

## Simulated Evolution による Partitioning 手法

7K-7

土肥実久, 澁谷利行, 河村 薫  
(株)富士通研究所

## 概要

生物が環境に対して適応していく様子をシミュレーションした Simulated Evolution を LSI の Partitioning 問題に対して適用した。従来までの Min-Cut では、カット数を減らすように deterministic に移動させるセルを決定していたが、これを本手法では Simulated Evolution に置き換えた。

階層的に2分割を繰り返し個々のセルの位置を決定して最終配置まで行なわせ評価を行なった。実行時間は従来の Min-Cut の数倍で、配線後の実配線長は5~20%の改善が見られた。

## 1 はじめに

ここでは、Youssef Saab らの提案した Evolution-Based Approach [4] の考え方を基とした Simulated Evolution (S.E.) を提案し、LSI の partitioning 問題に対して適用することによって、従来の Min-Cut [1, 2, 3] に比べて、より良い解を比較的短時間で求めることが可能であることを示す。

## 2 LSI の partitioning 問題

LSI の実装設計では、回路を構成するそれぞれの素子(セル)を配置し、物理的な位置を決定する。この後、これらの配置されたセルを接続するネットを配線し、実装設計は終了する。

ここでの配置の良否は、すべてのネットを配線できるかどうか、総配線長が短いかどうかによって決まる。

LSI の partitioning 問題とは、回路を構成するそれぞれのセルの配置を決定する際に、より良い配置を得られるように、セル間の論理的な接続関係を見て適切なグループに分割することである。

ここでは LSI の partitioning 問題を、分割において各グループ間のネット数を最小とするものが最適解であると考え、各グループを分割するカットラインに対して、カットラインと交差するネット数(カット数)を最小化する問題とする。

## 3 Simulated Evolution

Simulated Evolution は、生物が環境に対して適応していく様子をシミュレーションし、準最適解を求めようという最適化のアプローチの一つである。この考え方は、最近の Genetic Algorithm (G.A.) にも見られるように生物がその環境に適応してきて進化し生き残り続けてきたことにヒントを得ている。

G.A. は遺伝子の配列から得られる形質の優劣によって生存競争が行なわれた結果、最後にはもっとも環境に適応した形質を持つ遺伝子が残るであろうことを期待している。

それに対して、S.E. の基本的な考え方は、各生物個体が現在の環境に対して適合している場合には、その個体は生き残ることができるが、適合していない場合には、その環境下では生き残れないので、より生き残りやすい環境を求め新しい環境へと移住し、その結果、最終的にはそれぞれの個体ももっとも生きやすい環境にいるであろうことが期待される。

S.E. は、この考え方に基いて生物の各個体をそれぞれのセルに見て、各個体ももっとも生きやすい環境に移住する過程を、各セルがもっとも良いパーティションに分割される過程であると考えることによって、LSI の Partitioning 問題の準最適解を求めようというものである。

すなわち、従来までの Min-Cut 法では、カット数を減らすように deterministic に移動させるセルを決定していたが、セルを移動させカット数を最小化することに対して、S.E. を適用する。

## 4 アルゴリズム

ここでは、S.E. のアルゴリズムについて説明する。S.E. の基本的な動きは次のようになっている。

- (1) 初期パーティションを作成し、 $P_{cur}$  とする
- (2)  $P_{best}$  に  $P_{cur}$  をセーブし  $P_{cur}$  のカット数  $C_{cur}$  を  $C_{best}$  に代入する
- (3)  $count = 0, C_{better} = \infty$  とする
- (4)  $P_{cur}$  に関数 EVOLVE を作用させる
- (5) 新しくできた  $P_{cur}$  のカット数  $C_{cur}$  と  $C_{better}$  と比較する
  - $C_{cur} < C_{better}$  であれば  $count = 0$  とする
  - $C_{cur} < C_{best}$  であれば  $P_{best}$  に  $P_{cur}$  をセーブする
  - $C_{cur} \geq C_{better}$  であれば  $count = count + 1$  とする
- (6)  $count < MaxCount$  ならばパラメータ  $s, t$  をデフォルトにし(4)から繰り返す
- (7)  $C_{better}$  と  $C_{best}$  を比較し
  - $C_{better} < C_{best}$  であれば  $C_{best} = C_{better}, \gamma = \gamma - R$  とする
  - $C_{better} \geq C_{best}$  であれば  $\gamma = \gamma + 1$  とする
- (8)  $\gamma \leq R$  ( $R$  は定数) であればパラメータ  $s, t$  を更新して(3)から繰り返す
- (9)  $P_{best}$  をこの階層の最終的な分割とする

以下、階層的な領域分割、初期パーティション、関数 EVOLVE、パラメータの更新について順に説明する。

## 4.1 階層的な領域分割

Partitioning を行なう際に領域の分割を行なっていくが、ここでは、2分割を階層的に繰り返している。

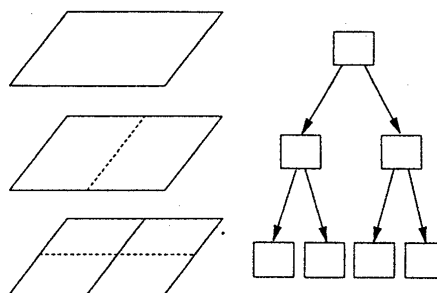


図1: 階層的な分割

図1の示すように、階層の1つ上位の領域を親とする2つの領域に分割する。これを順次繰り返して領域分割を行なっている。

## 4.2 初期パーティション

各階層での初期パーティションは、次のようにして与えている。

1. 親パーティションに含まれるセルのうち固定セルを含まれるべきグループに分配する
2. 残ったフリーセルについて、それぞれのセルを一方に入れた場合のカット数と、もう一方に入れた場合のカット数を求め、この差の大きなものから順に、カット数を小さくする方のグループに分配する
3. この後、白石らのアルゴリズム [2] によってカット数を最小化する

このようにして与えられる分割を S.E. の初期パーティションとしている。

## 4.3 関数 EVOLVE

S.E. の根幹となる関数 EVOLVE についてしめす。

関数 EVOLVE は migration, distribution の 2 つのステップからなっている。

migration では、各セルが現在所属しているパーティションから移動すべきかどうかを決定する。

セル  $c_i$  について

1.  $c_i$  は現在パーティション  $p_j$  に所属している。
2.  $p_j$  と親を共通にするもう一方のパーティションを  $p_l$  とする。
3.  $c_i$  を  $p_j$  から  $p_l$  に移動させた時、減少するカット数を  $c_i$  のセルゲイン  $g_i$  とする。
4.  $g_i$  をランダム整数  $I(I \in [s, t])$  と比較する。
5.  $g_i > I$  であれば  $c_i$  を  $p_j$  から  $p_l$  に移動させ、そうでなければ  $p_j$  に留め置く。

distribution では、migration で移動を決定したセルについて実際に移動した場合に行き先パーティションの面積の制約を満たすものは実際に移動させ、満たさないものについては移動はとりやめる。

これらの migration, distribution の操作をセルゲインの大きなセルから順にスキャンし、一度の EVOLVE の操作では 1 つのセルは 1 度だけ動くことができる。

## 4.4 パラメータ $s, t$ の更新

ここでの更新の対象となるパラメータは migration で用いられるランダム整数  $I(I \in [s, t])$  の乱数の範囲を制御する  $s, t$  であり、 $s = s - H$  ( $H$  は定数) としている。

## 5 実行結果

以下に、本論文の手法と Min-Cut の手法による実行の結果を示す。ここでの実行は定数  $R, H$  をそれぞれ  $R = 10, H = 3$  で行なった。また、パラメータ  $s, t$  のデフォルトはそれぞれ  $s = -1, t = 0$  とした。

使用したチップのデータは表 1 に示す。

表 1: チップデータ

チップ	セル			ネット数
	全数	フリー	固定	
10K	1218	879	339	1602
40K	3901	3046	855	4908
50K	8446	7664	782	11812
100K	9821	8523	1298	12270

これらのチップのデータを用いて、S.E. と Min-Cut の 2 つのアルゴリズムによって論理分割を行ない、同一のパッキングプログラムによって詳細配置を決定する。ここでのラットネストによるマンハッタン長 (M.D.)、及びこの後、同一の配線プログラムによって詳細配線を

表 2: 配線長の比較

	S.E.		Min-Cut	
	M.D.	配線長	M.D.	配線長
10K	235887 (93 %)	266258 (96 %)	252833	276549
40K	1603983 (82 %)	1648574 (82 %)	1957119	1999634
50K	2621643 (92 %)	2718397 (92 %)	2853065	2969737
100K	5036480 (81 %)	5145393 (81 %)	6187234	6371766

行なって、総配線長を求めた。これらの結果を表 2 に示す。また、それぞれの配置プログラムの実行に要した時間を表 3 に示した。

表 2 によると、S.E. は、従来の Min-Cut に比べて、マンハッタン長、実配線長ともに約 5 ~ 20 % 良い結果を得られている。

表 3: 配置実行時間

	実行時間 (min)	
	S.E.	Min-Cut
10K	4.6	2.4
40K	43	18
50K	180	66
100K	380	100

また、表 3 によれば、S.E. は Min-Cut に比べ、約 2 ~ 4 倍の実行時間を必要とする。

## 6 まとめ

本論文では、LSI の Partitioning 問題に対して Youssef Saab らの Evolution-Based Approach を基とした Simulated Evolution を提案し、本手法を用いることにより、従来の Min-Cut の数倍の実行時間がかかるものの、詳細配置後のマンハッタン長、及び詳細配線後の実配線長を約 5 ~ 20 % 縮小できることを示した。

## 参考文献

- [1] B.W. Kernighan and S. Lin "An Efficient Heuristic Procedure for Partitioning Graph" *Bell System Technical Journal*, Vol. 49, Feb. 1970, pp. 291-307
- [2] H. Shiraishi and F. Hirose "Efficient Placement and Routing for Masterslice LSI" *Proc. 17th Design Automation Conference*, June 1980, pp. 458-464
- [3] C.M. Fiduccia and R.M. Mattheyses "A Linear-Time Heuristic for Improving Network Partitions" *Proc. 19th Design Automation Conference*, January 1982, pp. 175-181
- [4] Youssef Saab and Vasant Rao "An Evolution-Based Approach to Partitioning ASIC System" *Proc. 26th ACM/IEEE Design Automation Conference*, 1989, pp. 767-770