Vol.2014-MPS-100 No.17
2014/9/25

リスト型遺伝子を用いた文法進化の改良について

丸田 峻也[1]　杉浦 秀幸[1]　北 栄輔[1,2,a)]

概要：文法進化 (Grammatical Evolution:GE) は，設計目的を満たす関数やプログラムを生成することを目的とした進化的計算手法の一つである．本研究では，GE における遺伝子型をリスト型で表現した改良型 GE を提案する．提案手法を関数同定問題に適用し，従来法と比較しながら収束性能について検討する．

キーワード：文法進化，リスト型遺伝子，関数同定問題

1. 緒論

進化的計算法とは生物進化の過程に着想を得た計算アルゴリズムである．進化的計算を関数同定やプログラム生成に適用しようとする研究も広く行われている．その代表的なものが，遺伝的プログラミング (GP)[1] である．GP では解探索過程で遺伝的操作により新たに生成された個体の遺伝子型から，文法的に正しい表現型を必ず生成できる保証がない．この問題を解決するために，文法進化 (GE)[2], [3] が提案されている．GE では，予め遺伝子型から表現型への変換方法をバッカス・ナウア記法 (Backus Naur Form : BNF) 等で定義して利用することで，任意の遺伝子型から必ず文法的に正しい表現型を生成するようにしている．また，GA と同様な個体遺伝子表現を用いることができるので，GA で用いられる遺伝操作を利用できる．

しかし，GE では，探索の途中で遺伝子の数値が変化すると，その後で選択される遷移規則が変化する．その結果，親個体において適切に選択されていた遷移規則が子個体に引き継がれないために探索効率が悪化することがある．この問題を解決するために，複数の染色体をまとめ個体の染色体とする GEMC や，条件分岐文の構造を比較的明示的に遺伝子に加える GE2DG が提案されている．これに対して，本研究ではリスト型遺伝子を用いる GE (GELG) を提案する．この方法では，順序リスト構造を用いて遺伝子のつながりを定義しているので，途中の遺伝子値が変化しても，それ以降の遺伝子の値は変化しないので，従来型 GE の問題点を解決できる．関数道程問題において提案手法を従来法と比較検討する．

[1] 名古屋大学
Nagoya University, Nagoya 464–8601, Japan
[2] 神戸大学
Kobe University, Kobe 657–0013, Japan
[a)] kita@is.nagoya-u.ac.jp

2. GELG

2.1 遺伝子定義

従来の GE では，2 進数列または 10 進数列で各個体を定義する．これに対して，本研究では，各遺伝子座の値を整数値とし，それらの順序関係を順序リストによって遺伝子を定義する．

2.2 アルゴリズム

GELG のアルゴリズムを以下に述べる．

(1) バッカス・ナウア記法 (BNF) を用いて遺伝子型を表現型に変換する文法を定義する．
(2) 個体をランダムに生成した整数を順序リストとした遺伝子により定義する．
(3) 個体の遺伝子型を表現型に変換する．
　(a) 最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．
　(b) 最左遺伝子値を β，β と n_α の剰余 γ とする．
　(c) α に対応する遷移規則の中で γ 番目の遷移規則によって α を置換する．
　(d) 非終端記号がなくなるまで，処理を繰り返す．
(4) 生成された構文を用いて適合度を計算する．
(5) 終了条件を満たせば終了．満たさなければ次へ進む．
(6) 個体集団に選択，交叉及び突然変異等の遺伝操作を適用し，新たな個体集団を生成する．
(7) ステップ (3) へ戻る．

GP では，ブロートを防ぐために枝狩りという操作を導入する．本研究では GE に同様な目的の操作を加えることにし，生成される構文の長さが最大構文長である $MaxN$ 以下となるようにする．

表 1　BNF 文法 (関数同定問題)
Table 1　BNF syntax (Symbolic regression problem)

(A)	$< expr >$::=	$< expr >< expr >< op >$	(A0)
		$\mid < var >$	(A1)
(B)	$< op >$::=	$+$	(B0)
		$\mid -$	(B1)
		$\mid *$	(B2)
		$\mid /$	(B3)
		$\mid \hat{}$	(B4)
(C)	$< var >$::=	$< num >$	(C0)
		$\mid < stock >$	(C1)
		$\mid < func >$	(C2)
(D)	$< num >$::=	1|2|3|4|5|6|7|8|9	(D0)〜(D8)
(E)	$< stock >$::=	X	(E0)
		Y	(E1)
(F)	$< func >$::=	$< expr > sin$	(F0)
		$< expr > cos$	(F1)
		$< expr > tan$	(F2)

表 2　シミュレーションパラメータ
Table 2　Simulation parameters

世代数	1000
個体数	100
トーナメントサイズ	5
エリート個体数	5
交叉率	0.0, 0.2, 0.4
突然変異率	0.01, 0.03, 0.05, 0.1, 0.3
遺伝子長 (GE)	1000
遺伝子長 (GEMC,GELG)	1000 × 6

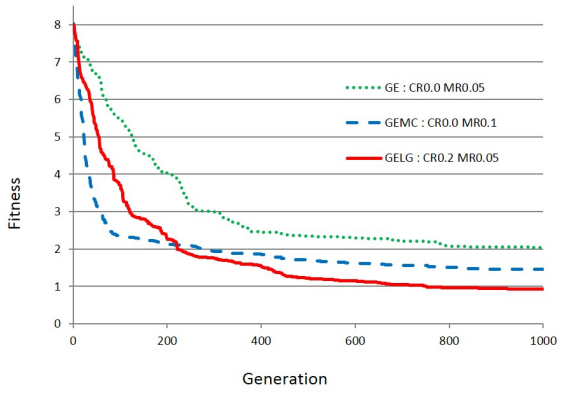

図 1　最良固体適合度の収束
Fig. 1　Convergence of best individual fitness

3. 解析例

性能比較のために，関数同定問題を扱う．関数同定問題とは，n 個の入出力データ $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ が与えられたとき，真の関数 f の近似関数 $\bar{f}$ を求めることである．但し，$y_i = f(x_i)$ である．

本実験では，真の関数 f として次式を用いる．

$$f(x) = 0.1(y-x^2)^2+2(2-y)^2+17\sin(1.5x)\sin(1.7xy) \quad (1)$$

式 (1) において，x の値を-10.0 から 10.0 まで 0.1 刻みで変化させた計 201 個のサンプル点を実験に用いる．

表 1 のような BNF 文法を定義する．開始記号は <expr> である．| は “または” を意味する．例えば，(A) の文字列が (A0) または (A1) に置き換えられることを示す．つまり，適合度は，サンプル点における真の関数と GE，GP によって生成した関数との平均二乗誤差を用いる．平均二乗誤差は式 (2) で与えられる．

$$E = \sqrt{\frac{1}{201}\sum_{i=1}^{201}(f_{(x_i)} - \bar{f}_{(x_i)})^2} \quad (2)$$

したがって，適合度は 0 に近づくほど良いこととなる．

シミュレーションパラメータを**表 2** に示す．実験は 30 回の試行を行い，最終世代における最良個体の平均適合度を示す．

本実験における最良個体の適合度の平均値のグラフを**図 1** に示す．横軸は世代数，縦軸は平均二乗誤差である．初期の段階 (世代数で 0 から 200 世代程度) では，GEMC が最も良い収束を示しているが，最終世代では GELG が GE，GEMC よりも良い結果を示している．

4. 結論

本研究では，文法進化における収束性能の改善のために，リスト型の遺伝子表現を用いた GELG を提案した．既存の GE と提案手法である GELG に対し，関数同定問題を適用し収束性能の比較実験を行った．GELG は親個体の形質を子個体に対して有効に遺伝できると考えられるため，既存 GE と比べ良い収束結果を得られた．

参考文献

[1] Koza, J. R.(ed.): *Genetic Programming II*, The MIT Press (1994).
[2] Ryan, C., Collins, J. J. and O'Neill, M.: Grammatical Evolution: Evolving Programs for an Arbitrary Language, *Proceedings of 1st European Workshop on Genetic Programming*, Springer-Verlag, pp. 83–95 (1998).
[3] C.Ryan and M.O'Neill: *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*, Springer-Verlag (2003).