## **JAIST Repository**

https://dspace.jaist.ac.jp/

Title	2次元トーラス空間における矩形配置のコード表現と最 適化
Author(s)	柴田,貴之
Citation	
Issue Date	2008-03
Туре	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/4353
Rights	
Description	Supervisor:金子峰雄,情報科学研究科,修士



Japan Advanced Institute of Science and Technology

## 修士論文

## 2次元トーラス空間における 矩形配置のコード表現と最適化

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

## 柴田 貴之

2008年3月

## 修士論文

## 2次元トーラス空間における 矩形配置のコード表現と最適化

#### 指導教官 金子峰雄教授

# 審查委員主查 金子峰雄 教授 審查委員 宮地充子 教授 審查委員 浅野哲夫 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

## 610042 柴田 貴之

提出年月: 2008年2月

Copyright © 2008 by Shibata Takayuki

矩形配置問題は,与えられた矩形集合をある座標空間で,矩形同士が互いに重なることな く配置する問題であり,LSIレイアウトなど様々な応用を持つ最適化問題である.このよ うな矩形配置における配置面積の面積最小化などの問題は,NP困難に属することが知ら れている.

本研究では,2次元トーラス空間における矩形配置問題を取り扱う.2次元トーラス空間は,2つの座標軸が回帰構造を持つ空間であり,矩形集合が有限回,または無限回,周期的に繰り返す空間と捉えることもできる.このような,矩形が周期的に繰り返す矩形配置は,同一の回路構造が繰り返し配置されるメモリ,セルとスイッチボックスが規則的に並ぶ FPGA などの LSI レイアウト問題や,トーラス型ネットワーク構造を持つマルチプロセッサシステムにタスクを割り当てるスケジューリング問題などにも応用することができる.

2次元平面上の矩形配置問題の解表現手法として、シーケンスペア(Sequence-pair[2])が 提案されている.シーケンスペアは、矩形名を要素とする2つの順列 $\Gamma_+$ 、 $\Gamma_-$ の順序対の 形をとっており、各順列には、全ての矩形名が厳密に1つずつ含まれる.このシーケンス ペアは、各矩形の配置座標の情報を持つわけではなく、矩形同士の相対位置関係のみを規 定している.従って実際の配置座標は、この規定された相対位置関係から計算にて求める ことになる.また、配置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 シーケンスペアの考え方は、その後、3次元空間内の直方体パッキングのためのシーケン ストリプル(Sequence-triple[3])やシーケンスクインタプル(Sequence-quintuple[3])、また、 2次元平面上で1つの座標軸のみ回帰構造を持つ一方向繰り返しパッキングのためのシー ケンストリプル[4]、周期的に繰り返す配置の効率的な表現方法[5]などへ拡張されている.

本研究にて対象とする2次元トーラス空間では,通常の2次元平面上のパッキング手法 であるシーケンスペア,一方向繰り返しでのパッキング手法であるシーケンストリプル, 周期的に繰り返す配置の効率的な表現方法等をそのまま使うことはできない.これは,2 次元トーラス空間内の矩形配置から抽出した Г+, Г-から2方向繰り返し配置を導くこと を考えた場合,1周期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 関係を規定する必要があるため,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 題が生じるためである.

これらの問題を解決するため,先ず基準矩形を定め,それによって規定される1周期領 域と,そこから導かれる周期内矩形集合,拡大周期内の矩形集合を定義する.拡大周期内 矩形集合は,周期内矩形のそれぞれを1回づつ,また周期境界上の矩形のそれぞれを2回 づつ含んだものとなっている.提案する2次元トーラス空間内配置コードは,この拡大周 期内矩形集合の相対位置関係を,シーケンスペアと同じ考え方で表現するものである.提 案コードは,同一の矩形が少なくとも1回,高々2回(添字をつけて,各々を区別)現れ, 通常のシーケンスペアと区別し,以下では拡大シーケンスペア( $\Pi_+$ , $\Pi_-$ )と呼ぶ.拡大 シーケンスペアは,矩形数をnとしたとき, $\mathcal{O}((2n)!)^2$ のサイズの解空間を持ち,異なる 順列は異なる相対位置関係を表す.

本論文では,提案手法である拡大シーケンスペアにおいて,任意の矩形配置に対する対応するコードの存在性,及び有効なコードであるための矩形名の並びの性質,拡大シーケンスペアからのトーラス空間内配置計算法,SAによって拡大シーケンスペアにて規定される解空間を探索するための隣接解生成方法を明らかしている.次いで,拡大シーケンスペアにより定義される解空間を SA 法により探索を行うプログラムを実装し,矩形配置最適化の実験をおこなった.様々な入力に対して解探索をおこなったところ,繰り返し境界上に矩形が存在しない矩形配置を出力する結果となった.これは,拡大シーケンスペアにおいて特徴的な隣接解生成操作である分離操作(拡大シーケンスペア上で周期内矩形を周期境界上に変更する操作)にて,隣接解生成時に暫定解より面積効率が非常に悪くなることが多いためと予想され,この点を改善する,より適切な分離操作について考察した.

今後の課題として,拡大シーケンスペアの特徴的な隣接解生成操作である分離,再結合において,配置変化が少ない隣接解生成方法,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考案が挙げられる.

## 目次

第1章	はじめに											
第2章	準備	3										
2.1	シーケンスペア	3										
2.2	矩形配置からコードの抽出...............................	4										
2.3	コードから矩形配置の抽出	6										
2.4	1 方向繰り返し配置への拡張	9										
	2.4.1 シーケンストリプル	9										
	2.4.2 シーケンストリプルから矩形配置へのデコード	10										
	2.4.3 シーケンストリプルの問題点	11										
	2.4.4 周期的に繰り返す配置の効率的な表現方法	11										
	2.4.5 周期的に繰り返す配置の効率的な表現方法から矩形配置	13										
第3章	拡大シーケンスペア	14										
3.1	2 次元トーラス空間内の矩形配置	14										
3.2	2次元トーラス空間内の矩形配置における問題点	15										
33	提案する表現手法の概要	15										
3.4	短形配置からコードの抽出	17										
3.5	拡大シーケンスペアから矩形配置	17										
010	3.5.1 制約グラフ	18										
	3.5.2. 周期の計算	21										
		23										
3.6	実現可能なコード	24										
穷 / 辛	破坏毒	26										
<b>牙 ₩</b> 早		20										
4.1		20										
4.2	YY Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y	20										
	4.2.1 押八	21										
	4.2.2 主父探	28										
	4.2.3 分離	28										
	4.2.4 円結台	29										

第5章	実験と考察	30
5.1	実験方法....................................	30
5.2	実験 1:拡大シーケンスペアを用いた解探索	30
5.3	実験 2:周期境界上の矩形数を固定した解探索	49
5.4	実験 3:周期境界上の矩形を固定した解探索	52
第6章	まとめ	55
6.1	2次元トーラス空間内におけるコード表現と最適化	55
6.2	今後の課題	55

## 第1章 はじめに

矩形配置問題は,与えられた矩形集合をある座標空間で,矩形同士が互いに重なることなく配置する問題であり,LSIレイアウトなど様々な応用を持つ最適化問題である.このような矩形配置における配置面積の面積最小化などの問題は,NP困難に属することが知られている.

本研究では,2次元トーラス空間における矩形配置問題を取り扱う.2次元トーラス空間は,2つの座標軸が回帰構造を持つ空間であり,矩形集合が有限回,または無限回,周期的に繰り返す空間と捉えることもできる.このような,矩形が周期的に繰り返す矩形配置は,同一の回路構造が繰り返し配置されるメモリ,セルとスイッチボックスが規則的に並ぶ FPGA などの LSI レイアウト問題や,トーラス型ネットワーク構造を持つマルチプロセッサシステムにタスクを割り当てるスケジューリング問題などにも応用することができる.

2次元平面上の矩形配置問題の解表現手法として、シーケンスペア(Sequence-pair[2])が 提案されている.シーケンスペアは、矩形名を要素とする2つの順列 $\Gamma_+$ 、 $\Gamma_-$ の順序対の 形をとっており、各順列には、全ての矩形名が厳密に1つずつ含まれる.このシーケンス ペアは、各矩形の配置座標の情報を持つわけではなく、矩形同士の相対位置関係のみを規 定している.従って実際の配置座標は、この規定された相対位置関係から計算にて求める ことになる.また、配置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 シーケンスペアの考え方は、その後、3次元空間内の直方体パッキングのためのシーケン ストリプル(Sequence-triple[3])やシーケンスクインタプル(Sequence-quintuple[3])、また、 2次元平面上で1つの座標軸のみ回帰構造を持つ一方向繰り返しパッキングのためのシー ケンストリプル[4]、周期的に繰り返す配置の効率的な表現方法[5]などへ拡張されている.

本研究にて対象とする2次元トーラス空間では,通常の2次元平面上のパッキング手法 であるシーケンスペア,一方向繰り返しでのパッキング手法であるシーケンストリプル, 周期的に繰り返す配置の効率的な表現方法等をそのまま使うことはできない.これは,2 次元トーラス空間内の矩形配置から抽出した  $\Gamma_+$ ,  $\Gamma_-$  から2方向繰り返し配置を導くこと を考えた場合,1周期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 関係を規定する必要があるため,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 題が生じるためである.

これらの問題を解決するため,先ず基準矩形を定め,それによって規定される1周期領域と,そこから導かれる周期内矩形集合,拡大周期内の矩形集合を定義する.拡大周期内

矩形集合は,周期内矩形のそれぞれを1回づつ,また周期境界上の矩形のそれぞれを2回 づつ含んだものとなっている.提案する2次元トーラス空間内配置コードは,この拡大周 期内矩形集合の相対位置関係を,シーケンスペアと同じ考え方で表現するものである.提 案コードは,同一の矩形が少なくとも1回,高々2回(添字をつけて,各々を区別)現れ, 通常のシーケンスペアと区別し,以下では拡大シーケンスペア( $\Pi_+$ , $\Pi_-$ )と呼ぶ.拡大 シーケンスペアは,矩形数をnとしたとき, $\mathcal{O}((2n)!)^2$ のサイズの解空間を持ち,異なる 順列は異なる相対位置関係を表す.

本論文の構成は以下の通りである.第2章では,準備として繰り返しのない2次元平面 上で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.第三章では, 本研究で対象とする周期的繰り返し配置について解説し,周期的繰り返し配置において シーケンスペアを適用する際の問題点について説明する.そして,その問題を解決する 手法として提案する"拡大シーケンスペア"と,そのアルゴリズムについて解説する.第四 章では,Simulated Annealing 法について説明し,拡大シーケンスペアを用いた Simulated Annealing 法における隣接解,隣接解生成アルゴリズムについて解説する.第五章では, 提案手法を実装し,従来手法との比較を行い,その結果と考察を述べる.

## 第2章 準備

本章では,本研究の準備として,2次元平面上の配置最適化問題として知られている シーケンスペアについて解説する.

#### 2.1 シーケンスペア

シーケンスペアは,繰り返しがない2次元平面内の矩形配置について,矩形同士の相対 位置関係を規定するコード表現手法として提案されている.シーケンスペアは,矩形名を 要素とする2つの順列 Г<sub>+</sub>, Γ<sub>-</sub>の組みの形をとっており,各順列には,全ての矩形名が厳 密に1つずつ含まれる.2つの矩形の相対位置関係は,シーケンスペアでの矩形名の出現 順序によって次のように定められる.

シーケンスペアは,2次元平面内の矩形配置について,矩形同士の相対位置関係を規定 するコード表現手法であり,矩形名を要素とする2つの順列 Г+, Г-の組みの形をとって いる.2つの矩形の相対位置関係は,シーケンスペアでの矩形名の出現順序によって次の ように定められる.

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

上記の様に矩形名 a, bが,  $\Gamma_+$  において a, bの順で出現し, かつ  $\Gamma_-$  において a, bの順 で出現するなら, 矩形 bは矩形 aの右(矩形 aは矩形 bの左)に位置する.

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

上記の様に矩形名a, bが,  $\Gamma_+$  においてb, aの順で出現し, かつ $\Gamma_-$  においてa, bの順 で出現するなら, 矩形bは矩形aの上(矩形aは矩形bの下)に位置する.

例として,  $(\Gamma_+, \Gamma_-) = (aebcd, abdce)$  に対応する左下詰め配置を図 2.1 に示す.なお, シーケンスペアは矩形数を n としたとき,  $(n!)^2$  のサイズの解空間を持ち, 異なるコード は必ず異なるモジュールの相対位置関係を表す.

シーケンスペアを用いた相対位置関係は,45度回転させた格子グラフを用いて図示することができる.格子グラフは,以下の手順によって作成することができる.

ステップ1. 矩形数をnとし, (n×n)の格子グラフを作成し, 45 度回転する.

ステップ2.右上がりの線分に  $\Gamma_+$  の矩形名を,右下がりの線分に  $\Gamma_-$  の矩形名を割り当てる.



図 2.1: (acdb, abcd)の左下詰め配置

ステップ3.同一ラベルの線分が交差する点に矩形を配置を配置する.

上記の手順で作成したコード $(\Gamma_+, \Gamma_-) = (aebcd, abdce)$ に対応する格子グラフを図2.2 に示す.

各矩形間の相対位置関係は,2つの矩形が置かれた交点を利用して表される.2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場合,その2つの矩形は左右関係を持ち,格子の上下の交点に位置される場合,その2つの矩形は上下関係を持つ.

#### 2.2 矩形配置からコードの抽出

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 れる手法が提案されている.これは,2次元平面上に配置された各矩形からポジティブス テップライン,ネガティブステップラインと呼ばれる線を引くことでコードを抽出できる. ポジティブステップラインは,左下端から右上端へ伸びる線であり,1つの矩形に対す るポジティブステップラインは上方向(下方向),右方向(左方向)にしか線を引くことがで きず,性質として

- 他の矩形
- 他の矩形のポジティブステップライン

と交わってはいけない.



図 2.2: (*aebcd*, *abdce*) における格子グラフ

矩形の右上端(左下端)から書き出したポジティブステップラインは,上記した2つの性 質を満たす間は上方向(下方向)に進み,性質を満たされない場合,右方向(左方向)に進む.性質が満たさせる場合,再度上方向(下方向)に進み,配置エリアの隅へ到達するまで これを繰り返す.

同様に,ネガティブステップラインは,右下端から左上端へ伸びる線であり,1つの矩形に対するネガティブステップラインは上方向(下方向),左方向(右方向)にしか線を引くことができず,上記したポジティブステップラインと同様の性質を持つ.

矩形の左上端(右下端)から書き出したネガティブステップラインは,上記した2つの性 質を満たす間は上方向(下方向)に進み,性質を満たされない場合,左方向(右方向)に進 む.性質が満たさせる場合,再度上方向(下方向)に進み,配置エリアの隅へ到達するまで これを繰り返す.

各矩形から伸びたステップラインをラベル付けしたときの並び順が配置に対応するコードとなる.すなわち,ポジティブステップラインのラベルを順に並べたものが $\Gamma_+$ ,ネガティブステップラインのラベルを順に並べたものが $\Gamma_-$ に対応するコードとなる.グリッディングの例は2.3,図2.4に示す.2.3,図2.4より抽出されたコードは,

 $(\Gamma_{-}) = (\mathfrak{a}_1 c_1 b f d c_2 e_2) ,$  $(\Gamma_{+}) = (e_1 f c_2 b a c_1 e_2 d)$ 

となる.



図 2.3: ポジティブステップライン

#### 2.3 コードから矩形配置の抽出

シーケンスペアのコードから各矩形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各矩形の相対位置関係を規定している.したがって,これを2つの 矩形間の制約とみなし,定式化することができる.先ず,各変数を次のように定義する.

- *x*(*a*)... 矩形 *a* の左辺 *x* 座標
- *y*(*a*)... 矩形 *a* の下辺 *y* 座標
- w(a)... 矩形 a の幅
- *h*(*a*)... 矩形 *a* の高さ

シーケンスペアの相対位置関係から,上下関係にある2つの矩形について,以下の制約 式が成り立つ.

$$y(b) \ge y(a) + h(a) \qquad (2.1)$$

上記の式は,座標平面状においてaの上に存在する矩形bのx座標は,矩形aのx座標 に矩形aの高さを加えた座標よりも大きいことを意味する.

同様にシーケンスペアの相対位置関係から,左右関係にある2つの矩形について,以下 の制約式が成り立つ.

$$x(b) \ge x(a) + w(a) \qquad (2.2)$$



図 2.4: ネガティブステップライン

上記の式は,座標平面状において a の右に存在する矩形 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ことを意味する.

このとき,矩形同士の重なりがなく全ての制約式を満たす最小の値を各矩形の座標とする.上記の座標計算のため垂直制約グラフ,水平制約グラフを作成する.

格子グラフにシーケンスペアのコードから抽出される左右関係,上下関係に基づいて, 垂直制約グラフ $G_V(V, E)$ 及び水平制約グラフ $G_H(V, E)$ を作成する.

垂直制約グラフG<sub>V</sub>(V,E)は以下の様にして構成される(図 2.5 参照).

頂点集合 V:

source s, sink t, 矩形名をラベルとして持つ n 個の頂点 x<sub>1</sub>,x<sub>2</sub>,...,x<sub>n</sub>

辺集合 E:

 $(s, x_i), 1 \leq i \leq n,$ 

 $(x_i, t), 1 \leq i \leq n,$ 

(x<sub>i</sub>, x<sub>i</sub>), 矩形 x<sub>i</sub>, x<sub>i</sub>間において式 (2.1) が成り立つ場合

辺重み:

s に接続される辺  $(s, x_i)$ の重みは0, s 以外に接続される辺  $(x_i, x_j)$ の重みは矩形  $x_i$ の高さ

同様に,水平制約グラフG<sub>H</sub>(V,E)は以下の様にして構成される(図 2.6 参照).

頂点集合 V:

source s,



図 2.5: 垂直制約グラフ

sink t,

矩形名をラベルとして持つn個の頂点 $x_1, x_2, \ldots, x_n$ 

辺集合 E:

 $(s, x_i), 1 \leq i \leq n,$  $(x_i, t), 1 \leq i \leq n,$  $(x_i, x_i),$  矩形  $x_i$ ,  $x_i$ 間において式 (2.2) が成り立つ場合

辺重み:

s に接続される辺 $(s, x_i)$ の重みは0, s 以外に接続される辺 $(x_i, x_j)$ の重みは矩形  $x_i$ の幅

水平制約グラフ,垂直制約グラフにおいて,sourceから各頂点への最長経路長を求めることにより,各矩形の x 座標, y 座標が定まり, sourceから sink への最長経路長が,矩形を全て配置したときの配置領域の幅と高さとなる.この計算は $O(n^2)$ 時間で実行することができる.



図 2.6: 水平制約グラフ

#### 2.4 1方向繰り返し配置への拡張

1方向繰り返し配置とは,矩形集合が水平方向へ1次元的に繰り返し配置される矩形集合である.この繰り返し配置では,各矩形は周期Lごとに繰り返し配置されるため,矩形 aの左辺 x 座標は,

$$x(a) + iL \qquad (-\infty \le i \le \infty)$$

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 の下辺 y 座標は,

y(a)

である.

この1方向繰り返し配置における配置表現手法として、シーケンストリプル[4]、周期 的に繰り返す配置の効率的な表現方法[5]が提案されている.一例として、図2.7を繰り 返し配置とする.

#### 2.4.1 シーケンストリプル

シーケンストリプルは,  $\Gamma_+$ ,  $\Gamma_-$ の順列に加え, 第3の順列 $\Gamma_s$ で構成されるか解表現 手法である.3つの順列は次のように定義される.

 $\Gamma_{+}$   $\Gamma_{+}$  は,周期的なグリッディングにより抽出されるポジティブステップラインの1周 期分の順列とする.



図 2.7: 水平方向繰り返し配置

- $\Gamma_{-}$   $\Gamma_{-}$  は,周期的なグリッディングにより抽出されるネガティブステップラインの1周 期分の順列とする.
- $Γ_s Γ_s lat , 周期的なグリッディングにより抽出されるネガティブステップラインの1周期$ 分の順列とし,部分的な配置に対するポジティブステップラインの順列を表す.
- 2.4.2 シーケンストリプルから矩形配置へのデコード

シーケンストリプルでは,  $\Gamma_+$ の要素が $\Gamma_-$ のどの要素と対応しているかわからない.そ こで $\Gamma_-$ の周期を基準とした $\Gamma_s$ を使用することにで,  $\Gamma_+$ の周期情報を決定している. 例として,図2.7から抽出した各順列を,  $\Gamma_+ = (fbdaec) \Gamma_- = (fecbad) \Gamma_s = (aefbdc) と$ し,  $\Gamma_s = (aefbdc)$ を用いた $\Gamma_+ = (fbdaec)$ の周期情報抽出方法を図2.8に示す.



図 2.8: Γ<sub>s</sub>を用いた Γ<sub>+</sub>の周期情報

シーケンストリプルの対応関係から得られる左右関係,上下関係に基づいて,垂直制約 グラフ及び水平制約グラフを作成する.2つの制約グラフの枝の重みは矩形の幅や高さと なるが,異なる周期の頂点間に枝がある場合の枝重みは,頂点間の周期の差を乗じた周期 幅をその矩形の幅から減じたものとなる.

垂直制約グラフにおいては,水平関係への繰り返しのみを考えているので,垂直方向での繰り返しは存在しない.よって source 点から各矩形へと向かう最長経路長を,各矩形の下辺 y 座標とする.また,水平制約グラフにおいては,異なる周期間での同名の矩形は必ず左右関係を持つので閉路が生じる.よって先ず,すべての閉路の重み和が非正となる

最小周期Lを求める.そして任意の1つの頂点から各頂点への最長経路長を求め,対応す る矩形への左辺 x 座標とする.水平制約グラフと垂直制約グラフの一例を図2.9,図2.10 に示す.



図 2.9: シーケンストリプルにおける垂直制約グラフ

#### 2.4.3 シーケンストリプルの問題点

シーケンストリプルの多様性は,1周期分の矩形数をnとしたとき(n!3)となる.矩形 数をn=2としたとき多様性は8で,表現できる繰り返し配置の相対位置関係は3種類だけ である.矩形数をn=3としたとき多様性は216で,表現できる繰り返し配置の相対位置 関係は44種類だけであり,シーケンストリプルは,多数の表現が同一の配置に対応して おり冗長である.また,シーケンストリプルでは,図2.11(a)の様に周期をまたいだ上下 関係を表現することはできない.シーケンストリプルにおける図2.11(a)のコード表現は, 図2.11(b)または(c)の様な矩形配置となる.

#### 2.4.4 周期的に繰り返す配置の効率的な表現方法

周期的に繰り返す配置の効率的な表現方法は,1周期分の $\Gamma_-$ と,2周期分の $\Gamma_+$ によって周期的繰り返し配置を表現する方法である.この表現方法は,シーケンストリプルより 少ない表現にてどんな1次元繰り返しでも表現できる.

周期的に繰り返す配置の効率的な表現方法は,水平方向の関係を優先した相対位置関係 だけを表現の対象としている.シーケンスペアでは,有限個の矩形の相対位置関係を表す



図 2.10: シーケンストリプルにおける水平制約グラフ



図 2.11: シーケンストリプルにおける表現

ため,相対位置関係も有限通りであった.繰り返し配置では,水平方向に無限に広がるため,ある配置に対する相対位置関係も無限通り存在してしまうためである.

□□ において以下の条件に則って周期を定め,1周期毎に矩形名に添字を付ける.

第1,2周期分の $\Gamma_-$ に対応する $\Gamma_+$ を $\Gamma_+^2$ とする.すなわち, $\Gamma_+$ において,添字が1及び2であるものを抜き出して $\Gamma_+$ を $\Gamma_+^2$ とする.また,以下の条件に従って $\Gamma_+^2$ を求める.

 $\Gamma_+^2$ 第1条件:  $\Gamma_+^2$ において矩形名は1周期目と2周期目で同順で現れる.

- $\Gamma_{+}^{2}$  第2条件:  $\Gamma_{+}^{2}$  において1周期目の矩形よりも同名の2週決めの矩形が前に現ることはない.
- $\Gamma^2_+$  第3条件:  $\Gamma^1_+$ の先頭の要素を $a_1$  としたとき,  $\Gamma^2_+$  において $a_2$  は1周期目の矩形より前に現れることははい.

 $\Gamma_{-}$ ,  $\Gamma_{+}^{2}$ から,周期的繰り返しの1周期の配置を求めることができるが,このコード表現から無限長に広がる元のコード表現も可能となる. $\Gamma_{-}$ の復元は簡単であり, $\Gamma_{-}$ 1の後ろに第2周期の $\Gamma_{-}^{2}$ を追加, $\Gamma_{-}^{2}$ の後ろに第3周期の $\Gamma_{-}^{3}$ を追加,という操作をおこなうことで任意の周期*i*までの $\Gamma_{-}^{i}$ を得ることが可能である.しかし, $\Gamma_{+}^{2}$ については以下の条件に従って挿入操作を行わなければならない.

挿入条件1:挿入する矩形は1周期目と同順に並ぶように挿入しなければならない.

挿入条件2:挿入される矩形は直前の周期の同名の矩形より後ろに挿入されなければなら ない.

挿入条件3:挿入される矩形は2周期以上異なる矩形の直前に挿入されてはならない.

#### **2.4.5** 周期的に繰り返す配置の効率的な表現方法から矩形配置

各順列の対応関係から得られる左右関係,上下関係に基づいて,垂直制約グラフ及び水 平制約グラフを作成する.

1周期分の矩形に対応する点, source 点を配置し,  $\Gamma_{-}^2$ ,  $\Gamma_{+}^2$ の対応関係から得られる上 下関係に基づいて, 矩形の高さを重みとした有効枝を張る. さらに source 点から全ての 点へ重み0の辺を張り垂直制約グラフとする. 垂直制約グラフでは, 繰り返しは存在しな いので, source 点から各矩形へと向かう最長経路長を, 各矩形の下辺 y 座標とする.

一方,水平制約グラフを作成するには1周期分の矩形に対応する点だけを配置し, $\Gamma_+^2$ から $\Gamma_+^3$ を求め( $\Gamma_+^3$ , $\Gamma_-^3$ )の対応関係から得られる,推移的ではない左右関係に基づいて,(矩形の幅)-(周期の差)×(周期L)を重みとした有効枝を張る.また,水平制約グラフにおいては,異なる周期間での同名の矩形は必ず左右関係を持つので閉路が生じる.よって先ず,すべての閉路の重み和が非正となる最小周期Lを求める.ここから先はシーケンストリプルと同様の手順にて配置を得る.

上記のシーケンストリプル,周期的に繰り返す配置の効率的な表現方法は,1次元方向 にのみ回帰構造を持つ空間における矩形配置を対象にしているが,2方向以上で回帰構造 を持つ空間への拡張は考えられていない.

## 第3章 拡大シーケンスペア

本章では,先ず初めに本研究で扱う2次元トーラス空間について紹介し,その座標空間 における特徴,シーケンスペアでの問題点について述べる.そして,2次元トーラス空間 における矩形配置表現手法として提案する「拡大シーケンスペア」について解説し,拡大 シーケンスペアにおいて実権可能なコードの性質について述べる.

#### 3.1 2次元トーラス空間内の矩形配置

本研究では,図3.1に示すような,2つの座標軸,垂直,水平方向において回帰構造を 持つ2次元トーラスと呼ばれる空間を扱う.ここで考えるトーラス空間は,

$$L_h \in \mathbb{R}_+$$
,  $L_v \in \mathbb{R}_+$ 

に対して,

$$T^2 = \{(x, y) | 0 \le x \le L_h , 0 \le y \le L_v\}$$

かつ加法

$$(a,b) + (c,d) = ((a+c)_{modL_b}, (b+d)_{modL_v})$$

について閉じた空間である.以降, L<sub>h</sub>, L<sub>v</sub>をそれぞれ水平方向, 垂直方向の周期と呼ぶ. 矩形配置は矩形数 n をとしたとき, サイズを持った矩形集合

$$\mathbb{B} = \{b_1, b_2, \dots, b_n\}$$

が与えられ, T<sup>2</sup>上での重なりのない矩形配置位置

$$p: \mathbb{B} \to T^2$$

を求める問題であり,指定された周期関数 $L_h$ , $L_v$ の下での矩形配置が存在するか否かの 判定問題(又は,実際に配置を求める問題), $L_h$ , $L_v$ を変数とし,それらについてのあ る目的関数値を最小にする矩形配置を求める最適化問題などのバリエーションを持つ.な お,2次元トーラス空間内の矩形配置は,2次元空間内の周期的繰り返し配置;

$$\dot{p}: \mathbb{B} \to \mathbb{R}^2$$

但し各矩形 b<sub>i</sub> は,

$$\dot{p}(b_i) + (k \cdot L_h, \ell \cdot L_v)$$
,  $k, \ell \in \mathbb{Z}$ 

に繰り返し配置される;と等価である.



図 3.1: 周期的2次元繰り返し配置

#### 3.2 2次元トーラス空間内の矩形配置における問題点

シーケンスペアは、2次元平面上において有限で表されたコードより、矩形同士の相対 位置関係を表現する手法である.上記で述べた2次元トーラス空間は、2次元平面上にお いて2方向に無限に広がる繰り返し空間と捉えることができるので、グリッディングで表 すコード表現も無限に広がり、シーケンスペアで表現することができない.しかし、周期 が直線で分割できるような繰り返し配置は、グリッティングで1周期分のコードを取り 出し周期関数 $L_h$ 、 $L_v$ を用いることで、シーケンスペアで表現することが可能となる.図 3.1 のように、周期が直線で分割できないような繰り返し配置はシーケンスペアでは表現 できない.

#### 3.3 提案する表現手法の概要

上記した通り,2方向周期的に繰り返す空間においてシーケンスペアでは,一部の矩形 配置を除き相対位置関係を一意に決定することができない.そこで,シーケンスペアを拡 張した提案手法を用いることで,2次元トーラス空間内での相対位置関係を一意に決定す ることが可能となる.

提案手法は,ある基準となる4つの矩形の角に囲まれた矩形を1周期とすることで,無限に広がる2次元トーラス空間を,有限の1周期と捉えることで矩形配置を表現する手法である.先ず,図3.1を2次元トーラス空間における矩形集合の一例として示す.

定義:1周期の領域



図 3.2: 周期的 2 次元繰り返しにおける 1 周期抽出



図 3.3: 拡大1周期

先ず,矩形 a の左下隅を基準点として,隣接する周期の4つの a の左下隅の基準点に囲まれた長方形を1周期の領域とする.図3.2において,中の太線にて囲まれた部分を1周期の領域とする.

定義:周期内矩形

1 周期の領域から得られる1 周期の矩形集合 {*a*,*b*,*c*,*c*,*d*,*e*,*e*,*f*} から,この領域内 に完全に含まれる矩形(集合)を周期内矩形(集合)とする.図 3.2 において,中の 太線の長方形に完全に含まれる矩形集合 {*a*,*b*,*d*,*f*} を周期内矩形(集合)とする.

定義:拡大周期内矩形

1 周期の領域の境界が矩形の内部を横切るような矩形(集合){*c*,*e*}を周期境界上の矩形(集合)とする.なお,矩形 *c*のように,垂直方向の繰り返しの境界上にある矩形を垂直方向の境界上の矩形とし,1周期領域の下側の境界上の矩形を,添字

1 を付けて c<sub>1</sub> とし,上側の境界上の矩形を,添字 2 を付け c<sub>2</sub> と表すものとする.同様に e のように,水平方向の繰り返しの境界上の矩形を水平方向の境界上の矩形とし,1 周期領域の左側の境界上の矩形を e<sub>1</sub>,右側の境界上の矩形を e<sub>2</sub> と表す.ここで,周期内矩形と添字 1 を持つ周期境界上の矩形,添字 2 を持つ周期境界上の矩形 全体の集合 {a,b,c<sub>1</sub>,c<sub>2</sub>,d,e<sub>1</sub>,e<sub>2</sub>,f} を拡大周期内矩形集合とする(図 3.3 参照).

定義:拡大シーケンスペア

提案する 2 次元トーラス空間内の矩形配置に対するコード表現は,拡大周期内矩形 集合の配置に対するシーケンスペアに相当するものとなる.以下これを通常のシー ケンスペアと区別して拡大シーケンスペア( $\Pi_+$ , $\Pi_-$ )とする.この,拡大シーケン スペアは,矩形数をnとしたとき,最大 $(2n!)^2$ のサイズの解空間を持ち,異なる順 列は必ず異なる矩形の相対位置関係を表す.

拡大シーケンスペアにける実現可能なコードの性質を以下に示す.

|性質1 同名の矩形が $\Pi_+$ , $\Pi_-$ におて1回もしくは2回現れる.

性質2 同名の矩形が П+, П- おいて同じ回数現れる.

性質3 Ⅱ\_の先頭は必ず添字を持たない周期内の矩形である.

#### 3.4 矩形配置からコードの抽出

矩形配置からコードの抽出については,シーケンスペアと同様の手順でおこなう.グ リッティングに用いる矩形集合は,周期境界上の矩形集合を含む拡大1周期である.例と して,拡大1周期の図3.3におけるコードの抽出を示す.

シーケンスペア同様,各矩形から伸びたステップラインをラベル付けしたときの並び順 が配置に対応するコードとなる.すなわち,ポジティブステップラインのラベルを順に並 べたものが Π<sub>+</sub>,ネガティブステップラインのラベルを順に並べたものが Π<sub>-</sub> に対応する コードとなる.グリッディングの例は図 3.4,図 3.5 に示す.抽出されたコードは,

$$(\Pi_{-}) = (\mathfrak{x}_1 c_1 b f d c_2 e_2) ,$$
  
$$(\Pi_{+}) = (e_1 f c_2 b a c_1 e_2 d)$$

となる.

#### 3.5 拡大シーケンスペアから矩形配置

次に,拡大シーケンスペアから矩形配置を計算する手続きを説明する.手続きは,垂直 制約グラフ $G_V(V, E)$ ,水平制約グラフ $G_H(V, E)$ の作成,最小周期 $L_v(L_h)$ の計算,定 められた $L_h$ , $L_v$ の下での配置座標計算から成る.



図 3.4: 拡大1周期におけるポジティブステップライン

3.5.1 制約グラフ

垂直制約グラフ $G_V(V, E)$ は,矩形同士の相対位置関係を表現するモデルであり,以下の様にして構成される.

頂点集合 V:

拡大シーケンスペアに現れる, 各シーケンスの長さと同じ個数 N 個の頂点  $x_1$ , $x_2$ , ..., $x_N$ 

辺集合 E:

順列  $\Pi_+$ ,  $\Pi_-$ の添字無し (p,q) に対して, (3.2) が成り立つ場合 順列  $\Pi_+$ ,  $\Pi_-$ の添字有り q に対して, (3.3) が成り立つ場合

辺重み:

順列  $\Pi_+$ ,  $\Pi_-$ の添字無し (p,q)に対して, (3.2) が成り立つ場合の辺重みは, 矩形 pの高さ

順列 $\Pi_+$ ,  $\Pi_-$ の添字有りqに対して, (3.3)が成り立つ場合の辺重みは,  $y(q)+h(q)-L_v$ となる

同様に水平制約グラフ $G_H(V, E)$ は,矩形同士の水平方向の相対位置関係を表現するモデルであり,以下の様にして構成される.

頂点集合 V:

拡大シーケンスペアに現れる,各シーケンスの長さと同じ個数 N 個の頂点  $x_1$ , $x_2$ ,..., $x_N$ 



図 3.5: 拡大1周期におけるネガティブステップライン

辺集合 E:

順列 $\Pi_+$ ,  $\Pi_-$ の添字無し(p,q)に対して,

 $(\Pi_+,\Pi_-) = (\dots q \dots p \dots, \dots q \dots p \dots)$  (3.1)

が成り立つ場合

順列 Π<sub>+</sub>, Π<sub>-</sub>の添字有り q に対して, (3.4) が成り立つ場合

辺重み:

順列  $\Pi_+$ ,  $\Pi_-$ の添字無し (p,q) に対して, (3.1) が成り立つ場合の辺重みは, 矩形 pの幅

順列 $\Pi_+$ ,  $\Pi_-$ の添字有りqに対して, (3.4)が成り立つ場合の辺重みは,  $y(q)+h(q)-L_h$ となる

各制約グラフ作成において,垂直方向,水平方向に繰り返し配置されるため,同一周期の上下関係,左右関係だけではなく,異なる周期への上下関係,左右関係も考慮しなければならない.

垂直制約グラフにおいて,垂直方向繰り返し境界上以外の矩形は,隣り合う周期の垂直 方向繰り返し境界上以外のすべての矩形集合と上下関係を持つとする.同様に,水平制約 グラフにおいて,水平方向繰り返し境界上以外の矩形は,隣り合う周期の水平方向繰り返 し境界上以外のすべての矩形集合と左右関係を持つとする.そのような異なる周期間で頂 点を結ぶ辺は,周期関数  $L_v$ ,  $L_h$  を減算することにより,全ての矩形は必ず閉路を持つこ とになる.これにより,シーケンスペアにおける制約グラフでは,頂点 source, sink を使 用したが,拡大シーケンスペアのおける制約グラフでは,上記の理由から source, sink を 使用する必要がない. 垂直制約グラフ,水平制約グラフの作成において,最終的に繰り返し境界上に存在する 同名の矩形同士をマージする.これは,垂直関係を考えた垂直制約グラフにおいて,水 平関係繰り返し境界上に存在する同名の矩形は,元は周期が違う同じ矩形なので,添字 を除いたとき同名となる水平関係繰り返し境界上の矩形同士の下辺 x 座標は同値である. (図 3.6 参照)また,垂直制約グラフの計算量は,フロイド法の計算を1回行っているので  $O(n^3)$ となる.

同様に,水平関係を考えた水平制約グラフにおいて,垂直関係繰り返し境界上に存在する同名の矩形は,元は周期が違う同じ矩形なので,添字を除いたとき同名となる垂直関係繰り返し境界上の矩形同士の下辺 y 座標は同値である.(図 3.6 参照)また,水平制約グラフの計算量は,フロイド法の計算を1回行っているので  $O(n^3)$ となる.



図 3.6: 垂直関係,水平関係繰り返し境界上の矩形に対する性質

例として,垂直制約グラフを以下に示す.

ステップ1.(П<sub>+</sub>, П<sub>-</sub>)からすべての矩形を周期内矩形, 垂直方向繰り返し境界上の矩形, 水平方向繰り返し境界上の矩形, に分類する.

ステップ2.  $\Pi_+$  と  $\Pi_-$  において逆順で現れる

 $(\Pi_+,\Pi_-) = (\dots p \dots q \dots q \dots p \dots) \tag{3.2}$ 

矩形対 *p*, *q* に対して,下の矩形 *p* から上の矩形 *q* に向かう有向枝を加える.また,この枝重みを下の矩形形 *p* の高さとする.(図 3.7 参照)

ステップ3.すべての周期内矩形と水平方向繰り返し境界上の矩形に向かう有効枝を加える.これらの枝重みを(始点矩形の高さ-L<sub>v</sub>)とする.(図 3.8 参照)

ステップ4. П+ と П− において逆順で現れる

 $(\Pi_+,\Pi_-) = (\dots q_2 \dots q_1 \dots \dots q_1 \dots q_2 \dots)$  (3.3)

添字1,2を持つ同名の矩形対qをマージする.また,添字2を持つ矩形の頂点に 入っていた枝の重みを(矩形の高さ -  $L_v$ )に変更する.

ステップ5. П+ と П- において同順で現れる

$$(\Pi_+,\Pi_-) = (\dots q_1 \dots q_2 \dots, \dots q_1 \dots q_2 \dots)$$

$$(3.4)$$

添字1,2を持つ同名の矩形対 q をマージする.

図 3.8 は拡大周期内の矩形について作られる垂直制約グラフ(step1~3)である.垂直 方向の繰り返し境界上の矩形について,添字1を持つ下側の境界上の矩形の座標を考える ものとすれば,対応する上側の境界上の矩形のy座標は,下の矩形のy座標+ $L_v$ と書ける. このことから,添字2をもつ矩形頂点と対応する添字1を持つ矩形頂点へマージすると同 時に,添字2の頂点に入っていた枝重みを矩形の高さ- $L_v$ に置き換える.一方,水平方向 の繰り返し境界上の矩形対は,同一のy座標を持つので対応する頂点とマージさせる.こ うして完成した,コード  $\Pi_+=e_1fc_2bac_1e_2d$ ,  $\Pi_-=ae_1c_1bfdc_2e_2$  に対する垂直制約グラフを 図 3.9 に示す.



図 3.7: 垂直制約グラフ-1

#### 3.5.2 周期の計算

このようにして作成された垂直制約グラフは,枝重みに変数 $L_v$ を含む有向グラフとなる.垂直制約グラフの枝(a,b)とその枝重みwは,矩形a,bのy座標,y(a),y(b)に対



図 3.8: 垂直制約グラフ-2

する

#### $y(b) \ge y(a) + w$

の制約を表しており、グラフの枝重み和が正となる閉路を持たないことが、制約を満たす y座標割り当てが存在する必要十分条件となる、そこで、作成された垂直制約グラフにつ いて、全ての閉路において枝重み和が非正となるような最小の周期L<sub>w</sub>を計算する、

次に,各周期の計算のアルゴリズムについて解説する.各周期変数の計算において2分 法を用いる.2分法は,解を含む範囲,上限,下限を定め,その範囲の中間点を求める操 作を繰り返すことによって方程式を解くアルゴリズムである.

先ず,変数 MIN を周期変数が採りえる最小の経路の最大値とし,変数 MAX を周期変数が採りえ最大の経路の最大値とする. MIN を下限, MAX を上限として2分法で求めた中間点を周期関数としてフロイド法により全ての閉路において枝重み和が非正となる最小の周期変数を求める.このとき,周期変数を複数個持つ閉路が存在するため,フロイド法で2回計算する.

- ステップ1 変数 MIN を周期変数が採りえる最小の経路の最大値とし,変数 MAX を周期 変数が採りえ最大の経路の最大値として設定する.
- ステップ2 MIN と MAX の中間点 MID を MID=(MIN+MAX)/2 とする.
- ステップ3 制約グラフの周期変数を MID してフロイド法で2回計算をおこなう.全ての 閉路において枝重み和が非正となるなら MAX に MID を代入し,全ての閉路におい て枝重み和が非正とならないなら MIN に MID を代入する.
- ステップ4 MAX-MIN が1なら終了する. それ以外ならステップ2に戻る.



図 3.9: 垂直制約グラフ - 3

このアルゴリズムは,矩形数をnとすると,ステップ3のフロイド法の計算を2回によ リ $O2(n^3)$ となるが,垂直制約グラフの作成においてフロイド法の計算を1回行っている ので,計算時間は $O(n^3)$ となる.また,2分法のループの回数は,矩形サイズを定数とす るときO(logn)となる.よってトータルの計算時間は, $O(n^3) \times logn$ となる.

#### 3.5.3 座標計算

最後に,得られた周期の下で,各矩形の座標を計算する.各矩形の下辺 y 座標の計算ア ルゴリズムを以下に示す.先ず,垂直制約グラフで計算したフロイド法の配列より,ある 拡大周期内矩形から周期内矩形への最長経路長を求める.次いで,フロイド法で計算し た全ての周期内矩形の最長経路長に,(周期-最長経路長)の値を加え,周期内矩形から拡 大周期内矩形の最長経路長を求め,(拡大周期内矩形の最長経路長-周期)を拡大周期内の 各矩形の下辺 y 座標とする.最後に,周期内矩形への最長経路長を各矩形の下辺 y 座標と する.

ステップ1 ある拡大周期内矩形から周期内矩形への最長経路長を求める.

- ステップ2 周期内矩形と隣接する周期の周期内矩形は必ず上下関係を持つため,フロイ ド法で計算した周期内矩形の最長経路長に,(周期-最長経路長)の値を加える.
- ステップ3 ステップ2で計算した周期内矩形から拡大周期内矩形の最長経路長を求め,(拡 大周期内矩形の最長経路長-周期)を拡大周期内の各矩形の下辺 y 座標とする.
- ステップ4 ステップ2,ステップ3から計算した周期内矩形への最長経路長を各矩形の下 辺 y 座標とする.

上記のアルゴリズムの計算時間は, $O(n^2)$ である.

#### 3.6 実現可能なコード

拡大シーケンスペアはシーケンスペアとは違い,内部の矩形と境界上に存在する矩形 を区別して扱うため,実現可能なコードの条件をいくつか持つ.性質1~3は既に述べた. 拡大シーケンスペアおいて実現可能なコードの性質は以下の通りである.

性質4 水平方向繰り返し境界上の矩形同士の上下関係は左右の境界線上で同じでなけれ ばならない.

この性質をコード表現としてみると,  $\Pi_+$  にて添字1が先に出現する矩形名だけを  $\Pi_+$ ,  $\Pi_-$  において見たとき,  $\Pi_+$  中の添字が1の矩形の出現順,  $\Pi_+$  中の添字が2の 矩形の出現順,  $\Pi_-$  中の添字が1の矩形の出現順の逆順,  $\Pi_-$  中の添字が2の矩形の 出現順の逆順がすべて等しいという性質になる.

同様に,垂直方向の繰り返し境界上の矩形同士の左右関係を見たとき上下の境界線上で同じでなければいけない.この性質をコード表現として見ると, $\Pi_+$ で添字2が先に出現する矩形名だけを $\Pi_+$ , $\Pi_-$ において見たとき, $\Pi_+$ 中の添字が1の矩形の出現順, $\Pi_-$ 中の添字が1の矩形の出現順, $\Pi_-$ 中の添字が2の矩形の出現順がすべて等しいという性質になる.

性質 5 周期境界上の矩形は,拡大周期内の矩形配置において,最外殻の矩形である.す なわち,垂直方向の繰り返し境界上の矩形で添字1を持つもの(下側の繰り返し境 界上の矩形)の下に他の矩形がなく,添字2を持つもの(上側の境界上の矩形)の 上に他の矩形がない.また,水平方向の繰り返し境界上の矩形で添字1を持つもの (左側の境界上の矩形)の左に他の矩形がなく,添字2を持つもの(右側の境界上の 矩形)の右に他の矩形がない.

この性質をコード表現として見ると, 垂直方向の繰り返し境界上の矩形をaとして,  $\Pi_+$ における $a_2$ の左にある矩形集合と,  $\Pi_-$ における $a_2$ の右にある矩形集合は共通 要素を持たない.また,  $\Pi_+$ における $a_1$ の右にある矩形集合と,  $\Pi_-$ における $a_1$ の 左にある矩形集合は共通要素を持たない.図3.10に一例を示す.また,水平方向の 繰り返し境界上の矩形をbとして,  $\Pi_+$ における $b_2$ の右にある矩形集合と,  $\Pi_-$ にお ける $b_2$ の右にある矩形集合は共通要素を持たない.また,  $\Pi_+$ における $b_1$ の左にあ る矩形集合と,  $\Pi_-$ における $b_1$ の左にある矩形集合は共通要素を持たない.図3.11 に一例を示す.



図 3.10: 垂直方向繰り返し周期上の矩形の性質



図 3.11: 水平方向繰返し周期上の矩形の性質

## 第4章 解探索

本研究では,配置を提案手法である拡大シーケンスペアのコードを用いて表す.この コードのにより得られる解空間を,Simulated Annealing法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,先ずSimulated Annealing法について解説する.次に,拡大シーケンスペアに おける隣接解生成について述べ,隣接解生成における各アルゴリズムについて解説する.

## 4.1 Simulated Annealing 法

Simulated Annealing (SA)は,ある1つの解から別の解(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.生成された隣接解が,現在保持している暫定解と比べ て改善されていればそれを受理して暫定解を更新し,そこからさらに隣接解生成を繰り 返す.改善されなかった場合は確率的に受理,不受理を決定する.改善量を $\Delta c$ ,温度と よばれるパラメータ値をTとしたとき,解が受理される確率Pを, $P = \exp(\Delta c/T)$ とす る.解空間の探索は温度パラメータTについて高温の状態から始め,徐々に温度を下げ ながら終了温度に達するまで続けられる.温度Tが高い間は,隣接解の評価が暫定解よ り悪くともその隣接解が採用される確率が高く,温度が下がるにつれ評価が悪い隣接解が 受理されにくくなる.これにより,十分に高い温度から探索を始めることで初期解に依存 せず,かつ局所最適解から脱出するメカニズムを備えた解探索が可能となる.

#### 4.2 隣接解

拡大シーケンスペアを用いた隣接解生成方法として,以下のものが考えられる.

- 挿入: Π<sub>+</sub> または Π<sub>−</sub> における 1 つの矩形の出現位置変更
- 全交換: Π<sub>+</sub>, Π<sub>-</sub> で任意の2つの矩形の出現位置交換
- 分離:境界上に存在しない矩形(添字のない矩形名)を矩形名に添字1,2を付けて 順列に挿入
- 再結合: 周期境界上に存在する矩形を再結合させ挿入

上記した分離,再結合は,拡大シーケンスペアの特徴的な隣接解生成方法となっている.実際の解空間探索に当たっては,これらの他に矩形の配置向きを変える回転や鏡像反転も重要であるが,以下ではコードに関係する上記4つの操作について,少し詳しく説明する.

4.2.1 挿入

Ⅱ+ における挿入

- ステップ1.現在の拡大周期内の矩形集合から移動する矩形aを乱数で1つ選ぶ. $\Pi_{+} a$ をシーケンス $\Pi_{+}$ からaを抜き取った残りのシーケンスとする.
- ステップ 2-1. 選ばれた矩形 a が添字無しの場合:

周期境界上の矩形が x であって,

 $(\Pi_{+} - a, \Pi_{-}) = (\dots x_1 \dots x_2 \dots \dots x_1 \dots x_2 \dots a \dots)$ 

であるものについて,  $\Pi_+ - a$  における  $x_2$  あるいは,

 $(\Pi_+ - a, \Pi_-) = (\dots x_2 \dots x_1 \dots, \dots a \dots x_1 \dots x_2 \dots)$ 

であるもについての,  $\Pi_{+} - a$  における  $x_1$  の中で最も左に位置するものの位置を上限として記録する.また,周期境界上の矩形 x であって,

 $(\Pi_+ - a, \Pi_-) = (\dots x_1 \dots x_2 \dots, \dots a \dots x_1 \dots x_2 \dots)$ 

であるものについての  $\Pi_{+} - a$  における  $x_{1}$ , あるいは,

 $(\Pi_+ - a, \Pi_-) = (\dots x_2 \dots x_1 \dots \dots x_1 \dots x_2 \dots a \dots)$ 

であるものについての,  $\Pi_{+} - a$  における  $x_2$  の中で最も右に位置するものの 1 つ右 の位置を下限として記録する.

ステップ 2-2. 選ばれた矩形 *a* が水平方向の繰り返し境界上の矩形であって,添字1を持つ場合(すなわち *a*<sub>1</sub>):

矩形名 x (添字の有無を問わない)であって,

 $(\Pi_+ - a_1, \Pi_-) = (\dots \dots x \dots, \dots \dots x \dots a_1 \dots)$ 

であるものについての  $\Pi_{+} - a_1$  における x あるいは,周期境界上の矩形 x であって,

 $(\Pi_+ - a_1, \Pi_-) = (\dots x_2 \dots x_1 \dots \dots a_1 \dots x_1 \dots x_2 \dots)$ 

であるもについての  $\Pi_{+} - a_{1}$  における  $x_{1}$  あるいは,  $\Pi_{+} - a_{1}$  における  $a_{2}$  の中で最も 左に位置するものの位置を上限として記録する.また,周期境界上の矩形であって,

 $(\Pi_{+} - a_1, \Pi_{-}) = (\dots x_1 \dots x_2 \dots \dots a_1 \dots x_1 \dots x_2 \dots)$ 

であるものについての  $\Pi_{+} - a_{1}$  における  $x_{1}$  あるいは,周期境界上の矩形 x であって,

 $(\Pi_{+} - a_1, \Pi_{-}) = (\dots x_2 \dots x_1 \dots \dots x_1 \dots x_2 \dots a_1 \dots)$ 

であるものについての  $\Pi_{+} - a_1$  における  $x_2$  の中で最も右に位置するものの 1 つ右の 位置を下限として記録する.

- ステップ2-3. 選ばれた矩形 a が水平方向の繰り返し境界上の矩形であって添字2を持つ 場合(すなわち x<sub>2</sub>):(冗長になるため省略する)
- ステップ 2-4,5. 選ばれた矩形 a が垂直方向の繰り返し境界上の矩形(x<sub>1</sub>又は x<sub>2</sub>)である場合:(冗長になるため省略する)
- ステップ3.求められた上下限の範囲から乱数で挿入位置を決定する

この操作は,矩形数を n とすると,ステップ 2-1,ステップ 2-2,ステップ 2-4 の計算時間は $O(n^2)$ となる.よって,トータルの計算時間は $O(n^2)$ となる.

#### 4.2.2 全交換

ステップ1. 乱数で矩形を1つ選ぶ (a とする)

- ステップ2. 乱数で矩形 a とは別の矩形を1つ選ぶ(b とする)
- ステップ**3**.  $\Pi_+$ ,  $\Pi_-$ のそれぞれの矩形 a と矩形 b を入れ替える(添字があれば,添字を そのままに矩形名だけを入れ替える)

この操作は,矩形数をnとすると,各ステップの計算時間はO(1)となる.よって,トータルの計算時間はO(1)となる.

#### 4.2.3 分離

水平関係に分離する場合

ステップ1. 乱数にて周期内矩形を1つ選ぶ(aとする). 周期内矩形がⅡ\_の最初に出てく る矩形以外に無ければ分離すべき矩形は無いと判断して手続きを終える.

ステップ2. $\Pi_+$ ,  $\Pi_-$  それぞれの順列から矩形 a を抜き出し添字を付け  $a_1$ ,  $a_2$  とする.

ステップ3.  $\Pi_{-} - a \land O a_{1} O 挿入$ :

性質 3 より  $\Pi_{-} - a$  における最初の矩形名の右隣を下限と記録する.また,垂直方向の繰り返しの矩形 x について,

 $(\Pi_{-} - a + a_1) = (\dots x_1 \dots x_2 \dots a_1 \dots)$ 

となるコード表現は性質 5 を満たすことができないので,  $\Pi_{-} - a$  における  $x_2$  の中 で最も左に位置するものを上限と記録する.

ステップ4.求められた上下限の範囲から乱数で挿入位置を決定する.

ステップ5.  $\Pi_{-} - a + a_1 \land O a_2 O$ 挿入:

水平方向の繰り返しの矩形が x であって,

 $(\Pi_{-} - a + a_1) = (\dots a_1 \dots x_1 \dots x_2 \dots)$ 

であるものについて,  $\Pi_{-} - a + a_1$  における  $x_2$  の中で最も左に位置するものを上限 と記録する.水平方向の繰り返しの矩形が x であって,

 $(\Pi_{-} - a + a_1) = (\dots x_1 \dots a_1 \dots x_2 \dots)$ 

であるものについて,  $\Pi_{-} - a + a_1$  における  $x_2$  あるいは,  $\Pi_{-} - a + a_1$  における  $a_1$  の中で最も右に位置するものの1つ右を下限と記録する.

ステップ6.求められた上下限の範囲から乱数で挿入位置を決定す.

ステップ7.  $\Pi_{+} - a$ への $a_{1}$ の挿入を「 $\Pi_{+}$ の挿入」を実行しておこなう.

ステップ8.  $\Pi_{+} - a + a_1 \land o a_2 o$  挿入を「 $\Pi_{+} o$  挿入」を実行しておこなう.

この操作は,矩形数をnとすると,ステップ7,ステップ8は挿入と同じ操作をするので,計算時間は $O(n^2)$ となる.そのため,トータルの計算時間は $O(n^2)$ となる.

#### 4.2.4 再結合

- ステップ1.境界上の矩形に対して水平関係,垂直関係を判別する
- ステップ2.周期境界上の矩形を1つ選ぶ(aとする) 周期境界上の矩形が無ければ再結合すべき矩形は無いと判断して手続きを終える.
- ステップ3. 矩形  $a_1$ ,  $a_2$  を  $\Pi_+$ ,  $\Pi_-$  それぞれの順列から取り出し添字を外す
- ステップ4. $\Pi_+$ における矩形 aの挿入位置を乱数で決定する
- ステップ5.Ⅱ\_における挿入を「Ⅱ\_の挿入」を実行しておこなう

この操作は,矩形数を n とすると,ステップ 5 は挿入と同じ操作をするので,計算時間 は $O(n^2)$  となる.そのため,トータルの計算時間は $O(n^2)$  となる.

## 第5章 実験と考察

本章では,拡大シーケンスペアのコード用いた SA 法による解探索について述べる.先ず,実験の方法について解説し,拡大シーケンスペア,シーケンスペアを用いた解探索の 実験を行い,拡大シーケンスペアの有効性について考察を述べる.

#### 5.1 実験方法

プログラムは c 言語を使用し,実験は以下のシステム上で行った.

#### CPU : AMD Opteron(tm) Processor 250 2.4GMHz × 2

Memory: 8.2GB

最適化アルゴリズムの入出力,配置評価については,以下の通りである.

入力 各矩形の高さ,幅の値を乱数で決定

出力 各矩形の配置座標

配置評価 周期関数  $L_v \times L_h$ 

## 5.2 実験1:拡大シーケンスペアを用いた解探索

矩形数:30

SA パラメータ: 開始温度:20000 終了温度:1 減少係数:0.98 繰り返しループ数:500

各矩形面積の総和: 6869

隣接解生成については,以下の操作から乱数で無作為に選択しておこなう.

- 挿入: Π<sub>+</sub> または Π<sub>−</sub> における 1 つの矩形の出現位置変更
- 全交換: Π<sub>+</sub>, Π<sub>−</sub> で任意の2つの矩形の出現位置交換
- 分離:境界上に存在しない矩形(添字のない矩形名)を矩形名に添字1,2を付けて 順列に挿入
- 再結合: 周期境界上に存在する矩形を再結合させ挿入
- 回転: 任意の矩形を回転(幅と高さの入れ替)

上記の隣接解生成方法による解探索結果は,図5.1に示す.配置面積の評価値は,7110( $L_v$ 90 ×  $L_h$ 79)で,SAの実行に要した時間は25分44秒であった.また,面積占拠率は0.962045 であった.

SA で実行した際の配置面積の変化と,受理回数の変化をグラフにしたものを,図 5.2, 図 5.3 に示す.図 5.3 における配置評価値は,1温度の最終的な値を出力している.図 5.4 から図 5.8 は,各隣接解生成操作における温度あたりの受理回数と不受理回数を示してい る.図 5.9 から図 5.13 は,各隣接解生成操作における温度あたりの面積増減の平均を示し ている.

同様の実験を数回繰り返した結果,最大面積は7221,最小面積は7030,配置面積の平均は7130.4となった.

シーケンスペアと同様の隣接解生成操作は,温度あたりの面積増減の平均値を比べる と,回転,全交換,挿入の順で少ない値を示している(図5.11参照,図5.10参照,図5.9 参照).これは,これらの隣接解生成操作を行ったとき,面積の変化が少ないことを示し ている.また,シーケンスペアと同様の隣接解生成操作は,温度あたりの受理,不受理回 数の交点温度を比べると,回転,全交換,挿入の順で交点温度を低く示している(図5.6 参照,図5.5参照,図5.4参照).

拡大シーケンスペアの特徴的な隣接解生成操作である分離,再結合は,図5.12図5.13 から起伏が激しいグラフを描いている.そのうち,図5.12の分離操作に関しては,各温 度における面積増減の平均が0以下になることがなく,高い数値を示している.これは, 暫定解から分離操作を行ったとき,暫定解が大きく変わり面積が増加する確率が高いとい ことであり,その結果として,分離操作における受理回数が減りかつ,不受理回数が増え るということにつながっている(図5.7参照).また,再結合操作における面積増減の平均 値は,分離の影響を受け面積が増加した矩形配置から操作をおこなうので,分離操作が受 理されている間は低くなるが,分離操作が行われなくなると対象外(再結合操作が行われ ない)が増え図5.13のようなグラフとなる.上記のため分離操作における隣接解生成は, 面積効率を悪化させる傾向があり,温度が低くなるとまったく受理されなく,図5.1のよ うな,繰り返し境界上に矩形が存在しないような矩形配置となった.

SA における,温度5,500,15000度付近の挿入,分離,全交換の各隣接解生成操作における面積増減の分散を図5.14から図5.31に示す.15000度付近の挿入,全交換では,暫

定解と比べた場合の面積増減の平均が0前後であり,面積の増減の分散は±5000以内に 収まっていることが図5.14,図5.15,図5.18,図5.19より確認できる.一方,15000度 付近の分離に関しては,暫定解と比べた場合の面積増減の平均が1300前後で,時には暫 定解より10000近く面積効率が悪化する場合がある(図5.16参照,図5.17参照).同様に, 500度付近の挿入,全交換に関して,暫定解と比べた場合の面積増減の平均が500前後で, 面積増減の変化が少ないことが確認できるが(図5.20参照,図5.21参照,図5.24参照,図 5.25参照),一方,500度付近の分離に関して,暫定解と比べた場合の面積増減の平均が 3500前後であり,時には暫定解より10000近く面積効率が悪化する場合があることが確 認できる(図5.22参照,図5.25参照).同様に,5度付近の挿入,全交換に関して,暫定解 と比べた場合の面積増減の平均が1000前後であり,面積増減の変化が少ないことが確認 できるが,(図5.26参照,図5.27参照,図5.30参照,図5.31参照).一方,5度付近の分 離に関しては,暫定解と比べた場合の面積増減の平均が4000前後であり,時には暫定解 より9000近く面積効率が悪化する場合がある(図5.28参照,図5.29参照).上記より,挿 入,全交換と比べ分離操作における面積の増減は,変化が大きく,また非常に面積効率を 悪化させる傾向があることが確認できる.





図 5.1: 実験結果







図 5.3: 受理回数の変化



















> 5.10
全交換操
作におけ
る面積増減
域の平均

[						
		Aver	age of A	rea Chai	nge	
	-200 -250	-100	-50	50 0	150	200
20000						
13352						
8914				:		
5951				÷		
3973						
2652						
1771				- 		
1182						
789.2			-			
526.9						Full
351.8				~		
<u>9</u> 234.8				14-14-1		har
2 156.8				1ml		l Ke
104.7				Produc		
09.88				hy vy A		
46.65				ž		
31.14				~~~~		
13.99				4-40		
9 267				~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~		
6187				~~~		
4 13				<b>1</b>		
2.757				~~~w		
1.841				2		
1.229						

図 5.9: 挿入操作における面積増減の平均

	200	2000	-1500	-1000	-500	0		1500	2000	2500	
	20000			T	-	Ł					
	13352					Ł					
	8914					The second secon					
	5951					Ł					
	3973					<u>₩</u>					
	2652					Ť.					
	1771					Ţ.₩					
	1182					N N	_				
	789.2					14	-				
	526.9						A				
	351.8						5				
Ten	234.8						Ş				
Iper	156.8						Ň				
atu	104.7						Z	:			
6	69.88						444	a. M			
	46.65							٨M			
	31.14						=	AMA			
	20.79							hav			
	13.88							1			
	9.267							2			
	6.187							W.			
	4.13							YMM.			
	2.757							NA S			
	1.841							Ž			

図 5.12: 分離操作における面積増減の平均



図 5.11: 回転操作における面積増減の平均

				,	Avera	age	ofAre	ea C	hang	e			
	2000	-2500	-2000	-1500	-1000	-500	0	500	1000	1500	2000	2500	
	20000	_		-	-	-	-	_	_	_	-		
	13352						ł						
	8914						F						
	5951						ŧ						
	3973												
	2652						4						
	1771						F						
	1182						<b>₽</b>						
	789.2						1						
	526.9						÷,						_
	351.8						1	.					Roti
Ten	234.8						Š						atio
per	156.8						white we						7
atu	104.7												
e l	69.88						4						
	46.65							₹					
	31.14						1						
	20.79							3					
	13.88							\$					
	9.267							~~~~					
	6.187							w.w.					
	4.13							~~~~					
	2.757							3					
	1.841							~~~~					
	1.229												



図 5.13: 再結合操作における面積増減の平均





図 5.14: 15000 度付近の分散 (挿入)







図 5.16: 15000 度付近の分散(分離)







図 5.18: 15000 度付近の分散 (全交換)







図 5.20: 500 度付近の分散(挿入)







図 5.22: 500 度付近の分散(分離)







図 5.24: 500 度付近の分散 (全交換)







図 5.26: 5 度付近の分散 (挿入)



図 5.29: 5 度付近の分散(分離)



図 5.28:5度付近の分散(分離)







図 5.30: 5 度付近の分散 (全交換)



#### 5.3 実験2:周期境界上の矩形数を固定した解探索

実験2では,繰り返し境界上に矩形数を固定した状態で解探索をおこなう.上記により 隣接解生成操作は,繰り返し境界上の矩形数を変化させる分離,再結合の操作は除外し, 挿入,全交換,回転操作に限定しておこなう.これにより,結果として出力される矩形配 置には,必ず繰り返し境界上の矩形が存在することになる.

矩形数:30

SA パラメータ: 開始温度:20000 終了温度:1 減少係数:0.98 繰り返しループ数:500

各矩形面積の総和: 6869

隣接解生成については,以下の操作から乱数で無作為に選択しておこなう.

- 挿入: Π<sub>+</sub> または Π<sub>−</sub> における 1 つの矩形の出現位置変更
- 全交換: Π<sub>+</sub>, Π<sub>−</sub> で任意の2つの矩形の出現位置交換
- 回転:任意の矩形を回転(幅と高さの入れ替)

水平方向繰り返し境界上に1つ,垂直方向繰り返し境界上に4つ,矩形を配置した.上記の隣接解生成方法による解探索結果は,図5.32に示す.配置面積の評価値は,7134(*L*<sub>v</sub>87 × *L*<sub>h</sub>82)で,SAの実行に要した時間は27分43秒であった.また,面積占拠率は0.962854 であった.

同様の実験を数回繰り返した結果,最大面積は7221,最小面積は7030,配置面積の平均は7070.4となった.

SA で実行した際の配置面積の変化と,受理回数の変化をグラフにしたものを,図 5.33, 図 5.34 に示す.

結果として,拡大シーケンスペアを用いた解探索と周期境界上の矩形数を固定した解探 索の面積は,近い値を示している.



図 5.33: 配意面積の変化

図 5.32: 実験結果





図 5.34: 受理回数の変化

#### 5.4 実験3:周期境界上の矩形を固定した解探索

実験3では,特定の矩形を境界上に固定した状態,すなわち真に周期境界上の矩形を固定した状態で解探索をおこなう.この真に周期境界上の矩形を固定した実験は,1周期分の配置をプロセッサ,周期的繰り返し配置をマルチプロセッサの配置に対応するものとしたとき,隣り合うプロセッサ間の通信を担当するプロセッサのモジュール(ある決まったモジュール)を境界上に配置することが必要になり,このような配置問題を想定したものである.

上記により隣接解生成操作は,繰り返し境界上の矩形数を変化させる分離,再結合を除 外し,全交換における拡大周期内矩形を除いた操作,挿入,回転操作に限定しておこなう. これにより,初期値で定めた繰り返し境界上の矩形は,必ず繰り返し境界上に存在する.

矩形数:30

SA パラメータ:

開始温度:20000 終了温度:1 減少係数:0.98 繰り返しループ数:500

各矩形面積の総和: 6869

隣接解生成については,以下の操作から乱数で無作為に選択しておこなう.

- 挿入: Π<sub>+</sub> または Π<sub>−</sub> における 1 つの矩形の出現位置変更
- 全交換: Ⅲ+, Ⅲ- で拡大周期内矩形を除く任意の2つの矩形の出現位置交換
- 回転: 任意の矩形を回転(幅と高さの入れ替)

水平方向繰り返し境界上に2つ,垂直方向繰り返し境界上に2つ,矩形を配置した.上記の隣接解生成方法による解探索結果は,図5.35 に示す.配置面積の評価値は,7138(*L*<sub>v</sub>83 × *L*<sub>h</sub>86)で,SAの実行に要した時間は27分46秒であった.また,面積占拠率は0.962314 であった.

SA で実行した際の配置面積の変化と受理回数の変化をグラフにしたものを,図 5.36,図 5.37 に示す.

同様の実験を数回繰り返した結果,最大面積は7138,最小面積は7050,配置面積の平均は7102.2となった.





図 5.35: 実験結果





図 5.37: 受理回数の変化

## 第6章 まとめ

#### 6.1 2次元トーラス空間内におけるコード表現と最適化

本研究では、2次元トーラス空間内の規定されるエリアから抽出する1周期領域と、そ こから導かれる周期内矩形集合,拡大周期内の矩形集合を使用し,拡大周期内の矩形配置 に対応するシーケンスペアとして定義する,拡大シーケンスペアを提案した.拡大シー ケンスペアは、同一の矩形が少なくとも1回、高々2回(添字をつけて、各々を区別)現 れ、矩形数をnとしたとき、最大(2n!)<sup>2</sup>のサイズの解空間を持ち、異なる順列は必ず異な る相対位置関係を表す.また、拡大シーケンスペアにおいて、任意の矩形配置に対して対 応するコードが必ず存在すること、及び有効なコードであるための矩形名の並びの性質、 拡大シーケンスペアからのトーラス空間内配置計算法、SA おける拡大シーケンスペアを 用いた隣接解生成方法を明らかにした.次いで、拡大シーケンスペアにより定義される解 空間を Simulated Annealing 法により探索をおこなうプログラムを実装し、解探索の実験 を行った.この結果として以下の事柄が確認できた.

- 解探索により,必ず繰り返し境界上に矩形が存在しない矩形配置となる
- シーケンスペアより多くの計算時間が必要となる

#### 6.2 今後の課題

拡大シーケンスペアの特徴的な隣接解生成操作である分離,再結合において,面積効率 がいい生成方法,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考案が挙げ られる.隣接解生成操作の分離は,面積効率を高い確率で悪化させている.垂直方向繰 り返し境界上の配置を例に説明すると,任意の周期内矩形  $\alpha$  を繰り返し周期上に配置す るとき,図 6.1 の様に添字 1 の矩形と添字 2 の矩形の左辺 y 座標が大きくずれる場合があ る. $\alpha_1 \ge \alpha_2$  の左辺 y 座標は同じ値を持つため,結果的に図 6.2 に示すとおり面積が大き くなってしまう.

この問題の解決方法として,繰り返し境界上に接してる矩形を分離し,かつ,添字2の 矩形と添字1の矩形が直接的関係を保持するような相対位置関係を求め必要がある(図 6.3 参照).すなわち,垂直方向繰り返し境界上に分離する矩形dは,矩形 $d_1$ のy座標を矩形 dとした場合,矩形 $d_2$ は, $y(d_1) \le y(d_2) \le y(d_1) + w(d_1)$ に近似する相対位置関係を持つ 必要がある.



図 6.1: 分離操作における問題点-1



図 6.2: 分離操作における問題点-2



図 6.3: 分離操作の解決方法

謝辞

本研究を進めるにあたり,適切かつ暖かいご指導,ご教授いただ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,多くの有用な意見,議論を頂いた岩垣剛助教,研 究室の皆様にも感謝いたします.

## 参考文献

- Hyunseung Choo, Seong-Moo Yoo, and Hee Yong Youn. "Processor Scheduling and Allocation for 3D Torus Multicomputer Systems". *IEEE Trans. Parallel and Distributed Systems* vol. 11, pp. 475-484, May 2000.
- [2] H.Murata, K.Fujiyoshi, S.Nakatake, and Y.Kajitani. "VLSI Module Placement Based on Rectangle-Packing by the Sequence-pair". *IEEE Trans. Comput* vol. 15, pp. 1518-1524 Dec. 1996.
- [3] Hiroyuki Yamazaki, Keishi Sakanushi, Shigetoshi Nakatake, and Yoji Kajitani. "The 3D-packing by meta data structure and packing heuristics". *IEICE Trans. Fundamentals* vol. E83-A, pp. 639-645, April 2000.
- [4] 金子 峰雄,小川 智之."周期的に繰り返す配置の解表現と配置最適化".信学技報 CAS2004-73 pp. 7-12, 2005.
- [5] 石原 啓祐, 逢坂 龍徳, 藤吉 邦洋. "周期的に繰り返す配置の効率的な表現方法". 軽井 沢ワークショップ pp. 301-306, April 2007.