

並列配線問題における並列引き剥し再配線処理の品質改善効果

佐野 雅彦[†] 高橋 義造^{††}

VLSI 等のパターン作成における配線問題は膨大な計算時間を要するため、並列処理による計算時間の短縮が期待され多くの研究が行われている。しかしこれまでのところ、並列処理では一部を除いて従来の逐次処理方式に比肩する配線品質が得られていないのが実状である。我々は既配線の引き剥し再配線を並列に行う方法を研究し、従来の並列配線方式より優れた配線品質を得ることができた。本論文で提案する並列経路改善方式では、従来の並列配線方式における逐次的な配線順序とは異なる配線順序を用いており、配線コストを用いた経路探索アルゴリズムと評価方法により配線問題に対する配線順序の依存性を抑え、複数既配線の並列引き剥し再配線処理により配線品質を並列に改善する。また、並列処理方式として我々が以前に開発したプロセッサ競合方式とネット割り当て法を使用しており、並列計算機のアーキテクチャに対する依存性を抑えている。本論文では、並列経路改善方式のアルゴリズムと、実際に並列計算機を用いて行った評価の結果について考察し、本方式の有効性について考察した結果について述べる。

Effect of Parallel Ripup and Rerouting to the Wiring Quality in Parallel Wire Routing Problems

MASAHIKO SANO[†] and YOSHIZO TAKAHASHI^{††}

Since the problem of generating routing patterns for VLSI or printed circuit board routing requires huge computing time, parallel computing methods for wire routing problems have been developed in order to shorten the processing time. In general, the quality of routing patterns generated by parallel methods is worse than that of conventional sequential wire routing methods. From this background, we have developed a new parallel wire routing method by using rip-up and reroute techniques in order to achieve higher wiring quality than conventional sequential methods. The Parallel Path Improvement Method, as described in this paper uses a net wiring order different from of sequential routing method. This method features a concurrent path improvement mechanism by using parallel rip-up and rerouting techniques with cost-based path searching algorithm, and an evaluation method that is less dependent on the wiring order. The competing processor and net allocation method, developed in our previous research, was employed as a parallel computing model. This model reduces dependency on the architect of the parallel computer to be used. In this paper, we describe details of the parallel path improvement method and show the results evaluation results using a MIMD parallel computer.

1. はじめに

配線問題の複雑化による計算時間の増加は、高品質な配線結果を実用的な時間で得ることを困難にする。幸いにして、並列計算機の発展による計算時間の短縮

が可能となった現在では、幾つもの並列処理方法が提案されている。それらの多くは配線処理中で最も時間を必要とする経路探索を並列処理するものである。しかし経路探索以外では配線情報の共有と配線順序の問題から逐次処理であることが多く、並列度を制限する原因となる。そのため、並列度を向上させるには従来の逐次アルゴリズムとは異なるものが必要であり、また並列計算機資源の有効活用の点から、計算機方式に対する依存性の低い並列処理アルゴリズムの開発が望まれる。このような背景から、我々は計算機方式に対する依存性の低い並列処理アルゴリズムを問題分野別

[†] 徳島大学大学院工学研究科
Graduate School of Engineering, Tokushima University

^{††} 徳島大学工学部知能情報工学科
Department of Information Science and Intelligent Systems, Faculty of Engineering, Tokushima University

に研究している。その一例に配線問題の並列処理があり^{3,4)}、我々は、プロセッサ競合方式とネット割り当て法を用いた並列処理アルゴリズムにより、速度向上率に対する有効性を確認している^{3,4)}。しかしながら配線品質において不十分であることが残された課題であった。

並列配線処理において配線速度より配線品質を優先させる場合、例えば複雑な配線問題を限られたスペースに収めることを考慮すると、これまでの逐次方式よりも高い配線品質で、少なくとも逐次方式よりも短い時間で処理できることが望ましい。なぜなら、逐次方式と同じ配線手法に従う並列配線方式では配線順序が同じであるため、得られる配線品質は逐次方式と同等^{5),6)}かそれ以下になるからである。そのため、逐次方式と同じ配線結果を得るには十分であるが、それ以上の配線品質を求めるには並列配線方式に適合する配線方式を検討すべきである。このための配線方式には、逐次方式では実用的時間で計算不可能な発見的手法、探索的手法が有効であると考えられる。そしてその処理時間を実用的な時間に短縮するために並列処理が必要である。このような処理方式により配線品質を向上させた例として MAPLE-RP^{7),8)} (以下、RP) がある。これは配線コストを制約条件に用いた経路探索アルゴリズムをルータと呼ばれる専用並列処理装置を用いて並列化したもので、その配線品質は一般的な逐次方式より優れている。

我々の研究目標は、配線問題において逐次方式よりも高い配線品質を得ることができるとする汎用並列計算機のための並列配線方式の開発である。その試みとして、プロセッサ競合方式において課題であった配線品質の改善に対して本論文で提案する並列経路改善方式を用いた結果、配線品質の改善とその台数効果を確認した。この並列配線方式は、複数ネットの引き剥し再配線処理を並列に処理する点で、これまでの逐次配線順序による並列配線方式と異なる。

本論文の構成は、2章で我々が以前に開発したプロセッサ競合方式による並列配線について説明し、3章では提案する並列経路改善方式について述べる。4章では MIMD 型並列計算機による実験結果を示し、5章でその考察をする。そして最後にまとめる。

2. プロセッサ競合方式による並列配線処理

プロセッサ競合方式 (以下、競合方式) による並列配線では、汎用並列計算機に対する依存性を抑えた実

装容易な配線方式を目標としており、複数の配線処理を複数のプロセッサにより同時に処理する方式を使用している^{3),4)}。競合方式の構成を図 1、基本処理の流れを図 2 に示す。これは一台のマスタプロセッサ (以下マスタ) と複数のスレーブプロセッサ (以下スレーブ) で構成される MIMD 型並列計算機を想定しており、ハイパーキューブ、多進木、格子結合などの多様な結合網に対する依存性を考慮して、以下に示すモデルの仮定を行っている⁹⁾。

- 各スレーブとマスタは一対一で通信する。
- 各スレーブは互いに独立して動作する。
- 放送機構を使用しない。
- 共有メモリを使用しない。

競合方式では次のように並列配線処理を実現している。まず、処理の割り当て方法としてネット割り当て法を使用しており、最小処理単位を 1本のネットとし

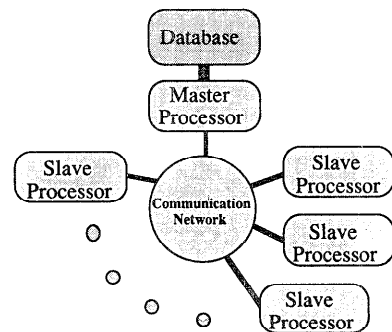
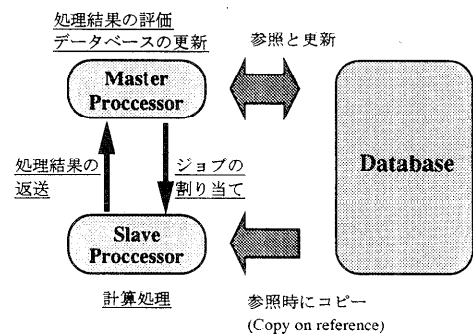


図 1 プロセッサ競合方式のモデル
Fig. 1 A model of competing processors method.



注) スレーブにおける Database の参照は、参照時コピー (Copy on reference) 方式で行われる。

図 2 プロセッサ競合方式における基本処理の流れ
Fig. 2 Processing flow of competing processors method.

* 基本的に計算機 OS に依存しない処理モデルのため、PVM 等の各種 OS 上での実装が簡単である。

て一台または複数台の処理プロセッサに割り当てる方法である。ネット間の並列性を考慮しておりネットの独立性が高いほど並列性が向上する。そして並列処理可能なネットは実行時に評価される。競合方式ではマスタが複数のスレーブに異なるネットを割り当てて並列に配線処理させ、その配線経路をマスタが先着順に評価*する。次に、評価結果を用いてデータベースを更新し、次のネットを割り当てる。各スレーブの処理時間の差とスレーブ間の競合から起きる配線経路の矛盾（経路の交差や接触）は、先に評価された経路を優先し、後から評価される経路を最新の配線情報を用いて再配線することにより解決する。そのため、他のスレーブに同期、通信等による影響をほとんど与えない。

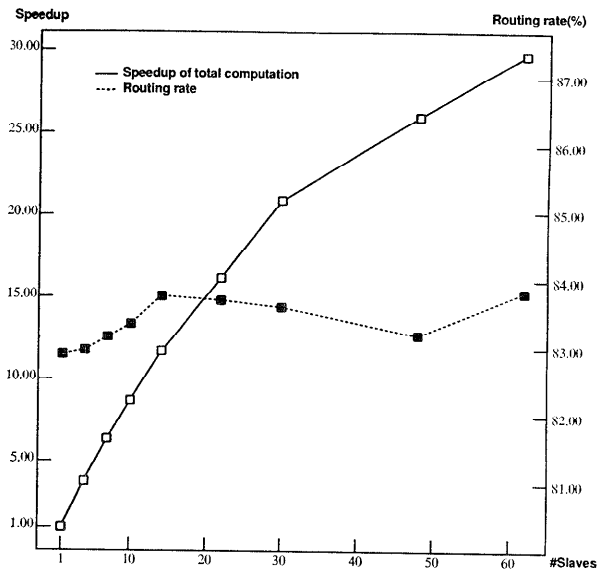
我々が 1991 年に行った、競合方式による並列配線方式を MIMD 型並列計算機 Coral-68K⁹⁾ で評価した場合の結果を示す⁴⁾。評価に用いた配線問題はグリッドサイズ 256×256 の 2 層配線問題で 3000 本のネットを持つ。図 3 (a) に速度向上率と配線率、図 3 (b) にスレーブ 62 台の場合の処理ネット数に対する全体の配線率と再配線率を示す。ただし、

再配線率 (%)

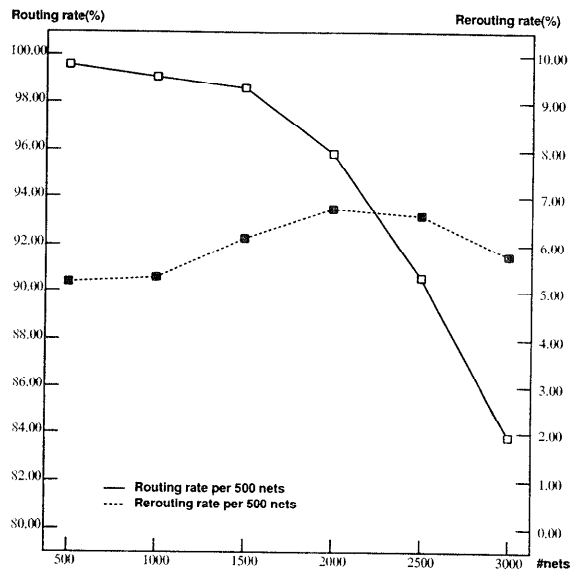
$$= \frac{\text{引き剥がし再配線回数}}{\text{総配線回数}} \times 100$$

評価の結果、速度向上に関しては有効であることが確認された。しかし、同じネットデータを用いて逐次処理方式の商用ルータである CADNETIX と比較した結果、競合方式の 82% に対して 96% となり、配線率に関して不十分であることが確認された。この理由は、まず、競合により発生する矛盾の解決を再配線処理だけで行ったことにある。そのため、配線済みネットが多くなると再配線のための経路が存在しなくなり、配線率が低下する結果となった。次に、競合方式では配線順序が一意に定まらないために、逐次配線処理を前提とした配線経路の評価方式が不相当であった。

* マスタがボトルネックになると、スレーブはネットの割り当てを待つことになる。



(a) 速度向上比および配線率とスレーブ数の関係
(a) Speedup and routing rate vs. number of slaves.



(b) 処理ネット数に対する配線率と再配線率
(b) Routing and rerouting rate per 500 nets vs. number of nets.

図 3 競合方式による評価

Fig. 3 Evaluation by competing processors method.

3. 並列経路改善方式

前述のプロセッサ競合方式で得られた経験から、配線順序に対する依存性の低い経路探索と評価方法による引き剥がし再配線処理の反復が、汎用並列計算機のため

めの配線方式として有効であると判断し、ここに並列経路改善方式を提案する。この方式はプロセッサ競合方式とネット割り当て法を用いた並列配線方式に、配線品質改善のための経路改善機構を取り入れており、複数の既配線経路の引き剥し再配線の反復を並列処理することにより並列経路改善を行うものである。また、配線経路に交差・接触を許容することで、競合方式により発生する矛盾を経路改善処理により解消する。

以下ではこれらの詳細を述べる。なお、本稿では並列配線方式を配線順序によって2つに分類し、それぞれ同順方式、異順方式と呼ぶことにする。同順方式は従来の逐次配線方式と同じ配線順序で並列配線処理するものであり、逐次方式と同じ配線結果と配線品質を目的としている。現在多くの並列配線方式がこれに属し、最近では PROTON⁵⁾、タイムワープによる並列無格子配線システム⁶⁾（以下タイムワープ方式）等がある。これに対して異順方式では前者と異なる方針の配線順序によって並列配線処理を行うものである。この方式では、与えられる配線問題に依存する配線順序と異なる配線順序で処理するので、高品質な配線結果を得るためには引き剥し再配線の反復処理が必要であるが、同順方式と比較して高い配線品質を得ることができる。RP⁹⁾がその代表例である。

3.1 並列処理方法

配線問題の並列処理方法には、ネット内の並列性を利用した配線経路の並列探索と、ネット間の並列性を利用した複数ネットの並列経路探索がある。しかし、前者ではプロセッサの利用率を向上させることが後者に比べて難しいことが問題であり、これを補う方法として配線経路の並列探索に加えて並列経路探索を行う方法が効果的であることが報告されている^{5),6)}。本方式では並列処理方法として、以下の理由により、並列経路探索を採用している。

並列処理による処理時間短縮を図るには、プロセッサ利用率を向上させることが重要である。しかし、配線経路の並列探索では、ネットごとの問題に応じて必要な計算量が常に変化するために、多数のプロセッサを使用する場合は、計算力が必要とする計算量を上回るためにプロセッサの利用率が低下する。

一方、複数ネットを並列経路探索する場合は、配線問題全体で必要とする計算量は配線経路の並列探索方法よりも遥かに大きいため、多数のプロセッサを利用してもプロセッサの利用率が低下する割合は配線経

路の並列探索に比べて小さいと考えられる。

また、本方式のマスタ・スレーブモデルは比較的計算粒度が荒い処理に向いており、ネット単位で並列処理の方が性能向上のためにより効果的であると判断したため、本方式では複数経路の並列経路探索のみを使用しているが、配線経路の並列探索を使用することにより、利用率をさらに改善することが期待できる。

3.2 引き剥し再配線処理

本方式では異順方式による並列配線処理と既配線経路の引き剥し再配線による経路改善処理を用いている。そして以下に示す理由から、引き剥すネットをマスタが評価・選択し、スレーブでネットを再配線する方法を使用する。

(1) **配線順序に依存しない評価** プロセッサ競合方式ではネット割り当て法による配線処理により、各スレーブの処理時間の違いによる並列性の低下を防ぐ。しかしながらその配線順序関係はランダム的になり、順序関係に対する依存性の低い選択方法、および、選択のための評価方法が必要となる。その有効な方法の1つとして、配線の要求を満たすために必要な条件（以後配線規則とする）をコストやペナルティとして表現し、それを用いて配線経路を評価する方法が考えられる。この評価方法により配線結果を配線規則に対するコストの合計値として表現できるため、配線順序に依存しない評価が可能となる。

(2) **並列性の維持** 競合方式ではスレーブ間の配線結果の矛盾が配線品質を低下させる場合があるが、矛盾を経路改善より解消するためにスレーブ間での同期化およびその通信が不要になり並列性を損なわない利点がある。このため、スレーブにおいて高い利用率が期待できる。一方、一般的な方式ではこの種の矛盾を起こさないアルゴリズムを使用している。例えば同順方式の PROTON⁵⁾では処理範囲を概略配線により分割しその範囲内で逐次処理することにより配線経路の矛盾を防ぐ。

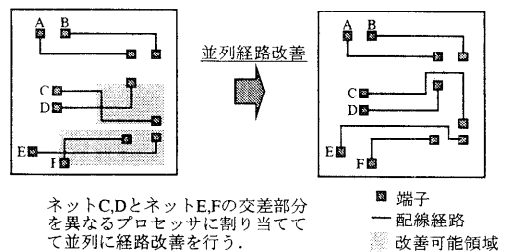


図4 並列経路改善の例

Fig. 4 An example of parallel path improvement.

(3) 経路改善の分散処理 経路改善の反復処理は、反復回数に比例した計算量が必要になるが、ネット割り当て法によりその計算を各スレーブに分散させることができる。また配線に必要な経路探索も並列化できるため、階層的な並列性を利用することができる*

(4) マスタ処理の簡素化 配線結果の評価はスレーブで処理された配線結果とマスタが持つ最新のデータベースの内容を比較することにより決定される。これはスレーブの処理に比べて十分に小さく、簡素化することができる。もし、マスタに負荷が集中する場合でもマスタ処理を幾つかの機能ごとに分割することにより軽くすることが可能である**。

3.3 並列経路改善

2章で述べた競合方式による並列配線では、複数のネットを同時に処理できる利点があり、本方式でもこの特徴を利用して複数経路の並列改善を行う。例えば、図4に示す1層の配線問題例のように複数の経路改善可能な経路（ハッチングされた領域に含まれる部分経路を指す）をネット割り当て法によって並列に処理することができ、これによって経路改善に要する全体の計算時間を問題の並列性に応じて短縮することが可能である。この経路改善の対象となるネットはすべての既配

* 幾つかの並列配線方式で階層的な並列性が用いられており、その有効性が確認されている。本論文で取り上げた PROTON、タイムワープ方式もその例である。

** マスタでは、配線経路の評価・データベースの更新、引き剥がすネットの選択・引き剥がし・割り当て、および、次のネット選択のための後処理の3つの機能から構成され、これらを分割して並列処理することによりマスタの負荷を分散して軽くすることが可能になる。

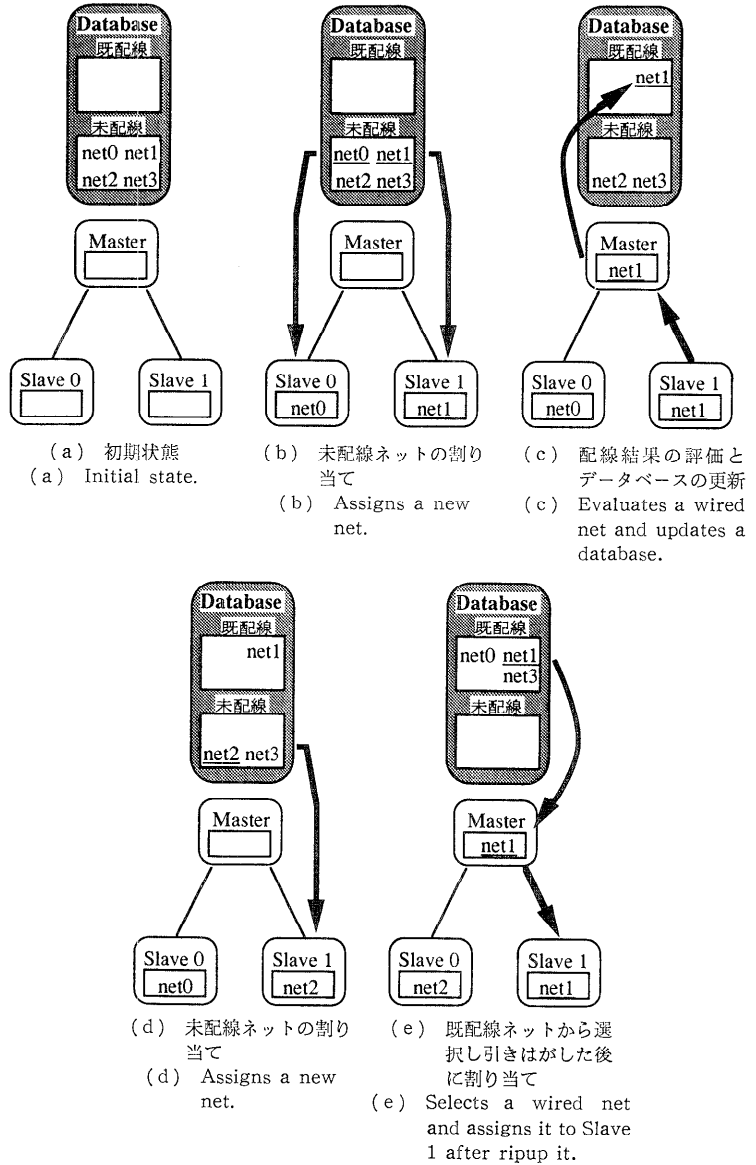


図5 並列経路改善方式のアルゴリズム
Fig. 5 An algorithm of parallel path improvement.

線ネットである。これは、経路間の交差などにより改善を必要とする経路が必ずしも経路改善可能ではなく、また、そのような経路はその周囲の経路を変更した後には経路改善可能になる場合があるためである。そして、品質改善可能なネットは改善されて全体の品質が向上し、同時にその局所的改善によって周囲の配線経路に影響し、配線結果に多様性を与えるため、局所解に陥りにくくまた陥ったときの脱出を容易にする効

果が期待できる。

3.4 アルゴリズム

本稿で提案する並列処理アルゴリズムの概要を図5を用いて説明する。この例では2台の Slave 0,1 で4本のネットを並列配線する。Database 内部では未配線と既配線に区別されており、図中では上段が既配線、下段が未配線を表す。経路探索に必要な経路情報は Database 内部にあり、ネットの割り当て時にスレーブにコピーされる。

(a)の初期状態では net0~net3 のすべてのネットが未配線状態にあり、既配線ネットは存在しない。まず、処理を開始した Slave 0,1 はネットの割り当て要求を Master に出す。Master は先着順に要求を受け、各々に異なる未配線ネットを割り当てる(b)。次に、割り当てられた各 Slave は Master の指示により一斉に配線処理を開始する。各 Slave は配線処理が終了すると配線結果を Master に送り、Master は Slave の配線結果を先着順に評価する。ここで、Slave 1 の配線処理が Slave 0 よりも先に終わった場合を仮定すると、Slave 1 は Master に配線結果を送り、Master はその結果を評価し、配線済みネットとして Database を更新する(c)。そして次に処理するネットを Slave 1 に割り当てる。このとき、未配線ネットを優先し(d)、未配線ネットが存在しない場合は既配線ネットから選択して割り当てる(e)。Slave 0 の場合も Slave 1 と同様である。

以後、Slave 0,1 および Master による引き剥し再配線処理の反復により、並列経路改善が行われる。このようにして、すべてのネットが既配線状態であり他のネットとの交差・衝突などの矛盾が完全に解消されたとき、その配線問題は1つの解を見つけたことになる。

図6に簡単な1層の配線問題による並列処理例を示す。ここでは4本のネット A, B, C, D の並列処理を4台のスレーブで処理することを考える。まず、これらのネットをスレーブに割り当てて並列処理を開始する。最初は Database に登録された配線経路が無いので各スレーブは最短距離で経路を生成する。この結果、(a)に示すようにすべてのネットが網線の部分で接触している。次に、引き剥し再配線による経路改善の反復処理が始まり、各経路は並列に改善される。(b)では各ネットの接触部分が小さくなっている。これはネットDの再配線によりネットCとの間に空き領域が生じ、ネットB, Cはその領域を使用することで

経路改善されているからである。さらに経路改善を行うと、ネットCは同じ経路の繰り返しに対して与えられるペナルティ(後述)により(c)に示す経路を生成する。その後、ネットDが再配線されると、これを迂回する経路を生成し、それより新たな配線領域ができるため、ネットCも再配線により(d)に示す経路が得られ、並列改善処理が終了する。

3.5 経路探索アルゴリズム

本方式での経路探索アルゴリズムに必要な性質は、他の経路と交差・接触のない最適な経路が存在しない場合でもそれらを許した準最適経路が生成できること、配線順序に対する依存性を抑えることである。そこで、経路探索アルゴリズムとして配線コストに基づく迷路法を用いた^{7),10)}。このアルゴリズムでは経路探索に伴う制約をコストで表現し、配線領域に対して配線コストによるラベル付けを行うことで、始点から終点までの配線コストが最小になるような経路を探索することが可能である。今回使用するコストは次に示す5種類である。

- 交差(C)
- 接触(T)
- 折れ曲がり(B)
- ピア(V)
- 配線長(L)

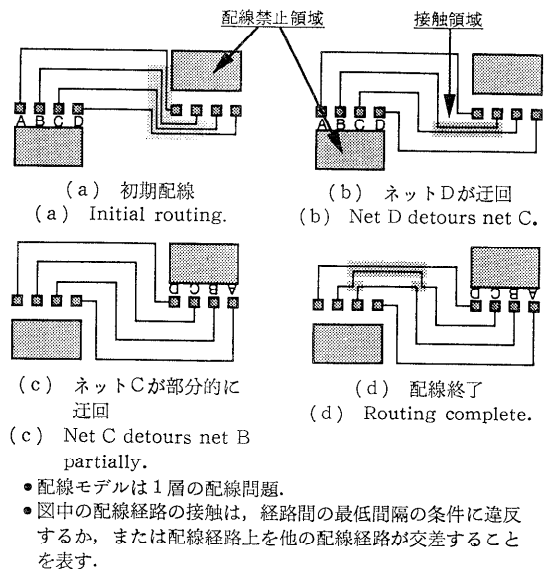


図6 簡単な例による並列配線処理

Fig. 6 Simple example of parallel wire routing.

配線コスト

$$-C^* \text{ 交差数} + T^* \text{ 接触数} + B^* \text{ 折れ曲がり数} \\ + V^* \text{ ピア数} + L^* \text{ 配線長}$$

スレーブで生成された配線経路はマスタによって評価される。この時、評価結果に応じてその経路にペナルティを与えることで、最適でない同じ経路の生成による局所状態を回避する。ペナルティを与えた後の評価結果はデータベースに反映されるため、スレーブは次の経路探索で利用できる。

3.6 経路の評価

マスタが行う配線経路の評価には、引き剥し再配線処理の局所状態の回避、引き剥すネットの効果的な選択の2つの目的がある。前者では、マスタはスレーブの配線経路をデータベースと比較して調べ、競合による配線経路の交差の有無等を検出・評価し、その評価結果によって配線経路と配線領域に対してペナルティを与えることで局所状態の回避と混雑度を緩和する。後者では並列経路改善を効果的に実行するために引き剥す配線経路の選択を行う。

配線経路に対するペナルティ (Pr) は配線経路の交差・接触数の合計数 x に対する関数 $f_p(x)$ と前回配線したときの経路との類似度 S を用いて、 $Pr=f_p(x) * S$ で与える。これは経路探索に使用する配線コストとは異なる。この理由を以下に述べる。局所状態の回避を目的として、本方式では、スレーブにおける経路探索で各ネットの処理回数に応じて、配線コストの係数を変化させる方法を採用している。この方法により各ネットの探索コストは探索ごとに変化するため、繰り返して選択された場合でも異なる探索結果が得られやすい。このため局所状態が回避できると予想される。しかし、ネット全体を評価する場合には公平性に欠けるため、マスタでは異なる評価方法が必要であった。そのため、マスタではペナルティ Pr を用いて配線経路を評価する。

配線領域に与えられるペナルティはその配線領域の

混雑度に応じて設定される。その結果、経路探索においてより混雑度の低い領域が選ばれることになり混雑度の緩和に反映される。

引き剥す配線経路の選択方法には、経路の状況に拘らず順番に選択し引き剥し再配線する方法⁹⁾やランダムに選択する方法、評価結果を基に確率的に選択するなど各種の方法が考えられる。我々は、汎用並列計算機のための並列処理方式を研究する上で基本的な選択方法と思われる、ランダムに選択するランダム選択法とペナルティ Pr を正規化したものを選択確率として確率的に選択する重み付き選択法の2つの方法で評価を行った。

以上の方法と配線コストによる経路探索アルゴリズムにより配線順序に対する依存性を抑える。

3.7 本方式の特徴

本方式の特徴について他の方式と比較しながら述べる。比較対象には PROTON、タイムワープ方式、および RP を取り上げた。

表1に並列処理方法に関する特徴を示す。本方式では並列処理の方法にネット間の並列性を用いた複数経路の並列処理を用いている。他の方式では経路探索の並列処理を行っているが、現在、本方式では使用していない。これは、プロセッサ競合方式において経路探索の並列性よりもネット間の並列性の方が効果的に活用できると判断したからである。また、本方式と同様な経路改善を行う RP では SIMD 型のハードウェアルータにより高速化を行っているが、このハードウェアルータは経路探索の並列化を目的としているため、ネット間の並列性への対応に問題がある。これらのことから本方式では複数経路の並列処理を用いている。

並列性の抽出は、PROTON では詳細配線前に行う静的方式であるのに対し、本方式を含む他の3つの方式では詳細配線時に行う動的方式である。静的方式では並列性やその計算量の概算を得ることができる利点があるが、引き剥し再配線を行う場合には抽出の計算

表1 並列配線処理方式の比較

Table 1 Comparison of parallel wire routing methods.

	並列処理の種類	ネット間の並列性の抽出	配線順序方式	矛盾解消方法
本方式	複数経路	詳細配線時	異順方式	継続, 経路改善
タイムワープ方式	複数経路, 経路探索	詳細配線時	同順方式	ロールバック, 再計算
RP	複数経路, 経路探索	—	異順方式	継続, 経路改善
PROTON	複数経路, 経路探索	詳細配線前	同順方式	—

注) 表中の複数経路, 経路探索はそれぞれ複数経路の並列処理, 経路探索の並列処理を意味する。

をしない必要があるため、このオーバーヘッドを考慮して本方式では動的に抽出する方式を採用している。

並列処理による矛盾の解消方法では、タイムワープ方式では配線情報を矛盾のあった時点まで履歴を遡って矛盾が生じる前の状態に復元（ロールバック）してから再計算するのに対して、本方式では配線情報を継続使用して経路改善を行うため、ロールバックの必要性が無く通信回数を減らすことができる。

引き剥しがしない同順方式では配線順序決定方法が配線品質に強く影響を与えるが、異順方式では同順方式と比較して配線順序の影響は小さい。そして、最適化問題として見た場合の異順方式は同順方式と比較して解空間の探索範囲が広いため、同順方式よりも高い配線品質を期待することができる。そこで同じ異順方式である RP を見ると、RP では逐次反復処理により経路を改善する。その配線順序は問題固有のものとは異なるが、逐次的に処理されており、その順序関係は保証される。一方、我々の方式では配線順序関係は一意的でないが、配線品質の低下を防ぐために配線順序に対する依存性の低い経路探索アルゴリズムと評価方法を用いている。また、プロセッサ間の競合により得られる処理結果の多様性を利用して局所解からの脱出を容易にしている。

以上のように、本方式は汎用並列計算機向けの並列配線、並列品質改善方式として提案するものであり、他の方式と比較した結果、異なる方式であると言える。

4. MIMD 型並列計算機 Coral-68K による実験

本方式の実験には我々の研究室で開発した MIMD 型並列計算機 Coral-68K⁹⁾ を使用した。表 2 に Coral-68K の諸元を示す。この実験で取り扱った配線問題と実験の条件は次のとおりである。

- グリッドサイズ 64×64 の二層配線問題*
- 配線規則には X-Y ルールを適用。
- IC の形に似せて生成した端子群から 2 端子間をランダムに結ぶ 90 本のネットを生成し、それらのネ

* Coral-68K では実装メモリの制約から 256×256 の 2 層配線問題を評価できなかった。そのため、より大規模な配線問題を評価するには別の並列計算機上で行う必要があり、今後の課題の 1 つである。

表 2 Coral-68K の概要
Table 2 Characteristics of Coral-68K.

計算機方式	分散メモリ型	ネットワーク	Binary-Tree
Processor Element		結合 Node 数	63
CPU	MC68000-10 MHz		
Local Memory	512KB	全体性能	
通信方式	DMA 方式 (2MB/s)	最大 40 MIPS, 0.5 MFLOPS	

表 3 ネットデータの諸元
Table 3 Statistics of the net data.

項目	データ 1	データ 2
ネット数	90	121
端子数	180	242
理想総配線長	3123	3290
理想平均長	34.7	27.2
分散	323	264
最長経路長	98	92
最短経路長	6	2

ットから構成される多端子ネットの理想配線長が最短になるように Minimum Spanning Tree を用いて端子間の接続を再構成した 2 端子ネットを使用。

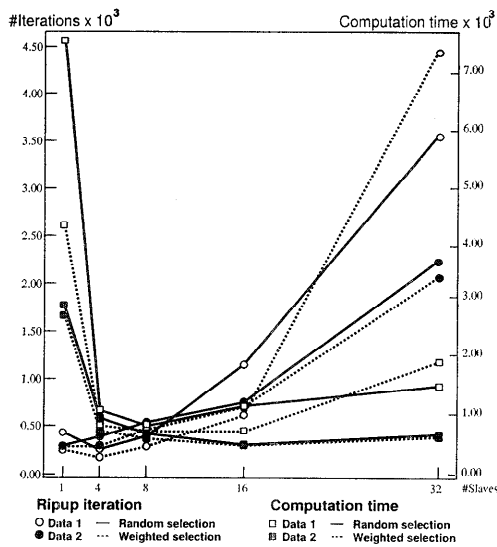
- 引き剥すネットの選択方法として、ランダムに選択する方法（ランダム選択法）と、経路の交差・接触数を重みとして確率的に選択する（重み付き選択法）2つの方法。
- 経路探索の並列化は使用しない。
- 配線終了条件は 100% 配線。

また、使用した 2 種類のネットデータの諸元を表 3 に示す。

実験結果を図 7 に示す。グラフ上の各点は 20 回の実行結果の平均値である。なお、配線品質の項目は配線率、総配線長、総ビア数とするが、100% 配線を前提とするため、総配線長と総ビア数について比較する。各グラフは、スレーブ数を変化させた時の引き剥しがし処理回数と処理時間 (a)、全体の速度向上比とネット 1 本当たりの平均処理時間の速度向上比 (b)、総配線長 (c)、総ビア数 (d)、スレーブ数 8 のときの交差・接触数 (e)* を示す。

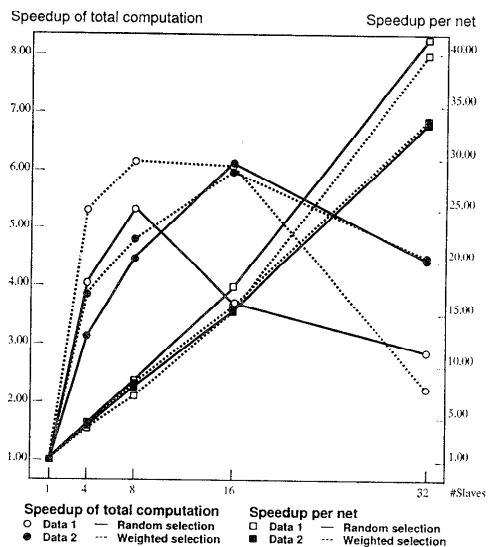
グラフ (b) からデータ 1 はスレーブ数が 8 の時に最も処理時間が短縮されており、この時の速度向上比は

* グラフの縦軸である Iteration は、引き剥し再配線処理回数を表しており、マスタによる引き剥し、スレーブによる再配線、マスタによる配線経路の評価を 1 Iteration としている。



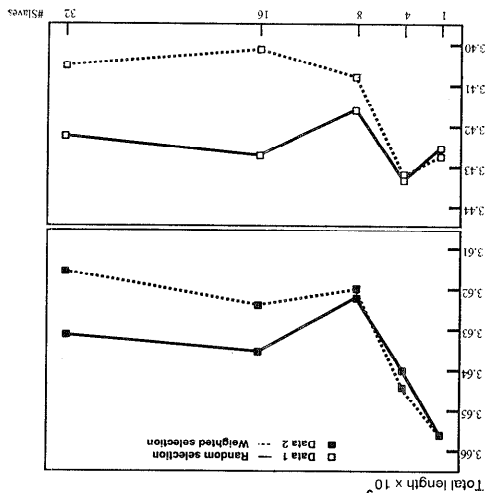
(a) 引き剥し回数および総処理時間とスレーブ数の関係

(a) Number of ripup and total computing time vs. number of slaves.



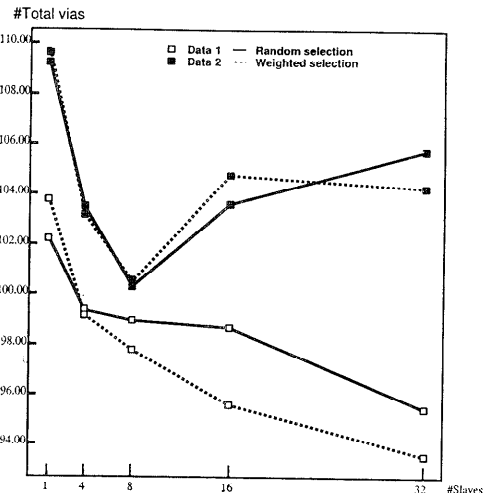
(b) 全体およびネット当りの速度向上比とスレーブ数の関係

(b) Speedup rate of total computation and by a net vs. number of slaves.



(c) 総配線長とスレーブ数の関係

(c) Total wire length vs. number of slaves.

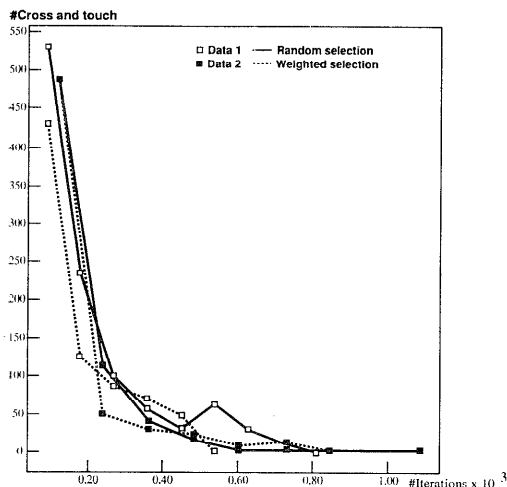


(d) 総ビア数とスレーブ数の関係

(d) Number of vias vs. number of slaves.

6.2 倍、データ 2 ではスレーブ数 16 の時に 6 倍となっている。ネット当りの速度向上比はスレーブ数に比例しており、マスタにおけるボトルネックがほとんど発生していないことがわかる。これはプロセッサ競合方式においてマスタのプロセスがスレーブのプロセスに比べて十分に小さく、スレーブの利用率が高いことを示している。一方、全体の速度向上比はデータ 1 ではスレーブ数 8、データ 2 ではスレーブ数 16 をそれ

ぞれ境にして低下しているが、これはスレーブ数の増加がもたらす競合による矛盾発生数の増加が原因である。この矛盾の増加は、図 7 (a) の引き剥し回数とスレーブ数の関係において、1 台の場合には競合が起きないことを考慮すると、スレーブ台数増加に伴う引き剥し回数の増加は競合によるものであると判断できる。以上の結果から、ネット 1 本当りの速度向上は十分な台数効果が得られているが、並列配線処理全体の



(e) 交差・接触数と繰り返し回数の関係 (8 スレーブ)
 (e) Number of crosses and touches vs. iterations
 (8 slaves).

図 7 Coral-68K による実験結果
 Fig. 7 Results of parallel computation on
 Coral-68K.

速度向上では、ある台数を越えると減少することがわかる。

次に配線品質を見ると、総配線長(c)はデータ 1, 2ともにスレーブ数の増加に対して1%程度しか改善されていない。これは総配線長が総理想配線長の約1.1倍程度と改善の余地が小さいためである。一方、総ビア数(d)はデータ 1, 2双方とも最大で1割改善されている。データ 1ではスレーブ数の増加に従って多く改善されているが、データ 2ではスレーブ数8の時に最も改善されている。

選択方法による違いを見ると、データ 1では重み付き選択方法が総配線長、総ビア数共に効果的である。一方、データ 2では総配線長に関しては効果が見られるが、総ビア数に関しては効果が小さい。選択法の違いによる全体の並列性に関しては、図 7 (b)より、台数の少ない場合から速度向上比が最高になる点までは重み付き選択法、それ以降はランダム選択法が効果的であることが確認された。また、図 7 (d)では並列処理の前半の300回までは早い速度で交差・接触数が減少するが、後半では遅くなっていることがわかる。

5. 考 察

実験結果から、並列経路改善方式の有効性と問題点について考察する。

(1) 配線品質 配線品質には配線率、配線長、ビア数などがあるが実験では100%の配線率を前提としているので、総配線長と総ビア数について考察する。図 7の結果では、総配線長(c)が1%程度改善され、総ビア数(d)は6%~10%改善されている。この理由を考察する。まず、総配線長の改善割合が小さいのは総配線長と理想総配線長の比が小さいため、配線問題の改善の余地が小さいことを示している。これは使用した配線問題が一樣乱数によって作成されており、ネットが配線領域全体に分布していることが考えられる。次に、総ビア数ではデータ 1ではスレーブ数の増加に従ってより改善されている。これは、例えばスレーブ 1台の場合を考えると、その配線処理は逐次的になり探索できる経路の組み合わせは少ないが、台数が増加するにつれて競合による効果も含め探索できる組み合わせが増加し、より良い解の発見に結び付いたものと判断できる。一方、データ 2ではスレーブ数が8の場合には前述の理由が当てはまるが、スレーブ数が多い場合には品質改善効果の低下が観測され、問題によっては、スレーブ数の増加による矛盾割合の増加により、品質改善効果の低下が起きると推測される。そのため使用するスレーブの有効台数は今後の研究課題である。

3.1節で述べたように、本方式では複数ネットの並列改善処理を採用しており、スレーブ 1台の時*に比べて配線品質が低下していないことから、配線経路の並列探索のみの場合と比較しても、品質面で遜色無いたことがわかる。

(2) 並列性 ネット 1本当りの並列性は十分であることから各スレーブの利用率は高く、3.1節で述べたスレーブの並列性維持および経路改善の分散処理の有効性が確認された。一方、スレーブ数の増加による矛盾発生割合の増加は配線結果の収束性に影響を与え、処理回数を増加させる結果となった。また、処理回数の増加による全体の並列性の低下が確認された。そこで、全体の並列性についてスレーブ数を固定した場合と変化させた場合について考察する。

スレーブ数を固定した場合、図 7 (e)で示すように、並列処理前半(約300回まで)では交差・接触数(矛盾数)が多く、選択される確率も高いため経路改善に関するスレーブの利用率は高いが、改善の進行による交差・接触数の減少はスレーブの矛盾解消に寄与

* スレーブが1台の場合、本方式は逐次経路改善になるため、RPで用いられる配線経路の並列探索のみを行った場合の配線品質に相当する。

する実質的な利用率を低下させる。これは配線問題の並列性が配線処理の進行に従って低下することを意味する。この場合、並列性の低下を別の方法で補う必要がある。例えば、実質的な並列性の低下により必要とするスレーブ数が減少するため、結果として余ったスレーブを他の処理に割り当てることで全体としての並列性を維持する方法が考えられる。また、余剰スレーブを用いた、ネットの並列性による経路探索の並列処理などが考えられる。

スレーブ数を増加させた場合は矛盾の発生割合が台数増加に従うため、並列処理終盤において矛盾改善のための反復処理を多く必要とする結果となり、実質的な並列性が低下する。そこで、必要に応じてスレーブ数を変化させるか、スレーブに割り当てる場合の割り当て戦略を改良することにより矛盾発生割合を抑制することが必要である。

(3) ネット選択方法の影響 配線品質では重み付き選択法が、配線長、ピア数共に効果を上げており、ランダム選択法に比べて有効性が高いといえる。一方、並列処理ではスレーブ数が少ない場合に重み付き選択法、多い場合にランダム選択法が良い結果を示している。今回の実験では2つの選択方法しか評価していないが、実際には数多くの方法が考えられるので、配線状況に応じて異なる選択方法を組み合わせる方法が有効であると考えられる。

(4) 最適プロセッサ台数 現在のアルゴリズムで、データ1では、8台が最適なスレーブ台数となっており、総ネット数に対する最適プロセッサ数の割合が全ネット90本に対して9%程度に相当する。データ2ではネット数121本に対して、速度向上の点で16台(13%)、配線品質では8台(7%)の場合が最適な台数となっている。この割合は配線問題の性質によって左右されるが、10,000本規模の配線問題において、仮に5%得られるとすると最大500台のスレーブが使用できることになり、現在利用できる並列計算機で使用できるプロセッサ台数を考慮すると、実用上十分な拡張性が得られることになる。つまり、総ネット数に対する最適プロセッサの割合が低くても、配線問題の規模が大きい場合には十分なプロセッサ台数となる。

(5) 競合方式との比較 2章で述べた競合方式と本方式を比較した場合、競合方式では配線経路の迂回路が無くなると再配線を行っても配線できなくなる。これは図3(b)の配線率の変化からも明らかである。一方本方式では、初期配線が終った段階では配線率が

低い反復処理により改善されるため、前者のように迂回路が存在しない場合でも、改善処理が停止することはない。さらに、生成された経路が周囲の経路に影響を与えるために、混雑度の緩和に貢献し、全体として品質改善につながる。実際、両方式を4章で使用したネットデータ1を使用して比較すると、競合方式では90%の配線率に対して本方式では100%の配線率であった。また、競合方式では配線結果が衝突した時のみ再配線が行われるため、ある時点における再配線率は競合の発生率と同等である。これに対して本方式では、再配線率は使用するスレーブ数に依存している。例えばデータ1において90本のネットで8台のスレーブで処理した場合、 $8/90=8.8$ で約9%の再配線率となる。逆に言えば、スレーブ数を変化させることにより再配線率を変化させ、競合発生割合を変化させることが期待できる。その場合、処理中の最適台数を決定する方法(例えば配線経路の競合の計測など)により、常に問題に適したスレーブ数で処理を行うことが可能になる。

以上のように、改善すべき点が残されてはいるが、本方式のスレーブ数増加による配線品質の向上の特徴は、汎用並列計算機、特に超並列計算機において探索できる組み合わせ数の多さと競合方式の汎用並列計算機に対する依存性の低さを併せて考慮すると、本方式の有効性は高いと言える。

6. おわりに

本研究の目的は汎用並列計算機に対して依存性が低く、効率の良い並列アルゴリズムを開発することである。そして配線問題に対するプロセッサ競合方式の経験から、逐次処理方式(同順方式)より高い配線品質を得るには異順方式による引き剥し再配線処理の反復が効果的であると判断してアプローチを行ってきた。

本論文ではプロセッサ競合方式による並列経路改善方式を提案した。同方式はプロセッサ競合方式の並列処理方式と配線品質向上のための並列経路改善機構により、複数ネットの引き剥しがし再配線の並列処理を実現した。これまでの並列配線方式と異なり、その配線順序が一意的でないため、配線順序に対する依存性を押さえるための経路探索アルゴリズムとネットの評価について述べた。

本方式をプロセッサ競合方式で問題であった配線品質の向上に用いた結果、配線品質の向上とその台数効果を確認した。また、ネット当たりの並列性は十分に

あるが全体の並列性は不十分であることが確認された。これは、経路改善の進行による配線問題の並列性の低下とプロセッサ数の増加による配線経路間の矛盾発生割合の増加が原因であるが、考察の結果、プロセッサ数の動的変化と余剰プロセッサの効果的な割り当て、およびスレーブへのネットの割り当て戦略の改良により、この問題は改善可能であると考えられる。

ランダム選択法と重み付き選択法を比較した結果、ランダム選択法はプロセッサ数が多い場合の並列処理において効果的であり、重みを付加した選択方法は配線品質面で効果的であるが、最終的には複数の選択方法を組み合わせたものが有効であると推測される。また、スレーブにおける経路探索方法にも複数の方法が考えられ、改善の余地が残されている。

今回の評価ではスレーブの処理量がマスタの処理量と比較して十分に大きく、スレーブ数が30台程度で多くなかったため、マスタにおけるボトルネックはほとんど観察されなかった。そのため、計算機方式に対する依存性の確認は不十分であるが、最適なスレーブ台数ではほとんど問題無いと判断する。

今後の課題は、評価・選択方法、スレーブ数の動的変更、並列性の低下を補うための経路探索の並列化の検討、経路探索アルゴリズムの改善、大規模問題による評価などである。

参 考 文 献

- 1) Takahashi, Y., Suzue, T. and Kashimoto, S.: Parallel Maze-Running and Line Search Algorithm for LSI CAD on Binary-Tree Multi-processor, *Proc. World Conf. on Information Processing/Communication*, Seoul, pp. 128-136 (1989).
- 2) Takahashi, Y. and Sasaki, S.: Parallel Automated Wire Routing With a Number of Competing Processors, *Proc. ACM International Conf. on Supercomputing*, pp.310-317 (1990).
- 3) Takahashi, Y. and Sano, M.: Parallel Computing with a Number of Competing Processors, *Parallel Computing '91*, Evans, D. J. et al., eds., pp. 339-346, Elsevier Science Publishers B. V. (North Holland), Amsterdam (1991).
- 4) 佐野, 高橋: 分散メモリ型と共有メモリ型マルチプロセッサによる並列配線処理の性能評価, 情報処理学会論文誌, Vol. 33, No. 3, pp. 369-377

(1992).

- 5) 山内, 中田, 石塚, 西口, 小池: MIMD 型並列計算機上の LSI ルーター—PROTON—, 情報処理学会論文誌, Vol. 34, No. 4, pp. 699-707 (1993).
- 6) 松本, 龍: タイムワープによる並列無格子配線システム, 並列処理シンポジウム '93, pp. 323-329 (1993).
- 7) Kawamura, K. et al.: Touch and Cross Router, *ICCAD-90*, pp. 56-59 (1990).
- 8) 河村, 進藤, 澁谷, 三渡, 大木, 土肥: 超並列配線マシン MAPLE-RP, 並列処理シンポジウム '91, pp. 373-379 (1991).
- 9) 高橋, 遠藤, 松尾, 榎谷: 二進木結合並列計算機 Cora 168 K の開発とその評価, 情報処理学会論文誌, Vol. 30, No. 1, pp. 46-57 (1989).
- 10) Ohtsuki, T.: *Layout Design and Verification*, Chapter 3 "Maze-running and Line-search Algorithms", pp. 99-131, Elsevier Science Publishers B. V. (North Holland) (1986).

(平成6年1月31日受付)

(平成6年11月17日採録)



佐野 雅彦 (学生会員)

昭和42年生。平成2年徳島大学工学部情報工学科卒業。平成4年同大学大学院工学研究科修士課程修了。現在同大学院博士課程在学中。配線問題の並列処理方式の研究に従事。修士(工学)。



高橋 義造 (正会員)

昭和7年生。神戸市出身。昭和30年京都大学工学部電気工学科卒業。同年(株)東芝に入社し、アナログ計算機、計算機制御、基本ソフト、オンラインシステム、並列分散処理システム等の研究開発に従事した。昭和52年に同社を退社し、徳島大学工学部教授に就任し、現在まで計算機ハード/ソフトの教育と、並列処理の研究を行っている。著書に「制御工学」、「電子計算機演習」(朝倉書店)、「計算機方式」(コロナ社)等、編著に「並列計算機構」(丸善)等がある。工学博士。電子情報通信学会、ソフトウェア科学会、ACM、IEEE Computer Society 各会員。