

帰納学習手法による戦略知識の獲得
- 運転整理エキスパートシステムにおける適用 -

荒木大† 小島昌一† 藤原裕二†† 池田克己†††

株式会社 東芝

† システム・ソフトウェア技術研究所 †† 交通システム開発部 ††† 府中工場

問題解決に有効な戦略知識を帰納学習技術を応用して獲得する手法を提案し、運転整理エキスパートシステムの知識獲得への適用を通して有効性を検証する。運転整理とは、列車に遅延が発生したときに運転ダイヤの変更を行う問題である。この問題では、状況を判断して整理対策を選択するノウハウが、遅延が収束するまでの時間がより短い整理案を作成する鍵となる。知識獲得は、戦略知識を持たない網羅探索型のオリジナルシステムを用いて、探索の成功例を構成するプリミティブな判断事例を集め、学習アルゴリズム INDECTS により戦略知識を生成する形で行った。獲得された戦略知識を実装することで、システムの探索能力の向上が実現された。

Acquiring Strategic Knowledge with Inductive Learning
- Experiment in Railroad Diagram Management Expert System -

Dai Araki† Shoichi Kojima† Yuji Fujiwara†† Katsumi Ikeda†††

TOSHIBA Corporation

† Systems & Software Engineering Laboratory
70, Yanagi-cho, Saiwai-ku, Kawasaki, Kanagawa, 210 Japan

†† Transportation Systems Sales & Engineering Department
1-1-1, Shibaura, Minato-ku, Tokyo, 105-01 Japan

††† Fuchu Works
1, Toshiba-cho, Fuchu, Tokyo, 183 Japan

We propose an acquiring method of the effective strategic knowledge with the inductive learning technique, and report an experiment of this idea for the railroad diagram management expert system. The diagram management means the diagram modification in such case that some delay occurs on one of the trains. The plan selection action with evaluating the current status is the key factor for making a good modification which has minimum total delay time. Knowledge acquisition has completed with the following steps. At first, we collected many examples of primitive plan selection from the successive solution searches. Next, the inductive learning system named INDECTS created strategic knowledge from the collected examples. The acquired knowledge produced a improvement of the problem solving power of original system.

1 はじめに

本章では、エキスパートシステム開発における戦略知識の獲得の重要性と、自動的にこれを獲得することを目的とした関連研究をまとめる。

「戦略知識とは、問題解決のある時点において適用条件が満たされるアクションが複数あるが、それぞれを実際に実行したときの正確な効果をあらかじめ測ることができないような状況で、どのアクションを選択すべきかを決定する知識である。」 [1, 2]

効率的な問題解決機構を実装する上で戦略知識の果たす役目は大きい。プロダクションシステムにおいては、戦略知識とは競合解消戦略を意味する。有効な戦略知識、すなわち競合解消規則が存在する場合には、状況を見極めて適切なアクションを行うルールを優先して選択することができ、解の探索効率を高めることができる。しかし、有効な競合解消規則が存在しない場合には、解探索は網羅的かつ盲目的に行なう必要が生じ、探索の後戻りに必要な記憶容量の制限によって解の発見そのものが不可能になる可能性もある。

既存の推論システムでは、開発者があらかじめ何らかの手段で競合解消規則を評価式のような形式で組み込んでおく必要があり、推論システム自体が競合解消規則を生成或いは修正する能力は無かった。しかも、有効な戦略知識は極めて専門的かつ発見的な知識である場合が多く、対象とする問題の解決を得意とする専門家にとっても、評価式あるいはルールのように形式的な形で表現することが著しく困難であった。また、外部から与えた戦略知識は、正当性が必ずしも保証されるものではなかった。

ASK [1, 2] は医療診断 ES である MUM の上に構築された、診断の手順に関する戦略知識の獲得を行うシステムである。ASK システムは、保存されている推論履歴を専門家に逐一提示し、なぜここでこの判断を行ったのかという支持理由を明らかにしようとする。このために、その段階では他にどんな診断仮説があったか、どんな診断テスト(薬の投与など)が考えられたのか、といった情報を提示するインターフェースを備えている。また、獲得した支持理由を、戦略知識と

して操作可能な形式に変換して、知識の一般化を支援する機能を持っている。しかしながら、このシステムは対話的に専門家から知識を聞き出そうとするものであり、システム自身が自律的に知識を生成する能力に乏しい。従って、専門家が戦略知識のある程度フォーマルな形式で所持している事が前提となる。

ここで取り上げる運転整理の問題においては、良い運転整理案を出せるかどうかは、複数の対策候補から1つの対策を選択する戦略知識の善し悪しにかかる。この戦略知識は、路線、ダイヤの形態に極めて依存する。しかしながら、運転指令員から体系的に知識を聞き出せるか、また、最終的な運転整理案を見据えて整理の途中段階で最適な判断を行う戦略知識を運転整理員が本当に持ち合わせているのかが疑わしい。

ここでは、良い運転整理結果を得るための戦略的な判断の事例を収集して、帰納学習により戦略知識を構成するための判断規則を自動生成することを試みた。

帰納学習の手法は、様々なアプローチから研究されている。ここで採用するアルゴリズムは、Quinlanが開発した ID3 [5] を報告者らが拡張した INDECTS [6] である。ID3 は事例を分類する知識を決定木の形で帰納学習するアルゴリズムである。INDECTS は数値データを扱える形に ID3 を拡張したものである。

2 運転整理エキスパートシステム

運転整理とは、事故や故障による列車ダイヤの乱れを正常状態に復帰させるために、計画運行ダイヤの変更を行う業務である。この問題は、大規模な組み合わせ問題であり、現在は運転指令員と呼ばれる専門家の経験的な判断にほとんど依存している。しかしながら、事由発生時にすみやかに正常状態に復帰できる整理案を短時間で作成するためには計算機による支援が不可欠となる [4]。

2.1 運転整理

「駅1で列車Aの発車が計画運行ダイヤよりも遅れた」という初期遅延情報から計画運行ダイヤ上でどのように整理を行うかを図1において示した。このよ

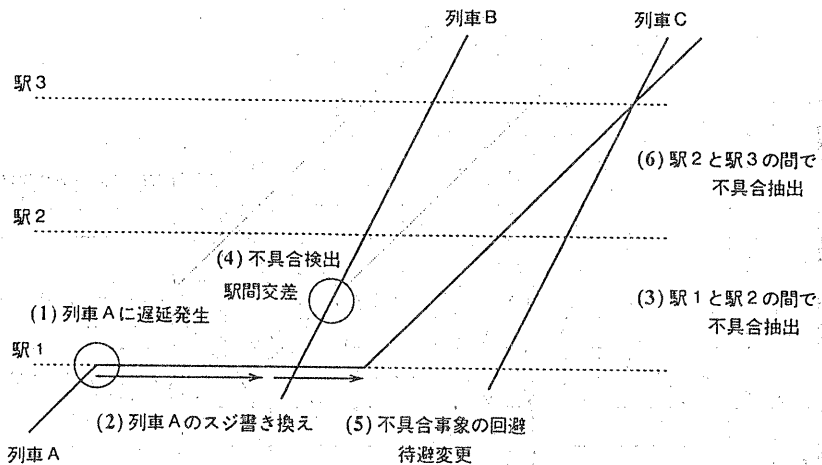


図 1: 運転整理の流れ

うな手順で順次ダイヤスジを書き換えてゆき、不具合が検出されなくなれば整理は完了したことになる。そこで、「駅1で列車Aの待避変更をする」といった個々の整理対策の総和が運転整理案となる。

このような手順を一般化すると、つぎの3つの手続きが問題解決のための基本タスクと見なせる。

1. 不具合事象の発見

正常な運行が不可能となる現象(以下で不具合事象と呼ぶ)を発見する

2. 対策の選択

不具合事象を解消するための対策を洗いだし、その中の1つを選択する。

3. スジの書き換え

選択した対策に従ってダイヤスジを書き換える。

不具合が発見されなくなるまで3つの基本タスクを繰り返すモデルが、運転整理 ES の問題解決方式となる。図2には運転整理 ES の基本的なシステム構成を示した。この中で、不具合抽出部、整理方法提案部、ダイヤ編集部が上で述べた3つの基本タスクを実行するモジュールである。自動運転整理マネージャーは最初に発生した事象をダイヤ編集部を介して運転ダイヤデータベースに書き込み、3つの基本タスクによる問題解決を開始させる。そして、不具合抽出部で不具合が発見できなくなるまで運転整理を実行させる。

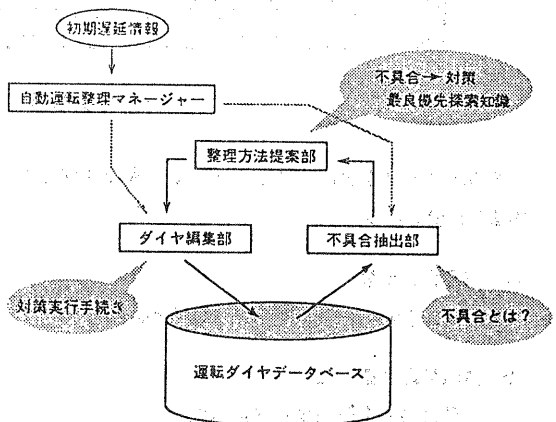


図 2: 運転整理 ES のシステム構成

システム [3] は汎用エキスパートシステム構築ツール ASIREX を用いて作成した。ただし、システムの大部分は C 言語でオリジナルに開発したものであり、整理方法提案部の知識のみをプロダクションルールで実装している。システムの実行画面を図3に示す。運転整理の実行は、画面上で初期遅延を設定して自動提案を起動させることで行う。運転整理結果は、運転整理を行った列車のスジが色を変更されて提示される。また、どの駅でどういふ対策を行ったかという運転整理の履歴をテキストの形で表示させることもできる。さらに、「次候補」ボタンにより他の運転整理案を第2案、第3案として順次求めることもできる。

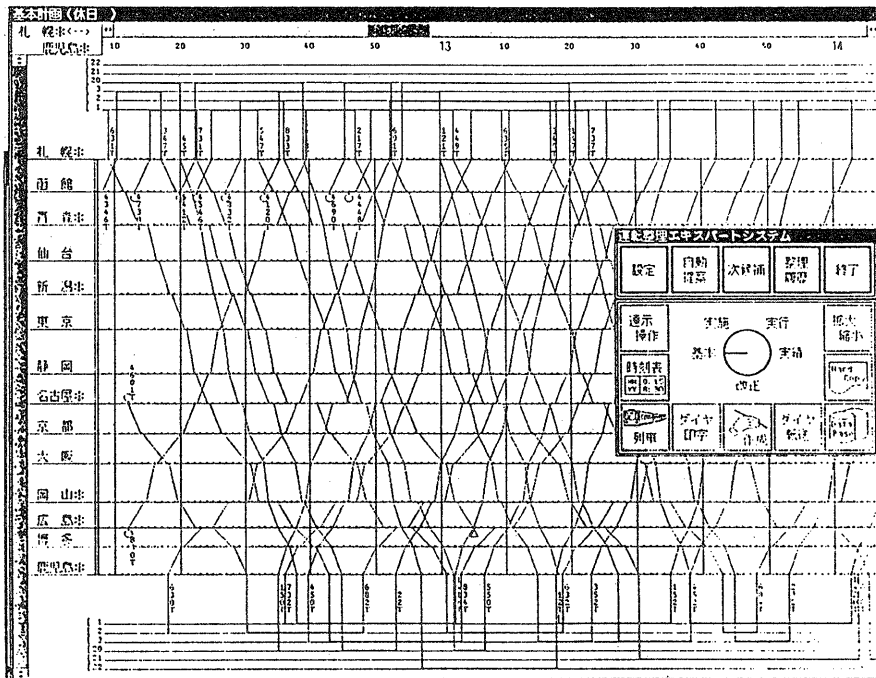


図 3: 運転整理 ES の画面構成

各基本タスクを実行してゆく上では、次の 4 種類の知識が必要となる。

1. 不具合の種類
不具合事象にはどんな種類があるか。
2. 運転整理対策知識
不具合事象を解消する対策は何か。
3. 最良優先探索知識
複数の対策が候補に上がった時にどれを選ぶか。
4. 対策実行手続き
選択した対策に対してどのようにダイヤスジを書き換えるか

この中で、「運転整理対策知識」と「対策実行手続き」は 1 対 1 に対応するものであり、ここではまとめて考えることにする。

2.2 不具合事象

不具合事象としては次の 4 つがある。

- 駅間交差：駅間で列車の追い越し(交差)が発生している状態

- 進入時隔異常：駅への進入の際の列車間の時間間隔が一定間隔以下になっている状態
- 進出時隔異常：駅からの進出の際の列車間の時間間隔が一定間隔以下になっている状態
- 番線競合：列車間で同一駅の同一番線が同時に使用されている状態
- 順序異常：駅を発着する順序が計画ダイヤと異なる状態

2.3 運転整理対策知識

不具合事象に対する対策は、現在までに次の 5 種類を実現している。「対策実行手続き」の知識は、対策を運転ダイヤデータベース上でのスジの書き換えとして実現する手続きを意味している。

- 待避変更：優等列車の待避駅を変更する。
- 時隔調整：到着または発車時刻を遅らせる。
- 番線変更：使用番線を変更する。
- 順序変更：発着順序を変更する。
- 運行振替え：2 列車の進行ダイヤを入れ替える。

不具合事象	状態	条件	対策	内容
駅間交差	ケース1 次駅 当駅 劣等 優等	1) 2) 劣等 3) 優等 4) 5) 待避可能 6) 番線空	#1 過避変更 #2 時隔調整	劣等列車の当駅発車時刻変更 劣等、優等列車の順序変更 番線競合の場合は番線変更 優等列車の次駅発時刻遅延
	ケース2 次駅 当駅 劣等 優等	1) 2) 劣等 3) 優等 4) 5) 待避不可 6)	#1 時隔調整	優等列車の次駅発時刻遅延
進入時隔異常	ケース1 当駅	1) 2) 3) 4) 5) 待避不可 6)	#1 時隔調整	後発列車の当駅発時刻遅延

凡例: 1) 異常の区分 2) 先発列車 3) 後発列車 4) 駅の種類 5) 待避 6) 番線

表 1: 運転整理対策知識 (一部)

また、現在のシステムでは実装されていないが、上の5種類以外に次のような整理対策が考えられる。これらについては、今後のシステムの拡張で順次取り入れて行く事を検討している。

- 運転打ち切り：計画ダイヤの終着駅より前の駅で運転を打ち切る。
- 運休：列車を運休させる。
- 臨時運転：遅延した列車に代わる新たな列車を運行させる。
- 途中折り返し：計画ダイヤの終着駅より前の駅で折り返す。

運転整理対策知識は、「不具合事象の状態を判断して、それに対して取り得る対策」というルール形式でまとめられている。表1にその一部を示す。

表1は、駅間交差の場合と進入時隔異常の場合の運転整理対策知識である。駅間交差に対する対策は2ケースあり、駅間交差が発生した箇所の前の駅で待避が可能な場合とそうでない場合がある。前者の場合には、劣等列車を臨時に待避させる待避変更と、優等列

車を遅らせる時隔調整の2通りが考えられる。後者の場合は、時隔調整しか対策は考えられない。

このように、運転整理対策知識だけで不具合事象の種類と状況から対策が一意に決まる場合とそうでない場合がある。運転整理対策知識は、不具合に対して物理的に実行可能な対策を結び付ける知識であり、良い運転整理案を出すためのノウハウといった、運転整理のための高度な知識を表現する物ではない。

2.4 オリジナルシステムの問題点

本システムでは、良い運転整理案を総遅延時間が短い運転整理案と考え、総遅延時間が短い運転整理案を出力することを目標とした。総遅延時間とは、すべての遅延が収束するまでに施した個々の整理対策において新たに加わった遅延時間の総計である。

しかしながら、オリジナルの運転整理システムでは、最良優先探索知識を実装していなかった。この知識は、運転整理対策知識を適用して洗い出した中でのどの対策を実行するかを判断する知識である。ここで行われる判断の積み重ねが、最終的に全ての遅延が収束するまでの総遅延時間を左右することになる。従って、良い運転整理案をシステムが早く出せるかどうか

の鍵は、最良優先探索知識の善し悪しにかかる。しかし、この知識は、路線、ダイヤの形態に依存する知識であり、運転指令員からこれを獲得することが著しく困難であった。

従って、オリジナルのシステムでは網羅的に解を探索する事により、最適な運転整理案の作成を行った。すなわち、「対策の選択」を行う際に、「運転整理対策知識」によって対策が一意に決まらない場合には、競合状態としてこの状態を記憶し、非決定的に対策を選択して運転整理を進めて行き、1つの運転整理案が出ると、競合状態に遡って残りの探索経路を試行することにより他の可能な運転整理案を出力するようにした。できあがった運転整理案を総遅延時間で比較することにより、最適解を判断するようにした。

この方式の問題点は推論時間がかかることである。込み入った運転ダイヤ上で運転整理を行うと、比較的短時間の遅延が発生した場合でも可能な運転整理案はかなりの数に上り、全ての解を出してから判断を行うのでは時間が掛かる。従って、解を出すまでの時間が数十秒までという実用的なシステムとして要求の実現は困難であった。

最良優先探索を行えば、必ずしも最適解を初めにみつけるとは限らないが、許容される時間内に質の高い解を発見できることが考えられる。そこで、専門家から直接知識を獲得することが困難である点を考慮し、帰納学習による最良優先探索知識の学習を試みた。

3 決定木学習アルゴリズム：INDECTS

ここで利用する基本的な学習アルゴリズムは、事例を分類する知識を決定木の形で帰納学習する ID3 [5] アルゴリズムである。しかしながら、オリジナルの ID3 では、事例を表現する属性はすべて記号で表現できる離散的な属性であることが要求され、数値で表されるような連続的な属性は扱えないという制約がある。我々は、数値で表された属性から例題の分類に適するように複数の区間に分割することによって離散的な新しい属性を生成するアルゴリズムを開発した。

INDECTS はこの数値区間分割方式と ID3 を組み

合わせた新しい決定木学習アルゴリズムで、記号で表現される質的な属性と数値で表現される量的な属性の双方を用いて評価を行う決定木を生成できる。すでに我々は ID3 で数値を扱うための従来研究と比較し、INDECTS の有効性を確認している [6]。

4 戦略知識の獲得

4.1 訓練事例の収集

運転整理対策知識を調べることによって、推論実行時に競合が発生する対策を抽出した。これは、2.3節で述べたように、不具合事象の種類と状況から対策が一意に決まらない部分である。

具体的には、運転整理対策知識の次の 5 カ所が戦略知識を必要とする。

1. 待避可能駅での進入時隔異常
対策は速度調整と順序変更の 2 種類がある。
2. 待避可能駅での進出時隔異常
対策は時隔調整と順序変更の 2 種類がある。
3. 待避可能駅での発車順序の順序異常
対策は時隔調整と順序変更の 2 種類がある。
4. 待避可能駅と次駅との駅間での駅間交差
対策は時隔調整と待避変更の 2 種類がある。
5. 待避可能駅で空き番線がある場合の番線競合
対策は時隔調整と番線変更の 2 種類がある。

競合状態が発生したときに、発生した競合状態と、そこでどのルールを選べば良かったのかという判断で構成される競合状況データを事例として保存できるようにした。そして、オリジナルのシステムを様々な初期設定から運転整理を実行させて、訓練事例を収集した。訓練事例の属性は競合状態の状況を示す次の項目で表現した。また、訓練事例の分類クラスは、可能なすべての整理案を作成した上で、個々の競合状態において総遅延時間が少ない方の結論に至る対策とした。

- 時刻 不具合の発生時刻

対策	時刻	駅	駅種類	番線	先着列車	後着列車	方向	待避駅	d1, d2, 1-2
速度調整	12:0:16	岡山	途中駅	4 在り	2003 急行	2004 特急	下り	1, 3	5, 5, 0, 等しい
順序変更	13:42:16	岡山	途中駅	4 在り	1022 普通	1020 急行	上り	3, 6	32, 76, -44, 遅延1
速度調整	12:8:16	広島	途中駅	0 なし	2001 普通	2003 急行	下り	2, 10	35, 35, 0, 等しい
順序変更	12:17:0	名古屋	途中駅	4 在り	2006 急行	2008 特急	下り	3, 4	29, 29, 0, 等しい
順序変更	13:34:16	岡山	途中駅	4 在り	1019 普通	1020 急行	上り	3, 6	27, 27, 0, 等しい

表 2: 進入時隔異常に対する事例データ (一部)

- 駅名 不具合が発生した駅名と種類
- 空き番線の有無
- 先着列車 先着列車の列車番号と列車種別
- 後着列車 後着列車の列車番号と列車種別
- 方向 列車の方向(上り, 下り)
- 次の待避駅までの駅数 当駅から次の待避駅までの駅数、2つ先の待避駅までの駅数
- 対策実施によって新たに発生する遅延 対策1によって発生する遅延(d1)、対策2によって発生する遅延(d2)、d1 - d2、d1 と d2 の大小関係

進入時隔異常に対する学習

次の待避駅までの駅数 $\leq 2 \Rightarrow$ 速度調整: 97% / 33

$2 <$ 次の待避駅までの駅数

駅名 == 鹿兒島 \Rightarrow 速度調整: 100% / 1

駅名 == 岡山 \Rightarrow 順序変更: 100% / 3

駅名 == 名古屋

$d1 \leq 29$ 秒 \Rightarrow 順序変更: 100% / 1

29 秒 $< d1 \Rightarrow$ 速度調整: 100% / 2

駅名 == 新潟

$d1 \leq 9$ 秒 \Rightarrow unknown: 0% / 2

9 秒 $< d1 \Rightarrow$ 速度調整: 100% / 1

(d1: 速度調整で生じる遅延)

4.2 最良優先探索知識の導出

ランダムに設定した約50通りの初期遅延から運転整理を実施し、5種類の競合状態に対して訓練事例を分割して収集した。それぞれの競合状態の発生頻度が異なるために訓練事例の数は一定しない。

(4) 待避可能駅と次駅との駅間での駅間交差 (5) 待避可能駅で空き番線がある場合の番線競合については、それぞれ「待避変更」「番線変更」の対策を良しとする事例が1つづつしか収集されなかった。これは、この2つの状態が発生することが希であることを意味する。例えば「駅間交差」については、使用したダイヤの各駅間が短いために、進入時隔異常や進入時隔異常として遅延が現われやすいことによると考えられる。そこで、両者については「待避可能駅と次駅との駅間での駅間交差の場合は待避変更」「待避可能駅で空き番線がある場合の番線競合が発生したら番線変更」を戦略知識として組み込むことにした。

表 3: 進入時隔異常に対する学習結果

「待避可能駅での進入時隔異常」に対する訓練事例(表2にその一部を示す)は43個収集できた。これに対してINDECTSで学習した結果が表3である。

対策の後ろに記した数字は、そのノードに属している訓練例題の個数と、ノードに属する訓練例題のうちで分類クラスがノードの解と一致する割合を示す正解率を意味する。

決定木を見ると、キーとなるのは「次の待避駅までの駅数」であることが判る。すなわち、「次の待避駅までの距離が短い(2駅以下)ときは速度調整とし、それ以上のときは駅ごとで対策を考える」というヒューリスティックを発見している。

判断が「unknown」となっている部分は、例題をこれ以上区別する属性が無かったからである。すなわち、これだけの属性ではこの部分に対する判定規則を作れないことを意味する。

進出時隔異常に対する学習

$d1 \leq 65$ 秒

次々待避駅までの駅数 ≤ 6

駅名 == 広島 \Rightarrow 順序変更 : 100% / 3

駅名 == 岡山 \Rightarrow 時隔調整 : 89% / 9

駅名 == 新潟 \Rightarrow 順序変更 : 89% / 9

駅名 == 青森 \Rightarrow 順序変更 : 100% / 5

$6 <$ 次々待避駅までの駅数 \Rightarrow 時隔調整 : 94% / 33

65 秒 $< d1 \Rightarrow$ 順序変更 : 82% / 51

($d1$: 時隔調整で生じる遅延)

表 4: 進出時隔異常に対する学習結果

「待避可能駅での進出時隔異常」に対する訓練事例は 110 個収集できた。これに対して INDECTS で学習した競合解消規則が表 4 である。

進出時隔異常に対しては、時隔調整を行ったときにどれだけの遅延が新たに加わるかが、判断の第一の鍵となる。さらに、INDECTS の数値データの区間分割機能により、判断の分かれ目が 65 秒であることが発見された。また、「次々待避駅までの駅数 ≤ 6 」「 $6 <$ 次々待避駅までの駅数」という判断基準は、実験対象の路線では、「2 つ先の待避駅が存在するか否か」という意味に解釈できる。このように、学習された決定木を見ることによって、新たな判断尺度を発見するという副次的な効果もある。

順序異常に対する学習

「待避可能駅での発車順序の順序異常」に対する訓練事例は 68 個収集できた。INDECTS で学習した競合解消規則が表 5(1) である。順序異常に対しては、駅ごとで判断規則が異なり、一般規則を発見できなかったことが判る。

表 5(2) は、より詳細な決定木を得られるように、最初の決定木で性能が悪い部分をさらに展開した物である。しかしながら、広島駅での判断で後発が特急であったときの判断、岡山駅での「順序変更で生じる遅延 == 時隔調整で生じる遅延」であったときの判断は unknown となり、結局決定木の全体性能を損なう

駅名 == 広島 \Rightarrow 順序変更 : 76% / 29

駅名 == 岡山

$d1 < d2 \Rightarrow$ 順序変更 : 100% / 3

$d1 == d2 \Rightarrow$ 時隔調整 : 77% / 17

駅名 == 名古屋

空き番線無し \Rightarrow 時隔調整 : 100% / 1

空き番線有り \Rightarrow 順序変更 : 100% / 1

駅名 == 新潟 \Rightarrow 順序変更 : 91% / 11

駅名 == 青森 \Rightarrow 時隔調整 : 100% / 6

(1)

駅名 == 広島

後発種別 == 普通 \Rightarrow 順序変更 : 100% / 1

後発種別 == 急行 \Rightarrow 順序変更 : 81% / 16

後発種別 == 特急 \Rightarrow unknown : 0% / 12

駅名 == 岡山

$d1 < d2 \Rightarrow$ 順序変更 : 100% / 3

$d1 == d2$

次々待避駅までの駅数 $\leq 3 \Rightarrow$ unknown : 0% / 14

$3 <$ 次々待避駅までの駅数 \Rightarrow unknown : 0% / 3

駅名 == 名古屋

空き番線無し \Rightarrow 時隔調整 : 100% / 1

空き番線有り \Rightarrow 順序変更 : 100% / 1

駅名 == 新潟 \Rightarrow 順序変更 : 91% / 11

駅名 == 青森 \Rightarrow 時隔調整 : 100% / 6

(2)

($d1$: 順序変更で生じる遅延, $d2$: 時隔調整で生じる遅延)

表 5: 順序異常に対する学習結果

結果になった。これは、このような場合に判断を行う良い属性がなかったためである。結局、最良優先知識としては (1) を採用した。

5 評価・検討

獲得した 5 種の戦略知識を、オリジナルシステムの最良優先探索知識として実装した。新たに初期事象を設定して運転整理案の作成を行ったところ、総遅延時間の観点で良い運転整理案が先に生成されるようになったことが確認された。つまり、必ずしも最適解を初めに出せるとは限らないが、最適解に近い方向から先に探索が行われるようになった。これは、訓練事例

として与えた初期事象に近い遅延を設定した時に顕著であった。

学習による知識獲得の効果

本実験により、戦略知識の獲得を、実用レベルのES開発における知識獲得の場で試行することができた。学習によって得られた戦略知識は決定木というフォーマルな形で表現されており、視覚的にもその内容が分かりやすいという利点がある。

このような学習で獲得された知識は、路線やダイヤの状況を反映した質の高い物であることが期待でき、専門家である運転指令員が持ち合わせていなかったノウハウまでをも発見できた可能性がある。学習により得られた知識を専門家にフィードバックすることにより、運転整理に必要な新たな制約を聞き出せたりするといった2次的な効果も期待できる。

運転整理に必要な知識の中で、不具合事象の種類や、運転整理対策知識は路線やダイヤには依存しない一般性の高い知識である一方で、戦略知識すなわち最良優先探索知識は路線やダイヤの状況に依存してその内容が異なる知識である。そこで、本システムを他の路線用に改造する、あるいはダイヤの変更があった時には、戦略知識だけをこの手法によって作り直すことで対応できる。今後、本システムをドメインシユル化する観点では、戦略知識の自動学習機能を付加することの意義は大きいと考えられる。

訓練事例の属性

INDECTSは最良優先探索知識を作る上で適当な属性を選び出す能力はあるが、これを作り出す能力は基本的にはない¹。従って、本手法による知識獲得を行うためには、訓練事例の属性はあらかじめ与えておく必要がある。

今回の実験では開発者らの話し合いにより、効果の期待できそうな属性を洗い出して学習を行った。しかしながら、得られた決定木の中には判定不可能の部分も見られた、この部分は、例題の数が少なかったこと

¹ただし、数値データを離散属性に変換する機能は、属性を作り出す能力と見なせる。

も一因であるが、良い属性が無かったことも原因である。今回使った属性は、不具合事象の当該列車に関する情報だけであったが、次の列車の接近状況や、近辺の列車密度といった、不具合事象が発生している回りの状況も対策を決める要因となったかもしれない。

従って、このような良い属性を見つけることが、運転整理のための知識獲得のキーとなる。これを専門家から聞き出すことは戦略知識そのものを聞き出すことよりも容易であると考えられる。つまり、運転整理対策を決定する上でどういった情報を考慮するのかという質問を運転指令員に行うことで属性のレパトリーを広げることができる。INDECTSは戦略知識を作る上で適当な属性を選び出す能力があり、これを効果的に活用した知識獲得が期待できる。

訓練事例の評価方法

今回の実験では、総遅延時間の最も小さい運転整理案を最も良い解であると見なした。この判断基準は電鉄会社ごとで異なることが予想される。解を評価する方法が正しくなければ訓練事例の正当性を損なうことにつながる。

実際には、「できるだけ特急列車に遅延が波及しないような運転整理案が望ましい」といった評価基準などが予想でき、総遅延時間以外に多重尺度で結果を評価すると考えられる。戦略知識の自動学習を行うためには、このような複数の判断基準を考慮して運転整理案を評価する評価関数を用意しておく必要がある。

しかしながら運転整理案を正しく相対評価できる評価関数を用意することは困難であると考えられる。そこで、訓練事例を作成する段階で、初期遅延に対して作成された複数の運転整理案を比較評価する作業を専門家の手に委ねることが考えられる。このような形で専門家に参加してもらうことにより、結果を評価するための判断基準を聞き出し、評価関数として形式化する作業を効率的に行えるといった効果も予想される。

訓練事例の数

質の高い戦略知識を学習する上では「良い属性」「正しい訓練事例の評価」と共に、「訓練事例の数」

が重要になる。今回の実験では、約50通りの初期遅延を設定して運転整理を行って訓練事例の収集を行った。戦略知識を学習するための訓練事例の数は、実際に INDECTS を使用した3種類の知識の学習で、43～110個である。今回使用したダイヤデータベースは、昼の2時間分ほどのダイヤしか実装しておらず、他の時間帯に対する判断規則を学習しきれていない。1日分の完全な運転ダイヤでの学習では、さらに数倍の訓練事例が必要となると考えられる。しかし、現在のところでは、この程度の訓練事例の数では、学習を行う上での INDECTS の性能上の問題はまったく無く、SPARC-LT 上で実装した INDECTS による学習に要する時間は数秒である。

決定木の中の判定不可能の部分は、この部分に対する例題の数が少なかったのも一因である。今回の手法では、運転整理案を作成する途中で例題が生成されるので、意図的に特定の状況の例題を作ることが困難である。この部分に対する例題が少ないことは、このような状況が発生することが希であることを意味する。特殊な状況のための規則を完備することのメリットがどこまであるかの判断が要求されよう。

また、今回の実験では、訓練事例収集のための初期遅延の設定は適当に行ったが、ダイヤサイズが時間ごとでパターン化している部分が多く、例題は時間帯ごとに典型例として作成することができることが判った。従って、例題の数が極端に少なすぎることはないと考えられる。

6 まとめ

帰納学習によって、遅延が収束するまでの時間ができるだけ短い運転整理案を生成するためのヒューリスティックを発見することができた。実際に学習された知識をシステムに組み込むことにより探索能力の向上を確認できた。

適用した知識獲得手法には、(1) 専門家が持ち合わせていなかったような新しい知識を発見できる可能性がある、(2) 獲得された知識が決定木という可読性の高い形式で表現される、(3) 実際の事例に基づいて学

習されたものであるために信頼性が高い、といったメリットがある。

本アプローチは、戦略知識の獲得に帰納学習の手法を用いて自動的に獲得しようとする点が従来アプローチと異なる点である。この様な知識獲得手法によって競合解消規則を生成あるいは修正する機能を実現することによって、有効な競合解消規則を持たない推論能力の低い推論システムが、推論の実行を積み重ねることにより解の探索効率を自律的に高めて行くことが可能となる。ここで試行した戦略知識獲得手法は、問題解決のための基本的なアクションだけを教示された状態から、試行錯誤を繰り返すことによって効率的な解探索のためのノウハウを獲得するという、人間の一学習形態をモデル化したものであるといえる。

現在、「運転打ち切り」「運休」といった対策を整理案として出せるように、システムの拡張を行っている。これらについても、新しい対策に対する「運転整理対策知識」「対策実行手続き」を実装した後に、最良優先探索知識の学習を行い、追加した対策に対する自動提案機能を完成させて行くことを考えている。

参考文献

- [1] Gruber, T. R. (1988). Acquiring Strategic Knowledge from Experts, *Int. J. of Man-Machine Studies*, 29, 579-597.
- [2] Gruber, T. R. (1989). A Method for Acquiring Strategic Knowledge, *Knowledge Acquisition*, 1, 255-278.
- [3] 原, 藤原, 牧口 (1990). 列車運転整理支援エキスパートシステムの開発. 電気学会全国大会, 6, 79.
- [4] 駒谷, 福田 (1988). 列車運転整理支援のエキスパートシステム. 人口知能学会誌, 3, 26-31.
- [5] Quinlan, J.R. (1985). Induction of Decision Trees, *Machine Learning*, 1, 71-99.
- [6] 荒木, 小島 (1991). 決定木学習における数値データの区間分割. 人口知能学会全国大会, 157-160.