遺伝アルゴリズムによる最適化に関する考察

東京工業大学 総合理工
※ 木村元，小林重信，山村雅幸，中村清彦

Abstract

従来,連続関数の最適化においてもGAを適用する試みがあったが,連続値をわざわざ離散値に変換して組合せ最適化として解いたり,他の探索手法との比較を十分に行っていないなどの問題があった.

本研究では,連続関数の最適化に適したGAの個体の表現方法や交叉方法を提案する.

この方法を用いて実験を行い,GAの挙動を観察する.また,GAの性能を他の探索手法と比較し,性能の位置付けを行う.

1 序論

"最適化"という概念は極めて日常的かつ普遍的な概念であり,システム科学においても主要な概念として位置づけられてきた.近年,生物系や物理系における自然現象からのアナロジーに基づき,最適化に対する全く新しい方法論が出現するに至った.Hopfieldモデル,Simulated Annealing, Boltzmanマシン,遺伝アルゴリズムなどがそうである.脳や神経回路網(ニューロンネットワーク)などにおける生体系の適応的な情報処理方式の仕組みを,最適化問題における準最適解の求解に応用する研究が活発に行なわれていが,その中でも,遺伝アルゴリズム(Genetic Algorithm:以下GAと呼ぶ)は新しい接近法の一つである.概してニューロンネットワークが生物の短期的な適応,学習過程であるのに対して,GAは長期的なそれに対応するものである.遺伝学を模倣することにより生まれたGAであるが,今までにも様々な問題,例えば組合せ最適化問題をはじめとして,従来の数理的方法では計算量の壁ゆえに接近が困難とされていた問題に対しブレークスルーを与える強力な手法として期待されている[Goldberg 89] [De Jong 89] [Richardson 89] [小野 91].

本論文では制約領域が凸のn変数xの関数$f(x)$の最適点を偏導関数を使わずに見い出す問題を考える.

従来の代表的な直接探索手法としてシンプレックス法や,Simulated Annealingなどがある.

本研究では,目的関数が単峰性と多峰性の場合について,提案する方法の有効性を確認することを目的とする.比較のためにシンプレックス法とランダムサーチを取り上げる.

Integer	*Binary*	*Gray*	*Integer*	*Binary*	*Gray*
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

表1: バイナリコードとグレーコード

2 従来法の問題点

従来,GAによる連続関数の最適化において,個体を染色体に表現するための代表的なコーディング方法に,バイナリコーディングとグレーコーディングがある[Caruana 88]. [表1]

GAは一点交叉しか行なわないSimple GAのみ考える.

1次元の連続関数の最適化におけるGAでは,染色体の交叉において以下のような特徴がある.

第15回 知能システムシンポジウム(平成4年3月18日・19日・東京)

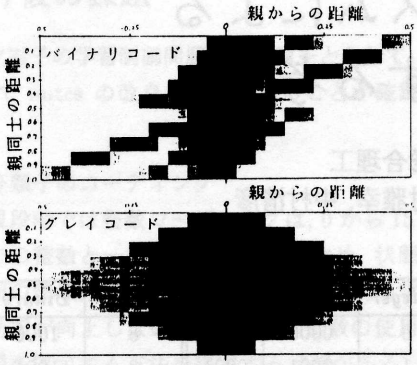

Fig. 1: バイナリコードとグレイコードの比較

例)

親 A 1101 0110 → 1101 1110

親 B 1011 1110 → 1011 0110

一点交差なので染色体は必ず上位部分と下位部分に分けて交換を行う．これは，バイナリコードでもグレーコードでも同様である．そこで上位部分のビットを受け継いだ子の個体をその親の個体の'直系の子'と定義する．同時に直系の子の直系の親を'直系の親'と定義する．

すると，染色体が交叉オペレータによって変化するのは下位ビットの部分だから，染色体を連続値に変換すると，直系の子はもう一つの子よりも数直線上で直系の親に近い場所に生成される．これを親の周辺と定義する．

本研究では直系の子が直系の親の周辺にどのような分布で生成されるかは，もう片方の親との数直線上の距離の影響を受けて変化するとの見方をする．

遺伝子は 0，1 の 2 値，染色体の長さ=8 ビットについてこの関係を Fig.1 に示す．

親から生成される子の分布の形をコーディングと考える．以後，親から生成される子の分布の形を交叉分布と定義する．

3 文字列に変換しない GA の提案

本章では従来の文字列による GA の問題点を考慮して，新たなデータ構造を持つ GA を提案する．

3.1 新たな GA の必要性

従来の GA では，交叉により次の世代の遺伝子へ文字の部分列として形質を保存して，組合せ最適化として問題を解決するという考え方である．GA を用いた連続関数の最適化においては，最適化する連続値のパラメータを離散表現である文字列に変換し，文字列の組合せ最適化に置き換えて最適化を行っていた．

連続関数の最適化において，個体の類似性は，個体のパラメータの相対的なユークリッド距離の大小で定義する．ところが文字列に変換して，組合せ最適化として個体の類似性をみると，連続値として見た場合似ている (近い) もの同士が，文字列では必ずしも近くないという場合が多い．

また，文字列で交叉を行うと，飽和現象が生じるという問題があり，これを取り除くための方法も数多く提案されているが，処理手順が複雑なものが多く，実装や解析を困難にしていた．この問題は，連続関数の最適化という視点で考えると，交叉分布が完全な確率分布ではないという原因によるものと言うことができる．

従来の連続関数の最適化手法 (例えばシンプレックス法) では，パラメータを連続値として扱い，計算機上に実装するときはやむを得ず浮動小数として扱っている．それに対して，従来の GA ではパラメータは固定長文字列により絶対的なコーディングを行う．そのため従来の連続関数の最適化手法と，GA を用いた連続関数の最適化手法とは，データ形式が違うことにより探索する空間の広さも異なり，直接性能の比較を行うことができないので，GA による方法の性能を位置付けることができなかった．

GA を用いた連続関数の最適化手法の性能を改善するには，次の世代の個体への形質を保存しつつ最適化を行うという GA の本質的な特徴を損なうことがないように，連続関数特有の形質 (類似性) を考慮したデータ形式にするべきである．

3.2 文字列に変換しない GA とは

3.2.1 データ構造

連続関数における個体の形質 (類似性) は，個体間のパラメータの相対的なユークリッド距離で定義する．従来，パラメータを文字列に変換していた部分を，文字列に変換しないでそのまま連続値として扱うのが，

文字列に変換しないGAである．これにより，従来の連続関数の最適化手法との比較ができるようになる．

3.2.2 遺伝的操作

個体を文字列に変換しないので，従来のGAで行っていたような，選択(selection)を除いた遺伝的操作(交叉，突然変異)はこのまま用いることはできない．そこで，直系の親の周辺に，ある確率分布(交叉分布)で直系の子を生成することを交叉オペレータと定義する．この交叉分布は，もう片方の親のパラメータとの差の関数として定義する．

従来の文字列によるGAでは，バイナリやグレーコードのの部分で示したように，この交叉分布を変えることは不可能だが，上記のようにデータ構造と交叉オペレータを定義することにより，分布を自由に変えることができるようになる．交叉分布は，文字列のGAでは親同士の距離の関数で定義したが，文字列に変換しないGAでは親同士の距離だけでなく，世代数や集団の分散の関数などでも定義できる．非常に自由度が大きいという特徴がある．例えば，交叉分布をただの一様分布にすると，ランダムサーチと等価になる．

3.2.3 処理手順

GAの枠組そのものは従来のものと変わらない．

4 文字列に変換しないGAの性質

本章では，前章で提案した方法について実験を行い，その有効性について確認する．

4.1 実験の目的

前章で提案した，文字列に変換しないGAについて，以下のような疑問点を明らかにする．

1. GAの個体集団が実際にどのような挙動を示すのか．
2. 適応度関数の違いでどのように性能が変化するのか．
3. 他の最適化手法と比較してどのような特徴があるのか．

4.2 実験方法

GAでは個体数20で最も適応度関数値が高い個体だけ次の世代に残すエリート戦略とした．条件を同じにするため，シンプレックス法においても端点を20個とった．

[実験1]

2次元の単峰性連続関数の最適化を，ランダムサーチ，シンプレックス法，文字列に変換しないGAの3通りの方法で行い，性能を比較する．

また，このときのGAの個体集団の変化を観察する．

[実験2]

2次元の多峰性連続関数の最適化を，ランダムサーチ，シンプレックス法，文字列に変換しないGAの3通りの方法で行い，性能を比較する．

山の多さを変化させた場合，どのように性能が変化するのか，また個体集団はどのような挙動を示すのか観察する．

4.3 実験結果

Fig.5.4.1にGAで用いた交叉分布を示す．

[実験1]〈単峰性関数の場合：Fig.2参照〉

$$f = 400 - |10x_1 - 5{x_2}^2| - |5x_2 - 10| - |5x_1x_2 - 20|$$

Fig.3は個体集団の世代変化の様子である．Fig.4にランダムサーチ，シンプレックス法，文字列に変換しないGAの3通りの方法による性能の比較を示す．

[実験2]〈多峰性関数の場合：Fig.5参照〉

$$f = 300\cos\left(\alpha\sqrt{3({x_1}^2 + {x_2}^2)}\right) e^{\left(-0.2\sqrt{3({x_1}^2+{x_2}^2)}\right)} + 300$$

定数$\alpha = 1, 3, 6, 12, 24$

全ての例題において，中心の大局解(global minimum)の周りに局所解(local minimum)が同心円状に分布している関数に統一した．問題の難易度は周期関数の周期を調節して局所解の数を変えることで対応した．

Fig.6に2次元関数の場合のランダムサーチ，シンプレックス法，文字列に変換しないGAの3通りの方法による性能の比較を示す．Fig.7に，Fig.6.(4)の場合のGAの個体集団の挙動の様子を示す．

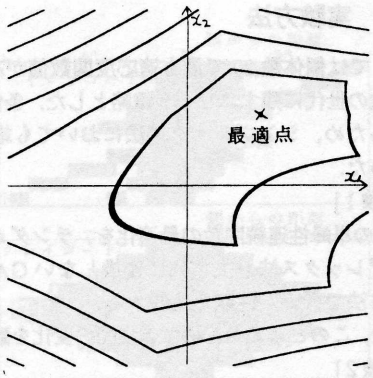

Fig. 2: 単峰性関数の概形

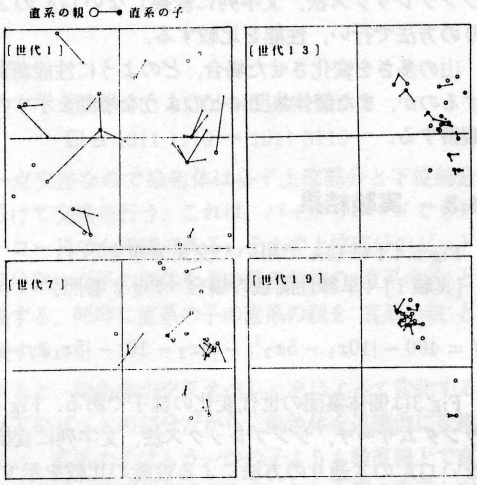

Fig. 3: 単峰性関数での世代変化

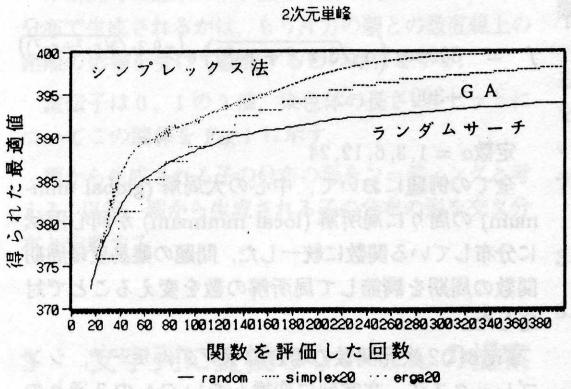

Fig. 4: 性能比較

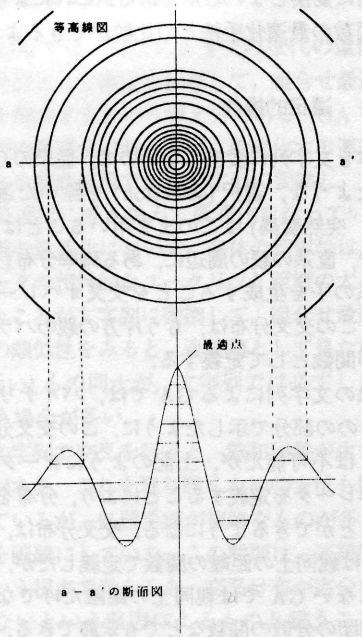

Fig. 5: 多峰性関数の概形の一例

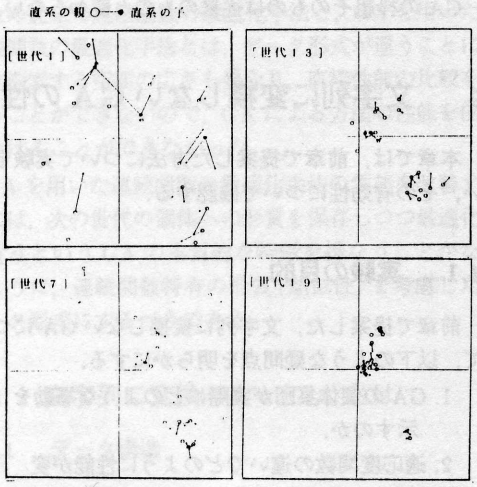

Fig. 6: Fig.6.(4) の場合の個体の挙動

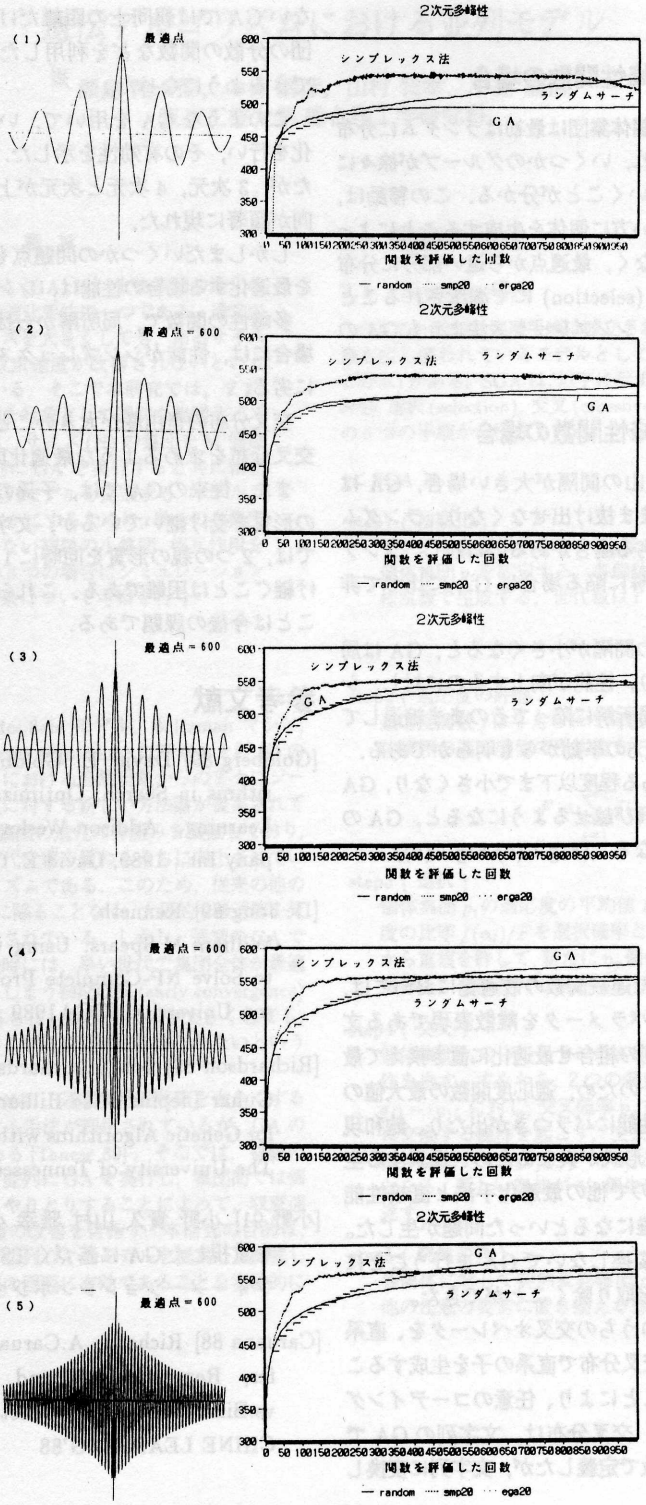

Fig. 7: 多峰性関数での性能比較

5 考察

5.1 実験1:単峰性関数の場合

Fig.3より，GAの個体集団は最初はランダムに分布していたのが淘汰され，いくつかのグループが徐々に最適点へと移動していくことが分かる．この移動は，意識的に最適点に近い方に個体を生成することによって生じているのではなく，最適点から遠い部分に分布している個体が選択 (selection) にて淘汰されることで生じる．そのため，シンプレックス法よりもGAの収束が遅い．

5.2 実験2:多峰性関数の場合

Fig.7より，関数の山の間隔が大きい場合，GAは局所解に陥り，そのまま抜け出せなくなり，ランダムサーチよりも性能が劣る場合があるのに対し，シンプレックス法では局所解に陥る場合はGAに比べて非常に少ない．

しかし，関数の山の間隔が小さくなると，GAは局所解に陥りにくくなり，性能が向上するのに対し，シンプレックス法では局所解に陥ってそのまま縮退してしまう．これは，Fig.6の挙動からも明らかである．

関数の山の間隔がある程度以下まで小さくなり，GAの個体が簡単に山を飛び越せるようになると，GAの性能向上は頭打ちになる．(Fig.7.(4),Fig.7.(5))

6 結論

従来のGAを用いた連続関数の最適化においては，最適化する連続値のパラメータを離散表現である文字列に変換し，文字列の組合せ最適化に置き換えて最適化を行っていた．そのため，適応度関数の最大値の位置によってGAの性能にバラつきが出たり、飽和現象が生じたり，データ形式が異なるため、探索する空間の大きさが異なるので他の最適化手法と直接性能を比較することが困難になるといった問題が生じた。

そこで、文字列に変換しないでGAを行うことにより、上記の問題点を取り除くことができた。

さらに遺伝的操作のうちの交叉オペレータを、直系の親の周辺に、ある交叉分布で直系の子を生成することと新たに定義することにより、任意のコーデイングを行うことができた。交叉分布は，文字列のGAでは親同士の距離の関数で定義したが，文字列に変換しないGAでは親同士の距離だけでなく，世代数や集団の分散の関数などを利用した大きな自由度で定義できるようになった。

このようなGAを用いて，いくつかの関数の最適化を行い，その有効性を示した．ここには挙げなかったが，3次元，4次元と次元が上がるとさらにこの傾向が顕著に現れた．

しかしまだいくつかの問題点もある．単峰性の関数を最適化する場合の性能は，シンプレックス法に劣る．

多峰性の関数で，局所解から抜け出しにくいような場合には，性能がシンプレックス法やランダムサーチに劣る．

交叉分布の自由度が大きいことにより，逆に最適な交叉分布を求めるような最適化問題を生じさせた．

また，従来のGAでは，子孫の個体は2つの親同士の形質を受け継いでいるが，文字列に変換しないGAでは，2つの親の形質を同時に1つの子孫の個体に受け継ぐことは困難である．これらの問題点を解決することは今後の課題である．

参考文献

[Goldberg 89] David E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company Inc., 1989, David E. Goldberg(editor).

[De Jong 89] Kenneth A.De Jong William M.Spears, Using Genetic Algorithms to Solve NP-Complete Probrems, George Mason University, ICGA1989

[Richardson 89] Jon T.Richardson Mark R.Palmer Gunar Liepins Mike Hilliard, Some Guidelines for Genetic Algorithms with Pentaly Functions, The University of Tennessee, ICGA1989

[小野 91] 小野 貴久 山村 雅幸 小林 重信, 形質遺伝を重視したGAに基づくTSPの解法, 知能のリフォーメーションシンポジウム,1991

[Caruana 88] Richard A.Caruana J.David Shaffer, Representation and Hidden Bias:Gray vs.Binary Coding for Genetic Algorithms, MACHINE LEARNING'88