

遺伝的アルゴリズムによる最適問題の解法

小野 俊彦 (情報工学部情報工学科)
池田 武史 (大学院工学研究科情報工学専攻)
土性 雅史 (大学院工学研究科情報工学専攻)
田代 順一 (大学院工学研究科情報工学専攻)

Solving Optimal Problems by Genetic Algorithms

Toshihiko ONO (Department of Computer Science and Engineering,
Faculty of Information Engineering)
Takeshi IKEDA (Communication and Computer Engineering,
Graduate School of Engineering)
Masafumi DOSYOU (Communication and Computer Engineering,
Graduate School of Engineering)
Junichi TASHIRO (Communication and Computer Engineering,
Graduate School of Engineering)

Three examples of solving optimal problems by genetic algorithms (GAs) are explained. The first example is the automatic generation of optimal school timetable. Organizing the school timetable is tedious and time-consuming, because it requires to solve a combinatorial problem, with various restrictions to be taken into, which range from a severe one to a relative mild one. Since the GAs are suitable for a multi-objective problem with various degrees of constraint, we have studied to apply these to the university timetable problem. The second is the application of GAs to the guillotine cutting of sheet, in which a number of rectangle pieces of given sizes are cut from a single rectangle sheet by two-dimensional cutting. The object of GAs is to determine the combination of cuts to make the amount of scraps produced minimum. The last is to determine the arrangement of given number of containers of various weights on a container ship by GAS. The arrangement is made so as to make the ship stable. For this object, the location of each container is determined by GAs so that the center of gravity is set at a given point, that is, the center of the ship and both the moment of inertia around a longitudinal center line and the one around a vertical center line are set at each given large value.

Key words: *genetic algorithms, timetable, optimal cut, guillotine cut, loading container*

1. ま え が き

遺伝的アルゴリズム (以下 GA) の最適問題の解法への適用として、過去2年にわたり研究内容の報告を行ってきた。文献8)では棒材の最適切断への応用と配送センター選定問題への適用を、文献9)では二次元材の最適切断、大規模 GA の分割・統合による

高速化およびニューラルネットワーク構造決定問題への適用を、それぞれ報告してきた。本報告では上記に続く研究結果として以下の三つの分野への適用について述べる。

第1のテーマは大学における時間割作成への適用である。時間割作成は一種の組み合わせ問題であるが、多くの制約条件があり、それらを満足する時間割作成は

多大の労力と時間を必要とする作業である。制約条件には一人の教員が二つの授業を同時にできないなどの絶対的な条件のほか、絶対ではないができるだけ満足させるのが望ましい各種の要望がある。これらの各種条件を満足させる組み合わせを求めるには GA が適している。現在、基本的な研究を行い、GA による可能性を確認することができた。さらに実用規模のシステムを完成させるための研究を進めているところである。

第 2 のテーマは二次元材の切断方法の一種であるギロチン切断への GA の適用である。二次元材の最適切断では、シート材より各種多角形状さらに自由形状の部品を切り出す際に必要なシート材を最小にする部品の配置を GA により決定する研究についてはすでに報告している^{6, 9, 10)}が、ここで取り上げるギロチン切断は別の切断方式であり、従って適用する方式も異なってくる。ギロチン切断は長方形形状のシートより各種寸法の長方形形状の部品を切り出す際に使用され、端から端に直線状に切断する方法であり、いわゆるカッターを使用して切断する時の切断方法である。このようなギロチン切断の組合せで、あらかじめ決められた各種寸法の長方形の部品を切断する際に、どのような切断組み合わせを行えば最小のスクラップで切断できるかを GA を使用して求める。本報告ではその基本的な方式について述べる。

第 3 のテーマは貨物船に各種重量のコンテナなどの荷物を積載する際の積載方法を GA を用いて求める研究である。積載する際に考慮すべき事項としてここで取り上げたのは、まず重心の位置を所定位置（一般には船の中心）にすることがある。これは船の静的な安定性から必要となる。さらに船の前後方向の中心線と左右方向の中心の回りの慣性モーメントを所定の値に保つことを考えた。これは船の動的安定度に関係する。この両者の条件を満足するコンテナの配置を GA を用いて決定する。本報告では方式とシミュレーション結果について述べている。

2. 最適問題への GA の適用

GA 一般については前々報⁹⁾にて、最適問題への適用一般については前報⁹⁾にて報告したので、ここでは今回の研究に関しての特記事項について述べる。

1. 複数遺伝子構造の採用：時間割作成問題への適用においてひとつのクラスの 1 週間の時間割を 1 遺

伝子で表すとクラス、学年、学科の各数を満足するだけの数の遺伝子を必要とする。従って複数遺伝子を保有する個体を作成し、交叉、突然変異は各遺伝子別に独立に行うとともに、適合値では相互の関係を取り入れて個体として計算する必要がある。そのためにデータ構造も複雑になり、また各種遺伝操作も特別なものを必要とする。

2. 多目的最適化：時間割作成問題のように多数の制約条件があり、かつ制約条件にも各種の程度があり、絶対条件のほかある程度許容できる条件もある。これらに対応できるように制約条件をクラス分けするとともに、それらの重みをユーザが適宜変更できるようにしている。
3. 遺伝子表現：GA における遺伝子表現の重要性はすでに述べている⁹⁾。遺伝子の表現方法により問題解決が容易になる。ギロチン切断においては演算子として縦切断と横切断の 2 つを考え、被演算子である長方形部品の集合体とで二部木構造を採用し、それにより遺伝子を表現した。これにより問題が容易に扱えるようになった。このように演算子群と被演算子群よりなる遺伝子に対する遺伝子操作即ち交叉と突然変異には、演算子間、被演算子間および両者間に対する操作の必要があり、これらに対する新しい遺伝子操作方法を研究した。
4. 遺伝子の配置：Goldburg の有名な図書¹⁾に詳しく分析されているように GA の動作の基本はビルディング仮説にある。この仮説は、GA においてはビルディングブロック（定義長が短く、次数が低く、かつ適合度の高いスキーマ）が世代の進行に応じて増加してゆくということである。これはスキーマ定理の式から推定されることである。従って、GA を適用する際にはビルディングブロックを考慮して、互いに関係の深い遺伝子は、染色体内の配置としてできるだけ近くに配置するのが望ましい。このように遺伝子の配置にも配慮する必要がある。このことは貨物船へのコンテナ積載問題において触れている。

3. 時間割作成問題への応用

3.1 問題設定

大学など学校の時間割は従来手作業で作成されているが、要求される条件の複雑さ（教員数、設備など）から、この作業には多大の時間が必要となる。そのた

め時間割作成の自動化による作業の効率化への要望が高まっている。本研究では、この問題に対してGAを適用して時間割作成の自動化を図る。

研究はつぎの2段階で行う。

ステップ1：時間割自動作成に最低限必要な部分の作成とその検証

ステップ2：各種制約条件部を組み込み、入出力やインターフェースを整備して、実用システムとして運用可能なものとする。

本報告ではステップ1の研究成果について述べる。

3.2 GAの適用における問題点

GAを適用する際の問題点として遺伝子の表現法がある。この設計の仕方によってはGAの性能に大きな影響を与えるため、ここではその表現法について述べる。

時間割の最も単純な遺伝子表現法は、全授業をその順番に一つの遺伝子にて表す方法である。しかし、この方法では遺伝子内の各学年組の区別が困難であり、また交叉・突然変異を適用する際に工夫が必要になる。そこで本研究では遺伝子を各学年各組ごとに分割し、複数遺伝子として一つの個体を表現する方法を採用した。この表現法を用いることにより、交叉・突然変異を各学年各組別に適用でき、解の速い収束が期待できる。この遺伝子表現を図1に示す。

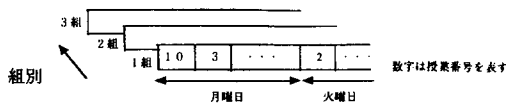


図1 遺伝子表現
Fig. 1 Composition of genes

つぎに2時間授業を配置する際の問題点について述べる。2時間授業では2時間を連続かつ適切な時間(昼休みを間に挟まない、別の日にまたがらないなど)に配置しなければならない。そこで2時間授業を一つのコマ要素として遺伝子を構成する方法が考えられるが、この方法では2時間授業があるクラスとないクラスで部分遺伝子長の違いが出て、クラスにまたがる条件(教員のダブリなど)の評価が非常に困難になる。そこで今回は2時間授業を1時間授業が二つあると定義し、評価前に遺伝子構造を変更して、2時間授業を連続かつ適切な時間に再配置する方法を採用した。

3.3 システム構成

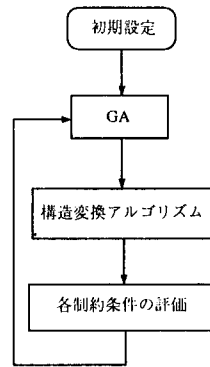


図2 システム系統図
Fig. 2 Schematic diagram for time table

本システムの構成を図2に示す。この図が示すように、本システムではGAによって生成された遺伝子をまず遺伝子構造変換アルゴリズムに渡し、2時間授業の再配置を行う。その後、各制約条件をチェックし、評価を行う。以下に遺伝子の構成、遺伝的操作、適合関数の順に説明する。

遺伝子の構成

3.2にて述べた複数遺伝子構造を使用し、各遺伝子授業番号を順序表現にて表す。順序表現にはいくつかの方法があるが、本研究で

は順序を直接表現するパス表現を用いた。遺伝子長を同じにするために、授業のない時間には空白時間要素を割り当てている。

個体の選択

選択にはランク方式を採用する。目的関数値の順にランクをつけ、それに比例した確率で遺伝子操作を行う個体を選び出す。

交叉

本研究では、順序表現用の交叉方式である循環交叉(CX法)、順序交叉(OX法)、部分写像交叉(PMX法)⁵⁾の三種類を適用し、シミュレーションにより性能の比較を行った。

突然変異

突然変異の方法には、部分遺伝子内の要素の位置を交換する位置交換突然変異(Swap法、図3)、要素の始点をずらす位置移動突然変異(Shift法、図4)、複数遺伝子内の任意の遺伝子をランダムに作り替える位置再構成突然変異(Scramble法)を採用し、それぞれの性能について比較した。

制約条件

制約条件として以下のものを当面考慮する。

1. 同一教員が同一時間に複数の授業を受け持つことがない。
2. 複数教員が同一時間に同一クラスを受け持つことがない。

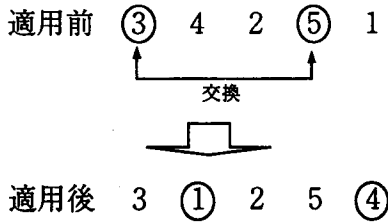


図3 位置交換突然変異
Fig. 3 Swap mutation

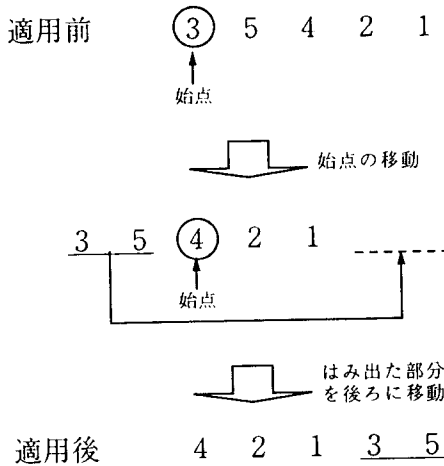


図4 位置移動突然変異
Fig. 4 Shift mutation

3. 2時間で一区切りの科目(実験など)は、連続且つ適切な時間に授業を割り当てる。
4. 教員の研修日を確保する。

制約条件にはこのほかにもあるが、ステップ1では時間割を作成するのに最低限必要な条件だけを組み込むため、これだけに絞った。その他の条件はステップ2で組み込む。

以上の制約条件をシステムに組み込むために表1のように各制約条件の評価値を設定する。これらの条件のうち、条件2は今回採用した遺伝子構造では起こり得ないので考慮する必要はない。また条件3は後述する遺伝子再配置アルゴリズムによって遺伝子を修正す

表1 制約条件の例

Table 1 Example of constraint

	条 件	評 価 値
1	同一教員が同一時間に複数の授業を持つことがない	ダブっている授業の数
2	複数教員が同一時間に同一クラスを持つことがない	
3	2時間授業の適切な時間への配置	
4	教員の研修日の確保	確保できなかった授業の数

るので、この条件もここでは考慮する必要はない。

目的関数

目的関数は以下のように設定する。

$$Obj = \sum_i w_i x_i \quad (1)$$

ここで、 w_i は各制約条件の重み(優先度)であり、制約の度合いに応じて変え、制約の度合いが大きい条件ほど大きくする。 x_i は各制約条件の評価値である。この目的関数値 Obj が小さいほど優良な個体である。

3.4 遺伝子再配置アルゴリズム

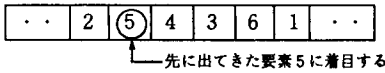
2時間授業では3.2で述べた条件を満たさなければならない。そこで本研究では、2時間授業を2つの1時間授業が集まったものと定義して、評価前に2時間授業を適切な位置に再配置するアルゴリズムを採用した。ここでは一日の授業時間を4時間とし、2時間授業が収まるべき適切な位置を1, 2時間目と3, 4時間目と仮定する。

まず、あらかじめ遺伝子内の各要素に2時間授業のフラグを用意し、これにより2時間授業を識別し、つぎの手順で2時間授業の再配置を行う。

1. 部分遺伝子内の先頭要素から2時間授業のフラグの状況をチェックする。
2. 2時間授業の要素が出てきたときに、その要素と対応する要素を残りの要素の中から探り出す。
3. 対応する要素を、先に出てきた要素の隣の要素と交換する。この時、先に出てきた要素が奇数番目(1または3時間目)か偶数番目(2または4時間目)かにより処理の仕方を変える。まず、奇数番目の場合は、その要素の後ろの要素と交換する。偶数番目の場合は、要素の前の要素と交換する。この操作によって2時間要素は1, 2時間目または3, 4時間目のいずれかに

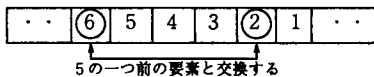
要素の5と6の二つで二時間目の授業を表すとする。

変換前：



変換後：

5が偶数番目要素の時



5が奇数番目要素の時



図5 遺伝子再配置方法

Fig. 5 Method of gene reshuffle

配置される。(図5)

4. 1~3を全ての2時間授業の位置交換が終了するまで繰り返す。

このアルゴリズムによって、2時間授業の配置に関する条件を必ず満たす遺伝子構造になる。ただし、この方式では1日の授業時間が奇数時間(1日5時間授業など)の場合は適用できない(別の日にまたがる場合がある)ので、ステップ2ではこの部分について改良を行う予定である。

表2 入力データの例

Table 2 Example of input data

教員リスト		科目リスト			
教員名	研修日	科目名	担当教員	授業時間	実施学年
吉水	月曜日	認知心理学	吉水	1時間	1年
佐村	火曜日	中国語初級	佐村	1時間	1年
田辺	水曜日	ドイツ語初級	田辺	1時間	1年
西川	木曜日	線形代数	西川	1時間	1年
柴村	金曜日	情報処理基礎	柴村	1時間	1年
中田	月曜日	基礎英語	中田	1時間	1年
		電気回路	松田	1時間	1年
		物理学	郷田	1時間	1年
		コンピュータ演習	柴村	1時間	1年
笠田	水曜日	システム工学	小田	1時間	4年
時本	木曜日	デジタル設計	笠田	1時間	4年
大野	金曜日	統計力学	時本	1時間	4年
		量子力学	郷田	1時間	4年
		通信法規	大野	1時間	4年

3.5 シミュレーション結果と検討

本システムの検証のために実施したシミュレーションとその検討結果について述べる。

シミュレーションおよび考察

このシミュレーションでは、本学情報工学科の1-4学年の1-3組の時間割作成を例に行った。ただし、ここで使用する教員名は架空のものであり、またコース分けは考慮していないなど、実際のものとは異なっている。集団の個体数は100個体とした。各学年で実施する授業とその担当教員および教員別の入力データの一部を表2に示す。

表3 シミュレーション結果

Table 3 Results of simulation

3年1組

	月曜日	火曜日	水曜日	木曜日	金曜日
1	情報工学 実験2 末野		Spolen English 初級 池野	信号処理 浜田	計算機アー キテクチャ 須田
2		ボタン情 報処理 石田		OS 柴村	倫理思想 史 安武
3	人工知能 演習 荒木	言論理論 横川	データ構 造論 寺島	哲学 塚出	論理回路 浜田
4		フランス 語初級 杉本	制御理論 相原	情報理論 前野	

教員数は40人、科目数は各学年合計で53科目である。同じ学年では各組は同じ授業を行う。

交叉法、突然変異法を変えたシミュレーションより得られたデータを比較した結果、交叉ではCX法、突然変異にSwap法を適用した場合に良い成績を得ることができた。以下この場合について述べる。まず得られた時間割の一例を表3に示す。このときの最良個体の目的関数の推移を図6に示す。図に示すように61世代で前述の制約条件を全て満たす良好な解を得ることができた。

つぎに、交叉方法別での性能の比較を図7に示す。この図は突然変異は行わず、交叉のみで100回のシミュレーションを行ったときの最良個体の目的関数値の推移を示す。この図から分かるようにCX法は突然変異なしでもかなり良好な値を得ることができた。

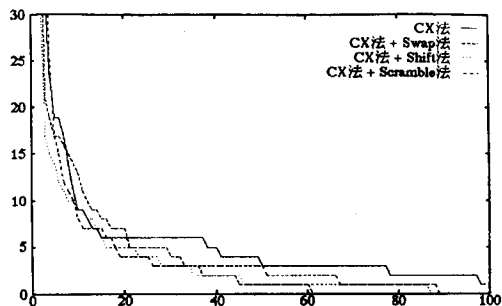


図6 最良値の推移
Fig. 6 Characteristics of convergence

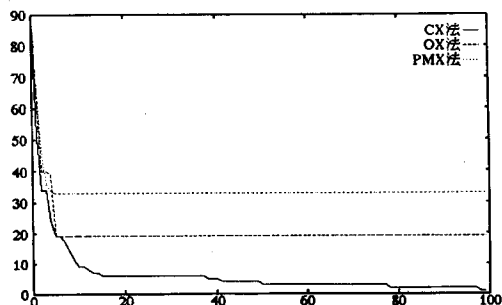


図7 交叉方法による最良値の推移の比較
Fig. 7 Comparison of convergence for various crossover methods

それに対してOX法およびPMX法は早い段階で収束してしまっているのが分かる。これは順序を保存するOX法と類似性と順序を保存するPMX法に比べ、各要素の親の位置を保存するCX法の方が多様性の維持に優れていることから生じたものと推定される。

今後の課題

本研究により時間割作成問題への遺伝的アルゴリズムの有効性を確かめることができた。また交叉法および突然変異法による性能の変化も確かめることができ、ステップ1の研究としてはまず満足のいく成果が得られた。今後はステップ2として前期および後期の時間割の同時作成、教室の割り振り、制約条件の追加・充実、対話的インターフェース部の開発などを中心にシステムの改良を行う予定である。

4. ギロチンカット方式への適用

4.1 問題設定

シート材より各種寸法の長方形部品をギロチンカット方式で切断し、そのときの必要シート長を最小とする部品の配置をGAにより求める。ギロチンカットとはシート材の端から端までを切断する切断方式であり、通常のカッターによる切断がこれに相当する。水平、垂直両方向のギロチンカットを組み合わせる必要の切断を行う。

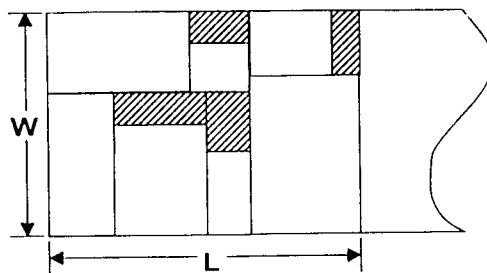


図8 ギロチンカットの例
Fig. 8 Example of guillotine cut

4.2 基本概念

本方式の目的は図8のように所定幅Wのシートにギロチンカットできるように部品を配置し端材(斜線部)の量を最小とする、すなわち必要シート長Lを最小にすることである。このためには遺伝子と配置が1対1に対応する配置表示を導入する必要がある。その方法として2項演算を用い、逆ポーランド記法(後置記法)を採用した。演算子としては水平配置(H)と垂直配置(V)の2種類を用意し、部品の一連番号と組み合わせて配置演算式を作り、これを遺伝子とする。つぎに配置記述と実際の配置の関係について簡単に説明する。

まず、水平配置の場合、記述式 "abH" によりbをaの右側に配置することを表す。a, bはそれぞれ部品をギロチンカット可能なように配置したものであり、最小は部品1個となる。同様に垂直配置の場合、"abV" はbをaの上に配置することを表す。GAにより各種演算式を作り、これを配置決定アルゴリズム(Guillotine Cut Layout Algorithm : GCLA)に送り、実際の配置を行ったのち、必要シート長を求

めてGAに返す。ギロチンカットが可能のためには、部品の配置に制約条件があるので、これを満足するよう考慮しつつ、遺伝子で与えられた配置方法に従って、全部品の配置を行う。2項演算式の採用により演算式(遺伝子)の長さは常に一定となり通常のGAを適用できる。

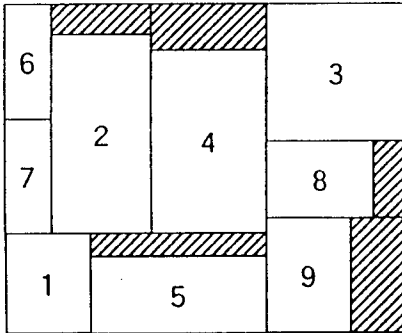


図9 部品配置図
Fig. 9 Layout of rectangular parts

4.3 配置の表現法と遺伝子表現

最初に部品の配置と遺伝子表現との関係を説明する。まず部品9個を例に取り、その配置が図9のとおりであったとする。この配置では明らかにギロチンカットが可能である。この配置を演算子H(水平配置)、V(垂直配置)を用いてバイナリー・ツリーで表現すると図10となる。Hにおいては左側の枝が左配置、右

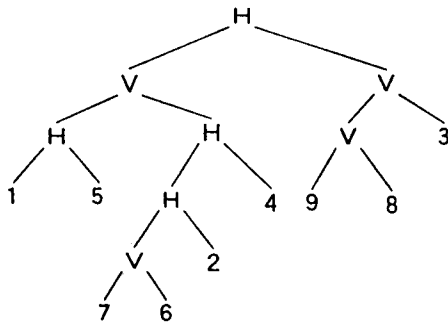


図10 配置演算ツリー図
Fig. 10 Binary tree diagram of part layout

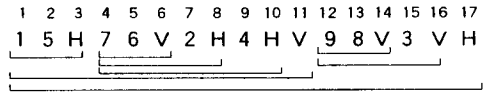


図11 逆ポーランド記法による配置の表示
Fig. 11 Representation of layout by inverse Polish notation

側の枝が右配置、Vにおいてはそれぞれ下側配置、上側配置を表す。このツリーを後置記法の演算式に変換したものが図11である。逆にこの演算式が与えられれば配置は図9のものに一意に定めることができる。

逆ポーランド記法の特徴は演算式にカッコを挿入しなくても一意に解釈できることである。また配置の如何にかかわらず演算子の数が一定であり、したがって遺伝子長も一定となる。部品数を n とすると演算子の数は $n-1$ 、遺伝子長は $2n-1$ となる。上記の例ではそれぞれ9、8、17となっている。さらに演算子の置き場所に対する制約は、各演算子より左側を見たとき、部品の数 n_p と自分自身を含む演算子の数 n_o はつぎの条件を満たす必要がある。

$$0 < n_o \leq n_p - 1 \quad (2)$$

実際のプログラムでの遺伝子の表現について各部品は整数で表現するので、H、Vも記号ではなく、1000、2000等の大きな整数で表すことにより、遺伝子を整数表現のみで扱うことができる。

4.4 システム構成

全体のフローチャートを図12に示す。以下、各機能について説明する。

1. 初期データ入力
 - (a) 部品データ(長方形)として個数および各部品の幅と長さを与える。
 - (b) シートデータとしての幅と最大シート長を与える。
2. 事前処理
3. GAを使用し、選択、交叉、突然変異などの遺伝的操作を行う。
4. 遺伝子の表す演算式を解釈し、実際の配置を定めて必要シート長を求める。
5. 次世代へのデータ蓄積
6. 初期入力で与えられた最大世代数と比較し、その世代数に達しているなら終了する。
7. 図形作成プログラムにより、実際の配置を表示する。

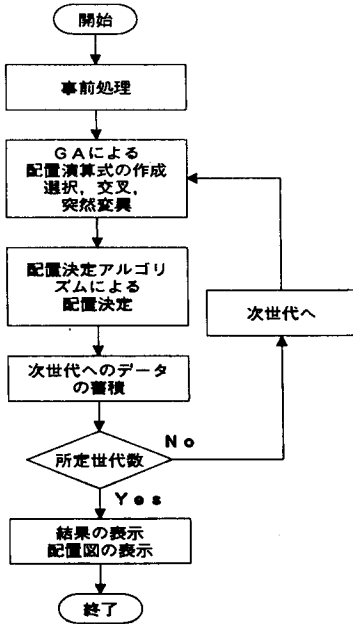


図12 システム系統図
Fig. 12 Schematic diagram for guillotine cut

4.5 GA オペレーター

初期集団の作成

初期集団は以下に示す方法を、あらかじめ定めた個体数になるまで行なって求める。

1. 部品番号を乱数を用いて並べる。
2. 演算子挿入位置を乱数により決定。但し式(2)の条件を満たすように
3. 演算子の種類を乱数により決める。

交叉方法

部品と配置演算子の両者を考慮した交叉法となり、通常の方式とは異なってくる。まず、マスクを用いて遺伝子を部品の部分と演算子の部分に分けたのち、以下の処置をおこない、再び一つの遺伝子に戻す。

1. 部品の順序のみを交叉させ、演算子はそのままとする。
2. 演算子のみを交叉させる。ただし演算子の位置は変えない。部品番号もそのままとする。
3. 両方式を組み合わせで行う。

なお、部品部分の交叉は整数表現の交叉（順序交叉、循環交叉など）を用い、演算子の部分にはバイナリ表現の交叉（Hを0、Vを1として）を用いる。図13

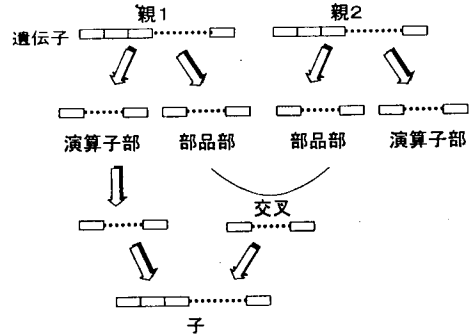


図13 交叉方法
Fig. 13 Crossover operation

に交叉の様子を示す。

突然変異

突然変異にはつぎのものを考える。

1. 部品同士の入れ替え
2. 演算子の変更 $H \rightarrow V, V \rightarrow H$
3. 演算子の位置の変更（ただし、式2を満たすこと）。

目的関数

シートの必要長を最小にするためシート長を目的関数とする。配置決定アルゴリズム（GCLA）により遺伝子に基づいて配置を定めたのち、必要なシート長を求める。

4.6 配置決定アルゴリズム（GCLA）

本アルゴリズムでは遺伝子で表された配置演算式を解釈し、実際の配置を定めたのち、シートの必要長を求める。つぎにその要点をまとめる。

1. スタックマシンの考えを導入する
2. 演算式を頭より読み出し演算子になったらスタックから部品群（以後、統合部品という、最小は部品1個）を2つ取りだし配置を決定しより大きい統合部品を作り再びスタックに戻す。図14に例を示す。統合部品を結合しつつ統合部品の幅と長さを計算する。演算子の種類によりつぎのように求める。

(a) Hの場合：統合部品の幅は両統合部品の幅の最大値より、統合部品の長さは両統合部品の長さの合計より、それぞれ求める。

(b) Vの場合：統合部品の幅は両統合部品の幅の合計より、統合部品の長さは両統合部品の長さ

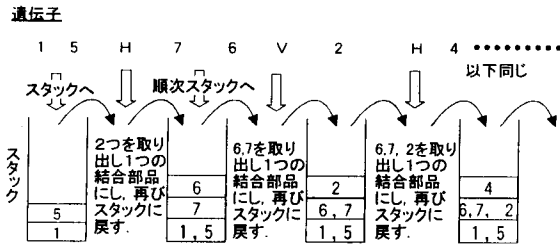


図14 配置決定方法
Fig. 14 Method of determining layout

の最大値より、それぞれ求める。

3. 統合部品の幅がシート幅 W を超えたときの処置には、つぎの2つの案が考えられる。

(a) A: その時点でその遺伝子は致死遺伝子とみなし最大必要シート長を返す。

(b) B: 遺伝子補正アルゴリズムを導入し、シート幅に収まるようにする。従ってつぎのように演算子を変更する。演算子 $V \rightarrow H$ に変更する。A 案では遺伝子を致死遺伝子として扱うので、あまり有効な方法ではない。従って現在は B 案を採用し研究を進めている。

4.7 ま と め

現在本論文に述べた方式によりプログラムを作成中であり、シミュレーション結果などは後日発表する予定である。

5. 貨物船の最適積載問題への応用

5.1 問題設定

貨物船に荷物を積み込むときは船の安定性を考えて、重心が所定の位置にあり、船の揺れが小さくなるように荷物を配置する必要がある。一般に荷物の可能な配置パターン数は、荷物の個数に対して階乗的に増加し、個数が増加するに伴い最適解を求めるのが困難になってくる。本研究ではこのような配置を GA を用いて求めることを目的とする。

本研究では貨物としてコンテナを考え、配置場所は縦方向 n 列、横方向 m 行より定まる $n \times m$ 個所のあらかじめ決められた位置から選択するものとした。また目標の重心位置は左右方向に対しては中央、前後方向に対しては自由に指定でき、目標の慣性モーメン

トの大きさも、左右と前後に分けて、波の周波数など各種の要因から決まる値を与えることができるようにした。

また、当然のことながら、1カ所には1つの荷物(コンテナ)しか置けないものとし、また荷物の個数は配置可能な位置の数以下であれば任意に設定できるようにした。

5.2 遺伝子表現

本研究ではこの問題を順序問題として取り扱い、遺伝子は整数の順序表現を用いている。また遺伝子の長さは配置個数と等しく、荷物の無い場所は質量 0 kg の荷物が用意されていると考える。

番号の付け方は図15に示すように渦巻き状の順に行った。これはビルディングブロックを考え、互いに関係の深いものを近い位置に置くことで交叉により壊れることを少なくするためである。

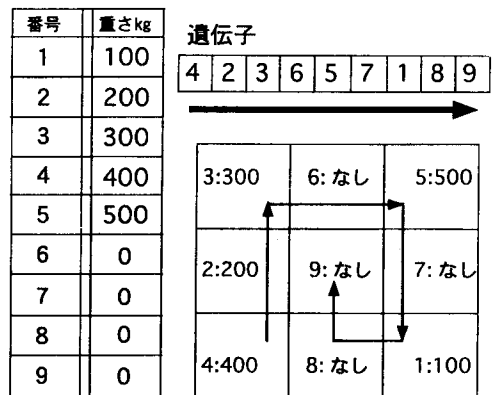


図15 遺伝子の表現方法
Fig. 15 Representation of genes

5.3 目的関数

目的関数としてはつぎの2点を考える必要がある。

1. 重心位置を目的の位置にできるだけ近づけること。
 2. 慣性モーメントの値を目的値にできるだけ近づけること。
- これらを考慮して目的関数 Obj を以下のように定めた。

$$Obj = gravity + moment \cdot coef \quad (3)$$

$$gravity = gravity_x + gravity_y \quad (4)$$

$$moment = moment_x + moment_y \quad (5)$$

ここで, $gravity_x$, $gravity_y$ は x 軸, y 軸方向における目的の重心位置と実際の重心位置の差, $moment_x$, $moment_y$ は x 軸, y 軸まわりの, 目的の慣性モーメントと実際の慣性モーメントの差, 係数

$coef$ は両条件の比重の程度を決めるための係数である。

5.4 シミュレーション結果と検討

本方式のシステムの性能を検証するため, 荷物数50個での最適積載問題を考え, GA のワークベンチとして Genitor⁹⁾ を使用してシミュレーションを行った。システム構成は図16のとおりである。シミュレーション結果はグラフィック表示される。

今回のシミュレーションでは, 交叉には辺再組み合わせ交叉 (ER)⁹⁾, 突然変異にはランダムに選んだ2つの遺伝子構成要素を交換する方法を使用した。図17に今回のシミュレーションで得られた結果のグラフィック表示を示す。ここで荷物を表す正方形の面積は荷物の質量に比例している。質量の重い荷物が外側に配置され, 貨物船が揺れにくくなっていることが分かる。また, 実際の重心位置も目的の重心位置とほぼ一致している。図18に1200世代までの目的関数の最良値の収束状況を示す。なお, 集団の個体数は50個体としている。

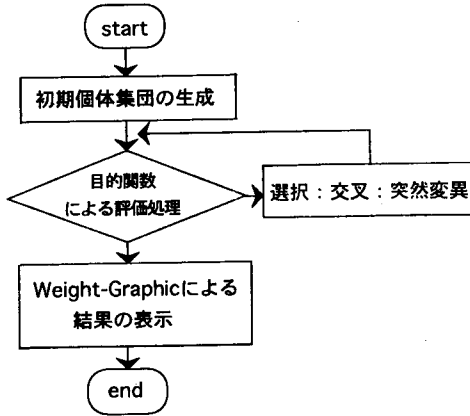
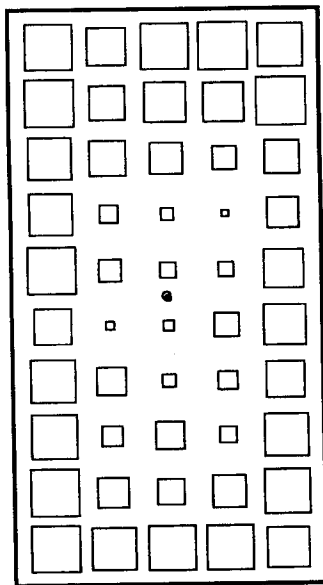


図16 システム系統図

Fig. 16 Schematic diagram for container layout



●: 理想の重心 ○: 実際の重心

図17 コンテナの配列例

Fig. 17 Example of container layout

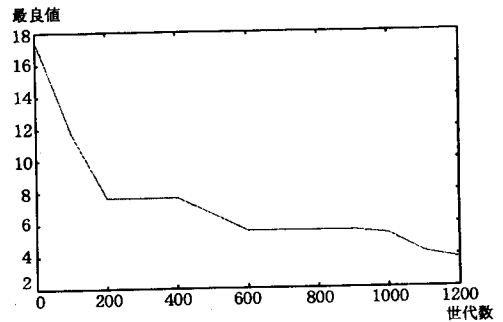


図18 最良値の推移

Fig. 18 Characteristics of convergence

6. あとがき

本論文では最適問題を遺伝的アルゴリズムを使用して解く研究として3つの分野, すなわち時間割作成, ギロチン切断, 貨物船のコンテナ積載へのGA適用について報告した。遺伝的アルゴリズムを最適問題に対して適用する場合, 問題に応じて具体的な方法を検討する必要がある。特に遺伝子の構造, 配置などの選定により解を得るまでの難易に差を生ずる。従って,

できるだけ多くの事例をとおして、それらの知識を蓄積するのが大事である。今回得られた成果は今後、類似の最適問題を解く際にも役立つものと考ええる。

参 考 文 献

- 1) J.H.Holland : Adaptation in Natural and Artificial Systems, The University of Michigan Pres, 211p (1975).
- 2) N.Christofides and C.Whitlock : An Algorithm for Two-Dimentional Cutting Problems, Operation Research, Vol.35,No, 1, pp.30-44 (1977).
- 3) L.Davis (Ed) : Handobook of Genetic Algorithms, Van Nostrand Reinhold, 385p (1991).
- 4) D.E.Goldberg : Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley Pub. Co., 412p (1989).
- 5) Z.Michalewicz : Genetic Algorithms + Data Structures = Evolution Programs (Second, Extended Edition), Springer-Verlag, 340p (1994).
- 6) T.One & G.Watanabe : Genetic Algorithms for Optimal Cutting, in Evolutionary Algorithms in Engineering Applications (Ed. by D. Dasgupta and Z. Michalewicz), May (1997).
- 7) 北野編 : 遺伝的アルゴリズム, 産業図書, 328p (1993).
- 8) 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).
- 9) 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研究所彙報, 第7巻, pp.57-70 (1996).
- 10) 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).



(平成9年5月29日 受付)