数 値 解 析 37−11
（1991. 7. 17）

帰納的に定義された自然数列の計算の超並列化

栗野俊一*・廣瀬健*・坂倉正純**・深澤良彰*

*早稲田大学理工学部
**早稲田大学情報科学研究教育センター

並列計算機システムで、問題を高速に解くには、その与えられた問題を並列化する必要がある。しかし、一般には、問題に含まれる順序性がこの並列化を阻害し、思うように高速化できないことが多い。帰納的に定義された数列の第n項を求める問題は、このような順序性をもつ典型的な問題の一つである。

本論文では、まずはじめに、この問題に対して制限を加え、その制限を利用して問題を並列化する手法を述べる。ところが、この手法は大量の計算機を必要とする。そこで次に、この制限を緩和することを目的とした、計算機台数と時間の交換手段、および、計算機台数と領域の交換手段をそれぞれ示す。

Super Parallelizing for Computation of Inductively Defined Natural Sequence

Syun'ichi Kurino*, Ken Hirose*, Masayoshi Sakakura**, Yoshiaki Fukazawa*

*School of Science and Engineering, **Centre for Informatics,
Waseda University, 3-4-1, Okubo Shinjuku-ku, Tokyo 169, Japan

In order to solve a problem speedy by a parallel computing system, it is necessary to parallelize the problem. But in many cases, internal sequentiality of the problem prevents from recognizing the parallelization. The problem to compute N-th term of a given natural sequence, which is inductively defined, is one of the typical problems in this category.

In this paper, at first, we describe a method to parallelize these problems under certain restriction. This method, however, requires a huge number of computers. Therefore, secondly, we show trade-offs, in which the number of computers is exchangable for its execution time, or for its necessary memory space.

第１章　はじめに

与えられた問題を高速に処理するための方法の一つに、多数の計算機を用いて、並列に計算を進めることがある。この方法を適用するには、与えられた問題から並列性を抽出し、それを複数の計算機に割り当てるといった作業が必要である。これまでにも、特定な問題から並列性を巧く抽出して実行するための並列アルゴリズムの研究[1]や、並列化されたアルゴリズムを効率良く実行するための計算機アーキテクチァの研究[2, 3, 4]などが行なわれてきた。これらの研究では、通常、明らかに並列性を含むような問題を対象とし、その並列性をいかに自然に表現し、効率を失わず、並列計算機に実現するかを議論の対象としている。しかし、現実には、多くの問題が順序性を内包しており、これらの方法が適用できず、その結果として、思うように並列化できないことが多い。例えば、帰納的に定義された数列の第ｎ項を計算する問題は、この様な順序性を内包する問題の一つである。

本論文では、はじめに、この順序性を持つ問題を並列化するための手法を述べる。ただし、この手法を問題に直接、適用すると、非現実的なほど大量の計算機が必要になる。そこで、必要とする計算機の台数を減らすための幾つかの手法を示す。そして、これらの方法の評価するために、必要な資源の量に関わる計算量を導入し、実際にそれらの比較を行なう。

第２章　並列化定理

２．１　本手法で扱う問題

本手法で扱う問題を次に示す。これは通常、計算を順次に行なう典型的な問題である。

問題Ｐ

次の様に、関数ϕを用いて、帰納的に定義された自然数列$\{\alpha_i\}$に対して、α_nを計算する。

$$\alpha_0 = c \quad (c \in \mathbf{N})$$
$$\alpha_i = \phi\ (\alpha_{i-1},\ i)$$

この数列の定義によって与えられる条件だけでは、十分な並列化を行なうことはできない。そこで、この問題に対して、さらに、次のような仮定をおく。この仮定によって、本手法の適用よる並列化が可能になる。

予測可能条件

数列$\{\alpha_i\}$に対して

$$\forall\, i \leqq n\ [u \leqq \alpha_i \leqq v]$$

となるｕ，ｖ$\in$Ｎがあらかじめ知られているとき、数列$\{\alpha_i\}$は、予測可能条件を満たすという。このとき、数列の取り得る値の範囲を意味するｖ－ｕ＋１をこの数列の幅と呼び、Ｌで表わす。

この制限は、かなり強い条件であり、与えられた問題によっては、この数列の幅をあらかじめ求めることが困難な場合がある。しかし、一般的な問題でも、あらかじめ、その答の値の取り得る範囲がわかっていることは多い。また、この制限は、精密な上限と下限を要求しているのではなく、単に粗い精度での上界と下界を要求しているだけである。したがって、予測可能条件を満たす問題のみを考えても、応用上、その一般性を失うことはないと思われる。そこで、本論文で扱う問題Ｐは、常に、この予測可能条件を満たすものとする。

２．２　並列アルゴリズム

前述の問題Ｐを分析する。一般に、帰納的に定義された数列の第ｎ項を求めるためには、第１項から第ｎ－１項も求める必要がある。ここで、すべてのｉに対して、第ｉ＋１項の値を計算するために第ｉ項の値が必要であれば、第ｎ項の計算には、ｎに比例した時間が必要となる。ところが、問題Ｐの数列$\{\alpha_i\}$は、離散的な要素を持つ自然数列であり、しかも、予測可能条件が成立するため、第ｉ項の値が与えられていなくても、第ｉ＋１項の値を有限個の候補の中から予測することができる。さらに、第ｎ項を求めるという目標が固定されているので、第１項から第ｎ項までの値をそれぞれ、一つ前の項の値を用いずに予測を行ない、同様に有限個の数列候補を予測することができる。その後、この予測が正しいかどうかの判定ができれば、第ｎ項を求めれることができる。以上から、問題Ｐを解くには、次のような二つの条件を満たす数のｎ重順序対、すなわち、順序を持ったｎ組を求めればよいことがわかる。

① $<\alpha_1, \alpha_2, \cdots, \alpha_{n-1}, \alpha_n> \in L^n$

② $\forall\, i\ (1 \leqq i \leqq n)\ [\alpha_i = \phi\ (\alpha_{i-1},\ i)]$
（ただし、$\alpha_0 = c$）

ここで、①の条件を満たすｎ組の生成も、②の条件の検査も、いずれも並列に行なうことができる。このために、ｎに比例した時間より、短い時間で第ｎ項の値が得られる可能性が生じる。以上の考察から、次のようなアルゴリズムを考える。

問題Ｐを解く並列アルゴリズム

与えられたｕ、ｖを用いて、次のような手順でα_nの値を計算する。

ステップⅠ）予測 $\{\beta_i^k\}$ の生成

$\forall i, k\ [u \leqq \beta_i^k \leqq v]$

$(1 \leqq i \leqq n,\ 0 \leqq k < L^n)$

ステップⅡ）$\{\beta_i^k\}$ の予測の正しさの判定

$\forall i\ [\beta_i^k = \phi(\beta_{i-1}^k, i)]$

$(1 \leqq i \leqq n)$

ステップⅢ）正しい予想 $\{\beta_i^k\}$ の選択

$\alpha_n \leftarrow \beta_n^k$

このアルゴリズムを利用することによって、次のような結果が得られる。

定理1（並列化定理）

問題Pに対して予測可能条件が成立するならば、最大 nL^n の並列度で、計算を行なうことができる。

【証明】

上記に説明した、並列アルゴリズムを適用すれば良い。すると、各ステップでは次のような並列性が生じる。

ステップⅠ）各数列の予測 $\{\beta_i^k\}$ は、互に独立に、生成することができ、その並列度は L^n である。

ステップⅡ）個々の数列の予測の正しさの判定は、並列に行なうことができ、その並列度は n である。さらに、L^n 個の数列候補の判定も並列に行なえるので、全体の並列度は、nL^n になる。ただし、これは、等号 $\beta_i^k = \phi(\beta_{i-1}^k, i)$ の検査の回数であり、$\phi(\beta_{i-1}^k, i)$ の値の計算は、nL回だけでよい。

ステップⅢ）これは、正しい数列から、必要となる α_n の値を取り出すだけなので、定数時間になる。

以上から、求める結果が得られる。

第3章　計算量

並列化定理によって、問題Pを解くために並列計算機が利用可能であることはわかった。本章では、どのくらい高速化ができるかを議論する。このためには、この並列アルゴリズムを実行する計算機構の存在と、その実行効率の評価方法を定める必要がある。以下では、計算機構とその評価法を述べ、本手法の有効性を評価する。

3．1　本手法で用いる計算機構

本手法で用いる計算機構には次のような機能があることを仮定する。

定義1（並列計算システムS）

1）z個の汎用計算機を内蔵し、それらはそれぞれ局所メモリーを内蔵している。これらは、完全に、並列かつ独立に動作する。この部分は関数 ϕ を計算するために用いられる。

2）z個の汎用計算機間では通信は行なわない。その代り、計算の開始時に、z個の汎用計算機に対して、必要なデータやプログラムを分配したり、あるいは、計算の終了時に、z個の汎用計算機の計算結果を集計して、一つの結果を取り出すために、$\log_2 z$ に比例した時間を消費する。

このシステムSを利用することにより、問題Pを実際に解くことができる。ただし、このシステムに含まれる汎用計算機の数zは次のような条件を満たさなければならない。

リッチマン条件

問題Pによって与えられるnとLに対し、システムSのもつ汎用計算機台数zが nL^n 以上であるときシステムSはリッチマン条件を満たすという。

3．2　計算量の定義

前述のような計算機上のアルゴリズムの評価を行なうために、次のような計算の尺度を定義する。

深さ計算量

入力が与えられてから、計算結果が得られるまでに消費される時間を、深さ計算量と呼び、Λで表わす。

幅計算量

アルゴリズムを実行するのに一時的に必要な計算機の台数の最大値を、幅計算量と呼び、Vで表わす。

領域計算量

アルゴリズムを実行するのに、一時的に必要な領域の最大値を領域計算量と呼び、Mで表わす。

ここで、深さ計算量は、アルゴリズムを実行したときの最長ステップ数にあたり、幅計算量は、アルゴリズムの実行時における、最高の並列度にあたる。単一計算機システムで利用されるアルゴリズムでは、深さ計算量と時間計算量と同じでものである。そして、この時は、幅計算量は1になる。

3．3　問題Pの計算量

アルゴリズムの実行機構と、その評価基準が与えられたので、アルゴリズムの計算量を評価することができる。

定理2（システムSでの計算量）

予測可能条件を満たす問題Pをシステムで解く時、Sがリッチマン条件を満足するならば、次の計算量で、問題が解ける。

$$\Lambda[\![\alpha_i]\!] = \max(\Lambda[\![\phi_i]\!]) + n\Lambda[\![\mathrm{cons}]\!] + \log_2(nL^n)\Lambda[\![\mathrm{and}]\!]$$
$$V[\![\alpha_i]\!] = nL^n$$

ただし、$\Lambda[\![\mathrm{cons}]\!]$と$\Lambda[\![\mathrm{and}]\!]$は、それぞれ、数列の候補を作る時に、i組からi＋1組に延長するために使われる演算 cons の深さ計算量と、二つの数列の候補から、正しい一つの候補を選択するために使われる論理演算 and の深さ計算量である。

【証明】

並列アルゴリズムを実行する。その詳細は、以下の通りである。

①長さn，範囲$u \leq \beta_i \leq v$となる順列$\{\beta_i\}$の生成
②$\beta_i = \phi(\beta_{i-1}, i)$の検査
③n個の論理積の計算
④L^n個の真偽値の中から真であるものが存在すれば その内の任意の一個、無ければ偽を取り出す計算

これに対し、アルゴリズムの各ステップに消費される深さ計算量は次のようになる。

$$\Lambda[\![①]\!] = n\Lambda[\![\mathrm{cons}]\!]$$
$$\Lambda[\![②]\!] = \max(\Lambda[\![\phi_i]\!])$$
$$\Lambda[\![③]\!] = \log_2 n\,\Lambda[\![\mathrm{and}]\!]$$
$$\Lambda[\![④]\!] = \log_2 L^n \Lambda[\![\mathrm{and}]\!]$$

したがって、これらの総和を計算すれば、求める結果を得ることができる。

この結果から、このアルゴリズムの深さ計算量には関数ϕの計算（以下、汎用計算と呼ぶ）に消費した計算量と、通信（以下、通信計算と呼ぶ）に消費した計算量という二つの要因があることがわかる。つまりこのアルゴリズムでは汎用計算量を減らすために、通信計算量を増大させている。したがって、汎用計算量に比べ、通信計算量が無視できるほど小さい時、つまり、次の交換レート条件が成立する時、この手法は有効である。

交換レート条件

通信に費やす二つの計算量$n\Lambda[\![\mathrm{cons}]\!]$、$\log_2(nL^n)\Lambda[\![\mathrm{and}]\!]$がいずれも汎用の計算$\Lambda[\![\phi_i]\!]$に比べ、十分に小さいとき、交換レート条件が成立するという。

これは、列の生成や、論理積を求めるなどのような通信に費やす特定な演算が一般的な関数ϕの計算に比べ、高速に実行し得るという仮定である。このことは、これらの演算を行なう専用のハードウェアを設計・実装することによって実現する。そして、議論を単純化するために、この様な通信のための演算の深さ計算量は常に1と考えることにする。

これらの結果を総合すると、次のような結果が得られる。

系3（従来のアルゴリズムとの比較）

予測可能条件を満たす問題Pをシステムsで解くとする。この時、システムSがリッチマン条件と交換レート条件を満足するならば、並列アルゴリズムを用いることにより、逐次的なアルゴリズムに比べ、深さ計算量が1／nになる。

【証明】

通常の逐次アルゴリズムで計算すると、その深さ計算量は、次のようになる。

$$\Lambda[\![\alpha_i]\!] = \Sigma\Lambda[\![\phi_i]\!] \quad \cdots\cdots(1)$$

これに対し、並列アルゴリズムでの深さ計算量は、次のようになる。

$$\Lambda[\![\alpha_i]\!] = \max(\Lambda[\![\phi_i]\!]) + n\Lambda[\![\mathrm{cons}]\!] + \log_2(nL^n)\Lambda[\![\mathrm{and}]\!]$$

ここで、システムSは交換レート条件を満たすので、ここに現れる3つの項のうち、後の2つは、最初の項に比べて無視できるほど小さい。したがって、

$$\Lambda[\![\alpha_i]\!] \fallingdotseq \max(\Lambda[\![\phi_i]\!]) \quad \cdots\cdots(2)$$

と考えることができる。(1)と(2)を比較すれば、本アルゴリズムが、1／nの最適化になっていることがわかる。

3．4　適用対象

前節の手法は、その制限や結果から、適用対象として相応しい問題と、むしろ、通常の逐次的なアルゴリズムの方が良い問題とに分けられる。

この手法を適用することが相応しい問題の性質は、次の2つである。

・Lがnに比べ、急速に増大しない。
（$L << C \times 2^n$：Cは定数）
・$\Lambda[\![\phi_i]\!]$は全てのiに対し、大きくかつ同程度である。

第4章　交換定理

前章で紹介した手法は、問題Pを並列に計算し、深さ計算量の点で、改良になっている。しかし、この方

法が必要としている仮定で、応用上最も重要な条件は、リッチマン条件である。これは、現時点では成立し難い。この章では、深さ計算量や領域計算量などを増やすことによって、幅計算量を減少させるための手法を示す。

4．1　幅計算量と深さ計算量の交換

次の定理は、与えられた数列を幾つかの部分列に分解して計算する方法である。

定理4（交換定理Ⅰ；{部分列化定理}）

並列化定理と同じ仮定が成立し、かつ$n=m^2$が成立つならば、次の表1にあるように、幅計算量と深さ計算量の交換が行なえる。

表1　交換定理Ⅰの結果

	幅計算量（V）	深さ計算量（Λ）
逐次	1	nc
手法A	mL^m	$m(c+m\log_2 L)$
手法B	$m(m+1)L^m$	$c+2m\log_2 L$
並列	nL^n	$c+n\log_2 L$

（ただし、$c=\max(\Lambda[\![\phi_i]\!])$）

【証明】

数列$\{\alpha_i\}$を次のように、それぞれの長さがm＋1であるm個の部分列$\{\beta_{j,k}\}$に分割する。

$$\{\alpha_i\}_{i=0..n}=\{\beta_{j,k}\}_{j=0..m,\,k=0..m}$$
$$\beta_{j,k}=\alpha_i \Leftrightarrow i=mj+k$$
$$\beta_{j,m}=\beta_{j+1,0}$$

そして、関数ϕ'を次の式で定義する。

$$\beta_{j,k}=\phi(\beta_{j,k-1},\ mj+k)$$
$$\beta_{j,0}=\phi'(\beta_{j-1,0},\ j)$$

この$\beta_{j,0}$に関する計算量は、並列化定理より、

$$\Lambda[\![\beta_{j,0}]\!]=\Lambda[\![\phi']\!]$$
$$=\max(\Lambda[\![\phi]\!])+m\log_2 L$$
$$V[\![\beta_{j,0}]\!]=mL^m$$

である。ここで、数列$\{\beta_{j,k}\}$の計算法として、2種類の方法、手法Aと手法Bを考える。

手法Aでは、数列$\{\beta_{j,k}\}$の値を、逐次的に計算する。この時、必要となる計算量は以下の通りである。

$$\Lambda[\![\alpha_n]\!]=m\Lambda[\![\phi']\!]$$
$$=m(\max(\Lambda[\![\phi]\!])+m\log_2 L)$$
$$V[\![\alpha_n]\!]=\max(V[\![\beta_{j,0}]\!])$$
$$=V[\![\beta_{j,0}]\!]$$
$$=mL^m$$

一方、手法Bは、数列$\{\beta_{j,k}\}$の値もまた、並列に計算するこの時、必要となる計算量は以下の通りである。

$$\Lambda[\![\alpha_n]\!]=\max(\Lambda[\![\phi']\!])+m\log_2 L$$
$$=\max(\max(\Lambda[\![\phi]\!])+m\log_2 L)+m\log_2 L$$
$$=\max(\Lambda[\![\phi]\!])+2m\log_2 L$$
$$V[\![\alpha_n]\!]=\Sigma(V[\![\beta_{j,0}]\!])$$
$$=mV[\![\beta_{j,0}]\!]+V[\![\beta_{j,0}]\!]$$
$$=(m+1)mL^m$$

以上により、表1の結果が得られる。

この定理は一般的に$n=m^k$に拡張可能である。したがって、このように並列度の低い問題では、深さ計算量Λと幅計算量Vの関係は、次のようになっている。ただし、C_1，C_2は問題によって定まる定数である。

$$C_1<\Lambda+\log_2 V<C_2$$

4．2　幅計算量と領域計算量の交換

次の定理は、数列の計算をラベル付きのグラフの経路探索問題[8]に還元して解く手法である。つまり、u以上v以下の数α，βを、節点と見なし、$\alpha=\phi(\beta,\ i)$が成立するときに、αとβの間にラベルiの付いた弧があると見なせばよい。すると、問題Pは、α_0を始点とし、i番目に、ラベルiの付いた弧を通り、n番目に到達する経路を求める問題と同値である。そして、この経路を隣接行列[8]を用いて求める。

定理5（交換定理Ⅱ；{ネットワーク化定理}）

仮定を並列化定理と同じとする。このとき、次のように幅計算量と領域計算量が交換できる。

$$\Lambda[\![\alpha_n]\!]=\max(\Lambda[\![\phi_i]\!])+(\log_2 n)(\log_2 L)$$
$$V_{GC}[\![\alpha_n]\!]=nL$$
$$V_{CM}[\![\alpha_n]\!]=nL^3$$
$$M[\![\alpha_n]\!]=nL^2$$

ただし、V_{GC}は汎用計算を行なうための計算機の台数であり、V_{CM}は隣接行列の計算、つまり通信計算を行なうための専用演算器の台数を表わす。また、領域計算量Mの単位はビットである。

【証明】

次のような順に計算を行なえば良い。ただし、説明

を簡単にするために$n=2^u$であるとする。

I）$L\times L$の隣接行列Δ_k（$0<k<n+1$）

$$\Delta_k=\{\delta_{\langle i,j\rangle k}\}$$

（$0<i<L+1$，$0<j<L+1$，$0<k<n+1$）

を計算する。ただしその要素は、全ての$\langle i,j\rangle_k$に対して、

$\delta_{\langle i,j\rangle k}\leftrightarrow j+u-1=\phi(i+u-1,k)$

と定義する。このΔ_kの要素を計算するための計算量は、次のようになる。

$$\Lambda([\![\mathrm{I}]\!])=\max(\Lambda([\![\phi_i]\!]))$$
$$V_{GC}([\![\mathrm{I}]\!])=nL$$

（$\because\delta_{\langle i,j\rangle k}$が真となる場合だけを考える）

$$M([\![\mathrm{I}]\!])=nL^2$$

II）$\prod_{i=1}^{n}\Delta_i$を計算する。

次のような隣接行列$\Delta_{j,k}$（$0\leqq k\leqq u$，$0\leqq j<2^k$）を定義する。

$$\Delta_{j,0}=\Delta_j$$
$$\Delta_{j,k}=\Delta_{2j-1,k-1}\cdot\Delta_{2j,k-1}$$

この時、次のような関係が成立する。

$$\Delta_{0,u}=\prod_{i=1}^{n}\Delta_i$$

$\Delta_{j,k}$の計算を並列に行なうと、$L\times L$の隣接行列の掛け算$\Delta\cdot\Delta$をu回行なうことになる。また、その時の並列度は最大$n/2$である。したがって、$\Delta_{0,u}$の計算量は次のようになる。

$$\Lambda([\![\Delta_{0,u}]\!])=u\Lambda([\![\Delta\cdot\Delta]\!])$$
$$V_{CM}([\![\Delta_{0,u}]\!])=nV([\![\Delta\cdot\Delta]\!])$$

一方、$\Delta\cdot\Delta$を計算するには、L^2個の要素δ_{pq}（$0\leqq p<L$，$0\leqq q<L$）を計算しなければならない。これらの計算は全て並列に行なえ、その並列度はL^2である。

$$\Lambda([\![\Delta\cdot\Delta]\!])=\Lambda([\![\delta_{pq}]\!])$$
$$V_{CM}([\![\Delta\cdot\Delta]\!])=L^2V_{CM}([\![\delta_{pq}]\!])$$

この要素δ_{pq}の計算は、L次元ベクトルの内積を求めることと同じである。この内積は、二つの真偽値の論理積をL組計算し、次に、それらのL個の結果の論理和を求めることになる。これらは、それぞれ並列に行なえ、その並列度はLとL／2である。したがって、その計算量は、次のようになる。ただし、$\Lambda([\![\mathrm{or}]\!])$は、論理演算 or を行なうために必要な深さ計算量である。

$$\Lambda([\![\delta_{pq}]\!])=\Lambda([\![\mathrm{and}]\!])+\log_2L\,\Lambda([\![\mathrm{or}]\!])$$
$$V_{CM}([\![\delta_{pq}]\!])=L$$

以上の結果をまとめると、次のような結果が得られる。なお、領域計算量は、n個のL次の正方隣接行列を記録するために消費する量である。

$$\Lambda([\![\mathrm{II}]\!])=\log_2n(\Lambda([\![\mathrm{and}]\!])+\log_2L\,\Lambda([\![\mathrm{or}]\!]))$$
$$V_{CM}([\![\mathrm{II}]\!])=nL^3$$
$$M([\![\mathrm{II}]\!])=nL^2$$

IとIIの結果から、目的とする結果が得られる。

4．3　分割統治法を用いた計算量の交換

この節では、再び幅計算量と深さ計算量の交換について述べる。並列化定理では、幅計算量がnL^nであった。交換定理Iでは、この値を減少させるために、指数部のnの減少を目的として、交換を行なった。さらに、交換定理IIでは、同じnに対し、領域までも考慮して交換を行なった。その結果、指数部分の影響は、解消された。そこで、この節では、底数部Lの減少を試みる。

次の交換定理は、数列$\{\alpha_i\}$を直接計算するのではなく、数列$\{\alpha_i\}$より、幅の小さな数列（以下、副数列と呼ぶ。）を幾つか計算し、その計算結果を用いてα_nを計算する手法である。この副数列に対して、前節の交換定理IIを用いると、幅の部分が減少し、結果として、定数部Lが減る。

さて、副数列の構成法は幾つか考えることができるが、ここでは、剰余類を用いることにする。具体的には次のような手順で、副数列を構成するとともに、副数列の定義を導き出す。

・数列$\{\alpha_i\}$を特長付ける関数ϕが多項式で模倣できること。

・多項式は、mod 演算と可換であり、自然数上の関数が、剰余類上の関数になること。

・複数の剰余類上の数列が、数列$\{\alpha_i\}$を計算するための副数列として利用できること。

そして、最後に、これらを利用して、交換定理IIIを示す。

定理6（多項式による関数の模倣）

任意の自然数から自然数への写像fが与えられたとき、その定義域を有限な領域Dに制限すれば、ある有理数係数の1変数多項式gが存在し、D上で、gがf

を模倣するようにできる。

【証明】（概略）

Dの要素数をMとするとき、g（X）をM－1次元の多項式$\Sigma a_m X^m$とし、係数a_m（$0 \leqq m < M$）に関する次のようなM元連立方程式を解けば良い。

$$g(x) = f(x) \quad (x \in D)$$

系7（模倣多項式）

数列$\{\alpha_i\}$を特徴付ける関数$\phi(\alpha, i)$に対して、$0 \leqq \alpha < L$，$0 \leqq i \leqq n$ならば、ある有理数係数の1変数関数F（x）が存在して、次の式が成立つ。

$$F(Li+\alpha) = \phi(\alpha, i)$$

この関数Fを関数ϕあるいは、数列$\{\alpha_i\}$の模倣多項式と呼ぶ。また、関数Fの係数を通分したときの分母qを繰返し因子と呼ぶことにする。

この系を利用することによって、一般性を失うことなく、模倣多項式Fを関数ϕの代りに用いることができる。

次に、p_jを素数として、次のような射影関数π_jを考える。

$$\pi_j(x) = x \bmod p_j$$

さらに、数列$\{\alpha_i\}$をこの射影関数π_jで射影した数列$\{\beta^j{}_i\}$を考える。

$$\beta^j{}_i = \pi_j(\alpha_i) = \alpha_i \bmod p_j$$

すると、次のような性質が成立つ。

定理8（多項式と剰余計算の可換）

素数p_jを関数Fの繰返し因子qと互いに素な素数とする。すると、次の高々p_j-2次の整数係数の関数ρ_jが存在し、数列$\{\beta^j{}_i\}$を次のような再帰的な数列で定義できる。

$$\beta^j{}_0 = \pi_j(\alpha_0)$$
$$\beta^j{}_{i+1} = \rho_j(\beta^j{}_i, i+1)$$

【証明】（概略）

$$\beta^j{}_{i+1} = \pi_j(\alpha_{i+1}) = \pi_j(F(L(i+1)+\alpha_i))$$

ここで、Fは多項式であり、積と和はπ_jと交換できるので、Fとπ_jは交換ができる。しかも、繰返し因子qは、p_jと素なので、qの逆元が存在し、ρ_jの係数を全て整数にできる。

$$= F(\pi_j(L(i+1)+\alpha_i))$$
$$= \rho_j(\pi_j(\alpha_i), i+1)$$
$$= \rho_j(\beta^j{}_i, i+1)$$

また、pが素数であることを用いると、フェルマーの定理[7]より、$x \neq 0$の時$x^{p-1} \equiv 1 \bmod p$なので、p_j-1次より高次な項は消去できる。

この関数ρ_jを関数ϕのp_jに対する射影と呼ぶ。

定理9（百五減算定理[7]）

$L < p_1 p_2 \cdots p_k$となる互に異なる素数p_j（$1 \leqq j \leqq k$）があるとする。整数αが自然数Lより小さい自然数ならば、αとk重順序対$<\pi_1(\alpha), \pi_2(\alpha), \cdots, \pi_k(\alpha)>$は、1対1に対応する。

したがって、数列$\{\beta^j{}_1 = \pi_j(\alpha_i)\}$を取れば、その幅は$p_j$でLより小さく、各$\{\beta^j{}_i\}$は個別に計算できる。しかも、$\alpha_i$とk重順序対$<\beta_{11}, \beta_{21}, \cdots, \beta_{k1}>$は、1対1に対応するので副数列として、利用できる。

以上の結果をまとめると次の定理が成立する。

定理10（交換定理Ⅲ；{百五減算定理[5,6]}）

並列化定理と同じ条件で、しかも、$L < p_1 p_2 \cdots p_k$となる互に異なり、かつ数列$\{\alpha_i\}$の繰返し因子qの約数でない素数p_j（$1 \leqq j \leqq k$）があり、さらに、関数ϕの各p_jに対する射影ρ_jが与えられたとする。

この時、次のように、幅計算量と深さ計算量を交換することができる。

$$\Lambda[\![\alpha_n]\!] = c_1 + k c_2 + (\log_2 n)(\log_2(P))$$
$$V_{GC}[\![\alpha_n]\!] = nP$$
$$V_{CM}[\![\alpha_n]\!] = P^3$$
$$M[\![\alpha_n]\!] = nP^2$$

（ただし、$P = \Sigma p_j$、c_1，c_2は定数）

【証明】

まず、与えられた$\{\alpha_i\}$に対して、次のような副数列群$\{\beta^j{}_i\}$を作成し次の手順で、計算を行なう。

$$\beta^j{}_i \equiv \alpha_i \bmod p_j$$

①α_0から、$\beta^j{}_0$を求める
②各々の数列$\{\beta^j{}_i\}$を求める
③各々の$\beta^j{}_n$からα_nを求める

これらの計算量を調べると、次のようになる。ただし、$\Lambda[\![\text{mod}]\!]$，$\Lambda[\![\text{add}]\!]$，$\Lambda[\![\text{mul}]\!]$は剰余演算 mod、整数演算 add，mul'を計算するために必要な深さ計算量である。

$$\Lambda[\![①]\!] = k\,\Lambda[\![\mathrm{mod}]\!]$$
$$V_{GC}[\![①]\!] = k$$
$$\Lambda[\![②]\!] = \max(\Lambda[\![\rho_j]\!]) + (\log_2 n)(\log_2 p_j)$$
$$V_{GC}[\![②]\!] = n\,p_i$$
$$V_{CM}[\![②]\!] = p_i^{\,3}$$
$$M[\![②]\!] = n\,p_j^{\,2}$$
$$\Lambda[\![③]\!] = k\,\Lambda[\![\mathrm{mul}]\!] + \log_2 k\,\Lambda[\![\mathrm{add}]\!]$$
$$V_{GC}[\![③]\!] = k$$

これらを用いて、全体の計算結果を求めると次のようになる。深さ計算量は、ステップ①～③の深さ計算量の総和になるが、このうち、ステップ②を除いた計算は、kに比例するだけなので、その比例定数をc_2とおく。また、幅計算量は、ステップ①～③の幅計算量の最大値をとるが、やはり、ステップ②を除いた他の計算がkに比例するので、ステップ②の幅計算量となる。

$$\Lambda[\![\alpha_n]\!] = c_1 + k\,c_2 + (\log_2 n)(\log_2(P)) \quad (c_1 = \max(\Lambda[\![\rho_i]\!]))$$
$$(\because \max p_i \leqq P)$$
$$V_{GC}[\![\alpha_n]\!] = n\,P$$
$$V_{CM}[\![\alpha_n]\!] = P^3 \quad (\because \Sigma p_i^{\,3} \leqq (\Sigma p_i)^3 = P^3)$$
$$M[\![\alpha_n]\!] = n\,P^2 \quad (\because \Sigma p_i^{\,2} \leqq (\Sigma p_i)^2 = P^2)$$

第5章　終わりに

本論文では、ある問題Pに対してその振舞いが予測できるという仮定のもとに、幾つかの並列化アルゴリズムを示した。これらは全て、幅計算量と深さ計算量あるいは、幅計算量と領域計算量の交換の度合いを表わしている。本論文で得られた結果を表2に示す。

表2　各計算量の交換

V_{GC}	V_{CM}	Λ	M
1	同左	n·C	0
$m \cdot L^m$	同左	$m(C+m\cdot\log_2 L)$	0
$m(m+1)L^m$	同左	$C+2m\cdot\log_2 L$	0
nL	nL^3	$C+(\log_2 n)(\log_2 L)$	nL^2
nP	nP^3	$C_1+k\cdot C_2+(\log_2 n)(\log_2(P))$	nP^2

ただし、表中の記号の意味は以下の通りである。

n　第n項目の値を計算する。
L　数列の幅
C　$\max(\Lambda[\![\phi_i]\!])$
m　$n = m^2$を満たす整数
k　$L < p_1 p_2 \cdots p_k$となる素数の個数
P　Σp_i
C_1　$\max(\Lambda[\![\rho_i]\!])$
C_2　α_iと$\beta^j{}_i$の相互変換を行なうためのkに比例する深さ計算量

また、Mの欄の0は、途中の計算結果を再利用のために保存しておく必要がないことを示す。

これらの交換定理は、用意できる計算機の量を表わす幅計算量の制約を与えて、どの程度の加速を図ることができるか、あるいは、待つことのできる最長の時間を表わす深さ計算量の制約を与えて、どの程度の計算機を用意すれば充分であるかを示したことになる。

今後の課題としては、

・具体的な応用分野と応用例
・細かい定量的な評価
・並列度が明確な問題との関係
・本手法に適したハードウェアの設計
・定理中のnやLについての条件の緩和

などが挙げられる。

謝辞

本論文は、早稲田大学情報科学研究教育センター「大規模並列計算機に対する数学的モデルの構築とその応用」部会の研究成果の一つであり、この研究活動を支援して下さった方々に、心より感謝致します。

参考文献

[1] L.H.Jamieson et al. "The Characteristics of Parallel Algorithms" MIT Press(1987)
[2] 富田眞治 "並列計算機構成論" 昭晃堂(1986)
[3] 村岡洋一 "ＶＬＳＩコンピュータ・アーキテクチャ" 近代科学社(1988)
[4] ダニエル・ヒリス著，喜連川優著訳「コネクションマシン」パーソナルメディア(1990)
[5] 吉田光由，大矢真一 "塵劫記" 岩波書店(1977)
[6] 和田秀男 "数の世界" 岩波書店(1981)
[7] 足立恒雄 "類体論へ至る道" 日本評論社(1979)
[8] C.ベルジュ著，伊理他訳 "グラフの理論" サイエンス社 (1976)