中央大学理工学研究所論文集　第8号　2003年
Journal of the Institute of Science and Engineering. Chuo University

VLSI のフロアプラン最適化における解の評価手法について

早川二郎*　藤田隆生*　築山修治*

論文概要

VLSI のフロアプランは，機能設計などの設計の初期段階から，遅延等のシステム性能を効果的に見積もる手段として有用である．この問題に対して，アニーリング法や遺伝的アルゴリズム等の解の変換を繰り返して解を求めるアルゴリズムを適用する際，どのように解を評価すべきかという問題は，問題の定式化とも関連する重要な問題で，これによって，解の質や，最適解への収束の速度が変化する．本文では，現在の解がどれだけ最適解に近いかを示す指標として，分散および密度という概念を導入し，これを目的関数に導入することを提案する．また，幾つかの実験により，その有効性を調べる．

1　はじめに

回路の大規模化と微細化が進み，素子に比べて配線の比重が増すに連れ，システム性能に対する配線遅延の影響が大きくなっている [1]．正確な配線遅延は配線設計を終了しないと分からないが，機能設計や論理設計などの設計の早い段階からフロアプラン設計を行い，機能ブロックの概略配置を決定しておけば，遅延見積もりの精度を上げることができるため，VLSI 設計におけるフロアプランの重要性が増している [2]．ここで，VLSI のフロアプランとは，矩形の機能ブロックを互いに重ならないように，チップ内に配置する問題であり，面積や総配線長の最小化などが目的関数である．特に，システム LSI のように，多くの機能を搭載した LSI においては，システム性能に対するフロアプランの影響は大きい．

フロアプラン問題に対して，これまで数多くの手法が提案されてきたが [3]，この問題は NP 困難であるため [4]，ヒューリステイック手法が実用的解法となる．これらの手法は，アニーリング (Simulated Annealing:SA) 法 [5] や遺伝的アルゴリズム (Genetic Algorithm:GA) [6] などの，解の変換を繰り返して最適解を見出すもの，力学的モデルを用いるもの [7]，矩形双対 (rectangular dual) を用いるもの [8] 等に大別できる．これらの中で，SA 法や GA 法等の解の変換を用いる手法は，比較的長い計算時間を必要とするものの，最適解あるいはそれに近い解を見出すという点で，有用な手法である．

しかし，フロアプラン問題のような多目的最適化問題を，SA 法や GA 法等のような解変換を繰り返す手法で解く場合，複数の目的を 1 つの関数に縮約してしまうため，どのような目的関数を設定するかが重要となる．特に，解の制約条件を目的関数の中に組み込み，制約条件を満たさない解は最適解から大きく外れるような値を持つよう，目的関数を変形して解くような場合，アルゴリズム中で解を評価するために用いる変形された目的関数の値は，必ずしも最適化したい目的関数の値となっておらず，費やすことができる計算時間に実用上上限があるため，目的関数の変形のさせ方によって，収束の速度や解の質など，アルゴリズムの性能が変化する．

本文では，SA 法や GA 法等の解変換を用いたフロアプラン手法における解の評価手法について考察する．ここで言う解の評価手法とは，解の変換を受理するか否かを判定するための値を計算するための手法で，上記の変形された目的関数の意味である．まず，総チップ面積，総配線長，並びに幾つかの指定されたネットの配線長等を考慮した目的関数を示し，その効果を調べる．その後，分散および密度という新しい概念を導入した解の評価手法を提案し，アルゴリズムの改良を行う．さらに，実験により，提案手法の有効性を確かめる．なお，本文は，文献 [9] にデータを追加し，纏め直したものである．

* 中央大学理工学部 電気電子情報通信工学科（〒112-8551 東京都文京区春日 1-13-27）

2　フロアプラン問題

ここで考えるフロアプラン問題とは，与えられた n 個の矩形ブロック $\{B_1, B_2, \ldots, B_n\}$ を互いに重なることなく最小面積の矩形内部に配置する問題である．その際，各ブロック B_i に対して，それを実現する上で必要となる矩形の形状の候補 $S_{Bi} = \{(w_1^i, h_1^i), (w_2^i, h_2^i), \ldots, (w_{m_i}^i, h_{m_i}^i)\}$ が指定される．ここで，各 h^i および w^i は，それぞれブロック B_i を実現する矩形の高さおよび幅である．また，ネットリストといくつかのネットに対する許容配線長 (ネット制約) も与えられる．ここでネットとは，結線すべきブロックの集合 $N_j = \{B_1^j, B_2^j, \ldots B_{k_j}^j\}$ であり，ネットリストとは，このようなネットの集合 $\{N_1, N_2, \ldots, N_l\}$ である．フロアプラン問題では，指定されたネットの配線長を許容範囲内にすることが求められることもある．このようなネット制約を $\{C_{N1}, C_{N2}, \ldots, C_{Nl}\}$ と表す．ここで，各 C_{Ni} は許容配線長を示し，制約が無い場合は $C_{Ni} = 0$ とする．

矩形の配置 (フロアプラン) を表す為に，ここではシークエンスペア法 [10] を用いる．矩形配置の表現方法には，ordered tree [11] など，より簡便な手法もあるが，本文ではこれを用いる．本文の主たる考察対象は，SA 法や GA 法における解の評価手法が最終解や収束時間に与える影響なので，解の表現手法の影響は比較的小さいと期待できる．

シークエンスペア法は，与えられた n 個のブロックの 2 つの系列 Γ_+ および Γ_- の対 (Γ_+, Γ_-) によって配置を表現するもので，次のような相対的な位置関係を規定する．

1. Γ_+, Γ_- の両系列において，ブロック x より後ろにあるブロックは，x より右に配置される．
2. Γ_+ ではブロック x より前，Γ_- ではブロック x より後ろにあるブロックは x より上に配置される．

どのような配置に対しても，それを表現するシークエンスペア (Γ_+, Γ_-) が存在するから，n 個のブロックに対して $(n!)^2$ 個のシークエンスペアが存在する．従って，それらの中から最適なシークエンスペアを見い出す問題がフロアプランであると言える [10]．なお，実際の配置は，シークエンスペアから得られる相対位置関係に基づき，ブロックを左下詰めで配置していくことにより，得ることができる [12]．

本文では，各ブロック B_i の形状番号 $s_i (1 \leq s_i \leq m_i)$ をブロック順に並べた系列 $S = (s_1, s_2, \ldots s_n)$ を用いて，各ブロックが取る形状を表わし，シークエンスペア (Γ_+, Γ_-) と形状の系列 S の組 (Γ_+, Γ_-, S) でフロアプランを表現する．

3　目的関数

フロアプランでは，チップ面積だけでなく，総配線長も最小化する必要があるが，必ずしもこれらを同時に最小化できるとは限らない．そこで，ここではこれらの重み付き総和を目的関数とする．その際，面積と総配線長の次元や絶対値の違いを考慮し，目的関数を以下のように定義する．

$$\alpha \frac{\text{チップの面積}}{\text{理想のチップ面積}} + \beta \frac{\text{総配線長}}{\text{理想の総配線長}} \tag{4.1}$$

ここで，チップ面積は全ブロックを囲む最小矩形の面積を，理想の面積は各ブロックの最小面積の総和を表す．また，本文では，各ネットの端子位置までは考慮していないので，各ネットの配線長は，そのネットに含まれる全てのブロックを囲む最小矩形の周囲長の半分で近似する．従って，総配線長はこのような配線長の総和である．理想の総配線長は，各ネットの理想配線長の総和であり，各ネットの理想配線長は，そのネットに含まれる各ブロックを実現する最小面積の総和の平方根の 2 倍である．明らかに，この目的関数の最小値は $\alpha + \beta$ である．α および β は重み係数で，チップ面積と総配線長のどちらを重視するかによってその値を決める．例えば，$\alpha = 0, \beta \neq 0$ とすると，フロアプラン問題を，チップ面積を無視し，総配線長だけを最小化する問題として定式化することになる．

この目的関数の効果を示すため，ランダムに生成した 200 個のブロックに関する配置結果を図 1 に示す．

この結果は，後述する SA 法を用いて得られた結果であり，ネットの個数は 444 個で，ネット制約は無いものとしている．また，総配線長を無視し，面積だけを最小化した時 $(\alpha = 100, \beta = 0)$ の面積と総配線長を基準

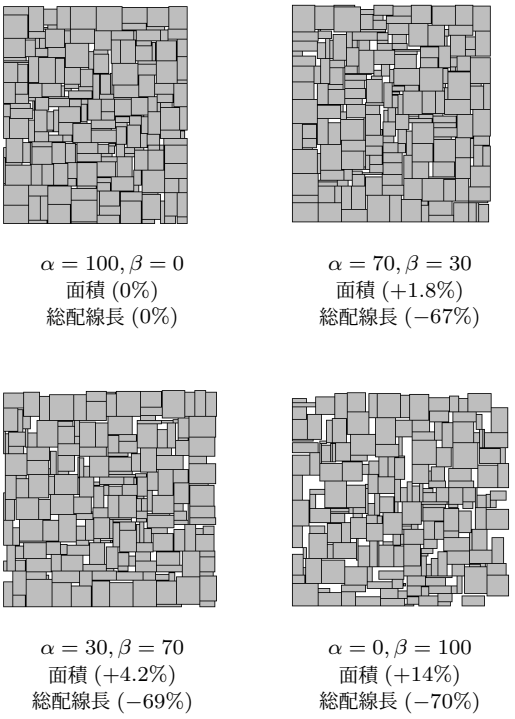

図 1　総配線長に関する実験

とし，α，β を変化させた場合の面積と総配線長の増減の割合を各図の下に示す．このように α と β を変化させることにより，目的に応じた配置を得ることができる．この結果から，$\alpha = 70$，$\beta = 30$ とすると，配線長が 67% も減少するにも関わらず，面積の増加を 1.8% に押さえることができることが分かる．そこで，本文では，目的関数は $\alpha = 70$，$\beta = 30$ とした式とし，これを最小化する問題を考えることにする．

4　SA 法と GA 法

SA 法 [13] は，解変換の繰り返しで最適解を見出すアルゴリズムで，解を改悪するような変換の受理を，温度と呼ばれるパラメータによって制御する手法である．これを本文で対象とするフロアプランに適用する際，解 (Γ_+, Γ_-, S) の変換手法として，次の 3 つを採用し，これらをランダムに選ぶことにする．

(**i**) Γ_+, Γ_- のどちらかの系列から，2 つのブロック番号をランダムに選び，その系列において 2 つの番号の位置を交換する．

(**ii**) Γ_+ の中から 2 つのブロック番号をランダムに選び，それらの番号を Γ_+ と Γ_- において同時に交換する．

(**iii**) S の中から，1 つのブロックを選び，その形状をランダムに変換する．

また，ブロックの系列 Γ_0 と形状の系列 S_0 をランダムに生成し，初期解 $(\Gamma_0, \Gamma_0, S_0)$ とする．明らかに，この初期解から，上に挙げた変換を繰り返して，任意の解に到達できる．

初期温度はどんな解の悪化も受理されるように設定しなければならないが，これは解の評価手法が決定しなければ決まらない．我々の目的関数では，チップ面積の比重を大きくしているので，チップ面積の差が理想面積の 2 倍，すなわち面積が 3 倍になるような変換も受理可能な温度を用いる．

実験における各温度での平衡条件は，ブロックの個数 n に対して，$100n$ 回の解変換をした時とし，温度の減少率を 0.97 とする．また，終了条件は，その温度での平衡条件を満たすまでに，新たな解の受理が 1 回もな

いことが 20 回続いたときとする．なお，これらは試行により実験的に定めた値である．

GA 法 [14] は，同時に複数個の解を管理しながら解変換を繰り返すアルゴリズムで，交叉と呼ばれる特徴的な変換が用いられる．本文で考察するフロアプラン問題に対しては，親となる解 (染色体) (Γ_+, Γ_-, S) の Γ_+ と Γ_- を交換するなど，幾つかの交叉手法が考えられる．しかし，これまでの試行では，最適な交叉手法が発見できなかったので，以下では，解 (Γ_+, Γ_-, S) のシークエンスペア (Γ_+, Γ_-) の一方の系列に対して，図 2 に示すような交叉を行うことにする．すなわち，親となる染色体を 2 つ選び，(Γ_+, Γ_-) のどちらの系列に対して交叉を行うかをランダムに選択する．例えば，Γ_+ の交叉を選択した場合，親 1 の Γ_+ の遺伝子を subset1 および subset2 に分け，subset1 に含まれる遺伝子は子 1 の同じ場所に写す．次に，subset2 に含まれる遺伝子を親 2 の遺伝子の順に並び替えて，子 1 の空いている場所に写す．子 2 には，subset2 の遺伝子をそのまま写し，subset1 の遺伝子を親 2 の順に並び替えて，子に写す [15]．子 1，2 ともに，Γ_- および S は，それぞれ親 1 の Γ_- および S をそのまま写す．

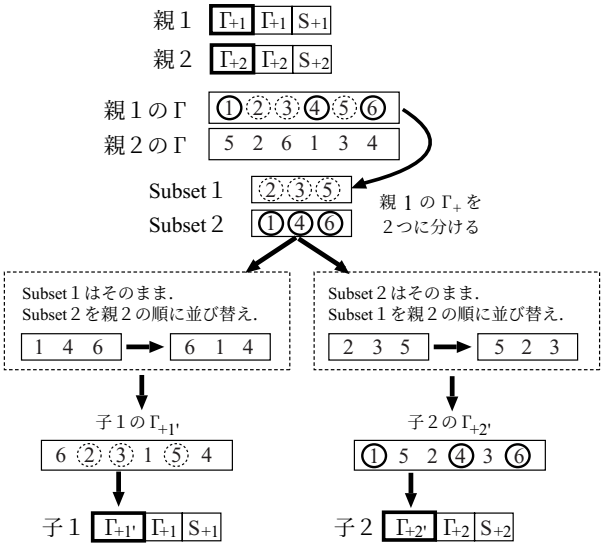

図 2　交叉

交叉以外の解変換として，突然変異も用いるが，これは，SA 法の場合の変換方法 (**i**) および (**ii**) に加えて，

(**iv**) Γ_+，Γ_- のいずれかの系列から，1 つのブロック番号を選び，これをその系列の別の場所に挿入する

という操作を付加する．また，交叉と突然変異は 6 : 4 の比率でランダムに選ぶ．

GA 法に必要な他のパラメータは，試行により，下記のように定めた．一世代の個体数は，$25 \times n$(n:ブロックの個数) とし，初期世代の個体はランダムに生成する．新たに生成された子の個数が $25 \times n \times 0.2$(一世代の個体数の 2 割) に達したら，世代を交代する．その際，淘汰される個体の選択方法や，親となる個体の選択方法は，ルーレット選択 [16] を用いる．終了条件は $150 \times n$ 世代に到達した場合とする．

5　解の評価手法

SA 法や GA 法においては，解の評価方法が重要である．ここでいう解の評価手法とは，解の変換を行ったとき，新たな解を受理するか否かを判定する際に用いる手法のことで，SA 法ではコスト関数，GA 法では適応度と呼ばれることが多い．SA 法においては，金属の焼き鈍し過程の模倣から，コスト関数はエネルギー関数の形が良いと言われている [13]．また，GA 法においては，少数の親のみが子を生成することが無いように，目的関数の値を適応度に対応付ける必要がある [15]．

フロアプランのように，制約条件を持つ問題の場合，最もよく用いられる方法は，目的関数に制約条件の項

を加えて，SA 法のコスト関数，GA 法の適応度の逆数 (最小化問題なので) とするものである．すなわち，

$$COST = \alpha \frac{\text{現在のチップ面積}}{\text{理想のチップ面積}} + \beta \frac{\text{現在の総配線長}}{\text{理想の総配線長}} + \delta \tag{4.2}$$

を SA 法の現在の解のコストの値，この逆数を GA 法の適応度とするものである．δ は，ネット制約を満たす為に，あるネットの配線長が制約長を越えたとき，他の項に比べて大きな値を取るような関数である．(なお，チップ形状の縦横比に制約があるような場合でも，このような δ を用いて，解のアスペクト比を制御できる.)

しかしながら，$COST$ の値が離散的に変化する場合，現在の解が，採用した変換に関して，より最適解に近いということが表現されず，最適解への収束が遅れるという問題が生じる．例えば，図 3 のような同じ大きさの正方形 36 個を 6×6 に配置する問題の場合，(a) ではあと 1 回の変換で最適解になるが，(b) では数回の変換が必要となる．しかし，$\beta = \delta = 0$ であると，$COST$ の値は同じである．

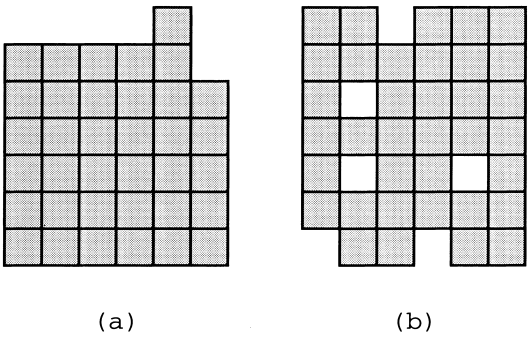

図 3 36 個のブロックの配置

そこで，$COST$ に新たな項を付加し，$COST$ をより連続的に変化させることを考える．そのような項として，以下では分散および密度という概念を提案する．これらは，どちらもブロックの配置の中心への詰まり具合いを表すものである．

5.1 分散

分散とは現在の配置の中心点から，各ブロックの中心点までの x 方向あるいは y 方向の距離のうち, 長い方の自乗を全てのブロックに関して加えたものである．例えば，図 4 のような配置に関しては，1 マスの 1 辺を長さ 1 とすると，分散 $= 5^2 + 6^2 + 5^2 + 5^2 = 111$ である．

この分散を面積や配線長と同様に理想値で割り，重み γ を掛けて評価関数に組み込む.

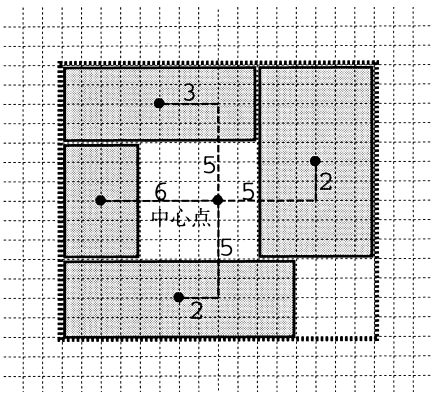

図 4 分散の例

$$COST = \alpha \frac{\text{現在のチップ面積}}{\text{理想のチップ面積}} + \beta \frac{\text{現在の総配線長}}{\text{理想の総配線長}} + \delta$$

$$+ \gamma \frac{\text{現在の分散}}{\text{理想の分散}} \tag{4.3}$$

この時，理想の分散の値は，理想のチップを正方形と考え，その正方形の中で全てのブロックの中心が，正方形の中心からチップの辺までの半分の位置にあるとした値

$$\text{理想の分散} = \left(\frac{\sqrt{\text{理想のチップ面積}}}{4} \right)^2 \times \text{ブロック数} \tag{4.4}$$

とする．

以下に述べるように，分散は，最適解への収束に効果を発揮するが，いくつかの問題点もある．

問題点 1：面積との整合性：　これは，チップ面積が小さくなるときに必ずしも分散が小さくなるとは限らないという問題である．

例えば，36 個の同じ大きさの正方形を配置する場合，面積が 7×7 の配置から 7×6 の配置になるとき，さらに，それが 6×6 の配置になるときには，幅と高さが共に減少するため，面積も分散も同時に小さくなる．しかし，$7 \times 7 = 49$ の配置が $8 \times 6 = 48$ の配置になるようなときには，面積は小さくなるが横幅が大きくなるため，分散は大きくなる．従って，γ の大きさによっては，解の評価に悪影響を与えかねない．

図 5 に，36 個の正方形の配置問題 (ネット無し) に対して，SA 法を適用した場合の影響を示す．図の各折れ線は，面積が小さい解が見つかる度に，その解の分散 ($COST$ の第 4 項の値) と面積 ($COST$ の第 1 項の値) をプロットしたもので，下から順に，(a) 分散の割合に 10 の重み ($\gamma = 10$) を掛けた値 a，(b) 分散の割合に 30($\gamma = 30$) の重みを掛けた値 b，(c) 面積の割合に 100 の重み ($\alpha = 100$) を掛けた値 c，(d) $c + a$ の値，並びに (e) $c + b$ の値である．

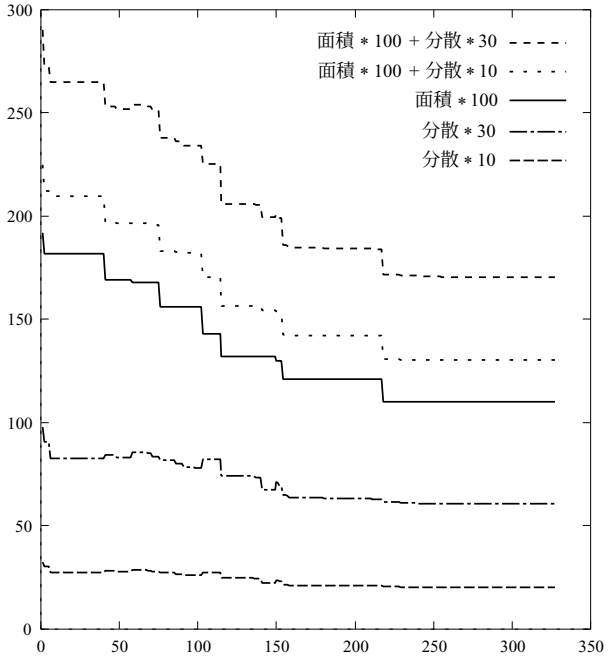

図 5　分散の効果

これから分かるように，面積のみ（c）のグラフが離散的なのに比べ，$\gamma = 10$ として分散を加えたもの (d) は，$COST$ の値が比較的連続的に変化する．また，$\gamma = 30$ として分散を加えたものは，面積が小さくなっているにもかかわらず，$COST$ の値が大きくなることがある．

問題点 2:ブロックの面積のばらつきの影響： 分散を小さくするには，できるだけ多くのブロックを配置の中心近くに配置するのが良い．従って，ブロックの面積のばらつきが大きい場合，小さな面積のブロックが中心に集まる傾向がある．

図 6，7 は，面積に差のある 200 個のブロックを SA 法を用いて配置した結果である．図 6 は分散を評価関数に組み込まないもの，図 7 は分散を組み込んだものである．図 7 では小さなブロックが中心に集まる傾向が見える．このことは最終的なチップ面積に大きな影響をおよぼすことはないが，総配線長に悪影響をおよぼす可能性がある．

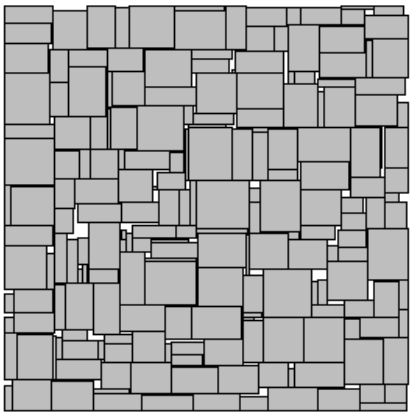

図 6 分散を使わない配置

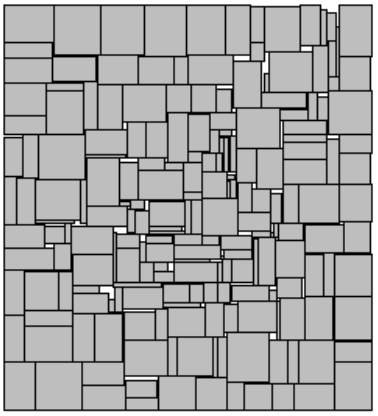

図 7 分散を使った配置

5.2 密度

分散の問題点を改善するため，分散の代わりに密度を導入する．

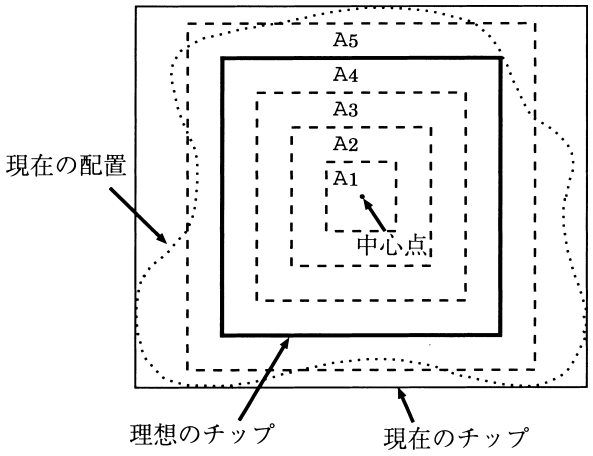

図 8 密度の定義

図 8 のように，配置の中心に理想面積と同じ面積をもつ正方形 (図の太線) と，その正方形の 1 辺の 1/4 倍，2/4 倍，3/4 倍，5/4 倍の長さを 1 辺とする正方形を置き，それらを用いて領域を $A_1, A_2, \cdots, A_5$ の 5 個に分割する．ここで，A_1 は，配置の中心から一番内側の正方形までの領域，A_2 は，一番内側の正方形からその外側にある正方形までの領域とし，同様に A_3，A_4，A_5 の領域を定める．これらの各領域において，実際にブロックによって占有されている面積の，各領域の面積に対する割合をそれぞれ $D_1, D_2, \cdots, D_5$ とし，密度を次のように定義する．

$$\text{密度} = 5 \cdot D_1 + 4 \cdot D_2 + 3 \cdot D_3 + 2 \cdot D_4 + 1 \cdot D_5 \tag{4.5}$$

密度は，最大化が目的となるので，理想の密度を現在の密度で割った形で評価関数に組み込む．理想の密度は，全ブロックが理想面積の正方形に入っている状態，すなわち，A_1 から A_4 に完全に詰まっている状態とすると，14 になる．

$$COST = \alpha \frac{\text{現在のチップ面積}}{\text{理想のチップ面積}} + \beta \frac{\text{現在の総配線長}}{\text{理想の総配線長}} + \delta$$
$$+ \gamma \frac{\text{理想の密度}}{\text{現在の密度}} \tag{4.6}$$

同じ大きさの正方形 36 個を 6×6 の最小面積に配置する問題に対して，分散と密度の有効性を検証した．表 1 に SA 法および GA 法を共に 10 回試行した結果を示す．ここで，平均変換回数とは，SA 法では解の変換を平均この回数，GA 法では新たな子孫が平均この回数生成された時点で，最適解が発見されたことを示す．なお，終了条件などは，前節に述べた通りである．

表 1　正方形 36 個の配置結果 ($\alpha = 100, \beta = 0, \gamma = 10$)

		密度	分散	無し
SA	成功 / 失敗	10 / 0	9 / 1	3 / 7
	平均変換回数	752,760	695,528	1,594,080
GA	成功 / 失敗	9 / 1	7 / 3	4 / 6
	平均変換回数	427,206	577,667	927,383

この実験より，SA 法，GA 法ともに，分散あるいは密度を導入することにより，限られた時間内に面積最小の解が発見される確率が高いことが分かる．また，分散よりも密度の方が最適解に到達する確率が高くなっている．さらに，最適解が見つかるまでの変換回数は，SA 法では分散より密度の方が多少変換回数が増えたが，GA 法では分散の時より密度の方が減少している．分散あるいは密度を用いない場合と比べて，この程度の回数の変化は誤差の範囲内であると考えられる．なお，分散および密度共に，ブロック数に比例する計算量で求めることができ，計算時間の負荷は大きくない．ちなみに，上記の例では，Ultra SPARC-IIi(333Mhz) で約 2,000 秒の CPU 時間を要したが，分散や密度を導入したときの増加量は，3% 以下であった．

6　実験結果

分散や密度が，ネットを持った実際のデータに対してどの程度効果があるかを調べた．その結果の一部を以下に示す．以下で，ami33，ami49 は MCNC ベンチマークデータで，残りはランダムに生成したデータである．ami33 は，ブロックの個数が 33，ネットの個数が 80，ami49 は，ブロックの個数が 49，ネットの個数が 376 である．その他は，初めの数字がブロックの個数で，ブロックの個数が 100 のデータはネットの個数が 320，200 個のデータはネットの個数が 444 である．名前の最後に s の付いているものはブロックの面積のばらつきが小さいデータ，l の付いているものはブロックの面積のばらつきが大きいデータ，f の付いているものはブロックの面積のばらつきが小さく，かつブロックの形状の候補が複数あるデータである．

表 2 および 3 は，それぞれ SA 法および GA 法において，分散あるいは密度を導入したとき，見出された評価関数（$COST$）最小の解の面積および配線長が，分散も密度も導入しなかったときに見出された評価関数最小の解の面積および配線長に対して，それぞれどの程度変化したかを示している．すなわち，1% は 1% 増加を，-1% は 1% 減少したことを示す．これらは，5 回の試行の平均で，$\gamma = 10$ としたときの値である．ちなみに，最終解の評価関数 $COST$ の値は，どれも 117 〜 129 の値であった．

これらの結果から，配線長を考慮した場合には，分散および密度の効果はそれほど顕著ではないと言える．しかし，ブロックの面積のばらつきが小さく，問題の規模が大きい場合，特に，各ブロックが複数の形状を持つような場合には，分散および密度は有効に働き，これらを導入しなかった場合に比べて，面積の小さい解を効率良く見出すことが分かる．また，平均的には，分散の方が面積削減効果が大きいことが分かる．しかし，配線長の増加が大きくなる傾向にあるため，実用的には密度の方が好ましいであろう．

表 2 SA における分散および密度の効果
$(\alpha = 70, \beta = 30, \gamma = 10)$

	面積		配線長	
	分散	密度	分散	密度
ami33	1.08%	0.64%	-1.77%	-2.59%
ami49	0.22%	-0.39%	2.38%	5.90%
100data_s	-0.59%	-0.65%	1.53%	2.16%
200data_s	-2.37%	-1.44%	1.27%	0.63%
100data_l	-1.21%	-0.68%	-1.14%	-0.54%
200data_l	-0.59%	-0.38%	1.77%	-1.89%
100data_f	-1.04%	-1.79%	3.70%	2.01%
200data_f	-8.34%	-4.81%	4.64%	1.81%
average	-1.61%	-1.19%	1.55%	0.94%

表 3 GA における分散および密度の効果
$(\alpha = 70, \beta = 30, \gamma = 10)$

	面積		配線長	
	分散	密度	分散	密度
ami33	-3.37%	0.71%	-0.01%	0.01%
ami49	0.72%	0.56%	0.01%	-0.91%
100data_s	-1.01%	-1.47%	1.42%	0.01%
200data_s	-2.09%	-0.99%	2.09%	0.56%
100data_l	-0.12%	-1.96%	1.02%	-0.09%
200data_l	0.02%	-0.97%	3.09%	2.21%
100data_f	-2.01%	-1.95%	0.71%	2.25%
200data_f	-4.10%	-2.09%	4.41%	1.08%
average	-1.50%	-1.02%	1.59%	0.64%

表 4 は，評価関数における重み係数 γ を 30 にしたとき，得られた解の総配線長が，分散も密度も導入しなかった時の総配線長からどの程度増加したかを示している．表 2 および 3 の総配線長の増加率と比べることにより，γ を大きくすると，総配線長が増加してしまうことが分かる．これは，分散も密度も面積に関係した項であるため，γ を増加させたことにより，評価関数における総配線長の重みが相対的に下がったことが原因である．従って，面積の最小化を行いつつ，総配線長を最小化したい場合には，γ は 10 程度が良いと考えられる．なお，本文での目的関数は，面積の最小化を主とし（$\alpha = 70, \beta = 30$），面積の離散的な変化を軽減する解の評価手法を考えたため，このようなパラメータの値が適切という結論になっている．

表 5 に SA 法と GA 法によって得られた解の比較を示す．これらは，評価関数に密度を用い，$\gamma = 10$ とした

表 4 γ の配線長への影響
$(\alpha = 70, \beta = 30, \gamma = 30)$

	SA		GA	
	分散	密度	分散	密度
ami33	3.24%	-0.29%	0.02%	0.01%
ami49	4.83%	6.49%	1.53%	-0.01%
100data_s	3.01%	2.92%	1.98%	0.02%
200data_s	5.17%	0.62%	2.71%	0.70%
100data_l	3.57%	2.06%	4.27%	1.02%
200data_l	1.53%	3.99%	3.99%	1.08%
100data_f	6.51%	5.73%	3.96%	2.98%
200data_f	9.48%	3.08%	5.24%	3.30%
total	4.67%	3.08%	2.96%	1.14%

表 5 SA に対する GA の結果

	面積	配線長
ami33	-0.07%	0.78%
ami49	0.01%	0.00%
100data_s	1.96%	2.01%
200data_s	2.84%	3.23%
100data_l	2.01%	2.21%
200data_l	3.01%	3.24%
100data_f	3.13%	3.26%
200data_f	4.55%	4.21%
total	2.18%	2.37%

とき，SA 法で得られた解の $COST$ の値に対して，GA 法で得られた解の増分である．問題の規模が小さい ami33 や ami49 ではほとんど差はないが，現在のところ，同じ評価関数を用いながら，SA 法の方が良い結果を出力する．これは GA 法で用いられる交叉が適切でないためと考えられるが，この改良は未解決である．

以上の実験は，C 言語を用いて作成したプログラムを，Ultra SPARC-IIi(333Mhz) で実行した．解の評価手法や解の変換手法などにおいてプログラムの効率化が不十分であり，ブロック 100 個のフロアプラン問題を解くために要した計算時間は，SA 法で約 6,733 秒，GA 法で約 35,011 秒であった．これらに関しては，問題点が分かっているので，プログラムの改良により，改善されるであろう．

7 むすび

本文では，ネット制約を持ち，チップ面積および総配線長の両方を最小化するというフロアプラン問題を対象に，SA 法や GA 法などの解の変換を用いた解法において，どのように解の良さを評価すべきかを検討した．離散的に変化する目的関数に，分散および密度という新しい指標を加え，それを評価関数とすることにより，得られる解の質や最適解への収束の速度が向上することを示した．また，提案した解の評価手法を用いた SA 法や GA 法を実際のデータに適用し，その有効性を検証した．なお，分散や密度の計算はブロックの個数の線形時間で行えるので，計算の負荷は数%以下であり，実用上問題はない．ただし，現状のプログラムの効率化は不十分であるため，その改良により計算時間が短縮される可能性がある．その場合には，計算負荷の割合が増加するかもしれない．

実験結果によれば，分散も密度も面積の最小化に対して効果があり，分散の方が削減率は大きかった．しかし，総配線長も含めて考えると，その効果は減少し，密度の方が総配線長の増加が少なく，実用的であると言える．今後の課題は，分散や密度にかける重みなど，各種パラメータの調整や，総配線長と整合のとれた分散や密度に対応する概念を見出すこと等が挙げられる．

最後に，横長な配置を実現したい場合について触れておく．この場合，分散や密度はそのままの形で用いる

ことはできない．それは，正方形に近い配置を最適として分散や密度を定義したからである．こうした理由は，目的関数が面積や総配線長だけでも，見出される解は正方形になる傾向が強いからである．しかし，分散や密度を用いて，横長な配置を求めることも可能である．例えば，縦横比が 3 : 1 の配置を得たい場合，分散を計算する際の x 方向の距離には 0.75 を，y 方向の距離には 0.25 を掛ける．また，密度の場合は，基準となる正方形の代わりに縦横比が 3 : 1 の長方形を用いればよい．しかしながら，どちらの方法も，縦横比が 2 倍から 4 倍というような幅のある要求を満足することは少々困難である．これらに対する考察も今後の課題である．

参考文献

[1] J.Cong, Z.Pan, L.Hei, C.K.Koh, and K.Y.Khoo, “Interconnect design for deep submicron ICs,” Dig. Tech. Papers ICCAD, pp.478-485, 1997.

[2] K.Bazargan, S.Kim, and M.Sarrafzadeh, “A floorplanner of uncertain designs,” IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, Vol.18, No.4, pp.389-397, 1999.

[3] N.A.Sherwani, Algorithms for VLSI Physical Design Automation, Kluwer Academic Publishers, pp.175-221, 1995.

[4] B.S.Baker, E.G.Coffman, and R.L.Rivest, “Orthogonal packings in two dimensions,” SIAM J.Comput., Vol 9, No.4, pp.846-855, 1980.

[5] D.F.Wong and C.L.Liu, “Floorplan design of VLSI circuits,” Algorithmica, Vol.4, pp.263-291, 1989.

[6] H.Esbensen and E.S.Kuh, “An interactive floorplanner for design space exploration,” Proc. ISCAS, Vol.4, pp.500-503, 1996

[7] H.Eisenmann and F.M.Johannes, “Generic global placement and floorplanning,” Proc. of Deseign Automation Conf., pp.269-274, 1998.

[8] S.Tsukiyama, M.Maruyama, S.Shinoda, and I.Shirakawa “A condition for a maximal planar graph to have a unique rectangular dual and its application to VLSI floor-plan,” Proc. Int. Symp. on Circuits and Systems, pp.931-934, 1989.

[9] 早川二郎, 藤田隆生, 築山修治 “フロアプラン最適化における解の評価について,” 2000 年電子情報通信学会総合全国大会論文集, A-3-7, p.75, 2000.

[10] H.Murata, K.Fujiyoshi, S.Nakatake, and Y.Kajitani, “VLSI module placement based on rectangle-packing by the sequence-pair,” IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, Vol.15, No.12, pp.1518-1524, 1996.

[11] P.-N.Guo, T.Takahashi, C.-K.Cheng, and T.Yoshimura, “Floorplanning using a tree representation,” IEEE Trans. on CAD, Vol.20, No.2, pp.281-289, 2001.

[12] 長尾明, 澤卓, 重弘裕二, 白川功, 神戸尚志, “方形パッキング法の一算法,” 電子情報通信学会論文誌, Vol.J81-A, No.10, pp.1362-1371, 1998.

[13] S.Kirkpatrick, C.D.Gelatt, and M.P.Vecchi, “Optimization by simulated annealing,” Science Vol.220, pp.671-680, 1983.

[14] D.E.Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

[15] B.R.Fox and M.B.McMahon, “Genetic operators for sequencing problems,” Foundation of Genetic Algorithms, pp.284-300, 1991.

[16] 坂和正敏, 田中雅博, 遺伝的アルゴリズム, 朝倉書店, pp.22-25, 1995.