

フロアプラン CAD のための階層型マルチエージェントシステム —エージェントの動作機構—*

六本木 陵 武田 有志 大津 金光 横田 隆史 馬場 敬信†

宇都宮大学^S

中村 義彦 須藤 重 峰 正高 鈴木 康弘‡

富士通キャドテック (株) §

1. はじめに

我々は LSI 設計のフロアプラン CAD において、アルゴリズム変更と設計者の知識反映の容易化を目的とした階層型マルチエージェントシステムを研究開発している。LSI 設計におけるフロアプランでは、複数存在する制約を全て考慮して設計を行う必要がある[1]。そこで我々は、制約の追加や入れ換えを柔軟に変更できる動作機構を持つ階層型マルチエージェントシステムのための動作機構を提案する。本エージェントは、各制約を適応度関数として独立に定義し、それらの合成によって行動を決定する。しかし、制約の追加や入れ換えを行う際、制約間での競合[2]が発生するため、新しい制約の追加や削除が困難になっている。そこで競合に対しては、制約を一時的に緩めるために、状態に応じて各適応度関数を変更して対処する。

2. システムの概要

本エージェントシステムは、上位エージェントと下位エージェントによる階層構造である。上位エージェントは、フロアプラン対象である LSI チップにひとつ、下位エージェントは、配置する各回路ブロックにひとつずつ割り当てられる。以下に各エージェントが担当する問題と必要機能について述べる。

2-1. 上位エージェント

上位エージェントはチップ規模の問題を担当する。フロアプランではチップ全体を見渡さないと解決できない問題がある。複数のブロックの処理を連動させて一斉に行う際に、各ブロックまでの結線長の違いにより信号遅延問題が発生してしまう場合がある。そのため、結線長を同程度になるように配置する必要がある。しかし、このような配置方法を行うためには、単一ブロックについての状態の取得や操作だけでは困難であり、ブロック間での関係も調べる必要がある。そこで、全てのブロックの状態から各ブロック同士の関係を調べ、それに応じて複数のブロックを操作する機能を必要とする。

2-2. 下位エージェント

下位エージェントはブロック規模の問題を担当する。フロアプランでは、複数存在する制約を考慮しながら、各ブロックにとっての最適な位置を調べて配置しなければならない。基本的な制約の例として、

- ・指定された範囲内 (チップ上) に配置する

- ・他のブロックとの重なりを回避する
- ・結線長を短くする

が挙げられる。これらの制約の追加や入れ換えを柔軟に行うために、ブロックを配置する予定の位置と、制約を満たす度合いである適応度との関係を表した適応度関数を用いる。ブロックの現在位置から x 軸と y 軸方向における、制約ごとの適応度関数を作成し、全制約を考慮するために論理積で合成した結果を元に行動を決定する機能が必要となる。さらに、複数の適応度関数を合成する際、適応度関数の作成方法によっては、項目間において競合が発生し、制約を満たさない場合がある。そこで、ブロックの状態に応じて各適応度関数を変更し、競合を起していると判断された制約を一時的に緩める機能も必要となる。

3. 各エージェントの動作機構

2. のシステムの概要により、各エージェントの動作機構を次のように設定する。また、各エージェントのフローチャートを図 1 に示す。

3-1. 上位エージェント

上位エージェントは、全下位エージェントの状態を取得して各ブロック同士の問題を調べる。その後、それらの問題に対応して、考慮する制約を各下位エージェントに指定する。

3-2. 下位エージェント

下位エージェントは、各自が担当しているブロックの x 軸と y 軸方向において、上位エージェントで指定された各制約の適応度関数の作成し、それらを論理積で合成する。その後、適応度関数の合成結果において適応度が最大となる方向と位置にブロックを移動する。最大となる位置が複数存在する場合には、現在の位置から最も近い位置に移動する。例として、2-2. で記したフロアプランの基本的な制約における適応度関数の作成方法と合成結果を図 2 に示す。本図は、Block_1 と Block_2 を既に配置されたブロックとして、そこへ Block_2 と結線され

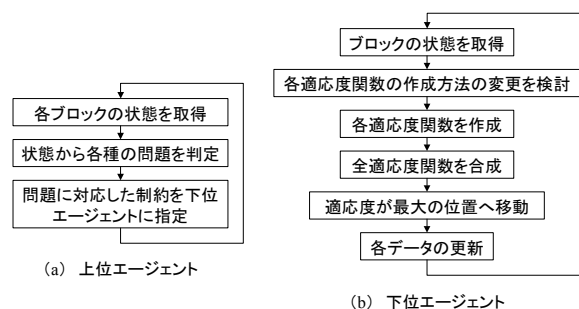


図 1 各エージェントのフローチャート

* A Hierarchical Multi-agent System for Floorplan CAD - Working Mechanism of Agents -

† Ryo Roppongi, Yuji Takeda, Ootsu Kanemitsu, Takashi Yokota, Takanobu Baba

‡ Yoshihiko Nakamura, Shigeru Sudo, Masataka Mine, Yasuhiro Suzuki

^S Utsunomiya University

§ FUJITSU CADTECH LIMITED

た Block_0 を配置する際の適応度関数を表している。横軸が Block_0 の左辺を基準とした配置位置、縦軸は適応度を示す。適応度は 0~1 で表され、満たしていれば 1、満たさないならば 0 である。

しかし、競合によって、x 軸と y 軸方向の全位置における適応度が 0 となり、制約を満たさない場合がある。これを防ぐために、適応度関数内で適応度が 0 から 1 に変化する点において、適応度が 0 の範囲を狭めるように傾斜付けを行う。このとき、傾斜を付ける割合を変更度として表し、これはブロックの状態に応じて変動する。例として、他のブロックとの重なりを回避する制約の適応度関数と変更度の関係を図 3 に示す。

この変更は、与えられた適応度関数に反するものであるため、最終的な変更度は 0 でなければならない。そこで、各制約における前回の行動による適応度と今回の行動による適応度を参照して、適応度関数を変更する。もし、前回と今回の適応度がともに 0 ならば変更度に任意の増加値 α を加算し、0 以外ならば減少値 β を減算する。これにより変更度を 0 に収束させる。図 1 (b) の各データとは、変更度を求めるためのデータ、前回の適応度と変更度を表している。

4. 実験結果

以上で示したエージェントを実装し、図 2 で示した 3 つの制約において、 α と β の値によるフロアプラン結果への影響について実験を行った。実験対象として MCNC ベンチマークの ami33 と ami49 を選び、それぞれ 100 回ずつ実行したときの、チップ内の総結線長と全ブロックを含む最大矩形の面積、平均イテレーション回数と収束率を表 1 と表 2 に示す。ここで、全ての下位エージェントが図 1 (b) のフローを 1 ループした場合を 1 イテレーションとし、さらに、1000 イテレーション以内に、全ての下位エージェントの各適応度が 0 以外、各変更度が 0 で行動を停止した場合を収束したと見なした。

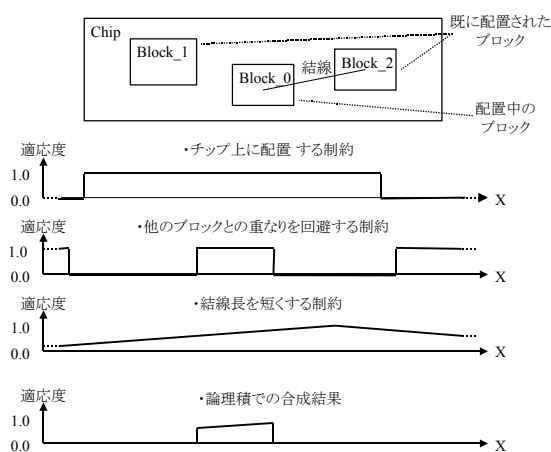


図 2 適応度関数の作成方法と操作方法の例

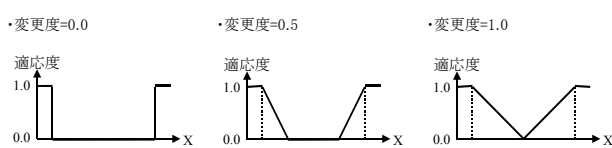


図 3 適応度関数と変更度の関係

表 1 実行結果 (ami33)

増加値 α	減少値 β	結線長 [mm]	面積 [mm ²]	収束率 [%]	平均イテレーション回数
0.00	0.00	124.9	2.13	100	10.0
0.01	0.01	125.0	2.14	73	12.3
0.05	0.01	125.4	2.12	100	11.4
0.10	0.01	126.0	2.11	100	11.5
0.10	0.02	124.3	2.14	100	11.5
0.20	0.04	123.8	2.15	99	11.5

表 2 実行結果 (ami49)

増加値 α	減少値 β	結線長 [mm]	面積 [mm ²]	収束率 [%]	平均イテレーション回数
0.00	0.00	1591.0	53.51	12	11.5
0.01	0.01	1656.5	53.59	4	652.5
0.05	0.01	1617.7	53.78	85	117.4
0.10	0.01	1614.5	53.28	92	90.7
0.10	0.02	1626.5	53.73	89	72.0
0.20	0.04	1595.7	53.53	80	56.6

表 1 では、 α と β による各項目への影響は少ない。これは、ami33 ではブロックとのチップの面積比が大きいので、競合が発生し辛いためである[3]。一方、表 2 では、 α と β の比 α/β が大きい程、収束率とイテレーション回数が良くなっている。これは α/β が高ければ変更度が 0 に収束し易くなるためである。また、 α/β が高くても、 α と β の大きさにより収束率と平均イテレーション回数は変化する。これは、値が小さければ、変更度が変動しにくいいため、収束までのイテレーション回数は増えるが、ブロックの動作が細くなるため収束率は上がる。逆に値が大きければ、変更度が変動し易いため、収束までのイテレーション回数は減るが、ブロックの動作が荒くなり収束率は下がる。

5. おわりに

各制約を適応度関数として定義し、それらの合成によって制約を柔軟に変更できる動作機構を持つ階層型マルチエージェントシステムにおける各エージェントの動作機構について述べた。4. の表 1 により、適応度関数と変更度によってフロアプラン結果の収束率や計算速度に影響を与えることができた。

今後の課題として、上位エージェントの戦略の検討、制約の追加、変更度の自動調整機構の検討などが挙げられる。さらにその上で、実用化に向けた性能向上を図る。

謝辞 本研究は、一部日本学術振興会科学研究費補助金(基盤研究(B)14380135, 同(C)14580362, 若手研究14780186)の援助による。

参考文献

- [1] 白石洋一訳：組合せ最適化アルゴリズムの最新手法－基礎から工学応用まで－，丸善，2002。
- [2] 古川康一訳：エージェントアプローチ人工知能，共立出版，1997。
- [3] 山崎 博之，三上 直人，高橋 篤司：モジュールの重なりを許さない力学的モデルによるモジュール配置手法，情報処理学会論文誌，Vol. 43, No. 5, pp. 1304-1314, May 2002。