文法進化における個体定義の改良について

Improvement of Individual Definition in Grammatical Evolution

杉浦 秀幸, 水野 貴央, 脇田佑希子, 北 栄輔 (名大)

Hideyuki SUGIURA, Takao MIZUNO, Yukiko WAKITA, Eisuke KITA,

Nagoya University

Key Words: Grammatical Evolution, Individual Definition, GEMC, GE2DG.

1 緒論

文法進化 (GE) は、関数やプログラムの探索を目的とした進化的計算手法である [1, 2]。GE では、個体を遺伝的アルゴリズムのような 1 次元の遺伝子で定義し、遺伝子型から表現型への変換にはバックス・ナウア記法 (BNF) で定義された文法を用いる。遺伝子の最適化には遺伝的アルゴリズムの探索アルゴリズムを用いる。

GE の BNF 文法に条件分岐構造を含む文法を定義する場合、これまでの研究では条件式や実行部にも 1 つの連続した 1 次元配列を用いている。そのため、遺伝的操作によって、親個体の有用な形質を破壊されやすく、探索効率が低下してしまう。そこで本研究では、条件式や実行部をそれぞれ独立した 1 次元配列を用いて定義し、条件分岐構造全体を 2 次元配列で定義する遺伝子構造を持つ GE(GE2DG) を提案する。

BNF 文法では非終端記号の遷移規則ごとに終端記号へ遷移するまでに使用する遺伝子の数が異なるので、遺伝的操作で遷移規則がかわれば、それに対応する遺伝子座から後ろの遺伝子が無意味となってしまう。このために、親個体の有用な形質が失われやすい。この問題に対し、GEMC では定義した文法の非終端記号と同じ数だけの染色体を個体の遺伝子型に用意する方法を提案している [3]。遺伝子型から表現型に変換する際には、選択された非終端記号と対応する染色体を用いる事で親個体の形質を保存しやすくなる。また交叉を 2 個体間の対応する染色体同士で行う事で、既存の GE よりも交叉によって適合度が改善しやすくなると期待できる。そこで、本稿では GE2DG と GEMC を組み合わせた遺伝子構造を持つ GE(GEMC+GE2DG) についても提案する。

解析例では、ステップ関数の関数同定問題を扱う。この問題において、GEMC, GE2DG, GEMC+GE2DG とオリジナルの GE の探索性能を比較し、提案手法の有効性を示す。

2 文法進化

2.1 文法進化のアルゴリズム

GE のアルゴリズムを以下に述べる。

1. BNF を用いて文法を定義する。

2. 初期個体群を生成する。
3. BNF 文法により個体の遺伝子型を表現型に変換する。
4. 表現型の適合度を計算する。
5. 個体集団に遺伝的操作を適用し、新たな個体集団を生成する。
6. 設定した終了条件を満たせば終了。満たさなければ 3 へ戻る。

従来型の GE では、各個体の遺伝子型を表現型に変換する操作は以下のようにして行う。

1. n -bit 毎に遺伝子を 2 進数列から 10 進数列へ基数変換する。
2. 生成している構文中で最も左にある非終端記号を α 、 α に対応する遷移規則数を n_α とする。
3. 10 進数列の値 β を n_α で割った剰余 γ を求める。
4. α に対応する遷移規則の中で γ 番目の遷移規則を選択し、 α をこの遷移規則にしたがって置き換える。
5. すべての非終端記号が終端記号に変換されるまでこれを繰り返す。

2.2 個体の遺伝子定義

本研究では、以下の 3 つの遺伝子定義の方法を比較する。

複数の染色体を用いる GE (GEMC) 文法に定義された非終端記号の数だけ染色体を遺伝子型に保持する。交叉を 2 個体間の対応する染色体同士で行う事で親個体の有用な形質を保持しやすくなると考えられる。

2 次元の染色体を用いる GE (GE2DG) 構文に含まれる多項式や関数をそれぞれ独立した 1 次元配列を用いて定義し、構文全体を 2 次元配列の遺伝子型で定義する。2 次元配列の遺伝子型を用いる事で、遺伝的操作によって親個体の形質を破壊されにくくできると考えられる。

GEMC と GE2DG の結合手法 (GEMC+GE2DG) GE2DG と同様に構文に含まれる多項式や関数をそれぞれ独立した遺伝子配列で保持するとともに、各遺伝子配列では文法に定義された非終端記号の数だけ染色体を保持する。以上の様にする事で GEMC と GE2DG の効果を持ち合わせることができると考えられる。

Table 1: 定義した文法

<code><element></code>	<code>::= if(<expr><cond><expr>){<expr>}</code> <code> else{ <element> }</code> <code> <expr></code>
<code><cond></code>	<code>::= < ></code>
<code><expr></code>	<code>::= <expr> <expr> <op> <var></code>
<code><var></code>	<code>::= <X> <num></code>
<code><op></code>	<code>::= + - * /</code>
<code><X></code>	<code>::= x</code>
<code><num></code>	<code>::= 0 1 2 3 4 5</code> <code> 6 7 8 9</code>

Table 2: パラメータ設定

全遺伝子定義 に共通	世代数	1000
	個体数	100
	交叉率	0.8
	突然変異率	0.3
Original GE	遺伝子長	10000
GEMC	遺伝子数	7
	遺伝子長	1500
GE2DG	遺伝子型の行数	10
GEMC+GE2DG	遺伝子型の1行長	1000
	遺伝子数(条件式)	6
	遺伝子数(実行部)	5
	遺伝子型の行数	50
	遺伝子長	200

3 解析例

解析例として、次式のステップ関数 $f(x)$ を用いた関数同定問題を考える。

$$f(x) = \begin{cases} 0 & (x < -3) \\ -1 & (-3 \leq x \leq 3) \\ 1 & (3 < x) \end{cases} \quad (1)$$

式(1)において、 $-10 \leq x \leq 10$ の値を0.1刻みで変化させた計201個のサンプル点のデータについて実験を行う。用いるBNF文法を表1に示す。開始記号はオリジナルGEとGEMCでは`<element>`、GE2DGとGEMC+GE2DGでは条件式が`<expr><cond_op><expr>`、実行部が`<expr>`である。遺伝的操作として、サイズ5のトーナメント選択、一様交叉、突然変異、エリート保存を用いる。エリートとして残す個体は上位5個体である。他のパラメータを表2に示す。GEのパラメータにおける交叉率と突然変異率は、事前実験において最も良い結果を示した値を設定している。実験は異なる乱数列を用いて100回の試行を行い、適合度の平均値によって比較する。適合度は、GEにより生成された関数から求めた値とサンプルデータの値の平均

Fig. 1: 関数同定問題の実験結果

二乗誤差として定義している。

最良個体の平均二乗誤差の平均値の収束を図1に示す。図の横軸は世代数、縦軸は平均二乗誤差を表す。図1より、GE2DGとGEMC+GE2DGが良い結果を示している事が分かる。このことから、条件式や実行部に用いられる多項式を独立した遺伝子配列で保存するGE2DGの遺伝子構造がステップ関数の予測に有効であったといえる。また、既存GEとGEMCを比較するとGEMCの方が良い適合度を示している事を考えると、GEMCの方が効率的に進化させることができる事が分かる。

4 結論

本研究ではGEにおける個体の遺伝子の定義を改良し、改良型GEと既存GEを実問題に適用した結果を比較することで改良型GEの探索性能について検討した。その結果、GEMC、GE2DG、GEMC+GE2DGは既存GEよりも良い探索性能を示すことがわかった。

参考文献

- [1] C. Ryan, J. J. Collins, and M. O'Neill. Grammatical evolution: Evolving programs for an arbitrary language. In *Proceedings of 1st European Workshop on Genetic Programming*, pp. 83–95. Springer-Verlag, 1998.
- [2] C. Ryan and M. O'Neill. *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*. Springer-Verlag, 2003.
- [3] 原 章, 山口智久, 市村 匠, 高濱徹行. 複数の染色体を用いたGrammatical Evolution. ファジィシステムシンポジウム講演論文集, 第24巻, 2008. CD-ROM.