

7F-1

制約, ルール, 解に優先度を導入した制約論理型言語の提案と実現

丸谷哲也 永田守男
慶應義塾大学理工学部管理工学科

1. はじめに

Prologを代表とする論理型言語は宣言的な記述が可能であり, バックトラックによる非決定的な動作をするなどの特徴を持つ. 近年, 論理型言語のユニフィケーションに対し, 制約解消という概念を導入し, 領域をエルプラン空間以外も含むようにした制約論理型言語が注目されている[1].

この言語を実際的な計画型問題に応用するために拡張し, 配車計画の問題に対して適用したので, 本研究での拡張の内容と適用結果を報告する.

2. 制約論理型言語

制約論理型言語では, 論理型言語の宣言的な記述を損なわずに, 制約を記述することができる. 例えばCLP(R)[2]のプログラムは次のような節の集合からなり, Prologのホーン節と同じ形をしている.

$$A_0 :- a_1, a_2, \dots, a_k. (1 \leq i \leq k)$$

ここで, A_0 は通常のリテラル, a_i は通常のリテラルまたは制約である. 制約は項を等式, 不等式で結んだものである.

計画型問題を制約論理型言語を用いて解く場合に, 柔軟に制約を記述し, かつ効率を著しく低下させることがないシステムを構築することが要請される. このため, 本研究では, 柔軟な記述のために, 制約節, 変数それぞれに優先度を導入し, さらに, 解の効率的な探索のコントロールのために, 節の間の優先度を導入する.

3. 計画型問題

ここで取り上げる計画型問題とは, 一般に組み合わせ最適化問題といわれ, 組み合わせ爆発を起こしやすいという特徴がある. また, 実際の計画型問題では, 最適解を得ることよりも, 決められた時間内で実行可能な解を求めることをより重視する場合が少なくない. 解の最適性を評価する関数が決定できないことも多い. さらに, 計画型問題における制約条件は, 絶対に守らなければならない強い制約と, 他の制約との競合によって無視できる弱い制約に分類することができる場合がある.

ここでの実験に用いた実際の配車計画問題を例にして具体的に説明する.

配車計画問題は, 受注した貨物(量)とそれを配送する車両との組み合わせに対して, 配送経路を決定する問題である. たとえば, 制約の例としてつぎのようなものがある.

配送先制約: 配送の要求があった配送先に配送する

配送先車種制約: 配送先に対する車種の制約

注文量制約: 注文を受けた量

配送指定時刻制約: 指定された配送時刻を守る

トラック台数制約: 使用できるトラックの種類と台数

トラック積載量制約: 各トラックの限界積載量

このうち, 配送先制約, 注文量制約は必ず満たさなければならない制約である. これに対し, トラックの台数の制約では必ずしも全てのトラックを使用する必要はない. このように制約の記述は柔軟である必要がある.

An Implementation of a Constraint Logic Programming
Language with Preferences on Constraints, Rules, and
Solutions
Tetsuya MARUTANI, Morio NAGATA
Keio University

4. 拡張にあたっての考え方

制約論理型言語に対し、計画型問題における制約を記述しやすくするために、次のような拡張を考える。

(1) 計画型問題における、必ず満たさなければならない強い制約と、他の制約との競合によって無視できる弱い制約を記述するために、制約間で優先度を表現できるようにする。

(2) 組み合わせ爆発を防ぐために、ルール間に優先度を与え、解の探索法をコントロールできるようにする。

(3) パラメータの役割や、問題や立場による視点を表示できるようにするために、個別の変数に優先度を与えられるようにする。

これらのうち、(1)については、既にCLP(R)の拡張版であるHCLPで実現されている[3]ので、ここでは(2)と(3)について説明する。

まず(2)について、Prologでは、同一のhead部を持つ節は、記述されてある順番にバックトラックによって実行される。これによって、同じhead部を持つ節に優先度を陰に与えている。これを陽に表し、かつhead部が異なる節にも優先度を与えることができるようにするために、本研究ではプログラム中の節のhead部を引数とする次のような述語を導入する。

```
preferred_rules(head1, head2, ..., headn).
(headi はルールのhead部)
```

次に(3)について。一般に計画型の問題では解の最適性を判断する評価関数が一意に決定できないことがある。これは立案者の立場や視点によって解の評価が変わることがあるためである。そこで、本研究では問題のパラメータを表すプログラム内の変数について、次のような記述を導入して優先度を表現できるようにした。これによって複数の異なる変数を重視した結果得られた解から、人間の判断によって最終的な解を選択することが可能になる。

```
preferred_variables(X1, X2, ..., Xn).
(Xi は変数)
```

5. 実現

本システムはCLP(R)[2]によって実現し、次のような処理から成る。

1. 与えられた節の集合を、制約間の優先度に従って並べ換える。
2. 並べ換えられた節をすべて調べ、同じ節内の矛盾が生じる可能性のある制約について、矛盾しない制約を組み合わせることによって、新たに節を生成する。制約間の優先度に従ってこれらの節の集合を分割する。(分割法は複数通りあって、それらには優先度が存在する)。
3. 優先度の高い分割法から順に、利用者に指示された優先度のものまで以下の処理を行なう。
 - 3.1. 節間の優先度に従った節の集合の並べ換え。
 - 3.2. 並べ換えられた節の集合を実行する。
4. 出てきた解の組の中から、個別変数の性質に従って、望ましい解を示す。

6. おわりに

以上の機能を持つシステムを作成し、実際の配車計画の問題をとりあげ、評価を行なった。その結果、一定範囲内の計画型問題に対しては、この提案が有効であることを確認した。

7. 参考文献

- [1] 淵, 溝口, 古川, J-L.Lassez: 制約論理プログラミング, 共立出版(1988).
- [2] Jaffar, J., Michaylov, S., Stuckey, P.J., and Yap, R.H.C.: The CLP(R) Language and System, ACM Transactions on Programming Languages and Systems, Vol 14, No. 3, July 1992, pp.339-395.
- [3] Borning, A., Maher, M., Martindale, A., and Wilson, M.: Constraint Hierarchies and Logic Programming, Proc. of the Sixth International Conference on Logic Programming, 1989, pp.149-164.