

タスクの位置情報に基づくスケジューリング・ルールの獲得

藤井進^{**} 諏訪晴彦^{**} 森田浩^{*}

^{*}神戸大学工学部情報知能工学科

^{**}神戸大学大学院自然科学研究科

インテリジェント・スケジューリングシステムの開発においては、効率的なルールの獲得方法の確立が重要である。著者らはこれまでに、多数のスケジュールの特性を解析することにより、適切なスケジューリング・ルールを獲得する方法について考えてきた。本稿では、ジョブ・ショップ・スケジューリング問題を対象として、帰納的学習法の1つであるC4.5学習アルゴリズムによるスケジューリング・ルールの獲得を考え、その可能性および有用性を検討する。ルール獲得のアプローチとしてタスクのスケジュール上での位置情報に着目し、2つのタスクを入れ換えることによりトレーニング・セットを生成する。また、本手法を適用することにより得られたルールの特性と有効性を検討する。

An Acquisition of Scheduling Rules Based on Information of Task Location

Susumu Fujii^{***}, Haruhiko Suwa^{**}, Hiroshi Morita^{*}

^{*}Department of Computer and Systems Engineering, Faculty of Engineering, Kobe University

^{**}Graduate School of Science and Technology, Kobe University

1, Rokkoudai-chou, Nada-ku, Kobe-shi, Hyogo, 658, Japan

In developing intelligent scheduling systems, it is important to establish an effective method for the acquisition of rules. This paper proposes a method for the acquisition of heuristic rules for job-shop scheduling from a training set by an inductive learning method and investigates its feasibility. We use C4.5 learning algorithm for the rule acquisition, which generates a decision tree inductively from a training set. The training set is proposed to be generated by interchanging two tasks in a schedule in this study. The applicability of the inductive learning and the effectiveness of the generated training set are investigated by applying the proposed method to job-shop scheduling problems.

1. はじめに

ファクトリー・オートメーションの進展、生産工程の複雑化、需要動向の変化などともない、生産の場におけるスケジューリングのコンピュータ化、自動化が大きな課題となってきた。このような状況に対する実用的手法の一つとして、生産の場での専門家の経験的知識を獲得し、知識ベース化したスケジューリング・エキスパートシステムの構築が進められている¹⁾。しかし、市場の需要動向をはじめ製品品種、設備等の社会的・生産的環境が変化した場合に、専門家が適切な判断知識を有しているとは限らず、状況に応じて新しいルールを知識ベース上に追加・更新していくことができるとは限らない。また、ルール化が困難な専門家の判断知識をいかに扱うかということも問題である。

これに代えて、スケジューリング事例を解析し、スケジューリング問題に対するルールを計算機により自動的に獲得するようなインテリジェント・スケジューリング・システムの構築が研究され始めている。このようなシステムの構築においては、効率的なルール獲得方法の確立化がなされていない。これらの研究では、予め考えられたスケジューリング・ルールの適切さをシミュレーションにより評価しておき、状況に応じた柔軟なルールの選択が可能となるという考え方がある^{2),3)}。著者らはこれらとは異なり、多数のスケジュールの特性を解析することにより、適切なスケジューリング・ルールを獲得する方法について考えてきた⁴⁾。つまり、既存のスケジューリング・ルールを予め用意するのではなく、スケジュール上の仕事の位置情報や移動操作から、スケジュールの改善に効果的なルールを獲得することを狙っている。これまでの研究では、任意に発生させた1つのスケジュールから2仕事を入れ換えることにより発生させた多数のスケジュール・パターンから、

ID3学習アルゴリズム⁵⁾による帰納的なスケジューリング・ルールを獲得する方法を提案した。この手法では、学習データの要素である事例の特徴を、入れ換え操作による評価として、スケジュールを改善するようなルールの獲得を行っている。1機械問題と2, 3機械フロー・ショップ問題を中心にこの手法の妥当性を検討した結果、最適解を与えるルールのわかっている問題では、それとほぼ等価なルールを得ることができた。

本稿では、ジョブ・ショップ問題を考え、上述のルール獲得法に基づいて拡張した学習データの表現および発生法の確立を目的とする。そこで、仕事のスケジュール上での位置情報を考慮したルールの獲得を考える。さらに、ID3学習アルゴリズムを拡張しルールの分類精度を高めたC4.5学習アルゴリズム⁶⁾を利用することによって、より信頼性の高いルールの獲得を図る。

2. スケジューリング問題におけるルール獲得手法のアプローチ

本研究で対象としている静的なスケジューリング問題は、加工決定済みの仕事の加工順序を決定する問題であり、実行可能な加工順序を全数探索することにより最適なスケジュールを得ることができる。例えば1機械問題であれば、ジョブ数を n としたとき、実行可能なスケジュールは $n!$ 個存在し、それらの中で目的関数を最適にするものが最適スケジュールとなる。

ルール獲得にあたってまず、スケジュールを全数発生させ、それらを目的関数値にしたがって並び換えたスケジュールを観察し、何らかの特性を得ることを考える。このようなスケジュールに共通する傾向を観察するため、解を全数発生できるジョブ数で考える。

例として、次のような7つの仕事を与えられている1機械最大納期遅れ最小化問題を考える。

Jobs	J_1	J_2	J_3	J_4	J_5	J_6	J_7
#1	37	8	19	15	44	36	49
#2	126	30	91	58	56	163	29

#1: 仕事の加工時間 #2: 仕事の納期

これらの仕事を納期の小さい順 ($J_7, J_2, J_5, J_4, J_3, J_1$) に並べ換えて、各々の仕事を図1のように記号化する。

$J_7 \rightarrow \dots\dots$ $J_2 \rightarrow \dots\dots\dots$
 $J_5 \rightarrow \text{————}$ $J_4 \rightarrow \text{*****}$
 $J_3 \rightarrow \text{○○○○○}$ $J_1 \rightarrow \text{◎◎◎◎}$
 $J_6 \rightarrow \text{●●●●}$

図1: 納期の記号化

このとき、実行可能解は $7! (=5040)$ 個存在し、それらを最大納期遅れの小さい順に並び換えたものは図2の左のように与える。 $7!$ 通りのスケジュールのうち、上位下位25個を取り出したとき、図2の右のようなスケジュール・パターンが得られる。

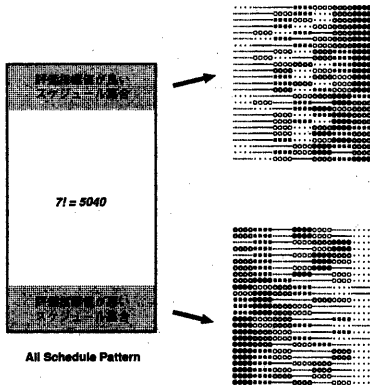


図2: 記号化したスケジュールの例

図2から、具体的な仕事の位置は特定できないが、上位に属するスケジュール群では、納期が比較的小さい仕事が先に加工されることがわかる。また、下位に属するスケジュール群では、それらが後で加工されることがわかる。最大納期遅れ最小化を評価指標とした場合、納期の早

い順に加工すること (EDDルール; Earliest Due Date)により最適解を得られることがわかっており⁷⁾、スケジュール群から得られる情報と評価基準に何らかの関係があると考えられる。すなわち、スケジュール上の仕事の位置情報を観察することにより、スケジューリングに関する何らかのルールが獲得可能であると考えられる。

著者らは上述のような観点から、多数発生させたスケジュール上の仕事属性 (加工時間、納期など) の大小関係を事例の特徴に、またスケジュールの評価をカテゴリに組み入れ、1機械問題およびフロー・ショップ問題を対象としたルール獲得を試みた。結果として、最適解を与えるルールのわかっている問題では、それと等価なルールが獲得可能であることを確認した。ここでは、ジョブ・ショップ問題を対象とし、最適解を得ることが困難な問題におけるルール獲得の可能性を考察する。

3. C4.5学習アルゴリズムの概要

本研究では、帰納的学習によるルールの自動獲得に、C4.5学習アルゴリズム (以下、C4.5と呼ぶ) を用いる。C4.5は、帰納的学習の1つであり、計算機により特定の例題から一般的ルールを推論する。帰納的学習のために用いる所与の事例集合をトレーニング・セットと呼ぶ。事例はそれを特徴づける性質の集合と、その事例が属するカテゴリで構成される。一般にカテゴリは2つ以上必要とされる。

C4.5は、ID3学習アルゴリズム (以下ID3と呼ぶ) の事例分類の再帰操作における特徴選択段階において、事例分類の精度を高めるように改良されている。ID3の欠点に、特定の事例を分類するために得られる分類木が複雑になりやすく、また特殊なルールが得られることなどがあげられる。C4.5ではこのような欠点を解消するために、生成した分類木に対して枝刈り (pruning) を行うことにより、分類木の複雑化やルールの

特殊化を軽減している。

これらの学習アルゴリズムは、事例集合であるトレーニング・セットを再帰的に分割操作し、分類木を構築する。事例の特徴は分類木のルートや中間ノードを、特徴の持つ値は枝を、カテゴリは終端ノードを構成する。ルートおよびノードの構築には、各特徴の利得情報を用いる。事例が α 個の特徴 X_i ($i=1,2,\dots,\alpha$)で構成され、トレーニング・セット中に2種類のカテゴリがそれぞれ α 、 β 含まれているとすると、特徴 X_i の持つ利得情報 $G(X_i)$ は、式(1)のように定義される。

$$G(X_i) = \{I(\alpha, \beta) - E(X_i)\} / S(X_i) \quad \dots (1)$$

$$I(\alpha, \beta) = -\frac{\alpha}{\alpha + \beta} \log_2 \frac{\alpha}{\alpha + \beta} - \frac{\beta}{\alpha + \beta} \log_2 \frac{\beta}{\alpha + \beta} \quad \dots (2)$$

$$E(X_i) = \sum_{u=1}^{v_i} \frac{\alpha'_u + \beta'_u}{\alpha + \beta} I(\alpha'_u, \beta'_u) \quad \dots (3)$$

$$S(X_i) = -\sum_{u=1}^{v_i} \frac{\alpha'_u + \beta'_u}{\alpha + \beta} \log_2 \frac{\alpha'_u + \beta'_u}{\alpha + \beta} \quad \dots (4)$$

- $I(\alpha, \beta)$: トレーニング・セットの期待情報量
- $E(X_i)$: X_i をルート(ノード)に持つ分類木に必要とされる期待情報量
- $S(X_i)$: トレーニング・セット中の v_i 個の部分集合に基づく X_i の期待情報量
- α'_u, β'_u : X_i の u 番目の枝でのカテゴリ各々の個数($u=1,2,\dots,v_i$)

C4.5では、全ての特徴をルートあるいはノードの候補とし、 $G(X_i)$ を最大化する特徴 X_i を選択し、分岐操作を行う。さらに選択された特徴を除いた事例の部分集合に対して、上述の操作を繰り返す。

生成された分類木の各々のパスは以下の形式をした1つの if - then ルールを形成する。すなわち、生成される分類木から1つのルール・

セットが得られる。

if $X_1 \wedge X_2 \wedge \dots \wedge X_n$ then *Category*

さらに、個々のルールに対して元のトレーニング・セットに依存する部分を除去するために、条件部として適用された特徴を評価し、分類上貢献しない特徴を除去することによって、ルールを簡潔なものにする。さらに、ルール・セット中の全体として必要のないルールを除去することにより、ルール・セットの洗練化を図る。

3. 対象問題の記述

3.1 ジョブ・ショップ問題

ジョブ・ショップ問題を次のように定義する。 n 個のジョブ J_j ($j=1,2,\dots,n$)を m 台の機械 M_1, M_2, \dots, M_m で加工するものとする。各ジョブ J_j は、 n_j 個のタスク T_l ($l=N_{j-1}+1, \dots, N_j$; $N_0=0; N_j=\sum_{o=1}^j n_o$)からなり、機械を通過する順序は与えられているものとする。また、各タスク T_l は所与の時間 p_l で分割なしに加工され、同一機械では2回以上加工されないものとする。ここでは、総所要時間最小化を評価基準とする。

3.2 選択グラフによる表現

ジョブ・ショップ問題は、タスクを節点とし、同一機械で加工される異なるジョブのタスク間を選択弧で、また所与のタスク間の先行関係を連結弧に各々対応させた選択グラフによる表現が可能である。これらの弧の長さとして、その始点となるタスクの加工時間を付加し、便宜上、開始節点および終了節点を付加しておく。このとき、全ての選択弧の対を解消することが、各機械上でのそれに対応するタスクの加工順序を決定することに相当する。そこで、閉路を生じさせないように選択弧を完全選択(全ての選択弧の対を解消)することによって、1つのスケジュールを得ることができ、開始節点 s から終了節点 t に至るクリティカル・パスの長さ(終了節点 t の最早開始時刻)が総所要時間 C_{\max} に対応

する。なお、クリティカル・パスは、開始節点 s から、最早開始時刻と最遅開始時刻が等しい節点をたどることにより得られる。

例えば、以下のような4仕事3機械ジョブ・シヨップ問題 (P_1) を選択グラフにより表現すると図3のようになる。

【問題 P_1 】

ジョブ : タスク (機械, 加工時間)

$J_1 : T_1(M_2, 11) T_2(M_1, 20) T_3(M_3, 19)$

$J_2 : T_4(M_3, 20) T_5(M_2, 14)$

$J_3 : T_6(M_1, 6) T_7(M_3, 10)$

$J_4 : T_8(M_1, 16) T_9(M_2, 11) T_{10}(M_3, 6)$

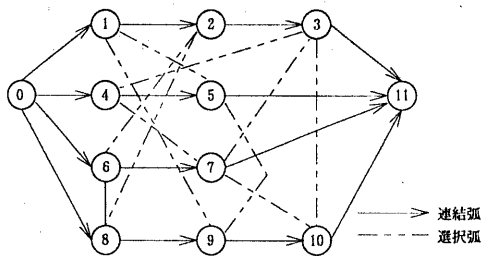


図3 : 問題 P_1 の選択グラフ

$T_6 \rightarrow T_2 \rightarrow T_8, T_1 \rightarrow T_9 \rightarrow T_5, T_4 \rightarrow T_7 \rightarrow T_{10} \rightarrow T_3$ のように解消すると、

$T_1 \rightarrow T_2 \rightarrow T_8 \rightarrow T_9 \rightarrow T_{10} \rightarrow T_3$

がクリティカル・パスとなり、総所要時間83のスケジュールを得る。

4 事例とトレーニング・セット

C4.5による帰納的学習法を本対象問題に適用する際、事例の表現法およびトレーニング・セットの生成法が重要である。以下では、事例とトレーニング・セットの作成法について述べる。

4.1 事例の作成

分類木から得られるルールは、その条件部が事例の特徴の部分集合で、結論部が事例のカテ

ゴリで表現される。したがって、事例を設定する際、ルールの条件部と結論部の表現形式を検討する必要がある。著者らはこれまでに事例表現の考え方として、仕事属性値に基づいてグループ化した仕事群のスケジュール上での位置特性による事例作成法⁸⁾や2仕事の入れ換えによる事例作成法³⁾を提案した。前者の方法では、ルールの条件部は仕事に、結論部はスケジュール上での仕事の置く位置に相当する。また、後者の方法では、条件部は入れ換える2つの仕事に、結論部は入れ換え後のスケジュールの評価に相当する。ここでは、後者の方法に基づいて、2仕事入れ換えの際の仕事の位置情報を考慮したスケジュール・パターンを事例の特徴として組み入れることを考える。

任意の実行可能スケジュール S において、同一機械で加工されるタスクのうち任意に選んだ2つのタスク T_p, T_q を入れ換えることにより得られる実行可能スケジュールを S' とする。このとき、スケジュール S を1つの事例に対応させる。ただし、スケジュール S および S' はセミアクティブ・スケジュール(機械上でのタスクの順序を保ったまま可能な限り早くタスクを加工する)に属するものとする。

2仕事入れ換え法では、事例の特徴は仕事属性値の差、カテゴリは入れ換え操作によるスケジュールの評価としていた。ここでは、選択弧対の解消を仕事を持つ位置情報と考え、各選択弧対の完全選択を事例の特徴とし、また入れ換えた2つのタスク T_p, T_q をカテゴリとする。以下では、事例特徴を $@_{pq} (\{p, q\})$; 選択弧の対 $(\{(p, q), (q, p)\})$ 、事例のカテゴリを (T_p, T_q) により表現することとする。また、事例の特徴 $@_{pq}$ は S 上で選択弧の対 $\{p, q\}$ が弧 (p, q) で解消されている場合、 $@_{pq} = 1$ 、弧 (q, p) では、 $@_{pq} = 0$ とする。

問題 P_1 において、 $T_6 \rightarrow T_8 \rightarrow T_2, T_1 \rightarrow T_9 \rightarrow T_5, T_7 \rightarrow T_4 \rightarrow T_3 \rightarrow T_{10}$ で解消されるスケジュールをサンプル・スケジュールとすると、

事例の特徴は、

(@_{6,2}, @_{6,8}, @_{2,8}, @_{1,9}, @_{1,5}, @_{9,5},
@_{7,4}, @_{7,10}, @_{7,3}, @_{4,10}, @_{4,3}, @_{10,3})

となる。このとき、3.2節の例において、2
タスク T_2, T_8 を入れ換えるとすると、

特徴： (0,0,1,0,0,0,1,0,0,0,0,0)

カテゴリ： (T_2, T_8)

のような事例が得られる。

4.2 トレーニング・セットの作成

ある与えられた n 仕事 m 機械に対するサンプル・スケジュール S_0 から任意の機械上の2つのタスクをランダムに選んで入れ換える。この入れ換え操作を s 回繰り返すことにより、 s 個のスケジュール $S_k (k=1, \dots, s)$ を発生させる。このとき、 S_0, S_1, \dots, S_s の評価値を I_0, I_1, \dots, I_s とし、 S_0, S_1, \dots, S_{s-1} を図4のように $D_{k-1} = I_k - I_{k-1}$ の昇順に従って並べる。

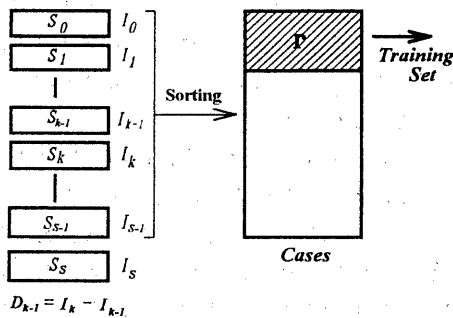


図4：トレーニング・セットの生成

さらに、これら全数のうち、 $D_k < 0$ すなわち、評価値が改善されているスケジュールのうち上位 r 個の事例をトレーニングセットとする。これによって、トレーニング・セットの大きさを限定するとともに、明確な特徴を有する事例を得ることができる。と期待できる。

5 学習実験

5.1 ルールの獲得

ジョブ数3~7、機械台数3~5、タスク数2~5とし、ランダムに20個の問題を作成する。各問題に対して、任意に発生させたサンプル・スケジュール上のタスク集合から任意の2つのタスクを逐次的に入れ換え、10000個の事例を作成する。これらのうち、 D_k の最小順に上位1000個取り出しトレーニング・セットとする。

ところで、トレーニング・セット中の事例は、そのカテゴリが入れ換えるタスクを意味するだけであって、いわゆる正・負の関係を有しているわけではない。さらに、特徴値は同一であるがカテゴリが異なる事例（事例集合）対が存在し、分類操作の際、一方の事例（事例集合）に全て吸収される、すなわちカテゴリが置き換えられてしまう場合がある。そこで、あるトレーニング・セットに3種類以上のカテゴリが存在した場合、そのうち1つのカテゴリを選び、それを有する事例を正の事例、それ以外の事例を負の事例（カテゴリ名は、"otherwise"）とし、ルール学習を行い、この操作を他のカテゴリについても同様に行う。さらに、各々の学習で得られたルール・セットを統合し、結論部が"otherwise"のルールを除去することとする。

発生させた20個の問題のひとつである、問題 P_1 において獲得されたルール・セットを示す。なお、ルールの形式は、

条件部：

(@_{6,2}, @_{6,8}, @_{2,8}, @_{1,9}, @_{1,5}, @_{9,5}, @_{7,4}, @_{7,10},
@_{7,3}, @_{4,10}, @_{4,3}, @_{10,3})

結論部：入れ換える2つのタスク(T_p, T_q)

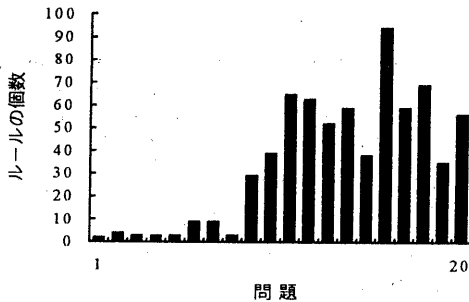
であり、(条件部) → (結論部)と表わされる。また、条件部となる個々の特徴 $@_{p,q}$ は*の場合、どちらの向きに解消されてもよいことを表す。

<問題P₁において獲得したルール>

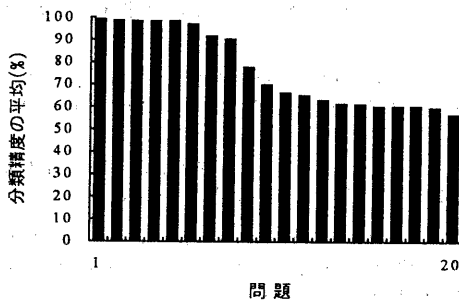
- [1] (*,1,*,*,*,*,*,*,0,*,*) → (T₆,T₂)
- [2] (*,0,*,*,*,*,*,1,*,*,*,*) → (T₂,T₈)
- [3] (*,1,*,*,*,*,*,*,1,*,*,*) → (T₂,T₈)
- [4] (*,0,*,*,*,*,*,1,*,*,*,*) → (T₇,T₁₀)
- [5] (*,*,*,*,*,*,1,*,*,*,*,*) → (T₁₀,T₃)

例えばルール[1]では、T₈→T₆、T₇→T₃、T₄→T₁₀のように解消されている場合、タスクT₆とT₂を入れ換える、という意味を持つ。

図5に、問題P₁~P₂₀で得られたルール・セットの(a)サイズと(b)分類精度の平均を示す。また、図6は各問題の総タスク数を表す。これらのグラフは、問題P₁~P₂₀を分類精度の平均の高い順に並べ換えている。分類精度は、得られたルール・セットとトレーニング・セット中の全ての事例とのマッチングに成功した割合を表す。



(a) ルール・セットのサイズ



(b) ルール・セットの分類精度の平均

図5：ルール獲得の結果

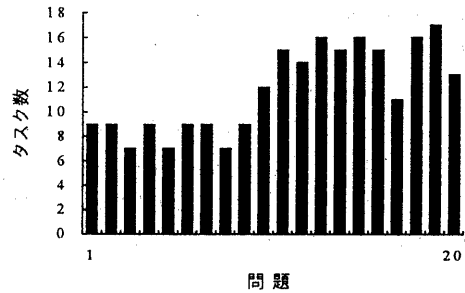


図6：各問題の総タスク数

図5において、平均分類精度が高い問題群と低い問題群に分ける。前者の問題では、後者の問題に比べて、ルール・セット中のルール個数および図6の総タスク数が多いことがわかる。前者の問題では、各機械上での入れ換え候補となるタスクの組み合わせが後者の問題よりも多く、カテゴリが多数存在する。このことから、同一の特徴を有しながらカテゴリの異なる事例群が発生しやすくなり、分類精度が悪くなることが考えられる。

5.2 ルールの適用

各々の問題に対して任意に発生させた初期スケジュールに、以下の操作を適用し、改善スケジュールを求める。

[Step 1] スケジュールを事例作成の場合と同様に、選択弧による数列表現に変換する。

[Step 2] ルールとのマッチングを行い、候補となるルールを全て列挙する。

[Step 3] マッチングに成功したルールの1つを適用し、2つのタスクを入れ換える。

[Step 4] スケジュールが実行可能かつその評価値が改善されていれば Step1に戻る。

[Step 5] 適用候補のルールがなければ Step6へ。あれば Step3に戻る。

[Step 6] 総所要時間が最小であるスケジュールを採用する。

各々の問題に対して、任意に発生させた50種類の実行可能スケジュールを作成し、得られたルール・セットを適用した結果を図6に示す。図6は、得られたルール・セットによるスケジュール f_0 と、加工時間の小さい順に加工すること（SPTルール；Shortest Processing Time）によるスケジュール f^* との平均相対偏差を表わす。 f_0 の平均相対偏差 $\text{dev}(f_0)$ は、式(5)により定義する。

$$\text{dev}(f_0) = \frac{f^* - f_0}{f^*} \times 100 \quad \dots (5)$$

この場合、正の偏差であれば、SPTルールによる解よりも良好な解である。

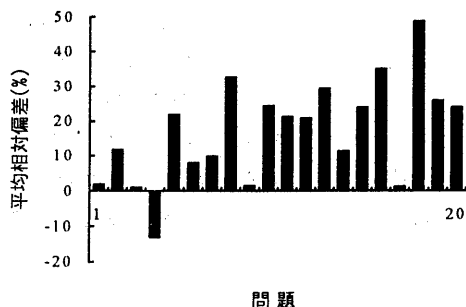


図6：SPTルールによる解との平均相対偏差

評価の対象となるスケジュールが初期スケジュールの良し悪しに影響するため、問題の規模に関係なく偏差にばらつきがあることがわかる。しかし、獲得したルールは、総所要時間最小化に効果的なSPTルールによるスケジュールよりも良好なスケジュールを導出することがわかる。

6. 結言

本論文では、ジョブ・ショップ問題を対象として、帰納的学習によるスケジューリング・ルール獲得法を提案し、その有効性を検討した。ルール獲得は、トレーニング・セットをサンプル・スケジュール上の2つのタスクを逐次的に

入れ換えることにより生成し、C4.5により分類木を求めることによって行った。結果をまとめると以下のとおりである。

- 1) タスクの位置情報に着目したルールの獲得は可能である。
- 2) 得られたルールは、スケジュールの改善に効果があることが確認された。

現在の事例作成法では、全てのタスクの位置情報を事例の特徴としているために、ルールの特性や適用場面が個々の問題に依存する。今後、仕事数、タスク数などに依存しないスケジュールの表現方法や、大多数の仕事を扱うことを前提とした事例作成の検討が必要である。

参考文献

- 1) 森下真一, 沼尾雅之, 戸沢義夫, "協調型スケジューリングによる製鋼工程スケジューリング・エキスパートシステム", 人工知能学会誌 Vol.5, No.2, pp.185-193, (1990)
- 2) M.J. Shaw, Heuristic Learning for Pattern Directed Scheduling in a Flexible Manufacturing Systems, *Proceeding of the 3rd ORSA / TIMS Conference on Flexible Manufacturing*, pp. 369-376, (1989)
- 3) 中須賀慎一, 吉田武稔, "概念学習における知識抽出を利用した製造ラインのダイナミック・スケジューリング" 第15回システムソフトウェア pp.367-372, (1989)
- 4) 藤井進, 諏訪晴彦, "2仕事入れ換え法に基づくスケジューリング・ルールの学習" *ISCIE第4回インテリジェントFAソフトウェア講演論文集* pp. 189-190, (1993)
- 5) J.R. Quinlan, Induction of Decision Tree, *Center for Advanced Computing Sciences Machine Learning*, pp.81-106,(1986)
- 6) J. R. Quinlan, *C4.5 : PROGRAMS FOR MACHINE LEARNING*, MORGAN KAUFMANN, (1993)
- 7) Jacek Blazewicz, K.Eccr, G. Schmidt, J. Weglarz, *Scheduling in Computer and Manufacturing Systems*, Springer-Verlag, (1992)
- 4) 藤井進, 諏訪晴彦, 住本直樹, "帰納的学習によるスケジューリング・ルールの獲得手法" 第37回ISCIE研究発表講演会講演論文集 pp. 71-72, (1993)