

論文 / 著書情報
Article / Book Information

論題(和文)	Q-sequenceとSA法を用いて高品質な配置を高速に得るためのパラメータ設定に関する一考察
Title(English)	Parameter Setting in Simulated Annealing using Q-sequence for Good Layouts in Short Time
著者(和文)	壺井雅史, 坂主圭史, 高橋篤司
Authors(English)	Masashi Tsuboi, Keishi Sakanushi, Atsushi Takahashi
出典(和文)	第15回 回路とシステム(軽井沢)ワークショップ 論文集, Vol. , No. , pp. 125-130
Citation(English)	Proc. the 15th Workshop on Circuits and Systems in Karuizawa, Vol. , No. , pp. 125-130
発行日 / Pub. date	2002, 4
URL	http://search.ieice.org/
権利情報 / Copyright	本著作物の著作権は電子情報通信学会に帰属します。 Copyright (c) 2002 Institute of Electronics, Information and Communication Engineers.

Q-sequence と SA 法を用いて高品質な配置を 高速に得るためのパラメータ設定に関する一考察

壺井 雅史 坂主 圭史 高橋 篤司

東京工業大学 集積システム専攻

東京都目黒区大岡山 2-12-1

TEL:+81-3-5734-3572 FAX:+81-3-5734-2902

Email:{tsuboi, keishi, atushi}@lab.ss.titech.ac.jp

Parameter Setting in Simulated Annealing using Q-sequence for Good Layouts in Short Time

Masashi Tsuboi Keishi Sakanushi Atsushi Takahashi

Department of Communications and Integrated Systems

Tokyo Institute of Technology

2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, Japan

1 はじめに

VLSI レイアウトにおける最初の工程はフロアプランであり、その後、モジュールの形状や詳細位置を決定する配置工程およびネットの配線をする配線工程が続く。フロアプラン工程では、モジュールの概略位置や配線の概略経路をもとに、配線長やタイミングなどの VLSI の性能を見積る。したがって、フロアプラン工程では、高品位な概略配置を得ることが求められている。また、フロアプラン工程での見積り結果が悪ければ、上位工程へ後戻りして設計を一部修正し、修正を反映した新しいフロアプランを得る。このためフロアプラン工程においては、高速性も要求される。

現在、モジュールの配置手法として、Simulated Annealing(SA) 法などの確率的探索手法を用いて最適なレイアウトを求める手法が盛んに研究されており、Sequence-pair[2], Bounded Sliceline Grid(BSG)[4], O-tree[6] などが提案されている。Sequence-pair, BSG は任意のパッキングを表現できるが、その示唆する制約に基づいた n モジュールの最密（左下詰め）パッキングを得るには Sequence-pair で $O(n \log(\log n))$ 時間[11], BSG で $O(n^2)$ 時間[4] かかる。O-tree はどんな左下詰めパッキングでも表現でき、 $O(n)$ 時間[6] にてその示唆する制約に基づいた左下詰めパッキングを得ることができるが、方形モジュール間の相対関係を表現できないため、配線の概略経路を考慮することが困

難である。

方形領域を垂直／水平線分にて分割して作る方形分割の表現方法として、近年、Quarter-state sequence (Q-sequence)[8] が提案された。これはどんな n 個の方形からなる方形分割でも表現でき、 $O(n)$ 時間にてその示唆する制約を満たすパッキングを得ることができる。また、[8] で提案された局所変更操作を繰り返すことで、任意の Q-sequence から任意の Q-sequence へ Q-sequence のみを経由して変換可能である。しかし、Q-sequence は方形分割の表現方法であるので、 n 方形からなる方形分割を表現する Q-sequence で表現できない n モジュールのパッキングが存在する。しかしながら、モジュールを割り当てない空き部屋 [3] を高々 $n-3$ 個導入することにより、任意のパッキングを表現でき、かつそれぞれのパッキングを $O(n)$ 時間で得ることできる [12]。任意のパッキングを表現するのに必要な空き部屋数に関しては、 $n - \lfloor \sqrt{4n-1} \rfloor$ が、[5, 10] で予想として挙げられ、[13, 14] で証明された。これは必要な空き部屋数の上界を与えているが、準最適なパッキングを高速に得るのに適した空き部屋数を示しているわけではない。また、SA 法のパラメータの設定などによっても、解を求める時間とその品質が大きく変わる。

そこで本研究では、Q-sequence を用い、Simulated Annealing 法にてパッキングを探索することを前提と

し、Q-sequence に導入する空き部屋数や、温度スケジュールなどのパラメータの違いによる解の品質と計算時間の関係を調べ、Q-sequence に対して SA を適用して高品質なパッキングを得るための最適なパラメータ設定に関する指針を与える。

2 準備

2.1 Q-sequence

n 部屋からなる方形分割とは、与えられた方形チップの、水平／垂直線分（セグと呼ぶ）による丁度 n 個の方形（部屋と呼ぶ）への切り分けであり、切り分けた各部屋に方形モジュールを高々 1 つ割り当てることでフロアプランとする。本論では、フロアプランの表現方法として、Quarter-state sequence (Q-sequence) を用いる。以下の定義は [7, 8] にて示されたものであるが、本論の理解を助けるために、ここで再掲しておく。

チップの右下隅を含む部屋を右下部屋と呼ぶ。右下部屋でない部屋 r の右下の角で垂直セグまたは水平セグのどちらか一方が終端している。この、終端しているセグを部屋 r の主セグと呼び、主セグに対して部屋 r の反対側に接している部屋を部屋 r の付部屋と呼ぶ。いま、主セグが垂直（水平）セグの場合、各付部屋を $\mathcal{R}_i$ ($\mathcal{B}_i$) の記号で表す。ここで、 i は対応する付部屋の部屋名である。また、付部屋のうち最も上（左）にある部屋のことを部屋 r の次部屋と呼ぶ。

フロアプランの左上隅の部屋から、次部屋を順次追っていくことで、フロアプランの部屋に固有の順序（阿部順序と呼ぶ）を定義することができる [1, 7, 8]。以下本論では、簡便のために、部屋名を阿部順序に従って $1, 2, \dots, n$ とする。

部屋 r の部屋名のあとに、下（右）から r の付部屋に対応する $\mathcal{R}_i$ 、または $\mathcal{B}_i$ を列挙した記号列を部屋 r の **Quarter-state (Q-state)** と呼び、 $Q(i)$ と書く。ただし、阿部順序で n 番目の部屋（右下隅部屋）の Q-state は、部屋名とする。また、チップの左壁（上壁）に対応する垂直（水平）セグの Q-state は、下（右）から順番に壁に接する部屋に対応する $\mathcal{R}_i$ ($\mathcal{B}_i$) を列挙した記号列と定義する。

Quarter-state sequence (Q-sequence) は左壁の Q-state, 上壁の Q-state, 阿部順序に従う各部屋の Q-state を順に列挙してできる記号列である。

図 1 に例を示す。図中のフロアプランにおける部屋 1 の Q-state は、 $Q(1) = (1\mathcal{R}_3\mathcal{R}_2)$ であり、部屋 2 の Q-state は $Q(2) = (2\mathcal{B}_6\mathcal{B}_4\mathcal{B}_3)$ である。また、このフ

ロアプランの Q-sequence は、
($\mathcal{R}_5\mathcal{R}_1\mathcal{B}_2\mathcal{B}_1$ $1\mathcal{R}_3\mathcal{R}_2$ $2\mathcal{B}_6\mathcal{B}_4\mathcal{B}_3$ $3\mathcal{R}_4$ $4\mathcal{B}_5$ $5\mathcal{R}_6$ 6)
となる。

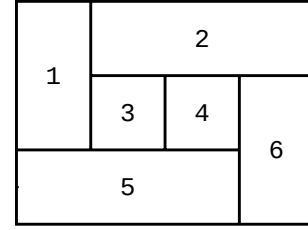
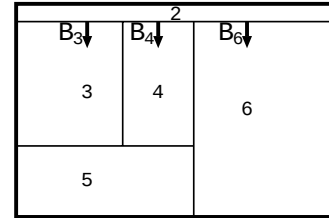


図 1: 部屋数 6 のフロアプラン

Q-sequence からフロアプランへの復元について説明する。まず、名前 n の方形チップを用意する。以下の操作を、 $i = n - 1$ から、1 まで繰り返す。

$\mathcal{R}_x(\mathcal{B}_x) \in Q(i)$ であるような部屋 x を、右（下）方向に押しつぶしながらチップの左（上）壁から部屋 i を挿入する。

図 2 に、部屋 2 を挿入している様子を示す。 $Q(2) = (2\mathcal{B}_6\mathcal{B}_4\mathcal{B}_3)$ であるので、部屋 6, 4, 3 を下方向に押しつぶしながら挿入される。



$\mathcal{R}_5\mathcal{R}_1\mathcal{B}_2\mathcal{B}_1$ $1\mathcal{R}_3\mathcal{R}_2$ $2\mathcal{B}_6\mathcal{B}_4\mathcal{B}_3$ $3\mathcal{R}_4$ $4\mathcal{B}_5$ $5\mathcal{R}_6$ 6

図 2: 部屋 2 の挿入の様子

Q-sequence からそれに対応するフロアプランを得る操作およびフロアプランから対応する Q-sequence を得る操作の計算量は、部屋数を n としたとき $O(n)$ である。

2.2 空き部屋

図 3(a) に例示するような 4 モジュールの左下詰めパッキングに対応する 4 部屋の方角分割は存在しない。そこで、任意の n モジュールのパッキングを n 部屋以上の方角分割で表現するために、図 3(b) に示す ε の様に、モジュールを割り当てない部屋（これを「空き部屋」と呼ぶ [3]）を導入する。

[12]にて、 n 個の部屋からなるフロアプランに、高々 $n-3$ 個の空き部屋を挿入することで、任意のパッキングを表現可能であるということが示された。また、[13, 14] では、任意のパッキングに必要な空き部屋数の上限は $n - \lfloor \sqrt{4n-1} \rfloor$ であるということが示された。

これらの個数の空き部屋をフロアプランに挿入しても、フロアプランの部屋数は $O(n)$ であるから、Q-sequence からパッキングを得る計算量も $O(n)$ 時間のままである。

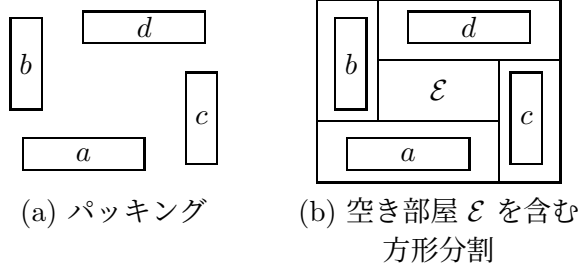


図 3: 方形分割では表現不可能な方形パッキングの例

2.3 Simulated Annealing 法

Simulated Annealing 法とは、解空間の一部のみを探索し、その中で最も良い解を採用する確率的探索手法の一種である。この手法では、解に隣接という概念が定義され、ある解からその解に隣接する別の解への変換操作が Move として定義される。

Annealing の際には、解を Move 操作によって変換し、変換後の評価値がもとの評価値よりも高い場合はその変換操作を採用する。評価が悪化した場合にも、ある確率でその変換を採用することで、局所最適解に陥るのを防ぐ。改悪となる Move 操作を採用するか否かは温度と呼ばれるパラメータにより決定される。改悪量を Δc 、温度を T とするとき、改悪 Move を採用する確率は、 $P = e^{-\frac{\Delta c}{T}}$ で表される。すなわち、温度が高い場合改悪を許す確率も高く、温度が十分に低いと改悪 Move はほとんど採用されない。はじめ十分に高い温度からはじめ、徐々に温度を下げながら Move 操作を繰り返し適用し、解空間を探索することで、よりよい解を得ようというのが Simulated Annealing 法である。

次章では、Q-sequence と SA 法を用い、さまざまな SA のパラメータを適用した実験を行い、SA においてより良いパッキングを得るために有効なパラメータおよび挿入空き部屋数に関する指針を示す。

3 計算機実験

3.1 実験概要

Q-sequence と SA 法を用いてよりよいパッキングをより短い時間で得るのに適した SA のパラメータおよびフロアプランに追加する空き部屋数を調べるために、MCNC ベンチマーク回路の ami49 (モジュール数 49 個) に対して計算機実験を行った。実験には、Athlon 1.4GHz を用いた。

[7, 8] で定義された以下の 4 つの Q-sequence の局所変更操作を SA の Move として採用した。

- 部屋に割り当てられたモジュールの対交換
- モジュールの回転
- 部屋の削除と挿入
- 記号 R および記号 B の移動

SA において各操作の採用確率は等確率とし、初期解はランダムに与えた。

3.2 温度設定

本来、SA 法では、各温度で、平衡状態に達した後に、一定の割合で温度を下げるが、本研究では、各温度において一定数の Move を行なうこととする。

SA 法においては、開始温度、終了温度が適切でないと、高密度な解を得られなかったり、計算時間が無駄に必要となったりする可能性がある。そのため、SA で効率的に探索を行なうための温度設定を実験的に調べる。これらの温度スケジュールは、入力回路が変化しても、評価値を正規化しておくことにより、同じ温度スケジュールで効率的に探索を行うことができる。本実験では、評価値をパッキング率 (モジュールの総面積/レイアウトの Bounding Box の面積 [%]) で与え、改悪量をパッキング率の変化で与える。

まず、開始温度を 100 度と 1 度の実験を行う。ここで、終了温度は 0.01 度とし、温度刻みは 0.99 倍とした。空き部屋数 10 個のときの、各温度におけるパッキング率について、10 回の試行における平均を図 4 に示す。この図から、両者は、ほぼ一致する変化を示しており、開始温度が 1 度で十分であるということが確認できる。

また、適切な終了温度を知るために、開始温度を 1 度、終了温度を 0.001 度として探索を行った。この実験の 10 回の試行における温度 0.1 度以下の各温度でのパッキング率の平均を図 5 に示す。これによると、

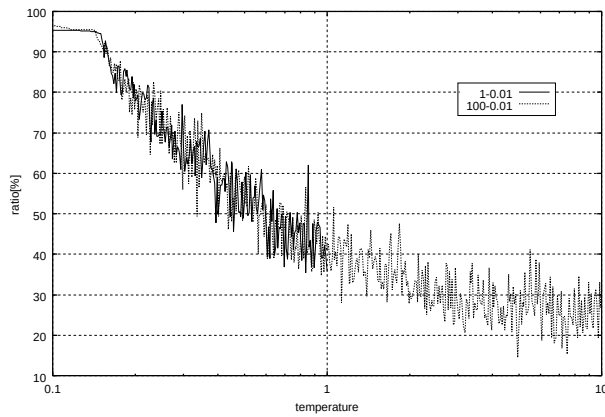


図 4: 2 種類の開始温度におけるパッキング率の変化

若干の上昇が見られるものの、高々0.5%程度であり、ほぼ収束していることが分かる。この結果から、以降の実験では、0.01 度を終了温度とする。

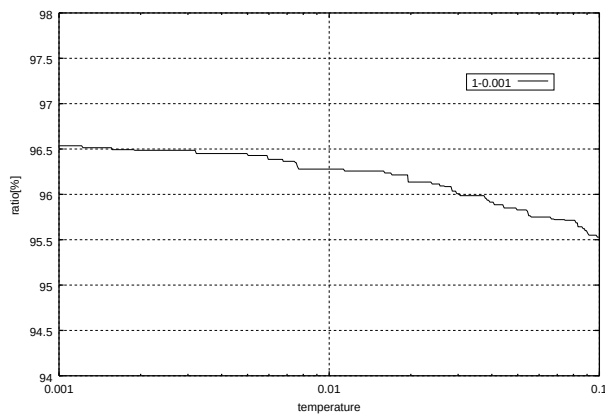


図 5: 1 度から 0.001 度までの温度とパッキング率の関係

以上では、初期温度、終了温度について調べたが、SA 法では、十分に初期温度が高く、終了温度が十分に低くとも、温度刻みの大きすぎると、十分な探索が行えない場合がある。本研究では、内部ループ数を変化させることで温度刻みの設定に代用する。そこで、この代用が妥当なものであることを示すために、Move の総試行回数をほぼ一定に保ったまま、温度刻みと各温度での Move の試行回数（内部ループ数）を変更した実験を行った。実験では、開始温度を 1 度、終了温度を 0.01 度と固定し、温度刻み 0.99、内部ループ数 10000（総ループ数 458 万回）とした実験と、温度刻み 0.98、内部ループ数 20000（総ループ数 454 万回）とした実験とを比べる。実験は 10 回ずつ行い、各温度でのパッキング率の平均を図 6 に示す。

これによると、両者はほぼ同様の曲線を描き、配置

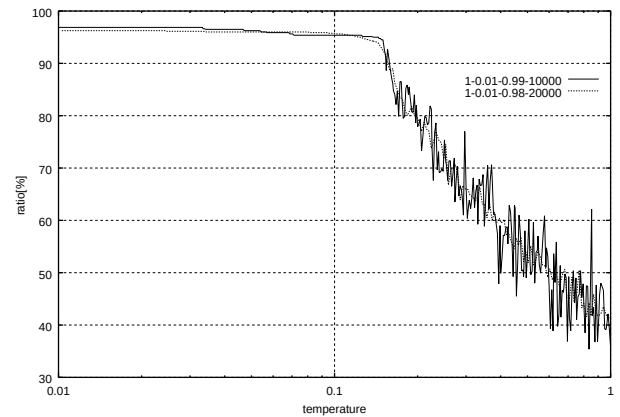


図 6: 温度刻み、内部ループ数を変更したときの温度変化

結果もほぼ同等のものを出力することが確認できる。よって、以降の実験では、温度刻みを 0.99 とし、内部ループ数を変化させることで温度刻みの設定に代用する。

3.3 内部ループ数と挿入空き部屋数

各温度における内部ループ数に関しては、入力回路に依存する問題であり、一般化することは困難であるが、ある入力に関する考察が、他の入力に対するパラメータ設定にも参考となるものと考え、以下の実験を行う。

ami49 に対して、温度スケジュールを、開始温度を 1 度、終了温度を 0.01 度、温度刻み 0.99、外部ループ数（温度更新回数）458 とし、各温度での内部ループ（Move 操作の試行回数）を 1000 回、5000 回、10000 回、20000 回、30000 回と変化させ、挿入空き部屋数を 0 個から 100 個まで 5 個ずつ変化させた実験を 50 回ずつ行った。

実験結果を以下に示す。横軸は挿入空き部屋数、縦軸はパッキング率であり、50 回試行した実験結果の平均値をグラフにしたものが図 7 である。また、最大値をグラフにしたものが図 8 である。

この実験結果より、いずれの内部ループ数についても、空き部屋が 20 個前後までは空き部屋の挿入数が増えるにしたがってパッキング性能が上昇していき、それ以上は空き部屋を挿入してもあまり効果があがらなくなるということが分かる。実験に用いた回路のモジュール数は 49 個であり、任意のパッキングを表現するのに必要な挿入空き部屋数の上限は、 $n - \lfloor \sqrt{4n-1} \rfloor$ の式より、36 個であるが、一般に、上限よりも小さい空き部屋数で十分であると予想され、実験においても、

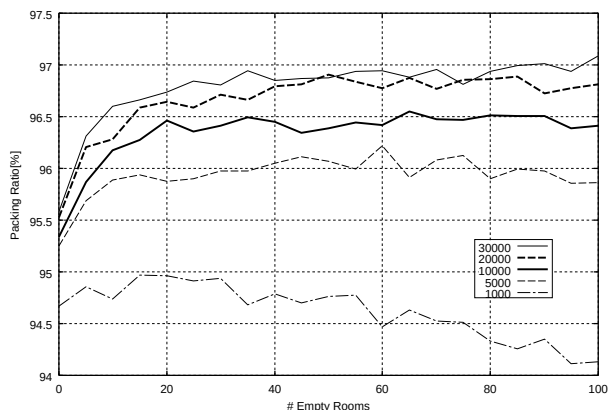


図 7: 内部ループ数を変化させたときの挿入空き部屋数とパッキング結果の関係（平均値）

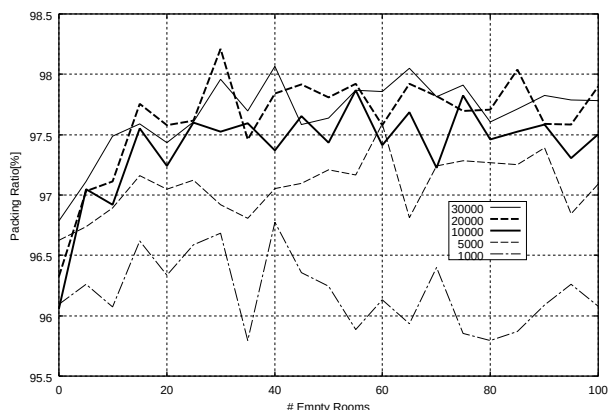


図 8: 内部ループ数を変化させたときの挿入空き部屋数とパッキング結果の関係（最大値）

20 個前後の空き部屋で効果があがらなくなっている。

内部ループ 1000 回の実験では、挿入空き部屋数が 20 個を越えたあたりから、パッキング性能はむしろ悪化してしまうということが分かる。これは、内部ループが 1000 回程度では、空き部屋数を増やす事による解空間の増大によって、各温度での探索が十分に行えなくなってしまうためであると考えられ、Move の試行回数が十分でないということが分かる。

また、基本的には内部ループを増やすにしたがって、パッキング性能は上がっていくが、50 回の試行に対する最大値を見ると、内部ループ 20000 回と 30000 回のいずれの場合でもほぼ同等の性能を示しており、内部ループが 20000 以上になっても、それ以上の効果はそれほど上がらないことが分かる。

これらの結果より、空き部屋の挿入および内部ループ数の増加により、パッキング性能は向上していくものの、両者とも計算時間を増大させ、さらにそれらパ

ラメータの増加による性能の向上にも限界があることが分かる。そこで両者に関して、時間の増加と性能の向上の関係を調べる。図 7 に示した実験の計算時間と解の性能の平均との関係を図 9 に示す。ただし、内部ループ数 1000 の実験結果に関しては、明らかに性能が劣っているため省略した。

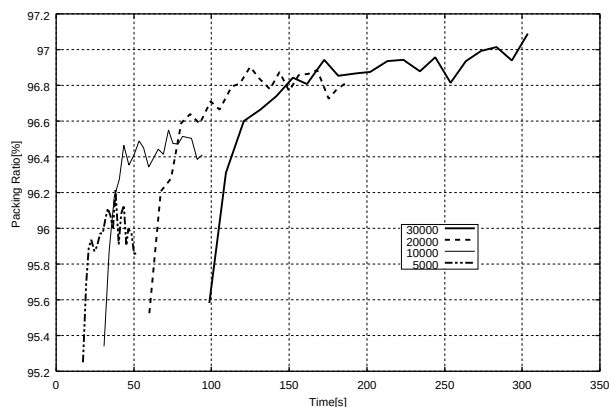


図 9: 計算時間とパッキング性能の関係

図より、指定された時間でより高品質な配置を得るためには、空き部屋を 20 個程度に設定したうえで、総ループ数なるべく多く設定するべきであることがわかる。

これらの空き部屋数、内部ループに関する考察は、ami49 に入力进行限定してのものである。しかし、その他の入力に関しても、同様の傾向が見られる。以下に、ami33（モジュール数 33）を入力として実験を行った結果を示す。空き部屋を 0 個から 45 個まで 5 個刻みで、内部ループを 1000, 10000, 20000 と変化させた実験を 10 回ずつ行い、その平均値を示した。

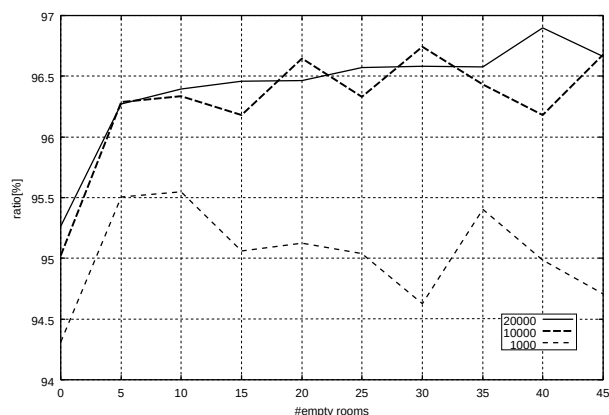


図 10: ami33 に対する実験結果

この結果より、入力回路を変えても、ami49 で行ったのと同様の傾向が見られることが確認できる。以上

より、効率的にパッキングを探索するには、空き部屋をある程度挿入した上で、入力回路の大きさに合わせた内部ループ数に設定するべきであるということが分かる。

4 まとめと今後の課題

本論では、Q-sequence と Simulated Annealing 法を用いてパッキングを探索する際、評価をパッキング率とすれば、開始温度を 1 度、終了温度を 0.01 度、温度刻み 0.99 とし、部屋数 50 個程度の回路であれば挿入空き部屋数を 20 程度、内部ループ数を 20000 程度に設定することでより短時間でよりよいパッキングを得ることができるということを実験的に示し、効率的にパッキングを探索するための SA のパラメータおよび挿入空き部屋数の決定に関する指針を与えた。

また、今後の課題としては、配線面積など、回路設計者の要求するその他のさまざまな条件を考慮したうえでパラメータ設定の調査などがあげられる。

謝辞

本研究は、テレコム先端技術研究支援センター SCAT 研究奨励金の援助を受けている。また本研究は、CAD21 プロジェクトの一部である。

参考文献

- [1] 安部 道雄, “正方形の内部を総て異なる有限個の正方形で隙間なく一重に覆う問題に就いて,” 日本数学会誌, Vol. 4, No. 4, pp. 359–366, 1930 年 12 月.
- [2] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, “VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair,” IEEE Transactions on Computer Aided Design, Vol.15, No.12, pp.1518–1524, Dec., 1996.
- [3] H. Murata, K. Fujiyoshi, T. Watanabe, and Y. Kajitani, “A Mapping from Sequence-Pair to Rectangular Dissection,” Proceedings of Asia and South Pacific Design Automation Conference '97, pp.625–633, Jan., 1997.
- [4] S. Nakatake, H. Murata, K. Fujiyoshi, and Y. Kajitani, “Module Packing Based on the BSG-Structure and IC Layout Applications,” IEEE Transactions on Computer Aided Design, Vol.17, No.6, pp.519–530, Jun., 1998.
- [5] 児玉 親亮, “Super Sequence-Pair —矩形パッキングの効率的な表現方法—,” 東京農工大学修士論文, 2001 年 1 月.
- [6] P. N. Guo, T. Takahashi, C. K. Cheng, and T. Yoshimura, “Floorplanning Using a Tree Representation,” IEEE Transactions on Computer-Aided Design, Vol.20, No.2, pp.281–289, Feb., 2001.
- [7] K. Sakanushi, K. Midorikawa, and Y. Kajitani, “A General and Fast Floorplaning by Reduct-Seq Representation,” Technical Report of IE-ICE, VLD2000-24, Vol.100, No.120, pp.109–116, 2000.
- [8] K. Sakanushi and Y. Kajitani, “The Quarter-State Sequence (Q-Sequence) to Represent the Floorplan and Applications to Layout Optimization,” Proceedings of IEEE Asia Pacific Conference on Circuits and Systems 2000, pp.829–832, Dec., 2000.
- [9] 高島 康裕, 村田 洋, “フロアプランにおける部屋数最小化問題,” 電子情報通信学会技術研究報告, VLD2001-16, Vol.101, No.46, pp.57–64, 2001 年 5 月 18 日.
- [10] 児玉 親亮, 藤吉 邦洋, “線形時間で配置復元可能な Sequence-Pair,” 電子情報通信学会技術研究報告, VLD2001-17, Vol. 101, No. 46, pp. 65–72, 2001 年 5 月.
- [11] Xiaopling Tang, D.F.Wong “FAST-SP: A Fast Algorithm for Block Placement based on Sequence Pair,” Proceedings of Asia and South Pacific Design Automation Conference 2001, pp.521–526, Jan., 2001.
- [12] M. Tsuboi, C. Kodama, K. Sakanushi, K. Fujiyoshi, A. Takahashi, “Linear Time Decodable Rectangular Dissection to Represent Arbitrary Packing using Q-sequence,” Proceedings of the Workshop on Synthesis And System Integration of MIXed Technologies 2001, pp.272–278, Oct., 2001.
- [13] Y. Takashima and H. Murata, “The Tight Upper Bound of the Empty Rooms in Floorplan,” Proceedings of the Workshop on Synthesis And System Integration of MIXed Technologies 2001, pp.264–pp.271, Oct., 2001.
- [14] C. Zhuang, K. Sakanushi, L. Jin, and Y. Kajitani, “An Enhanced Q-Sequence Augmented with Empty-Room-Insertion and Parenthesis Trees,” Proceedings of Design Automation & Test in Europe 2002, 発表予定, March., 2002.