

# ストリーム型遺伝的アルゴリズムの提案と GPU を用いた実装

楠堂 航<sup>†</sup> 吉見 真聡<sup>†</sup> 三木 光範<sup>†</sup> 廣安 知之<sup>††</sup>

<sup>†</sup> 同志社大学理工学部    <sup>††</sup> 同志社大学生命医科学部

## 1 はじめに

近年、科学計算に利用されているマルチコアプロセッサは、近接するコアのみを接続し、メモリレイテンシ、通信遅延を抑える特徴を持っている。特に GPU や Cell Broadband Engine といった Stream プロセッサは、データがコア間を一方方向に転送されつつ、各々の段階で処理されることで、シストリックアレイを構成し、並列処理と入出力ボトルネックの低減を実現している。

遺伝的アルゴリズム (Genetic Algorithm : GA) は様々な最適化問題で利用されており、分散 GA (Distributed GA : DGA) などの並列化手法の研究も盛んである。現在、Stream プロセッサ環境に適した GA は存在しないが、分散 GA を Stream 処理に対応させることで、Stream プロセッサ上で高速に演算できると考えられる。本研究報告ではデータフローを一方方向に固定した DGA を提案し、Stream プロセッサとして GPU を使用して実装した性能について議論する。

## 2 GA

### 2.1 概要

GA は確率的探索を用いて、最適化問題を解く手法である。その特徴は、最適化問題の対象とする目的関数における微分値を用いないため、関数の連続性、離散性に関わらず使用できることである。

GA では解候補の集合を母集団と呼び、解候補を個体と呼ぶ。個体に対し、「選択」「交叉」「突然変異」の遺伝的操作を繰り返すことで、最適解の探索を行う。この繰り返し単位を世代と呼び、母集団が一つである GA を単純 GA (Simple GA : SGA) と呼ぶ。

### 2.2 DGA

GA の負荷分散を行い、並列に計算する手法の一つに、DGA が存在する。これは母集団をいくつかのサ

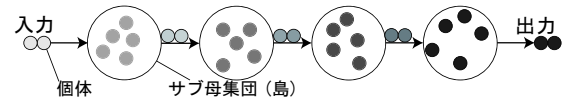


図 1: Stream GA の移住モデル

ブ母集団に分割し、各サブ母集団が並列に遺伝的操作を行う手法である。このような操作により、各サブ母集団において、異なる解の探索が行われる。また DGA では、一定の世代間隔で、サブ母集団間において個体を交換する「移住」と呼ばれる操作を行う。DGA はサブ母集団内の全ての個体が、局所最適解に陥った場合も、移住により再び解の探索が進む。DGA はこの操作により、SGA と比較し、優秀な解を得られることが知られている [1]。加えて移住では、個体を同一のサブ母集団へ転送するより、異なるサブ母集団に転送する方が優秀な解が得られることも知られている [2]。また、DGA はその並列性の高さから、PC クラスタ [3] や PC グリッド [4]、および GPU [5] といった、様々な実行環境に合わせたアルゴリズムが報告されている。

## 3 Stream GA

### 3.1 Stream GA の提案

Stream GA とは、DGA における移住時のデータフローを一方方向に固定し、GA を Stream プロセッサで実行できるようにしたものである。

図 1 に Stream GA の移住モデルを示す。Stream GA の特徴は、個体の転送が一方方向に固定される点、末端にあるサブ母集団の片方に新しい個体を入力する点、もう片方から出力された解を処理結果として採用する点である。入出力を末端で行う理由は、一般的な Stream プロセッサでは末端からのみ、データの入出力が可能であるためである。そして、Stream プロセッサにおいては、移住、入力、出力は並列に行うことが理想である。

### 3.2 Stream GA の実装

本章では、Stream GA を GPU 上で実装した手法について示す。GPU 上に大きな配列を用意し、配列の各要素に個体を代入する。それぞれのサブ母集団は単一の配列を共有しており、一定の区間を占有している。移住時のみ区間の境界を越えて、個体を転送する。図 2 にメモリ領域から見た移住方法を示す。メモリ上では、移住時に全ての個体を同様に移動させる。その

A proposal and implementation of Stream Genetic Algorithm

<sup>†</sup> Ko KUSUDO(kkusudo@mikilab.doshisha.ac.jp)

<sup>†</sup> Masato YOSHIMI(myoshimi@mail.doshisha.ac.jp)

<sup>†</sup> Mitsunori MIKI(mmiki@mail.doshisha.ac.jp)

<sup>††</sup> Tomoyuki HIROYASU(tomo@is.doshisha.ac.jp)

Faculty of Science and Engineering, Doshisha University (<sup>†</sup>)

Faculty of Department of Life and Medical Science, Doshisha University (<sup>††</sup>)

1-3 Miyakodani, Tataru, Kyotanabe, Kyoto 610-0321, Japan

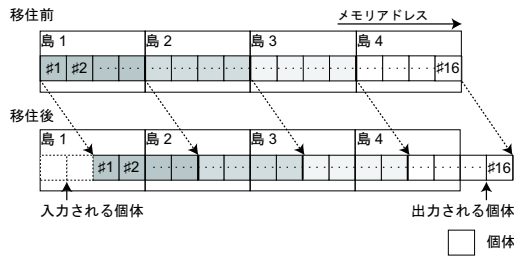


図 2: GPU 上での Stream GA の移住

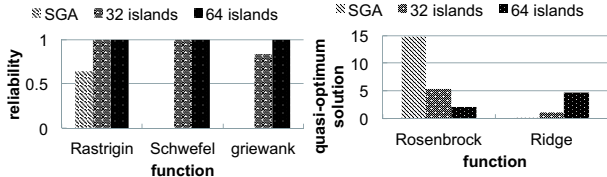


図 3: 信頼性の比較

図 4: 準最適解の比較

理由として、GPU は個別の処理は遅く、また、連続したメモリへのアクセスは高速なためである。

#### 4 数値実験

本章では、Stream GA と SGA を得られた解と実行速度の観点から比較し、Stream GA の性能について検討する。表 1 に、比較に使用した 5 つの最適化問題 (関数) を示した。これらの関数を用いて、それぞれの GA で 30 回ずつ試行した。表 2 に比較に使用したパラメータを示し、表 3 に実行環境を示す。

最適解を発見できた関数は、図 3 に発見した割合 (信頼度) を示す。一方、その他の関数については、図 4 に得られた準最適解の平均を示す。本論文で使用した関数において、最適解の値は全て 0 であり、0 に近いほど優秀な解である。図 3, 4 の結果より、Ridge 関数を除く全ての関数で、Stream GA が SGA に比べて、優秀な解を得られたことが確認された。一方 Ridge 関数では、SGA の方が優秀な解を得られた。これは Ridge 関数が単峰性であり、Stream GA は入力による多様性が存在するので、SGA と比較し優秀な解が得られるまでに、大きな世代を要するためと考えられる。

Rastrigin 関数において、図 5 に 1000 世代までの、図 6 に最適解が得られるまでの実行時間の平均を示す。

表 1: テスト関数

関数名	式	遺伝子長	次元数
Rastrigin	$F(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$	200	20
Ridge	$F(x) = \sum_{i=1}^n \sum_{j=1}^n x_j^2$	200	20
Griewank	$F(x) = \sum_{i=1}^n \frac{x_i}{4000} + \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}})$	200	20
Schwefel	$F(x) = \sum_{i=1}^n (-x_i \sin(\sqrt{ x_i }))$	450	15
Rosenbrock	$F(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2)$	200	20

表 2: パラメータ

総個体数	512
サブ母集団数	2,4,8,16,32,64
移住率	0.5
移住間隔	5 世代ごと

表 3: コンピュータのスペック

OS	ubuntu11.04
CPU	Core i5 2400K
GPU	GeForce GTX 460 1GB 336SPs
Memory	8GB
CPU code Compiler	CUDA toolkit 4.0
GPU code Compiler	gcc 4.4.5-15ubuntu11.04

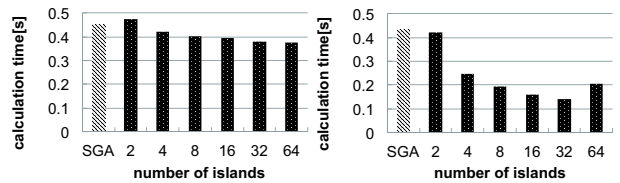


図 5: 1000 世代までの実行時間  
図 6: 最適解が出るまでの実行時間

Stream GA は、島数が多くなるにつれて、1 世代あたりの計算時間が短くなっていった。これは他の 4 つの関数も同様の結果であった。また Stream GA は SGA と比較し、短い実行時間で最適解を得られた。これは Schwefel, Griewank 関数も同様の結果であった。

#### 5 まとめと今後の展望

Stream GA は、実験に使用した 4 つの関数において、SGA と比較し短い計算時間で、同等以上の解を得られることが確認できた。これは Stream GA が複数の島を持つこと、および新しい個体の導入を行っていることから多様性が得られた結果だと考えられる。

今後の展望としては、Stream GA の島ごとにおける性質を明らかにすること、今まで GA の実装が困難であった、GPU 以外の Stream プロセッサ上で実装することが挙げられる。

#### 参考文献

- [1] Reiko Tanse *et al.* Distributed genetic algorithms. *ICGA*, pp. 434–439, 1989.
- [2] 廣安知之, 他. 実験計画法を用いた分散遺伝的アルゴリズムのパラメータ推定. *日本機械学会*, Vol. 68, No. 670, 2002.
- [3] Shigeyoshi Tsutsui *et al.* Solving quadratic assignment problems by genetic algorithms with gpu computation: a case study. *GECCO*, pp. 2523–2530, 2009.
- [4] 廣安知之, 他. 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. 最適化シンポジウム講演論文集, Vol. 2000, No. 4, pp. 305–310, 2000.
- [5] 谷村勇輔, 他. Grid 計算環境における遺伝的アルゴリズムの提案. *情報処理学会*, Vol. 43, No. 4, pp. 986–994, 2002.