

# 不確定作業を考慮したバッチスケジューリング

静岡大学  
Dalian Maritime University  
成蹊大学  
05000348 静岡大学

林多希与 HAYASHI Takito  
唐亮 TANG Liang  
加藤晴康 KATO Haruyasu  
\*呉偉 WU Wei

## 1. はじめに

スケジューリングを行うにあたり、設備不良による作業の遅れ・不良品発生による作業の追加等を考慮することが難しい為、時間を多めに見積もり、スケジュールの衝突を回避をする場面が多々あるが、余分に見積もってしまった際は効率の低下に繋がる。

本研究では、処理時間が不確かな作業が発生する際の最適スケジューリングを考える。作業が通常通り実行した際の基準処理時間に加え、予期しない不良品の発生や作業の遅れ、歩留まりロス等による追加処理時間を考慮し、スケジュールを組んで実際に実行したとき、最悪なタイミングで追加処理時間が発生した場合でも、より早く作業を終了できるバッチ処理スケジューリング問題をロバスト最適化問題として考える。問題における最悪シナリオ（与えられた解の目的関数値を最大にするシナリオ）の性質を明示し、ロバスト最適化問題を制約付き最短路問題へ帰着させ、最適解を求める多項式時間アルゴリズムを示す。

## 2. 総完成時刻最小化単一機械直列バッチスケジューリング問題

本研究の基礎となる総完成時刻最小化単一機械直列バッチスケジューリング問題は、一台のマシンに対し、与えられた複数のジョブをグループ（バッチ）分けし、全ジョブをなるべく早く処理完了とするスケジューリング問題において、バッチ処理間で段取り時間が発生する状況下で、ジョブのバッチ分けと、各バッチの実行順序を求める問題である。本研究では、全ジョブが時刻0から着手可能、各バッチ処理前の段取り時間は固定時間とし、バッチに属するジョブを直列で処理する直列バッチ（s-batch）スケジューリング問題を対象とする。

ジョブ集合  $J = \{1, 2, \dots, n\}$ 、各ジョブの処理時間  $p_j$ 、バッチの段取り時間  $s$  が入力として与え

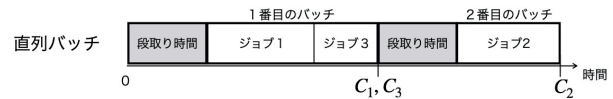


図 1: 直列バッチスケジューリングのチャート

られる。バッチ内全てのジョブが終了した時刻をバッチの終了時刻と呼び、各ジョブ  $j$  の完成時刻（滞留時間、フロー時間） $C_j$  は所属するバッチの終了時刻とする。総完成時刻最小化単一機械直列バッチスケジューリング問題  $1 | \text{s-batch} | \sum C_j$  は評価値  $\sum_{j \in J} C_j$  が最小となるスケジュールを求める問題である。

ジョブ実行順序を  $\pi$ 、各バッチに含まれるジョブ数を表す配列を  $\varphi$  とし、問題  $1 | \text{s-batch} | \sum C_j$  の解表現は、 $\theta = (\pi, \varphi)$  と表すことができる。例として、 $n = 3$  であるときの解  $\pi = (1, 3, 2)$ 、 $\varphi = (2, 1)$  のスケジュールチャートを図 1 に示す。

問題  $1 | \text{s-batch} | \sum C_j$  は Coffman ら [1] によって  $O(n \log n)$  の厳密解法が提案されている。

## 3. 不確かさの付与

より現実的な状況にする為、処理時間に不確かさを考慮する。任意のジョブ  $j$  に対して、予定通りに実行した際の処理時間（基準処理時間）を  $\bar{p}_j$ 、歩留まりロスの上限または遅延作業発生数を  $\Gamma$ （歩留まりロス率・遅延作業発生率  $\Gamma/n$ ）とする。実世界では、基準処理時間が大きいほど、追加処理時間が長くかかる傾向があるため、本研究では単調非減少関数  $f$  が与えられる下で、ジョブ  $j$  の追加処理時間を  $f(\bar{p}_j)$  とし、処理時間に関する不確定集合を以下の様におく：

$$P = \left\{ (p_1, p_2, \dots, p_n) \mid \forall j \in J, p_j = \bar{p}_j + \gamma_j f(\bar{p}_j), \right. \\ \left. 0 \leq \gamma_j \leq 1; \sum_{j \in J} \gamma_j \leq \Gamma \right\}. \quad (1)$$

処理時間の取りうる集合を式 (1) と設定することで、ジョブの処理時間に関する不確かさが内在している場面を表現できる。実行可能解集合を  $\Theta$ 、シナリオ  $p \in P$  の下で問題  $1 | s\text{-batch} | \sum C_j$  の解  $\theta$  の評価値を  $g(p, \theta)$  とし、本研究対象となる  $s$ -batch のロバスト最適化問題を以下の様に表す：

$$\min_{\theta \in \Theta} \max_{p \in P} g(p, \theta). \quad (2)$$

解  $\theta$  における問題  $\max_{p \in P} g(p, \theta)$  の最適解を最悪シナリオと呼ぶ。最悪シナリオを求める問題に対して、以下の補題を示す：

**補題 1.** 解  $\theta$  が与えられるとき、最悪シナリオ  $\omega(\theta)$  を  $\mathcal{O}(n \log n)$  時間で求めることができる。

#### 4. 厳密解法

紙面の都合上、 $\Gamma \in \mathbb{N}$  の場合を例に説明する。基準処理時間  $\bar{p}$  の非減少順でソートしたジョブ順序を  $\pi$  とし、以下の補題を示す：

**補題 2.** 問題 (2) において、解  $(\hat{\pi}, \varphi^*)$  が最適となる  $\varphi^*$  が存在する。

補題 2 から、便宜上ジョブの番号順を  $\hat{\pi}$  とする (すなわち  $j = \hat{\pi}(j)$ )。ここで最適なバッチ分け  $\varphi^*$  を求める問題をネットワーク上のある制約付き最短路問題に帰着させるために、頂点集合を  $V = \{v_{i,\gamma} \mid \gamma \in \{0, 1, \dots, \Gamma\}, i \in \{\gamma + 1, \gamma + 2, \dots, n + 1 + \gamma - \Gamma\}\}$ 、辺集合を  $E = \{(v_{i,\gamma}, v_{i',\gamma'}) \mid i < i', 0 \leq \gamma' - \gamma \leq i' - i\}$  となる有向非巡回グラフ  $G = (V, E, c, l, u)$  を構築する。 $n = 5, \Gamma = 2$  の例を図 2 に示す。

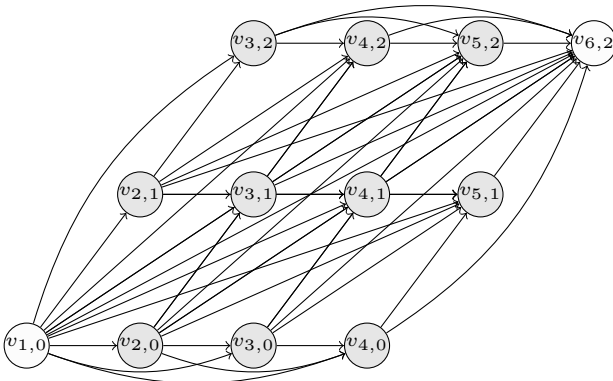


図 2:  $n = 5, \Gamma = 2$  のときのグラフ表現

辺  $e = (v_{i,\gamma}, v_{i',\gamma'}) \in E$  はジョブ  $i, i+1, \dots, i'-1$  のみを含むバッチを表現し、各ジョブ  $j$  の作業時間  $p_j$  が

$$p_j = \begin{cases} \bar{p}_j & \text{if } i \leq j \leq i' - \gamma' + \gamma - 1 \\ \bar{p}_j + \delta_j & \text{if } i' - \gamma' + \gamma \leq j \leq i' - 1 \end{cases}$$

となることを表す。辺  $e$  のコスト  $c_e$ 、上界  $u_e$ 、下界  $l_e$  を

$$c_e = (n + 1 - i) \left( s + \sum_{j=i}^{i'-1} \bar{p}_j + \sum_{j=i'-\gamma'+\gamma}^{i'-1} \delta_j \right),$$

$$l_e = \begin{cases} \min_{j \in J} (n - j + 1) \delta_j & \text{if } \gamma' - \gamma = i' - i \\ (n + 1 - i) \delta_{i'-\gamma'+\gamma-1} & \text{otherwise,} \end{cases}$$

$$u_e = \begin{cases} \max_{\gamma \in \{0, 1, \dots, \Gamma\}} (n - \gamma) \delta_{n-\Gamma+\gamma} & \text{if } \gamma' - \gamma = 0 \\ (n + 1 - i) \delta_{i'-\gamma'+\gamma} & \text{otherwise.} \end{cases}$$

として設定することで、問題 (2) は制約付き最短路問題 (constrained shortest path problem, CSPP)

$$\begin{aligned} & \underset{r}{\text{minimize}} && \sum_{e \in r} c_e \\ & \text{subject to} && \max_{e \in r} l_e \leq \min_{e \in r} u_e \\ & && r \text{ は } v_{1,0} \text{ から } v_{n+1,\Gamma} \text{ までのパス} \end{aligned}$$

としてモデル化することができる。帰着した CSPP に対して、 $\mathcal{O}(n^4 \Gamma^2)$  の厳密解法を提案した。

**定理 1.** CSPP の最適値は問題 (2) の最適値である。

CSPP を解くことで求めた最適解の情報をもとにグラフ  $G$  の性質を利用し、問題 (2) の最適なスケジュールに変換できる。

上記の様な解法を基本とし、問題 (2) で  $\Gamma \in \mathbb{R}^+ \setminus \mathbb{N}$  の場合も  $\Gamma \in \mathbb{N}$  のときと同じ計算量で最適解を求めることを示した。また、関数  $f$  が定数関数の場合は  $\mathcal{O}(n \log n)$  の厳密解法を提案した。

#### 参考文献

- [1] EG Coffman, Mihalis Yannakakis, MJ Magazine, and Cipriano Santos. Batch sizing and job sequencing on a single machine. *Annals of Operations Research*, Vol. 26, No. 1, pp. 135–147, 1990.