獲得した情報を用いる遺伝的ネットワークプログラミングによるデータマイニング

嶋田 香† 平澤 宏太郎† 古月 敬之†

遺伝的ネットワークプログラミング（Genetic Network Programming，GNP）を用いた興味深い相関ルールの抽出法を提案する．統計学で用いられる χ^2 値を指標の一部とした興味深い相関ルールを進化論的計算手法によって抽出する．相関ルールの指標は GNP の構造的な特徴を利用して算出される．ルール抽出は世代継続的に行われるため抽出された相関ルールはライブラリに蓄積される．抽出された相関ルールに関する情報は，抽出を継続中の GNP の個体評価および進化操作時に用いられる．したがって，本手法は通常の進化論的計算手法とは進化の方法が異なる．シミュレーション結果から，本手法が興味深い相関ルールの抽出を効率的に行うことが示された．

Data Mining Using Genetic Network Programming with the Use of Acquired Information

Kaoru Shimada,† Kotaro Hirasawa† and Takayuki Furuzuki†

A method of association rule mining using Genetic Network Programming (GNP) is proposed to improve the performance of rule extraction. The proposed system evolves itself by an evolutionary method and measures the significance of the association via the chi-squared test using GNP. Extracted association rules are stored in a pool all together through generations in order to find new important rules. These rules are reflected in genetic operators as acquired information. Therefore, the proposed method is fundamentally different from all other evolutionary methods in its evolutionary way. In this paper, we describe the algorithm capable of finding the important association rules and present some experimental results.

1. はじめに

情報化社会においては日々膨大なデータが産出されており，その効果的な運用が重要なテーマとなっている．経営戦略の決定や科学的な発見につながる知見の獲得を目的として，データマイニング[1]が注目され，決定木や相関ルール，クラスタ解析といった手法の効率化に関する多くの方法が提案されている．このうち相関ルール[2]は，関係データベースにおいて属性（アイテム）間にみられる関係に注目したものであり，「$X \Rightarrow Y$」の形で表現され，X，Y の部分に条件を満たす属性が入る．これは，あるレコードが X を満たせば Y も満たすというような事実を表現しようとするものである．

Agrawal ら[3]によって提案された相関ルール抽出のアルゴリズムによれば，サポート（レコードの出現頻度）と確信度（ルールを構成する属性の結び付きの強度）を指標として，ユーザの指定した基準を満たす相関ルールを導出することができる．しかしながら，属性数が増えると計算量が指数関数的に増えること，計算量を抑えようとすると重要なルールを失うなどの問題がある．また，アイテムの出現頻度が高くなると抽出された相関ルールの価値が明確でなくなるという問題もある．Brin ら[4]はサポートと確信度の両者が，どの程度の値を満たせば興味深いルールであると判断できるかという客観的な基準は存在せず，これらは相関ルールの重要度の判断には適当でないと指摘し，統計学で用いられる χ^2 値を指標とする方法に注目した．しかし，χ^2 値を算出する際に必要となる相関ルールの結論部のサポートをどのように効率良く扱うかという問題があり，相関ルールを構成する属性数が増加した場合には対応が困難となる．

これらの問題点を改善すべく，進化論的計算手法の 1 つである遺伝的ネットワークプログラミング（Genetic Network Programming，GNP）[5)~8)]を用いた相関ルールの抽出方法[9),10)]が提案されている．GNP は，相関ルール抽出への応用にあたって有益と考えら

† 早稲田大学大学院情報生産システム研究科
Graduate School of Information, Production and Systems, Waseda University

れる次のような特徴を持っている．

- グラフ上のノードの再利用/共有が可能であり，探索空間を有効に構成できる．
- GNP における 1 つのノードが相関ルールの 1 つの属性を表すものと考えるとノードの連結とその遷移を，相関ルールと対応させることができる．
- 興味深い相関ルールを抽出することができた場合，そのルールの一部を遺伝的操作によって変化させることでさらなるルールの探索が可能である．

また，この手法には，次のような特徴がある．

- 相関ルールのサポートと確信度を GNP を用いて直接算出することができる．統計学で用いられる χ^2 値の算出についても GNP の構造上の特性を活かして直接行うことができる．アプリオリ法のように頻出アイテム集合のすべてのサポートを記憶しておく必要はないため，各レコードに多くのアイテムが含まれるなどの特徴を持った密なデータベース[11] からのルール抽出が可能になる．
- ユーザが定義した興味深さの指標に基づいた相関ルールを抽出する．興味深さの定義には相関ルールの χ^2 値やルールを構成する属性数を含めることができる．また，GNP の適合度は，興味深さの指標を反映した関数を設定することができる．
- 通常の進化論的計算手法にみられるような最良の個体（解）を発見することを目的とはせずに，GNP の各世代の個体が抽出した興味深い相関ルールを相関ルールライブラリに蓄積する方式である．すべてのルールを抽出することはできないが，指定した世代終了時点で蓄積されている興味深い相関ルールをユーザは活用することができる．

本論文では，上記手法の抽出効率の向上とより多くの属性からなるデータベースへの対応を目的として，次のような拡張を提案する．

- GNP の判定ノード関数の数を可変化して相関ルールの抽出ができるように拡張する．文献 9)，10) で提案された手法ではデータベースの属性に対応した GNP の判定ノード関数の数を全世代において固定して利用している．このために扱うことのできる属性数に制限があったが，提案手法ではより多くの属性で構成されるデータベースからの相関ルール抽出が可能になる．
- 進化時の遺伝的操作は，ノードの接続先の変更によるものだけでなく，判定ノード関数の変更も追加する．この結果，興味深い相関ルールの候補がより柔軟に効果的に作成されることが期待できる．
- 稼働中の GNP において，ライブラリに蓄積されている相関ルールに関する情報を，以後の世代における新規の相関ルール抽出の効率化にも活用する．文献 9)，10) では，ライブラリの情報は，抽出された相関ルールの新規性の判定にのみ用いられている．したがって，本論文で扱う手法は GNP が自ら獲得した情報を GNP 自身の進化操作に反映するという特徴を持っている．

表 1 データベースの例
Table 1 An example of database.

TID	A	B	C	D
1	1	0	1	0
2	0	1	1	1
3	1	1	1	1
4	0	1	0	1

2. 相関ルール

本章では，データマイニングの手法の 1 つである相関ルール（association rule）[1),2)] について概説する．

関係データベースにおいて，1 つのレコードがある条件を満たすか否かの判断をテストとよぶ．相関ルールとは，$X \Rightarrow Y$ と記述できる関係であり，「あるレコードが X を満たせば Y も満たす」という事実を表現するものである．X は相関ルールの前提部（antecedent），Y は結論部（consequent）とよばれ，データベース中で，X と Y の両方を満たすレコードの割合を相関ルールのサポート（support），X を含むレコードのうち Y を含む割合を確信度（confidence）とよぶ．たとえば **表 1** において，$(A = 1) \wedge (C = 1) \Rightarrow (D = 1)$ のサポートは 0.25，確信度は 0.5 であり，$(B = 1) \wedge (C = 1) \Rightarrow (D = 1)$ のサポートは 0.5，確信度は 1 である．

Agrawal らは，サポートと確信度を相関ルールの重要度を評価する指標として用い，サポートと確信度がしきい値以上である相関ルールを重要なルールとして抽出するアプリオリ法（apriori algorithm）[3)] を提唱している．この方法は，顧客の商品購入に関する相関ルールを抽出するマーケット・バスケット分析に応用されている．アプリオリ法は，サポートと確信度の条件を満たす頻出アイテム集合をすべて抽出するため，この集合の数が巨大になる場合にはコンピュータの処理能力の問題が生じてくる可能性もある．アプリオリ法の効率を向上させるために，ハッシュを用いるなど多くの手法[12)] が提案されている．さらに，否定の属性を含む相関ルールの抽出[13)] やレコード間にまたがる相関ルールの抽出[14)] を行う拡張も提唱されている．また，相関ルールを密なデータベースから抽出する手

表 2 分割表
Table 2 The contingency of X and Y.

	Y	$\neg Y$	$\sum_{row}$
X	Nxy Nz	$N(x-xy)$ $N(x-z)$	Nx
$\neg X$	$N(y-xy)$ $N(y-z)$	$N(1-x-y+xy)$ $N(1-x-y+z)$	$N(1-x)$
$\sum_{col}$	Ny	$N(1-y)$	N

法についても関心が高まっている．密なデータベースとは，各レコードに含まれるアイテムの数が多いこと，出現頻度の高いアイテムが存在することなどの特徴を持つものであり，国勢調査やアンケート調査などがこれにあたる．密なデータベースにアプリオリ法を適用しようとすると，マーケット・バスケット分析で解析の対象となるデータベースと比較して，非常に多くの頻出アイテム集合を生じることになる．Bayardo ら[11]は，密なデータベースからサポートと確信度に関する条件を満たす有用なルールを抽出する方式を提案しているが，抽出対象となるルール数が多くなると実行時間が増大してしまうという問題がある．

ところで，サポートと確信度の両者が，どの程度の値を満たせば興味深いルールであると判断できるかという客観的な基準は存在せず，前提部と結論部の相関関係の判断には適当でないとの指摘が Brin らによってなされている[4]．Brin らは統計学で用いられる χ^2 値を指標とする方法に注目しており，これを用いたアルゴリズムを提唱している．相関ルール $X \Rightarrow Y$ において，X，Y および $X \wedge Y$ を満たすレコード数の割合（サポート）をそれぞれ x，y および z とし，レコードの総数を N とする．**表 2** は，X と Y が独立であると仮定したときの期待度数 $E(r)$（上段）と，実際のレコード数である観測度数 $O(r)$（下段）を示したものである．ただし，r は，分割表の 4 つのセルに対応し，$r \in R$，$R = \{1, 2, 3, 4\}$ とする．この分割表に対して，

$$T = \sum_{r \in R} \frac{(O(r) - E(r))^2}{E(r)} \tag{1}$$

とすると，T は自由度 1 の χ^2 分布に従うことが知られている．これに表 2 の各値を代入すると，

$$T = \frac{N(z - xy)^2}{xy(1-x)(1-y)} \tag{2}$$

となる．T の値によって，X と Y の相関を判断するが，ある有意水準を定めて相関ルールの取捨を行うことができる．たとえば，有意水準を 5% とすると $T > 3.84$，1% とすると $T > 6.63$ であればその相関ルールは興味深いものとして抽出できる．

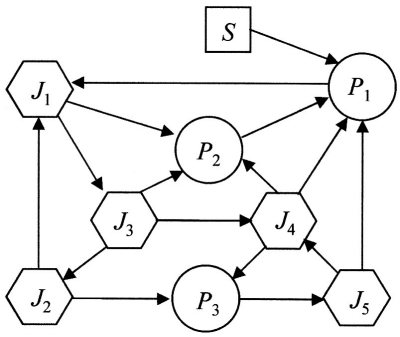

図 1 GNP の構造
Fig. 1 Structure of GNP.

3. 遺伝的ネットワークプログラミング（GNP）

本章では相関ルールの抽出に応用する遺伝的ネットワークプログラミング（GNP）[5)~8)] について概説する．

GNP では，ノードをネットワーク状に接続することによって，プログラムの自動生成を行う．GNP の基本構造を**図 1** に示す．GNP は，判定ノード（Judgement node）と処理ノード（Processing node）の 2 種類のノードからなり，有向グラフの構造になっている．ノード間の接続と遷移によりプログラムが生成される．

判定ノードは判定内容（判定ノード関数）が記載されているノードであり，あらかじめ設計者の用意した判定内容と入力を参照し分岐を行う．処理ノードは，遺伝的プログラミング（GP）の終端記号に対応づけることができ，そこには，環境に対して行う処理など扱う問題に応じて設計者の用意した処理内容が記載されている．図 1 において，J_n は判定ノード用のライブラリに記載されている n 番目の判定内容を示している．同様に，P_n は処理ノード用のライブラリに記載されている n 番目の処理内容を行うことを示している．図 1 の S は初期起動ノードでありただ 1 度だけ使用される．

GNP の遺伝子表現は，全ノードにそれぞれ固有なノード番号 i が与えられており，ノード番号 i の遺伝子の構成が**図 2** に示されている．図 2 の各変数を以下に示す．GNP の応用方法によっては，判定・処理に要する時間 d_i や遷移の際の遅れ時間 d_{ij} などが加わる．

NT_i：種別遺伝子
（0：初期起動ノード，1：判定ノード，2：処理ノー

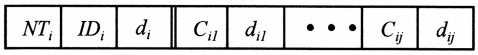

図 2 GNP の遺伝子構造（ノード番号 i）
Fig. 2 Gene structure of GNP (node i).

ド）

ID_i：判定・処理内容遺伝子

（GNP のライブラリに示されている判定，処理内容のコード番号）

C_{ij}：接続遺伝子

（ノード番号 i から分岐する j 番目のノード番号）

また，GNP の進化方法として，次に示す通常の方法を使用する．

- 突然変異：親個体のノードの接続先の変更とノード関数の変更を確率的に行い子個体を生成する．
- 交叉：親個体間の対応するノードの接続先とノード関数を確率的に交換し子個体を生成する．
- 再生：エリート保存選択，トーナメント選択を併用する．

4. GNP による興味深い相関ルールの抽出

本章では，GNP を応用して，データベース中に存在する興味深い相関ルールを抽出する方法を提案する．

4.1 概　　要

提案方法において，相関ルールを構成する属性は 1 または 0 の値をとるとする．したがって，対象となるデータベースは表 1 のような形態となる．抽出する相関ルールは，A_i をデータベースを構成する属性として，前提部と結論部を自動設定するもので

$(A_j = 1) \wedge ... \wedge (A_k = 1)$

$\Rightarrow (A_m = 1) \wedge ... \wedge (A_n = 1)$

（$A_j \wedge ... \wedge A_k \Rightarrow A_m \wedge ... \wedge A_n$ と略記する）とする．

提案手法は，通常の進化論的計算手法のように最適な個体（解）を発見するのではなく，初期世代から最終世代までの間に，進化をとげながら存在した全個体の中から興味深い相関ルールの抽出を実行する．その結果，各世代で出現する興味深いルールは相関ルールライブラリに蓄積されていく．

本論文のすでに提案されている手法[9),10)] と異なる点は，以下のとおりである．

- 判定ノード関数の数は進化時に可変とする．GNP 個体内に存在する判定ノードの総個数は一定だが，特定の属性に対応した判定ノードの個数は固定されていない．
- 進化操作において，判定ノード関数の突然変異を扱う．この結果，判定ノードの突然変異は，接続先の変更と属性の変更となる．

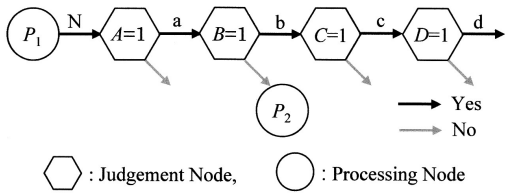

図 3 相関ルールと GNP
Fig. 3 GNP for association rule mining.

表 3 相関ルールと指標
Table 3 Association rules.

association rules	support	confidence
$A \Rightarrow B$	b/N	b/a
$A \Rightarrow B \wedge C$	c/N	c/a
$A \Rightarrow B \wedge C \wedge D$	d/N	d/a
$A \wedge B \Rightarrow C$	c/N	c/b
$A \wedge B \Rightarrow C \wedge D$	d/N	d/b
$A \wedge B \wedge C \Rightarrow D$	d/N	d/c

- ライブラリに格納された抽出済みの相関ルールに関する情報を GNP の進化操作時における条件設定に利用する．

4.2 GNP と相関ルールの対応

1 つの属性を GNP の 1 つの判定ノードで表現する．判定ノードでは指定した属性値であるかどうかを判定し，Yes/No に 2 分岐する．また，処理ノードは固有の順番を有しており，相関ルールの管理や諸指標の算出を行う．**図 3** に示すように「1 つの処理ノードを始点とする一定個数以内の判定ノードの連結の範囲」で相関ルールの集団を形成するものとし，データベースのレコード 1 つずつについて，GNP のノードを処理ノードから順に遷移していく．たとえば，表 1 の 1 番目のレコード（$TID = 1$）は，$A = 1$ を満たすが，$B \neq 1$ なので，図 3 において P_1 から遷移を始めて P_2 に達することになる．

判定ノードを Yes 側に連続して遷移する場合は，処理ノードから遷移したノード数があらかじめ指定した個数，すなわち相関ルールを構成する属性の数の最大数に達したら，判定ノードの判定結果にかかわらず次の順番の処理ノードに接続する．

図 3 の処理ノード P_1 を通過したレコード数を N とし，この N 個のレコードのうち，判定ノード群を Yes 側に進んだレコード数を判定ノードの位置で調査して a，b，c および d とする．このとき，**表 3** のように各相関ルールのサポートと確信度が算出できる．

判定ノードを No 側に進む場合は，次の順番の処理ノードに接続し，次の判定ノード群の探索を行う．

4.3 相関ルールの抽出

GNP のノードの接続は以下のように行われる．判

定ノードの Yes 側の接続先は判定ノードとする．また判定ノードの No 側の接続先は，次の順番の処理ノードとする．したがって，同一の判定ノードに遷移してきても，どの処理ノードに関連したレコード調査を行っているかによって No 側の接続先の処理ノードは異なる．

また，処理ノードは，分岐先は 1 つであり，分岐後判定ノードに接続する．これは，あらかじめ判定ノードだけでネットワークを構成しておき，あとから処理ノードを組み込んだことに相当する．なお，判定ノードの Yes 側の接続からなるネットワークは，ループを形成していたり，特定の判定ノードに複数の判定ノードから接続したりしていてもよい．このような GNP の特徴を活かした構成をすることで，判定ノード群の接続を変えることなく，処理ノードの接続先を変えるだけで GNP 内に存在する多数の相関ルールを探索することが可能となる．

たとえば，図 3 の処理ノード P_1 の接続先を $A=1$ の判定ノードから $B=1$ の判定ノードに接続を変更することにより，B，$B \wedge C$ および $B \wedge C \wedge D$ を満たすレコード数を求めることができる．したがって，たとえば，表 3 の結論部 B，$B \wedge C$ および $B \wedge C \wedge D$ のサポートが計算でき，表 2 に示した Ny が与えられることになるので，相関ルールの χ^2 値を算出できる．このように処理ノードの接続先を変化させながら，GNP を連鎖的に探索することで効率良く興味深い相関ルールの抽出が実行できる．

GNP の探索は，第 1 番目の処理ノード P_1 からスタートし，すべての処理ノードについてレコード調査を終えたらレコードを更新し，また P_1 に戻る．規定のレコードを探索したら，サポートや確信度，χ^2 値を計算して 1 世代を終える．この探索はレコードごとに全個体について並行して進める．

各世代で求まった興味ある相関ルールは，逐次，新規かどうかを世代間に共通の相関ルールライブラリを用いて判定する．このとき，相関ルールを構成する属性群を規格化した表現に改めるために，属性の重複があればこれを除き，前提部および結論部を構成する属性をある決まった順序（たとえば辞書順）にソートする．新規のルールが抽出された場合は，これをサポートや確信度，χ^2 値などの値とともに，相関ルールライブラリに蓄積していく．

4.4 GNP の適合度

興味深い相関ルールとして，サポートと χ^2 値を用いて，たとえば，

$$\chi^2 > 6.63 \tag{3}$$

$$support \geq sup_{min} \tag{4}$$

をともに満たす相関ルールを興味深いものとして抽出する．ただし，sup_{min} はユーザが指定するサポートの最小値である．

GNP 各個体の適合度（Fitness）F は，未抽出の興味深い相関ルールを次世代において抽出することが期待できる個体ほど大きな値を持つように設定することになるため，次のような個体が高く評価されればよい．

- 新規の興味深い相関ルールを抽出できた個体
- 新規ではなくても興味深い相関ルールを多く含む個体

また属性数に関しては，

- ルールを構成する属性数が多い興味深い相関ルールを含む個体

ほど高い適合度とする．こうした観点から，式 (3) および式 (4) の抽出の基準を満たす GNP 個体内の相関ルール i の中で，前提部と結論部を構成する属性数 $n(i_{ante})$，$n(i_{con})$ と相関ルール i の χ^2 値である ${\chi_i}^2$ を用いて適合度を設定する．さらに，その相関ルール i が新たに抽出されたものである場合は，次世代に残す価値が高いとして α_{new} を加算するものとする．したがって，

$$F = \sum_{i \in I} \{ {\chi_i}^2 + 10(n(i_{ante}) - 1) + 10(n(i_{con}) - 1) + \alpha_{i_{new}} \} \tag{5}$$

が適合度となる．ここで，I は式 (3)，かつ式 (4) の興味深い相関ルールの基準を満足する GNP 個体の相関ルールの集合である．また，

$$\alpha_{i_{new}} = \begin{cases} \alpha_{new} & (i \text{ is new}) \\ 0 & (i \text{ is not new}) \end{cases} \tag{6}$$

である．

4.5 GNP の進化

GNP の初期世代において，すべての判定ノードの属性と接続先，処理ノードの接続先はランダムに設定される．判定ノードおよび処理ノードの合計の個数は全世代を通して一定である．ところで，従来の手法[9),10)]では，同一の種類の判定ノードの個数が固定されていたため，全世代の全 GNP 個体が同一の種類の判定ノードを同数ずつ含んでいた．しかし，本手法においては各個体を構成する判定ノードの種類とその数は個体および世代に応じてそれぞれ異なることになる．

GNP の進化では，各個体の適合度の値が大きいものだけを選択する．未抽出の相関ルールを獲得するために，選択された全個体について，交叉または突然変異を行う．遺伝子は，判定ノードの遺伝子群と，処理

ノードの遺伝子群に区分して，それぞれのノード群内で交叉と突然変異の操作を行う．突然変異には，ノードの接続先をランダムに変える突然変異 1 と，判定ノードの属性をランダムに別の属性に変更する突然変異 2 の 2 通りを考える．なお，突然変異や交叉による操作は，前世代の個体内に存在する興味深い属性の組合せの一部を変更して新しいルールの候補を作り出すことに対応する．

選択は全個体中，適合度の高い上位 1/3 の個体をランクづけして選出し，各個体を 3 個ずつ再生する．その後，交叉，突然変異 1（接続先の変更），突然変異 2（ノード内容の変更）の操作をそれぞれ行い，選択前と同数の個体を形成する．判定ノードの遺伝子群における突然変異 1，突然変異 2 の発生確率 P_{m1}，P_{m2} の設定により，探索的なルールの抽出を行うか，すでに抽出されたルールの一部の属性を変更した精査的な抽出を行うかを調節できる．なお，処理ノードの遺伝子群については突然変異 1 を確率 1 で行い，接続先をランダムに決定する．また，交叉は，判定ノードの遺伝子群について 2 個体を 1 組として一様交叉をあらかじめ設定した確率 P_c で行う．

4.6 獲得した情報の利用

GNP の進化時における突然変異の操作のうち，判定ノードの属性を変更する場合において，どの属性を新たに選択するかを決定する際にライブラリに蓄積されている相関ルールの情報を用いることができる．具体的には，突然変異 2 の操作において，確率 P_{m2} で突然変異する判定ノードが，属性 A_j のノードとなる確率を以下に示すような P_j^{all} または P_j^g を用いて定める．

GNP が稼働中のある世代に注目した場合，初期世代から直前の世代までに抽出されたプール内の興味深い相関ルールの集合を考慮した属性 A_j の出現頻度を $n(A_j)$ とする．このとき，属性 A_j の選ばれる確率 P_j^{all} を

$$P_j^{all} = \frac{n(A_j)+1}{\sum_k (n(A_k)+1)} \tag{7}$$

とする．ここで，A_k は探索の対象となるすべての属性を表すものとする．

また，GNP が稼働中のある世代からみて直前の g 世代の間に抽出されたプール内の興味深い相関ルールの集合を考慮した属性 A_j の出現頻度を $n_g(A_j)$ とする．このとき，属性 A_j の選ばれる確率 P_j^g を同様に

$$P_j^g = \frac{n_g(A_j)+1}{\sum_k (n_g(A_k)+1)} \tag{8}$$

とする．稼働した世代数が g に満たない場合は，初期世代から直前の世代までに抽出された情報を用いるものとする．なお，式 (7)，(8) における 1 は，相関ルールがはじめて抽出されるまで，および P_j^g の計算において直前の g 世代において抽出されたルールがない場合において，すべての属性が等確率で選ばれる設定とするために使用している．

5. シミュレーション

本章では，人工的に作成したデータベースを用いて行ったシミュレーションについて，その設定と結果および考察を示す．シミュレーション 1 では獲得した情報の利用に関して，シミュレーション 2 では特にデータベースの属性数が多い場合の獲得情報の利用に関して検討している．

5.1 基本設定

データベース 1 は，A，B，C，…，Z の 26 属性からなり，レコード数は 200 個とする．各属性は 1 または 0 の値をとり，26 属性のうち，属性値が 1 であるレコード数は，A，G，L，R および V の 5 個の属性が 140 個，残り 21 個の属性が 100 個である．データベース 2 は，属性値が 1，0 となる確率がそれぞれ 0.5 であるような乱数で作成したもので，R_1，R_2，…，R_{26} の 26 属性からなりレコード数が 200 個である．また，データベース 3 は，データベース 1 とデータベース 2 を結合させたもので，52 属性，レコード数 200 個で構成される．

シミュレーションでは，

$$(A=1) \wedge (B=1) \wedge (C=1) \wedge (D=1) \Rightarrow (Y=1) \wedge (Z=1)$$

のように属性値が 1 である相関ルールを抽出する．興味深い相関ルールの定義は式 (3)，(4) に加えて以下の条件

$$sup_{min} = 0.1, n(i_{ante}) + n(i_{con}) \geq 6,$$
$$n(i_{ante}) \leq 5, n(i_{con}) \leq 5$$

を満たすものとし，これらを満たす相関ルール i からなる集合が式 (5) における I となる．また，式 (6) において $\alpha_{new} = 150$，相関ルールを構成する属性の数の最大数を 10 とする．

GNP の 1 個体中の処理ノード数は 10 個，判定ノード数は 78 個とし，個体数は 120 とする．シミュレーションは，乱数系列を変えて 10 回行う．また，4.3 節**相関ルールの抽出**で記述した処理ノードの接続先を変えて連鎖的に相関ルールを抽出するための処理ノードの接続先の変更を，各個体において 1 世代の間に 5 回行う．処理ノードの変更後の接続先は，当該の処理

表 4 交叉確率および突然変異確率の条件
Table 4 Conditions of crossover and mutation.

	Type-M	Type-L
P_c	15/78	10/78
P_{m1}	1/3	1/5
P_{m2}	1/5	1/6

(Note: 78 corresponds to the number of Judgement nodes)

ノードに連続して接続された判定ノード 5 個のいずれかとなるが，この選択はランダムに行う．

交叉確率 P_c および突然変異確率 P_{m1}，P_{m2} の条件として**表 4** に示した Type-M，Type-L の 2 組を用いる．抽出の対象となる相関ルールを構成する属性数が 6〜10 個であるので，P_c，P_{m1} および P_{m2} の確率を 1/8〜1/3 程度に設定し，それらの組合せとして相対的に値の大きなものを Type-M，相対的に値の小さなものを Type-L としている．実験は，1.50 GHz Pentium M，504 MB RAM の計算機で行い，C 言語を使用した．

本手法の有効性を比較評価するため，ノード数，個体数の設定は同一であるが，すべての判定ノードの属性と接続先，処理ノードの接続先を各世代でランダムに設定するランダムモデルについてもシミュレーションを行う．

5.2 シミュレーション 1

シミュレーション 1 では，データベース 1 を用いて，A〜Z の 26 個の属性から興味深い相関ルールを抽出する．突然変異 2 における属性の種類選択の確率が，

- 等確率（P_j^0 と表す）
- P_j^{all}
- $P_j^5 (g = 5)$

の 3 通りの場合について実験を行う．

図 4，**図 5** および **表 5** は抽出された興味深い相関ルールの総数を示している．また，**図 6** は図 4 の結果を世代数に替えて計算時間で示したものである．図は 10 回の試行の平均値による．図中の M，L は表 4 の設定を，Random はランダムモデルを表している．また，上述の 3 通りの確率設定を順に，Pj0，Pjall および Pj5 と略記している．なお，図中の*C と表されたものは文献 10) の設定による結果である．これらの結果から獲得した情報の利用は相関ルールの抽出効率を向上させることが分かるが，獲得した情報の利用を直前の 5 世代とした方が，全情報を活用するよりも有効であるといえる．直前の世代のみの情報活用ではルールを構成する属性の種類の傾向が異なる新たなルール群を抽出可能となることによるものと考えられる．**図 7** は GNP 個体の適合度の平均値の推移を示し

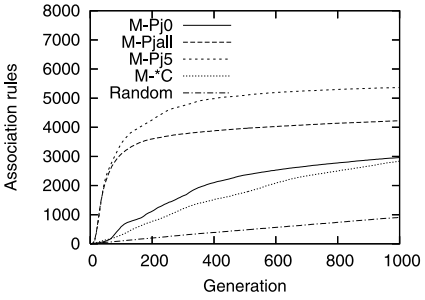

図 4 プールに抽出された相関ルールの数（Type-M, Simulation 1）
Fig. 4 Number of association rules in the pool (Type-M, Simulation 1).

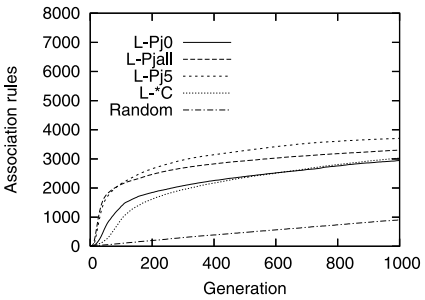

図 5 プールに抽出された相関ルールの数（Type-L，Simulation 1）
Fig. 5 Number of association rules in the pool (Type-L, Simulation 1).

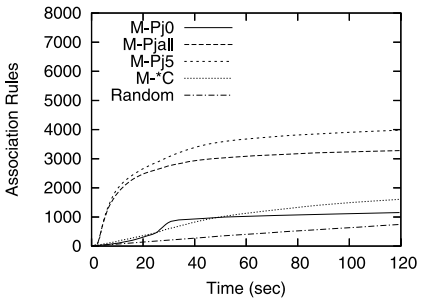

図 6 プールに抽出された相関ルールの数と計算時間（Type-M, Simulation 1）
Fig. 6 Number of association rules in the pool versus runtime (Type-M, Simulation 1).

ているが，これからも判定ノード関数を固定していた従来手法よりも，提案手法の方が各個体内により多くの相関ルールを構成していることが分かる．

図 8 は Type-M の設定で抽出された $n(i_{ante}) + n(i_{con}) = 7$ を満たす相関ルールの個数を表しており，この結果からも獲得した情報を用いた場合の方が抽出の効率が良いことが分かる．図 8 から，ランダムモデ

表 5 プールに抽出された相関ルールの数（シミュレーション 1）
Table 5 Number of association rules in the pool (Simulation 1).

		25^{th} generation			100^{th} generation			1000^{th} generation		
		Type-M	Type-L	Random	Type-M	Type-L	Random	Type-M	Type-L	Random
P_j^0	Max	63	527	—	1,404	2,267	—	4,029	3,637	—
	Ave	29.9	160.4	—	588.1	1,369.3	—	2,960.2	2,943.6	—
	Min	7	15	—	78	199	—	1,039	568	—
P_j^{all}	Max	1,250	1,437	—	3,428	2,747	—	5,054	4,012	—
	Ave	770.4	921.0	—	3,077.6	2,125.6	—	4,223.9	3,303.8	—
	Min	233	398	—	2,711	711	—	3,521	912	—
P_j^5 ($g = 5$)	Max	1,225	1,161	—	3,650	2,950	—	6,010	4,761	—
	Ave	726.6	705.1	—	3,431.8	2,148.2	—	5,363.6	3,704.5	—
	Min	46	266	—	3,229	791	—	4,887	976	—
Random	Max	—	—	44	—	—	107	—	—	939
	Ave	—	—	25.5	—	—	97.0	—	—	903.6
	Min	—	—	14	—	—	84	—	—	855

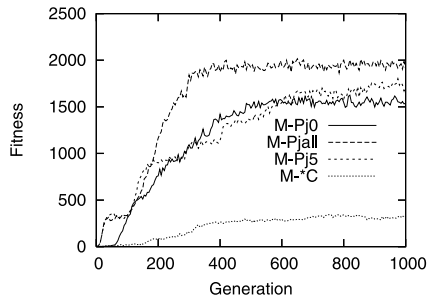

図 7 適合度曲線（Type-M, Simulation 1）
Fig. 7 Fitness curves (Type-M, Simulation 1).

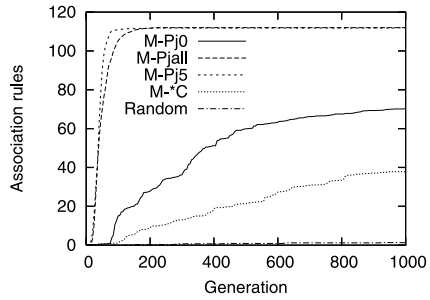

図 8 プールに抽出された属性数 7 の相関ルールの数（Type-M, $n(i_{ante}) + n(i_{con}) = 7$, Simulation 1）
Fig. 8 Number of association rules in the pool (Type-M, $n(i_{ante}) + n(i_{con}) = 7$, Simulation 1).

ルでは属性数 7 個の興味深い相関ルールをほとんど抽出することができないことが分かるが，これは 26 属性のデータベース 1 から 7 個の属性の相関ルールを抽出する場合には前提部・結論部の組合せが約 7,370 万通りあり，これを与えるような判定ノードの接続が乱数では発生しにくいためである．また，これを全探索で抽出しようとすると，相関ルールの χ^2 値を算出するためには，属性数 2〜5 個の属性の集合（約 8 万 4 千個）のサポートが記憶されている必要があり，前提部，結論部およびルール全体を構成する属性のそれぞれのサポートを随時読み込んですべての組合せを処理することになる．

図 9，**図 10** はとくに M-P_j^0 および M-P_j^5 の場合のプールに抽出されたルールの個数の 10 回の試行結果をそれぞれ示しているが，M-P_j^5 では，属性数 7 の興味深い相関ルールが短い世代数で効率良く抽出されている．

5.3 シミュレーション 2

シミュレーション 2 では，データベース 3 を用いて，52 個の属性から興味深い相関ルールを抽出する．なお，データベース 2 に対してシミュレーション 1 と同一条件の抽出を行ったところ，興味深い相関ルールは存在しないという結果を得ている．

シミュレーション 1 で抽出効率が高いと明らかになった P_j^5 とこれとの比較のため P_j^0 について表 4 の Type-M を用いて実験を行う．**図 11**，**図 12** は，それぞれ 52 個の属性群からプールに抽出された $sup_{min} = 0.1$，$n(i_{ante}) + n(i_{con}) \geq 6$，$n(i_{ante}) \leq 5$，$n(i_{con}) \leq 5$ を満たす相関ルールの数の 10 回の試行の結果である．データベースの属性数が 26 から 52 に増加したことで，属性数 6 の相関ルールを構成する属性の組合せ数は約 88 倍，属性数 7 の場合には約 203 倍となるため効率的な抽出が始まる世代が遅れる傾向にあり，試行によるバラツキもみられる．これは，適合度の高い個体が発生するのに要する世代数の違いが試行によって生じるためである．ランダムモデルを用いた場合には，1,000 世代経過時点で抽出されたルールの総数が，平均 42.4 個，最大 52 個，最小 34 個であった．これらのことから，データベースの属性が多い場合においても獲得した情報を用いる提案手法が有

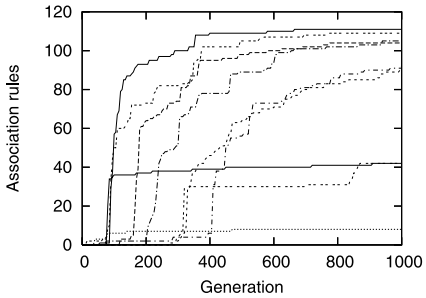

図 9 10 回の試行によるプールに抽出された属性数 7 の相関ルールの数（Type-M, P_j^0, $n(i_{ante})+n(i_{con})=7$, Simulation 1）

Fig. 9 Number of association rules in the pool by 10 trials (Type-M, P_j^0, $n(i_{ante})+n(i_{con})=7$, Simulation 1).

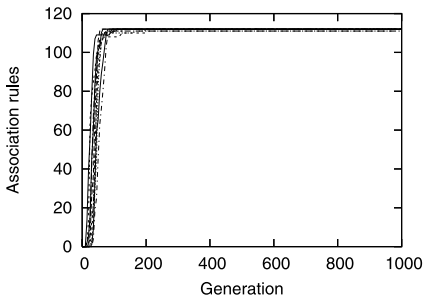

図 10 10 回の試行によるプールに抽出された属性数 7 の相関ルールの数（Type-M, P_j^5, $n(i_{ante})+n(i_{con})=7$, Simulation 1）

Fig. 10 Number of association rules in the pool by 10 trials (Type-M, P_j^5, $n(i_{ante})+n(i_{con})=7$, Simulation 1).

効であることが分かる．

図 13 は図 12 の結果を世代数に替えて計算時間で示したものである．52 属性からなるデータベース 3 から 120 秒程度で多くの興味深いルールを抽出することができるといえるが，この時間は図 6 (P_j^5) の約 40 秒と比較すると 3 倍程度である．このことから，提案手法がデータ規模が大きくなった場合においても計算時間が極度に増加しないことが分かる．

シミュレーション 1 で抽出される相関ルールは，すべてシミュレーション 2 でも抽出の対象になる．**図 14** は，シミュレーション 2 で抽出された相関ルールの中で，A〜Z の 26 属性から構成され，かつ $n(i_{ante})+n(i_{con})=7$ を満たすものの抽出結果を示している．なお，図 12，図 13，図 14 の線種は同一の試行を表している．効率的な抽出が始まる世代にバラツキがあるものの，図 10 との比較から興味深い相関ルールを効率良く抽出していることが分かる．一方，ランダムモデルを用いた場合には，10 回の試行すべ

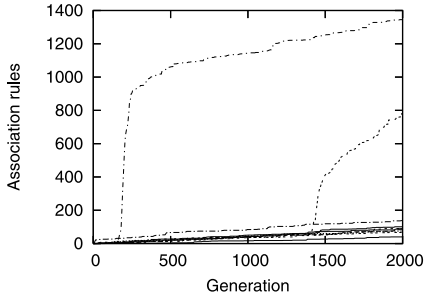

図 11 10 回の試行による 52 個の属性群からプールに抽出された相関ルールの数（Type-M, P_j^0, Simulation 2）

Fig. 11 Number of association rules in the pool by 10 trials (Type-M, P_j^0, Simulation 2).

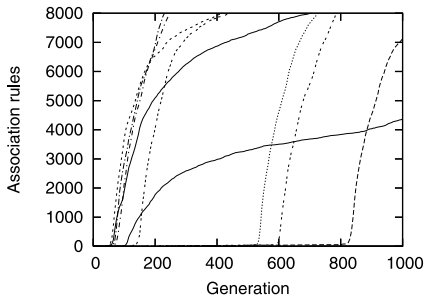

図 12 10 回の試行による 52 個の属性群からプールに抽出された相関ルールの数（Type-M, P_j^5, Simulation 2）

Fig. 12 Number of association rules in the pool by 10 trials (Type-M, P_j^5, Simulation 2).

てにおいて 1,000 世代経過時点でこの条件を満たすものは抽出されなかった．

データベース 3 からアプリオリ法を用いて，同一の興味深さ指標を満たす相関ルールを抽出しようとする場合には，属性数 6 個以上の頻出アイテム集合（属性値が 1 である属性の集合）をすべて抽出することになる．データベース 3 は，1 レコードあたりの属性値が 1 となる属性数は平均 27 個であり，密なデータベースと考えられる．したがって，データベース 3 の 3 個の属性からなる属性の集合（約 2 万 2 千通り）のほとんどと，4 個の属性からなる属性の集合（約 27 万通り）の大半は，頻出アイテム集合の条件（$sup_{min}=0.1$）に適すると考えられる．このため，χ^2 値算出のために，これらの集合それぞれについて，サポートと構成する属性を記憶しておく必要がある．属性数 6 個の頻出アイテム集合に対して，相関ルールの前提部・結論部の組合せは 62 通りずつあるため，それぞれについて前提部，結論部およびルール全体を構成する属性のそれぞれのサポートを随時読み込んで χ^2 を計算していくことになる．こうした計算量や記憶するデータ量

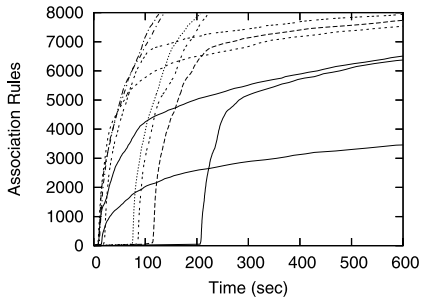

図 13 10 回の試行による 52 個の属性群からプールに抽出された相関ルールの数と計算時間（Type-M, P_j^5, Simulation 2）
Fig. 13 Number of association rules in the pool versus runtime by 10 trials (Type-M, P_j^5, Simulation 2).

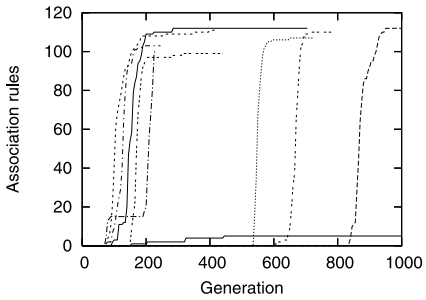

図 14 10 回の試行による 52 個の属性群中の特定の 26 個の属性群で構成された属性数 7 のプール内の相関ルールの数（Type-M, P_j^5, $n(i_{ante}) + n(i_{con}) = 7$, Simulation 2）
Fig. 14 Number of association rules consisting of A-Z out of 52 attributes in the pool by 10 trials (Type-M, P_j^5, $n(i_{ante}) + n(i_{con}) = 7$, Simulation 2).

を減らそうとすると，sup_{min} の値をより大きな値に変更することになる．したがって，提案手法のような多くの属性数で構成される相関ルールの効率的な抽出は困難となる．また，密なデータベースにおいて属性数がさらに多くなる場合には，アプリオリ法ではコンピュータの能力という問題が生じてくる．一方，提案手法では，GNP の判定ノード関数の数を増やすことで対応できる．提案手法は，データベースに存在するすべての興味深い相関ルールを抽出できないが，ユーザの指定した時間に応じて抽出できる興味深い相関ルールを獲得する方法である．

6. 結　　論

獲得した情報を用いる遺伝的ネットワークプログラミング（GNP）によって興味深い相関ルールの抽出を行う手法を提案した．GNP が，相関ルール候補の発生，ルール抽出のデータ処理および χ^2 値算出のための情報収集を自ら行っているだけでなく，自ら獲得した情報を進化時に活用しているという特徴を持っている．また，提案手法はルール中の属性群の条件などユーザの定義した条件を満足する興味深い相関ルールを抽出できるという特徴も持っている．シミュレーションの結果から，提案手法が効率良く興味深い相関ルールの抽出を可能とすることを明らかにした．このことは，進化論的計算手法をデータマイニングの分野に応用することの有効性を示している．シミュレーションで用いたデータベースは人工的に作成されたものであるが，今後は，種々の分野における実際のデータベースへの応用を検討していく予定である．

参 考 文 献

1) 福田剛志，森本康彦，徳山　豪：データマイニング，共立出版 (2001).
2) Zhang, C. and Zhang, S.: *Association Rule Mining: models and algorithms*, Springer (2002).
3) Agrawal, R. and Srikant, R.: Fast Algorithms for Mining Association Rules, *Proc. 20th VLDB Conf.*, pp.487–499 (1994).
4) Brin, S., Motwani, R. and Silverstein, C.: Beyond market baskets: generalizing association rules to correlations, *Proc. ACM SIGMOD*, pp.265–276 (1997).
5) Katagiri, H., Hirasawa, K. and Hu, J.: Genetic network programming — application to intelligent agents, *Proc. IEEE International Conf. on Syst., Man and Cybernetics*, pp.3829–3834 (2000).
6) Katagiri, H., Hirasawa, K., Hu, J. and Murata, J.: Network Structure Oriented Evolutionary Model — Genetic Network Programming, *Proc. Genetic and Evolutionary Computation Conference*, pp.219–226 (2001).
7) 平澤宏太郎，大久保雅文，片桐広伸，胡　敬炉，村田純一：蟻の行動進化における Genetic Network Programming と Genetic Programming の性能比較，電学論 C，121-6，pp.1001–1009 (2001).
8) 片桐広伸，平澤宏太郎，胡　敬炉，村田純一：ノード数可変型 Genetic Network Programming，電学論 C，123-1，pp.57–66 (2003).
9) 嶋田　香，平澤宏太郎，古月敬之：遺伝的ネットワークプログラミングによる相関ルールの抽出，第 14 回インテリジェント・システム・シンポジウム，pp.363–368 (2004).
10) Shimada, K., Hirasawa, K. and Furuzuki, T.: Association rule mining using genetic network programming, *The 10th International Symp. on Artificial Life and Robotics 2005*, pp.240–245 (2005).
11) Bayardo Jr., R.J., Agrawal, R. and Gunopu-

los,
D.: Constraint-Based Rule Mining in Large, Dense Databases, *Proc. 15th International Conf. on Data Engineering*, pp.188–197 (1999).
12) Park, J.S., Chen, M.S. and Yu, P.S.: An Effective Hash-Based Algorithm for Mining Association Rules, *Proc. 1995 ACM SIGMOD Conf.*, pp.175–186 (1995).
13) Wu, X., Zhang, C. and Zhang, S.: Efficient Mining of Both Positive and Negative Association Rules, *ACM Trans. Information Systems*, Vol.22, No.3, pp.381–405 (2004).
14) Tung, A.K.H., Lu, H., Han, J. and Feng, L.: Efficient Mining of Intertransaction Association Rules, *IEEE Trans. Knowledge and Data Engineering*, Vol.15, No.1, pp.43–56 (2003).

(平成 17 年 3 月 3 日受付)
(平成 17 年 9 月 2 日採録)

嶋田　　香（学生会員）

1986 年早稲田大学理工学部応用物理学科卒業．1999 年筑波大学大学院修士課程医科学研究科医科学専攻修了．ユニーテック（株）にて遺伝子発現頻度情報解析の研究に従事．現在，早稲田大学大学院情報生産システム研究科博士後期課程在学中．

平澤宏太郎（正会員）

1964 年九州大学工学部電気工学科卒業．1966 年同大学大学院工学研究科修士課程電気工学専攻修了．同年（株）日立製作所入社，日立研究所勤務，1989 年同研究所副所長，1991 年同大みか工場主管技師長．1992 年九州大学工学部教授，1996 年同大学院システム情報科学研究科教授，2000 年同大学大学院システム情報科学研究院教授．2002 年早稲田大学大学院情報生産システム研究科教授，現在に至る．計測自動制御学会，電気学会，IEEE の各会員．工学博士．

古月　敬之

1986 年中国中山大学大学院修士課程修了．同年同大学電子工学科助手，1988 年同講師．1997 年九州工業大学情報工学研究科博士後期課程修了．同年九州大学システム情報科学研究科助手，2000 年同大学大学院システム情報科学研究院助手．2003 年早稲田大学大学院情報生産システム研究科助教授，現在に至る．計測自動制御学会，電気学会の各会員．博士（情報工学）．