解　説

整合ラベリング問題と応用†

西　原　清　一††

1. まえがき

整合ラベリング問題 (consistent labeling problem) は，パターン解析や人工知能などの分野においてしばしば見かける一群の問題の総称である．本稿では，この問題を，Haralick とShapiro[13] にならって，CLP と略記する．CLP は，一般に問題の対象が，「複数個の構成要素」と「構成要素間に成り立つ拘束条件」とによってモデル化できる場合を扱う．たとえば，無向グラフの同形部分グラフの探索問題を例にとると，複数個の構成要素とは頂点のことであり，拘束条件とはどの頂点同士が接続しているかを表す辺のことである．このように複数構成要素と拘束条件とで定義される目標グラフと同形な部分グラフを与えられたグラフの中から探し出す問題である．ここで，探すべき無向グラフは，これによって過不足なく，すなわち完全に表現できることに注目したい．

CLP は，局所的な拘束条件が複数個あって，それらを全体的な観点から整合化させる，すなわち，すべての拘束条件を満足させるような問題である．したがって，CLP と意識されていなくても，実は CLP とみなすことができるような問題は非常に多い．たとえば，連立一次方程式において，変数は先の構成要素に対応し，各等式はその式に現れる変数の間に要請される局所的な拘束条件を表しているわけである．そして，連立方程式を解くというのは，全体的な観点から，すべての等式を満足するように変数に具体的な値を割り当てる作業を指す．

このように，CLP は広い範囲の問題をカバーする普遍的な性格をもっているが，拘束条件の充足問題として意識され始めたのは，図形や画像のパターン解析やグラフ・アルゴリズムなどの問題においてであろう．前者の初期の例としては，Huffman[17]，Waltz[34]，Rosenfeld[26] などがある．後者の例としては，Ullmann[32] などがある．また，Bertelè[1] は，拘束充足を非逐次的な動的計画法で解く手法を整理して解説したものである．CLP を，独立した一つの問題として形式的に捉え，理論的に考察した文献は，70 年代後半に集中的に現れた[7),9)~11)]．なかでも Haralick ら[13] は，CLP の一般的な定義を与え，諸性質を明らかにした基礎文献である．CLP は，拘束充足 (constraint satisfaction，制約満足とも) 問題[20]，関係整合 (relational consistency) 問題[22]，充足割付け (satisficing assignment) 問題[10] など，多くの呼び方がある．

CLP が解を有するかどうかを決定する問題は，NP 完全な組合せ探索 (combinatorial search) 問題であり，効率のよい汎用アルゴリズムは存在しない．したがって，具体的な応用問題ごとに，その特徴を生かした解法を開発する必要がある．本稿では，CLP の解法を，木探索法，弛緩法，併合法に大分類し，それぞれの解法を解説した後，処理の高速化の方法について述べる．以下，まず **2.** では，CLP の具体例として線画解釈を取りあげ，CLP の理解を深める．**3.** で，CLP の定義及び解法について述べる．**4.** では，応用例を紹介する．**5.** では，今後の展望を概観する．

2. 具体例

CLP の具体的なイメージを得るため，よく知られた線画の解釈の問題[17] を例として取りあげる．ただし，これは CLP の理解を深めるためであって，必ずしも実用的な応用例あるいは典型的な問題というわけではない．

図-1 の線画の稜線を解釈することを考えよう．線

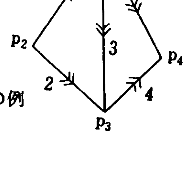

図-1　線画の例

---

† Consistent Labeling Problems with Applications by Seiichi NISHIHARA (Institute of Information Sciences and Electronics, University of Tsukuba).

†† 筑波大学電子・情報工学系

図-2 線分の接続の型

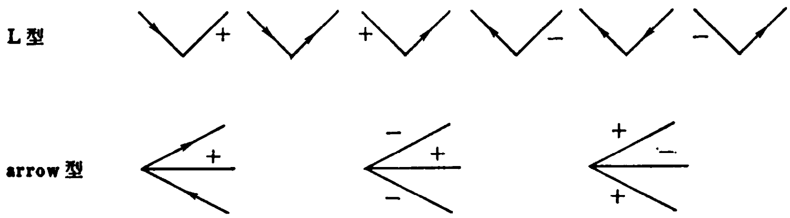

図-3 可能な解釈（ラベリング）

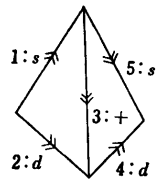

図-4 解の例

画は，三面頂点のみからなる多面体を一般位置から見た情景で，影はないものとする．読者は，これが三角錐であることを容易に理解されるであろう．**1.** で述べた「複数個の構成要素」とは，図-1 では稜線を表す５本の線分１～５である．線分につけられた２重矢印は後で必要になるもので，ここでは無視しておく．また，「構成要素間に成り立つ拘束条件」とは，各頂点 p_1～p_4 で出会う２本または３本の線分間の相互関係のことである．

Huffman[17] の観察によると，線分は図-2 に示す４種の接続の形しかないこと，また，各形には有限個の解釈しかありえないことが分かっている．それらのうち，arrow 型とＬ型について可能な解釈を図-3 に示す．ただし，線分に割り当てられる解釈のうち，矢印→は，稜線上を矢印方向に進むとき，右側は物体の面であり，左側は背景またはより向こうに存在する他の面であるという意味である．また，＋は凸でかつ両側の面が見えている稜を，－は凹の稜を表す．問題は，各頂点において図-3 の拘束条件を適用しつつ，全体として調和するようにすべての線分に→，＋，－の解釈を割り当てることである．ところで，矢印→には方向があるので，図-1 の２重矢印を線分固有の基準方向と考え，矢印→がこれと同じ向きの場合は s (same)，逆向きの場合は d (different) を割り当てることにしよう．解の一例を図-4 に示す．

3. 整合ラベリング問題

3.1 定義及び各種の変形

CLP は，４つ組 (U, L, T, R) で定義される[13,44]．$U=\{1, \ldots, M\}$ はユニット集合で，各要素（‘ユニット’）は先に述べた対象の構成要素にあたる，L はラベル集合で，各要素（‘ラベル’）はユニットに割り当てられる解釈や値の候補を表す．$T=\{t_1, \ldots, t_N\}$ はユニットの多項組（‘ユニット組’と呼ぶ）の集合で，‘ユニット拘束関係’と呼ぶ．ユニット組は，その成分であるユニット同士が局所的な拘束条件の下にあることを示している．そして，実際にどのようなラベル組が許されるかは，$R=\{R_1, \ldots, R_N\}$ で具体的に与える．各 R_i $(1 \leqq i \leqq N)$ は‘ラベル拘束関係’といい，ユニット組 t_i $(\in T)$ に対して付けることのできる可能ラベル組の集合を表す．N は，局所的な拘束条件の個数である．以後，これらの組 (t_i, R_i) を‘拘束条件ペア’と呼ぶ．拘束条件ペアとは，局所的な拘束条件を表す単位である．とくに添字を明示する必要のない場合は，(t, R_t) と書くこともある．なお，ユニット，ラベルの替わりにそれぞれ変数，値という用語を用いた例もある．

［例１］（CLP の例）

先の線画解釈の問題を定式化すると次のようになる．

$U=\{1, 2, 3, 4, 5\}$,

$L=\{s, d, +, -\}$,

$T=\{t_1, t_2, t_3, t_4\}$,

$t_1=(1, 3, 5)$, $t_2=(1, 2)$,

$t_3=(4, 3, 2)$, $t_4=(4, 5)$,

$R=\{R_1, R_2, R_3, R_4\}$,

$R_1=\{(s,+,s),(-,+,-),(+,-,+)\}$
$R_2=\{(d,+),(d,s),(+,s),(s,-),(s,d),(-,d)\}$,
$R_3=\{(d,+,d),(-,+,-),(+,-,+)\}$,
$R_4=\{(s,+),(s,d),(+,d),(d,-),(d,s),(-,s)\}$.

(例1終り)

CLP を解くとは，U から L への写像 h であって，任意の t_i $(=(u_1,\ldots,u_k))$ $(\in T)$ $(1\leqq i\leqq N)$ について，

$H(t_i)$ $(=(h(u_i),\ldots,h(u_k)))\in R_i$

を成り立たせるようなものをすべて見つけだすことである．すなわち，ユニット $(1,\ldots,M)$ に対するラベル組 $\lambda=(l_1,\ldots,l_M)$ であって，条件 $\forall t_i(\in T)$ $(\lambda(t_i)\in R_i)$ を満たすようなものをすべて見つけだすことである．ただし，$\lambda(t)$ は，λ の t に関する射影，すなわち，λ のうち t の成分ユニットに対応するラベルのみを抜きだして，相対的な順序を乱さずに並べたものである．上記の［例1］の解は次の三つである．

(1，2，3，4，5)
解1：(s，d，$+$，d，s)
解2：(s，$-$，$+$，$-$，s)
解3：($-$，d，$+$，d，$-$)

このうち，解1が先の図-4 と同じものである．

上記の CLP の定義に制限を加えたり，拡張を施したりして，種々の変形問題を定義することができる．主な変形を2～3述べる．

まず，ユニット拘束関係及びラベル拘束関係の次元を一定にした問題がある．これを‘同次 CLP’と呼ぼう．上記の例は，2次と3次が混在しているので，同次 CLP ではない．しかし歴史的には，2次の同次 CLP がもっともよく研究されてきた[19),22),23)]．また，先の CLP の定義では，どのユニットについても，取ることのできる候補ラベルの定義域は L であったが，ユニット i については L_i というようにユニットごとに固有のラベル集合を対応させるような定義の仕方もある．しかし，このような定義は，1次の拘束関係を導入すれば，本来の定義でも表現できるので，原理的に新しいタイプの問題ではない．実際，任意のユニット i に対して，1次のユニット組 (i) を T に追加し，これに対応するラベル拘束関係として L_i を追加すればよい．

次に，CLP の重要な変形として，ラベル組に重みをつけた CLP がある．これを‘概整合 CLP (inexact CLP)’という．たとえば，ラベル拘束関係に含まれる各ラベル組に対してそれぞれの相対的な平均出現頻度が重みとして与えられている場合などがそうである．この場合は，重みの値の大きいラベル組ほど良い解釈ということになる．逆に，解釈の妥当性を誤差で表す場合のように値が小さいほど良い場合もある．以下では，後者，すなわち重みが誤差を表す場合のみを考える．

概整合 CLP は，4つ組 $(U,L,T,\mathcal{W})$ で定義できる[31),41)]．ここに，U，L，T は先の定義と同じである．また，$\mathcal{W}=\{W_1,\ldots,W_N\}$ であって，各 W_i $(1\leqq i\leqq N)$ は‘ラベリング-誤差関数’といい，次のような部分関数である．

$W_i: L^{|t_i|}\rightarrow R$ (非負の実数全体)

ここに $|t_i|$ はユニット組 t_i の次元を表す．W_i は，ユニット組 t_i に対する可能なラベル組とその誤差を表す．つまり，誤差が小さいラベル組ほどユニット組 t_i の局所的解釈としての妥当性が高い．局所的なラベル組の誤差をどのように集積して全体の誤差とするかは，問題の性質によって決めるべきである．一般的には，重要性の高い局所的拘束条件の誤差ほど，全体誤差 ε への影響が大きくなるように決めるのがよい．もっとも簡単な計算法は，各ユニット組に対応する局所的解釈であるラベル組の誤差を単に加え合わせただけの値，すなわち単純和を全体の誤差 ε と見なす方法である．

［例2］（概整合 CLP の例）

4つ組 $(U,L,T,\mathcal{W})$ のうち，U，L，T は［例1］と同じ．

$\mathcal{W}=\{W_1,W_2,W_3,W_4\}$,
$W_1=\{(s,+,s)\rightarrow 0.1,\ (-,+,-)\rightarrow 2.0,\ (+,-,+)\rightarrow 5.0\}$
$W_2=\{(d,+)\rightarrow 1.0,\ (d,s)\rightarrow 3.0,\ (+,s)\rightarrow 1.0,\ (s,-)\rightarrow 1.0,\ (s,d)\rightarrow 0.1,\ (-,d)\rightarrow 1.0\}$,
$W_3=\{(d,+,d)\rightarrow 0.1,\ (-,+,-)\rightarrow 2.0,\ (+,-,+)\rightarrow 5.0\}$,
$W_4=\{(s,+)\rightarrow 1.0,\ (s,d)\rightarrow 3.0,\ (+,d)\rightarrow 1.0,\ (d,-)\rightarrow 1.0,\ (d,s)\rightarrow 0.1,\ (-,s)\rightarrow 1.0\}$.

(例2終り)

概整合 CLP を解くというのは，U から L への写像 h であって，(1)全体誤差 ε がある閾値 ε_0 以下のものをすべて求めるか，または，(2)全体誤差 ε が最小のものを求めるか，の2通りの場合がある．概整合 CLP において，誤差の値をすべて 0.0 にし，ε_0 も 0.0 に設定すると，普通の CLP になる．したがって，普

通の CLP は概整合 CLP の特殊な場合である．また，最小誤差の解を求める最適化問題は，動的計画法の問題[1]となる．

［例2］において，全体誤差として単純和を用いることにし，閾値 ε_0 を 1.0 とおくと，求めるべき解は［例1］の解1のみとなり，全体誤差は $\varepsilon=0.1+0.1+0.1+0.1=0.4$ となる．

3.2 解法の概要

CLP は，NP 完全[2),36)]な問題である．その理由は，Cook[2)] によって NP 完全であることが証明された原型問題 (archetypal problem) である連言標準型命題論理式の充足可能性問題が，等価な CLP に多項式還元 (polynomially reducible) できることからも明らかである．

CLP の解法は，大きく4つのアプローチ，すなわち，(1)列挙法 (enumeration method)，(2)木探索 (tree search) 法，(3)弛緩 (relaxation) 法，(4)併合 (merge) 法に分類できる．列挙法は，すべてのラベル組合せを生成しテスト (generate-and-test) する腕ずく (brute force) による方法で，とくに付言することはない．残りの三つについて，以下，簡単な補足を加える．

3.2.1 木探索法

もっとも単純な縦型探索によって，［例1］を処理してゆく過程を，図-5 に示す．図では，単にユニットの識別番号順にラベル割り当てを試みている．割り当て済みのユニット及び新たに割り当てようとするユニットのみを成分とするユニット組があるなら，そのユニット組に対応するラベル拘束条件 ($\in R$) を見ながら新たなユニットへのラベル付けを試みてゆく．つまり，整合できない拘束条件ペアが存在するなら，そのラベル割り当ては中止する．たとえば，図中，＊印のところで新たにユニット2にラベル s を割り当てようとしたが，拘束条件ペア (t_2, R_2)（ここに $t_2=(1,2)$）を調べると (s, s) というラベル組は R_2 に含まれていないので，ラベル s の割り当ては行われない．同様に，＊＊印の位置においてもユニット4へのラベル s の割り当ては行われない．図中，破線は，解1が見つかるまでの探索の軌跡を表している．破線が上向きのところが2カ所あるが，これらがバックトラック (backtrack) の箇所である．

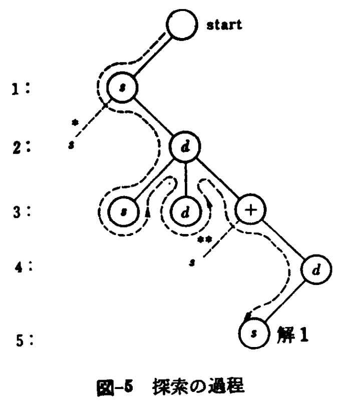

図-5 探索の過程

現実にはこのような単純な探索法では，無駄なバックトラックが頻発 (thrashing) するので効率が悪い．また，すべての解を求める場合は，結局，探索木全体をたどることになってしまう．そこで，無駄な分岐をなるべく避ける方法が望まれる．その一つが，発見的関数 (heuristic function) を導入する方法であり，一般的な諸性質もしだいに明らかにされつつある．しかし，重みのない普通の CLP においては，有効な発見的関数を見いだすことは必ずしも容易なことではない．Haralick ら[13),14)]は，拘束条件 T と R を用いた先読みオペレータ (look-ahead operator) や前方チェック (forward checking) などの分枝刈り込み (pruning) の技法を提案している[19),32)]．さらに，back cheking[11)]，back marking[9)]，back jump[10)] などの刈り込み技法が提案されている．これらの無駄な探索枝を刈り込む技法の効率を解析するのは，かなり困難なことが多い．計算機によるシミュレーション実験による効率評価及び解析結果は，Gaschnig[10)]，Haralick & Elliott[11)] などにみられる．また，Nudel[26)] は上記の方法に加え，種々の発見的手法を組織的かつ計画的に導入し，評価を与えた．

3.2.2 弛緩法

フィルタリング (filtering)，拘束伝播 (constraint propagation)，ネットワーク整合化 (network consistency) などともいう．2次の同次 CLP の例として，命題論理式

$$(A\vee B)\wedge(\bar{B}\vee\bar{C})\wedge(A\vee C)\wedge(\bar{A}\vee B)$$

の充足問題を，CLP に変換したものを取りあげる．これは，上式の4つの選言節に順にユニット1〜4を対応させ，ラベルとしては，命題変数 A〜C 及びその定否 $\bar{A}$〜$\bar{C}$ を用いる．すると，たとえば，$t_1=(1,2)$ について，ユニット1と2を同時に 'true' にする部分ラベリングは，$R_1=\{(A, \bar{B}), (A, \bar{C}), (B, \bar{C})\}$ となる．こうして得られた CLP を等価な '拘束ネットワーク' に変換したものを図-6 に示す．すなわち，頂

点はユニットに対応しており，辺はその両端の頂点（すなわちユニット）間に拘束条件が存在することを表す．したがって，辺には，拘束条件ペア，つまりユニット組 t_i とラベル拘束条件 R_i が付いている．また，各頂点には初期のラベル集合 L (3.2 参照) が付随している．

Mackworth[19] や McGregor[22] らは，拘束ネットワーク表現した CLP に対して弛緩法を導入した．これは，辺整合や路整合という概念を用いて，局所的な拘束条件を周辺に伝播させてゆき，全体として整合のとれたネットワークに縮約する方法である．図-6 の辺 (1, 2) に付随する拘束条件ペア (t_1, R_1) の R_1 に注目するとユニット 1 に割り当てられる可能性のあるラベルは A または B のみであることが分かる．つまり，頂点 1 に付けられた 6 つの候補ラベルのうち×印をつけた 4 つ，すなわち，$C, \bar{A}, \bar{B}, \bar{C}$ は割り当てられることはありえない．このような無効なラベルを除去する操作を拘束伝播という．このような弛緩処理はネットワーク全体が弛緩状態に収束するまで繰り返される．

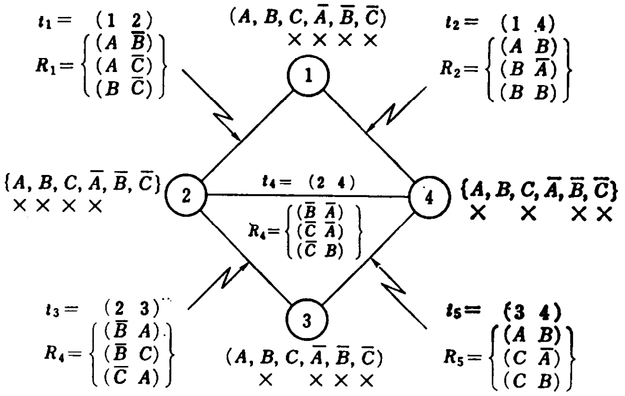

図-6 拘束ネットワーク

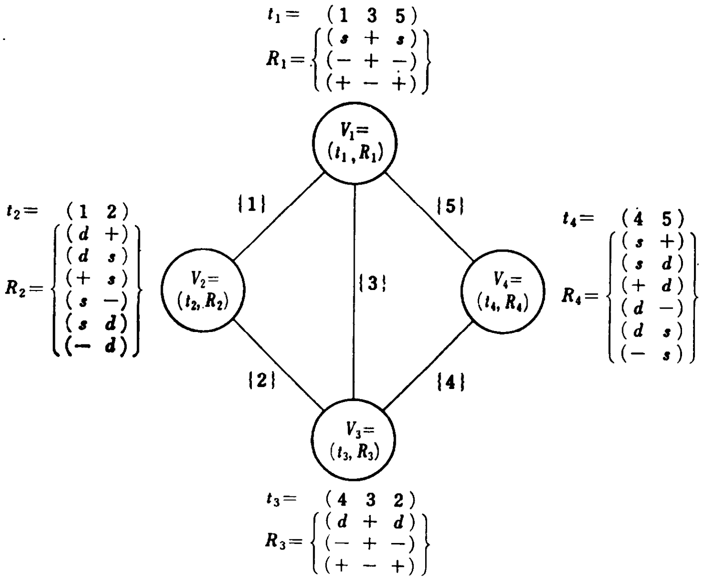

図-7 頂点拘束ネットワーク（3 項以上の関係を含む）

一般に，弛緩法の目的は，最終解を与えることではなく，無効ラベルをできるだけ取り除くことによって探索空間をなるべく小さくしておき，後続の組合せ探索を効率よく行えるようにすることである．

弛緩法に関して，拘束伝播操作の適用順序や実現方法などについて種々の改良が提案されている[3),23),34),39)]．また，2 次の同次 CLP だけでなく，一般の CLP もネットワークで等価表現（次節 **3.2.3** 参照）し，それに弛緩法を適用することもできる．なお，弛緩法は，並列処理アルゴリズム化が容易という性質があるが，それが効率改善に結びつくかどうかは今後の検討課題である．

3.2.3 併 合 法

Freuder[7] の統合法 (synthesizing) や動的計画法の考え方を応用した Seidel[29] の侵入法 (invasion) などが，ここに分類される．

まず，一般の CLP を等価的に表現する拘束ネットワークを導入する[40]．ここでは前節のものと区別するため，‘頂点拘束ネットワーク’と呼ぶ．CLP すなわち (U, L, T, R) に等価な頂点拘束ネットワーク G は次のような無向グラフである．

$$
\begin{aligned}
V(G) &= \{(t_i, R_i) \mid t_i \in T, R_i \in R\}, \\
E(G) &= \{(v_i, v_j) \mid s(t_i) \cap s(t_j) \neq \phi, \\
&\qquad v_i = (t_i, R_i),\ v_j = (t_j, R_j), \\
&\qquad v_i, v_j \in V(G)\}
\end{aligned}
$$

ただし，$s(t)$ は，$t = (u_1, \ldots, u_k)$ とすると，$s(t) = \{u_1, \ldots, u_k\}$．前節の拘束ネットワーク（図-6）では，頂点

$$t_{12} = (1\ 2\ 3\ 5)$$

$$R_{12} = \left\{ \begin{array}{l} (s\ -\ +\ s) \\ (s\ d\ +\ s) \\ (-\ d\ +\ -) \\ (+\ s\ -\ +) \end{array} \right\}$$

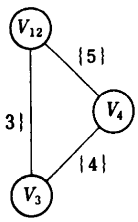

図-8 頂点 V_1 と V_2 (図-7) の併合後

と辺がそれぞれユニットと拘束条件ペアに対応しているのに対して，頂点拘束ネットワークでは，頂点と辺は，ちょうど，逆の役割を担っている．すなわち，頂点には拘束条件ペアが付随する．これにより，図-6 の場合とは異なり，２次元以外の拘束関係も表現できる．

この定義によって［例１］の CLP を等価表現したのが，**図-7** の頂点拘束ネットワークである．ここで，頂点 V_1 と V_2 を，共通ユニット１に関して整合性を保ちつつ単一の頂点（これを V_{12} で表す）にまとめると，**図-8** のような結果が得られる．このように二つの頂点を一つにまとめる操作を‘併合’という．併合操作を繰り返してゆくと，最終的には頂点１個だけのネットワークに縮退するが，そのとき得られる拘束条件ペアが最終解を与える．

併合法においては，併合操作の順番を工夫して，処理途中における中間解の個数がなるべく組合せ爆発を起こさないようにすることが重要である[29),44)]．

3.3 高速化と並列処理

すでに述べたように，CLP は NP 完全であるから，（クラス P≠クラス NP であるかぎり）一般の CLP を効率よく解く汎用アルゴリズムは存在しない．したがって，上に述べたように，現実には，個々に与えられた CLP ごとに，固有の拘束条件の構造に基づいた解法を開発し，適用してゆくほかない．高速化の手段はたかだか数種類の方法に大別することができる．すなわち，まず，アルゴリズムの改良に関して言えば，（a）木探索における分岐枝の刈り込みを行って，探索空間を縮小する方法，（b）問題を複数個の部分問題に分割して，組合せの総数を大幅に減らす方法（分割統治法 (divide-and-conquer)），（c）発見的関数 (heuristic function) などの評価関数を用いて，解の存在の可能性の高い方向へ探索を進めてゆく方法（分枝限定法 (branch-and-bound)）などがある．また，人が介入して，探索の開始節や探索方向を指示できるような対話システム (man-machine system) として実現するという接近法も考えられる．さらに，最近，複数の処理装置を用いた並列処理アルゴリズムの研究も盛んである[20),27),33),36)]．また，CLP の処理効率に関する研究も数多いが，紙面の都合で文献を紹介するにとどめる[4),6),16),21),25),29)~31)]．

4. 応　用

CLP は，非常に広い分野の問題を表現できるが，特に，図形や画像の解釈などに代表される人工知能及びその関連分野，さらに，グラフのアルゴリズムなどに代表される組合せ的問題において，多くの応用が見いだされる．ここでは，概観的な説明は省き，なるべく斬新な応用例について列挙する．

4.1 パターン解析への応用

線画解釈についてはすでに述べたが[17),34)]，これらの多くは，拘束伝播あるいはフィルタリングと呼ばれる解法を用いている．シーン解析においては，確率的弛緩法を用いた研究[17),28)]がある．また，並列弛緩法によるシーン・ラベリングの研究[16)]，衛星画像から河川や山脈の抽出処理への応用[15)]などがある．三面図の面図間に成り立つ拘束条件を用いて解釈し，もとの三次元モデルを復元する方法も報告されている[12),42)]．CLP を用いて，地図などの視覚的情報を構造化した知識として表現し，スケッチマップの認識や理解を組織的に行うことを目指す試みがある[24)]．また，文字などの構造的パターン認識への応用もある[46)]．

4.2 組合せ問題への応用

Ullmann[32)]は，同型部分グラフ問題を，木探索の方法で解くアルゴリズムを与えた．同様に，グラフの準同型問題，クリーク (clique) 探索問題，有向グラフの準同型及び同型問題などにも応用できる[8)]．他に，既述の論理式の充足問題，地図ぬりわけなども，この範疇に属する．

4.3 その他の関連問題

パズルやゲームの分野では，N クイーン問題，ラテン方陣，クロスワード・パズル，急性発狂 (instant insanity) パズルなど多くの問題が CLP で定式化できる．越村ら[37)]は，LK に基づく１階述語論理の証明手続きを，証明の半順序性に着目して並列に処理する方法を提案した．

大和田ら[43)]は，この拘束伝播操作を論理型言語において実現する方法を考察した．これは，本来，非決定的な性質をもつ論理型言語を，なるべく逐次的に効率

よく実行したいからである．これが実現されると，CLP を論理プログラムで表現できることになり，興味深い．またこれと関連して，プログラミング方法論において，拘束（制約）によって知識あるいは処理の意図を記述するという，制約論理プログラミング（constraint logic programming，これも略記すると CLP となる）の研究も行われつつある[18),35)]．これについては，本稿の主題と深い関連があるが，適切な解説[47)]があるので詳細は省く．

丸山ら[38)]は，日本語処理における係り受け解析に内在する組合せ的な曖昧さを拘束条件として表現し，拘束伝播アルゴリズムを適用した．また，Feldman ら[5)]は，カリキュラム編成への興味ある応用を報告している．今後，広く設計や計画問題への応用も広がるものと思われる[45)]．

5. むすび

CLP は，問題の対象をまず複数個の構成要素に分解し，さらに，構成要素間に成り立つべき拘束（制約）関係を明らかにし，それを手がかりにして対象全体の解析や解釈を行うというアプローチをとる．このような考え方は，連立方程式や数理計画法など工学の多くのモデル化手法として伝統的に行われてきたものである．これは，いわば，デカルトの構成論的方法*，つまり‘明証→分析→総合→枚挙’の過程の提案以来の伝統的な方法といえる．しかし，‘総合’すなわち CLP における整合化の処理自体が，現実には膨大な時間を要するという点が実は問題である．換言すると，少なくとも CLP に構成論的方法を適用するときは，処理の手間すなわち時間要素を無視できない．

ところで，CLP の拘束条件は，広い意味で宣言的知識といえる．実際，図-9 に例示するように，論理型プログラミング言語で CLP を表現することは容易で

(U, L, T, R)

$U=\{1, 2, 3\}$,
$L=\{A, B, C, D\}$,
$T=\{(1\ \ 2), (2\ \ 3), (1\ \ 3)\}$,
$R=\{R_1, R_2, R_3\}$,
$R_1=\{(A\ \ B), (D\ \ A)\}$,
$R_2=\{(C\ \ D), (A\ \ B), (B\ \ D)\}$,
$R_3=\{(A\ \ D), (D\ \ B), (C\ \ B)\}$

（a） CLP

$P_{123}(x, y, z) <- P_{12}(x, y),$
$\quad P_{23}(y, z), P_{13}(x, z).$
$P_{12}(A, B).$
$P_{12}(D, A).$
$P_{23}(C, D).$
$P_{23}(A, B).$
$P_{23}(B, D).$
$P_{13}(A, D).$
$P_{13}(D, B).$
$P_{13}(C, B).$
$?-P_{123}(x, y, z)$

（b） Prolog

図-9　CLP の例及び等価な Prolog プログラム

*「方法序説」落合太郎訳，岩波文庫，pp. 22-30 (1953).

ある．CLP における主要な興味は，構成論的方法論やプログラミング言語における記述容易性や表現論ではなく，時間的側面つまりいかに迅速に処理するかということである．人工知能の諸分野において，組合せ的探索に帰着される問題は非常に多い．しかも，これらの多くが NP 完全の範疇に入るということを考えあわせると，実用的な知識システムを企図するとき，少々，深刻にならざるをえない．計算の手間に関していえば，並列化による処理速度の向上も，原理的には組合せオーダの解消には寄与しない．今後，この計算量（computational complexity）の壁に対して，発見的手法の開発及び整備がますます重要性を増してゆくと思われる．

CLP におけるもう一つの問題点は，拘束条件の表現方法である．ユニット拘束関係とラベル拘束関係のように陽に表現するのが不可能な場合や，あるいは，ラベル拘束関係が膨大になってしまい，記憶量や手間の点で非常に不経済になる場合も多い．探索技法と関連づけながら，この問題を考察することも今後の課題である．

また，すでに述べたように，制約論理プログラミングとの関連に関する検討や，拘束条件の表現能力と限界に関する考察も今後の課題であろう．

最後に，広い応用と拡がりをもつ CLP を，限られた紙面で十分紹介できなかったことを惧れる．

参考文献

1) Bertelè, U. and Brioschi, F : Nonserial Dynamic Programming, p. 235, Academic Press, N. Y. (1972).
2) Cook, S. A.: 3rd Annual ACM Symp. on Theory of Computing, pp. 151-158 (1971).
3) Davis, E.: Artif. Intell., Vol. 32, pp. 281-331 (1987).
4) Dechter, R. and Pearl, J.: Artif. Intell., Vol. 34, No. 1, pp. 1-38 (1988).
5) Feldman, R. and Golumbic, M. C.: Interactive Scheduling as a Constraint Satisfiability Problem, to appear (1989).
6) Fernández-Baca, D.: Inf. Proc. Lett., Vol. 27, No. 6, pp. 323-326 (1988).
7) Freuder, E. C.: C. ACM, Vol. 21, No. 11, pp. 958-966 (1978).
8) Galil, Z.: Comput. Surveys, Vol. 18, No. 1, pp. 23-38 (1986).
9) Gaschnig, J.: Proc. Int'l. Conf. on Artif. Intell., p. 457, Cambridge, MA (1977).
10) Gaschnig, J.: 2nd Nat'l Canadian Conf. for

Computational Studies of Intell., pp. 268-277, Toronto, Canada (1978).
11) Haralick, R. M. and Elliot, G. L.: Artif. Intell., Vol. 14, pp. 263-313 (1980).
12) Haralick, R. M. and Queeney, D.: Comput. Graphics & Im. Proc., Vol. 20, pp. 244-258 (1982).
13) Haralick, R. M. and Shapiro, L. G.: IEEE Tr. Patt. Anal. Mach. Intell., Vol. PAMI-1, No. 2, pp. 173-184 (1979).
14) Haralick, R. M. and Shapiro, L. G.: IEEE Tr. Patt. Anal. Mach. Intell., Vol. PAMI-2, No. 3, pp. 193-203 (1980).
15) Haralick. R. M., Wang, S. and Elliot, D. B.: Proc. ICPR, 6th, Munich, pp. 502-516 (1982).
16) Henderson, T. C. and Samal, A.: Proc. ICPR, 9th, Rome, pp. 220-222 (1988).
17) Huffman, D. A.: Machine Intell., Vol. 6 (ed, Meltzer, B. & Michie, D.) pp. 295-323 (1971).
18) Leler, W.: Constraint Programming Language—Their Specification and Generation, Addison-Wesley (1988).
19) Macification and Generation, Addison-Wesley pp. 99-118 (1977).
20) Mackworth, A. K.: Encyclopedia of Artificial Intelligence (ed. Shapiro, S. C.), pp. 205-211, Wiley (1987).
21) McCall, J. T. et al.: IEEE Tr. Computers, Vol. C-34, No. 11, pp. 973-980 (1985).
22) McGregor, J. J.: Inform. Sciences, Vol. 19, pp. 229-250 (1979).
23) Montanari, U.: Inform. Sciences, Vol. 7, No. 2, pp. 95-132 (1974).
24) Mulder, J. A., Mackworth, A. K. and Havens, W.S.: IEEE Tr. Patt. Anal. Mach. Intell., Vol. PAMI-10, No. 6, pp. 866-879 (1988).
25) Nishihara, S. and Ikeda, K.: Proc. ICPR, 8th, Paris, pp. 198-200 (1986).
26) Nudel, B.: Artif. Intell., Vol. 21, pp. 135-178 (1983).
27) Pearl, J.: Heuristics, p. 382, Addison-Wesley, Reading, Mass. (1984).
28) Rosenfeld, A., Hummel, R. A. and Zucker, S. W.: IEEE Tr. Syst., Man & Cybernetics, Vol. SMC-6, No. 6, pp. 420-433 (1976).
29) Seidel, R.: Proc. IJCAI, 7th, pp. 338-342 (1981).
30) Shapiro, L. G.: Proc. ICPR, 7th, Montreal, pp. 313-315 (1984).
31) Shapiro, L. G. and Haralick, R. M.: IEEE Tr. Patt. Anal. Mach. Intell., Vol. PAMI-3, No. 5, pp. 504-519 (1981).
32) Ullmann, J. R.: J. ACM, Vol. 23, No. 1, pp. 31-42 (1976).
33) Wah, B. W., Li, G.-J. and Yu, C. F.: IEEE Computer, Vol. 18, No. 6, pp. 93-108 (1985).
34) Waltz, D.: Understanding Line Drawings of Scenes with Shadows, in The Psychology of Computer Vision (ed. Winston, P. H.), pp. 19-91, McGraw-Hill (1975).
35) 原田，溝口：日本ソフトウェア科学会大会，第5回，C 1-2, pp. 53-56 (1988).
36) 茨木：組合せ最適化一分枝限定法を中心として，p. 239, 産業図書 (1983).
37) 越村，瀧：日本ソフトウェア科学会大会，第5回，A 5-2, pp. 205-208 (1988).
38) 丸山，渡辺：日本ソフトウェア科学会大会，第5回，A 1-1, pp. 17-20 (1988).
39) 西原，原，池田：情報処理学会論文誌，Vol. 26, No. 1, pp. 1-8 (1985).
40) 西原，原，池田：信学論(D)，Vol. J 67-D, No. 7, pp. 745-752 (1984).
41) 西原，松尾，池田：人工知能学会誌，Vol. 3, No. 2, pp. 196-205 (1988).
42) 西原，渡辺，池田：情報処理学会論文誌，Vol. 28, No. 5, pp. 534-537 (1987).
43) 大和田，溝口：日本ソフトウェア科学会大会，第5回，C 1-3, pp. 57-60 (1988).
44) 塩澤，西原，池田：情報処理学会論文誌，Vol. 27, No. 10, pp. 927-935 (1986).
45) 杉本：信学論 (D-I), Vol. J 72-D-I, No. 5, pp. 335-342 (1989).
46) 山本：信学論 (D), Vol. J 65-D, No. 9, pp. 1167-1175 (1982).
47) 横井，相場：情報処理，Vol. 30, No. 1, pp. 29-38 (1989).

(平成元年8月2日受付)