Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

115

Automata, Languages and Programming

Eighth Colloquium,
Acre (Akko), July 1981

Edited by S. Even and O. Kariv

Springer-Verlag
Berlin Heidelberg New York

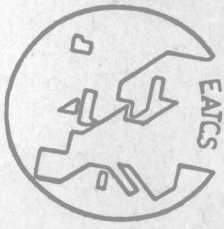
Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

115

Automata, Languages and Programming

Eighth Colloquium
Acre (Akko), Israel
July 13–17, 1981

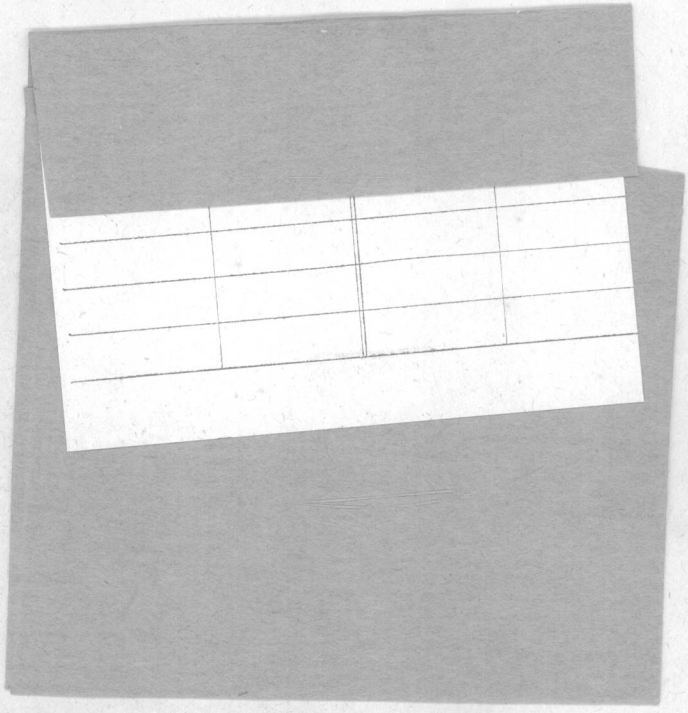


Edited by S. Even and O. Kariv

Springer-Verlag
Berlin Heidelberg New York 1981

Editors

Shimon Even
Oded Kariv
Dept. of Computer Science
Technion-Israel Institute of Technology
32000 Haifa, Israel



AMS Subject Classifications (1981): 68-XX
CR Subject Classifications (1974): 4.1, 4.2, 5.2, 5.3

ISBN 3-540-10843-2 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-10843-2 Springer-Verlag New York Heidelberg Berlin

Printed in Germany

Printing and binding: Beltz Offsetdruck, Hemsbach/Bergstr.
2145/3140-543210

PREFACE

ICALP is the acronym of the annual International Colloquium on Automata, Languages and Programming sponsored by the European Association for Theoretical Computer Science (EATCS). It is a broad-based conference covering all aspects of the foundations of computer science, including such topics as automata, formal languages, analysis of algorithms, computational complexity, computability, mathematical aspects of programming language definition, flow analysis, semantics of programming languages, parsing, program verification, dynamic logic, rewriting systems, cryptology, abstract data types, data structures and data base theory. Previous ICALP conferences were held in Paris(1972), Saarbrücken(1974), Edinburgh (1976), Turku(1977), Udine(1978), Graz(1979) and Noordwijkerhout(1980).

ICALP 81 is the 8th conference of EATCS, covering once again a broad spectrum of theoretic computer science. It is organized by the Computer Science Dept. of the Technion and is to be held on July 13-17,1981, in Acre (Akko), Israel.

There are 44 papers in this volume, including 2 invited papers (by J.D. Ullman and E. Engeler). The other contributed papers were selected by a Selection Committee (the names of its members are listed below) from 109 extended abstracts and draft papers submitted in response to the call for papers. Each submitted paper was sent for evaluation to three members of the Program Committee. The manuscripts, however, were not formally refereed as several of them represent preliminary reports of continuing research. It is anticipated that most of these documents will appear in more polished and complete form in scientific journals.

The chairman of ICALP 81 and the organizing committee wish to thank all those who submitted papers for consideration; members of the Program Committee (see below) for their help in the evaluation of the papers, and the many who assisted in this process (see next page); Maurice Nivat who was a host of the Selection Committee meeting in Paris; all the institutions and corporations which support ICALP 81 (see list of supporters below); Mr. Henry Lochoff of "Eden-Tours" Ltd. for handling the many touristic details related to the conference; Springer-Verlag for printing this volume. Finally we wish to thank Ms. Anat Even and Ms. Bella Gologorsky and the secretarial staff of the Computer Science Department of the Technion for their assistance in all organizational matters related to the conference.

April 16, 1981

S. Even and O. Kariv

Referees for ICALP 81

Albert D.
Apt K.R.

deBakker J.W.
Bancilhon F.
Baudet G.M.
Bečvář J.
Beeri C.
Ben-Ari M.
Bergstra J.A.
Berry G.
Beynon W.M.
Bird M.
Blikle A.
Boehm C.
deBruin A.
Bucher W.

Cohn A.G.
Coppo M.
Courcelle B.
Cousineau G.
Cremers A.B.

Dembinski P.
Dezani M.

Edelsbrunner H.
Ehrlich G.
van Emde Boas P.
Engelfriet J.
Even S.

Flajolet P.
Fraenkel A.S.
Francez N.

Galil Z.
Gaudep M.C.
Goeman H.J.M.
Gordon M.J.C.
Grabowski M.
deGroote H.F.

Harel D.
Harrison M.A.
Havel I.M.
Hennessy M.
Hofri M.

Itai A.

Janko W.
Jantzen M.
Jeanrond H.J.
Jerrum M.
Jones C.

Kantorowitz E.
Karhumäki J.
Kariv O.
Karp R.M.
Katz S.
Kleijn H.C.M.
Klop J.W.
Kramosil I.
Kreczmar A.

Lehmann D.
Lempel A.
Levy J.J.
Linna M.
Lipski W. Jr.
Lozinskii E.L.

Makowsky J.A.
Manna Z.
Maurer H.
Mayoh B.H.
McColl W.F.
Meseguer J.
Meyer J.-J.Ch.
Milner R.
Mirkowska G.
Monien B.
Monier L.M.
Muchnick S.S.
Müller K.

Nivat M.

Orlowska E.
Ottmann Th.

Park D.
Paterson M.
Paz A.
Penttonen M.
Perl Y.
Perrin D.
Pettorossi A.
Pittl J.
Plotkin G.D.
Pnueli A.
Pratt V.

Raz Y.
Rodeh M.
Roucairol G.
Rozenberg G.

Salomaa A.
Salwicki A.
Schnorr C.P.
Schreiber S.
Schwartz R.L.
Shamir A.
Shamir E.
Shields M.W.
Shiloach Y.
Shostak R.
Sippu S.
Six H.W.
Skyum S.
Soisalon-Soininen E.
Steinby M.

Thompson C.D.
Tiuryn J.
Tucker J.V.

Ukkonen E.
Ullman J.D.

Vuillemin J.E.

Wadge W.
Wegner L.
Wolper P.

Yacobi Y.
Yehudai A.
Yoeli M.

8th International Colloquium on Automata

Languages & Programming

ICALP 81

July 13-17, 1981
Acre (Akko), Israel

TABLE OF CONTENTS

REFINED ANALYSIS AND IMPROVEMENTS ON SOME FACTORING ALGORITHMS

C.P. Schnorr
Fachbereich Mathematik
Universität Frankfurt*

Extended Abstract

Abstract. By combining the principles of known factoring algorithms we obtain some improved algorithms which by heuristic arguments all have a time bound $O(\exp \sqrt{c \ln n \ln \ln n})$ for various constants $c \geq 3$. In particular, Miller's method of solving index equations and Shanks's method of computing ambiguous quadratic forms with determinant $-n$ can be modified in this way. We show how to speed up the factorization of n by using preprocessed lists of those numbers in $[-u,u]$ and $[n-u,n+u]$, $0 << u << n$ which only have small prime factors. These lists can be uniformly used for the factorization of all numbers in $[n-u,n+u]$. Given these lists, factorization takes $O(\exp[2(\ln n)^{1/3}(\ln \ln n)^{2/3}])$ steps. We slightly improve Dixon's rigorous analysis of his Monte Carlo factoring algorithm. We prove that this algorithm with probability 1/2 detects a proper factor of every composite n within $o(\exp \sqrt{6 \ln n \ln \ln n})$ steps.

1. A Refined Analysis of Dixon's Probabilistic Factoring Algorithm.

So far the asymptotically fastest run time of a factoring algorithm has been proved by Dixon (1978). Given a composite number n, this algorithm finds a proper factor of n with probability 1/2 within $O(\exp(4\sqrt{\ln n \ln \ln n}))$ steps. ln denotes the "logarithmus naturalis" with the Eulerian number e as base and exp is the inverse function to ln. Dixon mainly applies the method of "combining congruences" to generate solutions of $x^2 = y^2 \bmod n$. In Sections 2 and 3 we will see that this technique can well be combined with factoring algorithms proposed by J.C.P. Miller (1975) and D. Shanks (1971). We give an outline of Dixon's algorithm with an improved analysis. We decrease the constant 4 in

* This work has been started in Summer 1980 during a stay at the Stanford Computer Science Department. Preparation of this report was supported in part by National Science Foundation grant MCS-77-23738 and by the Bundesminister für Forschung und Technologie.

Dixon's bound to $\sqrt{6}$. The improved theoretical time bound results from a tighter lower bound on the number of quadratic residues mod n which can be completely factored over small primes (Lemma 1) and a specific method for detecting small prime factors. Here we do not focus on designing the most practical algorithm but we like to prove a rigorous asymptotical time bound as small as possible. We do not assume any distribution on the input data but we assume that some intermediate data are chosen at random.

Dixon's algorithm.

begin input n

stage 1 $\nu = \lfloor n^{1/2r} \rfloor$
comment the optimal choice of $r \in \mathbb{N}$ will be made below.
Form the list P of all primes $\leq \nu : P = \{p_1, .., p_{\pi(\nu)}\}$.
if $\exists\, p_i \in P : p_i \mid n$ then print p_i stop
$B := \emptyset$

stage 2 Choose $z \in [1, n-1]$ at random and independently from previous choices of z.
if $\gcd(z,n) \neq 1$ then print $\gcd(z,n)$ stop
$w := z^2 \bmod n$ with $0 \leq w < n$

stage 3 Compute $\underline{a} = (a_i \in \mathbb{N} \mid 1 \leq i \leq \pi(\nu))$ and w* with $w = w^* \prod_{i \leq \pi(\nu)} p_i^{a_i}$ and $\forall p \in P : p$ does not divide w*.

test 1 if $w^* \neq 1$ then goto stage 2
$B := B \cup \{\underline{a}\}$, $z_{\underline{a}} := z$
Try to find a nontrivial solution of

$$\sum_{\underline{a} \in B} f_{\underline{a}} \underline{a} = \underline{0} \bmod 2; \quad f_{\underline{a}} \in \{0,1\}. \tag{1}$$

test 2 if there is no nontrivial solution then goto stage 2

$$x := \prod_{f_{\underline{a}}=1} z_{\underline{a}}, \qquad y := \prod_{i \leq \pi(\nu)} p_i^{(\sum_{\underline{a} \in B} f_{\underline{a}} a_i)/2}$$

comment The construction implies $x^2 = y^2 \bmod n$; in case $x \neq \pm y \bmod n$, $\gcd(x \pm y, n)$ are proper factors of n.

test 3 if $x \neq \pm y \bmod n$ then print $\gcd(x \pm y, n)$ stop
Choose the first $\underline{a} \in B$ such that $f_{\underline{a}} = 1$.
$B := B - \{\underline{a}\}$, goto stage 2

end

Obviously a proper factor of n has been found as soon as test 3 succeeds. In the following analysis of the algorithm we suppose that n is an odd number with prime factor decomposition:

$$n = \prod_{i=1}^{d} q_i^{l_i} \qquad l_i \geq 1 \text{ and } d \geq 2.$$

Clearly the cases that n is even or a pure prime power can easily be handled in advance. The following facts are due to Dixon.

Fact 1. prob(x=±y mod n within test 3)$=2^{1-d}$ and the corresponding events for distinct passes of test 3 are mutually independent.

Let T(n) be the total time of the algorithm and let $T_3(n)$ be the time till the first pass of test 3. We count arithmetical steps mod n as single steps. T(n), $T_3(n)$ are random values depending on the random variables z of stage 2. Fact 1 immediately implies:

Fact 2. $E[T(n)] = (1-2^{1-d})^{-1} E[T_3(n)] \leq 2E[T_3(n)]$.
Here E[X] denotes the expectation of the random value X. Let $T_1(n)$ ($T_2(n)$, resp.) be the time spent from any entering of stage 2 till the first pass of test 1 (test 2, resp.) without counting the steps used to solve the various linear systems of equations (1). Since a linear dependence of the $\underline{a}$ with $\underline{a} \in B$ must exist as soon as $\#B \geq \pi(\nu)+1 = O(\nu/\ln\nu)$ it follows that there are at most $\pi(\nu)+1$ passes of test 2 before the first pass of test 3. Hence

Fact 3. $E[T_3(n)] \leq (\pi(\nu)+1)E[T_2(n)]+O(\pi(\nu)^3)$.
Here $O(\pi(\nu)^3)$ bounds the steps to solve all the linear systems (1) occurring in the various passes of stage 3. Indeed this task amounts to solve one system of linear equations with $\pi(\nu)+1$ unknowns. In order to analyze $E[T_2(n)]$ we define

$$Q := \{\text{set of quadratic residues mod } n\} \cap \mathbb{Z}_n^*$$
$$T(n,\nu) := \{r \in [1,n] : \text{all prime factors of } r \text{ are } \leq \nu\}$$
$$M(n,\nu) := \{z \in [1,n] : z^2 \bmod n \in Q \cap T(n,\nu)\}.$$

Let $\varphi(n) = \#\mathbb{Z}_n^*$ be the Eulerian function.

Fact 4. $E[T_2(n)] \leq O(E[T_1(n)]\,\varphi(n)/\#M(n,\nu))$.
Proof. We clearly have prob(w =1)$=\#M(n,\nu)/\varphi(n)$. Hence test 1 will atmost be passed about $\varphi(n)/\#M(n,\nu)$ times between two passes of test 2. □

$T_1(n)$ depends on how the factorization of w over the prime base P is done. An obvious bound is as follows:

Fact 5. $E[T_1(n)] \leq \pi(\nu)+\log n$.

Here log n bounds the number of multiple prime factors of n according to their multiplicity.

So far Facts 1-5 yield under the assumption $\log n \le \pi(\nu)$:

$$E[T(n)] \le O\left(\pi(\nu)^2\left[\frac{n}{\#M(n,\nu)} + \pi(\nu)\right]\right) \qquad (2)$$

and it remains to prove a sharp lower bound on $\#M(n,\nu)$. This will be our main improvement over Dixon's analysis. Let $\kappa: \mathbb{Z}_n^* \to \{\pm 1\}^d \approx \oplus_{i=1}^d Z_2$ be the quadratic character, defined as follows. For $a \in \mathbb{Z}_n^*$ let $\kappa(a) = (e_1,\ldots,e_d)$ with $e_i = \left(\frac{a}{q_i^{l_i}}\right)$. By definition the Jacobi symbol $\left(\frac{b}{q}\right)$ is 1,(-1, resp.) if b is a quadratic residue (non-residue) mod q. It is well known that $\kappa: \mathbb{Z}_n^* \to \oplus_{i=1}^d Z_2$ is a group homomorphism and $a \in Q$ iff $\kappa(a)$ is the group unit $(1,1,\ldots,1) \in \{\pm 1\}^d$.

Lemma 1. $\#M(n,\nu) \ge \pi(\nu)^{2r}/(2r)!$ for all natural numbers r with $\nu^{2r} \le n$ provided all prime factors of n are $>\nu$.

Proof. Let $T_r(m,\nu) := \{w \in [1,m] \mid w = \Pi_{p_i \le \nu} p_i^{a_i} \wedge \Sigma_i a_i = r\}$. Since all prime factors of n are $>\nu$ we have $T_r(\sqrt{n},\nu) \in \mathbb{Z}_n^*$. We partition $T_r(\sqrt{n},\nu)$ into classes $T_i, i=1,\ldots,2^d$ according to the 2^d possible values of κ. Then

$$\bigcup_{i=1}^{d} T_i T_i \subset T_{2r}(n,\nu) \cap Q.$$

Since for each $w \in T_{2r}(n,\nu) \cap Q$, $\#\{z \in \mathbb{Z}_n^* \mid z \bmod n\} = 2^d$ it follows

$$\begin{aligned} \#M(n,\nu) &\ge 2^d \#(T_{2r}(n,\nu) \cap Q) \\ &\ge 2^d \Sigma_{i=1}^{2^d} \#T_i^2 \frac{r!^2}{(2r)!} \end{aligned} \qquad (3)$$

Here $(\#T_i)^2$ counts the number of ordered pairs $(w_1,w_2) \in T_i \times T_i$ and $(2r)!/(r!)^2$ bounds for each $w \in Q$ the number of distinct pairs $(w_1,w_2) \in \cup_i T_i \times T_i$ that yield the product $w_1 w_2 = w$. The Cauchy Schwarz inequality implies

$$\Sigma_{i=1}^{2^d} (\#T_i)^2 \ge 2^{-d} (\Sigma_i \#T_i)^2 = 2^{-d} \#T_r(\sqrt{n},\nu)^2 \qquad (4)$$

(use $\Sigma_i u_i^2 \cdot \Sigma_i v_i^2 \ge (\Sigma_i u_i v_i)^2$ with $u_i = \#T_i, v_i = 1$).

Obviously we have $\#T_r(\sqrt{n},\nu) = \binom{\pi(\nu)+r-1}{r} \ge \pi(\nu)^r/r!$, since $\binom{\pi(\nu)+r-1}{r}$ is

the number of possibilities of choosing with repetitions r elements out of $\pi(\nu)$. Finally we obtain from (3), (4):

$$\#M(n,\nu) \geq \#T_r(\sqrt{n},\nu)\ \frac{r!^2}{(2r)!} \geq \frac{\pi(\nu)^{2r}}{r!^2}\ \frac{r!^2}{(2r)!} = \frac{\pi(\nu)^{2r}}{(2r)!} \quad \blacksquare$$

Putting (2) and Lemma 1 together we obtain

$$E[T(n)] = O\left(\pi(\nu)^2\left[\frac{n(2r)!}{\pi(\nu)^{2r}} + \pi(\nu)\right]\right)$$

provided $\log n \leq \pi(\nu)$ and $\nu^{2r} \leq n$. Using $\nu = n^{1/2r}$, $\nu \ln\nu \leq \pi(\nu) \leq 2\nu/\ln\nu$ (which follows from the prime number theorem) and $(2r)! = O(\sqrt{2r}(2r)^{2r}e^{-2r})$ (which follows from Stirling's formula) we obtain

$$E[T(n)] = O\left(\frac{(4r)^2 n^{1/r}}{(\ln n)^2}\left[\sqrt{2r}\,e^{-2r}(\ln n)^{2r} + \frac{4rn^{1/2r}}{\ln n}\right]\right) \qquad (5)$$

We choose $r \in \mathbb{N}$ as to minimize $n^{1/r}(\ln n)^{2r}$. This implies

$$r = \frac{1}{\sqrt{2}}\sqrt{\frac{\ln n}{\ln\ln n}} + \varepsilon\,,\ |\varepsilon| \leq 1/2$$

and

$$n^{1/r}(\ln n)^{2r} = O(\ln n \exp\sqrt{8 \ln n \ln\ln n}).$$

This finally yields the

Proposition 1.
$$E[T(n)] = O\left(\frac{\sqrt{2r}\,e^{-2r}}{\ln\ln n}\exp\sqrt{8 \ln n \ln\ln n}\right)$$
$$= o(\exp\sqrt{8 \ln n \ln\ln n}).$$

The asymptotic behaviour of this bound is quite attractive for excessively large n: n can be factored within $n^{\varepsilon(n)}$ steps with $\varepsilon(n) \to 0$ for $n \to \infty$. However, for reasonably sized values the exponent $\varepsilon(n)$ is not much smaller than 0.5 and the algorithm is not practical.

Can the above analysis of Dixon's algorithm still be refined leading to a constant in the exponent which is smaller than $\sqrt{8}$? We discuss two main points, (a) the tightness of our lower bound on $\#M(n,\nu)$ in Lemma 1, (b) the use of more sophisticated factoring algorithms for factoring w over the prime base P in stage 2.

We clearly have $\#M(n,\nu) \leq \psi(n,\nu) := \#\{w \in [1,n]$: all prime factors of w are

$\leq \nu\}$ The asymptotic behavior of $\psi(n,\nu)$ has been analyzed for a long time, see De Bruijn (1966) and Knuth, Trabb Pardo (1976). However, no exact values of $\psi(n,n^{1/2r})$ have been published for large n, say $n=2^{2^\nu}$ $\nu=7,8,9$ and reasonable r, say $4\leq r\leq 10$.

Instead of using within stage 2 the straightforward factoring algorithm that leads to Fact 5 we could use one of Pollard's algorithms that finds actors $\leq\nu$ of n in about $O(\sqrt{\nu})$ steps. By computational experience, Pollard's ρ-method (1975) detects factors $\leq\nu$ of n in $O(\sqrt{\nu \ln \nu})$ arithmetical steps mod n, see Guy (1975) and Knuth (1980). This method is highly practical although no rigorous theoretical time bound is known so far. Recently Brent succeeded in factoring $F_8=2^{2^8}+1$ by a variant of this method. Pollard (1974) also proposed a second method with a rigorous time bound. He computes for sufficiently many small $a\in\mathbb{Z}_n^*$, $\gcd(\Pi_{\nu\leq\sqrt{\nu}}(a^{\nu\sqrt{\nu}}-a^{-\mu}),n)$ for $\mu=1,2,\ldots,\nu$. For fixed a, these gcd-values can be computed by the fast Fourier transform within $O(\sqrt{\nu}(\ln \nu)^2 \ln\ln \nu)$ steps. In total, Pollard obtains a worst case time bound $O(\nu^{0.5+\varepsilon})$ for arbitrarily small $\varepsilon>0$, but the constant factor, expresses by O, increases in an unknown way as ε decreases. We give a similar but slightly stronger result, see Schnorr (1980) for a detailed proof, also compare Straßen (1976).

Lemma 2. The smallest prime factor $\leq\nu$ of n can be found in $O(\sqrt{\nu}(\ln \nu)^2 \ln\ln \nu)$ arithmetical steps mod n, provided $\ln n=O(\ln \nu)^2$.

Using the above procedure in stage 3 of Dixon's algorithm for factoring w over primes $\leq\nu$ clearly improves Fact 5 to

Fact 6. $T_1(n) = O(\nu(\ln \nu)^2 \ln\ln \nu)$.

This finally improves the bound of proposition 1 to $E[T(n)] = (\exp \sqrt{\ln n \ln\ln n})$. Thus we obtain the

Theorem 1. For each composite n let $E[T(n)]$ be the expected time that the above algorithm takes to find a proper factor of n. Then for all n

(1) $E[T(n)] = o(\exp \sqrt{6 \ln n \ln\ln n})$.

(2) The event that the algorithm does not find a proper factor of n within $kE[T(n)]$ steps has probability $\leq 2^{-k}$.

Statement (2) is an immediate consequence of the fact that the distinct events of "test 3" (test 1, resp.) failing" are mutually independent. A more practical factoring algorithm is obtained if the quadratic re-

sidues w in stage 2 are produced via the continuous fraction method (see Morrison and Brillhart, 1975) which implies $w=O(\sqrt{n})$ and if Pollard's ρ-method is used for detecting small prime factors of w.

Under the assumption

(AO) the continuous fraction of $\sqrt{n}$ generates quadratic residues mod n which are uniformly distributed in $[1,O(\sqrt{n})]$

the time bound (5) transforms into a time bound

$$E[T(n)] = O\left(n^{3/4r} \ln n\, e^{-r}(\ln n)^r + n^{3/2r}\left(\frac{2r}{\ln n}\right)^3\right) \quad (8)$$

with r even, for the Morrison-Brillhart method. By choosing

$$r = 2\left\lceil \frac{1}{4}\sqrt{\frac{3 \ln n}{\ln \ln n}} \right\rceil$$

we obtain

$$n^{3/4r}(\ln n)^r = O((\ln n)^2 \exp\sqrt{3 \ln n \ln \ln n})$$

$$n^{3/2r} = O(\exp\sqrt{3 \ln n \ln \ln n}).$$

By (8) this implies

Corollary 1. [Assume (AO)]. The Morrison-Brillhart method runs in average time $o(\exp\sqrt{3 \ln n \ln \ln n})$.

This last method is really practical. Wunderlich (1979) obtained average runtimes around $322n^{0.152} \approx n^{0.21}$ for $n \approx 10^{40}$.

2. An Analysis and Revision of J.C.P. Miller's Factoring Method.

J.C.P. Miller (1975) proposed a factoring method based on the computation of indices. We shall develop a slightly improved version of Miller's method which turns out to be quite similar to the previously analyzed Dixon algorithm. Under reasonable heuristic assumptions the runtime of our version of Miller's algorithm will be $O(\exp\sqrt{4.5 \ln n \ln \ln n})$. In particular Miller's method does not yield an independent factoring algorithm but merely a specific modification of the method of "combining congruences mod n". However, as we shall point out, this modification has some decisive advantages in the case that one likes to factor many numbers in the same range. So far all known factoring algorithms collect data which are only useful for factor-

ing one specific number. For instance the congruences collected in Dixon's algorithm cannot be used for different n's. This observation also applies to the factoring algorithms of Morrison-Brillhart (1975), Schroeppel (unpublished, see Monier 1980), Shanks (1971, 1974), and Pollard (1974, 1975). In our version of Miller's method we will collect products of small prime numbers which are near to the number n to be factored. These products of small primes can be uniformly used for factoring all numbers near to n. For the connection to Miller's method, see Schnorr (1980).

As an example, let n = 1037

stage 1: Generate many distinct representations of n or multiples of n as a sum or difference of two products of small primes. For instance we have

$$\begin{aligned} 1037 &= 2^8 5 - 3^5 \qquad \text{i.e.} \qquad 2^8 5 = 3^5 \bmod n \\ &= 2\cdot 3\cdot 5^2\cdot 7 - 13 \qquad 2\cdot 3\cdot 5^2\cdot 7 = 13 \bmod n \\ &= 2^2 3^5 + 5\cdot 13 \qquad 2^2 3^5 = -5\cdot 13 \bmod n \\ &= 3\cdot 7^3 + 2^3 \qquad 3\cdot 7^3 = -2^3 \bmod n \end{aligned}$$

We obtain by multiplying the above congruences:

$$2^{11} 3^7 5^3 7^4 = 2^3 3^5 5\cdot 13^2 \bmod n$$

Since no prime of our base divides n, this yields

$$2^8 3^2 5^2 7^4 = 13^2 \bmod n.$$

From $2^4\cdot 3\cdot 5\cdot 7^2 = 353 \bmod n$ we obtain

$$353^2 = 13^2 \bmod n$$

which gives us the proper factors

$$\gcd(353 - 13, n) = 17$$
$$\gcd(353 + 13, n) = 61.$$

A formal description of our method is as follows.

<u>begin</u> input n
$\nu := n^{1/2r}, u := n^{d/2r}$
<u>comment</u> the optimal choice of r and d will be made below
Form the list $P = \{p_0, p_1, \ldots, p_{\pi(\nu)}\}$ of all primes $\leq \nu$, including $p_0 = -1$