

スケジューリングタスク記述モデル: GTB モデル

3 D-1

進藤静一、安井照昌、清水広之、村井龍一、小林健三
三菱電機(株)

1. はじめに

エキスパートシステムのデータ構造と制御構造のテンプレート、及び、エキスパートシステムの作成のガイドラインを与えることは、知識獲得や知識ベースの再利用を容易にする [1][2][3]。我々は、この観点に立って、スケジューリング業務を対象を絞り、開発事例を分析することにより GTB(Generate & Test Box) モデルを得た。GTB モデルの特徴は、生成検査法のシーケンスで制御構造を表現することである。本稿では、GTB モデルの構成要素、GTB モデルに従ったシステム開発手順、及び、GTB モデルによる記述例を中心に報告する。

2. スケジューリングタスクの表現

スケジューリングタスクとは、「ジョブ、リソースに関する制約条件を満たし目的(評価基準)を満足するような、ジョブのリソース-時間平面への配置を決定すること」として捉えることができる。スケジューリングタスクの仕様は大きく分けて以下の2つの要素で構成される。

1. 対象: データ構造の記述である。ジョブとリソースの特性をフレーム形式で記述する。制約はフレーム内の属性値で記述する。例えば、制約 C1: 「マシン M1 が処理できるジョブは J1 と J2 である」、制約 C2: 「マシン M1 で J1 から J2 への段取替時間は T1、J2 から J1 への段取替時間は T2 である」、はフレーム M1 の属性として

```
M1::
  (attrib:jobs, value:[J1,J2]),
  (attrib:dandori,
   value:[[J1,J2,T1],[J2,J1,T2]]) ....
```

と記述する。

2. 解法: アルゴリズムの構造の記述である。フローチャートで記述する。

スケジューリングタスクは組合せ問題であり、組合せ的爆発を起こす可能性を持つ。それに対し、専門家は解の探索の制御方法を知識として持ち適用することによって、妥当な時間で満足のゆく解を得ていると考えられる。従って、スケジューリングタスクでの有効な知識は、探索の制御、即ち、解法の知識であり、解法の知識のテンプレートを構築することが重要である。

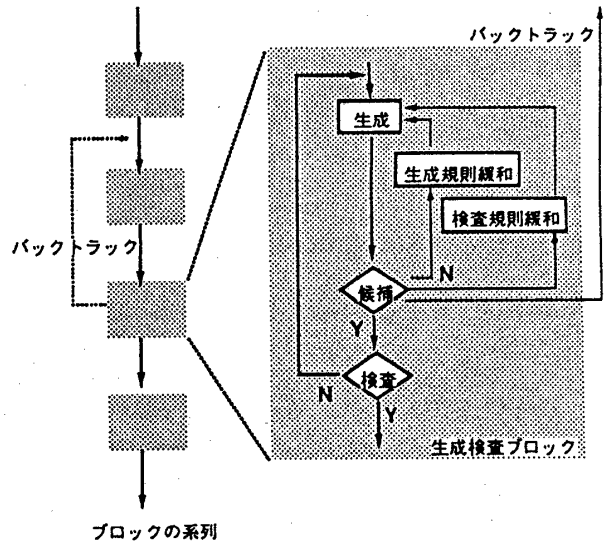


図 1: GTB モデルによる解法の表現

3. GTB モデル

GTB モデルは、既存の幾つかのエキスパートシステム [4] の制御構造の共通部分を抽出することにより得られた、スケジューリングタスクの解法を記述する枠組である。GTB モデルは、複雑な問題の解決の原則である分割統治法とスケジューリングタスクの一般的解法である生成検査法に基づく。GTB モデルで表現された解法の一般形を図 1 に示す。GTB モデルでは、(1) 図 1 のフローチャートを得る為の手順(ガイドライン)、(2) フローチャートの各箱に書かれる単語(語彙)、を提供することにより、解法の統一的な記述を容易に得ることを支援する。

フローチャート作成のガイドライン

有効な解を得る範囲内で、組合せ的爆発を避ける手段を、ブロック分割、最適化問題の変形、生成過程で使用する制約の選択、検査失敗の処理、の4段階で施すことによりフローチャートを得る。

1. ブロック分割: 部分問題間のバックトラックをなるべくなくすように問題を部分問題に分ける。典型例は、階層化とボトルネック箇所の切り分けて

ある。

- 階層化: 最初に荒い割当てを行ない、次にそれらを詳細化する(例. まず各ラインの担当ジョブをざっと決め、その割当てを前提として各ラインでのジョブの実行順序を決める)。

- ボトルネック箇所の切り分け: ボトルネックとは制約が厳しい箇所であり、この箇所をうまくスケジュールできれば他の箇所はほぼ必然的に割当てが決まる場合が多い。従って、ボトルネックの箇所を切り出して、その箇所のみスケジュールを最初に行なえば、組合せの数が少なくて解に到達する。

2. 最適化問題の変形: スケジュールタスクでは、スケジュールリング結果の判断基準としてある量の最大/最小化(例. 総コスト、総処理時間)が指定されることが多い。しかし、厳密に最適化を行なうためには全数探索が必要であり組合せ的爆発を起こしてしまう。そこで、最大化問題の場合は最低ラインを設けその値以上の結果が得られたならば探索を打ち切る等して、最適化問題を制約充足問題に変換することにより、組合せ的爆発を避ける。

3. 生成過程で使用する制約の選択: 制約条件は生成検査ブロックの生成か検査の何れかに使用されるが、生成に使用すると探索空間が小さくなるような制約を選択し、それらを生成に使用する。そのような制約の例として、ボトルネックに関わる制約、候補数が少ない制約(例. ジョブJ1の処理候補マシンが10台、ジョブJ2の処理候補マシンが3台の場合、J2の割当てを生成しJ1の処理候補マシンで検査する)、が挙げられる。

4. 検査失敗の処理: 生成検査ブロックの検査で全ての候補が失敗に終わった場合、なるべく当該ブロック内で再割当てを試みることにより、なるべくブロックを越えたバックトラックを発生させない。当該ブロック内で再割当ては、検査規則を緩和することにより行なう。また、ブロックを越えたバックトラックをせざるを得ない場合は、当該ブロックが用いる条件を決定したブロックまで遡る。

4. 記述例

3ラインから成る、フローショップ問題を例にとる。製品の種類、量、納期が指定されたジョブ群をうまくラインに振り分けることにより、納期遅れ無し、且つ、各ラインの負荷均等を達成することが問題である。各ジョブには、ラインの選好順序が付けられている。

この問題に対し、専門家は、(1)各ラインの負荷を基に、ジョブのラインへの割付を決定する、(2)各ラインで、ジョブのラインへの投入順序を決定する、という2ステップの方式をとっている。且つ、ステップ間でのバックトラックは行なわない。この各ステップがGTBモデルの1ブロックに相当する。

第1ステップでは、以下の手順をとる。(1)全ジョブをその第1優先のラインに割当て。 (2)全ライン

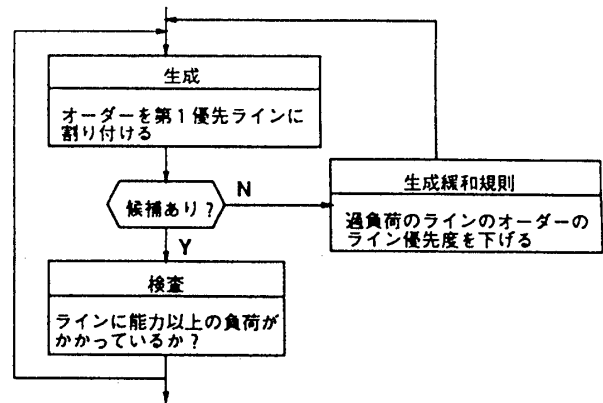


図 2: GTB モデルによる記述例

で負荷がオーバーしているか調べる。オーバーしていなければ終了。オーバーしていれば(3)へ進む。(3)負荷がオーバーしているラインで、処理量が最大のジョブを次点の優先順位のラインに割当て替える。(4)(2)へ戻る。

本来の目的である負荷の均一化を「全ラインが負荷量のある値を越えさせないこと」で代用していることに注意されたい(最適化問題の変形)。このステップに対応する生成検査ブロックを図2に示す。生成検査ブロック内で、生成過程は「第1優先度のラインにジョブを割当て」ことに、緩和規則は「次点の優先度のラインにジョブを割当て」ことに、対応する。

5. おわりに

GTBモデルはスケジュールリングタスクの制御構造の記述の汎用的枠組を与えることを目的とする。GTBモデルは、記述(フローチャート)作成へのガイドラインと、記述の語彙からなる。本稿では、ガイドラインを中心に報告した。現在、語彙の方を、[1][2]を参考にしながら構築中である。

参考文献

- [1] 溝口 et.al., 「知識の再利用を目指したタスク分析とタスクオートロジーの整備」, AIシンポジウム'92, 人工知能学会, 1992.
- [2] 中村 et.al., 「スケジュールリング知識共有化の試み - 問題解決過程の分析手法 -」, 第6回人工知能学会全国大会予稿集, 1992, pp.733-736.
- [3] Neches, R et.al., "Enabling Technology for knowledge sharing", AI magazine, Fall, 1992, pp.37-56.
- [4] 例えば, 安藤 et.al., 「石油精製におけるトッパー通油立案システムでの知識の運用について」, 第4回人工知能学会全国大会予稿集, 1990, p555-558.