

## リソース制約スケジューリング問題のモデル化と時間割作成システムへの適用

01507203 愛知女子短期大学 堀尾 正典 HORIO Masanori  
01204223 南山大学 鈴木 敦夫 SUZUKI Atsuo

## 1. はじめに

スケジューリング問題は、その性質によってさまざまな種類に分類される。その一つ、リソース制約型のスケジューリング問題（資源制約付きスケジューリング問題、有限資源プロジェクトスケジューリング問題とも言う）は、幾つかのリソース（マンパワー、設備、資金、資材など）とその供給量が与えられたとき、それらのリソースを消費する複数のタスクを、このリソース供給量の制限を越えることがないように実行順序を決定する問題である。リソース制約型問題は、機械、労働力、教室や教員などをリソースと考えるとそれぞれショップ問題、ナーススケジューリング問題、時間割作成問題となる。

このように、非常に広範囲な適用領域を持っているリソース制約型スケジューリング問題を容易に適用できるよう、一般的なモデル化を試み、その有効性の検証のため具体的な例として時間割作成問題への適用を行った。その結果、本モデルの有用性と、今後の検討課題などが明確となった。

## 2. リソース制約型スケジューリング問題のモデル化

リソース制約型のスケジューリング問題は、性質が異なる次の二つの制約を持った問題である。それらは、

- ①割付を行うタスクとリソース供給量との制約  
②タスクとタスクの前後関係（先行制約）  
である。これらは[1]で、

$$\textcircled{1} \sum_{j \in J_{r,t}} k_{j,r,t-s_j} \leq K_{r,t}, r \in R, t = 1, 2, \dots$$

$$\textcircled{2} s_{j_1} + p_{j_1} \leq s_{j_2}, j_1 \prec j_2$$

と定式化されている。ただし①式は、 $R$ : リソースの集合、 $J$ : 作業の集合として、リソース  $r$  が時刻  $[t-1, t)$  で供給可能な最大量が  $K_{r,t}$ 、作業  $j$  の開始時刻を  $s_j$  としたとき時刻  $t$  で作業  $j$  がリソース  $r$  を  $k_{j,r,t-s_j}$  だけ消費することを表している。一方②式は、 $j_1$  の作業時間を  $p_{j_1}$  とし

たとき、 $j_1$  が完了した以降に作業  $j_2$  が実施される必要があるという、先行制約（いわゆる離接制約）を表していることが分かる。

ここで、もしもすべてのリソースが供給量に等しい一定のリソース消費量を有するとすると、離接制約の表現に他ならなくなり、組み合わせ問題に帰着されることが分かる。

## (1)バックギングによる制約の視覚モデル化

我々は、上記の制約式を検討し、直観的で分かり易い制約表現の視覚化モデルとして、リソース制約型問題を次のような矩形パターンで表現する方法を考案した（図1）。

## (2)スケジューリング処理のルール化

スケジューリング能力を大きく左右する処理としてディスパッチングルール（タスクの割付順位の決定）、候補時刻の算出規則（リソース制約や先行制約を満たす時間帯の計算とその中から候補時刻を絞り込む規則）、バックトラックのルールが重要と判断し、これらの処理をそれぞれヒューリスティックを用いて実現した。また、実現したシステムは、これら規則に対して柔軟な参照・変更・追加ができるような構造（各処理のモジュール化）とした。

## (3)割り当て問題を用いたプランニング工程の導入

通常、リソース制約型スケジューリング問題では、タスクがどのリソースをどれだけ消費するかという条件は所与として考えることが多い。しかし、時間割作成の教員や教室、多機械のショップ問題などでは、どのタスクがどのリソースを消費するのが事前に決定されていない場合もある。

（この科目は先生Aもしくは先生Bが担当など）。この様な問題に対応するため、スケジューリングに先立ち、ネットワークの割り当て手法を用いて、目的関数が最小（最大）となるよう事前にタスクの消費リソースを決定するプランニング工程を導入する方法を考案した。

## 3. システム化 (Scheduling-Expert-Engine-System)

このモデルに基づいたスケジューリングシステムを SEES (Scheduling-Expert-Engine-System) と名付け C++ を用いて開発を行った [2]。開発環境などは、次の通りである。

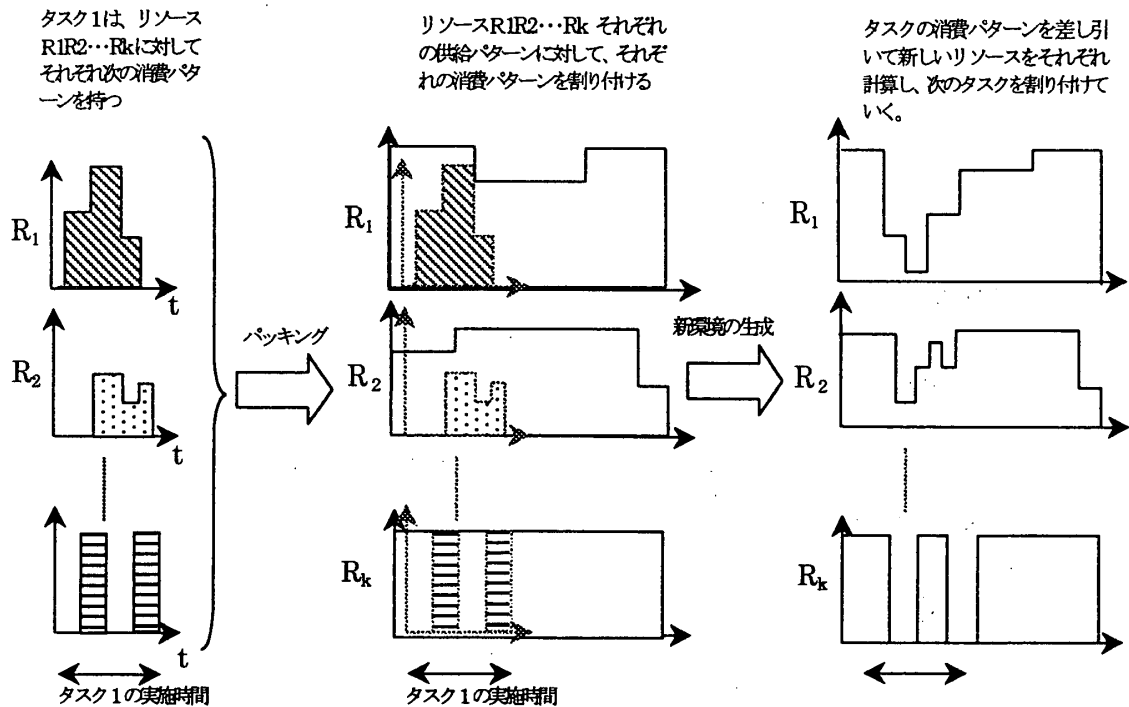


図1. パッキングによるモデル化

- ・動作環境：DOS/V, Windows98
- ・開発言語：Visual C++6.0
- ・サイズ：約40k
- ・開発工数：約2人月

また動作概要としては、定義された次の3つのC++クラスに対する各Objectが、リンク関係を継承し、バックトラックを繰り返しながらタスクを一つずつ割り付けていく構造となっている。

○<コントロールクラス>

実行制御を行うクラスで唯一のObjectを有する。各種割付ルールをメンバ関数として持っている。

○<タスククラス>

割付タスクの先行制約や名前、使用リソース名のリストなどの各種性質を情報として持つ。割付タスクそれぞれに一つずつのObjectが生成される。

○<環境クラス>

一つのタスクの、一つのリソース消費に対して、その矩形パターンを格納するクラス。パッキング可能時間帯やパターンの差し引きを行うメンバ関数を持っている。最大で(リソース数×タスク数)だけのObjectが生成される。

4. 時間割作成システムへの適用と評価

本システムを時間割作成システムに適用した。幾つかのテスト事例とその具体的な実行結果については当日発表する所存であるが、特に

- (1)同一種教室や同一種教員は高さのある一種類のリソースとして表現できると言う特質より、

計算量や変数を減らすことが可能となる。

- (2)制約の視覚化はスケジューリング知識や手法の獲得に効果的となる。

と言った点が評価しうると考えている。

5. 今後の課題

本手法をより高速化し、利用可能な範囲をより広げていくために、幾つかの検討課題が明らかとなった。主なものをあげておく。

- (1)枝(時刻)の発生と選択の方法(GAなどのメタヒューリスティックの導入etc)。
- (2)計算幾何学によるパッキングアルゴリズムの効率化。
- (3)適用範囲の拡大(様々なスケジューリング問題への適用、GUIの開発で利用しやすいものに)。

参考文献

- [1]野々部宏司、茨木俊秀：大きな構造物の生産スケジューリング問題に対するRCSPアプローチ、スケジューリングシンポジウム99, 1B3
- [2]堀尾正典：スケジューリング型エキスパートシエルの専用エンジン、愛知女子短期大学研究紀要第24号, 1991年3月

以上