平成 12 年年度

学士学位論文

遺伝的プログラミングを用いた時系列処理

Time Series Management with Genetic Programming

1010383　　神谷将司

指導教員　　坂本明雄

Ruck Thawonmas

2001 年 2 月 5 日

高知工科大学 情報システム工学科

要 旨

遺伝的プログラミングを用いた時系列処理

神谷将司

時系列処理問題は古くから研究され，実用価値も高い．そのため応用範囲も色々と考えられる．また実際に実用化されたものもある．この問題の目的は，過去のデータから未来のデータを近似できる関数を求めることである．本論文では，時系列予測問題における例外値を含むデータに対して処理する方法を示し，遺伝的プログラミングを組み合わせた解法について提案する．

キーワード　遺伝的プログラミング，時系列予測，Group Method of Data Handling

Abstract

Time Series Management with Genetic Programming

Kamiya Masashi

A time series prediction has been reserched sice the old days, and value of utility is high, too. Therefore it can think about various application ranges, too. A purpose in this question is to find the function that future data can be appoximated from the past data. A method to deal with to the data which contain exception value in the time series prediction is shown and proposed about the answer law which genetic programming was combined with by this thesis.

key words genetic programming, time series prediction

目次

図目次

第 1 章

序論

時系列予測問題は古くから研究され，その実用価値も高い．そのため応用範囲も色々と考え出され，実際に実用化された物もある．例えば，株価予測や貯水量，商品の売上の予測などが挙げられる．

一方，組合せ最適化問題の探索手法として，遺伝的プログラミング（Genetic Programming）がある，これは遺伝的アルゴリズムの考え方を計算機プログラムの自動生成に利用した進化論的な探索手法である．遺伝的プログラミングではプログラムを一つの個体として扱い，多数の個体を対象に遺伝的な進化をおこない，課題をもっともよく説明できる個体を進化させることが目的となる．遺伝的な手続きを用いることは共通しているが，遺伝的アルゴリズムが答えを直接求めることに対して，遺伝的プログラミングは手続きを求めるという点で違いがある．

本論文では，時系列予測問題を定式化し，これを遺伝的プログラミングで解くための工夫と実験を行った．

2 章では，時系列予測に用いられる GMDH という手法を紹介し，一度の試行でどれだけ近似できるかを実験する．

3 章では，遺伝的プログラミングについて簡単に紹介する．

第 2 章

時系列処理

2.1 時系列処理

ある関数 y があり，この関数 y の観測値（時系列データ）$x(1), x(2), x(3), \cdots, x(t)$ をもとにして，未来のデータの予測値 $x(t+1)$ を過去のデータで近似する関数

$$x(t) = f(x(t-1), x(t-2), x(t-3), \cdots, x(1))$$

を得たいとする．一度この近似関数を得れば過去のデータから未来を次々に予測することが可能になる．このような問題は時系列予測問題（Time Service Prediction）と呼ばれ，実用価値も高く古くから研究されてきた．一般に予測すべき時系列データの振舞いが非線形で複雑な場合が多く，時系列問題では近似関数の形式とパラメータをいかに導出するかがポイントになる．このような問題を一般にシステム同定問題といい，与えられた入出力データから未知のシステムの振舞いを予測する問題である．

2.2 GMDH

従来では，システム同定の技法はパラメータと関数の推定に基づいていた．しかしながらこれらのアプローチの多くは，訓練例やパラメータおよび制約の増加に伴って組み合わせ論的な暴発に陥りやすい．この種の手法の一つに，GMDH （Group Method of Data Handling）がある．これはシステム同定問題の解法に用いられる多変量解析手法であり，出力関数 y を近似するためのネットワークを構成する．ノードを表す関数は 2 次式となっている．

2.2 GMDH

GMDH の構成を図で表すと次のようになる

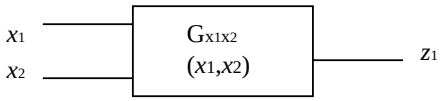

図 2.1 GMDH の構成

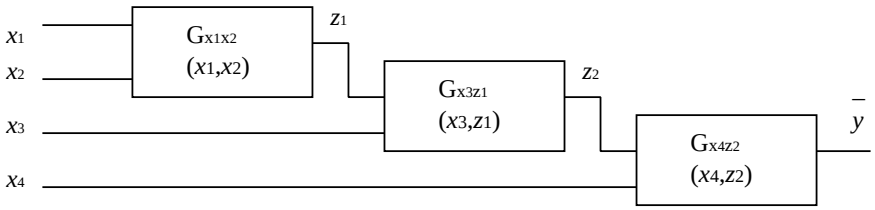

図 2.2 GMDH ネットワーク

GMDH の基本構成は図 2.1 となり，入力信号として 2 変数を受取り，出力信号として 1 変数を返す．図 2.2 では，ノードの構成が 3 つになっているが，必ず 3 つというわけではなく予測するものによってとりうる構成が変わってくる．

近似関数を導くための GMDH のアルゴリズムは次のようにして行われる．

Step 1 ：

入力変数を x_1, x_2, $\cdots$, x_m, 出力変数を y とする．$var = \{x_1, x_2, \cdots, x_m\}$ と初期化する．

Step 2 ：

var から 2 つの要素 z_1, z_2 を取り出す．多重回帰分析を用いて出力 y を z_1, z_2 により最小誤差で近似する式 $G(z_1, z_2)$ を構成する．

$z = G_{z_1,z_2}(z_1, z_2)$

Step 3 ：

z の近似の精度が，与えられた基準以上のとき y の完全形を z として終了する．

Step 4 ：

さもなくば $var = var \cup \{z\}$ として Step 2 に戻る．

ここで，2 次式 G はつぎのように表現する．

$$G_{z_1,z_2}(z_1, z_2) = a_0 + a_1 z_1 + a_2 z_2 + a_3 z_1 z_2 + a_4 z_1^2 + a_5 z_2^2 \tag{2.1}$$

また係数 $a_i (i = 0, ..., 5)$ は最小 2 乗法を用いて以下のように計算される．

取り出した観測位置 z_1, z_2 から参照してデータを取り出し，目的変数 y を参照して学習データが与えられる．出てきた 3 変数のデータを 1 組にして N 個のデータを用意する．

$$\begin{array}{ccc} z_{11} & z_{12} & y_1 \\ z_{21} & z_{22} & y_2 \\ & \cdots & \\ z_{n1} & z_{n2} & y_n \end{array} \tag{2.2}$$

2.2 のようにまとめることが出来る．次に，まとめたデータを用いて係数を求めるため N 個数分方程式を作成し，行列として形作る．方程式の形は最小 2 乗法で出てきた 2 次式を用いる．

$$X = \begin{pmatrix} 1 & z_{11} & z_{12} & z_{11}z_{12} & z_{11}^2 & z_{12}^2 \\ 1 & z_{21} & z_{22} & z_{21}z_{22} & z_{21}^2 & z_{22}^2 \\ 1 & z_{31} & z_{32} & z_{31}z_{32} & z_{31}^2 & z_{32}^2 \\ & & & \cdots & & \\ 1 & z_{n1} & z_{n2} & z_{n1}z_{n2} & z_{n1}^2 & z_{n2}^2 \end{pmatrix} \tag{2.3}$$

これらの行列を用いて下の式によって係数ベクトル a を求める．

$$a = (X'X)^{-1}X'y \tag{2.4}$$

一連の流れから算出された係数 a は式（2.1）で用いられる．

2.3　最小 2 乗法

x と y は一次式で $y = ax + b$ という関係がある．与えられた N 個のデータ (x_i, y_i) に最もよく合う a, b を求めると，以下のような式 $F(a, b)$ が考えられる．

$$F(a, b) = \sum_{i=1}^{N} (ax_i + b - y_i)^2$$

データに誤差がなく，データが厳密に $y = ax + b$ にのっていれば，$F(a, b)$ の値が 0 となる a, b が存在する．しかしデータが誤差を含む場合，必ずしも $F(a, b)$ が 0 にはならない．そこで $F(a, b)$ の値が最も小さくなるように a と b の値を決めることで近似的にデータの当てはめを行う．このように，$f(x_i, y_i)$ の 2 乗和が最小になるように方程式の係数を決める方法が最小 2 乗法である．

先程の式を計算すると

$$\begin{aligned} F(a, b) &= \sum_{i=1}^{N} (ax_i + b - y_i)^2 \\ &= \sum_{i=1}^{N} (a^2 x_i^2 + 2abx_i - 2ax_i y_i + y_i^2 - 2by_i + b^2) \\ &= Aa^2 + Bb^2 + Cab + Da + Eb + F \end{aligned}$$

簡略化するために係数に変換する A, B, C, D, E, F は x_i, y_i の式になるがこれらを定数と考えると，$F(a, b)$ を最小にするような a, b は以下の式を満たしている．

$$\frac{\partial F}{\partial a} = \frac{\partial F}{\partial b} = 0$$

したがって上記の式を計算して，連立方程式を解けば a, b を $A, B, \cdots, F$ で表すことが出来る．$A, B, \cdots, F$ は，データ群から計算されるので a, b が算出できる．この a, b が求めるべき方程式 $y = ax + b$ の係数である．

2.4　単体実験

実験の目的として，GMDH の評価を行う．データを読み込ませたとき，どの程度近似することができるかを実験する．単体での評価として，sin 関数を近似させる．入力データと

して sin 関数を 15° 毎にプロットした値を用いる．

実験手順は下記のように行う．

Step 1 ：

観測位置 z_1, z_2 を選択する．目的変数 y は固定.

Step 2 ：

観測位置と目的変数の値を参照して，データを取り出す.

Step 3 ：

取り出したデータから行列（2.3）の形にする．

$a = (X'X)^{-1}X'y$ から係数 a を算出する．

Step 4 ：

式（2.1）に係数を入れる．計算結果を入れる配列（仮に v とする）を用意し，

目標変数の大きさ分配列 v にデータを入れる．

Step 5 ：

$v[i+y] = G(v[i+z_1], v[i+z_2])$

Step 6 ：

i が指定回数を満たすまで，Step 5 を繰り返す.

Step 7 ：

v を出力する．

2.4.1　結果

観測位置 $z_1 = 0$〜$9, z_2 = 0$〜9 すべて実験した結果，出力結果は “近似する”，“途中まで近似する”，“近似しない”，“グラフとして出力できない”，の 4 パターンに分かれた.

4 パターンの内 3 つを以下に図 2.3，図 2.4，図 2.5 に示す.

図示すると，実際に図 2.3 が近似する例である．しかし，残りの図 2.4 は部分的に近似しているが結果として，値が収束する．図 2.5 は途中で近似できない．また出力できない例が

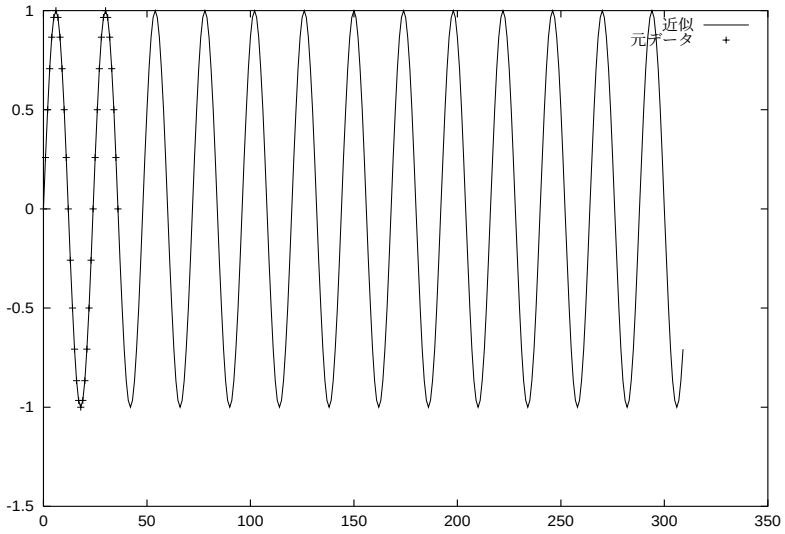

図 2.3　近似したグラフ

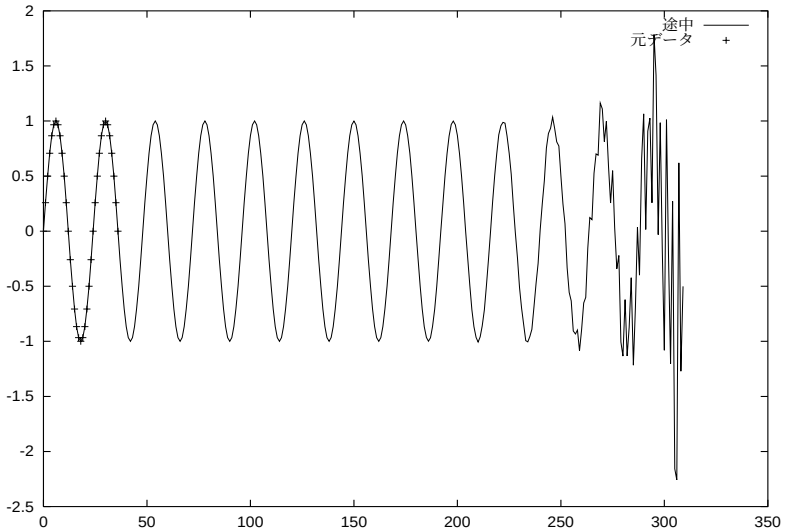

図 2.4　途中まで近似するグラフ

あり，この場合計算過程で値が極端に大きくなるため，$-\inf$ や NaN が出てきた．

原因として考えられるのは，観測位置の選択や実験手順 **Step**4 で与えられた学習データと目標データは同じものを使い，計算式 **Step**5 で出た予測値を目標データとして付け加えるので，繰り返すたびに徐々に算出される予測値が狂っていたと考えられる．

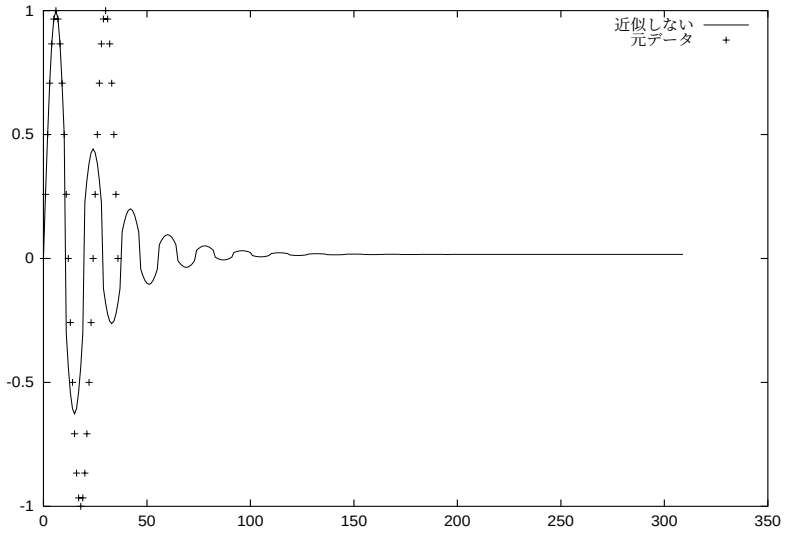

図 2.5　近似しないグラフ

2.5　単体実験 2

先の実験では，算出した予測値は目標データに付け加えた．ここでは算出した予測値を付け加えない場合を実験する．実験手順は，先の実験手順とほとんど変わらず，学習データも同じ sin 関数を 15° 毎にプロットした値を用いる．

2.5.1　実験結果

出力結果は，次の 2 例になる．観測位置が同じでなければグラフは図 2.6 のようになる．前の実験に比べて，グラフは元データに近似しない．

2.6　例外値

今までは，整ったデータを扱って近似関数の予測値を求めて来た．しかし実際に与えられるデータは，例外値を含むようなものが多い．実験として例外値を含むデータを近似したときの出力結果を示す．

重回帰分析（最小 2 乗法）では，モデルとデータとの誤差が平均 0 の正規分布に従う場合には，推定されたモデルは最適なものとなる．しかしデータに例外値が含まれているような

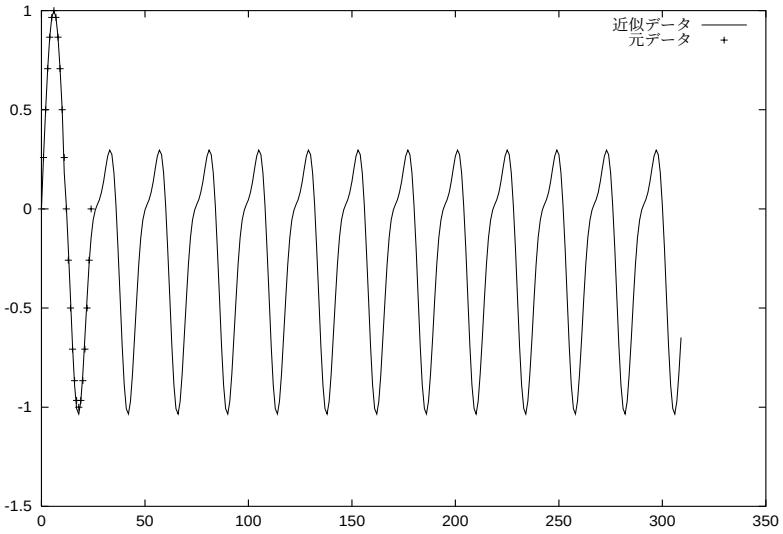

図 2.6　予測値を付け加えないグラフ

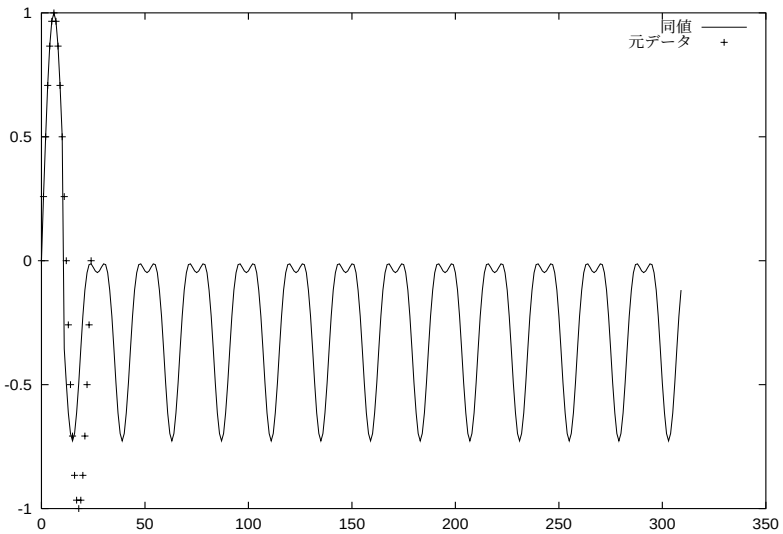

図 2.7　予測値を付け加えないグラフ（同値の場合）

場合には，推定結果は例外の影響を大きく受けて，必ずしもよいモデルが推定できるとは限らない．

次に例外値の前処理に付いて示す．例外値の前処理として挙げられるのは，平均値をとって確認を行うことを今回は行う．

2.7　単体実験 3

実験手順は，実験手順 2 とほとんど変わらず，学習データは例外値を含む sin 関数を 15° 毎にプロットした値を用いる．最初 2 つは例外値を含んだ状態での実験．後 2 つは例外値を除去する前処理を行ってから計算を行う．例外値を除去する条件として

$$|average - current| > 1 \ \& \ average < current$$

が満たされた場合平均化を行う．

ここで $average$ は計算によって算出された平均値，$current$ は平均値の場所の元データ

2.7.1　実験結果

例外値が入った場合，出力されるグラフは，図 2.8 のように目標データに近似しないことが多い．しかし観測位置が同じとき，振幅の小さい sin 形状図 2.9 になる．

例外値を除去した場合，出力結果はほぼ sin 関数に近似した．（図 2.10）また同値の場合には多少のずれがあるものの近似している．（図 2.11）

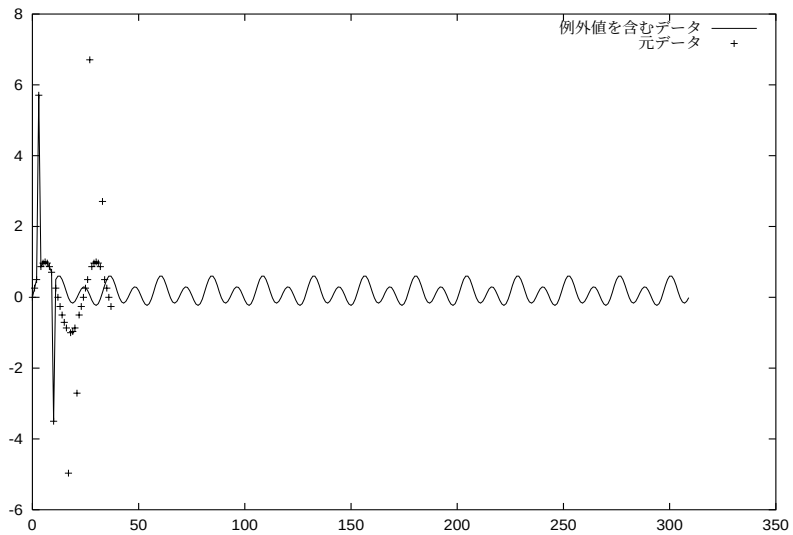

図 2.8　例外値を含むグラフ

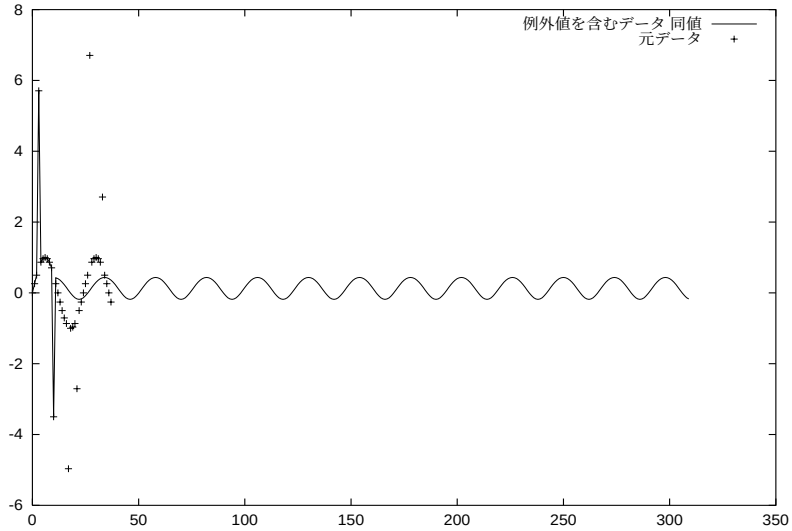

図 2.9　例外値を含むグラフ（同値の場合）

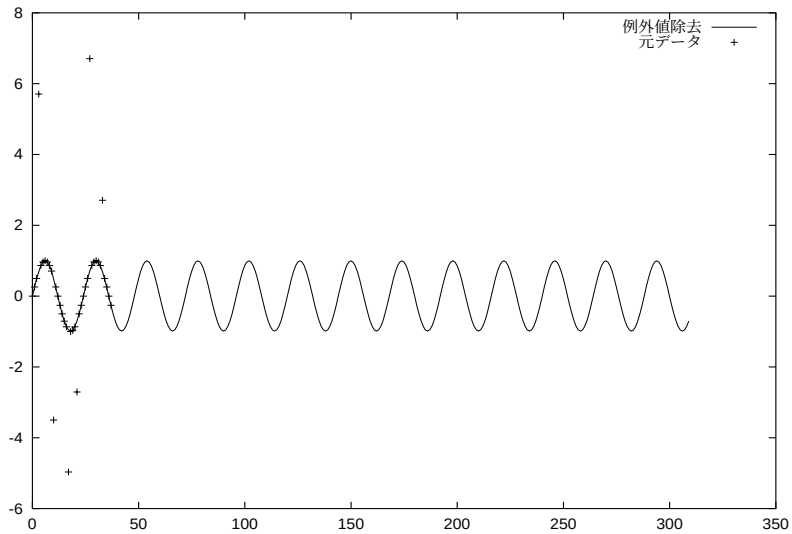

図 2.10　例外値を除去グラフ

2.8　考察

1 つのアルゴリズムについて様々な実験を行ったが，1 度の試行で正確な近似はほとんど得られなかった．1 段のみで近似することには限界がある．また，例外値を除去するために平均を用いているが，例外値が連続で出てくる場合には対応していない．そのため別の方法を考える必要がある．

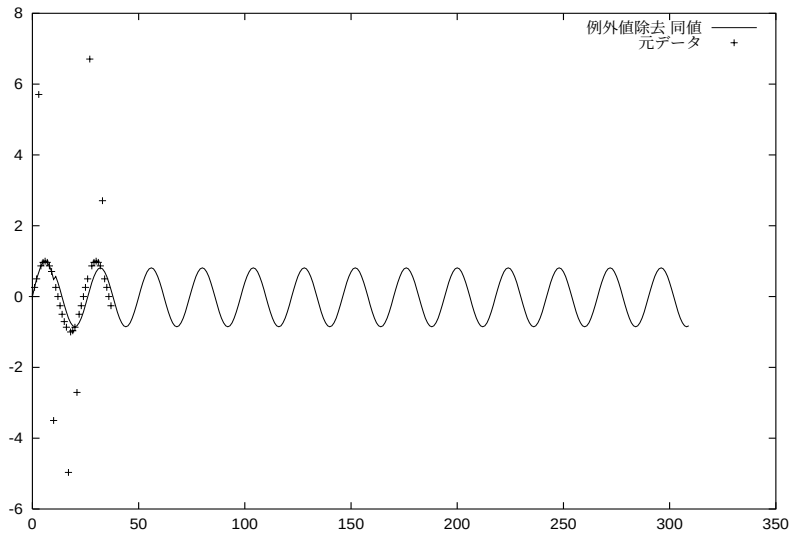

図 2.11　例外値を除去したグラフ（同値の場合）

第 3 章

遺伝的プログラミング

3.1　遺伝的アルゴリズム

遺伝的アルゴリズム（Genetic Algorithm, 以下 GA）は John Holland の論文において導入された手法で生物進化（選択淘汰・突然変異）の原理に着想を得たアルゴリズムであり，確率的探索・学習・最適化の一手法と考えることが出来る．

3.1.1　処理手順

GA の処理手順は，次のようになる．

最初に初期集団の生成を行い．各々の個体に対して適応度の評価を行う．適応度の計算は，解こうとする問題ごとに異なる．個体の適応度が決定されたら，それを基に選択交配を行う．基本的に，適応度のより高い個体がより多くの子孫を残す仕組みとなる．選択交配を行う個体対が決定されたら，染色体の交叉を行う．交差の方法も色々提案されているが代表的なものについては後に記す．突然変異はある一定の確率で染色体の一部の値を買える操作である．これら一連の操作が終了すると，新しい世代の集団個体が作られる．そして，新しい集団に対して，適応度計算を行い選択・交配・突然変異を行い新たな世代を作成する．

3.1.2　選択交配

個体群のすべての個体について適合度を求める．この適合度に基づき次の世代に残す個体を決定する．適合度は，問題における解の評価の高さで与えられた問題によって適合度の求

め方は変わる．よい解ほど高い適合度が得られるように評価関数を設定する．また選択の方法について，さまざまな手法がある．

3.1.3　交叉

交叉は二つの親の染色体を組み替えて子の染色体を作る操作である．交叉する位置を 1 点の場合は単純交叉と呼ぶ．また 2 点の場合は，2 点交叉と呼ぶ．交叉時にマスクをかけてどちらの親の遺伝子を受け継ぐかを決定する方法は一様交叉と呼ぶ．

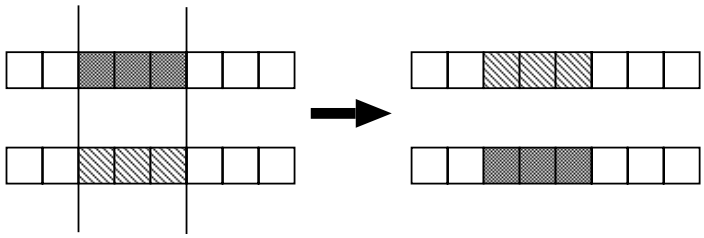

図 3.1　2 点交叉の例

3.1.4　突然変異

遺伝子を一定の確率で変化させる操作で，遺伝子が局所的な最適解に落ち着いてしまうことを防ぎ，より広い範囲で最適な解を探すことが目的である．突然変異がない場合，初期の遺伝子の組み合わせ以外探索することは出来ない．そのため求められる解の質に限界がでてくる．

3.2　遺伝的プログラミング

遺伝的プログラミング（Genetic Programming, 以下 GP）とは，GA が木構造を扱えるように拡張した手法である．一般的な GA が一次元配列を扱うのに対して，GP は木構造で個体を表現することにより，決定木・グラフ等の知識表現の獲得や，関数・プログラムの自動生成など，階層的な表現能力を要する問題を直接的に扱えるという利点を持つ．GP では

木と呼ばれる構造表現を扱う．木とはサイクルを持たないグラフのことであり，図 3.2 のような構造を持つ．

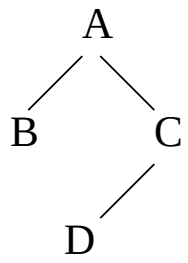

図 3.2　木構造

例えば，次の数式 $\sin(x+y)-\cos(x-y)$ は図 3.3 で表すことが出来る．図 3.3 では下の方にある木の葉に相当する x や y を「終端記号」，枝別れの部分にある $-, +, \sin, \cos$ を「非終端記号」と呼ぶ．関数を遺伝情報とする GP では，終端記号は定数または変数，非終端記号には関数名がくる．

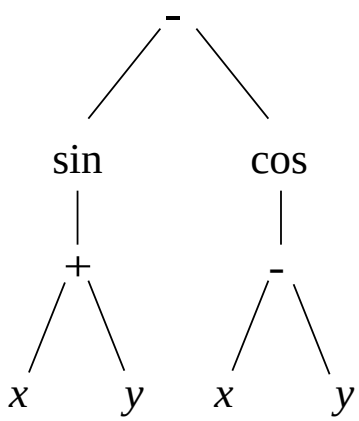

図 3.3　木構造で表した数式

GP では GA と同様に木構造に対して，交叉や突然変異・逆位などの遺伝子操作を用いることによってプログラム（木構造）を進化させる．

3.2.1　交叉

GP では２つの木構造から，それぞれの部分木を組み替えて新しい木構造を作る．一般に交叉する遺伝子の部分木の位置はランダムによる．

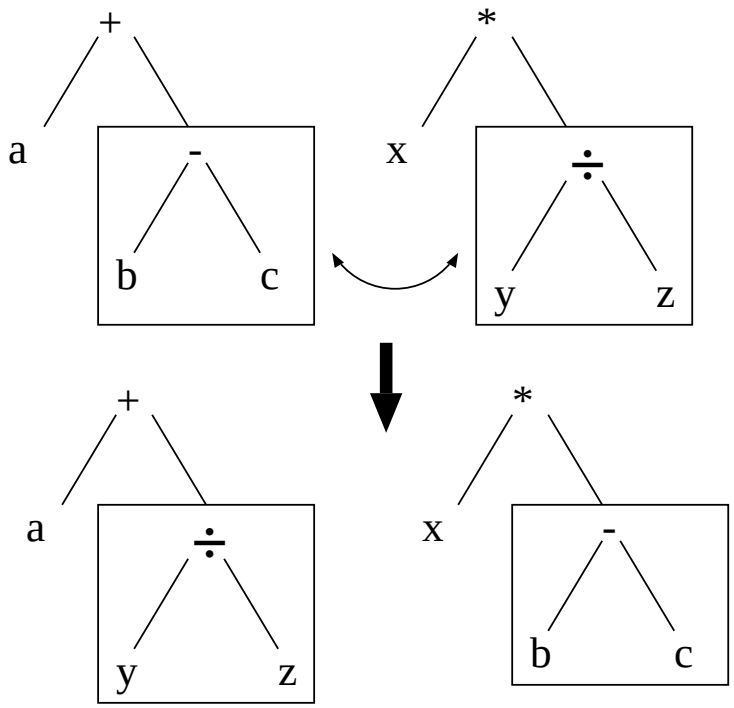

図 3.4　交叉

3.2.2　突然変異

突然変異は，遺伝子をある一定の確率で変化させる操作であり，終端記号を変化させ新しい木構造を作る．突然変異にも種類があり以下に示す．

1. 終端記号から非終端記号への変異

   この場合木構造は新しい部分木の生成を伴う
2. 終端記号から終端記号への変異

   ノードラベルをつけ替える
3. 非終端記号から終端記号への変異

   部分木の削除を伴う
4. 非終端記号から非終端記号への変異
   - 新しい非終端記号と古い非終端記号の子の数が同じ場合

     ノードラベルの付け替えを行う
   - 子の数が異なる場合

     部分木の生成・削除を伴う

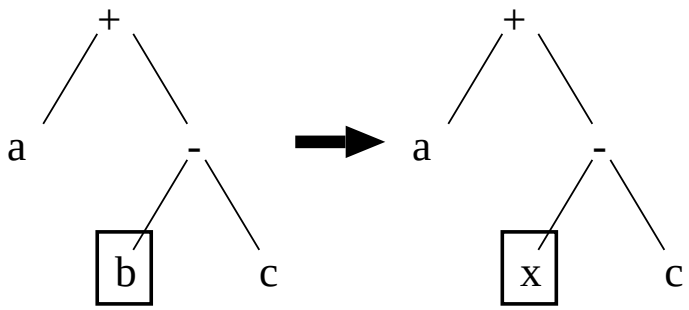

図 3.5　突然変異

3.2.3　逆位

逆位は，1 つの遺伝子の兄弟同士を交換させる操作である．

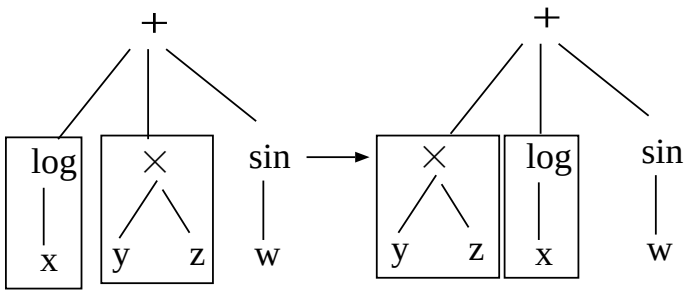

図 3.6　逆位

3.2.4　遺伝的プログラミングの流れ

GP の処理手順は，GA と同等の手順を行う．最初はランダムに初期集団を作り，そして初期集団それぞれの適応度を求める．つぎに交叉・突然変異などの遺伝的操作を行い，新たな集団を生成する．終了条件（最適解）を満たすまで，適応度計算・遺伝的操作を繰り返し行う．

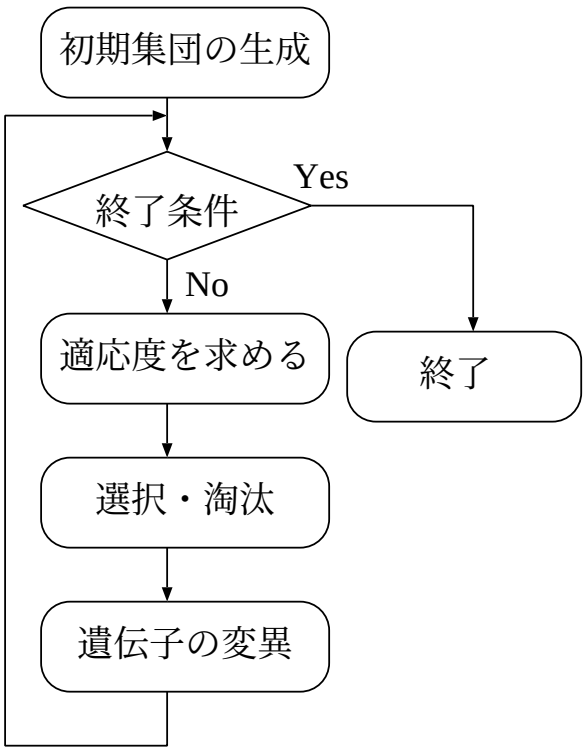

図 3.7　処理手順

3.3　遺伝的プログラミングの利点・問題点

遺伝的プログラミングの利点

1. 構造の視覚化

   従来の遺伝的アルゴリズムでは，遺伝子の構造が 1 次配列状になるため，どの部分に対して遺伝的操作が行われたかが解りがたい．しかし遺伝的プログラミングでは，木構造となっているため．どの部分に対して遺伝的操作が行われたかが視覚化しやすい．

2. 自己関数定義

   遺伝的プログラミングでは，関数を自身で定義して効率的に利用できる．自己定義の利点として，探索の過程で木構造が莫大になり探索効率が劣化するのを防ぐという目的がある．

遺伝的プログラミングの問題点

1. 計算時間

   遺伝的プログラミングでは，遺伝的操作を行うことによって自身の構造を書き換えることが出来る．最適解を見つけられない場合，木構造が深く，そして複雑になるため 1 つの遺伝子にかかる計算が大きくなる．

第 4 章

遺伝的プログラミングを用いた時系列処理

4.1 木の構造

遺伝的プログラミングを時系列処理問題に適用させる．GMDH は図 2.1 のような構造しているため．そのまま木構造として見ることができ，そして GMDH の構成を一つの遺伝子として扱う．

GMDH は第 2 章で 2 入力 1 出力であることを示した．木構造に入る情報として，木の端子には 2 つの観測位置が入り，非終端コードには，近似するための 2 次式が入る．そして出力されるのは，選択された観測位置内の近似結果である．

4.2 適応度関数

木構造に対する適応度は，近似されたデータと元データの最小 2 乗誤差から求める．ここで S_N^2 は最小 2 乗誤差，N はデータの数，$\overline{y_i}$ は近似されたデータ，y_i は元データとなる．

$$S_N^2 = \frac{1}{N}\sum_{i=1}^{N} |\overline{y_i} - y_i|^2$$

適応度は最小 2 乗誤差の最小値とする．

4.3　実験手順

GP を用いた時系列処理の実験手順を提案する.

他のシステム同定問題に用いられる非終端記号 $\{+, -, *, /\}$ は用いずに GMDH のみを非終端記号とする.

実験手順は以下の 2 点を除いて従来の GP と同一である.

1. 木が生成・変更されるたびに中間ノードが保持する係数を計算する
2. 最小 2 乗誤差による適合度計算を行う

第 5 章

結論

本論文では，遺伝的プログラミングまで最終的にたどり着くことが出来なかった．しかしGMDH アルゴリズムにおいては近似精度が測れた．今回の実験の中で，一度の試行で完全に近似するグラフは得られなかったが，2 段，3 段と組み合わせていく内に完全に近似するグラフが得られると思われる．今回例外値を除去するために平均値を用いたが，この方法は例外値が連続して現れると使うことが出来ない．よって，全体の偏差を見て例外値を消す工夫や連続して現れる場合についても処理できるようにするアルゴリズムを開発することが今後の課題と言える．

謝辞

本論文は，著者が 1999 年 7 月から 2001 年 3 月までの高知工科大学工学部情報システム工学科在学中に，同学科坂本研究室において行った研究の成果を記したものである．

はじめに，関係者一同に御迷惑をかけたことを深くお詫びします．特に就職や研究で始終心配をお掛けした坂本明雄教授には、感謝してもしたり無いぐらいです．また本研究の指導教員になってくれた Ruck Thawonmas 助教授には，心配ばかりお掛けして済みませんでした．

忘年会や新年会によく一升瓶を持って来ていた院生の橋本学先輩，お体に気を付けて飲んで下さい．また，始終こちらの研究の進み具合を心配してくれた井上祐介君，有賀洋介君，登伸一君，横谷将樹君ありがとうございます．また麻雀でもしましょう．しかし昨年の忘年会，鍋の具に “ニラ” を何故買ってきたのか今でも疑問に思っています．

院生の久保真理子さん，お弁当おいしく頂きました．ありがとうございます．永遠のオリッチこと折橋祐一君，大学で 4 ヵ月ほど見かけませんでしたが一体何をしていたのでしょうか？烏の行水みたいに入ったのか入らなかった解らない程研究室にいない山崎聖太郎君

同研究室現３年の “なぜかタメ口” の亀本学君，“バレーボールと歴史と焼酎にこだわる” 一恩邦至君，“なにかとアツく礼儀正しい” 木村大樹君，“音ゲー” 栃木隆道君，“なにかとやり手” の志摩浩君らには，研究のことや勉強関係のアドバイスはあまり出来ませんでしたが，こちらの馬鹿話に付き合ってくれて大分落ち着くことができ感謝しています．

最後に，いつも突然押しかけても嫌な顔せずこちらの愚痴を聞いてくれた友人の伊東禅君に感謝します．

参考文献

[1] Iba,H., deGaris,H. ans Sato,T.: “System Identification Approach to Genetic Programming”, ETL-TR-2, also Proc. IEEE World Congress on Computational Intelligence(WCCI94), 1994.

[2] 伊庭斉志, “遺伝的アルゴリズムの基礎 –GA の謎を解く– ”, オーム社, 1994.

[3] Ivakhnenko, A.G.: “Polynominal Theory of Complex Systems”, IEEE Tr.SMC, vol.SMC-1, no.4, 1971.

[4] B.W. カーニハン,D.M. リッチー, “プログラミング言語 C”, 共立出版株式会社, 1989.