

# Tree-染色体構造を持った GA での関数同定用の局所探索手法による同定高速化

Speed-up by Local Search Methods for Tree Chromosome Structure in a Genetic Algorithm to Identify Functions.

又吉 光邦\*<sup>1</sup>

Matayoshi, Mitsukuni

\*<sup>1</sup> 沖縄国際大学

Okinawa International Univ.

In this paper, a comparison of the speed-up methods in “Speed-up by Local Search Methods for Tree Chromosome Structure in a Genetic Algorithm to Identify Functions” is reported. In the speed-up method, there are a compulsory zero making, a compulsory change in variables or constants, and compulsory changes in operators. These speed-up methods function as a local search method in the genetic algorithm. In consideration by testing results, a qualitative discussion of the identification speed-up by each method is given.

## 1. はじめに

関数を同定する有名なシステムに Langley(1987)[1] の BACON がある。また、近年では遺伝的プログラミング ( Genetic Programming ; GP ) で関数同定を取り扱った伊庭 [2] の STROGANOFF や Koza[3] の研究も有名である。

進化計算手法による関数同定の研究論文のほとんどは GP によるものであったが、それは一般的に非常に時間がかかり、また、同定された関数は、いわゆるリスト構造の非常に長い表現式になっているため、その後人間の手作業により式の簡略化が必要である。そこで我々は、これらを解決すべく一般的な GA による関数の同定手法を考えた ( 又吉 [4] )。

本論文では、又吉 [4] の手法の “Tree-染色体構造を持った GA での関数同定” 用の局所探索法を 3 つ考案し、関数の同定の成功率の向上を目指すと共に、解の発見時間の短縮を目標としてその性能および性質を実験結果を基に報告する。

## 2. Tree-染色体構造を持った GA での同定

Tree-染色体構造 ( 図 2. 参照 ) の詳細は、文献 [4] を参照されたいが、関数式染色体 (Function Chromosome)、ポインタ染色体 (Pointer Chromosome)、変数染色体 (Variable Chromosome)、定数部染色体群 (Constant Chromosome Set)、演算子染色体 (Operator Chromosome) から構成されている。関数式染色体とポインタ染色体は、染色体ならびにそれを構成する遺伝子レベルにおいて一対一の対応関係があり、従って染色体の数も同数である。この 2 つの染色体は、交叉や突然変異による遺伝子操作を受けても、一対一対応が崩れることはない。また、変数染色体と演算子染色体は、それぞれ “ただ 1 つの染色体” であり、交叉や突然変異などの遺伝子操作を受けない固定された染色体である。定数部染色体群は、定数を求めるための染色体の集まりで、複数個の染色体が存在し、交叉や突然変異などの遺伝子操作を受け、進化計算の対象となる染色体である。ここで定数部染色体は、小数を含む演算が出来るように以下のように構成されている。

連絡先: 沖縄国際大学産業情報学部, 宜野湾市 2-6-1, 沖縄県 Dept. of Industry and Information Science, Okinawa International Univ., 2-6-1 Ginowan Okinawa, 901-2701 Japan

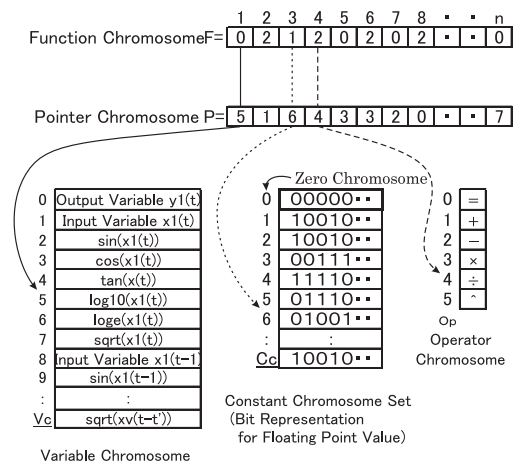


図 1: The Tree Chromosome Structure.

### 定数部染色体

$$= \underbrace{0 + \dots + 1 + 0}_{\text{定数部: } m \text{ ビット}} \quad \underbrace{0 + 1 + \dots + 0}_{\text{小数部: } n \text{ ビット}}$$

### 定数部染色体の値

$$= 0 \cdot 2^m + \dots + 1 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + \dots + 0 \cdot 2^{-n}.$$

ただし,  $m, n$  は正の整数。

### 2.1 関数の値を求める方法

染色体として表現されている同定中の関数について、その入力 ( 観測データ ) に対する出力値を得るには、染色体から関数式を生成し、それに変数の値として観測データを与えて計算しなければならない。我々はこの実現するために、スタックを用いた計算法を採用している [4]。演算子染色体にある “=” 演算子は、関数式の終了を示す終了演算子として機能する。

### 2.2 GA の基本オペレータ

Tree-染色体構造の一部を構成する 3 つの染色体 ( 関数式染色体, ポインタ染色体, 定数部染色体 ) に対して、交叉, 突然

変異，選択の GA の基本オペレータが施されるが，遺伝操作後に染色体が“関数式”の形をなさない場合は，致死染色体となるので，各基本オペレータで注意が払われる。

<< 交叉 >> 関数式染色体とポインタ染色体，および定数部染色体の交叉として一点交叉を用いる。ここで，関数式染色体とポインタ染色体の交叉する遺伝子座は同一である。

<< 突然変異 >> 突然変異は，突然変異を施す染色体によって，遺伝子の生成が異なる。関数式染色体とポインタ染色体については，関数式染色体の遺伝子座の遺伝子の変異が初めに行われ，その後，関数式染色体の遺伝子座に対応するポインタ染色体の遺伝子座の遺伝子の変異が行われる。関数式染色体の変異された遺伝子の遺伝子座が変数あるいは定数の箇所なら，変異により生成される遺伝子は変数か定数となる。

<< 選択 >> 選択は，エリート選択法を用い，次世代に現世代のエリート（最良個体）を必ず残す。すなわち，関数式染色体に対応するポインタ染色体とポインタ染色体によって指されている定数部染色体が，次世代に確実に残される。

<< GA の基本オペレータを施さない染色体 >> 変数染色体と演算子染色体は，交叉や突然変異の対象にならない染色体である。また，定数部染色体にあるゼロ染色体（常に数値=0の染色体）も同様に GA 基本オペレータを施さない。

### 3. Tree-染色体構造を持った GA での関数同定における局所探索

本論文において我々は「Tree-染色体構造をもった GA での関数同定」専用の局所探索法の 3 つの方法それぞれによる関数同定に要する時間の短縮などの効果を検証することを目的としている。次の 3 つが，検証する局所探索法である。

1. 変数・定数強制ゼロ化
2. 変数・定数強制変数化
3. 演算子強制変更

#### 3.1 変数・定数強制ゼロ化

変数・定数強制ゼロ化は，変数，あるいは定数を強制的に定数のゼロに設定するものである。例えば， $m$  を視点の高さ， $r$  を物体までの距離としたとき，仰角  $x$  からある物体の高さ  $f(x)$  を求める次の (1) 式において，右辺の第 4 項以降は余分な式となる。そこで，右辺の第 4 項以降を強制的にゼロ (2 式参照) にすれば，正しい解を得ることが出来る。

$$f(x) = m + r \cdot \tan(x) + x - 1/x^2. \quad (1)$$

$$f(x) = m + r \cdot \tan(x) + 0.0 - 0.0/x^2 \quad (2)$$

#### 3.2 変数・定数強制変数化

変数・定数強制変数化は，関数式染色体の変数あるいは定数を強制的に変数に置き換え，かつ関数式染色体と対をなすポインタ染色体の遺伝子を変数染色体の遺伝子を指すように強制する方法である。簡単な例として，原点を中心とする円の方程式の観測データが得られた場合の関数の同定中に，次の式が得られたとする。

$$f(x, y) = x \cdot y + y \cdot y \quad (3)$$

上式において， $x \cdot y$  の箇所の  $y$  を  $x$  に換えれば，(4) 式の原因を中心とした円の方程式を得ることができる。

$$f(x, y) = x \cdot x + y \cdot y \quad (4)$$

表 1: 実験に用いた関数の観測データ

関数	観測変数	未知変数	備考
(a)	$Y, t$	$\frac{1}{2}, g$	$0 \leq t \leq 20(s)$
(b)	$h, r, x$	$m$	$x$ : 仰角, $m$ : 1.5(視点高) $0 < x \leq 90, 0 < r \leq 10$
(c)	$Y, x_1 \sim 6$		$0 \leq x_1 \sim 6 \leq 10$

表 2: Tree-染色体構造での各種設定

同定関数	観測変数	関数式染色体長	使用関数	遅延
(a)	2	31	(0)	-
(c)	3	31	(3)[6]*	-
(d)	7	31	(0)	-

( ) : 自作関数の数 ;  $non(), not(), zero()$  .

[ ] : 超越関数の数 ;  $org1(), sin(), cos(), tan(), ln(), log()$  .

\* ; すべての自作関数とすべての超越関数 .

### 3.3 演算子強制変更

演算子強制変更は，基本的に四則演算子とべき乗演算子の合計 5 つの演算子を強制的に変更する方法であるが，“0”による除算を避けるため除算子への強制変更は行わない。この演算子強制変更による例としては，次の (5) 式を考えたとき，

$$f(x) = m \cdot r + \tan(x) + x \cdot 0.0. \quad (5)$$

$m \cdot r$  に用いられている乗算子を  $m + r$  の加算子に変更し， $r + \tan(x)$  に用いられている加算子を乗算子に変更して  $r \cdot \tan(x)$  にすれば，(6) の視点の高さと仰角から物体の高さを求める関数を得ることに成功することが分かる。

$$f(x) = m + r \cdot \tan(x) + x \cdot 0.0. \quad (6)$$

### 3.4 局所探索法の移動戦略

我々の提案する局所探索法は，近傍操作による探索（近傍探索）を用いている。我々の提案する近傍探索法は，適当な解  $\pi$  とその近傍解  $\pi'$  の染色体上の違いは，高々 1 である。検証する局所探索法に用いた移動戦略法は，最良移動戦略 (Best Admissible Move Strategy; BAMS) を用いた [5]。最良移動戦略 (BAMS) は，近傍  $N(\pi)$  を調べて近傍  $N(\pi)$  の中の最良解を改善解  $\pi'$  として移動する方法である。改善解に移動後， $\pi := \pi'$  として得られた新しい近傍  $N(\pi)$  を再度調べる操作を改善解が，もはや近傍に見つからなくなるまで繰り返す。

## 4. 実験

### 4.1 実験に用いた関数

本論文で実験に用いた関数，すなわち同定される関数（以下，同定関数）を以下に示す。同定関数 (a) ~ (c) は，文献 [4] から得ている。また，今後の関数同定の実験の参考のために，観測されたデータとして扱っている同定用データ作成についての詳細を表 1 に示し，今回の実験で関数式染色体が取る各種設定値を表 2 に示す。

$$(a) Y = \frac{1}{2}gt^2.$$

$$(b) h = r \cdot \tan(x) + m.$$

$$(c) Y = x_1x_2x_3 + x_4x_5x_6.$$

表 3: (a):自由落下の実験結果

	同定成功数	平均発見時間	標準偏差	平均式長
Non	88	11.60	9.87	6.63
Zero	97	9.08	7.66	6.71
Var.	92	9.23	7.51	6.60
Ope.	92	12.67	12.07	6.15
Rand.	94	8.75	9.09	7.23

表 4: (b):二箱問題の実験結果

	同定成功数	平均発見時間	標準偏差	平均式長
Non	82	53.43	48.90	11.17
Zero	85	98.10	73.45	11.40
Var.	75	95.54	79.09	11.34
Ope.	60	107.83	83.25	11.23
Rand.	78	88.95	80.20	12.15

#### 4.2 関数同定の成功要件

関数の同定は、誤差率  $G \leq \alpha$  かつ、以下に示される目的を達成したときに成功とする。ここで、適応度は、誤差率を用いており誤差率が小さいほど適応度が高いことを示す。

- (a) 自由落下：法則と定数  $\frac{1}{2} \cdot g \simeq 4.9$  の同定。
- (b) 物体の高さ：超越関数  $\tan(x)$  と未知の定数  $m$  の同定。
- (c) 二箱問題：観測変数が多い場合の関数を同定。

ここで、自由落下の法則の同定で定数項 " $\frac{1}{2} \cdot g \simeq 4.9$ " としているのは、今回我々の定数項を表現するビット長が 10 ビット (整数部 5 ビット, 小数部 5 ビット) であり、正確に "4.9" が表現できないためである。

本論文で使用されている関数は、合計 9 種類の関数がある。まず、自作関数として  $non(x)$ ,  $not(x)$ ,  $zro(x)$  があり、超越関数として  $\sin(x)$ ,  $\cos(x)$ ,  $\tan(x)$ ,  $\ln(x)$ ,  $\log(x)$ ,  $org1(x)$  がある。ただし、 $org1(x) = (x + \frac{1}{x})(x^2 + \ln(x))$  は、自作の関数であるが、関数内に  $\ln(x)$  があるので、超越関数の 1 つとして扱っている。 $non(x)$  は、変数  $x$  をそのまま返す関数であり、 $not(x)$  は  $|x| > 0$  なら "0",  $x = 0$  なら "1" を返す関数。 $zro(x)$  は、 $x$  の値に関わらず、"0" を返す関数である。

#### 4.3 計算機資源と GA の各種設定

実験は、Pentium-III 2GHz, Windows Me, Java で行った。実験の制限時間:5 分とし、収束判定は、誤差率  $G \leq 5\%$  としている。各染色体等の設定値は、関数式染色体数:100, ポインタ染色体数:100, 定数部染色体数:50, 演算子染色体数:1。また、定数部染色体長:10 (整数部:5 ビット, 小数部:5 ビット) としている。関数式染色体とポインタ染色体、および定数部染色体の突然変異率は、30%。関数式染色体とポインタ染色体の交叉率、および定数部染色体の交叉率は、突然変異を受けていない染色体 2 対をランダムに選び出し行うので、新しい親集団 (染色体) の全体に占める割合は、約 70% である。我々の手法では、解の多様性を維持するため、エリート以外の染色体は、ほぼ遺伝子の操作を受けさせている。各未知関数の同定実験回数は、100 回。観測データは、入力と出力の対で 20 個用意されている。

#### 4.4 実験結果・課題

表 3, 表 4, 表 5 に (a),(b),(c) のそれぞれ実験 100 回における関数同定の成功数, 関数同定成功時の平均発見時間, 同定

表 5: (c):木の高さの実験結果

	同定成功数	平均発見時間	標準偏差	平均式長
Non	53	163.71	86.18	8.42
Zero	29	158.19	94.09	8.93
Var.	19	175.43	81.95	8.05
Ope.	17	126.87	96.17	8.29
Rand.	42	124.05	89.89	9.47

した関数の平均式長を示す。標準偏差は、平均発見時間について求めたものである。表中、Non は、局所探索法を併用しない場合であり、Zero は、変数・定数強制ゼロ化、Var. は、変数・定数強制変数化、Ope. は、演算子強制変更、Rand は、局所探索法として Zero, Var., Ope. からランダムに 1 つ選び実行した結果を示す。

表 3 を見ると局所探索法併用が効果を示していることが分かる。一方、表 4, 表 5 は、局所探索法併用の効果が得られていないことを示す結果となっている。今回、考案した局所探索法は、同定する関数によって効果が得られるか否かが分かる。自由落下の法則のような単純な関数の同定には効果があるが、超越関数を含む関数の同定には、不向きのようなのである。超越関数を含む関数の同定には、実験結果のデータを観察すると局所解、すなわち近似解に捕らわれて探索終了、あるいは探索時間終了となっているケースが多く見られた。また、二箱問題では、解の発見数は Ope. 以外で大差がないものの解の発見時間に大きな開きがあり、局所探索法を併用しない方が短時間で解けていることを示している。これは、二箱問題がシンプルであることを考慮すれば、局所探索法にかかる時間が影響していると我々は考えている。

今後の課題として、局所探索法そのものの高速化、近似解からの効果的な脱出が挙げられる。また、Rand. では良好な結果を得ていると考えられるので、Zero, Var., Ope. の各局所探索法をどのタイミングで行うかを決定する研究も取り組みたい。

#### 参考文献

- [1] Langley, P., Simon, H. A., Bradshaw, G. L. and Zytkow, J. M., "Scientific Discovery: Computational Explorations of the Creative Process", Cambridge, MA: MIT Press, 1987.
- [2] Hitoshi, Iba et al., "System Identification using Structured Genetic Algorithms", Proc., 6th Int. Conf. on Genetic Algorithms(ICGA 95), 1995.
- [3] Koza, J. R., "Genetic Programming: On the Programming of Computers by Means of Natural Selection and Genetics", Cambridge, MA: MIT Press, 1992.
- [4] Mitsukuni Matayoshi, "Three Chromosome Structure in a Genetic Algorithm to Identify Functions", Systems and Computers in Japan, Vol. 31, No. 10, 2000.
- [5] 柳浦睦憲, 茨木俊秀, "組み合わせ最適化 - メタ戦略を中心として -", 朝倉書店, 2001.