スケジューリング知識に関する一考察

6C-5

森下 太朗　和田 正寛
シャープ(株) 情報システム研究所

1．はじめに

近年、エキスパートシステムの特化シェルが注目を集め開発が活発化している。これは類型タスクに対する認識が高まり、実システムの構築を通して個々の問題毎にタスクの解明を行ってきた結果である。次世代のドメインシェルにおいては、知識獲得や並列プログラミングが必須要素として組み込まれること予想し、それにふさわしい知識表現の方法を検討しておくことが重要である。ここでは、簡単なスケジューリング問題に対するプログラミングを通して、配置型問題の並列的な解法や知識表現について検討してみたのでその結果を報告する。

2．スケジューリング問題

我々が扱って来た問題は、秘書スケジューリングの問題である[1]。この問題はスケジューリング要求された複数の項目(スケジュールアイテム)に対して、各種の制約条件を守りながら、週間スケジュール表を求める問題であり、「制約条件の下で対象物のパラメタの値を決定する配置型の問題」として位置付けられる。

3．論理型言語による解法

従来のPrologによる解法では対象物を１つづつ逐次的に処理していた。すなわち、対象物のパラメタが取り得る候補を１つ生成し、その都度制約をチェックし、違反していなければ次の対象物の処理に移り、違反していればバックトラックするという解法を取っていた。この方法では、もし、値の決定している対象物や、他のパラメタへの影響の大きい対象物の決定順番が後になれば、非効率的なバックトラックを招いてしまうため、パラメタの決定順番に関する知識が効率的な解法上重要なものとなる。

これに対して、バックトラックを排除し並列に問題解決を行うGHC[2]では、個々の対象物のパラメタの値の候補を並列にストリームとして流し、制約条件でふるいにかけて違反するデータを流さないようにしながら最後に残った値の候補の組み合わせを解とする。　制約条件によるふるいのかけ方には、一本のストリームに対してだけの場合(個々のパラメタに対する単一制約)と２本以上のストリームに対する場合(パラメタ間の相互制約)とがあり、特に後者は呼び出し側の各プロセスからのデータを待ったうえで、それらを組み合わせたストリームを作成するためプロセスのすり合わせの負荷が大きい。GHCによる計算ではパラメタの決定順番はOR並列という立場から、本質的でなくなっている。GHCは、プロセスを中心としたストリームプログラミングに主眼がおかれているため、独立したプロセスの数が多いほど、またパイプライニングの数が多いほどその並列性は高い。従って、配置型問題のようなフィルタリングが主となるプログラムにおいては、何をストリームとして選択すれば AND並列におけるプロセス間の負荷が小さくなるかが重要な観点になるものと考えられる。
観点としては例えば以下が考えられる。

(1)人ごとに並列に流す　各人のスケジュール表に同一のアイテムがばらばらに存在することになり、意味的に無駄な組み合わせが大量にできてしまう。

(2)アイテムごとに並列に流す　その都度、人、資源のチェックができるので、意味的に無駄な組み合わせは抑えられる。

(3)資源ごとに並列に流す　問題が部分的になる。また(1)と同様の問題が生じる。

4．EPSILON/ONEによる解法

EPSILON/ONE[3]は、ICOTで開発された知識獲得支援ツールであり、システム主導の対話(プリポスト法)による知識の抽出、問題解決タスクの基本オペレーションによる表現(専門家モデル)、ＥＳＰによる実行・評価という一連の知識獲得の過程を支援するシステムである。 EPSILON/ONEでは、プリポスト関係により抽出されたいくつかのタスクの処理の流れに沿って、問題解決が行われる。このタイプの問題の場合、「タスク」とは、基本的には、「対象物のパラメタの値の候補の生成」と「パラメタの値の検査」だけである。プリポスト法のタスク抽出過程はシステム的な制御というより専門家が表明する上流のタスクの抽出を考えて設計されて

A study about scheduling knowledge,

Taro MORISHITA, Masahiro WADA, Information System Labolatories, SHARP Corporation.

いるため、バックトラックを考慮したタスクの表現はふさわしくないし、本質的でない。まず、制約のチェックのタスクを表現すると、他の、前後に必要なオペレーションが自然と見えてくる。 EPSILON/ONEによる解法知識の表現例(専門家モデル)は、「生成」タスクにおいてあらかじめ全ての解候補を生成し「検査」タスクにおいて全解候補の組み合わせをチェックし、残った組み合わせが解であるというタスクを表現しており、その内容はGHCのプログラミング結果と同一であることが分かる。

以下に知識獲得の特化機能として必要な項目を上げる。

(1)「制約」、「戦略」、配置範囲や配置物に対する知識の記述のみから専門家モデルを自動生成する。

(2) 処理効率を上げるために組み合わせ方に対する知識の記述枠を設ける。一番処理時間のかかる、組み合わせ－テストのプロセスを検出して、それを解消するためのタスク分割・組み替え手段を提示する。

(3) 値が予想されるものに対して先行計算ができるような記述枠を設ける。

5．制約緩和知識の表現

一般に設計・計画型の問題の場合は、制約が強すぎて設計出来ない場合の対処が問題となる。パーソナルスケジューリング問題では特に、追加・変更は日常的に要求されるものでありその際に制約緩和知識が用いられるものと考えられる。 制約緩和知識をメタ知識として表現する方向は正攻法であるが知識の表現が難しい。ここではユーザインタフェースを考慮した、制約空間のポテンンシャルに基づく緩和方法を提示しておく。

(1)各アイテムに対して各属性(曜日・時間・参加者)に対する好・嫌の度合(ポテンシャル)を設定する。

(2) 1で設定したポテンシャルの境界値を設定する。

(3) 2から各アイテム毎に制約空間(スケジュール表の制約が適用される場所)を設定する。制約空間はアイテムの配置候補の場所を規定する。

(4)配置できない場合、トレース過程の失敗したアイテムに着目して境界値を大きくする。

6．おわりに

本研究はICOT受託研究の一環として行われたものであり、研究の機会を与えて下さった各位に感謝致します。

[参考文献]

[1] 和田 他，"IS³における知識獲得手法"，情報処理学会第38回全国大会 2G-4(1989)

[2] 古川 他，"並列論理型言語ＧＨＣとその応用"，共立出版， 知識情報処理シリーズ6(1987)

[3] 滝 他，"知識獲得支援システム(EPSILON)における専門家モデル"，情報処理学会研究報告誌,知識工学と人工知能52-4(1987)

月曜日の午後(1時から5時まで)に開始時刻不定の会議(1時間)と面会(1時間)を配置する。
2時台に会議を置くこと、4時に面会を置くこと、会議と面会が連続することは禁止されている。

◎ ↑例題として問題を簡単化した一例
◎ GHCによる上記例題の解法の一部 →
◎ EPSILONによる上記例題の専門家モデル↓
◎ 制約緩和知識表現の一例↓

```
schedule(S):- true |
    gene-item(Kaigi,Menkai),
    check-k(Kaigi,X),        -> check-k([C1|CS],X)
    check-m(Menkai,Y),       -> check-m([M1|Ms],Y)
    check-km(X,Y,S).         -> gene-comb(X,Y,Z),
                                check-comb(Z,S)
```