

ランダム・ヘテロジニアスなパラメータ設定を用いた 島型遺伝的アルゴリズムの性能評価

猪鼻 真裕[†], 福永 Alex^{††}

東京大学教養学部[†], 東京大学大学院総合文化研究科^{††}

1 はじめに

遺伝的アルゴリズム (GA; Genetic Algorithm) による実数値関数最適化問題は, 工学的な目的に広く応用される重要な研究分野である. 評価関数が具体的に書けず, 多大な実行時間を必要とするシミュレーションを行わなければならない場合などは, 実行を並列化し計算速度を高める工夫が必要になる. GAは並列化に適したアルゴリズムであり, 並列化遺伝的アルゴリズムについて多くの研究が行われてきた. 並列化GAは大きく4つの種類に分類される. マスター・スレイブモデル, セルラーモデル (cellar model, fine-grained model), 島型モデル (Islands model, Distributed model, course-grained model), 階層モデルである¹⁾.

近年, クラウドコンピューティングやP2Pなどの並列化環境がますます整ってきており, 大規模並列化が容易になってきている. 島型モデルはこの環境で並列化するのに特に適したモデルであり, 最近, このモデルに関して多くの研究が行われている.

大規模並列化に際して, 各プロセッサ上でどのようなアルゴリズムを実行するのかを決めることは, 重要な問題である. 一般に, 複数のアルゴリズムを組み合わせるポートフォリオ戦略において, 挙動の異なるアルゴリズム同士を組み合わせることによって更なる性能向上を図ることができる⁴⁾. また, NFL定理¹⁰⁾により, どんな2つのアルゴリズムも全てのあらゆる問題を考慮するとその性能の期待値は同一である. すなわち, それぞれのアルゴリズムに得意・不得意な問題が考えられるため, 複数の異なるアルゴリズム

を組み合わせることがポートフォリオ戦略において重要である. 単一のアルゴリズムを単に並列化したものをHomogeneous環境, 複数のアルゴリズムを組み合わせる並列化したものをHeterogeneous環境と呼ぶ.

最近の様々なアルゴリズムに対する研究からHeterogeneousなポートフォリオ戦略の有効性が実証されたが^{7) 11) 2)}, その要素となるアルゴリズムの選び方は, 恣意的であるという問題点がある. 実数関数最適化に対しては, DEやPSO, ニッチングアルゴリズムなどのそれに特化したアルゴリズムやパラメータ設定を選べばよいが, 一般に, 扱う問題がブラックボックスである実世界の問題などでは, どのアルゴリズムを要素として選べばよいかが明確ではない. すなわち, 事前の知識が全くない状況では, これらの手法を適用するのは難しい. 探索中にオンラインでパラメータを調節する適応的なGAも多く研究されているが, 適応に用いるパラメータ値などの設定は試行錯誤しなければならず, 本質的な煩雑さは解消されていない.

それに対して, シンプルなGAの枠組みのみを用いた場合のHeterogeneous環境の有効性を調べる研究も行われている. Miki, et. al. は, 島型GAにおいて, まず突然変異率や交叉率の最適値が, 集団サイズや扱う問題, 島の数などによって異なることを示した⁶⁾. そして, それら複数のパラメータ設定を組み合わせた島モデルの有効性を実証した. しかし, パラメータ値の組み合わせは離散的であり, bestなチューニングがなされているとは言えない. さらに, 扱った問題も限定的であり, 並列化数も $N = 9$ であった.

Gong and Fukunaga は, 組み合わせるパラメータ値を連続的に, ランダムに選ぶランダム・ヘテロジニアスGAを提案し, $N = 100$ までの並列化を試しその有効性を実証した³⁾. これは, 要素となるアルゴリズムを選択するポートフォリオ戦略

Performance Evaluation of Random Heterogeneous Island-Model Distributed Genetic Algorithms

[†] Masahiro Inohana (line.craft.1087@gmail.com)

^{††} Alex Fukunaga (fukunaga@idea.c.u-tokyo.ac.jp)

Department of General Systems Studies, Tokyo University

(†)

Graduate School of Arts and Sciences, Tokyo University (††)

とは対極にある考え方であり、ポートフォリオ作成の手間と性能のトレードオフを目指した枠組みである。さらに重要なことは、これらのアルゴリズムはパラメータフリーなアルゴリズムであることである。一般に、あるアルゴリズムを採用する場合、そのパラメータは適切な値に設定しなければ探索はうまくいかないが、上記の枠組みはその煩雑さを本質的に解消している。

しかし、これらのアルゴリズムに関して、その挙動が十分に理解されているとは言い難い。例えば、島型GAの探索性能に重要な影響を与えるものに migration が挙げられるが、その影響は十分に調べられていない。migrationは探索性能を大幅に向上させると考えられているが⁵⁾、一概にそうとは言えない。Skolicki and De Jong は、島型GAにおける効果的な資源割り当てを、島内の多様性・島間の多様性という2つの観点から考察し、そのことによって migration の有効性を浮き彫りにした⁸⁾。島の数を増やすと migration によって島同士の情報交換が多くなされるが、問題に依ってはそれが負の効果をもたらすことがあることを、人工的な関数を用いて実証した。池田らは、問題の構造としてUV構造を提案し、同様に migration によって探索性能がおちる現象を指摘している¹²⁾。

本研究では、ランダム・ヘテロジニアスな島型GAにおいて、migrationの効果が探索性能に与える影響を、migrationを行う場合と行わない場合で比較し、検証する。さらに、新たに narrow Heterogeneous環境を提案し、migrationの効果を有効に用いる枠組みを検討する。また、 $N = 128$ まで並列化した際の挙動を調べ、narrow Heterogeneousの大規模並列化に際する挙動の変化を調査する。そして、パラメータフリーのアルゴリズムである narrow Heterogeneous GAのロバスト性を、代表的なパラメータ設定を用いた Homogeneous GAとの比較により検証する。

2 ランダム・ヘテロジニアスGA

先行研究³⁾ にならって、ランダム・ヘテロジニアスGAの実装を行った。トップレベルの擬似コードを Algorithm1 に示す。

実数関数値最適化問題に対しては実数値を遺伝型として用いる実数値GAも多く用いられているが⁹⁾、最もシンプルな実装としてバイナリコーディングを用いた。突然変異は各bitに対して突

Algorithm 1 Top level pseudo-code

```

int  $N$  // the number of islands
Initialize  $N$  islands
while termination criterion is false do
  for  $i = 1$  to  $N$  do
    Evolve_1step(island[ $i$ ])
  end for
  for  $i = 1$  to  $N$  do
    if island[ $i$ ] updated local best then
      Migrate local-elite to neighbor of island[ $i$ ]
    end if
  end for
  Check for terminal criterion
end while

```

然変異率 μ でビットを反転を行い、交叉は交叉率 c で2個体を用いた一点交叉とした。親の選択にはルーレット選択、淘汰にはエリート選択を用いた。世代交代モデルはsteady-stateとした。これにより、各島に均等に資源を分配し、探索を並列に行う環境をシミュレートできるようにした。この部分の擬似コードを Algorithm2 に示す。

Algorithm 2 Evolve_1step

```

for  $i = 1$  to 2 do
  parent[ $j$ ]  $\leftarrow$  choose_parent(island[ $i$ ]) // using
  roulette selection
end for
if Random[0,1]  $\leq$   $Prob_{crossover}$  then
  ( $child[1], child[2]$ )
   $\leftarrow$  Cross(parent[1], parent[2])
end if
for  $i = 1$  to 2 do
  if Random[0,1]  $\leq$   $Prob_{mutation}$  then
    Mutate(child[ $j$ ])
  end if
end for
Evaluate children
Reproduce(island[ $i$ ])

```

migrationは先行研究³⁾と同様に最良個体が更新されたときに行うものとした。グラフのトポロジには、双方向リングを用いた。以上の設定によって、決定すべきパラメータは集団サイズ p 、突然変異率 μ 、交叉率 c の3つのみとなる。migrationのタイミングを上記の設定にすること

表 1 本実験で使用したテスト関数

function	Name	features
f_1	Sphere	separable, unimodal
f_2	Griewank	separable, multimodal
f_3	Rastrigin	separable, multimodal
f_4	Schwefel	separable, multimodal
f_5	Rosenbrock	non-separable, multimodal
f_6	Schwefel-1.2	non-separable, unimodal

で, migrationの周期パラメータを削減している。ランダム・ヘテロジニアスGAでは, 島の初期生成の際に, パラメータをランダムに設定する。先行研究では, $p \in [10, 100]$, $\mu \in [0.0, 1.0]$, $c \in [0.0, 1.0]$ から一様に選んだ。しかし, 一般的に有効と考えられている突然変異率や交叉率と比べて, この設定では明らかに探索性能が悪い島が生成される確率が高い。特に突然変異率の推奨値はビット長 L に対して $\mu = 1/L$ といわれているが, 今回の設定では後述するように $1/L = 0.01$ となるので, 最適ではない突然変異率を持った島が生成される確率が高い。よって今回は, 一般によいとされているパラメータ値を含むような幅を取って, $p \in [10, 120]$, $\mu \in [0.0, 0.02](= [0.0, 2/L])$, $c \in [0.3, 1.0]$ から一様に選んでくる設定を提案する。前者を broad Heterogeneous, 後者を narrow Heterogeneous と呼んで区別する。これを, 各問題に対して最適なパラメータ値を用いたもの (best Homogeneous) と比較することによって, 探索性能を評価する。

評価関数は多くの研究で用いられているベンチマーク問題から6題を選び, 全て10次元とした。関数の名前と特徴を, 表1にまとめる。

3 Homogeneous環境における最良設定

まずは, 各問題に対する best なパラメータチューニングを求める実験を行った。各問題に対して一定の基準を設け, その基準に達したら探索成功とみなし, 探索を終了する。終了までに評価回数を評価した回数 (NFE; Number of Fitness Evaluation) で探索性能を評価する。NFEが小さいほど, よい探索性能をもつとみなす。

表 2 実験により求めた各関数の最良パラメータ設定

function	pop-size	mut-rate	cross-rate	ave-NFE	rank
f_1	10	0.02	1.0	343.6	best
	10	0.04	1.0	348.6	second
f_2	100	0.02	1.0	1512.8	best
	50	0.02	0.7	1523.4	second
f_3	120	0.01	0.8	2983	best
	50	0.04	1.0	3248.8	second
f_4	25	0.01	1.0	2303.3	best
	25	0.005	1.0	2378.2	second
f_5	50	0.04	1.0	3554.4	best
	120	0.02	1.0	5069.4	second
f_6	120	0.02	1.0	3051.2	best
	120	0.04	1.0	3329.8	second

実験は, $p \in [10, 25, 50, 75, 100, 120]$, $\mu \in [0.005, 0.01, 0.02, 0.04, 0.1, 0.3, 0.6, 0.8, 1.0]$, $c \in [0.1, 0.3, 0.6, 0.7, 0.8, 1.0]$ のそれぞれから全通りの組み合わせを試し, 各設定で10回ずつ実験を行い, そのNFEの平均でbestを決めた。先行研究でもbestとの比較を行っているが, 200回のランダム設定のうち, 最もよいものをbestとしていた³⁾。今回は, 各関数ごとに $6 \times 9 \times 6 \times 10 = 3240$ 回の実験を行いbestを求めたので, より精度よくチューニングできていると考えられる。

実験結果のうち, 上位2つのパラメータ設定を表2にまとめる。各関数において, 表2の1段目の設定をbestとし, 以降の実験で用いることとする。

結果を見ると, 突然変異率と交叉率については一定の傾向が見られる。今回の設定のGAではどの関数においても, $\mu = 0.02$, $c = 1.0$ 付近に最適値が存在する。逆に集団サイズについては関数ごとにばらつきが見られる。Sphere関数(f_1)のようになめらかで局所解が1つしかない簡単な関数では, 集団サイズを小さくして集中的に探索を行ったほうがよいことが分かる。逆にGriewank関数(f_2)やRastrigin関数(f_3)のように多峰性の関数では, 集団サイズが大きいが初期サンプリングを大域的に行えるため, 有利であると考えられる。以上のことは既存の研究結果で得られていた知識だが, 今回再確認し, 高精度で最適なパラメータ値を求めることができた。

4 島数を変えた場合における各手法の性能の比較実験

2章で述べた設定及び前節で得られた設定を用いて6つの関数の最適化問題に取り組み, best Homogeneous, broad Heterogeneous, 今回の提案手法である narrow Heterogeneous の性能比較を行った. migrationはある場合とない場合の両方を実行し, 各100回の実験を行ってその平均で評価した. 島の数は $N = 16, 64, 128$ の場合を試し, 逐次環境で並列環境をシミュレートした. 各実行において $MAX_NFE = 100000$ としたが, 逐次環境で並列環境をシミュレートするため実際は $MAX_NFE = 100000 \times N$ とした. そして, 性能評価に用いるNFEは(実際の NFE)/ N として, 並列環境を擬似的に再現した. 並列GAの研究において, 総NFE(=全体の集団サイズ)を分配すべき資源として考える研究もあるが⁸⁾, 今回は単に並列化資源を均等に分配し, どのようなポートフォリオ戦略を立てるかのみ注目した. よって島ごとの集団サイズはばらばらであり, その方が実用に近い環境だと考えられる.

結果のうち, 特にSphere関数, Griewank関数の場合の結果を図1,2にまとめる.

Sphere関数はbest Homogeneousが一番よい性能を示したものの例で, Griewank関数はnarrow Heterogeneousが一番よい性能を示したものの例である. 他の関数についても, 定性的な挙動はこのどちらかに分類される.

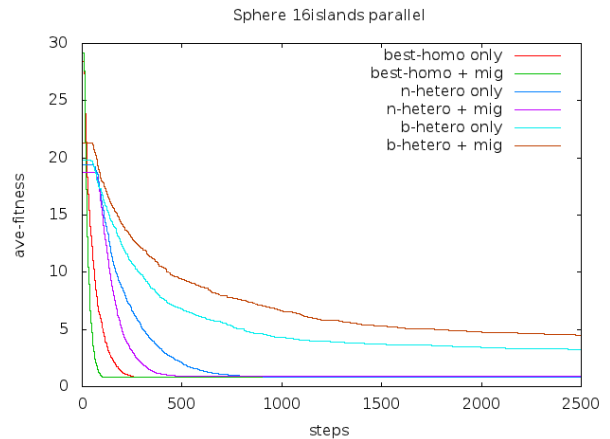
先行研究の枠組みであるbroad Heterogeneousが性能の面では一番劣っている. 無駄な島が多く生成されているという直感に反しない結果となった.

5 Migrationの効果の検証

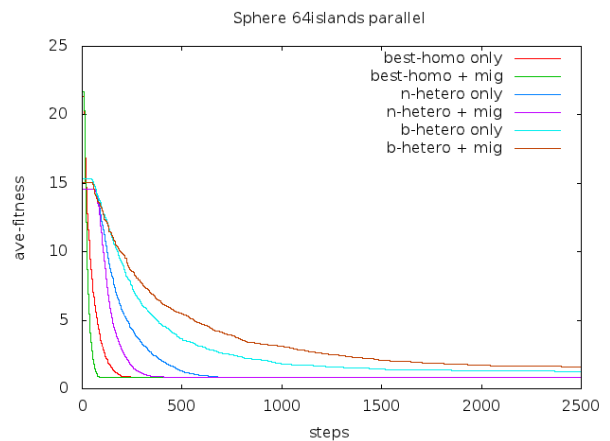
migrationの効果について, narrow Heterogeneousとbest Homogeneousではmigrationを行ったほうが収束が早まっているが, broad Heterogeneousではmigrationを行ったほうが探索性能が落ちていく様子を確認できる. パラメータ設定が不十分である場合には, migrationを行い全体の多様性を下げってしまうことで, 初期収束に陥ってしまったと考えられる.

以上の結果から, migrationの効果をはっきりさせるため, migrationあり・なしの間の有意差検定を行った. 今回, NFEについて正規性を仮定できる理由はなかったため, ノンパラメトリックな検

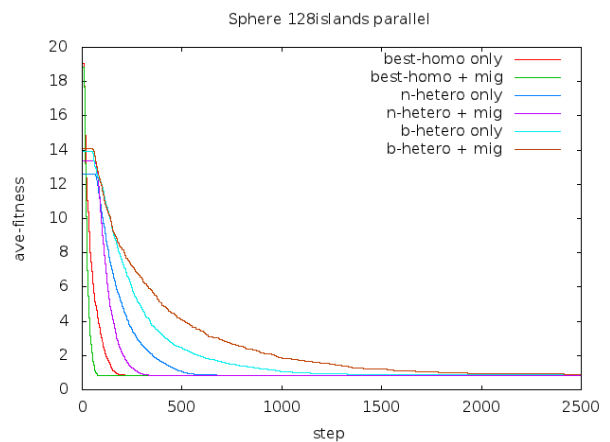
図1 評価回数の経過に対する最良評価値の推移(Sphere関数)



(a) Sphere $N = 16$

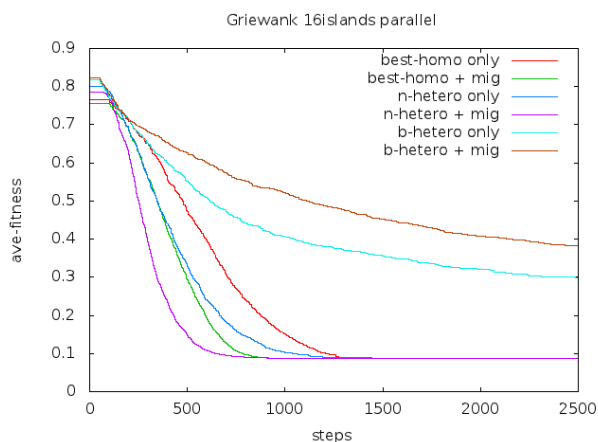


(b) Sphere $N = 64$

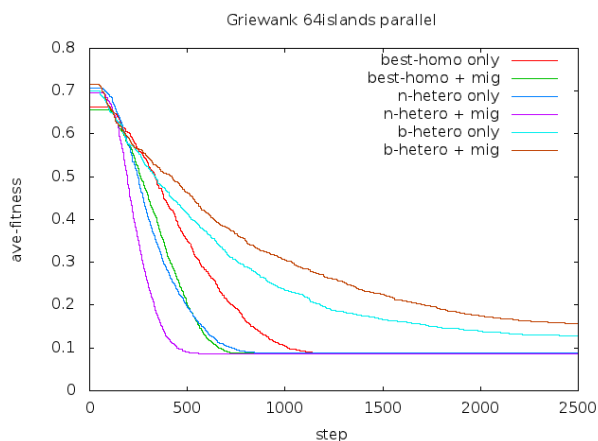


(c) Sphere $N = 128$

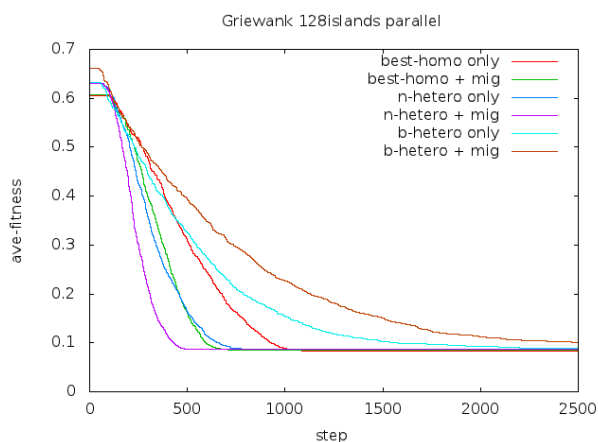
図2 評価回数の経過に対する最良評価値の推移 (Griewank関数)



(a) Griewank $N = 16$



(b) Griewank $N = 64$



(c) Griewank $N = 128$

表3 各手法におけるmigrationの有無に関するNFEの有意差検定 ($N = 128$): 表中の m は優位差があった場合のmigrationの効果を示す変数. $m = '+'$: migrationはプラスの効果, $m = '-'$: migrationはマイナスの効果, を表す.

f	b-Heterogeneous		n-Heterogeneous		b-Homogeneous	
	p-Value	m	p-Value	m	p-Value	m
f_1	$< 2.2e-16$	-	$< 2.2e-16$	+	$< 2.2e-16$	+
f_2	$2.279e-8$	-	$< 2.2e-16$	+	$< 2.2e-16$	+
f_3	$2.962e-5$	-	$< 2.2e-16$	+	$< 2.2e-16$	+
f_4	$1.746e-8$	-	$< 2.2e-16$	+	$< 2.2e-16$	+
f_5	$4.24e-4$	-	$< 2.2e-16$	+	$< 2.2e-16$	+
f_6	$1.258e-4$	-	$< 2.2e-16$	+	$< 2.2e-16$	+

定であるWilcoxonの順位和検定を行った. 特に $N = 128$ の場合における結果を表3にまとめる.

表3の m は, 有意差があった場合に, migrationあり・なしのどちらが性能がよいかを表す変数である. $m = '+'$ ではmigrationがあった方がよく, $m = '-'$ ではmigrationはない方が, 性能が良くなることを示している. 有意水準0.05の検定において, narrow Heterogeneous, best Homogeneousではmigrationを行った方が有意に性能がよい. これは, よい設定や問題条件の下ではmigrationは探索効率を高めるという従来⁸⁾の知見と一致する⁵⁾. 一方, broad Heterogeneousにおいてはmigrationを行うと有意に性能が落ちる. これは, パラメータ設定が上手くない場合はmigrationの効果が負に働く, ということを示している.

同じHeterogeneousの枠組みでも, ランダムサンプリングするパラメータ空間を調整することによってmigrationの有効性は異なってくる. narrow Heterogeneousでは, broad Heterogeneousに比べてmigrationによる効果をプラスにし, 性能の向上を達成することができた.

また, 今回はマイグレーショントポロジに双方向リングを用いたが, 他に2次元グリッドや完全グラフなどのトポロジも考えられる. 様々なトポロジを用いた場合における影響の調査は今後の課題であるが, broad Heterogeneousにおいて, 最も情報伝播が遅いリング状のトポロジで負の効果があるということは, 更に情報伝播が速い場合には更なる負の影響が予測される.

表 4 narrow Heterogeneous VS best Homogeneous

f	n-Heterogeneous		b-Homogeneous		p-Value
	SR	NFE	SR	NFE	
f_1	100	270.59	100	68.34	<2.2e-16
f_2	100	398.17	100	571.34	<2.2e-16
f_3	100	604.73	100	872.86	<2.2e-16
f_4	100	1685.27	100	567.42	<2.2e-16
f_5	100	828.60	100	745.40	2.01e-4
f_6	100	463.13	100	456.82	0.8649

6 島数が島モデルGAの探索に与える影響の調査

3つの枠組みのうち broad Heterogeneous の性能が劣っていることは分かったが, narrow Heterogeneous と best Homogeneous の間にどの程度の違いがあるのか, Wilcoxon の順位和検定により調べた. 前章の結果より, migration ありの場合について比較を行えばよい. 特に $N = 128$ の場合の結果を, 表4にまとめる.

表4において, NFEに有意差があった場合には, 性能が良い方を太字で示した. Schwefel-1.2 関数 (f_6) では, 2つの間に有意差は出なかったが, その他の関数については全て有意な差が出た. ベンチマーク問題のうち, 半分に関しては best Homogeneous が優れているが, Griewank 関数 (f_2), Rastrigon 関数 (f_3) に関しては narrow Heterogeneous の方が優れている. これは, best な設定として用いたパラメータの組み合わせは島1つの場合の best であり, 島が複数に増えたことでその環境での best パラメータ値とは異なってしまったことによるものと考えられる. Miki, et. al. も, 島の数によって best なパラメータ値が変わることを指摘している⁶⁾.

表4の結果を見ると, narrow Heterogeneous は十分考慮に値する枠組みだと思われる. 本実験で比較対象とした best Homogeneous の設定には, 異なる問題, 異なる並列環境それぞれの場合で多大な労力が必要であり, 実用上では, 時間コスト・金銭コストなどの観点から, best なパラメータ設定を問題ごとに探していくのは現実的ではない. 一方, narrow Heterogeneous GA は, 探索性能は bset に劣るものの探索成功率は同等であり, 特に事前の労力を全く必要としない. 本質的にパラメータフリーなランダム・ヘテロ

ジニアスGAが表4の性能を示すことは, ランダム・ヘテロジニアスGAが有用な手法であることを示していると考えられる.

さらに, 図1, 2において $N = 16, 64, 128$ の結果を順に見比べると, N を大きくするにつれて各アルゴリズム間の差が縮まっている様子が確認できる. 図3に, narrow Heterogeneous が勝っていなかった4つの関数について, 縦軸に best Homogeneous との NFE の差, 横軸に島の数をとったものを示す.

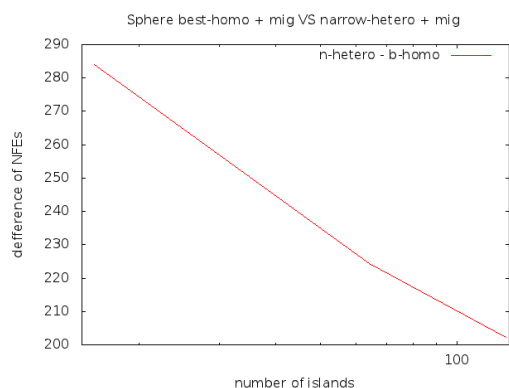
図3を見ると, $N = 16 \rightarrow 64$ のときに差が大きく縮まり, $N = 64 \rightarrow 128$ においてもさらに差が縮まっていることが分かる. このまま N を大きくしていくと, 徐々に best の性能に近づいていき, いずれは best Homogeneous の性能とそう変わらない性能が得られることが予想される. この現象は Gong and Fukunaga にも言及されていたが³⁾, 島の数が増えるにしたがって, どこかの島は best に近いパラメータ設定を得て, 探索を進めるからだと思われる. この現象は, narrow Heterogeneous GA が, 用意されたパラメータ空間から一様にパラメータをサンプリングしてくるために起こっていると考えられる. すなわち, 本実験のように連続的なパラメータ設定を用いた場合に期待できる現象であり, 先行研究⁶⁾ のように離散的なパラメータ設定を用いてしまうと, この効果は期待できないように思われる. あらかじめ離散的に設定しておいた値の中に, 扱う関数に対する最良なパラメータ設定が入っていないと, best Homogeneous に近い性能は得られないことが予測される.

特に Biazzi, et. al.²⁾ のような大規模並列化の環境 ($N = 2^{16}$) では, narrow Heterogeneous GA の更なる性能向上が期待できる.

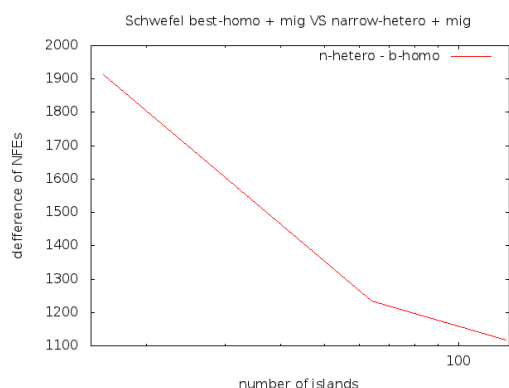
7 Heterogeneous環境のロバスト性の検証

本章では, best なパラメータ値や学習によるパラメータ調整などを仮定しない場合における narrow Heterogeneous の有効性を調べるため, 一般によいとされているパラメータ値を用いた Homogeneous GA との比較を行った. 直感的には, 推奨値を用いる Homogeneous よりも narrow Heterogeneous の方がロバストであると考えられる. Homogeneous のパラメータ設定は $p = 50, \mu = 0.01 (= 1/L), c = 0.6$ とし, この設定を standard Homogeneous と呼ぶ. Homogeneous のパラメータ値を変えた以外は全く同じ環境の下で,

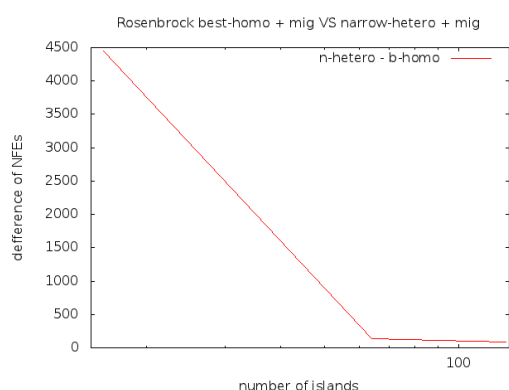
図3 narrow Heterogeneousが劣っていた関数におけるbest HomogeneousとのNFEの差



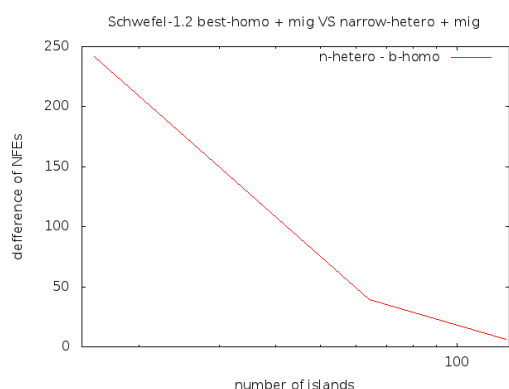
(a) Sphere関数



(b) Schwefel関数



(c) Rosenbrock関数



(d) Schwefel-1.2関数

表5 narrow Heterogeneous VS standard Homogeneous

f	n-Heterogeneous		s-Homogeneous		p-Value
	SR	NFE	SR	NFE	
f_1	100	270.59	100	232.30	9.582e-9
f_2	100	398.17	100	374.12	1.987e-3
f_3	100	604.73	100	672.80	3.632e-4
f_4	100	1685.27	0	100000.0	< 2.2e-16
f_5	100	828.60	100	1099.04	1.989e-15
f_6	100	463.13	100	455.26	0.687

同様のベンチマーク問題を解き、Wilcoxonの順位和検定を行った。N = 128の場合の結果を、表5に示す。

Sphere関数(f_1)、Griewank関数(f_2)ではstandard Homogeneousの方が有意に性能がよい。特にGriewank関数はbest Homogeneousよりもstandard Homogeneousの方が性能がよいが、これは前述したとおりNを大きくしたことによりbestなパラメータ値が変化した影響だと考えられる。

また、特にstandard HomogeneousのSchwefel関数(f_4)における挙動は特徴的である。standard Homogeneousの枠組みでは、 f_4 に関して探索成功率が極端に低くなっており、これはある一定のパラメータ設定を用いる場合の限界を示していると考えられる。

一方、narrow Heterogeneousではどの関数も今回設定した基準で探索に成功している。よって、6章の結果と合わせて十分ロバストな枠組みだと考えられる。

8 おわりに

本研究では、ランダム・ヘテロジニアスGAにおいてランダムサンプリングするパラメータ空間を無制限に取った場合、すなわち $\mu = [0.0, 1.0]$, $c = [0.0, 1.0]$ としたときに、migrationの効果が負に働くことを実験的に示した。それを踏まえてnarrow Heterogeneousの枠組みを提案し、migrationによる正の効果を得て、性能を向上させることに成功した。実際、narrow Heterogeneousのパラメータサンプリングの範囲の決定は実験的に求めたのではなく、標準的に用いられているパラメータ付近に幅を持たせたものであることに注意して欲しい。突然変異率の推奨値は $\mu = 1/L$ であるため、 $\mu \in [0.0, 2/L]$ とし、交叉率の推奨値は

0.6, 0.7などであるため, $c \in [0.3, 1.0]$ とした. これには, パラメータの推奨値という情報を除いて, 事前の知識を全く仮定していない. また, このパラメータ値の推奨値という情報のみを持っていた場合における探索法として, narrow Heterogeneousとstandard Homogeneousの枠組みの比較を行った. その結果, Homogeneousでは探索が失敗してしまうケースも見られたが, narrow Heterogeneousは十分な性能を示した. これにより, narrow HeterogeneousはHomogeneousに比べて十分ロバストな枠組みだと考えられる. よって, 少なくともまずシンプルなGAを試す場合には, Homogeneousではなくnarrow Heterogeneousを試すことが有効な手段であるといえる.

さらに, 島の数を増やすにつれ narrow Heterogeneousとbest Homogeneousの差が縮まっていく様子が観察された. これはランダムサンプリングにより, 島の数が増大するほどよいパラメータ設定の島が生成される確率が高くなり, 島全体のうち一部でもよいパラメータ設定の島があれば探索は上手くいくからだと考えられる. 更に島の数を増大させれば, narrow Heterogeneousの性能はbestなものに近づいていくと考えられる. そのため今後の課題としては, $N = 1000$ 以上の更なる並列化が挙げられる. 今回の比較対象としたbest Homogeneousの設定は, 事前に数千回の実験を行って求めたものであり, 多大なコストを費やさなければならない. 一方 narrow Heterogeneousは, 事前の努力をせずにその性能に近づいていくことのできる枠組みであり, 実用的かつ十分に有効な戦略だといえる.

本研究ではGAを用いてnarrow Heterogeneousの枠組みを試したが, 島の数が増えるにつれbestな設定が得られ, bestな設定と同程度の性能が得られる, という考察は, パラメータ値を用いるアルゴリズムに共通なものではないかと考えられる. 本研究で得られたnarrow Heterogeneousにおける考察がGA以外のアルゴリズムにも適用可能かどうかは, 今後の研究課題である.

参考文献

- 1) Enrique Alba and José M. Troya. A survey of parallel distributed genetic algorithms. *Complexity*, Vol. 4, No. 4, pp. 31–52, 1999.
- 2) Marco Biazzi, Balazs Banhelyi, Alberto Montresor, and Mark Jelasity. Distributed hyperheuristics for real parameter optimization. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, GECCO '09, pp. 1339–1346, New York, NY, USA, 2009. ACM.
- 3) Yiyuan Gong and Alex Fukunaga. Distributed island-model genetic algorithms using heterogeneous parameter settings. In *Evolutionary Computation (CEC), 2011 IEEE Congress*, pp. 820–827, 2011.
- 4) Bernardo A. Huberman, Rajan M. Lukose, and Tad Hogg. An economics approach to hard computational problems. *Science*, Vol. 275, No. 5296, pp. 51–54, January 1997.
- 5) Jörg Lässig and Dirk Sudholt. The benefit of migration in parallel evolutionary algorithms. In *GECCO*, pp. 1105–1112, 2010.
- 6) M. Miki, T. Hiroyasu, M. Kaneko, and K. Hatanaka. A parallel genetic algorithm with distributed environment scheme. In 1999 IEEE International Conference, editor, *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings.*, Vol. 1, pp. 695–700, 1999.
- 7) Fei Peng, Ke Tang, Guoliang Chen, and Xin Yao. Population-Based Algorithm Portfolios for Numerical Optimization. *IEEE Transactions on Evolutionary Computation*, Vol. 14, No. 5, pp. 782–800, October 2010.
- 8) Z. Skolicki and K. De Jong. The importance of a two-level perspective for island model design, 2007.
- 9) Shige-yoshi Tsutsui, Masayuki Yamamura, and Takahide Higuchi. Multi-parent recombination with simplex crossover in real coded genetic algorithms. In Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, Vol. 1, pp. 657–664, Orlando, Florida, USA, 13–17 July 1999. Morgan Kaufmann.
- 10) David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, Vol. 1, No. 1, pp. 67–82, 1997.
- 11) E. L. Yu and P. N. Suganthan. Ensemble of niching algorithms. *Inf. Sci.*, Vol. 180, No. 15, pp. 2815–2833, August 2010.
- 12) 池田心, 小林重信. GAの探索におけるUV現象とUV構造仮説. *人工知能学会論文誌 = Transactions of the Japanese Society for Artificial Intelligence : AI*, Vol. 17, pp. 239–246, nov 2002.