人　工　知　能 81－7
（1992. 3. 5）

0－1整数計画法を用いて最適解を得る仮説推論法

岡本 知樹　　石塚 満
東京大学生産技術研究所

仮説推論は次世代知識ベースシステムの礎として重要であるばかりでなく、診断・設計といった問題への応用が効くといった面でも有用なパラダイムである。しかし、推論速度が十分でないといった問題点を抱えている。これに対し、0－1整数計画法を用いて最適解を得る推論法を考案した。この方法では、全ての知識を制約ととらえることにより制約充足問題におきかえ、ついでこれらを1次不等式に変換して0－1整数計画法を適用している。更に、近似解法として掃き出し補数法を用いることにより、準最適解が高速に得られることを示す。仮説推論の計算複雑度はNP完全であり、従来の探索法に留まっている限り指数オーダの推論時間の壁を越えるのは不可能である。しかし、近似解法というシステマティックなアプローチがその壁を破り得ることを本論文は物語っている。

A Hypothetical Reasoning Method Based on 0-1 Integer Programming

Tomoki OKAMOTO　　Mitsuru Ishizuka
Institute of Industrial Science, University of Tokyo
7-22-1, Roppongi, Minatoku, Tokyo 106, JAPAN

A hypothetical reasoning is an important framework from its theoretical basis and its usefulness for practical problems including diagnosis and design. However, the most serious problem with the hypothetical reasoning is its slow inference speed. In order to improve its inference speed, this paper presents a method based on mathematical programming. In this method, we regard all the knowledges as constraints, and transform the logical knowledge into inequations to apply 0-1 integer programming. This paper also shows that an approximate method called pivot and complements method provides a fast inference time below exponential order for obtaining a near-optimal solution.

1 はじめに

今日の知識ベースシステムは、いわゆる演繹的推論に根ざしたものである為、その知識は完全なものであることが要求される。また、知識の存在する事態には対処出来るものの、少しでも知らない事態に対しては無力とならざるを得ない。こういった知識ベースの脆弱性を打破することが次世代知識ベースシステムを構築する為の重要な鍵となる [石塚 88]。言い替えれば、不完全な知識を扱えるようにすることが必要というわけである。

不完全な知識を扱う推論系は非単調推論と総称されているが、その一つに仮説推論がある。仮説推論では、矛盾の可能性をはらんだ知識を仮説として扱うことにより知識の幅を広げている。他の非単調推論と比べた場合には、最もシンプルな形式でありながら、診断・設計といった実用的問題へ直接応用出来る枠組となっている点が特長的である。しかし、幾多の非単調推論の例に洩れず、推論速度の遅さが大きなボトルネックとなっている。

仮説推論の高速化手法として、推論パスネットワークを利用するシステム [伊藤 91] がある。このシステムでは、ATMS（assumption-based truth maintenance system[de Kleer86]）を採用することにより仮説間の無矛盾性管理を行なう際に生じるバックトラックを回避しており、結果として従来の Prolog による仮説推論に比して推論速度の目覚しい向上を実現している。しかしながら、推論時間が仮説の個数に対して指数関数的に増加する傾向を免れてはいない。最近、仮説推論を初めとする非単調推論の計算複雑度は NP 完全であると証明されており [Kautz89]、この指数オーダの推論速度の壁は、通常の探索による求解法にとどまっていたのでは克服出来ないであろうと予想される。

この為、類推や学習といった、獲得知識の利用により平均値的に指数オーダの壁を破ろうとする試みもなされている [阿部 90、牧野 90]。しかし、問題領域全体をカバーすることが出来ないという欠点を有している。そこで仮説推論の枠組を制約充足問題（CSP:constraint satisfying problem）ととらえ、制約充足の立場から推論の能率化を図ることを考えた。これが今回０－１整数計画法を用いて仮説推論を行なうことに対する所似である。

本稿では、０－１整数計画法を用いて最適解を求める仮説推論法について述べる。

2 論理に基づく仮説推論

2－1 仮説推論の原理

一般に仮説推論とは、真偽の明らかでない事象を取り敢えず真として（仮説を立てて）推論を進め、うまくゴールに到達したら、たてた仮説は正しかったと見なす推論方法である。本論文で取り扱っている仮説推論は、Poole らが Theorist によって提唱した、論理に基づく仮説推論である [Poole87]。これは以下のように説明される。

知識を予め、矛盾の可能性のない完全知識 (CK) と、矛盾の可能性を有する不完全な知識 (IK) とに分ける。そして、ゴール G としてある観測事象が与えられて CK だけでは G を証明出来ない時、CK と合わせても無矛盾かつ、CK と合わせることにより G を証明することが出来るような部分集合 h($\in$ IK) を求める推論である。

h を定式化すると、

$$\mathrm{h}\,(\subseteq \mathrm{IK})\ s.t.\ \mathrm{F}\cup\mathrm{h}\vdash\mathrm{G}\quad\&\quad\mathrm{F}\cup\mathrm{h}\nvdash\Box \tag{2.1}$$

のようになり、これを解仮説と呼ぶ。解仮説は一般に複数個存在し、その中には冗長なものもあり得る。冗長とは、ある解仮説が他の解仮説を包摂することを指す。一般に解仮説と言う時には、冗長なものは考えに入れない。

2－2 仮説推論における解仮説

仮説推論の動作は、ゴールが与えらた時にそれを説明する解仮説を求めることであった。これを実際の問題解決という視点に立って考えてみよう。

設計においては、完全知識を要素部品の動作に、不完全知識を設計方法（要素部品の選択・接続方法など）に、そして観測事象を設計仕様に相当させる。この時求められる解仮説は、設計仕様を満足するような設計結果となる。この場合にも解仮説が複数個存在するかもしれないが、1個を選べばよい（単解探索)。というのは、設計問題の性質上、設計結果は1個得られれば十分だからである。

この時、一般に複数個存在する解仮説の中からどの解を1個選ぶかが問題となってくる。これに対しては、予め解仮説の評価基準を設けてそれを最適化するようなものを選ぶという最適解探索が好ましい。評価基準としては以下の方法が考えられる。即ち、仮説素全てに重み付け係数を割り当てておき、得られる解仮説の中で重み付け係数の和の最小となるものを最適解とする。設計問題においては、仮説素は部品や接続方法に相当するから、その重み付け係数を例えばコストと考えれば、総コストを最小化するような設計仕様が得られることになる。従って、仮説素の重み付け係数及び評価関数に実用的な意味を持たせることも可能である。

本稿では、最適解探索を主眼とする。

2－3 仮説推論における知識ベースの形

高速化を主題としているので、扱う知識の範囲を Horn 節命題論理とする。

解仮説の探索については、最適解探索を対象とする。それは、現実的な側面を考えた際にはすぐれた解仮説を1個求めれば十分なことが多いからである。これについては2－2において言及した。

仮説間の矛盾については、その定義が翻意されることがないものとする。即ち、矛盾に関する知識は完全知識であるとする。

このような条件の元で、更に知識ベースを変換する為に以下の規則を適用する。

・規則1：ルール型仮説からファクト型仮説への変換

不完全知識 IK の中に、ルール型の仮説素

$$p \leftarrow q_1, \cdots, q_n. \tag{2.1}$$

があった時には、新しいアトム r を用意して

$$p \leftarrow q_1, \cdots, q_n, r. \tag{2.2}$$

$$r. \tag{2.3}$$

という2つの知識に変形する。そして、(2.2) を完全知識の集合に加え、(2.1) の代わりに (2.3) を不完全知識の集合に加えればよい。この時、もし解仮説に r が含まれていたならば、元の知識ベースにおける仮説推論において解仮説 $p \leftarrow q_1, \cdots, q_n$ が支持されることとなる。

上記の変換により、仮説素は全てファクト型であるとして差し支えない。

・規則２：観測事象の単節化

ゴールとしての観測事象が

$$p_1, \cdots, p_n \leftarrow q_1, \cdots, q_m. \tag{2.4}$$

という形をしていた時には、新しいアトム o を導入して、

$$\begin{cases} o \leftarrow p_1, \cdots, p_n. \\ q_1. \\ \vdots \\ q_m. \end{cases} \tag{2.5}$$

を完全知識に追加し、観測事象を $o.$とすればよい。

以上をふまえることにより、本論文で取り扱う仮説推論における知識の形は次のいずれかとなる。

1) ファクト型　$a.$

2) ルール型　$a \leftarrow b_1, \cdots, b_n.$

3) 矛盾型　$inc \leftarrow b_1, \cdots, b_n.$

（b_1かつ…かつ b_nが同時に成り立ってはいけない。）

完全知識は、1)、2)、3) いずれの形もとり得る。不完全知識（仮説）は 1) の形しかとり得ない。なお、仮説には重み付け係数を与えるので、実際の知識ベースには

$$hypo(h_1, 2). \qquad \text{（アトム } h_1 \text{は仮説でその重み付け係数は２）} \tag{2.6}$$

のように記述する。

3 仮説推論で扱う知識の制約不等式への変換

3－1 閉世界仮説と完備化

ある知識ベースが与えられている時、それが「全て」を言い表しているとの見方に立つことを、閉世界仮説（closed world assumption）と呼ぶ。閉世界仮説の元では、肯定的知識は全て与えてあるとの仮定が成立する。従って、肯定の結果が得られなければそれは否定の結果を意味すると解釈する。これを失敗による否定（negation as failure）という。例えば、知識ベースとして

$$\begin{cases} a \leftarrow b, c. \\ a \leftarrow d. \\ e \leftarrow f. \\ b. \\ f. \end{cases} \tag{3.1}$$

を考えてみよう。この知識ベースにおいて、a を証明しようとして失敗することは明らかである。この時、a は成立しないと考えるのである。

ここで、「失敗による否定」にモデル論的意味を付与する為、完備化の概念を導入する。簡単に言うと完備化とは、$a \leftarrow b.$（b が成立するならば a が成立する）を $a \leftrightarrow b.$（a が成立するのは b が成立する時であり、またその時に限る）と書き改めることである。また、アトム a があるルールの本体に現れているものの、a を頭部とするようなルールないしファクトがない時には、$\neg\exists a$ を追加する。上の知識ベースに対して完備化を行なうと、

$$\begin{cases} a \leftrightarrow ((b \vee c) \wedge d) \\ \exists b \\ \neg\exists c \\ \neg\exists d \\ e \leftrightarrow f \\ \exists f \end{cases} \tag{3.2}$$

となる。Horn 節命題論理の場合には、完備化と閉世界仮説とは同じ結果をもたらすことが知られている。

さて、仮説推論の枠組を考えた場合、完全知識については閉世界仮説が成り立っていると考えてよい。問題は不完全知識をどう扱うかである。不完全知識を構成する個々の仮説素はその真偽が定かではない。そこで、とりあえず仮説を形式的に真（ファクト）とみなした上で、完全知識について完備化を行なう。そして、完備化された完全知識及び与えられるゴールを制約と見立てて、その制約を充足する仮説素集合の存在を調べればよい。これは、仮説推論を制約充足問題としてとらえていることを意味している。

3－2 与えられた知識の制約不等式への変換

与えられた知識を制約と見る場合、それを数式として取り扱えれば、数理計画法の適用が可能となる。知識を制約不等式に変換することを以下で考える。

まず、命題論理をブール代数を用いてモデル化することを考える。それには、アトムの真／偽を真理値１／０で、そして対等記号"$\leftrightarrow$" を等号"$=$" で置き換えればよい。この時知識ベース中の全ての知識は、

(1)　$p = q_1 \vee \cdots \vee q_n$

(2)　$p = (q_1 \wedge \cdots \wedge q_n) \vee r$

　（但し、$p, q_j, r \in \{0, 1\}, \quad \forall j \in \{1, \cdots, n\}$）

のいずれかの形に帰着される。

次にこれを１次不等式を用いて等価的に表すことを考える。表現方法は１通りではないが、例えば以下の表現が考えられよう。

(1′)　$\dfrac{q_1 + \cdots + q_n}{n} \leq p \leq q_1 + \cdots + q_n$

(2′)　$\dfrac{q_1 + \cdots + q_n + nr - (n-1)}{2n} \leq p \leq \dfrac{q_1 + \cdots + q_n + nr}{n}$

これらの組合せであるは、適宜アトム変数を補うことにより、全て１次不等式を用いて表される。

なお、恒真アトム（ファクト）、恒偽アトムにはそれぞれ値１、０を割り当てればよい。また、ゴールに相当するアトムについては、それを導かなければならないことから値１を割り当てればよい。

ここにおいて、仮説推論を０－１整数計画問題として定式化出来た。つまり、全ての制約不等式を充足する有限個の０－１整数解が解仮説（但し冗長な解を含む）に相当している。評価関数としては、仮説

素の重み付け係数の和を考えたので（２－３参照)、これを最小化することにより最適値及び最適解仮説が得られることになる。

４ ０－１整数計画法を用いた仮説推論システム

４－１ システムの構成

本システムの構成は図２のようになっている。

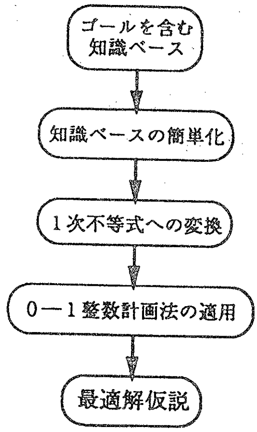

図２ 仮説推論システムの構成

４－２ 知識ベースの簡単化

知識ベースの簡単化とは、０－１整数計画法を適用する前に、与えられたゴールに関連した知識部分を切り出し、かつ値の定まるアトム変数の存在を利用して知識部分を出来るだけ簡単にする処理を指す。これは、推論パスネットワークによる高速仮説推論システム [伊藤 91] で用いられているものと同種の処理であり、簡単化に要する計算量が知識の規模と線形の関係にあることがわかっている [Dowling84]。０－１整数計画法の計算量が問題の規模に対して指数的に増大することを考えると、なくてはならない処理であるといえよう。

簡単化のプロセスは３つのフェーズに大別される。１つめはゴール指向の推論パスの生成、２つめは推論パスの簡略化、３つめは関係ある矛盾型知識の選択である。

推論パスの生成は推論アトム列の求め方に準じている。AND-枝や OR-枝においては、左深さ優先でパスを生成して行く。そして、恒真アトム、恒偽アトム、仮説、既出のアトムのいずれかに出会うことにより、１つの枝の探索を終止する。

推論パスの簡略化とは、真偽の明らかなアトムの真理値を推論パスに沿って流し、パスを簡単にすることである。真理値の流し方にはトップダウンとボトムアップの２通りがある。トップダウンについては、真理値１を AND-枝に流すことで、未知のアトムの値を決めることが出来る。一方、ボトムアップについては、真理値０ないし１を流すことで、枝の簡略化が図れ、場合によっては未知のアトムの値を決めることが出来る。

関係のある矛盾型知識の選択とは、矛盾型知識の中からゴールに関連した部分を切り出す処理を指す。推論パスの簡略化によって得られたルールはいずれもゴールと関連を持つ。よって、この中に現れない仮説素を含む矛盾型知識は不等式に変換する必要がないので選択しない。

４－３ 知識の制約１次不等式への変換

３－２に示した変換法を採用する。矛盾型知識"$inc \leftarrow h_1, \cdots, h_m.$" については、$inc = 0$ とした上で変換を行なう。また場合によっては、前出の不等式のうちの一方のみで済むこともあり得るので、余分な不等式を作らないように注意する。

４－４ ０－１整数計画法の利用

４－３で得られた制約１次不等式を充足しつつ、仮説の重み付け係数の和を評価関数として、これを最小化する。これに対して０－１整数計画法を利用する。

０－１整数計画法としては全整数法 (all integer method)、部分列挙法 (implicit enumeration method)、掃き出し補数法 (pivot and complement method) の３つを採用した。これらについての説明は極簡単なものにとどめる。詳細は参考文献を参照されたい。

全整数法[今野81]は、変数の整数条件を常に維持しつつ、切除平面を次々に導入して目的関数の値を0から漸増させ（最小化の場合）、最終的に制約領域を満たす解を探し当てる手法である。解の探索経路は降下法である。解の収束は一般に悪く、また最適解を得るまで可能解が一つも得られないといった欠点を有する。

部分列挙法[今野81]は、変数に0／1の値を（一部）割り当てた部分解が最適可能解となり得るか否かをテストする事により、（間接的に）全ての可能性を列挙し、最終的な解を得る手法である。原始的であるにもかかわらず、正確な解を求める手法の中では一応有効とされている。解の探索経路は山登り法である。

掃き出し補数法[Balas80]は、準最適解を高速に求める手法である。手順は次の通りである。まず実数緩和問題を解く。ついで非整数度指数を減らすような掃き出しを行ないつつ目的関数の値を漸増させ（最小化の場合）、適宜丸めを行なうことにより可能解を見い出す。続いて、見い出した可能解の周辺を局所探索することにより、解の改善を行う。解の探索経路は折衷法である。

4－5 推論動作の例

以下の知識ベースを例にとる。

完全知識：	c←h.	inc←n,q.	hypo(j,1).
a←d,e.	h←m,n.	inc←i,o.	hypo(k,1).
d←q.	h←o.	inc←f,q.	hypo(l,3).
d←i,j.	m.	inc←g,l.	hypo(n,4).
b←e.	r←m,o,p.		hypo(o,5).
b←f.	r←q.	仮説とその重み付け係数：	hypo(q,2).
b←g.	s←a,g.	hypo(f,2).	
e←p,k.	t←r,s.	hypo(g,2).	ゴール：
e←l.	t←u.	hypo(i,1).	goal←a,b,c.

図3 知識ベースの例

これに対するゴール指向の推論パスを図4に示す。知識ベースの簡単化を行なった結果生成される推論パスを図5に示す。

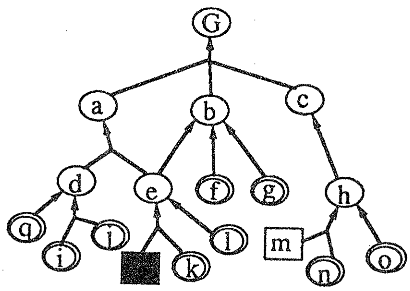

図4　推論パス

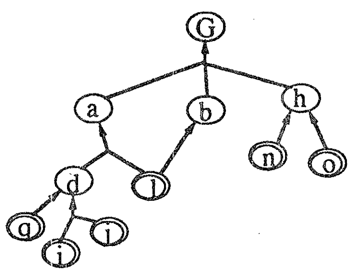

図5　推論パス(最終)

m = 1
G = 1
a = b = c = 1
d = e = 1
q∨(i∧j) = 1
p = 0
l = 1 (解仮説素)
h = 1
n ∨ o = 1

○ 通常アトム ■ 恒偽アトム □ 恒真アトム ◎ 仮説

ついで関係ある矛盾型知識をロードして制約1次不等式に変換した結果が以下の0－1整数計画問題である。

$$\begin{cases} z = i + j + 4n + 5o + 2q \rightarrow \text{最小化} \\ i + j + 2q \geq 2 \\ n + o \geq 1 \\ -n - q \geq -1 \\ -i - o \geq -1 \\ i, j, n, o, q \in \{0, 1\} \end{cases} \tag{4.1}$$

これを解いた結果は以下のようになる。

$$z_{min} = 6, \quad \text{解}:(i, j, n, o, q) = (1, 1, 1, 0, 0) \tag{4.2}$$

これと、簡単化のプロセスで既に得られている解仮説素 l（重み付け係数3）とを合わせて、最終的に

最適解仮説：l, i, j, n、　最適値：9

を得る。

5 推論速度の評価

今回作成したシステムのパフォーマンスを評価する為に、例題を用意してシステムの推論時間を測定した。例題としては、論理回路の故障診断知識ベースを採用した。回路の規模を変えることにより、問題の規模によって推論時間がどのように推移するかを測定した。推論時間は、仮説推論自体の探索に費やされる cpu 時間とした。即ち、「知識の簡単化＋1次不等式への変換＋0－1整数計画問題の求解」にかかる時間である。システムはC言語によって SUN4/380 上にインプリメントされている。

全加算器の故障診断での測定結果を図6の表、及び図7のグラフに示す。

問題番号	仮説数 簡単化前→簡単化後	計算時間 [sec] 簡単化	全整数法	部分列挙法	掃出補数法	目的関数値 最適値	近似値
1	15→10	0.01	0.02	0.01	0.04	9	9
2	30→24	0.02	75.53	9.73	2.25	14	14
3	45→38	0.03	-	1301.04	12.01	19	19
4	60→52	0.06	-	-	35.77	?	24
5	75→66	0.09	-	-	87.57	?	29
6	15→7	0.01	0.00	0.01	0.02	10	11
7	30→24	0.02	9.73	40.60	1.63	15	15
8	45→38	0.04	13094.15	8172.21	11.86	20	20
9	60→52	0.05	-	-	35.27	?	25
11	15→11	0.01	0.02	0.10	0.08	9	9
12	30→25	0.02	83.75	55.03	1.98	14	15
13	45→39	0.03	-	8598.74	11.96	19	21
14	60→53	0.07	-	-	*50.18	?	25
16	15→11	0.01	0.03	0.05	0.12	4	7
17	30→25	0.02	2.56	18.15	2.16	8	15
18	45→39	0.04	1850.41	1877.5	12.01	12	17
19	60→53	0.06	-	-	39.67	?	25

図6 システムのパフォーマンス例（全加算器の故障診断）

簡単化に費やされる時間は極僅かであり、「推論時間≈ 0－1整数計画問題を解くのにかかる時間」と見なせる。図6をグラフに表したものが図7である。図7において縦軸は0－1整数計画問題を解くのにかかる時間としている。

厳密解法である全整数法及び部分列挙法では、当然指数オーダの壁をやぶることが出来ず、それが測定結果にも現れている。一方、掃き出し補数法を用いた場合、近似解法でありながら厳密解法と遜色のないすぐれた解を得られている。肝心の推論時間は、近似解法であることから多項式オーダになっている（回帰分析の結果では、仮説数$^{4.7}$に比例)。

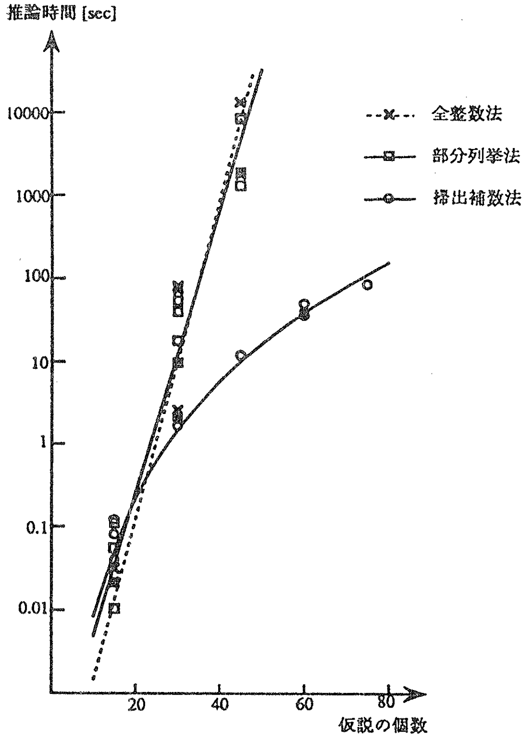

図7　推論時間の推移（全加算器の故障診断)

6 まとめ

0－1整数計画法を用いて最適解を得る仮説推論について言及した。厳密解法では指数オーダの壁を破ることが出来なかったが、近似解法では、最適性を譲歩することにより指数オーダの壁を破ることが可能となった。しかも得られる解は準最適として差し支えない程度のものであった。

以上の結果から、仮説推論を0－1整数計画問題として定式化した場合、掃き出し補数法の有効性が示された。

しかし掃き出し補数法はそのアルゴリズムの性質上、

・0－1整数計画問題が強制約の場合

・制約領域における実数最適解と整数最適解とがかけ離れている場合

には不向きである。つまり、解が存在するにもかかわらず解探索に失敗する可能性が低くないということである。というのは、掃き出し補数法が整数実行可能性よりも最適性に重きを置いているからである。

いかなる0－1整数計画問題に対しても強力であるような解法は（厳密解法であれ近似解法であれ）未だ見つかっていない。掃き出し補数法は、汎用性と効率の良さから極めてすぐれた近似解法であると言われているが、それでも上述のような欠点が避けられない。最近の流れからいえば、個々の0－1整数計画問題の特殊構造を多かれ少なかれ採り入れない限り、実用的なアルゴリズムとなり得ないだろうというのが支配的な考えである。

仮説推論を定式化した0－1整数計画問題においては、制約不等式の形が決まっているというのが特殊構造であろう。制約不等式というのは推論パスに相当しているから、これを活用するのも一計と考えられる。また、削り出し平面（カット）や整数多面体といった代数的なアプローチを元に、ブール代数式の1次不等式への変換法を工夫するのもまたしかりである。

これらを踏まえた上で、より汎用性に富み、かつ最適に近い解をより高速に得るアルゴリズムを模索していきたい。

＜参考文献＞

[石塚 88] 石塚：不完全な知識の操作による次世代知識ベースシステムへのアプローチ、人工知能学会誌、vol.3,No.5, pp.552-562(1988)

[伊藤 91] 伊藤、石塚：推論パスネットワークによる高速仮説推論システム、人工知能学会誌、6, 4. pp.501-509(1991)

[de Kleer86]de Kleer J.：An Assumption-based TMS, Artificial Intelligence, 28, pp.127-162(1986)

[Kautz89]H.A.Kautz, B.Selman：Hard Problems for Simple Default Logics, Proc. KR'89, Tront, pp.189-197(1989)

[阿部 90] 阿部、石塚：推論パスネットワーク上での類推による高速仮説推論システム、情処学会人工知能研資、72-2(1990)

[牧野 90] 牧野、石塚：経験に基づく学習機能を備えた仮説推論システム、１９９０年信学会秋全大、D-155(1990)

[Poole87]D.Poole, R.Aleliunaas and R.Goebel：Theorist; A Logical Reasoning System for Defaults and Diagnosis, in the Knowledge Frontier：Essays in the Knowledge Representation (N.J.Cercone and G.McCalla (eds.)). Springer-Verlag, N.Y.(1987)

[Dowling84]W.F.Dowling, J.H.Gallier：Linear-time Algorithm for testing the Satisfiability of Propositional Horn Formulae, J. of Logic Programming, Vol. 3, pp.267-284(1984)

[今野 81] 今野：整数計画法、産業図書 (1981)

[Balas80]Balas,E., and C. Martin：pivot and Complement- A Heuristic for 0-1 Programming, Management Science, 26, pp.86-96(1980)