

G-013

## サーバ・クライアントモデルによる非同期並列遺伝的アルゴリズム Asynchronous Parallel Genetic Algorithm using Server-Client model

小嶋 和徳<sup>†</sup>  
KOJIMA Kazunori

石亀 昌明<sup>†</sup>  
ISHIGAME Masaaki

松尾 広<sup>‡</sup>  
MATSUO Hiroshi

### 1. はじめに

本研究は、遺伝的アルゴリズム (GA) の処理を並列化する並列分散遺伝的アルゴリズムに (PDGA) について、従来の踏み石モデル、非同期通信モデル、Sigma-Exchangeモデルと、本研究で使用しているサーバ・クライアント型モデルとの比較を行うものである。

本稿では、実装面および性能面の比較を行い、サーバ・クライアント型モデルが、実装が容易で、通信量が少なく、性能の劣化がない優れた方法であることを報告する。

### 2. 従来手法

従来の PDGA[1] で代表的なものは、リング状のトポロジーを使用し、定期的に隣の島へ移民を送り、別の隣の島から移民を受け取る踏み石モデルと呼ばれるものである。踏み石モデルは実装は非常に容易であるが、個々の部分集団に対する移民の必要性については考慮されないため、通信の面で無駄が多いモデルである。

非同期的に移民を行う方法として、一定確率でランダムに移民を行う方法があるが、この方法でも移民の必要性については考慮されておらず、また、非同期的なイベントを処理するのは一般的に非常に難しい。C 言語で実装するためにはソフトウェアシグナルやスレッドプログラミングなどを用いる必要がある。ただし、実装できた場合でも通信頻度が多くなるとデッドロックが起りやすくなるなどの問題から、非常に実現が難しい方法である。

これらに対し、部分集団の標準偏差の変動を感知し、ある程度収束したら移民を行う、Sigma-Exchange が提案されている [2]。棟朝らはこの方法を独自のシステムで実装したため、一般的な並列計算機やネットワーク接続されたワークステーションや PC 上で実装するのは非常に難しい。特に、移民のタイミングを決定する、収束を観測するためのパラメータは、問題の性質や集団サイズにより変更する必要があるため、ユーザにとっては非常に負荷となる。また、この方法は移民時にメッセージのやり取りを複数回行う必要があるため、通信頻度が高くなるとデッドロックが起りやすくなり、安定したものを実現するのは非常に難しい。

### 3. 実装の比較

図 1 および図 2 は、Sigma-Exchange モデルのメイン部とメッセージ処理部の問題分析図 (PAD) を示している。

これらの図からもわかる通り、Sigma-Exchange モデルでは、移民を行う際に近隣へのパラメータ要求、近隣からのパラメータ受信、移民要求、移民受信と複数回のメッセージ交換を行う必要があり、またこれらのメッセー

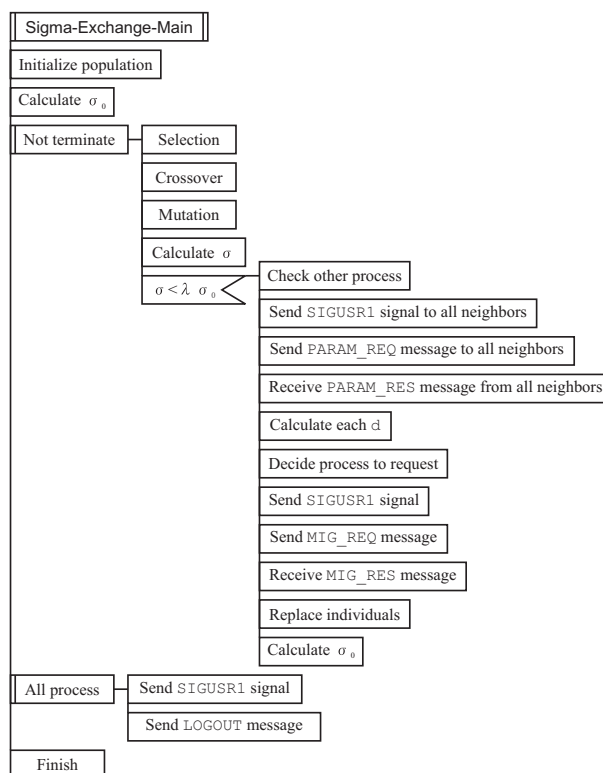


図 1: Sigma-Exchange メイン部の PAD

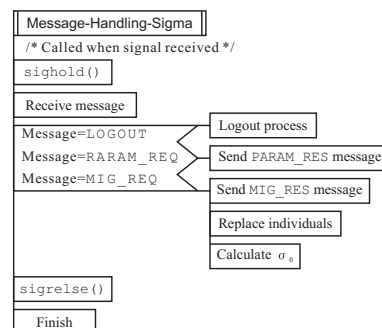


図 2: Sigma-Exchange メッセージ処理部の PAD

<sup>†</sup>岩手県立大学, Iwate Pref. Univ.

<sup>‡</sup>三島学園女子短期大学, Mishima Gakuen Women's Jr. Coll.

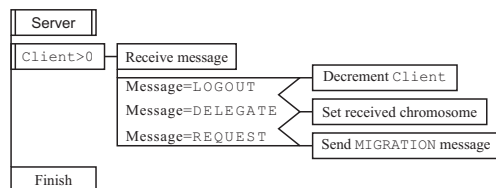


図 3: サーバプログラムの PAD

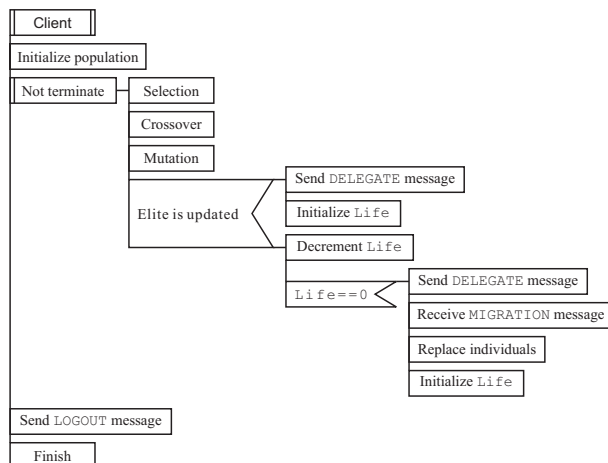


図 4: クライアントプログラムの PAD

ジはその性質上、他のメッセージと別に処理を行う必要があるため、実装は非常に難しいものとなる。

移民のタイミングを決めるパラメータ  $\lambda$  は、収束しやすさなどの問題の性質や集団サイズ、あるいはトポロジーにより適宜変更しなければならず、非常に不便である。特に、パラメータ  $\lambda$  の設定が適当なものでなければ、ほとんど移民が行われなかったり、逆に頻繁に移民が行われデッドロックを引き起こしたりしてしまう。このことから、最適な値を決定するためには、非常に多くの予備実験を行う必要があり、使用するユーザにとっては非常に負担となるものである。

図 3 および図 4 は、サーバ・クライアントモデルの PAD である。Sigma-Exchange と比較しても、非常に簡素に実装できることがわかる。各クライアントはサーバとのみ通信を行い、他のクライアントプログラムとの通信の必要がないため、個々のクライアントプログラムは完全に独立している。また、通信の方向が決まっているため、複雑にならず、デッドロックもほとんどおこらない安定した手法である。

#### 4. 性能の比較

ここでは、32 の CPU を持つ SGI 製の Origin2000 に、踏み石モデル (Synchronized)、非同期ランダムモデル (Random)、Sigma-Exchange モデル (Sigma)、そしてサーバ・クライアント型モデル (Server-Client) を実装し、5 次元の多峰性関数に適用した結果について述べる。

表 1 は最大適合度 (最適値 1.0) の比較、表 2 は実行時間の比較、表 3 は通信量の比較である。これらの表で

表 1: 最大適合度の比較

	The number of islands		
	1	8	32
Synchronized	0.989	0.995	0.999
Random	0.989	0.997	0.999
Sigma	0.989	0.995	0.999
Server-Client	0.989	0.996	0.999

表 2: 実行時間の比較 (sec)

	The number of islands		
	1	8	32
Synchronized	4483.226	74.441	7.907
Random	4483.226	77.044	8.206
Sigma	4483.226	77.331	8.120
Server-Client	4483.226	76.331	8.849

表 3: 通信量の比較 (bytes)

	The number of islands		
	1	8	32
Synchronized	0.00	5000.00	1200.00
Random	0.00	10483.36	2542.08
Sigma	0.00	1168.32	202.56
Server-Client	0.00	197.92	214.88

は、各パターンにつき 25 回の試行を行い、その平均値を表している。また、通信量は各部分集団が送受信した通信量を示している。

最大適合度および実行時間いずれも、どの手法でもほぼ同等の結果となっている。

通信量では、サーバ・クライアントモデルが他の手法と比べて多くの場合通信量が少ないことがわかる。

#### 5. まとめ

本稿では、並列分散遺伝的アルゴリズムのモデルである踏み石モデル、非同期ランダムモデル、Sigma-Exchange モデルおよびサーバ・クライアント型モデルについて、実装面と性能面での比較を行った。

実装面では、Sigma-Exchange と比較し、簡素に実装できることが PAD からいえる。また、トポロジーやデッドロックの心配をする必要がなく、ユーザにとっては扱いやすいモデルであるといえる。

性能面では、他の手法と比べても性能的にも、時間的にも同等となっているのに対し、通信量ではかなりの削減に成功している。

つまり、サーバ・クライアントモデルは通信量が少なく、性能の劣化がなく、かつトポロジーやデッドロックの心配がない、非常に有効な手法である。

#### 参考文献

- [1] Erick C.: A Survey of Parallel Genetic Algorithms, IlliGAL Report No.97003, 1997
- [2] 棟朝, 高井, 佐藤: 集団分割型非同期並列遺伝的アルゴリズムにおける個体交換アルゴリズムの改良と評価, 情報科学技術フォーラム, Vol.35, No.9, pp.1815-1827, 1994