\} \subseteq N_H(u_1)$ and $\{w_2, w_3\} \subseteq N_H(u_2)$, and*
   - *$\{w_1, w_2, w_3\}$ is an independent set in H, and the graph G_6 obtained from $H - \{v, u_1, u_2\}$ by merging $\langle w_1, w_2, w_3\rangle$ is a 1-planar graph.*
6. *A* desired sextuple *for v is an ordered list $(u_1, u_2, w_1, w_2, w_3, w_4)$ of six distinct neighbors of v in H such that*
   - *$d_H(u_1) \leq 8$ and $d_H(u_2) \leq 8$,*
   - *$\{w_1, w_2\} \subseteq N_H(u_1)$ and $\{w_3, w_4\} \subseteq N_H(u_2)$, and*
   - *neither $\{w_1, w_2\}$ nor $\{w_3, w_4\}$ is an edge in H, and the graph G_7 obtained from $H - \{v, u_1, u_2\}$ by merging $\langle w_1, w_2\rangle$ and merging $\langle w_3, w_4\rangle$ is a 1-planar graph.*
7. *A* useful sextuple *for v is an ordered list $(u_1, u_2, w_1, x_1, w_2, x_2)$ of six distinct vertices in H such that*
   - *$d_H(u_1) \leq 8$, $d_H(u_2) \leq 8$, and $\{u_1, u_2\} \subseteq \{v\} \cup N_H(v)$,*
   - *$w_1 \in \{v\} \cup N_H(v)$ and $w_2 \in \{v\} \cup N_H(v)$,*

- $\{w_1, x_1, w_2, x_2\} \subseteq N_H(u_1)$ *and* $\{u_1, w_1, x_1\} \subseteq N_H(u_2)$, *and*
- *neither* $\{w_1, x_1\}$ *nor* $\{w_2, x_2\}$ *is an edge in* H, *and the graph* G_8 *obtained from* $H - \{u_1, u_2\}$ *by merging* $\langle w_1, x_1\rangle$ *and merging* $\langle w_2, x_2\rangle$ *is a 1-planar graph.*

8. A useful triple *for* v *is an ordered list* (u, w, x) *of three distinct vertices in* H *such that*

- $d_H(u) \leq 7$ *and* $u \in N_H(v)$,
- $w \in N_H(v)$ *and* $\{w, x\} \subseteq N_H(u)$, *and*
- $\{w, x\}$ *is not an edge in* H *and the graph* G_9 *obtained from* $H - \{u\}$ *by merging* $\langle w, x\rangle$ *is a 1-planar graph.*

Theorem 4.8 *For a critical vertex* v, *call an edge* e *in* H *a* basic critical edge *for* v *if at least one endpoint of* e *is* v *or a small neighbor of* v *in* H, *Moreover, for a critical vertex* v, *call an edge* e *in* H *a* critical edge *for* v *if* e *is a basic critical edge for* v *or* e *crosses a basic critical edge for* v *in* H. *Then, for every critical vertex* v, *the following hold:*

1. If $d_H(v) = 7$, *then we can use the sub-embedding of* H *induced by the set of critical edges for* v *to find a mergable pair for* v *in* $O(1)$ *time such that the graph* G_1 *defined in Definition 4.2 has a 1-plane embedding* H' *satisfying the following three conditions:*

  (C1) *For every pair of edges* e_1 *and* e_2 *in* H', e_1 *and* e_2 *cross each other in embedding* H' *if and only if they cross each other in embedding* H.

  (C2) *For every vertex* x *in* H' *that is neither* v *nor a small neighbor of* v *in* H, *and for every sequence* $\langle e_1, \ldots, e_k\rangle$ *of edges in* H *that are incident to* x *but incident to neither* v *nor a small neighbor of* v, *if edges* $e_1, \ldots, e_k$ *appear around* x *consecutively in this order in embedding* H, *then edges* $e_1, \ldots, e_k$ *appear around* x *consecutively in this order in embedding* H'.

  (C3) *Same as (C2) but with both occurrences of the word "consecutively" deleted.*

2. If $d_H(v) = 8$, *then we can use the sub-embedding of* H *induced by the set of critical edges for* v *to compute one of the following for* v *in* $O(1)$ *time:*

  - *A mergable triple such that the graph* G_2 *defined in Definition 4.3(1) has a 1-plane embedding* H' *satisfying the above conditions (C1) through (C3).*
  - *Two simultaneously mergable pairs such that the graph* G_3 *defined in Definition 4.3(2) has a 1-plane embedding* H' *satisfying the above conditions (C1) through (C3).*
  - *A desired quadruple such that the graph* G_4 *defined in Definition 4.3(3) has a 1-plane embedding* H' *satisfying the above conditions (C1) through (C3).*
  - *A favorite quintuple such that the graph* G_5 *defined in Definition 4.3(4) has a 1-plane embedding* H' *satisfying the above conditions (C1) through (C3).*
  - *A desired quintuple such that the graph* G_6 *defined in Definition 4.3(5) has a 1-plane embedding* H' *satisfying the above conditions (C1) through (C3).*
  - *A desired sextuple such that the graph* G_7 *defined in Definition 4.3(6) has a 1-plane embedding* H' *satisfying the above conditions (C1) through (C3).*
  - *A useful sextuple such that the graph* G_8 *defined in Definition 4.3(7) has a 1-plane embedding* H' *satisfying the above conditions (C1) through (C3).*
  - *A useful triple such that the graph* G_9 *defined in Definition 4.3(8) has a 1-plane embedding* H' *satisfying the above conditions (C1) through (C3).*

Now, we are ready to explain how to construct G'. The construction of G' from H is done as follows.

1. For each critical vertex v with $d_H(v) = 7$, find a mergable pair for v as guaranteed in Theorem 4.8.
2. For each critical vertex v with $d_H(v) = 8$, find a mergable triple, two simultaneously mergable pairs, a desired quadruple, a favorite quintuple, a desired quintuple, a desired sextuple, a useful sextuple, or a useful triple for v as guaranteed in Theorem 4.8.
3. For each critical vertex v with $d_H(v) = 7$ and the mergable pair (u, w) found for v in Step 1, remove v from H and further merge $\langle u, w\rangle$.

4. For each critical vertex v with $d_H(v) = 8$, perform the following:
   (a) If a mergable triple $\{u_1, u_2, u_3\}$ was found for v in Step 2, then remove v from H and further merge $\langle u_1, u_2, u_3 \rangle$.
   (b) If two simultaneously mergable pairs (u_1, u_2) and (w_1, w_2) were found for v in Step 2, then remove v from H, merge $\langle u_1, u_2 \rangle$, and further merge $\langle w_1, w_2 \rangle$.
   (c) If a desired quadruple (u, w_1, w_2, w_3) was found for v in Step 2, then remove v and u from H, and further merge $\langle w_1, w_2, w_3 \rangle$.
   (d) If a favorite quintuple (u, w_1, w_2, w_3, w_4) was found for v in Step 2, then remove v and u from H, merge $\langle w_1, w_2 \rangle$, and further merge $\langle w_3, w_4 \rangle$.
   vspace-.15cm
   (e) If a desired quintuple $(u_1, u_2, w_1, w_2, w_3)$ was found for v in Step 2, then remove v, u_1, and u_2 from H, and further merge $\langle w_1, w_2, w_3 \rangle$.
   (f) If a desired sextuple $(u_1, u_2, w_1, w_2, w_3, w_4)$ was found for v in Step 2, then remove v, u_1, and u_2 from H, merge $\langle w_1, w_2 \rangle$, and further merge $\langle w_3, w_4 \rangle$.
   (g) If a useful sextuple $(u_1, u_2, w_1, x_1, w_2, x_2)$ was found for v in Step 2, then remove u_1 and u_2 from H, merge $\langle w_1, x_1 \rangle$, and further merge $\langle w_2, x_2 \rangle$.
   (h) If a useful triple (u, w, x) was found for v in Step 2, then remove u from H and further merge $\langle w, x \rangle$.
5. Remove all $v \in I$ with $d_H(v) \leq 6$ from H.
6. For each $v \in I$ such that $N_H(v)$ contains a vertex u with $d_H(u) \leq 6$, remove all such vertices u from H.

References

[1] K. Appel and W. Haken. Every planar map is four colorable, Part I: Discharging. *Illinois J. Math.*, 21:429–490, 1977.

[2] K. Appel, W. Haken, and J. Koch. Every planar map is four colorable, Part II: Reducibility. *Illinois J. Math.*, 21:491–567, 1977.

[3] D. Archdeacon. Coupled colorings of planar graphs. *Congres. Numer.*, 39:89–94, 1983.

[4] O.V. Borodin. Solution of Ringel's problems on vertex-face coloring of planar graphs and coloring of 1-planar graphs (in Russian). *Met. Discret. anal., Novosibirsk*, 41:12–26, 1984.

[5] O.V. Borodin. A new proof of the 6 color theorem. *J. Graph Theory*, 19:507–521, 1995.

[6] Z.-Z. Chen, M. Grigni, and C.H. Papadimitriou. Planar map graphs. In *Proc. ACM STOC'98*, pages 514–523, 1998.

[7] Z.-Z. Chen, M. Grigni, and C.H. Papadimitriou. Map graphs. *J. ACM*, 49:127–138, 2002.

[8] N. Chiba, T. Nishizeki, and N. Saito. A linear 5-coloring algorithm of planar graphs. *J. Algorithms.*, 8:470–479, 1981.

[9] M. Chrobak and K. Diks. Two algorithms for coloring planar graphs with 5 colors. Tech. Report, Columbia University, January, 1987.

[10] G.N. Frederickson. On linear-time algorithms for five-coloring planar graphs. *Inform. Process. Lett.*, 19:219–224, 1984.

[11] D.W. Matula, Y. Shiloach, and R.E. Tarjan. Two linear-time algorithms for five-coloring a planar graph. Tech. Report STAN-CS-80-830, Stanford University, November, 1980.

[12] O. Ore and M.D. Plummer. Cyclic coloration of planar graphs. *Recent Progress in Combinatorics (Proc. 3rd Waterloo Conf. on Combinatorics, 1968)*, Academic Press, New York-London, pages 287–293, 1969.

[13] G. Ringel. Ein Sechsfarbenproblem auf der Kugel. *Abh. Math. Sem. Univ. Hamburg*, 29:107-117, 1965.

[14] W.T. Tutte. *Connectivity in Graphs.* University of Toronto Press, Toronto, 1966.

[15] M.H. Williams. A linear algorithm for colouring planar graphs with five colours. *Comput. J.*, 28:78–81, 1985.