

# 小さな個体を探索する GP の効率化<sup>1</sup>

5 T-3

足立 充宏      元木 達也<sup>2</sup>新潟大学大学院 自然科学研究科 情報・計算機工学専攻<sup>3</sup>

## 1 はじめに

遺伝的プログラミング (GP)[4,5] において小さく正しい個体を探索する場合、通常、適合度は正しさの項と複雑さの項の重み和として設定される。[2,3,4,5] しかし、それまでに見つかった近似最良解周辺の探索が必ずしも最適解への最短の探索であるとは限らない。本稿では最小の7入力ソーティングネットワーク [1] を見つける問題を考え、その問題に対する適合度を工夫する。これによって、一般に、問題に依存した発見的な知見を適合度に組み込めば GP の探索をより効率的に行なえることが分かる。

## 2 ソーティングネットワーク

ソーティングネットワーク (S-Net) は図1のように表される。整列する要素数分のラインをもち、入力側からそれぞれの要素がそのラインを流れていくものとする。ライン同士をつなぐ縦の連結部分で、流れてきた要素同士の比較をし、逆順であればそこで入れ替えを行う。

## 3 GP で S-Net を探索する

**個体表現:** 例えば、図1の S-Net は GP の個体として図2のように表す。

**S-Net の正しさの評価:** いくつかの入力例に対して、出力データ列内の隣合った二つの数を比較し、逆順になっている箇所を数える。それらの全ての入力例に渡る総計を評価値とする。Zero-one 原理 [1] によれば、0 と 1 だけから成る全ての入力例に対して正しく整列するネットワークはどんな入力例に対しても正しく整列動作をすることが保証されるから、 $n$  入力の S-Net の正しさの評価するためには、入力例としては全部で  $2^n$  個のビット列を考えればよい。

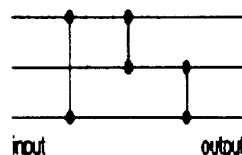


図1: 3入力 S-Net

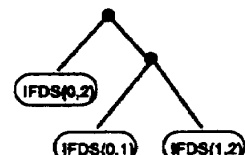


図2: 図1の個体表現

## 4 最小の S-Net を探索するための適合度関数

改良の可能性の項 ( $F_2$ ) を付加した、次のような適合度関数を導入する。

$$F = F_0 + F_1 * w_1 + F_2 * w_2$$

ここで、

- $w_1, w_2$  は各項のバランスをとるためのウェイト。
- $F_0$  は前節3で述べた正しさの評価の項。
- $F_1$  は評価対象の個体のターミナルの数。
- $F_2$  は改良の可能性の項。入力例  $i$  に対し、交換が行われた回数を  $I_i$ 、入力例の個数を  $N$  としたとき、 $F_2 = (\sum_{i=1}^N I_i) / (F_1 * N)$  とする。

つまり、個体中の全ターミナルの内、整列時に実際に要素の入れ替えが行われたターミナルの割合を  $F_2$  とし、使用されなかった無駄なターミナルの割合が高いほど良いとする。この  $F_2$  の項により、誤差と大きさ (の和) が同じ個体が存在した場合、ターミナルの使用回数の少ない方の個体、つまり整列においてより少ない交換で同じ能力を発揮する方の個体を、「その時点においては余分なターミナルを持っているが、その後の遺伝的操作により最適解になれる (近づける) 可能性が高い」と期待して選ぶ。

## 5 7入力の最小 S-Net を探索する実験

### 5.1 $F_2$ の項の有用性について

集団の大きさは300、選択方式はトーナメント、 $F_1$  のウェイト  $w_1$  は1.0(固定)とし、 $F_2$  のウェイト  $w_2$  を変えてそれぞれ200世代までに最小の解が出現するかどうかの実験を250回繰り返した。200世代までに

<sup>1</sup> Efficient GP search for small and correct solution

<sup>2</sup> Mitsuhiro Adachi and Tatsuya Motoki

<sup>3</sup> Information and Computer Engineering Course, Graduate School of Science and Technology, Niigata University, Niigata 950-2181, Japan

最適解が出現する割合は図3の通りである。 $(w_2 = 0$  で  $F_2$  を付加しない場合となる。)

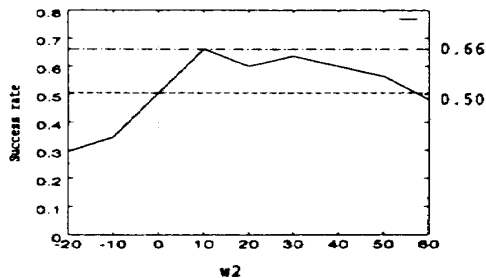


図3:  $F_2$  を付加した場合の最適解出現率

そこで  $F_2$  の項の代わりに乱数を付加し、実験を行なった(図7)。この結果より、探索効率の向上は単なる多様性の拡大によるものではないと分かる。

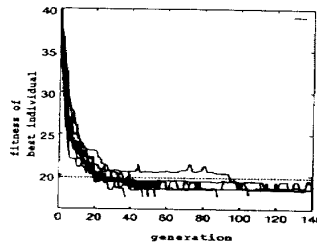


図5: 最良適合度の推移

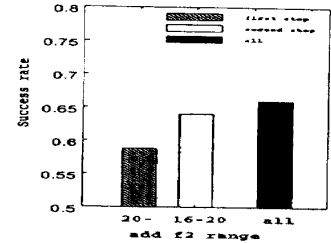


図6:  $F_2$  の付加を制限した場合の最適解出現率

### 5.2 $F_2$ の影響の大きさ

$F_2$  の項の付加によって選択操作の結果も変わってくる。 $F_2$  の項を加えることによってどの位の割合で選択操作の勝者に影響があるかを、 $w_2 = 10, 20, 30$  に対して、それぞれある GP 実行において世代毎に記録してみた(図4)。

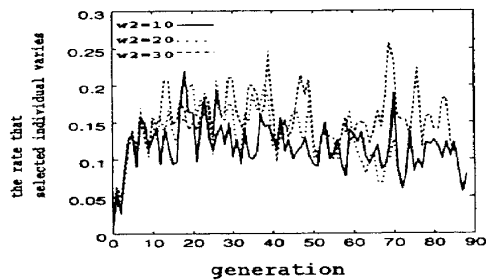


図4:  $F_2$  によって選択個体が変わる割合

### 5.3 どの段階で $F_2$ が効果を発揮するのか

進化のどの段階で  $F_2$  の項が効果を発揮しているのかを確認するために、まず、最良適合度の推移を10回の実行により観察した(図5)。推移が急激で個体間の適合度の差が大きいと思われる第一段階とその後の第二段階を最良適合度の値が20以上かどうかによって分け、それぞれの段階に  $F_2$  を付加し、 $w_1 = 1.0, w_2 = 10.0$  として実験を行なった(図6)。この結果と図3(特に  $w_2 = 0$  の所)より、 $F_2$  の項は探索の全般に渡って効果をもたらすことが分かる。図6だけを見ると  $F_2$  は初期よりも終盤に効果を発揮する様にも見えるが、世代数の長さを考慮すると、それも一概には言えない。

### 5.4 $F_2$ の効果は多様性の拡大か

$F_2$  の項を付加した場合の探索効率の向上は単に多様性の拡大による効果であるという可能性もある。

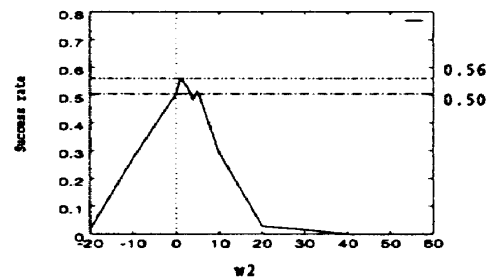


図7: 乱数を付加した場合の最適解出現率

## 6 おわりに

「改良の可能性」を考慮した適合度関数を用いて探索を行い、その効果を確認した。今後の課題としては次のようなことが挙げられる。

- 「改良の可能性」の考え方を他の探索対象に適用できるかどうか調べる。
- さらに探索効率が高く、探索の対象に依存しにくい評価基準を考える。

### 参考文献

- [1] Donald E. Knuth: *The Art of Computer Programming volume3: Sorting and Searching*, Addison-Wesley Publishing Company, 1973.
- [2] Conor Ryan: Pygmies and Civil Servants, in "K.E.Kinnear Jr.(ed.), *Advances in Genetic Programming*, pp.243-263, MIT Press, 1994."
- [3] Byoung-Tak Zhang and Heinz Muhlenbein: Adaptive Fitness Functions for Dynamic Growing/Pruning of Program Trees, in "P. J. Angeline and K. E. Kinnear Jr.(eds.) *Advances in Genetic Programming volume2*, pp.241-256, MIT Press, 1996."
- [4] 伊庭齊志: 遺伝的プログラミング, 東京電機大学出版局, 1996.
- [5] J.R.Koza: *Genetic Programming*, MIT Press, 1992.