修 士 論 文

配線長を考慮した半順序制約付き シーケンスペアによるモジュール配置

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

矢野 勇生

2007 年 3 月

修 士 論 文

配線長を考慮した半順序制約付き シーケンスペアによるモジュール配置

指導教官　金子峰雄 教授

審査委員主査　金子峰雄 教授
審査委員　宮地充子 助教授
審査委員　平石邦彦 教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

510105 矢野 勇生

提出年月: 2007 年 2 月

概 要

LSI（Large Scale Integration）は，情報処理や信号処理を行う上で欠かせない主要要素であり，情報通信基盤設備から個人ユースのPC，モバイル機器，デジタル家電まで，広く計算，通信，制御機器に内蔵されている．微細化技術の進展に伴って複雑化したLSIは，高速化，低消費電力化などの要求に加え，製造時に生じるばらつきの考慮や，テスト容易化設計への対応を求められるなど，その設計は困難さを増している．このため，効率的なLSI設計のためのCAD（Computer Aided Design）ツールに関する研究は重要なテーマとなっている．

LSIのレイアウト設計における部分問題の一つとして，モジュール配置問題がある．これは，各モジュールの二次元平面内における配置座標を決定するものであり，モジュール数をnとするとき，その解空間は$\mathbb{R}^{2n}$となる．このモジュール配置問題に対して，これまでに力学的手法，再帰的二分割法，解析的手法などの様々な手法が提案されてきた．こうした中，村田らによって，シーケンスペアと呼ばれる配置表現コードを用いた配置最適化手法が提案された．このコードは，絶対座標の情報を持たず，モジュール名の順列の対によってモジュール同士の相対位置関係のみを表現するものであり，これによりモジュール配置の解空間のサイズを$(n!)^2$としている．また，配置の最適化はシーケンスペアによって表現された解空間を，Simulated Annealing（SA）やGenetic Algorithmなどの探索アルゴリズムを用いて探索することにより実現された．

このコード表現とSAを組み合わせた手法の特長として，目的関数の選び方によって多種の配置問題に柔軟に対応できることや，十分に時間をかけることで良質な解を生成できることなどが挙げられる．一方でシーケンスペアの解空間サイズ$(n!)^2$は，大規模な回路へこの手法を適用する場合，実用時間内に良質な解を得ることを困難ならしめている．この問題に対する解として，シーケンスペアのデコードアルゴリズムの高速化や，シーケンスペアそのものの解空間縮小などが考えられる．本研究では，後者の考え方に基づき，探索を効率化することで大規模回路へも適用可能な配置最適化手法を提案する．

異なるシーケンスペアコードは必ず異なる相対配置を表現することから，解空間を縮小するためには，シーケンスペアが表す配置の中から，探索に不要な配置を見つけ出し，そのコードを解空間から除外する必要がある．しかし，どのコードも具体的なインスタンスが与えられない限り，そのコードが表す配置の要不要を判断することはできない．そこで本研究では，インスタンスのネット情報を活用し，配線長にとって好ましくないコードを解空間から除外することで，解空間の縮小を図る．力学的手法と呼ばれる配置手法はモジュール同士の重なりを許すものの，二次配線長評価の下での最適な配置をモジュール数に関する多項式数回の四則演算にて求めることができる．得られた配置を，コードの要不要を判断する指標となるモデル配置とし，このモデル配置とモジュール同士の相対位置関係が大きく異なる配置を，解空間から除外することを基本方針とする．

モデル配置が持つモジュールの座標情報から，モジュール同士の相対位置関係が得られる．二つの順列の対からなるシーケンスペアに対し，順列内のモジュールの出現順序を制約することでモデル配置のもつ理想的な相対位置関係を解空間に反映させることができる．本研究では，このような制約が付与されたシーケンスペアを POSP（Partially Ordered Sequence-Pair）と名づけ，これによって定義される解空間内のみを SA にて探索する手法を提案する．

本研究では先ず，モデル配置からシーケンスペア上の制約への具体的な変換手法を明らかにした後，縮小された解空間を SA にて探索するための準備を行った．すなわち，初期 POSP コード生成，隣接解定義を行い，この POSP 解空間が可到達性（任意の POSP コードから他の任意の POSP コードへと POSP コードのみを通って到達できる）を有することを証明した．次いで，提案手法を評価するため，ベンチマーク回路を用いた配置実験を行い，以下の事柄を確認した．

1. 力学的手法によるモデル配置導出の際，モジュール同士の重なり除去操作を適切に行うことで，モデル配置としての有効性が高まる．
2. 制約を付与する対象を，モジュール間距離の遠い 2 モジュールから順に選んだとき，その付与数が解の質に影響する．一般に，回路規模と比べ総探索数が少ない場合，付与する制約を多くし，また一方，回路規模と比べ総探索数が十分にある場合，付与する制約を少なくすることで良質な解が得られる．
3. 従来手法と比較して，適用する回路の規模に対し総探索数が少ない場合ほど，二次配線長評価の良い配置を探索することができる．また，実行時間はいずれの実験においても 10% ほどの増加となっている．

今後の課題として，POSP に適した隣接解生成方法の検討，配線長のバウンディングボックス評価に適したモデル配置，制約付与方法の提案，解空間の縮小効果に関するより詳細な検証などが挙げられる．

目 次

第1章　まえがき

1.1　研究背景

情報処理や信号処理を行う上で欠かせない主要要素であり，情報通信基盤設備から個人ユースのPC，モバイル機器，デジタル家電まで，広く計算，通信，制御機器に内蔵されている．半導体の微細化技術進展に伴って複雑化したLSIは，低消費電力化，高速化などの要求に加え，製造時のばらつきの考慮や，テスト容易化設計への対応を求められるなど，その設計は困難さを増している．このため，効率的なLSI設計のためのCAD（Computer Aided Design）ツールに関する研究は重要なテーマとなっている．

LSIのレイアウト設計は，CAD分野の中でも最も早い時期から研究が行われてきた分野である．この部分問題の一つとして，モジュールの配置座標を決定するモジュール配置問題がある．これは，各モジュールの二次元平面上での座標を決定するもので，モジュール数をnとするとき，その解空間は$\mathbb{R}^{2n}$となる．この問題に対しこれまでに，力学的手法，再帰的二分割法，解析的手法などの様々な手法が提案されている[1].

こうした中，村田ら[2]によって，シーケンスペアと呼ばれる配置表現コードを用いた配置最適化手法が提案された．このコードは，絶対座標の情報を持たず，モジュール名の順列の対によってモジュール同士の相対位置関係のみを表現するものであり，これによりモジュール配置の解空間のサイズを$(n!)^2$としている．また，配置の最適化はシーケンスペアによって表現された解空間を，Simulated Annealing（SA）やGenetic Algorithmなどの探索アルゴリズムを用いて実現された．

このコード表現とSAを組み合わせた手法の特長として，目的関数の選び方によって多種の配置問題に柔軟に対応できることや，十分に時間をかけることで良質な解を得られることなどが挙げられる．一方でシーケンスペアの解空間サイズ$(n!)^2$は，大規模な回路へこの手法を適用する場合，実用時間内に良質な解を得ることを困難ならしめている．

この問題に対し，これまでにシーケンスペアのデコードアルゴリズムの高速化[3, 4, 5]や，探索時の隣接解生成に遷移確率を与える手法[6]などの研究が行われてきた．LSIの複雑化が進む今日，更なる高速化，探索効率化の手法が求められている．

1.2　目的

本研究では，より大規模な回路への適用を可能にするため，シーケンスペアの解空間を縮小し，SAによる探索を効率化する．探索の効率化は，配置最適化アルゴリズム全体の

計算時間短縮につながるため，その分扱うモジュール数を増やしても実用時間内に解を求めることが可能となる．

提案するアルゴリズムは，シーケンスペアにおいて探索するべき解空間解空間を縮小する．解空間の縮小には，回路のネット情報を用いる．ネット情報を利用することで，あらかじめモデルとなる理想的配置を求め，そこからモジュール同士の相対位置関係を抽出する．抽出された相対位置関係をシーケンスペアに半順序制約として反映することで，解空間を縮小し，探索を効率化する．

本論文の構成は以下の通りである．第二章では，本研究で扱うモジュール配置問題の定式化を行い，さらにシーケンスペアと SA を用いた配置最適化について説明する．第三章で，モデル配置に基づいた半順序制約付きシーケンスペアPOSP と，その解空間についての考察を行う．第四章で，モデル配置の導出として用いる力学的手法について説明する．第五章では，提案手法を実装し，従来手法との比較を行う．ベンチマーク回路を用いて配置実験を行い，その結果と考察を述べる．第六章でまとめと今後の課題について述べる．

第2章　シーケンスペアを利用したSAによる配置最適化

本研究では、配置をシーケンスペアと呼ばれるコードを用いて表す．このコードにより，モジュール同士の相対位置関係を規定し，それによって表現される解空間を Simulated Annealing(SA) と呼ばれる繰り返しアルゴリズムによって探索する．

本章ではまず，本研究で扱う問題を定式化し，次にシーケンスペアと SA について説明を行う．

2.1　問題の定式化

本研究で扱う問題は，二次元平面内での矩形モジュール配置問題である．最適化の目標は，モジュール同士の重なりがない配置を実現しながら，後述するコスト関数 C を最小化することである．

- **入力**：モジュール情報（高さ，幅），外部端子情報（端子位置），接続関係情報（ネットリスト）
- **出力**：各モジュールの配置座標
- **評価**：面積コスト，配線コスト

2.2　シーケンスペア

2.2.1　シーケンスペアによる相対位置関係の表現

シーケンスペアは，1996 年に村田ら [2] によって提案された配置のコード表現法である．．シーケンスペアは，モジュール間の相対位置関係を表し，そのコードはモジュールを要素とする二つの順列 Γ_+，Γ_- の組み合わせからなる．二つのモジュール間の相対位置関係は，次のようにシーケンスペア上での出現順序によって定められる．

$$(\Gamma_+, \Gamma_-) = (\ldots a \ldots b \ldots, \ldots a \ldots b \ldots) \Rightarrow b \text{ は } a \text{ の右} \tag{2.1}$$

$$(\Gamma_+, \Gamma_-) = (\ldots b \ldots a \ldots, \ldots a \ldots b \ldots) \Rightarrow b \text{ は } a \text{ の上} \tag{2.2}$$

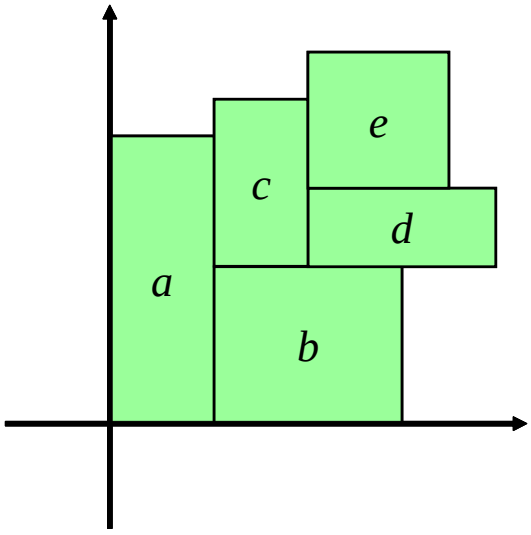

図 2.1: $(\Gamma_+, \Gamma_-) = (acedb, abcde)$ の左下詰め配置例

例として，$(\Gamma_+, \Gamma_-) = (acedb, abcde)$ に対応する左下詰め配置を図 2.1 に示す．

ここで，シーケンスペアの相対関係の表現方法について注意すべき点がある．$L(x)$，$R(x)$ をそれぞれモジュール x の左端，右端の座標を表すとする．シーケンスペアのコードにおいて二つのモジュール a, b が式 (2.1) のように表されたとき，実際の配置では，$R(a) \leq L(b)$ であることのみを意味し，y 軸上の相対位置については何も規定しないものとする．同様にモジュール同士が上下の関係にあるとき，左右の位置関係については規定されない．

2.2.2 配置座標からのシーケンスペアコード生成

配置から対応するシーケンスペアのコードを求める手法として，グリッディングと呼ばれる手続きが提案されている．これは，二次元平面上に配置された各モジュールからポジティブステップライン，ネガティブステップラインと名づけられた二種類の階段状の線を引くことを基底とするものである．

一つのポジティブステップラインは，一つのモジュールの右上端，左下端からそれぞれ配置エリアの右上端，左下端へと伸びる階段状の区分的直線である．一つのモジュールに対するポジティブステップラインは，

- 他のモジュール
- 他のモジュールに対するポジティブステップライン
- チップエリアの境界

と交わってはならない．

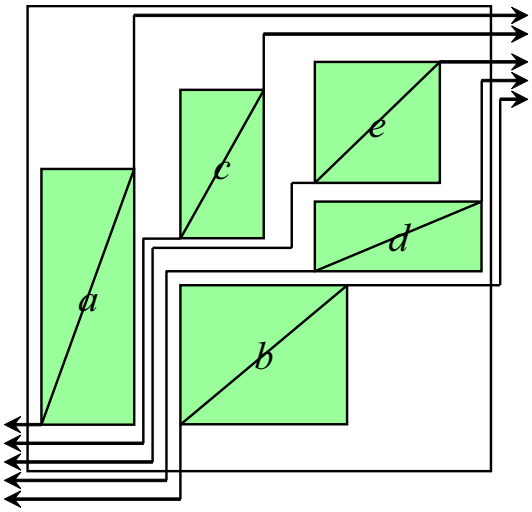

図 2.2: ポジティブステップライン

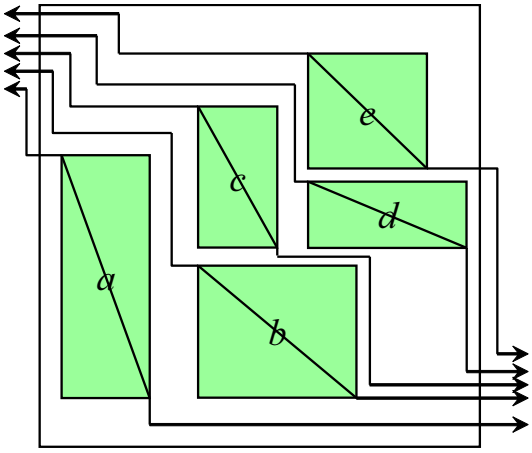

図 2.3: ネガティブステップライン

モジュールの右上端（左下端）を出発したポジティブステップラインは，上述した三つの障壁に衝突しない限り，右（左）に進み，衝突した時点で上（下）へと進む．障壁が無くなれば再度右（左）へと進み，配置エリアの隅へと到達するまでこれを繰り返す．

一方ネガティブステップラインは，各モジュールの左上端，右下端からそれぞれ配置エリア左上端，右下端へと伸びる階段状の直線である．ポジティブステップラインと同様に左右方向への進行を優先し，ステップラインが配置エリア左上端，右下端へと到達すると終了する．例として図 2.1 の配置に対し，ポジティブステップライン，ネガティブステップラインを引いたものをそれぞれ図 2.2，図 2.3 に示す．

各モジュールから伸びたステップラインをモジュール名でラベリングしたとき，この並び順が配置に対応するコードとなっている．すなわち，ポジティブステップラインのラベルを上から順に並べたものが Γ_+，ネガティブステップラインのラベルを下から順に並べ

たものがΓ_-となる．

2.2.3　シーケンスペアからの配置座標計算

シーケンスペアのコードからのモジュール配置座標の計算について説明する．前述したように，シーケンスペアのコードはモジュール同士の相対的な位置関係を規定する．したがって，これをモジュール同士に課せられた制約とみなし，よって,「シーケンスペアから定まる制約を満たした上で，最小のパッキングを行う」ことによって配置座標を計算する．

モジュール数をnとし，$(n \times n)$の格子グラフを考える．格子の垂直方向，水平方向への線にそれぞれΓ_+，Γ_-に出てくる順番のモジュール名をラベリングする．さらに，この格子グラフを時計方向に45 °回転させ，同じラベルxをもつ垂直方向，水平方向の線分が交差する点にモジュールxを配置する．こうして作成された傾斜格子グラフの例を図2.4に示す．この傾斜格子グラフから抽出される左右関係，上下関係に基づいて，水平制約グラフ$G_h(V, E)$及び垂直制約グラフ$G_v(V, E)$を作成する．

水平制約グラフ$G_h(V, E)$は以下のようにして構成される（図2.5参照）．

1. 頂点集合V：
   ソースs,
   シンクt,
   モジュール名でラベリングされたn個の頂点$x_1, x_2, \cdots, x_n$

2. 辺集合E：
   $(s, x_i),\ 1 \leq i \leq n$,
   $(x_i, t),\ 1 \leq i \leq n$,
   $(x_i, x'_j),\ x_j$がx_iの右であるとき

3. 頂点重み：
   $\lambda : V \to \mathbb{R}$,　ただし$\lambda(s) = \lambda(t) = 0, \lambda(x_i) = w_i$（モジュール幅）

同様にして垂直制約グラフ$G_v(V, E)$も構成することができる（図2.6参照）．上下の関係にある頂点同士に辺を設け，頂点重みにはモジュールの高さを用いる．

水平制約グラフG_h，垂直制約グラフG_vにおいて，sから各頂点への最長パス長を求めることにより，モジュールのx座標，y座標が定まり，sからtへの最長パス長がチップ全体の幅と高さとなる．この計算は$O(n^2)$時間で実行することができる．

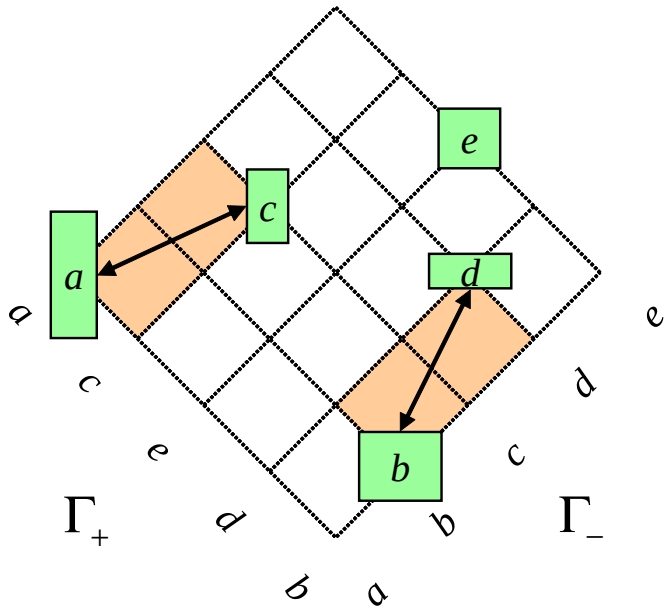

図 2.4: 傾斜格子グラフ上のパッキング

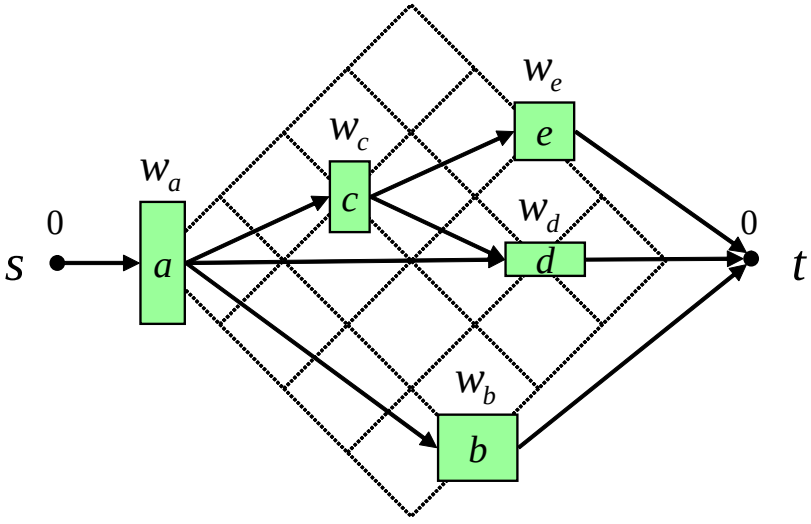

図 2.5: 水平制約グラフ

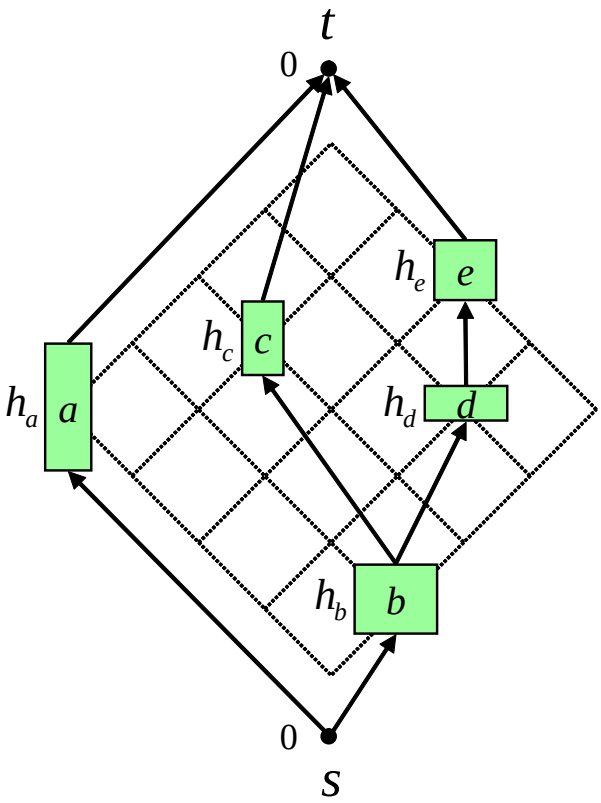

図 2.6: 垂直制約グラフ

2.3 Simulated Annealing(SA)

2.3.1 アルゴリズムの流れ

SA は，暫定解から別の解 (隣接解) を生成する操作を繰り返すことで，解空間を探索する．生成された隣接解はその都度コスト関数によって評価され，現在保持している暫定解と比較し，受理（その暫定甲斐を隣接解にて置き換える）不受理を決定する．ただし，生成された隣接解の評価が暫定解と比べて悪くとも確率的にそれを受理することにより，局所的な最適解に捕らえられることを防いでいる．

SA には，温度 t，温度降下係数 d_c，内部ループ数 lp，といったパラメータが存在する．まず，温度について高温の状態 $t = t_{start}$ から探索を始める．隣接解の生成は一つの温度につき lp 回行う．現在の温度 t に降下係数 d_c を乗じたものを次の温度とし，徐々に温度を下げながら終了温度 t_{end} に達するまで続けられる．暫定解と比較し解が悪化している場合は，確率 $P = \exp(-\Delta c/t)$ にてこれを受理する．ただし，Δ_c は暫定解と隣接解の評価の差の絶対値である．

ある時点での暫定解を S，そこから生成した隣接解を S' とする．解のコスト関数を C とすると，SA の隣接解評価手順は以下のように示される．

- $C(S) - C(S') \geq 0$ の場合
  - 暫定解を S' に更新する
- $C(S) - C(S') < 0$ の場合
  - if $\left(\exp(\frac{C(S)-C(S')}{t}) \geq rand(0,1)\right)$
    - 暫定解を S' に更新する
  - else
    - 元の暫定解から新たに別の隣接解を生成する

SA 全体のフローチャートを図 2.7 に示す.

2.3.2 シーケンスペアにおける隣接解生成法

シーケンスペアのコードは順列の対で構成されるため，その隣接解生成は順列の並びの入れ換えによって行われる．基本的な隣接解生成操作を以下に示す.

- **挿入**：任意の一つのモジュールを一つの順列から取り出し，別の位置に挿入する
- **交換**：任意の二つのモジュールの一つの順列内の位置を入れ換える
- **全交換**：任意の二つのモジュールについて，それらの位置を Γ_+，Γ_- 両方で入れ換える

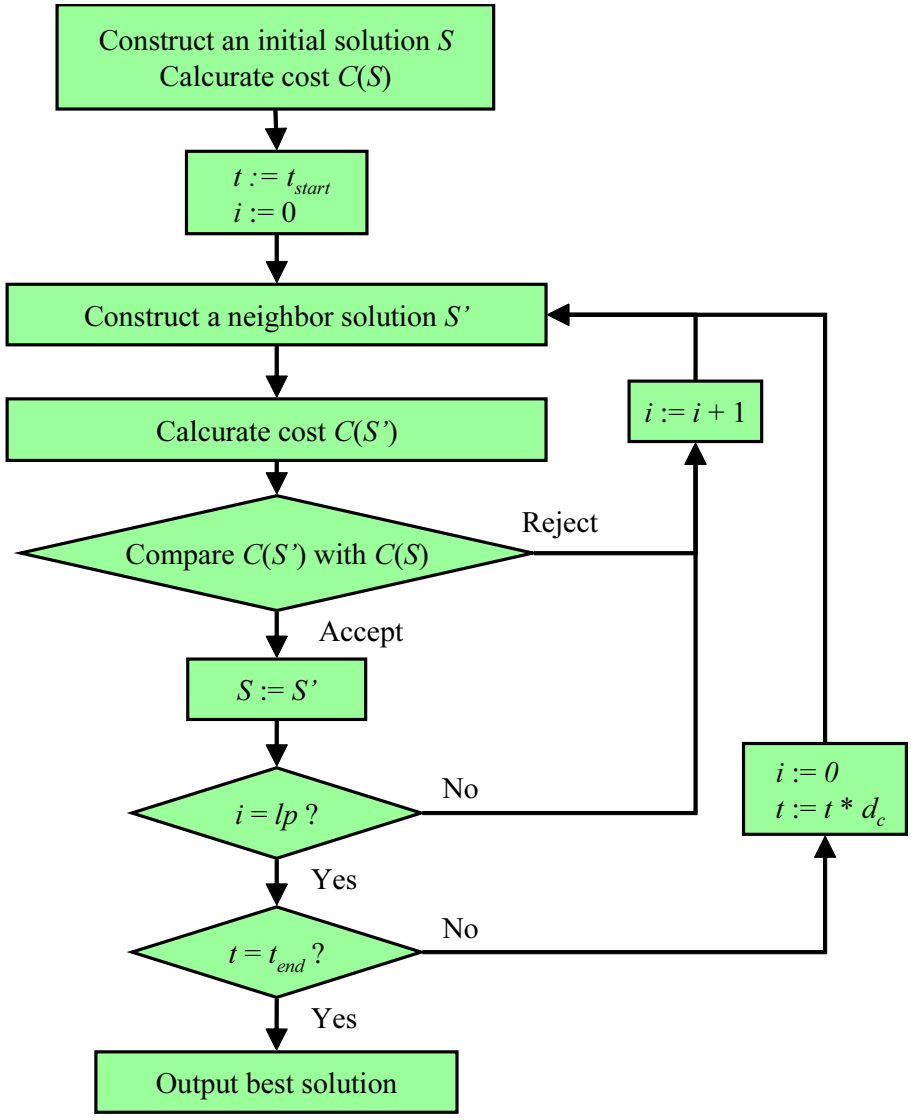

図 2.7: SA のフローチャート

また，シーケンスペアそのものに対する操作ではないが，モジュールの向きを 90 °変える**回転**や，上下や左右を入れかえる**反転**といった操作も存在する．

第3章　モデル配置に基づく解空間縮小

3.1　方針

シーケンスペアは，モジュール数を n としたとき $(n!)^2$ のサイズの解空間を持つ．このため，モジュール数の増加に伴ってその解空間が急速に増加し，良質な解を得るまでの時間が急激に増加してしまう．この問題を解決する一つの手法として，シーケンスペアの解空間縮小が挙げられる．例えば極めて大きな面積を必要とする配置や，極端に長い配線を必要とする配置などを，あらかじめ解の候補から除外することができれば，探索効率の改善につながると考えられる．

本研究では，まず初めに，モデルとなるある理想的な配置を導出し，次にその配置に似た配置のみを探索すべく，シーケンスペアに制約を加えることで，元のシーケンスペアの解空間から新しく縮小された解空間を定義する．

3.2　Partially Ordered Sequence-Pair(POSP)

今，モジュールの重心座標がモデル配置として与えられているとする．このとき座標情報からシーケンスペア上での制約の抽出を行う方法について考える．例として，図 3.1（a）の位置関係にあるモジュール対 a, b から制約を抽出する．a の重心位置は b の重心位置の右上，すなわち「b の右，かつ上」にある．シーケンスペアはモジュール間に，上下もしくは左右，どちらか一方の関係しか定義できないため,「右かつ上」という関係を厳密に表すことはできない．しかし，この関係を緩和し，a は「b の右，または上」と定義する（図 3.1（b））と，その相対位置関係は次のように表される．

$$(\Gamma_+, \Gamma_-) = (\ldots b \ldots a \ldots, \ldots b \ldots a \ldots) \tag{3.1}$$

または

$$(\Gamma_+, \Gamma_-) = (\ldots a \ldots b \ldots, \ldots b \ldots a \ldots) \tag{3.2}$$

このように，どちらのシーケンスペアも Γ_- におけるモジュール a, b の出現順序は等しく $''ba''$ となる．この出現順序を a, b 間の緩和された制約と見なすことができる．

同様にして,「a は b の左，かつ上」を「a は b の左，あるいは上」,「a は b の右，かつ下」を「a は b の右，あるいは下」,「a は b の左，かつ下」を「a は b の左，あるいは下」と緩和した場合も，Γ_+, Γ_- どちらか一方の順列において a, b の出現順序が等しくなる．二

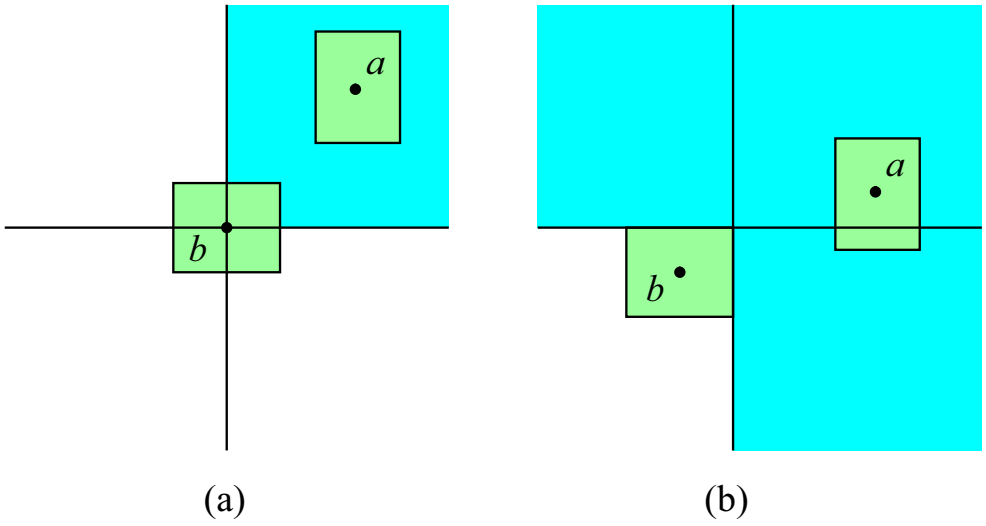

図 3.1: 相対位置関係の緩和

表 3.1: モデル配置の相対位置関係と半順序制約

相対位置関係	半順序制約
a は b の右，かつ上	$\Gamma_-^{-1}(a) > \Gamma_-^{-1}(b)$
a は b の左，かつ上	$\Gamma_+^{-1}(a) < \Gamma_+^{-1}(b)$
a は b の左，かつ下	$\Gamma_-^{-1}(a) < \Gamma_-^{-1}(b)$
a は b の右，かつ下	$\Gamma_+^{-1}(a) > \Gamma_+^{-1}(b)$

つのモジュール座標が等しい場合を除き，全てのモジュールは他のモジュールとの間に緩和されたシーケンスペア上の制約を得ることができる．この制約はΓ_+, Γ_-上のモジュール出現順序に対する半順序関係となる．それぞれの半順序関係を守ったモジュール順列の対のみをコードとして採用することにより，シーケンスペアの解空間を縮小する．本研究では，このようにして半順序制約を課せられたシーケンスペアを，Partially Ordered Sequence-Pair と名づけ，以降 POSP と表記する．

表 3.1 に，モデル配置の相対位置関係に対応する半順序制約を示す．ただし，$A(x)$ は順列 A での x 番目にあるモジュール，$A^{-1}(b)$ は順列 A でのモジュール b の位置を表すものとする．

3.3 可到達性

前節において，モデル配置からの制約抽出方法について述べた．本研究では，半順序制約により制限されたシーケンスペア POSP の解空間内を SA により探索することを考えて

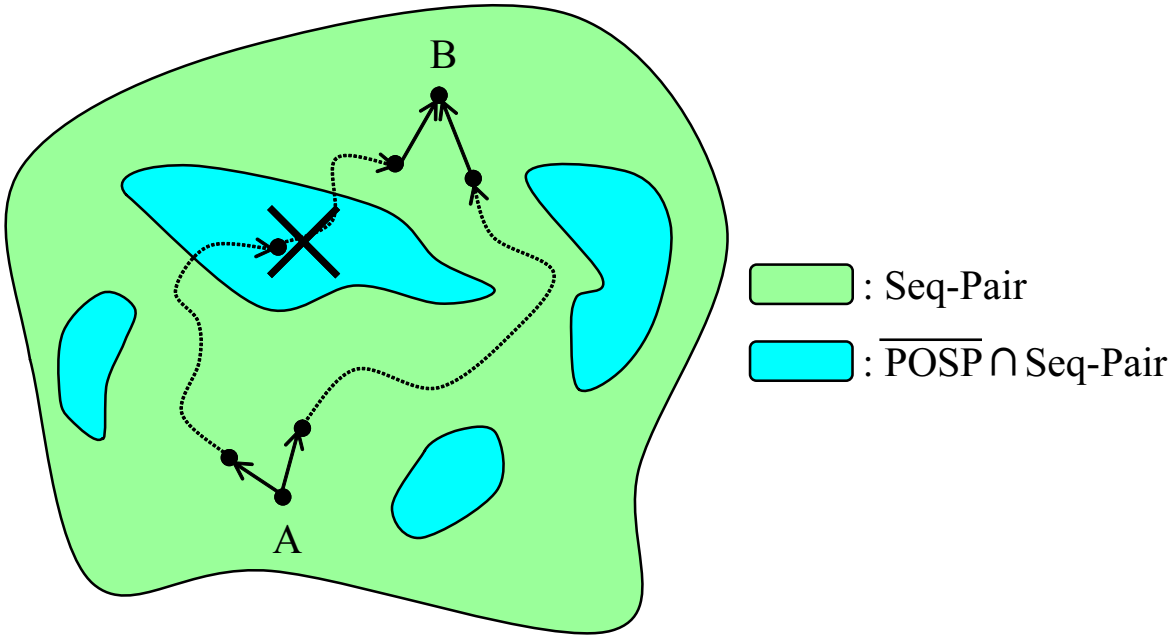

図 3.2: シーケンスペアの解空間と可到達性

おり，探索における解の可到達性について確認する必要がある．

> **可到達性**：
> 問題のサイズの多項式回の隣接回生成操作によって，任意の許容解から任意の許容解まで，許容解のみを経由し，到達できること．

この性質が満たされない場合，初期解によっては十分に時間をかけても最適解に到達できないといった現象が生じてしまう．

隣接解生成操作である挿入，交換操作に関する可到達性について考える．制約の無いシーケンスペアにおいては，この二つの操作が可到達性を満たすことは明らかである．二つの順列の対からなるシーケンスペアは，モジュール数 n としたとき，高々$2(n-1)$ 回の挿入操作，もしくは交換操作を繰り返すことで任意の解から任意の解へと移動できる．

次に，POSP についての可到達性を考える．これは即ち，POSP の解空間内の任意の解から任意の解まで，POSP の解空間内のみを通過し，モジュール数の多項式時間で到達できるかどうかを意味する．この性質が満たされない場合，十分に長い時間をかけて探索しても最適解に到達できない可能性が生じてしまうため，好ましくない．例として，図 3.2 の解空間内を任意の解 A から B へと移動することを考える．右側の経路は POSP の解空間内のみを通過しているが，左側の経路では，POSP 以外の解空間内を通過しているため，POSP で定められた半順序制約に違反している．

ここで，POSP の解空間に関して次の定理が成り立つ．

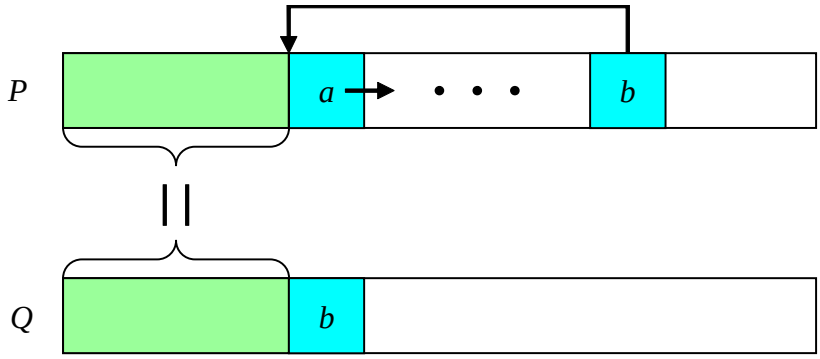

図 3.3: 半順序制約下での挿入操作による移動

Theorem 1
半順序制約が与えられたシーケンスペアは，モジュール数を n として，高々 $2(n-1)$ 回の挿入操作によって任意の許容解から任意の許容解へと，許容解のみを経由して到達することができる．

[**証明**]
制約を満たした任意の 2 つの順列を P, Q とし，P から Q への移動を考える．まず P, Q を先頭から順に比較し，モジュール名の順列の中で最初に異なるモジュールが出てくる列を i，またそのモジュール名を $P(i) = a, Q(i) = b$ とする．この時 $(i-1)$ 番目まで P, Q は等しいことから，$P^{-1}(b) > i, Q^{-1}(a) > i$ がわかる．

P 中のモジュール b を挿入操作によって i 番目に移動することを考える．Q において b は i 番目の位置にあるので，P において b と i 番目以降のモジュールとの間に制約は無い．よって b は P において a の位置，すなわち i 番目に挿入可能である．この操作によって P と Q の共通部分は一つ以上増える．以降同様に，次に異なるモジュールがある部分を探し，挿入操作を繰り返すことで任意の順列から任意の順列へと移動する（図 3.3）．

シーケンスペアは二つの順列の対からなるので，高々 $2(n-1)$ 回の挿入操作により任意の許容解から任意の許容解へと，許容解のみを通って到達することができる．□

また，交換操作の可到達性も成り立つ．

Theorem 2
半順序制約が与えられたシーケンスペアは，モジュール数を n として，高々 $\frac{n(n-1)}{2}$ 回の交換操作によって任意の許容解から任意の許容解へと，許容解のみを経由して到達することができる．

[**証明**]
定理 1 の証明と同様に，制約を満たした任意の順列 P, Q の P から Q への移動を考える．モジュール名の順列の中で，最初に異なるモジュールが出てくる列を i，そのモジュール名を $P(i) = a, Q(i) = b$ とする．a と b の交換操作を考えると，$(i+1)$〜$(j-1)$ 番目のモ

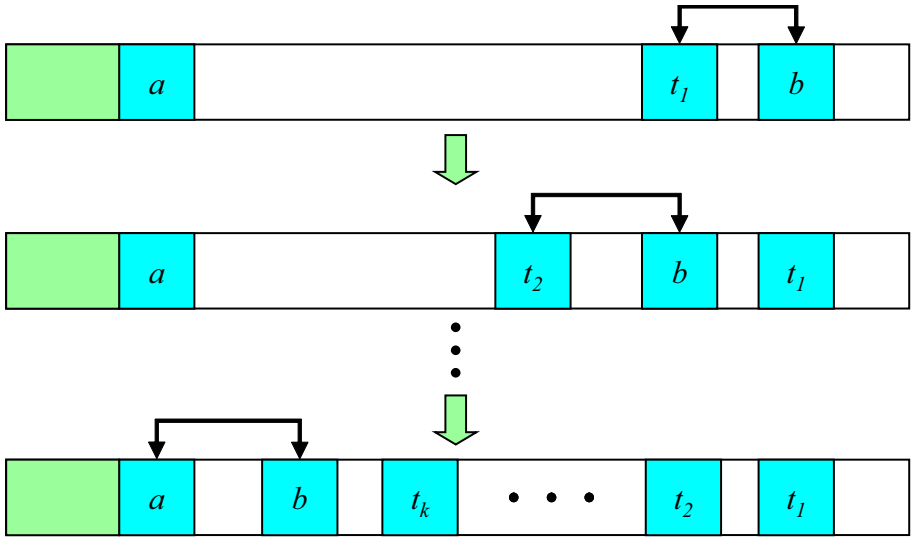

図 3.4: 半順序制約下での交換操作による移動

ジュールの中に，半順序制約 $P^{-1}(a) < P^{-1}(t_k)$ を持つモジュール t_k が存在する可能性がある．この場合，交換操作を行うと制約に違反してしまうため，a と b の交換は行えない．

そこで，t_k と b の交換を考える．この操作を行うと，たとえ a と b の交換を行っても a と t_k との間に制約の違反は生じない．したがって，以下に示す手順で交換操作を行うことにより，順列 P から Q への移動を行う（図 3.4）．

1. 順列 P, Q の，先頭から見て最初に異なる部分を探す
2. 交換したいモジュール対 a, b の間にあり，a との間に $P^{-1}(a) < P^{-1}(t_j)$ の制約を持つモジュール t_j を，順列の後方から順に探す
3. t_j と b との交換操作を行う
4. t_j が見つからなくなるまで 2.〜3. を繰り返す
5. a と b の交換を行う
6. P, Q が一致するまで 1.〜5. を繰り返す

以上の手続きで最も交換操作の回数が多い場合は，$\sum_{l=1}^{n-1}(n-l) = \frac{n(n-1)}{2}$ となる．□

これまでの議論を踏まえ，本研究ではシーケンスペアに対する隣接解の生成操作として，挿入，交換，全交換の三種類を用いることとする．また，これ以外にモジュールの向きを 90 °変える回転操作も導入する．

第4章　二次配線長評価に基づくモデル配置の導出と解空間縮小

4.1　方針

前章で提案した半順序制約付きシーケンスペア POSP を導入するには，何らかの指標に基づいたモデル配置を，事前に求めておく必要がある．しかし，どのようなシーケンスペアのコードであっても，インスタンスの情報なしに要・不要を判断することはできない．なぜならば，どのようなコードが与えられてもそれが最適であるようなモジュール集合のインスタンスが存在し，同様にそれが最適であるようなネット集合のインスタンスが存在するからである．例として，図 2.1 のシーケンスペア $(\Gamma_+, \Gamma_-) = (acedb, abcde)$ を考えると，図 4.1 のようなモジュールのインスタンスが与えられた時，このコードは明らかに面積最小である．したがって，解空間の縮小にはインスタンスのサイズ情報やネット情報を陽に用いることが必要となる．

本研究では，インスタンスのネット情報を利用することで，配線長の面で理想的な配置を導出，シーケンスペアに制約を課すことで不要な解を除外し，POSP による新しい解空間を定義する．

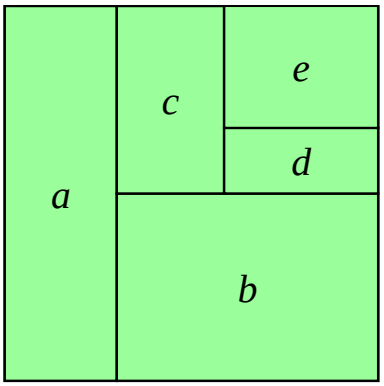

図 4.1: $(\Gamma_+, \Gamma_-) = (acedb, abcde)$ のコードをもつ面積最小の配置例

4.2　提案手法の流れ

提案する配置最適化アルゴリズムは，次の三つのステップからなる．

- **ステップ１：モデル配置の導出**
  – 力学的手法により，二次配線長を考慮した配置を算出する．
- **ステップ２：制約の抽出**
  – 求めた位置関係からシーケンスペア上の半順序制約を抽出し，POSP による解空間を定義する．
- **ステップ３：SA による解探索**
  – 制限された解空間を SA にて探索し，準最適配置を得る．

4.3　力学的手法

本研究では，力学的手法によって求められた配置をモデル配置として利用し，POSP の解空間を定義する．力学的手法は，モジュールや外部端子同士の接続をばねのようにみなし，ばねの復元力を働かせ，モジュールを力学的な安定点に移動させることにより配置を求める．この手法はモジュールを質点として扱うため，モジュール同士の重なりを許してしまうという欠点はあるが，二次配線長最小となる配置をモジュール数の多項式数回の四則演算で求めることができるという特長がある．

4.3.1　ネットから復元力への変換

k 個のモジュールを接続するネット p について，スタイナー点を一つ持つスター配線を考える．$(x_i, y_i), 1 \leq i \leq k$ を端子座標，(x_p, y_p) をスタイナー点の座標とおく（図 4.2 参照）．

次に，そのスター配線の二次配線長 L_p を次式 (4.1) のように定義する．

$$L_p = \sum_{i=1}^{k} \Big((x_p - x_i)^2 + (y_p - y_i)^2 \Big) \tag{4.1}$$

ここで，L_p を最小化するようにスタイナー点の座標を決めると，

$$x_p = \frac{\sum_{i=1}^{k} x_i}{k},\ \ y_p = \frac{\sum_{i=1}^{k} y_i}{k} \tag{4.2}$$

x_p を再び式（4.1）に代入して x 方向成分のみを取り出すと，x 方向の二次配線長 L_{p_x} が求まる．

$$\therefore L_{p_x} = \sum_{i=1}^{k} \Big(\frac{k-1}{k} \Big) x_i^2 - \sum_{i=1}^{k} \sum_{j=1, j\neq i}^{k} \frac{x_i x_j}{k} \tag{4.3}$$

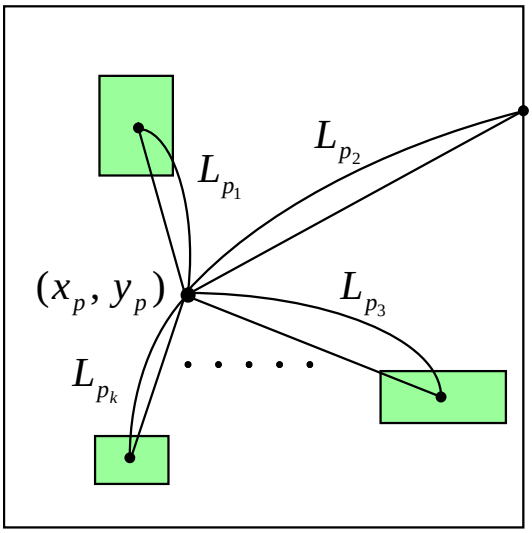

図 4.2: スター配線とネットのスタイナー点

式 (4.3) を全てのネットに対して適用し，足し合わせたものを総配線長 L_x とし，これを最小化する．

$$L_x = \sum_{all\_net} L_{p_x} \longrightarrow min \tag{4.4}$$

上式を行列の形で表すと，以下のようになる．

$$\frac{1}{2}\boldsymbol{x}^T \boldsymbol{Q}_x \boldsymbol{x} + \boldsymbol{d}_x \boldsymbol{x} + \boldsymbol{c}_x \longrightarrow min \tag{4.5}$$

ここで行列 $\boldsymbol{Q_x}$ は対称正定値行列で，接続関係を表すハイパーエッジを表現している．また，$\boldsymbol{c}_x$，$\boldsymbol{d}_x$ はそれぞれ，モジュールと外部端子の接続，モジュールとモジュールとの接続による項である．これを解いて次式を得る．

$$\boldsymbol{Q}_x \boldsymbol{x} + \boldsymbol{c}_x = 0 \tag{4.6}$$

これにより，各モジュールの x 座標が求まる．また，y 方向成分についても同様にして解くことができ，モジュールの配置位置が一意に定まる．

4.3.2 重なり除去操作

式 (4.6) によって求まる配置は一般に，配置エリア中央にモジュールが集中し，モジュール同士の重なりが多数存在する．本研究で定義した制約抽出では，重なりを許したモデル配置からでも制約は抽出可能である．しかし，重なりが多数ある配置は，実際の重なりの無い配置との差異が大きく，そこから抽出された制約が有効に機能しない恐れがある．

そこで，H. Eisenmann ら [7] によって提案された，モジュール密度を利用した重なり除去を導入する．この手法は，配置エリアの粗密の状態を示すモジュール密度を考え，式（4.6）にこれに応じた新たな力 $\vec{f}$ を繰り返し加えることで，モジュールを徐々に拡散させる．x 成分については，次のように示される．

$$\boldsymbol{Q}_x\boldsymbol{x}_l + \boldsymbol{c}_x + \sum_{m=1}^{l-1} \alpha_m \boldsymbol{f}_{x_m} + \alpha_l \boldsymbol{f}_{x_l} = 0 \tag{4.7}$$

ただし，l は繰り返し回数，α_l は配線長に関する重みである．この重みが適切に設定されていない場合，中央に集中していたモジュールが一斉にエリア境界へと広がり，移動した先で他のモジュールと重なり，再び中央へと戻ってきてしまうといったことが起こってしまうため，注意しなくてはならない．

力 $\boldsymbol{f}_l$ は，現時点でのモジュールの座標と，次式で表される領域内の各地点に生じる力 $\vec{f}(x,y)$ によって，繰り返しの都度計算される．

$$\vec{f}(x,y) = k \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} D(x',y') \frac{\vec{r}(x,y) - \vec{r}(x',y')}{|\vec{r}(x,y) - \vec{r}(x',y')|^2} dx'dy' \tag{4.8}$$

ここで，$f(x,y)$ は地点 (x,y) 上にあるモジュールに働く力，$\vec{r}(x,y)$ は地点 (x,y) のベクトル表現である．また，モジュール密度 $D(x,y)$ は，その地点で許容できるモジュールの包含容量に対するモジュールの占有率を示している．

実際にこの手法を導入する際には，領域内を適当な大きさの部分領域に分割して計算を行う．まず図 4.3 のように，$(\Delta x \times \Delta y)$ の大きさを持つ部分領域 bin へと分割し，次に，各 bin のモジュール密度 $D_b(i,j)$ を求める．

$$D_b(i,j) = -\frac{\sum_n w_n \cdot h_n}{TotalArea} + Overlap_b(i,j) \tag{4.9}$$

ここで，w_n，h_n，$TotalArea$ はそれぞれ，モジュールの幅，高さ，領域の総面積であり，$Overlap_b(i,j)$ は，(i,j) の bin におけるモジュールの占有率の合計である．

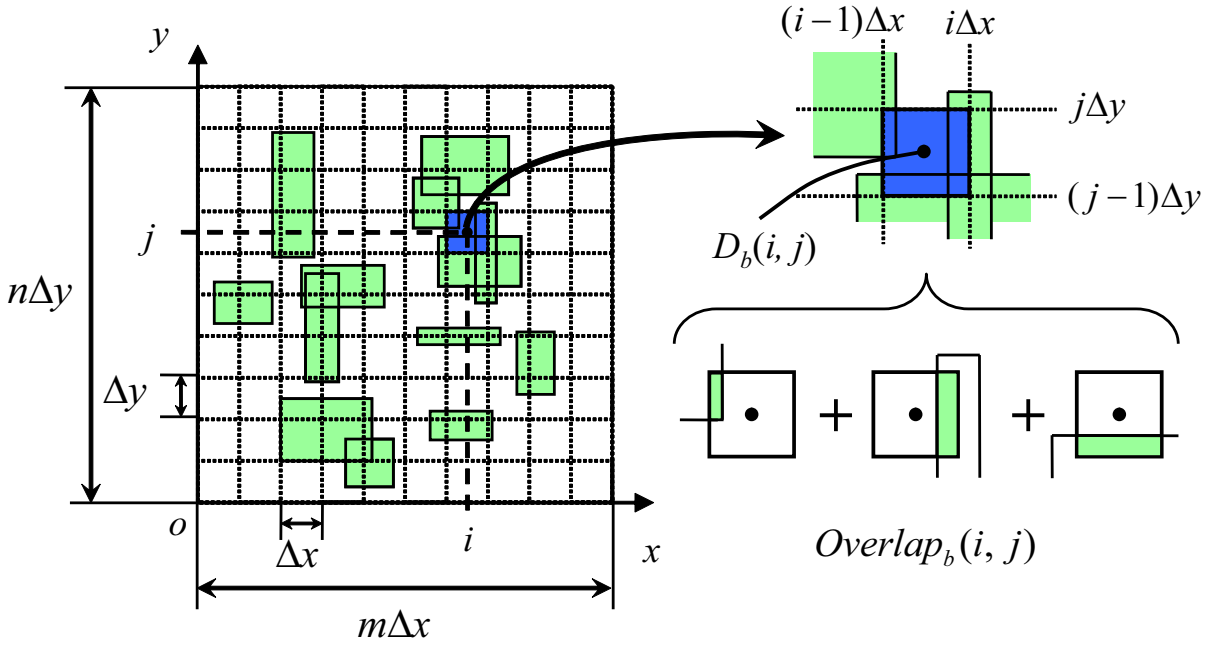

図 4.3: 領域の分割と密度関数 D

次に式 (4.8) を変形する.

$$\begin{aligned}
\vec{f}(i,j) &= k\sum_{i'=1}^{m}\int_{(i'-1)\Delta x}^{i'\Delta x}\left(\sum_{j'=1}^{n}\int_{(j'-1)\Delta y}^{j'\Delta y} D_b\left(i',\ j'\right)\frac{\begin{pmatrix} x \\ y \end{pmatrix}-\begin{pmatrix} x' \\ y' \end{pmatrix}}{\left|\begin{pmatrix} x \\ y \end{pmatrix}-\begin{pmatrix} x' \\ y' \end{pmatrix}\right|^2}dx'\right)dy' \\
&\approx k\sum_{i'=1}^{m}\sum_{j'=1}^{n}\int_{(i'-1)\Delta x}^{i'\Delta x}\int_{(j'-1)\Delta y}^{j'\Delta x}\left(D_b(i',j')\frac{\begin{pmatrix} x \\ y \end{pmatrix}-\begin{pmatrix} (i'-1/2)\,\Delta x \\ (j'-1/2)\,\Delta y \end{pmatrix}}{\left|\begin{pmatrix} x \\ y \end{pmatrix}-\begin{pmatrix} (i'-1/2)\,\Delta x \\ (j'-1/2)\,\Delta y \end{pmatrix}\right|^2}\right)dx'dy' \\
&= k\sum_{i'=1}^{m}\sum_{j'=1}^{n} D_b(i',j')\frac{\begin{pmatrix} x \\ y \end{pmatrix}-\begin{pmatrix} (i'-1/2)\,\Delta x \\ (j'-1/2)\,\Delta y \end{pmatrix}}{\left|\begin{pmatrix} x \\ y \end{pmatrix}-\begin{pmatrix} (i'-1/2)\,\Delta x \\ (j'-1/2)\,\Delta y \end{pmatrix}\right|^2}\int_{(i'-1)\Delta x}^{i'\Delta x}\int_{(j'-1)\Delta x}^{j'\Delta x}dx'dy' \\
\vec{f}(i,j) &= k\cdot\Delta x\cdot\Delta y\cdot D_b(i',j')\frac{\begin{pmatrix} x \\ y \end{pmatrix}-\begin{pmatrix} (i'-1/2)\,\Delta x \\ (j'-1/2)\,\Delta y \end{pmatrix}}{\left|\begin{pmatrix} x \\ y \end{pmatrix}-\begin{pmatrix} (i'-1/2)\,\Delta x \\ (j'-1/2)\,\Delta y \end{pmatrix}\right|^2} \qquad (4.10)
\end{aligned}$$

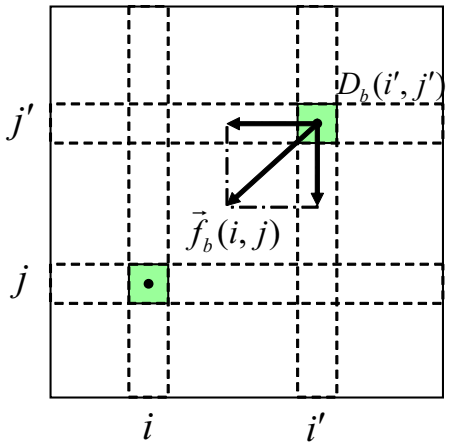

図 4.4: セルに対する力 $\vec{f_c}$

$\tilde{k} = k \cdot \Delta x \cdot \Delta y$ と置いて,

$$\therefore \vec{f}(i,j) = \tilde{k} \cdot D_b(i',j') \frac{\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} (i'-1/2)\,\Delta x \\ (j'-1/2)\,\Delta y \end{pmatrix}}{\left| \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} (i'-1/2)\,\Delta x \\ (j'-1/2)\,\Delta y \end{pmatrix} \right|^2} \tag{4.11}$$

$x = \left(i - \frac{1}{2}\right)\Delta x$, $y = \left(j - \frac{1}{2}\right)\Delta y$ として,

$$\vec{f}\left(\left(i-\frac{1}{2}\right)\Delta x, \left(j-\frac{1}{2}\right)\Delta y\right) = \tilde{k} \sum_{i'=1}^{m} \sum_{j'=1}^{n} D_b(i',j') \frac{\begin{pmatrix} (i-\frac{1}{2}\Delta x) \\ (j-\frac{1}{2}\Delta y) \end{pmatrix} - \begin{pmatrix} (i'-\frac{1}{2}\Delta x) \\ (j'-\frac{1}{2}\Delta y) \end{pmatrix}}{\left| \begin{pmatrix} (i-\frac{1}{2}\Delta x) \\ (j-\frac{1}{2}\Delta y) \end{pmatrix} - \begin{pmatrix} (i'-\frac{1}{2}\Delta x) \\ (j'-\frac{1}{2}\Delta y) \end{pmatrix} \right|^2} \tag{4.12}$$

これを解いて，bin にかかる力 $\vec{f_b}$ を得る（図 4.4）.

$$\therefore \vec{f_b}(i,j) = \tilde{k} \sum_{i'=1}^{m} \sum_{j'=1}^{n} D_b(i',j') \frac{\begin{pmatrix} (i-i')\Delta x \\ (j-j')\Delta y \end{pmatrix}}{\left((i-i')\Delta x\right)^2 + \left((j-j')\Delta y\right)^2} \tag{4.13}$$

上式 (4.13) から分かるように，$D(i,j) < 0$ ならば (i',j') の bin に引き寄せられる方向, $D(i,j) > 0$ ならば (i',j') の bin から遠ざかる方向に力が働く.

次に $\vec{f_b}$ をモジュールに対する力 $\vec{f_m}$ に変換する（図 4.5 参照）.

$$\therefore \vec{f_m}(a) = \sum_{i=1}^{i=m} \sum_{j=1}^{j=n} \vec{f_c}(i,j) \cdot \frac{Overlap_m(a,i,j)}{\Delta x \cdot \Delta y} \tag{4.14}$$

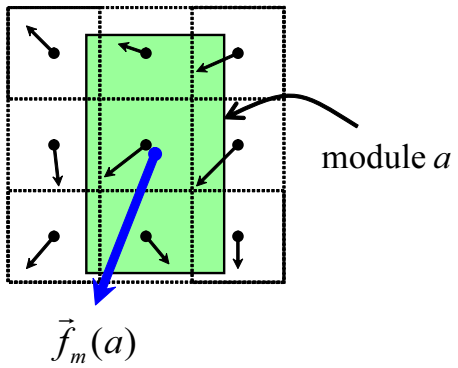

図 4.5: モジュールに対する力 $\vec{f_m}$

ただし，$Overlap_m(a,i,j)$ はモジュール a が $bin(i,j)$ を占める面積を表すとする．これにより，行列 $\boldsymbol{f}_x$，$\boldsymbol{f}_y$ が求まった．

4.3.3 力学的手法の流れ

力学的手法の流れを以下にまとめる．

1. 二乗配線長を最小化した配置を導出する（式 (4.5)）
2. モジュール密度 D を用い，各モジュールに働く力 $\vec{f}_m$ を求める（式 (4.13)）
3. 二乗配線長最小化の式に $\vec{f}_m$ を加えた上で，各モジュールの力学的な安定点を求める（式 (4.14)）
4. 2.〜3. の手順を繰り返し行い，全てのモジュールに対する力がゼロになった時点で終了する

4.4 制約の抽出

力学的手法で得られたモデル配置から，半順序制約を抽出する．モジュールの重心座標からモジュール同士の相対位置関係を判断し，3.2 に示した方法でシーケンスペア上のモジュールの出現順序を限定する．

抽出された制約は，制約リスト L_{gp}, L_{gm} によって記憶する．このリストは符号付のモジュール名の部分集合からなる．順列中で自身よりも他のモジュールが後に出現する場合そのモジュール名を，前に出現する場合マイナスの符号をつけたモジュール名をリストに追加する．例えば $\Gamma_+^{-1}(a) < \Gamma_+^{-1}(b)$ の制約を持つ場合，$L_{gp}(a)$ に $-b$ を，$L_{gp}(b)$ に a を追加する．

POSP の半順序制約はモジュール同士の相対位置関係のみから求まるが，距離が近いモジュール同士の場合，座標が少し異なるだけでその相対位置関係は入れ替わってしまう．

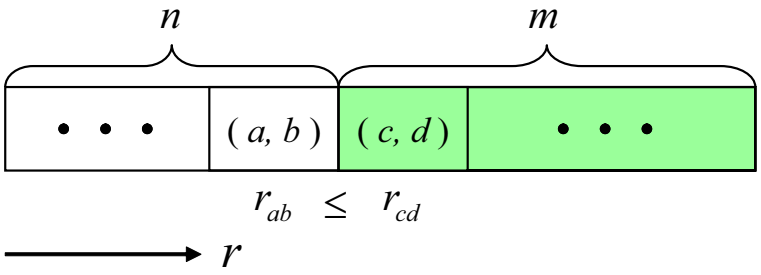

図 4.6: 制約付与率 ρ，制約除外半径 r_e

このため，そのようなモジュール同士の制約は，配線長最小化にとって重要度が低いと考えられる．本研究では，図 4.6 のように，全てのモジュールの組み合わせをモジュール間のユークリッド距離でソートし，距離の遠いモジュール同士から一定の割合だけ半順序制約を付与することとした．これを制約付与率 ρ で表記する．また，このとき制約を付与されなかったモジュール対の最大距離を制約除外半径とし，r_e で表す．図 4.6 中の例では，$(m+n)$ 個のモジュール対中 m 個に制約をつけ $\rho = \frac{m}{m+n}$，制約の付かない最後のモジュール対は (a, b) であり $r_e = r_{ab}$（ab 間のユークリッド距離）となる．

4.5 SA による解探索

4.5.1 初期解

力学的手法によって得られたモデル配置を用い，初期解の生成を行う．初期解となるシーケンスペアは，2.2.2 で述べた方法を基に行う．ただし，モデル配置はモジュール同士の重なりを許しているため，モジュールを質点として扱い，ステップラインを直線に簡略化することで，コード生成を簡略化する．

図 4.7 のように，各モジュールの重心座標を通過する傾き $1, -1$ の直線 $y = x + p_i$，$y = -x + q_i$（ただし i はモジュール名）をそれぞれ求める．p_i を降順，q_i を昇順にソートし，この順でモジュール名をそれぞれ並べたものをシーケンスペアの初期解 Γ_+, Γ_- とする．図 4.7 の例では，初期解 $(\Gamma_+, \Gamma_-) = (abcd, cadb)$ を得る．

このコードはモデル配置の座標を基に作成されるため，その相対位置関係を保持し，表 3.1 に示した半順序制約を全て満足する．

4.5.2 評価関数

モジュール配置問題では，最適化の指標として面積と配線長が用いられることが多く，本研究でもそれに倣う．一般に面積最小化と配線長最小化はトレードオフの関係にあることが多く，最適化の際は両方を考慮しなくてはならない．

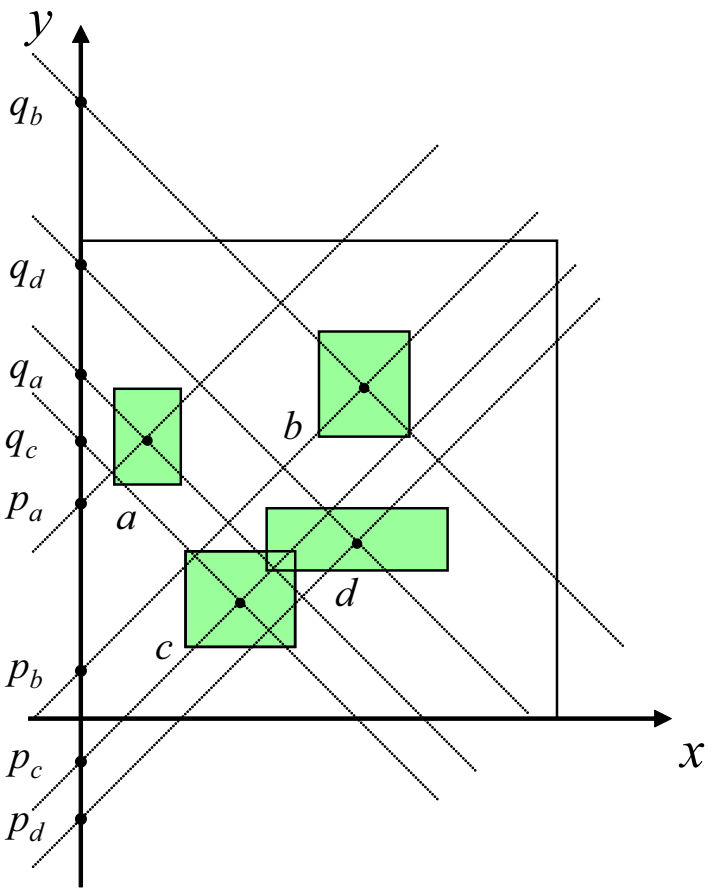

図 4.7: 初期解の生成

本研究では，SA での探索時に用いる評価関数として，次式 (4.15) のように，面積コスト C_{area}，配線コスト C_{wire} のそれぞれに重み α, β をつけたコスト関数 C を定義する．

$$C = \alpha \cdot C_{area} + \beta \cdot C_{wire} \tag{4.15}$$

ただし，$\alpha + \beta = 1$ とする．面積コストの見積もり C_{area} は，全矩形を囲む最小の矩形の面積とする．また配線コストの見積もり C_{wire} に関しては，バウンディングボックスを用いたもの，二次配線長を用いたものなどが考えられるが，ここではモデル配置との整合性を考慮して二乗配線長見積もり式 (4.3) を採用する．

4.5.3 隣接解

SA での探索は POSP の解空間内，すなわち付与された半順序制約を満たした解空間内でのみ行われる．一方隣接解の生成は，制約を付与しない従来手法で一般的に用いられる挿入，交換，全交換，回転を用いて行う．このうち，シーケンスペアの順序を変更する挿入，交換，全交換については，POSP 以外の解を探索しないように，隣接解の制約チェックを行う必要がある．

以降に示す手続きでは，すでに制約チェックリスト L_{gp}, L_{gm} は得られているものとする．挿入操作の制約チェックでは，あらかじめ選ばれたモジュール m_i の順列中の挿入可能な範囲の上下限 $p, q (1 \leq p < q \leq n)$ を求める．以下に Γ_+ 上での挿入操作時の制約チェック手続きを示す．

- **ステップ 1**：Γ_+^{-1} を作成する．
- **ステップ 2**：$p = q = \Gamma_+^{-1}(m_i)$ とする．
- **ステップ 3**：制約リスト $L_{gp}(m_i)$ 内の全ての要素 x について，
  - $x < 0$ の場合
    $\Gamma_+^{-1}(|x|) < p$ ならば，$p = \Gamma_+^{-1}(|x|) + 1$
  - $x > 0$ の場合
    $\Gamma_+^{-1}(|x|) > q$ ならば，$q = \Gamma_+^{-1}(|x|) - 1$
- **ステップ 4**：p, q を出力して終了する．

この手続きには，モジュール数を n として，ステップ 1，ステップ 3 にそれぞれ $O(n)$ の計算が発生する．よってトータルの計算量は $O(n)$ となる．Γ_- の挿入操作についても全く同様の方法で制約チェックが行える．

次に Γ_+ 上での交換操作時の制約チェック手続きを示す．この手続きでは，モジュール対 $m_i, m_j (\Gamma_+^{-1}(m_i) < \Gamma_+^{-1}(m_j))$ の交換操作が可能かどうかを判定する．可能ならばそのモジュール対の位置 $p, q (1 <\leq p < q \leq n)$ を出力し，手続きを終了する．不可能な場合はモ

ジュール対を新たに選び直し，再度判定を行う．また，判定には一定の上限$loop$を儲ける．この回数以内に交換操作を行えるモジュール対が見つからなかった場合，$p = q = 0$を出力し手続きを終了し，別の隣接解生成操作を試みる．

- **ステップ１**：Γ_+^{-1}を作成する．
- **ステップ２**：$l = 1$とする．
- **ステップ３**：$l \geq loop$ならば$p = q = 0$を出力し，終了する．そうでなければモジュール対m_i, m_jを選び直す（ただし$\Gamma_+^{-1}(m_i) < \Gamma_+^{-1}(m_j)$）．
- **ステップ４**：制約リスト$L_{gp}(m_i)$内の全ての要素$x(> 0)$について，
  - $\Gamma_+^{-1}(|x|) < \Gamma_+^{-1}(m_j)$の場合，
    $l = l + 1$とし，ステップ２に戻る
- **ステップ５**：制約リスト$L_{gp}(m_j)$内の全ての要素$y(< 0)$について，
  - $\Gamma_+^{-1}(|y|) > \Gamma_+^{-1}(m_i)$の場合，
    $l = l + 1$とし，ステップ２に戻る．
- **ステップ６**：$p = \Gamma_+^{-1}(m_i), q = \Gamma_+^{-1}(m_j))$を出力し，終了する．

この操作は，ステップ１，ステップ４，ステップ５にそれぞれ$O(n)$の計算量が必要となる．またステップ４，ステップ５は最悪$loop$回繰り返される．以上より，トータルの計算量は$O(n \times loop)$となる．このことから，$loop$を少なくすることで，制約チェックによるアルゴリズムの時間的オーバーヘッドを抑制できることがわかる．Γ_-の交換操作についても全く同様の方法で制約チェックが行える．また全交換については，ステップ４，ステップ５の操作をΓ_+, Γ_-両方で行い，計算量は同じく$O(n \times loop)$である．本研究では$loop = 10$に設定し，次章の実験を行った．

第5章　実験と考察

本章では，提案アルゴリズムをC言語により実装し行った実験の結果を示し，考察を行う．

5.1　実験対象とパラメータ設定

実験の対象は，GSRCベンチマークのそれぞれ規模の異なる三つの回路である（表5.1参照）．SAのパラメータ開始温度t_{start}，終了温度t_{end}，温度降下係数d_c，コスト関数の面積重みα，配線重みβは表5.2のように設定した．またSAの内部ループ数は$lp = 100, 200, 1000, 2000, 10000, 20000, 100000$の七つを設定し，各設定ごとに実験を行った．

以降特に記述の無い限り，各回路に対し，表5.2のパラメータで実験を行ったものとする．また，全ての実験結果は，同一条件の下各五回行った結果の平均を取ったものである．

5.2　重なり除去操作の選択

本研究では，力学的手法によるモデル配置の導出の際，半順序制約の質の向上を狙って重なり除去操作の採用を提案した．モデル配置からは，モジュール同士の相対位置関係さえ分かれば制約を抽出できるため，重なり除去操作の有効性について議論する必要がある．ここでは，4.3.2で述べた重なり除去操作について，実験を行い，その効果を検証した．

本研究では，特に重なり除去を行わず配線長のみを考慮した配置，モジュール密度関数Dにより各モジュールにかかる力$\vec{f_m}$の全てが一定値以下になるまで繰り返すした配置，

表5.1: GSRCベンチマーク

Circuit	#module	#net	#I/O
n100a	100	885	334
n200a	200	1585	564
n300a	300	1893	569

表 5.2: SA とコスト関数 C ののパラメータ設定

Circuit	t_{start}	t_{end}	d_c	$\alpha : \beta$
n100a	5.0×10^5	0.1	0.98	100 : 1
n200a	1.0×10^6	1.0	0.98	300 : 1
n300a	5.0×10^6	1.0	0.98	300 : 1

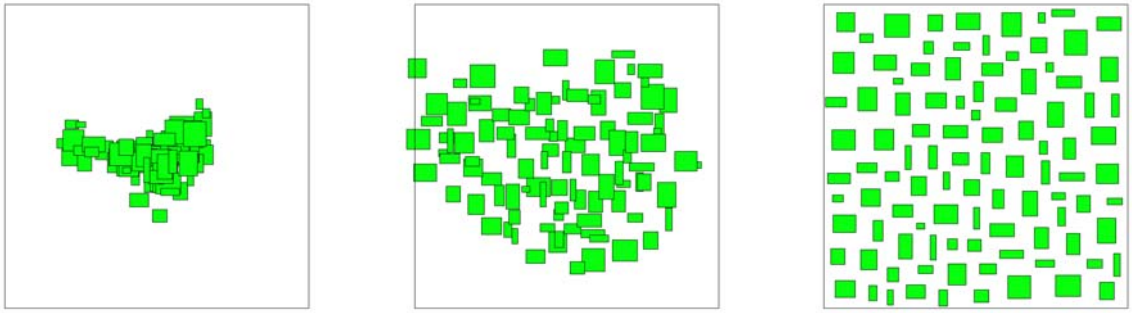

図 5.1: 力学的手法による n100a の配置結果（左：FD_0，中央：FD_+，右：FD_∞）

完全な力学的安定点に移動するまで重なり除去を行って得られる配置の三つをモデル配置として使用する．以降この三つを FD_0，FD_+，FD_∞ と表記する．

また，POSP については二つのパラメータを付加し，$POSP_i^j$ と表記する．i では上述の三つのモデル配置を表し，j では後述の制約付与率 ρ の値を示すものとする．

まず，FD_0，FD_+，FD_∞ によって得られた各回路のモデル配置を図 5.1，図 5.2，図 5.3 に示す．FD_0 ではほとんどのモジュールが配置エリアに集中しているのに対し，FD_+ では各モジュールが分散し，重なり度合いが緩和されている．$FD\infty$ の配置結果は，いずれの回路も完全に重なりが除去されており，モジュール同士が比較的等間隔に並んでいるのがわかる．

5.2.1 重なり除去操作の有効性

重なり除去操作自体の有効性を検証するため，FD_0 と FD_+ からの配置結果を比較する．各回路で FD_0，FD_+ それぞれのモデル配置から半順序制約を抽出し，配置最適化を行った結果を図 5.4〜図 5.6，表 5.3〜表 5.5 に示す．

回路ごとの比較を行うと，いずれの回路も大部分で $POSP_0$ よりも $POSP_+$ のほうが良い解を出力している．特に n300a の回路の配線長評価においては，最大で 7.32%，平均で 4.51% もの削減率を達成している．

次に，面積と配線長について比較すると，大部分で $POSP_0$ よりも $POSP_+$ が良い解を出

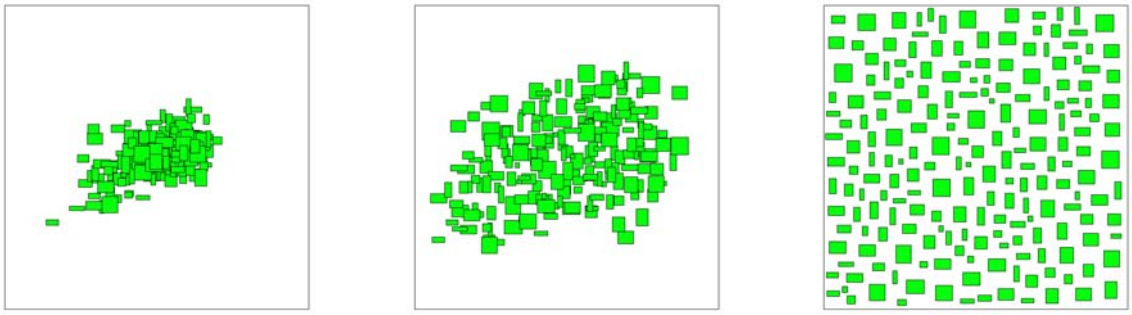

図 5.2: 力学的手法による n200a の配置結果（左：FD_0，中央：FD_+，右：FD_∞）

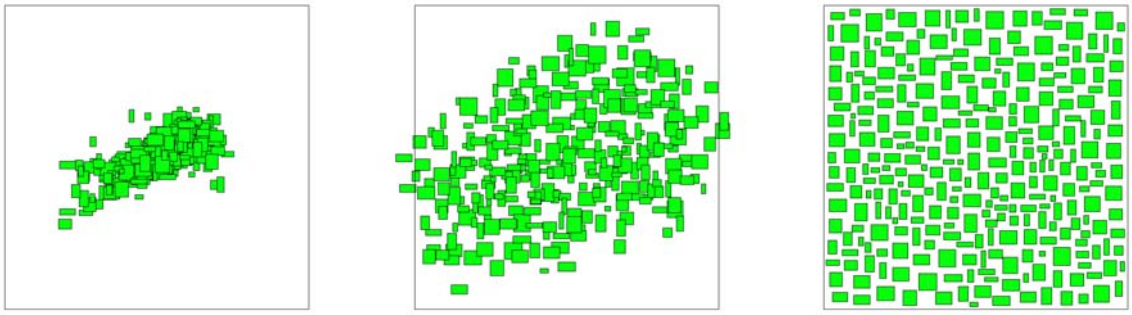

図 5.3: 力学的手法による n300a の配置結果（左：FD_0，中央：FD_+，右：FD_∞）

表 5.3: 重なり除去操作による面積コスト，配線コストの変化（n100a）

ρ	lp	C_{area}			C_{wire}		
		POSP_0	POSP_+		POSP_0	POSP_+	
0.4	100	200,608	197,109	(-1.74%)	27,610,624	27,256,850	(-1.28%)
	200	197,288	194,427	(-1.45%)	26,940,932	26,610,860	(-1.23%)
	1,000	194,897	192,933	(-1.01%)	26,233,582	25,944,541	(-1.10%)
	2,000	191,517	190,845	(-0.35%)	25,677,247	25,516,876	(-0.62%)
	10,000	189,984	189,997	(+0.01%)	24,970,947	25,025,678	(+0.22%)
	20,000	189,288	189,782	(+0.26%)	24,790,142	24,992,180	(+0.81%)
	100,000	187,881	187,868	(-0.01%)	24,515,357	24,476,857	(-0.16%)
0.6	100	203,189	199,378	(-1.88%)	28,013,640	27,404,446	(-2.17%)
	200	203,025	198,301	(-2.33%)	27,707,875	27,108,145	(-2.16%)
	1,000	193,371	195,107	(+0.90%)	26,867,525	26,311,858	(-2.07%)
	2,000	192,739	192,015	(-0.38%)	26,635,272	26,011,383	(-2.34%)
	10,000	190,235	190,813	(+0.30%)	26,160,297	25,456,952	(-2.69%)
	20,000	190,950	191,028	(+0.04%)	25,934,367	25,378,532	(-2.14%)
	100,000	188,544	188,382	(-0.09%)	25,562,621	24,897,950	(-2.60%)
0.8	100	206,761	200,424	(-3.06%)	28,502,405	27,763,502	(-2.59%)
	200	201,284	198,238	(-1.51%)	28,062,778	27,277,417	(-2.80%)
	1,000	197,999	194,917	(-1.56%)	27,642,304	26,729,532	(-3.30%)
	2,000	196,168	195,423	(-0.38%)	27,322,564	26,754,182	(-2.08%)
	10,000	192,276	191,155	(-0.58%)	27,112,531	26,157,689	(-3.52%)
	20,000	192,586	190,650	(-1.01%)	26,757,469	25,974,006	(-2.93%)
	100,000	190,558	189,628	(-0.49%)	26,190,334	25,753,273	(-1.67%)

力しているのは両者同じであるが，配線コストのほうが面積コストよりも削減率が若干高い．n100a の平均削減率は，面積 0.78%，配線長 1.83%，n200a については，面積 0.60%，配線長 1.35%，n300a については，面積 1.43%，配線長 4.51% となっている．これは，重なりのない実配置に近づいたことで，面積に対してより有効なモデル配置になったのではないかと予想される．

以上の結果から，重なり除去操作は，モデル配置の有効性を高めると判断した．

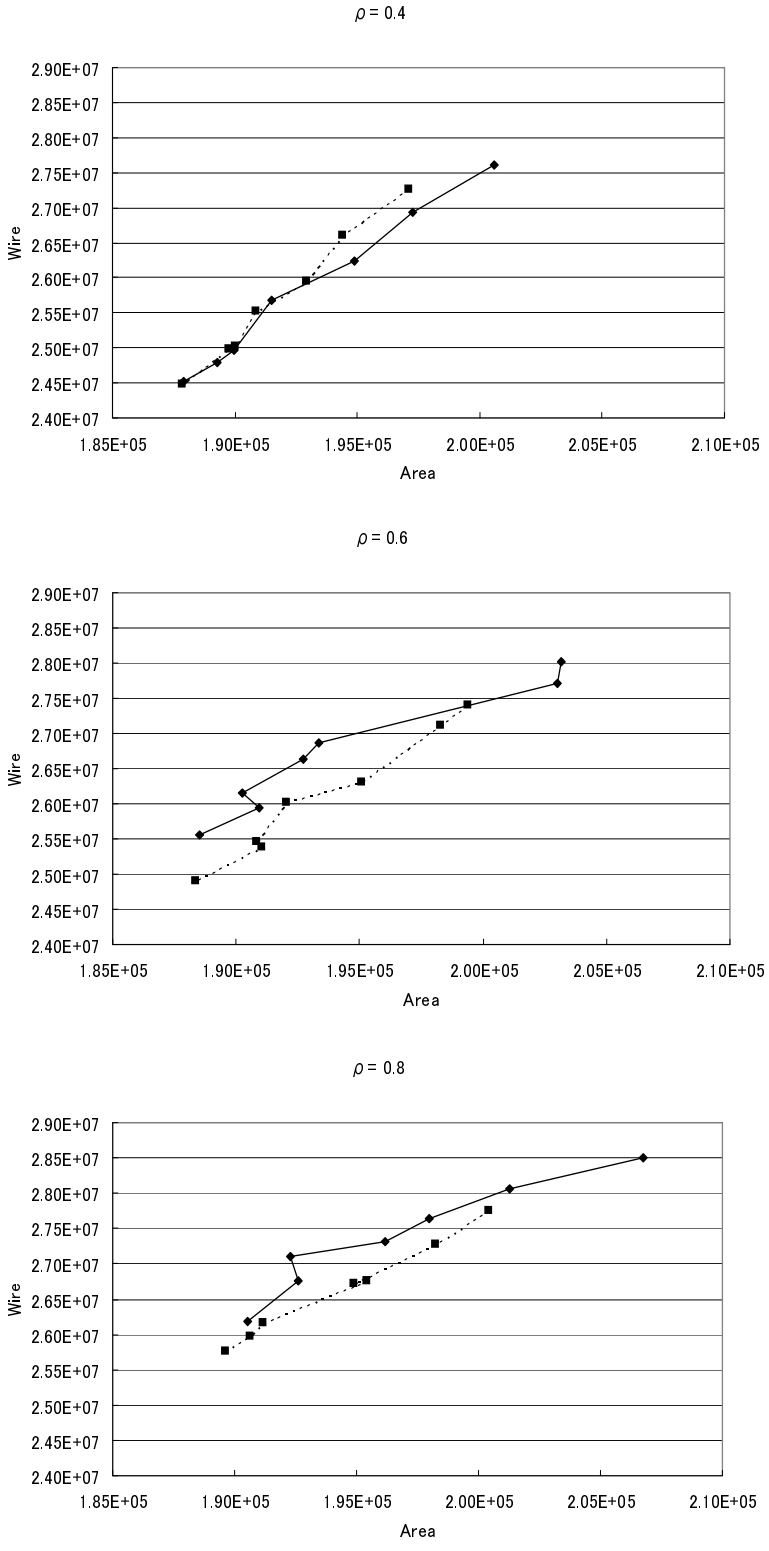

図 5.4: $POSP_0$，$POSP_+$ の比較（n100a，実線：$POSP_0$，破線：$POSP_+$）

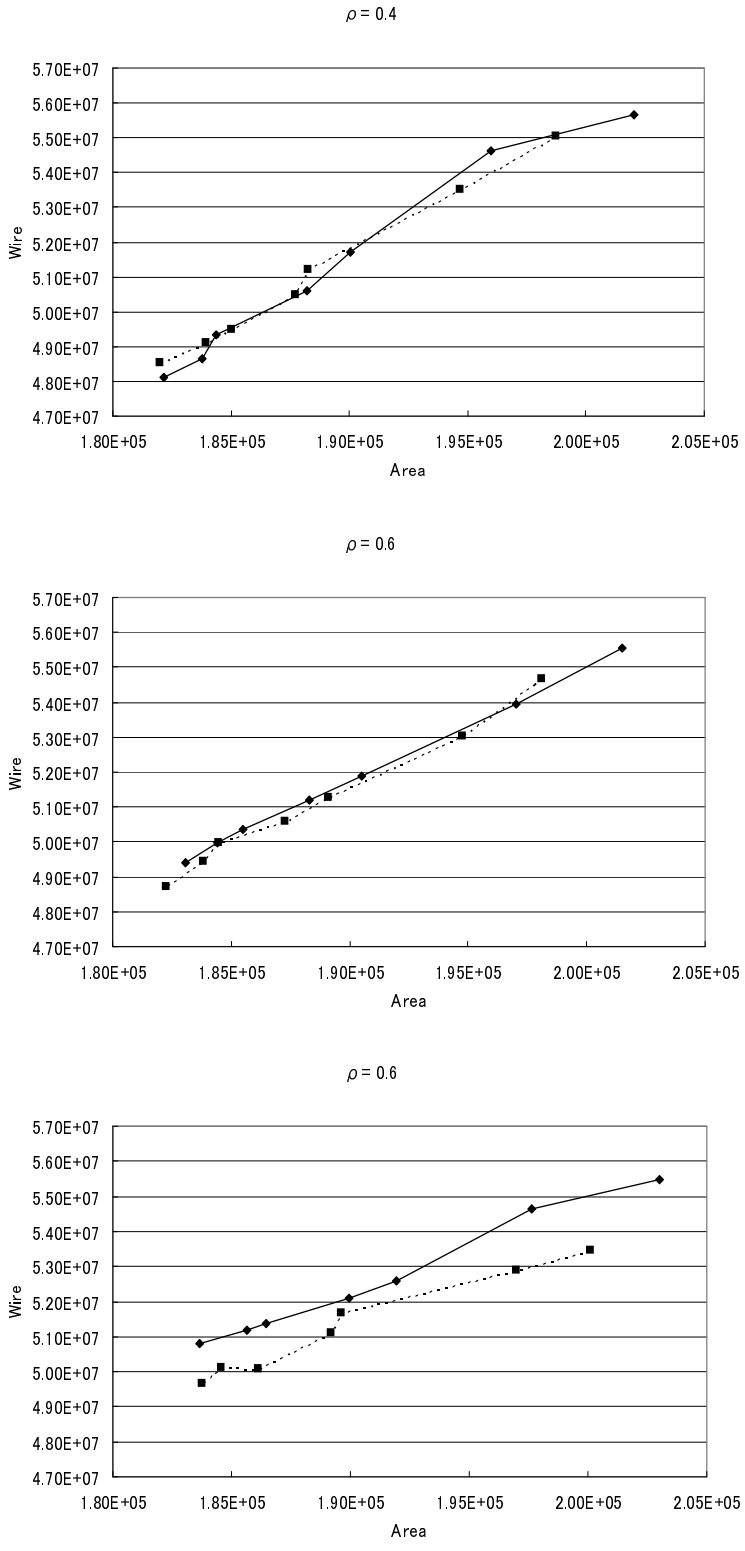

図 5.5: $POSP_0$，$POSP_+$ の比較（n200a，実線：$POSP_0$，破線：$POSP_+$）

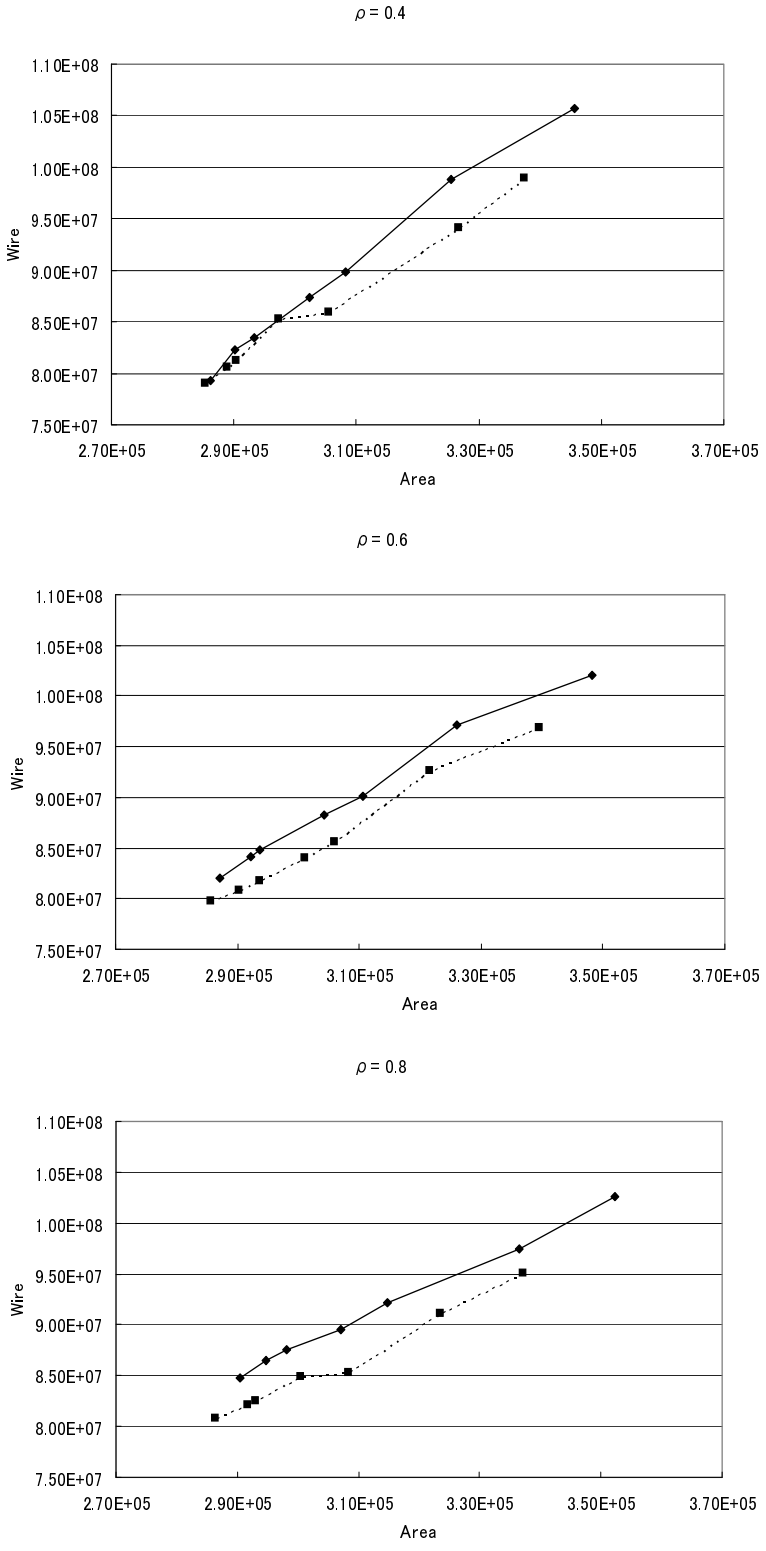

図 5.6: POSP_0，POSP_+ の比較（n300a，実線：POSP_0，破線：POSP_+）

表 5.4: 重なり除去操作による面積コスト，配線コストの変化（n200a）

ρ	lp	C_{area}			C_{wire}		
		POSP_0	POSP_+		POSP_0	POSP_+	
0.4	100	202,029	198,720	(-1.64%)	55,672,432	55,046,530	(-1.12%)
	200	195,977	194,703	(-0.65%)	54,614,560	53,501,204	(-2.04%)
	1,000	190,036	188,237	(-0.95%)	51,704,397	51,229,120	(-0.92%)
	2,000	188,185	187,720	(-0.25%)	50,593,971	50,505,619	(-0.17%)
	10,000	184,376	185,045	(+0.36%)	49,349,098	49,506,270	(+0.32%)
	20,000	183,770	183,964	(+0.11%)	48,665,050	49,095,046	(+0.88%)
	100,000	182,151	181,994	(-0.09%)	48,112,689	48,528,547	(+0.86%)
0.6	100	201,506	198,105	(-1.69%)	55,557,733	54,686,503	(-1.57%)
	200	197,040	194,780	(-1.15%)	53,943,928	53,021,595	(-1.71%)
	1,000	190,531	189,118	(-0.74%)	51,882,788	51,266,151	(-1.19%)
	2,000	188,284	187,298	(-0.52%)	51,211,222	50,602,263	(-1.19%)
	10,000	185,496	184,478	(-0.55%)	50,363,803	49,963,254	(-0.80%)
	20,000	184,441	183,847	(-0.32%)	49,959,752	49,446,384	(-1.03%)
	100,000	183,046	182,267	(-0.43%)	49,409,842	48,717,899	(-1.40%)
0.8	100	202,997	200,098	(-1.43%)	55,466,710	53,447,658	(-3.64%)
	200	197,620	196,972	(-0.33%)	54,646,602	52,909,905	(-3.18%)
	1,000	191,925	189,614	(-1.20%)	52,598,309	51,687,452	(-1.73%)
	2,000	189,951	189,212	(-0.39%)	52,111,636	51,095,273	(-1.95%)
	10,000	186,455	186,131	(-0.17%)	51,363,577	50,078,090	(-2.50%)
	20,000	185,634	184,585	(-0.57%)	51,181,835	50,110,694	(-2.09%)
	100,000	183,636	183,780	(+0.08%)	50,805,209	49,678,413	(-2.22%)

表 5.5: 重なり除去操作による面積コスト，配線コストの変化（n300a）

ρ	lp	C_{area}			C_{wire}		
		POSP_0	POSP_+		POSP_0	POSP_+	
0.4	100	345,628	337,365	(-2.39%)	105,739,562	98,885,615	(-6.48%)
	200	325,366	326,700	(+0.41%)	98,875,266	94,142,909	(-4.79%)
	1,000	308,304	305,549	(-0.89%)	89,856,168	85,982,089	(-4.31%)
	2,000	302,315	297,273	(-1.67%)	87,393,684	85,331,955	(-2.36%)
	10,000	293,236	290,443	(-0.95%)	83,424,149	81,239,446	(-2.62%)
	20,000	290,132	288,884	(-0.43%)	82,264,088	80,551,682	(-2.08%)
	100,000	286,091	285,331	(-0.27%)	79,312,310	79,083,411	(-0.29%)
0.6	100	348,212	339,549	(-2.49%)	102,054,400	96,913,052	(-5.04%)
	200	326,008	321,676	(-1.33%)	97,181,324	92,655,049	(-4.66%)
	1,000	310,685	305,864	(-1.55%)	90,155,875	85,596,723	(-5.06%)
	2,000	304,150	301,054	(-1.02%)	88,213,477	84,036,531	(-4.74%)
	10,000	293,626	293,696	(+0.02%)	84,866,155	81,742,578	(-3.68%)
	20,000	292,234	290,306	(-0.66%)	84,206,277	80,839,353	(-4.00%)
	100,000	287,075	285,748	(-0.46%)	82,046,010	79,817,970	(-2.72%)
0.8	100	352,197	337,227	(-4.25%)	102,574,650	95,068,854	(-7.32%)
	200	336,558	323,505	(-3.88%)	97,401,221	91,132,948	(-6.44%)
	1,000	314,728	308,447	(-2.00%)	92,171,672	85,272,111	(-7.49%)
	2,000	307,087	300,557	(-2.13%)	89,517,988	84,896,497	(-5.16%)
	10,000	298,048	293,026	(-1.69%)	87,517,397	82,506,695	(-5.73%)
	20,000	294,778	291,721	(-1.04%)	86,535,014	82,127,813	(-5.09%)
	100,000	290,473	286,497	(-1.37%)	84,790,641	80,852,928	(-4.64%)

表 5.6: 制約除外半径 r_e

Circuit	ρ	r_e	
		POSP_+	POSP_∞
n100a	0.4	322.51	447.17
	0.6	245.61	338.28
	0.8	162.58	225.98
n200a	0.4	261.46	449.22
	0.6	194.48	338.56
	0.8	128.28	223.21
n300a	0.4	338.21	442.72
	0.6	251.91	333.64
	0.8	165.57	219.56

5.2.2 POSP_+ と POSP_∞ の比較

POSP_+ と POSP_∞ の比較を行う．各回路での制約除外半径 r_e を表 5.6 に示す．各 ρ における r_e を見ると，POSP_∞ が POSP_+ よりも重なりを取った分だけモジュール間の距離が遠くなり，r_e が大きくなっていることがわかる．なお，全ての回路において，力学的手法実行時の配置エリアの大きさは 800×800 となっている．

更に，n300a の回路において POSP_+，POSP_∞ の比較を行う．ρ を変化させたときの面積コスト，配線コストの比較結果を表5.7 に，トータルコストの比較結果を表5.8 に示す．

両者の面積コストと配線コストを比較すると，POSP_+ は面積コストで POSP_∞ に平均 0.36% 劣っているが，配線コストで 0.88% 勝っている．またトータルコストに関しては，POSP_+ が平均 0.24%POSP_∞ に勝っている．

POSP_∞ は，重なり除去の繰り返し回数を増やすことでモジュール同士を分散させ，重なりのない実配置により近づけることができる．しかし，モジュールを大きく動かすことによって，それまでに保たれていた配線長にとって理想的な相対位置関係を崩してしまったのではないかと考えられる．

表 5.7: POSP_{+} と POSP_{∞} の比較（n300a，面積コスト，配線コスト）

ρ	lp	C_{area}			C_{wire}		
		FD+	POSP_{∞}		FD+	POSP_{∞}	
0.4	100	337,365	335,435	(-0.57%)	98,885,615	100,945,047	(+2.08%)
	200	326,700	323,758	(-0.90%)	94,142,909	94,931,870	(+0.84%)
	1,000	305,549	301,822	(-1.22%)	85,982,089	88,056,337	(+2.41%)
	2,000	297,273	298,354	(+0.36%)	85,331,955	85,342,751	(+0.01%)
	10,000	290,443	291,707	(+0.44%)	81,239,446	82,168,353	(+1.14%)
	20,000	288,884	290,359	(+0.51%)	80,551,682	80,799,471	(+0.31%)
	100,000	285,331	286,552	(+0.43%)	79,083,411	79,338,965	(+0.32%)
0.6	100	339,549	334,330	(-1.54%)	96,913,052	97,016,514	(+0.11%)
	200	321,676	319,186	(-0.77%)	92,655,049	93,923,150	(+1.37%)
	1,000	305,864	307,273	(+0.46%)	85,596,723	86,575,775	(+1.14%)
	2,000	301,054	299,110	(-0.65%)	84,036,531	85,186,306	(+1.37%)
	10,000	293,696	291,642	(-0.70%)	81,742,578	82,564,693	(+1.01%)
	20,000	290,306	290,758	(+0.16%)	80,839,353	81,515,466	(+0.84%)
	100,000	285,748	285,689	(-0.02%)	79,817,970	80,536,755	(+0.90%)
0.8	100	337,227	335,044	(-0.65%)	95,068,854	95,155,743	(+0.09%)
	200	323,505	320,000	(-1.08%)	91,132,948	91,336,999	(+0.22%)
	1,000	308,447	305,330	(-1.01%)	85,272,111	86,148,194	(+1.03%)
	2,000	300,557	299,257	(-0.43%)	84,896,497	85,616,987	(+0.85%)
	10,000	293,026	292,473	(-0.19%)	82,506,695	83,239,870	(+0.89%)
	20,000	291,721	290,162	(-0.53%)	82,127,813	82,895,036	(+0.93%)
	100,000	286,497	287,338	(+0.29%)	80,852,928	81,317,881	(+0.58%)

表 5.8: POSP_+ と POSP_∞ の比較（n300a，トータルコスト）

ρ	lp	POSP_+	POSP_∞	
0.4	100	664,768	669,686	(+0.74%)
	200	638,382	638,070	(-0.05%)
	1,000	590,189	593,366	(+0.54%)
	2,000	579,780	580,894	(+0.19%)
	10,000	559,377	563,722	(+0.78%)
	20,000	555,538	557,831	(+0.41%)
	100,000	547,118	549,184	(+0.38%)
0.6	100	660,391	655,534	(-0.74%)
	200	628,432	630,163	(+0.28%)
	1,000	589,222	593,880	(+0.79%)
	2,000	579,245	581,127	(+0.32%)
	10,000	564,291	564,975	(+0.12%)
	20,000	557,910	560,608	(+0.48%)
	100,000	549,974	552,304	(+0.42%)
0.8	100	651,950	650,063	(-0.29%)
	200	625,198	622,382	(-0.45%)
	1,000	590,719	590,522	(-0.03%)
	2,000	581,607	582,705	(+0.19%)
	10,000	566,161	568,046	(+0.33%)
	20,000	563,602	564,597	(+0.18%)
	100,000	554,160	556,543	(+0.43%)

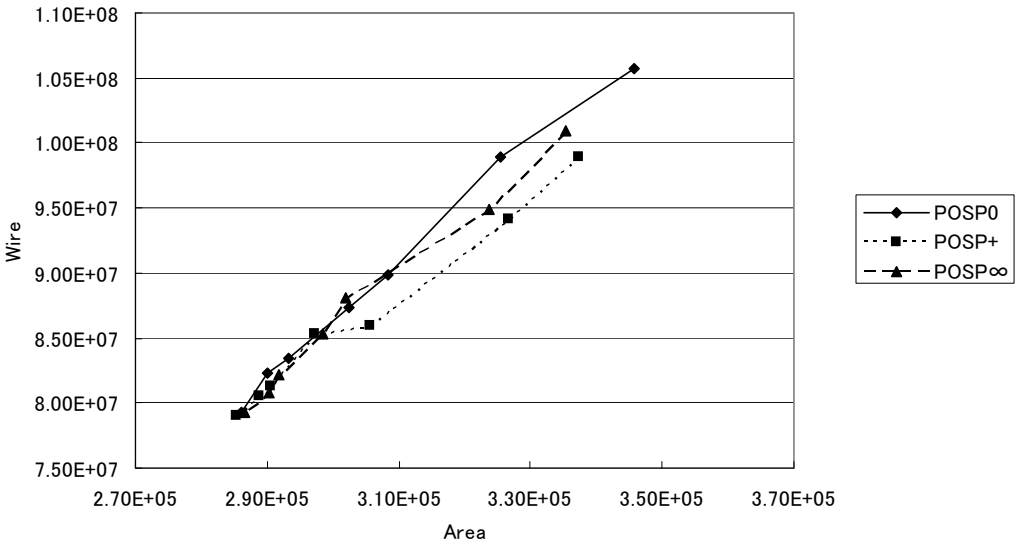

図 5.7: POSP_0，POSP_+，POSP_∞ の比較（n300a，$\rho = 0.4$）

5.2.3 制約付与率の影響度

制約付与率と重なり除去との関係を調査する．n300a 回路において，POSP_0，POSP_+，POSP_∞ それぞれを重なり除去に用いた配置結果を図 5.7，図 5.8，図 5.9 に示す．横軸は面積コスト，縦軸は配線コストである．また，各点をつないだ折れ線は，内部ループ数を $lp = 100$ から $lp = 100000$ まで増やしたときの解の変化の様子を表している．ループ数を増やすにつれて SA の総探索数が増加し，図右上から図左下へと解が改善していく．

まず制約付与率 $\rho = 0.4$ の結果についてみると，ループ数が少ない場面は POSP_0 に対する POSP_+，POSP_∞ の優位性が伺えるが，ループ数を増やすにつれて POSP_0，POSP_+，POSP_∞ はほぼ同等の解を出力している．一方，$\rho = 0.6, 0.8$ ではループ数を増やしても，POSP_0 に対する POSP_+，POSP_∞ の優位性は変わらない．表 5.6 からも分かるように，制約付与率が小さいということは，比較的距離が離れているモジュール対にのみ制約を付与していることを意味する．したがってこのようなモジュール対に関しては，重なり除去を行う前後での相対位置の変化が起こりにくいと考えられる．この場合，POSP_0，POSP_+，POSP_∞ の制約はほぼ等しくなり，初期解のみが異なるので，ループ数を増やして探索数が多くなると，解の差が縮まったのではないかと予想される．

以上これまでの議論を踏まえ，本研究でのモデル配置の導出には，POSP_+ が最もふさわしいと判断し，以降の実験には POSP_+ を提案手法の重なり除去に使用する．

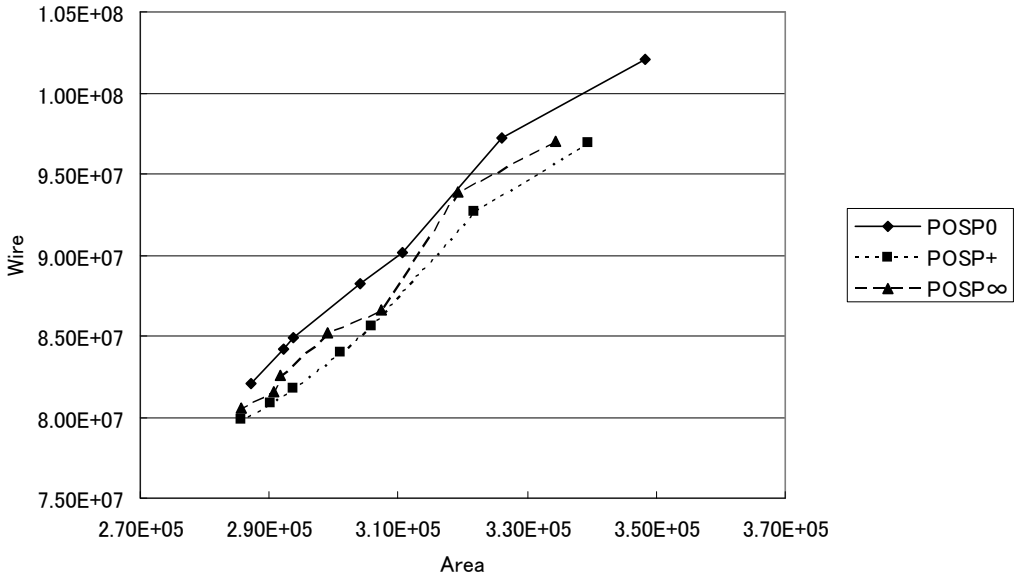

図 5.8: POSP_0，POSP_+，POSP_∞ の比較（n300a，$\rho = 0.6$）

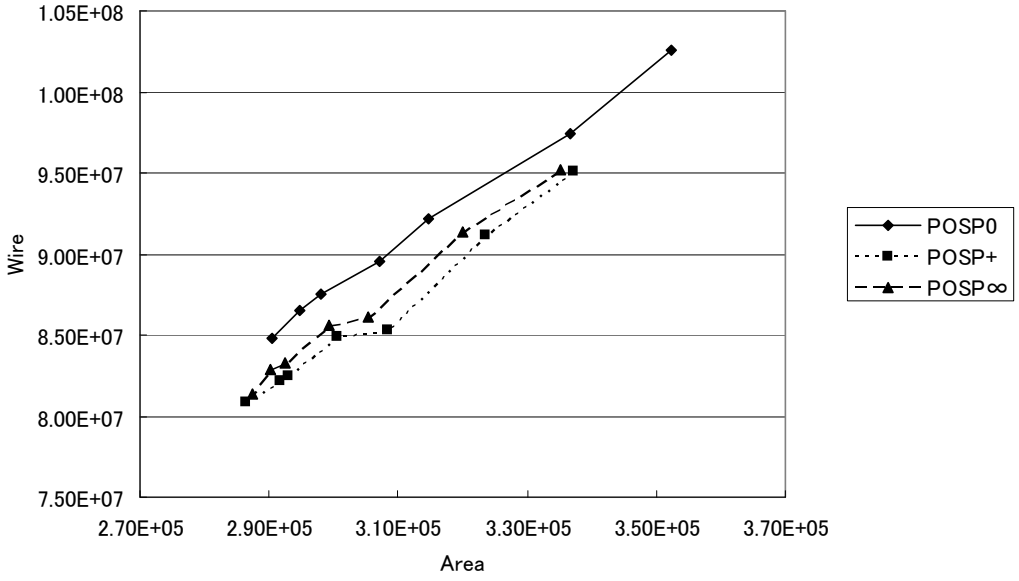

図 5.9: POSP_0，POSP_+，POSP_∞ の比較（n300a，$\rho = 0.8$）

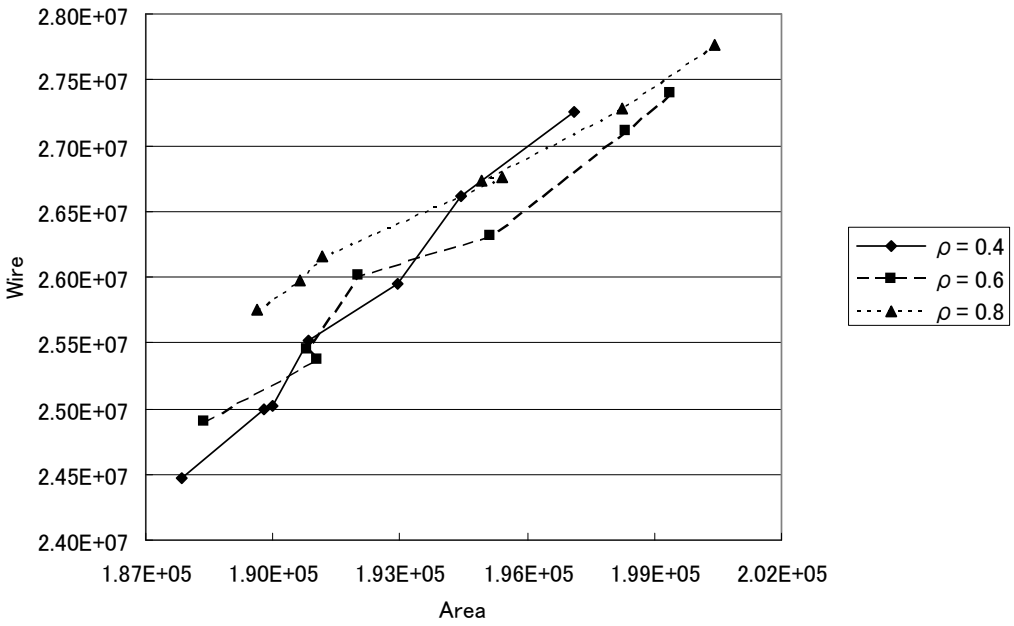

図 5.10: 制約付与率 ρ による解の変化（n100a, POSP_+）

5.3 制約付与数による解の変化

制約付与率 ρ が，解に及ぼす影響を観察する．図 5.10，図 5.11，図 5.12 は，それぞれ n100a，n200a，n300a の回路に対し，$\rho = 0.4, 0.6, 0.8$ と変化させたときの様子をグラフ化したものである．横軸は面積コスト，縦軸は配線コストを表している．

n100a の回路を除き，図右上，すなわちループ数の少ない場面では，ρ が小さい方の結果が配線コストに関して良い結果となっているが，ループ数が増すと，逆に ρ が大きいほど配線長コストが少ない．また，n100a の回路に関しても，グラフ上の線の傾きが ρ が小さいほど大きくなる傾向を示している．したがって，探索数を十分に取れる場合は，POSP に用いる半順序制約を絞り，時間的に探索数があまり取れない場合は，制約を増やすことで探索を効率化できると考えられる．

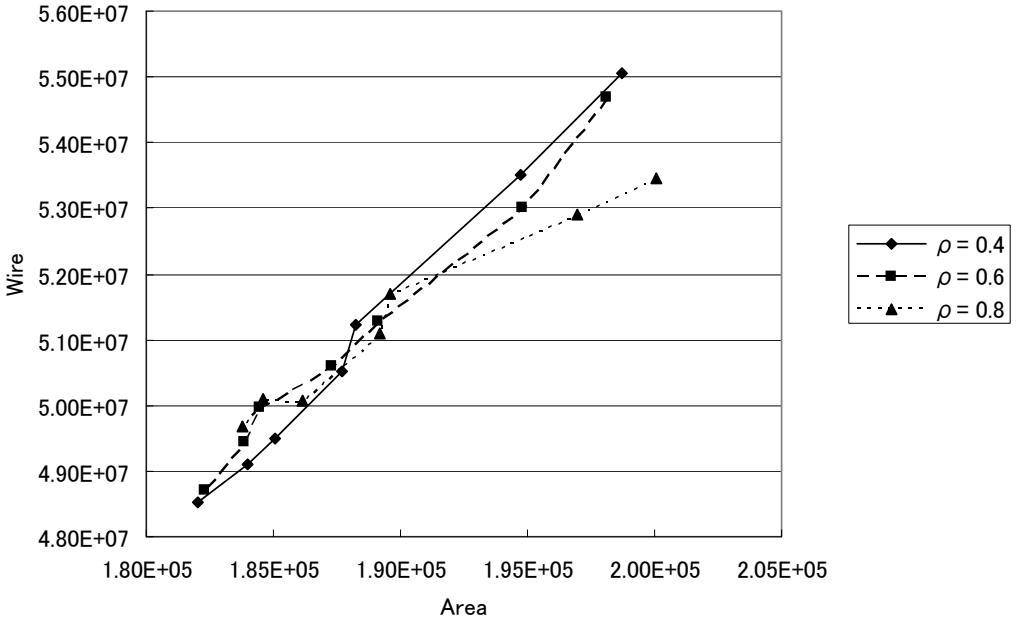

図 5.11: 制約付与率 ρ による解の変化（n200a, POSP_+）

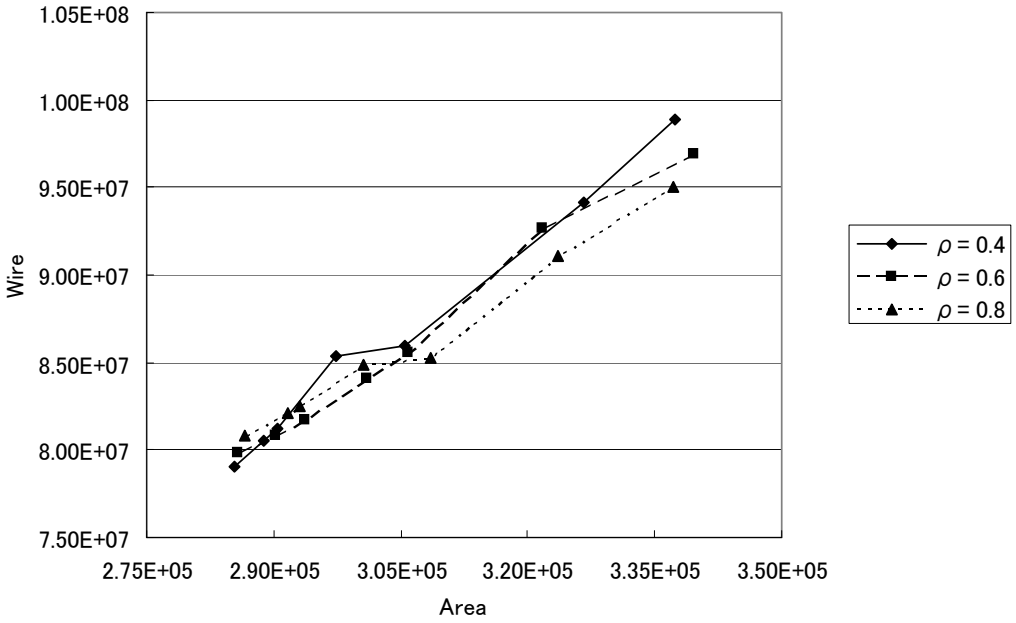

図 5.12: 制約付与率 ρ による解の変化（n300a, POSP_+）

5.4 従来手法との比較

制約付与のないシーケンスペアを SA で探索する従来手法と，力学的手法を利用して生成した POSP の解空間を探索する提案手法とを比較する．

5.4.1 コスト関数の重みによる解の変化

本研究で提案する手法は，シーケンスペアに二次配線長を配慮した制約を設けている．面積よりも配線長を重視した場合，従来手法ではコスト関数の面積と配線長の重み比 $\alpha : \beta$ を調節することでこれを実現する．したがって，従来手法の重み設定では達成できない配置を探索することができれば，提案手法の優位性が明らかになる．

そこで，$\rho = 0.4$ で固定した上で，配線長の重みを表 5.2 で示した重み比から，$\beta \times 1, \beta \times 3, \beta \times 5, \beta \times 10$ として，従来手法との比較実験を行った．n100a，n200a，n300a において，各 lp ごとに従来手法と提案手法を比較した結果を，それぞれ図 5.13，図 5.14，図 5.15 に示し，各回路でこれらをまとめた結果を図 5.16 図 5.17 図 5.18 に示す．各図とも横軸は面積コスト，縦軸は配線コストを表している．

SA の内部ループ数で見た場合，各回路とも lp が一定以上であれば従来手法が面積，配線ともコストが少ないが，それ以下であれば，提案手法は配線長に関して従来手法よりも低い値を導き出している．面積コストに関しては，各 lp で傾向が多少異なるが，従来手法と同等か若干大きい．

また，三回路で比較すると，n100a で $lp = 100, 200$，n200a では $lp = 2000 \sim 10000$，n300a では $lp = 10000 \sim 20000$ 程度まで，提案手法の配線コストが少ない．この点から，提案手法は，回路規模に対して SA の総探索数が少ないほど POSP の制約効果が大きく，従来手法と比較し配線長の短い配置を見つけ出すことができると考えられる．

最後に，提案アルゴリズムが出力した各回路の配置結果を図 5.19，図 5.20，図 5.21 に示す．

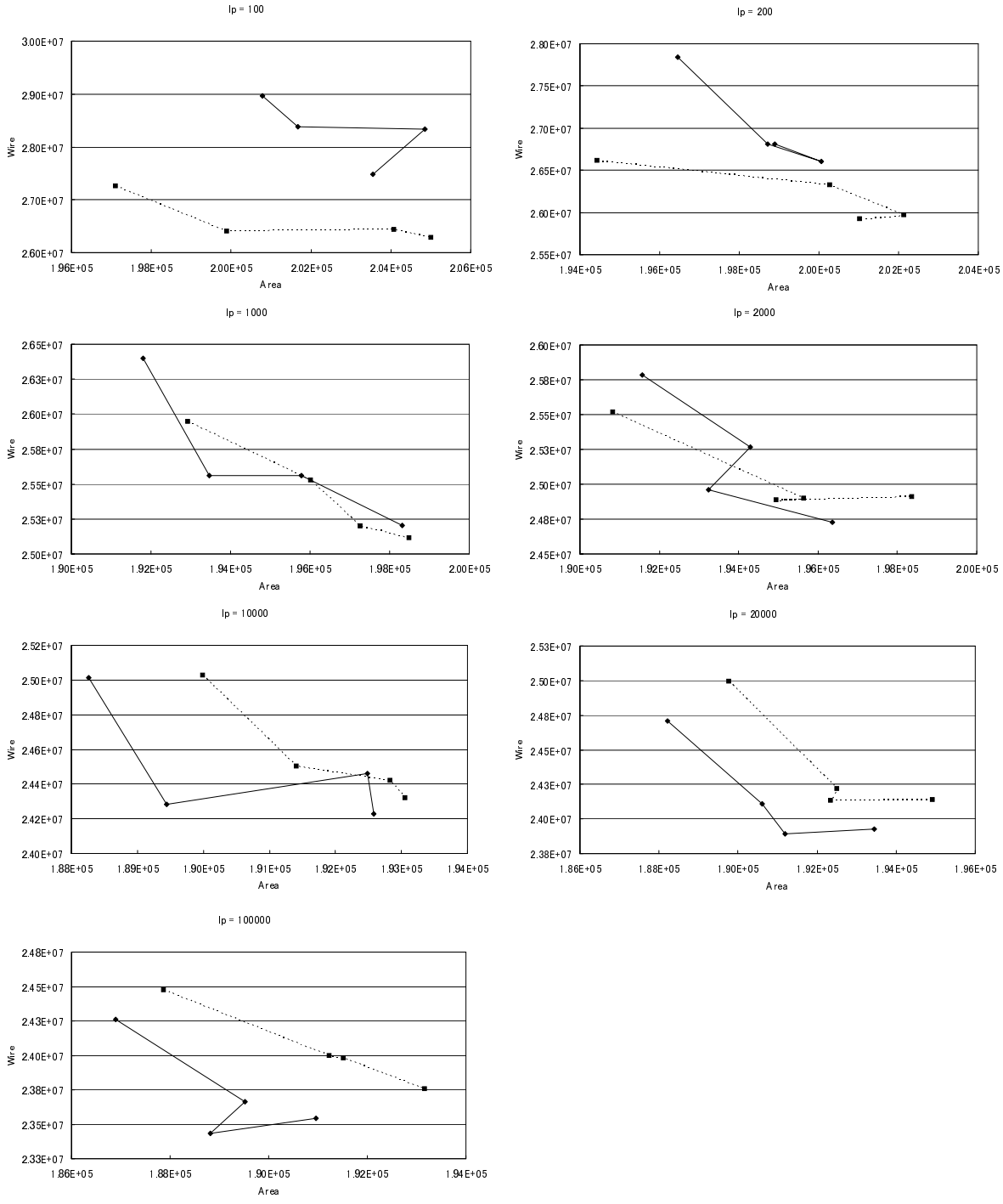

図 5.13: コスト関数の重み変化による解の変化（n100a，実線：SP，破線：POSP_+）

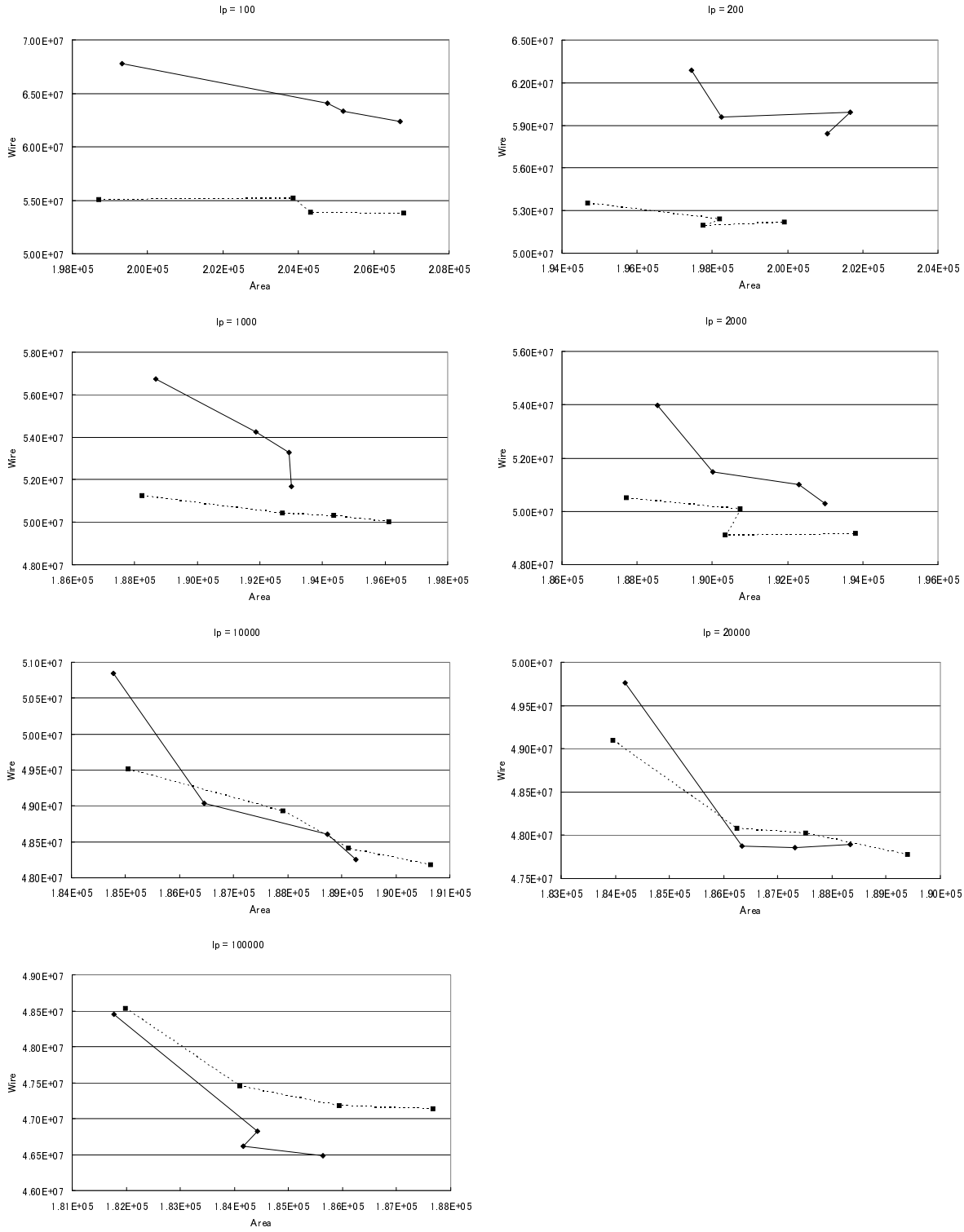

図 5.14: コスト関数の重み変化による解の変化（n200a，実線：SP，破線：POSP_+）

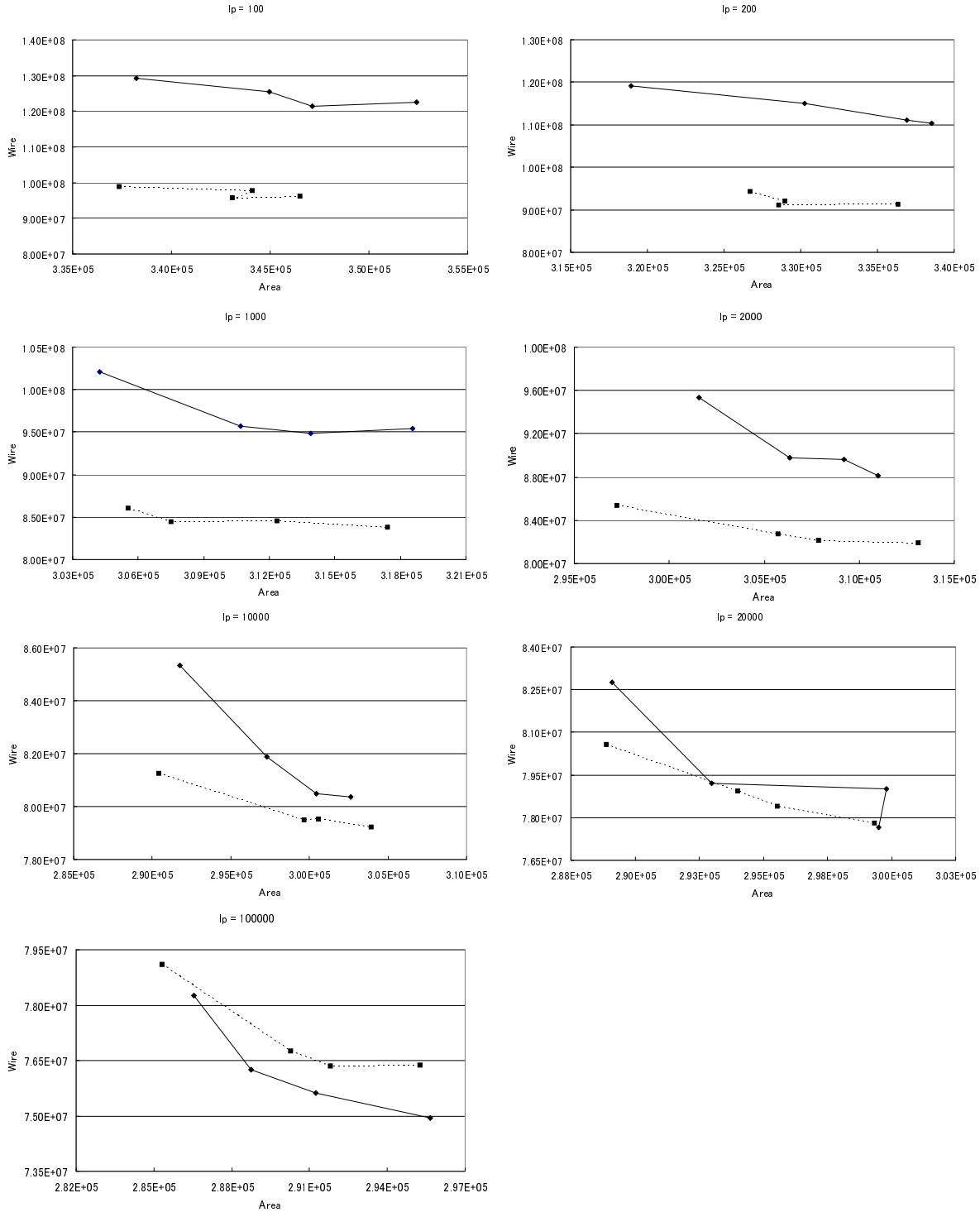

図 5.15: コスト関数の重みによる解の変化（n300a，実線：SP，破線：POSP_+）

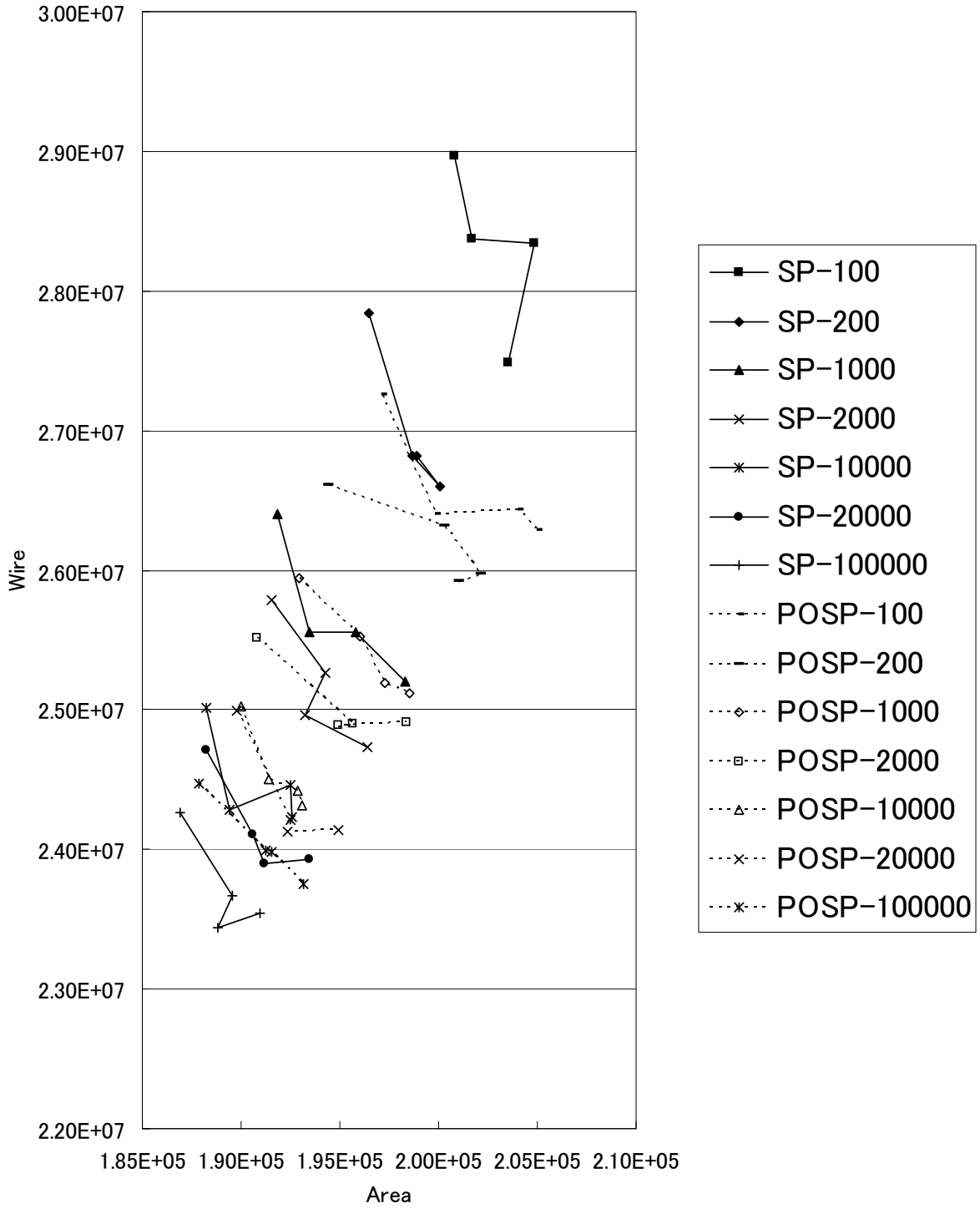

図 5.16: コスト関数の重みによる解の変化のまとめ（n100a，実線：SP，破線：POSP_+）

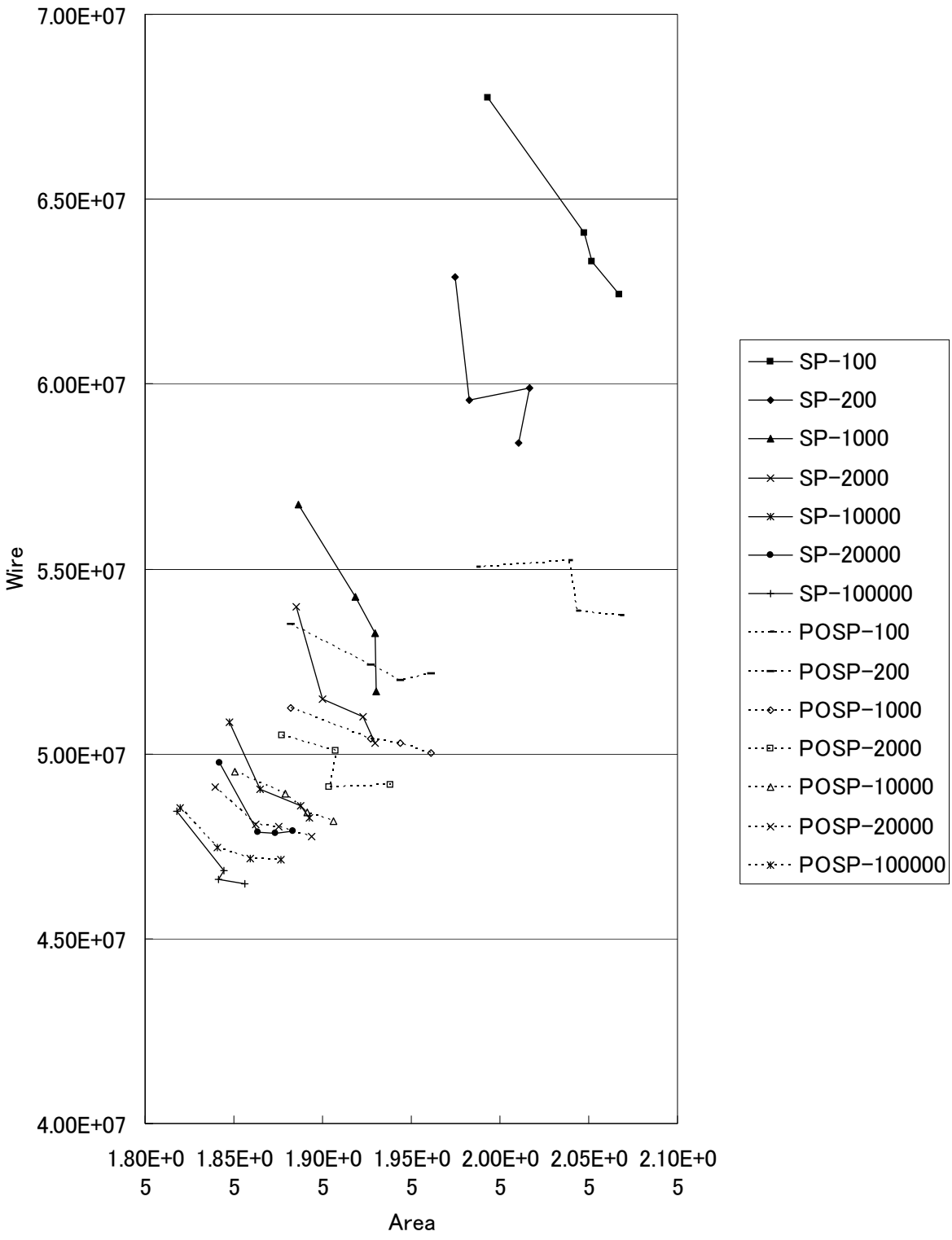

図 5.17: コスト関数の重みによる解の変化のまとめ（n200a，実線：SP，破線：POSP_+）

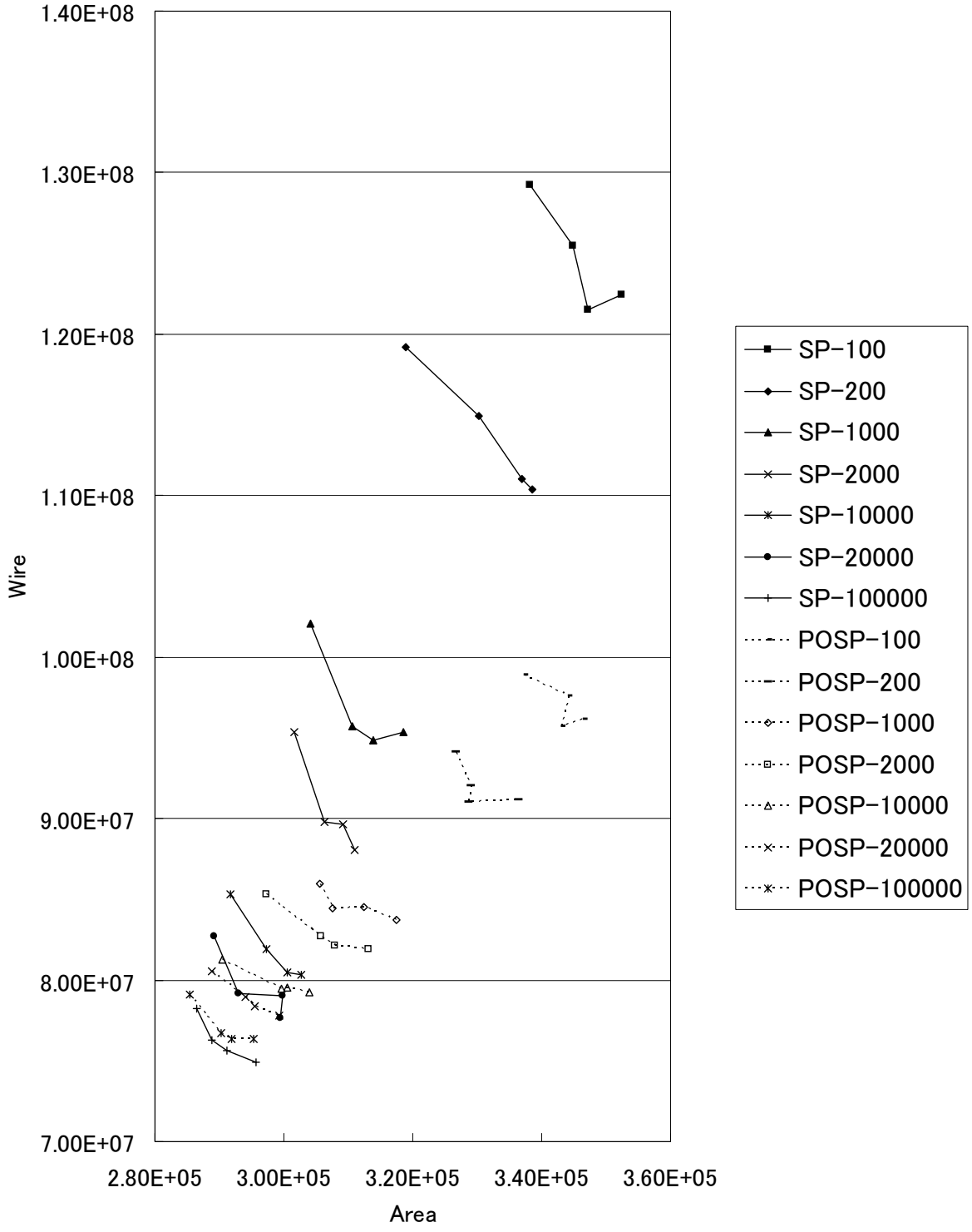

図 5.18: コスト関数の重みによる解の変化のまとめ（n300a，実線：SP，破線：POSP_+）

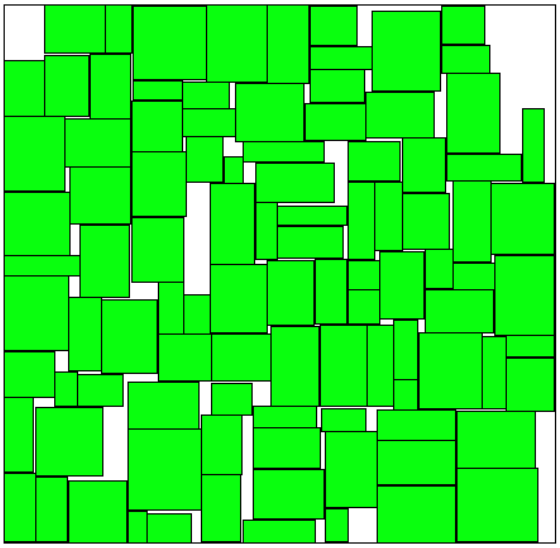

図 5.19: n100a の配置結果（$\mathrm{POSP}_{+}^{0.4}$，$lp = 100000$）

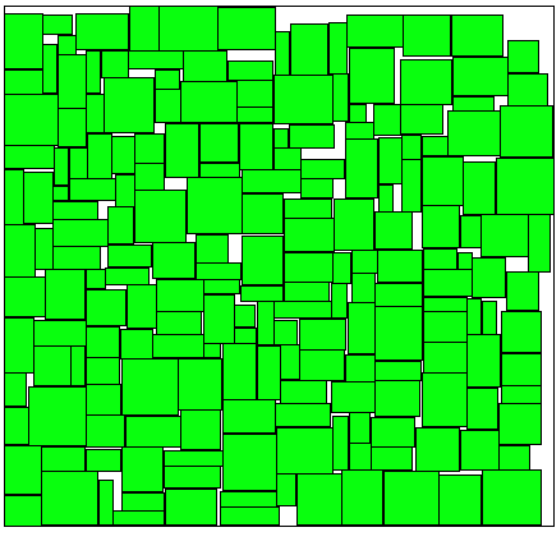

図 5.20: n200a の配置結果（$\mathrm{POSP}_{+}^{0.4}$，$lp = 100000$）

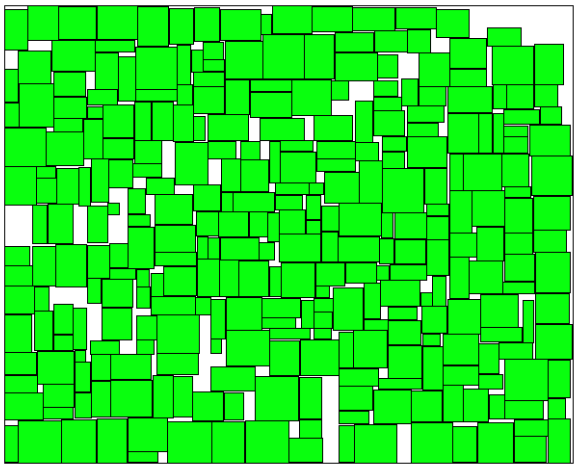

図 5.21: n300a の配置結果（$\mathrm{POSP}_{+}^{0.4}$，$lp = 100000$）

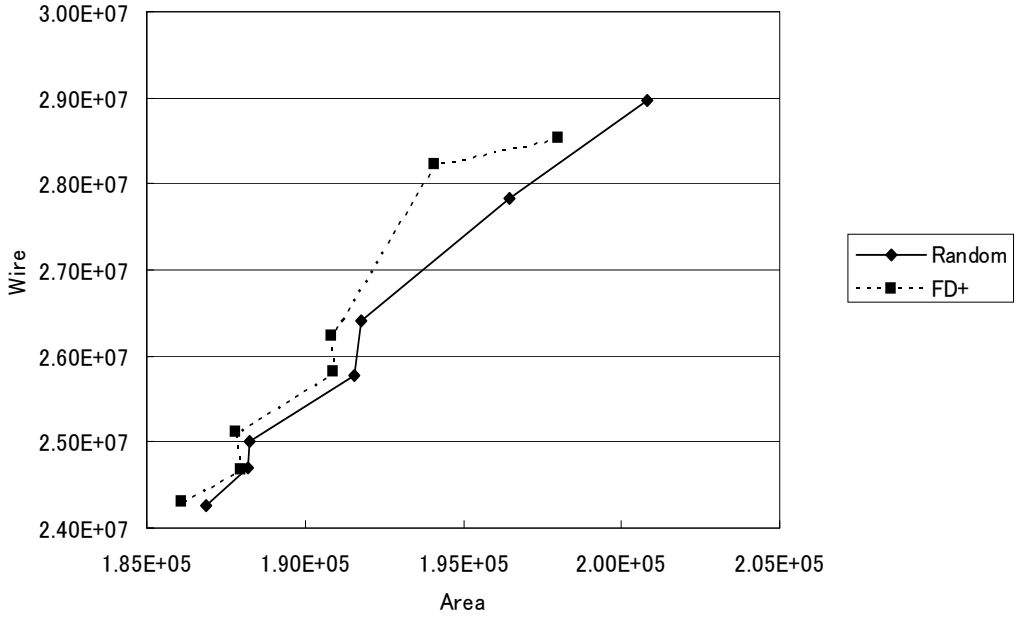

図 5.22: 初期解による影響（n100a）

5.4.2 初期解の影響

本研究での提案手法において，力学的手法によって導き出された初期解が，最終的に出力される解にどの程度の影響を与えているか検証する．初期解が最終結果に影響を与えていないとすれば，提案手法による解の質の改善は，初期解によるものではなく，シーケンスペアに課した半順序制約によるものであるとみなすことができる．

通常の SA で用いるランダムなシーケンスペアの初期解と，POSP_+ のモデル配置から得た初期解の二つから出発した SA による解を比較する．各回路の結果を図 5.22，図 5.23，図 5.24 に示す．ただし，Random はランダムなシーケンスペアの初期解を用いた結果，POSP_+ は力学的手法 POSP_+ を行って得られた初期解を用いた結果を表す．

いずれの回路に関しても，内部ループ数が多い場面では Random と POSP_+ との差は微小なものとなっており，SA の総探索数が一定以上であれば，初期解が最終結果に及ぼす影響は限定的であると考えられる．

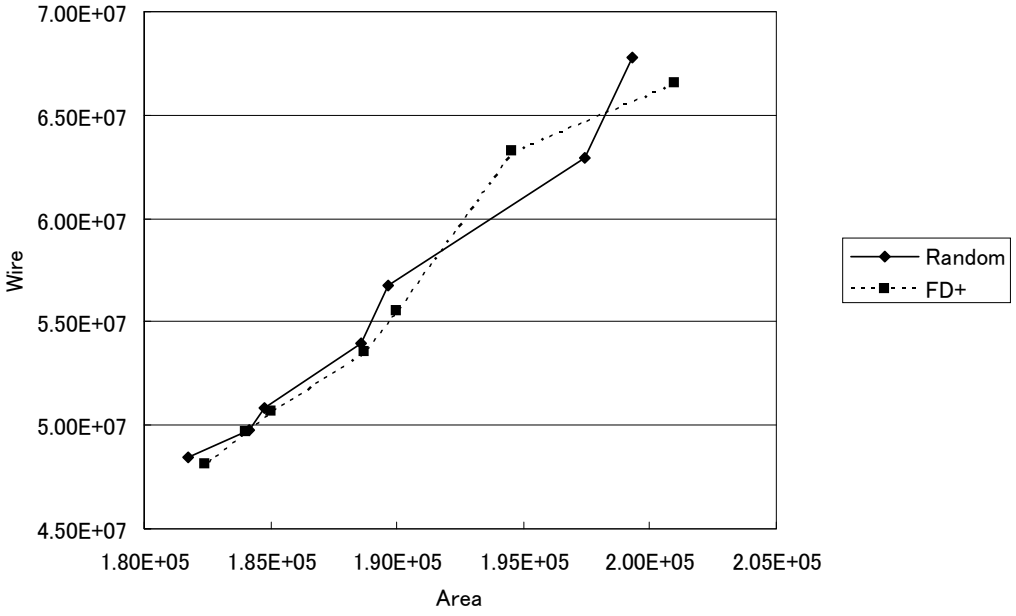

図 5.23: 初期解による影響（n200a）

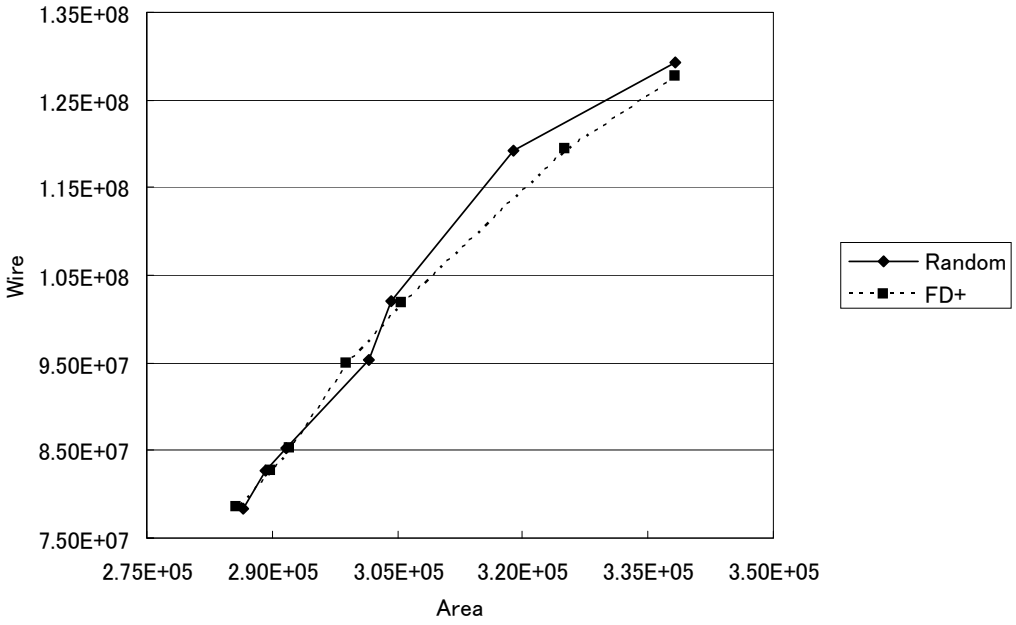

図 5.24: 初期解による影響（n300a）

表 5.9: n300a 回路における計算時間比較

lp	Computation time [sec]	
	SP	$\mathrm{POSP}_{+}^{0.8}$
100	2.67×10^1	2.94×10^1 (+10.32%)
200	5.36×10^1	5.89×10^1 (+ 9.95%)
1,000	2.68×10^2	2.94×10^2 (+ 9.37%)
2,000	5.37×10^2	5.88×10^2 (+ 9.61%)
10,000	2.69×10^3	2.94×10^3 (+ 9.26%)
20,000	5.36×10^3	5.88×10^3 (+ 9.57%)
100,000	2.68×10^4	2.96×10^4 (+10.42%)

5.4.3 実行時間

力学的手法による配置計算や，POSP の制約チェックにかかる時間的ペナルティの影響を調査するため，提案アルゴリズムのトータルの実行時間を計測し，検証を行った．比較対象は，半順序制約を設けない通常のシーケンスペアの解空間を探索する従来法である．実験環境として，OS が SUSE Linux 9.2，CPU は AMD Opteron 2.4GHz，メモリ 8.2GB，C コンパイラは gcc2.95 を用いた．n300a の回路に対し，SP の解空間を探索したものと，POSP（POSP_{+}，$\rho = 0.8$）の解空間を探索したものとで，計算にかかった時間を計測した．結果を表 5.9 に示す．

いずれの内部ループ数でも，約 10% 計算時間が延びる結果となった．4.5.3 で示したように，制約のチェックには挿入，交換操作ともに $O(n)$ の計算量を要する．本研究では，計算量が $O(n^2)$ であるデコードアルゴリズムを採用しているが，シーケンスペアのデコードは，$O(n \log n)$[3, 4]，$O(n \log\log n)$[5] など，より高速なアルゴリズムがすでに提案されており，これらを用いた場合では，制約チェックに要する時間がより大きな割合を占める可能性がある．しかし，提案手法によって得られる制約の中には，以下に示すような冗長な制約が含まれる．

図 5.25 のように，順列 A 中でモジュール a, b, c, d が，

$$A^{-1}(a) < A^{-1}(d) \tag{5.1}$$

かつ

$$A^{-1}(b) < A^{-1}(d) \tag{5.2}$$

かつ

$$A^{-1}(a) < A^{-1}(b) < A^{-1}(c) < A^{-1}(d) \tag{5.3}$$

のような半順序制約を持っている場合，式（5.3）の制約があれば，式（5.1），式（5.2）の制約が不要であるのは明らかである．これらの冗長な制約は，ρ を大きくとった場合に

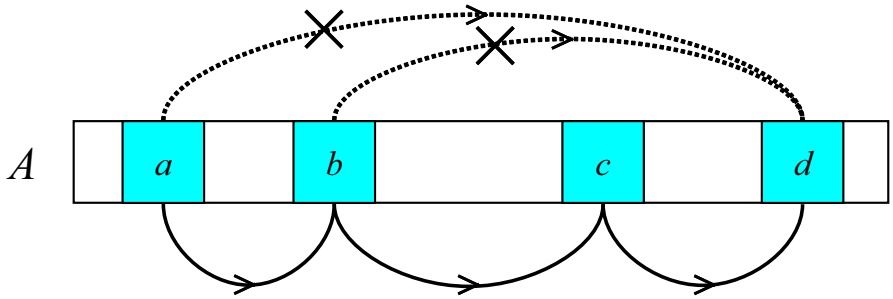

図 5.25: 冗長な半順序制約の除去

多く発生すると予想され，これらを取り除いて制約の総数を減らすことで，制約チェックに要する時間を短縮できると考えられる．

また，POSP の結果に関しては，力学的手法によるモデル配置の計算時間が含まれていないが，その計算はいずれの回路でも数分以内に終了している．この計算は一度行っておけば，SA を繰り返し行う場合でも計算し直す必要はない．したがって，他のより大規模な回路への適用においても，クリティカルな時間的ロスを引き起こすことはないと思われる．

以上のことから，本研究で提案したアルゴリズムのモデル配置導出，制約チェック等に要する時間的ペナルティは十分に許容可能であると思われる．

第6章　まとめ

6.1　モデル配置に基づく解空間縮小について

本研究では，モデルとなる配置におけるモジュール対の相対位置関係から，シーケンスペア上のモジュール出現順序を半順序制約として抽出した新しい解空間 POSP（Partially Ordered Sequence-Pair）を提案した．提案アルゴリズムは，力学的手法を利用して二次配線長が最小化された理想的なモデル配置を求め，POSP の制約に反映させた後に，SA による探索を行う．

本研究では先ず，モデル配置からシーケンスペア上の制約への具体的な変換手法を明らかにした後，縮小された解空間を SA にて探索するための準備を行った．すなわち，初期 POSP コード生成，隣接解定義を行い，この POSP 解空間が可到達性（任意の POSP コードから他の任意の POSP コードへと POSP コードのみを通って到達できる）を有することを証明した．

また，提案手法を評価するため，ベンチマーク回路を用いた配置実験を行った．この結果，以下の事柄が確認できた．

- 力学的手法によるモデル配置導出の際，モジュール同士の重なり除去操作を適切に行うことで，モデル配置としての有効性が高まる
- 制約を付与する対象をモジュール間距離の遠い 2 モジュールから順に選んだとき，その付与数が解の質に影響することから，以下のような方針を立てることで良質な解を得られる
  - 回路規模と比べ総探索数が少ない場合，付与する制約を多くする
  - 回路規模と比べ総探索数が十分にある場合，付与する制約を少なくする
- 従来手法と比較し，
  - 適用する回路の規模に対し総探索数が少ない場合ほど，二次配線長評価の良い配置を探索することができる
  - 実行時間がほぼ同等

6.2 今後の課題

今後の検討課題として，POSP に適した隣接解生成手法，配線長のバウンディングボックス評価に適した制約付与方法，解空間の縮小効果に関する検証などが挙げられる．

SA 探索における解空間の重要な性質として，可到達性のほかに，解空間の滑らかさが挙げられる．これは，隣接解同士が「似ている」ことを意味する．解空間が滑らかであれば，良いコストを持つ隣接解の近くに良いコストを持つ隣接解が存在することになり，効率的な SA 探索が可能となる．本研究では半順序制約を設けた新しい解空間 POSP を提案したが，隣接解の生成には，従来のシーケンスペアと同様の手法を用いた．解空間の滑らかさを実現するには，POSP に適した隣接解の生成法を検討する必要がある．

次に，本研究では配線長評価に二乗配線長評価を用いた．しかし，バウンディングボックスによる評価も一般的に用いられているため，これへの対応が求められる．力学的手法によるモデル配置では，二乗配線長の最小化が行われる．そのため，抽出される制約はバウンディングボックス評価に適さない恐れがあり，力学的手法に変わるモデル配置導出法や，提案手法とは異なる制約抽出法の検討が望まれる．

最後に，提案手法による配置実験では，従来手法と比較し，配線長に関するアドバンテージは認められた．しかし，探索する解空間を縮小したことによって期待された面積評価に関するアドバンテージは，はっきりと確認することができなかった．より大規模な回路の配置実験を行うことで，その効果が確認できる可能性がある．さらに，大規模回路では解空間の増大に伴い，従来法では実用時間内で良質解を得ることが困難となる．提案手法は総探索数が回路規模に対して少ない場合に従来法よりも良好な結果を導き出しており，大規模回路への適用において，従来法に対する優位性が期待される．

謝辞

本研究を進めるにあたり，北陸先端科学技術大学院大学 金子峰雄教授より，適切かつ暖かいご指導を受け賜りました．ここに深く感謝の意を表します．また多くの有用な意見，議論を頂いた同助教 岩垣剛氏，横河電機株式会社 大橋功治氏，研究室の皆様にも感謝いたします．

参考文献

[1] C. Chang, J. Cong, M. Romesis, and M. Xie, "Optimality and scalability study of existing placement algorithms," *IEEE Trans. CAD*, Vol. 23, No. 4, pp. 537–549, Apr. 2004.

[2] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "VLSI module placement based on rectangle-packing by the sequence-pair," *IEEE Trans. CAD*, Vol. 15, No. 12, pp.1518–1524, Dec. 1996.

[3] 高橋俊彦, "矩形パッキングのための最大重み減少列を求めるアルゴリズム," 信学技法, VLD96-30, pp. 31–35, Jul. 1996.

[4] X. Tang, R. Tian, and D. F. Wong, "A fast evaluation of sequence pair in block placement by longest common subsequence computation," *Proc DATE 2000*, pp. 106–111, 2000.

[5] X. Tang, and D. F. Wong, "FAST-SP: a fast algorithm for block placement based on sequence pair," *Proc. ASPDAC 2001*, pp. 521–526, 2001.

[6] S. Tayu, T. Obata and M. Kaneko, "Efficient search on solution space based on sequence-pair for simulated annealing approach," *IEICE Technical Report (VLD2002-5)*, Vol. 102, No. 72, pp. 25–30, May 2002.

[7] H. Eisemann and F. M. Johannes, "Generic global placement and floorplanning," *Proc. Design Automation Conf.*, San Francisco, CA, Jun. 1998, pp. 269–274.

[8] http://www.cse.ucsc.edu/research/surf/GSRC/bench1.html

[9] M. Kaneko, "Solution space reduction of sequence pairs using model placement," *IEICE Technical Report (CAS2006-5)*, Vol. 106, No. 111, pp. 25–28, May 2006.

[10] 矢野勇生, 金子峰雄, "二次配線長最小化を利用したシーケンスペアの解空間縮小," 信学技報, CAS2006-59, pp. 25–30, Nov. 2006.

[11] 藤吉邦洋, 大村智一, 井尻堅大, "Simulated Annealing 法探索に適した Sequence-Pair によるパッキング解空間," 信学技法, VLD99-118, pp. 9–16, Mar. 1999.