A lower bound on probabilistic algorithms for distributive ring coloring

Moni Naor
IBM Research Division
Almaden Research Center
San Jose, CA 95120

Abstract

Suppose that n processors are arranged in a ring and can communicate only with their immediate neighbors. We show that any probabilistic algorithm for 3 coloring the ring must take at least $\frac{1}{2}\log^* n - 2$ rounds, otherwise the probability that all processors are colored legally is less than $\frac{1}{2}$. A similar time bound holds for selecting a maximal independent set. The bound is tight (up to a constant factor) in light of the deterministic algorithms of Cole and Vishkin [CV] and extends the lower bound for deterministic algorithms of Linial [L].

1 Introduction

In [L] Linial considered the following problem: n processors are connected in a ring and can communicate with their immediate neighbors. They wish to decide on an assignment of one of three colors to each processor, such that no two neighboring processors are assigned the same color (a legal coloring). The question is what is the radius of the neighborhood around each processor which must be considered in order to decide on the coloring. The system is assumed to be completely synchronous, the communication reliable, and there are no limitations on the internal computation of each processor or on the length of the messages sent. The processors are identical, except that each one has a unique id in the range $\{1..n\}$. The id's are assigned in some arbitrary manner, not known initially to the processors. The radius of the neighborhood that affects how a processor is colored is exactly the number of rounds it takes to execute the algorithm.

Linial [L] has shown a lower bound of $\frac{1}{2}\log^* n - 4$ rounds on any deterministic algorithm for coloring the ring with 3 colors. This bound is tight up to a constant factor, since Cole and Vishkin [CV] and Goldberg, Plotkin and Shannon [GPS] have provided an $O(\log^* n)$ round algorithm for achieving it.

In this paper we consider probabilistic algorithms for that task. Each processor is equipped with a perfect source of randomness, and the processor's actions can depend in any way on its coin flips. The performance of an algorithm is now measured in terms of the probability of success as a function of the number of rounds. We show that allowing the processors to flip coins does not help: any algorithm that runs in less than $\frac{1}{2}\log^* n - 2$ rounds has a high probability of failure, i.e. there will be at least two adjacent nodes whose color is the same.

The 3-coloring problem is closely related to the maximal independent set problem: Each processor should decide if it is in the set or not, no two adjacent processors are allowed to be in the set and for every processor not in the set, one of its neighbors must be in the set. Any algorithm for 3-coloring a ring can be translated with 2 additional rounds into one that finds a maximal independent set and vice versa. Thus a lower bound on the 3 coloring problem provides a similar lower bound for the maximal independent set problem. Cole and Vishkin [CV] provided an algorithm for the maximal independent set, and Goldberg, Plotkin and Shannon [GPS] have generalized it to colorings of various degree bounded graphs.

In [BNN] the number of bits of communication required in order to achieve 3–coloring is investigated. (I.e. messages are 1-bit long.) It is shown that in any deterministic algorithm it must be $\Omega(\log n)$ which is tight by the [CV] algorithm. Interestingly, for randomized algorithms it is $\Theta(\sqrt{\log n})$.

2 The lower bound

Theorem 2.1 *Let $k = n^{\frac{1}{3}}$. Any probabilistic algorithm for 3-coloring a ring of n processors that takes less than $t = \frac{1}{2}\log^* n - b - 2$ rounds, has probability at most $(1 - \frac{1}{\log^{(b)} n})^{\frac{k}{2t}} + \frac{2t}{k}$ to produce a legal coloring.*

Proof: In any probabilistic algorithm it can be assumed that the processors first make their random choices and from then on act deterministically. Since the processors actions are determined by the order of the id's and the random numbers selected in the system, an algorithm that runs in t rounds can be simulated by one where the processors send to each other their id number and their random selections. After t rounds each processor knows the random numbers selected by $2t+1$ processors: itself and the $2t$ processors that are of distance at most t from it. Based on this information it decides on a color. Let D be the range from which the processors make their random selection and let $R = D \times \{1..n\}$. Any $r \in R$ corresponds to a selection for the radnom choices of a processor concatenated with its id. After t rounds the information any processor has corresponds to a vector $(r_1, r_2, \ldots r_{2t+1})$ where $r_i \in R$. Thus, any t rounds algorithm induces a 3-coloring of the vectors $\{(r_1, r_2, \ldots r_{2t+1}) | r_i \in R\}$ by associating a vector with the color the algorithm assigns a processor with neighborhood information represented by the vector.

We concentrate on a segment of $k+2t$ consecutive processors on the ring. Suppose that the adversary assigns each processor an id by choosing it independently from $\{1..n\}$. With probability at least $1-\frac{2t}{k}$ all the id's in the segment are unique. This is true, since the probability that at least two processors choose the same id is bounded by $\binom{k+2t}{2}$ times the probability that two specific processors chose the same id which is $\frac{1}{n}$ and $\binom{k+2t}{2}\cdot\frac{1}{n}\leq\frac{2t}{k}$. If the id's chosen are not unique, we consider it as if the algorithm "won". The lower bound of the theorem will follow if we can bound by $(1-\frac{1}{\log^{(b)} n})^{\frac{k}{2t}}$ the probability that the processors of the segment choose a legal coloring in case each processor i selects at random $r_i\in\{1,\ldots,R\}$. This is true since for any two events A and B, $Pr[A|B]\leq Pr[A]+Pr[\overline{B}]$. In our case A is the event that the algorithm succeeds and B is the event that the adversary assigns unique id's to the segment.

Consider the directed graph $G_{R,2t+1}$: each nodes corresponds to a vector $(r_1,r_2,\ldots r_{2t+1})$ such that $r_i\in R$; node $(r_1,r_2,\ldots r_{2t+1})$ is connected to node $(s_1,s_2,\ldots s_{2t+1})$ iff $r_i=s_{i+1}$ for $2\leq i\leq 2t$. The edge in this case is called $(r_1,r_2,\ldots,r_{2t+1},s_{2t+1})$ (or equivalently $(r_1,s_1,\ldots,s_{2t+1})$).

This graph was used in the lower bound proof for deterministic algorithms in [L]. It was shown that any algorithm that colors the ring must define a legal coloring of $G_{R,2t+1}$ and by deriving a bound on the chromatic number of $G_{R,2t+1}$ as a function of t, the lower bound was shown. Here the situation is more complicated, since the ring coloring algorithm does not necessarily define a legal coloring of $G_{R,2t+1}$: the probability of selecting an edge with similarly colored end points might be small.(We call such an edge monochromatic.) Instead, we will show a lower bound on the fraction of monochromatic edges.

The process of selecting the random numbers by the $k+2t$ processors in the segment corresponds to selecting a (not necessarily simple) path of length k in the graph $G_{R,2t+1}$: if the random numbers selected are $r_1,r_2,\ldots r_{k+2t}$, then the path selected is $v_1,v_2,\ldots v_{k+2t}$ where $v_i=(r_{i-t},\ldots r_i,\ldots r_{i+t})$. Let $z_1,z_2,\ldots z_k$ be the edges of this path. Each z_i is uniformly distributed over the edges of $G_{R,2t+1}$, and z_i is independent of all z_j for j such that $|j-i|\geq 2(t+1)$. Therefore we have $\frac{k}{2(t+1)}$ random variables $z_1,z_{2t+2},\ldots z_k$ that are mutually independent and each is a random choice of an edge in $G_{R,2t+1}$.

For any coloring (not necessarily legal) of $G_{R,2t+1}$ we call an edge *monochromatic* if both of its endponits are assigned the same color. Let p be the probability that an edge chosen at random in $G_{R,2t+1}$ is monochromatic. For a randomly chosen path of length k in $G_{R,2t+1}$,

$$\begin{aligned} Prob[\text{ some edge is monochromatic}] &\geq \\ Prob[\text{at least one of } \{\mathrm{z}_1,\mathrm{z}_{2t+2},\ldots\mathrm{z}_k\} \text{ is monochromatic}] &\geq 1-(1-p)^{\frac{k}{2t+2}} \end{aligned}$$

If we show that for $t=\frac{1}{2}\log^* n-b-2$, for all 3 colorings of $G_{R,2t+1}$, $p\geq\frac{1}{\log^{(b)}}$, then the probability that any t rounds algorithm succeeds is at most $(1 - \frac{1}{\log^{(b)} n})^{\frac{k}{2t}} + \frac{2t}{k}$.

Consider the series of graphs $G_{R,1}, G_{R,2}, \ldots, G_{R,2t+1}$, where $G_{R,i}$ is defined similarly to $G_{R,2t+1}$. Let $c_t = 3$ and $c_i = 2^{c_{i+1}}$. Define $p_1, p_2, \ldots, p_t$ by setting $p_1 = \frac{1}{c_1}$ and $p_{i+1} = \frac{p_i^2}{2c_{i+1}}$. We will show that p_i is such that for every coloring of $G_{R,i}$ with c_i colors $Prob[\text{random edge in } \mathrm{G_{R,i}} \text{ is monochromatic}] > p_i$.

Proposition 2.1 *For any coloring of $G_{R,1}$ with c_1 colors,*

$$Prob[\text{random edge in } \mathrm{G_{R,1}} \text{ is monochromatic}] \geq p_1 = \frac{1}{c_1}$$

Proof: $G_{R,0}$ is actually a complete graph with self loops. Therefore, in order to minimize the probability that two nodes have the same colors, all color classes should be of the same size, and we get that $p_1 = \frac{1}{c_1}$.□

Lemma 2.1 *Assume that for any coloring of $G_{R,i}$ with c_i colors, the probability that a random edge is monochromatic is at least p_i. Then for any coloring of $G_{R,i+1}$ with c_{i+1} colors*

$$Prob[\text{random edge in } \mathrm{G_{R,i+1}} \text{ is monochromatic}] \geq p_{i+1} = \frac{p_i^2}{2 \cdot c_{i+1}}$$

Proof: The nodes of $G_{R,i+1}$ correspond naturally to the edges of $G_{R,i}$. Selecting a random edge in $G_{R,i+1}$ corresponds to selecting a path of length two in $G_{R,i}$. If we can show that for every coloring of the edges of $G_{R,i}$ with c_{i+1} colors the probability that two edges in a random path have the same color is at least $\frac{p_i^2}{2 \cdot c_{i+1}}$, then we are done.

Given a coloring of the edges of $G_{R,i}$ with c_{i+1} colors we define a corresponding coloring of the nodes of $G_{R,i}$ with $c_i = 2^{c_{i+1}}$ colors by the following procedure:

For a node v call a color c *frequent* for v if at least a fraction $f_{i+1} = \frac{p_i}{2c_{i+1}}$ of the edges starting at v are colored c. An edge $e = (v, u)$ whose color is frequent for v is called *frequent*. Otherwise it is called *infrequent*. Let S_v be the set of frequent colors of v and let $C_v \in \{0,1\}^{c_{i+1}}$ be the characteristic vector of S_v. Node v is assigned the color C_v.

This is a refinement of the coloring used in [L], where the color a node is assigned is the characteristic vector of the set of all colors that meet that node.

Claim 2.1 *The fraction of infrequent edges is at most $f_{i+1} \cdot c_{i+1} = \frac{p_i}{2}$.*

Proof: For every node, at most f_{i+1} of the edges starting from it colored by any color not in S_v. Thus, the fraction of infrequent edges is at most $c_{i+1} \cdot f_{i+1} = \frac{p_i}{2}$. □

Claim 2.2 *For every edge coloring of $G_{R,i+1}$ with c_{i+1} colors and the corresponding node coloring with $c_i = 2^{c_{i+1}}$ colors, at least $\frac{p_i}{2}$ of the edges are both frequent and monochromatic.*

Proof: By assumption, p_i of the edges are monochromatic and by the previous claim at most $\frac{p_i}{2}$ of the edges are infrequent. Thus, at least $\frac{p_i}{2}$ of the edges are both monochromatic and frequent. $\Box$

Fix a coloring of the edges of $G_{R,i}$ and its corresponding node coloring. Suppose that a path of length 2 is randomly selected at $G_{R,i}$. If the first edge $e = (v, u)$ is monochromatic and frequent, then the color of e is frequent at u as well as at v, since (v, u) being monochromatic means that the lists of frequent colors at v and u are the same. Therefore, there is probability at least f_{i+1} that the second edge (starting from u) will be colored as $e = (v, u)$. Thus the probability that both events occur is at least $\frac{p_i}{2} \cdot f_{i+1} = \frac{p_i}{2} \cdot \frac{p_i}{c_{i+1}} = \frac{p_i^2}{2c_{i+1}}$ concluding the proof og the lemma. $\Box$

Applying the lemma t times we get that

$$Prob[\text{random edge } z_j \textit{ in } G_{R,2t+1} \text{ is monochromatic}] > p_t$$

By definition

$$p_t > \left(\frac{p_1}{2c_1}\right)^{2^t} = \frac{1}{(2c_1^2)^{2^t}}$$

Now,

$$c_1 = 2^{2^{2^{\cdot^{\cdot^{\cdot^{2^3}}}}}} \Big\} 2t+1.$$

Thus, if $t = \frac{1}{2}\log^* n - b - 2$ for some $b > 0$, then $p_t > \frac{1}{\log^{(b)} n}$ and we get our theorem.$\Box$

A different proof for the fact that there are many monochromatic edges was suggested by Noga Alon (personal communication): It relies on the fact that there is lower bound on the chromatic number of $G_{R,i}$, and thus for any large enough subset of $\{1..R\}$, the induced subgraph contains at least one edge which is monochromatic.

Acknowledgements

Stimulating discussions with Richard Karp, Vijaya Ramachandran and Danny Soroker are gratefully acknowledged. I would like to thank the two Josephs, Yossi Azar and Seffi Naor and the anonymous referee for carefully reading the manuscript.

References

[BNN] A. Bar-Noy, J. Naor and M. Naor, *One-bit algorithms*, Ditributed Computing, 4 (1990), pp. 3-8.

[CV] R. Cole and U. Vishkin, *Deterministic coin tossing with applications to optimal parallel list ranking*, Information and Control, 70 (1986), pp. 32-53.

[GPS] A. Goldberg, S. Plotkin and G. Shannon, *Efficient parallel algorithms for* $(\Delta + 1)$ *– coloring and maximal independent set problems*, Proc. 19th ACM Symp. on Theory of Computing, pp. 315-324, 1987.

[L] N. Linial, *Distributive graphs algorithms – global solutions from local data*, Proc. 28th IEEE Foundations of Computer Science Symposium, pp. 331-335, 1987.