

ブール代数を用いた制約充足問題の定式化とその解法について

永井 保夫

(株) 東芝 情報通信システム技術研究所

1 はじめに

制約充足は人工知能や画像理解の分野をはじめ、グラフ同形判定、グラフラベリング、巡回セールスマン問題などのグラフの問題やパズルなどの探索問題、いわゆる組み合わせ問題を対象として研究が行われている。このような制約充足問題の代表的な解法としては、探索法や局所近似を用いる方法がある。本稿では、このような解法とは異なるあらたな代数的アプローチについて報告する。本アプローチは制約充足問題をブール代数を用いて定式化し、ブール方程式を解くことにより充足処理をおこなうものである。まず、制約充足問題について触れ、ブール代数による定式化を述べる。次に、ブール代数による制約充足の解法を具体的な例題を用いて説明する。

2 制約充足問題とブール代数による定式化

2.1 制約充足問題

制約充足問題(CSP)とは、次のような変数の有限集合および各変数に対して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に各変数に対して値を求める問題である。制約とは適用対象の構成要素およびその属性間で成立する関係を宣言的に記述したものである。制約充足問題では、探索の効率化を目的としたバックトラック探索アルゴリズムの性能を改善する研究がおこなわれている[1,3]。

- 変数: $V = \{v_1, \dots, v_n\}$
- ドメイン: 各変数 v_i は離散値の有限集合 $D_i = \{d_1, \dots, d_m\}$ $i = 1, \dots, n$ から値 d_m を選択する。
- 制約: 集合 $C = \{c_1, \dots, c_l\}$ where $c_n \equiv (v_1, v_2, \dots, v_n) \subseteq D_1 \times D_2 \times \dots \times D_n$

制約充足問題では、制約は制約ネットワークとして静的な問題の記述に適した知識表現をとる場合が多い。その場合、制約ネットワークはグラフによってモデル化され、制約グラフとして表現される。ネットワークを用いたアプローチの利点は、知識の構造化が容易であり、知識の無矛盾性を効率的に管理できることである。制約ネットワークは、 n 個の変数 $v_1, v_2, \dots, v_n$ 、各変数に対するドメイン $D_1, D_2, \dots, D_n$ および制約 (n 項関係) 集合をネットワークとして表現したものである。 n 個の変数 $v_1, v_2, \dots, v_n$ に対する制約 $c(v_1, v_2, \dots, v_n)$ とは、 n 個の集合 $\{D_1, D_2, \dots, D_n\}$ に対して成り立つ関係 (つまり n 項関係) を表し、無矛盾な変数値の直積 $D_1 \times D_2 \times \dots \times D_n$ の部分集合である。例えば、2項制約ネットワークはすべての制約が2項関係である (高々2個の変数からなる) ネットワークである。CSPにおける制約充足とは、制約ネットワークを解くことであり、すべての制約が満足されるようにネットワーク中のすべての変数に対する値を求めることに相当し、単解または全解を求めることである。

2.2 ブール代数による制約充足問題の定式化検討

ブール代数とは、0と1のみからなるブール集合 B 上で交換律、結合律、分配律、べき等律、補元律、対合律、普遍境界の7つの性質を満足する代数系 $\langle B, \wedge, \vee, \neg, 0, 1 \rangle$ である[5]。以下では、ブール代数を用いたCSPの定式化について示す。変数集合 $V = \{v_1, \dots, v_n\}$ 、各変数 v_i がとりうる離散値の有限集合 $D_i = \{d_1, \dots, d_m\}$ 、 $i = 1, \dots, n$ とし、変数 v_i に対して

ドメイン D_i の要素 d_j を割り当てることを変数 x_{ij} によりあらわす。

$$\bigvee_{j=1}^m x_{ij} = 1, i = 1, \dots, n$$

また、各変数 v_i はドメイン D_i からその要素である値 d_j をひとつだけとり、これをブール式として表現すると次のようになる。

$$\bigwedge_{1 \leq j, k \leq m, j \neq k} \neg x_{ij} \vee \neg x_{ik} = 1, i = 1, \dots, n$$

または

$$\left(\bigwedge_{1 \leq j, k \leq m, j \neq k} x_{ij} \wedge x_{ik} = 0, i = 1, \dots, n \right)$$

ここでは制約を2つの変数間において成り立つ二項関係として考える。つまり、変数 v_i にドメイン D_i の要素を割り当て変数 v_j にドメイン D_j の要素を割り当てた場合の可能な組み合わせ、もしくは2変数間に対して割り当てを許さない値の組み合わせを二項のタプル集合としてあらわす。

前者の場合は、変数 v_i と変数 v_j のそれぞれのドメイン D_i と D_j の要素である d_k と d_l の組み合わせを二項タプル集合 $T_{ij} = \{(d_k, d_l) \mid d_k \in D_i \wedge d_l \in D_j\}$ とし、以下のブール式により表現する。

$$\bigvee_{(d_k, d_l) \in T_{ij}} x_{ik} \wedge x_{jl} = 1, i = 1, \dots, n, j = 1, \dots, m$$

後者の場合は、変数 v_i と変数 v_j のそれぞれのドメイン D_i と D_j に対して値の割り当てを許さない d_k と d_l からなる組み合わせを二項のタプル集合 $\bar{T}_{ij} = \{(d_k, d_l) \mid D_i \times D_j - T_{ij}\}$ とし、以下のブール式により表現する。

$$\bigwedge_{(d_k, d_l) \in \bar{T}_{ij}} \neg x_{ik} \vee \neg x_{jl} = 1, i = 1, \dots, n, j = 1, \dots, m$$

または、

$$\left(\bigwedge_{(d_k, d_l) \in \bar{T}_{ij}} x_{ik} \wedge x_{jl} = 0, i = 1, \dots, n, j = 1, \dots, m \right)$$

3 ブール代数による制約充足の解法

3.1 ブール方程式の解法

ブール方程式の解法は数値処理や記号 (数式) 処理の分野で用いられている代数方程式の解法についての考え方に基づいている。前者の数値処理分野では、ガウス消去法に代表される直接法や繰り返し法に基づいた解法が提案されている[4]。一方、後者の記号処理分野ではコンピュータ論に基づいた解法が提案されている[2]。

我々は制約充足問題をブール代数を用いて定式化し、得られたブール方程式に対して上記の解法を適用することで充足解を求めるアプローチをとり、以下では制約充足問題の具体例を用いて説明をおこなう。

3.2 制約充足問題の具体例による説明

図のような3つの変数 X, Y, Z とそのドメイン $D_X = \{5, 2\}$, $D_Y = \{2, 4\}$, $D_Z = \{5, 2\}$ からなる制約ネットワークを例とした制約充足問題について説明する。制約は変数 X と Z 間の関係を示すタプル集合 $(X, Z) = \{(5, 5), (2, 2)\}$ と変数 Y と Z 間の関係を示すタプル集合 $(Y, Z) = \{(2, 2), (4, 2)\}$ からなる。



変数 $X (= v_1)$ はそのドメインが $D_X = \{5, 2\}$ であり、ドメインの要素からは値 d_j をひとつだけとすると、 $X = v_1$, $D_X = D_1$, $d_1 = 5, d_2 = 2$ であり、これをブール式により表現すると次のようになる。

$$x_{11} \vee x_{12} = 1, \neg x_{11} \vee \neg x_{12} = 1$$

同様に、変数 $Y (= v_2)$ と $Z (= v_3)$ に対してもブール式を求めると次のようになる。

$$x_{21} \vee x_{22} = 1, \neg x_{21} \vee \neg x_{22} = 1$$

$$x_{31} \vee x_{32} = 1, \neg x_{31} \vee \neg x_{32} = 1$$

さらに制約 $(X, Z) = \{(5, 5), (2, 2)\}$ と $(Y, Z) = \{(2, 2), (4, 2)\}$ には、割り当てる値の組み合わせと値の割り当てを許さない組み合わせの2通りの記述が考えられる。前者の場合の制約には $(X, Z) = T_{13}$, $(Y, Z) = T_{23}$ という関係があり、タプル集合には $T_{13} = \{(d_1, d_1), (d_2, d_2)\}$, $T_{23} = \{(d_1, d_2), (d_2, d_1)\}$ という関係が成り立つとする。

a) 割り当てを許す値の組み合わせを考えた場合

上記の関係をを用いて制約 (X, Z) と (Y, Z) のタプル T_{13} と T_{23} をブール式に変換すると次のようになる。

$$(x_{11} \wedge x_{31}) \vee (x_{12} \wedge x_{32}) = 1$$

$$(x_{21} \wedge x_{32}) \vee (x_{22} \wedge x_{31}) = 1$$

前者の式は変数 X と Z 間の制約をあらわし、後者の式は変数 Y と Z 間の制約を示している。

b) 割り当てを許さない値の組み合わせを考えた場合

変数 X と Z 間および変数 Y と Z 間において値の割り当てが許されないタプルをそれぞれ $\bar{T}_{13}$, $\bar{T}_{23}$ とすると $\bar{T}_{13} = \{(d_1, d_2), (d_2, d_1)\}$, $\bar{T}_{23} = \{(d_1, d_1), (d_2, d_2)\}$ となる。これをブール式により表現すると次のようになる。

$$(\neg x_{11} \vee \neg x_{32}) \wedge (\neg x_{12} \vee \neg x_{31}) = 1$$

$$(\neg x_{21} \vee \neg x_{31}) \wedge (\neg x_{22} \vee \neg x_{32}) = 1$$

または、

$$\begin{cases} x_{11} \wedge x_{32} = 0, x_{12} \wedge x_{31} = 0 \\ x_{21} \wedge x_{31} = 0, x_{22} \wedge x_{32} = 0 \end{cases}$$

3.3 生成される方程式と求められるべき充足解

上記の例題の定式化においては、割り当てを許す値の組み合わせをとる場合 a) と割り当てを許さない値の組み合わせをとる場合 b) という2つの場合が考えられ、それぞれに対して次のようなブール式の集合 $BE1$ と $BE2$ が生成される。

$$\left. \begin{aligned} x_{11} \vee x_{12} &= 1, \neg x_{11} \vee \neg x_{12} = 1 \\ x_{21} \vee x_{22} &= 1, \neg x_{21} \vee \neg x_{22} = 1 \\ x_{31} \vee x_{32} &= 1, \neg x_{31} \vee \neg x_{32} = 1 \\ (x_{11} \wedge x_{31}) \vee (x_{12} \wedge x_{32}) &= 1 \\ (x_{21} \wedge x_{32}) \vee (x_{22} \wedge x_{31}) &= 1 \end{aligned} \right\} = BE1$$

$$\left. \begin{aligned} x_{11} \vee x_{12} &= 1, \neg x_{11} \vee \neg x_{12} = 1 \\ x_{21} \vee x_{22} &= 1, \neg x_{21} \vee \neg x_{22} = 1 \\ x_{31} \vee x_{32} &= 1, \neg x_{31} \vee \neg x_{32} = 1 \\ (\neg x_{11} \vee \neg x_{32}) \wedge (\neg x_{12} \vee \neg x_{31}) &= 1 \\ (\neg x_{21} \vee \neg x_{31}) \wedge (\neg x_{22} \vee \neg x_{32}) &= 1 \end{aligned} \right\} = BE2$$

一般、CSP をブール代数を用いてブール式に変換すると、ブール式の数が非常に増えるため、規模の大きな方程式を解くことが必要となる。例えば、後者 ($BE2$) の場合には、 mn 個の変数 x_{ij} からなる $2n + |\bar{T}|$ 個のブール式が生成される。

最終的には、上記のブール式の集合 ($BE1$ または $BE2$) を連立方程式として解くことにより、次のような充足解を求めることができる。ここでは、3.2 に示されている例題を制約論理プログラミング言語 CAL [6] で記述し、記号的にブール方程式の求解をおこなうために、ブール制約評価系を利用した。

$$x_{11} = 0, x_{12} = 1, x_{22} = x_{21}, x_{31} = 0, x_{32} = 1$$

上記の結果から解を列挙すると、 $(x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}) = (0, 1, 0, 1, 0, 1)$ と $(x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}) = (0, 1, 1, 0, 0, 1)$ が得られる。前者は充足解が変数 X はドメイン D_X の2番目の要素2、 Y は変数 Y のドメイン D_Y の2番目の要素4、 Z はドメイン D_Z の2番目の要素2であることを示しており、同様に後者は $X = 2, Y = 2, Z = 2$ であることを示している。

4 むすび

制約充足問題に対する従来の探索を基本とした解法とは異なるあらたな解法を提案した。本提案は制約充足問題をブール代数により定式化し、ブール方程式を解くことによる代数的なアプローチである。具体的には制約充足の例題を取り上げて定式化をおこない、ブール方程式を解くために、制約論理プログラミング言語 CAL のブール制約評価系を用いた。その結果、制約充足問題の解法として探索法がよく知られているが、それとは別に方程式を解くという解析的な方法が有効であることを示した。しかしながら、先程示したように定式化で得られるブール式の数が増えるので、大規模な問題に対処するためにはより有効な定式化を考えることが必要である。

謝辞

本研究は第5世代コンピュータプロジェクトの一環として行なわれた。本研究の機会を与えてくださり、常にご指導いただいたICOTの第一専任所長、古川康一研究次長、生物重治部長代理、新田克己第7研究室室長ならびに、有益なコメントをいただいた長谷川隆三部長代理に深く感謝いたします。また、 CAL を利用するにあたりいろいろ教えて頂いた、相模原第4研究室室長代理をはじめとした第4研究室の皆様にも感謝いたします。

参考文献

- [1] Johan de Kleer, A Comparison of ATMS and CSP Techniques, IJCAI-89, (1989)
- [2] 佐々木健昭, 古川昭夫, コンピュータ環境, 情報処理 Vol.27 No.4, (1986)
- [3] Rina Dechter, Enhancement Schemes for Constraint Processing: Back-jumping, Learning, and Cutset Decomposition, Artificial Intelligence 41, (1990)
- [4] 津田孝夫, 岩波講座 ソフトウェア科学 9, 数値処理プログラミング, 岩波書店, (1988)
- [5] A. D. Polimeni and H. J. Straight, Foundations of Discrete Mathematics, Brooks/Cole Publishing Company, (1985)
- [6] A. Aiba, K. Sakai, Y. Sato, D.J. Hawley, and R. Hasegawa, Constraint Logic Programming Language CAL, Proc. of Int. Conf. on FGCS88, (1988)