ヘテロ環境における DGA の性能評価

Evaluation of DGA on the Hetero Environment

片浦 哲平
Teppei KATAURA

Abstract: Because the Homo system is balanced, it is easy to synchronise. Although the Heteo system lacks of balances, it is essential to realize the global computing system.
In this paper, I have tested the performance of Hetero Distributed Genetic Algorithm(DGA) and report results.

1　はじめに

Genetic Algorithm(以下 GA) は多点探索のため膨大な計算時間を必要とする．この問題を解決するためにの並列モデルのひとつに Distributed GA(以下 DGA) がある．DGA において母集団は複数のサブ母集団 (以下島) に分割され，各島ごとにプロセッサが割り当てられそれぞれ独自に DGA が実行される．プロセッサ間の通信は移住の時のみに限られるのでクラスタシステムなどのような低速なネットワーク下でも高い分散効果が期待される．　一方で，大量の処理が必要な場合にはグローバルコンピューティングの分野が注目をあびている．グローバルコンピューティング環境で GA を実行する場合に問題となると考えられるのが，各サイトごとの処理速度の違いである．クラスタなどでは，各プロセッサの処理速度が一定であるため，移住の際に同期を簡単にとることができるが，グローバルコンピューティング環境においては，サイトごとの処理速度が異なるため，各プロセッサごとの計算速度に偏りが生じ，移住の際の同期も難しくなる．

本報告ではグローバルコンピューティング環境における並列 GA について検討する．

2　ヘテロな環境における DGA

ヘテロな環境における DGA とは，各島ごとに計算速度が異なる DGA のことをいう．これに対し均質な環境の DGA をホモな環境の DGA という．ヘテロな環境の DGA では，計算速度の違いから，移住の際の同期や，遅い島の個体の扱いなどホモな環境の DGA に比較してさらにパラメータの設定等が困難になる．

3　数値実験方法

処理速度の均質なクラスタシステムにおいてヘテロな環境下の DGA を作成するために Fig. 1 ように計算実行世代と待機世代を作り，プログラム上でヘテロな環境を作成した．

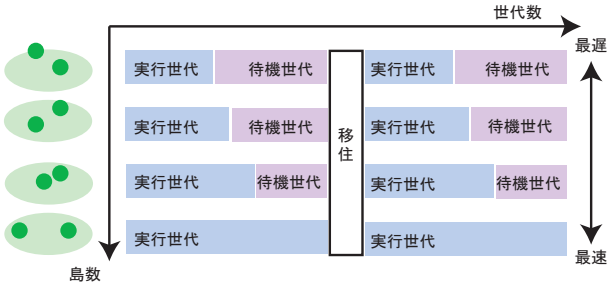

Fig. 1　ヘテロな環境の実装

ヘテロな環境の DGA は，まず初期パラメータで島の移住を行うまでの世代数を定める．各島の移住までの世代数を比較して移住世代数の最も大きいものを基準にして，移住を行う．従って，最も大きい値の移住世代数から各島の移住世代数を引いた世代数が待機状態の世代数となり，待機状態の多い島ほど遅い島となる．移住の際の同期に関しては，前述の方法の仕様から，同期を合わせてとる形になっていて，本来のヘテロ環境で必要な非同期な通信は行っていない．

4　ヘテロな環境の DGA の性能評価

4.1　SGA との性能比較

ヘテロな環境の DGA は各島ごとに計算世代数が異なるので全島からの最適解の履歴を表示するには移住で同期をとった世代に限られる．しかし，移住の世代ごとの履歴は他の手法の世代数の履歴と評価計算回数が異なるため比較することができない．このため，その代わりとして計算に要した時間で履歴を表示することにした．時間は最も計算速度の速い島を基準にして，遅い島は時間軸をスケーリングして比較できるようにした．

まず，ヘテロな環境の DGA の遅い島に解探索の効果があるのかを調べた．比較する方法として，次の 4 種類の測定を行った．

- 処理速度の異なる 4 島で実行
- 最も遅い島を消して 3 島で実行
- 遅い 2 島を消して 2 島で実行
- 最も速い 1 島だけで実行

4 番目の方法は個体数が 4 分の 1 になった SGA ということになる．初期パラメータを Table 1 に，各島の実行状態，待機状態を Table 2 に，rastrigin 関数の実行結果を Fig. 2 に，griewank 関数の実行結果を Fig. 3 に示す．

Table 1　初期パラメータ

世代数	最速島が 2000 世代
総個体数	400
交叉率	1.0
突然変異率	1/L
遺伝子長	400
設計変数	20
島数	4
移住率	0.5
移住個体	50
目的関数	rastrigin, griewank
移住トポロジー	ランダムリング
試行回数	5

Table 2　実行条件

	島 1	2	3	4	移住間隔
実行 1	35, 0	25, 10	15, 20	5, 30	35
実行 2	25, 0	15, 10	5, 20	–	25
実行 3	15, 0	5, 10	–	–	15
実行 4	–	–	–	–	–

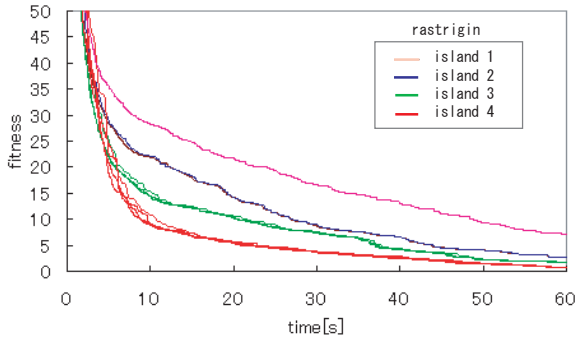

Fig. 2　ヘテロな環境の DGA の効果:rastrigin

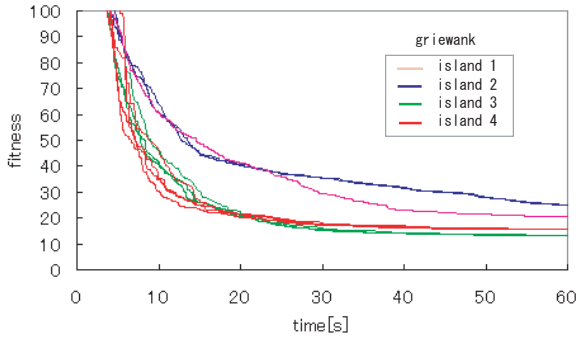

Fig. 3　ヘテロな環境の DGA の効果:griewank

Fig. 2，Fig. 3 から，遅い島も移住に加わることで解探索に影響を与えているということが分かった．特に rastrigin 関数において効果が顕著であった．

4.2　島ごとの処理速度差を大きくした場合

次に上記の結果から，ヘテロな環境の DGA において，遅い島がどのような影響を与えるかについて調べた．初期パラメータを Table 3 に，実行状態，待機状態を Table 4 に示し，実行結果を Fig. 4〜Fig. 6 に示す．

Table 3　初期パラメータ

総世代数	5000
総個体数	400
交叉率	1.0
突然変異率	1/L
遺伝子長	400
設計変数	20
島数	4
移住率	0.1
エリート個体	5
移住エリート個体	5
移住トポロジー	ランダムリング
試行回数	10

Table 4　各島の状態

	実行世代数	待機世代数
island 1	35	0
island 2	25	10
island 3	15	20
island 4	5	30

実行結果から rastrigin 関数，ridge 関数で移住間隔差が少ない条件のものが良好な解を得た．ヘテロな環境の

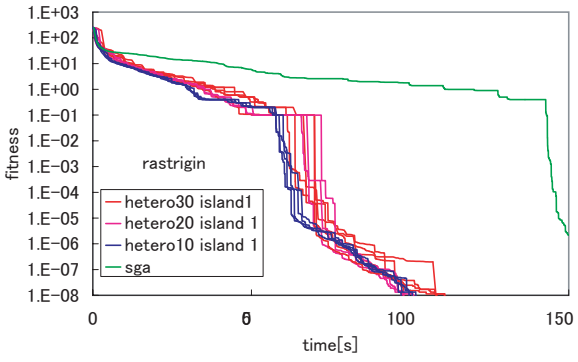

Fig. 4　rastrigin 関数の実行結果

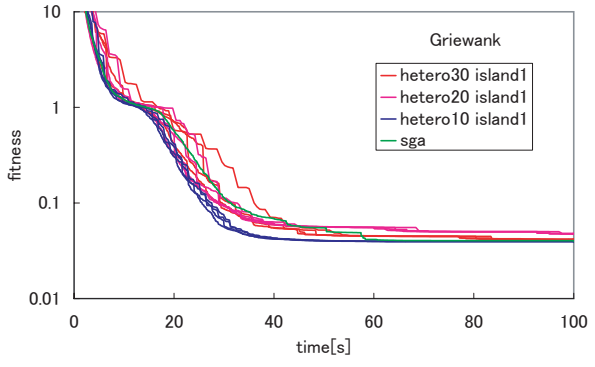

Fig. 5　griewank 関数の実行結果

DGA の中で比較すると，rastrigin 関数では，移住間隔差が少ない方が速く収束しているため遅い島も解探索に影響を及ぼしているが，ある程度収束してしまうと，解の改善が行われにくくなり，移住間隔差のあるヘテロな環境の DGA とあまり精度は変わらなくなった．このことから，解探索の進んだ後半は，逆に頻繁に遅い島から移住が行われることで解探索の妨げになってと考えられる．ridge 関数は，単峰性の関数であるため，島の数がそのまま解の精度に影響を及ぼしているようである．griewank 関数では，SGA とほとんど変化は見られなかった．

4.3　ホモな環境の DGA との性能比較

次に，ホモな環境の DGA に対して性能にどのような違いがあるかを調べた．ホモな環境の DGA とヘテロな環境の DGA は処理速度の違いから単純な履歴では比較をすることができない．そこで，Table 5 のように終了世代を調整することで計算量と情報交換量を等しくした．また，時間とリソース (個体数×世代数) を比較するために，適合度にしきい値を定めて，適合度がしきい値に到達した時間とその時の世代数の合計を求めることでヘテロな環境の DGA とホモな環境の DGA の時間とリソースを比較した．また，その際の最終的な適合度を表し性能を比較した．初期パラメータを Table 6，実行結果を Fig. 7〜Fig. 10 に示す．

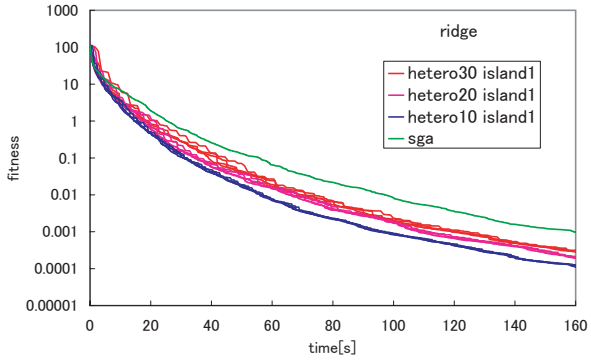

Fig. 6　rigde 関数の実行結果

Table 5　実行条件

	島 1	2	3	4	移住間隔	終了世代
実行 1	5, 0	5, 0	5, 0	5, 0	5	6250
実行 2	6, 0	6, 0	6, 0	2, 4	6	7500
実行 3	8, 0	6, 2	4, 4	2, 6	8	10000
実行 4	9, 0	9, 0	1, 8	1, 8	9	11250

Table 6　初期パラメータ

総世代数	25000
総個体数	100
交叉率	1.0
突然変異率	1/L
遺伝子長	400
設計変数	20
島数	4
移住率	0.04
エリート個体	1
移住エリート個体	1
移住トポロジー	ランダムリング
試行回数	20

Fig. 7 の結果から，rastrigin 関数ではしきい値をクリアしたリソースは概ね各条件の処理速度通りとなった．また，Fig. 8 の結果から griewank 関数では処理速度の差が大きいほど良い結果が得られなかった．この結果は Fig. 5 の結果からも言えることである．設計変数間に依存関係のある関数は，世代後半は解の探索が進むかどうかは交叉，突然変異の確率に大きく依存してしまうので，個体にバラツキの出るヘテロな環境の DGA の方が，良

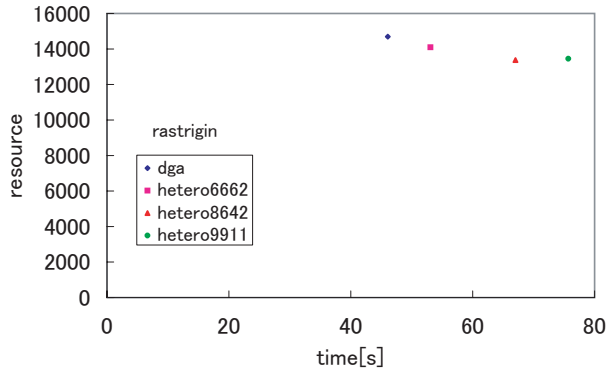

Fig. 7　rastrigin 関数の時間とリソース

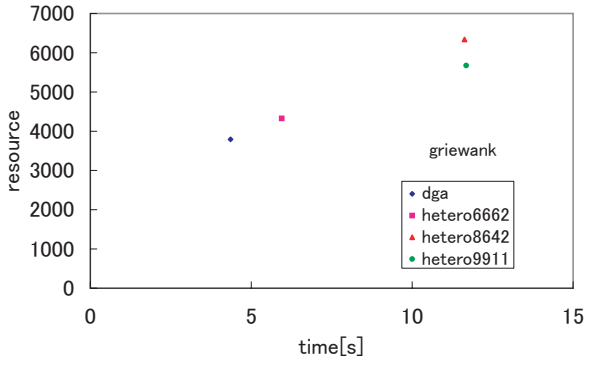

Fig. 8　griewank 関数の時間とリソース

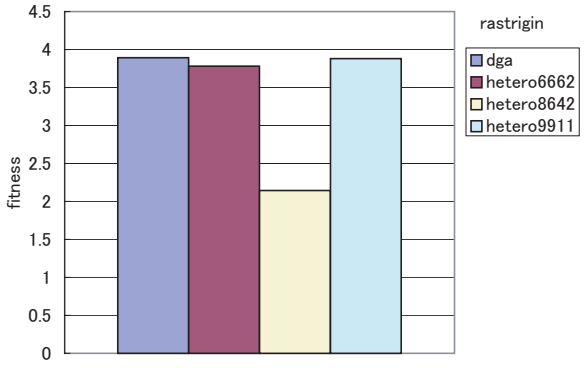

Fig. 9　rastrigin 関数の適合度

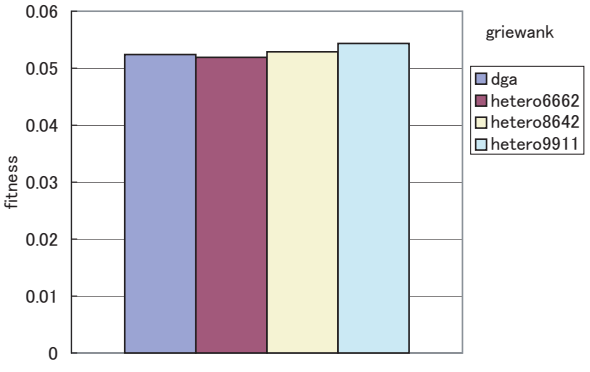

Fig. 10　griewank 関数の適合度

い結果を得られると思われるが，griewank 関数では設計変数を増やすと依存関係が弱くなる傾向にあるので，設計変数を減らして実行した場合を計測する必要があると考えられる．

Fig. 9 の結果から，ヘテロな環境の DGA の方が良好な解を得られた．設計変数が増え，後半世代で解の探索が進みにくくなる場合には解のバラツキは重要な要素のようである．Fig. 10 の結果から最終的な精度はどの方法でもほとんど違いは見られなかった．Fig. 8 の結果を考えると，ヘテロな環境の DGA は後半世代で解探索が進んだと言える．

5　結論と今後の課題

ヘテロな環境の DGA における処理速度の遅い島は SGA と性能比較を行うことによって効果があるということが分かった．また，ヘテロな環境の DGA はホモな環境の DGA と比較してリソースの面ではほぼ同等の性能があると分かった．しかし，ヘテロな環境の DGA とホモな環境の DGA を同じパラメータで実行したならば，実行時間は確実に遅くなることも分かった．今後はヘテロな環境の DGA に適応した移住モデルを検討，具体的には，無理に同期を合わせるような移住方法ではなく，非同期の移住モデルを考え，それに合わせて移住トポロジーをランダムリングから他の方法で検討したいと考えている．

参考文献

1) 畠中一幸『遺伝的アルゴリズムの分散並列化』（同志社大学卒業論文，1998）

2) 吉田純一『GENEsYs と分散遺伝的アルゴリズムの性能比較』（同志社大学研究報告資料，2001）

3) 上浦二郎『分散遺伝的アルゴリズムにおけるパラメータの検討　第 1 報：母集団内パラメータの解探索能力への影響』（理工研報告書，2001）

4) 上浦二郎『分散遺伝的アルゴリズムにおけるパラメータの検討　第 2 報：移住に関連するパラメータの検討』（理工研報告書，2001）

5) 金子美華『分散 GA における解探索能力』（人工知能学会全国大会，1999）