

## 拘束条件の局所性に基づいた整合ラベリング・アルゴリズム

A consistent labeling algorithm based on the locality of constraints

塩澤 恒道 西原 清一 池田 克夫

Tsunemichi SHIOZAWA Seiichi NISHIHARA Katsuo IKEDA

(筑波大学 電子・情報工学系)

Institute of Information Sciences and Electronics, University of Tsukuba

A Consistent Labeling Problem (CLP) involves finding labels for a set of units subject to a set of constraints on an arbitrary number of units. The purpose of CLP is to provide an efficient way of analyzing or understanding an object whose structural description is given by its primitive parts and their interrelationships. Applying dynamic programming technique, we propose a novel algorithm, whose basic idea is to decompose the given CLP into smaller subproblems. After estimating some complexity of the algorithm, we give a theorem, called Division Theorem, clarifying a property holding between the optimal decomposition and the structure of constraints.

## 1. まえがき

複数個の構成要素から成る対象を解析したり、解釈したりするには、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ない解釈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、同型部分グラフの探索、Nクイーン問題などの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<sup>[1]</sup> (consistent labeling problem, CLP)、拘束充足問題 (constraint satisfaction problem, CSP) など、様々な名前で呼ばれているが、ここではCLPと略記する。

CLPは、NP-完全な探索問題の一種であり、<sup>[5]</sup> 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二つの方法に大きく分類される。本稿では、これらと異なる動的計画法(DP)の手法を用いた解法について述べる。これは、1個以上の構成要素に対して全体の解の部分解となる候補を求め、それらの解を生成した順と逆に辿る事によって、全体の解を得るというものである。この結果CLPを解く為の計算量は、部分解を求める計算量の和により与えられることになる。

以下では、第2章でCLPの定義を与え、その等価表現である拘束ネットワーク定義する。第3章では、動的計画法の手法を用いた解法を示す。第4章では、第3章で示した解法を最適化する為の一般的な性質について論じる。

## 2. 問題の設定

## 2.1 CLPに関する諸定義

CLPは、四つ組  $(U, L, T, R)$  で与えられる。 $U = \{1, \dots, M\}$  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  $R = \{R_1, \dots, R_{\#T}\}$  で与えられる。 $\#$  は集合のサイズを表す。ここで、各  $R_i$  は'ラベル拘束関係'と呼ばれる。すなわち  $t_i (\in T)$  は、その成分(ユニット)がある拘束条件を満たす必要がある事を表し、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。

[CLPの例]

$$U = \{1, \dots, 9\}, L = \{a, \dots, k\},$$

$$T = \{t_1, \dots, t_7\},$$

$$t_1 = (1, 2, 4), t_2 = (1, 2, 3), t_3 = (2, 3, 4, 5),$$

$$t_4 = (5, 8), t_5 = (4, 6, 7), t_6 = (6, 7, 8, 9),$$

$$R = \{R_1, \dots, R_6\},$$

$$R_1 = \{(a, b, c), (b, a, c)\},$$

$$R_2 = \{(b, a, b), (a, b, c), (a, b, a)\},$$

$$R_3 = \{(a, b, c, d), (a, b, e, d), (b, c, c, d)\},$$

$$R_4 = \{(d, h), (e, i)\},$$

$$R_5 = \{(c, d, f), (d, c, e), (c, f, g)\},$$

$$R_6 = \{(f, g, h, h), (d, f, h, j), (g, h, i, j)\}. \quad \square$$

(2.1)

CLPを解くとは、全ユニット(1, . . . , M)に対するラベル組 $\lambda=(l_1, \dots, l_M)$ のうち、条件 $\forall t_i (\in T) (\lambda(t_i) \in R_i)$ を満たすすべての $\lambda$ を見付け出す事である。ただし、 $\lambda(t)$ は、 $\lambda$ のうち $t$ の構成ユニットに対応するラベルのみを取り出す射影である。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で与えられる拘束を満足している。

### 2.2 冗長ユニットとCLP

CLPの複雑さは、ユニットにある拘束条件を満足するようにラベルを与えたとき、そのラベル付けが、同時に他のユニット拘束条件および、ラベル拘束条件に矛盾しないものでなければならない点に起因している。したがって、ひとつのユニット拘束関係にしか現れないユニットに対するラベル付けは、他に同時に満たすべき拘束条件がないので、問題の複雑さに影響を与えない。このように、Uの要素で、高々1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'と呼ぶ。

上記の例で冗長ユニットは9であり、これを取り除いたCLPは次のようになる。

$U = \{1, \dots, 8\}$ ,  $L = \{a, \dots, k\}$ ,  
 $T = \{t_1, \dots, t_6\}$ ,

$t_1, \dots, t_5$  は(2.1)に同じ,  $t_6 = (6, 7, 8)$ ,

$R = \{R_1, \dots, R_6\}$ ,  $R_1, \dots, R_5$  は(2.1)に同じ,

$R_6 = \{(f, g, h), (d, f, h), (g, h, i)\}$ . □ (2.2)

すなわち、(2.2)の解でユニット(6, 7, 8)に対するラベル組が(d, f, h), (g, h, i)であるものについてはユニット9にラベルjを、(f, g, h)であるものについてはユニット9にラベルhを与えれば、(2.1)のCLPの解が得られる。

### 2.3 拘束ネットワークによる等価表現

ここで、CLPを等価表現するために次のような無向グラフ(V, E)を考え、'拘束ネットワーク'と呼ぶ。

$V = \{(t_i, R_i) \mid t_i \in T\}$ ,

$E = \{(V_i, V_j) \mid s(V_i) \cap s(V_j) \neq \emptyset, V_i, V_j \in V\}$

ただし、 $s(V_i) = s(\{(t_i, R_i)\}) = s(\{(u_1, \dots, u_k)\}) = \{u_1, \dots, u_k\}$  □

上の(t<sub>i</sub>, R<sub>i</sub>)を'拘束条件組'と呼ぶ。本定義は、従来のものと異なる。従来の拘束ネットワークは、ユニット拘束関係が、二項関係であるCLPについてのみ定義されており、各ユニットとユニット拘束関係が、それぞれ頂点と辺に1対1対応していた。本稿の定義では、頂点に対応するのは拘束条件組であり、任意の2

つの拘束条件組の間で、ユニット組の中に共通のユニットが一つ以上存在する事が、辺に対応している。(2.1)のCLPを等価表現する拘束ネットワークを図2.1に示す。

### 3 拘束ネットワークの構造とinvasion

#### 3.1 拘束ネットワーク上での節併合操作<sup>(5)</sup>

拘束ネットワーク上でCLPを解く方法として、節併合操作を繰返し、漸次頂点の個数を減少させてゆく方法がある。

拘束ネットワークG内の頂点V<sub>1</sub>とV<sub>2</sub>の節併合操作はGを次のような(V', E')に変換する操作である。

$V' = V(G) - \{V_1, V_2\} \cup \{V_{1,2}\}$ ,

$E' = E(G) - \{(V_1, V) \mid (V_1, V) \in E(G)\}$

$- \{(V_2, V) \mid (V_2, V) \in E(G)\}$

$\cup \{(V_{1,2}, V) \mid (V_1, V) \in E(G) \text{ or}$

$(V_2, V) \in E(G), V \in V'\}$

ここで、頂点V<sub>1,2</sub>は、ユニットs(t<sub>1</sub>) $\cup$ s(t<sub>2</sub>)を成分とするユニット組t<sub>1,2</sub>と、t<sub>1,2</sub>に対して拘束条件組V<sub>1</sub>とV<sub>2</sub>を満足するラベル拘束関係R<sub>1,2</sub>との組からなる。□

図2.1に示された拘束ネットワークにおいて頂点V<sub>1</sub>とV<sub>2</sub>に節併合操作を行えば図3.1に示すような拘束ネットワークが得られる。

中間結果の候補ラベリングをなるべく少なく押えるためには、節併合操作は隣接した頂点の間で行う必要がある。したがって、拘束ネットワークが非連結であるような場合には、各連結成分ごとに解を求めれば、それらの直積、すなわち解のすべての組合せが元の拘束ネットワークの最終解となっている。このため以下では、連結な拘束ネットワークのみを取り扱う。

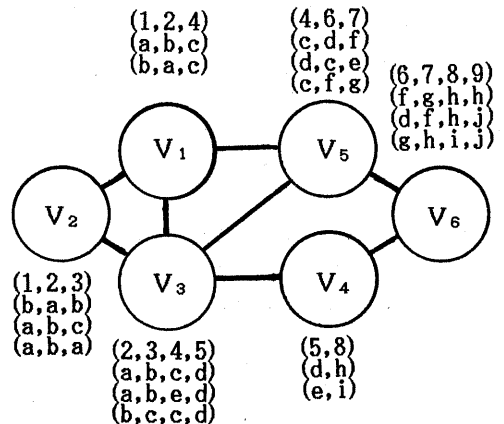


図2.1 拘束ネットワークの例

前節で述べたように、冗長ユニットを除去しても、単純な後処理(この後処理のための情報を効率良く保存する方法については、3.3節で述べる。)を行なうことで、元の拘束ネットワークの解は得られる。節併合操作の順序を適切に選ぶことによって冗長ユニットを早い段階で発生させることが出来る。図3.1に示された拘束ネットワークは、節併合操作の結果、ユニット1が冗長ユニットとなっている。次に、 $V_{1,2}$ と $V_5$ を併合した結果(図3.2(a))と $V_{1,2}$ と $V_3$ を併合した結果(同図(b))とを比較してみよう。前者には、冗長ユニットは発生しないが、後者では、ユニット2、3が冗長ユニットとなっている。

冗長ユニットを早期に発生させ、それに伴う情報を各段階での部分解として保存しておき、これを用いて解を求めようとするのが、本稿での基本的な考えである。

### 3.2 invasionとfront指数

前節で述べたように、併合操作の適用順序が処理の効率に影響する。ここでは、この順序に関する諸定義を文献[2]を拡張して与える<sup>[7]</sup>。

前節で述べた節併合操作の順序を与えるためにinvasionを次のように定義する。

$G$ の'invasion'とは、列 $\{G_i\}, i=1, \dots, \#T$ 。ここに、各 $G_i$ は $i$ 個の頂点を持つ $G$ の誘導部分グラフであり、 $G_i \subseteq G_{i+1}$ なる関係を持つ。□

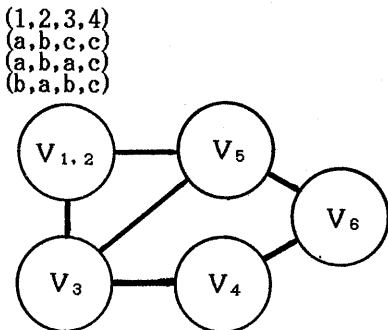


図3.1  $V_1$ と $V_2$ の節併合後の拘束ネットワーク

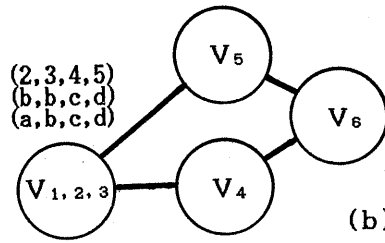
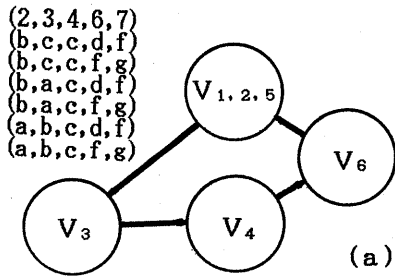


図3.2 2回目の節併合後の拘束ネットワーク

節併合操作の結果生成された頂点のユニット組から冗長ユニットを取り除いた残りの全ユニットをfrontという。すなわち、

$G_i (\subseteq G)$ の'front' $f(G_i)$ は、次のような $U$ の部分集合である。  $f(G_i) = s(V(G_i)) \cap \overline{s(V(G_i))}$ 。

$$\overline{V(G_i)} = V(G) - V(G_i),$$

$$s(V) = \bigcup_{V_i \in V} s(V_i) = \bigcup_{(t_i, R_i) \in V} s((t_i, R_i)) = \bigcup_{(t_i, R_i) \in V} s(t_i) \square$$

節併合操作によって生成された各頂点から冗長ユニットを取り除いた頂点の中に現れるユニット数の最大値として、front指数を定義する。すなわち、

$G_i (\subseteq G)$ の'front指数' $\phi_i$ は、 $G_i$ の連結成分を $G_i^{(1)}, \dots, G_i^{(c)}$ とすると、

$$\phi_i = \begin{cases} \max_{1 \leq j \leq c} \#f(G_i^{(j)}) & : \\ 0 & : [f(G_{i-1}) - f(G_i) \neq \phi \text{の時}] \\ & [f(G_{i-1}) - f(G_i) = \phi \text{の時}] \end{cases}$$

で定義される。ただし、 $G_0$ は空グラフである。□

すなわち、節併合操作をinvasionに沿って進めてゆき、冗長ユニットが生じたならfront指数 $\phi_i$ は上段をとり、それ以外のときは0である。 $\phi_i$ は、節併合操作によって冗長ユニットが生じた時に、各連結成分ごとに節併合操作を行い、冗長ユニットを除去した頂点に含まれるラベル拘束関係の次元の最大値を示すものである。冗長ユニットが発生しない場合は、頂点間の整合をとる処理(後述)を行わない。

invasion $\{G_i\}, i=1, \dots, \#T$ のfront指数('invasionのfront指数') $\Phi$ は、 $\max \phi_i$ と定める。□

$G$ の'最適invasion'とは、 $G$ の全てのinvasionのうち最小のfront指数を持つものをいう。□

図2.1に示した拘束ネットワークのinvasionの例を図3.3に示す。それぞれのfront指数 $\phi_i$ は、 $G_1$ においては冗長ユニットが発生しないので0、 $G_2$ においては $f(G_2) = \{2, 3, 4\}$ であり連結成分が一つであるので $\phi_2 = 3$ 、同様に、 $\phi_3, \phi_4, \phi_5, \phi_6$ はそれぞれ、3、3、3、0となる。よって、このinvasionのfront指数 $\Phi$ は3となる。ここに示された例は、 $G$ の最適invasionの例にもなっている。

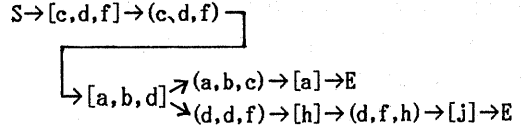
### 3.3 解グラフの生成

前節のinvasionは節併合操作の順序を与えるものである。節併合操作によってネットワーク内に冗長ユニットが発生する。これらの冗長ユニットは、上述のfrontとの整合情報を保存した後、拘束ネットワークから取り除く。このような情報を保存するのが、解グラフである。解グラフには、まずノードE、Sが設定されており、順次ノードEからノードSへ向かって成長していく。各ノードは、invasionにおいて、冗長ユニットが生じた時(すなわち、front指数が非零になった時)に整合化処理を行い、必要個数だけ生成される。ここで、各ノードに対しては、frontに対して可能な整合ラベル組をラベルとして与え、アークに対しては、冗長ユニットに対する整合ラベル組をラベルとして与える。

図3.3のinvasionにより得られる解グラフを図3.4に示す。

解グラフは、節併合操作によって生じる(節併合操作は実際には行わない)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。このグラフのアークに付けられたラベル組は、冗長ユニットに対する全体の解の部分解の候補を与えている。この部分解

が、現実には最終解の一部を構成するものであるためには、アークの始点に与えられているラベル付けを満足しなくてはならない。一方、アークの終点においては、アークのラベル付けと始点のラベル付けに整合するラベル付けが行われている。ゆえに、この解グラフのノードSからEに至る道のうち、『各ノードを始点とする同一のラベルをもつアーク(一つとは限らない)を選んで、各アークの終点ノードに進む』という処理によって得られる道は矛盾しない全体の解を与える。たとえば、図3.4の例において、



のように、ノードとアークを辿って行くと(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)、次のような最終解が一つ得られる。

Unit 4 6 7 2 3 5 8 9 1  
Label c d f a b d h j b □

これは、(2.1)で示した例解である。

このように、各段階で可能なすべての部分解を求め、最後にこれらを生成した順とは逆に辿ることにより解

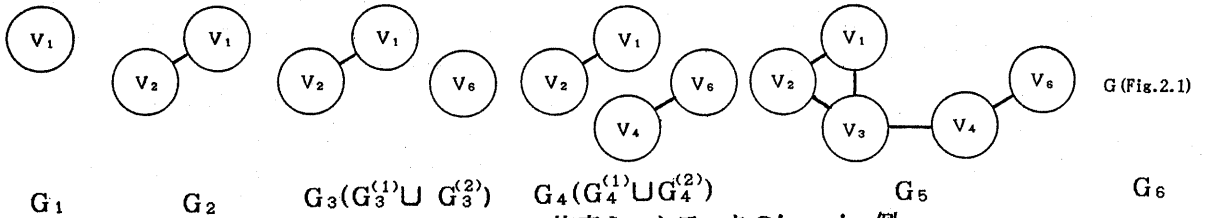


図3.3 図2.1の拘束ネットワークのinvasion例

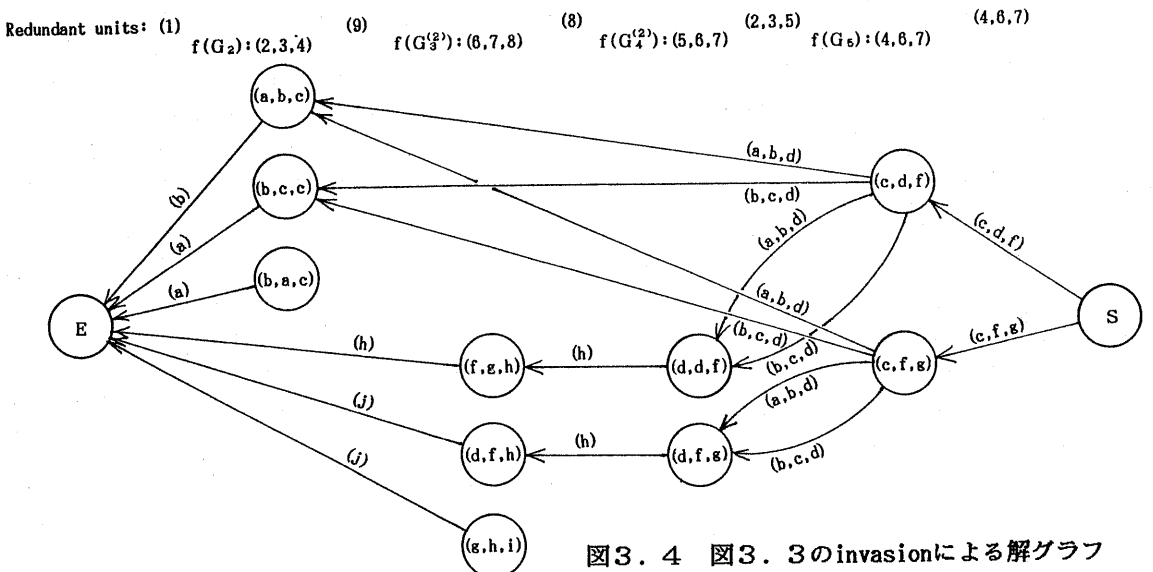


図3.4 図3.3のinvasionによる解グラフ

を求める方法は、動的計画法(DP)の手法を援用したものである。その決定変数、状態変数にはユニット、およびinvasionの連結成分に対するfrontが、それぞれ対応している。解グラフの意味から明らかなように、もし、ある段階で連結成分のfrontに対するラベル(ノード)が空となった場合は、全体の解も空である。図3.4に示された解グラフより得られる解は、次の4つであり、これは、(2.1)のCLPの解である。

[例題の解]

Unit	4	6	7	2	3	5	8	9	1
Label	c	d	f	a	b	d	h	j	b
	c	d	f	b	c	d	h	j	a
	c	f	g	a	b	d	h	h	b
	c	f	g	b	c	d	h	h	a

ここで、解グラフ生成のための計算量について考察する。

[前提] #R<sub>i</sub>=k, i=1, ..., #T

また、計算量の単位はinvasionの各段階で考慮すべきラベル組の個数とする。

invasionの各段階でラベル付けを考慮しなくてはならないユニットの数は、高々除去可能なユニット(すなわち冗長ユニット)の数と、front指数の和である。しかし、除去可能なユニットは、一つの頂点にすべて含まれているので、それらのユニットに対するラベル組の総数は、高々k個である。また、仮定より、連結成分のfrontに含まれるユニットに対するラベルは、それぞれ高々k個であるから、第i番目で考慮すべきラベルの総数は、高々、

$$O(k^{\phi+1})$$

となる。さらに、解グラフは、冗長ユニットの発生と共に成長するが、その回数を $\Omega$  ( $\leq \#T$ )とすれば、invasionによる全体の計算量は、高々、

$$O(\Omega \times k^{\phi+1}) \quad (5.1)$$

である。

#### 4 invasionに関する性質

前節で述べたように、front指数が小さくなると、計算量の上限が減少する。ここでは、最適invasionを求める上での手掛りとなるinvasionとfront指数についての基本的な性質を述べる。

##### 4.1 諸性質

性質4.1 invasionは、拘束条件組の列によって等価表現される。

証明) 拘束条件組の列 $\{v_i\}, i=1, \dots, \#T$ 。が与えられた時、 $G_i = \langle \bigcup_{j=1}^i \{v_j\} \rangle_{\alpha}$  (4.1) で与えられる列 $\{G_i\}, i=1, \dots, \#T$ は、一意に定まり、これは、invasionを与える。

ここで、 $\langle V' \rangle_{\alpha} (V' \subseteq V(G))$ は、GのV'による誘導グラフ $(V', \{(v,w) \in E(G) | v \in V' \text{ and } w \in V'\})$ であることを表す。

逆は明らかである。

拘束条件組の列中の要素であって、冗長ユニット(それより後の拘束条件組には現れないユニット)を一個以上含むものは、解グラフを成長させるための整合化処理を実施すべき拘束条件組である。このような拘束条件組を、'整合点'と呼び他と区別する。

定理4.2 拘束条件組の列 $\{v_1, \dots, v_{\#T}\}$ があり、それと等価なinvasionのfront指数を $\Phi$ とする。任意の整合点 $v_m$ に含まれる冗長ユニットの集合をSとする。さらに、 $1 < m$ で最大のlを持つ整合点を $v_l$ とする。この時、部分列 $v_{l+1}, \dots, v_m$ の要素を任意に入れ替えても、その部分列の最後の要素 $v_m$ が $S \subseteq s(v_m)$ を満たすならば、この拘束条件組の列と等価なinvasionのfront指数は、 $\Phi$ に等しい。

証明) 与えられた列と、 $v_{l+1}, \dots, v_m$ を任意に入れ替えることにより得られる列 $\{v'_1, \dots, v'_{\#T}\}$ から得られるinvasionを式(4.1)により、それぞれ、 $\{G_i\}, \{G'_i\}, i=1, \dots, \#T$ とする。 $G_i = G'_i, i=1, \dots, l, m, \dots, \#T$ であるのでfront指数は変化しない。

また、lの選び方より、 $v_{l+1}, \dots, v_{m-1}$ の成分となっているユニットは、すべて $v_m, \dots, v_{\#T}$ のいずれかの成分となっている。さらに、 $G_l$ のfrontの要素もまた、すべて $v_m, \dots, v_{\#T}$ のいずれかの成分となっており、

$$f(G_l) = f(G'_{l-1}) \cup s(v'), \quad i=l+1, \dots, m-1$$

である。ゆえに、 $f(G'_{l-1}) - f(G'_l) = \emptyset, i=l+1, \dots, m-1$ であるので、入れ替えによりfront指数は変化しない。

性質4.1はinvasionをグラフの列で与える必要は無く、頂点の列、即ち、拘束条件組の列として等価に取り扱う事ができることを意味している。また、定理4.2はinvasionを指定することと冗長ユニットが生成される順序(これを、レベルと呼ぶこととする。)を与えることとは、等価に取り扱う事ができることを意味する。ここで、同一のレベルのユニットに対しては、必ずそれらをすべて成分とするユニット拘束条件が存在しなくてはならない。

また、定理4.2に加えて、次の定理は、拘束条件組の順序を変化させてもfront指数を増加させない場合を与えている。

定理4.3 任意の拘束条件組の列 $\{v_i\}$ ,  $i=1, \dots, \#T$ が与えられ、これに対応するinvasionのfront指数を $\Phi$ とする。整合点でない任意の拘束条件組 $v_j$  ( $j \leq \#T$ )について、 $v_{j+1}$ が $s(v_j)$ の要素を冗長ユニットとして含む整合点でないとする。このとき、 $v_j, v_{j+1}$ の順序を入れ替えた列に対応するinvasionのfront指数を $\Phi'$ とすれば、 $\Phi' \leq \Phi$ である。 ■

証明)  $v_j, v_{j+1}$ の順序を入れ替えた拘束条件組の列を $\{v_i'\}, i=1, \dots, \#T$ とする。拘束条件組の列 $\{v_i\}$ ,  $\{v_i'\}, i=1, \dots, \#T$ より与えられるinvasion $\{G_i\}$ ,  $\{G_i'\}, i=1, \dots, \#T$ に対応するfront指数の列をそれぞれ $\phi_i, \phi_i', i=1, \dots, \#T$ とする。

ここで、 $\phi_i = \phi_i', i=1, \dots, j-1, j+2, \dots, \#T$ 。

$v_{j+1}$ が整合点でないとき、定理4.2より、

$$\phi_j = \phi_j', i=j, j+1.$$

$v_{j+1}$ が整合点のとき、 $\phi_j = \phi_{j+1}'$ 。さらに、 $v_{j+1}$ の冗長ユニットは $v_j'$ でも冗長ユニットとなる。また、 $G_j' \subseteq G_{j+1}$ であるので、 $G_j'$ で同一の連結成分に現れるユニットは $G_{j+1}$ でも同一の連結成分に現れるので、 $\phi_j' \leq \phi_{j+1}$ 。

ゆえに、 $\Phi' \leq \Phi$ となる。 □

この定理は、任意の整合点でない拘束条件組を、そこに現れるユニットが、初めて冗長ユニットとなる整合点の前まで後送りにしても、front指数を増加させないことを示している。さらに、この定理を最適invasionに関して適応することによって、考慮すべき拘束条件組の列を限定することができる。

#### 4.2 分割定理

ここで、拘束ネットワークをユニットの結合のうえから二つに分割して考えられる事を述べる。

まず、拘束ネットワークを最も弱い結合部分で分割するために拘束ネットワークのカットセットを次のように定義する。

'front\_cut\_set'  $C (\in E(G))$ とは、連結グラフ $G$ に対して、 $G' = (V(G), E(G) - C)$ が非連結グラフとなる辺集合である。 □

'front\_cut\_set'  $C$ の大きさとは、

$$\# \bigcup_{(v,w) \in C} \{s(v) \cap s(w)\}$$

である。 □

'最小front\_cut\_set'とは、あらゆるfront\_cut\_setの中で大きさが最小のものである。 □

図2.1の拘束ネットワークにおいては、たとえば、 $C = \{(v_1, v_5), (v_3, v_5), (v_3, v_4)\}$ が、front\_cut\_setのとなる。このとき、front\_cut\_setの大きさは $2(\#\{4, 5\})$ であり、このfront\_cut\_setは最小front\_cut\_setでもある。最小front\_cut\_setについて、次のような性質が成り立つ。

補題4.4 連結グラフ $G$ の最小front\_cut\_setを $C^*$ とし、 $G' = (V(G), E(G) - C^*)$ とする。 $G'$ の一つ以上の連結成分から成る任意のグラフの頂点集合を $W$ とすると、 $\langle W \rangle_G$ は連結グラフである。 ■

証明)  $G'$ の任意の二つの連結成分 $G'^{(1)}, G'^{(2)}$ の間に、共通なユニットが存在することをいえよ。

$f^* = \bigcup_{(v,w) \in C^*} \{s(v) \cap s(w)\}$ とすると、 $C^*$ は連結グラフ $G$ の最小front\_cut\_setであり、 $G'^{(1)}$ と他の連結成分に共通に現れるユニットはすべて $f^*$ の要素となっているので、 $f^* \subseteq s(V(G'^{(1)}))$ 。

ゆえに、 $s(V(G'^{(1)})) \cap s(V(G'^{(2)})) = f^*$ となり、 $G$ は連結なので $\#f^* \geq 1$ であるから、 $G'^{(1)}, G'^{(2)}$ の間に、共通なユニットが存在する。 □

定理4.5 (分割定理) 連結な拘束ネットワーク $G$ の最小front\_cut\_setを $C^* (f^* = \bigcup_{(v,w) \in C^*} \{s(v) \cap s(w)\})$ 、

$G' = (V(G), E(G) - C^*)$ とする。 $G'$ の一つ以上の連結成分から成る任意のグラフの頂点集合を $W$ とする。

さらに、 $G^A = \langle W \rangle_G, G^B = \langle V(G) - W \rangle_G$ とおく。このとき、 $G^A$ または $G^B$ をinvasionの系列中に持つ最適invasionが、少なくとも一つ存在する。 ■

証明)  $G^A \subseteq G_{\#T-1}$ なる最適invasion $\{G_i\}, i=1, \dots, \#T$ が、少なくとも一つ存在するとしても一般性を失わない。

このとき、 $\langle V(G_i) \cap V(G^A) \rangle_G, i=1, \dots, \#T$ より得られる $\#V(G^A)$ 個のグラフの列を

$$\{G_i\} (\#V(G_i) = i), i=1, \dots, \#V(G^A)$$

とする。さらに、 $\langle V(G_i) \cup V(G^A) \rangle_G, i=1, \dots, \#T$ より得られる $\#(V(G) - V(G^A))$ 個のグラフの列を

$$\{G_i'\} (\#V(G_i') = i), i=\#V(G^A)+1, \dots, \#T$$

とする。ここで、 $\{G_i'\}, i=1, \dots, \#T$ に対してfront指数 $\phi_i'$ を考える。

$i \geq \#T - 1$ のとき、 $G^A \subseteq G_{\#T-1}$ より、 $G_i' = G_i$ であるので、 $\phi_i' = \phi_i$ 。

$i < \#T - 1$ のとき、 $p = \#(V(G^A) \cap V(G_i))$ とおくと、

$$p = i \text{ ならば、 } \phi_i' = \phi_i.$$

$p < i$ ならば、次のように $S, R, f_1, f_2$ を定める。

$$S = s(V(G_i) - V(G^A)) \cap s(V(G) - V(G_i))$$

$$R = s(V(G_i) - \overline{V(G^A)}) \cap s(V(G) - V(G_i))$$

$$f_1 = s(V(G_i) - V(G^A)) \cap s(V(G_i) - \overline{V(G^A)})$$

$$f_2 = s(V(G^A) \cap s(\overline{V(G^A)} - V(G_i)))$$

ここで、補題4.4より $G - G^A$ は、連結であり、 $G^A \subseteq G_{\#T-1}$ であるので、 $S \neq \emptyset$ 。

また、 $f_1 \cup f_2 \subseteq f^*$ である。

$C^*$ は最小front\_cut\_setであるので、

$\#(f_1 \cup f_2) \leq \#f^* \leq \#(S \cup f_2)$ であり、

ゆえに、 $\#(f_1 - f_2) \leq \#(S - f_2)$ である。

$S - f_2$ 内のユニットで、 $R$ に含まれるユニットは  $f_1 - f_2$ に含まれる。ゆえに、 $f_1 - f_2$ 内のユニットで  $R$ に含まれないユニットの個数は、 $S - f_2$ 内のユニットで  $R$ に含まれないユニットの個数より少ない。また、 $f_1 \cap f_2$ なるユニットは、全て  $S$ 内に含まれるので、 $\#(R \cup f_1) \leq \#(R \cup S)$ となる。

ゆえに、 $\phi_p' \leq \phi_1$ 。

同様に、 $q = \#(V(G^A) \cup V(G_1))$ とおくと、

$q = i$ ならば、 $\phi_1' = \phi_1$ 。

$q > i$ ならば、次のように  $S, R, f_1, f_2$ を定める。

$$S = s(\overline{V(G_1)} - \overline{V(G^A)}) \cap s(V(G_1))$$

$$R = s(\overline{V(G_1)} - V(G^A)) \cap s(V(G_1))$$

$$f_1 = s(\overline{V(G_1)} - \overline{V(G^A)}) \cap s(\overline{V(G_1)} - V(G^A))$$

$$f_2 = s(\overline{V(G^A)}) \cap s(V(G^A) - \overline{V(G_1)})$$

ここで、補題4.4より  $G^A$ は、連結であり、 $q > i$ であるので、 $S \neq \emptyset$ 。

上記の場合と同様にして、 $\phi_q' \leq \phi_1$ となる。

よって、invasion $\{G_i'\}, i=1, \dots, \#T$ は、最適invasionである。 □

図2.1に示した拘束ネットワーク  $G$ について、図3.3に示す最適invasionより得られる、 $\langle \{V_1, V_2, V_3\} \rangle_G$ をその系列に含む最適invasionを図4.1に示す。

定理4.5より、拘束ネットワーク  $G$ の最適invasionを求めるには、最小front\_cut\_setを  $G$ から除去することによって得られる連結成分を二つに分割し、そのそれぞれの頂点に対応する拘束条件組のあらゆる列について検討すればよいことになる。

#### 5 あとがき

整合ラベリング問題(CLP)は、一般的に、NP-完全な問題であることが知られている。本稿で示した解法は、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部分解を矛盾無く結合するこ

とによって全体の解を得るものであり、これは、動的計画法の手法を用いた解法である。この解法は、拘束条件の局所性が強い程、有効である。また、問題を分割する最良の方法は、最適invasionとして与えられる。この最適invasionのfront指数は、拘束条件の構造の中で、分割によってそれ以上問題のサイズが小さくならない限界値を与えている。これは、個々のCLPの拘束条件の構造及び、計算量に関するクラスを与えるものである。特に、本稿では、従来のように拘束条件を二項関係に限定せず、一般の多項関係を対象としているので、これらの持つ構造情報を利用することによって、さらに、ラベルの組合せの総数を減少させる可能性を持つ。

本稿で示した解法は、文献[2]の多項関係への単純な拡張ではなく、多項関係より成るCLPの最適invasionによる解法は、一般的に、拘束条件の二項関係への変換により得られるものでは無いことを示している。すなわち、多項関係であることを積極的に利用することによって、部分解の計算量をより厳密に評価し、より最適なinvasionを得るようにしたものである。また、front指数の定義を連結成分ごとに評価するよう拡張することにより、一般的なCLPの拘束条件の構造の局所的な複雑さを、front指数で正確に表すことが可能になった。

最後に、最適invasionを求めるための幾つか性質について述べた。ここで示された性質は、最適invasionを求める上での、基本的な性質である。特に、定理4.5は、最適invasionと、拘束条件間の関係が弱い部分との間にある関係を与えている。

一般に、最適invasionを求めることは、簡単では無い。しかし、最適invasionの持つ性質を調べる事によって、CLPの持つ構造を明らかにすると共に、準最適なinvasionを求める発見的アルゴリズムを評価することが可能となる。

現在、準最適invasionを求める高速アルゴリズムについて幾つかの有用な結果を得ており、これらを評価することと、拘束ネットワークやinvasionの持つ性質について、さらに、検討することが今後の課題である。

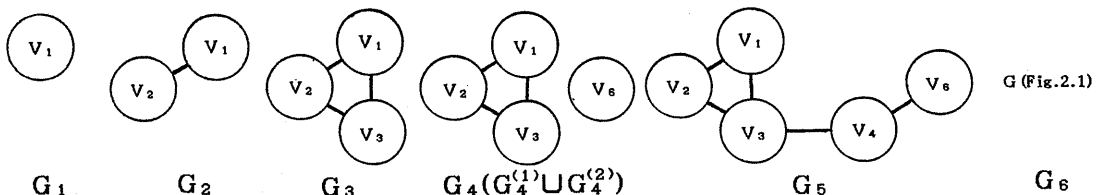


図4.1  $\langle \{V_1, V_2, V_3\} \rangle_G$ を系列に含む最適invasion

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] Seidel, R.: A New Method for Solving Constraint Satisfaction Problems, Proc. IJCAI, 7th, pp. 338-342 (1981).
- [3] Freuder, E.C.: A Sufficient Condition for Backtrack-Free Search, JACM, Vol. 29, No. 1, pp. 24-32 (1982)
- [4] Lipton, R.J. and Tarjan, R.E.: A Separator Theorem for Planar Graphs, SIAM J. Appl. Math., 36, pp. 177-189 (1979)
- [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. J67-D, No. 7, pp. 745-752 (1984).
- [6] 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, 情報処理学会第30回全国大会, 7N-1, pp. 1309-1310 (1985)
- [7] 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, 情報処理学会第31回全国大会, 4P-3, pp. 1273-1274 (1985)
- [8] Bertele, U. and Brioschi, F.: Nonserial Dynamic Programming, Academic Press (1972)
- [9] Behzad, M., Chartrand, G. and Lesniak-Foster, L.: Graphs & Digraphs, Prindle, Weber & Schmidt (1979)