

ビット空間におけるGAの解探索モニタリングシステム

同志社大院 赤塚 浩太 同志社大工 廣安 知之 同志社大工 三木 光範

Monitoring System of Searching History of GA in a Bit Space

Kouta AKATSUKA, Graduate School of Engineering, Doshisha University
Tomoyuki HIROYASU, Faculty of Engineering, Doshisha University
Mitsunori MIKI, Faculty of Engineering, Doshisha University

Abstract: Usually in Genetic Algorithms (GAs), design variables of candidate solutions are encoded for continuous optimum solutions. Therefore, the searching space of GAs is different from the real number space. This condition causes that it is very difficult to figure out how GAs search optimum solutions. In this paper, the monitoring system of searching state by GAs is proposed. In the proposed system, the searching process is monitored using not only real number of design variables but also hamming distance, topology and building block entropy of individuals. By this system, we can recognize the efficiency of GAs, such as the characteristics and difficulty of problems. Through the numerical examples, the availability of the proposed system is examined and discussed.

1 はじめに

遺伝的アルゴリズム (Genetic Algorithms: GA) は生物の進化を工学的に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、一般に、GAでは連続関数最適化問題において対象問題の設計変数値をコーディングして探索を行うため、我々が把握している対象問題の解空間とはまったく異なった空間を探索していると考えられ、現在の対象問題がGAにより解探索が容易か否かや、各遺伝的操作がどのような効果をもたらすかを把握することは非常に難しい。そのため、これまでも対象問題のランドスケープを把握する手法や対象問題を分類する手法に関して研究が行われてきた²⁾。しかし、探索途中の個体群の様子をリアルタイムで把握することはこれまで困難であった。

そこで、本研究では連続関数最適化問題におけるGAの探索の様子をリアルタイムで把握できるモニタリングシステムを構築した。提案システムでは、設計変数値を用いた個体の分布の他に、コーディング後のビット列のハミング距離や位相などいくつかの特徴的な手法を用いて個体の分布を調査する。これらの手法を用いることで、設計変数値を用いた個体分布では得られない探索中の個体集団の特徴を、より、GAの探索に近い形で得ることができる。これにより、提案システムでは対象問題や遺伝的操作によって個体群の分布にどのような影響があるかを把握することができる。また、提案システムでは未知

の対象問題がGAにより解探索が容易か否かを探索中の個体群の様子からある程度把握することができる。本報告では、提案システムの紹介をした後、数値実験によって対象問題の性質や遺伝的操作の影響を提案システムを用いて調査し、提案システムの有効性を検討する。

2 解探索モニタリングシステム

2.1 目的

前述した通り提案システムでは、GAが解探索を行う様子をモニタリングすることにより、遺伝的操作が個体集団に与える影響や、対象問題が探索過程の個体集団に与える影響などを視覚的に捕らえることを目的とする。連続関数最適化問題において、GAは通常対象問題の設計変数値をコーディングして探索に用いるため、通常的设计変数値による個体集団の視覚化では探索の様子を正確に把握する事が困難である。そこで、本システムではコーディング後の個体集団を、より、GAの探索に近い形で視覚化するために後述するいくつかの手法を用いている。

2.2 システムの構成

提案システムの概要を Fig. 1 に示す。ユーザーはGUIを通じてGAのパラメータを設定する。図においてGA部ではGA操作を行い、その探索中の個体情報をGUI部に送り設計変数値情報として表示するか、Analyze部に送りハミング距離や位相の計算などを行った後、GUI部に情報を送り表示する。3Dまたは2Dの表示部は探索過程を把握するための様々な表示手法を持つ。

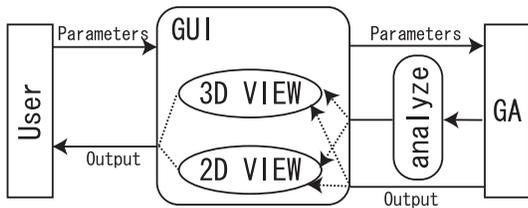


Fig. 1 Outline of Proposed System

本システムの特徴は以下の通りである。

- 設計変数値ではなく、関数評価値、位相、真の解からのハミング距離を用いて個体の分布を把握できる。これにより、連続関数最適化問題において、コーディングされているため設計変数値だけでは把握が難しいGAの探索の様子を、より、GAの探索に近い状態で把握できる。また、3設計変数以上の関数でも全設計変数を考慮に入れた情報が把握できる。また、離散的な関数の最適化問題においても同様である。
- 個体を関数評価値、位相などで表示するモードにおいて、個体群の傾向をよりわかりやすく把握するため、全個体の共分散行列による楕円(体)を用いた個体の分布を出力できる。
- 母集団が複数の場合(分散GA)に個体群を色分けして表示する機能を持つ。また、楕円体を表示するモードでは複数母集団毎に楕円体を表示する。このため、各分割母集団の傾向が容易に理解できる。
- GA部はGUI部とほぼ独立しているため、GAの改良などはGUI部を意識せずに行うことが可能である。
- 対象問題に外部のSolverを用いることが可能である。具体的にはGAの評価や初期化において、対象問題を内蔵する外部のアプリケーションとSocket通信によって結合する。そのため、自由に対象問題を設定することが可能である。
- 提案システムはJavaのAppletとして作成されているため、Java2とJava3Dのランタイムさえあれば多くのブラウザ環境で実行が可能である。

本システムが持つ機能の一覧を Fig. 2 にまとめた。GUI 部には設定可能な各種パラメータや探索開始と終了のボタンがある。また、表示部における 2, 3Dim は設計変数値による表示モード、FHTE, HT は位相やハミング距離による表示モード、History は履歴モードであり、FHTE と HT モードのみ全個体を点で表示するモードの他に個体分布を表す楕円(体)を表示することが可能である。

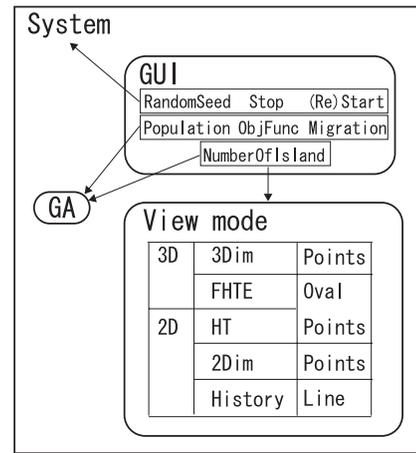


Fig. 2 Functions of Proposed System

2.3 表示方法

2.3.1 ハミング距離と位相, BuildingBlockのエントロピー

提案システムである解探索モニタリングシステムでは、コーディングされた個体群の分布を把握するために、ハミング距離と位相, BuildingBlockのエントロピーの3種の情報を活用する。

位相(Topology)は、染色体の遺伝子を並び順に同じ遺伝子をグループに分け、合計いくつのグループからなるかを計算し求める(Fig. 3)。

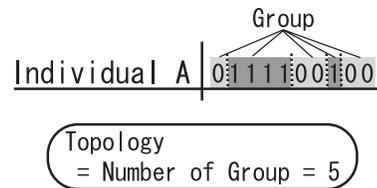


Fig. 3 Topology

ハミング距離は、2つの個体間の遺伝子がどの程度異なっているかを調べるもので、2つの個体の遺伝子を遺伝子座ごとに比較し、遺伝子の異なっている遺伝子座の数によって求める(Fig. 4)。

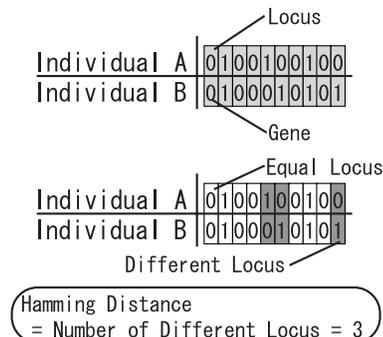


Fig. 4 Hamming Distance

BuildingBlockのエントロピーは、1個体の染色体中のBuildingBlockのパターンを調べ、同じ遺伝子が連続するブロックごとに分割し、各ブロックを事象とする。そして、遺伝子と連続数を元にした各事象の発生確率を元にエントロピーを計算する。Fig. 5の左の例では、遺伝子ブロック1が1つ、11が3つ...となり、各遺伝子ブロックについて全ブロック中の発生確率を求め、その確率を元にエントロピーを求める。

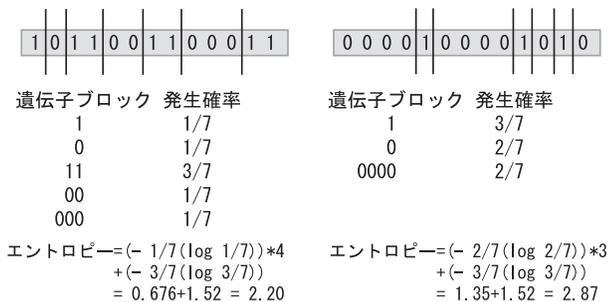


Fig. 5 Entropy of BuildingBlock

ここで、位相と BuildingBlockのエントロピーの間にはある程度の相関が生まれる。位相が多い場合には、同じ遺伝子の連続するブロックが短くなり、ブロックの数は多くなる。このため、エントロピーを求める際に同じ事象が比較的多く発生することになり、その結果 BuildingBlockのエントロピーは低くなる。位相が少ない場合には逆にエントロピーは高くなる。

2.3.2 設計変数値による表示

設計変数値による個体の表示では、対象問題を探索中の各個体の設計変数値の内、2次元モードでは任意の2設計変数、3次元モードでは同3設計変数を抜き出して表示する。その他の設計変数値は考慮しないため、探索の様子の一部を把握する事になる。

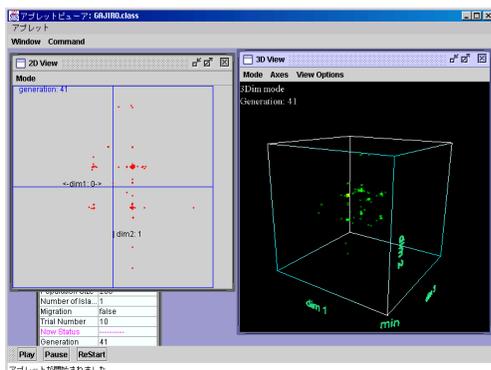


Fig. 6 Design Variable mode of Proposed System

このモードでは、個体分布のコーディングによる影響を全く考えていないため、対象問題のどの部分を探索しているかは把握できるが、GAの探索に関して有意な情報を把握することは難しい。

2.3.3 ハミング距離、位相、対象問題の評価値による表示

このモードでは、先に述べた関数評価値や位相、真の解からのハミング距離と BuildingBlockのエントロピーから任意の3軸を選んで、各個体をこれら3軸からなる3次元空間上にプロットし、その様子を把握することで探索の様子を把握する (Fig. 7)。

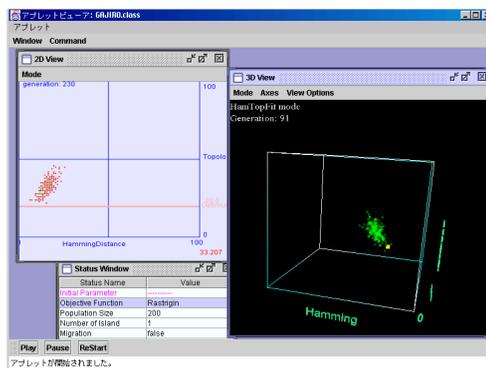


Fig. 7 Fitness, Topology, Hamming Distanse mode

分布の計算に設計変数値ではなく遺伝子を元にした情報を用いるため、設計変数値による表示に比べGAが探索を行っている様子を把握しやすいと考えられる。

2.3.4 共分散行列による全個体の楕円体表示

ハミング距離や位相を用いて探索過程を視覚化する場合、個体集団が密集し個体群全体としての傾向が把握しづらくなる。そこで、全個体の分散/共分散を求め、その行列の固有値/固有ベクトルを求めることで、個体群全体の傾向を示す楕円(体)を求め、個体の分布を楕円によって表すことで、個体群の傾向をよりつかみやすくする (Fig. 8)。

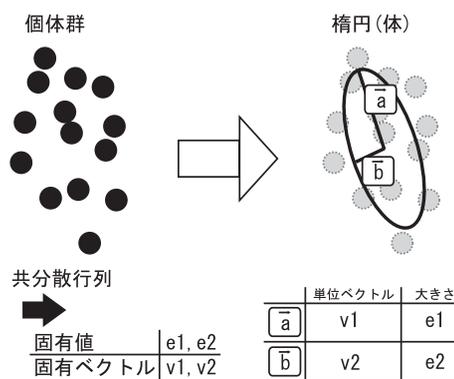


Fig. 8 Oval

2.4 設定可能パラメータ

現在システムが有する対象問題は、Rastirgin, Rosenbrock, Schewefel, Griewank, Ridge, Sphere, WeightedSphere, dejong 4, dejong 5, arkley, bohachevsky, rotatedRastirginの各関数である。また、先にも述べた

ように外部 Solver を用いることも可能である。その他 GUI から設定可能なパラメータとして、総個体数の他一般的な GA パラメータおよび島数、移住の有無などがあげられる。

3 数値実験

3.1 概要

提案システムの有効性を検証するために以下の 3 つの実験を行った。

- 対象問題の性質
GA の個体分布が、対象問題の難易度によって受ける影響を提案システムで把握可能か否かを検証した。
- 遺伝的操作の影響
GA の個体分布が、遺伝的操作によって受ける影響を提案システムにて検討した。
- 対象問題の分類
提案システムにより、対象問題が GA による解探索が容易か否かを判別可能か検討した。

対象問題はいずれも GA により容易に探索可能であるとされている Rastrigin 関数と、困難とされている Rosenbrock 関数をそれぞれ 10 設計変数で用いた。

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

$$-5.12 \leq x_i < 5.12$$

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

$$-2.048 \leq x_i < 2.048$$

3.2 対象問題の性質

3.2.1 概要

提案システムが、対象問題の個体に与える影響を把握するのに有効である事を検証するために実験を行った。GA のパラメータには、総個体数 400、交叉率 1.0、突然変異率 $1/L$ 、ルーレット選択、エリート保存戦略を用いた。

3.2.2 結果

Fig. 9 と Fig. 10 に、探索 200 世代目の設計変数値による個体の分布を示す。

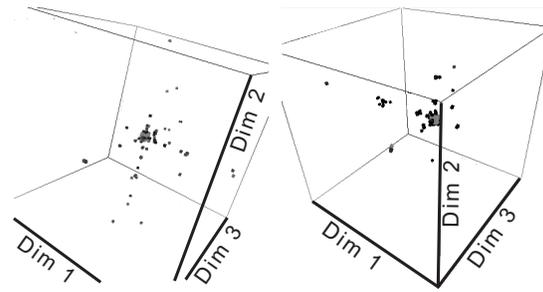


Fig. 9 Rastrigin Fig. 10 Rosenbrock

両関数とも、個体の分布がほぼ格子状になり、関数の性質による分布の差はほとんど無い事がわかる。次に、Fig. 11 と Fig. 12 に、探索 200 世代目のハミング距離や位相などによる個体の分布を示す。

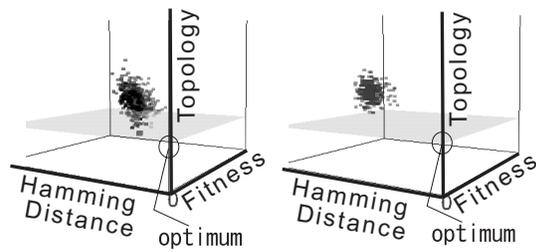


Fig. 11 Rastrigin Fig. 12 Rosenbrock

Rastrigin 関数においては、個体群は最適解に向かって斜めに分布し、最適解方向に探索領域が広がり、最適解付近が比較的多く探索されていることがわかる。一方 Rosenbrock 関数では、個体群は中心部に球状に存在し、探索領域が最適解から遠く、最適解付近がほとんど探索されていないことがわかる。

3.3 遺伝的操作の影響

3.3.1 概要

提案システムが、遺伝的操作の個体に与える影響を把握するのに有効である事を検証するために、分散 GA (Distributed GA: DGA)³⁾ における移住という遺伝的操作に着目し、実験を行った。DGA では、母集団を複数に分割し探索を行い、一定世代おきに移住という操作を行うことで、個体の情報交換を行う。この移住という操作を行う場合と行わない場合で、個体集団にどのような差が生まれるかを調べる。また、移住操作が与える影響が対象問題ごとに異なるかどうかを検証するため、先ほど用いた Rastrigin 関数と Rosenbrock 関数の両方で実験を行った。実験は、総個体数 200、島数 4 島、移住率 0.6、移住間隔 5 世代で行った。その他のパラメータは先の実験と同じである。

3.3.2 Rastrigin 関数における結果

Fig. 13 と Fig. 14 に、探索 100 世代目の設計変数値による個体の分布を示す。

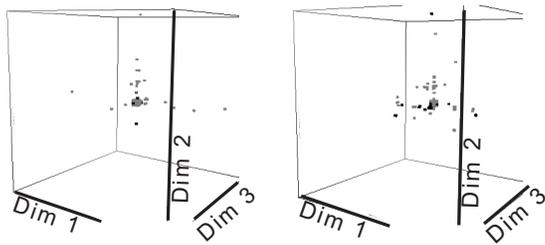


Fig. 13 Migration Fig. 14 Non-Migration

結果は、4島すべての個体を表示している。移住ありの場合でも、移住なしの場合でも個体はエリート個体付近に集まり、その分布に大きな差はない。Fig. 15とFig. 16に、同じ個体群の関数評価値、位相などによる個体の分布を示す。

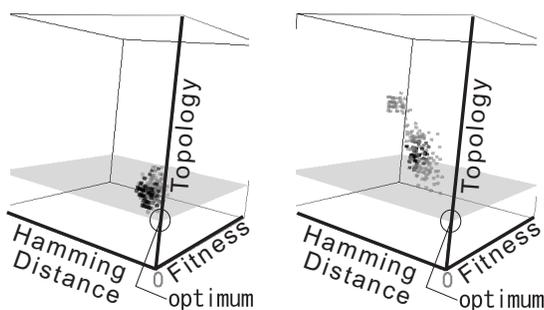


Fig. 15 Migration Fig. 16 Non-Migration

移住ありの場合、個体はエリート個体付近にほぼ集まっているのがわかる。一方移住なしの場合は、個体が比較的広範囲に分布していることがわかる。つまり、移住ありと移住なしでは個体分布に大きな差があることがわかる。

ここで、個体分布の各島による違いをより明確にするため、島ごとに全個体の分散を求め、個体の分布を示す共分散行列による楕円体を求めた。結果をFig. 17, Fig. 18に示す。

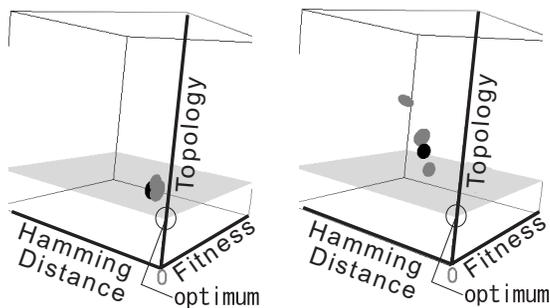


Fig. 17 Migration Fig. 18 Non-Migration

楕円体の様子から、より明らかに移住ありと移住なしの島毎の探索領域の違いがわかる。移住ありでは、すべての島がほとんど同じ領域を探索しており、島毎の探索領域の差はほとんど無く各島間で情報が交換されているこ

とがわかる。一方移住なしでは、各島はいずれも全く異なった領域を探索していることがわかり、各島間で情報の交換がなされていないことがわかる。また、移住なしと移住ありを比較した場合、移住ありの方が移住なしより真の解に近い領域を各島が探索していることがわかる。また、移住なしの場合には最も探索が進んでいる島以外の島は探索が進んでいる島の後を追う、あまり効果的な探索をしているとはいえない。これは、Rastrigin関数を解く場合に、移住を行った方がよい結果が得られるという経験的な傾向にも一致する。

3.3.3 Rosenbrock関数における結果

Rosenbrock関数における、位相、ハミング距離、関数評価値を元にした探索200世代目の個体分布を楕円体で示したものをFig. 19, Fig. 20に示す。

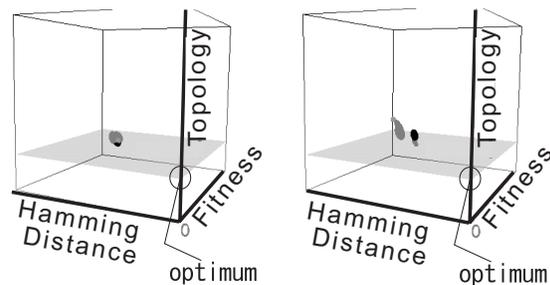


Fig. 19 Migration Fig. 20 Non-Migration

いずれの場合の傾向もRastrigin関数とほぼ同じである。移住ありの場合は、各島の分布はほとんど重なっており、各島間で情報交換が行われた結果、すべての島でほぼ同じ領域を探索していることがわかる。一方移住なしでは各島の分布は大きく異なり、情報交換が行われず各島が独立して探索していることがわかる。ただ、Rastrigin関数の場合と異なり、移住ありの方が探索が進んでいるという事は無い。また、移住なしの場合では、各島が異なった領域を探索し、より広範囲の探索を行っていると考えられる。Rosenbrock関数には、移住をあまり行わないほうが比較的良好な結果が得られるという経験的な傾向がある。これは実験結果の移住ありでは局所的な探索に陥り、移住なしではより広範囲な探索が行えるという結果と一致する。

3.4 対象問題の分類

3.4.1 概要

すでに提案システムを用いて、Rastrigin関数とRosenbrock関数の探索過程の差異は明らかになった。しかし、その際に用いた手法は最適解を必要とするため、テスト問題に対してGAの探索過程を把握するには有効であるが、未知の問題に適用して対象問題の分類を行うことは不可能であった。そこで、位相とBuildingBlockのエントロピー、関数評価値を用いて対象問題の分類が行える

かを検証した。対象問題，GA のパラメータなどは同じである。

3.4.2 実験結果

両関数の探索初期，中期，後期の個体分布を示す。なお，Rastrigin 関数は 350 世代ほどで解に到達するため 100,200,300 世代目を，Rosenbrock は 1000 世代経過しても解に到達しないため 100,400,800 世代目をそれぞれ初期，中期，後期とした。

探索初期 両関数とも個体が右斜め上向きに分布している。これは，Entropy と Topology にある程度の相関関係があるためで，ランダムに発生した個体はこのような分布となる (Fig. 21, Fig. 22)。

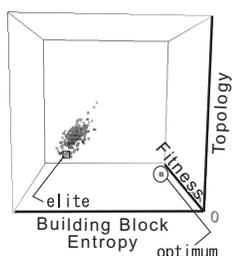


Fig. 21 Rastrigin

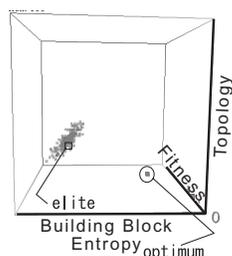


Fig. 22 Rosenbrock

探索中期 Rastrigin 関数では，エリート個体がランダムに形成されていた個体群から抜け出している。一方 Rosenbrock 関数では，エリート個体は依然ランダムに形成される個体群中にある (Fig. 23, Fig. 24)。

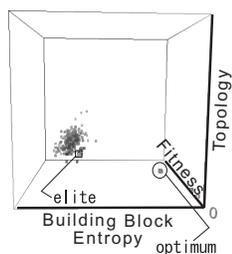


Fig. 23 Rastrigin

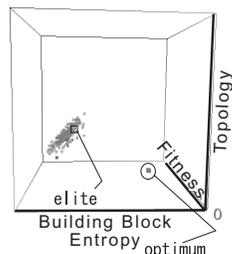


Fig. 24 Rosenbrock

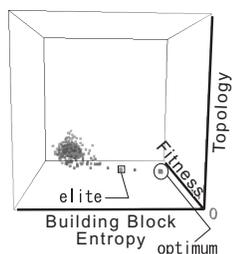


Fig. 25 Rastrigin

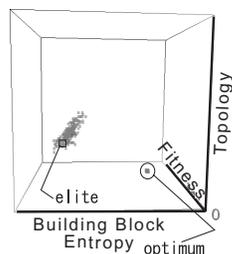


Fig. 26 Rosenbrock

探索後期 Rastrigin 関数では，抜け出したエリート個体の影響により個体群自体が最適解方向に広がっている。一方 Rosenbrock 関数では，探索初期からの進展は見られない (Fig. 25, Fig. 26)。

3.4.3 考察

GA による解探索が容易とされている Rastrigin 関数では，探索初期こそエリートは個体群の中にあるが，中期，後期と進むにつれて個体群から徐々に離れていることがわかる。これは，ランダムによって生み出される個体と，エリート個体の性質が徐々に異なっていく事が原因と考えられる。一方 GA による解探索が困難とされている Rosenbrock 関数では，エリート個体は終始個体群の中にあり，ランダムによって偶然生まれていることが考えられる。また，エリート個体の位置は更新されるたびに大きく変わるため，この個体群はほぼランダムサーチを行っているのではないかと考えられる。以上のように，GA が機能しているか，ランダムサーチに陥っているかを判別することにより，対象問題が GA による解探索が容易か否かを検証することができる。

4 結論

GA は一般に，連続関数最適化問題において，設計変数をコーディングして探索を行うため，その探索過程を把握することは容易ではない。そこで，GA の探索過程を把握するための解探索モニタリングシステムを提案した。本システムを用いることで，設計変数値と関数評価値だけではわからなかった関数による探索の差異や，遺伝的操作の探索に及ぼす影響の一部を把握することができた。

現在のシステムでは，対象問題の判別の際に用いる BuildingBlock のエントロピーが最適解のビット列に依存し，対象問題の判別が失敗する可能性があるため，今後最適解に依存しないシステムを構築する。

謝辞

本研究は文科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] Ken NAITOU. Four-group equation of genetic algorithm. JSME International Journal, Vol. 38, No. 2, 1995.
- [3] Reiko Tanese. Distributed genetic algorithms. Proc. 3rd International Conference on Genetic Algorithms, pp. 434-439, 1989.

出典：

第14回自律分散システム・シンポジウム論文集,

pp. 105-110

(2002年1月25日・26日・東京)

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>)