許容度を有する整合ラベリング問題解法の効率化とシステムについて

3M-4

松尾嘉和　西原清一　池田克夫
(筑波大学　電子・情報工学系)

1．はじめに

対象物の構造情報が局所的解釈とその妥当性を表す重みによって与えられている場合、局所的解釈の重みの和がある値を越えないような全構成要素の解釈を求める問題を、inexact-CLP(Consistent Labeling Problem)[1]と呼ぶ。このような問題は、航空写真より地形の構造を求めたり、構成要素に優先順位があるようなパターンの認識など多くの分野に見出される。本稿では、文献[2]で述べた動的計画法(DP)の手法を援用した解法で、集積誤差を一段先まで読む方法[3]について効率改善を行い、計算機実験により評価を行う。また、アルゴリズムを状況により選択するシステムの構想について触れる。

2．inexact-CLPと拘束ネットワーク

inexact-CLPは、五つ組(U,L,T,W,ε_0)で与えられる。ここに、U={1,...,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釈や解析に対応する。T={t_1,...,$t_{|T|}$}は'ユニット拘束関係'といい、ユニットの多項組t_i('ユニット組')の集合である。

また、W={w_1,...,$w_{|T|}$}。ここで、各w_i:$L^{|s(t_i)|}$→Real。ただし、$s(t_i)$=s((u_1,...,u_k))={u_1,...,u_k}。ε_0は、この集積誤差の'許容度'を表す任意の実数。

[inexact-CLPの例]

U={1,...7},L={a,...,e},

T={t_1,...,t_5},

t_1=(1,2,3),t_2=(1,3,5),t_3=(2,4,5),t_4=(3,4,6),t_5=(4,6,7),

W={w_1,...,w_6},

w_1:(a,e,d)→5 ,(d,d,e)→12,(b,e,c)→58,(e,a,e)→81,

w_2:(a,d,b)→4 ,(d,e,c)→9 ,(a,d,e)→36,(b,b,a)→79,

w_3:(d,a,c)→28,(e,c,b)→36,(e,e,a)→78,

w_4:(d,c,a)→16,(e,a,b)→21,(c,a,a)→39,

w_5:(c,a,a)→11,(a,b,e)→22,(e,b,a)→66,(b,c,e)→94,

ε_0=96。

inexact-CLPを解くとは、全ユニット(1,...,M)に対して、条件

$$\varepsilon = \sum_{t_i \in T} w_i(h(u_1),\ldots,h(u_{|s(t_i)|})) \leq \varepsilon_0$$

を満たす全ての写像h:U→Lを見付けだす事である。ただし、t_i=(u_1,...,$u_{|s(t_i)|}$)。εをhによる'誤差'と呼ぶ。

[例題の解] 1 2 3 4 5 6 7

a e d c b a a　ε=72

d d e a c b e　ε=92

次に、inexact-CLPに対して'拘束ネットワーク'を次のような無向グラフ(V,E)で定義する。

V={(t_i,w_i)|$t_i \in T$},

E={(v_i,v_j)|s(t_i)∩s(t_j)≠φ,v_i,v_j∈V}

また、'冗長ユニット'とは、Uの要素で、高々一つのユニット組の成分としかならないユニットである。例においては、ユニット7が冗長ユニットである。'front'(f)とは、冗長ユニットを含むユニット組内の成分で冗長ユニットでないものの集合である。

'頂点併合操作'とは、拘束ネットワーク上の隣接した二つの頂点v_i,v_jを、拘束ネットワークの定義に矛盾しないように一つの頂点v_{ij}=(t_{ij},w_{ij})で置き換える操作である。ただし、$s(t_{ij})$=$s(t_i)\cup s(t_j)$、w_{ij}は、w_i,w_jを用いてユニット集合$s(t_{ij})$の関数として合成したものである。w_{ij}の値は、w_iとw_jの値の和をとる。図1において、頂点v_1とv_3を頂点併合操作した結果を、図2に示す。

頂点併合操作を繰り返し適用していくことによって、解を得ることが可能であるが、新たなノードに含まれるべき情報量は急激に増大する。そこで、冗長ユニットを除去することによって情報量の増大を出来るだけ抑制する方法が考えられる。まず、頂点併合操作の順序を与えるものとしてinvasion、すなわち拘束ネットワークの部分グラフの列をSeidel[4]を拡張して次のように定義する。

Gの'invasion'は、列{G_i},i=1,...,|T|。各G_iはi個のノードを持つ誘導グラフであり、$V(G_i) \subseteq V(G_{i+1})$である。図3は、invasionの例である。頂点併合操作は各部分グラフの連結成分において行う。

冗長ユニットとfrontに関する情報を解グラフに残し、冗長ユニットを取り除いた拘束ネットワークについて解を求め、その解のfrontに整合する冗長ユニットに関する情報を付け加えた解の中で、誤差がε_0を超えないものが、元の問題の解である。従って、invasionの各段階において冗長ユニットが発生した場合、それを除去することによって、考慮するユニットを減少させることができる。また、頂点併合を行う順序によって考慮すべきラベルの数が異なるので、最も効率のよいinvasionを探しだすことが重要となる。

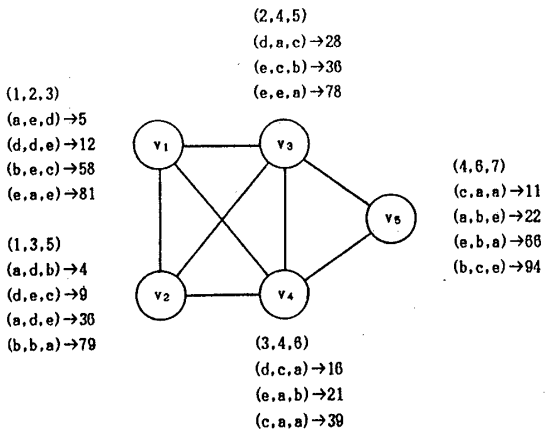

図1　例題の拘束ネットワーク(G)　|U|=7,|L|=5,|T|=5

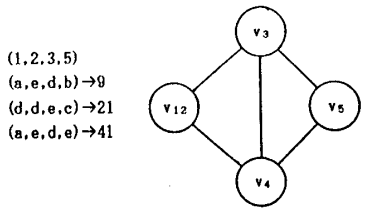

図2　頂点併合操作後の拘束ネットワーク

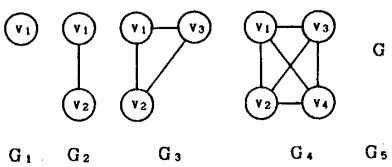

図3 図1に示した拘束ネットワークのinvasionの例

Improvement and System Integration of Inexact Consistent Labeling Problems

Yoshikazu Matsuo,Seiichi Nishihara,Katsuo Ikeda
University of Tsukuba

頂点併合の際、まず、併合したい頂点間で共通なユニットを探しだす。次に、'共通ユニット'に対応したラベルに対して全てのラベルの組み合わせを考慮し、値が許容度を越えないラベル組だけを残す。ここで、共通ユニットの数が多い程考慮するラベルは減少することが考えられる。また、invasionのより早い段階にこうした共通ユニットが集中することによって考慮すべきラベル組の増大を防ぐことができる。一般に、一つの最終解を求める場合、見込みのありそうな方向へ探索を進めていくheuristics('縦形heuristics')を用いる。しかし、本問題では、全ての解を求める必要があるので、よりきびしい拘束条件を先に適用し、なるべく早い時点で多くの候補を除外するheuristics('横形heuristics')を用いる。つまり、invasionのより早い段階に、より多くの共通ユニットを集中させることによって、そのinvasionは準最適となる。

3. 実験

inexact-CLPの生成は、まず、ランダム変数の基数を与え、|U|(ユニット数),|L|(ラベル数),|T|(タップル数)とαを入力して行った。ここで、αはランダムに選出したラベル組が、w_iに含まれる確率(%)である。次に、|U|と|T|により、各t_iに含まれるユニットを選出する。また、各t_iの次元、すなわち、$|s(t_i)|$は3とした。ラベル組は、αを基に生成し、そのラベル組のとる値は、[0,99]の範囲でランダムに選出した。次に、基数を変えて4回実験を行い、その平均をとった。また、inexact-CLPを解くアルゴリズムとして、以下の4つを用いた。

アルゴリズム1(BT):最も単純な解法。backtrackを伴う深さ優先探索である。

アルゴリズム2(BTFC):アルゴリズム1に発見的手法である先読み(forward checking)[1]を導入した方法。

アルゴリズム3(DP):先に[2]提案した解法。invasionを用い、DPの手法を援用した解法である。

アルゴリズム4(DPFC):アルゴリズム3に先読みを導入した方法[3]。

アルゴリズム3と4は、与えるinvasionによって計算時間、および、使用領域が大きく左右されるので、あらかじめ準最適invasionを求めておき、それを用いることとした。アルゴリズム1と2は、各ユニットにあるラベルを与え、その都度整合性を判定していくので、|U|や|L|が増大すると、計算時間が膨大になることが予想される。一方、アルゴリズム3や4は、|T|回のinvasionに従って頂点が併合されるが、冗長ユニットを取り除いていくので、|U|による影響を極力押えることが可能である。

|U|を変化させ、計算時間を計測したのが、図4である。ここで、ε_0のかわりに$\beta=\varepsilon_0/|T|$を用いた。これは、|T|の値の影響を除くためである。アルゴリズム4,3,2,1の順に良い結果が得られているが、その理由として、|U|による影響と、先読みの効果があげられる。つまり、深さ優先探索(BT,BTFC)では、|U|が、探索木の深さに対応しているので、|U|が増大すると計算時間も増大する。一方、DP的手法(DP,DPFC)の場合、|U|による影響は小であるので、|U|が大きくなる程、効果は顕著になる様である。また、先読みを行うことによって、考慮すべきラベル組の個数が減少し、計算時間が縮小された。ここでは、invasionを求めるための時間は含まれていないので、実際は、DPとDPFCの計算時間に、invasionを求める時間を加算する必要がある。残る問題は、準最適invasionアルゴリズムを完成させることであるが、invasionの各段階において、最も共通ユニットの多い頂点を選択していけばよいので、計算時間に対する影響は小であると考えられる。

図5は、|U|=|L|=|T|とし、問題のサイズの増大による、DPとDPFCに対する計算時間と展開したノード数の変化を調べたものであるが、ともにDPFCが優れていることがわかる。特に、展開したノード数は、DPFCが圧倒的に少ないが、その要因としては、先読みの効果があげられる。しかも、問題のサイズが増大する程この差は大きくなり、計算時間に大きな影響を与えている。一般に、問題のサイズが大きくなると、使用領域が増大するため、これを押える必要があるが、領域抑制に加えて、計算時間も押えることが可能なDPFCが、最も適当なアルゴリズムであることが実証された。

4. 結び

問題のサイズが、大きくなればなる程、DPFCによって良い結果がもたらされることがわかった。また、拘束ネットワークの形が細長い場合、すなわち、ユニットの局所性が高い程DPFCが適している。反対に、|U|が小さく、|T|が大きい場合は、拘束がきつく、ユニットの局所性が低いので、BTの方が適している。以上より、inexact-CLPのシステムを構築し、また、準最適invasionを求めるプログラムを完成させることが、今後の課題として挙げられる。

[文献] 1) Shapiro,L.G.&Haralick,R.M.,IEEE Tr.PAMI,3,5(1981).
2) 塩澤,西原,池田,情報処理学会第31回全国大会,4P-3(1985).
3) 松尾,塩澤,西原,池田,情報処理学会第32回全国大会,4M-4(1986).
4) Seidel,R.,Proc.IJCAI,7th(1981).

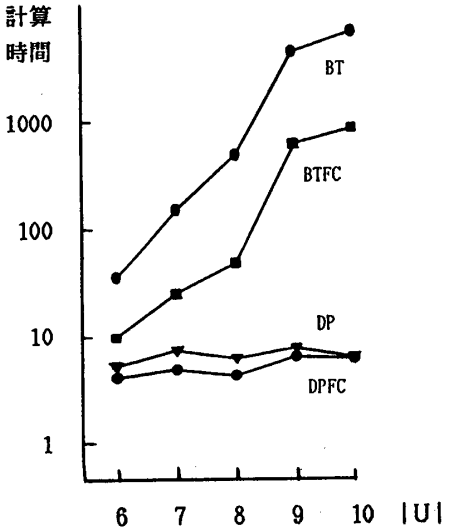

図4 |L|=10,|T|=10,α=10,β=20
の時の計算時間

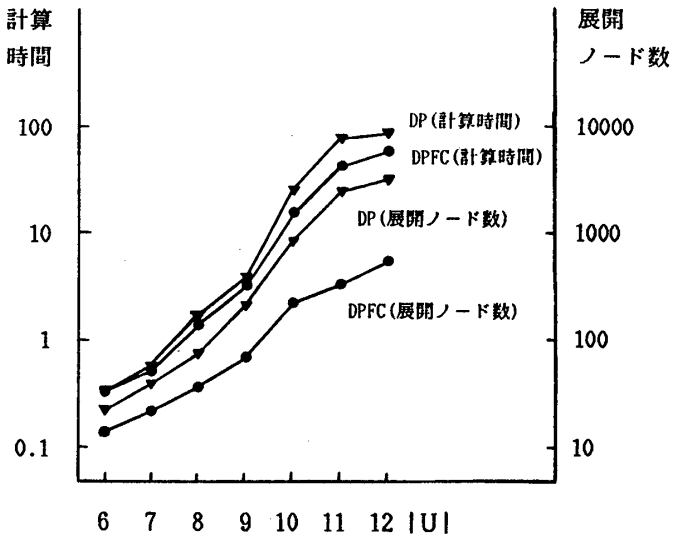

図5 |U|=|L|=|T|,α=10,β=20
の時の計算時間と展開ノード数