StreamGA の提案と GPU を用いた性能評価

楠堂 航
Ko KUSUDO

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つ．特に Stream プロセッサは，データがコア間を一方向に転送されつつ，各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 [1]．

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 [2]．

我々は DGA 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た．本研究報告ではデータフローを一方向に固定した DGA を提案し，GPU を使用して実装した性能について議論する．

2 GA

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．

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ブ母集団 (島) に分割し，各島が独立して操作をし，島ごとに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，島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 は島内の全ての個体が，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 [2] [3]．加えて移住では，個体を同一の島へ転送するより，異なる島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 [4]．

3 Stream GA

3.1 Stream GA の提案

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

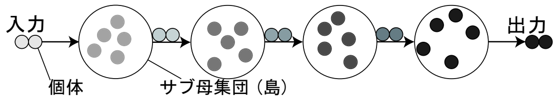

Fig. 1 StGA の移住モデル

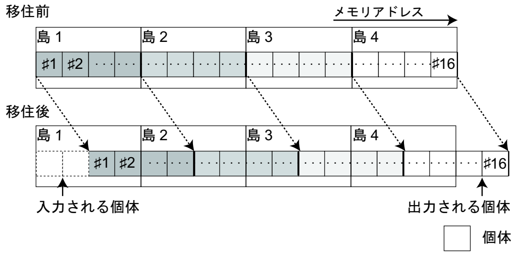

Fig. 2 GPU 上での Stream GA の移住

Fig. 1 に StGA の移住モデルを示す．StGA の特徴は，個体の転送が一方向に固定される点，末端にある片方の島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なためである．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

本節では，StGA 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．それぞれの島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．Fig. 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

実行回数に対する最適解を発見する割合，準最適解の値，スケーラビリティから StGA の性能を調査した．調査には，Table 1 に示す異なる 5 つのテスト関数を使用した．テスト関数とは，最適化手法の性能を測るために用いられる関数である．数値実験に使用するパラメータ，実行環境を Table 2，3 に示す．

4.1 信頼性

一定世代までの探索を複数回行った結果，最適解を発見できた割合を信頼性と呼ぶ．Fig. 3 に，各々のテスト

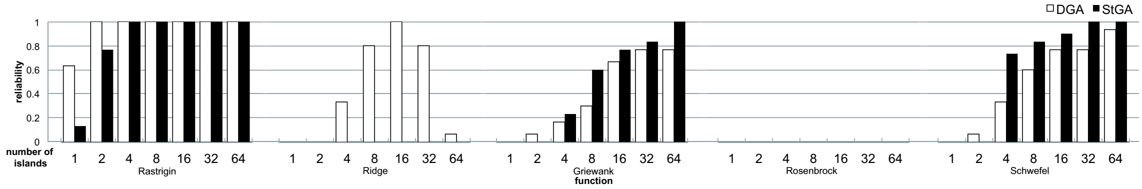

Fig. 3　信頼性

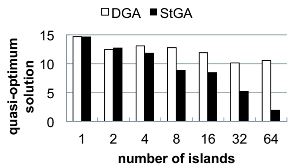

Fig. 4　Rosenbrock 関数の準最適解

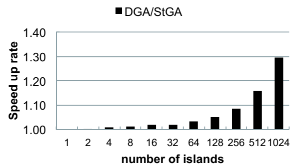

Fig. 5　スケーラビリティ

関数を 1000 世代まで，30 回試行した際の信頼性を示す．

Fig. 3 より島数が 4 以上のとき，テスト関数 F1，F3，F5 で DGA 以上の信頼性を持つことがわかった．

4.2　準最適解

Fig. 3 より実行した全ての GA で F4 の最適解を見つけてないことがわかる．これは F4 が，局所解を多数持つため，GA での探索が困難であるからと考えられる．よって，得られた準最適解を用いて比較した．Fig. 4 に，30 試行の準最適解の平均値を示す．F4 の最適解は 0 であり，0 に近い値が優秀な解である．Fig. 4 より F4 においても，StGA が優秀な解を得ていることがわかる．

4.3　スケーラビリティ

Fig. 5 にテスト関数に F1，個体数に 4096 を用いた際の，StGA における島数と実行時間の関係を示す．Fig. 5 の縦軸は，同数の島数における「DGA の実行時間 / StGA の実行時間」である．Fig. 5 より島数が増加に応じて DGA より StGA が速くなっていることがわかる．その理由として，島数の増加に応じて，StGA は入出力する個体数が減少すること，また，DGA は終了処理で行う解の判定に必要な計算時間が増加することが挙げられる．以上のことから，StGA はコア数に合わせて多数の島を用いることが可能なアルゴリズムだと言える．

Table 1　テスト関数

関数名	式	遺伝子長	次元数	
F1	Rastrigin	$F(x) = 10n + \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i))$	200	20
F2	Ridge	$F(x) = \sum_{i=1}^{n}(\sum_{j=1}^{n} x_j)^2$	200	20
F3	Griewank	$F(x) = \sum_{i=1}^{n}\frac{x_i}{4000} + \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})$	200	20
F4	Rosenbrock	$F(x) = \sum_{i=1}^{n-1}(100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2)$	200	20
F5	Schwefel	$F(x) = \sum_{i=1}^{n}(-x_i \sin(\sqrt{\lvert x_i \rvert}))$	450	15

Table 2　パラメータ

総個体数	512
島数	2,4,8,16,32,64
移住率	0.5
移住間隔	5 世代ごと

Table 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	gcc 4.4.5-15ubuntu11.04
GPU code Compiler	CUDA toolkit 4.0

5　議論

StGA では多くのテスト関数で，DGA より高い信頼性，あるいは優秀な準最適解を得られた．この理由として，新しい個体の導入により前方の島で広域的に優秀な解を探索し，後方の島で島内の優秀な解の近傍を探索していることが考えられる．このため，局所解が多い F3 や，局所解と最適解が離れている F5 では優秀な解を得られている一方で，F2 のような単峰性の関数の一部では，最適解から距離の遠い範囲を探索し，信頼性が低下したと考えられる．以上より，StGA は単峰性でない問題において十分な解探索性能を得られると考えられる．

StGA は移住操作時の通信処理の複雑さ，および終了処理時の最適解判定に必要な計算量が島数に依存しない．このことから，StGA は多数のコアを持つ Stream プロセッサ上での実行に適していると言える．

6　おわりに

本論文では GPU を用いて StGA を実装し，その信頼性，準最適解，スケーラビリティの観点から DGA と比較した．StGA は F2 を除く使用した全てのテスト関数で，DGA と同様かそれ以上の信頼性を持つことを確認した．また，StGA の GPU 実装における 1 世代あたりの実行速度は，島数の増加に応じて DGA より高速になった．以上の結果により StGA は Stream 処理による GA として，十分な性能を持つことが確認できた．

今後の展望として，コア数の増減が可能な Stream プロセッサ上に StGA を実装し，コア数の増加に応じて実行時間が短縮されることを確認することが挙げられる．

参考文献

1) Ujval J. Kapasi *et al.* The imagine stream processor. *ICCD*, pp. 282–288, September 2002.

2) Reiko Tanse. Distributed genetic algorithms. *ICGA*, pp. 434–439, 1989.

3) Theodore C.Belding. The distributed genetic algorithm revisited. *ICGA*, Vol. 10, No. 2, pp. 114–121, 1995.

4) 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.