

初期解をファジイ推論する GA/SA を用いた VLSI フロアプラン手法の精度向上

Accuracy Improvement of VLSI Floorplanning Based on Fuzzy Inference and GA/SA

今井 宏規[†] 梅澤 登志矢[†] 江口 一彦^{††}

Hiroki IMAI, Toshiya UMEZAWA, Kazuhiko EGUCHI

Abstract Rapid increase of the scale of integration requires higher knowledge and well trained skills of experienced design engineers. However it is usually difficult for novice engineers to perform optimized design of initial and macroscopic placement in floorplanning. This paper proposes to apply the soft computing technology mostly fuzzy inference and genetic algorithms to automate the floorplanning design which decides a macroscopic placement of the top layer of LSI physical implementation. ISPD98 benchmark data is used for evaluation. The relation among several parameters of fuzzy inference and genetic algorithms and placement cost is discussed. The relation between I/O pins and the cost is also discussed. Simulated annealing is employed after genetic algorithms to avoid local optimization.

1. はじめに

VLSI の詳細設計段階における配置配線では高度な自動化が進んでいるが、大きな IP やブロックを含む最上位階層のレイアウト概略配置、即ちフロアプランの段階では、熟練設計者による人手の介入が必要になるケースは少ない。

さらに最近では集積度の増大に伴い、論理合成後に行われる通常フロアプランだけでなく、RTL の段階で最終的なチップイメージを把握して論理合成のためのタイミング制約を高精度に求めることを目的とした RTL フロアプランも広く行われるようになった。

いずれのフロアプランにおいても、物理的な形状とサイズが固定したハード IP、アスペクト比が決まっていないソフト IP、詳細設計もまだ定まっていないさまざまなブロックや単体レベルの機能素子等が混在するなか、チップサイズ、性能、特性、シグナルインテグリティ、消費電

力等々きわめて広範囲にわたる設計要素のトレードオフを考慮しながら最上位のマクロ的な配置を決定して行かなければならない。一方詳細物理設計へ入ってからフロアプランへ手戻りすることは開発期間、開発コストへ多大な影響を及ぼし、それ故にフロアプランの良否がチップの開発期間とコストを左右する。

LSI のフロアプラン自動化は Otten による Slicing 手法の提案^[1]と D.F.Wong らのグループによるその改良^{[2]-[8]}、またグラフ理論に基づいた手法^{[9]-[11]}や Cohoon その他による遺伝的アルゴリズムの適用^{[12]-[15]}など種々の手法が提案・開発されてきた。

また配置問題を、 m 個の実寸法を持つ長方形を要素とする集合 M を各要素の長方形が重なる (overlap) ことなく最小の面積に詰め合わせる (pack) Rectangular Packing Problem として解く研究も続けられている^{[16][17]}

一般に LSI 上での配置問題は指定されたコスト関数に対する最適化問題として定式化されるが、フロアプランにおいては配置対象となるブロックの形状は長方形に限定できてもそのサイズやアスペクト比は小さなものから

[†]愛知工業大学 電気電子工学専攻 (豊田市)

^{††}愛知工業大学 電子工学科 (豊田市)

大きなものまで幅広く分布し,考慮しなければならない設計要素の多さ故に,コスト関数の設定が困難である.

同一テクノロジーで同じような規模であっても,例えば ASIC の場合と汎用のマイクロコントローラーの場合とではチップ物理設計における最適解のトレードオフが異なるように,対象 LSI がどんな用途を狙っているかによってもフロアプランにおけるコスト関数は変わってくる.このため商用の自動フロアプランツールを使っても後から設計者による人手介入が必要になるケースが多い.熟練した設計者は配置問題に対する定量的な定式化がなくても自らの技術と知見に基づき,そのチップのアプリケーションも考慮してフロアプランの設計を行なうことが出来る.

上田らは熟練技術者の知見をフロアプランの初期値に反映することを狙いとして,ファジィ推論と遺伝的アルゴリズムを核としたソフトコンピューティングをフロアプランに適用する報告^{[18]-[20]}を行ってきた.本研究では,上田らの手法を改良するために,I/Oピン集合モデルの拡張,及び,遺伝的アルゴリズムの処理後にシミュレーテッドアニーリング法を適用して,部分最適に陥った個体の救済を提案し,IPSD98 ベンチマークスイート^[21]のうちibm01及びibm05を適用して評価・考察を行った.

2. 従来手法の概要

従来手法^[20]は大きく分けて2段階のプロセスで構成される.まず熟練設計者の経験と知見に基づいてファジィ規則を導出し,ファジィ推論によって主要なブロックの配置位置を決定する.但しファジィ推論だけですべてのブロック配置を推論することはファジィ規則が複雑になりすぎて実用的ではなくなる可能性がある.このため,ファジィ推論で位置決定することが適切ではないブロック,すなわちメンバーシップ関数によって計算される適合度があるレベルより低いブロックの配置は遺伝的アルゴリズムによる最適化問題として解いている.図1に処理の流れの概要を示す.

2.1 遺伝的アルゴリズムの Coding

遺伝的アルゴリズムにおいて遺伝子と対象問題の対応を定義する coding を図2に示す.

1つの配置案を1個体とし,X,Y,Rの3種類の染色体(Chromosome)を考える.X,Yは各ブロックの中心座標 (x_i, y_i) に,Rは回転に対応する.R=0は回転なし,R=1は90度時計方向回転を意味する. n 個のブロックがあるときは,各染色体は n 個の遺伝子(gene)を持つ.

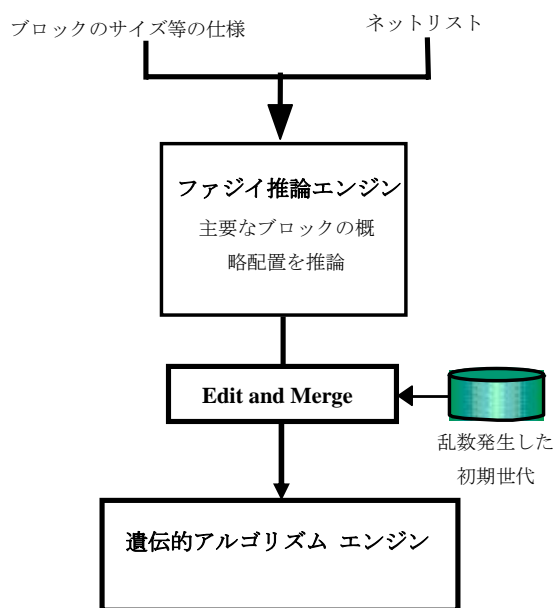


図1 ファジィ推論処理手順

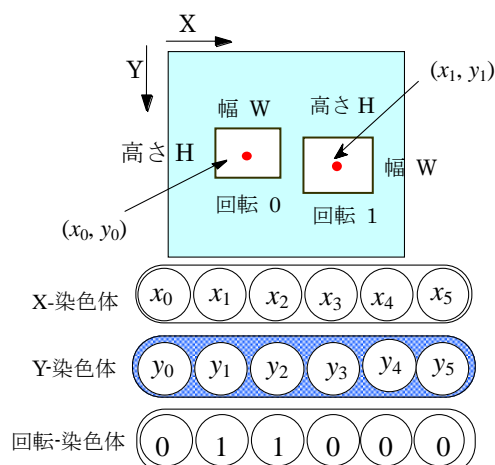


図2 遺伝的アルゴリズムの coding

2.2 評価関数

簡単のためブロック間の総仮想配線長最小を評価関数とする.遺伝的アルゴリズムの処理の過程で発生するブロック同士の重なり,チップ領域外へのはみ出しを除くため,これらには高いペナルティをつけて評価関数に組み入れコスト関数とした.また総配線長を評価する際に,I/Oパッドと内部ブロック間の配線については内部ブロック間同士の配線に対して10ないし20倍の重みをつけている.

VLSIでは内部ブロック間配線のネット数は $10^3 \sim 10^5$

のオーダーになるが、外部ピンは通常多くても数百のオーダーである。外部ピンすなわちI/Oバッファへ接続するネットは、本数は少ないが配置への影響は極めて大きいからである。式(1)にコスト関数 E の計算式を示す。

$$E = \alpha \sum_i L_i + \beta \sum_i S_i + \gamma \sum_i OV_i \quad (1)$$

ここに L_i : 配線長, S_i : ブロック同士が重なり合う場合その面積, OV_i : チップ領域外へはみ出るブロックがあるときその面積, を示す。 α, β, γ はチューニングパラメータである。以下に述べるプログラム実験では $\alpha=1, \beta=10, \gamma=1000$ とした。

3. シミュレーテッドアニーリング

シミュレーテッドアニーリング(Simulated Annealing :SA)^[22] はMetropolisらが1953年に発表^[23]した焼きなましと呼ばれる過熱炉内の固体の冷却過程をシミュレートするアルゴリズムに端を発し、最適化問題、特に組み合わせ最適化問題を解く汎用近似解法の1つとして用いられている。その概略処理を図.3に示す。シミュレーテッドアニーリングは、局所探索をランダムに行いながら、更に解に改良が見られない場合でも新しい解に移る可能性を残すことで局所解に陥ることを防ぐことができる点に特徴がある。

本報告では遺伝的アルゴリズムによって得た解が広域探索の結果部分最適に陥っている場合もあると仮定し、この解を初期値としてシミュレーテッドアニーリングを適用した。この試みにより部分最適を脱してコスト関数値が改善するかを検討した。

クーリングではアニーリング(徐冷)の第 k ステップの温度 T_k を与え、次のステップの温度 T_{k+1} を設定する。最適解への漸近収束性を保障するためには、式(2)に示す対数型アニーリング以上に冷やしてはならないが、それでは現実の応用にはあまりにも遅い。そこで予備実験においてはもう少し速い式(3)を使用した。この式でも一度の計算が数時間かかったため、さらに計算速度の速い式(4)を採用した。一方である程度の解精度を得るために以下のプログラム実験においては $\gamma=0.93$ としている。ここに T は温度、 k は計算回数を示し a, b, γ は定数である。

4. ISPD98 ベンチマーク

ISPD(International Symposium on Physical

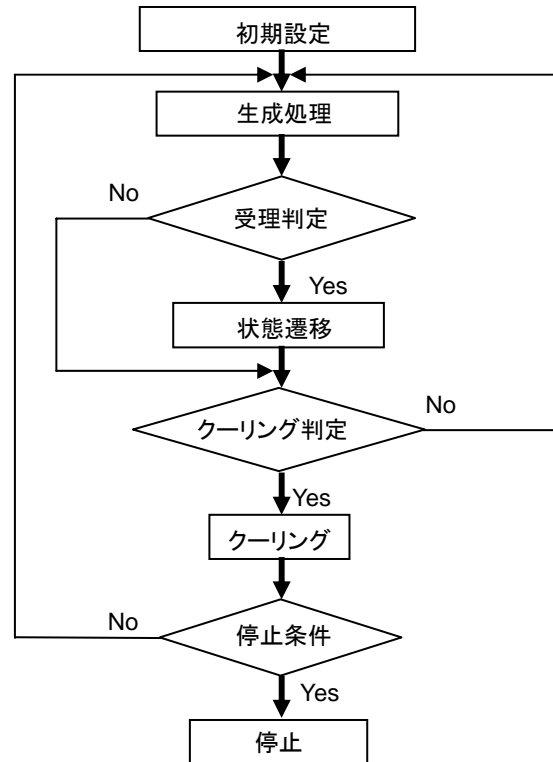


図3 SA法の流れ

$$T_{k+1} = \frac{T_1}{\log k} \quad (2)$$

$$T_{k+1} = \frac{a}{1 + bk} \quad (3)$$

$$T_{k+1} = \gamma T_k \quad (0.8 \leq \gamma < 1) \quad (4)$$

表1 ibm01, ibm05Data

Circuit	# Cells	# Pads	#Modules	# Nets	#Pins	Max%
ibm01	12506	246	12752	14111	50566	6.37
ibm05	28146	1201	29347	28446	126308	0

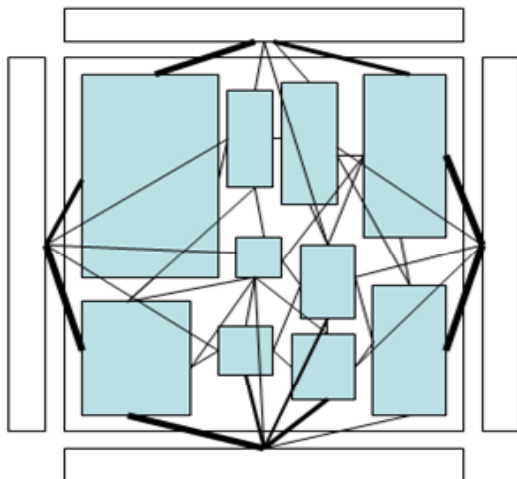
Design)98ベンチマークスイート^[21]はIBMにより1998年に発表されたベンチマークスイートである。現在ではISPD2005が最新のものであるが、今回は過去の報告において使用例が多く、パラメータの取り扱いも容易なISPD98 ibm01及びibm05を採用した。表1にibm01及びibm05のデータ仕様を示す。ISPD98ベンチマークスイートは各セルの幅と高さは公開されていないため、実験における面積は特定のサイズを与えない比率として扱う。

5. ベンチマークデータの作成

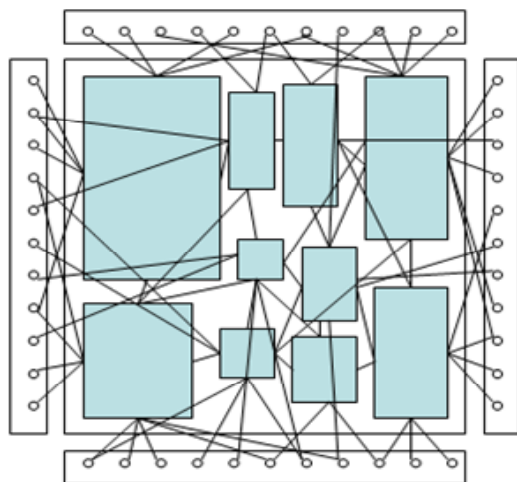
フロアプランにおいては、チップ最上位層に配置されるブロックの面積と各ブロック間相互の結合度、及び各ブロックと I/O ピン間の接続データを必要とする。そこでベンチマークデータから評価に必要なパーティションデータを得るため、分割プログラムを作成した。Module 間の結合度を基準として、最上位層で扱うブロックの数をある程度制限して分割を行った。以下の実験においては分割数 20 のデータを使用する。

6. I/O バッファブロックモデルの拡張

従来プログラムでは、チップ上下左右の四辺に外部ピンの集合、すなわち I/O バッファブロックを配置し、各 I/O



(a)従来の I/O 部と配線



(b)新しい I/O 部と配線

図 4 I/O ピンの扱い

バッファブロックの中心座標と各ブロックの中心間の配線長を求めて各ブロックと I/O バッファ間の配線長とするモデルを仕様していた。本研究では、I/O バッファブロックとして一括にせず、実際のピン配置に即して I/O ピンに座標を与え、個々の I/O ピンとブロックの中心間との配線長を計算するモデルへと拡張した。これにより I/O バッファまわりの配線長はより現実的なものを得ることが出来ると考える。

7. プログラム実験

まず従来プログラムで使用したデータにより、開発プログラムによるシミュレーテッドアニーリングの適用前と適用後の解の評価を行った。次に第 5 節にて作成したベンチマークデータを用いてシミュレーテッドアニーリングの適用前と適用後の解と実行時間の評価を行った。

シミュレーテッドアニーリングは、遺伝的アルゴリズムで探索した最終世代の解すべてを初期値として適用し、求めたすべての解のコスト関数値を比較して最良の解を最終解とした。温度設定は初期温度を 4000、終了温度を 1 とした。また、終了条件として 100 回以上の計算を行っても解の更新が認められなかった場合終了するようにした。初期温度はその温度で 100 回アニーリングを行ったとき、最も改善量が大きかった場合でも 50%の確率で変更が受理される温度とした。

8. 実験結果とその評価

図 5 に従来プログラムによる試行例を示す。これに対してシミュレーテッドアニーリングを適用した結果を図 6 に示す。

図 7 は横軸に遺伝的アルゴリズムの世代数をとり、縦軸にコスト関数の推移を示したものである。遺伝的アルゴリズムは 200 世代まで実行してコスト関数値は 32929 で収束し、それ以上の改善は見られなかった。遺伝的アルゴリズムの 50 世代目、100 世代目、200 世代目の最終値を初期値としてシミュレーテッドアニーリングを適用した。その結果それぞれ 29963, 29195, 29995 と早い世代で適用するほど大きな改善効果を得ることができた。

次に ibm01 を適用して、世代数は 200、初期人口を 3000、チップエリアの相対面積を 3100×3100 としてシミュレーテッドアニーリングの適用前後における比較実験を行った。図 8 にシミュレーテッドアニーリング適用前の試行例を示す。これに対してシミュレーテッドアニーリングを適用した結果を図 9 に示す。図 8 のコストは 1179957、図 9 のコストは 1179350 となり、約 2.3%のコストの改善

が見られた。このとき処理時間は CPU 時間で図 8 のほうが 536.92sec であり、その結果にシミュレーテッドアニーリングを適用した図 9 は 569.53sec で、シミュレーテッド

アニーリングの適用による時間増加は約 4%程度で収まっている。

また、ibm05 を適用して、世代数は 150、初期人口は 3000、チップエリアの相対面積を 3300×3300 としてシミュレーテッドアニーリングの適用による改善実験を行った。図 10 にシミュレーテッドアニーリング適用前の試行例を示す。これに対してシミュレーテッドアニーリングを適用した結果を図 11 に示す。

図 10 のコストは 1387990、図 11 のコストは 1347853 となり、約 2.9%のコストの改善が見られた。このとき処理時間は CPU 時間で図 10 のほうが 528.36sec であり、その結果にシミュレーテッドアニーリングを適用した図 11 は 584.84sec で、シミュレーテッドアニーリングの適用による時間増加は約 10.6%程度となった。

図 12 は ibm01 と ibm05 のそれぞれのデータを横軸に遺伝的アルゴリズムの世代数を取り、縦軸にコスト関数の

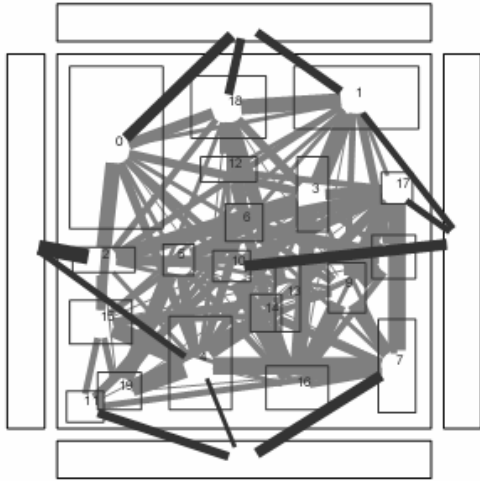


図 5 従来データ(SA 適用前)

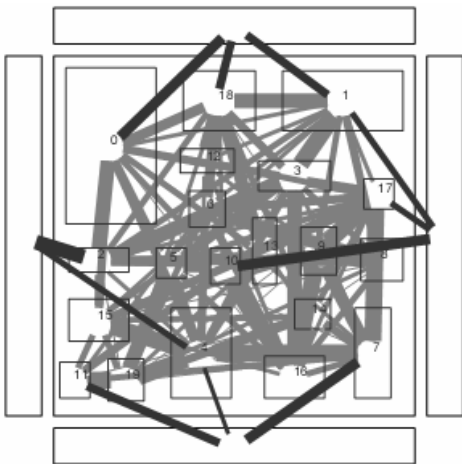


図 6 従来データ(SA 適用後)

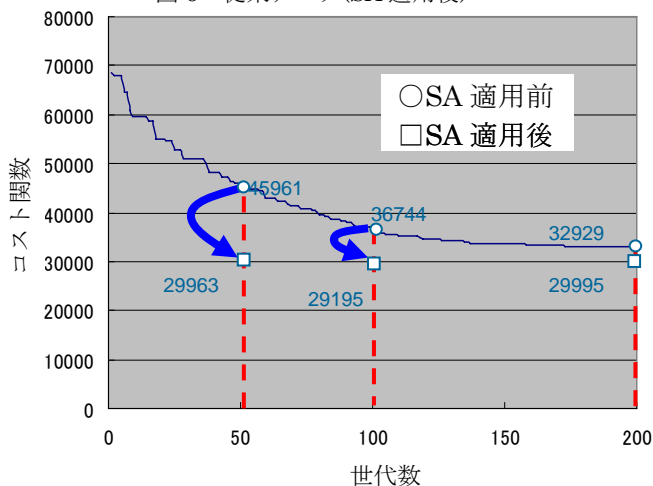


図 7 コスト関数の対世代変化

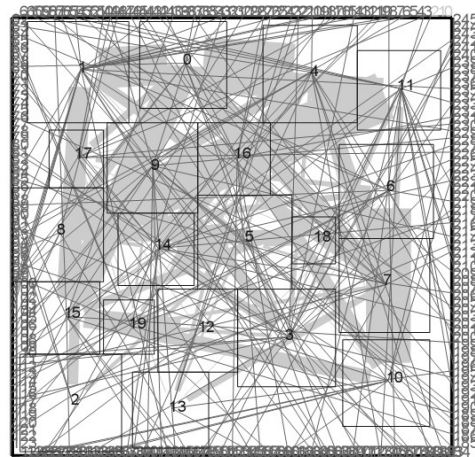


図 8 ibm01 試行例(SA 適用前)

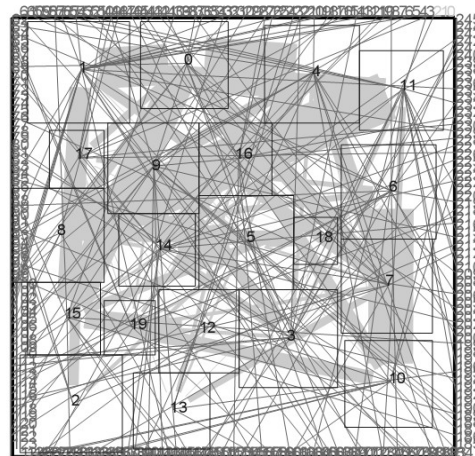


図 9 ibm01 試行例(SA 適用後)

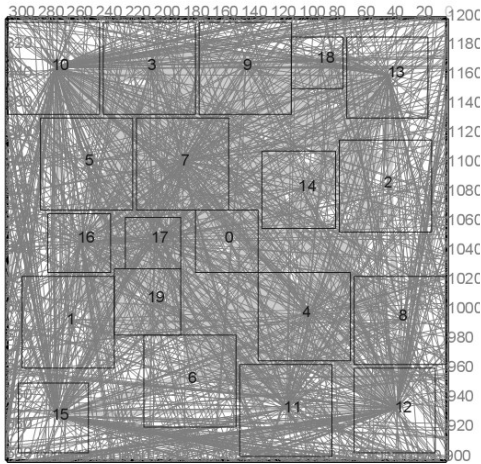


Fig.6.5 ibm05 試行例(SA 適用前)

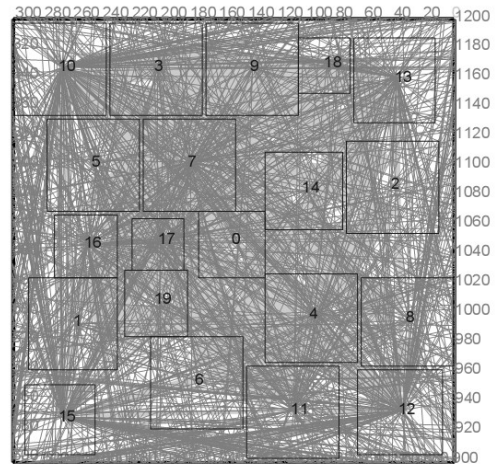


Fig.6.6 ibm05 試行例(SA 適用後)

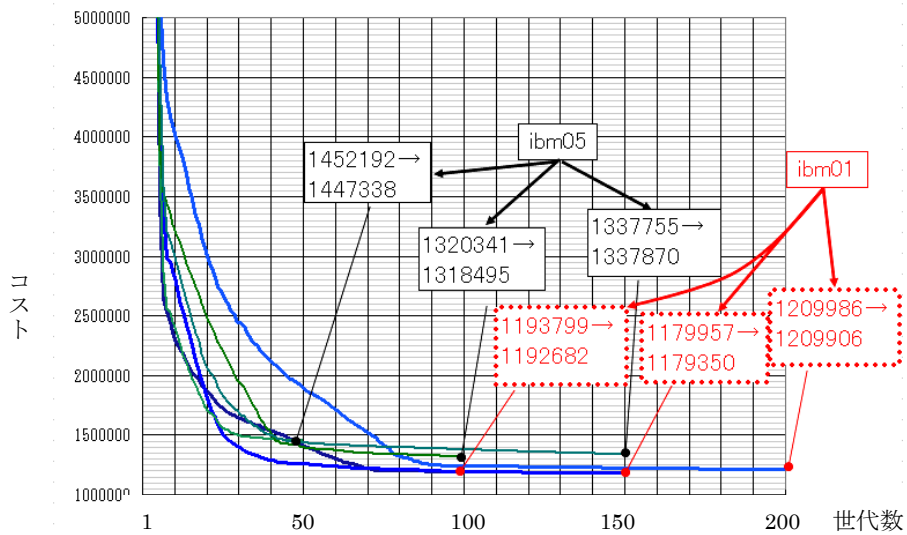


図 12 各世代のコスト比較

推移を示したものである。遺伝的アルゴリズムは 200 世代まで実行してコスト関数値は 1209986 で収束し、それ以上の改善は見られなかった。遺伝的アルゴリズムの 100 世代目、150 世代目、200 世代目の最終値を初期値としてシミュレーテッドアニーリングを適用した。そのときの CPU 時間とコストの結果を図 13 に示す。コスト改善においては 200 世代で改善幅が一番小さく、150 世代でもっとも改善された。

この結果から、ある程度初期解に依存するが、遺伝的アルゴリズムの世代数が比較的早い段階でシミュレーテッドアニーリングを適用することにより、解の改善をする

ことが出来ると考えられる。特に遺伝的アルゴリズムでは、解の優れた個体が出現すると一気に他の個体にも広まり、そのまま収束へ向かって解が改善されなくなってしまった。しかし、この解にシミュレーテッドアニーリングを適用することで、遺伝的アルゴリズムの局所最適解を脱し、さらなる解の改善をすることが確認できた。また、シミュレーテッドアニーリングを追加したことによる CPU 時間の増加は最大でも 10%前後に留まっており、クーリングで採用した式と γ の値が適切な範囲であったと考えられる。

最後に本研究の配置結果をS.N. Adyaらの配置結果^[24]と比較した。S N. AdyaらはISPD98 ベンチマークスイートのひとつであるibm02 を適用して配置を行っている。配置は相対面積 3500×3500 のチップエリア中に配線長をHPWL(Half-Perimeter Wire-Length)で考慮してフロアプランを行っている。本研究のプログラムによるフロアプランでは、配線長の考慮をHPWLと同等のマンハッタン距離で行っているものの、一般的に研究されているフロアプランと異なりチップ周辺部のI/Oピンを考慮しているため、その配置結果を比較することは難しい。しかし現実の配置においてはI/Oピンの存在は無視できず、より現実的な配置の出来る本手法の方が優位であると考えられる。

9. 結論

ファジイ推論と遺伝的アルゴリズムによるフロアプラン自動化手法に加え、I/O バッファモデルの拡張とシミュレーテッドアニーリング法を適用して部分最適に陥った可能性のある個体を救済する手法を提案した。また、この手法に対して ISPD98 ベンチマークスイートを適用して評価を行った。

プログラム実験により、本手法に適用した ISPD98 ベンチマークスイートのうち ibm01 及び ibm05 を適用したどちらの場合も、遺伝的アルゴリズムの解にシミュレーテッドアニーリングを適用することで部分最適に陥っていた解を現実的な計算時間で改善することが出来た。

また、より良い配置結果を得るための改善策及び課題として、ファジイ推論、遺伝的アルゴリズム、シミュレーテッドアニーリングの各種パラメータの調整、コスト関数の改善、ISPD2005 等新しいベンチマークデータの適用、他手法との相対的評価が必要である。

参考文献

[1] R.H.J.M. Otten: "Automatic Floorplan Design," Proc. of 19th Design Automation Conference, pp261-267, June 1982
 [2] D.F. Wong, CL. Liu: "A New Algorithm for Floorplanning," Proc. of 23rd Design Automation Conference, pp101-105, June, 1986
 [3] T. Wang, D.F. Wong: "An Optimal Algorithm for Floorplan Area Optimization," Proc. of 27th Design Automation Conference, pp 180- 186, June 1990
 [4] T. Wang, D.F. Wong: "Optimal Floorplan Area Optimization," IEEE Trans. on Computer-Aided Design, Vol. 11, No. 8, pp992-1002, August 1992

[5] S. Wimer, I. Koren, I. Cederbaum: "Optimal Aspect Ratios of Building Blocks in VLSI," IEEE Trans. On Computer-Aided Design, Vol. 8, No.2, ~139-145, February 1989
 [6] M.Z. Kang, W.W. Dai: "Arbitrary Rectilinear Block Based on Sequence Pair," Proc. of ICCAD 98, pp259-266,

	100世代	150世代	200世代
コスト(SA適用前)	1193799	1179957	1209986
コスト(SA適用後)	1192682	1179350	1209906
SA適用前(sec)	478.77	536.92	780.2
SA適用後(sec)	514.28	569.53	818.91

図 13 各世代のコストと CPU 時間

November 1998
 [7] F.Y. Young, D.F. Wong: "Slicing Floorplans with Pre-placed Modules," Proc. of ICCAD 98, pp252-258, November 1998
 [8] F.Y. Young, D.F. Wong: "How Good Are Slicing Floorplans," Proc. of International Symposium on Physical Design 97, pp144-149, 1997
 [9] B. Lokanathan, E. Kinnen: "Performance Optimized Floor Planning by Graph Planarization," Proc. of 26th Design Automation Conference, pp116-121, June 1989
 [10] T. Wang, D.F. Wong: "A Graph Theoretique to Speed up Floorplan Area Optimization," Proc. of 29th Design Automation Conference, pp62-68, June 1992
 [11] P.S. Gupta, S. Sur-Kolay: "Slicibility of Rectangular Graphs and Floorplan Optimization," Proc. of International Symposium on Physical Design 97, pp150-155, 1997
 [12] J.P. Cohoon, W.D. Paris: "Genetic Placement," Proc. of ICCAD 86, pp422-425, November 1986
 [13] J.P. Cohoon, S.U. Hedge, W.N. Martin, D. Richards: "Floorplan Design Using Distributed Genetic Algorithms," Proc of ICCAD 88, pp452-455, November 1988
 [14] J.P. Cohoon, S.U. Hedge, W.N. Martin, D.S. Richards: "Distributed Genetic Algorithms for the Floorplan Design Problem," IEEE Trans. on Computer-Aided Design, Vol. 10, No. 4, pp483-492, April 1991
 [15] M. Rebaudengo, M.S. Reorda: "GALLO: A IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 15, No. 8, pp943-951, August 1996
 [16] H. Murata, K. Fujiyoshi, Y. Kajitani: "VLSI module

placement based on Rectangle-Packing by the Sequence-Pair,” IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems, Vol 15, No. 12 pp1518-1524, 1996

[17] W. Shen, T. Yoshimura: “Solving the rectangular packing problem by a Combined Order-based GA/SA based on sequence-pair,” DA シンポジウム 2005 論文集, IPSJ Symposium Series Vol. 2005, No.9 pp109-113

[18] K. Eguchi, J. Suzuki, S. Yamane, K. Ohshima: “An Application of Genetic Algorithms to Floorplanning of VLSI,” Proceeding of RSCTC '98 (International Conference on Rough Set and Current Trend in Computing), LNAI 1424 Springer, pp263-270, June 1998

[19] K. Eguchi, O. Yamashiro, H. Kawamoto, N. Tsuji, S. Yamane, K. Ohshima: “Application of Fuzzy Inference and Genetic Algorithms to VLSI Floorplanning Design,” Proc. of IEEE International Conference on Industrial Electronics, Control and Instrumentation

(IECON-2000) IT2-FNN-3 pp184 – 188

[20] 上田隆之,江口一彦,川本洋,山城治,大嶋健司:“ファジイ推論と遺伝的アルゴリズムを応用したVLSIフロアプラン自動化の試み”DA シンポジウム 2003 論文集, IPSJ Symposium Series Vol.2003 No.11 pp97-102

[21]C.J.Alpert: “The ispd98 circuit benchmark suite,” <http://vlsicad.ucsd.edu/UCLAWeb/cheese/errata.html>

[22] Kirkpatrick, S., Gelett Jr. C. D, Vecchi, M.P: “Optimization by Simulated Annealing,” Science, 220, 671-680, 1983.

[23] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller,A., Teller,E: “Equation of State Calculation by Fast Computing Machines,” Journal of Chemical Physics, 1953.

[24] Saurabh N. Adya, Igor L. Markov: “Consistent Placement of Macro-Blocks Using Floorplanning and Standard-Cell Placement,” University of Michigan, EECS Department, Ann Arbor, MI 48109-212

(受理 平成19年3月19日)