7Ａ－４　制約グラフの局所性を用いた併合法の並列化について

内野寛治　窪田信一郎　李江洪　山下正吾　西原清一
筑波大学　電子・情報工学系

1　はじめに

制約充足問題（以下CSP）は，人工知能を始めとする各分野で多く見受けられる問題の総称である．CSPは問題の対象を「構成要素」に分解し，構成要素間に存在する局所的な条件，すなわち「制約」を記述することで対象全体の性質が表現できるような問題である．このような問題はパターン解析や知識処理など人工知能の各分野，特に原理的に組み合せ探索を含むような問題に多く適用できる．

CSPの解法の１つとして制約の整合化を繰り返し行なう併合法がある．本研究ではCSPをグラフ表現し，そのグラフの構造の局所性に注目し，併合を並列に行なうときに適した併合順序を考察するものである．

2　諸定義

2.1　CSPの定義

CSPは４つ組(U,L,T,R)で定義される．U={1,..,M}はユニットの集合で,各要素は問題対象の構成要素に対応する．Lはラベル集合で，各要素はユニットに与えるべき解釈や値の候補を表す．Tはユニット多項組の集合で‘ユニット制約関係’と言う．そしてそのそれぞれのユニット組に対して可能な局所的解釈が$R=\{R_1,...,R_{|T|}\}$で与えられ，各R_iを‘ラベル制約関係’と言う．組(t_i,R_i)を‘制約条件ペア’という．CSPを解くとは全ユニット(1,...,M)に対し，全ての制約条件ペアを満たすラベル組$(l_1,...,l_M)$を求める操作である．CSPの例を図１に示す[3]．

U={1,...,4}, L={a,b,c,d,e},
T={t1,...,t4}　t1=(1,2), t2=(1,3), t3=(2,3,4), t4=(1,4),
R={R1,...,R4}
　R1={(a,b),(a,e)},
　R2={(a,c),(a,e),(e,b)},
　R3={(b,a,e),(b,c,d),(e,b,e)},
　R4={(a,d),(a,e),(b,e)}.

全ユニット：(1,2,3,4)
解　　　　：(a,b,c,d)

図１　CSPの例

2.2　CSPのグラフ表現

CSPをグラフによって等価表現できる．本稿では，制約条件ペアをグラフの頂点，２つの制約条件ペアに共通して含まれるユニットを辺に対応させ表現した制約グラフを用いる．

On Parallelism of Merge Method by Using Locality of Constraint Graphs
Kanji UCHINO, Shinichirou KUBOTA, Jiang-Hong LI,
Shogo YAMASHITA, Seiichi NISHIHARA
Inst. Inf. Sci. & Electr., University of TSUKUBA

先の例を制約グラフで表現したものを図２に示す．

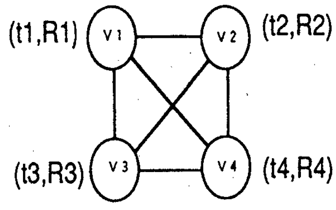

図2　制約グラフ

2.3　併合法と併合系列

CSPの解法として‘制約グラフ中の２つの頂点をまとめて、新たに１つの頂点で置き換える操作’すなわち頂点併合操作を順次繰り返していき最終的に１つの頂点に縮退させる方法がある．これを併合法と呼ぶ．例の場合，V_2とV_3とを併合させると新たに得られる頂点V_{23}の制約条件ペアは，t_{23}=(1,2,3,4), R_{23}={(a,b,c,d),(e,e,b,c)}となる．制約グラフの頂点に繰り返し併合操作を施し，最終的に１つの頂点になったときの制約条件ペアが最終解を与える．併合法の処理に要する‘計算コスト’を

<u>併合する２頂点のＲのサイズの積の総和</u>

で定義する．計算コストは頂点の併合順序に大きく左右される．これは併合する頂点によって中間解の個数が異なるためである．本稿ではこの併合順序を‘併合系列’と呼ぶ[2,3]．

3　併合法の並列化

3.1　構造レベルの並列化

併合法を並列に処理する方法として，制約グラフ中の複数部分を独立に頂点併合を行なう構造レベルの並列化が考えられる．先の例で,V_1,V_2とV_3,V_4の併合操作を２つのプロセッサで並列に行なう操作がそれにあたる．このような考えは分割統治法においてよくみられる．構造レベルの並列化の問題点はCSPを分割することで中間解の制約が少なくなり，その結果中間解の個数を抑制する効果が期待できなくなる点である．場合によっては逐次的に解いた方が並列に解くどんな場合よりコストを少なく処理できるものもある[4]．

3.2　併合系列の表記

本稿では併合系列の表記としてカッコ表現を導入する．

(ａ ｂ)

はａとｂを併合し新たに１つの頂点で置き換える操作を表す．ａ，ｂはそれぞれ頂点又はカッコ表現を表す．

ａ，ｂの少なくとも一つがカッコ表現の場合は，ａ，ｂを直ちに併合することはできない．その場合，ａ，ｂ共にそれぞれ一つの頂点に縮退した後，最終的に併合が施される．従って，一般的にカッコ表現においては，最も深いレベルのカッコが先に併合される必要がある．

図２中の頂点をV_1-V_2→V_{12}, V_{12}-V_3→V_{123}, V_{123}-V_4→V_{1234}の順に併合処理を行なうような併合系列は上の定義から次のように表される．

$$(((V_1\ V_2)\ V_3)\ V_4) \quad ...(1)$$

本稿ではこのような表現を省略して,単に

$$(V_1\ V_2\ V_3\ V_4) \quad ...(2)$$

と表記する場合もある．次に

$$((V_1\ V_2)\ (V_3\ V_4)) \quad ...(3)$$

は，与えられたCSPを部分問題に分割することを表す．すなわち，頂点を $\{V_1,V_2\}$，$\{V_3,V_4\}$ の２グループに分け，それぞれが一つの頂点に縮退した後，それらを併合することを表す．これら２種類の系列をシングルプロセッサ，マルチプロセッサで解く場合の処理は表１のようにまとめられる．

アーキテクチャ ＼ 併合系列	CSPを分割しない	CSPを分割する
シングルプロセッサ	逐次処理	準並列処理
マルチプロセッサ	なし	並列処理

表１　併合処理の分類

3.3　併合処理の計算コスト

表１で分類した併合処理の計算コストを考察する．(1),(3)の併合系列をCSPを分割，CSPを非分割の例として考える．併合系列(1)をシングルプロセッサで処理したときの計算コストをCs，併合系列(3)をシングルプロセッサで処理したときをCp'，併合系列(3)をマルチプロセッサで処理したときをCpとするとそれぞれの処理の計算コストは次のようになる．
（但しmc(x y)はxとyを併合するときの計算コストを与えるものとする）

$$Cs = mc(V_1\ V_2)+mc(V_{12}\ V_3)+mc(V_{123}\ V_4)$$
$$Cp' = mc(V_1\ V_2)+mc(V_3\ V_4)+mc(V_{12}\ V_{34})$$
$$Cp = \max(mc(V_1\ V_2),mc(V_3\ V_4))+mc(V_{12}\ V_{34})$$

3.4　併合系列の諸性質

表１で示したように併合系列はCSPを分割しないもの(1)と分割するもの(3)に大きく分けられる．シングルプロセッサでの処理を仮定した場合，CSPを分割しない併合系列の中で最少の計算コストを与える系列の処理を‘最適逐次処理’，CSPを分割した併合系列の中で最小の併合コストを与える系列の処理を‘最適準並列処理’と呼ぶことにする．この２つの処理には次のような性質が成立する．（証明は省略）

> 性質1.　計算コストについて
>
> <u>最適逐次処理≧最適準並列処理</u>
>
> となるCSPが存在する

この性質はCSPが本質的に分割に向くものと，向かないものに分けられることを意味している．

次に最適準並列処理をマルチプロセッサで行なう場合を考える．これを‘最適並列処理’と呼ぶ．このとき次のような性質と命題が成立する．

> 性質2.　計算コストについて
>
> <u>最適逐次処理≦最適並列処理</u>
>
> となるCSPが存在する

これは[4]で報告されており，図１のCSPはその例になっている．（証明は省略）

> 命題3.　同じ併合系列を処理する場合の
> 計算コストにおいて
>
> <u>準並列処理＞並列処理</u>
>
> である

3.5 局所性の問題点

制約グラフの局所性とはその部分がグラフ全体から見てどの程度独立した存在であるかをみるものである．併合法を分割統治法により並列的に行なうときの効率の評価には，制約グラフの局所性の定量化が必要である．また，グラフの構造でなく，頂点間の共通ユニットの個数やラベル制約関係に含まれる構造等も含めた局所性の評価が必要となる．

4　おわりに

本稿ではCSPの解法の１つである併合法について，新たな併合系列の表記とそれに伴う併合処理の諸性質，そしてそれら表記を逐次，並列処理した場合の効率について考察した．またCSPを並列に処理するためにはCSPを部分問題に分割する必要があるが，分割を行なう際の指標として制約グラフの局所性が考えられる．その局所性の問題点についての考察も行なった．

今後の課題としては，局所性の評価を明らかにしCSPを本質的に逐次的な問題と並列的な問題とに分けることが考えられる．局所性の評価については併合法の処理コストを解析的に求めるアプローチも有効であると思われる[1]．

参考文献

[1]窪田、内野、李、山下、西原:併合法による制約充足の並列効果について，第46回情報処理学会全国大会(1993).
[2]塩澤,西原,池田:拘束条件の構造を考慮した整合ラベリング問題の解法,情報論文誌 ,27,10,927-935(1986).
[3]西原:制約充足問題の高速解法，知識ベースシステムにおける高速推論技術チュートリアル資料、情処学会(1992).
[4]西原、松尾:整合ラベリング問題における併合解法の並列化について、人工知能学会誌,6,1,pp.124-128(1991).