

遺伝的アルゴリズムと人工知能

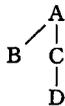
伊庭 斉志

1. はじめに

本稿では、人工知能における記号的な表現を扱えるように遺伝的アルゴリズム (GA) の手法を拡張する。本章の目的は、従来のGAを次のように補うを試みるものである。

1. 探索のための的確な部分構造の把握
2. 問題の表現形式にもとづいた効果的な探索の実現
3. より高次の知識の適応的な学習システムの構築

以下では、構造的な表現としてグラフ構造 (特に木構造) を扱う。木はサイクルをもたないグラフのことであり、



のような構造を言う。これは人工知能における多くの知識表現で用いられる構造である (たとえば概念形成の決定木や意味ネットワークなど)。木構造は LISP の S 式で記述でき、たとえば上の木は、

(A (B
(C (D))))

もしくは簡略化して、

(A B
(C D))

となる。したがって以下では木構造と LISP の S 式を同一視する。

構造的表現をGAで扱う試みはさまざまな分野で研究されている。たとえば [Kitano90] では、ニューラルネットワークの結合状態を示すグラフをGAを用いて設計した。また [Davis91] は、GAを用いてグラフの採色問題 (一定の色数でグラフのノードを採色し同色となる隣接ノードの数を最小にする問題) の近似解を求めた。こうした

いば ひとし

電子技術総合研究所 知能情報部 推論研究室

〒305 つくば市梅園1-1-4

従来の研究では、グラフを1次元文字列の遺伝子コードに写像し、通常のGAの枠内で最適化を行っていた。たとえば [Kitano91] では、グラフの隣接行列を遺伝子の1次元配列に写像する手法と、Lシステムにもとづくグラフ生成ルールを文字列として扱う手法が提案されている。これらの手法は各領域で一定の成果をあげているが、グラフなどの構造的表現をそのままの形式で遺伝子として扱ってはならず、その結果グラフのもつ構造的特性が適応評価に十分には反映されない場合も多いと考えられる。

このような問題を解決するため、本章では構造的表現を直接遺伝子コードとして扱えるようにGAを拡張する。これは Genetic Programming と呼ばれ、Koza により提唱された手法である [Koza90]。また関連する理論的研究としては、Lシステムやグラフ変形文法がある [Ehrig87]。以下ではKozaの手法をもとにして、GAを用いた構造的表現の適応学習を説明し、関数合成や知識獲得への応用について述べる [Iba92b, 93] [伊庭90, 91 a, b]。

本章では、便宜上GAで扱う情報として、PTYPEとGTYPEの2層構造からなる表現形式を採用する [Langton89]。GTYPEは遺伝(子)型のアナロジーであり、GAのオペレータの操作対象となる。PTYPEは表現型であり、GTYPEの環境内での発達に伴う大域的な行動や構造の発現を表わす。次世代を生成する際の選択はPTYPEをもとにして計算された適合度に依存する。すなわち、適合度の高いものほど生き残りやすく、より多産なようにGAは進行する。

2. 構造的表現へのGAの拡張

木に対するGAのオペレータとして、以下を導入する (図1)。これらはビット列を対象とする従来のGAオペレータの自然な拡張である。

Gmutation ノードのラベルの変更

Ginversion 兄弟の並べ換え

Gcrossover 部分木の取り換え

これらのオペレータをLISPの表現木に適用した例を

図2に示す。この適用をLISPのS式で記述すると表1のようになる。ただしオペレータの適用部位には下線を付した。

構造的表現に対する遺伝アルゴリズムは次のようになる。

Step 1 ランダムに木構造 GTYPE $\{g_i(i)\}$ を構成する

Step 2 各 GTYPE $g_i(i)$ の表現型 $p_i(i)$ に対して適合度 $f_i(i)$ を求める。

Step 3 適合度の大きな GTYPE に対して一定数のペアを取り出す

Step 4 取り出したペアに対して **Crossover** を適用し、適合度の小さな GTYPE と置き換える

Step 5 各 GTYPE に関してランダムに **Inversion, Mutation** を適用する

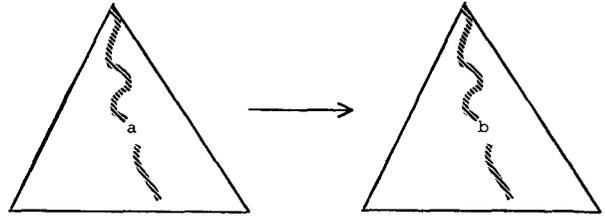
Step 6 以上によって求められた新しい GTYPE を次の世代の $\{g_{i+1}(i)\}$ として、STEP2 へ戻る

こうして拡張したGAの枠組みは、

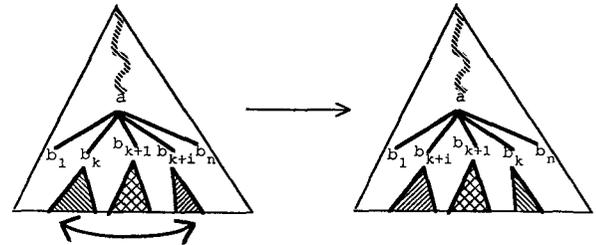
- GTYPE から PTYPE への変換
- 適合度の計算

を適切に設計することでさまざまな問題に適用可能である。GAによる構造的表現の探索能力については、グラフの獲得やランダム木の収束などの実験において、その有効性を確認した。詳しくは[Iba92b]を参照されたい。以下では、関数合成、知識獲得などへの本手法の応用につ

mutation



ginversion



gcrossover

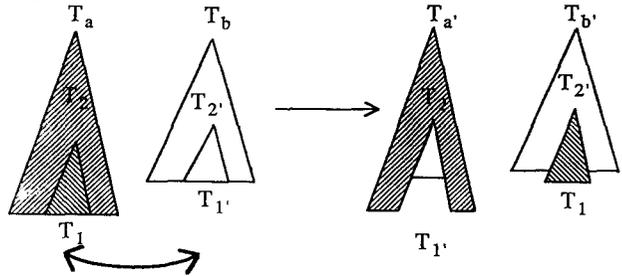
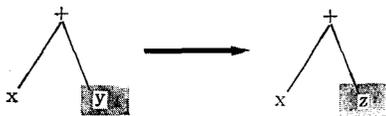


図 1

Gmutation



Ginversion

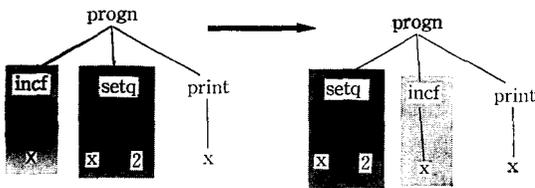


図 2 (その1)

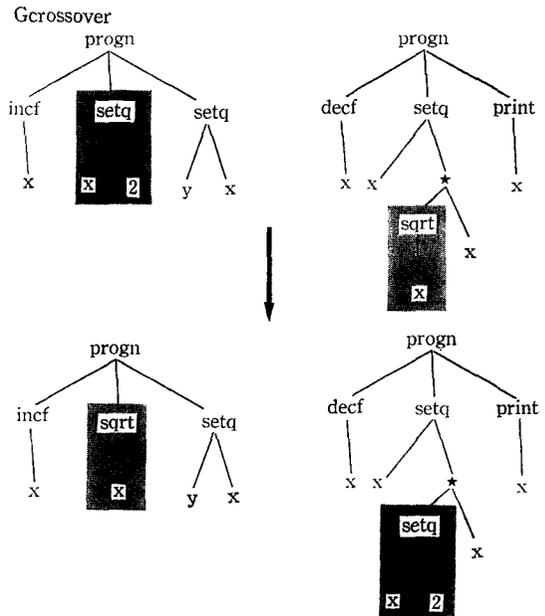


図 2 (その2)

表 1 S式の変換

Gmutation	(+ x y) ↓ (+ x z)
Ginversion	(progn (incf x) (setq x 2) (print x)) ↓ (progn (setq x 2) (incf x) (print x))
Gcrossover	(progn (incf x) (setq x 2) (setq y x)) (progn (decf x) (setq x (* (sqrt x) x) (print x)) ↓ (progn (incf x) (sqrt x) (setq y x)) (progn (decf x) (setq x (* (setq x 2) x) (print x))

いて説明する。

3. 関数の合成問題

再帰的定義を含む Fibonacci 関数を GA を用いて自動合成する。Fibonacci 数列とは、1, 1 を初めの 2 項として、前の 2 項をつぎつぎに足して得られる数列 (1, 1, 2, 3, 5, 8, 13, ...) である。Fibonacci 関数 (*fib n*) は、この数列の第 *n* 項を与える関数である。

GA にもとづく学習の実現のために、GTYPE を V_N と V_T から生成される木構造、PTYPE を GTYPE の表現を評価した結果とする。ここで V_N は、非終端記号、一般には関数記号である。木構造において V_N に属するノードの子の数は関数としての引数の数 (arity) に一致し、一定のものと不定のものがある。 V_T は終端記号であり、変数や定数を表現する。

たとえば、 $V_N = \{f, +, -\}$, $V_T = \{0, 1, 2, x\}$ のときを考える。ここで *f*, +, - の arity は 1, 2, 2 である。このとき、

$$GTYPE = (f (f (- (- (+ 0 x) 1) 1)))$$

に対応する表現型は、通常のセマンティックスでは、

$$PTYPE = (f (f (- x 2)))$$

となる。

以下では、

$$V_N = \{fib, mn+, my-\},$$

$$V_T = \{1, 2, j-index\}$$

とする。ここで *my+* と *my-* は 2 引数をとる通常の加算と減算を示し、*fib* は再帰定義に用いる関数名である。 *j-index* は Fibonacci 関数の引数となる変数を示す。適合度としては、*j-index* にさまざまな値 (1 から 20) を代入したときの PTYPE が表わす関数が返す値と、実際に Fibonacci 関数が返すべき値との合致率を

採用する。したがって適合度が 1.0 となる PTYPE が、Fibonacci 関数と入出力が一致する関数である。ただし必ずしも定義としては一致しない。さらに、 V_N と V_T からランダムに生成された関数 (2 節のアルゴリズムの初期生成部分, STEP 1) は一般に無限ループを含み得るので、eval-hook 等の機能によりループの過度の巡回を回避する必要がある。その場合の出力値は正解とは一致しないものとする。

ここでは GA のパラメータとして、集団数 200, Gcrossover の確率 0.6, Gmutation の確率 0.0333 とする。簡単のため Ginversion は省略した。この場合、第 7 世代において、

```
(defun fib (j-index)
  (my+ (fib (my-j-index 1))
       (fib (my-j-index 2)))))
```

というような関数を獲得した。これは Fibonacci 関数の定義と一致する。ただし簡単のため *j-index* が 1 と 2 の場合の値(1)は既知としている。図 3 はこの実験における関数生成のトレース (第 5 から 7 世代まで) を示す。図からわかるように第 7 世代において、115 番目 (適合度 0.0714) と 130 番目 (適合度 0.050) の親の Gcrossover によって上の GTYPE が得られた。Gcrossover の適用部位が図において Dewey Decimal Notation ([Knu- th67]) で示されている。この場合は、115 番目の GTYPE の 1 番目の子ノードの 1 番目の子ノード (0.1.1) と、130 番目の GTYPE の 2 番目の子ノードの 1 番目の子ノード (0.2.1) との交叉である。この時まで適用されたオペレータの数は、Gmutation が 251 回、Gcrossover が 438 回である。図中の s-fitness は適合度をスケールした値を示す [Goldberg89]。これは適合度の差異を際立たせるために用いられる。

筆者は同様の手法を用いて、いくつかの関数の合成実験や構造的クラシファイアによる補食行動の進化的学習の実験を行ない、その有効性を確認した [Iba92c]。また Koza らはこの手法をもとにしてさまざまな関数合成の実験を行ない成功を収めている。詳しくは文献 [Koza 90] を参照されたい。

4. 構造的 GA にもとづく GMDH

システム同定問題は、未知のシステムのふるまいを入出力の観測値から予測するものである。この問題の難しさは、扱うべき変数、データ、制約などによる組合せ論

Population report				
#	string	Generation	fitness	s-fitness
4)	(FIB (MY- 2 J-INDEX))	5	0.0714	0.0708
128)	(FIB 2)	5	0.0714	0.0708
141)	(MY+ 1 (MY+ (FIB (MY- J-INDEX 1)) (FIB 1)))	5	0.0714	0.0708
150)	(FIB (FIB (MY- J-INDEX J-INDEX)))	5	0.0714	0.0708

#	parents	xsite (Dewey Decimal Notation)	fitness	s-fitness
115)	(128,150)	0.1 & 0.1.1.2	0.0714	0.0720
130)	(4,141)	0 & 0.2	0.0050	0.0041

Note: max = 0.0833 min = 0.0010 avg = 0.0472
 nmutation = 217 ncross = 369 nedit = 366
 maxn = 0.0791 minn = 0.0009 avgn = 0.0328 (Last50)

(a) 第5, 6世代

Population report				
#	string	Generation	fitness	s-fitness
115)	(FIB (FIB (MY- J-INDEX 2)))	6	0.0714	0.0720
130)	(MY+ (FIB (MY- J-INDEX 1)) (FIB (MY+ 1 J-INDEX)))	6	0.0050	0.0041

#	parents	xsite (Dewey Decimal Notation)	fitness	s-fitness
129)	(115,130)	0.1.1 & 0.2.1	1.0000	0.1156

Note: max = 1.0000 min = 0.0010 avg = 0.0578
 nmutation = 251 ncross = 438 nedit = 432
 maxn = 0.2106 minn = 0.0009 avgn = 0.0364 (Last50)

(b) 第6, 7世代

図 3

的爆発にある。古典的な手法として GMDH (Group Method of Data Handling) が開発されている[Iva-

khnenko71]. これは、変数群から選択された適当な2変数の2次式を重回帰分析により構成し、それを新たな変数とみなして変数群に加え、目標の近似精度を得るまでこの変数の選択と生成をくりかえすという手法である。GMDHについては、選択にヒューリスティックが必要であることや、探索が極値に陥りやすいことなどの欠点が指摘されている[Tenoiro90].

筆者らは、構造的GAを用いてシステム同定問題の解決システムを構築した。GMDHで構成される変数の関係が2分木であることに注意された。たとえば、入力変数が $\{x_1, x_2, x_3, x_4\}$ で出力変数が y である関数のGMDHの結果が、

$$z_1 = G_{x_1, x_2}(x_1, x_2), \quad (1)$$

$$z_2 = G_{x_2, z_1}(x_3, z_1), \quad (2)$$

$$\bar{y} = G_{x_4, z_2}(x_4, z_2), \quad (3)$$

のときを考える。ここで z_1 と z_2 は中間変数、 \bar{y} は推論結果である。各関数 G は重回帰分析で求めた2次式である。このとき y は、入力変数を終端ノードとする木、

(NODE 1

(NODE 2

(NODE 3 (x_1) (x_2))

(x_3)

(x_4))

で表現される。ここでNODE1, NODE2, NODE3は各々 \bar{y}, z_2, z_1 に対応する。各ノードは G についての情報を保持する。

以上のように考えると、2節で述べた構造的GAの手法がGMDHによる探索にそのまま応用できる。さらにGAの選択に際してはMDLにもとづく適応度計算を採用した。これは木の記述長と当てはめ誤差のトレードオフを評価するためである。MDLのGAへの応用については決定木の学習実験などでその有効性を確認している[Higuchi93].

図4は時系列予測の結果を示す。(a)は予測すべきデータである。これはMackey-Glass微分方程式と呼ばれ、

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x^{10}(t-\tau)} - bx(t) \quad (4)$$

を満たす。ただし、 $a=0.2, b=0.1, \tau=17$ である。この式は3.5フラクタル次元のストレンジ・アトラクター

を有する[Tenorio90]. 時系列予測問題とは, 現在値を過去のデータの関数,

$$x(t) = f(x(t-1), x(t-2), \dots, x(t-M)). \quad (5)$$

として求めることである (この実験では $M=10$ とする). 以下では $x(1) \sim x(100)$ を訓練データ, $x(101) \sim x(500)$ をテストデータとした. 用いたパラメータは, 集団数60, **Gcrossover**の確率0.6, **Gmutation**の確率0.0333である. また終端ノードは $\{(1), (2), \dots, (10)\}$ である. ただし (i) は $x(t-i)$ を表わす. 図4(b)と(c)はGAによる233と1740世代での予測結果である. 世代1740で獲得された構造表現とそのノードに記録されている係数の一例を表2に示す (ただし $G_{z_1, z_2}(z_1, z_2) = a_0 + a_1 z_1 + a_2 z_2 + a_3 z_1 z_2 + a_4 z_1^2 + a_5 z_2^2$). (c)においてはテストデータにおいて完全な一致が見られることに注意されたい (平均誤差 5.06×10^{-8}). この他にパターン認識の実験などにより, 構造的GAがGMDHの探索において有効なことを確認した. なお本節の実験の詳細や結果の議論については[Iba93]を参照されたい.

5. おわりに

本稿では, GAを構造的表現に拡張した適応的手法にもとづく知識の獲得と生成のための枠組みを示し, 有効性を試すための実験について説明した.

GAの枠組みの利点は, PTYPE/GTYPEの表現を通して構造をオペレータで直接操作し, 適応選択による結果を構造的表現として得られることである. この点が分散表現を重視するコネクショニスト・アプローチとは異なり, 記号レベルの学習との統合をはかる上で有利である. 特に高次の知識表現の変換手法(representational

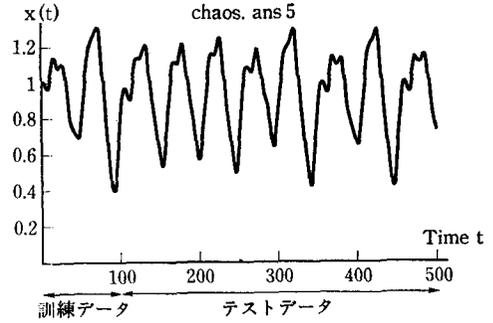


図 4 (a)

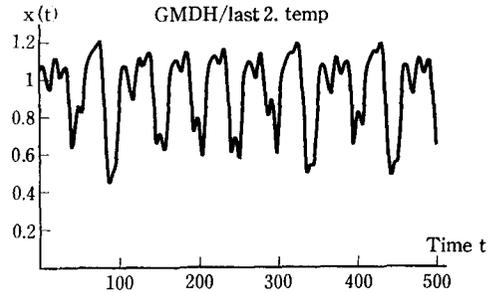


図 4 (b)

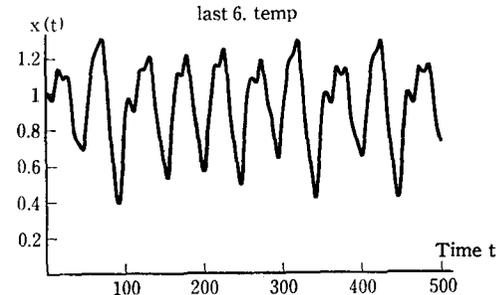


図 4 (c)

表 2 獲得された木構造

```
(NODE95239 (7)
(NODE95240
(NODE95241
(NODE95242
(NODE95243 (NODE95244 (8) (NODE95245 (8) (NODE95130 (2) (3))))
(NODE95173 (10) (NODE95174 (NODE95175 (4) (1) (5))))
(5))
(6))
(NODE95178 (NODE95179 (8) (3) (10))))
```

Node	a_0	a_1	a_2	a_3	a_4	a_5
NODE95239	0.093	0.133	0.939	-0.029	0.002	-0.009
NODE95240	-0.090	1.069	-0.051	1.000	-0.515	-0.421
NODE95179	0.286	-0.892	1.558	1.428	-0.536	-0.844

change) としての可能性を示唆し, 機械学習における「発見的論理」[Lakatos76]手法として重要であると考える。

参 考 文 献

- [伊庭90] 伊庭斉志, 情報の進化と創造的学習について, 人工知能学会基礎論研究会, SIG-F/H/K-9001-10, 1990
- [伊庭91a] 伊庭斉志, 構造的表現の適応的学習とその応用, 情報処理学会人工知能研究会, AI-76-2, 1991
- [伊庭91b] 伊庭斉志, 適応的学習に基づく知識の獲得と生成, 第5回人工知能学会全国大会, 1991
- [Davis 91] Davis, L.: Order-based genetic algorithms and the graph coloring problem, in *Handbook of genetic algorithms*, (ed. Davis, L.), Van Nostrand Reinhold, 1991
- [Ehrig87] Ehrig, H., Nagl, M., Rozenberg, G. and Rosenfeld, A.: Graph-grammars and their application to computer science, Lecture notes in computer science, vol. 291, Springer-Verlag 1987
- [Fantana90] Fontana, W.: Algorithmic chemistry in *Artificial Life II*, Addison Wesley, 1990
- [Goldberg89] Goldberg, D.E.: Genetic algorithms in search, optimization and machine learning, Wesley, 1989
- [Higuchi et al.93] Higuchi, T., Iba, H., de Garis, H. Niwa, T., Tanaka, T. and Furuya, T.: Evolvable Hardware : genetic-based generation of electronic circuitry at gate and hardware description language levels, submitted to ICGA 93, 1993
- [Iba et al.92a] Iba, H., Akiba, S., Higuchi, T. and Sato, T.: BUGS: A bug-based search strategy using genetic algorithms, In *Proc. of 2nd Parallel Problem Solving from Nature (PPSN 92)*, North-Holland, 1992
- [Iba et al.92b] Iba, H. and Sato, T.: Meta-level strategy learning for GA based an structured representation, ETL-TR92-12, In *Proc. of 2nd Pacific Rim International Conference on Artificial Intelligence (PRICAI92)*, 1992
- [Ibe et al.92c] Iba, H., and Higuchi, T.: Evolutionary Learning of Predatory Behaviors Based on Structured Classifiers, ETL-TR92-34, In *Proc. of 2nd International Conference on Simulation of Adaptive Behavior (SAB92)*, 1992
- [Iba et al.93] Iba, H., Kurita, T. de Garis, H. and Sato, T.: System identification based on structured genetic algorithms, ETL-TR93-1, submitted to ICGA93, 1993
- [Ivakhnenko71] Ivakhnenko, A. G.: Polynomial, theory of complex systems, *IEEE Tr. SMC*, vol. SMC-1, no. 4, 1971
- [Kitano90] Kitano, H.: Designing neural networks using genetic algorithms with graph generation system, *Complex systems*, vol. 4, 1990
- [Knuth67] Knuth, D. E.: The art of computer programming, vol.1. Reading, M. A : Addison-Wesley, 1967
- [Koza90] Koza, J. Genetic programming : A paradigm for genetically breeding populations of computer programs to solve problems, Report No. STAN-CS-90-1314, Dept. of Computer Science, Stanford Univ., 1990
- [Langton89] Langton, C. G.: Artificial life, In *Artificial Life*, Addison-Wesley, 1989
- [Lakatos76] Lakatos, I.: Proofs and refutation the logic of mathematical discovery, Cambridge University Press, 1976 (邦訳 : 数学的発見の論理証明と論駁, 佐々木力訳, 共立出版, 1980)
- [Tenorio et al.90] Tenorio, M. F. and Lee, W.: Selforganizing network for optimum supervised learning, *IEEE Tr. Neural Networks*, vol. 1, no. 1, 1990