T2R2 東京科学大学 リサーチリポジトリ
Science Tokyo Research Repository

論文 / 著書情報
Article / Book Information

論題	目標端子対接続の実現を目指す集合対間配線アルゴリズム
Title	A Set-Pair Routing Algorithm Realizing Target Pin-Pair Connections
著者	赤木佳乃, 佐藤真平, 高橋篤司
Authors	Kano Akagi, Shimpei Sato, Atsushi Takahashi
出典	第30回 回路とシステムワークショップ 論文集, , , pp. 180-185
Citation	Proc. the 30th Workshop on Circuits and Systems, , , pp. 180-185
発行日 / Pub. date	2017, 5
URL	http://search.ieice.org/
権利情報 / Copyright	本著作物の著作権は電子情報通信学会に帰属します。 (c) 2017 Institute of Electronics, Information and Communication Engineers

Powered by T2R2 (Science Tokyo Research Repository)

目標端子対接続の実現を目指す集合対間配線アルゴリズム
A Set-Pair Routing Algorithm Realizing Target Pin-Pair Connections

赤木 佳乃　　佐藤 真平　　高橋 篤司
東京工業大学 工学院 情報通信系

Kano AKAGI　　Shimpei SATO　　Atsushi TAKAHASHI
Department of Information and Communications Engineering
Tokyo Institute of Technology

1 はじめに

プリント基板やLSIの配線問題では，接続の実現だけでなく，配線遅延を減少させる短い配線や，信号伝搬遅延を揃える配線などが求められる．様々な厳しい要求を満たす配線が必要とされる場合，手動設計が主流である．しかし，大規模化への対応や設計時間短縮のために，自動化への要望は強く，自動配線ツールの性能向上が求められている．本稿では，集合対間配線において配線長が短く，配線長差が小さい配線を得る手法について検討する．

集合対間配線問題 [1] の入力は，2つの端子集合間に与えられる接続要求である．出力は，片方の端子集合に含まれる端子と，もう一方の端子集合に含まれる端子を接続する配線である．全ての端子は1対1で接続されるが，端子の組み合わせは自由である．

集合対間配線問題において，総配線長が最小である配線は多項式時間で得られる [2, 3]．一方，配線長差が最小の配線を得ることは難しいと考えられている．例えば，総配線長最小の配線を初期解とし，配線長差のみを評価指標とし配線を修正すると，局所最適解に陥り，配線長差は初期解から削減されないことが多い．既存研究 [4] では，配線長差と相関がある評価指標を用い，逐次的に配線を修正し，総配線長および配線長差が小さい配線を発見的に得る手法を提案している．

既存手法 [4] では，総配線長最小の配線を初期解として生成し，総配線長を変更させずに，配線長差が小さくなるように配線を修正する．総配線長最小の配線は，一般に幅優先探索で近い端子対を接続することを繰り返したのち，総配線長を削減する操作を適用することで得られる．そのため，離れた端子対が接続される配線を含み，配線長差が大きいことが多い．既存手法 [4] では，総配線長最小の初期解に対し，配線長差を削減する修正方法を効率よく見つけることができず，配線の修正に多くの計算時間を必要としている．初期解とし，離れた端子対が接続されていない，配線長差の小さい配線を効率よく生成できれば，修正により，総配線長と配線長差が小さい配線を効率よく得ることが期待できる．

本研究は，集合対間配線問題に対して，近い端子対の集合を目標端子対集合として設定し，それらを接続する初期解を生成することで，総配線長と配線長差の小さい配線を効率よく生成することを目指す．

本稿では，初期解生成において，目標端子対集合に含まれる端子対をできる限り接続する配線を生成するアルゴリズムを提案する．提案アルゴリズムでは，各配線長および配線長差の小さい配線が得られると予測される目標端子対集合が，入力として与えられることを想定する．しかし，目標端子対集合の全ての端子対を接続する配線が存在しないことを許容する．

提案アルゴリズムでは，目標端子対集合に含まれる端子対を全て接続する配線の実現を目指し，重なり度を導入し，他の目標端子対の接続を妨げない配線を優先して採用する．しかし，必ずしも目標端子対を接続する配線が得られるとは限らない．

実験では，目標端子対集合を手動で決定し，提案アルゴリズムで得られた初期解を既存の最短配線冗長化アルゴリズム [4] で修正することで，配線長差の小さい配線を効率よく得ることを確認した．

2　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である．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 [2, 5] 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考えることができる．

部分問題として扱われる配線問題は，全体として遅延制約を満たすために，各信号の遅延や複数の信号間の遅延差が小さい配線を生成することを目的にすることが多い．複数の信号間で遅延差を抑えるために，伝搬させる配線をできる限り並走させることや，遅延が配線長に比例するとみなせる場合において，配線長を揃えることなどが試みられる．

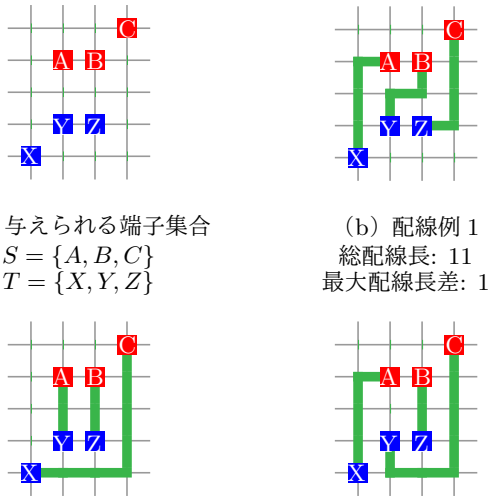
集合対間配線問題は，入力としてソース端子の集合 S とシンク端子の集合 T が与えられ，S に含まれるソース端子と T に含まれるシンク端子を接続する配線を求める問題である．片方の集合に含まれる端子は，もう一方の端子集合に含まれる端子と接続する．全ての端子は 1 対 1 で接続されるが，接続する端子対の選択は自由である．

提案アルゴリズムが対象とする問題の配線領域は，平面とする．配線は，グリッド上に制限する．グリッドの間隔は，プリント基板製造における配線の最小間隔などに設定することを想定する．本稿では配線領域を，グリッドの交点をグラフの頂点，交点間の線分を辺とする配線グラフで表現する．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 [6, 7] を用い，求めることができる．

本稿では，与えられるソース端子とシンク端子の数は等しく，配線可能な入力が与えられることを前提とする．

図 1 に集合対間配線問題の例を示す．(a) は問題の入力で，端子集合 $(S, T) = (\{A, B, C\}, \{X, Y, Z\})$ が与えられている．(b)，(c)，(d) は問題の出力例である．

(b) の各配線長は 4，3，4 であり総配線長は 11，最大配線長差は 1 である．(c) の各配線長は 2，2，7 であり総配線長は 11，最大配線長差は 5 である．(d) の各配線長は 4，7，2 であり総配線長は 13，最大配線長差は 5 である．総配線長と配線長差がより小さい配線が求められる場合，(b) が適切な配線である．



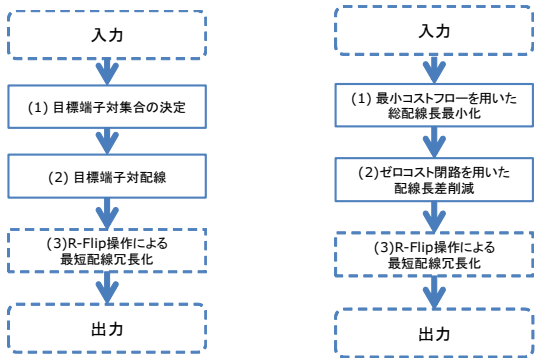
（a）与えられる端子集合
$S = \{A, B, C\}$
$T = \{X, Y, Z\}$

（b）配線例 1
総配線長: 11
最大配線長差: 1

（c）配線例 2
総配線長: 11
最大配線長差: 5

（d）配線例 3
総配線長: 13
最大配線長差: 5

図 1: 集合対間配線問題



（a）本研究　　（b）既存研究

図 2: 配線長差削減のための配線設計フロー

3　既存手法

既存手法 [4] では，集合対間配線問題の入力に対して 3 つのステップを経て配線を生成する．既存手法の配線設計フローを図 2（b）に示す．

（1）では各辺のコストを 1 として，最小コストフローを求める．（2）ではゼロコスト閉路を用い，総配線長を変えずに配線長差を削減する．（3）では R-Flip 操作 [8, 9] を用い，短い配線を最長配線長と同等の長さまで伸ばす．

配線長差を削減するゼロコスト閉路を，効率よく求める求める方法は知られていない．既存手法では（2）の配線長差削減のステップにおいて，発見的手法を用いてゼロ閉路を探索し，その閉路を基に配線を修正したとき，評価が改善したら適用することを繰り返す．（2）のステップでは，フローグラフの頂点

数を V，辺数を E，集合対間配線の本数を N，総配線長を L とすると, $O(VENL)$ 時間の計算が必要であり，他のステップと比べると非常に時間を必要とする．

また，既存手法では必ずしも配線長差を削減するゼロ閉路を発見できるとは限らないため，入力によっては十分に配線長差を削減できない場合がある．

4　目標端子対の接続を目指す配線アルゴリズム

4.1　概要

本研究は，既存の手法よりも短い計算時間で，総配線長と配線長差が小さい配線を生成することを目指す．本研究で提案する配線長差削減のための配線設計フローを図 2（a）に示す．本稿で提案するアルゴリズムは（2）の目標端子対配線である．

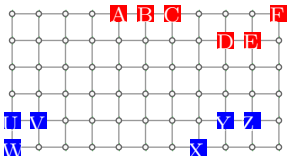
（2）の目標端子対配線のステップでは,（1）のステップで決定する目標端子対集合の端子対全ての接続を目指して配線を行う．ただし，目標端子対集合に含まれる端子対全てを接続する配線が存在するとは限らない．そこで，目標端子対は配線の目安として用いる．配線は，他の目標端子対の配線が通過する可能性の高い配線領域を避けて生成する．提案アルゴリズムでは，目標端子対集合に含まれる端子対を全て接続する配線が存在する場合でも，目標端子対と異なる端子対が接続されることがある．

ネットのすべ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，重なり度と定義する．例えば，重なり度 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，重なり度 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．

提案アルゴリズムでは，目標端子対のバウンディングボックスを，それら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，提案アルゴリズムでは重なり度 1 の辺を優先的に探索する．

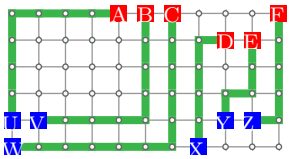
4.2　目標端子対配線アルゴリズム

図 2（a）中の（2）目標端子対配線における配線アルゴリズムについて詳細に述べる．このステップの入力は，2 つの端子集合と目標端子対集合である．出力は配線である．



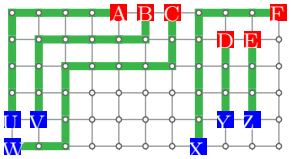
$S = \{A, B, C, D, E, F\}$
$T = \{U, V, W, X, Y, Z\}$

（a）入力



$(A, U), (B, V), (C, W), \underline{(D, X)}, \underline{(E, Y)}, \underline{(F, Z)}$

（b）提案アルゴリズム



$(A, U), (B, V), (C, W), \underline{(D, Y)}, \underline{(E, Z)}, \underline{(F, X)}$

（c）総配線長最小化アルゴリズム

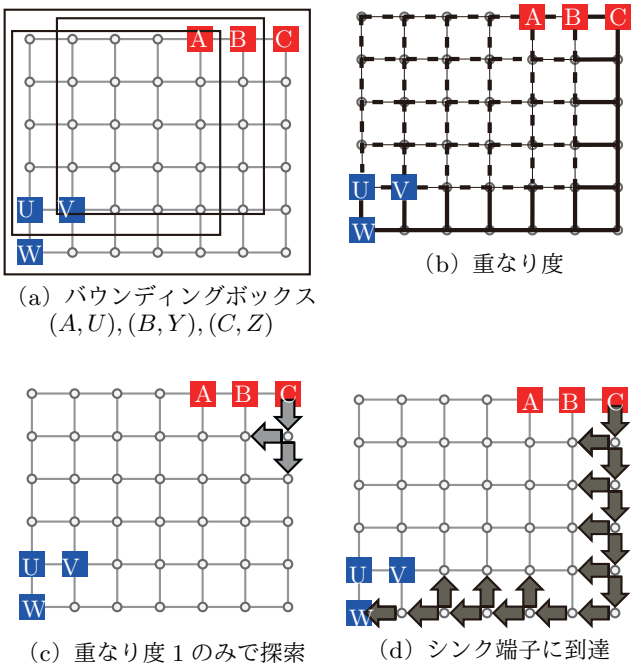
図 3: 例題の入出力と接続端子対

提案アルゴリズムでは，配線グラフを大ソース，大シンクを 1 つずつ持つフローグラフに置き換える．フローグラフのすべての点の容量を 1 とする．大ソースから大シンクへのフローは，フローの量が 1 の単位フローからなるとする．大ソースから大シンクへの各単位フローは，他の単位フローとは交わらないため，配線に対応する．

提案アルゴリズムではソースからシンクへのフロー増加路の探索を，幅優先探索を用いて繰り返す．ただし，探索では重なり度 1 の辺を優先する．

はじめに，重なり度 1 の辺のみでフロー増加路の探索を行う．フロー増加路が見つからない時は，重なり度 1 以上の辺で探索する．それでもフロー増加路が見つからない時は，すべての辺で探索する．入力として，配線可能な 2 つの端子集合を仮定するので，フロー増加路は必ず見つかる．

図 3（a）に示す例題に対して，提案アルゴリズムと既存手法で生成される配線をそれぞれ，図 3（b）（c）に示す．端子集合 (S, T) は $(\{A, B, C, D, E, F\}, \{U, V, W, X, Y, Z\})$ である．提案アルゴリズムでは目標端子対集合を $\{(A, U), (B, V), (C, W), (D, X), (E, Y), (F, Z)\}$ としたとき，全ての目標端子対が接続される．一方，



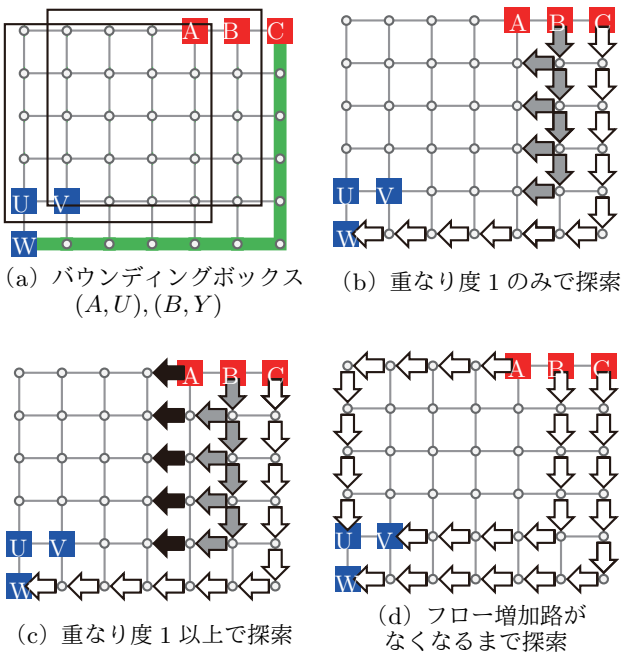
(a) バウンディングボックス $(A,U),(B,Y),(C,Z)$

(b) 重なり度

(c) 重なり度 1 のみで探索

(d) シンク端子に到達

図 4: 重なり度 1 の辺のみでの探索



(a) バウンディングボックス $(A,U),(B,Y)$

(b) 重なり度 1 のみで探索

(c) 重なり度 1 以上で探索

(d) フロー増加路がなくなるまで探索

図 5: 重なり度 1 以上の辺での探索

既存手法では，異なる $\{(D,Y),(E,Z),(F,X)\}$ が接続されている．

図 4 に図 3 に示す例題の一部を示す．図 4 に示す例題を用いて，重なり度 1 の辺のみでのフロー増加路の探索について説明する．図 4（a）に目標端子対 $(A,U),(B,V),(C,W)$ のバウンディングボックスを，図 4（b）に重なり度を示す．重なり度 1 の辺は太い線，重なり度 2 以上の辺は点線で示している．ソース端子 A，B に接続している辺は，B，C 間の辺を除き，全て重なり度が 1 ではない．図 4（c）に重なり度 1 の辺のみで探索している途中を示す．図中の矢印は探索した辺を示している．図 4（d）に重なり度 1 の辺のみの探索により，ソース端子 C からシンク端子 W へのフロー増加路を見つけるまで探索した結果を示す．(B,V) のバウンディングボックス内の辺は探索されない．

図 5 は (C,W) が配線により接続された状態を示す．図 5 を用いて重なり度 1 以上の辺でのフロー増加路の探索について説明する．図 5（a）に未配線の端子対 $(A,U),(B,V)$ のバウンディングボックスを示す．図 5（b）に重なり度 1 の辺のみの探索を示す．すでにフローが流れている辺は白塗りの矢印，探索された辺はグレーの矢印で示している．図 5（b）において，これ以上重なり度 1 の辺のみで探索はできない．フロー増加路が見つからないため，この状態から重なり度 1 以上の辺で探索を行う．図 5（c）に重なり度 1 以上の辺で探索を許容したときの探索の様子を示す．探索を許容された辺を黒塗りの矢印で示している．A に接続する辺と，B から探索を進めていた辺と隣接する辺が探索される．

図 5（d）にフロー増加路がなくなるまで探索を繰り返した結果を示す．図 3（a）に示す例題に提案アルゴリズムを適用すると，図 3（b）に示す配線を出力する．

本研究では，端子対の選択および目標端子対配線を行なった後，最短配線冗長化アルゴリズム [4] により配線を修正する．

5 実験

提案アルゴリズムと本研究で提案する配線設計フローの有効性を，計算機実験にて確かめる．

実験では，提案アルゴリズムを実装し，既存手法の総配線長最小化，配線長差削減，最短配線冗長化の実装を使用する．実装は C++言語を用い，Intel Core i7 4790K CPU，メモリ 32GB，Ubuntu 16.04 の環境を用いる．コンパイラは g++ 5.4.0 を用い，最適化オプションは O2 を用いる．目標端子対集合は手動で与える．

集合対間配線問題の入力については，広く用いられているベンチマークが存在しない．そのため，既存研究 [4] の実験で使われた例題を用いる．集合対間配線問題の実例として想定される条件を基に，端子対と障害をランダムに生成した問題である．エス

ケープ配線を想定したIO端子配置（E），バス配線を想定したアレイ状の端子配置（B），フリップチップ実装を想定したグリッド上の端子配置（F），ランダムに生成した端子配置（S）の4種類である．

表1に，配線問題のグリッドの大きさ，配線の本数，障害物の数，目標端子対集合に含まれるすべての目標端子対のマンハッタン距離の総和，その最大値，その最小値をそれぞれGrid，#Net，#Obst，D_{total}，D_{max}，D_{min}で示す．接続端子対の集合と目標端子対集合が一致するかを，提案アルゴリズムと既存の総配線長最小化，配線長差削減のステップ毎に示す．接続端子対の集合と目標端子対集合が一致する時は○，不一致の時は×である．

表2に提案アルゴリズムと既存手法の各ステップの，表3に本研究と既存研究で提案する配線設計全体の，総配線長，最大配線長，最小配線長，最大配線長差，実行時間（秒）をそれぞれL_{total}，L_{max}，L_{min}，δ_{max}，Tで示す．

図6に提案アルゴリズムによる例題E2，B2に対する配線結果を示す．

5.1 提案アルゴリズムの評価

表1に示されているように，提案アルゴリズムは目標端子対の接続を多くの例題で実現し，既存の総配線長最小化は目標端子対とは異なる端子対を接続することが多い．例題E2，B3は提案アルゴリズムを用いても，接続されない目標端子対が存在する．

表2に示されているように，提案アルゴリズムでは，既存手法と比較し，最大配線長L_{max}と最大配線長差δ_{max}が，B1，B2では大きく，B1，B2を除く例題では等しい．提案アルゴリズムでは多くの例題で，最大配線長と配線長差の小さい配線を生成することが分かる．

B1，B2の配線には蛇行した配線が存在している．どちらも障害物により，バウンディングボックスを内で目標端子対を接続できない．提案アルゴリズムでは，目標端子対のバウンディングボックス内の辺を優先的に使うため，バウンディングボックスの内と外とを出入りする蛇行した配線を生成している．

表2に示されているように，提案アルゴリズムでは，既存手法と比較し，計算時間Tが全ての例題で短い．しかし，目標端子対集合を手動で決定しているため，必ずしも公平な評価ではないことに注意されたい．

目標端子対配線アルゴリズムは，短い計算時間で目標端子対接続を実現し，最大配線長と配線長差の小さい配線を生成する．しかし，配線がバウンディングボックス内で配線不可能であるとき，改善の余地がある．

5.2 提案する配線設計フローの評価

表3に示されているように，提案する配線設計のフローでは，既存手法と比較し，配線長差δ_{max}が，例題B1，S1では小さく，E1, E2, B2，B3，F1では等しく，S2では大きい．提案する配線設計のフローは，既存研究で提案された配線設計と同等の配線長差の小さい配線を生成することが分かる．

S2の配線は配線された目標端子対のバウンディングボックスが含む辺を，他の配線が使っており，冗長化のための領域が存在しない．最短配線が最大配線と同等の長さまで冗長化されず，配線長差の大きい配線を生成している．

表3に示されているように，提案する配線設計のフローでは，既存手法と比較し，計算時間Tが全ての例題で短い．多くの計算時間を必要とする配線長差削減アルゴリズムを用いないため，計算時間が短い．

6 まとめ

本稿では，集合対間配線問題において，配線長差の小さい配線を効率よく得るための配線アルゴリズムを提案した．既存手法では，総配線長最小の配線を初期解に対し，効率よく配線長差を削減する修正方法を見つけることができず，多くの計算時間を必要としていた．初期解は，離れた端子対を接続する配線を含み配線長差が大きいことが多かった．提案するアルゴリズムは，近い端子対の集合を目標端子対集合として設定し，全ての目標端子対の接続の実現を目指し，他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を避けて配線を生成する．配線長差の小さい配線を初期解とし，効率よく生成することで，修正により配線長差の小さい配線を効率よく得る．計算機実験を行い，提案するアルゴリズムの有効性を確かめた．今後の課題として，目標端子対集合の具体的な決定アルゴリズムの考案や，目標端子対接続が実現されない例題に対する配線アルゴリズムの考案が挙げられる．

参考文献

[1] 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，第111巻，pp.23–28，Sep. 2011.

表 1: 目標端子対集合と接続結果

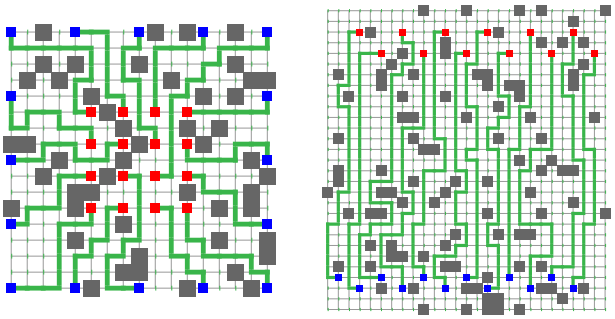
Instance	Grid	#Net	#Obst	D_{total}	D_{max}	D_{min}	提案手法	総配線長最小化	既存手法
E1	16×16	16	0	136	10	8	○	×	○
E2	16×16	16	50	142	16	8	×	×	×
B1	16×19	6	50	81	17	10	○	×	○
B2	26×28	12	100	294	25	24	○	×	×
B3	26×28	12	100	288	24	24	×	×	×
F1	26×26	12	120	135	19	4	○	×	○
S1	9×9	6	10	44	12	3	○	×	○
S2	19×19	8	60	89	16	6	○	×	○

表 2: 提案手法と総配線長最小化，既存手法適用結果

	提案手法					総配線長最小化					既存手法				
Instance	L_{total}	L_{max}	L_{min}	δ_{max}	$T[sec]$	L_{total}	L_{max}	L_{min}	δ_{max}	$T[sec]$	L_{total}	L_{max}	L_{min}	δ_{max}	$T[sec]$
E1	136	10	8	2	0.0091	136	10	6	4	0.0372	136	10	8	2	0.6164
E2	152	16	6	10	0.0138	146	18	6	12	0.0211	146	16	6	10	0.2350
B1	95	18	13	5	0.0023	93	19	10	9	0.0519	93	17	13	4	0.0722
B2	348	35	25	10	0.0272	338	35	23	12	0.1896	338	30	25	5	5.1226
B3	368	46	26	20	0.0374	356	46	22	24	0.2383	356	46	26	20	4.1536
F1	132	19	3	16	0.0087	132	23	3	20	0.0910	132	19	3	16	1.3462
S1	44	12	3	9	0.0009	44	17	3	14	0.0077	44	12	3	9	0.0038
S2	91	16	6	10	0.0030	91	22	5	17	0.0469	91	16	6	10	0.2253

表 3: 提案する配線設計のフロー適用結果

	目標端子対配線+最短配線冗長化					既存手法+最短配線冗長化				
Instance	L_{total}	L_{max}	L_{min}	δ_{max}	$T[sec]$	L_{total}	L_{max}	L_{min}	δ_{max}	$T[sec]$
E1	160	10	10	0	0.0099	160	10	10	0	0.6542
E2	156	16	8	8	0.0140	154	16	8	8	0.2565
B1	105	18	17	1	0.0026	97	17	15	2	0.1243
B2	414	35	34	1	0.0310	354	30	29	1	5.3126
B3	438	46	34	12	0.0411	420	46	34	12	4.3955
F1	192	19	15	4	0.0113	196	19	15	4	1.4396
S1	48	12	5	7	0.0009	46	12	4	8	0.0115
S2	103	16	10	6	0.0035	117	16	14	2	0.2732



E2　　B2

図 6: 提案アルゴリズムの配線結果

[2] W.-H. Liu, M.-S. Chang, and T.-C. Wang, “Floorplanning and signal assignment for silicon interposer-based 3D ICs,” Proc. Design Automation Conference (DAC), pp.1–6, 2014.

[3] M. R. Garey and D. S. Johnson, Computers and Intractability, A Guide to the Theory of NP-Completeness, Freeman and Co., New York, 1979.

[4] Y. Nakatani and A. Takahashi, “A length matching routing algorithm for set-pair routing problem,” IEICE Trans. Fundamentals, vol.98, no.12, pp.2565–2571, 2015.

[5] Y.-K. Ho and Y.-W. Chang, “Multiple chip planning for chip-interposer codesign,” Proc. Design Automation Conference (DAC), pp.1–6, 2013.

[6] T.C. Hu, “Integer programming and network flows,” Technical report, DTIC Document, 1969.

[7] J. A. Bondy and U. S. R. Murty, Graph Theory with Applications, North-Holland, Amsterdam, 1976.

[8] 久保ゆき子，高島康裕，中武繁寿，梶谷洋司，“Flip により自己変換するスタイナ木とその VLSI 最適配線への応用，” 情報処理学会論文誌，第 41 巻，pp.881–888，Apr. 2000.

[9] Y. Kohira, S. Suehiro, and A. Takahashi, “A fast longer path algorithm for routing grid with obstacles using biconnectivity based length upper bound,” IEICE Trans. Fundamentals, vol.E92-A, no.12, pp.2971–2978, 2009.