データベースシステム　97－2
（１９９４．　３．　１８）

分類学習を用いた機能モジュールの自動選択法に関する検討

大竹 孝幸、平野 泰宏、井上 潮
ＮＴＴ情報通信網研究所

拡張可能ＤＢＭＳにおいて、実行時の環境に合わせて最適な機能モジュールを自動的に選択するために分類学習を用いる方式を提案し、ＤＢＭＳのバッファ管理機能を対象としてモジュール選択機構のプロトタイプを作成し評価した結果を述べる。バッファ管理機能のように大量のメモリ資源を管理する場合、環境の変化の度に機能モジュールを変更すると変更によるオーバヘッドから性能が劣化する事が予想される。この問題を解決するために、二段階に機能モジュール変更の判定を行なう方式を考案し、高い性能レベルを維持しながら機能モジュールの変更回数を大幅に削減できる事を確認した。

An Automatic Module Selection Mechanism Using Machine Learning Algorithms for Classification Rules

Takayuki Ohtake, Yasuhiro Hirano and Ushio Inoue
NTT Network Information Systems Laboratories

In this paper, we propose an automatic module selection mechanism according to transition of environments on extensible DBMS. We apply machine learning algorithms to it. The effictiveness of them are evaluated by prototyping of a buffer management functional module. In this case, we should not change modules whenever environments are changed, because flushing buffers to disk makes the system performance worse. Therefore, we developed a novel way of module selection with two stages. This approach decreases the number of module changes without the serious performance down.

1 はじめに

DBMSは複数の機能から成り立っている。そして、各々の機能を実現するための方式は複数考えられる。よって、システム設計者は設計段階で処理性能を見積り、方式を絞り込む必要がある。しかし、処理性能に影響を及ぼす要素は多数あり、性能の見積りは困難である。更に、ある程度の見積りが可能であっても、全ての環境下に渡って最高処理性能が得られる方式は無い。これより、システム運用後のサービス内容に変更が生じた場合、当初に最良の性能が期待できるとして決定した方式では満足な性能が得られなくなる可能性がある。

現在の汎用DBMSでは、常に最高の性能が期待できなくても、ある環境で著しく性能が劣化する心配のないオールマイティな方式を選択する。そして、いくつかのパラメータに対しては、ユーザがチューニングを行なう余地を残している。しかし、この汎用DBMSで要求される性能が得られない場合、特化した使用目的や使用形態において最良だと思われる方式を選択した専用DBMSを新規に開発する必要が生じるし、膨大なマンパワーが必要になる。

この問題を解決するため、DBMSの機能を実現するモジュール（以下では部品と呼ぶ）をシステム設計後でも入れ替えられる拡張可能DBMSが研究されている[1][2]。この拡張可能DBMSを用いて運用時に最良な部品の選択、入れ替えを行なえば、広い範囲での使用形態で高い性能を得る事ができるようになる。

拡張可能DBMSでは、(1)システム運用中の動的な部品の変更、(2)部品インターフェイスの依存関係の記述、(3)一貫性制約、(4)自動最適化などの未解決な課題がいくつか残っている。本稿では、課題(4)の最良の性能を得ることができる部品の決定を自動的に行なう手段を検討する。

拡張可能DBMSは、DBMSを構成している機能毎に部品を選択し入れ換える事を可能にしているが、本テーマでは１つの具体的なケーススタディとしてバッファ管理機能[7]を対象とする。

2 方式の自動選択へのアプローチ

2.1 方針

従来のDBMSは問い合わせの形式やインデックスの有無等を用いてアクセスパスを選択する最適化は行なっていた。しかし、より多彩な環境変化に対応し、依存関係がある複数の部品の組み合わせでシステムを構成する場合には、最適化ルールが複雑になるために最適化が困難であった。

一方、知識処理の分野では、多数の事例からルールを自動的に発見する分類学習法が研究されており（図１参照）、代表的な手法としてAq11、ID3が使用されている[3][4][5]。本研究では、部品を選択するルールを自動的に得るために、分類学習を用いることにした。分類学習では、事例を表すいくつかのパラメータと、その事例が属するカテゴリーから構成される事例データを用いて、どのようなパターンの事例がどのカテゴリーに分類されるのが適当であるかというルールを得る。

分類学習を用いた部品選択を行なうシステム

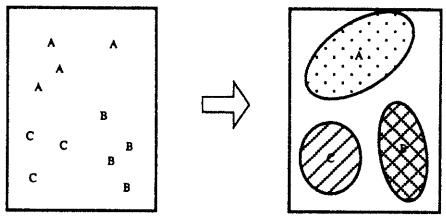


図１：事例からの分類学習

構成を図２に示す。まず、DBMS運用中に性能（レスポンス時間、スループット、など）を測定し、その性能が得られた時の環境（運用中のAP種類、トラヒック、など）と、使用部品を合わせた情報を測定データとして蓄積する。次に、この測定データを用いて、環境をパラメータとし、その環境で最良な部品をカテゴリとする事例データを作成する。そして、事例データを学習機構が分類学習することにより部品選択ルールを得る。この部品選択ルールを活用することにより、現在の使用環境で最良な部品を自動的に選択し、DBMSにその部品への変更要求を行なう。

分類学習を用いることにより、測定データが得られていない環境下でも最良な部品を予測することができる。また、新たに得た測定データを用いてルールを進化させる事ができる。

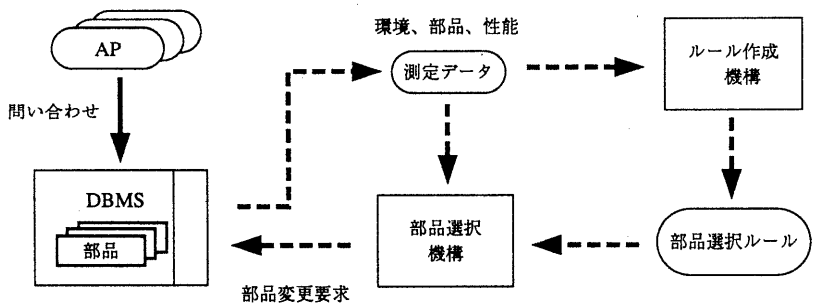


図２：部品選択を行なうシステム構成

2.2 課題

上記の部品選択を行なうシステム構成を実現させる上で以下の課題が存在する。本稿では、これらの課題について検討する。

(1) 入れ替えオーバヘッドを考慮した部品変更方法

環境の変化に応じて常に最良な部品を用いる方法では、頻繁に部品変更が生じる可能性がある。アクセスパスの選択のように個々の処理に閉じた最適化であれば、問い合わせ毎に複数の部品から選択することは容易である。しかし、バッファ管理のように、それまでの処理内容が現在の処理に影響を及ぼす場合、部品変更に伴うオーバーヘッドがかえって性能を劣化させる事が予想される（図３参照）。具体的に述べると、バッファ管理は頻繁にアクセスされるデータをより高速にアクセスする機能であるが、部品変更に伴ってバッファ領域のフラッシュが生じ、一時的に性能を劣化させる。したがって、入れ替えのオーバヘッドを考慮して部品変更を行なわねばならない。

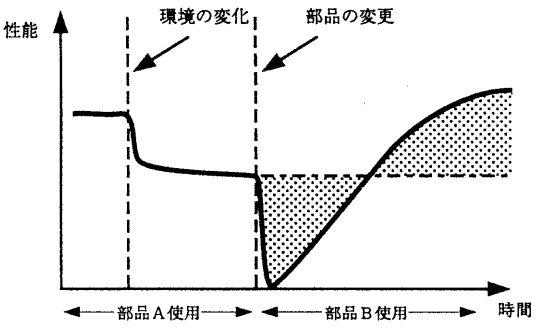


図３：部品変更による性能劣化

(2) カテゴリーの決定基準

分類学習を行なう上で、事例データ内に記述すべきカテゴリーを決定する教師が必要となる[6]。しかし、自動化を実現する上で、この教師が外部から与えられた指示によるものではなく、学習機構自体に内在的に組み込まれたものである必要がある。このカテゴリー決定の基準（教師）の設定を行なわなければならない。ただし、絶対的な値を宣言的に与える方法ではなく（レスポンス時間が10msec.以上と未満とで分けるなど）、より柔軟な方法であることが望ましい。

3 課題の解決方法

3.1 部品の変更方法

2.2 で述べたように、常に最良の部品を選択すると部品変更が頻繁に生じる可能性がある。この問題を解決するために、以下のように二段階で変更の判定を行なう機構を提案する。

第一段階：現在の環境で使用中の部品の性能が高い方か低い方かを調べる。性能が閾値以下ならば第二段階へ移行する。

第二段階：現在の環境で最良な部品を選び、使用中の部品を変更する。

例えば、性能を10段階にレベル分けし、閾値をレベル8に設定したとする。第一段階におい

て、現在の環境で使用中の部品の性能がレベル8以上と判定されれば、ほぼ良好な処理性能を期待できる。レベル8未満ならば、他の部品を用いた方がよいと判定されて、第二段階に進行する。第二段階では、現在の環境で最良と判定された部品に変更する。

本方式には以下の2種類のルールが必要である。第一段階で部品の性能レベルを判定するための、環境と部品をパラメータとし、性能レベルをカテゴリーとした事例データを分類学習させたルールと、第二段階で最良の部品を選択するための、環境をパラメータとし、最良の部品をカテゴリーとした事例データを分類学習させたルールである。

3.2 カテゴリー決定の基準作成

第一段階

第一段階では、測定した性能がその環境下で閾値以上の性能レベルに属するか否かを決定する。性能値のとり得る範囲は環境によって異なるため、全ての環境において使用できる絶対的なレベル分けの基準や閾値を決定することは困難である。そこで、本検討では、環境毎に以下のように基準を設定した。ある環境下で収集された測定データの中で、部品毎に性能値の平均をとり、その環境下での性能の最大値と最小値を得る。その間をそれぞれ範囲が均等になるようにレベル個数で分割する。そして、各部品の性能値が分割した何番目の区間に含まれるかを求め、それが閾値以上か未満かによりカテゴリー分けを行なう（図4参照）。

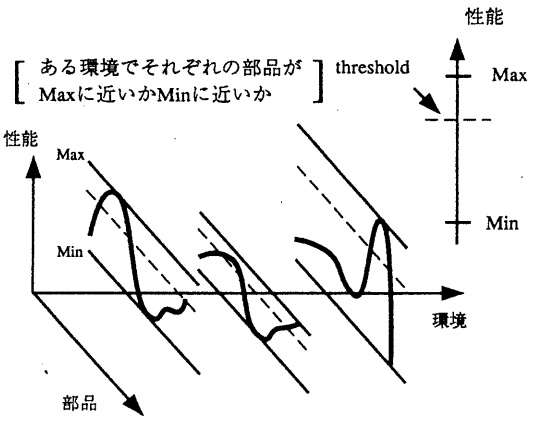


図4：第一段階の判別方法

本方式では、各環境内での相対的な評価ができ、閾値の設定も容易である。ただし、カテゴリー決定における確信度*を考慮せずに、一つの環境下で充分多くの性能測定がなされているとした。

(* Certainty Factor ：事例やルールに対して不確実性を考慮するために確率量を付与して扱う。通常-1から1の範囲を用い、これを確信度と呼ぶ。例えば、「40度以上の熱があり、咳がでるなら0.8の確信度で風邪である」のような用いられ方をする)

第二段階

第二段階では、各環境でどの部品を選ぶことが最良であるかを決定することが必要である。ここでは、第一段階の基準作成の際に性能が最大値であった部品をカテゴリーとして用いる。

4 評価

バッファ管理の部品を変更でき、環境パラメータと各部品の性能を測定し、記録できるようにしたプロトタイプを実装した。そのプロトタイプと学習アルゴリズムのAq11法、ID3法を用いて以下の評価を行なった。

・二段階の部品変更判定の有効性を確認する。
・部品変更ルールが未測定環境下でも的確な判定ができるかを評価する。

4.1 評価環境

本評価で用いた計算機の環境を以下に示す。

機種	SUN SPARC Station 2
OS	SunOS 4.1.1-JLE 1.1.1 RevB
主記憶	64MB

4.2 モデル

4.2.1 バッファ管理機能

本評価で用いたバッファ管理方式およびパラメータを以下に示す。

置換アルゴリズム	RANDOM, LRU, FIFO, MRU
ハッシュエントリ数	1, 4, 16
ページサイズ	8KB
ページ数	128
置換ページ数†	4, 8, 16
最小未使用ページ数†	4

（†未使用ページ数が最小未使用ページ数以下になると、置換アルゴリズムで置換ページ数だけバッファから追い出し、未使用ページとする ）

4.2.2 データベースモデル

レコード長100bytesとし、１ページに最大64レコードを格納する。データテーブルのページ情報とインデックステーブルのページ情報を示すスキーマは共有メモリに常駐する。インデックスはB-treeとする。

以下の２種類のデータベース構成で評価を行なった。

[データベース１]
Table1 100,000records(1563pages), Index 393pages
Table2 100records(2pages), Index 1page
Table3 10records(1page), Index 1page
[データベース２]
Table1 50,000records(782pages), Index 260pages
Table2 50,000records(782pages), Index 260pages
Table3 50,000records(782pages), Index 260pages

4.2.3 トランザクションモデル

バッファ管理機能に対してページのリード/ライト要求を出し、バッファ上のデータの比較、コピーを行なう疑似トランザクションを作成した。処理内容は以下の３種類である。

[Index search] ・Table1のレコードを索引を用いて検索、更新 ・Table2のレコードを索引を用いて検索、更新 ・Table3のレコードを索引を用いて検索、更新 （索引カラムの更新はしない）
[Full search] ・Table1を全件検索、10%の確率でヒットし更新 ・Table2を全件検索、10%の確率でヒットし更新 ・Table3を全件検索、10%の確率でヒットし更新
[Hash join] ・Table1を全件検索し10%の確率でヒットしたものとTable2を全件検索し10%の確率でヒットしたものを結合する（ジョイン選択率10%）

4.2.4 性能指標

本評価においては性能指標の一例として、図５で示した重み付けから評価値を導きだし、これを元にGood、Badの２つにカテゴリー分けを行なった。

レベル8～10	Good
レベル7～1	Bad

図５：本評価で用いた性能指標

4.2.5 事例データフォーマット

以下の事例データフォーマットを用いた。

第一段階：

（ 評価値 （ アルゴリズム, 置換ページ数, 最小未使用ページ数, ハッシュエントリ数, ページ数, Indexプロセス数, Fullプロセス数, Joinプロセス数, Indexトラヒック, Fullトラヒック, Joinトラヒック,データベース構成））

例：

（Good （LRU 4 4 16 128 2 1 0 1 20000 20000 1））

第二段階目：

（ 最良部品 （ Indexプロセス数, Fullプロセス数, Joinプロセス数, Indexトラヒック, Fullトラヒック, Joinトラヒック, データベース構成））

例：

（LRU-4-4-16-128（2 1 0 1 20000 20000 1））

4.3 評価結果

・二段階の部品変更判定の効果

図６は、第一段階の部品変更ルールを得るために収集した事例データの一部のレーダチャートである。4つの部品が10種類の環境下で付与さ

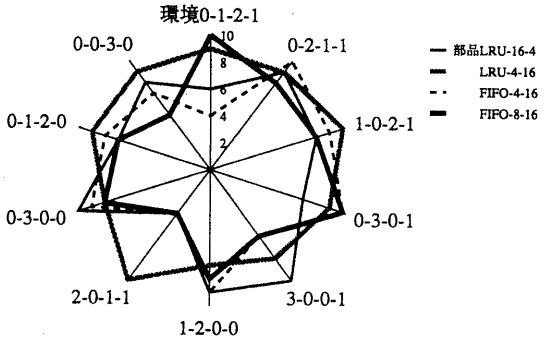


図６：各部品の性能レベル

れたレベルを示す。この4つの部品は、少なくとも１つの環境下で最良な性能が得られると判定されたものである。しかし、その部品が全ての

環境で最良なわけではないことがわかる。

この4種の部品から、最良な部品を学習機構に選択させた結果を図７に示す。幅広い環境において常に良好な性能が得られることがわかる。

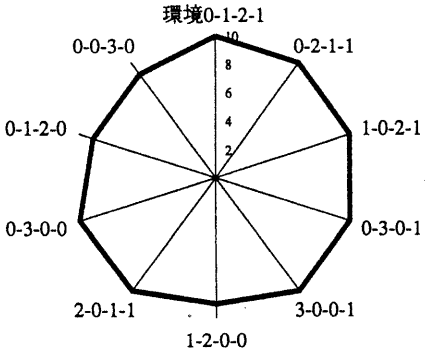


図７：最良な部品の性能レベル

しかし2.2で述べたように、常に最良部品を選択し入れ替えを行なっていてはオーバーヘッドが大きくなる。図８では、環境の変化に伴って、選択された部品がどのように移り変わるかを、常に最良な部品を選択する方法と二段階の選択による方法とで比較した。性能をあまり劣化させずに、部品入れ替え回数を削減できることがわかる。

・未測定環境下での部品変更判定

測定データのない環境における判定の妥当性を調べるために、一部の環境の事例データを用いてレベル判定ルールを作成し、ルールの作成には用いなかったものも含めた全ての環境の事例のレベルを判定させ、正解率を求めた。37の環境で測定した3152件の測定値から、レベル分けのための1110件の事例データを作成した。

図９はルールを作成する時に用いた事例データ数と、作成したルールの正解率である。環境の組合せによって正解率が変化するので、1つの環境の事例データで作成したルールの正解率の分布範囲も矢印で示した。測定環境を適切に選択すれば、少数のデータでも的確な判断を行なえるルールが作成される事がわかる。また、50%の事例データを用いたルールの正解率は70%程度である。これは、今後に運用される環境のバリエーションが倍になったとしても、それまでのルールで70%程度の正解率が得られることを示している。これにより、サービス内容がより多様になったとしても、本機構でルールの変更を伴わずにそのサービスに対処することが可能であると言える。

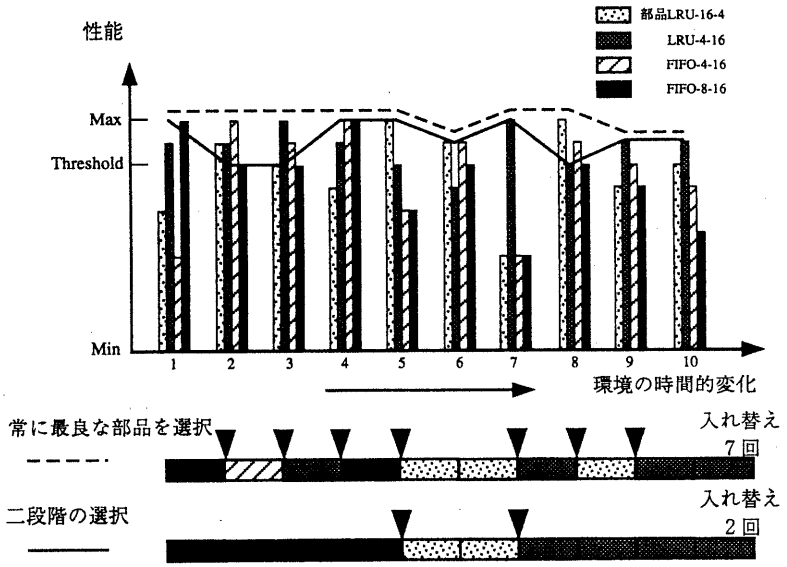


図８：環境の変化と性能レベル

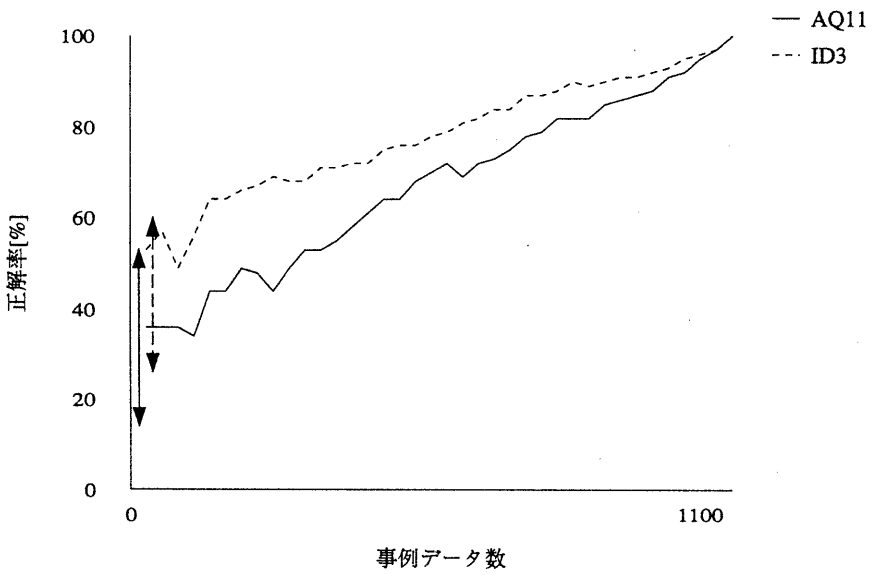


図９：ルール作成の事例データ個数とルールの正解率

5 まとめ

拡張可能DBMSの各機能を実現する部品の選択を分類学習により実行時の環境に合わせて自動的に行なう方式を提案し、DBMSのバッファ管理機能を対象として、プロトタイプを作成し評価を行なった。

バッファ管理機能のように、それまでの処理内容が現在の処理に影響を及ぼす場合、環境の変化の度に部品を変更すると、部品変更のオーバヘッドによりかえって性能が劣化することが予想される。この問題を解決するために、二段階に部品変更の判定を行なう方法を提案した。プロトタイプを用いた評価によって、提案法は性能をあまり劣化させずに、入れ替え回数を削減できることを示した。

また、事例データ数と作成された部品選択ルールの正解率の定量的評価を行ない、測定点を適切に選べば、少数のデータでも的確な判断を行なえるルールが作成できることを示した。

今後は、以下の課題について検討を進める必要がある。

機能部品変更のオーバーヘッド

本稿において、部品変更が頻繁に生じないよう、二段階の判定を行なう事を提案した。部品変更ルールによる判定により部品入れ替え機構を実現し、変更のオーバヘッドや過渡特性を評価する必要がある。そして、変更のオーバヘッドが小さい部品については、第一段階から第二段階への移行を判断する閾値を大きくする方式が考えられる。

機能部品選択ルールの進化

測定値のない新しい環境、もしくは新しい部品の追加や、ルールの精度向上のために、新たに性能値を測定して事例データを作成し、部品選択ルールの進化を行なう必要がある。この進化がシステム運用中において随時、自動的に行なわれるようにする。本検討で得られた事例データを用いて、ルールの進化を模式化したものを付録に示す。

確率を考慮したルールの作成

本稿では事例データの確信度が充分に満足されているものとした。しかし、事例データを作成する際に用いた測定データ数、その分布状況から確率的な要素を組み入れた事例データ、ルールの実現方法も考慮しなければならない。

参考文献

[1] Lohman,G.M., Lindsay,B., etal., " Extexsions to Starburst : Object, Types, Functions, and Rules ", Comm. of ACM Vol.34, No.10, pp.94-109. (1991)

[2] Batory,D.S., Barnett,J.R., etal., " GENESIS : An Extensible Database Management System ", IEEE Trans. Soft. Eng. Vol.14, No.11 pp.1711-1730. (1988)

[3] Aha,D.W., Kibler,D. and Albert,M.K., " Instance-Based Learning Algorithms ", Machine Learning pp.37-66. (1991)

[4] Quinlan,J.R., " Learning efficient classfication procedures and their application to chess end games ", Machine Learning pp.463-482.(1983)

[5] Quinlan,J.R. and Rivest,R.L., " Inferring Decision Trees Using the Minimum Description Length Principle ", Inf. and Comp. Vol.80, No.3, pp.227-248. (1989)

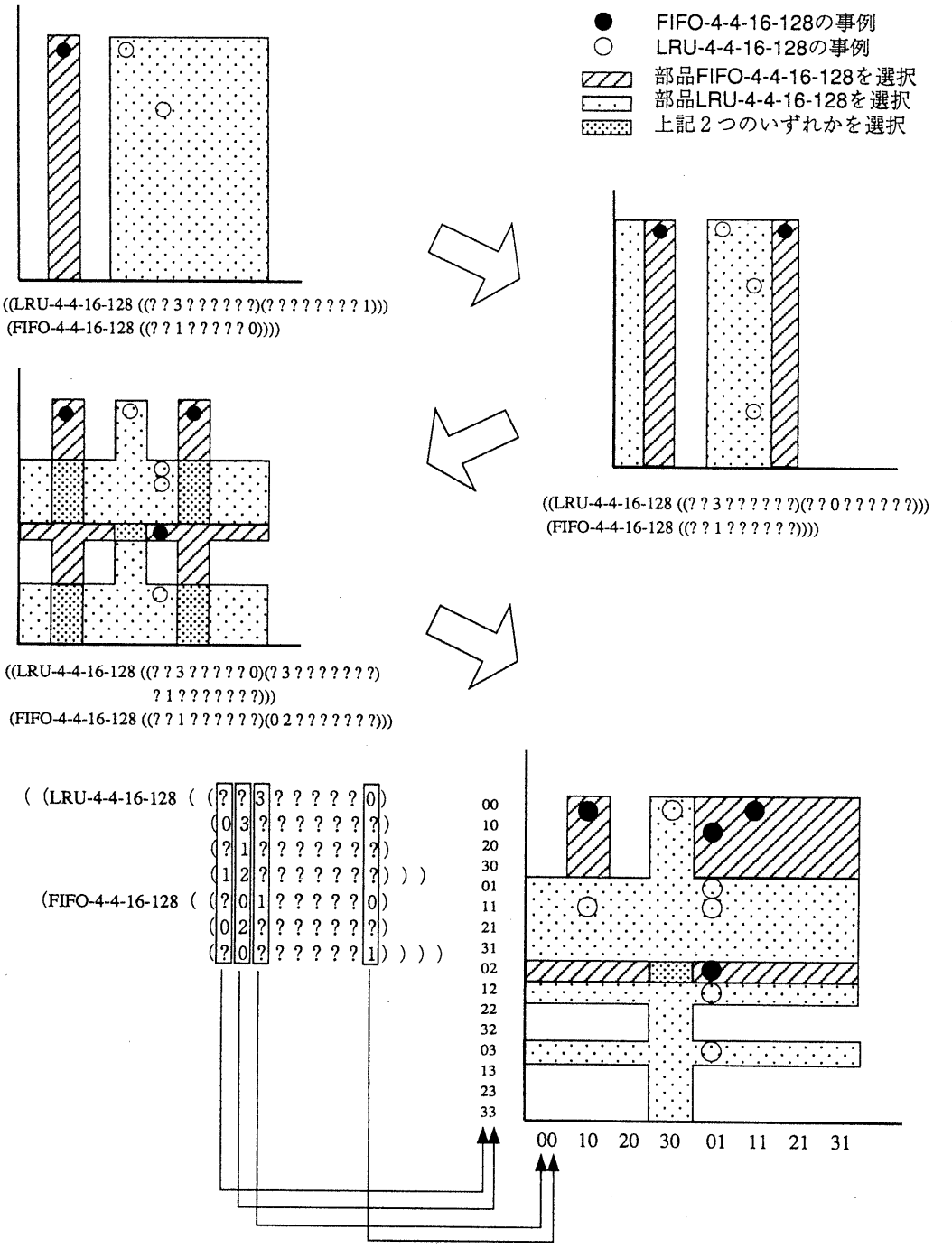
[6] 土屋俊, 他, " AI事典 ", UPU 1988発行

[7] Effelsberg,W. and Haerder,T., " Principles of Database Buffer Management ", ACM Trans. Data. Sys. Vol.8, No.4, pp.560-595. (1984)

[8] 渡辺俊典, " 学習機能を備えた最適化アルゴリズム知識ベース ", 人工知能学会誌 Vol.8, No.3, pp.10-15. (1993)

付録

付図は、使用する事例データの数を変えて機能部品選択ルールを作成し、ルールが進化する過程を模倣したものである。用いた学習アルゴリズムはAq11法であり、二種類のカテゴリーのみの事例データを使用した。



付図：機能部品選択ルールの進化