7A−1

併合法による制約充足の並列化効果について*

窪田信一郎　内野寛治　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と略記）[1, 5] が解を有するかどうかを決定する問題は、NP完全問題のクラスに属しており、その意味で効率のよい汎用アルゴリズムは存在しないと予想される [1]。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。

組合せ探索問題においては、並列処理の研究が盛んに行われている。分割統治を応用した並列化はその例である。CSPの解法の1つとして、制約間の関係を表す制約グラフ上で、制約の整合化を繰り返す方法、すなわち併合法があるが、併合法も分割統治を応用して並列処理できる。しかし、分割統治による並列化は、CSPにおいては必ずしも有効とは限らないことが実験によって明らかにされている。本稿では、CSPの解法の1つである併合法を、分割統治を応用して並列化する効果について考察する。

2 CSPと併合法

2.1 CSPの定義と制約グラフ

CSPは4つ組 (U, L, T, R) で定義される [1, 5]。$U = \{1, \cdots, M\}$ はユニットの集合で、各要素（'ユニット'）は問題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べき解釈や値の候補を表す。$T = \{t_1, \cdots, t_N\}$ はユニットの多項組（'ユニット組'）の集合で、'ユニット制約関係' と言う。各ユニット組に対応する許されるラベル組の集合は、'ラベル制約関係' $R = \{R_1, \cdots, R_N\}$ で具体的に与える。CSPを解くとは、全ユニット $(1, \cdots, M)$ に対する、全ての制約条件を一斉に満たすラベル組 $\lambda = (\ell_1, \cdots, \ell_M)$ を求める操作である。

任意のCSPは'制約グラフ'によって等価表現される [1, 2]。定義の詳細は文献に譲るが、グラフの頂点は制約条件ペアを表し、辺はその両端頂点に対応する制約条件ペアのユニット組の共通ユニット集合を表す。[5] のCSPの例を制約グラフで等価表現したものを図1に示す。

2.2 併合法

併合法は、2つの頂点を局所的に整合性を保ちつつ1つの頂点にまとめる操作、すなわち頂点併合操作を順次繰り返して行き、最終的に1つの頂点に縮退させる方法である [1]。ここで、最終的に得られた頂点1個だけのグラフはもとのCSPの最終解を与える。併合法は、併合する頂点の順序によって中間解の個数（途中の制約グラフにおいて頂点 v_i に付随する R_i のサイズ）が影響を受け、処理時間や作業領域に大きく影響する。CSPにおいては、最終解の個数とは別に、中間解の個数が組合せ的爆発を起こすことがある。このような問題を回避するための、最適、または、準最適な併合順序を求める方法が提案されている [2, 4]。

3 併合法の並列化

併合法を並列に処理する方法としては、制約グラフの構造に依存する構造レベルの並列化と、頂点併合操作それ自身を並列に処理する併合レベルの並列化とがある。並列粒度は前者が粗く、後者が細かい [3]。本稿では、前者の構造レベルの並列化について考察する。

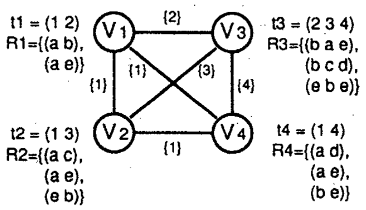

図 1: 制約グラフ

3.1 構造レベルの並列化

制約グラフのある局所連結部分（辺）において頂点併合操作を行いつつあるとき、グラフの他の部分はそれとは独立に並行して頂点併合を施すことができる。このような考えに基づく構造レベルの並列化は、分割統治法においてよく見られる。並列処理においても、併合する頂点の選び方によって中間解の個数が影響を受け、その結果が処理効率にも影響する。そこで、併合する頂点を選ぶ際には、まずユニット組 t_i が t_j を含むような頂点ペア v_i, v_j を優先する。このような頂点を併合する操作を縮退操作 [4] と言い、この操作では中間解は増加しないことが保証される。縮退操作がない場合には、併合後の中間解の個数ができるだけ小さくなるような頂点ペアを求める。併合後の中間解の個数は予測値として、$k_{ij} = \frac{|R_i| \cdot |R_j|}{|L|^{\alpha_{ij}}}$ を用いる。α_{ij} は t_i と t_j に共通して含まれるユニットの個数であり、共通指数 [4] と言う。一般に、この共通指数が大きい程、指数オーダーで中間解の個数が抑制される。以上のような処理方法は、恣意的にいわば目先の最適頂点ペアに併合操作を施すという方針をとっており、greedy法（欲張り法）の並列化と言える。

3.2 並列処理の実験結果

松尾らは併合法の2つのレベルの並列化の効率を、実験によって評価している [3]。これによると、併合レベルの並列化は、実行時間の短縮に非常に有効であるが、構造レベルの並列化では、場合によっては、駆動プロセッサ数の増加と共に、大幅な効率低下を招いている。これは、ユニット制約関係 T のサイズ（制約グラフの頂点数に等しい）やユニット総数が小さいほど、さらに、R_i の平均サイズが大きいほど助長される傾向がある。これは、構造レベルの並列化では、局所的な制約条件を分断することにより、中間解への制約が少なくなり、中間解の増大を引き起こしかねないことを示している。また、実験により、並列処理に完全に適さないような種類のCSPが存在することが明らかとなっている。図1はその例である。

4 併合法の定量評価

4.1 計算量評価

併合操作自体の計算コストは、併合しようとする2頂点の中間解のサイズの積で評価される。したがって、併合法の逐次処理全体の計算コストはこれらのコストの総和で表せ、並列処理全体の計算コストは各併合レベルでの最大コストの総和で表せる。また、

*On Parallelism of Merge Method for Constraint Satisfaction
Shinichiro KUBOTA, Kanji UCHINO, Jiang-Hong LI,
Shogo YAMASHITA, Seiichi NISHIHARA
Inst. Inf. Sci. & Electr., University of TSUKUBA

R_i 中に現れるラベルが均一であると仮定すると、併合後の中間解のサイズが予測できる。ラベル制約関係 R_1,R_2 をそれぞれ持つ2頂点を併合すると、併合後の中間解の個数の予測値は、

$$|R_{12}| = \frac{|R_1| \cdot |R_2|}{|L|^{(1-\beta)\cdot\alpha_{12}}} \qquad (1)$$

ただし、α_{12} は2頂点の共通指数である。また、β は $0 \le \beta \le 1$ であり、R_1 と R_2 の共通ユニットに対するラベルの相関を表している。以上を考慮すると、制約グラフが密（辺の数が多い）であり、解の個数が経験的にわかっている場合、頂点数 $n (n \geq 2)$ の制約グラフを逐次処理, 並列処理で併合する時の計算コスト C_s,C_p は、

$$C_s(n) = K^2 + \frac{K^3}{\ell^{M_1}} + \frac{K^4}{\ell^{M_1+M_2}} + \cdots + \frac{K^n}{\ell^{M_1+M_2+\cdots+M_{n-2}}} \qquad (2)$$

$$C_p(n) = \begin{cases} K^2 & (\text{for } m = 1) \\ K^2 + \frac{K^4}{\ell^{2M_1}} + \frac{K^8}{\ell^{4M_1+2M_4}} + \cdots \\ \quad + \frac{K^{2^m}}{\ell^{2^{m-1}M_1+2^{m-2}M_4+\cdots+2M_{4m-2}}} & (\text{for } m \geq 2) \end{cases} \qquad (3)$$

（ただし、$\ell = |L|^{1-\beta}$、並列処理においては $n = 2^m$, $m = 1, 2, \cdots$）
と表せる。ただし、各 R_i のサイズは全て一定値 K とし、並列処理の場合、2^{m-1} 個のプロセッサが利用可能であるものと仮定している。また、β は解の個数の予測値

$$N = \frac{K^4}{|L|^{(1-\beta)(M_1+M_2+\cdots+M_{n-1})}} \qquad (4)$$

から逆算して求める。M_p は、制約グラフから任意に p 種類の2頂点を取り出した時に、何種類の共通ユニットが現れるかの期待値を表す。図1のような制約グラフをもつCSPを解いたときの計算コストの評価結果を図2に示す。

4.2 並列化による効果

前節では、併合法の計算コストを R_i のサイズ,L のサイズと期待値 M_p で表した。M_p は定義より、p の増加関数であり、T の構造に依存する。すなわち、同じ制約グラフでも T の構造により計算コストの取り方が異なることがありうる。式(2),(3)の値を比較すると、図2のように R_i のサイズが大きいほど、並列処理の効率が低下している。また、図3のように、pについて増加量の小さい M_p ほど、並列処理の効率が低下している。M_p の増加量が小さい事は、ユニットの数に対して辺の数が多い事を意味する。この場合、各ユニットに対する部分問題は制約グラフ全体に薄く広く分散している事になり、並列化した時にこれらの部分問題が分断され、中間解への制約条件がより少なくなり、中間解の爆発を招くと考えられる。また、T のサイズが小さいほど、相対的に制約グラフの辺の数が多くなり、また、ユニットの数が少ないほど、同じ共通ユニットを持つ辺が多く存在することになり、規模の大きな部分問題が分散していることになる（各ユニットに関する部分問題は完全グラフになることに注意）。この考察は先の実験結果と結果が一致していることがわかる。

ここで、一見すると M_p の増加量が小さい事は、制約グラフが疎（辺の数が少ない）であることを意味し、容易に並列化可能であり、並列処理の効率が上がるように思われる[5]。しかし、ここでの並列処理の計算コストは、処理途中の頂点数の $\frac{1}{2}$ の個数のプロセッサを駆動させることを仮定している。したがって、制約グラフが疎である場合は、辺の引かれていない2頂点を併合することがあり、中間解の個数が爆発し、処理効率が低下する。このことにより、制約グラフを並列処理する時には、処理に参加するプロセッサの個数が、効率に大きく影響するものと考えられる。

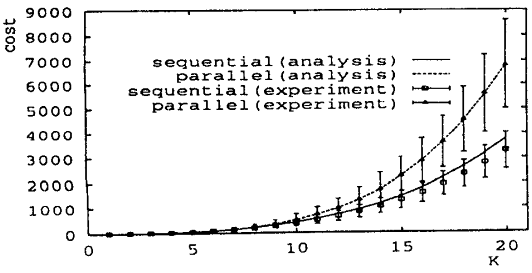

図 2: 計算コストの評価結果

5 おわりに

本稿では、CSPにおける制約充足法の1つ、併合法の並列処理の定量評価をし、並列化の効果について考察した。これは、CSPを本質的に逐次的な問題と並列化可能な問題とに分類するための1つのアプローチである。定量評価においては、図2を見ると、計算コストの実測値にかなりの幅があることがわかる。これは、併合処理の計算コストには制約グラフの構造以外のもの（R_i の構造etc.）も大きく関わっているものと考えることができる。今後、これら計算効率の要因を見つけ出すことが課題となる。

参考文献

[1] 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, 情報処理学会 (1992)

[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決, 情報処理学会論文誌, Vol.27, No.10, pp.927-935 (1986)

[3] 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, Vol.6, No.1, pp.124-128 (1991)

[4] 西原, 松尾, 池田: 概整合ラベリング問題における併合法の最適化と効率評価, 人工知能学会誌, Vol.3, No.2, pp.196-205 (1988)

[5] 内野, 窪田, 李, 山下, 西原: 制約グラフの局所性を用いた併合法の並列化について, 第46回情報処理学会全国大会 (1993)

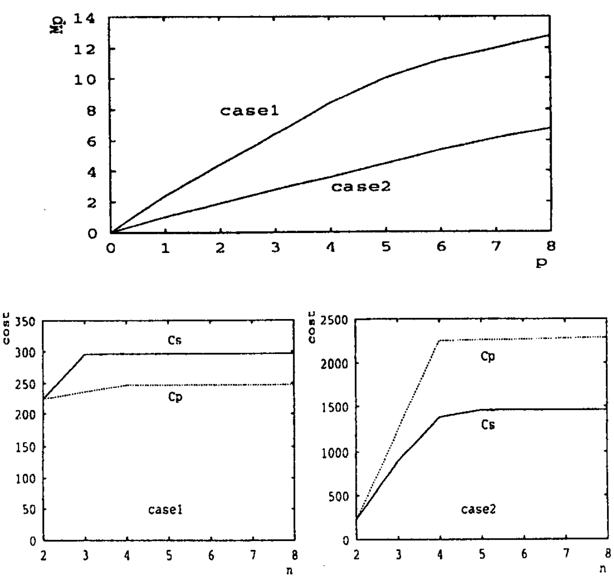

図 3: M_p と計算コストとの関係