

GP による再帰的プログラム生成の試み

1M-10

西口 真人

藤本好司

龍谷大学理工学部数理情報学科

1 はじめに

近年、遺伝的プログラミングにおいて再帰的なプログラムを生成する試みが行なわれるようになってきた。しかし、中には再帰構造や文法に制約を加えるものもある[2]。これは、再帰構造の進化が困難であるからであろう。そこで、今回作成した mn-GP (multi niche GP) では、構造・文法に制約を加えないで自由な再帰構造の進化を考え、効率的な再帰構造の進化を行なえるニッチ的進化 (niche evolution) を提案し、ペナルティー的進化手法との比較実験を行なった。

2 mnGP とニッチ的進化

mnGP は、木構造の評価にスタックを取り入れることで再帰的なプログラムを評価することが出来る。しかし、それだけでは自由な再帰構造の個体を効率的に進化させることは困難である。なぜならば、再帰構造に必然的に必要な再帰呼び出しとその引数、再帰を止める条件分岐の再帰のための3個の部品の関係が強いために、僅かな変化が個体の振舞いを大きく変えて適応度をさらに大きく変える。したがって、再帰的な個体を進化させるためには、適応度を低下させることなく、かつ再帰停止する状態のままでこれらの関係を修正しなくてはならない。それは、複数箇所の同時修正が必要な場合もある。

そこで、mnGP のニッチ的進化では、複数の個体群を進化のニッチと考える。すなわち、各ニッチは同じ特性を持つ個体だけが集まる局所的な世界で、大局的に異種の個体が共存できる世界を構成している。それぞれの個体の特性は、その評価時の振舞いから分類する。この進化法を再帰的プログラム生成に適用すると、最も簡単なニッチの構成は、「再

帰停止する」ニッチ・「再帰しない」ニッチ・「再帰するが停止しない」ニッチの3種類である。他の問題でのニッチ構成は、ヒットした適応ケースによる複雑な階層的な構成であり、ニッチ的進化のニッチの構成は、問題に応じて自由な構成を持たせることが可能である。それぞれのニッチに属する個体は、遺伝的操作によってその属性が変わると他のニッチに移動する。すなわち、同じ特性を持つ個体だけが同じニッチに住むことができるように入れ換えが行なわれる。

また、このような移動以外に、遺伝的操作時の親として他のニッチから確率的に移民を受け入れることもある。移民元となるニッチは、その個体群サイズに比例した確率で決められて、設定された選択方法によりそのニッチから選択される。ただし移民となった親には、複製操作以外の遺伝的操作が適用される。

一方、ペナルティー的進化は、ひとつの個体群に3種類の個体が共存し、再帰停止する個体を優先的に親として選択するために、その個体以外の個体の適応度を修正する(悪くする)進化法である。

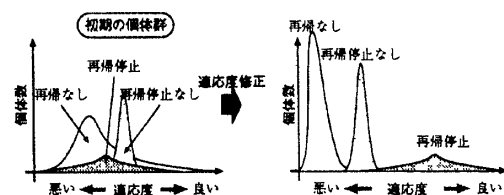


図 2.1: 適応度の修正: ペナルティー的進化

また、mnGP では遺伝的操作として従来の突然変異以外に、ノードの追加操作・ノードの削除操作・ノード変化操作を加えている。ノードの追加操作は、選択されたノード下の部分木を引数とする新しい関数ノードや染色体ノードを追加する。ノードの削除操作は、逆にノードを削除してそのノードが持っていた引数のひとつを新しい引数とする。最後にノー

ド変化操作は、選択されたノードを他の種類のノードに変える操作である。これらの突然変異は従来の突然変異よりも個体構造の変化が小さく、親の選択されたノード下の個体構造を継承できるので、僅かな構造変化が個体の振舞いを大きく変える再帰プログラムでは有効である。

mnGP は、これらの多くの遺伝的操作を個体に一度に複数回適用することが出来る。実際の適用回数は、確率的に決定される。

3 実験と結果

実験は、数値 X のフィボナッチ級数を再帰的プログラムで求めさせることである。フィボナッチ級数の再帰的プログラムは、必ず条件分岐で 2 個の再帰呼び出しと定数値 1 の返り値を持たなければならない。したがって、GP はこの関係を探索空間から見つけなければならない。

この実験では、下図のようなニッチ構成を用いる。このようなニッチ構成は、再帰停止することを第一に考えたものである。すなわち、再帰停止可能な個体が遺伝的操作により再帰構造が破壊された場合に延命させることと、後の世代で有益であるかもしれない再帰の部品を下位のニッチにキープすることで局所解への収束を防ぐことを目的としたものである。

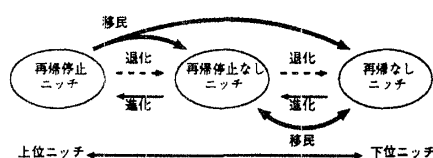


図 3.1: ニッチの関係: ニッチ的進化

この実験で用いたノード集合は次のようなものである。ノード CALL-0 は、自分自身を呼び出す再帰呼び出しを行なうものである。

$$C = \{ \text{CALL-0} \}$$

$$F = \{ +, -, \text{IFLTE} \}$$

$$T = \{ X, 0, \pm 1, \pm 2, \pm 3, \pm 4, \pm 5 \}$$

ペナルティ的進化は個体群サイズを 3000 とし、実験を行なった。ニッチ的進化は 3000 を分割して、2000, 700, 300 とし、各ニッチに割り当てた。選択方法はトーナメント法 (10 試合と 20 試合) で、親の移民は「再帰なし」・「再帰停止なし」ニッチで 50% の確率で発生し、それぞれのニッチからト

ナメント法 (10 試合) で移民を選択する。下表は、最大世代数を 200 世代とした 30 回の実験結果である。

	ペナルティ	ニッチ
最適解生成確率	50.00%	70.0%
平均生成世代	40.8	68.57
局所解収束率	50.00%	26.67%

表 3.1: フィボナッチ級数問題の実験結果

表 3.1 から、ニッチ的進化は、ペナルティ的進化に比べて 1.4 倍の確率でパーフェクトな解を生成することがわかる。ただし、ニッチ的進化の平均世代が悪いのは、次のグラフが示すように、後半の世代でも局所解に収束せずに解を生成しているからである。

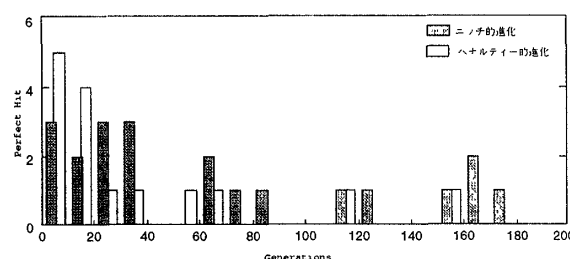


図 3.2: ニッチの関係: ニッチ的進化

4 考察と今後の課題

実験の結果からは、局所解へ収束する確率を低下させることができたが、まだ改良が必要であると考えられる。今後の課題は、数値を計算する再帰的なプログラムの局所解からの脱出のために、さらに延命の効果を向上させることである。

また、我々は現在、再帰的な探索問題で再帰構造による分類ではなく、適応ケースによるニッチ構成の実験を行なっている。このニッチ構成は、退化時の延命ではなく、再帰構造の進化を促すものである。

本研究は文部省科学研究費基盤研究 C(08680417) の補助を得て行なわれた。

参考文献

- [1] John Koza, Genetic Programming 1992
- [2] ManLeung Wong, Learning Recursive Functions