

# 分散遺伝的アルゴリズムにおける移住低減に関する検討 A study on a distributed genetic algorithm with reduced migration

学籍番号 08606 氏名 濱野 賢治

指導教員 主査 吉野 純一  
副査 内田 健

## 1. はじめに

分散遺伝的アルゴリズム(Distributed Genetic Algorithm :DGA)[1]は、遺伝的アルゴリズム(Genetic Algorithm :GA) [2]における計算コスト高の問題を、母集団を複数のサブ母集団(島)に分割し、個々の島にGAを適用することで解決するGAの分散処理モデルである。また、母集団分割により失われる個体の多様性を保持するため、DGAは島間で個体を交換(移住)する操作を必要とする。DGAを並列計算機に実装する場合、島は計算ノード、移住は計算ノード間通信として実現されるが、DGAの優れた解探索を実現するため高頻度の移住を必要とする最適化問題の場合、移住の頻発による並列化効果低減が危惧される。

本研究では、DGAの並列化効果を低下させる一因と考えられる移住を少なくすることを目的に、解探索途中で移住間隔を拡張する手法を提案する。また、提案手法がDGAの解探索におよぼす影響について数値実験を行いその効果について検討する。

## 2. 提案する移住低減手法

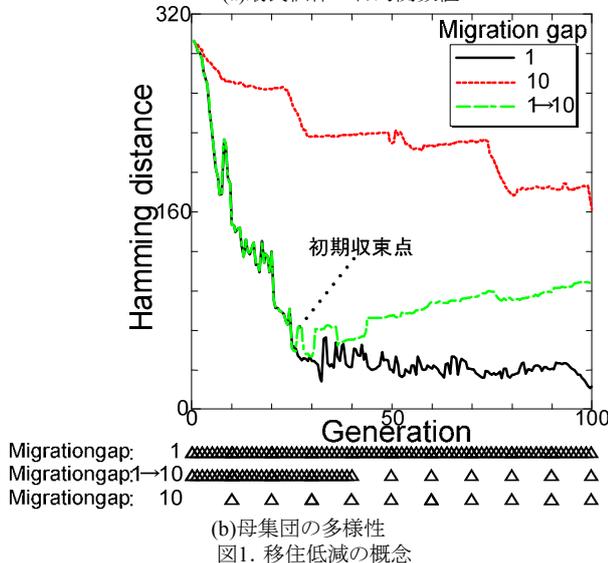
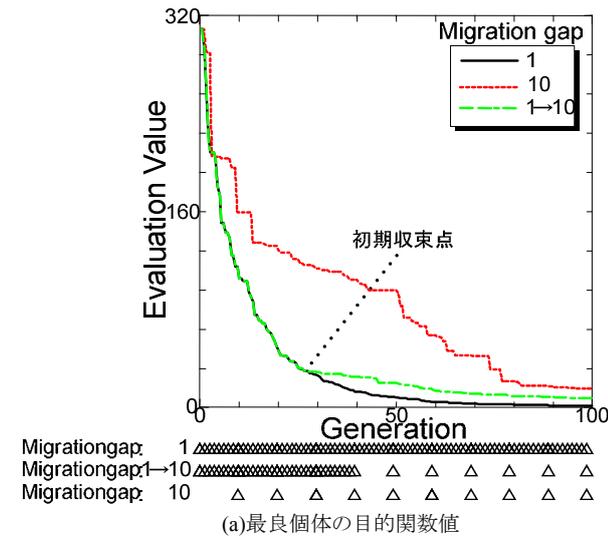


図1. 移住低減の概念

DGAの移住低減手法として、解探索の途中で移住間隔を拡げる。移住間隔とは、移住を行う世代の間隔をいい、移住間隔を拡げることで、移住を行う回数そのものを低減させる。図1に解探索の様子と移住低減の概念を示す。なお、同図中のマーク“△”は移住を表わす。図1(a)は、最良(適応度が最も高い)個体の目的関数値の世代経過であり、図1(b)は式(1)で得られる母集団の多様性の世代経過を示す。ここで、式(1)中にある $g_{max}(j)$ とは世代 $j$ における最良個体で、テスト関数値 $f(\cdot)$ によって $g_{max}(j) = \max_{g \in P(j)} f(g)$ と定義される。また、 $P(j)$ は世代 $j$ での母集団を示し、 $d_H(g_1, g_2)$ は個体 $g_1$ と個体 $g_2$ のハミング距離を意味する。

$$v(j) = \frac{\sum_{g \in P(j) - \{g_{max}(j)\}} d_H(g_{max}(j), g)}{|P(j) - \{g_{max}(j)\}|} \quad (1)$$

移住間隔を拡張する世代は、母集団の多様性の世代経過を指標に決定する。個体の多様性が解探索初期で急速に失われる初期収束点で、移住間隔を初期値から拡張することで、図1(a)のような、移住間隔を初期値のまま進化させたものと同程度の解探索を行うのではないかと期待している。さらに、移住間隔を途中で拡張せず、解探索初期から拡張後の移住間隔で進化させた場合と比較し、解探索が速くなると考えている。

## 3. 数値実験

数値実験では、島数 $4 \cdot 8 \cdot 16 \cdot 32 \cdot 64 \cdot 256$ を対象とし、解探索の途中で移住間隔を拡張した場合の解探索性能への影響を調査する。また、提案手法を適用した解探索が目的関数の影響を受けるか確認するため、式(2)~式(4)に示す3つのテスト関数を目的関数とする。これらは、いずれも30次元( $n = 30$ )で最適値として0をとる。

• Griewank

$$1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \left( \cos \left( \frac{x_i}{\sqrt{i}} \right) \right) \quad (2)$$

$(-512 \leq x_i < 512)$

• Rastrigin

$$10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) \quad (3)$$

$(-5.12 \leq x_i < 5.12)$

• Ridge

$$\sum_{j=1}^n \left( \sum_{i=1}^j x_i \right)^2 \quad (4)$$

$(-64 \leq x_i < 64)$

数値実験は300,000世代までの試行を30回行う。数値実験の初期の移住間隔を1とし、拡張タイミングとして初期収束前・初期収束後の2世代点を移住間隔拡張世代とする。また、移住間隔の拡張量が解探索におよぼす影響を見るため、初期収束前後での拡張量を5, 10, 30, 50の4通り用意する。そのほか、DGAで用いるパラメータの設定値を表1に示す。

表1. DGAのパラメータ

項目名	設定値
個体数	512
島数(分割数)	4,8,16,32,64,256
移住率	0.5
突然変異率	30×20bit
交叉率	1.0
交叉方法	一点交叉

実験結果において、解探索性能を比較するため移住間隔を1で固定した場合と、初期収束後で移住間隔を拡張した場合の結果を表3～表7に示す。なお、結果は30回試行の平均値である。発見速度において、表中(a)Griewank, (c)Ridgeの両関数は、最適解を高頻度で発見していないため、最終世代における最適座標と最良個体とのHamming距離(bit数)を発見速度とする。一方、同表中(b)Rastrigin関数は高頻度で最適解を発見していることから、最適解発見世代の平均値を発見速度とする。

表3～表7の(b)Rastrigin関数より、移住間隔を拡張することで最適解への収束が遅れる傾向が見られ、島数が増えることこの傾向は顕著となる。従い本手法は、Rastriginに適用することは解探索性能維持の観点から不適当であると考え、検討項目から除外することとした。

島数が解探索におよぼす影響をみるため、表3～表7の(a)Griewank, (c)Ridge関数における実験結果から、最良値のみを選び出し表2として以下に示す。このとき、島数が解探索におよぼす影響をみるため、島数が少ないケース(4,8,16)を表2(a)とし、島数が多いケース(32,64,256)を表2(b)として示す。

表2. 島数と移住間隔が解探索におよぼす影響

表2(a) 島数4,8,16のケース					
移住 間隔	島数4, 8, 16				
	Griewank		Ridge		
	速度	頻度	速度	頻度	頻度
1	8	13	9	0	0
1→5	8	14	16	0	0
1→10	9	12	17	0	0
1→30	7	15	18	0	0
1→50	7	15	19	0	0

表2(b) 島数32,64,256のケース					
移住 間隔	島数32,64,256				
	Griewank		Ridge		
	速度	頻度	速度	頻度	頻度
1	1	29	0	30	0
1→5	2	25	6	30	0
1→10	1	27	22	0	0
1→30	1	28	25	0	0
1→50	1	28	27	0	0

島数が4,8,16と少ない場合、Griewankは移住間隔の拡張量に関係なく、移住間隔を1で固定したものとほぼ同等の最適解発見速度・頻度を有している。Ridgeについては、拡張量が5であれば、移住間隔固定のものとはほぼ同等の解探索を行っている。

島数が32,64,256と多いケースにおいても、Griewankは移住間隔の拡張量に影響をほとんど受けず解探索を行っている。一方で、Ridgeの場合は少ない島のケースと同じく移住間隔の拡張量が増えると最適解への収束が遅くなる傾向がでており、島数が多くなるとこの傾向は顕著になる。

#### 4. おわりに

本研究では、DGAにおいて並列化効果低減の一因となっている移住を減らすことを目的に、解探索途中で移住間隔を拡張することによって移住を低減することを検討した。数値実験から、以下の知見を得た。

- 設計変数間に依存関係を持たないRastrigin関数に対する本手法の適用は、最適解への収束速度を低下させるため適用できない。
- 設計変数間に弱い依存関係を持つGriewank関数に対する本手法の適用は、島数や移住間隔の拡張量に依存することなく

適用できる。

- 設計変数間に強い依存関係を持つRidge関数に対する本手法の適用は、島数や移住間隔の拡張量に依存し、限定的な範囲(少ない島数, 少ない移住間隔の拡張量)でのみ適用できる。

今回、異なる3種類の性質を持つ関数に対して本手法を適用した。実験結果から、特定の性質を持つ(設計変数間に依存関係のある)関数に対して、提案手法が解探索性能を維持しつつ移住を低減することができる可能性を示唆した。しかし、いずれの性質を持つ関数も1つしか確認していない。そこで、今後の課題として、本手法を適用できる関数のクラスにおいて、関数の種類を増やし、今回の実験と同様の知見が得られることを確認する必要がある。

#### 参考文献

- [1] R. Tanese: Distributed genetic algorithms, *Proc. of the 3rd international conference on genetic algorithms*, pp.434-439, 1989.
- [2] D.E. Goldberg: *Genetic algorithms in search, optimization and machine learning*, Addison Wesley, 1989.
- [3] 濱野賢治, 内田健, 吉野純一: 分散遺伝的アルゴリズムにおける移住低減に関する検討, 第8回情報科学技術フォーラム, pp.235-238, 2009.

表3. 移住間隔1

(a)Griewank			(b)Rastrigin			(c)Ridge		
島数	速度 bit	頻度 回数	島数	速度 世代	頻度 回数	島数	速度 bit	頻度 回数
4	12	8	4	16657	30	4	9	0
8	10	9	8	15078	30	8	11	0
16	8	13	16	15973	30	16	11	0
32	8	11	32	24009	30	32	13	0
64	9	10	64	27850	30	64	11	0
256	1	29	256	13019	30	256	0	30

表4. 移住間隔1→5

(a)Griewank			(b)Rastrigin			(c)Ridge		
島数	速度 bit	頻度 回数	島数	速度 世代	頻度 回数	島数	速度 bit	頻度 回数
4	13	6	4	28960	30	4	16	0
8	10	9	8	27520	30	8	17	0
16	8	14	16	29114	30	16	20	0
32	8	10	32	27404	30	32	20	0
64	8	13	64	50128	30	64	19	0
256	2	25	256	20653	30	256	6	30

表5. 移住間隔1→10

(a)Griewank			(b)Rastrigin			(c)Ridge		
島数	速度 bit	頻度 回数	島数	速度 世代	頻度 回数	島数	速度 bit	頻度 回数
4	11	8	4	32034	30	4	17	0
8	9	10	8	34660	30	8	19	0
16	9	12	16	31273	30	16	20	0
32	8	11	32	30661	30	32	23	0
64	7	17	64	55288	30	64	22	0
256	1	27	256	32087	30	256	25	0

表6. 移住間隔1→30

(a)Griewank			(b)Rastrigin			(c)Ridge		
島数	速度 bit	頻度 回数	島数	速度 世代	頻度 回数	島数	速度 bit	頻度 回数
4	10	10	4	38716	30	4	18	0
8	10	9	8	36477	30	8	22	0
16	7	15	16	44470	30	16	24	0
32	8	12	32	41052	30	32	25	0
64	9	12	64	88754	30	64	26	0
256	1	28	256	72675	30	256	79	0

表7. 移住間隔1→50

(a)Griewank			(b)Rastrigin			(c)Ridge		
島数	速度 bit	頻度 回数	島数	速度 世代	頻度 回数	島数	速度 bit	頻度 回数
4	10	10	4	38716	30	4	19	0
8	10	9	8	36477	30	8	23	0
16	7	15	16	44470	30	16	25	0
32	8	12	32	41052	30	32	27	0
64	9	12	64	88754	30	64	31	0
256	1	28	256	72675	30	256	109	0