遺伝的アルゴリズムによるLSIパッド位置設計

川上　かおり†　　田中　秀俊†　　三石　彰純†　　後藤　明広‡
三菱電機（株）　　†情報技術総合研究所　　‡設計システム技術センター

ＬＳＩパッド位置設計は、連続値を解とし、多数の制約／最適化条件からなる多目的配置問題である。現状のパッド位置設計自動化は、ヒューリスティクスな手法に基づく均一生成である。最終的な制約充足及び最適化のための調整は、ＣＡＤ等を利用して人手で行われるため、多くの時間を要するという課題があった。我々はこのような多目的配置問題に対し、遺伝的アルゴリズムを採用し、アルゴリズム適用に際して、問題のモデル化を行った。制約条件の一部から配置可能領域を算出し、残りの制約／最適化条件から評価値を算出する。配置可能領域算出により、配置対象領域の絞込みだけでなく、制約条件の検証が可能になる。

LSI pad layout designing system using a genetic algorithm

Kaori Kawakami†　　Hidetoshi Tanaka†　　Akitoshi Mitsuishi†　　Akihiro Goto‡
Mitsubishi Electric Corp.　†Information Technology R&D Center　‡Design Systems Engineering Center

Automatic LSI chip designing requires quick optimization of pad's layout, which should satisfy many constraints. For the past few years, LSI pad's layout has been designed using heuristic methods based on empirical knowledge of LSI designers. We applied one of the genetic algorithms (GA) to solve this problem. Our method uses the constraints in two ways: some are used to define feasible wiring positions; and the others are transformed into the penalties in the objective function.

1　はじめに

近年の急速なLSI集積技術の進歩を迅速にLSIに導入して実現するために、設計時間の短縮がこの分野での最重要課題のひとつになっている。この設計短縮を目的として、LSIのリードフレームの調達方法も、LSIチップに合わせてカスタム設計する方法から標準品の利用へと移行している。この移行に伴い、従来はリードフレーム側の各リードのワイヤリング位置の決定を問題としていたが、近年は逆にチップ側のワイヤリング位置であるボンディングパッド位置（以下パッド位置）を決定する方法が問題になってきている。

現状のパッド位置設計の自動化方法は、ヒューリスティックによって均一配置から若干修正する程度の方法が一般的であり[1]、制約充足や最適化調整はCAD等を用いて人手で行われている[2]。本稿では、著者らが構築中のパッド位置設計自動化を含むワイヤリング位置設計システムにおいて、どのようにして、パッド位置設計問題を多目的配置問題としてモデル化し、遺伝的アルゴリズム(Genetic Algorithm, GA)[3]を問題解決手法として適用したかについて述べる。

2　ワイヤリング位置設計問題の分析

まず、ワイヤリング位置の設計について図1を用いて簡単に紹介する。ワイヤリング位置設計は、LSIパッケージ設計の一構成要素である。LSIパッケージは、チップ、チップを取り巻く100本前後のリード、チップとリード間のワイヤを主部品として構成されている。図1はチップとリード間のワイヤリングの例を示している。以下、リード上のワイヤリング点をボンド、チップ上のワイヤリング点をパッドと呼ぶことにする。ワイヤリング位置設計問題は、パッド位置が規定された状態でのボンド位置の決定、ボンド位置が規定された状態でのパッド位置の決定という2種類の問題に分類される。

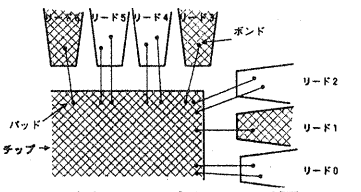

図1：ワイヤリング図

2.1　ボンド位置設計問題

ボンド位置設計問題は、パッド位置すなわちチップ側のワイヤリング位置を、例えば均一配置もしくは所与の位置として固定し、リードとチップ側パッドとの対応が与えられ、さらに設計上のさまざまな制約、コストの条件が与えられた上で、ボンド位置すなわちリードフレーム側のワイヤリング位置を最適化する、制約つき多目的最適化問題である。従来はチップ側が既に作製されたものとして、リードフレームをそれに合わせてカスタム設計していたため、この問題が重要視された。著者らは遺伝的アルゴリズムの一種を用いて設計システム構築を試みた[4]。それによれば、実用的な制約数の問題において実用時間内で実用的な解が得られている。

2.2 パッド位置設計問題

パッド位置設計問題は、ボンド位置設計問題とはちょうど逆に、ボンド位置を例えば均一配置もしくは所与の位置として固定もしくは仮決めし、リードとパッドとの対応が与えられ、さらに設計上のさまざまな制約、コストの条件が与えられた上でパッド位置を最適化する。これも制約つき多目的最適化問題である。本稿ではこの問題に着目する。LSIパッケージ設計効率化の強い要請から、近年はリードフレームをカスタマイズ設計せずに、既存のリードフレームを統合した標準リードフレームを利用し、チップ側で柔軟に対応させようという傾向にある。最終目標はボンド位置未定の状態でパッド位置を求めることだが、それをひとつ簡単化した問題として、本問題のようにボンド位置の仮決めに基づいてパッド位置設計を行うものが重要視されつつある。

2.3 ワイヤリング位置設計の制約・最適化条件

ワイヤリング位置を決定する際に考慮すべき条件の例を図2を用いて紹介しておく。条件には、絶対守るべき条件（制約条件と呼ぶ）と、守った方が良い条件（最適化条件と呼ぶ）がある。多数存在する最適化条件すべてを、目的関数として考えるのは難しいため、一部をこのような形で条件と見なすことでモデルの簡略化を図っている。

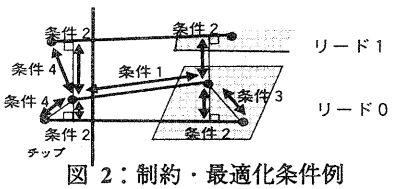

図2：制約・最適化条件例

[条件1] パッドとボンドを結ぶワイヤの長さ「ある一定範囲内で、かつ短ければ短いほど良い」とする。従って、条件1は最適化条件であり、かつ制約条件となる。

[条件2] 隣接するワイヤ間の距離「ある一定距離以上でなければならない」とする。制約条件である。

[条件3] 同一リード上のボンド位置座標間の離「ある一定距離以上でなければならない」とする。制約条件である。単一リード上に複数本ワイヤリングする場合(以下、マルチワイヤと呼ぶ)のボンド位置の相対位置に依存する。

[条件4] 隣接するパッド位置座標間の距離「ある一定距離以上でなければならない」とする。制約条件である。パッド位置の相対位置に依存する。

条件1、条件2はボンド位置及びパッド位置の両方に依存する条件であるが、条件3、条件4はそれぞれボンド位置、パッド位置間の相対位置にのみ依存する。このようにパッド位置設計問題とボンド位置設計問題には対照的な制約／最適化条件が存在する。したがって、仮にパッド位置もしくはボンド位置のいずれかを固定し、他方を設計する問題とした場合、パッド位置設計はボンド位置設計の逆問題と見なせる。

パッド位置設計は、条件4のパッド位置の相互関係だけでなく、条件2が存在することにより、隣接パッドを対象としたワイヤ間の相互干渉も発生するため、より複雑な問題となる。この他、ワイヤリング問題に対しては全部で 20 程度の制約・最適化条件が存在し、これら全ての制約条件に違反することなく、最適に近い解を求めるところにワイヤリング位置設計問題の難しさがある。ワイヤリング位置設計のような多目的最適化問題に対して、人手で解く方法では多くの時間を要する。

2.4 パッド位置設計問題とボンド位置設計問題との比較

パッド位置設計とボンド位置設計とでは、ワイヤリング加工に関する制約条件で対照的な条件が存在するなど、多くの共通要素が存在するが、明らかな相違点の一つに配置対象領域の違いがある。この点に着目して、ボンド位置設計とパッド位置設計とを比較し、パッド位置設計独自に検討が必要な事項の洗い出しを行った。

配置対象領域は、ボンド位置設計がリード領域であるのに対し、パッド位置設計ではチップ全域となる。配置可能領域の重複に関しては、配置対象領域の違いがそのまま、配置可能領域の重複の度合いに関係しており、ボンド位置設計はマルチワイヤ箇所のみであるのに対し、パッド位置設計では任意の近隣パッド間で重複する。制約／最適化条件に関しては、共にワイヤリング加工に関する制約がある他、パッド位置設計に関しては、例えばチップ中心付近への配置不可などのロジック等に関する制約の存在も予想されるが、設計指針として明らかになっていない。実データによる配置可能領域算出実験を行い、重複の度合いを調べ、その度合いに応じて、評価関数の設定方法の工夫や配置可能領域の切り分け等を検討する必要があると思われる。

3 パッド位置設計問題：解法の選択

3.1 実行可能領域の提示

システムの本来の使命は位置の最適化だが、そ

のための条件、つまり絶対に守るべき制約条件やそうあるべきという最適化条件、要求性能、要求コストなどの設定は人手で決めたものであり、中には変更可能なもの、暫定的なものが混じっている。そこで、制約条件から一部を選択させると実行可能領域を算出して設計者に提示する機能をシステムに設けた。この機能により、どの条件が設計に影響を与えているかが試行錯誤できるだけでなく、制約条件、要求コスト、要求性能等について再考を促すことができる。これはこのような多目的最適化問題において、実用上非常に有効な機能と言える。

3.2 近似的最適化法の採用

解のパッド位置は平面上の位置、すなわち 1 パッドあたり 2 変数で表現できる。制約もこれら位置に関する数式で書き下せる。これらの性質から、条件さえ決まれば数理的最適化による厳密解が得られそうにも思える。しかし、制約には距離や角度があるので線形問題にはならず、隣接する複数のパッド間の制約の性質上、実行可能領域が凸になるかどうかは制約によると思われる。これは数理的最適化の適用の妨げとなる要因である。さらに、多目的最適化問題でもあることから、数理的最適化による厳密解を求めることは実用上得策ではないと考えられる。

4 パッド位置設計問題の解法：どんな GA を選んだか

よって著者らは、近似解法の中から、多目的最適化問題に有力な手法の一つで、かつ比較的実装が容易である遺伝的アルゴリズムを解決法として採用した。上述のように、パッド位置設計問題とほぼ同様な性質を示すことが予想されるボンド位置設計への GA の適用実験において、アルゴリズムの実用性が示されており[4]、同様の効果が期待できる。設計アルゴリズムの概要を以下に示す。

4.1 遺伝子コーディングと初期個体生成

各ワイヤに対するパッド位置座標 x,y（実数値表現）をそれぞれ 1 遺伝子とし、チップ上の全パッド位置の組を 1 個体とする。連続値を解とする多目的最適化問題において、実数値表現適用の効果が報告されている[5][6]。

初期個体は、配置可能領域の頂点に生成するなど、ヒューリスティクスな手法で生成する。

4.2 遺伝操作

交叉方法としては、両親 2 個体の中点座標を子とする中点法[7]、及び同様に両親の外点を子とする外点法を採用した。中点法は実数値表現を適用した場合において、形質遺伝に有効で、実用に適した手法と考えられている[7]。外点法についても同様な効果が期待できる。突然変異は 1 遺伝子をランダムに変更し、配置可能領域外に出る場合は境界線に留め、致死遺伝子は発生させないようにする。また交叉後の選択により次世代に残る予定の子個体に対し、同一世代では突然変異を適用しない。交叉と突然変異の組み合わせ実験から、本手法による集中的探索実現への効果が示されている[8]。

4.3 世代交代モデル

交叉前の選択（複製選択）は、ランダムに 2 個体を選ぶ。ランダム抽出により、大域的探索が期待できる。また、交叉を適用した個体は再び同世代での複製選択の対象とする。限られた個体数での多様な個体の生成につながり、大域的探索が期待できる。

交叉後の選択（生存選択）は、親子 3 個体の中から評価値が良いもの 2 つの残す。親と子が同じ場合は子を優先する。選択範囲を限定することにより、多様性の維持が期待できる。

世代交代の範囲は、個体群全体(全家族を対象)とする。進化の活性化が期待できる。

突然変異を適用する個体の選択は、乱数で決定するが、各世代における最優良個体には適用しない(エリート保存戦略)。これは個体群の収束よりも、最良個体の発見を目的とすることによる。

5 パッド位置設計問題の解法：制約条件・最適化条件・適応度

次に、制約／最適化条件のパッド位置設計への適用方法について述べる。

5.1 配置可能領域の算出

全制約条件のうち、パッド位置に依存せずに境界領域を定義可能である条件は、パッド位置の配置可能領域算出に適用する。ＧＡの進化過程において、配置可能領域外にパッドがはみだしてしまった解を致死扱いすることにより、それらの制約条件の評価値算出は不要となる。配置可能領域算出例を図 3に示す。

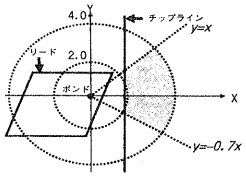

図 3：配置可能領域算出例

図 3はボンドを原点として、ｙ軸がチップライン

に平行な座標系を表す。配置可能領域算出に適用する制約条件が仮に「条件A：パッドとボンドを結ぶ直線の傾き a に対して $-0.7 < a < 1.0$」「条件B：ワイヤ長さ w に対して $2.0 < w < 4.0$」の2つであるとすると、配置可能領域は図 3の斜線部分のように定義される。配置可能領域算出は、配置領域の初期領域に対し、制約条件によって表される境界領域を順に追加して行き、その共通領域を配置可能領域とする。

5.2 適応度（目的関数）

配置可能領域の定義に適用しない制約条件及び最適化条件に対しては、適応度算出を行う。各個体の適応度 F は式（1）で定義される。*WrID* はワイヤＩＤ、r は評価項目、*f[wrID][r]*は評価値、*w[r]*は重み値、*Nw* はワイヤ数を表す。

$$F = \sum_{wrID} \sum_{r} \frac{f[wrID][r] \times w[r]}{Nw} \qquad (1)$$

$F = 1.0$ の時が最良解となる。制約条件、最適化条件それぞれに対する評価関数を、図 4、図 5に示す。*var* は各評価項目の評価対象値、図 4の $KMIN \leq \mathrm{var} \leq KMAX$ は評価対象値が制約を満たす範囲（以下、制約範囲）、図 5の $SMIN \leq \mathrm{var} \leq SMAX$ は評価対象値が最適と見なせる範囲（以下、最適化範囲）を表す。

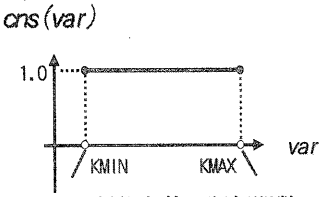

図 4：制約条件の評価関数

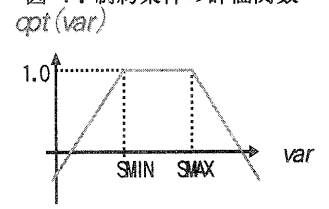

図 5：最適化条件の評価関数

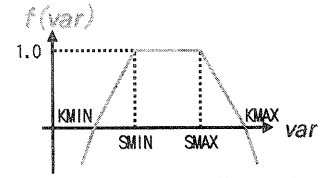

図 6：制約／最適化条件の評価関数

図 6は制約条件及び最適化条件の両方が存在する評価項目の評価関数 *f(var)*である。制約範囲内は最適化条件の評価関数 *opt(var)*、それ以外は制約条件の評価関数 *cns(var)* と等しく、2つの評価関数を合わせた関数となっている。制約範囲外の評価値を極端に低い値にする方法も考えられるが、制約範囲から少しだけはずれた解が進化して最適解になる可能性も高い。また、制約範囲外の個体を残すことにより、解集合の多様性も保つことができる。以上より、制約範囲外の評価関数を制約範囲内と連続するように設定する。

6 まとめ

本稿では、パッド位置設計を含むワイヤリング位置設計問題のモデル化、パッド位置設計に適用するアルゴリズムの概要、制約／最適化条件の適用方法について述べた。パッド位置設計アルゴリズムとして、ボンド位置設計と同様に、ＧＡを採用した。制約条件／最適化条件は、パッド配置可能領域算出と適応度算出の2方法で適用する。実行可能領域である配置可能領域算出により、配置対象領域の絞込みを行うだけでなく、制約条件の検証が可能になる。さらに、配置可能領域を表示することにより、設計者向け設計支援ツールとしての役割を担う。今後は本システムのアルゴリズム検証のため、シミュレーション向けシステムを実装し、シミュレーション実験を行い、配置可能領域の重複に関する検証と、その結果に応じて配置可能領域の切り分け等を検討する予定である。

参考文献

[1] 加藤　他, 自動設計装置および自動設計方法, 特開平 8-340017, 1996.

[2] 松井, レイアウト編集装置, 特開平 5-233760, 1993.

[3] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, Mass., 1989.

[4] 川上　他, LSIパッケージリードフレーム設計へのＧＡの適用, 情処学会 人工生命とその応用シンポジウム論文集, pp.87-94, 1997.

[5] L. Davis, *The Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1990.

[6] C.Z. Janikow and Z. Michalewicz, *An experimental comparison of binary and floating point representations in genetic algorithms*, Proc. 4th International Conference on Genetic Algorithms, pp.31-36, 1991.

[7] H.P. Schwefel, *Numerical Optimization of Computer Models*, Wiley, Chichester, 1981.

[8] K. Kawakami et al., *An application of a genetic algorithm to LSI lead frame designing*, Proc. 4th International Conference on Information Systems, Analysis and Synthesis, pp.297-303, 1998.