

ルールと事例を用いた統合化推論の一手法

渡辺博芳[†] 奥田健三^{††}

ルールベース推論と事例ベース推論にはそれぞれ得失があるため、それらを統合してより高度な問題解決を行おうとする研究が数多く行われている。本論文では、1つの推論方式が失敗したら他方に切り換えるような弱い結合形態でなく、より強い結合形態について検討を行った。提案する推論方式は、ルールと事例を同じ形式で表現して1つの知識ベースに格納し、1つの推論エンジンがそれらの知識を用いて推論を行うことによって、部分問題ごとにルール、事例の選択が可能になる点に特徴がある。このようなルールと事例を用いた推論の枠組みを提案し、電力系統事故時復旧問題における計算機シミュレーションによって評価した。その結果、異なる問題条件に対しても、事例不足をルールで補完したり、反対にルール不足を事例で補完するなど柔軟な推論を行えることが分かった。また、CBRが失敗した場合にRBRで解を得るような両者の弱い結合形態に比較して、本手法は全体として処理時間を大幅に短縮することを明らかにした。

An Integrated Reasoning Method Using Rules and Cases

HIROYOSHI WATANABE[†] and KENZO OKUDA^{††}

Since both of the rule-based reasoning (RBR) and the case-based reasoning (CBR) have their own advantages, a number of methods for combining RBR and CBR have been proposed in the literature. In this paper, we propose a tighter combination method instead of a loose combination method which switches to another reasoning method when one reasoning method fails. The proposed method can select rules or cases for each sub-problem by representing rules and cases in the same form, saving them to the same knowledge-base and performing reasoning processes by one engine. The simulations in the domain of electric power systems show that our method performs a flexible reasoning, while supplementing the lack of rules by using cases, or vice versa. It is also demonstrated that the proposed method reduces the processing time drastically compared with a loose combination method, which switches to RBR when CBR fails.

1. はじめに

ルールベース推論 (RBR) と事例ベース推論 (CBR) は相補的な関係にあることが指摘されており¹⁾、現実の問題においては、ルール、または事例のどちらか一方で問題解決のための知識をすべて表現することが困難な場合が多い。たとえば、我々は事例ベース推論を用いてプレス加工工程設計支援システム²⁾を開発したが、この分野ではこれまでに計算機による設計支援の試みがほとんどなく、専門家からのルール獲得の難易に応じて、事例とそれを修正するためのルールを用いることが有効であった。Chiら³⁾は、応用分野によって使用可能な知識が異なっており、領域知識 (ルール) も領域の経験 (事例) も豊富でない分野では、RBR

と CBR の2つのエージェントが協調した推論システムが有効であると指摘している。また、一般的に問題解決システムはプロトタイプが完成した後もつねに発展途上にあり、ルールを新しく定義したり、それまでのルールでは対応できないケースに対処する必要がある。ルールと事例を同じ枠組みの上で扱えれば、このような問題に比較的容易に対応することができる。

本論文では RBR と CBR の類似性に基づいた統合化の一手法⁴⁾を提案し、電力系統事故時復旧問題に対する応用を通して評価を行う。電力系統事故時復旧におけるルール⁵⁾は、宣言的というよりは手続き的であり、ルール数は比較的少数であるが、1つのルールが何度も適用されるので、RBRによる問題解決には時間を要する。事例の役割が探索の効率化にあることは松原⁶⁾によって指摘されているが、以前、我々はこの問題に対して CBR^{7),8)}を応用することにより、問題解決の効率化を図った。そこでは、最初に CBR を適用し、解が得られない場合に RBR を適用するような

[†] 帝京大学理工学部

School of Science and Engineering, Teikyo University

^{††} 作新学院大学経営学部

Business School, Sakushin Gakuin University

枠組みをとっていた。本論文で提案する推論方式では、ルールと事例を同一の形式で表現し、1つの推論エンジンがそれらの知識を扱うことによって、部分問題ごとにルール、事例を選択することを可能にする。それによって、一方が失敗したら、他方に切り換えるという弱い結合形態でなく、問題解決のレベルでのRBRとCBRの相互補完が実現できると考えられる。さらに、様々な条件で問題解決を行ったとき、弱い結合形態に比較して、全体として効率的な問題解決が行えることが期待される。以下、2章では、RBRとCBRの統合化推論方式の基本的な枠組みについて述べ、3章で実現システムを説明する。4章以降で、電力系統事故時復旧問題を対象としたシステムによる実験結果および関連研究について述べる。

2. 推論方式の基本的な枠組み

2.1 ルールベース推論と事例ベース推論の類似性

ルールと事例を同じ枠組みで扱うために、(1)知識表現と(2)推論エンジンに関して、RBRとCBRの類似性に基づいた統合を試みる。

(1) 知識表現の類似性

図1に示すように、ルールは一般に条件部と結論部から構成される。多くの場合、条件部は条件要素が連言的に接続されており、結論部は結果である事実を記述する場合と実行する手続き名を記述する場合がある。条件部から外部関数を呼び出せるものも多い。それに対して、事例は一般に問題記述と解記述から構成される。解を導くための解法を含むこともある。事例は一般に大きなチャンクであり、フレーム形式で表現されることが多い。したがって、問題記述と解記述は

属性-値の組となる。これらを比較すると、ルールの条件部と事例の問題記述、ルールの結論部と事例の解記述がそれぞれ対応する。そこで、前者をLHS、後者をRHSとして表現の統合を図る。

(2) 推論エンジンの類似性

CBRの推論エンジンは一般に複雑であり、システムによって大きく異なる。河野ら⁹⁾はCBRシステムのタイプについて考察を行っている。知識と問題の照合をとる過程(recognize-cycle)と知識を適用する過程(act-cycle)の2つに大きく分けるとすれば、RBRとCBRの推論エンジンは図1(b)のように対応付けることができる。RBRにおいては認識-行動サイクルが解を得るまで、あるいは一定の条件を満たすまで繰り返されるが、CBRにおいて図1(b)の推論過程は多くの場合1つの問題解決において1回である。本推論モデルでは、認識-行動サイクルのような基本となる推論のサイクルを定義し、問題を解決するまで、あるいは問題解決の失敗が明らかになるまで繰り返すものとする。

2.2 知識表現

本推論方式では、ルールと事例をフレーム形式で表現する。フレームはフレーム識別子、タイムタグ、スロットリストからなる。スロットはフレームの属性を表し、スロット名、スロット型、スロット値、その他の複数のファセットから構成される。また、フレームのひな型はクラスフレームと呼ばれるフレームに定義し、クラスフレームのスロット値をデフォルト値として用いる。このようなフレームのうち、特にルールと事例などの判断に関する知識を表すフレームをkframe(knowledge frame)と呼ぶ。kframeは、goal、name、lower-priority-than、LHS、RHSなどのスロットを持つ。goalはそのkframeが解決しようとする問題、具体的には値を求めようとするスロット名である。lower-priority-thanは、当該知識よりも優先されなければならないkframeの識別子を表す。ルール間の競合の解消を行う手法としてルールの重要度をあらかじめ与えておく方法があるが、ここでは必要に応じてルールの重要度を相対的に指定する。これは必ずしも存在する必要はない。個々のスロットには、表1のようなファセットを指定することができる。

2.3 推論エンジン

RBRでは前向き推論と後向き推論があり、それぞれに長短があるが、本推論方式ではそれらの中間的な推論法をとる。すなわち、各kframeにそのkframeが解決しようとする問題(スロット名)を目標として与えておく。ここで、kframeのRHS部は必ずしも目標

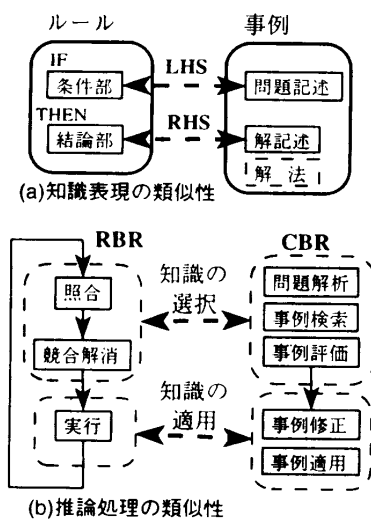


図1 RBRとCBRの類似性

Fig. 1 Similarities between RBR and CBR.

表1 ファセット
Table 1 Facets.

ファセット名	説明
weight	類似度計算時の重み
operator	not, =, <, >, ≥, ≤ など
matching-requirement	照合要求 (perfect, similar)
matching-strategy	照合戦略 (3.5 節参照)
application-strategy	適用戦略 (3.6 節参照)
if-needed	if-needed デーモン名

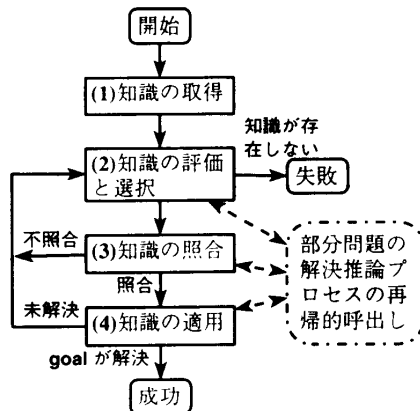


図2 推論処理

Fig. 2 Reasoning processes.

と同じでなくともよい。ある問題が与えられたとき、その問題を目標とする複数の kframe を用いて推論を行う。使われる kframe がルールであれば、前向き推論が行われる。このような推論のユニットを目標駆動で実行する。本推論方式の流れを図2に示す。

(1) 知識の取得

まず、与えられた目標をスロット goal の値として持つ kframe を取得する。

(2) 知識の評価と選択

取得した kframe の評価を行い、適用する kframe を選択する。ここでは、たとえば、目標、すなわち、部分問題に応じてルール優先、事例優先などの戦略をあらかじめ決めておくことも考えられる。また、事例の類似度を計算し、あるしきい値以上の類似度を持つ事例が存在しない場合は、ルールを用いる方法もある。ルール間の競合は先に述べた相対的な優先度の指定によって解消するが、その指定がない場合は任意のルールが選択される。選択すべき kframe が存在しない場合、推論は失敗する。

(3) 知識の照合

選択された kframe の適用可能性を調査するために、その kframe の LHS と現在着目している問題記述フレームの詳細な照合を行う。個々の

スロットの照合において、ファセット matching-requirement の値が perfect なら完全照合, similar なら類似性に基づく照合を行う。

(4) 知識の適用

詳細な照合の結果、kframe が適用可であれば、その kframe の RHS を適用する。その結果、目標とするスロットの値が得られれば推論は成功するが、目標とするスロットの値が得られない場合は (1) に戻る。ここで、残りの kframe の候補から選択するのではなく、最初に戻るのは、kframe の適用によって問題記述フレームの状態が変化するためである。

このような推論処理の知識の評価、知識の照合、知識の適用の過程で、値が未決定のスロットが存在すれば、そのスロット名を副目標として、それを目標を持つ kframe を用いて推論を再帰的に行う。また、知識適用において、事例修正に対応するような kframe の修正処理は行わないものとする。

3. 実現したシステムの推論処理

3.1 概要

2章で述べた基本的な枠組みにしたがって、実現した実験的なシステムについて述べる。実現したシステムにおいて、知識の評価と選択では知識 (kframe) と問題の予備照合を行い、類似度を計算することによって、最も類似度の大きい kframe を選択することとした。また、電力系統事故時復旧問題における手続き的なルールを表現するために、単に kframe と問題記述フレームを比較するだけの照合では十分でない。そこで、kframe の各スロットのファセットとして照合戦略と適用戦略を設けて、より複雑な照合処理を行えるようにした。

3.2 問題解決器

問題解決器の役割をあるフレームのあるスロットの値を求めることと定義する。本モデルはフレームシステムを基礎としているので、値の求め方として、if-needed デーモンによる方法と推論による方法がある。実現システムでは前者を優先することとし、問題解決過程は図3のように表される。ここで、goal は目標、pframe は問題記述のフレームである。与えられた goal に対するデーモンが存在するときは、Call (demon, pframe) により、デーモンを呼び出してスロットの値を得る。デーモンが存在しないとき、または、デーモンにより値が得られないときは、推論エンジンを起動する。

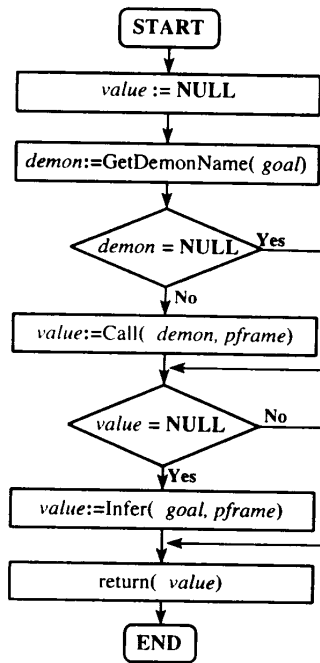


図3 問題解決器 Solve(goal, pframe) のアルゴリズム
 Fig. 3 An algorithm of the problem solver Solve(goal, pframe).

3.3 推論エンジン

図3における Infer() が推論エンジンが行う処理である。その処理内容を図4に示す。まず、GetKnowledge() では目標が goal と等しい kframe の識別子のリストを cclist に得る。次に PreMatch() では、cclist 内の kframe の LHS が参照するスロット値が pframe でいまだ与えられていない場合に、3.4 節で述べる戦略に応じてそのスロット値を求めた後、cclist 内の各 kframe と問題 pframe との類似度を計算する。そして、SelectKframe() で類似度とスロット lower-priority-than の値を考慮して、cclist から kframe を1つ選択し、それを kframe とする。kframe を cclist から削除する。Match() では、kframe の LHS と問題 pframe の間でバックトラックをとる詳細な照合をとり、結果を論理値で返し、result に代入する。予備照合 PreMatch() から詳細な照合 Match() までの処理は照合が成功するか、cclist が空 (NULL) になるまで繰り返す。cclist が NULL の時は適用可能な知識が存在しないので、推論による解決は失敗し、値として NULL を返す。照合が成功した場合は、kframe の RHS を問題に適用する。ここで知識の適用に成功したとしても、goal の値が得られるとは限らない。そのようなとき、ApplyKnowledge() は NULL を返す。以上の処理を goal の値が得られるまで繰り返す。

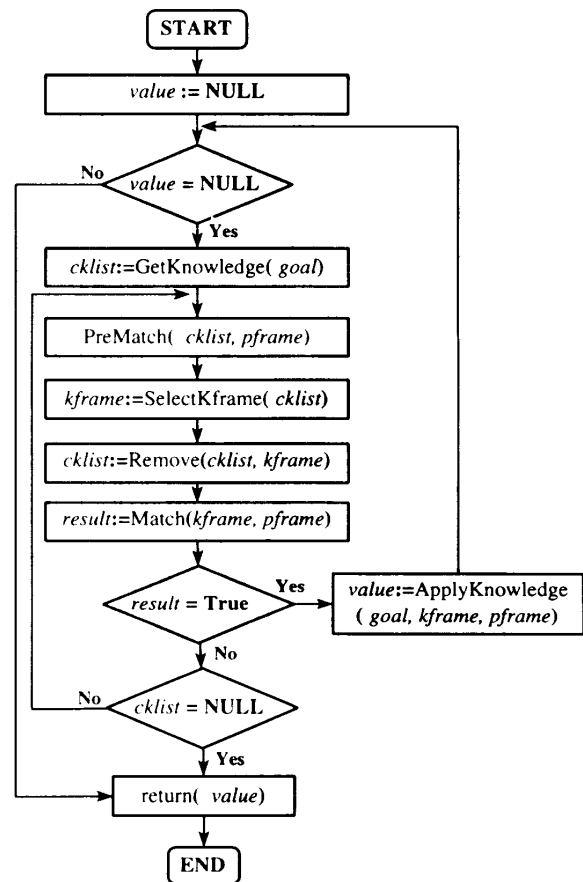


図4 推論エンジン Infer(goal, pframe) のアルゴリズム
 Fig. 4 An algorithm of the inference engine Infer(goal, pframe).

3.4 予備照合

図4の PreMatch() では、必要に応じて副問題を解決し、予備照合を行うことによって、各 kframe の類似度を計算する。知識を k 、問題を p とすると、 j 番目の知識と問題の類似度は次の式で計算する。

$$Sim(k_j, p) = -\sqrt{diff1(k_j, p) + diff2(k_j, p)} \quad (1)$$

ここで、 $diff1(k_j, p)$ は、 k に存在するスロットについての差であり、 $diff2(k_j, p)$ は k に存在しないスロットの差である。 k に存在しないスロットの差を考慮するのは、一般にルールの方が事例よりもスロット数が少ないために、類似度が大きくなる傾向があるので、それを調整するためである。

k に存在するスロットの差の計算では、 p の同名のスロットと operator ファセットを考慮して比較を行い、照合か不照合かを判定する。 k に存在するスロットが p に存在しないとき、または、 k のスロット値が変数であるときは、未照合スロットと判断する。 $diff1(k_j, p)$ は次のように計算する。

$$\text{diff1}(k_j, p) = \sum_{i=1}^{n_j} w_i \times d(k_{ji}, p_i) \quad (2)$$

ここで、 n_j は知識 k_j のスロット数、 w_i は i 番目のスロットの重みである。また、 $d(k_{ji}, p_i)$ は、数値スロットに対して、

$$d(k_{ji}, p_i) = \begin{cases} 0 & \text{when matching} \\ d_{num}(k_{ji}, p_i) & \text{when mismatching} \\ 0.25 & \text{otherwise} \end{cases} \quad (3)$$

ここで、

$$d_{num}(k_{ji}, p_i) = ((k_{ji} - p_i) / (\max_i - \min_i))^2 \quad (4)$$

である。 \max_i 、 \min_i は、それぞれ i 番目のスロットの値の最大値と最小値を表す。数値以外のスロットに対しては、

$$d(k_{ji}, p_i) = \begin{cases} 0 & \text{when matching} \\ 1.0 & \text{when mismatching} \\ 0.25 & \text{otherwise} \end{cases} \quad (5)$$

となる。

一方、 $\text{diff2}(k_j, p)$ は次のように計算する。

$$\text{diff2}(k_j, p) = w_j \times m_j \quad (6)$$

ここで、 w_j は j 番目の k frame に存在しないスロットに対する重みであり、 m_j は p に存在して、 k に存在しないスロット数である。 w_j は w_i に対して小さい値にすべきである。今回の電力系統事故時復旧問題では、ルールは中間仮説を含むようなルールであり、いわゆる事例の一般形であるルールは用いていないので、 $w_j = 0$ として実験を行った。

k frame の LHS スロットのオペレータが *not* であるとき、上で述べた類似度が大きいほど、その知識が採用される可能性は小さい。すなわち、否定により、類似度の意味が逆になるので、*not* が指定されている場合は類似度の逆数をとる。

最後に *cklist* の知識のうち、*matching-requirement* ファセットの値が *perfect* であるスロットが1つでも不照合であり、スロット LHS の *operator* ファセットの値が *not* でない知識は照合する可能性がないので、これらを *cklist* から削除する。

3.5 詳細な照合

図4の *Match()* では、知識と問題の詳細な照合をとる。個々のスロットの照合は、まず、知識のスロットが問題に存在しなければ、そのスロットを副目標として、*Solve()* を呼び出す。そして、(1) スロットの型が変数の場合は、その変数が変数と値のテーブルに登

録されているかを調べ、もし登録されていれば、そのテーブルに登録されている値と問題のスロット値を比較する。もし登録されていなければ、その変数と問題のスロット値のペアを変数と値のテーブルに登録する。バックトラックが生じた際には、登録した変数と値のペアをリストから削除し、他の組合せを探る。また、(2) スロットの型がフレーム識別子の場合は、照合戦略に基づいてサブフレームの照合を行う。(3) それ以外の場合は、*operator* ファセットを考慮して照合、不照合を判定する。

サブフレームの照合戦略には、*match*、*retrieve*、*generate* がある。戦略 *match* は、一般的な戦略であり、知識のサブフレームと問題のサブフレームを上で述べた方法にしたがって、再帰的に照合をとる。戦略 *retrieve* は、知識のサブフレームのスロット値を満たすようなフレームを検索し、存在すれば照合は成功する。知識のサブフレームに値の決まっていない変数があれば、検索で得たフレームとの照合により、単一化される。図5は、電力系統事故時復旧問題において、系統の接続情報などをもとに復旧用電源の最初の候補を生成するルールである。LHS フレームの *connect-point* の照合戦略は *retrieve* であるので、*failure-subsystem-name* で単一化された変数 [FSYS] を用いて *connect-point* クラスのフレームを検索する。具体的には、スロット *subsystem* の値であるサブフ

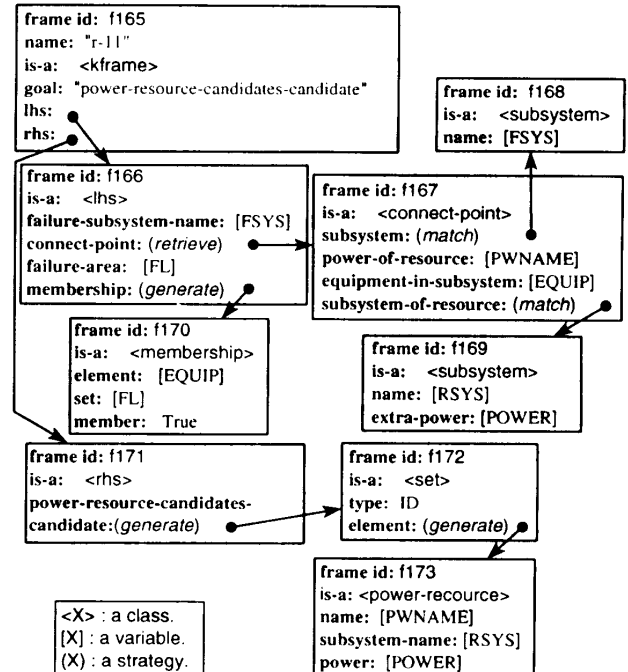


図5 電力系統事故時復旧問題におけるルールの例
Fig. 5 An example of a rule for fault restoration in electric power systems.

フレームが subsystem クラスに属しており、そのサブフレームの slot name の値が [FSYS] であるような connect-point クラスのフレームを検索する。そして、検索の結果得たフレームとフレーム f167 の照合を行い、残りの変数、すなわち、[PWNAME], [EQUIP], [RSYS], [POWER] を単一化する。戦略 generate は、知識のサブフレームの値の slot のうち、単一化された変数を用いて新しくフレームを作成した後、そのフレームと知識のサブフレームの照合をとる。図 5 の LHS フレームの membership の照合では、まず、値の決まっている変数 [EQUIP] と [FL] を用いて、membership クラスのフレームを生成する。次に生成されたフレームには slot member が存在しないので、member を副目標として値を求めた後、生成されたフレームとフレーム f170 の照合をとる。すなわち、生成されたフレームの member の値が True であれば、照合は成功する。

全体として、照合が成功した場合、類似度を改めて計算しなおし、類似度があるしきい値以上なら、結果は成功となる。

3.6 知識の適用

知識の適用は、基本的には知識 *kframe* の RHS の値であるフレームに存在する slot とその値の組を問題 *pframe* に登録することである。ただし、*pframe* 以外のフレームに slot を追加したり、あるフレームからある slot を削除することも可能である。これらは、action クラスのフレームによって行う。slot 値がフレーム識別子である場合の適用戦略は次のとおりである。

- 適用戦略が perform のときは、action クラスのフレームを実行する。
- 適用戦略が generate なら、問題に対して知識と同様のフレームを新たに作成する。
- retrieve なら、知識のフレームの slot 値を用いて、問題のためのフレームを検索によって得る。

RHS フレームの slot の型が変数であるものは、詳細な照合を行ったときの変数と値のテーブルの値を *pframe* の値とする。それ以外では、知識の slot 値をそのまま登録する。目標と同名の slot が *pframe* に登録された場合はその値を返す。図 5 のルールでは、slot 名、power-resource-candidate-candidates と、その slot 値、すなわち、power-resource クラスのフレームを element の slot 値として持つような set クラスのフレームが *pframe* に登録される。

4. 実験結果

4.1 実験条件

電力会社の 1 つの支店が管轄する程度の広がりを持つ 60 kV 電力系統における事故時復旧支援を対象として、シミュレーションを行う。ここでは事故による停電の影響が大きく、復旧操作が複雑になると予想される各変電所内の高圧側母線における永久事故を扱う。復旧案作成にあたっての基本的な方針は、停電電力の最小化、停電時間の短縮、最適な復旧案の生成にある。最適な復旧案としての判断基準は種々考えられるが、今回の実験では (1) 遮断器の操作回数が最少、(2) 復旧用の電源の個数が最少、とする。そして、10 個の部分系統からなる 60 kV 電力系統 (負荷ブロック数 52) のうちの 1 つの部分系統を選択し、系統条件を変更してシミュレーションを行った。

対象とした部分系統 C を図 6 に示す。この図において、bk-x は変圧器バンクを表す。また、L-x で示す負荷ブロックとは遮断器の投入により充電の及ぶ範囲に含まれる送電線、負荷、母線をまとめたものである⁵⁾。負荷ブロックと b-x で示す母線の間には遮断器が存在し、開放されている遮断器を ×印で表現し、投入されている遮断器の表示は省略した。開放された遮断器は部分系統間の接続点となる。系統間の接続点付近に記された矢印と数字 (MW) は各部分系統からの送電可能電力を表す。図 6 は C 系統の標準的な系統構成であり、ここで高圧側母線 b-4 に永久事故が生じると、母線 b-13A, b-14, b-15 と負荷ブロック L-7, L-8, L-9, L-10, L-11, L-12 が停電となる。このとき、復旧用電源として、部分系統 A の L-4, 部分系統 D1 の b-13B, 部分系統 F1 の b-25 が候補となる。負荷の状態をピーク時に対する % で表したものを負荷レベルと呼ぶが、復旧に必要な電源数は負荷レベルに応

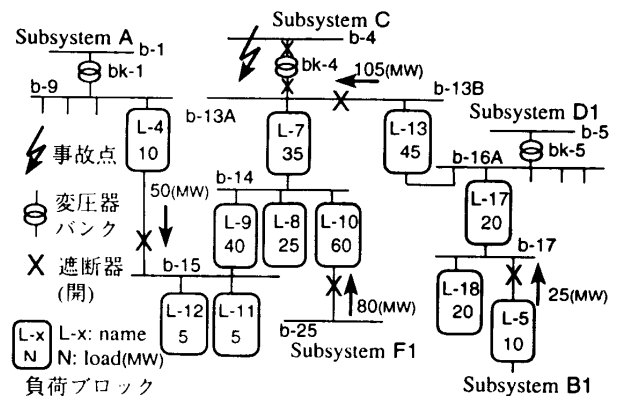


図 6 対象とする部分系統

Fig. 6 Target subsystems.

じて異なる。さらに重負荷時には、これらの電源では停電が解消されない可能性もある。このような場合は、停電範囲に隣接する部分系統の負荷の一部をそれに隣接する他の部分系統に切り換えることによって、停電範囲への送電電力の増加を図る。たとえば、図6の部分系統D1に属するb-17とL-18をD1に隣接する部分系統B1に切り換えることによって、D1からの送電電力を105 MWから125 MWに増加できる。このような操作を負荷切り換えと呼ぶ。全体復旧案は、負荷切り換え数*i*と復旧用電源数*j*を用いて*iCjR* (*i*=0なら単に*jR*)で表す。これを復旧パターンと呼ぶ。

3章で述べたシステムをワークステーション (SONY NWS-3865) 上にC言語を用いてインプリメントした。電力系統事故時復旧のための39のルールをフレーム形式で定義し、手続き的な処理を行う16のデモンを用意した。ただし、これら39のルールには負荷切り換えを行うルールは含まれていない。また、C系統の標準的な系統構成における事故に対して、負荷レベルに応じた典型的な4つの事例を用意した。具体的には復旧パターンが3R (3電源)、2R (2電源)、1R (1電源)が2個である。知識ベースの内容によって、RBR、CBR、本推論方式を実現して、それらの比較を行った。RBRは39のルールを持つ。CBRは問題解析に関連する13のルールと4つの事例を持ち、事例修正機能はない。本推論方式はCBRと同じ4つの事例とRBRと同じ39のルールの他に以下のような2つのルールを持つ。

- Rule 40
もし、復旧用電源候補に存在しない電源からの復旧案が存在するならば、その復旧案を削除する。
- Rule 41
もし、全体復旧案候補の復旧設備リストに停電範囲に存在しない設備名が存在するならば、その設備名を削除する。

以降、本推論方式をINTERCARE (INTEgrating Rule-based and CAsE-based REasoning)と呼ぶ。それに対して、CBRが失敗したら、RBRによって解を得るような両者の弱い結合形態をCBR+RBRと表すことにする。

系統構成や負荷レベルの変化、負荷の増加など、現実的な条件変化を想定して、シミュレーションを行った。たとえば、電力系統では、設備の保守点検や季節ごとの負荷分布に応じて、系統構成を変えることがある。そこで、標準的な系統構成をconfig0とし、図7のような2つの系統構成を考える。(a) config1はb-15、L-11、L-12の負荷をA系統に切り換えて運転してお

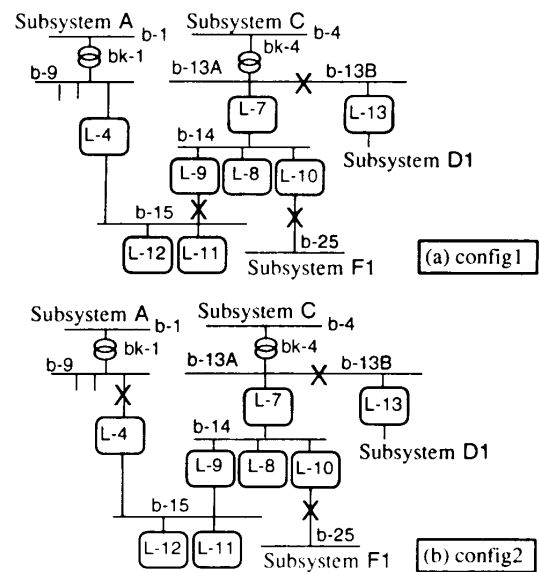


図7 config1およびconfig2の系統構成
Fig. 7 System configurations of config1 and config2.

り、(b) config2はA系統のL-4をC系統に切り換えて運転している。

また、電力系統では、1日のうちの時間や季節によって、負荷のレベルがピークからその約50%程度の間で変動することから、その範囲でいくつかの異なる負荷レベルを設定した。

4.2 実験結果と考察

(1) RBRとCBRの相互補完

まず、INTERCAREのルールと事例を用いた問題解決過程を調べるために、いくつかのケースについて問題解決過程をトレースする。

(a) 系統構成 config1 で負荷がピーク時にb-4に事故が生じた場合

事故地点が与えられると、いくつかのルールによって、事故系統、停電範囲などを同定する。停電範囲は[b-13A, L-7, b-14, L-8, L-9, L-10]となる。復旧用電源候補を求めた後、全体復旧案の候補を求めるときに、復旧パターンが3Rの事例が選択される。それによって、次のような復旧案候補が作成される。

- 電源 b-13B (D1 系統) 復旧リスト [b-13A, L-7, b-14, L-8, L-9]
- 電源 b-25 (F1 系統) 復旧リスト [L-10]
- 電源 NULL (A 系統) 復旧リスト [b-15, L-11, L-12] (事例の電源 L-4 は復旧用電源候補に含まれないので、NULL となる。)

次にRule40によって、3つめの復旧案が削除される。すると、停電範囲がすべて復旧されるので、2Rの復旧案が得られる。

(b) 系統構成 config2 で負荷がピーク時に b-4 に事故が生じた場合

停電範囲は [b-13A, L-7, b-14, L-8, L-9, L-10, b-15, L-11, L-12, L-4] となる。(a)の場合と同様に CBR により上のような復旧案が仮に作成され、Rule40 により3つめの復旧案が削除される。しかし、停電範囲がすべて復旧されていないので、全体復旧案候補に含まれない電源、すなわち、b-9 からの復旧案を RBR により求め、上の全体復旧案候補と組み合わせることにより、次のような 3R の復旧案を得る。

- 電源 b-13B (D1 系統) 復旧リスト [b-13A, L-7, b-14, L-8, L-9]
- 電源 b-25 (F1 系統) 復旧リスト [L-10]
- 電源 b-9 (A 系統) 復旧リスト [L-4, b-15, L-11, L-12]

(c) 負荷の増加

図6の系統構成において、電力の需要が増えたため、各負荷ブロックが2(MW)ずつ増えたと仮定しよう。この場合は、3つの電源を使っても復旧できないので、負荷切り換えを行う必要がある。しかし、実験システムでは負荷切り換えに関する知識をルール形式で与えていないので、現状では RBR で解を導くことはできない。ルールを獲得するのに対して、事例を獲得するのは比較的容易である。INTERCARE がルールの不足を事例で補完できることを示すために、D1 系統の負荷ブロックを D1 系統に隣接する B1 系統に切り換える 1C3R の事例を知識ベースに登録しておくこととする。これにより INTERCARE は次のような解を得る。

- 電源 b-13B (D1 系統) 復旧リスト [b-13A, L-7, b-14, L-8, L-9]

D1 系統に対する負荷切り換え案

電源 L-5 (B1 系統) 復旧リスト [b-17, L-18]

- 電源 b-25 (F1 系統) 復旧リスト [L-10]
- 電源 b-9 (A 系統) 復旧リスト [L-4, b-15, L-11, L-12]

以上で述べたとおり、INTERCARE では、(a)、(b)のように事例の一部にルールで導いた部分解を組み合わせたり、(c)のようにルールの不足を事例で補完することができる。すなわち、一方の推論方式が失敗した場合、他方を用いるのではなく、問題解決過程で RBR と CBR が協調して推論を行うことが分かる。また、部分問題に事例の解を用いるようなケースもあり得るが、そのような場合の動作例は文献4)で述べている。

(2) 問題解決処理の効率化

標準的な系統構成 config0 について 20 ケース、config1 と config2 について 15 ケースずつ、全部で 50 ケースに対するシミュレーション結果を表2に示す。表2は4.1節で述べた判断基準に基づく最適解の割合と処理時間の平均である。RBR はすべてのケースについて最適解を求められるが、処理に長時間を要する。標準的でない系統構成での事故 30 ケースについては、事例がないため、CBR では解を得ることはできない。そこで、CBR+RBR では CBR で解を得られない 30 ケースについては RBR で解を得る。INTERCARE はそれら 30 ケースのうち 29 ケースについて事例を用いて解を導き、29 ケース中 16 ケースに対して最適解を得た。最適解を得られなかったケースも復旧用電源が1個増える程度ではほぼ実用的な解を得ている。処理時間を比較すると、INTERCARE は平均で RBR の 8.1%、CBR+RBR の 12.4% と大幅に短縮していることが分かる。INTERCARE は最適解の割合が 72% とやや劣るが、すべてはほぼ実用的な解を導いており、事故時復旧の緊急性を考慮すれば、最適でなくとも実用的な解を迅速に得る点で有効である。もちろん、CBR+RBR と INTERCARE の処理時間は事例の獲得状態によって異なる。今回の実験では系統構成が異なるものを含むため、与えられた問題の半数以上について類似事例が存在しないので、事例が比較的獲得されていない状態と考えてよい。事例が豊富になると、CBR+RBR と INTERCARE の処理時間の差は小さくなると考えられる。

表3は今回の実験のうち標準的な系統構成における 20 ケースについての結果である。標準的な系統構成での事故では負荷レベルに応じて4個の事例が存在するので、相対的には事例が豊富な状態といえる。この

表2 推論方式の比較

Table 2 Comparison of reasoning methods.

推論法	最適解の割合 (%)	処理時間の平均 (s)
RBR	100.0	119.2
CBR	38.0*	5.7
CBR+RBR	98.0	78.5
INTERCARE	72.0	9.7

* 標準的な系統構成以外では解を得ることができない。

表3 標準的な系統構成に対する実験結果

Table 3 Experimental results for standard system configurations.

推論法	最適解の割合 (%)	処理時間の平均 (s)
RBR	100.0	116.0
CBR+RBR	95.0	5.4
INTERCARE	95.0	5.4

場合、CBR+RBRとINTERCAREはCBRと同じ性能を示す。これらの場合の最適解の割合が95%にとどまっているのは、系統条件が2Rと3Rの境界に近い1ケースにおいて、2Rで復旧可能な場合に3Rの解を導いたためであるが、この解は十分実用的である。また、より細かい条件に関して事例を豊富に獲得しておけば、最適解の割合を100%とすることも可能である。

以上のことから、INTERCAREはRBRやCBRとRBRの弱い結合形態に比較して、全体として効率的であることが明かである。

(3) 現実的な問題に対する考察

現実的な問題における(a)環境条件の変化や(b)最適解の判断基準の多様性に関して考察を加える。

(a) 環境条件の変化

CBRとINTERCAREは、事例を豊富に獲得すれば同じ性能になるが、現実的には、電力設備の増設や消費電力の増加にともなう環境条件の変化により、過去に有効であった事例が現在の条件には適さなくなってしまう場合がある。したがって、一度事例が豊富になったとしても、環境の変化によって過去の事例が使えなくなると、事例があまり豊富でない状態に戻ってしまう。すなわち、事例は豊富になる一方ではなく、環境変化によって、事例が豊富な状態と豊富でない状態が繰り返されると考えるのが妥当である。そのような状況を仮定すると、INTERCAREの問題解決処理はCBRとRBRの弱い結合形態に比較して有効である。

(b) 最適解の判断基準の多様性

今回の実験では復旧用の電源の個数が最少である復旧案を最適としたために、これらの基準を反映させたRBRも構築しやすく、RBRが導く最適解の割合を100%とすることができた。しかし、現実的には復旧案の最適性の判断基準はさまざまである。たとえば、復旧後の各部分系統での負荷の増加に対応したり、続いて生じるかもしれない事故に対応するために、各部分系統の送電可能電力にある程度余裕を持たせておく必要もある。また、最適性の判断基準が各部分系統に固有な系統条件に依存する可能性もある。このような判断基準を完全に反映させたRBRシステムを構築することは困難である。むしろ、それらの判断基準を反映して専門家が作成した復旧案を事例として登録しておけば、CBRで導いた解の方が最適な復旧案に近いこともあり得る。したがって、現実的問題ではRBRのみのシステムでは不完全であり、RBRとCBRの相互補完を行うシステムが有効であるといえる。

5. 関連研究

これまでも推論の効率化や高精度化を目的としてRBRとCBRの統合化が試みられている¹⁰⁾。効率化の面では、一般的にRBRではルールの適用が多段にわたるうえに、一般にルールの照合は処理時間がかかるために、効率が悪くなる傾向があるが、ルールを適用した結果である事例を直接用いることにより、多段にわたるルール適用の長いプロセスを節約できる。そのような場合、CBRで解を求めるのをデフォルトとし、CBRで解を得られない場合にRBRやモデルベース推論を行うと有効である^{7),11)}。一方、RBRにおいてルールの適用が1度ですむような場合は、CBRよりもRBRの方が効率的であり、事例からルールを導いておくことが有効になる¹²⁾。また、事例を用いてルールを洗練化する研究^{13),14)}も多く行われている。

高精度化を目的とした統合モデルでは、ルールと事例が独立して獲得されることが多い。それによって、ルールの不足を事例で補完したり、事例の不足をルールで補完することによって、片方の推論手法では解を得られないケースに対応することが考えられる。また、一般的な知識をルールで表し、例外的な知識を事例で表すことによって、解の精度を高めることもできる。Risslandら¹⁵⁾は、法律の解釈の分野で、スタンドアローンで実行可能なRBR部とCBR部を持つシステム(CABARET)を開発し、ヒューリスティクスを用いて、RBRとCBRの制御を行った。Goldingら¹⁰⁾は、まず、RBRを問題に適用して近似解を導き、それが例外に類似していると判定されるときCBRによって解を得るようなRBRとCBRの統合モデルを提案し、姓の発音の予測に応用した。彼らの実験では、RBRとCBRをそれぞれ単体で用いる場合に比較して、統合化システムの方が高い割合で正解を導くことが示されている。また、安信ら¹⁶⁾はIf-Thenルール(Rルール)とIf-Search-Thenルール(Cルール)を持つ統合モデルを提案した。これらのルールは同一の知識ベースに格納されており、Cルールが選択された場合には、Search部の検索条件に基づいて事例ベースから事例を検索し、Then部の加工手続きによって事例を加工して適用する。このモデルは帳票記入支援に応用され、意志決定の高度化を達成した。Chiら³⁾は、ルールと事例の両方が豊富でない分野を対象として、まず、RBRとCBRを順次試み、それでも解が得られない場合には、ルールで事例を一般化して問題解決に利用する方法を提案し、人事評価システムに応用した。

これらの統合化モデルでは、RBR と CBR では別の推論エンジンをういたり、別の知識ベースに格納したりしている。それに対して、本推論方式は1つの知識ベースにルールと事例を同じ形式で格納し、1つの推論エンジンがそれらの知識を用いて推論を行う。

6. おわりに

ルールと事例を同一の枠組みで扱う RBR と CBR の統合化の一手法を提案した。計算機によるシミュレーションの結果、本推論方式は、ルール不足を事例で補完したり、CBR 主体のシステムにおいても事例の不足をルールで補完するなど、柔軟な推論を行うことが示された。また、CBR の適用に失敗した場合に RBR を用いるという弱い結合形態に対して、全体として処理時間を大幅に短縮した。本論文で対象とした事故時復旧支援のように緊急性があり、最適でなくとも実用になる解を速く求めたい分野では有効であるといえる。本手法ではルールと事例をフレームを用いて同一形式で表現するので、事例のみを扱う場合に比較していくつかのファセットが余分に必要になり、事例の記述がやや複雑になる。そのため、事例獲得時の負荷の増加が懸念されるが、それらのファセットにデフォルト値を与えておけば、大きな増加はないと考えられる。

以上で述べたように、対象とした事故時復旧問題に対して良好な結果を得たが、ルールと事例の利用法についてはさらに検討を要する。たとえば、今回実現した実験システムでは、ルールか事例かにかかわらず、類似度の大きい知識が選択されるが、ルールと事例の優先関係が明らかな部分問題についてはそれを指定できるようにする必要がある。また、部分問題によっては事例を用いることが不適切な場合もある。たとえば、今回の例題では、停電範囲などの部分問題は過去の事例によることなく、現在の系統構成情報を用いてルールで導いた方がよい。そのようなきめ細かい設定が必要となろう。また、文献 4) のシミュレーションで示すように、本推論方式では副問題ごとに異なる事例を用い、全体として複数事例を用いることができるが、副問題間の干渉への対処やある副問題に対して複数の事例を利用することは今後の課題である。これらについて検討を続けたい。

謝辞 本研究を進めるにあたり、貴重なご意見をいただきました帝京大学藤原祥三教授、助手荒井正之博士に感謝いたします。

参考文献

1) 小林重信：事例ベース推論の現状と展望，人工

- 知能学会誌，Vol.7, No.4, pp.559-565 (1992).
 2) 渡辺博芳，奥田健三，東野長生，渡辺保之：事例ベース推論によるプレス加工工程設計支援，電子情報通信学会論文誌，Vol.J78-D-II, No.2, pp.340-348 (1995).
 3) Chi, R.T.H. and Kiang, M.Y.: Reasoning by Coordination: An Integration of Case-Based and Rule-Based Reasoning Systems, Knowledge-Based Systems, Vol.6, No. 2, pp.103-113 (1993).
 4) 渡辺博芳，奥田健三：ルールベース推論と事例ベース推論の統合化の一手法，情報処理学会研究報告 94-AI-97, pp.11-20 (1994).
 5) 奥田健三，渡辺博芳，王 方揚，山崎勝弘，馬場敬信：二次系統における事故時復旧操作への知識工学の応用，電気学会論文誌 (B)，Vol.107, No.10, pp.509-516 (1987).
 6) 松原 仁：推論技術の観点からみた事例に基づく推論，人工知能学会誌，Vol.7, No.4, pp.567-575 (1992).
 7) 奥田健三，渡辺博芳，山崎勝弘，馬場敬信：事例ベース形推論による二次系統の事故時復旧支援方式，電気学会論文誌 (B)，Vol.108, No.12, pp.553-560 (1988).
 8) 奥田健三，山崎勝弘：特集「事例ベース推論」電力系統事故時復旧支援，人工知能学会誌，Vol.7, No.4, pp.582-591 (1992).
 9) 河野 毅，末田直道，田中利一：CBR システムの構築環境，人工知能学会誌，Vol.9, No.1, pp.91-99 (1994).
 10) Golding, A.R. and Rosenbloom, P.S.: Improving Rule-based Systems Through Case-based Reasoning, Proc. of AAAI-91, pp.22-27 (1991).
 11) 栗山 健，寺野隆雄：RBR と CBR を統合したヘルプシステムの実現と評価，情報処理学会研究報告，93-AI-87, pp.1-8 (1993).
 12) 朝倉敬喜，宮下敏昭：知識ベース・事例ベース統合環境を利用したクレーム処理エキスパートシステム，第7回人工知能学会全国大会論文集，pp.767-770 (1993).
 13) Kobayashi, S. and Nakamura, K.: Knowledge Compilation and Refinement for Fault Diagnosis, IEEE, Expert, Vol.6, No.5, pp.39-46 (1991).
 14) Nakatani, Y. and Israel, D.: Tuning Rules by Cases, Topics in Case-Based Reasoning, Lecture Notes in Artificial Intelligence, Vol.837 (Wess, S. et al. ed.), Springer-Verlag, pp.313-324 (1994).
 15) Rissland, E.L. and Skalak, D.B.: Combining Case-based and Rule-based Reasoning: A Heuristic Approach, Proc. of IJCAI-89, pp.524-530 (1989).
 16) 安信千津子，山田 弘，源田晋司，鎌田芳栄：ルールベース推論と事例ベース推論の統合化の

一方法, 人工知能学会誌, Vol.7, No.6, pp.1087-1095 (1992).

(平成7年8月8日受付)

(平成8年1月10日採録)



渡辺 博芳 (正会員)

1963年生. 1986年宇都宮大学工学部情報工学科卒業. 1988年同大学大学院修士課程修了. 栃木県工業技術センターを経て, 1991年より帝京大学理工学部助手. 知識工学に関する研究に従事. 電子情報通信学会, 人工知能学会, AAAI 各会員.



奥田 健三 (正会員)

1927年生. 1952年早稲田大学第一理工学部電気工学科卒業. 工学博士. (株)日立製作所, 宇都宮大学工学部情報工学科教授を経て, 1993年より作新学院大学経営学部教授. 主たる研究テーマは知識工学, 事例ベース推論, ニューロコンピューティング, ファジィ理論の応用, 電力系統工学. 電気学会, 電子情報通信学会, IEEE 各会員.