

先読み探索に基づく新配置最適化アルゴリズム

上田 俊晃

(株) 東芝 ULSI研究所

配置問題を状態探索問題として取扱い、配置改良の為に多段階な配置状態の先読みを行うことにより、効率的に配置状態を最適化する新しい配置手法を提案する。提案する手法は、配置状態の更新の方法として単一のセルの移動、交換、及び、1:多のセルグループとしての交換処理を使い分けて逐次配置改良を進めていくものである。さらに局所最適解に捕らわれ難くするために、先読みにより将来の改良が見込める一時的な改悪状態を受け入れながら配置改良を強力に進めるものである。本手法を実際のゲートアレイ回路に適用し、その有効性を確認したので、他の従来手法との比較も含めて報告する。

A New Placement Algorithm using Advance Search Method

Toshiaki UEDA

ULSI Research Center, TOSHIBA Corporation
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki 210, Japan

This paper deals with a combinatorial optimization process (finding the minimum of a given function for the placement problem in electrical circuit layout), such as minimizing the total wire length of signal nets. A new heuristic iterative improvement procedure is proposed. The aim of the overall scheme is to find much better solution by getting over many locally optimum solutions as possible within short computational time. This procedure repeats placement improvement by a sequence of local transformations. The proposed method is able to get over some local optimum solutions and improves a solution efficiently. The iterative improvement algorithm presented here is different from previous ones, such as simulated annealing or probabilistic improvement, in the sense that it considers group migration more than two cells at the same time and examines only a small portion of hopeful configurations which has high probability of being better by accepting the worse configuration. Experimental results show this procedure gives better solutions in computational time and quality.

1. はじめに

LSIのレイアウトCADは、素子をチップ上に配置する事と、素子間の信号線を配線する事からなる。これらを融合して取り扱った文献[1]も報告されているが、通常は配置と配線に分割して取り扱うのが現実的である。このような分割を行っても、各々の問題はNP-hardである。そのため従来考案されてきた手法の多くは、近似解を効率の良いheuristicsで求めるものにならざるを得ない。

従来、配置の良さの評価基準として、よく用いられるものは、仮想配線長や、カット数である。しかし、実際のデータでは不十分な場合もあり、局所的な配線の混雑を十分に反映した目的関数を設定して配置するといった文献が増加している[2-5]。

一方、最適化手法については、従来、目的関数(コスト)の減少する方向にのみ素子の位置交換を繰り返し、最適化を図っていた。しかし、この方法では、局所最適解に陥り、十分な最適化が達成できないという問題があった。局所最適解から抜け出すためには、目的関数値が悪くなる方向へも状態遷移する可能性を与えることが必要で、これを実現した代表的な例がシミュレイトッドアニーリング法である[6]。また、生物学的な進化の過程を模擬したアルゴリズム(シミュレイトッドイミューション法[7])を適用して、最適化を図るものもある。以上のように、配置レイアウトのメインテーマである最適化手法は、比較的単純なheuristicsから始まり、物理現象や生物学的な進化の過程の模擬といった手法も提案されて、研究開発が進められてきた。

しかし、CADツールへの要求は、より高性能なものを目指して強まる一方であり、それを解決する為の方策が種々様々な側面から追求されている。

研究対象の一つである配置レイアウトに於いては、前述したように目的関数をどの様に設定するかという問題と、その目的関数の下でいかに最適化するかという問題があり、それらはかなり独立した課題である。本稿は、主に最適化手法(アルゴリズム的側面)に焦点を当てて開発した、新しい配置最適化アルゴリズムについて報告するものである。

以下、まず第2節でレイアウトモデルとシステム構成について説明し、第3節で対象とする配置問題の定式化を行う。次に第4節で今回開発した新配置最適化アルゴリズムの特徴を述べ、第5節で本アルゴリズムの詳細な解説を行う。第6節では、実験結果をもとに、本アルゴリズムの適用効果について考察、評価する。最後に、第7節で今後の課題について述べてまとめる。

2. レイアウトモデルとシステム構成

2.1 レイアウトモデル

本稿で対象とするものは、SEA OF GATES TYPE ゲートアレイのレイアウトである(図1)。チップ周辺には1/0ブロックが配置されており、チップコア領域には、全面にトランジスタが敷き詰められている。チップ母体上には縦・横各方向にカットラインを設定し、各カットラインで囲まれる領域を格子と呼ぶ。そして、この格子は、配置改良に於ける改良の処理単位とする。

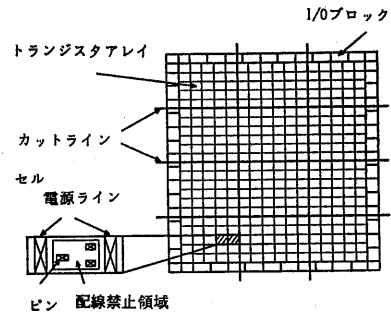


図1 レイアウトモデル

2.2 システム構成

配置レイアウトシステムの全体構成を図2に示す。配置システムは、初期配置、配置改良、詳細配置の各プログラムで構成されている。初期配置では、まず接続関係の強いセル同士をクラスタリング[9]し、これらを単位としてミニカット配置手法[8]により配置の初期状態を作成する。配置改良では、セルの移動あるいは交換を繰り返し、各格子に割り付けられているセルを部分的に入れ換えて大局的な配置最適化を図る。これについては、3節以下で詳細に記述

する。詳細配置では、格子内でセル配置位置を調整し、セル同士の重なりを取り除いて最終的なセルの位置を決定する。

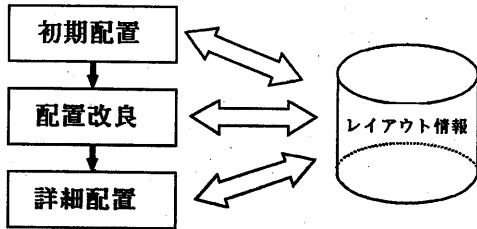


図2 配置システムの構成

3. 配置改良問題の定式化

配置改良手法としては、力学系モデルに基づく方法、セルのペア交換による方法、割当問題に帰着させる方法など[10-12]、様々な方法が提案されている。また、多目的な複数の評価指標間のトレードオフを考慮して、総合的に配置改良を図っていく方法も既に提案されている[13]。

そこで本研究では、目的関数(評価指標)をどのように設定するかという事よりも、むしろ、与えられた目的関数の下でいかに最適化していくかという最適化手法に重点を置く。従って、目的関数としては、配置で使用するどの様なものを採用しても良い。ここでは、一例として力学系モデルに基づく緩和法(Force Directed Relaxation method)での評価関数を採用する。

この配置モデルによれば、セルaに働く張力の合力が0となる位置、 (G_{ax}, G_{ay}) は、

$$G_{ax} = \frac{\sum_i C_{ai} X_i}{\sum_i C_{ai}}, \quad G_{ay} = \frac{\sum_i C_{ai} Y_i}{\sum_i C_{ai}} \quad (3.1)$$

と表される。ここで、 C_{ai} はセルa, i間の接続数、 (X_i, Y_i) はセルiの位置である。また、 (G_{ax}, G_{ay}) を含むチップ上の格子は目標格子と呼ばれる。例えば図3(a)の配置に於けるセル3の目標格子は図3(b)のEである。緩和法の基本的な考え方は、セルを目標格子へ移動して配置全体の張力を順次緩め、配置改善を図ろうとするものである。

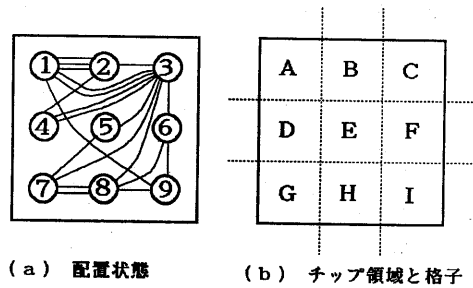


図3 緩和法の配置モデル

そこで、本研究の対象とする配置改良問題を以下の様に定義する。まず、配置すべきセルは大小様々な矩形形状である。そして、カッパラインで囲まれた各格子にどの様なセルのグループを配置するかを決定する問題を配置改良問題として取り扱う。各格子に於いては、格子内に配置されているセルの面積の総和(S_{sum})とその格子内部にセルを配置できる許容量としての配置可能面積(C_{max})の比である格子内セル充填率(D_{grid})が計算できる(3.2)。

$$D_{grid} = S_{sum} / C_{max} \quad (3.2)$$

配置改良処理としては、格子内セル充填率(占有率)を可能な限り均一化しながら、配置の最適化を進める。最適化の評価指標である目的関数としては、各セルの力学的な平衡位置からのズレの総和を採用し、これにより配置の良さを評価するものとする。従って、本稿では配置最適化問題を下の(3.3)式で表される目的関数(評価関数)の最小化問題として定式化する。そして、格子内セル充填率の均一化は配置状態の探索時に於ける制約条件として取り扱う。

Find a state (p) which minimizes the following objective function, subject to $p \in P$.

$$\text{Obj. Func.} = \sum_i \{ (G_{ix} - X_i)^2 + (G_{iy} - Y_i)^2 \} \quad (3.3)$$

ここで、 (G_{ix}, G_{iy}) は(3.1)式で求められる各セルの重心であり、座標 (X_i, Y_i) は各セルのチップ上での配置位置である。また、Pはすべての可能な配置状態の集合であり、pはその中の配置状態の一つで(3.3)式の目的関数の値が最小になる配置解(配置結果)である。

4. 新配置最適化アルゴリズムの特徴

今回提案する新しい配置最適化アルゴリズムは、局所的な最適状態にあることを認識したら、その状態を強力に脱出（山越え:hill-climbing）する手段を持つ。具体的には、何ステップか先まで新配置状態の生成を試行して改良が見られれば、その方向に改良を進めるという多段階な先読み探索機能を持つ。

本アルゴリズムでの山越えは、セルグループの連鎖的な交換、移動による状態改良により行われる（Group-Migration-Method [14]）。一般にある可能解を γ とすると、 λ 個以下の要素の交換によって γ より到達できる解の集合のどの要素よりも γ の方が良ければ、 γ は λ -Optimumと呼ばれる。 λ -Optimumは、実際には限定された特定の要素を調べることによって保証される場合が多いことから、文献[15]では、特定の要素の選択基準として重心の近傍を選ぶことにより、調べる要素数を飛躍的に減少させている。

提案する新配置最適化アルゴリズムは、この考え方を踏襲しているが、セルを玉突状に追い出すことはしない。何故なら、セルの形状、格子内セル充填率は一般に不均一であるので、セル数が1:1の玉突状の追い出し処理を行うと、格子内セル充填率の不均一性をさらに進め、配線に対して悪影響を及ぼす。従って、新アルゴリズムでは、格子ペア間で独立にセルの移動あるいは交換を連鎖させていき、1ステップ次の配置状態を生成するための試行として、その場の状況に合わせて、単一セルの移動、1:1のセル交換、或いは一つのセルと複数のセル集合との交換を使い分ける。この為に予め当該格子内でセルのクラスタリング（ソフトグルーピング）を行って、グループ単位で移動あるいは交換をする。そして、格子内セル充填率の均一化によりチップ上のセル密度を均一化する。

5. 先読み探索に基づく新配置最適化アルゴリズム

本配置最適化プログラムでは、最適化のための目的関数として(3.3)式を採用するものとし、先読みによる改良判定は、このコストの差分の累積を保持して評価する。

図4は本アルゴリズムのプロチャートである。このフローの中の、配置状態の先読み処理を図式的に示したものが図5である。最適化を進めるに当たっての基本的な考え方としては、初めから局所最適解に陥らないように人工的な手順でセルを動かしていく様な制御は非常に困難であるという観点から、一時的に局所最適解の状態になることを許しておく。ただし、ここで最適化が止まらない様にするために、局所最適解にトラップされている事を認識して、この状態から強力に抜け出して山越え（hill-climbing）をしていく。これに対応するのが先読み機能である。

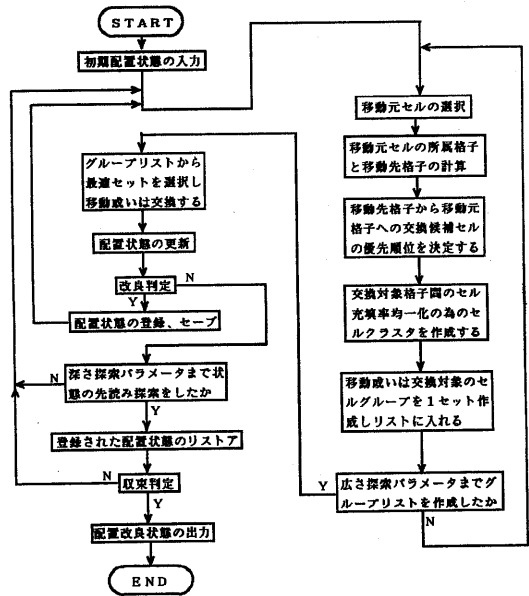


図4 本アルゴリズムのプロチャート

以下、図5を使って具体的にその詳細を説明する。まず、塗りつぶされた大円で示されているものをある配置状態（S1）とすると、ここから次候補の配置状態をパラメータに従っていくつか生成してやる。図では小円でこれらの配置状態を示している。この配置状態の候補群の中からコストの減少効果の高いものを選択し、次の配置状態として展開していく。選択された配置候補は中円（t1～t4）で示されている。図の場合は、S1の配置状態からt1の配置状態が次の配置状態として選択されることになる。

この時、配置状態が改良されれば状態更新を確定するが、S1が局所最適解の場合はこの段階での改良が見込めないため、コスト変化の累積を計算しながら、この改悪を一時的に受け入れて、次の配置状態を同様に展開していく。この様にして、次々と有力配置候補を採択し、パラメータに従った深さまで何手か先を読みながら状態遷移を続けていく。これにより、局所最適解から抜け出した、元の配置状態S1より良い配置状態（図では塗りつぶされた大円S2で示してある）を発見する。図では矢印のパスの長さは5なので、5手先まで先読みして山越えをしている事になる。そして、この時初めてS1からS2への状態更新（登録）を一度に確定して改良が進む。これにより、将来の改良に必要な改悪のみが許される事になる。

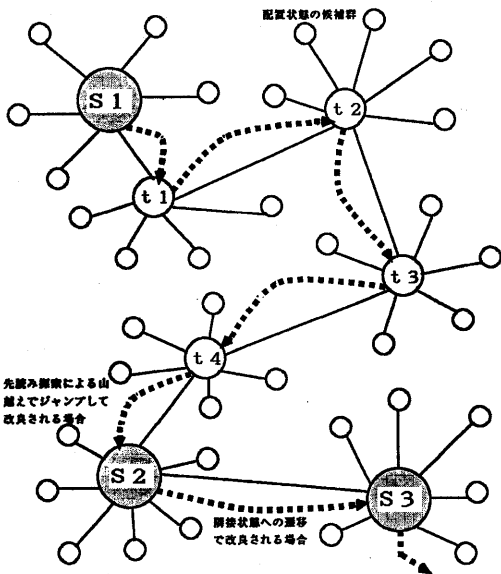


図5. 先読み探索による山越え(Hill Climbing)の過程

先読みは、ある配置状態から到達可能なより良い状態を探索するために行われる。したがって、いつも先読みにより状態空間上で離れた状態に一気に移るのではなく、隣接する（状態が近い）状態で改良されるものがあれば、貪欲戦略（greedy algorithm）的に状態更新を確定し、改良を進める。図では塗りつぶされた大円の状態S2から状態S3に移るときがこの場合を示

している。これによって、改良が進みやすい最適化の初期段階では通常の改良方向への状態遷移だけが起り、急速に良い状態へ収束する。そして、後半で局所最適解にトラップされて改良が進みにくくなったあたりで、先読みの触手を伸ばし、局所最適解から脱出する。この後半段階でS1からS2のような山越え遷移が起り、更なる最適化が進む可能性が開ける。

山越えを行うに際しては、一連の連鎖的な各格子間でのセルの入れ替えが効率的なものになる様に、その配置状態から広さパラメータに従って探索し、配置最適化に効果のあるものを連鎖系列の一要素として選んでいく。これは、処理効率を上げる為、なるべく短い連鎖で改良判定を行う為である。

本アルゴリズムに於いて、図中、小円で示した次候補の配置状態をどの程度広く生成するかという広さパラメータは、次の一配置試行（一手先）をどれほど慎重に探索するかを決定しており、このパラメータが大きいほど、現配置状態の近傍を丁寧に探索する。配置状態の先読みについては、深さパラメータで与えられる連鎖の長さを限度として探索し、改良状態を発見していく。この連鎖の長さは、現配置状態からどの程度遠い状態まで先読み探索するか（何手先まで読むか）を決めるパラメータといえる。

この最適化手法、特に先読みに関しては、将棋等の対局での人間の思考過程と類似(Analogy)している所が多い。将棋に於いても有利な状態に局面を変えていく時、最初（序盤）は比較的容易に速いテンポで良さそうな手を打って進んでいくが、煮詰まってくると先を読み出し、ゆっくりとした慎重な展開になる。そして、最善手を決める状況での状態数は非常に多いので、このような場合の先読みとしては、最も良さそうな手順でかなり深く先を読むことになる。そして有利と判断される時には、その方向に局面（状態）を変えていく手を打つことになる。このような人間の思考過程（プロセス）を本アルゴリズムでは取り入れている。

一方、(3.3)式の目的関数下で配置状態の最適化を進めるにしたがって、各格子間でのセルグループの移動あるいは交換が多数行われるので、その間に格子内セル充填率（占有率）の均一化も同時に進んでいく。

6. 実験結果とその評価

本稿で提案した先読み探索に基づく新配置最適化手法の有効性を確認する為、この最適化アルゴリズムをSEA OF GATES TYPE ゲートレイ配置改良プログラムとしてインプリメントし、計算機実験を行った。プログラムはC言語で作成し、使用した計算機は、10 MIPS 相当のワークステーション(SUN4/260)である。使用したデータは約800セルの実際のゲートレイ回路のものであり、表1にその諸元を示す。

表1 使用したデータ

母体サイズ (Eゲート)	セル数	ネット数	ピン/ネット	ゲート利用率 (%)
5	855	784	3.54	52.4

以下で本アルゴリズムの適用結果について述べる。表2は、今回開発した先読み探索に基づく配置最適化手法と、シミュレ-ティッド'ア-リソク'法等の従来提案されている最適化手法を、それぞれ適用した配置実験結果である。本手法の最適化効果・処理効率を他の手法と比較評価する為、最適化処理が収束した後の目的関数値と改良が収束するまでの処理時間を各手法別に示してある。目的関数値は、第3節(3.3)式を基に算出した値であり、小さい方が良い配置状態を表している。表3は、それぞれの手法による最適化処理を実行し、約1時間後の最適化状況を目的関数値で示したものである。

表2 各手法を適用した場合の配置性能の比較結果

	初期配置 状態	従来手法	シミュレ-ティッド' ア-リソク'法	シミュレ-ティッド' E&Vソ-ソソソ法	本提案手法 (先読み法)
目的関数値 (コスト)	41783220	25633085	17925234	29083693	19717758
処理時間 (CPU, sec)	—	2795	153904	3203	3337

表3 最適化効率の比較結果

(改良処理を開始してからCPU timeで約1時間後の最適化状況)

	初期配置 状態	従来手法	シミュレ-ティッド' ア-リソク'法	シミュレ-ティッド' E&Vソ-ソソソ法	本提案手法 (先読み法)
目的関数値 (コスト)	41783220	24975826	36322185	27504861	19305621

表3の実験結果によると、従来手法('ア'ソリス'ソソソソソ'法)により最小化された評価関数値を1とした場合に、本手法(先読み探索手法)は0.77、シミュレ-ティッド'ア-リソク'法は1.45となっている。また、表2の結果を見るとシミュレ-ティッド'ア-リソク'法は改良が収束した段階では0.72までコストが下がるが、その為には他の手法より約50倍長い時間を掛けて実行を続けなければならないことが判る。改良が収束した後の最適化効果を見てみると、本手法は、従来手法に比べて約2.4%の性能向上が実現できている。

これら各手法の最適化の過程を視覚的に示すために、処理時間を横軸に目的関数値を縦軸にプロットしたものが図6～図9である。実用的な時間内でどこまで目的関数値(コスト)が下がっているかで、その最適化アルゴリズムの性能の善し悪しが判断できる。ただし図7のシミュレ-ティッド'ア-リソク'法はこの処理時間内ではコストが殆ど下がらず収束もしない。そこで、これが収束するまで実行を続けた結果を図10に示す。図10のグラフの縦軸は図6～図9のものと同じスケールであるが、処理時間を示す横軸については、図6～図9のものに対して1/100に縮小してある。

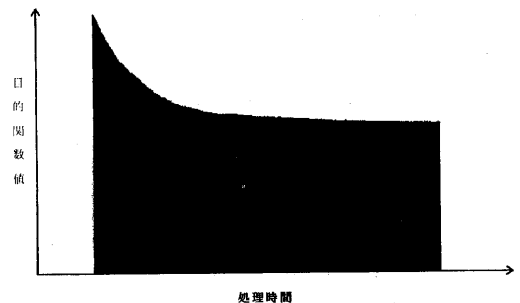


図6 従来手法

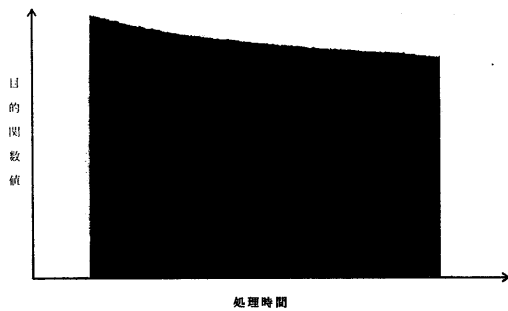


図7 シミュレーテッドアニーリング法

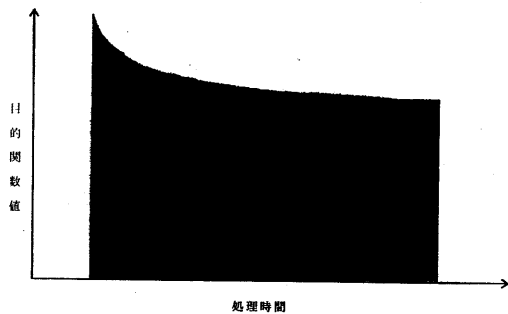


図8 シミュレーテッドエボリューション法

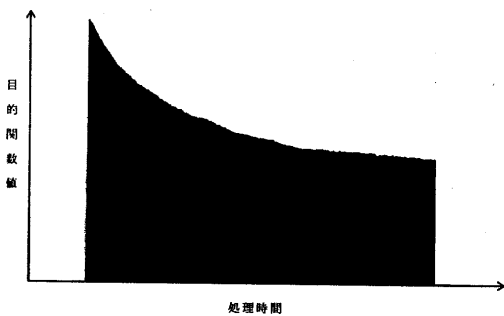


図9 本提案手法(先読み探索に基づく新配置最適化アルゴリズム)

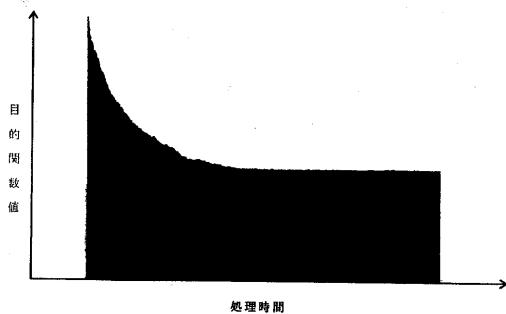


図10 シミュレーテッドアニーリング法の改良取束状況

これらのグラフの様子から、最適化の経済性を考えて、処理時間と処理効果の両面から総合的に判断すれば、今回提案したアルゴリズムは、最適化の制御をより効率的に行う有効な手法であると考えられる。

この優位性をもたらす、先読み探索による山越えの過程を、時間と共に示したものが図11である。このグラフも処理時間を横軸に、評価関数値を縦軸にプロットしてあり、一つの棒グラフは、一つの配置状態に対応している。そして、その高さが配置の善し悪しを表しており、低い方が良い配置状態である。図から判るように、複数の局所最適解から脱出しながら、最適化が進んでいる。グラフ上で谷になっている所が局所最適解に対応しており、幾つもの山を越えながら、次のより低い谷へ改良が進んでいる。つまり、この何手か先の谷を探索して、発見できた時のみ、山の部分の状態改悪をアクセプトしながら山越えをする。これを繰り返して更なる改良が進み、強力な最適化がなされている。

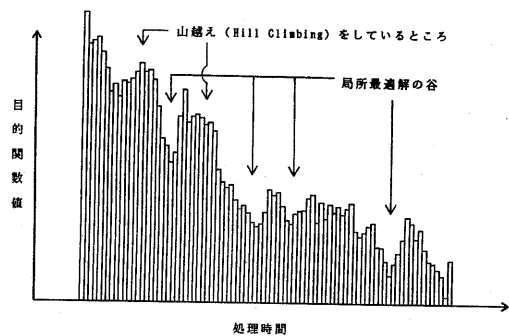


図11 本提案手法により局所最適解を脱出しながら配置改良が進んでいく様子

7. まとめ

配置レイアウト技術には二つの重要な側面がある。一つは目的関数の設定の仕方であり、他の一つは最適化の進め方である。本研究では、最適化性能の向上という観点から、後者の問題に焦点を当てて配置レイアウトを取り扱った。そして、効率の良い最適化アルゴリズムを新たに考案し、これに基づく配置改良プログラムを開発して、その有効性を確認した。

今後の研究課題としては、状態探索の並列化、階層処理による高速化などが挙げられる。

[参考文献]

- [1] K. J. Loosemore, "Automated Layout of integrated circuits", Proc. 1979 IEEE Int. Symp. Circuits and Systems. (Tokyo, Japan), pp. 665-668, 1979.
- [2] K. H. Khokhani and A. M. Patel, "The Chip Layout Problem - A Placement Procedure", Proc. 14th Design Automation Conference, 1977, pp. 291-297.
- [3] A. M. Patel, "A Wirability Placement Algorithm for Hierarchical VLSI Layout", Proc. ICCD'84, 1984, pp. 344-350.
- [4] H. Shiraishi and F. Hirose, "Efficient Placement and Routing Technique for MasterSlice LSI", Proc. 17th Design Automation Conference, 1980, pp. 458-464.
- [5] K. Klein, et al., "A Study on Bipolar VLSI Gate-Arrays Assuming Four Layers of Metal", IEEE J. of Solid State Circuits, vol. sc-17, no. 3, June, 1982, pp. 472-480.
- [6] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by simulated annealing" Science, vol. 220, no. 4598, May 1983, pp. 671-680.
- [7] Ralph M. King and Prithviraj Banerjee, "ESP: Placement by Simulated Evolution", IEEE Transactions on Computer-Aided Design, vol. 8, no. 3, March 1989, pp. 245-256.
- [8] M. A. Breuer, "MIN-CUT Placement", J. of Design Automation and Fault-Tolerant Computing, vol. 1, no. 4, Oct. 1987, pp. 343-362.
- [9] J. M. Kurtzberg, "Algorithms for backplane formation", Microelectronics in Large Systems, Washington, D. C.: Spartan Books, 1965, pp. 51-76.
- [10] M. Hanan, P. K. Wolff Sr. and B. J. Agule, "Some experimental results on placement techniques", Proceedings of the 13th Design Automation Conference, 1976, pp. 214-224.
- [11] M. Hanan and J. M. Kurtzberg, "Placement Techniques", Chap. 5 in Design Automation of Digital Systems, 1 (Ed. by Breuer, M. A.), 1972, pp. 213-282. Prentice Hall.
- [12] L. Steinberg, "The backboard wiring problem: a placement algorithm", SIAM Review, vol. 3, no. 1, January 1961, pp. 37-50.
- [13] M. Murakata, et al., "A Placement Algorithm for Sea-of-gate Gate Array by Goal Programming Method", Information Processing Society of Japan, vol. 90, no. 14, 90-DA-51-2, 1990.
- [14] B. W. Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs", Bell System Technical Journal, vol. 49, no. 2, 1970, pp. 291-307.
- [15] S. Goto, "An Efficient Algorithm for the Two-Dimensional Placement Problem in Electrical Circuit Layout", IEEE Transactions on Circuits and Systems, vol. CAS-28, no. 1, 1981, pp. 12-18.