A Heuristic for 4-Colouring a Planar Graph

Hamish Carr and William Kocay*
Computer Science Department
University of British Columbia
Vancouver, BC, Canada, R3T 2N2
and
Computer Science Department
St. Paul's College, University of Manitoba
Winnipeg, Manitoba, Canada, R3T 2N2
e-mail: hcarr@cs.ubc.ca, bkocay@cs.umanitoba.ca

Abstract

A technique is described that constructs a 4-colouring of a planar triangulation in quadratic time. The method is based on iterating Kempe's technique. The heuristic gives rise to an interesting family of graphs which cause the algorithm to cycle. The structure of these graphs is described. A modified algorithm that appears always to work is presented. These techniques may lead to a proof of the 4-Colour Theorem which does not require a computer to construct and colour irreducible configurations.

1. Kempe's Algorithm

One of the most well-known, and well-studied, problems in graph theory is the "Four-Colour Problem" – the proposition that any planar graph has a proper colouring using no more than 4 colours. The first published proof was that of Kempe [9], although this was later shown to be in error by Heawood [7]. Since then, numerous researchers have studied the problem, including Allaire and Swart [1], Appel and Haken [2,3,4], Birkhoff [5], Heesch [8], Ore [10], Robertson *et al* [11,12,13], Saaty & Kainen [14,15], Tutte [16], and Whitney [16].

In 1976, a proof was finally achieved by Appel and Haken [2,3,4], but this proof relied on a computer to resolve nearly 1800 cases (later reduced to 1400), and was considered by many to be unsatisfactory. This proof has since been improved by Robertson *et al* [11,12,13], using 633 cases. Although still relying on a computer to check all the cases, this newer proof gives assurance that no fatal errors were made in the original proof.

---

* This work was supported by an operating grant from the Natural Sciences and Engineering Research Council of Canada.

The common factor in Kempe's approach and those of Appel and Haken and Robertson *et al*, is the *configuration* [1,8]. Each of these approaches relies on showing that any planar graph contains one of a number of configurations, and that for each configuration, a proper colouring of a smaller (reduced) graph can be extended to a proper colouring of the initial graph.

The Appel and Haken proof is claimed to result in a quartic-time algorithm, while the Robertson *et al* proof results in a quadratic-time algorithm. However, in each case, it is necessary to find one of the numerous configurations in the graph.

We attempted to find a simple extension of Kempe's algorithm that would correctly 4-colour all planar graphs in quadratic time or better. The algorithm which we have developed appears to run in quadratic time for all planar graphs, but we are unable to prove that it always works – hence the designation "heuristic". A proof that the algorithm always works would be a proof of the 4-Colour Theorem.

Since our algorithm is based on that of Kempe [9], we review Kempe's algorithm first, then show how to modify it to produce a new algorithm. We then describe a class of graphs for which the new algorithm cycles, and make some observations about this class of graphs. The result is a heuristic which appears always to work.

Let G be a connected undirected simple graph on n vertices $V(G)$. If $u, v \in V(G)$, then $u \to v$ means that u is adjacent to v (and so also $v \to u$). The reader is referred to [6] for other graph-theoretic terminology. Without loss of generality, we will assume that G is a connected planar triangulation. We will use induction on n. As our base case, we note that graphs with $n \leq 4$ can easily be 4-coloured.

In any triangulation with at least 5 vertices, there is always a vertex of degree 3, 4, or 5. We follow Kempe's method in dealing with degrees 3 and 4, and modify his method for degree 5. When a vertex z in G of degree k is deleted, a face bounded by a polygon of degree k is created. We will always assume that $G - z$ is drawn so that this polygon is the *outer* face.

For any graph G, define the *Kempe subgraph* for colours i and j, K^{ij}, to be the subgraph of G induced by the vertices of colour i or j. Each connected component of this subgraph will be called a *Kempe component*. We will identify a Kempe component by one or more of its vertices: $K^{ij}(u)$ will refer to the Kempe component of colours i and j that contains vertex u. A uv-path in $K^{ij}(u)$ will be called a *Kempe chain*. We will sometimes write a uv-Kempe chain of colours i and j, or alternatively, an ij-Kempe chain from u to v.

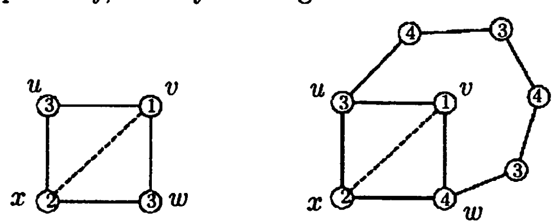
1.1 Degree 3:

Let z be a vertex of degree 3 in G. Let the adjacent vertices be u, v and w. Notice that $G - z$ is a triangulation on $n - 1$ vertices. We 4-colour $G - z$ and re-attach z, giving it a colour different from u, v and w.

1.2 Degree 4:

Let z be a vertex of degree 4 in G. Let the adjacent vertices be u, v, w and x. Deleting z gives a quadrilateral face $uvwx$ (see Fig. 1). To maintain the triangulation, we insert either uw or vx. Note that we can always do so, since uw and vx cannot both be in G, as G is a planar graph.

By hypothesis, the reduced graph is 4-colourable. If any two of u, v, w, x are the same colour, then there is a colour available for z, as in graph H of Fig. 1. We therefore assume that u, v, w, x are of colours $3, 1, 4, 2$ respectively, as in graph K of Fig. 1. Consider the Kempe component $K^{34}(u)$. If $w \in K^{34}(u)$, we can interchange the colours 3 and 4 in $K^{34}(u)$ to get a new colouring of G' in which u, v, w, x are coloured 4,1,4,2, respectively, thereby allowing us to colour z with colour 3. Otherwise $w \in K^{34}(u)$. Let C be a uw-Kempe chain in $K^{34}(u)$. Now construct the Kempe component $K^{12}(v)$. Clearly $x \in K^{12}(v)$, since any vx-Kempe chain must intersect C, which is coloured with colours 1 and 3 only. Therefore we can interchange colours 1 and 2 in $K^{12}(v)$ to get a new colouring of G' in which u, v, w, x are coloured 3,2,4,2, respectively, thereby allowing us to colour z with colour 3.



H *K*

Fig. 1, Graphs H and K

This argument, that two complementary Kempe components cannot intersect, is at the heart of the algorithm, and we will see it repeatedly in what follows. The main operation in the algorithm will be constructing a Kempe component $K^{ij}(u)$ for colours i and j and vertex u, and interchanging colours. Note that a Kempe component can be constructed in $O(n)$ time using a breadth-first search, and that colours can be interchanged also in $O(n)$ time.

1.3 Degree 5:

Let z be a vertex of degree 5 in G. Let the adjacent vertices be u, v, w, x and y. Deleting z gives a pentagonal face $uvwxy$. To maintain the triangulation, we must add two diagonals to the pentagonal face. We first try uw and ux, provided that neither edge is in G already. If uw is in G, then vx and vy are not, as G is planar. Similarly, if ux is in G, then vy and wy are not. In either case, we relabel the graph, so that the common vertex of the two new edges is u. After colouring this reduced graph, there are only four colourings on (u, v, w, x, y) possible, up to isomorphism: $(1,2,3,2,3)$, $(1,2,3,4,2)$, $(1,2,3,4,3)$, and $(1,2,3,2,4)$. In the first case, colour 4 is available, and we apply it to z. In the other cases, we note that each has two vertices of the same colour, and re-label both vertices and colours so that the two vertices of the same colour are v and y, with colour 2, so that (without loss of generality) we have (u, v, w, x, y) is coloured $(1,2,3,4,2)$. An example is given by the graph CK_0 in Fig. 3.

If $w \in K^{13}(u)$, we interchange colours in $K^{13}(u)$ releasing colour 1 to apply to z. Otherwise, if $x \in K^{14}(u)$, we interchange colours in $K^{14}(u)$, again releasing colour 1 to apply to z.

At this point, Kempe observed that a $(1,3)$-chain from u to w precludes the existence of a $(2,4)$-chain from v to x, and that a $(1,4)$-chain from u to x precludes the existence of a $(2,3)$-chain from w to y. Interchanging colours in both $K^{24}(v)$ and $K^{23}(y)$ would create a colouring $(1,4,3,4,3)$ on the pentagon, thereby releasing colour 2 for z. The error here lies in the assumption that both intherchanges may be performed simultaneously [7]. Note that in the graph CK_0 of Fig. 3, that interchanging colours in $K^{24}(v)$ creates a $(2,3)$-chain from w to y.

2. Iterating Kempe's Algorithm

Let (u, v, w, x, y) be the pentagonal face of $G - z$. If this cycle is coloured with 4 colours, there is exactly one colour that occurs twice, with one vertex adjacent to the vertices with the repeated colour. This vertex is called the *apex* of the pentagon. Without loss of generality, let (u, v, w, x, y) be coloured $(1,2,3,4,2)$, so that u is the apex. Suppose that $w \in K^{13}(u)$. It follows that $x \in K^{24}(v)$. Interchange colours in $K^{24}(v)$. The vertices (u, v, w, x, y) are now coloured $(1,4,3,4,2)$. The new apex of the pentagon is w. See Fig. 2.

The $(1,3)$-Kempe chain from w to u still exists. If $y \in K^{23}(w)$, then we can interchange colours in $K^{23}(w)$, releasing colour 3 for z. Otherwise $u \in K^{14}(x)$, so that we can interchange colours in $K^{14}(x)$. The pentagon (u, v, w, x, y) is now coloured $(1,4,3,1,2)$, so that y has now become the apex. We iterate this technique.

2.1 Algorithm

Let (u, v, w, x, y) be a pentagon coloured (i, j, k, ℓ, j), using 4 colours, so that u is the apex. Suppose further that an $i\ell$-Kempe chain from u to x exists.

1. Construct $K^{ik}(u)$. If $w \in K^{ik}(u)$, there is no ik-Kempe chain from u to w. Interchange colours in $K^{ik}(u)$. The pentagon (u, v, w, x, y) is now coloured (k, j, k, ℓ, j). Colour z with colour i and stop.
2. Otherwise there is an ik-Kempe chain from u to w. Construct $K^{j\ell}(v)$ and interchange colours in it. The pentagon (u, v, w, x, y) is now coloured (i, ℓ, k, ℓ, j). The apex is now w. The ik-Kempe chain from u to w still exists.
3. Relabel the vertices and colours so that (w, x, y, u, v) becomes (u, v, w, x, y) and (k, ℓ, j, i, ℓ) becomes (i, j, k, ℓ, j). The ik-Kempe chain from u to w of step 2 has become an $i\ell$-Kempe chain from u to x. Go to step 1.

Note that this relabelling rotates the arrangement of the 4 colours on the pentagon. This is illustrated in Fig. 2, where the apex of the pentagon is indicated by an arrow. We repeat this iteration until Kempe's algorithm succeeds. We say that executing the 3 steps of the above algorithm consitutes *one iteration*. We say that a graph causes the algorithm to iterate, if it must perform at least one iteration before terminating.

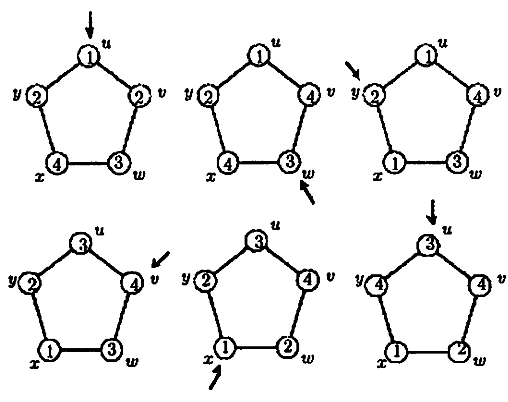


Fig. 2, Rotation of the apex

Under what conditions does Algorithm 2.1 fail to terminate? Since the changes created by interchanging colours in a Kempe component are not local to the face under consideration, it is extremely difficult to analyze. Computer experimentation seems to indicate that such graphs are very rare. Since there are a finite number of colourings for any given graph, if the algorithm fails to terminate, there must be a repeated colouring, after which the algorithm cycles. Suppose that G is a graph for which the algorithm cycles. Let the pentagon (u, v, w, x, y) be initially coloured $(1, 2, 3, 4, 2)$, with u as the apex. Notice that after 5 iterations, the apex has returned to its original position, but that the colours have been permuted according to the permutation $(1, 3, 2, 4)$, which has period 4 (refer to Fig. 2). Thus, if the algorithm is to cycle, the number of iterations required to return the graph to its original colouring is a multiple of 20.

In an attempt to find graphs for which the algorithm cycles, we constructed graphs by pseudo-randomly inserting vertices of degree 3, 4 and 5 into K_4, according to some fixed ratio, in order to create a pseudo-random planar triangulation. Each graph was then 4-coloured by Kempe's algorithm, using Algorithm 2.1 to deal with vertices z of degree 5. Any graphs arising during the course of the algorithm for which Algorithm 2.1 iterated at least once was saved to a file. Every such graph has minimum degree 5. Some of these graphs (the smaller ones) were tested by constructing all possible colourings, and counting the number of colourings requiring 0 iterations, 1 iterations, 2 iterations, etc. The following points are in order.

1. We coloured planar graphs with up to $n = 3000$ vertices. We found they were coloured in a short time by this algorithm. A formal upper bound of $O(n^2)$ can be proved, provided that all graphs terminate within a fixed number of iterations (e.g. 20).
2. Although a simple upper bound of $4 \cdot 3^{n-1}$ distinct colourings of a connected planar graph on n vertices can be written, the number of colourings found in these triangulations of degree 5 and more was approximately $O(1.3^n)$, allowing us to test exhaustively any given triangulation up to about $n = 55$ vertices in a reasonable amount of time.
3. The number of colourings requiring j iterations was roughly 1/2 the number of colourings requiring $j - 1$ iterations, and quite frequently less.
4. Taking the planar dual of a triangulation which causes Algorithm 2.1 to iterate, and truncating it, often resulted in finding more graphs for which the algorithm iterated.

3. The *CK* Family of Graphs

After some time, we discovered a family of graphs for which Algorithm 2.1 cycles, the smallest of which, CK, has 16 vertices arranged in 3 concentric rings of 5 vertices, with a single centre vertex. Recall that z is assumed to be in the outer face. Fig. 3 shows a number of colourings of CK, denoted CK_0, CK_1, CK_2, etc. When the algorithm is about to colour z, CK may have one of the colourings CK_i.

3.1 Lemma. Algorithm 2.1 cycles when input colouring CK_0, through a sequence of 20 distinct colourings.

Proof. When Algorithm 2.1 is applied to CK_0, we get a sequence of colourings as follows. After 1 iteration the colouring is CK_1, after 2 iterations it is CK_2, and so forth. After 4 iterations (CK_4), the apex has moved to vertex x. The colouring is now identical to CK_0, except that the entire graph has been rotated. The rotation of the outer pentagon can be described as (u, x, v, y, w). As this permutation has period 5, we conclude that CK_{20} is the first colouring identical to CK_0.

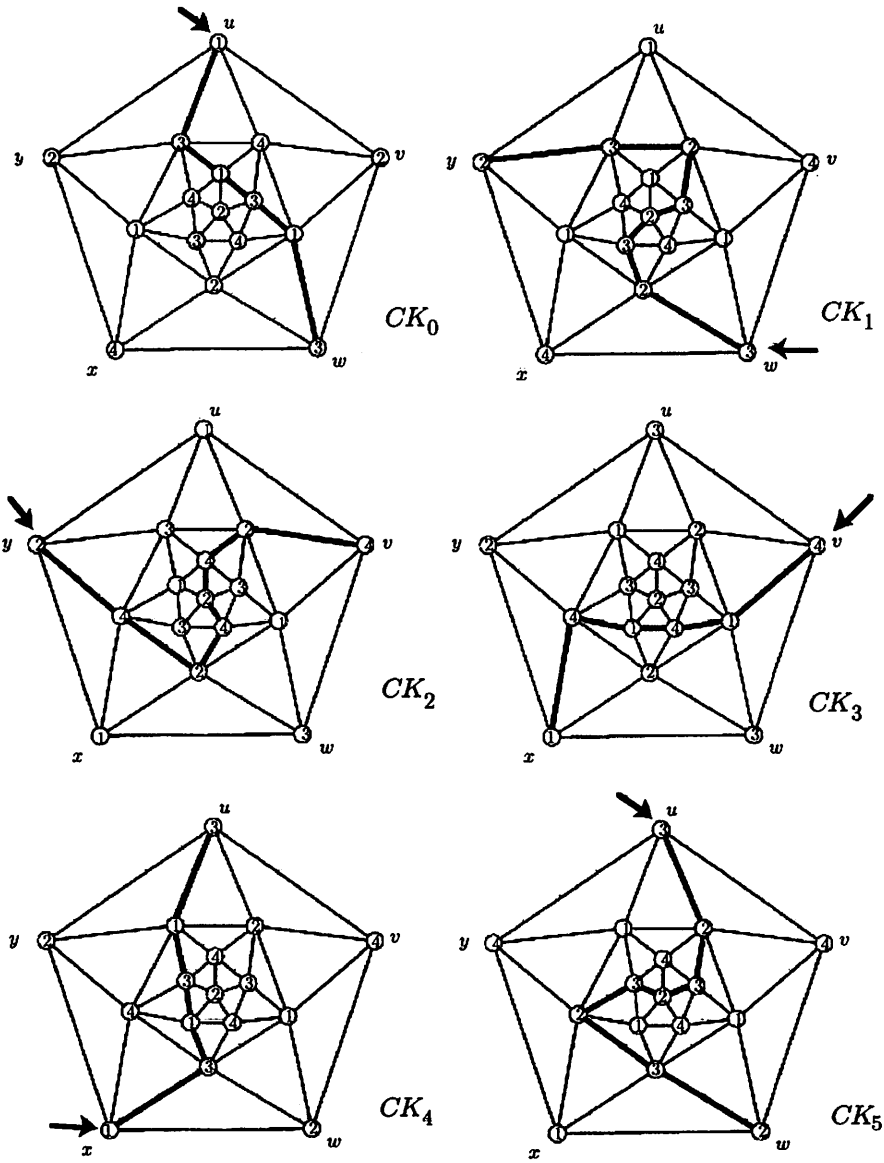


Fig. 3, Colourings of CK

Notice that the centre vertex of CK has the same colour in each of $CK_0, \ldots, CK_{19}$, as it is not affected by any of the colour-interchanges in any Kempe component.

3.2 Lemma. The colourings CK_0, CK_1, and CK_2 are all non-isomorphic as coloured graphs.

Proof. In CK_0 and CK_2 the second outermost ring is coloured in 4 colours. In CK_1, only 3 colours are used. Notice that the central vertex is coloured 2 in all graphs CK_i. In CK_2, the apex of the outer ring is coloured the same as the central vertex. This is not the case in CK_0. This shows that CK_0, CK_1, and CK_2 are distinct as coloured graphs.

We remark that CK_3 is isomorphic to CK_1, with $(3,4)(v,w)(u,x)$ being a permutation of vertices and colours mapping CK_3 to CK_1.

Algorithm 2.1 is used to colour a vertex z of degree 5 in a planar triangulation. Suppose now that G is any coloured graph to which the algorithm is applied. G will always be a near triangulation with one pentagonal face, which we always take to be the outer face. Without loss of generality, we will denote this face by (u,v,w,x,y), coloured $(1,2,3,4,2)$. Construct two new coloured graphs G_A and G_B as follows. Add 2 pentagonal rings (a,b,c,d,e) and (u',v',w',x',y') to the pentagonal face of G, as indicated by Fig. 4. In G_A, (a,b,c,d,e) is coloured $(1,3,4,3,4)$, and (u',v',w',x',y') is coloured $(1,2,3,4,2)$. In G_B, (a,b,c,d,e) is coloured $(2,1,3,4,1)$, and (u',v',w',x',y') is coloured $(1,2,3,4,2)$. Notice that as coloured graphs, G_A and G_B are not isomorphic.

3.3 Theorem. Let G be any coloured graph for which Algorithm 2.1 cycles. Then Algorithm 2.1 also cycles on G_A and G_B.

Proof. The successive colourings of G_A and G_B as Algorithm 2.1 is applied are shown in Figs. 4 and 5. Write $G_{A_0} = G_A$ and $G_{B_0} = G_B$. Notice that for $G_{A_0}, G_{A_1}, G_{A_2}, G_{A_3}$, the ij-Kempe chain from the apex of G_A to the base contains the ij-Kempe chain of G. Furthermore, interchanging colours in $K^{13}(u)$ in G_{A_0} also interchanges colours in $K^{13}(u)$ in G. Algorithm 2.1 applied to G_{A_0} induces the sequence of colourings $G_0, G_1, G_2, \ldots$ of G. As in Lemma 3.1, we see that G_{A_4} is identical to G_{A_0}, except that the graph has been rotated. It follows that Algorithm 2.1 also cycles on G_A, with period 20. The proof for G_B is identical, although the sequence of colourings is different.

Theorem 3.3 allows us to construct a family of graphs for which Algorithm 2.1 cycles. Beginning with CK, we have $CK_A, CK_B, CK_{AA}, CK_{AB}$, $CK_{BA}, CK_{BB}, \ldots$, etc. Each of these coloured graphs cycles with period 20.

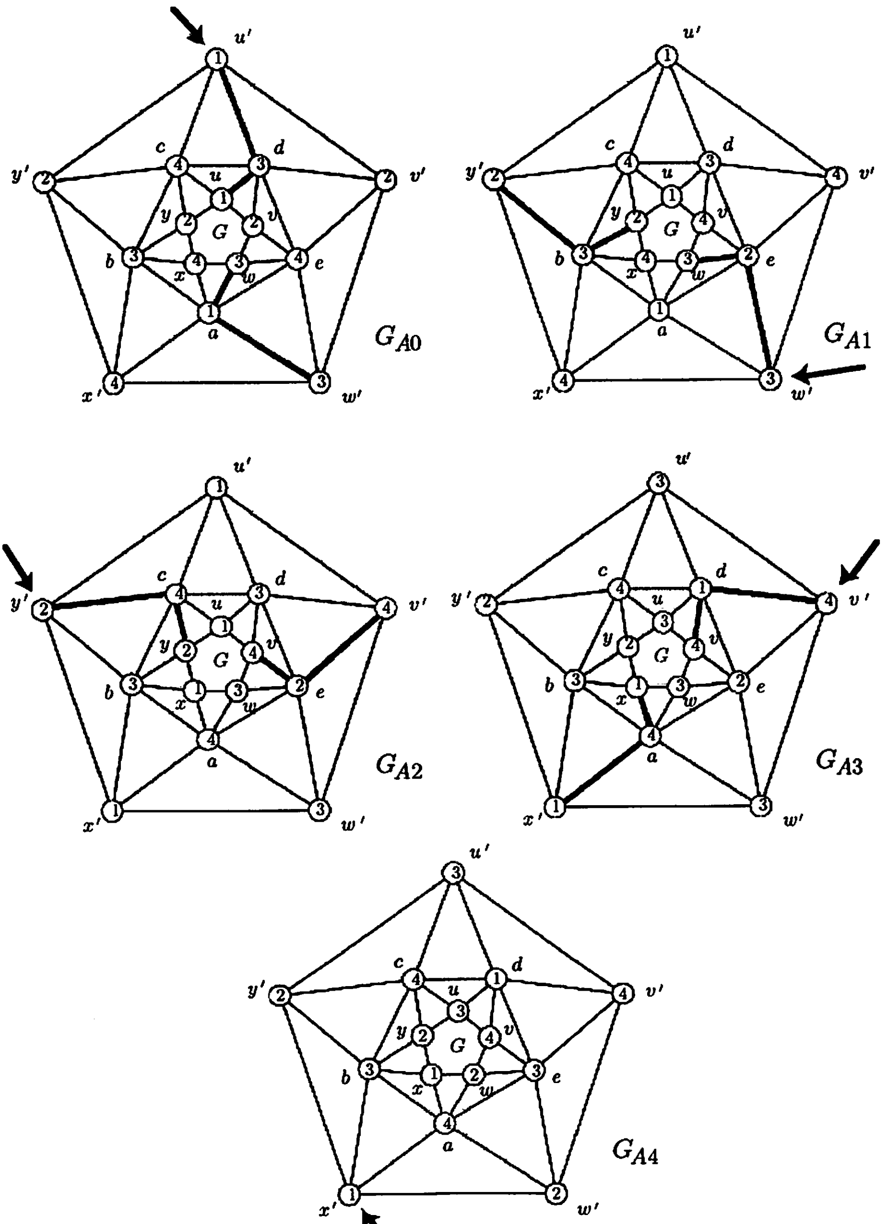


Fig. 4, Colourings of G_A

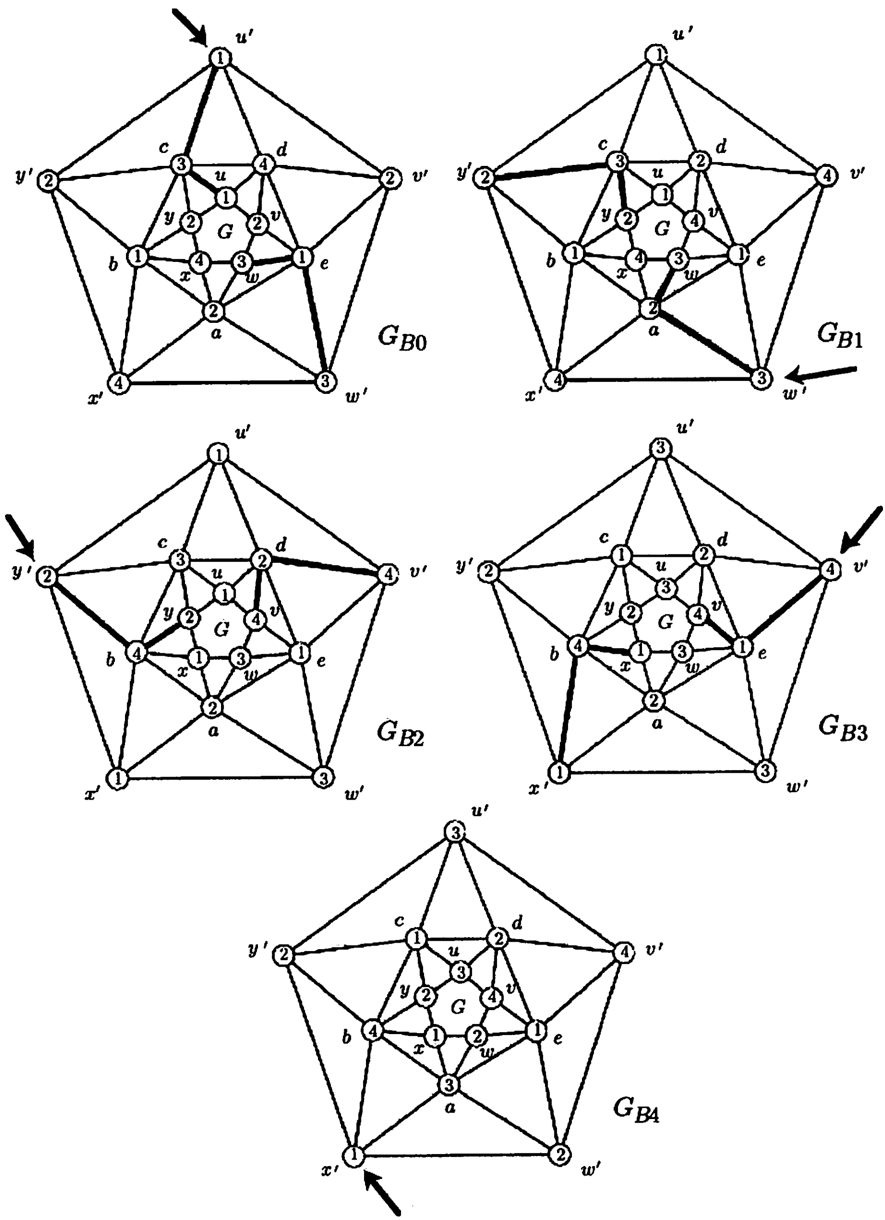


Fig. 5, Colourings of G_B

Note: We constructed all possible colourings of CK_A, and applied Algorithm 2.1 to each one. The only colourings which cycled are those in the sequence of CK_{A_0} and CK_{B_0}. We also constructed all possible colourings of the graph obtained from CK by adding only one ring (a, b, c, d, e) and applied Algorithm 2.1 to each one. None of these graphs cause the algorithm to cycle.

An additional transformation of the graphs in the CK-family is possible.

3.4 Construction. Let (u, v, w) be any face in any coloured graph G in this family. Let H be any coloured triangulation. We can choose any face of H and identify it with (u, v, w), embedding H inside the face, where it may be necessary to permute the colours of H to make them agree with the colours of (u, v, w). The result is a triangulation G' with a separating 3-cycle (u, v, w).

All the edges of the Kempe chains of G that cause Algorithm 2.1 to cycle are still present in G'. Consequently, Algorithm 2.1 will also cycle with input G', with period a multiple of 20.

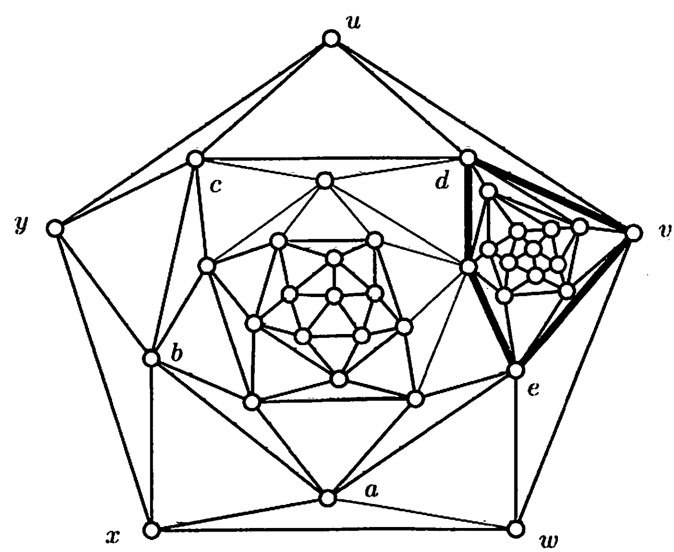


Fig. 6, Creating a separating 4-cycle

Notice that in the graphs G_A and G_B of Figs. 4 and 5, that the edges of the 5-cycles (u, v, w, x, y) and (a, b, c, d, e) are not used in any of the Kempe chains.

3.5 Construction. Let rt be any edge of the pentagon (a, b, c, d, e) in a graph G of the CK family of graphs. Edge rt is contained in 2 triangles, so that it is part of a 4-cycle (q, r, s, t). Let H be any near triangulation whose outer face is a 4-cycle coloured the same as (q, r, s, t). Delete the edge rt from G and identify the outer face of H with the 4-cycle (q, r, s, t), embedding it inside the face. The result is a triangulation G' with a separating 4-cycle (q, r, s, t).

All the edges of the Kempe chains of G that cause Algorithm 2.1 to cycle are still present in G'. Consequently, Algorithm 2.1 will also cycle with input G', with period a multiple of 20. An example is shown in Fig. 6.

These constructions can be used to generate a large number of graphs for which Algorithm 2.1 cycles. They are all based on the Kempe chains in the graph CK. We have not found any other graphs for which the algorithm cycles.

Question: Are there any graphs which cause Algorithm 2.1 to cycle which are not based on the graph CK?

4. The Modified Algorithm

We now present a modified version of the iterated Kempe algorithm, designed to avoid the problems with the CK family of graphs. If a graph G has a separating 3-cycle or 4-cycle, we use the method described in Saaty and Kainen [14] to split G into 2 subgraphs H and K and colour each part separately. As H and K will both have fewer vertices than G, we can assume that the algorithm will colour them correctly.

4.1 Separating 3-Cycle.

Let G have a separating 3-cycle (u, v, w). Let H be the subgraph inside (u, v, w) and let K be the subgraph outside (u, v, w). Note that (u, v, w) is a cycle in both H and K. Colour H and K independently. Permute the colours of K, if necessary, so that (u, v, w) is coloured identically in both H and K. This gives a colouring of G.

4.2 Separating 4-Cycle.

Let G have a separating 4-cycle (u, v, w, x). let H be the subgraph inside (u, v, w, x) and let K be the subgraph outside (u, v, w, x). Note that (u, v, w, x) is a cycle in both H and K. Add the diagonal edge vx to both H and K to create triangulations, and colour $H + vx$ and $K + vx$ independently. In both H and K, vertices v and x have different colours. Without loss of generality, let v have colour 1 and x have colour 2 in both H and K. Let u have colour 3 in H. We can permute the colours of K if necessary so that u also has colour 3 in K. If w has the same colour in H and K, we have a colouring of G. Refer to Fig. 1.

Otherwise we can assume that w has colour 3 in H, but colour 4 in K. If $u \in K^{34}(w)$ in K, interchange colours in $K^{34}(w)$. Vertices u and w are now both coloured 3 in H and K, giving a colouring of G.

If, however, $u \in K^{34}(w)$ in K, there is then no $(1,2)$-Kempe chain from v to x in K. Now colour $H + uw$ to get a colouring of H in which u and w have different colours. Without loss of generality, we can take these colours to be 3 and 4, respectively. If v and x are coloured differently in H, we can take v to be coloured 1 and x to be coloured 2, as in K, giving a colouring of G. But if v and x are both coloured 2, we then interchange colours in $K^{12}(v)$ in K, so that H and K agree on the 4-cycle (u, v, w, x). We again get a colouring of G

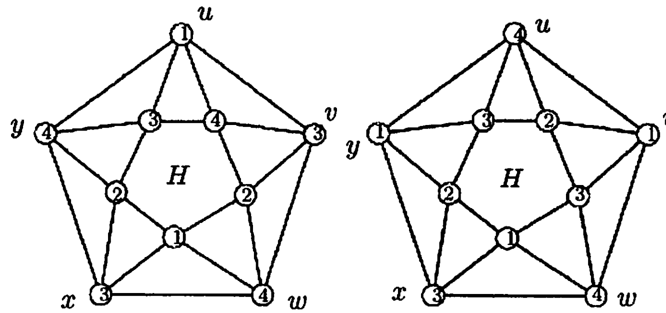
In each of the graphs which cause Algorithm 2.1 to cycle, the missing vertex z is adjacent to a 5-cycle (u, v, w, x, y), which is in turn adjacent to another 5-cycle (a, b, c, d, e). If every vertex of (u, v, w, x, y) has degree 5 in G, then there is an adjacent 5-cycle (a, b, c, d, e), since G is a triangulation. This makes it easy to detect whether there is an adjacent 5-cycle. We modify the algorithm to handle such graphs. Let (u, v, w, x, y) be the outer pentagon, and let (a, b, c, d, e) be the next pentagon.

4.3 Colouring Algorithm

Given a triangulation G on n vertices.

1. If G has a vertex of degree 3, proceed as in 1.1.
2. If G has a vertex of degree 4, proceed as in 1.2.
3. If G has a vertex of degree 5, let (u, v, w, x, y) be the outer pentagon of $G - z$.
   - **3a.** If every vertex of (u, v, w, x, y) has degree 5, then (u, v, w, x, y) is adjacent to another pentagon (a, b, c, d, e). Construct H by deleting the pentagon (u, v, w, x, y). Colour H. There are 2 possible colourings of (a, b, c, d, e) – either $(1,2,3,4,2)$ or $(1,2,1,2,3)$. In the first case, colour $G - z$ according to Fig. 7a. In the second case, colour according to Fig. 7b. Assign colour 2 to z.
   - **3b.** If there is a vertex of (u, v, w, x, y) with degree greater than 5, determine whether it is part of a separating 3-cycle or 4-cycle. If so, proceed as in 4.1 or 4.2.
   - **3c.** Otherwise (u, v, w, x, y) is not adjacent to a pentagon, and has no vertex forming part of a separating 3-cycle or 4-cycle. Proceed as in Algorithm 2.1.

A proof that Algorithm 4.3 always succeeds would be a proof of the 4-Colour Theorem. If there is *any* proof of the 4-Colour Theorem which avoids computers to check irreducible configurations, it may follow from this algorithm, which is based on the simple idea of iterating Kempe's technique. The existence of the CK-family of graphs with their remarkable properties in relation to the algorithm strengthens this idea.



(a) (b)

Fig. 7, Colouring (u, v, w, x, y)

There is another technique which can be used to colour the CK-family of graphs. The colour interchanges performed by Algorithm 2.1 are for every colour pair except $(1, 2)$ and $(3, 4)$. Notice that in the colouring G_{A_0} of Fig. 4, vertices w' and x' are part of a separating 6-cycle coloured 3 and 4. If we interchange colours in $K^{12}(u')$, the result is to destroy the $u'w'$-Kempe chain. Similarly, if we interchange colours in $K^{34}(w')$ in G_{B_0}, we destroy the $u'w'$-Kempe chain. Algorithm 2.1 could be modified to count the iterations. If it finds that a graph G causes 20 iterations, it could then interchange colours in one of $K^{12}(u')$ or $K^{34}(w')$, depending on whether a separating 6-cycle containing w' and x' exists.

Complexity of Algorithm 4.3

Each step of Algorithm 4.3 can be programmed to run in linear time. If the number of times that Algorithm 2.1 must iterate can be limited to a constant k (say $k < 20$), then the entire algorithm will run in quadratic time. Let G have n vertices. Since G is a planar triangulation, it has $3n-6$ edges. The degrees of the vertices of G can be computed by summing the incident edges for each vertex v, and storing the results in an array. They must also be sorted into buckets according to degree 3, 4, 5, and more than 5. These steps can be done in $O(n)$ time.

A vertex z of minimum degree is then chosen. If z has degree 3, the algorithm colours $G - z$ recursively. It takes $O(n)$ steps to do this, and re-attach z. If z has degree 4, two Kempe components may have to be constructed. $O(n)$ steps are required to build a Kempe component and interchange colours. Thus, $O(n)$ steps are required.

If z has degree 5, we must determine whether a vertex of the outer pentagon (u, v, w, x, y) is incident on a separating 3-cycle or 4-cycle. We first check whether each vertex has degree 5 in G. Refer to Fig. 6. Consider vertex u. If u has degree 5, triangles (u, v, d), (y, u, c), and (v, d, c) are uniquely determined, as G is a triangulation. None of these can be separating cycles. If u has degree 6 or more, we take each adjacent vertex t in turn, and test whether it is adjacent to v and y. If we find such a vertex t, it determines a triangle, say (u, v, t). It is easy to determine whether this is a separating 3-cycle. We do this for each vertex on the outer pentagon, and determine the incident separating 3-cycles in $O(n)$ time. This may also determine 5 vertices a, b, c, d, e, as in Fig. 6. We now take each of these vertices in turn, and decide whether there is a vertex t adjacent to 2 consecutive vertices of this sequence. Every separating 4-cycle can be detected in this way. $O(n)$ steps are required.

If 5 vertices a, b, c, d, e adjacent to u, v, w, x, y as in Fig. 6 do not exist, then G cannot be in the CK family of graphs, and Algorithm 2.1 is applied. Algorithm 2.1 must build a number of Kempe components, and interchange colours in them. A breadth-first seach can build a Kempe component in $O(n)$ steps. It also takes $O(n)$ steps to interchange colours. Thus, each iteration of 2.1 is linear. If the number of iterations is bounded by a constant, the complexity of Algorithm 2.1 will also be linear.

As there are n vertices in G, and $O(n)$ steps is required to colour each vertex, the complexity of Algorithm 4.3 is $O(n^2)$.

References

1. F. and E.R. Swart, "A Systematic Approach to the Determination of Reducible Configurations in the Four-Color Conjecture", Journal of Combinatorial Theory (B) 25 (1977), 339-362.
2. K. Appel and W. Hake, "The Solution of the Four-Color-Map Problem", Scientific American (1976), 108-119.
3. K. Appel and W. Haken, "Every Planar Map Is Four Colorable. Part I: Discharging", Illinois Journal of Mathematics 21 (1977), 429-490.
4. K., W. Haken and J. Koch, "Every Planar Map is Four Colorable. Part II: Reducibility", Illinois Journal of Mathematics 21 (1977), 491-567.
5. G.D. Birkhoff, "The Reducibility of Maps", American Journal of Mathematics 35 (1913), 195-206.
6. J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications*, American Elsevier Publishing, New York, 1976.
7. P. J. Heawood, "Map-Colour Theorem", Quarterly Journal of Pure and Applied Mathematics 24 (1890), 332-338.
8. H. Heesch, "Untersuchungen zum Vierfarbenproblem", Bibliog. Institut, AG, Mannheim, 1969.
9. A. B. Kempe, "On the Geographical Problem of the Four Colours", American Journal of Mathematics 2 (1879), 193-200.