

スケジューリング最適化ソルバー OptSeq と その応用

野々部宏司（法政大学）

概要

- 背景・目的
- 資源制約付きプロジェクトスケジューリング問題
(Resource-Constrained Project Scheduling Problem, RCPSP)
- アルゴリズムの概要
- 適用例

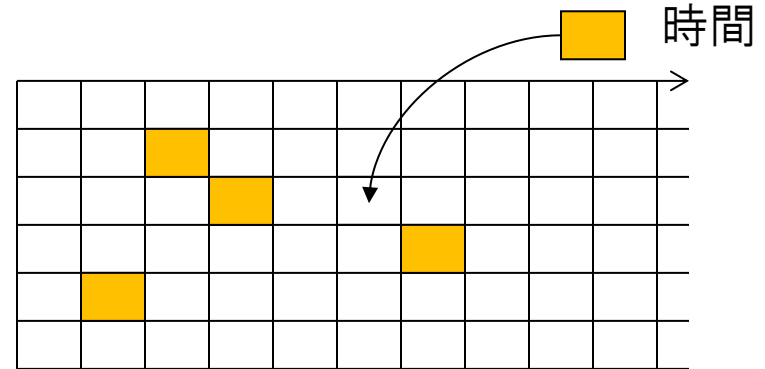
スケジューリング問題

- **限られた資源**（人・原材料・資金・場所など）のもとで複数の**作業**（または仕事・活動・工程）を行うとき、それぞれの作業を**いつ行うか決定**するとともに、時間軸を考慮しながら**資源配分**を行う問題
 - 割当てタイプ
 - 順序付けタイプ
 - 複合タイプ

スケジューリング問題

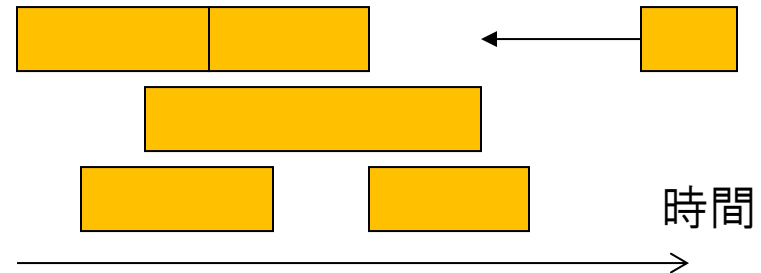
□ 割当てタイプ ⇒ MIPソルバー, WCSPソルバー (SCOP)

- 時間割作成
- ナーススケジューリング
- スポーツスケジューリング
- ...



□ 順序付けタイプ ⇒ RCPSPソルバー (OptSeq)

- ジョブショップ型の生産スケジューリング
- プロジェクト・スケジューリング
- ...



□ 複合タイプ

目的

- スケジューリング問題（順序付け＋資源配分）に特化した最適化ソルバー（スケジューリングエンジン）の開発
- 方針
 - 広い適用範囲
 - 対象をスケジューリング問題に限定しながらもある程度の汎用性を確保
 - ⇒ 資源制約付きスケジューリング問題（RCPSP）の拡張モデル
 - 計算時間・解の質の両面で実用に耐え得る性能
 - 厳密な意味での最適化は追求しない
 - ⇒ メタヒューリスティクスを利用
 - 利用の容易性
 - ソルバーを実行可能なソフトウェアとして提供

RCPSP : 基本モデル

- (再生可能) 資源 $r \in R$: 機械・原材料・工具・マンパワー・...

- 時間 $[t-1, t)$ の利用可能量 : $K_{r,t}$

- 作業 (活動, 工程) $i \in I$

- 所要時間 : d_i (中断不可)

- 必要資源量 : $k_{i,r}$

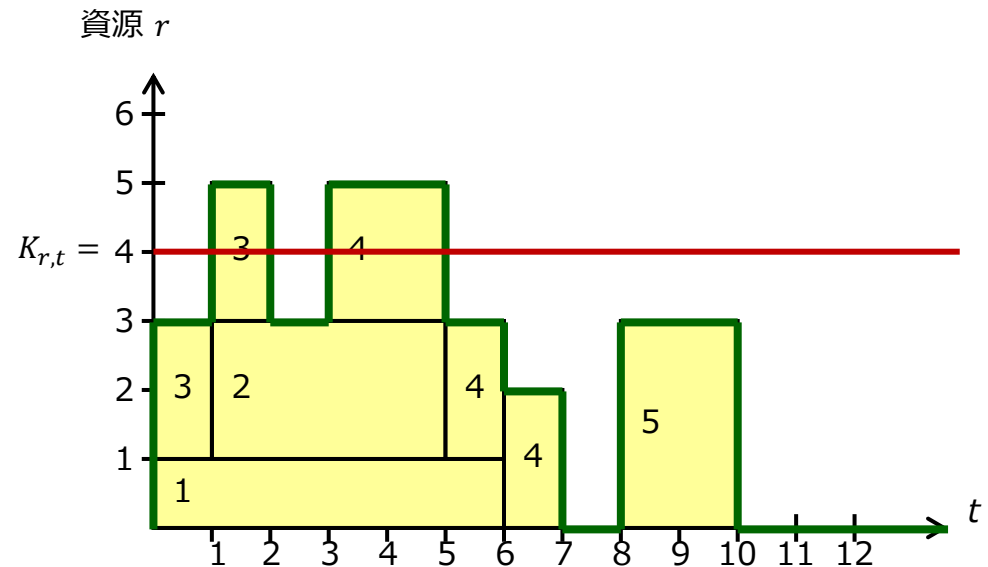
- **開始時刻** (決定変数) : s_i

- 先行制約 $i \rightarrow j$

- $s_i + d_i \leq s_j$

- 目的関数

- 最大完了時刻 $\max_i s_i + d_i$ (最小化)



RCPSP : 拡張モデル

(+) : 追加
(→) : 拡張

□ (再生可能) 資源 $r \in R$: 機械・原材料・工具・マンパワー・...

- 時間 $[t-1, t)$ の利用可能量 : $K_{r,t}$

(+) 再生不可能資源
(+) 状態変数

□ 作業 (活動, 工程) $i \in I$

- 所要時間 : d_i (中断不可)
- 必要資源量 : $k_{i,r}$
- **開始時刻** (決定変数) : s_i

(→) 中断可能, 最大中断時間
(→) 必要資源量が処理経過とともに変化
(+) 処理モード

□ 先行制約 $i \rightarrow j$

- $s_i + d_i \leq s_j$

(→) 時間制約 (一般化先行制約), 最大遅延時間

□ 目的関数

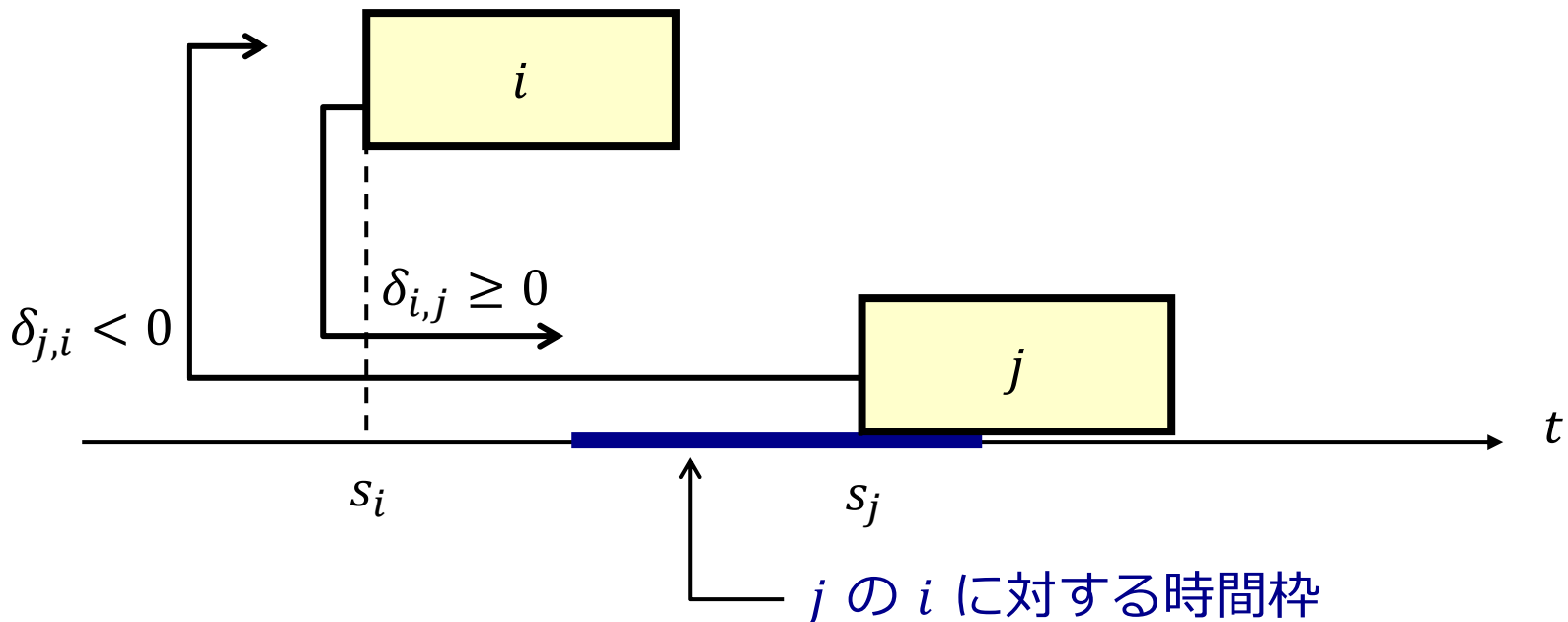
- 最大完了時刻 $\max_i s_i + d_i$ (最小化)

(→) 重み付き納期遅れ和
(+) 再生不可能資源制約違反度

時間制約（一般化先行制約）

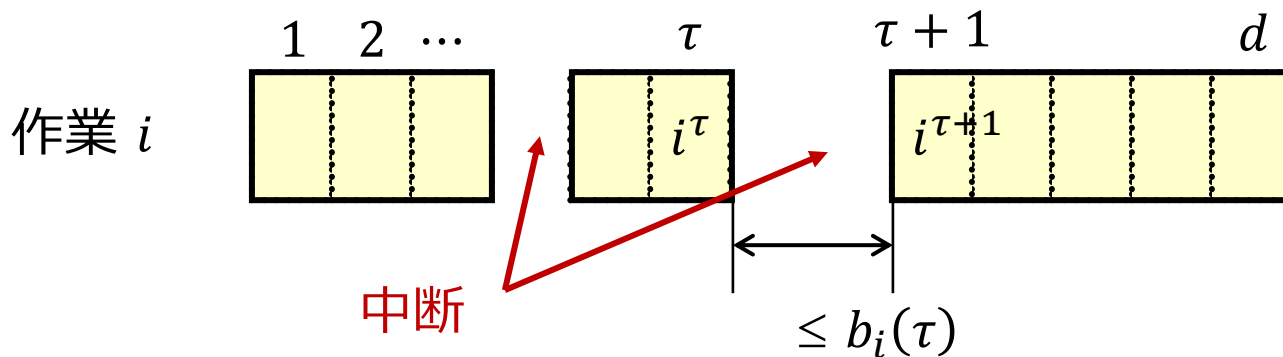
□ $s_j - s_i \geq \delta_{i,j}$

- $\delta_{i,j} \geq 0$ のとき, 最小遅延時間 ($\delta_{i,j} = d_i$ のとき, 通常の先行制約 $i \rightarrow j$)
- $\delta_{i,j} < 0$ のとき, 最大遅延時間
- 2 作業間の相対的な位置関係に関する制約を記述可能



作業の中断

- **作業の中断, 処理の再開が可能**
- 「所要時間 d の作業 i 」 = 「 d 個の小作業 $i^1, i^2, \dots, i^d$ 」
 - 小作業間で中断可能
 - 各小作業 i^τ の開始時刻 s_i^τ が決定変数
- $i^\tau, i^{\tau+1}$ 間の最大中断時間 $b_i(\tau)$ を設定可能
 - $b_i(\tau) = 0$ であれば中断不可
- **中断中も資源を消費可能**
 - 所要時間可変の作業を記述可能



処理モード

- **作業の処理方法** (= 所要時間・最大中断時間・必要資源量) を複数指定可能
 - 各作業は**必ずいずれか1つの処理モードで処理**
⇒ 各作業について, 処理モードと開始時刻を決定
- 作業 i とその処理モード $m \in M_i$ に対して, 0-1変数 $x_{i,m}$
 - i が m で処理されるとき (されないとき) $x_{i,m} = 1$ ($x_{i,m} = 0$)

$$\sum_{m \in M_i} x_{i,m} = 1$$

再生不可能資源制約

□ 再生不可能資源制約

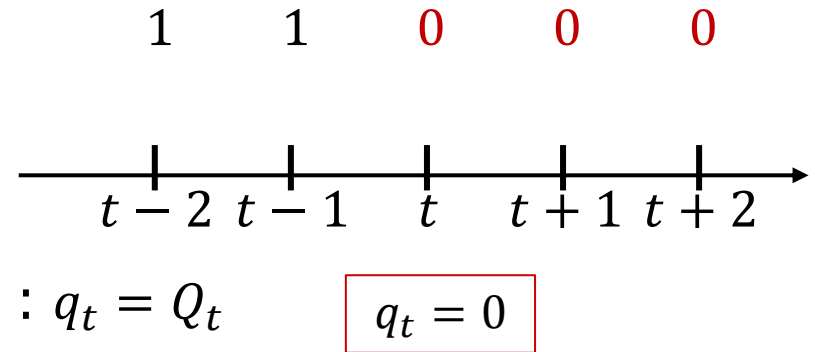
$$\sum_i \sum_{m \in M_i} a_{i,m} x_{i,m} \leq b$$

- $x_{i,m} = 1 \Leftrightarrow$ 作業 i が処理モード $m \in M_i$ で処理される
- **処理モードの組合せに関する制約（作業の開始時刻に依存せず）**
- 係数 $a_{i,m}$, 右辺定数 b は負でもよい
- **考慮制約として定義可能**
 - 「重み付き納期遅れ和 + 違反度の総和」を最小化
 - 処理モードの「好ましさ」を記述可能

状態変数

□ 時間の経過とともに値 (=状態) が変化

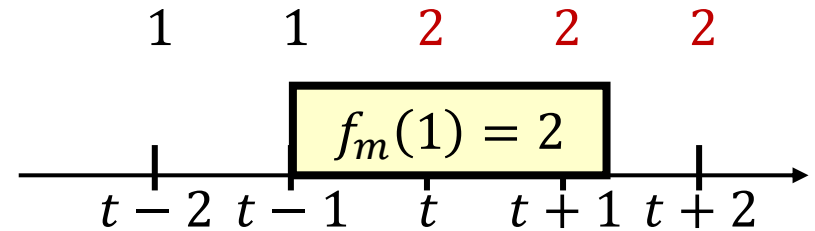
- Z : 状態の集合
- $q_t \in Z$: 時刻 t における状態
- 以下の場合を除き, $q_t = q_{t-1}$
 - 時刻 t における状態が予め指定 : $q_t = Q_t$



状態変数

□ 時間の経過とともに値 (=状態) が変化

- Z : 状態の集合
- $q_t \in Z$: 時刻 t における状態
- 以下の場合を除き, $q_t = q_{t-1}$
 - 時刻 t における状態が予め指定 : $q_t = Q_t$
 - 状態を変化させる処理モード m で作業が時刻 $t-1$ に開始 :
 $q_t = f_m(q_{t-1})$
 - $f_m: Z' \rightarrow Z$ は状態の遷移を定義する関数 ($Z' \subseteq Z$)
 - 処理モード m に状態遷移関数が定義されている場合,
 - 開始時刻 t の状態に応じて状態が変化
 - $q_t \notin Z' \Rightarrow$ 時刻 t に処理モード m で処理を開始することはできない



状態変数

□ 時間の経過とともに値（=状態）が変化

- Z : 状態の集合
- $q_t \in Z$: 時刻 t における状態
- 以下の場合を除き, $q_t = q_{t-1}$
 - 時刻 t における状態が予め指定 : $q_t = Q_t$
 - 状態を変化させる処理モード m で作業が時刻 $t - 1$ に開始 :
 $q_t = f_m(q_{t-1})$
- 同一時刻に複数の状態変化を発生させることはできない

□ 処理モードの選択が機械や設備の状態に依存する状況を記述可能

- 段取り作業など

アルゴリズムの概要（1）

- リストスケジューリング+バックトラックによるスケジュール生成
 - 作業の順列（リスト）にしたがって，各作業の開始時刻を決定
 - **再生可能資源制約・時間制約・状態変数に関する制約を満たす最早開始時刻に割付け**（フォワード・スケジューリング）
 - **割付け可能な時刻が存在しない場合はバックトラック**
 - 先に割り付けた作業の開始時刻を遅らせる
 - バックトラック回数には上限を設定．上限に達した場合は生成失敗
 - 処理モードは予め固定，もしくは割付け時に自動選択
 - 再生不可能資源制約（絶対制約）は満たされていることを想定

アルゴリズムの概要（2）

- リストスケジューリング+バックトラックによるスケジュール生成
 - 作業の順列（リスト）にしたがって、各作業の開始時刻を決定
 - 再生不可能資源制約（絶対制約）は満たされていることを想定
- （近似）最適な「リストと処理モードの組合せ」をタブー探索によって発見
 - 近傍操作

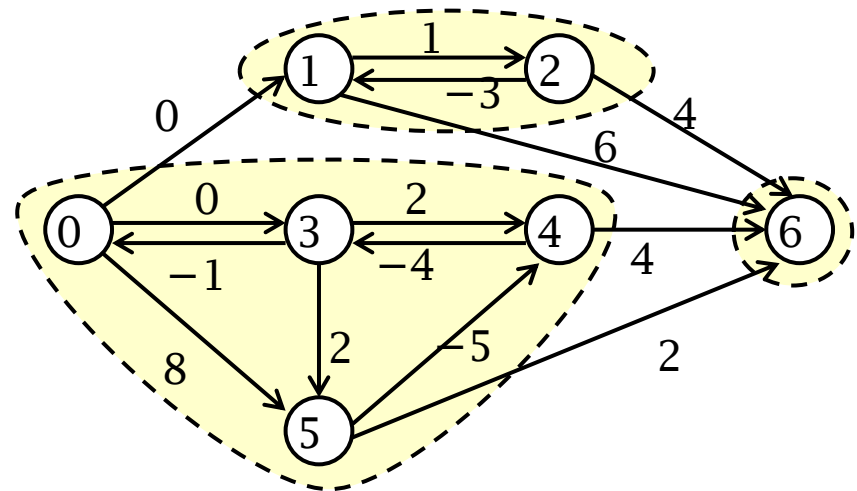
ある1つの作業について、リスト内の位置・処理モードのいずれかを変更

 - 処理モードの変更によって再生不可能資源制約（絶対制約）を満たさなくなる場合は、他の作業の処理モードも変更

探索の効率化 (1)

探索空間の制限

- Activity-on-node project network の利用
 - 作業集合 = 節点集合
 - 時間制約 $s_j - s_i \geq \delta_{i,j} \Leftrightarrow$ 長さ $\delta_{i,j}$ の枝 (i,j)
- 同一強連結成分内の作業が連続して出現するリストのみを探索**
⇒ バックトラック回数を減らす効果



探索の効率化 (2)

□ 近傍の削減

□ クリティカルグラフの活用

リストスケジューリングの計算過程において,

「作業 i のために作業 j の開始時刻を早めることができない」

作業の組 (i, j) を記録

適用例

□ 配船計画

天然資源の輸入．隻数制限・同時着船隻数制限・輸入量制限・貯蔵量制限

□ 作業場所を考慮した生産スケジューリング

作業開始日・作業場所を決定

□ 作業の中断を許す生産スケジューリング

手動処理＋自動処理．休憩時間において手動処理は中断

□ 鑄造日程計画

資源（溶解炉・組杵・ピット・クレーン…），休日，仕掛品，溶解条件，…

□ 製品開発プロジェクトスケジューリング

タスクの同期，納期遅れ最小化

□ …

まとめ

- **実務利用を念頭に置いた汎用スケジューラの開発**
 - 拡張RCPSPに対するメタヒューリスティックアルゴリズム
- **今後の方向性**
 - 性能向上
 - 適用範囲拡大：標準問題の拡張，その他の標準問題
 - 利用指針：標準問題の選定，モデリング