論 文 Technical Papers

# コストに基づく仮説推論の２種の連続値最適化問題への置換法とその協調による推論法

## Tranformation of Cost-based Hypothetical Reasoning into Two Continuous Optimization Problems and a Reasoning Method with their Collaboration

松尾 豊 東京大学工学系研究科
Yutaka Matsuo School of Engineering, University of Tokyo
matsuo@miv.t.u-tokyo.ac.jp, http://www.miv.t.u-tokyo.ac.jp/~matsuo/

石塚 満 （同 上）
Mitsuru Ishizuka ishizuka@miv.t.u-tokyo.ac.jp, http://www.miv.t.u-tokyo.ac.jp/~ishizuka/

**keywords:** hypothetical reasoning, augmented Lagrangian method, near-optimal solution, optimization

**Summary**

Cost-based Hypothetical Reasoning is an important framework for knowledge-based systems; however it is a form of non-monotonic reasoning and thus an NP-hard problem. To find a near-optimal solution in polynomial time with respect to problem size, some algorithms have been developed so far using optimization techniques. In this paper, we show two major ways of transforming propositional clauses in the hypothetical reasoning problems into constraints. One transforms the clauses into linear inequalities and is good at getting a low-cost solution, while the other transforms them into non-linear equalities and is good at finding a feasible solution. We them show a method of integrating these two transformations by using augmented Lagrangian method. Here, each variable and constraint is regarded as a processor and the searh is realized by their interaction. Two kinds of processors are derived from the two transformations; the structure of these processors are changed dynamically during the search. The cooperation of these two processors allows to obtain better near-optimal solutions than by our previous SL method. This effect is shown in the experiments using two problems with different problem structures.

## 1. ま え が き

仮説推論は，真か偽か不明な事柄をとりあえず真と考えて推論をすすめ，矛盾なくゴール（観測事象）を導くことができれば立てた仮説は正しかったと考える推論形式である．ゴールを証明する仮説の組は複数ある場合もあるため，コストに基づく仮説推論では各仮説に重みを付与し，その重み和を最小とする解（仮説の組）を最も望ましいものとする [石塚 96]．仮説に与えるコストを適切に設定することにより，回路の故障診断 [牧野 90]，文書検索 [松村 99] などの問題に適用することができる．

真とすることのできる仮説の集合 $H$ があらかじめ与えられている命題論理表現の仮説推論で，ゴールを説明する仮説を見つける問題は NP 完全である．また，コストが最小の仮説を見つける問題は NP 困難となる [Eiter 95]．したがって，コストに基づく仮説推論の最適解を見つけるには，最悪ケースで問題規模に対して指数オーダの推論時間がかかる．そこで，コストが最小に近い準最適解を，平均として多項式オーダの推論時間で求める研究が行われてきた．

NBP 法 [大澤 94] は，0-1 整数計画法の近似解法である掃出し補数法をもとに効率化を図った手法であり，仮説数 $n$ に対し $n^{1.4}$ のオーダの計算時間で準最適解が得られている．しかし，アルゴリズムが複雑であること，メモリを多く必要とすることなどの理由で，大規模問題への適用は難しい．そこで，アルゴリズムがシンプルであり，大規模問題にも適用可能な SL 法 [松尾 98] が提案された．SL 法は，コストの低い解の存在位置の目星をつける線形計画法と，その近傍の仮説推論の解を見つける非線形計画法を組み合わせた手法であり，実験的に仮説数 $n$ に対し $n^{1.8}$ のオーダの計算時間で準最適解を得ている．

これらの手法は，論理に基づく仮説推論問題の各ホーン節を不等式制約で置き換えることで，0-1 整数計画問題に帰着させ，それを緩和した線形計画問題を効率的なシンプレックス法で解く．得られた実数最適解の近傍を探索することにより，準最適となる 0-1 解を求めるというのが要点となっている．

一方，命題論理表現の仮説推論と関連の深い SAT (satisfiability) 問題では，Breakout 法 [Morris 93]，GSAT extention[Selman 93] など，節の重みを変更していく方法が多く提案されており，最近では DLM(Discrete Lagrangian-based search Method) が非常に良い成果を挙げている [Wah 97, Wu 00]．また，SAT 問題を制約なし非線形計画問題に置き換えて解く手法 [Gu 94] なども提案されている．これらは陽には示されていないものの，各節を非線形等式で置き換えた問題を異なる方法で解いているとみなすこともできる．

本論文では，仮説推論で従来から用いられてきた節を不等式制約に置き換える方法と，等式制約に置き換える方法を統合することを試みる．2 種類の異なる置換法による制約を扱うために，並行計算の手法を利用し，変数と制約をそれぞれプロセッサと考えることで，制約の付加・除去を実現する方法を示す．それにより，探索がコストの低い方向にガイドされ，質の高い解を効率的に得ることができる．

本論文は，以下のように構成される．まず，2 章で仮説推論問題を最適化問題として置き換える方法について述べる．その際必要な前処理について述べた後，節を制約に置き換える 2 種類の方法について述べる．3 章では，これら 2 種類の置き換えを同時に扱うアルゴリズムについて述べる．4 章で例題を用いて本手法の評価を行い，最後に本論文のまとめを述べる．

## 2. 最適化問題への置き換え

### 2·1 コストに基づく仮説推論

仮説推論の知識ベースには，対象世界で常に成り立つ背景知識 $\Sigma$ と，常に成り立つとは限らず他と矛盾する可能性を有する仮説知識の集合 $H$ を記述する．$H$ の部分集合を $h$，すなわち $h \subseteq H$ とする．仮説推論の基本動作は次の通りである．ゴール $G$ が与えられた時，まず $\Sigma$ から $G$ が演繹的に証明できるか確かめる．$\Sigma$ からだけでは証明できない時，次の条件を満たす $G$ を証明（説明）するのに必要な解仮説 $h$ を求める．

(1) $\Sigma \cup h \vdash G$ （$\Sigma$ と $h$ から $G$ が証明できる；$\vdash$ は演繹的証明を示す論理記号）
(2) $\Sigma \cup h$ は無矛盾

さらに個々の要素仮説に重みが付されているとき，含まれる要素仮説の重みの和をコストと定義すると，コストが最小の解仮説 $h$ を求めたい場合をコストに基づく仮説推論という [Charniak 94, 石塚 96]．なお，本論文では，背景知識は仮説間の矛盾制約 (Inconsistency ルール: Inc と表す) を含む命題ホーン節で表されるとする．

命題論理表現のコストに基づく仮説推論問題を最適化問題に置き換えるには，まず，命題論理変数 $x_i$ の真/偽を 1/0 に対応させ，目的関数 $f$ を以下のようにおく．

$$f = \sum_{i \in H} w_i x_i \tag{1}$$

(ただし，$w_i$ は要素仮説 $i$ の重み)
この $f$ を最小化するコスト最小の解仮説を求めることになる．

次に，知識ベース中のホーン節知識を変数間の制約に置き換える．前処理，置換法について，以下に順に述べる．

### 2·2 ホーン節知識の前処理

知識ベース KB 中のホーン節を効率的に制約に変換するため，以下の操作を行う．

(1) 真（もしくは偽）であることが自明なファクトノードに真（もしくは偽）の値を代入する．不必要となったホーン節は除去する．
(2) ゴールに関連した知識だけを取り出す．
(3) 完備化を施す．
(4) トップダウンのルールを取り出す．

(1) は，SAT における Unit clause（リテラルが 1 つの節）の除去 [Davis 60] に相当する．(2) は，Relevant reasoning[Levy 97] と呼ばれる処理である．まず，ゴールをヘッド部とするルールを取り出し，さらにそのボディ部に出現するアトムをヘッド部とするルールを取り出す．これを順次行うことによって，ゴールの証明に必要なルールだけを取り出すことができる．取り出したルールのいずれにも出現しない仮説は，ゴールの証明に無関係な仮説であり，このような仮説を含む矛盾制約は除去することができる．推論パスネットワーク法 [伊藤 91] では (1) と (2) を合わせて，推論パスネットワーク形成フェーズと呼んでいる．

(3) では，知識ベースに対し完備化を施す．完備化は，“$q \leftarrow p$” を” ”$p \leftrightarrow q$” とする操作であり，$\{q$ if $p\}$ を $\{q$ if and only if $p\}$ とすることに対応する[*1]．この完備化によって，例えば,「ゴールノードは真，他のノードはすべて偽」という自明な解が得られるのを避けることができる．

完備化の手順は，次のようになる．

(a) あるアトムが複数のホーン節のヘッド部に出現するなら，各節ごとにその名前を付けかえる．さらに，付けかえた複数のアトムの選言をボディ部に，元のアトムをヘッド部にもつ新たな OR ルールを知識ベースに加える．
(b) 各ルールに対して，逆方向のルールを知識ベースに加える．

例えば，以下の知識ベース

$$a \leftarrow b \wedge c. \tag{2}$$
$$a \leftarrow c \wedge d. \tag{3}$$

*1 なお，矛盾制約には完備化の必要はない．

は，(a) の処理により，次のようにヘッド部のアトムの名前の付けかえが行われる．

$$a1 \leftarrow b \wedge c. \tag{4}$$
$$a2 \leftarrow c \wedge d. \tag{5}$$
$$a \leftarrow a1 \vee a2 \tag{6}$$

さらに (b) の処理により，以下のルール（下線部）が付け加えられる．

$$a1 \leftarrow b \wedge c. \tag{7}$$
$$\underline{b \leftarrow a1.} \tag{8}$$
$$\underline{c \leftarrow a1.} \tag{9}$$
$$a2 \leftarrow c \wedge d. \tag{10}$$
$$\underline{c \leftarrow a2.} \tag{11}$$
$$\underline{d \leftarrow a2.} \tag{12}$$
$$a \leftarrow a1 \vee a2. \tag{13}$$
$$\underline{a1 \vee a2 \leftarrow a.} \tag{14}$$

したがって，得られた知識ベースはホーン節でないものも含まれることになる．以下では，下線部のルールをトップダウンのルールと呼ぶことにする．

(4) の処理は仮説推論に特有の処理である．仮説推論はゴールを証明する仮説を見つけるので，探索の過程でゴールの証明に必要なノードは真となる．しかし，ゴールの証明に不必要な要素仮説が真となると解コストが大になるため，コストの小さな解を求めようとすると，必然的に不必要な要素仮説は除去される．そのため，トップダウンのルールだけを考慮して推論を行えばよい．

以上の操作により，仮説推論問題から，以後の探索に必要な少ない数の節（非ホーン節も含む）を取り出すことができる．この節からなる知識ベースを KB' とする．

### 2･3 線形不等式制約への変換

さて，知識ベース中に以下の節があるとする．

$$p1 \vee \neg p2 \vee \neg p3 \tag{15}$$

この節は 3 つのリテラルのうちどれかが真になればよいという要請を表わすので，以下の不等式制約と等価である．

$$x_{p1} + (1 - x_{p2}) + (1 - x_{p3}) \geq 1$$

即ち，節から以下のように等価な不等式制約を作ることができる．

**< 置換 L >**

- リテラル $p$, $\neg p$ をそれぞれ $x_p$, $(1 - x_p)$ に置き換える．
- $\vee$ を+に置き換え，左辺とする．
- (左辺) $\geq 1$ とする不等式制約を作る．

ホーン節を不等式制約に置き換える方法はいくつか提案されていたが [石塚 96, Santos, Jr. 94]，ホーン節の前処理+節の置換 L による制約への変換として説明することができる．この置換 L は，節を充足する 0-1 点を内部に含む線形不等式制約のうちで最小のものである．変数の 0-1 制約を考慮しない場合，この制約を満たすことは，元の節を満たすための必要条件となる．

変数の 0-1 制約を緩和することで，次の問題 LP が得られる．

**問題 LP:**

$$\text{Minimize} \quad f = \sum_{i \in H} w_i x_i$$

$$\text{subject to} \ \ g_j(\boldsymbol{x}) \geq 0 \ \ (j \in C)$$
$$0 \leq x_i \leq 1 \ \ (i \in N) \tag{16}$$

(ただし，$g_j(x)$ は，KB' に含まれる各節を置換 L により線形不等式制約に変換したもの．また，C は制約集合，N は変数集合．)

この線形計画問題は，シンプレックス法により高速に解（実数最適解）を得ることができるが，そのメリットは以下のようにまとめることができる．

- 実数最適解が 0-1 値なら，仮説推論の最適解が得られたことになる．
- そうでなくとも，コストの低い 0-1 解（準最適解）への指針となる．
- 実数最適解のコストは，0-1 最適解の解コストの下界値を示す．
- 問題 LP が実行不可能であれば，実行可能な 0-1 解も存在しない．

仮説推論では，問題 LP で得られた実数最適解の近傍の 0-1 解を探索することにより，コストの低い解を見つける手法が幾つか提案されてきた [大澤 94, 松尾 98]．しかし，仮説推論に比べ制約の厳しい SAT 問題に対しては，すべての値が 0.5 となってしまうことがしばしば生じ，よい指針とならないことが分かっている [Selman 97]．

### 2･4 等式制約への変換

式 (15) の節は，

$$\neg(\neg p1 \wedge p2 \wedge p3)$$

と変形できるので，$\neg p1 \wedge p2 \wedge p3$ が偽となればよい．したがって，以下のように等式制約でも表現することができる．

$$(1 - x_{p1}) x_{p2} x_{p3} = 0 \tag{17}$$

これは一般化すると以下の置換となる．

**< 置換 NL >**

- リテラル $p$, $\neg p$ をそれぞれ $(1 - x_p)$, $x_p$ に置き換える．
- $\vee$ を $\times$ に置き換え，左辺とする．
- (左辺) $= 0$ とする等式制約を作る．

置換 L が”∨” を”+” で置き換えていたのに対し，置換 NL は各節を AND 形式で表した上で “∧” を”×” で置き換えている．

置換 NL により，次の問題 NLP が得られる．

**問題 NLP:**

$$\text{Minimize} \quad f = \sum_{i \in H} w_i x_i$$

$$\text{subject to } h_j(\boldsymbol{x}) = 0 \ (j \in C)$$
$$0 \le x_i \le 1 \ (i \in N) \qquad (18)$$

(ただし，$h_j(x)$ は，KB' に含まれる各節を置換 NL により等式制約に変換したもの．また，C は制約集合，N は変数集合.)

問題 NLP を，制約付き非線形最適化の手法であるペナルティ法で解くと，以下の関数の制約なし最小化問題に帰着される．

$$f' = \sum_{i \in H} w_i x_i + k \sum_{j \in C} |h_j(x)|^2 \qquad (19)$$

この手法は，SAT 問題に対して Gu が提案した手法 [Gu 94] に相当する．また，Wah らは SAT 問題に対して有効な DLM(Discrete Lagrangian Method)[Wu 00] を提案しているが，これは問題 NLP をラグランジュ法で解くことにより得られる手法である．

### 2·5 2 つの変換法について

節を不等式または等式制約に置き換える方法にはどのようなものがあるのだろうか．“∧” を”×” に，“∨” を”+” に置き換えるのは解釈として自然であるので，これらについて考えてみると以下の 4 通りの方法が挙げられる．

(1) ∨ で結ばれたリテラルが真となる制約: 各項の和が 1 以上.

(2) ∨ で結ばれたリテラルが偽となる制約： 各項の和が 0.

(3) ∧ で結ばれたリテラルが真となる制約： 各項の積が 1(以上).

(4) ∧ で結ばれたリテラルが偽となる制約： 各項の積が 0.

各節の否定を取ることによって，(1) と (4)，(2) と (3) はそれぞれ等価なものとして変換することができる．このうち，(2) と (3) は，それぞれ各項が 0，1 であるという制約に分解することができ，問題を解く上ではその方が扱いやすい．(1) と (4) は相互に変換できるが，変換した制約式の形は異なる．この (1) と (4) が，前節までに述べた置換 L と置換 NL に相当する．したがって，ここで述べた 2 つの置換法 L，NL が基本的な 2 つの置換法と考えることができよう．

次に，この 2 つの置換法の特徴について考える．まず，置換 L は，変数が節を満たすための必要条件である．したがって，問題 LP を解いても，元のホーン節を満たす解が得られないかもしれない．図 1 にそのような状況が起こる典型的な例を示す．一方，置換 NL は，変数が節を満たすための必要十分条件である*2.

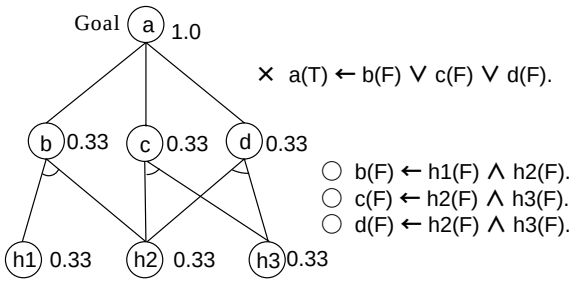


**図 1** 置換 L の解がもとの節を満たさない例：図中の数字は変数値を表す．また，a(T),a(F) は a がそれぞれ真，偽であることを示し，○，×はそれぞれ充足，非充足を示す.

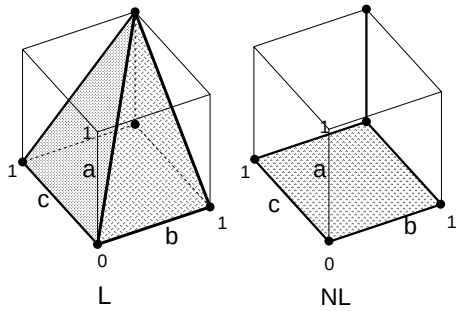


**図 2** 実行可能領域の違い ($a \leftarrow b \wedge c.$ を変換したもの)

図 2 は，”$a \leftarrow b \wedge c$” を置換 L もしくは置換 NL で変換し，$0 \le x_a \le 1, 0 \le x_b \le 1, 0 \le x_c \le 1$ の範囲内での実行可能領域を示したものである．まず，完備化を施し，トップダウンのルールだけ考慮すると，”$b \leftarrow a$”, “$c \leftarrow a$” というルールが得られる．このルールに対し，置換 L を適用し制約 $x_b + (1 - x_a) \ge 1$, $x_c + (1 - x_a) \ge 1$ が得られるが，これによる実行可能領域は図 2 左となる．また，置換 NL により得られた制約 $(1 - x_b)x_a = 0$, $(1 - x_c)x_a = 0$ の実行可能領域は図 2 右となる．いずれも，$x_a = 0$ もしくは $\{x_a = 1, x_b = 1, x_c = 1\}$ をその内部に含んでいるが，置換 L は，実行可能領域を線形な制約で切り出しているため，コストの低い方向に進むことができるのに対し，置換 NL による実行可能領域は凸凹しており，コストの勾配方向に進むのは難しい．

## 3. 2 種類のプロセッサの協調による推論法

置換 L と置換 NL はそれぞれ，コストの低い方向に探索を進める，及び実行可能解を見つけるという特徴がある．したがって，それぞれの特徴を生かすように両方を同時に扱いたい．

*2 節を満たす変数が制約式を満たすことは明らかである．逆に，制約式を満たす変数値が得られたとき，1 ならば真に，0 ならば偽に，どちらでもなければ don't care なので真偽いずれかに決めると，節を充足する．

ここでは，並行計算の手法を用いることによりこれを実現する．並行計算による拡張ラグランジュ法 [Bertsekas 89] では，各変数，各制約をそれぞれプロセッサと考える．つまり $n$ 個の変数と $m$ 個の制約があれば，$n+m$ 個のプロセッサのインタラクションにより探索を行う．したがって，この枠組では制約を付け加えたり取り除いたりすることが，制約に対応するプロセッサを付け加えたり取り除いたりすることに相当する．

本節では，まず拡張ラグランジュ法を用いた並行計算の手法について概説した後，具体的にどのようなアルゴリズムを構築することができるかを述べる．

なお，ここでの並行計算の目的は複数のプロセッサを用いて推論速度を上げることではなく，2 種類の制約の協調をより分かりやすい形で捉えることにある．

### 3·1 拡張ラグランジュ法

以下の制約つき最適化問題を考える．

$$\begin{aligned} &\text{Minimize} \quad f(\boldsymbol{x}) \\ &\text{subject to} \quad h_j(\boldsymbol{x}) = 0 \quad (j \in C) \\ &\qquad\qquad \boldsymbol{x} \in P, \end{aligned} \tag{20}$$

（ただし，$C$ は制約集合，$P$ は $x$ の定義域.）

ラグランジュ法に基づくと，次の最適性の必要条件を満たす点を見つけることが目的となる．

$$\begin{cases} \nabla_x L(\boldsymbol{x},\boldsymbol{\lambda}) = \nabla f(\boldsymbol{x}) + \nabla h(\boldsymbol{x})\boldsymbol{\lambda} = \boldsymbol{0} \\ \nabla_\lambda L(\boldsymbol{x},\boldsymbol{\lambda}) = h(\boldsymbol{x}) = \boldsymbol{0} \end{cases} \tag{21}$$

ここで $L$ はラグランジュ関数で，

$$L(\boldsymbol{x},\boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{j\in C} \lambda_j h_j(\boldsymbol{x})$$

で定義され，$\lambda_j$ はラグランジュ乗数と呼ばれる．ただし，拡張ラグランジュ法では，計算上のメリットから通常のラグランジュ関数にペナルティ項を加えた

$$L_c(\boldsymbol{x},\boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}' h(\boldsymbol{x}) + \frac{c(t)}{2} h(\boldsymbol{x})^2$$

（$c(t)$ は反復 $t$ に対して非減少な正の関数）*3

が用いられる．

式 (21) を勾配法によって解くには，以下のように $\boldsymbol{x}$ と $\boldsymbol{\lambda}$ を交互に更新していく．

$$\boldsymbol{x}^{(k)} := \arg\min_{\boldsymbol{x}} L_c(\boldsymbol{x},\boldsymbol{\lambda}^{(k)}) \tag{22}$$

$$\boldsymbol{\lambda}^{(k+1)} := \boldsymbol{\lambda}^{(k)} + c(t) h(\boldsymbol{x}^{(k)}) \tag{23}$$

式 (22) は，$L_c$ を最小にするような $\boldsymbol{x}$ の値を $\boldsymbol{x}^{(k)}$ に代入することを表す．即ち，違反している制約のラグランジュ乗数を大きくしながら，現在のラグランジュ関数を $x$ について最小化するというプロセスを繰り返す．

*3 本論文中では，$c(t)$ は反復 30 回ごとに 2 倍としている．一般的には数十回の反復ごとに 2〜10 倍するのがよいとされている [Bertsekas 89].

さて，仮説推論問題は制約と変数が局所的に結び付いた構造をしている．したがって，1 つの変数もしくはラグランジュ乗数は，近傍のラグランジュ乗数もしくは変数値が分かれば，更新することができる．

ここで，1 つの変数の値の更新や受渡しを行うプロセッサを**変数プロセッサ**，ラグランジュ乗数の値の更新・受渡しを行うプロセッサを**制約プロセッサ**と呼ぶことにする．各変数プロセッサはまわりの制約プロセッサ $j$ から $\partial L/\partial\lambda_j^{(k)}$ を受取り，それをもとに，以下により変数値を更新する．

$$x_i^{(k+1)} := \arg\min_{0\le x_i\le 1} L_c(\boldsymbol{x}^{(\boldsymbol{k})},\boldsymbol{\lambda}^{(k)}) \tag{24}$$

一方，各制約プロセッサはまわりの変数プロセッサ $i$ から変数値 $x_i^{(k)}$ を受取り，それをもとに，以下によりラグランジュ乗数を更新する．

$$\lambda_j^{(k+1)} := \lambda_j^{(k)} + c(t) h_j(\boldsymbol{x}^{(k)}) \tag{25}$$

すべての制約プロセッサと変数プロセッサは，値の更新と伝達を繰り返し，いずれの値も変わらなくなれば (収束すれば)，問題 (20) の局所最適解が得られたことになる．なお，問題 (20) は等式制約だけを考えているが，不等式制約にも拡張することができる．

以上が拡張ラグランジュ法による並行計算の概要であるが，この手法を用いるメリットとして，置換 L による制約 $g(\boldsymbol{x}) \ge 0$ と置換 NL による制約 $h(\boldsymbol{x}) = 0$ を同時に扱うことが可能である点*4，各プロセッサはローカルな値の受渡しだけを行うので，探索の途中でプロセッサの構成を変更 (新たなプロセッサの追加，削除) が行えるという点が挙げられる．

### 3·2 アルゴリズム

前節の拡張ラグランジュ法により，複数の制約を動的に扱うための準備ができた．以下では，置換 L による制約を扱うプロセッサを**制約プロセッサ L**，置き換え NL による制約を扱うプロセッサを**制約プロセッサ NL** とよぶことにする．

まず，制約プロセッサ L と変数プロセッサにより探索を行った場合には，シンプレックス法と同じく線形計画問題を解いていることになる．得られる解は実数最適解となる．一方，制約プロセッサ NL と変数プロセッサで探索を行った場合，得られる解は 0-1 値の実行可能解となる．

したがって，まず制約プロセッサ L により探索を行い，0-1 解が得られない場合，制約プロセッサ NL に切替える，もしくは制約プロセッサ NL を追加する方法がよいと思われる．

ここでは，以下の 2 種の戦略を試みた．

**プロセッサの協調による推論法 [全節戦略]**

はじめに制約プロセッサ L，その後制約プロセッサ L と制約プロセッサ NL を併用する方法である．

*4 $g(\boldsymbol{x}) \ge 0$ を扱うには、$\max(0, -g(\boldsymbol{x})) = 0$ と考える．

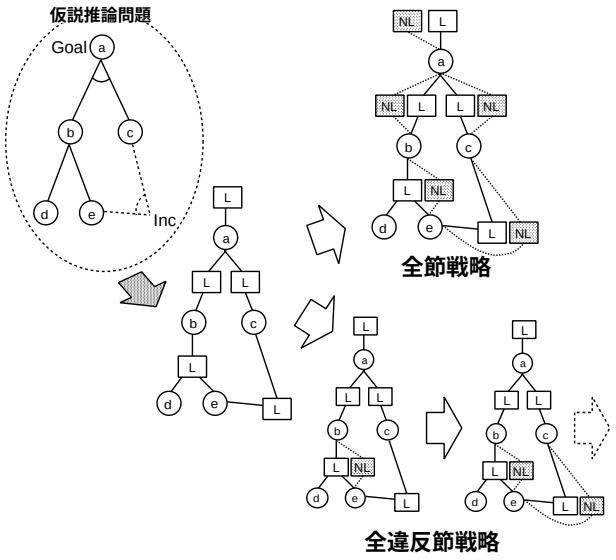


**図 3** 制約プロセッサ NL を加えていく 2 種の戦略

(1) **初期フェーズ** 制約プロセッサ L と変数プロセッサで探索を行う．0-1 解が得られれば終了．
(2) **構成変更** すべての制約に制約プロセッサ NL を加え，探索を再開する．
(3) **終了条件** 実行可能解が得られれば (4) へ．規定の反復数を経ても解が得られなければ探索失敗．
(4) **解改善フェーズ** 不必要な仮説が含まれていれば除去し，終了．

**プロセッサの協調による推論法 [全違反節戦略]**

収束ごとに，違反している制約に順次制約プロセッサ NL を加えていく方法である．

(1) **初期フェーズ** 制約プロセッサ L と変数プロセッサで探索を行う．0-1 解が得られれば終了．
(2) **構成変更** 違反している制約すべてに制約プロセッサ NL を取りつけ，探索を再開する．
(3) **終了条件** 探索の途中で実行可能解が得られれば (4) へ．収束しても実行可能解が得られなければ (2) へ．規定の反復数を経ても解が得られなければ探索失敗．
(4) **解改善フェーズ** 不必要な仮説が含まれていれば除去し，終了．

これらの戦略を図 3 に示す．全節戦略は構成変更が 1 回なのに対し，全違反節戦略では反復的に構成変更を行う．制約プロセッサ NL を必要以上に追加しないので，よりコストの低い解を探索する可能性が高い．

この他の戦略として，全 OR 節戦略 (すべての OR ルールと仮説間の矛盾を表す Inc ルールに制約プロセッサ NL を追加する)，最大違反節戦略 (最も違反度の高い制約にだけ制約プロセッサ NL を追加する)，全節取り換え戦略 ( すべての制約に対して，制約プロセッサ L を制約プロセッサ NL で取り換える) なども試みたが，実験的にここに挙げた 2 つが探索時間と解の質の面から代表的であったので，他は省略する．

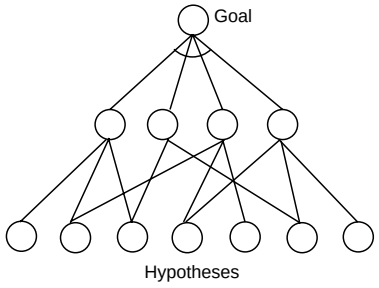


**図 4** 集合被覆問題の構造

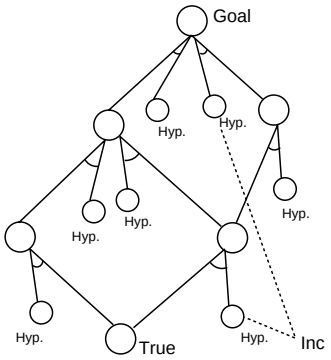


**図 5** 最短経路問題の構造

なお，実装上は初期フェーズではシンプレックス法を用いる．次章で行った評価実験の問題規模の範囲内では，制約プロセッサ L による初期フェーズよりもシンプレックス法の方が高速であったためである．そして，得られた変数値を変数プロセッサに，双対変数の値を制約プロセッサ L に渡し，構成変更フェーズ以下のアルゴリズムを続行する．

## 4. 異なる構造をもつ問題による評価実験

本論文で述べた手法に対し評価実験を行った．システムは C 言語で記述し，SUN Enterprise 400MHz 上で実行した．

ここでは 2 種類の問題を用いた．1 つは，NP 完全問題である集合被覆問題を仮説推論問題により表したものである (図 4)．いくつかの代替要素から仕様を満足しコストを最小化する設計問題を単純化した問題であり，与えられた概念を説明する適切な文書集合を見つける [松村 99] などの応用例がある．問題構造としては，横長のグラフ構造となる．もう 1 つは最短経路問題である．ある 2 点間の経路を通るかどうかを仮説で表し，ある地点に到達した状態を中間ノードの真偽で表す (図 5)．最短経路問題はクラス P に属するが，仮説間の矛盾まで考えると組合せ的な問題となる．あるノードの真偽がその祖先ノードの真偽に影響する縦長のタイプの問題となっており，回路の故障診断問題などがこのような縦長のグラ

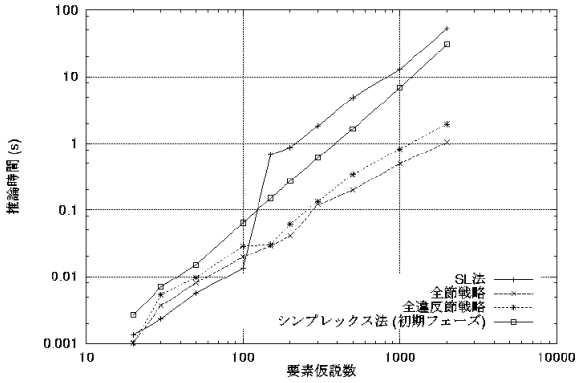


**図 6** 集合被覆問題の推論時間

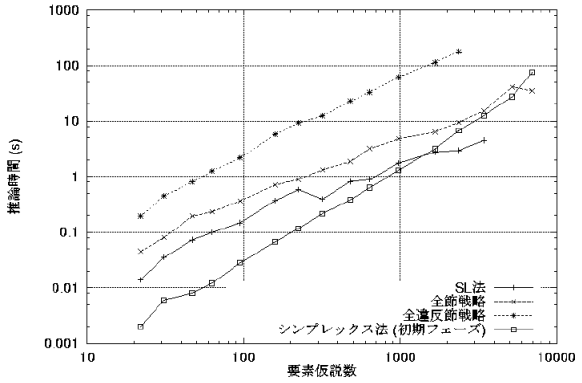


**図 7** 最短経路問題の推論時間

**表 1** 集合被覆問題の解コスト

| | SL 法 | 全節戦略 | 全違反節戦略 | 最適解 |
|---|---|---|---|---|
| 平均コスト | 107.3% | 101.7% | 101.6% | 100.0% |

**表 2** 最短経路問題の解コスト

| | SL 法 | 全節戦略 | 全違反節戦略 | 最適解 |
|---|---|---|---|---|
| 平均コスト | 136.3% | 133.6% | 116.6% | 100.0% |

フ構造をしている．

これらの問題のスケールを変えて，プロセッサの協調による推論法の全節戦略と全違反節戦略に対して推論時間を測定したものが，図 6，図 7 である．また，得られた解コストを表 1，表 2 にそれぞれ示す．比較のために SL 法の推論結果も記した．なお，図中の推論時間は，初期フェーズ以降の推論時間であり，初期フェーズのシンプレックス法による計算時間は図中に別に記した．

まず，集合被覆問題に対しては，全節戦略，全違反節戦略ともに，推論時間，解コストともに SL 法よりもよい．しかし，実際には初期フェーズのシンプレックス法の計算時間が支配的となるため，推論時間に関しては大きな差はないことになる．なお，SL 法が要素仮説数 100 を越えたところから急に時間がかかっているのは，局所解に陥ることが頻繁に発生するようになるためである．

一方の最短経路問題に対しては，逆に SL 法の推論時間が最も速い．全節戦略はそれよりもやや遅く，全違反節戦略は非常に遅くなる．しかし，解の質では，全違反節戦略が非常に良く，次いで全節戦略，SL 法の順となっている．即ち，簡単な解はすぐに見つかるが，コストの低い解を見つけるのが難しい問題であるといえる．

初期フェーズ以降の探索は，全節戦略では変数プロセッサと各節に対応するプロセッサ L およびプロセッサ NL により行われる．したがって，変数の数を $n$，節の数を $m$ とすると，プロセッサの数は $n+2m$ である．全違反節戦略では，変数プロセッサと各節に対応するプロセッサ L および違反した節に対応するプロセッサ NL により探索が行われる．プロセッサの数は，$n+(1+\alpha)m$ であり，$\alpha$ は集合被覆問題では平均 18%，最短経路問題 1.8%であった．反復回数は集合被覆問題の場合では多くても 1 回であったのに対し，最短経路問題では，ほとんどの場合，1 回の反復で 1 個ずつしかプロセッサ NL が加わらないため，反復回数はおよそ $m \times 1.8\%$ 回であった．

集合被覆問題と最短経路問題で探索時間および解の質に差がでるのは，次のように考えることができる．[大澤 98] で示されているように，仮説推論問題がシンプレックス法 (つまり置換 L) だけで解けるかどうかは，問題のグラフ構造に依存する．最短経路問題の方がより閉路の多い構造をしており，置換 L によるコストの低い解への推測がうまく働かない場合が多い．制約プロセッサ NL を加えることにより，この込み入った構造を少しずつほどいていくことができるが，集合被覆問題が「ほどけやすい」のに対して，最短経路問題の方は「ほどけにくい」ため，少しずつ制約プロセッサ NL を加えていく全違反節制約が良い解を得るための有効な戦略となる．以上は直観的な説明であるが，今後どのような問題に対しどの戦略が有効か検討を重ねていく予定である．

以上をまとめると，推論時間に関しては初期フェーズのシンプレックス法の計算時間のオーダが大きく，手法ごとの差はあまりない．解コストは SL 法よりもよい解が得られており，単純に実数最適解の近傍を探索する SL 法と比べ，制約プロセッサ L と制約プロセッサ NL が協調することで，よりコストの低い方向に探索がガイドされていることが示されている．

また当然，推論時間と解コストのトレードオフがあるが，パラメータだけが変更可能なアルゴリズムに比べて，プロセッサの構成を変更できる本手法は，問題の性質やユーザの希望に応じて，より柔軟な対応が今後可能である．

## 5. ま と め

本論文では，仮説推論を不等式及び等式制約による最適化問題へ置き換える 2 種類の置換をもとに，制約プロセッサ L と制約プロセッサ NL という 2 種類のプロセッ

サを用い，その構成を変更することでより質の高い解を得る手法を提示した．この手法は，変数値をどう変更するかではなく，探索を行うプロセッサの構成をどう変更するかという，一段メタなレベルに着眼点がある．[Gomes 00] では探索アルゴリズムの今後の発展の方向として，問題を解く前の静的な分析+探索中に得られた情報による動的なアルゴリズムの変更が重要であると述べられているが，本手法は，探索中に動的にアルゴリズムを変化させることのできる枠組みであり，探索問題の難易度に応じた効率の良いアルゴリズムに向けての１つの重要なアプローチの方向を示していると思う．

## ◇ 参 考 文 献 ◇


[Bertsekas 89] Bertsekas, D. P. and Tsitsiklis, J. N.: *Parallel and Distributed Computation*, Prentice-Hall (1989).

[Charniak 94] Charniak, E. and Shimony, S. E.: Cost-based abduction and MAP explanation, *Artificial Intelligence*, Vol. 66, pp. 345–374 (1994).

[Davis 60] Davis, M. and Putnam, H.: A Computing Procedure for Quantification Theory, *Journal of the Association for Computing Machinary*, Vol. 7, No. 3, pp. 201–215 (1960).

[Eiter 95] Eiter, T. and Gottlob, G.: The complexity of logic-based abduction, *Journal of the Association for Computing Machinary*, Vol. 42, No. 1–2, pp. 3–42 (1995).

[Gomes 00] Gomes, C.: Structure, Duality, and Randomization: Common Theme in AI and OR, in *Proceedings 17th National Conference on Artificial Intelligence (AAAI-2000)*, pp. 1152–1158 (2000).

[Gu 94] Gu, J.: Global Optimization for Satisfiability Problem, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 6, No. 3, pp. 361–381 (1994).

[石塚 96] 石塚満：知識の表現と高速推論, 第 6 章, 丸善 (1996).

[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, Vol. 6, No. 4, pp. 501–509 (1991).

[Levy 97] Levy, A. Y., Fikes, R. E., and Sagiv, Y.: Speeding up Inferences using Relevance Reasoning: a Formalism and Algorithms, *Artificial Intelligence*, Vol. 97, pp. 83–136 (1997).

[牧野 90] 牧野, 石塚：制約評価機構つき仮説推論システムとその回路ブロック設計への応用, 人工知能学会誌誌, Vol. 5, No. 5, pp. 640–648 (1990).

[松尾 98] 松尾, 二田, 石塚：SL 法：線形・非線形計画法の併用によるコストに基づく仮説推論の準最適解計算, 人工知能学会誌, Vol. 13, No. 6, pp. 953–961 (1998).

[Morris 93] Morris, P.: The Breakout Method for Escaping from Local Minima, in *Proc. AAAI-93*, pp. 40–45 (1993).

[大澤 94] 大澤, 石塚：仮説推論における準最適解を多項式時間で計算するネットワーク化バブル伝播法, 電子情報通信学会論文誌, Vol. J 77-D-II, No. 9, pp. 1817–1829 (1994).

[大澤 98] 大澤, 石塚：コストに基づく仮説推論を多項式時間で達成する新しい十分条件, 人工知能学会誌, Vol. 13, No. 3, pp. 415–423 (1998).

[Santos, Jr. 94] Santos, Jr., E.: A Linear Constraint Satisfaction Approach to Cost-Based Abduction, *Artificial Intelligence*, Vol. 65, No. 1, pp. 1–27 (1994).

[Selman 93] Selman, B. and Kautz, H.: Domain-Independent Extensions to GSAT:Solving Large Structured Satisfiability Problems, in *Proc. IJCAI-93*, pp. 290–295 (1993).

[Selman 97] Selman, B., Kautz, H., and McAllester, D.: Ten Challenges in Propositional Reasoning and Search, in *Proceedings 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, pp. 50–54 (1997).

[Wah 97] Wah, B. W. and Shang, Y.: Discrete Lagrangian-Based Search for Solving MAX-SAT Problems, in *Proc. IJCAI-97*, pp. 378–383 (1997).

[Wu 00] Wu, Z. and Wah, B. W.: An Efficient Global-Search Strategy in Discrete Lagrangian Methods for Solving Hard Satisfiability Problems, in *Proceedings 17th National Conference on Artificial Intelligence (AAAI-2000)*, pp. 310–315 (2000).

[松村 99] 松村, 大澤, 谷内田：AAS: 文書の組合わせによってユーザの興味を満足する検索システム, 人工知能学会誌, Vol. 14, No. 6, pp. 245–253 (1999).


〔担当委員：佐藤　健〕

2000 年 12 月 4 日　受理

## 著 者 紹 介

**松尾　豊**(学生会員)

1997 年東京大学工学部電子情報工学科卒業．現在同大学院博士課程 3 年在学中．推論，数理計画法，探索アルゴリズムだけでなく，その結果を人間にとってどう役に立てるかにも興味があり、価値の高い情報の提示を目指している。電気学会，AAAI 会員．

**石塚　満**(正会員)

1971 年東京大学工学部電子卒業．1976 年同大学院博士課程修了．工学博士．同年 NTT 入社，横須賀研究所．1978 年東京大学生産技術研究所助教授，同教授を経て，1992 年より工学部電子情報工学科教授．研究分野は人工知能，知識処理，マルチモーダル擬人化エージェント，ネットワーク化知的情報環境．IEEE，AAAI，情報処理学会，電子情報通信学会，映像情報メディア学会，画像電子学会等の会員．