

ネットワーク性能に合わせた分散遺伝的アルゴリズムにおける最適な移住についての検討

Examination about optimal migration on DGA considering network performance

斉藤 宏樹

Hiroki Saito

Abstract: A migration is an important parameter in DGA. It's parameter depends on a problem. It is difficult to set it as the optimal value beforehand. A migration parameter is decided experientially. We cannot use a resource effectively. We examine about optimal migration not algorithm but network performance of a parallel computer.

1 はじめに

最適化手法の一つである遺伝的アルゴリズム (GA : Genetic Algorithm) は高い並列性を有している。そのため、その並列性を利用して GA を改良し、より有効な解を求めることを目標とした並列遺伝的アルゴリズム (PGA : Parallel Genetic Algorithm) の研究が行われている。本研究では PGA の粗粒度モデルの代表である分散遺伝的アルゴリズム (DGA : Distributed Genetic Algorithm) に注目した。DGA における重要なパラメータとして移住がある。最適な移住パラメータは問題に依存し、経験的に決定されるが、そのようなパラメータでは並列計算機で実行する際に、リソースの有効利用を考慮しているとはいえない。本研究ではアルゴリズムからではなく、並列計算機のネットワーク性能に合わせた最適な移住についての検討を行う。

2 DGA

DGA は母集団を複数の分割母集団 (島) に分割し、各島内で遺伝的操作を行う GA の並列モデルの一つである。DGA は移住という操作によって各島間で個体の交換が行われる。移住は数世代に一度、各島内で選ばれた 1 つまたは複数個の個体を別の島と交換し、解の多様性を維持することで探索性能を向上させることが目的である。各島内における移住個体の割合を移住率、何世代おきに移住するかを移住間隔と呼ぶ。移住相手となる島及び、移住を行う個体の選択方法にはさまざまな方法が考えられているが、本研究では移住相手の選択方法にランダムリングを、移住個体の選択にはランダムに選択する方法を採用した。

3 DGA の数値実験

3.1 実験マシンの概要

本研究では DGA の性能を測定するマシンとして、Gregor と Maia を用いた。Gregor では最大 40 プロセッサを

使用し、Maia では最大 16 プロセッサを使用した。Gregor と Maia の仕様を Table 1, Table 2 に示す。Gregor の通信には Myrinet と Ethernet を利用する。Myrinet では、独自の通信プロトコルを用いる低遅延ソフトウェアの GM を利用し、Ethernet, SP スイッチでは通信プロトコルに TCP/IP を用いる。Gregor では通信ライブラリに MPI の LAM を用いて、Maia では MPICH を用いて DGA を実装する。

Table 1 Gregor の性能

CPU	Pentium 1GHz × 64 × 2
Memory	512M × 64 (計 32GB)
OS	Kondara MNU/Linux
通信ライブラリ	MPI, PVM
通信プロトコル	GM, TCP/IP
通信媒体	Myrinet 2000, Fast Ethernet

Table 2 Maia の性能

CPU	POWER3- 375MHz × 16
Memory	4GB × 4 (計 16GB)
OS	AIX
通信ライブラリ	IBM MPI for AIX
通信プロトコル	US プロトコル, TCP/IP
通信媒体	SP スイッチ, Fast Ethernet

3.2 実験方法

実験は、DGA の実行時間と移住に要した通信時間を計測し、並列計算機のリソースが有効利用されているかを検討する。テスト関数に Ridge 関数を用いる。Ridge 関数は単峰性であり設計変数間に依存関係がある。本実験においては、20 次元の Ridge 関数を用いる。

$$F_{Ridge}(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2 \quad (1)$$

$$(-64 \leq x_i < 64)$$

$$\min(F_{Ridge}(x)) = F(0, 0, \dots, 0) = 0$$

各 DGA のパラメータを Table 3 に示す .

総個体数	480
遺伝子長	200
設計変数	20
移住率	0.5
移住間隔	1
交叉	一点交差
交叉率	1.0
突然変異率	0.005
選択	トーナメント選択
トーナメントサイズ	4
エリート保存	1
終了世代	1000
試行回数	20

3.3 実験結果

移住間隔が 1 と 5 の場合において , プロセス数を変化させた場合の実行時間の変化を Fig. 1 , Fig. 2 に示し , そのときの実行時間に対する移住時間の割合を Fig. 3 , Fig. 4 に示す . 実行時間は 20 回試行の平均値で示している .

Fig. 1 , Fig. 2 より , Myrinet , SP スイッチは Ethernet よりも実行時間が短いことが確認できる . これは通信速度の違いと通信のオーバーヘッドの違いが原因である . Myrinet , SP スイッチは高速スイッチを用いており , 通信データの衝突や送受信待ちが少ない . Ethernet ではスイッチングハブを用いており , 低機能であるため通信データの衝突や送受信待ちが多い . また , Myrinet では GM による独自のプロトコルを用いており , Ethernet で用いた TCP/IP よりもオーバーヘッドが小さい . 通信プロトコルに TCP/IP を用いると , パケットのデータをカーネルバッファ領域を利用してコピーするため , データコピーの処理に多くの時間を費す . Myrinet では GM による独自のプロトコルを用いており , このようなオーバーヘッドを軽減できる .

Ethernet において移住間隔 1 の場合 , Fig. 1 から Gregor においては 20 プロセスから , Fig. 2 から Maia では 8 プロセスから , 並列化効率が悪化していることが確認できる . Fig. 3 , Fig. 4 からは , Ethernet を用いた場合に , 実行時間に対して移住による通信時間の占める割合が大きいことが確認できる . つまり通信に時間を消費しており , CPU は通信の待ち時間のためにリソースを有効に利用できていないことがわかる .

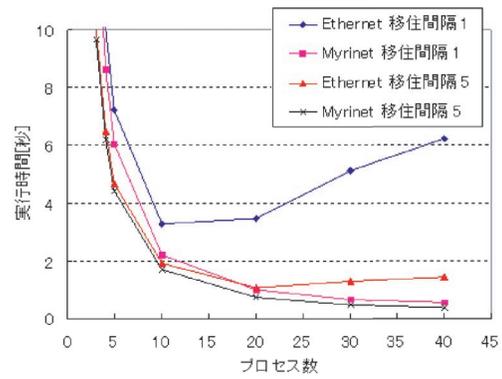


Fig. 1 プロセス数と実行時間 (Gregor)

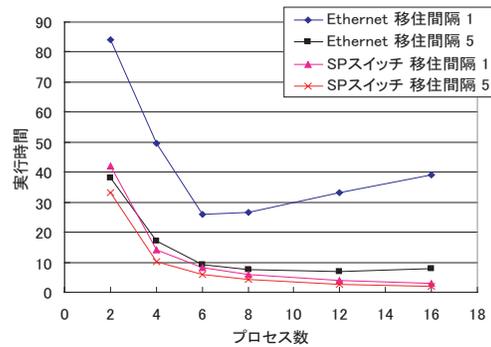


Fig. 2 プロセス数と実行時間 (Maia)

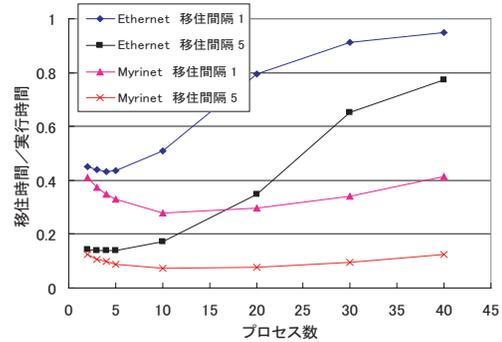


Fig. 3 実行時間と移住時間の割合 (Gregor)

3.4 実験結果の考察

実験結果から通信を要する移住は , 並列計算機のネットワークの性能によって , その処理時間が大きく左右されることがわかる . そのためネットワークの性能に適した移住間隔を設定しなければ , 並列計算機のリソースを有効利用できなくなる . これまでは , 解の性能に重点をおいた移住間隔が経験的に設定されていたが , Grid などの通信が常に考慮される環境においては , リソースの有効利用は重要な問題となる . そこで本研究では , 並列

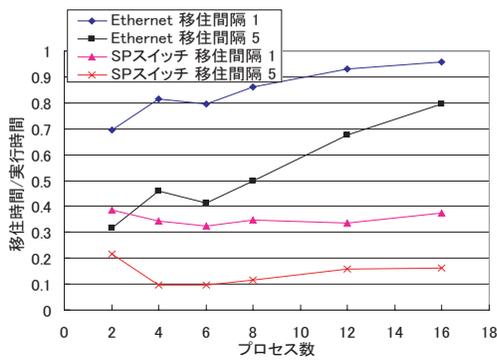


Fig. 4 実行時間と移行時間の割合 (Maia)

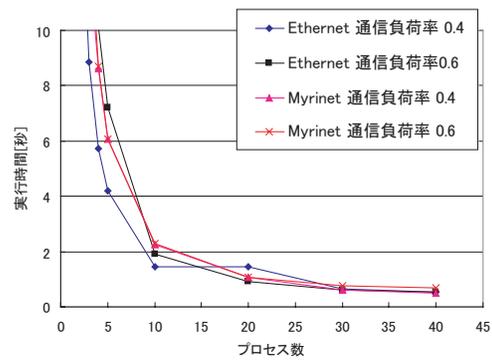


Fig. 5 プロセス数と実行時間 (平均値)

計算機のリソースの有効利用に重点をおいた移住の検討を行うこととした。

4 リソースの有効利用に重点をおいた移住の検討

4.1 実験方法

実験は、DGA の 1 世代の処理に要する時間と移住に要する時間を計測し、1 世代の処理に対して移住の占める割合を計算する。この割合を通信負荷率と呼ぶことにする。通信負荷率に応じて実行中に移住間隔を変化させることで、並列計算機のリソースを有効に使用できるかどうかの検討を行う。また、移住間隔が一定の DGA と解の性能を比較する。実験は Gregor のみで行う。テスト関数には前節同様 Ridge 関数を用いる。DGA の実装、パラメータにおいても同様である。通信負荷率が 0.4、0.6 より大きくなった場合に、移住間隔を 2 倍にするという実装において、それぞれ実験を行った。初期の移住間隔は 1 である。解の性能については、適合度が 1 以下になるまでの実行時間で検討する。移住間隔を経験的に定めた DGA と、通信負荷率により移住間隔が変化する DGA の実行時間において、解の性能を比較をする。

4.2 実験結果

プロセス数を変化させた場合の実行時間の結果を Fig. 5 に示す。Fig. 5 より Ethernet, Myrinet ともにプロセス数が増加するに従って、実行時間が短くなっていることがわかる。これより並列計算機のリソースが有効に利用できていることがわかる。

また、解の性能比較の結果を Fig. 6, Fig. 7 に示す。Fig. 6 より Ethernet の場合においては、プロセス数が 20 以上になると、通信負荷率により移住間隔が変化する DGA の方が解性能が良い。Fig. 7 により Myrinet の場合においては、移住間隔を固定した DGA の方が解性能が良い。つまり、通信負荷が高くなるネットワークにおいては、移住間隔を通信負荷率に応じて変えるほうが

良いが、ネットワークの負荷が高くない場合には、ある通信負荷率で変更しない方が良いことになる。

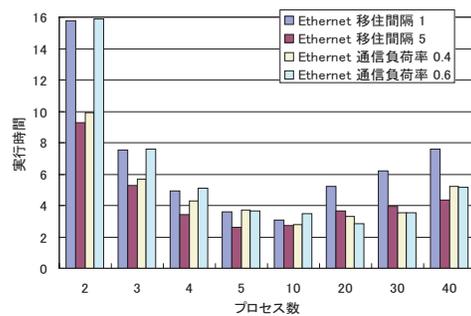


Fig. 6 解の性能比較 (Ethernet)

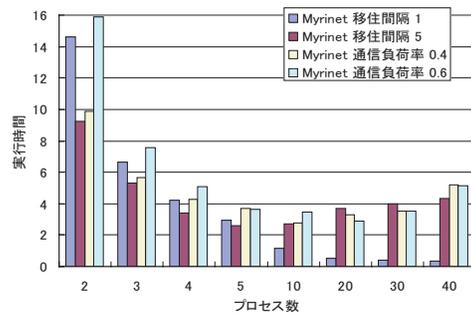


Fig. 7 解の性能比較 (Myrinet)

5 まとめ

数値実験の結果、移住間隔を通信負荷率によって実行中に変化させる方法は、通信負荷のかかる性能の悪いネットワーク上では有効であることがわかった。しかし、ネットワーク性能が良い場合は、解の性能が多少悪くなることが確認できた。これは、移住間隔を変更させる条件の設定が単純であったためであると考えられる。設定方法を改良すれば、解性能が良くなると考えられる。