遺伝的プログラミングを用いた文字列処理プログラムの生成

4AG－10　掛川 智央　岡部 洋一
東京大学 大学院 工学系研究科 電気電子専攻
kakepee@okabe.rcast.u-tokyo.ac.jp

1 はじめに

近年、遺伝的アルゴリズム(Genetic Algorithm: GA)やその一種である遺伝的プログラミング(Genetic Programming: GP)を用いて、様々な問題を解いたり、プログラムを自動生成するという研究が盛んに行われている。

本研究では文字列に着目し、型付きのGPを用いて、いくつかの例示データから、入力された文字列を別の文字列へ変換するプログラムを生成することを目指す。

2 本システムの構成

2.1 例示データ

欲しいプログラムを得るために、文字列の例示データを設定する。例示データは入力文字列と教師文字列の対が複数組になったものである。

2.2 データ型

型付きGPで用いるデータ型として、本システムでは、文字列を表すString、整数を表すInteger、真偽値を表すBooleanの3つのデータ型を用いている。

2.3 プログラムの構成要素

各個体のプログラムは木構造で表されるが、木構造を構成する要素には終端部分、非終端部分の2種類ある。終端部分が定数、入力を表し、非終端部分が関数を表す。以下は本システムで用いている要素である。

- 終端部分

値	型名
0	Integer
true	Boolean
Input(入力文字列)	String

- 非終端部分

戻り値	関数名(引数)
String	AddString(String, String)
String	CharAt(String, Integer)
Boolean	Equals(String, String)
String	Insert(String, Integer)
Boolean	Match(String, String)
String	Sub(String, Integer, Integer)
String	ToUpper(String)
String	ToLower(String)
Integer	Inc(Integer)
Integer	Dec(Integer)
任意	If(Boolean, 任意, 任意)

2.4 適応度の計算

各個体の適応度を次式にしたがって設定する。$|output|$ はプログラムの出力文字列の長さ、$|right|$ は教師文字列の長さ、$|same|$ は出力文字列と教師文字列の最大の共通部分の長さ、n は例示データの個数である。

$$fitness = \sum_{n} \{(|output| - |same|) + (|right| - |same|)\}$$

3 プログラム生成

実際に表1のような文字列変換をするプログラムの生成を試みた。

	変換前	変換後
1	"String"	"trin"
2	"string"	"String"
3	"*string", "string"	"string", "string"

表1: 生成を試みるプログラム

各実験で共通に用いたパラメータは表2の通りである。

Tomoo KAKEGAWA, Yoichi OKABE
Dept. of Electronic Engineering,
Faculty of Engineering, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113, JAPAN

パラメータ	値
集団数	100
実行停止世代	20
世代間ギャップ	0.5
mutation 発生確率	0.05(回/個体)
inversion 発生確率	0.1(回/個体)
crossover 発生確率	1.0(回/個体)
エリート個体の割合	0.02
トーナメントサイズ	2
個体の大きさの制限	25(1 個体当り)
例示データ数	5

表 2: 各パラメータ

3.1 生成例 1

結果は、10 回試行して 8 回、期待するプログラムが得られた。2 回は 20 世代内では期待したプログラムは得られなかった。図 1 は、試行のうち 12 世代目に期待するプログラムが得られた例の、平均適応度の推移である。図 2 は期待するプログラムの生成された過程である。

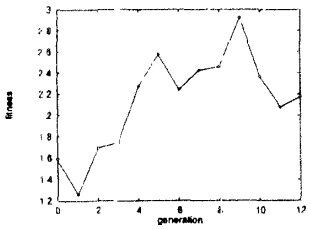

図 1: 平均適応度の推移

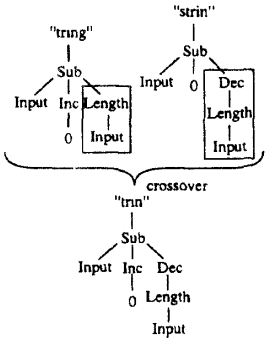

図 2: プログラム生成の過程

3.2 生成例 2

結果は、10 回試行して 4 回、期待するプログラムが得られた。6 回は 20 世代内では期待したプログラムは得られなかった。図 3は、試行のうち 3 世代目に期待するプログラムが得られた例の、平均適応度の推移である。図 4 は期待するプログラムの生成された過程である。

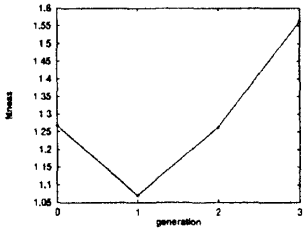

図 3: 平均適応度の推移

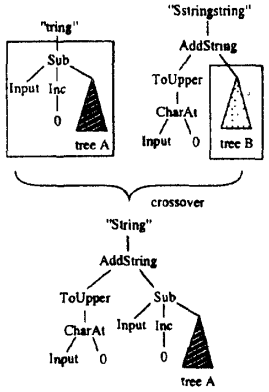

図 4: プログラム生成の過程

3.3 生成例 3

結果は、10 回試行して 1 度も 20 世代内ではプログラムは得られなかった。先頭一文字を削除するという解や、何も操作せずそのまま入力を出力するという解などの局所解に落ち込んでしまった。

4 おわりに

本研究では、遺伝的プログラミングの手法を用いて、文字列を処理するプログラムの生成を試みた。システムが開発段階ということで、繰り返し操作や変数操作を含んだプログラムが生成できない。また、個体の適応度の計算手法もあまり根拠があるものとは言えない。また、コンピュータ上の問題から集団数が 100 と非常に少ない条件になっているので、これは致命的である。このように問題は非常に多いが、今後はこれらの問題を少しづつ解決していく予定である。

参考文献

[1] 北野宏明編:「遺伝的アルゴリズム」産業図書 1993

[2] 伊庭斉志:「遺伝的プログラミング」電機大出版局 1996

[3] David J.Montana "Strongly typed genetic programming", *Evolutionary Computation* 3(2):199-230, 1995