

(28) 形質遺伝を考慮した 遺伝的アルゴリズムの1方法

A METHOD OF GENETIC ALGORITHMS CONSIDERING CHARACTER INHERITANCE

小林 重信[※]

Shigenobu KOBAYASHI

The most important components in genetic algorithms are gene representation and crossover. The both are mutually dependent. A number of existing works on genetic algorithms have put emphasis on suppression of fatal genes. We consider that good characters of parents should be appropriately inherited to children. To realize character inheritance, we present more rational methods about gene representation and crossover. Moreover, a method of code reduction, that make genetic algorithms possible to apply to large-scale problems, is presented. The effectiveness of proposed methods is shown through their application to the travelling salesman problem.

Key Words : Genetic Algorithms, Gene Representation, Crossover,
Code Reduction, Travelling Salesman Problem

GAにおいてもっとも重要な要素は、コード化(遺伝子の表現)と交叉であり、両者は表裏一体の関係にある。巡回セールスマン問題(TSP)を対象とした従来の研究の多くは致死遺伝子の抑制を重視している。これに対し、本研究では両親の形質を子に適切に継承させることが重要との立場をとり、コード化および交叉について、より合理的な方法を提案する。さらに、大規模問題への適用を可能とするために、コードを縮約する方法(ブロック化)を提案する。TSPへの適用を通じて、提案した方法の有用性を示す。

1 はじめに

Genetic Algorithm (以下、GA という) は生物における進化のメカニズムを模倣した探索アルゴリズムの1種であり、組み合わせ最適化問題のように、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、近似解を効率よく求めることが期待できる手法として、最近、多くの研究者の関心を集めている。GAの一般的な手順は確立されているわけではないが、本研究ではつぎのような手順を前提として議論を進めることにする。

[※] 工博 東京工業大学教授 大学院総合理工学研究科知能科学専攻

227 横浜市長津田4259, TEL 045-922-1111 ex. 2648

step 1 (初期化) ランダムな遺伝子をもつ m 個の個体 C_i からなる初期生物集団 P_0 を設定する。 $t=0$ とする。

$$P_t = \{C_1, C_2, \dots, C_m\}$$

step 2 (交叉による繁殖) P_t の中でランダムに対 (pair) をつくり、交叉によって両親の遺伝子を部分的に継承した新しい個体を k 個生成し、 P_t に加える。

$$P_t = \{C_1, \dots, C_m, \dots, C_{m+1}, \dots, C_{m+k}\}$$

step 3 (突然変異) P_t の各個体に対し、ある生起確率で、遺伝子の一部を他の任意の要素に置き換える操作を施すことにより、突然変異を起こさせる。

step 4 (淘汰) P_t の各個体 C_i の適応度 $f(C_i)$ を計算する。集団としての平均適応度 F を考慮に入れて各個体の選択確率を決定し、これに基づいて m 個の個体を選択し、次世代の生物集団 P_{t+1} とする。

$t = t + 1$ として、step2 に戻る。

GA の適用に際して、まず考慮すべきことは、遺伝子の表現方法、すなわちコード化 (coding) の問題である。コード化はつぎの条件を満足すべきである。

- 冗長性の排除
1 つの解と 1 つの遺伝子が 1 対 1 対応となること。
- 致死遺伝子の抑制
交叉による致死遺伝子の生成が抑制されること。
- 形質の遺伝
交叉によって生成される子は両親の形質を適切に継承できること。

遺伝オペレータの中で、交叉はもっとも重要なオペレータである。交叉オペレータとして、例えば、つぎのようなオペレータが提案されている。

- one point crossover
染色体上の予め指定された 1 箇所でのみ交叉を許す。
- multi-points crossover
染色体上の予め指定された複数箇所での交叉を行わせる。

- segmented crossover
染色体上の任意の複数箇所での交叉が許される。

- blended crossover
遺伝子が連続的な値を取る場合、両親の値の中間の値を子に継承させる。

コード化と交叉は表裏一体の関係にあり、両者は共に問題依存性である。従って、対象問題の性質に即してコード化と交叉は適切に定義される必要がある。

2 TSP 問題への接近

2.1 問題の所在

TSP を対象として、コード化と交叉の問題を考えよう。TSP の解である 1 つのツアーにおける都市の訪問順序をそのまま文字列としてコード化し、文字列の入れ替えによる交叉を行うことを考える。つぎの 2 つの親の遺伝子の 4 番目以降を入れ替えることによって生成される子は、明らかに TSP の制約を満たさないいわゆる“致死遺伝子”をもつ。

$$\begin{array}{ll} \text{親: } (a, b, c, \boxed{d, e}) & (a, b, e, \boxed{c, b}) \\ \text{子: } (a, b, e, \boxed{c, b}) & (a, b, e, \boxed{d, e}) \end{array}$$

このように致死遺伝子が生まれる原因は、「1 つの都市は 1 度しか訪問しない」という制約がコード化および交叉方法にうまく表現されていないことにある。[Grefenstette 85] は、文字列の入れ替えによる交叉によって致死遺伝子が生まれないようなコード化方法として、“ordinal representation”を提案している。しかし、この方法では、交叉点より後の方では、全く親の形質が子に受け継がれないため、交叉点が遺伝子の前にあるほど、子のツアーは親とは似ても似つかぬものになってしまう欠点をもつ。[Goldberg 85] は、これとは反対に遺伝子の表現にはツアーをそのまま用い、交叉においては任意のサブツアーを交換した後、致死遺伝子を生じないようにその他の部分を調整する交叉方法として、“partially mapped crossover”を提案している。しかし、これも交換されるサブツアー以外は親の形質が子に受け継がれない欠点をもつ。これらの方法は、致死遺伝子の生成を抑制することだけを考慮しており、親の形質の遺伝に対する考慮が不十分なのが指摘される。致死遺伝子を生成することなく、親の形質をできるだけ継承させることができるコード化および交叉方法が必要である。

2.2 コード化の方法

本研究では、都市の訪問順序を自然にコード化するものとする。例えば6都市問題の1つの順序 $a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow f \rightarrow a$ をつぎのように表現する。

右回り表現 (a, b, c, d, e, f, a)

左回り表現 (a, f, e, d, c, b, a)

ここでは、距離行列は対称であることを想定しているが、対称でない場合でも、右回り表現と左回り表現の両方を採用するものとする。両方向表現を採用することにより、2つの親A,Bの間では、

- 1) 親A(右回り)と親B(右回り)での交叉
- 2) 親A(右回り)と親B(左回り)での交叉

の2通りを考えることにより、4つの異なった子が生成される。なお、

- 3) 親A(左回り)と親B(右回り)での交叉
- 4) 親A(左回り)と親B(左回り)での交叉

によって生成される子は1と2で生成されるものと同じである。

2.3 交叉の方法

本研究では、交叉方法として、2点交叉 (two points crossover) を採用し、かつ致死遺伝子の生成を抑制するために、交叉のための切断箇所間に存在する都市集合が一致するときのみ交叉を行わせるものとする。例えば、2つの親A,Bがつぎのように与えられたものとする。

親A(右回り): (a, b, c, d, e, f, a)

親B(右回り): (a, e, c, b, d, f, a)

親Aに対しては1番目と4番目を、親Bに対しては2番目と5番目を、それぞれ切断箇所として選択するものとする。

親A(右回り): $(a|b, c, d|e, f, a)$

親B(右回り): $(a, e|c, b, d|f, a)$

切断箇所間に存在する都市集合は、いずれも $\{b, c, d\}$ となることから、交叉によって生成される子は必ず実行可能なツアーを形成することになる。この例の場合、つぎの4通りの実行可能な子が生成される。

- 1) 親A(右回り表現) : $(a|b, c, d|e, f, a)$
- 親B(右回り表現) : $(a, e|c, b, d|f, a)$

- | | | |
|--------------|---|-------------------------|
| 子1 | : | (a, c, b, d, e, f, a) |
| 子2 | : | (a, e, b, c, d, f, a) |
| 2) 親A(右回り表現) | : | $(a b, c, d e, f, a)$ |
| 親B(左回り表現) | : | $(a, f d, b, c e, a)$ |
| 子3 | : | (a, d, b, c, e, f, a) |
| 子4 | : | (a, f, b, c, d, e, a) |

3 コードの縮約

本研究では、個体集団中で同一の遺伝子をもつ個体は1個として取り扱う。

本論文で提案する遺伝子のコード化および交叉方法によってTSPを解く場合、世代が進み解が収束しつつある時には、populationの中の遺伝子は、同じような順序のsubtourによって構成されるようになる。こうしたsubtourにおいては、交叉が可能である場合が多い反面、同一の遺伝子を持つ子孫を発生しやすいという特徴がある。先に述べたように、本研究では同一の遺伝子をもつ個体は1個として取り扱うので、このような無駄を省くことが望ましい。そこで、次に示す方法で親の遺伝子を縮約して同一の遺伝子を持つ子孫が発生しないような工夫を行った。

[Subtour compress coding]

1. 2つの親に共通するsubtourを見つける。
2. そのsubtourを1つの都市として表現を変更する。
3. 縮約されたコードにより交叉を行い、子孫を作る。
4. 子孫を逆コード化して、元のtourの形に戻す。

例えば、次の2つの親を考える。

$(a, j, h, i, f, d, b, c, e, g, a)$

$(a, e, g, f, i, h, d, b, c, j, a)$

下線部は同一であるので、ここにまたがるsubtour、例えば (f, i, h) 、 (f, i, h, d) 、 (f, i, h, d, b) 等に基づく交叉はすべて同一の子孫を生じる。これらの親は次のように交叉によって同一の子孫を生じないように変更される。

$(a, j, h, i, f, [d, b, c], [e, g], a)$

$(a, [e, g], f, i, h, [d, b, c], j, a)$

ここで、都市数の縮退に伴ってsubtourの探索にともなう計算量も大幅に削減されることに注意されたい。

4 アルゴリズム

初期値として、第0世代10遺伝子をランダムに生成する。N-1世代の全ての遺伝子の対(pair)に対して、一定の確率(交叉確率)で交叉を行うか行わないかを決め、一定の回数(交叉回数)だけ交叉を行い、子孫を生む。繁殖した子孫を、全行程距離によって評価した後、ある確率分布(淘汰曲線)にしたがってそれを残すかどうかを選択する。生き残った子孫がN世代となる。

N-1世代から子孫を増殖し、N世代を生成するまでの過程を次に詳しく示す。

1. N-1世代の遺伝子で可能な親の組合せ(pair)をすべて作り、各pairに対して以下の処理を行う。
 - (a) 一定の交叉確率で交叉させるかどうかを決定する。
 - (b) subtour compress coding で遺伝子を縮約する。
 - (c) 一定の交叉回数(N) 回だけ交叉を適用する。
 - (d) 評価値を計算する。
2. 生成された子孫の遺伝子集団について、重複する遺伝子を削除し、一定の淘汰曲線 $F(x)$ にしたがって淘汰し、N世代とする。

5 実験

5.1 実験の目的

TSPにおける解の探索空間は、一般に複雑な形状をしており、ニューラルネットワーク等の山登りのアプローチでは、解の評価値よりもむしろ解の探索空間における誘引領域の広さにより、収束する解が決定される。本実験では、個体の表現形(TSPの場合順路)が異なる複数の局所解を持つ問題として、2重円に都市が配置されているものをとり上げ、GAによる解法がどのような挙動を示すかを求め、本研究で示した遺伝システムの適正を考察するとともに、GAの組合せ最適化問題に対する性能について考察する。

5.2 実験例題

実験に用いた都市配置およびGAの各パラメータを以下に示す。

●都市データ形式

都市数 : 48

都市配置: 2重円

内側都市数 24、外側都市数 24

内外半径比率 (外側円の半径 0.5 に対する内側円の半径)

(1) 0.2	(7) 0.3845
(2) 0.25	(8) 0.392
(3) 0.3	(9) 0.4
(4) 0.325	(10) 0.42
(5) 0.35	(11) 0.45
(6) 0.37	(12) 0.48

この問題は、局所解として、図1に示すように2つの解を持つ。それぞれのツアーの長さ(評価値)は都市数 n と内側の円と外側の円の半径からそれぞれ次のように求められる。

●歯車型:

$$\text{tour} = (0.5 + x) \sin \frac{2\pi}{n} + n(0.5 - x)$$

●C型:

$$\text{tour} = 2\{(0.5 + x)(\frac{n}{2} - 1) \sin \frac{2\pi}{n} + (0.5 - x)\}$$

これより、最適解は、外側の円と内側の円の半径比が $0.5:0.38454\dots$ の場合(7)よりも小さい場合はc型に、大きい場合はo型になる。

●GAパラメータ設定

交叉確率	0.4
交叉回数	10
世代数	100

5.3 実験結果

各収束型を以下のように略記する。

- c : ドーナツ型
- c' : ドーナツ型で、切れ目が一都市分ずれているもの
- c'' : ドーナツ型で、切れ目が二都市分以上ずれているもの
- o : 歯車型
- o' : 歯車型であと一回の交叉で最適解になるもの
- o'' : 歯車型で二回以上の交叉で最適解になるもの

Fn : 風車型(羽の数)

各半径比について、それぞれ100回ずつ実験を行ない、100世代目の最良解を上記に従って分類し、その割合を求めた。結果は図3の通りである。

(7) について各収束型が最初に現れた世代の平均は以下の通り

c	56.833
c'	48.530
o	76.900
o'	70.172

5.4 考察

収束する解の割合は、半径比が小さい場合、c 型に多く収束し、大きい場合には o 型に多く収束する傾向が観測された。これより提案した遺伝システムの設定は、目的関数（評価関数）を反映したものであるといえる。

また、100 世代と比較的少ない世代で実験を行

なったにもかかわらず、最適解、準最適解に収束することから、組合せ最適化問題に対する GA の有効性が示された。

半径比 0.3 : 0.5 の結果では、当初予想していなかった図 2 のような型が観測された。この、扇風機の羽のような型（以下 F 型）は、c 型と o 型の合いの子と見ることができる。この場合の収束の様子を世代を追って確認すると、Fin の部分が共通する個体が大勢を占めていることが観測された。このことは提案する遺伝システムの設定が、親の形質、すなわちサブツアーの状態を保存する能力を有することを示している。このような局所解の生成は、親の形質を保存する本遺伝システムの設定がもたらす悪影響とみなせるが、これはむしろ、解の評価をツアーの長さのみで行っていたことに原因があると思われる。淘汰の評価基準を別の角度から検討する必要があるだろう。

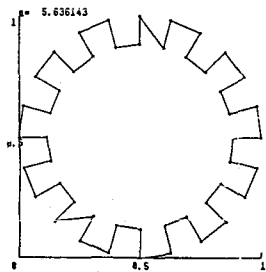


図1 歯車型（O型），風車型（C型）局所最適解

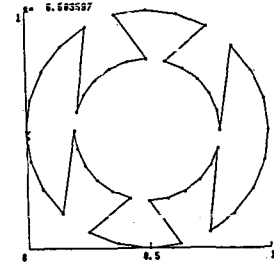
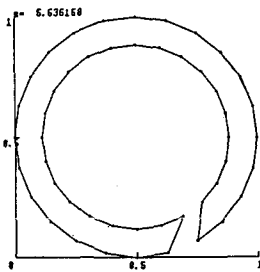


図2 F型（F型）局所最適解

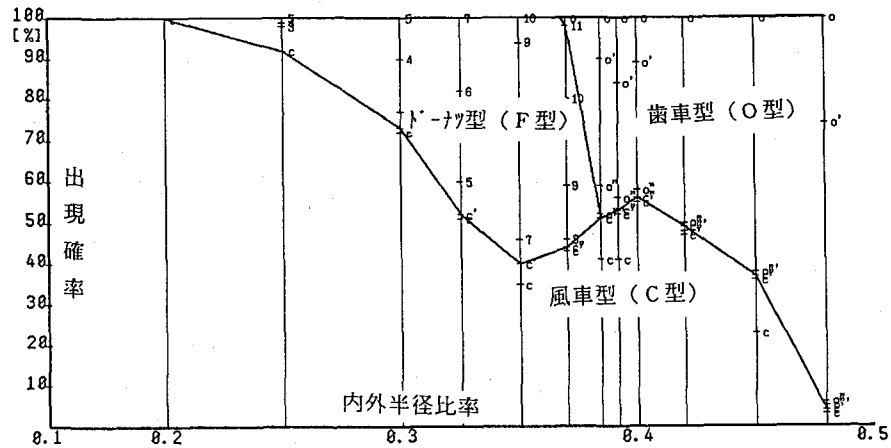


図3 局所最適解の出現確率

6 結論

本研究では、GA においては親の形質が子に継承されることが重要との立場をとり、特に遺伝子のコード化と交叉オペレータの設定が本質的であることから、親の形質を保存する遺伝子の表現方法と、遺伝オペレータを提案した。組合せ最適化問題の一つである TSP への適用を試み提案する形質遺伝型 GA が十分に優れた性能を持つことを実験的に示した。

参考文献

- [Grefenstette 85] John J. Grefenstette, Rajeev Gopal, Brian Rosmaita, Dirk Van Gucht, Genetic Algorithm for the Traveling Salesman Problem, Proceeding of 1st International Conference on Genetic Algorithm and their Applications, 1985, John J. Grefenstette(editor).
- [Goldberg 85] David E. Goldberg and Robert Lingle, Jr., Alleles, Loci and the Traveling Salesman Problem, Proceeding of 1st International Conference on Genetic Algorithm and their Applications, 1985, John J. Grefenstette(editor).
- [Whitley 89] Darrel Whitley, Timothy Starkweather and D'Ann Fuquary, Scheduling Problems and Traveling Salesmen: The Genetic Edge Recombination Operator, Proceeding of 3rd International Conference on Genetic Algorithm and their Applications, 1989, J. David Shaffer(editor).
- [Cleveland 89] Gary A. Cleveland and Stephen F. Smith, Using Genetic Algorithms to Schedule Flow Shop Releases, Proceeding of 3rd International Conference on Genetic Algorithm and their Applications, 1989, J. David Shaffer(editor).
- [Muhlenbein 89] H. Muhlenbein, Parallel Genetic Algorithms, Population Genetics and Combinatorial Optimization, Proceeding of 3rd International Conference on Genetic Algorithm and their Applications, 1989, J. David Shaffer(editor).
- [Goldberg 89] David E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company Inc., 1989, David E. Goldberg(editor).
- [Syswerda 89] Gilbert Syswerda, Uniform Crossover in Genetic Algorithms, Proceeding of 3rd International Conference on Genetic Algorithm and their Applications, 1989, J. David Shaffer(editor).
- [Davis 87] Lawrence Davis and Martha Steensrup, Genetic Algorithms and Simulated Annealing: An Overview, Genetic Algorithms and Simulated Annealing, Morgan Kaufmann Publishers inc., 1987, Lawrence Davis(editor).
- [Montana 89] David j. Montana and Lawrence Davis, Training Feedforward Neural Networks Using Genetic Algorithms, Proceeding of 11th International Joint Conference on Artificial Intelligence, (IJCAI-89).
- [菅原 91] 菅原 隆 山村 雅幸 小林 重信, 形質遺伝を重視した Genetic Algorithm による Traveling Salesman 問題への接近, 第 13 回知能システムシンポジウム, 1991
- 探索空間上で距離の離れた複数の局所解を持つ TSP の例題として、二重円上に都市が配置されている問題をとりあげ、実験結果から収束比率が解の評価値と遺伝オペレータの設定を反映したものになることを確認した。
- 遺伝オペレータの設定により、解の収束率が影響を受けることが確認され、GA を組合せ最適化問題に適用する際には、遺伝子のコード化、遺伝オペレータの設定が重要な問題であることが再確認できた。