

# 設計プロジェクトのスケジューリング

## -優先つき代替リソースを考慮したスケジューリング-

01404660 成松 克己\* (株)東芝 研究開発センター†

### 1 はじめに

プロジェクトスケジューリングとは、プロジェクトを遂行するのに必要な各工程のスケジュールを立てることである。これまでも PERT/CPM[1], RAMPS, PERT/LOAD[2] など様々な研究がなされ、主に建設現場などのスケジューリングに効果を上げている。

これらの従来のプロジェクトスケジューリングでは、集団に対する作業スケジュールを作成するのが目的であった。このため負荷山積みを行ない、最大人数を越えないように時間軸方向に山崩しを行なう方法が数多く論じられている。しかしながら、代替リソースを考慮して時間軸と垂直な方向への山崩しの方法についてはあまり研究がなされていない。

設計作業のようなプロジェクトを考えた場合には基本的には個人作業となり、各個人に対して能力差等を考慮したスケジューリングが必要となる。従って最大人数で山崩しを行なうだけではなく、代替リソースを考慮した時間軸方向への山崩しを含んだ問題となる。さらに、作業内容と作業の間での所要時間に反映されない選好関係を考慮する必要があり、より複雑な問題となる。

本稿では、設計作業のような優先つき代替リソースを考慮したプロジェクトスケジューリングに対するヒューリスティックな解法について述べる。

### 2 問題の定義

$J_i$ :  $i$  番目の工程  
 $b_i$ :  $J_i$ の開始時刻  
 $c_i$ :  $J_i$ の終了時刻  
 $r_i$ :  $J_i$ の割り付けリソース  
 $C_r$ : リソース  $r$ の容量  
 $t_{ir}$ :  $J_i$ のリソース  $r$ に対する工期  
 $f_{ir}$ :  $J_i$ のリソース  $r$ に対する選好性  
 $S_i$ :  $J_i$ の最早開始時刻  
 $D_i$ :  $J_i$ の希望納期  
 $P_{ij}$ : 順序制約の有無

$$P_{ij} = \begin{cases} 1 & J_i \text{の後に } J_j \\ 0 & \text{制約なし} \end{cases}$$

$\delta_{irk}$ :  $J_i$ が時間  $k$ にリソース  $r$ を占有しているかどうか

各工程に対する一つのリソース  $\{r_i\}$ と、開始時刻  $\{b_i\}$ を決定する問題である。各工程の順序関係は  $\{P_{ij}\}$ によって任意に与えられる。ただし通常の PERT ネットワークと同様、有向グラフはサイクリックになってはならない。また、 $\{S_i\}$ および  $\{D_i\}$ の値はすべての工程に与えられるのではなく、一部の工程に対して与えられ、 $\{D_i\}$ が与えられた工程を特に「マイルストーン工程」と呼ぶことにする。さらに、 $\{S_i\}$ および  $\{D_i\}$ のデフォルト値は  $\infty$ とし、この値の場合はこれらの変数の値は無視される。

#### 2.1 スケジュールに対する要求

スケジュールに対する要求は、1) 納期に合わせる、2) いくつかでまとまった流れのある工程はできるだけ同じ人に割り当てる、の2つを考える。2) に対してはある工程を決定した時点で動的に制約が発生する難しい問題となるため、ここではあらかじめ希望担当者を割り当て、できるだけこの希望担当者に割り当てるというリソース選好関係の問題とする。目的関数は次のようになる。

$$Q_1 = \min_{\{b_i\}\{r_i\}} \sum_i |D_i - c_i|$$

$$Q_2 = \max_{\{b_i\}\{r_i\}} \sum_i f_{ir_i}$$

$Q_1$ は工期に関するもの、 $Q_2$ はリソース選好性に関するものである。 $f_{ir_i}$ は、作業の得意/不得意を表現するものではなく、先に述べた希望担当者に割り当てられているかどうかによって値が決まるものであり、例えば希望担当者なら  $f_{ir_i} = 1$ 、それ以外なら  $f_{ir_i} = 0$ とする。

#### 2.2 制約条件

- (1) 開始時刻:  $S_i \leq b_j$
- (2) 工期:  $b_i + t_{ir_i} = c_i$
- (3) 順序制約:  $c_i \leq b_j$  (if  $P_{ij} = 1$ )
- (4) リソース容量:  $\sum_i \delta_{irk} \leq C_r$  ( $r, k = 1, 2, \dots$ )

\*なりまつ かつみ narimatu@ssel.toshiba.co.jp

†〒210 川崎市幸区柳町 70

現実には休日も考慮しなければならないが、この省略は解法にはほとんど影響しない。

### 3 解法

選好性のない代替リソース付きの Job-Shop 問題に対しては、[3] でラグランジュ緩和法により近似的に解かれているが、この方法で優先を考慮した場合に拡張した解法はまだ報告されていない。本稿では、最適化手法ではなくヒューリスティックな解法でこの問題を解く。基本的には PERT 計算の結果を参照して各工程のリソース割り付けを行なうことになるが、ポイントは、納期と希望担当者という二つの目標をどうやってうまくバランスさせた解を見つけるかである。また本稿では、前詰めでスケジューリングを行なうことを前提とする。

**Step1:** PERT 計算により、最早開始時刻  $F_i$  と最遅終了時刻  $L_i$  を計算する。すでに割り付けが終了しているものは  $b_i = F_i$ ,  $c_i = L_i$  として計算し、最遅終了時刻はマイルストーン工程の  $D_i$  をもとに計算する。

**Step2:** PERT 計算の結果を参照して緊急度の高い工程  $J_i$  を選択し、割り付け対象工程とする。

**Step3:**  $J_i$  を各リソース  $r$  に割り付けを行なったときの、すべての工程の最早開始時刻  $F'_i$  を計算し、評価値  $V_{i,r}$  の最も小さいリソースに実際に割り付けを行なう。

$$V_{i,r} = \max_{i, D_i \neq \infty} (F'_i - F_i - \alpha_i f_{i,r})$$

**Step4:** Step1 へ戻る。すべて割り付けたら終了。

以上のステップでスケジューリングが行なわれる。基本的には「希望する担当者には少々遅らせてでも割り付けるが、あまり遅れる場合には他の人に割り付ける」ようにしているだけであり、この遅らせる限度をうまく調整して納期と割り付けリソースのバランスをとるという考え方である。

#### 3.1 重みの設定

普通、最早開始時刻で山積みを行なったときの必要リソース量は全体のリソース量を越えているため、マイルストーン工程の  $F_i$  は徐々に遅れていくことになる。従ってこの遅れ方が  $\alpha_i$  により制御される。ここで、 $\alpha_i$  の変化がスケジューリング終了時における終了時刻と割り付けリソースに対する満足度である  $\sum f_{i,r}$  にどのように影響を与えるかを図 1 に示す。  $F_i^0$  は、初回の PERT 計算における最早開始時刻  $F_i$  である。

この問題は組合せ問題であるため、連続、単調なグラフとはならないが、大まかな動きとしては図 1 のようになることが予想される。

したがって  $\alpha_i$  の値が図 1 の  $\alpha'_i$  になっているようにすればよいが、この点を正確に求めるためには周囲の値をすべて求めなければならない。  $\alpha'_i$  を決定すべき主要因は、a) 希望終了日  $D_i$  と  $c_i(\alpha_i = 0)$  の差、b) カーブ自体によるもの、の 2 つであり、これらの要因のうち、スケジューリング時に最も変化しそうなものは希望終了日  $D_i$  である。

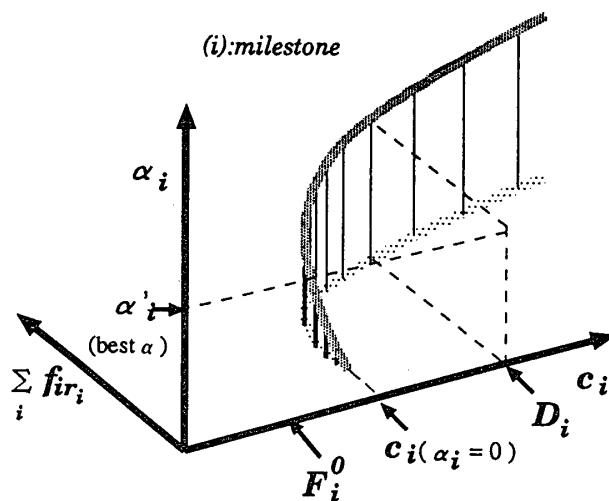


図 1: 重みと終了時刻,  $\sum f_{i,r}$  の関係

そこで本稿では大雑把に  $\alpha_i$  を見積もるために、 $D_i$  の関数にする。例えば、

$$\alpha_i(D_i) = \frac{\beta_i(D_i - F_i^0)}{N}$$

( $N$ はそのマイルストーンの前に行なう工程数) とし、カーブを決める主な原因である PERT ネットワークの形や、 $c_i(\alpha_i = 0)$  を決める主な要因であるリソースの混雑状況によって  $\beta_i$  を調節するようにすればよい。

### 4 おわりに

本稿で述べた手法により、ユーザは希望担当者の設定とマイルストーンに対する納期  $D_i$  および重み係数  $\beta_i$  の設定だけでスケジュールを作成することができる。目的関数は明確にしていないが、各設定変数によりスケジュール結果の性質を変えることができるため、ヒューリスティックな手法でありながらスケジュール結果に対する要求を意識的に反映させることが可能になる。また、この手法では、各個人の休日を避けるように割り付けを行なうだけで休日を簡単に扱うことができ、実際の作業指示が出せるスケジューリングができる。

### 参考文献

- [1] 関根智明, “PERT/CPM” 日科技連, 1965
- [2] 都島, 薦田 “負荷山崩し機能を拡張した資源配分手法の提案” 電学論 C Vol.110-C, No.2, P.P.63-9, 1990
- [3] D.J.Hoijtmt:P.B.Luh:K.R.Pattipati, “A Practical Approach to Job-Shop Scheduling Problems” IEEE transactions on robotics and automation Vol.9, No.1, Feb 1993