

## O12 個体群探索分岐型の遺伝的プログラミングについて

宮島 和洋 元木 達也

新潟大学大学院自然科学研究科

あらまし 遺伝的アルゴリズム (以下 GA)・遺伝的プログラミング (以下 GP) では、探索の初期段階で個体群の多様性が急速に失われてしまう早熟な収束が問題とされている。GA においては早熟な収束に対する様々な研究がなされており、例えば、局所解が多く存在する多峰性問題への対応を主眼とした個体群探索分岐型遺伝的アルゴリズム (Forking GA, 以下 fGA) [1] [2] が提案されている。本研究では fGA の個体群分岐の考え方を GP に適用し、GP における早熟な収束に対して有効であるのかを実験によって検証する。

## 1 Forking GA

fGA では 1 つの親個体群と複数の子個体群を用いて探索を行う。特に、親個体群においては、個体群の収束状況を時々観測し、収束に近づいたと判断できる時に親個体群の受け持っていた探索空間を、

- その時点で親個体群が集中的に探索している部分空間  $S_1$  と
- それ以外の部分空間  $S_2$

とに分割する。そして、 $S_1$  を子個体群に割り当て、 $S_2$  を親個体群に割り当て、それ以降、子個体群は  $S_1$  という小さな空間内でより効率的な探索を、親個体群は別の局所解の探索を、並行して行う様にする。この操作を個体群分岐 (図 1) という。分岐の際には、親個体群に含まれていた個体のうち、 $S_1$  内のは子個体群に移し、それによって不足した親個体群の個体を補う。

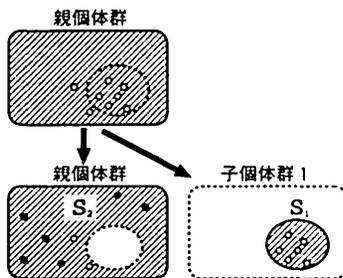


図 1: 分岐

個体群分岐の終了後、それぞれの個体群は縮小した探索空間内を探索する。親個体群では  $S_1$  内の個体は致死個体とし、子個体群では  $S_1$  からはみ出さないように遺伝的操作を施すことで、探索空間を狭めている。親個体群は更に世代を進め、個体群分岐を繰り返す。子個体群の数には上限を設けておき、上限に達したときには古いものから削除していく。

## 2 Forking GP

本研究では前節に示した個体群分岐の考え方を GP に適用する。ただし、以下の 2 点において fGA と大きく異なっている。

- (1) 収束判定の方法・探索空間の分割の方法
- (2) 個体群分岐の際の親個体群の個体の補い方

収束判定・探索空間分割について fGA のとっている方法は遺伝子長を固定しなければ適用できず、木構造をした GP の遺伝子に適用可能な方法を考える必要がある。そこで本研究では、GP においては世代が進むにつれて個体の根の付近が固まってくることに着目し、親個体群の中のある閾値以上の個数の個体が根から予め決めておいた深さまで全く同じ構造 (固定スキーマと呼ぶ、図 2) をしていた場合、すなわち木の構造も節点のラベルの付け方も全く同じである場合には、親個体群が近く収束すると判定する。そして、得られた固定スキーマ (i.e. 根付近の固まった構造) に基づいて探索空間の分割を行う。そして、子個体群ではその固定スキーマの部分には遺伝的操作を適用しないこととし、親個体群ではその固定スキーマを持った個体は致死とする。

GP で広く用いられている部分木交叉・部分木突然変異では、根に近い節点ほどこれらの操作を受けにくいという特徴があるため、世代を重ね、選択される度に根付近の構造の多様性は失われていくと考えられる。この固定スキーマによる個体群分岐は多様性の失われやすい根付近の探索も可能にすると期待される。

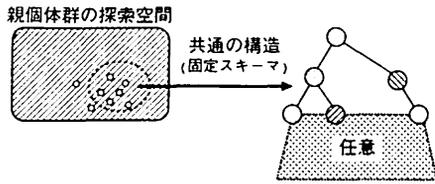


図 2: 固定スキーマ

分岐の際の個体の補充について fGA では個体群分岐 (図 1) の際に親個体群の中から  $S_1$  上の個体を子個体群に移し、それによって不足した  $S_2$  内の個体をランダムに生成することで補っていた。これに対して、本研究では分岐の際に親個体群をすべて入れ換えることとした。これは、分岐の際に  $S_1$  に含まれなかった個体も  $S_1$  に含まれる個体に類似しており標準的な初期個体と比べても適合度が高いために、親個体群が再び  $S_1$  に酷似した局所解に陥ってしまうことを防ぐためである。これにより探索効率の向上が期待できる。

図 3 に実装した GP の流れ図を示す。

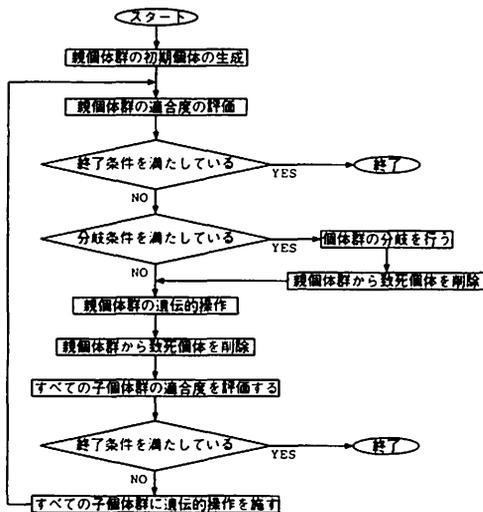


図 3: 実行の流れ

### 3 実験

Forking GP の探索能力を評価するため、通常の GP と Forking GP を関数同定問題と人工蟻問題に適用し、それぞれ 100 実行させて成功率を測定した。Forking GP については子個体群が 3 個の場合と 0 個の場合と

を測定した。子個体群が 0 個の場合には親個体群が収束に近付いたときに、親個体群を総入れ換えし、そのときの固定スキーマを探索禁止にするだけであり、子個体群は生成しない。GP では個体群の多様性が急速に失われるため、短い世代で実行を打ち切って何回も実行するという方法がよく採られているが、子個体群が 0 個の場合には、この方法に類似する。また、GP パラメータは問題ごとに固定して実験を行なった。図 4 は関数  $(x-1)^2x^2(x+1)^2$  を同定する問題、図 5 は人工蟻問題に対する実験結果である。

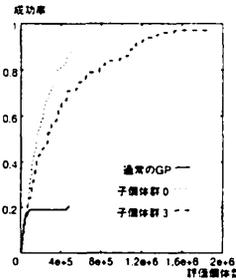


図 4: 関数同定問題

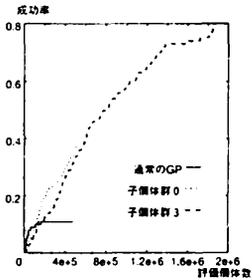


図 5: 人工蟻問題

これらの結果から、Forking GP は通常の GP に比べて評価個体数が増加しても成功率の向上が頭打ちになりにくいといえ、分岐の有効性が確認できる。しかし、子個体群が 3 個の場合は 0 個の場合よりも、成功率では上回るものの、探索効率においては劣っていることが読み取れるため、子個体群の探索効率を向上させる必要があるといえる。

今後の課題 子個体群の探索効率が芳しくない原因として以下の 2 点が考えられる。

- 親個体群が過度に収束してから収束の判定をしている
  - 子個体群の集団サイズ・遺伝的操作が適切ではない
- 今後は、これらを適切に調整し、Forking GP の探索効率を向上させることを課題とする。

### 参考文献

[1] 筒井茂義 藤本好司 “個体群探索分岐型遺伝的アルゴリズム fGA(Forking GA) の提案” 人工知能学会誌 Vol.8 No.5 pp.741-747, 1994.

[2] 筒井茂義 藤本好司 “表現型個体群探索分岐型遺伝的アルゴリズム p-fGA(Phenotype Forking GA)” 人工知能学会誌 Vol.11 No.4 pp.619-628, 1996.