

平成 12 年度
学士学位論文

遺伝的アルゴリズムを用いた
電子部品挿入順序問題

Insertion Ordering of Electronic Parts on
Printed Circuit Boards Using a Genetic Algorithm

1010428 登 伸一

指導教員 坂本 明雄

2001 年 2 月 5 日

高知工科大学 情報システム工学科

要 旨

遺伝的アルゴリズムを用いた 電子部品挿入順序問題

登 伸一

本論文では，電子部品の自動挿入機における挿入順序問題を定義し，これを遺伝的アルゴリズムを用いて解くための手法を提案する。

電子部品自動挿入機は，入力データとして挿入順序を与えれば，自動的に電子部品をプリント基板に挿入する機械である。しかし，指定した順序によっては挿入過程でロスタイムが生じる。

そこで遺伝的アルゴリズムの特徴である符号化や複製，交叉，突然変異などをどのように用いて電子部品挿入順序問題を解くかを提案する。さらに，実際の部品データを用いた計算機実験を行うことにより，提案手法の有効性を確認する。

キーワード 遺伝的アルゴリズム，電子部品挿入順序問題，プリント基板，ロスタイム

Abstract

Insertion Ordering of Electronic Parts on Printed Circuit Boards Using a Genetic Algorithm

Nobori Shin-ichi

In this paper, we define an insertion order problem in the automatic insertion machine of the electronic parts, and propose the technique to untie it using the genetic algorithm(GA).

An electronic parts automatic insertion machine is the machine which inserts an electronic parts into the printed circuit board automatically on input insertion order. But, loss-time is formed in the insertion process by the specified order.

So, we propose that it is decided how to untie an insertion ordering of electronic parts problem by using encoding which is the characteristics of genetic algorithms, and reproduction, crossover, mutation, and so on. Then, the validity of the proposal technique is reported by doing the computer experiment which actual data were used for.

key words Genetic Algorithm, Insertion Ordering of Electronic Parts Problem, Printed Circuit Board, Loss-time

目次

第 1 章	序論	1
第 2 章	電子部品挿入順序問題	3
2.1	問題の定義	3
2.2	問題の定式化	5
2.2.1	準備	5
2.2.2	電子部品挿入順序問題	7
2.3	ロスタイムに関する考察	8
2.3.1	一定範囲内のロスタイム	8
2.3.2	ロスタイム発生タイミング	9
2.4	まとめ	11
第 3 章	遺伝的アルゴリズム	12
3.1	概要	12
3.2	特徴	14
3.3	まとめ	15
第 4 章	遺伝的アルゴリズムにおける電子部品挿入順序問題	16
4.1	実際の流れ	16
4.2	符号化	17
4.3	適応度の計算	17
4.3.1	スケーリング	18
4.4	複製	18
4.5	交叉	19
4.5.1	Order crossover (OX)	19

目次

4.5.2	Partially mapped crossover (PMX)	20
4.5.3	Cycle crossover (CX)	21
4.6	突然変異	22
4.7	終了条件	22
4.8	まとめ	23
第 5 章	実験結果	24
5.1	実験 1	25
5.1.1	考察	28
5.2	実験 2	28
5.2.1	考察	29
5.3	実験 3	29
5.3.1	考察	30
5.4	実験 4	30
5.4.1	考察	31
5.5	実験 5	31
5.5.1	考察	32
5.6	各世代毎の適応度（最良，最悪，平均）の変化	32
5.7	まとめ	33
第 6 章	結論	34
	謝辞	35
	参考文献	36
付録 A	電子部品自動挿入機の仕様	37

目次

2.1	電子部品自動挿入機の概略図	4
2.2	部品挿入時におけるロスタイム発生条件	5
2.3	ヘッドの動作タイミング	9
3.1	GA の流れ	14
4.1	実際の流れ	16
4.2	染色体 α	17
4.3	1 point right OX	19
4.4	1 point left OX	19
4.5	2 point inside OX	20
4.6	2 point outside OX	20
4.7	1 point right PMX	20
4.8	1 point left PMX	21
4.9	2 point inside PMX	21
4.10	2 point outside PMX	21
4.11	CX	22
4.12	突然変異	22
5.1	部品数 165 , 品種数 23 の電子部品挿入順序問題	24
5.2	部品数 165 , 品種数 23 の問題における <i>losstim</i> 6 の部品挿入順序	32
5.3	各世代毎の適応度 (最良 , 最悪 , 平均) の変化	33

表目次

5.1	パラメータ	24
5.2	部品数 165, 品種数 23 の問題に対する実験 2 の結果	29
5.3	部品数 165, 品種数 23 の問題に対する実験 3 の結果	30
5.4	様々な部品数, 品種数の問題に対する実験 4 の結果	31
5.5	部品数 165, 品種数 23 の問題に対する実験 5 の結果	31

第 1 章

序論

電子機器の大量生産を支える技術の一つに、プリント基板の電子部品自動挿入機がある。電子部品自動挿入機は、どの電子部品をプリント基板のどこにどの順序で挿入するかをあらかじめプログラムしておけば、自動的に電子部品をプリント基板に挿入する機械である。しかし、指定した順序によっては挿入過程でロスタイムが生じる。したがって、この工程において最適な部品挿入順序を入力データとして与えれば、ロスタイムが発生することなくすべての部品を挿入することができ、工程数の低減および生産性の向上に寄与できる。

一方、生物の進化を模倣した最適化手法の一種として、遺伝的アルゴリズム (Genetic Algorithm) がある。このアルゴリズムは自然界における生物の進化モデル、すなわち世代を形成している個体の集合 (解候補) の中で、環境への適応度の高い個体が次世代により多く生き残り、また複製、交叉および突然変異を繰り返しながら次世代を形成していく過程を模した最適化法であり、より環境に適した適応度の高い個体、つまり最適解を得ようとするアルゴリズムである。

本論文では、電子部品の自動挿入機における挿入順序問題を定義し、その解法を提案する。電子部品挿入順序問題においては、[1] でハミルトン閉路探索問題に帰着し、ヒューリスティックな閉路探索法と遺伝的アルゴリズムを併用した解法が提案されている。この解法によれば、部品数 165、品種数 23 の実験データにおいてロスタイム 0 の解が求められているが、その計算時間に難点があった。そこで本論文では、二種類の評価関数を導入した遺伝的アルゴリズムによる解法を提案し、アルゴリズムの高速化を図った。さらに、提案する手法を用いて計算機実験を行ない、その有効性を述べる。

2 章以降の概要を以下に示す。

2章では、電子部品自動挿入機の構造、動作を説明した後、電子部品自動挿入順序問題を定式化する。3章では、遺伝的アルゴリズムについて述べる。4章では、遺伝的アルゴリズムにおける電子部品挿入順序問題の解法について説明する。そして5章で計算機実験による結果を示し、結論と今後の課題を6章で述べる。

第 2 章

電子部品挿入順序問題

電子部品自動挿入機（付録 A 参照）は、どの電子部品をプリント基板のどこにどの順序で挿入するかをあらかじめプログラムしておけば、自動的に電子部品をプリント基板に挿入する機械である。しかし、指定した順序によっては挿入過程でロスタイムが生じる。したがって、この工程において最適な部品挿入順序を入力データとして与えれば、ロスタイムが発生することなくすべての部品を挿入することができ、工程数の低減および生産性の向上に寄与できる。

本章では、電子部品の自動挿入機における挿入順序問題を定式化する。

2.1 問題の定義

電子部品自動挿入機は、図 2.1 に示すように電子部品を種類別に収納した“品種スロット”，部品を受け取り自動挿入する“ヘッド”，および X，Y 方向に移動可能な“プリント基板”から構成されている。そして、品種スロットから渡された電子部品がヘッド内を受端から先端まで移動する時間以内であれば、品種スロットが左右に移動しても、プリント基板が X および Y 方向に移動しても“ロスタイム”が発生しない構造になっている。具体的には、現在の位置から品種スロットは左右 2 スロット以内、プリント基板は X，Y 共に 50 [mm] 未満の移動という条件を同時に満たせば、ロスタイムが発生することなく部品の連続挿入が可能である。また、全部品を挿入し終えた完成基板から次の基板への切り替え時においても同様の条件が当てはまるため、部品挿入終了位置から品種スロットとヘッドそれぞれの初期位置への移動も考慮しなければならない。品種スロットの初期位置は左端のスロット 0，

2.1 問題の定義

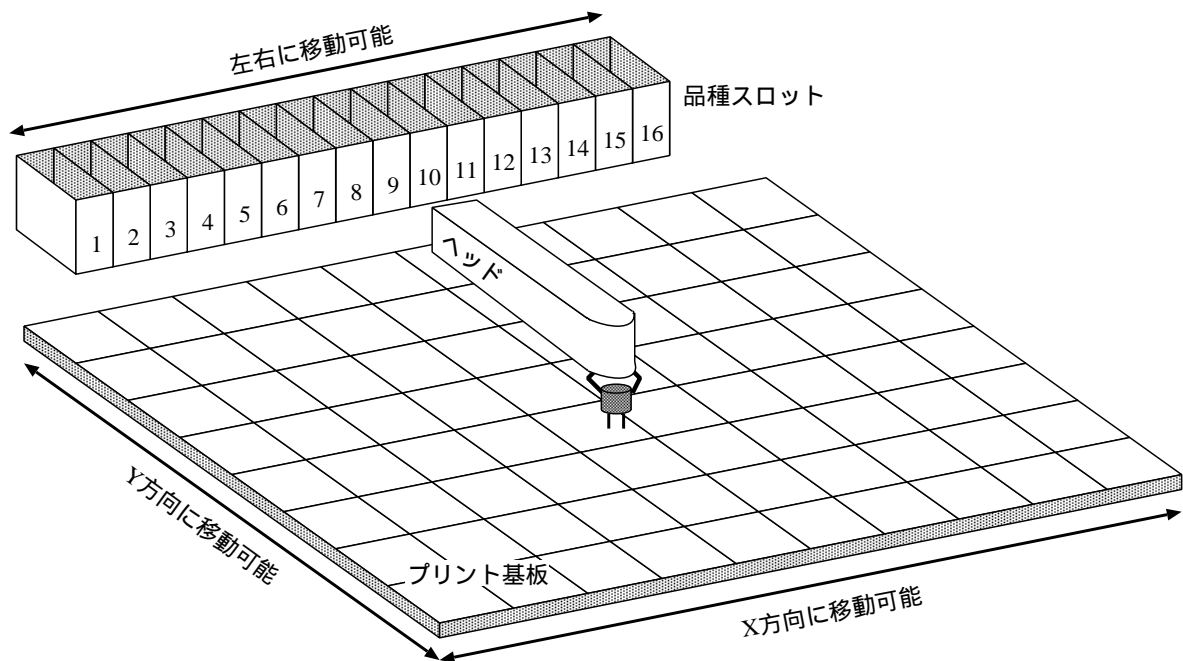


図 2.1 電子部品自動挿入機の概略図

ヘッドはプリント基板左下の基準穴を基準とした絶対座標で座標 (16500, 21000) の位置である。

図 2.2 は部品挿入時におけるロスタイム発生条件を表した図である。“ ”, “x” はスロット, 基板それぞれの“現在”からの移動条件を表し, “ ” はロスタイムが発生することなく連続挿入可能, “x” は不可能を表す。今, ヘッドが品種“F”の部品をスロットから受取り挿入しているとすると, 次に扱う部品として図中“C”はスロットの移動条件のため, “A”および“G”は基板の移動条件のために, “F”からロスタイムが発生することなく部品を連続挿入することが不可能である。しかし“F”の次に“D”を挿入すれば, “D”はスロットと基板の移動条件を同時に満たすため, ロスタイムは発生しない。

電子部品挿入順序問題とは, すべての部品挿入において上述した機械的制約に起因するロスタイムが発生しないように, “どの品種をどのスロットに収納するか” および “どういった順序で部品を挿入するか” を決定する問題である。

2.2 問題の定式化

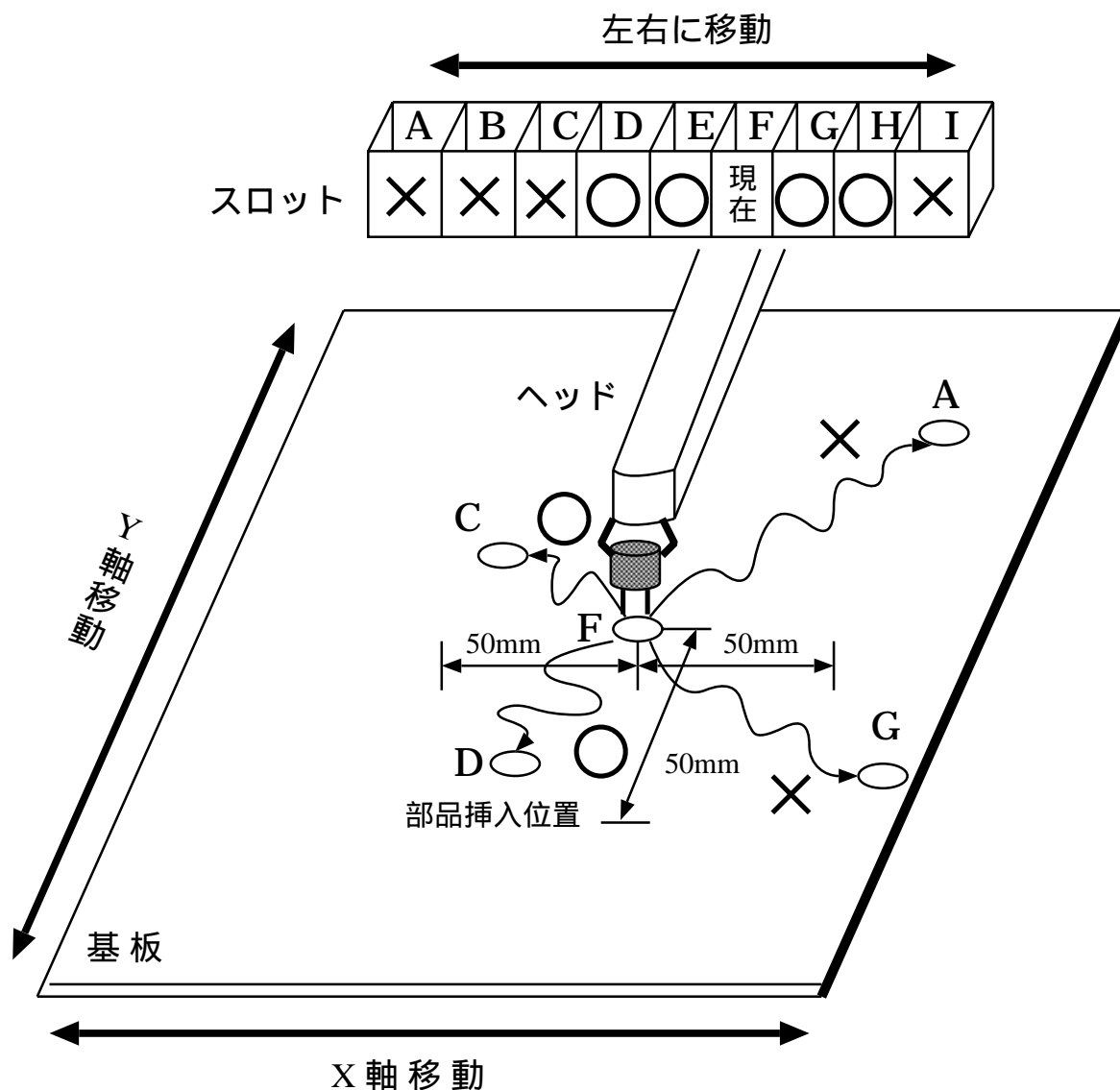


図 2.2 部品挿入時におけるロスタイム発生条件

2.2 問題の定式化

2.2.1 準備

基板上の座標 (x, y) へ品種 p の部品を挿入する要求があるとき, x, y および p をまとめたものを部品データとよび, $v = (x, y, p)$ と表す。また, i 番目の部品データ v_i を $v_i = (x_i, y_i, p_i)$ と書く。なお, 電解コンデンサやトランジスタなどの有極部品については,

2.2 問題の定式化

同一の部品であってもその挿入方向に応じて異なる品種とみなす場合がある。また、 $i \neq j$ のときは $x_i \neq x_j$ かつ $y_i \neq y_j$ である。

部品挿入の開始時と終了時に、基板が一定の位置にあり、かつ、そのときのスロット位置は左端でなければならない。このことを考慮するために、特別な部品データ $v_0 = (x_0, y_0, p_0)$ を用意しておく。ここで、 x_0 と y_0 はあらかじめ与えられた定数 $(x_0, y_0) = (16500, 21000)$ であり、 p_0 は実在しない品種である。

以下では特に断らない限り、部品データは v_0 を含めて $n + 1$ 個あるものとする。部品データの集合を v_0 を含めて V で表す。各部品データのある順序づけを一般に π で表す。すなわち、 π は集合 V の各要素を 0 から n のいずれかの整数に対応させる全単射であるが、いかなる π に対しても $\pi(v_0) = 0$ であるとする。

また、すべての品種の集合を p_0 を含めて P で表し、その要素数を $q + 1$ とする。すなわち、 $p_i \in P$ ($i = 0, 1, 2, \dots, n$) であり、 $|P| = q + 1$ である。ただし、使用する自動挿入機の制限から $q \leq q_{max}$ であるものとする。なお、 q_{max} のデフォルト値は 40 である。

各品種のスロット割当を一般に σ で表す。すなわち、 σ は集合 P の各要素を 0 から q のいずれかの整数に対応させる全単射であるが、いかなる σ に対しても $\sigma(p_0) = 0$ であるものとする。

二つの部品データ v_i と v_j の間の距離を次式で定義する。

$$dis(v_i, v_j) = \max(|x_i - x_j|, |y_i - y_j|)$$

このように定義された距離は、次の距離の公理を満足する。

- (1) $dis(v_i, v_j) \geq 0$ ($\forall i, j$) , 特に $dis(v_i, v_j) = 0 \Leftrightarrow v_i = v_j$
- (2) $dis(v_i, v_j) = dis(v_j, v_i)$ ($\forall i, j$)
- (3) $dis(v_i, v_j) \leq dis(v_i, v_k) + dis(v_k, v_j)$ ($\forall i, j, k$)

δ_d および δ_s を与えられた定数とするとき、二つの部品データの間に関係を定義する。ただし、 δ_d はロスタイムが発生しない基板移動距離で、デフォルト値は 50 [mm] であ

2.2 問題の定式化

る。また, δ_s はロスタイムが発生しないスロット移動距離で, デフォルト値は 2 である。

$$v_i \Delta v_j \Leftrightarrow \text{dis}(v_i, v_j) < \delta_d$$

$$v_i \sigma v_j \Leftrightarrow v_i \Delta v_j \text{ and } |\sigma(p_i) - \sigma(p_j)| \leq \delta_s$$

品種のスロット位置を無視したとき, $v_i \Delta v_j$ であればロスタイムが発生することなく v_i の次に v_j を挿入できることを意味している。また, 品種のスロット割当 σ が与えられているとき, $v_i \sigma v_j$ であればロスタイムが発生することなく v_i の次に v_j を挿入できることになる。

数学的には, 上で定義した二つの関係は, 共に反射律と対称律を満足する。すなわち, 任意の v について, $v \Delta v$ かつ $v \sigma v$ が成立する。また, $v_i \Delta v_j$ であれば $v_j \Delta v_i$ であり, $v_i \sigma v_j$ であれば $v_j \sigma v_i$ である。しかし, 推移律は満足しない。つまり, $v_i \Delta v_j$ かつ $v_j \Delta v_k$ であっても $v_i \Delta v_k$ であるとは限らない。

2.2.2 電子部品挿入順序問題

電子部品挿入順序問題とは, n 個の部品データ v_1, v_2, \dots, v_n が与えられたとき, 次の条件を満足するスロット割当 $\sigma : P \rightarrow \{0, 1, 2, \dots, q\}$ と部品データの順序づけ $\pi : V \rightarrow \{0, 1, 2, \dots, n\}$ を求めることである。

[条件]

$0 \leq i \leq n-1$ であるすべての i に対して関係 $\pi^{-1}(i) \sigma \pi^{-1}(i+1)$ が成立し, かつ, $\pi^{-1}(n) \sigma \pi^{-1}(0)$ である。

2.3 ロスタイムに関する考察

2.3.1 一定範囲内のロスタイム

2.2節では、 δ_d および δ_s をそれぞれロスタイムが発生しない基板移動距離およびロスタイムが発生しないスロット移動距離としたが、ここではさらに、 δ_d^* ($> \delta_d$) および δ_s^* ($> \delta_s$) をそれぞれ一定範囲内のロスタイムで収まる移動距離とする。

すなわち、

$$\delta_d \leq \text{dis}(v_i, v_j) < \delta_d^*$$

であれば、基板の移動距離が δ_d 以上であるため、部品データ v_i の基板上への挿入が終了してから次の部品データ v_j の挿入が始まるまでの間にロスタイムが生じることになる。しかし、その移動距離は δ_d^* 未満であるから、ロスタイムは一定範囲内に収まる。

同様に、

$$\delta_s < |\sigma(p_i) - \sigma(p_j)| \leq \delta_s^*$$

のときは、スロットの移動距離が δ_s より大きく δ_s^* 以内であるため、ヘッドがスロットから p_i の部品を受け取ってから次の p_j を受け取るまでの間に、一定範囲内のロスタイムが生じることになる。

ここで、“一定範囲内のロスタイム” という表現をしたのは、ロスタイムと基板上の移動距離あるいはスロットの移動距離との関係が必ずしも明確ではないことによる。今後、総ロスタイムなどは、上記の一定範囲内のロスタイムの整数倍と考えることにし、連続する部品データの挿入において、一定範囲を超えるロスタイムは許されないものとする。

以後の議論を簡単にするため、二つの部品データ v_i と v_j の間の基板ロス $\text{pwd}(v_i, v_j)$ およびスロットロス $\text{slot}(v_i, v_j)$ をそれぞれ次式のように定義する。

$$\text{pwd}(v_i, v_j) = \begin{cases} 0 & \text{if } \text{dis}(v_i, v_j) < \delta_d \\ 1 & \text{if } \delta_d \leq \text{dis}(v_i, v_j) < \delta_d^* \\ \infty & \text{otherwise} \end{cases}$$

2.3 ロスタイムに関する考察

$$slot(v_i, v_j) = \begin{cases} 0 & \text{if } |\sigma(p_i) - \sigma(p_j)| \leq \delta_s \\ 1 & \text{if } \delta_s < |\sigma(p_i) - \sigma(p_j)| \leq \delta_s^* \\ \infty & \text{otherwise} \end{cases}$$

2.3.2 ロスタイム発生時のタイミング

基板移動距離	ヘッドの動作	スロット移動距離
	p_{i-1} を挿入する	$ \sigma(p_{i-1}) - \sigma(p_i) $
$dis(v_{i-1}, v_i)$	p_i を受け取る	
	p_i : 受端 \rightarrow 先端	$ \sigma(p_i) - \sigma(p_{i+1}) $
	p_i を挿入する	
$dis(v_i, v_{i+1})$	p_{i+1} を受け取る	
	p_{i+1} : 受端 \rightarrow 先端	$ \sigma(p_{i+1}) - \sigma(p_{i+2}) $
	p_{i+1} を挿入する	
$dis(v_{i+1}, v_{i+2})$	p_{i+2} を受け取る	

図 2.3 ヘッドの動作タイミング

ヘッドは、まずスロットから部品を受け取り、部品がヘッド内を受端から先端まですべる間に基板が操作位置へ移動してその部品を挿入し、またスロットから次の部品を受け取る。一方スロットは、ヘッドに部品を渡したあと次の部品の操作が始まるまでに、次の部品の入っているスロットがヘッドの部品受渡し位置にくるように移動しておく必要がある。この移動距離が δ_s 以内であればロスタイムは発生しないが、 δ_s を超えて δ_s^* 以内であれば、ヘッドが部品を受け取る時刻になっているのにスロットの移動が完了していない状況になり、一定範囲内のロスタイムが生じる。同様に基板の移動は、部品の挿入終了時から次の部品の挿入開始時まで完了しておく必要があり、その移動距離が δ_d を超えるとロスタイムが生じる。これらの時間的な関係は図 2.3 のようになる。ただし、図中の $v_i = (x_i, y_i, p_i)$ は i 番目に挿入する部品データである。

部品データ v_i を処理するまでにロスタイムは生じていないものとし、 v_i と v_{i+1} の間の

2.3 ロスタイムに関する考察

スロットロスが 1 の場合，すなわち，スロット移動において一定範囲内のロスタイムが生じる場合を考えてみる。このとき，もし v_i と v_{i+1} の間の基板ロスも 1 であったとしても，基板移動のロスタイムは p_{i+1} を受け取る時刻を一定範囲内遅らせることで吸収される。逆の場合も同様であり，基板ロスが 1 であったとき，その直後のスロットロスが 1 であってもそのロスタイムは，部品を挿入する時刻を一定範囲内遅らせることで吸収できる。

以上の考察から，スロット割当と部品挿入順序が決まれば，ロスタイム発生のタイミングを考慮した厳密な意味での総ロスタイム『*losstime*』は次のようにして計算できる。

[*losstime* の計算]

部品挿入順序が $v_0, v_1, v_2, \dots, v_n$ であるとする。スロットロスと基板ロスを交互に並べた次の列 λ を考える。

$$\lambda = \text{slot}(v_0, v_1), \text{pwd}(v_0, v_1), \text{slot}(v_1, v_2), \text{pwd}(v_1, v_2), \dots \\ \dots, \text{slot}(v_{n-1}, v_n), \text{pwd}(v_{n-1}, v_n), \text{slot}(v_n, v_0), \text{pwd}(v_n, v_0)$$

ただし，列 λ には 0 または 1 しか現われないものとする。すなわち，基板ロスおよびスロットロスが ∞ となるような二つの部品データを連続して処理することはないものとする。この部品挿入順序における *losstime* は，列 λ を左から調べ，連続する二つの 1 がある場合は右側の 1 を 0 に置き換える操作を順次施して得られた列における 1 の総和に等しい。

上記の *losstime* の計算法によれば，厳密な総ロスタイムが求められるが，この値は連続する部品データ v_i と v_{i+1} の関係のみからは決定できない。そこで，二つの部品データ v_i と v_j の間の $\text{loss}(v_i, v_j)$ を

$$\text{loss}(v_i, v_j) = \begin{cases} 0 & \text{if } \text{pwd}(v_i, v_j) + \text{slot}(v_i, v_j) = 0 \\ 1 & \text{if } \text{pwd}(v_i, v_j) + \text{slot}(v_i, v_j) = 1 \text{ or } 2 \\ \infty & \text{otherwise} \end{cases}$$

と定義する。

2.4 まとめ

部品挿入順序が $v_0, v_1, v_2, \dots, v_n$ であるとき, その簡易ロスタイムを

$$loss(v_0, v_1) + loss(v_1, v_2) + \dots + loss(v_{n-1}, v_n)$$

と定める。この値は, 厳密な総ロスタイムより大きくなることもある。しかし, 簡易ロスタイムが 0 ならば厳密な総ロスタイムも 0 である。

本研究では, $losstime$ を計算するのに簡易ロスタイムを

$$loss(v_i, v_j) = \begin{cases} 0 & \text{if } pwd(v_i, v_j) + slot(v_i, v_j) = 0 \\ 1 & \text{if } pwd(v_i, v_j) + slot(v_i, v_j) = 1 \text{ or } 2 \\ \vdots & \\ n & \text{if } pwd(v_i, v_j) + slot(v_i, v_j) = n \text{ or } n + 1 \end{cases}$$

と定義し, 部品挿入順序が $v_0, v_1, v_2, \dots, v_n$ であるとき,

$$loss(v_0, v_1) + loss(v_1, v_2) + \dots + loss(v_{n-1}, v_n)$$

と定め, この簡易ロスタイムを与えられた部品挿入順序の $losstime$ と考えることにする。

2.4 まとめ

本章で述べたように, 簡易ロスタイムが 0 ならば厳密な総ロスタイムも 0 である。本研究では, 部品挿入順序とスロット割当ともに $losstime$ が 0 を満たすために遺伝的アルゴリズムを用いての解法を提案する。次章以降で, 遺伝的アルゴリズム, 遺伝的アルゴリズムにおける電子部品挿入順序問題について説明する。

第3章

遺伝的アルゴリズム

遺伝的アルゴリズム (Genetic Algorithm : GA) [2][3][4][5] は、生物進化 (選択淘汰・突然変異) の原理に着想を得たアルゴリズムである。すなわち、生物の進化と同様にその世代を形成している個体の集合 (解候補) を用い、その中で環境への適応度の高い個体を次世代に多く残し、また交叉や突然変異を発生させながら次世代を形成させる。それにより、より環境に適した適応度の高い個体が生まれ出されていく。遺伝的アルゴリズムはこのような過程を繰り返すことによって最適な解を得ようとする探索アルゴリズムであり、大域的かつ並列的な探索手法である。

本章では、遺伝的アルゴリズムの流れについて簡単に説明する。

3.1 概要

一般的に生物は環境にうまく適応できると、生命の存続と繁殖が可能となる。繁殖を行う際には、その個体が持っている生物としての設計情報を何らかの形で子孫に伝授しなければならない。そのような設計情報は染色体の特定の位置に存在する遺伝子に書き込まれている。

遺伝子によって生物としての情報伝達が親から子へ行われるが、高等生物の繁殖は有性生殖であり、この場合は父方と母方の遺伝子が混ざり合ったものを遺伝子として受け継ぐことになる。この過程は交叉と呼ばれている。一方、遺伝子のコピーを行う際などに微妙なエラーが生じることがあり、これは突然変異と呼ばれる。次の世代には、各個体の中でもより優れた、つまり環境への適応度の高い個体の遺伝子情報が優先的に伝えられる。適応度の低

3.1 概要

い個体は短命であったり，増殖できなかつたりするからである。同時に適応度の低い種族は自然淘汰されていく。このような原理に基づいて世代を重ねていくと，次第に環境への適応度が高い個体が多くなっていく。これが遺伝と進化の基本的な原理である。

自然界での個体を特徴づけているものは，遺伝子が一定の順序で配列している染色体であり，遺伝的アルゴリズムでは有限固定長の記号列を染色体に，個々の記号を遺伝子に対応させている。

$M(t)$ は m 個の個体から成る生物集団， t を世代としたとき， $M(t)$ は同一の形質をもつ複数の個体が存在するため多重集合になる。自然界の生物は複数の染色体をもつが，このアルゴリズムでは個体 i は 1 つの染色体 α_i をもつものとする。 α_i ($i = 1, 2, \dots, m$) は，集合 Z の要素である n 個の遺伝子 Z_{ij} の並び $\alpha_i = (Z_{i1}, Z_{i2}, \dots, Z_{in})$ によって作られている。

また，染色体の集合を遺伝子空間 Z^n と呼び，染色体上で遺伝子の置かれる位置を遺伝子座，そこに配置され得る遺伝子に対立遺伝子，遺伝子の配列からなる染色体を遺伝子型，その遺伝子の並びによって定まる発現を表現型と呼ぶ。ただし，最適化問題を解く場合には，遺伝子型と表現型は 1 対 1 に対応しているものとする。

このアルゴリズムにおいて，生物（個体）がどれだけ環境に適応しているのかの度合は適応度と呼ばれ，自然淘汰に対する個体の有利さを表す尺度になっている。これは個体が次の世代に残す子孫の数に対応しており，このような適応度を個体に対応させる関数は，適応関数と呼ばれている。

遺伝的アルゴリズムは，図 3.1 のように大きく 4 つの過程，初期化，適応度の計算と複製，交叉，そして突然変異に分けられる。以下ではこれらの過程について説明する。

- (1) 初期化： $t = 1$ の生物集団のすべての個体に対して染色体を初期化する過程である。
- (2) 適応度の計算と複製：世代 t の個体群中の各個体 i の遺伝子型の適応度 $f(i)$ を計算する。生物集団 $M(t)$ の適応度の総和を，

$$F = \sum_{i=1}^m f(i)$$

とし， $M(t)$ からランダムに重複を許して新たに m 個の個体を選び出し，次の世代の両

3.2 特徴

親となる生物集団 $M'(t)$ を作る。この時、個体 i が選ばれる確率は $f(i) / F$ であり、適応度の高い個体ほど多く選ばれる。

(3) 交叉：複製の過程によって作られた生物集団 $M'(t)$ に対して $m / 2$ 個の個体の対を作り、あらかじめ定められた生起確率 p_c で、各個体の対ごとに一様分布に従う乱数によってランダムに分解位置を定め、染色体を入れ換える。

(4) 突然変異： $M'(t)$ の各個体について 1 つの遺伝子座をランダムに選び出し、他の対立遺伝子に変更して次の世代の生物集団 $M(t + 1)$ を作る。この過程は、生物集団にあらかじめ定められた生起確率 p_m で遺伝子ごとに生起する。

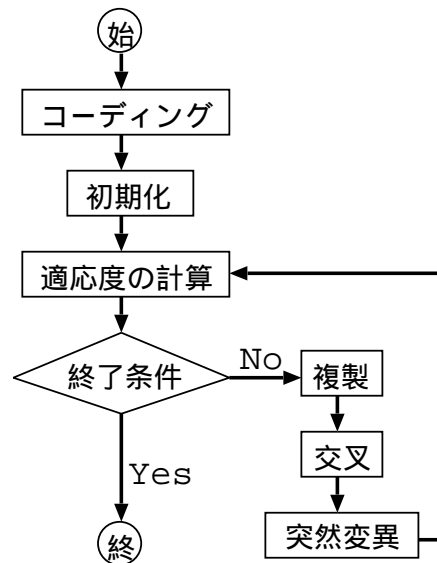


図 3.1 GA の流れ

3.2 特徴

遺伝的アルゴリズムは、単独の個体ではなく生物集団を用いること、および個体の適応度（適応関数の値）自体を用いることに特徴がある。状態空間における単独の点ではなく、状態空間の広い範囲にわたって散在する点の集合について、適応度を求め、最適値に近い値を与える点の集合を再構成し、交叉や突然変異によって次の点の集合を作るという操作を繰り返す。

3.3 まとめ

返している。このアルゴリズムは、最適化問題における局所解からの脱出という課題を点の集合を用いることによって解決しているのである。つまり、遺伝的アルゴリズムは、局所解に陥る可能性が低いアルゴリズムといえる。

3.3 まとめ

本章では、遺伝的アルゴリズムについて簡単に述べた。次章では、電子部品挿入順序問題に対し、遺伝的アルゴリズムをどのように用いたかについて述べる。

第 4 章

遺伝的アルゴリズムにおける電子部品挿入順序問題

3 章で述べた遺伝的アルゴリズム（以下，GA）を電子部品挿入順序問題の最適化を行うために用いる。本研究では，GA の符号化，適応度の計算，複製，交叉および突然変異を本章で述べるように行ない最適解を求めた。

4.1 実際の流れ

本論文で提案する，電子部品挿入順序問題における GA の流れを説明する。

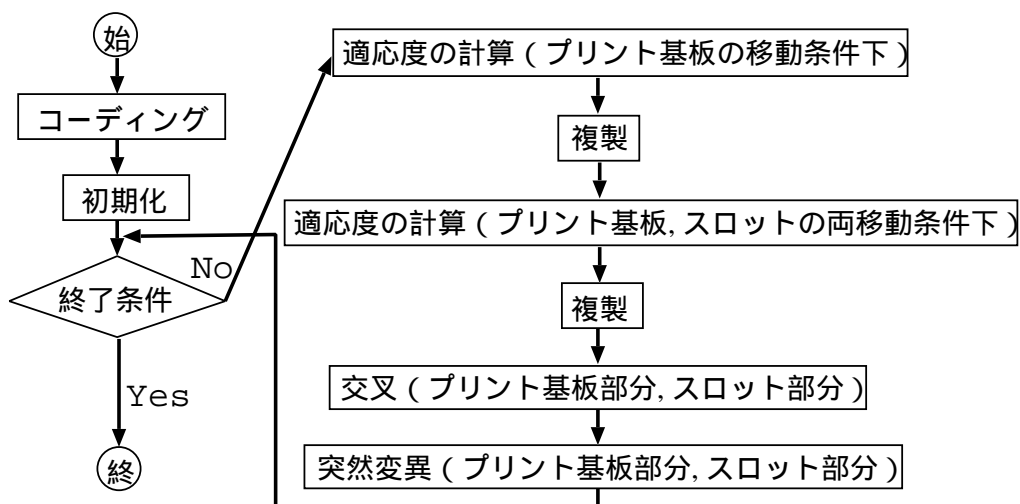


図 4.1 実際の流れ

1. 乱数を用いて，初期染色体集団を発生する。

4.2 符号化

2. プリント基板の移動条件のみを考え、各染色体が表現する部品挿入順序におけるロスタイムを算出する。
3. 適応度を求め、適応度比例戦略により複製を行う。
4. スロット、プリント基板の両移動条件を考え、各染色体が表現する部品挿入順序および品種のスロット割当におけるロスタイムを算出する。
5. 求めたロスタイムから適応度を算出し、複製、交叉、突然変異の遺伝的操作を行う。
6. 終了条件が満たさなければ、2以降を繰り返す。

終了後、得られた集団中で最も適応度の高い染色体をデコードし、部品挿入順序およびスロット割当を得る。

以下では、各部位での働きについて説明する。

4.2 符号化

図 4.2 に示すように染色体 α は、部品 V の挿入順序 π を表す v_0 を先頭とする順列と品種 P へのスロット割当 σ を表す p_0 を先頭とする順列を連結させた符号化法を採用する。その長さは挿入開始地点の部品 v_0 、品種 p_0 を含めて $|V| + |P|$ である。また、遺伝子 v_0, p_0 は交叉、突然変異においては無視され、常に各部位の先頭に配置される。

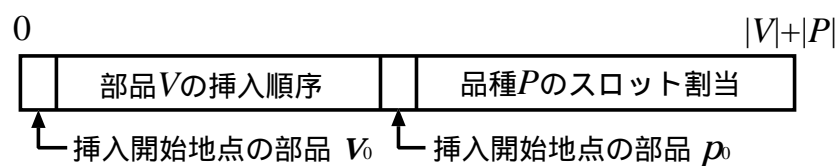


図 4.2 染色体 α

4.3 適応度の計算

染色体 α より、以下を求める。

4.4 複製

1. プリント基板の移動条件のみを考え、各染色体が表現する部品挿入順序におけるロスタイム
2. スロット、プリント基板の両移動条件を考え、各染色体が表現する部品挿入順序および品種のスロット割当におけるロスタイム

1, 2 それぞれの時、個体 i の適応度 $f(i)$ を

$$f(i) = \frac{1}{\text{losstime} + 1}$$

とする。

個体の適応度は、その表現型である部品挿入順序および品種のスロット割当の総 losstime が小さいほど高くなる。つまり、総 losstime が 0 の時 $f(i) = 1$ となり、最も適応度が高くなる。

4.3.1 スケーリング

適応度が決定されたとき、何らかの関数を導入し適応度の違いを拡大または縮小させることがある。このような関数を導入することを、スケーリングと呼ぶ。本研究では、べきスケーリングを用いる。

べきスケーリングは、もとの適応度を f 、新たな適応度を f' とすると、

$$f' = f^3$$

で表わされる。

4.4 複製

より適応度の高い個体が次の世代の集団に高い確率で選ばれるという、適応度比例戦略による複製を行なう。

4.5 交叉

本手法では符号化に順列表現を用いているため、通常の GA の交叉（二つの染色体の交叉部位の交換）を用いることはできない。交叉により順列表現が壊れ、致死遺伝子が発生するからである。そこで本手法では交叉手法として、Order crossover (OX), Partially mapped crossover (PMX), Cycle crossover (CX)[6] の 3 種類の手法を用いる。以下では、各交叉手法について説明する。なお、交叉は染色体の部品挿入順序部、スロット割当部の各部内で行なう。

4.5.1 Order crossover (OX)

OX は、親染色体の交叉箇所をもう一方の親染色体の遺伝子順序に組み換え、子染色体を生成する交叉である。図 4.3 に、交叉法 1 point right OX の場合の親、子染色体を示す。図中、“*” はランダムに選ばれた交叉点である。

Parent 1: e d a f c h b g	→	Offspring 1: e d a b c f g h
*		*
Parent 2: a b c d e f g h		Offspring 2: a b c e d f h g

図 4.3 1 point right OX

図 4.3 に示されるように、1 point right OX により、子 1 の交叉点より前半部は親 1 の遺伝子をそのまま継承し、後半部は親 1 の遺伝子順序に組み換えられる。子 1 に対しても同様である。以下に、他の交叉法での親、子染色体を示す。

Parent 1: e d a f c h b g	→	Offspring 1: a d e f c h b g
*		*
Parent 2: a b c d e f g h		Offspring 2: a c b d e f g h

図 4.4 1 point left OX

4.5 交叉

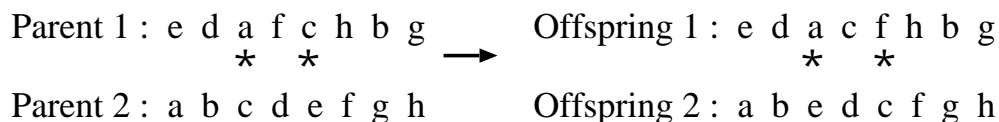


図 4.5 2 point inside OX

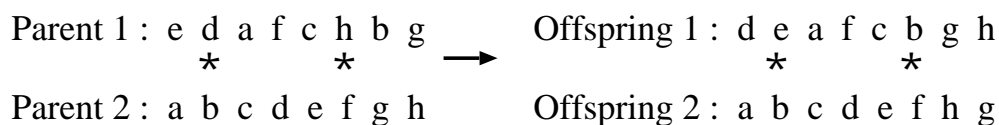


図 4.6 2 point outside OX

4.5.2 Partially mapped crossover (PMX)

PMX は、親染色体の交叉箇所を子染色体に継承し、致死遺伝子が発生しないように部分的にパッチをあてる交叉である。図 4.7 に、交叉法 1 point right PMX の場合の親、子染色体を示す。

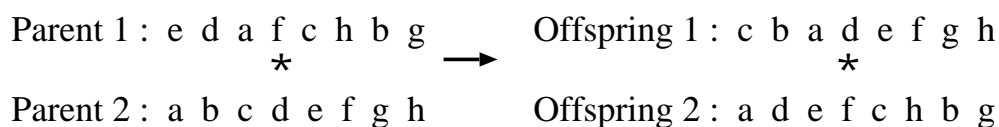


図 4.7 1 point right PMX

図 4.7 に示されるように、1 point right PMX により、子 1 の交叉点より後半部は親 2 の交叉箇所をそのまま継承する。前半部は親 1 の遺伝子を継承するが、その後、子 1 に致死遺伝子が発生しないように親 1 の親 2 に対する写像をあてる。例えば図 4.7 の場合、遺伝子 a はそのまま親 1 から継承しても致死遺伝子とはならない。しかし、遺伝子 d は致死遺伝子と

4.5 交叉

なるので，dの親2に対する写像である遺伝子bを継承する。同様に，遺伝子eに対し親2に対する写像aを継承するが，これも致死遺伝子となるのでさらに写像をたどり，aに対する写像遺伝子cを継承する。子2に対しても同様の操作が行なわれる。以下に，他の交叉法での親，子染色体を示す。

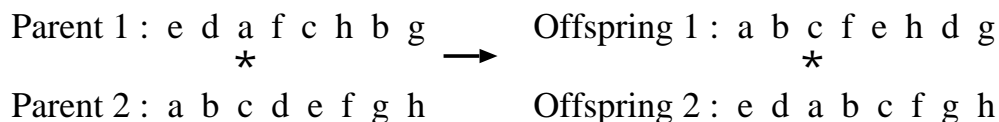


図 4.8 1 point left PMX

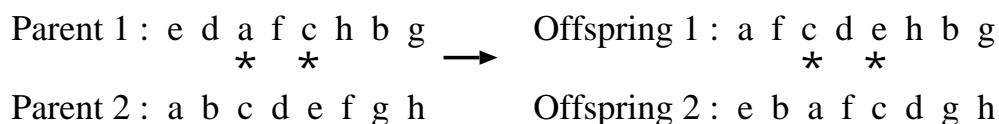


図 4.9 2 point inside PMX

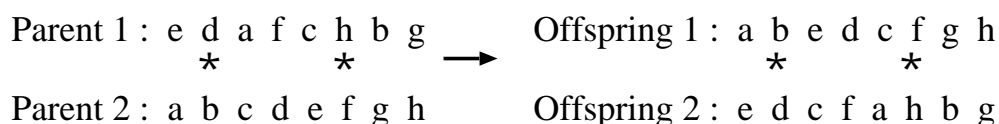


図 4.10 2 point outside PMX

4.5.3 Cycle crossover (CX)

CX は，両親染色体の遺伝子の cycle を求め，その遺伝子を子染色体の各遺伝子座に継承する。図 4.11 に示すように，親1の遺伝子座1の遺伝子を cycle の開始点として親2への写像をたどり，e a c e という cycle を得る。これが子1の各遺伝子座に継承され

4.6 突然変異

る。次に、親 2 の遺伝子座 2 の遺伝子を cycle の開始点として、b d f h g b の cycle を得、子 1 の各遺伝子座に継承する。この操作は cycle が得られなくなるまで繰り返され、子 2 に対しても同様の操作が行なわれる。

CX により、子染色体は両親染色体の遺伝子をそれぞれ 0.5 の確率で継承する。しかし、子染色体が両親染色体と同じ染色体になる可能性もある。

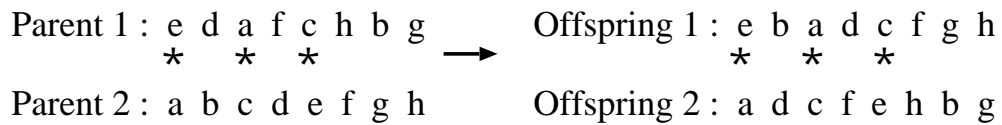


図 4.11 CX

4.6 突然変異

図 4.12 に示すように、染色体の遺伝子をランダムに 2 点選び交換する、最も単純な突然変異を用いる。なお、突然変異は染色体の部品挿入順序部、スロット割当部の各部内で行なう。

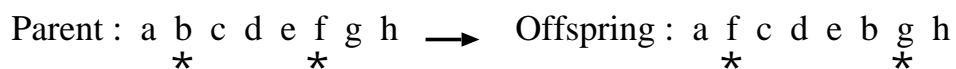


図 4.12 突然変異

4.7 終了条件

最良解を得てから T_{stop} 世代にわたって解が更新されないとき終了するものとする。

4.8 まとめ

以上により，電子部品挿入順序問題を解くことができる。つまり，提案する GA のアルゴリズムでは，二種類の適応度による複製・淘汰を行なった染色体に対し，交叉と突然変異をそれぞれ部品挿入順序部とスロット割当部で行なうことにより解に幅を持たせ，良い解を発見しやすくしている。

したがって GA を用いることにより，初期世代では *losstime* の大きい解しか得られないが，複製，交叉（部品挿入順序部，スロット割当部），突然変異（部品挿入順序部，スロット割当部）を繰り返しながら世代が進むにつれ，部品挿入順序とスロット割当の *losstime* が小さい，もしくは 0 の解が得られる。次章では，その実験結果を示す。

第 5 章

実験結果

4 章で述べた遺伝的アルゴリズムを用いた手法で電子部品挿入順序問題を解くため、すなわち最適解 ($losstime = 0$) を得るために計算機実験を行った。ここで、実験で使用するパラメータを表 5.1 に示し、図 5.1 に部品数 165、品種数 23 の電子部品挿入順序問題を示す。以下に Celeron (634MHz) による計算機実験の結果を示す。

表 5.1 パラメータ

個体数	M
世代終了数	T_{stop}
交叉確率	p_c
突然変異確率	p_m
交叉手法	OX, PMX, CX



図 5.1 部品数 165、品種数 23 の電子部品挿入順序問題

5.1 実験 1

5.1 実験 1

各交叉方法での交叉確率 p_c , 突然変異確率 p_m がどの程度が良い解を発見できるかを決定するために , $M = 300$, $T_{stop} = 300$ として乱数系列を変えて各 10 回の試行における最良解 ($losstime$) を求めた。ここで , \bullet は各交叉方法での最良解である。

1. Order crossover (1 point right)

$p_c \setminus p_m$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	68	61	134	172	176	185	201	201	209
0.2	66	55	130	171	185	192	209	208	201
0.3	58	\bullet 45	156	158	183	201	206	217	222
0.4	65	106	160	171	201	213	211	216	217
0.5	69	147	175	174	206	214	226	205	224
0.6	57	138	165	195	200	216	220	236	222
0.7	66	163	182	199	219	225	227	233	219
0.8	123	160	211	216	230	218	233	239	224
0.9	115	200	208	218	233	228	242	240	218

2. Order crossover (1 point left)

$p_c \setminus p_m$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	69	61	135	162	182	181	202	203	209
0.2	82	65	113	174	181	196	191	208	206
0.3	68	79	138	175	192	181	207	212	209
0.4	79	70	160	185	196	200	209	220	222
0.5	61	158	147	189	203	217	215	215	219
0.6	\bullet 49	161	184	202	197	223	210	224	238
0.7	119	157	193	205	217	215	230	242	236
0.8	71	168	207	217	199	223	233	237	232
0.9	104	180	213	215	234	231	231	240	241

5.1 実験 1

3. Order crossover (2 point inside)

$p_c \setminus p_m$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	88	94	137	158	178	196	189	195	215
0.2	83	95	114	161	170	198	190	218	211
0.3	76	● 68	152	161	186	197	208	197	226
0.4	70	88	159	182	187	204	210	219	224
0.5	83	111	157	190	202	206	210	210	212
0.6	80	109	149	189	195	213	220	218	233
0.7	98	102	155	170	191	211	230	227	224
0.8	98	96	138	181	208	220	230	216	232
0.9	97	82	171	198	182	206	223	233	237

4. Order crossover (2 point outside)

$p_c \setminus p_m$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	77	100	132	164	184	194	200	205	212
0.2	83	44	144	164	180	201	206	216	225
0.3	65	90	162	179	199	211	209	218	224
0.4	71	149	178	195	204	198	223	223	231
0.5	67	● 60	166	204	220	218	227	232	227
0.6	71	160	189	209	216	217	236	239	237
0.7	73	169	167	201	203	232	234	237	239
0.8	92	153	202	229	217	238	234	212	227
0.9	87	68	204	217	231	231	230	240	248

5. Partially mapped crossover (1 point right)

$p_c \setminus p_m$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	70	69	68	94	124	165	177	180	207
0.2	62	53	56	62	49	136	178	183	201
0.3	71	54	51	45	62	117	156	160	192
0.4	58	60	45	56	62	99	120	173	189
0.5	71	52	58	48	50	53	124	166	177
0.6	61	53	53	57	45	51	92	148	182
0.7	70	112	51	43	45	50	59	96	163
0.8	63	49	54	● 33	44	46	46	157	162
0.9	63	53	51	41	42	49	80	142	184

5.1 実験 1

6. Partially mapped crossover (1 point left)

$p_c \setminus p_m$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	72	62	48	68	132	154	181	191	196
0.2	75	54	56	45	98	127	152	172	190
0.3	71	52	63	53	38	110	149	165	196
0.4	69	68	46	55	48	82	141	167	187
0.5	64	63	56	47	60	54	99	170	184
0.6	67	60	56	51	48	54	78	128	173
0.7	61	54	53	47	41	● 34	68	77	169
0.8	66	56	56	42	45	47	60	104	144
0.9	68	54	53	53	51	40	45	137	170

7. Partially mapped crossover (2 point inside)

$p_c \setminus p_m$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	77	58	64	69	140	139	168	184	191
0.2	78	62	39	45	77	131	162	179	192
0.3	62	49	62	51	53	68	143	175	185
0.4	70	53	53	48	56	69	171	150	172
0.5	66	50	54	● 34	41	72	63	164	194
0.6	53	50	54	54	47	51	77	183	193
0.7	63	57	48	40	57	64	186	200	223
0.8	63	55	47	49	58	196	238	244	242
0.9	55	47	43	231	246	241	246	247	254

8. Partially mapped crossover (2 point outside)

$p_c \setminus p_m$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	72	55	46	53	132	145	179	194	206
0.2	63	66	44	73	66	142	167	182	192
0.3	74	58	44	49	51	132	155	179	203
0.4	52	63	57	47	72	66	146	177	181
0.5	78	67	51	● 37	40	68	102	174	187
0.6	56	54	49	49	42	62	116	150	202
0.7	62	62	58	45	42	54	97	165	219
0.8	68	58	53	55	57	43	155	196	219
0.9	67	59	59	50	52	59	204	211	224

5.2 実験 2

9. Cycle crossover

$p_c \setminus p_m$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	63	72	59	• 48	109	126	153	185	191
0.2	103	96	82	90	120	143	124	168	186
0.3	161	167	142	150	181	177	184	199	205
0.4	224	203	209	207	201	206	226	209	217
0.5	244	227	230	235	218	232	233	239	238
0.6	269	255	246	233	244	258	249	241	249
0.7	283	281	263	265	269	265	261	256	248
0.8	295	281	272	274	263	275	265	265	266
0.9	295	282	279	277	269	282	270	270	226

5.1.1 考察

実験 1 では、9 種類の交叉手法についてそれぞれの程度の交叉確率 p_c 、突然変異確率 p_m が良いかを求める実験を行なった。そして、Order crossover において 1 point right では交叉確率 p_c 30 [%]、突然変異確率 p_m 20 [%]、1 point left では交叉確率 p_c 60 [%]、突然変異確率 p_m 10 [%]、2 point inside では交叉確率 p_c 30 [%]、突然変異確率 p_m 20 [%]、2 point outside では交叉確率 p_c 50 [%]、突然変異確率 p_m 20 [%] で最良解を得ることができた。また、Partially mapped crossover において 1 point right では交叉確率 p_c 80 [%]、突然変異確率 p_m 40 [%]、1 point left では交叉確率 p_c 70 [%]、突然変異確率 p_m 60 [%]、2 point inside では交叉確率 p_c 50 [%]、突然変異確率 p_m 40 [%]、2 point outside では交叉確率 p_c 50 [%]、突然変異確率 p_m 40 [%] で最良解を得ることができた。そして、Cycle crossover では交叉確率 p_c 10 [%]、突然変異確率 p_m 40 [%] で最良解を得ることができた。

5.2 実験 2

実験 1 において各交叉手法で最良解が得られた p_c 、 p_m を用い、どの交叉手法が最も良い解を出せるかを乱数系列を変えて各 10 回の試行における最良解とその平均値を求めた。そ

5.3 実験 3

の結果を表 5.2 に示す。

表 5.2 部品数 165, 品種数 23 の問題に対する実験 2 の結果

交叉手法		p_c	p_m	$M = 300$			
				$T_{stop} = 300$		$T_{stop} = 1000$	
				Best	Average	Best	Average
OX	1 point right	0.3	0.2	45	111.5	41	64.8
	1 point left	0.6	0.1	49	116.0	34	53.9
	2 point inside	0.3	0.2	68	138.7	36	61.6
	2 point outside	0.5	0.2	60	165.1	58	114.9
PMX	1 point right	0.8	0.4	33	63.1	33	46.5
	1 point left	0.7	0.6	34	78.4	● 25	45.6
	2 point inside	0.5	0.4	34	62.1	30	● 45.2
	2 point outside	0.5	0.4	37	59.4	31	48.0
CX		0.1	0.4	48	104.2	34	56.0

5.2.1 考察

実験 2 の結果から, 3 種類の交叉手法 (OX, PMX, CX) の中で平均値が良い手法は PMX であった。その中でも, 最良解が得られた Partially mapped crossover (1 point left) が最も優れていることが分かる。

5.3 実験 3

実験 2 で得られたパラメータ (Partially mapped crossover (1 point left), $p_c = 0.7$, $p_m = 0.6$) を用いて, M , T_{stop} を様々な値に設定し, 乱数系列を変えて各 10 回試行した時の最良解と平均値および平均計算時間 (秒) を求めた。その結果を表 5.3 に示す。

5.4 実験 4

表 5.3 部品数 165, 品種数 23 の問題に対する実験 3 の結果

M	T_{stop}	Best	Average	CPUtime (second)
100	300	162	190.7	15.48
	1000	138	168.5	45.34
	10000	131	144.7	299.84
200	300	51	134.8	49.71
	1000	31	50.0	261.21
	10000	● 12	24.6	1481.13
300	300	34	78.4	126.05
	1000	25	45.6	375.80
	10000	14	● 21.4	2137.92
500	300	36	60.3	233.74
	1000	26	41.3	528.64
	10000	20	25.1	3356.25
1000	300	35	48.1	455.35
	1000	29	38.8	945.4
	10000	14	24.7	7328.27

5.3.1 考察

実験 3 から, $M = 200$, $T_{stop} = 10000$ の時に最良解を得ることができた。本手法においては, 個体数を増やすことで良い解を得られるとは限らず, 演算時間の増大をまねく。ロスタイムが小さい解を得るためには, 世代終了数 T_{stop} を大きく設定した方が良かった。

5.4 実験 4

実験 3 で得られたパラメータ (Partially mapped crossover (1 point left), $M = 200$, $T_{stop} = 10000$, $p_c = 0.7$, $p_m = 0.6$) を用いて, 様々な部品数, 品種数の問題に対して乱数系列を変えて各 10 回実験を行なった。その最良解と平均値および平均計算時間 (秒) を表 5.4 に示す。

5.5 実験 5

表 5.4 様々な部品数，品種数の問題に対する実験 4 の結果

部品数	品種数	Best	Average	CPUtime (second)
100	20	25	32.8	667.60
	40	46	66.3	971.55
200	20	410	427.8	962.37
	40	763	801.9	916.27
300	20	619	638.0	1552.84
	40	1369	1387.9	1740.88

5.4.1 考察

実験 4 では，様々な部品数，品種数の問題に対して実験を行なったが，部品数 165，品種数 23 の問題に比べ，良い結果を得ることはできなかった。このことより，部品数 165，品種数 23 の問題は様々な部品数，品種数の問題より解き易い問題であると思われる。

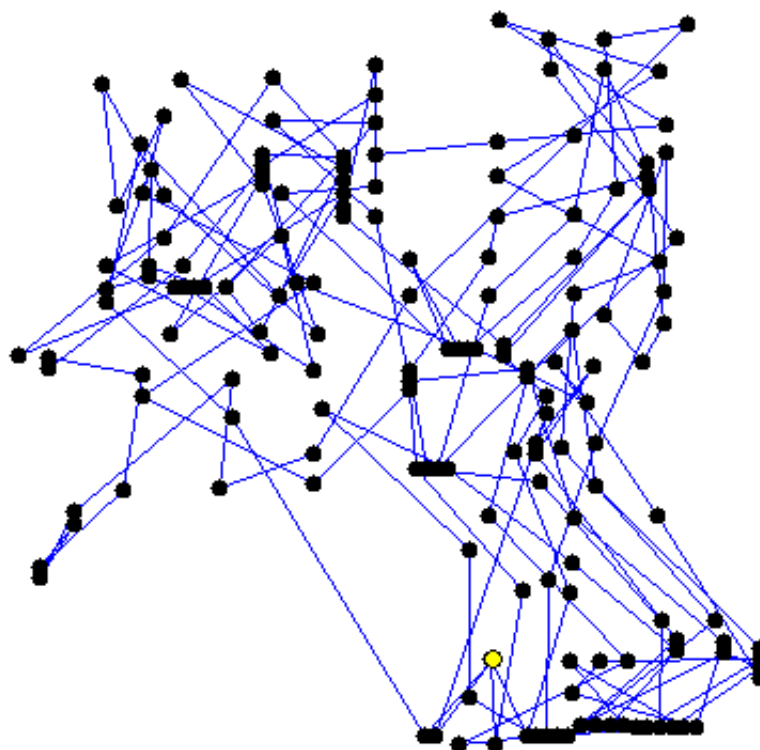
5.5 実験 5

実験 5 では，様々な部品数，品種数の問題に対して良い結果が得られなかったため，もう一度実験 3 で得られたパラメータ (Partially mapped crossover (1 point left) , $M = 200$, $p_c = 0.7$, $p_m = 0.6$) を用いて $T_{stop} = 100000$ に設定し， $losstime = 0$ の解を発見するために実験を行なった。その結果を表 5.5，得られた部品挿入順序を図 5.2 に示す。

表 5.5 部品数 165，品種数 23 の問題に対する実験 5 の結果

M	T_{stop}	Best	Average	CPUtime (hours)
200	100000	6	12.8	2.73

5.6 各世代毎の適応度（最良，最悪，平均）の変化



losstime: 6

図 5.2 部品数 165，品種数 23 の問題における *losstim 6* の部品挿入順序

5.5.1 考察

実験 5 から，パラメータ T_{stop} の値を大幅に大きな値にすることにより，*losstime 6* の値を得ることができた。しかし， T_{stop} の値を大きな値にすることで演算時間も増大した。

5.6 各世代毎の適応度（最良，最悪，平均）の変化

部品数 165，品種数 23 の問題に対して，パラメータ（Partially mapped crossover (1 point left)， $M = 200$ ， $T_{stop} = 10000$ ， $p_c = 0.7$ ， $p_m = 0.6$ ）を用いて実験を行った時の各世代毎の最良な適応度と最悪な適応度，そして世代中の平均適応度の変化を図 5.3 に示す。

5.7 まとめ

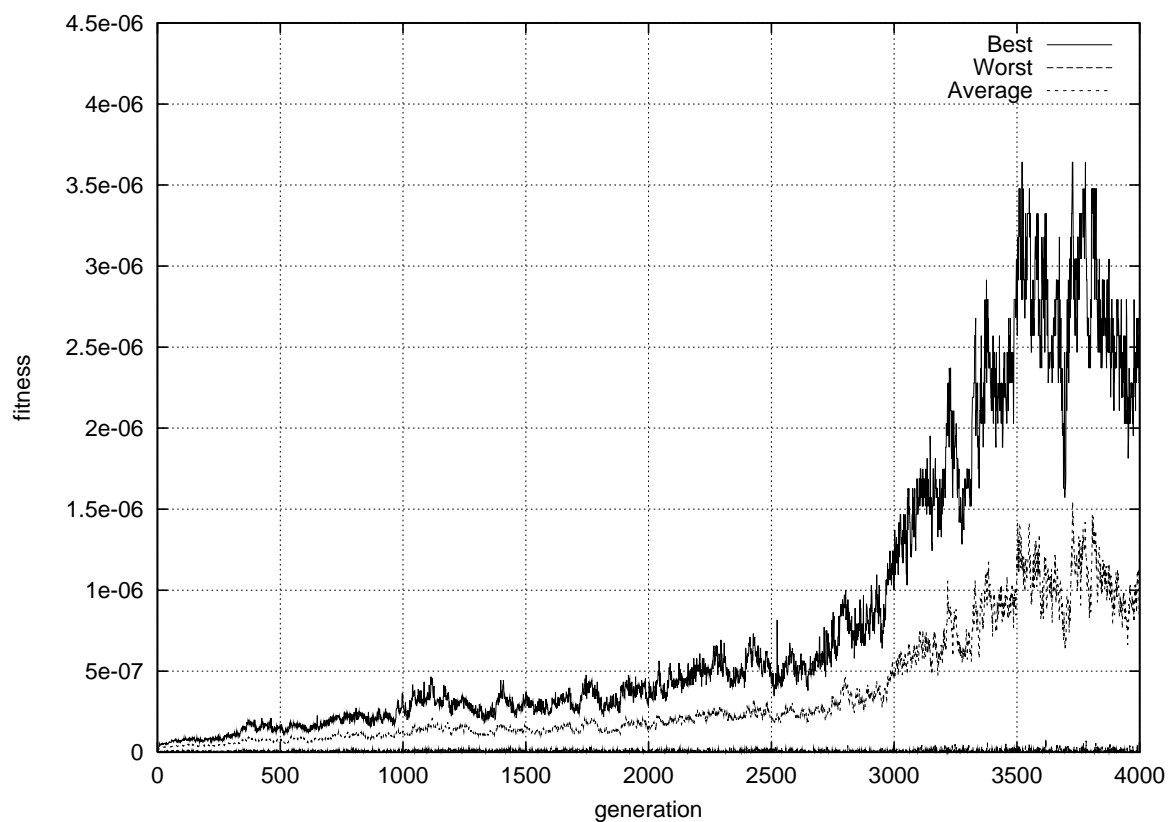


図 5.3 各世代毎の適応度（最良，最悪，平均）の変化

5.7 まとめ

最後に，いずれの実験においても *losttime* 0 の解を得ることはできなかった。このことは，本研究で提案したアルゴリズムにおいて十分改良の余地があることを示している。

次章では，以上の実験結果を踏まえた上での結論と今後の課題について述べる。

第 6 章

結論

本研究では，[1] で定義された電子部品挿入順序問題に対して，遺伝的アルゴリズムのみを用いることによりアルゴリズムの高速化を図る解法を提案した。そして計算機実験の結果，最良解として，部品数 165，品種数 23 の問題に対して *losstime* 6 の部品挿入順序およびスロット割当を発見することができた。[1] においては *losstime* 0 の部品挿入順序およびスロット割当が発見されているが，その演算時間は 36.1 [hours] 要している。本研究のアルゴリズムでは *losstime* 6 の部品挿入順序およびスロット割当を発見するのに 3.3 [hours] 要している。これは演算時間の観点から言えば，十分にアルゴリズムの高速化を図ることができたと言える。

しかし，本研究のアルゴリズムでは *losstime* 0 の解を発見することはできなかった。また，様々な部品数，品種数の問題に対しては第 5 章の表 5.4 で示したように，良い結果を得ることができなかった。したがって，今後の課題として以下の問題点を挙げる。

アルゴリズムの高速化

本研究のアルゴリズムでは第 5 章の図 5.3 に示すように，解の収束が遅いため演算時間においてタイムロスが発生している。しかし，適応度の計算方法等を改良することによりさらなる演算時間の短縮が可能になると思われる。

新たな突然変異の提案

本研究のアルゴリズムでは，最も単純な突然変異しか採用していないため，新たな突然変異を用いることで *losstime* 0 の解を発見することの手助けになると思われる。

最後に，先に述べた問題点の解決をすることで本研究でのアルゴリズムをさらに改良し，*losstime* 0 の解を発見できるアルゴリズムを開発することが今後の課題である。

謝辞

本論文は、著者が1999年7月から2001年2月までの高知工科大学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った研究の成果を記したものである。

はじめに、著者の大学生活が有意義に楽しく送ることができたことに感謝致します。

GAから本論文、プログラムなど種々の書式に至るまで、御指導・御教示を賜った坂本明雄教授、本研究の全般を通じて直接御指導・アドバイスを下さった、尊敬できるお兄さんで大学院博士後期課程院生の橋本 学氏に深く感謝致します。

また、研究室活動において、ダンディーな有賀 洋介氏、行動派の井上 祐介氏、やる気なしの折橋 祐一氏、時間ぎりぎりで燃える神谷 将司氏、最近姿を見るようになった山崎 聖太郎氏、途中で姿を見なくなった山下 由紀子さん、いつもはじけてる横谷 将樹氏、いつも心優しい久保 真理子さんに種々の面で支えになって頂いたことに感謝致します。

そして、研究室の3年生諸君の暖かい気配りに感謝すると共に、これから迫りくる研究活動を頑張って下さい。

さらに、研究活動中に、種々の相談にのってもらい、いろいろな面でアドバイスをくれた同学科の友池 貴之氏、橋本 正和氏、浜田 真世さんに感謝致します。

最後に、著者が本大学入学時から今まで過ごしやすい環境を整えて頂いた情報システム工学科の諸先生方に感謝の意を表します。

参考文献

- [1] 橋本学, “遺伝的アルゴリズムを用いた電子部品挿入順序問題の解法”, 修士論文, 1998.
- [2] 北野宏明, “遺伝的アルゴリズム”, 産業図書, 1993.
- [3] 北野宏明, “遺伝的アルゴリズム 2”, 産業図書, 1995.
- [4] 北野宏明, “遺伝的アルゴリズム 3”, 産業図書, 1997.
- [5] 伊庭斉志, “遺伝的アルゴリズムの基礎—GA の謎を解く—”, オーム社, 1994.
- [6] X. Liu ,A. Sakamoto ,and T. Shimamoto , “Genetic channel router” ,*IEICE Trans. Fundamentals* , vol.E77-A , no.3 , pp.492–501 , 1994.

付録 A

電子部品自動挿入機の仕様

電子部品自動挿入機の仕様を以下に示す。

部品自動挿入機	ラジアル機：NM-8202（松下製）
電気 CAD	ハードウェア：EWS-4800/350（NEC 製） ソフトウェア：CADVANCE-V（横河デジタルコンピュータ製）
対象電子部品	127mm ラジアルテーピング部品，縦型抵抗器，磁器コンデンサ（ $\phi 3 \sim \phi 12$ ） 電解コンデンサ（ $\phi 3 \sim \phi 10$ ），フィルムコンデンサ，ピーキングコイル トランジスタ（TO-92 タイプ，S タイプ），タンタル，ストレートリードタイプ
ヘッドの初期位置	X：16500，Y：21000*
挿入方向	X 軸，Y 軸 2 方向
部品種類	40 種類
使用可能基板寸法	MAX：508×305 MIN：150×80 mm
挿入可能基板寸法	MAX：508×295 MIN：150×70 mm
挿入サイクルタイム	約 0.6 秒 / 個
ローディング時間	約 6.0 秒 / 枚
挿入穴ピッチ	5.0±0.1 mm
挿入穴径	$\phi 1.0+0.1$ mm
挿入折角寸法	1.5±0.3 mm
挿入折角角度	約 15°
下型ヘッド	カット & クリンチ
部品貯蔵数	1,000～3,000 個 /1 パック
使用貯蔵荷姿	パック式（リール式）テーピング
使用周囲温度	20°C±10°C
機械の大きさ	巾 5,805×奥行 1,860×高さ 1,600mm（標準組合せ）
機械重量	約 2,000kg（本体重量）
使用空気圧	5kg/cm ³
空気消費量	250Nℓ/min， $\frac{1}{4}$ B

* 基板左下 5mm×5mm の基準穴を基準とした絶対座標で 1mm の 100 倍値

- 有極部品は部品の挿入方向に極性があるため，挿入角度によっては部品供給部が一部品で 2 つ必要（電解コンデンサ，集積回路，発光ダイオード，トランジスタ，ラジアル型抵抗，サイリスタなど）