Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 21, no. 4, pp. 649–661 (2017)
DOI: 10.7155/jgaa.00433

# $D$-resolvability of vertices in planar graphs

*James A. Tilley*

Retired, Unaffiliated

**Abstract**

It is well known that a vertex of degree 5 in a planar graph is not *D-reducible*. However, it remains an open question whether all vertices in such a graph are *D-resolvable*. If so, it would be a stronger result than mere 4-colorability. To investigate the question, we designed and implemented two algorithms, one to 4-color a planar graph and the other to generate internally 6-connected triangulations randomly. The coloring algorithm greedily obtains a $k$-coloring and then, if $k > 4$ (highly likely), transforms the $k$-coloring into a 4-coloring solely by means of Kempe exchanges (generally possible only if all vertices are $D$-resolvable). We used the random generator to test the coloring algorithm systematically in over 200,000 trials that included many tens of thousands of non-isomorphic graphs. We also tested the algorithm on the graphs of Errera, Kittell, Heawood and others. We encountered no failures in any of the tests.

| Submitted: | Accepted: | Final: | Published: |
|---|---|---|---|
| March 2017 | April 2017 | April 2017 | June 2017 |

| Article type: | Communicated by: |
|---|---|
| Regular paper | G. Liotta |

*E-mail address:* jimtilley@optonline.net (James A. Tilley)

## Terminology and background

In this article, we consider vertex-colorings of planar graphs and use the integers 1, 2, 3, 4, ... to indicate color labels for the vertices. We define a *proper* coloring of such a graph $G$ to be a partition of its vertex set $V(G)$ into independent sets, each set assigned a distinct color. We consider proper colorings only and refer to them merely as colorings. An *independent set* of vertices in $G$ is a set in which no two vertices are adjacent. If there are $k$ such sets in a particular partition, that partition represents a $k$-coloring of $G$ and, by convention, it uses consecutive color labels 1, ..., $k$. Two colorings of $G$ are considered identical if they partition $V(G)$ into identical independent sets. Thus, two colorings of $G$ are not considered distinct if they differ merely by a permutation of colors. We refer to a coloring of $G$ as a *state* of $G$. In a given state of $G$, a $j$-$k$ *Kempe chain* ($j \neq k$) is a maximal connected subgraph induced by vertices colored either $j$ or $k$. There can be several distinct, mutually disconnected $j$-$k$ Kempe chains. A $j$-$k$ *Kempe exchange* ($j \neq k$) in a coloring of $G$ is the interchange of colors $j$ and $k$ on a non-empty proper subset of all $j$-$k$ chains in $G$. If an $i$-$j$ Kempe chain and a $j$-$k$ Kempe chain ($i \neq k$) intersect at one or more vertices colored $j$, then the two chains are said to be *entangled*.

Our coloring algorithm relies on the concept of a *state transformation graph*. We generalize the definition from that used in [8] by admitting the possibility of colorings of a planar graph $G$ in which a subset $V_{\leq 4}$ of $V(G)$ uses colors in the color set {1,2,3,4} while its complement $V_{>4}$ uses colors in the color set {5,6, ...}. The state transformation graph $H_G$ associated with $G$ is a graph of all distinct states of $G$ in which the colors of vertices in $V_{>4}$ are fixed, but the colors of vertices in $V_{\leq 4}$ are not. Each vertex in $H_G$ represents such a state. Two states are considered adjacent if a Kempe exchange involving a pair of colors from the color set {1,2,3,4} transforms one into the other. Because Kempe exchanges induce an *equivalence relation* on the states of $G$, each component of $H_G$ is an *equivalence class* under Kempe exchanges and all states in a component of $H_G$ are said to be *Kempe-equivalent*.

Let $G$ be a planar graph and $G_v$ the graph obtained from $G$ by deleting an arbitrary vertex $v$ and all its incident edges. Let $R$ be the ring of vertices (cycle) adjacent to $v$ in $G$. We distinguish two types of state in $H_{G_v}$: (i) *initial state*: $R$ uses all colors in the color set {1,2,3,4} and (ii) *solution state*: $R$ uses only two or three colors from the color set {1,2,3,4}. These terms are logical: for $G$ to be 4-colorable, we must be able to replace $v$ in $G_v$ with color 1, 2, 3, or 4 not used in $R$; thus, given an initial state of $H_{G_v}$, we must be able to transform it into a solution state. If a component of $H_{G_v}$ consists only of initial states, then each state in that component is said to be *difficult* because it cannot be transformed by a sequence of Kempe exchanges into a solution state. The conjecture that the coloring algorithm we present in section 2 works for all planar graphs is equivalent to the statement that, for any such $G$ and any vertex $v$ in $G$, the state transformation graph $H_{G_v}$ has no difficult initial states.

By analogy to the notion of *D-reducibility* (see [15]), we say that a vertex $v$ in a planar graph $G$ is *D-resolvable* if any 4-coloring of $G_v$ can be transformed, as necessary, *by Kempe exchanges alone*, into a 4-coloring of $G$. To prove $D$-reducibility, we are constrained to resolve a deleted vertex $v$ by interchanging colors only on Kempe chains that include at least one vertex in the ring $R$ adjacent to $v$, whereas, in the coloring algorithm we present in section 2, we are permitted to interchange colors on chains that involve only vertices not in $R$, perhaps at a considerable remove from $R$. $D$-reducibility is independent of the structure of $G$ exterior to $R$ whereas $D$-resolvability might not be. In a sense, it is a question of "local" versus "non-local" effects. That is why a vertex of degree 5 turns out to be $D$-resolvable even though it is not $D$-reducible. But it is a one-way street: $D$-reducibility implies $D$-resolvability, not vice versa. Vertices of degree 3 and 4, trivially $D$-reducible, are $D$-resolvable. The conjecture that the state transformation graph $H_{G_v}$ has no difficult initial states is equivalent to the statement: *Every vertex in a planar graph is $D$-resolvable.* This statement is stronger than the 4-color theorem; the conjecture clearly implies the 4-color theorem, but the converse is not true.

# 1 Introduction and overview

Proving the 4-color conjecture [1, 2, 7] established that any planar graph can be 4-colored but did not specify how to achieve such a 4-coloring. In a paper describing their own work on this coloring problem, Hutchinson and Wagon [4] cited several papers that investigate the use interchanges of colors on Kempe chains to 4-color an arbitrary planar graph. The Hutchinson-Wagon coloring algorithm, in particular, deals with the matter of entangled Kempe chains by introducing various random elements. The goal of our research was to design and implement a coloring algorithm that does not rely on any random feature. In section 2, we present a deterministic coloring algorithm; in the first part of section 4, we report the results of tests that use the algorithm to establish the $D$-resolvability of *all vertices* in the well-known graphs of Errera [9], Fritsch [10], Heawood [11], Kittell [12], de la Vallée-Poussin [13], and Soifer [14].

Our coloring algorithm never appeals to a random choice of color, vertex, or exchange. Its underpinning, the concept that enables a deterministic approach, is the state transformation graph. We visualize the process of coloring a planar graph as one of first finding a starting initial state by means of a greedy algorithm that partitions the vertex set into independent sets and then navigating methodically through the state transformation graph from the starting state to a nearest solution state by means of Kempe exchanges. Entanglements, it seems, can prolong, but not prevent, finding a path to a solution state.

Our code is written in APL, an interpretive language highly useful for mathematical computations, but less efficient than a compiled language. The computer programs were run on a MacBook Pro with an Intel i7 processor. With that computing power, the coloring algorithm is practical for graphs up to order 75. Beyond that, we could assure practicality only by improving the efficiency of the algorithm and, to that end, we introduced a random element into the process. Interestingly, that random element is one of several used by Hutchinson and Wagon [4]—how to number the vertices of a graph of order $n$. Like them, we use the notion of a *shuffle* of the vertex numbering, a random permutation of the labels 1 to $n$, but unlike them, we do not need to use it in the basic version of our coloring algorithm, only in the modified version, and then, only for purposes of moderating the long tail in the distribution of execution times involved in finding a solution. This modification to the basic coloring algorithm is the subject of section 3.

The results of extensive tests of the modified coloring algorithm are reported in the second and third parts of section 4. For those tests, we employed an algorithm to generate random internally 6-connected triangulations; a qualitative description of the algorithm can be found in appendix A. From the results for higher-order graphs, we conclude that, with our computing resources, the graph-coloring algorithm can be rendered practical for graphs up to order 125. With machines a few hundred times faster, the algorithm would be practical for graphs up to order 200. The limitation is driven primarily by the number of Kempe chains encountered as the process moves systematically from a starting initial state in the state transformation graph to a nearest solution state.

## 2 The basic coloring algorithm

Our coloring algorithm has two parts: (a) the use of a greedy algorithm to establish a starting $k$-coloring $\chi$ for the planar graph $G$ to be colored, and (b) if $k > 4$, the systematic use of Kempe exchanges starting from the initial state $\chi$ to find its nearest solution state $\chi'$. Let $G$ have order $n$. We begin by numbering its vertices 1 to $n$ and then specify the $n \times n$ adjacency matrix $A_G$ for $G$.

Part (a) starts with $v_1$, the vertex numbered 1, and uses $A_G$ to find the next-lowest-numbered vertex not adjacent to $v_1$ and then the next-lowest-numbered vertex not adjacent to the previous two, and so on, until an independent set has been established. The process continues by selecting the lowest-numbered vertex in $G$ not in the first independent set and finding the next-lowest-numbered vertex not adjacent to it that is not in the first independent set, and then the next-lowest-numbered vertex not adjacent to the previous two and not in the first independent set, and so on, until the second independent set has been established. The process ends when a partition $P_{v_1}$ of $V(G)$ has been created. This entire process is repeated starting with the vertex $v_2$ and ending with a partition $P_{v_2}$, always adding non-adjacent vertices from the set of vertices not already assigned to an independent set and, among those, always selecting the lowest-numbered vertex. When the set of $n$ partitions $P_{v_j}$, $1 \leq j \leq n$, has been created, each a $k$-coloring of $G$ for some value of $k$, we select only those partitions having the smallest $k$, and then, from those, we select the lowest-numbered partition that has the least number of vertices in its smallest independent set. In that partition, we list the independent sets in non-increasing order by cardinality and assign the colors 1, 2, 3, 4, ... to the partitions so ordered. This establishes the starting $k$-coloring $\chi$ for $G$. If $k \leq 4$, we are finished. If not, we continue to part (b). Whether part (b) is needed might depend on how we number the vertices in $G$.

If we arrive at part (b), we know that there are one or more vertices assigned colors $i > 4$. We relabel such vertices $w_j$, with $1 \leq j \leq m < n$. Part (b) starts with $w_1$ and deletes it from $G$ to obtain the planar graph $G_{w_1}$. If possible, we recolor $w_1$ to 1, 2, 3, or 4. If not, then we know that $\chi$ is an initial state. We determine all possible Kempe exchanges for each of the six color-pairs 1-2, 1-3, 1-4, 2-3, 2-4, and 3-4 in order to locate $\chi$'s neighbors lying a distance 1 away in the state transformation graph for $G_{w_1}$. If no solution state (in the sense of being able to replace $w_1$ with color 1, 2, 3, or 4) exists among those neighbors, we locate all states lying a distance 2 away from $\chi$ and, if no solution state exists among those, we continue to states lying a distance 3 away, and so on, until a solution state $\chi'$ is found. Our experience has been that it is rare to have to proceed beyond distance 3 from $\chi$. In $\chi'$, the vertex $w_1$ has the color 1, 2, 3, or 4. We continue by using $\chi'$ as the starting coloring for $G_{w_2}$ (in which we have deleted $w_2$ from $G$) and, as necessary, we expand our search in the network of states in the vicinity of $\chi'$ until we find a solution state $\chi''$ in which $w_2$ can be replaced with color 1, 2, 3, or 4. We continue in this fashion for $w_3$, $w_4$, $\ldots$, $w_m$ until all the $w$ vertices have been resolved and $G$ is 4-colored.

# 3 Modification of the coloring algorithm

How long it takes the computer to find a solution state in part (b) of the algorithm very much depends on which initial state it starts from in the state transformation graph. A nearest solution state might lie a distance 1 away from a given starting state, or distance 2, or distance 3, .... Moreover, one might find a solution state lying at distance 1 by processing only 1-2 exchanges, or one might have to proceed in succession through 1-3, 1-4, 2-3, 2-4, and 3-4 exchanges before a solution is found, and then, if none has been found, have to repeat the process of analyzing Kempe exchanges commencing at initial states lying a distance 1 from the starting state, and so on. Thus, for a given graph, over many different starting states there will be a distribution of execution times to find a solution. While the coloring algorithm is deterministic in its specification, a probability distribution of execution times results when it is launched from different starting states. For a higher-order graph, it may take a short time or a long time to find a solution state. A priori we don't know what starting state part (a) of the algorithm will provide and, even if we did, we wouldn't know for that starting state whether or not a solution state lies close by. For low- and medium-order graphs, this distribution of execution times to find a solution state is inconsequential from a practical viewpoint because the execution times are all very small. But for a higher-order graph, the distribution of execution times matters, particularly because the distribution happens to be long-tailed.

In navigating the state transformation graph from a starting initial state to the states lying a distance 1 away and then, if and as necessary, to the states lying a distance 2 away, 3 away, and so on, the program may have to exchange colors on many different combinations of Kempe chains. The number of those combinations increases as the order of the graph increases. In a given coloring state $\chi$, suppose that the number of $j$-$k$ Kempe chains present is $m_{jk}$. Because a single Kempe exchange involves selecting a non-empty proper subset of those $m_{jk}$ chains and because selecting no chains or all chains does not lead to a distinct coloring state and because the set of partitions of distinct chains into those included in a given exchange and those not included ends up double-counting the number of distinct states attainable, we see that there are $\frac{1}{2}[2^{m_{jk}} - 2]$ distinct states lying a distance 1 away from $\chi$ that can be reached by a single Kempe exchange involving the color-pair $j$-$k$. The total number of coloring states lying a distance 1 away from $\chi$ is the sum of those terms over all $j < k$. Thus, execution times can be expected to grow at least exponentially as the number of chains increases, rendering the algorithm impractical for graphs of high order.

There is a partial remedy to the problem of long tails in the distribution of execution times. Suppose we implement a *stopping rule* that halts the processing of Kempe chains whenever the computer program encounters a number of chains for a color-pair that exceeds a specified limit or whenever the elapsed time spent analyzing Kempe chains exceeds a specified limit. We could then return to the beginning and proceed with a different starting state that might resolve more quickly than the one abandoned. Because the starting state is determined by way of the greedy algorithm—part (a) of the overall coloring algorithm—we are highly likely to obtain a different starting state if we begin with a different numbering of the vertices of the graph, a random permutation of the numbers 1 to $n$, which we call a *shuffle*. We use this observation to make a modest modification to the coloring algorithm described in section 2. For purposes of reducing the long tail, we introduce a random element into the procedure by launching the coloring algorithm with a random shuffle of the vertex numbering. We then apply part (a) precisely as described in section 2 and proceed to part (b) as described in section 2, but stop whenever the number of chains for a color-pair exceeds a stated limit or whenever the aggregate time spent processing Kempe chains exceeds a stated limit, thus forcing a return to the beginning and another shuffle. We continue in this manner until a solution state is attained.

The reason that this type of modification can succeed in reducing the long tail in the distribution of execution times is that, for large $n$, part (a) tends to execute more rapidly than part (b) by orders of magnitude. Shuffling vertex numbering labels consumes virtually no CPU time and the greedy algorithm remains relatively fast even for large $n$. A practical implementation of the stopping rule involves setting the limits low enough to prevent situations when the computer would otherwise bog down analyzing a large number of Kempe exchanges but not so low that a solution state can never be attained. While the modified algorithm does not eliminate the long tail in the distribution of execution times, when tuned properly by setting the stopping limits to appropriate values, it can reduce that tail considerably. With our computing resources, this modification was able to render the coloring algorithm practical for $n$ up to 125.

Although imposing limits on the number of Kempe chains of any color-pair and on the aggregate execution time spent processing Kempe chains is not necessary for low- and medium-order graphs for which the mean execution times are naturally low anyway, it does not hurt to include them for those graphs and thus adopt the modified version of the coloring algorithm as the default algorithm. That is what we have done for most of our tests. In all situations, we use mean execution time for parts (a) and (b) combined as a measure of efficiency of the algorithm. Admittedly, this is an oversimplification, because the dispersion of the distribution of execution times is important as well. We study these features in the second and third parts of section 4.

## 4 Tests of the coloring algorithm

We performed three sets of tests. The first involved graphs devised to illustrate the flaw in Kempe's attempted proof of the 4-color conjecture [5]. For these graphs, by using the coloring algorithm precisely as described in section 2, without the modification described in section 3, we showed that Kempe exchanges always succeed in achieving a proper 4-coloring. The second set of tests involved randomly generated triangulations of low and medium order. (Refer to appendix A for a description of the random-graph generator.) The modified coloring algorithm succeeded in all cases examined. The third set of tests extended the investigations of the second set to triangulations of higher order with the principal purpose of assessing the efficiency of the coloring algorithm as a function of graph order.

*Graphs well-known in the history of the 4-color problem*

For the well-known graphs of Errera [9], Fritsch [10], Heawood [11], Kittell [12], de la Vallée-Poussin [13], and Soifer [14], we have performed the most complete tests. For each graph, we deleted a designated vertex (considering it to have been colored 5) and computed the entire state transformation graph for the resulting near-triangulation—all possible 4-colorings—to demonstrate that there are no difficult initial states. (The number of initial states in a state transformation graph $H_{G_v}$ depends on which vertex $v$ is deleted in $G$, but the number of solution states in $H_{G_v}$ is independent of $v$: for the Errera, Kittell, and Heawood graphs, the numbers of solution states are 40, 212, and 256, respectively.) We then repeated this exercise for each vertex in the graph. Thus, given a 5-coloring of any of these graphs in which only a single vertex is colored 5, we are assured that for any initial state there is a Kempe-equivalent solution state. We can summarize our findings by stating that for any of these graphs, every Kempe chain entanglement encountered can be worked through with ease.

*Internally 6-connected triangulations of low and medium order*

We focused the rest of our tests on internally 6-connected triangulations. An *internally 6-connected* triangulation is one that has minimum vertex degree 5 and in which there is no *separating cycle* of length 5 or less for which there are two or more vertices both inside and outside the cycle. Our algorithm can be applied to any planar graph, but because vertices of degree 3 and 4 are trivially $D$-reducible (and therefore $D$-resolvable), graphs that contain them are of lesser interest when exploring the question of whether Kempe exchanges can be used to 4-color a graph. Internally 6-connected triangulations were our primary concern because any minimal counterexample to the 4-color conjecture that might have existed would have had to be internally 6-connected [3].

The smallest internally 6-connected triangulation has order 12. It is the icosahedron. Our initial goal was to demonstrate that the coloring algorithm succeeds for every possible internally 6-connected triangulation up to and including order $n = 25$. For the lowest orders, that task is easy. There is only a single isomorphism class for each of $n = 12$, 14, and 15, and only three for order $n = 16$. There is no internally 6-connected triangulation whatsoever of order $n = 13$. However, enumerating all isomorphism classes accurately for a given $n$ is tedious at best and the number of isomorphism classes begins to explode for $n > 16$. Instead, we settled for a lesser goal of sampling the full set of isomorphism classes by applying the coloring algorithm to 10,000 random internally 6-connected triangulations for each $n$ from 15 to 25 inclusive. The aggregate number of trials so processed obviously overstates the strength of the test because a very high proportion of the randomly generated triangulations will belong to the same isomorphism class. Still, there is a benefit in conducting an experiment that repeats the 4-coloring exercise for the same graph several times—it tests the validity of the $D$-resolvability conjecture for different vertices in that graph. We found no failures of the coloring algorithm across this set of 110,000 trials. Nor were there any when the algorithm was tested on 10,000 random internally 6-connected triangulations each for orders 30, 35, 40, 45, 50, 55, 60, 65, 70, and 75; for this collection of graphs, we established a lower bound on the number of non-isomorphic graphs. For each graph, we catalogued the set of its vertex degrees. Two graphs cannot be isomorphic if their sets of vertex degrees differ. Using this approach, we determined the lower bounds for the 10 samples of 10,000 graphs each; they were 122, 336, 833, 1803, 3352, 5312, 6785, 8016, 8664, and 9156, respectively, for an aggregate lower bound of 44,379 non-isomorphic graphs in the sample of 100,000. This is a very weak lower bound because we know that two planar graphs with identical sets of vertex degrees can be non-isomorphic by reason of differing adjacencies. Nevertheless, we can state with confidence that among all the tests we performed there were many tens of thousands of non-isomorphic graphs.

All tests for these various internally 6-connected triangulations used the modified coloring algorithm with a limit of 6 for the maximum number of Kempe chains to be tolerated for any color-pair and a limit of 1 second for the aggregate execution time to be tolerated in processing Kempe chains within a given shuffle before being forced to make a new shuffle in part (a) as a prelude to commencing part (b) anew. The mean execution times, measured in seconds, to find a nearest solution state in 10,000 trials each for orders 30, 35, 40, 45, 50, 55, 60, 65, 70, and 75 were 0.030, 0.055, 0.086, 0.150, 0.232, 0.357, 0.516, 0.826, 1.433, and 2.518, respectively. The ratios of successive mean execution times for these various orders are 1.8, 1.6, 1.7, 1.5, 1.5, 1.4, 1.6, 1.7, and 1.8. From this result we get a clear sense that the growth in mean execution time is not less than exponential in the order of the graph, a conclusion supported by the results for the third set of tests we conducted.

*Specific randomly generated graphs of orders* $50, 75, 100,$ *and* $125$

The third set of tests involved two experiments. In the first experiment, we generated a random internally 6-connected triangulation of order 50, one of order 75, one of order 100, and one of order 125. For each of those four graphs we applied the modified version of the coloring algorithm 100 times in order to estimate the mean and median of the frequency distribution of execution times required to find a solution state. In the second experiment, after fixing the order at 100, we generated 1000 different random triangulations and applied the modified version of the coloring algorithm once to each random graph, thus producing a different kind of frequency distribution of execution times, essentially a "cross-sectional" distribution across graphs of order 100. The results of the two experiments are recorded in table 1. The four consecutive columns titled order 50, order 75, order 100, and order 125 relate to the first experiment and the last column, titled order 100, relates to the second experiment. In the first experiment, the coloring problem for each graph is solved 100 times; in the second experiment, the coloring problem for each of 1000 graphs is solved only once per graph. A better estimate of the cross-sectional distribution would use not only a large sample of graphs but also a large number of starting colorings for each graph in the sample.

There are other statistics of note not reported in table 1. In applying the greedy algorithm in part (a) to the four triangulations, there were never more than six colors in the starting coloring, most often only five. For the graph of order 50, the number of vertices needing resolution ranged from 0 to 8 across the various shuffles; for order 75, from 1 to 13; for order 100, from 2 to 17; and for order 125, from 4 to 21. Resolving the vertices colored 5 and 6 was the task of part (b) of the coloring algorithm. The per-shuffle limits on the number of Kempe chains and the Kempe chain aggregate execution time were operative in virtually all trials for the highest-order graphs. For the random graph of order 50, the number of shuffles to find a solution ranged between 1 and 3; for order 75, that range was from 1 to 16; for order 100, from 1 to 69; and for order 125, from 1 to 988. With only the computing power of our laptop available, it is clear from table 1 that applying the algorithm to graphs of order greater than 125 would frequently result in impractically long execution times. It is likely that the efficiency of the coloring algorithm can be further improved by tuning the two stopping limits—number of Kempe chains and aggregate time spent processing Kempe exchanges—optimally, but it is difficult to see how the long tail can be reined in sufficiently that graphs of significantly higher order than 125 can be handled without substantially more computing power.

Before leaving this section 4, we emphasize that our conjecture regarding $D$-resolvability applies to any planar triangulation, regardless of order or connectiv-

| | order 50 | order 75 | order 100 | order 125 | order 100 |
|---|---|---|---|---|---|
| time interval | | | | | |
| $0-1$ | 93 | 51 | 7 | 0 | 54 |
| $1-2$ | 2 | 12 | 6 | 0 | 90 |
| $2-4$ | 5 | 12 | 11 | 0 | 137 |
| $4-8$ | 0 | 18 | 14 | 1 | 174 |
| $8-16$ | 0 | 6 | 18 | 2 | 189 |
| $16-32$ | 0 | 1 | 23 | 6 | 218 |
| $32-64$ | 0 | 0 | 15 | 13 | 112 |
| $64-128$ | 0 | 0 | 6 | 28 | 25 |
| $128-256$ | 0 | 0 | 0 | 25 | 1 |
| $256-512$ | 0 | 0 | 0 | 18 | 0 |
| $512-1024$ | 0 | 0 | 0 | 6 | 0 |
| $1024-2048$ | 0 | 0 | 0 | 1 | 0 |
| mean time | 0.341 | 2.557 | 19.876 | 192.625 | 15.965 |
| median time | 0.152 | 0.923 | 14.462 | 127.604 | 10.013 |
| stopping limits | | | | | |
| number of chains | 7 | 8 | 9 | 10 | 9 |
| chain execution time | 2 | 4 | 8 | 16 | 8 |

Table 1: The results of tests using randomly generated internally 6-connected triangulations. The four consecutive columns titled order 50, order 75, order 100, and order 125 apply to a single randomly generated graph of each order; in those columns, the frequency histograms apply to 100 trials using the modified version of the coloring algorithm. All execution time intervals and the stopping limit on chain execution time are measured in seconds. The stopping limit on the number of Kempe chains applies to each color-pair separately. The last column in the table applies to a single trial of the modified version of the coloring algorithm for each of 1000 randomly generated triangulations of order 100.

ity. That includes *akempic* triangulations [6] for which there is a 4-coloring that has no Kempe-equivalent 4-coloring. The icosahedron is perhaps the best known graph in the class of akempic triangulations—it has 10 distinct 4-colorings, none of which is Kempe equivalent to any of the others. Despite this feature, the conjecture asserts that every vertex in an akempic triangulation is $D$-resolvable, that deleting any vertex in such a triangulation yields a near-triangulation for which all initial states have Kempe-equivalent solution states, a statement easily shown to be true for the icosahedron.

## 5 Conclusion

We have presented an algorithm for 4-coloring a planar graph that does not involve or rely on any feature of a random nature in its basic implementation. It is driven by the concept of a state transformation graph, which is based on the notion of a Kempe exchange—the exchange of colors on a non-empty proper subset of all individual Kempe chains of a specified color-pair. We have tested our algorithm thoroughly: first, as a basic proof of concept, on graphs well-known in the history of the 4-color problem—graphs specifically designed to defeat 4-coloring by means of Kempe exchanges—and then, in hundreds of thousands of trials that included many tens of thousands of non-isomorphic randomly generated internally 6-connected triangulations of order up to 125. No failures of the algorithm occurred.

We contend that our tests are sufficient to permit us to hypothesize with confidence that every vertex in an internally 6-connected planar triangulation is $D$-resolvable; in other words, that Kempe exchanges will always succeed in transforming a $k$-coloring ($k > 4$) of such a graph into a 4-coloring. If that is true, it is likely that the conjecture is true for every vertex in an arbitrary planar triangulation because we know that all vertices of degree 3 and 4 are $D$-resolvable, and if true for every planar triangulation, then it would also be true for every planar graph. However, from a practical viewpoint, applicability of the algorithm is limited by the computing power available. For our current computing resources, that limit was graphs of order 125, and then only because we implemented a modification to the basic algorithm that tightened the distribution of execution times for finding a solution.

## Appendix A

To test the coloring algorithm presented in this article thoroughly, it is essential to have a large supply of non-isomorphic graphs with specified properties. It is feasible to accomplish that only through the random generation of such graphs.

Our random graph generator builds an internally 6-connected triangulation of a specified order $n$ by starting with a graph of order 6—an interior vertex of degree 5 joined to the vertices in its boundary ring—and then successively adding vertices and edges in the exterior region. The permitted actions at each stage of the build are: (1) joining two non-adjacent vertices on the boundary ring to create a triangle, (2) adding an exterior vertex and joining it to two or more consecutive vertices on the boundary ring, and (3) for $p \geq 5$, adding a $p$-gon in the exterior region using a single edge on the boundary ring as its base. The $p$-gon includes an interior vertex of degree $p$.

We have programmed the generator so that during any stage the additions are 0, 1, or 2 exterior edges and either an exterior vertex or a p-gon. A user-specified set of probabilities governs which of these outcomes occurs. Where to add an edge, what set of consecutive vertices in the boundary ring to join to an exterior vertex, and what edge to use as the base for adding a $p$-gon are selected randomly from the available choices. In the implementation underlying the tests reported in this article, the value of $p$ was selected randomly according to probabilities $c/p$ for $5 \leq p \leq 20$, the constant $c$ chosen so that the probabilities sum to 1. An action that would create either an interior vertex of degree less than 5 or an offending separating cycle is forbidden.

At the completion of each stage of the build, there is a near-triangulation in place. The somewhat intricate penultimate stage in the construction is to add multiple edges and vertices as necessary to assure a boundary ring of minimum degree 4. The final stage turns that near-triangulation into a triangulation by joining the vertices in the boundary ring either to a single vertex added in the exterior region or, if the ring has order at least 12, to three vertices added in the exterior region, in either case forming a 3-cycle boundary ring. When a choice between those alternatives is possible, it is made randomly. The intermediate stages are continued up to the point that the final two stages will produce an internally 6-connected triangulation having order $m \geq n$, where $n$ is the desired order. If $m > n$, the triangulation is rejected and a new triangulation is constructed.

An additional constraint is imposed on the construction to assure the non-existence of an offending separating cycle of length 5 or less: at the end of any stage after the initial pentagon, the boundary ring is required to have length 6 or more, except that a boundary ring of length 5 is accepted if and only if the build can be completed at that point by adding a single vertex in the exterior region so that the resulting triangulation has minimum degree 5, as would be the case, for example, in building the icosahedron. This constraint on the order of the boundary ring also assures that any single exterior vertex added in the final stage has minimum degree 5.

Any internally 6-connected triangulation can be assembled in this fashion, generally in many different ways. Certain triangulations have many fewer ways of being constructed under our set of permitted actions than do other triangulations of the same order and will therefore rarely be sampled.

## References


[1] K. Appel and W. Haken. Every planar map is four colorable, Part I: Discharging. *Illinois J. Math.*, 21(3):429–490, 1977.

[2] K. Appel, W. Haken, and J. Koch. Every planar map is four colorable, Part II: Reducibility. *Illinois J. Math.*, 21(3):491–567, 1977.

[3] G. Birkhoff. The reducibility of maps. *Amer. J. Math*, 70:114–128, 1913.

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

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

[6] B. Mohar. Akempic triangulations with 4 odd vertices. *Discrete Math.*, 54(1):23–29, 1985. `doi:10.1016/0012-365X(85)90059-7`.

[7] N. Robertson, P. Sanders, P. Seymour, and R. Thomas. The four-color theorem. *J. Comb. Theory B*, 70(1):2–44, 1997. `doi:10.1006/jctb.1997.1750`.

[8] J. Tilley. The a-graph coloring problem. *Discrete Appl. Math.*, 217(2):304–317, 2017. `doi:10.1016/j.dam.2016.09.011`.

[9] E. Weisstein. Errera graph. MathWorld—A Wolfram Web Resource. URL: `http://mathworld.wolfram.com/ErreraGraph.html`.

[10] E. Weisstein. Fritsch graph. MathWorld—A Wolfram Web Resource. URL: `http://mathworld.wolfram.com/FritschGraph.html`.

[11] E. Weisstein. Heawood graph. MathWorld—A Wolfram Web Resource. URL: `http://mathworld.wolfram.com/HeawoodFour-ColorGraph.html`.

[12] E. Weisstein. Kittell graph. MathWorld—A Wolfram Web Resource. URL: `http://mathworld.wolfram.com/KittellGraph.html`.

[13] E. Weisstein. Poussin graph. MathWorld—A Wolfram Web Resource. URL: `http://mathworld.wolfram.com/PoussinGraph.html`.

[14] E. Weisstein. Soifer graph. MathWorld—A Wolfram Web Resource. URL: `http://mathworld.wolfram.com/SoiferGraph.html`.

[15] R. Wilson. *Graphs, Colourings and the Four-colour Theorem.* Oxford University Press, 2002.