

Title	周期的繰り返し矩形配置の解表現と配置最適化
Author(s)	小川, 智之
Citation	
Issue Date	2005-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1929
Rights	
Description	Supervisor:金子 峰雄, 情報科学研究科, 修士

修 士 論 文

周期的繰り返し矩形配置の解表現と配置最適化

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

小川 智之

2005 年 3 月

修 士 論 文

周期的繰り返し矩形配置の解表現と配置最適化

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

310025 小川 智之

提出年月: 2005 年 2 月

概要

矩形配置問題とは、二次元平面上に複数の矩形を互いに重なることなく配置する問題であり、VLSI レイアウトを始めとして、様々な応用を持つ最適化問題である。矩形の配置領域の面積最小化などの最適配置問題はNP 困難に属することが知られており、近似解法や Simulated Annealing 法などの確率的最適化法が考案されている。

本研究では、繰り返し配置と呼ばれる特殊な配置問題を取り扱う。これは基本矩形集合が周期的に複数回（有限回、あるいは無限回）繰り返し並ぶ配置である。このような矩形の周期的繰り返し配置は、同一構造のプロセッサが複数個並ぶオンチッププロセッサアレイ、基本論理セルとスイッチボックスが規則正しく並ぶFPGA、あるいはメモリのような同一の回路構造が繰り返し配置されるVLSI レイアウトに現れる問題である。また、複数セットの部品切り出しや、矩形集合の円筒側面上への配置（平面化すると無限繰り返しとなる）、ループ状の計算処理におけるループパイプラインスケジューリング（ガントチャート表現）など、回路レイアウトにとどまらない応用も期待される。

本研究では、繰り返し配置に対する研究の第一段階として、水平方向への一次元的な繰り返し配置について検討し、繰り返し配置の組み合わせ問題としての解表現、配置の最適化手法の開発を行った。

繰り返しのない通常の矩形配置問題の解表現手法としてシーケンスペアが提案されている。シーケンスペアは矩形名からなる二つの順列 (Γ_+, Γ_-) によって構成され、各順列の並びによってすべての二矩形間の相対位置関係を矛盾無く規定するものである。与えられた矩形配置からシーケンスペアを抽出（エンコード）する操作としてグリッディングと呼ばれるアルゴリズムがあるが、周期的繰り返し配置に対してもそれを適用することが可能であり、抽出される二つの順列 Γ_+ と Γ_- は、ともに矩形名が周期的に繰り返す順列となる。一方、 Γ_+, Γ_- から逆に繰り返し配置を導く（デコード）することを考えた場合、繰り返し配置では一周内の相対位置関係だけでなく異なる周期にある矩形同士の相対位置関係も規定する必要があるため、二矩形間の相対位置関係を一意に決定することができないという問題に直面する。

このデコードにおける問題を解決するため、二つの順列に加え第三の順列 Γ_s を導入することで、周期的繰り返し配置を解表現する新たなコーディングシステムを開発した。以下、この三つの順列による解表現手法をシーケンスペアにならい、シーケンストリプルと呼ぶ。シーケンストリプルでは、周期的に繰り返される Γ_- の一周内における Γ_+ の並びを Γ_s によって規定する。シーケンスペアと比べ、順列を一つ多く使うが、一周内の矩形間の相対位置関係だけでなく、異なる周期間にある矩形同士の相対位置関係をも規定することが可能となった。なお、矩形数を n とするとき、本研究で開発したシーケンストリプルのデコードの計算量は $O(n^3)$ であり、解空間の大きさは $O((n!)^3)$ である。

シーケンストリプルにより定義される解空間を Simulated Annealing 法により探索し、準最適な矩形配置を求めるプログラムをC言語を用いて実装し、矩形配置生成の実験を

行った．入力は無作為に生成した各矩形の幅と高さとし，“繰り返し周期の幅 L × 配置領域の高さ H ”が最小となることを最適化の目標とした．様々な入力に対して実験を行った結果，最適化される配置には縦長になる傾向があることがわかった．この傾向は，ランダムに生成されたシーケンストリプルにおいて，二つの矩形が上下関係になる確率と左右関係になる確率に偏りがあることから裏付けられ，良好な解の探索のためには，こうした偏りを打ち消すような探索上の工夫が必要と考えられる．

今後の課題として，良好な解を得るための解空間探索手法，デコードアルゴリズムの高速化，計算量と解空間の大きさについてより優れた解表現手法の開発が挙げられる．

目次

第1章	はじめに	1
第2章	準備	2
2.1	シーケンスペア	2
2.2	矩形配置からシーケンスペアへ	4
2.3	シーケンスペアから矩形配置へ	6
2.4	シーケンスペアの計算量	9
第3章	シーケンストリプル	10
3.1	矩形集合の周期的繰り返し配置	10
3.2	問題点	11
3.3	提案手法	11
3.4	シーケンストリプルから矩形配置へ	13
3.5	補足	19
第4章	シーケンストリプルのアルゴリズム	22
4.1	シーケンストリプルによる矩形配置座標計算	22
4.2	45度回転の格子における矩形のアドレス	22
4.3	矩形間の左右関係	23
4.4	矩形間の上下関係	24
4.5	全体アルゴリズム	25
4.6	シーケンストリプルの計算量	25
第5章	実験	26
5.1	実験方法	26
5.2	実験結果	27
5.3	考察	41
第6章	結論	42
6.1	まとめ	42
6.2	今後の課題	42

第1章 はじめに

様々なサイズの矩形を二次元平面上に互いに重なることなく配置する問題は、VLSI レイアウトを始めとして、様々な応用を持つ最適化問題である。矩形の配置領域の面積最小化などの最適配置問題は、NP 困難に属することが知られており、近似解法や Simulated Annealing 法などの確率的最適化法が考案されている。

本研究では、繰り返し配置と呼ばれる特殊な配置問題について考える。これは基本矩形集合が周期的に複数回（有限回、あるいは無限回）繰り返される配置である。このような矩形の周期的繰り返し配置は、同一構造のプロセッサが複数個並ぶオンチッププロセッサアレイや、基本論理セルとスイッチボックスが規則正しく並ぶ FPGA、あるいはメモリのような同一の回路構造が繰り返し配置される VLSI レイアウトにしばしば見られる配置である。また、複数セットの部品切り出しや、矩形集合の円筒側面上への配置（平面化すると無限繰り返しとなる）、ループ状の計算処理におけるループパイプラインスケジュール（ガントチャート表現）など、回路レイアウトにとどまらない応用も期待される。

本研究では、繰り返し配置に対する研究の第一段階として、水平方向への一次元的な繰り返し配置について検討する。繰り返し配置の組み合わせ問題としての解表現手法、そして、周期的繰り返し配置の最適化手法の開発を目的とし、研究を行った。

本論文では、まず始めに、繰り返しの無い通常の矩形配置問題の解表現手法として提案されているシーケンスペアについて解説する。次に、本研究で扱う周期的繰り返し配置を紹介し、その特徴や、周期的繰り返し配置に対してシーケンスペアを適用する際に生じる問題点についてを説明する。そして、その問題を解決するために開発した解表現手法“シーケンストリプル”と、そのアルゴリズムについて解説する。さらに、シーケンストリプルと Simulated Annealing 法を用いて準最適な矩形配置を求める実験について説明し、実験結果からシーケンストリプルの特徴や効率について考察する。

第2章 準備

本章では、本研究の準備として、矩形の配置最適化問題において利用されるシーケンスペアと呼ばれる矩形配置の解表現手法について解説する。

2.1 シーケンスペア

繰り返し配置ではない通常の矩形配置について、矩形配置の解表現手法として、シーケンスペア (Sequence-Pair) が提案されている。シーケンスペアは、配置される矩形名で表される二つのシーケンス (Γ_+, Γ_-) で構成される。それぞれのシーケンスには、それぞれのシーケンスには各矩形名が1個ずつ含まれ、各シーケンスの並びによって任意の二つの矩形間の相対位置関係が表現される。例えば、

$$(\Gamma_+, \Gamma_-) = (\dots a \dots b \dots, \dots a \dots b \dots)$$

というように、それぞれのシーケンスに矩形 a 、矩形 b が同じ順に現れる場合、“矩形 a は矩形 b の左側に位置する（矩形 b は矩形 a の右側に位置する ”と解釈され、また、

$$(\Gamma_+, \Gamma_-) = (\dots a \dots b \dots, \dots b \dots a \dots)$$

というように、矩形 a 、矩形 b がそれぞれ逆順に現れる場合、“矩形 a は矩形 b の上側に位置する（矩形 b は矩形 a の下側に位置する）”と解釈される。

この相対的な位置関係は、45度回転させた格子を用いて図示することができる。例として、 $(\Gamma_+, \Gamma_-) = (abc, bac)$ に対する図示を考える。

1. 45度傾いた右上がりなライン、右下がりなラインを矩形の数だけ、それぞれのラインが交差するように引く。
2. 右上がりのラインに Γ_+ の順、右下がりのラインに Γ_- の順に、矩形名のラベルを付ける。
3. 同一のラベルを持つラインが交差する点に、そのラベル名の矩形を置く。

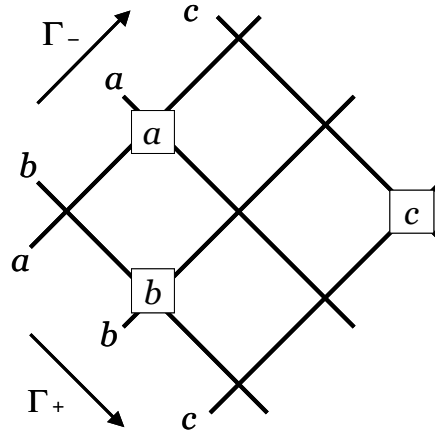


図 2.1: $\Gamma_+, \Gamma_- = (abc, bac)$

この例では，図 2.1 の配置が得られる．

各矩形間の相対位置関係は，二つの矩形が置かれた交点を利用して表される．二つの矩形がラインにより形成される格子の左右の交点に配置される場合，その二つの矩形間には左右の関係があり，格子の上下の交点には位置される場合，その二つの矩形間には上下の関係がある (図 2.2)．図 2.1 の例では，矩形 a と矩形 c の間，矩形 b と矩形 c の間には左右の関係があり，矩形 a と矩形 b の間には上下の関係がある．

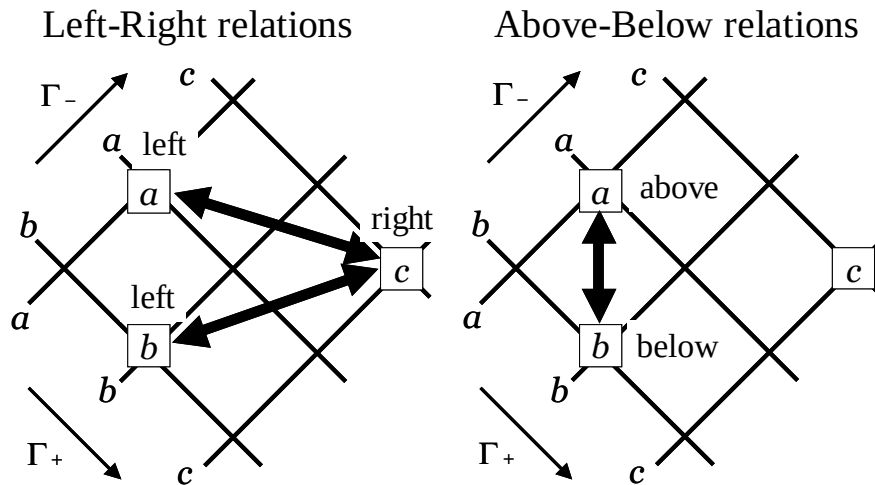


図 2.2: 相対位置関係

このように抽出された矩形間の相対位置関係は，矩形配置では図 2.3 のようになる．

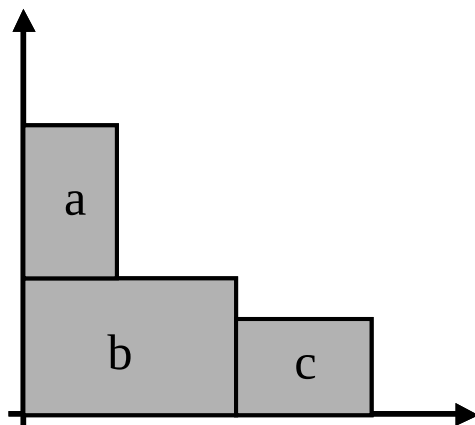


図 2.3: 矩形配置例

2.2 矩形配置からシーケンスペアへ

実際の矩形配置をコード化し、シーケンスペアを抽出するには、Gridding と呼ばれる手続きを行う。Gridding は、以下のように定義される。

まず、水平方向の相対位置関係について各矩形 x について、

1. 矩形 x の左上の頂点より上方に向けてラインを引く。
2. 他の矩形、または他の矩形から引かれているラインに突き当たった場合、ラインを引く方向を左方向へと変更する。
3. 再び、上方へラインを引くことが可能となった場合、上方へラインを引く。
4. 他の矩形や他の矩形から引かれているラインと交差しないように、手順 2, 3 を矩形配置領域の上方の境界に到達するまで繰り返す。

このようにして、矩形 x から得られるラインを x の Upper-Right step-line と呼ぶ。さらに、同様にして矩形の左下の頂点から、 x の Left-Down step-line を描ける。このようにして得られる x の Upper-Right step-line, Left-Down step-line と二つの step-line を結ぶ対角線を統合したラインを x の Positive step-line と呼ぶ。Positive step-line は互いに交差しないため、この step-line の並びは、各ラインに相当する矩形の並びとなる。Positive step-line を左から順に並べ、その順に矩形名を並べたものが Γ_+ である。図 2.4 は Positive step-line の例である。この配置例から抽出される Γ_+ は $acbde$ である。

次に、垂直方向の相対位置関係についても以下のように Gridding を行う。
各矩形 x について、

1. 矩形 x の左上の頂点より上方に向けてラインを引く.
2. 他の矩形, または他の矩形から引かれているラインに突き当たった場合, ラインを引く方向を左方向へと変更する.
3. 再び, 上方へラインを引くことが可能となった場合, 上方へラインを引く.
4. 他の矩形や他の矩形から引かれているラインと交差しないように, 手順 2, 3 を矩形配置領域の上方の境界に到達するまで繰り返す.

このようにして矩形 x から得られるラインを x の Upper-Left step-line と呼ぶ. 同様に, 矩形の右下の頂点から, x の Right-Down Step-line を描ける. このようにして得られる x の Upper-Right step-line, Right-Down step-line と二つのステップラインを結ぶ対角線を統合したラインを x の Negative step-line と呼ぶ. Negative step-line も, Positive step-line 同様, 互いに交差することはないため, この step-line の並びは, 各ラインに相当する矩形の並びとなる. Negative step-line を左から順に並べ, その順に矩形名を並べたものが Γ_- である. 図 2.5 は Negative step-line の例である. この配置例から抽出される Γ_- は $dcaeb$ である.

このように, Gridding を行うことにより, 任意の矩形配置をコード化し, シーケンスペアとして表現することができる.

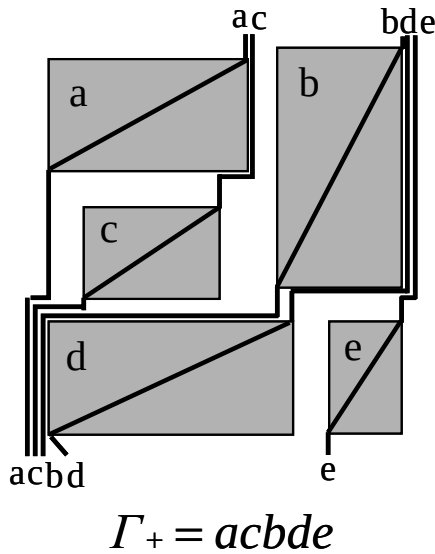


図 2.4: Positive step-lines

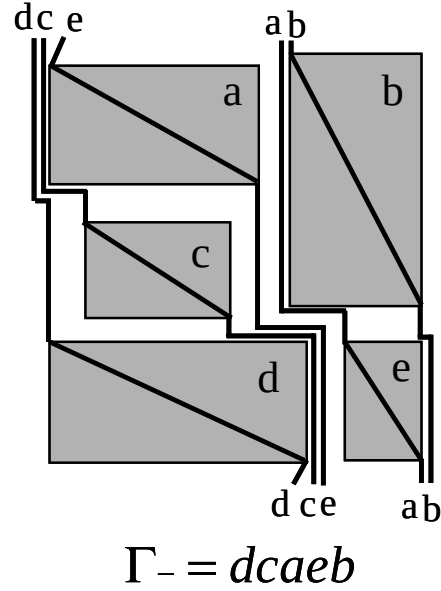


図 2.5: Negative step-lines

2.3 シーケンスペアから矩形配置へ

次に、シーケンスペアから実際の矩形配置を抽出する手法について解説する。2.1 節で解説したように、シーケンスペアによって、すべての二つの矩形間についての相対位置関係を特定できる。本節ではこの相対位置関係を定式化し、各矩形の配置座標を計算することについて解説する。

まず、各変数を次のように定義する。

- $x(a) \cdots$ 矩形 a の左下頂点の x 座標
- $y(a) \cdots$ 矩形 a の左下頂点の y 座標
- $w(a) \cdots$ 矩形 a の幅
- $h(a) \cdots$ 矩形 a の高さ

すると、相対位置関係から、左右関係にある二つの矩形について、次の制約式が得られる。

- 矩形 b が矩形 a の右側に位置する場合

$$x(b) \geq x(a) + w(a) \quad (2.1)$$

式 2.1 は、矩形 b の x 座標は、矩形 a の x 座標に矩形 a の幅を加えた座標よりも大きい（つまり座標平面上において右側に位置する）ことを意味する。また、上下関係については次の制約式が得られる。

- 矩形 b が矩形 a の上側に位置する場合

$$y(b) \geq y(a) + h(a) \quad (2.2)$$

式 2.2 は、矩形 b の y 座標は、矩形 a の y 座標に矩形 a の高さを加えた座標よりも大きい（つまり座標平面上において上側に位置する）ことを意味する。

シーケンスペアから矩形配置座標を導くには、

1. すべての矩形間について、左右関係もしくは上下関係を数え上げ、制約式を定式化する。
2. 各矩形について、すべての制約式を満たす x 座標、 y 座標を計算する。

また、矩形の重なり無しに、配置領域の面積が最小となることを目的とするため、各矩形の x 座標、 y 座標は、取りうる値のうち最小のものとする。このような制約の下で最適な矩形配置座標を計算するために、グラフを利用する。

矩形間の左右関係を水平制約グラフ $G_H(V, E)$ 、上下関係を垂直制約グラフ $G_V(V, E)$ として表現する。それぞれの制約グラフは重み付き有向グラフである。 $G_H(V, E)$ は次のように構成される。

V : source s , sink t , 矩形名をラベルとして持つ m 個の頂点

E : 各矩形 a について辺 (s, a) , 辺 (a, t) , 式 2.1 が成り立つ二つの矩形 a , 矩形 b 間を接続する辺 (a, b)

辺の重み: s に接続される辺 (s, a) の重みは 0, s 以外から出ている辺 (a, b) の重みは矩形 a の幅

また、 $G_V(V, E)$ は次のように構成される。

V : source s , sink t , 矩形名をラベルとして持つ m 個の頂点

E : 各矩形 a について辺 (s, a) , 辺 (a, t) , 式 2.2 が成り立つ二つの矩形 a , 矩形 b 間を接続する辺 (a, b)

辺の重み: s に接続される辺 (s, a) の重みは 0, s 以外から出ている辺 (a, b) の重みは矩形 a の高さ

いずれのグラフも閉路を含まない。矩形 a の x 座標は、 G_H における s から a までの最長経路長であり、矩形 a の y 座標は、 G_V における s から a までの最長経路長である。 G_H は a , 矩形 b が左右関係にある場合のみ辺 (a, b) が存在するため、結果として得られる矩形配置の水平方向には矩形の重なりは無い。同様に、 G_V は矩形 a , b が上下関係にある場合のみ辺 (a, b) が存在するため、結果として得られる矩形配置の垂直方向には矩形の重なりは無い。また、どの二つの矩形も、同時に左右関係、上下関係の両方の関係であることは無いため、矩形の重なりは無い。

矩形配置領域の幅と高さは、それぞれ G_H の s から t までの最長経路長、 G_V の s から t までの最長経路長となる。矩形配置領域の幅と高さは最小であるため、結果として得られる矩形配置は、制約下でもっとも矩形配置領域の面積の小さいものとなる。

例として、 $(\Gamma_+, \Gamma_-) = (acbde, dceab)$ から抽出される矩形の相対位置関係をグラフ表現したものが図 2.6、図 2.7 である。

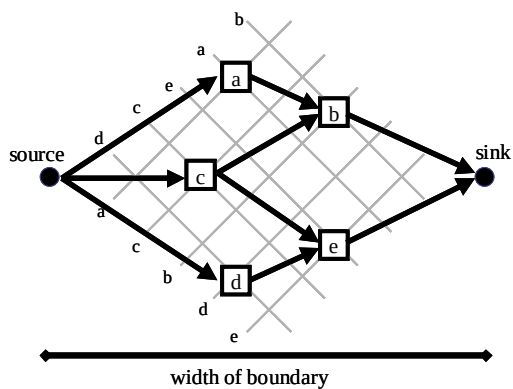


図 2.6: 水平制約グラフ G_H

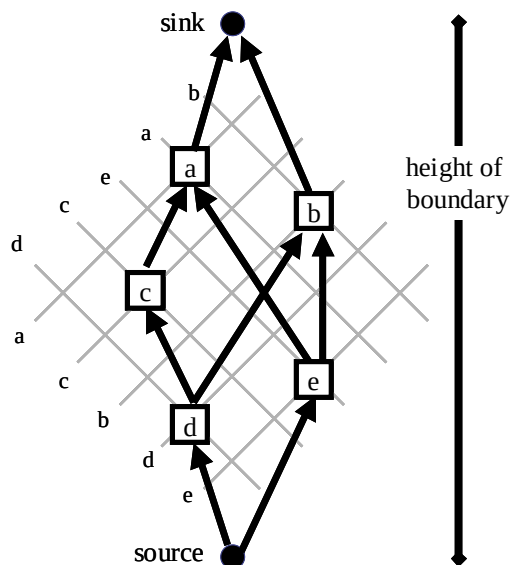


図 2.7: 垂直制約グラフ G_V

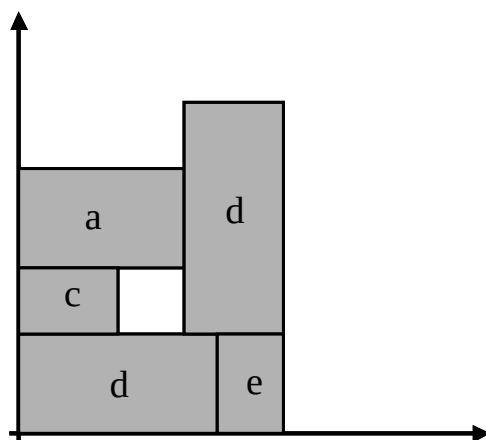


図 2.8: 得られた矩形配置

また、この (Γ_+, Γ_-) について、各矩形の幅を $w(a) = 5$, $w(b) = 3$, $w(c) = 3$, $w(d) = 6$, $w(e) = 2$, 高さを $h(a) = 3$, $h(b) = 7$, $h(c) = 2$, $h(d) = 3$, $h(e) = 3$ とした場合、図 2.8 の矩形配置が導き出される。

2.4 シーケンスペアの計算量

矩形数を n とすると、シーケンスペアの座標計算の計算量は $O(n^2)$ である。また、解空間の大きさは $(n!)^2$ である。

第3章 シーケンストリプル

本章では，まず始めに本研究で扱う矩形集合の周期的繰り返し配置について紹介し，その特徴やシーケンスペアで解表現する際に遭遇する問題点を解説する．そして，周期的繰り返し配置を扱うために開発した解表現手法“シーケンストリプル”について解説する．

3.1 矩形集合の周期的繰り返し配置

本研究では，図 3.1 のような，矩形集合が水平方向へのみ一次元的に繰り返し配置される矩形集合を扱う．この繰り返し配置では，各矩形は周期幅 L ごとに繰り返し配置されるため，矩形 a の左下の頂点の x 座標は，

$$x(a) + iL \quad (i = \dots - 2, -1, 0, 1, 2 \dots)$$

である．一方，垂直方向への繰り返しは扱わないため，矩形 a の左下の頂点の y 座標は，

$$y(a)$$

である．

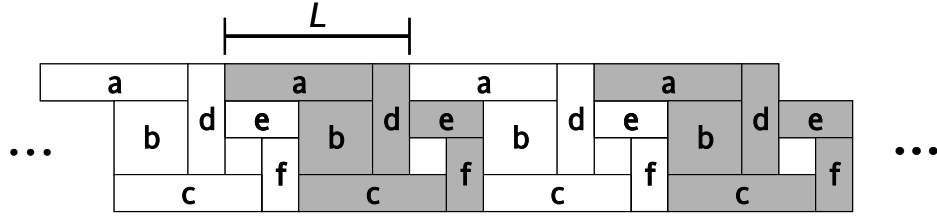


図 3.1: 周期的繰り返し配置

このような繰り返し配置に対しても，通常の矩形配置と同様にグリiddingを行うことにより，シーケンスペアを抽出することが可能である（エンコード）．抽出されたシーケンスペアは， Γ_+ ， Γ_- とともに一定周期ごとに繰り返される．例えば，図 3.1 についてグリiddingを行うと，図 3.2 のように，次のシーケンスペアが抽出される．

$$(\Gamma_+, \Gamma_-) = (\dots aecfbdaecfbdae \dots, \dots fecbadfecbadfe \dots)$$

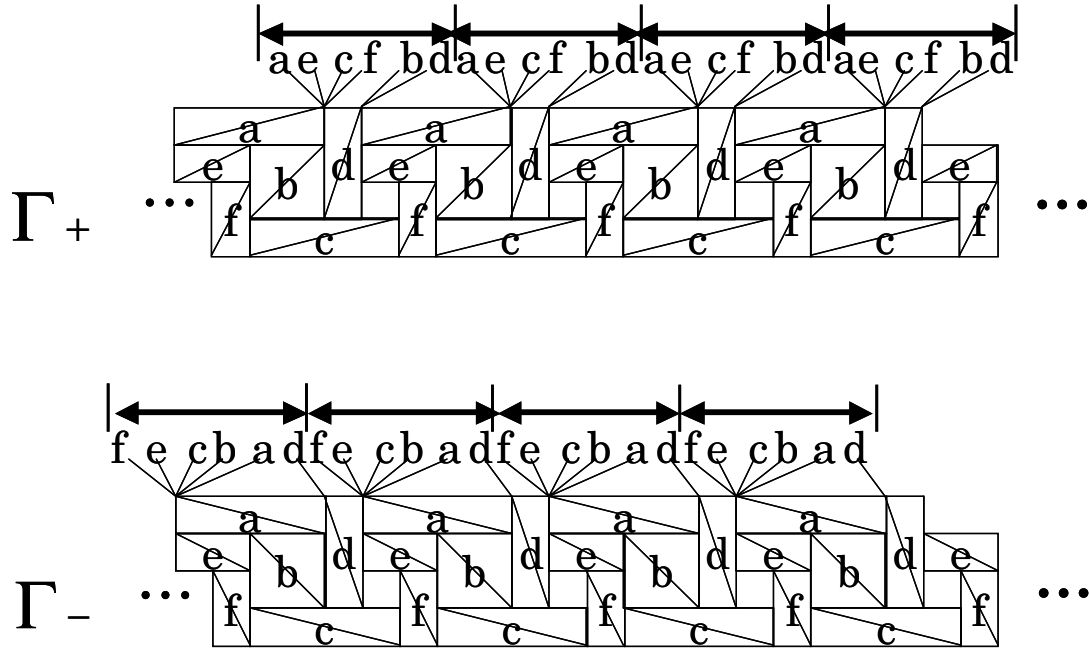


図 3.2: Gridding

3.2 問題点

次に、与えられたシーケンスペアから矩形の配置座標を導くこと（デコード）を考える．まず、シーケンスペア同様、相対位置関係は45度回転させた格子状のラインに Γ_+ 、 Γ_- をラベル付けする．すると、 Γ_+ 、 Γ_- ともに周期的に繰り返されるため、それぞれをラベル付けされたラインも水平方向に周期的に繰り返される．そのため、 Γ_+ における矩形名 a が、 Γ_- におけるどの a と対応付けられるのか一意に決定することができない(図3.3)という問題が出てくる．すなわち、矩形間の相対位置関係を一意に決定することができない．

3.3 提案手法

以上のような、周期的繰り返しとなっているシーケンスペアでは相対位置関係を一意に決定できない問題を解決するために、本研究では、 Γ_+ 、 Γ_- に加え、第三のシーケンス Γ_0 。

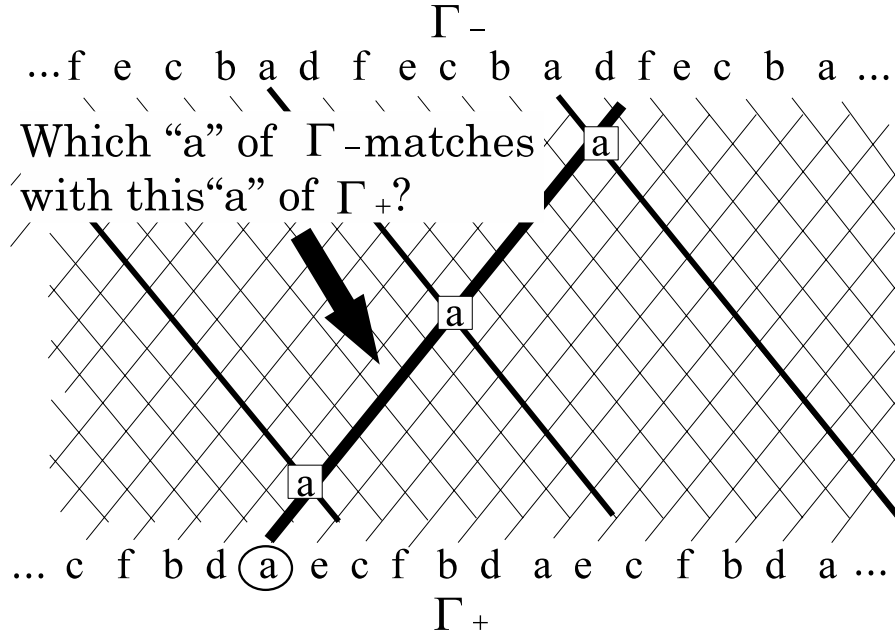


図 3.3: 相対位置関係

を導入する．また， Γ_+ ， Γ_- ， Γ_s から構成される解表現手法をシーケンスペアにならい，シーケンストリプル (Sequence Triple) と呼ぶ．

三つのシーケンスはそれぞれ次のように定義する．

Γ_+ : 周期的繰り返し配置では，グリッディングにより抽出される Positive step-line は周期的に繰り返す順列となるが，シーケンストリプルでは， Γ_+ はそのうちの一周期分の順列とする．

Γ_- : Γ_+ 同様，周期的繰り返し配置では，グリッディングにより抽出される Negative step-line は周期的に繰り返す順列となるが，シーケンストリプルでは， Γ_- はそのうちの一周期分の順列とする．

Γ_s : Γ_- によって特定される Negative step-line に相当する矩形集合について，その矩形集合から抽出される Positive step-line の順列とする．

Γ_s について，図を用いて説明する．例として，図 3.1 の周期的繰り返し配置について考える．まず，この周期的繰り返し矩形配置に対しグリッディングを行うと， $\Gamma_+ = aecfbd$ ， $\Gamma_- = fecbad$ が抽出される．ここでは， $\Gamma_- = fecbad$ に対する Γ_s を考える． $\Gamma_- = fecbad$ によって特定される Negative step-line に相当する矩形集合は図 3.4 である．繰り返すが， Γ_s は Γ_- によって特定される Negative step-line に相当する矩形集合から抽出される Positive step-line の順列である．この図 3.4 における Positive step-line の順列は $aefbdc$ である．すなわち，この順列が $\Gamma_- = fecbad$ に対しての Γ_s であり， $\Gamma_s = aefbdc$ である．

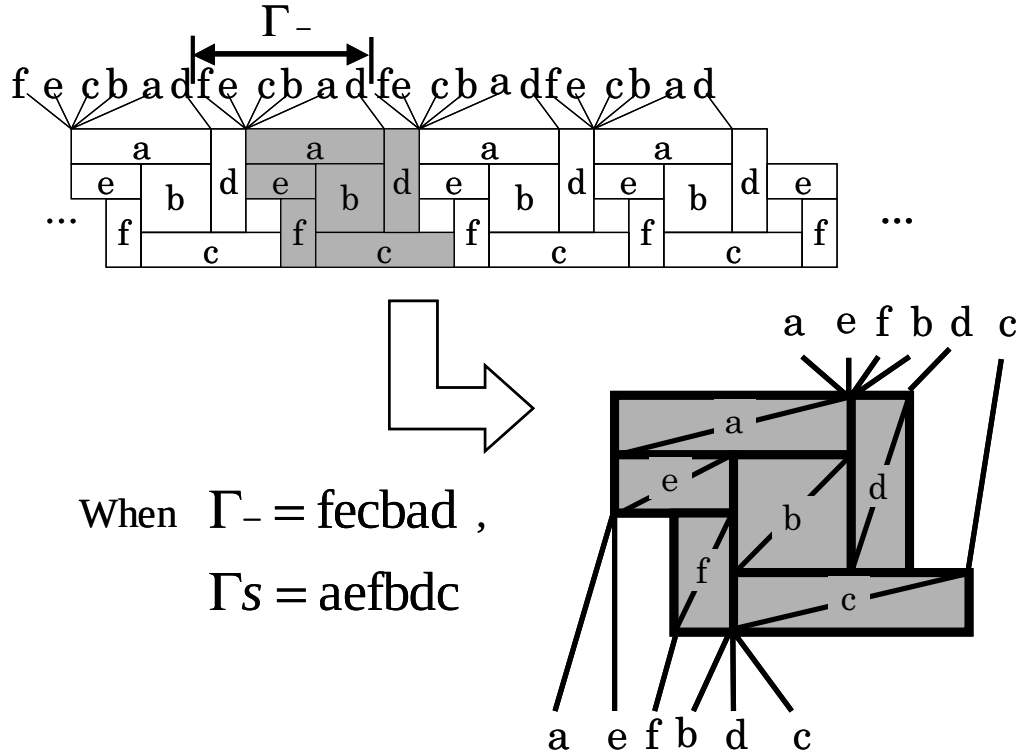


図 3.4: Γ_s

3.4 シーケンストリプルから矩形配置へ

シーケンストリプルによる相対位置関係の抽出について解説する．ここでは，例として $(\Gamma_+, \Gamma_-, \Gamma_s) = (fbdaec, fecbad, aefbdc)$ とする．

まず， Γ_+ ， Γ_- はシーケンスペア同様，45 度回転した格子状の右上がりのライン，右下がりのラインに繰り返しラベル付けする． Γ_s は Γ_- の順列により特定される Negative step-line に相当する矩形集合の Positive step-line であるので， Γ_- の一周期の中において Γ_s の順に Γ_+ が対応付けられる．このシーケンストリプルの例では， Γ_- の *fecbad* という一周期においては， Γ_+ は Γ_s の順，すなわち *aefbdc* という順で Γ_- に対して対応付けられる．図示すると図 3.5 のようになる．

このように，シーケンストリプルを用いることで， Γ_+ と Γ_- だけでは不可能であった Γ_+ と Γ_- の対応付けが可能となった．よって相対位置関係を一意に決定できる (図 3.5)．矩形間の左右関係，上下関係も抽出でき，図 3.7，図 3.8 のようになる．

水平方向に繰り返される矩形配置であるため，左右関係においては，同一周期内に配置

$$\Gamma_+ = fbdaec \quad \Gamma_- = fecbad \quad \Gamma_s = aefbdc$$

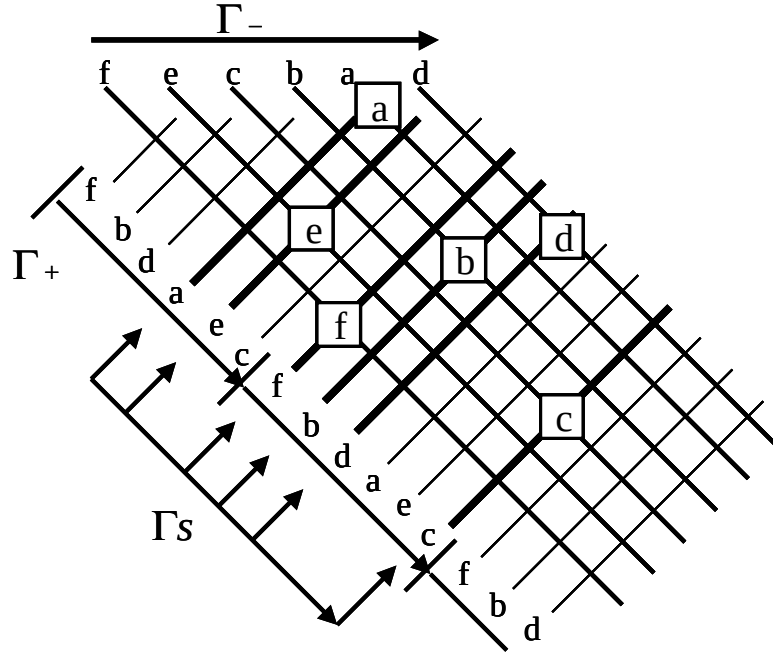


図 3.5: Γ_s

されている矩形同士が左右関係となるだけではなく，異なる周期に配置されている矩形同士が左右関係となる．また，左右関係は水平方向へループする．一方，上下関係についても，異なる周期の矩形と上下関係になる場合もあるが，垂直方向へは繰り返しとならないので，ループにはならない．

矩形の配置座標は，シーケンスペア同様，左右関係を水平制約グラフ，上下関係を垂直制約グラフとして計算する．

左右関係は水平方向へ繰り返しとなるため，水平制約グラフでは，座標変数を Γ_- において同一周期にある矩形の x 座標とする． $G_H(V, E)$ は次のように構成される．

V : 矩形名をラベルとして持つ m 個の頂点

E : 左右関係にある二つの矩形 a ，矩形 b 間を接続する辺 (a, b)

辺の重み: 同一周期内で左右関係である矩形 a ，矩形 b 間の辺 (a, b) の重みは矩形 a の幅，矩形 a と矩形 a が属する周期の 1 つ先 (座標平面上で右に位置する) 矩形 b 間の辺 (a, b) の重みは矩形 a の幅から周期幅 L を減じたもの

水平方向へ繰り返す配置という特性上，隣り合う周期に属する矩形と左右関係となる矩形が必ず存在する．そのような異なる周期内で左右関係となる頂点間を結ぶ辺からは周期

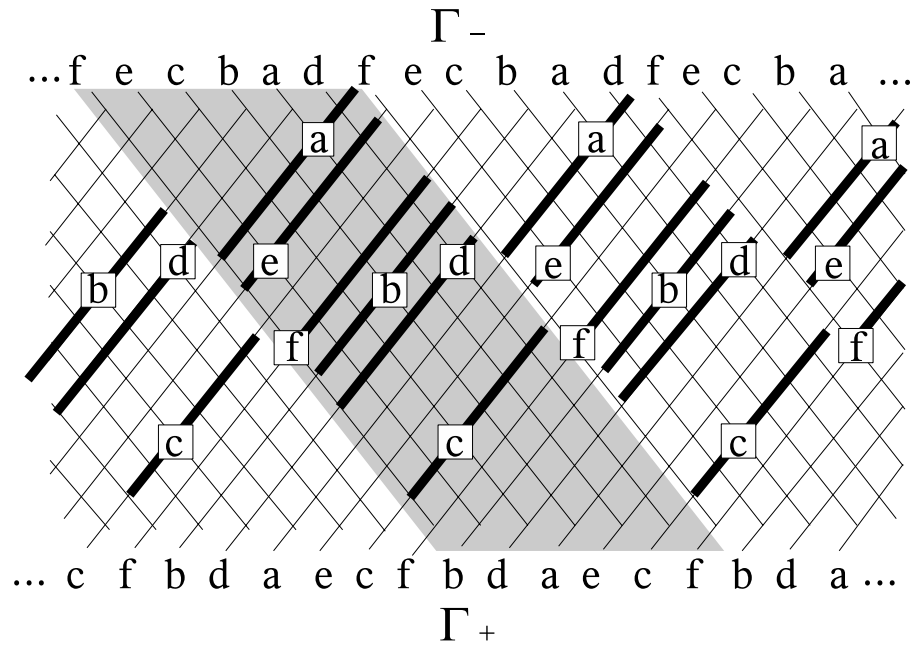


図 3.6: 相対位置関係

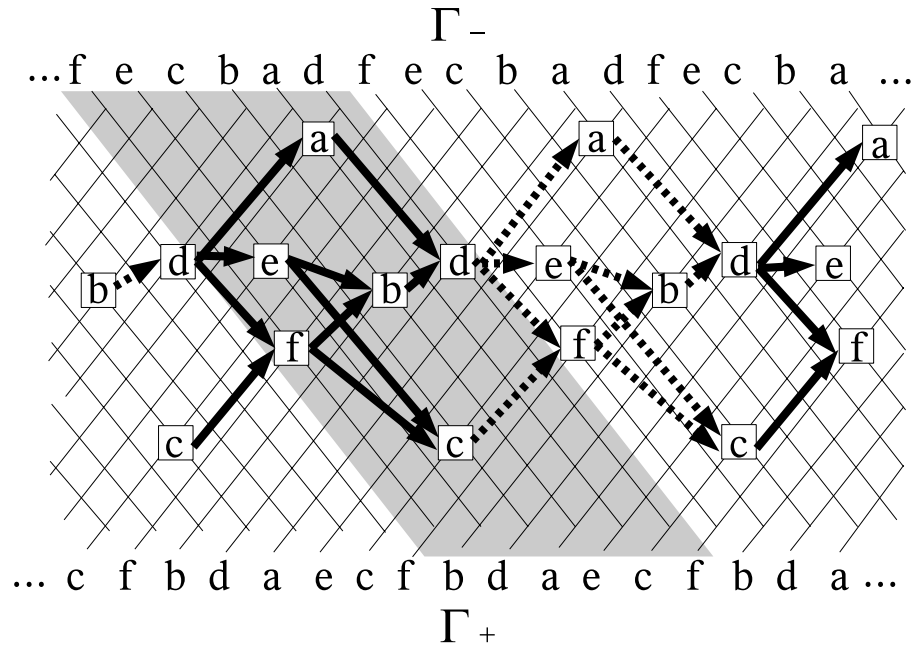


図 3.7: 左右関係

幅 L を減算する．また，シーケンスペアの水平制約グラフでは頂点 source, sink を設けたが，シーケンストリプル の水平制約グラフは，上記の理由から閉路を含むグラフとなるた

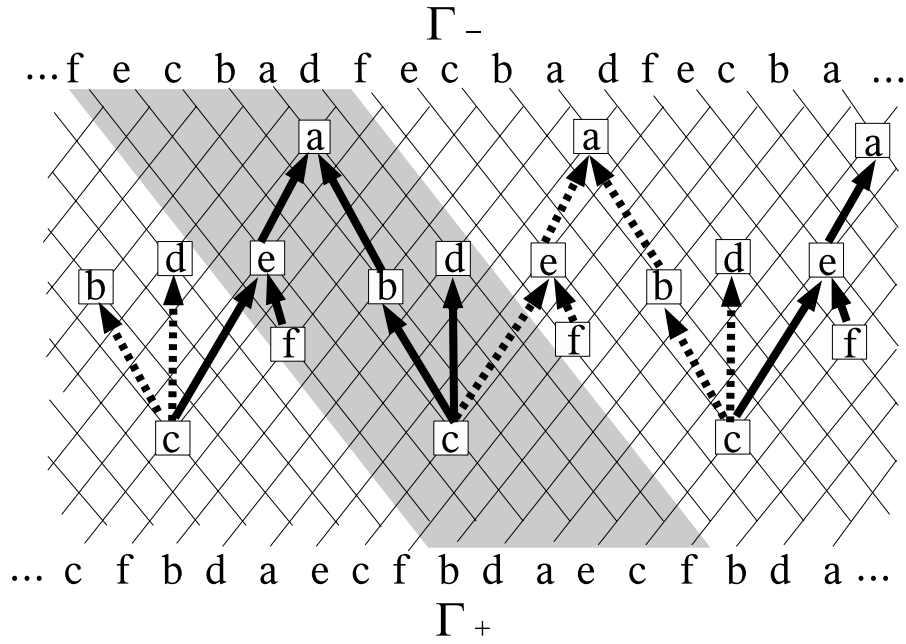


図 3.8: 上下関係

め, source, sink を設けない. 任意の頂点を一つ選び基準点とする. 図 3.7 の左右関係は, 図 3.9 のようにグラフ表現できる.

各頂点に相当する矩形の座標は, 基準点とした頂点からの最長経路長である. また, 閉路を含むグラフであるため, 正しく計算できるためには閉路を構成するすべての辺の重みの和が非正である必要がある. また, この“閉路を構成するすべての辺の重みの和が非正”となる最小の L は最小の繰り返し周期幅である.

一方, 上下関係は繰り返しが無いので, シーケンスペアと同じようにグラフを構成する. また, $G_V(V, E)$ は次のように構成される.

V : source s , sink t , 矩形名をラベルとして持つ m 個の頂点

E : 各矩形 a について辺 (s, a) , 辺 (a, t) , 上下関係である二つの矩形 a , 矩形 b 間を接続する辺 (a, b)

辺の重み: s に接続される辺 (s, a) の重みは 0, s 以外から出ている辺 (a, b) の重みは矩形 a の高さ

図 3.8 の上下関係は図 3.10 のようにグラフとして表現できる.

各頂点に相当する矩形の配置 y 座標は s からの最長経路長である. また, 矩形配置領域の高さは s から t までの最長経路長である.

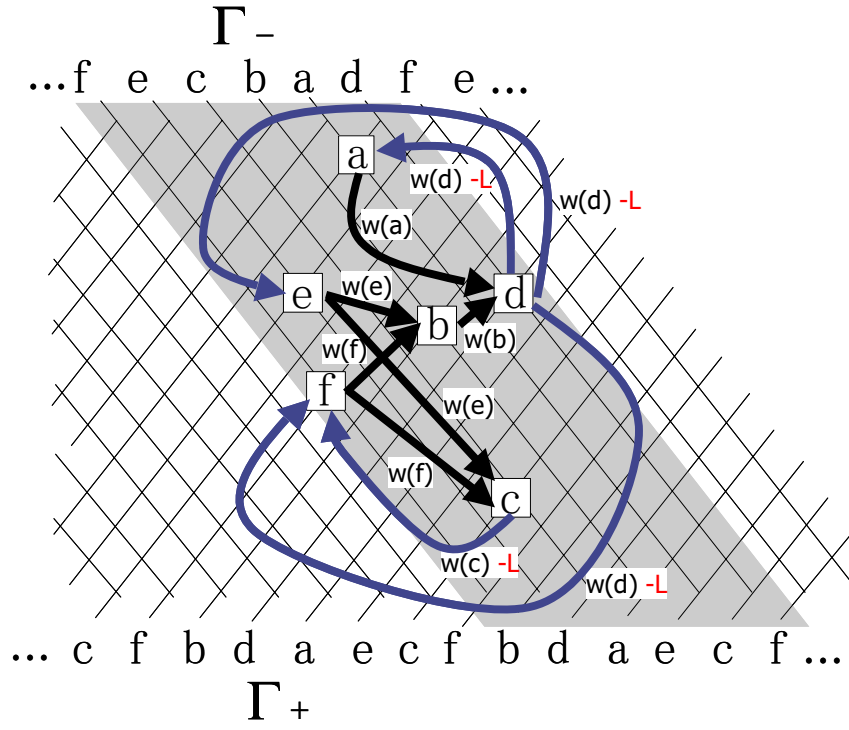


図 3.9: 水平制約グラフ

例として, $(\Gamma_+, \Gamma_-, \Gamma_s) = (fbdaec, fecbad, aefbdc)$ について, 各矩形の幅を $w(a) = 4$, $w(b) = 2$, $w(c) = 4$, $w(d) = 1$, $w(e) = 2$, $w(f) = 1$, 高さを $h(a) = 1$, $h(b) = 2$, $h(c) = 1$, $h(d) = 3$, $h(e) = 1$, $h(f) = 2$, 周期幅 $L = 5$ とした場合, 図 3.11 の矩形配置が導かれる.

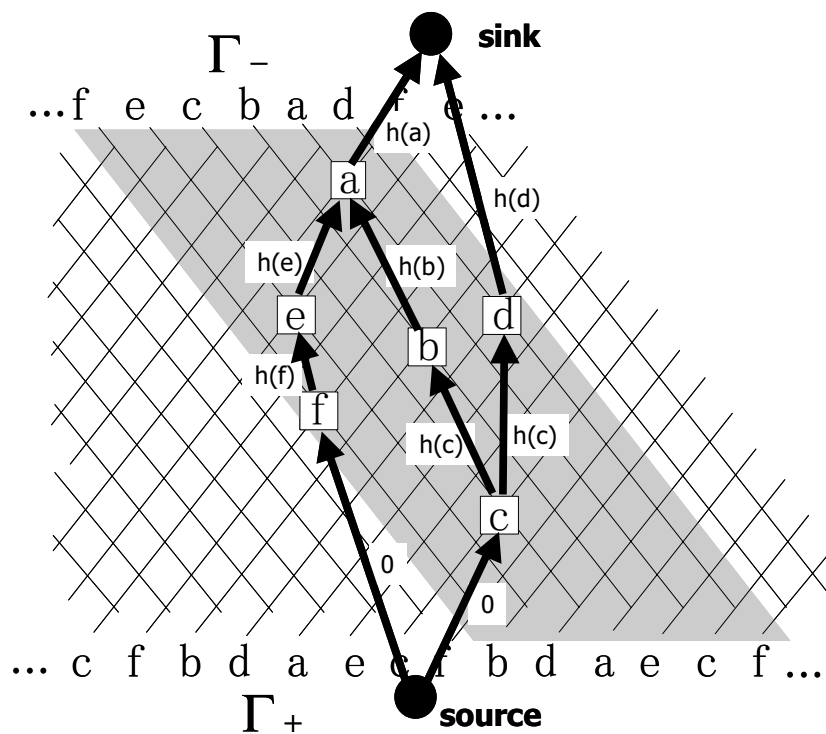


図 3.10: 垂直制約グラフ

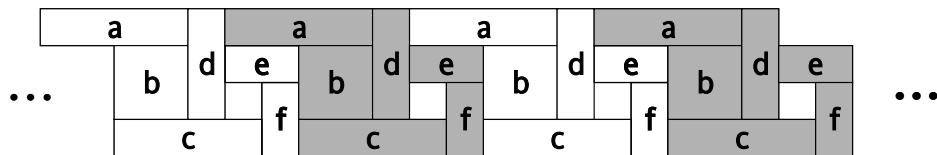


図 3.11: 導かれた矩形配置

3.5 補足

Γ_s は, Γ_- の一周期において Γ_+ を Γ_- に対応づける順序を規定しているが, 本研究では, Γ_+ と Γ_- は, Γ_s の順に最も近い矩形と対応付けている. 例えば, $\Gamma_s = aefbdc$ である場合, まず矩形 a を Γ_- と対応付け, 次に対応付けられた a の右へ最も近い距離に位置する矩形 e を対応付け, さらに, その対応付けられた e の右へ最も近い距離に位置する矩形 f を対応付け... というように対応付けを行う.

そこで, 上記のように対応付けられた各矩形から最も近い位置の矩形を順に対応付けるのではなく, Γ_s の順序を維持したまま, 一周期分を跳ばして対応付けると相対位置関係や実際の矩形配置がどのようなようになるか検証する.

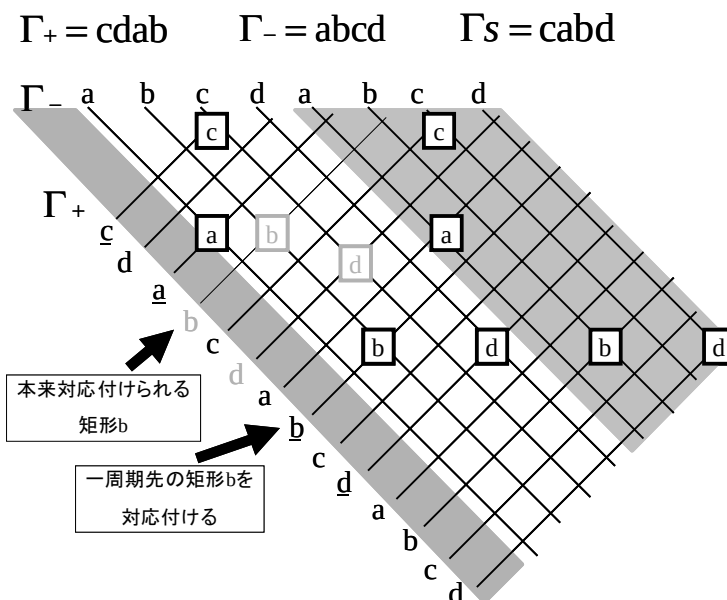


図 3.12: 一周期跳ばした対応付け

図 3.12 について, この例では, 矩形 a まで通常通り対応付けを行い, そのあと一周期分跳ばし, 続けて矩形 b から順に対応付けを行う. すると, 図 3.13 のような相対位置関係となる. この図から分かるように, 一周期分跳ばした矩形 a と矩形 b の間は, 一周期に渡って矩形が存在しない空白となる.

ここで, 一周期跳ばした後の矩形集合を, その矩形集合の中の順序を変更することなく並べる. さらに, それに続けて, 同様に一周期跳ばす前の矩形集合を並べ, Γ_- の並べ替えを行う. つまり, 一周期の空白の後に現れる矩形を Γ_- の前方に, 空白の前に現れる矩形を Γ_- の後方に配置する. この例では, $\Gamma_- = bdac$ というように並び替えられる (図 3.14).

次に, 一周期分の空白を圧縮する. すると図 3.15 のような相対位置関係を得られる. こ

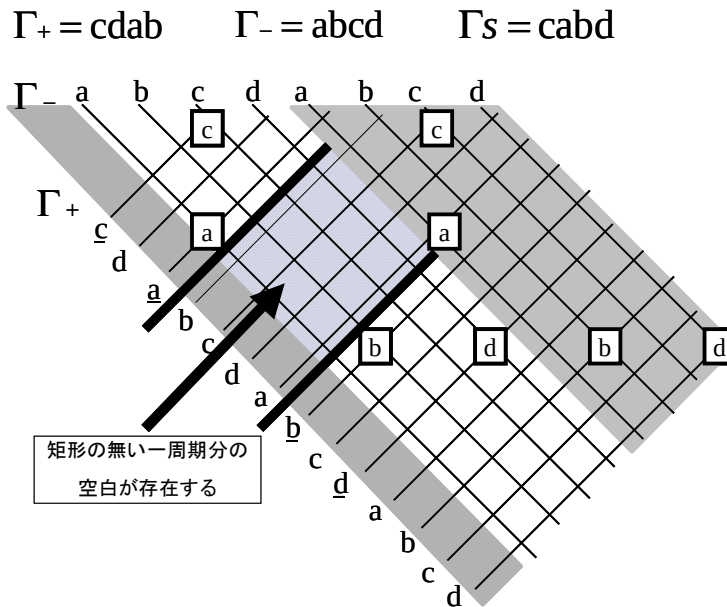


図 3.13: 一周期の空白

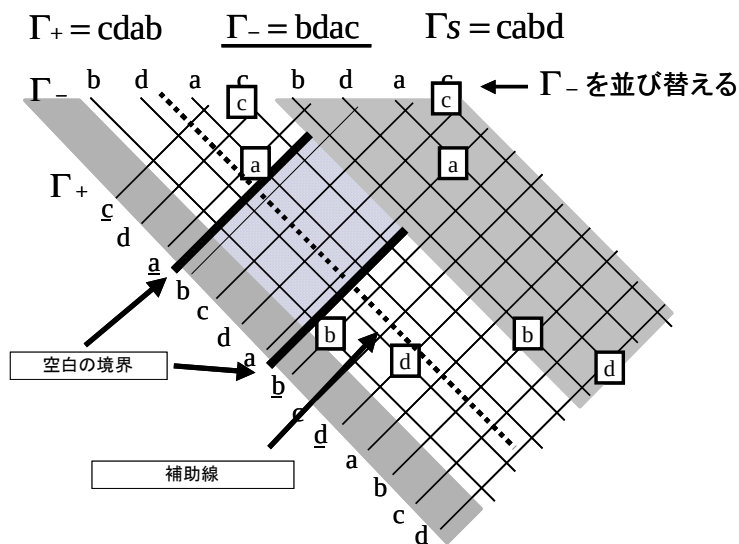


図 3.14: Γ_- の並び替え

の図より，境界線と補助線で形作られる十字の上に位置する矩形集合における相対位置関係は，当初の Γ_- (つまり図 3.13) から抽出される相対位置関係と変化していないことが分かる．また，十字の下に位置する矩形集合についても同様である．

周期幅を決めるのは，クリティカルパスとなる矩形の並びであるが，クリティカルパ

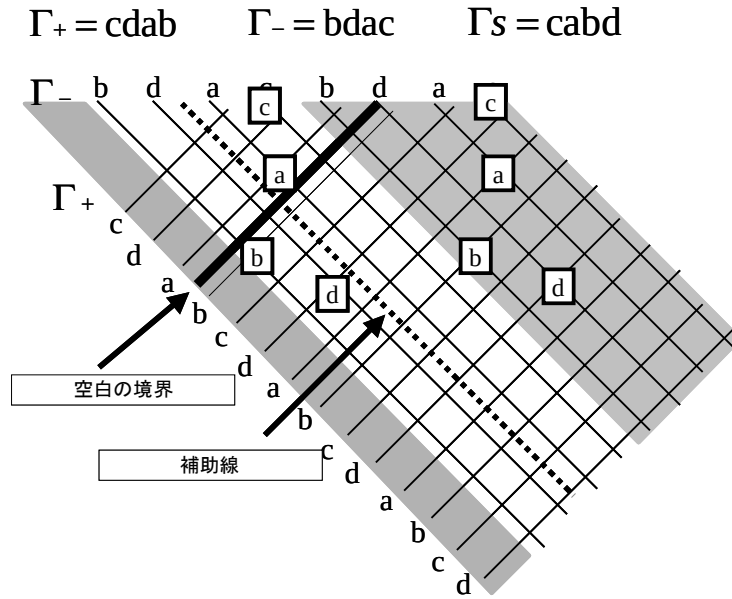


図 3.15: 空白の圧縮

スは交差しないため、十字の上下の矩形集合のどちらかの左右関係だけで最小の周期幅 L が決定される。上下それぞれの矩形集合の相対位置関係は変化していないため、跳ばしの無い場合と同じ左右関係が抽出される。よって、周期幅も跳ばしの有無に依らず同じである。また、上下関係は、跳ばしの無い場合の相対位置関係と同様となる（一周期における上下関係は変化する場合もあるが、隣接する周期における矩形も含めた上下関係は変化していない）。

このように、跳ばしを考えた場合でも、跳ばしが無い場合と周期幅と高さは同じ、つまり解（周期幅 $\times$ 高さ）が等しい矩形配置を得る手続きが存在することが分かる。よって、 Γ_+ と Γ_- の対応付けを考えると、 Γ_s の跳ばしを考慮する必要はないと考えられる。

第4章 シーケンストリプルのアルゴリズム

本章では、シーケンストリプルから矩形配置座標を求めるアルゴリズムについて解説する。

4.1 シーケンストリプルによる矩形配置座標計算

前述したように、水平方向への繰り返し配置では、それぞれの矩形は x 軸方向に対し共通の周期幅 L ごとに配置される。一方、各矩形の y 座標は同一である。

ここで、繰り返し配置の 0 番目の“周期”における矩形 a の x 座標を $x(a)$ と定義する。“周期”とは、 Γ_- により特定される Negative step-line に相当する矩形集合であることを強調しておく。

4.2 45 度回転の格子における矩形のアドレス

まず、一周期分の矩形配置について、45 度回転の格子を考える。これは図 4.1 のような格子である。この格子における矩形配置点をアドレスと呼び、 a のアドレス $(p_+(a), p_-(a))$ を以下のように定義する。

$p_+(a)$: Γ_+ の繰り返しにおける矩形 a の位置

$p_-(a)$: Γ_- における矩形 a の位置 (Γ_- の左端から矩形 a までの距離)

例として、 $\Gamma_+ = fbdaec$, $\Gamma_- = fecbad$, $\Gamma_s = aefbdc$ について考える (図 4.1)。すると、 $(p_+(a), p_-(a)) = (4, 5)$, $(p_+(b), p_-(b)) = (8, 4)$, $(p_+(c), p_-(c)) = (12, 3)$, $(p_+(d), p_-(d)) = (9, 6)$, $\dots$ となる。

このように、0 番目の周期における矩形 a のアドレスを定義すると、 k 番目の周期における各矩形 a のアドレスを

$$(p_+(a), p_-(a)) + k(n, n)$$

と表現できる。 n とは矩形の数である。

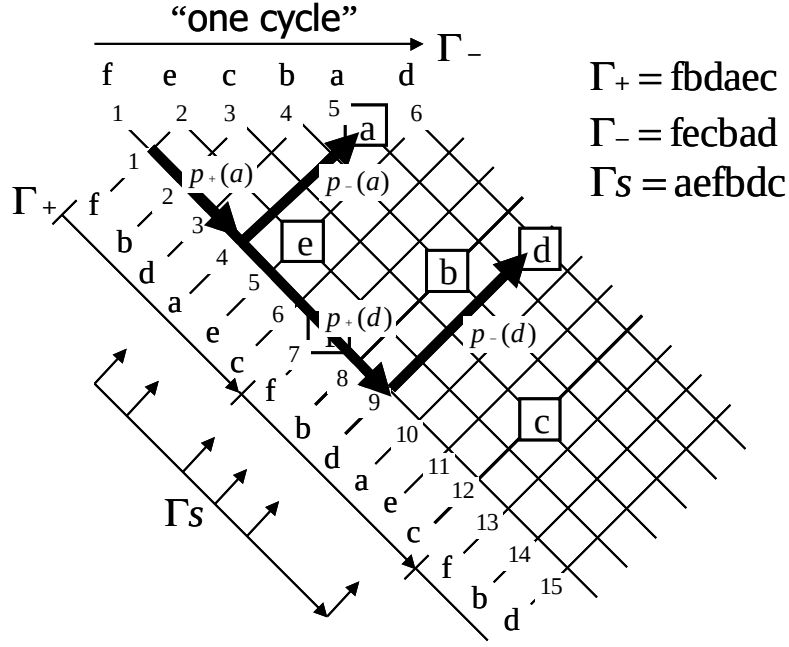


図 4.1: アドレス

4.3 矩形間の左右関係

まず，矩形間の左右関係の特定について解説する．45 度回転の格子におけるアドレスについて，次の補題が成り立つ．

補題 1: 以下の式が成り立つ時かつその時に限り， l 番目の周期にある矩形 b は k 番目の周期にある矩形 a の右に位置する．

$$\left. \begin{aligned} (p_+(b) + l \cdot n) - (p_+(a) + k \cdot n) &< 0 \\ (p_-(b) + l \cdot n) - (p_-(a) + k \cdot n) &< 0 \end{aligned} \right\} \quad (4.1)$$

式 4.1 により，1 番目の周期にある矩形 b が k 番目の周期にある矩形 a の右に位置する場合における $x(a)$ ， $x(b)$ について，次の制約式を導出できる．

$$x(b) + l \cdot L \geq x(a) + k \cdot L + w(a) \quad (4.2)$$

さらに，式 4.2 を変形すると，

$$x(b) \geq x(a) + (k - l) \cdot L + w(a) \quad (4.3)$$

式 4.3 で，最も重要な変数は $(k - l)$ である． $(k - l)$ は以下の補題によって与えられる．

補題 2: 以下のように $q_+(a, b)$, $q_-(a, b)$ を定義する.

$$\begin{aligned} q_+(a, b) &= \left\lfloor \frac{p_+(a, b) - p_-(a, b)}{n} \right\rfloor \\ q_-(a, b) &= \left\lfloor \frac{p_+(a, b) - p_-(a, b)}{n} \right\rfloor \end{aligned} \quad (4.4)$$

ここで $(k - l)$ は, 次の式によって求めることができる.

$$(k - l) = \min\{q_+(a, b), q_-(a, b)\} \quad (4.5)$$

4.4 矩形間の上下関係

次に, 矩形間の上下関係の特定について解説する.

左右関係と同様に, 格子上におけるアドレスを用いると, 次の補題を導くことができる.

補題 3: 以下の式を満たす整数 m が存在する時かつその時に限り, 矩形 b は矩形 a の上に位置する.

$$\left. \begin{aligned} (p_+(b) - (p_+(a) + m \cdot n) &< 0 \\ (p_-(b) - (p_-(a) + m \cdot n) &> 0 \end{aligned} \right\} \quad (4.6)$$

また, 式 4.6 により次の補題も導出される.

補題 4: 補題 3 より次の補題を導出できる.

次の不等式が成り立つ場合, 矩形 b は矩形 a の上側に位置する.

$$q_-(a, b) \geq q_+(a, b) + 1 \quad (4.7)$$

次の不等式が成り立つ場合, 矩形 b は矩形 a の下側に位置する.

$$q_+(a, b) \geq q_-(a, b) + 1 \quad (4.8)$$

次の不等式が成り立つ場合, 矩形 b は矩形 a の間には上下関係は無い.

$$q_-(a, b) = q_+(a, b) \quad (4.9)$$

4.5 全体アルゴリズム

シーケンストリプルから矩形配置座標を算出するアルゴリズムについて解説する.

手順 1: Γ_+ , Γ_- から各矩形のアドレス

手順 2: (補題 2) を利用して左右関係を抽出と $(k - l)$ の算出.

手順 3: (補題 4) を利用して上下関係を抽出.

手順 4: 水平制約グラフ, 垂直制約グラフを生成.

手順 5: 水平制約グラフを探索する関数に対して, L を引数として渡す. 引数として与えられた L を利用して各辺の重みを計算し, (手順 6) を行う. L がすべての閉路の辺の重みの和が非正となる最小の値 (= 最小の繰り返し周期幅) となるような値になるまで二分法による L の探索を行う. また, L の初期値は, 各矩形の $\min\{\text{幅}, \text{高さ}\}$ のうち, 最も大きい値である.

手順 6: 頂点を一つ選び基準点とする. 基準点から各頂点までの最長経路長がその頂点の x 座標である.

まず, 同一周期の左右関係における, $-L$ の重みが付加されていないすべての辺について, 全点間の最長経路長を計算する. 次に, $-L$ の重みが付加されている全ての辺について, L を含めた辺の重みが非負となる L を計算する. そのような L のうち最も大きい値が最小の周期幅となる.

手順 7: 垂直制約グラフに対し, source から各頂点までの最長経路長を計算, 各矩形の y 座標を算出する.

4.6 シーケンストリプルの計算量

本研究で開発したシーケンストリプルのでコードの計算量は, 以下の補題を利用して求めることができる.

補題 1: critical path は真に交差することはない.

補題 1 より, critical path は, $-L$ の重みが付加される辺を唯一つだけ含む.

まず, 同一周期の左右関係における, $-L$ の重みが付加されていないすべての辺について, 全点間の最長経路長を計算する. この計算には Floyd のアルゴリズムを利用し, 矩形数を n とする時, 計算量は $O(n^3)$ である. 次に, $-L$ の重みが付加されている全ての辺について, L を含めた辺の重みが非負となる L を計算する. この計算量は高々 $O(n^2)$ である.

よって, デコード全体の計算量は $O(n^3)$ である. また, 解空間の大きさは $(n!)^3$ である.

第5章 実験

5.1 実験方法

シーケンストリプルにより与えられる解空間を Simulated Annealing 法を用いて探索し、最適な矩形配置を求めるプログラムを実装して実験を行った。プログラムは C を用いて記述し、実験には Intel Pentium 4 (2.2GHz), Linux(kernel 2.6.0) ベースのシステム上で行った。

実験の概要については以下のとおりである。

- 入力： 矩形の幅，高さ（乱数を用いて無作為に生成）
- 出力： 各矩形の配置座標
- 最適化の目標： 繰り返し周期幅 L $\times$ 矩形配置領域の高さ H の最小化

隣接解は，生成の度に生成方法を乱数により以下から一つ選択し，生成する。

1. Γ_+ 内で任意の二つの矩形名の入れ替え
2. Γ_- 内で任意の二つの矩形名の入れ替え
3. Γ_s 内で任意の二つの矩形名の入れ替え
4. Γ_+ と Γ_s 内で同じペアの二つの矩形名を同時に入れ替え
5. Γ_- と Γ_s 内で同じペアの二つの矩形名を同時に入れ替え
6. 任意の矩形を回転（幅と高さを入れ替え）

上記の内容で，Simulated Annealing の温度パラメータや矩形数を変更して実験を行った。また，シーケンストリプルとの比較のためにシーケンスペアについても同様の最適化を行うプログラムを実装した。シーケンスペアについての実験の概要は以下の通りである。

- 入力： 矩形の幅，高さ（乱数を用いて無作為に生成）

- 出力： 各矩形の配置座標
- 最適化の目標： 矩形配置領域の幅 W $\times$ 矩形配置領域の高さ H の最小化

隣接解は、生成の度に生成方法を乱数により以下から一つ選択し、生成する．

1. Γ_+ 内で任意の二つの矩形名の入れ替え
2. Γ_- 内で任意の二つの矩形名の入れ替え
3. Γ_+ と Γ_- 内で同じペアの二つの矩形名を同時に入れ替え
4. 任意の矩形を回転（幅と高さを入れ替え）

5.2 実験結果

実験 1 一つ目の実験として、以下の内容で実験を行った．

- **矩形数**: 30
- **SA パラメータ**
開始温度: 10000, 終了温度: 0.01, 減少係数: 0.98
一温度辺りの繰り返しループ数: 500
- **各矩形の面積の総和**: 22222

結果として得られた矩形配置は以下のとおりである． Γ_- の周期ごとに色分けしたものが図 5.1 であり、周期幅を x 座標で規定して色分けしたものが図 5.2 である．配置領域の面積（周期幅 $L \times$ 配置領域の高さ H ）は、23100（ 60×385 ）であり、面積の占有率（各矩形の面積の総和 $\div$ 配置領域の面積）は、0.961991 である．また、SA の実行時間は、1119 秒であった．

縦に細長い配置が計算されていることがわかる．

SA について、面積の変化と accept 回数の変化をグラフにしたものが図 5.3、図 5.4 である．なお、面積の変化についてのグラフは、一温度における最も良い値をプロットしている．

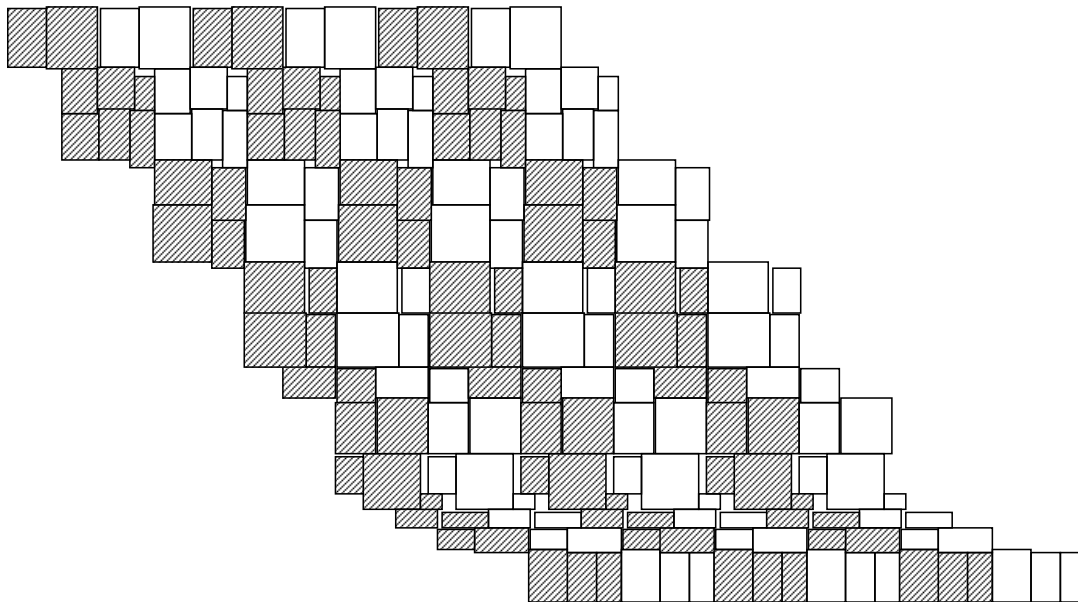


図 5.1: 実験1 配置結果 (Γ_- の周期で色分け)

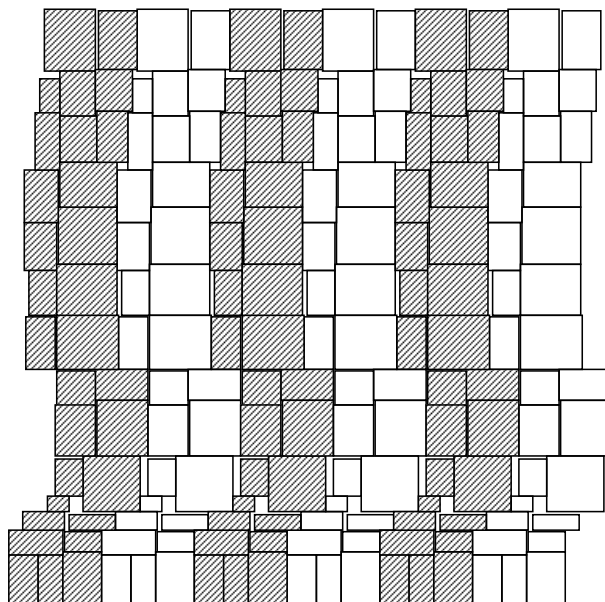


図 5.2: 実験1 配置結果 (x 座標で周期を規定して色分け)

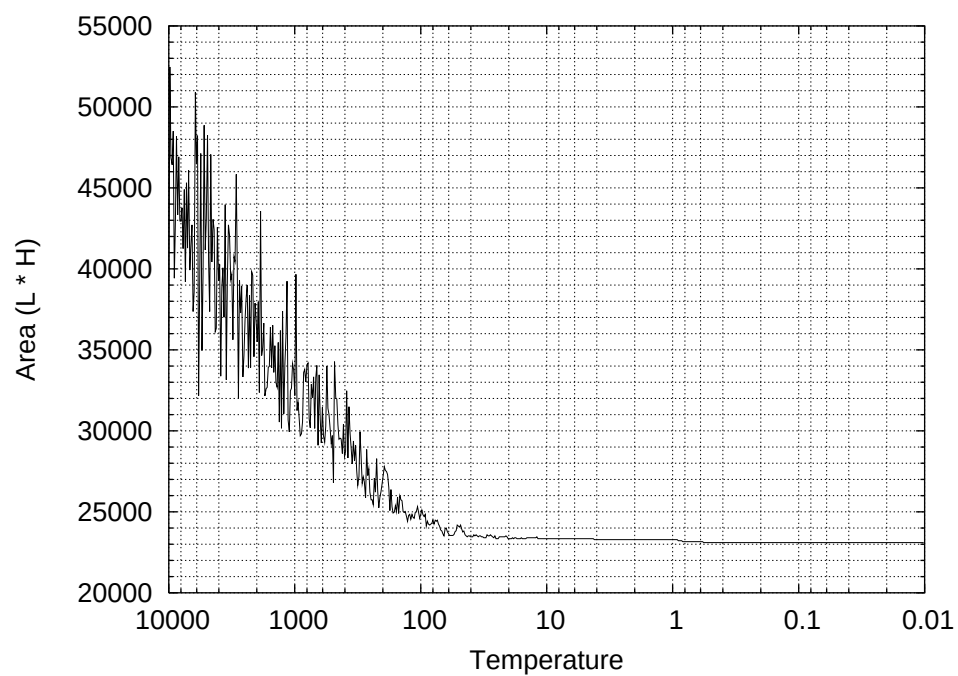


図 5.3: 実験 1 面積の変化

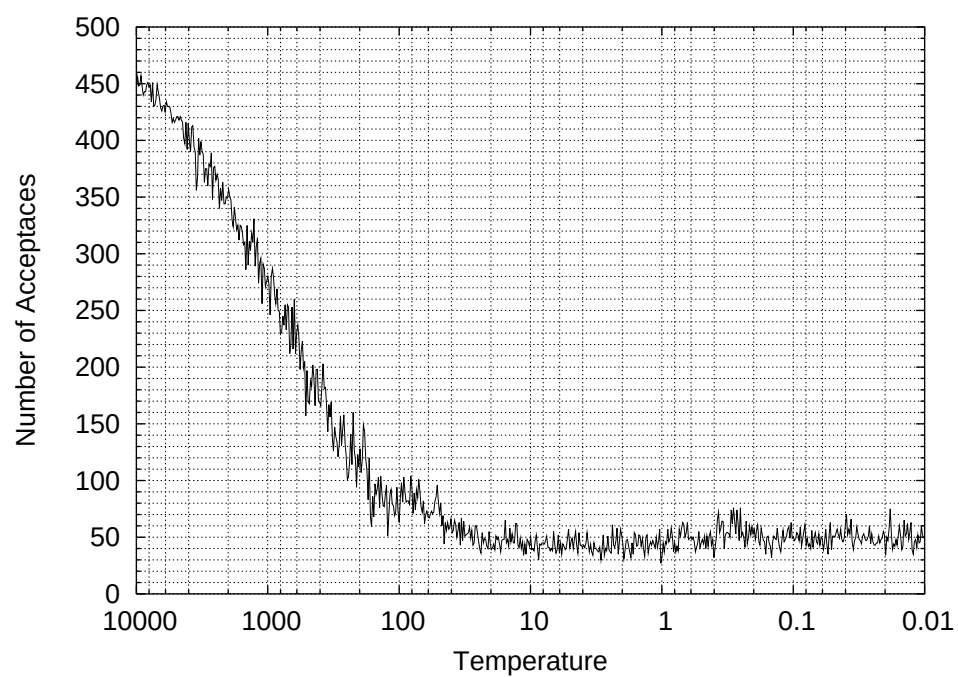


図 5.4: 実験 1 accept 数の変化

実験2 次に、シーケンスペアとの比較のため、実験1と同一の矩形データ、矩形数、SAパラメータを用いて実験を行った。なお、シーケンスペアでは、矩形配置面積は、(配置領域の幅 W × 配置領域の高さ H) である。

- **矩形数:** 30

- **SA パラメータ**

開始温度: 10000, 終了温度: 0.01, 減少係数: 0.98

一温度辺りの繰り返しループ数: 500

- **各矩形の面積の総和:** 22222

結果、以下の配置が得られた。配置領域の面積は、22932 (156×147) であり、面積の占有率は、0.969039 である。また、SA の実行時間は、6 秒であった。

SA について、面積の変化と accept 回数の変化をグラフにしたものが図 5.6, 図 5.7 である。

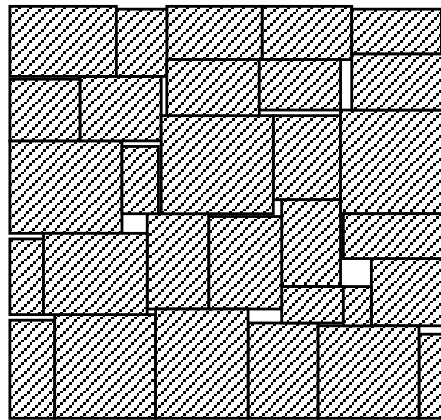


図 5.5: 実験2 配置結果

シーケンストリプルでの SA 結果とシーケンスペアでの SA 結果を比較すると、まず、シーケンストリプルでの SA の実行時間はシーケンスペアのそれに比べ、非常に低速であることがわかる。また、シーケンスペアの配置結果が正方形に近い形となるのに対し、シーケンストリプルの配置結果は非常に縦長になる傾向が見られる。一方、面積の占有率は、どちらも大きな違いはない。ただ、シーケンストリプルはシーケンスペアに比べ解空間が大きいので、同じ温度パラメータではシーケンスペアより良い解を得るのは難しいかもしれない。そこで、実験3では一温度あたりのループ回数を 5000 として実験を行った。矩形データやその他のパラメータは同じである。

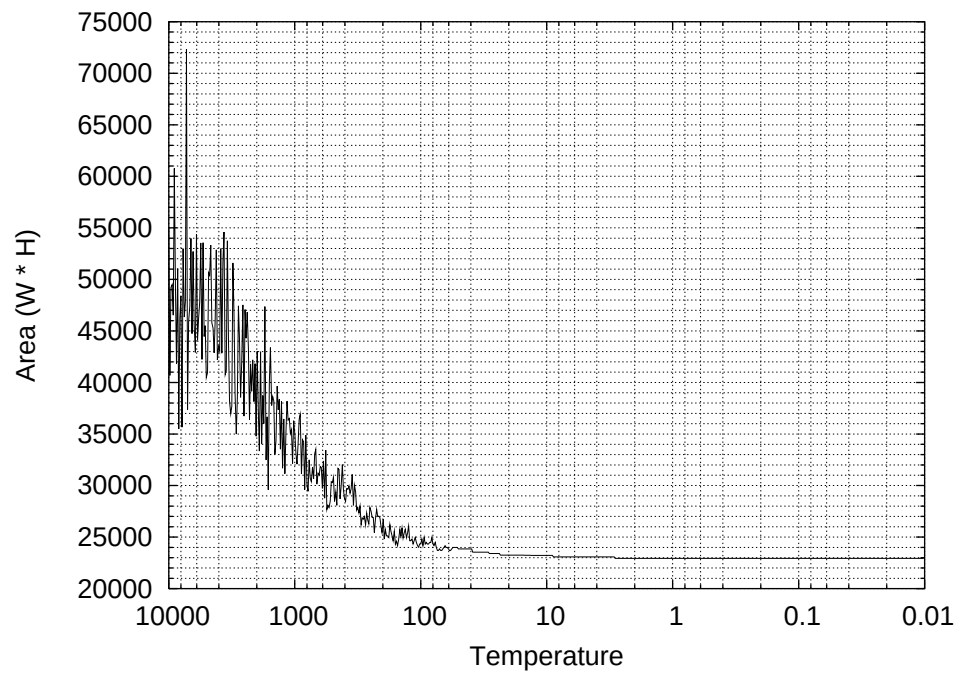


図 5.6: 実験 2 面積の変化

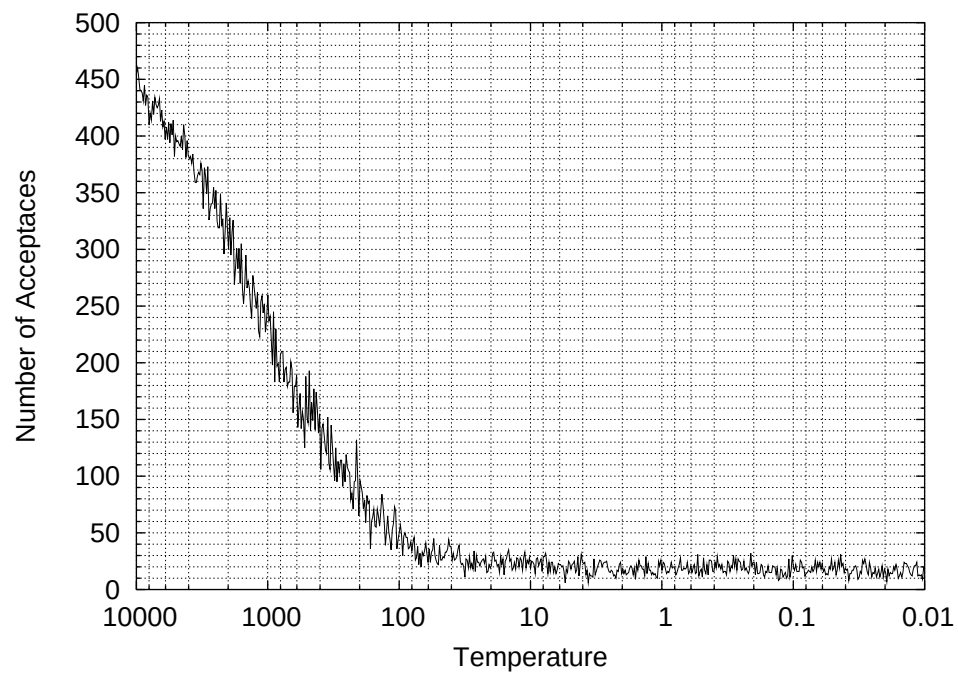


図 5.7: 実験 2 accept 数の変化

実験 3

- 矩形数: 30

- SA パラメータ

開始温度: 10000, 終了温度: 0.01, 減少係数: 0.98

一温度辺りの繰り返しループ数: 5000

- 各矩形の面積の総和: 22222

結果, 以下の配置が得られた. 配置領域の面積は, 23048 (67×344) であり, 面積の占有率は, 0.964162 である. また, SA の実行時間は, 11287 秒であった.

SA について, 面積の変化と accept 回数の変化をグラフにしたものが図 5.10, 図 5.11 である.

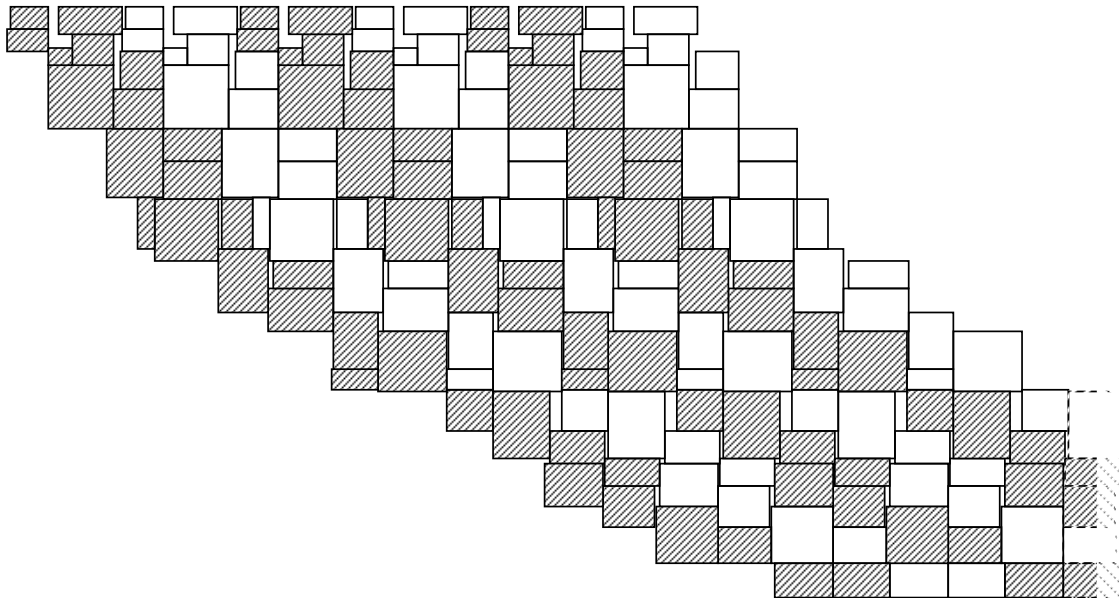


図 5.8: 実験 3 配置結果 (Γ の周期で色分け)

温度パラメータを変更し, 一温度あたりのループ回数を 10 倍したが, 実行時間もおおよそ 10 倍となった. しかし, 解は多少良くなった程度である. 実験 1 の温度スケジュールでは十分な探索ができないというわけでは無いと考えられる. 矩形配置はやはり縦長となる傾向がある. これはシーケンストリプルにそのようになる特性があるものと考えられる.

次に, 縦横比が 1 に近づく傾向のあるシーケンスペアと比較するために, シーケンスペアから得られる矩形配置の周期幅と高さの比率が 1 に近づくように指定して実験 4 を行った.

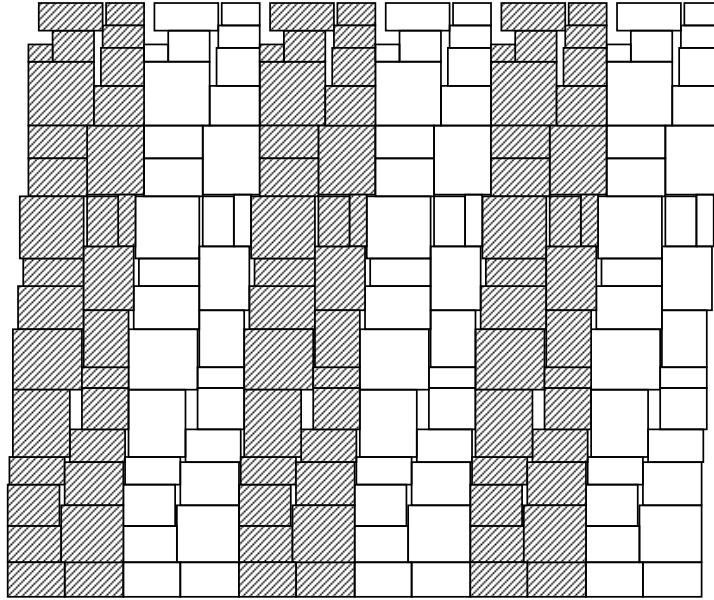


図 5.9: 実験 3 配置結果 (x 座標で周期を規定して色分け)

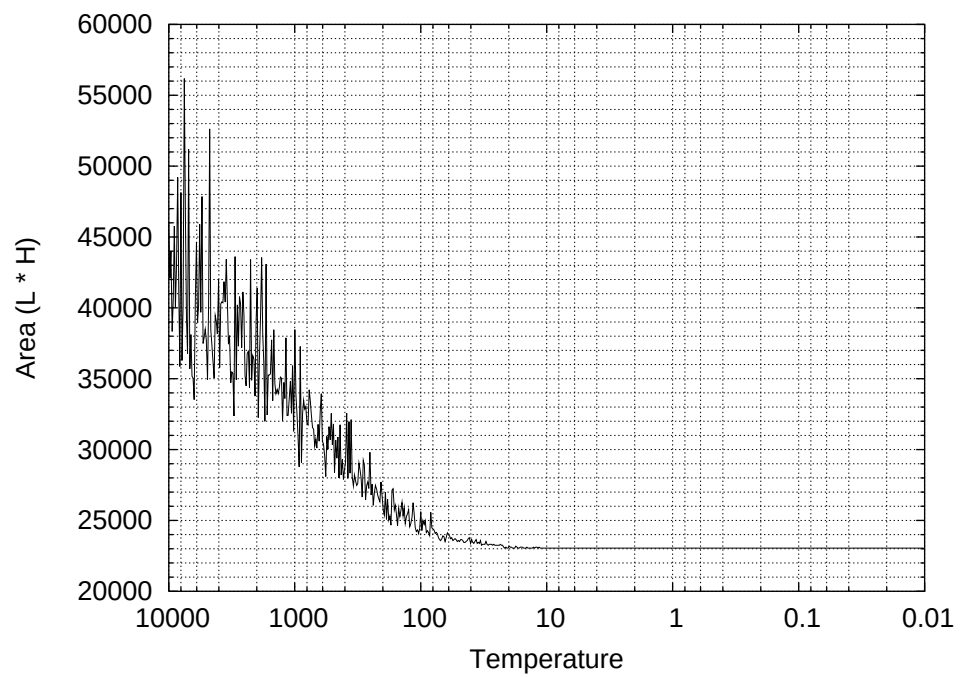


図 5.10: 実験 3 面積の変化

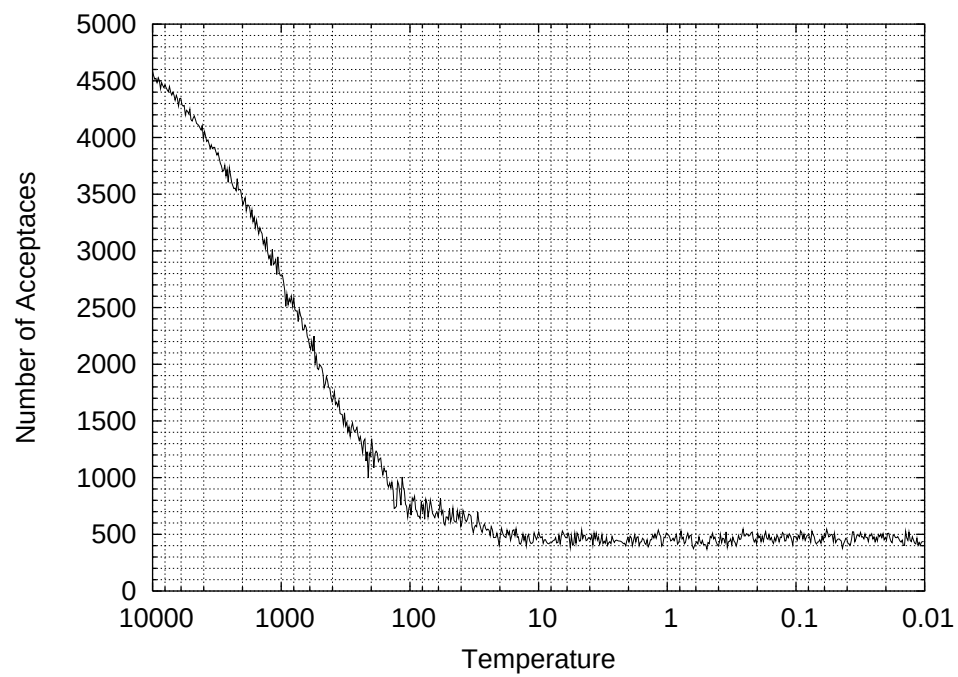


図 5.11: 実験3 accept 数の変化

実験 4

- 矩形数: 30
- SA パラメータ
開始温度: 10000, 終了温度: 0.01, 減少係数: 0.98
一温度辺りの繰り返しループ数: 5000
- 各矩形の面積の総和: 22222
- 縦横比: 1

結果, 以下の配置が得られた. 配置領域の面積は, 23716 (154×154) であり, 面積の占有率は, 0.937005 である. また, SA の実行時間は, 1240 秒であった.

SA について, 面積の変化と accept 回数の変化をグラフにしたものが図 5.14, 図 5.15 である.

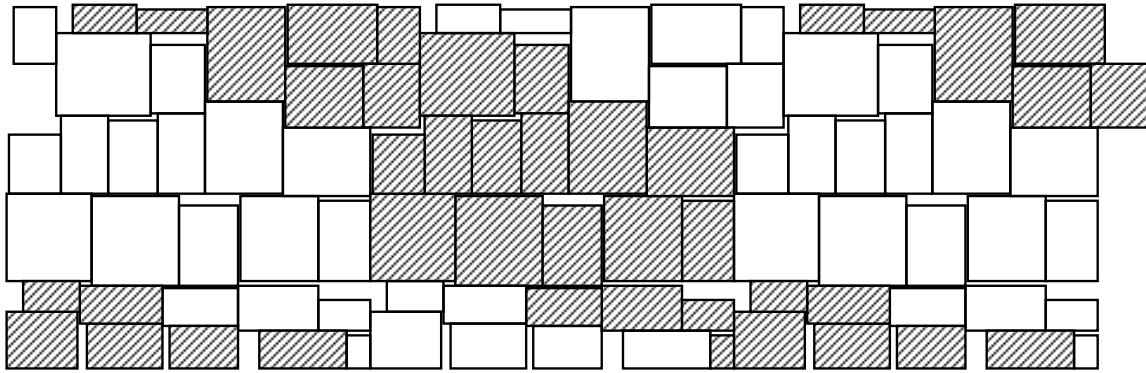


図 5.12: 実験 4 配置結果 (Γ_+ の周期で色分け)

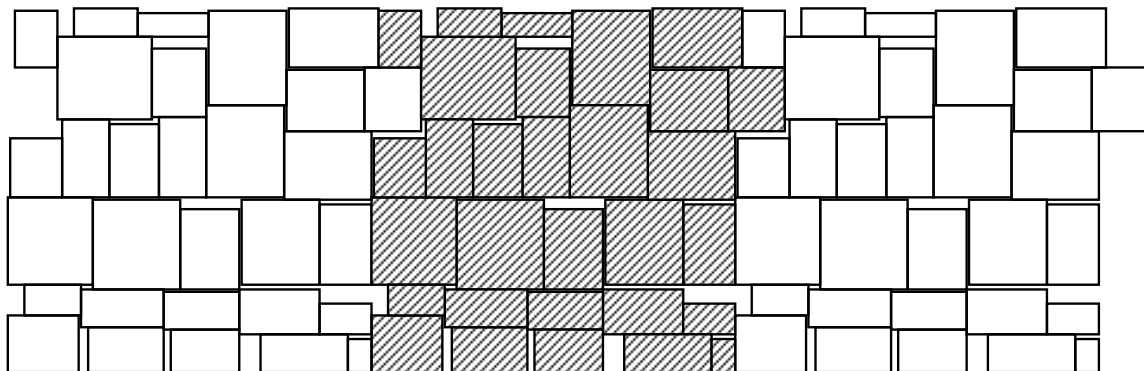


図 5.13: 実験 4 配置結果 (x 座標で周期を規定して色分け)

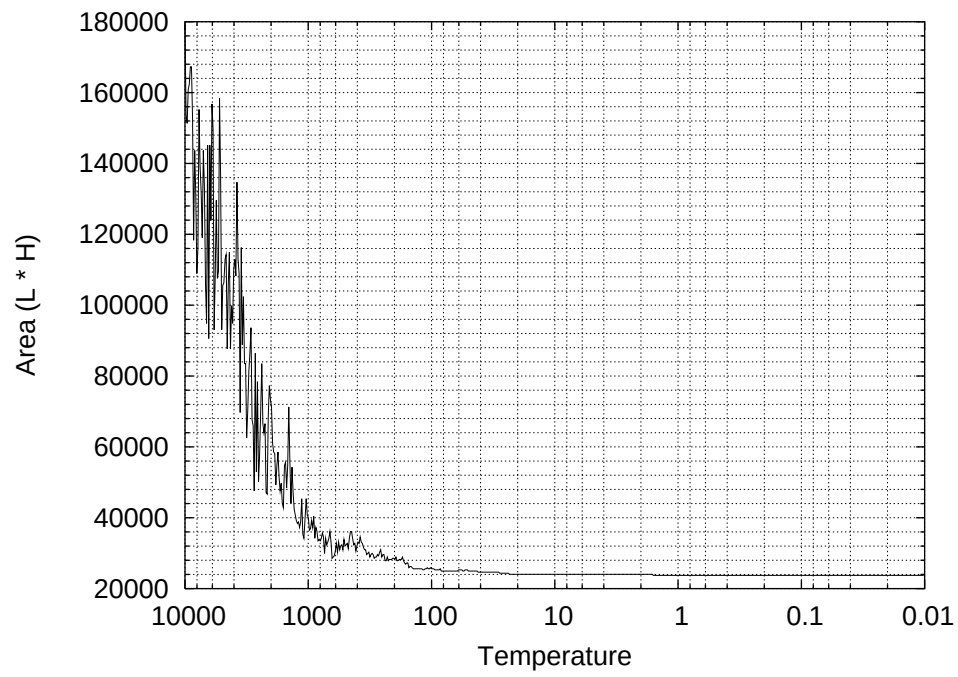


図 5.14: 実験 4 面積の変化

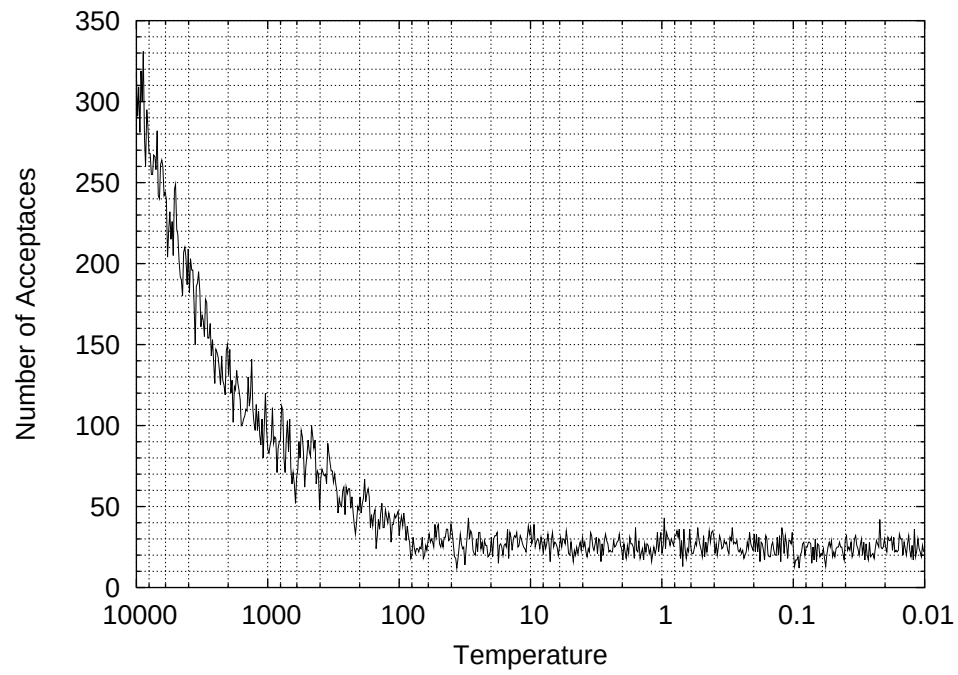


図 5.15: 実験 4 accept 数の変化

縦横比はシーケンスペアの配置に近づいたが、解は縦横比を指定せずに解探索を行ったものよりも悪くなってしまった。シーケンストリプルの特性を無視した縦横比指定は、解に良い影響を与えないと考えられる。また、比率を1に近づけたところで、シーケンスペアのように正方形に近づくというわけではない（シーケンスペアのように配置領域の周囲が凹凸があまり無い直線にはならない）。

次に、矩形数を多くした場合、計算される矩形配置にどのような変化があるか確認するために、矩形数を50にして実験を行った。

実験 5

- 矩形数: 50
- SA パラメータ
開始温度: 10000, 終了温度: 0.01, 減少係数: 0.98
一温度辺りの繰り返しループ数: 2500
- 各矩形の面積の総和: 29357

なお、矩形を増やすと、解空間も指数関数的に大きくなるため、一温度あたりのループ回数を増やして実験を行った。

結果、以下の配置が得られた。配置領域の面積は、30674 (49×626) であり、面積の占有率は、0.957065 である。また、SA の実行時間は、20582 秒であった。著しく縦長な配置である。得られた矩形配置の縦横比が約 6.4:1 であった実験 1 と比べても、図 5.17 の縦横比は約 12.8:1 と、大幅に縦長になっていることがわかる。これは、矩形が多くなったことによってさらに縦長になる傾向が強まったためであると考えられる。

SA について、面積の変化と accept 回数の変化をグラフにしたものが図 5.18, 図 5.19 である。

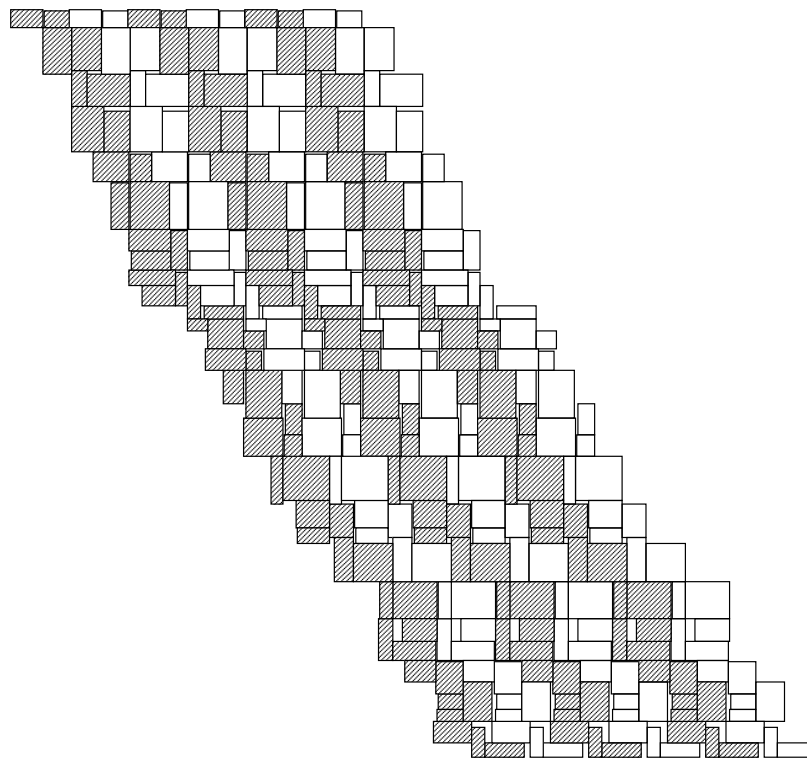


図 5.16: 実験 5 配置結果 (x 座標で周期を規定して色分け)

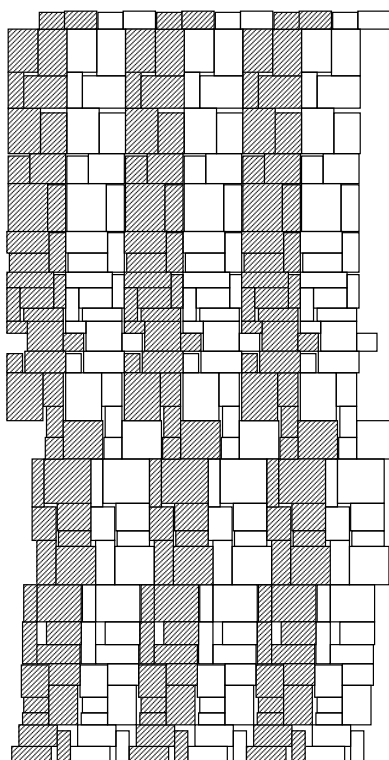


図 5.17: 実験 5 配置結果 (x 座標で周期を規定して色分け)

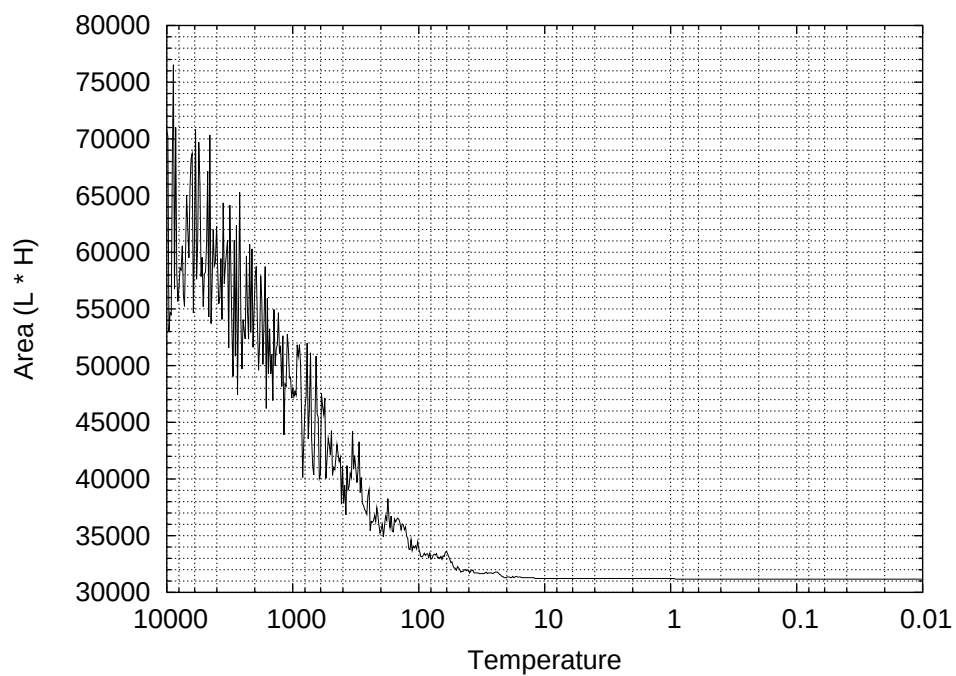


図 5.18: 実験 5 面積の変化

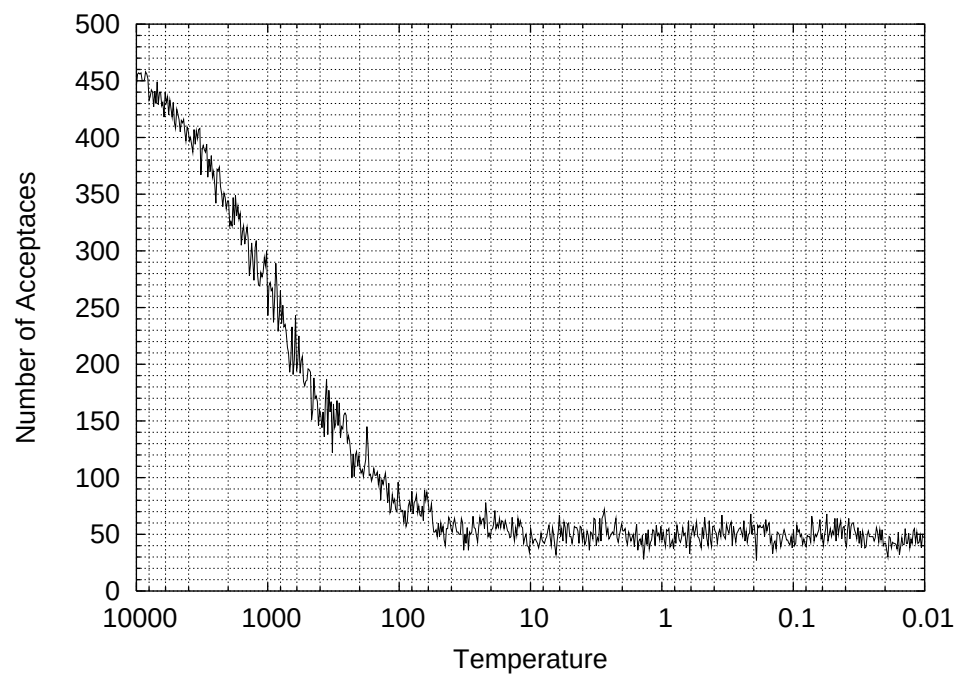


図 5.19: 実験 5 accept 数の変化

5.3 考察

以上のように、実験を行った結果、以下の考察を得た。

1. シーケンストリプルにより定義される解空間を SA 法で探索して配置最適化を行う場合、最適化される配置には縦長になる傾向がある。
2. また、縦長になる傾向は、矩形数に比例して強くなる。

配置が縦長となる傾向は、ランダムに生成されたシーケンストリプルにおいて、二つの矩形が上下関係になる確率と、左右関係になる確率に偏りがあることに由来する。矩形数が n であるとき、任意の二つの矩形が左右関係にある確率は、以下の式で求めることができる。

$$\frac{1}{2(n-1)n} \times \sum_{i=1}^{n-1} \sum_{j=1}^{n-i} \frac{1}{j!} \quad (5.1)$$

2 から 50 までの矩形数に対し、式 5.1 によって矩形間の相対位置関係が左右関係となる確率を求めたものが、以下のグラフである。

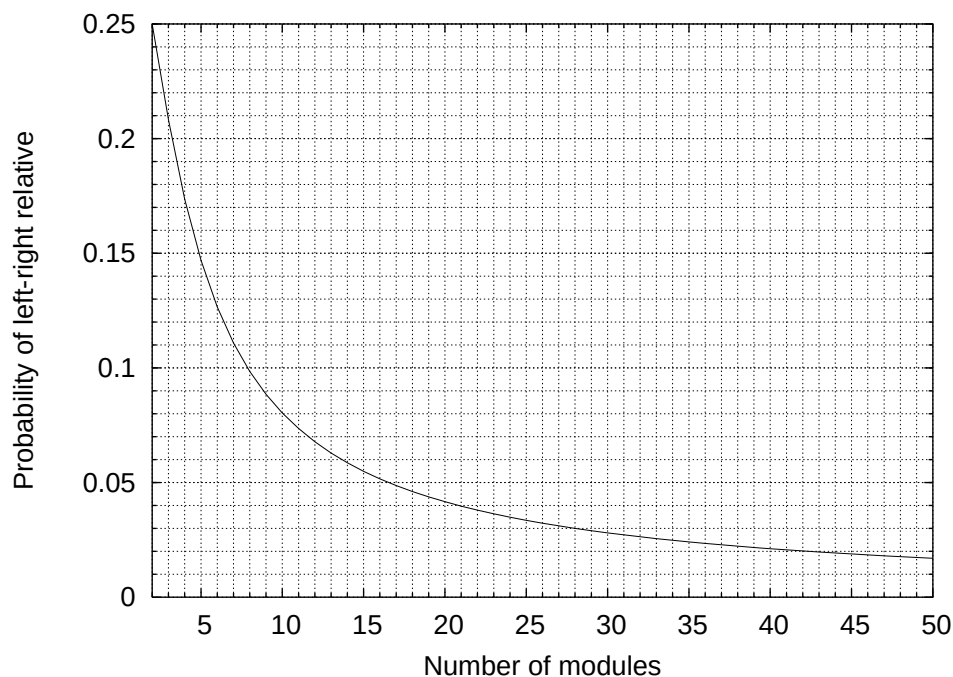


図 5.20: 任意の二つの矩形が左右関係となる確率

第6章 結論

6.1 まとめ

本研究では、矩形集合の周期的繰り返し配置に対する解表現手法として、シーケンストリプルを提案した。これは、繰り返しでない通常の矩形配置の解表現手法であるシーケンスペアに順列を一つ加えたものである。順列を一つ加えたことにより、繰り返し周期の一周期内での矩形間の相対位置関係だけでなく、異なる周期においても矩形間の相対位置関係を規定することが可能となった。

本研究で開発したシーケンストリプルのデコードの計算量は $O(n^3)$ であり、解空間の大きさは $(n!)^3$ である。

また、シーケンストリプルにより定義される解空間を Simulated Annealing 法を用いて探索し、最適な矩形配置を計算する実験を行った。

6.2 今後の課題

今後の課題は、

1. 最適な隣接解生成ルールを選択
2. シーケンストリプルのデコードアルゴリズムの高速化
3. 計算量と解空間の大きさについて、より優れたコーディングシステムの開発が、挙げられる。

謝辞

本研究を進めるにあたり、終始ご指導、ご教授頂きました金子峰雄教授に、心より深く感謝いたします。また、多くのご助言、ご協力を頂きました大橋功治氏はじめ研究室の皆様に、深く感謝と御礼を申し上げます。

最後に、金子研究室の益々のご発展をお祈り致します。

参考文献

- [1] H.Murata, K.Fujiyoshi, S.Nakatake, “VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair”, IEEE Trans. on Computer-Aided Design, 15(12): pp.1518-1524, Dec.1996.
- [2] S.Nakatake, H.Murata, K.Fujiyoshi and Y.Kajitani, “Module Packing Based on the BSG-Structure and IC Layout Applications”, IEEE Trans.CAD, pp.519-530(1998).
- [3] P.N.Guo, C.K.Chen, and T.Yoshimura, “An O-tree representation of non-slicing floorplans and its applications”, Proc.DAC-99, pp.268-273(1998).