JOURNAL OF ALGORITHMS 7, 567–583 (1986)

A Fast and Simple Randomized Parallel Algorithm for the Maximal Independent Set Problem

NOGA ALON*

Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, and Department of Mathematics, Tel Aviv University, Tel Aviv, Israel

LÁSZLÓ BABAI

Department of Computer Science, University of Chicago, Chicago, Illinois 60637, and Department of Algebra, Eötvös University, Budapest, Hungary

AND

ALON ITAI

Department of Computer Science, University of Chicago, Chicago, Illinois 60637, and Department of Computer Science, Technion, Haifa, Israel

Received May 30, 1985

A simple parallel randomized algorithm to find a maximal independent set in a graph $G = (V, E)$ on n vertices is presented. Its expected running time on a concurrent-read concurrent-write PRAM with $O(|E|d_{\max})$ processors is $O(\log n)$, where $d_{\max}$ denotes the maximum degree. On an exclusive-read exclusive-write PRAM with $O(|E|)$ processors the algorithm runs in $O(\log^2 n)$. Previously, an $O(\log^4 n)$ deterministic algorithm was given by Karp and Wigderson for the EREW-PRAM model. This was recently (independently of our work) improved to $O(\log^2 n)$ by M. Luby. In both cases randomized algorithms depending on pairwise independent choices were turned into deterministic algorithms. We comment on how randomized combinatorial algorithms whose analysis only depends on d-wise rather than fully independent random choices (for some constant d) can be converted into deterministic algorithms. We apply a technique due to A. Joffe (1974) and obtain deterministic construction in fast parallel time of various combinatorial objects whose existence follows from probabilistic arguments.

*Research supported in part by the Weizmann Fellowship for Scientific Research and by NSF Grant 8406100.

0196-6774/86 $3.00

1. Introduction

An *independent set* in a graph is a set of vertices, no two of which are adjacent. A *maximal independent set* is an independent set that is not properly contained in any other independent set. Answering a problem raised by Valiant in [Va] (see also Cook [Co]), Karp and Wigderson [KW] described a fast parallel algorithm which accepts as input a graph G with n vertices and $|E|$ edges and produces a maximal independent set of vertices. On an EREW–PRAM (PRAM without concurrent write or read) their algorithm executes in time $O((\log n)^4)$ and requires $O((n/\log n)^3)$ processors.

Here we describe a simple randomized (Las Vegas) algorithm for the above problem. Its expected running time on a CRCW–PRAM with $O(|E|d_{\max})$ processors is $O(\log n)$ where $d_{\max}$ is in the maximum degree in the graph. Our algorithm can also be implemented on an EREW–PRAM with $|E|$ processors and expected running time $O(\log^2 n)$. This latter result was independently obtained by M. Luby who subsequently showed that his algorithm can be made deterministic without loss of EREW–PRAM time [L]. We are unable to make our faster CRCW–PRAM algorithm deterministic and leave this as an open problem.

Applying the algorithm to the line graph of a graph G, one obtains a similarly efficient parallel algorithm for finding a maximal matching in G. An equally time-efficient algorithm for this problem appears in [II] which requires only $O(|E|)$ processors.

Another immediate consequence of any maximal independent set algorithm is a similarly efficient algorithm for coloring a graph with $d_{\max} + 1$ colors. The following well-known trick, mentioned in [L] (cf. [Lov, Exercise 9.6]), provides a reduction. It is easy to see that $d_{\max} + 1$-colorings of the graph G are in one-to-one correspondence with the maximal independent sets of the Cartesian product of G by the complete graph on $d_{\max} + 1$ vertices. (This graph consists of $d_{\max} + 1$ copies of G with edges between all pairs of corresponding points added.) This observation applies to our $O(\log n)$ Las Vegas CRCW algorithm as well as to Luby's $O(\log^2 n)$ deterministic EREW algorithm. We note that Luby [L] gave a separate, similarly time-efficient deterministic EREW algorithm for this coloring problem, using fewer processors.

The paper is organized as follows. After introducing some notation in Section 2 we describe the algorithm in Section 3. In Section 4 we prove the combinatorial lemmas that guarantee that the expected number of "phases" of our algorithm is $O(\log n)$. This leads to a trivial implementation of our algorithm on an EREW–PRAM with expected running time $O(\log^2 n)$. In Section 5 we show how to implement our algorithm on a CRCW–PRAM with expected running time $O(\log n)$. Motivated by the results of [KW] and

[L], in Section 6 we describe a general technique, due to A. Joffe [Jo], to convert any Monte Carlo algorithm that uses d-wise independent random choices into a deterministic parallel algorithm without loss of time and a polynomial increase in the number of processors for any constant d. We mention several combinatorial applications in Section 7.

2. Notation

For every graph H, $V(H)$ is the set of vertices of H and $E(H)$ is the set of its edges. Let $G = (V, E)$ be a graph. For $X \subseteq V$, $N(X)$ is the set of all neighbours of vertices in N, i.e., $N(X) = \{u \in V: uv \in E \text{ for some } v \in X\}$. Thus $X \subseteq V$ is a maximal independent set of G if $X \cap N(X) = \emptyset$ and $X \cup N(X) = V$. If $K \subseteq V$ the *induced subgraph* of G on K, denoted simply by K, is a subgraph on the set of vertices K with edge set $E(K) = \{uw | u, w \in K, uw \in E\}$. For $v \in K$, $d_K(v)$ is the degree of the vertex v in the subgraph K.

3. The Algorithm

As in [KW], our algorithm consists of phases. In each phase an independent set S of an induced subgraph H of G is added to the current independent set and $S \cup N(S)$ is deleted from H. Starting with $H = G$ and ending when $H = \emptyset$ we get a maximal independent set. As H is by definition an *induced* subgraph of G, it is uniquely determined by its vertex set $V(H)$.

The following Procedure *IN* describes our algorithm for constructing a maximal independent set of the graph $G = (V, E)$. It refers to another procedure, $IN(H)$, to be described subsequently.

```
Procedure IN
begin
I ← ∅, V(H) ← V.
    while V(H) ≠ ∅ do begin
        S ← IN(H)
        V(H) ← V(H) − (S ∪ N_H(S))
        I ← I ∪ S
    end
end
```

Each execution of the body of the **while** loop is a *phase* of the algorithm. The following procedure selects an independent set S of H and is used in each phase of *IN*.

Procedure $IN(H)$.
begin
  for each vertex v of H **do in parallel**
    if $d_H(v) = 0$ **then** mark v.
      else mark v with probability $1/d_H(v)$.
  for each edge uv of H **do** in parallel
    if both u and v are marked erase randomly exactly one of the marks:
    that of u with probability $d_H(v)/(d_H(u) + d_H(v))$
    else that of v.
  $S \leftarrow$ set of vertices that remain marked.
end

All the random choices in IN are independent. One can easily check that Procedure $IN(H)$ always produces an independent set S of H and hence the algorithm eventually finds a maximal independent set of G. In the next section we show that the expected number of edges deleted from H in each phase is $\Omega(|E(H)|)$. Thus the expected number of phases is $O(\log n)$.

4. The Expected Number of Phases

In this section we show that the expected number of edges deleted from H in each phase of the algorithm is $\Omega(|E(H)|)$. Put $H = (V, E)$. For $v \in V$ let us abbreviate $d_H(v)$ by $d(v)$. A vertex $v \in V$ is *bad* if the degree of at least $\frac{2}{3}$ of its neighbors is greater than its own degree. A vertex is *good* if it is not bad. An edge is *bad* if both its endpoints are bad. Otherwise it is *good*. We will show that a constant fraction of the edges of H are good, and that the probability that a good edge is deleted during a phase is at least some positive constant. These two facts imply that the expected number of deleted edges is $\Omega(|E|)$, as desired. The following easy lemma was essentially proved in [II].

Lemma 4.1. *Suppose $u \in V$ is a good vertex of degree $d > 0$. Then the probability that some neighbor of u receives a mark during the procedure $IN(H)$ is $\geq 1 - e^{-1/3}$.*

Proof. By definition u has $k > d/3$ neighbors $u_1, \ldots, u_k$ with $d_j = d(u_j) \leq d$ $(j = 1, \ldots, k)$. Prob(some neighbor of u is marked) $\geq 1 -$ Prob(no u_i is marked) $= 1 - \prod_{j=1}^{k}(1 - 1/d_j) \geq 1 - (1 - 1/d)^k \geq 1 - e^{-1/3}$. □

Lemma 4.2. *Suppose $u \in V$ has degree $d > 0$. Then, in the execution of $IN(H)$, Prob($u \in S | u$ was marked) $\geq 1/e$, i.e., if u received a mark during $IN(H)$ then with probability $\geq 1/e$ it remains marked at the end of $IN(H)$.*

Proof. Let $u_1, u_2, \ldots, u_d$ be the neighbors of u, with $d_j = d(u_j)$. For $1 \geq j \geq d$ let A_j denote the event that u_j was marked and that when the edge uu_j was considered the mark of u was erased. Clearly, these d events are independent. Hence

$$\begin{aligned}\text{Prob}(u \in S | u \text{ was marked}) &= \text{Prob}\left(\text{no } A_j \text{ occurs} | u \text{ was marked}\right) \\ &= \prod_{j=i}^{d} \left(1 - \text{Prob}(A_j)\right) \\ &= \prod_{j=1}^{d} \left(1 - \frac{1}{d_j} \cdot \frac{d_j}{d + d_j}\right) \\ &\geq \left(1 - \frac{1}{d+1}\right)^d \geq 1/e. \quad \square\end{aligned}$$

COROLLARY 4.3. *Suppose $u \in V$ is a good vertex of degree $d > 0$. Then its probability to be deleted in this phase is $\geq (1 - e^{-1/3})/e$.*

Proof. Let $u_1, \ldots, u_d$ be the neighbors of u. Then

$$\begin{aligned}&\text{Prob}(u \text{ is deleted}) \\ &\quad \geq \text{Prob}(u_j \in S \text{ for some } j) \\ &\quad = \sum_{j=1}^{d} \text{Prob}(u_j \text{ was marked and } u_1, \ldots, u_{j-1} \text{ were not}) \\ &\qquad \times \text{Prob}(u_j \in S | u_j \text{ was marked and } u_1, \ldots, u_{j-1} \text{ were not}) \\ &\quad \geq \sum_{j=1}^{d} \text{Prob}(u_j \text{ was marked and } u_1, \ldots, u_{j-1} \text{ were not}) \\ &\qquad \times \text{Prob}(u_j \in S | u_j \text{ was marked}) \\ &\quad \geq \frac{1}{e} \sum_{j=1}^{d} \text{Prob}(u_j \text{ was marked and } u_1, \ldots, u_{j-1} \text{ were not}) \\ &\quad = \frac{1}{e} \text{Prob}(u_j \text{ was marked for some } j) \geq \frac{1}{e} \cdot (1 - e^{-1/3})\end{aligned}$$

as needed.

The last two inequalities follow from Lemmas 4.1 and 4.2. □

The authors of [II] proved that any graph with $|E|$ edges has at least $\frac{1}{3}|E|$ good edges. The following is a somewhat simpler proof of a slightly better result.

LEMMA 4.4. *Let $H = (V, E)$ be a graph. Then the number of good edges of H is $\geq \frac{1}{2}|E|$.*

Proof. Direct each edge of H from the smaller degree endpoint to the higher degree endpoint (arbitrarily if these degrees equal). For $A, C \subseteq V$ let $E(A, C)$ be the set of all edges from a vertex of A to a vertex of C. For $v \in V$ let $d^+(v)$ and $d^-(v)$ be the number of edges emanating from and entering, respectively, v. Let $B \subseteq V$ be the set of all bad vertices and let $G = V - B$ be the set of all good ones. Note that for $v \in B$ $d^+(v) \geq 2d^-(v)$. One can easily check that

$$\begin{aligned}
&2|E(B,B)| + |E(B,G)| + |E(G,B)| \\
&\quad = \sum_{v \in B} (d^+(v) + d^-(v)) \\
&\quad \leq 3 \sum_{v \in B} (d^+(v) - d^-(v)) \\
&\quad = 3 \sum_{v \in G} (d^-(v) - d^+(v)) \\
&\quad = 3(|E(B,G)| - |E(G,B)|) \\
&\quad \leq 3(|E(B,G)| + |E(G,B)|).
\end{aligned}$$

Thus $|E(B, B)| \leq |E(B, G)| + |E(G, B)|$, i.e., at most half of the edges of H are bad, as claimed. □

COROLLARY 4.5. *The expected number of deleted edges in a phase is $\geq (1/2e)(1 - e^{-1/3})|E(H)|$. Hence the expected number of phases of the algorithm is $O(\log|E(G)|)$.*

Proof. In each phase the number of good edges is $\geq \frac{1}{2}|E(H)|$, by Lemma 4.4. Each such edge is incident with a good vertex, which is deleted with probability $\geq 1/e(1 - e^{-1/3})$. Hence the expected number of deleted edges is $\geq 1/2e(1 - e^{-1/3})|E(H)|$. Since no more than $|E(H)|$ edges can be deleted this shows that, say, at least $\frac{1}{100}|E(H)|$edges are deleted with probability $\geq \frac{1}{100}$. As is easily checked this implies that the expected number of phases needed to delete all edges is $O(\log|E(G)|)$. One more phase will suffice then to add all the remaining isolated vertices of H to the independent set and complete the algorithm. □

5. IMPLEMENTATION

Recall than $n = |V(G)|$, $|E| = |E(G)|$. Is is straightforward to implement our algorithm on an EREW–PRAM with $O(|E|)$ processors and running time $O(\log n) = O(\log|E|)$ for each phase. In this section we show

how to implement the algorithm (in fact, a slightly modified version of it) on a CRCW–PRAM with running time $O(1)$ for each phase. A basic tool here is the Random Choice Operation (RCO) introduced in [II]; this is the operation of choosing randomly a nonzero entry of a nonzero boolean vector, where all nonzero entries have the same probability to the chosen, and with a small probability, no entry is chosen.

Let x be a boolean vector of length d. The RCO on x is implemented in a CRCW–PRAM with d processors $p_1, \ldots, p_d$ and constant time. Note that one concurrent write is enough to check that x is not identically zero.

RCO (see [II]):

(1) entry := nil

(2) Each processor p_i chooses at random a number r_i, $1 \le r_i \le d$.

(3) If $x_{r_i} \neq 0$ then entry := r_i.

Note that here we use the common model [Go] that in case of a write conflict one of the processors (say the lowest numbered) succeeds. However, by increasing the number of processors we can implement RCO in the more restrictive model in which a concurrent write succeeds only if all the processors try to write the same content. This can be done by finding the lowest numbered processor p_i for which $x_{r_i} \neq 0$ and putting entry := r_i. We omit the details.

The RCO *succeeds* if it chooses an r such that $x_r = 1$. It *fails* if for all i, $x_{r_i} = 0$.

We need the following easy observation.

OBSERVATION 5.1. *Let x be a boolean vector of length d with $d_1 > 0$ nonzero entries. Then the probability that the* RCO *on x succeeds is* $\ge 1 - e^{-d_1}$.

Proof. Prob(RCO on x fails) $= (1 - d_1/d)^d \le e^{-d_1}$. □

The only difficulty in implementing a phase of the algorithm in a CRCW–PRAM in constant time is the implementation of the two probability choices in the procedure *IN*. For each vertex v of degree d in the original graph G, the first choice is a random choice with probability $1/d_H(v)$, where $1 \le d_H(v) \le d$. This is done by applying RCO to the vector of edges incident with v, where zero denotes an edge that had already been removed. Once an edge is chosen we check if it is the lowest numbered edge (as we already mentioned, minimum can be easily found in constant time) and in case it is we mark v. Note that because of the failure probability, v is in fact marked with a slightly smaller probability than $1/d_H(v)$. However, by Observation 5.1

$$1/2d_H(v) \le \text{Prob}(v \text{ is marked}) \le 1/d_H(v)$$

and the events that distinct vertices are marked are independent. One can easily check that the assertion of Lemma 4.1 (with a different constant) still holds.

To implement the second random choice in IN, we apply, for each edge uv of H with both sides marked, an RCO on the concatenation of the vector of edges incident with u and the vector of edges incident with v. If our RCO chooses a nonzero entry of the first vector we erase the mark of v, otherwise (even if it fails) we erase the mark of u. Again, by Observation 5.1,

$$\text{Prob(the mark of } u \text{ is erased)} \leq \frac{d_H(v)}{d_H(u) + d_H(v)} + e^{-(d_H(u)+d_H(v))} \leq \frac{2d_H(v)}{d_H(u) + d_H(v)}$$

and certainly

$$\text{Prob(the mark of } v \text{ is erased)} \leq 2d_H(u)/(d_H(u) + d_H(v)).$$

A close look at the proof of Lemma 4.2 reveals that its assertion (with a different constant) still holds under these conditions. Hence the expected number of phases of this slightly modified algorithm is still $O(\log|E(G)|)$. Clearly, now each phase requires constant time (on a CRCW–PRAM). The greatest demand for processors occurs during the RCO's on the edges. Here each edge uv requires $d_G(u) + d_G(v)$ processors. The total number of processors is thus $O(\Sigma_{v \in V} d_G^2(v)) = O(|E(G)|d_{\max})$ where $d_{\max}$ is the maximum degree in G.

6. Converting Randomized Algorithms into Deterministic Ones

Karp and Wigderson [KW] introduced a method to convert certain randomized algorithms into deterministic ones. The method is based on the fact that the analysis of their randomized algorithm depends only on pairwise rather than fully independent random choices.

The objective of this section is to generalize the Karp–Wigderson technique to situations where d-wise independence of the random choices is required for some constant d.

The basic idea, as in [KW], is to *replace an exponentially large sample space* by one of *polynomial size*. Clearly, if a random variable over such a small sample space takes a certain value with positive probability then we can actually find such a point in the sample space deterministically without loss of time using a polynomial number of parallel processors.

It was observed long ago that in order to construct pairwise independent random variables, exponentially smaller sample spaces suffice compared to what is required for full independence. Lancaster [Lan] credits an example in Bernstein's textbook [Ber] for the basic idea and constructs $n-1$ pairwise independent random variables on a sample space of size n for every n. For further development, cf. [O'Br].

A. Joffe [Jo] generalized this result to d-wise independence. His remarkably simple construction uses finite fields. The variables he constructs are uniformly distributed over a sample space of prime power order. We shall see that Joffe's idea can be generalized so that the distributions of the variables approximate given probability distributions. We also consider the problem, how much smaller the sample space could be made. Using BCH codes, we obtain a tight bound for the case when each variable is uniformly distributed over a set of two values.

These results can be used to turn randomized algorithms whose analysis depends on d-wise independence into deterministic algorithms at a cost of essentially raising sequential time or number of parallel processors to power d.

Mostly independently of our work, Joffe's method has recently been used or rediscovered by several authors in computer science ([ACGS], [AW], [An], [Be], [CG], [CGHFRS], [L]). The basic construction remained the same. We believe it is worth stating the conclusions in the generality given below.

Luby [L] uses the method for $d = 2$, and applies it to his EREW–PRAM algorithm for the maximal independent set problem.

Our analysis of the algorithm presented in this paper requires full independence of the random choices. We note that the analysis of the EREW–PRAM implementation can be changed, along the lines of [L], into one depending on pairwise independent choices, thereby reproducing Luby's deterministic $O(\log^2 n)$ result (with a slightly different algorithm). Unfortunately, however, we are unable to turn our $O(\log n)$ CRCW–PRAM algorithm into a deterministic one.

After these comments, we turn to the description of the method.

DEFINITION 6.1. Let ξ and ξ' be two random variables. We define the *distance* between the *distributions* of ξ and ξ' to be $\operatorname{dist}(\xi, \xi') = \sup_{x \in R}(|\operatorname{Prob}(\xi < x) - \operatorname{Prob}(\xi' < x)|)$.

DEFINITION 6.2. We call a finite probability space (Ω, P) *uniform* if each elementary event (element of the sample space Ω) has the same probability $1/|\Omega|$.

We shall say that a random variable is *combinatorial* if its range consists of a finite numer of rational numbers each taken with rational probability.

The distributions of such variables can be given *explicitly* by listing the values and the corresponding probabilities.

The following result achieves the goal outlined above.

PROPOSITION 6.3. *Let $\xi_1, \ldots, \xi_n$ be random variables. Let further q be a prime power, $q \geq n$ and $d \geq 1$. Then there exists a uniform probability space (Ω, P) over a sample space of size $|\Omega| = q^d$ and random variables $\xi_1', \ldots, \xi_n'$ over (Ω, P) such that $\operatorname{dist}(\xi_i, \xi_i') \leq 1/2q$ $(i = 1, \ldots, n)$ and the ξ_i' are d-wise independent. Moreover, if the ξ_i are combinatorial in the above sense and are explicitly given then for fixed d the ξ_i' can be evaluated in logspace.*

The last sentence assumes that Ω is identified with the integers between 1 and q^d; q is part of the input and is given in unary.

We remark that in the space to be constructed all probabilities $\operatorname{Prob}(\xi_i' = x)$ will be of the form a/q where a is an integer. In particular, if $\operatorname{Prob}(\xi = x)$ is a number of this form, then $\operatorname{Prob}(\xi = x) = \operatorname{Prob}(\xi' = x)$.

Proof. Let $F_i(x) = \operatorname{Prob}(\xi_i < x)$ be the distribution function of ξ_i. Let us approximate F_i by a function G_i defined as follows.

$$G_i(x) = \frac{1}{q}\left\lceil \left(qF_i(x) - \frac{1}{2} \right) \right\rceil. \tag{6.1}$$

Clearly, G_i is again a probability distribution function and the distance between G_i and F_i is at most $1/2q$.

Let $x_{i1} < \cdots < x_{ir_i}$ be the points of discontinuity of G_i. Clearly, $r_i \leq q$. Let $a_{ij} = G(x_{ij} + 0) - G(x_{ij})$. We have

$$\sum_{j=1}^{r_i} a_{ij} = 1. \tag{6.2}$$

Let now $F = GF(q) = \{f_1, \ldots, f_q\}$ be the field of q elements. Let us associate with each ξ_i a partition $F = \bigcup_j A_{ij}$ where $|A_{ij}| = qa_{ij}$.

Our sample space Ω will be the set of q^d polynomials of degree $\leq d - 1$ over F. We define the random variables ξ_i' over the uniform space on Ω as follows.

For $p \in \Omega$, let $\xi_i'(p) = x_{ij}$ if $p(f_i) \in A_{ij}$. Clearly, $\operatorname{Prob}(\xi_i' = x_{ij}) = a_{ij}$. Hence the distribution function of ξ_i' is G_i and consequently $\operatorname{dist}(\xi_i, \xi_i') \leq 1/2q$. The fact that the ξ_i' so constructed are d-wise independent follows immediately from the first result of interpolation theory: a polynomial $p \in \Omega$ is uniquely determined by fixing its value at d different places.

The claim of efficient evaluation of the ξ_i' is clearly justified. □

Naturally, the question arises whether or not we could compress the sample space even further. The following simple observation shows that under quite general circumstances, no such compression is possible.

If d independent random variables ξ_i are to take at least q different values each with positive probability then for suitable functions f_i, the random variables $\eta_i = f_i(\xi_i)$ take at least q different prime numbers as values and disjoint sets of primes for distinct variables. Therefore their product takes at least q^d values with positive probability. This is not possible over a sample space of size less than q^d.

This observation still leaves the possibility of considerable improvement when the variables take only a small number of distinct values, less than n say. In this case in Proposition 6.3 we may choose q to be less than $2n$ and our bound is $|\Omega| = O(n^d)$. The following proposition shows that this result is not far from best possible; even if the variables take only two values, at least about the square root of this number is necessary. A variable is *almost constant* if it takes a single value with probability 1.

Let $m(n, d)$ denote the following sum of binomial coefficients:

$$m(n,d) = \begin{cases} \displaystyle\sum_{j=0}^{d/2} \binom{n}{j} & \text{if } d \text{ is even} \\ \displaystyle\sum_{j=0}^{(d-1)/2} \binom{n}{j} + \binom{n-1}{(d-1)/2} & \text{if } d \text{ is odd.} \end{cases}$$

PROPOSITION 6.4. *Assume that the random variables $\xi_1, \ldots, \xi_n$ over the sample space Ω are d-wise independent and not almost constant. Then $|\Omega| \geq m(n, d)$.*

A similar result was found independently by [CGHFRS]. Their "Uniform Projection Lemma" is equivalent to the special case of Proposition 6.4 when we restrict the probability space to be uniform and the ξ_i to be 0 or 1 with probability $\frac{1}{2}$ each.

Proof. We may assume that each variable has zero expected value. (Other wise we replace ξ_i by $\xi_i - E(\xi_i)$.) For any subset S of $\{1, \ldots, n\}$ let $\alpha_S = \Pi_{i \in S} \xi_i$. By the d-wise independence of the ξ_i.

$$E(\alpha_S \alpha_T) = \begin{cases} \textit{positive} & \text{if } S = T \text{ and } |S| \leq d; \\ 0 & \text{if } S \neq T \text{ and } |S \cup T| \leq d. \end{cases}$$

Let now $S_1, \ldots, S_m$ be subsets of $\{1, \ldots, n\}$ such that $|S_i \cup S_j| \leq d$ for every i, j, where $m = m(n, d)$. (Take all sets of size $\leq d/2$ and, if d is odd, add those $(d+1)/2$-sets containing element 1.)

We claim that the functions α_{S_i} $(i = 1, \ldots, m)$ are linearly independent (in the function space $\mathbf{R}^{\Omega}$). This implies the inequality $|\Omega| \geq m$ stated in the proposition.

Let $\alpha_1 = \alpha_{S_i}$. Suppose some linear combination $\varphi = \sum_{i=1}^{m} a_i \alpha_i$ is zero. Then for any j,

$$0 = E(\varphi\alpha_j) = \sum_{i=1}^{m} a_i E(\alpha_i\alpha_j) = a_j E\left(\alpha_j^2\right).$$

Consequently all coefficients a_j are zero. □

For fixed d, Proposition 6.4 shows that any probability space with n d-wise independent (not almost constant) random variables has size $\Omega(n^{\lfloor d/2 \rfloor})$. Using the binary BCH code we show that this is best possible in the case when each ξ_i takes only two values, with probability $\frac{1}{2}$ each.

PROPOSITION 6.5. *Suppose $n = 2^k - 1$ and $d = 2t + 1 \leq n$. Then there exists a uniform probability space Ω of size $2(n+1)^t$ and d-wise independent random variables $\xi_1, \ldots, \xi_n$ over Ω each of which takes the values 0 and 1 with probability $\frac{1}{2}$.*

The space and the variables are explicitly constructed, given a representation of the field $F = GF(n+1)$ as a k-dimensional algebra over $GF(2)$.

Proof. Let $x_1, \ldots, x_n$ be the n nonzero elements of F, represented as column-vectors of length k over $GF(2)$. Consider the following $1 + kt$ by n matrix over $GF(2)$,

$$H = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ x_1 & x_2 & x_3 & & x_n \\ x_1^3 & x_2^3 & x_3^3 & & x_n^3 \\ \vdots & & & \vdots & \vdots \\ x_1^{2t-1} & x_2^{2t-1} & x_3^{2t-1} & \ldots & x_n^{2t-1} \end{pmatrix}.$$

This is the parity check matrix of the (primitive, narrow-sense) binary BCH code [MS, Chap. 7.6] of length n and designed distance $2t + 2$, augmented with a parity check bit. It is well known, that any $d = 2t + 1$ columns of H are linearly independent [MS, Chap. 7.6, Theorem 8].

Let $\Omega = \{1, 2, \ldots, 2(n+1)^t\}$ and let $A = (a_{ij})$ $(i \in \Omega, 1 \leq j \leq n)$ be the $(0,1)$-matrix whose $2(n+1)^t = 2^{kt+1}$ rows are all linear combinations of the rows of H. We endow the sample space Ω with uniform probability measure. The random variable ξ_j over Ω will be defined by the formula $\xi_j(i) = a_{ij}$ $(i \in \Omega, 1 \leq j \leq n)$.

To prove that the ξ_j are d-wise independent, we have to show that for every J of up to d columns, the rows of the $|\Omega|$ by $|J|$ matrix $A_J = (a_{ij})_{j \in J}$ take on each of the $2^{|J|}$ $(0,1)$-vectors equally often. This follows immediately from the fact that the columns of the corresponding submatrix

H_J of H are linearly independent and therefore its rows span the space $GF(2)^{|J|}$. □

We remark that the matrix A constructed in the proof is an *orthogonal array* of strength d [MS, Chap. 5.5, Theorem 8] and any other orthogonal array of strength d can be used to construct d-wise independent random variables in a similar fashion.

7. Deterministic Construction of Combinatorial Objects

Following [KW] we say that a relation R lies in $\widetilde{NC}$ if there exists a deterministic PRAM algorithm that produces for any input x of size n an output y with $(x, y) \in R$ in time $(\log n)^{O(1)}$ using $n^{O(1)}$ processors.

In combinatorics, the existence of certain objects is often established by proving that random choice leads to the desired object with positive probability. If this probability is not negligible then such procedures usually yield random polynomial time algorithms to construct the desired object, and often they allow parallelism to yield an $\widetilde{RNC}$ search algorithm.

Using Proposition 6.3 one can show that many of these combinatorial search problems actually belong to $\widetilde{NC}$. Here are four examples. We note that even the weaker corollary that these problems thus belong to P is of interest since the probabilistic existence proofs usually do not yield this. An outstanding open problem is to construct, in polynomial (of n) time, graphs with n vertices and with clique and anticlique size $O(\log n)$. (An anticlique is an independent set.) An old result of Erdős says that almost all graphs satisfy this condition [E]. This simple result turned out to be quite a triumph of the then new probabilistic method. Ingenious explicit constructions [F, FW] yield clique and anticlique sizes of the order of $\exp c\sqrt{\log n}$.

A. Independent Sets in Sparse Hypergraphs

A *hypergraph* $\mathcal{H} = (V, \mathcal{E})$ is a system $\mathcal{E}$ of subsets of V called *edges*. $\mathcal{H}$ is d-uniform if every edge has d elements. A set $W \subseteq V$ is independent if W contains no edge.

Proposition 7.1. *Let $\mathcal{H} = (V, \mathcal{E})$ be a d-uniform hypergraph on $n = |V|$ vertices ($d \geq 2$, fixed). Let $k = c(n^d/|\mathcal{E}|)^{1/(d-1)}$ for a suitable constant c. Then finding an independent set W of size $\geq k$ is in $\widetilde{NC}$.*

Proof. First we show that for $k \geq 7$ and $c \leq \frac{1}{18}$ such a set W always exists. Let us construct a random subset R of V by picking each element of V with probability $(3k)/n$. Let ζ_i denote the indicator variable of the event $i \in R$. The size of R is $\rho = |R| = \sum_{i=1}^{n} \zeta_i$. Clearly, $E(\rho) = 3k$, and,

assuming the ζ_i are *pairwise* independent, $D^2(\rho) = \sum_{i=1}^{n} D^2(\zeta_i) < 3k$. By Chebyshev's inequality we find that

$$\text{Prob}(|R| \leq 2k) < 3/k. \tag{7.1}$$

For each edge $E_j \in \mathscr{E}$, let η_j be the indicator of the event $E_j \subseteq R$. The probability of this event is

$$E(\eta_j) = \left(\frac{3k}{n}\right)^d \tag{7.2}$$

assuming d-wise independence of our choices. Hence the expected number of those E_j contained in R is

$$E\left(\sum \eta_j\right) = |\mathscr{E}|\left(\frac{3k}{n}\right)^d < \frac{k}{2}. \tag{7.3}$$

(The last inequality will hold if $c < 1/18$.) Therefore

$$\text{Prob}\left(\sum \eta_i \geq k\right) < \tfrac{1}{2}. \tag{7.4}$$

Consequently, the event $C = \{\sum \eta_j < k \text{ and } |R| \geq 2k\}$ has probability $> \frac{1}{2} - 3/k$. But if C holds then removing one point of each E_j contained in R from R results in an independent set W such that $|W| > k$.

This completes the proof of existence and provides an $\widetilde{RNC}$ algorithm. To make the algorithm deterministic, let $n/k < q \leq 2n/k$ and select every point with equal probability between $3k/n$ and $4k/n$. This way, (7.1) will continue to hold, and (7.3) will hold as well if our choices were d-wise independent and $c \leq \frac{1}{32}$. By Proposition 6.3, such choices can be accomplished over a uniform probability space of size q^d. Using q^d n-tuples of processors we can try all these choices at once, and event C will hold for at least one n-tuple of processors. □

B. Large d-Partite Subhypergraph

PROPOSITION 7.2. *Fix $d \geq 2$ and let $\mathscr{H} = (V, \mathscr{E})$ be a d-uniform hypergraph. Then the following problem is in $\widetilde{NC}$. Find a partition $(V_1, \ldots, V_d)$ of V such that the number of edges of $\mathscr{H}$ having precisely one vertex in each class V_i is at least $\lfloor |\mathscr{E}| d!/d^d \rfloor$.*

Proof. Let ξ_i denote a random number from the set $\{1, \ldots, d\}$. Let us color vertex i by "color" ξ_i. Then, assuming the ξ_i to be d-wise independent, the probability that a given edge is "good" (has all the colors) is $d!/d^d$. The expected number of good edges is $|\mathscr{E}| d!/d^d$ and therefore a coloring satisfying the condition exists.

Using Proposition 6.3 with $q > 2d^2|\mathscr{E}|$ we are able to construct d-wise independent random variables ξ_i' such that the probability that a given edge is good as greater than

$$d!\left(\frac{1}{d} - \frac{1}{2d^2|\mathscr{E}|}\right)^d > \frac{d!}{d^d}\left(1 - \frac{1}{2|\mathscr{E}|}\right).$$

Therefore the expected number of good edges is greater than $|\mathscr{E}|d!/d^d - 1/2$. Hence, with positive probability, it is at least $\lfloor |\mathscr{E}|d!/d^d \rfloor$. □

We remark that Proposition 7.2 remains valid if we omit $\lfloor\ ,\ \rfloor$ but we have to use $q > 2d^{d+1}|\mathscr{E}|$.

C. *Ramsey-Type Problems*

PROPOSITION 7.3. *The following problem is in $\widetilde{NC}$. For fixed $k \geq 2$ color the edges of the complete graph on n vertices by l colors such that no more than $1 + \binom{n}{k} l^{-\binom{k}{2}+1}$ complete k-subgraphs be monochromatic. (The input numbers n and l are written in unary.)*

Proof. Similar to 7.2, with $d = \binom{k}{2}$. We omit the details. □

D. *Sidon-Subsets of Groups*

Let G be a finite abelian group of odd order. A subset $S \subseteq G$ is a *Sidon subset* if all the pairwise sums $x + y$ $(x, y \in S)$ are different.

PROPOSITION 7.4. *The following problem is in $\widetilde{NC}$. Let G be a finite abelian group given by its Cayley table. Let A be a subset of G, $|A| = n$. Find a Sidon subset $S \subseteq A$ of size $|S| \geq cn^{1/3}$.*

Proof. This essentially follows from Proposition 7.1. The edges of the hypergraph to consider are the quadruples $\{x_1, \ldots, x_4\}$ where $x_i \in A$, $x_1 + x_2 = x_3 + x_4$, $x_i \neq x_j$ and the triples $\{x_1, x_2, x_3\}$ where $x_i \in A$, $x_1 + x_2 = 2x_3$, $x_i \neq x_j$. A Sidon set is precisely an independent set in this hypergraph. We have to choose $d = 4$. □

We remark that some care has to be taken when defining Sidon sets for groups of even order (one has to exclude sums of the form $x + x = 0$) and to nonabelian groups (there are two natural generalizations, cf. [BS]). However, the $\widetilde{NC}$ algorithm outlined above will in all cases produce Sidon sets of size $cn^{1/3}$.

If $A = G$, clearly no Sidon subset of A will be greater than $(2n)^{1/2}$. For certain subsets, one can guarantee the existence of substantially larger

Sidon sets, and in fact one can find such sets quickly in parallel. By a reasoning similar to the one above, one can modify arguments from [AE] to prove the following.

PROPOSITION 7.5. *The following problem is in* $\widetilde{NC}$. *Let* G *be a finite abelian group given by its Cayley table. Let* A *be a subset of* G, $|A| = n$. *Suppose that each* $g \in G$ *can be written as a sum of two elements of* A *in at most* k *distinct ways* (k *constant*). *Find a Sidon subset* $S \subseteq A$ *of size* $|S| \geq c_k n^{2/3}$.

REFERENCES

[AW] M. AJTAI AND A. WIGDERSON, "Deterministic Simulation of Probabilistic Constant Depth Circuits," Proc. 26th FOCS, pp. 11–19, Portland Or., 1985.

[ACGS] W. ALEXI, B. CHOR, O. GOLDREICH, AND C. P. SCHNORR, RSA/Rabin bits are $\frac{1}{2} + \frac{1}{\text{poly}\log(n)}$ secure, Proc. 25th STOC, pp. 449–457, Singer Island, FL 1984.

[AE] N. ALON AND P. ERDŐS, An application of graph theory to additive number theory, *European J. Combin.*, in press.

[An] R. ANDERSON, Set splitting, preprint, MIT, Cambridge, Mass., 1985.

[BS] L. BABAI AND VERA T. SÓS, Sidon sets in groups and induced subgraphs of Cayley graphs, *European J. Combin.*, **6** (1985), 101–114.

[Be] P. BEAME, Private communication by M. Luby.

[Ber] S. BERNSTEIN, "Theory of Probability" (3rd ed.), GTTI, Moscow 1945.

[CG] B. CHOR AND O. GOLDREICH, On the power of two points based sampling, preprint, MIT, Cambridge, Mass., 1985.

[CGHFRS] B. CHOR, O. GOLDREICH, J. HASTAD, J. FRIEDMAN, S. RUDICH, AND R. SMOLENSKY, "The Bit Extraction Problem or t-Resilient Functions," Proc. 26th FOCS, pp. 396–407, Portland Or., 1985.

[Co] S. A. COOK, An overview of computational complexity, *Comm. ACM* **26** (1983), 400–408.

[E] P. ERDŐS, Some remarks on the theory of graphs, *Bull. Amer. Math. Soc.* **53** (1947), 292–294.

[E] P. ERDŐS, "The Art of Counting," Selected Writings, (J. Spencer, Ed.) MIT Press, Cambridge, Mass., 1973.

[F] P. FRANKL, A constructive lower bound for some Ramsey numbers, *Ars Combinatoria* **3** (1977), 371–373.

[FW] P. FRANKL AND R. M. WILSON, Intersection theorems with geometric consequences, *Combinatorica* **1** (1981), 357–368.

[Go] L. M. GOLDSCHLAGER, "Synchronous Parallel Computation," Ph.D. thesis, Univ. of Toronto, 1977: Proc. 10th ACM STOC, 89–94, 1978, *J. Assoc. Comput. Mach.* **29** (1982), 1073–1086.

[II] A. ISRAELI AND A. ITAI, A fast and simple randomized parallel algorithm for maximal matching, preprint, 1984.

[Jo] A. JOFFE, On a set of almost deterministic k-independent random variables, *Ann. Probability* **2** (1974), 161–162.

[KW] R. M. KARP AND A. WIGDERSON, "A Fast Parallel Algorithm for the Maximal Independent Set Problem," Proc. 16th ACM STOC, pp. 266–272, Washington

D.C. 1984.

[Lan] H. O. LANCASTER, Pairwise statistical independence, *Ann. Math. Stat.* **36** (1965), 1313–1317.

[LOV] L. LOVÁSZ, "Combinatorial Problems and Exercises," Akadémiai Kiadó, Budapest and North-Holland, Amsterdam, 1979.

[L] M. LUBY, "A Simple Parallel Algorithm for the Maximal Independent Set Problem," Proc. 17th ACM STOC, pp. 1–10, Providence, R.I., 1985.

[MS] F. J. MACWILLIAMS AND N. J. A. SLOANE, "The Theory of Error Correcting Codes," North-Holland, Amsterdam, 1977.

[O'Br] G. L. O'BRIEN, Pairwise independent random variables, *Ann. Probability* **8** (1980), 170–175.

[Va] L. G. VALIANT, "Parallel Computation," Proc. 7th IBM Symposium on Math. Foundations of Computer Science, pp. 173–189, Japan, 1982.