6F−5

動的制約充足問題のための制約維持機構について

野中 哲、杉本 勉、大濱 寛樹
(NTT データ通信株式会社)

1 はじめに

画像処理や人工知能の分野をはじめとするさまざまな分野で扱われる、古典的かつ重要な問題の1つに制約充足問題 (Constraint Satisfaction Problem) がある。この問題は、NP 完全な組合せ探索問題であるために、効率的な汎用アルゴリズムは存在せず、個々の問題の特徴を生かした解法を得る必要がある。このような制約充足問題のうち同じ制約 (式) が何度も使われたり、既に使われている制約への制約の追加・変更が頻繁に起こるといった動的に制約が変化する問題に対しては、制約の状態や一度評価した制約を保持しておくことが有効であると考えられる。本稿は、このような動的制約充足問題を解決するために、制約状態や制約 (式) を保持したライブラリ (「制約ライブラリ」と呼ぶ) を用いて問題解決を行う、制約維持機構について述べる。

2 動的制約充足問題

制約充足問題とは、変数の有限集合、および各変数にとりうる値を離散値の有限集合として与えた場合に、すべての制約を満足するような各変数の値を求める問題である。

- 変数： $V = V_1, ..., V_n$
- 変数の領域：各変数 V_i は、離散値の有限集合 D_i から値を選択する
- 変数間の制約:$r_l(V_i, ..., V_j) \subseteq D_i \times ... \times D_j$
  $1 \le i, j \le n$
- 変数間の制約の候補： $r_l(V_i, ..., V_j)$ に対して各変数 $V_i, ..., V_j$ が取り得る解の候補の集合

具体例としては、上記のような定義を用いた場合に、変数間の制約の候補として r_1(a,b,c);r_1(a,c,c);r_1(b,b,a); r_2(c,a);r_2(c,b);r_3(b,c) を与えた場合、変数間制約の組 $r_1(V_1, V_2, V_3), r_2(V_3, V_4), r_3(V_2, V_3)$ を満たす $[V_1, V_2, V_3, V_4]$ の組を求めるといった問題である。(解は $[V_1, V_2, V_3, V_4]$ = [a,b,c,a]; [a,b,c,b] である)

これら従来の制約充足問題に対して以下の2つの変化を考慮に入れた問題を、ここでは動的制約充足問題と呼ぶものとする。

1. 求める変数間制約の組の変化
   例では $r_1(V_1', V_2', V_3'), r_2(V_3', V_1')$ を 満たす変数の組を新たに求める場合などを指す
2. 変数間の制約の候補の変化
   例では $r_3(c, c)$ が変数間の制約の候補として付け加わることなどを指す

前者は、変数間の制約の組が多少異なる場合や、変数間の制約の中の変数の順序が異なる新たな問題を頻繁に解く場合であり、定性推論 [1] やタイプ推論などの制約充足の処理で現れる問題である。

後者は、変数間の制約の候補を新たに付け加えたり、あるいは以前からあった制約の候補を取り除く場合である。ただし、ここでの制約候補の変化は、問題を解いている最中の変化ということではなく、一度問題を解いた後に変化する場合である。

したがって、本稿において「動的」とは、インクリメンタルに制約が加わる場合や、制約の候補の変化を指しており、[2] における動的制約充足問題の定義とは異なる。

3 制約の維持

求める変数間制約の組の変化に対応する動的な制約充足問題を解く場合に重要となるのは、変数間制約の集合における変数と変数間制約の依存関係である。これは、先の例題を用いて説明すると次のようなことである。

まず、$r_1(V_1, V_2, V_3), r_2(V_3, V_4), r_3(V_2, V_3)$ を求める際の $r_1(V_1, V_2, V_3), r_2(V_3, V_4)$ 部分に着目すると、これは r_1 の第3引数と r_2 の第1引数が同じであることを示している。一方、その後に解く問題の $r_1(V_1', V_2', V_3')$, $r_2(V_3', V_1')$ の場合は、r_1 の第3引数と r_2 の第1引数が同じという条件に加え、r_1 の第1引数と r_2 の第2引数が同じという条件も加わったと考えられる。

これらは、ある条件の下ではある制約の集合が成り立ち、他の条件ではまたある制約の集合が成り立つということを保持し続けることであり、動的な制約充足問題では、ATMS の多重コンテキストのような機能を持たせる必要があることを意味している。そこで、こうした変数の依存関係を含めての制約の状態、変数間制約およびその候補の集合を保持するために開発したのが制約ライブラリを中心とする制約維持機構である。

4 制約維持機構の概要

制約維持機構の構成を図1に示す。

Constraint Maintenance System for Dynamic Constraint Satisfaction Problems
Satoru NONAKA,Tsutomu SUGIMOTO,Hiroki OOHAMA
NTT DATA COMMUNICATIONS SYSTEMS Corp.

制約維持機構は、推論エンジンなどから変数間制約と変数間制約の候補の組が与えられた場合、制約評価器によって変数間制約の候補の構造を分析し、制約ネットワークの形でネットワークライブラリに保持する。また、変数間制約の集合である制約充足問題が与えられた場合、制約評価器は変数と変数間制約の依存関係を分析しネットワークライブラリのネットワーク構造を変化させる。その後、ネットワーク構造から制約充足問題の全解を収集し、推論エンジンに解を返す。シンボリックライブラリは、変数間制約と変数間制約候補の組や、得られた解をリスト形式で保持することで、解の出力を高速に行うためにある。現在、これらは SICStus-Prolog を用いて開発中である。ただし、ネットワークライブラリ部分は C 言語で記述している。

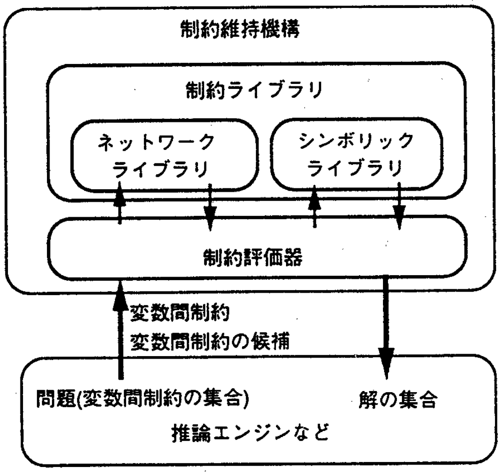

図 1: 制約維持機構の構成図

5 ネットワークライブラリにおける制約ネットワーク

制約状態や変数間制約、その候補などは制約ネットワークと呼ばれるネットワーク構造でネットワークライブラリ内に蓄積されている。制約ネットワークにおけるノードは

- 変数間制約の名前に対応するノード (制約ノード)
- 変数の引数の順序に対応するノード (引数ノード)
- 変数間制約の候補値に対応するノード (候補値ノード)

から成り立っている。

実際の処理の流れの概要は以下のようになる。

変数間制約の候補が与えられると、制約ノードと引数ノードから候補値ノードに向かってアークを引く。制約ノードと候補値ノードを結ぶアークは、出てきた候補値の順に番号を付ける。(図 2 では変数間の制約の候補として r_1(a,c);r_1(b,c);r_2(c,b);r_2(c,c) を与えた場合である。変数間制約の候補 $r_1(a,c)$ が与えられると、制約ノード r1 から候補値ノード a,c に 1 という番号付きのアークが引かれると同時に、引数ノード r1(1) と候補値ノード a、引数ノード r1(2) と候補値ノード c が結ばれる)

変数間制約の集合により問題が与えられると、変数を共有している引数ノードに結合している候補値ノードから同一の候補値ノードを探し、そのノード間を結合する。引数ノードには変数が同じという条件をすでに調べたという意味のマークを付ける。(問題 $r_1(V_1,V_2)$,$r_2(V_2,V_3)$ を満たす解を求める場合、r1(2) ノードと r2(1) ノードからの同一の候補値ノード c を探し、その間を太実線で結んでいる。引数ノード r1(2) と r2(1) の下の {} の中が、マークである。次に、$r_1(V_1',V_2')$,$r_2(V_2',V_1')$ を満たす解を求める場合には、r1(2) ノードのマークに r2(1) のマークがあるので r1(1) ノードと r2(2) ノードに結合した候補値ノード間結合 (太点線) とマーク付けを行う)

問題の解を出力する際は、変数が等しいという条件を満たす候補値間アークをマークを利用して探し、このアークに結合した結合に矛盾のない番号付きのアークをすべて探索することにより解を得る。(先の問題 $r_1(V_1',V_2')$, $r_2(V_2',V_1')$ を満たす解を求める場合は、候補値ノード間結合の太実線、太点線に結合した番号付きアーク 2,3 を求めることで、$[V_1',V_2']=[b,c]$ を得る)

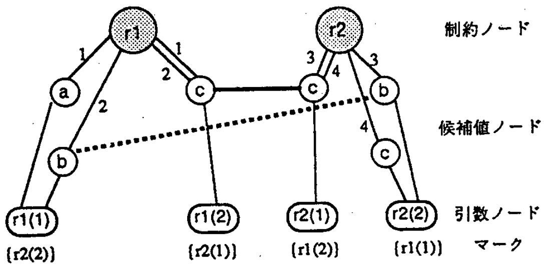

図 2: ネットワークライブラリ内での制約ネットワーク

変数間制約の候補の変化に対しては、候補値ノードにつながる結合を変数間の条件を考慮して切断、結合を行うことで対処することができる。

6 おわりに

以上、動的制約充足問題を解くための制約維持機構について述べた。本方式は動的制約充足問題に対応するため、ネットワーク構造をもった制約ライブラリを用いて制約の状態を保持することで、以前に評価した制約を有効に利用しようというものである。今後、当機構を定性推論へ適用する予定である。

参考文献

[1] de Kleer J and Brown,J.S. “A qualitatibe physics based on confluences,Artificial Intelligence,24,pp.7-83,1984

[2] Sanjay Mittal and Brian Falkenhainer “Dynamic Constraint Satisfaction Problems”, Proc,Eighth Nat.Conf.on Artificial Intelligence (AAAI),1990