Linear-time algorithms to color topological graphs

Warren D. Smith
WDSmith@fastmail.fm

June 6, 2005

Abstract —

We describe a linear-time algorithm for 4-coloring planar graphs. We indeed give an $O(V+E+|\chi|+1)$-time algorithm to C-color V-vertex E-edge graphs embeddable on a 2-manifold M of Euler characteristic χ where $C(M)$ is given by Heawood's (minimax optimal) formula. Also we show how, in $O(V+E)$ time, to find the exact chromatic number of a maximal planar graph (one with $E=3V-6$) and a coloring achieving it. Finally, there is a linear-time algorithm to 5-color a graph embedded on any fixed surface M *except* that an M-dependent constant number of vertices are left uncolored. All the algorithms are simple and practical and run on a deterministic pointer machine, except for planar graph 4-coloring which involves enormous constant factors and requires an integer RAM with a random number generator. All of the algorithms mentioned so far are in the ultra-parallelizable deterministic computational complexity class "NC." We also have more practical planar 4-coloring algorithms that can run on pointer machines in $O(V\log V)$ randomized time and $O(V)$ space, and a very simple deterministic $O(V)$-time coloring algorithm for planar graphs which conjecturally uses 4 colors.

1 Summary

In 1890 P.J.Heawood gave $\lfloor(7+\sqrt{49-24\chi})/2\rfloor$ as the minimum number of colors required to color the vertices of any graph embeddable on a 2D surface of Euler characteristic χ. This formula was subsequently proven in every case except the Klein bottle, for which it gives 7 while the correct answer is 6. The most difficult proof by far is the 4-colorability of planar graphs.

We show there is an algorithm running in linear time $O(V+E+|\chi|+1)$ that will find a C-coloring of any V-vertex, E-edge graph with known embedding on any surface with Euler characteristic χ, and with $C \leq$Heawood's corrected bound. (Also, there are linear-time algorithms to detect and color the cases of 1- and 2-colorability; the 3-colorable case can also be handled if the graph-embedding is fully triangulated.) We also give a linear-time algorithm to 5-color a graph embedded on any 2D surface S, except that an S-dependent-size subset of the vertices are left uncolored. All of these algorithms are in the ultra-parallelizable class "NC."

All these algorithms are simple and practical except for planegraph 4-coloring. Then the constants hidden in the O in both the space and time upper bounds are enormous (10^6-10^{15}). Furthermore, although all the other algorithms were deterministic and ran on a pointer machine, the 4-coloring algorithm is (mildly) randomized and requires an integer RAM; it may be derandomized if the runtime bound is multiplied by $O(\log V)$.

Denote large hidden constants with "$\hat{O}$." One source of this hugeness is: our algorithm actually outputs not merely a single 4-coloring, but in fact an $\hat{O}(V)$-size data structure that implicitly simultaneously represents a possibly-enormous number of different 4-colorings. One of those colorings may then be extracted in $O(V)$ time in conventional format.

The best previously known algorithm for 4-coloring planar graphs required $O(V^2)$ time and $O(V)$ space. Only exponential runtime bounds were known for practical 4-coloring algorithms. We describe some hybrid algorithms which attempt to combine the good features of all of these. One has $O(V^2)$ runtime and $O(V)$ space bounds but empirical evidence suggests it usually would run in only slightly superlinear time. Another (which depends heavily on randomization) has an $O(V\log V)$ expected runtime and $O(V)$ space bound, and should be heavily parallelizable since it in the class $\mathrm{NC}_{\mathrm{randomized}}$ (and can indeed be derandomized to show it is in deterministic NC). Finally, another has $O(V+E)$ time and space bound and is fairly easily programmed, but only conjecturally necessarily produces a 4-coloring.

2 Historical Introduction

V, E, and F shall denote the number of vertices, edges, and faces of a graph. A graph is k-colorable iff each vertex may be colored one of k colors with no two adjacent vertices having the same color. A.B.Kempe in 1879 and P.G.Tait in 1880 published different incorrect proofs that planar graphs were 4-colorable. Despite their incorrectness both proofs had some merit.

Tait correctly proved that the vertices of a *m*aximal (i.e. fully triangulated) *p*lanar *g*raph (MPG) were 4-colorable if and only if its edges could be 3-colored in such a way that the 3 edges of each triangle-face had different colors. This in turn is true iff the vertex set of the planar-dual (3-valent) graph is partitionable into disjoint even-perimeter cycles.[1] There are simple linear-time algorithms for interconverting any of these

[1]"Cycles" with perimeter≤ 2 are not allowed – we demand genuine "simple" cycles. Many of Tait's results actually hold for fully-triangulated graphs on arbitrary 2D surfaces, not just the plane.

3 types of solutions, explained on p.103-105 and p.111-112 of [97].[2]

Tait's 4-colorability "proof" then followed from the false assumption ("proven" by J.Chuard in 1932) that the planar dual of any MPG had a Hamiltonian cycle. This is the simplest kind of such "even-cycle partition." The first counterexample MPG was found by W.T.Tutte [112] in 1946. Every MPG (and its planar dual) is necessarily 3-connected by a theorem of Whitney 1932. Holton and McKay [57] showed that the smallest nonHamiltonian 3-valent 3-connected planar graph were six found by Lederberg, Barnette, and Bosak in about 1965 [17] each with 38 vertices, 57 edges, and 21 faces. Tait's original argument therefore shows that every planar graph with ≤ 20 vertices is 4-colorable, but fails at 21 vertices. Indeed, Holton and McKay showed that there are exactly 6 different nonHamiltonian 3-valent 3-connected planar graphs with < 40 vertices (all 6 of them are exhibited in [57] and each has 38 vertices), and all 6 of them happen to be 4-colorable. Tait's argument therefore shows that every planar graph with ≤ 21 vertices is 4-colorable. It has been conjectured that every "fullerene" (3-connected 3-valent planar graph with every face a hexagon or pentagon) has a Hamiltonian cycle. Indeed computers [3][22] have verified this for every fullerene with ≤ 250 vertices. Thus Tait's argument proves that every maximal planar graph with valencies 5 and 6 only, and ≤ 250 triangular faces (i.e. ≤ 127 vertices), is 4-colorable. I can prove[3] that V-vertex fullerenes have cycles with at least $V - O(\sqrt{V})$ vertices, which proves every maximal planar graph with valencies 5 and 6 only is 4-colorable if $O(\sqrt{V})$ vertices are deleted.

The error in Kempe's [69] proof was found by P.J.Heawood in 1890, who gave a 25-country counterexample. This and two additional simpler counterexamples by Errera (17 countries) and de la Vallée Poussin (14 countries) are depicted on pages 123-124 of [117] and in [64]. Gethner and Springer [?] argued that Kempe's proof validly shows the 4-color theorem is true when coloring ≤ 8 countries, but fails at 9 countries. Nevertheless Kempe's key "Kempe chain" idea played an essential role in the 4-color problem's eventual resolution.

Heawood was able to prove, quite simply, that any planar graph is 5-colorable. Furthermore, he showed (essentially) that any graph drawable on a orientable or non-orientable surface with Euler number χ could be colored with $\leq \lfloor (7 + \sqrt{49 - 24\chi})/2 \rfloor$ colors – *except* that Heawood was unable to settle the 3 cases of surfaces with zero "handles."

The Euler number χ of a graph with V vertices, E edges, and F faces is $\chi = V - E + F$. A graph drawn on a surface with $h \geq 0$ handles and $c \in \{0, 1\}$ crosscaps has Euler number $\chi = 2 - 2h - c$. There is only one other kind of 2-manifold, namely the Klein bottle, which has $h = 0$ and $c = 2$ and $\chi = 0$. Almost everything in topological graph theory is a consequence almost solely of Euler's formula and the obvious equalities and inequalities $E \leq (V-1)V/2$ with equality exactly in the case of a complete[4] graph, $2F \leq 3E$ with equality exactly in the fully-triangulated cases, and $V, E, F \geq 0$. For example the theorem that K_5 is nonplanar follows from $E \leq 3V - 3\chi$ so that in our case $V = 5, \chi = 2$ we get $E \leq 9$ in contradiction with $E = (V-1)V/2 = 10$. The fact that K_7 cannot embed on the projective plane (and hence not on the Möbius strip either) arises in exactly the same way. The fact that $K_{3,3}$ is nonplanar is slightly trickier; we need to use the fact that, since this graph is bipartite, it can have only even-sided faces. If all faces have $\geq s$ sides then $2F \leq sE$. Using the resulting inequality $E \leq (V - \chi)s/(s-2)$ where s is the girth (perimeter of the shortest cycle) of the graph (which we here assume is not a tree) yields a contradiction $9 \leq (6-2)4/2 = 8$ proving $K_{3,3}$ is nonplanar.

The quantity $(7 + \sqrt{49 - 24\chi})/2$ in Heawood's formula is exactly the number V of vertices that would cause our two expressions $(V-1)V/2$ (for the number of edges in a complete graph) and $3V - 3\chi$ (the number of edges in a fully triangulated graph) to be equal.

For example, Heawood's theorem states that torus graphs ($h = 1$ handles, $c = 0$ crosscaps, Euler number $\chi = 2 - 2h - c = 0$) are 7-colorable. This is best possible because K_7 is a torus graph (as was originally shown by Möbius and Heawood independently, see figure). The two-holed torus has $h = 1$, $c = 0$, $\chi = -2$ and graphs drawn on it are 8-colorable.[5]

As we shall explain, Heawood's proofs in fact yield *linear-time algorithms* for coloring a graph with a known embedding on an h-handle body (with or without crosscaps) with at most this many colors.

The three cases Heawood could not handle are the following.

In the non-orientable $h = 0$, $c = 1$, $\chi = 1$ case (projective plane and Möbius strip) Heawood's formula says 6 colors should suffice. This is true and we shall present a linear-time 6-coloring algorithm for graphs with known embeddings in the projective plane or Möbius strip. This is best possible because K_6 is embeddable on the Möbius strip (and hence on both the projective plane and Klein bottle, see figure).

In the non-orientable $h = 0$, $c = 2$, $\chi = 0$ case (Klein bottle) Heawood's formula says 7 colors should suffice. This is not best possible since 6 colors suffice and we shall present a linear-time 6-coloring algorithm for graphs with known embeddings on the Klein bottle. From K_6 that is best possible.

[2] Vertex⇒edge coloring: If the two vertices sharing edge E are colored 0&1 or 2&3, then color E with a; if 0&2 or 1&3 then b; and if 1&2 or 0&3 then c. Edge coloring⇒even-cycles: The cycles consist of the edges of two particular colors only. Even-cycles⇒vertex coloring: To 4-color the vertices regard the colors as 2-bit binary numbers; the first bit is the parity of the number of ab-cycles containing that vertex, and the second bit is the parity of the number of bc-cycles.

[3] My proof of this is not published.

[4] The "complete" graph K_n is the n-vertex graph with $(n-1)n/2$ edges.

[5] The smallest number of vertices of a *polyhedron* in 3-space homeomorphic to an h-holed torus, is $4, 7, 10, 10, 11$ for $h = 0, 1, 2, 3, 4$. The tetrahedron is homeomorphic to a sphere and has graph K_4. A polyhedron homeomorphic to a torus with graph K_7 was found by Császár [28] in 1949, and a dualized version by Lajos Szilassi [102] was described in [40] in 1977. Both of these are known to be combinatorially unique. These suffice to prove the optimality of Heawood's bound when $h = 0, 1$, but the cases $h = 2, 3, 4$ from [20] [21] [15] [16] [91] [66] do not suffice for that purpose. It is presently unknown whether a 2-holed-torus-type polyhedron exists that requires 8 colors.

The 1968 proof of Ringel and Youngs [92][91] that the genus of K_n is $\lceil (n-3)(n-4)/12 \rceil$ if $n \geq 3$ proves the optimality of Heawood's bound on orientable surfaces, but at present the greatest known genus for an n-vertex *polyhedron* is only $g \sim n \log n$ [81] and the maximum possible growth of this function of n is unknown.

Finally, in the orientable $h = 0$, $c = 0$, $\chi = 2$ case (sphere) Heawood's formula says 4 colors should suffice. (And this is best possible since K_4 is planar.) This in fact was eventually proven, with great exertion, by Appel, Haken, and Koch in 1976. The ideas underlying their proof were due to Heinrich Heesch. Haken became confident based on a heuristic probabilistic analysis of Heesch's ideas that a proof of his sort must exist and, further, one must exist that is small enough that it should be feasible for a computer to find it. It then became a matter of creative computer programming to guide a machine-search for the proof. Frank Allaire later produced another proof based on essentially the same set of ideas [4]. The latest proof based on these ideas is due to Robertson, Sanders, Seymour, and Thomas in 1997 [93]. It is quite streamlined and error-free compared to the original Appel-Haken proof. However, it still cannot be considered simple since it requires executing about 10^{12} computer instructions to verify it.

E.F.Moore and W.Haken both found reasons to believe that no Heesch-type 4-colorability proof can ever be short and simple: in 1963 Moore found a map (exhibited p.220 of [100], page 95-96 of [97] and 183 of [117]) claimed[6] not to contain any reducible configuration with ring size$\leq$ 11. Haken's probabilistic estimates suggest that no Heesch-type proof exists with only ring-sizes$\leq$ 13 and that any unavoidable set of configurations that do not include Heesch's "obstacles" is necessarily very large.[7]

Ringel & Youngs [91] in 1968[8] constructed 12 infinite families of graph embeddings, plus several "sporadic" small-n cases, which in combination show that genus$(K_n) = \lceil (n-3)(n-4)/12 \rceil$ if $n \geq 3$. (They also showed genus$(K_{st}) = \lceil (s-2)(t-2)/4 \rceil$ if $s, t \geq 2$.) This proves that Heawood's formula cannot be improved upon in the case of orientable surfaces. Ringel [90] in 1959 had already shown the non-improvability of Heawood's formula in all non-orientable cases other than the Klein bottle; shorter proofs are in [91][73] of the fact that K_n embeds in a nonorientable surface with h handles when $h = \lceil (n-3)(n-4)/6 \rceil$ when $n \geq 3$ and $n \neq 7$, and K_7 embeds with $h = 3$.

In view of Möbius's complete classification of 2-dimensional topologies [7] all this has completely settled the coloration problem for graphs drawable on every possible 2-dimensional surface topology. We may summarize the situation as

Theorem 1 (Heawood's improved color↔genus formula). *Let S be a 2D surface, not necessarily orientable, with Euler number χ. Let G be a graph drawn on S without crossings. Then*

$$C = \lfloor (7 + \sqrt{49 - 24\chi})/2 \rfloor - 1_{S=\text{Klein bottle}} \qquad (1)$$

colors are sufficient to color the vertices of G, and on every surface there exists a graph G such that this number of colors is necessary.

Theorem 1 represents the combined fruit of work by L.Euler, A.Legendre, S.Lhuilier, A.F.Möbius, F.Guthrie, A.B.Kempe, P.G.Tait, P.J.Heawood, L.Hefter, A.Bernhart, G.D.Birkhoff, P.Franklin, W.Gustin, J.Mayer, J.W.T.Youngs, G.Ringel, H.Heesch, K.Appel, W.Haken, and J.Koch during 1750-1977.

Remarks. It has often been erroneously claimed (e.g. [86][37][70][71][18][52]) that the Appel-Haken proof [6] leads to a quadratic-time 4-coloring algorithm for planar graphs. Robertson et al. [93][95] claimed, however, that it actually only yields a quartic-time algorithm (and claimed Appel and Haken knew this in 1989). The confusion arises from subtleties concerning the precise definition of "embedding" of a reducible configuration, and of the definition of "reducible configuration" itself – e.g. can the configuration "overlap itself"? According to [93], new "block-count reducibility" concepts proposed by D.I.A.Cohen and by Gismondi & Swart [45] in an effort to allow proofs simpler than those attainable purely by means of Appel & Haken's reduction-types, may make the algorithmic situation even worse – perhaps even not yielding a polynomial-time 4-coloring algorithm at all. Fortunately their new simpler 4-colorability proof [93][94] yields a quadratic-time coloring algorithm [95].

In practice, the experimentally-best available 4-coloring algorithms for planar graphs seem to be by Morgenstern and Shapiro [85][64]. Using their techniques plus an outer implicit enumeration or randomization-restart, one may build an algorithm that empirically always seems to succeed in slightly-superlinear time ($V^{1.1}$?) but might, as far as is presently known, require C^V steps for some $C > 1$ on some as-yet-unknown infinite family of "bad" planar graphs.[9]

Wagner [113][118] in 1937 extended the 4-color theorem by showing that graphs without K_5 minors are 4-colorable if and only if planar graphs are; Robertson et al. [96] further showed that graphs without K_6 minors are 5-colorable if and only if planar graphs are 4-colorable. H.Hadwiger conjectured in 1943 that every connected c-chromatic graph is

[6] Presumably this could be shown by checking every ring with size$\leq$ 11 in Moore's map by computer to demonstrate the non-reducibility of the configuration it encloses. But I am unaware of any published proof of this and do not understand how Moore could have accomplished that in 1963, considering [117] the first computerized reducibility tests were made by Heesch in 1965. A complete list of all reducible configurations with ring-size$\leq$ 10 was prepared by Allaire and Swart [5] in 1978. According to Bernhart [12] this was extended to ring size 11 in work that was never published. That would have enabled Moore to do what he claimed – albeit 15 years after he did it. And indeed, in [12] a much larger map (864 countries) was given by Moore on 26 March 1977 again claimed to contain no reducible configuration with ring size$\leq$ 11 (but it does contain one with ring-size 12), because a previous map had proved "inadequate."

[7] Haken suggests that an extreme example of this sort of phenomenon might be the conjecture that the first n digits of π, infinitely often, yield an integer that is a perfect square. Haken conjectures this statement both is true and has no finite-length proof. The existence of proofless true statements was shown by K.Gödel. The existence of an infinite set of provable statements, the Nth of which is $O(N)$ bits long, but whose shortest proof is asymptotically longer than any computible function of N, was (essentially) shown by A.M.Turing.

[8] L.Hefter ($n \leq 12$), C.Terry, L.Welch ($n = 12k$), W.Gustin ($n = 12k + \{1, 4, 9\}$), J.Mayer ($n = 20, 23, 30$) and R.Guy ($n = 59$) were also involved in the proof. Hefter in 1891 pointed out that Heawood had failed to prove optimality except in the torus case, and then constructed optimality proofs for genus 1,2,3,4,5,6 and found the genus of K_n if $n - 7$ is divisible by 12.

[9] They tested their program on graphs with up to 512,000 vertices and in all tests so far, success has been achieved rapidly and a restart has never been needed. However, these experiments are not entirely convincing because their test-graphs, although supposedly designed to be hard to color, constitute only a tiny fraction of all planar graphs and, e.g. include no graphs having any very-high-valent vertices. Thus from [3] we know that all 3-connected maximal planar graphs with $\leq$ 90 vertices and maximum valence 6 are 4-colorable because their duals are Hamiltonian; this suggests Morgenstern and Shapiro's test graphs may actually have been *easier* to color than general planar graphs.

[10] In combination with a result of Dirac [30] handling the cases $c \leq 4$.

contractible to K_c; these results[10] proved that for $c \le 6$. Theorem 10.4.5 of [87] shows Hadwiger for $c+1$ colors implies Hadwiger for c colors. R.Halin [53] showed that for every integer $m > 0$ there is an $F(m)$ such that every $F(m)$-chromatic graph is contractible to K_m, and by theorem 10.4.6 of [87] it is permissible to take $F(m) = m$ for $m = 2, 3, 4, 5, 6$ and $F(m) \le 2F(m-1) - 1$ for $m \ge 7$. S.Khuller [70] converted Wagner's theorem to an $O(N^2)$-time algorithm for 4-coloring graphs without K_5 minors, and [71] also showed that graphs without $K_{3,3}$ minors are 5-colorable (which is best possible since K_5 is such a graph). Khuller's arguments (combined with the present paper's demonstration planar 4-coloring is in NC) indeed now show that both these coloring tasks are in the class "NC." Furthermore, I can show[11] that, *assuming* [67] is correct, that there are $O(E + V\alpha(E, V))$-time, $O(V + E)$-space sequential algorithms for 4-coloring a graoph with no K_5 minor, or for proving or disproving it has a K_5 minor; here α denotes an inverse Ackermann function [104][99]. The present paper's linear-time 4-coloring algorithm for planar graphs now shows easily that there is a linear-time algorithm for 5-coloring graphs without $K_{3,3}$ minors.[12] The [96] proof implies a polynomial-time algorithm to 5-color graphs without K_6 minors, but I am not sure of the degree of the polynomial. Jørgensen conjectured that every 6-connected graph without K_6 minors is an "apex graph" (i.e. a graph with a "apex vertex" whose removal renders it planar). If that is true then a cubic-time 5-coloring algorithm would exist.

Matiyasevic [79] proved the 4CT equivalent to the claim that two probabilistic events about random edge colorings, are positively correlated.

Kauffman [68] proved the insanity-inducing theorem that the "associativity" statement "for any two ways to parenthesize a 3D vector-cross-product $\vec{a} \times \vec{b} \times \cdots \times \vec{z}$ there exists an assignment of $\{i, j, k\}$ to the vectors causing the two expressions to be equal and nonzero" is true if and only if the 4-color theorem is true. (Here $i = (1, 0, 0)$, $j = (0, 1, 0)$, $k = (0, 0, 1)$.)

In fact, Kauffman's argument can be used to show, not only equivalence, but in fact *linear time* equivalence, i.e.

Theorem 2 (Improved Kauffman). *There is a linear-time algorithm for 4-coloring planar graphs if and only if there is a $O(N)$-time algorithm that, given any two parenthesizations of an N-term 3D vector cross product, will find an assignment of i, j, k to the variables causing the two products to be equal and nonzero.*

Proof sketch. Given the two parenthesizations, regard them as rooted binary trees and "glue" their leaves together with the leaf of tree #1 representing the kth variable being glued to the leaf of tree #2 representing the kth variable, thus combining both leaf-edges into a longer single edge. Also add an edge joining the two tree roots. The result is a 3-regular planar graph, and Kauffman [68] shows how from a 4-coloring of its faces we may easily deduce an i, j, k assignment or vice versa.

In the other direction: given a maximal planar graph G, in linear time find [59] its triconnected components and separating triangles [24]. Each triconnected component is itself a maximal planar graph and has a Hamiltonian cycle by a 1931 theorem of Whitney.[13] For a linear-time algorithm to find such a Hamiltonian cycle see [26]. The triangle-faces lying "inside" the Hamiltonian cycle form a tree in the planar dual G', and the ones lying "outside" it do also. On these two trees regarded as parenthesized products call the i, j, k-assigner, then use its result to 4-color that triconnected component. Finally, glue together the 4-colorings of all the triconnected components of G (by color-renaming if necessary) in linear time (which can be done if the gluings are done in a root-outward order guided by the tree-structure of the triconnected components) to get a 4-coloring of the whole of S. □

Kauffman's result is highly related to an earlier result of H.Whitney's about "n-fold arranged sums" [115]. A full parenthesization such as $[(a_1+[(a_2+a_3)+(a_4+[a_5+a_6])])+a_7]$ of a sum $a_1+a_2+\cdots+a_n$ is called an "n-fold arranged sum," and a subsum enclosed by parentheses (such as a_2+a_3) is a "partial arranged sum." Whitney proved that the 4CT is equivalent to the statement that, given any two n-fold arranged sums, one can give integer values to $a_1, \ldots, a_n$ so that none of the partial arranged sums are multiples of 4.

Hadwiger [51] showed that the 4CT is true if and only if, for any convex polyhedron there exists some sequence of "corner cutting-off" operations eventually yielding a polyhedron each of whose face sidecounts is divisible by 3. (And at most $V/2$ such cut-off operations are necessary for a V-vertex polyhedron.)

Minty [82] showed that the 4CT is true if and only if there is a way to orient the edges of any planar graph so that around any cycle, the ratio of forward to backward orientation-counts is $\le$ 3:1. This equivalence arises from a linear-time algorithm in one direction but a slightly superlinear algorithm in the other.[14]

We also should note that any V-vertex graph embeddable on a surface of Euler number χ is "almost" 4-colorable since by removing $O(\sqrt{|\chi - 2|V})$ vertices it may be made planar [63][31].[15] Joan Hutchinson's 1984 question [61] remains open: can any graph embeddable on any fixed surface S be made 4-colorable by removing a number of vertices depending on S *only*?

Thomassen [106] showed that any graph embeddable, without small-cardinality noncontractible cycles, on any fixed surface S (orientable or not) is 5-colorable. This goes beyond

[11] My proof of this is not published, and since the "extended abstract" [67] states many of their results without proof, it would first be necessary to go through [67] converting it into a genuine mathematical paper, before converting my unpublished proof into the same.

[12] Split the graph into its triconnected components [59]; use Wagner's theorem that these pieces all are either planar or K_5 to color the pieces; finally glue the colorings with the aid of appropriate color-renamings in linear time.

[13] Whitney's theorem was later extended by Tutte and Thomassen, who showed every pair of vertices in a 4-connected planar graph is connected by a Hamiltonian path; by Thomas & Yu, who showed every edge of a 5-connected torus graph and 4-connected projective-plane graph is part of a Hamiltonian cycle, and by Jackson & Xu [65] who showed that if every 4-connected piece of a maximal planar graph shares a triangle with at most three other pieces, then G is Hamiltonian. Grünbaum and Nash-Williams conjectured that 4-connected torus graphs are Hamiltonian.

[14] The easy direction: orient each edge from the lesser to the greater color in $O(V + E)$ time. The hard direction involves solving a single-source shortest path problem in a directed version of our graph involving arc "lengths" -1 for going with the arrow and $+3$ for going against it, and with no negative-length cycles. This may be done in $O(V \log^3 V)$ time according to [33]. The colors are then the distances mod 4.

[15] And there is an $O(|\chi| + V + 1)$-time algorithm to find such a vertex set, if the graph's embedding is known.

Hutchinson's [61] theorem that any graph embeddable on any fixed orientable surface S in such a way that all edges are short compared to the minimum-length noncontractible curve on S, is 5-colorable.[16] Steve Fisk gave an infinite class of 5-chromatic graphs embeddable on the torus and projective plane with all edges short (specifically, any triangulation of the torus or projective plane with exactly two odd-degree vertices, is 5-chromatic) so these Hutchinson and Thomassen theorems are best possible.

It is a consequence of Thomassen's result (he did not mention this consequence, but presumably knew it) that any graph embeddable on any fixed surface S (orientable or not) can be made 5-colorable by removing a number of vertices depending only on S. (Proof: remove the vertices in a low-cardinality noncontractible cycle [increasing χ]; keep doing it until either 5-colorability or planarity is reached.) This also shows that a polynomial-time algorithm exists to find the vertex subset to remove and to find the 5-coloring, albeit with the degree of the polynomial depending dramatically on S.

What Thomassen did not know, is that there is a *linear* time algorithm to do this, which also yields a far simpler proof. We shall describe it in §3.

Gallai [39] (as improved by Krivelevic [75]) showed that any k-color-critical graph with $V > k$ vertices and E edges obeys

$$E \geq \left(\frac{k-1}{2} + \frac{k-3}{2k^2-4k-2} \right) V \qquad (2)$$

from which it follows (from Euler's formula) that there are only a finite number of 7-color critical graphs[17] on any fixed surface S. Thomassen [110] showed there are only a finite number of 6-color critical graphs on any fixed surface S, but for each surface other than the sphere Fisk's construction gives an infinite number of 5-color critical graphs. From this it follows there is a polynomial-time algorithm[18] to determine if a graph embeddable on S is C-colorable for each $C \geq 5$. (The algorithm is to see if any of the bad graphs are in there – if not it is C-colorable. We claim without proof that a polytime algorithm also can actually find the coloring.)

Thomassen [111] showed that, on any fixed 2D surface S (orientable or not) there is a polynomial time algorithm to determine the chromatic number of any graph of *girth*≥ 5 embeddable on S. (And obviously, deciding whether girth≥ 5 is a polynomial-time task.)

Finally, 4 colors is best possible in the very strong sense that there are an infinite number of planar graphs whose maximum independent set has cardinality $V/4$ where V is the number of vertices.[19] As [52] observe, any C-coloring algorithm may be converted into an algorithm to find a maximal[20] independent set of $\geq V/C$ vertices by simply taking the most popular color class, then for each other color class (considered one color at a time) adjoining all the vertices in it that one can.

For one beautiful example of how large independent sets can be useful (they yield an $O(N)$-space data structure for 2D point location which may be built in $O(N)$ time and used in $O(\log N)$ time) see [72].

3 Linear-time algorithms in the non-spherical cases

First linear-time algorithm for coloring an embedded graph with C colors:

1. If the graph has ≤ 4 vertices, color it optimally by brute force. Stop.
2. Find a vertex X with minimum valency ν.
3. Delete X from the graph. (Optionally, the ν-gonal "hole" may now be triangulated by diagonals.)
4. Color the remaining smaller graph recursively.
5. Color X with the first color not used by its neighbors. Stop.

Because $E \leq 3(V-\chi)$ the average valency is $6(V-\chi)/V$ and hence in step 2 it will always happen that $\nu \leq \lfloor 6(V-\chi)/V \rfloor$ and hence $C \leq \lfloor 7 - 6\chi/V \rfloor$. This gives $C \leq 6$ in the case of the sphere, projective plane, cylinder, plane, or Möbius strip, and $C \leq 7$ in the case of a torus or Klein bottle. In the case of a fully-triangulated graph, the C produced is no larger than Heawood's improved formula, in every case except the sphere and Klein bottle.

Second linear-time algorithm for coloring an embedded graph with C colors:

1. If the graph has ≤ 4 vertices, color it optimally by brute force. Stop.
2. Find a vertex X with minimum valency ν; if $\nu = C$ then among X's neighbors, find (if one exists) a nonadjacent pair A, B with $\max\{\deg(A), \deg(B)\} \leq 11$. (If no such A, B exist, then skip this X, i.e. go back to step 2 and find another minimum-valence X.)
3. Delete X from the graph and shrink[21] A, B into a single vertex, and uniquify any duplicated edges created by the pair-shrinks. (Optionally, additional such shrinks of other nonadjacent pairs may also be done, provided this does not destroy embeddability, and the "holes" could now be triangulated by diagonals.)
4. Color the remaining smaller graph recursively.
5. Color X with the first color not used by its neighbors. Stop.

Our second linear-time algorithm betters or matches the improved Heawood formula in every case except for the spherical case.

[16] But Thomassen's theorem does not actually obsolete Hutchinson's because his S-dependent constants are considerably worse than her's. Thomassen showed on a genus-g surface that if all noncontractible cycles have at least 2^{14g+6} edges, then the graph is 5-colorable.

[17] A graph is "k-color-critical" if it has chromatic number k but any proper subgraph has smaller chromatic number.

[18] But the degree of the polynomial depends on S, probably rather dramatically.

[19] E.g. simply consider a bunch of disconnected K_4s, which later may optionally be interconnected.

[20] A maxim*al* independent set cannot be enlarged by adjoining another vertex, but is not necessarily of maxim*um* cardinality among all independent sets.

[21] We use the word "shrink" instead of the more standard word "contract" because we wish to avoid confusion; the latter word occurs with a different meaning – namely the legal, or bridge-player meaning – elsewhere in this paper.

For the analysis showing how to make these algorithms run in linear time, see theorem 4. This analysis will depend on

Lemma 3 (Reducible vertices are common). *In the plane 5-coloring and Klein bottle 6-coloring cases, not only must a minimal-valence vertex with valence C ($C = 5, 6$ respectively) have two nonadjacent neighbors, but in fact one must exist with both those neighbors having valency≤ 11, and indeed, at least a positive constant fraction of all minimal-valence vertices must either have valence$< C$ or obey this property.*

Proof: Franklin [36] in 1934 showed that K_7 does not embed on the Klein bottle (also proven in [91]). We have seen already that K_5 is nonplanar.

It therefore follows that, in the Klein bottle case, $\nu \leq 6$ always *and* step 2 will always succeed in finding a nonadjacent pair A, B. Similarly, in the sphere case, $\nu \leq 5$ always *and* step 2 will always succeed in finding a nonadjacent pair A, B.

This already proves the second algorithm (if the degree-constraints on A and B are ignored) will always succeed in 5-coloring any plane graph or 6-coloring any Klein bottle graph, and will achieve the same C-bounds as the first algorithm in the other cases.

The rest of the lemma may be proven purely from Euler's formula. We know it is not possible for every neighbor of every C-valent vertex to have degree≥ 12 (or even for too large a constant fraction of C-valent vertices to have every neighbor of degree≥ 12) if the minimum valency is C. That is because, if that were possible, then the average valency would be larger than is permitted by Euler's formula. □

Remarks. For other 5-coloring algorithms for planar graphs see [37][52][80][95][116] and ch.30 of [2]. Among all these algorithms, the best[22] is Frederickson's [37], which is based on the theorem[23] that a planar graph either contains a vertex of valence≤ 4 or a 5-valent vertex with two mutually nonadjacent (≤ 7)-valent neighbors. However, an idea of Williams [116] may be used to simplify Frederickson's algorithm further. Frederickson's theorem can be strengthened to show that *at least a positive constant fraction* of all the 5-valent vertices in a planar graph with minimum valence 5 are of his type. Therefore, it is permissible to simply go through the queues of ≤ 4- and 5-valent vertices applying the reduction for those vertices of Frederickson's types, and skipping them otherwise, in which case the skipped vertex is moved to the rear of the queue. Because all valencies are nonincreasing, and because of the strengthened Frederickson theorem, whenever the algorithm has used up enough of the queue so that it scans a vertex again, the number of vertces in the graph has decreased by at least a constant factor. In view of the convergence of geometric series, this forces linear runtime. (And Williams recommends tuning Frederickson's constant "7" by considering replacing it with larger values if that improves performance.) The resulting Frederickson-Williams-hybrid 5-coloring method then corresponds, essentially, to the plane $C = 5$ special case of our second linear time algorithm.

The linear-time 5-coloring algorithm in [95] for *embedded* fully-triangulated planar graphs, is inadequately and incorrectly sketched. It is based on Wernicke's 1904 theorem [114][24] that in a fully-triangulated planar graph, there must exist either a vertex of degree≤ 4 or a vertex of degree$= 5$ having a neighbor of degree≤ 6. They should have stated that we need to maintain two doubly-linked lists (not "a stack") of all vertices of degrees ≤ 4, and of all vertices of degree 5 having a degree≤ 6 neighbor, and should have noted that these lists all may be fully updated in $O(1)$ time after performing any edge deletion, bounded-valence vertex deletion, or edge-shrinkage, and this fast-updating is only possible because we have an *embedding* represented, e.g. using the Guibas-Stolfi winged-edge data structure [50].[25]

Thomassen [109] did not actually state an algorithm, but merely an elegant 5-colorability proof.[26] His proof is convertible into a quadratic-time algorithm for 5-coloring embedded planar graphs all of whose faces except for the external face are triangles. Heawood's original 5-colorability proof [55] also is convertible into a quadratic-time 5-coloring algorithm.

Theorem 4 (Linear space and time). *The plain version of our first algorithm, may be made to run in time and space $O(V + E + |\chi| + 1)$ on a deterministic pointer machine. The second algorithm, may be made to run in the same space bound and in time $O(E + |\chi + 3|V)$. In neither case is it necessary to actually know a graph embedding. These bounds also are valid for the optional variant algorithms but in those cases the embedding is needed.*

Proof: 1. We employ "degree-queues." Each vertex X knows its degree, for each integer ν with $0 \leq \nu \leq H$ (where $H = V - 1$ most simply, but we can make H be the corrected Heawood color bound if that is smaller) there is a doubly linked list of all the vertices with valence ν.

2. To represent the graph we make each vertex have a doubly-linked list of its neighbors (and each vertex X points not only to its neighbor Y but in fact to X's entry in Y's neighbor list) This permits deleting a vertex of valence ν, or visiting all its neighbors, in $O(\nu + 1)$ time, adding or deleting an edge ab between two vertices *provided* the position of that edge in a's and b's lists are known, in $O(1)$ time, and shrinking a given edge in $O(1)$ time.

3. Note that all the degree information associated with each vertex and in the degree-queues may be updated as any of

[22] Frederickson's method does not require the graph to be triangulated and does not require an embedding to be known, and it will work for many nonplanar graphs (the algorithm either succeeds in finding a 5-coloring, or fails in which case that proves the graph was nonplanar), and it is fast and simple.

[23] Whose proof as usual requires nothing more than Euler's formula.

[24] Which also may be proven using Euler's formula alone. Indeed, one may show $4e_{55} + e_{56} \geq 60$ in any maximal planar graph with minimum valence 5, where e_{ab} is the number of edges between an a-valent and b-valent vertex. If each face of an icosahedron is subdivided into 4 triangles then we get a polyhedron with $V = 42$, $E = 120$, $F = 80$, $e_{55} = 0$, and $e_{66} = e_{56} = 60$, showing that this bound is tight.

[25] And they alluded to an "amortized analysis" of their algorithm. But no amortizing is needed to analyse the corrected algorithm.

[26] Actually, he proves something rather stronger than mere 5-colorability, e.g. he gets 5-colorability even if the colors on the outer cycle are chosen from arbitrary presepcified 3-element lists and the remainign colors are chosen from arbitrary 5-element lists of eligible colors, one such list for each vertex. This result is best possible because a 63-vertex planar graph with 4-element eligible-color lists at each vertex, which cannot be colored [83].

these operations are processed, while only increasing their runtimes by constant factors at most.

4. In the second algorithm we may test adjacency of arbitrary vertices a and b in $O(1)$ steps because they have bounded valence.

5. The above ideas (and the fact that the vertex we delete always has valency $O(\sqrt{|\chi|+1})$) suffice to make the plain versions of both algorithms run in the claimed time bounds. The optional enhancements also run in those time bounds if the graph's embedding is initially available. □

Third linear time algorithm:

1. Find a vertex A with minimum valency ν.
2. If $\nu \geq 7$ then fail (i.e. leave all the vertex colors blank).
3. If $\nu = 6$ then, among A's neighbors, find (if one exists) a mutually nonadjacent triple X, Y, Z or two nonadjacent pairs W, X and Y, Z not dovetailed in the cyclic order around A (with all vertices in these pairs or triple required to have degree≤ 12). See the figure. If no such triple or two pairs exist, then skip this vertex and try again with another 6-valent vertex – if there is nonexistence for every 6-valent vertex then fail.
4. If $\nu = 5$, then among A's neighbors, find (if one exists) a mutually nonadjacent pair S, T; if no such pair exists then skip this vertex and try again with another 5-valent vertex (and also allow trying 6-valent ones as in the previous step if we run out of 5s) – if there is nonexistence for every 5- and 6-valent vertex then fail.
5. Delete A, identify (i.e. shrink) the $X = Y = Z$, or $W = X$ and $Y = Z$, or $S = T$ (or no vertex identifications are needed if $\nu \leq 4$) as the case may be. Uniquify any duplicated edges created by these pair-shrinks.
6. Color the remaining smaller graph recursively.
7. Color A with the first color not used by its neighbors. Stop.

Theorem 5 (Linear-time 5-coloring if $\leq F(\chi)$ vertices are removed). *The third algorithm will run in $O(V+E+1)$ time and space on a deterministic pointer machine provided an embedding of the graph on some surface of Euler number χ is provided. It will not fail until the number of vertices in the reduced graph is $\leq F(\chi)$ for some function F. It will find a 5-coloring of all the other $\geq V - F(\chi)$ vertices.*

Proof: This theorem and its proof are heavily based on Hagerup et al. [52]. Hagerup et al indeed even discussed extending their results to handle the positive genus case, but that part of their discussion was incorrect,[27] because their lemma 1 depended upon planarity. Our contribution is to detect and repair that error.

Their lemmas 1-4 show that, as a consequence of Euler's formula, in any χ-embeddable graph with $\geq F(\chi)$ vertices, at least a positive constant fraction of those vertices are "reducible" i.e. have valence≤ 6 and obey at least one of our four failure-preventing conditions. All these lemmas were proven [52] for planar graphs but still are true for graphs embedded on a surface of Euler number χ, except that lemma 1 needs to be modified as we shall now discuss.

It is not true that every 5-valent vertex must have 2 nonadjacent neighbors (although this is true in the plane since K_6 is nonplanar). However, every occurrence of K_6 in the surface cuts that surface into pieces each of which have smaller topological complexity as measured by χ. That forces there to be at most $F(\chi)$ different induced K_6 subgraphs of G. That forces every 5-valent vertex *except* for at most a constant number of exceptions, to have 2 nonadjacent neighbors.

It also is not true that every 6-valent vertex must have 2 nonadjacent nondovetailed neighbor-pairs or a nonadjacent neighbor triple – although this is true in the plane due to the planarity argument shown in the figure. However, again, whenever this planarity argument fails A and its 6 neighbors must induce a nonplanar subgraph of G, and by the same argument as before, the number of occurrences of any particular nonplanar induced subgraph of G inside G, is bounded by $F(\chi)$. That forces every 6-valent vertex *except* for at most a χ-dependent number of exceptions, to obey [52]'s lemma 1; further, in at least a positive constant fraction of these instances, A's neighbors all must have bounded valence as a consequence of Euler's formula.

We again employ degree queues for degrees≤ 6 with skipped vertices being moved to the rear of the queue. To represent the graph and its embedding we employ a data structure in which each vertex has a doubly-linked list of its neighbors in clockwise[28] order (and A's pointer to neighbor B points not only to B, but in fact to where in B's clockwise order A lies). Or, equivalently, the elegant "winged edge" data structure of Guibas & Stolfi [50] could be employed.

Hagerup et al. [52] go further and in fact show how to implement our algorithm in *parallel* on an EREW PRAM in $O(\log N \log^* N)$ time on a machine with $O(N/(\log N \log^* N))$ processors. □

Remarks. The best $F(\chi)$ is probably $O(|\chi| + 1)$ but I have only proven the much weaker statement $F(\chi) \leq 35^{3-\chi}$, $-\infty < \chi \leq 2$.

Actually all three of these linear-time algorithms may also be regarded as NC parallel algorithms if we alter them slightly and take advantage of lemmas that the reducible configurations actually are dense in the grah, i.e. occur in at least order V locations simultaneously; we then find a vertex-disjoint set of such configurations with cardinality of order V using a fast parallel independent set finding algorithm [46] and reduce them all.

4 The cases of one, two, and three colors

It is trivial to decide (in linear time $O(V + E)$) if a graph is 1-colorable (has no edges) or 2-colorable (bipartite) and to find the (essentially unique) colorings if so.

Lemma 6 (3-colorability). *A fully-triangulated plane graph is 3-colorable if and only if every valence is* even.

[27] It is, however, correct if their algorithm's goal is merely to find a 7-coloring, rather than, as here, a 5-coloring; and this arguably is all they were claiming. In that case our result is new.

[28] If this order is not known then we regard the embedding as not being known.

Proof: The "if" part is readily deduced from Tait's even-cycle decomposition theorem and the recognition that the dual graph necessarily is bipartite and hence has only even cycle perimeters. The "only if" is from considering coloring just a vertex and its immediate neighbors; if the vertex has odd valence, it cannot be done. □

This fact was (essentially) stated by Kempe as a side-remark to his 1879 "proof" in American J. Math'cs.

Lemma 7 (More powerful 3-colorability fact – equivalent in the plane[29]). *A fully-triangulated graph embedded on a 2D surface is 3-colorable if and only if the triangle-faces are of two types, "light" and "dark," with no two triangles of the same type sharing an edge, i.e. if the graph dual to the triangulation is bipartite.*

Theorem 8 (Fourth linear-time algorithm[30]). *It takes linear time $O(V+E+F)$ to decide whether a fully-triangulated graph embedded on a 2D surface is 3-colorable, and any such coloring is necessarily unique[31] and is readily found by first coloring the vertices of one particular triangle-face, then always coloring a vertex adjacent to an already-colored pair of adjacent vertices.*

Remarks. Our *second* linear-time algorithm will also work to find a 3-coloring (if there is one) of any fully-triangulated graph embedded in the plane. That is because the minimum valence (since all valencies are even) is always 4 and since K_5 is nonplanar step 2 will always find a nonadjacent pair AB. After removing the 4-valent vertex all 4 of its neighbors get odd valency, but upon shrinking AB and uniquifying the now-duplicated edges they all are restored to even valency and the reduced graph still is fully triangulated.

Our "more powerful fact" was essentially known to Heawood in the 1930s. Furthermore, according to [101], the following more general planar theorem was known to Heawood [56] and was rediscovered by various other authors e.g. [78]: A planar graph G is 3-colorable *if and only if* there exists an even-valent triangulation T such that G is a subgraph of T. (But this theorem usually seems unhelpful for the purpose of deciding 3-colorability!)

Stockmeyer [41] showed it is NP-complete to tell whether a planar graph is 3-colorable, even if the graph is connected and the minimum and maximum valences are 2 and 4, but as we've just seen there are linear-time algorithms in the fully triangulated case. By modifying[32] the Stockmeyer proof, we can force his graph to be *3-connected* planar and optionally can also force the minimum and maximum valencies to be 3 and 4, thus proving the NP-completeness of deciding 3-colorability for *polyhedral* graphs which are *not* fully triangulated.[33]

There is also a (trivial) linear-time algorithm to find a 3-coloring (which always exists) if the graph is a triangulated simple polygon.

A famous theorem of Grötzsch ([47][49][107][108][74] and [87] theorem 13.2.1) states that any triangle-free planar graph is 3-colorable; Kowalik [74] was able to turn this theorem into an $O(N \log N)$-time, $O(N)$-space algorithm (involving a remarkable data structure invented by him) to find the 3-coloring. Kronk and White [76] showed that triangle-tree torus graphs were 4-colorable and Thomassen [107] that girth$\geq$ 5 torus graphs were 3-colorable; Gimbel & Thomassen [44] showed girth$\geq$ 6 graphs on the double-torus are 3-colorable. (There are known linear-time algorithms [88][24] to test planar and torus graphs for triangle-freeness and quadrangle-freeness and these were generalized to arbitrary fixed cycle length and fixed genus by Eppstein [32].)

5 Some remarks about coloring geometrical graphs

Many graphs that arise in computational geometry settings (such as finite-element codes) obey the following **one-sided valence bound property:** there exists an ordering of the vertices and a constant B such that each vertex has $\leq B$ neighbors among the vertices with larger index in the ordering.

For example, let there be V (possibly overlapping) objects, each a convex set of bounded aspect ratio (ratio of circumradius over inradius) in any fixed-dimensional space, such that each point of space lies within at most some constant number of objects. Then: the intersection graph of the objects obeys a one-sided valence bound property, using as the vertex-ordering, just the ordering of the objects by increasing volume.

Useful Fact. Any class of graphs closed under vertex-deletion and obeying a one-sided valence bound property has chromatic number $\leq B+1$ and our first linear time coloring algorithm works to produce such a coloring.

Many other properties of graphs have been proposed that seem of less computational-geometric interest/usefulness. For example, graphs with "excluded minors" are of little use because the nearest- and second-nearest-neighbor graph of the equilateral triangle grid does not exclude *any* minor. For another example, the class of graphs embeddable in the plane with few (compared to V or E) edge crossings, also do not exclude any minor and indeed can include K_n as an *induced* subgraph for arbitrarily large n and hence have arbitrarily

[29] Whose proof is essentially the same. See footnote 2. Warning: the fact that all faces have an even number of sides, while sufficient to assure bipartiteness of a planar graph, does *not* suffice for graphs on other 2D surfaces. Indeed an infinite number of 4-chromatic quadrangulations of the Klein bottle and projective plane are known, as are embeddings of K_n on handlebodies with all faces even and arbitrarily large n. See [62][35][84].

[30] This algorithm also runs in $O(\log V)$ *parallel* time using $O((V+E)/\log V)$ processors by checking bipartiteness fast by computing the distance-2 graph (which is a linear-size graph if the original graph has bounded valences; and 3 is bounded) and then using a fast algorithm [42] to compute connected components. This works because 3 is bounded; I do not know of an efficient highly-parallel algorithm to check bipartiteness in graphs with no upper bound on valence. However, we *can* test bipartiteness in polyhedral planar graphs in $O(\log V)$ steps using $O((V+E)/\log V)$ processors; create the (linear-size) graph consisting of distance-2 relations between vertices *on the same face* and then use a fast algorithm [42] to compute connected components.

[31] Up to color renamings.

[32] Clone v_2 and v_3 in the original nonplanar construction so that v_1 has valency 4 instead of 2 and to increase the connectivity. Now apply the usual crossover [41] gadgets to make it planar, then optionally employ the node-duplication gadgets to make the maximum and minimum valencies be 4 and 3.

[33] Steinitz's theorem [48] says the graphs of convex polyhedra are exactly the 3-connected planar graphs.

large chromatic number.

6 Four colors suffice

Wlog our planar graph is fully triangulated, i.e. has $E = 3V-6$, and is already embedded in the plane. (Linear-time algorithms are known for testing planarity [60], producing such an embedding [25][19] and for adding diagonals to reach fully triangulated form.) Although one might presume it to be easier to 4-color *non*maximal planar graphs, we shall ignore that possibility because our focus here is to devise the simplest possible algorithm that works even for worst-case input, not to produce the most practical or fastest algorithm.

Theorem 9 (Robertson,S,S,T 1997 [93][95]). *Any embedded maximal planar graph G with at least 10 vertices contains either (or both) of the following:*

1. *A k-cycle with $k \in \{3,4,5\}$ containing at least $\lfloor (k-1)/2 \rfloor$ vertices of G in its interior and at least $\lfloor (k-1)/2 \rfloor$ vertices of G in its exterior. This case arises automatically if any vertex in G has degree≤ 4. All k of the vertices on the k-cycle can be demanded to have valence ≤ 11 each and the cycle can be demanded to be chordless.*
2. *One or more of the 633 "reducible configurations" pictured in the 10-page appendix of [93].*[34]

Let us discuss Robertson et al's 633 configurations, and the sense in which they are "contained" in G. A "configuration" C within G is a subset of G's vertices, triangle-faces, and edges, and a function $\gamma(v)$ mapping vertices v of the configuration to integers, such that

1. The configuration's vertices, edges, and triangles form a connected planar graph all of whose faces, except for the single external face, are triangles.
2. Each vertex of C is also a unique vertex of G.
3. Each edge of C is also an edge of G. If any two vertices $a, b \in C$ are adjacent in G, they are also adjacent in C, i.e. C contains the edge ab.
4. Each internal triangle-face of C is also a $\triangle$-face of G.
5. For each vertex $v \in C$ its graphical degree (number of neighbors) in G is $\deg_G(v) = \gamma(v)$, and $\gamma(v) \in \{5,6,7,8,9,10,11\}$. (The vertex of degree 8 in the very last configuration in the appendix of [93] has $\gamma(v) = 11$; other than that every $\gamma(v)$ in every configuration obeys $\gamma(v) \le 10$.) Also $\deg_C(v) \le \gamma(v) \le \deg_C(v) + 3$.
6. Various "Heesch obstacles to reducibility" (certain sub-configurations) do not occur (e.g. see p.222 of [12]).
7. Removing a vertex v of C splits it into at most 2 components; and if exactly 2, then $\gamma(v) = 2 + \deg_C(v)$.
8. If a vertex $v \in C$ is not on the external face, then $\gamma(v) = \deg_C(v) = \deg_G(v)$. Otherwise $\gamma(v) = \deg_G(v) > \deg_C(v)$.
9. $6 \le \sum_{v \in C_o} [\gamma(v) - \deg_C(v) - 1] \le 14$. Here C_o is the subset of C's vertices lying on its external face and not including any vertex whose removal would disconnect C. Consequently the set of G-vertices adjacent to a vertex of C (but not in C) has cardinality≤ 14. This cardinality is called the "ring size" by [93].
10. The number of vertices of C is in $[4, 12]$. The number of vertices on C's external face is in $[4, 10]$. The number of vertices internal to C is in $[0, 2]$.
11. Each one of the 633 configurations has graphical *diameter*≤ 4, i.e. for any pair $\{A, B\}$ of vertices in a configuration, B is reachable from A via a path of ≤ 4 edges, all of which are in that configuration.
12. More strongly, each has graphical *radius*≤ 2, i.e. each configuration contains a vertex X from which any other one of its vertices may be reached via a path of ≤ 2 configuration-edges.
13. Each of the 633 configurations contains at least one vertex with G-valence$= 5$.
14. Each configuration is known to be "D_k-reducible" for some $k \in \{0,1,2,3,4\}$ where k is the cardinality of [93]'s "contract."
15. The fact that the 633 configurations are "unavoidable" was proved [93] via a "discharging argument" in which charge was always pushed between a pair of vertices, each of valence≤ 8, according to 32 "discharging rules."[35] Each such vertex pair was a member of one of 32 "discharging rule" configurations, each of which is a connected planar graph with ≤ 10 vertices, graphical radius≤ 2, and all vertex G-valencies in the set $\{5,6,7,8\}$, and containing at least one vertex with G-valence 5.
16. There is an essentially unique way, called by [93] the "free completion" in which a configuration C can be joined to the neighboring vertices of G, and there are exactly r such vertices where r is that configuration's ring size.
17. C does not contain a nonfacial triangle, nor even one edge of a nonfacial triangle, but there could be a nonfacial triangle abc in G where $a \in C$ and b, c are G-neighbors of a that are not in C.

These facts were extracted from [93][95]and/or by examining their electronic file of the 633 configurations.

7 Our 4-coloring algorithm

7.1 The properties we shall require of the underlying 4-colorability proof

We shall assume in the rest of this paper that a 4-colorability proof with the same properties as [93]'s exists, albeit with the following changes allowed.

1. The number of configurations need not be 633; any constant will do.
2. The maximum number of vertices, the maximum "ring-size," the maximum graphical radius and diameter, and the maximum γ, allowed in a configuration can be any constants, not necessarily the values peculiar to [93]'s proof.
3. The maximum allowed vertex valence in a separating k-cycle ($3 \le k \le 5$) can be any constants, not necessarily the ones in [93]'s proof.

[34] There actually are more than 633 if chirality is taken into account.

[35] It is important to note that these discharging rules are used *once* per edge and *not* reiterated forever.

4. We allow D_k reducibility for $k = 0, 1, 2, 3, 4, \dots, K$ for any constant K. We do not insist, as in [93], that $K \leq 4$, but it is simplest if $K \leq 1$. (Robertson et al [93] stated that they "expected" proofs with $K = \{0, 1\}$ only, and even with $K = 0$ only, would exist but did not bother to find them. They also said they had found a proof involving "about 900" $D_{\leq 1}$-reducible configurations and only a "handful" of D_k-reducible ones with $2 \leq k \leq 4$.)
5. For the $O(V \log V)$-time algorithm in §8 we shall also require the existence of a finite set of discharging rules each involving only bounded-valence vertices, having bounded graphical diameter, and with each discharging rule, and each configuration, involving at least one 5-valent vertex. Furthermore, we shall demand that all these discharging rules have the "make and preserve progress" property that they are only used in a circumstance where at least one of their vertices has a positive charge, and then, after using the rule, at least one of their vertices *still* has a positive charge. (All 32 of the rules pictured in figure 4 of [93] make and preserve progress.)[36]

To keep things simple, we shall often just employ the numbers in [93]; just keep in mind these are not sacred.

7.2 The set of 4-colorings we shall produce

Let Q be the maximum number of vertices in a configuration, and let ν be the maximum G-degree γ of any configuration-vertex. A set of G-vertices adjacent to, or on, a connected subgraph with ($\leq Q$)-vertices all of valency$\leq \nu$, is called **"eligible."** Eligible sets have cardinality bounded by a constant $\leq (\nu + 1)Q$. The number of possible ($\leq$ 4)-colorings of the vertices of an eligible set is bounded by a (large) constant, at most $(4/3) \cdot 3^{(\nu+1)Q}$, i.e. $\hat{O}(1)$ for short. Therefore the number of eligible vertex sets inside G is $\hat{O}(V)$.

We are going to produce a set S of 4-colorings of our maximal planar graph G ([37]) with the following

key invariant inclusiveness property: (3)

> No matter which eligible set s of vertices is chosen, and no matter what ($\leq$ 4)-coloring c of them is chosen (provided it is extendible to some 4-coloring of all of G's vertices), at least one coloring C in S agrees with c on s.

Because there are only $\hat{O}(V)$ possible 2-tuples (s, c), our set of colorings S could in principle be required to have cardinality $\hat{O}(V)$ at most, and conceivably even fewer colorings (perhaps only $\hat{O}(\log V)$ or even $\hat{O}(1)$) are really minimally required, but our data structure shall in fact encode a potentially far larger number – often exponential in V – of colorings.

For cognoscenti: the motive behind the insane-sounding idea of maintaining many colorings rather than just one, is that it allows avoiding what [95] call "Kemping" – a linear-time operation – in favor of constant-time operations.[38] Among the multitude of colorings we maintain, all the already-Kemped ones are already present without any need for us to produce them – via Kemping – ourselves.

If all the N colorings in S were to be printed out in conventional form, the total output might be $O(NV)$ long, which would be unacceptable since we are seeking a linear-time and linear-space algorithm and N can be at least as large as order V. Therefore all these colorings will be simultaneously represented inside an $O(V)$-space data structure, to be described soon. Note that if each coloring were a random string from a 4-letter alphabet, then obviously it would not be possible to compress N colorings into less than order NV memory. The compression only is possible because our colorings are all highly related to each other.[39]

Each eligible vertex-set s, and each possible coloring c of it, will "know" which colorings in S agree with c on s.

7.3 The 4-coloring algorithm

The **4-coloring algorithm** is essentially as follows. But to really understand how it works you will need to know about certain data structures and ideas which shall be discussed later, especially in §7.6.

1. [Input] Input the embedded, plane, maximally-triangulated graph G. Refuse invalid input. (Notes: on recursive calls, step 1 is skipped. A linear-time planarity tester and embedder [60][19] would enable inputting even a non-maximal planar graph, checking its validity, embedding it, and triangulating it, all in $O(V + E + 1)$ time.)

2. [Handle small graphs] If the graph has ≤ 9 vertices then find all ($\leq$ 4)-colorings of it by brute force in constant time. Stop.

3. [Build data structures] Build various data structures described in §7.6.

4. [Handle small-degree vertices] If there exists a vertex v with $\deg(v) \leq 4$ then delete v from G; if the resulting "hole" is not a triangle then identify two of its non-adjacent vertices (two such must exist since K_5 is nonplanar). Remove any redundant duplicated edges this identification creates. (Of course as we make these graph-modifications we update all vertex-degree, degree-list, configuration-list, and edge information, see §7.6.) Now recursively 4-color the resulting smaller embedded plane graph G', and then color v

[36] The fact #3 that either one of the 633 configurations exists, or a separating k-cycle with bounded valencies, follows from the fact that the discharging argument only moves charges among vertices of bounded valence and *ignores* vertices of too-large valence.

[37] Actually, of $G*$, which is a (still planar) graph got by identifying certain pairs of vertices within G; these identifications can help us by reducing the number of allowed 4-colorings; and at intermediate stages, we shall color various subgraphs of $G*$; it is entirely possible to phrase everything purely in terms of G, not $G*$, at an unimportant cost in efficency, with the existence of $G*$ merely used as a mental crutch to see that suitable colorings of G exist, and without the algorithm ever needing to actually construct $G*$; this is done by e.g., simply not performing the k vertex-pair-shrinks in the "contracts" associated with the D_k-reducible configurations.

[38] A "Kempe chain" is a connected subgraph of G induced by vertices colored with some particular 2-element subset of colors. The operation of "Kemping" a Kempe chain K is to interchange those two colors on the vertices of K. Morgenstern & Shapiro [85] empirically found that their Kempe-chains had cardinality $\approx V^{0.6}$, which would force any coloring algorithm utilizing Kemping to help color at least a constant fraction of vertices, to consume at least order $V^{1.6}$ time.

[39] Or unrelated. For example one can store a description of all 2^n strings of n bits in only $O(\log n)$ space.

with a color not used by its neighbors. Stop. Note: finding such a low-degree vertex (or realizing that none exist) requires only constant time by taking advantage of our "degree lists."

5. [Search for configuration or suitable separating k-cycle] Find a configuration C (from among the magic 633 types) inside G. If one does not exist, then the search for that configuration will automatically discover a chordless k-cycle R with $k \in \{3,4,5\}$ containing at least $\lfloor (k-1)/2 \rfloor$ vertices of G in its interior and at least $\lfloor (k-1)/2 \rfloor$ vertices of G in its exterior and each vertex of which has bounded valence. (Indeed [95], one of the 633 possible Cs always will be there, *but* perhaps with some of its vertices identified or joined by an extra edge, causing the k-cycle.) In that case go to step 7. Note: finding such configuration or k-cycle requires only constant time by taking advantage of our "configuration lists" and the boundedness of configuration vertex-valencies and of the configuration's cardinality, see §7.6.

6. [Handle configuration C]. Shrink each edge in the set T of $\le k$ edges (some perhaps artificial) that was pre-tabulated as the "contract" for that configuration [93], i.e. identifying certain vertex-pairs in G, and/or delete C entirely (i.e. delete all its vertices) from G, leaving a hole of perimeter ≤ 14 – whichever the reduction instructions prepackaged with that configuration say to do. Remove any redundant duplicated edges this creates and triangulate by diagonals any nontriangular polygonal faces thus created. (As usual as we make these graph-modifications we update all vertex-degree, degree-list, configuration-list, and edge information, see §7.6.) The result is a smaller embedded fully triangulated plane graph G', with $O(1)$ fewer vertices. Recursively 4-color it.

Now unshrink the $O(1)$ shrunk vertex-pairs and restore the $O(1)$ deleted edges, vertices, and faces. Finally, go to step 8 to handle coloring the $O(1)$ uncolored vertices and associated updating.

7. [Handle separating k-cycle R]. Produce two smaller graphs G' and G'' consisting of the vertices inside-or-on, and outside-or-on, the cycle R. If $k=4$ the other side of the cycle should be triangulated by adjoining a diagonal ab so that the recursive call may be done on fully-triangulated graphs; the same diagonal must be added in both cases and a and b must be nonadjacent in G to allow this (and some nonadjacency must exist since K_5 is nonplanar). If $k=5$ then we triangulate the other side of the cycle by adjoining an artificial extra vertex adjacent to every vertex on the cycle. Note: when producing G' and G'' we do *not* perform a graph exploration (which would take linear time[103][40]), but instead simply label the cut edges as "cut" and pass the old graph (and a start vertex) in situ in $O(1)$ time. Recursively 4-color them. Also note: when we match up the colorings of G' and G'' on R by taking advantage of our key fact that, e.g., a G' coloring is *already* available that extends *every* extendible coloring of R alone. Therefore there is *no need* for us to recolor G' by permuting the color names, in order to be able to match it to G''. Finally, go to step 8 to handle (in $\hat{O}(1)$ time) the updating of the many-colorings data structure for eligible sets in the neighborhood of R.

8. [Coloring of uncolored vertices and associated updating]. We now have a partial coloring of G, i.e. G is fully colored except on a certain known connected set K of $O(1)$ uncolored vertices, each of bounded valency. Consider every possible coloring of them (there are only a constant number of possibilities to consider at most). For each possible eligible vertex-subset s intersecting or neighboring K (there are at most a constant number of such sets) and for each possible coloring c of it (of which there are only a constant number at most) create a link to all colorings of G from among those we already have from the recursive call (or two) that agree with c on s for each c compatible with the coloring of K currently under consideration. To avoid creating order-V or more links here (we only shall create $O(1)$) make each such link really consist of only $O(1)$ links to previously constructed and stored link lists associated with colorings of node-subsets of s which agree with the old set S' of colorings on the old smaller graph G' (or S' and S'' on the two smaller graphs G', G''). See §7.6 for more precise and complete discussion of the coloring-containing data structure and how to update it. Stop.

9. There is a final stage of the algorithm, related to producing the output, discussed in §7.6.

Note that the total number of eligible-vertex-sets s produced in the course of this algorithm, including all recursions on all smaller graphs, is $O(V)$ since indeed $O(1)$ new ones are produced each pass. Therefore, the total size of the data structure storing all the colorings is $O(V)$.

7.4 The two phases of the algorithm

The algorithm may be regarded as **(1)** first "shrinking" the graph by deleting vertices until a constant-size graph is (or several disjoint constant-size graphs are) reached, which may be colored trivially. These colorings obviously will fit in $O(V)$ memory space. Then **(2)** it "grows" the graph back, coloring vertices as they are added.

The first phase of the algorithm is actually plausibly practical[41] because in a *fully triangulated* planar graph, searching for a configuration centered at a given (bounded valence) vertex is easy – *no* "backtracking" is required and the worst-case runtime is linear in the size of the configuration times the valence of the start-vertex. Our first stage may be thought of as producing an *ordering* of the vertices of G (reverse of the chronological ordering of their deletion), which will be colored, in that order, by the second stage. Outrageously large constant factors in the space and time bounds occur only in the second stage. The second stage colors each vertex, in order, *once* and thereafter never "backtracks" to recolor it. (Or, alternatively, we can regard some backtracks as happening, but we never backtrack more than a constant distance in the vertex ordering, i.e. once vertex number $v+\kappa$ is colored,

[40] One could also do the G' and G'' graph production and adjustment in conventional format in $O(\min\{|G'|,|G''|\})$ time, via graph explorations of the small side, where $|X|$ denotes the memory-size required to represent X, i.e. in our case $|G'| = 1+E'+V'$. However, this could lead in the worst case to superlinear runtime of order $V \log V$, because the solution of the recurrence $T(N) = T(A)+T(B)+\min(A,B)$ with $A+B=N$ and $A,B>0$ is $T(N) \approx N \log N$ if $A=B=N/2$ every time. To assure a linear runtime bound we need a recurrence like $T(N) = T(A)+T(B)+c\min(A,B)^p$ for some constants c, p with $0 \le p < 1$ and $c > 0$. By the no-recoloring and no-exploration tricks we achieve this, indeed with $p=0$.

[41] But a fairly determined programmer would be required, since all 633 configurations from [93] are involved...

the colors of vertices numbered $\leq v$ are permanently frozen, where κ is some constant.)

7.5 How to build a practical compromise algorithm

If we *omit* our second phase and instead simply employ a coloring algorithm like Morgenstern and Shapiro's [85] *using* the vertex ordering found in the first phase, then we get a *practically-implementable* hybrid algorithm which (1) should exhibit the same excellent empirical performance as the methods of [85], (2) runs in $O(V^2)$ worst-case time and (3) consumes $O(V)$ worst-case space (with only a small constant hidden in this O). This is because we *know* that with our magic vertex ordering, 4-coloring occurs in such a way that any "impasse" (attempt to color a vertex which already has neighbors of all 4 colors) is resolvable purely by local recoloring and local Kemping. These are precisely the methods [85] used to break impasses. They had no theoretical guarantee that their impasse-breaking attempts would succeed.[42] But we (with the right choice of limits on the local-search size and the number of Kemping attempts) do: our hybrid method will *always* succeed in breaking any impasse after at most a constant number of local recoloring and local Kemping steps.[43] Each Kempe could conceivably take linear time in which case the total runtime would be $O(V^2)$. However [85] empirically found that the average Kempe only took $O(V^{0.6})$ time (in which case our runtime would be $O(V^{1.6})$) and also found that by setting a high enough cutoff on their backtrack- and "wandering 5th color" based local recolorers they only needed to resort to Kemping very rarely. Hence they, empirically, achieved near-linear runtime. We too would expect that typical behavior, with the only major difference being that we have a $\hat{O}(V^2)$ worst-case runtime bound.

7.6 The data structures (and other minor tricks) we need to make it run in linear time

First of all, we employ all the data stuctures used in the proof of theorem 4. We also employ:

Vertex self-knowledge: Each G-vertex knows its degree d, and for each $i = 1, 2, \ldots, 633$, knows whether it is the centerpoint of a configuration of type i.

Graph representation: Each vertex has a doubly-linked list of its neighbors in clockwise[44] order (and A's pointer to neighbor B points not only to B, but in fact to where in B's clockwise order A lies). Or, equivalently, the elegant "winged edge" data structure of Guibas & Stolfi [50] could be employed.

Hashed edge-table: Each edge ab is stored in a dynamic hash table [23][29] so that vertex-pair adjacency testing may be done in $O(1)$ steps. Edges may be added or deleted from this table in $O(1)$ expected time and each hash table entry points to the location of that edge in its endpoints' clockwise angular lists.

Degree-lists. There are 4 "degree-lists," list i containing all the vertices known to have degree $i+2$ for $i = 1, 2, 3, 4$ (but all degrees ≥ 6 are lumped into the single list #4).

Configuration-lists. There are 634 "configuration-lists," list i containing all the vertices known to be the centerpoint of a configuration of type i, for $i = 1, 2, \ldots, 633$ (and list #0 contains all the remaining vertices).

Short chordless cycles. Each vertex knows if it is a member of a chordless separating k-cycle with $3 \leq k \leq 5$ with all vertices of the cycle having valence ≤ 11; and for each such k there is a list of all such vertices.

All of of the above may be updated, as the graph is modified via vertex-deletions or additions of bounded-valence vertices, or adding or deleting an edge ab between two vertices (*provided* the position of that edge in a's and b's clockwise orders are known), in $O(1)$ total extra time. That is because of the bounded graphical radius of, and bounded valencies of all the vertices in, our configurations and short-cycles; this causes the total exploration outward from a graph-modification (to visit everything it affects) to be possible in $O(1)$ steps.

Quick removal of duplicated edges: If an edge ab is shrunk to a single node, then all redundant duplicated graph edges created by this shrinking may be removed in $O(1)$ time *provided* one of more among $\{a, b\}$ (say a) had bounded valence. That is because the duplicated edges arise from common neighbors of a and b and there are at most $\deg(a)$ of them; we examine them and for each enquire (using the hash table) if it is a neighbor of b. In particular, note that all the "contract" edges [93] in configurations C have at least one endpoint in C and hence have bounded valency, so their shrinkage is $O(1)$ time.

Quick removal and re-insertion of configurations: All the vertices in a configuration C have bounded valency, and there are a bounded number of them, so the entire C can be removed in $O(1)$ time. Note that the graph vertices neighboring C (of which there are a bounded number) might include some of unboundedly high valency, but because we know the location of their edges to C in their cyclic adjacency lists, their information can still be updated in $O(1)$ time.

If we desire to retriangulate (by diagonals) the "hole" (of bounded perimeter) left by some such removal, then that too may be done in $O(1)$ steps and note that adjacency testing of all pairs of hole-perimeter vertices may be done in $O(1)$ time with the aid of the hash table.

A precomputed list of all 633 configuration-types and their "contracts" and "centerpoints." Downloadable from [93].

A precomputed list of all the eligible sets we shall ever need in all the graphs (and partial graphs) we get by adding in the vertices, in phase 1's output-order, one at a time. May be found in $\hat{O}(V)$ time and space after phase 1 but before starting phase 2. Should be stored in the form of a trie to permit access to any s-element eligible set in $O(s)$

[42] And hence the possibility remained open that algorithms of their sort might have $\leq C^V$ success probability for some constant C with $0 < C < 1$, or $\geq C^V$ expected runtime for some constant $C > 1$.

[43] Undoubtably algorithm "tuning" by adjusting the sizes of the local-search and Kemping cutoffs, and perhaps having several increasing layers of such cutoffs, would yield better performance in practice.

[44] If this order is not known then we regard the embedding as not being known.

time. Each trie node conatins an $O(V)$-size pointer array but since there are $\hat{O}(1)$ eligible sets and hence trie nodes the whole trie consumes $\hat{O}(V)$ space.

The coloring-containing data structure.[45] At the start of phase 2, the graph has only a constant number of vertices and the "data structure" is merely an $\hat{O}(1)$-size list of every possible 4-coloring, found by brute force. (If there are several disjoint constant-size graphs, these lists could take $\hat{O}(V)$ space, $\hat{O}(1)$ space per graph.) As phase 2 proceeds, we add more vertices to the graph. At any moment each new eligible set s in the current graph (which was not eligible in the older, smaller graph a constant number of vertex-additions ago) is made to have a list of all possible colorings c of it alone, each with a list of *pointers* that point back to all possible colorings of the older graph that are compatible with c. After adding a bounded-by-a-constant number of new vertices, the number of new s and c is bounded by a constant. The number of pointers that we need to create per (s, c) two-tuple is also bounded by a *constant* because we do *not* create one pointer per coloring of the whole previous graph; each pointer really encodes a possibly very large set of colorings described via a pointer to the most recent among the older eligible sets which are a maximum-cardinality subset of (s, c). Also each pointer comes with a full list of the extra (color,vertex) pairs that are being adjoined to the old coloring (this list has $\hat{O}(1)$ entries) and a *count* of how many colorings it concerns[46] (i.e. the number it is pointing back to, times the number of new extensions). Also, whenever we insert a pointer we also create a back-pointer leading in the other direction (to allow later bidirectional exploration of the data structure we are creating).

Incidentally, mental cleanliness seems aided if we agree not only to add pointers back to compatible colorings of the old graph, but *also* to *in*compatible ones (but these pointers should be labeled with a special flag bit saying "do not use this pointer, it indicates an incompatibility").

As we proceed through phase 2, we do not actually compute the counts (except in the cases of the brute force colorings of the initial graphs, for which we know the exact counts, all of whose values are 1 for each coloring); we merely leave the count-fields in the data structure *blank* or, if it is discovered that some coloring c of some eligible set s is disallowed since it leads to an incompatibility with all previous colorings, then we overwrite its count field with *zero*.

Final stage of algorithm. After phase 2 is complete, we now do a linear-time re-exploration of our $O(V)$-size data structure, this time filling in the count fields. This consists of, first, a backwards pass filling in zeros in count fields all of whose chronological successors have zeros. I.e. if, in some eligible subset there are backpointers to the present field, and all those backpointers have the "incompatible" flag set, then overwrite the present field count with zero. Then, second, we do a forward pass filling in each count field with a true count (or at least, a valid lower bound which is nonzero whenever the true count is nonzero). These counts may be filled in from knowledge of their predecessor count-fields (and their zero-entries).

The *final* count field (corresponding to the chronologically-last eligible vertex set) is the one that matters most, since it tells us how many valid 4-colorings of the graph we have found. Due to the validity of the 4-color theorem and its proof via the 633 configurations and associated reduction rules, this final count will be ≥ 1. Because plenty of planar graphs exist whose 4-colorings are *unique* (up to color-renaming) it could be only $O(1)$. But more often it will be some exponentially-large number of order C^V for some C with $1 \leq C \leq 2$.

Note: this final count will *not* necessarily be equal to the number of 4-colorings of the graph, e.g. because the underlying 4-coloring proof-via-reductions is not claimed to generate every possible 4-coloring, merely ≥ 1 valid colorings. So our count will only be a lower bound on the true count – but usually a large lower bound. Incidentally, the counts could easily become so large that multiprecision arithmetic would be required. To make the algorithm run in linear time with $O(\log(1 + V + E))$-bit-wide words of memory we therefore would recommend representing all sufficiently large counts via approximate floating-point arithmetic, with relative error $\leq 0.01/(1 + V + E)$, say.

Finally, it is a simple matter to extract some valid colorings from the data structure at the end by exploring through the back-pointers starting from some positive-count on a final eligible set, outputting vertex colors from each set visited during the backward exploration. This takes $\hat{O}(V)$ time.

Theorem 10 (Main result). *There is a $O(V+E+1)$-space and expected-time randomized algorithm, running on an integer RAM with $O(\log(2+E+V))$-bit wide words and unit-time arithmetic operations, to find a 4-coloring of a planar graph. Indeed it will produce a $\hat{O}(V)$-space data structure representing a set S of 4-colorings of the graph which obeys property (3).*

Proof: The considerations above have shown that the algorthm runs in $O(V + E + 1)$-space and expected time. The randomization is used only for hashing [38][23][29][47] and any future invention of a deterministic hashing scheme would derandomize it.[48] Also, one could derandomize it at the cost of introducing an extra factor of $O(\log(2 + E + V))$ into the

[45] The following toy example may be helpful for those who just want to understand how it can be logically possible for an $O(N)$-size data structure to store inside it, an exponential number of colorings. Consider 3-coloring the N-vertex path graph. It has $3 \cdot 2^{N-1}$ colorings, but we can represent them all in $O(N)$ space! In fact walk down the path. At the nth node say "the colorings of the n nodes so far are: color this node 1 and adjoin it to all colorings of the previous stuff that have the last node colored 2 or 3. Or color this node 2 and adjoin... 1 or 3. Or ... 3 ... and adjoin... 1 or 2." Writing these sentences requires $O(1)$ space. So the total mass of all the sentences ever written requires $O(N)$ space. Each sentence may be thought of as consisting of 2 back-pointers to colorings in the previously built part of the data structure. The full data structure we are proposing in this paper is similar except that: all pointers are bidirectional; we add not one, but a bounded-size chunk, of new vertices each step; we backpoint not merely to a set of colorings specified by statements like "the colorings of $1, \ldots, n-1$ which have color 1 on vertex $n-1$" but instead to a set of colorings specified by statements like "the colorings of $1, \ldots, n-1$ which have colors 1,3,2,3,3 on vertices 77, 87, 21, 22, 23 respectively."

[46] Or merely a lower bound on such a count suffices, provided it is nonzero whenever the true count is.

[47] E.g. for the purpose of fast adjacency testing in the 4-valent vertex case in step 4 or during duplicated-edge removal.

[48] Probably a slightly better underlying 4-colorability proof could be devised which would eliminate the need for hashing. I suspect one can prove that one of the 633 configurations have to arise even if 4-valent and 3-valent vertices are *permitted*, provided these 4- and 3-valent vertices have at least one sufficiently high-valent neighbor. If such a strengthening of the 4-colorability proof were established, then we would no longer need to

runtime bounds, by simply replacing all the hash tables by AVL trees.

The only thing to prove, then, is the algorithm's correctness. That follows from (1) the (assumed) correctness of the underlying 4-colorability proof [93] and (2) the fact that the key property (3) is obviously true at the beginning of phase 2, and inductively and by design continues to be correct as the algorithm proceeds. □

8 A simpler $O(V \log V)$-expected time randomized 4-coloring method

This section will present a fairly simple $O(V \log V)$-expected time $O(V+E)$-space randomized 4-coloring method for planar graphs. The ideas behind it trace back to two obsolete 5-coloring methods [18][77]. Although its running time bound is asymptotically worse (by a $O(\log V)$ factor) than the linear-time 4-coloring method, the linear time method is almost certainly impractical whereas this method is probably practical (although those are unanswered experimental questions).

That is because

1. The $O(V \log V)$ method involves only one coloring, not a data structure encoding many colorings, hence only has a small constant in its space bound,
2. There is reason to suspect it may also have a small constant in its expected time bound. That hinges on the expected number of random Kempes that need to be done before a configuration becomes 4-colorable. This quantity could have been (but was not) computed by the same computer that brought us the 4-colorability proof [93], and quite plausibly for most or all of the 633 configurations, it is reasonably small.[49] (If this is true, then the $O(V^2)$-time method in §7.5 also will have a small constant.)

Also, our $O(V \log V)$-time method may be heavily parallelized (it is in NC) whereas the linear-time method seems inherently sequential.

Obstacles to parallelization. Why is it that finding a 5-coloring of a planar graph is a highly parallelizable task [52], whereas our 4-colorer above seems inherently sequential? The underlying problem is that a huge planar graph might contain only $O(1)$ instances of any of the magic 633 reducible configurations. Any 4-coloring algorithm based on removing reducible configurations could therefore only remove one (or at most a few) at a time. The following lemmas are designed to remove that obstacle.

Lemma 11 (Shield existence and non-rarity). *A fully triangulated V-vertex planar graph either*

1. *contains at least order $V/\log^2 V$ disjoint instances of the 633 magic configurations and/or chordless separating k-rings ($k \le 5$) with bounded vertex valencies, i.e. the "rarity" of these things is "at most logarithmic" within the graph,* or
2. *at least a constant fraction of the vertices of the graph are members of disjoint "n-shields," meaning a regular-hexagon-shaped chunk of the equilateral-triangle lattice with sidelength $n \ge 3$ and n odd (see figure; these n-shields are allowed to have different values of n),*
3. *or both.*

Proof: Consider the fact [93] that a reducible configuration (where we here are counting the k-rings and (≤ 4)-valent vertices among them) is entirely contained within distance≤ 2 (distances measured along paths of bounded-valence vertices only) of each "positively charged" vertex after the 32 "discharging[50] rules" of figure 4 of [93] have been applied. Consider also that all vertices of valence k initially have "charge" $120(6-k)$ and that all the charge-motion rules involve pushing charge between ≤ 8-*valent* vertices that lie within distance≤ 2 of a 5-valent vertex (distance measured along paths containing ≤ 8-valent vertices only). Consider also that there are only a finite number (32 in fact) of discharging rules and each one moves ≤ 2 units of charge a distance ≤ 1 and each rule is applied only once per edge, and with an arbitrary chronological order among the edges. Finally, consider the fact that each discharging rule "makes or preserves progress" in the sense that it only does anything if there is a positively charged vertex included inside its pattern, and after it does it, there *still* is a positively charged vertex included inside its pattern. From all these facts we see that we can apply the discharging rules in an order such that

1. Ultimately, all positively charged vertices are at distance≤ 2 from a 5-valent vertex, and
2. The expected distance a positive charge moves, if all (edge,rule) 2-tuples are utilized in random[51] order, is $O(1)$, and indeed there exists[52] some ordering causing *every* charge to move a distance $\le O(\log E/\log\log E)$, with *almost all*[53] of them traveling only a distance $O(1)$. All these "distances" are measured along paths consisting of bounded-valence vertices only.

test adjacency of neighbors of removed 4-valent vertices – except when those neighbors had low valence, in which case we would have enough speed even without hashing. Also, we perhaps could eliminate the need for hashing when getting rid of (or otherwise worrying about) duplicated edges, by simply *permitting* duplicated edges. With no hash tables, the algorithm would then be deterministic. It would still require a RAM because it uses tries.

[49] Indeed, in the Appel-Haken proof, configurations were abandoned (even if they might be reducible) if they were not "easy" to show reducible. This tended to cause only configurations for which only a small expected number of random Kempes were needed, to predominate.

[50] A better name would have been "charge motion."

[51] To see this, it will help to consider the following toy problem Suppose there are E spots, one of which is You. A random ordering of them is chosen from the $E!$ possibilities but kept secret from you. A blacksmith now goes through the spots in order, hitting them with his mallet. If you are hit, you move to another dot. What is the expected total number of times T you are hit? The probability P_h you are hit $\ge h$ times obeys $P_h \le P_{h-1}/h$ so that $P_h \le 1/h!$. Therefore the expected number E_h of hits you suffer tends to $e \approx 2.71828$ as $E \to \infty$ and the tail probability drops superexponentially.

[52] Since our probability argument shows the probability is positive that a random ordering accomplishes this. It is completely irrelevant how hard it is algorithmically to find such an ordering, since nobody is every going to find it; we only care that some such ordering exists so that we know the positive charges need not move very far form their initial locations at ≤ 5-valent vertices.

[53] That is, a fraction which may be made arbitrarily near to 100% by making the "$O(1)$" large enough.

Therefore (considering the expected-distance and alomost-all distances result; the logarithmic maximum-distance result does not matter for this) there are two possibilities. Either vertices of valence≤ 5 are common (are at least some positive constant fraction of all vertices), in which case reducible configurations are common (at least cV exist for some constant $c > 0$) or ≤ 5-valent vertices are rare.

We conclude that reducible configurations can only be rare if vertices with valency ≤ 5 are rare, and that in turn can only happen (in view of Euler's formula) if vertices of valency≥ 7 also are rare. More strongly, edges AB with either A or B not 6-valent, must be rare, constituting < 0.001 of all edges. Consequently vertices of valency$= 6$ dominate (e.g. $> 0.999V$ are 6-valent). That in turn forces at least a nonzero constant fraction of the vertices to lie inside n-shields with $n \ge 3$ and n odd. That is because maximal planar graphs in which all vertices have valency 6 are "locally unique." Specifically, start at at some 6-valent vertex. Consider the 6 triangle-faces involving it. Now consider the 6 other vertices on those faces, and the 6 triangles ringing them. and so on. In this way we see that every 6-valent vertex of a maximal planar graph, which is graphical distance$\ge n$ away from any vertex of valency$\ne 6$, must be the center of an n-shield. Because the edges of valency$\ne (6,6)$ are rare, n-shields must be common (occur at least a constant fraction of the time). ☐

Lemma 12 (Shield reducibility). *If the external vertices (lying on the hexagon boundary) of an n-shield ($n \ge 3$ and odd) are 3-colored in any manner, then the interior vertices may be compatibly 4-colored by a linear time algorithm.*

Proof sketch. Induction on n. The claim is obvious when $n = 1$, and may be proven by exhaustive enumeration when $n = 3$; these form the basis for the induction. There is a linear time algorithm to decrease n to $n - 2$ by coloring the "outer two layers" with the inner of these 2 layers being 3-colored, as follows: first 4-color the outer layer using the 4th color at its corners and at positions even distances along the hexagon sides joining those corners. Second, 3-color the inner layer. ☐

We now present **Our randomized 4-coloring algorithm:**

1. [Input] Input the embedded, plane, maximally-triangulated graph G. Refuse invalid input [89][8].

2. [Handle small graphs] If the graph has < 300 vertices then color it by brute force. Stop.

3. [Find reducible things] Find a set of vertex-disjoint reducible things comprising, in toto, at least a fraction $c > 0$ of all the vertices in the graph. The "things" can be either: (≤ 4)-valent vertices, the 633 magic reducible configurations, chordless separating k-rings for $3 \le k \le 5$, or n-shields with n odd and $n \ge 3$. For maximum parallelizability, we must impose an upper bound on n, say $3 \le n \le 9$. However, if we only plan to use $O(\sqrt{V})$ processors, no upper bound on n needs to be imposed, since the shield-coloring algorithm in lemma 12 can be parallelized with n processors.

4. [Reduce] Remove all the interior vertices of each n-shield, replacing them with just a single vertex attached to all the external vertices. Perform the usual reductions on the other cases, e.g. omitting vertices of reducible configurations. (See [93] near the end for the method of handling the k-rings.) The result is a maximal planar graph (or several disjoint such graphs) with $\le cV$ vertices, for some absolute constant c with $0 < c < 1$.

5. [Recursive color] Recursively 4-color this smaller graph (or graphs).

6. [Partial restoration I] Restore all the omitted vertices inside n-shields and 4-color them using lemma 12.

7. [Partial restoration II] Restore all the omitted vertices inside the magic 633 reducible configurations.

8. [Randomized Kemping] The properties of "reducibility" are such that it may not be possible to extend the coloring from the ring of ≤ 14 precolored vertices surrounding a configuration into the configuration itself. However, it is always possible to do that after we "Kempe" some subset of the Kempe chains containing outer-ring vertices; at most a constant number of Kempings are needed to recolor the graph in such a way that any particular configuration becomes 4-colorable. Because there are at most 4 colors and at most 14 ring-vertices, there are at most a constant number of possible types of Kemping. (See footnote 38 for Kempe definitions.) So if we were to choose randomly among all Kempe possibilities, then we would expect at least some positive constant fraction of the configurations to become colorable. So, this is precisely what we shall do: Call the following be an "i, j-randomized Kempe:" We choose a color-pair (i, j) from among the $\binom{4}{2} = 6$ possibilities, find all the connected components in our graph whose vertices are colored i and j, and for each such connected component we, based on flipping a coin (one independent coin flip per component), randomly decide whether to interchange colors $i \leftrightarrow j$ within it. We do (i, j)-randomized Kempes (choosing the (i, j) also randomly) a constant number of times; each one can be done in $O(V + E)$ time and space.

9. [Coloring] Color the configurations which now are 4-colorable. In expectation, at least some constant fraction $C > 0$ of them are 4-colorable. Because at most 100% are colorable, we therefore conclude that with probability $\ge C/2$, at least a fraction $C/2$ are colorable. (As a practical programming matter, it probably would be better to intermingle steps 8 and 9.)

10. [Loop back] Go back to step 8 until all configurations are colored. The expected number of loopbacks is $O(\log V)$.

11. [Final restoration] Restore (mainly via gluing together) all the partial graphs resulting from the cutting along separating k-cycles, renaming colors in each subgraph as necessary to make this legitimate. Note: the properties of the k-ring reduction are such that no Kemping is required. Also note: when we color G' and G'' we use a V-entry "color indirection array." That is, the 4 colors used for G' are, say, `col[1]`, `col[2]`, `col[3]`, and `col[4]` and the 4 colors used for G'' are, say, `col[5]`, `col[6]`, `col[7]`, and `col[8]`. Finally we permute the colors within G' so that G' and G'' match on R. This *graph-wide* recoloring may be done in $O(1)$ time by simply writing appropriate data into `col[1]`, `col[2]`, `col[3]`, and `col[4]`. These color-indirection arrays can in fact be the "set-union/find" linear-space data structure analysed by Tarjan [104][99], where each unification across a k-ring causes, e.g. the vertex-set represented by `col[3]` and `col[7]` to become unified. This data structure introduces an time-overhead of $\alpha(V + E)$ where α is an inverse Ackermann function, but this is swamped by the $\log V$ overhead we have elsewhere and hence does not affect the runtime bound. Stop.

Remarks. Because the configurations have bounded graphical radius and consist solely of bounded degree vertices, and because of the parallel algorithms for finding maximal independent sets [46], in step 3 we can find a set of disjoint reducible configurations of at least constant density in $O(\log^* V)$ time by V processors, or in $O(V)$ time sequentially.

Because the configuration set is dense, each recursion colors a graph with $\le cV$ vertices for some constant $c > 0$ and therefore at most $O(\log V)$ levels of recursion are needed.

In step 1, planarity checking (and embedding, if desired!) may be done in $O(\log V)$ expected steps with $O((E+V)/\log V)$ processors via combining [89][11][42][54].

In step 8, all the connected components of the i,j-colored subgraph may be found in $O(V+E)$ sequential time and space, or $O(\log(V+E+2)$ parallel time using $O((E+V)/\log V)$ processors [42]. (The connected components algorithm has even been redone for an EREW PRAM [54] and derandomized [27] although the latter at the cost of introducing an extra $O(\log\log N)$ factor into the runtime.)

The total expected sequential runtime is $O(V \log V)$ with the log factor coming solely from the expected number of loopbacks in step 10.

A tricky part of parallelizing the algorithm is dealing with the separating k-rings. The trouble is that coloring the graph on one side needs to be done *before* coloring the graph on the other (since the second graph needs to be modified in a way *dependent* on the coloring of the first graph ([93], [14], [87] theorem 12.1.5). This by itself is ok (it only multiplies the parallel runtime by a factor of 2), but if there are many separating k-rings, parallelism might be destroyed. The worst case is where the splittings conceptually form a balanced binary tree, in which case the parallel runtime would be at least proportional to the number of leaves of that tree, which could be as large as V^c for some constant c with $0 < c < 1$. A valid bound on the total parallel runtime, using $O(1+V+E)$ processors, is thus $O(R + (\log V)^2)$ where $R = O(V)$ is the total number of nontrivial separating k-rings employed during the reduction process.

Although the R term can in principle destroy parallelizability, in practice in many situations, R will be small, and furthermore with an additional trick each recursive level ℓ of the algorithm may be implemented to run in $O(\log R_\ell + \log V)$ steps. So in practice there should usually be substantial parallelizability.

But if this is viewed as insufficient, let us hasten to reassure those theorists, completely unconnected to reality, who enjoy the class "NC" of ultraparallelizable algorithms, that planar 4-coloring is in (randomized) NC.[54] The reason is that we can deal with a k-ring by simply doing *all* the possible graph modifications (and colorings of the modified graph) in parallel on the smaller side of the separating ring. Letting $P(N)$ denote the number of processors for a size-N problem, we then have the recurrence $P(N) \le P(A) + cP(B)$ where $A \ge B \ge 0$, $A + B = N/k$ for some constants $k > 1$ and $c > 1$. The solution of this recurrence is $P(N) = O(N^{\log_k(c/2+1)})$. Thus, a polynomial($N$) number of processors suffice to 4-color a planar graph in expected time $O((\log N)^2)$.

Theorem 13 (Practical 4-coloring). *There is a randomized algorithm to 4-color planar graphs in $O(V \log V)$ sequential expected time and $O(V+E)$ space. Using a polynomial (in V and E) number of processors, a 4-coloring may be computed in $O((\log V)^2)$ expected parallel time, and §8.1 will show how to accomplish that without any randomization.*

8.1 Derandomization to show planar 4-coloring is in deterministic NC

Here is a useful way to generate a deterministic replacement for N random bits. To get more generality we shall consider not merely bits (mod 2), but instead trits (mod 3) or more generally "pits" – integers modulo any fixed prime p.

Start with some $k \times N$ matrix of full-rank modulo p. For a concrete example, a Vandermonde matrix with ab entry a^{b-1} mod p will do.

Quick review of linear algebra: The "row rank" of a rectangular matrix M over a *field* is the maximum number of rows of M which are linearly independent. It is a theorem that the row and column ranks are equal. An $r \times c$ matrix has "full rank" if its rank is $\min(r,c)$. A square Vandermonde matrix with ab entry x_a^{b-1} has determinant $\prod_{i>j}(x_i - x_j)$ which is nonzero if the x_i are distinct, which forces our proposed matrix to have full rank if $p > \max(k,N)$.

Now consider the p^k possible linear combinations (operating mod p) of its k rows. The result is p^k different chunks of N bits.

Lemma 14 (k-wise independence). *These pits have the property that if any k-element subset of the N pit-positions is chosen, then our p^k chunks yield, in those k locations alone, all p^k possibilities.*

Proof: If they yielded fewer then there would be some nonzero linear combination of the rows of our $k \times k$ matrix which would be 0 mod p. Hence the row-rank of this matrix would be $< k$. Hence, the original $N \times k$ matrix could not have had full rank, contrary to assumption. □

By using these bits (with a large enough value for the constant k) we can assure that every possible bounded-size configuration sees every possible Kemping and hence assure that it will be successfully colored.

Unfortunately this converts an N-bit randomized algorithm into a $p^k N$-bit deterministic algorithm with p of order N, i.e. what was a linear-time algorithm ($O(N)$) now has a large polynomial runtime $O(N^{k+1})$. That is of no practical use. However – this *is* good enough for the theoretical purpose of proving that 4-coloring is in the "deterministic NC" computational complexity class.

9 Simple linear-time 4-coloring?

In this section, we sketch a simple algorithm that, given a V-vertex, E-edge planar graph and an integer $k \ge 1$, will run

[54] That is: there is an NC algorithm which inputs the planar graph and $O(V+E)$ random bits, and with probability$> 3/4$ outputs a 4-coloring. In §8.1 we shall see how to replace these $O(V+E)$ random bits by $(V+E)^{O(1)}$ deterministic bits to get probability=1 of outputting a 4-coloring.

in $O(1+E+16^k kV)$ steps and $O(V+E+4^k)$ memory locations, and will output a 5-coloring of the graph that uses the 5th color on $\le V/k$ of the vertices. (Here the 5th color is a special "wild-card" color: two vertices with color 5 are allowed to be adjacent.) Our algorithm is an apparent improvement of one invented by Brenda Baker [9].

Conjecture 15 (Simple 4-colorer). *If k is sufficiently large ($k \ge 8$) then the coloring output by our "improved-Baker" algorithm below will be a 4-coloring.*

If this conjecture is correct, then this linear-time 4-coloring algorithm should render much of the rest of this paper (and perhaps indeed of 4-coloring theory generally) obsolete.

Essentially the same algorithm will find an independent set in the planar graph of the maximum possible cardinality except possibly for $\le V/k$ missing vertices, in $O(1+E+4^k kV)$ steps and $O(V+E+2^k)$ memory locations. We similarly *conjecture* that if k is any sufficiently large constant (to be concrete, if $k \ge 8$) then the independent set it outputs will have cardinality $\ge V/4$.

Baker defines the graph induced by the vertices of the outer face of a planar graph G to be *outerplanar*. If these vertices are deleted, then the graph induced by the vertices of the outer face of the new planar graph also is outerplanar, and the graph induced by the union of these two vertex sets is *2-outerplanar*. And so on: if we remove the vertices of the outer face k times successively, then the subgraph of G induced by these vertices is *k-outerplanar*.

Baker observed that it is possible to find a maximum-cardinality independent set in a V-vertex k-outerplanar graph in $O(1+4^k kV)$ steps and $O(V+2^k)$ memory locations by "dynamic programming."[55] Furthermore, it similarly is possible to find a c-coloring, or prove one does not exist, or more generally can find a $(c+d)$-coloring that uses the last d colors the minimum possible number of times, in $O(1+(c+d)^{2k} kV)$ steps and $O(V+(c+d)^k)$ memory locations. Other well known NP-hard problems also are soluble in linear time when the graph is k-outerplanar.

Baker's proposed coloring algorithm,[56] then, would be to choose a number $i \in \{0,1,2,\dots,k-1\}$ and delete all the vertices of the original planar graph G in layer-ℓ inward, for each ℓ congruent to i mod k ([9] page 158). She would then color all the resulting $(k-1)$-outerplanar graphs optimally and finally the deleted vertices would be colored with an *additional* 3 colors (note: an outerplanar graph has a 3-coloring which may easily be found in linear time). Since every k-outerplanar graph could be 4-colored, Baker would thus find a 7-coloring of the full planar graph. By using the best value of i (found by trying all possibilities) the number of vertices colored with colors 5-7 would be assured to be $\le V/k$.

Now let us describe our improvement of Baker's algorithm. We observe that it is possible to find a maximum-cardinality independent set I in a V-vertex $(k-1)$-outerplanar graph *in which the outermost layer's vertices' memberships (and non-memberships) in I are pre-specified* in $O(1+2^{2k} kV)$ steps and $O(V+2^k)$ memory locations by dynamic programming. Furthermore, we can find a c-coloring, or prove one does not exist, or more generally can find a $(c+d)$-coloring that uses the last d colors the minimum possible number of times, in a V-vertex $(k-1)$-outerplanar graph *in which the outermost layer's vertices' colors are pre-specified* in $O(1+(c+d)^{2k} kV)$ steps and $O(V+(c+d)^k)$ memory locations – and only $O(1+c^{2k} kV)$ steps and $O(V+c^k)$ memory locations if we agree simply to fail whenever we cannot find a c-coloring.

Our improved-Baker coloring algorithm optimally colors the $(k-1)$-outerplanar graph consisting of the outer k layers of G. Assume G has L layers total, i.e. is L-outerplanar. For $i = 1,2,\dots,L$ we then optimally color[57] the k-outerplanar graph consisting of layers i, $i+1$,... $i+k-1$ of G *in which the (outermost) layer i's vertices' colors are pre-specified to agree with those got in the previous for-loop iteration.*

This algorithm presumably produces a coloring at least as good as some coloring output by Baker's algorithm, because any k-outerplanar it colors, it colors optimally, whereas Baker's coloring of the same k-outerplanar might be nonoptimal (using extra colors more times than we do). Unfortunately that presumption, while very likely, is unproven because it is possible that, even though we do at least as well as Baker on any particular k-outerplanar, we might do worse on the planar graph as a whole because "greed" on one k-outerplanar hurts us on the next. Because of her buffer layers, Baker's k-outerplanars are independent and greed on one cannot hurt the next. (One may make similar remarks about the independent-set version of our versus Baker's algorithm.)

Our algorithm has the advantage over Baker's that it can often (although perhaps not always) totally avoid using colors beyond the 4th, since it, unlike Baker's method, has no need for "safety buffers" every kth layer.

It would be very interesting to see a proof or disproof of our conjecture. The version of this conjecture that states that, for some sufficiently large constant k, our Bakerian coloring algorithm will always find a 3-coloring of any 3-colorable planar graph, is definitely *false* due, essentially, to the NP-completeness construction [41]. However, if G is a maximal planar graph, i.e. fully-triangulated, then that version of the conjecture definitely becomes *true* by §4. The 4-color version of our conjecture is therefore more likely to be true if we demand that G be fully-triangulated.

10 Open problems

1. [chromatic # of fully-△-graph] For any fixed 2D surface topology S (orientable or not): Is there a polynomial-

[55] Actually, Baker had 8^ks in her bounds, because she used a sillier kind of dyanmic programming. The right procedure is this. Regard our graph as subsumed inside some fully triangulated k-outerplanar graph. Pick a central point (inside the innermost layer) and proceed "around" the graph "clockwise" with respect to p considering consecutive "radial" k-sets of vertices, according to a "topological scan." For each tabulate the 2^k possible vertex subsets and for each, whether it is allowed in an independent set and (if it is) what is the largest coardinlaity compatible independent set among the vertices scanned so far. When the scan goes "all the way around" 360° we demand compatibility with the original start-set, and this procedure has to be used for all 2^k possible start-sets.

[56] Actually, Baker did not discuss the question of coloring planar graphs, although she did discuss many other NP-complete problems. However, she could have, since her techniques can address coloring.

[57] By which we mean, color using the minimum possible number of occurrences of the colors > 4.

time algorithm to determine the chromatic number of a fully-triangulated graph embedded on S? We have seen the answer is "yes" if the chromatic number is either ≤ 3 or ≥ 5 [39][110], or if S is the sphere. But it might be hard to distinguish between the 4- and 5-chromatic graphs. Gimbel & Thomassen [44] showed that there for each S there is a polynomial-time algorithm to determine the chromatic number k of a triangle-free graph embedded on S, if $k \geq 4$; and there is a polynomial-time algorithm to determine the chromatic number of a girth≥ 6 graph embedded on S.

2. [4-colorability with bounded number of exceptions [61]] For any fixed 2D surface topology S (orientable or not): Is there a constant C_S such that removing $\leq C_S$ vertices from an S-embeddable graph, suffices to render it 4-colorable? We have seen this is true if "4" is replaced by "5." Removing $O(\sqrt{hV})$ vertices from a V-vertex graph is known to suffice on an h-handle body [63][31] to render the graph planar.

3. [chromatic number bounds for polyhedra] What about coloring the vertices (or faces) of *polyhedra*? See footnote 5. Thomassen ([10], [110] p.98) showed that every toral polyhedron with *convex* faces has 6-colorable vertices and 5-colorable faces (it is unknown if these statements are tight) although without the face-convexity restriction 7 colors can be required in both cases [28][102][40]. Polyhedra with genus-g surface and all faces convex can be face-colored using $o(g^{3/7})$ colors [44], which is smaller than Heawood's $g^{1/2}$ but still might be very weak. The true answer conceivably might be $O(\log g)$ or even $O(1)$.

4. [matchings and edge colorings] Tait's even-cycle partition result also may be used in the other direction. By taking every other edge in each even cycle we find a perfect matching. Thus this paper's linear-time 4-coloring algorithm for planar graphs immediately yields a linear-time algorithm to find a perfect matching in any 3-regular 2-connected planar graph. In fact, we actually produce *three* edge-disjoint matchings corresponding to Tait's three edge colors. Petersen's theorem of 1891 had shown that any 3-regular 2-edge-connected graph has a perfect matching, but until very recently the fastest known algorithm to find it was $O(V^{1.5})$ time. This is discussed in [13], who in 2000 found an entirely different linear-time matching algorithm in the planar case (but which only finds one, not three, matchings) and a $O(V \log^4 V)$-time algorithm for general 3-valent graphs. That algorithmic version of Petersen caused some excitement because it suggested there might be some unknown new algorithm for finding perfect matchings in V-vertex E-edge graphs that would be far superior to the Micali-Vazirani $O(E\sqrt{V})$-time algorithm. However, it seems impossible to generalize their algorithm even to, e.g, 5-regular 4-edge-connected graphs. The algorithm here, in contrast, depends on finding an even-cycle partition or an edge coloring. *Those* concepts perhaps *can* be generalized. The question then is: which r-regular graphs have even-cycle partitions or edge-colorings? Considering the immense effort that it took to prove the 4-color theorem, this question might be very difficult. It is known ("Vizing's theorem" [34]) that every graph with maximum valence ν can be edge-colored with either $\nu + 1$ or ν colors, the only difficulty is deciding which. Holyer [58] proved that this decision problem is NP-complete even for 3-regular graphs, and in the 3-regular case the even-cycle-partition problem is equivalent to edge 3-colorabilty and hence also is NP-complete, but as we've seen, both are linear time for *planar* 3-regular graphs. (Holyer's construction involves a large amount of nonplanarity.) So this approach may actually be a step backward. Schrivjer [98] showed that every *bipartite* regular graph has an edge-coloring that can be found in $O(E^2/V)$ time. Robertson, Seymour, and Thomas (unpublished) recently claim to have proven Tutte's conjecture that every 3-regular 2-connected graph without Petersen-graph minors, has an edge 3-coloring (and it may be found by a polynomial time algorithm).

5. [multidimensional version of 4-color conjecture] Consider a d-dimensional convex polyhedron with simplex $(d-1)$-faces.[58] Can each $(d-2)$-face be colored so that each $(d-1)$-simplex face has all its $(d-2)$-faces (of which there are d) different colors, and there are d colors in all in your palette?

When $d = 3$ the answer is "yes" – this is Tait's form of the 4-color theorem. When $d = 2$ this is the claim that we can 2-color the vertices of a polygon – which is true for even-gons and false for odd-gons. When $d = 4, 5, 6, \ldots$ these are new questions. I have a slight amount of evidence their answers always are "yes."

11 Acknowledgments

John Boyer's questions helped me to improve the exposition.

References

[1] A.V.Aho, J.Hopcroft, J.Ullman: The design and analysis of computer algorithms. Addison-Wesley, Reading, Massachusetts, 1974.

[2] Martin Aigner & Günter Ziegler: Proofs From the Book, 3rd ed. Springer 1998.

[3] R.E.L. Aldred, S. Bau, D.A. Holton, B.D. McKay: Nonhamiltonian 3-connected cubic planar graphs, SIAM J. Discrete Math. 13 (2000) 25-32.

[4] Frank Allaire: Another proof of the 4 color theorem, I, Proc 7th Manitoba Conf. Numerical Math'cs and Computing = Congressus Numerantium 20 (1977) 3-72.

[5] F.Allaire & E.R.Swart: A systematic approach to the determination of reducible configurations in the four-color conjecture, J.Combin.Theory 25,3 (Dec 1978) 339-362.

[6] K.Appel & W.Haken: Every planar map is four colorable, AMS (contemporary math. #98) 1989 (741-page book written with the collaboration of J. Koch). The solution of the four-color map problem, Scientific American 237,4 (1977) 108-121. The four-color proof suffices, Mathematical Intelligencer 8,1 (1986) 10-20, 58. Haken: An attempt to understand the four color problem, J.Graph Theory 1 (1977) 193-206.

[7] M.A.Armstrong: Basic Topology, Springer (UTM) 1983, 1997.

[8] David A. Bader & Sukanya Sreshta: A new parallel algorithm for planarity testing, TR 03-12, Electrical & Computer Engineering Department University of New Mexico, Albuquerque, NM 87131.

[9] Brenda S. Baker: Approximation algorithms for NP-complete problems on planar graphs, J. ACM 41, 1 (January 1994) 153-180.

[58] The prefix denotes dimensionality.

[10] D. Barnette: Coloring polyhedral manifolds, pp.192-195 in *Discrete geometry and convexity* (J. E. Goodman, E. Lutwak, J. Malkevitch, and R. Pollack, Eds.) New York Acad. Sci., 1985.

[11] Hannah Bast & Torben Hagerup: Fast Parallel Space Allocation, Estimation, and Integer Sorting, Info. & Computation 123,1 (1995) 72-110.

[12] F.R.Bernhart: A digest of the four color theorem, J.Graph Theory 1 (1977) 207-225.

[13] Therese C. Biedl, Prosenjit Bose, Erik D. Demaine, Anna Lubiw: Efficient Algorithms for Petersen's Matching Theorem, J. Algorithms 38 (2001) 110-134.

[14] George D. Birkhoff: The reducibility of maps, Amer. J. Math 35 (1913) 115-128.

[15] Jürgen Bokowski & Ulrich Brehm: A new polyhedron of genus 3 with 10 vertices, pp.105-116 in Intuitive geometry (Siofok, 1985), vol. 48 of Colloq. Math. Soc. János Bolyai, North-Holland, Amsterdam, 1987.

[16] J. Bokowski & U. Brehm: A polyhedron of genus 4 with minimal number of vertices and maximal symmetry, Geom. Dedicata 29,1 (1989) 53-64.

[17] J. Bosák: Hamiltonian lines in cubic graphs, Proc. International Seminar on Graph Theory & Applications (Rome 1966) 33-46. Gordon and Breach 1967.

[18] J.F. Boyar & H.J. Karloff: Coloring planar graphs in parallel, J. Algorithms 8 (1987) 470-479.

[19] John Boyer & Wendy Myrvold: Stop Minding Your P's and Q's: A Simplified O(n) Planar Embedding Algorithm ACM-SIAM Symposium on Discrete Algorithms (SODA) 10 (1999) 140-146. Journal version (30 pages) submitted to J. Graph Algorithms and Applications.

[20] Ulrich Brehm: Polyeder mit zehn Ecken von Geshlecht drei, Geometriae Dedicata 11 (1981) 119-124.

[21] U. Brehm: A maximally symmetric polyhedron of genus 3 with 10 vertices, Mathematika 34,2 (1987) 237-242.

[22] Gunnar Brinkmann, Brendan D. McKay, Ulrike von Nathusius: Backtrack search and look-ahead for the construction of planar cubic graphs with restricted face sizes, MATCH (Communications in mathematical and computer chemistry) 48 (2003) 163-177.

[23] J.L. Carter & M.N. Wegman: Universal Classes of Hash Functions, Journal of Computer & System Sciences 18 (1979) 143-154.

[24] Norishige Chiba & Takao Nishizeki: Arboricity and subgraph listing algorithms, SIAM J. Comput. 14,1 (1985) 210-223.

[25] N. Chiba, T. Nishizeki, S. Abe, T. Ozawa: A linear algorithm for embedding planar graphs using PQ-tree, J. Comput system Sci. 30 (1985) 54-76.

[26] N. Chiba and T. Nishizeki: The hamiltonian cycle problem is linear-time solvable for 4-connected planar graphs, J. Algorithms 10,2 (1989) 187-211.

[27] K.W.Chong & T.W.Lan: Finding connected components in O(log n loglog n) time on the EREW PRAM, J.Algorithms 18 (1995) 378-402.

[28] A. Császár: A polyhedron without diagonals, Acta Sci. Math. Szeged 13 (1949) 140-142. Gives a 7-vertex 14-triangular-face polyhedron, homeomorphic to a torus, whose edge-graph is K_7.

[29] M. Dietzfelbinger, A. Karlin, K. Mehlhorn, F. Meyer auf der Heide, H. Rohnert, R.E. Tarjan: Dynamic perfect hashing: upper and lower bounds, SIAM J. Computing 23,4 (1994) 738-761.

[30] G.A. Dirac: A property of 4-chromatic graphs and some results on critical graphs, J. London Math'l Soc. 27 (1952) 85-92.

[31] H.N.Djidjev & S.M.Venkatesan: Planarization of graphs embedded on surfaces, pp.62-72 in *graph theretical concepts in computer science* 21st Int'l workshop, Springer (LNCS #1017) 1995. Warning: the validity of this paper has recently been questioned.

[32] David Eppstein: Subgraph isomorphism in planar graphs and related problems, J. Graph Algorithms & Applications 3,3 (1999) 1-27.

[33] Jittat Fakcharoenphol & Satish Rao: Planar graphs, negative weight edges, shortest paths, and near linear time, Annual Symposium on Foundations of Computer Science FOCS 42 (2001) 232-242.

[34] S.Fiorini & R.J.Wilson: Edge colourings of graphs, Pitman, London 1977.

[35] S.Fisk & B.Mohar: Coloring graphs without short non-bounding cycles, J.Combin.Theory B 60,2 (1994) 268-271.

[36] Ph. Franklin: A six colour problem, J.Math.Phys. 13 (1934) 363-369.

[37] Greg N. Frederickson: On linear-time algorithms for 5-coloring planar graphs, Info. Proc. Lett. 19,5 (Nov. 1984) 219-224.

[38] M.L. Fredman, J. Komlos, E. Szemeredi: Storing a sparse table with O(1) worst case access time, J. ACM 31,3 (July 1984) 538-544.

[39] T.Gallai: Kritische graphen I, II, Publ. Math. Inst. Hungar. Acad. Sci. 8 (1963) 165-192, 373-395.

[40] Martin Gardner: "Mathematical Games" column in Scientific American 239,5 (Nov. 1978) 22-32. Describes Lajos Szilassi's (1977) polyhedron, which is a toroidal polyhedron with 7 faces, each a 6-gon tangent at an edge to each other face.

[41] M.Garey, D.S.Johnson, L.J.Stockmeyer: Some simplified NP-complete graph problems, Theoretical Computer Sci. 1,3 (1976) 237-267.

[42] Hillel Gazit: An optimal randomized parallel algorithm for finding connected components in a graph, SIAM J. Computing 20,6 (December 1991) 1046-1067.

[43] Ellen Gethner & William M. Springer II: How false is Kempe's proof of the four color theorem? Proc. 34th Southeastern International Conference on Combinatorics, Graph Theory and Computing, Congressus Numerantium 164 (2003) 159-175.

[44] J. Gimbel & C. Thomassen: Coloring graphs with fixed genus and girth, Trans. Amer. Math. Soc. 349,11 (1997) 4555-4564.

[45] S.J.Gismondi & E.R.Swart: A new type of 4-colour reducibility, Congressus Numerantium 81 (1991) 33-48.

[46] A. Goldberg, S. Plotkin and G. Shannon: Parallel symmetry-breaking in sparse graphs, Proc. ACM Symposium on Theory of Computing 19 (1987) 315-324.

[47] H. Grötzsch: Ein Dreifarbensatz für dreikreisfreie Netze auf der Kugel, Wissenschaftliche Zeitschrift der Martin-Luther-Universität Halle-Wittenberg, Mathematisch-Naturwissenschaftliche Reihe 8 (1958/1959) 109-120.

[48] Branko Grünbaum: Convex Polytopes, Second edition edited by Volker Kaibel, Victor Klee and Günter M. Ziegler, Springer (GTM #221) 2003.

[49] Branko Grünbaum: Grötzsch's theorem on 3-colorings, Michigan Math. J. 10 (1963) 303-310.

[50] L. J. Guibas & J. Stolfi: Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams, ACM Transactions on Graphics 4,2 (1985) 74-123.

[51] H.Hadwiger: Ungelöste Probleme, Elem. Math. 12 (1957) 61-62.

[52] T. Hagerup, M. Chrobak, K. Diks: Optimal parallel 5-coloring of planar graphs, SIAM J. Comput. 18,2 (1989) 288-300.

[53] Rudolf Halin: Über einen Satz von K. Wagner zum Vierfarbenproblem, Math. Annalen 153 (1964) 47-62.

[54] S.Halperin & U.Zwick: Optimal randomized EREW PRAM algorithms for finding spanning forests and for other basic graph connectivity problems, J. Algorithms 39 (2001) 1-46.

[55] Percy J. Heawood: Map-colour theorem, Quarterly J. Pure & Applied Math (Oxford) 24 (1890) 332-338.

[56] Percy J. Heawood: On the four-color map theorem, Quarterly J. Pure & Applied Math 29 (1898) 270-285.

[57] D.A. Holton & B.D. McKay: Cycles in 3-connected cubic planar graphs (II), Ars Combinatoria 21A (1986) 107-114.

[58] Ian Holyer: The NP-completeness of edge coloring, SIAM J. Computing 10,4 (1981) 718-720.

[59] J. Hopcroft & R.E. Tarjan: Dividing a graph into triconnected components, SIAM J. Computing 2,3 (1972) 135-158.

[60] J. Hopcroft & R. E. Tarjan: Efficient planarity testing, Journal of ACM 21 (1974) 449-568.

[61] Joan P. Hutchinson: A five color theorem for graphs on surfaces, Proc. Amer. Math. Soc. 90,3 (1984) 497-504.

[62] J.P.Hutchinson: Three-Coloring Graphs Embedded on Surfaces with All Faces Even-Sided, J.Combin.Theory B 65,1 (1995) 139-155.

[63] J. P. Hutchinson & G. L. Miller: Deleting vertices to make graphs of positive genus planar, Discrete Algorithms and Complexity theory, ed. Johnson et al., Academic Press (Boston 1987) 81-98.

[64] Joan Hutchinson & Stan Wagon: Kempe revisited, Amer. Math. Monthly 105,2 (Feb. 1998) 170-174.

[65] Bill Jackson & Xingxing Yu: Hamilton cycles in plane triangulations, J. Graph Theory 41,2 (2002) 138-150.

[66] M.Jungerman & G.Ringel: Minimal triangulations on orientable surfaces, Acta Math. 145, 1-2 (1980) 121-154.

[67] A.Kanevsky, R.Tamassia, G.Di Battista, J.Chen: On-line maintenance of the four-connected components of a graph (extended abstract), Symposium Foundations of Computer Science 32 (1991) 793-801.

[68] Louis H. Kauffman: Map coloring and the vector cross product, J. Combin. Theory B 48,2 (1990) 145-154.

[69] A.B.Kempe: How to color a map with 4 colors, Nature 21 (1880) 399-400.

[70] Samir Khuller: Coloring Algorithms for K_5-Minor Free Graphs, Info. Process. Lett. 34,4 (1990) 203-208.

[71] Samir Khuller: Extending Planar Graph Algorithms to K_{33}-free Graphs, Foundations of Software Technology and Theoretical Computer Science, FSTTCS 8 (1988) 67-79 (Pune, India, December 21-23, 1988). Springer (LNCS #338) 1988.

[72] D.G. Kirkpatrick: Optimal Search in Planar Subdivisions, SIAM J. Computing, 12,1 (1983) 28-35.

[73] Vladimir P. Korzhik: Another proof of the map color theorem for non-orientable surfaces, J.Combin.Theory B 86,2 (2002) 221-253.

[74] Łukasz Kowalik: Fast 3-coloring Triangle-Free Planar Graphs, European Sympos. Algorithms 12 (2004) 436-447 (Springer LNCS#3221).

[75] M.Krivelevich: An improved bound on the minimum number of edges in color-critical graphs, Electronic J. Combin. 5,1 (1998) paper R4 (4 pages).

[76] H.V.Kronk & A.T.White: A 4-color theorem for toroidal graphs, Proc. Amer. Math'l. Soc. 34 (1972) 83-86.

[77] Richard J. Lipton, Raymond E. Miller: A Batching Method for Coloring Planar Graphs, Info. Process. Lett. 7,4 (1978) 185-188.

[78] N.I. Martinov: 3-colorable planar graphs, Serdica 3 (1977) 11-16.

[79] Yuri Matiyasevich: Some probabilistic restatements of the Four Color Conjecture, Journal of Graph Theory 46,3 (2004) 167-179.

[80] D.W. Matula, Y. Shiloach, R.E. Tarjan: Analysis of two linear-Time Algorithms for Five-Coloring a Planar Graph, Congr. Numer. 33 (1981) 401-???; and STAN-CS-80-830 tech rept from Stanford CS dept.

[81] C. McMullen, C. Schulz, J. M. Wills: Polyhedral 2-manifolds in E^3 with unusually large genus, Israel J. Math. 46 (1983) 127-144.

[82] George J. Minty: A theorem on N-coloring a linear graph, Amer. Math'l. Monthly 69,7 (1962) 623-624.

[83] Maryam Mirzakhani: A small non-4-choosable planar graph, Bull. Inst. Combin. Applic. 17 (1996) 15-18.

[84] B.Mohar & P.D.Seymour: Coloring locally bipartite graphs on surfaces, J.Combin.Theory B 84,2 (2002) 301-310.

[85] C.A. Morgenstern & H.D. Shapiro: Heuristics for Rapidly 4-Coloring Large Planar Graphs, Algorithmica 6 (1991) 869-891.

[86] T. Nishizeki & N. Chiba: Planar Graphs: Theory and Algorithms, North-Holland Mathematics Studies 140, Elsevier 1988.

[87] Oystein Ore: The four color problem, Academic Press, New York, 1967.

[88] C.H. Papadimitriou & M. Yannakakis: The clique problem for planar graphs, Information Processing Letters 13 (1981) 131-133.

[89] V. Ramachandran & J.H. Reif: Planarity testing in parallel, J. Computer & System Sciences 49,3 (December 1994) 517-561.

[90] Gerhard Ringel: Färbungsprobleme auf Flächen und Graphen, VEB Deutsche Verlag Wissensch. Berlin 1959.

[91] G. Ringel: Map color theorem, Springer 1974 (GMW 209).

[92] G. Ringel & J.W.T. Youngs: Solution of Heawood map coloring problem, Proc. Nat'l Acad. Sci. USA 60,2 (1968) 438-445.

[93] Neil Robertson, Daniel Sanders, Paul Seymour, Robin Thomas: The Four-Colour Theorem, J. Combin. Theory B 70,1 (1997) 2-44. Electronic version at `http://www.math.gatech.edu/~thomas/FC/fourcolor.html`.

[94] Neil Robertson, Daniel Sanders, Paul Seymour, Robin Thomas: A new proof of the Four-Colour Theorem, Electronic Research Announcements of the AMS 2,1 (1996) 1-9

[95] Neil Robertson, Daniel Sanders, Paul Seymour, Robin Thomas: Efficiently four-coloring planar graphs, Proc. 28th ACM Symp. Theory of Computing (1996) 571-575.

[96] Neil Robertson, Paul Seymour, Robin Thomas: Hadwiger's conjecture for K_6-free graphs, Combinatorica 13,3 (1993) 279-361.

[97] Th.L. Saaty & P.C.Kainen: The four-color problem: assaults and conquest, Dover 1986 reprint of McGraw-Hill 1977 original. An earlier and more concise version of much of this material is Saaty: Amer. Math'l. Monthly 79 (1972) 2-43.

[98] Alexander Schrivjer: Bipartite-edge coloring in Δm time, SIAM J. Comput. 28,3 (1999) 841-846.

[99] Raimund Seidel & Micha Sharir: Top-Down Analysis of Path Compression, SIAM J. Computing 34,3 (2004) 515-525.

[100] L.Steen: Solution of the four color problem, Mathematics Magazine 49,4 (Sept 1976) 219-222.

[101] R. Steinberg: The state of the three color problem, pp.211-248 in Quo Vadis, Graph Theory?=Annals of Discrete Mathematics 55 (1993).

[102] Lajos Szilassi: Regular toroids, Structural topology 13 (1986) 69-80.

[103] R.E. Tarjan: Depth first search and linear graph algorithms, SIAM J. Comput. 1 (1972) 146-160.

[104] Robert E. Tarjan: Efficiency of a Good But Not Linear Set Union Algorithm, J.ACM 22,2 (1975) 215-225.

[105] Robin Thomas: An update on the four-color theorem, Notices AMS 45,7 (Aug 1998) 848-859.

[106] Carsten Thomassen: Five-coloring maps on surfaces, J. Combinatorial Theory B 59,1 (1993) 89-105.

[107] C. Thomassen: Grötzsch's 3-color theorem and its counterparts for the torus and projective plane, J. Combinatorial Theory B 62,2 (1994) 268-279.

[108] C. Thomassen: A short list color proof of Grötzsch's theorem, J. Combinatorial Theory B 88,1 (2003) 189-192.

[109] C.Thomassen: Every planar graph is 5-choosable, J. Combinatorial Theory B 62,1 (1994) 180-181.

[110] Carsten Thomassen: Color-Critical Graphs on a Fixed Surface, JCT B 70,1 (1997) 67-100.

[111] Carsten Thomassen: The chromatic number of a graph of girth 5 on a fixed surface, J. Combin. Theory B 87,1 (2003) 38-71.

[112] W.T.Tutte: On Hamiltonian circuits, J. London MAth'l Soc. 21 (1946) 98-101.

[113] K.Wagner: Über eine Eigenschaft der ebenen Komplexe, Mathematische Annalen 114 (1937) 570-590.

[114] P.Wernicke: Über den kartographischen Vierfarbensatz, Mathematischen Annalen 58 (1904) 413-426.

[115] H.Whitney: A numerical equivalent of the four-color map theorem, Monatsh. Math. und Physik 45 (1937) 207-213.

[116] M.H.Williams: A linear algorithm for colouring planar graphs with five colours, Computer J. 28 (1985) 78-81.

[117] Robin Wilson: Four colors suffice, Penguin 2002. (Popular exposition.)

[118] H.Peyton Young: A quick proof of Wagner's equivalence theorem, J. London Math'l Soc. 3 (1961) 661-664.

FIGURE CAPTION. Top left: K_7 embedded on a torus.
Top right: K_6 embedded on a Möbius strip.
Bottom left: The "3-shield" graph.
Bottom right: Hagerup et al [52]'s 5-coloring reduction for a 6-valent vertex w in a maximal planar graph. If all neighbors of w are nonadjacent, then three of them, such as 2,4,6, may be shrunk down to a single vertex after w's removal, without destroying planarity. If there is an adjacency we can assume wlog it is 1-3. If 1 is adjacent to both 3 and 5, then vertices 2,4,6 must be mutually nonadjacent and hence can be shrunk down. But if 1 is adjacent to 3 but not 5, then (1,5) and (2,4) would be shrinkable.