ストリームプロセッシングに対応した分散遺伝的アルゴリズムの提案

楠堂　航

1　はじめに

Intel CPU のような汎用プロセッサは 2000 年頃から，処理速度の成長率が鈍化の傾向にあり，頭打ちになると考えられる．それに伴い従来，画像処理，動画処理に用いられたメディアプロセッサが汎用処理に用いられるようになってきた．メディアプロセッサの代表としては，GPU や Cell BroadbandEngine(Cell/B.E.) がある．このようなプロセッサは，CPU で実行されるようなアルゴリズムでは性能を発揮することができないが，並列性を持ったアルゴリズムの処理を得意としたアーキテクチャとなっており，特に NVIDIA の GPU や Cell/B.E.[1] では，Stream 処理と呼ばれるデータフローを固定したアルゴリズムを得意とするアーキテクチャになっている．そのためアルゴリズムを変更する必要があり，高速フーリエ変換 [2] もそのようにアルゴリズムの変換が行われた．今後これらのアーキテクチャの普及が進むにあたり，アーキテクチャに特化したアルゴリズムの設計が必要となる．本研究では様々なアルゴリズムが研究されている遺伝的アルゴリズム (Genetic Algorithm:GA) を対象に，ストリーム化について，実装と性能評価を行う．

2　近年のハードウェアとストリーム処理

2.1　近年のハードウェア

近年，GPU，Cell/B.E.，FPGA などのハードウェアの成長が注目されている．いずれのハードウェアもマルチコアとなっており，ストリーム処理が行える．これにより，高速な演算が可能であり，CPU も，拡張命令として，Streming SIMD Extension(SSE) と呼ばれる命令がついている，このようにストリーム処理が広まってきている現在，プログラムを効率よく動作させるためには，プログラムをストリーム型にすべきである．

2.2　ストリーム処理

ストリーム処理とは，一方向に流れるデータに対し，パイプライン的に処理を行う手法である．ハードウェア上のストリーム処理流れを，Fig.1 に示す．Fig.1 のように，隣接したコア間にバスが接続しており，コア間で一方向通信を行う．データは最初のコアから，最後のコアまでシーケンシャルに処理されていくため，多数のコアから同時に出力したデータが衝突することなく伝わる．ストリーム処理に対して，共有メモリ型におけるハードウェア上のデータの流れを Fig.2 に示す．共有メモリ型の並列処理では，Fig.2 のように，コア間通信を行うとき，いずれも共有メモリを通過する必要がある．多数のコアが同時にコア間でデータの交換を行うと，メモリの帯域幅を超えてしまう．

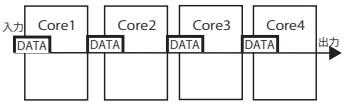

Fig.1　ストリーム型

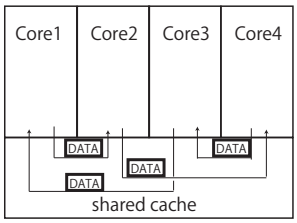

Fig.2　共有メモリ型

3　GA

3.1　概要

GA は最適化問題を確率的探索を用いて解く手法である．GA は連続性，非連続性，可微分性，および局所的最適解の多さに制限がないことから多様な分野で使用される一方で，探索負荷が非常に大きく，現実的な時間で解が求まらない場合も数多く存在する．そのため GA の基である「単純 GA(SimpleGA:SGA)」以外に，負荷分散を行い，並列に処理する GA が多数考えられた．そのような GA の一つに「分散 GA(Distributed Genetic Algorithm:DGA)」が存在する．DGA は負荷分散を行うだけでなく，解に到達する可能性が高くなることが知られている [3]．また DGA には PC クラスタに対応した DGA[4] など様々なアーキテクチャに合わせたものが作成されている．

3.2　SGA

SGA では，初期化と呼ばれる作業をはじめに行い，母集団を生成する．母集団はいくつかの個体からなり，個体とは解である．初期化以降は，個体群に対して，選択，交叉，突然変異の 3 つの遺伝的操作および評価を繰り返し，最適解を探索する．遺伝的操作によって変化した個体を評価し，解の探索を進める．これらの作業を繰り返し行うことで最適解を探索する．この一連の流れを世代と呼ぶ．1 世代の中で，選択では優秀な個体を増加させ，劣悪な個体を淘汰する．次に交叉では個体を組み合わせて，新しい個体を作る．突然変異で個体の一部を書き換え，局所的最適解に留まらないようにする．評価ではそれぞれの個体の適合値を求める．適合値が高い個体は優秀な個体である．終了判定で条件を満たしていれば，SGA を終了し，解の値を表示する．終了条件は水準以上の解を見つけた場合と指定した回数の繰り返し行なった場合である．

3.3 DGA

SGA の母集団が 1 つであることに対し，DGA は母集団をいくつかの島 (サブ母集団) に分ける．島ごとに，SGA と同様に選択，交叉，突然変異および評価を行う．DGA では，数世代ごとに移住と呼ばれる操作が加わる．移住は，ある島のいくつかの個体を，別の島に送る操作である．DGA では，各々の島を環状になるようにつなぎ，繋がれた島に移住する．この環の形は，毎世代変更するほうが，解の発見率が高まることが知られている [3]．そのため，今回は環の形を毎世代変更する方法を採用した．今回使用する DGA の移住方法を Fig.3 に示す．DGA は，移住間隔や移住個体の選び方などパラメータが増大するという欠点がある一方で，利点として複数の島で別々に探索を行うため，解の探索範囲が広がることや移住操作によって，局所最適解から抜け出しやすくなる他，島ごとに一つのプロセッサを与えることで，マルチコア化が簡単に可能であるということが上げられる．しかしこの移住は，データが一方向に流れないため，ストリームには行えず，共有メモリ型のハードウェアで実装することとなる．これは Fig.2 のコアを島に，データを移住する個体に対応させたものとなる．よって移住では，共有メモリを同時にアクセスし，ボトルネックが発生する．

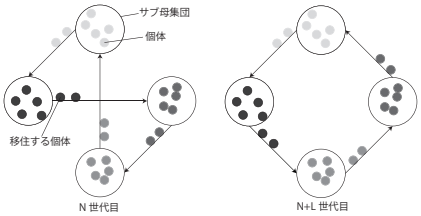

Fig.3　DGA の移住方法

4 StreamGA の提案

4.1 StreamGA の概要

ストリームプロセッサで実行できるようにした DGA の拡張モデルとして，StreamGA を提案する．これは GA をストリームに実行できるように，島を直線状に繋いだものである．個体は島に沿って一方向に流れる．さらに DGA と異なり StreamGA では島の順番を変更をできない．この移住方法を Fig.4 に示す．移住に 2 種類の方法を提案する．1 つ目は，島の個体が減らないように，新しい個体を作成し，初めの母集団に送り続ける方法である．これは Fig.4 に示す移住方法であり，初めの母集団とは Fig.4 では，Core2 に対応する．この手法を N.StreamGA と定義する，2 つ目は出力結果を再び，入力に戻す方法である．この手法を L.StreamGA と定義する．ただし L.StreamGA は，出口が一箇所であり，パイプを繋ぎ直さない DGA である．そのため本研究では DGA と比べて違いが大きい N.StreamGA を中心に述べる．いずれのストリーム処理も移住方法は Fig.1 のようなハードウェアで実行出来るため，DGA で移住のときに生じるボトルネックがなくなり，処理時間が短くなると考えられる．

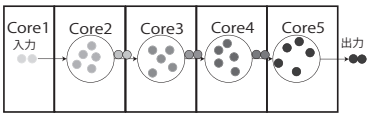

Fig.4　StreamGA の移住方法

4.2 StreamGA の特徴

StreamGA は，移住方向が決まっているため，固定の島の間の連結が強く，局所最適解に長くとどまり続けることが考えられる．DGA では，島が局所最適解に陥ったときに，様々な島からの移住で，別の個体を得られた．また StreamGA は，解の出力は最後の島からしか得られないということもある．DGA はいずれかの島で最適解が求まると，GA の動作が終了するが，StreamGA では途中の島で最適解となる個体が求まっても，最後の島まで個体が移住しなければ最適解が得られない．この問題が与える影響を考慮し，5 章で StreamGA の性能を検証する．

5 StreamGA の評価

5.1 検証内容

StreamGA を DGA と比較し検証する．検証対象の問題として Rastrigin,Ridge,Griewank,Schwefel を採用した．それぞれの特徴を Table1 に示す．今回使用した問題はテスト関数と呼ばれ, 最適化手法の性能の指標を調べるために，しばしば用いられる．また今回の実験に使ったパラメータを Table2 に示す．

5.2 テスト結果

Fig.5，Fig.6，Fig.7 は，何回目の評価で解が出力されたかを示す．評価回数に比例し，計算処理が増加する．今回はそれぞれのテスト関数を SGA，DGA，StreamGA で 12 回ずつ解き，その平均の評価回数をに示した．また Griewank 関数は最適解の発見に至ることが少なく，評価

Table1　テスト関数

表示	関数名	式	定義域	最適値
F1	Rastrigin	$F1(x) = 10n + \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i))$	$(-5.12 \leq x_i < 5.12)$	$min(F(x)) = F(0, 0, ..., 0) = 0$
F2	Ridge	$F3(x) = \sum_{i=1}^{n}(\sum_{i=1}^{n} x_j)^2$	$(-64 \leq x_i < 64)$	$min(F(x)) = F(0, 0, \cdots, 0) = 0$
F3	Griewank	$F4(x) = \sum_{i=1}^{n} \frac{x_i}{4000} + \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}})$	$(-512 \leq x_i < 512)$	$min(F(x)) = F(0, 0, \cdots, 0) = 0$
F4	Schwefel	$F2(x) = \sum_{i=1}^{n}(-x_i \sin(\sqrt{\lvert(x_i\rvert)}))$	$(-512 \leq x_i < 512)$	$min(F(x)) = F(429.968750, \cdots,) + 418.982887 * n = 0$

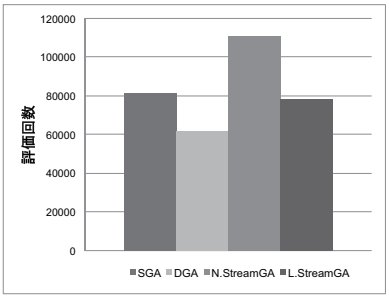

Fig.5　Rastrigin の評価回数

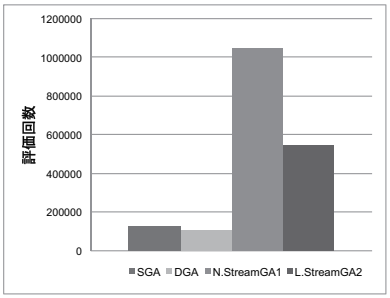

Fig.6　Ridge の評価回数

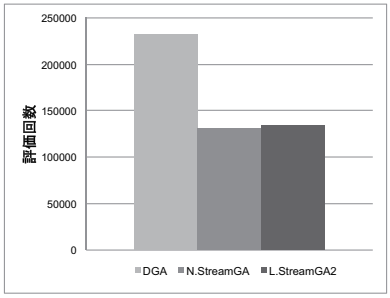

Fig.7　Schwel の評価回数

Table2　パラメータ

Number of indivdual	400
Number of elite	1
chromesome Length	150
chromesome Length(Griewank)	80
Selection	Tournament selection
Crossover rate	1.pt crossover
Mutation	1/150
Migration gap	5generation
Migration rate	0.5
Number of variable	10
Number of variable(Rastrigin)	20
Threshold	0.0
Threshold(Ridge)	0.045
Threshold(Schwefel)	0.083

回数も安定しなかったため省略する．

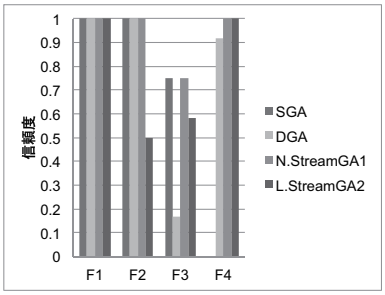

Fig.8　信頼度

最適解の発見割合 (信頼度) を Fig.8 に示す．この結果より N.StreamGA は，他の移住方法と比べ，最適解の発見割合が高いことが分かる．最後に，どの島が優秀な個体を持つかを調べる．Rastrigin において島数が 10 のとき，入力側の島から順番に，島 1, 島 2, …. として割り当てる．その島内の一番優秀な個体の評価値と世代数の関係を，Fig.9 に表す．Fig.9 によると，後の島ほど優秀な個体が存在することがわかる．

6　議論

最後の島の出力からのみ結果が取れるという N.StreamGA の弱点は，Fig.9 の結果より問題ないことが分かる．最適解の個体は最後の島におり，移住の際に結果として出力されると考えられる．

信頼度の比較で，N.StreamGA に比べ，L.StreamGA は信頼度が上がるということがわかった．これは新しい個体を入力し続けることで，個体の多様性得られ，探索範囲が広ったからだと考えられる．この影響を顕著に受

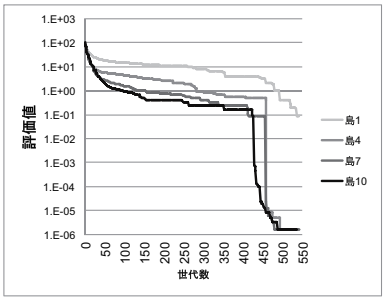

Fig.9　島の順序と評価

けているのは，Griewank 関数である．この関数は多数の局所最適解を持つ，その局所最適解からの脱出に新しい個体が影響を与えたと考えられる．L.StreamGA は，収束までにかかる評価回数が大きかった．これは最後の島に存在する優秀な個体を出力によって捨ててしまうことが原因だと考えられる．この影響を大きく受けているのは局所最適解の無い Ridge 関数で，優秀な個体が捨てられているため解が求まるまでに長い時間がかかったと考えられる．ストリーム処理に対応させた GA であるため，ストリームプロセッサで実行させれば，DGA との計算時間の差は，今回に比べ小さくなると考えられる．

7　まとめと今後の展望

本研究では，DGA を Stream 型に拡張したものを，ソフトウェア上でシミュレーションし検討した．シミュレーション結果を見ると，DGA と比べ，最適解を求めるまでにかかる評価回数が多いことが欠点である．一方で，解に到達できる可能性が高いという利点がある．ただし評価回数の増加に伴う計算時間は実際にストリームプロセッサ上に実装し，効率よく動作させた上で，DGA と比較し，計算時間がどのくらいになるのか検討する必要がある．

参考文献

1) 上村剛, 大溝孝, 粟津浩. Cell リファレンスセット概要, 2006.
2) 浦岡祥. FFT のデータ駆動型並列実現法, 2005.
3) 三木光範, 廣安知之, 畠中一幸, 吉田純一. 並列分散遺伝的アルゴリズムの有効性, 2001.
4) 廣安知之, 三木光範. PC クラスタシステムにおける並列分散遺伝的アルゴリズム, 2000.