超並列配線マシンのアルゴリズム

4P−2

三渡秀樹* 河村薫 進藤達也 澁谷利行 大木由江
株式会社富士通研究所

1．はじめに

我々は高い配線率を得ることが可能な制約緩和迷路法を開発した．

しかし，この手法には配線を終了するまでに要する時間が長いという問題が存在する．我々は，並列処理を用いてこの問題に対処している．本論文では制約緩和迷路法について説明し，経路探索の主要な処理であるラベリングのアルゴリズムについて説明する．

2．制約緩和迷路法

制約緩和迷路法では，経路探索時に既配線パターンに対する交差を認めている．全ネットを繰り返し，再配線する毎に交差に対して制限を加えていき，交差数を減少させていく．

配線パターン間の相対関係を考慮するために交差は図1に示すようなクロスとタッチの2種類に区別されている．2ネット以上の配線パターンが互いに他を横切る場合をクロス，平行に重なる場合をタッチと呼ぶ．

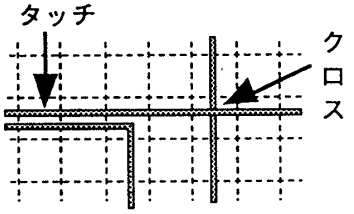


図1　クロスとタッチ

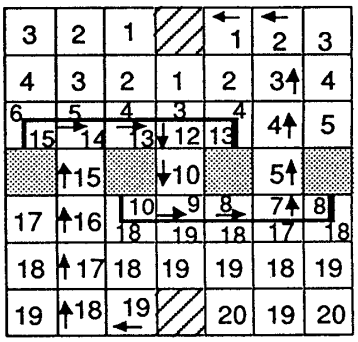
従来手法における配線長，ビア数に対するコストに加え，クロス数，タッチ数に対してもコストを設け，2点間の経路探索に用いるコスト関数を以下のように定義する．

$$\text{コスト} = a\times\text{配線長} + b\times\text{ビア数} + c\times\text{クロス数} + d\times\text{タッチ数}\quad (a,\ b\geqq 1,\ c,\ d\geqq 0)$$

a，b，c，dはコスト係数である．

経路探索には迷路法を用いる．各セルに付加されるコスト値はマンハッタン距離が1進んだ時にaが加算される．この時にビアを発生すればbが加算され，クロスした時，タッチした時にはそれぞれc，dが加算される．

図2にラベリングの様子を示す．クロスとタッチを区別するために既配線パターンの存在するセルは既配線パターンにより2つ以上に分割される．図中斜線で示したところはそれぞれソースとターゲットである．



a=1, c=1, d=10の時

図2　経路探索終了結果

最短経路が最小コストの経路とは限らないため，ラベル値は書き換えられる可能性がある．したがって，ターゲットにラベルがついた後も，全てのウェーブフロント上のセルのコスト値がターゲットのコスト値を越えるまで，経路探索を続ける必要がある．また，複数セルにターゲットが存在する場合には，ターゲットについているラベルの中で最小のコスト値をウェーブフロント上のセルのコスト値と比較すればよい．図中の矢印はバックトレースの様子を示している．ラベルの書き換えに対処するため，各セルでは最小のコスト値を与えた隣接セルへの方向を記録する必要がある．

コスト係数c，dは交差に関する制限を与える．これらの値を増加させて再配線すると，これらのコス

Routing Algorithm of the Massively Parallel Routing Machine
Hideki Miwatari, Kaoru Kawamura, Tatsuya Shindo, Toshiyuki Shibuya and Yoshie Ohki
FUJITSU LABORATORIES LTD., KAWASAKI

ト係数によるコストの増分をコスト係数a，bによるコストで吸収できるネットは配線長やビアを増やすことにより，交差数を減少させる．プロトタイプによる実験で，これらc，dの値をそれぞれ0から単調増加させながら再配線を繰り返した結果，高い配線率が得られることが確認された．

3．ラベリング

ラベリングはコスト関数値を最小にする経路を求めるための手段である．ウェーブフロント上の1セルにおけるラベリングの処理の流れは図3に示す通りである．このラベリングはコスト値の書き換えがなくなった時，すなわち，ウェーブフロントがなくなった時終了する．

ターゲットの最小コスト値を利用してラベリングの終了を早めている．コスト最小経路上のセルのコスト値はターゲット上のコスト値よりも大きくはなりえないという性質を利用すると，ターゲットのコスト値を越える値をセルに付けることを省くことができるので，不要なウェーブフロントの発生を回避出来る．

ここで，costはセルに付けられているコスト値，

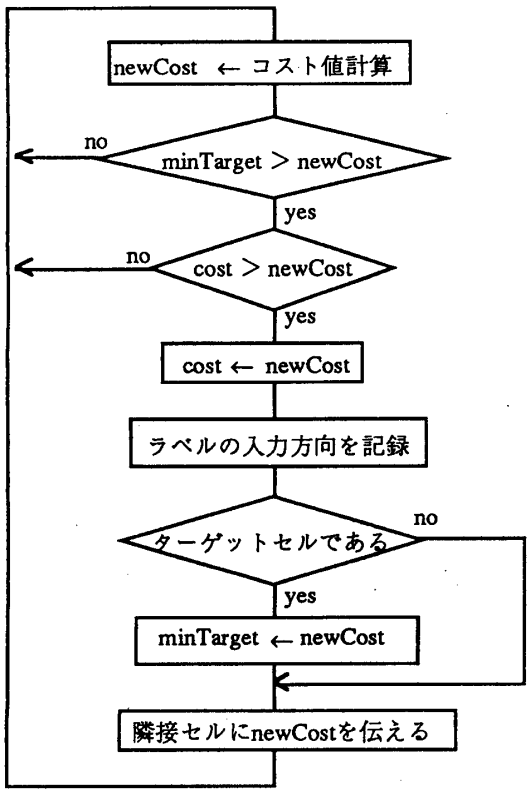


図3　ラベリング処理の流れ

newCostはコスト関数に従って，計算されたコストである．minTargetはターゲットに付いているコスト値の最小値であり，全セルに共通な値であり，全セルから参照する．

まず最初にウェーブフロントに隣接しているセルはウェーブフロントからラベルを伝えられる．そしてこれらのラベル値をコスト関数に代入してnewCostを求める．このnewCostはminTargetと比較される．もしminTargetより大きな値であれば，このセルはコスト最小経路上のセルにはなりえないので何も実行しない．このnewCostとすでに付いているcostを比較し，これより小さな値であれば，このセルはウェーブフロントとなり，コスト最小経路上のセルになる可能性があるのでnewCostを新しいコストとする．また，バックトレースのためにラベルの入力方向を記録しておく．newCostを新しいコストとした後でこのセルがターゲットであれば，minTargetにnewCostを代入する．その後でこのセルの隣接セルにnewCostを伝える．以降，ウェーブフロントがなくなるまでウェーブフロント上のセルにおいて以上の処理を繰り返す．

4．並列処理

迷路法ではウェーブフロントの存在するセルを並列に処理できることが知られている．制約緩和迷路法においても同様にウェーブフロント上のセルを並列に処理することが可能である．複数のプロセッサにウェーブフロント上のセルを割り当て，それぞれのプロセッサにおいて上記のアルゴリズムを実行する．

制約緩和迷路法による配線は通常の迷路法による配線よりも配線終了までに要する時間が長くなる．これは，セルのコスト値が書き換えられる場合が存在することと，再配線の度に経路探索を繰り返すことに起因している．制約緩和迷路法ではラベリング処理の占める割合が大きく，実用規模のデータでは全体の9割を占める．我々は，並列処理を用いて制約緩和迷路法のラベリングを高速に実行することで，この問題に対処している．

5．まとめ

制約緩和迷路法について述べた．制約緩和迷路法では経路探索処理が全体の大部分の時間を占めるため実用化のためには経路探索アルゴリズムを高速に実行可能な並列処理装置が必要である．