離散最適化基礎論

ジョブ・スケジューリングのアルゴリズム

第 10 回

ショップ・スケジューリング：基礎

岡本 吉央　(電気通信大学)

okamotoy@uec.ac.jp

2024 年 12 月 24 日

最終更新：2024 年 12 月 24 日　21:35

スケジュール (前半)

1. スケジューリング問題の分類 (10/1)
＊ 休み (出張) (10/8)
＊ 休み (体育祭) (10/15)
2. 整列による解法 (10/22)
3. 動的計画法 (10/29)
4. NP 困難性と計算量の分類 (11/5)
5. 計算複雑性による問題の分類 (11/12)
6. リスト・スケジューリング (11/19)

スケジュール (後半)

7.	先行制約：基礎	(11/26)
＊	休み (秋ターム試験)	(12/3)
8.	先行制約：多機械	(12/10)
9.	先行制約：他の半順序	(12/17)
10.	**ショップ・スケジューリング：基礎**	(12/24)
＊	休み (冬季休業)	(12/31)
11.	ショップ・スケジューリング：機械数が定数	(1/7)
12.	ショップ・スケジューリング：機械数が可変	(1/14)
13.	近似可能性と近似不可能性	(1/21)
14.	多項式時間近似スキーム	(1/28)
＊	なし	(2/4)

本日の内容

1. **ショップ・スケジューリングの用語**
2. $F2 \mid\mid C_{\max}$ のアルゴリズム
3. $F3 \mid\mid C_{\max}$ の強 NP 困難性
4. $J2 \mid\mid C_{\max}$ のアルゴリズム

ショップ・スケジューリング (shop scheduling) と呼ばず，
多段階スケジューリング (multi-stage scheduling) や
多工程スケジューリング (multi-operation scheduling) と
呼ぶこともある

ジョブと工程

ショップ・スケジューリングの設定

- 1 つのジョブは 1 つ以上の **工程** (operation) から成る
- 各工程を処理できる機械が決まっている
- 同じジョブを構成する複数の工程は，同時刻に処理できない

記法：O_{ij} = ジョブ J_j を構成する工程で，
機械 M_i で処理するもの

p_{ij} = 工程 O_{ij} の処理時間

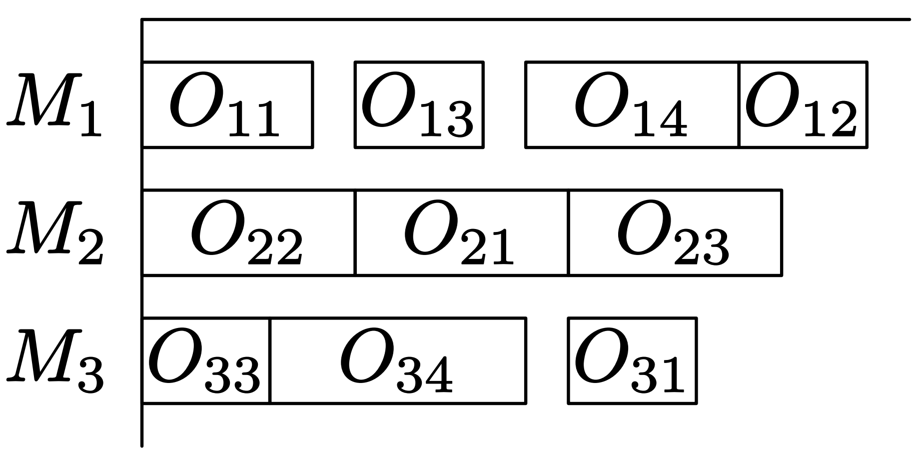
注：
1 つのジョブにおける複数の工程を同じ機械で処理する場合も考えられているが，この授業では扱わない

	J_1	J_2	J_3	J_4
M_1	O_{11}	O_{12}	O_{13}	O_{14}
M_2	O_{21}	O_{22}	O_{23}	
M_3	O_{31}		O_{33}	O_{34}

ジョブと工程：スケジュールの例

ショップ・スケジューリングの設定

- 1 つのジョブは 1 つ以上の **工程** (operation) から成る
- 各工程を処理できる機械が決まっている
- 同じジョブを構成する複数の工程は，同時刻に処理できない

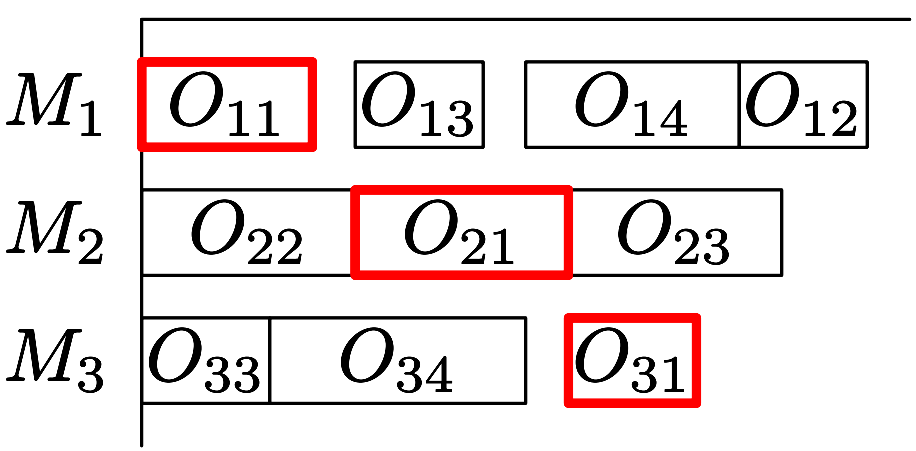


	J_1	J_2	J_3	J_4
M_1	O_{11}	O_{12}	O_{13}	O_{14}
M_2	O_{21}	O_{22}	O_{23}	
M_3	O_{31}		O_{33}	O_{34}

ジョブと工程：スケジュールの例

ショップ・スケジューリングの設定

- 1 つのジョブは 1 つ以上の **工程** (operation) から成る
- 各工程を処理できる機械が決まっている
- 同じジョブを構成する複数の工程は，同時刻に処理できない

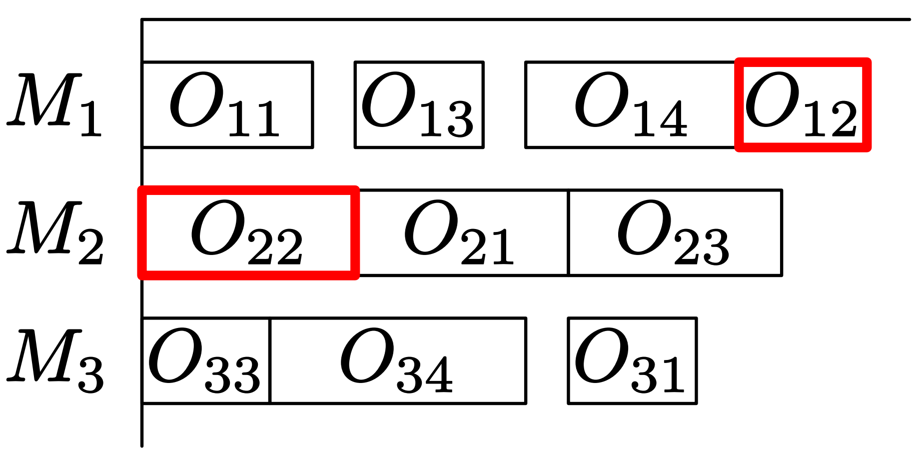


	J_1	J_2	J_3	J_4
M_1	O_{11}	O_{12}	O_{13}	O_{14}
M_2	O_{21}	O_{22}	O_{23}	
M_3	O_{31}		O_{33}	O_{34}

ジョブと工程：スケジュールの例

ショップ・スケジューリングの設定

- 1 つのジョブは 1 つ以上の **工程** (operation) から成る
- 各工程を処理できる機械が決まっている
- 同じジョブを構成する複数の工程は，同時刻に処理できない

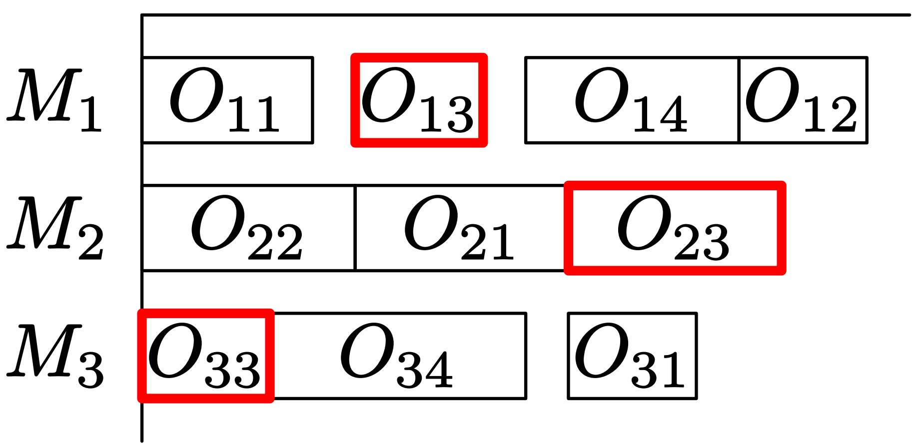


	J_1	J_2	J_3	J_4
M_1	O_{11}	O_{12}	O_{13}	O_{14}
M_2	O_{21}	O_{22}	O_{23}	
M_3	O_{31}		O_{33}	O_{34}

ジョブと工程：スケジュールの例

ショップ・スケジューリングの設定

- 1 つのジョブは 1 つ以上の **工程** (operation) から成る
- 各工程を処理できる機械が決まっている
- 同じジョブを構成する複数の工程は，同時刻に処理できない

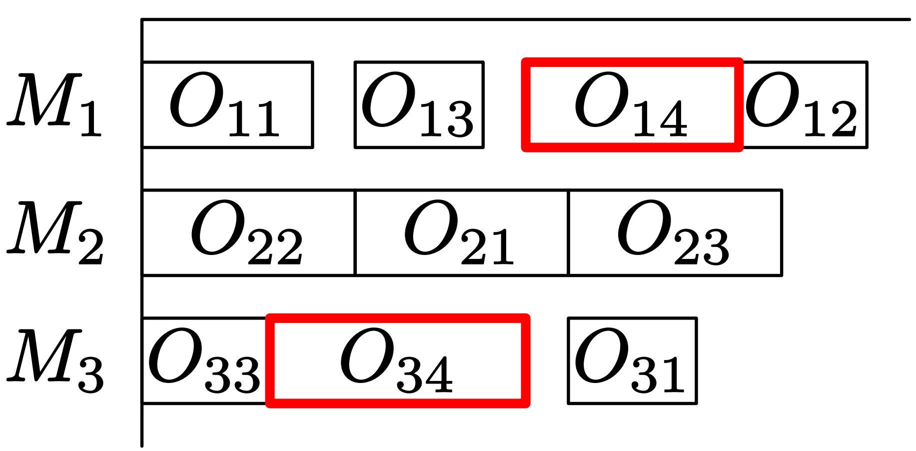


	J_1	J_2	J_3	J_4
M_1	O_{11}	O_{12}	O_{13}	O_{14}
M_2	O_{21}	O_{22}	O_{23}	
M_3	O_{31}		O_{33}	O_{34}

ジョブと工程：スケジュールの例

ショップ・スケジューリングの設定

- 1 つのジョブは 1 つ以上の **工程** (operation) から成る
- 各工程を処理できる機械が決まっている
- 同じジョブを構成する複数の工程は，同時刻に処理できない



$$
\begin{array}{c|cccc|}
 & J_1 & J_2 & J_3 & J_4 \\
\hline
M_1 & O_{11} & O_{12} & O_{13} & O_{14} \\
M_2 & O_{21} & O_{22} & O_{23} & \\
M_3 & O_{31} & & O_{33} & O_{34} \\
\hline
\end{array}
$$

ショップ ってなに？

Oxford Advanced Learner's Dictionary にて

shop *noun*

2. (also workshop) [countable] (especially in compounds)

a place where things are made or repaired, especially part of a factory where a particular type of work is done

＝ 工場において，特定の作業を行う場所

オープンショップ・スケジューリング

次の 3 つの制限・変種をよく考える

オープンショップ・スケジューリング では

- 任意のジョブ J_j と機械 M_i に対して，工程 O_{ij} がある

$$
\begin{array}{c|cccc|}
 & J_1 & J_2 & J_3 & J_4 \\
\hline
M_1 & O_{11} & O_{12} & O_{13} & O_{14} \\
M_2 & O_{21} & O_{22} & O_{23} & O_{24} \\
M_3 & O_{31} & O_{32} & O_{33} & O_{34} \\
\hline
\end{array}
$$

フローショップ・スケジューリング

次の 3 つの制限・変種をよく考える

フローショップ・スケジューリング では

- 任意のジョブ J_j と機械 M_i に対して, 工程 O_{ij} がある
- $O_{ij} \to O_{i+1,j}$ という先行制約がある

$$
\begin{array}{c|cccc|}
 & J_1 & J_2 & J_3 & J_4 \\
\hline
M_1 & O_{11} & O_{12} & O_{13} & O_{14} \\
 & \downarrow & \downarrow & \downarrow & \downarrow \\
M_2 & O_{21} & O_{22} & O_{23} & O_{24} \\
 & \downarrow & \downarrow & \downarrow & \downarrow \\
M_3 & O_{31} & O_{32} & O_{33} & O_{34} \\
\hline
\end{array}
$$

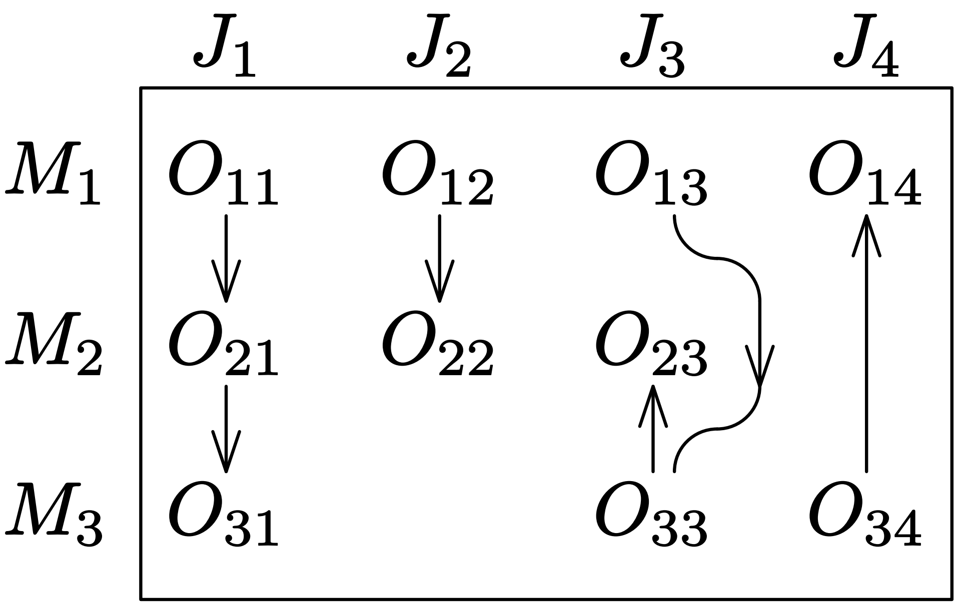
直感：流れ作業

ジョブショップ・スケジューリング

次の 3 つの制限・変種をよく考える

ジョブショップ・スケジューリング では

- 各ジョブに対して，その工程の間に鎖で表される先行制約がある



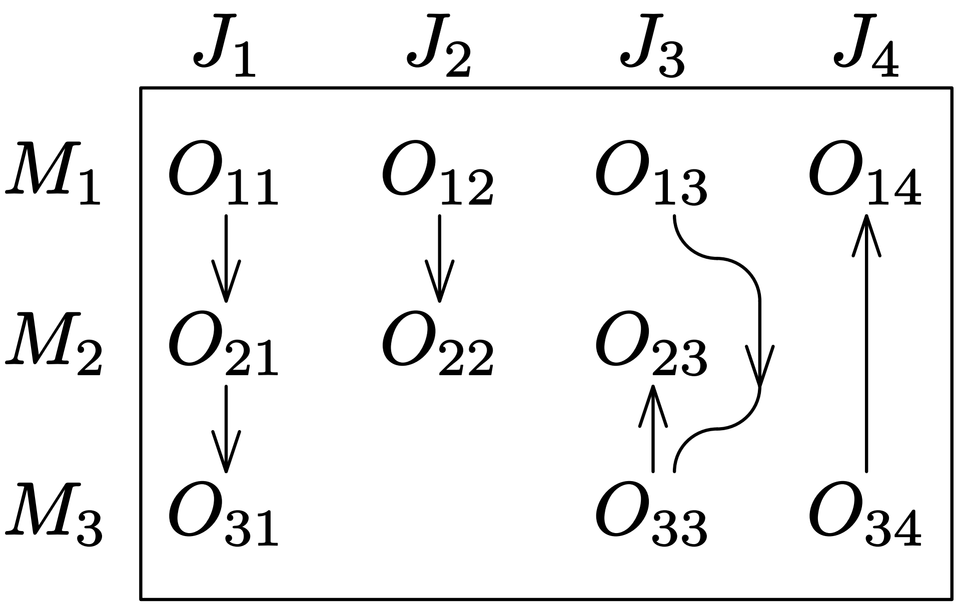
注：フローショップはジョブショップの特別な場合

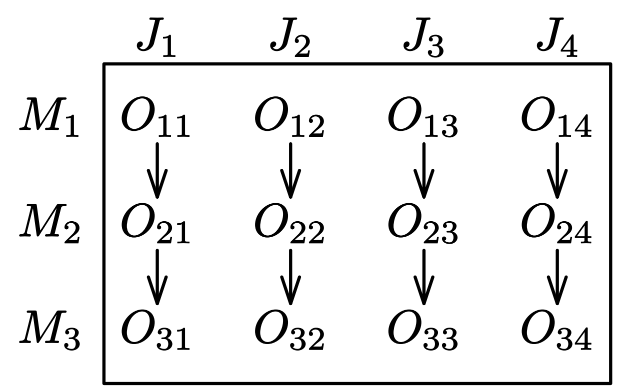
ジョブショップ・スケジューリング

次の 3 つの制限・変種をよく考える

ジョブショップ・スケジューリング では

- 各ジョブに対して，その工程の間に
  鎖で表される先行制約がある





注：フローショップはジョブショップの特別な場合

3 つ組記法

3 つ組記法 $\alpha \mid \beta \mid \gamma$ の α に次を書く ($\alpha =$ 機械の環境)

- O オープンショップ 機械数は入力の一部
- F フローショップ 機械数は入力の一部
- J ジョブショップ 機械数は入力の一部

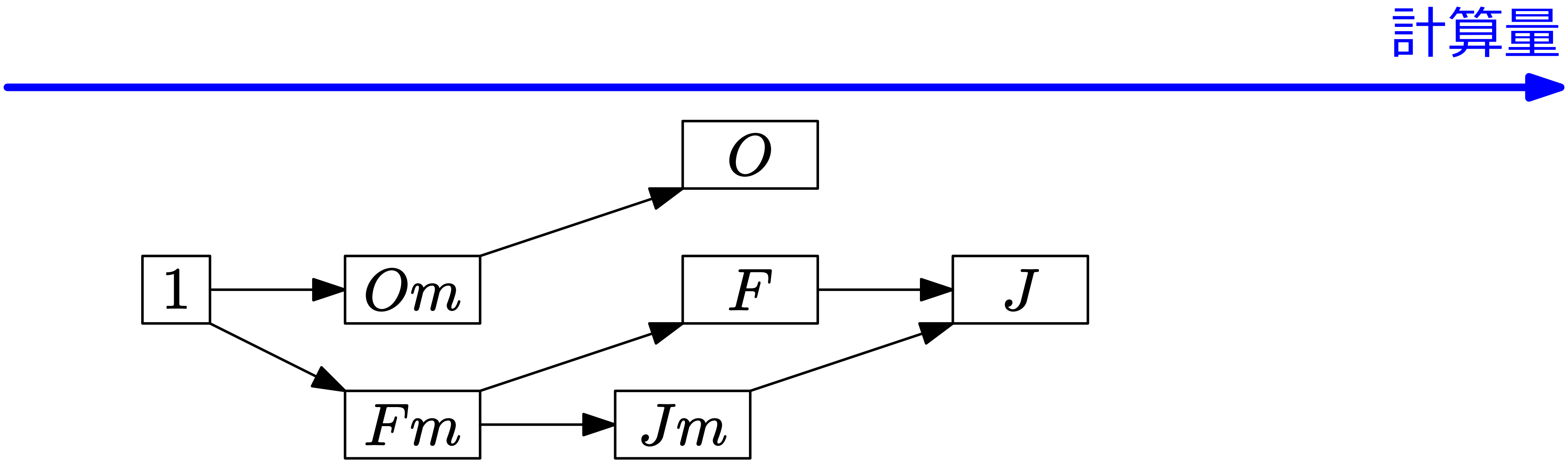
- Om オープンショップ 機械数は m で固定
- Fm フローショップ 機械数は m で固定
- Jm ジョブショップ 機械数は m で固定

3 つ組記法

3 つ組記法 $\alpha \mid \beta \mid \gamma$ の α に次を書く ($\alpha =$ 機械の環境)

- O オープンショップ 機械数は入力の一部
- F フローショップ 機械数は入力の一部
- J ジョブショップ 機械数は入力の一部

- Om オープンショップ 機械数は m で固定
- Fm フローショップ 機械数は m で固定
- Jm ジョブショップ 機械数は m で固定



本日の内容

1. ショップ・スケジューリングの用語
2. **$F2 \mid\mid C_{\max}$ のアルゴリズム**
3. $F3 \mid\mid C_{\max}$ の強 NP 困難性
4. $J2 \mid\mid C_{\max}$ のアルゴリズム

- S. M. Johnson, Optimal two and three-stage production schedules with steup times included. *Naval Research Logistics Quarterly* 1 (1954) pp. 61–67.

$F2 \mid\mid C_{\max}$

機械の環境

- フローショップ，機械数 $= 2$
- $p_{1j} =$ 機械 1 における工程 O_{1j} の処理時間
- $p_{2j} =$ 機械 2 における工程 O_{2j} の処理時間

最適化の目的

- 最大完了時刻の最小化

$$
\begin{array}{c|ccccccc|}
 & J_1 & J_2 & J_3 & J_4 & J_5 & J_6 & J_7 \\
\hline
M_1 & O_{11} & O_{12} & O_{13} & O_{14} & O_{15} & O_{16} & O_{17} \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
M_2 & O_{21} & O_{22} & O_{23} & O_{24} & O_{25} & O_{26} & O_{27} \\
\hline
\end{array}
$$

$F2 \mid\mid C_{\max}$

機械の環境

- フローショップ，機械数 $= 2$
- $p_{1j} =$ 機械 1 における工程 O_{1j} の処理時間
- $p_{2j} =$ 機械 2 における工程 O_{2j} の処理時間

最適化の目的

- 最大完了時刻の最小化

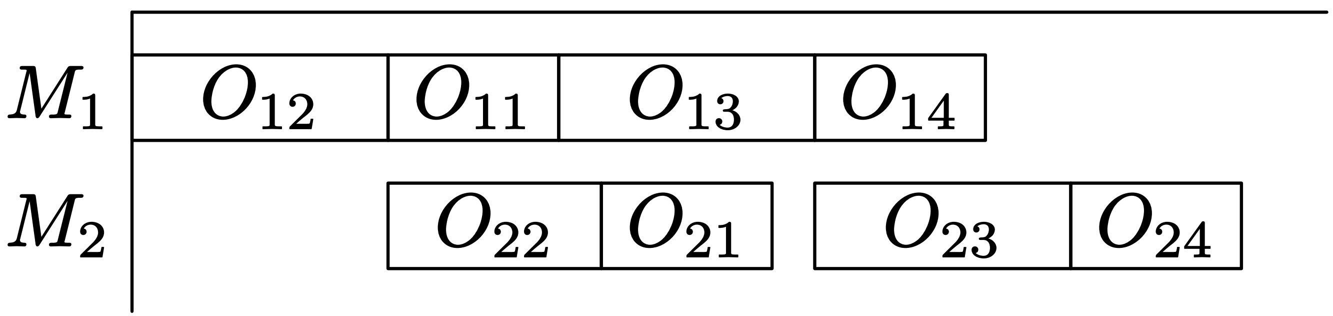
	J_1	J_2	J_3	J_4	J_5	J_6	J_7
M_1	4	2	5	1	3	2	0
M_2	1	5	3	2	3	5	6

同順に処理する最適解

性質

$F2 \,||\, C_{\max}$ において，次を満たす最適解が存在する

機械 1 で O_{1j} を $O_{1j'}$ より先に処理する $\Leftrightarrow$
機械 2 で O_{2j} を $O_{2j'}$ より先に処理する



	J_1	J_2	J_3	J_4
M_1	4	6	6	4
M_2	4	5	6	4

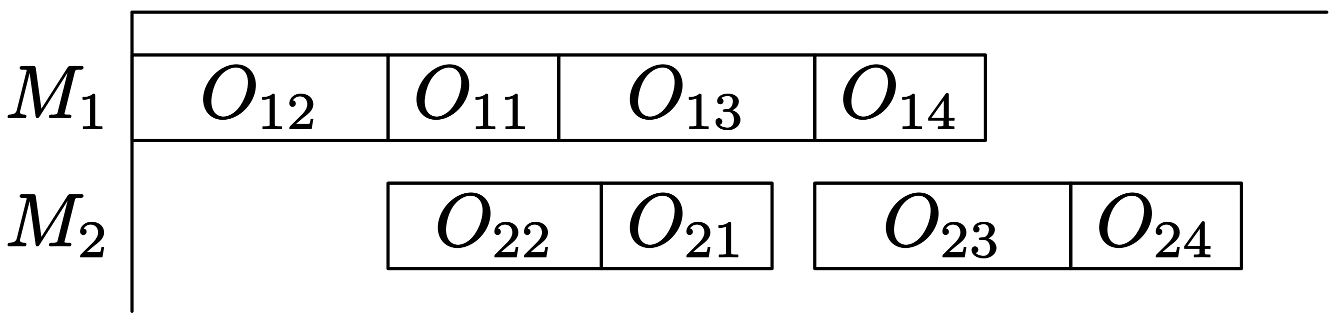
同順に処理する最適解

性質

$F2 \mid\mid C_{\max}$ において，次を満たす最適解が存在する

機械 1 で O_{1j} を $O_{1j'}$ より先に処理する $\Leftrightarrow$

機械 2 で O_{2j} を $O_{2j'}$ より先に処理する

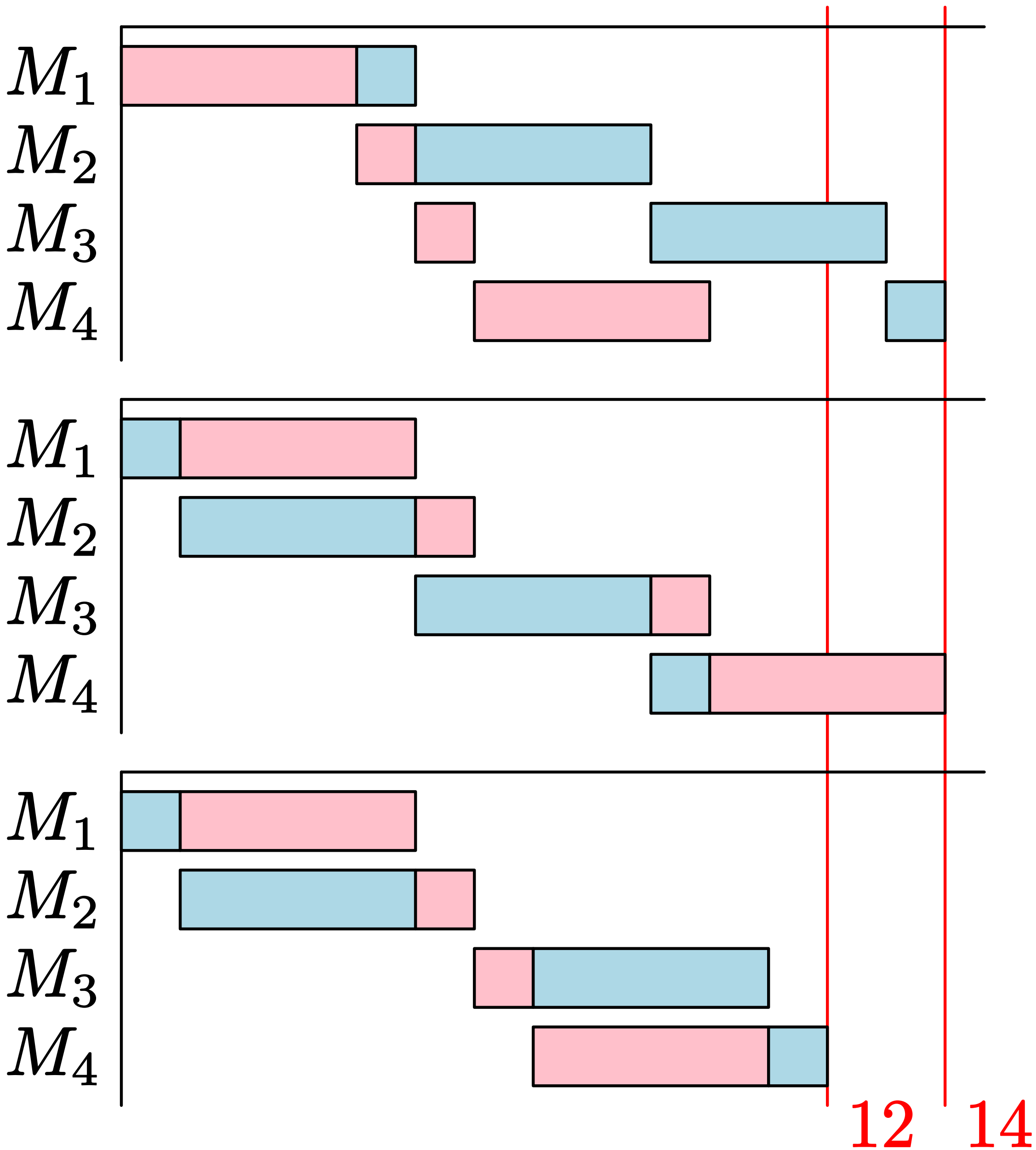


ジョブの処理順：J_2, J_1, J_3, J_4

	J_1	J_2	J_3	J_4
M_1	4	6	6	4
M_2	4	5	6	4

同順に処理しない最適解

注意：先の性質は $F4 \mid\mid C_{\max}$ では成り立たない

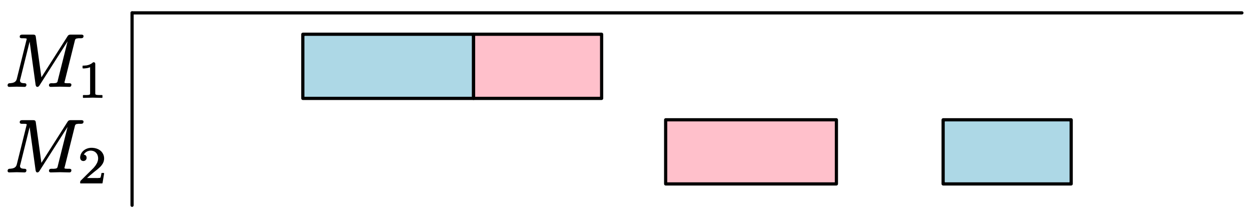


	J_1	J_2
M_1	4	1
M_2	1	4
M_3	1	4
M_4	4	1

同順に処理する最適解：証明

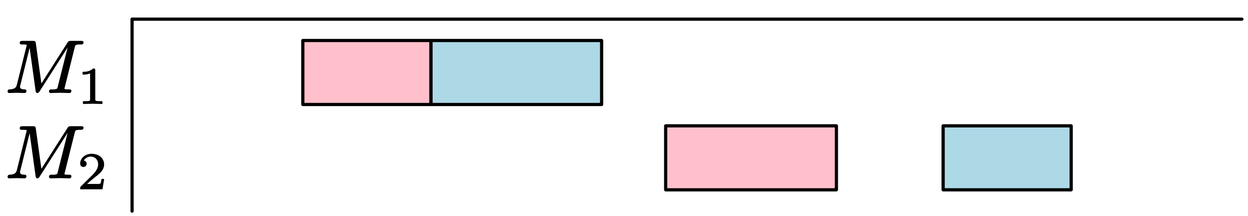
証明：最適解を 1 つ持ってきて，σ^* とする

σ^* で，M_1 が処理するジョブの連続ペアを順にみて，
M_2 の順と異なるところを特定する



最適解 σ^*

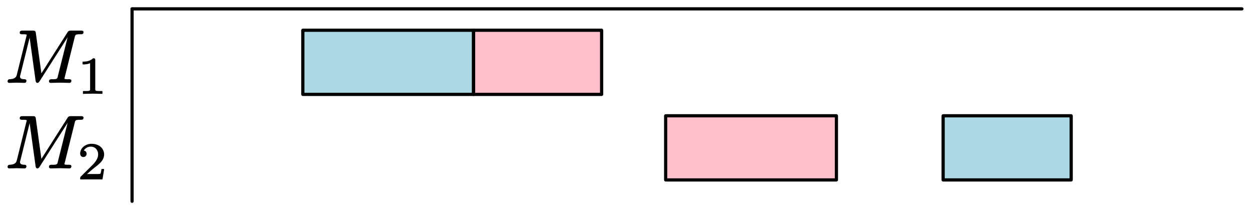
そのペアの M_1 における処理順を交換する



同順に処理する最適解：証明

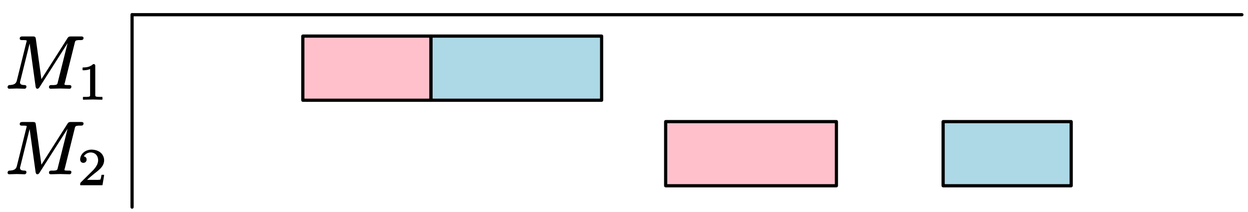
証明：最適解を１つ持ってきて，σ^* とする

σ^* で，M_1 が処理するジョブの連続ペアを順にみて，M_2 の順と異なるところを特定する



最適解 σ^*

そのペアの M_1 における処理順を交換する



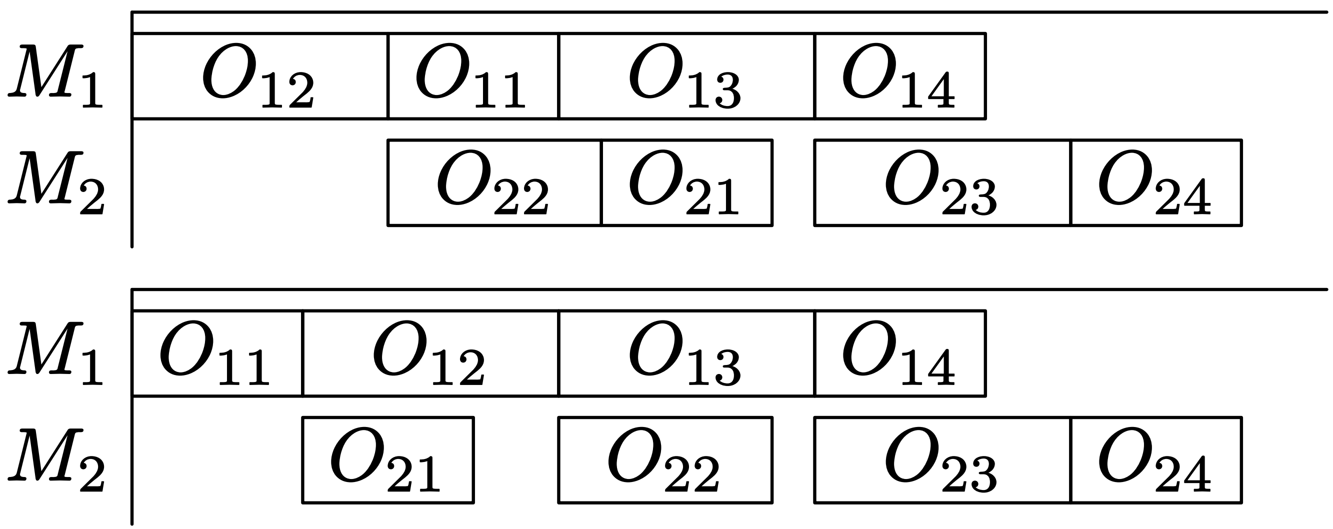
これも最適解

このとき，最大完了時刻は変わらない □

$F2 \mid\mid C_{\max}$：アイデア

フローショップでは，最後の機械の完了時刻が $C_{\max}$ になる

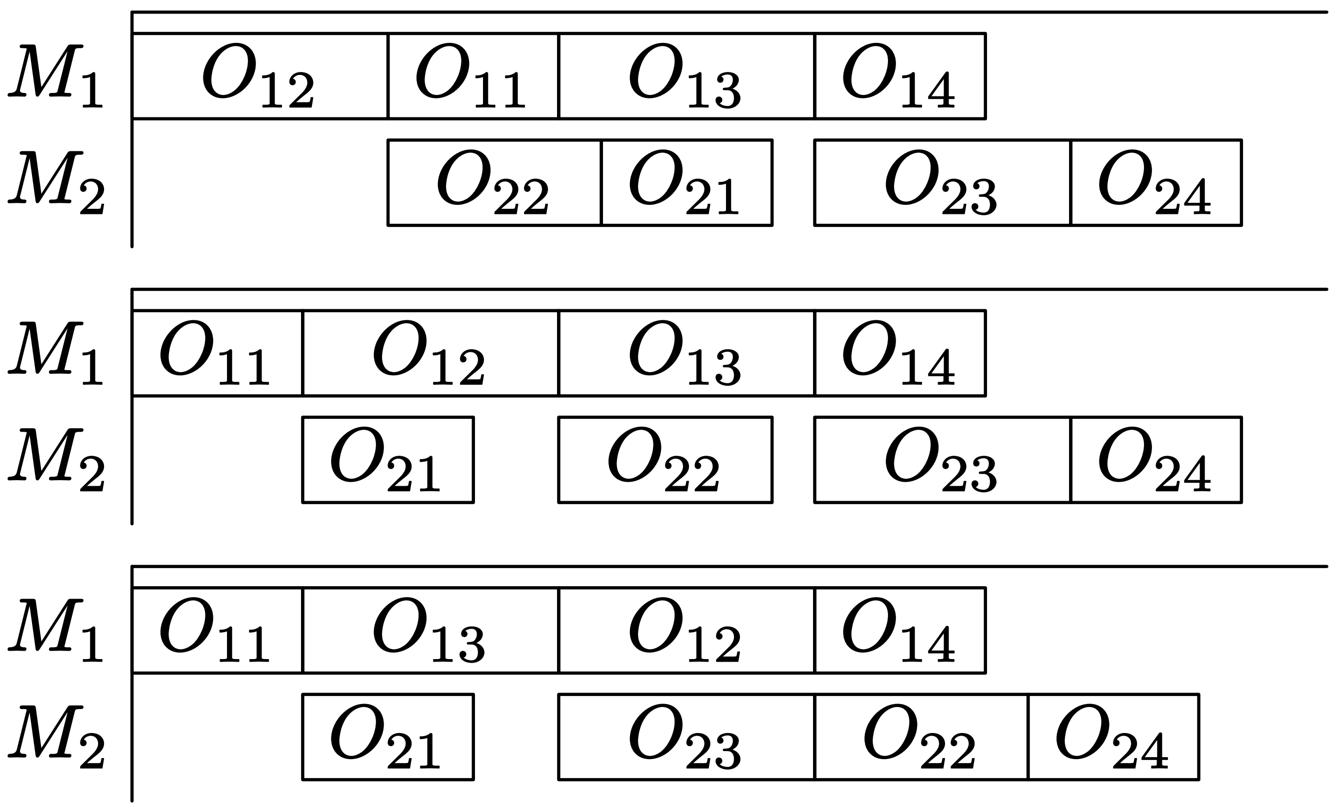
- 機械 1 で短い工程を先に処理したほうが，
  機械 2 の処理を早く開始できて，よい
- 機械 2 で長い工程を先に処理したほうが，
  機械 2 の遊休時間 (idle time) を減らせて，よい



$F2 \,||\, C_{\max}$：アイデア

フローショップでは，最後の機械の完了時刻が $C_{\max}$ になる

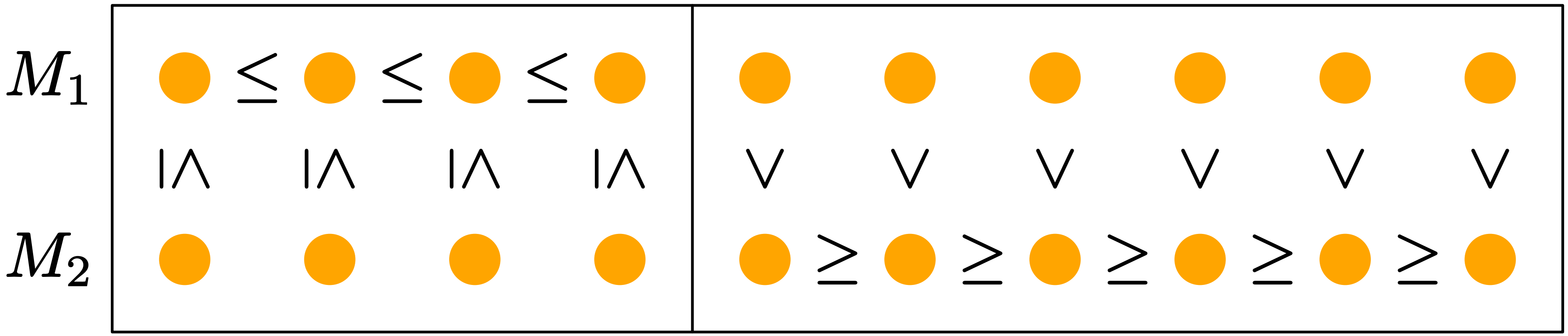
- 機械 1 で短い工程を先に処理したほうが，
  機械 2 の処理を早く開始できて，よい
- 機械 2 で長い工程を先に処理したほうが，
  機械 2 の遊休時間 (idle time) を減らせて，よい



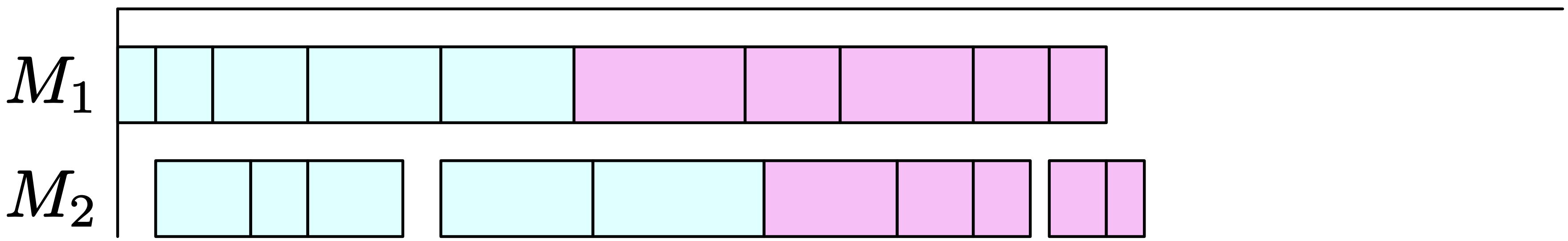
Johnson のアルゴリズム (Johnson ’54)

次のようにジョブの処理順を決める

1. $p_{1j} \leq p_{2j}$ を満たすジョブ J_j を先に処理する
   それらの中では，p_{1j} が小さい方から処理する
2. $p_{1j} > p_{2j}$ を満たすジョブ J_j をその後で処理する
   それらの中では，p_{2j} が大きい方から処理する



Johnson のアルゴリズム：例



$$
\begin{array}{c|ccccccccc|ccccccccc|}
M_1 & 2 & \leq & 3 & \leq & 5 & \leq & 7 & \leq & 7 & 9 & & 5 & & 5 & & 4 & & 3 \\
 & \rotatebox{90}{$\leq$} & & \rotatebox{90}{$\leq$} & & \rotatebox{90}{$\leq$} & & \rotatebox{90}{$\leq$} & & \rotatebox{90}{$\leq$} & \vee & & \vee & & \vee & & \vee & & \vee \\
M_2 & 5 & & 3 & & 5 & & 8 & & 9 & 7 & \geq & 4 & \geq & 3 & \geq & 3 & \geq & 2
\end{array}
$$

定理

Johnson のアルゴリズムは問題 $F2 \mid\mid C_{\max}$ を $O(n \log n)$ 時間で解く

解く ＝ 必ず最適解を出力する

Johnson のアルゴリズム (Johnson ’54)

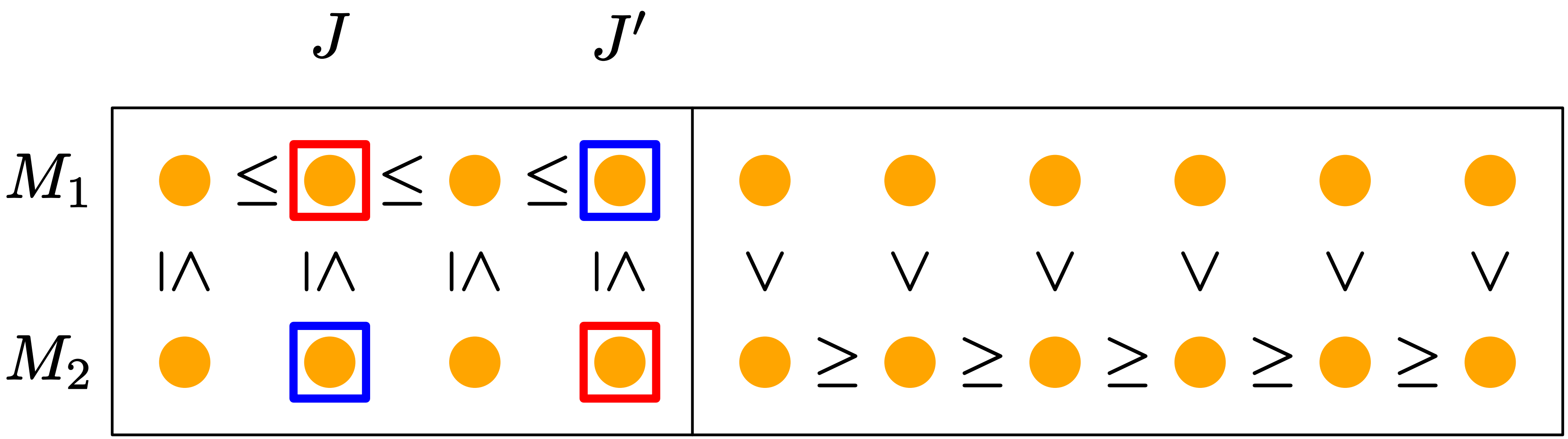
次のようにジョブの処理順を決める

1. $p_{1j} \leq p_{2j}$ を満たすジョブ J_j を先に処理する
   それらの中では，p_{1j} が小さい方から処理する
2. $p_{1j} > p_{2j}$ を満たすジョブ J_j をその後で処理する
   それらの中では，p_{2j} が大きい方から処理する

Johnson のアルゴリズム：証明 (1)

$\boxed{\text{並べ方の性質}}$：J が J' の前にある ⇒

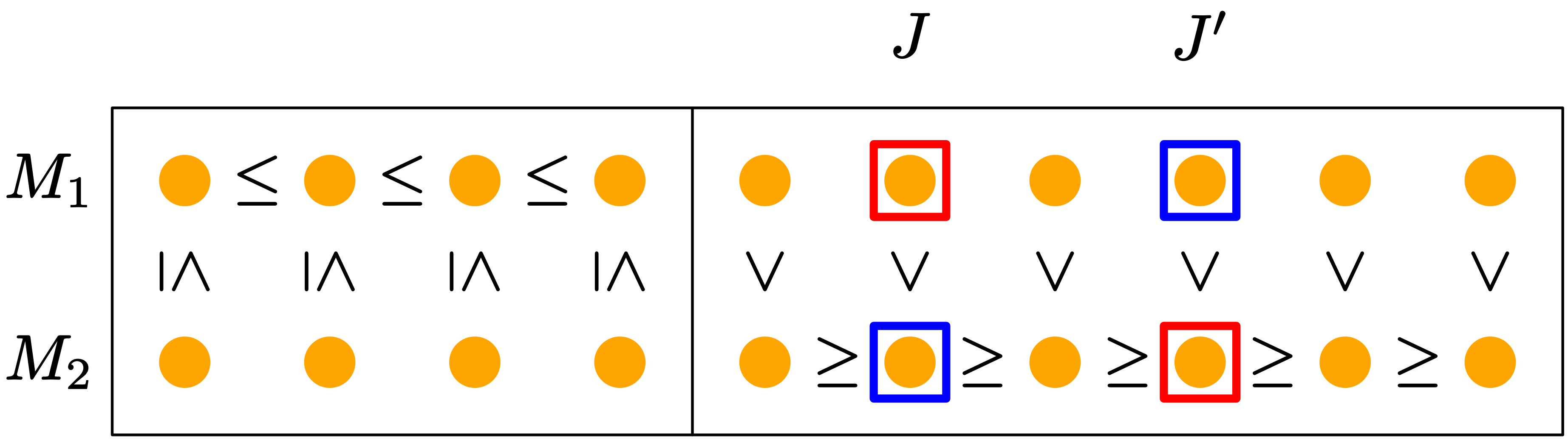
$$\min\{p_1, p'_2\} \leq \min\{p'_1, p_2\}$$



Johnson のアルゴリズム：証明 (1)

並べ方の性質：J が J' の前にある ⇒

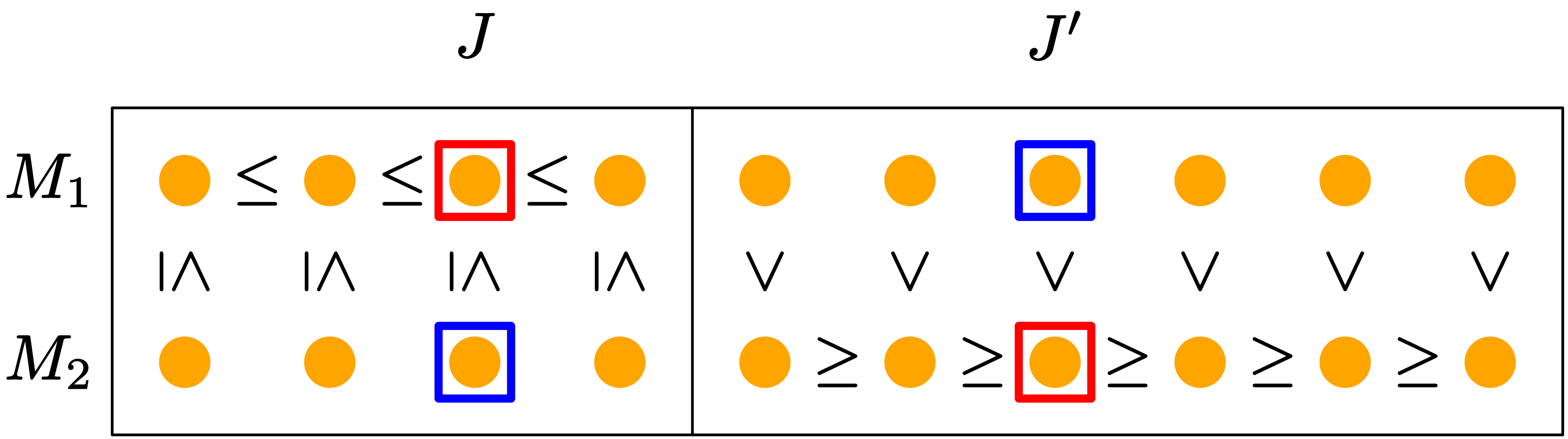
$$\min\{p_1, p_2'\} \le \min\{p_1', p_2\}$$



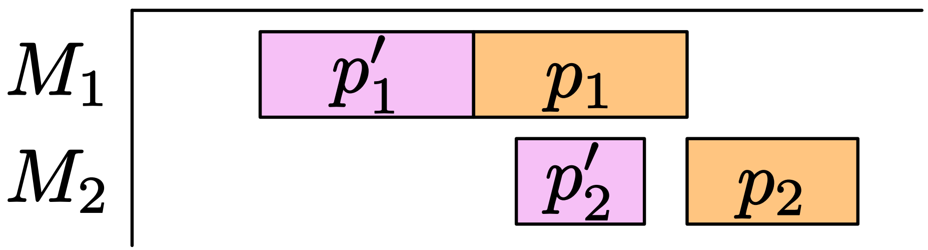
Johnson のアルゴリズム：証明 (1)

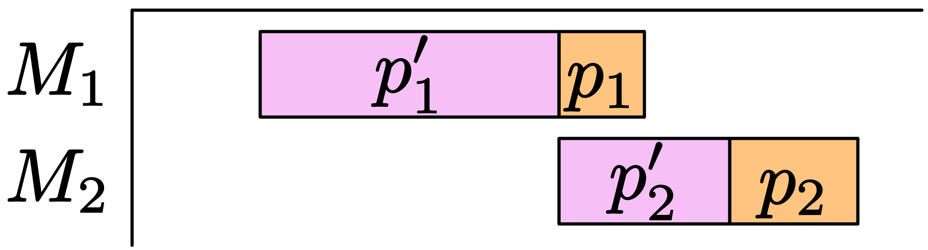
$\boxed{\text{並べ方の性質}}$：J が J' の前にある ⇒

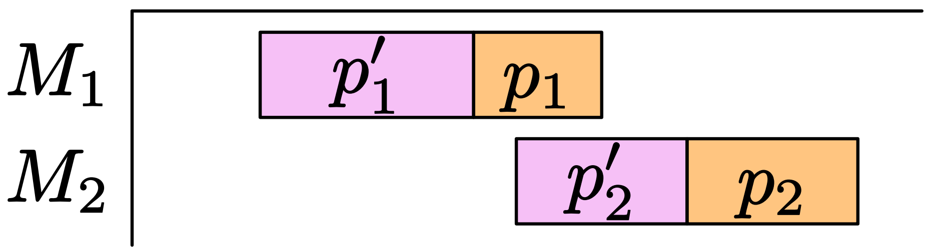
$$\min\{p_1, p'_2\} \leq \min\{p'_1, p_2\}$$

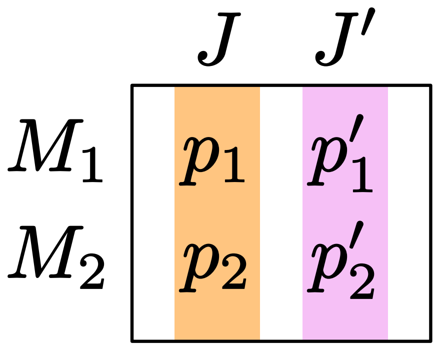


Johnson のアルゴリズム：証明 (2)



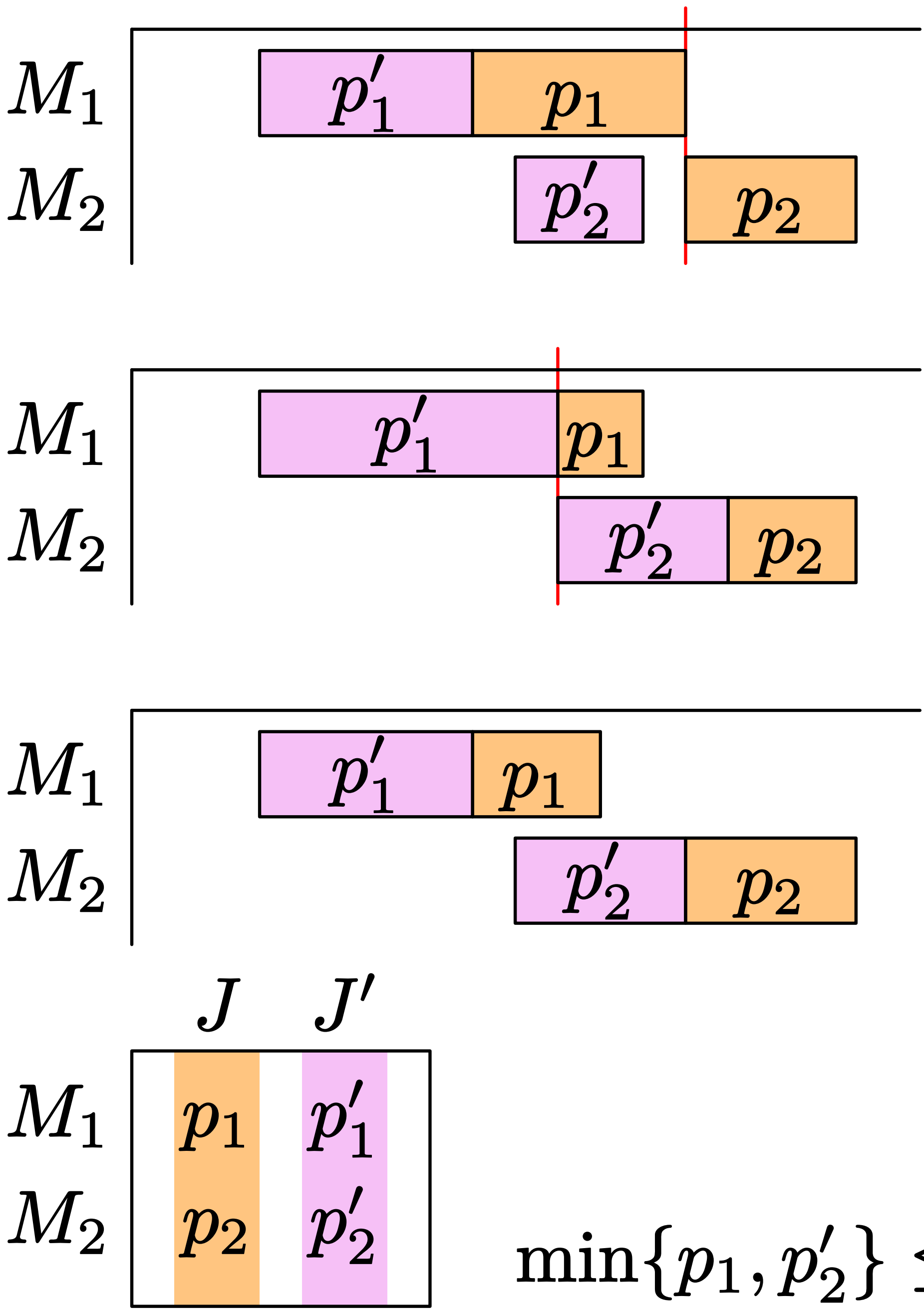






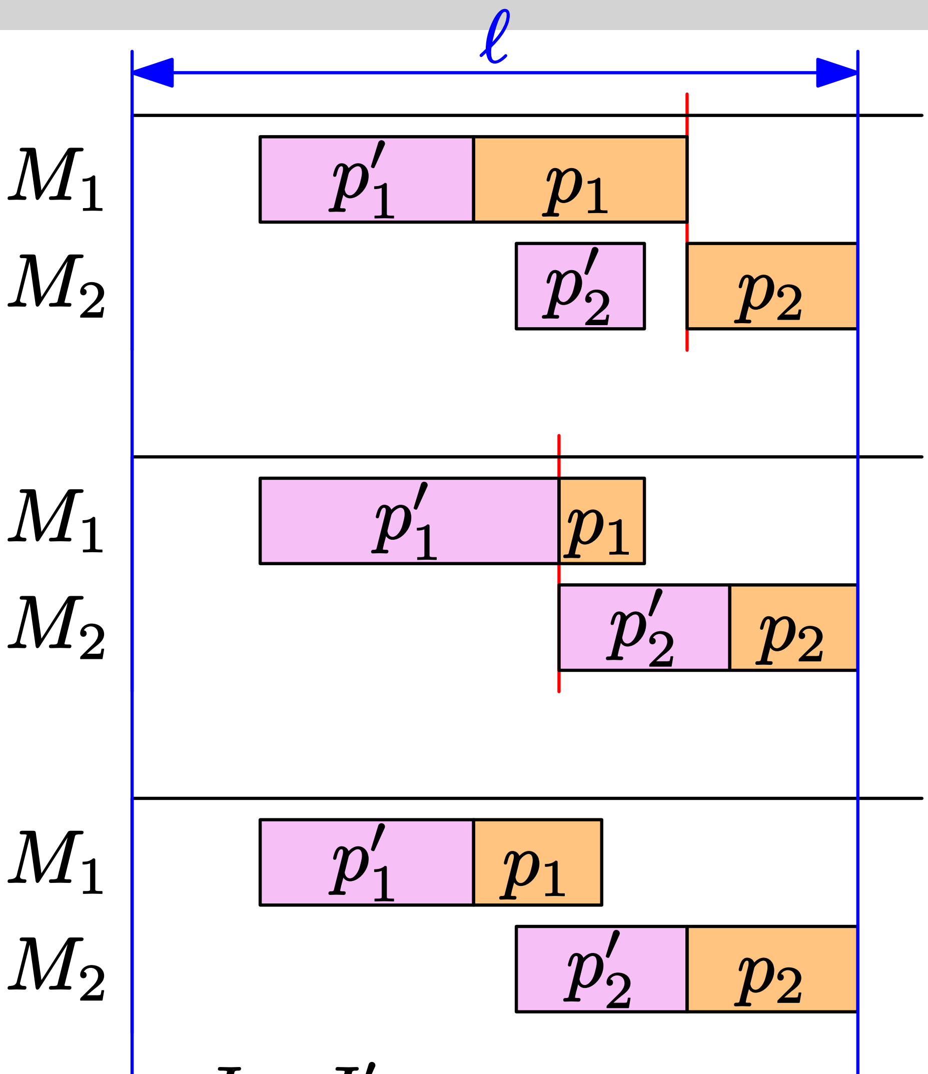
$$\min\{p_1, p'_2\} \leq \min\{p'_1, p_2\}$$

Johnson のアルゴリズム：証明 (2)



$$\min\{p_1, p'_2\} \leq \min\{p'_1, p_2\}$$

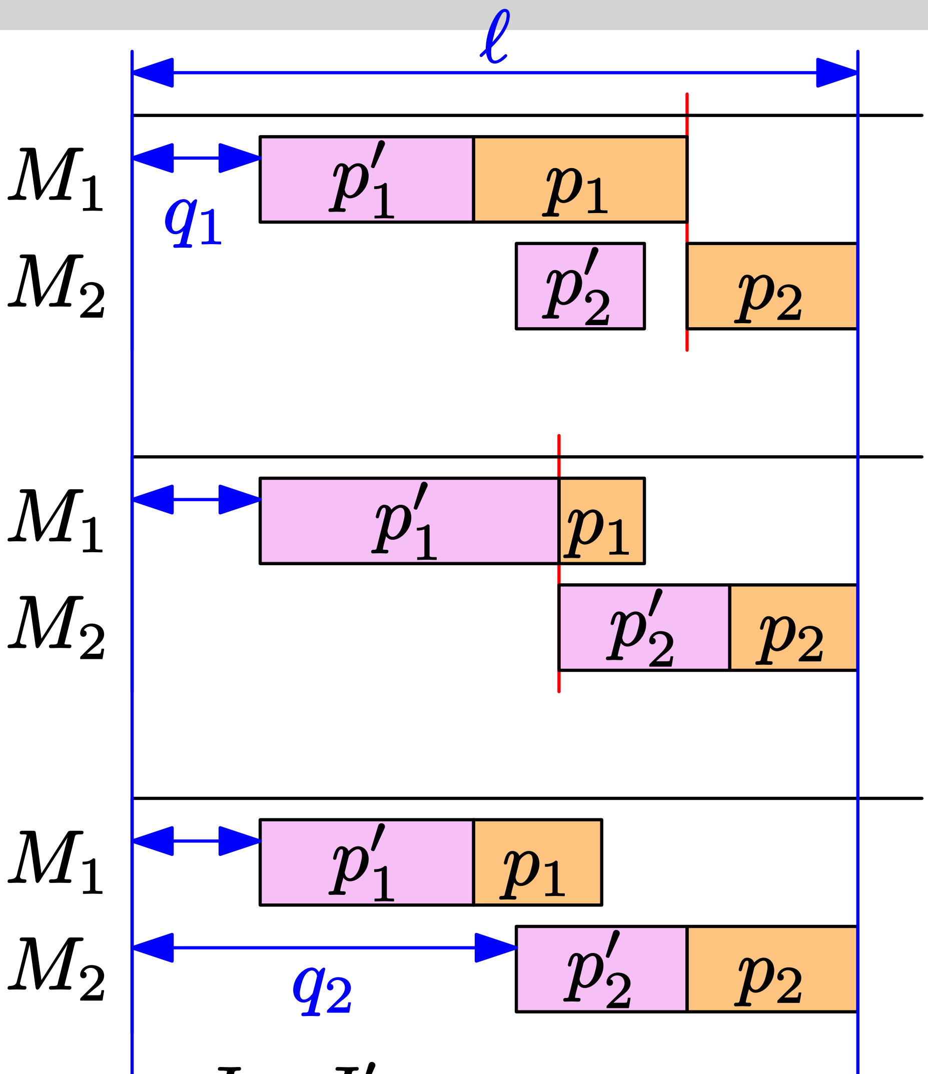
Johnson のアルゴリズム：証明 (2)



	J	J'
M_1	p_1	p_1'
M_2	p_2	p_2'

$$\min\{p_1, p_2'\} \le \min\{p_1', p_2\}$$

Johnson のアルゴリズム：証明 (2)



$q_1 + p'_1 + p_1 + p_2$

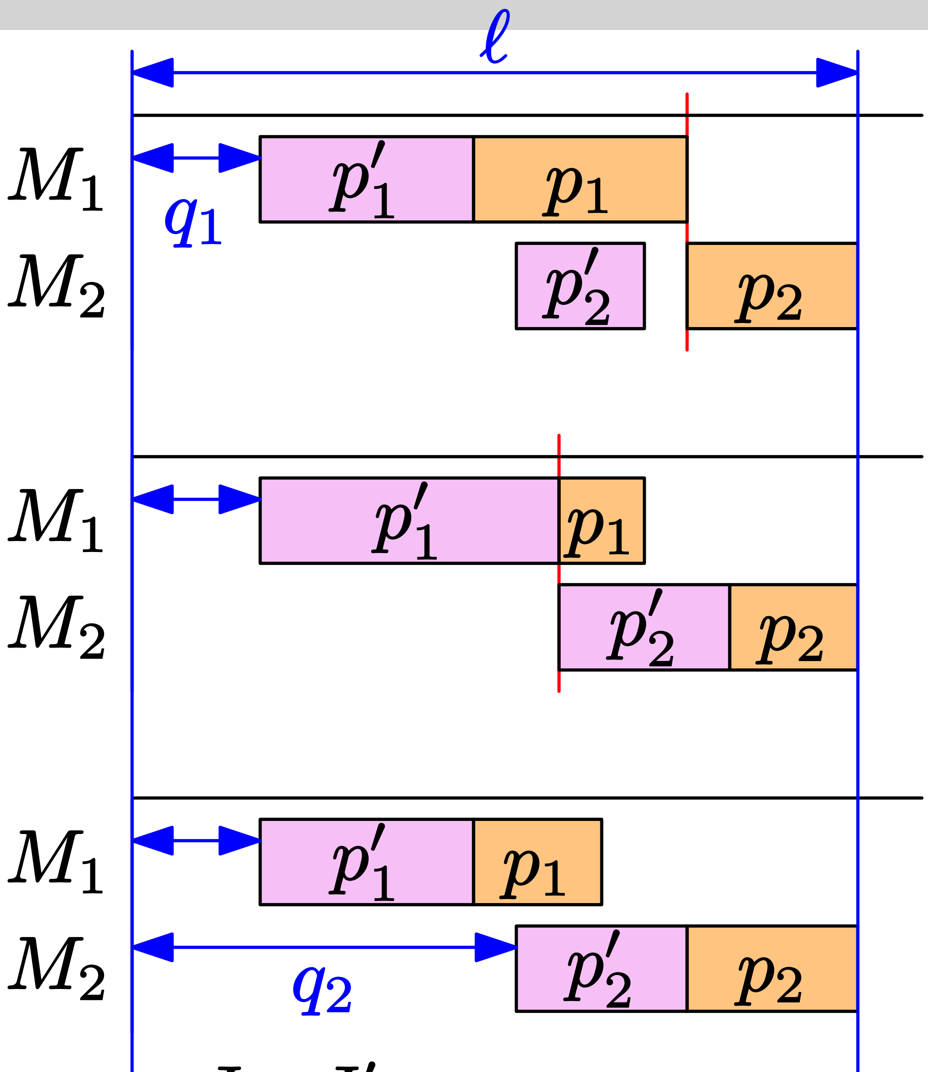
$q_1 + p'_1 + p'_2 + p_2$

$q_2 + p'_2 + p_2$

	J	J'
M_1	p_1	p'_1
M_2	p_2	p'_2

$$\min\{p_1, p'_2\} \le \min\{p'_1, p_2\}$$

Johnson のアルゴリズム：証明 (2)



$$q_1 + p'_1 + p_1 + p_2$$

$$q_1 + p'_1 + p'_2 + p_2$$

$$q_2 + p'_2 + p_2$$

	J	J'
M_1	p_1	p'_1
M_2	p_2	p'_2

$$\ell = \max \left\{ \begin{array}{l} q_1 + p'_1 + p_2 + \max\{p_1, p'_2\}, \\ q_2 + p'_2 + p_2 \end{array} \right\}$$

$$\min\{p_1, p'_2\} \le \min\{p'_1, p_2\}$$

Johnson のアルゴリズム：証明 (3)

J', J の順に処理：

$$\ell = \max\left\{\begin{array}{l} q_1 + p'_1 + p_2 + \max\{p_1, p'_2\}, \\ q_2 + p'_2 + p_2 \end{array}\right\}$$

J, J' の順に処理：

$$\ell = \max\left\{\begin{array}{l} q_1 + p_1 + p'_2 + \max\{p'_1, p_2\}, \\ q_2 + p_2 + p'_2 \end{array}\right\}$$

	J	J'
M_1	p_1	p'_1
M_2	p_2	p'_2

$$\min\{p_1, p'_2\} \le \min\{p'_1, p_2\}$$

Johnson のアルゴリズム：証明 (3)

J', J の順に処理：

$$\ell = \max \left\{ \begin{array}{l} q_1 + p'_1 + p_2 + \max\{p_1, p'_2\}, \\ q_2 + p'_2 + p_2 \end{array} \right\}$$

J, J' の順に処理：

$$\ell = \max \left\{ \begin{array}{l} q_1 + p_1 + p'_2 + \max\{p'_1, p_2\}, \\ q_2 + p_2 + p'_2 \end{array} \right\}$$

$$\begin{aligned} p'_1 + p_2 + \max\{p_1, p'_2\} &= p'_1 + p_2 + p_1 + p'_2 - \min\{p_1, p'_2\} \\ &\geq p'_1 + p_2 + p_1 + p'_2 - \min\{p'_1, p_2\} \\ &= p_1 + p'_2 + \max\{p'_1, p_2\} \end{aligned}$$

	J	J'
M_1	p_1	p'_1
M_2	p_2	p'_2

$$\min\{p_1, p'_2\} \leq \min\{p'_1, p_2\}$$

Johnson のアルゴリズム：証明 (3)

J', J の順に処理：

$$\ell = \max\left\{\begin{array}{l} q_1 + p'_1 + p_2 + \max\{p_1, p'_2\}, \\ q_2 + p'_2 + p_2 \end{array}\right\}$$

$\geq$

J, J' の順に処理：

$$\ell = \max\left\{\begin{array}{l} q_1 + p_1 + p'_2 + \max\{p'_1, p_2\}, \\ q_2 + p_2 + p'_2 \end{array}\right\}$$

$\square$

$$\begin{aligned} p'_1 + p_2 + \max\{p_1, p'_2\} &= p'_1 + p_2 + p_1 + p'_2 - \min\{p_1, p'_2\} \\ &\geq p'_1 + p_2 + p_1 + p'_2 - \min\{p'_1, p_2\} \\ &= p_1 + p'_2 + \max\{p'_1, p_2\} \end{aligned}$$

	J	J'
M_1	p_1	p'_1
M_2	p_2	p'_2

$$\min\{p_1, p'_2\} \leq \min\{p'_1, p_2\}$$

まとめ：$F2 \mid\mid C_{\max}$

定理

Johnson のアルゴリズムは問題 $F2 \mid\mid C_{\max}$ を $O(n \log n)$ 時間で解く

解く ＝ 必ず最適解を出力する

$O(n \log n)$ 時間 ＝ ジョブのソーティングにかかる時間

Johnson のアルゴリズム (Johnson '54)

次のようにジョブの処理順を決める

1. $p_{1j} \leq p_{2j}$ を満たすジョブ J_j を先に処理する
   それらの中では，p_{1j} が小さい方から処理する
2. $p_{1j} > p_{2j}$ を満たすジョブ J_j をその後で処理する
   それらの中では，p_{2j} が大きい方から処理する

本日の内容

1. ショップ・スケジューリングの用語
2. $F2 \mid\mid C_{\max}$ のアルゴリズム
3. $F3 \mid\mid C_{\max}$ **の強 NP 困難性**
4. $J2 \mid\mid C_{\max}$ のアルゴリズム

- M. R. Garey, D. S. Johnson, R. Sethi, The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research* 1 (1976) pp. 117–129.

$F3 \mid\mid C_{\max}$ の強 NP 困難性

定理 (Garey, Johnson, Sethi '76)

問題 $F3 \mid\mid C_{\max}$ は強 NP 困難

注：$F2 \mid\mid C_{\max}$ は $O(n \log n)$ 時間で解ける

3 分割問題

定義：3 分割問題 (3-partition problem)

次を満たす正整数 $a_1, a_2, \ldots, a_{3n}$ が与えられる

$$\frac{1}{4}T < a_i < \frac{1}{2}T \quad \text{ただし，} \; T = \frac{1}{n}\sum_{i=1}^{3n} a_i$$

それらを同じ和の n 個のグループに分けられるか？

例：$119, 120, 122, 123, 125, 127, 130, 130, 130, 140, 145, 149$

- $120 + 125 + 145 \; = 390$
- $123 + 127 + 140 \; = 390$
- $119 + 122 + 149 \; = 390$
- $130 + 130 + 130 \; = 390$

できる

事実 (Garey, Johnson ’75)

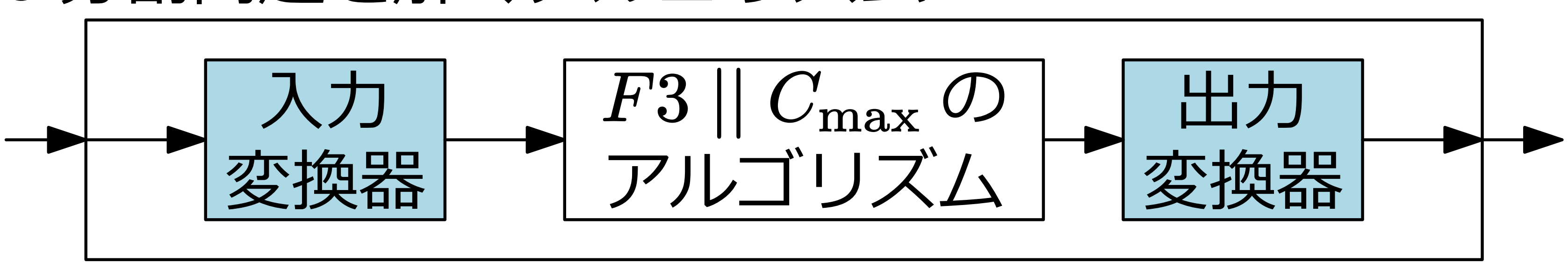
3 分割問題 は強 NP 困難

<u>証明</u>：3 分割問題を帰着する

入力の変換：$119, 120, 122, 123, 125, 127, 130, 130, 130, 140, 145, 149$

M_1	0	0	0	0	0	0	0	0	0	0	0	0	0	780	780	780	780
M_2	119	120	122	123	125	127	130	130	130	140	145	149	390	390	390	390	390
M_3	0	0	0	0	0	0	0	0	0	0	0	0	780	780	780	780	0

3 分割問題を解くアルゴリズム



$F3 \,||\, C_{\max}$：証明 (1)

証明：3 分割問題を帰着する

入力の変換：$119, 120, 122, 123, 125, 127, 130, 130, 130, 140, 145, 149$

↓

M_1	0	0	0	0	0	0	0	0	0	0	0	0	0	780	780	780	780
M_2	119	120	122	123	125	127	130	130	130	140	145	149	390	390	390	390	390
M_3	0	0	0	0	0	0	0	0	0	0	0	0	780	780	780	780	0

0

a_i

0

3 分割問題を解くアルゴリズム

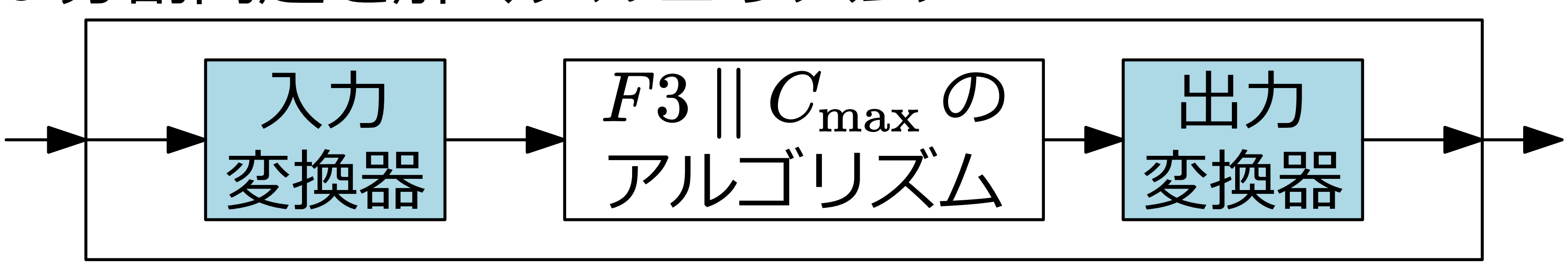
<u>証明</u>：3 分割問題を帰着する

入力の変換：$119, 120, 122, 123, 125, 127, 130, 130, 130, 140, 145, 149$

M_1	0	0	0	0	0	0	0	0	0	0	0	0	0	780	780	780	780
M_2	119	120	122	123	125	127	130	130	130	140	145	149	390	390	390	390	390
M_3	0	0	0	0	0	0	0	0	0	0	0	0	780	780	780	780	0

0
T
$2T$

3 分割問題を解くアルゴリズム



$F3 \,||\, C_{\max}$：証明 (1)

証明：3 分割問題を帰着する

入力の変換：$119, 120, 122, 123, 125, 127, 130, 130, 130, 140, 145, 149$

M_1	0	0	0	0	0	0	0	0	0	0	0	0	0	780	780	780	780
M_2	119	120	122	123	125	127	130	130	130	140	145	149	390	390	390	390	390
M_3	0	0	0	0	0	0	0	0	0	0	0	0	780	780	780	780	0

$2T$

T

0

3 分割問題を解くアルゴリズム

<u>証明</u>：3 分割問題を帰着する

入力の変換：$119, 120, 122, 123, 125, 127, 130, 130, 130, 140, 145, 149$

$n-1$ 個

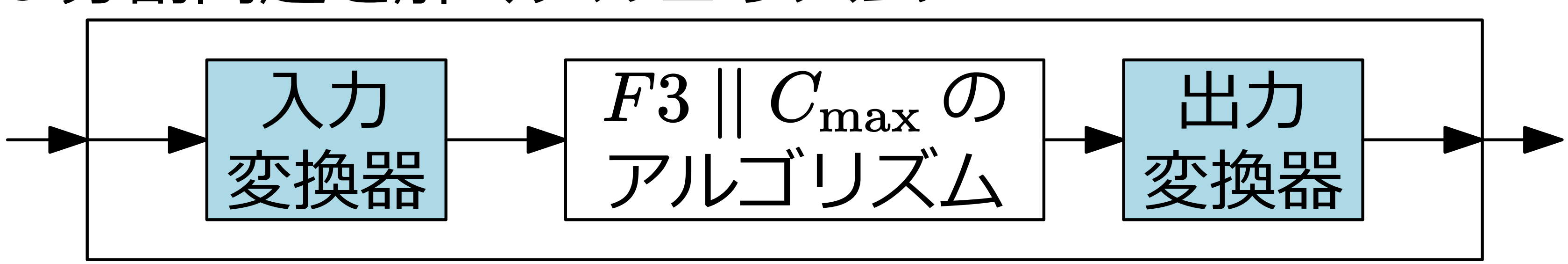
M_1	0	0	0	0	0	0	0	0	0	0	0	0	0	780	780	780	780
M_2	119	120	122	123	125	127	130	130	130	140	145	149	390	390	390	390	390
M_3	0	0	0	0	0	0	0	0	0	0	0	0	780	780	780	780	0

$2T$

T

$2T$

3 分割問題を解くアルゴリズム



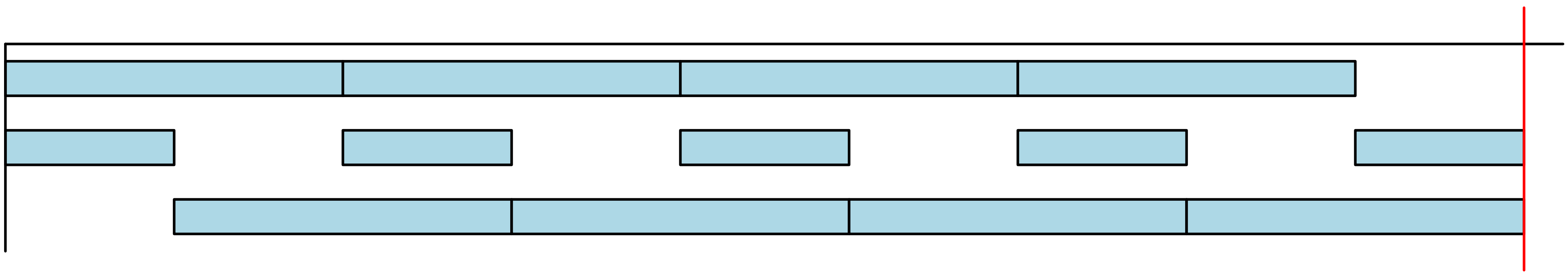
証明：3 分割問題を帰着する

出力の変換：

最適値 $\leq (2n+1)T \Rightarrow$ 3 分割問題の答えは「できる」

最適値 $> (2n+1)T \Rightarrow$ 3 分割問題の答えは「できない」

$\square$



M_1	0	0	0	0	0	0	0	0	0	0	0	0	0	780	780	780	780
M_2	119	120	122	123	125	127	130	130	130	140	145	149	390	390	390	390	390
M_3	0	0	0	0	0	0	0	0	0	0	0	0	780	780	780	780	0

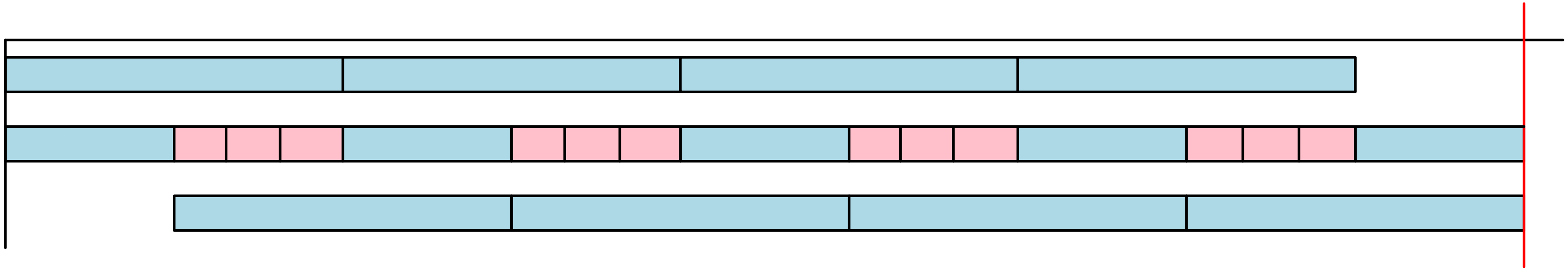
証明：3 分割問題を帰着する

出力の変換：

最適値 $\leq (2n+1)T \Rightarrow$ 3 分割問題の答えは「できる」

最適値 $> (2n+1)T \Rightarrow$ 3 分割問題の答えは「できない」

□



M_1	0	0	0	0	0	0	0	0	0	0	0	0	0	780	780	780	780
M_2	119	120	122	123	125	127	130	130	130	140	145	149	390	390	390	390	390
M_3	0	0	0	0	0	0	0	0	0	0	0	0	780	780	780	780	0

本日の内容

1. ショップ・スケジューリングの用語
2. $F2 \,||\, C_{\max}$ のアルゴリズム
3. $F3 \,||\, C_{\max}$ の強 NP 困難性
4. $J2 \,||\, C_{\max}$ **のアルゴリズム**

- S. M. Johnson, Optimal two and three-stage production schedules with steup times included. *Naval Research Logistics Quarterly* 1 (1954) pp. 61–67.

$J2 \,||\, C_{\max}$

機械の環境

- ジョブショップ，機械数 $= 2$
- $p_{1j} =$ 機械 1 における工程 O_{1j} の処理時間
- $p_{2j} =$ 機械 2 における工程 O_{2j} の処理時間

最適化の目的

- 最大完了時刻の最小化

注：この授業のジョブショップでは，各ジョブに対して 1 つの機械で処理する工程が高々 1 つ

$$
\begin{array}{c|ccccccc|}
 & J_1 & J_2 & J_3 & J_4 & J_5 & J_6 & J_7 \\
\hline
M_1 & O_{11} & O_{12} & & O_{14} & O_{15} & O_{16} & O_{17} \\
 & \downarrow & \downarrow & & \uparrow & \uparrow & \uparrow & \\
M_2 & O_{21} & O_{22} & O_{23} & O_{24} & O_{25} & O_{26} & \\
\hline
\end{array}
$$

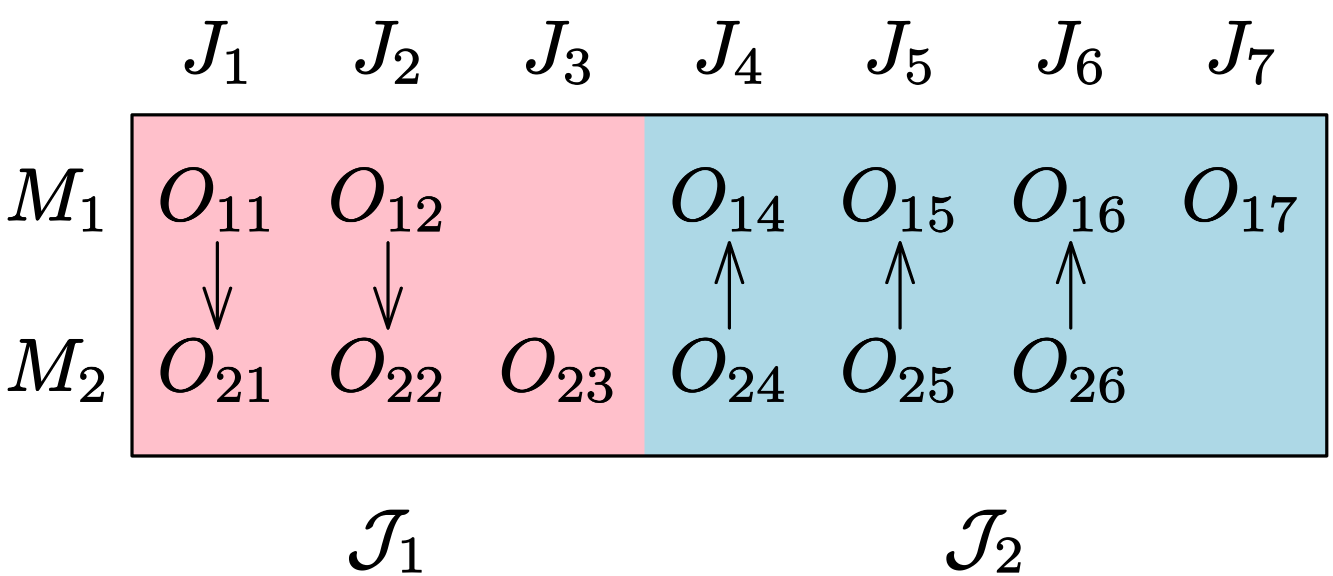
機械の環境

- ジョブショップ，機械数 $= 2$
- $p_{1j} =$ 機械 1 における工程 O_{1j} の処理時間
- $p_{2j} =$ 機械 2 における工程 O_{2j} の処理時間

最適化の目的

- 最大完了時刻の最小化

注：この授業のジョブショップでは，各ジョブに対して 1 つの機械で処理する工程が高々 1 つ



機械の環境

- ジョブショップ，機械数 $= 2$
- $p_{1j} =$ 機械 1 における工程 O_{1j} の処理時間
- $p_{2j} =$ 機械 2 における工程 O_{2j} の処理時間

最適化の目的

- 最大完了時刻の最小化

	J_1	J_2	J_3	J_4	J_5	J_6	J_7
M_1	4	2		1	3	2	5
M_2	1	5	3	2	3	5	

$\mathcal{J}_1$ $\mathcal{J}_2$

$J2 \mid\mid C_{\max}$

機械の環境

- ジョブショップ，機械数 $= 2$
- $p_{1j} =$ 機械 1 における工程 O_{1j} の処理時間
- $p_{2j} =$ 機械 2 における工程 O_{2j} の処理時間

最適化の目的

- 最大完了時刻の最小化

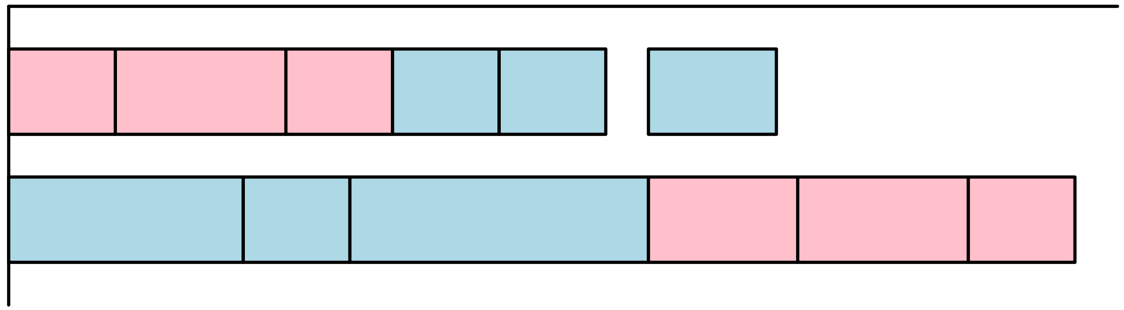
	J_1	J_2	J_3	J_4	J_5	J_6	J_7
M_1	4	2	0	1	3	2	5
M_2	1	5	3	2	3	5	0
		$\mathcal{J}_1$				$\mathcal{J}_2$	

特徴を持つ最適解

性質

$J2 \mid\mid C_{\max}$ では，次の条件を満たす最適解が存在する

- M_1 では，$\mathcal{J}_1$ の工程をすべて処理した後で，$\mathcal{J}_2$ の工程を処理する
- M_2 では，$\mathcal{J}_2$ の工程をすべて処理した後で，$\mathcal{J}_1$ の工程を処理する

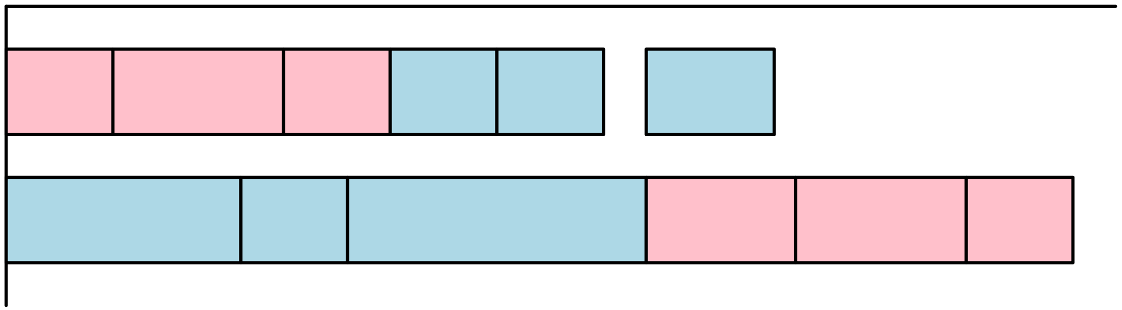


特徴を持つ最適解

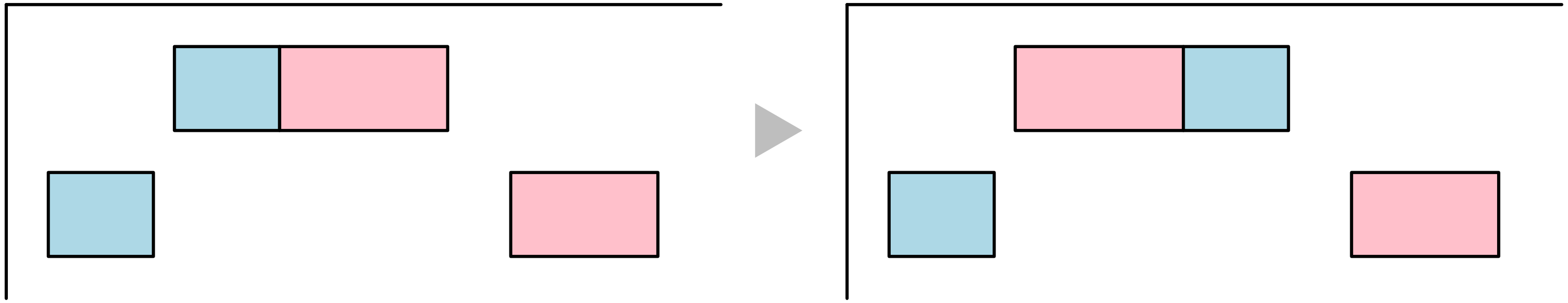
性質

$J2 \mid\mid C_{\max}$ では，次の条件を満たす最適解が存在する

- M_1 では，$\mathcal{J}_1$ の工程をすべて処理した後で，$\mathcal{J}_2$ の工程を処理する
- M_2 では，$\mathcal{J}_2$ の工程をすべて処理した後で，$\mathcal{J}_1$ の工程を処理する

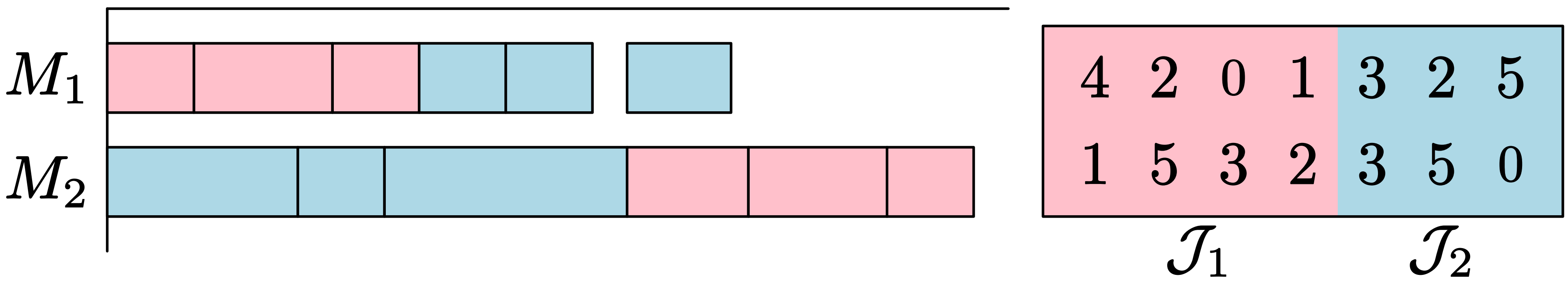


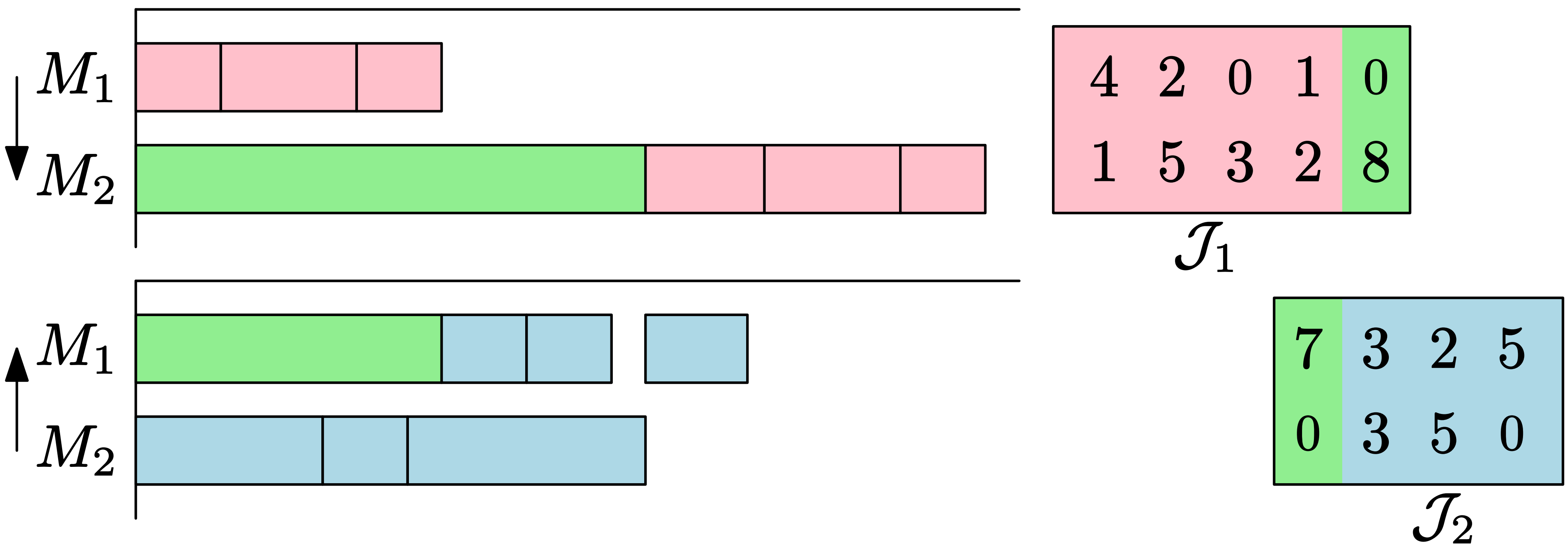
証明：



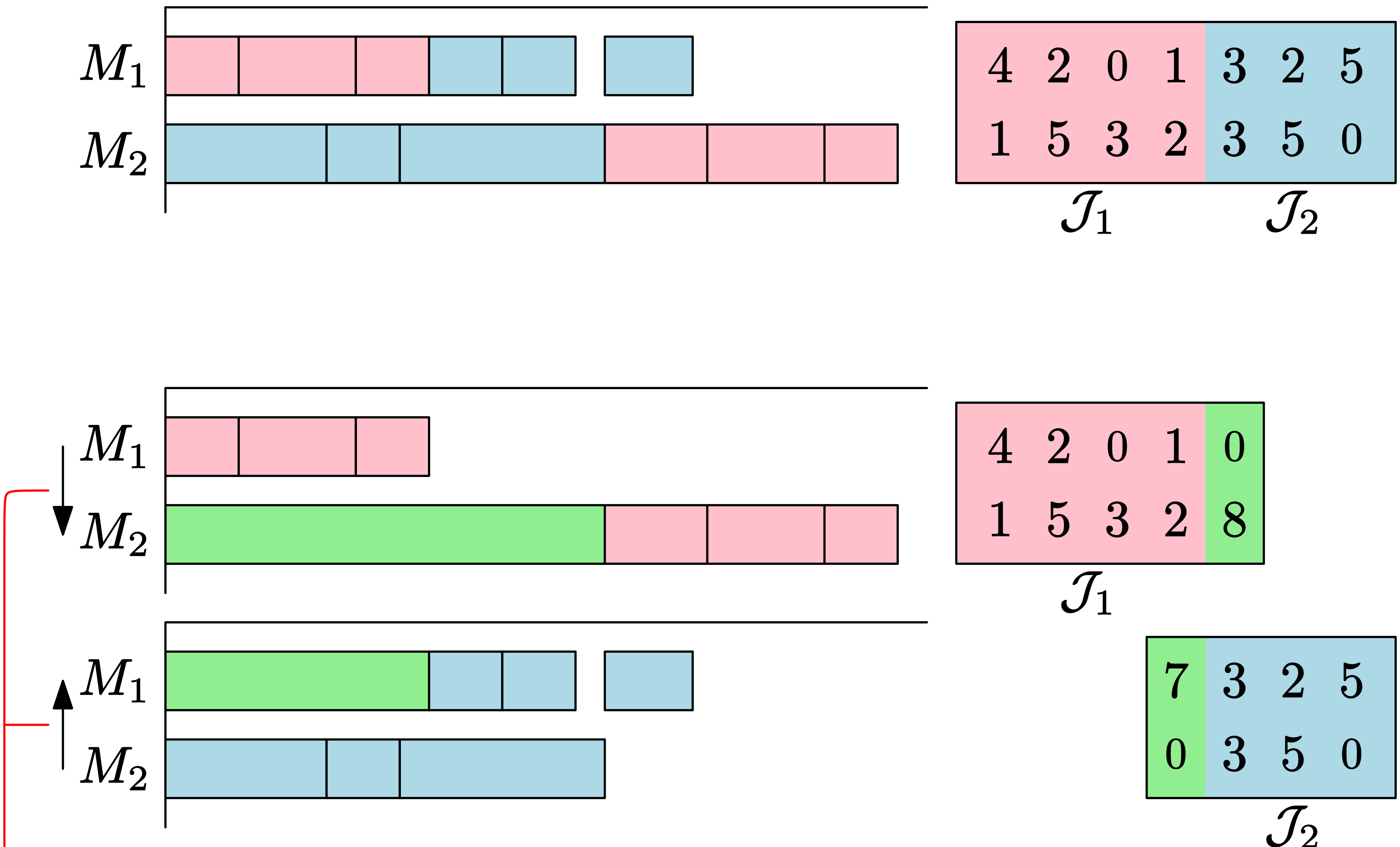
□

アルゴリズムの考え方


M_1
M_2
4 2 0 1 3 2 5
1 5 3 2 3 5 0
$\mathcal{J}_1$
$\mathcal{J}_2$


M_1
M_2
4 2 0 1 0
1 5 3 2 8
$\mathcal{J}_1$
M_1
M_2
7 3 2 5
0 3 5 0
$\mathcal{J}_2$

アルゴリズムの考え方



$F2 \mid\mid C_{\max}$ ($\because$ $O(n \log n)$ 時間で解ける)

$J2 \mid\mid C_{\max}$：アルゴリズム

$J2 \mid\mid C_{\max}$ の Johnson のアルゴリズム

1. $\mathcal{J}_1$ に対するスケジュールを
   $F2 \mid\mid C_{\max}$ の Johnson のアルゴリズムで作る
2. $\mathcal{J}_2$ に対するスケジュールを
   $F2 \mid\mid C_{\max}$ の Johnson のアルゴリズムで作る
3. この 2 つのスケジュールを組み合わせる
   - M_1 では $\mathcal{J}_1$ を処理してから $\mathcal{J}_2$ を処理する
   - M_2 では $\mathcal{J}_2$ を処理してから $\mathcal{J}_1$ を処理する

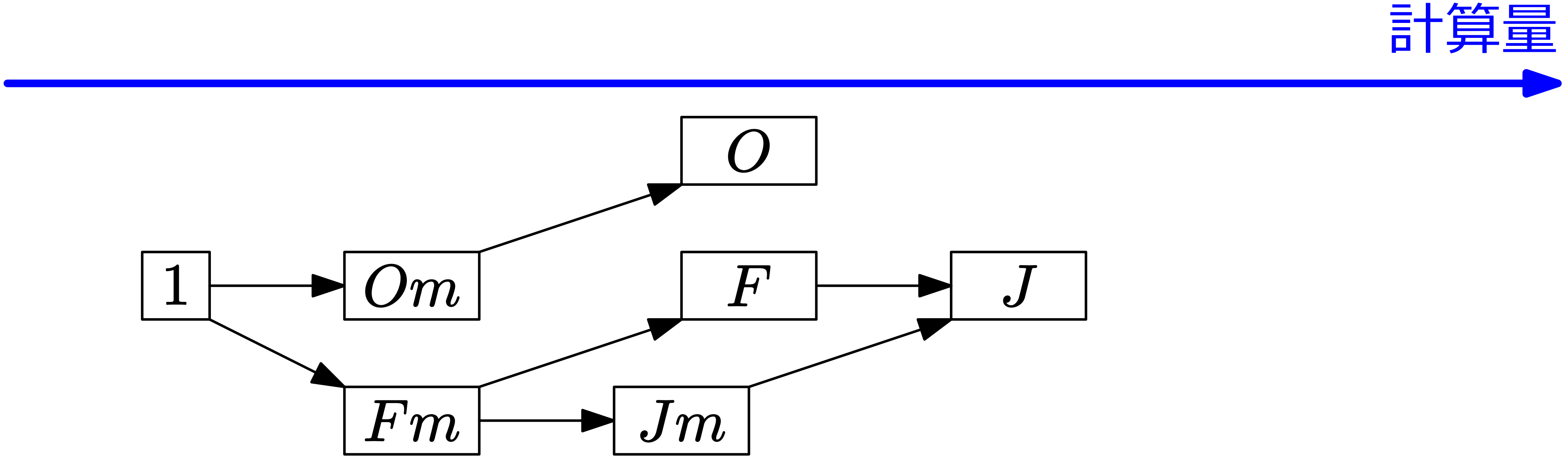
定理 (Johnson '54)

Johnson のアルゴリズムは
$J2 \mid\mid C_{\max}$ を $O(n \log n)$ 時間で解く

正当性と計算量は ここまでの議論から分かる

今日のまとめ

- $F2 \mid\mid C_{\max}$ $O(n \log n)$ (Johnson ’54)
- $F3 \mid\mid C_{\max}$ 強 NP 困難 (Garey, Johnson, Sethi ’76)
- $J2 \mid\mid C_{\max}$ $O(n \log n)$ (Johnson ’54)



次回の予告

- $O2 \mid\mid C_{\max}$　　$O(n)$
- $O3 \mid\mid C_{\max}$　　弱 NP 困難

<u>未解決問題</u>：$O3 \mid\mid C_{\max}$ は強 NP 困難か？