

# 並列分散遺伝的アルゴリズムにおける最適な交叉スキーム\*

三木 光範, 廣安 知之, 吉田 純一, 大向 一輝

同志社大学 工学部

並列分散遺伝的アルゴリズム (Parallel Distributed Genetic Algorithms : PDGA) において, 高品質な解が得られる要因として, サブ母集団ごとに生成された良好なスキームが移住によって他のサブ母集団のスキームと結合し, 成長していくことが考えられる. 本論文ではこのメカニズムを考慮し, 並列分散 GA の性能を高めるハイブリッド生成交叉と最良組合せ交叉という2つの新しい交叉スキームを提案する. 4つの代表的なテスト関数を用いて実験を行った結果, 提案したスキームは良好な性能を示した.

## Optimal Crossover Scheme for Parallel Distributed Genetic Algorithms

Mitsunori MIKI, Tomoyuki HIROYASU, Jun-ichi YOSHIDA, and Ikki OHMUKAI

Knowledge Engineering Dept., Doshisha University

This paper proposes new schemes for parallel distributed genetic algorithms (PDGA). Distributed GAs with multiple subpopulations provide better solutions than conventional GAs with a single population, and the proposed schemes, "Hybridization crossover" and "Best Combinatorial crossover" are developed to enhance the mechanism of PDGAs. The proposed scheme provides high local search ability in each subpopulation and high global search ability in the migration process, and is evaluated with four standard test functions. The experimental results showed that the proposed schemes is very effective.

### 1 はじめに

遺伝的アルゴリズム (Genetic Algorithms : GA) は生物の進化を模倣した確率的な最適化アルゴリズムである<sup>1)</sup>. この手法は, 従来の最適化手法では解くことが困難であった複雑な連続および離散の問題に適用できるうえ, 実装も比較的容易であるという長所がある. しかしながら, GA は膨大な反復計算を必要とするため, 計算コストが高い.

このため, GA の並列化については多くの研究がなされてきた<sup>2)</sup>. その中の一つに母集団を複数のサブ母集団に分割して GA を実行する島モデル (Island Model) がある. 本論文では島モデルの GA における各サブ母集団をプロセッサに割り当てたものを並列分散 GA (Parallel Distributed GA : PDGA) と呼ぶ. 島モデルでは各サブ母集団に対して独立に遺伝的操作が行われ, サブ母集団間で定期的に移住と呼ばれる個体の交換を行う. 島モ

デルの GA では, 並列化によって計算時間が短縮されるだけでなく, アルゴリズムとしても, 従来の GA に比べて, より適合度が高い個体の発見が可能であると報告されている<sup>3, 4)</sup>.

しかし, 母集団を分割して移住を行う並列分散 GA の解探索メカニズムは, 単一母集団の GA とは異なる<sup>5)</sup>. このため, 並列分散 GA の解探索能力をさらに高めるためには, 遺伝的オペレータのスキームを並列分散 GA に特化したものにする必要があると考えられる.

遺伝的オペレータの最適な調整に関しては, これまでに多くの研究が行われてきた<sup>6)</sup>. しかしながら, それらの大部分は単一母集団 GA に関するものであり, 並列分散 GA においてなされた研究はほとんどない. このため, 本論文では, GA の性能を大きく左右する交叉オペレータに注目し, 並列分散 GA に適した交叉スキームを提案する.

\* 出典:  
情報処理学会・数理モデル化と問題解決シンポジウム  
論文集 (情報処理学会シンポジウムシリーズ Vol.2000,  
No.5), pp49-56, 2000

## 2 並列分散 GA

### 2.1 GA の基本概念

GA では、個体の集まりを母集団 (population) とよび、ある世代を形成している個体群のうち環境への適合度 (fitness) の高い個体ほど高い確率で生き残るように選択 (selection) される。さらに、個体間の交叉 (crossover) や突然変異 (mutation) によって、次の世代が形成される。このような世代の更新が繰り返されることによって、よりよい個体 (最適解に近い個体) が増え、やがて最適解に近づく。

### 2.2 並列分散 GA の基本概念

並列分散 GA では、母集団を複数のサブ母集団に分割し、各サブ母集団ごとに独立に遺伝的操作を行い、一定期間ごとに異なるサブ母集団間で移住と呼ばれる個体の交換を行う。結果として、すべての個体がひとつの母集団を形成するよりも多様性が大きくなり、より効率的な探索を進めることが可能である<sup>5)</sup>。並列分散 GA と移住の概念を図 1 に示す。

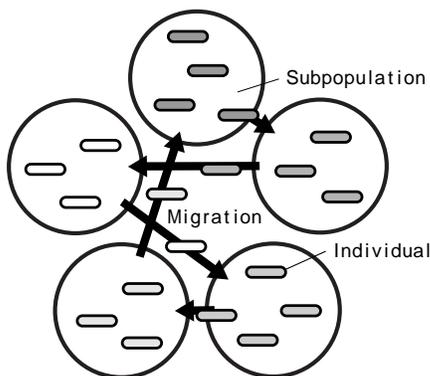


図 1: 並列分散 GA

並列分散 GA では、単一母集団の GA とは異なり、移住に関するパラメータが必要となる。移住を行う世代間隔を移住間隔 (migration interval) と呼び、サブ母集団の個体数に対する移住個体の割合を移住率 (migration rate) と呼ぶ。

### 2.3 並列分散 GA における解の成長

単一母集団の GA では、比較的良好なスキーマをもつ個体が多数を占め、他の良好なスキーマの成長を妨げ、早熟収束を起こすことがある。したがって、良好なスキーマを保存するだけでなく、時には破壊して大域的な探索を行う必要がある。

並列分散 GA においてもサブ母集団に注目すると、各々の個体数は小さくなるために早熟を起こしやすい。しかしながら、移住により複数のサブ母集団間で異なる進化をした個体を交換することで、母集団全体としては多様性を保つことができる。並列分散 GA ではサブ母集団ごとに生成された良好なスキーマが移住によって他のサブ母集団のスキーマと結合し、成長していくことで、単一母集団の GA よりも高速に、かつ高品質な解を求めることができる<sup>5)</sup>。

このように、単一母集団の GA と並列分散 GA では、解の探索メカニズムが異なるため、遺伝的オペレータは、それぞれの探索メカニズムに適した設定や調整を行うのが望ましいと考えられる。

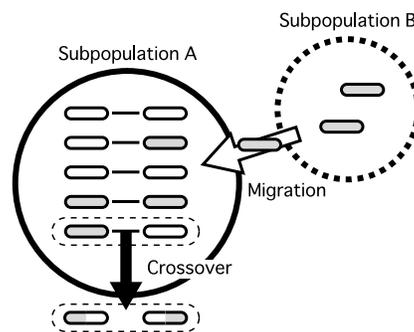


図 2: 並列分散 GA における解の成長

## 3 並列分散 GA における交叉の影響

交叉は、染色体の組み換えにより新しい個体を生み出す遺伝的オペレータである。交叉により、個体間で情報交換が行われ、初期個体内に含まれるビットが適切に組合せられる。したがって、GA の性能は実装する交叉法および交叉率の最適な選択に依存していると言える。

前節で述べた解の成長のメカニズムを考慮すると、並列分散 GA における交叉の役割は移住の前後で変化していると考えられる。移住前は各サブ母集団内で良好なスキーマを育て、移住後は他のサブ母集団で成長したスキーマを組み合わせる。本節では、この 2 つの役割を果たす交叉法と交叉率について考える。

### 3.1 対象問題

本論文で対象にする関数は表 1 に示す 4 つの 10 次元のテスト関数である。これらの関数のうち、Rastrigin 関数 (F1) および Schwefel 関数 (F2) は設計変数間に依存関係はない。Griewank 関数 (F3)

表 1: 対象問題

$$\begin{aligned}
 F1 &= 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) \\
 &\quad x_i \in [-5.12, 5.12], n = 10 \\
 F2 &= \sum_{i=1}^n -x_i^2 \sin(\sqrt{|x_i|}) \\
 &\quad x_i \in [-512, 512], n = 10 \\
 F3 &= 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \left( \cos\left(\frac{x_i}{\sqrt{i}}\right) \right) \\
 &\quad x_i \in [-512, 512], n = 10 \\
 F4 &= \sum_{i=1}^5 \left( \sum_{j=1}^i x_j \right)^2 \\
 &\quad x_i \in [-64, 64], n = 10
 \end{aligned}$$

は設計変数間に中程度の依存関係を有する．Ridge 関数 (F4) は，設計変数間に強い依存関係を有する．いずれも大域的最適解は 0 である．

本論文における数値実験では，染色体のビット長  $L$  を 100bit (1 設計変数 10bit) とした．また，選択オペレータはルーレット選択であり，エリート保存戦略を用いた．

### 3.2 並列分散 GA における交叉法の影響

以下の議論は，本研究で用いた連続関数の最大化問題において，連続変数を 2 進符号化した場合のものである．

#### 3.2.1 交叉法

交叉の働きは実装する交叉法の違いによって異なる．並列分散 GA における最適な交叉の条件とは，親個体のスキーマを保存し，成長させる交叉であるといえる．そこで本論文では，代表的な交叉法のうちスキーマの保存に優れた交叉として 1 点交叉 (1X) と 2 点交叉 (2X)，スキーマを保存しない交叉として一様交叉 (Uniform Crossover: UX) を用いて，交叉法が GA の性能におよぼす影響を検証した．それぞれの交叉の特徴は，文献<sup>7)</sup>に詳しい．

#### 3.2.2 交叉法の影響

並列分散 GA の性能に交叉法がどのように影響するかを考える．ここでは 1X, 2X および UX の 3 種類の交叉法を表 1 の関数に適用した．母集団

サイズは 400 または 800，交叉率は 0.6，突然変異率は  $1/L$  とした．並列分散 GA の場合には，サブ母集団数を 8，移住間隔は 5 世代，移住率は 0.3 とした．

母集団サイズ 400 における 2000 世代での適合度を表 2 に示した．2000 世代までに最適解を発見できたものについては最適解発見世代 ( # ) を示している．なお，結果は 20 試行の平均値である．

表 2 に示すように，すべての関数で同様の傾向が現れている．まず，SGA と PDGA を比較すると，PDGA の方がすべての関数で良い成績を示した．PDGA のみに注目すると 1X および 2X が良い成績を示している．3.2.1 で述べたように，1X および 2X がスキーマを保存するのにに対し，UX はスキーマを破壊する．このため 1X・2X と比べて UX では前の世代の情報が有効に利用できず，探索の効率が悪い．並列分散 GA においては，他のサブ母集団で成長し，移住してきた部分解を破壊してしまうため移住の効果が得られないと考えられる．

### 3.3 並列分散 GA における交叉率の影響

#### 3.3.1 交叉率

SGA において一般的に用いられる交叉率は 0.45 から 0.95 である<sup>8)</sup>．Tuson & Ross は交叉率を 0.05 から 0.95 まで 0.05 刻みで変化させる大規模な研究を行った<sup>9)</sup>．その結果，最適な交叉率は解くべき問題によって変化した．また，交叉率が過度に高い場合，早熟に陥る可能性がある．一方，PDGA における最適な交叉率に関する研究は行われていない．

#### 3.3.2 交叉率の影響

SGA および PDGA における最適な交叉率を調べるため，数値実験を行った．設定する交叉率は 0.3, 0.6 および 1.0 とした．また，個体数との関連を調べるため，母集団サイズは 160, 400 および 800 とした．交叉法は 1 点交叉である．並列分散 GA においては，サブ母集団数を 8，移住間隔を 5 世代，移住率を 0.5 とした．対象問題は表 1 のテスト関数である．

母集団サイズ 400 における 1000 世代での適合度を表 3 に示した．1000 世代までに最適解を発見できたものについては最適解発見世代 ( # ) を示している．結果は 20 試行の平均値である．

SGA における最適な交叉率は，関数によって大きく異なる．母集団サイズを変化させた場合にも

表 2: 交叉法の影響：最大世代での適合度（#最適解発見世代）

	PDGA			SGA		
	1X	2X	UX	1X	2X	UX
Rastrigin	# 386	# 420	# 804	-0.206904	-0.150967	-2.6451
Schwefel	# 370	# 373	# 708	# 1228	# 1235	-0.00591232
Griewank	-0.100508	-0.137165	-0.179465	-0.286196	-0.327726	-0.329169
Ridge	-0.000935	-0.000575	-0.00435	-0.102055	-0.098925	-0.229515

表 3: 交叉率の影響：最大世代での適合度（#最適解発見世代）

	PDGA			SGA		
	$P_c = 0.3$	0.6	1.0	$P_c = 0.3$	0.6	1.0
Rastrigin	# 220	# 197	# 166	-0.302925	-0.79939	-0.956702
Schwefel	# 214	# 174	# 168	-0.0059123	-0.0118246	-0.648429
Griewank	-0.126492	-0.103997	-0.0576392	-0.372117	-0.306758	-0.843136
Ridge	-0.0085938	-0.0046875	-0.0039063	-0.202539	-0.420031	-0.911453

その傾向は変わるため、最適な交叉率の設定は対象問題および個体数ごとに行わなければならないことがわかる。一方、PDGAでは、対象問題、個体数にかかわらず全ての場合について交叉率が高いほうが良い結果となり、最適な設定は 1.0 である。

図 3 は Rastrigin 関数を、母集団サイズを 400 とした場合の SGA と PDGA の適合度の推移である。SGA では、探索の前半段階では交叉率の高いものが性能が良いが、後半では適合度の上昇が遅くなり、順位は逆転する。よって最適な交叉率は世代によって異なるといえる。一方、PDGA では全ての世代において交叉率が高いものほど適合度が高く、最終的には良好な解を速く得られる。この結果から、PDGA における解探索は移住とその後の交叉によって行われており、交叉率を 1.0 とするのがその機能を生かせる設定であることがわかる。

#### 4 並列分散 GA に適した交叉スキーム

実験の結果から、並列分散 GA の解探索のメカニズムが単一母集団の GA と異なることがわかった。本節では並列分散 GA の解探索能力をさらに高めるための新しい交叉スキームを提案する。

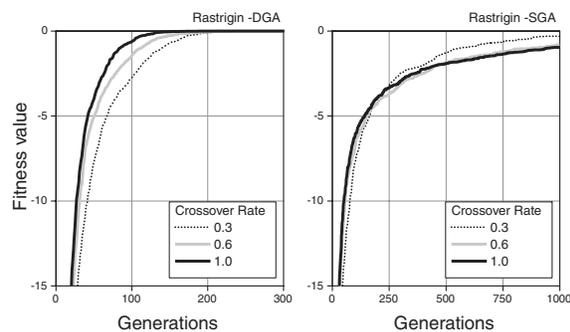


図 3: 適合度の推移 (PDGA / SGA)

#### 4.1 最良組合せ交叉 (BCX)

##### 4.1.1 最良組合せ交叉の概念

並列分散 GA の性能を高めるためには、各サブ母集団において良好なスキームを成長させることが重要である。3.2.2 において 1X・2X が UX よりもよい性能を示したのは、UX が親個体のスキームを破壊するのに対し、1X・2X は親個体のスキームを保存し、成長させるためである。そこで、並列分散 GA において確実に良好なスキームを生成するための交叉法として、最良組合せ交叉 (Best Combinatorial Crossover: BCX) を提案する。

BCX では、親個体から 1X または設計変数間 1X によって生成される全ての子個体を評価し、親

個体とそれらの子個体の中から，もっとも適合度の高い2個体を子として採用する．従って，確率的な要因に左右されることなく適合度の高い個体を生成することが可能であり，両親の持つスキーマのうち最も有効なものを選択することができる．しかしながら，評価計算回数が膨大になるという欠点がある．本論文では，BCXのうち1Xを用いるBCXをビット間BCX(B-BCX)，設計変数間1Xを用いるBCXを変数間BCX(V-BCX)と呼ぶことにする．

BCXのメカニズムを分かりやすくするために，親個体が11111と00000の場合のB-BCXの例を図4に示す．B-BCXでは交叉点を1ビットずつシフトさせて，1点交叉を行い，そこで生成される子のなかで適合度の高い個体を子とする．

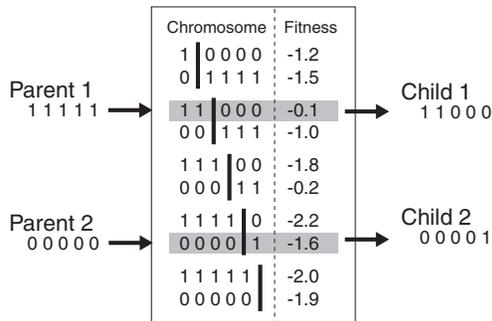


図 4: 最良組合せ交叉 (BCX)

#### 4.1.2 BCXの計算負荷の軽減

BCXにおいて両親の染色体に共通部分が多い場合には，全てのビット間で交叉を行っても，生成される子には重複が多く，無駄な評価計算を何度も行うことになる．そこで，B-BCXでは交叉を行う前に両親の染色体を検査し，共通部分とそうでない部分をフラグビット列に記録する．フラグビット列をもとに交叉点を決定することで，子個体の重複を避けることができる(図5)．

#### 4.1.3 BCXの適用

BCXを表1に示す4つのテスト関数に適用し，提案手法の解探索能力を検討する．B-BCXとV-BCXおよび1Xを用いて数値実験を行った．母集団サイズは400，交叉率は1.0，突然変異率は $1/L$ ，サブ母集団数は8，移住間隔は5世代，移住率は0.5とした．2000世代における適合度と評価計算回数を表4に示す．2000世代までに最適解を発見できた物については最適解発見世代( # )を示し

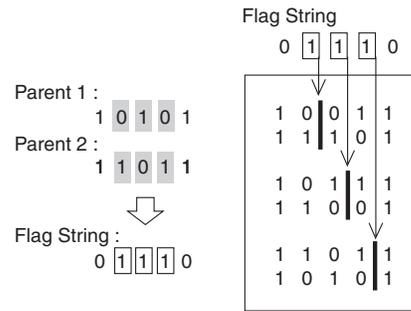


図 5: フラグビット列の生成と交叉点の決定

ている．また，関数における適合度の推移の様子を図6に示す．

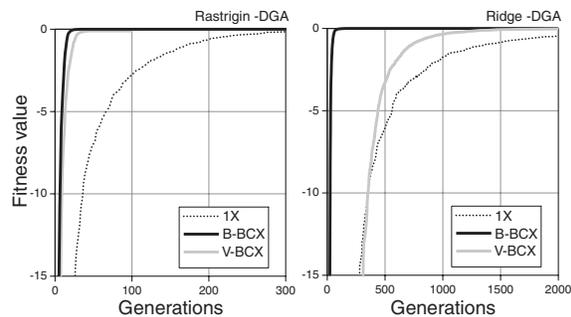


図 6: 適合度の推移 (Rastrigin / Ridge)

図6が示すように，BCXでは探索の序盤において急速に適合度が上昇し，少ない世代数で最適解に達している．ただし，BCXでは1回の交叉につき何回もの評価計算を行っているため，世代数だけで性能を評価することはできない．評価計算回数に注目すると，Rastrigin関数ではBCXの性能は1Xに劣っている．しかしながら，そのほかの関数では，1Xよりも高い性能を示している．特に，設計変数間に依存関係をもつ Griewank 関数や Ridge 関数は1Xでは最適解を得るのは困難であるが，B-BCXでは100%の確率で最適解を発見した．V-BCXは設計変数間に依存のない問題についてはB-BCXよりも少ない評価計算回数で最適解を発見したが，依存関係のある問題では最適解を発見することはできなかった．しかし，1Xよりは良好な解を発見することができた．このように，BCX，特にB-BCXは1Xと比較して，高い解探索能力を持つことが示された．

表 4: 最大世代での適合度 (カッコ内は評価計算回数)

	1X	B-BCX	V-BCX
Rastrigin	# 366 (147,040)	# 37 (247,286)	# 65 (178,618)
Schweffel	-0.126761 (800,000)	# 28 (236,272)	# 39 (178,640)
Griewank	-0.127025 (800,000)	# 137 (794,261)	-0.165822 (800,000)
Ridge	-0.467969 (800,000)	# 138 (764,499)	-0.03125 (793,160)

## 4.2 ハイブリッド生成交叉

### 4.2.1 ハイブリッド生成交叉の概念

並列分散 GA では、各サブ母集団において部分解を形成するとともに、その部分解を移住後の交叉によって組み合わせることが重要である。交叉率を最大の 1.0 とすることは、この機能を高める設定でもあり、結果として良好な解を得ることが可能であることが確認された。

ここでは、移住およびその後の交叉による解探索機能をさらに高めるために、ハイブリッド生成交叉 (Hybridization Crossover: HX) を提案する。

まず、分散 GA における移住および交叉の機能を数値化するために、ハイブリッド生成率というパラメータを定義する。ハイブリッド生成率とは、ある世代においてサブ母集団 A にサブ母集団 B からの個体が移住してきた場合に、交叉によってサブ母集団 A に属していた個体 (以下 Native) とサブ母集団 B からの個体 (以下 Migrant) との混血が生まれる割合とする。

従来の交叉スキームでは、ハイブリッド生成率 (Hybridization Rate)  $H$  は、移住率を  $\mu$ 、交叉率を  $P_c$  とすると以下のように求められる。

$$H = 2P_c\mu(1 - \mu)$$

移住率の上限は 0.5 であり、この式によって得られる  $H$  の上限も 0.5 である。交叉率を 1.0、移住率を 0.5 という最大限の設定を行ったとしても、解探索の主役であるハイブリッド個体は 50%しか生み出すことができない。その他の個体は、Native 同士の子供、もしくは Migrant 同士の子供であり、これらの個体は移住の影響を受けていない。

HX は、図 7 のように移住および交叉の方法を変更することにより、Migrant を確実に Native と交叉させ、ハイブリッド個体をより多く生み出すことができるようにしたものである。

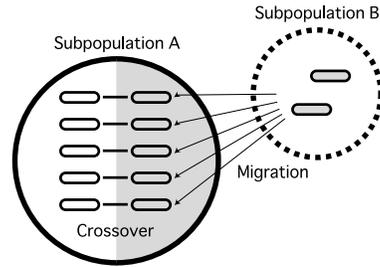


図 7: ハイブリッド生成交叉のメカニズム

- 交叉スキーム

交叉において、親となる個体は半分に分割したサブ母集団のそれぞれから 1 個体ずつ選ぶこととする。

- 移住スキーム

Migrant は、半分に分割したサブ母集団のどちらか一方に集中して配置されるようにする。

このスキームでの  $H$  は以下の式で表すことができる。

$$H = 2P_c\mu$$

$H$  の上限は 1.0 となり、交叉率を 1.0 とすることで  $H$  を 0~100%まで自由に制御することが可能になった。また、これにより移住および交叉による解の探索が実際にどれほど有効に機能しているかを調べることができる。

### 4.2.2 HX の適用

ハイブリッド生成率が並列分散 GA の性能に及ぼす影響を調べるため、表 5 のパラメータ設定で実験を行った。この実験では全て HX を用いている。交叉法は 1X、対象問題は 4 種類のテスト関数である。

母集団サイズを 160,400 および 800、サブ母集団数を 8 とした場合の結果を表 6 に示す。概ね  $H$

表 5: パラメータ設定

$H$ (移住率)	0.2	0.5	1.0
母集団サイズ	160	400	800
サブ母集団数	4	8	16

の高いものほど良い結果を得られることがわかる。しかしながら, Rastrigin 関数や Schwefel 関数において母集団サイズが大きい場合には,  $H$  を変えても収束世代数にはそれほど大きな差はみられない。また, Griewank 関数ではわずかながら  $H$  の低いほうが良い結果となっている。

表 6: HX の性能

Pop.	$H = 0.2$	0.5	1.0
<b>Rastrigin</b>			
160	# 337	# 296	# 281
400	# 197	# 188	# 175
800	# 154	# 147	# 138
<b>Schwefel</b>			
160	# 263	# 336	# 228
400	# 186	# 176	# 171
800	# 161	# 157	# 146
<b>Griewank</b>			
160	-0.0768976	-0.0734775	-0.0817898
400	-0.0552862	-0.0451388	-0.0479925
800	-0.0192981	-0.0129534	-0.0277259
<b>Ridge</b>			
160	-0.090625	-0.0789062	-0.0453125
400	-0.0101563	-0.0132812	-0.00625
800	# 870	# 808	# 681

図 8 は Ridge 関数での適合度の推移である。 $H$  が GA の探索に及ぼす影響は探索の前半と後半で大きく異なることが明らかになった。まず, 探索の前半では,  $H$  が高いほうが適合度の上昇は速い。図 9 に示すように, 最良個体に対するハミング距離の平均は  $H$  が高いものほど急速に小さくなっている。これにより, 移住および交叉による解探索は,  $H$  を高めることによって有効に機能していることがわかる。また, サブ母集団数が多い場合や,

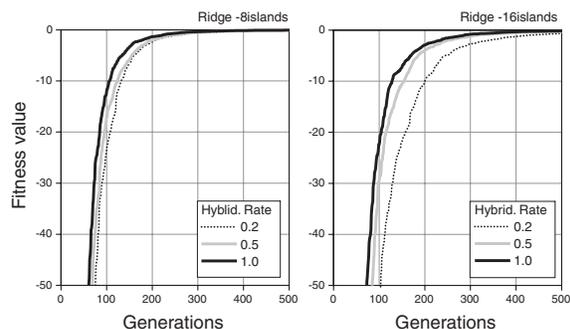


図 8: 適合度の推移 ( 8islands / 16islands )

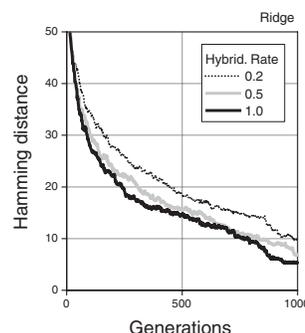


図 9: 最良個体とのハミング距離

母集団サイズが少ない場合には  $H$  が高いものと低いものとの間に大きな差が現れる。このことから, HX はサブ母集団内の個体数が少ない場合に特に有効であるといえる。

しかし, 最適解付近での適合度の上昇は,  $H$  が高いほど遅くなっており, 結果として最適解の発見世代はどの  $H$  でもそれほど変わらない。また, Griewank 関数では後半の世代で  $H$  の低いものが逆転する結果も見られる。

探索の終盤では個体間の多様性は低く, よって移住および交叉による解の成長はあまり期待できない。そのような状況では突然変異によって 1 ビットずつ探索していくが,  $H$  が高すぎると, 移住世代のたびに島内の傾向が大きく変わるために局所探索性能が落ちると考えられる。よって, 分散 GA では探索の終盤には移住をしない, もしくは移住間隔や移住率の調整が必要であることがわかる。

以上の結果より,  $H$  は分散 GA の探索性能に直結するパラメータであるといえる。しかしながら,  $H$  が与える影響は探索の前半と後半で大きく異なるため, ヒューリスティックルールなどにより適切に設定する必要がある。

### 4.3 BCX と HX の融合

本論文で提案する BCX と HX は、ともに並列分散 GA の性能を高めるための手法であるが、その役割はそれぞれ異なる。すなわち、BCX は両親のスキーマを効率よく成長させる交叉法であり、HX は移住後の交叉によってそのスキーマを組み合わせるための交叉スキームである。したがって、これらを同時に適用することにより並列分散 GA の性能はさらに高まると考えられる。

そこで、本節では、HX に BCX を組み込んだ BCX+HX と、通常の BCX を表 1 の関数に適用し、性能を比較する。GA のパラメータ設定は次の通りである。母集団サイズは 160, 400, 800, サブ母集団数は 8, 交叉率 1.0, 突然変異率  $1/L$ , 移住率 0.5, 移住間隔は 5 世代とした。表 7 に最適解発見世代数または 1000 世代における適合度を示す。概ね BCX+HX がよい性能を示しているといえる。特に Rastrigin 関数や Ridge 関数においてその傾向は顕著である。Griewank 関数については通常の BCX の方がよい性能を示したのものもある。これについても 4.2.2 節で述べたとおりである。また、母集団サイズが小さいほど、HX の効果は大きい。

表 7: BCX + HX の性能

Pop.	B-BCX	B-BCX + HX
<b>Rastrigin</b>		
160	# 88	# 69
400	# 34	# 35
800	# 29	# 28
<b>Schwefel</b>		
160	# 47	# 44
400	# 33	# 31
800	# 29	# 28
<b>Griewank</b>		
160	-0.0237857	-0.0261324
400	-0.0037525	-0.0024569
800	# 65	# 61
<b>Ridge</b>		
160	# 211	# 186
400	# 118	# 109
800	# 77	# 72

### 5 結論

並列分散 GA における解の成長メカニズムが単一母集団の GA とは異なることに注目し、並列分散 GA の性能を高める交叉法および交叉率を検討した。その結果、並列分散 GA においてはスキーマを保存する交叉法を実装し、交叉率は 1.0 が適当であることがわかった。また、これを元に並列分散 GA の性能をさらに高める 2 種類の交叉スキーマを提案した。(1) 確実に良好なスキーマを生成するための最良組合せ交叉 (BCX) と、(2) 移住およびその後の交叉による解探索機能をさらに高めるためのハイブリッド生成交叉 (HX) である。提案手法を 4 つの代表的なテスト関数に適用した結果、良好な結果を示した。また、BCX に HX を適用した結果、BCX よりもよい性能を示した。

HX については、解探索の終盤においては有効に機能しないことが明らかになった。探索の進行状況を考慮して、 $H$  の値を変更させる必要があるがこれについては今後の課題である。

### 参考文献

- 1) D.E.Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning*. Addison-Wesley, 1989.
- 2) Erick Cantú-Paz. A survey of parallel genetic algorithms. *Calculateurs Paralleles*, Vol. 10, No. 2, 1998.
- 3) Reiko Tanese. Distributed genetic algorithms. *Proc. 3rd International Conference on Genetic Algorithms*, pp. P.434-439, 1989.
- 4) Theodore C. Belding. The distributed genetic algorithm revisited. *Proc. 6th International Conference on Genetic Algorithms*, pp. P.114-121, 1995.
- 5) 三木, 廣安, 金子. 分散母集団遺伝的アルゴリズムにおける解探索能力. 人工知能学会全国大会, 1999.
- 6) Zbigniew Michalewics Ágoston Endre Eiben, Robert Hinterding. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 2, JULY 1999.
- 7) Kenneth A. De Jong and William M. Spears. A formal analysis of the role of multi-point crossover in genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 1992.
- 8) Lashon Booker. *Handbook of Evolutionary Computation*, chapter 3.3:1. Oxford University Press, 1997.
- 9) Peter Ross Andrew Tuson. Cost based operator rate adaptation: An investigation. *Proc. 4th Conference of Parallel Problem Solving form Nature*, 1996.