

論 文

実時間探索を導入したコストに基づく仮説推論システムにおける ヒューリスティック関数の改良

越野 亮[†] 岡峰 正[†] 木村 春彦[†] 広瀬 貞樹^{††}

Improving Heuristic Function of Cost-Based Abduction System Using Real-Time Heuristic Search

Makoto KOSHINO[†], Tadashi OKAMINE[†], Haruhiko KIMURA[†], and Sadaki HIROSE^{††}

あらまし コストに基づく仮説推論は、背景知識に適当な仮説を付加することで与えられた観測事象の最も好ましい説明を求める推論であり、診断や設計、プランニング、自然言語理解などに応用され、重要な推論の枠組みとなっている。しかしながら、仮説推論の計算量が極めて大きいため、探索空間を絞り込む工夫や仮説間の無矛盾性を調べる矛盾処理の高速化が重要な課題となっている。A*探索を導入した従来の仮説推論システムでは、効率的な推論が実現されているが、展開したノードをすべて記憶しておく必要があるため、膨大なメモリ量が必要となるという問題がある。本論文では、先読み深さを設定し、その中で最も良いノードのみを残し、他のノードを削除する実時間探索を導入するが、特に、推論途中にほかのノードを削除する場合、最終的に残ったノードがすべて矛盾となり、推論の失敗に陥る可能性が高いという問題があるため、このような失敗に陥らないように推論を促し、かつ矛盾処理の高速化を図るヒューリスティック関数を導入した。また、実験により本提案手法の有効性を検証した。

キーワード コストに基づく仮説推論、ヒューリスティック関数、A*探索、実時間探索、矛盾処理

1. ま え が き

コストに基づく仮説推論 (Cost-based Abduction) [1], [2] は、背景知識に適当なコスト (確率や信頼度などから求める) が付加された仮説を選択し追加することで、与えられた観測の最良の説明を求める推論の枠組みであり、診断 [3] や設計 [4], プランニング [5], 自然言語理解 [6] などに応用されている。しかし、最適解を求める計算量は NP 困難 [7] であるため、効率的な最適解探索は困難である。また、矛盾知識を含む仮説推論の計算量は NP 完全 [8] であるため、矛盾処理に膨大な計算時間がかかるという問題があり、実用化に向けて高速化が重要な課題となっている [9]。

文献 [10] では、A*探索 [11] をコストに基づく仮説推論に応用することにより効率的な推論を実現してい

る。しかしながら、A*探索は展開したノードをすべて記憶しておく必要があるため、膨大なメモリ量が必要となるという問題がある。

この問題を解消するために、ノードをすべて保存することをやめ、一定の深さの先読みを行い、その先読み結果で最も良いノードだけを残し、他のノードは削除する実時間探索を導入した仮説推論システムが考えられるが、従来、他のノードを削除する方法は、矛盾制約が厳しい仮説推論においては、残された仮説がすべて矛盾となり推論の失敗に陥る可能性が高いため導入は難しい [12] と考えられていた。また、展開した膨大な数のノードすべてに対して矛盾が生じないかを調べる矛盾処理を行う必要があるため、膨大な計算時間がかかるという問題が依然として残っている。そこで、本論文では、実時間探索を導入した仮説推論において、推論の失敗に陥らないように推論を促し、かつ矛盾処理の高速化を図るヒューリスティック関数に改良する。

以下に、2. で仮説推論を概説し、A*探索を導入した仮説推論システム [10] を紹介する。3. で実時間探索による仮説推論の方法と問題点を解説し、問題点を

[†] 金沢大学大学院自然科学研究科, 金沢市
Graduate School of Natural Science, Kanazawa University,
Kanazawa-shi, 920-8667 Japan

^{††} 富山大学工学部知能情報工学科, 富山市
Faculty of Engineering, Toyama University, Toyama-shi,
930-8555 Japan

解消するシステムを提案する。続いて 4. で提案システムの評価実験を行い、5. で本論文をまとめる。

2. 仮説推論

本章では、Poole らによる仮説推論 [13] の定義を述べ、本研究が対象とする A*探索による仮説推論システムについて簡単に説明する。

2.1 コストに基づく仮説推論

コストに基づく仮説推論は、仮説の選択の基準として知識ベースに含まれる各仮説に重み（コスト）が与えられており、与えられた観測に対して最適な説明を求める仮説推論である。観測の説明のコストは観測の説明に現れる仮説の重みの和で与えられるものとし、説明が最適であるとは説明のコストが最小であることとする。

[定義 1] F をホーン節集合（「背景知識」と呼ぶ）、 H をアトム集合（「仮説知識」と呼ぶ）とする。また、 g を存在束縛されたアトムの連言（「観測」、または単に「ゴール」と呼ぶ）とする。このとき、 g の最適な説明とは、 F と合わせて g を証明できるような H の無矛盾なコスト最小の部分集合 E を求めることである。つまり、以下の条件を満たす解仮説 E を求めることである。

- $E \subseteq H$ (E は H の部分集合である)
- $F \cup E \vdash g$ ($F \cup E$ から g が証明される)
- $F \cup E \not\vdash \perp$ ($F \cup E$ は無矛盾である)
- E のコストは最小

[定義 2] 仮説間の矛盾を次のような制約知識で表記し、この集合を IC （「矛盾知識」と呼ぶ）とする。

$$inc := A_1, A_2, \dots, A_n.$$

ただし、 A_k ($1 \leq k \leq n; 2 \leq n$) はアトムであり、 inc は *inconsistent* の略で矛盾を表す。

仮説推論の知識ベースは $\langle F, H, IC \rangle$ の 3 字組で定義され、以下、これらを KB と表記する。

[例 1] (Prolog による仮説推論)

図 1 に知識ベース $KB_1 = \langle F, H, IC \rangle$ を示す。 KB_1 に対して、ゴール g が $p(X, Y)$ として与えられたときの Prolog による推論 (SLD 反駁 [14]) の様子を図 2 に示す。ここで用いている演算子 M は仮説を立てたことを示すマークである。また、アトムの選択規則は最左優先とする。

ここで inc をルートノードとする推論木の葉に現れる仮説集合 $\{e(1), i(1)\}$, $\{e(2), i(2)\}$ は矛盾を引き起

F	
$p(X, Y) :- a(X), c(Y).$	
$p(X, Y) :- b(X), d(Y).$	
$a(X) :- e(X), i(X).$	

H	Cost
$c(1).$	3
$c(2).$	2
$b(1).$	5
$b(2).$	4
$d(1).$	6
$d(2).$	4
$e(1).$	1
$e(2).$	4
$i(1).$	1
$i(2).$	3

IC
$inc :- e(X), i(X).$

図 1 知識ベース KB_1
Fig. 1 Knowledge-base KB_1 .

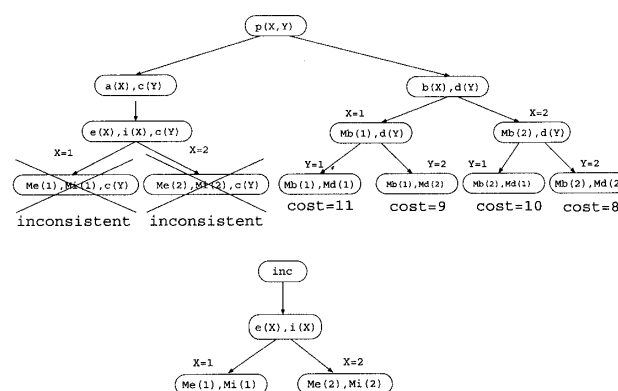


図 2 Prolog による推論
Fig. 2 Inference by Prolog.

こすことから、 $p(X, Y)$ の推論木におけるこれらの仮説集合は取り除く。この矛盾を引き起こす仮説集合を取り除く処理を矛盾処理と呼ぶ。また、 $p(X, Y)$ の推論木からコスト最小で無矛盾な仮説集合 $\{b(2), d(2)\}$ が最適解として与えられる。

2.2 A*探索による仮説推論システム

本節では、本研究が対象とする A*探索を導入した仮説推論システム [10] について述べる。以下、この A*探索による仮説推論システムを A*システムと呼ぶことにする。

(1) A*探索

図 2 のように、一つの最適解を得るために全解を求めるのは、無駄な探索空間が大きく極めて非効率であり、大きい問題に対しては計算量の組合せ的爆発により現実的には実行不可能である。そこで、A*探索 [11]

論文/実時間探索を導入したコストに基づく仮説推論システムにおけるヒューリスティック関数の改良

を仮説推論に導入し、それぞれのノードに評価値を与え、評価値の低いノードを優先して展開することで、探索効率の向上を図る。

[定義 3] 最適解探索におけるノード n の評価値 $f(n)$ を以下のように定義する。

$$f(n) = g(n) + \hat{h}(n)$$

ただし、 $g(n)$ はゴールからノード n までに用いられた仮説のコストの和で、 $\hat{h}(n)$ は実際の最適解のコストより小さくならないようなノード n から葉までの最小の推定値である。

(2) 推定値 $\hat{h}(n)$ の導出方法

AI の分野における探索アルゴリズムでは、与えられた問題に対して実行可能なヒューリスティックは抽象化された問題から生成することができ、更に、このようにして得られたヒューリスティックが効果的なものとなるには、抽象化された問題が、それを効率的に解くことができる範囲でもとの問題の近似である必要がある [15] ことが知られている。文献 [10] では、述語論理式で表現された知識ベースに対して、その近似として述語の引き数を無視して抽象化した命題論理の知識ベースを対象として推定値 $\hat{h}(n)$ を求めている。

[定義 4] 論理式 (あるいはその集合) S に対し、これを命題論理に抽象化したものを \bar{S} と表記する。

[定義 5] 集合 D に含まれる仮説に付加されたコストの和を求める関数を $cost(D)$ のように表記する。

[定義 6] KB を述語論理で表現された知識ベース、 KB を述語の引き数を無視して命題論理に抽象化したものを \overline{KB} とする。 KB に含まれる仮説 E に対して、 \bar{E} のコストは以下の式で与えられる。

$$cost(\bar{E}) = \min\{cost(L) | L \text{ は } KB \text{ に含まれる仮説かつ } \bar{L} = \bar{E}\}$$

ここでは、知識ベース KB に含まれる矛盾知識 IC を除いた知識ベース $KB \setminus IC$ を考える。

[例 2] (抽象化した知識ベース)

図 1 の知識ベース KB_1 を命題論理に抽象化した知識ベース $\overline{KB_1 \setminus IC}$ を図 3 に示す。ここで、例えば、仮説 c のコストは KB_1 に含まれる仮説 $c(1), c(2)$ のコストの最小値 2 で与えている。

[定義 7] KB を知識ベース、 g を与えられた観測、 A を KB に含まれるアトムとし、 H_A を A の最適な説明を表す仮説集合とする。ノード n からある成功葉

\bar{F}	
$p :- a, c.$	
$p :- b, d.$	
$a :- e, i.$	

\bar{H}	Cost
c.	2
b.	4
d.	4
e.	1
i.	1

図 3 命題論理に抽象化した知識ベース $\overline{KB_1 \setminus IC}$
Fig. 3 Abstracted knowledge-base into propositional logic $\overline{KB_1 \setminus IC}$.

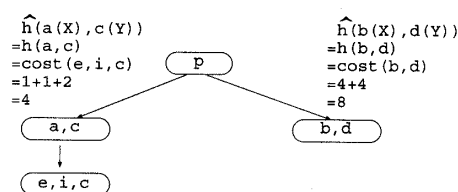


図 4 $\overline{KB_1 \setminus IC}$ による推定値の計算
Fig. 4 Calculation of estimated value by $\overline{KB_1 \setminus IC}$.

に至る最適解のコストの予測値 $\hat{h}(n)$ は以下の式で定義される。

$$\hat{h}(n) = h(\bar{n}) = cost(H_{\bar{n}})$$

$h(\bar{n})$ は \overline{KB} における \bar{n} に対する最適解のコストを示す。

[例 3] (推定値の計算)

知識ベース KB_1 (図 1) とゴール $p(X, Y)$ が与えられ、 $p(X, Y)$ において背景知識を適用し、 $a(X), c(Y)$ と $b(X), d(Y)$ が展開されたとする。このときノード $a(X), c(Y)$ の推定値 $\hat{h}(a(X), c(Y))$ は、抽象化した知識ベースにおけるノード a, c の最適な説明 $H_{a,c} = \{e, i, c\}$ のコストの和である 4 と与えられる。同様に、ノード $b(X), d(Y)$ の推定値 $\hat{h}(b(X), d(Y))$ は、8 と与えられる (図 4)。

[例 4] (A*システムによる推論)

A*システムによる推論の様子を図 5 に示す。評価値 f の計算において、 g は実際に適用された抽象化以前の仮説のコストの和であり、 \hat{h} は定義 7 に基づいて計算された値である。

図 2 と比べて枝の数及び M 演算子付きアトムの数が少ないことから、仮説推論の過程において真と仮定される仮説の数が制限され、「合理的な仮説の選択」が実現されていることがわかる。このように、推論に

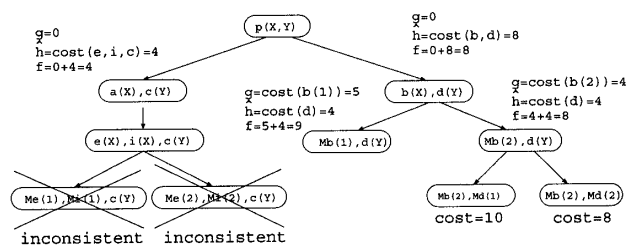


図5 A*システムによる推論
Fig. 5 Inference by A*system.

において生成されるノードの数を制限することができ、仮説推論の探索空間を絞り込むことが可能となる。

A*探索は、ヒューリスティックな評価関数を最も効率的に利用できる最適解探索法として広く知られているが、同時に展開したノードをすべて記憶しておく必要があるため、メモリ量が極めて大きいということが問題点となっている。また、コストに基づく仮説推論の最適解探索の計算複雑度はNP困難であるため、計算量は指数オーダーとなると考えられている。したがって、大規模知識ベースでは膨大な時間計算量と空間計算量のため実行不可能になることがある。

3. 提案システム

この章では、まずA*システムの問題を解消する実時間探索を導入した仮説推論システムCARTARSについて説明する。次に、CARTARSの問題点を述べ、その問題点を解消する新しいヒューリスティック関数を提案する。

3.1 実時間探索を導入した仮説推論システム CARTARS

実時間探索 (Real-time heuristic search) [16] とは、ノードをすべて保存することをやめ、一定の深さの先読みを行い、その先読み結果で最も良いノードだけを残し、他のノードは削除する手法である。先読み探索を制限してしまうことで、最適解を保証することができないが、近似解を多項式オーダーで求めることができる。この実時間探索を導入した仮説推論システムを以降、CARTARS (Cost-based Abduction using Real-Time A* Reasoning System) と呼ぶことにし、アルゴリズムを以下に示す。

Step1 (初期設定) 知識ベース KB とゴール g を与える。先読みの深さ l を設定する。 KB に含まれる矛盾知識を取り除き、述語の引数を無視した $\overline{KB \setminus IC}$ に抽象化する。矛盾知識 inc の最小の仮説集合を求める。現在の探索ノード x を g とし、

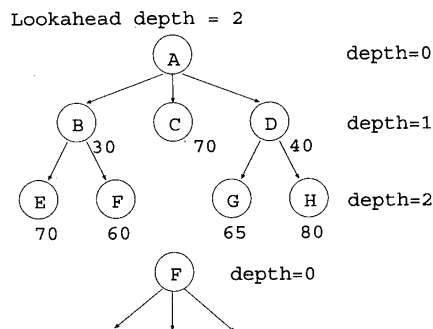


図6 CARTARSの推論
Fig. 6 Inference of CARTARS.

SG をノードのリストとする。

Step2 (展開) x と KB における1段階のSLD導出で得られるサブゴールをすべて求めて SG に追加する。

Step3 (終了判定) $SG = \phi$ (サブゴールがないとき) x の仮説集合を解として出力し、終了する。

Step4 (矛盾処理) SG に現れる仮説が矛盾知識の仮説集合を含むかどうかを調べ、矛盾するサブゴールを SG から削除する。

Step5 (先読み探索) 最小の評価値 $\min\{f(x) = g(x) + \hat{h}(x) | x \in SG\}$ をもつノードを x とする。

Step6 (先読み深さチェック) もし、ノード x の深さが先読み深さ l なら SG を削除し Step2 へ。それ以外なら、 SG を保持し Step2 へ。

[例5] (CARTARSによる推論の様子)

簡単な例として図6を用いてCARTARSの推論の様子を説明する。先読みの深さを2とする。ゴール A が与えられたとして、展開されたサブゴールが $SG = \{B, C, D\}$ である。その中で評価値が最小のサブゴール B を展開する。 B で展開されたサブゴールは $\{E, F\}$ なので、 $SG = \{E, F, C, D\}$ となる。同様に推論を進めていくと $SG = \{E, F, C, G, H\}$ となる。評価値が最小のサブゴール F は、深さ2であるので、他のサブゴールをすべて削除し、現在のサブゴール F を深さ0のゴールと考えて推論を行う。

A*システムの探索空間のオーダーは、分枝数を b 、探索の深さを d とすると $O(b^d)$ である。CARTARSの探索空間のオーダーは、分枝数を b 、探索の深さを d 、先読みの深さを l とすると、 $b^l \cdot \frac{d}{l} \rightarrow O(d \cdot b^l)$ である。両手法の探索空間の概略を図7に示す。

A*システムでは、知識ベースの規模を大きくすると、探索の深さ d 、分枝数 b が増加し、探索空間は指数関数的に大きくなるため、時間計算量も空間計算量

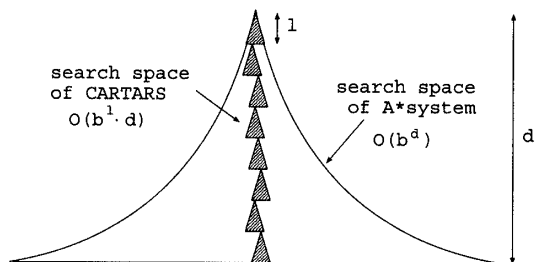


図7 A*システムと CARTARS の探索空間の概念
Fig. 7 Concept of search space on A*system and CARTARS.

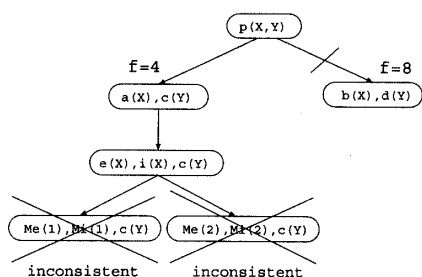


図8 CARTARS で推論の失敗に陥る例
Fig. 8 Example of failing inference by CARTARS.

も指数オーダになる。CARTARS では、探索空間のオーダ $O(d \cdot b^l)$ の l をユーザが設定することができる。これにより、時間計算量も空間計算量もユーザが与えるパラメータ l の多項式オーダで近似解を求めることができる。しかしながら、CARTARS では、残された仮説がすべて矛盾となり、推論の失敗に陥る可能性が高いという問題がある。

[例 6] (CARTARS で推論の失敗に陥る例)

図 1 の知識ベース KB_1 において、先読み深さ 1 で実時間探索を行ったとすると、残ったノードがすべて矛盾となる (図 8)。

また、展開されたノードすべてで、矛盾を含む仮説集合を取り除くという矛盾処理を行う必要があるため膨大な計算時間がかかるという問題もある。そこで、実時間探索を導入した CARTARS において、推論の失敗に陥らないように推論を促し、かつ矛盾処理を高速化するようにヒューリスティック関数を改良する。

3.2 ヒューリスティック関数の改良による成功率向上と矛盾処理の高速化

残された仮説が矛盾にならないようにするために、矛盾になる可能性のあるノードは展開しない方法をとることにする。これにより、展開したノードでは矛盾になる可能性はないため矛盾処理を省くことができる。また、初期設定で行う矛盾知識の最小の集合を求める

処理も省くことができる。

[定義 8] 矛盾知識 IC を命題論理に抽象化した知識ベースを \overline{IC} とする。A*探索の推定値の計算において、ノード n の最適な説明 H_n に、抽象化した矛盾知識 \overline{IC} に含まれる仮説集合がある場合は推定値 h を ∞ とする。

提案するヒューリスティック関数では、矛盾になる可能性のあるノードを展開しないようにするために、命題論理に抽象化した知識ベース上で矛盾が含まれるかどうかの矛盾処理を行っている。この改良したヒューリスティック関数による CARTARS を New-CARTARS と呼ぶことにし、アルゴリズムを以下に示す。

Step1 (初期設定) 知識ベース KB とゴール g を与える。先読みの深さ l を設定する。 KB を述語の引数を無視した \overline{KB} に抽象化する。現在の探索ノード x を g とし、 SG をノードのリストとする。

Step2 (展開) x と KB における 1 段階の SLD 導出で得られるサブゴールをすべて求めて SG に追加する。

Step3 (終了判定) $SG = \phi$ (サブゴールがないとき) x の仮説集合を解として出力し、終了する。

Step4 (先読み探索) 最小の評価値 $\min\{f(x) = g(x) + \hat{h}(x) | x \in SG\}$ をもつノードを x とする。

Step5 (矛盾処理) ノード x の最適な説明に抽象化した矛盾知識 \overline{IC} が含まれる場合は $h = \infty$ とし、Step4 へ。

Step6 (先読み深さチェック) もし、ノード x の深さが先読み深さ l なら SG を削除し Step2 へ。それ以外なら、 SG を保持し Step2 へ。

[例 7] (New-CARTARS による推定値の計算)

図 1 の知識ベース KB_1 が与えられたとすると、命題論理に抽象化した知識ベース $\overline{KB_1}$ は図 9 のように表される。提案するヒューリスティック関数による推定値の計算の様子を図 10 に示す。例 3 の図 4 では、ノード a, c の最適な説明は e, i, c であり、推定値 $\hat{h}(a(X), c(Y))$ は 4 と与えられたが、New-CARTARS では、 e, i は矛盾を引き起こす可能性があることから、推定値 $\hat{h}(a(X), c(Y))$ を ∞ としている。

[例 8] (New-CARTARS の推論の様子)

図 1 の知識ベースが与えられたとすると、先読み深さ 1 の New-CARTARS の推論の様子は図 11 のようになる。これにより、図 8 で問題となった推論の失敗に陥ることを防ぎ、解を求めることができる。

\overline{F}
$p := a, c.$
$p := b, d.$
$a := e, i.$

\overline{H}	Costs
c.	2
b.	4
d.	4
e.	1
i.	1

\overline{IC}
$inc := e, i$

図9 命題論理に抽象化した知識ベース \overline{KB}_1
 Fig. 9 Abstracted knowledge-base into propositional logic \overline{KB}_1 .

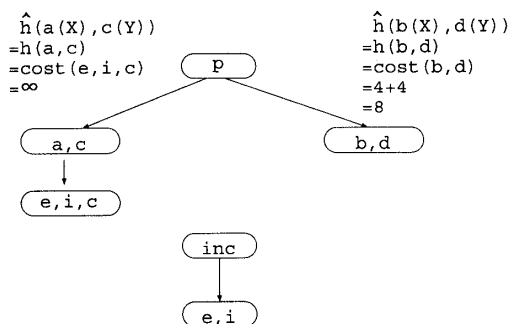


図10 New-CARTARSによる推定値の計算
 Fig. 10 Calculation of estimated value by New-CARTARS.

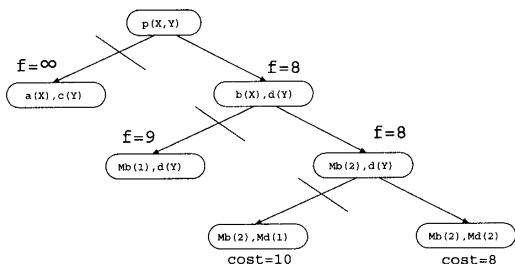


図11 New-CARTARSによる推論
 Fig. 11 Inference by New-CARTARS.

3.3 ヒューリスティック関数の改良による矛盾処理の高速化の効果と有効範囲

CARTARSにおける矛盾処理の計算量は、矛盾知識 IC の数を I_r 、探索空間の大きさを S_r とすると、 $I_r \cdot S_r$ で与えられる。New-CARTARSにおける矛盾処理の計算量は、抽象化された矛盾知識 \overline{IC} の数を I_a 、抽象化された探索空間の大きさを S_a とすると、 $I_a \cdot S_a$ で与えられる。したがって、本提案手法の有効範囲は、述語論理で与えられた知識ベースを命題論理に抽象化した知識ベースが小さい場合であると考え

表1 n -bit 全加算回路の故障診断を行う知識ベース
 Table 1 Knowledge-base of diagnostics of n -bit ripple carry adder circuit.

背景知識 F	回路の素子間の結線
仮説知識 H	各素子の状態（正常，開放，短絡）
矛盾知識 IC	一つの素子が二つ以上の状態をとること
観測事象 g	出力（観測状態）

表2 知識ベースの規模 n に対する CARTARS と New-CARTARS の推論時間の比較

Table 2 Comparison with inference time of CARTARS and New-CARTARS by increasing size n of knowledge-base.

Size n of KB	Inference time of CARTARS [sec]	Inference time of New-CARTARS [sec]
20	17	4
25	33	8
30	223	51
35	514	55
40	549	132

られる。つまり、与えられた問題の述語の引き数の取り得る値が多い場合であり、述語論理の引き数が少なく抽象化した知識ベースと同等の大きさをもつ問題では高速化の効果を得ることは難しいと考えられる。しかしながら、実際の述語論理の知識ベースでは、引き数が数百、数千もの数になることも稀ではない[17]ため、本提案手法が有効である問題の範囲は広いと考えられる。

4. 実験

提案システム New-CARTARS の有効性を示すために、 n -bit 全加算回路の故障診断を行う知識ベース（表1）を用いて CARTARS との比較実験を行った。また、各仮説のコストは各素子が上記の状態となる確率 P を考え、 $-\log_e P$ で与えた。このようにコストを与えると、確率の積 $\prod_i P_i$ が正の数の和 $\sum_i (-\log_e P_i)$ で表現でき、確率最大の説明を求める問題をコスト最小の説明を求める問題に帰着することができる。

4.1 実験 1

n をパラメータとして n -bit 全加算回路の知識ベースの規模を変化させて実験を行った。CARTARS, New-CARTARS とともに先読み深さは1として $n = 20 \sim 40$ で実験を行った結果を表2に示す。

実験結果より、約4倍の速度向上を実現していることがわかる。このことを3.3で述べた CARTARS と New-CARTARS の矛盾処理の計算量と表3を用いて考察する。本研究で用いた知識ベースでは、矛盾知識における述語の引き数は、素子の状態を表しているた

論文/実時間探索を導入したコストに基づく仮説推論システムにおけるヒューリスティック関数の改良

表3 4倍の速度向上についての考察

Table 3 Consideration about speed up of four times.

Size n of KB	Number of $Node$	Number of $Node$	Number of IC	Number of IC
20	732	504	291	97
25	922	634	366	122
30	1112	764	441	147
35	1302	892	516	172
40	1492	1024	591	197

め、引き数の取り得る値は三つである。したがって、抽象化した矛盾知識の数は1/3になる。また、抽象化した探索空間も実際の探索空間より約3/4に小さくできたため、約4倍の速度向上になったと考えられる。

従来のA*システムでは、 $n = 20$ 以上にするとメモリ不足のため実行が困難であり、実行時間についても20000秒以上必要としていたが、CARTARS, New-CARTARSとも実用的なメモリ量での推論を実現し、実行時間についても実用的な時間で実現している。また、CARTARSについては推論に失敗することが多かったが、New-CARTARSにおいてはすべての知識ベースで解を求めることに成功していた。これにより、New-CARTARSは、最適解を必ずしも必要とせず、近似解をできるだけ早く求める場合に有効だと考えられる。

4.2 実験 2

次に、CARTARSとNew-CARTARSにおいて、推論に成功する確率（成功率）と最適解が求められた確率（最適率）を調べるために、20種類の8bitの知識ベースにおいて実験を行った。8bitの知識ベースを対象とした理由は、大規模知識ベースでは、A*システムで実行ができないため、最適解を求めることができない場合があり、8bitより小さい知識ベースでは、すぐ解が求められてしまい、実行時間の比較ができないためである。CARTARSとNew-CARTARSにおいて先読み深さを変えたときの推論時間の平均、成功率、最適率をそれぞれ表4、表5に示す。

実験結果より、先読み深さが小さい場合（1~20）、CARTARSでは成功率が15~55%、New-CARTARSでは、最適率が30~45%とかなり低い。これは、回路の故障診断問題の性質として、先読みが小さい場合、推論に失敗しやすく、また、推論に失敗しないように推論を促すと最適解を逃しやすい問題であると考えられる。しかしながら、先読み深さをある程度大きくすると推論時間は大幅に大きくなるが、成功率、最適率とも大幅に改善されることがわかる。

表4 CARTARSにおける先読み深さに対する推論時間の平均、成功率、最適率

Table 4 Average of inference time, success rate and optimal rate versus lookahead depth on CARTARS.

Lookahead depth	Inference time [sec]	Success rate [%]	Optimal rate [%]
1	0.18	15	100
5	0.60	40	90
10	2.42	45	100
20	4.14	55	95
30	15.35	75	100
40	34.64	85	100
50	86.47	100	100

表5 New-CARTARSにおける先読み深さに対する推論時間の平均、成功率、最適率

Table 5 Average of inference time, success rate and optimal rate versus lookahead depth on New-CARTARS.

Lookahead depth	Inference time [sec]	Success rate [%]	Optimal rate [%]
1	0.45	100	30
5	0.80	100	30
10	0.95	100	35
20	2.25	100	45
30	6.45	100	75
40	24.10	100	100
50	24.30	100	100

また、先読み深さが小さい場合は、CARTARSより実行時間がかかっているが、これは、展開されたノード数が少ないため、ヒューリスティック関数のオーバーヘッドが省くことができる矛盾処理より大きかったためであると考えられる。しかし、そのオーバーヘッドは1秒以内であり、先読み深さが大きい場合や、実験1のように大規模知識ベースでは、提案システムの速度向上を実現しているため有効だと考えられる。

5. むすび

A*探索を導入した従来の仮説推論システムでは、効率的な推論が実現されているが、展開したノードをすべて記憶しておく必要があるため、膨大なメモリ量が必要となり、また、展開したノードすべてに矛盾処理を行う必要があるため、膨大な計算時間がかかるという問題があった。すべてのノードを記憶すると膨大なメモリ量が必要となるので、一定の深さの先読みを行い、その中で最も良いノードのみを残し、他のノードを削除する方法である実時間探索を導入する方法が考えられるが、仮説推論では、推論途中に他のノードを削除する場合、最終的に残ったノードがすべて矛盾

となり、推論の失敗に陥る可能性が高いという問題がある。

そこで、本論文では、命題論理に抽象化した矛盾知識を用いることで、矛盾が生じないノードのみを展開し、推論の失敗に陥らないように推論を促すヒューリスティック関数を提案した。また、矛盾が生じないノードのみを展開するため、これにより実際の推論時の矛盾処理も省くことができ、高速化を実現した。回路の故障診断の知識ベースに対して比較実験した結果、New-CARTARSは高い確率で推論に成功することができた。したがって、提案手法は効率の良い近似解を求める場合に有効であると考えられる。

謝辞 査読者の先生方から有益な御教示、御助言を頂いた。ここに記して謝意を表する。

文 献

- [1] E. Charniak and S.E. Shimony, "Probabilistic semantics for cost based abduction," Proc. AAAI-90, pp.106-111, 1990.
- [2] E. Charniak and S. Husain, "Cost-based abduction and MAP explanation," Artif. Intell., vol.66, pp.345-374, 1994.
- [3] R. Reiter, "A theory of diagnosis from first principles," Artif. Intell., vol.13, pp.57-95, 1987.
- [4] J.J. Finger, "Exploiting constraints in design synthesis," Technical Report STAN-CS-88-1024, Department of Computer Science, Stanford University, Stanford, CA, 1987.
- [5] K. Eshghi, "Abductive planning with event calculus," Proc. 5th ICLP, pp.562-579, 1988.
- [6] J. Hobbs, M. Stickel, D. Appelt, and P. Martin, "Interpretation as abductions," Artif. Intell., vol.63, pp.69-142, 1993.
- [7] B. Selman and H.J. Levesque, "Abductive and default reasoning: A computational core," Proc. AAAI-90, pp.343-348, 1990.
- [8] T. Bylander, D. Allemang, M.C. Tanner, and J.R. Josephson, "The computational complexity of abduction," Artif. Intell., vol.49, pp.25-60, 1991.
- [9] 石塚 満, "仮説推論の計算量と高速化メカニズム," 人工知能誌, vol.9, no.3, pp.342-349, 1994.
- [10] 加藤昇平, 世木博久, 伊藤英則, "コストに基づく仮説推論における最適解探索の一方法," 情処学論, vol.36, no.10, pp.2380-2390, 1995.
- [11] P.E. Hart, N.J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," IEEE Trans. Systems Science and Cybernetics, vol.SSC-4, no.2, pp.100-107, 1968.
- [12] 近藤朋子, 石塚 満, "述語論理知識を扱う仮説推論における最適解の高速推論法," 人工知能誌, vol.9, no.2, pp.110-118, 1994.
- [13] D. Poole, R. Aleliunas, and R. Goebel, "Theorist: A

logical reasoning system for defaults and diagnosis," The Knowledge Frontier: Essays in the Representation of Knowledge, Springer-Verlag, N.Y., 1987.

- [14] J.W. Lloyd, Foundations of Logic Programming, Springer, 1984.
- [15] A.E. Frieditis, "Machine discovery of effective admissible heuristics," Machine Learning, vol.12, pp.117-141, 1993.
- [16] R.E. Korf, "Real-time heuristic search," Artif. Intell., vol.42, pp.189-211, 1990.
- [17] 棚橋一也, 福田茂紀, 石塚 満, "命題レベルの高速解法の利用を図るコストに基づく述語論理版仮説推論法," 人工知能誌, vol.14, no.6, pp.1100-1107, 1999.

(平成13年3月1日受付, 8月14日再受付)



越野 亮 (学生員)

2000 金沢大・工・電気・情報卒。同年同大大学院自然科学研究科電子情報システム専攻博士前期課程入学, 現在に至る。その間, 仮説推論の高速化の研究に従事。情報処理学会, 人工知能学会各会員。



岡峰 正

2001 金沢大・工・電気・情報卒。同年同大大学院自然科学研究科電子情報システム専攻博士前期課程入学, 現在に至る。その間, 仮説推論の高速化の研究に従事。



木村 春彦 (正員)

1974 東京電機大・工・応用理化学卒。1979 東北大学院工学研究科情報工学専攻博士課程了。工博。同年富士通(株)勤務。1980 金沢女子短期大学講師。1984 金沢大学経済学部助教授を経て, 現在, 同大学院自然科学研究科教授。その間, 最適コード変換, プログラクションシステムの高速化の研究に従事。情報処理学会, 人工知能学会各会員。



広瀬 貞樹 (正員)

1974 富山大・工・電子卒。1980 東北大学院工学研究科情報工学専攻博士課程了。工博。同年, (株)富士通研究所入社。1984 神奈川大学工学部助教授。1989 富山大学工学部助教授, 1998 同学部知能情報工学科教授。その間, オートマトン, 形式言語理論, アルゴリズム解析, 計算の複雑さの理論などの研究に従事。情報処理学会会員。