involve
a journal of mathematics

How false is Kempe's proof of the Four Color Theorem? Part II

Ellen Gethner, Bopanna Kallichanda, Alexander S. Mentis,
Sarah Braudrick, Sumeet Chawla, Andrew Clune, Rachel Drummond,
Panagiota Evans, William Roche and Nao Takano

mathematical sciences publishers

2009 Vol. 2, No. 3

INVOLVE 2:3(2009)

How false is Kempe's proof of the Four Color Theorem? Part II

Ellen Gethner, Bopanna Kallichanda, Alexander S. Mentis,
Sarah Braudrick, Sumeet Chawla, Andrew Clune, Rachel Drummond,
Panagiota Evans, William Roche and Nao Takano

(Communicated by Ken Ono)

We continue the investigation of A. B. Kempe's flawed proof of the Four Color Theorem from a computational and historical point of view. Kempe's "proof" gives rise to an algorithmic method of coloring plane graphs that sometimes yields a proper vertex coloring requiring four or fewer colors. We investigate a recursive version of Kempe's method and a modified version based on the work of I. Kittell. Then we empirically analyze the performance of the implementations on a variety of historically motivated benchmark graphs and explore the usefulness of simple randomization in four-coloring small plane graphs. We end with a list of open questions and future work.

1. Introduction

The Four Color Theorem for plane graphs states that, given a plane graph Γ, the vertices of Γ can be properly colored with at most four colors. While the Four Color Theorem was proven in 1977 through the use of a computer and irreducible sets [Appel and Haken 1976/77; 1977; Appel et al. 1977; Robertson et al. 1996; 1997], no proof has been found that can be verified by a human without the use of a computer. Alfred Kempe seemingly came close to accomplishing this in 1879 when he presented a proof of the Four Color Theorem in [Kempe 1879]; however, his proof contained a flaw, discovered by Heawood [1890] and independently by de la Vallée Poussin in 1896 [Wilson 2002b]. Although Kempe was unable to repair the flaw, his innovation of Kempe chains and Kempe chain switches remain useful to graph theorists, and it is interesting to explore the boundaries of his technique [Gethner and Springer 2003]. In particular, we focus our attention on the work of

MSC2000: 05C15, 68R10, 90C35.
Keywords: four color theorem, Kempe, Kittell.

Errera, who was the first person to study the importance of the order in which the vertices are labeled [Errera 1921]. For a comprehensive history of the Four Color Theorem, see [Wilson 2002a; 2002b; Ore 1967; Fritsch and Fritsch 1998; Biggs et al. 1986].

Following [Errera 1921; Hutchinson and Wagon 1998; Gethner and Springer 2003; Wagon 2009], we implemented Kempe's method of proof as a recursive algorithm (Algorithm Kempe) on different vertex labelings for some well known graphs of nine vertices or more. For labelings resulting in the algorithm's inability to properly four-color the graphs, we identify vertices that cause irrevocable Kempe chain failures (the source of the flaw in Kempe's proof), and quantify the graphs' failure rates. In acknowledgement that Algorithm Kempe sometimes correctly four-colors the vertices of a plane graph, we explore some improvements to Algorithm Kempe including random selection among all Kempe chain choices and using random Kempe–Kittell chain switches to overcome irrevocable Kempe chain tangles, following [Kittell 1935; Hutchinson and Wagon 1998; Wagon 2002; 2009; Archuleta and Shapiro 1986; Morgenstern and Shapiro 1991]. While there may be different flaws that also result in failure to four-color a plane graph, our improvements focus solely on circumventing the flaw identified by Heawood and Poussin, since that is the flaw addressed by our implementation of Kittell's approach. Where Kempe–Kittell chain switches allow Algorithm Kempe to continue, we correlate the identified vertex with the number of Kempe–Kittell chain switches required to overcome the tangle.

2. Definitions and algorithm

It is important to understand Kempe's alleged proof and the flaw that led to our investigations. For completeness and ease of reference, the following definitions and algorithm are taken directly from [Gethner and Springer 2003]. In all of the following, R, G, B, Y refer to the four possible colors, and C_i is an element of $\{R, G, B, Y\}$.

Definition 1 (C_1C_2-Kempe chain). Let Γ be a plane graph whose vertices have been properly colored and suppose $v \in V(\Gamma)$ is colored C_1. The *C_1C_2-Kempe chain containing v* is the maximal connected component of Γ that contains v and contains only vertices colored C_1 or C_2.

Importantly, the maximality of the set of colored vertices in a C_1C_2-Kempe chain guarantees that interchanging all occurrences of C_1 and C_2 preserves the proper coloring of Γ.

Definition 2 (C_1C_2-Kempe chain switch). Let K be a C_1C_2-Kempe chain. A *C_1C_2-Kempe chain switch* interchanges all values of C_1 and C_2 in K.

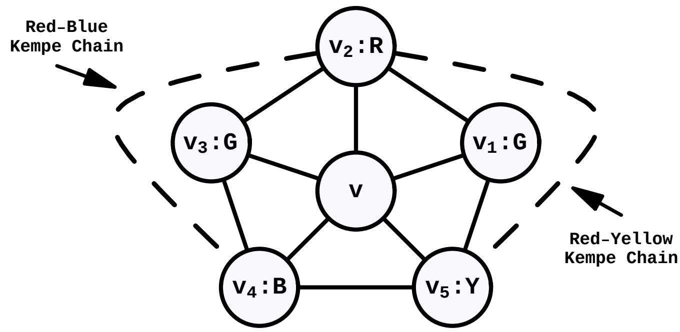


Figure 1. Setup for the faulty case in Kempe's proof.

We need one more notion to illustrate the potential flaw in Kempe's method. To help visualize the setup, see Figure 1.

Definition 3 (Irrevocable Kempe chain tangle). Let Γ be a plane graph, all of whose vertices, with the exception of one vertex v of degree 5, have been properly colored with four colors. Denote the five neighbors of v in cyclic counterclockwise order by v_1, v_2, v_3, v_4, v_5, and assume they are colored G, R, G, B, Y respectively. Moreover, assume that the RB-Kempe chain of v_2 contains v_4, and that the RY-Kempe chain of v_2 contains v_5.

Denote the GB-Kempe chain containing v_1 by K_1 and the GY-Kempe chain containing v_3 by K_2. We say that Algorithm Kempe causes an *irrevocable Kempe chain tangle on vertex* v if either

- following a GB-Kempe chain switch on K_1 by a GY-Kempe chain switch on K_2 causes v_5 to be recolored G, or
- following a GY-Kempe chain switch on K_2 by a GB-Kempe chain switch on K_1 causes v_4 to be recolored G.

In particular, at least one of the original barriers afforded by either the RB-Kempe chain containing v_2 and v_4, or the RY-Kempe chain containing v_2 and v_5 has been broken by two successive GX-Kempe chain switches, where $X \in \{Y, B\}$. Moreover, the second GX-Kempe chain contains two vertices in the neighborhood of v, which reintroduces a vertex colored G as a neighbor of v; thus the procedure has not made G available for vertex v.

We use the adjective *irrevocable* in Definition 3 because under the initial hypotheses, a Kempe chain tangle might occur: that is, one of either the RB-Kempe chain or the RY-Kempe chain may be "broken" by the two successive GX-Kempe chain switches, but the procedure need not force any of the neighbors of v to be recolored with G. In that case, v will be properly colored with G.

With these definitions, we can now describe Algorithm Kempe.

Algorithm Kempe.

INPUT:

- A connected plane graph Γ with n vertices, labeled (in some order) with distinct elements from $\{1, \ldots, n\}$.
- An ordering (C_1, C_2, C_3, C_4) of the set of permissible colors $\Sigma = \{R, G, B, Y\}$.

OUTPUT:

- Either a proper vertex coloring of Γ with colors from Σ, or
- the message "Kempe's algorithm has encountered an irrevocable Kempe chain tangle at vertex v and hence has failed to properly four-color Γ."

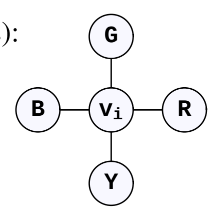
Gadget relabel (relabel the vertices):

- Search $\Gamma_0 := \Gamma$ for the first occurrence of a vertex of degree five or less; the existence of such a vertex is guaranteed by Euler's formula. The first occurrence is dictated by the given ordering of the vertices. Call this vertex v_1.
- Recursively label the other $n-1$ vertices of Γ, choosing for v_{i+1} $(0 \leq i < n)$ the first occurrence of a vertex of degree five or less in $\Gamma_i := \Gamma \setminus v_i$.

Gadget greed (color greedily whenever possible):

- Color v_n in Γ_{n-1} with the available color of lowest index from C. In this case, since no colors have been used, v_n will be colored C_1.
- Color v_{n-1} in Γ_{n-2} with the available color of lowest index in C; if v_{n-1} is not adjacent to v_n, then v_{n-1} is colored C_1. On the other hand, if v_{n-1} is adjacent to v_n, color v_{n-1} is colored C_2.
- In general (if possible) color v_i in Γ_{i-1} with the available color of lowest index from C.

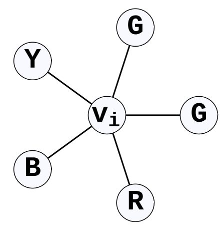
Gadget 4 (perform Kempe chain switches on degree four vertices):



- We encounter a vertex v_i of degree four that cannot be greedily colored. That is, suppose degree $v_i = 4$ and the neighbors are colored R, G, B, Y (say) in counterclockwise order, as on the right.
- If there is an RB-Kempe chain containing both the R and B neighbors of v_i, there cannot be a YG-Kempe chain that contains *both* of the Y and G neighbors of v_i. In that case a YG-Kempe chain switch leaves a color available for v_i.
- Otherwise, if there is no RB-Kempe chain containing both the R and B neighbors of v_i, perform an RB-Kempe chain switch to make a color available for v_i.

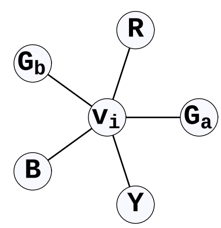
Gadgets 5_1 **and** 5_2 (Kempe chain switches on degree-five vertices): Suppose we encounter a vertex v_i of degree five that cannot be greedily colored; a priori, one color is used exactly twice and the other three are used exactly once on the five neighbors of v_i. Without loss of generality, suppose the twice-used color is G. Up to rotation and reflection, only two configurations can occur, illustrated in the two diagrams below.

Gadget 5_1 (degree $v_i = 5$; two G neighbors next to each other):



- In Configuration 1, a gadget much like Gadget 4 will succeed in coloring v_i. Suppose the five neighbors of v_i are colored, in counterclockwise order, by *GGYBR*.
- If there is no *RY*-Kempe chain containing both Y and R neighbors of v_i, then a *RY*-Kempe chain switch will leave a color available for v_i.
- Therefore, assume there is an *RY*-Kempe chain containing both Y and R neighbors of v_i. Thus a *GB*-Kempe chain containing the B neighbor of v_i contains neither of the G neighbors of v_i.
- In that case, a *GB*-Kempe chain switch makes B available for v_i.
- In all cases, v_i can be properly colored.

Gadget 5_2 (degree $v_i = 5$; two G neighbors are separated by another neighbor of v_i):



- This is the case in which an irrevocable Kempe chain tangle might occur, causing Algorithm Kempe to halt before completing a proper four-coloring of the graph. Suppose the neighbors of v_i are colored in counterclockwise order by G_aRG_bBY (at this point, it is helpful to distinguish between the two G vertices).
- If there is an *RB*-Kempe chain that does not contain both the R and B neighbors of v_i then an *RB*-Kempe chain switch leaves a color available for v_i.
- If there is an *RY*-Kempe chain that does not contain both R and Y neighbors of v_i then an *RY*-Kempe chain switch makes a color available for v_i.
- Otherwise, we must attempt both a G_aB-Kempe chain switch followed by a G_bY Kempe chain switch (or vice versa).
- If no irrevocable Kempe chain tangle occurs, then we successfully color v_i with G and move on to vertex v_{i-1}.
- Otherwise, halt and return an error message that the offending vertex is v_i.

BEGIN

Step 1: Use Gadget relabel to label the vertices of Γ.

Step 2: For $i = n$ down to 1, attempt to color v_i in graph Γ_{i-1} as follows:

(a) if v_i can be greedily colored in graph Γ_{i-1} by Gadget greed then do so, else

(b) if degree $v_i = 4$ in Γ_i then color v_i using Gadget 4 else

(c) if degree $v_i = 5$ in Γ_{i-1} in configuration 1 then color v_i using Gadget 5_1 else

(d) if degree $v_i = 5$ in Γ_{i-1} in configuration 2 then try to color v_i using Gadget 5_2 employing two Kempe chain switches: try both orders if necessary.

END

Thus, it is obvious that it is possible to color vertices of degree three or less with no more than a fourth color, and it has been shown that it is always possible to color vertices of degree four through the use of Kempe chain switches [Heawood 1890]. Algorithm Kempe only encounters difficulties upon vertices of degree five or more, but it has been shown that Algorithm Kempe will always succeed in properly four-coloring any graph containing eight or fewer vertices (which may contain vertices of degree five or more) [Gethner and Springer 2003]. In light of the fact that Kempe's method of proof works in some, but not all cases, we were interested in identifying patterns of when the algorithm halts without producing a proper four-coloring on our benchmark graphs. In particular, we explore the usefulness of simple randomization when used with Kempe–Kittell chain switches to improve its success on small plane graphs.

3. Results

Identification of vertex failures. We first implemented Algorithm Kempe and explored its success in nine well known, properly four-colored graphs. The first five, shown in Figure 2 (ignore the coloring of vertices for the moment), were introduced in [Heawood 1890], [Fritsch and Fritsch 1998], [Soifer 1997], [Errera 1921] (see also [Hutchinson and Wagon 1998; Wagon 2009]), and Poussin's writings (see [Wilson 2002a]); they are all known counterexamples for Algorithm Kempe with at least one labeling [Gethner and Springer 2003]. The remaining four are the edge graphs of the icosahedron, dodecahedron, octahedron, and cube; since these last three have vertices of degree at most four, Algorithm Kempe must always successfully color them, and they served as benchmarks for our implementations. Further, although the Icosahedron graph is five-regular, we did not expect Kempe's method to fail on any labeling of vertices, and thus that graph served as a benchmark graph as well. See also Open Question 5 in Section 4.

Five groups worked independently to implement Algorithm Kempe and test it on these graphs. For the graphs containing nine vertices or fewer, each group explored Algorithm Kempe's results for all $n!$ labelings. For the graphs containing more than nine vertices, each group independently tested a random subset of at least 9! labelings.

While we expected different failure rates for the graphs with more than nine vertices due to the use of different labeling subsets among the groups, we expected the failure rates for Fritsch, Soifer, and the four benchmark graphs to be the same. Instead, while the benchmark graphs produced no failures, as expected, failure rates did vary for Fritsch and Soifer due to differences in the individual implementations or failure rate calculations. In the case of Group 2, when Gadget 5_2 is required the implementation only tests one of the two possible Kempe chain switch orders, resulting in a higher failure rate. This difference in implementation, however, gives us an idea of how many Kempe chain tangles can be "fixed" by changing the order in which the switches are performed (see Table 1 on the next page).

We initially compared the vertices that caused irrevocable Kempe chain tangles for each implementation on all graphs. Because each group tested all 9! labelings for the two nine-vertex graphs (Fritsch and Soifer), each implementation agreed on the vertices that caused failures for Fritsch and Soifer, as expected. An interesting and unpredicted discovery, however, was that despite the differences in the labeling subsets tested by each group for the graphs containing more than nine vertices, there was considerable consensus among the groups on the vertices that cause failures.

In Figure 2, vertices shown in red are those vertices that all groups found to result in an irrevocable Kempe chain tangle for at least one labeling. Vertices

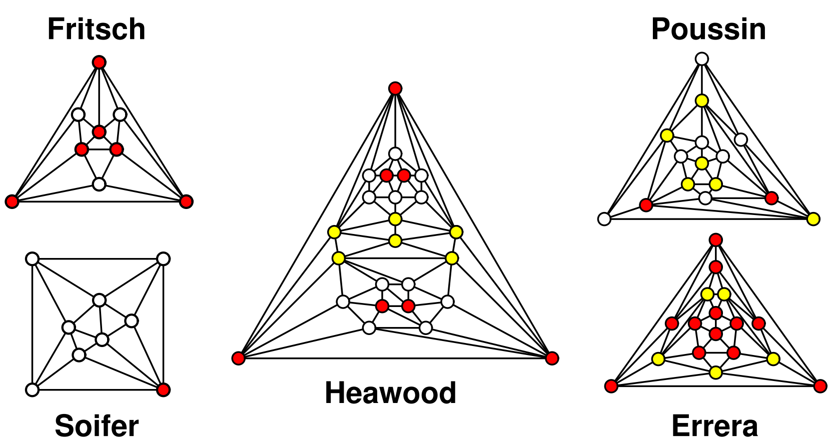


Figure 2. Graph vertex failures.

shown in yellow are those vertices that at least one group, but not all, found to fail. Vertices shown in white were not found by any groups to cause a failure on the labelings tested. As one can see, the failure patterns of the vertices are highly symmetrical for all graphs except the Poussin graph, which itself is a fairly asymmetrical graph. For the Fritsch and Soifer graphs, since all $n!$ labelings were tested, we know that the vertices shown in white will never cause Kempe chain tangles for any labeling. For the remaining graphs, we predict that the vertices shown in yellow would eventually become red as more labelings are explored. We cannot predict anything for the vertices shown in white of degree five or more — they may eventually fail, or they may not. Nevertheless, these results lead us to ask the question: are there commonalities among these vertices that can be exploited to improve Algorithm Kempe? We leave this as an open question.

The next step was to add randomization, studied in [Kittell 1935; Hutchinson and Wagon 1998; Wagon 2002; 2009; Archuleta and Shapiro 1986; Morgenstern and Shapiro 1991], through the application of Kempe–Kittell chain switches [Kittell 1935] and the use of randomization of the choice of Kempe or Kempe–Kittell chain switches, rather than heuristics, at various stages of the algorithm. In contrast to the study of randomization for large graphs in [Archuleta and Shapiro 1986;

Group		F	S	O	C	I	D	P	E	H
1	min	13.947	1.692	0.000	0.000	0.000	0.000	0.610	9.730	7.089
	avg	13.947	1.692	0.000	0.000	0.000	0.000	0.620	9.755	7.124
	max	13.947	1.692	0.000	0.000	0.000	0.000	0.627	9.772	7.153
2	min	14.031	1.783	0.000	0.000	0.000	0.000	0.149	3.350	0.372
	avg	14.083	1.832	0.000	0.000	0.000	0.000	0.153	3.383	0.382
	max	14.131	1.859	0.000	0.000	0.000	0.000	0.156	3.402	0.390
3	min	3.598	0.520	0.000	0.000	0.000	0.000	0.014	8.140	1.089
	avg	3.599	0.523	0.000	0.000	0.000	0.000	0.017	8.168	1.097
	max	3.600	0.525	0.000	0.000	0.000	0.000	0.019	8.186	1.111
4	min	13.687	1.635	0.000	0.000	0.000	0.000	0.165	7.302	0.387
	avg	13.687	1.635	0.000	0.000	0.000	0.000	0.165	7.302	0.387
	max	13.687	1.635	0.000	0.000	0.000	0.000	0.165	7.302	0.387
5	min	13.630	1.620	0.000	0.000	0.000	0.000	0.180	10.680	3.199
	avg	13.635	1.620	0.000	0.000	0.000	0.000	0.186	10.798	3.203
	max	13.637	1.620	0.000	0.000	0.000	0.000	0.190	10.866	3.210

Table 1. Kempe method failure rates by graph. The column heads stand for Fritsch, Soifer, Octahedron, Cube, Icosahedron, Dodecahedron, Poussin, Errera, Heawood.

Morgenstern and Shapiro 1991], we continue to investigate small, historically significant benchmark graphs.

Randomization implementation. In our recursive implementation of the Kempe method, there are several points at which we must choose one among multiple Kempe chains upon which to perform a switch. In both Gadget 4 and Gadget 5_1, in the case where the vertex cannot be greedily colored, there will be up to four Kempe chains from which to choose; our implementation randomly chooses one that results in a successful coloring. In Gadget 5_2, two Kempe chain switches must be performed, but the order of the switches is not specified in the algorithm. Theorem 4 shows that the order in which the switches are performed can influence the success of the operation. In light of this knowledge, we randomize the choice of which Kempe chain switch to perform first, and perform the alternative order only if the first order fails.

Theorem 4 (Gadget 5_2 is order-dependent). *In Algorithm Kempe, Gadget 5_2 is sometimes noncommutative. That is, the order in which one chooses to execute the Kempe chain switches on K_1 and K_2 may matter; in one order an irrevocable Kempe chain tangle can occur, whereas in the other no Kempe chain tangle occurs.*

Proof. It suffices to exhibit a plane graph Γ and a labeling of the vertices of Γ that cause Algorithm Kempe to execute Gadget 5_2 in the following way: upon that execution, one of the two choices of Kempe chain switch orders causes an irrevocable Kempe chain tangle while the other does not. To this end, we call upon the Fritsch graph, which we denote by F. In Figure 3, the labeling of the vertices in F (the uppermost graph) leads to a successful four-coloring with the exception of vertex 1, whereupon Gadget 5_2 must be invoked. Following the arrows marked A, one choice of Kempe chain switch order has been executed successfully, and vertex 1 is colored G. Following the arrows marked B, the other Kempe chain switch order has been followed, leading to an irrevocable Kempe chain tangle. $\square$

In the case that both orders fail, we encounter the previously defined irrevocable Kempe chain tangle and turn to Kempe–Kittell chain switches in an attempt to solve the impasse. Kempe–Kittell chains present yet another opportunity for randomization of choices. To better understand these choices, we first define the eight Kempe–Kittell chains identified by Kittell [1935].

We use exactly the notation and Kempe chain switches as suggested by Kittell [1935]. The new gadget, called *Gadget Kittell*, is invoked only when Gadget 5_2 is called upon in Algorithm Kempe and fails. For reference, see Figure 1.

Definition 5 (Gadget Kittell). (1) *Chain α*: perform an *RB*-Kempe Chain switch beginning either on v_2 or v_4.

(2) *Chain β*: perform an *RY*-Kempe Chain switch beginning either on v_2 or v_5.

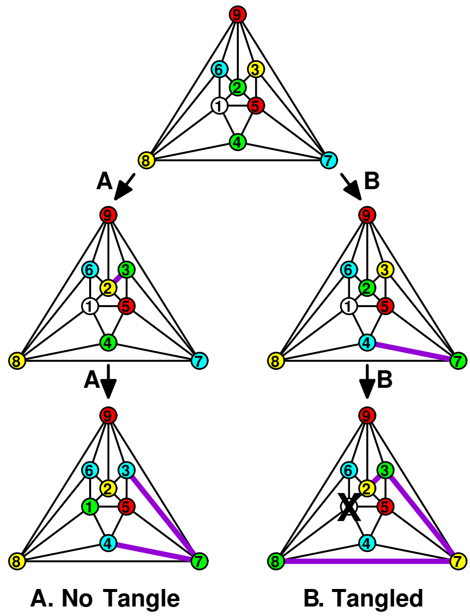


Figure 3. Gadget 5_2 in Algorithm Kempe does not commute. The thick purple lines highlight the current Kempe chain switch.

(3) *Chain* γ: perform a *GY*-Kempe Chain switch beginning either on v_1 or v_5.

(4) *Chain* δ: perform a *GB*-Kempe Chain switch beginning either on v_3 or v_4.

(5) *Chain* ϵ: perform a *BY*-Kempe Chain switch beginning either on v_4 or v_5.

(6) *Chain* ζ: perform a *GB*-Kempe Chain switch beginning either on v_1 or v_4.

(7) *Chain* η: perform a *GY*-Kempe Chain switch beginning either on v_3 or v_5.

(8) *Chain* θ: perform an *RG*-Kempe Chain switch beginning on any of v_2 or v_1.

Upon encountering an irrevocable Kempe chain tangle, we randomly choose one of the eight Kempe–Kittell chains in Gadget Kittell and continue to randomly execute switches from that list until we reach a coloration of the graph that allows us to successfully color the vertex causing the impasse or until a fixed number of Kempe–Kittell chain switches have failed (we chose an upper limit of 100 Kempe–Kittell chain switches).

Thus Algorithm Kempe is modified as follows:

Algorithm Kempe–Kittell. BEGIN

Step 1: Use Gadget relabel to label the vertices of Γ.

Step 2: For $i = n$ down to 1, attempt to color v_i in graph Γ_{i-1} as follows:

(a) if v_i can be greedily colored in graph Γ_{i-1} by Gadget greed then do so; else

(b) if degree $v_i = 4$ in Γ_i then color v_i using Gadget 4 on a randomly selected viable Kempe chain; else

(c) if degree $v_i = 5$ in Γ_{i-1} in configuration 1 then color v_i using Gadget 5_1 on a randomly selected viable Kempe chain; else

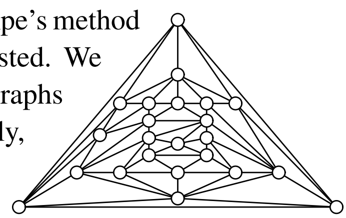
(d) if degree $v_i = 5$ in Γ_{i-1} in configuration 2 then try to color v_i using Gadget 5_2 employing two Kempe chain switches (randomly select an order in which to perform the switches, and try both orders if necessary);

(e) if an irrevocable Kempe chain tangle is reached then select a random Kempe–Kittell chain using Gadget Kittell until v_i successfully colored or 100 attempts have failed.

END

The fixed limit on the number of failures is required because it is unknown if there is always a series of Kempe–Kittell chain switches that will result in successful resolution of the impasse. The set of possible Kempe–Kittell chain switch combinations that can affect the five vertices adjacent to v_i, called the *impasse group*, is known to have a lower bound of 120 [Kittell 1935], but it is impractical to determine and check the upper bound for even a small arbitrary graph. The use of heuristics to guide the search of the impasse group has been studied for large graphs [Morgenstern and Shapiro 1991], but our interest was in determining algorithm performance when executing a purely random sequence of Kempe–Kittell chain switches to color a small graph, as this could provide an easy way to improve the performance of Algorithm Kempe for those cases.

Our randomized recursive implementation of Kempe's method always succeeded in four-coloring the graphs we tested. We ran the algorithm 500 times for each of the nine graphs tested in the original implementation and, additionally, the Kittell graph [1935], shown on the right.



We kept track of the number of times a Kempe–Kittell chain switch was required to solve an impasse (Table 2) and the vertices causing the impasse (see Table 3 on the next two pages).

	F	S	O	C	I	D	P	E	H	K
Kempe–Kittell switches (max)	9	7	0	0	0	0	6	73	6	11

Table 2. Maximum number of Kempe–Kittell chain switches required for any vertex. For the letters on the top row, see Table 1 (plus K = Kittell).

Graph	Node label	Chain switches required min	max	avg	Kittell colored nodes	Kittel use (%)
F	a	3	8	4.52	1088000	0.5996
F	b, f	3	8	4.46	1088000	0.5996
F	c	3	8	4.44	1088000	0.5996
F	d, e, h	0	0	0.00	0	0.0
F	g	3	7	4.49	1088000	0.5996
F	i	3	9	4.50	1088000	0.5996
S	a-b, d-i	0	0	0.00	0	0.0
S	c	3	7	4.45	988089	0.5446
O	a-f	0	0	0.00	0	0.0
C	a-h	0	0	0.00	0	0.0
I	a-l	0	0	0.00	0	0.0
D	a-t	0	0	0.00	0	0.0
P	a, b, g, i-o	0	0	0.00	0	0.0
P	c	1	4	2.26	526486	0.2902
P	d	2	6	2.58	399678	0.2203
P	e	2	5	2.69	323390	0.1782
P	f	2	5	2.82	339534	0.1871
P	h	2	5	2.81	474890	0.2617
E	a	35	73	47.01	4937929	2.7215
E	b	3	8	4.14	463228	0.2553
E	c	3	7	4.11	463000	0.2552
E	d	3	7	4.05	463086	0.2552
E	e	3	7	4.03	464390	0.2559
E	f	2	5	2.83	440437	0.2427
E	g	2	5	2.82	440226	0.2426
E	h	2	5	2.80	440679	0.2429
E	i	3	6	4.05	464517	0.2560
E	j	2	6	2.85	439756	0.2424
E	k	2	5	2.84	440513	0.2428
E	l	3	7	4.07	462705	0.2550
E	m	3	7	4.07	462517	0.2549
E	n	3	7	4.10	463978	0.2557
E	o	35	71	46.29	4938719	2.7220
E	p	3	7	4.07	463436	0.2554
E	q	3	8	4.06	461792	0.2545

Graph	Node label	Chain switches required min	max	avg	Kittell colored nodes	Kittel use (%)
H	a	2	6	3.22	775216	0.4273
H	b	2	6	3.29	1217733	0.6711
H	c	2	5	3.13	847294	0.4670
H	d-f, h-i, q-s, v-y	0	0	0.00	0	0.0
H	g	2	5	2.78	690371	0.3805
H	j	1	1	1.00	6058	0.0033
H	k	1	2	1.56	6037	0.0033
H	l	2	6	3.36	907479	0.5002
H	m	3	8	4.60	1405876	0.7748
H	n	3	7	3.75	828677	0.4567
H	o	2	5	3.15	813156	0.4482
H	p	2	5	3.19	1032339	0.5690
H	t	2	6	3.29	92157	0.0508
H	u	2	6	3.27	91564	0.0505
K	a	2	4	2.79	636989	0.3511
K	b	2	5	2.75	478186	0.2636
K	c,g	0	0	0.00	0	0.0
K	d	2	6	3.33	675394	0.3722
K	e	3	8	4.76	475928	0.2623
K	f	5	9	6.57	792681	0.4369
K	h	2	6	3.25	82375	0.0454
K	i	5	9	6.32	782074	0.4310
K	j	2	5	2.95	457668	0.2522
K	k	4	7	4.92	400408	0.2207
K	l	2	5	2.84	473726	0.2611
K	m	3	6	3.99	386798	0.2132
K	n	5	11	6.48	537869	0.2964
K	o	5	8	5.93	496916	0.2739
K	p	5	9	6.35	691822	0.3812
K	q	0	1	0.98	1676	0.0009
K	r	5	11	6.44	667972	0.3682
K	s	2	6	3.16	660014	0.3638
K	t	2	5	2.83	489486	0.2698
K	u	2	6	3.51	209896	0.1157
K	v	2	5	2.81	551059	0.3037
K	w	0	1	0.98	1966	0.0011

Table 3. Algorithm Kempe–Kittell results (see also top of next page).

In Table 3, the last column gives the percentage of Kittel colored nodes relative to the total number of colored nodes, which is 50 times the numbering of labelings per trial. The latter number, as already explained, was $9! = 362880$ for all graphs except those with less than 9 vertices (O with $6! = 720$ and C with $8! = 40320$).

Our fixed upper limit of 100 for Kempe–Kittell chain switches was more than sufficient since, for most impasses encountered in our tests, eleven or fewer randomly chosen Kempe–Kittell chain switches were sufficient to achieve a successful four-coloring. The exception to this was the Errera graph, which contained two vertices that required over 70 randomly chosen Kempe–Kittell chain switches on the regions marked A and B in Figure 4 to achieve successful four-coloring. These two vertices are the only two vertices in the Errera graph that do not have any neighbors of degree greater than five, and they are the polar regions of Errera's 17-country counterexample when described as a spherical map as shown in [Hutchinson and Wagon 1998, Figure 2; Wagon 2009] and as a fullerene, of molecular formula C_{30}, in our Figure 4.

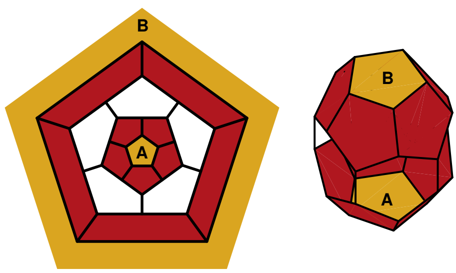


Figure 4. Errera map: planar representation and coordinatized as a fullerene (C_{30}) in $\mathbb{R}^3$.

Comparison of original and randomized implementations. We achieve proper four-coloring of all of our graphs on 100% of our runs through the inclusion of randomly selected Kempe–Kittell chain switches. In addition to this, the percentage of times that Gadget Kittell was required in our algorithm indicates the percentage of irrevocable Kempe chain tangles encountered by our randomized algorithm, which we compare in Table 4 to the failure rates from the original five groups' implementations (see also Table 3).

When we make this comparison to the average failure rate of the original five implementations, we see that our algorithm outperforms the average original algorithm's performance on the two graphs for which all $n!$ labelings were tested (Fritsch and Soifer graphs). In fact, our randomized implementation nearly matches the lowest failure rates observed among the original six implementations: 3.60% and 0.52% for Fritsch and Soifer, respectively (Table 1 and Table 4).

On the Errera and Heawood graphs, randomization results in a higher rate of irrevocable Kempe chain tangles than the average rate of the original algorithm,

Graph	Original algorithm Min % Failure	Original algorithm Max % Failure	Original algorithm Avg % Failure	Randomized algorithm % Failure
Fritsch	3.598	14.131	11.790	3.60
Soifer	0.520	1.859	1.461	0.55
Poussin	0.014	0.627	0.228	1.18
Errera	3.350	10.866	7.881	9.28
Heawood	0.372	7.153	2.439	4.83

Table 4. Comparison of original Algorithm Kempe to randomized version with Kempe–Kittell chains.

but it is still within the range of the minimum and maximum failure rates of the original implementations. On the asymmetrical Poussin graph, our algorithm results in significantly more Kempe chain tangles than the average (Table 4), but this is mitigated by the success of the Kempe–Kittell chain switches in coloring the graph. We exclude the graphs of the platonic solids, as they cause no failures for either algorithm.

4. Conclusions

We evaluated the performance of the version of Algorithm Kempe in [Gethner and Springer 2003] and its performance after the addition of Kempe–Kittell chain switches, which successfully overcame all irrevocable Kempe chain tangles in our benchmark graphs. We have proven that the order in which Kempe chain switches are performed affects the outcome of the algorithm and have shown that the application of randomization to the selection of Kempe and Kempe–Kittell chain switches in this algorithm is a useful method for making the choice of which switch to perform first. In 500 test runs on each of ten benchmark graphs, the use of randomized chain choices resulted in successful four-coloring of the graphs with fewer than 12 random choices for any vertex most of the time. When compared to the original algorithm, there appears to be a performance trade-off in that randomization causes the use of Gadget Kittell in the Poussin graph more often than would have been required by the nonrandomized version. Finally, we discovered that some vertices appear to be more likely to cause irrevocable Kempe chain tangles than others, and we identify those vertices in the hopes of being able to characterize them.

Open questions for future work.

1. Given a plane graph Γ with e edges and n vertices, what is the minimum value of e for which Kempe's method is probably guaranteed to succeed in properly coloring Γ?

2. What percentage of all plane graphs on nine vertices serve as counterexamples to Kempe's method?
3. Do the vertices that cause irrevocable Kempe chain tangles or require high numbers of Kempe–Kittell chain switches share properties which can be exploited to improve Algorithm Kempe–Kittell?
4. Let Γ be a plane graph on n vertices. It follows from [West 2001, Exercise 6.1.9] that, when $n \leq 11$, there is some labeling for which Algorithm Kempe succeeds in properly four-coloring Γ. What is the smallest value of $n > 11$ for which Algorithm Kempe–Kittell is provably guaranteed to succeed?
5. It is not difficult to show that the Icosahedral graph will be properly four-colored by Kempe's algorithm regardless of the labeling of the vertices (and this is confirmed by Table 3). Characterize all plane graphs that will be properly four-colored by Kempe's algorithm under all possible orderings of the vertices. Short of that potentially difficult goal, find interesting families of plane graphs (with at least 11 vertices and whose average vertex degree is at least 5) for which Kempe's algorithm will always succeed.

5. Addendum

Stan Wagon (personal communication, 2008) reports the discovery that the plane graph corresponding to the contiguous 48 United States plus Lake Michigan and the oceanic waters admits a labeling that leads to a Kempe impasse at the great state of Illinois. Details will appear in [Wagon 2009].

References

[Appel and Haken 1976/77] K. Appel and W. Haken, "Every planar map is four colorable", *J. Recreational Math.* **9**:3 (1976/77), 161–169. MR 58 #27598f Zbl 0357.05043

[Appel and Haken 1977] K. Appel and W. Haken, "The solution of the four-color-map problem", *Sci. Amer.* **237**:4 (1977), 108–121, 152. MR 58 #27598e

[Appel et al. 1977] K. Appel, W. Haken, and J. Koch, "Every planar map is four colorable", *Illinois J. Math.* **21**:3 (1977), 429–567 and microfiche supplement. MR 58 #27598b Zbl 0387.05010

[Archuleta and Shapiro 1986] R. A. Archuleta and H. D. Shapiro, "A fast probabilistic algorithm for four-coloring large planar graphs", pp. 595–600 in *Proc. ACM Fall joint computer conference* (Dallas, 1986), IEEE Press, Los Alamitos, CA, 1986.

[Biggs et al. 1986] N. L. Biggs, E. K. Lloyd, and R. J. Wilson, *Graph theory. 1736–1936*, 2nd ed., Oxford University Press, New York, 1986. MR 88e:01035 Zbl 0595.05003

[Errera 1921] A. Errera, *Du colorage des cartes et de quelques questions dánalysis situs*, Ph.D. thesis, Université Livre de Bruxelles, 1921.

[Fritsch and Fritsch 1998] R. Fritsch and G. Fritsch, *The Four-Color Theorem*, Springer, New York, 1998. MR 99i:05079

[Gethner and Springer 2003] E. Gethner and W. M. Springer, II, "How false is Kempe's proof of the four color theorem?", *Congr. Numer.* **164** (2003), 159–175. MR 2005a:05084 Zbl 1050.05049

[Heawood 1890] P. J. Heawood, "Map colour theorem", *Quart. J. Pure Appl. Math.* **24** (1890), 332–338.

[Hutchinson and Wagon 1998] J. Hutchinson and S. Wagon, "Kempe revisited", *Amer. Math. Monthly* **105**:2 (1998), 170–174. MR 1605875 Zbl 0915.05058

[Kempe 1879] A. B. Kempe, "On the geographical problem of the four colours", *Amer. J. Math.* **2**:3 (1879), 193–200. MR 1505218

[Kittell 1935] I. Kittell, "A group of operations on a partially colored map", *Bull. Amer. Math. Soc.* **41**:6 (1935), 407–413. MR 1563103 Zbl 0012.03501

[Morgenstern and Shapiro 1991] C. A. Morgenstern and H. D. Shapiro, "Heuristics for rapidly four-coloring large planar graphs", *Algorithmica* **6**:6 (1991), 869–891. MR 92g:05169

[Ore 1967] O. Ore, *The four-color problem*, Pure and Applied Mathematics **27**, Academic Press, New York, 1967. MR 36 #74 Zbl 0149.21101

[Robertson et al. 1996] N. Robertson, D. P. Sanders, P. Seymour, and R. Thomas, "A new proof of the four-colour theorem", *Electron. Res. Announc. Amer. Math. Soc.* **2**:1 (1996), 17–25. MR 97f:05070 Zbl 0865.05039

[Robertson et al. 1997] N. Robertson, D. Sanders, P. Seymour, and R. Thomas, "The four-colour theorem", *J. Combin. Theory Ser. B* **70**:1 (1997), 2–44. MR 98c:05065 Zbl 0883.05056

[Soifer 1997] A. Soifer, "Map coloring in the victorian age: problems and history", *Mathematics Competitions* **10** (1997), 20–31.

[Wagon 2002] S. Wagon, "A machine resolution of a four-color hoax", pp. 181–193 in *Proc. 14th Canad. Conf. Comput. Geom.*, 2002.

[Wagon 2009] S. Wagon, *Mathematica in Action*, 3rd ed., Springer/TELOS, New York, 2009. To appear.

[West 2001] D. B. West, *Introduction to graph theory*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 2001. Zbl 0845.05001

[Wilson 2002a] R. Wilson, *Four colors suffice: How the map problem was solved*, Princeton University Press, Princeton, NJ, 2002. MR 2004a:05060

[Wilson 2002b] R. A. Wilson, *Graphs, colourings and the four-colour theorem*, Oxford University Press, Oxford, 2002. MR 2003c:05095 Zbl 1007.05002

Received: 2008-11-23 Accepted: 2008-11-23

ellen.gethner@cudenver.edu *University of Colorado Denver, Department of Mathematics and Department of Computer Science, Denver, CO 80217, United States*
http://carbon.cudenver.edu/~egethner/

bkallichanda@gmail.com *eCollege, 4900 S. Monaco Street Suite 200, Denver, CO 80237, United States*
http://www.linkedin.com/pub/bopanna-kallichanda/4/145/b3a

alexander.mentis@us.army.mil *United States Military Academy, Department of Electrical Engineering and Computer Science, PO Box 67, Fort Montgomery, NY 10922, United States*
http://www.eecs.usma.edu/webs/people/Mentis/

s_braudrick@msn.com	*Lockheed Martin, 12999 W Deer Creek Canyon Rd, Littleton, CO 80127-5146, United States* http://www.linkedin.com/pub/sarah-braudrick/7/22/793
sumeet.chawla@gmail.com	*6885 West 91st Court, Apt #23-301, Westminster, CO 80021, United States* http://ouray.cudenver.edu/~schawla/
andy.clune@gmail.com	*Syntellect, 16610 North Black Canyon Highway Suite 100, Phoenix, AZ 85053, United States* http://www.linkedin.com/pub/andy-clune/4/11/204
rachel.drummond@gmail.com	*OPX Biotechnologies, Inc., 2425 55th Street Suite 100, Boulder, CO 80301, United States* http://www.blogger.com/profile/11006794721285468833
pennye_gr@hotmail.com	*Interactive Intelligence, 7601 Interactive Way, Indianapolis, IN 46278, United States* http://www.linkedin.com/pub/panagiota-evans/12/b01/144
wrroche1@msn.com	*1625 S Iola Street #202, Aurora, CO 80012, United States* http://www.big-baboon.com/
dumnas@comcast.net	*Aurora Loan Services, 10350 Park Meadows Drive, Littleton, CO 80124, United States* http://www.linkedin.com/pub/nao-takano/3/964/24