2個体分散遺伝的アルゴリズムの移住低減に関する検討
A study on Dual individual distributed genetic algorithm with reduced migration

s06608　松尾遼　　指導教員　内田健

概要
本稿では2個体分散遺伝的アルゴリズム（Dual DGA）の並列化の効果を上げるために移住の少ない並列 モデルを検討する．並列効果の妨げとなる移住量を低減するため，移住間隔を進化途中で変更した．その結果，Ridge関数のような単峰性の関数に対して移住間隔を固定した場合と同等の最適解発見回数を維持できることを確認した．

1.　はじめに

GAの並列モデルについて数多くの研究がなされ，その一つとして個体の母集団を複数の島に分割し，各々の島に対して遺伝的操作を独立に適用する分散遺伝的アルゴリズム（島モデル:DGA）が提案された[1]．

他のGAと違い，DGAでは解検索に必要とされる交叉率などのパラメタ設定に加えて母集団の分割数（島数）や移住間隔などのパラメタ設定が必要となる．この問題に対して 2個体分散遺伝的アルゴリズム（Dual DGA）は解検索性能を保ちつつ必要なパラメタ設定を2個にまで削減している[2]．

しかし，Dual DGAでは各島の個体数が最小の2個であり，母集団を最大分割することからMPI（Message Passing Interface）などを用いてクラスタ型並列計算機上で実行した場合，並列化の効果が低減されるおそれがある．

そこで，移住間隔を途中で変更したときに解探索性能に与える影響について数値実験の結果を示す．

2.　移住間隔の解探索性能への影響

Dual DGAは必要なパラメタ設定 の数が少ないため，設定 項目の1つである移住間隔は解探索性能に直結する重要なパラメタであると考えられる．そこでこの移住間隔 に注目し，移住間隔を変更した場合の解検索性能について数値実験を行う．

2-1.　実験方法

移住間隔がDual DGAの解探索性能 に与える影響を調べるため，異なる移住間隔の設定ごとに解探索性能がどのように変化するかを実験する．そのため，Dual DGAを主にベンチマークテストで使われるテスト関数に使用することで移住間隔が解探索に与える影響を調べる．

今回用いるテスト関数は Griewank関数，Rastriigin関数，Ridge関数の3つで，いずれも30次元の関数であり，最適解は0である．

$F\,1$:Rastrigin関数

$$F1 = 10n + \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i))$$

$$-5.12 < x_i \leq 5.12, n = 30$$

$F\,2$:Ridge関数

$$F2 = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$$

$$-64 < x_j \leq 64, n = 30$$

$F\,3$:Griewank関数

$$F3 = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \left(\cos\left(\frac{x_i}{\sqrt{i}} \right) \right)$$

$$-512 < x_i \leq 512, n = 30$$

上記のテスト関数を用いて移住間隔がDual DGAに与える影響を調べる．移住間隔を(1, 5, 10, 50)の各値に設定し，各世代ごとの最適解を記録する．この計算を世代数が 100,000世代に到達するまで行う．この計算を各移住間隔について10回づつ試行する．実験パラメタの設定は個体数 512，島数256，遺伝子長30×20ビット，交叉率は1.0，突然変異率は1/遺伝子長，移住率は0．5，交叉方法は一点交叉に設定した．なお，紙面の都合上，解収束特性のRastrigin関数のグラフ及び最適解の発見世代と発見回数の表は割愛する．

2-2.　実験結果

表1，2はRidge関数とGriewank関数の最適解の発見頻度と最適解の発見世代の平均を各移住間隔ごとに表したものである．

図1，2はこのRidge関数の関数値の算出結果である．横軸に世代数，縦軸に関数値 を示し，10回の計算結果の平均値を4000世代目までプロットしたものである．

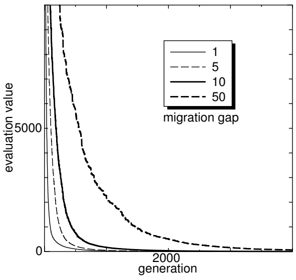

図1　Ridge収束特性

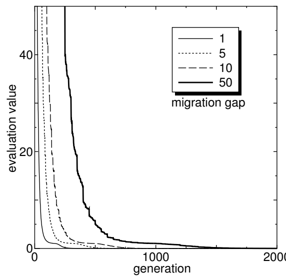

図2　Griewank収束特性

図1，2によると移住間隔が短いほど解 の収束が早いことが測定結果からわかる．移住間隔が 10世代以下であればおおむね1000世代で最適解に近い値まで収束している ように見える．これはRidge関数だけでなく，Griewank関数及びRastrigin関数も同様の傾向が見られた．

しかし，表1によるとRidge関数では1000世代付近では最適解には収束せず，1世代毎に移住を行った場合では最適解の算出に平均で15,992世代かかり，50世代毎に移住を行った場合には100,000世代までかかった．50世代毎に移住を行った場合には1世代毎に移住を行った場合と比べて約 6倍の計算が必要にだったことになる．また，最適解の発見回数も 50世代間隔は10回の試行のうち3回しか最適解を発見できていない．

表2を参照すると，Griewank関数ではRidge関数とは逆に，移住間隔が長い方が最適解の発見回数が上がる傾向が見られ，最適解の発見世代も移住間隔が長いほど早い段階で発見される傾向が見られた．

Rastrigin関数についてはGriewank関数と同様に移住間隔が長いほど最適解の発見世代が早くなる傾向がみられたが，移住間隔に関わらず，全ての試行で最適解を発見している．

表1　Ridge関数の最適解の発見回数と発見世代の平均

移住間隔	発見回数	発見世代
1 世代毎	10	15，992
5 世代毎	10	28，449
10 世代毎	10	37，787
50 世代毎	3	97，389

表2 Griewank関数の最適解の発見回数と発見世代の平均

移住間隔	発見回数	発見世代
1 世代毎	4	60, 450
5 世代毎	8	21, 753
10 世代毎	10	2, 913
50 世代毎	10	4, 843

これらの結果から移住間隔を短く設定することによって解の収束を早める効果があることが判明した. しかし, 最適解の発見頻度や, 発見世代への影響は, Ridge関数と Griewank関数 , Rastrigin関数で傾向が異なる. この違いは Ridge関数が単峰性, Griewank関数, Rastrigin関数が多峰性の性質を持つ関数であることが原因であると推測できる. そのため, 移住間隔が解探 索性能に対して与える影響は, 解の収束速度に対して効果があるが, 解探索性能が向上するとは限らないことがわかった.

3. 移住間隔を途中での変更

多峰性の性質を持つ関数に対しては移住間隔を長く設定しても最適解の発見頻度が悪化しないことがわかった.

そのため, 移住量を低減するためには, 移住間隔を長く設定しても解探索性能に影響が少ない多峰性 の関数はともかく, 移住感覚の変更によって影響を受けやすい単峰性の性質を持つ関数への対策が必要となる.

そこで本節では, 対象問題の性質が解らない場合でも移住量の増加を最低限に 留めつつ, 最適解を算出するために, 移住間隔の計算途中での変更を提案する.

3-1. 実験方法

移住間隔を計算途中で少ない数値から大きい数値に変更を行うことにより, 移住間隔を1世代毎に固定した場合よりも移住量を低減することができ, 結果的に計算ノード間通信を抑制できるのではないかと考えられる.

今回の実験では 移住間隔の変更が解探索性能に与える影響について調査を行う. 2-2の実験同様 , 3つのテスト関数を用いて最適解を算出し , 最適解の発見回数 , 平均発見世代数について調べる.

また, 移住間隔の変更をどのように行うかについては, 初期の移住間隔を1に設定し, 変更が行われるまでは 1世代毎に移住が行われるようにする . あらかじめ 世代数が 10世代目 , 20世代目 , 30世代目, 40世代目, 50世代目, 100世代目に到達したところで移住間隔を50に変更, 50世代毎に移住が行われるように設定し, 規定世代数に達した時点で移住間隔の変更を行う.

また, この実験で用いるテスト関数は, 2-2で用いた3つのテスト関数, Griewank関数, Rastriigin関数, Ridge関数の3つを再び用いる. また, 移住間隔以外のパラメタ設定についても同じ数値を用いることとする . この設定で 100,000世代に到達するまで計算を行い, 10回試行する.

3-2. 実験結果

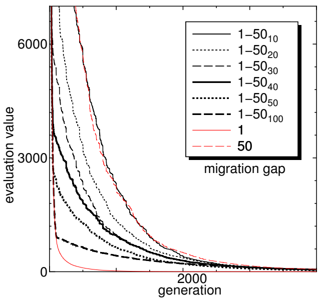

図3 Ridge関数収束特性

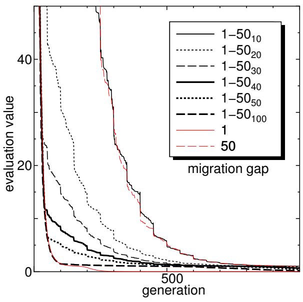

図4 Griewank関数収束特性

図3, 4はこのRidge関数及びGriewankの関数値の算出結果である. 横軸に世代数, 縦軸に関数値を示し, 10回の計算結果の平均値を4000世代目までプロットしたものである.

表3, 4はRidge関数及びGriewank関数の最適解の発見回数と発見世代を, 移住間隔を変更した世代ごとにまとめたものである.

Ridge関数の場合, 40世代目で変更した場合を除いては, 移住間隔を50世代目で固定した場合と比べて最適解の発見回数が上がっていることがわかる.

1世代毎に移住を行った場合と比較すると発見回数 に不安が残るものの, 1世代毎に移住を行う期間を 50世代目以上にすることである程度発見回数が高くなっている. 解収束特性のグラフと比較すると50世代目, 100世代目ともに解の収束が進んでいる段階であるため, 移住 間隔の変更は解の収束がある程度進んだ段階で行うのが効果的であると思われる.

また, Griewank関数は全ての試行で最適解を発見したが, 発見世代数は50世代毎に移住を行った場合とあまり変化はなかった.

Rastrigin関数については Griewank関数と同様に全ての試行で最適解を発見したが, 発見世代数は 50世代毎に移住を行った場合よりも増加する傾向が見られた.

表3 Ridge関数の最適解の発見回数と発見世代の平均

変更世代	発見回数	発見世代
10 世代目	4	96, 679
20 世代目	4	98, 005
30 世代目	6	96, 148
40 世代目	2	98, 230
50 世代目	8	97, 527
100 世代目	7	95, 454

表4 Griewank関数の最適解の発見回数と発見世代の平均

変更世代	発見回数	発見世代
10 世代目	10	4941. 5
20 世代目	10	4722. 5
30 世代目	10	4595. 8
40 世代目	10	4675. 1
50 世代目	10	4605. 1
100 世代目	10	4390. 5

4. おわりに

移住間隔を途中で変更した場合, 単峰性の性質を持つRidge関数は50世代毎の結果と比較すると最適解の発見回数が上昇した. 移住間隔を1世代毎に設定した場合と比較すると移住量を約1/8ほどに低減することができたことになる.

しかし, 多峰性の性質を持つGriewank関数やRastrigin関数に対してはかえって計算量を増やしてしまい, あえて使用するメリットはないことがわかった.

移住量低減のためには, 移住間隔の途中での変更は単峰性の関数を解く場合使用する. 多峰性の性質を持つ関数には移住間隔の数値を大きく設定しておき, 途中での変更はしないことが有効である.

参考文献

[1] R.Tanese: Distributed genetic algorithms, Proc. of the 3rd international conference on genetic algorithms, pp.434-439.

[2] 廣安知之, 三木光範, 佐野正樹, 谷村勇輔, 濱崎雅弘 : 2個体分散遺伝的アルゴリズム , 計測自動制御学会論文集, Vo138, No11, 990-995, (2002)