

配置配線パズルのための平面グラフを用いた解法

甘利 拓斗^{1,a)} 蓮見 平八郎^{1,b)} 藤吉 邦洋^{1,c)}

概要: DA シンポジウム 2020 で開催されるアルゴリズムデザインコンテストの題材は配置配線パズルである。この問題から配置配線前に解くべき問題を切り出して、グラフを平面に埋め込む手法を用いて解を求めることができるように、配置配線パズルをグラフに変換する手法を提案した。

キーワード: 配置配線パズル、グラフの平面埋め込み

A Method Using a Planar Embedding of a Graph for Placement and Routing Puzzle

TAKUTO AMARI^{1,a)} HEIHACHIROU HASUMI^{1,b)} KUNIHIRO FUJIYOSHI^{1,c)}

Abstract: The subject of “algorithm design contest” at DA Symposium 2020 is “placement and routing puzzle”. We propose a method to transform a puzzle problem into a graph to solve the preliminary problem of the puzzle using a planar embedding of a graph.

Keywords: placement and routing puzzle, planar embedding of a graph

1. はじめに

DA シンポジウム 2020 で開催されるアルゴリズムデザインコンテスト 2020 (ADC2020) [1] で扱われる問題は、配置配線パズルと呼ばれ、配線すべき端子が一部に与えられたブロック集合が与えられ、ブロックの配置と端子の配線を行い配置配線領域の最小化を目的としたものであった。問題例とその解答例を図 1 に示す。端子が与えられたブロックは、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、ADC2020 の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貢献すると期待できる。

ADC2020 の問題は ADC2019 と同じ問題であり、「配置配線前に端子から配線を出す方向を決定する問題」と「得られた方向を守って配置配線を行う問題」に分けて解くことができる。そこで我々は ADC2019 で、「端子から配線を出す方向を決定する問題」を、島上で位置が限定されない端子を含む平面位相配線問題に変更し、元の平面位相配線

問題を解く既存手法を拡張 [2] しこの問題を解いた。

本稿では、「端子から配線を出す方向を決定する問題」に対して解を求める手法として、配置配線パズルを工夫してグラフ化し、そのグラフを既存の平面描画アルゴリズム [3] に入力して平面描画を得て、その平面描画から配線方向を決定する手法を提案する。

2. 配置配線パズル

本稿では DA シンポジウム 2020 内で行われるアルゴリズムデザインコンテスト 2020 (ADC2020) [1] で使用される配置配線パズルから切り出した問題を扱う。そこでまず配置配線パズルの入力、出力、目的を以下に示す。

入力 ブロック集合

出力 ブロックの配置、端子間の配線

目的 ブロックと配線を含むバウンディングボックスの面積最小化

ブロックの形状は、格子の 1 区画 (マスと呼ぶ) からなるモノミノか、4 マスからなるテトロミノである。鏡像を区別し、90 度回転を同一視すると、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。ブロックは、回転

¹ 東京農工大学
Tokyo University of Agriculture and Technology
^{a)} amari@fjlab.ei.tuat.ac.jp
^{b)} hasumi@fjlab.ei.tuat.ac.jp
^{c)} fujiyosi@cc.tuat.ac.jp

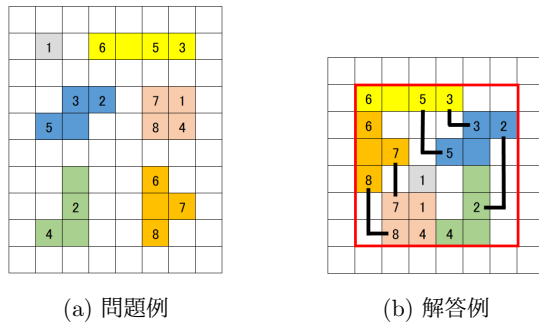


図 1: 配置配線パズルの問題例とその解答例

及び反転は許されない。ブロックのマスには、ネット端子が与えられる場合がある。このようなマスを端子マスと呼ぶ。入力として与えられるブロック集合には必ず同一ネットの端子マスが2つずつ存在し、それらの端子マス同士は配線で結ぶ必要がある。

配置配線パズルでは、与えられた各ブロックを格子状の1つの盤面に配置し、同数字の端子マス同士を全て配線したものを解とし、解はブロック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。

問題例とその解答例を図1に示す。

2.1 配置配線パズルの特徴

配置配線パズルは特定の端子についてそこから配線を出す方向によっては、問題全体で配線が不可能になってしまうことがある。図2に示す例で、J型のブロック上の2の端子は左右どちらの方向にも配線を出ることができるが、左に出すと図2(a)に示すように1か3のどちらかのネットが配線できなくなってしまうが、右に出すと図2(b)に示すように平面性を失わずに1と3のネットを両方とも配線できる。従ってこの例の場合2の端子から配線を出すべき方向は一意に決まる。このように詳細な配置配線の前に配線を出す方向を決定することで、詳細な配置配線段階において無駄な探索を減らすことができる。

2.2 問題設定

本稿では無駄な探索を減らすために、上下もしくは左右方向のみに配線を出せる端子は詳細な配置配線の前に配線を出す方向を決定することにして、以下の問題に設定する。

入力 ブロック集合

出力 上下もしくは左右方向のみに配線を出せる端子の配線を出す方向

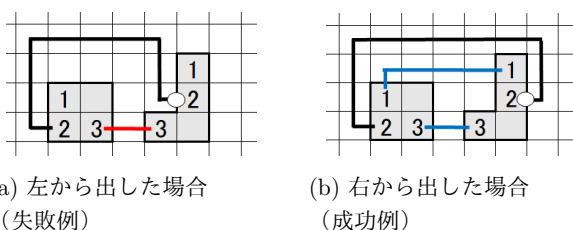


図 2: J型ブロックの2の端子から配線を出す方向によって配線が不可能になってしまうブロック2つからなる問題

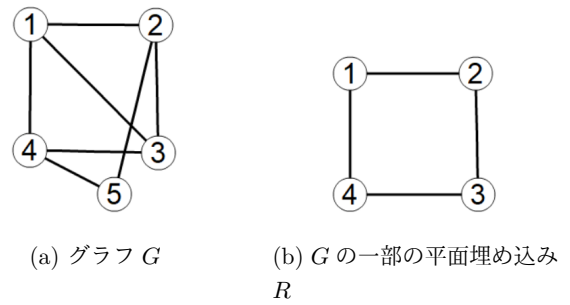


図 3: グラフ G の一部の平面埋め込み R における2つの断片 p_1, p_2 とそれぞれの接触頂点の例

3. Demoucron の平面性アルゴリズム [3]

Demoucron の平面性アルゴリズムは、入力が平面グラフであるならば平面埋め込みを返し、平面グラフでなければ平面ではないと判定するアルゴリズムである。初めに用語の定義を行う。 G がグラフであり、 R を G の部分グラフ S の平面埋め込みであるとして、 R に関連する G の断片 p を以下のいずれかとして定義する。

- (1) $u, v \in V(R)$ である $E(G) - E(R)$ 内の一つの枝 (u, v) 、または
- (2) $G - R$ の連結成分、並びに、その連結成分と R の頂点を結ぶ枝

断片 p における片側もしくは両側に単点を含まない枝に付随する R の頂点を R に関連する G の断片 p の接触頂点と呼ぶ。あるグラフ G について、その一部の平面埋め込み R であるときの2つの断片とそれぞれの接触頂点の例を図3に示す。

p が R に関連する G の断片であり、 R の面 f について p の全ての接触頂点が $V(f)$ に含まれるなら、 p は面 f で描画できる。

Demoucron のアルゴリズムはまず R を G の任意の閉路から始める。 R が G の平面埋め込みになる、もしくはいずれかの断片 p の全ての接触頂点が含まれる面が存在しなくなる (G が非平面) まで、グラフ G の部分グラフ S の平面埋め込み R を繰り返し拡張する。各断片 p について描画可能な複数の面を特定する。そして、描画可能な面 f が単独しかない断片 p とその面 f を優先的に選択する。描画可能な面 f が単独しかない断片 p が存在しない場合は

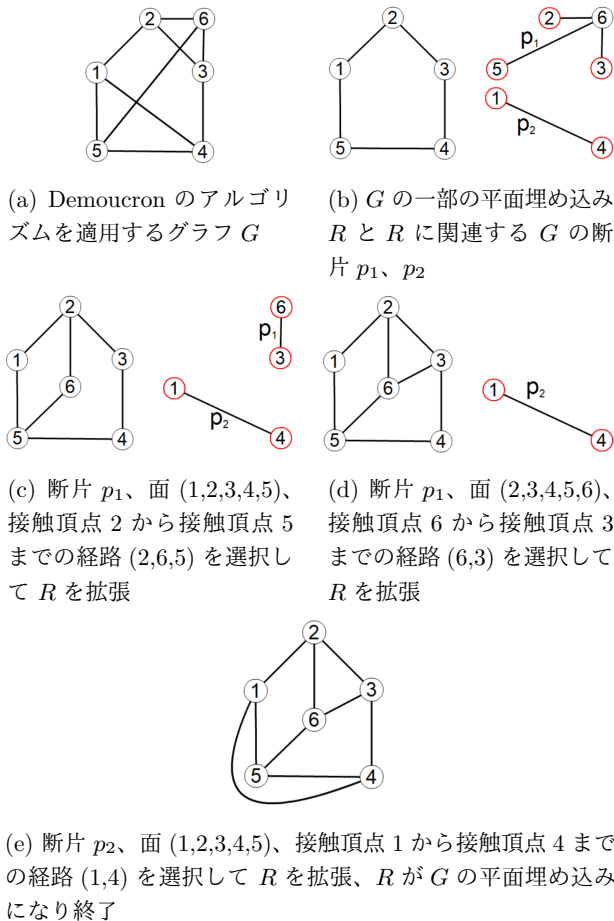


図 4: Demoucron のアルゴリズムの動作例

任意の断片 p 、任意の p が描画可能な面 f を選択する。 p を通る p のある接触頂点から p の他の接触頂点までの経路を選択して、 f を 2 つの面に分断する経路を描き R を拡張する。どの面に対しても描画可能条件を満たさない断片が見つかったときのみ、処理が止まる。その場合 R は G に平面拡張不可であり、 G は非平面。Demoucron のアルゴリズムの動作例を図 4 に示す。

4. 提案手法の基本部分

提案手法では平面描画アルゴリズム [3] を用いて平面描画を得ることにより、端子から配線を出す方向を決定できるように配置配線パズルをグラフ化する。まず配置配線パズルのグラフの基本変換について述べる。

4.1 配置配線パズルのグラフ化

配置配線パズルはまずブロック毎に 1 つの連結成分に変換し、そののちネット配線の代わりになる枝を張って全体をグラフ化する。ブロック毎の変換では、ブロック毎にそのブロックの外周において配線を出す順番を、連結成分上でも保持するという指針に沿ってグラフ化していく。

各ブロックのグラフ化

ブロック毎にその上の各端子を節点に一対一対応させる。次にブロックの外周に沿って一周した際に、ブロック上の端子の順番に応じて隣り合う端子に対応する節点間に枝を

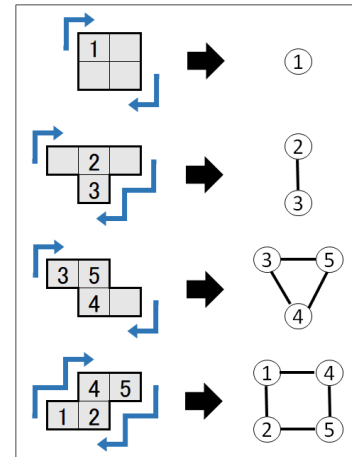


図 5: 両側に出せる端子を持たないブロックからなる連結成分の例

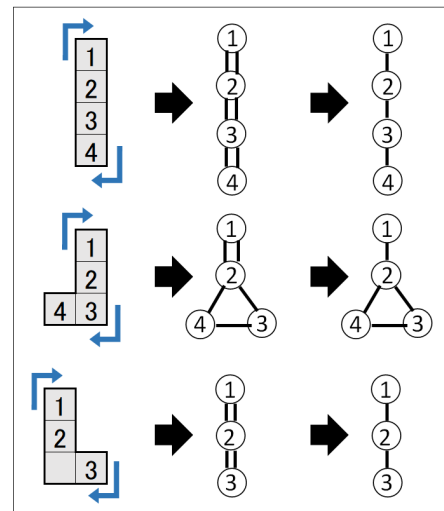


図 6: 両側に出せる端子を持つブロックからなる連結成分の例

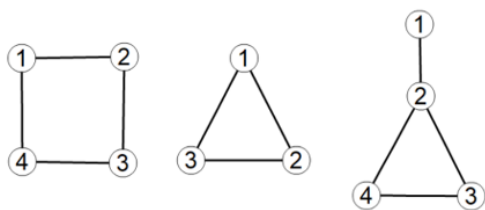
張る。両側から配線を出せる端子があるときも、同様に順番に応じて隣り合う端子に対応する節点間に枝を張る。このとき外周を一周する間に両側から配線を出せる端子が 2 回現れるがこれらは別物扱いせず、隣り合った節点間を枝で結び、二重になった枝は一本の枝に変換する。

図 5 に両側に出せる端子がないブロックの変換例を示す。図 6 に両側に出せる端子があるブロックの変換例を示す。このようにして各ブロックを 1 つの連結成分に変換するがここで張った枝をブロック枝と呼ぶことにする。

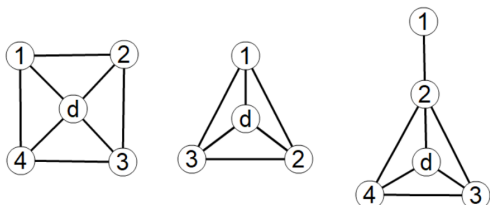
ブロックに対応する連結成分単体で閉路を持つものを図 7(a) に示す。理由は後述するがこのように閉路を持つ連結成分にはダミー節点を一つ追加し、その閉路上の全節点との間にダミー枝を張る。図 7(b) に図 7(a) にダミー節点を追加した例を示す。

配置配線パズル全体のグラフ化

ブロック毎に連結成分に変換した後は、同ネット端子対の節点同士に枝を張ることで、配置配線パズルに対応したグラフを得る。この時に張った枝はネット枝と呼ぶことにする。



(a) ブロックに対応する連結成分単体で閉路を持つもの



(b) (a) に示した各々の連結成分にダミー節点とダミー枝を追加

図 7: ダミー節点とダミー枝の追加

4.2 配線を出す方向の決定方法

配線を出す方向を決定するために、グラフを平面描画する。問題をグラフに変換することで得られるグラフは高々 1 通りしかないが、その平面描画は複数通り存在する。例えば、図 8 に示すようにダミー節点を持つ連結成分の平面描画は 2 通りに分類でき、ブロックの外周の端子の順と連結成分の外周の節点の順が同じ平面描画を表向きとし、逆順が同じ平面描画を裏向きとする。

配線を出す方向は、ダミー節点を持つ連結成分が全て表向きな平面描画が得られれば、得られた平面描画でネット枝が結ばれている方向と同じ方向に決定することができる。

図 2 で例示した問題を配置配線パズルのグラフ化に基づいて変換したグラフとその 2 通りの平面描画を図 9 に示す。図 9(b) の平面描画が得られたら、O 型ブロックに対応した連結成分が表向きなので右の J 型ブロックの 2 の端子は右から配線を出すことに決定する。もしも図 9(c) の平面描画が得られたら、O 型連結成分が裏向きなので右の J 型ブロックの 2 の端子は反転を考慮して右から配線を出すことに決定する。

ここで表向きと裏向きなダミー節点を持つ連結成分が混ざった平面描画では配線方向を決定することができない。従って Demoucron のアルゴリズムに入力したときに、ダミー節点を持つ連結成分が全て表向きもしくは裏向きな平面描画が得られるようにグラフ化していく。

5. 提案手法の発展部分

5.1 枝の除去による複数の連結成分への分割

表向きと裏向きなダミー節点を持つ連結成分が混在する平面埋め込みでは配線方向を得られないので、グラフを分割することが望まれる。そもそも平面描画は連結成分毎に別個に埋め込めばよく、ブリッジがあるならそれを除去し連結成分を分断して別個に平面描画すればよい。(連結成分において、ある枝を取り除くと連結成分が 2 つに分かれる枝を、ブリッジと呼ぶ。) 本稿では簡単のためブロック枝

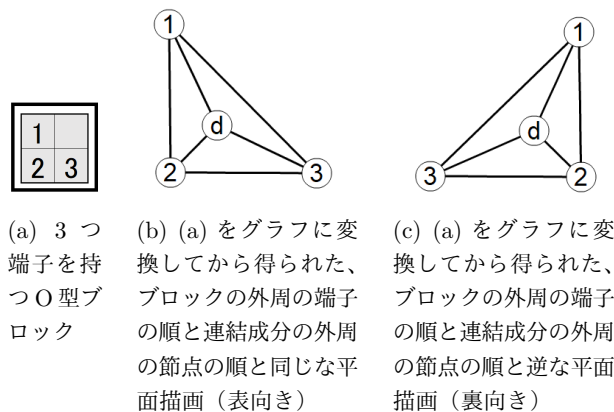
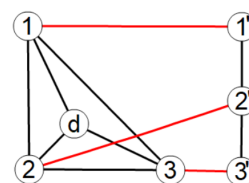
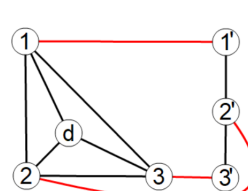


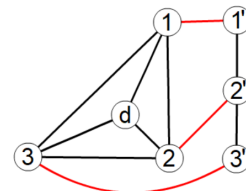
図 8: ダミー節点を持つ連結成分の平面描画



(a) 図 2 に示した問題を変換したグラフ (赤枝はネット枝)



(b) (a) の平面描画
(O 型が表向き)



(c) (a) の平面描画
(O 型が裏向き)

図 9: 図 2 に示した問題を変換したグラフとその平面描画

は除去せず、ネット枝の除去のみで分割を行う。

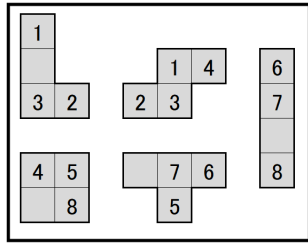
ブリッジを除去することで、2 つの連結成分へと分割できるグラフの例を図 10(a)(b)(c) に示す。枝 (4, 4') を除去すると、2 つの連結成分に分かれるため、枝 (4, 4') はブリッジであり、それを除去することで 2 つの連結成分に分割した。

さらに 2 本の枝の除去で連結成分が 2 つに分割できる場合、2 つの連結成分それぞれにおいて、除去された枝が接続していた 2 つの点を結ぶ枝を追加してから平面に埋め込めば、別個に解くことができる。グラフから 2 本の枝の除去で連結成分が 2 つに分割できる枝のペアを全て除去したことにより得られる連結成分毎に Demoucron のアルゴリズムで埋め込み、配線を出す方向を決定する。

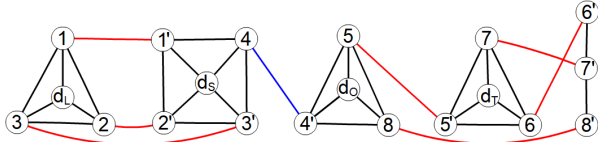
グラフを複数の連結成分に分割する例を図 10 に示す。

5.2 表向きと裏向きなダミー節点を持つ連結成分の混在阻止のためのネット枝の二重化

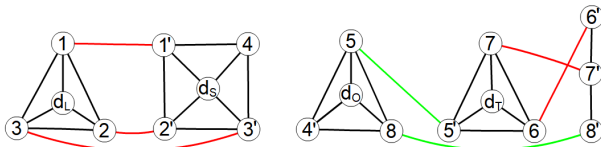
表向きと裏向きなダミー節点を持つ連結成分が混在しているため、平面描画から配線を出す方向を決定できない例を図 11(a)(b) に示す。図 11(c) のようにたまたま全て表向



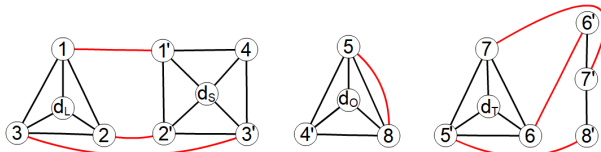
(a) L,S,O,T,I 型ブロックからなる問題



(b) グラフとブリッジ (青い枝)



(c) ブリッジを除去し 2 つの連結成分に分割されたグラフと 2 本の枝の除去で連結成分が 2 つに分割できる枝のペア (緑枝)



(d) 2 本の枝の除去で連結成分を更に 2 つに分割し、得られた 3 つの連結成分

図 10: 枝の除去による複数の連結成分への分割例

きな平面描画が得られる可能性もある。

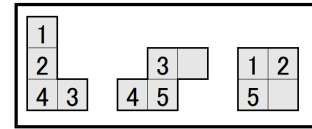
このような表向きと裏向きなダミー節点を持つ連結成分の混在を防ぐために、ネット枝の二重化を行う。二重化に用いる節点は、ダミー節点を持つ連結成分における、両側から配線を出せる端子に対応する節点を除いた閉路上の節点として、図 12 に示す。これらの節点同士がネット枝で結ばれている場合、それらを結んでいる端子対の節点とネット枝を二重化する。

図 11(d) のようにネット 3,4,5 を二重化することでそれらの連結成分は、表向きか裏向きに統一することができ、二重化の枝で全てのダミー節点を持つ連結成分が結ばれていれば、ダミー節点を持つ連結成分が全て表向きもしくは裏向きな平面描画を得ることができる。

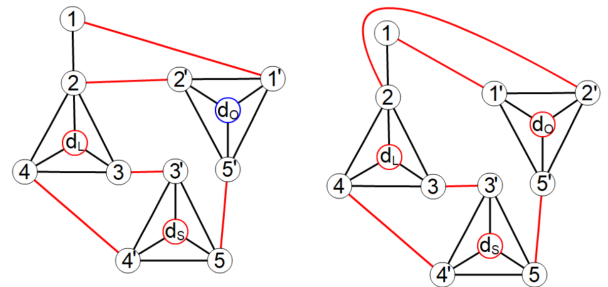
今回の手法の動作例を図 13 に示す。

5.3 表向きと裏向きなダミー節点を持つ連結成分の混在を阻止できない例

分割や二重化を使っても対応できない問題を図 14(a) に示す。この問題ではグラフをこれ以上分割することができない上に、L と O 型ブロックはダミー節点を持つ連結成分ではあるが、二重化できるネット枝が一つもないため二重



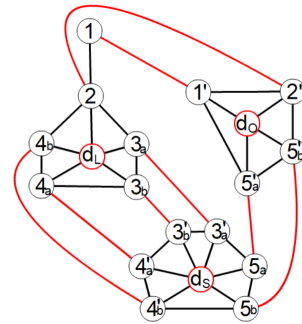
(a) L,S,O 型ブロックからなる問題



(b) (a) の平面描画

(c) (a) の平面描画

(L,S 型ブロックが表向き、
O 型ブロックが裏向き)



(d) ネット枝の二重化を用いた (a) の平面描画
(L,S,O 型ブロックが表向き)

図 11: ネット枝の二重化による表向きと裏向きなダミー節点を持つ連結成分の混在の阻止

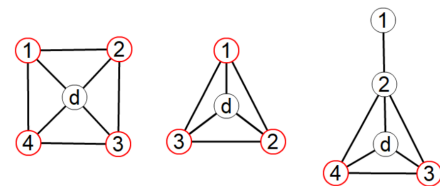
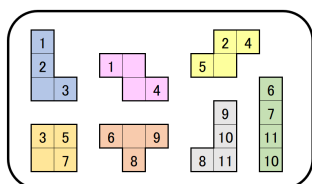


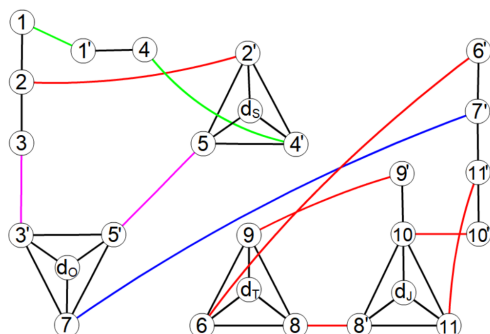
図 12: ダミー節点を持つ連結成分の二重化に用いる節点 (赤い節点)

化もできない。従って図 14(c) のように表向きと裏向きなダミー節点を持つ連結成分の混在する平面描画が得られる可能性がある。

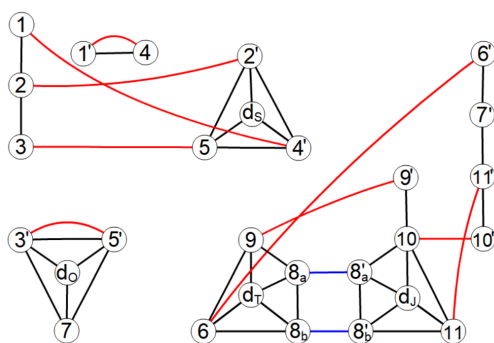
分割された連結成分の中で全てのダミー節点を持つ連結成分が、二重化したネット枝で結ばない場合に関しては、配線を出す方向を決められない可能性がある。



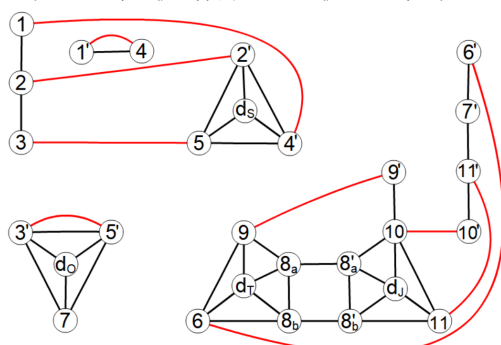
(a) L,Z,S,O,T,J,I 型ブロックからなる問題



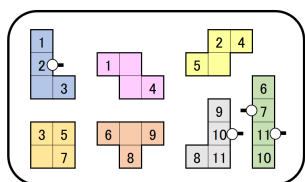
(b) 問題のグラフ化とブリッジ (赤い枝) と 2 本の枝の除去で連結成分が 2 つに分割できる枝のペア (緑の枝) とブリッジ除去後に 2 本の枝の除去で連結成分が 2 つに分割できる枝のペア (ピンクの枝) (グラフ化途中)



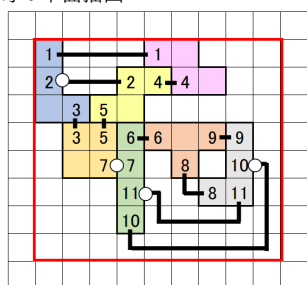
(c)(b) で示した 5 本の枝の除去とネット枝の二重化 (グラフ化終了)



(d) 連結成分毎の平面描画

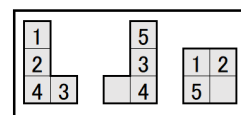


(e) 平面描画から得た配線方向

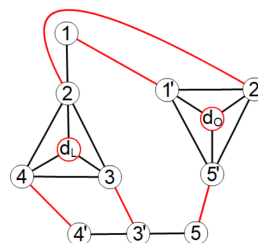


(f) 平面描画から得た配線方向

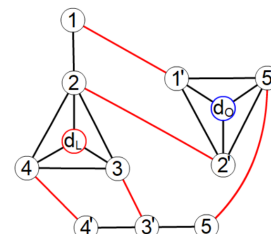
図 13: 動作例



(a) L,J,O 型ブロックからなる 2 重化できるネット枝がない問題



(b) (a) の平面描画 (L,O 型ブロックが表向き)



(c) (a) の平面描画 (L 型ブロックが表向き、O 型ブロックが裏向き)

図 14: 表向きと裏向きなダミー節点を持つ連結成分の混在を阻止できない例

6. まとめ

平面描画から上下もしくは左右方向のみに配線を出せる端子の配線を出す方向を決定できるように配置配線パズルをグラフに変換する基本的な方法を提案した。この方法に加えて、枝を除去して問題を分割する手法と、ネット枝の二重化により表向きと裏向きなダミー節点を持つ連結成分の混在を阻止する手法を提案した。分割手法においてはネット枝の除去のみを許容したが、ブロック枝の除去を行うことで更に問題を分割して解くこともできるが、複雑になりすぎてしまうため本稿ではネット枝の除去のみとした。

今後の課題としては、以前の手法との比較や、表向きと裏向きなダミー節点を持つ連結成分の混在を阻止できないものに対する対応策、以前提案した手法と今回の手法を組み合わせた手法の提案等が挙げられる。

参考文献

- [1] “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/>, (参照 2020/7/22)
- [2] 甘利拓斗, 蓮見平八郎, 藤吉邦洋, “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.
- [3] James A. McHugh, “ALGORITHMIC GRAPH THEORY”, new jersey:Prentice Hall.1990.