SIAM J. COMPUT.
Vol. 21, No. 1, pp. 193–201, February 1992

015

LOCALITY IN DISTRIBUTED GRAPH ALGORITHMS*

NATHAN LINIAL†

Abstract. This paper concerns a number of algorithmic problems on graphs and how they may be solved in a distributed fashion. The computational model is such that each node of the graph is occupied by a processor which has its own ID. Processors are restricted to collecting data from others which are at a distance at most t away from them in t time units, but are otherwise computationally unbounded. This model focuses on the issue of *locality* in distributed processing, namely, to what extent a global solution to a computational problem can be obtained from locally available data.

Three results are proved within this model:

- A 3-coloring of an n-cycle requires time $\Omega(\log^* n)$. This bound is tight, by previous work of Cole and Vishkin.
- Any algorithm for coloring the d-regular tree of radius r which runs for time at most $2r/3$ requires at least $\Omega(\sqrt{d})$ colors.
- In an n-vertex graph of largest degree Δ, an $O(\Delta^2)$-coloring may be found in time $O(\log^* n)$.

Key words. distributed algorithms, graph theory, locality, lower bounds

AMS(MOS) subject classifications. 05C35, 68R10, 68Q99

1. Introduction.

In distributed processing all computations are made based on local data. The aim of this paper is to bring up limitations that follow from this local nature of the computation. Note that within the various computational models for parallel computers this difficulty is specific to the distributed model. Shared memory allows for fast dissemination of data, but no such means exist when dealing with distributed systems.

In the present paper we are mostly interested in proving lower bounds, and therefore assume a powerful version of the distributed model: Each node of the undirected graph $G = (V, E)$ is occupied by a processor. Computation is completely synchronous and reliable. At each time unit a processor may pass messages to each of its neighbors, and message size is unrestricted. Also, any computations carried out by individual processors take one time unit and are not restricted in any way. This paper is only concerned with the radius of the neighborhood around each node from which data may be collected, this radius being the only significant parameter in this model, as we later elaborate. Of interest is the time complexity of various "global" functions of G, and the concrete examples are coloring and finding maximal independent sets. Thus the theme of this paper is how local data may be utilized to find globally defined solutions.

Before we proceed, symmetry-breaking has to be addressed (see [JS] and the references therein for literature on symmetry-breaking). It is well known that most functions cannot be computed in a distributed fashion by anonymous processors, even for very simple graphs G. This impossibility usually results from symmetries that G may have. Such symmetries are usually broken by means of either randomization or the use of IDs, and the present paper is concerned only with the latter. Thus we assume that there is a mapping ID from the set of vertices V to the positive integers.

* Received by the editors July 17, 1989; accepted for publication (in revised form) December 4, 1990.

† Computer Science Department, Hebrew University, Jerusalem 91904, Israel. Part of this work was done while the author was visiting the IBM Research Center, Almaden, California, and the Department of Computer Science, Stanford University, Stanford, California 94305. This work was supported in part by grants from the Israel–US Binational Science Foundation and the Israeli Academy of Sciences.

In most cases ID is a bijection onto $1, \cdots, \mid V \mid$. It is assumed that at time zero the processor occupying a node in G knows the ID of that node. Incidentally, all our lower bounds hold even if every processor knows in advance what the graph G is, and only the labeling function ID is unknown.

It is clear that the present model allows us to compute every function of G in time $O(\text{diameter}(G))$. After this amount of time every processor obtains complete knowledge of both G and ID. The problem is thus solved if, in the memory of each processor a solution for the entire problem is prestored, for every possible labeling. Our concern is therefore only with time complexities below $\text{diam}(G)$. The major question we raise is which functions may be computed by a nontrivial algorithm in this model, in the sense that they can be computed faster than $\text{diam}(G)$.

The model proposed here is, of course, of purely theoretical interest. Of the many difficulties arising in distributed processing, it focuses only on transforming local data into a global solution. Further research into this model may help classify problems as either locally computable (solvable in time shorter than $\text{diam}(G)$) or not. One may also look for bounds on run times which depend on graph parameters other than the diameter. In accordance with the theory of the complexity class NC, it seems natural to investigate graph problems whose time complexity in this model is polylogarithmic in the number of vertices.

Here are our main results:

(1) Finding a maximal independent set distributively in a labeled n-cycle, requires time $\Omega(\log^* n)$. This bound is tight in view of the $O(\log^* n)$ algorithm by Cole and Vishkin [CV]. (Technically, their result was stated in the PRAM model, but it extends without change to the distributed model as well.) Our proof relies on the interesting construct of *neighborhood graphs*. An alternative proof based on the Ramsey theorem was found by a number of other investigators in the area [A].

(2) Coloring trees: Let T be the d-ary tree of height r. In time $2r/3$, it is impossible to color T in fewer than $\sqrt{d}$ colors. Note in contrast the algorithm by Goldberg and Plotkin [GP], which shows that if every node in T "knows its parent in the tree," i.e., a consistent orientation from the root outwards is given, then a 3-coloring can be found in time $O(\log^* n)$. Though stated for the PRAM model, it is easy to see that this result of [GP] applies to the distributed model as well.

(3) In a labeled graph of order n with maximal degree Δ, it is possible to find an $O(\Delta^2)$-coloring in time $(1 + o(1)) \log^* n$. This was previously shown in [GP] only in the case of constant Δ.

Our terminology is standard: a *k-labeling* of a graph $G = (V, E)$ is 1:1 mapping $f : V \to \{1, \cdots, k\}$. In case $k = \mid V \mid$ a k-labeling is called a *labeling*. Given a k-labeling, G is said to be k-labeled, etc. Logarithms are to base 2. The k times iterated logarithm is denoted by $\log^{(k)} x$, i.e., $\log^{(1)} x := \log x$ and $\log^{(k)} x = \log(\log^{k-1} x)$. The least integer k for which $\log^{(k)} x < 1$ is denoted by $\log^* x$.

2. Lower bound on finding a maximal independent set in a cycle. In [CV] a very nice algorithm was presented to find a maximal independent set of vertices (= MIS) in the n-cycle C_n in time $\log^* n$. In this section we show that this is optimal even in the present model, where computation takes no time. The algorithm presented in §4 also achieves this time bound.

A basic observation is that in the present model there is no loss of generality in assuming that processing proceeds by first collecting all data and then deciding. That is, at time t each processor knows the labeling of all nodes at distance t or less away. Also known are all edges between these nodes, except for edges both endpoints

of which are at distance exactly t. Note that no further information can reach a processor by time t. This allows us to view the problem in purely combinatorial terms.

Let us state our theorem.

THEOREM 2.1. *A synchronous distributed algorithm which finds a maximal independent set in a labeled n-cycle must take at least $\frac{1}{2}(\log^* n - 1)$ units of time. An algorithm of the same class which colors the n-cycle with three colors requires time at least $\frac{1}{2}(\log^* n - 3)$. The same bounds hold also for randomized algorithms.*

Proof. The proof holds even under the assumption that there is a consistent notion of clockwise orientation common to all processors. Given an algorithm which finds a maximal independent set in the n-cycle endowed with a clockwise orientation, it is easily seen that in one more timestep, the cycle may be 3-colored. The lower bound is established for 3-coloring.

Coming back to the previous observation, at time t the data known to a processor P is an ordered list of $2t+1$ labels, starting t places before it, through its own and on to the next t labels. Let V be the set of all vectors $(x_1, \cdots, x_{2t+1})$ where the x_i are mutually distinct integers from $\{1, \cdots, n\}$. The algorithm is nothing but a mapping c from V into $\{1, 2, 3\}$.

Let us denote by $B_{t,n}$ the graph whose set of vertices is V. All edges of $B_{t,n}$ are given by:

$$(x_1, \cdots, x_{2t+1}) \quad \text{and} \quad (y, x_1, \cdots, x_{2t})$$

are neighbors for all $y \neq x_{2t+1}$. So $B_{t,n}$ has $n(n-1)\cdots(n-2t)$ vertices and is regular of degree $2(n-2t-1)$. Note that the mapping $c : V \to \{1, 2, 3\}$ is, in fact, a proper 3-coloring of $B_{t,n}$. For suppose that c assigns

$$(x_1, \cdots, x_{2t+1}) \quad \text{and} \quad (y, x_1, \cdots, x_{2t})$$

the same color. Then the 3-coloring algorithm for the n-cycle fails in case the labeling happens to contain the segment:

$$y, x_1, x_2, \cdots, x_{2t+1}.$$

The proof follows now by standard graph-theoretic arguments which show that the chromatic number $\chi(B_{t,n})$ of $B_{t,n}$ satisfies

$$\chi(B_{t,n}) = \Omega(\log^{(2t)} n),$$

the $2t$ times iterated logarithm of n. Therefore, for $\chi(B_{t,n})$ to be at most 3, we must have $t = \Omega(\log^* n)$.

The lower bound on $\chi(B_{t,n})$ is proved, using a family of digraphs $D_{s,n}$ closely related to $B_{t,n}$. The vertices of $D_{s,n}$ are all sequences $(a_1, \cdots, a_s)$ with $1 \leq a_1 < a_2 < \cdots < a_s \leq n$. The outneighbors of $(a_1, \cdots, a_s)$ are all vertices of the form $(a_2, \cdots, a_s, b)$ with $a_s < b \leq n$. Note that $B_{t,n}$ contains the underlying graph of $D_{2t+1,n}$, so in particular $\chi(B_{t,n}) \geq \chi(D_{2t+1,n})$.

Given a digraph $H = (V, E)$ its *dilinegraph* $DL(H)$ is a digraph whose vertex set is E with (u, w) an edge if $head_H(u) = tail_H(w)$. The relation between the digraphs $D_{s,n}$ is given by Proposition 2.1.

PROPOSITION 2.1. *$D_{1,n}$ is obtained from the complete graph of order n by replacing each edge by a pair of edges, one in each direction, and $D_{s+1,n} = DL(D_{s,n})$ for all $s \geq 1$.*

Proof. The statement concerning $D_{1,n}$ is just the definition. For the other claim, identify the edge connecting $(x_1, \cdots, x_s)$ and $(x_2, \cdots, x_s, y)$ in $D_{s,n}$ with the vertex $(x_1, \cdots, x_s, y)$ in $V(D_{s+1,n})$ and check that the adjacency relationship in $D_{s+1,n}$ is that of $DL(D_{s,n})$. □

The bound on $\chi(D_{s,n})$ is derived from the following simple and well-known proposition.

PROPOSITION 2.2. *For a digraph G,*

$$\chi(DL(G)) \geq \log \chi(G).$$

Proof. A k-coloring of $DL(G)$ may be thought of as a mapping $\Psi : E(G) \to \{1, \cdots, k\}$ such that if $u, w \in E(G)$ and $head(u) = tail(w)$, then $\Psi(u) \neq \Psi(w)$. Now vertex-color G by associating with node x the set

$$c(x) = \{\Psi(u) \mid x = tail(u)\}.$$

This is easily seen to be a 2^k vertex-coloring of G. Indeed if $u = (x, y) \in E(G)$, then $\Psi(u) \in c(x)$ but $\Psi(u) \notin c(y)$, or else Ψ is improper. Therefore $c(x) \neq c(y)$. □

The main claim can be derived now. If an MIS can be found in time $t - 1$, then necessarily

$$3 \geq \chi(B_{t,n}).$$

But

$$\chi(B_{t,n}) \geq \chi(D_{2t+1,n}) \geq \log^{(2t)} n.$$

So

$$2t \geq \log^* n - 1, \qquad t \geq \frac{1}{2}(\log 2^* n - 1),$$

as claimed.

The claim on randomized algorithms is proved in Corollary 2.1 below. □

In contrast with the low time complexity of 3-coloring, we can show that for an even n, finding a 2-coloring of C_n requires time $\Omega(n)$.

THEOREM 2.2. *A synchronous distributed algorithm which 2-colors a labeled $2n$ cycle with labels from $\{1, \cdots, 2n\}$ must take at least $n - 1$ units of time.*

Proof. Now the lowest t has to be found such that $B_{t,2n}$ is bipartite. But even for $t = n - 2$, the graph $B_{t,2n}$ contains an odd cycle:

$$(1, \cdots, 2t + 1), (2, \cdots, 2t + 2), (3, \cdots, 2t + 3), (4, \cdots, 2t + 3, 1),$$

$$(5, \cdots, 2t + 3, 1, 2), \cdots, (2t + 3, 1, \cdots, 2t), (1, \cdots, 2t + 1).$$

The claim follows. □

Let us point out that for an even n the last theorem implies that finding a *maximum* independent set requires time $\lceil n/2 \rceil - 1$. It is easily verified that the same is true for odd n as well.

We want to elaborate on the method developed here and point out its general features. Given a graph $G = (V, E)$ of order n, and $t \geq 1$, the *t-neighborhood graph of*

$G, N_t(G)$ is constructed as follows: For every $x \in V$ let $S_t(x)$ be the subgraph of G spanned by those vertices y whose distance from x is at most t. For every x consider all the n-labelings of $S_t(x)$.

Every such labeling Ψ is a node in $N_t(G)$. Let $\Psi_1 : V_t(x) \to \{1, \cdots, n\}$ and $\Psi_2 : V_t(y) \to \{1, \cdots, n\}$ be two of these vertices of $N_t(G)$. They are taken to be neighbors in $N_t(G)$ if $[x, y] \in E(G)$ and there is a labeling $\Phi : V(G) \to \{1, \cdots, n\}$ such that

$$\Phi \mid S_t(x) = \Psi_1$$

and

$$\Phi \mid S_t(y) = \Psi_2.$$

Some easy observations regarding $N_t(G)$ follow.

PROPOSITION 2.3. *Neighborhood graphs have the following properties:*

(1) $\chi(N_t(G))$ *is the least number of colors with which* G *may be colored distributively on time* t.

(2) $\chi(N_t(G)) = \chi(G)$ *for* $t \geq \operatorname{diam}(G)$.

(3) $\chi(N_t(G))$ *is nonincreasing with* t.

(4) *For* $G = C_n$, *the graph* $B_{t,n}$ *is obtained from* $N_t(C_n)$ *by identifying vertices in* $N_t(C_n)$ *with identical sets of neighbors. In particular,*

$$\chi(B_{t,n}) = \chi(N_t(C_n)).$$ □

As in the case of the cycle, it is helpful to identify nonadjacent vertices with an identical set of neighbors. Such an operation is called a *reduction.* Clearly it does not change the chromatic number, while it may significantly simplify the graph. Neighborhood graphs seem to be very interesting and promising constructs. Unfortunately, the only case where we managed to calculate their graphical parameters is that of a cycle. Particularly interesting are cases where in G the graph $S_t(x)$ is independent of x, as is the case, for example, with vertex transitive graphs.

Proposition 2.3 yields the following easy but interesting corollary.

COROLLARY 2.1. *In the present model the time required to properly color a given graph with a given number of colors cannot be reduced by using randomization.*

Proof. The most general form of a randomized algorithm in the present model allows each processor to precede each round of computation with any number of coin flips, the outcomes of which are then passed to its neighbors along with all other information, as in the deterministic version. Consider the following different class $\mathcal{C}$ of randomized graph algorithms: A sufficiently long string of bits σ is first selected at random by some random source (not necessarily one of the processors in the graph) and then announced to all processors. From this point on, the algorithm proceeds in the usual deterministic way, where each processor gets to see its labeled t-neighborhood (as well as the random string σ, of course). It is not hard to see that any randomized algorithm which can be performed in the present model can be simulated by an algorithm in $\mathcal{C}$. This is because σ can be made long enough to include as many random bits as may be required in any run of the original algorithm by any processor at all. In the algorithm from $\mathcal{C}$, processors get the advantage of being informed of all these random bits, and in advance, which can only help.

Since we are interested in a lower bound, there is no loss of generality in considering only algorithms from the class $\mathcal{C}$. Such an algorithm running in time t entails a

function which computes a color for a vertex from its t-neighborhood and the random σ. Fix a graph G, an integer t, and two adjacent vertices x and y in G. Consider the adjacent vertices ξ and η in $N_t(G)$ which represent compatible labelings of t-neighborhoods of x, y, and a random string σ. The color is chosen based on seeing the labeled neighborhood ξ, and the random σ must differ from the one given for η, σ. In other words, the coloring function constitutes a valid coloring for a graph which is the disjoint union of copies of $N_t(G)$, one copy per each σ. This graph has, of course, the same chromatic number as $N_t(G)$, and the claim follows. □

3. Lower bound on coloring trees.

THEOREM 3.1. *Let $T = T_{d,r}$ be the rooted d-regular tree of radius r. Any synchronous distributed algorithm running in time $\leq \frac{2}{3}r$ cannot color T by fewer than $\frac{1}{2}\sqrt{d}$ colors. The same bound holds for randomized algorithms as well.*

Proof. There are d-regular graphs $R_{d,n}$ on n vertices of chromatic number $\geq \frac{1}{2}\sqrt{d}$, where all cycles have length $\geq (4/3)(\log n/\log(d-1))$ (see [LPS]). Consequently for $t < (2/3)(\log n/\log(d-1))$, the graph $N_t(T_{d,r})$ contains a copy of a reduction of $N_t(R_{d,n})$. Therefore

$$\chi(N_t(T_{d,r})) \geq \chi(N_t(R_{d,n})) \geq \chi(R_{d,n}) \geq \tfrac{1}{2}\sqrt{d},$$

and so $T_{d,r}$ cannot be colored with fewer than $\frac{1}{2}\sqrt{d}$ colors in time t. The conclusion follows now on observing that for $T_{d,r}$,

$$\frac{\log n}{\log(d-1)} \geq r.$$

Corollary 2.1 establishes the bound for randomized algorithms as well. □

Two remarks are in order now: It is probably possible to improve the lower bound of $\frac{1}{2}\sqrt{d}$ to $\Omega(d/\log d)$ by using an appropriate random graph rather than Ramanujan graphs (e.g., [B, §11.4]). It is shown in the next section that for a d-regular graph, an $O(d^2)$-coloring can be found in time $O(\log^* n)$. The gap between $(d/\log d)$ and d^2 is quite intriguing and is closely related to the complexity of finding an MIS distributively, as we explain below. Also, it is not clear how the number of colors sufficient to color the tree goes down as t grows from $(2r/3)$ to $2r$, where already two colors suffice.

4. An $O(\log^* n)$ algorithm for $O(\Delta^2)$-coloring.

This section contains some positive results on what can be achieved distributively in this model. It is shown how to find an $O(\Delta^2)$-coloring in time $O(\log^* n)$. The first proof is based on the existence of certain hypergraphs for which no explicit construction is known. However, as Karloff pointed out [K], a slightly weaker, though constructive, set system suffices to achieve this goal.

THEOREM 4.1. *Let G be a graph of order n and largest degree Δ. It is possible to color G with $5\Delta^2 \log n$ colors in one unit of time distributively. Equivalently,*

$$\chi(N_1(G)) \leq 5\Delta^2 \log n.$$

Proof. We need some combinatorial preparation.

LEMMA 4.1. *For integers $n > \Delta$, there is a family J of n subsets of $\{1, \cdots, 5\lceil \Delta^2 \log n \rceil\}$ such that if $F_0, \cdots, F_\Delta \in J$, then*

$$F_0 \not\subseteq \bigcup_1^\Delta F_i.$$

Proof. The existence of such families was considered in the literature [KSS], [EFF], and follows, e.g., from Theorem 3.1 in [EFF]. This lemma is simple enough, though, for a proof to be reproduced here. To prove the lemma, let $m = 5\lceil \Delta^2 \log n \rceil$ and consider a random collection J of n subsets of $\{1, \cdots, m\}$ which is constructed as follows: For $1 \le x \le m$ and $1 \le i \le n$, let $Pr(x \in S_i) = (1/\Delta)$. All the decisions on whether $x \in S_i$ are made independently.

We claim that there is a selection of such a family for which the lemma holds. The probability that for a given $1 \le x \le m$ and given $F_0, \cdots, F_\Delta \in J$ (i.e., $F_i = S_{\mu_i}$ for appropriately chosen indices), there holds

$$x \in F_0 \setminus \left(\bigcup_1^\Delta F_i \right)$$

is

$$\frac{1}{\Delta} \left(1 - \frac{1}{\Delta}\right)^\Delta \ge \frac{1}{4\Delta}.$$

Therefore, the probability that $F_0 \subseteq \cup_1^\Delta F_i$ is at most $(1 - (1/4\Delta))^m$. The number of ways for choosing $F_0, \cdots, F_\Delta$ is $(\Delta + 1)\binom{n}{\Delta+1}$. Therefore if

$$\left(1 - \frac{1}{4\Delta}\right)^m (\Delta + 1) \binom{n}{\Delta + 1} < 1,$$

then there is a selection of a family J which satisfies the lemma. Standard estimates show that this holds when

$$m > 5\Delta^2 \log n. \qquad \square$$

To prove the theorem, fix a family $J = \{F_1, \cdots, F_n\}$ as in the lemma. Consider a vertex with label i whose neighbors' labels are $j_1, \cdots, j_d$ where $d \le \Delta$. Since

$$F_i \not\subseteq \bigcup_{\nu=1}^d F_{j_\nu},$$

there is a $1 \le x \le m$ with $x \in F_i \setminus (\bigcup_{\nu=1}^d F_{j_\nu})$. The color of this vertex is x. It is easily verified that this is a proper m-coloring of G. $\square$

The same proof yields, more generally, the following corollary.

COROLLARY 4.1. *Let G be a graph whose vertices are properly colored with k colors and whose largest degree is Δ. It is possible to color G with $5\Delta^2 \log k$ colors in one unit of time distributively.*

By iterating the coloring given by the corollary $\log^* n$ times, a $10\Delta^2 \log \Delta$-coloring is obtained. To eliminate the $\log \Delta$ term we need a result of the same type as Lemma 4.1, only in that range.

LEMMA 4.2. *Let q be a prime power. Then, there is a collection J of q^3 subsets of $\{1, \cdots, q^2\}$ such that if $F_0, \cdots, F_{[(q-1)/2]} \in J$ then $F_0 \not\subseteq \bigcup_1^{[(q-1)/2]} F_i$.*

Proof. Use Example 3.2 in [EFF] with $d = 2$. □

Select q to be the smallest prime power with $q \geq 2\Delta + 1$. There is certainly one with $4\Delta + 1 \geq q \geq 2\Delta + 1$. This construction transforms an $O(\Delta^3)$-coloring to an $O(\Delta^2)$-coloring as in the proof of Theorem 4.1.

Theorem 4.2 now follows.

THEOREM 4.2. *Let G be a labeled graph of order n with largest degree Δ. Then in time $O(\log^* n)$ it is possible to color G with $O(\Delta^2)$ colors in a distributive synchronous algorithm.* □

The previous proof is nonconstructive in that there is no known explicit construction as good as that which Lemma 4.1 guarantees. However, as Karloff pointed out [K], the explicit geometric construction in Example 3.2 of [EFF] enables us, in the same way, to reduce a k-coloring to an $O((\Delta^2 \log^2 k))$-coloring in one timestep. (The previous proof reduces a k-coloring to an $O((\Delta^2 \log k))$-coloring in a step.) The rest of the proof remains unchanged, yielding the same results with only slightly worse constants.

Proposition 3.4 of [EFF] sets a bound on the power of the present method, showing that it does not enable one to color with fewer than $\binom{\Delta+2}{2}$ colors. That proposition shows that set systems of the type that would allow further reduction of the number of colors do not exist. Other algorithms may still be capable of coloring with fewer colors. It would be interesting to decide whether this quadratic bound can be improved when time bounds rise from $O(\log^* n)$ to polylog, for instance.

5. Some final remarks. It is well known now how to find a maximal independent set in a graph by means of an NC algorithm ([KW], [ABI], [L]). The algorithm in [L] can even be viewed as a randomized distributed algorithm. However, an algorithm which is both deterministic and distributed still eludes us. In [GP] it is shown how to find an MIS in time $\log^* n$ for graphs of bounded degree. It is not clear what the situation is for unbounded degrees. In particular, can it always be found in polylogarithmic time in the present model? This, if true, would be a significant improvement over the above-mentioned studies.

A standard trick (e.g., [L]) allows us to transform an efficient MIS algorithm to one for $(\Delta + 1)$-coloring: Take the cartesian product of G with $K_{\Delta+1}$. It is easily verified that $(\Delta+1)$-colorings of G and MISs of $G \times K_{\Delta+1}$ are in a natural 1:1 correspondence. It is therefore particularly interesting to find out the best time complexity in terms of n for finding a $(\Delta + 1)$-coloring, and in particular whether polylogarithmic time suffices.

The fact that randomness does not help in coloring (Corollary 2.1) is thought-provoking: Getting a deterministic polylog-time algorithm for MIS seems hard, though simple randomized distributed algorithms are known. It is an intriguing problem to classify distributed graph problems according to how much randomization can help in solving them.

6. Acknowledgments. Comments recieved by a number of colleagues helped improve this paper in a significant way. A comment by Noga Alon helped simplifying the proof of Theorem 2.1. Howard Karloff's comment on the proof of Theorem 4.1 is recorded in the text, and a similar remark was made by Noga Alon as well. That randomization does not help came up in discussions with Dror Zernik. Helpful comments were also received from the referees. I am grateful to all of them.

REFERENCES

[A] B. AWEBUCH, Private communication.

[ABI] N. ALON, L. BABAI, AND A. ITAI, *A fast and simple randomized algorithm for the maximal independent set problem*, J. Algorithms, 7 (1986), pp. 567–583.

[B] B. BOLLOBAS, *Random Graphs*, Academic Press, New York, 1985.

[CV] R. COLE AND U. VISHKIN, *Deterministic coin tossing and accelerating cascades: micro and macro techniques for designing parallel algorithms*, in Proc. 18th ACM Symposium on Theory of Computing, 1986, pp. 206–219.

[EFF] P. ERDÖS, P. FRANKL AND Z. FÜREDI, *Families of finite sets in which no set is covered by the union of r others*, Israel J. Math., 51 (1985), pp. 79–89.

[GP] A. V. GOLDBERG AND S. A. PLOTKIN, *Efficient parallel algorithms for* $(\Delta + 1)$*-coloring and maximal independent set problems*, in Proc. 19th ACM Symposium on Theory of Computing, 1987, pp. 315–324.

[JS] R. E. JOHNSON AND F. B. SCHNEIDER, *Symmetry and similarity in distributed systems*, in Proc. 4th ACM Symposium on Principles of Distributed Computing, 1985, pp. 13–22.

[K] H. KARLOFF, Private communication.

[KSS] D. J. KLEITMAN, J. SHEARER AND D. STURTEVANT, *Intersections of k-element sets*, Combinatorica, 1 (1981), pp. 381–384.

[KW] R. M. KARP AND A. WIGDERSON, *A fast parallel algorithm for the maximal independent set problem*, in Proc. 16th ACM Symposium on Theory of Computing, 1984, pp. 266–272.

[L] M. LUBY, *A simple parallel algorithm for the maximal independent set problem*, in Proc. 17th ACM Symposium on Theory of Computing, 1985, pp. 1–10.

[LPS] A. LUBOTSKY, R. PHILLIPS AND P. SARNAK, *Ramanujan graphs*, Combinatorica, 8 (1988), pp. 261–278.