

## 並列遺伝的プログラミングにおける島モデルの実験的解析\*

飛岡 良明  
東京大学工学部<sup>†</sup>

伊庭斉志  
東京大学大学院新領域創成科学研究科<sup>‡</sup>

## 1 はじめに

遺伝的プログラミング(以下 GP)は、プログラム自動生成のアルゴリズムとして多くの分野で利用され、また多数の研究がなされている。しかしアルゴリズムの性質上、計算量が膨大に膨れ上がってしまうので、複数のプロセッサを同時に利用して並列的に GP を行う PGP が現在の主流となってきている。しかし、いまだにその性質には未解明の部分が多々残されており、そのような性質を明らかにすることで更なる PGP の性能の向上が期待される。

PGP のなかでも通信によるオーバーヘッドが少なく、結果も良好であるといわれている手法として島モデルが挙げられる [2, 6]。島モデルは、母集団を小さな部分母集団(島)に分割して、各々の島ではお互いに独立に GP で探索を行う。また、ある間隔ごとでプロセッサ間の通信を行い、各島の一部が他の島へと送られる。島モデルは、単一母集団 GP よりも適合度の高い個体が発見されることが知られている。この理由としては、集団の多様性が部分集団を超えて保持されるため、早熟な収束が回避されるというようなことが考えられている。そこで本稿では島モデルの性能向上のために、性質の検証を行う。

## 2 本研究について

### 2.1 島モデルの問題点

島モデルはすでに多数の研究がなされている [2, 6] が、まだ未解明の要素も残っている

島モデルのまだ未解明である性質の一つとして、部分母集団の数(以下島の数)と部分母集団に属する個体数(以下個体数)との関係がある。この両者に何かしらの相関関係があるならば、島の数を決定するとそれに最適な個体数を決められる、といった非常に有用な情報を得られることが期待されうる。

### 2.2 先行研究

先行研究により、島の数を  $i$ 、その島の数におけるベストな個体数を  $n$  とすると

「 $n \times i \sim const$ 」という仮説が提唱されている [1]。以下  $n \times i$  の値を「ベストな総個体数」と呼ぶ。

ここで「その島数におけるベストな個体数」とは、評価個体数を固定して個体数をさまざま変えた時に、最終的にもっとも成績の良い個体を作成することができる個体数のことである。

### 2.3 本研究での検証手法

先行研究では関数同定、even parity 5 の 2 種類の問題での研究であった。そこで本稿ではさらに GP でよく使われるベンチマーク問題である 11 マルチプレクサ問題、Ant 問題、さらに回路作製問題の 3 種の問題で実験を行った。

## 3 ベストな総個体数検証についての実験

### 3.1 実験 1: 11 マルチプレクサ問題

11 マルチプレクサとは入力 3bit(A0 ~ A3) で 0 ~ 7 を表現しそれに対応する入力 (B0 ~ B7) を出力するマルチプレクサのブール代数を同定する問題である。適合度は 適合度 = 全入力 (2048 通り) 中の誤り率 と定義している。

### 3.2 実験 2: 回路作製問題

回路作製問題としては、GP を用いて要請される出力を持つ回路を自動作製する問題である [4]。今実験ではバンドパスフィルタを作製した。適合度は

$$\text{適合度} = \sum_{f=1}^n [W(d(f), f)] d(f)] \text{ と定義している。}$$

$f$  は周波数、 $d(f)$  は  $f$  に対応する要請と作製した回路での出力との差、 $W(d, f)$  は重みである。

### 3.3 実験 3: Ant 問題

Ant 問題とは、簡単な人工的な蟻の行動を制御する問題である。蟻の目標としては、 $32 \times 32$  のマス目上においてある計 89 個の餌をできるだけたくさん取ることである。本研究では、餌が一続きでなくところどころにギャップがあるのでより難しいといわれている Santa Fe Trail [3] を用いた。適合度は 適合度 = 蟻の取り残した餌の数 と定義している。

### 3.4 結果と考察

全ての問題に対して、島の数に対する最良の成績を示す個体数を求めた結果は図 1 のようになった。ここで横軸が島の数で、縦軸がベストな総個体数を島の数が 2 で

\*The Experimental Analysis of Parallel Genetic Programming

<sup>†</sup>Yoshiaki Tobioka: The University of Tokyo

<sup>‡</sup>Hitoshi Iba: Graduate School of Frontier Science, The University of Tokyo

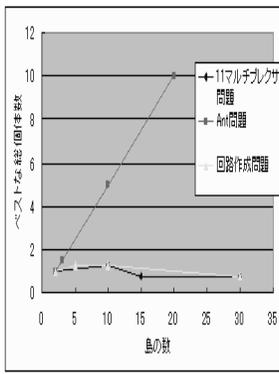


図 1: ベストな総個体数の検証

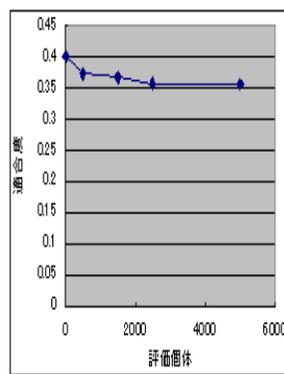


図 2: 11 マルチプレクサ

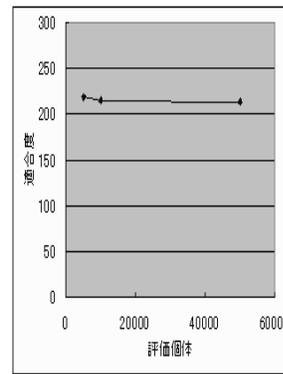


図 3: 回路作製問題

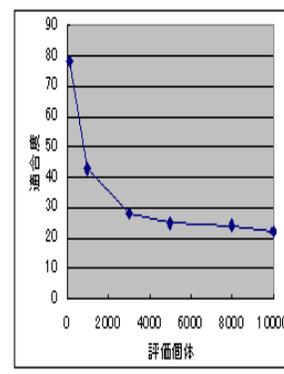


図 4: Ant 問題

あるときを 1.0 として正規化したものである。実験 1, 実験 2 に関しては「ベストな総個体数がほぼコンスタント」と仮説 [1] が確認された。一方実験 3 に関してはまったく対照的な、島の数に依らずに常に個体数が最大の時が一番成績がよいという結果になった。

3 種の実験を比較してみると、非常に対照的な結果が得られた。11 マルチプレクサ問題、回路作製問題に関しては仮説に合うような結果が得られたのであるが、Ant 問題に関しては仮説と全く異なった結果が出てしまった。この理由は以下のようなのではないかと考えられる。

先行研究と本研究で計 5 種類の問題を解いた結果 Ant 問題にのみ仮説が当てはまらなかった。これは Ant 問題の特殊性と関係があるのではないかとと思われる [5]。

島の数に依らずに、個体数が多いほど成績がよいということは、世代を重ねるという探索手法の意味が弱いことを示している。つまり、GP で探索しなくてもランダム探索で成績のよい個体が見つかることを意味していると思われる。

一方ベストな総個体数がコンスタントになる問題では、ランダム探索では成績のよい個体を作ることができず、GP の性質を用いて世代を重ねて探索をしていかなければならないのではないかと。これを確認するために次のような検証実験を行った。

#### 4 ランダム性の確認についての実験

11 マルチプレクサ問題、回路作製問題、Ant 問題各々に関して、ランダムに個体を作成して適合度を検証した。

結果をグラフにしたものが図 2~4 である。横軸はランダムに作成した個体数、縦軸は適合度であり、すべてにおいて適合度が小さいほど成績がよい。

図 2,3 と図 4 を比べると明らかなように、多数の個体をランダムに作成することで、Ant 問題以外では成績の良い個体を作成することはできないが、Ant 問題では作成できる。

以上を踏まえて、ランダムには解空間を探索できない問題では「ベストな総個体数がコンスタントになる」という仮説が適用できるとと思われる。

#### 5 おわりに

今回の研究で出た結論は以下になった。

- ランダム探索では良い成績の個体がでない問題  
島モデル GP を用いて解く場合には、ベストな総個体数がコンスタントになる。
- ランダム探索で成績のよい個体が出る問題  
いかに多数の個体を一度に計算できるかが重要であるので、島をいくつに分けたとしても個体数は多くとるべきである。

一方いまだに正確に判明していない点も残されている。それを今後の展望として記す。

1. 島を多数に分けた時の最終的な解の性能についても実験、解析を行う。
2. 個体の遺伝子を解析することによって、多様性について解析する。
3. 移民のトポロジーなど固定化しているパラメータを変化させたとしても、仮説が適用できるのか調べる。

#### 参考文献

- [1] Fernandez, F. et, al. Experimental study of multipopulation parallel genetic programming. *European Conference, EuroGP 2000*, 2001.
- [2] S. Kamio and H. Iba. A co-evolutionary approach to parallel distributed genetic programming. *Proc. of the 4th International Workshop on Emergent Synthesis*, 2002.
- [3] J. R. Koza. *Genetic Programming II-Automatic Discovery of Reusable Program*. MIT Press, 1994.
- [4] Koza, J. R. et, al. *Genetic Programming III*. Morgan Kaufmann Publishers, 1999.
- [5] W. B. Langdon and R. Poli. *Foundations of Genetic Programming*. Springer, 2002.
- [6] R. Tanese. Distributed genetic algorithms. *Proc. 3rd International Conference on Genetic Algorithms*, 1998.