設計自動化　64－5
(1992.10.22)

ブロック間の相対位置制約の操作に基づくビルディングブロックフロアプラン設計手法

山谷　勝巳　　若林　真一　　小出　哲士　　吉田　典可

広島大学工学部
東広島市鏡山１丁目４－１

あらまし　　ビルディングブロック方式ＶＬＳＩチップのフロアプラン設計において重なりのないフロアプランを得るために重なりのあるブロック間にブロックの相対位置に関する制約を挿入していく手法や重なりのないフロアプランから制約を除去していく手法が知られている．本稿ではこれらの手法を拡張し，全てのブロック対に対して水平および垂直方向の位置関係が類推できるような水平および垂直方向制約グラフが与えられたとき，制約の除去・挿入と共にブロックの変形を行いながら重なりのないフロアプランを求める手法を提案する．また，元の問題の部分問題を定義し，この問題に対して遺伝的アルゴリズムに基づく手法を提案する．

和文キーワード　　フロアプラン設計，制約グラフ，フロアプラン最適化問題，トポロジ制約，遺伝的アルゴリズム

A Floorplanning Method for Building Block Layout Based on the Manipulation of Topological Constraints

Katsumi YAMATANI　Shin'ichi WAKABAYASHI　Tetsushi KOIDE　Noriyoshi YOSHIDA

Faculty of Engineering, Hiroshima University
4-1, Kagamiyama 1 chome, Higashi-Hiroshima, 724 JAPAN

Abstract　　On the building block floorplan design of VLSI circuits, two different types of methods to obtain a floorplan with no overlaps have been known. One method added extra constraints between blocks with overlaps. The other removed redundant constraints between blocks with no overlaps. In this paper, we extend these methods. That is, given a strongly complete constraint set of a set of blocks such that horizontal or vertical separation can be inferred from the set given for each pair of blocks, we present several methods with the manipulation of constraints and block reshaping to construct a floorplan with optimal area. We also formalize a subproblem of the original problem, and describe the method based on the genetic algorithm to this problem.

英文 key words　　floorplan design, constraint graph, floorplan optimization problem, topological constraint, genetic alogrithm

1．はじめに

ビルディングブロック方式レイアウト設計は，ＶＬＳＩレイアウト設計の中で最も設計工数および設計期間を必要とする設計手法である．そこでレイアウト設計において重要なステップであるフロアプラン設計に関して様々な研究が精力的になされている．フロアプラン設計問題は非常に複雑な組合わせ最適化問題となるため，基本的なフロアプラン設計手法としてこれを概略配置と詳細配置に分けて行う手法が提案されている[1],[2]．概略配置ではスタンダードセル方式フロアプラン設計で適用されていた最小カット法[3]，重心緩和法（ＦＤＲ）[4]およびシミュレーティッドアニーリング[5]などと同様，総配線長を指標として配置を行う．ビルディングブロック方式フロアプラン設計にＦＤＲ等の手法をそのまま適用するとブロックの形状および面積が不均一なためにブロック間に重なりを生じる一方，総配線長の評価に基づいてフロアプラン設計を行うためにブロック間の相対位置に関する情報が得られる．概略配置で得られた相対位置に関する情報は水平および垂直方向制約グラフやスライシング・ツリーなどで表すことができる[6]．一方，詳細配置ではチップ面積を指標として概略配置で得られた相対位置に関する情報を考慮してブロック間に重なりの存在しない配置を求める．

相対位置に関する情報から得られた制約の集合が与えられたとき，全ての制約を満足しなくてもよい場合がある．例えば，制約の集合において水平および垂直両方向に離されるように制約されたブロック対を考えた場合，それらが必ずしも両方向に離されなければならないというわけではない．特にチップ面積を考えた場合，両方向同時に制約を満たさなければならないとすると解の質が悪くなると予想される．従って，そういう場合は一方の制約を除去することにより解の質が改善されるものと考えられる．

この考察に基づいて文献[7]では，初期フロアプランから得られた制約の集合から冗長な制約を除去していくことによりブロック間に重なりの存在しないフロアプランを得る手法を提案している．一方，初期フロアプランにおいて重なりが存在した場合に重なりのあるブロック間に制約を挿入していく手法が文献[8]で提案されている．しかしながら，相対位置を考慮して制約の除去と挿入を同時に扱いブロックの変形を行いながら重なりの存在しないフロアプランを得る手法は知られていない．

そこで，本研究では全てのブロック対に対して水平あるいは垂直方向の位置関係が類推できるような水平および垂直方向制約グラフが与えられたとき，制約の除去および挿入に加えてブロックの変形を行うことにより重なりのないフロアプランを求める手法について考察を行った．本稿では，擬２次元コンパクション手法の１つであるシフトコンパクション[9]に基づくアルゴリズムASCRIP，生物進化の概念を組合わせ最適化問題に適用した遺伝的アルゴリズムに基づくアルゴリズムGBA，および仮想的な制約の挿入による広範囲な解空間の探索を指向した制約の仮挿入に基づくアルゴリズムASCIの各アルゴリズムを示し，計算機実験によりそれぞれの手法の有効性を確認する．

以下，２．では文献[7]に基づく用語の定義等を与えた後，３．ではアルゴリズムASCRIPを，４．ではアルゴリズムGBAを，５．ではアルゴリズムASCIをそれぞれ説明する．さらに提案アルゴリズムに対する実験結果を６．で示した後，７．では本稿のまとめと今後の課題を示す．

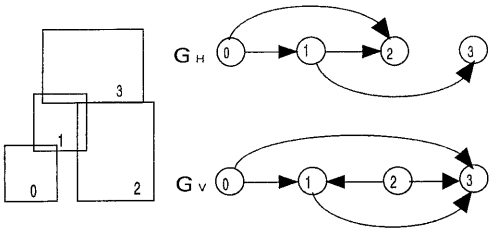
2．準備

この節では，文献[7]に基づいて基本的な用語の定義，問題の定式化，初期フロアプランから入力の制約の集合を求める手法の概要および従来法の概要について述べる．

2．1　用語の定義

・*制約の集合（G_H，G_V）*

ブロックの集合に対する制約の集合は２つの有向非巡回的グラフ（G_H，G_V）で与えられる．ここで，G_Hを水平方向制約グラフ，G_Vを垂直方向制約グラフという．G_HとG_Vの節点はブロックに対応している．有向枝（b_i，b_j）$\in E$（G_H）はb_iがb_jの左に配置されなければならないことを，有向枝（b_i，b_j）$\in E$（G_V）はb_iがb_jの下に配置されなければ



（ａ）初期フロアプラン（ｂ）対応する制約の集合

図１．初期フロアプランと対応する制約の集合の例

ならないことを表す．図1に初期フロアプランと対応する制約の集合の例を示す．

・*強完全性(strong completeness)*

全てのブロック対b_i，b_jに対してG_H，G_V，あるいはその両方に有向枝（b_i，b_j）あるいは（b_j，b_i）が存在すれば，その制約の集合（G_H，G_V）は強完全であるという．

・*正当なフロアプラン (legal floorplan)*

フロアプランに重なりが存在せず，ブロックの形状が縦横比の制約を満たすときこのフロアプランは正当であるという．

・*強関連(strong respect)*

全てのブロック対b_i，b_jに対して，これらがG_HかG_Vのどちらかの制約に従って位置関係が決定されるならば，ブロックの集合Bに対する正当なフロアプランは強完全な制約の集合（G_H，G_V）に強く関連しているという．

・*強冗長性(strong redundancy)*

$(G_H - \{e\}) \cup G_V = G_H \cup G_V$ならば$G_H$の有向枝eは強く冗長であるという．$G_V$の有向枝eの強冗長性も同様に定義される．

2．2　問題の定式化

本稿で考察するフロアプラン設計問題の定義を以下に示す．ここで，ハードブロックとは形状が予め与えられたブロック，ソフトブロックとは形状は与えられず，面積と許容縦横比のみが与えられたブロックを表す．

【問題SCRIP】

入力：1．ソフトブロックの集合$\mathbb{B}_s$およびハードブロックの集合$\mathbb{B}_h$

2．強完全な制約の集合（G_H，G_V）

出力：目的関数を最小とし，制約を満足するブロックの形状および絶対位置

目的関数：全ブロックを囲む最小矩形の面積

制約：1．ソフトブロックは許容される縦横比の範囲内の形状でなければならない．

2．出力されるフロアプランは入力の制約の集合（G_H，G_V）に強く関連していなければならない．

3．ブロック間に重なりが存在しない．

2．3　入力の制約の集合の導出方法

初期フロアプランから問題SCRIPの入力である強完全な制約の集合を求めるには以下のように行う．すなわち，全てのブロック対に対し，水平あるいは垂直方向にしか離れていないブロック対に対してはそれぞれの方向にのみ制約を設定する．また，水平，垂直両方向に離れているブロック対および重なっているブロック対に対してはブロックの中心座標の位置関係に応じて制約を設定する．

2．4　従来法の概要

文献[7]の従来法は制約の除去とブロックの変形操作を基本とした逐次改良法に基づいている．従来法の概要を以下に示す．

【アルゴリズムVT】

チップ面積が改善されなくなるまで以下のフェーズを繰り返せ．

p1：最長経路上に冗長な制約がなくなるまで冗長な制約を除去せよ．

p2：最長経路上のブロックを最長経路長が短縮されるように変形せよ．　□

3．アルゴリズムASCRIP

3．1　制約の再挿入を行うフェーズ

従来法は1次元方向の制約の除去しか考えておらず一度除去した冗長な制約は二度と用いられることがなかった．しかし場合によっては，一度除去した制約を再挿入してもう一方の制約を除去したほうがチップ面積が減少することがあるものと考えられる．そこで，擬2次元コンパクション手法の1つであるシフトコンパクション[9]の概念を適用してチップ面積の減少を図ることを考える．

文献[7]で提案された手法のフェーズ1が終了した時点では制約の集合（G_H，G_V）の最長経路P_H，P_V上には冗長な制約が存在しなくなっている．つまり，従来法では制約の除去に関してはこれ以上は操作が行えなくなってしまっている．そこで，冗長な制約の存在しなくなった最長経路から入力時に冗長であった制約を除去し，他方の制約グラフに制約の再挿入を行えばチップ面積はさらに減少する可能性がある．このような考えに基づくアルゴリズムSSを以下に示す．

【アルゴリズムSS】

S1：P_HかP_Vのうちより長いほうを選ぶ．

S2：選ばれた最長経路上の制約を1つ取り出して現在の最長経路から除去し他方の制約グラフに再挿入できるかどうかを検証する．

S3：もし，再挿入が可能ならば以下を行う．
S3.1：実際に制約の除去および再挿入を行う．
S3.2：トポロジカルソートを行い新しい配置を求める．
S3.3：S1へ．
S4：P_HかP_V上の全ての制約を検証するまでS2，S3を繰り返す．□

次に，S2で行う検証の対象となる再挿入可能条件を示す．
条件1：その制約を再挿入しても現在の最長経路が長くならない．
条件2：再挿入する制約は入力の制約の集合に存在する．

この２つの条件を満たす制約の除去および再挿入を行うことでチップ面積が減少する可能性がある．また，最長経路上に除去，再挿入を行える制約が複数存在した場合は面積減少率の最も高い制約を選択するものとする．

３．２　全体のアルゴリズム

アルゴリズムSSをフェーズ２とし，従来法のフェーズ１とフェーズ２の間に行うアルゴリズムをアルゴリズムASCRIPとし，その概要を以下に示す．

【アルゴリズムASCRIP】

チップ面積が改善されなくなるまで以下のフェーズを繰り返せ．

ｐ１：最長経路上に冗長な制約がなくなるまで冗長な制約を除去せよ．

ｐ２：最長経路上にシフトできる制約がなくなるまでシフトを試行する（アルゴリズムSS）．

ｐ３：最長経路上のブロックを最長経路長が短縮されるように変形せよ．□

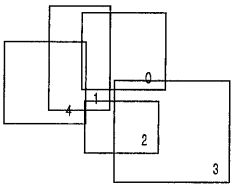
この手法の適用例として図２（ａ）のような初期配置が与えられたとき，図２（ｂ）の制約の集合に従来法を適用すると図２（ｃ）のような配置結果を，本手法を適用すると図２（ｄ）のような配置結果をそれぞれ出力し，これらから本手法の有効性を確認できる．

４．問題ARERとアルゴリズムGBA

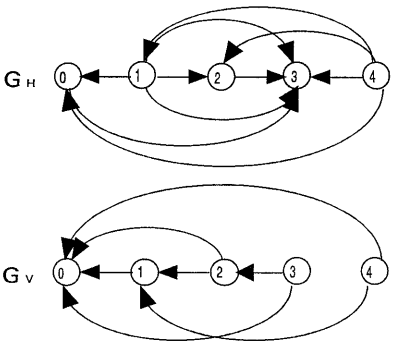
４．１　問題ARER

従来法およびアルゴリズムASCRIPでは除去する冗長な制約をできるだけ少なくしたうえでチップ面積の最小化を行ってきたが，逐次改良法に基づいているため除去される制約の系列が一通りしか考慮できず，しかも除去されなかった冗長な制約のためにブロックの変形ができないという不都合が生じることがあった．そこで，問題SCRIPに以下の制約条件を加えた問題を問題ARERとし，この問題に対して遺伝的アルゴリズムに基づいたアルゴリズムGBAの開発を行った．この問題は元の問題がNP-困難であることからNP-困難であるものと予想される[10]．

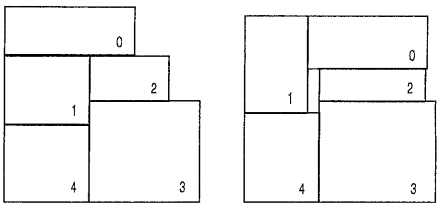
制約：４．冗長な制約については必ずどちらか一方を除去する．



（ａ）初期フロアプラン



（ｂ）対応する制約の集合



（ｃ）従来法　　（ｄ）ASCRIP

図２．従来法とアルゴリズムASCRIPによる配置

4．2　遺伝的アルゴリズムの概要

遺伝的アルゴリズム（ＧＡ）は組合わせ最適化問題の発見的解法の１つとして知られている[11],[12]．ＧＡは生物進化の計算機シミュレーション用に開発された手法を最適化問題に応用したアルゴリズムである．ＧＡは人口と呼ばれる解の集合を保持し，世代交代を繰り返しながら集合内での適者生存によって集合としても改善が進むという進化の性質を利用して，最適化問題の解を得る手法である．この手法には，多数の初期状態から出発して解を探索するため，状態空間中のより広い範囲を効率的に探索できる可能性があるが一般に多大な計算時間を必要とするという特徴がある．

人口は解の集合であり有限個の個体から構成される．各個体は染色体，フィットネス値などからなり，染色体はふつう遺伝子と呼ばれるバイナリビットの文字列で表される．世代と呼ばれる繰り返しを通して現在の人口における各個体がフィットネス関数を用いて評価される．このフィットネス関数の値により，より環境に適していると思われる２個体を親として選択する．そして，これらに交配，突然変異などの操作を施して子孫と呼ばれる新しい個体を生成する．さらに子孫のフィットネス関数の評価を行い，親と子孫の集合からフィットネス関数の値に基づいて新しい世代を形成する個体を選択していく．

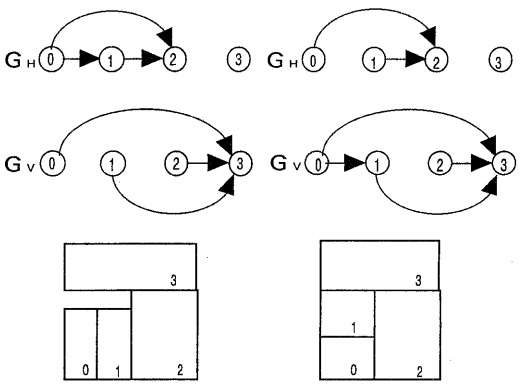
4．3　アルゴリズムGBAの概要

図１に示した初期フロアプランと対応する制約の集合について従来法を適用して重なりのないフロアプランを構成すると図３（ａ）に見られるような結果が得られる．これに対し，最初から図３（ｂ）の制約の集合のように冗長な制約を除去した後でブロックの変形を行うと同図に見られるようにチップ面積に関して最適な重なりのないフロアプランを得ることができる．そこで，遺伝的アルゴリズムに基づいたアルゴリズムGBAの開発を行った．

入力の制約の集合から冗長な制約の集合が分かるのでその要素数だけの長さを持たせた染色体を用意する．遺伝子はそれぞれ冗長な制約の組に対応しており，値０は水平方向制約グラフにのみ制約が存在することを，値１は垂直方向制約グラフにのみ制約が存在することを表すものとする．フィットネス関数の値により重み付けされた確率で親を選択し交配を行うことで子孫を生成し，ある確率で突然変異を起こす．このようにして得られた新しい世代の候補の中からフィットネス関数の値により重み付けされた確率で新しい世代の構成要素を選択する．この手法の概念図を図４に示す．ただし，各個体のフィットネス関数の評価は以下のように行う．

S1：入力時の制約の集合のコピーを行う．
S2：S1で得られたコピーから染色体の情報に従って制約の除去を行う．
S3：トポロジカルソートを行ってチップ面積の計算を行い，初期値を設定する．
S4：S2で得られた制約の集合と同じものを用いて以下のようにブロックの変形を行う．

for(i=0;i<#reshape;i++) {
S4.1：ブロックの現在の形状を保存する．
S4.2：P_HとP_Vのうち長いほうを選択する．
S4.3：選択された最長経路上のブロックの中でP_HとP_Vの両方に属するブロック以外のブロックをすべて変形する．
S4.4：トポロジカルソートを行ってチップ面積の計算を行う．



（ａ）従来法　（ｂ）最適配置

図３．制約の違いによる配置結果の相違

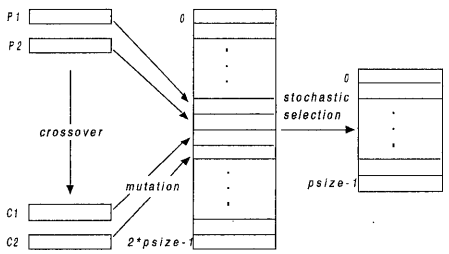


図４．アルゴリズムGBAの概念図

S4.5：チップ面積が改善されていたならばブロックの形状を更新する．そうでなければループを抜ける．

S5：最終的に得られたチップ面積からフィットネス関数の評価を行う．

4．4　フィットネス関数と計算複雑度

遺伝的アルゴリズムを問題ARERに適用するためには，目的関数の最小化をフィットネス関数の最大化に変換しなければならない．また，フィットネス関数の値は非負の値をとらなければならないので文献[11]および予備実験からその変換式を次式のように定義する．

$$F = -S + 3 \times S_{opt} \qquad (1)$$

ただし，

F：フィットネス関数

S：チップ面積

S_{opt}：ブロックの面積の総和

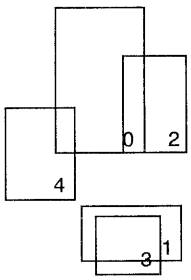
この手法では制約の除去，挿入といった操作は陽には行っておらず単に解空間の広範囲な探索の実現のみを主な目的としている．しかし，1つの染色体に対し変形を行う回数をrとした場合，O(r(n+m))の手間を必要とするので，染色体数をc，世代数をgとしたとき全体でO(cgr(n+m))の実行時間を必要とするため，実用上，計算時間が問題点となる可能性がある．

5．アルゴリズムASCI

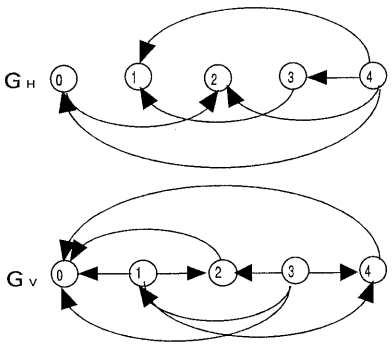
5．1　動機および言葉の定義

先に述べたように従来法も3．で提案したアルゴリズムASCRIPも逐次改良法に基づいているために除去する制約の系列が一通りしか考慮できず，しかも除去されなかった制約のためにブロックの変形ができないという不都合が生じることがあり，そのために質の悪い局所解に陥ってしまう場合があった．また，4．で提案したアルゴリズムGBAにおいては冗長な制約を全て除去した後でブロックの変形を行っているためにブロックを変形しながら制約の操作を行うという逐次改良的な良さが失われてしまっている．そこで，逐次改良法の利点を生かしながらより広い解空間の探索を行うことにより質の良い解を求めると予想される手法の開発を行った．

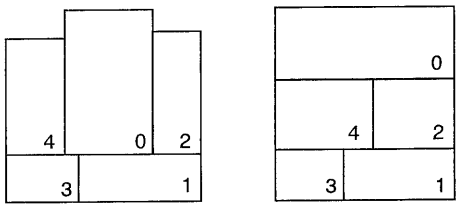
これまでに本稿で提案した手法では入力の制約集合はアルゴリズムの実行中，常に同一としていた．そこで入力の制約集合に存在していない制約を予め*仮挿入(tentative insertion)*しておき，その後で従来法を用いることで従来法よりもチップ面積が減少する場合が考えられる．すなわち，除去される制約の系列が変わりブロックの変形とより影響し合えば従来法に比べて質の良い解を求める可能性があるものと考えられる．例えば，図5（a）の初期フロアプランから得られた図5（b）の制約の集合に対して従来法を適用すると図5（c）に見られるような配置結果となるが，図5（b）の水平方向制約グラフに制約2→3を仮挿入した後で従来法を適用すると図5（d）に見られるような配置結果を得る．そこで，この考察および予備実験から制約の仮挿入を行い問題SCRIPを解くアルゴリズムASCIを開発した．



（a）初期フロアプラン



（b）対応する制約の集合



（c）従来法　　（d）ASCI

図5．従来法とアルゴリズムASCIによる配置

言葉の定義として，入力の制約の集合に仮挿入する制約をソフトな*制約(soft constraint)*，入力の制約の集合に最初から存在する冗長でない制約をハードな*制約(hard constraint)*と呼ぶことにする．この２種類の制約で仮想上の冗長な制約を作り上げる．

５．２　アルゴリズムの概要

ソフトな制約を入力の制約の集合に仮挿入した場合，有向巡回路を生成しこのために制約の集合そのものが不当なものとなることがある．従って，仮挿入を行うたびに有向巡回路が存在するかどうかを検証し，存在する場合にはこの後の一連の操作は行わないものとしなければならない．

また，最終結果が入力の制約の集合に関連していなければならないため，ソフトな制約は除去してもよいがハードな制約はたとえ除去の対象になったときでも除去してはならないものとする．ソフトな制約が残存したまま解が収束してしまった場合にはさらにチップ面積を減少させるためにこれを除去してから従来法を用いることにより最終結果を得るものとする．

【アルゴリズム ASCI】

Ｓ１：従来法を用いて初期解を求める；
Ｓ２：for(i=0;i<(2×($_{n}C_{2}$−(冗長な制約の数)));i++) {
　　ソフトな制約 s_i を入力の制約の集合に仮挿入する；
　　サイクルの存在を検証；
　　if(サイクルが存在)　continue；
　　else {
　　　従来法；
　　　if(ソフトな制約が残存) {
　　　　ソフトな制約を除去；
　　　　従来法；
　　　}
　　　if(暫定解＞生成解)　暫定解の更新；
　　}
　}

このアルゴリズムは従来法に比べて明らかに実行時間は増加するが，解の質は良くなるものと予想される．

６．実験結果

提案アルゴリズムをサンマイクロシステムズ社SPARCstation 2上で実現し，制約の挿入を考慮していない従来法との比較実験を行った．実験は人手により作成したデータおよび乱数により発生させたデータに対して行った．これらのデータをそれぞれ表１，表２に示す．ブロックは全てソフトブロックとし，許容される縦横比の範囲を[1/3,3]

表１．人手により作成したデータ

ﾃﾞｰﾀNo.	ﾌﾞﾛｯｸ数	水平方向の制約数	垂直方向の制約数
1	6	8	11
2	11	24	46
3	21	130	179

表２．乱数により発生させたデータ

ﾃﾞｰﾀNo.	ﾌﾞﾛｯｸ数	水平方向の制約数	垂直方向の制約数
4	10	40	29
5	15	81	73
6	20	142	142

表３．表１のデータに対する実験結果（チップ面積）

ﾃﾞｰﾀNo.	従来法	ASCRIP	GBA	ASCI
1	39.99	38.44	38.15	38.42
2	30.31	29.70	28.14	29.03
3	54.54	53.39	52.31	50.66

表４．表１のデータに対する実験結果（CPU時間）

ﾃﾞｰﾀNo.	従来法	ASCRIP	GBA	ASCI
1	0.1	0.2	104.6	2.2
2	0.4	0.4	213.2	22.3
3	1.6	2.4	644.6	308.9

表５．表２のデータに対する実験結果（チップ面積）

ﾃﾞｰﾀNo.	従来法	ASCRIP	GBA	ASCI
4	54.78	52.62	51.54	47.62
5	66.48	65.55	63.44	62.28
6	115.04	114.40	121.02	110.55

表６．表２のデータに対する実験結果（CPU時間）

ﾃﾞｰﾀNo.	従来法	ASCRIP	GBA	ASCI
4	0.2	0.3	261.3	8.8
5	0.8	1.0	360.0	76.2
6	1.0	0.9	873.2	190.0

とした．ブロックの変形フェーズ１回の反復回数は2000回とし，最長経路上のブロックは１％の減少率で変形を行うものとする．なお，遺伝的アルゴリズムに基づくアルゴリズムGBAのパラメータは，世代数50，ポピュレーションサイズ20，交配確率0.5，突然変異確率0.2とした．

表１のデータに対する実験結果のうちチップ面積に関する結果を表３に，CPU時間（秒）に関する結果を表４に示す．また，表２のデータに対する実験結果のうちチップ面積に関する結果を表５に，CPU時間（秒）に関する結果を表６に示す．チップ面積に関し，従来法と比較してアルゴリズムASCRIPは平均２．３％，アルゴリズムGBAは平均３．５％，アルゴリズムASCIは平均６．４％程度良い解を求めていることが分かる．

アルゴリズムASCRIPはその性質上従来法より質の悪い解を求めることもある．アルゴリズムGBAでブロック数が増加するにつれて解が悪くなっているのは局所解への早期収束およびパラメータ調整の不備によるものと考えられる．アルゴリズムASCIはブロック数にかかわらず質の良い解を求めているがこれは探索範囲の広さによるものと考えられる．また，ランダムデータによる初期フロアプランではブロックが密集しているうえにブロックの形状がかなり不均一なため質の悪い解を求め易かった．従って，ブロックがチップ全面に分散しているようなデータに対してさらに実験を行う必要があると考えられる．

７．おわりに

本稿ではビルディングブロック方式ＶＬＳＩレイアウト設計におけるフロアプラン設計問題において，ブロック間の相対位置に関する制約が与えられたときに重なりが存在しないようにしながらチップ面積を最小にするフロアプラン設計手法を提案した．アルゴリズムASCRIPは擬２次元コンパクション手法の１つであるシフトコンパクションの概念に基づいており，従来法では除去の対象にならなくなっていた制約も除去することによりチップ面積の減少を図る．アルゴリズムGBAは遺伝的アルゴリズムに基づいており，除去する制約の系列を一通りに限定することなく，また冗長な制約については必ず一方を除去することによりブロックの変形の自由度を高め，チップ面積を減少させる．アルゴリズムASCIは意図的に冗長な制約を作りだすという制約の仮挿入に基づいており，逐次改良法の良さを保ちながら広範囲な解空間の探索を行うことにより質の良い解を求める．計算機実験の結果，いずれの手法も従来法と比較してほとんどのデータに対して改善が見られ提案手法の有効性を確認できた．

いずれの提案手法にもさらに改善の余地があるものと考えられる．特に，制約の操作とブロックの変形をよりインタラクティブに行う手法の開発が望まれる．また，概略配置で得られた総配線長とチップ面積のトレードオフを考慮しつつフロアプランを設計する手法の開発も今後の課題として挙げられる．

本研究の成果の一部は文部省科学研究費補助金奨励研究（Ａ）（課題番号04750345）による．

文献

[1]S-S.Kim et al. : "Global placement of macro cells using self-organization principle", Proc.1991 ISCAS, pp.3122-3125 (1991).

[2]S-G. Choi et al. : "A floorplanning algorithm using rectangular Voronoi diagram and force-directed block shaping", Proc.1991 International Conference on Comput.-Aided Des., pp.56-59 (1991).

[3]A.E.Dunlop et al. : "A procedure for placement of standard-cell VLSI circuits", IEEE Trans. on Comput.-Aided Des. Integrated Circuits & Syst., Vol.CAD-4, No.1, pp.92-98 (1985).

[4]N.R.Quinn et al. : "A force-directed component placement procedure for printed circuit boards", IEEE Trans. on Circuits & Syst., Vol.CAS-26, No.6, pp.377-388 (1979).

[5]C.Sechen et al. : "TimberWolf3.2 : A new standard cell placement and global routing package", Proc. 23rd Design Automation Conf., pp.432-439 (1986).

[6]渡辺，他："ＶＬＳＩの設計Ｉ", 岩波書店 (1985).

[7]G.Vijayan et al. : "A new method for floor planning using topological constraint reduction", IEEE Trans. on Comput.-Aided Des. Integrated Circuits & Syst., Vol.10, No.12, pp.1494-1501 (1991).

[8]S-K.Dong et al. : "Constrained floorplan design for flexible blocks", Proc.1989 International Conference on Comput.-Aided Des., pp.488-491 (1989).

[9]H.Onodera et al. : "Step by step placement strategies for building block layout", Proc.1989 ISCAS, pp.921-926 (1989).

[10]S.Sastry et al. : "The complexity of the two dimensional compaction of VLSI layouts", Proc. Int. Conf. Circuits and Computers, pp.402-406 (1982).

[11]D.E.Goldberg : "Genetic Algorithms in Search , Optimization , and Machine Learning", Addison-Wesley Publishing Company , Inc. (1989).

[12]J.P.Cohoon et al. : "Genetic placement", IEEE Trans. on Comput.-Aided Des., Vol. CAD-6, No.6, pp.956-964 (1987).