

ベクトル計算機を用いた迷路法の高速化

6J-9

三木良雄 鈴木 敬 高嶺美夫

日立製作所

1. はじめに

プリント基板、LSIの自動配線アルゴリズムとして迷路法[1]が広く利用されている。迷路法は配線可能領域をしらみつぶ的に探索する手法であり、

- ①経路が存在すれば必ず発見できる。
- ②最短経路が発見できる。
- ③配線問題としての制約条件を受けにくい。

などの特徴を備えている一方、

- ①処理時間がかかる。
- ②探索用のメモリを多く要する。

などの欠点を有しており、この克服が30余年にわたる配線アルゴリズムの研究を支えてきたと言っても過言ではない。

本稿では、迷路法の処理時間高速化技術としてベクトル計算機の利用を提案し、そのアルゴリズムと実験結果について報告する。

2. 迷路法アルゴリズム

2.1 基本アルゴリズム

迷路法はその基本アルゴリズムが提案されて以来多くの改善手法が発表されて[1,2]いる。本節では以降に述べるベクトル化迷路法との関連を明らかにするため、本研究での迷路法の基本アルゴリズムを定義する。図1は配線経路探索の例を示すもので、図中斜線部分は配線禁止領域を、点S、Tはそれぞれ経路を求める際の探索開始点および探索目標点を表す。

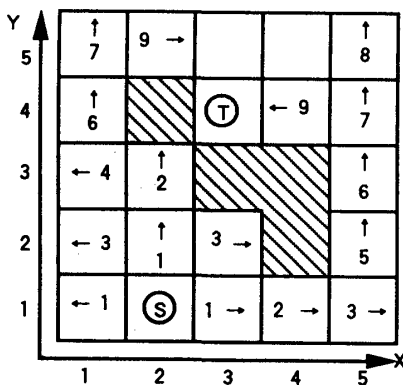


図1 配線経路探索の例

「基本アルゴリズム」

- (1)探索開始点をウェーブフロント(WF)に登録。
- (2)WF要素数=0または経路発見までループ。
 - (2-1)WFから格子点の一つ取り出す(着目点)。
 - (2-2)着目点から探索できる格子点(探索点)を計算。
 - (2-3)探索点が探索済みでも配線禁止でもない場合。
 - (2-3-1)探索点の探索コストを計算する。
 - (2-3-2)探索点が目標点の場合、経路発見とする。
 - (2-3-3)探索点を探索済みとし、WFに登録する。

上記のアルゴリズムはコスト付き迷路法とよばれるものであり、図1の例に於いて格子中の数字と矢印はそれぞれ探索コストと探索方向を表す。

2.2 高速化手法の検討

迷路法の高速化のためには並列実行可能性を検討する必要がある。

(1) ネット並列性

多くの配線問題ではネットを並列処理の単位として考え、異電位の配線パターンを独立に求めても問題ない[3]。しかし、ネット並列性に着目した場合、配線経路の探索中に異電位ネットのショートを随時解消する処理が必要となるため、ベクトル計算機やSIMD(Single Instruction Multiple Data)タイプの並列計算機のように同一の処理をまとめて実行する計算機には適さない。

(2) データ並列性

迷路法では各格子点に着目した場合「次の探索点の座標計算」「探索コストの計算」はほぼ独立に実行可能である。全格子点に独立した処理機能(プロセッサ)を割り当てた場合に最大の並列実行性を引出すことができるが、例えば図1の同一コストを持つ格子点(1,2)(3,2)(5,1)のように、同時に並列処理可能である格子点は限定されるため、プロセッサの稼働効率は低い。

3. ベクトル化迷路法

ベクトル化迷路法では並列処理可能な格子点をベクトルデータとして考え、「探索座標計算」「探索コスト計算」の処理をベクトルレジスタと主記憶との間の関係演算として実現する。

3.1 探索座標計算

次に探索する座標は着目している格子がどこから探索されてきたかによって決定する。例えば配線可能な領域が平面であり、下方から探索が進んできた場合には次の探索点は上、右、左の3方向となる。これは配線可能領域が3次元の場合にも同様となり、次の探索座標を計算するためのオフセット値は着目している格子の探索方向によって決定する。

探索座標計算はベクトル計算機の間接ロード機能

- を用いて以下の手順で実現可能である。(図2)
- (0)探索方向と次の探索座標のオフセット値の関係を主記憶上にセットしておく。
 - (1)同一コストを持つ格子点について、その探索方向をベクトルレジスタ1(VR1)に、格子点座標値をVR2にロードする。
 - (2)VR1をアドレス値としてVR3~VR6に主記憶上のデータを読み込む
 - (3)VR2の内容とVR3~VR6とを加算して次の探索座標を得る

図中、平面の探索において上、下、左、右の探索方向をそれぞれ1、2、3、4と符号化した。

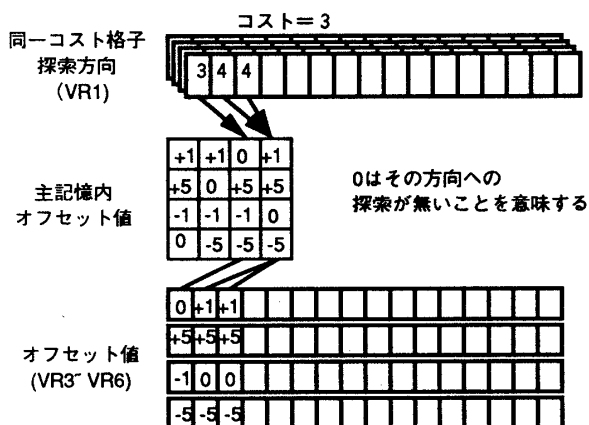


図2 ベクトル計算機上での探索座標計算

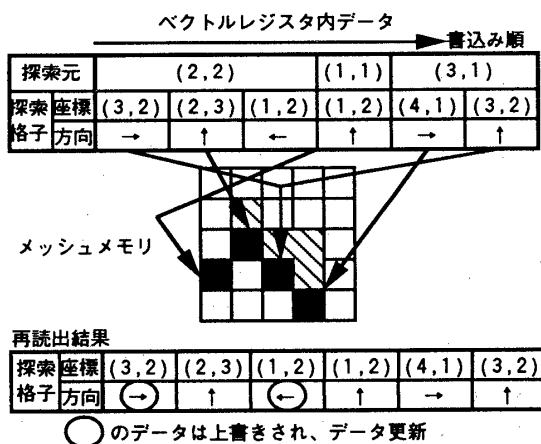


図3 重複探索除去処理

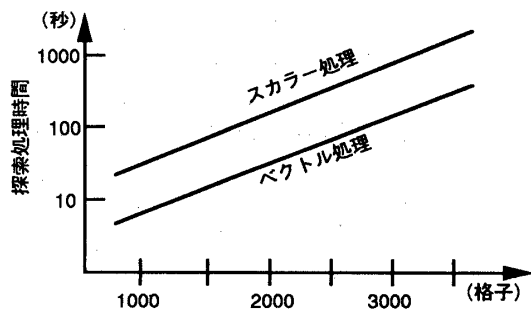


図4 経路長と処理時間

3.2 探索コスト計算

探索コストは経路の形状をコントロールしたり探索範囲を限定するために導入するものである。図1に示した探索の例では直線状の配線形状を得易くするために、探索方向が変わる場合に+2、探索方向が変わらない場合に+1のコストを加算している。このように探索コストの計算では、先の探索座標計算と同様に元の探索方向と新しく求められた探索方向の組によって加算すべきコストを計算することが可能である。

3.3 重複探索除去処理

ベクトル計算機では同一ベクトルレジスタ内のデータ同士で競合関係になるような処理を行うことはできない。従って、図1の例における点(3, 2)のように、同じ格子点に対して異なる方向からの探索が重複する場合に上記の競合関係が発生する。その解消処理は以下のとおりである。(図3)

- (1)ベクトルレジスタ内の新しい探索格子の探索方向をメッシュテーブルに書き込む。
- (2)再び、ベクトルレジスタ内の探索方向と、メッシュテーブルに記録されている探索方向を比較する。
- (3)探索方向が異なる場合に該当するベクトルレジスタ内のデータを無効にする。

4. 実験結果

2章で述べた迷路法、並びに3章で述べたベクトル化迷路法をFORTRANでコーディングし、スカラー計算機、ベクトル計算機上で経路探索時間を測定した。図3は結線すべき端子間の距離と探索処理時間の関係を表したものである。処理時間はマシンサイクルで正規化されている。実験では約5倍の高速性が確認できたが、アセンブラコーディング等でさらに高速化が可能である。またアルゴリズムの点では、コスト別にウエーブフロントを管理する処理が、いわゆるコスト値をキーとした分類処理となりベクトル計算機との整合性が悪い。この点については今後の課題とし、継続的に報告したい。

4. 結言

ベクトル計算機の高速度性を利用した配線アルゴリズム、ベクトル化迷路法を提案し、実用的にはほぼ5倍の高速化を確認した。

[参考文献]

- 1)C.Y.Lee:An Algorithm for Path Connections and Its Applications,IRE Trans.on EC pp.346-364(1961)
- 2)J.H.Hoel:Some Variations of Lee's Algorithm,IEEE Trans.on Compt.C-25 No.1 pp.19-24(1976)
- 3)進藤、川戸：プロセッサの協調動作に基づく並列配線方法、信学技報CAS-85-152 pp.33-40