SIMD型超並列計算機における素因数分解

高橋大介[†] 鳥居泰伸[††,☆] 湯淺太一[†††]

本論文では，SIMD型超並列計算機の新しい応用分野として素因数分解法の一つである楕円曲線法を考察する．SIMD型超並列計算機であるMasPar MP-1上で楕円曲線法を実現して，得られる結果について述べる．楕円曲線法は分散処理が可能であり，実行時における分岐やループ回数の不揃いという問題が少ないため，SIMD型超並列計算機でも高い並列度が維持される．ランダム整数の素因数の理論を楕円曲線法に適用することによって，従来の逐次型計算機との戦略の違いを明らかにし，SIMD型超並列計算機におけるアルゴリズムを検討する．このアルゴリズムで，円分数の素因数分解を行った結果，円分数において最大41桁の素因数が見つかった．また，$b^n \pm 1$の形をしたカニンガム数の分解にも成功した．これらの結果は現在のところ楕円曲線法において世界的な水準に匹敵するものであり，SIMD型超並列計算機が大きな整数を素因数分解するのに非常に適していることが示された．

An Implementation of Factorization on Massively Parallel SIMD Computers

DAISUKE TAKAHASHI,[†] YASUNOBU TORII[††,☆] and TAIICHI YUASA[†††]

In this paper, we consider the Elliptic Curve Method (ECM) for factorization as a new application on massively parallel SIMD computers. We have implemented ECM on the SIMD massively parallel computer MasPar MP-1, and have evaluated the results of the implementation. ECM has high parallelism and the parallelism is maintained on SIMD computers owing to a few branches and numbers of iterations for all processors. We clarify the difference of the strategy from that on conventional sequential computers, and discuss our algorithm for massively parallel SIMD computers. By factoring several cyclotomic numbers with this algorithm, we discovered a 41-digit prime factor. We also succeeded in factoring some Cunningham numbers $b^n \pm 1$. These results are very close to the largest prime factor so a discovered by ECM. It is shown that massivery SIMD computers are highly useful for factoring large numbers.

1. はじめに

最近はVLSI技術の進歩により，数百個から数千個のプロセッサを接続した並列計算機を作ることが可能になってきている．並列計算機はSIMD（Single Instruction stream Multiple Data stream）型とMIMD（Multiple Instruction stream Multiple Data stream）型に大別される．SIMD型はMIMD型に比べプログラミングが容易という利点があるが，データによる分岐が多い計算や繰り返し回数の不揃いな計算には不向きという欠点がある．そのためSIMD型並列計算機は画像処理などに用途が限られているのが現状である．

一方で大きな整数を素因数分解する問題は数論の分野で古くから難問とされている．特に1970年代後半から少ない計算量で素因数分解を行うアルゴリズムが盛んに研究されている．素因数分解の応用としてRSA公開鍵暗号[1]がある．この暗号の安全性は，大きな整数の素数判定は比較的容易であるが，素因数分解が困難であることによる．

素因数分解法の一つである楕円曲線法[2),3)]は分散処理が可能であり，実行時において高い並列度が維持できる．さらに，各プロセッサごとに独立した計算ができるので，通信量が少ないといった特徴もある．そのため楕円曲線法はSIMD型を含む並列計算機に非常

† 東京大学大学院理学系研究科情報科学専攻
Department of Information Science, Graduate School of Science, University of Tokyo
†† 豊橋技術科学大学電気・電子工学系
Department of Electrical and Electronic Engineering, Toyohashi University of Technology
☆ 現在，富士通（株）
Presently with Fujitsu Limited
††† 豊橋技術科学大学情報工学系
Department of Information and Computer Sciences, Toyohashi University of Technology

に適した手法である．

本論文ではSIMD型並列計算機の新しい応用分野として，楕円曲線法と呼ばれる素因数分解アルゴリズムを実行した結果について論じる．

SIMD型超並列計算機上で楕円曲線法を実行するとき，従来の逐次型計算機やMIMD型並列計算機とは異なった独自の工夫を試み，その効果を考察した．楕円曲線法の実行はSIMD型超並列計算機MasPar MP-1[4]を用い，SIMD型のコードを意識したプログラムを作成した．現在進行中である素因数分解プロジェクトに参加し，数値実験を行った．実際に得られた成果からSIMD型超並列計算機が大きな整数を素因数分解するのに非常に適していることを示す．

まず，2章で楕円曲線法の原理と計算量について述べる．次に3章でSIMD型超並列計算機におけるアルゴリズム，4章でSIMD型並列計算機における並列化効率を議論する．本論文ではSIMD型並列計算機における並列化効率向上よりも，処理時間が短くなるアルゴリズムを考察する．5章で今回得られた成果を検討・考察する．最後に6章で結論を述べる．

2. 楕円曲線法

本章では，素因数分解に用いた楕円曲線法の原理を示す．楕円曲線法とは，楕円曲線を合成数の素因数を法とする剰余類体で考えたとき，そこで定義される加法が有限可換群になり，その群の位数が素因数ごとに異なることを利用したものである．

2.1 楕円曲線

楕円曲線 E

$$E:\ By^2 = x^3 + Ax^2 + x \tag{1}$$

を体 F 上で考えると，E 上のすべての点の集合 $E(F)$ は可換な加法群をなす．加法の単位元は無限遠点であり，これを零点と呼び $\boldsymbol{O}$ で表す．E 上の点 $\boldsymbol{P}=(x,y)$ の逆元 $-\boldsymbol{P}$ は $(x,-y)$ である．

E 上の任意の2点 $\boldsymbol{P}_1=(x_1,y_1)$ と点 $\boldsymbol{P}_2=(x_2,y_2)$ に対する加法は次のように定義される．2点のいずれかが零点のときは和は自明であり，もし $\boldsymbol{P}_1$ が $\boldsymbol{P}_2$ の逆元ならばその和は $\boldsymbol{O}$ である．それ以外の場合，つまり，$\boldsymbol{P}_1 \neq -\boldsymbol{P}_2$ のとき，和 $\boldsymbol{P}_1+\boldsymbol{P}_2=\boldsymbol{P}_3=(x_3,y_3)$ を

$$\begin{cases} x_3 = Bk^2 - x_1 - x_2 - A \\ y_3 = k(x_1 - x_3) - y_1 \end{cases}$$

と定義する．ここで，

$$k = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if} \quad \boldsymbol{P}_1 \neq \boldsymbol{P}_2 \\ \dfrac{3x_1^2 + 2Ax_1 + 1}{2By_1} & \text{if} \quad \boldsymbol{P}_1 = \boldsymbol{P}_2 \end{cases}$$

である．点 $\boldsymbol{P}$ を M 回加えた点を $M\boldsymbol{P}$ と表す．

楕円曲線 E を素数 p に関する剰余類体 $GF(p)$ 上で考えたときの，$E(GF(p))$ の位数を S で表す．$E(GF(p))$ 上のすべての点 $\boldsymbol{P}$ に対して $S\boldsymbol{P}$ は零点 $\boldsymbol{O}$ になる．また M が S の倍数のときも $M\boldsymbol{P}=\boldsymbol{O}$ になる．位数 S は楕円曲線の係数 A, B に依存するが，Hasseの定理（例えば文献5), 6)を参照）によると S の値は

$$p+1-2\sqrt{p} < S < p+1+2\sqrt{p} \tag{2}$$

の範囲にある．

2.2 アルゴリズム

以下では，実際のプログラムに利用した，Montgomeryによる改良版楕円曲線法[3]について説明する．

ある合成数 N の素因数を楕円曲線法によって求めるには，まず式(1)で与えられる楕円曲線 E の A, B を適当な値に設定し，初期点 $\boldsymbol{P}(x_1,y_1)$ を E 上に適当に選ぶ．次に点 $\boldsymbol{P}$ を素数べきの積 $(M = 2^{n_1}\times 3^{n_2}\times 5^{n_3}\times\cdots\times L_1{}^{n_m})$ 倍することによって，点 $M\boldsymbol{P}$ を求める．(ここで L_1 は適当な上界である．)実際の計算では，M を計算すると莫大な数になるため，$\boldsymbol{P}$ を直接 M 倍するようなことをせず，$\boldsymbol{P}$ を 2^{n_1} 倍し，さらにそれを 3^{n_2} 倍し，さらにそれを 5^{n_3} 倍しというようにして $M\boldsymbol{P}$ を求める．

m と n を相異なる整数とすると，式(1)の楕円曲線上の点 $m\boldsymbol{P}, n\boldsymbol{P}, (m-n)\boldsymbol{P}, (m+n)\boldsymbol{P}, (2n)\boldsymbol{P}$ の x 成分の値の間には以下の式が成り立つ．

$$x_{m+n} = \frac{(x_m x_n - 1)^2}{x_{m-n}(x_m - x_n)^2} \tag{3}$$

$$x_{2n} = \frac{(x_n^2 - 1)^2}{4x_n(x_n^2 + Ax_n + 1)} \tag{4}$$

点 $k\boldsymbol{P}$ の x 成分を有理数 X_k/Z_k と表すと，式(3), (4)は次式で与えられる．

$$\begin{cases} X_{m+n} = Z_{m-n}(X_m X_n - Z_m Z_n)^2 \\ Z_{m+n} = X_{m-n}(X_m Z_n - Z_m X_n)^2 \end{cases} \tag{5}$$

$$\begin{cases} X_{2n} = (X_n^2 - Z_n^2)^2 \\ Z_{2n} = 4X_n Z_n(X_n^2 + AX_n Z_n + Z_n^2) \end{cases} \tag{6}$$

$\boldsymbol{P}$ を r 倍するときには，文献12)に従って r を2進展開して式(5), (6)を順次適用する．このようにすれば $\boldsymbol{P}$ を r 倍するのに必要な反復回数は $\lceil \log_2 r \rceil$

だけですむ．M が $E(GF(p))$ の位数 S の倍数であれば，$M\boldsymbol{P}$ は零点 $\boldsymbol{O}$ になり，このとき

$$Z_M \equiv 0 \pmod{p} \tag{7}$$

となる．実際には p は未知数であるため，式 (7) は素因数分解には直接使うことができない．そのため，N と Z_M で最大公約数をとり，その値が 1 より大きく N より小さければ N の因数が見つけられる．その因数が素数 p ならば N の素因数が見つかったことになる．

Hasse の定理から位数 S の範囲は p によって制限され，一般には N よりはるかに小さい数である．このために，任意の M がたまたま位数 S の倍数になる確率は比較的高い．Hasse の定理によると，素因数 p が大きくなると S も大きくなり，p を見つけるのが困難になってくる．このようなときは素数べきの積 M の素数を大きくしたり，楕円曲線を変えて位数 S を変化させ p が見つかる可能性を増やしていく．

式 (1) で与えられる楕円曲線 E の A と初期点 x_1 の選び方としては木田ら[13]が行っている方法を我々は採用した．u を任意の有理数として，

$$a = \frac{2u}{3u^2-1} \tag{8}$$

とおき，

$$A = \frac{-3a^4-6a^2+1}{4a^3}, \quad x_1 = \frac{3a^2+1}{4a} \tag{9}$$

と定める．B については $\boldsymbol{P}$ を素数べき倍する計算で扱うことはないが初期点 x_1 によって定まるものと考える．この方法では位数 S が合成数（12 の倍数）になるので，式 (9) のように A と初期点 x_1 を選ばない場合に比べて，素因数が見つかる可能性が少し高くなる．

今まで説明した $M\boldsymbol{P}$ を計算する操作を第 1 段階と呼ぶ．実際の位数 S を素因数分解すると M を構成する素数のほかにやや大きい素数 q が一つだけ乗じられている場合がある．このようなときには実際に $M\boldsymbol{P}$ を q 倍することなく $qM\boldsymbol{P}$ が $E(GF(p))$ の零点 $\boldsymbol{O}$ になっているかを判定する方法があり，これを第 2 段階と呼ぶ．その方法を簡単に説明すると，$q = 420t \pm s$ $(0 < s < 210$ で s と 210 は互いに素$)$ とおいて

$$x_{420tM} - x_{\pm sM} \equiv 0 \pmod{p} \tag{10}$$

となるかで判定する．上の条件を満たす s は 48 個あり，t を 1 回固定するごとに式 (10) の判定を 48 回行う．実際の計算では第 2 段階を開始する前に式 (10) の $x_{\pm sM}$ の部分を法 N による逆数計算[15]によって整数化し，s の個数である 48 個分保存しておく．そして t を順に増やしながら判定を進めていく．このとき x_{420tM} の部分は式 (5) の加法公式に従い分子，分母をそれぞれ求めていく．

式 (10) の判定を行うとき，この x_{420tM} の部分を逆数計算によって整数化して行う方法と，この部分を分数扱いにし分母を払って

$$X_{420tM} - x_{\pm sM} Z_{420tM} \equiv 0 \pmod{p} \tag{11}$$

とした式を用いて行う方法がある．前者は 1 回の逆数計算により 48 回の乗算，剰余算を減らすことができる．

以下に第 1 段階と第 2 段階の処理速度比を考察する．ここでは大ざっぱな方法ではあるが，準備やチェックを除いた部分の乗算数で比較を行い，加減算は考慮しないものとする．また，第 2 段階では式 (10) の判定式を用い，逆数計算はない状態での比較とする．

ある程度大きい L に対して，L から $L+\Delta L$ まで第 1 段階を行ったときの乗算数を見積もる．点 $\boldsymbol{P}$ を L 倍するのに 2 進展開法により反復数は $\log L$ と近似する．1 回のループで加法公式，倍角公式を 1 回ずつ使用する．乗算数は加法公式で 6 回，倍角公式で 5 回である．また素数定理の近似により，L 付近での整数が素数である確率は $1/\log L$ である．よって第 1 段階における当区間の乗算数は

$$\log L \times (6+5) \times \frac{\Delta L}{\log L} = 11\Delta L$$

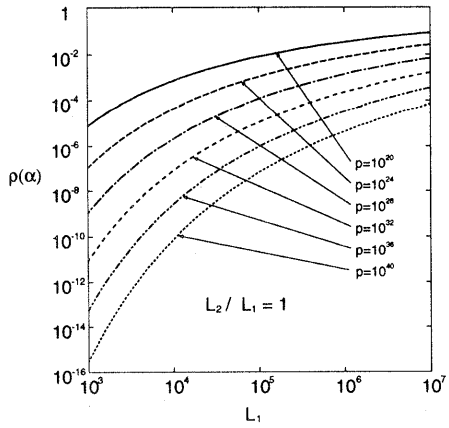
となる．

一方，第 2 段階では L が 420 だけ進む間に式 (10) の判定が 48 回行われる．式 (10) の左辺を計算するのに乗算 1 回と時間がかかる最大公約数の計算をなるべく回避するために左辺同士を掛け合わせてから判定を行うため，乗算はもう 1 回必要となる．さらに x_{420tM} を更新するのに前回，前々回の結果を使うことにより，1 回の加法公式，すなわち 6 回の乗算が行われる．よって ΔL だけの区間を進めるためには

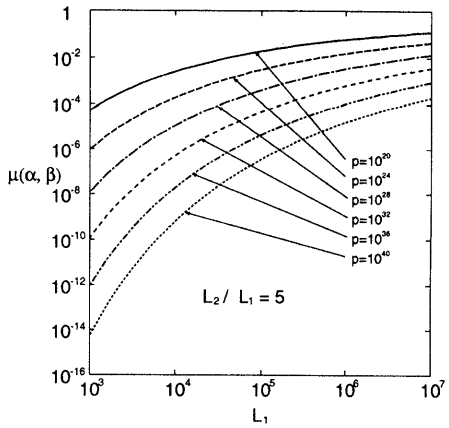
$$[(1+1) \times 48 + 6] \times \frac{\Delta L}{420} = 0.243\Delta L$$

の乗算が行われることになる．第 1 段階と第 2 段階の処理速度比はこれらの比をとって $11/0.243 \approx 45$ 倍となる．実際には乗算以外の演算もあり，第 2 段階で式 (11) の判定式を使うことがあるので，速度比は 45 倍を前後する．

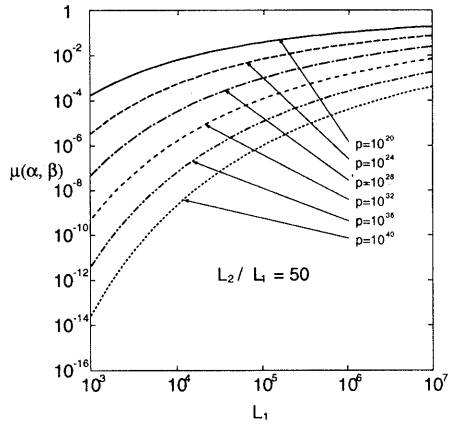
楕円曲線法における成功の条件をまとめると以下のようになる．位数 S の最大素因数が第 1 段階の上界 L_1 以下であれば，第 1 段階だけで素因数 p が見つかる．そして S の最大素因数が第 2 段階における q の上界（これを L_2 とする）以下であり，かつ S の 2 番目に大きい素因数が L_1 以下である場合にも第 2 段階



(a) $L_2/L_1 = 1$ の場合



(b) $L_2/L_1 = 5$ の場合



(c) $L_2/L_1 = 50$ の場合

図1 楕円曲線法の成功率
Fig. 1 Success rate of elliptic curve method.

で素因数 p が見つかる．

2.3 計 算 量

楕円曲線法の計算量を見積もるにあたって，ランダム整数の素因数の理論[16]を簡単に示しておく．十分大きな正のランダム整数 M を考え，M の素因数を大きい順に $n_1, n_2, \cdots$ とする．このとき

$$\rho(\alpha) = \lim_{M\to\infty} \Pr(n_1 < M^{1/\alpha}) \quad (\alpha \geq 1) \quad (12)$$

$$\mu(\alpha, \beta) = \lim_{M\to\infty} \Pr(n_2 < M^{1/\alpha} \text{ and } n_1 < M^{\beta/\alpha}) \quad (\alpha \geq \beta \geq 1) \quad (13)$$

と定義する．この $\rho(\alpha)$ および $\mu(\alpha, \beta)$ は第1段階および第2段階で素因数が見つかる確率を示す．ここで $\rho(\alpha)$ に関しては次の微分方程式が成り立つ．

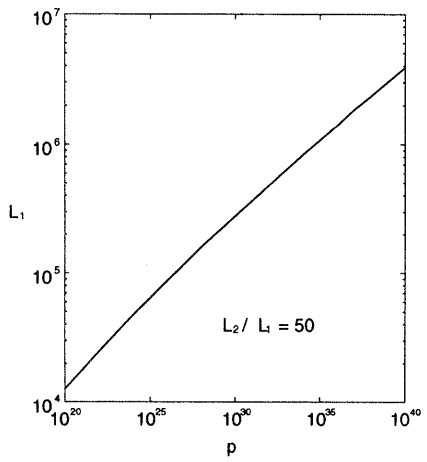
$$\alpha\rho'(\alpha) + \rho(\alpha - 1) = 0 \quad (14)$$

また $\mu(\alpha, \beta)$ に関しては次の式が成り立つ．

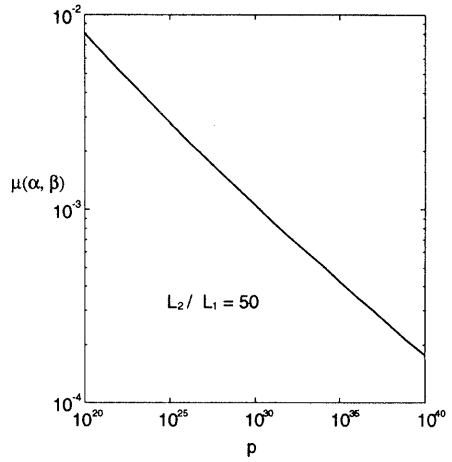
$$\mu(\alpha, \beta) = \rho(\alpha) + \int_{\alpha-\beta}^{\alpha-1} \frac{\rho(t)}{\alpha - t} dt \quad (15)$$

実際の楕円曲線法では位数 S は 12 の倍数になり，Hasse の定理から $S \approx p$ と近似して $\alpha = \log(p/12)/\log L_1$，$\beta = \log L_2/\log L_1$ とする．第1段階と第2段階においての素数べきの範囲 L_1, L_2 を定めたとき，楕円曲線法が成功する確率 $\rho(\alpha)$, $\mu(\alpha, \beta)$ を式 (14), (15) を用いて数値計算したものを**図1** (a)～(c) に示す．この図に示される $\rho(\alpha)$, $\mu(\alpha, \beta)$ は逐次型計算機における成功率を示し，これらの逆数は素因数を発見するまでに必要な曲線数の平均を表す．前述のように第2段階は第1段階よりも数十倍高速に処理でき，図1 (c) と図1 (a) を比べればわかるように成功率はかなり高い．このことから第2段階が極めて効果的であることが予想できる．

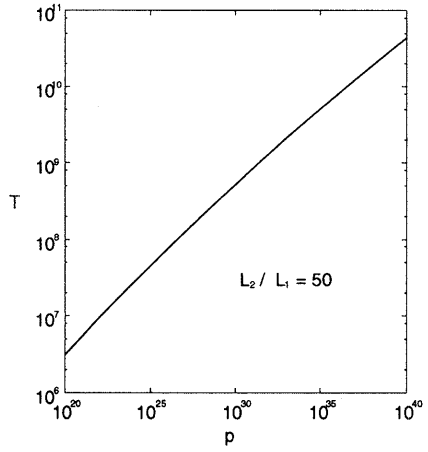
楕円曲線法における最適計算条件を考察する．第1



(a) 第 1 段階の上界



(b) 素因数が見つかる確率



(c) 計算量の期待値

図 2 楕円曲線法における最適条件
Fig. 2 Optimum condition of elliptic curve method and expected value of calculation quantity.

段階，第 2 段階の処理速度をそれぞれ V_1，V_2 とすると，両段階にまたがる計算量の期待値は次のようになる．

$$T = \left(\frac{L_1}{V_1} + \frac{L_2 - L_1}{V_2}\right) \Big/ \mu(\alpha, \beta) \qquad (16)$$

この T が最小になるような条件を求めてみる．第 1 段階と第 2 段階の処理速度比を $V_1 : V_2 = 1 : 50$ とする．一般的には第 1 段階と第 2 段階の上界の比を処理速度比に一致させることが多いので，$L_2/L_1 = 50$ とする．このとき，発見したい素因数 p に対する L_1 の値を**図 2** (a) に示す．また，この状態での成功率 $\mu(\alpha, \beta)$ と計算量の期待値 T を図 2 (b)，(c) に示す．

3. SIMD 型超並列計算機におけるアルゴリズム

楕円曲線法を並列計算機上で実現するとき，Lenstra ら[11]と同様に各プロセッサに一つずつ楕円曲線を割り当てる方法をとる．例えば，i 番目のプロセッサにおいては式 (8) の u を $i+2$ とすると，各プロセッサごとに異なった楕円曲線を割り当てることができる．一度にプロセッサの個数分だけの楕円曲線がテストできるので大きな素因数を発見できる可能性が高くなる．

第 1 段階では，計算時間の大部分が多倍長の乗算と剰余計算に費やされる．現在，楕円曲線法で分解が試みられている合成数の大きさは 300 桁くらいまで及んでおり，この程度の桁数に対する乗算には再帰的二分割法[15]が適している．また分解しようとする合成数を

N とするとき，楕円曲線法では常に $\bmod N$ で計算を進めればよいため，剰余計算には文献17)に掲載されているようなテーブル参照方式が利用できる．これらの計算は数値によるプログラムの分岐がほとんどなく，繰り返し部分の回数も揃っている．よって，SIMD型並列計算機においても並列度が高い状態で計算を進めることができる．

第2段階では式(10)の逆数を使う方法と式(11)の逆数を使わない方法の二つの判定方法があり，どちらが有利かを検討する必要がある．

乗算，剰余算それぞれ48回の計算時間が逆数計算1回よりも短ければ式(11)が有利となり，そうでなければ式(10)が有利となる．本論文では実際の計算に要する時間を測定することにより式(10)の方法を採用した．しかし，4章で詳しく述べるように，式(10)の逆数計算においては数値による分岐が多く，繰り返し部分の回数にばらつきがある．よってSIMD型並列計算機では，並列度が低くなる．それに逆数計算，乗算，剰余計算のアルゴリズムによって計算時間が左右されるため，インプリメントの方法により式(11)が有利になることもあり得る．SIMD型並列計算機では，逐次型計算機に比べ，第2段階の並列度が低下するが，それでも第1段階よりも格段に速く処理できる．

次に楕円曲線法を並列計算機で実行したときの成功率を考察する．プロセッサ数を P とするとき楕円曲線法の第1段階および第2段階における成功率はそれぞれ

$$\sigma = 1 - [1 - \rho(\alpha)]^P \tag{17}$$

および

$$\sigma = 1 - [1 - \mu(\alpha, \beta)]^P \tag{18}$$

となる．逐次型計算機における成功率 $\rho(\alpha)$，$\mu(\alpha, \beta)$ が小さいところでは，プロセッサの台数にほぼ比例して並列計算機における成功率 σ が大きくなり，楕円曲線法における並列計算機の威力が発揮される．

通常，逐次型計算機で楕円曲線法を行うとき，次のような戦略をとる．まず目標とする素因数の大きさを決め，その素因数の大きさに応じて最適条件に近い素数べきの範囲 L_1，L_2 およびテストする曲線数 C を決める．適当な楕円曲線からスタートし第1段階，第2段階をそれぞれ L_1，L_2 まで処理しても素因数が発見できないときは別の曲線に移り，同様な処理を繰り返す．曲線数を C だけテストしても素因数が発見できないときは L_1，L_2，C を拡げて再挑戦するか，楕円曲線法を断念しなければならない．楕円曲線法のパラメータ L_1，L_2，C はそれぞれトレードオフの関係にあるため，これらの値の設定が分解成功の決め手となる．

これに対しSIMD型やMIMD型の並列計算機で楕円曲線法を行うとき，従来の逐次型計算機の場合とは戦略が異なってくる．目標とする素因数発見に必要な曲線数 C よりプロセッサ数 P が多いとき，一度にテストできる曲線数が多くなる分だけ素数べきの範囲 L_1, L_2 が小さくできる．言い換えるとプロセッサ数に応じて処理する範囲を変えるといった戦略の変更も必要となってくる．また素因数の発見に失敗した場合も L_1 と L_2 だけを拡げて再挑戦すればよいという利点も出てくる．つまり設定するパラメータ数が一つ減るので並列計算機上では楕円曲線法の試行が容易となる．並列計算機のプロセッサ数 P が必要な曲線数 C より小さい時は逐次型計算機の場合と同様に L_1，L_2，C（ただし C は P の倍数）を設定して試行する．

4. 演算の並列化効率

楕円曲線法の実行には多種類の多倍長演算が必要となる．その中で時間的にウエートを占める演算に関して，SIMD型並列計算機での並列化効率を議論する．本論文では楕円曲線法全体を通しての処理時間が短くなるアルゴリズムを考察しているため，必ずしも最良の並列化効率は与えていない．

4.1 乗　　算

実用的な楕円曲線法では再帰的2分割法が有効となり，この方法では全プロセッサで計算順序が同一である．そのため並列化効率は100%となる．

4.2 剰 余 算

文献16)を改良した方法では，まず行列の乗算と同様な計算を行う．この部分は全プロセッサ同一の計算であるため並列化効率は100%となる．次に足し合わせた結果で桁あふれが生じた時の処理を行う．この部分はプロセッサによって実行される/されないという分岐が起こるが，剰余算の中では短時間の処理である．よって剰余算全体の並列化効率は100%近い状態となる．

4.3 除　　算

筆算方式に1ビットずつ処理していく方法では，数の大小によって引き算をする/しないという分岐が起こるため並列化効率は低くなる．また逐次型計算機においても剰余算に比べると計算時間がかかるため，極力除算は避けるようにして剰余算に置き換えている．

4.4 逆 数 計 算

法が奇数 N の下での逆数計算を論じるにあたって次のアルゴリズムを考える．このアルゴリズムは a/b と合同な数を求めるものである．最終的に得られる b

は最初の b と N の最大公約数であり，最終的な a/b が答となる．(最初の b と N が互いに素ならば a 自身が答である．)

```
c:=a;
d:=b;
a:=0;
b:=N;
while d ≠ 0 do begin
  for i:=1 to R do begin
    if d mod 2 = 0 then begin          ┐
      d:=d/2;                          │
      if c mod 2 = 1 then c:=c+N;      ├---(A)
      c:=c/2;                          │
    end if                             ┘
  end for
  if d mod 2 = 1 then begin
    if d<b then begin                  ┐
      swap(d,b);                       ├----------(B)
      swap(c,a);                       │
    end if                             ┘
    d:=d-b;
    c:=c-a;
  end if
end while ここで R は1以上の整数とする．
```

(A) の部分は最大 R 回 d が偶数である限り繰り返し実行される．(B) は d が奇数のときのみ実行される．条件文やジャンプの処理時間が無視できれば，逐次型や MIMD 型では R の値によらず逆数計算の時間は同じである．しかし SIMD 型の場合には R の値によって逆数計算の処理時間が変化する．

最初に (A) の部分における並列化効率を考える．2回目以降の while ループでは，ループの先頭の時点で d は偶数である．よって (A) の for ループの1回目が実行される確率は1である．以下 for ループが1回実行されるごとに d が偶数のままである確率は半分ずつになっていく．プロセッサ数の多い SIMD 型並列計算機では for ループの実行回数は R と近似できる．よって (A) の部分の並列度は

$$E_A = \frac{1}{R}\left(1+\frac{1}{2}+\cdots+\frac{1}{2^{R-1}}\right) = \frac{2}{R}\left(1-\frac{1}{2^R}\right)$$

となる．これに対して (B) の部分の並列化効率は，(A) で d が奇数となる確率であるから，

$$E_B = \frac{1}{2}+\frac{1}{2^2}+\cdots+\frac{1}{2^R} = 1-\frac{1}{2^R}$$

となる．R の値が大きくなると E_A は低下，E_B は上昇するという，互いにトレードオフの関係になる．

表1 逆数計算における while ループの並列化効率
Table 1 Parallelization factor of recipro calculate on while loop.

	P=16	64	256	1024	4096	16384
R=1	0.957	0.943	0.931	0.921	0.912	0.904
2	0.966	0.956	0.947	0.939	0.932	0.925
3	0.963	0.952	0.943	0.935	0.929	0.921
4	0.957	0.945	0.935	0.925	0.917	0.913
5	0.953	0.939	0.928	0.918	0.907	0.899
6	0.951	0.936	0.923	0.912	0.901	0.893

一方 while ループの実行回数のばらつきも並列度に影響を与える．そこでプロセッサ P に対するばらつきの並列度 E_L をシミュレーションで求めた．このシミュレーションでは while ループの平均実行回数をプロセッサ内での最大実行回数で割った値を E_L とし，その結果を**表1**にまとめた．パラメータ R の値によらずプロセッサ数の増加により E_L は低下していくことが分かる．

(A) と (B) の処理時間をそれぞれ T_A，T_B とすると逆数計算全体の効率 E_T は

$$E_T = E_L\left(\frac{T_A E_A + T_B E_B}{T_A + T_B}\right) \tag{19}$$

となる．$R=2$ のとき $E_A = E_B = 0.75$ であるため，E_T は7割程度の並列度が得られる．(A) と (B) の並列度，処理時間のバランスによって R の最適値は変わるが，2〜4が最適と考える．本論文では $R=4$ として以下のような工夫をした．(A) の部分では d が何ビット右シフト可能かを見て，そのビット数だけ一挙に右シフトを行うようにして E_A を向上させた．このようにして実際の逆数計算ではプロセッサ数が16,384個程度の場合でも70%以上の並列化効率が得られていると考える．このように，逆数計算の並列化効率は乗算・剰余算よりも低く，全体としての並列化効率も低下する．しかし，逆数計算に代わる乗算・剰余算よりも処理時間が短くなるため，並列化効率の低下にもかかわらず，逆数計算を採用した．

MIMD 型並列計算機においては，分岐による並列度の低下はないので，逆数計算の並列化効率は SIMD 型に比べて高くなる．しかし，実行時に同期をどの場所で取るかなど，MIMD 型に特有の問題も増えてくると考えられる．

4.5 最大公約数（GCD）計算

3.4 節の逆数計算において b と N の最大公約数を求めることができるため，これを利用する（ただし変数 a，c は不要）．しかし逆数計算自身時間がかかるのでプログラム中で行う回数を減らしたり，多くの数

$b_i(i=1,\ldots,n)$ に適当するときは，素因数分解の観点から $b=\Pi_i b_i$ として b と N の最大公約数を求める．

4.6 全体の並列化効率

第1段階では乗算・剰余算が時間的に大部分を占める．これらの演算の並列化効率はほぼ100%であるため，第1段階の並列化効率は100%に近くなる．第2段階では，計算時間を短くするために48個の乗算・剰余算を逆数計算に置き換えた．SIMD型並列計算機では逆数計算は並列度が低く，第2段階では逆数計算が1/2弱の時間を占めるため，第2段階での並列化効率が低下する．第1段階，第2段階の処理時間比を1:1とすれば，逆数計算が占める割合は1/4以下になることから，4.4節で述べたようにプロセッサ数が16,384個程度のときの逆数計算の並列化効率を70%とし，乗算，剰余算の並列化効率を100%，逆数計算の割合を1/4に仮定したとしても，楕円曲線法における全体の並列化効率 S は

$$S=\frac{0.7+3}{4}\times 100=92.5[\%]$$

となり，少なくとも90%以上に達していると考えられる．

5. 成　　果

数値実験にはSIMD型並列計算機MasPar MP-1[4]を用いた．楕円曲線法のプログラムは多倍長演算部を含めてすべてMPL[18),19)]で作成した．1語を32ビット整数とし，このうち上位数ビットをキャリービットに割り当てた．楕円曲線法では，乗算や剰余計算が大量に出てくるため，これらの演算をいかに高速にするかが重要になる．実際のプログラムでは，剰余計算におけるテーブル参照方式[17)]に改良を加えてさらなる高速化を目指した．

カニンガム・プロジェクト[10)]において，楕円曲線法により近年発見された素因数は30桁前後が主流になっている．そこで $p\approx 10^{30}$ の素因数 p を見つける戦略を検討する．式(18)の T を最小にする L_1 は 2.8×10^5 であり，成功率 $\mu(\alpha,\beta)$ は 1.1×10^{-3} となっている．この値は1/1024より大きいため，1,024個を持つMasPar MP-1で十分であると考えられる．

そこでプロセッサ数1,024個の並列計算機上で，$L_1=2\times 10^5$ とし，$L_2=50L_1=1\times 10^7$ と戦略を決めた．すると $p\approx 10^{30}$ 程度の素因数発見の成功率 σ は1/2以上になる．この並列計算機上で同じ L_1, L_2 まで計算を進めると $p\approx 10^{28}$ の素因数が発見できる確率が9割近くになり，28桁以下の素因数は大部分発見できると期待される．さらに $p\approx 10^{34}$ の素因数が発見できる確率も1割程度あり，多くの合成数を試せば35桁以上の素因数を発見する可能性もある．

1,024個のプロセッサを搭載したMasPar MP-1で上述の戦略を用いた場合，プログラムの実行時間は92桁の合成数に対して，第1段階を $L_1=2\times 10^5$ まで行うのに19時間弱，第2段階を $L_2=1\times 10^7$ まで行うのに17時間弱である．この程度の計算時間であるために数多くの合成数の分解に挑戦できる．

素因数分解はまず円分数を対象とした．円分多項式 $\Phi_n(x)$ は，次の因数分解によって，帰納的に定義される．

$$\Phi_1(x)=x-1,\quad x^n-1=\prod_{d|n}\Phi_d(x)$$

ここで $\prod_{d|n}$ は自然数 n の約数 d にわたっての積を表す．n と x を自然数としたときの円分多項式 $\Phi_n(x)$ の値を円分数という．森本ら[12)~14)]の円分数分解プロジェクトにおいて，分解されていない合成数は最大90桁で，楕円曲線法の実験を行うには手ごろな大きさである．1992年6月の時点で未分解の78～90桁の合成数について上述の戦略（$L_1=2\times 10^5$, $L_2=1\times 10^7$）で楕円曲線法を実行した．その結果多くの合成数の中から今まで未発見の素因数が見つかった．その素因数のうち36桁以上のものを**表2**に示す．

素因数が見つかった曲線（パラメータ）と位数 S も表2に示す．なお B については初期値を x_1 とするとき，$x_1^3+Ax_1^2+x_1$ が見つかった素因数 p に関して平方剰余となるかどうかで位数が決まるようであり，これをLegendreの記号 $\left(\frac{B}{p}\right)$ で表す．今回の実験では円分数$\Phi_{31}(836)$に最大41桁の素因数が見つかった．ランダム整数の素因数の理論によると $L_1=2\times 10^5$, $L_2=1\times 10^7$ としたとき1,024個のPEを有する並列計算機で $p\approx 10^{40}$ の素因数が見つかる確率は0.4%しかなく，41桁の素因数の発見は貴重なものであるといえる．カニンガム・プロジェクトにおいて現在までに楕円曲線法で発見されている大きな素因数は $10^{201}-1$ と $2^{603}-1$ の42桁，$3^{415}-1$ の39桁となっている．楕円曲線法では素因数の大きさに応じた計算量が必要なことから，SIMD型超並列計算機MasPar MP-1上で楕円曲線法の有効性が実証された．

円分数のほかに，いくつかのカニンガム数に対しても楕円曲線法を試みた．現在のところ100桁以下の合成数はほとんど分解が済んでいる．100～120桁の合成数に対しても残りの素因数の桁数が大きく，また近

表2 楕円曲線法によって発見された大きな素因数
Table 2 Large prime factors discovered by ECM.

合成数	素因数・パラメータ・位数
$\Phi_{31}(836)$	$p = 267276413439148721576506359276626205065 89$ $u = 697$, $\left(\frac{B}{p}\right) = +1$ $S = 2^5 \cdot 3^3 \cdot 5 \cdot 11 \cdot 19 \cdot 29 \cdot 653 \cdot 12589 \cdot 14551 \cdot 72937 \cdot 143281 \cdot 158443 \cdot 5153779$
$\Phi_{62}(646)$	$p = 190884901781586671955172825357167832709$ $u = 723$, $\left(\frac{B}{p}\right) = +1$ $S = 2^3 \cdot 3^2 \cdot 7 \cdot 17 \cdot 19 \cdot 23 \cdot 37 \cdot 151 \cdot 359 \cdot 13033 \cdot 49499 \cdot 63559 \cdot 109891 \cdot 5641019$
$\Phi_{62}(881)$	$p = 10818526999467303902301548896494300121$ $u = 31$, $\left(\frac{B}{p}\right) = -1$ $S = 2^2 \cdot 3^3 \cdot 5^2 \cdot 13 \cdot 37 \cdot 89 \cdot 577 \cdot 677 \cdot 719 \cdot 4337 \cdot 4513 \cdot 7121 \cdot 16573 \cdot 144271$
$\Phi_{62}(954)$	$p = 830329191455300071146897898641915547$ $u = 529$, $\left(\frac{B}{p}\right) = -1$ $S = 2^2 \cdot 3^3 \cdot 5 \cdot 359 \cdot 3461 \cdot 4373 \cdot 8863 \cdot 13291 \cdot 30983 \cdot 118147 \cdot 656291$

いうちに複数多項式二次ふるい法[7),8)]で分解される可能性がある．さらに数体ふるい法[9)]が適用された合成数も150桁程度にまで及んでいる．そこで楕円曲線法でしか分解が困難と思われる160桁以上の合成数を分解の対象とした．その結果 $2^{2022}+1$ の中に含まれている180桁の合成数に25桁，$2^{995}-1$ に含まれている185桁の合成数に29桁の素因数が発見された．これらの成果は文献10)のNews LetterのPage 67とPage 68にそれぞれ記載された．楕円曲線法の計算時間は乗算・剰余算の計算アルゴリズムに支配され，本プログラムでは合成数の桁数の2乗弱に比例する．よって同程度の大きさに素因数を発見するにしても，180桁程度のカニンガム数の分解には，90桁程度の円分数の分解よりも4倍弱の手間がかかる．カニンガム・プロジェクトでは分解が完了していない合成数の桁数が大きく，未発見の素因数の桁数も大きくなっている．そのためカニンガム・プロジェクトで成果を出すことが困難になっている．それだけにこのプロジェクトに貢献できたことは意義あるものと考える．

6. 結　　論

SIMD型超並列計算機MasPar MP-1を用いて楕円曲線法による素因数分解を試みた．楕円曲線法はSIMD型超並列計算機に適した方法であり，この種の計算機の性能を十分に活かせる．楕円曲線法の計算量を見積もり，30桁程度の素因数発見を目標にして，いくつかの円分数やカニンガム数の分解に挑戦した．円分数の素因数分解では最高41桁の素因数が発見され，楕円曲線法における貴重な分解の実例と考える．また素因数分解の分野で最高の水準にあるカニンガム・プロジェクトにも成果を出すことができた．これらの成果から，SIMD型超並列計算機が大きな整数を素因数分解するのに非常に適していることが示された．素因数分解は計算機のソフトウェアとハードウェアの両方を駆使して初めて成果が出せるものである．従って素因数分解がどこまで進行しているかは，その時点における計算機科学の進歩を表す尺度の一つと考える．

参 考 文 献

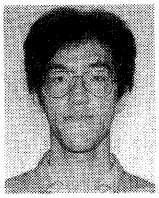
1) Rivest, R.L., Shamir, A. and Adelman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryposystems, *Comm. ACM*, Vol.21, pp.120–126 (1978).
2) Lenstra, H.W., Jr.: Elliptic Curve Factorization and Primality Testing, *Proc. of Computational Number Theory Conference* (Aug. 1985).
3) Montgomery, P.L.: Speeding the Pollard and Elliptic Curve Methods of Factorization, *Math. Comp.*, Vol.48, pp.243–264 (Jan. 1987).
4) MasPar Computer Corporation: MasPar System Overview, PN:9300-0100-2790 (July 1990).
5) Joly, J.R.: Equations et Variétés Algebriques sur un Cops Fini, *L'Enseignement Mathématique*, Vol.19, pp.1–117 (1973).
6) Tate, J.T.: The Arithmetic of Elliptic Curves, *Invent. Math.*, Vol.23, pp.179–206 (1974).
7) Pomerance, C.: *The Quadratic Sieve*, Lecture Notes in Computer Science 209, pp.169–182 (1985).
8) Silverman, R.D.: The Multiple Polynomial Quadratic Sieve, *Math. Comp.*, Vol.48, pp.329–339 (1987).
9) Lenstra, A.K., Lenstra, H.W., Jr., Manasse, M.S. and Pollard, J.M.: The Number Field Sieve, *Proc. 22nd STOC*, pp.564–572 (1990).
10) Brillhart, J., Lehmer, D.H., Selfridge, J.L., Tuckerman, B. and Wagstaff, S.S., Jr.: Factorizations of $b^n \pm 1$, $b = 2, 3, 5, 6, 7, 10, 11, 12$ up to High Powers, *Contemporary Mathematics*, Vol.22, second edition, American Mathematical Society, Providence, Rhode Island (1988).

11) Dixon, B. and Lenstra, A.K.: Massively Parallel Elliptic Curve Factoring, *Eurocrypt '92. Workshop on the Theory and Applications of Cryptographic Techniques. Proceedings*, pp.183–193 (1992).
12) 森本光生，木田祐司：円分数の素因数分解，上智大学数学講究録 26 (1987).
13) 森本光生，木田祐司，斎藤美千代：円分数の素因数分解（その 2），上智大学数学講究録 29 (1989).
14) 森本光生，木田祐司，小林美千代：円分数の素因数分解（その 3），上智大学数学講究録 35 (1992).
15) Knuth, D.: *The Art of Computer Programming*, 2 (seminumerical algorithms), Addison-Wesley (1981).
16) Brent, R.P.: Some Integer Factorization Algorithms Using Elliptic Curves, Technical report, CMA-R32-85 (Sep. 1985).
17) 伊東利哉，佐古和恵：有限体上のアルゴリズムと多倍長・剰余演算の高速演算法，情報処理，Vol.34, No.2, pp.170–179 (1993).
18) MasPar Computer Corporation: MasPar Parallel Application Language (MPL) User Guide, PN:9302-0100-2790 (July 1990).
19) MasPar Computer Corporation: MasPar Parallel Application Language (MPL) Reference Manual, PN:9302-0100-2790 (July 1990).

(平成 7 年 1 月 10 日受付)
(平成 7 年 9 月 6 日採録)

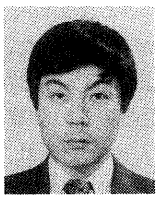
高橋 大介（学生会員）

1970 年生．1991 年呉工業高等専門学校電気工学科卒業．1993 年豊橋技術科学大学工学部情報工学課程卒業．1995 年同大学院工学研究科情報工学専攻修士課程修了．現在，東京大学大学院理学系研究科情報科学専攻博士課程在学中．並列アルゴリズムの研究に従事．



鳥居 泰伸

1964 年生．1985 年岐阜工業高等専門学校電気工学科卒業．1987 年豊橋技術科学大学工学部電気・電子工学課程卒業．1993 年同大学院工学研究科総合エネルギー工学専攻博士課程修了．博士（工学）．現在富士通（株）勤務．数値計算アルゴリズムに興味を持っている．



湯淺 太一（正会員）

1952 年神戸生．1977 年京都大学理学部卒業．1982 年同大学理学研究科博士課程修了．同年京都大学数理解析研究所助手．1987 年豊橋技術科学大学講師．現在，同大学教授．理学博士．記号処理と超並列計算に興味を持っている．著書「Common Lisp 入門（共著）」ほか．ソフトウェア科学会，電子情報通信学会，IEEE，ACM 各会員．