

## 3C-7

# モデルベース故障診断システムにおける 制約伝播の効率的実現

土屋 隆 司・荻野 隆 彦

鉄道総合技術研究所

## 1. はじめに

モデルベースの診断システムでは、診断対象の構造と機能のモデル化が前提となるが（文献〔1〕参照）、一旦モデル化が行われると当該モデル上の動作（計算結果）と診断対象の動作（観察結果）の間の矛盾を発見することが第一義的な問題となる。診断対象とそのモデルの間の矛盾を発見するためには、非明示的な事実を明示的に表現するための計算プロセスとしての制約伝播が本質的である。本論文では、組合せ回路ないしは組合せ回路としてモデル化できる物理対象や概念対象への適用を念頭においた、モデルベースの診断システムについて、制約伝播の効率的な実現法について述べる。

## 2. 制約の表現と伝播の対象

制約伝播系を実現する際のポイントは、個々の制約の表現法、それらの制約を介して伝播する情報の選択、及び伝播の制御の方法である。ここでは組合せ回路への適用を考慮し、文献〔2〕にならない、以下のような定式化を行った。

## 1) 制約ネットワーク

制約ネットワークは（制約ノードの集合、観測点ノードの集合、エッジの集合）の3項組によって定義される。制約ノードとは、系の構成要素の機能の表現である制約を代表するものであり、観測点ノードとは、系における観測点の代表である。エッジは各制約ノードと観測点ノードを結び、有向であるとする。エッジの向きは、入力から出力の方向とする。図1に制約ネットワークの例を示す。

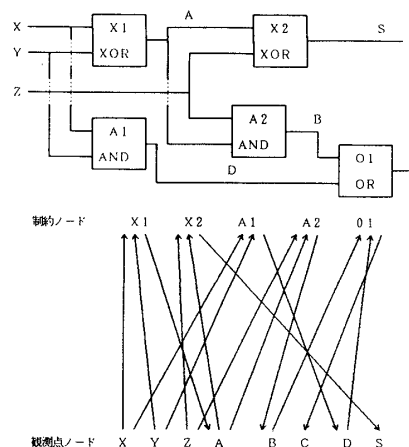


図1

## 2) 制約ネットワークと属性

まず制約ノードに対しては「伝播手続き」と呼ばれる属性を定義する。これは、ある制約ノードに隣接する観測点ノード群の値（以下に述べる「計算値」）をアドホックな方法で局所的に更新する手続きである。たとえば、入力に対応する観測点ノードの値から、出力に対応する観測点ノードの値を求めてもよいし、あるいは、可能な場合には、逆に出力から入力をもとめてもよい。

次に、観測点ノードに対しては、「観測値」と「計算値」と言う二つの属性を付与する。「観測値」は、診断対象に属する点で当該観測点ノードに対応したものの観測値（測定値）である。一方、「計算値」は、（伝播された値、依存する制約の集合）の二項組で表される。依存する制約の集合（以下、依存集合）とは、当該ノードに伝播された値に直接、間接を問わず寄与していると考えられる制約ノードの集合である。

伝播された値の依拠している制約の集合を計算値と共に伝播することは、TMSのような依存関係に基づくバックトラッキングを行うシステムでは必須の機能であるが、ここでは、依存関係の伝播の制御自体を、制約ノードに付与されたアドホックな手続きを用いて行うことにした。（理由は後述）

## 3) 矛盾集合と候補集合

診断対象となる系の要素の故障を検出するには、系の内部で発生した矛盾を適切な形で表現することが必要である。要素の少なくとも一つが故障している（モデルと異なる動作をしている）ことが演繹的に導かれるような制約の集合を矛盾集合と言う。候補集合は、その集合の要素すべての故障を仮定することにより、系の不正動作が説明できるもののことである。（これらの定義はおもに文献〔3〕による）

## 4) 大域的制約伝播

すでに、個々の制約に対する局所的伝播手続きを規定したが、次に系全体の伝播を制御する機構について述べる。これはWaltzのアルゴリズム（文献〔4〕）を利用した。これは区間ラベルの伝播（文献〔2〕）でも用いられている汎用的な制約伝播の手法である。図2に、アルゴリズムの概要を示す。

```

procedure Waltz;
  var cstack:stack of constraints;
      {系を構成する制約の集合}
      affected_constraints
      :set of constraints;
begin
  while (cstack is not empty)
    {cstackが空になるまで以下の繰り返し}
  begin
    c:=pop_from_cstack;
    {cstackから制約ノードをポップする。}
    affected_constraints:=propagate(c);
    {制約c固有の伝播手続きを用いて、制約伝
    播を行い、その結果影響を受ける可能性の
    ある他の制約ノードの集合を求める}
    push_all_constraints
    (affected_constraints);
    {上で求めた制約群をcstackにプッシュする}
  end
end;

```

図 2

### 3. 組合せ回路への適用

例えば図3のような回路を考えてみる。( )内に示すのは観測値、[ ]内に示すのは計算値である。この回路について、入力から出力の向きで制約伝播を行った結果、ノードJに、計算値[1, {C1, C2, C3, C4, C5}]が付与されたとする。これは観測値と矛盾しているので、故障の存在を示唆する兆候である。しかし、{C1, C2, C3, C4, C5}は矛盾集合としては大きすぎる。ORの出力が(モデルによって示される)本来取るべき値1ではなく、0を示しているのは、ORの入力の内、本来は1であるはずものがすべて0を示している結果と解釈できる。C5の入力Iには制約伝播の結果、計算値[0, {C2, C3, C4}]が付与されているので、このノードは、ノードJの矛盾には寄与していない。したがって、矛盾集合としては{C1, C5}が適当である。図4に適切な制約伝播の結果を示す。このように関連する制約(素子)の特性に応じて矛盾集合を計算方法を変えるには、個々の制約に付与された「伝播手続き」の中で、依存集合の伝播の方法を規定する方法を採用するのが便利である。(これは、Waltzのアルゴリズム自体が局所的な計算値の更新の方法、すなわち制約に付与された伝播手続きからは独立しているという事実を利用している)。これにより、制約に対する付加手続きの変更だけで、無駄な伝播の抑制、故障候補箇所の適切な絞り込みが可能になる。また経験的知識に基づく探索範囲の限定など、各推論ステップへのユーザーの介入を可能にした対話型ユーザーインタフェースの構築にも向いている。さらに、個々の制約に付与された伝播手続きは、系全体の制約伝播を制御する機構とは独立であるので、制約(素子)の種類が増えた場合の拡張も容易である。

### 4. まとめ

組合せ回路へ適用可能なモデルベースシステムの効率的な実現法について述べた。現実の組合せ回路(物理的対象)への適用ばかりではなく、専門家の経験的知識(ノウハウ)をAND、OR等の論理演算子を用いて木構造で表現したもの(概念的対象)にも、まったく同じ枠組みが適用可能である。今後は適用範囲を順序回路にまで拡張していくことを検討したい。

### 5. 参考文献

- [1] Randall Davis, Diagnostic Reasoning Based on Structure and Behaviour, Artificial Intelligence 24(1984)
- [2] Earnest Davis, Constraint Propagation with Interval Labels, Artificial Intelligence 32(1987)
- [3] John de Kleer, Brian C. Williams, Diagnosing Multiple Faults, Artificial Intelligence 32(1987)
- [4] Waltz, D., Understanding line drawings of scenes with shadows, in P. Winston(Ed.) Psychology of Computer Vision(McGraw-Hill, New York, 1975)

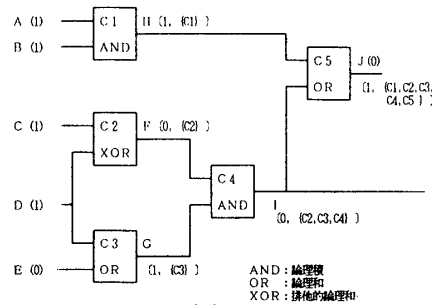


図 3

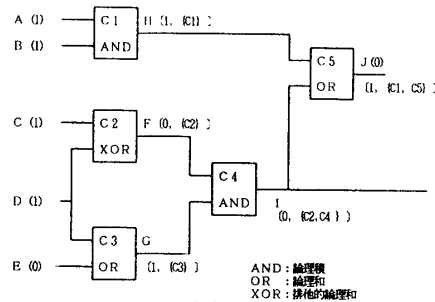


図 4