

3次元 VLSI におけるニューラルネットワークを用いた 初期配置手法

正木 貴大 瀬尾 賢治 大村 道郎

広島工業大学工学部 〒731-5193 広島市佐伯区三宅 2-1-1

E-mail: {u97b105, u99b066, ohmura}@cc.it-hiroshima.ac.jp

あらまし 近年の集積回路の大規模化に伴い、3次元VLSIに関する研究が注目を集めている。大規模な集積回路を設計するには、レイアウト設計の自動化が不可欠である。また、LSIの設計自動化に伴う 組合せ最適化問題の多くは、NP困難と呼ばれるクラスに属する。このため、問題の規模が大きくなると逐次処理を前提とした列挙法により、最適解を探索することは事実上不可能と考えられる。本研究では、シーケンストリプルで表された3次元VLSIにおけるフロアプランを、ニューラルネットワークを用いて決定する手法を提案する。本稿ではそのアルゴリズムと、比較的小規模なデータについて行った実験結果について述べる。

キーワード ニューラルネットワーク、シーケンストリプル、フロアプラン

Initial Placement Method using Neural Networks in 3-D VLSI

Takahiro MASAKI Kenji SEO and Michiroh OHMURA

Faculty of Engineering, Hiroshima Institute of Technology

2-1-1 Miyake, Saeki-ku, Hiroshima, 731-5193 Japan

E-mail: {u97b105, u99b087, ohmura}@cc.it-hiroshima.ac.jp

Abstract In recent years, the research on 3-D VLSI has been the focus of attention. In order to design large-scale integrated circuits, layout automation is indispensable. Moreover, many of combinatorial optimization problems in the VLSI layout automation are NP-hard. Therefore, it is impossible to find the optimum solution by the enumeration search based on serial processing when the scale of the problem grows. This paper proposes a 3-D VLSI floorplanning method using Sequence-Triple and neural networks. The experimental results of small-scale data are also described.

Keyword Neural Network, Sequence Triple, Floor Plan

1. はじめに

近年、集積回路プロセス技術の分野において、MCM (Multi Chip Module) , FPGA (Field Programmable Gate Array) など、様々な 2 次元集積回路を重ね合わせて 3 次元化する研究が行われるようになった^{[1][2][3]}。将来的に性能面で有利になると考えられる回路素子自体の 3 次元集積化についても、1980 年代後半には、小規模ながらもすでに試作が行われている^{[4][5]}。大規模な集積回路を設計するには、レイアウト設計の自動化が不可欠であるが、3 次元 VLSI のレイアウト設計における初期配置^[6]に関しては、まだほとんど研究されていない。

VLSI の初期配置設計は方形パッキング問題と深い関係を持つ。方形パッキング問題とは、様々な大きさの直方体を、重なりなく、できるだけ小さい面積の方形内に配置する問題である。この問題は複雑な組合せ最適化問題であるため、実用上の大きさの問題に対して厳密解を見出すことはきわめて困難である。3 次元パッキング問題については、配置領域の底面の大きさを固定し、高さの最小化を目標とする構成的手法が提案されている^[7]。近年の計算機性能の向上に伴い、構成的手法に代わり、「多数の解を生成し、その中から探索的に評価を最良にする解を選び出す」という探索的手法が有用な方法として確立しつつある。この手法の鍵は構築する解空間にあり、解空間の性質が探索の効率を左右する。3 次元パッキング問題と同種の困難さをもつ 2 次元パッキング問題においてシーケンスペア^{[8][9]}により高密度な 2 次元配置が得られることを報告しており、シーケンストリプルとはシーケンスペアの概念を 3 次元に拡張したものである。

また、組合せ最適化問題を解く有効な手段として、ニューラルネットワークが挙げられる。ニューラルネットワークでは、最適化問題求解アルゴリズムの研究が盛んに行われており、これを用いることにより問題の高速・高密度化が非常に期待できる。本研究では、シーケンストリプル^[10]で表された 3 次元 VLSI におけるフロアプランを、ニューラルネットワークを用いて決定する手法を提案する。本稿では、そのアルゴリズムと比較的小規模なデータについて行った実験結果について述べる。

2. 準備

今回用いたシーケンストリプルについて説明するには、シーケンスペアについて理解する必要があるため、まず必要な部分について説明し、そのあとシーケンストリプルについて概要を述べる。

2.1. シーケンスペア

シーケンスペアは、矩形集合 M を入力とする 2 次元パッキング問題において、

(1) 順列対 (Γ_+, Γ_-) により矩形間の位置関係を表現

(2) 位置関係を制約とみなして制約グラフを構築

(3) 最長パス計算により矩形の物理座標を決定する。

以下、シーケンスペアにより配置を得るアルゴリズムの詳細を述べる。

矩形集合 M を入力とすると、シーケンスペアは矩形間の相対位置関係を、長さ $|M|$ の矩形名の順列 Γ_+ , Γ_- の対 (Γ_+, Γ_-) により表す。ここで、 $\Gamma_+(i)$ は Γ_+ の中で i 番目の矩形を指し、 $\Gamma_+^{-1}(a)$ は Γ_+ の中で矩形 a が何番目かを指す。 Γ_- についても同様である。

左右関係: $\Gamma_+^{-1}(a) < \Gamma_+^{-1}(b)$ かつ $\Gamma_-^{-1}(a) > \Gamma_-^{-1}(b)$ のとき、矩形 a の右に矩形 b がある、あるいは矩形 b の左に矩形 a があるとする。

上下関係: $\Gamma_+^{-1}(a) < \Gamma_+^{-1}(b)$ かつ $\Gamma_-^{-1}(a) > \Gamma_-^{-1}(b)$ のとき、矩形 a の下に b がある、あるいは矩形 b の上に a があるとする。

Γ_+ , Γ_- は矩形名がちょうど 1 回現れる順列なので、異なる矩形間には上下左右のいずれか 1 つの位置関係が、シーケンスペア (Γ_+, Γ_-) により与えられる。

次に矩形間に与えられた位置関係を制約として、最小面積の方形内に配置するために、水平制約グラフ $G_h(V_h, E_h)$, 垂直制約グラフ $G_v(V_v, E_v)$ を構築する。

$G_h(V_h, E_h)$ は、点重み付き有向無閉路グラフで、各矩形に点に対応してその横幅が点の重みとなり、左右制約関係にある全ての矩形対について左側の矩形に対応する点から右側の矩形に対応する点に有向枝が張られたものに、ソース点とシンク点を付加したものである。 $G_v(V_v, E_v)$ は、点の重みが矩形の高さで枝が上下制約関係により張られる以外は $G_h(V_h, E_h)$ と同じである。そして、よく知られた有向無閉路グラフの最長パスアルゴリズム (時間複雑度 $O(|V| + |E|)$) を用いることにより、矩形数を n として $O(n^2)$ 時間にてこれらの制約の下で最も密なパッキングを容易に求めることができる。図 1 にシーケンスペア (bcade, cdbea) から求められる水平制約グラフを示す。

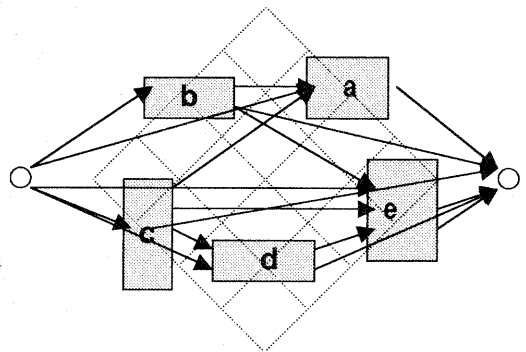


図 1 $(\Gamma_+, \Gamma_-) = (bcade, cdbea)$ の水平制約グラフ

2.2. シーケンストリプル

シーケンストリプルでは、矩形の相対位置関係を矩形名の順列 $(\Gamma_1, \Gamma_2, \Gamma_3)$ の形で表す。ここで、 $\Gamma_1(i)$ は Γ_1 中で第 i 番目の矩形を指し、 $\Gamma_1^{-1}(a)$ は Γ_1 中で矩形 a が何番目かを指す。 Γ_2, Γ_3 についても同様であ

る。そして、相対位置関係を制約とみなして制約グラフを構築し、最長パス計算によりモジュールの物理座標を求め、3次元配置を得る(図2参照)。

ここでは相対位置関係の表現のみを説明する(表1参照)。

- 左右関係: $\Gamma_1^{-1}(a) < \Gamma_1^{-1}(b)$ かつ $\Gamma_2^{-1}(a) > \Gamma_2^{-1}(b)$ かつ $\Gamma_3^{-1}(a) < \Gamma_3^{-1}(b)$, または $\Gamma_1^{-1}(a) > \Gamma_1^{-1}(b)$ かつ $\Gamma_2^{-1}(a) > \Gamma_2^{-1}(b)$ かつ $\Gamma_3^{-1}(a) < \Gamma_3^{-1}(b)$ のとき, aの右にbがあることを意味する。
- 前後関係: $\Gamma_1^{-1}(a) < \Gamma_1^{-1}(b)$ かつ $\Gamma_2^{-1}(a) < \Gamma_2^{-1}(b)$ かつ $\Gamma_3^{-1}(a) < \Gamma_3^{-1}(b)$ のとき, aの後にbがあることを意味する。
- 上下関係: $\Gamma_1^{-1}(a) > \Gamma_1^{-1}(b)$ かつ $\Gamma_2^{-1}(a) < \Gamma_2^{-1}(b)$ かつ $\Gamma_3^{-1}(a) < \Gamma_3^{-1}(b)$ のとき, aの上にbがあることを意味する。

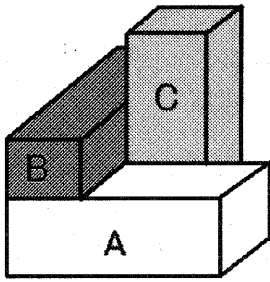


図2 ($\Gamma_1, \Gamma_2, \Gamma_3$) = (bac, acb, abc)に対応する配置

Γ_1	Γ_2	Γ_3
$(\dots a \dots b \dots)$	$(\dots a \dots b \dots)$	$(\dots a \dots b \dots)$
$\rightarrow a$ の後にb	$\rightarrow a$ の後にb	$\rightarrow a$ の後にb
$(\dots a \dots b \dots)$	$(\dots a \dots b \dots)$	$(\dots a \dots b \dots)$
$\rightarrow a$ の後にb	$\rightarrow a$ の後にb	$\rightarrow a$ の後にb
$(\dots a \dots b \dots)$	$(\dots a \dots b \dots)$	$(\dots a \dots b \dots)$
$\rightarrow a$ の後にb	$\rightarrow a$ の後にb	$\rightarrow a$ の後にb
$(\dots a \dots b \dots)$	$(\dots a \dots b \dots)$	$(\dots a \dots b \dots)$
$\rightarrow a$ の後にb	$\rightarrow a$ の後にb	$\rightarrow a$ の後にb
$(\dots a \dots b \dots)$	$(\dots a \dots b \dots)$	$(\dots a \dots b \dots)$
$\rightarrow a$ の後にb	$\rightarrow a$ の後にb	$\rightarrow a$ の後にb
$(\dots a \dots b \dots)$	$(\dots a \dots b \dots)$	$(\dots a \dots b \dots)$
$\rightarrow a$ の後にb	$\rightarrow a$ の後にb	$\rightarrow a$ の後にb

表1 モジュールの相対位置関係の指定

3. ニューラルネットワーク

3.1. 相互結合型ニューラルネットワーク

ニューラルネットワークを構造に注目して分類すると,階層的な構造をもつ階層型ニューラルネットワークと,全てのニューロン間に結合がある相互結合型ニューラルネットワークとに分類できる。

ホップフィールドネットワークは相互結合型ニューラルネットワークの代表的なものである。すべてのユニットは,自分自身をのぞく他のすべてのユニットと結合している。ホップフィールドネットワークはある入力に対して,ネットワークが出力値を出し,それをもとに学習していくという階層型ニューラルネットワークとは異なり,ネットワークのエネルギーが最小になるように状態変化を繰り返すというもので学習をしない。このエネルギー関数を

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n W_{ij} X_i X_j - \sum_{i=1}^n \theta_i X_i$$

と定義する。但し, W_{ij} は素子 X_i, X_j 間の結合加重, θ_i は閾値を表す。ホップフィールドネットワークを使った計算は,次のように行われる。

- STEP1 全ニューロンの出力に, 適当な初期値(0または1)をセットする。
- STEP2 任意のニューロンを一つ選択し, その時点における入力から出力を計算してニューロンの状態を更新する。
- STEP3 未更新のニューロンについて STEP2を繰り返し, 全ニューロンの状態を更新する。
- STEP4 状態変化がなくなるまで STEP2・STEP3を繰り返す。

3.2. ボルツマンマシン

ボルツマンマシンでのニューロンの出力は,入力と0との大小関係で一意的に決まるものではなく,確率的な計算により決定する(図3参照)。こうして,場合によってはエネルギーが増加することもできるようになる。

その結果,極小値に収束した場合でも抜け出せることができ,最小値にたどりつける可能性が大きく増える。この確率的な動作は大変工夫されている。

最初のうちはエネルギーを増加させる方向への動作がかなり許されるようにしてある。

こうしてエネルギー関数上で比較的広い範囲を動きながら,最小値のありそうな場所を探す。

次第にそのようなランダムな動作が制限されてきて,エネルギーの小さい方向へ向かっての動作が大半を占めるようになってくる。

最終的に一番低い部分に落ちつきやすくなる。

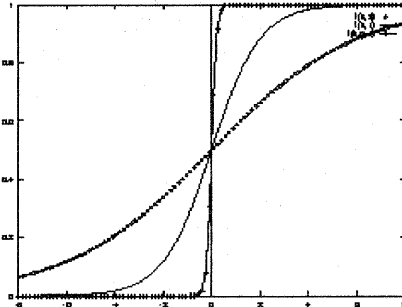


図3 入力和と発火確率

ボルツマンマシンでは,このような動作を焼きなまし(アニーリング)法と呼んでいる。

3.2.1. ボルツマンマシンのアルゴリズム

- STEP1 問題に応じたエネルギー関数を設定し,同時に焼きなましのための温度を下げるスケジュールを決定する。
- STEP2 ネットワークの中から,無作為に1つのニューロンを選ぶ。
- STEP3 他のニューロンから,選ばれたニューロンへの入力和を求める。
- STEP4 入力和と温度によって定まる確率 p に基づき,

ニューロンの出力を決める。

$$p(\text{発火確率}) = \frac{1}{1 + \exp[-(\text{入力和})/(\text{温度})]}$$

図3に示したように温度が高いと確率はなだらかに変化し、温度が低くなると0付近で急に変化するようになる。

STEP5 STEP2~4の動作を繰り返す。但し、温度はSTEP1で定められた手順で下げられる。

4. 問題の定式化

まず問題をネットワークの形で表現する。例としてa, b, cの3つの直方体についてシーケンストリプルで矩形の相対位置関係を(acb, bca, cab)と表す場合、 $\Gamma_1^{-1}(a)=1$ なので、 $a:(1\ 0\ 0)$ と表すことができる。他の矩形についても同様に考えると、次のような行列を構成できる。

$$\begin{matrix} & \Gamma_1 & \Gamma_2 & \Gamma_3 \\ a: & (1\ 0\ 0, & 0\ 0\ 1, & 0\ 1\ 0) \\ b: & (0\ 0\ 1, & 1\ 0\ 0, & 0\ 0\ 1) \\ c: & (0\ 1\ 0, & 0\ 1\ 0, & 1\ 0\ 0) \end{matrix}$$

次に、この問題についての制約条件を考える。

- 1 入力されたモジュールの集合Mに対して、全てのモジュールを囲む体積を最小にする。
- 2 決められた2端子配線の配線長を最小にする。
- 3 Γ に入るモジュールに同じものがないようにする。これらの制約条件があげられる。

まず3の制約条件を定式化する。まず上の行列での Γ_r についての行列 $k=r$ 層、 i 行、 j 列の要素を X_{ijk} と置く。行 i すなわちモジュール i については、ある1つの順番のみが発火し、他は発火しないことから、

$$(X_{i1k} + X_{i2k} + \dots + X_{iNk} - 1)^2$$

(但しNはモジュールの総数)

が最小となるときとして定義できる。それがすべてのモジュールについて成立するので、 i についての総和をとって、

$$\sum_{k=1}^3 \sum_{i=1}^N \left(\sum_{j=1}^N X_{ijk} - 1 \right)^2$$

が最小となればよい。

また、列 j すなわち Γ 内の j 番目のモジュールは1つのみが発火し、他は発火しないことから、

$$(X_{1jk} + X_{2jk} + \dots + X_{Njk} - 1)^2$$

(但しNはモジュールの総数)

を最小とするときである。従って、

$$\sum_{k=1}^3 \sum_{j=1}^N \left(\sum_{i=1}^N X_{ijk} - 1 \right)^2$$

が最小となればよい。

以上をまとめて、制約条件3を表現する目的関数 ϕ_3 は次のようになる。

$$\phi_3 = \frac{A}{2} \sum_{k=1}^3 \sum_{i=1}^N \left(\sum_{j=1}^N X_{ijk} - 1 \right)^2 + \frac{B}{2} \sum_{k=1}^3 \sum_{j=1}^N \left(\sum_{i=1}^N X_{ijk} - 1 \right)^2$$

係数AとBは、ネットワークがうまく収束するように調節するためのものである。

次に、制約条件1について考えてみる。モジュール

i の k 層に対する長さを表す行列 L_{ik} を用いれば、制約条件1を表現する目的関数 ϕ_1 は次のようになる。

$$\phi_1 = \frac{C}{2} \sum_{j=1}^N \sum_{k=1}^3 \sum_{i=1}^N L_{ik} \cdot X_{ijk}$$

$$L_{ik} = \begin{cases} \text{モジュール}i\text{が最長パス上にあれば} \\ k=1\text{のとき} & \text{ソース点からの}x\text{座標距離}w \\ k=2\text{のとき} & \text{"} & \text{y座標距離}n \\ k=3\text{のとき} & \text{"} & \text{z座標距離}d \\ \text{なければ} & 0 \end{cases}$$

配置領域の大きさは $(\sum w + \sum n + \sum d)$ の大小で考える。

最後に、制約条件2について考える。モジュール i と m の k 層に対する距離を表す行列 N_{imk} を用いれば、制約条件2を表現する目的関数 ϕ_2 は次のようになります。

$$\phi_2 = \frac{D}{2} \sum_{k=1}^3 \sum_{i=1}^N \sum_{j=1}^N X_{ijk} \left(\sum_{m=1}^N N_{imk} \cdot X_{mj k} \right) \quad (\text{但し } m \neq i)$$

$$N_{imk} = \begin{cases} \text{2端子ネット}(i, m)\text{があれば} \\ k=1\text{のとき} & x\text{距離} \\ k=2\text{のとき} & y\text{距離} \\ k=3\text{のとき} & z\text{距離} \\ \text{2端子ネット}(i, m)\text{がなければ} \\ & 0 \end{cases}$$

以上より、この問題の目的関数は

$$\phi = \phi_3 + \phi_1 + \phi_2$$

$$\begin{aligned} &= \frac{A}{2} \sum_{k=1}^3 \sum_{i=1}^N \left(\sum_{j=1}^N X_{ijk} - 1 \right)^2 + \frac{B}{2} \sum_{k=1}^3 \sum_{j=1}^N \left(\sum_{i=1}^N X_{ijk} - 1 \right)^2 \\ &+ \frac{C}{2} \sum_{j=1}^N \sum_{k=1}^3 \sum_{i=1}^N L_{ik} \cdot X_{ijk} \\ &+ \frac{D}{2} \sum_{k=1}^3 \sum_{i=1}^N \sum_{j=1}^N X_{ijk} \left(\sum_{m=1}^N N_{imk} \cdot X_{mj k} \right) \end{aligned}$$

と表される。この目的関数を一般的なニューラルネットワークのエネルギー関数と比較し、結合の重みWと閾値hを決定する。

まず、Wを求める。WはXに関する2次の項の係数となっているので、上記のエネルギー関数のXに関する2次の項を集める。この問題の右辺第1項、第2項、第3項をそれぞれ順に $\phi_A, \phi_B, \phi_C, \phi_D$ とし、次のように変形する。

$$\begin{aligned}\phi_A &= \frac{A}{2} \sum_{k=1}^3 \sum_{i=1}^N \left(\sum_{j=1}^N X_{ijk} - 1 \right)^2 \\ &= \frac{A}{2} \sum_{k=1}^3 \sum_{i=1}^N \left(\sum_{j=1}^N X_{ijk} \sum_{m=1}^N X_{imk} - 2 \sum_{j=1}^N X_{ijk} + 1 \right) \\ &= \frac{A}{2} \sum_{k=1}^3 \sum_{i=1}^N \sum_{j=1}^N \sum_{m=1}^N \delta_{im} (1 - \delta_{jn}) X_{ij} X_{mn} - A \sum_{k=1}^3 \sum_{i=1}^N \sum_{j=1}^N X_{ijk} + Const \\ &= \frac{A}{2} \sum_{k=1}^3 \sum_{ij} \sum_{mn} \delta_{im} (1 - \delta_{jn}) X_{ij} X_{mn} - A \sum_{k=1}^3 \sum_{ij} X_{ijk} + Const\end{aligned}$$

この変形の3段階目で Σ が1つ増えたのは、エネルギー関数を考慮するためである。但し、条件として $i=m$ が必要となる。変形の4段階目は単に Σ をまとめて、見やすくしただけである。条件 $j \neq n$ と $i=m$ は $\delta_{im}(1-\delta_{jn})$ という表記で書き換えておく。

但し、 δ はクロネッカーのデルタといって、次のように定義される。

$$\delta_{ij} = \begin{cases} 1 & (i=j) \\ 0 & (i \neq j) \end{cases}$$

なお、変形の途中からConst.とあるのは、ユニットの状態によらない定数項をまとめたものである。

ϕ_B の変形も同様に行える。変形の途中は省略して結果だけを示す。

$$\begin{aligned}\phi_B &= \frac{B}{2} \sum_{k=1}^3 \sum_{j=1}^N \left(\sum_{i=1}^N X_{ijk} - 1 \right)^2 \\ &= \frac{B}{2} \sum_{k=1}^3 \sum_{ij} \sum_{mn} \delta_{jn} (1 - \delta_{im}) X_{ijk} X_{mn} - B \sum_{k=1}^3 \sum_{ij} X_{ijk} + Const\end{aligned}$$

次に、 ϕ_C についてだが、 ϕ_C はXの1次式のみのため変形の必要はない。

最後に ϕ_D の変形を行う。

$$\begin{aligned}\phi_D &= \frac{D}{2} \sum_{k=1}^3 \sum_{i=1}^N \sum_{j=1}^N X_{ijk} \left(\sum_{m=1}^N N_{imk} \cdot X_{mjk} \right) \\ &= \frac{D}{2} \sum_{k=1}^3 \sum_{i=1}^N \sum_{j=1}^N \sum_{m=1}^N \sum_{n=1}^N X_{ijk} X_{mnk} N_{imk} \quad (n=j, m \neq i) \\ &= \frac{D}{2} \sum_{k=1}^3 \sum_{ij} \sum_{mn} \delta_{jn} (1 - \delta_{im}) N_{imk} X_{ijk} X_{mnk}\end{aligned}$$

これよりWを求めるためにXの2次の項を集める。

$$W_{k,ij,nn} = -A\delta_{im}(1-\delta_{jn}) - B\delta_{jn}(1-\delta_{im}) - D\delta_{jn}(1-\delta_{im})N_{imk}$$

さらにXの1次の項を集めると、hを決めることができる。

$$h_{ijk} = A + B - \frac{C}{2} L_{ik}$$

このようにして求めた結合の重みと閾値を用いて、状態変化を行う。

5. 実験結果

ニューラルネットワークとシーケンストリプルを

もとにフロアプランを求めた。

ホップフィールドネットワークを用いた提案手法とシーケンストリプルで表される全ての組み合わせを調べる手法との比較を表2に示す。また、提案手法のモジュール数と時間との関係を図4に示す。

表2 比較結果

直方体数 M	本手法		比較手法	
	体積 V	時間	体積 V	時間
3	126	0.10	96	0.02
4	264	0.26	168	0.62
5	480	0.50	360	1:46.13

Hopfield Network						
M	TIME	V	A	B	C	D
3	0:00.10	126	1	2	0.173	0
4	0:00.26	264	1	2	0.13	0
5	0:00.50	480	1	2	0.1	0
6	0:00.95	792	1	2	0.08	0
7	0:01.71	1680	1	2	0.065	0
8	0:02.66	2601	1	2	0.06	0
9	0:04.18	3584	1	2	0.05	0
10	0:06.14	5060	1	2	0.05	0
11	0:08.84	7800	1	2	0.04	0
12	0:12.34	9968	1	2	0.03	0
13	0:17.25	8640	1	2	0.03	0
14	0:23.46	9408	1	2	0.03	0
15	0:23.46	9384	1	2	0.03	0
16	0:40.51	6496	1	2	0.02	0
17	0:54.00	12852	1	2	0.02	0
18	1:05.94	7280	1	2	0.02	0
19	1:17.45	13583	1	2	0.019	0
20	1:35.88	20163	1	2	0.015	0

図4 モジュール数と時間

相互結合型ニューラルネットワークを用いた解法は、ホップフィールドネットワークでは初期値に完全に依存し、ネットワークがエネルギーの極小値に収束する理由から、最適解を求めるのは困難である。

そこで、エネルギーの最小値を探すことのできるボルツマンマシンを使う。ボルツマンマシンを用いた提案手法とシーケンストリプルで表される全ての組合せを調べる手法との比較を表3に示す。温度 $T(t)$ を下げるスケジュールは $T(t) \geq c/(1+t)$ である。ここで c はスタートの温度である。また、提案手法のモジュール数と時間との関係を図5に示す。

表3 比較結果

直方体数 M	本手法		比較手法	
	体積 V	時間	体積 V	時間
3	96	0.38	96	0.02
4	168	0.57	168	0.62
5	360	0.71	360	1:46.13

Boltzmann Machine						
M	TIME	V	A	B	C	D
3	0:00.38	96	1	2	0.21	0
4	0:00.70	168	0.4	0.4	0.0584	0
5	0:00.71	360	1	1.5	0.115	0
6	0:00.94	840	1	2	0.11	0
7	0:01.18	1596	1	2	0.08	0
8	0:01.48	2464	1	2	0.07	0
9	0:01.84	2600	1	2	0.07	0
10	0:02.07	4320	1.5	1.8	0.069	0
11	0:02.58	4830	1.5	1.8	0.069	0
12	0:03.00	5070	1.5	1.8	0.069	0
13	0:03.56	4800	1.5	1.8	0.069	0
14	0:04.24	5280	1.5	1.8	0.069	0
15	0:04.57	6912	1.5	1.8	0.05	0
16	0:05.29	12420	1.5	1.8	0.05	0
17	0:06.32	12540	1.5	1.8	0.05	0
18	0:07.13	11400	1.5	1.8	0.05	0

図5 モジュール数と時間

ホップフィールドを用いた場合と、ボルツマンマシンを用いた場合では時間はかかるようになるがボルツマンマシンを用いた方がより最適解に近い値が出ることがわかる。

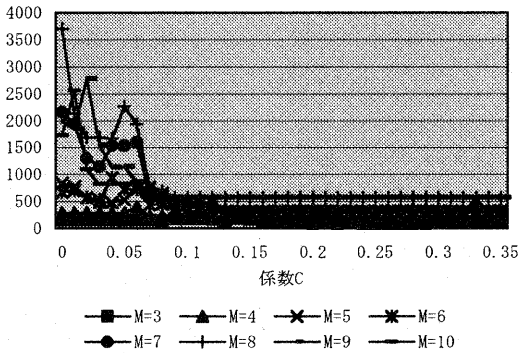


図6 係数Cの増減と配置領域の変化

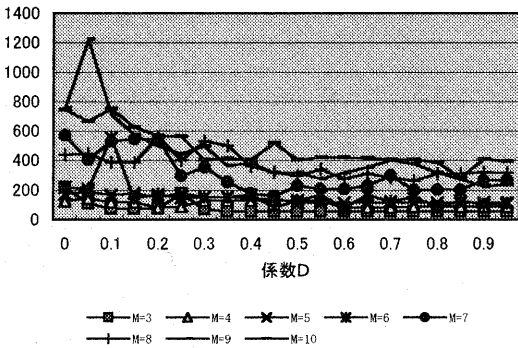


図7 係数Dと配線長の変化

次に目的関数に用いた係数の値によって、どの目的関数を重要視するかを決めることができるということを示す。図6は他の係数を一定にし、係数Cを大きくした場合の配置領域の変化を示した図である。また、図7は他の係数を一定にし、係数Dを大きくした場合の総配線長の変化を示した図である。どちらも係数を大きくした場合、配置領域と総配線長が小さくなっていることが分かる。

6. まとめ

本稿では、ニューラルネットワークおよびシーケンストリップルを用いて3次元VLSIにおけるフロアプランを求める手法を提案した。

文 献

- [1] V. Krishnamurthy, J. Rose, R. Saia, and K. Durocher, "Towards an ultra-high density interconnect (UHD) technology", Proceedings of 1995 Symposium on VLSI Technology, pp.135-136 (1995).
- [2] S. Takahashi, T. Onodera, Y. Hayashi, and T. Kunio, "A new 3-D MCM fabrication technology for high-speed chip-to-chip communication: vertically connected thin-film chip (VCTC) Technology", Proceedings of 1995 Symposium on VLSI Technology, pp.135-136 (1995).
- [3] W. M. Meleis, M. Leiser, P. Zavracky, M. M. Vai, R. B. Brown, A. T. Ishii, "Architectural design of a three dimensional FPGA", Proceedings of Seventeenth Conference on Advanced Research in VLSI, pp.256-268 (1997).
- [4] T. Kunio, K. Oyama, Y. Hayashi, M. Morimoto, "Three dimensional ICs, having four stacked active device layers", Proceedings of 1989 IEEE International Electron Devices Meeting, pp.837-840 (1989).
- [5] K. Yamazaki, Y. Itoh, A. Wada, K. Morimoto, and Y. Tomita, "4-layer 3-D IC technologies for parallel signal processing", Proceedings of 1990 IEEE International Electron Devices Meeting, pp.599-602 (1990).
- [6] 加藤圭太, 大村道郎, "3次元VLSIレイアウト設計における仮想位置に基づく配線手法", 情報処理学会論文誌, 42, 6, pp.1730-1733 (2001).
- [7] F. K. Miyazawa, and Y. Wakabayashi, "An Algorithm for the Three-Dimensional Packing Problem with Asymptotic Performance Analysis", Algorithmica, Vol. 18, No1, pp.122-144 (1997).
- [8] 村山洋, 藤吉邦洋, 中武秀寿, 梶谷洋司, "A Solution Space of Size $(n!)^2$ for Optimal Rectangle Packings", 第8回 回路とシステム梶井沢ワークショップ論文集, pp.109-144 (1995).
- [9] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair", IEEE Trans. on CAD, Vol.15, No12, pp.1518-1524, (1996).
- [10] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair", IEEE Trans. on CAD, Vol.15, No12, pp.1518-1524 (1996).