

RNA から DNA への逆転写メカニズムを適用した 遺伝的プログラミングの実現

Implementation of Genetic Programming to which applied Reverse Transcription Mechanism from RNA to DNA

田中 正造^{*1}
Shozo Tanaka

外山 勝彦^{*1}
Katsuhiko Toyama

宮尾 克^{*2}
Masaru Miyao

^{*1} 名古屋大学 大学院情報科学研究科 ^{*2} 名古屋大学 情報連携基盤センター
Graduate school of Information Science, Nagoya University Information Technology Center, Nagoya University

In this paper, reverse transcription mechanism of genes from RNA to DNA is introduced into the evolution operation of GP to implement real time learning where there exist a few numbers of individuals. The experiment on generating a shooting simulation program shows that performance of the population generated by this model is more than 25% higher than the one by normal GP.

1. はじめに

遺伝的プログラミング(GP)[Koza 92]は、遺伝的アルゴリズム(GA)の遺伝子としてプログラムの構造に関する情報を表現することにより、問題解決をおこなうために最適化されたプログラムの生成を可能にした。しかし、ロボット制御、学習機能を持つ組み込みシステム、シミュレーションにおけるエージェントなどのプログラムの生成に GP を利用するときには、フロントエンドの処理で得られた情報に対応して、学習結果をリアルタイムにフィードバックさせる即時性が必要である。そのためには、集団の個体数が少ない状態で、進化性能を向上させるアルゴリズムが必要となる。

そこで、本稿は、RNA から DNA への遺伝子の逆転写メカニズムを GP の進化操作に導入し、学習のリアルタイム性を実現し、個体数が少ない状態で進化性能を向上させる手法を提案する。真核生物では、遺伝子情報(DNA)を RNA に転写し、RNA に写し取られた遺伝情報が翻訳されて蛋白質が合成されるが、その過程において、RNA から DNA へ遺伝情報が逆転写されることが Temin と Baltimore らによって明らかにされた(Fig.1)。さらに最近の遺伝子解析の研究[IHGSC2001]によると、人間の DNA における逆転写遺伝子の割合が他の動物と比べ、格段とその割合が高いことが明らかになっている。このように、RNA から DNA への遺伝子の逆転写は、人間を含む高等生物の進化に寄与したとされる。本稿はこの知見を GP に応用するものである。

2. 先行および関連研究

本研究の先行研究および関連研究として、ウイルス進化論に基づく遺伝的アルゴリズム(VE-GA) [中谷, 金杉, 近藤 99]がある。VE-GA では、進化操作として交叉を用いず、突然変異と個体の集団のウイルス感染によって進化する。その最適化メカニズムは、適応度の高い染色体を持つ遺伝子の一部を集団全体に拡散するものである。それに対して、逆転写を宿主に対する感染と見なすこともできるが、しかし、本提案手法では、遺伝的操作として交叉を導入し、また、セントラルドグマを拡張する方式を採っている。

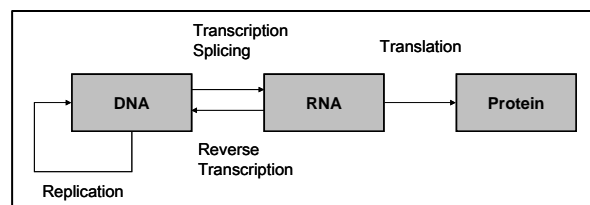


Fig. 1 Reverse Transcription from RNA to DNA

一方、ウイルス進化論を線形表現型 GP[Mike and Martin 94]に適用したウイルス進化型遺伝的プログラミング(VE-LGP) [中村, 山田 2002]も提案されている。VE-LGP では GP の個体長の変異性を考慮しウイルスと同じ構造を持つ宿主の遺伝子列をランダムに探索し、ウイルスの遺伝子列を宿主に感染させる操作をおこなう。それに対して本提案手法では、DNA 型遺伝子から RNA 型遺伝子に転写し、適応度の高い RNA 型遺伝子から DNA 型遺伝子に逆転写する。

3. 提案手法

3.1 セントラルドグマ逆転写型モデル

本稿で提案するセントラルドグマ逆転写型モデルでは、個体に DNA 型遺伝子と RNA 型遺伝子を実装する。DNA 型遺伝子は、複数の木構造遺伝子からリニアな遺伝子に変換した遺伝子で、数十個の遺伝子の転写パターンが存在する。その遺伝子パターンを RNA 型遺伝子にランダムに転写することにより、多様性を継続的に保持することができる。転写後の RNA 型遺伝子の適応度を評価し、適応度の高い RNA 型遺伝子を DNA 型遺伝子に逆転写することにより、個体の適応度の向上をはかる。

また、このモデルでは、遺伝子の転写および逆転写が元の遺伝子構造を壊さない。したがって、適応度の高い遺伝子を評価でき、同時に、元の遺伝子は適応度が高くなくてもそのまま利用し続けることができる。

本稿で提案するアルゴリズムを次に示す(Fig.2)。

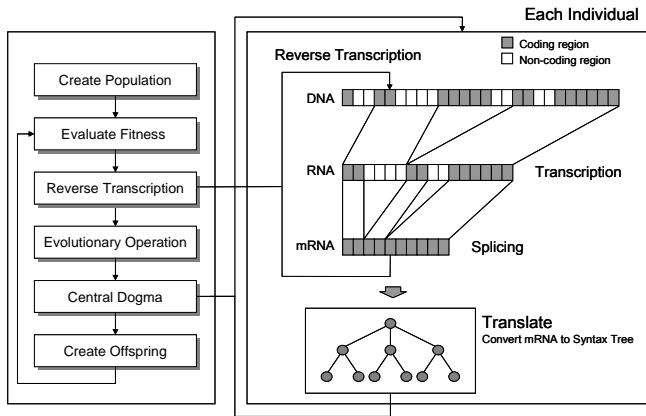


Fig. 2 Proposed Algorithm Method

①集団の生成

木構造 RNA 型遺伝子をランダムに生成する。その際、プログラムの構文規則に合った正しい木構造を作るためのテンプレートをあらかじめ用意しておき、それに従って生成する。

次に、 n 個の木構造 RNA 型遺伝子をリニアな RNA 型遺伝子に変換し、それらを連結する。

②個体の適応度の評価

木構造 RNA 型遺伝子で表現されたプログラムを実行し、適応度を評価する。

③RNA 型遺伝子から DNA 型遺伝子への逆転写

適応度の高い木構造 RNA 型遺伝子を木構造 DNA 型遺伝子に逆転写する。

④進化的操作(交叉と突然変異)

木構造 DNA 型遺伝子の部分木に対して、ノードの種類を用いて交叉可能なノードを特定した後、交叉をおこなう。同時に、初期個体で生成されたリニアな DNA 型遺伝子に対しても交叉をおこなう。この操作は、次の⑤において、転写される元の遺伝子中の優良な遺伝子を伝播させることを考慮したものである。

また、木構造 DNA 型遺伝子に対する突然変異をおこなう。なお、リニアな DNA 遺伝子の突然変異は、その実現の方法を検討しているため本モデルでは導入しない。

⑤DNA 型遺伝子から RNA 型遺伝子への転写

木構造 DNA 型遺伝子をリニアな DNA 型遺伝子に変換する。次に、その断片をリニアな RNA 型遺伝子に転写する。

⑥翻訳

リニアな RNA 型遺伝子を木構造 RNA 型遺伝子に変換する。

⑦世代交代

②へ戻る。

3.2 木構造遺伝子とリニアな遺伝子の変換

木構造遺伝子とリニアな遺伝子は相互に変換することができる。以下にその方法を述べる。

(1) 木構造遺伝子からリニアな遺伝子へ変換する場合

木構造遺伝子の各ノードを巡回しながら、プログラムの構造に応じたノードの種類 a 、ノードの子ノード数 n 、および、ノードのラベル i から構成される 3 項組 (a, n, i) をエンコードする。

(2) リニアな遺伝子から木構造へ変換する場合

エンコードされた 3 項組 (a, n, i) のうち、ノードの種類 a とノードのラベル i を用いて、再帰的に子ノードを n 回呼び出して木構造遺伝子に変換する。

5 章で述べる評価実験に用いたプログラムにおけるノードの種類を Table.1 に示す。また、上述のコードとそれに対応したプログラムの構造の例を Fig.3 に、また、それをデコードして得られたプログラムを Fig.4 にそれぞれ示す。

Table 1 Nodes of program trees

Index	Kinds of node
0	Assign
1	Binary Function
2	Binary Operator
3	BooleanUnaryFunction
4	Condition
5	Constant
6	Expression
7	Relational Operator
8	Statement
9	Terminal Function
10	Unary Function
11	Variable

8 1 0	Compound statement
4 3 2	Condition
7 2 0	Relational Operator
1 2 0	Binary Function
5 0 176	Constant
5 0 208	Constant
1 2 1	Binary Function
5 0 296	Constant
5 0 424	Constant
8 1 0	Compound statement
9 0 2	Terminal Function
8 1 0	Compound statement
4 3 2	Condition
7 2 0	Relational Operator
1 2 1	Binary Function
5 0 256	Constant
5 0 256	Constant
10 1 1	Unary Function
5 0 160	Constant
8 1 0	Compound statement
9 0 1	Terminal Function
8 1 0	Compound statement
9 0 2	Terminal Function

Fig. 3 Coded program and its structure (example)

```

if( CountEnemyUpRight( 176, 208 ) > CountEnemyUpLeft( 296, 424 ) )
{
    Shoot( void );
}
else
{
    if( CountEnemyUpLeft( 256, 256 ) > CountEnemyRightRadius( 160 ) )
    {
        MoveLeft( void );
    }
    else
    {
        Shoot( void );
    }
}
    
```

Fig.4 Decoded program (example)

3.3 転写する遺伝子領域

環境に対して自律可能なロボット制御やシミュレーションのエージェントの学習には、環境に適応し、問題解決するためルール生成が重要である。このようなルール生成を対象とする問題では、条件の最適化が重要な要素であることから、プログラム中の条件ブロックの系列の生成を対象とする。

本手法では、木構造の遺伝子を生成するときに、条件処理に対して真理値を返すブロックをテンプレートを用いて生成し、真理値を返す条件ブロックをリニアな DNA 型遺伝子からリニアな RNA 型遺伝子へ転写できるようにしている。その転写領域は、条件式の先頭を示すコードから条件式の終了を示すコードまでとする。Fig.5 に転写領域の例を示す。

3.4 逆転写する遺伝子領域

本手法では、木構造 RNA 型遺伝子から木構造 DNA 型遺伝子へ逆転写する遺伝子領域は、RNA 型遺伝子の木構造の遺伝子全体とする。

4. 評価実験

本手法の有効性を検証するために評価実験をおこなった。本手法により生成するプログラムとして、シューティングシミュレーションにおけるエージェントのルールを取り上げ、エージェントの得点アップを個体の性能向上の指標とした。

4.1 シミュレーション仕様

シューティングシミュレーションの仕様は次の通りである。自機は 8 ドットごとに左右に移動可能で、1 フレームに自弾を 10 発だけ発射できる。敵機はランダムな位置から 8 ドットごとに落下する。自機は当初 100% のエネルギーを保持するが、自弾を 1 発発射するたびにエネルギーを 10% ずつ消耗し、また、敵機に自弾が命中するとエネルギーが 20% ずつ回復する。敵機が自機に衝突すると自機はダメージを受けてエネルギーが 50% 減少し、エネルギーがなくなるとシミュレーションは終了する。

4.2 使用する関数

本シミュレーションで使用する自機用のレセプタ関数とエフェクタ関数を示す。

レセプタ関数 1 (Fig.6), レセプタ関数 2 (Fig.7), レセプタ関数 3 (Fig.8) は、それぞれ、自機の前方左右の矩形領域、自機の前方および左右、自機の左右の扇形領域に存在する敵機の存在の有無、または、その数を取得する関数である。

一方、エフェクタ関数 (Fig.9) は、自機を左右に 8 ドット単位で移動する関数、自弾を発射する関数、何も動作を行わない関数の 4 種類がある。

4	3	2	Condition
7	2	0	Relational Operator
1	2	0	Binary Function
5	0	176	Constant
5	0	208	Constant
1	2	1	Binary Function
5	0	296	Constant
5	0	424	Constant

Fig. 5 Transcription region (example)

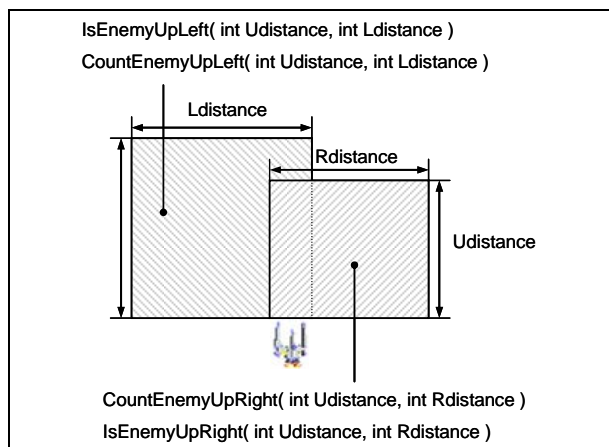


Fig.6 Receptor function 1

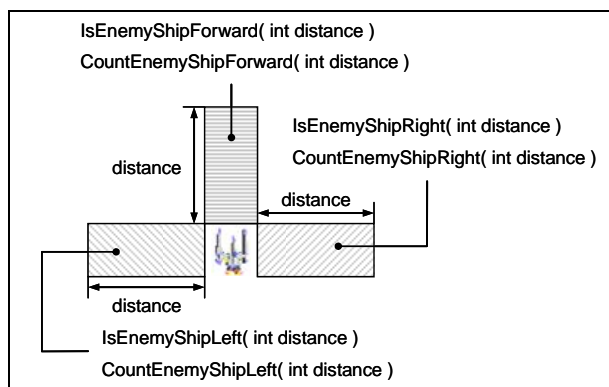


Fig.7 Receptor function 2

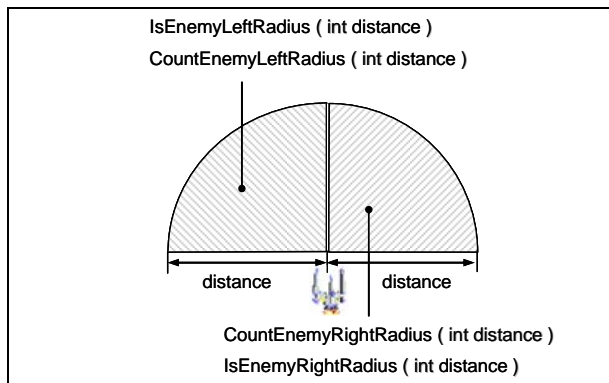


Fig.8 Receptor function 3

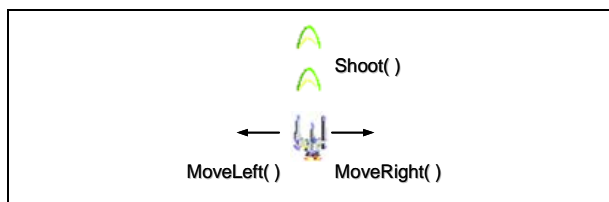


Fig.9 Effector function

4.3 実験パラメータ

評価実験において用いたパラメータとその値を Table 2 に示す。特に、本手法は個体数が少ない場合を対象としているので、個体数は 50 個とした。

Table 2 Parameters of experiment

最大世代数	200
個体数	50
交叉率	0.8
突然変異率	0.6
木構造 DNA 型遺伝子数	20
適応度(標準適応度)	撃墜した敵機数+フレーム数

以下、Table 2 に示す適応度を標準適応度とよぶ。標準適応度は 10 試行の結果の平均とする。

さらに、標準適応度を次の(1)式により、0.0 から 1.0 までの実数値に調整したものを調整適応度とよぶ。

$$\text{調整適応度} = \frac{1}{1 + \text{標準適応度}} \quad (1)$$

4.4 実験結果

(1) 実験 1: 従来手法との性能比較

従来手法と本手法(逆転写型 GP)の性能比較をおこなった。その際、調整適応度が 0.6 以上の個体に対して逆転写をおこなった。実験結果を Fig.10 に示す。グラフの横軸は世代数、縦軸は標準適応度を示す。

従来手法に比較して、本手法の結果は 100 世代と 200 世代でも平均 25%の性能の向上が見られた。また、最優良個体の性能では、200 世代で 28%の性能向上が見られた。

(2) 実験 2: RNA から DNA への逆転写する個体の調節

本手法において、RNA から DNA への逆転写の対象とする個体を適応度の違いにより選別したときの性能比較をおこなった。逆転写の対象とする個体を調整適応度 0.6 以上のものとする場合と、同 0.8 以上のものとする場合について、実験をおこなった結果を Fig.11 に示す。グラフの横軸は世代数、縦軸は標準適応度を示す。

調整適応度 0.6 以上の場合より、調整適応度 0.8 以上の場合の方が、100 世代で平均 20%の性能低下、200 世代では平均 17%の性能低下が見られた。しかし、最優良個体の性能については、両者に差は見られなかった。

この結果より、逆転写の対象とする個体を適応度の高いものに絞り込むことは、わずかな数の超エリート個体を生成することができるが、集団の適応度向上にはつながらないことが分かる。実際、実験では 1 試行でしか良好な結果は得られなかった。

一方、両者の間に最優良個体の性能の違いが見られなかった。このことから、集団の個体の標準適応度を高い水準で維持するには、比較する調整適応度の値を低く調節し、逆転写するための個体の数を増やすことが重要であると考えられる。

5. まとめ

本稿は、RNA から DNA への遺伝子の逆転写メカニズムを GP の進化操作に導入し、進化性能を向上させる手法を提案し

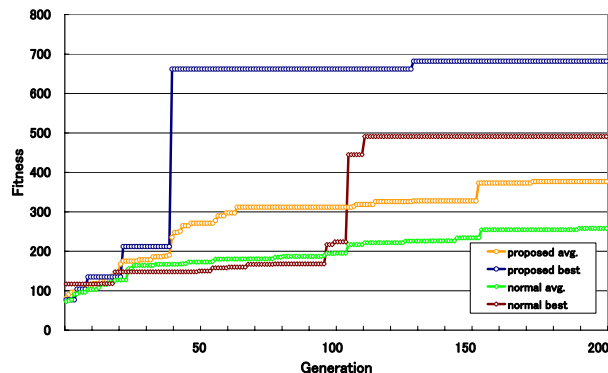


Fig. 10 Comparison of normal GP and proposed GP

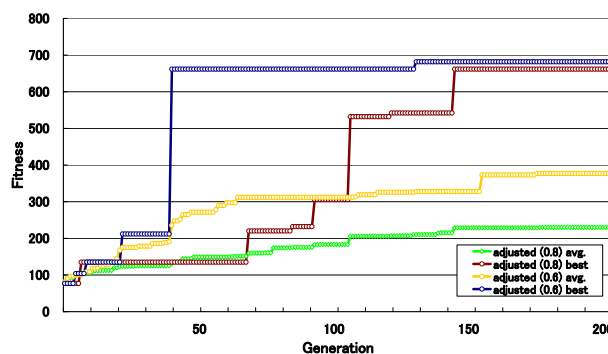


Fig. 11 Regulation of reverse transcription

た。評価実験を行った結果、個体数が少ない場合に対して、従来の GP に比べ、集団の性能を 25%以上向上できることが明らかになった。このことは、フロントエンドの処理に対して学習結果をフィードバックするときの計算量を低減できることを意味する。

したがって、リアルタイム性を要求するプログラム生成に対して、本手法は有効であると考えられる。

今後の課題として、リニアな DNA 型遺伝子の突然変異の実現や機能化されたプログラムモジュールの生成手法の検討などがある。

参考文献

[Koza 92] John R. Koza: Genetic Programming: On the programming of computers by means of natural selection. MIT Press, 1992.
 [Mike and Martin 94] Mike J. Keith and Martin C. Martin: Genetic Programming in C++: Implementation Issues, Chapter 13 of Advances in Genetic Programming, MIT Press, 1994.
 [IHGSC2001] International Human Genome Sequencing Consortium: Initial sequencing and analysis of the human genome, Nature Vol. 409, No.6822, pp. 860-921, 2001.
 [中谷, 金杉, 近藤 99] 中谷, 金杉, 近藤: ウィルス進化論に基づく進化型アルゴリズム, 情報処理学会論文誌, Vol.40, No.5, pp.2346-2355, 1999.
 [中村, 山田 2002] 中村, 山田: ネットオークション不正行為の GP オートマソンによるモデル化に関する研究, 電気通信普及財団研究調査報告書第17号, pp. 568-575, 2002