

A Computational Approach to the Four-Color Problem

By Hidehiko Yamabe and David Pope

Introduction. The problem of coloring a map on a sphere with four colors was shown by Heawood [3] in 1898 to be equivalent to the solution of a certain set of congruences, which we will call the *assignment problem* in this paper. The assignment problem is shown to be amenable to a simple computational searching routine, which can be easily coded for a binary automatic digital computer. A brief description of such a code for the UNIVAC Scientific 1103 computer is given, with two examples.

This work was completed just before Professor Yamabe's untimely death on November 20, 1960. It was hoped that the experimental results found by the computing machine might lead to an induction hypothesis, in particular, one involving the *complete set* of solutions to the assignment problem.

1. The Assignment Problem. By a "map" we will mean here a polygonal partition of the 2-sphere into a finite number of open simply-connected domains ("countries"). Each country will be bounded by a polygon with a finite number of edges and vertices. Two countries will be called *adjacent* if they have at least one edge in common. It is easily shown to be sufficient to look at the case when any two adjacent countries have only one edge in common. By a vertex we mean a point where at least three edges meet. By a coloring of the map we mean an assignment of one color to each country such that no two adjacent countries have the same color. The coloring problem is easily reduced to the case of a *normal* map, that is, where each vertex meets exactly three edges (Cf. [4], p. 298). In the rest of this paper we shall consider only normal maps.

We now define an *assignment* of the map as a function ϕ of the vertices of the map, giving each vertex v of the map the value $\phi(v) = +1$ or -1 . Then we define a *proper assignment* of the map as an assignment such that for each country, the sum of the values $\phi(v)$ taken over the vertices on the boundary of that country is congruent to zero modulo 3. We then have the following theorem, due to Heawood [3].

THEOREM 1. *A normal map can be colored with four colors if and only if there exists a proper assignment of the map.*

To derive the four-coloring of a map from a solution to the assignment problem, we first label the edges of the map a , b , or c such that at each vertex v three different letters meet, and the ordering of the edges defined by $a \rightarrow b \rightarrow c$ is a rotation in the positive direction around the vertex v if $\phi(v) = +1$, and a rotation in the negative direction if $\phi(v) = -1$. This labeling is consistent if and only if the assignment is a proper one.

Given such a labeling of the edges, the closed curves formed by the edges labeled alternately a and b separate the sphere into two disjoint (but not necessarily simply connected) regions, to which we may assign the numbers 0 and 1. The closed curves formed by the edges a and c also separate the sphere into two regions, to which we

Received July 20, 1960; revised December 27, 1960.

assign the numbers 0 and 2. The sum of these two numbers gives us the four-coloring of the map expressed as the "colors" 0, 1, 2, 3. (For a further discussion and proof, see [1] p. 226 and [2] p. 13.)

2. Description of the Code. The solutions to the assignment problem for the vertices of a given map are found by a searching routine, which was coded on the UNIVAC Scientific 1103 computer at the University of Minnesota Scientific Computation Laboratory. It is of interest to note that for this problem a binary computer is more efficient than a binary-coded decimal computer.

Each vertex of the map is represented in the computer by one of the 36 bits of an 1103 word; thus the present code is restricted to 36 or fewer vertices. Each country is then defined in one word, containing a binary "1" in each of the positions representing a vertex lying on its boundary, all the other binary digits being zero. The vertices are numbered from 1 to 36, and vertex j is represented by the j th binary digit from the left.

An assignment of +1 or -1 to each vertex of the map is also given in a single word, called Δ in the code, with +1 represented by a binary one, and -1 represented by a binary zero in the bit corresponding to the given vertex. Thus every possible assignment can be formed in Δ by successively adding one to the least significant digit, running Δ from zero to $2^{36} - 1$. (The complement form used for negative numbers is circumvented in this code.)

In practice, of course, this method of assignment generation would be highly inefficient. However, we may eliminate many assignments at once by a proper arrangement of the countries and vertices. To describe this arrangement, let $v(k)$ be the vertex with highest index lying on the boundary of country k . We then rearrange the ordering of the countries so that if $k_1 < k_2$, then $v(k_1) \leq v(k_2)$. We then check the given assignment, beginning with country 1. If the assignment does not satisfy the congruence for country 1, we may add $2^{36-v(1)}$ to Δ instead of 1, since the

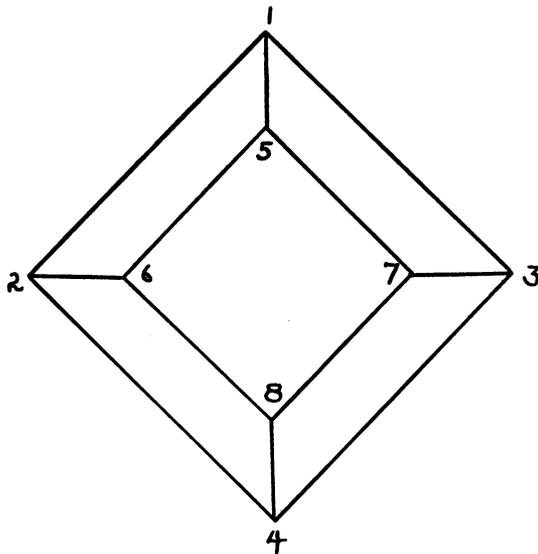


FIG. 1.—Map One

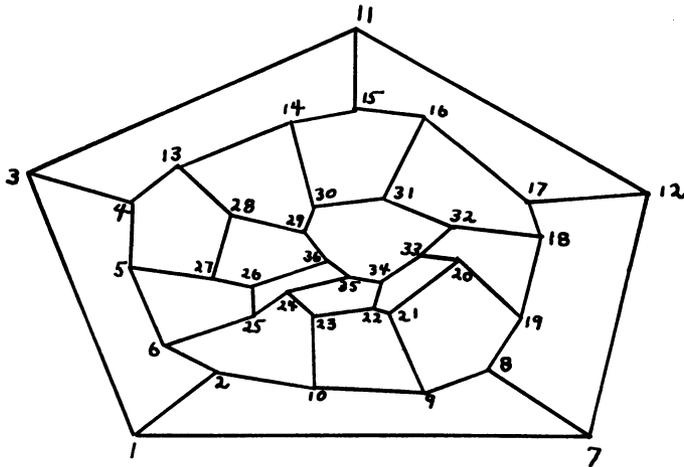


FIG. 2.—A Map With 20 Countries

bits $b_{36-v(1)} \cdots b_0$ of Δ do not affect this congruence. If this assignment does satisfy the congruence for country 1, we proceed to country 2, and so on. In this way many possibilities are eliminated at once for the countries with low index. The assignments satisfying the congruence for all countries are then printed out. Although the code will print out all the solutions to the assignment problem, clearly only half of these need to be computed as the other half are the reflection of these solutions obtained by interchanging + and -.

3. Examples. The simplest example computed was Map One, given in Figure 1. The total computing time for this, including output, was less than two seconds. The input data describing this map are given below, in octal.

MAP ONE INPUT

Address	Contents	Explanation
00201	74 00000 00000	Code words describing countries 1 to 6
00202	63 00000 00000	
00203	52 40000 00000	
00204	03 60000 00000	
00205	25 20000 00000	
00206	14 60000 00000	

The output of the code gives the eight solutions to the assignment problem, as follows:

MAP ONE OUTPUT

-	-	+	+	+	+	-	-
-	+	-	+	+	-	+	-
-	+	+	-	-	+	+	-
-	+	+	-	+	-	-	+
+	-	-	+	-	+	+	-
+	-	-	+	+	-	-	+
+	-	+	-	-	+	-	+
+	+	-	-	-	-	+	+

Note that the last four solutions are the images of the first four under the basic reflection, in addition to the other symmetries generated from the symmetries of the map.

The second example, given in Figure 2, has 20 countries and 36 vertices. The total number of solutions to the assignment problem is 146, or 73 if we discard the half given by the basic reflection. The first solution discovered was ---- ----
 -+-+ +-+ ---- ---- +-+ -+-+ +--- +--- +---
 ----, or 003 500 524 440 in the octal abbreviation. The total computation time for the 73 solutions was about 25 minutes.

University of Minnesota
 Minneapolis, Minnesota, and

University of California
 Davis, California

1. W. W. R. BALL, *Mathematical Recreations and Essays*, 11th Edition, London and New York, 1939.
2. H. S. M. COXETER, "Map coloring problems," *Scripta Math.*, v. 23, 1957, p. 11-25.
3. P. J. HEAWOOD, "On the four-colour map theorem," *Quart. J. Pure and Appl. Math.*, v. 29, 1898.
4. D. HILBERT & S. COHN-VOSSEN, *Anschauliche Geometrie*, Dover, New York, 1944.