

F-036

大規模問題における階層型並列分散遺伝的アルゴリズムの影響についての考察 Consideration about the Effects of Layered Parallel Distributed Genetic Algorithm on a Large Problem

小嶋 和徳†
KOJIMA Kazunori

石亀 昌明†
ISHIGAME Masaaki

牧野 正三‡
MAKINO Shozo

1. はじめに

現在ある並列遺伝的アルゴリズムの研究では、大域並列化、粗粒度並列化、細粒度並列化などがあるが、その中で特に粗粒度並列化に関する研究が多くなされてきている。それらの研究の多くは同期的に移民を行う方法を使用し、比較的小規模な問題に対しての探索性能評価のみが行われ、大規模な問題に対しての考察がほとんど行われていなかった。

本研究では、並列分散遺伝的アルゴリズムにおけるトポロジや移民法に関する研究を行っており、エリート移民法や階層サーバ・クライアント型モデルについて、これまで閉数問題や巡回セールスマン問題 (TSP) の 30 都市, 100 都市に対して検討を行ってきた。本稿では、大規模な問題、ここでは特に TSP の 2392 都市に注目し、その問題における並列分散遺伝的アルゴリズムの効果を検討すると共に、階層サーバ・クライアント型モデルの効果について探索性能や移民信時間の面から考察する。

2. 並列モデルおよびトポロジ

並列遺伝的アルゴリズムは、大きく以下の 3 種類に分類される [1, 2]。

(1) 大域並列化; 1 つの母集団に対し、評価計算や遺伝的処理を分散処理させる。母集団は分割されないため、単純 GA と同様の性質を持つ。評価計算に非常に多くの時間を必要とする問題に適しているとされる。

(2) 粗粒度並列化; 母集団を複数の部分集団に分割し、複数のプロセッサに割り当てる。部分集団間では、定期的な移民により情報交換を行う。この方法は、単純 GA よりも良い性能が得られる場合が多いとされる。

(3) 細粒度並列化; 非常に小さな部分集団をプロセッサに割り当てる。最も細かいものでは、グリッド上に配置された各プロセッサに 1 個体を割り当て、近隣のプロセッサとのみ遺伝的操作が行われる。このモデルは非常に特殊な環境でのみ実装される。

これらの中で、本研究で対象とする並列モデルは粗粒度並列化、つまり集団を複数の部分集団に分割したモデルである。

また、本稿で検討するトポロジは以下の 3 種類である。

(1) リング型モデル (Ring); 図 1 の様に、部分集団をリング型に配置したモデルであり、このときの移民は一定世代間隔で同期的に行われる場合が多い。

(2) サーバ・クライアント型モデル (SC)[3]; 図 2 の様に、サーバを中心としてスター型にクライアントを配置したモデルで、各クライアントが部分集団となり GA を実行し、サーバは各クライアントのエリートを管理する。各クライアントはそれぞれ独立に GA を実行し、ク

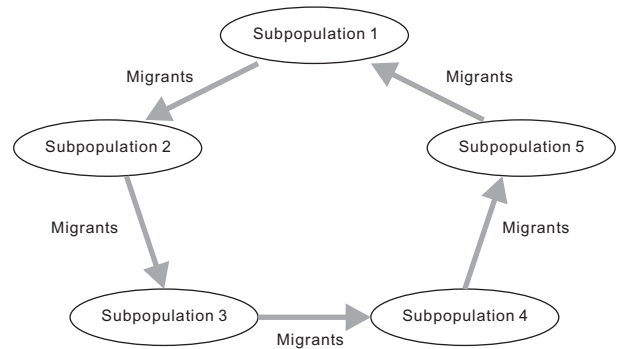


図 1: リング型モデル (Ring)

ライアントのエリートが更新されるとそのエリート情報をサーバに送信する。また、各クライアントのエリートが一定世代間更新されない場合、サーバが管理しているエリート情報の数個体分を受け取ることで移民を行う。

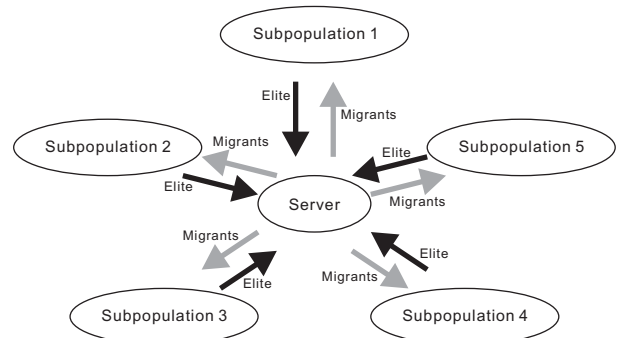


図 2: サーバ・クライアント型モデル (SC)

(3) 階層サーバ・クライアント型モデル [4]; 図 3 の様に、このモデルでは複数のクライアントを管理するサブサーバと、複数のサブサーバを管理するマスタサーバが存在する構造となる。各クライアント、サブサーバ、マスタサーバ間の通信は、サーバ・クライアント型モデルと同様に行われる。つまり、クライアントからサブサーバおよびサブサーバからマスタサーバへの通信は、エリートが更新されたタイミングで行われ、サブサーバからクライアントおよびマスタサーバからサブサーバへの通信は一定期間エリートが更新されないタイミングで行われる。

階層サーバ・クライアント型モデルは、以前の研究により、TSP30 都市および 100 都市の問題においてサブ

†岩手県立大学, Iwate Pref. Univ.

‡東北大学, Tohoku Univ.

サーバが管理するクライアントの数および通信頻度を調整することで、通信量を抑えながらサーバ・クライアント型モデルと同等の探索性能を示すことがわかってい

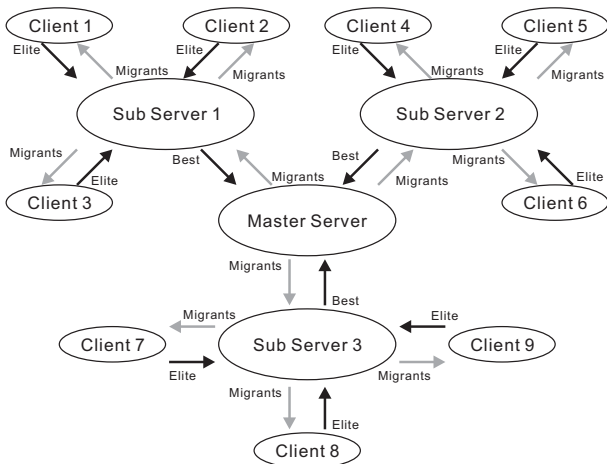


図 3: 階層サーバ・クライアント型モデル (LSC)

3. 実験

前節で挙げた 3 つのトポロジモデルを、SGI Altix350 (Intel Itanium2 1.6GHz, 16CPU) 上に C 言語および MPI を利用して実装し比較評価を行った。ここでの実験条件は以下の通りである。適用問題: TSPLIB, pr2392[5], 染色体: パス表現, 部分集団サイズ: 32, 部分集団数: 256, 選択: ルーレット選択法, 交叉: 60%, 部分一致交叉, 突然変異: 3%, 任意の 2 点間の逆順処理, 世代更新数: 100,000 世代, 試行回数: 1 回。

また、移民に関しては、移民率: 20%, 移民間隔: 10 世代。サーバ・クライアント型モデルでは、寿命: 50 世代, 移民数: 2, エリート送信間隔: 2 回に 1 回となっている。階層サーバ・クライアント型モデルでは、マスターサーバ数: 1, サブサーバ数: 4, 1 サブサーバ当たりのクライアント数: 64 という構造となっている。

表 1: 最大適合度, 試行時間, 移民時間の比較

	最大適合度	試行時間	移民時間
Ring	0.124	2:31:23	0:27:02
SC	0.638	2:28:07	0:24:04
LSC	0.631	2:10:31	0:07:52

表 1 は、それぞれのモデルの最大適合度 (最適値 1.0), 試行時間および移民時間 (時:分:秒) の比較である。試行時間および移民時間は、部分集団当たりの平均時間となっている。

Ring では、最大適合度がかなり低い値となった。これは、リング型では、部分集団数が多くなると局所的な情報交換しか行われなくなり、100,000 世代更新でもエリート情報が全体に伝わりにくくなるためであると考え

られる。それに対し、SC および LSC では Ring と比較してかなり高い適合度を得られた。これら 2 つでは、到達度としてはまだ高い値とは言えないものの、エリート情報を有効に使うことで全体の適合度の引き上げに成功し、特に規模の大きな問題に対しての有効性が確認できた。

移民時間に関しては、Ring では移民処理が同期的に行われるため、隣の部分集団からの移民を待つ必要がある。部分集団数が多くなると待つ時間が多くなり、結果的に移民に多くの時間が必要となる。

SC では通信は非同期的に行われるため、Ring の様に隣の部分集団の移民を待つ必要はないが、256 もの部分集団の通信がサーバに集中するため、サーバからの応答に時間がかかり、Ring よりは移民時間が少ないが、それでも多くの時間を必要とする。

それに対し LSC では、サーバにかかる負荷を分散させたことで、部分集団当たりの移民時間をかなり短縮できており、かつ適合度の悪化はほとんど見られなかった。

なお、試行時間に関しては移民時間の差がほぼそのまま時間差として現れていることから、移民以外の部分による差はないと考えられる。

以上のことから、階層サーバ・クライアント型モデルは、大規模な問題に対して有効なモデルであることが確認できた。

4. まとめ

本稿では、並列分散遺伝的アルゴリズムにおいて、大規模な問題、特に巡回セールスマン問題の 2392 都市に対し、リング型モデル、サーバ・クライアント型モデル、階層サーバ・クライアント型モデルを使用した場合の比較を行った。その結果、リング型モデルでは、問題の大規模化による探索性能の低下および通信時間の増大が確認できた。また、サーバ・クライアント型モデルでは、探索性能についてはリング型モデルよりは良かったものの、クライアント数が多いことによる通信時間の増大が確認できた。それに対し、階層サーバ・クライアント型モデルでは、最大適合度はサーバ・クライアント型モデルと同等程度でありながら、移民時間をかなり短縮できていることが確認でき、このモデルの大規模な問題での有効性が確認できた。

参考文献

- [1] Erick C.: A Survey of Parallel Genetic Algorithms, IliGAL 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, 情報処論文誌, Vol.35, No.9, pp.1815-1827, 1994
- [3] 小嶋, 石亀, 松尾: 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, FIT2003 一般講演論文集第 2 分冊 G-013, pp.297-298, 2003
- [4] 小嶋, 石亀: 階層型サーバ・クライアントモデルによる非同期並列分散 GA の実装, 平成 16 年度電気関係学会東北支部連合大会, 1E10, p.171, 2004
- [5] TSPLIB: <http://elib.zib.de/pub/Packages/mp-testdata/tsp/tsplib/tsplib.html>