

# 処理時間の不確かさを考慮した総滞留時間最小化単一機械 バッチスケジューリング

林 多希与<sup>†</sup>  
静岡大学<sup>†</sup>

唐 亮<sup>‡</sup>  
大連海事大学<sup>‡</sup>

呉 偉<sup>§</sup>  
静岡大学<sup>§</sup>

## 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ため、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。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時、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、その下で、基準処理時間と追加処理時間の上限が比例関係となる状況を含む特殊ケースにおいて、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 総滞留時間最小化単一機械バッチスケジューリング問題

本研究の基礎となる総滞留時間最小化単一機械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



図 1: バッチスケジューリングのチャート

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と、バッチに属するジョブを全て並列で処理する並列バッチ (p-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  $s$  が入力として与える。ジョブ  $j$  の完成時刻（滞留時間, フロー時間）を  $C_j$  とし、総滞留時間最小化単一機械直列バッチ (resp. 並列バッチ) スケジューリング問題  $1 | s\text{-batch} | \sum C_j$  (resp.  $1 | p\text{-batch} | \sum C_j$ ) 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。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、ジョブ完成時刻は所属するバッチの終了時刻とする。

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、単一機械バッチスケジューリング問題の解表現は、 $\theta = (\pi, \varphi)$  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  $1 | s\text{-batch} | \sum C_j$  と  $1 | p\text{-batch} | \sum C_j$  は共に 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  $\bar{p}_j$ , 不良品の発生や作業遅延等による追加処理時間の

Single machine batch scheduling to minimize the total flow time under uncertain processing time

<sup>†</sup>Takito HAYASHI, Shizuoka University

<sup>‡</sup>Liang TANG, Dalian Maritime University

<sup>§</sup>Wei WU, Shizuoka University

上限を  $\delta_j$ , 歩留りロスの上限または遅延作業発生数を  $\Gamma$  (歩留りロス率・遅延作業発生率  $\Gamma/n$ ) とし, 処理時間に関する不確定集合を以下の様におく:

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

実行可能解集合を  $\Theta$ , シナリオ  $p \in P$  の下で問題  $1 \mid \text{s-batch} \mid \sum C_j$  (resp.  $1 \mid \text{p-batch} \mid \sum C_j$ ) の解  $\theta$  の評価値を  $g^s(p, \theta)$  (resp.  $g^p(p, \theta)$ ) とし, 本研究対象となる s-batch と p-batch のロバスト最適化問題を以下の様に表す:

$$\min_{\theta \in \Theta} \max_{p \in P} g^s(p, \theta), \quad (1)$$

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

紙面の都合上, 問題性質及び解法説明は問題 (1) のみを対象として行う. 解  $\theta$  における問題  $\max_{p \in P} g^s(p, \theta)$  の最適解 (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  $\omega^s(\theta)$  を  $\mathcal{O}(n \log n)$  時間で求めることができる.

#### 4 特殊ケースにおける厳密解法

実世界では, 基準処理時間が大きいほど, 追加処理が発生する際, 追加処理時間が長くなる傾向がある. よって, 任意の  $i, j (i < j)$  に対して,  $\bar{p}_{\pi(i)} \leq \bar{p}_{\pi(j)}$  と  $\delta_{\pi(i)} \leq \delta_{\pi(j)}$  を満たすジョブ順列  $\pi$  が存在するときのケースに注目し, 紙面の都合上,  $\Gamma \in \mathbb{N}$  の場合を例に, 以下の補題を示した上で,  $\pi$  が存在するときの厳密解法を示す.

**補題 2.** 順列  $\pi$  が存在するとき, 問題 (1) において, 解  $(\pi, \varphi^*)$  が最適となる  $\varphi^*$  が存在する.

補題 2 より, 便宜上ジョブの番号順を  $\pi$  とする (すなわち  $j = \pi(j)$ ). 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経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

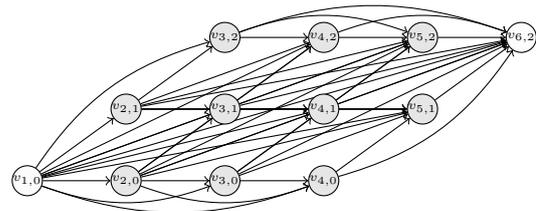


図 2:  $n = 5, \Gamma = 2$  のときのグラフ  $G$

$G = (V, E, c, l, u)$  を構築する ( $n = 5, \Gamma = 2$  の例を図 2 に示す). 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. 辺コスト  $c_e$  は辺  $e$  のバッチの処理時間とそのバッチ以降に実行されるジョブの個数の積を示す. 最悪シナリオでの評価値を保証するために各辺  $e$  に  $l_e, u_e$  を設定し (紙面の都合上詳細を省略する), 最適なバッチ分け  $\varphi^*$  を求める問題は制約付き最短経路問題

$$\text{minimize}_r \sum_{e \in r} c_e \quad (3)$$

$$\text{subject to } \max_{e \in r} l_e \leq \min_{e \in r} u_e \quad (4)$$

$$r \text{ は } v_{1,0} \text{ から } v_{n+1,\Gamma} \text{ までのパス} \quad (5)$$

としてモデル化できる. 制約付き最短経路問題 (3)–(5) に対して,  $\mathcal{O}(n^4 \Gamma^2)$  の厳密解法を提案した.

上記の様な解法を基本とし, 問題 (1) で  $\Gamma \in \mathbb{R}^+ \setminus \mathbb{N}$  の場合も  $\Gamma \in \mathbb{N}$  のときと同じ計算オーダーで最適解を求めることを示した上, 問題 (2) のとき  $V, E$  の設定は変えずに,  $c, u, l$  の設定を変える事で,  $\mathcal{O}(n^4 \Gamma)$  にて最適解を求めるアルゴリズムを提案した.

加えて, 全ての追加処理時間上限が同じである ( $\delta_j = \delta$ ,  $\pi$  が  $\bar{p}_j$  の非増加順) ときの特殊ケースでは  $\mathcal{O}(n \log n)$  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.