

# スケジューリング高速化のための 知識分割処理方式

2B-1

齋 礼 矢島 敬士 増位 庄一  
((株)日立製作所関西システムラボラトリ)

## 1. はじめに

製造業における生産計画問題を計算機を利用して処理したいというニーズが高まっている。このため、従来から診断型などの分野で利用されてきたエキスパートシステム(以後、ESと略す)構築ツール<sup>[1]</sup>による計画型ESの開発が各所でなされている。現在は比較的小規模な問題への適用が進められているが、より適用範囲を広げるためには、構築工数や処理速度の面での一層の改善が必要となる。本稿ではこの改善への試みとして、ESの知識を分割して処理する方式について報告する。

## 2. スケジューリングシステム構築上の問題点

近年の、多様化しかつ激しく変化する傾向にある消費者ニーズに対応するためには、生産計画の変更を柔軟かつ迅速に行うことが要求される。一般に、計画問題の特徴は、計画対象となるデータ数が多く、複数の制約条件を考慮して解を求めるという組合せ問題である点にある。

従来から利用されている汎用のES構築ツールは、制約条件などの記述が容易であるという点で計画問題に好都合である。処理速度の一層の改善が進むと、多段処理工程等の計画データが多く、人間でも難しい問題を実用レベルで解くことも可能となる。

従来この問題に対して、手続き型言語による処理や、問題を部分問題に置き換えることによって高速化が試みられてきたが、それだけでは十分な効果をあげることができない場合がある。

## 3. 問題解決のための手段

上記の問題点を解決するために、本稿では知識を内容によって複数の部分に分割し、それぞれを独立に<sup>[2]</sup>かつ処理順序を制御する方式の提案を行う。

### 3.1 計画知識の分割

本稿では、RETE<sup>[3]</sup>のような条件照合アルゴリズムを用いたプロダクションシステムを前提としたルールの分割を行う。(図3.1)

従来、ESの知識ベース内のルールを、メタルールと(一般の)ルールという形で、階層的に記述する方法があった。ルールは、その処理内容により、ルール群に分割される。メタルールは、ルール群の実行順序を制御する。

メタルールの概念により、表記上ルールを分割して記述することが可能であったが、ツールの処理系は、分割して記述されたルールを独立に処理できなかった。ルー

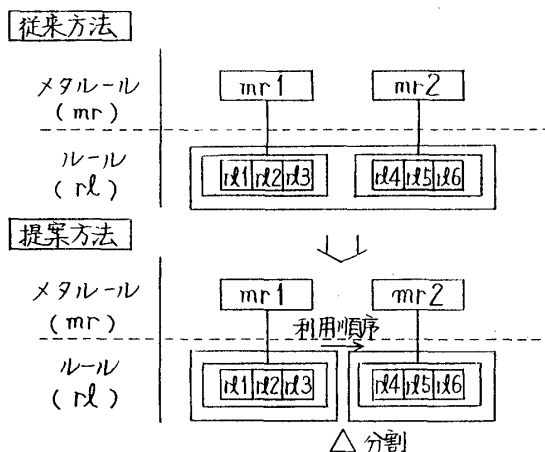


図3.1 ルールの分割と利用順序

ルの実行によりフレームの内容が変更になると、すべてのルールのネットワークが更新の対象となる。例えば、ルール毎に参照されるフレームのスロット値(属性値)が異なる場合のように、すべてのネットワークを更新の対象とする必要はないことがある。このような場合には、条件照合のオーバーヘッドが大きくなり、RETEアルゴリズムはその有効性が抑えられ、かえって遅くなる。ルールを分割して処理することにより、ディメリットが解消される。

さらに、複数の処理がある場合、これらの間に処理順序を設定すると効果的に高速化できる場合がある。例えば、1ルールで1データを抽出する処理と、1ルールはデータ群のサブセットを抽出し、これを繰り返して最終的に1データを抽出する処理の両方が存在する場合である。この時には、先に前者のルールを利用して、次いで後者の処理を行うと、データ処理上、効率が良い。

### 3.2 効率的な処理を実現する手段

前記の方式を実現するために、ユーザは計画に必要なルールをその内容に応じて分割して記述する。分割の条件は、例えば、生産納期の厳しい計画データを処理するものとそうでないもの、処理に時間のかかるものとそうでないもの、などである。

ツールでは、分割されたルールごとに条件照合のネットワークを構成し、あるルールで起こったフレームの変化を、すぐにその他のネットワークに反映するのではな

く、知識ベース内に変更フレーム保存ファイルを設けていったん保存し、知識利用の過程で利用対象のルール部分に変更になった場合に、新しく利用対象になったルールと変更フレーム保存ファイルに保存されているフレームとを照合し、次の実行可能ルールを決定する。

### 3.2 知識処理方法の概要

本稿の知識処理方法の概要を、ルールを2分割して記述する例を用いて説明する。(図3.2、図3.3)

ルールは、別々のファイル、AとBに格納される。推論エンジンは、分割されたルールファイルごとにネットワークを構成し、実行されるルールを含むネットワークのみを更新対象とする。

まず、Aのルールが始めに利用対象となる。初期段階のフレーム(処理データの特徴を示すフレームなど)に関するAのネットワークの更新が終了すると、Aの中に実行可能なルールがあるかどうかを判断する。実行可能なルールがあれば、それが実行される。

実行の結果フレームが変更されたら、そのフレームを変更フレーム保存ファイルに保存し、現在時刻を最終更新時間としてフレーム内に設定する。以前に変更になったフレームが再度変更になったら、新しく更新になったフレームを保存する。次に、変更フレームに関して、Aのネットワークの更新を行い、再び実行可能なルールがあるかどうかを調べる。

Aで実行可能なルールがなくなったら、Bに記述されているルールが新規に利用対象となる。利用対象となるルール部分がAからBに変更になると、先に保存されたフレームと、Bのネットワークが照合され、実行可能なルールがあるかどうかを調べられる。ネットワークの更新は、フレームに付加されているフレーム変更時間を参照して行われる。即ち、更新対象となるネットワークが

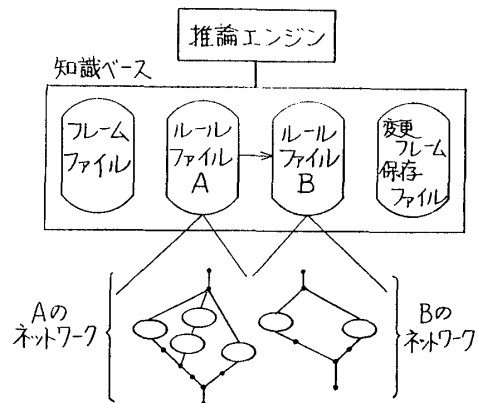


図3.3 ルール分割

AからBに変わった時、新しく更新対象になったネットワークが、前回更新対象となった最終時間からの現在時刻までの間に、ルールの実行で変更されたフレームを流し込んでネットワークを更新する。

### 3.4 本知識処理方法の効果

本稿の知識処理方法によれば、

- (1) 知識変更内容の照合を、分割した知識のそれぞれに限定して行うので、照合範囲が狭くなり、処理時間が短縮される。
- (2) 同じ知識が複数回更新される場合、最終変更値のみが次の利用対象知識部分で照合されるので、処理時間が短縮される。
- (3) 短時間で終了する処理を先に行うように制御できるので、後処理における処理データが減少し、処理時間が短縮される。

### 4. おわりに

上記のルール分割方式によれば、(机上の計算ではあるが)同時に考慮するデータが1割~2割減少し、その処理時間は3割~5割減少することから、現時点では有効なものであると考える。

また、知識分割についての、知識を手続き的なものと宣言的なものの2種類に分割する事例<sup>[4]</sup>はあるが、それら2つの利用順序、その他の知識の分類観点、及びその処理系について検討したのが本報告の特徴である。

今後実システムへの展開の中で、その有効性を確認していく。

### 参考文献

- [1] 増位, 船橋: 知識ベース技術の実システムへの応用; 情報シンポジウム論文集, vol.87, No.3, pp51-60 (1987)
- [2] 石田, 桑原: プロダクションシステムの高速度化記述; 情報処理, 29巻, 5号, pp.467-477 (1988)
- [3] Forgy, C.L.: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, Artif. Intell., Vol.19, pp.17-37 (1982)
- [4] 田中: 設計効率化の切り札, 本格的な普及期を迎えるインテリジェントCAD; 日経コンピュータ, 1989.2.27号, pp74-89 (1989)

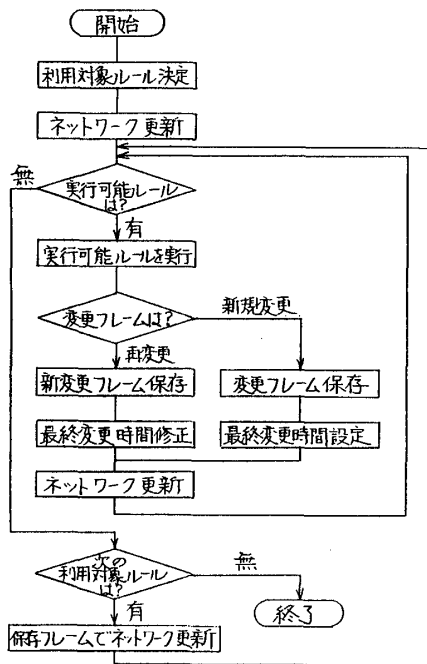


図3.2 知識処理手順