

Adaptive Distributed Genetic Algorithm System

Tomoyuki HIROYASU*, Mitsunori MIKI**, Kouta AKATSUKA***

(Received February 5, 2001)

Proposed in this paper is a new system of genetic algorithm called "Adaptive Distributed Genetic Algorithm System: ADGAS". The problems to determine the values of design variables that maximize or minimize the objective function under definite constraints are called optimization problems. If the value of an arbitrary design variable of the optimum solution is equal to that obtained by minimizing or maximizing the objective function keeping other design variables arbitrarily constant, these design variables of the problem are independent each other. In this case, the search space can be reduced and the total calculation cost can be suppressed. In the proposed system, the given problems are analyzed first and independency of design variables is investigated. Then, three algorithms are applied to the problems concerning independency among the design variables. The proposed system is applied to solving the numerical test functions to discuss and examine the effectiveness of the proposed system. Through numerical examples, it is clarified that the proposed system has high capability for problems in which design variables are independent. Even in problems whose design problems are not independent, the proposed system has almost the same search ability for distributed genetic algorithms. From these results, it can be concluded that the proposed system is superior to conventional distributed genetic algorithms.

Key words : Evolutionary strategy, Genetic Algorithms, Distributed model, Landscape

キーワード : 進化戦略, 遺伝的アルゴリズム, 分散モデル, ランドスケープ

適応的分散遺伝的アルゴリズムシステム

廣安知之・三木光範・赤塚浩太

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¹⁾。この手法は、従来の最適化手法の適用が困難であった離散的問題などに適用できるうえ、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²⁾。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を実行する分散GA(Distributed GA: DGA)³⁾がある。

DGA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して遺伝的操作を行う。また、各島内での多様性の維持などのため、他の島の解と情報を交換する移住と呼ばれる操作を行う。移住では、

* Department of Knowledge Engineering and Computer Sciences, Doshisha University, Kyoto
Telephone:+81-774-65-6434, Fax:+81-774-65-6796, E-mail:mmiki@mail.doshisha.ac.jp

** Department of Knowledge Engineering and Computer Sciences, Doshisha University, Kyoto
Telephone:+81-774-65-6638, Fax:+81-774-65-6796, E-mail:tomo@is.doshisha.ac.jp

*** Graduate Student, Department of Knowledge Engineering and Computer Sciences, Doshisha University, Kyoto
Telephone:+81-774-65-6716, Fax:+81-774-65-6796, E-mail:kouta@mikilab.doshisha.ac.jp

あらかじめ定められた移住間隔ごとに、移住率とよばれる割合に基づいて島内の一部の個体の情報を交換する。移住に関して、移住の頻度、移住先や移住個体の選択方法が解に与える精度など、様々な研究が行われている^{4) 5)}。移住の頻度に関しては、各サブ母集団が一斉に移住を行う同期移住や、サブ母集団ごとに移住のタイミングが異なる非同期移住などがあり、一般的には移住先が移住機会ごとにランダムに決定され同期的に移住するモデルが使用される⁶⁾。

また、DGA は 並列化による計算時間の減少の他に、SGA と比較して解を求めるまでの計算量自体が減少するという特徴を持つ⁷⁾。

これまでに述べたように GA の問題点の一つは、計算コストが高い点にある。DGA 等の並列化手法によって計算時間を減少させる事は可能であるが、計算量は依然多く、計算機に負荷をかける事に変わりはない。そこで、まず解決すべき問題の特性を解析し、その特性に基づいて探索空間を狭めることが可能であれば、計算量を減少させることができ、計算コストを減少させることができる。本研究では、設計変数間に依存関係が無い問題では設計変数毎に独立して解を求めることができる点に注目した。そうする事で設計変数の探索すべき組み合わせの数を減少させることが可能となる。そこで、まず対象問題における設計変数間の依存関係を解析し、その結果に応じて依存関係の無い設計変数を独立に探索するアルゴリズムを提案する。本研究で提案するアルゴリズムを適応的分散遺伝的アルゴリズムシステム (Adaptive Distributed GA System: ADGAS) と呼び、数値実験を通じてその有効性を検討する。

2. 分散遺伝的アルゴリズム

2.1 遺伝的アルゴリズム⁸⁾

遺伝的アルゴリズム (Genetic Algorithm: GA) は生物の遺伝と進化を模擬した多点探索手法であり、探索点は個体 (individual) と呼ばれる。個体には染色体 (chromosome) が与えられ、この染色体により問題空間のある 1 つの状態を表す。染色体は複数個の遺伝子 (gene) からなる。問題空間上の 1 点を GA の個体へ変換する事をコーディングという。Fig. 1 に問題空間上の点 (1,5) がコーディングによってある個体に変換される様子を示す。

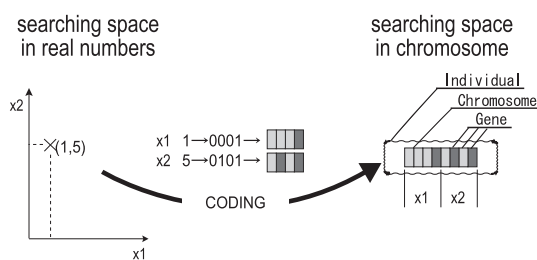


Fig. 1. Coding from real numbers to chromosome.

個体の集合を母集団 (population) と呼び、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。これら選択、交叉、突然変異は遺伝的操作と呼ばれる。GA では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 分散遺伝的アルゴリズム

DGA 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、一定世代ごとに異なるサブ母集団間で移住と呼ばれる複数個体の交換を行う。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。ここで、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。DGA と移住の概念を Fig. 2 に示す。

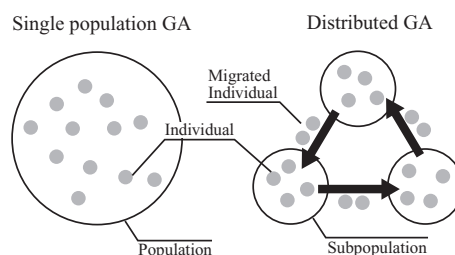


Fig. 2. Single population GA and Distributed GA.

DGA の特徴として、並列化効率の高さと母集団の分割による利点がある。すなわち、各サブ母集団を各プロセッサで処理することにより、各プロセッサ間の通信は移住による個体の情報交換のみですむ。このため、並列化効率が非常に高い。また、母集団を分割することにより各島で個体の異なった遺伝子が進化して

いる場合、移住によって他の島の個体と混ざり合い交叉を行うことで、優れた遺伝子同士が集まった、非常に優れた個体が誕生する可能性がある⁷⁾。

3. 適応的分散遺伝的アルゴリズム

3.1 概要

すでに述べた通り、遺伝的アルゴリズムの一つの問題点は、繰り返し計算の多さによる計算コストの高さである。あらかじめ問題を解析し、探索空間を狭めることができるならば、計算量を減少させることができ、GAによる解探索が有効となる。本研究ではその手法として、問題の設計変数間の依存関係に着目した。設計変数間の依存関係とは、対象問題を最適化する際、設計変数ごとに独立して最適化が可能か否か、である。すなわち、設計変数間に依存関係が無い場合には、GAにおいても各設計変数を個別に探索することが可能となる。そのため、探索すべき組み合わせが減少することとなり、探索空間を狭めることが可能となる。

提案する適応的分散遺伝的アルゴリズムシステム (Adaptive Distributed Genetic Algorithm System) は、まず問題の設計変数間の依存関係を解析し、その結果に応じて適用するアルゴリズムを選択するシステムである。問題の解析は解の探索中に数度行い、適応的にアルゴリズムを変化させる。適用するアルゴリズムは、設計変数間に依存関係が無い場合に探索領域分割型 GA (DGA/Divided Searching area: DGA/DSa) を行うアルゴリズム、依存関係が有る場合に探索領域可変型 DGA (DGA/Variable Feasible region: DGA/VFr) を行うアルゴリズム、及び一部の設計変数間に依存関係が有る場合に行う DGA/DSa と DGA/VFr の複合アルゴリズムである。

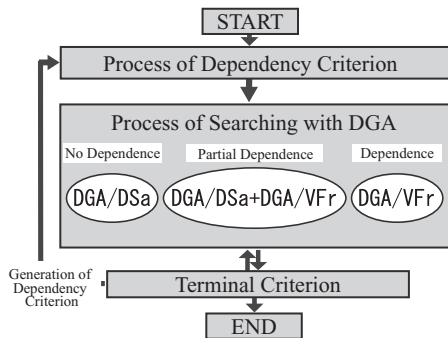


Fig. 3. Outline of ADGAS.

3.2 依存関係評価

ここで、依存関係評価プロセスにおける対象問題の設計変数間に有る依存関係の有無を評価するアルゴリズムについて以下に説明する。

依存関係の評価はすべての設計変数を同時に行うのではなく、2変数ずつ行う。このため N 設計変数の場合 $N(N-1)/2$ 回、以下の操作を繰り返す。

Step 1 対象となる2設計変数に対して、対象設計変数の探索領域内に評価を行うための点を100点設定する。この100点は以下のように設定する。まず、各設計変数を5分割する。次にこの5分割した点の近傍に点を設定する。近傍とは具体的には、問題空間からGAの遺伝子型にエンコードする際に生まれる差分で、1設計変数当たり10bitでエンコードするなら対象設計変数の実行可能領域/ 2^{10} だけ離れた点である。

この各設計変数の10点の交叉点を評価点とする (Fig.4)。

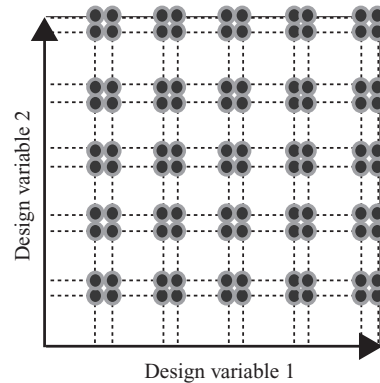


Fig. 4. 100 evaluation points.

Step 2 Step 1で設定した点の適合度値を求める。この際対象外の設計変数には任意の定数を用いる。

Step 3 Step 2で計算した100点の座標と適合度値をもとに依存関係を評価する。まず対象となる設計変数の片方の値が同じ点に対し適合度値の順にランク付けを行う。残った設計変数の値が同じ点のランクがすべて同じであれば、その設計変数間には依存関係がなく、同じでなければ依存関係がある。

$f(x, y) = x^2 + y^2 + z^2 (0 \leq x, y, z \leq 10)$ を例に評価を行うための点を1設計変数当たり4点取った場合の解説を行う。この関数は依存関係の無い関数である。なお、簡単のため近傍点は1.0だけ離れた点とした。

3 設計変数なので、 $3(3-1)/2 = 3$ 回次の操作を繰り返す。

Step 1 1 回目に対象となる設計変数は (x,y) である。まず x の探索領域 ($0 \leq x \leq 10$) 内に 4 点を取ると、 $(5,6,9,10)$ となる。 y も探索領域が同じなので、同じ点となる。

Table 1. Evaluation values of 16 points of $f(x,y)$.

$x \setminus y$	5	6	9	10
5	50	61	106	125
6	61	72	117	136
9	106	117	162	181
10	125	136	181	200

Step 2 次に $(5,5),(5,6),\dots,(6,5),(6,6),\dots,(10,10)$ の評価値を求めるが、この時対象外の変数である z には任意の定数 0 を用いる。すると、評価値は Table1 のようになる。

Step 3 次に、片方の設計変数 x が同じ値の点に対し、評価値の順にランクをつける。 $x=5$ の場合を例にとると、 y の値 $5,6,9,10$ に対し評価値は $50,61,106,125$ となり、 y の値 $5,6,9,10$ の順に評価値が低い。そこで、 y の各値のランクは $1,2,3,4$ となる。同様に、 $x=6,9,10$ の場合もランク付けを行うと、 Table2 のようになる。ここで、 x の値にかかわらず、 y の値のランクは $1,2,3,4$ となり同順である。したがって、この関数の設計変数 (x,y) 間には依存関係は無い。

Table 2. Ranking y .

$x \setminus y$	5	6	9	10
5	1	2	3	4
6	1	2	3	4
9	1	2	3	4
10	1	2	3	4

次に、 $g(x,y,z) = (x-y-z)^2 (0 \leq x,y \leq 10)$ を例に $f(x,y,z)$ と同じ条件で解説を行う。この関数は依存関係の存在する問題である。なお、STEP 1 は $f(x,y,z)$ と同じなので省略する。

Step 2 次に $(5,5),(5,6),\dots,(6,5),(6,6),\dots,(10,10)$ の評価値を求めると、 Table3 のようになる。

Table 3. Evaluation values of 16 points of $g(x,y)$.

$x \setminus y$	5	6	9	10
5	0	1	16	25
6	1	0	9	16
9	16	9	0	1
10	25	16	1	0

Step 3 次に、片方の設計変数 x が同じ値の点に対し、評価値の順にランクをつける。 $x=5$ の場合を例にとると、 y の値 $5,6,9,10$ に対し評価値は $0,1,16,25$ となり、 y の値 $5,6,9,10$ の順に評価値が低い。そこで、 y の各値のランクは $1,2,3,4$ となる。 $x=6$ の場合、 y の値 $5,6,9,10$ に対し評価値は $1,0,9,16$ となり、 y の値 $6,5,9,10$ の順に評価値が低い。そこで、 y の各値のランクは $2,1,3,4$ となる。同様に $x=9,10$ の場合もランク付けを行うと、 Table4 のようになる。ここで、 x の値によって、 y の値のランクは異なり同順にはならない。したがって、この関数の設計変数 (x,y) 間には依存関係が有る。

Table 4. Ranking y .

$x \setminus y$	5	6	9	10
5	1	2	3	4
6	2	1	3	4
9	4	3	1	2
10	4	3	2	1

3.3 探索領域分割型遺伝的アルゴリズム

設計変数間に依存関係が無いと評価された場合には、提案する手法である探索領域分割型 GA(DGA/Divided Searching area: DGA/DSa) を用いる。ここではそのアルゴリズムについて説明する。

DGA/DSa では、設計変数と同等数の分割母集団を準備し、各母集団ごとに探索する設計変数を決定し、それぞれ独立して探索する。各島内では担当している設計変数のみの探索を行う。 Fig.5 に概念図を示す。

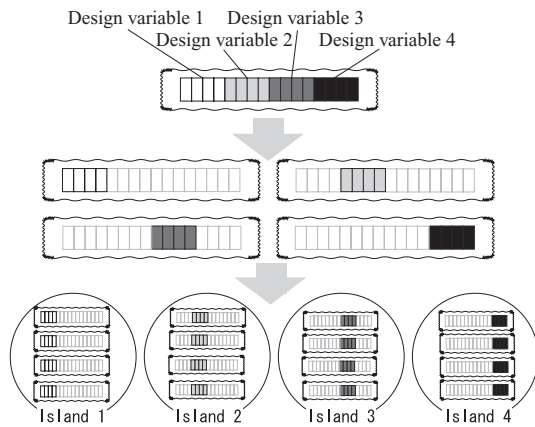


Fig. 5. Outline of DGA/DSa.

Fig.5 では、4 変数の場合を示している．そのため、分割母集団数は 4 であり、例えば設計変数 3 は島 3 が担当する．

また、各島で対象とする設計変数が異なるため、移住という各島間で個体の交換を行う操作はこのモデルでは必要ない．

DGA/DSa は基本的には対象となる設計変数を各島で SGA を用いて解く．しかし SGA の操作の一部は DGA/DSa 用に変更する必要がある．以下に、変更した操作を説明する．

● 交叉，突然変異

SGA における、交叉は染色体中のすべての遺伝子の中から、交叉点と呼ばれる交叉を行う点を選び、この点を元に点より前と後で異なった親の遺伝子を引き継ぐ．しかし、DGA/DSa では交叉点を選ぶ際に、対象とする設計変数の遺伝子中から選び、その点を元に通常の交叉を行う．また、突然変異に関しても同様で、SGA ではすべての遺伝子の中から突然変異点を選ばれるが、DGA/DSa では対象とする設計変数の遺伝子中から突然変異点を選ばれる．Fig.6 に交叉、Fig.7 に突然変異の様子を示す．

● 評価

評価では、SGA においては染色体内のすべての遺伝子からすべての設計変数の値がデコードされ、評価が行われるが、DGA/DSa では対象となる設計変数のみ、GA の個体からデコードされ、それ以外の設計変数ではあらかじめ用意した任意の定数を用いて評価を行い、適合度の計算が行われる．Fig.8 は、設計変数が 3 の場合で、設計変数 1 を対象とした個体と設計変数 2 を対象とし

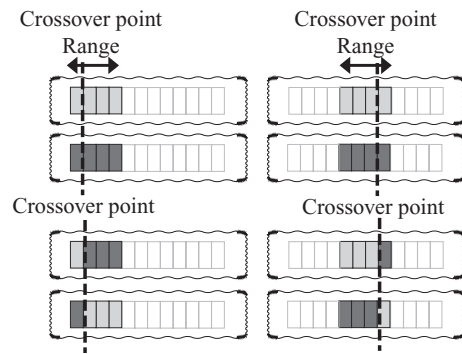


Fig. 6. Crossover of DGA/DSa.

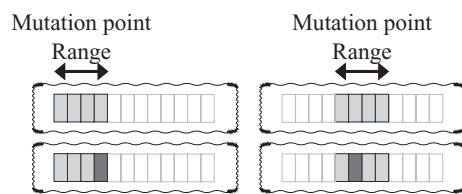


Fig. 7. Mutation of DGA/DSa.

た個体の適合度の計算を示した．設計変数 1 を対象とした場合、対象とする設計変数からは値をデコードする (x) がそれ以外の設計変数はあらかじめ用意した任意の定数 (a) を用いる．

● エリート保存

エリート保存は DGA と同じように各島で 1 個体ずつ行うが、各エリートは対象とする設計変数の値しか持っていない．

● 最適解の生成

DGA/DSa では、各島内の最適個体は対象とする設計変数のみが最適である．そのため、各世代での最適解を求める際には、次の操作を行う必要がある．すべての島のエリートから対象としている設計変数値を結合し、すべての設計変数が最適な

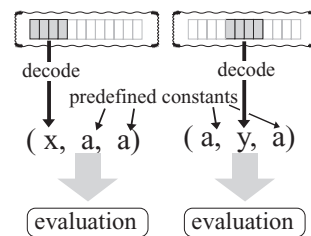


Fig. 8. Evaluation of DGA/DSa.

個体を1個生成する．この個体の全設計変数を対象として評価を行い適合度を求める (Fig.9)．その結果が現在このGAが持つ最適解であり，この個体を用いて終了判定などに用いる．

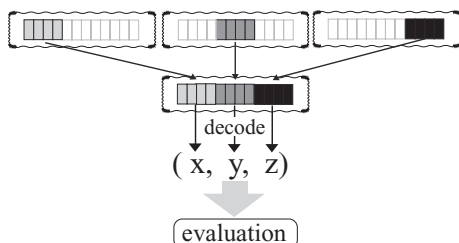


Fig. 9. Best value of DGA/DSa at each generation.

3.4 適応的分散遺伝的アルゴリズムシステムにおける遺伝的操作

3.4.1 設計変数の探索領域の適応的な変化

ADGAS では依存関係評価プロセスで設計変数の探索領域を適応的に変化させている．この方法として，GAを行っている全個体の持つ設計変数の値をもとに，その最大値と最小値を次に依存関係を評価する世代までの探索領域とする方法を用いる．GAの個体は全体として徐々に最適解に近づくため，世代を進めるにつれて探索領域は小さくなる．

これにより初期の探索領域内で依存関係が有る場合でも，探索を進めることにより探索領域内の依存関係が存在しなくなる可能性がある．探索領域内に依存関係が存在しないと，DGA/VFr に比べて非常に性能が良い DGA/DSa を用いることができ，解探索能力の大幅な向上が期待できる．Fig.10 の例の場合，初期状態では設計変数の探索領域内に依存関係が有るため DGA/DSa は適応できず DGA/VFr を行う．しかし，探索を進めると依存関係の有る部分が設計変数の探索領域外になり，DGA/DSa を用いることが可能となり，解探索能力の向上が期待できる．

しかし，GAの世代を進めるに伴って探索領域がなくなってしまう場合がある．これは島内の個体数が少ない場合に顕著で，探索領域が極端に小さい場合や1点となってしまった場合，依存関係の評価が正しく行われなくなる可能性がある．このため ADGAS では探索領域が1点となった場合，再び探索領域を最大にとり，依存関係の評価に影響が出ないようにしている．またこの操作は，GAが最適解ではなく局所解に到達し探索領域内に最適解が含まれない場合にも，探索領

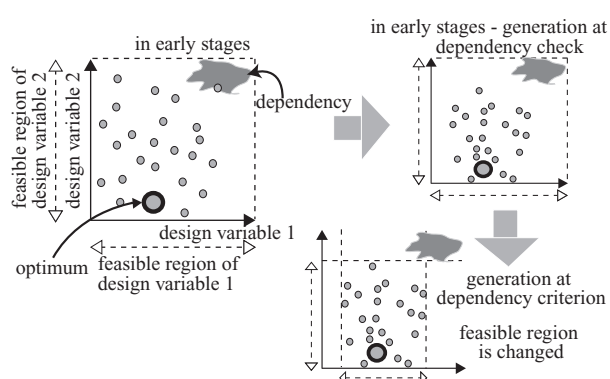


Fig. 10. The case where searching area has partly dependency.

域を設定しなおすことで局所解から脱出する可能性を残すために有効である．

3.4.2 適応的な構造変化

ADGAS では設計変数間に依存関係がある問題には DGA/VFr を，依存関係が無い問題には DGA/DSa を用いる．ただし，依存関係が無い問題に適用する DGA/DSa にも，探索領域可変型の仕組みを用いる．つまり，突然変異と交叉を探索領域を超えないように変更する．また，設計変数間の一部に依存関係が有る問題の場合は，DGA/DSa と DGA/VFr を組み合わせた手法を用いる．DGA/DSa では依存関係の有る設計変数同士を1つのグループとして1島で探索を行うが，ADGAS では1つのグループを1島で探索するのではなく，グループ数だけ島を用意し，この島間では DGA/VFr を用いる．これは，グループとなる設計変数の探索を1つの島で行うと，島内の個体数を一定にした場合，設計変数当たりの個体数が変化し解探索能力に影響が生じるからである．また，島の個体数を可変としても，設計変数間の依存関係が多くの設計変数に及べば及ぶほど SGA に近づき，解探索能力に影響が出る可能性がある．そこで，1つのグループを1島では探索せず，グループにした設計変数の数だけ島を用意し，この島間で DGA/VFr を行うことにより，これらの問題を解消する．

3.5 探索領域可変型分散遺伝的アルゴリズム

探索領域を可変にした DGA である探索領域可変型分散遺伝的アルゴリズム (DGA/Variable Feasible region: DGA/VFr) について述べる．DGA/VFr では，探索領域を可変にするために，交叉，突然変異の手法を若干変更している．これは，交叉突然変異では，そ

れまで個体が保持していた設計変数の値が大きく変わり、個体の持つ解が探索領域外になる可能性が有るからである。そこで、今回用いた交叉、突然変異について詳しく述べる。

• 交叉

交叉の場合、設計変数の値が変わるのは交叉点を持つ設計変数のみである。そのため、交叉の後の点が探索領域外の場合は、交叉点をその設計変数の直後に移動して設計変数間の交叉とし、設計変数の値が探索領域内に存在するようにしている。

• 突然変異

突然変異では変異後の設計変数の値が探索領域外に存在する場合、設計変数の値が探索領域内に存在するまで突然変異点を変化させることで、設計変数の値が探索領域を超えないようにしている。

4. 数値実験と結果

今回提案する、依存関係が無い設計変数間において設計変数を各島で独立に解く手法が有効であることを示すために、数値実験を行う。このため数値実験1では、対象問題として全ての設計変数間に依存関係が無い Rastrigin 関数と Schwefel 関数、一部の設計変数間に依存関係のある Original1 関数を用いる。また、数値実験に用いた GA(Genetic Algorithms) は通常の DGA(Distributed GA) と提案するアルゴリズムである DGA/DSa(DGA/Divided Searching area) である。

続いて、依存関係を評価しその結果に応じて適応するアルゴリズムを変化させるシステムである ADGAS(Adaptive Distributed Genetic Algorithm System) の有効性を検証するために数値実験2を行う。数値実験2では先ほど数値実験1で使用した対象問題に加え、全ての設計変数間に依存関係がある Rosenbrock 関数、Griewank 関数、Ridge 関数と、探索領域の一部に依存関係がある Original2 関数を対象問題として用いた。結果は全て 20 試行の平均を示す。

4.1 対象問題

4.1.1 設計変数間に依存関係が無い問題

Rastrigin 関数は、式 (1) で表される関数で、設計変数間に依存関係が無い。この関数はすべての設計変数値が 0 の際最小値 0 を取り、その周辺に格子状に複数の準最適解を持つ。

Schwefel 関数は、式 (2) で表される関数で、設計変数間に依存関係が無い問題である。この関数はすべて

の設計変数値が 412 の際最小値 (418.98276403 × 設計変数数) を取る。

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

$$-5.12 \leq x_i < 5.12$$

$$f_{Schwefel} = \sum_{i=1}^n \{-x_i \sin \sqrt{|x_i|}\} \quad (2)$$

$$(-512 \leq x_i \leq 512)$$

4.1.2 一部に依存関係が有る関数

Original1 関数は式 (3) で表される関数で、第 2n 番目と第 2n+1 番目の設計変数間に依存関係が有る関数である。最適解はすべての設計変数値が 1 の際最小値 0 を取る。

Original2 関数は式 (4) で表される関数で、すべての設計変数間に依存関係が有る。この関数は、Rastrigin 関数と Ridge 関数を組み合わせた関数で、設計変数の探索領域の一部に依存関係が有るが、最適解付近の中心部には依存関係が無い。最適解は、すべての設計変数値が 0 の際最小値 0 を取る。

$$f_{original1} = \sum_{i=1}^{n/2} \{100(x_{2i-1} - x_{2i}^2)^2 + (x_{2i} - 1)^2\} \quad (3)$$

$$(-2.048 \leq x_i \leq 2.048)$$

$$f_{original2} = \begin{cases} g(x_i) & \text{if } g(x_i) > h(x_i) \\ h(x_i) & \text{if } g(x_i) \leq h(x_i) \end{cases} \quad (4)$$

$$g(x_i) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$$

$$h(x_i) = 5\sqrt{n} \{10n + \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i)]\}$$

$$(-64 \leq x_i \leq 64)$$

4.1.3 設計変数間に依存関係が有る関数

Rosenbrock 関数は、式 (5) で表される関数で、すべての設計変数間に依存関係が有る問題である。この関数はすべての設計変数値が 1.0 の際最小値 0 を取る。

Ridge 関数は、式 (6) で表される関数で、設計変数間に依存関係が有る問題である。この関数はすべての設計変数値が 0 の際最小値 0 を取る。

Griewank 関数は、式 (7) で表される関数で、設計変数間に依存が有る問題である。この関数はすべての設計変数値が 0 の際最小値 0 を取る。また、この関数は全体を粗く見ると、単峰性の依存関係が無い関数に見えるが、最適解付近を詳しく見ると多峰性の依存関係が有る関数であることがわかる。

Table 5. Used parameters.

chromo length	100
population size	600
mutation rate	0.01
crossover rate	1.0
crossover method	1-point
dimension	10

$$f_{rosenbrock} = 100 \sum_{i=2}^n (x_{i-1} - x_i^2)^2 + (1 - x_i)^2 \quad (5)$$

$$-2.048 \leq x_i < 2.048$$

$$f_{ridge} = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2 \quad (6)$$

$$(-64 \leq x_i \leq 64)$$

$$f_{griewank} = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^N \left\{ \cos\left(\frac{x_i}{\sqrt{i}}\right) \right\} \quad (7)$$

$$(-512 \leq x_i \leq 512)$$

4.2 実験 1:探索領域分割型遺伝的アルゴリズムの有効性の検証

DGA/DSa は設計変数間に依存関係の無い問題において、各設計変数を各島で独立に探索するため、DGA より探索領域が狭まり性能の向上が見込める。そこで、これを確かめるため以下の実験を行った。対象問題は、式 (1) で表される Rastrigin 関数と式 (2) で表される Schwefel 関数、式 (3) で表される Original1 関数とした。これは、Rastrigin, Schwefel 関数は設計変数間に依存関係の無い関数であり、また Original1 関数は、 $2n$ と $2n + 1$ 個目 (n =任意の整数) の設計変数間にのみ依存関係が有る関数のため、今回提案する DGA/DSa に適した関数であるからである。これらの問題を用いて、今回提案した GA である DGA/DSa と、移住率・移住間隔を変えた DGA で比較を行った。なお、本実験に用いたパラメータは、DGA/DSa と DGA に共通するパラメータとして Table5 を、残りのパラメータは Table6 を使用した。

Rastrigin と schwefel 関数の結果を Fig.11 に示す。

グラフの縦軸は解を求めるのに要した評価計算回数であり少ない方が優れている。グラフ中黒く塗りつぶされたものが今回提案する DGA/DSa による結果、白色が DGA において移住を行わない場合の結果、残りが DGA において移住率・移住間隔を変化させたもの

Table 6. Parameters of DGA and DGA/DSa.

	DGA	DGA/DSa
terminal criterion	when the optimum value is discovered	
island size	10	
migration size	0, 0.3, 0.6	—
migration interval	2, 5, 10	—

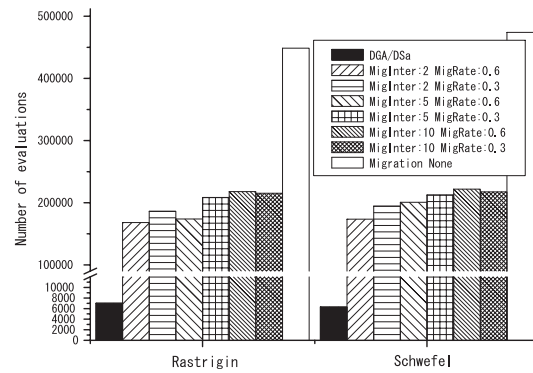


Fig. 11. Results of rastrigin and schwefel.

である。結果をみると、DGA/DSa が DGA より非常に優れているのがわかる。これは、DGA/DSa では各島で対象とする設計変数が $1/10$ になるため、探索すべき組み合わせが大幅に減少しその結果、非常に優れた結果を得ることができるからである。次に Fig.12 に Original1 関数の結果を示す。

なお、この関数に関しては真の解を求めるまで GA を行うと DGA において、10000 世代を超える場合が出たため、DGA, DGA/DSa とともに真の解との誤差が 0.001 を下回った時点で、終了としている。縦軸、凡例は Fig.11 と同様とする。この関数においても、DGA/DSa は DGA より非常に優れた結果を示している。

このように、設計変数間に依存関係が無い場合、DGA/DSa は DGA に比べて非常に優れた探索能力を持つことがわかった。また、一部の設計変数間に依存関係が有る場合も、依存関係が有る設計変数同士を 1 つのグループとみなして、DGA/DSa を適用することで、効果があることが明らかとなった。

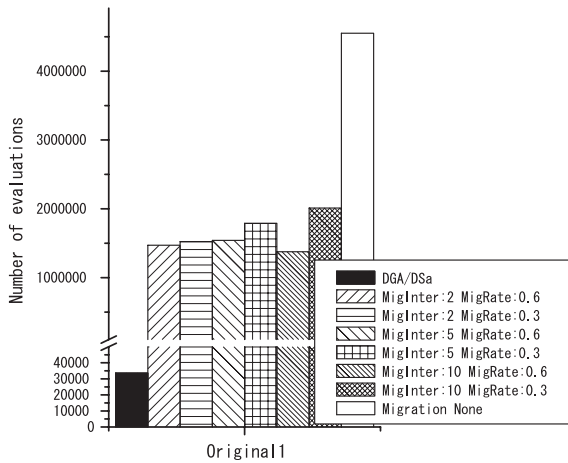


Fig. 12. Result of original1.

4.3 実験 2:適応的分散遺伝的アルゴリズムシステムの有効性の検証

ここでは、本研究で提案する手法である適応的分散GAシステムの有効性の検証を行うため、以下の実験を行う。設計変数間に依存関係が無い問題、一部に有る問題、全体にある問題に関して、提案する手法であるADGASとDGAの比較を数値実験で行い、結果を考察する。

なお、10設計変数の場合ADGASが1回の依存関係評価に必要とする評価計算回数は、 $100 \times N(N-1)/2 = (10 \times 9)/2 \times 100 = 4500$ 回である。また、すべての結果に依存関係評価に必要とした評価計算回数も含めてある。

4.3.1 設計変数間に依存関係の無い問題

本研究で提案するADGASは、設計変数間に依存関係が無い場合、探索領域分割型GA(DGA/DSa)を行うことにより、解探索の性能がDGAと比較して大幅に向上することが期待できる。そこで、設計変数間に依存関係が無い問題であるRastrigin関数(式1)とSchwefel関数(式2)を対象に実験を行い性能の比較を行った。この際用いたパラメータは、Table5とTable7に示すパラメータを用いた。

結果をFig.13に示す。

依存関係の無い問題では、ADGAS、DGAともにすべて最適解が発見できたため、グラフの縦軸には評価計算回数を用いた。少ないほど解探索能力に優れているといえる。グラフ中黒色がADGAS、それ以外は移住間隔を変えたDGAで左から移住間隔2,5,10,20,

Table 7. Parameters of DGA and ADGAS.

	DGA	ADGAS
terminal criterion	when the optimum value is discovered	
island size	10	
migration size	0, 0.3	—
migration interval	2, 5, 10, 20	—

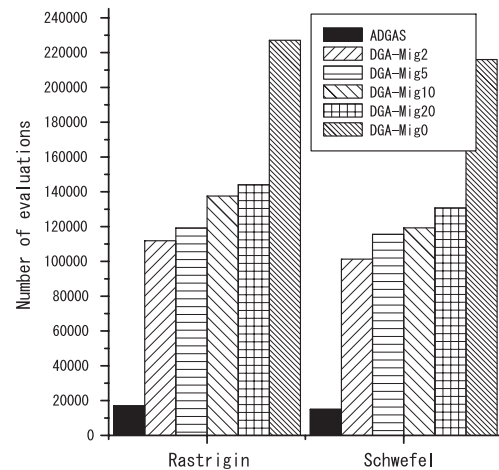


Fig. 13. Results of no dependency functions.

移住無しである。ADGASはDGAの最適なパラメータ設定と比較して1/5程度の評価計算回数で解を発見しており、非常に優れた結果を示している。

4.3.2 一部に依存関係が有る問題

ADGASでは、一定世代(本研究では100世代とした)ごとに依存関係を評価しなおす。このため、Original2関数(式4)のような、探索領域の一部に依存関係が有る問題でも、DGAと比較した場合性能の向上が期待できる。また、Original1関数(式3)のように一部の設計変数間に依存関係が有る問題でも、依存関係が有る設計変数同士を1つのグループとみなし、DGA/DSaを適応することで性能の向上が期待できる。

結果をFig.14に示す。

一部に依存関係が有る場合の関数でも、すべての場合において最適解が発見されたため、縦軸に最適解が発見されるまでの評価計算回数を用いた。先に述べたように、少ない方が解探索能力が優れている。なお、Original1関数において、移住を行わないDGAの結果が非常に悪くなったため、縦軸を200000回から

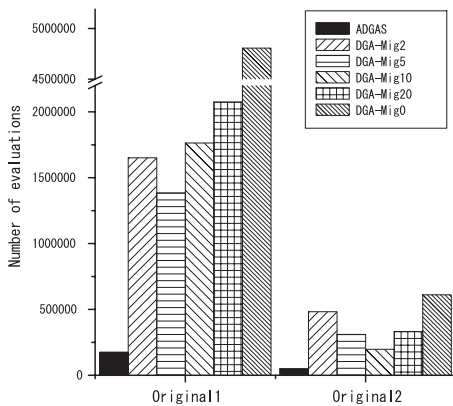


Fig. 14. Results of partly dependency functions.

450000 の間で破断させている．依存関係が無い関数同様，ADGAS が非常によい結果を示している．また，DGA では移住間隔が異なると性能に大きな差が生じ，Original1 関数と Original2 関数の結果より最適な移住間隔は問題に依存していることがわかる．

4.3.3 依存関係が有る問題

ADGAS では依存関係が有る問題においては，DGA を移住無しに行うモデルと考えると良い．したがって，ADGAS の性能は DGA の移住無しモデルに近いのではないかと予想される．ただし，ADGAS では依存関

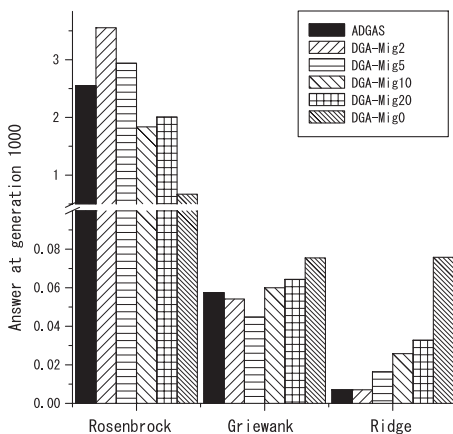


Fig. 15. Results of dependency functions.

係の評価に 10 設計変数では若干ではあるが評価計算が必要となる．とくに 100 世代おきに行っているため，終了世代が 1000 世代の場合は通常の DGA より 45000 回多く評価計算が必要となる．今回の実験は，依存関係が強いとされている Rosenbrock 関数 (5)，中程度とされている Griewank 関数 (式 7)，そして Ridge 関

数 (式 6) を用いて行った．結果を Fig.15 に示す．

依存関係がある問題では，すべての DGA,ADGAS において，最適解が発見される確率が低かったため，1000 世代での解を縦軸に用いて性能の比較を行った．結果は予想と大きく異なり，移住無しの DGA と ADGAS とでは大きく異なる結果となった．ADGAS は Ridge 関数では移住間隔が 2 世代の DGA とほぼ同じ結果となった．ADGAS は依存関係が有る問題の場合，移住を行わない DGA とほぼ同じ構造となる．

それにもかかわらず移住間隔 2 世代の DGA と同じ結果となった原因を調べるために，探索領域の絞込みに注目した．この操作は DGA では行わず ADGAS で行っている操作である．探索領域の絞込みを行わない ADGAS を行った結果を Fig.16 に示す．

グラフ中白色が ADGAS において絞込みを行わないモデル，黒色が絞込みを行う通常の ADGAS であり，これら以外は Fig.15 と同じである．

この結果，探索領域の絞込みを行わない ADGAS では，移住を行わないモデルとほぼ同じ結果となった．したがって，Ridge 関数では探索領域を絞り込むことにより，収束を加速させ良い結果を得ることができたものと考えられる．

一方，Rosenbrock 関数では ADGAS は移住を行わないモデルであるにもかかわらず，DGA の移住を行わないモデルに比べて非常に性能が悪く，DGA の移住をより多く行うモデルに近い結果となっている．これは先ほど述べた探索領域の絞込みに原因があると考えられる．そこで，Rosenbrock 関数においても探索領域の絞込みを行わない ADGAS を行ったところ，Fig.16 に示すように，移住を行わない DGA とほぼ同じ結果となった．したがって，Rosenbrock 関数において ADGAS では，探索領域の絞込みにより局所解に収束してしまい性能が悪化するものと考えられる．

Griewank 関数では，最も良い結果となった DGA の移住間隔 5 世代のモデルと，最も悪い結果となった移住を行わないモデルの，ほぼ中間の結果となった．そこで，Griewank でも探索領域の絞込みを行わずに実験を行ったところ，Fig.16 に示すように性能が向上し 2 番目に良い結果となった．これは，Griewank も局所解が存在し探索領域を絞り込むことにより，局所解に収束してしまうことが考えられる．

4.4 考察

第 4.2 節より DGA/DSa は設計変数間に依存関係が無い場合や，一部の設計変数間に依存関係が有る場合

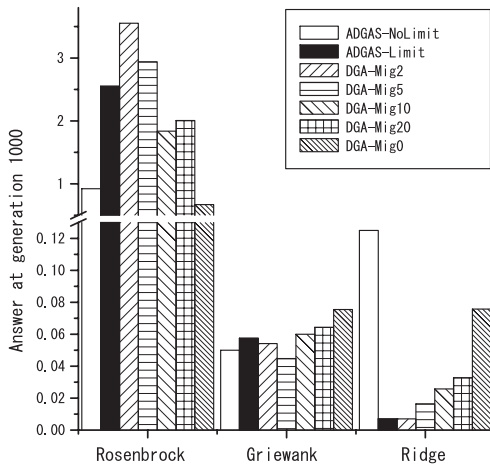


Fig. 16. The effects of changing feasible region.

は、DGA の最適なパラメータ設定と比較しても非常に優れた結果となり、有効な手法であることがわかった。

第 4.3 節より、提案する手法である ADGAS は依存関係の無い問題では大幅に性能が向上することが明らかとなった。また、一部に依存関係が有る問題でも、性能が向上する。これは、DGA の構造を見なおした DGA/DSa がこれらの問題に対し非常に優れているため、対象問題を評価し DGA/DSa が適応できる時には DGA/DSa を適応するためである。依存関係が全体的に有る問題では、DGA では Ridge 関数で最適な解を得るパラメータは Rosenbrock 関数ではもっとも悪いパラメータとなるなど、最適なパラメータ設定は困難であるといえる。ADGAS では、問題により DGA の最適なパラメータには及ばない場合も有ったが、DGA の最も悪いパラメータよりは良い結果となった。加えて、DGA の最適なパラメータと同程度の性能を得ることができる場合も有った。

したがって、ADGAS は依存関係が存在しない問題において、DGA と比較して非常に優れた探索能力を持つ。また、依存関係が存在する問題においても、DGA で探索を行う場合には予備実験を行い、対象問題に適したパラメータを選ばないと探索に大きな影響を与えるが、ADGAS では予備実験を行わずに探索を行ってもある程度の品質の解が得られる。これらのことから、本手法は有効な手段であるといえる。

5. 結論

従来の遺伝的アルゴリズム (Genetic Algorithms: GA) や分散遺伝的アルゴリズム (Distributed GA

:DGA) の問題点の 1 つは、計算コストの高さである。そこで、設計変数の依存関係に着目し探索空間を狭める事が可能な適応的分散遺伝的アルゴリズムシステム (Adaptive DGA System: ADGAS) を提案した。提案システムは依存関係の評価を行い、その結果に応じて GA のアルゴリズムを適応的に選択するシステムである。ADGAS は、設計変数間に依存関係が無いと評価された問題や、一部に依存関係が有ると評価された問題では依存関係の無い設計変数を各島で独立に探索する探索領域分割型 GA (Divided Searching Area GA: DGA/DSa) を適用することで非常に優れた解探索を可能とした。また、開始時の評価で依存関係が有るとされた問題でも、設計変数の探索領域を可変にすることで一部の問題で DGA/DSa の適用を可能とし、解探索能力が向上した。設計変数間に依存関係が有る問題においても、問題によって得られる解の品質がばらつくものの、おおむね DGA と同等以上の解の品質を得ることができた。したがって、本研究で提案する手法は、DGA に比べて優れたモデルであるといえる。

本研究は文部省からの補助を受けた同志社大学の学術フロンティア研究プロジェクト「知能情報科学とその応用」における研究の一環として行った。ここに記して謝意を表す。

参考文献

- 1) D.E.Goldberg. Genetic algorithms in search optimization and machine learnig. 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. 434-439, 1989.
- 4) M.Munemoto, Y.Takai, and Y.Sato. An efficient migration scheme for subpopulation-based asynchronously parallel genetic algorithms. *in Proceedings of Fifth International Conference on Genetic Algorithms*, p. 649, 1993.
- 5) Marin F.J., Trelles-Salazar O., and Sndoval F. Genetic Algorithms on LAN-Message Passing Architectures using PVM: Application to the routing problem. *Parallel Problem Solving from Nature, PPSN III*, pp. 534-543, 1994.
- 6) 三木光範, 廣安知之, 中村康範. 遺伝的アルゴリズムの分散並列化に関する研究. 日本機械学会日本機械学会論文集 (A 編), Vol. 66, No. 645, pp. 972-977, 1989.
- 7) 金子美華. 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.
- 8) 坂和正敏, 田中雅博. 遺伝的アルゴリズム. 朝倉書店, 1995.

出典：

同志社大学理工学研究報告 Vol.42, No.1

pp. 28-38

(2001年04月)

問い合わせ先：

同志社大学工学部/ 同志社大学大学院工学研究科

知的システムデザイン研究室

(<http://mikilab.doshisha.ac.jp>)