Vol.2019-SLDM-189 No.31
Vol.2019-EMB52 No.31
2019/11/15

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

和田 邦彦[†1,a)]　大和田 真由[†1]　山本 克治[†1]　堀本 遊[†1]　佐藤 真平[†1]　高橋 篤司[†1]

概要：ADC2019 で題材となった配置配線パズルは，盤面上にポリオミノを配置しポリオミノ間のネットを交差なく配線するルールであり，実際の集積回路設計における配置配線問題と非常に親和性が高い．
本研究では問題をグラフ描画問題とモデル化し，グラフの位相埋め込みを考慮して平面描画を行うことでポリオミノ間ネットの配線を行った後，配線済みの状態を保ちながら最良な配置を探索する求解手法を提案する．また，提案手法による求解が効率的に良質な解を得られることを計算機実験で示す．

1. はじめに

集積回路設計における電子部品の配置配線問題は古くから広く研究されており，様々な目的に対して多くの手法が開発されてきた．なかでも，回路面積の最小化を目的とした配置配線問題は，歩留まりの向上，製造コストの削減，配線長を最小化することによる配線遅延の削減などの理由から，その重要性は極めて高い．

DA シンポジウム 2019 で開催されたアルゴリズムデザインコンテスト 2019(ADC2019) で題材となった問題は，前年までのナンバーリンク[*1]を題材とした配線問題に配置問題の要素が加わり，実際の集積回路設計における配置配線問題とより親和性の高い問題なった．前年まで題材となっていたナンバーリンクは，盤面上に位置が固定されたピン同士を配線し，総配線長や総折れ曲がり数などによって定義される解の品質をなるべく高めることを目的としていた．対して，ADC2019 ではセルにピンが定義されたポリオミノのセットが与えられ，ポリオミノの配置とピンの配線を行い配置配線領域の最小化を目的としている．ピンが定義されたポリオミノは，ネット端子を持つ集積回路上の機能モジュールの単純化モデルとみなすことができ，ADC2019 問題を効率的に解くシステムの開発は実際の集積回路設計における配置配線の効率化に大いに貢献すると期待できる．

ナンバーリンクが題材となった前年のコンテストでは，東工大チームは集合対間配線による初期解の生成と引きはがし再配線による解の改良を繰り返す手法 [1] で効率的な解法を示した．ADC2019 問題においても，ポリオミノの配置を決定することで，盤面上に障害物のあるナンバーリンク問題とみなせ，前年の解法を適用できる．しかし，前年の解法ではピンは全ネットが配線可能なように配置されていると仮定していたのに対し，ADC2019 問題ではポリオミノの配置で決まるピンの位置によって配線不可能なネットが生じる場合がある．一般に全ネットの配線可能性を確かめることは難しく，また，配線可能性を考慮しながら配置面積を最小化するような配置を得ることは非効率的である．

本稿では，前年の解法の問題点を解決するため，最初にポリオミノ間の接続要求をすべて満たした配置配線を初期解として得た後に，ポリオミノ間の接続を保ちながらポリオミノを移動させて配置配線領域の最小化を行う手法を示す．提案手法では，求解の第一段階でポリオミノを点，ポリオミノ間の接続要求を辺とみなしたグラフの平面位相埋め込みを生成し，それに基づいてポリオミノの配置配線の最小化を行う．グラフの平面位相埋め込みは，点回りの辺の順序で一意に決まる．ADC2019 問題ではグラフの元となるポリオミノの形状により，考慮すべき平面位相埋め込みの数が一般のグラフと比べて大きく削減できる．そこで，すべての平面位相埋め込みを探索して最良解を得ることを試みる．配置配線領域の面積最小化は，グラフの各連結成分に対して行う．各連結成分に対しての配置配線の最小化は，既存のグラフ描画アルゴリズムを用いてグラフの直交描画を行い，描画されたグラフをポリオミノと配線に変換し初期解とする．さらに，初期解に対して，モジュールの重なりを許さない Force-Deirected Relaxation(FDR) 法 [2] を基にした手法で解を改良する．

[†1] 現在，東京工業大学
Presently with Tokyo Institute of Technology
[a)] wada@eda.ict.e.titech.ac.jp
[*1] 「ナンバーリンク」は株式会社ニコリの登録商標です．

Vol.2019-SLDM-189 No.31
Vol.2019-EMB52 No.31
2019/11/15

計算機実験では，ADC2019 で出題された問題を提案手法によって求解しその結果を解析することで，探索が効率的に行え良質な解が得られることを確認した．

以下，2 章で以降の議論のための準備として ADC2019 のルールの詳細のおよび本稿におけるグラフの位相埋め込みについて説明する．3 章では提案手法を解説し，4 章では実験によって提案手法の性能を評価する．最後に，5 章で結論を示す．

2. 準備

2.1 ADC2019 ルール

本節では，本稿で対象とする ADC2019 問題のルールを説明する．

問題の入力，出力，目的は以下の通りである．

入力 盤面サイズ，ポリオミノのセット
出力 ポリオミノの配置，ピン間の配線
目的 ポリオミノと配線をすべて含むバウンディングボックスの面積の最小化

ポリオミノは複数の正方形を隣合うように繋げた多角形である．本稿では，ポリオミノを構成する最小単位である正方形をセルと呼ぶ．1 つのセルから成るポリオミノをモノミノ，4 つのセルから成るポリオミノをテトロミノと呼ぶ．鏡像体を区別し，回転体を同一視するとモノミノは 1 種類，テトロミノは 7 種類存在する．7 種類のテトロミノは一般に，形状が類似するアルファベットから O,I,T,S,Z,J,L ミノと呼ばれ，本稿でもこの呼称を用いる．

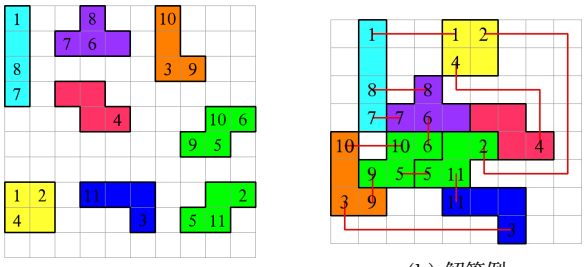
入力として与えられるポリオミノの種類は 7 種類のテトロミノとモノミノの 8 種類である．また，ポリオミノは向きも指定され，回転および反転は許されない．ポリオミノにはセル上に 0 個以上の数字が与えられる．このようなセルをピンセルと呼ぶ．入力として与えられる全ポリオミノ中には必ず同一の数字が与えられたピンセルが 2 つずつ存在する．

ADC2019 問題では，盤面上にポリオミノを配置し，同数字のピンセル同士を配線する．以降，同数字のピンセルの接続要求をネットと呼ぶ．ポリオミノのセルと盤面のマスは同サイズであり，ポリオミノはセルが完全に盤面のマスに重なるように配置する．一般のナンバーリンクと同じように，配線は盤面の 1 マスに 1 本のみ縦横に引かれ，交差および分岐は許されない．また，同数字のピンセル同士が隣合っている場合，それらのネットは配線されているとする．ADC2019 問題は盤面上にポリオミノの配置配線を行うため，配置配線領域が盤面サイズを超えた場合は無効な解とする．

盤面サイズは最大で 72×72 である．

問題の目的はポリオミノの配置配線領域の面積の最小化である．解の品質はポリオミノと配線を含むバウンディングボックスの面積で評価される．

図 1 に問題例と有効な解の 1 つを示す．



(a) 問題例　(b) 解答例

図 1 ADC2019 問題

2.2 グラフの位相埋め込み

本節では本稿におけるグラフの位相埋め込みの定義および，平面位相埋め込みに関する ADC2019 問題の性質の説明をする．

本稿では，グラフ描画がグラフの点と辺が二次元平面上の座標と対応付けられた状態とする．対して，グラフの位相埋め込みは二次元平面上の点や辺の位相的な位置関係を表す．また，平面位相埋め込みは，二次元平面上で辺の交差が無い平面グラフの位相埋め込みである．

グラフの全ての点の点周りの辺の順序を決めると平面位相埋め込みが一意に決まる．そのため，本稿では点回りの辺の順序でグラフの平面位相埋め込みを表現する．

本稿で示す提案手法では，ポリオミノとピンセルの接続要求をそれぞれ点と辺とみなしたグラフを作成し，その全ての平面位相埋め込みの探索を試みる．

一般のグラフでは，グラフの全点の点周りの辺の順序の決め方はグラフの点数乗オーダーとなるため，点数によっては平面位相埋め込みの数が膨大である．しかし，ADC2019 問題では，点の元となるポリオミノの形状に注目すると，ポリオミノの種類によっては点周りの辺の順序が固定されることがわかる．例えば，図 2 に示す S ミノは，その形状から，点とみなしたときの点回りの辺順序は $(1,2,3,4)$ の 1 通りになる．Z,T,O ミノおよびモノミノも同様に点回りの辺順序は 1 通りになる．

対して，図 3 に示す L ミノは，1 の配線を行う方向を考慮すると，点とみなした時の点回りの辺順序は $(1,2,3,4),(2,1,3,4)$ の 2 通りになる．一般のグラフの次数 4 の点の点回りの辺順序は 6 通りであり，考慮すべき場合が少ない．I, J ミノも同様に点回りの辺順序は一般のグラフの点より点回りの辺順序の場合の数が少ない．

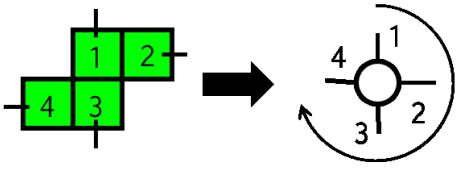


図 2 S ミノの辺順序

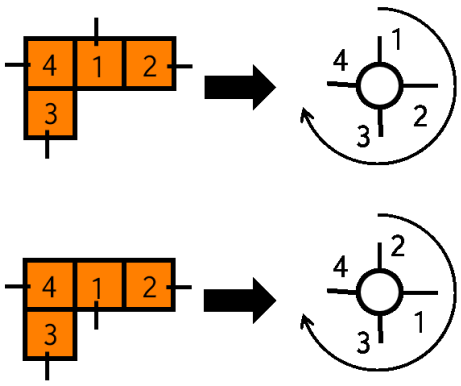


図 3 L ミノの辺順序

以上の性質から，ADC2019 問題から作成するグラフは一般のグラフと比べて位相埋め込みの数が少なく，提案手法による高速なイテレーションによって平面位相埋め込みの全探索が妥当な時間で終了すると考える．

3. 提案手法

3.1 全体概要

本節では，提案手法の全体概要を説明する．

図 4 に提案手法のフローチャートを示す．(a) は全体の流れを示し，(b) は (a) で呼ばれるサブルーチンを示している．

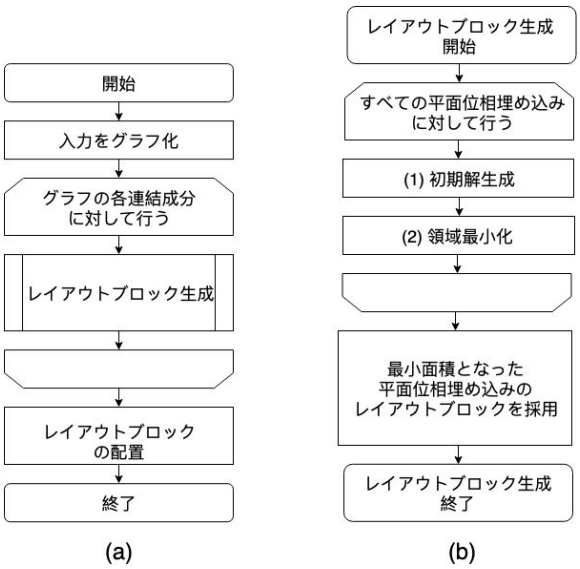


図 4 提案手法

提案手法では，まずポリオミノを点，ポリオミノ間の接続要求を辺とみなしてグラフを作成する．

次のフローでは作成したグラフの各連結成分に対してループ内の処理を行い，最小レイアウトブロックを作成する．ただし，本稿においてレイアウトブロックとは，相対的な位置関係とピンセル間の配線路が固定された連結なポリオミノと配線を指すこととする．

図 4(b) で示すサブルーチンでは，入力として与えられる連結グラフのすべての平面位相埋め込みに対してループ内の処理を行う．ループ内の処理は初期解の生成と領域最小化の 2 段階に分かれている．

第一段階目では，グラフの平面位相埋め込みを基に，グラフ描画アルゴリズムを用いてグラフの直交描画を得る．さらにグラフの直交描画において点をポリオミノに，辺を配線路に戻すことで，レイアウトブロックを初期解として生成する．このフローについては次節で詳細に説明する．

第二段階では，重なりを許さない FDR 法を基にした手法を用い，前段階で生成したレイアウトブロックの面積の最小化を試みる．FDR 法においてポリオミノの位置を移動させる際，新たなポリオミノの配置によって接続不可能なネットが生じる移動を禁じることで全ネットの接続状態を保たれることにより，ネットの配線可能性を考慮することなくより良いなポリオミノの位置を探索できる．このフローについては次々節で詳細に説明する．これらの処理により，各平面位相埋め込みに対してレイアウトブロックを生成され，それらのうち面積が最小なレイアウトブロックを最小レイアウトブロックとして採用する．

2.2 で説明した ADC2019 問題の性質から，考慮するべき平面位相埋め込みの数が少なく (b) ループ回数はそう多くないと考えられるが，問題によっては，I,J,L ミノが多いことにより，平面位相埋め込みの数つまり (b) のループ回数が膨大化することが考えられる．そのため，(b) のループ回数には上限を設け，上限を超えたらループをアボートし，それまでに得られたレイアウトブロックから最小レイアウトブロックを採用する．

以上の処理によりグラフの連結成分と同数の最小レイアウトブロックが得られ，それらを入力で定義されたサイズの盤面上に配置して結果を出力する．

図 5 に求解の経過を示す．ただし，図 5 中の赤数字はポリオミノ番号，黒数字はネット番号を表す．

(a) は入力で与えられたポリオミノのセットである．(b) では入力をグラフに変換した結果を表しており，連結成分が 2 つであることが確認でき，(c) から (e) まではこれら各連結成分に対しての結果を示す．(c)，(d) は図 4(b) 中 (1) の処理による経過を表している．(c) はグラフの直交描画結果を表しており，(d) は (c) をポリオミノと配線に変換した結果を表している．(e) では図 4(b) 中 (2) の処理により (d) の各レイアウトブロックの領域が最小化され，最小レイアウトブロックが得られていることを確認できる．最後に，(f) は最小レイアウトブロックが全体の配置配線領域が最小になるように配置された結果を表している．

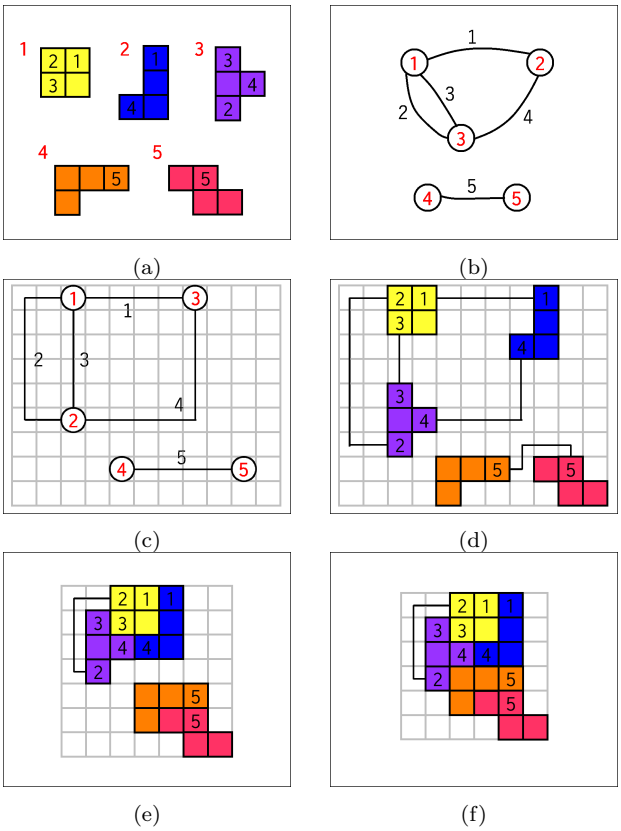


図 5 手案手法の経過

3.2 レイアウトブロックの初期解生成

本節では，図 4(b) 中 (1) に示された，レイアウトブロックの初期解を生成する処理について説明する．

この処理では，入力としてグラフの平面位相埋め込みが与えられ，それを基にレイアウトブロックを生成する．

まず，入力として与えられた平面位相埋め込みに基づき，グラフの平面上の描画を行う．グラフの点はテトロミノまたはモノミノを点とみなして作成される．よって，グラフの点の次数は高々 4 であり，グラフの直交描画が可能である．グラフの直交描画には，最小コストフロー問題の解法利用したアルゴリズム [3] を用いた．[3] のアルゴリズムでグラフの直交描画を行うには，入力としてグラフの平面位相埋め込みに加えて，外閉面を指定する必要がある．外閉面の選択は平面位相埋め込みの各面を解析し，外閉面として最適な面を推測して決定する方法が考えられるが，ADC2019 問題では問題サイズが限られているため，提案手法では各閉面を外平面とした場合の全探索を行う．次章の実験で用いた提案手法の実装でも，外閉面は全探索した．

3.3 レイアウトブロック最小化

本節では，図 4(b) 中 (2) に示された領域最小化について説明する．

レイアウトブロック最小化処理では，入力として与えられたレイアウトブロックのポリオミノの配置と配線を変更して面積の最小化を行う．

レイアウトブロック最小化処理は，モジュールの重なりを許さない FDR 法 [2] を調整した方法を用いる．本節で示す手法も，[2] で示された手法と同様に，ポリオミノ間の配線を力学的バネとみなし，バネによって引かれる力の方向にポリオミノを逐次的に動かし，ポリオミノが力学的安定点にある配置を探索する．

ポリオミノの移動は配線の接続可能性を保ちながら行う．ポリオミノを移動させる際，そのポリオミノに接続されている配線をすべて引きはがす．また，ポリオミノの移動先となるマスに配線がある場合は，その配線を引きはがす．ポリオミノの移動後に引きはがしたすべての配線の再配線を行う．再配線に失敗した場合はポリオミノおよび引きはがした配線をすべてポリオミノの移動前の状態に戻し，ポリオミノの移動の失敗とする．

ポリオミノの移動先にほかのポリオミノが配置されている場合，ポリオミノを移動前の状態に戻し，ポリオミノの移動の失敗とする．

ポリオミノ移動後のレイアウトブロックの面積が移動前より増大していた場合，ポリオミノを移動前の状態に戻す．以降の説明の便宜上，この場合もポリオミノの移動の失敗と呼ぶ．

ポリオミノの一度のイテレーションでの移動は，マス回りの 8 方向のうちどれか 1 方向に 1 マスとする．マスの移動方向は優先度 1 から 3 までの 3 方向を決め，優先度 1 の方向への移動に失敗した場合，優先度 2 の方向へ移動し，優先度 2 の方向への移動に失敗した場合，優先度 3 の方向へ移動する．なお，優先度 3 の方向への移動に失敗した場合は，ポリオミノは移動を行わない．

同数字のピンセル P_1, P_2 の座標をそれぞれ $(x_{p1}, y_{p1}), (x_{p2}, y_{p2})$ とするとき，P_1 に働くバネによる力ベクトルを $(x_{p2} - x_{p1}, y_{p2} - y_{p1})$ とする．そして，ポリオミノに働く力ベクトルをポリオミノ上のすべてのピンセルに働く力ベクトルの総和とする．ポリオミノに働く力ベクトルが (f_x, f_y) のとき，ポリオミノの移動ベクトル (v_x, v_y) を以下のように決める．

ポリオミノの一度のイテレーションでの移動は，マス回りの 8 方向のうちどれか 1 方向に 1 マスとする．移動方向は優先度 1 から 3 までの 3 方向を決め，優先度 1 の方向への移動に失敗した場合，優先度 2 の方向へ移動し，優先度 2 の方向への移動に失敗した場合，優先度 3 の方向へ移動する．ただし，移動失敗の条件は後で示す．なお，優先度 3 の方向への移動に失敗した場合は，ポリオミノは移動を行わない．優先度 1, 2, 3 の移動方向の移動ベクトル v_1, v_2, v_3 は以下のように決める．

$$v_1 = \begin{cases} (\frac{f_x}{|f_x|}, 0) & (|f_x| \geq |f_y|) \\ (0, \frac{f_y}{|f_y|}) & (|f_x| < |f_y|) \end{cases}$$
$$v_2 = \begin{cases} (0, \frac{f_y}{|f_y|}) & (|f_x| \geq |f_y|) \\ (\frac{f_x}{|f_x|}, 0) & (|f_x| < |f_y|) \end{cases}$$
$$v_3 = (\frac{f_x}{|f_x|}, \frac{f_y}{|f_y|})$$

以上の処理によるポリオミノの移動を全ポリオミノに対して 1 度行うことを 1 イテレーションとする．レイアウトブロックの最小化では，イテレーションを，1 イテレーション中にレイアウトブロックの面積が減少しなくなるまで繰り返す．

4. 実験

実験では，ADC2019 問題を提案手法を実装したプログラムで解き，その結果を解析することで提案手法の性能を評価する．提案手法を実装し，ADC2019 で出題された問題を求解しその結果を解析する．実装には C++言語を用い，メモリ 32GB，Intel(R) Core(TM) i7-2700K CPU3.50GHz CPU，Ubuntu 16.04 の環境を用いた．また，グラフ描画アルゴリズムで最小コストフロー問題を解く際には Google's OR-tool[4] を用い，バックエンドは数理最適化ソルバの SCIP[5] を利用した．

本実験で使用した ADC2019 で出題された問題のうち 7 問についての結果を表 4 に示す．#Minos は配置するポリオミノの数，#Nets はネット数，Grid は盤面サイズを表す．また，#Connected は連結成分の数は表す．#Embeddings は連結成分ごとの位相埋め込みの数の総和である．#Dencity は問題の密度を表し，密度はポリオミノのセル数の総和を盤面のマス数で割った値とする．つまり，密度が大きいほど盤面上のレイアウトブロックの配置の自由度が少ない．Solved は有効な解が得られたかどうかを示す．BB_{size} は提案手法プログラムが出力した配置配線を囲むバウンディングボックスのサイズを表す．よって，BB_{size} が Grid に収まっている場合，有効な解が得られている．$T[sec]$ は求解にかかった時間を秒で表す．C_{total} はポリオミノの配置と配線に使われたマス数を表す．そして，C_{route} は配線に使われたマス数を表す．

実験の結果，Q25 以外の 6 問で有効な解が得られた．また，すべての問題でネットの配線に成功した．

表 4 から，有効な解が得られた問題は，いずれも計算時間は短く，Q11 以外は 1 秒以内で解が得られた．Q11 は，すべてのポリオミノが 4 つピンセルがある I ミノであるため，位相埋め込みの数が他の問題と比較して著しく多く，計算時間が大きくなった．

C_{total} と C_{route} に注目する．C_{total} に対する C_{route} の割合は，レイアウトブロックに対する配線の占める割合に等しい．表 4 の結果から，C_{route} の割合はいずれも 0.3 以下である．この事実から提案手法ではレイアウトブロックの最小化が効率的に行えることがわかる．

Q25 は，C_{route} の割合がもっとも少ないが，レイアウトブロックを盤面上に収めることに失敗した．これは，本実験ではレイアウトブロックの配置に単純な BL 法を用いたためであると考えられる．レイアウトブロックの配置に BL 法を用いると，レイアウトブロックが矩形に近い形状ではない場合，盤面上の未使用の領域の断片化が起こり，盤面上が疎となる．これにより，密度が比較的大きい問題である Q25 の求解に失敗したと考えられる．この問題を解決するためには，レイアウトブロックの配置する処理において，形状を考慮した工夫を加える必要がある．

5. まとめ

本稿では，ADC2019 問題をグラフ描画問題とモデル化し，グラフの位相埋め込みを考慮して平面描画を行うことでポリオミノ間ネットの配線を行った後，配線済みの状態を保ちながら最良な配置を探索する求解手法を提案した．また，計算機実験により各平面位相埋め込みに対して効率的なイテレーションが行われていることを確認した．

今後の課題として，レイアウトブロックの配置において，形状を考慮するなどのさらなる工夫を加えることが挙げられる．また，レイアウトブロックの閉面内に他のレイアウトブロックが入り込む場合の探索など，他のレイアウトブロックとの関係を考慮したレイアウトブロックの最小化にも改良の余地があると考えられる．

参考文献

[1] 大和田真由, 和田邦彦, 赤木佳乃, 佐藤真平, 高橋篤司. : 集合対間配線問題ソルバと引きはがし再配線の ADC2018 問題への適用，情報処理学会研究報告, Vol.2018-SLDM-185, No.13, (2018) pp.1-6
[2] 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 43.5 (2002): 1304-1314.
[3] Di Battista, G., Eades, P., Tamassia, R., Tollis, I, G.: *Graph Drawing: Algorithm for the Visualization of Graphs*, Prentice Hall, New Jersey, (1999)
[4] Google: *Google's OR-Tools* (2019) Retrieved 2019-10-09 from `https://developers.google.com/optimization/`
[5] Gleixner, Ambros, et al.: *The SCIP optimization suite 5.0.* (2017).

表 1 実験結果

Instance	#Minos	#Nets	Grid	#Connected	#Embedding	Density	Solved	BB_{size}	$T[sec]$	C_{total}	C_{route}
Q1	80	40	72×72	41	41	0.0617	○	30×32	0.1280	442	122
Q2	49	35	72×72	14	14	0.0378	○	26×30	0.0088	268	72
Q10	20	20	72×72	3	4	0.0154	○	12×12	0.0052	124	44
Q11	6	12	72×72	1	4096	0.0046	○	8×9	2.2241	38	14
Q16	5	6	72×72	2	2	0.0038	○	6×6	0.0048	30	10
Q25	50	25	20×20	25	25	0.4975	×	20×21	-	208	8
Q41	4	5	10×10	1	1	0.2000	○	6×7	0.0033	21	5