処理時間の不確かさを考慮したバッチスケジューリング

呉　偉，林　多希与，加藤　晴康

総完了時刻最小化単一機械バッチスケジューリング問題は多項式時間で厳密に解けることが知られている．本稿では，不良品の発生や作業員の欠員など，不確かな事象発生を考慮し，通常どおり実行できた場合の基準処理時間，不良品発生などによる不測の追加処理時間の上限，不測事象の発生頻度が与えられた場合のバッチスケジューリングをロバスト最適化問題として考える．本問題においては，直列バッチ処理と並列バッチ処理の各々の場合を扱い，多項式時間の厳密解法を提案する．また，不測事象発生による追加処理時間が全作業に等しくかかる直列バッチの場合においての高速な厳密解法を示す．

キーワード：バッチスケジューリング，ロバスト最適化，総完了時刻最小化

1. はじめに

スケジューリングを行うにあたり，急用による作業員の欠員，設備不良による作業の遅れ，不良品発生による処理の追加などの不測事象を考慮し，時間を多めに見積もり，スケジュールの衝突を回避する場面が多々ある．結果として，作業が滞りなく処理できたとしても，無駄な時間，余計なエネルギーの排出など，効率性の低下に繋がる場合がある．

本稿では，段取り時間を考慮したバッチスケジューリング問題に焦点を当てる．一台のマシンに対し，与えられた複数の作業（ジョブ）をグループ（バッチ）分けし，バッチとバッチの間に一定の段取り時間が発生する設定の下，全ジョブをなるべく早く処理完了とするジョブのバッチ分けと，ジョブの処理順序を求める問題を考える．

ここでは本問題定義に加え，処理に不確かなジョブが発生する場合でも，効率の良いスケジュールを作り出すアルゴリズムについて考える．通常どおり処理した場合の基準処理時間に加え，予期しない事象（不良品の発生やジョブの処理遅延，歩留まりロスなど）による追加処理時間および予期しない事象の発生頻度を考慮し，起こりうるすべてのシナリオの下で，ロバストなスケジュールを作成する方法の提案を行う．

ご　い，はやし　たきと
静岡大学
〒 432–8561 静岡県浜松市中区城北 3–5–1
goi@shizuoka.ac.jp
hayashi.takito.17@shizuoka.ac.jp
かとう　はるやす
成蹊大学
〒 180–8633 東京都武蔵野市吉祥寺北町 3–3–18

2. 問題定義

2.1 総完了時刻最小化単一機械バッチスケジューリング問題

総完了時刻最小化単一機械バッチスケジューリング問題とは，ジョブ集合 $J=\{1,2,\ldots,n\}$，各ジョブ j の処理時間 p_j と段取り時間 d が与えられるとき，1 台のマシンに対し，各ジョブ j の完了時刻（滞留時間，フロー時間）C_j の総和が最小となるジョブの処理順列とジョブのグループ（バッチ）分けを求める問題である．

本稿では，全ジョブが時刻 0 から着手可能，各バッチの段取り時間は一律と仮定し，バッチに属するジョブの処理方式として，直列で処理する**直列バッチ** (serial batch, s-batch) と，並列で処理する**並列バッチ** (parallel batch, p-batch) の 2 種類を考える．それらに対応する総完了時刻最小化単一機械バッチスケジューリング問題を $1\mid\text{s-batch}\mid\sum C_j$ と $1\mid\text{p-batch}\mid\sum C_j$ と書く．

例として，複数商品を発注する会社とそれらの商品を直列で（resp. 並列で）生産する工場を考える．工場では，決められた発送タイミングで，そのタイミングまでに完成した商品をまとめて発送する．この条件下で，発注会社ができるだけ早く全商品を手に入れたい問題を考える．この例では工場側の生産順序計画と発送計画を合わせた問題を $1\mid\text{s-batch}\mid\sum C_j$ (resp. $1\mid\text{p-batch}\mid\sum C_j$) として考えられる．

ジョブ処理順列を π，各バッチに含まれるジョブ数を表すベクトルを φ とすると，単一機械バッチスケジューリング問題の解は，(π,φ) で表すことができる．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．

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

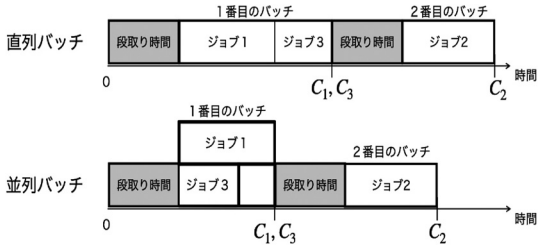

図 1　バッチスケジュールのチャート

各ジョブの処理時間 $p = (p_1, p_2, \ldots, p_n)$ が与えられた下で問題 $1 \mid \text{s-batch} \mid \sum C_j$ (resp. $1 \mid \text{p-batch} \mid \sum C_j$) の解 (π, φ) の評価値を $g^{\mathrm{s}}(p, \pi, \varphi)$ (resp. $g^{\mathrm{p}}(p, \pi, \varphi)$) とする．すべてのジョブ処理順列の集合を Π, バッチ分けの実行可能解の集合を Φ とし，問題 $1 \mid \text{s-batch} \mid \sum C_j$ と $1 \mid \text{p-batch} \mid \sum C_j$ は

$$1 \mid \text{s-batch} \mid \sum C_j : \quad \min_{\pi \in \Pi,\, \varphi \in \Phi} g^{\mathrm{s}}(p, \pi, \varphi),$$
$$1 \mid \text{p-batch} \mid \sum C_j : \quad \min_{\pi \in \Pi,\, \varphi \in \Phi} g^{\mathrm{p}}(p, \pi, \varphi)$$

と定義できる．

問題 $1 \mid \text{s-batch} \mid \sum C_j$ と $1 \mid \text{p-batch} \mid \sum C_j$ は共に Coffman et al. により $\mathcal{O}(n \log n)$ 時間の厳密解法が提案されている [1].

2.2　不確かさの付与

実践の世界では，設備不良による作業遅延や，不良品発生などの不測の事態による追加作業が発生することで処理が予定どおりに行われない場合が多々ある．

本稿では処理時間の不確かさを新たに考慮する．そのため，歩留りロスまたは遅延ジョブ発生数の上限を扱うことのできる Bertsimas and Sim が提案した不確定集合 [2] を用いる．

ジョブ j に対して，予定どおりに処理した際の処理時間（基準処理時間）を $\bar{p}_j$，不良品の発生やジョブ遅延などによる追加処理時間の上限を δ_j，歩留りロスまたは遅延ジョブ発生数の上限数を Γ（歩留りロス率・遅延ジョブ発生率 Γ/n）とする．ここで，Γ は整数とは限らないことに注意しておく．処理時間に関する不確定集合を以下のようにおく：

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

ここで，基準処理時間が大きければ，追加処理時間の上限も大きくなる，すなわち，$\bar{p}_i < \bar{p}_j$ の場合，$\delta_i \le \delta_j$ と仮定する．この設定は多くの実践現場で想定される仮定と考える．

本稿で対象となるロバスト最適化問題 s-batch robust scheduling problem (S-RSP) と p-batch robust scheduling problem (P-RSP) は

$$\text{S-RSP:} \quad \min_{\pi \in \Pi,\, \varphi \in \Phi} \max_{p \in P} g^{\mathrm{s}}(p, \pi, \varphi), \tag{1}$$
$$\text{P-RSP:} \quad \min_{\pi \in \Pi,\, \varphi \in \Phi} \max_{p \in P} g^{\mathrm{p}}(p, \pi, \varphi) \tag{2}$$

と定義する．

3.　最適処理順序と最悪シナリオ

問題 $1 \mid \text{s-batch} \mid \sum C_j$ と $1 \mid \text{p-batch} \mid \sum C_j$ において，SPT (shortest processing time) 順が最適なジョブ処理順であることが知られている [1]．S-RSP と P-RSP においては，$\acute{\pi}$ を基準処理時間の非減少順 (shortest standard processing time, SSPT) とし，以下の補題が成り立つ：

補題 1.　S-RSP と P-RSP に対して，ジョブ処理順が SSPT 順 $\acute{\pi}$ となる最適解 $(\acute{\pi}, \varphi)$ が存在する．

補題 1 を利用することにより，S-RSP と P-RSP は (1) と (2) から

$$\text{S-RSP:} \quad \min_{\varphi \in \Phi} \max_{p \in P} g^{\mathrm{s}}(p, \acute{\pi}, \varphi), \tag{3}$$
$$\text{P-RSP:} \quad \min_{\varphi \in \Phi} \max_{p \in P} g^{\mathrm{p}}(p, \acute{\pi}, \varphi) \tag{4}$$

に書き換えることができる．

S-RSP (resp. P-RSP) に対して，解 $(\acute{\pi}, \varphi)$ が与えられるとき，$\max_{p \in P} g^{\mathrm{s}}(p, \acute{\pi}, \varphi)$ (resp. $\max_{p \in P} g^{\mathrm{p}}(p, \acute{\pi}, \varphi)$) の最適解を解 $(\acute{\pi}, \varphi)$ の最悪シナリオと呼ぶ．ジョブの処理順列が $\acute{\pi}$ であるとき，バッチ分け φ においてジョブ j を含むバッチから最終バッチまでのすべてのバッチに含まれるジョブ数の総和を $\theta(\varphi, j)$ と定義する．各ジョブ j に対して，値 $\theta(\varphi, j) \cdot \delta_j$ を非増加順でソートされたジョブ順列を $\acute{\pi}$ とし，$\acute{\pi}$ における j の順位を $\acute{\pi}(j)$ とする．S-RSP において，解 $(\acute{\pi}, \varphi)$ の最悪シナリオ $\hat{p}^{(\varphi)}$ が $\mathcal{O}(n \log n)$ 時間で構築できることを示す：

補題 2.　S-RSP において解 $(\acute{\pi}, \varphi)$ の最悪シナリオ $\hat{p}^{(\varphi)} = (\bar{p}_1 + \delta_1 \hat{\gamma}_1^{(\varphi)}, \bar{p}_2 + \delta_2 \hat{\gamma}_2^{(\varphi)}, \ldots, \bar{p}_n + \delta_n \hat{\gamma}_n^{(\varphi)})$ は以下の $\hat{\gamma}_j^{(\varphi)}$ を用いて定義できる：

$$\hat{\gamma}_j^{(\varphi)} = \begin{cases} 1 & \text{if } 1 \le \acute{\pi}(j) \le \lfloor \Gamma \rfloor \\ \Gamma - \lfloor \Gamma \rfloor & \text{if } \acute{\pi}(j) = \lfloor \Gamma \rfloor + 1 \\ 0 & \text{otherwise.} \end{cases}$$

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

次に，P-RSP に対しても解 $(\acute{\pi}, \varphi)$ の最悪シナリオ $\check{p}^{(\varphi)}$ が $\mathcal{O}(n \log n)$ 時間で構築可能であることを示す．解 $(\acute{\pi}, \varphi)$ が与えられるとき，各バッチの最終ジョブ（順列 $\acute{\pi}$ に添って処理したときの各バッチ内で最後に完了するジョブ）で構成されるジョブ集合を J' とし，J' の各ジョブ j を値 $\theta(\varphi, j) \cdot \delta_j$ の非増加順でソートすることで得られる順列を $\check{\pi}$ とする．$\check{\pi}$ における j の順位を $\check{\pi}(j)$ とする．

補題 3. P-RSP において解 $(\acute{\pi}, \varphi)$ の最悪シナリオ $\check{p}^{(\varphi)} = (\bar{p}_1 + \delta_1 \check{\gamma}_1^{(\varphi)}, \bar{p}_2 + \delta_2 \check{\gamma}_2^{(\varphi)}, \ldots, \bar{p}_n + \delta_n \check{\gamma}_n^{(\varphi)})$ は以下の $\check{\gamma}_j^{(\varphi)}$ を用いて定義できる：
$\Gamma \geq |\varphi|$ のとき

$$\check{\gamma}_j^{(\varphi)} = \begin{cases} 1 & j \in J' \\ 0 & \text{otherwise;} \end{cases}$$

$\Gamma < |\varphi|$ のとき

$$\check{\gamma}_j^{(\varphi)} = \begin{cases} 1 & j \in J' \text{ and } 1 \leq \check{\pi}(j) \leq \lfloor \Gamma \rfloor \\ \Gamma - \lfloor \Gamma \rfloor & j \in J' \text{ and } \check{\pi}(j) = \lfloor \Gamma \rfloor + 1 \\ 0 & \text{otherwise.} \end{cases}$$

補題 2 と補題 3 を適用することで，S-RSP と P-RSP は (3) と (4) から

$$\text{S-RSP:} \qquad \min_{\varphi \in \Phi} g^{\mathrm{s}}(\hat{p}^{(\varphi)}, \acute{\pi}, \varphi),$$

$$\text{P-RSP:} \qquad \min_{\varphi \in \Phi} g^{\mathrm{P}}(\check{p}^{(\varphi)}, \acute{\pi}, \varphi)$$

に書き換えることができる．

4. LU 制約付き最短路問題

5 節と 6 節で説明するように，S-RSP と P-RSP の部分問題は LU 制約付き最短路問題に帰着することができる．本稿で考える LU 制約付き最短路問題とは，有向非巡回グラフ (directed acyclic graph, DAG) $G = (V, E)$，始点 $v_{\mathrm{s}} \in V$，終点 $v_{\mathrm{t}} \in V$，および辺 $e \in E$ に対してコスト c_e，下限値 l_e，上限値 u_e $(l_e \leq u_e)$ が与えられるとき，LU 制約 (6) を満たす最短 s-t パス（v_{s} から v_{t} までのパス）を求める問題である：

$$\min_{\text{s-t パス } r} \quad \sum_{e \in r} c_e \tag{5}$$

$$\text{s.t.} \quad \max_{e \in r} l_e \leq \min_{e \in r} u_e. \tag{6}$$

制約付き最短路問題に関する研究は多く存在し [3, 4]，付加制約のあるいくつかの問題に関して NP 困難性が知られている．しかし，LU 制約付き最短路問題は既存の研究がなく，多項式時間アルゴリズムを提案する．

4.1 多項式時間厳密解法

グラフ G に対して，値 k が与えられるとき，辺集合

$$E_k = \{e \in E \mid l_e \leq k \leq u_e\}$$

の辺誘導部分グラフを G_k とする．任意の k に対して，以下の補題が成り立つ：

補題 4. グラフ G_k において，v_{s} と v_{t} が連結であれば，G_k 上の任意の s-t パスが LU 制約を満たす．

また，K_{l} と K_{u} を

$$K_{\mathrm{l}} = \{l_e \mid \forall e \in E\}$$
$$K_{\mathrm{u}} = \{u_e \mid \forall e \in E\}$$

とし，以下の補題が成り立つ：

補題 5. グラフ G 上の，LU 制約を満たす任意の s-t パス r に対して，$k' \in K_{\mathrm{l}}$ と $k'' \in K_{\mathrm{u}}$ が存在し，r がグラフ $G_{k'}$ と $G_{k''}$ に含まれる．

補題 4 と補題 5 により，K_{l}（もしくは K_{u}）の各要素 k に対する辺誘導部分グラフ G_k の最短路のなかで総コストが最小となるパスが LU 制約付き最短路問題の最適解となる．LU 制約付き最短路問題に対する $\mathcal{O}(|E|^2)$ 時間の厳密解法を以下に示す：

Require: グラフ $G = (V, E)$，始点 v_{s}，終点 v_{t}，各辺 e の c_e, l_e, u_e.
1: **if** $|K_{\mathrm{l}}| \leq |K_{\mathrm{u}}|$ **then**
2: $K \leftarrow K_{\mathrm{l}}$.
3: **else**
4: $K \leftarrow K_{\mathrm{r}}$.
5: **end if**
6: **for** $k \in K$ **do**
7: G_k を構築し，s-t 連結性を確認する．
8: G_k 上の最短路 r_k を求め，その長さが暫定解 r^* の長さより短い場合，$r^* \leftarrow r_k$.
9: **end for**

4.2 高速化

提案した多項式時間アルゴリズムの高速化を図る．LU 制約 (6) を満たすときの必要条件を利用し，グラフ縮小手法を考案する．

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

本案では，与えられるグラフ G と各辺 e の値 l_e, u_e に対し，以下の二つの問題を考える：

$$\min_{\text{s-t パス } r} \max_{e\in r} l_e, \tag{7}$$

$$\max_{\text{s-t パス } r} \min_{e\in r} u_e. \tag{8}$$

グラフ G が DAG であれば，s-t パスの始点を v_s，終点を j とするときの問題 (7)（resp. 問題 (8)）の部分問題の最適値を $f^\mathrm{l}(j)$ (resp. $f^\mathrm{u}(j)$) と定義する．任意の辺 e に対して，LU 制約を満たす s-t パスに含まれる必要条件を

$$l_e \le f^\mathrm{u}(v_\mathrm{t}) \text{ and } u_e \ge f^\mathrm{l}(v_\mathrm{t}) \tag{9}$$

と表し，条件 (9) を満たしていない，すなわち，

$$l_e > f^\mathrm{u}(v_\mathrm{t}) \text{ or } u_e < f^\mathrm{l}(v_\mathrm{t})$$

のすべての辺 e を前処理で，G から除くことができる．問題 (7) の最適値 $f^\mathrm{l}(v_\mathrm{t})$ と問題 (8) の最適値 $f^\mathrm{u}(v_\mathrm{t})$ は以下の動的計画法で $\mathcal{O}(|E|)$ 時間で計算できる：

$$\text{（初期値）}\quad f^\mathrm{l}(j) = \begin{cases} 0 & \text{if } j = v_\mathrm{s}, \\ +\infty & \text{otherwise}, \end{cases}$$

$$\text{（漸化式）}\quad f^\mathrm{l}(j) = \min_{i:(i,j)\in E} \left\{ f^\mathrm{l}(j), \max\left\{ l_{ij}, f^\mathrm{l}(i) \right\} \right\};$$

$$\text{（初期値）}\quad f^\mathrm{u}(j) = \begin{cases} 0 & \text{if } j = v_\mathrm{s}, \\ -\infty & \text{otherwise}, \end{cases}$$

$$\text{（漸化式）}\quad f^\mathrm{u}(j) = \max_{i:(i,j)\in E} \{ f^\mathrm{u}(j), \min\{u_{ij}, f^\mathrm{u}(i)\} \}.$$

f^l と f^u の漸化式はトポロジカル順序に沿って計算を行う．

5. S-RSP に対する厳密解法

最初にすべてのジョブを SSPT 順にソートし，π を得る．一般性を失わないため，本節からジョブの番号順を SSPT 順とする．すなわち，$\forall i, j \in J$，$i < j$ であれば $\bar{p}_i \le \bar{p}_j$ とする．

入力 Γ が整数のときの S-RSP が LU 制約付き最短路問題に帰着できることを示す．S-RSP の問題例を以下のように LU 制約付き最短路問題の問題例に変換する：

- 頂点集合 $V = \{v_{i,\gamma} \mid \gamma \in \{0,\dots,\Gamma\}, i \in \{\gamma+1,\dots,n+1+\gamma-\Gamma\}\}$,
- 始点 $v_\mathrm{s} = v_{1,0}$，終点 $v_\mathrm{t} = v_{n+1,\Gamma}$,
- 辺集合 $E = \{(v_{i,\gamma}, v_{i',\gamma'}) \mid i < i', 0 \le \gamma' - \gamma \le i' - i\}$.

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 の処理時間 p_j が

$$p_j = \begin{cases} \bar{p}_j & \text{if } i \le j \le i' - \gamma' + \gamma - 1 \\ \bar{p}_j + \delta_j & \text{if } i' - \gamma' + \gamma \le j \le i' - 1 \end{cases} \tag{10}$$

である．式 (10) により，追加処理を行うジョブ数は $\gamma' - \gamma$ 個であることがわかる．追加処理が行われない最後のジョブを $\hat{i} = i' - 1 - (\gamma' - \gamma)$，追加処理が行われる最初のジョブを $\check{i} = i' - (\gamma' - \gamma)$ とし，c_e, u_e, l_e を以下のように設定する：

$$c_e = (n+1-i)\left(d + \sum_{j=i}^{i'-1} \bar{p}_j + \sum_{j=\check{i}}^{i'-1} \delta_j \right);$$

$$l_e = \begin{cases} (n+1-i)\delta_{\hat{i}} & \text{if } \hat{i} \ge i, \\ 0 & \text{otherwise;} \end{cases}$$

$$u_e = \begin{cases} (n+1-i)\delta_{\check{i}} & \text{if } \check{i} \le i' - 1, \\ +\infty & \text{otherwise.} \end{cases}$$

ジョブ数 $n = 6$，遅延ジョブ発生数の上限 $\Gamma = 2$ のときの例を図 2 に示す．

構築したグラフにおいて，辺 $e = (v_{i,\lambda}, v_{i',\lambda'})$ は，以下のことを表す：

- ジョブ $i, i+1, \dots, i'-1$ を含むバッチがスケジューリングされる；
- ジョブ $j = i, i+1, \dots, i'-\lambda'+\lambda-1$ の処理時間は $\bar{p}_j$，ジョブ $j = i'-\lambda'+\lambda, i'-\lambda'+\lambda+1, \dots, i'-1$ の処理時間は $\bar{p}_j + \delta_j$ となる．

LU 制約を満たす $v_{1,0}$ から $v_{n+1,\Gamma}$ までのパスは最悪シナリオの下のスケジュールと対応しており，S-RSP は LU 制約付き最短路問題に帰着することができる．問題入力が

$$n = 6, \Gamma = 2, d = 1,$$
$$\bar{p} = \delta = (1, 1, 2, 2, 3, 3)$$

のときの対応関係を表す例を図 3 に示す．

変換後の LU 制約付き最短路問題の最適解から S-RSP の最適解を導くことができ，4.1 節の多項式時間厳密解法を利用することで，S-RSP が $\mathcal{O}(n^4\bar{\Gamma}^2)$ 時間で厳密に解けることになる．ここで，$\bar{\Gamma} = \min\{\Gamma, n-\Gamma\}$ である．

入力 Γ が整数でないときの S-RSP に対しても LU 制約付き最短路問題に帰着できることを示す．補題 2 より，すべての解 (π, φ) に対してちょうど一つの $\hat{\gamma}_j^{(\varphi)}$ が小数 $\Gamma - \lfloor\Gamma\rfloor$ となる最悪シナリオ $\hat{p}^{(\varphi)}$ が存在する．そ

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

の性質を利用することで，Γ が整数でないときの S-RSP も LU 制約付き最短路問題に帰着し，$\mathcal{O}(n^4\bar{\Gamma}^2)$ 時間の厳密解法の設計が可能となる．

すべてのジョブの追加処理時間の上限が一律である特殊ケース $(\forall i,j \in J, \delta_i = \delta_j)$ において，以下の補題が成り立つ：

補題 6. $\forall \varphi \in \Phi$，解 (π,φ) の最悪シナリオ $\tilde{p} = (\bar{p}_1+\delta_1\tilde{\gamma}_1, \bar{p}_2+\delta_2\tilde{\gamma}_2, \ldots, \bar{p}_n+\delta_n\tilde{\gamma}_n)$ を用いて定義できる：

$$\tilde{\gamma}_j = \begin{cases} 1 & \text{if } 1 \le \pi(j) \le \lfloor\Gamma\rfloor \\ \Gamma - \lfloor\Gamma\rfloor & \text{if } \pi(j) = \lfloor\Gamma\rfloor + 1 \\ 0 & \text{otherwise.} \end{cases}$$

補題 6 より，特殊ケースの s-batch 問題は

$$\min_{\varphi\in\Phi} g^{\mathrm{s}}(\tilde{p},\pi,\varphi). \tag{11}$$

になる．最悪シナリオ $\tilde{p}$ が解に依存しないため，古典的な $1 \mid \text{s-batch} \mid \sum C_j$ 問題として Coffman et al. の提案した $\mathcal{O}(n\log n)$ アルゴリズム [1] が適用できる．

6. P-RSP に対する厳密解法

補題 3 により，P-RSP を二つの部分問題

$$\text{P1:} \quad \min_{\varphi\in\Phi} g^{\mathrm{P}}(\tilde{p}^{(\varphi)},\pi,\varphi) \quad \text{s.t.} \quad |\varphi| \le \Gamma;$$

$$\text{P2:} \quad \min_{\varphi\in\Phi} g^{\mathrm{P}}(\tilde{p}^{(\varphi)},\pi,\varphi) \quad \text{s.t.} \quad |\varphi| > \Gamma$$

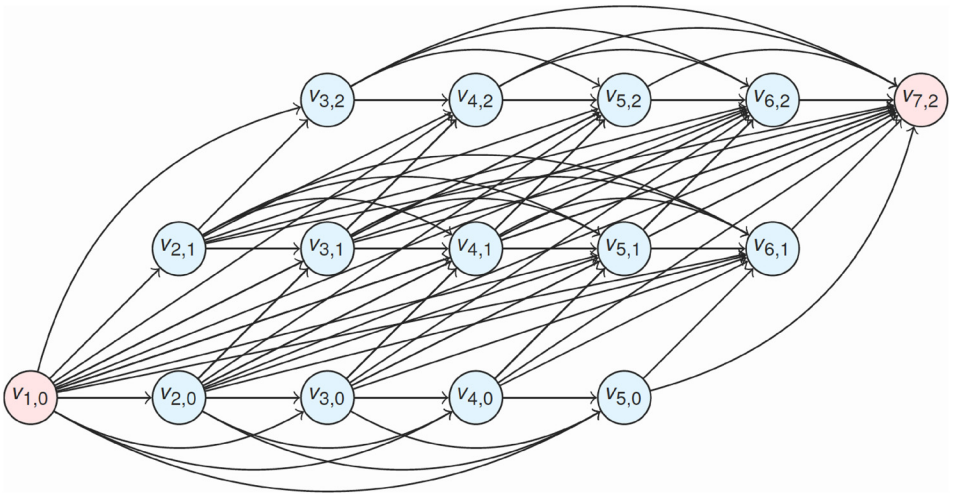

図 2 $n = 6$，$\Gamma = 2$ のときのグラフ表現

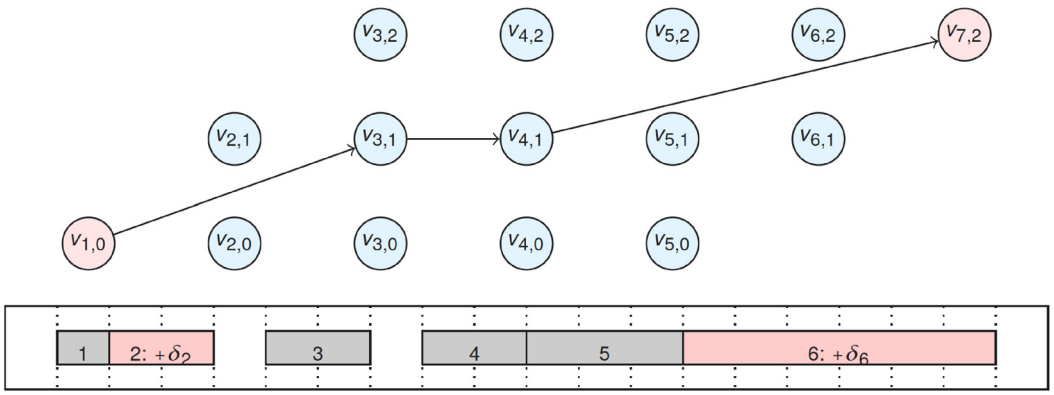

図 3 s-t パスとスケジュールの対応関係

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

に分割する．部分問題 P1 を最短路問題, 部分問題 P2 を LU 制約付き最短路問題に帰着することで，P-RSP に対し $\mathcal{O}(n^4\tilde{\Gamma})$ 時間で厳密に解けるアルゴリズムを構築できる．ここで，$\tilde{\Gamma} = \min\{\Gamma, \frac{n-\Gamma}{2}\}$ である．

7. まとめ

不確定ジョブを考慮したバッチスケジューリング問題を考えた．直列バッチ方式のときの問題と並列バッチ方式のときの問題に対して，最短路問題または LU 制約付き最短路問題に帰着できることを示し，多項式時間アルゴリズムを提案した．

参考文献

[1] E. G. Coffman, M. Yannakakis, M. J. Magazine and C. Santos, "Batch sizing and job sequencing on a single machine," *Annals of Operations Research*, **26**, pp. 135–147, 1990.

[2] D. Bertsimas and M. Sim, "The price of robustness," *Operations Research*, **52**, pp. 35–53, 2004.

[3] S. Irnich and G. Desaulniers, "Shortest path problems with resource constraints," *Column Generation*, G. Desaulniers, J. Desrosiers and M. M. Solomon (eds.), Springer, pp. 33–65, 2005.

[4] L. Lozano and A. Medaglia, "On an exact method for the constrained shortest path problem," *Computers & Operations Research*, **40**, pp. 378–384, 2013.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.