

## 遺伝的アルゴリズムと最適問題

小野 俊彦 (情報工学部情報工学科)

田代 順一 (大学院工学研究科情報工学専攻)

横井 智幸 (大学院工学研究科情報工学専攻)

### Genetic Algorithms and Optimal Problem Solving

Toshihiko ONO (Department of Computer Science and Engineering,  
Faculty of Information Engineering)

Junichi TASHIRO (Communication and Computer Engineering,  
Graduate School of Engineering)

Tomoyuki YOKOI (Communication and Computer Engineering,  
Graduate School of Engineering)

Three studies on solving optimizing problems by genetic algorithms (GAs) are described. The first is to make the optimal schedule of production lines in manufacturing factories, where various kinds of products are processed through several machines. Special considerations are paid to re-scheduling after recovery from interruption caused by fault. The second is to determine the optimal layout of parts of various patterns on a sheet by GAs, to achieve the minimum amount of sheet, where the rotation of the parts is allowed. This is the extension of the previous method and realized by multiple genes construction. The last is the application of the Pareto method to GAs, which is one of the methods used for multiobjective problems. By making use of parallel calculation done by each of population, the Pareto method is successfully realized by GAs.

**Key words:** *genetic algorithms, optimal problem, scheduling, optimal cutting, pareto method*

#### 1. ま え が き

生物の遺伝機能のメカニズムをコンピュータソフトのアルゴリズムに活用して各種問題を解くことを目的にしている遺伝的アルゴリズム (以下 GA: Genetic Algorithms) は現在注目されている技術分野の一つである。モデル化が困難な問題に対して適していることから最適問題の内でもこの種の問題を対象に研究を進めてきて、その成果を報告してきた [10] [11]。本報告でそれに続く研究として以下の3テーマについての研究内容について述べる。

第1のテーマは製造工場におけるスケジューリング問題である。モデルとしては各種の工作機よりなる複数の生産ラインにて製品を製造する際に最も効率的な

工程をGAを用いて解く研究である。この種の研究はこれまでも多くの研究があるが、本研究では特に故障などにより工程に異常を生じた際に、この工程の組み替えをGAで行う点に特徴がある。故障を発生した時点での状態を記録し、これに基づき残りの工程を最適に求める方法について研究しており、その途中経過について報告する。

第2のテーマは二次元材料の最適切断へのGA適用の研究である。先に二次元材の最適切断へのGAの適用について報告した [11], [12]。この研究ではシート材上に各種自由形状の部品を、必要とするシート材を最小にする配置をGAにより求めていた。今回の研究ではこの配置の際にさらに部品の回転を許す方法を求めることにある。模様のないプラスチックシー

トなどのように方向性のないシート材から部品を切り出す際には、部品の配置に方向を考慮する必要がないので、方向を考慮する場合に比べて、さらに経済的に配置することが可能となる。この方法を実現するために各部品の遺伝子に順序のほか回転角度を表す遺伝子を追加した。部品の配置の方法は順序問題であるので、両者の遺伝子操作は異なるので、別々の遺伝子操作を行う複合遺伝子方式として実現した。シミュレーションにより性能の確認ができた。

第3のテーマは多目的最適化の一つの方法であるパレート法をGAに適用する研究である。パレート法については次章で説明するので、詳しい説明はここでは省略し、概略を述べるに留める。パレート法では一つの最適解を求めるのではなく、パレート最適解と呼ぶ複数の最適解の集合を求める。多目的最適化問題は複数の目的関数を最適にすることを求めるので、すべての目的関数を同時に最適にすることは一般にはできず、そこにトレードオフが必要となり、それを如何にするかを決めないと個々の最適解は求まらない。パレート法ではこの問題は一旦脇に置いて、パレート最適解とよぶ解の集合をまず求める。GAにおいては多数の集団により並列的に最適解を求めるので、これを利用して個体集団の各個体にパレート最適解を割り当てる方法になっている。本研究は先に提案されている方法[4]をもとに、この方法の有効性を検証することであり、シミュレーションにより確認ができた。

## 2. 多目的最適化とGA

最適化問題の多くは複数の目的関数を最適化する多目的最適化であり、GAで取り扱う最適化問題の多くもこのような多目的最適化である。多目的最適化の解法については各種の方法が研究されている[8]。その中でも一般に広く用いられているのが、すべての目的関数の加重和により一つの目的関数に変換し、単一目的関数の最適化問題として解く方法である。もちろん、各目的関数の最適化の方向が異なる場合(すなわち、あるものは最大、あるものは最小というように)にはすべての目的関数の最適化の方向が同一方向になるように変形する必要がある。また、関数の加重和を求める際は各目的関数に与える重みの大きさを全体の目的に沿うように設定する必要がある。この値は通常意志決定者が決めることになるが、これをどのように設定するかに関しての一般的な法則はなく、この決定法が

しばしば問題となる。

以上の問題を選けた最適解としてパレート最適解がある。以下では最適問題を最小化問題として述べるが、これにより一般性は失わない。まず本定義で必要となる優越関係を最初に定義して置く。

[優越関係の定義] 二つの変数  $x^1, x^2$  (いずれもベクトル) 間において

- $f(x^1) \leq f(x^2)$  のとき、 $x^1$  は  $x^2$  に優越するという。
- $f(x^1) < f(x^2)$  のとき、 $x^1$  は  $x^2$  は強い意味でに優越するという。

ここで  $\leq$  はベクトルのすべての要素間で  $<$  の関係が成り立つことを意味する。また、 $\leq$  はベクトルの各要素間に  $\leq$  の関係があるとともに、少なくとも1個の要素間には  $<$  の関係があることを意味する。すなわち

$$a \leq b \Leftrightarrow a_i \leq b_i (\forall i = 1, \dots, p)$$

$$\text{かつ } a_i < b_i (\exists i = 1, \dots, p)$$

[パレート最適解の定義]  $x^* \in X$  を最適解とするとき、つぎの解が存在する。

- $x^*$  に強い意味で優越する  $x \in X$  が存在しないとき、 $x^*$  を弱パレート最適解という。
- $x^*$  に優越する  $x \in X$  が存在しないとき、 $x^*$  を(強)パレート最適解という。
- $x^*$  が任意の  $x \in X$  に対して優越するとき、 $x^*$  を(完全)最適解という。

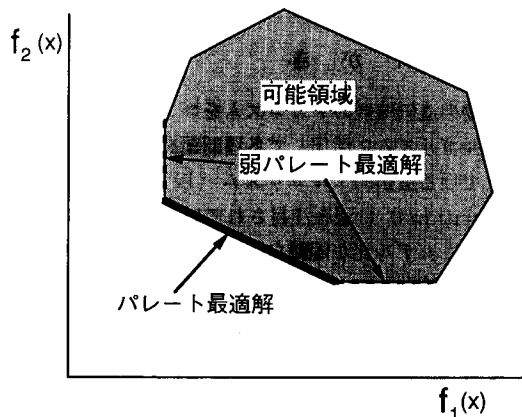


図1 パレート最適解

Fig. 1 Pareto optimal solution

これらの解の関係を目的関数が2個の場合（いずれも最小を目標）について図示したのが、図1である。図は（完全）最適解が得られない場合を示しており、多くの場合はこのようになる。このときはパレート最適解が合理的な解であることが分かる。

パレート最適解は図でも分かるように複数解からなる集合である。GAにおいては個体からなる集団の解として求めているので、それぞれの個体にパレート最適解を割り付けることができれば、比較的簡単にパレート最適解を求めることができる。この考えに基づく方法がここで採用した方法である。

### 3. スケジュールの最適化への適用

#### 3.1 問題設定

本研究の目的は複数の機械よりなる工場で複数の製品を製造する際に、すべての製品が納期に間に合うような機械の選定とスケジュールの決定をGAを用いて行うことである。各製品は完成までに数回の工程を必要とし、工程の順序および各工程で加工を行う機械と加工時間は決まっているとす。また製品を機械から機械へ運ぶ際の移動時間、加工するための治具の交換時間も考慮する。もし作業時に作業用マシンの故障や大幅なスケジュールの遅れが生じた場合は、新しいスケジュールを再構築してスケジュールの遅れを最小にする。

#### 3.2 システム構成

システム全体についてまず説明を行う。図2が示すように本システムは2つのアルゴリズムGA、スケジュー

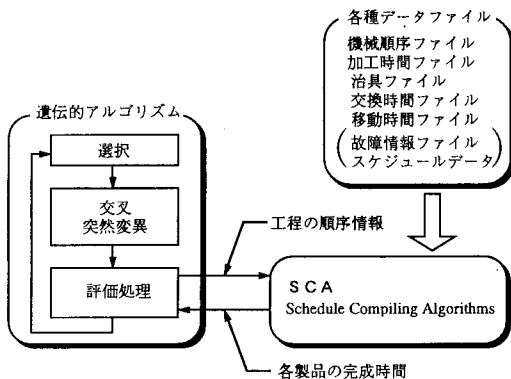


図2 システム系統図  
Fig. 2 Schematic diagram

ル作成アルゴリズム（SCA: Schedule Compiling Algorithms）と各種データファイルで構成している。本システムの主な動作は次のようになっている。

まずGAでは選択した遺伝子に交叉や突然変異の遺伝的操作を行った後に評価処理部へと進む。ここで遺伝子をSCAへ渡しスケジュールの作成を行う。SCAは制約条件を与える5種類のデータファイル（機械順序ファイル、加工時間ファイル、治具ファイル、交換時間ファイル、移動時間ファイル）を参照しながらスケジュールを作成する。その後SCAは各製品の完成時間をGAへ返し、GAが評価処理を行う。以上の操作をくり返すことで本システムは最適なスケジュールを探索する。

故障発生時のスケジュール再構築は故障発生前のスケジュールデータと故障情報ファイルの2つのデータファイルを使用して行う。

次に各種データファイルについて簡単に説明を行う。**各種データファイル**

機械順序ファイルには各工程で加工を行う機械の番号を、加工時間ファイルには加工に必要な時間を記述する。治具ファイルには各製品の各工程で使用する治具の番号を、交換時間ファイルにはある治具から別の治具への交換時間を記載する。移動時間ファイルでは各機械間の移動時間を与える。なお移動時間は製品の種類に左右されないものとする。

#### 3.3 GAの適用

GAを適用して問題を解く際は遺伝子表現が重要になってくるので、まず本研究での遺伝子表現法について述べ、続いてスケジュール作成アルゴリズム及び遺伝的操作について説明する。

##### 遺伝子表現

遺伝子は工程をガントチャートに描き出す順序を表しており、それを構成する要素は製品番号である。今回のスケジューリング問題には“各製品は第1工程

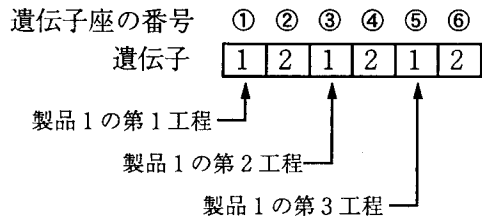


図3 遺伝子表現  
Fig. 3 Representation of genes

から順に加工しなければならない”という制約条件が存在するので、図3で示すように最初に現れた製品番号はその製品の第1工程を表し、n番目に現れた製品番号はその製品の第n工程を表すようにした。このように遺伝子表現はパス表現となり、この問題を順序問題として扱うことができる。

スケジュール作成アルゴリズム (SCA)

SCAは遺伝子が持っている工程の順序情報を参照しつつ隙間のないスケジュールを作成するアルゴリズムである。各種制約条件を考慮するためにSCAは複数のデータファイルを使用する。SCAの簡単なフローチャートを図4に示す。以下、SCAを構成する各ステップを説明する。

1. 参照する遺伝子座を最初の遺伝子座とする。
2. 遺伝子座を参照して製品番号を取り出す。また、その製品番号が現れた回数から工程番号も求める。
3. ステップ2で取り出した工程に関する情報（機械番号、加工時間、治具の交換時間、製品の移動時間）

を各種データファイルから取得する。

4. ステップ2および3で得られた製品番号、工程番号、機械番号、加工時間、交換時間、移動時間を考慮し、可能な限り最早の開始時間を求めてガントチャートに書き出す。
5. すべての遺伝子情報を読み出していれば終了。そうでなければ参照する遺伝子座を次に進めてステップ2へ戻る。

遺伝的操作

GAワークベンチにはGenitor [5]を使用した。交叉法に循環交叉 (CX: Cyclic crossover) を適用し、突然変異法にはランダムに選び出した2点の遺伝子座を交換する方法を導入した。

目的関数

目的関数値はSCAによって各製品の完成時間を求めた後に、次式によって求める。

$$Obj = \sum_i w_i x_i \tag{1}$$

$i$ は製品番号を指す。 $x_i$ は各製品の完成時間であり、 $w_i$ は各製品の優先度である。納期がせまり早い段階での完成が要求される製品には高い優先度を与えることで、スケジュールに反映させることができる。なお、目的関数値  $Obj$  が小さいほど評価は良くなる。

3.4 再構築時のスケジュール作成

スケジュールの再構築を実現するために、次の2点を考慮してスケジュールを作成する。

まず最初に考慮すべき点は「すでに実行されたスケジュールの取り扱い方法」である。前節で述べた遺伝子表現では“加工済みの工程”と“今から加工する工程”を区別できないため、再構築時に不都合が生じる。そこで本研究では、再構築後は“加工済みの工程”を取り除いた遺伝子構造とする。

つぎに考慮すべき点は、正常に加工を行っている工程の終了時間を記録し、これを工程に反映することである。これはスケジュールを再構築する際、故障後の開始時間が一定とは限らない点に起因している。

3.5 シミュレーション

本方式の特性を検証するために次のシミュレーションを行い、その結果について検討した。

シミュレーションでの制約条件

3つの機械 {A・B・C} からなる工場で3種類の製品 {X・Y・Z} を製造する必要がある場合のシミュレーションを行う。3種類の製品 {X・Y・Z} はす

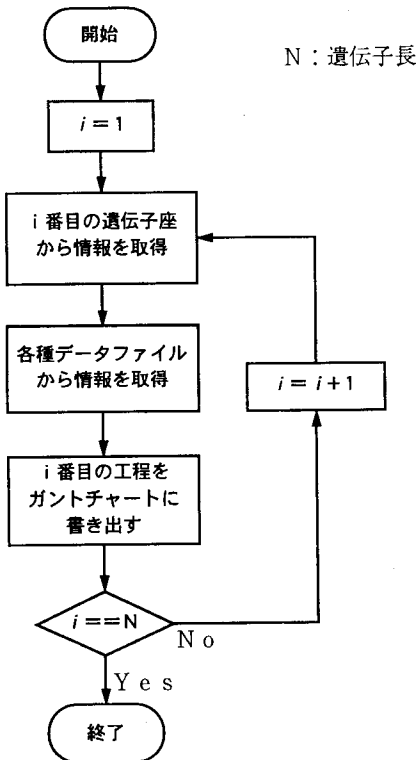


図4 SCAのフローチャート  
Fig. 4 Flowchart of SCA

べて3回の工程を経て完成する。各工程において加工を行う機械とその加工時間は表1のように決まっているものとする。

表1 シミュレーションでの制約条件

Table 1 Boundary conditions of simulation

		製品 X	製品 Y	製品 Z
第1工程	加工する機械の番号	A	B	C
	加工に必要な時間	3	2	3
第2工程	加工する機械の番号	B	C	A
	加工に必要な時間	3	1	3
第3工程	加工する機械の番号	C	A	B
	加工に必要な時間	4	3	1

時間単位 (分)

3種類の製品 { X・Y・Z } すべてにおいて、第1工程では1番の治具、第2工程では2番の治具、第3工程では3番の治具を使用する。このシミュレーションではどの治具を交換する際も、交換時間は一律に2分とする。また機械から機械への製品の移動時間は表2で与える。この表に示すように各機械間の移動時間は方向に依存しない。なお、今回はスケジュールの遅れや機械の故障は考慮しない。

表2 シミュレーションでの移動時間

Table 2 Transfer time in simulation

	機械A	機械B	機械C
機械A	—	1	2
機械B	1	—	1
機械C	2	1	—

時間単位 (分)

GAについては個体数は100個体とし、交叉確率は1.0、すなわち常に交叉を行い、突然変異確率は0.1とした。

シミュレーションの結果と検討

以上の条件でシミュレーションを行った結果、図5のようなスケジュールを得ることができた。また100世代までの最良値の変移を図6に示す。この図で示されるように本システムは早い段階で制約条件を満たす解を発見できた。移動時間や治具の交換時間を考慮し

ない場合と比べても収束状況に大きな差異は見られなかった。これは移動時間や治具の交換時間を考慮しても探索空間が増加しないためと思われる。

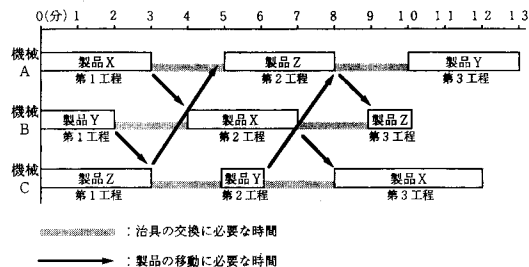


図5 シミュレーション結果

Fig. 5 Results of simulation

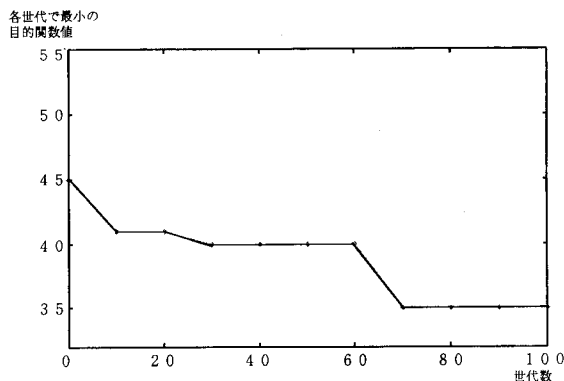


図6 最良値の推移

Fig. 6 Characteristics of convergence

なお再構築を可能とするシステムについて目下研究中であり、詳細は今後報告する予定である。

4. 二次元材の最適切断

4.1 問題設定

ここで取り扱う最適切断問題は図7に示すような一定幅 W のシート材よりグレーの部分で表す自由形状の部品を切り出す際に、次の三点を満足する部品配置を決定することである。

- 部品を重なりなく配置する。
- 配置に際しては部品の回転を許す。
- スクラップの面積を最小にする。すなわち、必要なシート長を最小にする。

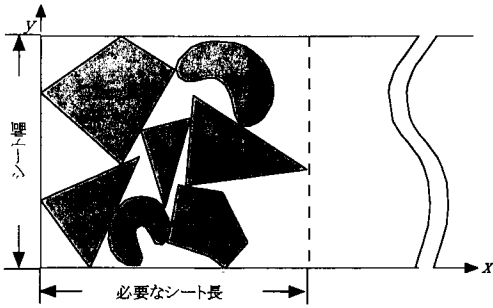


図7 シート材の切断例  
Fig. 7 Example of cutting of sheet

部品の回転を許さない方式についてはすでに報告しており [6] [11] [12], 今回新たに, 部品の回転を許すよう発展させた.

### 4.2 概念

本研究では, 遺伝子は各部品の配置を決定する順序と, その部品の回転させる角度との2種を表すようにし, 各部品を遺伝子より与えられた配置決定順序に従い, シート材上に並べる配置決定アルゴリズム (LDA: Layout Determining Algorithms) に部品の回転を追加することにより, 回転配置を実現し, これにGAを組み合わせることで目的のシステムを構築する.

### 4.3 システム構成

このシステムの系統図を図8に示す. GAにおいて, 一連の遺伝的操作を終え新たに生成した個体の遺伝の情報, すなわち, 配置決定順序と回転角度の情報をLDAに送る. LDAは, この情報に従って配置する

順番に回転処理部に部品の回転を要求し, 回転処理部から送られてきた回転済みの部品を, LDAがシート材に隙間なく, かつ, 重なりなく配置を行なう. そして, すべての部品配置が終了すると, LDAはシート材の必要長をGAに返す. GAでは, この値を目的関数値として, これを最小にするように最適化を行なっていく. GAワークベンチにはGenitorを使用している. Genitorには遺伝的操作として基本的な選択と交叉が用意されており, これに以下に述べるような独自に作成した突然変異操作と目的関数値計算部, 複数遺伝子に対応した交叉とをそれぞれ組み込みシステムを完成させた.

### 遺伝子の構成

本研究では, シート材上の各部品の配置決定順序と各部品の回転角度との2つの情報を, 2つの遺伝子で表現する方法を採用した. 遺伝子の構成例を図9に示す. 第1の配置の順序を表す遺伝子は, 各部品に整数の通し番号をつけ, その番号を順序に従って並べたものを遺伝子としている. 第2の回転角度を表す遺伝子は, 部品の番号に対応して, その部品の回転させる角度を遺伝子としている. なお, 配置決定順序を表す遺伝子表現には順序をそのまま遺伝子とするパス表現を, 回転角度を表す遺伝子表現には, 0~359までの整数の列を用いた.

遺伝子	2	3	1	4
	45	90	135	180

図9 遺伝子の構成例  
Fig. 9 Construction of genes

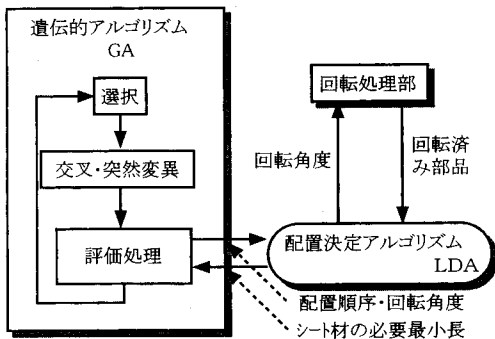


図8 システム図  
Fig. 8 Schematic diagram

### 選択操作

選択操作は前回と同じくランク方式を用いた. この方式は一般に適合値に応じて, その値の高いものから順位をつけ, 順位に比例して低減する確率により個体を選択する. 目的関数値はシート材の必要長となる. 選択確率は順位1で平均値の1.9倍(可変)に設定し, 以下順位が下がるに従い, それに比例して低減させている.

### 交叉操作

交叉操作として次のような方法を使用した. パス表現の交叉法である循環交叉(CX)を使用し, この交叉とともに対応する回転角度を表す遺伝子の要素も移動を行なう. この操作例を図10に示す.

親遺伝子1	1	2	3	4	5
	10	20	30	40	50
親遺伝子2	3	4	5	2	1
	60	70	80	90	100
	⇓ 交叉				
子遺伝子	1	4	3	2	5
	10	70	30	90	50

図10 交叉の操作例  
Fig.10 Crossover operation

突然変異操作

遺伝子が2つあるので2種類の突然変異を用意した。まず、第1の配置決定順序用遺伝子には、ランダムに2点を選びその要素を交換する位置交換突然変異を導入した。このとき、対応する2つの遺伝子の要素を交換することになる。第2の回転角度を表す遺伝子に対して、ランダムに選んだ要素を0~359の整数にランダムに変化させる突然変異を導入した。この操作例を図11に示す。

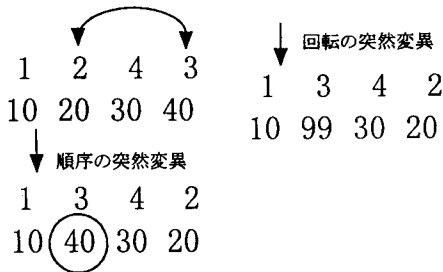


図11 突然変異の操作例  
Fig.11 Mutation operation

4.4 配置決定アルゴリズム (LDA)

LDAは各種形状の部品を遺伝子の与える配置決定順序に従い、隙間なくシート材上に配置を行なわせるアルゴリズムで、本研究では、遺伝子より回転角度情報も与えられ回転配置を行なえるようになっている。GAより渡された配置決定順序情報、回転角度情報に従い部品群より順次1つずつ部品を取り出し、図12に示すように回転を行なったのち、配置可能な位置を探索する。この動作を全部品が配置されるまで繰り返すことによって、シート材の必要長を求める。図ではNO.6部品の配置が決定し、NO.7部品の配置を決定

する状況を示す。矢印は走査方向を示す。詳しくは前報 [11] [12] を参照。

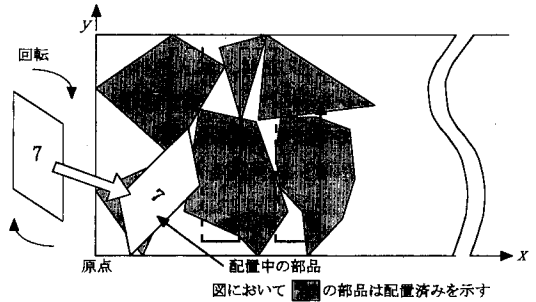


図12 LDAによる配置例  
Fig.12 Arranging parts by LAD

4.5 回転処理のアルゴリズム

点(x, y)を原点について角度θだけ時計方向に回転した後の座標(x', y')は、次の式で求められる。

$$x' = x \cdot \cos \theta - y \cdot \sin \theta$$

$$y' = x \cdot \sin \theta + y \cdot \cos \theta$$

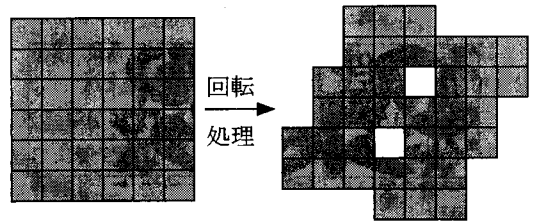


図13 回転後の隙間  
Fig.13 Gaps in patterns caused by rotation

2つの式を用いてx, yを基準にして走査を行うと、図13に示すように隙間ができてしまう。これは、座標は実数ではなく整数で、ドットという1単位の長さで表すため、斜めが階段状になり、その段がきれいに噛み合わず隙間ができるからである。そこで、逆に(x', y')に対応する変換元の位置(x, y)を求め、x', y'を基準に走査を行うというアルゴリズムを採用した。変換式は次のようになる。

$$x = x' \cdot \cos \theta + y' \cdot \sin \theta$$

$$y = -x' \cdot \sin \theta + y' \cdot \cos \theta$$

この式を用いて、回転後の矩形領域を全て走査する事により、隙間のない図形を得ることができた。

#### 4.6 シミュレーション結果と検討

本システムを検証するために自由形状部品を用いて部品数30個、個体数を100個体、最大世代を400世代としシミュレーションを行った。結果を図14に示す。また、回転配置無しでの結果を図15に示し、その2つの収束状況を図16に示す。結果図を見ると、初期世代で

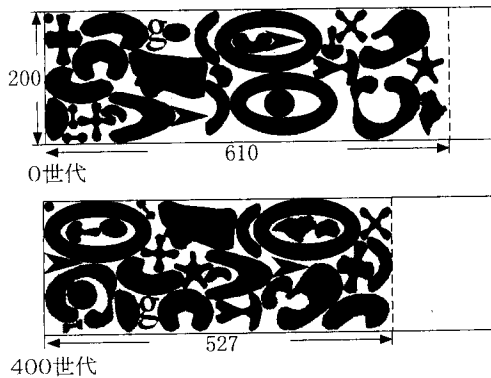


図14 回転配置有りのシミュレーション結果  
Fig.14 Results of simulation (with rotation)

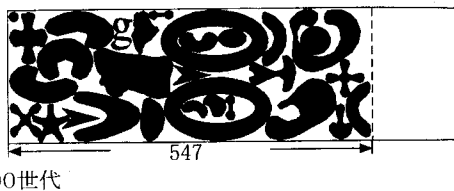


図15 回転配置無しのシミュレーション結果  
Fig.15 Results of simulation (without rotation)

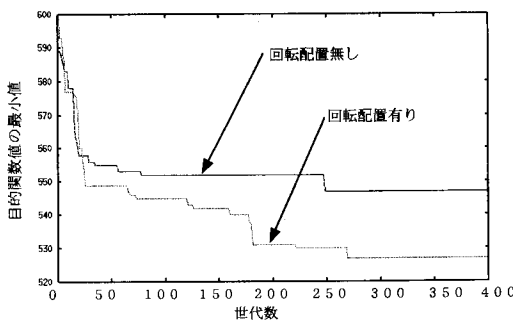


図16 収束状況  
Fig.16 Characteristics of convergence

は隙間が多いが、最終世代では高い密度で配置されている。また、回転配置無しと比較すると、回転配置有りの方がより高い密度で配置されている。収束図では、回転配置有りの方はまだシート材の必要長が最小化に向かっていることが分かる。

2つを比較すると、部品の形状によるが、回転配置によってさらに高密度で配置できることが確認でき、本方式の有効性が確認できた。

### 5. パレート GA

先に述べたように多目的最適解を求める方法のひとつとしてパレート最適解を求める方法がある。パレート最適解は唯一の解ではなく最適解の集合である。パレートGAではGAの個体集団にそれぞれのパレート最適解を分散させ、並列的に解を求めるのが、その基本の考えである。本研究ではパレートGAの性能を検証するためにJ. Horn及びN. Nafpliotisにより提案された方式[4]をベースに多少の修正を行って実施した。

#### 5.1 システム構成

本方式の全体のフローチャートを図17に示す。この方式の特徴は遺伝的操作の中で選択操作にあるので、図ではこの部分を詳細に示し、他の交叉、突然変異操作については省略している。以下、選択操作について説明する。

先ず個体集団の中から候補個体として2個体をランダムに選び出す。この2個体の優劣関係を決定するために複数個の個体からなる比較個体集合を同じくランダムに選び出す。この比較個体集合と候補2個体の優越関係を調べる。候補個体のいずれかが比較個体集合に対して優越し、他の個体が優越しない場合は、両候補個体間に優越関係があると判断し、優越した方を次世代の集団に入れる。

優越関係が定まらない場合(すなわち両候補個体がいずれも優越するか、あるいは優越しない場合)はシェアリングを行う。これについては別項で述べるが、考え方としてはパレート集団を作るためにできるだけ個体を分散させる必要があるため、すでに選び出された次世代の個体集団から離れたものを優先的に選び出すようにしている。

以上を所定の個体数に達するまで実施し、それが終わると新個体集団に対して交叉、突然変異操作を行い次世代とする。所定の世代数に達するまで、繰り返す。



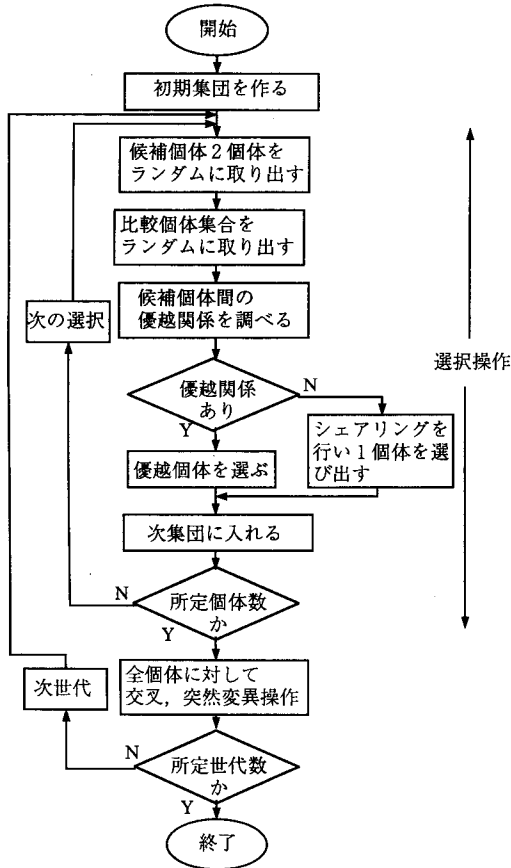


図17 パレート GA  
Fig.17 Pareto GA

優越関係の決定

候補2個体間の優越関係はつぎのようにして行う。適当数の任意の個体数からなる比較個体集合を作り、その各個体と候補個体の優越関係を比較する。ある候補個体が比較個体集合のすべての個体に対して優越しているとき、この候補個体は優越しているとする。個体間の優越関係について、当初はすべての目的関数の値で優れているとしたが、よい結果が得られなかったため、いずれかの値が優れていればよいとした。

シェアリング

これは個体集団をできるだけ分散させるために行うもので、新しく入れる個体としてできるだけ隙間を埋められるものを選ぶ操作である。2候補個体の各々に対して次式で定義する隙間数  $m_i, i = 1, 2$  を求め、その値の小さい方を選ぶ。

$$m_i = \sum_{j \in \text{popsize}} sh(d[i, j])$$

ここで  $popsize$  は現時点での新集団の個体数、 $sh(*)$  はシェアリング関数と呼ばれ、次式で定義している。

$$sh(d) = 1.0 - d/\sigma_{share}, \quad (0 < d \leq \sigma_{share})$$

$$= 0, \quad (d > \sigma_{share})$$

ここで  $d$  は2個体間の距離であり、次式で求める。

$$d(x, y) = \left( \sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}$$

ここで  $x, y$  は各個体、 $n$  は目的関数の数をそれぞれ表す。また  $p$  は距離を求める場合の次数を表し、1から  $\infty$  までの値が考えられるが、ここでは計算が容易な1に選んだ。

$\sigma_{share}$  は隙間の限界を示す数値(隙間半径)であり、あらかじめ決めて置く。式から分かるように隙間関数  $sh(*)$  はこの半径内にある個体について求め、距離が開くに従って小さくなるようにしている。従って個体の密集している場所ではこの値が大きくなるので、これが小さいことは隙間が広いことを表している。

5.2 GAプログラム

GAプログラムはGA用ワークベンチ GENESIS [5] を使用し、これに選択操作を中心に必要な改造及び追加を行って実現した。選択操作は全面的に新規に作成した。また交叉方法には2点交叉を採用しており、突然変異も行っているが、特別変わったものではないので、ここでは説明を省略する。

5.3 シミュレーション結果

いくつかの問題についてシミュレーションを行い性能の検証を行ったので、その内の二つの例について述べる。

シミュレーション-1: 2関数の最適化

ここで扱う問題は次式で表される2つの関数を最小にする例である [9]。

$$f_1(x) = \frac{1}{4} x_1^2$$

$$f_2(x) = x_1(1-x_2) + 5$$

ただし、 $1.0 \leq x_1 \leq 4.0, 1.0 \leq x_2 \leq 2.0$  とする。

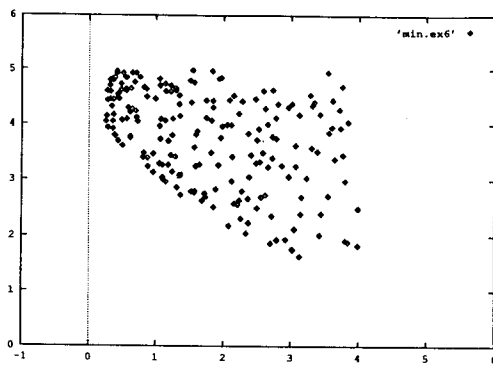
図18にシミュレーション結果を示す。a) は最初の世

代のものでランダムに設定している。b) は100世代経過して収束した状態を示している。横軸、縦軸はそれぞれ  $f_1(x)$ ,  $f_2(x)$  の値である。最初は広く分布していたものが100世代ではパレート集合が得られていることが分かる。なおシミュレーションに使用したパラメータ値は集団の個体数は200, 交叉確率は0.5, 突然変異確率は0.001, 隙間半径は0.5, 比較個体数は8にそれぞれ設定している。

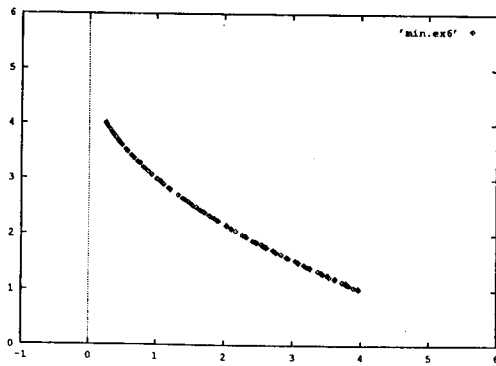
シミュレーション-2: ダブルナップザック問題

表3に示すように2種類の価値を持つ25個の物体を

適当数組み合わせで、総重量100以内で各々の価値の和が最大になる組み合わせを求める問題である。図19にシミュレーション結果を示し、a) 初期世代、b) 200世代の各状態である。横軸、縦軸はそれぞれ価値1及び価値2の合計を示す。個体数は200, 交叉確率は0.6, 突然変異確率は0.001, 隙間半径は0.3, 比較個体数は7にそれぞれ設定している。図より分かるように今回は前の例のようにきれいな線上に並んでいない。これは今回の問題が離散事象の組み合わせであるためと思われる。



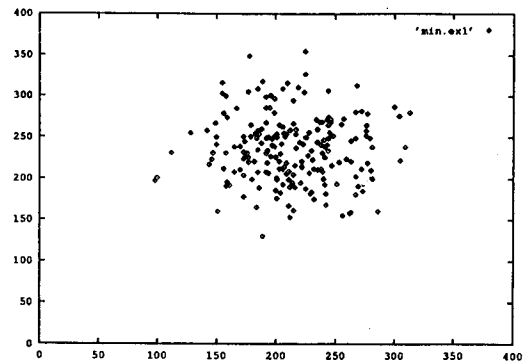
a) 初期世代



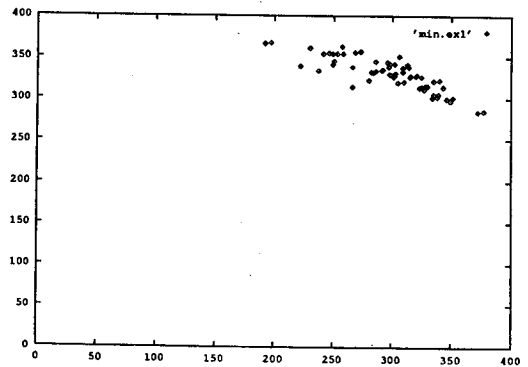
b) 100世代

図18 2関数の最適化

Fig.18 Optimization of two functions



a) 初期世代



b) 200世代

図19 ダブルナップザック問題

Fig.19 Double knapsack problem

表3 物体の価値と重量

Table 3 Values and weights of objects

物体No	重量	価値1	価値2	物体No	重量	価値1	価値2
1	10	23	46	14	7	14	41
2	17	56	19	15	13	18	39
3	15	12	37	16	17	20	10
4	11	20	10	17	19	38	20
5	11	10	20	18	12	12	12
6	8	55	33	19	19	60	40
7	10	10	30	20	3	4	9
8	12	42	61	21	16	30	35
9	18	11	52	22	9	12	35
10	9	3	40	23	3	9	10
11	20	34	12	24	11	33	22
12	15	78	32	25	17	20	40
13	9	40	3				

### 5.4 まとめ

多目的最適化問題をGAで解く際の方法としてパレート最適解を求める方法について検証を行った結果、本法が有効であることが分かった。こうして得られたパレート解から最適解を求めるには、意志決定者の志向などを考慮する必要があるため、エキスパートシステムのような対話的な方法が適しているであろう。

### 6. あとがき

本論文では遺伝的アルゴリズムの3つの分野の応用に関する研究について報告した。第1のスケジュール問題では計画スケジュールの最適化とともに一部の設備が故障などで中断した場合の組み替えスケジュールの最適化も目的としたものであり現在も引き続いて研究を進めている。第2の二次元材の最適化では部品の配置の際にその回転も許せる場合を対象にさらなる最適化を狙ったものであり、複数遺伝子構成で表現した。最後のパレートGAは多目的最適化の解法であるパレート法をGAにより実現する方法である。この研究は卒業研究生の本木潔君に行ってもらったものであり、同君の努力に対し感謝する。

以上の研究を通じて遺伝的アルゴリズムの応用に関する知見をさらに増やすことができた。

### 参考文献

[1] J. H. Holland : Adaptation in Natural and Artificial Systems, The University of Michigan Press, 211p, 1975

[2] L. Davis (Ed) : Handbook of Genetic Algorithms, Van Nostrand Reinhold, 385p, 1991

[3] D. E. Goldberg : Genetic Algorithms in Search, Optimization, and Machine Learning, AddisonWesley Pub. Co., 412p, 1989

[4] J. Horn & N. Nafpliotis : Multiobjective Optimization Using The Niche Pareto Genetic Algorithm, IlliGAL Report No.93005, p.32, University of Illinois at Urbana-Champaign, 1993

[5] Z. Michalewicz : Genetic Algorithms + Data Structures = Evolution Programs (Second. Extended Edition), Springer-Verlag, 340p, 1994

[6] T. Ono & G. Watanabe : Genetic Algorithms for Optimal Cutting, in Evolutionary Algorithms in Engineering Applications (Ed. by D. Dasgupta and Z. Michalewicz), 554p, 1997

[7] 北野編 : 遺伝的アルゴリズム, 産業図書, 328p, 1993

[8] 北野編 : 遺伝的アルゴリズム2, 産業図書, 335p, 1995

[9] 玉置ほか : 遺伝的アルゴリズムを用いたパレート最適解集合の生成法, 計測自動制御学会論文集, Vol.31, No.8, pp.1185-1192, 1995

[10] 小野, 渡辺 : 遺伝的アルゴリズムの最適問題への応用, 福岡工業大学言語情報工学研究所彙報, 第6巻, pp.89-96, 1995

[11] 小野, 渡辺, 池田, 土性 : 最適問題への遺伝的アルゴリズムの適用, 福岡工業大学言語情報工学研究所彙報, 第7巻, pp.57-70, 1996

[12] 渡辺, 小野 : 遺伝的アルゴリズムによる二次元材切断方法の決定, 電気学会論文誌 D, 117巻, 3号, pp.356-363, 1997

[13] 小野, 池田, 土性, 田代 : 遺伝的アルゴリズムによる最適問題の解法, 福岡工業大学言語情報工

学研究所彙報, 第8巻, pp.19-29, 1997

~~~~~

(平成10年5月30日 受付)