

研究速報

改良山登り法によるコストに基づく仮説推論の高速最適解法

越野 亮[†]* (正員) 参沢 匡将[†] (学生員)
木村 春彦[†] (正員) 広瀬 貞樹^{††} (正員)

High Speed Optimization Method on Cost-Based Abduction
Using Improved Hill Climbing Algorithm

Makoto KOSHINO[†]*, *Regular Member*,
Tadanobu MISAWA[†], *Student Member*, Haruhiko KIMURA[†],
and Sadaki HIROSE^{††}, *Regular Members*

[†] 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

* 現在, 富士通株式会社

あらまし 本論文では, コストに基づく仮説推論における最適解法の手法として, 山登り法を改良した手法を提案する. また, 回路の故障診断問題に適用し, 従来の高速化を実現した A* アルゴリズムによるコストに基づく仮説推論システムとの比較実験を行い大幅な速度向上を実現できることを示す.

キーワード コストに基づく仮説推論, 山登り法, A* アルゴリズム, 最適解法

1. まえがき

仮説推論 [1], [2] は, 真か偽かわからない事柄をとりあえず真と考えて (仮説を立てて) 推論を進め, 矛盾なく観測事象を導くことができたなら立てた仮説は正しかったと考える推論法である. 診断 [3], 設計 [4], プランニング [5], 自然言語理解 [6], 画像認識 [7], スケジューリング [8], 文献検索 [9], [10] など様々な分野に応用され, 重要な推論の枠組みとなっているが, 仮説推論の計算複雑度は NP-完全, または, NP-困難である [11] ため, 効率的な推論は困難であり, 実用化に向けて高速化が重要な課題となっている [12].

仮説推論では一般に, 説明を与える仮説集合は複数存在する. しかしながら, 診断, 設計問題などにおいて見られるように, ユーザーが要求する解はすべての説明ではなく, むしろ最も好ましい説明であることが多い. 最良の説明を求めるための推論方法として, 各仮説にコスト (確率や信頼度などから求める) を付け最小コストの説明を求める推論としてコストに基づく仮説推論 (Cost-based Abduction) [13] がある.

文献 [14] では, A* アルゴリズム [15] をコストに基づく仮説推論に導入することにより, 効率的な最適解

探索を実現している. A* アルゴリズムはノードに評価値を与え, 評価値の良いものから順に展開することで効率化することができるが, 最適解の可能性のあるノードをすべて保存しておく必要があるため, 知識ベースの規模が大きい場合, メモリ空間が指数関数的に大きくなる. それに伴い, 膨大なメモリ空間の中から評価値が最も良いノードを探す時間が極めて大きくなる.

一方, 山登り法は評価値が最も高いノードのみを展開し, 他のノードは削除するという方法をとることで, 探索空間の爆発を抑え線形オーダーのメモリ量, 計算時間で準最適解を求めることができるが, 途中で最適解の可能性のあるノードも削除してしまう手法であるため, 最適解の保証のある解を求めることができない.

本論文では, まず, 山登り法の要領で準最適解を効率良く求め, その解を暫定解とし, 後戻り (バックトラック) を行いながら, 暫定解より評価値が悪いノードは枝切りをし, 繰り返し山登り法を行い最適解を求める手法を提案する. また, 本手法では, 時間を制限して途中で現在求められた解を出力することもできる.

従来, 山登り法の改良案として, ランダムに初期ノードを変更して繰り返して探索を行う手法や, シミュレーテッドアニーリングや遺伝的アルゴリズムのように確率的に局所最適解から脱出させる手法があるが, 本提案手法は最適解探索におけるメモリ空間の削減のために山登り法を改良したものである. つまり, 最適解探索に必要なノードは削除しないため, 最適解の保証のある解を求めることができるのに対し, 上で述べた手法では最適解の保証のある解を求めることはできない.

2. 従来研究

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] F の中に存在する仮説間の矛盾を矛盾知識と呼び、次のような制約知識で表記する。

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

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 $\langle F, H \rangle$ の 2 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 $KB_1 = \langle F, H \rangle$ 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 [16]) 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 $\{b(1), d(1)\}$, $\{b(2), d(2)\}$ は矛盾を引き起こすことから、 $p(X, Y)$ の推論木におけるこれらの仮

説集合は取り除く。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 $\{b(1), d(2)\}$ が最適解として与えられる。

2.2 A*アルゴリズムによる仮説推論システム

仮説推論を前節のように定義し、各仮説に好ましさの基準を与えることによって、最適な説明を求める問題は最適解探索の問題に帰着される。

人工知能の分野における最適解探索において、最も効率の良いアルゴリズムの一つとして A*アルゴリズム [15] がある。A*アルゴリズムは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ものである。以下に、

背景知識 F	
$p(X, Y) :- a(X), c(Y).$	
$p(X, Y) :- b(X), d(Y).$	
$a(X) :- e(X).$	
$e(1).$	
$inc :- b(X), d(X).$	

仮説知識 H	コスト
$c(1).$	6
$c(2).$	5
$b(1).$	1
$b(2).$	4
$d(1).$	4
$d(2).$	2

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

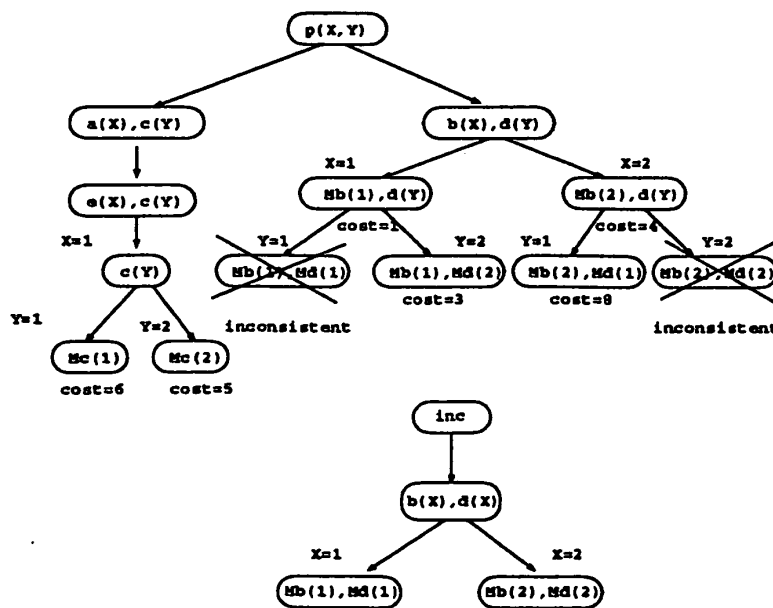


図 2 Prolog による推論
Fig. 2 Inference by Prolog.

コストに基づく仮説推論における評価値の定義 [14] を述べる。

[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。

文献 [14] では、推定値 $\hat{h}(n)$ を述語論理式で表現された知識ベースを命題論理に抽象化した知識ベースをヒューリスティックな知識として用いることによって効率的な探索が実現されているため、本研究でもこの手法を用いているが、詳細については文献 [14] を参照されたい。

3. 提案手法

A*アルゴリズムを用いた仮説推論システム（以降、A*システムと呼ぶ）は、命題論理に抽象化した知識をヒューリスティックな知識として用いることで、効率的な探索を実現したが、A*アルゴリズムは最適解の可能性のある未探索ノードをすべて保存する必要があるため、メモリ量が膨大になるという問題が生じる。それに伴い、膨大になった未探索ノードのリストから最小コストのノードを検索する時間が膨大になるという問題がある。

まず、A*アルゴリズムを山登り法（深さ優先の最良優先探索）に改良し、高速に準最適解（最適解の保証のない解）を探す。次に、その解を暫定解として、後戻り（バックトラック）を起こし、分枝限定法の要領で最適解の可能性のない暫定解より評価値が悪いノードを枝切りしながら、最適解探索に必要なノードがなくなるまで、繰り返し山登り法を用いることで、最適解を求めることができる。この改良した山登り法を導入した仮説推論システムを以降 CARIHC（Cost-based Abduction Reasoning using Improved Hill Climbing algorithm）と呼ぶことにする。

以下に、CARIHC のアルゴリズムを述べる。

Step1（初期設定） 知識ベース KB とゴール g を与える。現在の探索ノード x を g とし、 SG , SG_{tmp} をノードのリストとする。暫定解 $solution$ のコストを ∞ とする。制限時間 T を設定する。

Step2（終了条件 1） 制限時間 T を超えた場合、現

在の暫定解 $solution$ を出力し、終了する。

Step3（展開） x と KB における一段階の SLD 導出で得られるサブゴールをすべて求めて、その集合を SG_{tmp} とする。

Step4（矛盾処理） SG_{tmp} 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_{tmp} から削除する。

Step5（枝切り） 暫定解 $solution$ のコストより SG_{tmp} のもつノードのコストが大きい場合は、最適解に達する可能性のないノードのため削除する。

Step6（暫定解の更新） $SG_{tmp} = \phi$ のとき、 x の仮説集合を暫定解 $solution$ とする。最大の深さ $\max(\text{depth}(SG))$ をもつノードの集合を SG_{tmp} とする。ただし、 $\text{depth}()$ はノードの深さを返す関数である。

Step7（終了条件 2） $SG = \phi$ （最適解探索に必要なノードがない）場合、 x の仮説集合を最適解 $solution$ として出力し、終了する。

Step8（選択） 最小の評価値 $\min\{f(x) = g(x) + \hat{h}(x) | x \in SG_{tmp}\}$ をもつノードを x とし、 SG_{tmp} から x を取り除き、 SG_{tmp} を SG に追加する。Step2 に戻る。

[例 2] 図 3 を用いて提案手法を説明する。仮説推論

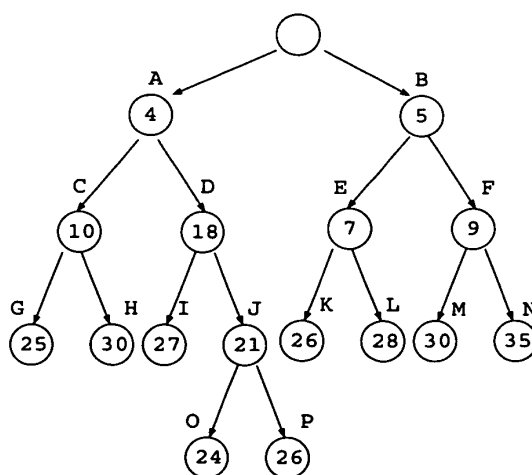


図 3 提案手法の説明のための探索空間の例
Fig. 3 Example of search space for explanation of proposed method.

では、末端のノード（葉ノード）が解となる。この図 3 ではノード $G, H, I, O, P, K, L, M, N$ が解となり、最適解はコスト 24 のノード O である。

CARIHC では深さ優先で評価値の最も良いノードを選び、暫定解を求める。この図では、ノード A 、次にノード C の順に展開され、コスト 25 の暫定解 G が求められる。次にノード D に戻り、ノード I, J を展開する。ノード I は評価値が 27 で暫定解より大きいので削除し、ノード J を展開する。ノード O は暫定解より良い解であるため、暫定解を更新する。ノード P は暫定解より、大きい値のため削除する。そして、ノード B に戻り、展開するノードがなくなるまで探索を続ける。このとき、分枝数 b 、深さ d とすると、CARIHC は A*システムと同じ探索空間 ($O(b^d)$) であるが、ノード展開ごとにおける評価値の最も良い（コストが最も低い）ノードを検索する際の計算量に大きな違いがある。A*システムでは展開途中のノードをすべて保存するためメモリ空間は $O(b^d)$ となり、この中から評価値の最も良いノードを検索しなければならないため、ノード展開ごとに $O(b^d)$ の計算量が必要である。これに対し、CARIHC は山登り法のためのメモリとバックトラックのためのメモリが必要であるため、メモリ空間は $O(b \cdot d)$ となる。更に、評価値の最も良いノードを検索する際には、展開したノードのメモリ空間 ($O(b)$) に対してのみ処理が必要であることから、 $O(b)$ の計算量で十分である。具体的に図 3 を用いて説明する。CARIHC によるメモリ空間の様子を表 1 に、A*システムによるメモリ空間の様子を表 2 に示す。 i はノードの展開回数である。これらの表からわかるように、A*システムはノードを展開するごとにメモリ空間が増大し、展開回数が増えるほど評価値の最も良いノードの検索に多くの計算量が必要であるが、CARIHC では、メモリ空間が大きくなるらず、評価値の最も良いノードは SG_{tmp} 内を検索すればよいため、多くの計算量を必要としないことがわかる。また、枝切りにより更に計算量が少なくなる可能性がある。

以上のように、A*アルゴリズムでは、評価値の最も良いノードを優先して展開することで、効率良く最適解を求めることができるが、途中のノードをすべて保存しておく必要があるため、メモリ空間が膨大になることが問題になるが、CARIHC ではこの問題を解消し、かつ、最適解を求めることができる。

表 1 CARIHC のメモリ空間
Table 1 Memory space of CARIHC.

i	SG_{tmp}	SG	solution
1	{A,B}	ϕ	∞
2	{C,D}	{B}	∞
3	{G,H} {D}	{B,D} {B}	∞ G:25
4	{I,J}	{B}	G:25
5	{O,P} {B}	{B} ϕ	G:25 O:24
6	{E,F}	ϕ	O:24
7	{K,L} {F}	{F} ϕ	O:24 O:24
8	{M,N}	ϕ	O:24

表 2 A*システムのメモリ空間
Table 2 Memory space of A* system.

i	SB
1	{A,B}
2	{C,D,B}
3	{C,D,E,F}
4	{C,D,K,L,F}
5	{C,D,K,L,M,N}
6	{G,H,D,K,L,M,N}
7	{G,H,I,J,K,L,M,N}
8	{G,H,I,O,P,K,L,M,N}

表 3 CARIHC と A*システムの実行時間
Table 3 CPU time of CARIHC and A* system.

n	CARIHC [s]	A* system [s]
8	1.13	1.23
9	1.98	2.48
10	3.47	5.45
11	7.43	11.45
12	10.45	20.22
13	31.34	130.45
14	65.52	630.52

4. 実 験

提案システム CARIHC 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を用いて A*システムとの比較実験を行った。また、各仮説のコストは各素子が故障する確率 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。

CARIHC と A*システムを C++言語で実装し、実験は、CPU PentiumIII 1GHz、メモリ 2GB の AT 互換機を用いて行った。

回路の規模 n に対する CARIHC と A*システムのそれぞれの実行時間を表 3 に示す。この結果より、CARIHC は A*システムに比べ大幅な速度向上を実現

していることがわかる。また、知識ベースの規模が大きくなるにつれて差が大きくなっているため、本提案手法は大規模な問題に対してより有効であると考えられる。

5. む す び

本論文では、コストに基づく仮説推論において、最適解を求められるように山登り法を改良した手法を提案した。

従来の A*アルゴリズムでは、探索空間のすべてのノードを保存しておく必要があり、知識ベースの規模が大きい場合、メモリ空間が指数関数的に大きくなる。それに伴い、膨大なメモリ空間の中から評価値が最も良いノードを探す時間が極めて大きくなる。本研究では、線形オーダのメモリ量で準最適解探索を行う山登り法を最適解が求められるように改良した手法を提案し、コストに基づく仮説推論システムに導入した。また、回路の故障診断問題に適用し、A*アルゴリズムによる仮説推論システムとの比較実験を行い有効性を検証した。

今後は、故障診断の問題という限られた分野のみの実験だけでなく、一般的な評価を行うために設計問題やスケジューリング問題など様々な問題に本提案手法を適用し、有効性を検証していきたい。

文 献

- [1] 國藤 進, “仮説推論,” 人工知能誌, vol.2, no.1, pp.22-29, 1987.
- [2] D. Poole, R. Aleliunas, and R. Goebel, “Theorist: A logical reasoning system for defaults and diagnosis,” in the knowledge frontier: Essays in the Representation of Knowledge, Springer-Verlag, N. Y., 1987.
- [3] R. Reiter, “A theory of diagnosis from first principles,” Artif. Intell., vol.13, pp.57-95, 1987.
- [4] 牧野俊郎, 石塚 満, “制約評価機構付き仮説推論システ

ムとその回路ブロック設計への応用,” 人工知能誌, vol.5, no.5, pp.640-648, 1990.

- [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] 辻野広司, エドガー・ケルナー, 榎谷知彦, “画像認識のためのマルチエージェントによる仮説推論,” 人工知能誌, vol.12, no.3, pp.440-447, 1997.
- [8] 河原耕治, 山本 剛, 佐々木博司, 久保川淳司, 朝原春海, “仮説推論を用いた停電作業スケジューリングに関する一考察,” 電学論 (B), vol.118-B, no.4, pp.460-466, 1998.
- [9] 大澤幸夫, 須川敦史, 谷内田正彦, “ユーザーの変化する興味を理解し表現する文献検索支援システム Index Navigator,” 人工知能誌, vol.13, no.3, pp.461-469, 1998.
- [10] 松村真宏, 大澤幸夫, 谷内田正彦, “AAS: 文書の組み合わせによってユーザの興味を満足する検索システム—コストに基づく仮説推論の一応用,” 人工知能誌, vol.14, no.6, pp.1177-1185, 1999.
- [11] T. Eiter and G. Gottlob, “The complexity of logic-based abduction,” J. Association for Computing Machinery, vol.42, no.1-2, pp.3-42, 1995.
- [12] 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.
- [13] E. Charniak and S. Husain, “Cost-based abduction and MAP explanation,” Artif. Intell., vol.66, pp.345-374, 1994.
- [14] 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.
- [15] 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.
- [16] J.W. Lloyd, Foundations of Logic Programming, Springer, 1984.

(平成 14 年 3 月 15 日受付, 7 月 23 日再受付)