

SIMD型遺伝アルゴリズムと専用プロセッサ要素の設計

井上 富夫, 稲富 祐介, 佐野 雅彦, 高橋 義造

徳島大学 工学部

E-mail: { tom, ina, sano, taka } @ is.tokushima-u.ac.jp

遺伝アルゴリズムを現実の問題に適用しようとする、膨大な個体数と計算量が必要になり、通常の計算機ではとうてい処理できない。遺伝アルゴリズムを真に効果的な手法とするためには計算速度を現在の数百倍から数千倍に画期的に高める方策が必要となる。本研究ではSIMD型遺伝アルゴリズムを提案し、このアルゴリズムを用いて交差と淘汰を高速に行う専用プロセッサ要素を設計し、大規模問題に適用できる遺伝アルゴリズム専用のSIMD型計算機の開発を目的とする。本論文では始めにSIMD型遺伝アルゴリズムを提案し、次にこのアルゴリズムを実現する計算機システムとそのプロセッサ要素のアーキテクチャについて述べ、探索問題への適用法について述べる。

SIMD Genetic Algorithm and the Design of a Processing Element Adapted to the Algorithm

Tomio Inoue, Yusuke Inatomi, Masahiko Sano, Yoshizo Takahashi

Faculty of Engineering, The University of Tokushima

When genetic algorithms (GAs) are applied to real-world problems, the necessary size of the population and processing time are too large and, as a consequence, implementation in conventional computers is not practical. To make GAs truly effective, it is necessary to develop computers with processing speed from several hundred to thousand times higher than those of current machines. The research reported in this paper aims at proposing and developing a customized SIMD machine for GA applications to large-scale problems, with customized processors which implement fast crossover and selection operations. In this paper, first a SIMD GA is proposed, then the architecture of the computer system and each processing element to implement it is described, and finally an application to a search problem is given.

1. まえがき

1960年代に開発された遺伝アルゴリズム[1, 2]は、最近の計算機の画期的な性能向上のお陰でいくつかの定型的な問題についての効果が確認されるようになった。しかしこのアルゴリズムを実際の問題に適用しようすると、大量の個体を取り扱うことが必要となり、膨大な計算量が必要となるので[3]、通常の計算機で行うのは非現実的であるとされる。並列計算機を使用す

るにしても、従来の逐次型プログラムをそのまま並列化するだけでは、せいぜい数十倍程度の時間短縮にしかならない[4, 5, 6]。遺伝的アルゴリズムを真に効果ある実用的手法とするには、計算速度を現在の数百倍から数千倍に画期的に高める方式の研究を行う必要がある。本研究ではSIMD型遺伝アルゴリズムを提案し、このアルゴリズムに専用的に用いる並列計算機を開発することを目指しているが、それに必要な個体の交差

と淘汰を専用に行うプロセサ要素について考察する。そしてその計算機を32ビットの最適ビット列探索問題に適用することについて考える。2節ではSIMD型遺伝アルゴリズムを提案し、3節ではシステム構成、4節ではプロセサ要素の構成、5節では命令セットについて、6節では探索問題の解法について述べ、7節にまとめを述べる。

2. SIMD型遺伝アルゴリズム

遺伝アルゴリズムは多数の個体の集合が、その中で交配、突然変異、淘汰を繰り返しながら進化し、最終的に最も優れた個体を生成するというモードを最適化問題に適用したもので、図1 (a)のように多数の候補解の集合に遺伝子演算を加えて最終的に最適解を求めるものである。このアルゴリズムの問題点は長時間の計算を行う必要があることで、これを短縮するために個体数を制限すると局所解に陥って解の信頼性が損なわれる。そこで並列計算機を利用して全体の個体を多数の部分集合に分割し、部分集合ごとに交配、淘汰を繰り返し、一定の周期で部分集合間で優秀な解を交換しながら計算を進める分散型遺伝アルゴリズム[7, 8]とよばれる方法が研究されている。しかし、この方法ではよい遺伝子が集合全体に伝搬するのが遅れ、収束が悪くなるのが指摘されている。

本研究では、個体の集合を分割せず、一つの集合の中で遺伝子演算と淘汰を高速に行うような新しい並列遺伝アルゴリズムを開発することを目指している。その一つとして図1 (b)に示すようなSIMD型遺伝アルゴリズムを考えた。このアルゴリズムでは、各個体は自律的に動作して一斉に交配、突然変異、淘汰を行うものとする。すなわち各個体は自己評価を行い、相手の評価の値にもとづき優れた個体をランダムに選び、交差演算を行い子孫を生成する。突然変異はやはりランダムに選ばれた個体自身に変化する。評価値により優秀な相手を選ぶので、優秀な個体が遺伝されていく。次にこのSIMD型遺伝アルゴリズムの処理手順を示す

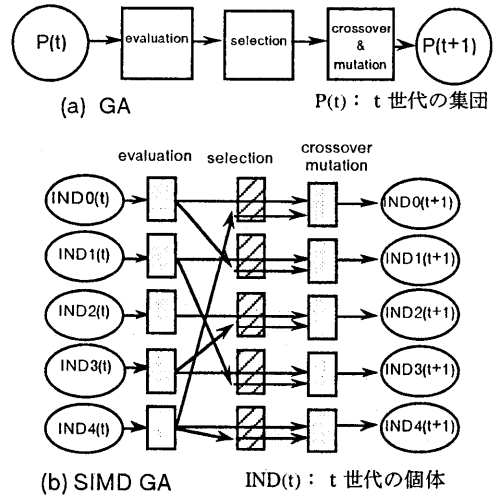


図1. 遺伝アルゴリズム (a) と SIMD型遺伝アルゴリズム (b)

```

SIMD_GA {
  初期個体の発生；
  自己評価；
  目標値が見つかるまでループ {
    良い相手を見つける(selection)；
    if (#) 子供を生む(crossover & mutation)；
    自己評価( evaluation)；
    世代交代；
  } /* ここで #は交差率, 突然変異率 */
}

```

3. 専用並列計算機のシステム構成

SIMD型遺伝アルゴリズムを実現するための専用並列計算機システムの構成を図2に示す。ホスト計算機はプログラム送出、実行状況および結果を表示する。SIMDコントローラはプロセサ要素PEに命令を送り出し、PEの動作を制御する。PEはSIMD型遺伝アルゴリズム専用プロセサで、SIMDコントローラから送られる命令を実行する。このシステムはSIMD型として動作させるので、全てのPEが同時に書き込みあるいは読み出しを行うため、Shared Memory Matrix (SMM)[4, 6]としてSWMR (Single Write Multiple Read) 型メモリを用い、遺伝アルゴリズムにおける全てのデータ、すなわち個体、評価値、適応度の記憶に用いる。SMMでは、i番目のPEはi番目の行のすべてのメモリセルに同じ

データを書き込み、i番目の列の任意のセルからデータを読むことができる。すべてのPEはSMMに対して同時に読み書きを行うことができる。

図3に一つのメモリセルの構成を示す。メモリセルは適用する問題から、個体記憶領域として32ビット、適応度の記憶領域として16ビットをもつ。1台のPEには2個の個体を割り当て、1セル当たり2組の個体を記憶する。SMM、SIMDコントローラ、およびPEはFPGA (Field Programmable Gate Array) を用いて製作するので、適用する問題に応じて個体数、記憶容量などの変更には柔軟に対応することができる。実際に8×8のSMMをデコーダ、入出力の制御回路などのメモリ関連回路と共に1個のLCA(XC4013-240)にインプリメントすることができた。内部のCLB (Control Block) は98%を使用した。

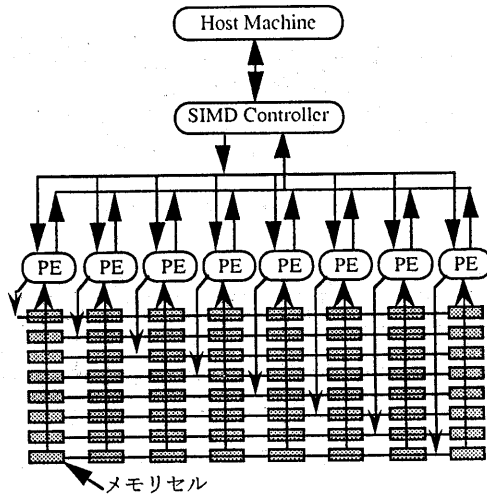


図2. 専用並列計算機のシステムの構成

遺伝子情報 (32ビット)	適応度 (16ビット)
遺伝子情報 (32ビット)	適応度 (16ビット)

図3. メモリセルの構成

4. PEのアーキテクチャ

PEに必要なGA専用機能として

1. 個体の評価値から適応度を求めること
2. 乱数の発生
3. 交差演算
4. 突然変異演算
5. ルーレットの作成
6. 淘汰演算
7. ソーティング

が挙げられる。

これらの機能を備えたPEのアーキテクチャを図4に示す。PEには個体情報記憶用に32ビットおよび16ビットのレジスタをそれぞれ2個、32ビットのワーキングレジスタが2個、5ビットのインデックスレジスタ、5ビットのアドレスレジスタがそれぞれ1個ある。また、32ビットの基本論理演算を行うALU、9ビットの全加算器、32ビットのシフター、評価回路、交差回路、絶えず新しい乱数を発生する乱数レジスタおよびSMMの接続のための入出力インターフェースを備えている。これらの演算回路、レジスタ、インターフェース間は32ビットバスにより接続し、SIMD命令によってスイッチングされ、データの流れを制御する。以下に回路の特徴について説明する。

ALU: 32ビットの基本的な算術論理演算を行う。演算問題はBUS2に出力され、フラグがセットされる。フラグはゼロ(Z)、キャリー(C)、オーバーフロー(O)、サイン(S)があり、次にフラグに影響する命令が実行されるまで保持される。

シフター (SHIFT): 32ビットのBarrel Shifterで、CNTに設定された回数のシフトおよびローテイトを行う。32ビットのシフトを5サイクルで行う。

交差回路 (XOVER): 遺伝子の2点交差を行う専用回路で、シフターと共に用いることにより、任意の位置の2点交差が実現できる。乱数によりそれぞれの個体の交差点を7, 2, 交差数を4とした場合の任意の2点交差の場合を図5に示す。ここで、交差点から交差数だけマスクされたビットを切り出すことにより交差を行う。この回路はゲート素子だけで構成している

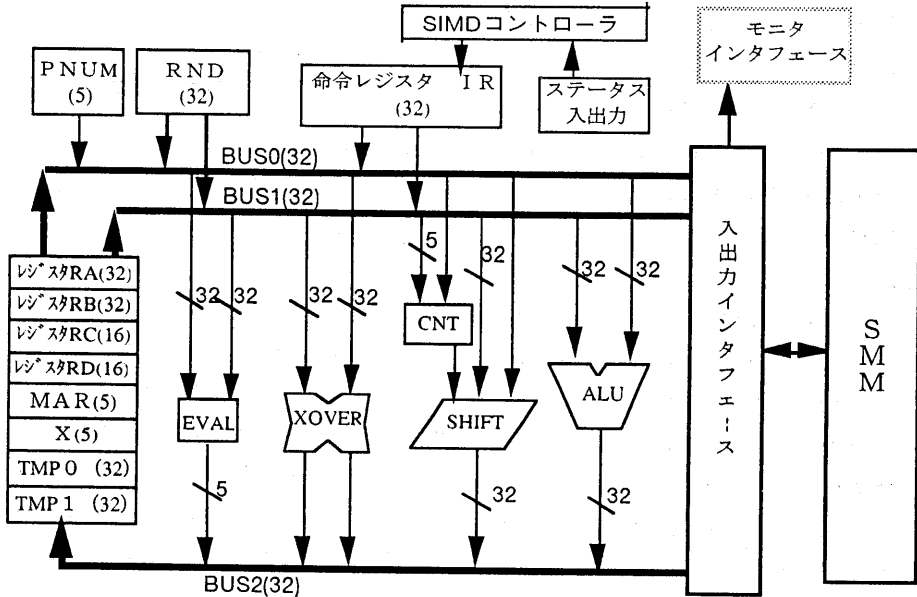


図4. プロセサのアーキテクチャ

ので、ゲートの遅延時間だけで演算を行うことができる。

乱数レジスタ (RND) : 乱数の発生はGAの重要な要素であるが、それぞれのPE上に回路化するためにはできるだけ小さい規模が望ましく、Linear Feedback Shift Register を用いた疑似乱数発生回路で構成する。ガロア体 $GF(2^m)$ の原始既約多項式を用いて周期 $2^m - 1$ の疑似乱数を発生させる。ここでは原始既約多項式

$$X_{31} = X_{27} \oplus X_{13} \oplus X_8 \oplus X_0,$$

$$X_{30} = X_{22} \oplus X_{19} \oplus X_5 \oplus X_3$$

を用いる。図6に乱数発生回路のモジュールの略図を示す。この回路は絶えず新しい乱数をレジスタRNDに出力するので、RNDは常に最新の乱数値をもつ。

プロセサ番号レジスタ (PNUM) : PE固定の番号で、その内容はプロセサ番号の下位に2ビットの00を付け加えた定数がハードウェアにより設定されている。

メモリアドレスレジスタ (MAR) : MAR (5ビット) はSMMへの書き込み、読み出しを行うときのアドレスレジスタである。図7に示すようにMARの上位4ビットはSMMの行を示すCELL (3ビット) とデータ

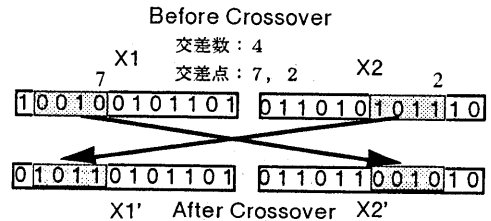


図5. 交差のアルゴリズム

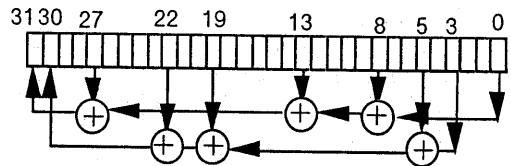


図6. 乱数発生回路のモジュールの略図

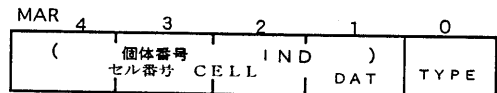


図7. メモリアドレスレジスタMAR

番号DAT (1ビット) で、まとめて个体番号IND (4ビット) としてもアクセスできる。また、最下位ビットはメモリアイプTYPE (1ビット) で、个体情

報と適応度の領域を区別するためのレジスタである。
また、これらの5ビットのレジスタをまとめて、MARとする。SMMへの書き込みはPE番号によって書き定めるメモセルの行が決まっているので、MARの下位2ビットだけを設定すればよく、このときCELLに入っている値は無視される。

インデックスレジスタ (X) : X (5ビット) はMARの保存、その他のインデックスとして使用する。

命令レジスタ (IR) : IR (32ビット) はSIMDコントローラからのSIMD命令コードと付随するデータを保持するレジスタで、命令コードによってレジスタ、演算回路、専用回路とBUS、入出力インターフェースの間のスイッチングが行われる。

その他のレジスタ : 図4において、レジスタRA, RBは1PE当たり2個の個体を割り当てるため、32ビットの個体を記憶するレジスタをおく。レジスタRC, RDは16ビットのレジスタであり、評価値(6ビット)、適応度(6ビット)、ルーレット(10ビット)作成のワーキングレジスタに使用する。ワーキングレジスタTMP0, TMP1(32ビット)は交差時のワーキングレジスタとして使用するほかに、乱数値を一時保存しておくためにも使用する。

5. PEの命令セット

命令語はSIMDコントローラからPEの命令レジスタに送られ、PEはこの命令語によって内部のデータフローを制御する。図8の命令コードに示すように、命令語は全て1ワード(32ビット)で構成する。全ての命令で、命令コードのMSBのifビットが0のときは無条件にその命令を実行し、ifビットが1のときは条件を満たす場合にその命令を、条件を満たさない場合はNOPを実行する。

命令は表1に示すように全部で26種類あり、シフト命令6、算術論理演算命令11、フラグセット命令2、移動命令3、NOP命令、停止命令、XOVR命令、EVAL命令などがある。EVAL命令、XOVR命令はともに専用回路による演算を行う。

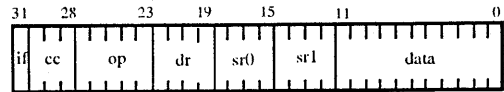


図8. 命令コード

表1. PEの命令セット

タ	NOP		何もしない
イ	HLT		停止命令
ブ	RCF		桁上げフラグCをリセット
1	SCF		桁上げフラグCをセット
タ	SRA	r	右への算術シフト
イ	SLA	r	左への算術シフト
ブ	SRL	r	右への論理シフト
2	SLL	r	左への論理シフト
	RRL	r	右への論理巡回シフト
	RLL	r	左への論理巡回シフト
タ	LD	r	ロード命令
イ	ST	r	ストア命令
ブ	MOV	rd, rs	ムーブ命令
3	INC	r	インクリメント命令
	DEC	r	デクリメント命令
タ	CMP	r0, r1	比較命令
イ	AND	rd, r0, r1	ビット毎の論理積演算命令
ブ	OR	rd, r0, r1	ビット毎の論理和演算命令
3	XOR	rd, r0, r1	ビット毎の排他的論理和演算命令
	NOT	rd, r0	ビット毎の否定演算命令
	ADD	rd, r0, r1	キャリーなし加算命令
	ADC	rd, r0, r1	キャリー付き加算命令
	SUB	rd, r0, r1	キャリーなし減算命令
	SBB	rd, r0, r1	キャリー付き減算命令
タ	EVAL	rd, rs	評価専用回路によりビットが1をカウント
イ	XOVR	r0, r1	XOVR専用回路により交差を行う
ブ	4		

6. 探索問題への適用

上記のSIMD型GA専用プロセッサを探索問題の解法に適用する。探索問題は特定の32ビットパターンを探索する問題を取り上げる。個体の評価は目標値と一致したビット数で表す。

初期集団の生成 : 乱数レジスタRNDは常に新しい乱数を表示しており、初期個体はRNDから32ビットの乱数を2個取り出し、SMMに書き込む。

ソーティング : 評価値の大きい順に個体をO(n)のソーティングを行う。SMMの評価値領域上位の余っている部分に個体番号を入れ、評価値と共にソーティ

ングを行い、完了した後に個体番号のメモセルから個体情報をロードすると、無駄な個体移動がなくより高速にソーティングが行える。この結果、最も評価値の高い個体と低い個体は両側に分かれるので、エリート戦略の適用にも便利であり、適応度の計算においても評価値の最大最小値をメモリからロードするだけで知ることができる。また、ソーティングによりランク法を使用することができる。

ルーレットの作成： 評価値と最小の評価値から適応度を計算し、これをもとにルーレットを作成する。ルーレットを作成するアルゴリズムを以下に示す。par文はそれぞれのPEが並列に同じ操作を実行することを示している。このアルゴリズムにより、n個の適応度をもとにルーレットを作成する演算はオーダー $O(\log n)$ であり、16個のデータであれば各プロセッサは4回の加算を行うだけでメモリ上にルーレットを作成することができる。

```

Making roulette() {
  par (i = 0; i <= n; i++) {
    for (t = 1; t <= log n; t++) {
      a[i] = a[i] + a[i - 2**(t-1)];
    }
  }
  /* here, i < 0 then a[i] = 0 */
}

```

淘汰(Selection)： 乱数RNDの値が作成したルーレット上のどの個体になるかを求め、次の親となる個体を選択する。この結果、ルーレットの幅の広いもの、すなわち、適応度の大きいものほど選択される確率が高く、適応度の低いものは淘汰される。ルーレットを回す操作は二分探索法を用いて行う。

エリート保存戦略を用いる場合はPE番号が0のものだけ、一方の親をエリートとすることによって保存することができる。

交差 (Crossover)： 2点交差、単純交差、一様交差を行うことができる。交差率と乱数により交差を行うかを決定し、交差を行う。2点交差、単純交差は専用回路を用いてXOVR命令により行う。一様交差は次のようなプログラムで実行する。

```

MOV MAR, d1      AND TMP1, M
LD RA            MOV MAR, d1
MOV MAR, d2      AND TMP2, M
LD RB            OR RA, TMP1
MOV TMP1, RND    OR RB, TMP2
NOT TMP2, TMP1   ST RA
AND RA, TMP1     MOV MAR, d2
AND RB, TMP2     ST RB

```

突然変異 (Mutation)： 突然変異率と乱数により突然変異を起こすか否かを決定し、乱数によって求めた特定のビットを反転することにより突然変異を起こす。

7. まとめ

提案したSIMD型遺伝アルゴリズムと専用PEを用いて、特定の32ビットパターンを探索する問題への適応法について考察した。その他の探索問題への適用として、SMMの構成を変更することによりナップザック問題にも適用することができる。

このSIMD型GA専用PEはN T T社から提供を受けたPARTHENON(Ver.3.1)を用いて回路記述を行い、FPGAを用いて製作している。これまで専用回路の設計を行ったが、これからの課題として、システム全体の設計を早急に行い、SIMD型遺伝アルゴリズム専用計算機としての機能を拡張していきたい。

参考文献

- [1] Lawrence Davis: Handbook of Genetic Algorithms, pp.1-22, Van Nostrand Reinhold, 1991
- [2] D. E. Goldberg: Genetic Algorithms in Search, optimization and Machine Learning, pp. 208-212, Addison Wesley, 1989
- [3] 北野宏明編： 遺伝的アルゴリズム, pp.1-66, 産業図書, 1993
- [4] 高橋義造編：並列処理機構, pp. 199-206, 丸善, 1988
- [5] Hennessy, Patterson (富田眞治訳)：コンピュータアーキテクチャ, 日経BP出版センター, pp. 191-235, 1992
- [6] Kai Hwang: Advanced Computer Architecture, McGraw-Hill, pp. 238-247, 1993
- [7] Tanese, R: Distributed Genetic Algorithm, Proc. of 3rd Int. Conf. on Genetic Algorithms, pp.434-439, June 4-7, 1989
- [8] Lawrence Davis: Genetic Algorithms and Simulated Annealing, Morgan Kaufmann Publishers, pp. 129-140, 1987