

## 遺伝的アルゴリズムにおける関数同定の一手法

又吉光邦

名城大学 国際学部 経営情報学科

沖縄県名護市為又 1 2 2 0 - 1

matayosi@mis.meio-u.ac.jp

あらまし

本論文では、遺伝的アルゴリズムを用いて未知の関数の同定に成功した一つの方法を示す。本論文で取り上げる未知の関数は、入力変数が単一／多変数であり、かつ単一出力を出すものであり、代数関数および初等超越関数で表せる括弧のない展開された関数である。例)  $f(x,y)=x^2+\cos(y)$  の形をしたものをいう。また、何らかの定義された関数をプログラムに組み込むことで、その関数を使った未知関数の同定も可能となる。実験結果では、本論文で提案した手法が、高速に未知の関数の同定に成功していることが示されている。また、発見されたいくつかの問題点も述べる。

キーワード 遺伝的アルゴリズム, 関数同定, システム同定

### A Method for Function Identification using a Genetic Algorithm.

Mitsukuni Matayoshi

Department of Management and Information Sciences, Meio University.

1220-1, Bimata Nago Okinawa, 905 Japan.

Abstract

This paper introduces a method using a Genetic Algorithm for identifying unknown functions. The method focuses on functions of more than one variable which produce a single output value. Unknown functions consisting of combinations of algebraic functions, trigonometric functions and logarithmic functions, such as  $f(x,y)=x^2+\cos(y)$ , can be identified correctly.

Furthermore, user-defined functions can be added to the solution-set of identifiable functions. Experimental tests produced a high success rate, although further consideration of failed attempts is necessary.

key words Genetic Algorithm, Function identification, System identification.

## 1 はじめに

科学者が観測されたデータから法則を発見するように、機械にデータを与えるだけで自動的に法則を発見させる研究は、人工知能の分野で古くから行われてきた。そしてそれは、人類の長年の夢でもある。特に、[koza, 92]の提唱した遺伝的プログラミング（以下、GP:Genetic Programming）による方法は、関数の持つ構造的表現を直接遺伝子コードとして扱えるように遺伝的アルゴリズム（以下、GA:Genetic Algorithm）を拡張し、GAの基本的なオペレータである交差や突然変異などを木構造に適用し、プログラミングLISP(S式)上でや概念木を探索することができる。つまり、任意の法則などを発見することができる画期的な方法である。

しかしながら、現状のGPの大きな問題は、計算量が多いことである。実際の実験の実行では、数日かかることも珍しくない[伊庭,96]。これは、グラフ構造の多様性を保持するために初期集団数が一般的なGAに比べて桁違いに多いことに起因している。また、交差・突然変異によるスキーマの破壊によって、有益な部分木構造が失われたり、木の意味が劇的に変化する意味破壊を繰り返すため、探索が安定せずに時間が費やされる。これは、GAにも見られることであるが、GPでは初期集団数が多いため影響は大きい。

一方、木構造を用いない、いわゆるGAを用いた関数の同定問題は、ほとんど手が着けられていない。これは、GAがある制約条件の下、既知の関数の最適値探索を行うことを目的とした探索アルゴリズムとして設計されることが多いからである。もちろん[伊庭,95]の囚人のジレンマ問題におけるしつぺ返しスキーマを遺伝子上に出現させるものなど、いろいろ工夫を凝らした興味深い研究も数多くなされている。

本論文では、GAで未知の関数の同定を行う手法を提案している。この手法を用いれば、未知関数の複雑さにもよるが、 $f(x, y) = x^2 + \cos(y)$ 程度の問題ならPentium166MHz程度のノート型パソコンで1分以内に解くことが可能である。

## 2 法則の発見

観測データからそれに対してあてはまる関数の形式と適切な係数を求めることを目的とした記号

あてはめは、科学者が観測データから試行錯誤の後に何らかの法則を発見する行為を計算機を用いて実現しようとするものである。この種の代表的なシステムにBACON[Langley,87]があり、近年では[koza, 92]や[伊庭, 94]のSTROGANOFFが有名である。

BACONでは、様々なヒューリスティクスを用い変数間の関係、定数となる関係を導き出している。

一方、[伊庭, 94]らの研究では、統計的手法を用いて、ランダムな係数導出のため適切な係数の発見がきわめて困難であったGPの欠点を克服し、数々の記号当てはめの問題を解いている。

現在では、記号当てはめに代表されるような何らかの法則を計算機を用いて自動的に効果的・効率的に発見するのは、GPの主要な研究課題であり、かつ独壇場であるといつて過言でないと思われる。

本論文で述べる手法は、一般的なGAを用いた記号当てはめ的手法であり、システム同定問題の解法に用いられる多変量解析手法などの統計的手法すら用いていない非常にシンプルなものである。そしてそれゆえ、実計算時間も非常に小さく、かつこれからの応用範囲に期待がもてる手法である。しかも、超越関数などが簡単に組み込めることも大きな特徴である。

従来のGAを取り巻く研究は、最適値探索という限られた研究領域が主であったが、これによりシステム同定、時系列予測などの研究が進むことが期待できる。

## 3 GAによる関数同定

本論文で述べる手法は、基本的に単純GAである。しかしながら、そのデータ構造に工夫がなされている（図1参照）。

### 3.1 染色体構造

GAで用いる染色体構造は、本論文の主要な点の一つである。以下に述べるようなデータ構造を用いることで、GAにおいて記号あてはめをはじめとする関数同定問題を解くことが可能となる。

以下、図1に染色体構造を示す。

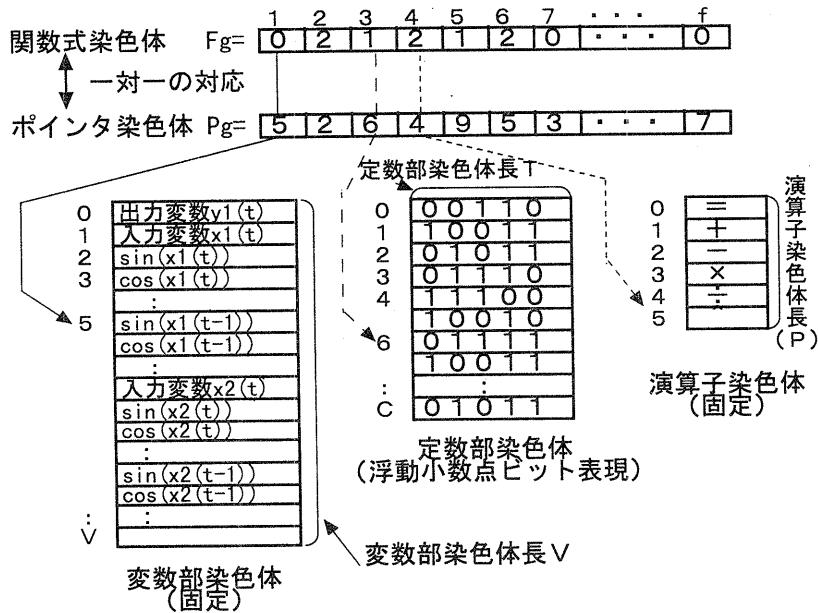


図1 染色体構造

### 3.2 関数式染色体の構造

関数式染色体は、未知関数のコード化された表現であり、以下の規則に従って、その構造が決められている (図1 参照)。

【規則】

$$1 \leq \text{関数式の染色体長} \leq f \quad (f = 1, 2, \dots, n)$$

$$f = \begin{cases} \text{odd} & (\text{変数部染色体 or 定数部染色体}) \\ \text{even} & (\text{演算子染色体}) \end{cases}$$

$$Fg[f] = \begin{cases} 0 & (f:\text{odd} \text{ 変数部染色体を参照}) \\ 1 & (f:\text{odd} \text{ 定数部染色体を参照}) \\ 2 & (f:\text{even} \text{ 演算子染色体を参照}) \end{cases}$$

ところで、関数式染色体の構造には、GAの基本オペレータである、突然変異、交叉法 (一転交叉)、保存法 (エリート保存法) を適用する。

### 3.3 ポインタ染色体の構造

ポインタ染色体は、関数式の染色体と一対一の対応があり、関数式染色体で現れた \$F\_g\$ の値に従っ

て、変数部染色体、定数部染色体、演算子染色体のいずれかの染色体を選び、ポインタ染色体の \$P\_g\$ の値で示される遺伝子や染色体 (定数部染色体の1つの個体) を選択する。

【規則】

$$1 \leq \text{ポインタ染色体長} \leq f \quad (f = 1, 2, \dots, n)$$

$$f = \begin{cases} \text{odd} & (\text{変数部染色体 or 定数部染色体}) \\ \text{even} & (\text{演算子染色体}) \end{cases}$$

$$Pg[f] = \begin{cases} V & (V \in \text{正整数, 変数部染色体長}) \\ C & (C \in \text{正整数, 定数部個体数}) \\ P & (0 \leq P \leq 5, \text{演算子染色体長}) \end{cases}$$

注) \$V, C, P\$ については、3.4 ~ 3.6 を参照。

ところで、ポインタ染色体の構造には、GAの基本オペレータである、突然変異、交叉法 (一転交叉)、保存法 (エリート保存法) を関数染色体の構造に適用したのと同じように適用する。

以上、2つの染色体構造が、未知の関数の同定を行う主要染色体である。

### 3.4 変数部染色体の構造

変数部染色体は固定長であり、3.2 関数式の染色体、3.3 ポインタ染色体と異なり、GA オペレータを施さない、確定遺伝子である。これは、ちょうど生物界にある、「人」「猫」その他の生物を確定する遺伝子にヒントを得ている。あるいは、「人」には、「眼」「鼻」「口」... があり、それらはおおむね同じ形をしていることにも通じる。

これと同じ確定遺伝子であるのが、演算子染色体である。

変数部染色体は、固定長であるが、これは以下の規則に則って生成される。ところで、この部分の構造は、実装時に計算処理により求めている。すなわち、プログラム上で論理的に存在しているにすぎないため、記憶領域（メモリ）の消費などは無い。

【規則】

時刻  $t$  の出力変数:  $Y(t)$  ( $1 \leq t \leq$  入力データ数)

時刻  $t-m$  の入力変数:  $X_n(t-m)$

( $0 \leq m < t$ :  $m$  は遅延)。

また、現在用意してある三角関数と対数関数ならびに平方根、そして何もせずにそのまま値を返す *non* 関数（以後、まとめて特殊関数と呼ぶことにする）は、時刻  $t-m$  の入力変数  $X_n(t-m)$  ( $0 \leq m < t$ ) の時、以下のようになる。

ここで、添え字は特殊関数の番号を示す。  
 $\text{non}(X_n(t-m))$ ,  $\text{sin}(X_n(t-m))$ ,  $\text{cos}(X_n(t-m))$ ,  
 $\text{tan}(X_n(t-m))$ ,  $\text{ln}(X_n(t-m))$ ,  $\text{log}(X_n(t-m))$ ,  
 $\text{sqrt}(X_n(t-m))$ .

ここで、もし観測データが時系列なら  $Y(t)$ ,  $X_m(t-m)$  は時変数となり、それ以外はある時点に観測されたというだけのデータを示す。従って、この場合、観測データの順序を入れ換えても問題はない。

ところで、任意の入力変数をそれぞれの関数に代入しようとした時、計算不可能な場合が存在する。例えば  $\text{sqrt}(X_n(t-m) \leq 0)$  の時、すなわち、変数が負で平方根を求めようとした場合などである。このような場合が発生した時、本手法では統一して、致死遺伝子（評価値最小）として処理している。

以上より、変数部染色体の長さは、次のように求まる。

変数部染色体長  $V = (\text{入力変数の個数}) \times$   
 $m \times \text{用意してある関数の個数}.$

従って、変数部染色体とその遺伝子は、以下のようになる。

遺伝子 =  $\{V\}$  ( $V \in$  正整数)。

染色体 =  $\bigcup_{i=0}^{+\infty} V^i$ .

【例】出力変数の個数 = 1, 入力変数の個数 = 2, 特殊関数の個数 = 7 (上述の 7 個), 時刻  $t = 10$ , 遅延  $m = 5$  で、関数式の染色体の遺伝子 = 0, ポインタ染色体の遺伝子 = 46 の時、

$46 \div (7 \times 5) = 1$  (入力変数名) .

$46 \bmod (7 \times 5) = 11$ .

$11 \div 7 = 1$ . (遅延)

$11 \bmod 7 = 4$ . (関数番号) .

より、 $\text{ln}(X_1(10-1))$  が計算され値を返す。

### 3.5 定数部染色体の構造

定数部染色体の構造は 2 次元の配列で構成されており、列方向は定数部染色体の個数、行方向に一つの染色体が存在する。

また各染色体は、以下に示すように浮動小数点を扱うための構造をしている。

定数部染色体個数 =  $C$  ( $C \in$  正整数)。

遺伝子 =  $\{0, 1\}$ .

染色体長  $T = m + n$ . ( $m, n \in$  正整数)

染色体

$$= \underbrace{\underbrace{0 + \dots + 1 + 0}_{\text{定数部: } m \text{ ビット}}}_{i=0}^{i=m} + \underbrace{\underbrace{0 + 1 + \dots + 0}_{\text{小数部: } n \text{ ビット}}}_{j=1}^{j=2} + \dots + \underbrace{0}_{j=n}^{j=n}.$$

定数部染色体の値

$= 0 \cdot 2^m + \dots + 1 \cdot 2^1 + 0 \cdot 2^0 +$

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

【例】染色体長 = 7 ( $m = 4, n = 3$ ) のとき、定数部染色体の一つが [0100110] である時、その値は以下のようになる。

定数部染色体の値

$= 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 +$

$1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} = 4.75.$

ところで、定数部染色体には、GA の基本オペレータである、突然変異、交叉法（一転交叉）、保存法（エリート保存法）を適用する。

### 3.6 演算子染色体の構造

演算子染色体は、変数部染色体と同様に固定長であり、前述した 3.2 関数式の染色体、3.3 ポイン

タ染色体, および 3.5 定数部染色体と異なり, GA オペレータを施さない, 3.4 変数部遺伝子と同じ確定遺伝子である.

演算子染色体は, 必要な演算子を示すだけの 1 次元配列で作られるものである. 以下に演算子染色体の規則を示す.

ところで, この部分の構造も変数部染色体と同様, 実装時に計算処理により求めているので, 記憶領域 (メモリ) の消費などは無い.

#### 【規則】

遺伝子 =  $\{0, 1, 2, 3, 4, 5\}$ . (Genotype)

遺伝子 =  $\{=, +, -, \times, \div, ^\wedge\}$ . (Phenotype)

演算子染色体長  $P = 6$  ( $0 \leq P \leq 5$ ).

演算子遺伝子の優先順位と遺伝子との対応:

$$\frac{0}{=} < \frac{1}{+} \leq \frac{2}{-} < \frac{3}{\times} \leq \frac{4}{\div} < \frac{5}{^\wedge}.$$

ただし,  $^\wedge$  は累乗を表し, 例えば  $3^\wedge 4$  なら  $3^4 = 81$  を示す.

【例】関数式染色体の遺伝子 = 2, ポインタ染色体の遺伝子 = 3 の時,

演算子染色体上の遺伝子 = 3 (Genotype)  
=  $\times$  (Phenotype).

従って, 演算は「 $\times$ 」, つまり「積」が行われる.

## 4 GA オペレータ

本論文で提案した手法では, 関数式染色体, ポインタ染色体, 定数部染色体に GA の基本オペレータである交叉法, 突然変異, 選択法が必要である. 染色体に対する 3 つの基本オペレータは, 同一の方法を採っており, 以下のようになる.

交叉法 : 一点交叉

突然変異 : 遺伝子の強制的変換

選択法 : エリート保存法

交叉法と突然変異は, 確率的に行われる.

## 5 実験

本論文の実験に用いた関数を, 以下に示す.

- (1)  $Y = \frac{1}{2}gt^2$ . (自由落下の法則)
- (2)  $r^2 = x^2 + y^2$ . (原点を中心とする円)
- (3)  $h = r \cdot \tan(x) + m$ . (仰角からの高さ計算)
- (4)  $Y = x_1x_2x_3 - x_4x_5x_6$ . (二つ箱の問題)
- (5)  $a_{n+2} = a_{n+1} + a_n$ . (フィボナッチ数列)

$$(6) N_{m+1} = N_m(1 + \alpha - \beta N_{m-1}).$$

(遅延時間のある離散ロジスティック成長モデル)

### 5.1 物理学の法則

ここでは, 本手法で (1) ~ (2) の物理学の法則を求めてみる. (1) は, 自由落下の法則であり, ここで求めたいのは, その法則の完全形だけでなく定数の  $\frac{1}{2} \cdot g (= 4.9)$  である. 一般的な GP では, ランダムに定数生成を行っている. 本手法では, 定数生成も GA で行っている.

一方, (2) は, 円の方程式である. これは観測データの作り方に工夫がしてある. つまり, 円の方程式では, ある  $x$  の値の時に  $y$  の値が「正」と「負」の両方が存在する. このことを考慮に入れ, 「正」「負」両方を観測データとしている.

この問題が, 難しい問題に入るのか否かは, 他の手法との比較ができないので分からないが, 同じ  $x$  が観測された時に符号の違う値 ( $y$ ) が観測されることに人間がこのような現象に遭遇すると, 非常に不可解な現象として悩まされるだろう. そのため, 今回, この問題を取り入れて見た.

まず, 与えられる (観測される) 既知のデータと, 未知のデータを分けると以下の表 1 のようになる.

表 1 物理学の法則の実験関数

| 関数  | 既知データ     | 未知データ            | 備考                           |
|-----|-----------|------------------|------------------------------|
| (1) | $Y, t$    | $\frac{1}{2}, g$ | $t$ : 時刻<br>$g: 9.8$ (重力加速度) |
| (2) | $r, x, y$ |                  | $r$ : 半径                     |

### 5.2 その他の問題

同様に, 本手法で (3) ~ (4) の問題を解いてみる. (3) の問題は, 表に示すように観測されたデータ  $h, r, x$  から,  $\tan$  関数を見つけることと, 観測データからでは, 全く予想されないデータ  $m$  (視点の高さ) を求めることである. (4) の問題は, [伊庭, 94] で紹介された問題で, 二つの箱の体積の差を示す式である.

表 2 その他の問題

| 関数  | 既知データ             | 未知データ | 備考                      |
|-----|-------------------|-------|-------------------------|
| (3) | $h, r, x$         | $m$   | $x$ : 仰角<br>$m$ : 視点の高さ |
| (4) | $Y, x_{1 \sim 6}$ |       |                         |

### 5.3 時系列の問題

次は、本手法で (5) ~ (6) の問題を解いてみる。(5) の問題は、有名なフィボナッチの数列で、前 2 項の和が、次の項の値となるものである。見方を変えれば、2 入力、1 出力の 2 階差分の時系列問題である。

また、(6) の問題は、[稲垣, 92] にある単純化された遅延時間のある離散ロジスティック成長モデルである。

このモデルは、毎年測定されるしかの個体群  $N_m$  を求める際、ある与えられた地域における単位年度間の鹿の増加は、植生に依存するという単純化された 2 階差分方程式モデルである。このモデルで求めたいものは、 $\alpha$  (環境の影響のない成長率) と  $\beta$  (個体群の減少率) である。

| 表 3 時系列問題 |       |                 |                          |
|-----------|-------|-----------------|--------------------------|
| 関数        | 既知データ | 未知データ           | 備考                       |
| (5)       | $a_n$ |                 | $n = 0 \sim 19$<br>観測データ |
| (6)       | $N_n$ | $\alpha, \beta$ | $n = 0 \sim 19$<br>観測データ |

## 6 実験結果

以下、表 4, 5 にもっとも成績のよかったものの実験結果を示す。ただし、全ての実験において、世

代数上限 1000 回、関数式染色体、ポインタ染色体、定数部染色体の各突然変異率を 10% としている。

また、関数式染色体とポインタ染色体の個体数は  $f = 300$  に固定してある。一方、定数部染色体の個体数は  $C = 100$ 、定数部染色体長  $T = 10$  (整数部: 5 ビット, 小数部 5 ビット) に固定してある。

実験回数は、各問題に対して 5 回行う。入力データ数は各問題 20 個用意し、表 5 の誤差和とは、発見した関数が 20 個の入力データから出力する値と観測出力データとの誤差の絶対値和である。

一方、 $f$  の値は、入力変数の数  $\times 2$  以上の設定が必要である。また、変数部の染色体長は、求める関数に合わせて超越関数や遅延を組み込んでいる。本来、一つのコンピュータで一つ一つ問題を解く場合は、代数関数 + 超越関数 + 遅延で実験を行うべきだが、このようにするとなかなか解を発見できないので求める関数に合わせている。この方法に異論があるかと思われるが、複数のコンピュータで並列処理、あるいは J A V A 等を用いて並列プログラミングで実装すれば、今回の実験法は、その中の一つの処理部分と等しいものとなるので大した問題ではない。

ただ、代数関数 + 超越関数 + 遅延でできた未知の関数を見出す場合は、何らかの改良が必要となる。現在、鋭意研究中である。

表 4 各種設定

| 関数                                          | 出力変数の数 | 入力変数の数 | 関数式染色体長 | 超越関数の数 | 遅延の数 |
|---------------------------------------------|--------|--------|---------|--------|------|
| $Y = \frac{1}{2}gt^2$                       | 1      | 2      | 7       | 0      | 0    |
| $r^2 = x^2 + y^2$                           | 1      | 3      | 15      | 0      | 0    |
| $h = r \cdot \tan(x) + m$                   | 1      | 3      | 15      | 6      | 0    |
| $Y = x_1x_2x_3 - x_4x_5x_6$                 | 1      | 7      | 15      | 0      | 0    |
| $a_{n+2} = a_{n+1} + a_n$                   | 1      | 1      | 15      | 0      | 5    |
| $N_{m+1} = N_m(1 + \alpha - \beta N_{m-1})$ | 1      | 1      | 15      | 0      | 5    |
| $x^2 + \cos(y)$                             | 1      | 2      | 15      | 6      | 0    |
| 要旨の中の関数                                     |        |        |         |        |      |

表 5 実験結果

| 関数                                          | 発見関数                                                             | 誤差和                  | 時間(秒)  | 発見世代 | 発見実験数          |
|---------------------------------------------|------------------------------------------------------------------|----------------------|--------|------|----------------|
| $Y = \frac{1}{2}gt^2$                       | $4.906t^2$                                                       | 15.4375              | 112.44 | 1000 | 5*             |
| $r^2 = x^2 + y^2$                           | $x^2 + y^2$                                                      | $1.8 \times 10^{-5}$ | 4.68   | 25   | 5              |
| $h = r \cdot \tan(x) + m$                   | $r \cdot \tan(x) + \frac{14.812}{9.938}$                         | 0.1886               | 4.98   | 44   | 5*             |
| $Y = x_1x_2x_3 - x_4x_5x_6$                 | $x_1x_2x_3 - x_4x_5x_6$                                          | $7.8 \times 10^{-5}$ | 8.12   | 803  | 1              |
| $a_{n+2} = a_{n+1} + a_n$                   | $a_{n+2} = a_{n+1} + a_n$                                        | 2.0                  | 162.45 | 1000 | 5*             |
| $N_{m+1} = N_m(1 + \alpha - \beta N_{m-1})$ | $x_m + x_m - x_{m-1} + \frac{12.125}{x_m} - \frac{2.938}{8.312}$ | 21.7454              | 203.13 | 1000 | 1 <sup>‡</sup> |
| $x^2 + \cos(y)$                             | $x^2 + \cos(y)$                                                  | 0                    | 29.93  | 118  | 5*             |
| 要旨の中の関数                                     |                                                                  |                      |        |      |                |

注) 発見実験数の \* のついたデータは、まるめ誤差を考えた場合、正解であると見なせるもの。

‡ のついたデータは、近似解と見なせる見なせるもの。

実験結果から分かることは、GP等で困難であった定数の割り出しもかなりの精度で求めながら、非常に高速に関数の同定を行っていることである。実験に使用した166MHzの計算機で、数分以内に同定を成功させている。また、 $h = r \cdot \tan(x) + m$ の $m$ のような、観測データからは分からない定数項を求めることができたことは、正直なところ驚きであった。

一方、2つ箱の問題は、なかなか解（実験5回中1回の正答）けないでいる。

また、離散ロジスティック成長モデルなどは、解答そのものに至っていない。しかしながら、この問題に対する本手法が出した「解」での誤差和は約22であり、観測データ一つにつき1.1の誤差を出しているにしかすぎない。従って、かなり質の高い近似解を提示していると考えられる。

## 7 考察

本論文では、一般的な遺伝的アルゴリズムを用いた、記号当てはめをはじめとする関数の同定を行う手法を提案した。この手法では、データ構造に工夫がなされており、定数項、超越関数を含む記号当てはめや時系列関数の同定を行っている。

しかしながら、2つ箱問題、離散ロジスティック成長モデルなどは成績が悪い。また、代数関数+超越関数+遅延の3つを同時に含む問題を解かせてい

ないこと、ならびに括弧付き演算もまだ実装していない。

今後の課題は、これらの問題点を克服することと、まだ法則の発見されていない何らかの観測データに本手法を適用し、法則の発見を行いながら本手法の有効性を検証することである。また、他の手法との比較も行う必要がある。

## 参考文献

- [1] 伊庭斉志, 斉藤泰介, “システム同定アプローチに基づく遺伝的プログラミング”, 人工知能学会誌, Vol.10 No.4 1994.
- [2] Koza, J. R.: Genetic Programming,, On the Programming of Computers by means of Natural Selection, MIT Press, 1992.
- [3] 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.
- [4] 伊庭斉志, “遺伝的プログラミング”, 東京電機大学出版局, 1996.
- [5] 伊庭斉志, “遺伝的アルゴリズムの基礎”, オーム社, 1994.
- [6] 稲垣宣生, “生態系の微分方程式”, 現代数学社, 1992.