計算機アーキテクチャ 136－2
（2000．1．7）

遺伝的アルゴリズムを用いたオンライン分岐予測機構の設計

石田 圭太郎† 松崎 元昭†
安藤 秀樹† 島田 俊夫†

これまで，分岐の振る舞いを解析することにより，さまざまな分岐予測機構が提案されてきた．一般的に，分岐の振る舞いは実行する命令列によって変化する．その時々の分岐の振舞いに適した予測方法で予測を行うことによって予測精度の向上を期待することができる．そのためには予測機構が実行中に分岐の振る舞いの変化に追従する必要がある．本稿では，命令列の実行中に予測機構自身の構造を変化させることにより進化するオンライン予測機構を提案をする．オンライン予測機構の枠組には2レベル予測機構を利用し，遺伝的アルゴリズムを用いてマッピング関数を進化させた．実験の結果，提案するオンライン予測機構は，命令列の実行中に分岐の振る舞いの変化に追従し適応し，マッピング関数を予測精度の高い構造へ進化させることができることを確認した．

Online Branch Predictor Designs Using a Genetic Algorithm

KEITARO ISHIDA,† MOTOAKI MATSUZAKI,† HIDEKI ANDO†
and TOSHIO SHIMADA†

A variety of branch predictors were previously proposed by analyzing the behavior of branches. In general,the tendency of branch behavior varies according to the sequence of executed instructions. Branch prediction accuracy can be improved if the prediction scheme can be changed so that it is adapted to branch behavior. To realize this, the branch predictor must follow the changes of branch behavior. This paper proposes an online branch predictor that dynamically adapts to branch behavior by changing its own structure at run time. We use the existing framework of two-level branch predictors and evolve the mapping function with a genetic algorithm. We confirmed through our experiment that the proposed online branch predictor can evolve the structure of the mapping function, leading to higher prediction accuracy, by following and adapting to the branch behavior.

1. はじめに

近年の高性能プロセッサは深いパイプライン構成及びスーパースカラに代表される複数命令の同時発行機構により命令レベル並列性を引き出し性能を向上させている．分岐命令は，こうした命令レベル並列性を利用したプロセッサの性能を厳しく制限する．この影響を緩和するために，分岐命令の結果が判明する以前にその分岐方向を予測し後続命令を投機的に実行するという手法が一般的に用いられている．

分岐予測機構はこの分岐方向の予測を行なう機構である．この予測が誤っていた場合には，投機的に発行したすべての命令の実行を取り消してその分岐命令が実行される以前のプロセッサ状態に戻した上で，新たに正しい分岐方向の制御流に従い命令実行を再開しなければならない．現在においてもパイプライン段数がより深く命令発行幅も広がる傾向にあり，予測に失敗することで破棄される命令数は今後も増加すると考えられる．従って，分岐予測機構の予測正確さは今後ますます重要なものとなる．

過去，さまざまな分岐予測機構が提案されてきた．これらの予測機構は，これまでの研究者らの解析結果から得た分岐の一般的な振る舞いの傾向や規則性に関する知見に基づいて構成されてきた．従って，現状の予測機構よりさらに予測精度の高い予測機構を構成するためには，これまで以上に詳細な分岐の一般的な振る舞いの傾向や規則性を把握した上で，これらを効果的に予測に反映させる仕組みを考える必要がある．しかし，分岐の振舞いの一般的な傾向や規則性を把握するためには，かなりの分量の分岐について一見無秩序にも見える振舞いを丹念に観察する作業を必要とする．これは，人間が行なうには大変な困難を伴う作業である．

この一連の作業を遺伝的アルゴリズム(Genetic Algorithm:GA)を用いて自動化することで，高性能な予

† 名古屋大学大学院工学研究科
Graduate School of Engineering, Nagoya University

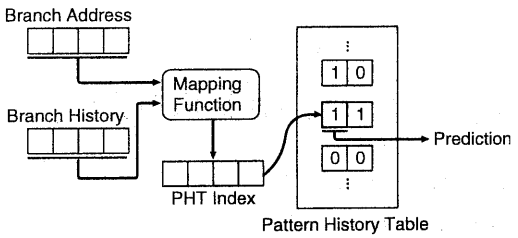

図1　2レベル予測機構の仕組み

測機構の生成を試みた例がある[1)2)3)]．文献1)では実装可能な予測機構の生成には成功していない．また，予測性能についても既存の予測機構を上回るには至らなかった．これは分岐予測機構全体を一つの遺伝子として表現するのは，GAによって現実的な解を求めるには問題規模が大きすぎるためと考えられる．一方，文献2)3)では，遺伝子構造を既存の主要な予測機構である2レベル予測機構の枠組を利用し遺伝子構造を大幅に簡素化することで，高性能でかつ実装可能な予測機構の生成に成功している．

しかしながら，文献2)3)では特定の実行命令列を繰り返し用いてGAの学習を行なうため，GAの学習に用いた命令列の分岐の振る舞いに特化した予測機構を生成しているといえる．一般的に，分岐の振る舞いは実行する命令列に伴い変化する．そのため特定の命令列についてGAの学習を行なって自動生成した分岐予測機構では，未学習の命令列に対しては有効性が低下する．

また分岐の振る舞いの変化によって最適な予測方法も変化する．そのため，一つの予測方法で予測を行ない続けるのではなく，その時々に適した予測方法で予測を行うことによって予測精度の向上を期待することができる．実行する命令列によって異なる分岐の振る舞いに対し予測方法を適応させるためには，予測機構が実行中に分岐の振る舞いの変化に追従し，動的に予測方法を変化させることが必要である．本稿では，実行する命令列に対し動的に適応可能な予測機構を提案する．

2章では分岐予測機構及び分岐予測機構の自動生成に関する過去の研究について述べる．3章では我々の提案する分岐予測機構の詳細について述べる．4章では我々の提案する分岐予測機構の評価及び考察を行なう．5章では本稿をまとめる．

2. 関連研究

2.1 2レベル予測機構

現在の分岐予測方式の主流となっているものが，**2レベル予測機構**[6)7)]と呼ばれるものである．**図1**にこの機構の概要を示す．この機構は，分岐命令の命令アドレスと，過去の分岐履歴の二つの情報をインデックスとして，**パターン履歴テーブル**（PHT:Pattern History Table）中の対応する2ビットカウンタを参照しこの最上位ビットの値を予測値として出力する．分岐結果が判明するとこの情報が予測機構へフィードバックされ，分岐履歴を保持するシフトレジスタの更新及び2ビットカウンタの値の増減を行ない自らの蓄える情報を更新する．

分岐の履歴の取り方には，ローカル履歴と呼ばれる方法とグローバル履歴と呼ばれる方法の2つがあるが，ここではグローバル履歴についてのみ簡単に述べる．グローバル履歴は，命令アドレスで区別することなく分岐の履歴を単一のシフトレジスタ上に記録する．従って，この履歴には予測を行う分岐以外の分岐の履歴も同じシフトレジスタ上に記録されている．グローバル履歴を用いた予測機構はif-then-elseのような制御の流れにある規則性を抽出し予測を行うことができる．

マッピング関数は，単純にアドレスのビット情報と，分岐履歴のビット情報をつなぎあわせる形でPHTのインデックスを決定するものと，簡単なハッシュ関数を用いてPHTのインデックスを決定するものの2種類の手法がある．後者のハッシュ関数を用いる代表的な予測機構に**gshare**[4)]がある．この予測機構は，グローバル履歴とアドレスのビットとのXORを取り，これをPHTのインデックスとするものである．

2.2 遺伝的アルゴリズム

GAは生物の進化過程を模倣した確率的探索アルゴリズムである．このアルゴリズムは準最適解を求めるのに適しており，さまざまな応用が研究されている．

GAでは最初に，個体と呼ばれる解候補をランダムに複数作成する．それぞれの個体は，**適応度**と呼ばれる問題に対する評価値が算出される．そして，この適応度をもとに遺伝的操作を行ない新たな個体を生成する．その後は，生成された個体について適応度を求め遺伝的操作による新たな個体生成を繰り返す．こうして，与えられた問題に対する評価の高い解を得ることができる．

遺伝的操作には，**選択・淘汰，交叉，突然変異**の操作がある．選択・淘汰は個体群から任意の個体を選び出す操作である．この操作は，適応度の高低を加味した確率的操作となる．選択されなかった個体は，結果的にその構造を次の世代に反映させることができず淘汰されることになる．交叉は，選択された個体のデータを部分的に入れ換える操作を行なう．これは，生物界における遺伝子の交叉を模倣したものであり，そのため個体の持つデータは遺伝子と呼ばれる．突然変異は，個体の遺伝子をランダムに変化させる処理であり，個体の多様性を確保するため行なう．

3. 提案する分岐予測機構

3.1 提案する分岐予測機構の詳細

我々の提案する分岐予測機構の詳細について述べる．提案する予測機構の動作の目的は，分岐の振る舞いの変化に追従し，命令列を実行中に動的に予測方法を適応させることである．提案する予測機構は，命令列を実行中に予測方法を適応するためにGAの進化計算をオンラインで処理する．文献1)2)3) の分岐予測機構自動生成ではオフラインのGAで進化するのに対し，今回我々の提案する予測機構ではオンラインのGAで進化する．以下では提案する予測機構を**オンライン予測機構**と呼ぶこととする．

予測方法の適応方法は，GAを用いてマッピング関数を実行中のプログラムに追従して変化させることにより実現する．2レベル予測機構の性能を決定する構成要素はいくつかあるが，その中で分岐の命令アドレス及び分岐履歴を入力からPHTのインデックスを決定するマッピング関数は，PHTの大きさやプログラムによって，最適な構造が変化するため一意に決定するのは難しい．また，その設定は予測機構の性能に大きな影響を与える．そこで，マッピング関数を論理式で表記し遺伝子として与え，これにGAを適用することとした．

図2にオンライン予測機構の構成図を示す．オンライン予測機構は**遺伝的アルゴリズム専用ハードウェア** (GAエンジン) と個体数分のPHT及びマッピング関数回路によって構成されている．GAエンジンとはGAの処理に特化し高速動作可能な専用ハードウェアである．オンライン予測機構内でのGAの処理は命令列の実行と同時に行なうので，専用ハードウェアにより高速に処理する必要がある．

GAエンジンから生成される個体の遺伝子情報を用いてマッピング関数回路を構成し，それぞれのマッピング関数回路に対応したPHTを参照して分岐予測を行なう．GAエンジンには，個体の適応度を計算するためにそれぞれの個体の分岐予測精度を入力する．それぞれの個体の適応度計算に用いる分岐予測精度は，分岐結果及びそれぞれの個体の分岐予測からカウンタを用いて計算する．マイクロプロセッサに出力する分岐予測は，GAエンジンがどの個体を使用するか決定し，マルチプレクサによって選択する．

3.2 遺伝的アルゴリズムの適用

遺伝的アルゴリズムとして，単純GAを用いた．以下では，GAのオンライン予測機構への適用の詳細について述べる．

3.2.1 遺伝子構造

個体の遺伝子構造について述べる．遺伝子はマッピング関数として機能する論理回路を表現するものでなくてはならない．ここでは遺伝子構造として完全2分木型の構造をとった．

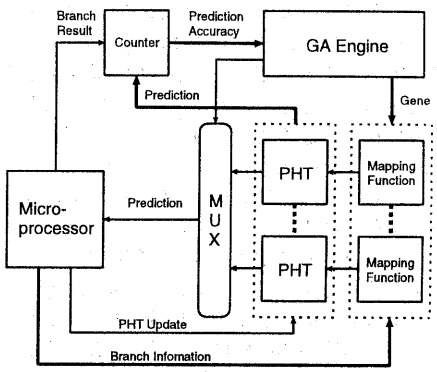

図2 オンライン予測機構の構成

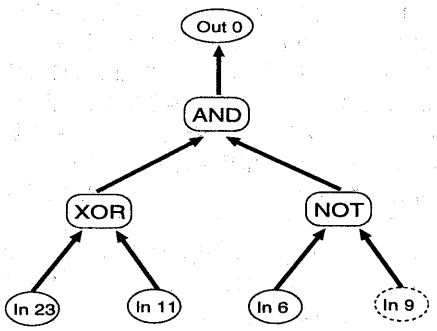

図3 マッピング関数の遺伝子構造

マッピング関数の出力の各ビットのそれぞれが図3のような2分木構造を持つ．2分木の各ノードは，AND，OR，XOR，NOTの4種の論理ゲートないしは，1ビットの入力であるInのいずれかを取る．これらのノードのうちNOTは1入力1出力のノードであり，またInは他のノードからの入力が全くない0入力1出力のノードである．遺伝子上ではこれらのノードの種類によって構造を変化させるようなことはせず，1入力の場合は片方の入力を，0入力の場合は両方の入力の値を無視することで，その機能を実現する．そのため，2分木構造に記述されたノードの中には図3の点線部分のノードのように全く使用することのないノードも含まれる．

実装の難しい複雑な論理回路を生成することを防ぐため，本実験ではノードの深さを2段までに制限することにした．

3.2.2 オンライン予測機構進化の流れ

オンライン進化の概略について述べる．オンライン予測機構の進化はプロセッサの動作と並行して行う．

オンライン予測機構中でのGAの動作について説明する．GAの一世代は，個体評価フェーズと遺伝的操作フェーズにより成り立つ．個体評価フェーズでは，一定の実行命令の区間において，個体群の適応度を計

算する．遺伝的操作フェーズでは交叉，突然変異，選択を行ない次世代のための個体群を生成する．分岐予測機構の出力する予測には，前世代の最大の適応度を示した個体の予測を用いる．このように命令列の実行に伴って世代交替を繰り返す．

3.2.3 個体群

個体群について述べる．通常の GA では初期個体群はランダムに生成する．しかしオンライン予測機構においてランダムに生成された性能の悪い可能性のある初期個体群を用いると，それらを進化させるための期間では低い予測精度で予測を行なわなくてはならない．そこで初期個体群にはランダムに生成された個体のほかに，あらかじめ作成した次の２つの個体を挿入した．

- gshare 予測機構のマッピング関数と等価な個体
- オフラインの進化によって求めた個体

gshare 予測機構のマッピング関数と等価な個体は，gshare 予測機構のマッピング関数を遺伝子表現したものである．オフラインの進化によって求めた個体は，文献 2)3) の分岐予測機構自動生成法に基づき，学習用に用意した実行トレースを用いて，繰り返し学習を行なうことにより生成した．

これらの２つの個体は，遺伝的操作によって破壊しないこととした．個体群はこれらの２つの個体の影響を受け，有用な構造を採り入れることができる．また仮に新たに生成された個体群の性能がこれら２つの予測機構に比べて低い場合でも，オンライン予測機構の予測精度の下限をこれらの個体によって保証する．

3.2.4 遺伝的操作

遺伝的操作の詳細について述べる．GA の遺伝的操作には，選択・淘汰，交叉，突然変異の操作がある．個体の選択方式には，個体の適応度の高さに応じて確率的に選択するルーレット選択方式を利用した．それとともに最も高い適応度を持つ個体は破壊せず必ず保存するエリート保存戦略を用いた．個体の交叉には，一様交叉を用いた．一様交叉による新たな個体の生成では，選択された２つの個体の遺伝子に対して一様に任意のノードを入れ替える．突然変異の操作は遺伝子交叉におけるノードの入れ替えの際に一定確率でランダムなノードに置き換える形式で行った．この突然変異操作とは別に，低確率で出力ビットに対応する木に属するノードのすべてがランダムに書き換えられるような突然変異処理も行った．

3.2.5 個体の評価

個体の適応度には分岐予測シミュレーションから得られる予測精度を利用した．適応度は，アルゴリズムがより効率的に働くために，予測精度に対し線形スケーリングしたものを用いた．線形スケーリングの手法は，Goldberg[5] の提案する以下の手法である．
個体の予測精度 f，求める個体の適応度 f' とする．
個体 $i = 1, 2, \ldots, N$ に対して

$$f_i' = a \cdot f_i + b$$

$$a = \begin{cases} \dfrac{(c_{mult}-1.0)\cdot\overline{f}}{f_{max}-\overline{f}} & f_{min} > \dfrac{c_{mult}\cdot\overline{f}-f_{max}}{c_{mult}-1.0} \\ \dfrac{\overline{f}}{\overline{f}-f_{min}} & f_{min} \leq \dfrac{c_{mult}\cdot\overline{f}-f_{max}}{c_{mult}-1.0} \end{cases}$$

$$b = \begin{cases} \dfrac{\overline{f}\cdot(f_{max}-c_{mult}\cdot\overline{f})}{f_{max}-\overline{f}} & f_{min} > \dfrac{c_{mult}\cdot\overline{f}-f_{max}}{c_{mult}-1.0} \\ -\dfrac{f_{min}\cdot\overline{f}}{\overline{f}-f_{min}} & f_{min} \leq \dfrac{c_{mult}\cdot\overline{f}-f_{max}}{c_{mult}-1.0} \end{cases}$$

ここで個体群の平均適応度 $\overline{f}$，最大適応度 f_{max}，最小適応度 f_{min} である．c_{mult} は最良の個体が次世代に残す個体数の期待値であり $c_{mult} = 2.0$ とした．

それぞれの個体はマッピング関数の論理回路を表現するが，これら全てを命令列を実行中に同時に評価するためにそれぞれの個体ごとに PHT を持つ．個体が遺伝的操作によって変化したとき新しい個体がその PHT を使用するため，それまでその個体のマッピング関数によって学習した PHT の状態をフラッシュする．しかし，前述のように gshare 予測機構のマッピング関数と等価な個体，オフラインの進化によって求めた個体は破壊せず必ず保存するためそれらの PHT はフラッシュしない．

PHT をフラッシュする個体については，PHT の学習の間一時的に予測精度が低下する．我々は，PHT の学習期間に相当する世代交代後の一定期間の予測精度については，GA の適応度計算に含めないこととした．

4. オンライン予測機構の動作検証及び考察

4.1 オンライン予測機構の実行シミュレーション

オンライン予測機構を評価するための実行シミュレーション及び考察を行なった．

オンライン予測機構の利用できる PHT は 8K エントリ（インデックス長１３ビット）とし，分岐履歴を２０ビット，分岐の命令アドレスを１３ビット利用できるとした．この予測機構のマッピング関数は，３３入力１３出力の回路となる．

ベンチマークは SPECint95 の m88ksim, vortex, li, perl, compress95, gcc, ijpeg, go の８本を用いた．使用するトレースはそれぞれベンチマークのはじめの８０００万命令とし，全ベンチマークプログラムを連続して実行した．ベンチマークが変わっても個体群は初期化せず，前のベンチマークによって学習した個体群をそのまま用いることとした．１世代の評価フェーズは１２５万命令である．よって１本のベンチマークにつき６４世代実行することになる．

オンライン予測機構は GA エンジン及びそれぞれの個体のもつ PHT を必要とする．個体数はそのハードウェア的な制限をふまえ，８個体とごく少数に設定した．個体群に挿入する gshare 予測機構のマッピング関数と等価な個体は，PHT のインデックス長１３ビット，分岐履歴長７ビットの gshare 予測機構のものを使用した．ここで，分岐履歴長の７ビットは，今

回用いたトレースに対してベンチマーク平均で最も予測精度の良い分岐予測を行なう最適値であり，実験により求めた．また，同じく個体群に挿入するオフラインの進化によって求めた個体は，前述の SPECint95 の８本のベンチマークのそれぞれの特定の部分によって求めた．ここでオフラインの進化の繰り返し学習には膨大な時間を要するため，オフラインの進化の学習に使用するトレースは，それぞれのベンチマークのはじめの４００万命令とした．この個体は，世代数１００００世代まで進化を行なった１００個体のうちの最優秀な個体で，適応度計算にはベンチマーク平均予測精度を用いた．

表１ に m88ksim, vortex, li, perl, compress95, gcc, ijpeg, go の順序で測定したベンチマークごとの予測ミス率を示す．表中の Ga はオンライン予測機構の予測ミス率を示す．Ga_off は，個体群中のオフラインの進化によって求めた個体の予測ミス率を示している．この個体は破壊されることはなくその PHT もフラッシュすることはないので，Ga_off はそれぞれのベンチマークのそれぞれのはじめの４００万命令のトレースを用いて自動生成した分岐予測機構のそれぞれのベンチマークの８０００万命令のトレースに対する予測ミス率である．

gshare は個体群中の gshare 予測機構のマッピング関数と等価な個体を示している．これも同じように個体は破壊されることはなくその PHT もフラッシュすることはないので，gshare 予測機構のそれぞれのベンチマークの８０００万命令のトレースに対する予測ミス率である．

表１　ベンチマーク別の予測ミス率

Benchmark	Ga(%)	Ga_off(%)	gshare(%)
m88ksim	1.4363	1.4391	1.4363
vortex	1.9411	4.3323	1.9489
li	5.9537	5.9509	6.8500
perl	7.8368	8.2552	8.1062
compress95	7.1916	7.2314	8.0796
gcc	8.6035	10.5253	8.7588
ijpeg	11.1701	11.7276	11.1858
go	19.2603	28.1822	20.1724
average	7.9242	9.7055	8.3173

表１から分かるようにオンライン予測機構は，gshare に対して compress95, li, go では約 1%程度の予測精度を向上できている．その他は，同程度もしくは 0.3%程度精度向上しており，平均では約 0.4%程度精度向上できている．

図４ と**図５** に進化の様子の例として go と perl のベンチマークの予測ミス率の推移を示す．ベンチマーク go の例をみると，オンライン予測機構は５世代目から gshare と比べてより適応した解を発見し，予測精度が向上している．一方ベンチマーク perl の例で

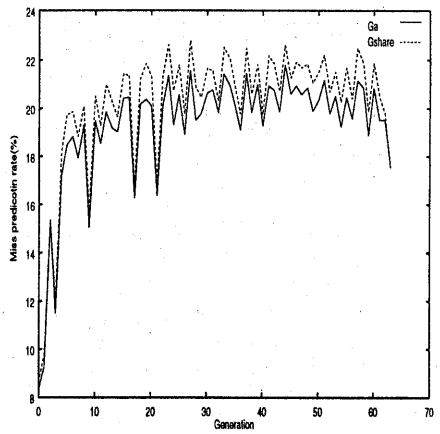

図４　進化の例 (go)

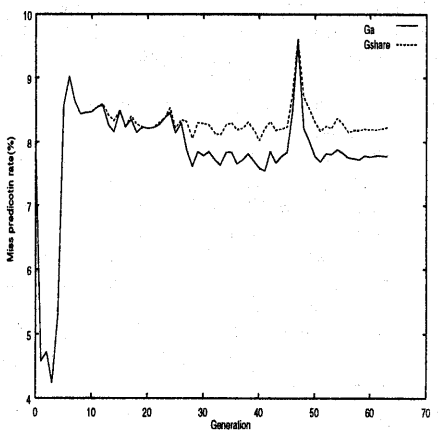

図５　進化の例 (perl)

は２５世代までは gshare よりも良い解を発見することができずほぼ同程度の予測精度を示しているが，それ以後から４７世代までは予測精度が向上している．４７世代では gshare と同じ予測精度まで落ちるがそれ以後はまた予測精度を向上できている．

マッピング関数が変化することのない gshare 予測機構やオフライン進化によって自動生成した予測機構と比較すると，提案するオンライン予測機構は実行するトレースの始めの部分においては gshare 予測機構を上回ることができないが，その後は GA の学習によって予測精度が向上することができている．このことから，オンライン予測機構はマッピング関数を予測精度の高い構造へ適応していることが分かる．

4.2　ベンチマーク実行順による予測精度の測定

個体群は過去の進化の過程で得た優秀な個体の情報を継承し探索のために役立てている．しかしオンライン予測機構の適応すべき分岐の振る舞いは命令列の実行に応じて様々に変化するため，必ずしも過去の学習の過程で得た情報が有効であるとは限らない．そのた

表2　ベンチマーク実行順による予測ミス率

Benchmark	ga(%)	gshare(%)
順序1	7.9242	8.3173
順序2	8.0163	8.3193
順序3	8.0237	8.3214
順序4	8.0530	8.3202
順序5	7.9810	8.3187

めオンライン機構の予測精度は，過去に適応したベンチマークによって別の進化をたどるため，同じベンチマークであっても異なる．

そこでベンチマーク実行順による予測精度の変化を測定した．まず，gshare 予測機構 (PHT インデックス長 13 ビット，履歴長 7 ビット) によってそれぞれのベンチマークの予測ミス率を測定し，それによってベンチマークに順番を付けた．予測ミス率によってベンチマークに昇順に順番をつけると，m88ksim, vortex, li, perl, compress95, gcc, ijpeg, go となる．gshare 予測機構のそれぞれのベンチマークの予測ミス率を，ベンチマークの特徴の一つと考え，以下の 5 つの順序で実行した際のオンライン予測機構の予測ミス率を測定した．

(1) 予測ミス率が徐々に増える順序 (m88ksim, vortex, li, perl, compress95, gcc, ijpeg, go)
(2) 予測ミス率が徐々に減る順序 (go, ijpeg, gcc, compress95, perl, li, vortex, m88ksim)
(3) 予測ミス率が増減する順序 (m88ksim, compress95, voretx, gcc, li, ijpeg, perl, go)
(4) 予測ミス率の変化の度合が，前半は小さく後半は大きい順序 (m88ksim, go, vortex, ijpeg, li, gcc, perl, compress95)
(5) 予測ミス率の変化の度合が，前半は大きく後半は小さい順序 (perl, compress95, li, gcc, vortex, ijpeg, m88ksim, go)

表 2 にベンチマーク実行順による予測ミス率を示す．順序 1 が最も予測ミス率が低く，順序 4 が最も予測ミス率が高かった．今回測定した全てのベンチマーク実行順の予測ミス率では，最大で約 0.13%程度の開きが出たが，ほぼ同一の値を示した．

gshare 予測機構の示したそれぞれのベンチマークの予測ミス率の高低を，それぞれのベンチマークの分岐の振舞いの性質の一つと考えると，今回の実験で測定したベンチマーク実行順によって，分岐の振舞いの変化は全く異なる．このことから，オンライン予測機構は分岐の振る舞いの変化に追従し，適応できていることが分かる．

5. ま と め

これまで，分岐の振る舞いを解析することにより，さまざまな分岐予測機構が提案されてきた．一般的に，分岐の振る舞いは実行する命令列によって変化する．その時々の分岐の振舞いに適した予測方法で予測を行うことによって予測精度の向上を期待することができる．そのためには予測機構が実行中に分岐の振る舞いの変化に追従する必要がある．本稿では，命令列の実行中に予測機構自身の構造を変化させることにより進化するオンライン予測機構を提案をした．オンライン予測機構の枠組には 2 レベル予測機構を利用し，遺伝的アルゴリズムを用いてマッピング関数を適応させた．実験の結果，提案するオンライン予測機構は，命令列の実行中に分岐の振る舞いの変化に追従し適応し，マッピング関数を予測精度の高い構造へ進化させることができることを確認した．

謝辞　本研究は，財団法人名古屋産業科学研究所の「創発型ソフトコンピュータの開発」のプロジェクトの一部として行なわれた．また，本研究の一部は，文部省科学研究費補助金基盤研究 (C)「広域命令レベル並列によるマイクロプロセッサの高性能化に関する研究」(課題番号 1068034)，文部省科学研究費補助金基盤研究 (C)「分岐予測と投機的実行に関する研究」(課題番号 11680351)，および財団法人カシオ科学振興財団研究助成「高性能マイクロプロセッサのアーキテクチャに関する研究」の支援により行った．ここに感謝の意を表す．

参 考 文 献

1) J.Emer and N.Gloy : A Language for Describing Predictors and its Application to Automatic Synthesis, In *Proc. 24th Annual International Symposium on Computer Architecture*, pp.304-314(1997).
2) 野口良太, 松崎元昭, 小林良太郎, 安藤秀樹, 島田俊夫 : 遺伝的アルゴリズムを用いた分岐予測機構設計, 電子情報通信学会研究報告 97-CPSY-107, pp.45-50(1998).
3) 野口良太, 松崎元昭, 小林良太郎, 安藤秀樹, 島田俊夫 : 遺伝的アルゴリズムを用いた分岐予測機構設計, 計測自動制御学会論文集 Vol.35 No.11,pp.1431-1437(1999).
4) S.McFarling : Combining Branch Predictors, *WRLTechnical Note TN-36*, Digital Equipment Corporation(1993).
5) D.E.Goldberg : *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley(1989).
6) T-Y.Yeh and Y.Patt : Two-Level Adaptive Branch Prediction, In *Proc. 24th Annual International Symposium and Workshop on Microarchitecture*, pp.55-61(1991).
7) T-Y.Yeh and Y.Patt : A Comparison ofDynamic Branch Predictors that use Two Levels of Branch History, In *Proc. 20th Annual International Symposium on Computer Architecture*, pp.257-266(1993).
8) 北野宏明 : 遺伝的アルゴリズムの基礎, 北野宏明編「遺伝的アルゴリズム」産業図書 (1993).