The Complexity of Near-Optimal Graph Coloring

M. R. GAREY AND D. S. JOHNSON

Bell Laboratories, Murray Hill, New Jersey

ABSTRACT Graph coloring problems, in which one would like to color the vertices of a given graph with a small number of colors so that no two adjacent vertices receive the same color, arise in many applications, including various scheduling and partitioning problems In this paper the complexity and performance of algorithms which construct such colorings are investigated For a graph G, let $\chi(G)$ denote the minimum possible number of colors required to color G and, for any graph coloring algorithm A, let $A(G)$ denote the number of colors used by A when applied to G Since the graph coloring problem is known to be "NP-complete," it is considered unlikely that any efficient algorithm can guarantee $A(G) = \chi(G)$ for all input graphs In this paper it is proved that even coming close to $\chi(G)$ with a fast algorithm is hard Specifically, it is shown that if for some constant $r < 2$ and constant d there exists a polynomial-time algorithm A which guarantees $A(G) \leq r \cdot \chi(G) + d$, then there also exists a polynomial-time algorithm $\bar{A}$ which guarantees $\bar{A}(G) = \chi(G)$

KEY WORDS AND PHRASES computational complexity, analysis of algorithms, approximation algorithms, timetable problems, graph coloring, chromatic number, worst-case analysis, NP-complete

CR CATEGORIES: 5.25, 5 32

Graph coloring problems, of which the infamous "four-color problem" is a special case, have received considerable attention because of their conceptual simplicity and wide applicability. The basic problem is simply stated: Color the vertices of a given graph G, using as few colors as possible, so that no two adjacent vertices receive the same color. In view of the potential applications, it would be useful to have an efficient algorithm capable of coloring any graph G with this minimum number of colors (called the *chromatic number* of G and denoted by $\chi(G)$). Unfortunately no such efficient algorithm is currently known. In fact, no algorithm is known which requires an amount of time bounded by any polynomial in the number of vertices of the input graph. Moreover, recent work has shown that the graph coloring problem belongs to the class of "NP-complete" problems [1, 9], and therefore it seems unlikely that any such polynomial time-bounded algorithm exists.

One practical approach to such a computationally intractable problem is to relax the optimality constraint. Instead of requiring an optimal coloring, one might be perfectly willing to settle for a coloring which only uses "close" to the optimal number of colors. Any algorithm A which colors the vertices of any arbitrary graph G so that adjacent vertices of G receive different colors will be called a *graph coloring algorithm*, and we use $A(G)$ to denote the number of colors used by A when applied to G. Thus the revised approach may be stated as: Instead of searching for an efficient algorithm A which guarantees that $A(G) = \chi(G)$, try to devise an efficient algorithm A which merely guarantees that $A(G)/\chi(G)$ will be close to 1. Analogous fast "approximation algorithms" have been obtained for a number of other NP-complete problems [3, 5, 6, 8, 12].

In accord with this approach, quite a number of fast algorithms for near-optimal graph

Authors' address. Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974

coloring have been proposed, e.g. [11, 15, 16]. However in [7] it is shown that typical algorithms of this kind all have associated classes of graphs for which $A(G)/\chi(G)$ is unbounded. (In fact, the ratio can be made to grow essentially linearly with the number of vertices in G.) No proposed algorithm has yet proven better than this.

In this paper we shall prove a result which, while not showing the necessity of the arbitrarily bad behavior discussed above, does indicate that it is unlikely that any fast graph coloring algorithm can guarantee good approximations to the chromatic number. We prove that, unless there exists a polynomial time-bounded algorithm for determining the exact chromatic number of an arbitrary graph, no polynomial time-bounded graph coloring algorithm A can guarantee $A(G)/\chi(G) < r$, for any $r < 2$.

The foundation for our main result is provided by a theorem of Stockmeyer [2, 14], which we state as follows.

THEOREM 1 (Stockmeyer). *If there exists a polynomial time-bounded algorithm which, for any graph G, determines whether or not $\chi(G) \leq 3$, then there exists a polynomial time-bounded graph coloring algorithm A which guarantees that $A(G) = \chi(G)$.* [1]

We shall prove the following theorem.

THEOREM 2. *If, for some constant $r < 2$ and constant d, there exists a polynomial time-bounded graph coloring algorithm A which guarantees $A(G) \leq r \cdot \chi(G) + d$, then there exists a polynomial time-bounded algorithm for determining whether or not an arbitrary graph G satisfies $\chi(G) \leq 3$.*

Our main result is the following theorem, which is an immediate consequence of Theorems 1 and 2.

THEOREM 3 *If, for some constant $r < 2$ and constant d, there exists a polynomial time-bounded graph coloring algorithm A which guarantees $A(G) \leq r \cdot \chi(G) + d$, then there exists a polynomial time-bounded graph coloring algorithm $\bar{A}$ which guarantees $\bar{A}(G) = \chi(G)$.*

In order to prove Theorem 2 we require some additional graph theoretic machinery. A graph G will be denoted by an ordered pair (V, E) where V is the set of vertices and E the set of edges. A *multicoloring* [4] of $G = (V, E)$ is a function f defined on V whose values are sets satisfying $f(u) \cap f(v) = \varnothing$ whenever $\{u, v\} \in E$. For positive integers k, m, a *(k, m)-coloring* of $G = (V, E)$ is a multicoloring f of G such that $|f(v)| = m$ for each $v \in V$ and $|\bigcup_{v \in V} f(v)| = k$. The *$m$-chromatic number* $\chi_m(G)$ is the least integer k such that there exists a (k, m)-coloring of G. (This differs from the definition in [4].) Notice that for $m = 1$ these definitions correspond to the usual graph coloring notions.

Now we define a special class of highly structured graphs. For each positive integer $n \geq 3$, define the graph G_3^n to have vertex set

$$V = \{\{i, j, k\} : i, j, k \text{ distinct elements of } Z_n = \{1, 2, \cdots, n\}\},$$

with an edge joining two such vertices if and only if they are disjoint sets. Observe that G_3^{n-1} is a subgraph of G_3^n.

A useful fact about these special graphs is given in the following lemma.

LEMMA 1. *For $n \geq 7$, if I is an independent set[2] of vertices in G_3^n satisfying $\bigcap_{v \in I} v = \varnothing$, then $|I| \leq 3n - 8$.*

The proof of Lemma 1 is a tedious, though not complicated, case analysis and can be found in the Appendix. We shall use Lemma 1 to prove:

LEMMA 2. *For $n \geq 6$,*

(1) $\chi_3(G_3^n) = n$, *and*

(2) $\chi_4(G_3^n) = 2n - 4$.

[1] Actually [2, 14] only prove that the stated assumptions imply the existence of a polynomial time-bounded algorithm which, for any integer k and graph G, decides whether $\chi(G) \leq k$ However, it is an easy exercise to show how such an algorithm could be used to construct a polynomial time-bounded coloring algorithm A which guarantees $A(G) = \chi(G)$

[2] An independent set of vertices in G is a subset of vertices from G containing no two which are adjacent.

PROOF. We first prove (1) by induction. It is trivially true for $n = 6$. Assuming it is true for $n = N - 1$, we shall prove that it is true for $n = N$. First, recalling that each vertex of G_3^N is a set of three integers from Z_N, it is easy to see from the definition of G_3^N that one obtains an $(N, 3)$-coloring merely by assigning to each vertex v the set of colors $\{C_i : i \in v\}$. Thus $\chi_3(G_3^N) \leq N$. Now suppose there were an $(N - 1, 3)$-coloring of G_3^N. If some color appears on more than $3N - 8$ vertices (which must form an independent set I), then Lemma 1 implies that there exists some $i \in Z_n$ such that $i \in \bigcap_{v\in I}v$. But then this $(N - 1, 3)$-coloring of G_3^N contains an $(N - 2, 3)$-coloring of the copy of G_3^{N-1} formed by deleting all vertices of G_3^N which contain the element i, a contradiction to the inductive assumption. Therefore no color appears on more than $3N - 8$ vertices. Since there must be three colors on every vertex, we must have

$$(N - 1)(3N - 8) \geq 3\tbinom{N}{3}.$$

Rearranging, we obtain $(N - 1)(N - 4)^2 \leq 0$, which is false for all $N \geq 5$. This contradicts the assumed existence of an $(N - 1, 3)$-coloring and, by induction, completes the proof of (1).

We now prove (2), also by induction. It is trivially true for $n=6$. Assuming it is true for $n = N - 1$, we shall prove that it also holds for $n = N$. First, to obtain a $(2N - 4, 4)$-coloring of G_3^N, we extend the $(N, 3)$-coloring given previously by assigning one additional color from the set $\{C_1', C_2', \cdots, C_{N-4}'\}$ to each vertex. For $p \leq N - 5$, assign color C_p' to vertex v whenever $p = \min\{i \in v\}$. Assign color C_{N-4}' to each remaining vertex. Since each vertex v assigned color C_p', $p \leq N - 5$, contains the element p, no two such vertices can be adjacent in G_3^N. Furthermore, any vertex v assigned color C_{N-4}' must satisfy $v \subset \{N-4, N-3, N-2, N-1, N\}$, so that any two such vertices must contain a common element and cannot be adjacent in G_3^N. Thus, we have $\chi_4(G_3^N) \leq 2N - 4$.

To prove the other direction, suppose we had a $(2N - 5, 4)$-coloring of G_3^N. We first show that in this coloring there can be at most N colors which each appear on more than $3N - 8$ vertices. Since the set of vertices on which one particular color appears must form an independent set, Lemma 1 implies that if color C_l appears on independent set I_l with $|I_l| > 3N - 8$, then there exists an element $a_l \in Z_N$ such that $a_l \in \bigcap_{v\in I_l}v$. If there are $N + 1$ such colors, then there must be two of them, say C_k and C_l, such that $a_k = a_l$. But then the $(2N - 5, 4)$-coloring of G_3^N contains a $(2N - 7, 4)$-coloring of the copy of G_3^{N-1} formed by deleting all vertices which contain the element a_k, contradicting the inductive assumption.

Thus at least $N - 5$ colors each appear on no more than $3N - 8$ vertices. By Lemma 1, the vertices belonging to any larger independent set must all contain a common element. Thus the remaining colors can each appear on at most $\binom{N-1}{2}$ vertices. Since four colors must appear on each vertex of G_3^N, we must have

$$4\cdot\tbinom{N}{3} \leq (N - 5)(3N - 8) + N\cdot\tbinom{N-1}{2}.$$

Rearranging, we obtain

$$N(N - 10)(N - 11) + 30\ (N - 8) \leq 0,$$

which is false for all $N \geq 7$. This contradicts the assumed existence of a $(2N - 5, 4)$-coloring and, by induction, completes the proof of (2). □

Now we are prepared to prove Theorem 2.

PROOF OF THEOREM 2. Assume there exists a polynomial time-bounded graph coloring algorithm A which for a fixed $r < 2$ and constant d guarantees $A(G) \leq r\cdot\chi(G) + d$. We shall show how to use this algorithm to obtain a polynomial time-bounded algorithm for deciding whether or not an arbitrary graph G satisfies $\chi(G) \leq 3$.

Define $N = \min\{n \in Z : n \geq 6 \text{ and } (2n - 4) > r\cdot n + d\}$, which exists because $r < 2$. Let $G = (V, E)$ be any arbitrary graph. We construct a new graph G^* by replacing each *vertex* of G_3^N by a copy of the graph G and replacing each *edge* of G_3^N by a complete bipartite graph between the vertex sets of the two copies of G which replaced the endpoints of that edge. More specifically, for each vertex $x \in X$ of $G_3^N = (X, D)$, let $G(x) = (V(x), E(x))$ be a distinct copy of G. Then $G^* = (V^*, E^*)$ is defined by $V^* = \bigcup_{x \in X} V(x)$ and

$$E^* = (\textstyle\bigcup_{x \in X} E(x)) \cup (\bigcup_{\{x,y\} \in D} \{\{u, v\} : u \in V(x), v \in V(y)\}).$$

We make two claims concerning G^*:

CLAIM 1. *If* $\chi(G) \leq 3$, *then* $\chi(G^*) \leq N$.

CLAIM 2. *If* $\chi(G) > 3$, *then* $\chi(G^*) \geq 2N - 4$.

To prove Claim 1, assume $\chi(G) \leq 3$. To color G^*, fix an $(N, 3)$-coloring of G_3^N (which exists by Lemma 2). Since $\chi(G) \leq 3$, we can color the vertices in each subset $V(x)$ of V^* using the three colors assigned to x in the $(N, 3)$-coloring. Furthermore, since x, $y \in X$ receive disjoint sets of colors in the $(N, 3)$-coloring whenever $\{x, y\} \in D$, the construction of G^* guarantees that no two vertices $u \in V(x)$, $v \in V(y)$ for which $\{u, v\} \in E^*$ can receive the same color in this coloring of G^*. Thus we have colored G^* with N colors, which proves Claim 1.

To prove Claim 2, assume $\chi(G) > 3$. Then any coloring of G^* must use at least four distinct colors on the vertices $V(x)$ of each copy of G. Furthermore, any such coloring of G^* which uses k colors induces a $(k, 4)$-coloring of G_3^N, formed by assigning to each $x \in X$ any four of the colors which appear on the vertices in $V(x)$. By Lemma 2, however, this implies that k must be at least $2N - 4$, which proves Claim 2.

Now observe that, since N depends only on the fixed values of r and d, G^* can be constructed in time which is polynomial in the size of G. Thus, to decide in polynomial time whether or not $\chi(G) \leq 3$, we may first construct G^* and then apply algorithm A to G^*. By choice of N and Claims 1 and 2, algorithm A will use fewer than $2N - 4$ colors on G^* if and only if $\chi(G) \leq 3$. This completes the proof of Theorem 2. □

Having reached our stated goal, a few final comments are appropriate. Even if there is no efficient algorithm for determining the exact chromatic number of an arbitrary graph, our result leaves open the possibility that there may exist some constant $r \geq 2$ for which there is an efficient graph coloring algorithm A which guarantess $A(G)/\chi(G) < r$. We suspect that no such finite r exists, but our methods do not seem to generalize to values of r larger than 2. It might also be of interest to investigate the expected value of $A(G)/\chi(G)$ for fast graph coloring algorithms. However, at present few tools are available for accurate analysis of expected performance of algorithms, and in any case the applicability of such results in specific situations may be relatively weak.

Similar questions may also be asked for other NP-complete problems [13]. One such problem which is particularly interesting is that of finding the largest "clique" (complete subgraph) of a given graph. Denote the size of the largest clique in G by $C(G)$ and, for any clique finding algorithm A, let $A(G)$ be the size of the largest clique found by A when applied to G. Using a construction like that in the proof of Theorem 2, but embedding G into itself instead of G_3^N, it is not hard to prove the following result. If for some constant $r > 1$ there exists a polynomial time-bounded clique finding algorithm A which guarantees $A(G) \geq C(G)/r$, then for any $\epsilon > 0$ there exists a polynomial time-bounded clique finding algorithm A_ϵ which guarantees $A_\epsilon(G) \geq C(G)/(1 + \epsilon)$. Thus, if we can achieve *some* constant bound in polynomial time, then we can achieve *any* fixed constant bound within polynomial time. At present no polynomial time-bounded clique finding algorithm is known which guarantees such a bound, and perhaps none exists.

We should note that our lemmas are interesting graph theoretic results in themselves. In fact, Lemma 2 can be used to prove that $\chi(G_3^n) = n - 4$. This follows directly from the observation that $\chi_4(G_3^n) \leq \chi_3(G_3^n) + \chi(G_3^n)$, which can be demonstrated with a straightforward construction. The definition of G_3^n can be extended naturally to define graphs G_k^n which have vertices that are k-element subsets of Z_n and an edge joining any two which are disjoint. It has been conjectured [10] that $\chi(G_k^n) = n - 2k + 2$, which coincides with our result for $k = 3$. To our knowledge, this question remains open for $k > 3$.

Appendix

LEMMA 1. *For $n \geq 7$, if I is an independent set of vertices in G_3^n satisfying $\bigcap_{v \in I} v = \varnothing$, then $|I| \leq 3n - 8$.*

PROOF. Let I satisfy the hypothesis of the lemma. We may assume $I \neq \varnothing$ since the result is trivial otherwise. We begin by selecting three special elements of Z_n which will be used to partition I.

Choose $a \in Z_n$ such that $|\{v \in I : a \in v\}|$ is maximized. Then choose $b \in \{i : \exists j$ with $\{a, i, j\} \in I\}$ so that $|\{v \in I . a \notin v, b \in v\}|$ is as large as possible. Finally choose $c \in \{j : \{a, b, j\} \in I\}$ to maximize $|\{v \in I : a \notin v, b \notin v, c \in v\}|$. Note that, since I is nonempty, such a, b, and c exist.

We partition I into the following four sets:

$$I(ab) = \{v \in I : \{a, b\} \cap v = \{a, b\}\}, \qquad I(b) = \{v \in I : \{a, b\} \cap v = \{b\}\},$$

$$I(a) = \{v \in I . \{a, b\} \cap v = \{a\}\}, \qquad I(c) = \{v \in I : \{a, b, c\} \cap v = \{c\}\}.$$

The four sets are clearly disjoint. Furthermore, since $\{a, b, c\} \in I$ and I is an independent set for G_3^n, every member of I contains at least one element of $\{a, b, c\}$ and hence belongs to one of the four sets. Thus we have indeed partitioned I.

By choice of a, b, and c, we have $|I(a)| \geq |I(b)| \geq |I(c)|$. Therefore, since $\bigcap_{v \in I} v = \varnothing$, we must have $|I(b)| > 0$.

The lemma is an immediate consequence of the following two inequalities, which will be proved by case analysis.

CLAIM 1. $|I(a)| + |I(b)| \leq 2n - 6$.

CLAIM 2. $|I(ab)| + |I(c)| \leq n - 2$.

PROOF OF CLAIM 1.

Case 1.0. There exist distinct $i, j, k, l \in Z_n - \{a, b\}$ such that $\{\{b, i, j\}, \{b, k, l\}\} \subseteq I(b)$. In this case, each member of $I(a)$ must contain a and one element from each of $\{i, j\}$ and $\{k, l\}$. Thus

$$I(a) \subseteq \{\{a, i, k\}, \{a, i, l\}, \{a, j, k\}, \{a, j, l\}\} \quad \text{and}$$

$$|I(a)| + |I(b)| \leq 2 \cdot |I(a)| \leq 8 \leq 2n - 6 \quad \text{for} \quad n \leq 7.$$

Case 1.1. $|I(b)| = 1$. Let $\{b, i, j\}$ be the single element in $I(b)$. Then each member of $I(a)$ must contain either i or j. Thus

$$I(a) \subseteq \{\{a, i, l\} : l \in Z_n - \{a, b, i\}\} \cup \{\{a, j, l\} : l \in Z_n - \{a, b, i, j\}\} \quad \text{and}$$

$$|I(a)| + |I(b)| \leq (n - 3) + (n - 4) + 1 = 2n - 6.$$

Case 1.2. $|I(b)| = 2$ and case 1.0 does not hold. In this case, there exist distinct $i, j, k \in Z_n - \{a, b\}$ such that $I(b) = \{\{b, i, j\}, \{b, i, k\}\}$. This implies that

$$I(a) \subseteq \{\{a, i, l\} : l \in Z_n - \{a, b, i\}\} \cup \{\{a, j, k\}\},$$

so $|I(a)| + |I(b)| \leq (n - 3) + 1 + 2 = n < 2n - 6$ for $n \geq 7$.

Case 1.3(a). $|I(b)| \geq 3$ and $\bigcap_{v \in I(b)} v = \{b, i\}$ for some $i \neq b$. Since $|I(b)| \geq 3$, there must be distinct $j, k, l \in Z_n - \{a, b, i\}$ such that $\{b, i, j\}$, $\{b, i, k\}$, and $\{b, i, l\}$ all belong to $I(b)$. Any member of $I(a)$ which does not contain i would be disjoint from one of these three, contrary to I being an independent set for G_3^n. Thus

$$I(a) \subseteq \{\{a, i, h\} : h \in Z_n - \{a, b, i\}\} \quad \text{and}$$

$$|I(a)| + |I(b)| \leq 2 \cdot |I(a)| \leq 2(n - 3) = 2n - 6.$$

Case 1.3(b). $|I(b)| \geq 3$ and neither case 1.0 nor 1.3(a) holds. Let $\{b, i, j\} \in I(b)$. Since 1.3(a) does not hold, there must be some other member of $I(b)$ which does not contain i. Furthermore, since 1.0 does not hold, that member must contain j Thus there exists $k \in Z_n - \{a, b, i, j\}$ such that $\{b, j, k\} \in I(b)$. A similar argument shows that $I(b)$ must contain $\{b, i, k\}$. Since 1.0 does not hold, $I(b)$ contains only these three members. Because each member of $I(a)$ must intersect each member of $I(b)$ and $|I(a)| \geq |I(b)|$, we can conclude that $I(a) = \{\{a, i, j\}, \{a, i, k\}, \{a, j, k\}\}$, so that $|I(a)| + |I(b)| = 6 < 2n - 6$ for $n \geq 7$.

This completes the proof of Claim 1.

PROOF OF CLAIM 2.

Case 2.0. $|I(c)| = 0$. Since $I(ab) \subseteq \{\{a, b, i\} : i \in Z_n - \{a, b\}\}$, we have $|I(ab)| + |I(c)| \leq (n - 2) + 0 = n - 2$.

Case 2.1. $|I(c)| = 1$. Let $\{c, i, j\}$ be the single member. Then we must have

$$I(ab) \subseteq \{\{a, b, c\}, \{a, b, i\}, \{a, b, j\}\} \quad \text{and}$$

$$|I(ab)| + |I(c)| \leq 4 < n - 2 \quad \text{for} \quad n \geq 7.$$

Case 2.2(a). $|I(c)| \geq 2$ and $\bigcap_{v \in I(c)} v = \{c, i\}$ for some $i \neq c$. In this case we have $I(c) \subseteq \{\{c, i, j\} : j \in Z_n - \{a, b, c, i\}\}$ and, since $|I(c)| \geq 2$, $I(ab) \subseteq \{\{a, b, c\}, \{a, b, i\}\}$. Thus $|I(ab)| + |I(c)| \leq (n - 4) + 2 = n - 2$.

Case 2.2(b). $|I(c)| \geq 2$ and $\bigcap_{v \in I(c)} v = \{c\}$. Since $\{a, b, i\} \in I(ab)$ implies that every member of $I(c)$ must contain i, we have that $I(ab)$ can contain only one member, $\{a, b, c\}$. Moreover, Claim 1 and $|I(a)| \geq |I(b)| \geq |I(c)|$ imply that $|I(c)| \leq n-3$. Thus $|I(ab)| + |I(c)| \leq 1 + (n - 3) = n - 2$.

This completes the proof of Claim 2 and the proof of Lemma 1. □

REFERENCES

1. AHO, A. V, HOPCROFT, J. E, AND ULLMAN, J. D *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, Mass, 1974, Chap 10
2. GAREY, M R, JOHNSON, D. S, AND STOCKMEYER, L. Some simplified NP-complete graph problems *Theor Comput Sci* (to appear).
3. GRAHAM, R. L Bounds on multiprocessing timing anomalies *SIAM J on Appl. Math. 17* (1969), 416–429.
4. HILTON, A J. W., RADO, R., AND SCOTT, S. H. A (<5)-colour theorem for planar graphs. *Bull. London Math. Soc. 5* (1973), 302–306.
5. IBARRA, O H, AND KIM, C. E. Fast approximation algorithms for the knapsack and sum of subset problems Tech. Rep 74-13, Dep of Computer, Information, and Control Sciences, U. of Minnesota, Minneapolis, Minn., 1974.
6. JOHNSON, D. S. Approximation algorithms for combinatorial problems *J. Comput. and Syst. Scis 9* (1974), 256–278
7. JOHNSON, D. S Worst case behavior of graph coloring algorithms Proc of the Fifth Southeastern Conf on Combinatorics, Graph Theory, and Computing, Utilitas Mathematica Publishing, Winnipeg, Canada, 1974, pp 513–528.
8. JOHNSON, D S, DEMERS, A, ULLMAN, J D, GAREY, M. R, AND GRAHAM, R. L Worst-case performance bounds for simple one-dimensional packing algorithms *SIAM J Comput. 3* (1974), 299–326
9. KARP, R M Reducibility among combinatorial problems. In *Complexity of Computer Computations*, R E. Miller and J W. Thatcher, Eds, Plenum Press, New York, 1972, pp 85–104.
10. LOVÁSZ, L. (Personal communication.)
11. MATULA, D. M, MARBLE, G, AND ISAACSON, J D Graph coloring algorithms In *Graph Theory and Computing*, R. C Read, Ed., Academic Press, New York, 1972, pp 109–122.

12 ROSENKRANTZ, D J., STEARNS, R. E , AND LEWIS, P M Approximate algorithms for the traveling salesperson problem Proc 15th Annual Symp. on Switching and Automata Theory, IEEE, 1974, pp 33–42
13. SAHNI, S , AND GONZALES, T. *P*-Complete problems and approximate solutions Proc. 15th Annual Symp on Switching and Automata Theory, IEEE, 1974, pp. 28–32
14. STOCKMEYER, L. Planar 3-colorability is polynomial complete. *ACM SIGACT News 5*, 3 (1973), 19–25
15 WELSH, D J A., AND POWELL, M. B An upper bound to the chromatic number of a graph and its application to time tabling problems *Comput J 10* (1967), 85–86.
16 WOOD, D. C. A technique for coloring a graph applicable to large scale time-tabling problems. *Comput. J 12* (1969), 317–319

RECEIVED FEBRUARY 1975, REVISED MAY 1975