

A-001

## グラフ構造のプログラム自動生成手法のための子個体生成方法の提案

## A Generation Method of Individuals for Graph Structured Program Evolution

石堂 眞大<sup>†</sup> 白川 真一<sup>†</sup> 長尾 智晴<sup>†</sup>

Masahiro Ishido Shinichi Shirakawa Tomoharu Nagao

## 1. はじめに

近年、コンピュータプログラムを自動生成する自動プログラミングに関する研究が活発に行われている。自動プログラミングは、人がソースコードを記述することなく、入出力例を与えるだけでコンピュータが全自動で目的の処理を実現するアルゴリズムを生成する方法である。

自動プログラミングの代表的な手法として、木構造のプログラムを自動生成する遺伝的プログラミング (Genetic Programming; GP) [1] が挙げられる。また、グラフ構造のプログラムを自動生成する手法として Graph Structured Program Evolution (GRAPE) [2] が提案されている。GRAPEではグラフ構造を用いることによって、木構造では表現することが困難なループなどを含む構造が表現可能である。しかし、従来の GRAPE にはプログラムの自動生成が難しい問題領域が存在する。また、様々な問題で自動生成を行うためには、より効率良くプログラムを進化させる必要がある。

本研究では、従来の GRAPE におけるプログラムの進化について解析を行い、その結果からプログラムを効率良く生成するための子個体生成方法を提案する。そして、いくつかの問題に対してプログラム自動生成実験を行い提案手法の有効性を示す。

## 2. Graph Structured Program Evolution (GRAPE)

GRAPE はグラフ構造のプログラムを自動生成する手法である。図1に GRAPE の構造例を示す。GRAPEではグラフ構造のプログラムを一次元の整数列で表される遺伝子型にマッピングし、遺伝子型に対して GA で用いられるような比較的単純な遺伝子操作を行うことでプログラムを進化させる。

また、生成されるプログラムは有向グラフとデータセットから構成されるため、データセットに様々なデータ型を用意することで複数のデータ型を1つのプログラム中で扱うことができる。データセットは有向グラフ中を流れ、各ノードで演算などの処理が行われる。プログラムはスタートノードから開始し、いくつかの処理・判定ノードを経て出力ノードで実行結果を出力して終了する。

## 3. プログラムの生成における交叉の影響

GRAPE では、子個体を生成する際に、遺伝子操作として一様交叉と突然変異を使用している。一般的に、交叉では両親の優れた部分が組み合わせられることで、より優れた個体が生成され、個体集団の進化が促進される。

そこで GRAPE におけるプログラムの進化に対する交叉の有効性に関して検証を行う。ここでは、子個体生成時に交叉を行う確率である交叉率を変化させて階乗を求め

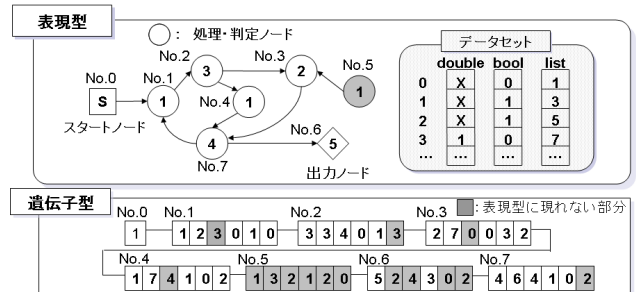


図1 GRAPE の構造例

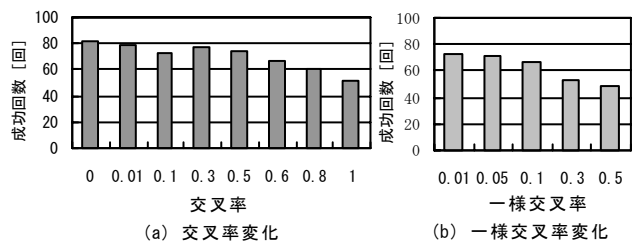


図2 成功回数の変化

プログラムの自動生成実験を行い、交叉率ごとの成功回数を比較する。また、交叉時に遺伝子を入れ替える割合である一様交叉率を変化させることによって、2つの親個体から遺伝子を受け継ぐことが良いか、片方の個体の形質を多く受け継ぐ方が良いかということを考察する。

図2(a)に交叉率による成功回数の変化、(b)に一様交叉率による成功回数の変化を示す。(a)から、交叉率が小さいほど成功回数が増加していることがわかる。また、(b)では一様交叉率が小さいほど成功回数が増加している。

ここで、交叉を行うことで成功回数が減少している理由について考察する。GRAPE はグラフ構造を一次元の遺伝子型で表現しているため、本実験で用いているような一般的な交叉方法を用いた場合、グラフ上の有効な構造を保持した交叉を行わせることができなかつたと考えられる。このため、交叉よりも低い確率で遺伝子を変化させる突然変異のほうが構造を保持したまま進化することができ、成功回数が多くなつたと考えられる。また、遺伝子を多く入れ替えるほど構造が破壊されやすくなり、進化が進まなくなつたと考えられる。したがって、GRAPE においては一般的な交叉ではプログラムを効率よく進化させることができないと考えられる。

## 4. 提案手法

解析の結果から、次に示す点を基本方針とした個体の進化方法を提案する。

- ・ 遺伝子操作として突然変異を積極的に使用する
- ・ 親個体を徐々に変化させながら進化を行う

<sup>†</sup> 横浜国立大学大学院環境情報学府, Graduate School of Environment and Information Sciences, Yokohama National University

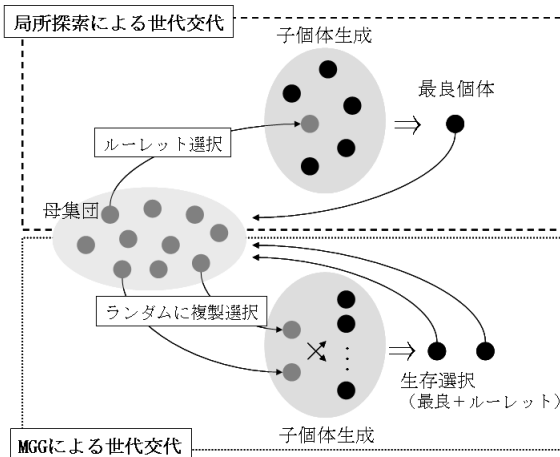


図3 提案手法の概略

図3に提案手法の概略を示す。提案手法では、選択された1つの個体に注目して子個体の生成を行う“局所探索”による世代交代と、従来のGRAPEで使用されている2つの個体を世代交代させるMinimal Generation Gap (MGG) [3]を併用する。このとき、局所探索では遺伝子操作として突然変異だけを使用して子個体の生成を行い、MGGでは一様交叉と突然変異によって子個体を生成する。局所探索の流れを次に示す。

#### 1. 親個体の選択

母集団からルーレット選択によって親個体を1つ選択する。この個体を局所探索における最初の局所探索親個体とする。

#### 2. 局所探索子個体の生成

局所探索親個体の一部を突然変異によって変化させた個体(局所探索子個体)を1つ生成する。

#### 3. 局所探索親個体の選択

生成した個体の適応度が局所探索親個体より高い、または同じであれば、この個体を新たな局所探索親個体とする。適応度が低かった場合は、局所探索親個体はそのままとする。

#### 4. 2および3の繰り返し

あらかじめ定めた数の局所探索子個体を生成するまで、局所探索子個体の生成を行う。

#### 5. 局所探索の終了

現在の局所探索親個体を、局所探索における子個体として母集団に戻す。

このように局所探索では母集団における1回の世代交代のうちに探索点を変化させて個体の生成を行う。また、適応度が同じ個体が生成された場合にも局所探索親個体の入れ替えを行っている。これは、GRAPEで生成されるプログラムには表現型に現れない遺伝子が多数存在するため、適応度が同じ個体でも遺伝子の構造は異なっているからである。これによって探索点を変化させながら探索を行うことができ、親個体を固定して複数の子個体を生成する場合と比べて、広範囲の探索を行うことができる。

表1 自動生成の成功回数(100試行中)

	階乗		フィボナッチ		累乗	
	学習	未知	学習	未知	学習	未知
局所+MGG	100	100	10	10	78	51
局所探索	100	100	14	14	69	53
従来手法	70	70	5	5	27	20

## 5. プログラム自動生成実験

提案手法の有効性を検証するため、階乗、フィボナッチ数列、累乗を求めるプログラムを自動生成する実験を行う。これらは木構造では表現することが困難な問題である。実験では、局所探索とMGGを併用した場合、局所探索だけの場合、およびMGGを用いるGRAPEの従来の進化方法についてそれぞれ100回のプログラムの自動生成を行い、その成功回数で比較を行う。なお、各試行の進化の終了条件は子個体を2,500,000個体生成するまでとした。また、提案手法において局所探索による世代交代とMGGによる世代交代を行う比率は予備実験から99:1とした。使用するノード関数は(+, -, \*, /, %, =, >, <)とし、複数の(入力, 出力)のペアを学習データセットとして与える。実験における成功は入力されたデータセットに対して正しい値を出力する構造が構築された場合とする。表1にそれぞれの成功回数を示す。ここで、未知とは構築されたプログラムに対して、学習に使用していないデータセットを入力として与えたときの結果である。

表1から、従来手法に比べて局所探索を用いた場合のほうが目的のプログラムが生成される回数が増加した。これらの結果から、探索点を徐々に移動させる局所探索が有効に働いているといえる。

## 6. まとめと今後の課題

本研究ではまず、従来のGRAPEの進化方法でプログラムの自動生成実験を行い、進化に有効な個体の生成方法について考察を行った。その結果から、1つの個体を突然変異によって徐々に変化させる局所探索を導入することで、効率良くプログラムの進化させる方法を提案した。また、局所探索がプログラムの進化において有効に働いていることを実験によって示した。これらの結果から、本研究で扱った問題より複雑な問題に対しても、従来手法より効率良くプログラムを自動生成することができると期待される。

今後の課題は、本研究で扱った問題より複雑な問題に対して提案手法を適用し有効性を示すことである。また、本研究では主に突然変異によって進化を行ったが、有効な構造を保持できるような交叉方法を検討することも重要であると考えている。

### 参考文献

- [1] John R. Koza, “Genetic Programming: On the Programming of Computers by Means of Natural Selection”, MIT Press, (1992).
- [2] S. Shirakawa, S. Ogino, T. Nagao: “Graph Structured Program Evolution”, Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2007), Vol.2 pp.1686-1693 (2007).
- [3] 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”, 人工知能学会誌, Vol.12, No.5, pp.734-744 (1997).