Diversity Coloring for Information Storage in Networks

Anxiao (Andrew) Jiang[1]
California Institute of Technology
Pasadena, CA 91125, U.S.A.
e-mail: jax@paradise.caltech.edu

Jehoshua Bruck[1]
California Institute of Technology
Pasadena, CA 91125, U.S.A.
e-mail: bruck@paradise.caltech.edu

Abstract — **We propose a new file placement scheme using MDS codes, and formulate it as the Diversity Coloring Problem. We then present an optimal diversity coloring algorithm for trees.**

I. Introduction

Storing multiple copies of a widely shared file on a network, compared to storing just a single copy, reduces communication costs for file retrieving and improves fault-tolerance [1]. Performance and reliability can be further improved by introducing parity-check information. Some pioneering work on it has been done by Naor and Roth in [3].

If we split a file into K equally long segments, and encode them using an (N, K) MDS code to generate N segments, then the file can be reconstructed by decoding any K of those N segments. We propose the following file placement scheme for a network: store one segment on each node of the network in such a way that every node can reconstruct the file by retrieving and decoding the segments stored within m hops. If we represent the N segments with N different colors, then the file placement scheme can be formulated as the following coloring problem which we call the 'Diversity Coloring Problem':

Diversity Coloring Problem: *Given a graph and N colors, how to assign one color to each vertex, (different vertices can have the same color), such that for every vertex, there are at least K different colors within m hops?* (A coloring that satisfies the above condition is called a diversity coloring. And fixing the parameters K and m and the graph, a diversity coloring is called *optimal* if it uses the smallest parameter N.)

II. Diversity Coloring on Trees

We can prove the following result:

Theorem 1 *Given a tree $G(V, E)$ and parameters N, K and m $(N \geq K)$, there exists a diversity coloring on the tree if and only if for every vertex $v \in V$, there are at least K vertices within m hops from v (including v itself).*

Before we present the diversity coloring algorithm for trees, let's define some terms. Given a tree $G(V, E)$, we use v_0 to denote its root. For any two vertices $u \in V$ and $v \in V$, $d(u, v)$ denotes the number of edges in the unique path connecting u and v (that is, u is $d(u, v)$ hops away from v). For any vertex $v \in V$, we say v is at level $d(v, v_0)$. A vertex u is called a 'quasi-descendant of v' if and only if $u = v$ or u is a descendant of v.

The following is the diversity coloring algorithm for trees.

Algorithm: Diversity Coloring on Tree $G(V, E)$
Input: A tree $G(V, E)$, parameters N, K and m $(N \geq K)$.
Output: A diversity coloring on $G(V, E)$.
Prerequisite: $\forall\, v \in V$, there are at least K vertices within m hops from v (including v itself).

Algorithm:
1. Let $R = \{v_0\}$.
2. while $R \neq \emptyset$ do
{ Arbitrarily select a vertex $v \in R$. Let $R \leftarrow R - \{v\}$.

Inspect the following vertices in order: v, a (any) descendant of v at level $d(v, v_0) + 1$, a (any) descendant of v at level $d(v, v_0) + 2, \cdots$. Stop the inspection as soon as such a vertex w is found—there are less than K different colors within m hops from w. Let C denote the set of colors that are not within m hops from w. Use colors in C to color quasi-descendants of v that haven't yet been colored in the following way: use each color in C at most once; always color vertices at a smaller level before coloring vertices at a greater level; keep coloring until all colors in C are used or until there is no uncolored quasi-descendant of v left.

Let S denote the set of vertices that satisfies the following two conditions: (1) $\forall\, u \in S$, u is a quasi-descendant of v; (2) $\forall\, u \in S$, u has at least one quasi-descendant for which there are less than K different colors within m hops, and u has at least one uncolored quasi-descendant; if we use a to denote a quasi-descendant of u at the smallest level for which there are less than K different colors within m hops, and use b to denote an uncolored quasi-descendant of u at the smallest level, then $d(a, u) + d(b, u) \leq m$.

Let $R \leftarrow R + \{x | x \in S$; no ancestor of x is in $S.\}$.
}
3. Arbitrarily color the remaining uncolored vertices.
□

It can be shown that given the distance matrix, the above algorithm can be carried out with time complexity $O(K|V| + m|V|)$.

By letting $N = K$, we can always use the above algorithm to get an optimal diversity coloring on a tree. Note that for general graphs, an optimal diversity coloring might have $N > K$. For example, if the graph $G(V, E)$ is a ring with no less than $2m + 1$ vertices and $K = 2m + 1$, then we can prove that an optimal diversity coloring on the ring has $N = K + \lceil \frac{|V| \bmod K}{\lfloor \frac{|V|}{K} \rfloor} \rceil$, which is greater than K when $|V|$ is not a multiple of K. And actually when $K \geq m + 2 \geq 3$, for any arbitrarily large integer M $(M \geq K)$, we can show that there exist graphs for which an optimal diversity coloring exists only if $N \geq M$. For more details of this paper, please refer to [2].

References

[1] L. W. Dowdy and D.V. Foster, "Comparative models of the file assignment problem", *Computing Surveys*, vol. 14, no. 2, pp. 287–313, 1982.

[2] A. Jiang and J. Bruck, "Diversity coloring for distributed storage in mobile networks", technical report, http://www.paradise.caltech.edu/papers/etr038.pdf, 2001.

[3] M. Naor and R. M. Roth, "Optimal file sharing in distributed networks", *SIAM J. Comput.*, vol. 24, no. 1, pp. 158–183, 1995.

[1] This work was supported in part by the Lee Center for Advanced Networking at Caltech.