

# 最適概念形成への遺伝的アルゴリズムの適用可能性の検討

高橋 良 英\*

## An Application of Genetic Algorithm to Selecting the Optimum Classification Tree in Conceptual Clustering System

Ryouei TAKAHASHI\*

### Abstract

This paper experimentally investigates an efficiency of genetic algorithm (GA) to construct the optimum binary classification tree in COBWEB, a conceptual clustering system, developed by Fisher. We study *CU* (Category Utility) used in COBWEB as an objective criterion of the hierarchical concept-representation tree. This tree classifies  $n$  objects within a certain class  $C$  into two classes  $C_1$  and  $C_2$ . One of problems of the conceptual clustering system is to select the optimum classification tree with the highest *CU* value among all the exponential order  $(2^n - 2)$  classification trees, where  $n$  is the number of objects in the class  $C$ . We report in this paper how to apply GA to obtaining the approximate solution of the optimum concept-learning tree and its experimental result.

**Key words**: learning by observation, COBWEB, genetic algorithm, category utility, information criterion, re-use of open C software

### 1. はじめに

本報告では、最適な概念学習木 [1] を、遺伝的アルゴリズム (GA) [2][3][4] により選択する手法を提案し、その有効性を実験により検証する。本検討での学習概念モデルは、Fisher が開発した概念階層モデル COBWEB [5] である。COBWEB では、概念木のあるクラスに属する標本を、属性が異なる幾つかのクラスに階層的に分類・整理する幾つかの方法がある場合、それらの方法相互間を、カテゴリ有用度 (*CU*: Category Utility) により横並びに定量的に評価し、*CU* の最も高い分け方で標本をクラス分けする。*CU* はクラス内にある標本の属性の類似性を測る尺度であり、シャノンの「情報量」を測る尺度 [6] の変形である。本研究では概念学習木の中で最も基本的な 2 分岐概念学習木の最

適構成法を研究した。クラス  $C$  に属する  $n$  個の標本を 2 つのクラスに分ける方法として  $(2^n - 2)$  通り考えられるが、COBWEB による概念学習の課題の 1 つは、 $(2^n - 2)$  通りあるクラス分けから最も有効なクラス分けの手法を探すことである。本検討では、これ等のクラス分けの中から、最もカテゴリ有用度の高いクラス分けの方法を近似的に選択する手法として、遺伝的アルゴリズム (GA) により選択する手法が有効であることを小実験 (4 種類の標本) により検証したので報告する。

### 2. 背景

ソフトウェア品質判別木によるソフトウェア品質評価手法 [7] においては、「例題による概念学習法 (learning from examples)」の手法の 1 つである Quinlan が開発した ID3 [8] の有効性が示されている。ID3 の手法は標本属性から

平成 15 年 12 月 19 日受理

\* システム情報工学科・教授

目的クラスが+ (品質が良い) か- (品質が悪い) を推論する方法であるが、標本が追加される度に判別木を再構成しなくてはならない等の問題がある。本検討では追加標本があっても、部分的な改良で概念学習木を再構成できる「観察による概念学習法 (learning by observation)」の1つである Fisher が開発した COBWEB によりソフトウェア品質の良し悪しを予測制御するソフトウェア品質評価システムについて検討することとした。

ソフトウェア品質評価する時の課題は、評価の精度を向上させることであり、このためには、観察対象システムの特徴 (システムを構成する標本の属性、標本は「規模」, 「複雑さ」, 「開発期間」, 「開発 OS」, 「マニュアル」, 設計書の整備状況」等の様々な属性を持つ) を考慮して最適な概念学習木を形成する必要がある。本検討では、最適な概念2分岐モデルを COBWEB の手法に基づき構築する方法について検討した。

### 3. COBWEB における概念学習木の評価法

#### 3.1 COBWEB における概念学習法

Fisher が開発した COBWEB [5] は、標本属性が似たもの同士を合理的にグループ化し最適な概念学習木を構成する「観察による学習法」方法の1つである。グループされた集合をクラスと呼び、各クラスの特徴を表現する用語が「概念」である。別名、教師 (目的変数) なし学習と呼ばれる。尚、「観察による学習法」には COBWEB 以外に、Feigenbaum の EPAM 等の方法が知られている [9]。COBWEB ではカテゴリ有用度 ( $CU$ : Category Utility) [1] により分類の有用度を評価する。属性未定の新しい標本  $\alpha$  が発生した時、生成した概念学習木に基づいて、標本  $\alpha$  が所属するクラスと未定の属性を推論する。概念木の各ノードは分類された「概念」を示す。木のエッジは「概念」を更に詳細な「概念」に分割することを示す。子クラスは親クラスの「概念」を共通属性として継承するので、親

クラスと子クラスの関係は is a 関係または特化/汎化関係と呼ばれる [10]。

#### 3.2 概念木の評価指標：カテゴリ有用度 $CU$

親クラス  $C_0$  に属する標本を子クラス  $C_1, C_2, \dots, C_n$  の  $n$  個のクラスに分けてできる概念木を客観的に横並びに評価する評価指標がカテゴリ有用度  $CU$  であり、以下の (1) から (4) に示すように段階的に定義する。 $CU$  は「子クラスの総合属性類似度の平均値の親クラスの総合属性類似度からの向上度」を子クラス数で正規化した尺度である。

- (1) クラス  $C$  における属性  $A$  の属性類似度:  $EU(C; A)$

クラス  $C$  を構成する標本が属性の似たもの同士が集まっているか否かは、標本を特徴づける属性  $A$  について、同一の属性値を持つか否かで判断する。属性  $A$  が属性値  $V_1, V_2, \dots, V_l$  の  $l$  個の属性を持つ場合、以下の  $EU(A; C)$  で表現する。

$$EU(A; C) = \sum_{j=1}^l P(A = V_j | C)^2$$

属性類似度:  $EU(C; A)$  の意味

有用度は概念木の定量的評価尺度であり、その値が大きければ有用度は高くなり、1 が最大値となる。例えばある属性  $A$  が2つの値  $V_0$  と  $V_1$  を持ち、各値の確率を  $p$  と  $(1-p)$  とすると、属性  $A$  のクラス  $C$  内での有用度  $EU(C|A)$  は  $EU(C|A) = P(A = V_0 | C)^2 + P(A = V_1 | C)^2 = p^2 + (1-p)^2 = 2(p-1/2)^2 + 1/2$  となり、 $P=1/2$  で最小値  $1/2$ ,  $p=0, 1$  で最大値 1 をとる (図1参照のこと)。図1の形から、 $EU(A; C)$  は、良く知られた以下の情報量を測る尺度「シャノンのエントロピー」 $H(A; C)$  [6] の正負の符号を反対にした関数と同じ役割を果たしていることがわかる。

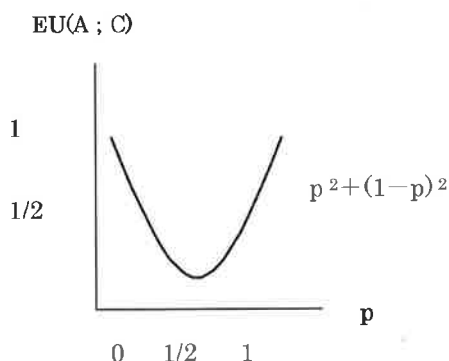


図 1. 2つの値  $V_0$  と  $V_1$  を持つ属性  $A$  のクラス  $C$  内における類似度  $EU(C; A)$  と、属性  $A$  が  $V_0$  を取る確率  $P(A = V_0|C) = p$  との関係

$$H(A; C) = -\sum_{j=1}^l P(A = V_j|C) \log_2 P(A = V_j|C)$$

(2) クラス  $C$  における総合属性類似度：  
 $EU(C)$

標本が  $m$  個の属性  $A_1, A_2, \dots, A_m$  個を持ち、属性  $A_i$  は  $V_{i1}, V_{i2}, \dots, V_{il}$  の実現値を持つとする。

この時、クラス  $C$  を構成する標本は、以下の総合属性類似度  $EU(C)$  で評価する。

$$EU(C) = \sum_{i=1}^m EU(A_i; C) = \sum_{i=1}^m \sum_{j=1}^{l_i} P(A_i = V_{ij}|C)^2$$

(3) 総合属性類似度の向上度： $CU_0(C_0 \rightarrow C_1, C_2, \dots, C_n; A_1, A_2, \dots, A_m)$

親クラス  $C_0$  に属する標本を子クラス  $C_1, C_2, \dots, C_n$  に分ける場合の総合属性類似度の向上度  $CU_0$  を以下で定義する。

$$\begin{aligned} CU_0(C_0 \rightarrow C_1, C_2, \dots, C_n; A_1, A_2, \dots, A_m) \\ &= \text{子クラスの総合属性類似性の平均値} - \text{親クラス属性類似度} = EU(C_1, C_2, \dots, C_n) - EU(C_0) \\ &= \sum_{i=1}^n (\text{各クラス } C_i \text{ に属する標本の割合 } (P(C_i)) \times \text{各クラス内での総合属性類似度}) - (\text{親クラス属性類似度}) \\ &= \sum_{k=1}^n p(C_k) E(C_k) - E(C_0) \end{aligned}$$

$$\begin{aligned} &= \sum_{k=1}^n p(C_k) \sum_{i=1}^m E(A_i; C_k) - \sum_{i=1}^m E(A_i; C_0) \\ &= \sum_{k=1}^n p(C_k) \sum_{i=1}^m \sum_{j=1}^{l_i} P(A_i = V_{ij}|C_k)^2 \\ &\quad - \sum_{i=1}^m \sum_{j=1}^{l_i} P(A_i = V_{ij}|C_0)^2 \end{aligned}$$

(4) カテゴリ有用度  $CU$

一般に、分類するクラス数  $n$  が多くなれば、属性類似性の向上度  $CU_0$  は大きな値となる。従って、分類するクラス数  $n$  が異なる2つの分類方法を、 $CU_0$  で比較できない。

クラス数の違う2つの分類方法は、以下のカテゴリ有用度  $CU$  で相互比較評価する。 $CU$  は子クラスの属性類似度の平均向上度のことで、 $CU_0$  を子クラス数  $n$  で割った（この操作を正規化と呼ぶ）値であり、以下の式で定義する。

$$CU(C_0 \rightarrow C_1, C_2, \dots, C_n; A_1, A_2, \dots, A_m) = CU_0/n.$$

### 3.3 カテゴリ有用度 $CU$ の計算例

クラス  $C_0$  に属する100個の標本は「大きさ」( $=A_1 = \{\text{中}, \text{小}\}$ )と「色」( $=A_2 = \{\text{白}, \text{黒}\}$ )の各属性で特徴づけられるとする。この時、クラス  $C_0$  に属する標本を、2つのクラス  $C_1$  と  $C_2$  に分ける。クラス  $C_1$  は  $A_1 = \text{中}$ ,  $A_2 = \text{白}$  の属性値を取る70個の標本から成り、クラス  $C_2$  は  $A_1 = \text{小}$ ,  $A_2 = \text{黒}$  の各属性値を取る30個の標本から成りとする。この分類モデルのカテゴリ有用度  $CU$  は、 $CU = (P(C_1) * EU(C_1) + P(C_2) * EU(C_2) - EU(C_0)) / 2 = \{[0.7 * (1^2 + 0^2 + 12 + 0^2) + 0.3 * (0^2 + 1^2 + 0^2 + 1^2)] - \{(0.7)^2 + (0.3)^2\} + \{(0.7)^2 + (0.3)^2\} / 2 = (2 - 1.16) / 2 = 0.84 / 2 = 0.42$  と計算される。

## 4. 概念学習木形成の問題と対策

概念学習木形成の課題を、全く新たに概念を構成しようとした場合に生じる課題[11]と、既存の概念を再構成する時に生じる課題に分け

る。前者の課題の1つとして、「親クラス  $C_0$  に属する  $x$  個の標本を子クラス  $C_1, C_2, \dots, C_n$  に分ける方法は、一般に  $n^x - 2$  個ある。この中から最もカテゴリ有用度  $CU$  の高いモデルをどのように選択するか」が概念学習木形成時の課題となる。このように、ある集合を構成する標本の数  $x$  が増えればそれに対応して指数オーダーで増えていくあるクラス分けの中から最適な解を探索する問題は計算機でも実効上解くことが困難な  $NP$  完全問題として知られている[12]。巡回セールスマン問題のような  $NP$  完全問題については、その近似解を遺伝的アルゴリズム (GA) による手法で求めることが有効であることが知られている[2][4]。このため、本検討においても、遺伝的アルゴリズム (GA) を適用して最適な二分岐概念学習木の近似解を選択するモデルについて検討することとした。

## 5. 遺伝的アルゴリズム (GA) による最適な概念木の形成法

### 5.1 概念学習木の遺伝子型表現法

遺伝的アルゴリズムでは環境に適合した個体を残すため、複数の初期個体を準備すると共に、集団を構成する個体の遺伝子に対して交叉・突然変異を行うことで局所的最適解に陥らない工夫をしている。以下に本検討での「個体」の考え方、「各個体の環境への適応度計算法」、「自然淘汰の実現方法」[2][4]を整理する。

#### (1) 個体とその表現法

##### 個体

個体は親クラス  $C_0$  を構成する標本をどのクラスに分類するのかその分類方法を示している。

##### 遺伝子型 (図2)

個体は、親クラス  $C_0$  を構成する標本数分の遺伝子から構成される。遺伝子は1か0の値をとる。

標本①の 所属群	標本②の 所属群	標本③の 所属群	標本④の 所属群
(1/0)	(1/0)	(1/0)	(1/0)

図2. 概念学習木の遺伝子型表現

個体を構成する個々の遺伝子は、親クラス  $C_0$  を構成する個々の標本番号を示している。標本が発生した順番に標本が並べられている。遺伝子の取る値は、標本が所属する子クラス番号  $C_i$  ( $i=1, 2, \dots, n$ ) を識別している。本実験では2分岐木モデル ( $n=2$ ) を研究しており、「遺伝子が値1を取った時はクラス  $C_1$  に標本が属すること」、「遺伝子が値0を取った時はクラス  $C_2$  に標本が属すること」を示している。

##### 表現型

個体は親クラス  $C_0$  から生成される2分岐モデルを示す。例えば、遺伝子型が1100の場合、クラス  $C_1$  の標本は標本番号①②、クラス  $C_2$  の標本は標本番号③④から構成されていることを示す。

#### (2) 環境への適応度計算

COBWEBでのカテゴリ有用度  $CU$  で個体の環境への適応度を計算する。

#### (3) 自然淘汰

「自然淘汰」や「進化」という言葉で表現される現象は、集団を構成する個体が世代交代をすることであり、これを、「両親を選択する」、「両親が遺伝子を交叉させ、2人の子供を出産する。子供と親が交代する（子供を残し親は死滅する）」、「子供の遺伝子に突然変異が起こる」という考え方で行う。これ等の現象は確率的に実行される。

## 5.2 遺伝的アルゴリズム (GA) プログラムの再利用

遺伝的アルゴリズムの基本的考え方は研究者

によって大きく変わらない [2][3][4]。本検討での GA アルゴリズムは、文献 [2]「平野廣美著、遺伝的アルゴリズムプログラミング」で紹介されている手法とその公開 C ソースコードのアルゴリズム実現部(peaks10.c)を、「概念生成」向けに修正して実現している。5.3 で述べられるアルゴリズムのうち、③(i) 交叉手法を「二点交叉手法」から「一点交叉手法」に修正、② 親選択法 (ii) 適応度計算を「山登り評価関数」から「カテゴリ有用度」に変更して実現した。その他は再利用して実現した。

### 5.3 遺伝的アルゴリズム

以下に GA のアルゴリズムを整理する。

#### ① 初期設定

充分な個体を遺伝子プールに発生させる（プログラム起動条件で与える）。個体を構成する各遺伝子の初期値は、一様乱数で決定する。乱数は、0 から 1 の間の数値を一様にとる。乱数が 0.5 以上の場合 1, 0.5 以下の場合 0 とする。予備遺伝子プールを準備しておく。

#### ② 親の選択

遺伝子プールの中から、以下の選択基準で親を 2 個体決める。

(i) 遺伝子プールに並んでいる順番に各個体の適用度を計算する。

(ii) これを足し合わせて適用度総和を求める。

(iii) 適用度総和に対してある割合を満たした時の個体を親として選択する。この割合は試行の度に変化し、その値は乱数で決める。

#### ③ 遺伝的操作

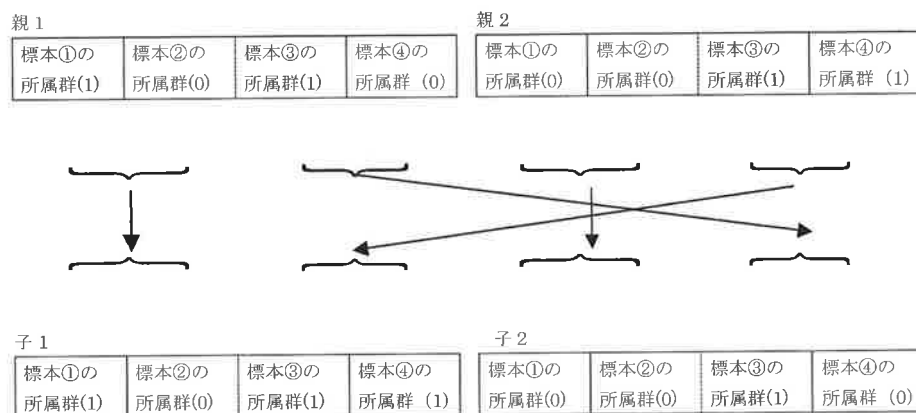
両親の間で交叉 (Crossover) をさせ、子供 2 個体を生成する。突然変異 (Mutation) により、個体の遺伝子型を変化させる。これらの操作を行う場合には、それぞれ交叉確率  $PC$ 、突然変異確率  $Pm$  で行う。 $PC$ 、 $Pm$  はプログラム起動条件で与える。

一様乱数を発生させその値が  $PC$  または  $Pm$  を越えた場合に交叉と突然変異を行う。

(i) 一点交叉：遺伝子の部分的な交換を 2 箇所で行う。どの遺伝子の位置で交叉を行うかを乱数で決定する（図 3 参照）。

(ii) 突然変異：ある遺伝子の on, off 状態を反転させる。どの遺伝子を突然変異させるかは乱数で決める（図 4 参照）。

④ 新たに産まれた個体を予備遺伝子プールへ格納する。この予備遺伝子プールが、親の個体数と同じになるまで、② から繰り返す。同じ



<注意>図は、破線...で示した、2ビット目と3ビット目の境目で一点交叉が起こった事例である。

図 3. 一点交叉による概念学習木の生成事例

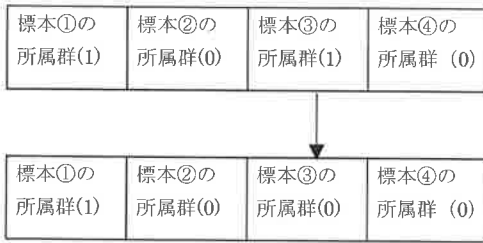


図4. 突然変異による概念学習木の生成事例

になったら⑤へ行く。

⑤ 予備遺伝子プールから遺伝子プールへ移す。このとき旧世代はすべて、新世代に置き代わる。

⑥ 終了処理(集団の評価)…終了条件を調べて、終了していなければ、②から繰り返す。終了条件は、通常繰り返し回数をあらかじめプログラム起動条件として与えておく。

#### 5.4 GA手法の定性的評価

上記5.3で述べた手続きにおいて環境への適応度の高い個体が淘汰されて残る理由について以下に整理する。

(1) GA手法では、5.3節①の手続きで複数の個体を最適解候補として初期状態で選択できるので、 $CU$ の最も大きな概念学習木を選択できることの大きな根拠となっていること [2] [3] [4]。

(2) 「定義長が短くてオーダが小さな、適応度の高い個体を他の個体と遺伝子交叉させれば適応度の高い個体が生成される」という仮説は、積み木仮説 [3] と呼ばれており交叉により最適化が進むことの根拠となっている。概念学習木の適応度はカテゴリ  $CU$  である。標本属性の同じ標本が異なるクラスにグループ分けされたり、属性の異なるクラスにグループ分けされると必ず  $CU$  が下がる。また、親の個体の定義長やオーダは、個体の長さとも一致している。従って、5.3節②の手続きより、親として選択される確率の高い親の  $CU$  は高く、そのような親の

選択交叉で産まれる2人の子の  $CU$  がいずれも低い可能性はなく、積み木仮説の条件を満たしている。

(3) 突然変異は遺伝子型を強制的に破壊し、局所最適解に陥らせないための対策となっている。

## 6. 実験例

### (1) 標本空間(表1)

表1に示す4つの標本①②③④をその要素とする集合(クラス  $C_0$ )を、2つのクラス  $C_1$  と  $C_2$  に分ける方法は  $2^4 - 2 = 14$  通りある。各分類モデルについて、カテゴリ有用度  $CU$  を計算し、「いずれの分類方法が、より属性の似たもの同士で標本のクラス分けをしているか」を比較評価した。GA解をCプログラミングで求め、机上評価結果と一致しているかを調べた。

#### <留意事項>

「遺伝子型の長さ」をできる限り短くするため、 $C_0$ を構成する属性の異なる標本の構成比は最大公約数が1になるように予め調整する。本事例では属性値の異なる4つのグループに構成標本を分けた時、各グループを構成している標本の標本数の構成比が1:1:1:1の事例である。

### (2) 机上評価による最適解の選択(表2)

14個の学習概念木の  $CU$  計算結果を、表2に示す。最適概念学習木の木構造を図5に、 $CU$ の内訳を表3に示す。この結果、個体③(遺伝子

表1. 標本内訳

標本ID	属性		
	$A_1$ : 大きさ	$A_2$ : 形	$A_3$ : 色
①	小 (S)	四角 (S)	白 (W)
②	小 (S)	四角 (S)	灰 (G)
③	中 (M)	丸 (C)	黒 (B)
④	大 (L)	丸 (C)	黒 (B)

表 2. 最適概念学習木の選択

◎最適モデル (CU 最大モデル)

個体 ID	遺伝子型	表現型		CU: カテゴリ有用度	EU(C <sub>1</sub> , C <sub>2</sub> ): クラス内総合属性類似度の平均値	P(C <sub>1</sub> ): 子クラス C <sub>1</sub> を構成する標本の割合	EU(C <sub>1</sub> ): 子クラス C <sub>1</sub> での総合属性類似度	P(C <sub>2</sub> ): 子クラス C <sub>2</sub> を構成する標本の割合	EU(C <sub>2</sub> ): 子クラス C <sub>2</sub> での総合属性類似度
		クラス C <sub>1</sub> を構成する標本	クラス C <sub>2</sub> を構成する標本						
1	1110	①②③	④	0.292	1.83	0.75	1.44	0.25	3
2	1101	①②④	③	0.292	1.83	0.75	1.44	0.25	3
◎ 3	1100	①②	③④	0.625	2.50	0.5	2.5	0.5	2.5
4	1011	①③④	②	0.292	1.83	0.75	1.44	0.25	3
5	1010	①③	②④	0.125	1.50	0.5	1.50	0.5	1.5
6	1001	①④	②③	0.125	1.50	0.5	1.50	0.5	1.5
7	1000	①	②③④	0.292	1.83	0.25	3	0.75	1.44
8	0111	②③④	①	0.292	1.83	0.75	1.44	0.25	3
9	0110	②③	①④	0.125	1.50	0.5	1.50	0.5	1.5
10	0101	②④	①③	0.125	1.50	0.5	1.50	0.5	1.5
11	0100	②	①③④	0.292	1.83	0.75	1.44	0.25	3
◎12	0011	③④	①②	0.625	2.50	0.5	2.5	0.5	2.5
13	0010	③	①②④	0.292	1.83	0.25	3	0.75	1.44
14	0001	④	①②③	0.292	1.83	0.25	3	0.75	1.44

表 3. 最適概念学習木のカテゴリ有用度 CU の内訳

クラス			C <sub>0</sub>	C <sub>1</sub>	C <sub>2</sub>	CU
P(Ci) : 標本の重み付け (i=0,1,2)			1	0.5	0.5	$CU = CU_0/2 = (EU(C_1, C_2) - EU(C_0))/2 = ((EU(C_1) * P(C_1) + EU(C_2) * P(C_2)) - EU(C_0))/2 = (2.5 * 0.5 + 2.5 * 0.5 - 1.25)/2 = 0.625$
属性	A1	Small (S)	0.25 (=0.5 <sup>2</sup> )	1 (=1 <sup>2</sup> )	0	
		Medium (M)	0.0625 (=0.25 <sup>2</sup> )	0	0.25 (=0.5 <sup>2</sup> )	
		Large (L)	0.0625 (=0.25 <sup>2</sup> )	0	0.25 (=0.5 <sup>2</sup> )	
	A2	Square (S)	0.25 (=0.5 <sup>2</sup> )	1 (=1 <sup>2</sup> )	0	
		Circle (C)	0.25 (=0.5 <sup>2</sup> )	0	1 (=1 <sup>2</sup> )	
	A3	White (W)	0.0625 (=0.25 <sup>2</sup> )	0.25 (=0.5 <sup>2</sup> )	0	
		Gray (G)	0.0625 (=0.25 <sup>2</sup> )	0.25 (=0.5 <sup>2</sup> )	0	
		Black (B)	0.25 (=0.5 <sup>2</sup> )	0	1 (=1 <sup>2</sup> )	
クラス内類似性度 EU (Ci)			1.25	2.5	2.5	

型 1100) と個体 ⑫ (遺伝子型 0011) が最も CU が高く共に 0.625 であった。

2 通りの中から最も CU が高いモデル ③ を GA プログラムは選んだ。

GA 実行条件:

(3) GA 適用結果

以下の GA 実行条件にて, 4 世代目で, 全 2<sup>4</sup> -

●人口数=3,

●乱数種=1,

● 交叉率=0.95, 突然変異率=0.1;

#### <留意事項>

##### ① 人口数

二分岐概念学習モデルのあるクラス分けについて構成する標本の遺伝子のオンオフを全て逆にしたクラス分けのカテゴリ有用度  $CU$  は変わらないので同一のクラス分けと考えられる。このため、実験ではこれを同一クラスとして扱い、人口数  $[(2^4 - 2)/2^2] = 3$  のプールで各世代の個体を管理するモデルを検討した。

##### ② 一様乱数発生条件

本検討では、Kernighan-Richie 著、プログラミング言語 C [13] でのライブラリ関数 `srand`, `rand` 関数を利用して乱数を発生させている。`rand` 関数は、0 から  $2^{15}$  の間の数値を一様に発生させる。このリターン値を  $2^{15}$  で割って、0 から 1 の間の一様乱数を得ている。乱数を発生させる発生の初期値は `srand` のパラメータで与えており、このパラメータは乱数種と呼ばれる。乱数種により最適解に到達するまでの世代数が異なってくることが実験からわかっている。乱数種はプログラム起動条件として与える。

#### (4) プログラム実行結果リスト:

図 6 にプログラム実行結果を示す。図にて、`avg_fit` は平均適応度、`max_fit` は最大適応度、`maxId=1 0011` はプール番号 1 番の個体が最も適応度が高く遺伝子型が 0011 であることを示す。リストから、(3) で机上で求めた机上の最適解と GA が選択した最適解が一致するのは 4 世代目であることが読み取れる。

##### (5) 適応度の推移 (図 7)

世代数と個体の平均適応度の推移を図 7 に示す。図は、「(3) の GA 適用条件下で、81 世代目に全ての個体が最適解に到達すること」、「4 世代目に、最適モデルを選択できてきていること」、「122 世代以降 163 世代のモデルは、その平均適

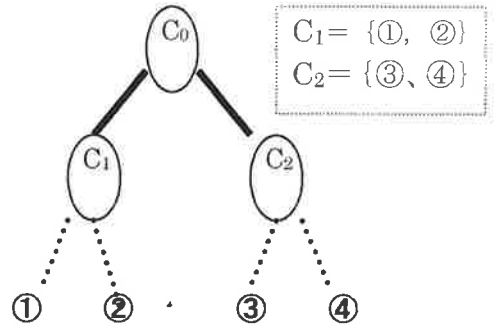


図 5. 最適モデル (モデル 12) の概念学習木

#### リスト

初期状態: `avg_fit=0.083, max_fit=0.125, maxId=1 0101`  
 第 1 世代: `avg_fit=0.125, max_fit=0.125, maxId=2 0101`  
 第 2 世代: `avg_fit=0.125, max_fit=0.125, maxId=2 0101`  
 第 3 世代: `avg_fit=0.125, max_fit=0.125, maxId=2 0101`  
 第 4 世代: `avg_fit=0.347, max_fit=0.625, maxId=0 0011`  
 第 5 世代: `avg_fit=0.236, max_fit=0.242, maxId=2 0010`

図 5. 最適個体を選択するまでの GA のプロセス

応度と最大適応度いずれも、最適解モデルの適応度またはその近傍値をとり、安定状態にあること」を示している。

(6) ソースコードの全 C ソースコード行数の再利用率は、コメント行と改行を除いて、命令文+データ宣言文約 400 行中、約 40% であった。尚、コメント行と改行の全 C ソースコード行数に占める割合は約 40% であった。

## 7. 結 論

親クラスの標本を 2 つの子クラスに分ける方法として  $2^n - 2$  通りがあるが、この中から、最もカテゴリ有用度の高いクラス分けの方法を近似的に選択する手法として、遺伝的アルゴリズム (GA) により選択する手法を検討しその有効性を、小実験により検証した。

今後の課題は、(1) 概念学習法について…

- ① 2 分岐モデルから  $n$  分岐モデルへの拡張、
- ② 標本追加時の概念木再構築法、③ 追加標本



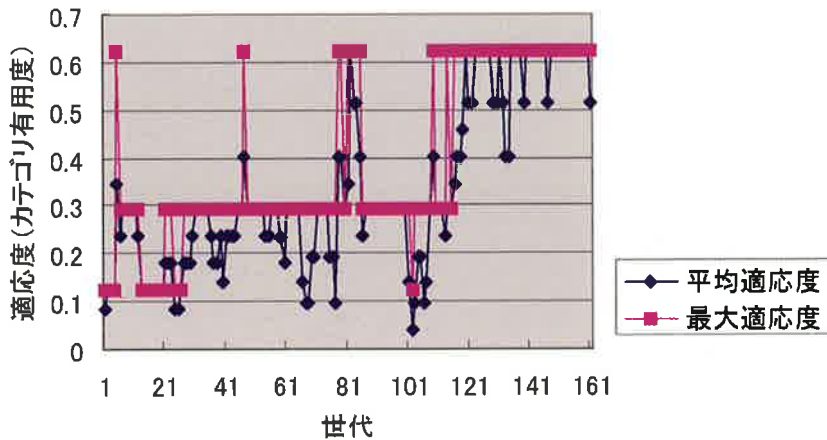


図7. 平均適合度・最大適合度の推移状況

の未定属性を既存の概念木から推論する方式の評価法，④ シャノンの情報量によるモデル相互評価法，(2) GA について… ① 一点交叉法と二点交叉法の比較，② ルーレット方式による両親の選択方式とランダム標本の中から両親を選択する方式の比較，③ 予備プールによる個体の一括淘汰方式と，1 プールによる個体の逐次淘汰方式の比較，④ 「環境への適応度」が最も低い個体と産んだ子供を交代させる方法と両親と産んだ子供を交代させる方法の比較，⑤ 乱数種が最適解到達世代数に与える影響についての分析，⑥ 初期人口数が最適適応度個体を選択するまでの世代数と平均適応度が適応度最大となる個体に満たされる世代数に与える影響の分析，⑦ 標本数の拡大，(4) ソフトウェア品質評価モデルへの応用である。

#### 参考文献

- [1] 廣田薫編著，知能工学概論，pp. 43-62，昭晃堂，1996.
- [2] 平野広美著，応用例でわかる遺伝的アルゴリズムプログラミング，pp. 17-28，pp. 232-pp. 238，パーソナルメディア社，1995.
- [3] メラニーミッチェル著，伊庭斉志監訳，本堂直

- 浩，伊藤拓也，丹羽竜哉，高島一哉，野添敏秀訳，遺伝的アルゴリズムの方法，pp. 34-38，東京電気大学出版局，1997.
- [4] 伊庭斉志著，遺伝的アルゴリズムの基礎，pp. 28-36，東京電気大学出版局，1994.
- [5] Fisher, D.H., "Knowledge Acquisition via Incremental Conceptual Clustering", Machine Learning, Vol. 1.2, 103-138, 1987.
- [6] 磯道義典，情報理論，pp. 6-12，コロナ社，1981.
- [7] 高橋良英，村岡洋一，中村行宏，"ソフトウェア品質分類木の生成・評価方法，"電子情報通信学会論文誌 (D-I)，Vol. J81-D-I, No. 4, pp. 393-404, April 1998.
- [8] Quinlan, J.R., "Learning Efficient Classification Procedures and Application to Chess End Games", Michalski, R.S., Carbonell, J.G., and Mitchell, T.M. (Eds.). Machine Learning: An Artificial Intelligence Approach, San Mateo, CA: Morgan Kaufmann, pp. 463-482, 1983.
- [9] Feigenbaum, E., "The Simulation of Verbal Learning Behavior", Proc. of the western Joint Computer Conf., pp. 121-132. 1990.
- [10] 青木淳，オブジェクト指向システム分析設計入門，pp. 65-99，ソフトリサーチセンタ，1993.
- [11] 市川伸一，考えることの科学，中公新書，1997.
- [12] 岩間一雄，アルゴリズム入門，pp. 119-152，昭晃堂，2001.
- [13] Kernighan-Richie 著，石田晴久訳，プログラミング言語 C (ANSI 規格準拠)，pp. 56-57，共立出版，1989.