

配置配線パズルのための逐次合体アルゴリズムとその展望

鈴木 修平^{†1,a)} 蓮見 平八郎^{†1,b)} 藤吉 邦洋^{†1,c)}

概要: DA シンポジウム 2019 以降のアルゴリズムデザインコンテストの題材である配置配線パズルは集積回路の自動配置配線と親和性が高く、解法に集積回路の自動配置配線アルゴリズムを応用できる。そこで配置しつつ配線を行う階層的配置配線手法を参考にすると共に、既存の平面位相配線手法を拡張して無駄な探索を減らし、解を求める手法を提案する。本稿ではこの手法の中で階層的配置配線手法の拡張を中心に述べる。

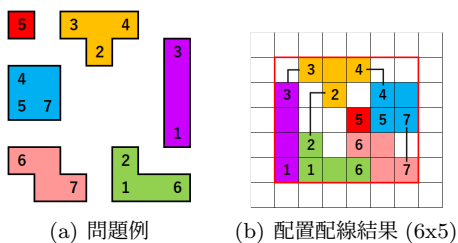


図 1 配置配線パズルの問題例とその配置配線結果

1. はじめに

DA シンポジウム 2019 以降のアルゴリズムデザインコンテスト (ADC) [1][2][3] の題材である「配置配線パズル」という問題は、盤面にブロックを配置し、ブロック上に記入された同じ数字同士を線で結び、全ての数字が線で結ばれて制約違反を起こしていない状態を解とする。図 1 に配置配線パズルの問題例とその解答例を示す。

この問題に対し、同ネット端子間に余裕をもって配置配線した初期解を得た後に同ネット端子間の配線を保ちながらブロックを移動させて配置配線領域の最小化を行う手法 [4] や、複数のソルバ (密接配置ソルバ, QUBO ソルバ, SA ソルバ, SAT ソルバベースソルバ) を並列で実行する手法 [5] が提案されてきた。

配置配線パズルは、集積回路の自動配置配線と親和性が高く、解法に集積回路の自動配置配線アルゴリズムを応用できる。自動配置配線アルゴリズムでは、配置と配線を分けて行うものが一般的である。しかし配置と配線を分ける

と、配置と配線の相互依存性により、満足のいく解を得るためには配置と配線の繰り返しが多回数必要になる。そこで配置しつつ配線を行う階層的配置配線手法 [6][7] が提案された。そこで自動配置配線アルゴリズムを参考に、ブロックから配線を出す方向を“事前に確定させる段階”とそれによって“配置配線を行う段階”に分け、平面位相配線の考えを用いて効率よく探索する手法を提案し、ADC2019・2020・2021 の 3 年連続で優勝した。

本稿ではこの提案手法を説明し、また実際の集積回路設計における配置配線問題ではモジュールが回転可能なことが多いため、回転を含む問題も考える。回転を許容すると組合せが増加し計算時間がかかるためこれを効率的に解く方法も提案する。

2. 配置配線パズルの詳細 [1][2][3]

配置配線パズルとは、ナンバーリンクに配置問題の要素を加えた新たなパズルである。ナンバーリンクとは盤面にある同じ数字同士を線でつなぐペンシルパズルであり、株式会社ニコリの登録商標である [8]。

配置配線パズルは、数字が書かれた複数のブロックが用意されており、ブロックそれぞれにブロック番号が割り振られている。これらを格子状の 1 つの盤面に配置し、ナンバーリンク同様盤面にある同じ数字の間を線で接続するパズルである。盤面のサイズは問題ごとに与えられ、最大サイズは 72×72 である。格子の 1 区画をマスと呼ぶ。問題例とその解答例を図 1 に示す。配置配線パズルのルールを以下に示す。

- (1) 問題で指定された全てのブロックを盤面上に配置する。
 - (a) ブロックの形状は 1 マスのモノミノ、および 4 マスのテトロミノ 5 種のうちいずれかである。
 - (b) ブロックは回転、反転させてはならない。

^{†1} 現在、東京農工大学
Presently with Tokyo University of Agriculture and Technology

a) suzuki@fjlab.ei.tuat.ac.jp
b) hasumi@fjlab.ei.tuat.ac.jp
c) fujiyosi@cc.tuat.ac.jp

- (c) ブロック同士は重なって配置してはならない。
- (2) 盤面上に配置されたブロック上にある同じ数字の間を交差・分岐の無い配線で接続する。(以下ナンバーリンクと同じ)
- (a) 配線の端点が存在するマスは、X,Y方向に隣接する最大4個のマスのうち1個と接続される。
- (b) 配線は1マスに1本のみ引くことができ、交差・分岐してはいけない。
- (c) 配線を構成する端点以外のマスは、X,Y方向に隣接する最大4個のマスのうち2個と接続される。
- (3) 数字が書かれていないブロック内の領域には、線を引けない。
- (4) 解の品質は「全てのブロックと配線を囲む最小の矩形面積」で評価する。

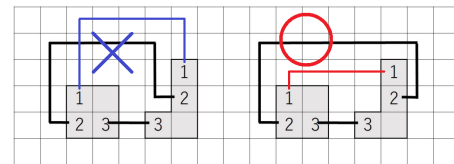


図2 右の多角形の“2”から配線を出す方向によって配線不可能になってしまう例(左:配線不可能, 右:配線可能)

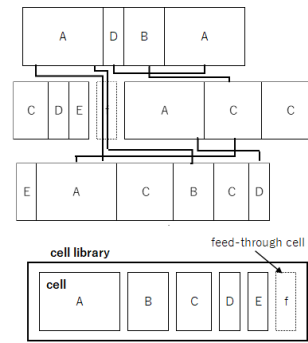


図3 スタンダードセル方式

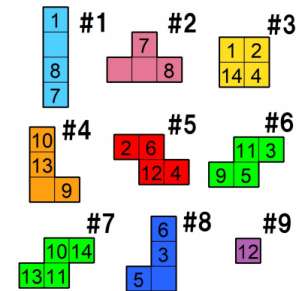


図4 入力されたブロックの例

ナンバーリンクは、集積回路の配線問題と親和性が高いパズルである。ナンバーリンクでは数字マスの位置は固定されているが、配置配線パズルでは数字は盤面上に配置するブロック上に記入される。従って、数字間の配線とブロックの位置を同時に考慮しつつ、できるだけ小さい面積で配置配線を行うことが目標であり、集積回路の自動配置配線と親和性が高いパズルであるといえる。

2.1 配線方向の決定

配置配線パズルの面白い特徴として、配線を出す方向次第では配線不可能になってしまうことが挙げられる。この特徴は配置配線パズルを手動で解くときなどとりあえず配線する手法でよく障害となり、また陥ったときにどの配線が原因なのか分かりづらいという問題がある。ブロックから配線を出す方向を事前に確定させる段階とそれによって配置配線を行う段階に分けて解く場合も、配線方向の決定の仕方によってはどのように配置配線しても他の配線と交差するため、配線不可能になってしまうことが起こりうる。例を図2に示す。この例では、右側の多角形の端子番号“2”から配線を出して左の多角形と結ぶ際に、配線を左側から出すと、次に“3”の配線を行った後に、端子“1”の間の配線が行えなくなってしまうという問題が生じてしまう。

この問題は平面位相配線問題に置き換えることができる。平面位相配線問題とは、各ネットについて全ての端子間を配線同士が交差せず結ぶという平面配線問題において、配線の幅を考慮せずに配線可能性を判定し、可能であれば配線するという問題である。提案手法では事前に拡張CHORD-LAST法[9]やその後に提案された平面グラフを用いる手法[10]を適用し、配線が出る方向を確定させることで解決するためこの問題は取り扱わない。これにより平面位相配線可能性が保証されているものとし、問題が位相

的に配置配線不可能ではないことを前提とする。

3. 関連手法

3.1 階層的配置配線手法 [6][7]

階層的配置配線手法を用いたレイアウトシステム BEAR は、できるだけ小さい面積内に配置配線をするシステムである。BEAR で用いられている階層的配置配線手法では、まずクラスタ成長法という手法を取り入れている。この手法ではいくつかのブロックを選択し、これと接続の強いブロックを次々にボトムアップ的にグループ化していく。この結果全体が1個に統合され、クラスタリング木と呼ばれる木が出来上がる。このクラスタリング木の葉はブロックとなり、末端の階層のクラスタからボトムアップ的に配置と概略配線を行う。同じ階層のクラスタの配置と概略配線が全て終わったら、次にもう一段上の階層のクラスタで配置と概略配線を繰り返す。最終的に全体の配置を得て、詳細配線を行う。

3.2 スタンダードセルの配置配線手法 [11]

スタンダードセル方式は、図3のように高さが統一された矩形のセル(スタンダードセル)をあらかじめ最適設計してライブラリとして登録しておき、それらのセルを用いてチップを設計する。マクロセルの豊富なラインナップと自由度の高い設計手法が特徴で、基本セルを用いるゲートアレイ方式に比べ開発期間は長くなるが、チップサイズが小さく低コストの開発が実現できる。

4. 配置配線パズル [1][2][3] の提案手法

本稿では階層的配置配線手法 [6][7] を参考に、関係の深い2つのブロックを繋ぎ合わせることで合体して1つのブ

表 1 図 5 のグラフに対応する優先度

節点 v_i	a_i	E_{max}	$ E_i $	p_i
v_1	0	2	2	0.22
v_2	1	2	1	1.21
v_3	0	2	3	0.23
v_4	0	2	2	0.22
v_5	0	2	3	0.23
v_6	0	2	3	0.23
v_7	0	2	3	0.23
v_8	0	2	2	0.22
v_9	1	1	1	1.11

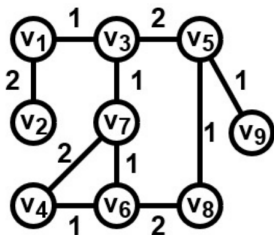


図 5 入力ブロックのグラフ

ロックとし、このブロック上にある未配線の同ネット間を配線する手順を繰り返し行い、最終的に解を得る手法を提案する。これによりブロックが 1 つになり、かつ全ネットの配線が完了していれば配置配線成功となり、そうでなければ配置配線失敗となる。

この配置配線パズルの提案手法の概略を以下に示す。

step0: 入力ブロックのグラフ化 入力で与えられたブロック集合をグラフに変換する。

step1: 優先度の設定 優先して合体するブロックを選択する指針となる、ブロックごとの優先度を決定する。

step2: 合体するブロックの選択 優先度に従って step3 で仮合体する 2 つのブロックを選択する。

step3: ブロックの仮合体 片方のブロックの外周にもう片方のブロックが隣接するように仮合体する。外周 1 周するまで仮合体を繰り返し、配線を含めた外周矩形がより小さい仮合体を保存する。仮合体がすべて失敗した場合、step2 に戻り次点で優先度が大きいブロックの仮合体を行う。

step4: ブロックの合体 保存された仮合体を採用し、実際に合体する。

step5: 優先度の更新 合体したブロックとそれに隣接するブロックの優先度を更新する。残りブロックが 2 つ以上かつ、全てのブロックの優先度が 0 でないなら step2 に戻る。

step6: 残りのブロックの合体 残りのブロックが 2 つ以上なら、ブロックを外周矩形が小さくなるように合体し 1 つのブロックとする。step6 終了時に全ネットの配線が完了しているなら成功、そうでなければ失敗となる。

step0: 入力ブロックのグラフ化

準備として、入力で与えられたブロック集合をグラフに変換し、ブロック番号 i に対し節点 v_i が対応する。同ネット端子を持つ 2 つのブロック i, j に対応する節点間を枝 $e(i, j)$ で結ぶ。 $e(i, j)$ に v_i, v_j 間の同ネット数に等しい重みを与える。 v_i に接続される枝の集合を E_i とする。

図 4 に示したを入力とした時のグラフを図 5 に示す。

step1: 優先度の設定

優先して合体するブロックを選択する指針を以下に示す。

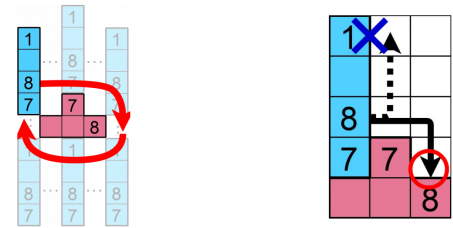


図 6 仮配置 20 通りのうち 6 通りを示した例 図 7 仮配線の例

- 指針 1:** ただ 1 つのブロックとのみ同ネット端子をもつブロックを優先する。
- 指針 2:** 自身以外の 1 つのブロックとの同ネット端子をより多く持つブロックを優先する。
- 指針 3:** 自身と同ネット端子を持つブロックの数が多しブロックを優先する。

この指針に沿って各ブロックの優先度 p を計算し、ブロック i の優先度 p_i を以下の通り設定する。 E_{max} は E_i 内で最大の重みを指す。

$$a_i = \begin{cases} 1 & (|E_i| = 1) \\ 0 & (|E_i| \neq 1) \end{cases} \quad (1)$$

$$p_i = a_i + 0.1 \times E_{max} + 0.01 \times |E_i| \quad (2)$$

図 5 に示したグラフに step1 を適用して得られる p を表 1 に示す。優先度最大の節点に対応するブロックを合体するブロックを選ぶ。

step2: 仮合体するブロックの選択

最も高い優先度 p_i を持つブロック i と、節点 v_i で最大の重みを持つ枝 $e(i, im)$ で繋がる節点 v_{im} に対応するブロック v_{im} を、仮合体するブロックとして選択する。最も高い優先度を持つブロックが複数存在する場合、配線を含めた外周矩形面積が大きいブロックを優先する。それでも決まらない場合、ブロック番号が小さいブロックが優先される。

表 1 の中から step2 で選択されるブロックは、優先度が最も高い節点 v_2 に対応するブロック #2 と、節点 v_2 と最大の重みを持つ枝と繋がる節点 v_1 に対応するブロック #1 である。

step3: ブロックの仮合体

合体後のブロックの配線を含めた最外周矩形面積が小さくなるように仮の合体と配線を繰り返し行う。2 つのブロックが、最低でも 1 点で接し合う全ての位置関係について仮合体と step3.1, 3.2 を行う。

step3 終了時に仮合体したブロックが得られなかった場合、step2 に戻り次点で優先度が大きいブロックの仮合体を行う。仮合体の繰り返しの様子を図 6 に示す。

step3.1: 仮配線

ブロック中に同じネット番号を持つ未配線ネットがなくなるまで次の手順を繰り返す。

仮合体したブロックの、同じネット番号を持つ未配線ネットの内、マンハッタン距離が最も近いネットの片方の子から他方に向け、配線がブロックに沿ってに、右回りと左回りの2方向で経路探索を行う。

経路探索中に配線が目的の端子ではない、未配線の端子の隣にたどり着いた場合、同時に目的の端子にたどり着いたのでなければ、その経路の探索を打ち切る。経路が目的の端子にたどり着いた場合、同時に目的外の未配線の端子の隣にたどり着いたのでなければ、その間を仮配線で結ぶ。経路が目的の端子と目的外の未配線の端子に同時にたどり着いた場合、目的外の未配線端子が配線不可能にならないなら、目的の端子に仮配線を行う。その後、同様に他の未配線ネットの配線を試みる。経路が目的の端子にたどり着けなかった場合、一度そのネットを後回しにし、マンハッタン距離が次に近いネットで仮配線を試みる。仮配線の例を図7に示す。図7ではネット番号“7”のネットが隣り合って存在するため、ネット番号“8”のネット間の配線の経路探索の図になっている。

step3.2: 仮合体の一時保存

今までに仮合体したブロックの中で、ここまでの既配線、仮配線を含めた外周矩形がより小さいものを一時的に保存する。

step4: ブロックの合体

step3で合体した2つのブロックに対応する頂点間の辺を除く、2つの頂点に繋がる辺すべてを、2つの頂点をまとめた頂点に繋ぎ直す。

step5: 優先度の更新

step4で2頂点が1つにまとめられたため、1つにまとめられた頂点とそれに隣接した全ての頂点の優先度 p を式(2)のように設定し直す。またstep3で一時的に値を0に変更した優先度を設定し直す。未配線ネットが残っていて、かつ全ての頂点の優先度が0でなければstep2に戻る。

step6: 残りのブロックの合体

配線が完了していないネットがある場合、配置配線失敗と出力して終了する。全ネットの配線が完了していて、かつブロックが2つ以上残っている場合、ブロックが1つになるまで、配線を含めた外周矩形が最小になるように合体を行う。全ネットの配線が完了していて、かつブロックが1つである場合、配置配線が完了した結果を出力して終了する。

5. 回転を考えた場合の拡張手法

図1(a)の問題の回転を許容した場合の配置配線した解答例を図8に示す。接続する順番が対応する枝の重みで決まる提案手法をそのまま回転を考えた場合に拡張すると、図9のように部分的な最小矩形面積を求めた結果、最終的に解が悪くなることもある。この問題を解決するにあたって、高集積で高性能なLSIが期待できるスタンダードセル

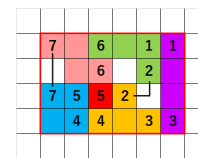
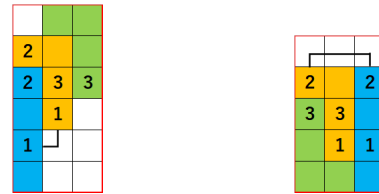


図8 図1(a)の問題の回転を許容した場合(6x4)



(a) 配置配線パズルの提案手法の解 3x6 (b) 回転を考えた場合の拡張手法の解 3x5

図9 接続方法の違いによる改善の例

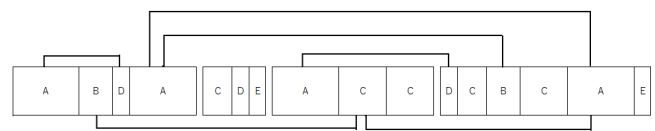
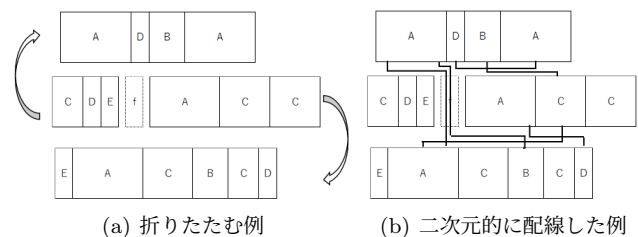


図10 スタンダードセルの一次元的な配置例



(a) 折りたたむ例 (b) 二次元的に配線した例

図11 図10の一次元的な配置を3行に折りたたんだ配置配線

方式[11]を参考にした。

スタンダードセルの一般的な配置配線手法[12]では配線長最短を優先とし、高さを揃えているため横に長く繋げて解を得る(図10)。そこで、得られた解から図11のように配置配線を折りたたむことで配線長の改善を図ることができると考えた。この動作は回転ができる場合の配置配線パズル[1][2][3]と非常が相性がよく、同様に解くことで横に細長い解が得られる。このとき、折りたたんだ後の解が指定された縦横比に近づくように一边を揃えると都合がよいのでこの一边を基準線とし、それより上に接するように配置していく。

回転を考えた場合の拡張手法では1つのブロックを種ブロックとし、他のブロックを繋げて成長させながら配置配線し、1つの解を得る。

この回転を考えた場合の拡張手法の概略を以下に示す。

step1: 種ブロックの生成と成長

step1-1: 種ブロックの生成 端子のあるブロックのうち、端子数最小のブロックの1つを種ブロックに選び、下方向に配線を伸ばせる端子数が最も少なくなるよう回転し配置する。

step1-2: 配線するネットの選択 基準線に接する辺から反時

計回りに種ブロックの外周を走査して配線するネットを選択する。

step1-3: ブロックの仮合体 回転したすべての向きで仮合体し、配線するネットの配線長が最短となり負の y 座標を含まない組み合わせの中から、評価値が最大の仮合体を選ぶ。評価値は合体するブロックの y 座標最小値とブロックと配線を囲う矩形面積が小さいほど大きく設定する。

step1-4: ブロックの合体と種ブロックの更新 評価値最大で合体し、合体したブロックと配線を新たな種ブロックとして更新し、step1-2に戻る。

step2: 種ブロック内の未配線ネットの配線 種ブロックの外周に沿うように配線を行う。

step3: 残りのブロックの合体 残りのブロックが2つ以上なら、ブロックを外周矩形が小さくなるように合体し1つのブロックとする。

step4: 解の保存 得られた解を保存する。

step5: 種ブロックの再配置配線と解の更新 種ブロック内で配線長最長の配線が最短になる位置にブロックを再配置し再配線する。解が改善された場合のみ解を更新し、step3に戻る。

step1: 種ブロックの生成と成長

step1-1: 種ブロックの生成

端子のあるブロックのうち、端子数最小のブロックの1つを種ブロックに選び、下方向に配線を伸ばせる端子数が最も少なくなるよう回転し配置する。以降、このブロックの最下端に接する直線を基準線 ($y = 0$, 上向き正) とする。端子数最小のブロックが複数存在する場合、ブロック番号が小さいブロックを優先する。

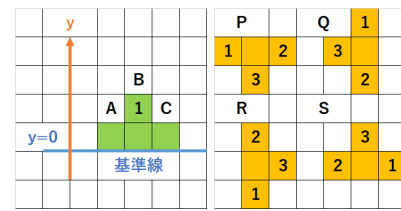
step1-2: 配線するネットの選択

基準線に接する辺から反時計回りに種ブロックの外周を走査してネットを選択する。同ネットの他方の端子が他のブロックにあるなら、そのネットの種ブロック側の端子 α と他のブロック側の端子 β を次に配線するネットに選び step1-3に進む。同ネットの他方の端子が同じ種ブロック内にあるなら、その配線は後の step2で行う。

次に配線するネットの候補が無く他のブロックに未配線端子がまだ残っているなら step1-1に戻り別の種ブロックを生成する。次に配線するネットの候補が無く他のブロックに未配線端子が残っていないなら、step2に進む。

step1-3: ブロックの仮合体

端子 β を持つブロックを B とし、ブロック B を90度ずつ回転した4通りの回転すべてに対して、端子 α と端子 β の配線長が最短となるように種ブロックとブロック B を仮合体を試す。仮合体は以下の条件をすべて満たす場合に限定する。



(a) 種ブロックと仮 (b) ブロック B の回
合体する位置 転パターン

図 12 種ブロックにブロック B を仮合体する例

表 2 図 12 のネット 1 を合体させるときの評価値計算

仮合体する位置 i	回転パターン j	評価値 e_{ij}
B	P	$-1^2 - 4 * 3 = -13$
A	R	$-1^2 - 3 * 4 = -13$
A	S	$-1^2 - 5 * 3 = -16$
B	R	$-2^2 - 3 * 5 = -19$
B	S	$-2^2 - 4 * 4 = -20$
C	P	$-0^2 - 5 * 2 = -10$
C	R	$-1^2 - 4 * 4 = -17$

- 条件 1: 配置制約と配線制約を満たす。
- 条件 2: 未配線端子の接続可能性を無くさない。
- 条件 3: 合体するブロック B を基準線より上 ($y \geq 0$) で仮合体する。

この条件を満たす中で端子 α と端子 β のマンハッタン距離が最短となる位置で仮合体を行う。仮合体の候補が複数ある場合、端子 β と置く位置 i が重なるようにブロック B を重ねる。4通りの回転パターンを j とし、各仮合体にそれぞれ評価値 e_{ij} を計算し、評価値 e_{ij} が一番大きい仮合体を採用し合体する。評価値 e_{ij} は以下の通り設定する。 $\min Y$ はブロック B の各マスが持つ y 座標の最小値を、 $\min S$ は仮合体を囲む最小矩形面積を表す。

$$e_{ij} = -\min Y^2 - \min S \quad (3)$$

図 12(a) の種ブロックに 12(b) のブロック B を仮合体するとき、step1-3を適用して得られる評価値を表 2 に示す。評価値を計算した結果、C の位置に回転パターン P を置いた仮合体で実際に合体することとなる。

step1-4: ブロックの合体と種ブロックの更新

評価値最大で合体し、合体したブロックと配線を新たな種ブロックとして更新し、step1-2に戻る。種ブロックが成長し終えた場合、step2に進む。

step2: 種ブロック内の未配線ネットの配線

種ブロックの未配線端子は同じ種ブロック内に同ネットの他方の端子を持つため、個々の種ブロックに対して、外周で隣り合う端子から順に種ブロックの外周に沿うように配線を行う。配線経路は配置可能性を極力減らさないために、種ブロックとその配線の外周 1 マスを縁取り、その中で幅優先探索を行う。

表 3 提案手法と若泉らの手法 [5] の配置配線結果の比較

ADC2020 問題番号	ブロック数 (端点なし数)	ネット数	非連結な ブロック群数	提案手法		若泉らの手法 [5]	
				面積	時間	面積	時間
Q1	6(0)	5	1	35	0 秒	35	0 秒
Q8	6(0)	12	1	88	1 秒	誤答	8 秒
Q12	100(30)	60	44	624	1 秒	936	241 秒
Q41	2(0)	4	1	16	0 秒	16	0 秒

step3: 残りのブロックの合体

step2 終了地点で未配線端子はないため、ここで解が得られてない場合は、種ブロックを用いない未配線端子に頼らない解法が追加が必要となる。残りのブロックが2つ以上なら、ブロックを外周矩形が小さくなるように合体し1つのブロックとする。

step4: 解の保存

得られた解を保存する。

step5: 種ブロックの再配置配線と解の更新

種ブロックを成長させた逆順にブロックを外し、外側の配線のうち最長の配線長が最短になる位置に再配置配線し、step2 と同様に再配置配線による未配線端子を配線する。外周矩形面積を計算し、解が改善されなくなるか候補がすべて再配置配線済みになるまで繰り返す。既に保存されている場合は解が改善された場合のみ解を更新する。

6. 計算機実験

メモリ 16GB, インテル (R) Core(TM) i7-10700K プロセッサ (3.80GHz), Windows 10 の環境で提案手法を C 言語で実装し, ADC2020 の問題に対して配置配線パズル [1][2][3] の提案手法を適用したところ, 41 問中 23 問で解を得る事ができた。他方, 若泉らはこのうち 4 問の解について記していた [5] ため, 比較を表 3 に示す。ここで「非連結なブロック群数」とはネットで繋がれたブロックのまとまりの数である。比較に用いた問題それぞれの特徴を以下に示す。

Q1: ブロックと配線の数が少なく人間が容易に解ける問題

Q8: ブロック数は少ないが, 配線が非常に困難な問題

Q12: ブロック数が多い問題

Q41: 配線は複雑だが, 人間が容易に解ける問題

これらすべての問題に対して, 若泉らの手法と同じかより良い結果を短時間で得ることができた。

また配置配線パズルの提案手法と回転を考えた場合の拡張手法の比較実験を行った。ADC2019 の入力に対して適用し求めた外周矩形面積を配置配線パズルの提案手法と比較した。結果を表 4 に示す。ブロック数の少ない問題で解の改善が確認できた。しかし, ブロック数の多い問題では再配置配線を繰り返しても外周矩形面積を小さくできないことがあった。

表 4 配置配線パズルの提案手法で得られた面積#NoRot と回転を考えた場合の拡張手法で得られた面積#Rot の比較

ADC2019 問題番号	ブロック数 (端点なし数)	ネット数	#NoRot	#Rot
Q3	2(0)	4	56	8
Q4	3(0)	3	30	16
Q5	4(0)	4	24	16
Q6	4(0)	4	16	24
Q23	20(1)	25	100	171

7. おわりに

本稿では配置配線パズルを解くため, 階層的配置配線手法を参考にし平面位相配線の考えを用いて無駄な探索を省く手法を提案し, 平面位相配線手法の拡張を中心に説明した。この方法に加えて, スタンダードセルの配置配線手法を参考にし, 回転を含む問題の手法も提案した。

参考文献

- [1] “アルゴリズムデザインコンテスト 2019 (ADC2019) ルール説明”, 2019/4, <https://dasadc.github.io/adc2019/>
- [2] “アルゴリズムデザインコンテスト 2020 (ADC2020) ルール説明”, 2020/7, <https://dasadc.github.io/adc2020/>
- [3] “アルゴリズムデザインコンテスト 2021 (ADC2021) ルール説明”, 2021/6, <https://dasadc.github.io/adc2021/>
- [4] 和田ら, “グラフの位相埋め込みの配置配線パズルへの適用に関する一検討”, Vol.2019-SLDM-189 No.31, pp.1-6, 2019
- [5] 若泉ら, “多様なアルゴリズムを用いた配置配線パズルの協調システム”, VLD2020-35, pp.132-137, 2020
- [6] W. M. Dai and E. S. Kuh, “Simultaneous floor planning and global routing for hierarchical building block layout”, IEEE Trans. CAD, vol. 6, No.5, pp.828-837, 1987.
- [7] W. M. Dai, B. Eschermann, E. S. Kuh, M. Pedram, “Hierarchical placement and floorplanning for BEAR”, IEEE Trans. CAD, vol. 8, pp. 1335-1349, 1989.
- [8] “ナンバーリンクの遊び方, ルール, 解き方—WEB ニコリ” 2019/7, <https://www.nikoli.co.jp/ja/puzzles/numberlink/>
- [9] 甘利ら, “配置配線パズルにおける求解アルゴリズム”, DA シンポジウム 2019 論文集, pp.51-56 (2019)
- [10] 甘利ら, “配置配線パズルのための平面グラフを用いた解法”, DA シンポジウム 2020 論文集, pp.104-109 (2020)
- [11] 今井正治, “ASIC 技術の基礎と応用”, 電子情報通信学会 (1994)
- [12] 国枝博昭, “集積回路設計入門”, コロナ社 (1996)