オペレーションプランの自律的獲得に関する研究*

—— GA に基づく機械学習によるアプローチ ——

川 上 敬** 嘉 数 侑 昇***

A Study on Autonomous Acquisition Systems of Operation Plans
- An Approach by GA-based Machine Learning Systems -

Takashi KAWAKAMI and Yukinori KAKAZU

This paper presents an attempt to automatically and flexibly solve the sequencing problem of machining operations that enable to produce a part under machining constraints. To realize an flexible planning mechanism, two approaches are performed. In these approaches, operation planning problems are modeled as combinatorial optimization problems. First, genetic algorithms (GAs) are applied directly. To implement the GAs, a string is encoded as an order of removing operations. Thus, a process that produces a finished part from a raw material by removing the machinable volumes is represented as a sequence of the numbers of machinable volumes. As the second approach, the classifier system (CS) is applied to this problem to obtain the further advantages. In this approach, a classifier rule corresponds to a production rule that instructs the next removal volume for a current machining state. As a result of the learning, a near optimal process plan is represented as a chain of classifiers with higher strength values. Moreover, a construction of synthesized planning systems based on GA-approach and CS-approach is also discussed. On the basis of the proposed approaches, a genetic based operation planning simulator is constructed and some numerical experiments are carried out.

Key words : operation planning problems, genetic algorithms, classifier systems, machine learning, autonomy

1. はじめに

ロボティクスやスケジューリングの分野では、様々な行動や工程を計画するプランニング問題が頻繁に出現し、その解決のために多くの AI 手法が研究されてきた¹⁾。また近年の生産技術の進歩や多品種少量生産への要求に加え、耐故障性などの観点から、さらに柔軟な自動生産システムの実現が必要とされている。そのためにも、より自律的にプランニング問題を解くための新しいプランニングスキームの実現が期待されている。なかでもプロセスプランニングは、自律化が困難な問題の1つである。これまで、プロセスプランニングを自動化するため、バリエーション方式や創成方法などの多くの試みがなされてきた²⁾が、製造の各分野に広く適用できるシステムは完成していない。この理由としては、対象物の形状と加工法の関係などが一定せず、新しい仕様の製品が出来る度にそれらの関係が大幅に変化するため、製造の各分野に広く適用できるシステムには到達していない。また、AI を利用したアプローチも提案されている³⁾が、既存のエキスパートプロセスプランニングシステムのほとんどは各部品をあらかじめ定義された特徴を用いて記述するという前提に基づき設計され、固定されたルールや手続きを用いているため与えられた問題空間への依存性という点で課題が残る。一般にプロセスプランニングタスクは幾つかのフェーズに分解される³⁾。各フェーズの持つ多様さのためにすべてのフェーズを単一のアプローチにより行うのは困難であるが、そのフェーズの幾つかは知識ベースに基づいたサブシステムによって解決されている。しかし、加工(除去)部分の順序づけ問題は、加工条件を満足する多数の順序組み合わせのなかから最適なシーケンスを選択する必要があるため、解決困難な問題である。ゆ

えに自律型プロセスプランニングシステムを構築するためにはこの順序決定フェーズの自動的な解決が重要な問題となる。CAM の分野ではこの加工順序や作業者の処理手順の決定問題はオペレーションプランニング問題と呼ばれている⁴⁾。この問題に対して分枝限定法^{5,6)}などが適用されているが、厳密解を探索する手法であるがゆえに膨大な計算量を必要とし現実的なシステムに適用するのは困難となる。またフレキシブルな生産システムを考えた場合、そこに存在する様々なプランニング・スケジューリング問題をできる限り統一的な手法で扱うことが望ましい。

そこで本論文では自律型汎用プランニングシステムの実現を目指し、オペレーションプランニング問題を自動的かつ柔軟に解決するための2つのアプローチを試みる。

まず第一に遺伝的アルゴリズム (GA)^{7~11)}を直接的に適用する。本アプローチでは GA を適用するために加工オペレーションの順序がストリングとしてコード化される。つまり原材料から加工部分の除去により完成品を生成するようなプロセスが加工部分の番号のシーケンスとして表現される。

第二のアプローチとしてクラシファイアシステム (CS)^{9~13)}を本問題に適用し、機械学習による解決を試みる。本アプローチではある加工状態において次に加工するに最も適した加工部分を示すような戦略がクラシファイアと呼ばれるルールによって表現される。CS を用いた機械学習の結果、高い強度を持つクラシファイアの列により良好なプランが示される。

そしてさらに有効な自律的プランニングシステムを構築するために先の2つのアプローチを統合する試みについても議論する。すなわち、各々のアプローチの利点を組み合わせ、欠点を補うようなシステムの構成を提案する。

最後に、提案したアプローチに基づき計算機上にプランニングシミュレータを試作し幾つかの計算機実験を行い、得られた結果により本アプローチの有効性を考察する。

* 原稿受付 平成9年9月18日

** 正 会 員 北海道女子大学短大部 (江別市文京台23)

*** 正 会 員 北海道大学工学部 (札幌市北区北13条西8丁目)

2. 問題の設定

以下では本論文で対象とするオペレーションプランニング問題のモデル化を行う。加工対象となる機械部品の表現を幾何的特徴ではなく加工部分に基づいた定義を行う。ただし幾何的制約を考慮しない訳ではなく、各加工部分は各々の幾何情報を有している。このような形状表現はオペレーションプランニング問題には適した表現方法であることが示されている²⁾。

ここで図1における部品Pのような形状を考えた場合、Pを得るためにもとの部材からQで示された部分を除去する必要がある。

Fig. 1 A part P and a decomposition of removal volume Q

このPを工作機械により加工するために、除去部Qは幾つかの加工可能単位に分割される。ここでの加工可能単位とは1つの工作機械のもとで治工具や取付具を変更することなしに同時に加工できる部分を示す。したがって、部品の形状などにより加工可能単位への分割方法が数通りも存在する場合があり、その決定によって加工コストが大きく異なる。そこでここでは以下の手順により加工可能単位への分割を行う²⁾。

- (1) 除去部Qを幾何的条件などから最小の加工単位を示す加工エレメント $v_i (i=1, \dots, n)$ に分解する。各加工エレメントは他のエレメントとの幾何的共通部分は面のみであるため式(1)、式(2)を満足する。

$$\bigcup_{i=1}^n v_i = Q \quad (1)$$

$$\bigcap_{i=1}^n v_i = \phi_v \quad (2)$$

ここで ϕ_v は体積的に空集合であることを意味し、一般的な ϕ とは区別する。

- (2) 各加工エレメントに対して、様々な加工仕様を考慮し、必要な工作機械、治工具、取付具などを選択する。
- (3) 各加工エレメント v_i を同時に加工可能な部分 $V_j (j=1, \dots, m)$ にセグメント化する。ここでは V_j を加工可能単位と呼ぶこととし、図1の場合は例えば、 $V_1=\{v_1, v_2, v_3\}$, $V_2=\{v_4\}$, $V_3=\{v_2, v_4\}$, $V_4=\{v_1\}$, $V_5=\{v_3\}$ とセグメント化されるものとする。つまり、セグメント化された各加工可能単位 V_j は加工エレメント v_i の集合の部分集合となり、各加工エレメントは最低1つの加工可能単位に含まれることになる。この加工エレメントと加工可能部分との関係は関係行列により表現される。

以上の各手順は形状処理技術や知識ベースシステムに基づいたサブシステムより解決されることが報告²⁾されている。このようにセグメント化された加工可能単位に対して、加工に関する制約条件に基づき、部品を完成させるための選択と順序付けが行われる。この選択・順序付けの結果により加工コストや治工具、取付具の交換コストなどが大きく異なるため、最適な選択・順序付けを行う必要がある。しかし、加工可能単位を選択・順序付けを行うオペレーションプランニング問題は、コストを最小にするための組合せ最適化問題であり、決定的なアルゴリズムが存在しないため解の獲得は困難である。

そこで本論文ではこのオペレーションプランニング問題を組合せ最適化問題として解くために以下の単純なモデル化を行う。すなわち、

$$v = \{v_i; i \in I\} \quad (3)$$

$$V = \{V_j; j \in J\} \quad (4)$$

$$A = \{a_{ij}; i \in I, j \in J\} \quad (5)$$

ここで、 v, V はそれぞれ加工エレメントと加工可能単位の集合で、行列 A により各々の関係が記述される。行列中の a_{ij} は v_i が V_j に含まれる場合に1、それ以外は0となる。また各加工可能単位に対して次の加工条件が定義される。

$$C = \{c_j; j \in J\} \quad (6)$$

$$T = \{t_j; j \in J\} \quad (7)$$

$$F = \{f_j; j \in J\} \quad (8)$$

ここで C, T, F はそれぞれ加工可能単位に対する加工コスト、使用工具、使用取付具の集合である。さらに加工終了までの合計コストを計算するために工具の交換コスト e_i と取付具の交換コスト e_f を定義する。

このモデルにおける目的関数は次式のような合計コスト Z で示され、この合計コストの最小値 $Z^* = \min[Z]$ を導出するような加工可能単位を選択と順序の決定が問題の解となる。

$$Z = \sum_{j \in J} c_j x_j + \sum_{i \in I} n_i e_i + \sum_{f \in F} n_f e_f \quad (9)$$

ここで、 x_j は加工可能単位 V_j が選択された場合に1、それ以外は0を示す関数である。また n_i, n_f はそれぞれ、加工順序により決定される工具と取付具の交換の有無を示す関数で、1または0の値をとる。

また各加工エレメントが必ず最低1つの選択された加工可能単位に含まれている必要があるため、制約条件として次式が与えられる。

$$\sum_{j \in J} a_{ij} x_j \geq 1, (\text{for all } i \in I) \quad (10)$$

したがって、ここで設定した問題は式(9)、式(10)で示される目的関数と制約条件を持つ集合被覆問題とみなすことができる。すなわち本アプローチでは上記問題設定に従って、与えられた初期状態(材料)から目標状態(完成部品)へ到達可能とするようなオペレーションの可能な組合せのうち、合計コスト Z を最小にするような $X = \{x_j; j \in J\}$ と加工シーケンスが解として導出される。

3. GAの適用によるアプローチ

ここで、加工オペレーションの好ましいシーケンスを自動的に導出するような自律的プランニングメカニズムを実現するために、GAを適用する。すなわち本問題に対してGAを適用することにより、良好な合計加工コストを与える加工可能単位の順列を発見する。本アプローチでは一つのストリング OP は例えば、 $OP = (2\ 4\ 1\ 3\ 5)$ のように加工可能単位 V_j の番号の順列として直接表現される。この場合は始めに V_2 を加工し、次に V_4 を加工して最後に V_5 を加工するようなオペレーションプランを意味する。したがって一般的にストリング OP は与えられた加工可能単位数 N_v を用いると次式のように書ける。

$$OP = \{\text{int}\}^{N_v} \quad (11)$$

このストリングのランダムな集まりとしてオペレーションプランの初期集団 $POP(0)$ が構成され、世代を重ねるごとに集団が進化し世代 t における集団として $POP(t)$ が作られる。

$$POP(t) = \{OP_i; i=1 \sim \text{pop_size}\} \quad (12)$$

ここで pop_size はあらかじめ決められた集団中のストリング

数である。しかしながら本手法でランダムにストリングを生成した場合、不適当なストリングが生成される可能性がある。つまり、幾何的制約条件などにより、指示された順序では加工が不可能な場合や製品が完成した後も加工オペレーションが存在する冗長なストリングである。これらの問題に対して本アプローチでは特別な付加手続きはとらずに、ストリングの先頭から順にデコードし、示された加工可能単位を除去する。そして次の条件を満たす時にデコードプロセスを終了するものとし、それまでの累積加工コストを出力する。

c1) ストリングの途中でも製品が完成した場合。

c2) 幾何的制約条件などにより、加工が不可能となった場合。

ただし、この場合は合計コストを無限大として扱う。

ここでは遺伝的オペレータとして再生、交叉、突然変異を適用し、集団を進化させる。

再生はその世代における各ストリングの実行結果に基づいて計算される再生確率値に従って行われるもので、一般にルーレット型と呼ばれる方式⁹⁾を採用する。すなわち、一つの世代 t において、各ストリングが実行され、ストリング OP_i のプランニング結果として合計加工コスト $PC_i (i=1 \sim pop_size)$ を得る。この値に対応して、ストリング i の再生確率 P_repro_i は、その世代における最悪の結果との差の2乗値から次式により求めるものとする。

$$P_repro_i = \frac{pc_i}{\sum_{i=1}^{pop_size} pc_i} \quad (13)$$

$$pc_i = (PC_i - PC^*)^2 \quad (14)$$

$$PC^* = \max[PC_i] \quad (15)$$

これによりストリング OP_i の世代 $t+1$ に対する再生数 N_repro_i は、式(16)で示される。

$$N_repro_i = P_repro_i \cdot pop_size \quad (16)$$

さらに適応したストリングを生成するために交叉を作用させることになるが、本アプローチのようにストリングが順序に基づいた形式で表現される場合には標準的な交叉は適用できない。そこでここでは順序型ユニフォーム交叉¹⁰⁾を採用する。このオペレータは、ユニフォーム交叉がストリングに対して行う操作を順序型染色体に対処できるように変更したものである。すなわち、第1の親の一部分を維持し、それ以外の部分について第2の親から情報を取り入れる。順序型ユニフォーム交叉の効果により、2つの親染色体上のノードの序列が2つの子孫中で結合される。親1と親2が与えられた時、この順序型ユニフォーム交叉は以下の手順で作用し、子孫1と子孫2を生成する。

step1) 親と同じ長さを持ち、値 "0" または "1" で構成されるストリングをランダムに生成する。

step2) step1で生成したストリング上で値が "1" である位置で、親1の値を子孫1上に複写する。

step3) 親1から子孫1に複写していない位置の値のリストを作成し、これらの値を親2において現れるのと同じ順序になるようにリストを並べ替える。

step4) 並べ直した値を順に子孫1のすきまに埋める。

step5) 子孫2を作るために、step2 から 4 のプロセスを実行する。ただし複写する値は親2の値である。

突然変異に関しても交叉と同様に一般的な突然変異を作用させることは出来ない。そこでここでは順序型突然変異オペレータとして、ストリング中のランダムに選択した2つの位置の要素を相互に交換することにより子孫を生成するものとする。

4. CS による ア プ ロ ー チ

オペレーションプランニング問題への第二のアプローチとしてCSを適用する。ここでCSを適用する目的は、(1)自律的学習メカニズムの実現、(2)既知の知識の蓄積と有効活用による学習性能の向上である。つまり本アプローチでは学習概念を包含させることにより、さらに柔軟なプランニングシステムの構築を目指す。CSにおいて、ある環境に対する行動(戦略)はクラシファイアと呼ばれるストリングルールによって表現され、このクラシファイアの有限個の集合によりシステムの挙動が制御される。また各クラシファイアには、環境への適応度に応じて増減される“強度”が割り当てられ、システムは、学習回数を重ねるごとにクラシファイア群を問題に適応するように進化させ、この進化したクラシファイア群によりシステムは解を導出する。この進化機構はクラシファイアの問題への貢献度の分配と新しいルールの生成により実現される。

一般的なCSは幾つかの構成要素から構成されており、その中心としてクラシファイアの集合が存在する。1つのクラシファイア cf_i は、条件部分 $condition_i$ と行為部分 $action_i$ から構成されるストリングルールで、条件部と行為部はそれぞれ次のような単純な記号で表現される。

$$condition_i = \{0, 1, \# \}^{lc} \quad (17)$$

$$action_i = \{0, 1 \}^{la} \quad (18)$$

ここで、 $\#$ は 0, 1 の両者とマッチするワイルドカード記号、 lc , la はそれぞれ、あらかじめ決められた、条件部と行為部のストリング長である。このような単純な表記法により、2つの利点もたらされる。すなわち、

(1) 条件部分のマッチングが非常に容易に行える。

(2) 解の探索空間を広げるために、GA を利用した新しいルール生成が可能。

これを本問題に適用する場合、 $condition_i$ は現在の加工状況にマッチングされ、 $action_i$ により次に除去する加工可能単位 V_j が指示される。この条件マッチングは、インタフェースによりデコードされた環境からの情報と各クラシファイアの条件部との照合により行われ、条件を満たしたクラシファイアが活性化される。この時、環境からの情報として各加工エレメント v_i の現在の加工状況が次のルールに従いビット列 E にコード化される。

$$E = (e_1 e_2 e_3 \dots e_{Nv}) \quad (19)$$

$$e_i = \begin{cases} 1: & \text{if } v_i \text{ is already removed} \\ 0: & \text{otherwise} \end{cases} \quad (20)$$

ここで Nv は与えられた加工エレメント数で、クラシファイアの条件部のストリング長に等しい。

CSでは、複数のクラシファイアが条件にマッチした場合、クラシファイア間の競合により1つのルールが選択される。この競合は、各クラシファイアの環境に対する貢献度に基づいて解決される。すなわち、学習の過程でよりシステムに貢献した実績を持つクラシファイアほど勝者となるチャンスを有する。ここではこの貢献度を各クラシファイアの強度 S と呼び、これに基づき計算される指値 B が最も高いクラシファイアが勝者として選択される。本アプローチでは指値は強度に指値パラメータ $\alpha (0 < \alpha < 1)$ を乗じることにより得られるものとする。

$$B(cf_i) = \alpha S(cf_i) \quad (21)$$

ここで、 $S(cf_i)$ は第 i クラシファイア cf_i の強度で、 $B(cf_i)$ はそのクラシファイアが示す指値である。

次に勝者となったクラシファイアの行為部分が次の加工可能単位の番号に翻訳され、その部分が加工される。すなわち、加

工は次の加工ルールに従って行われる。

if $a_{ij}=1$ then remove v_i (for all $i \in I$)

この時指示された部分 V_j に属する加工エレメント v_i が既に除去されていても、加工コストは加算されるものとする。

以上の動作を繰り返し、すべての加工エレメントが除去された時に動作が終了する。すなわち、終了条件は次式で表現される。

$$e_i = 1, (\text{for all } i \in I) \quad (22)$$

ここで、各クラシファイアの強度をその環境への適応度に応じて変化させるルールの強化則が必要となる。すなわち、あるルールにより採られた戦略の結果をフィードバックさせ、最終的な成功に貢献したルールにはより大きな強度を持たせ、役に立たないルールには小さな強度を与える機構が必要である。しかしある動作がルールの連鎖により表現される場合には、各クラシファイアのシステムへの貢献度の判断は困難である。そこでここでは、ルール列に対する局所的な強化則であるバケットブリゲードアルゴリズム (BBA)¹²⁾を採用する。これは各クラシファイアの強度に基づいた指値の連続的な支払いにより行われる。すなわち、時刻 $t-1$ において選択されたクラシファイア $Wcf(t-1)$ を実行することにより時刻 t の環境情報 $E(t)$ が得られる。この $E(t)$ に対してマッチしたクラシファイアが競合のために指値を差し出し、そのうちで勝者となったクラシファイア $Wcf(t)$ の指値 B が、クラシファイア $Wcf(t-1)$ に支払われる。そしてクラシファイア $Wcf(t)$ は後続のクラシファイアから支払を受け、ルール連鎖の最後のクラシファイアは環境から報酬を受取る。この環境報酬はタスクの達成の度合いに従って与えられる値で、本問題の場合は部品完成までの合計加工コストが小さいほど大きな環境報酬が与えられるように設定される。これにより連鎖中のクラシファイア $Wcf(t)$ の強度値 S は次式により変更される。

$$S(Wcf(t), t+1) = S(Wcf(t), t) - B(Wcf(t), t) + R(Wcf(t), t+1) \quad (23)$$

ここで $S(Wcf(t), t)$ は時刻 t において勝者となったクラシファイア $Wcf(t)$ の時刻 t の時点での強度を示し、 $B(Wcf(t), t)$ はその時のルール競合のために差し出した指値で、 $R(Wcf(t), t+1)$ は次の時刻 $t+1$ に後続のクラシファイアから $Wcf(t)$ に支払われる値もしくはタスク達成によって得られる環境報酬である。

BBAにより現存のルール中の有効なルールを発見できるが、クラシファイアの集合中にすべての可能なルールを登録することは不可能であるため、何らかの方法により作成された初期集団に対して学習が行われる。そこで、さらに良い解を獲得するために、GAによる新ルールの生成が可能となる。CSにおいてはルールが単純な記号により表現されるため、簡単に新しいルールが生成できる。

5. GAアプローチとCSアプローチの統合

ここではさらに強力な自律型プランニングシステムを構築するために先の2つのアプローチを統合する試みについて検討する。すなわち、各々のアプローチの利点を組み合わせ、欠点を補うようなシステムの構成を提案する。

人間が行う知的な問題解決においては、過去に経験した問題解決に基づき、新しい問題を効率的に解くことができる。このメカニズムを模倣した機械学習システムでは、類推による学習により、既に解決した問題の学習結果を新しい問題用に転送し、より高速な問題解決機構が実現できる。この機械学習システムを構成するツールとしてCSは多くの利点を持っている。つまり、

オペレーションプランニング問題への適用を考えると、GAアプローチにより探索される解は順序づけられたオペレーションのシーケンスで表現されるため、新しい問題に対する解を得るためにそのシーケンスをモディファイすることは容易ではない。それに対してCSでは知識をヒューリスティックルールの非順序集合として保存しており、そのルールが適用される順序はルール間の動的な競合により決定される。すなわち事前の知識として有効と思われるルール集合が与えられれば、その順序付けはCS内で学習できる。

またCSの特徴であるルールの競合メカニズムにより、不適当なルールが誤った類推によって与えられた場合でも、これらのルールが問題解決を失敗させることはない。つまり不適当なルールが単に他のルールと一緒に競合の場に置かれるだけである。これらのルールは競合機構の結果、問題に適応しないことが識別され、消去される。したがって、正しい類推により与えられた知識はシステムにとって強力な助けとなるが、誤った類推結果でも致命的な困難をもたらすわけではない。ただしCSでは、他の機械学習システムと同様に、すべての可能なルールをシステム中に始めから登録することが困難であるという欠点を挙げる事ができる。この問題を解決するためにCS中に新ルールの生成機構を有しているが、ここにも新ルールの生成と既存ルールの利用とのトレードオフという別の問題が存在する。すなわち、既存のルールが問題解決に有効かどうかはある程度回数問題解決を繰り返し学習することにより判定される。したがって十分に学習が進行していない時に新しいルールを生成して既存のルールと入れ替えることは大きな危険をはらんでいる。その結果として、一般のCSでは学習10回につき1つのルールを生成するようなシステムとなっているため、ルール空間が非常に大きい場合にはこの新ルール生成機構にかかる負担は問題解決に悪い影響を与える。

そこで、ここではCSアプローチにおけるクラシファイアの初期集団にGAアプローチにより獲得したプランから生成したルールを登録し、その後CSアプローチにより学習を行うような新しいシステムを考える。さらにCSにより学習した知識を類似の問題に対して活用するメカニズムを導入することにより学習性能を向上させる機構についても提案する。ここで提案するシステムの概要を図2に示す。

Fig. 2 A proposed autonomous operation planning system synthesized GA approach and CS approach

図2中のシステムの実行サイクルは次のように示される。新しい問題が与えられた時、有効に活用できる知識がない場合、GA探索器 (GA-based searcher) により解を探索し、その結果をルール生成器 (Rule generator) によりクラシファイアへコード化する。生成されたクラシファイアを含む初期ルール集団によりCS学習器 (CS-based learner) により学習が行われ、学習

結果がプランベース (Plan base) に蓄えられる。次に類似の問題が新しく与えられた場合には、プランベースから学習済み知識を抽出し、ルールモディファイア (Rule modifier) によりルールを修正してCS中に入れ込む。このシステムが同じ問題領域内で多くのタスクを学習することにより、プランベースに蓄積される知識は徐々に成長し、過去の経験を活用することによりその学習性能を向上させることが期待される。

6. 計算機実験

上記方法論により計算機上にオペレーションプランニングシミュレータを構築し幾つかの数値実験を行った。まず始めにGAアプローチの探索能力を検証するためにプランのチューニング実験を行い、次にCSアプローチによる学習実験を行った。ここで実験に用いたテストデータを表1に示し、特に図3の形状を持つ部品1に関する加工条件データを以下に示す。

$$[a_{ij}] = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (24)$$

$$C = [69.8, 18.3, 29.3, 22.1, 50.3, 16.4, 19.4, 58.1, 56.0, 2.7, 10.7, 45.8, 43.7] \quad (25)$$

$$T = [t_1, t_2, t_2, t_2, t_2, t_3, t_3, t_1, t_1, t_4, t_2, t_5, t_1] \quad (26)$$

$$F = [f_1, f_1, f_1, f_1, f_1, f_1, f_1, f_1, f_1, f_2, f_2, f_3, f_1] \quad (27)$$

$$e_i = 10.0 \quad (28)$$

$$e_f = 10.0 \quad (36)$$

加工条件は行列 a_{ij} で表され、行列中の第 i 行は分解された加工エレメント v_i を表し、第 j 列は加工可能単位 V_j を表す。 C は各加工可能単位 V_j に対する加工コスト、 T, F はそれぞれ各加工可能単位 V_j に対する使用工具、取付具である。また e_i, e_f は簡便のため固定値とした。

6.1 実験 1

上記3種の実験データに対してGAアプローチによるプランニング実験を行った。表2に本実験で使用したGAパラメータを示し、図4に3種データに対するオペレーションプランの探索結果を示す。図中、横軸はGAを適用する世代数を表し、縦軸には最良ストリングによって求められた結果、すなわちその世代における合計加工コストの最小値を示す。この結果から、世代を重ねるに従って、より高い環境適応能力を持つ個体が生まれていることが分かり、GAの優れた解探索能力が示される。またこの時の集団全体の挙動を見るために、各世代における最良値、最悪値、平均値を図5に示す。これにより集団全体が進化し、適応できない個体が淘汰されていることが分かる。

Table 1 Given experimental Parts

	The number of elementary volumes v_i	The number of segmented machinable volumes V_j
Part 1	9	13
Part 2	20	28
Part 3	29	45

Fig 3 Three-dimensional and two-dimensional views of the part 1 with decomposed elementary volumes

Table 2 Experimental GA parameters

Population size	30
Crossover rate	0.5
Mutation rate	0.01

Fig. 4 Search results of given three planning tasks by GA approach

Fig. 5 The behavior of the population while GA evolution

6.2 実験 2

次に表1の3種のデータをCSアプローチによる機械学習により解く実験を行った。ここではCSの学習性能を検証するために、全く事前の知識がない状態から学習を開始する。本実験においてクラシファイアの初期集団は乱数により作成し、重複を含まない2000本のルールで構成される。この時、各クラシファイアの持つ初期強度値はすべて同じ値を設定し、ルール競合において同強度のクラシファイアが複数活性化した場合にはランダムに勝者を選択した。またGAによる新ルールの生成は学習回数5回につき1度行われるように設定した。この時の学習結果が図6である。図中、縦軸は加工終了時のコストの合計、横軸は学習回数を示している。このグラフから分かるように、プランニングの解はかなり振動しながら学習を進め、次第に収束し、その後は何度繰り返しても、同じオペレーションプランを示す。これは学習により、高い強度を持つクラシファイアの連鎖が生成されたことを示している。

次に、環境からの報酬の値に従って学習過程がどのように変化するかを調べるために、3種類の報酬分配手法を実行した結果を図7に示す。グラフ中のタイプ2はタイプ1の2倍、タイプ3はタイプ1の4倍の報酬を環境から与えた場合の結果である。この結果では環境からの報酬が大きい方が解の収束が速いことが分かる。しかしタイプ3のように収束が速すぎるため他のタイプが到達した解を学習できない場合がある。このように学習に関するパラメータのとり方によって、学習曲線や収束速度が異なるためパラメータの最適化機構が必要となることが分かる。

6.3 実験 3

最後にGAアプローチとCSアプローチを組み合わせたプランニングシステムの実現に向けた実験を行った。ここでは部品1をGAアプローチにより探索した結果を、計算機中に試作したルール生成器によりCSが有するクラシファイアの初期集団中に入れ込み、学習を行った。その時の結果と全ルールをランダムに生成した場合の比較を図8に示す。この実験結果ではGAにより探索した結果をルールとして有している方が学習効率が良いことが分かる。この理由としては、ランダムに生成したルールの初期集団中に、問題に有効なルールの一部が存在していなか

Fig. 6 The learning curves of three planning tasks by CS approach

Fig. 7 Learning efficiencies for different learning types (the part 1)

Fig. 8 Learning curves of Part 1 using some special classifiers generated from the tuned plan by GA approach

Fig. 9 An additional planning experiment and the result; (a) two-dimensional view of the Part 4 that is similar as a part 1, (b) the improvement of learning performances through knowledge accumulation.

ったことが考えられる。

さらに学習した知識（クラシファイア群）の有効利用に関する実験を行った結果を示す。ここで与えた部品4(図9(a))はPart1と類似であるがより複雑な形状を持つ。この形状に対して全く知識のない状態から学習を行った場合と、部品1を学習済みのルールをモディファイしたCSにより学習を行った場合の収束までの学習回数を図9(b)に示している。この実験では形状の幾何的特徴によりルールのモディファイ方法を設計者が与えた。この結果から、学習した知識を蓄積するオペレーションプラン・ベースを構築することにより学習性能の向上が期待される。

7. 結 言

本論文では、プロセスプランニングにおけるオペレーションプランニング問題を対象として、問題の自律的解決を目的として以下のことを行った。

- (1) GA を直接適用するアプローチを提案し、ここでは問題向きの遺伝的オペレータを適用し進化的機構により加工可能単位の選択・順位付けを行った。
- (2) CS アプローチではある状態における次に加工するにあわしい部分をクラシファイアルールとして表現し、BBAを用いたルールの再強化則により学習を行った。
- (3) 上記両アプローチに基づき幾つかの計算機実験を行い、その解探索能力や機械学習性能を検証した。
- (4) さらに両アプローチの利点を組み合わせる試みとして、両アプローチを統合したより柔軟な自律型プランニングシステムの構造を提案し、それに向かう実験として、(a)GAにより得られた結果をCSの初期ルール集団に入れ込むことによるルール発見機構の負担の軽減、ならびに(b)学習済みクラシファイア群を類似の問題に適用することによる学習性能の向上に関する実験を行った。

しかしながらここでは、蓄えられたルール群を新しい問題に適用するためのモディファイ機構に関する一般的構築手法には言及していないため、今後の研究課題として解決しなければならない課題と考えられる。本論文で提案したアプローチの目的は、与えられた問題空間に依存しない柔軟なプランニングスキームの構築である。したがって本手法はここで対象としたオペレーションプランニング問題に特有のアプローチではなく、他の様々な工学的プランニング問題やスケジューリング問題に適用可能であると考えられる。

参 考 文 献

- 1) 例えば, A. Tate et al.: A Review of AI Planning Techniques, in *Readings in Planning*, A. Tate ed., Morgan Kaufmann, San Mateo, CA, (1990) 26.
- 2) 千田豊満: CAD/CAMシステム, 理工学社 (1990) 99.
- 3) A. Kusiak: *Intelligent Manufacturing Systems*, Prentice-Hall, New York, (1990) 142.
- 4) 沖野教朗: 自動設計の方法論, 養賢堂, (1982) 152.
- 5) 関根智明: スケジューリングの理論, 日刊工業新聞社, (1971) 75.
- 6) 佐久間敬三ほか: 機械工作法, 朝倉書店, (1984) 165.
- 7) J. H. Holland: *Adaptation in Natural and Artificial Systems*, Univ. of Michigan Press, (1975) 20.
- 8) L. Davis and M. Steenstrup: *Genetic Algorithms and Simulated Annealing: An Overview*, in *Genetic Algorithms and Simulated Annealing*, L. Davis ed., Pitman, London (1987) 74.
- 9) D. E. Goldberg: *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley, Reading, MA, (1989) 1.
- 10) L. Davis: *Handbook of Genetic Algorithms*, VNR, New York, (1991) 72.
- 11) 北野宏明: 遺伝的アルゴリズム, 産業図書, (1993) 1.
- 12) J. H. Holland et al.: *Induction*, The MIT Press, (1986) 81.
- 13) S. W. Wilson and D. E. Goldberg: A Critical Review of Classifier Systems, *Proc. 3rd Int. Conf. on Genetic Algorithms*, (1989) 244.
- 14) T. Kawakami and Y. Kakazu: A GA-based Machine Learning System for Process Planning, *Proc. IEEE Int. Workshop on Neuro-Fuzzy Control*, (1993) 311.
- 15) 川上敬, 嘉数侑昇: マルチエージェント環境における3次元箱詰め戦略の獲得 (GAによる階層型チューニング), 日本機械学会論文集, C編, 60, 577 (1994) 3219.