社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－MPS－61（6）
2006／9／14

# スキーマ理論に基づいた遺伝的プログラミング計算の提案

荒関 仁志*

**概 要**

遺伝的プログラミングは様々な分野に応用されている。遺伝的プログラミングの進化メカニズムは、積木仮説に基づくスキーマ理論によって説明されている。しかし、この積木構造を実際の進化計算に直接利用した研究は少ない。

本論文では、スキーマ理論で重要な役割を果たしていると考えられている積木構造を直接遺伝的プログラミング計算に取り入れた、新しい計算モデルを提案する。このモデルでは、積木構造を集団から探索し、次世代に安全に継承するメカニズムを持っている。実験の結果、通常の遺伝的プログラミングに比べて明らかな進化速度の改善が見られ、実際の計算においても積木構造が重要な役割を果たしていることが分かったので報告する。

# Evolutional Calculation based on Schema Theory for Genetic Programming

Hitoshi Araseki*

**Abstract**

Genetic Programming is used for problem solving in various fields. It is well known that the building blocks of the Schema theory have played an important role in evolutional calculation. Despite considerable research oriented to the study of the Schema theory for Genetic Programming, little is known about the real calculations necessary to handle the building blocks directly.

This paper describes an examination of the role of the building blocks of Genetic Programming used in real evolutional calculations. In this paper a new calculation approach for Genetic Programming is presented. This new approach has specific mechanisms which can find the building blocks in population and accurately pass these building blocks on to the next generation. We found experimentally that the building blocks play an important role in the real calculation process in Genetic Programming.

## 1　はじめに

GP における進化過程の理論的な説明は、スキーマ理論によってなされている。スキーマ理論によると進化のメカニズムは、集団内に「積木と呼ばれる有効な構造（以下積木構造）」が生成され、その積木構造が次世代に継承され、それらを組合わせることでより適合度の高い個体を生成することができるというものである。しかし、通常の GP[1]（以下 SGP) における実際の進化計算では、個体の評価値を基準に選択淘汰を行うため、評価値に対応しない積木構造を直接扱うことはできない。

本論文では、GP の進化過程における「効率的な積木構造の探索と継承」を目的として、従来の遺伝的プログラミングの進化計算に新しいメカニズムを加えることを提案する。このモデルでは、積木構造を集団から探索し、次世代に安全に継承するメカニズムを持っている。実験の結果、SGP に比べ進化能力に関して明らかな改善が見られたので報告する。

## 2　研究の背景

現在までに、GP に関する様々なタイプのスキーマ理論が提案されている [5]。そのどれもが積木仮説に基づいたスキーマ（有効構造）を次世代へ継承する場合の確率的取り扱いを行うものである。これらは進化過程のダイナミクスを表現することが可能

*日本大学 (Nihon University)

な理論となっている。この理論では、進化過程において積木構造の有効性を強く示唆するものである。一方、これらのスキーマ理論を直接実際の進化計算に取り入れることは難しい。進化計算における個体の選択淘汰では、個体の評価値を基準としているため、スキマー理論が進化計算に有効としている積木（スキーマ）構造を、直接個体の評価値に反映できないからである。したがって、スキーマ理論を実際の計算に取り入れる場合には、積木構造を直接個体の評価値に取り入れるようなメカニズムが必要となる。もし、積木構造と評価値との間に関係付けが可能であれば、進化計算の改善が期待できる。

進化計算を行う上では、この積木構造をどのように集団内に生成し探索するかが重要な問題となる。現在までに、積木構造を明示的に生成・探索するいくつかの試みがなされている。その１つが Koza による ADFs(Automatically Defined Functions)[2] である。これは SGP の進化過程とは別に、積木構造 (関数) を生成する過程を用意し、その積木構造を各個体が利用することで進化能力の改善を計る試みである。実験の結果、進化能力の大きな改善が見られることが分かった。しかし、この ADFs を生成するためには、予め積木構造の構造を決めておかねばならず、ADFs を利用する場合には、与えられた問題に対する知識が必要となるため汎用的な取り扱いとは言い難い。

この ADFs の考え方を、より汎用的な取り扱いにするために、Angeline と Pollack により GliB(Geneic Library Builder) が提案された [3]。GLiB は、個体内からランダムに切り出された木構造を取り出し、切り出した木構造を積木構造と仮定し、その木構造を再利用のためのライブラリに登録する。各個体は進化の過程で、このライブラリ内の積木構造を利用することで進化能力の向上を目指したものである。しかし、実験の結果、ランダムに取り出された部分木には、有効な積木構造は少なく、進化能力に関しては、SGP に比べて優位な差を見つけることはできなかった。

これらの研究から、評価値の高い個体中には、積木構造が存在する可能性が高いことが理解できる。

# 3　GPoS モデルの概要

GP においては、個体（プログラム）は木構造で表現される。個体には複数の部分木が内在しており、固体の評価値を計算する場合、各部分木の総和として評価値が決まる。ただし、内在する部分木が高い評価値を持っていた場合でも、内在する部分木の総和としての個体の評価値においては、評価値の高い部分木よりも、低くなる場合が多い。これは、同じ構造の部分木を持つ個体でも、個体のどの枝に部分木が位置するかによって、個体の評価値が変わってくるからである。SGP には、個体に有効な部分木が内在していても、それを直接取り出し、次世代に継承するメカニズムはない。また、進化速度の向上のために様々な方法が試みられているが [6] そのどれもが積木構造を直接的に探索し継承するメカニズムを持たない。

進化能力を高めるためには、積木構造を集団から探索し、遺伝的演算から保護し、かつ次世代に安全に継承することができる進化計算モデルを提案する必要がある。本研究では、この有効部分木を集団内から探索し保護・継承するメカニズムを持つ GPoS (Genetic Programming based On Schema theory) モデルを提案する。

個体の評価値計算は、部分木の総和として計算される。本論文ではこのことに着目し、評価値が最も高い部分木を、その個体の有効部分木（= 積木構造）と仮定する。今、世代 $t$ での $k$ 番目の個体 $x^k$ を考えると、この個体 $x^k$ の評価値 $f(t)^k$ は以下のように定義できる（ここでは評価値が大きいものが評価値が高いとする）。

$$f(t)^k = max\{f(t)^k_0, f(t)^k_1, \cdots, f(t)^k_n\}$$

ここで $f(t)^k_i, i = 1, \cdots, n$ は、個体 $x^k$ 内での $n$ 個の部分木を表し、この場合 $i$ 番目の部分木の評価値を表している。

個体に内在する有効部分木が見つかった後、それを保護し次世代に継承することを考える。有効部分木を破壊するものは遺伝子演算であるが、その多くは交叉演算によるものである [4]。ここでは、有効部分木を保護する目的で、有効部分木は遺伝子演算（交叉演算と突然変異演算）の作用を受けないものとする。これによって有効構造が次世代に確実に継承されることとなる。個体間の交叉演算は以下の 3 通りが考えられる。

1. 交叉演算対象の 2 個体が共に有効部分木を内在する場合、有効部分木を交換し、新たに 2 個体を生成する。

2. 交叉演算対象の 1 個体だけが有効部分木を内在する場合、有効部分木を一方の個体に受け渡し、新たに 1 個体を生成する。有効部分木を受け渡した個体は、もう一方の個体と通常の交叉演算を行い、新たに 1 個体を生成する。

3. 交叉演算対象の 2 個体が共に有効部分木を持たない場合には、通常の交叉演算を行い、新たに 2 個体を生成する。

突然変異演算では、突然変異対象個体に有効部分木が内在する場合は、有効部分木を避けて突然変異演算を行う。これにより、突然変異演算による有効部分木の破壊を防ぐことができ、次世代に安全に継承させることができる。

# 4 進化実験

実験は、Even-4-parity 問題と 11-Multiplexer 問題の 2 種類のバイナリ問題について行った。また実験で使用した各種パラメータは、表 4 の通りである。

| パラメータ名 | パラメータ値 |
|---|---|
| 集団数 | 4000 |
| 突然変異率 | 10% |
| 選択則 | MGG(Minimal Generation Gap) |

選択則には MGG を使う [7] 。この MGG は進化演算において多様性を維持する能力が高く、進化計算の高速化が期待できることが知られている。

本論文では評価関数として、以下のような Adjusted Fitness[1] を使用する。評価回数 $t$（世代数に対応）の $k$ 番目の個体の $j$ 番目の部分木の評価値は以下のようになる。

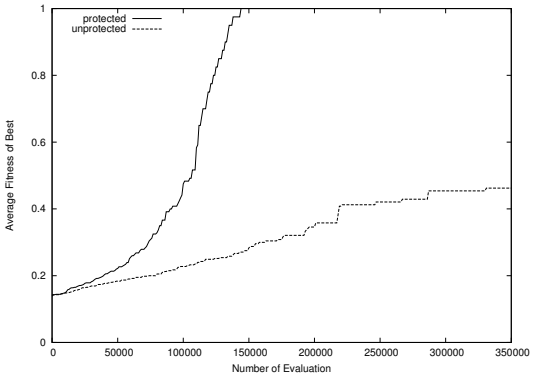
$$f(t)_j^k = \frac{1}{1 + \Sigma_i^m \mid r_{ij}^k - r_{i0} \mid}$$

ここで、$m$ は教師信号の数、$r_{ij}^k$ は $j$ 番目の部分木の $i$ 番目の入力値に対する部分木の値である。また $r_{i0}$ は $i$ 番目の入力値に対する教師信号となる。実験はすべて 20 回行いその平均を実験結果とした。

ここでは、GPoS モデルに基づいた評価値計算と SGP との比較を行う。計算の比較に使った SGP は、GPoS モデルでの有効部分木探索および保護機能をはずしたもので計算した。したがって、koza などの一般的な GP とは多少異なっているので、実験結果の比較には注意が必要である。評価値計算では、GPoS モデルを使ったものを *protected* で表示し、SGP の結果を *unprotected* と表示している。縦軸 (Average Fitness of Best) は最良個体の評価値の平均 (20 回）を示す。横軸 (Number of Evaluation) は評価回数を示す。集団数が 4,000 の場合、評価回数が 4,000 で、エリート戦略の選択則における 1 世代に対応する。

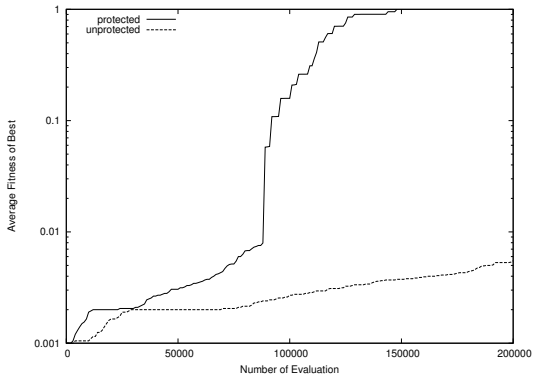
## 4.1 Even-4-Parity 問題

図 4.1 に、最適個体の評価値を示す。実線は GPoS モデル (protected) で 144,000 回の計算回数ですべて正解となっている。SGP(unprotected) の計算を点線で示した。本実験では 350,000 回までの計算では正解を得られなかった。




## 4.2 11-Multiplexer 問題

図 4.2 に、最適個体の評価値を示す。実線は GPoS モデル (protected) で 148,000 回の計算回数ですべて正解となっている。ここでの縦軸は *log* スケールになっている。SGP(unprotected) の計算を点線で示した。本実験では 200,000 回までの計算では正解を得られなかった。




## 5　実験の結果と考察

Even-4-Parity 問題に関して、koza[1] の GP 計算の結果では、評価回数 384,000 回ですべての実験で正解を得ている。残念ながら本実験（SGP) では 350,000 回の評価回数では正解は得られなかった。これは集団数と選択則の違いによるものと考えられる。一方、GPoS モデルでは 144,000 回ですべて正解を得ているので、koza の GP に比べて進化速度の改善が見られる。また、Koza の ADFs[2] の計算でも評価数 176,000 回ですべて正解を得ているが、それと比べても GPoS モデルが優位であることがわかる。また、11-Multiplexer 問題につても評価回数 148,000 回ですべて正解を得ている。この計算でも SGP(unprotect) と比較して優位な結果で得られた。

これらの計算結果から、SGP と比較して GPoS モデルを使った GP 計算には、進化速度の向上が顕著に見られる。これは、バイナリ問題の解が、多くの繰り返し構造を持っているため、積木構造の効果が有効に働いているものと考えられる。この結果、バイナリ問題に関しては、GPoS モデルでの有効部分木の取り扱いが効果的であることを示すものであり、積木仮説に基づくスキーマ理論が実際の進化計算においても有効な理論であること示しているものと考えられる。

## 6　まとめ

遺伝的プログラミングの改善に有効なスキーマ理論に基づいた GPoS モデルを提案した。このモデルを用いた実験の結果、通常の遺伝的プログラミングに比較的単純な取り扱いを加えることで（有効部分木の探索と保護・継承メカニズム)、進化計算の改善が可能となることが分かった。このことは、少なくてもバイナリ問題に関しては GPoS モデルが有効であり、スキーマ理論が実際の進化計算においても有効な理論であることを示している。また、従来の GP では、確率的な取り扱いでしか有効構造を探索し継承する方法がなかったが、GPoS モデルを使うことで、評価値と有効構造との間に直接的な関係をつけることができる可能性が得られた。今後は、バイナリ以外の問題（人工蟻問題など）でも評価する予定である。

## 参考文献

[1] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection.*, MIT Press, 1992

[2] J. Koza, *Genetic Programming II: Automatic Discovery of Reusable Programs*, MIT Press, 1994

[3] P. J. Angeline, *Genetic Programming and Emergent Intelligence*, Advances in Genetic Programming, MIT press, 1994

[4] P. Nordin, F. Francone and W. Banzhaf, *Explicitly Defined Intron and Destructive Crossover in Genetic Programming*, Advances in Genetic Programming II, MIT press, 1996

[5] R.Poli, *General Schema Theory for Genetic Programming with Subtree-Swapping Crossover*, European Conference on Genetic Programming 2001, pp.143-159(2001)

[6] M.D.Platel, M.Clergue and P.Collard, *Maximum Homologous Crossover for Linear Gentic Programing*, European Conference on Genetic Programming 2003, pp.194-203(2003)

[7] 佐藤 浩, 小野 功, 小林 重信, **遺伝的アルゴリズムにおける世代交代モデルの提案と評価**, 人工知能学会, Vol.12, No.5,pp.734-744(1997)