The Mapmaker's Dilemma

by

Richard Beigel*
Department of Computer Science
Yale University
P.O. Box 2158, Yale Station
New Haven, CT 06520-2158.

William I. Gasarch**
Department of Computer Science
Institute for Advanced Study
University of Maryland
College Park, MD 20742

1) Introduction

We examine the problem of coloring part of a k-colorable graph, while not knowing the rest of it. To illustrate some of our concepts, we describe a somewhat whimsical scenario, which we call the Mapmaker's Dilemma.

Picture a 12th-century mapmaker who is given a map of Europe and the countries adjacent to Europe, and is told to 4-color the European countries in a manner that is consistent with some coloring of the entire world. Unfortunately, Asia has not yet been explored. He cannot expect to find a consistent coloring given incomplete information, but the people must have their maps colored, so he colors Europe based on the information at hand. The world is small, and he does not have anything else to do, so this takes negligible time. From time to time, he receive reports from explorers of newly discovered countries and their neighbors. (This is a relatively peaceful time, so countries and borders do not disappear from the map.) This new information may invalidate his 4-coloring of Europe, so that he has to recolor it, at great cost to his publisher. Four-color photocopying is a

* Supported in part by the National Science Foundation under grants CCR-8808949 and CCR-8958528. Research performed while this author was at the Johns Hopkins University.

** Supported in part by the National Science Foundation under grants CCR-8803641 and DCR-8405079.

novelty that few can afford; therefore he hopes, through some cleverness, to minimize the number of times he has to recolor the map of Europe.

We study a problem similar to that of the mapmaker. In particular, we study the problem of coloring a subgraph H of a k-colorable graph G, where G is given to us only a little at a time.

In Sections 2 and 3 we formalize this problem in terms of a game with parameters k and n. The players in the game are named "The Mapmaker" and "The Explorer." Intuitively, the mapmaker tries to color n nodes of a k-colorable graph G, while the explorer reveals more and more of the graph; the explorer wins if he can make the mapmaker change his mind many times. In Section 4 we prove our main theorem: for $3 \leq k \leq n$ the Explorer has a winning strategy. In Section 5 we show that if $k = 2$ then the Mapmaker has a winning strategy.

Informally, our main theorem says that coloring part of a graph in an extendible way is hard in that it may require looking at all possible colorings. In Section 6 we formalize and prove this statement in the context of recursive graph theory. For the related problem of graph k-colorability it is a major open question whether such a brute force algorithm is required, namely the P=?NP question. Additional discussion of connections between recursive graph theory and complexity theory may be found in [6]. Other connections between the work here and complexity theory are in [9]. Lakshmipathy and Winklmann [9] have previously proved the $k = 3$ case of Lemma 4, and applied it to communication complexity theory in an interesting way.

2. Conventions, Definitions, and Notation

Definition: Let $G = (V, E)$ be a graph and let $k \in N$. A *k-coloring of G* is a function $c : V \to \{1, \ldots, k\}$ such that if $\{v_1, v_2\} \in E$ then $c(v_1) \neq c(v_2)$. A graph G is *k-colorable* if such a c exists.

Definition: Let G be a graph, let H be a subgraph of G, and let c be a coloring of H. A coloring c' of G is an *extension* of c if for all v in H, $c'(v) = c(v)$. A coloring c of H is *(G, k)-extendible* if there exists a k-coloring c' of G that is an extension of c. If the value of k is understood then we say that c is G-extendible. If the graph G is also understood then we say that c is a *local coloring of H*.

Two colorings c_1 and c_2 are *isomorphic* (denoted $c_1 \simeq c_2$) if there exists a permutation σ of $\{1, 2, \ldots, k\}$ such that, for all v in H, $c_1(v) = \sigma(c_2(v))$. By convention we regard isomorphic colorings as being the same. A statement such as "G has two k-colorings" means "G has two nonisomorphic k-colorings." Formally the term "coloring" should be replaced by "isomorphism class of colorings" throughout.

Notation: If A is a set and $k \in N$ then $[A]^k$ denotes the set of all k-element subsets of A, and $[A]^{<\omega}$ denotes the set of all finite subsets of A.

Let G be a k-colorable graph and let H be a subgraph of G. $NI(G, H, k)$ denotes the number of nonisomorphic (G, k)-extendible colorings of H. $NI(k, n)$ denotes $NI(H, H, k)$. where H is the graph consisting of n isolated nodes. It is easy to show that $NI(k, n)$ is exponential in n. We include a proof for completeness. The quantity $NI(k, n)$ is clearly equal to $\sum_{s=1}^{k} S(n, s)$ where $S(n, s)$ is the number of ways of partitioning n identical objects into s classes (also called a Stirling number of the second kind). We take $S(n, 0) = 0$. By inclusion-exclusion,

$$S(n, s)s! = \sum_{r=0}^{s} (-1)^r \binom{s}{r} (s - r)^n.$$

Recall that

$$NI(k, n) = \sum_{s=0}^{k} S(n, s) = \sum_{s=0}^{k} \sum_{r=0}^{s} \frac{1}{s!} (-1)^r \binom{s}{r} (s - r)^n = \sum_{s=0}^{k} \sum_{r=0}^{s} \frac{(-1)^r (s - r)^n}{r!(s - r)!}.$$

Replacing $s - r$ by t and summing in a different order we obtain

$$NI(k, n) = \sum_{t=0}^{k} \sum_{r=0}^{k-t} \frac{t^n}{t!} \frac{(-1)^r}{r!} = \sum_{t=0}^{k} \frac{t^n}{t!} \sum_{r=0}^{k-t} \frac{(-1)^r}{r!}.$$

For large n and fixed k this is approximately $k^n / k!$. $\boxtimes$

3. The Local Coloring Game

We describe the *local coloring game* with parameters k and n, $k \leq n$ with two players named "The Mapmaker" and "The Explorer." Formally we define the game as follows. In the first round of the game the Explorer presents $G_1 = H$, and the Mapmaker responds with a k-coloring c_1 of H. In the s^{th} round the Explorer presents a k-colorable graph

$G_s \supseteq G_{s-1}$, and the Mapmaker responds with a k-coloring c_s of H that is G_s-extendible. The game goes on indefinitely. If in the course of the game the Mapmaker presents $NI(k,n)$ different colorings then the Explorer wins; otherwise the Mapmaker wins.

The main theorem of this paper is that if $k \geq 3$ then the Explorer has a winning strategy. The idea is that whatever G_{s-1}-extendible coloring c_{s-1} of H that the Mapmaker tries during round $s-1$, there exists a k-colorable graph $G_s \supseteq G_{s-1}$ such that c_{s-1} is not G_s-extendible; but all other G_{s-1}-extendible colorings c, $c \not\approx c_{s-1}$, are G_s-extendible. Thus the Explorer can invalidate exactly one k-coloring of H per stage. The Mapmaker is then forced to try all k-colorings of H.

4. A Winning Strategy for the Explorer

In the next four lemmas we exhibit graphs that the Explorer will use in his winning strategy. We only supply the descriptions (and pictures) of the graphs; the proofs that they work are easy exercises left to the reader.

Lemma 1: Let $k, m \in N$ be such that $m \geq 1$ and $k \geq 3$, let $A = \{v_1, \ldots, v_m\}$ be a set of vertices, and let v be a vertex that is not in A. There exists a k-colorable graph $SAME(A, v)$ such that

1) there exists a k-coloring c of $SAME(A, v)$ where all the vertices in A are colored the same,
2) if c is a k-coloring of $SAME(A, v)$, where all the vertices in A are colored the same, then v also has that color,
3) if c is a k-coloring of A, where not all the vertices are colored the same, then for any color a, there exists a k-coloring c' of $SAME(A, v)$ that extends c such that $c'(v) = a$.

Proof:

If $m = 1$ then let $SAME(A, v)$ be the graph consisting of v and v_1 connected to every vertex of a $(k-1)$-clique (see Figure 1).

We consider the $m = 2$ case. Let W be a clique on $k-1$ new vertices $\{w_1, \ldots, w_{k-1}\}$. To form $SAME(A, v)$ connect v to every vertex in W, connect v_1 to every vertex in W except w_{k-1}, and connect v_2 to w_{k-1} (see Figure 2).

We now consider the case for general m. Let $A = \{v_1, \ldots, v_m\}$. Let $w_1, \ldots, w_{m-2}$ be new vertices, and let w_{m-1} be v. Let

$$SAME(A, v) = SAME(\{v_1, v_2\}, w_1) \cup \bigcup_{i=1}^{m-2} SAME(\{w_i, v_{i+2}\}, w_{i+1})$$

(see Figure 3). ⊠

Lemma 2: Let $j, k \in N$, $2 \leq j \leq k$, and let $A = \{v_1, \ldots, v_j\}$ be a set of vertices. There exists a k-colorable graph $LESS(A)$ such that

1) if c is a k-coloring of $LESS(A)$ then $v_1, v_2, \ldots, v_j$ cannot all be assigned different colors, i.e. $|\{c(v) : v \in A\}| < j$, and
2) any $(j-1)$-coloring of A can be extended to a k-coloring of $LESS(A)$.

Proof:

Let W be the complete graph on $k - j + 1$ vertices. $LESS(A)$ is formed by connecting every vertex in A to every vertex in W (see Figure 4). ⊠

Lemma 3: Let $k \in N$, $k \geq 3$, and let $A = \{v_1, \ldots, v_m\}$ be a set of vertices. There exists a k-colorable graph $NEQ(A)$ such that

1) if c is a k-coloring of $NEQ(A)$ then the vertices in A must use more than one color, i.e. $|\{c(v) : v \in A\}| > 1$, and
2) if d is a k-coloring of A such that $|\{d(v) : v \in A\}| > 1$ then d can be extended to a k-coloring of $NEQ(A)$.

Proof:

Let I be the graph consisting of an edge connecting two new vertices x_1 and x_2. Let $NEQ(A) = SAME(A, x_1) \cup SAME(A, x_2) \cup I$ (see Figure 5). ⊠

The next lemma is the key to the proof of our main theorem. The lemma roughly states that if $H \subseteq G$ and c is an extendible coloring of H, then G can be extended to G' in such a way to make c not extendible to a coloring of G', but not to exclude any other coloring.

Lemma 4: Let

a) $k \in N$ be such that $3 \leq k$,

b) G be a k-colorable graph,

c) H be a subgraph of G,

d) c be a G-extendible coloring of H.

There exists a graph $G' = SPOIL(k, G, H, c)$ such that $G' \supseteq G$ and

1) c is not G'-extendible, and

2) if d is any G-extendible coloring of H, $d \not\approx c$, then d is G'-extendible.

Proof:

Assume c uses $j \leq k$ colors. We consider the case $j \geq 2$ first.

For $1 \leq i \leq j$ let $A_i \subseteq H$ be the set of vertices v such that $c(v) = i$. Let $v_1, \ldots, v_j$ be new vertices. Let $G' = G \cup (\bigcup_{i=1}^{j} SAME(A_i, v_i)) \cup LESS(\{v_1, \ldots, v_j\})$ (see figure 6.)

We now consider the $j = 1$ case. In this case c assigns all the vertices of H the same color. Let $G' = G \cup NEQ(H)$. ☒

Theorem 5: Let $k, n \in N$, $3 \leq k \leq n$. The Explorer has a winning strategy for the local coloring game with parameters k and n.

Proof:

During stage $s \geq 1$ the Mapmaker will present a k-coloring c_s of H that is G_{s-1}-extendible. The Explorer's winning strategy is to play

$$G_s = \begin{cases} SPOIL(k, G_{s-1}, H, c_s) & \text{if } SPOIL(k, G_{s-1}, H, c_s) \text{ is } k\text{-colorable,} \\ G_{s-1} & \text{otherwise.} \end{cases}$$

At every stage $s < NI(k, n)$ the Explorer eliminates exactly one k-coloring from being a G_s-extendible coloring of H. Thus the Mapmaker is forced to present $NI(k, n)$ different k-colorings of H. ☒

5. The $k = 2$ Case

Throughout this paper we have been assuming that $k \geq 3$. We show that this is necessary, i.e. if $k = 2$ then the Mapmaker has a winning strategy.

Theorem 6: Let $n \in N$, $k = 2$. The Mapmaker has a winning strategy for the local coloring game. In fact, if $k = 2$ then the Mapmaker has a strategy such that he presents at most n different colorings.

Proof:

In round 1 the Mapmaker colors H arbitrarily. In each subsequent round, if the Mapmaker can use the same coloring he used in the proceeding round he does so, else he uses some arbitrary coloring that has not been ruled out.

The only rounds where the Mapmaker must present a new coloring are those where the Explorer connects two components of H (the path may use vertices that are not in H). Hence the Mapmaker presents at most n different colorings. $\boxtimes$

In the $k = 2$ case, the Explorer can easily force the Mapmaker to present exactly n colorings. Hence the result above is tight.

6. An Analog to P=?NP in Recursive Graph Theory

We use the techniques of Section 3 to solve a problem in recursive graph theory that resembles P=?NP. The problem asks (informally) if a particular problem that can be solved by a naive exponential brute force algorithm can be solved more efficiently. We show that the brute force algorithm is optimal. The problem has its roots in [5,7].

Definition: A (possibly infinite) graph $G = (V, E)$ is *recursive* if every vertex has finite degree and both $V \subseteq N$ and $E \subseteq [N]^2$ are recursive (i.e., decidable by a Turing machine [10,11])

Many references to articles on recursive graph theory can be found in [3].

Definition: A recursive graph $G = (V, E)$ is *recursively k-colorable* if there exists a recursive function $f : V \to \{1, 2, \ldots, k\}$ that is a coloring of G (i.e. there is a Turing machine that computes a k-coloring of G).

Bean [1] proved that there exists a 3-colorable recursive graph that is not recursively k-colorable for any k. Carstens and Pappinghaus [5] considered coloring algorithms that are allowed to "change their mind" b times where b is a fixed constant. They showed that for all $k \geq 3$ there exists a k-colorable recursive graph that cannot be k-colored by such an algorithm. These results have been extended in ([7] chapter 5.2).

We consider a similar type of coloring problem and improve on the results in [5,7].

Definition: Let $G = (V, E)$ be a k-colorable recursive graph. A *local k-coloring of G* is a function that takes a finite set $H \subseteq V$ and outputs a G-extendible k-coloring of H.

We examine the complexity of local k-colorings. Our measure of complexity is "mind-changes." In particular we study algorithms for local k-colorings that are allowed to change their mind $g(n)$ times on inputs consisting of n vertices. The function g is the complexity of the algorithm.

In what follows we will interpret the input to a Turing machine as an ordered pair (H, s) where H is a finite set of vertices and s is a parameter; and the output as a coloring of those vertices.

Definition: Let f be a function from $[N]^{<\omega}$ to N, and let g be a function from N to N. The function f is *computable by a g-mind-change algorithm* if there exist a Turing machine M, that halts on every input, such that for all $H \in [N]^{<\omega}$

a) $\lim_{s\to\infty} M(H,s) = f(H)$ (i.e., $(\exists s_0)(\forall s \geq s_0)M(H,s) = f(H)$).

b) $|\{s : M(H,s) \neq M(H,s+1)\}| \leq g(|H|)$,

Algorithms like M in the above definition are called *mind-change algorithms.* If $|\{s : M(H,s) \neq M(H,s+1)\}| \geq b$ then we say that "M has changed its mind at least b times on H." Many references to mind-change algorithms can be found in [4].

Carstens and Pappinghaus [5] showed that one can color a recursive graphs with a mind-change algorithms that changes its mind an exponential number of times. We sharpen their result and put it in our terminology.

Theorem 7: Let $G = (V, E)$ be a k-colorable recursive graph. There exists a local k-coloring of G that is computable by a g-mind-change algorithm where $g(n) = NI(n,k) - 1$.

Proof:

The following mind-change algorithm M changes its mind only when the k-coloring it thought was correct is shown not to be G-extendible.

ALGORITHM for M

1) Input(H, s).
2) If $s = 0$ then output the coloring that maps all elements of H to 1.
3) Compute $c = M(x, s-1)$ and $G_s = (V_s, E_s)$ where

$$V_s = (V \cap \{1, 2, \ldots, s\}) \cup H$$
$$E_s = E \cap \{(x, y) : x, y \in V_s\}$$

4) If c is G_s-extendible then output c, else output a k-coloring of H that is G_s extendible (for definiteness take the least coloring relative to a fixed ordering).

END OF ALGORITHM

Let $H \subseteq V$, $|H| = n$. If a k-coloring c of H is not G_s-extendible then, for all $t \geq s$, c is not G_t-extendible. Therefore each k-coloring of H is tried at most once. Since there are $NI(n,k)$ different k-colorings of H

$$|\{s : M(H,s) \neq M(H,s+1)\}| \leq NI(k,n) - 1.$$

Since the number of mind-changes is bounded, $\lim_{s\to\infty} M(H,s)$ exists. We denote this k-coloring by c. We show that c is G-extendible. Assume, by way of contradiction, that c is not G-extendible. By a compactness argument (similar to those in [8]) there exists $t \in N$ such that c is not G_t-extendible. Hence $\lim_{s\to\infty} M(H,s) \neq c$, a contradiction. $\boxtimes$

We now show that the brute-force algorithm in Theorem 7 is optimal.

Let $M_1, M_2, M_3, \ldots$ be an acceptable numbering [10,11] of all Turing machines (i.e., from e the code for M_e can be recovered and M_e can be run on an input).

Definition: A (infinite) recursive partition of N is a partition $N = \bigcup_{i=1}^{\infty} Z_i$ where each Z_i is a (infinite) set, and the function that maps x to the i such that $x \in Z_i$ is recursive.

Definition: Let $\langle -,- \rangle$ be a recursive bijection from $N \times N$ to N (e.g. $\langle 4,17 \rangle$ is the number that the ordered pair $(4,17)$ gets mapped to).

Theorem 8: Let $k \geq 3$. There exists a k-colorable recursive graph G such that *every* mind-change algorithm that computes a local k-coloring of G requires $NI(k,n)-1$ mind-changes on an infinite number of inputs H of arbitrarily large cardinality.

Proof:

We construct G to satisfy the following requirements, indexed by $\langle e,n \rangle \in N$:

$R_{\langle e,n\rangle}$: *If M_e computes a local k-coloring of G in the limit, then $(\exists H \subseteq V)$, $|H| = n$, such that M_e changes its mind at least $NI(k,n) - 1$ times on H.*

Let $\{Z_{\langle e,n\rangle}\}_{\langle e,n\rangle=1}^{\infty}$ be an infinite recursive partition of N. For every $\langle e,n\rangle \in N$ we construct a k-colorable recursive graph $G(e,n)$ whose vertex set is a subset of $Z_{\langle e,n\rangle}$, to satisfy requirement $R_{\langle e,n\rangle}$. To construct $G(e,n)$ we essentially play the local coloring game with parameters k and n, where the algorithm M_e plays the role of the Mapmaker, and we play the role of the Explorer. Since each $G(e,n)$ is recursive, and $\{Z_{\langle e,n\rangle}\}_{\langle e,n\rangle=1}^{\infty}$ is a recursive partition, the graph $G = \bigcup_{\langle e,n\rangle=1}^{\infty} G(e,n)$ is recursive.

Fix $\langle e,n\rangle \in N$. Let Z be $Z_{\langle e,n\rangle}$. Let M be M_e. We construct $G = G(e,n)$ in stages. G_s denotes G at the end of stage s. The final graph G will be $\bigcup_{s=1}^{\infty} G_s$.

CONSTRUCTION

Stage 0: Let H be the set consisting of the least n numbers in Z. Let G_0 have vertex set H and no edges.

Stage s+1: Run $M(H,s)$. If the computation halts then let t be the number of steps it took. If $M(H,s)$ is a G_s-extendible k-coloring of H, and $SPOIL(k, G_s, H, M(H,s))$ is k-colorable, then set

$$G_{s+1} = SPOIL(k, G_s, H, M(H,s)),$$

making sure that the new vertices added are the least numbers in Z that are greater than both t and the total number of steps the construction has taken before this stage; otherwise set G_{s+1} to G_s.

END OF CONSTRUCTION

If there is an s such that $M(H,s)$ does not converge then the construction never goes past stage s. Even though we do not know if this happens or not, the graph G is recursive. A number t is a vertex iff t enters the graph during the first t steps (not stages) of the construction. A pair (t_1,t_2) is an edge iff (t_1,t_2) becomes an edge of the graph in the first $\max(t_1,t_2)$ steps. Both of these conditions can be tested recursively by running the construction for a finite number of steps.

We show that requirement $R_{\langle e,n\rangle}$ is satisfied. Assume $M = M_e$ computes a local k-coloring of G in the limit. Let H be as in the construction of $G(e,i)$. Assume, by way of contradiction, that

$$|\{s : M(H,s) \neq M(H,s+1)\}| < NI(k,n) - 1.$$

Let s be the minimal stage such that $M(H, s)$ has reached its limit. At stage s of the construction $M(H, s)$ will be seen to be a G_s-extendible coloring. Since the number of mind-changes before s is less than $NI(k, n) - 1$, the number of G_s-extendible colorings of H that have been spoiled is less than $NI(k, n) - 1$. Therefore $SPOIL(k, G_s, H, M(H, s))$ is k-colorable, so G_{s+1} will be set to $SPOIL(k, G_s, H, M(H, s))$. Hence $M(H, s)$ cannot be a G_{s+1}-extendible coloring of H, so M cannot compute a local k-coloring of G in the limit. $\boxtimes$

There are stronger recursive conditions that can be imposed on a graph.

Definition: A graph G is *highly recursive* if it is recursive and the function that produces all the neighbors of a given vertex is recursive.

Theorem 8 is true for highly recursive graphs with the same proof. The status of Theorem 8 for decidable graphs, as defined by Bean [1], is unknown.

7. Open Problems

One can add more parameters to the local coloring game. For example, a bound g on the genus of G can be specified as a parameter. In the technical report version of this paper [2] we show that for $g = 0$ (G planar), with $k = 3$, the Explorer has a winning strategy. For $g \geq 1$ all problems associated with such games are open.

Another variation allows the Mapmaker to use m colors where $m > k$. That is, although the explorer is constrained to keep the graph k-colorable, the mapmaker can use $m > k$ colors, where m is an added parameter of the problem. By techniques used in [1] to show that every highly recursive k-colorable graph is recursively $2k$-colorable, one can show that if the Mapmaker can use $2k$ colors than he has a strategy in which he presents only $O(n^2)$ different colorings. For values of m between k and $2k$ it is an open problem to determine who wins.

8. Acknowledgements

We would like to thank Mark Berman, Clyde Kruskal, Kevin Rappoport and Rodica Simion for useful discussions.

REFERENCES

1. Bean, D.R. Effective Coloration. *Journal of Symbolic Logic 41* (1976), 469-480.

2. Beigel, R. and Gasarch, W.I. *The Mapmaker's Dilemma.* The University of Maryland at College Park, TR 2262 (also UMIACS TR 89-59).

3. Beigel, R. and Gasarch, W.I. On the Complexity of Finding the Chromatic Number of a Recursive Graph I: The Bounded Case. *Annals of Pure and Applied Logic 45* (November 1989), 1-38.

4. Beigel, R., Gasarch, W.I., and Hay, L. Bounded Queries Classes and the Difference Hierarchy. *Archive for Math. Logic 29* (December 1989), 69-84.

5. Carstens, H.G. and Pappinghaus, P. Recursive Coloration of Countable Graphs. *Annals of Pure and Applied Logic 25* (1983), 19-45.

6. Carstens, H.G. and Pappinghaus, P. Extensible Algorithms. In *Logic and Machines: Decision Problems and Complexity*, Rodding, Ed., Springer-Verlag, Berlin, 1984. Lecture Notes in Computer Science 171.

7. Gasarch, W.I. *Recursion Theoretic Techniques in Complexity Theory and Combinatorics.* Ph.D thesis, Harvard University 1985.

8. Graham, R., Spencer, J., and Rothchild, A. *Ramsey Theory.* Wiley, 1980.

9. Lakshmipathy, N. and Winklmann, K. Global Graph Problems Tend to be Intractable. *Journal of Computing and System Science 32* (1986), 407-428.

10. Rogers, H. Jr. *Theory of Recursive Functions and Effective Computability.* McGraw Hill, New York, 1967. (Also MIT Press, Cambridge MA., 1987).

11. Soare, R.I. *Recursively Enumerable Sets and Degrees.* Springer Verlag (Omega Series), Berlin, 1987.