計算機アーキテクチャ　103－1
設　計　自　動　化　69－1
（1993. 12. 16）

Hill-Climbingを用いたパーティショニング最適化手法

澁谷利行 河村薫
(株)富士通研究所
川崎市中原区上小田中1015

あらまし

本稿では，パーティショニングにおけるmin-cutの最適化手法としてStable-Net-Transition法(SNT)について述べる．min-cutを実行したとき，カットされたままの状態のネットをstableネットと定義する．SNTでは，このstableネットに着目してhill-climbingを行なうことにより，実用的な時間内でカットサイズの最適化を行なうことを特徴としている．
SOGゲートアレイの実レイアウトデータを用いて実験を行ない，カットサイズ，収束性，配線長，配線率を評価し，その有効性を示した．

和文キーワード　配置，パーティショニング，VLSI，レイアウト

An Efficient Hill-Climbing Algorithm For Partitioning

Toshiyuki SHIBUYA, Kaoru KAWAMURA
FUJITSU LABORATORIES LTD.
1015, Kamikodanaka Nakahara-Ku, Kawasaki 211, Japan

Abstract

In this paper, we present a **stable-net-transition method**(SNT) for min-cut partitioning. SNT is a hill-climbing method for optimizing the cut set size of a network in practical amount of time. Good experimental results have been observed for cut set size , wire length and routing ratio.

英文 key words　placement, partitioning, VLSI, layout

1 はじめに

最近の製造技術の進歩により，VLSIの大規模化，高集積度化には著しいものがある．このような大規模な回路データは，数万以上のセルと，そのセル間を電気的につなぐネットワークで構成されており，それらは複雑に絡み合っている．このため，配置問題において，分割統治法のように，問題を階層的に小さな問題に分割して行き，問題の複雑さと規模を小さくする方法が，大規模，かつ複雑な問題を高速に処理するために重要であると考えられる．

分割統治法の一つであるパーティショニングは，与えられた回路をK個の部分回路に分割して，各部分回路間をまたがるネットの本数，すなわち，カットサイズを最小化することを目的とした手法である．パーティショニングは，問題の計算複雑度がNP-completeに属すると言われており[5]，何らかのヒューリスティクなアルゴリズムを用いて，カットサイズの最適化を行なう必要がある．パーティショニングの代表的なアルゴリズムであるmin-cut[11][3][2]は，すでに配置の一手法として広く利用されている．

その中でも，FiduccaとMattheyses[3]によって提案された手法は，セルにつながったピンの総和Pに対して，線形の計算オーダ$O(P)$で処理されることを特徴とした高速な手法である．以下，この手法をFMと呼ぶことにする．FMは，収束性が高い反面，最適化の際に簡単なhill-climbingしか行なわないので，すぐにlocal minimumに陥ってしまう．そのため，最終解が初期解に大きく依存してしまうという欠点がある．

FMのアルゴリズムをそのまま用いて，FMのこの欠点を補う方法としては，異なる複数の初期解をFMに与えて実行するか[6]，constructiveにクラスタ化して分割した結果を初期解として，FMに与える[10]方法が提案されている．

Krishnamurthy[9]は，先読みをしたcell-gainを一般化させてlevel gainと名付け，FMを改良した手法を提案した．先読みを行なうことにより，FMよりもカットサイズの改善が行なわれたが，結果が初期解に依存することには変わらない．また，計算オーダは，$O(P)$のままだが，levelの深さをlとしたときに，メモリのオーダが$O(P^l)$となり，大規模な回路データを扱うことは，現実的ではない．

また，最近では，clusteringによってネットリストのサイズを小さくして，回路の大規模化に対応ながら質の高い解を得ようという研究も活発であるが[1][4]，global minimumを得るための根本的な解決方法にはなっていない．

本論文で提案するStable-Net-Transition(SNT)法は，stable状態にあるネットを検出し，そのネットにつながっているセルを移動させることにより，大域的にhill-climbingを行なう手法である．以下，SNT法をFMと組み合わせたアルゴリズムをSNT-FMと呼ぶ．SNT-FMは，FMに比べて初期解に依存しにくく，大規模な問題に対しても，実用的な時間で，質の良いパーティショニングの結果が得られることを特徴としている．また，本手法では，一つのカットラインに対して，解の質が異なる準最適解が複数個求まることも特徴の一つである．

以下，第2章では，SNT法の考え方を述べる．第3章では，SNT-FMのアルゴリズムについて説明する．第4章では，SNT-FMの性能評価を行なう．

2 Stable-Net-Transition法の考え方

図1は，一本のネットにつながっているセルのcell-gain(g)が，カットラインにより分割されている状態によって，どのような値を取るかを示したものである．

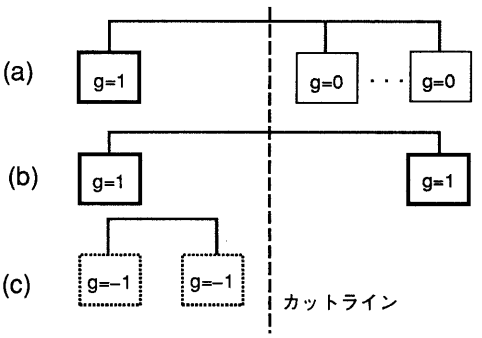

図 1: cell-gainの定義

図1(a)(b)の太線で囲まれているセルは，cell-gainが1であり，相手のブロックに移動すると，そのネットがカットされない状態になる．また，点線で囲まれているセルは，cell-gainが-1であり，現在このネットはカットされていないが，相手のブロックへ移動すると，ネットがカットされてしまうことを意味する．

FMのアルゴリズムは，このcell-gainに従って，カットサイズを減らすように動作をする．ところが，FMには，新たに他のセルのcell-gainが増えるような指標を持っていない．そのため，一本のネットに，たくさんのセルがつながっているネットが初期解として図2のような状態で置かれていると，このネットは始めから最後までカットされたままの状態である可能性が確率的に高くなると考えられる．つまり，FMはセル単位でカットサイズを小さくすることは考えてるが，ネット単位で図1(a)(b)の様な状態のネットをつくり出す努力をしていないのである．.

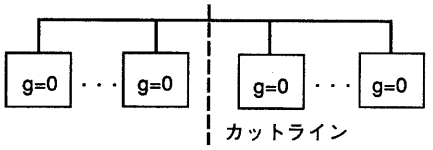

図 2: 両ブロックに複数セルが存在する状態

このことが，FMで得られる解が，初期解に大きく依存する原因の一つだと考えられる．実際，実レイアウトデータの中には，たくさんのセルがつながっているネットが数多く存在している．従って，そのようなネットは，初期解として両方のブロックに複数のセルが存在する状態となる可能性が高くなってしまい，FMでは十分な結果が得られないと考えられる．

一度FMを行なった後，ランダムにセルを動かしてlocal minimumから脱出する方法による実験も[6]で行なわれている．しかし，動かすセルの比率を高めると，ランダム解から始めるのと変わらなくなり，また，比率が低いとlocal minimumを抜け出すことはない．

また，単純にセルをランダム化させたのでは，たくさんのセルがつながっているネットの場合，図2のような状態に陥り易くなってしまう．従って，この方法を実レイアウトデータに適用しても，効率的にlocal minimumを抜け出すことは難しい．

以降，図2ように，FMを適用してもカットされたまま変化の無い状態のネットを「stable状態」であると言い，そのネットを「stableネット」と呼ぶことにする．

そして，我々は，実レイアウトデータにおいて，stableネットがどのくらい存在するかを，次のような実験で調査した．

表1は，SOGゲートアレイの実レイアウトデータである．チップのゲート規模は，10K～100Kゲートである．実験では，表1のデータに対して，10種類のランダムな初期解を作り，FMを適用させた．そして，最終的にカットされているネットの中で，どのくらいのネットがstableネットであるかを調べた．

表2には，初期カットサイズ(CS)と最終カットサイズ，Stableネットの数(#SN)，最終カットサイズにおけるStableネットの数の比率(SN率)の平均値を示した．

この実験から分かったことは，最終カットとして残るネットの80%以上が初期解の状態で，すでにカットされているということである．

そこで，このstableネットが，今実行したFMの解をlocal minimumに陥らせるとともに，解の質を決定した要因だと考え，できるだけstableネットがカットされていない初期解をFMに与えることが，カットサイズの最適化に効果的であると考えた．SNT法では，できるだけstableネットがカットされていない初期解を生成するために，FMを実行後に，stableネットを検出して，stableネットにつながっているセルを，強制的に片方のブロックに移動させる．そして，FMとSNT法を繰り返し実行することにより，カットサイズの最適化を行なって行く．また，SNT法では，最終的にカットされているネットの種類が，FMを繰り返すごとに大きく変わる可能性があるため，カットされているネットの種類が全く異なる準最適解が複数求まる可能性がある．

表 1: データのサイズ

データ	セル数	ネット数	外部端子数	ピン数
AA71	962	1602	71	4674
BB10	1682	2234	133	6570
CC30	3389	4908	218	16080
DD39	7870	11812	150	36051
EE00	9021	12270	329	39427
FF12	14357	17440	195	53838

表 2: FMの結果

データ	初期CS	最終CS	#SN	SN率(%)
AA71	952.8	232.2	201.6	86.8
BB10	1309.5	424.3	352.8	83.1
CC30	2872.8	743.4	642.2	86.3
DD39	6930.3	1801.8	1484.7	82.4
EE00	7393.8	1886.2	1630.3	86.4
FF12	10550.0	2635.2	2260.5	85.7

3 SNT-FMアルゴリズム

3.1 アルゴリズム全体の流れ

あるカットラインに対して，SNT-FMによりパーティショニングする時のアルゴリズム全体の流れを図3に示す．

アルゴリズム全体を制御するパラメータとして，繰り返し係数Rがある．Rは，処理時間とカットサイズの質を決める重要な値である．つまり，Rの値を小さくすれば，処理時間は短くなるが，あまりカットサイズの最適化が行なわれない．また，Rを大きくすれば，カットサイズの最適化は進むが，処理時間が長くなる．経験的には，分割される領域に含まれるセルの個数に比例した値をRに用いると，性能・時間ともに効率的であることが分っている．本論文で行なわれた評価は，分割されるべきブロックに含まれるセル数をNとしたときに，式1で求めた繰り返し係数Rを用いている．階層的に回路分割

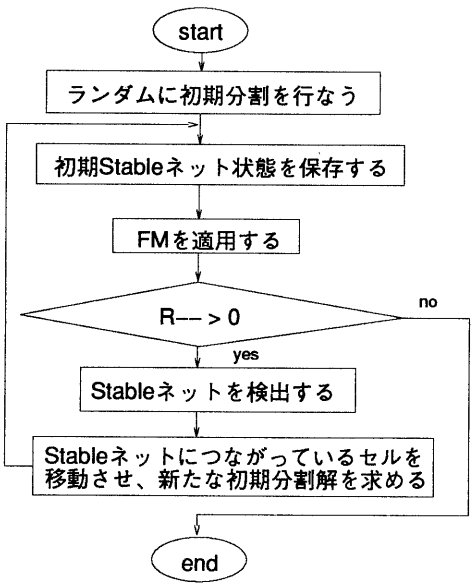

図 3: SNT-FM全体のフロー

が進むにつれて，分割されるべきブロックに含まれるセル数は少なくなるので，Rの値は小さくなる．また，繰り返し係数Rについての実験的な考察は，第4.2章で行なう．

$$R = 0.011 \times N + 15 \qquad (1)$$

SNT-FMに適用したFMは，基本的には[3]と同じものだが，繰り返し係数Rによる繰り返しを通して，常にカットサイズがベストな状態を保存しておく処理が加えられている．また，ブロック内のセルサイズのバランスを保つために，セルサイズの総和の大きいブロックの方から小さい方へセルを移動させることにする．

stableネットを検出する処理は第3.1.1章で，新たな初期分割を求める処理は第3.1.2章で説明する．

3.1.1 Stableネットの検出方法

stableネットを検出するためには，ネットがカットされているかどうかの状態を，FMを適用する前後で比較する必要がある．

そのために，ネットがstable状態にあるかどうかを保存するためのフラグ*SNFlag*を,各ネットごとに用意する．まず，FMを適用する前にネットの状態を調べ，カットされているなら*SNFlag*に*TRUE*を，そうでないなら*FALSE*をセットする．次に，FMを適用させた後，また，ネットの状態を調べて*SNFlag*が*TRUE*であるネットが，最終的にカットされていない場合には，*SNFlag*を*FALSE*に戻す．

こうして，最終的に*SNFlag*が*TRUE*であるネットが，Stableネットとして検出される．

例として，図4の様な初期解に対してFMを適用し，stableネットの検出を行なう．

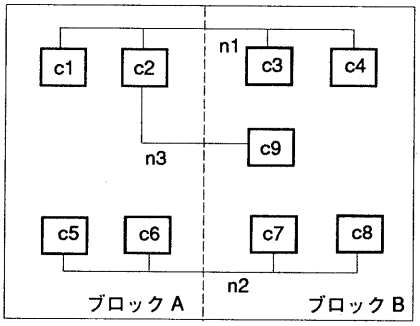

図 4: 初期配置

分割される領域は，A, Bのブロックに分れる．セルは，9つ(C1〜9)あり，皆サイズは等しく，固定セルは無いとする．ネットは，3つ(n1〜3)で，全てのネットがカットされているので，FM前の各ネットの*SNFlag*は皆*TRUE*である(表3)．

この初期解に対し，FMを適用させた結果，c9のみが移動して図5のようになったとする．すると，n3はカットされなくなったので，*FALSE*になり，n1, n2がstableネットと認識される(表3)．

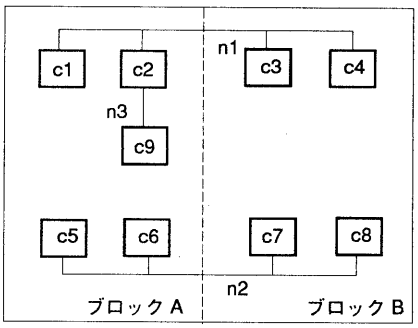

図 5: FM後の配置

3.1.2 SNTアルゴリズム

stableネットにつながっているセルを移動させるときに，注意すべきことは次の３つである．

1. ブロックの一方向だけに偏ってセルを移動させるとブロック内のセルサイズの総和のバラ

表 3: SNFlag

ネット	FM前のSNFlag	FM後のSNFlag
n1	TRUE	TRUE
n2	TRUE	TRUE
n3	TRUE	FALSE

ンスが，大きく崩れてしまう．

2. 固定セルにつながっているネットの場合，動ける方向は限定される．

3. 一度SNTにより移動したセルは，固定セルと見なされ，他のstableネットにより再度移動させられることはない．

stableネットを一本一本順々に移動させる時に，3番目の注意事項より，stableネットが何本か移動すると，固定されるセルが増え，移動できなくなるstableネットが増えてきてしまう．つまり，初期に選択されるstableネットほどstable状態を抜け出す可能性が高くなる．そこで，どのstableネットを初期の移動対象として選択するかが重要になる．

以下で述べるアルゴリズムでは，移動対象となるstableネットを乱数で選択している．このことにより，繰り返し係数Rを大きくすればするほど，より最適解が求まるようになっている．

まず，アルゴリズムで使用する変数について説明する．

SNList: SNTにおいて，まだ処理されていないstableネットを管理するためのリストである．

M: 一回のSNTによって移動可能な最大stableネットの本数である．stableネットとして選ばれたネットを可能な限り全て移動させてしまうと，初期値としてのカットサイズが大きくなり過ぎてしまうので，次に実行するFMの収束が悪くなってしまう．そこで，あらかじめ最大移動可能stableネット率αを定数として設定しておき，stableネットの本数にαを掛けた値をMの値として代入しておく．αの値は，速度・性能の面から0.4〜0.6くらいが効果的である．第4章での実験結果は，全てαを0.5に固定して実験を行なった．

次に，SNTのアルゴリズムを図6に示す．このアルゴリズムで移動の対象となるセルは，分割される領域内に含まれるセルのみである．

終了条件としては，次の3つがある．

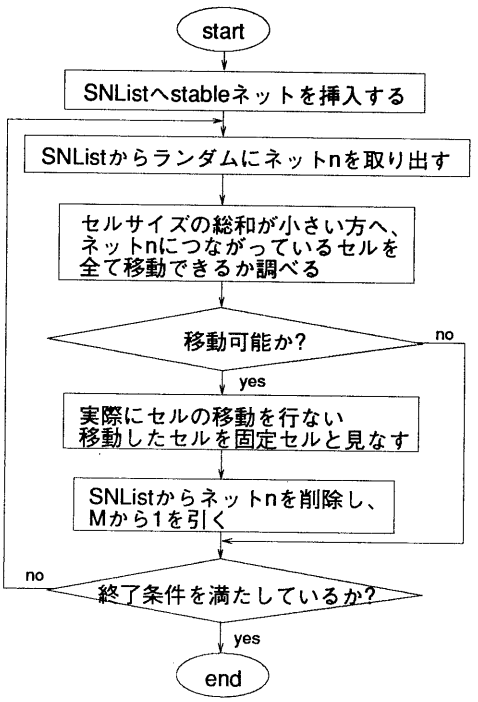

図 6: SNT-FM全体のフロー

1. *SNList*が空である．処理すべきstableネットが無い．

2. Mが0である．移動可能な最大stableネットの本数分だけstableネットが移動した．

3. 現在SNListに入っているstableネットの本数回，stableネットの移動に失敗した．これ以上stableネットを移動させようとしても，移動できる可能性が低い．

図5の結果を例にSNTアルゴリズムの動作を説明する．このデータでは，ネットの本数が少ないので，Mによる終了条件は考えないことにする．

まず，stableネットとして認識された2のネットn1, n2を*SNList*に挿入する．

$SNList = \{n1, n2\}$

ランダムに*SNList*からネットを取り出した結果，n1が選ばれたとする．セルサイズの総和は，ブロックBの方が小さいので，n1につながっている全てのセルは，ブロックBの方に移動可能なので移動を行なう．実際に移動したセル(c1, c2)は，固定セルと見なされてSNTが終了するまで移動できない．n1は，*SNList*からを取り除かれる．

次に，*SNList*からn2を取り出し，セルサイズの総和の小さいブロックAへ移動させる．*SNList*から

n2を取り除くと$SNList = \phi$となるので，SNTを終了する．その結果を，図7に示す．この結果を新たな初期解としてFMを適用すると，c9がブロックBに移動して最適解が求まる．

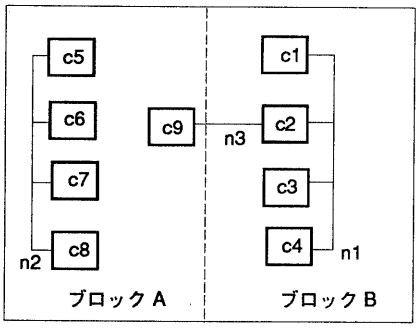

図 7: SNT後の配置

以上が，SNTの基本的なアルゴリズムであるが，実験的に次のような処理をSNTに加えると，より最適な解が求まることが分っている．まず，前回，stableネットとして認識されたが，移動できなかったネットを記憶しておく．そして，今回もstableネットとして認識されているなら，乱数を発生する際に重みをつけて優先して移動させるという処理である．次章の実験結果で用いるSNT-FMでは，この処理を組み込んでいる．

4 実験結果

我々は，表1のデータを基にSNT-FMによるカットサイズの質，アルゴリズムの収束性，配線長・配線率について評価を行なった．評価に使用した計算機は，Fujitsu S4/630ワークステーションを使用した．

4.1 カットサイズ

各データに対し，SNT-FMを用いてチップの真ん中で分割した時のカットサイズの質について評価を行なった．我々は，カットサイズの質を評価する一つの目安として，標準偏差を用いた．

まず，最初に行なった実験は，各データに対して，ランダムな初期解からFMを1000回実行した時に得られたカットサイズの最大，最小，平均(m)，標準偏差(σ)，平均処理時間を求めた(表4)．カットサイズの相対度数分布が正規分布に従っていると仮定すると，FMを適用したときカットサイズの値は，99.7%の確率で$m-3\sigma$と$m+3\sigma$の間に来ることになる．そこで，カットサイズの質の良し悪しを$m-3\sigma$よりも小さいかどうかで評価することにした．

次に，各データに対して，ランダムな初期解からSNT-FMを20回実行した時に得られたカットサイズの最大，最小，平均，平均処理時間を求める実験を行なった．その結果を，FMの$m-3\sigma$の値，各データごとの繰り返し係数(R)と共に表5に示した．

この実験結果から，BB10のデータを除く，全てのデータで$m-3\sigma$相当，もしくはそれよりも小さなカットサイズが求まっていることが分る．BB10のデータの場合は，つながっているセルの数が2個しかないネットの比率が，他のデータに比べると，はるかに多い．このことが，他のデータ程効果が現われていない理由と考えられる．それでも，SNT-FMの平均値は，FMの最小値よりも小さいので実用的には，十分効果があると考えられる．

SNT-FMの平均処理時間は，FMの平均処理時間と繰り返し係数Rを掛け合わせたものよりも小さい．その理由は，SNT-FMでは，二回目以降に実行されるFMの初期解は，ランダムな初期解よりもカットサイズが小さいので，FMの収束が早いからである．

4.2 アルゴリズムの収束性

SNT-FMでは，Rの値をどのように決定するかが，SNT-FMの性能と速度のトレードオフ決める．そこで，AA71, CC30, EE00のデータにおいて，あるランダム解に対してSNT-FMを適用したときの，各繰り返し時における現在のカットサイズの値と，その繰り返しまでで最も小さいカットサイズの値をグラフ化して，Rと性能の関係を調べた．

AA71, CC30, EE00の結果を図8，9，10に示す．AA71, CC30はRを100に，EE00はRを200にして実行している．また，参考のため表5で求めた$m-3\sigma$の値もグラフ上に示した．

これらの結果から，SNT-FMの収束はかなり早いことが分る．具体的には，データの規模に関わりなく，Rが10以下でカットサイズが$m-3\sigma$の値に近付いている．性能・時間の効率を考えると第3.1章の式1で示したRの値が適当であるが，LSIの設計において，セルの使用率が低い場合のように，必ずしも最適なカットサイズを求める必要がない場合には，Rを10近辺の値に固定してSNT-FMも実行しても，実用的には十分な結果が得られる．

また，SNT-FMは，収束が早いだけでなく，Rの値を大きくすればするほど，より最適なカットサイズが得られることもこれらの結果から分る．このとき，注目したいのは，SNTで毎回ネットを大きく移動させているにも関わらず，現在のカットサイズは，$m-3\sigma$の近辺もしくは，それ以下の値に

表 4: FM

データ	最大	最小	平均(m)	標準偏差(σ)	平均処理時間(Sec)
AA71	242	102	167.64	24.03	1.43
BB10	415	201	301.39	45.26	5.14
CC30	889	404	651.88	72.25	8.71
DD39	1852	1394	1645.86	69.42	29.44
EE00	1805	1065	1469.97	119.42	62.11
FF12	2639	1964	2332.73	92.24	164.31

表 5: SNT-FM

データ	R	最大	最小	平均	$m-3\sigma$	平均処理時間(Sec)
AA71	26	79	49	65.95	95.55	23.40
BB10	33	219	171	183.25	165.66	58.02
CC30	52	439	349	401.35	435.12	212.63
DD39	101	610	335	436.95	1437.60	1316.61
EE00	114	804	647	727.00	1111.71	3001.15
FF12	172	912	489	702.20	2056.01	11047.33

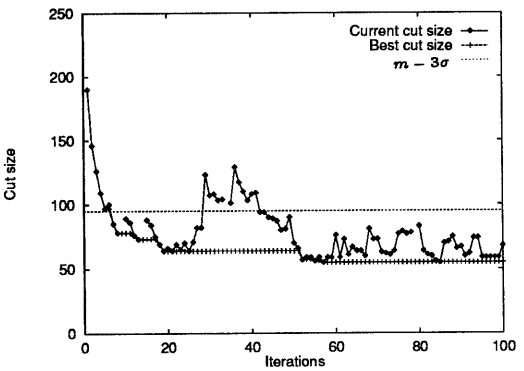

図 8: AA71

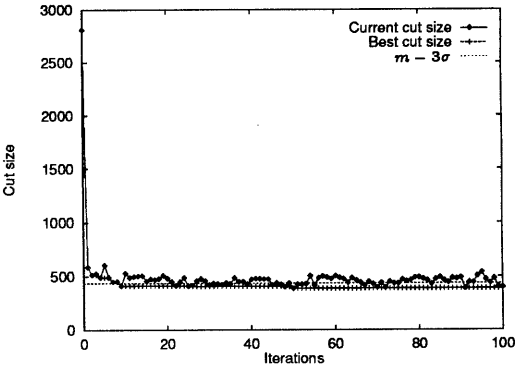

図 9: CC30

なっていることである．このことは，SNT-FMでは，複数種類の準最適解が求まる可能性があることを示している．さらに，SNTのhill-climbingの効果から，それらの準最適解の中には，まったくカットされているネットの種類が異なるものが含まれている可能性があると考えられる．

そこで，AA17とCC30のデータの中から準最適解のを選びだし，その中に解の質が異なるものが存在するか調べてみた．具体的には，求まったベストに近い解をいくつかピックアップして，カットされているネットの種類の相関を調べる実験を行なった．その結果，どちらのデータからも，50%以上ものネットの種類が異なる場合の準最適解のペアが複数個求まっていることが分った．このことから，SNT-FMでRの値を大きくすると，準最適解が複数個求まる可能性があることが，確認できた．

4.3 配線長・配線率

SOGゲートアレイの配置システムとしての，実用性を評価するために，シミュレーティッドアニーリング(SA)[8]をベースにした配置システムを用いて，時間と配線長，配線率の比較を行なった(表6)．

配線長は，論理配線長と実配線長の両方の評価を行なった．論理配線長は，rectilinear minimal spanning treeにより求めた結果である．また，実配線長は，「Touch and Cross Router」[7]のアルゴリズムをワークステーションに実装したものを実行した結果から求めた．

表 6: SAとの比較

	SA				SNT-FM			
	論理配線長	実配線長	#未配線	配置時間(H)	論理配線長	実配線長	#未配線	配置時間(H)
AA71	363999	372408	0	4.0	241680	269645	0	0.05
BB10	800789	866650	0	2.4	552249	629463	0	0.20
CC30	2266377	2270595	0	14.6	1690339	1721200	0	0.53
DD39	2994888	3044334	0	28.4	2520943	2635274	0	1.7
EE00	6275039	6338882	5	43.7	5100211	5191976	0	4.1
FF12	5678409	5827302	0	50.8	5544878	5751586	0	5.1

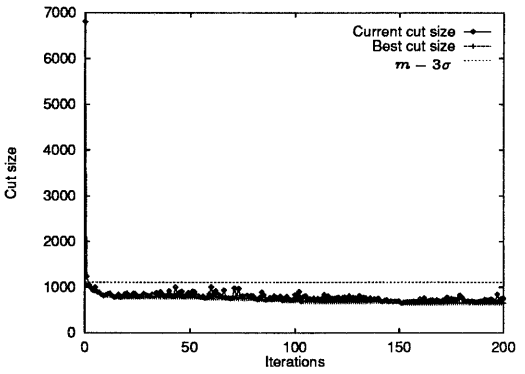

図 10: EE00

SNT-FMの詳細配置は，概略配置の結果を基に指定された分割ブロックの中にセルを詰めて行く．セルを詰めた後の配置改善は行なっていない．

SNT-FMの配線率は，ほぼSAに等しい．また，論理配線長・実配線長においても，全てのデータでSAよりも短くなっている．特に，AA71においては，33%も論理配線長が短くなっている．配置時間は，SAの1/10以下に収まっている．

5 まとめ

本論文において，FMにhill-climbingを組み合わせることによりカットサイズを最適化する配置手法SNT-FMについて述べた．実データを用いた評価を行ない，SNT-FMの有効性を示した．特に，SOGゲートアレイの配置手法として，速度，性能共に，十分実用性が高いことを示した．

謝辞

日頃御指導・御助言をいただく，並列処理研究センター白石部長に深謝いたします．また，本研究に当たり，多大なる御協力を頂いた，富士通ソフトウェア開発部の皆様に深く感謝いたします．

参考文献

[1] J. Cong, L. Hagen, and A. Kahng. Random walks for circuit clustering. In *IEEE 4th Int'l ASIC Conf.*, pages 14–21, 1991.

[2] A. E. Dunlop and B. W. Kernighan. A procedure for placement of standard-cell vlsi circuits. *IEEE Trans. on Computer-Aided Design*, CAD-4(1):92–98, January 1985.

[3] C.M. Fiducca and R.M. Mattheyses. A linear-time heuristic for improving network partitions. In *Proc. 19th DAC*, pages 175–181, 1982.

[4] J. Garbers, H. J. Promel, and A. Steger. Finding clusters in vlsi circuits. In *Proc. ICCAD*, pages 520–523, 1990.

[5] M. R. Garey and D. S. Johnson. *Computers And Intractability.* W. H. FREEMAN AND COMPANY, 1979.

[6] M. R. hartoog. Analysis of placement procedures for vlsi standard cell layout. In *Proc. 23rd DAC*, pages 314–319, 1986.

[7] K. Kawamura, T. Shindo, T. Shibuya, M. Miwatari, and Y. Ohki. Touch and cross router. In *Proc. ICCAD*, pages 56–59, 1990.

[8] S. Kirkpatrick, C. D. Gelatt. Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[9] B. Krishnamurthy. An improved min-cut algorithm for partitioning vlsi networks. *IEEE trans. on Computers*, C-33(5):438–446, 1984.

[10] T. Ohtsuki. *Layout Design and Verification*, volume 4. North-Holland, 1991.

[11] H. Shiraishi and F. Hirose. Efficient placement and routing techniques for master slice lsi. In *Proc. 17th DAC*, pages 191–197, 1980.