拘束条件の構造を考慮した整合ラベリング問題の解法†

塩澤恒道†† 西原清一††† 池田克夫†††

整合ラベリング問題は，複数個の構成要素から成る対象を解釈するのに，まず，各構成要素に対して局所的解釈の候補を求め，それらの中から対象物全体の矛盾のない解釈を求める問題である．このような問題は，画像処理や人工知能など多くの分野に見いだされる．この問題に対しては，従来より，大別して，バックトラッキングを用いた木探索による解法，弛緩操作や拘束伝播を用いたフィルタリングによる前処理を含む解法の二つが示されている．本稿で示す解法は，これらの方法とは異なり，動的計画法の手法に基づくものである．すなわち，まず，拘束条件の構造に注目して，与えられた問題をより小さなサイズの問題に分解する．そして，それらより得られる複数個の部分解を矛盾なく結合することによって全体の解を得ようというものである．ここでは，分解された小問題のサイズを表す front 指数を導入し，与えられた個々の問題に固有の front 指数をもとに計算量を評価した．さらに，与えられた問題の分解と front 指数に関する幾つかの性質について述べ，front 指数が下限をとるような最適分解を求めるアルゴリズムを与えた．

1. まえがき

複数個の構成要素から成る対象を解釈したり，解析したりする問題では，まず各要素に関する局所的解釈の候補を求め，それらの中から対象物全体の矛盾のない解釈の組合せを求める解法が考えられる．このような問題は画像処理や人工知能の分野における画像のラベリングや線画理解などの問題をはじめ，Nクイーン問題などのパズル，さらに同型部分グラフの探索など極めて広い分野に見られる．これらは，整合ラベリング問題 (consistent labeling problem[1], CLP[2]), 拘束充足問題 (constraint satisfaction problem[4]) など，様々な名前で呼ばれているが，本稿では CLP と略記する．

CLP は，一般に NP-完全な探索問題の一つであり[2]*，その解法としては，バックトラッキングを基本とした深さ優先木探索による方法と，弛緩操作や拘束伝播を用いたフィルタリングによる前処理を含む方法の二つが従来行われてきた．これらに対して本稿では，動的計画法 (DP) の手法を援用した解法について述べる．これは，まず前半で，構成要素の組に対して可能な解を局所的に求める処理，すなわち，対象全体を部分問題に分解しそれらの局所解を求め保存するという処理を繰り返し，後半において，それらの局所解を，生成した順番と逆にたどることによって最終解を得る方法である．これにより，CLP の総処理コストは，各部分問題の処理に要する計算量の和で与えられることになる．

以下では，まず第2章で，CLP およびその等価表現である拘束ネットワークの定義を与える．第3章では，上述した DP の手法に基づいた方法を，解グラフの生成アルゴリズムを中心に述べる．第4章では，前章で示した解法の諸性質を明らかにし，処理効率向上のための最適化手法を提案する．

2. 問題の定義とその等価表現

2.1 CLP の定義

CLP は，四つ組 (U, L, T, R) で与えられる．$U=\{1, \cdots, M\}$ はユニット集合で，各要素 ('ユニット') は対象の構成要素に対応する．Lはラベル集合で，各要素 ('ラベル') は，ユニットに与えられる解析や解釈の候補を表す．T 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．それぞれのユニット組に対して可能な局所的解釈が $R=\{R_1, \cdots, R_{|T|}\}$ で与えられる．$||$ は集合のサイズを表す．各 R_i は 'ラベル拘束関係' と呼ばれる．ここに，$t_i(\in T)$ は，その成分 (ユニット) 同士がある拘束条件のもとに関連し合っていることを表し，その具体的なラベル組の候補集合は $R_i(\in R)$ によって与えられる．以後，これらの組 (t_i, R_i) を '拘束条件ペア' と呼ぶ．また，とくに添字を明示する必要のない場合は，(t, R_t) と表すこともある．

[例 1] CLP の例

† A Consistent Labeling Algorithm Taking into Account the Structure of Constraints by TSUNEMICHI SHIOZAWA (Department of Scientific Technology, University of Tsukuba), SEIICHI NISHIHARA and KATSUO IKEDA (Institute of Information Sciences and Electronics, University of Tsukuba).

†† 筑波大学大学院理工学研究科

††† 筑波大学電子・情報工学系

* NP-困難であることは，グラフの彩色問題 (k-colorability problem)[3] が，CLP で記述できることから明らか．

$U=\{1,\cdots,9\}$, $L=\{a,\cdots,j\}$,
$T=\{t_1,\cdots,t_7\}$,
$t_1=(1,2,4)$, $t_2=(1,2,3)$, $t_3=(2,3,4,5)$,
$t_4=(5,8)$, $t_5=(4,6,7)$, $t_6=(6,7,8)$,
$t_7=(7,8,9)$,
$R=\{R_1,\cdots,R_7\}$,
$R_1=\{(a,b,c),\ (b,a,c),\ (b,a,e)\}$,
$R_2=\{(b,a,b),\ (a,b,c),\ (a,b,a)\}$,
$R_3=\{(a,b,c,d),\ (a,b,e,d),\ (b,c,c,d)\}$,
$R_4=\{(d,h),\ (e,i)\}$,
$R_5=\{(c,d,f),\ (d,c,e),\ (c,f,g)\}$,
$R_6=\{(f,g,h),\ (d,f,h),\ (g,h,i)\}$,
$R_7=\{(f,h,j),\ (g,h,h)\}$.

CLP を解くとは，全ユニット $(1,\cdots,M)$ に対するラベル組 $\lambda=(l_1,\cdots,l_M)$ のうち，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付け出すことである．ただし，$\lambda(t)$ は，λ の t に関する射影，すなわち，λ のうち t の要素ユニットに対応するラベルのみを取り出したものである．上記の例の解の一つを次に示す．

(1 2 3 4 5 6 7 8 9)
(b a b c d d f h j).

この解は明らかに上記の T, R で与えられる拘束条件を満足している．

2.2 拘束ネットワークによる等価表現

ここで，CLP を等価表現する '拘束ネットワーク' を定義する．拘束ネットワーク G は四つ組 (U, L, T, R) を用いて次のように与えられる無向グラフである．ただし，$V(G)$, $E(G)$ は，G の頂点および辺の集合を表す．とくにグラフ G が自明のときは，単に，V, E と書くこともある．

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

ただし，$t=(u_1,\cdots,u_l)$ とすると，$s(t)=\{u_1,\cdots,u_l\}$.

本定義では，頂点は拘束条件ペアを表し，辺は両端の頂点のいずれのユニット組にも含まれる共通ユニットの存在を示している．一方，従来の拘束ネットワーク[4]は，これと異なり，各頂点は一つのユニットを表し，辺は二項（拘束）関係の存在を示していた．本稿の定義は，複数個の一般の多項関係を自然に表現するのに適している．頂点と拘束条件ペアとは1対1に対応しているので，以後，特に区別せずに用いることがある．

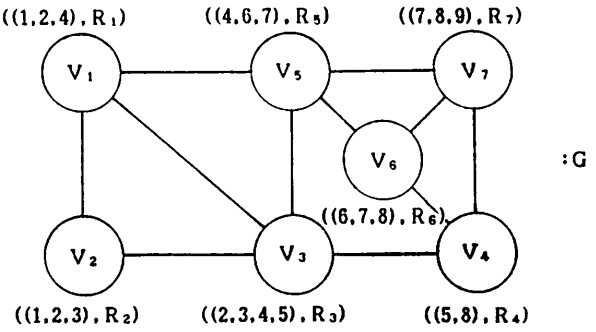


図 1 拘束ネットワーク G の例 ($v_i=(t_i,R_i)$)
Fig. 1 Example of a constraint network, G.

図1は [例 1] の CLP を等価表現する拘束ネットワークを示す．

2.3 CLP の冗長性

図1において，ユニット9は頂点 v_7 のユニット組にのみ現れている．このようなユニットは，他の頂点からの影響を，当該ユニット組の他のユニット（ここでは7と8）を介してのみ間接的に受けていると言える．すなわち，ユニット7，8に対するラベルが決まれば，ユニット9のラベルも決定される．このように，たかだか一つのユニット拘束関係のみに含まれるユニットを '冗長ユニット' という．この例で，ユニット7，8，9に可能なラベル組の集合（ラベル拘束関係）を記憶しておけば，冗長ユニットは CLP から除去することができる．この結果，[例 1] の CLP は次のようになる．

[例 2] 冗長ユニット9を除去した CLP
$U=\{1,\cdots,8\}$, $L=\{a,\cdots,j\}$,
$T=\{t_1,\cdots,t_7\}$,
$t_1,\cdots,t_6$ は [例 1] に同じ，$t_7=(7,8)$,
$R=\{R_1,\cdots,R_7\}$, $R_1,\cdots,R_6$ は [例 1] に同じ，
$R_7=\{(f,h),\ (g,h)\}$.

このように，冗長ユニットは，CLP をより簡単な問題に縮退させるのに役立つ．ところで，図1において，ユニット1は頂点 v_1, v_2 にのみ現れている．もし，頂点 v_1, v_2 を，局所的に整合をとり一つの頂点にまとめると，ユニット1も冗長となるであろう．本稿の方法はこのように，複数の頂点を整合処理し，一つの頂点に併合して冗長ユニットを発生させ，徐々に与 CLP のサイズを小さくしていくものである．

3. 拘束ネットワークの処理と invasion

3.1 頂点併合操作

拘束ネットワーク上で CLP を解く方法として，二

つの頂点を一つの頂点で置き換える操作を繰り返し，漸次，頂点の個数を減少させてゆく方法が従来よりある[5]．ここでは，この（頂点）併合操作を，一般の複数個の頂点の場合について定義する．拘束ネットワーク G 内の頂点 $v_i, \cdots, v_j$ の '併合操作 (join)' は，次のようなネットワーク G' に縮退させる処理である．$W=\{v_i, \cdots, v_j\}$ とする．

$$V(G')=(V(G)-W)\cup\{v_{i\cdots j}\},$$

$$E(G')=(E(G)-\{(v_k, v)(\in E(G))\mid v_k\in W\})\cup\{(v_{i\cdots j}, v)\mid \exists v_k(\in W)[(v_k, v)\in E(G) \text{ かつ } v\notin W]$$

ここに，新しい頂点である $v_{i\cdots j}$ は，ユニット集合 $\cup v_k\in W(t_k)$ からなるユニット組 $t_{i\cdots j}$ と，$v_k\equiv(t_k, R_k)$ $(v_k\in W)$ を満足するようなラベル組 λ（すなわち，$\forall v_k(\in W)$ $[\lambda(t_k)\in R_k]$ を満たす λ，2.1 節参照）の集合 $R_{i\cdots j}$ とから成る拘束条件ペア $(t_{i\cdots j}, R_{i\cdots j})$ を表す．このように，併合操作は，二つ以上の頂点から成る部分ネットワークを，CLP の部分問題として解き，一つの頂点で置き換える操作である．

図1において二頂点 v_1 と v_2 に併合操作を施すと**図2**が得られる．前節で述べたように，ユニット1が新たに冗長ユニットとなるので除くことができる．これは，換言すれば，頂点 v_1 と v_2 はユニット1に関する局所的問題を形成しているといえる．

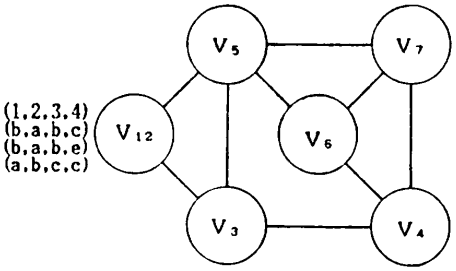


図 2　v_1 と v_2 の節併合後の拘束ネットワーク
Fig. 2　The network after performing join on v_1 and v_2 of G.

次に，図2において，頂点 v_{12} と v_3 を併合した場合(**図3**(a))と，頂点 v_{12} と v_5 を併合した場合（同図(b)）とを比較してみよう．前者 v_{123} では，ユニット2，3が冗長ユニットとなっているが，後者 v_{125} では冗長ユニットは発生しない．その結果，前者では非冗長ユニットは2個（4と5），後者では5個（2, 3, 4, 6および7のすべて）となる．これらの非冗長ユニットは残りの他の頂点と直接影響し合っており，後刻の併合操作に関与することになる．

ユニット u について次のようなユニット組集合 $\tau(u)$ を定義しよう．

$$\tau(u)=\{t\in T\mid u\in s(t)\}$$

$\tau(u)$ の要素を第一成分とする拘束条件ペアの全体から得られる部分拘束ネットワークは，'ユニット u に関する部分問題' を表している．したがって，u が冗長になったということは，この部分問題の併合処理が完了したことを意味する．さらに，併合後の頂点（ただ一つ）において冗長ユニットを除いた残り，すなわち非冗長ユニットの個数が少ないほど，上記の（冗長ユニットに関する）部分問題の局所性が高かったことを示している．とくに，非冗長ユニットがゼロ個となるのは，拘束ネットワークが複数の連結成分からなる場合である．

このような一連の性質は，非直列的 (non-serial) DP[6] において，局所的処理の適用順序を最適化する問題と類似している．本稿の方法は，CLP と等価な拘束ネットワークの構造を解析し，そこに含まれる局所性の高い部分問題を探し，優先的に併合操作を施すという方法である．

3.2 invasion と front 指数

前節で，併合操作の適用順序が，冗長ユニットの発生順序を決定し処理効率に影響を与えることを示した．ここでは，この併合操作の適用順序に関する諸定義を与える[4),7)]．

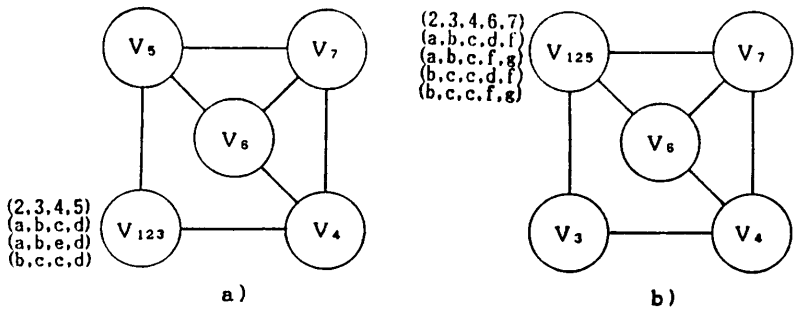


図 3　2回目の節併合後の拘束ネットワーク
Fig. 3　Two example networks after performing join operations on a) v_{12} and v_3, and b) v_{12} and v_5.

拘束ネットワーク G の 'invasion' とは，列 $\{G_i\}, i=1, \cdots, |T|$ である．ただし，G_i は，$V(G)$ 内の i 個の頂点からなる G の誘導部分グラフ (induced subgraph)[8] であり，かつ $V(G_i)\subseteq V(G_{i+1})$ とする．G_0 は空グラフ．$|V(G)|=|T|$ であるから，$G_{|T|}=G$．また，$V(G_i)-V(G_{i-1})$ なる頂点を，

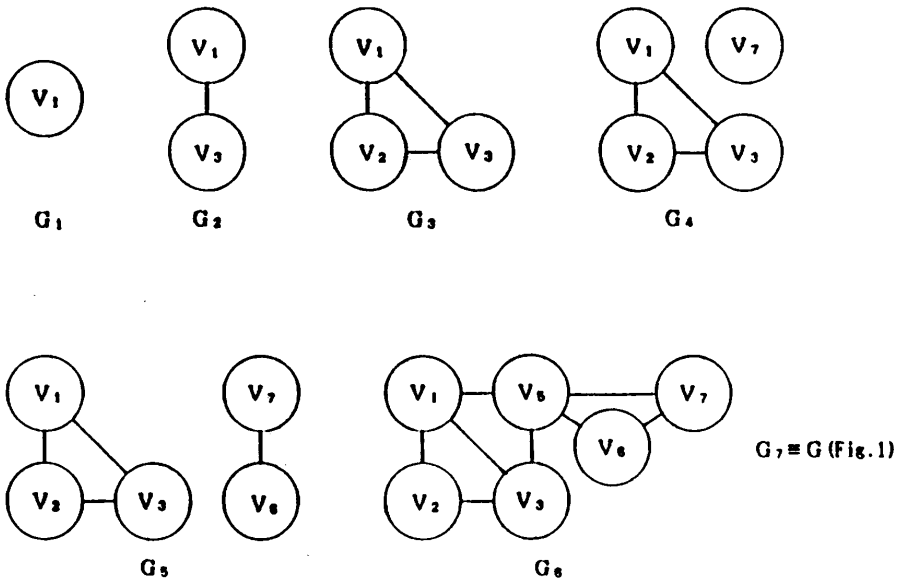


図 4 図1の拘束ネットワークの invasion の例
Fig. 4 Example of the (optimal) invasion for constraint network G of Fig. 1.

G_i の '固有頂点' といい，$v_{(i)}$ で表す．

[補題 1] 拘束条件ペアの重複のない順列 $\{v_i\}, i=1,\cdots,|T|$ は，invasion を一意に決定する．また，逆も正しい．

(略証) $G_i=\langle \bigcup_{j=1}^{i}\{v_j\}\rangle_G$ とすればよい．[ただし，$\langle S\rangle_G$ は G の誘導部分グラフ[8]を表す．すなわち，$S\subseteq V(G)$ のとき，$\langle S\rangle_G$ は，頂点集合 S，辺集合が $\{(v,w)|(v,w)\in E(G),\ v,w\in S\}$ なる部分グラフである．] □

図4は図1の拘束ネットワーク G の invasion の一例である．

一般に invasion の項 G_i は，一個以上の連結成分からなっており，そのうちの一つは固有頂点を含むことに注意．項 G_i の連結成分を $G_i^{(1)},\cdots,G_i^{(c)}$ とする．各連結成分 $G_i^{(j)}(1\leqq j\leqq c)$ について，

$$f(G_i^{(j)})\triangleq\Big(\bigcup_{v(=(t,R_t))\in V(G_i^{(j)})} s(t)\Big)\cap\Big(\bigcup_{v(=(t,R_t))\in V(G)-V(G_i^{(j)})} s(t)\Big)$$

を G_i の 'front' という．すなわち，G_i の front とは，ある連結成分に現れる全ユニットのうち，G_i より後の固有頂点，つまり $v_{(i+1)},\cdots,v_{(|T|)}$ のいずれかに再び現れるものの集合である．したがって，G_i の front は，その連結成分の個数 (c 個) あり，それらは互いに素である．front は前節の非冗長ユニット集合に対応しており，そのサイズが小さいほどその連結成分が部分問題としての局所性が高いことを意味する．

図4の invasion の項 G_4 を例にとると，連結成分は二個あり，それぞれの front は {4, 5} と {7, 8} となる．

ここで，G_i の固有頂点を $v_{(i)}=(t_{(i)}, R_{(i)})$ とし，それを含む連結成分を $G_i^{(p)}$ としよう．このとき，$s(t_{(i)})-f(G_i^{(p)})$ を，G_i の '冗長ユニット集合' といい，$r(G_i)$ で表す．図4の G_3 の例では，$v_{(3)}=v_2$ であり，$r(G_3)=\{1, 2, 3\}$ となる．$r(G_i)$ が空でないとき，項 G_i を '併合点' という．各 G_i の 'front 指数' ϕ_i は，G_i の連結成分を $G_i^{(1)},\cdots,G_i^{(c)}$ とすると，

$$\phi_i=\begin{cases}\max\limits_{1\leqq j\leqq c}|f(G_i^{(j)})| & : r(G_i)\neq\phi \text{ のとき},\\ 0 & : r(G_i)=\phi \text{ のとき},\end{cases}$$

で定義する．すなわち，G_i が併合点のとき，各連結成分の front のうち最大サイズを表す．また，ϕ_i $(1\leqq i\leqq|T|)$ のうち最大値 Φ を，その invasion の front 指数という．

次節で述べるように，併合点においては，冗長ユニットが生じるので，固有頂点を含む連結成分に併合操作を施し，一つの頂点に縮退させる処理が行われる．

図4の例では，併合点は G_3, G_4, G_6, G_7 の四つで，それぞれ front 指数などはつぎのようになる．

	G_3	G_4	G_6	G_7
固有頂点	v_2	v_7	v_5	v_4
冗長ユニット集合 $r(G_i)$	{1, 2, 3}	{9}	{4, 6, 7}	{5, 8}

front	{4,5}	{4,5}, {7,8}	{5,8}	ϕ
front 指数 ϕ_i	2	2	2	0

3.3 処理方法

3.3.1 解グラフの生成（前進過程）

拘束ネットワークを G とし，その invasion $\{G_i\}$, $i=1,\cdots,|T|$ が与えられているとする．以下では，この invasion の併合点において併合操作を施しながら，解グラフを生成していく方法について述べる．invasion を各併合点の直後で分断すると，一般に複数個の部分列が得られる．本方法は，そのそれぞれの部分列を独立性の高い部分問題とみなして局所的に解き，中間結果を解グラフに格納・保存してゆく．

一般に，併合点 G_i において，初めのネットワーク G は既に併合操作によりある程度縮退している．ここでさらに，G_i の固有頂点 $v_{(i)}$ とその隣接頂点に（頂点）併合操作（3.1 節）を施し，その結果，$v_{(i)}$ を含んでいた（G_i の）連結成分 $G_i^{(p)}$ は一個の頂点に縮退する．この頂点は，上述した部分問題の解を表す．なお，この頂点に現れる全ユニットは，冗長ユニット集合 $r(G_i)$ と front $f(G_i^{(p)})$ とに分類できるが，併合後，$r(G_i)$ およびその部分ラベリングはネットワークから削除する．また，front は，後刻の部分問題と関連しあっているので，削除できない．そのかわり，解グラフには，新たに，ノード（front $f(G_i^{(p)})$ の情報を持つ）とアーク（冗長ユニット集合 $r(G_i)$ の情報を持つ）* を追加し，部分解を記憶させておく．

解グラフ生成のアルゴリズムの概要を**図 5** に示す．まず，ノード S と E だけの解グラフから出発し，順次，併合点（merging-point）において，上記のように部分問題の中間解，すなわち，$r(G_i)$ と $f(G_i^{(p)})$ のラベリングを求めながら，S から E に向けて成長させていく．ただし，アークの向きは成長の方向とは逆になっており，その終点は同一連結成分 $G_i^{(p)}$ に関してこれまでつくられていたノードである．そのようなノードがないときは，ノード S を終点とする．

なお，ある併合点において，部分解が存在しないときは，最終解も存在しないことは明らかなので，直ちに処理を打ち切る．

図 4 の invasion により得られる解グラフを**図 6** に示す．図中，$G_i\langle[\cdots],(\cdots)\rangle$ は，併合点 G_i において生成されるアークとノードの表すユニット組を示す．とくに，G_6 では，二つの連結成分が一つの頂点に縮退するので，ノード (d, h) からは 2 系統（2 本／系統）のアークが出ていることに注意．

* 拘束ネットワークの用語（頂点，辺）との混乱を避けるため，解グラフではノード，アークを用いる．

```
procedure Forward;
   initialize solution graph with only two nodes, S and E;
   for every merging-point v(i) (∈V(Gi)) do
      begin
         join:find all locally consistent labelings for
              r(Gi) and f(Gi(p));
         replace Gi(p) with a single vertex eliminating
              r(Gi);
         extend the solution graph by adding arcs (associated
              with labelings for r(Gi)) and nodes (associated
              with labelings for f(Gi(p))) {Notice:
              if f(Gi(p))=φ, add only the arcs and connect
              them to node E}
      end.
```

図 5 解グラフ生成アルゴリズム
Fig. 5 Algorithm that produces a solution graph.

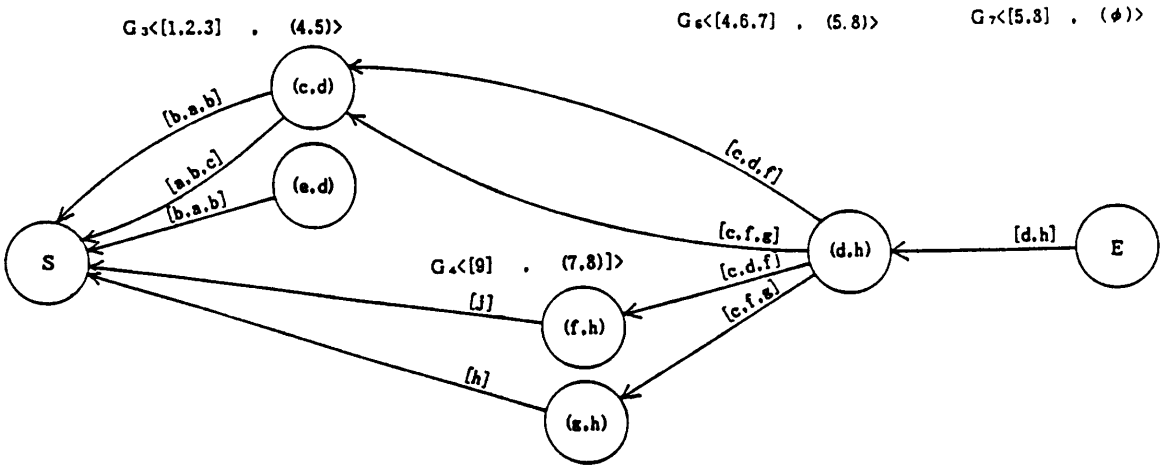


図 6 図4の invasion による解グラフ
Fig. 6 The solution graph derived by using the invasion of Fig. 4.

3.3.2 解グラフの走査（後退過程）

解グラフは，併合操作によって生じる冗長ユニットとそれに伴う情報を保存するデータ構造を与えている．最終解は，ノード E から始めて，始点とアークの対につけられた部分問題の解を，ラベルの整合性に注意しながら，ノード S までたどることにより得られる．すなわち，ノード E から S へ至るすべての道上の冗長ユニットのラベルが，すべての最終解である．ただし，ノード (d,h) のように，複数系統のアークが出ている場合は，各系統から同じラベルのものを一つずつ取り出し，一斉に平行走査する．

例えば，図6において，

$$E\rightarrow[d,h]\rightarrow(d,h)\rightarrow[c,d,f]\begin{cases}\nearrow(c,d)\rightarrow[b,a,b]\rightarrow S\\ \searrow(f,h)\rightarrow[j]\rightarrow S\end{cases}$$

のように，ノードとアークをたどってゆくと，次のような最終解が一つ得られる．ただし，[…] はアークの，また (…) はノードのラベル組をそれぞれ表す．

Unit (5 8 4 6 7 1 2 3 9)
解 (*d h c d f b a b j*)

これは，2.1 節で示した例解と同じものである．

3.4 計算量の評価

上に述べた方法は，invasion が，問題の局所的な部分構造を適切に取り出し，独立性の高い部分問題の系列を与える場合に，効果を発揮する．このことを計算量の解析により調べる．

まず，解グラフの走査（後退過程）においては，ノード E から S への道をたどる操作が最終解に対応しているため，処理時間は，問題のサイズ一定のとき，解の個数に比例する．本来，無駄な走査を含まないので，解析の対象から除く．

次に，解グラフの生成過程（前進過程）について考える．併合点 G_i における併合操作では，固有頂点を $v_{(i)}=(t_{(i)}, R_{(i)})$ とすると，冗長ユニット集合 $r(G_i)$ は $s(t_{(i)})$ の部分集合であるから，$r(G_i)$ に対して可能なラベル組の個数は，ラベル拘束関係 $R_{(i)}$ の大きさを越えない．また，$v_{(i)}$ を含む連結成分を $G_i^{(p)}$ とすると，front $f(G_i^{(p)})$ に対して可能なラベル組の個数は，$(\min\{\max r_i, |L|\})^{\phi_i}$ を越えない．ここに，$\max r_i$ は，$G_i^{(p)}$ 内の頂点（拘束条件ペア）の成分であるラベル拘束関係の最大サイズを表す．ϕ_i は G_i の front 指数．ここで，簡単のため，ラベル拘束関係のサイズを一定値 K とおくと，前進過程でチェックされる冗長ユニットと front の候補ラベル組の総数の最大値は

$$\sum_{i\in I} K\cdot(\min\{K, |L|\})^{\phi_i}$$

となる．ただし，I は併合点のインデックス集合．したがって，処理に要する計算時間および解グラフのための記憶領域の大きさが $O(|I|\times K\times(\min\{K,|L|\})^{\Phi})$ で押えられることがわかる．ただし，Φ は invasion の front 指数*．

この解析結果は，invasion の front 指数 Φ が計算量に影響を与えることを示す．また，最も単純に探索する場合，各ユニットに関して考えうるラベル組の総数は，$O(\min(|L|^{|U|}, |K|^{|T|}))$ となる．これに対し，上の結果は，部分問題ごとの計算量の和で済むこ

* これは，CLP が多項式時間で処理できることを意味するものではない．式中，$|I|, K, |L|, \phi$ はいずれも，問題および invasion に影響される値である．

とを示すもので，これは DP に共通する性質である．

4. invasion の最適化について

前節で述べたように，invasion の front 指数 Φ が小さいということは，**図7**の概念図に示すように，front が与拘束ネットワークの小さいカットセットであって，分断後の各連結成分が局所性の高い部分問題となることを示す．

ここでは，Φ を最小化するための方法について述べる．

4.1 諸性質

補題1 (3.2 節)より，invasion と頂点（拘束条件ペア）の列とを区別せずに用いることができる．

[定理 2] 頂点列 $v_1, \cdots, v_{|T|}$ で表される invasion があって，G_n を任意の併合点とする．また v_n より前にあるもののうち最も後にある併合点の頂点を v_m $(m<n)$ とする．ただし，v_n より前に併合点が存在しないならば，$m=0$ とする．このとき，部分列 $v_{m+1}, \cdots, v_n$ を任意に並べかえても，その最後の項 $v_n'=(t_n', R_n')$ が，$r(G_n)\subseteq s(t_n')$ を満たすならば，全体の invasion の front 指数は，始めの invasion のそれと等しい．

(証明) もとの頂点列および部分列 $v_{m+1}, \cdots, v_n$ を並べかえた頂点列の各 invasion を $\{G_i\}$, $\{G_i'\}$, $i=1, \cdots, |T|$ とする．まず，$i=1, \cdots, m, n, \cdots, |T|$ においては，$G_i=G_i'$ である（頂点集合が等しいゆえ）ので，3.2 節の定義より，それぞれの front 指数 ϕ_i と ϕ_i' は等しい．

次に，$m+1\leqq i\leqq n-1$ なる任意の i について，並べかえ後の第 i 項を $v_i'=(t_i', R_i')$ とする．ここで，G_n' は，その固有頂点 $v_n'=(t_n', R_n')$ に関する前提より，併合点であることに注意．$u\in s(t_i')$ なる任意のユニット u は，G_i' の冗長ユニットかまたは front の要素である．もし，冗長ユニットならば，$u\in s(t_n')$ である．ところが前提より，G_n' の冗長ユニットは $s(t_n')$ に含まれるから，u は G_n' の front の要素でなければならない．これは，u が G_i' の冗長ユニットであるという仮定に矛盾．したがって，$s(t_i')$ は冗長ユニットを含まないので，G_i' は併合点でない．よって，定義より G_i' の front 指数 ϕ_i は 0 となり，$\phi_i=\phi_i'(=0)$ が言える． □

[系 3] 定理2において，部分列 $v_{m+1}, \cdots, v_{n-1}$ を任意に並べかえても，invasion の front 指数は不変である．

定理2および系3は front 指数に関して，invasion のかわりに，より簡単な冗長ユニット集合列を調べてもよいことを意味する．

[定理 4] 拘束ネットワークを G，任意の $s(t)$ $(t\in T)$ の（空でない）部分集合を σ で表す．また，$\Gamma(\sigma)$ を

$$\Gamma(\sigma)=\bigcup_{u\in\sigma}\{(t, R_t)\,|\,t\in\tau(u),\ (t, R_t)\text{は拘束条件ペア}\}$$

なる頂点集合とする．このとき，

$$\Phi_{\min}\geqq\min_{\sigma}|f(\langle\Gamma(\sigma)\rangle_G)|$$

が成り立つ．ただし，$\Phi_{\min}$ は，G の invasion の最小 front 指数．

(証明) $\langle\Gamma(\sigma)\rangle_G$ は σ の選び方より連結グラフである．上記中，$|f(\langle\Gamma(\sigma)\rangle_G)|$ は既述のように，((G の $\Gamma(\sigma)$ に関する誘導部分グラフ[9])の front サイズ) を意味する．最適 invasion (その front 指数が $\Phi_{\min}$) における最初の併合点を G_i としよう．定義より，G_i 以降には，$r(G_i)$ の要素 (ユニット) は現れない．したがって，$r(G_i)$ の要素を成分とするようなユニット組はすべて G_i に現れている．このようなユニット組を成分とする拘束条件ペアの全集合は，

$$\bigcup_{u\in r(G_i)}\{(t, R_t)\,|\,t\in\tau(u),\ (t, R_t)\text{は拘束条件ペア}\}$$

となる．ここで，$\sigma\equiv r(G_i)$ とおくと，$r(G_i)\subseteq s(t_i)$ (t_i は G_i の固有頂点のユニット組) である．また，$\Phi_{\min}$ は G_i の front サイズ以上の値であるから，定理の不等式が導かれる． □

[定理 5] 部分集合 σ $(\subseteq U)$ が冗長ユニット集合となるには，

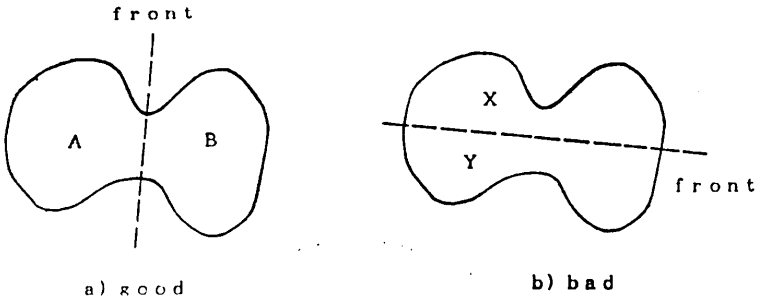


図 7 front の概念図
部分問題 A, B の方が，X, Y より局所性が高い．
Fig. 7 An image of fronts. Local problems A and B have higher locality than X and Y.

$\exists t(\in T)(\sigma \subseteq s(t))$

が成り立つことが必要である.

(証明) 略.

4.2 invasion の最適化アルゴリズム

最適 invasion を求める最も単純な方法は，補題1より，拘束条件ペアのあらゆる順列を調べ，front 指数が最小のものを求めるものである．しかし，この方法は，invasion を併合点で分断して得られる各部分列内で，任意に項を並べかえたものもすべて調べるという無駄を含む．実際には，定理2で示したように，併合点の相対位置，すなわち冗長ユニット集合の現れる順序のみを調べれば十分である．

系3および定理5を利用した最適 invasion アルゴリズムを図8に示す．図中，$\phi(V'' \cup \bigcup_{u\in\sigma} \tau_u)$ は，誘導部分グラフ $\langle V'' \cup \bigcup_{u\in\sigma} \tau_u \rangle_G$ の front 指数を表す．

手続き TRY は，ユニット部分集合 $U'(\subseteq U)$ および頂点集合 $V'(\subseteq V(G))$ に対して，広域変数 Min_front の値（これまでに調べた invasion の front 指数の最小値）より小さいものを探す再帰的手続きである．候補となる冗長ユニット集合 σ を，①〜③の条件のもとにチェックする．条件①は，冗長ユニット集合は一つのユニット組の成分として一斉に含まれる（定理5）という性質を表す．条件②は σ を導入することによって併合点に含まれるようになる（σ 以外の）関連ユニットはすべて front の要素であることを要請するものである．また，新たに調べようとする併合点の front 指数が，これまで発見済の最良の invasion (Solution にある) の front 指数 (Min_front) より小さくなければ，これ以上調べるのは無駄である．条件③はこのことを表す．

```
program OPTIMAL_INVASION;
  procedure TRY(U'; V'; F);
    begin
     V'':=V-V'; T':={t|(t,Rt)∈V'};
     for each σ(⊆U')[∃t(∈T')(σ⊆s(t))                                    ①
                    ∧{u|u∈s(t),(t,Rt)∈V'-⋃_{v∈σ} τv}=U'-σ                ②
                    ∧φ(V''∪⋃_{u∈σ} τu) < Min_front)]  do                 ③
       begin
         push σ to Dummy_solution;
         if U'-σ=empty then
            begin
               Solution:=Dummy_solution;
               Min_front:=F;
            end
          else
            TRY(U'-σ; V'-⋃_{u∈σ} τu); max{φ(V''∪⋃_{u∈σ} τu),F});
       end;
    end;{TRY}
  begin{main}
    Min_front:=∞;
    TRY(U; V; 0);
  end.{main}
```

図8 最適 invasion を求めるアルゴリズム
Fig. 8 An algorithm for an optimal invasion.

現在，試行中の invasion の途中解は，Dummy_solution スタックに格納されてゆく．そして，全ユニット U が出尽くした時点で，その試行解が新たな最良解であると判断し，Solution に格納・更新する．新たな front 指数は F に設定されている．

処理結果は，Solution スタックに格納されている．スタックの底から上へ順にユニット集合を取り出したものが，冗長ユニット集合列で，系3より，容易に最適 invasion が得られる．

5. あとがき

整合ラベリング問題 (CLP) を，等価な拘束ネットワーク構造に変換し，局所的な部分問題の系列に分解して解く方法を提案した．これは，ネットワークにおいて，1)小さなカットセットを探し，なるべく独立性の高い部分グラフに分解する，2)カットセットをグラフのいわば長軸（最大の二点間最短距離を与える道）方向に前進させつつ，局所的に拘束条件の整合化を行う，という基本的な考え方を用いた方法である．したがって，拘束ネットワークが線状または木に近い形状を有する場合[9]は，本方法の有効性が発揮される．しかし，各頂点の度数が大あるいは完全グラフに近いような場合は，効果がうすい．

なお，CLP の計算量の評価を front 指数を用いて行ったが，これは，問題を簡単な部分問題に分解するときの下限を示すものと考えることができる．一方，現実の問題においては，ユニットやユニット拘束関係の個数，ラベル拘束関係の大きさなどを考慮したより詳細な解析が要求される．

invasion の最適化については，発見的手法 (heuristics)[5] を導入した柔軟な準最適アルゴリ

ズムの開発が望まれる．また，今後の課題としては，許容度を有する CLP[10] の解法の研究があげられる．

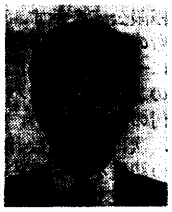
参考文献

1) Haralick, R. M. and Shapiro, L. G.: The Consistent Labeling Problem, Part I, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-1, No. 2, pp. 173-184 (1979).
2) Nudel, B.: Consistent-Labeling Problems and Their Algorithms: Expected-Complexities and Theory-Based Heuristics, *Artif. Intell.*, Vol. 21, No. 1 & 2, pp. 135-178 (1983).
3) Garey, M. R. and Johnson, D. S.: *Computers and Intractability*, Freeman, San Francisco, CA (1979).
4) Seidel, R.: A New Method for Solving Constraint Satisfaction Problems, *Proc. IJCAI, 7-th*, pp. 338-342 (1981).
5) 西原，原，池田：拘束ネットワークを用いた整合ラベリング法，電子通信学会論文誌，Vol. J 67-D, No. 7, pp. 745-752 (1984).
6) Bertele, U. and Brioschi, F.: *Nonserial Dynamic Programming*, Academic Press, N. Y. (1972).
7) 塩澤，西原，池田：整合ラベリング問題における拘束条件の構造について，第30回情報処理学会全国大会論文集，7 N-1, pp. 1309-1310 (1985).
8) Behzad, M., Chartrand, G. and Lesniak-Foster, L.: *Graphs & Digraphs*, Wadsworth, Belmont, CA (1979).
9) Lipton, R. J. and Tarjan, R. E.: A Separator Theorem for Planar Graphs, *SIAM J. Appl. Math.*, Vol. 36, No. 2, pp. 177-189 (1979).
10) 塩澤，西原，池田：ラベル組に重みを持つ整合ラベリング問題の解法，第31回情報処理学会全国大会論文集，4 P-3, pp. 1273-1274 (1985).

(昭和60年6月13日受付)
(昭和61年7月16日採録)

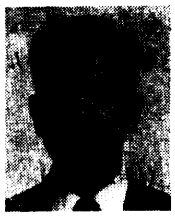
塩澤 恒道（正会員）

昭和37年生．昭和59年群馬大学工学部情報工学科卒業．昭和61年筑波大学大学院理工学研究科修士課程修了．現在，NTT 情報通信処理研究所勤務．プロセッサ論理構成の研究に従事．グラフ理論，組合せ最適化問題に興味を持つ．



西原 清一（正会員）

昭和21年生．昭和43年京都大学工学部数理工学科卒業．昭和43年同大学大型計算機センター助手，昭和50年より筑波大学電子・情報工学系．現在，助教授．昭和57年より1年間文部省在外研究員として，米国ヴァージニア工科大学に留学．工学博士．グラフィックス，画像処理，データ構造と非数値処理，探索問題に興味を持つ．本会論文賞（昭和50年）受賞．ACM, IEEE, 電子通信学会，人工知能学会各会員．



池田 克夫（正会員）

昭和12年生．昭和35年京都大学工学部電子工学科卒業．昭和37年同大学大学院修士課程修了．昭和40年同博士課程学修退学，同年京都大学助手．昭和46年同助教授．昭和46年9月より1年間文部省在外研究員として，米国ユタ大学および MIT に留学．昭和53年筑波大学教授，電子・情報工学系．コンピュータ組織法，LAN，画像処理に興味を持つ．著書に「オペレーティングシステム論」（電子通信学会）などがある．工学博士．電子通信学会，IEEE, ACM 各会員．