

集合対間配線手法の ADC2018 への適用に関する一考察

赤木 佳乃^{1,a)} 大和田 真由^{2,b)} 和田 邦彦^{2,c)} 佐藤 真平^{1,d)} 高橋 篤司^{1,e)}

概要：ADC2018 では、ナンバーリンクを二端子ネットの多層配線に拡張した問題が出題されている。引きはがし再配線による従来手法では、配線領域の混雑度を考慮して初期配線は生成されるが、その品質がよくないため、全てのネットを接続する配線を得るまでに、多くの再配線を必要とすることがあった。本研究では、引きはがし再配線において、必要な再配線の回数が少ないことが期待される配線パターンを、初期解とすることを検討している。T2H2 は、(1) 集合対間配線によって初期解を生成し、(2) その初期解を引きはがし再配線によって修正する、初期解は、ネットの端子対を接続するとは限らないが離れた端子を接続せず、必要な再配線の回数が少ないことが期待される。本稿では、(1) のステップにおける初期解生成方法の有効性について考察する。

キーワード：ナンバーリンク、アルゴリズムデザインコンテスト、目標端子対配線、引きはがし再配線

1. はじめに

アルゴリズムデザインコンテスト（以下、ADC）は、情報処理学会が主催する DA シンポジウムにて開催されている競技会である。「ナンバーリンク *1」のソルバーを設計し、どれほど多くの問題を制限時間内に解いたかを競う。ナンバーリンクとは、二端子ネットの平面配線を模したパズルである。ADC2018 [1] では、ナンバーリンクを二端子ネットの多層配線に拡張した問題が出題されている。問題サイズの拡大により、効率よく解を得ることが求められている。

二端子ネットの配線問題において、全てのネットを接続する配線パターンを得ることは、一般に NP 困難である [2]。SAT ソルバーを用いる手法 [3] や、引きはがし再配線 [4] による発見的手法などが提案されている。

引きはがし再配線による手法では、未配線のネットを含む配線パターンを初期解として生成し、その初期解を修正することで解を求める。多くの場合、初期解は、ネットの二端子を最短経路で接続する操作を逐次的に行うことで生成される。先に接続される配線は、配線領域の混雑度を考慮するなどすることで、後に接続するネットの配線を考慮し

て生成することは可能ではあるが、後に接続するネットほど配線の困難さは増していくなど、配線順序への依存が大きい。ネットの二端子間を接続する配線を、一度に生成するのではなく、部分的に生成することを繰り返すことで、配線順序への依存性を軽減できると考えられるが、効率的な部分的な配線生成についての知見は少ない。

本研究では、引きはがし再配線において、必要な再配線の回数が少ないことが期待される配線パターンを、初期解とすることを検討している。初期解の生成に集合対間配線を用い、異なる端子対を接続する配線を許容する。端子間の配線を部分的に生成するのと同等の効果を得ることで、初期解として、必要な再配線の回数が少ないことが期待される配線パターンを得ることを目指す。

「とりあえずつないではんせいするほうほう [5]（以下、T2H2）」は、我々が ADC2018 で用いた配線手法である。(1) 集合対間配線問題 [6] とみなし初期解を生成する「とりあえずつなく」ステップと、(2) 初期解の修正を行う「はんせいする」ステップからなる。(1) では、ネットワークフローを用いたアルゴリズム [7]（以下、目標端子対配線）を適用する。全てのネットを接続する配線パターンの生成は保証されないが、端子間の配線を部分的に生成するのと同等の効果の配線パターンを生成する。

本稿では、(1) のステップについて有効性を考察する。初期解により、効率よく解が求まるかを検討する。

¹ 東京工業大学工学院情報通信系

² 東京工業大学工学部情報工学科

a) akagi@eda.ict.e.titech.ac.jp

b) owada@eda.ict.e.titech.ac.jp

c) wada@eda.ict.e.titech.ac.jp

d) satos@ict.e.titech.ac.jp

e) atsushi@ict.e.titech.ac.jp

*1 ナンバーリンクとは、株式会社ニコリの商標登録である。

2. 前準備

2.1 ADC2018

ADC では、ナンバーリンクソルバーを競う。ナンバーリンクとは、盤面内に書かれている同じ数字同士を線でつなぐパズルである。1マスに引ける線は最大1本である。縦線と横線のみが使用でき、斜め線は使用不可である。交差する線や、枝分かれする線を含む回答はルール違反である。盤面内の数字を端子、同じ数字のペアをネット、数字同士をつなぐ線を配線と見なすと、ナンバーリンクは1層の二端子ネット配線問題と等価である。

ADC2018 では、ナンバーリンクを多層配線に拡張した問題が出題されている。1つの問題に対し、複数の解が存在する問題も含まれている。回答には、正解による得点と解の品質に応じた得点が与えられる。参加者は、各回答の合計得点を競う。

ADC2018 のルールを以下に示す。

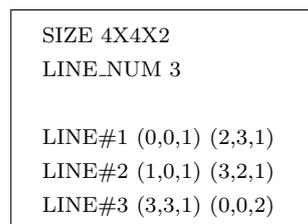
- (1) 問題 問題のサイズとネットの2端子の座標が、入力として与えられる。盤面のサイズは最大 72×72 であり、層数は最大8である。ネット数の上限は定められていない。
- (2) 回答 各層の配線パターンを回答する。交差や枝分かれする配線はルール違反とする。
- (3) 評価方法 正解ポイント、品質ポイント、ボーナスポイントの合計得点で評価される。正解ポイントは、正解したときに1点与えられる。品質ポイントは、解の品質に応じて10点が山分けされる。ボーナスポイントは、他の参加チームが作成した問題に正解したとき、1点与えられる。

ADC2018 では、配線長、配線の折れ曲がり回数、隣接している配線の長さをコストとしてみなしている。解の品質評価では、「総配線長 + 折れ曲がり回数の和 + $1/3 \times$ 隣接配線境界数」を評価式として用いている。隣接配線境界数とは、平面内に5マス以上隣接している配線があるとき、その隣接している配線の長さを表す。

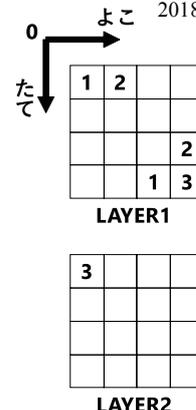
図1にADC2018の問題例を示す。図1(a)は問題の入力を表す。図1(b)は(a)の二次元平面図を表す。 $4 \times 4 \times 2$ の配線領域とネットリスト $\{1, 2, 3\}$ の二端子の座標が与えられている。

図2(a)に図1(a)の回答例を示す。図2(b),(c)は図1(b)の配線例を表している。図2(b)は図2(a)に対応する配線パターンを表している。

(b)の各配線長は、5, 4, 7であり総配線長は16である。各配線の折れ曲がり回数は1であり、折れ曲がり回数の総和は3である。隣接境界数は0である。したがって、(b)の解の品質は、 $16 + 3 + 1/3 \times 0 = 19$ である。(c)の

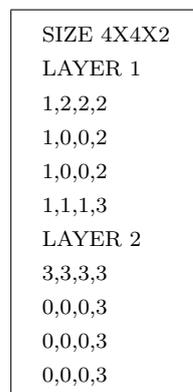


(a) 入力

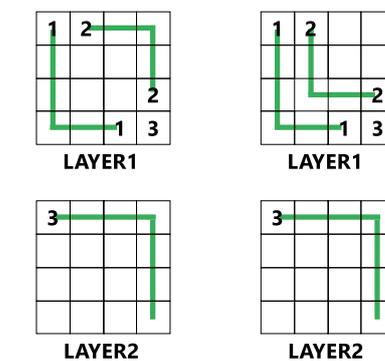


(b) 各層の盤面図

図1 ADC2018 問題例: SIZE は配線領域の大きさ, LINE_NUM はネット数, LINE#はネットの二端子の位置を表す。配線領域は、 $a \times b \times c$ の形で表す。aは横方向のマスの数, bは縦方向のマスの数, cは層の数である。第z層の盤面において、左からx番目, 上からy番目のマスの座標は $(x-1, y-1, z)$ の形で表す。端子の位置は、その端子が置かれているマスの座標で表す。



(a) 回答例



(b) 配線例 1 隣接配線境界数: 0
(c) 配線例 2 隣接配線境界数: 5

図2 ADC2018 回答例

各配線長は、5, 4, 7であり総配線長は16である。各配線の折れ曲がり回数は1であり、折れ曲がり回数の総和は3である。隣接配線境界数は5である。したがって、(c)の解の品質は、 $16 + 3 + 1/3 \times 5 = 20.67$ である。ADC2018では、(b)が品質の高い配線パターンである。

2.2 T2H2: とりあえずつないではんせいするほうほう

T2H2は、(1) 目標端子対配線によって初期解を生成する「とりあえずつなぐ」ステップと、(2) 引きはがし再配線によって修正を行う「はんせいする」ステップからなる。(1)において、必要な再配線の回数が少ないことが期待される配線パターンを生成する。必ずしもネットの端子対を接続するとは限らないが、離れた端子を接続せず、必要な再配線の回数が少ないことが期待される。

T2H2の各ステップの概要を以下に示す。本稿では、初期解生成のステップをsr, 修正の各ステップをそれぞれ

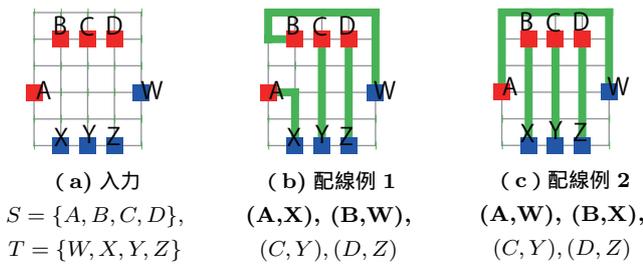


図 3 集合対間配線問題の配線例

h_1, h_2, h_3 で表す.

初期解の生成 (sr)

ネットリストに含まれる全ての端子を接続する配線パターンを求める. 生成した配線パターンを初期解として出力する. このとき, ネットと異なる端子対を接続する配線パターンが求まることを許容する.

修正

(h1) 接続実現している配線の経路を変えることなく, より多くのネットが接続実現されるように初期解を修正する. 初期解は, 接続未達成のネットの二端子を最短路で接続する操作を逐次的に行う. このとき, 配線で使用される辺の交差は認めない.

(h2) より多くのネットが接続実現されるように h_1 で得られる配線パターンを修正する. 一時的に配線の交差を許容することで, 接続未達成のネットを接続する配線を生成する.

(h3) 解の品質が向上されるように, h_2 で得られる配線パターンの再配線を行う.

ネットリスト内の全てのネットを接続する配線パターンを得るか, 繰り返し回数の上限に達するとアルゴリズムは終了する.

2.3 集合対間配線問題

集合対間配線問題は, プリント基板や LSI の配線設計の部分問題に見受けられる. プリント基板設計におけるプルアップ・ダウン抵抗やバイパスコンデンサなどの配線, シリコンインターポーザ [8], [9] 上の上層と下層を中継する配線は, 接続する端子対を入れ替えることが可能なため, 集合対間配線問題として扱われる.

集合対間配線問題は, 入力としてソース端子の集合 S とシンク端子の集合 T が与えられ, S に含まれるソース端子と T に含まれるシンク端子を接続する配線パターンを求める問題である. 片方の集合に含まれる端子は, もう一方の端子集合に含まれる端子と配線で接続される. 全ての端子は 1 対 1 で接続されるが, 接続する端子対の組み合わせは自由である.

図 3 に集合対間配線問題の例を示す. 図 3 (a) は問題の入力で, $(S, T) = (\{A, B, C, D\}, \{W, X, Y, Z\})$ 赤と青の 2 つの端子集合が与えられている. 図 3 (b), (c) は (a)

に対する配線パターンの例を示す.

本稿では, 配線領域を配線グラフを用いて表現する. 接続要求は, 配線グラフ上の点集合の対で表す. 配線は, 配線グラフの 2 点を接続するパスで表す. 集合対間配線問題は, グラフを用いて定式化すると, 集合対間の点非共有のパスを求める問題となる. 集合対間の点非共有パスは, フローネットワークのアルゴリズム [10], [11] を用いることで求まる.

2.4 目標端子対配線

目標端子対配線では, 全ての目標端子対の接続実現を目指し, 配線パターンを生成する. 目標端子対は, 配線で接続の実現を目指す端子の対を意味する.

集合対間配線問題において, 全ての目標端子対を接続できる目標端子対の集合を設定することは困難である. 目標端子対配線では, 全ての目標端子対を接続する配線パターンの存在は保証しない. 目標端子対を配線で接続する端子対の目安として用いる. 目標端子対集合に含まれる端子対を全て接続する配線が存在する場合でも, 目標端子対と異なる端子対が接続されることがある.

混雑度の高い領域を通過する配線は, 他の目標端子対の接続実現を妨げる可能性が高い. 目標端子対配線では, 他の目標端子対の配線が通過する可能性の高い領域を避けて配線を生成することで, 全ての目標端子対の接続実現を目指す.

ネットの全ての端子を囲む最小限界矩形を, ネットのバウンディングボックスと呼ぶ. 配線グラフの各辺に対し, 未配線の目標端子対のバウンディングボックスに含まれる数を, 混雑度と定義する. 例えば, 混雑度 0 の辺は, 未配線の目標端子対のどのバウンディングボックスにも含まれない辺であり, 混雑度 1 の辺は 1 つのバウンディングボックスに含まれる辺である.

目標端子対配線では, 目標端子対のバウンディングボックスを, その目標端子対を接続する配線が通過する可能性の高い配線領域とみなす. 2 つのネットのバウンディングボックスに含まれる辺は, それらを接続するどちらの配線でも使う可能性が高い. そこで, 混雑度 1 の辺を優先的に探索する.

3. 集合対間配線手法の ADC2018 への適用

T2H2 では, 初期解の生成に目標端子対配線を用いる. 異なる端子対を接続する配線を許容し, 端子間の配線を部分的に生成する配線パターンを生成する.

ネットの片方の端子をソース端子, もう一方の端子をシンク端子, ネットリストの各ネットを目標端子とする. 初期解は, アルゴリズムが出力する配線パターンに対応する. 盤面が図 3 (a), ネットリストが $\{(A, W), (B, X), (C, Y), (D, Z)\}$ のとき, 目標端子対

表 1 問題

問題	Size	Area	#Net	D_{tot}	D_{ave}
A1(Q11)	72 × 72 × 8	41472	375	19687	52.5
A2(Q2)	50 × 50 × 1	2500	320	1556	4.9
A3(Q17)	72 × 72 × 8	41472	99	2431	24.6
A4(Q19)	72 × 72 × 8	41472	22	436	19.8
B1(Q21)	32 × 32 × 5	5120	62	2015	32.5
B2(Q6)	72 × 72 × 8	41472	375	19077	50.9
B3(Q1)	32 × 32 × 3	3072	62	1953	31.5
B4(Q23)	72 × 72 × 8	41472	1500	8701	5.8

配線は、図 3 (b) に示す配線パターン生成する。T2H2 は、図 3 (c) に示す配線パターンを最終的に出力する。

4. 考察

目標端子対配線を初期解生成に適用することの有効性を考察する。計算機実験を行い、必要な再配線の回数が少ない配線パターン生成をするかを検証する。初期解により、端子間の配線を部分的に生成されているかを検証する。

以下では、ネットリストを入力として与え、迷路法 [12] によって初期解を生成し、その初期解に h2, h3 を順に適用して配線パターン生成する手法を「従来手法」と呼ぶ。従来手法の初期解生成のステップを「mr」と表す。

4.1 実験

ADC2018 で出題されている問題(Q1 から Q29)を T2H2 と従来手法によって解を求める。T2H2 は、文献 [5] の計算機実験と同じプログラムを用いる。従来手法の初期解生成のステップは、C++で実装したプログラムを用いる。コンパイラは g++ 5.4.0、最適化オプションは O2、Intel Core i7 4790K CPU、メモリ 32GB、Ubuntu 16.04 の環境で実行する。

T2H2、従来手法はいずれも正解の配線パターンを求めている。従来手法と比較し、T2H2 の合計の実行時間が短い問題は 7 題である。それ以外の 22 題の問題で、T2H2 は従来手法と比較して実行時間が長い。2 倍以上実行時間が長い問題は 3 題である。2 倍以上短い問題は 1 題である。以下では、T2H2 と従来手法で実行時間に大きく差が出ている 8 題の結果を示す。従来手法と比較し、T2H2 の合計の実行時間が短い問題 4 題と、長い問題 4 題について考察を行う。

表 1 に問題のサイズ、マス数、ネット数、ネットの二端子間のマンハッタン距離の総和をそれぞれ、Size, Area, #Net, D_{total} , D_{ave} で示す。カッコ内の文字は、ADC2018 での問題番号を表す。問題は、従来手法と T2H2 の実行時間の差が昇順になるように並べている。解を得るまでの実行時間が、T2H2 は従来手法と比較し短い問題は A1~A4、長い問題は B1~B4 である。表 2 に各ステップの実行時間(秒)、合計の実行時間(秒)を示す。カッコで示されてい

る数字は、T2H2 の合計の実行時間を 1 とした時の実行時間の割合を表す。Diff. は T2H2 の合計の実行時間と従来手法の合計の実行時間の差を示す。差が負の時、T2H2 は従来手法と比較し短い実行時間で解を求めたことを表す。表 3 に各ステップにおいて、接続実現されているネット数を示す。カッコで示されている数字は、ネット数に対し接続実現されているネット数の割合を表す。表 4 に D_{ave} に対する配線長の平均と目標端子までの距離の平均の割合を、 $r(L_{ave}), r(D_{goal})$ で示す。目標端子までの距離は、端子からその端子の目標端子までのマンハッタン距離を表す。小さいほど解に近い配線パターンであることを意味する。接続実現されている時、そのネットの端子は、目標端子までの距離が 0 である。接続未達成の時、目標端子までの距離は、その端子が接続されている配線の端点から目標端子までのマンハッタン距離である。

4.2 計算時間の評価

必要な再配線の回数が少ないとき、h2 のステップにおける実行時間は小さくなると考えられる。T2H2 による h2 の実行時間と従来手法の h2 の実行時間を比較する。

表 2 に示すように、T2H2 の h2 において、実行時間が 0 の問題は、A3, A4, B4 である。従来手法の h2 と比較し、実行時間が小さい問題は、A1, A2 である。5 個の問題に対し、sr は必要な再配線の回数が少ない配線パターンを生成している。

従来手法の h2 において、実行時間が 0 の問題は、A3, A4, B1, B3, B4 である。従来手法の h2 において、T2H2 の h2 と比較し、実行時間が小さい問題は、B2 である。6 個の問題に対し、mr は必要な再配線の回数が少ない配線パターンを生成している。

sr は、再配線の回数が少ない配線パターンを生成する。mr には劣る傾向にあるが、いくつかの問題では、再配線の回数が少ない配線パターンを生成する。

4.3 接続実現するネット数の評価

必要な再配線の回数が少ないとき、T2H2 の h1 のステップにおいて接続実現するネット数は多くなると考えられる。T2H2 の h1 と mr のステップにおける接続実現するネット数を比較する。

表 3 に示すように、T2H2 の h1 のステップにおいて、全てのネットの接続実現する問題は 3 題である。mr のステップでは、5 題である。T2H2 は、従来手法と比較し、接続実現するネット数の数が少ない傾向にある。一方、いくつかの問題では多くのネットの接続を実現することが示される。

T2H2 の h1 のステップによって接続実現するネットの数は、mr のステップと比較し、A1~A4 では同等が多い。B1~B4 では同等か少ない。多くのネットの接続を実現す

表 2 実行時間 (秒)

問題	T2H2					従来手法				Diff.
	sr	h1	h2	h3	Total	mr	h2	h3	Total	
A1	53.28	46.37	97.67	16.66	213.99	85.51	1682.76	18.21	1786.47(8.35)	-1572.48
A2	0.26	0.29	5.06	0.44	6.05	0.19	7.68	0.45	8.32(1.38)	-2.27
A3	4.00	5.60	0.00	5.48	15.09	10.19	0.00	6.67	16.87(1.12)	-1.78
A4	0.64	0.00	0.00	1.06	1.71	1.44	0.00	1.13	2.58(1.51)	-0.87
B1	0.17	0.16	3.64	0.12	4.09	0.27	0.00	0.14	0.42(0.10)	3.68
B2	52.18	50.47	23.98	20.82	147.45	86.60	14.77	19.38	120.75(0.82)	26.70
B3	0.09	0.04	60.53	0.04	60.69	0.11	0.00	0.04	0.15(0.00)	60.53
B4	89.84	77.88	0.00	113.63	281.35	74.63	0.00	114.26	188.90(0.67)	92.45

表 3 接続実現されたネット数

問題	#Net	T2H2		従来手法
		sr	h1	mr
A1	375	1(0.00)	366(0.98)	357(0.95)
A2	320	97(0.30)	260(0.81)	239(0.75)
A3	99	18(0.18)	99(1.00)	99(1.00)
A4	22	22(1.00)	22(1.00)	22(1.00)
B1	62	13(0.21)	60(0.97)	62(1.00)
B2	375	1(0.00)	368(0.98)	368(0.98)
B3	62	22(0.35)	49(0.79)	62(1.00)
B4	1500	627(0.42)	1500(1.00)	1500(1.00)

るとき、実行時間は小さくなる傾向にある。

4.4 配線パターンの評価

表 4 に示すように、T2H2 の h1 のステップによって生成される配線パターンの $r(D_{goal})$ は、mr のステップによって生成される配線パターンと比較し、A1 ~ A4 では同等か小さい。B1 ~ B4 では同等か大きい。

A3, A4, B4 では、T2H2 の h1 のステップ、mr のステップいずれのステップで、 $r(D_{goal})$ は 0 である。T2H2 の h1 のステップによって生成される配線パターンの $r(L_{ave})$ は、mr のステップと比較し同等か小さい。

T2H2 の h1 のステップによって生成される配線パターンの $r(D_{goal})$ は、従来手法の mr のステップと比較し、A1, A2 では小さい。B2 では大きい。

T2H2 の h1 のステップにおいて、全てのネットの接続実現する問題では、 $r(L_{ave})$ が従来手法の mr のステップと比較して短い傾向にある。目標端子対配線により、必要な再配線の回数が少ない配線パターンが生成されていることが確かめられる。

4.5 考察

T2H2 の h1 のステップと従来手法の mr を比較すると、A1, A2, B1, B3 の結果から、より多くのネットの接続を達成している手法は、合計の実行時間が短い傾向にあることが確かめられる。A1, A2, B2 の結果から、接続実現し

ているネット数が同じとき、より $r(D_{goal})$ が短い手法は、h2 のステップの実行時間が短い傾向にあることが確かめられる。多くのネットの接続を達成する配線パターンを初期解に用いることで、修正に必要な計算時間が短縮され、効率よく解が求まることが期待される。

5. まとめ

二端子ネットの配線問題において、引きはがし再配線による従来手法では、初期解生成において、すべてのネットを配線を得るまでに多くの再配線を必要とすることがあった。配線領域の混雑度を考慮して初期配線は生成されるが、その品質がよくない傾向にあった。

T2H2 では、初期解の生成に集合対間配線を用い、異なる端子対を接続する配線を許容することで、端子間の配線を部分的に生成するのと同等の効果を得ることを目指す。初期解とし、必要な再配線の回数が少ないことが期待される配線パターンを用いることで効率的に解を求める。

本稿では、計算機実験を行い、初期解生成のアルゴリズムの有効性を考察した。従来手法と目標端子対配線のうち、多くのネットの接続を達成する配線パターンを初期解に用いることで、修正に必要な計算時間が短縮され、効率よく解が求まることが期待される。今後の課題として、より多くのネットの接続を実現する目標端子対配線のアルゴリズムの考案や、各配線手法が得意とする問題の特徴の分析が挙げられる。

表 4 配線長と目標端子までの距離

	T2H2						従来手法			
	sr		h1		h2	h3	mr		h2	h3
問題	$r(L_{ave})$	$r(D_{goal})$	$r(L_{ave})$	$r(D_{goal})$	$r(L_{ave})$	$r(L_{ave})$	$r(L_{ave})$	$r(D_{goal})$	$r(L_{ave})$	$r(L_{ave})$
A1	0.25	0.99	1.19	0.03	1.25	1.18	1.08	0.04	1.29	1.20
A2	0.91	0.83	0.76	0.32	1.20	1.18	0.86	0.36	1.20	1.19
A3	0.76	0.85	1.00	0.00	1.00	1.00	1.04	0.00	1.04	1.00
A4	1.00	0.00	1.00	0.00	1.00	1.00	1.00	0.00	1.00	1.00
B1	1.02	0.38	1.04	0.03	1.42	1.28	1.05	0.00	1.05	1.00
B2	0.26	0.99	1.13	0.02	1.17	1.12	1.11	0.01	1.14	1.11
B3	1.02	0.38	0.89	0.21	1.04	1.04	1.06	0.00	1.06	1.00
B4	0.95	0.95	1.04	0.00	1.04	1.01	1.05	0.00	1.05	1.01

参考文献

- [1] 2018DA シンポジウム実行委員会：DA シンポジウムアルゴリズムデザインコンテスト，情報処理学会（オンライン），入手先（<https://dasadc.github.io/adc2018/>）（参照 2018-11-05）。
- [2] Garey, M. R. and Johnson, D. S.: *Computers and Intractability, A Guide to the Theory of NP-Completeness*, Freeman and Co., New York (1979).
- [3] 迫龍哉, 川原征大, 宋剛秀, 番原睦則, 田村直之, 鍋島英知：SAT 型制約ソルバーによるナンバーリンクの解法とその評価，人工知能学会全国大会論文集 2016 年度人工知能学会全国大会（第 30 回）論文集，一般社団法人人工知能学会，pp. 1D4OS02a3–1D4OS02a3 (2016).
- [4] Dees, W. A. and Karger, P. G.: Automated rip-up and reroute techniques, *Design Automation, 1982. 19th Conference on*, IEEE, pp. 432–439 (1982).
- [5] 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 (SLDM) (2017).
- [6] 高橋篤司：集合対間配線問題に関する一考察，電子情報通信学会技術研究報告，Vol. 111, pp. 23–28 (2011).
- [7] Akagi, K., Sato, S. and Takahashi, A.: An Idea for Maximizing Target Pin-Pair Connections in Set-Pair Routing, *The 32nd International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2017)*, pp. 62–65 (2017).
- [8] Liu, W.-H., Chang, M.-S. and Wang, T.-C.: Floorplanning and signal assignment for silicon interposer-based 3D ICs, *Proc. Design Automation Conference (DAC)*, pp. 1–6 (2014).
- [9] Ho, Y.-K. and Chang, Y.-W.: Multiple chip planning for chip-interposer codesign, *Proc. Design Automation Conference (DAC)*, pp. 1–6 (2013).
- [10] Hu, T. C.: Integer programming and network flows, Technical report, DTIC Document (1969).
- [11] Bondy, J. A. and Murty, U. S. R.: *Graph Theory with Applications*, North-Holland, Amsterdam (1976).
- [12] Lee, C.: An Algorithm for Path Connections and Its Applications, *Electronic Computers, IRE Transactions on*, Vol. EC-10, No. 3, pp. 346–365 (1961).