

平成 14 年度  
学士学位論文

# 部品配置問題の解法とその評価

Evaluation of Several Solution  
Methods of A Parts Placement Problem

1030264 河野 兼祐

指導教員 坂本 明雄

2003 年 2 月 12 日

高知工科大学 情報システム工学科

# 要 旨

## 部品配置問題の解法とその評価

河野 兼祐

部品を配置するスロットが2次元配列状に並んでいる長方形領域を想定し、すべての部品をスロットに割り当てたとき、部品間の接続数と部品の位置から総配線長を見積もることができる。本研究では、総配線長を最小にする部品配置を求める問題について3種類の解法をプログラミングし、それぞれから求められた解の精度と実行時間を比較して、これらの解法を評価する。

解法としては、可能なすべての部品配置を生成して、最適な解を求める列挙法、ランダムな初期配置から始めて2つの部品位置を交換していく逐次改善法、及び遺伝的アルゴリズムによる解法を取り上げる。

列挙法は部品数が10を超えると実用的な時間内では終了しないこと、逐次改善法は部品数が30を超えると解の改善があまり望めないことなどがわかった。これに対し遺伝的アルゴリズムによる解法は、部品数が100程度でも比較的短時間で良い解を得られることが確認できた。

**キーワード** 部品配置問題, 総配線長, 遺伝的アルゴリズム, 列挙法, 逐次改善法

# Abstract

## Evaluation of Several Solution Methods of A Parts Placement Problem

KOUNO Kensuke

When the slot placing parts assumes the 2-dimensional placement rectangle domain and all the parts are assigned in the slot, the total wiring length can be estimated from the numbers of the connections between the parts and the positions of the parts.

In this research, three kinds of solution methods are programmed in terms of a parts placement problem. In addition, the solution methods are compared with each other and evaluated regarding the accuracy and the execution time of the solution.

As solution methods, we adopted Enumerating Method for calculating optimal solutions, Sequential Improvement Methods, and Genetic Algorithm Method.

As a result, we found that Enumerating Method is unusable if the number of the parts exceeds 10. It was also found that Sequential Improvement Method is inefficient if the number of the parts is over 30. On the contrary, Genetic Algorithm Method proved to be the one by which good solutions can be obtained in quite a short time, even if the number of the parts is about 100.

**key words** Parts Placement Problem, Total Wiring Length, Genetic Algorithm Method, Enumerating Method, Sequential Improvement Method

# 目次

<b>第 1 章</b>	<b>序論</b>	<b>1</b>
<b>第 2 章</b>	<b>部品配置問題</b>	<b>3</b>
2.1	部品配置問題の定義 . . . . .	3
2.2	問題作成プログラム . . . . .	5
<b>第 3 章</b>	<b>列挙法によるアプローチ</b>	<b>6</b>
3.1	列挙法の流れ . . . . .	6
3.2	順列生成 . . . . .	7
3.3	列挙法の特徴 . . . . .	8
<b>第 4 章</b>	<b>逐次改善法によるアプローチ</b>	<b>9</b>
4.1	逐次改善法の概要 . . . . .	9
4.2	逐次改善法の時間短縮化 . . . . .	10
4.3	逐次改善法の流れ . . . . .	11
4.4	逐次改善法の特徴 . . . . .	12
<b>第 5 章</b>	<b>遺伝的アルゴリズムによるアプローチ</b>	<b>13</b>
5.1	遺伝的アルゴリズムの概要 . . . . .	13
5.2	遺伝的アルゴリズムの流れ . . . . .	14
5.3	遺伝的アルゴリズムにおけるモデル . . . . .	15
5.3.1	初期集団の生成 . . . . .	15
5.3.2	適応度の算出 . . . . .	15
5.3.3	両親の選択 . . . . .	15
5.3.4	交叉 . . . . .	16

## 目次

5.3.5	突然変異 . . . . .	16
5.4	部品配置問題における遺伝的アルゴリズム . . . . .	16
5.4.1	符号化 . . . . .	16
5.4.2	初期解の生成 . . . . .	17
5.4.3	適応度関数 . . . . .	17
5.4.4	両親の選択方法 . . . . .	17
5.4.5	交叉方法 . . . . .	18
5.4.6	突然変異 . . . . .	21
5.4.7	終了条件 . . . . .	22
5.5	遺伝的アルゴリズムの特徴 . . . . .	22
<b>第 6 章</b>	<b>実験結果と解法の評価</b>	<b>23</b>
6.1	実験 1(列挙法) . . . . .	23
6.2	実験 2(逐次改善法) . . . . .	24
6.3	遺伝的アルゴリズムのパラメータ設定 . . . . .	25
6.4	パラメータ設定の考察 . . . . .	26
6.5	実験 3(遺伝的アルゴリズム) . . . . .	26
6.6	各解法の評価 . . . . .	27
<b>第 7 章</b>	<b>結論</b>	<b>29</b>
	<b>謝辞</b>	<b>30</b>
	<b>参考文献</b>	<b>31</b>

# 目次

2.1	部品配置問題	3
2.2	総配線長が最小となる部品配置	4
3.1	列挙法における部品配置	7
3.2	1~ $n$ 個の順列の問題	7
3.3	再帰呼び出し	8
4.1	部品配置を交換することで総配線長が小さくなる例 (改良前)	10
4.2	部品配置を交換することで総配線長が小さくなる例 (改良後)	10
5.1	遺伝的アルゴリズムの基本的なアルゴリズム構成	14
5.2	両親の選択方法	18
5.3	切断点によって変化する操作対象	19
5.4	交叉方法	21
5.5	突然変異方法	22

# 表目次

4.1	逐次改善法改良前後の総配線長と時間 . . . . .	11
6.1	列挙法で求めたときの総配線長と実行時間 . . . . .	23
6.2	逐次改善法で求めたときの総配線長と実行時間 . . . . .	25
6.3	遺伝的アルゴリズムで求めたときの総配線長と実行時間 . . . . .	27

# 第 1 章

## 序論

2次元配列上にスロットが並んでいる基板へ電子部品を組み込むときにおいて、部品数、部品を配置するスロットの行数と列数、各部品間の配線数が指定されたとき、配線長の合計数が最小となる部品のスロットへの割り当てを求める問題を部品配置問題という。

部品配置問題は総配線長が最小となる配置を見つけ出す組合せ最適化問題であるが、問題のサイズの多項式時間で解くアルゴリズムは存在しないであろうと言われている、NP-困難な問題に属している。このような NP-困難な問題を解く多項式時間アルゴリズムは発見されていないため、NP-困難な問題を解く際には、それに対する近似解で満足しなければならないと考えられている。

部品配置問題に対して、最適解を保証する解法として、あり得るすべての配置を総当たりで調べる列挙法をまず最初にプログラミングした。次に解の導出時間の効率化も期待したいので、逐次解を改善する方向へと向けていく逐次改善法を用いることにした最後に、最適化手法のひとつとして注目を浴びている、遺伝的アルゴリズム (Genetic Algorithm: GA) を3種類目に評価する手法として採用した。遺伝的アルゴリズムは、生物種の進化過程を模倣したアルゴリズムである。生物界と同様に選択淘汰、交叉、突然変異といった遺伝的操作を繰り返し行ない、環境に適した解へと改良を重ねていく手法である。この問題に対してこれら3種類の解法をプログラミングし、それぞれから求められた解の総配線長と実行時間を評価し、考察する。

2章以降については、以下の通りである。

2章で部品配置問題についての具体例を示して説明する。3章では、最適解を得る列挙法のアプローチについて、4章では、逐次改善法のアプローチについて、5章では遺伝的アル



ゴリズムのアプローチについて個々に説明する．6章では3種類の手法を実験して得られた結果から解法の評価を行ない，最後に7章で結論を述べる．

## 第 2 章

# 部品配置問題

### 2.1 部品配置問題の定義

本研究における部品配置問題では，図 2.1 に示すように長方形のロットが 2 次元配列状に並んでいる基板を想定する．このロット上に，部品間の配線情報を見ながら部品配置を行なう．ロット上に適切な部品配置がされた場合には総配線長が小さくなり，余分な配線長を必要としないためコストの削減に繋がる．

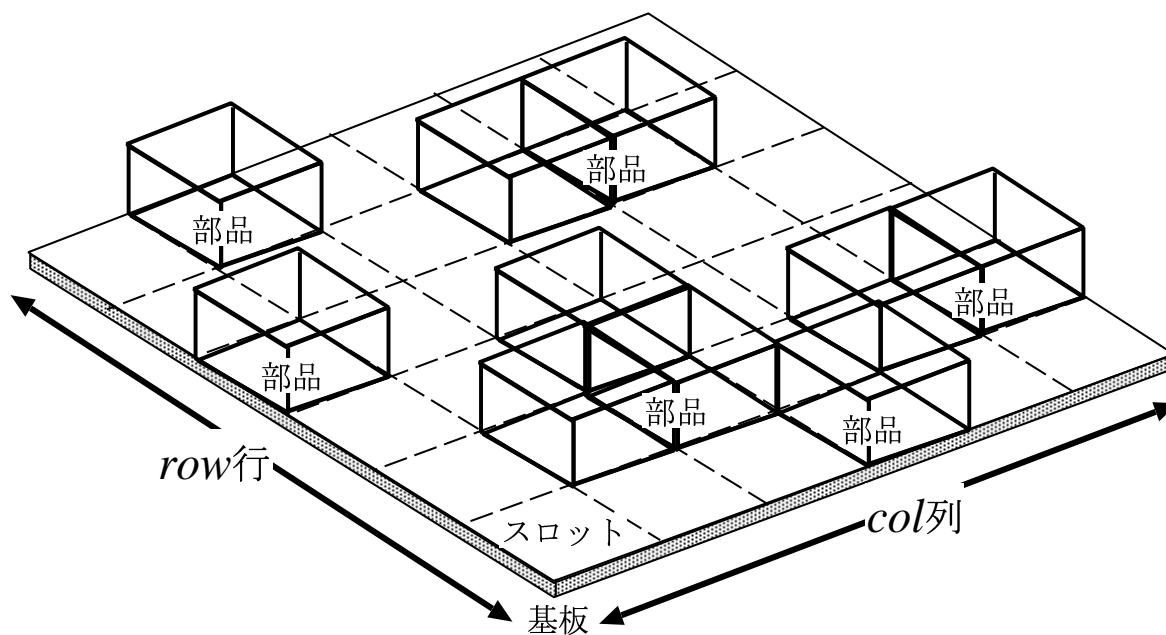


図 2.1 部品配置問題

## 2.1 部品配置問題の定義

まず,  $row \times col$  個のスポットが 2 次元配列状に並んでいる長方形領域に  $n$  個の部品を配置する問題を考える. ただし,  $row \times col \geq n$  とする.

2 つのスポット  $k$  と  $l$  の座標がそれぞれ  $(x_k, y_k)$  と  $(x_l, y_l)$  であるとき,  $k$  と  $l$  のマンハッタン距離  $d_{kl}$  は, 次式で定義される.

$$d_{kl} = |x_k - x_l| + |y_k - y_l|$$

部品  $i$  と  $j$  の接続数が  $c_{ij}$  であり, それぞれがスポット  $k$  と  $l$  に配置されているとき,  $i$  と  $j$  の配線長は,  $c_{ij} \times d_{kl}$  である. すべての部品の配置が定められると, その配置の総配線長は, これらの総和になる.

部品配置問題は, 部品数  $n$ ,  $row \times col$  個のスポット, 接続数  $c_{ij}$  が与えられたとき, 総配線長が最小となるスポットへの部品の割り当てを求める問題である [1].

図 2.2 に各部品間に接続されている配線の接続数とそれに対する総配線長が最小となる部品配置の例を示す.

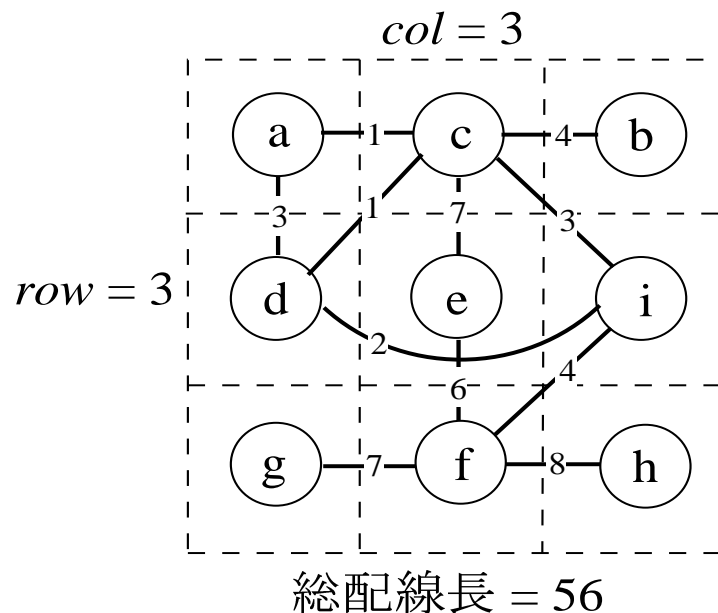


図 2.2 総配線長が最小となる部品配置

## 2.2 問題作成プログラム

本研究では，問題に対して部品数  $n$ ，行数  $row$  と列数  $col$  を与えると，ランダムに問題を作成するプログラムを製作した．各部品は，1～4 個の部品と接続しており，その接続数は 1～3 のランダムな値を与えるように問題を設定した．もし，前回と同じ部品が続けて選択された場合は，前回選択された部品間の接続数に新たに作成された接続数が追加される．この問題設定に応じてパラメータを設定することにより，各部品から接続される部品の個数とその接続数が自動作成される．

## 第 3 章

# 列挙法によるアプローチ

列挙法はあり得るすべての配置を総当たりで調べることにより、最適な解を導くことができ、最適解が保証される手法である。また、列挙法は調べるべき対象数が少ない場合は、短時間で厳密に解を得ることができるが、対象数が増えるにつれ時間を膨大に費やす。

### 3.1 列挙法の流れ

まず、 $n$  個の部品を順列生成により、 $n!$  通りの部品配置を生成する。そして、それぞれの順列に対して総配線長の計算を行ない、総配線長が最小となる部品配置を得る。

ここで、順列生成により得られた順列は、部品番号に対する配置位置を表す。生成された順列に対する部品配置の例を図 3.1 に示す。ただし、スロット番号と区別するため、部品番号を  $a, b, c, d$  とする。

図 3.1 の例では、まず、順列  $[1\ 3\ 4\ 2]$  に対応する部品配置は、左上に部品  $a$  が置かれ、右上に部品  $d$  が置かれ、左下に部品  $b$  が置かれ、右下に部品  $c$  が置かれている。この配列の  $1, 3, 4, 2$  の数字は部品が置かれるスロット番号を示し、配列の順が部品番号に相当する。スロット番号は、左上を  $1$  として右へ進み、 $1$  行目が終われば、 $2$  行目の左端へ順に番号がつけられる。よって、順列  $[1\ 3\ 4\ 2]$  の場合、部品  $a$  がスロット番号  $1$  に置かれ、部品  $b$  がスロット番号  $3$  に置かれ、部品  $c$  がスロット番号  $4$  に置かれ、部品  $d$  がスロット番号  $2$  に置かれることになる。同様に、順列  $[2\ 4\ 1\ 3]$  が生成されたとすると、部品  $a$  がスロット番号  $2$  に置かれ、部品  $b$  がスロット番号  $4$  に置かれ、部品  $c$  がスロット番号  $1$  に置かれ、部品  $d$  がスロット番号  $3$  に置かれる。

### 3.2 順列生成

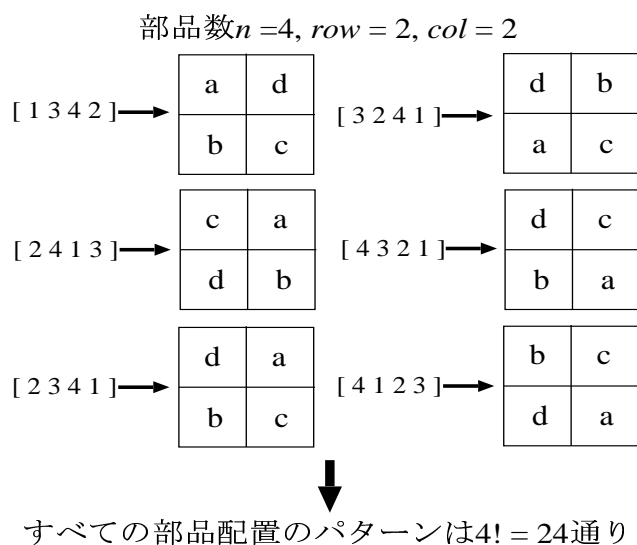


図 3.1 列挙法における部品配置

### 3.2 順列生成

$n$  個の部品数  $1, 2, 3, \dots, n$  の並べ方は,  $n! = n(n-1)(n-2)\dots 3 \cdot 2 \cdot 1$  通りある.

例えば,  $n=3$  ならば,

123, 132, 213, 231, 312, 321

の  $3! = 3 \cdot 2 \cdot 1 = 6$  通りある. この並べ方をすべて列挙する.

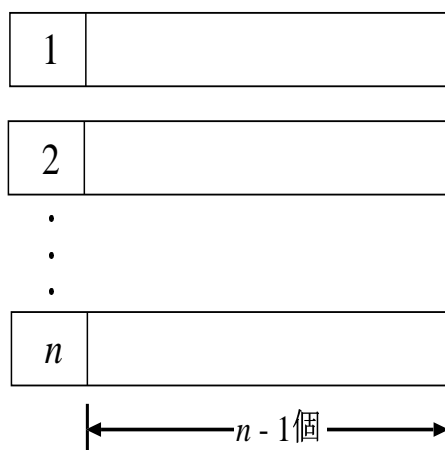


図 3.2  $1 \sim n$  個の順列の問題

図 3.2 のようにそれぞれ  $1 \sim n$  を先頭とする  $n-1$  個からなる順列の問題に分解でき,

### 3.3 列挙法の特徴

$n - 1$  個からなる順列についても同様なことがいえる。

先頭に  $1 \sim n$  の値を持ってくるようにするには、数列の第 1 項と第 1 項～第  $n$  項のそれぞれを逐次交換することにより行なう。

$\{1, 2, 3, 4\}$  に対して以下に例を示す。1, 2, 3, 4 の順列は、

- 1 と 1 を交換してできる 1, 2, 3, 4 のうちの 2, 3, 4 の順列の問題に分解
- 1 と 2 を交換してできる 2, 1, 3, 4 のうちの 1, 3, 4 の順列の問題に分解
- 1 と 3 を交換してできる 3, 2, 1, 4 のうちの 2, 1, 4 の順列の問題に分解
- 1 と 4 を交換してできる 4, 2, 3, 1 のうちの 2, 3, 1 の順列の問題に分解

となり、再帰呼び出しで表現できる。再帰呼び出しの前に第 1 項と第 1 項～第  $n$  項のそれぞれを交換する処理を置き、再帰呼び出しの後に、交換した数列を元に戻す処理を置く [3]。

図 3.3 のように交換の基点  $i$  は再帰呼び出しが深くなるたびに 1 から  $n$  に向かって、1 つずつ右に移っていく。

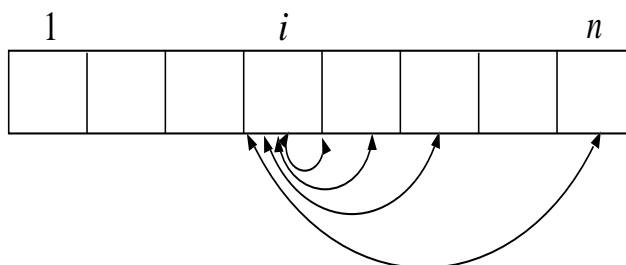


図 3.3 再帰呼び出し

### 3.3 列挙法の特徴

列挙法は、すべてのパターンを総当たりで計算することによって、最適解を割り出すという手法である。つまり、要素数が増えるに伴って、探索時間も膨大になる。

このことから、部品配置問題において適切な場所に部品が置かれているかは、順列生成で生成されたすべての部品配置の総配線長を計算しなければならない。その結果、部品数に対応する探索範囲に伴って計算時間が加算されていく。

## 第 4 章

# 逐次改善法によるアプローチ

逐次改善法は、問題の持っている性質を取り入れ、逐次、新たな解を探し改善を重ねていく手法である。

### 4.1 逐次改善法の概要

本研究の部品配置問題に対する逐次改善法は、まず解探索を始める前に初期配置を乱数によって生成する。生成された配置から総配線長が小さくなるような配置を捜し出すために、乱数により選ばれた 2 点の部品の位置を交換する。

このとき、部品の位置を交換することで解が改善される場合はその交換を採用する。解が改善されない場合は、以前の配置に戻すという操作を行なう。この操作を繰り返すことで、逐次改善法における最良解が得られる。

逐次改善法を用い部品の位置を交換することによって、総配線長が小さくなる例を図 4.1、図 4.2 に挙げる。

図 4.1 のように配線接続数とそれに対応する配置に部品が置かれていたとする。この図の配線接続数は部品 1 と部品 2 の間に 2 本、部品 1 と部品 3 の間に 1 本、部品 1 と部品 4 の間に 3 本、部品 3 と部品 4 の間に 1 本である。図 4.1 の配置では、総配線長は 10 である。図 4.1 の配置を逐次改善法で、部品 3 と部品 4 の位置を入れ換えたたとすると図 4.2 のような配置になる。この配置の総配線長は 8 であり、解が改善されたといえる。



## 4.2 逐次改善法の時間短縮化

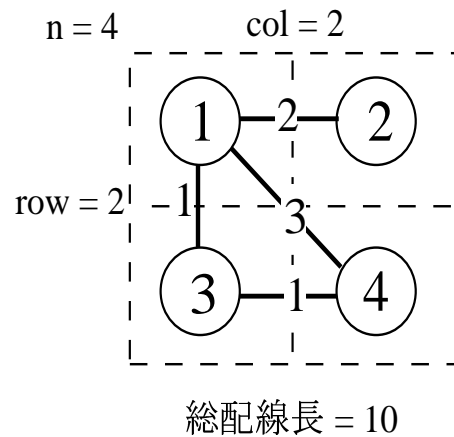


図 4.1 部品配置を交換することで総配線長が小さくなる例 (改良前)

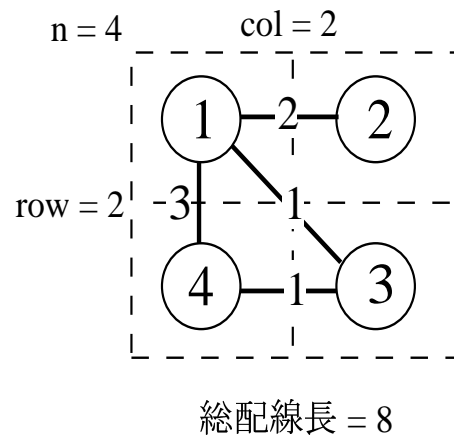


図 4.2 部品配置を交換することで総配線長が小さくなる例 (改良後)

## 4.2 逐次改善法の時間短縮化

逐次改善法は、終了条件として定める繰り返し回数をより増やすことで、解が改善される可能性は高くなるが、繰り返し回数を増やせばそれに伴い、計算時間は増大する。逐次改善法の計算時間短縮化を図ることで、繰り返し回数を増やしたとしても短時間に精度が高い解を導出することが期待できる。

逐次改善法によってランダムに選ばれたその2部品の位置を交換するとき、交換が必要か不必要かは交換する部品に関わる配線接続数と配線長で決まる。したがって、以前の逐次改善法では、部品位置を入れ換えるかどうかの判定は総配線長の比較によって決めていたが、

### 4.3 逐次改善法の流れ

改良後の逐次改善法では、逐次改善を行なうとき、交換する2部品に関する配線接続数と配線長を計算することで、逐次改善法の時間短縮が望める。

各部品数の繰り返し回数における、改良前の逐次改善法と改良後の逐次改善法で解を求めたときの実行時間の比較を表 4.1 に示す。

表 4.1 逐次改善法改良前後の総配線長と時間

部品数	繰り返し回数	総配線長	改良前	改良後
			時間 [sec]	時間 [sec]
50	1,000	723	0.057	0.012
	5,000	636	0.306	0.056
	10,000	630	0.606	0.121
	50,000	630	3.072	0.606
	100,000	630	6.094	1.218
100	1,000	2198	0.120	0.019
	5,000	1791	0.601	0.056
	10,000	1969	1.207	0.116
	50,000	1612	5.975	0.581
	100,000	1608	11.910	1.160

### 4.3 逐次改善法の流れ

逐次改善法のアルゴリズムを以下に示す。

1. 繰り返し回数を  $T$  回と設定する。
2. ランダムに部品配置を作成し、総配線長を計算する。
3. 作成された部品配置の中から、ランダムに2つの部品を選択する。

#### 4.4 逐次改善法の特徴

4. 選択された2つの部品の位置を交換したときと交換する以前のときとで、その部品に関わりのある部品の配線長の差をみて、交換したときの配線長が、交換する以前の配線長以下になれば部品位置を交換する。以前の配置よりも配線長が大きくなる場合は、部品位置の交換を行なわない。
5. 3~4 を、1 によって指定された数まで繰り返す。
6. 繰り返し交換が行なわれた中で、繰り返し回数  $T$  まで解が改善されなかったとき、最終的に配線長が小さい配置を最小総配線長として得る。

逐次改善法での終了条件は、あらかじめ指定した繰り返し回数の間、解が改善されなかったとき終了する。指定した繰り返し回数の途中で解の更新がされた場合には、繰り返し回数  
の値を初期化する。

#### 4.4 逐次改善法の特徴

逐次改善法は、アルゴリズムが比較的単純で短時間に解を導くことができるが、初期配置によって解探索の経路方向は決定してしまうことから、局所解に陥りやすく、最適性に欠ける。また、逐次改善法は、いつ最適解に到達したか明確に知ることができないため盲目的であるという特徴をもつ。

## 第 5 章

# 遺伝的アルゴリズムによるアプ ローチ

生物界では，任意の生物種は移りゆく環境変化への適応の善し悪しで何世代もかけて選択淘汰される．結果として環境により良く適合した染色体を持つ生物種が生き残る．

この生物種の進化過程を模倣する形で，解集団を世代毎に逐次改良する解析方法を遺伝的アルゴリズム (Genetic Algorithm: GA) という [1]．

### 5.1 遺伝的アルゴリズムの概要

遺伝的アルゴリズムは，ダーウィンの理論にもとづき自然界の進化過程をシミュレートした探索技術である．最適解を目指し系統的に解空間を探索するために，生物の進化過程を模倣した手法である [4]．

遺伝的アルゴリズムは，記号列としてコード化された個体の集団を操作する．これらの記号列は解空間の点を表し，世代と呼ぶ．それぞれの繰り返し処理において，世代ごとのいくつかの記号列を交叉させて，新しい解を表す記号列の集合が生成される．遺伝的アルゴリズムは，個体集合のなかで，最も適応するものを生存させることで行なう情報交換と探索を結びつけている．最も高い適応度を持つものが生き残るという生物学的原理と結びつけて自然界の生殖過程をシミュレートすることにより，この手法も同様に世代が進むにつれて高い適応度を持つ個体 (解) が生まれると期待している [1]．

## 5.2 遺伝的アルゴリズムの流れ

遺伝的アルゴリズムは、図 5.1 のようなアルゴリズムによって構成されている。

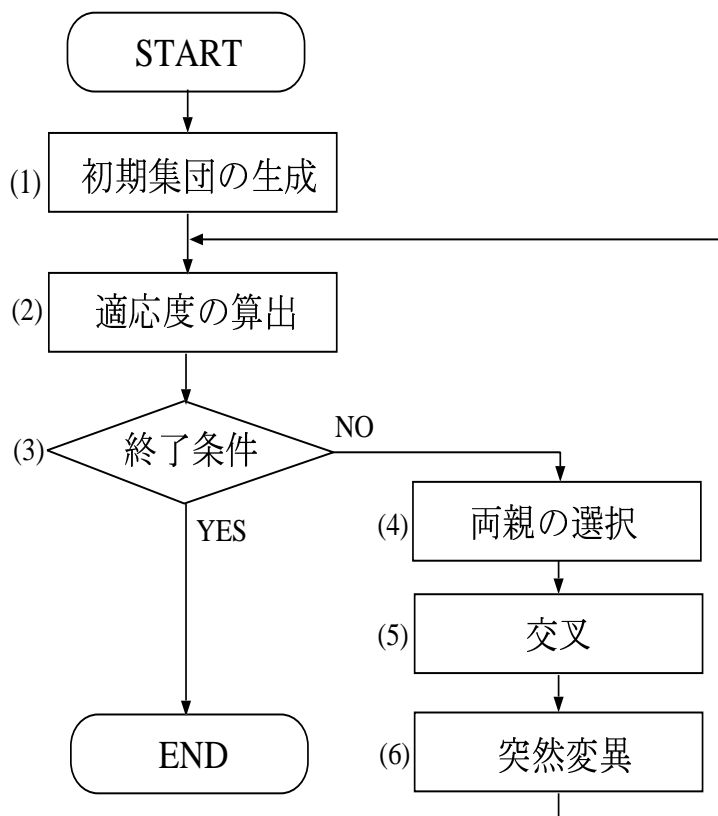


図 5.1 遺伝的アルゴリズムの基本的なアルゴリズム構成

まず、(1) 初期集団の生成を行ない、決められた個体数の染色体をランダムに生成する。初期集団が生成されると、各々の個体に対して (2) 適応度の算出を行なう。個体に適応度が決定されたら、それを基に (3) で終了条件の判定を行なう。終了条件を満たしていない場合は、(4)、(5)、(6) へ移行する。満たした場合は終了する。終了条件を満たしていないとき適応度を基に (4) 両親の選択を行なう。適応度の高い個体が親として子孫が残せる。これにより、よりよい個体を形成する遺伝子が集団中に広がる。交配を行なう個体対が決定されたら、染色体の (5) 交叉を行なう。交叉は基本的に双方の親となる染色体の一部ずつを採ってきて、子の染色体を作る。次に (6) 突然変異を加える。これは、ある確率で染色体の一部の値を変える操作である。これらの操作が終了すると、新しい世代の個体集団が作られたこと

### 5.3 遺伝的アルゴリズムにおけるモデル

になる。そして、再度、(2) 適応度の算出を行ない、(3) 終了条件を満たしていなければ、両親の選択、交叉、突然変異を行ない、新たな世代を作っていく。条件を満たした場合は、生成された個体集団中で最も適応度が高い個体を遺伝的アルゴリズムにおける最良解として得る [2]。

## 5.3 遺伝的アルゴリズムにおけるモデル

### 5.3.1 初期集団の生成

初期集団の生成が子孫の解の質に影響する。これは、各世代数において生成される子孫によって受け継がれてゆくのは、まさに、初期集団の要素の特徴であるという理由による。初期集団がよければ、これらの子孫のうちのひとつが望む解になるかもしれない。しかし、全パターンの最適解が盲目的であるため初期集団の生成は通常、ランダムに構成される。

### 5.3.2 適応度の算出

遺伝的アルゴリズムでは、各個体が自らの遺伝子の並びによって環境にどれだけ適応できるかを表す関数値として適応度を算出する。

染色体の適応度を示す値は、個体の良さを示す尺度となる正整数である。最小化問題の場合、より小さい評価値を持つ解が、より適応性が高い個体に対応する。

### 5.3.3 両親の選択

現世代の個体の中から次世代に子を残すであろう親個体の適応度に応じた選択により多数選別される。すなわち環境にそぐわない個体は淘汰される。しかし、適応度に応じて過度に反応して選択が行われると遺伝子の多様性が失われ易く初期段階の世代で、最適化問題でいう局所解に落ち込む恐れが生じる。通常は各世代での個体数はほぼ一定に保つようするため、選択された親の個体数を調整する。

## 5.4 部品配置問題における遺伝的アルゴリズム

### 5.3.4 交叉

交叉は、両親の持つ遺伝子の組み合わせの中で一部の遺伝子を交換する形で変化を起こさせる。通常の遺伝的アルゴリズムでは、主に両親の染色体の一部を確率(交叉確率)で、交換し変化を起こす。交叉方法は分析の目的や問題の設定に応じて適切にする必要がある。特に交叉の方法についての条件がないのであれば、同じ位置で同じ文字数からなる部分文字列を交換する。この操作によって、より優れた両親の遺伝的形質の多くを引き継ぐことと、多種多様な染色体(解候補)を同時に生み出すことが期待できる。

### 5.3.5 突然変異

突然変異は、幾つかの遺伝子に関して対立遺伝子の値をランダムに変更することによって、子孫にランダムな変化を与える。突然変異は、集団のすべての要素に対して適用されるのではなく、幾つかの要素に確率的に適用される。突然変異は、親の世代のどの要素も持っていない新しい特徴を導入するために、親となる染色体の一部の遺伝子をランダムに変化させる。突然変異は、あまり大きな変異確率に設定すると、染色体の特徴が破壊され、ランダム・サーチと化してしまうおそれがあることから確率をできるだけ小さい値に設定する方が有効的である。

## 5.4 部品配置問題における遺伝的アルゴリズム

部品配置問題を遺伝的アルゴリズムを使用して解くにあたり、順列符号化 GA(坂本研究室, 赤間 寛君 作成) の環境を利用した。

### 5.4.1 符号化

遺伝的アルゴリズムを部品配置問題に適用するために、順列符号化を使用した。順列符号化は、順列から形成されているコードであり、各遺伝子座の数字がそれぞれ違う整数値となっている。この順列符号化を利用すると、各遺伝子座が部品番号に、遺伝子が部品位置に

## 5.4 部品配置問題における遺伝的アルゴリズム

対応する.

### 5.4.2 初期解の生成

初期解の生成は、染色体の個数  $n$  とひとつの染色体の遺伝子数  $m$  の設定によりランダムに生成する。つまり、 $n$  個の染色体のそれぞれに対して、0 から  $m - 1$  までの整数がランダムに列べられたものを割り当てる。

### 5.4.3 適応度関数

適応度の関数  $f$  は、部品間のマンハッタン距離 × 部品間の接続数の和である総配線長を  $length$  とするとき、以下の式で表される。

$$f = \frac{1}{length}$$

$f$  は  $length$  が小さいほど適応度が優れていることを意味する。

### 5.4.4 両親の選択方法

ルーレット戦略は、ルーレット盤をシミュレートする。集団のそれぞれの個体に対して、円盤上にその相対適応度に比例する面積の扇形領域を割り当てる。親を選択するためには、円盤を回転させ、停止したときに印に位置する領域の親を求める。したがって、この方法では、低い適応度をもつ個体は、すぐに淘汰されるのではなく、交叉処理のために選択される小さい確率を持つことができる。適応度が低い個体は扇形領域が小さいため選択されにくく、適応度が高い個体は領域が広いため選択されやすい。図 5.2 のルーレット盤では、最も高い適応度を持つ個体は最も広い面積を占めており、その占有率は 42% であり、最も低い適応度を持つ個体は、最も狭い面積しか与えられず占有率は 5% である。よって、最も高い適応度を持つ個体は最も低い適応度を持つ個体よりも親として選ばれる可能性が高い。

トーナメント戦略は、集団から決められた数の個体が無作為に選択し、その中で最も適応度の高い個体を次世代に残すという手続きを次の世代に残したい数の個体を選択されるまで



## 5.4 部品配置問題における遺伝的アルゴリズム

繰り返す。これは、集団化されている染色体の間で、比較(競争)させることから構成されている。この選択方法では、最悪の染色体は全く選択されないが、平均値以下の適応度を持つ個体は、まだ親として選択されるチャンスが残されている。図 5.2 の例では、[4 3 2 1 5 6] のように各個体が並べられている。番号は各個体の適応度を示しており、この場合では、数が大きくなるほど適応度値が高いことを示す。個体を並べる順はランダムにする。並べられた個体を図 5.2 では、2 グループに分けている。この 2 集団から、2 組の親を選び出す。選び出す方法は、その親の持つ適応度によって決まる。集団内で最も高い適応度を持つ個体を選ぶようにする。よって常に最も高い適応度を持つ個体 6 は親として選ばれ、最も低い適応度を持つ個体 1 は選ばれることはないが、ランダムに集団構成するため、それ以外の 5, 4, 3 の適応度を持つ個体はどの個体が選ばれるかは一定していない。

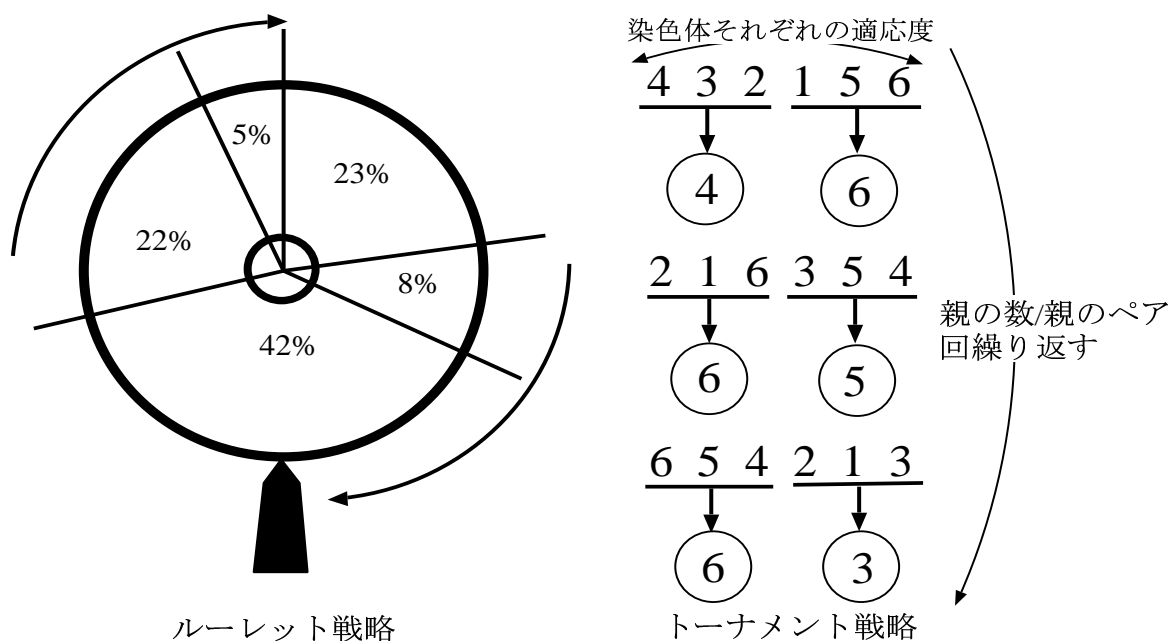


図 5.2 両親の選択方法

### 5.4.5 交叉方法

交叉方法は、順序交叉 (Order Crossover: OX), 部分写像交叉 (Partially Mapped Crossover: PMX), サイクル交叉 (Cycle Crossover: CX), 一様順序交叉 (Template Order

#### 5.4 部品配置問題における遺伝的アルゴリズム

Crossover: TOX) の 4 種類を用いた。またその内、順序交叉と部分写像交叉については切断点によって特性が変化する。切断点を 1 つにする場合と切断点を 2 点にする方法がある。また、切断点に対して、左右のどちらかの辺を操作対象として扱うか、切断点を 2 つにした場合切断点に対して内外のどちらを操作対象とするかによっても交叉が変化する。したがって、交叉は全部で 10 種類になる。切断点を決めたときの操作対象を図 5.3 に示す。

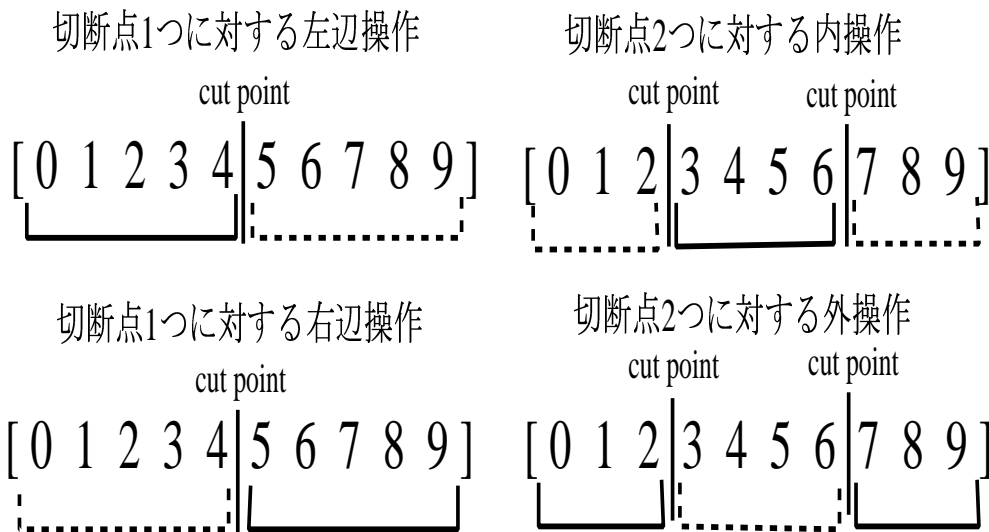


図 5.3 切断点によって変化する操作対象

順序交叉 (OX) は、対立遺伝子の相対位置に注目する。親となる染色体の遺伝子順序を交叉するもう一方の親の染色体に組み換え、子となる染色体を新たに生成する。図 5.4 のようにまず切断点を決め、親  $P_1$  の遺伝子左辺を子  $H_1$  に写す。子  $H_1$  の右辺は、親  $P_2$  の遺伝子の出現順序に従い、親  $P_1$  の左辺にない遺伝子を子  $H_1$  の右辺に写す。

部分写像交叉は、対立遺伝子の絶対位置に注目し、親によって部分的に決まる遺伝子の順序情報を持つ。親となる染色体の交叉箇所を子となる染色体に継承し、致死遺伝子をなくすように補正して子を生成する。図 5.4 のようにまず切断点を決め、親  $P_1$  の遺伝子左辺を子  $H_1$  に写す。子  $H_1$  の右辺は、親  $P_1$  の対立遺伝子である親  $P_2$  の遺伝子座を見つつ、親  $P_1$  の染色体の左辺に対立遺伝子と同じ遺伝子がないかを確認して子  $H_1$  に写す。同じ遺伝子である場合は、その親  $P_2$  の対立遺伝子と親  $P_1$  の同じ遺伝子を探し、その親  $P_1$  の遺伝子の対

#### 5.4 部品配置問題における遺伝的アルゴリズム

立遺伝子である親  $P_2$  の遺伝子座を写す.

サイクル交叉は、両親の染色体の遺伝子の *cycle*(2つの親の対立遺伝子からなる位置情報)を求め、その遺伝子を子染色体に継承する. 図 5.4 のように1回目、親  $P_1$  の先頭の位置からサイクル交叉の処理が行なわれる. 親  $P_1$  の先頭に位置する遺伝子を子  $H_1$  に写し、その対立遺伝子である親  $P_2$  の遺伝子座にある遺伝子を子  $H_1$  に写す. その遺伝子と同じ親  $P_1$  の遺伝子を探す. その遺伝子の対立遺伝子である親  $P_2$  の遺伝子を見て子  $H_1$  に写す. この繰り返しを始めた遺伝子までサイクルする.

一様順序交叉は、一様交叉と順序交叉を組み合わせた交叉方法である. 一様交叉は、まず、2進記号列のテンプレートをランダムに生成する. 例えば、テンプレートが記号列 [101001] のとき、子孫の第1ビット、第3ビット、及び、最後のビットが親1から、残りの3ビットが親2からテンプレートにしたがって、割り当てられる. 第2の子孫も、 $P_2$  と  $P_1$  を逆にすることによって、同様に生成される. 一様順序交叉でも、このランダムな2進記号列のテンプレートを使用し交叉させる. テンプレートにおける1は  $P_1$  から  $H_1$  へコピーされ、0ならば  $P_2$  から  $H_2$  へと写される. この操作の終了後、まだ子孫へ写されていない遺伝子集合については、 $P_1$  の遺伝子集合は  $P_2$  に出現順序に従い、 $P_2$  の遺伝子集合ならば  $P_1$  の出現順序に従って、各々の集合をソートし再構築する. その構築された集合を  $H_1$ ,  $H_2$  のそれぞれ空いている遺伝子座に複写していく.

5.4 部品配置問題における遺伝的アルゴリズム

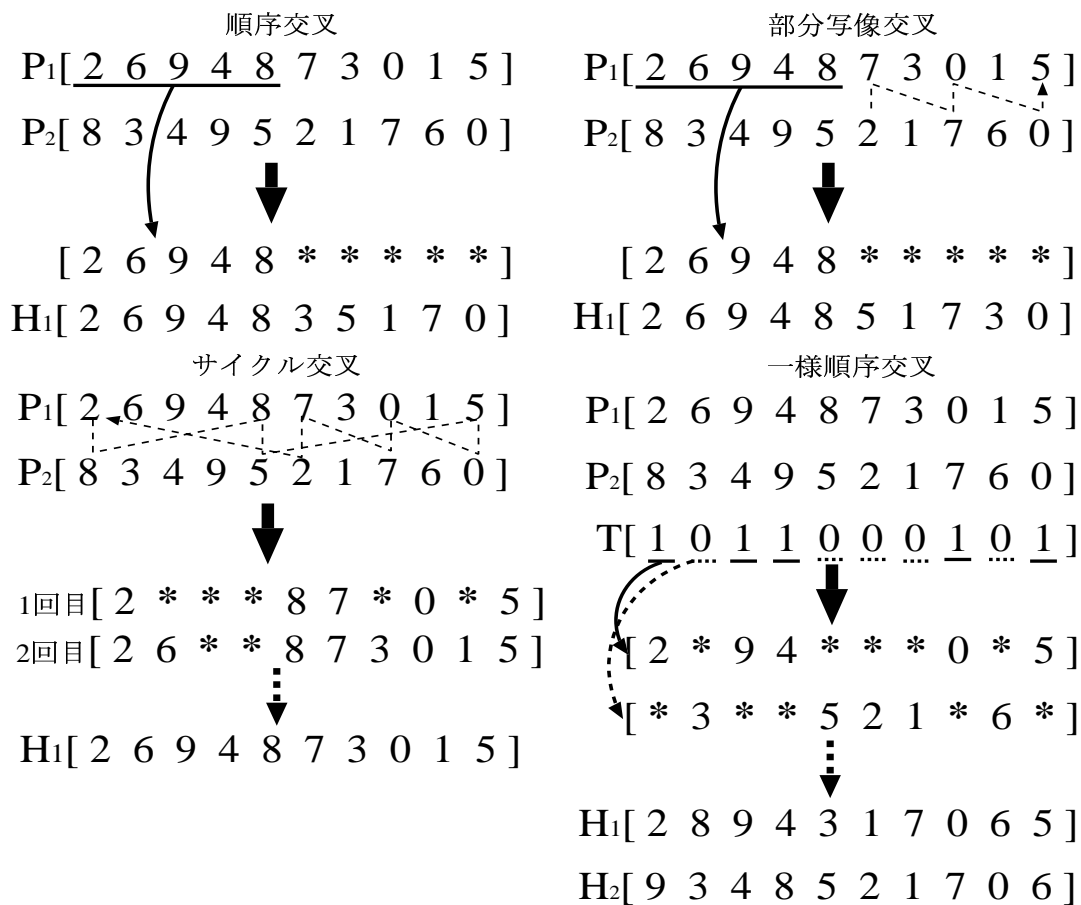


図 5.4 交叉方法

5.4.6 突然変異

ランダムな 2 点間入れ換えの方法は、乱数によって選ばれた親となる染色体の遺伝子を 2 つ選び、その遺伝子の遺伝子座を交換する。

ランダムな 2 点間逆順並べ換えの方法は、乱数によって選ばれた親となる染色体の遺伝子を 2 つ選び、2 点間の間で逆順に転換させ、遺伝子座に入れていく操作をする。また、その 2 点間の外側を逆順に転換する場合もある。

ランダムな 2 点間をランダムに攪拌する方法は、乱数によって選ばれた親となる染色体の遺伝子を 2 つ選び、2 点間のある遺伝子をランダムに遺伝子座へ配置する。また、その 2 点間の外側にある遺伝子をランダムに配置する場合もある。

## 5.5 遺伝的アルゴリズムの特徴

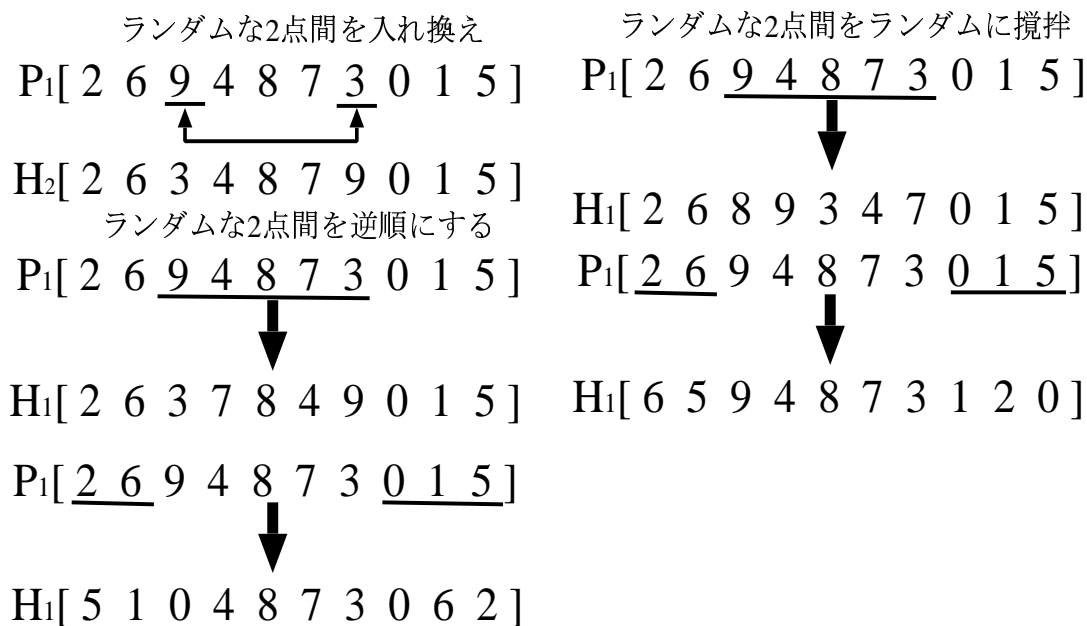


図 5.5 突然変異方法

### 5.4.7 終了条件

終了条件は、世代を繰り返していく中で、最良解を得た時点から解が  $T$  世代数まで未更新の状態であるとき終了とする。

## 5.5 遺伝的アルゴリズムの特徴

遺伝的アルゴリズムは、逐次改善法と同様に決定的な遷移規則はなく、確率的な遷移規則を使用する。よって、必然的に最適解を発見できる保証はなく、非決定的な手法である。また、遺伝的アルゴリズムはいつ最適解に到達したかが明確に知ることができないという意味で、盲目的であるといえる。したがって終了条件を外部から指示しなければならない。遺伝的アルゴリズムは、交叉によって前の染色体の有利な特徴を利用してより優れた解を構築でき、突然変異によって局所解に陥る可能性が低いアルゴリズムであるため、単純な並列的探索手法に比べて高品質の解が期待できる。パラメータ設定での終了条件を厳しい設定にすれば、実行時間が長くなる場合もあるがより良い解が求められやすい。

## 第 6 章

# 実験結果と解法の評価

各手法それぞれに対して、部品数を変えた部品配置問題に当てはめて実験を行なう。実験から得た結果から、部品配置問題における 3 種類の解法の総合的な評価をする。各解法に対して部品数  $n$  のときの、総配線長と実行時間を示す。逐次改善法と遺伝的アルゴリズムは、部品数  $n$  に対して乱数を変え 10 回試行したときの総配線長の平均 (総配線長 Avg) とその総実行時間、試行した中での最良解 (総配線長 Best) を表に示す。ただし、時間が 0.000 となっているのは、0.001 秒未満であることを意味する。

### 6.1 実験 1(列挙法)

最適解を求める列挙法から得られた実験結果を表 6.1 に示す。

表 6.1 列挙法で求めたときの総配線長と実行時間

部品数	総配線長	実行時間 [sec]
4	21	0.000
6	35	0.021
8	57	2.604
10	73	355.853
12	—	—

## 6.2 実験 2(逐次改善法)

列挙法は最適解を求められる保証を持っている。よって、表 6.1 の総配線長は最適な部品配置の値を示していることから、精度が高い解法であることがいえる。表 6.1 から部品数 4 ではほとんど実行時間を費やしていないことがわかる。それ以降、部品数が増すにつれて、同時に実行時間も長くなっていることが読みとれる。部品数 12 のとき結果が得られなかった理由としては、部品数 12 の場合、 $12!$ 通りの部品配置が存在するからだと考えられる。部品数 10 のとき実行時間で約 355 秒かかっていることから、さらに部品配置方法が増える部品数 12 のときは、より探索範囲が広がるため一般的にみて、実用的な時間を超えてしまうと推測される。

## 6.2 実験 2(逐次改善法)

逐次改善法から得られた実験結果を表 6.2 に示す。

表 6.2 から、部品数が増えたとしても総実行時間はそれほど急激には増えていないことがわかる。部品数 100 であっても総実行時間は 0.2 秒以下であり、解を求めるまでの速度は列挙法に比べて有効的な時間である。また部品数がより少ない場合に総配線長 Best の解は、列挙法による最適解とほぼ同等の値を得ている。試行回数を増やすことにより、精度の高い解を見つけ出せられると考えられる。

### 6.3 遺伝的アルゴリズムのパラメータ設定

表 6.2 逐次改善法で求めたときの総配線長と実行時間

部品数	総配線長 Avg	総配線長 Best	総実行時間 [sec]
4	21	21	0.000
6	35	35	0.005
8	58	57	0.007
10	83	73	0.014
12	106	96	0.019
20	186	172	0.034
30	375	347	0.039
40	563	528	0.045
50	802	761	0.050
60	945	886	0.061
70	1323	1247	0.070
80	1531	1473	0.076
90	1809	1671	0.116
100	2222	2050	0.187

### 6.3 遺伝的アルゴリズムのパラメータ設定

遺伝的アルゴリズムを扱うにあたり、各種のパラメータを定めた。遺伝的アルゴリズムには、親の選択、交叉方法、突然変異方法、交叉確率、突然変異確率というパラメータがある。パラメータ設定では、親の選択が 2 種類、交叉が 10 種類、突然変異が 3 種類、交叉確率と突然変異確率は 10%~90%までの 9 種類の確率と決め、部品配置問題での設定とした。よってすべての組合せは、(親の選択)10 × (交叉)10 × (突然変異)3 × (交叉確率)9 × (突然変異確率)9 から計 4860 通りある。4860 通りを部品数 100 としたときの問題から総配線長を求め、その値が最も小さいパラメータを部品数  $n$  すべての部品配置問題に適用した。このと



## 6.4 パラメータ設定の考察

き，染色体数と世代未更新限度数を 50 と設定して実験を行なった．

本研究で決定したパラメータ設定は以下の通りである．

- 親の選択：トーナメント戦略
- 交叉方法：一様順序交叉
- 突然変異方法：ランダムな 2 点の入れ換え操作を行なう突然変異
- 交叉確率：90%
- 突然変異確率：80%

## 6.4 パラメータ設定の考察

今回，ひとつ部品配置問題の問題例を挙げ，4860 通りの組合せから総配線長の小さいパラメータを選び出したが，必ずしもこのパラメータが部品配置問題に対して最高の組合せだとは限らない．部品数，配線数，配線接続数によっても各問題の特性が変化するため，問題それぞれに対してのパラメータ設定を行なうべきであったように考える．例えば，総配線長が小さくなる並びの部品配置であったとしても，突然変異確率 80%では，その並びが壊されてしまう可能性が高い．よってこの環境から新しく子が生まれたとしても，子の染色体は，親の染色体の性質を十分に受け取ることができない．

## 6.5 実験 3(遺伝的アルゴリズム)

遺伝的アルゴリズムから得られた実験結果を表 6.3 に示す．

遺伝的アルゴリズムは，表 6.3 から実行時間は部品数が増すにつれて，徐々に長くなっていることがわかる．逐次改善法に比べ解を求めるまでの時間は遅い．しかし，解の精度は列挙法に近い値を算出しており，部品数が増えたとしても総配線長 Avg が，逐次改善法の総配線長 Best とよりも小さい値を求められていることが確認できる．

## 6.6 各解法の評価

表 6.3 遺伝的アルゴリズムで求めたときの総配線長と実行時間

部品数	総配線長 Avg	総配線長 Best	総実行時間 [sec]
4	21	21	0.689
6	35	35	0.811
8	57	57	0.939
10	76	73	1.032
12	100	96	1.301
20	168	161	2.562
30	317	299	4.389
40	476	455	9.122
50	643	612	15.318
60	694	629	22.430
70	1036	992	33.835
80	1182	1149	41.262
90	1388	1312	58.331
100	1676	1592	69.113

## 6.6 各解法の評価

各手法の表から分かるように、列挙法は部品数が少ない場合であれば短時間で解を求めることが可能で最適解が保証されるが、部品数が多い場合は逐次改善法や遺伝的アルゴリズムに比べ時間を膨大に費やしてしまう。

逐次改善法に対しては、部品数が増大したとしても一定的な短時間で解を得ることができると共に、部品数がより少なければ列挙法と非常に近い解が得られるが、部品数が増えると遺伝的アルゴリズムに比べ解の精度が低い。

遺伝的アルゴリズムは、部品数が多くなるにつれ、逐次改善法より徐々に時間を費やすよ

## 6.6 各解法の評価

うになっていくが，逐次改善法より小さい総配線長の解を得ることが確認できた．また部品数が少なければ列挙法の近似解が算出される．実験から逐次改善法より小さい総配線長を得られたが，逐次改善法と同様に最適性が保証されないことから，総配線長が最適解であるということは確認できない．

## 第 7 章

# 結論

本研究では列挙法，逐次改善法，遺伝的アルゴリズムという 3 種類の解法を部品配置問題に対して適用し，3 種類の解法それぞれの性能について評価した。

部品数  $n$  が少ない場合は，最適解を得る列挙法や，逐次改善法が有効であることを確認した。しかし，部品数  $n$  が多くなる場合では，逐次改善法や遺伝的アルゴリズムと比較して，列挙法は膨大な時間を要してしまう。逐次改善法もまた，部品数  $n$  が増えるにつれて遺伝的アルゴリズムよりも解の精度が低い。それに対して，遺伝的アルゴリズムは部品数  $n$  が少ない場合は，列挙法とほぼ同等の精度を持ち，部品数  $n$  が多くなったとしても逐次改善法よりも小さい総配線長が得られる。

よって，部品配置問題に対する本実験で使用した 3 種類の手法の中で，最も利便性が高い解法は，遺伝的アルゴリズムであると本研究では評価した。

最適な部品配置を知りたい場合は列挙法に頼るしか術はないが，部品数  $n$  が膨大になると時間がかかり過ぎてしまうことを考えると現状では，遺伝的アルゴリズムの近似解に満足すべきであると判断した。

今後の課題としては，LSI の実装など実際の電子機器の部品配置に対して，これらの手法がどの程度有効であるのか検証する必要がある。

# 謝辞

本論文は、著者が2001年7月から2003年2月までの高知工科大学工学部情報システム工学科在学中に、同学科坂本研究室において行なった研究活動の成果を記したものである。

まず、本論文を著わすにあたって、プログラムの書式から就職活動まで、多忙の中全面的に御指導頂いた坂本明雄教授に深く感謝致します。

プログラム作成、輪講、 $\text{\LaTeX}$ 作成作業などご協力頂いた登伸一さん、折橋祐一さん、卒業論文の書き方等、私が気が付かなかった点を指摘して頂いた友池貴之さんに深く感謝致します。

4年間勉学を共にし、大学生活における悩みの相談や卒業研究のC言語のプログラム作成にご協力頂いた赤間寛君、西村章君、河内友彦君、萩山直樹君に心より感謝致します。

卒業研究のアドバイスや励みのお言葉を頂いた福本昌弘助教授、福本研究室のみなさまに心より感謝致します。

いつも輪講を頑張っている3年生、就職活動や卒業研究など忙しい大学生活になると思いますが頑張って下さい。

同期のみなさま、今まで御指導頂いた情報システム工学科の先生方に感謝の意を表します。

最後に高知工科大学工学部情報システム工学科のコンピュータ環境、並びに勉学の場を与えて頂いたことに感謝致します。

## 参考文献

- [1] 白石洋一訳. 組合せ最適化アルゴリズムの最新手法, 丸善株式会社, 2002
- [2] 北野宏明編. 遺伝的アルゴリズム, 産業図書株式会社, 1993
- [3] 川西朝雄著. C 言語によるはじめてのアルゴリズム入門, 技術評論社, 1992
- [4] <http://lecture.ecc.u-tokyo.ac.jp/>