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 12 回

ショップ・スケジューリング：機械数が可変

岡本 吉央　(電気通信大学)

okamotoy@uec.ac.jp

2025 年 1 月 14 日

最終更新：2025 年 1 月 14 日　12:44

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)

[復習] 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

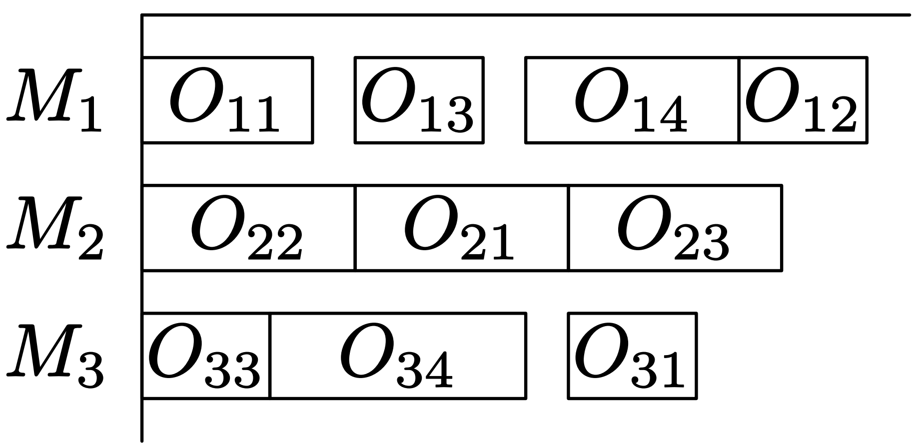
注：
1 つのジョブにおける複数の工程を同じ機械で処理する場合も考えられているが，この授業では扱わない

	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}

[復習] 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

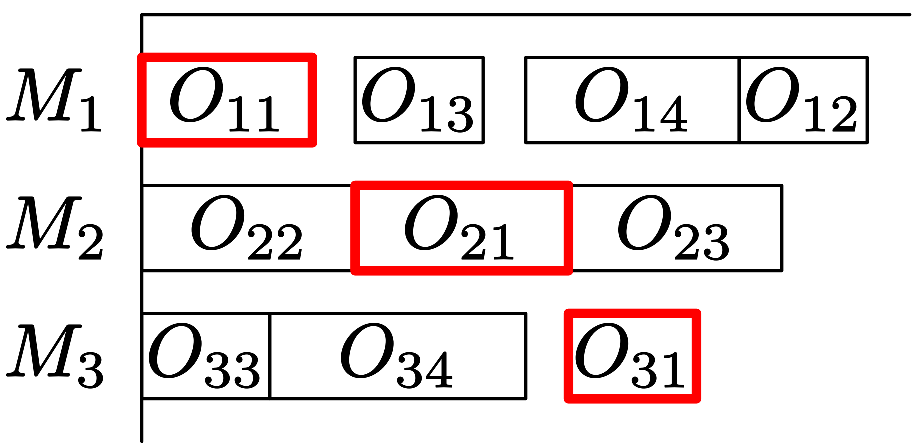


$$
\begin{array}{c|cccc|}
 & J_1 & J_2 & J_3 & J_4 \\
\hline
M_1 & O_{11} & O_{12} & O_{13} & O_{14} \\
M_2 & O_{21} & O_{22} & O_{23} & \\
M_3 & O_{31} & & O_{33} & O_{34} \\
\hline
\end{array}
$$

[復習] 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

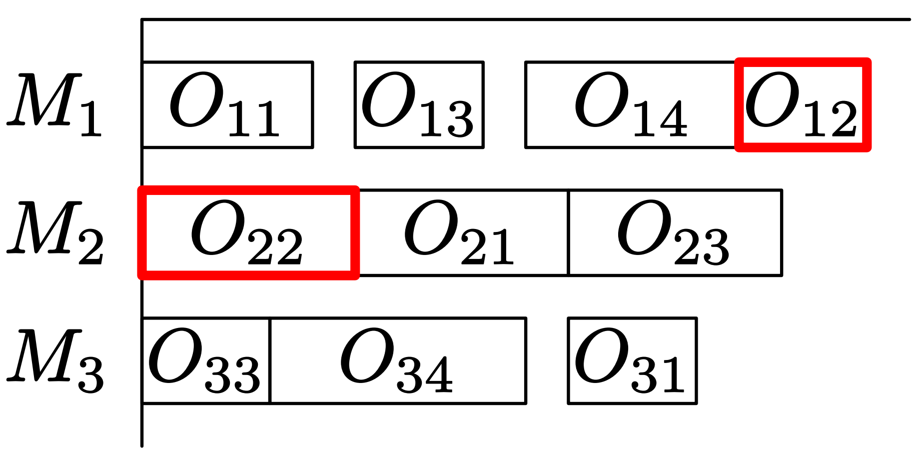


	J_1	J_2	J_3	J_4
M_1	O_{11}	O_{12}	O_{13}	O_{14}
M_2	O_{21}	O_{22}	O_{23}	
M_3	O_{31}		O_{33}	O_{34}

[復習] 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

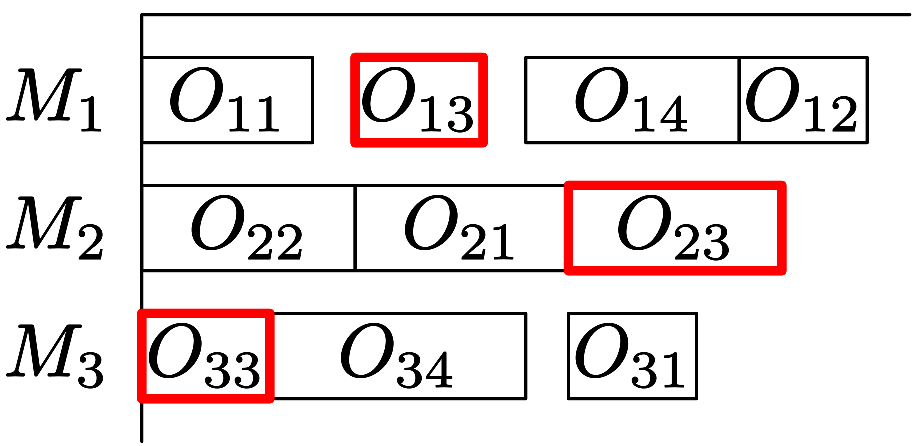


	J_1	J_2	J_3	J_4
M_1	O_{11}	O_{12}	O_{13}	O_{14}
M_2	O_{21}	O_{22}	O_{23}	
M_3	O_{31}		O_{33}	O_{34}

[復習] 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

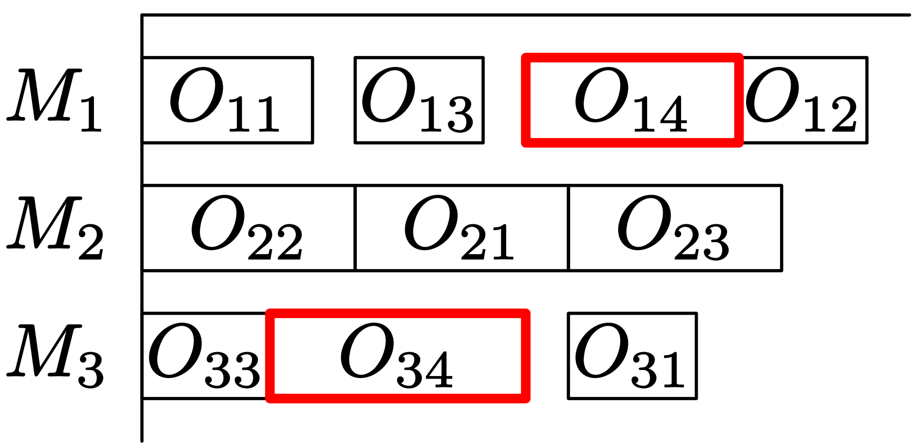


	J_1	J_2	J_3	J_4
M_1	O_{11}	O_{12}	O_{13}	O_{14}
M_2	O_{21}	O_{22}	O_{23}	
M_3	O_{31}		O_{33}	O_{34}

[復習] 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}

[復習] 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

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

[復習] 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

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

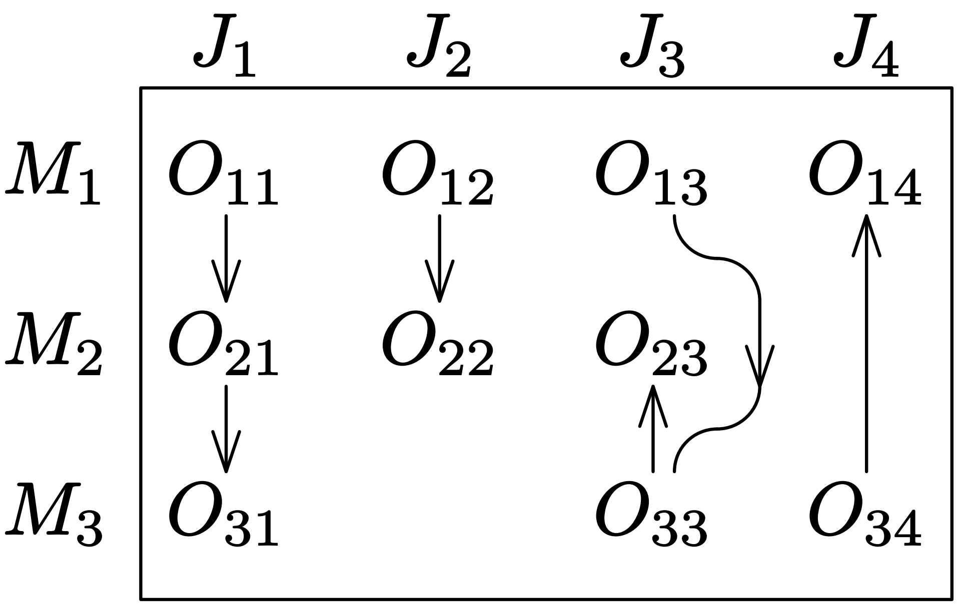
直感：流れ作業

[復習] 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

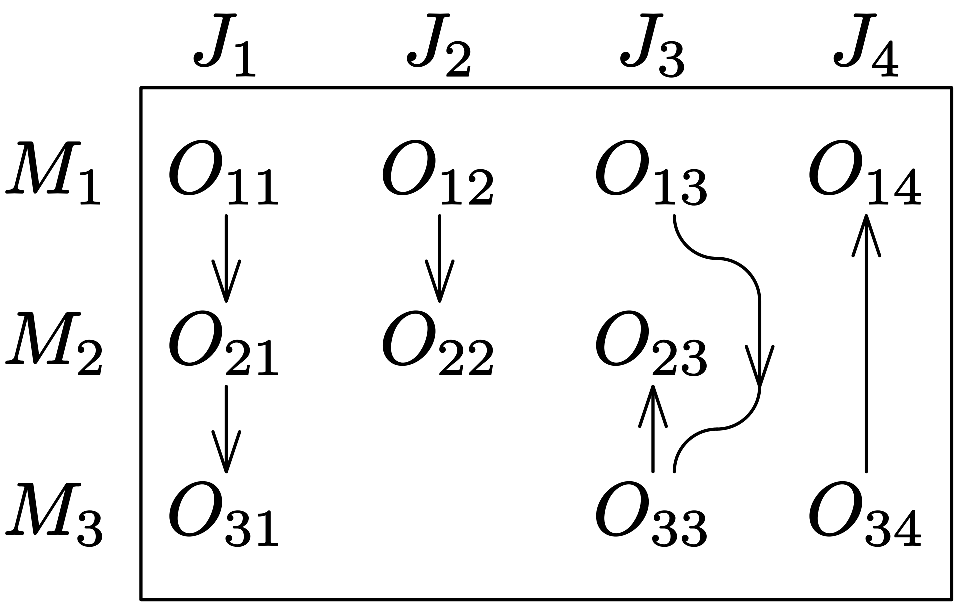


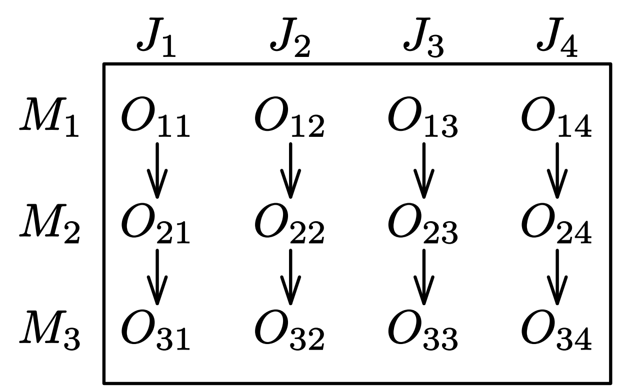
注：フローショップはジョブショップの特別な場合

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

[復習] 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

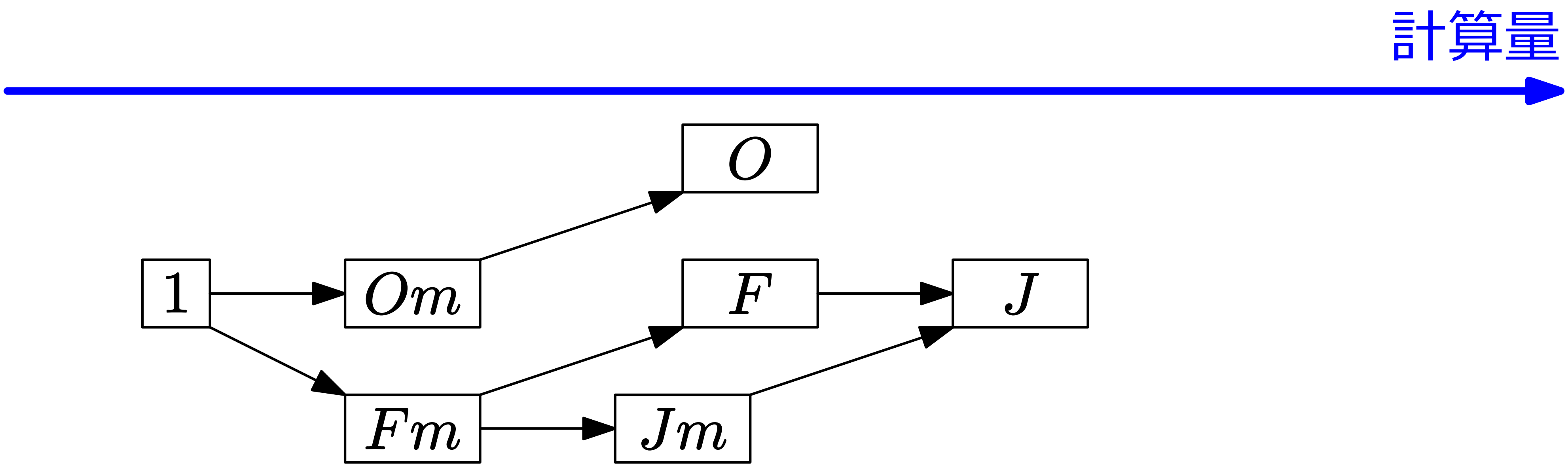
- Om オープンショップ 機械数は m で固定
- Fm フローショップ 機械数は m で固定
- Jm ジョブショップ 機械数は m で固定

[復習] 3 つ組記法

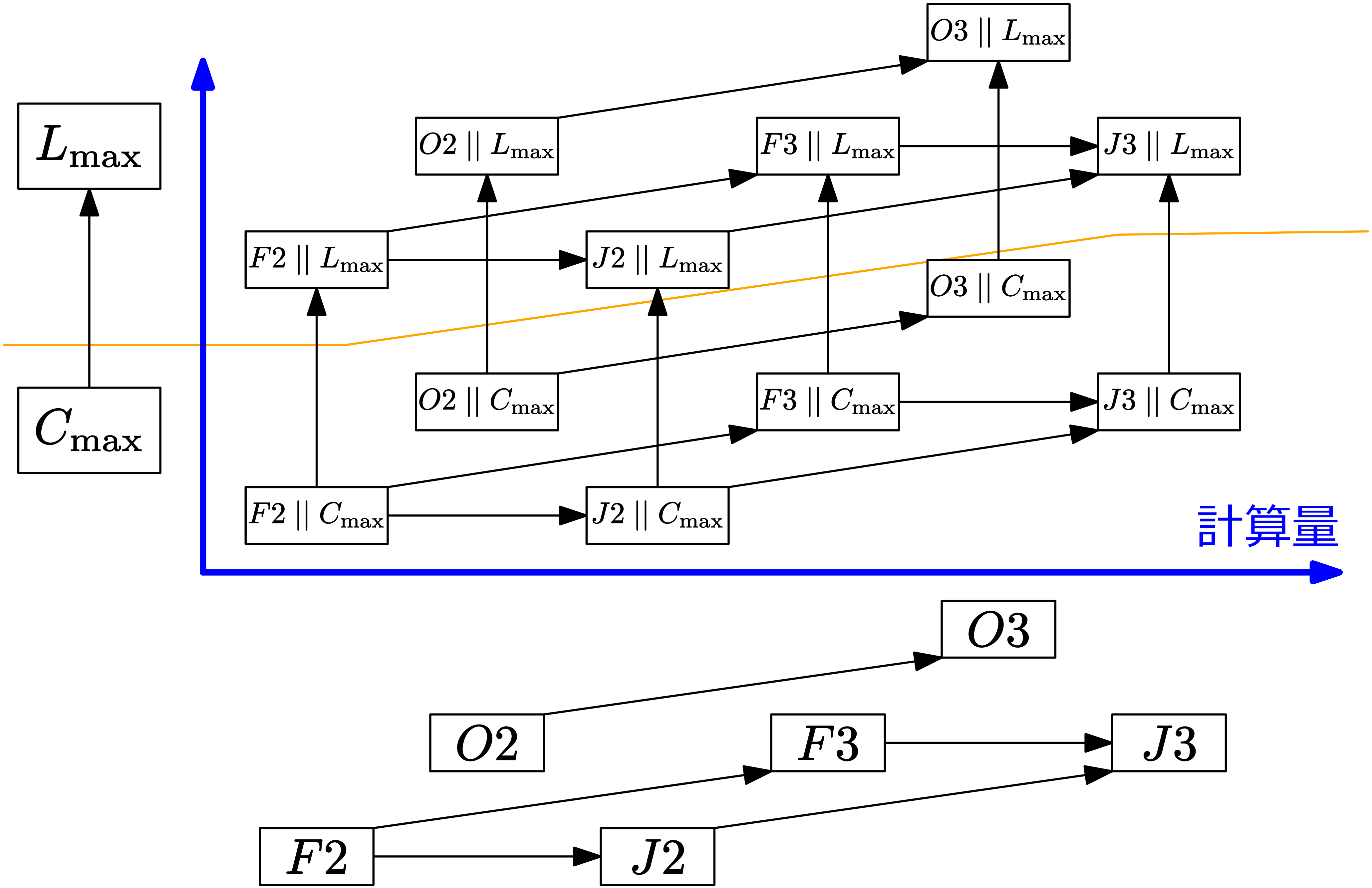
3 つ組記法 $\alpha \mid \beta \mid \gamma$ の α に次を書く ($\alpha =$ 機械の環境)

- 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

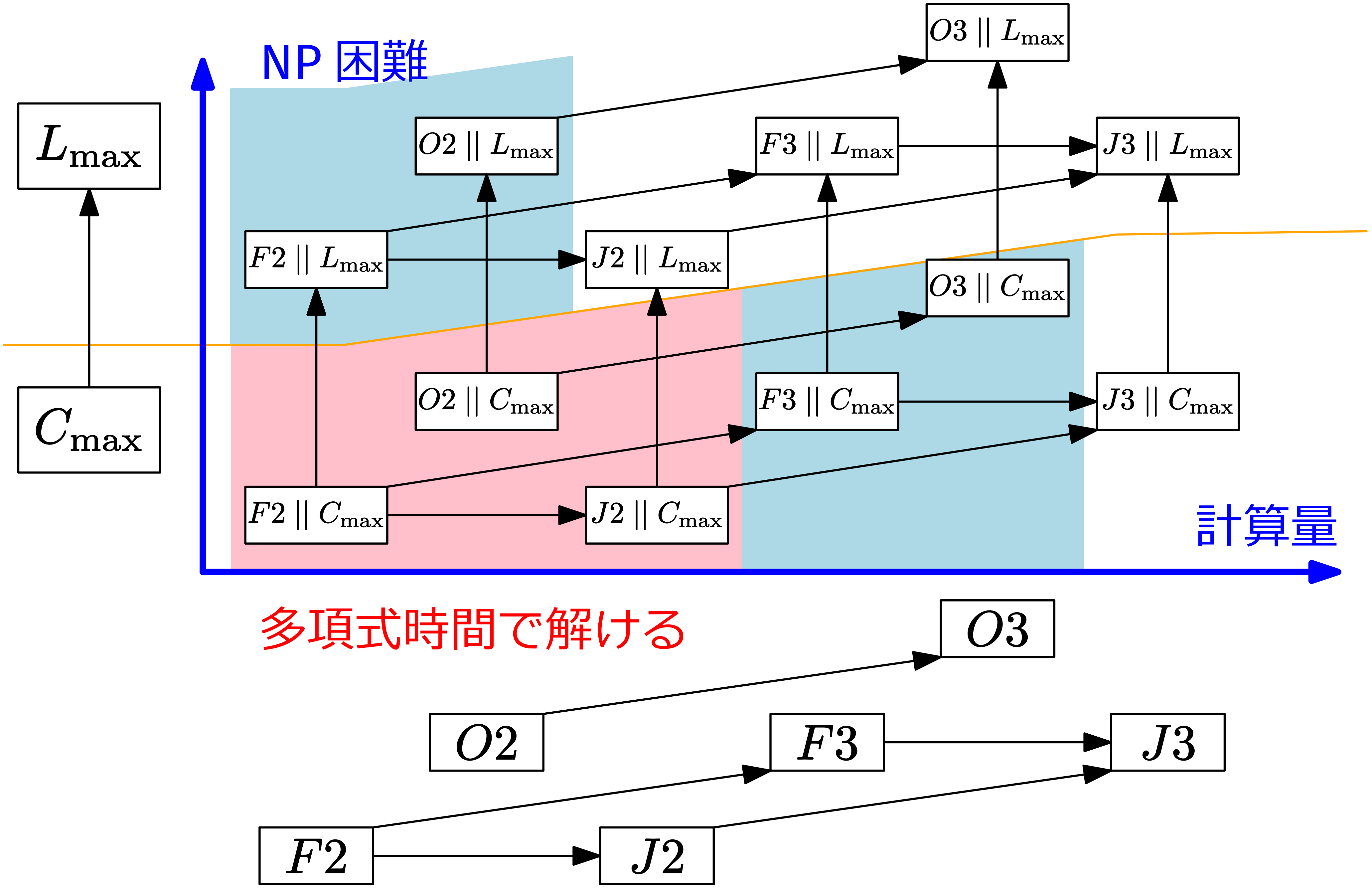
- Om　オープンショップ　機械数は m で固定
- Fm　フローショップ　機械数は m で固定
- Jm　ジョブショップ　機械数は m で固定



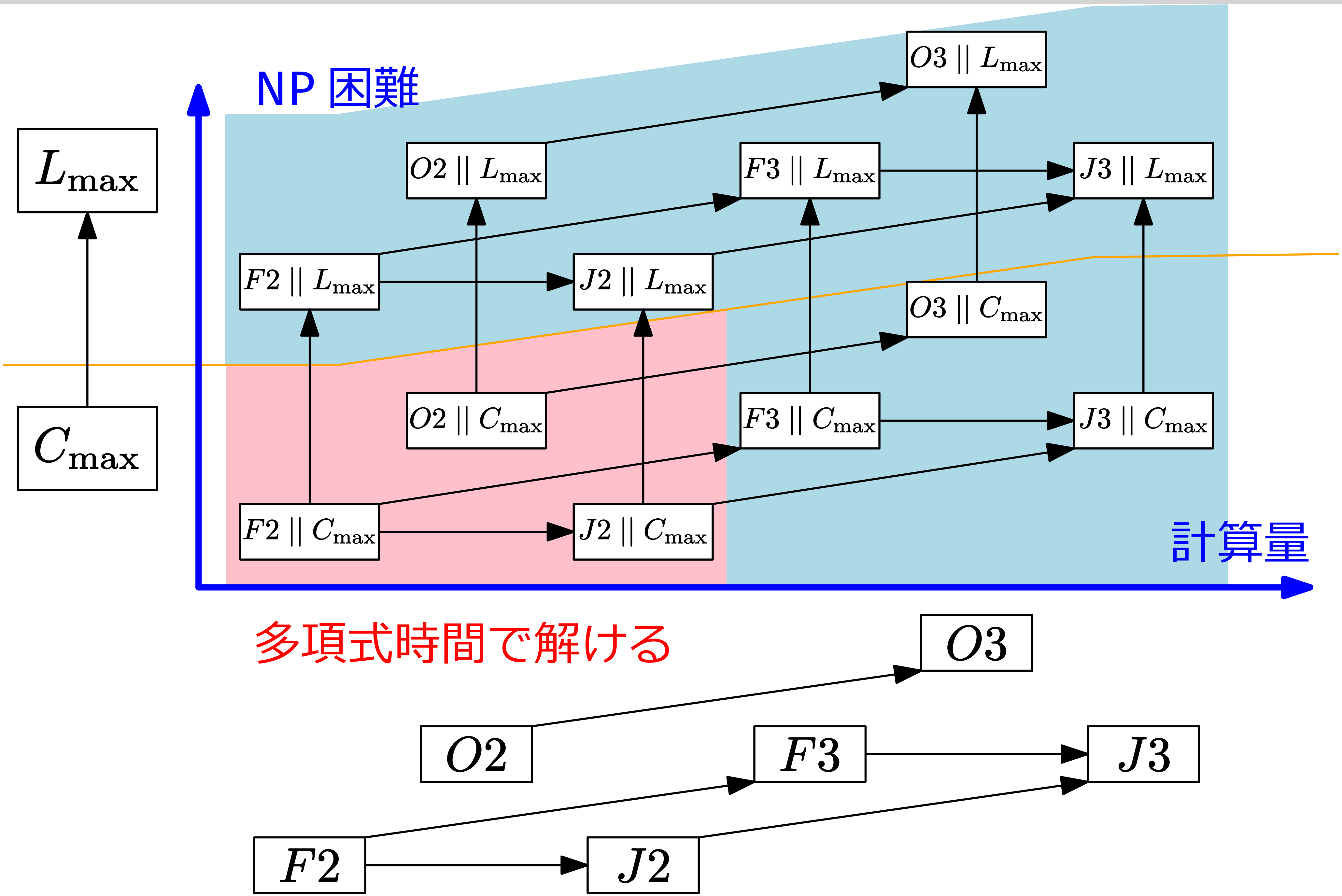
前回，前々回の内容


$O3 \mid\mid L_{\max}$
$L_{\max}$
$O2 \mid\mid L_{\max}$
$F3 \mid\mid L_{\max}$
$J3 \mid\mid L_{\max}$
$F2 \mid\mid L_{\max}$
$J2 \mid\mid L_{\max}$
$O3 \mid\mid C_{\max}$
$O2 \mid\mid C_{\max}$
$F3 \mid\mid C_{\max}$
$J3 \mid\mid C_{\max}$
$C_{\max}$
$F2 \mid\mid C_{\max}$
$J2 \mid\mid C_{\max}$
計算量
$O3$
$O2$
$F3$
$J3$
$F2$
$J2$

前回，前々回の内容


NP 困難
計算量
多項式時間で解ける
$L_{\max}$
$C_{\max}$
$O3 \,||\, L_{\max}$
$O2 \,||\, L_{\max}$
$F3 \,||\, L_{\max}$
$J3 \,||\, L_{\max}$
$F2 \,||\, L_{\max}$
$J2 \,||\, L_{\max}$
$O3 \,||\, C_{\max}$
$O2 \,||\, C_{\max}$
$F3 \,||\, C_{\max}$
$J3 \,||\, C_{\max}$
$F2 \,||\, C_{\max}$
$J2 \,||\, C_{\max}$
$O3$
$O2$
$F3$
$J3$
$F2$
$J2$

前回，前々回の内容


NP 困難
$L_{\max}$
$C_{\max}$
$O3 \mid\mid L_{\max}$
$O2 \mid\mid L_{\max}$
$F3 \mid\mid L_{\max}$
$J3 \mid\mid L_{\max}$
$F2 \mid\mid L_{\max}$
$J2 \mid\mid L_{\max}$
$O3 \mid\mid C_{\max}$
$O2 \mid\mid C_{\max}$
$F3 \mid\mid C_{\max}$
$J3 \mid\mid C_{\max}$
$F2 \mid\mid C_{\max}$
$J2 \mid\mid C_{\max}$
計算量
多項式時間で解ける
$O3$
$O2$
$F3$
$J3$
$F2$
$J2$

ショップ・スケジューリングに対する近似アルゴリズム

$F \mid\mid C_{\max}$ (m：機械数)

- m 近似
- $\lceil m/2 \rceil$ 近似

$O \mid\mid C_{\max}$

- 2 近似

[復習] α 近似

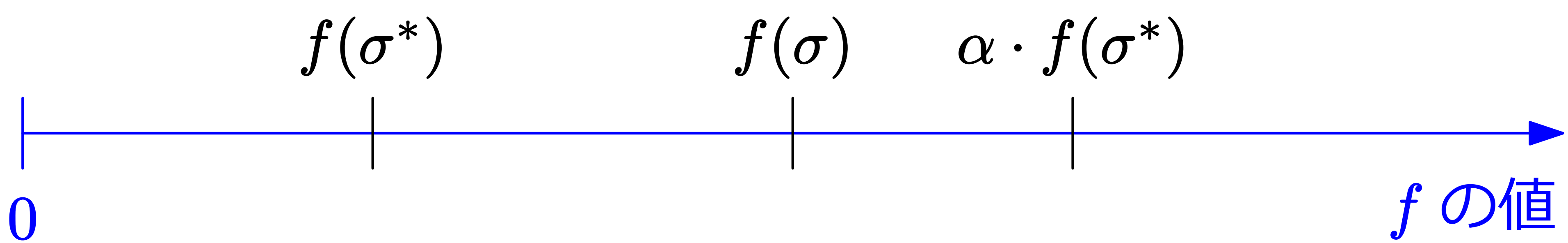
$\alpha \geq 1$ は実数

定義：α 近似 (α-approximation)

スケジュール σ が α **近似** であるとは,
最適スケジュール σ^* に対して

$$f(\sigma^*) \leq f(\sigma) \leq \alpha \cdot f(\sigma^*)$$

を満たすこと (ただし，f は目的関数)



別の表現：$1 \leq \dfrac{f(\sigma)}{f(\sigma^*)} \leq \alpha \quad (f(\sigma^*) > 0$ のとき)

$\alpha \geq 1$ は実数

定義：α 近似アルゴリズム (α-approximation algorithm)

アルゴリズム $\mathcal{A}$ が α **近似** であるとは，
任意の入力に対して，$\mathcal{A}$ の出力が α 近似であること

<u>気持ち</u>：α が小さい $\Leftrightarrow$ よいアルゴリズム

$$\text{入力} \longrightarrow \boxed{\mathcal{A}} \longrightarrow \alpha \text{ 近似}$$

用語：近似比，近似保証

この α を $\mathcal{A}$ の $\begin{cases} \textbf{近似比} & \text{(approximation ratio)} \\ \textbf{近似保証} & \text{(approximation guarantee)} \end{cases}$

と呼ぶことがある

1. **$F \mid\mid C_{\max}$ の近似アルゴリズム**
2. $O \mid\mid C_{\max}$ の近似アルゴリズム
3. 近似可能性と近似不可能性

- T. Gonzalez, S. Sahni, Flowshop and jobshop schedules: complexity and approximation. *Operations Research* 26 (1978) 36–52.

$F \mid\mid C_{\max}$ (m 近似)

リスト・スケジューリング (任意リスト)

1. ジョブを任意の順に並べる
2. すべての機械に対して，その順でジョブを処理する (できるだけ休みなく)

$$
\begin{array}{c|cccc|}
 & J_1 & J_2 & J_3 & J_4 \\
\hline
M_1 & 2 & 5 & 2 & 3 \\
 & \downarrow & \downarrow & \downarrow & \downarrow \\
M_2 & 3 & 3 & 4 & 4 \\
 & \downarrow & \downarrow & \downarrow & \downarrow \\
M_3 & 4 & 5 & 1 & 2 \\
\hline
\end{array}
$$

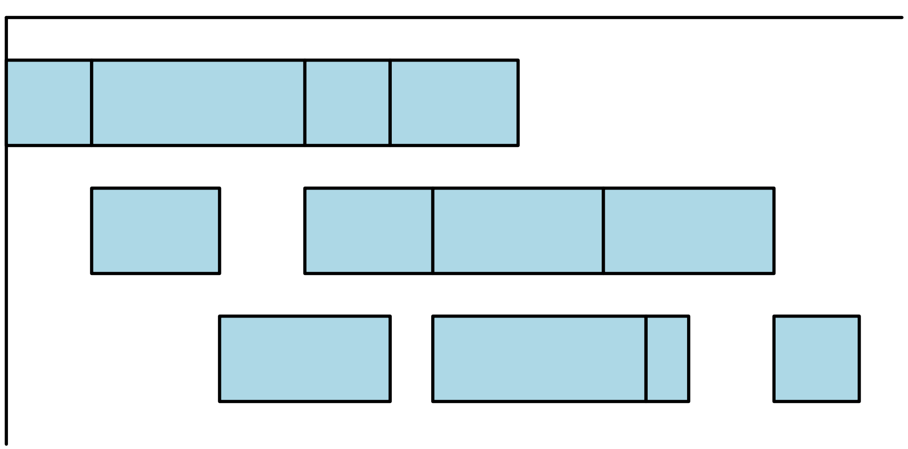
リスト・スケジューリング (任意リスト)

1. ジョブを任意の順に並べる
2. すべての機械に対して，その順でジョブを処理する (できるだけ休みなく)

	J_1	J_2	J_3	J_4
M_1	2	5	2	3
M_2	3	3	4	4
M_3	4	5	1	2

リスト・スケジューリング (任意リスト)

1. ジョブを任意の順に並べる
2. すべての機械に対して，その順でジョブを処理する (できるだけ休みなく)



	J_1	J_2	J_3	J_4
M_1	2	5	2	3
	↓	↓	↓	↓
M_2	3	3	4	4
	↓	↓	↓	↓
M_3	4	5	1	2

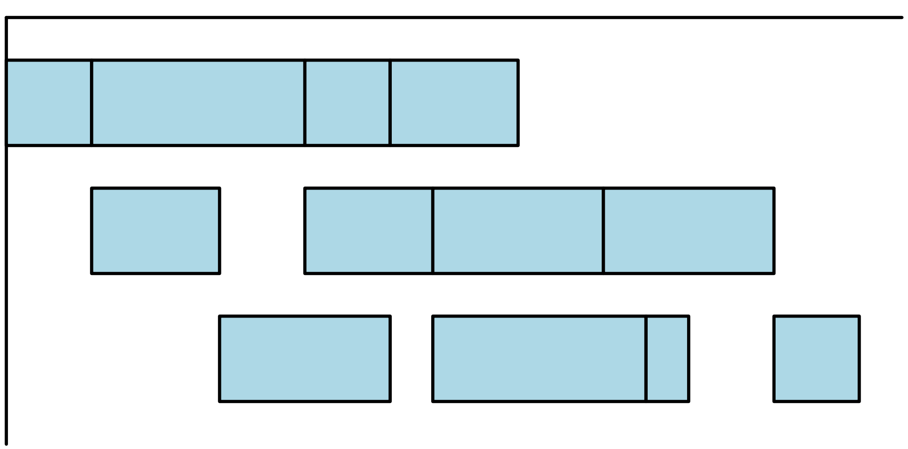
$F \mid\mid C_{\max}$ (m 近似)

リスト・スケジューリング (任意リスト)

1. ジョブを任意の順に並べる
2. すべての機械に対して，その順でジョブを処理する (できるだけ休みなく)

定理 (Gonzalez, Sahni ’78)

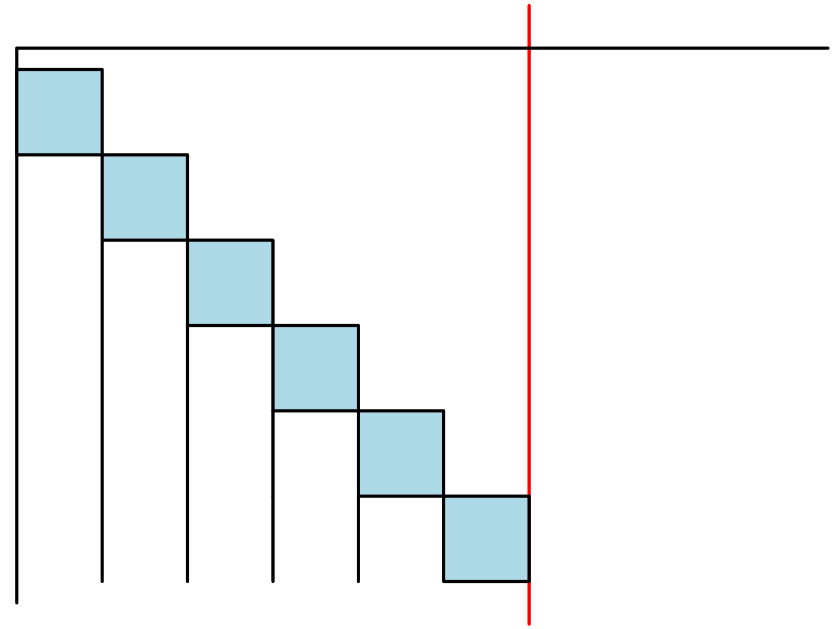
リスト・スケジューリングは m 近似アルゴリズム



	J_1	J_2	J_3	J_4
M_1	2	5	2	3
M_2	3	3	4	4
M_3	4	5	1	2

$F \mid\mid C_{\max}$ (m 近似)：タイトな例

	J_1	J_2	J_3	J_4	J_5	J_6
M_1	1	0	0	0	0	0
M_2	0	1	0	0	0	0
M_3	0	0	1	0	0	0
M_4	0	0	0	1	0	0
M_5	0	0	0	0	1	0
M_6	0	0	0	0	0	1

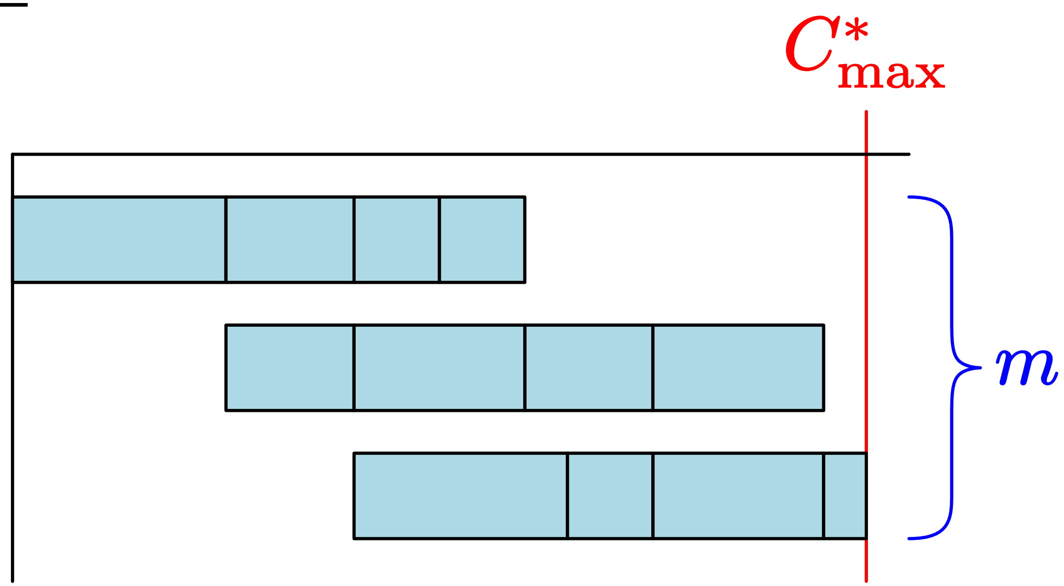


性質：最適値の下界

最適値を $C^*_{\max}$ とするとき，次が成り立つ

- $m \cdot C^*_{\max} \geq \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij}$

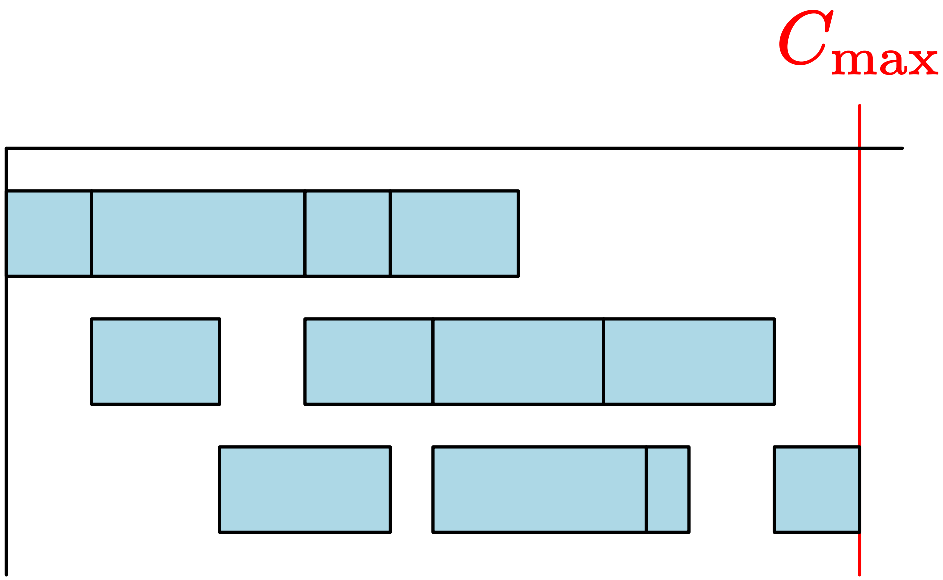
証明：



$F \parallel C_{\max}$：近似比の導出

証明：アルゴリズムの出力値を $C_{\max}$，
最適値を $C^*_{\max}$ とすると

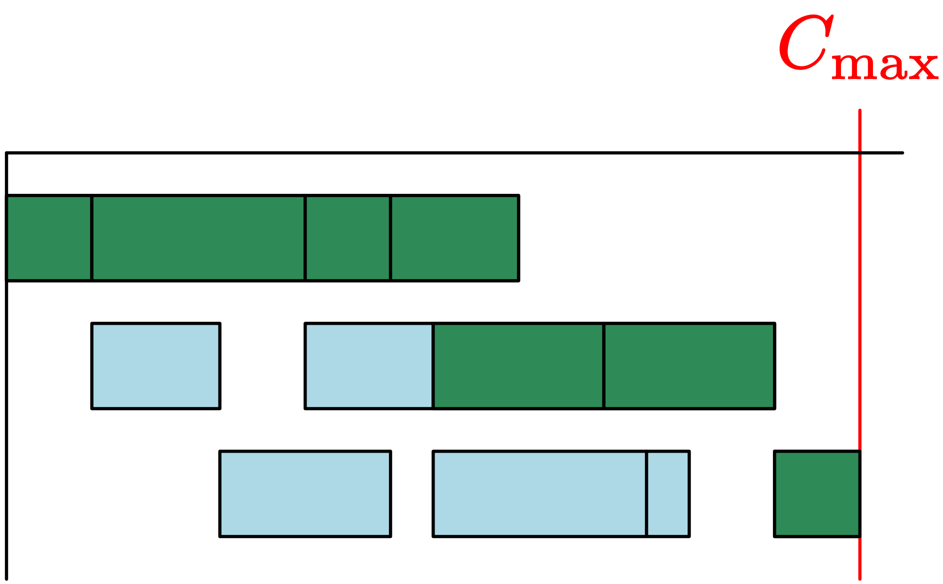
$$C_{\max} \leq \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij} \leq m \cdot C^*_{\max} \qquad \square$$



性質：$m \cdot C^*_{\max} \geq \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}$

証明：アルゴリズムの出力値を $C_{\max}$，
最適値を $C^*_{\max}$ とすると

$$C_{\max} \leq \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} \leq m \cdot C^*_{\max} \qquad \square$$



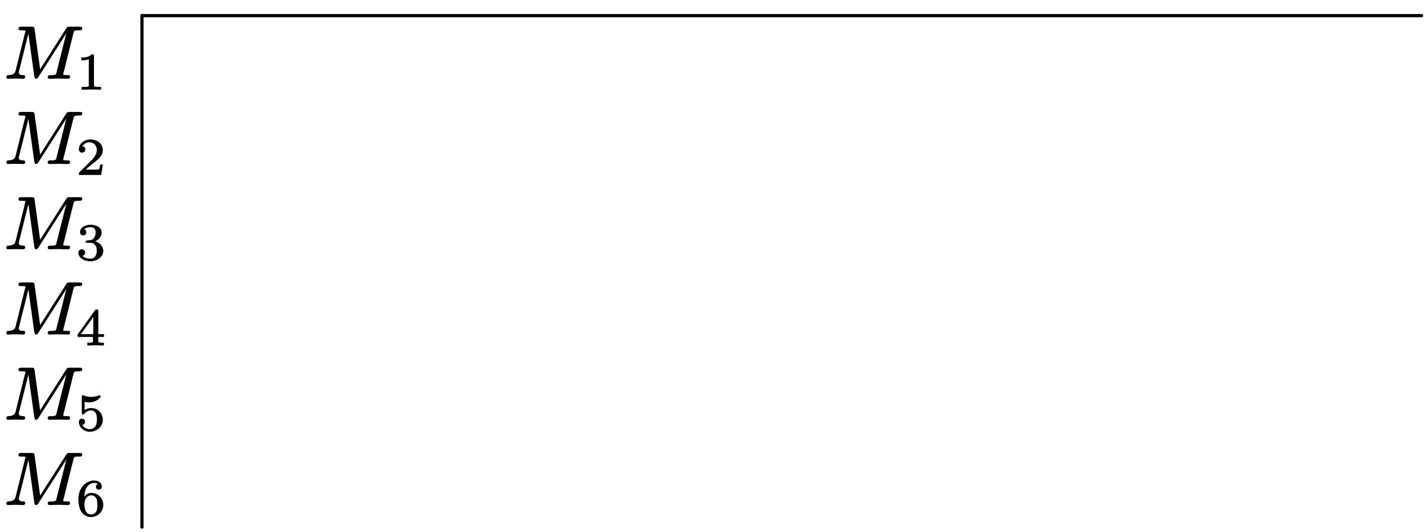
性質：$m \cdot C^*_{\max} \geq \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij}$

$F \mid\mid C_{\max}$ ($\lceil m/2 \rceil$ 近似)：アルゴリズム

繰り返し Johnson アルゴリズム

- M_1, M_2 に対する $F2 \mid\mid C_{\max}$ を最適に解く
- M_3, M_4 に対する $F2 \mid\mid C_{\max}$ を最適に解く

  $\vdots$
- M_{m-1}, M_m に対する $F2 \mid\mid C_{\max}$ を最適に解く
- 得られたスケジュールを結合する

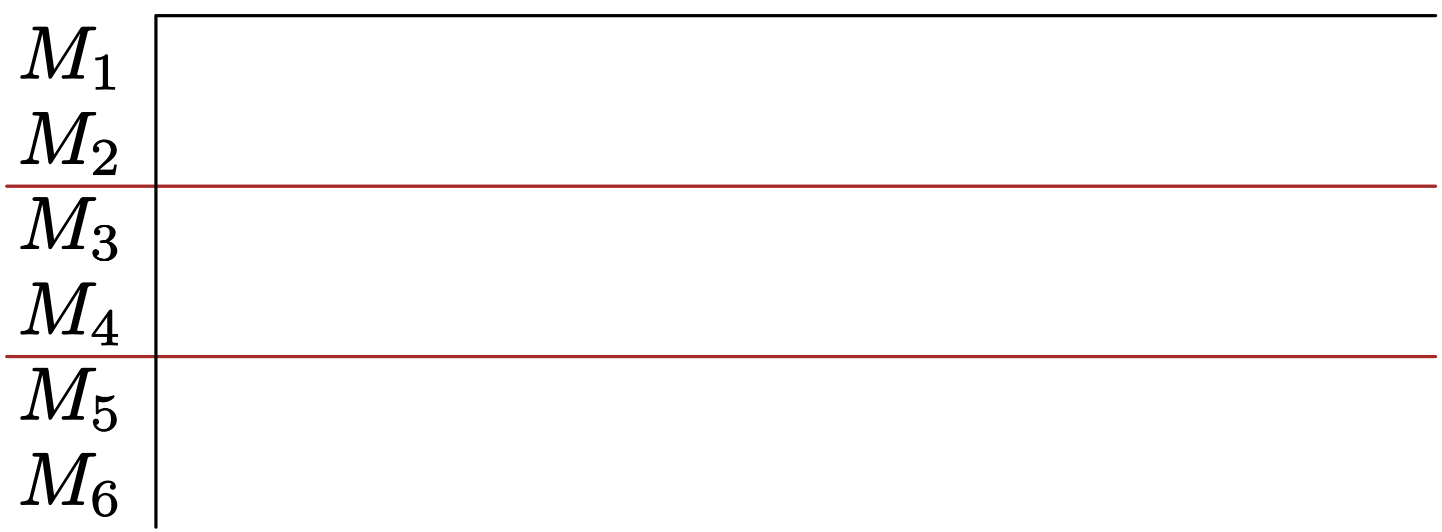


$F \,||\, C_{\max}$ ($\lceil m/2 \rceil$ 近似)：アルゴリズム

繰り返し Johnson アルゴリズム

- M_1, M_2 に対する $F2 \,||\, C_{\max}$ を最適に解く
- M_3, M_4 に対する $F2 \,||\, C_{\max}$ を最適に解く

  $\vdots$
- M_{m-1}, M_m に対する $F2 \,||\, C_{\max}$ を最適に解く
- 得られたスケジュールを結合する

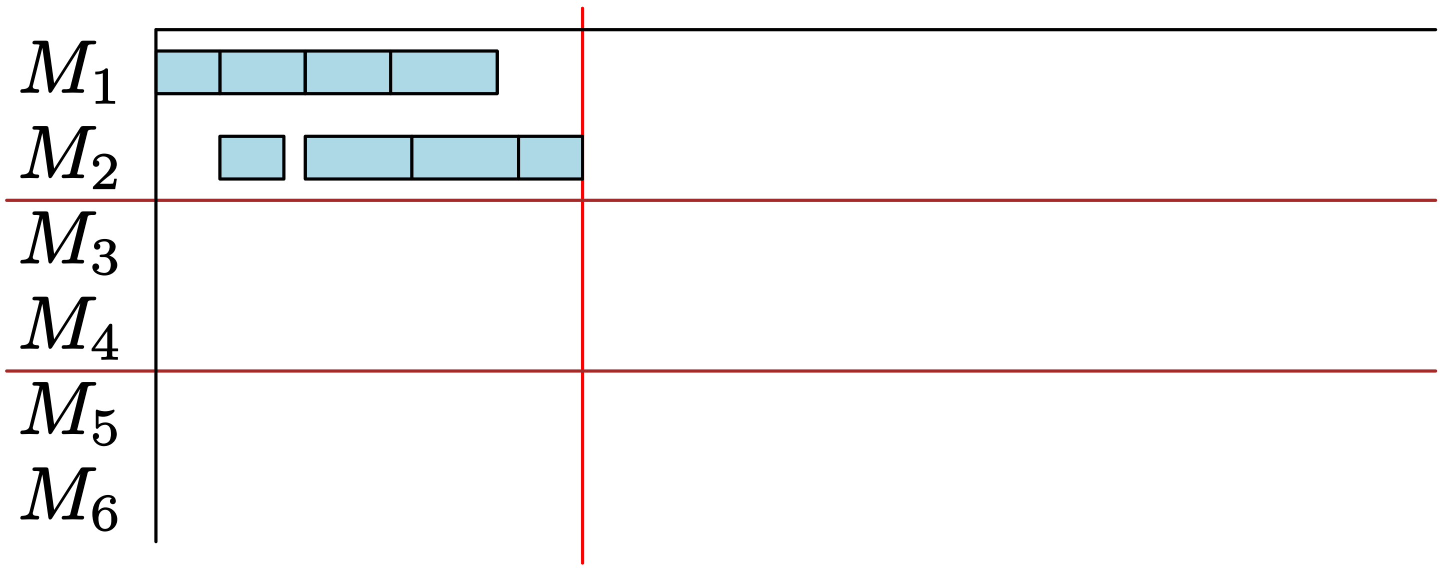


繰り返し Johnson アルゴリズム

- M_1, M_2 に対する $F2 \mid\mid C_{\max}$ を最適に解く
- M_3, M_4 に対する $F2 \mid\mid C_{\max}$ を最適に解く

  $\vdots$

- M_{m-1}, M_m に対する $F2 \mid\mid C_{\max}$ を最適に解く
- 得られたスケジュールを結合する

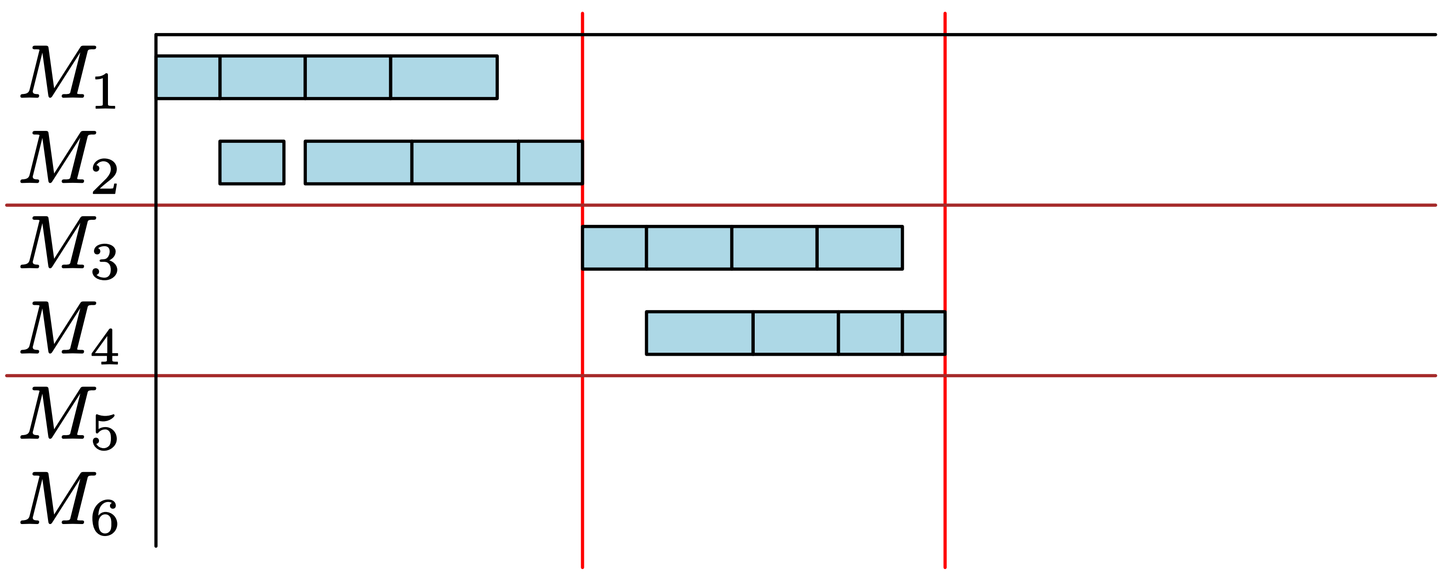


$F \,||\, C_{\max}$ ($\lceil m/2 \rceil$ 近似)：アルゴリズム

繰り返し Johnson アルゴリズム

- M_1, M_2 に対する $F2 \,||\, C_{\max}$ を最適に解く
- M_3, M_4 に対する $F2 \,||\, C_{\max}$ を最適に解く

  $\vdots$
- M_{m-1}, M_m に対する $F2 \,||\, C_{\max}$ を最適に解く
- 得られたスケジュールを結合する

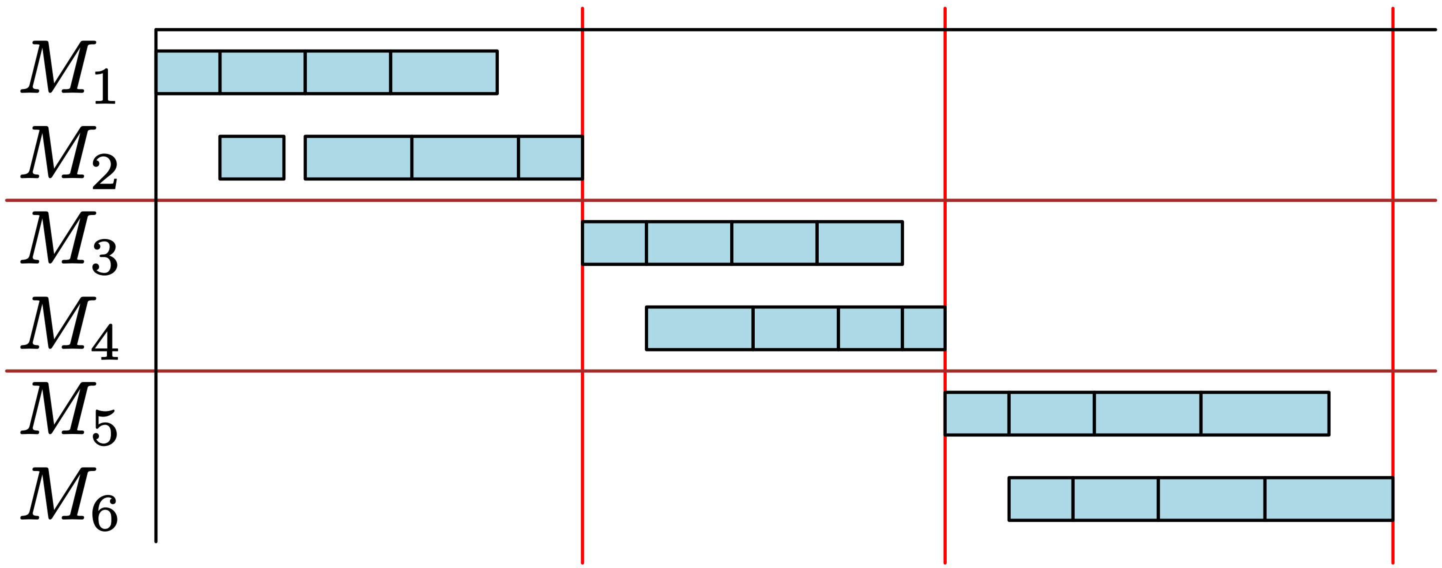


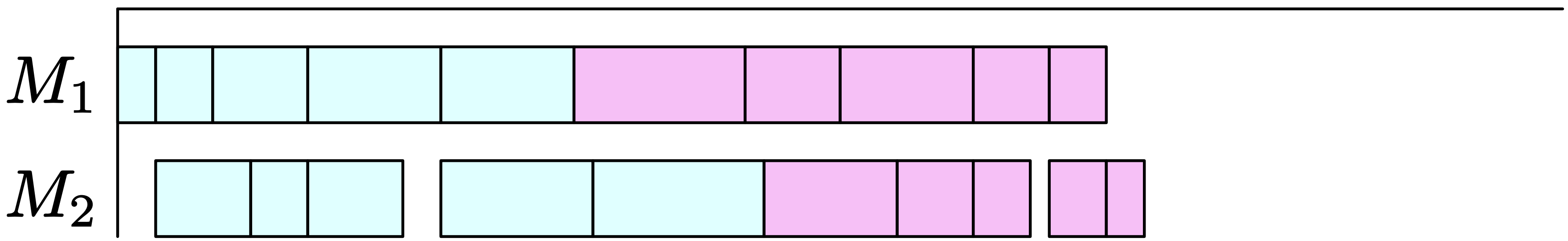
繰り返し Johnson アルゴリズム

- M_1, M_2 に対する $F2 \mid\mid C_{\max}$ を最適に解く
- M_3, M_4 に対する $F2 \mid\mid C_{\max}$ を最適に解く

$\vdots$

- M_{m-1}, M_m に対する $F2 \mid\mid C_{\max}$ を最適に解く
- 得られたスケジュールを結合する





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

定理 (Johnson ’54)

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

$F \mid\mid C_{\max}$ ($\lceil m/2 \rceil$ 近似)：定理

定理 (Gonzalez, Sahni '78)

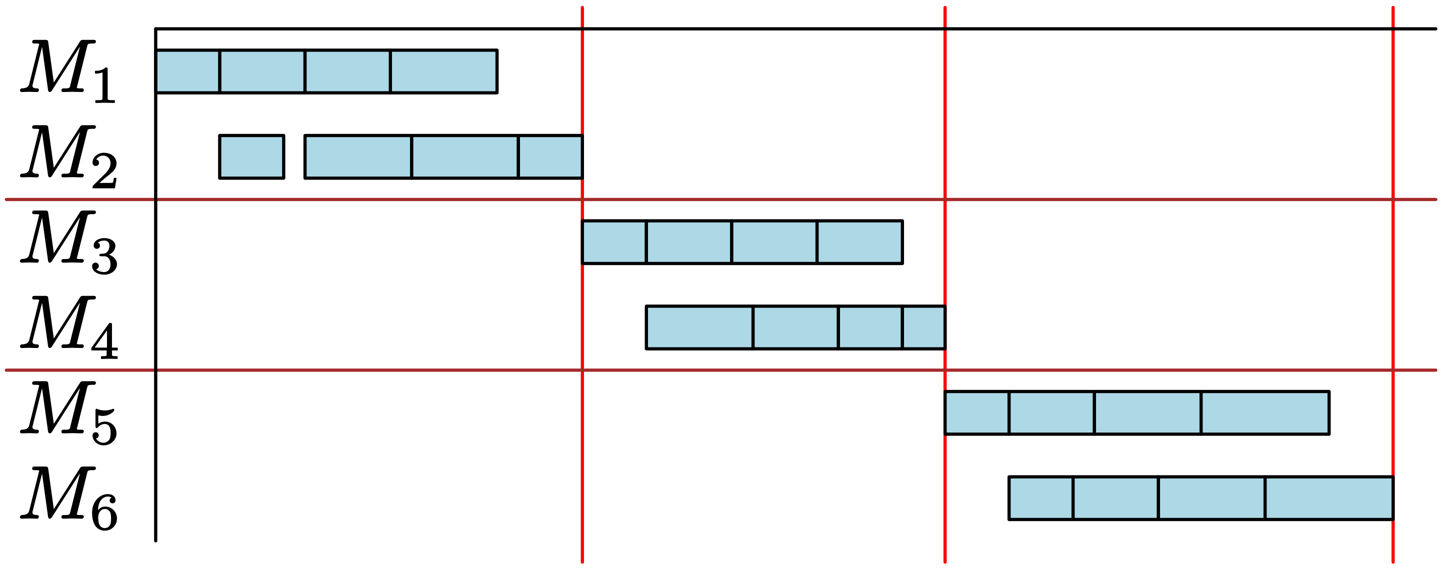
繰り返し Johnson アルゴリズムは
$\lceil m/2 \rceil$ 近似アルゴリズム

繰り返し Johnson アルゴリズム

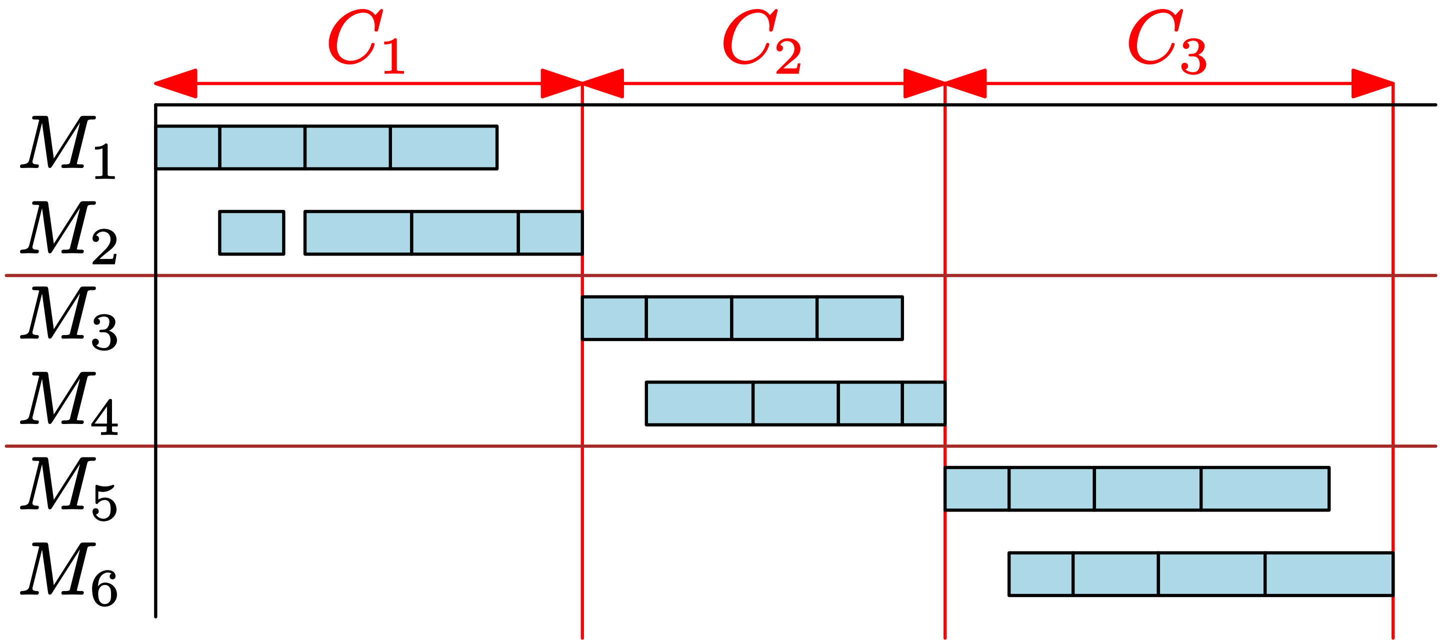
- M_1, M_2 に対する $F2 \mid\mid C_{\max}$ を最適に解く
- M_3, M_4 に対する $F2 \mid\mid C_{\max}$ を最適に解く

  $\vdots$
- M_{m-1}, M_m に対する $F2 \mid\mid C_{\max}$ を最適に解く
- 得られたスケジュールを結合する

アルゴリズムの出力

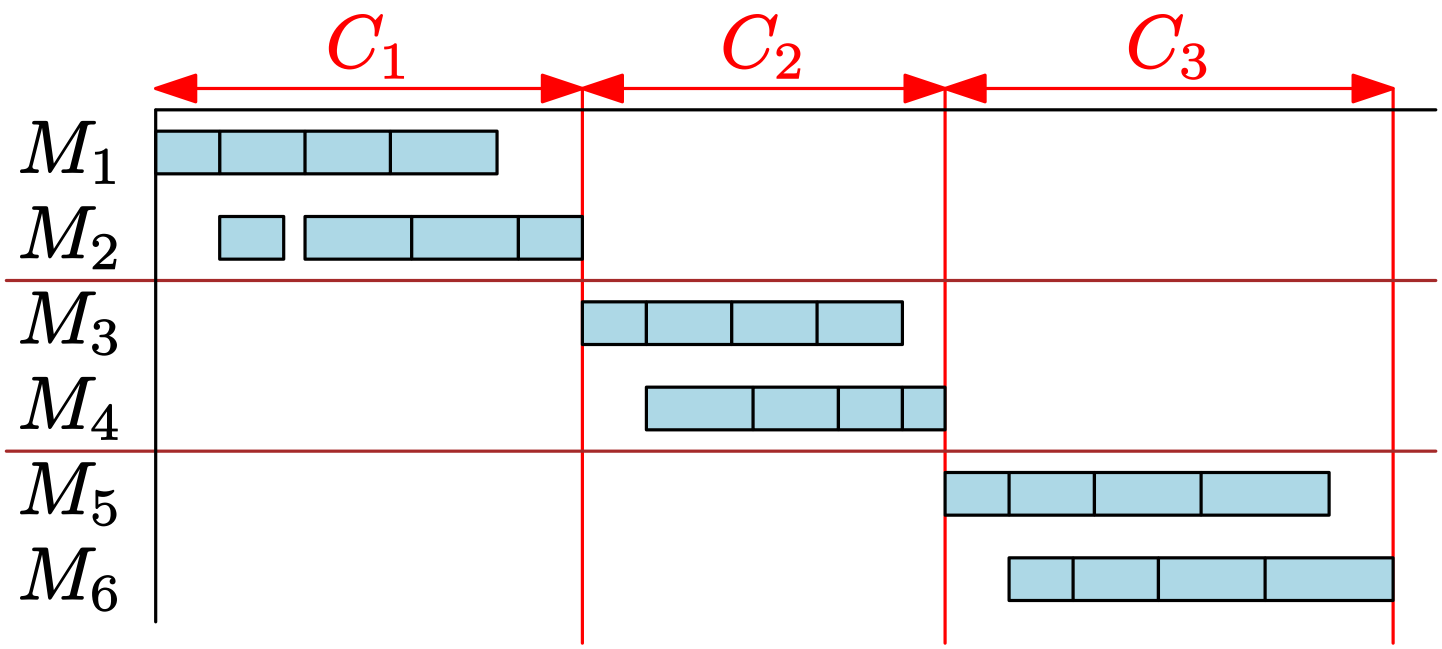


アルゴリズムの出力

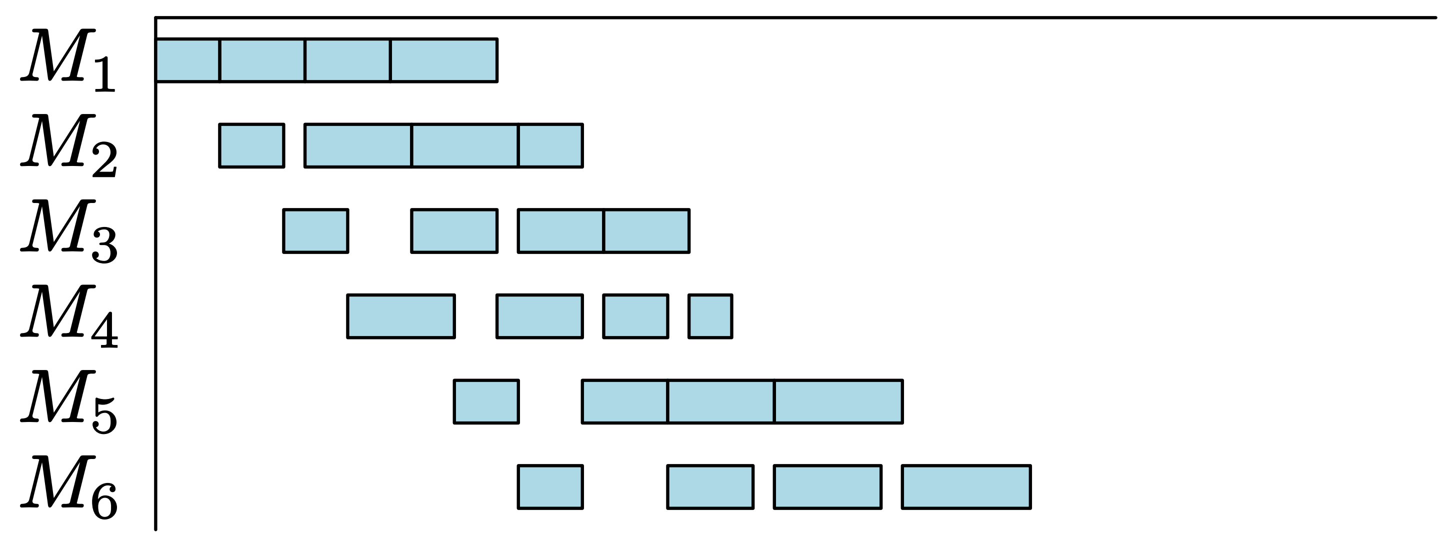


$F \mid\mid C_{\max}$ ($\lceil m/2 \rceil$ 近似)：最適値の下界

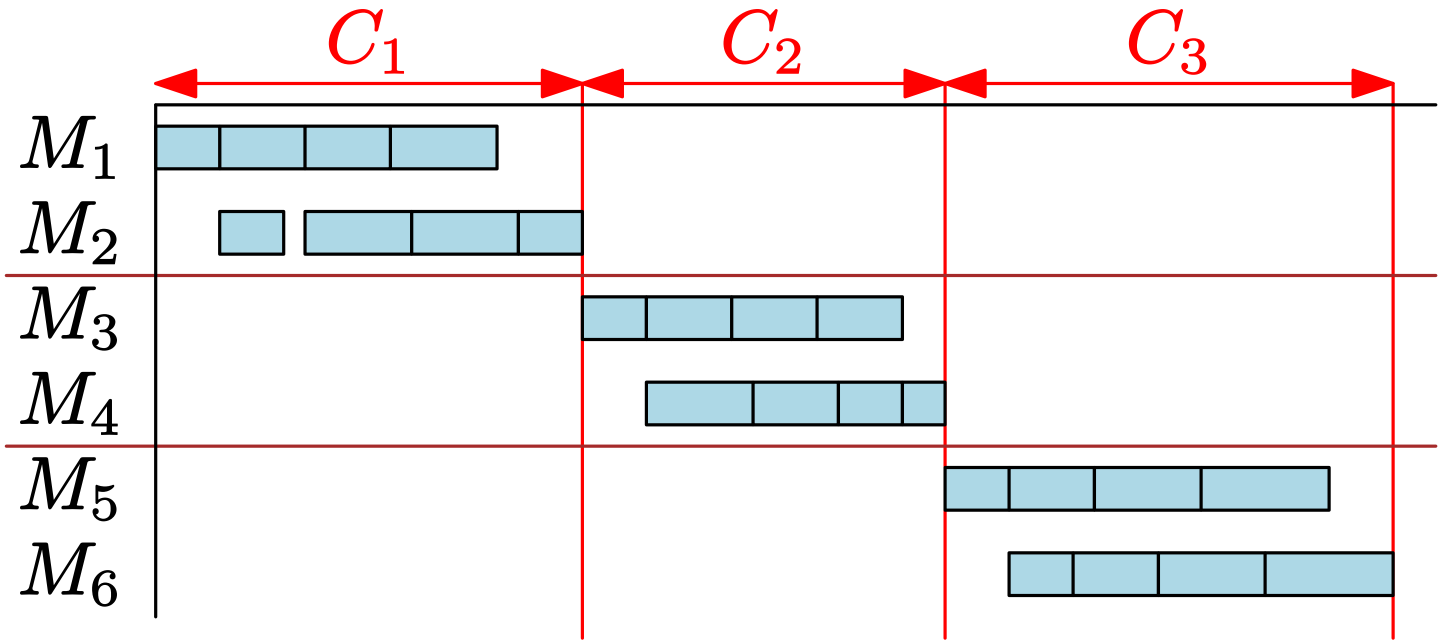
アルゴリズムの出力



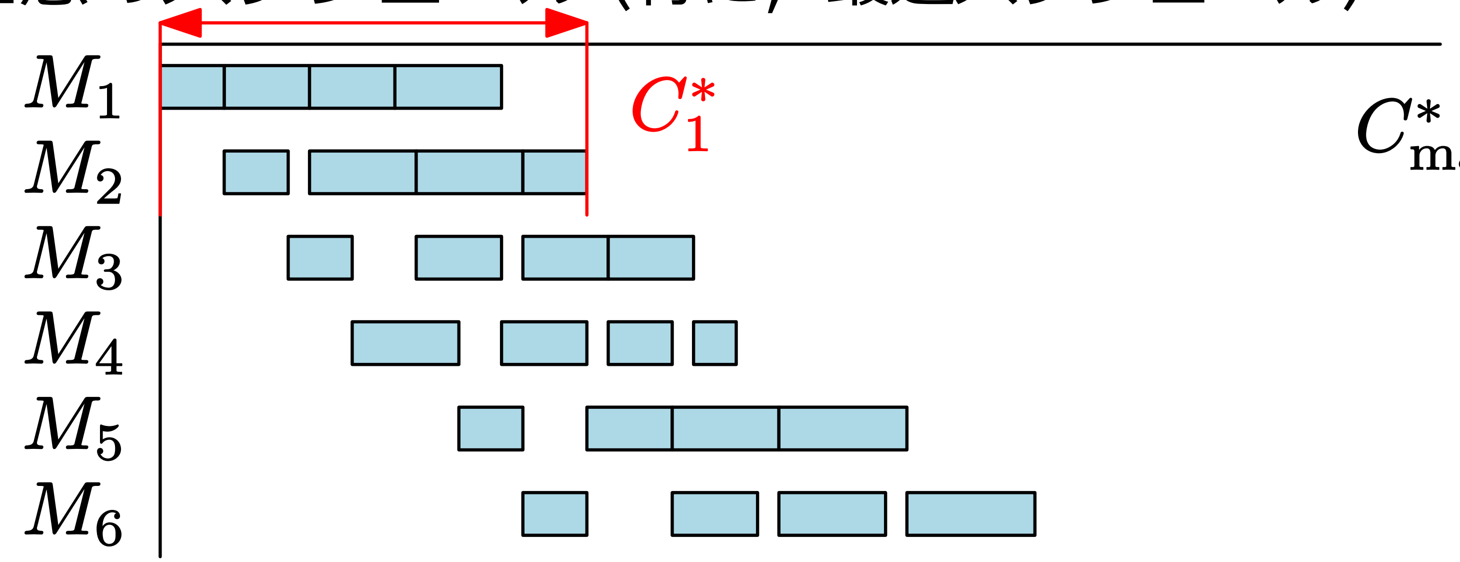
任意のスケジュール (特に，最適スケジュール)



アルゴリズムの出力



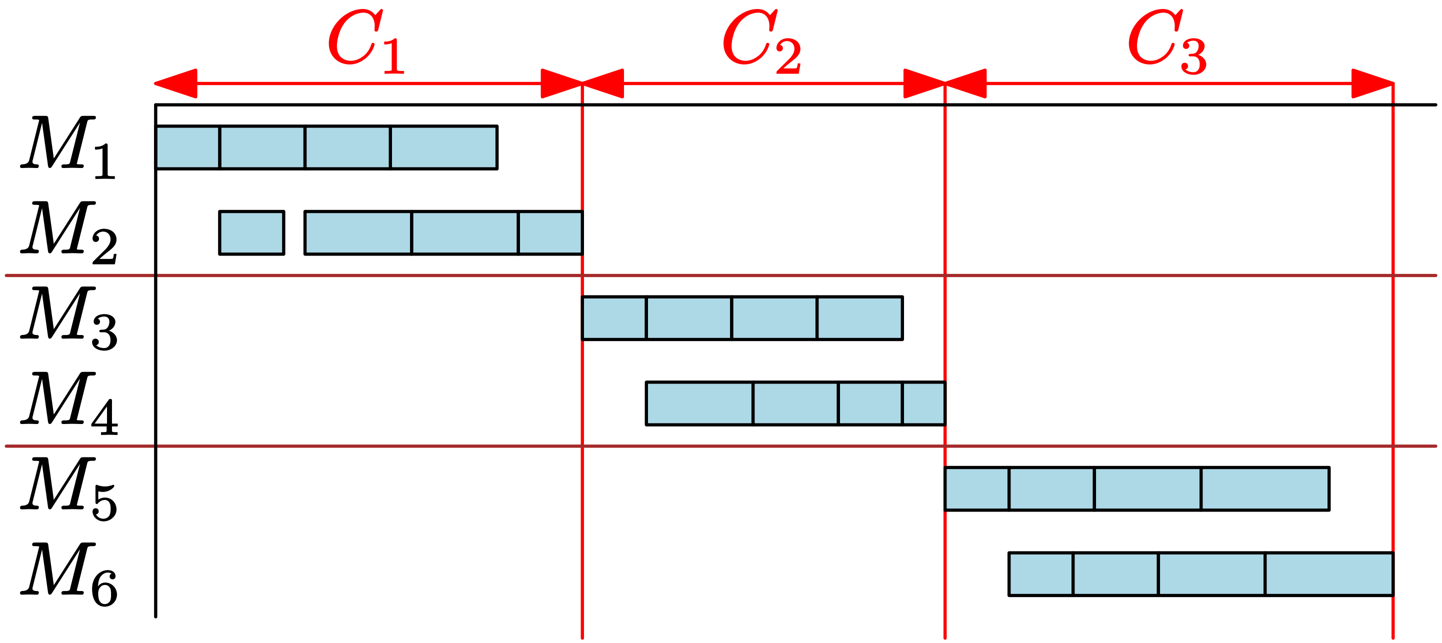
任意のスケジュール (特に，最適スケジュール)



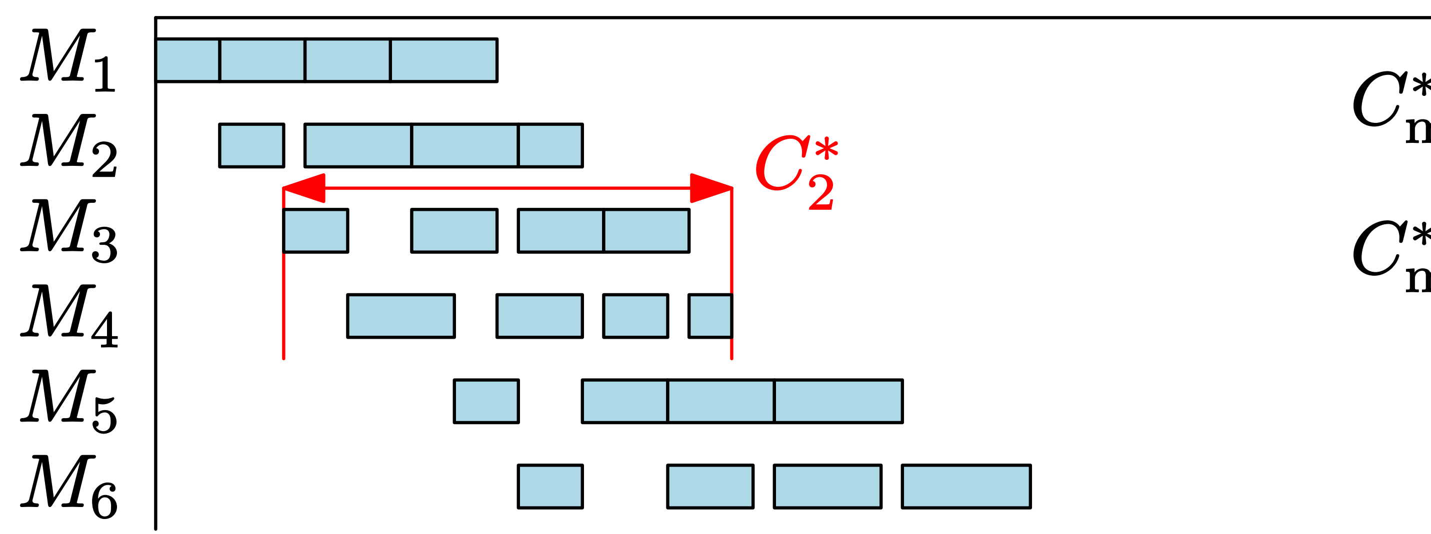
$C^*_{\max} \geq C^*_1 \geq C_1$

$F \parallel C_{\max}$ ($\lceil m/2 \rceil$ 近似)：最適値の下界

アルゴリズムの出力



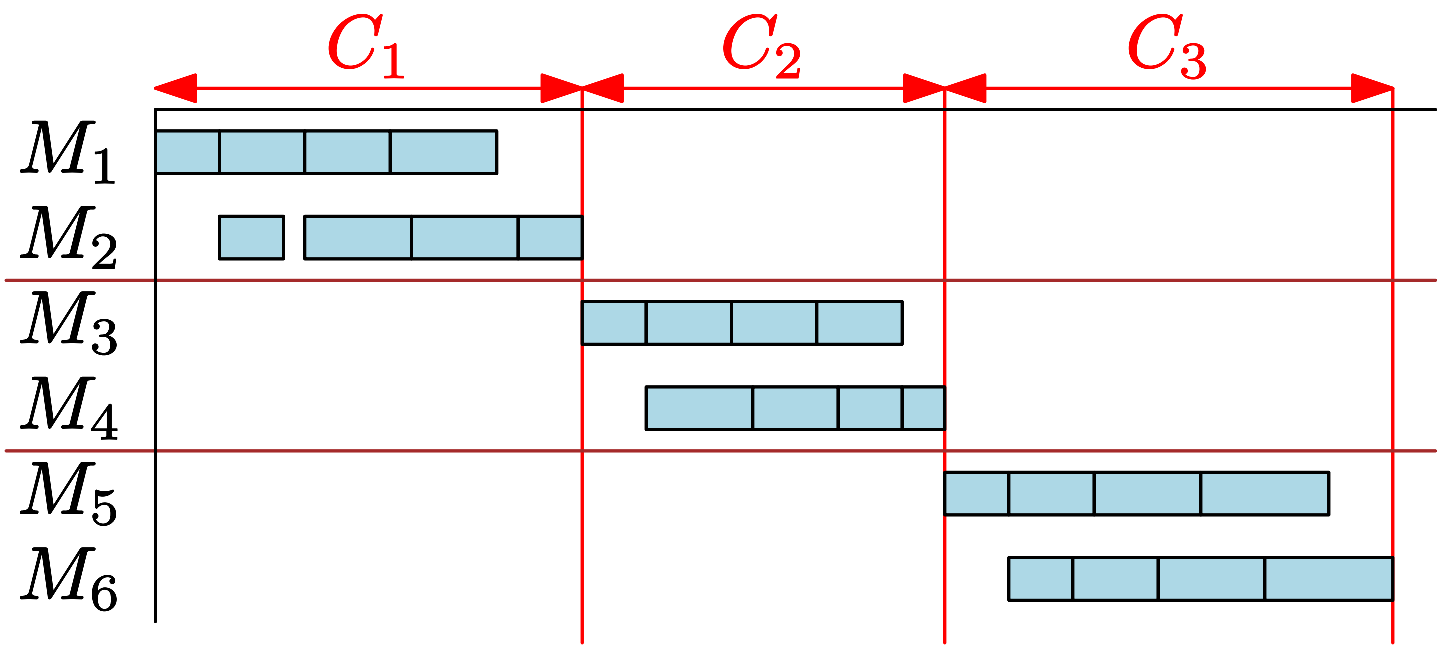
任意のスケジュール (特に，最適スケジュール)



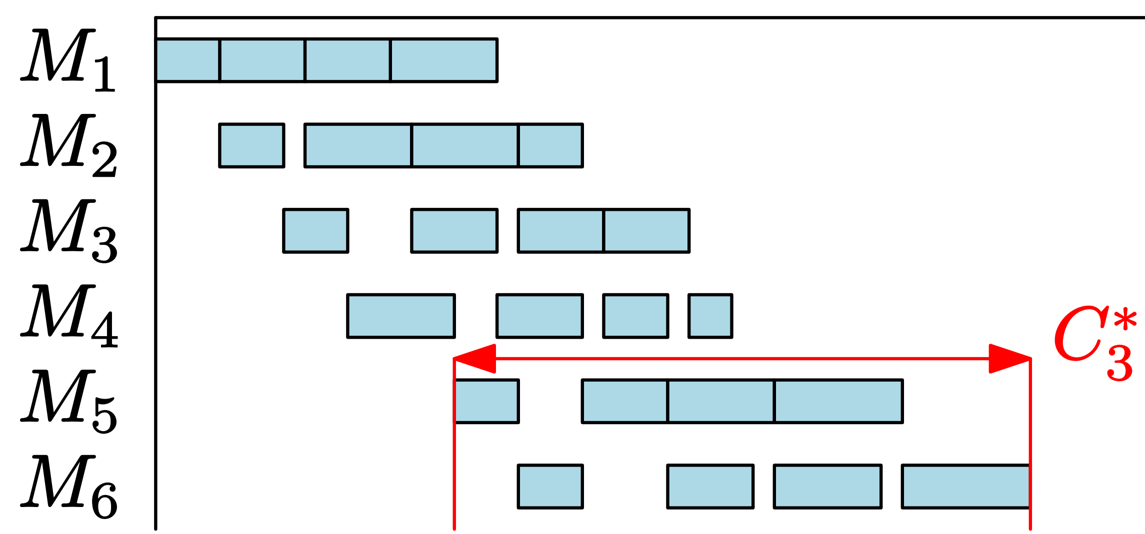
$$C^*_{\max} \geq C^*_1 \geq C_1$$

$$C^*_{\max} \geq C^*_2 \geq C_2$$

アルゴリズムの出力



任意のスケジュール (特に，最適スケジュール)

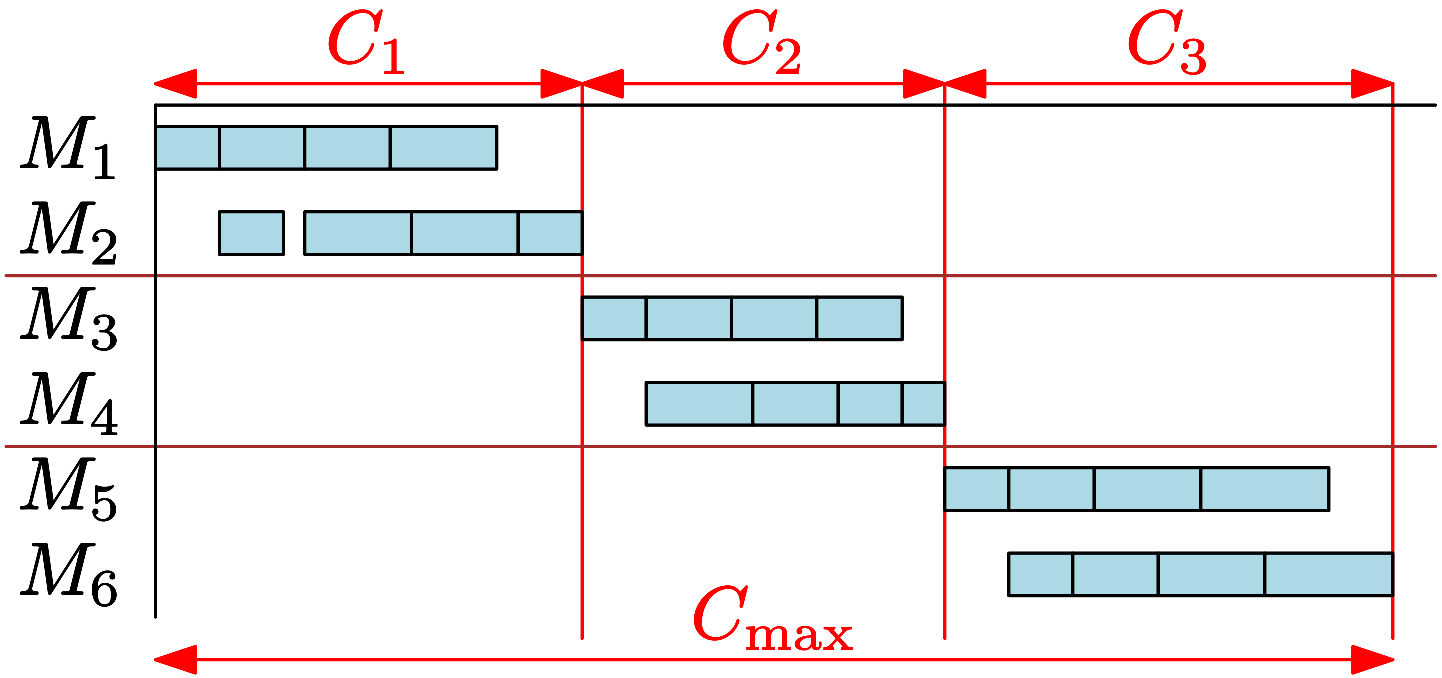


$$C^*_{\max} \geq C^*_1 \geq C_1$$

$$C^*_{\max} \geq C^*_2 \geq C_2$$

$$C^*_{\max} \geq C^*_3 \geq C_3$$

アルゴリズムの出力

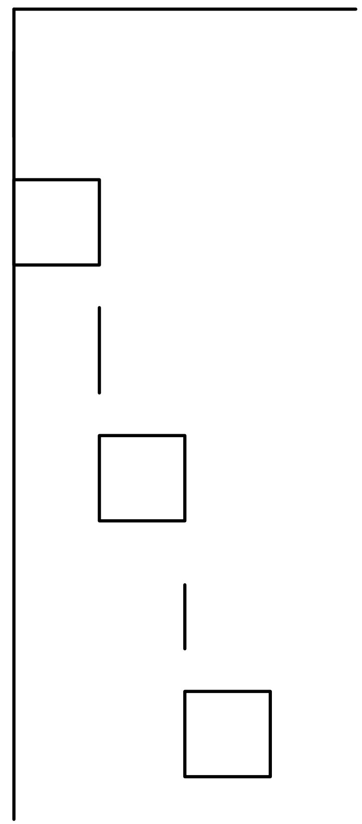


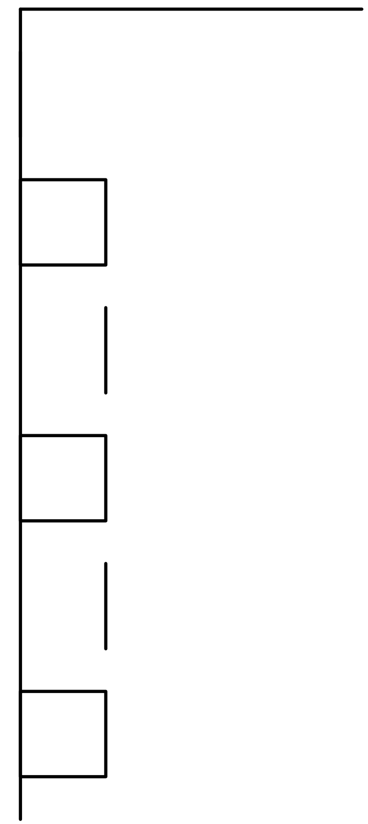
$$C_{\max} = C_1 + C_2 + \cdots + C_{\lceil m/2 \rceil} \leq \left\lceil \frac{m}{2} \right\rceil \cdot C^*_{\max} \qquad \square$$

$F \parallel C_{\max}$ ($\lceil m/2 \rceil$ 近似)：タイトな例 1

機械数 m が偶数のとき

$$
\begin{array}{c|ccc|}
 & J_1 & J_2 & J_3 \\
\hline
M_1 & 0 & 0 & 0 \\
M_2 & 1 & 0 & 0 \\
M_3 & 0 & 0 & 0 \\
M_4 & 0 & 1 & 0 \\
M_5 & 0 & 0 & 0 \\
M_6 & 0 & 0 & 1 \\
\hline
\end{array}
$$

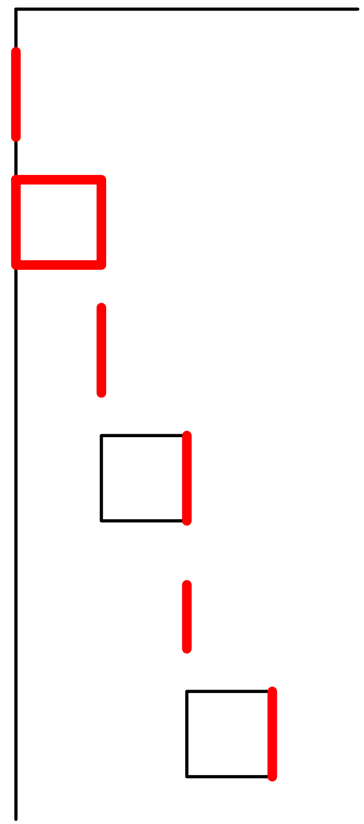


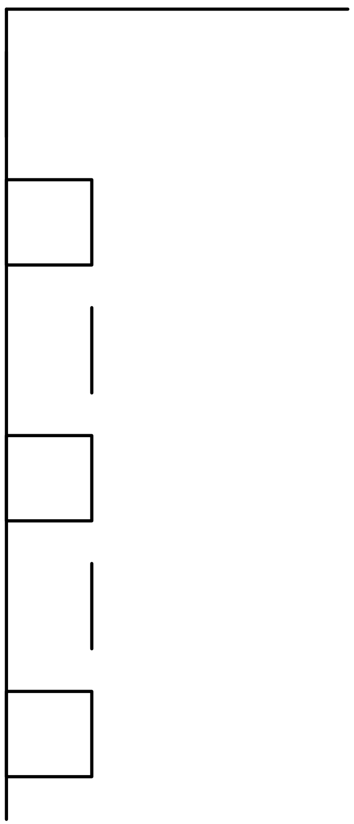


$F \parallel C_{\max}$ ($\lceil m/2 \rceil$ 近似)：タイトな例 1

機械数 m が偶数のとき

$$
\begin{array}{c|ccc|}
 & J_1 & J_2 & J_3 \\
\hline
M_1 & 0 & 0 & 0 \\
M_2 & 1 & 0 & 0 \\
M_3 & 0 & 0 & 0 \\
M_4 & 0 & 1 & 0 \\
M_5 & 0 & 0 & 0 \\
M_6 & 0 & 0 & 1 \\
\hline
\end{array}
$$

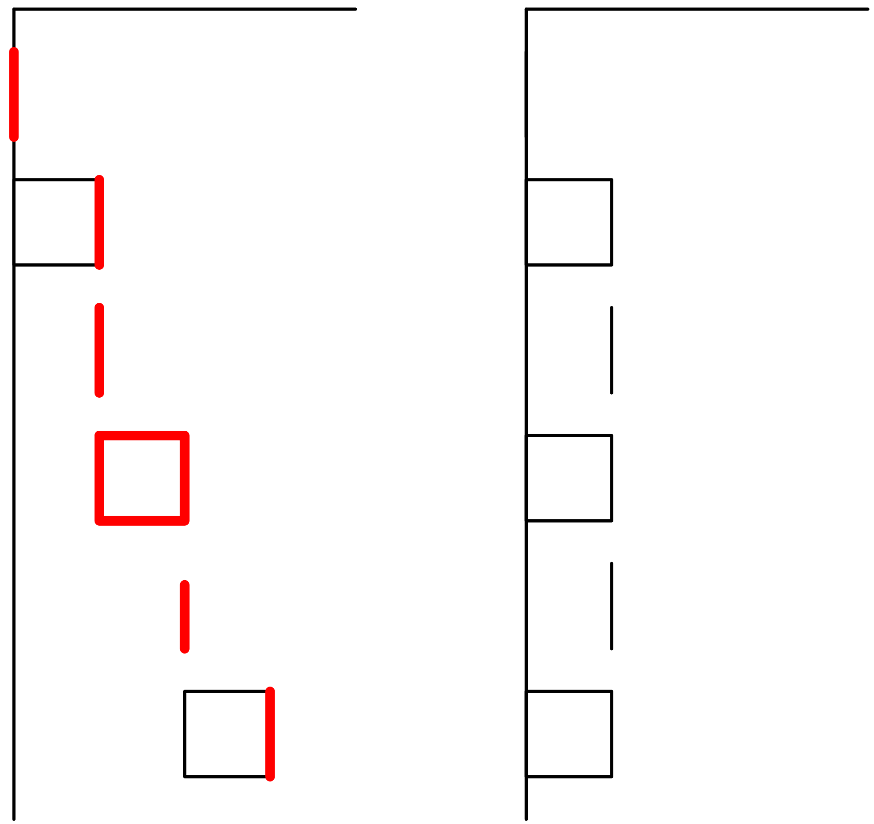




$F \mid\mid C_{\max}$ ($\lceil m/2 \rceil$ 近似)：タイトな例 1

機械数 m が偶数のとき

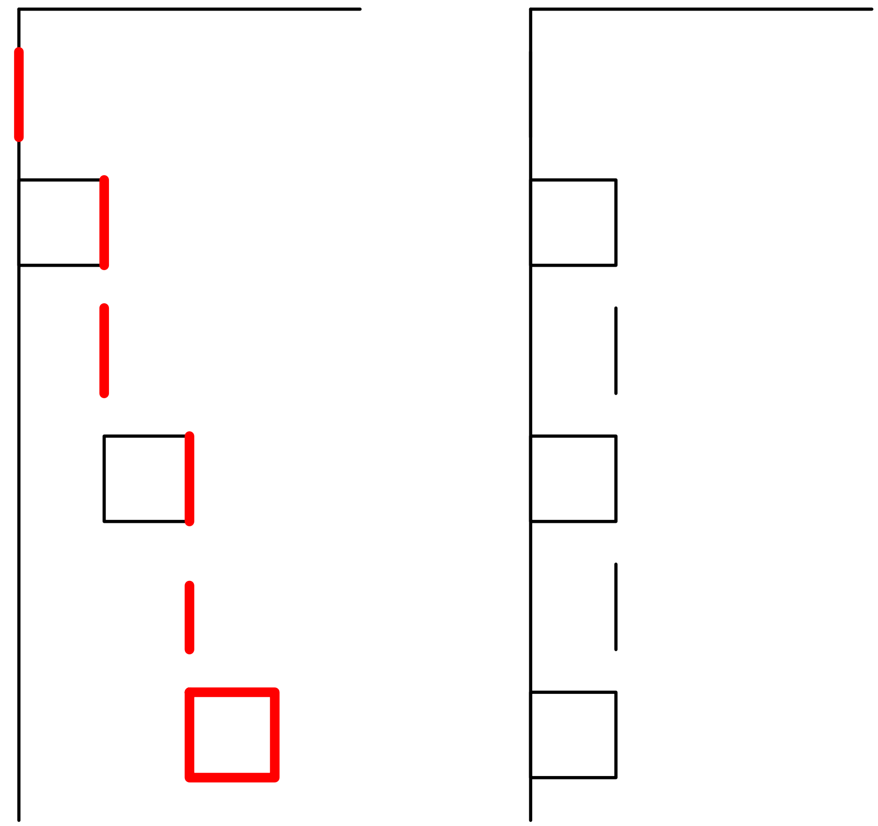
	J_1	J_2	J_3
M_1	0	0	0
M_2	1	0	0
M_3	0	0	0
M_4	0	1	0
M_5	0	0	0
M_6	0	0	1



$F \mid\mid C_{\max}$ ($\lceil m/2 \rceil$ 近似)：タイトな例 1

機械数 m が偶数のとき

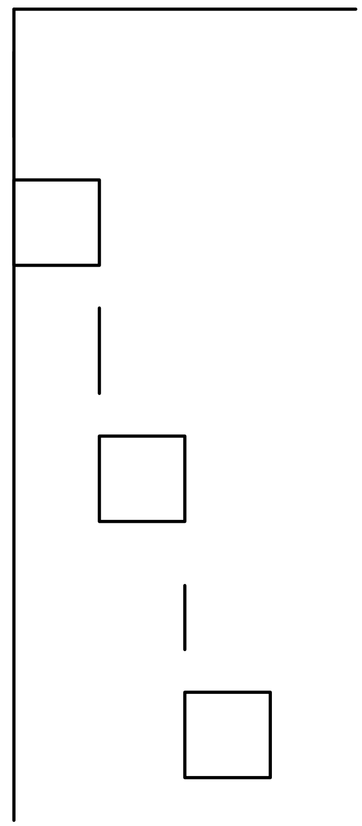
	J_1	J_2	J_3
M_1	0	0	0
M_2	1	0	0
M_3	0	0	0
M_4	0	1	0
M_5	0	0	0
M_6	0	0	1

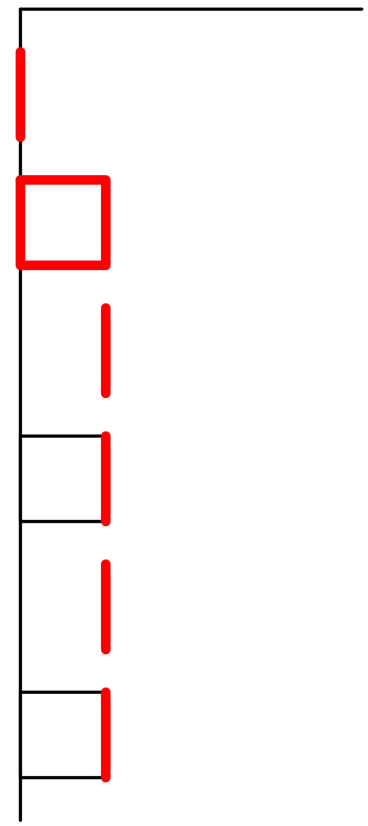


$F \mid\mid C_{\max}$ ($\lceil m/2 \rceil$ 近似)：タイトな例 1

機械数 m が偶数のとき

	J_1	J_2	J_3
M_1	0	0	0
M_2	1	0	0
M_3	0	0	0
M_4	0	1	0
M_5	0	0	0
M_6	0	0	1

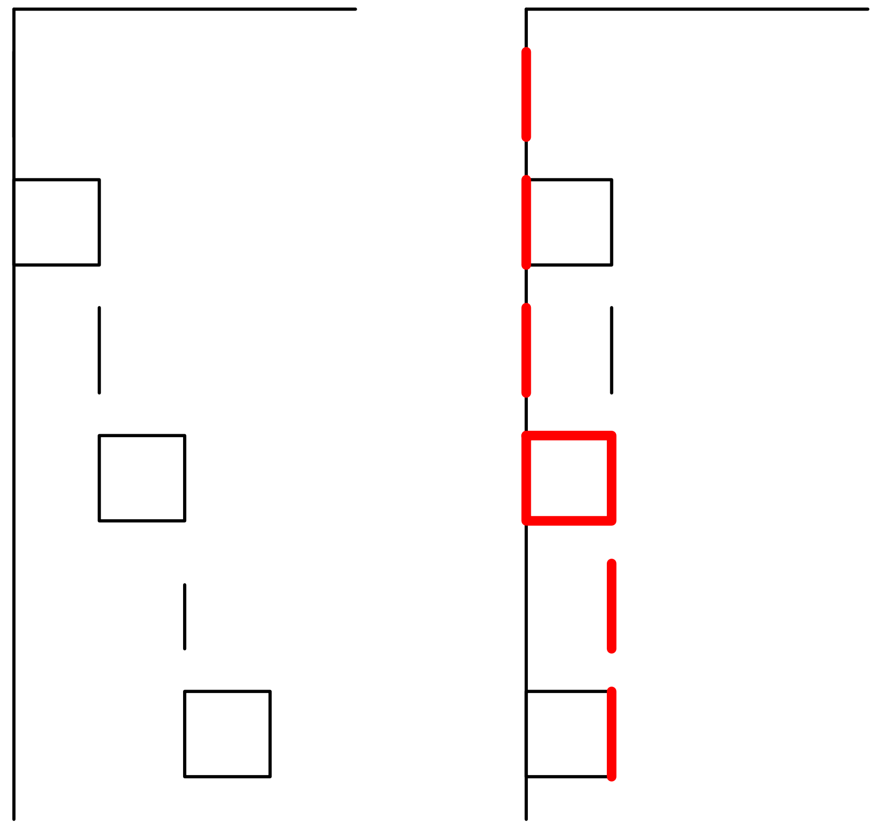




$F \parallel C_{\max}$ ($\lceil m/2 \rceil$ 近似)：タイトな例 1

機械数 m が偶数のとき

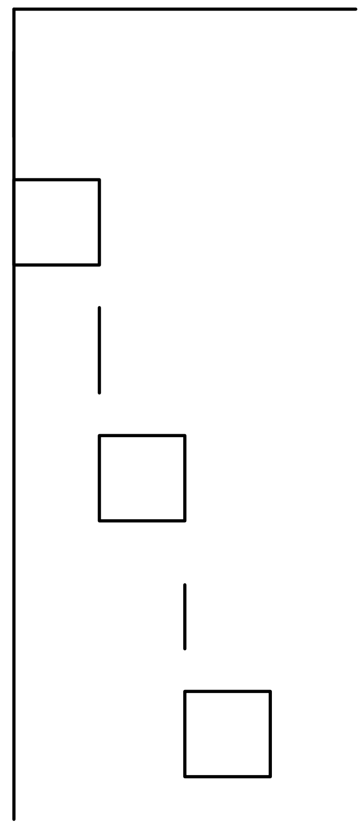
	J_1	J_2	J_3
M_1	0	0	0
M_2	1	0	0
M_3	0	0	0
M_4	0	1	0
M_5	0	0	0
M_6	0	0	1

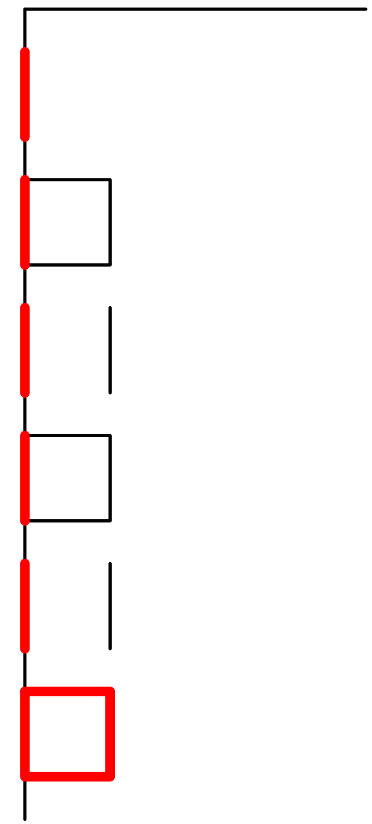


$F \mid\mid C_{\max}$ ($\lceil m/2 \rceil$ 近似)：タイトな例 1

機械数 m が偶数のとき

	J_1	J_2	J_3
M_1	0	0	0
M_2	1	0	0
M_3	0	0	0
M_4	0	1	0
M_5	0	0	0
M_6	0	0	1

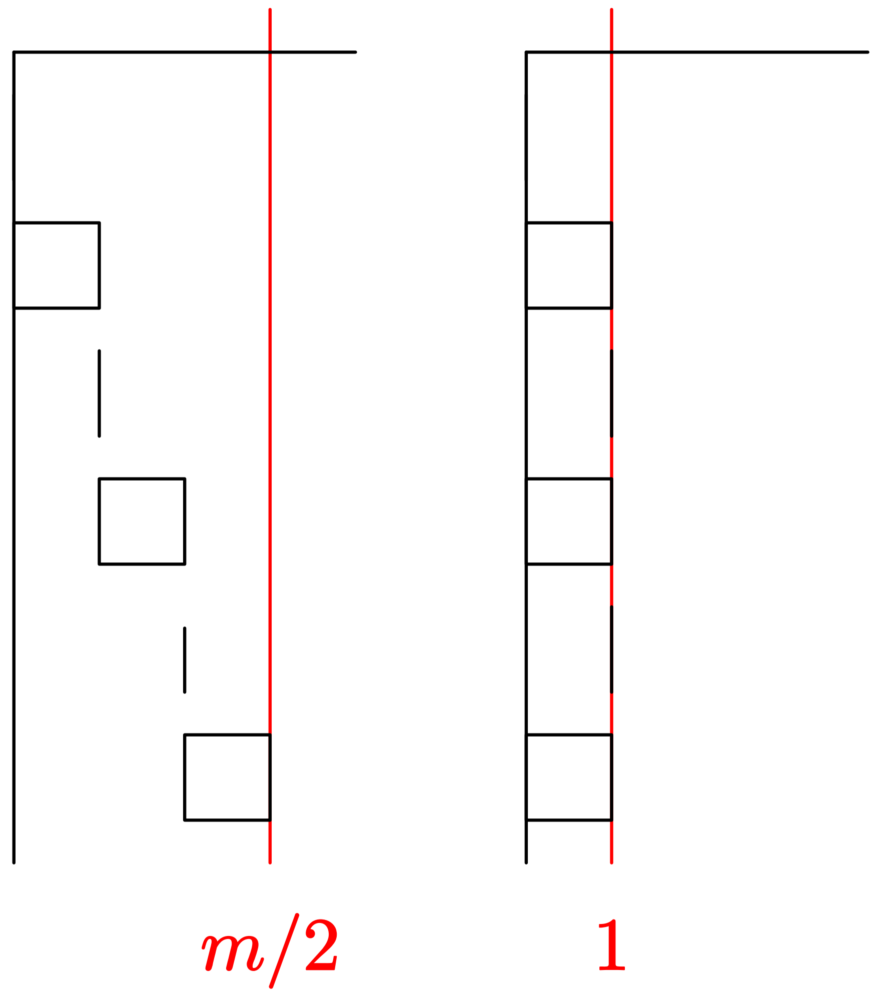




$F \,\|\, C_{\max}$ ($\lceil m/2 \rceil$ 近似)：タイトな例 1

機械数 m が偶数のとき

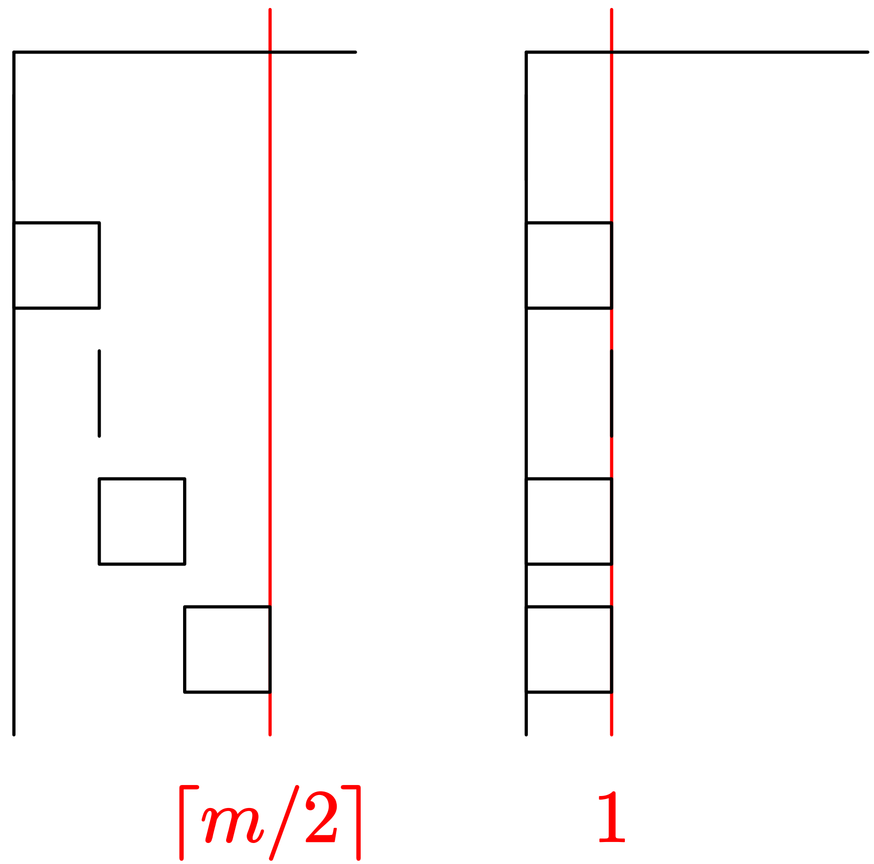
$$
\begin{array}{c|ccc|}
 & J_1 & J_2 & J_3 \\
\hline
M_1 & 0 & 0 & 0 \\
M_2 & 1 & 0 & 0 \\
M_3 & 0 & 0 & 0 \\
M_4 & 0 & 1 & 0 \\
M_5 & 0 & 0 & 0 \\
M_6 & 0 & 0 & 1 \\
\hline
\end{array}
$$



$F \mid\mid C_{\max}$ ($\lceil m/2 \rceil$ 近似)：タイトな例 2

機械数 m が奇数のとき

$$
\begin{array}{c|ccc|}
 & J_1 & J_2 & J_3 \\
\hline
M_1 & 0 & 0 & 0 \\
M_2 & 1 & 0 & 0 \\
M_3 & 0 & 0 & 0 \\
M_4 & 0 & 1 & 0 \\
M_5 & 0 & 0 & 1 \\
\hline
\end{array}
$$



本日の内容

1. $F \mid\mid C_{\max}$ の近似アルゴリズム
2. $O \mid\mid C_{\max}$ **の近似アルゴリズム**
3. 近似可能性と近似不可能性

- I. Bárány, T. Fiala, Többgépes ütemizési problémák közel optimális megoldása. *Szigma*, 15 (1982) 177–191.
- J. Wein, Algorithms for scheduling and network problems. Ph.D. Thesis, Massachusetts Institute of Technology, 1991.
- R. Chen, W. Huang, Z. Men, G. Tang, Open-shop dense schedules: Properties and worst-case performance ratio. *Journal of Scheduling* 15 (2012) 3–11.

リスト・スケジューリング (任意リスト)

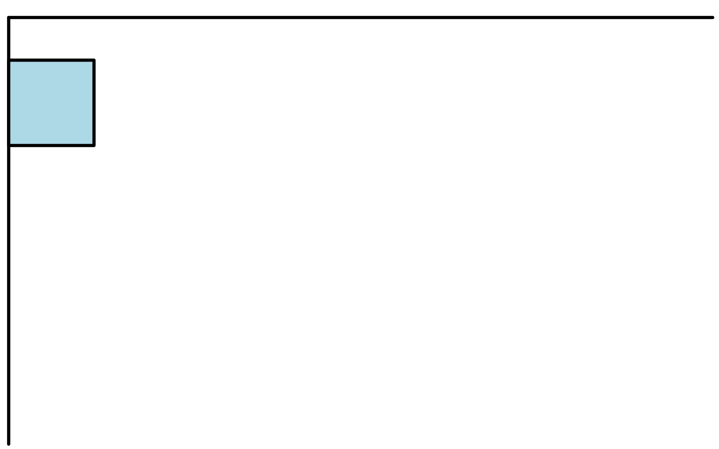
1. ジョブを任意の順に並べる
2. 各機械で，リストの先頭から工程を見ていき
   その時刻に処理できる工程があれば，それを処理する

	J_1	J_2	J_3	J_4
M_1	2	5	2	3
M_2	3	3	4	4
M_3	4	5	1	2

$O \,||\, C_{\max}$：近似アルゴリズム

リスト・スケジューリング (任意リスト)

1. ジョブを任意の順に並べる
2. 各機械で，リストの先頭から工程を見ていき
   その時刻に処理できる工程があれば，それを処理する

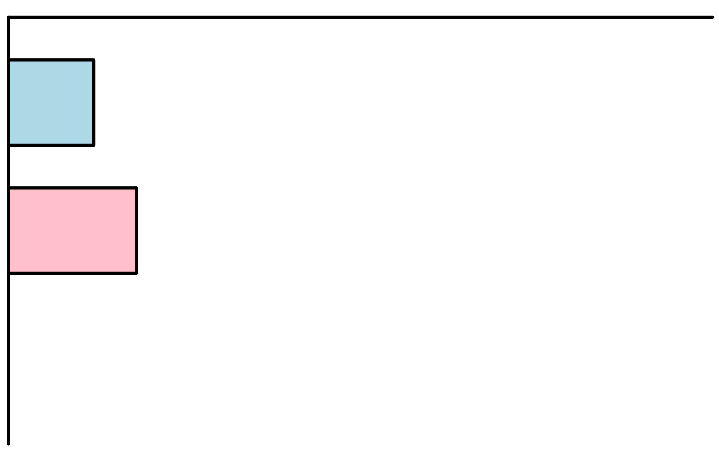


	J_1	J_2	J_3	J_4
M_1	2	5	2	3
M_2	3	3	4	4
M_3	4	5	1	2

$O \mid\mid C_{\max}$：近似アルゴリズム

リスト・スケジューリング (任意リスト)

1. ジョブを任意の順に並べる
2. 各機械で，リストの先頭から工程を見ていき
   その時刻に処理できる工程があれば，それを処理する

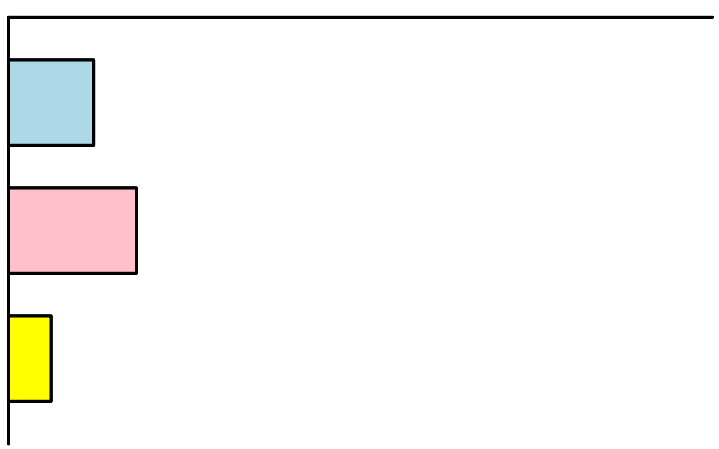


	J_1	J_2	J_3	J_4
M_1	2	5	2	3
M_2	3	3	4	4
M_3	4	5	1	2

$O \mid\mid C_{\max}$ ：近似アルゴリズム

リスト・スケジューリング (任意リスト)

1. ジョブを任意の順に並べる
2. 各機械で，リストの先頭から工程を見ていき
   その時刻に処理できる工程があれば，それを処理する



	J_1	J_2	J_3	J_4
M_1	2	5	2	3
M_2	3	3	4	4
M_3	4	5	1	2

$O \mid\mid C_{\max}$：近似アルゴリズム

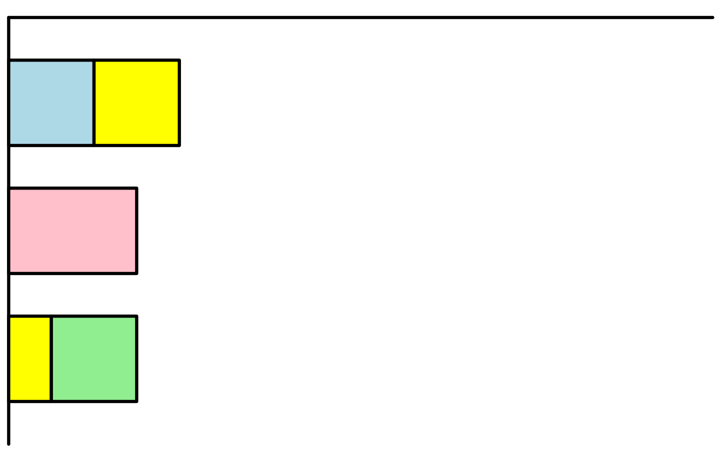
リスト・スケジューリング (任意リスト)

1. ジョブを任意の順に並べる
2. 各機械で，リストの先頭から工程を見ていき
   その時刻に処理できる工程があれば，それを処理する

	J_1	J_2	J_3	J_4
M_1	2	5	2	3
M_2	3	3	4	4
M_3	4	5	1	2

リスト・スケジューリング (任意リスト)

1. ジョブを任意の順に並べる
2. 各機械で，リストの先頭から工程を見ていき
   その時刻に処理できる工程があれば，それを処理する

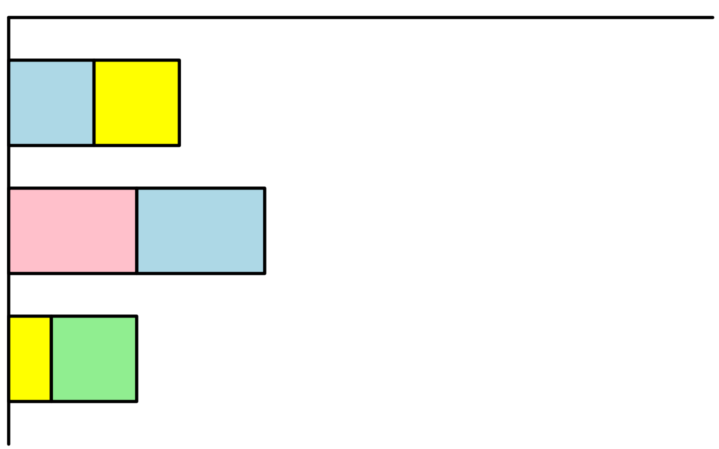


	J_1	J_2	J_3	J_4
M_1	2	5	2	3
M_2	3	3	4	4
M_3	4	5	1	2

$O \mid\mid C_{\max}$：近似アルゴリズム

リスト・スケジューリング (任意リスト)

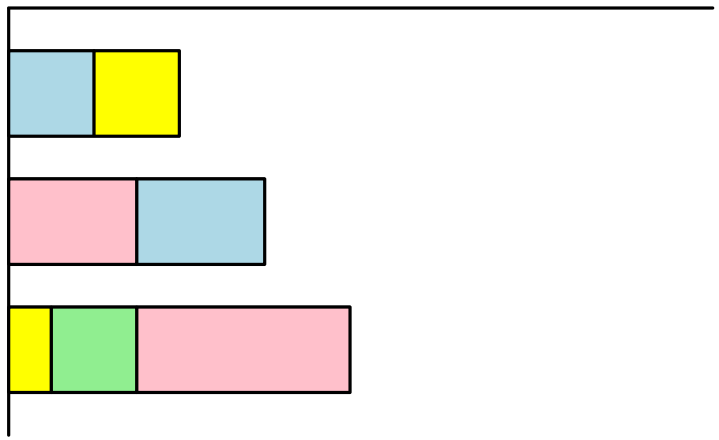
1. ジョブを任意の順に並べる
2. 各機械で，リストの先頭から工程を見ていき
   その時刻に処理できる工程があれば，それを処理する



	J_1	J_2	J_3	J_4
M_1	2	5	2	3
M_2	3	3	4	4
M_3	4	5	1	2

リスト・スケジューリング (任意リスト)

1. ジョブを任意の順に並べる
2. 各機械で，リストの先頭から工程を見ていき
   その時刻に処理できる工程があれば，それを処理する

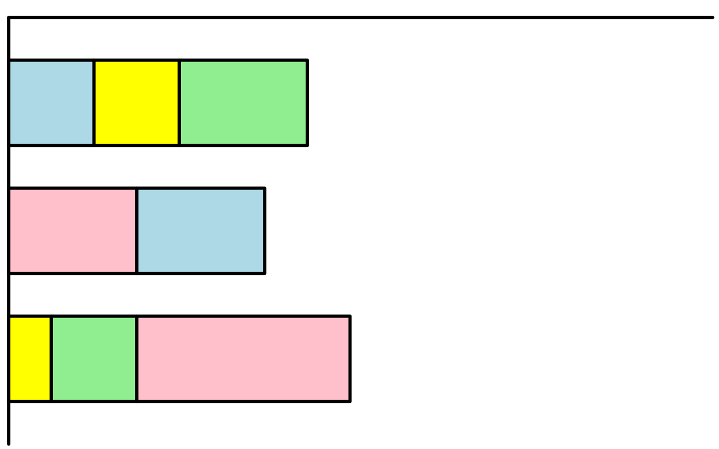


	J_1	J_2	J_3	J_4
M_1	2	5	2	3
M_2	3	3	4	4
M_3	4	5	1	2

$O \| C_{\max}$：近似アルゴリズム

リスト・スケジューリング (任意リスト)

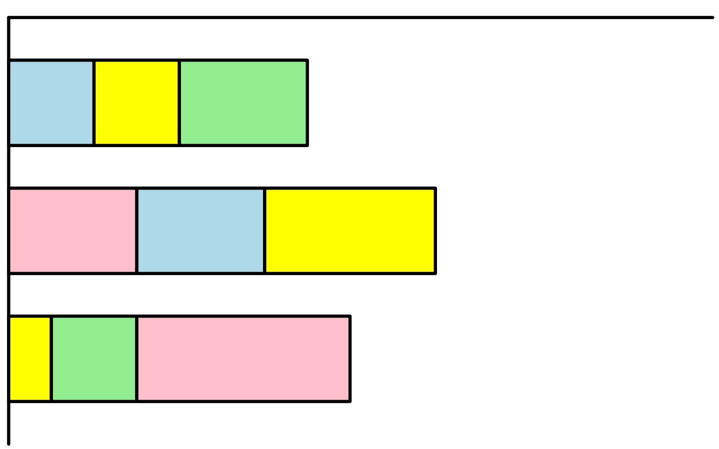
1. ジョブを任意の順に並べる
2. 各機械で，リストの先頭から工程を見ていき
   その時刻に処理できる工程があれば，それを処理する



	J_1	J_2	J_3	J_4
M_1	2	5	2	3
M_2	3	3	4	4
M_3	4	5	1	2

リスト・スケジューリング (任意リスト)

1. ジョブを任意の順に並べる
2. 各機械で，リストの先頭から工程を見ていき
   その時刻に処理できる工程があれば，それを処理する

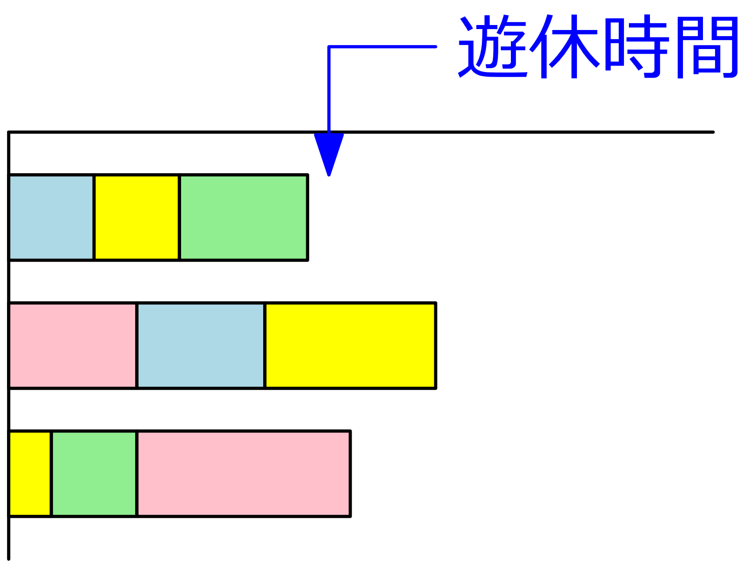


	J_1	J_2	J_3	J_4
M_1	2	5	2	3
M_2	3	3	4	4
M_3	4	5	1	2

$O \mid\mid C_{\max}$：近似アルゴリズム

リスト・スケジューリング (任意リスト)

1. ジョブを任意の順に並べる
2. 各機械で，リストの先頭から工程を見ていき
   その時刻に処理できる工程があれば，それを処理する

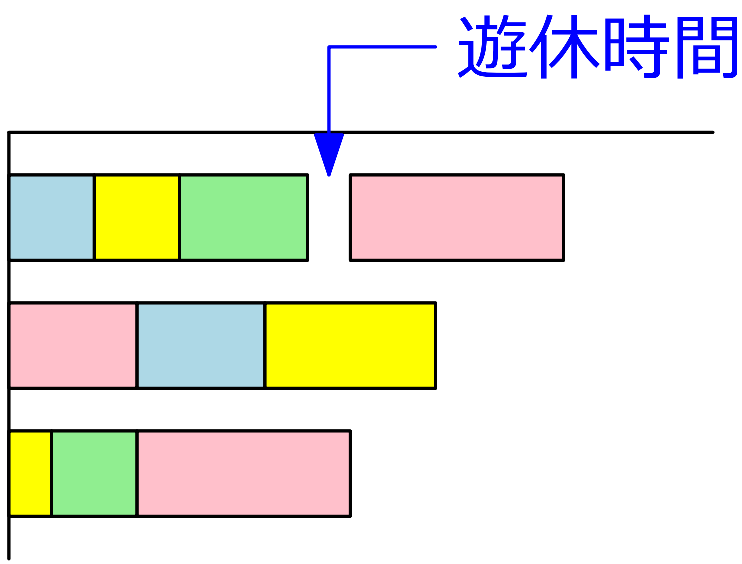


	J_1	J_2	J_3	J_4
M_1	2	5	2	3
M_2	3	3	4	4
M_3	4	5	1	2

$O \parallel C_{\max}$：近似アルゴリズム

リスト・スケジューリング (任意リスト)

1. ジョブを任意の順に並べる
2. 各機械で，リストの先頭から工程を見ていき
   その時刻に処理できる工程があれば，それを処理する

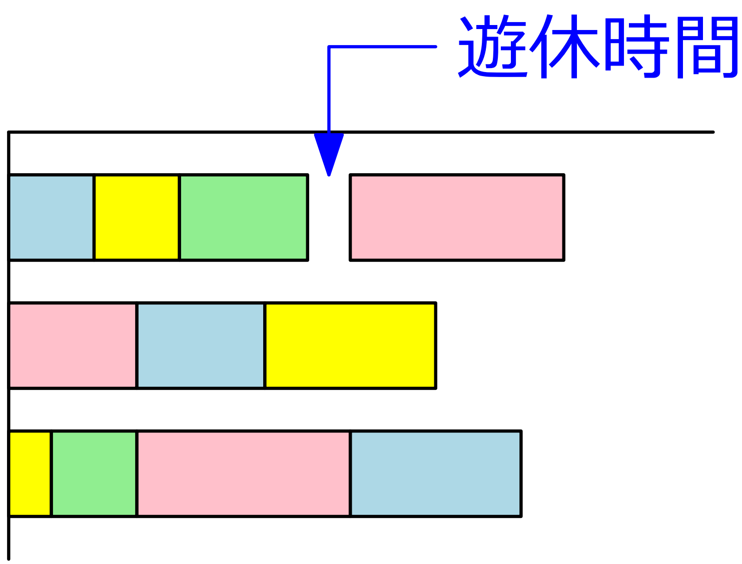


	J_1	J_2	J_3	J_4
M_1	2	5	2	3
M_2	3	3	4	4
M_3	4	5	1	2

$O \mid\mid C_{\max}$：近似アルゴリズム

リスト・スケジューリング (任意リスト)

1. ジョブを任意の順に並べる
2. 各機械で，リストの先頭から工程を見ていき
   その時刻に処理できる工程があれば，それを処理する

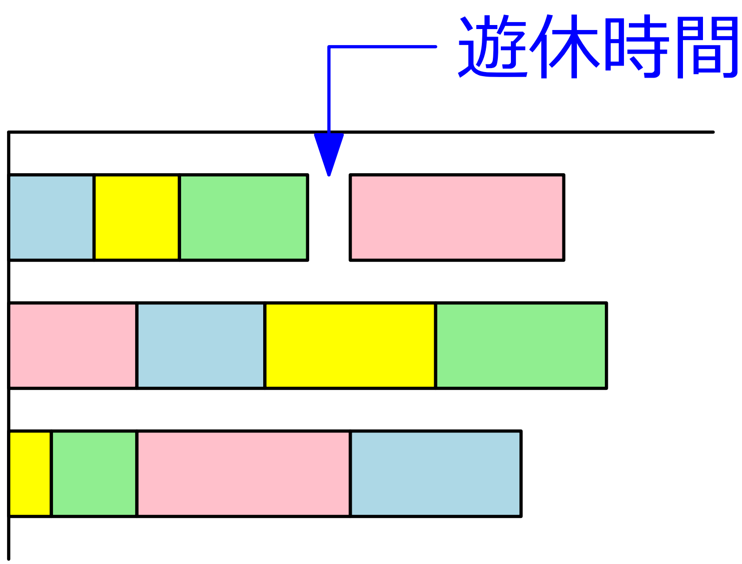


	J_1	J_2	J_3	J_4
M_1	2	5	2	3
M_2	3	3	4	4
M_3	4	5	1	2

$O \parallel C_{\max}$：近似アルゴリズム

リスト・スケジューリング (任意リスト)

1. ジョブを任意の順に並べる
2. 各機械で，リストの先頭から工程を見ていき
   その時刻に処理できる工程があれば，それを処理する



	J_1	J_2	J_3	J_4
M_1	2	5	2	3
M_2	3	3	4	4
M_3	4	5	1	2

定理 (Racsmány '82, Wein '91)

リスト・スケジューリングは 2 近似アルゴリズム

定理 (Racsmány '82, Wein '91)

リスト・スケジューリングは 2 近似アルゴリズム

予想 (未解決) (Chen, Strusevich '93)

リスト・スケジューリングは $2 - \dfrac{1}{m}$ 近似アルゴリズム

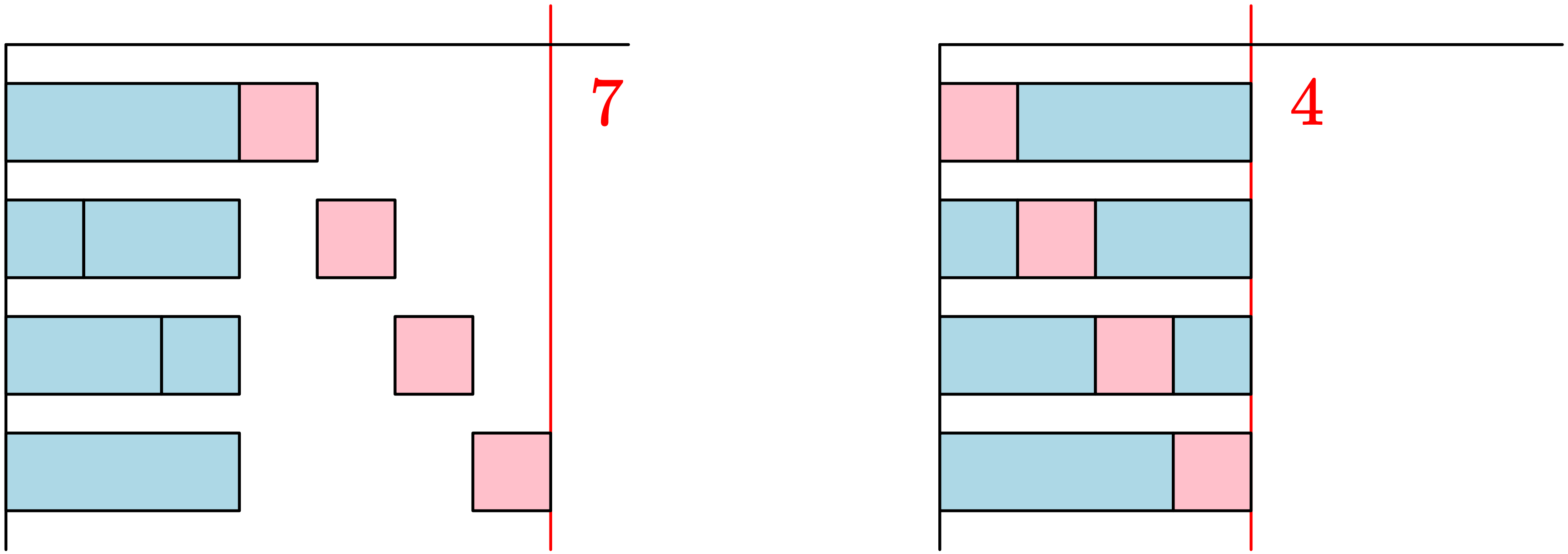
(m は機械数)

$m \leq 8$ のとき，予想は正しいと知られている

(Chen, Huang, Men, Tang '01)

近似比が $2 - \dfrac{1}{m}$ である例 ($m = 4$)

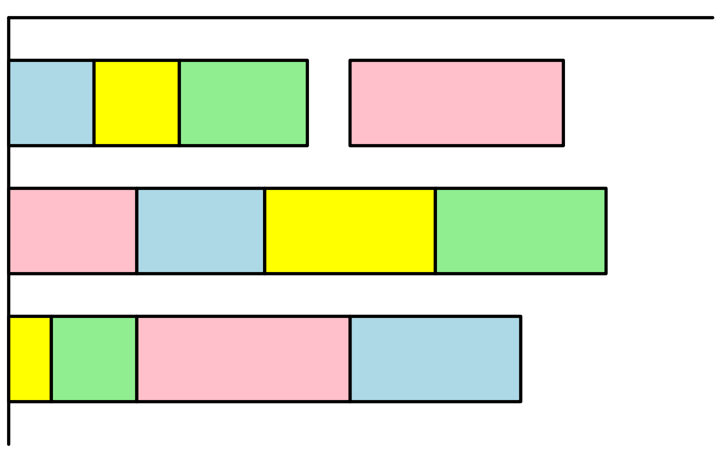
	J_1	J_2	J_3	J_4	J_5	J_6	J_7
M_1	3	0	0	0	0	0	1
M_2	0	1	2	0	0	0	1
M_3	0	0	0	2	1	0	1
M_4	0	0	0	0	0	3	1



性質

$O \mid\mid C_{\max}$ において，最適値 $C^*_{\max}$ は次をすべて満たす

- $C^*_{\max} \geq \displaystyle\sum_{j=1}^{n} p_{ij}, \quad i = 1, 2, \ldots, m$
- $C^*_{\max} \geq \displaystyle\sum_{i=1}^{m} p_{ij}, \quad j = 1, 2, \ldots, n$

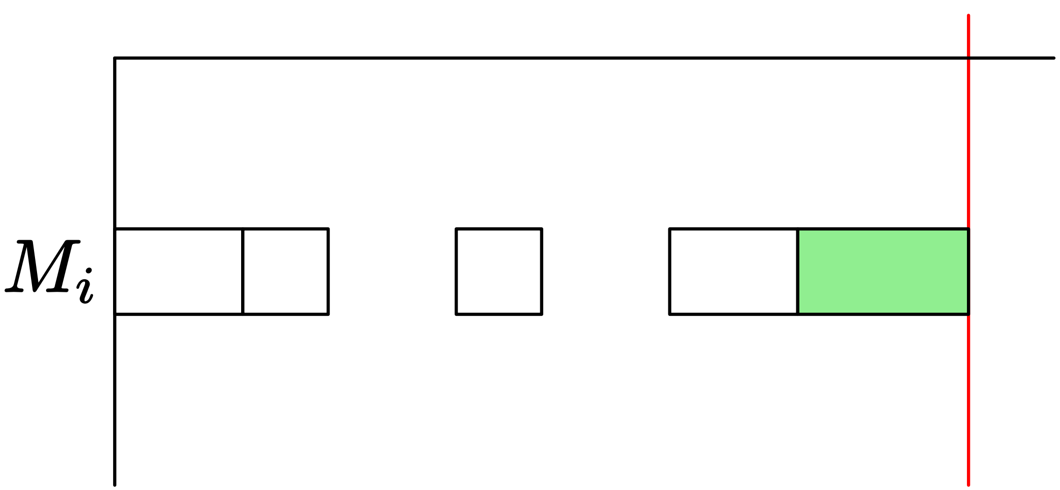


	J_1	J_2	J_3	J_4
M_1	2	5	2	3
M_2	3	3	4	4
M_3	4	5	1	2

$O \mid\mid C_{\max}$：近似比の導出

アルゴリズムの出力スケジュールを σ とする

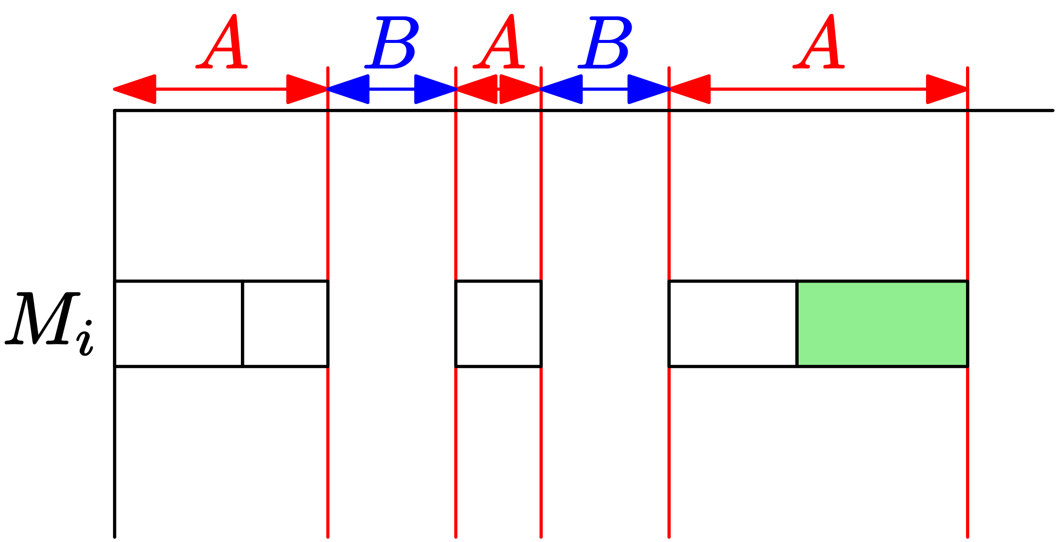
- $J_j =$ 最後に完了する工程のジョブ　とする
- $M_i = J_j$ が完了する機械　とする
- 時間 $[0, C_{\max}(\sigma)]$ を次の 2 つに分ける
  - $A = M_i$ がジョブを処理している時間
  - $B = M_i$ の遊休時間
- B では，J_j の工程がどこかで処理されている　（なぜ？）



$O \mid\mid C_{\max}$：近似比の導出

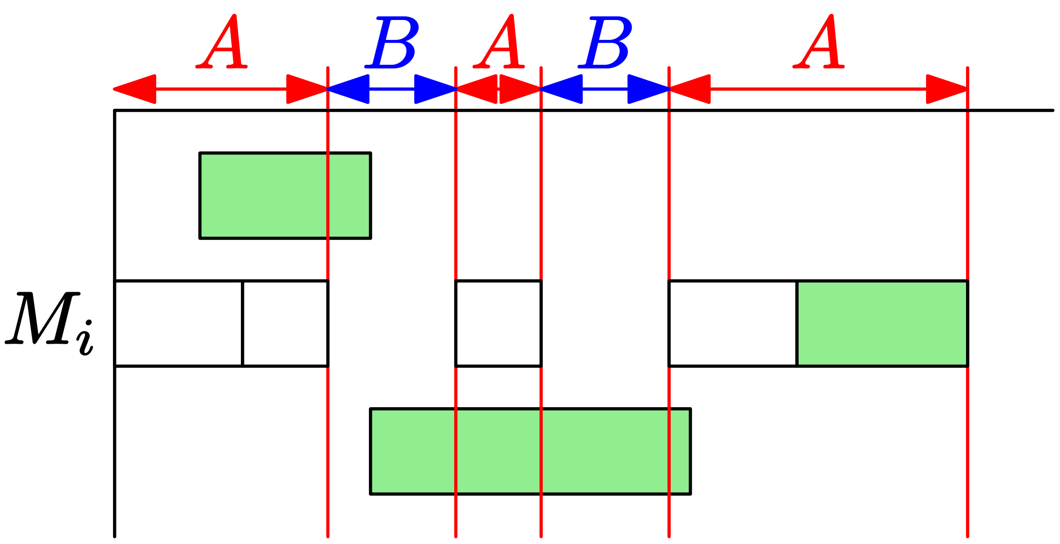
アルゴリズムの出力スケジュールを σ とする

- J_j = 最後に完了する工程のジョブ　とする
- $M_i = J_j$ が完了する機械　とする
- 時間 $[0, C_{\max}(\sigma)]$ を次の 2 つに分ける
  - $A = M_i$ がジョブを処理している時間
  - $B = M_i$ の遊休時間
- B では，J_j の工程がどこかで処理されている　（なぜ？）



アルゴリズムの出力スケジュールを σ とする

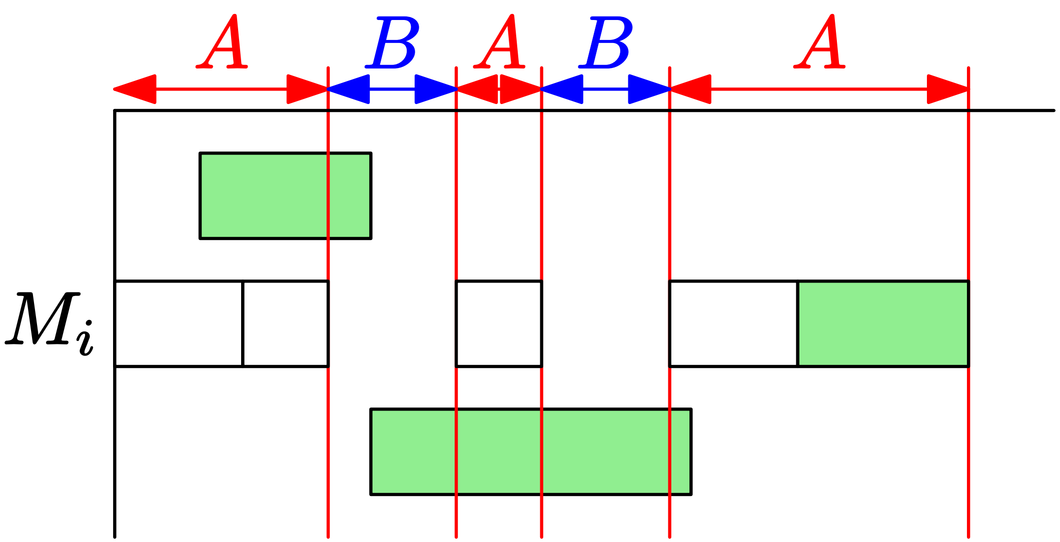
- J_j = 最後に完了する工程のジョブ　とする
- $M_i = J_j$ が完了する機械　とする
- 時間 $[0, C_{\max}(\sigma)]$ を次の 2 つに分ける
  - $A = M_i$ がジョブを処理している時間
  - $B = M_i$ の遊休時間
- B では，J_j の工程がどこかで処理されている　(なぜ？)



$O \,||\, C_{\max}$：近似比の導出

アルゴリズムの出力スケジュールを σ とする

- J_j = 最後に完了する工程のジョブ　とする
- $M_i = J_j$ が完了する機械　とする
- 時間 $[0, C_{\max}(\sigma)]$ を次の 2 つに分ける
  - $A = M_i$ がジョブを処理している時間
  - $B = M_i$ の遊休時間
- B では，J_j の工程がどこかで処理されている　（なぜ？）



したがって

$$
\begin{aligned}
& C_{\max}(\sigma) \\
& = A + B \\
& \le \sum_{j'=1}^{n} p_{ij'} + \sum_{i'=1}^{m} p_{i'j} \\
& \le 2C^*_{\max}
\end{aligned}
$$

□

本日の内容

1. $F \mid\mid C_{\max}$ の近似アルゴリズム
2. $O \mid\mid C_{\max}$ の近似アルゴリズム
3. **近似可能性と近似不可能性**

- P. Schuurman, G. J. Woeginger, Polynomial time approximation algorithms for machine scheduling: Ten open problems. *Journal of Scheduling* 2 (1999) 203–213.
- M. Mastrolilli, O. Svensson, Hardness of approximating flow and job shop scheduling problems. *Journal of the ACM* 58 (2011) Article 20, 32 pages.

ここまでのまとめ

	近似比	計算量
$\boxed{F \mid\mid C_{\max}}$		
	m	$O(mn)$
	$\lceil m/2 \rceil$	$O(mn \log n)$
$\boxed{O \mid\mid C_{\max}}$		
	2	$O(mn^2)$

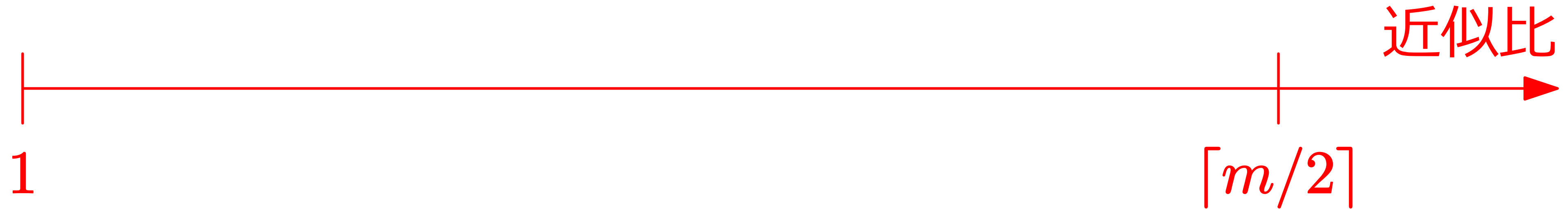
ここまでのまとめ

	近似比	計算量
$\boxed{F \mid\mid C_{\max}}$		
	m	$O(mn)$
	$\lceil m/2 \rceil$	$O(mn \log n)$
$\boxed{O \mid\mid C_{\max}}$		
	2	$O(mn^2)$

疑問

多項式時間アルゴリズムで
どこまで近似比を小さくできるだろうか？

$F \mid\mid C_{\max}$：多項式時間近似可能性

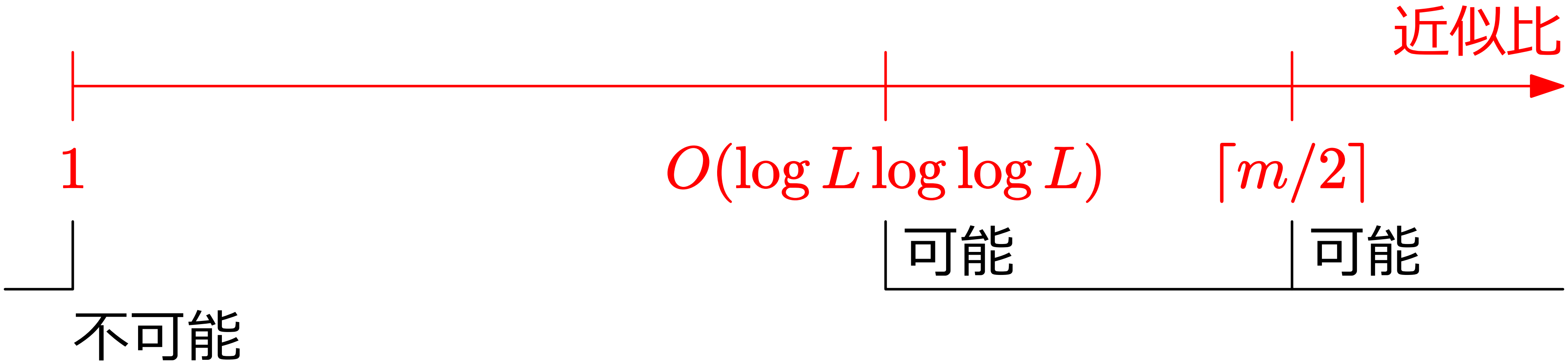


$F \mid\mid C_{\max}$：多項式時間近似可能性

P $\neq$ NP の仮定の下で

$F \mid\mid C_{\max}$：多項式時間近似可能性

P $\neq$ NP の仮定の下で



ただし， $L = \max\left\{\max_i \sum_{j=1}^{n} p_{ij}, \max_j \sum_{i=1}^{m} p_{ij}\right\}$

(Feige, Scheideler '02)

$F \mid\mid C_{\max}$：多項式時間近似可能性

P $\neq$ NP の仮定の下で

NP $\not\subseteq$ ZTIME$(2^{\log^{O(1/\varepsilon)} n})$ の仮定の下で

近似比

1 $O(\log^{1-\varepsilon} L)$ $O(\log L \log\log L)$ $\lceil m/2 \rceil$

可能 可能

不可能 不可能

$$\text{ただし，} L = \max\left\{\max_i \sum_{j=1}^{n} p_{ij}, \max_j \sum_{i=1}^{m} p_{ij}\right\}$$

(Feige, Scheideler ’02)

(Mastrolilli, Svenson ’11)

$F \mid\mid C_{\max}$：多項式時間近似可能性

P $\neq$ NP の仮定の下で

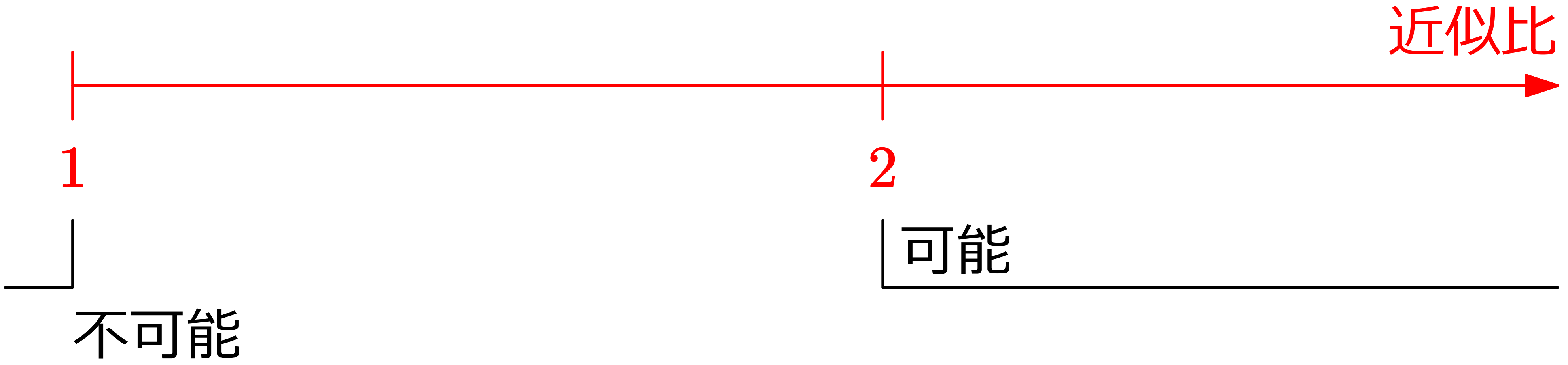
NP $\not\subseteq$ ZTIME$(2^{\log^{O(1/\varepsilon)} n})$ の仮定の下で

ただし，$L = \max\left\{ \max_i \sum_{j=1}^{n} p_{ij}, \max_j \sum_{i=1}^{m} p_{ij} \right\}$

(Feige, Scheideler '02)

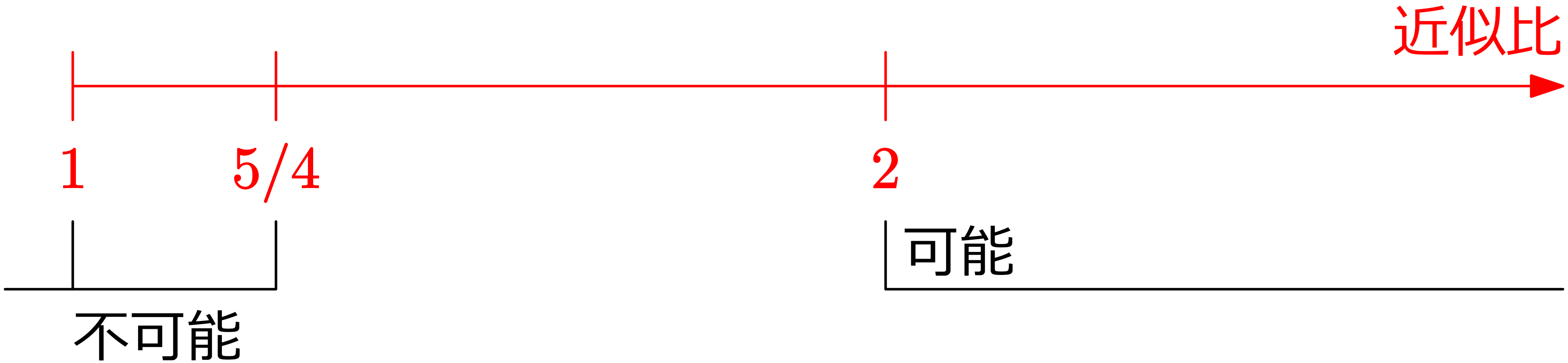
(Mastrolilli, Svenson '11)

P $\neq$ NP の仮定の下で



$O \parallel C_{\max}$：多項式時間近似可能性

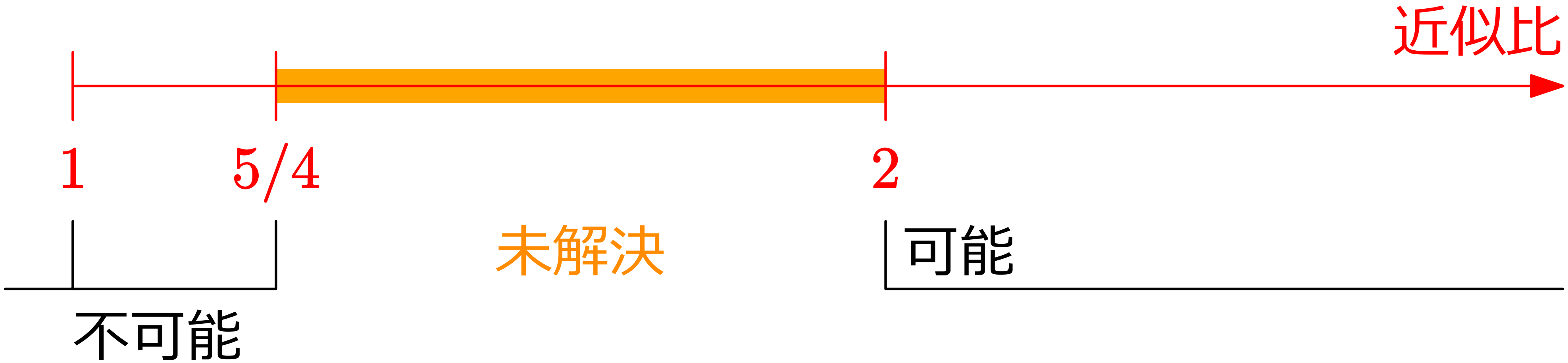
P $\neq$ NP の仮定の下で



(Williamson, Hall, Hoogeveen, Hurkens, Lenstra, Sevastianov, Shmoys '07)

$O \parallel C_{\max}$：多項式時間近似可能性

$\mathrm{P} \neq \mathrm{NP}$ の仮定の下で

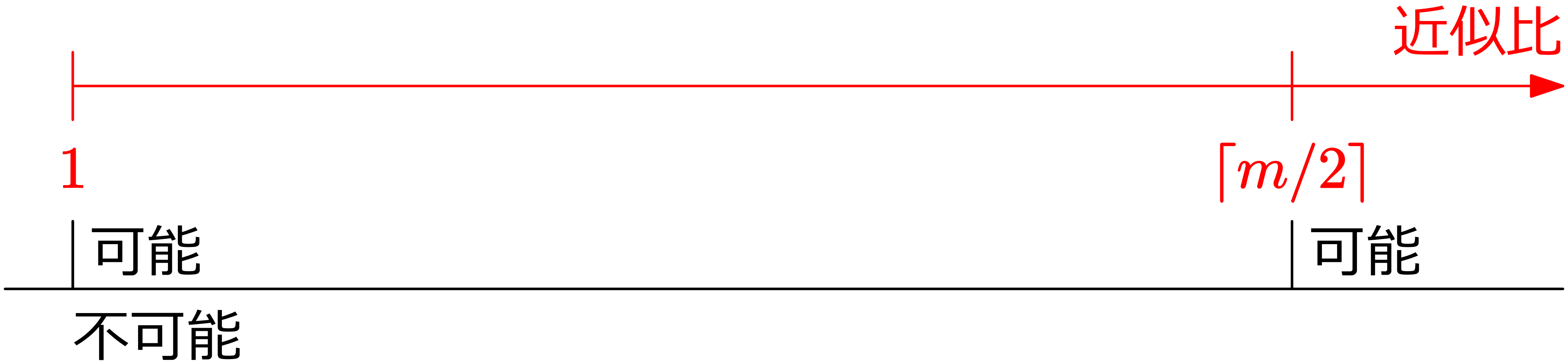


(Williamson, Hall, Hoogeveen, Hurkens, Lenstra, Sevastianov, Shmoys '07)

P $\neq$ NP の仮定の下で

$Fm \,||\, C_{\max}$: 多項式時間近似可能性

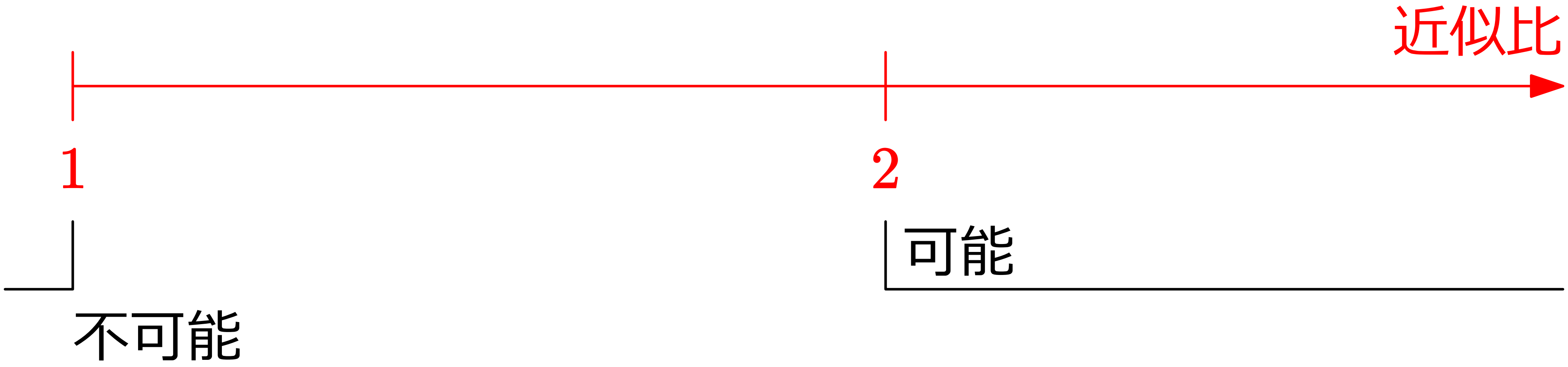
P $\neq$ NP の仮定の下で



(Hall ’98)

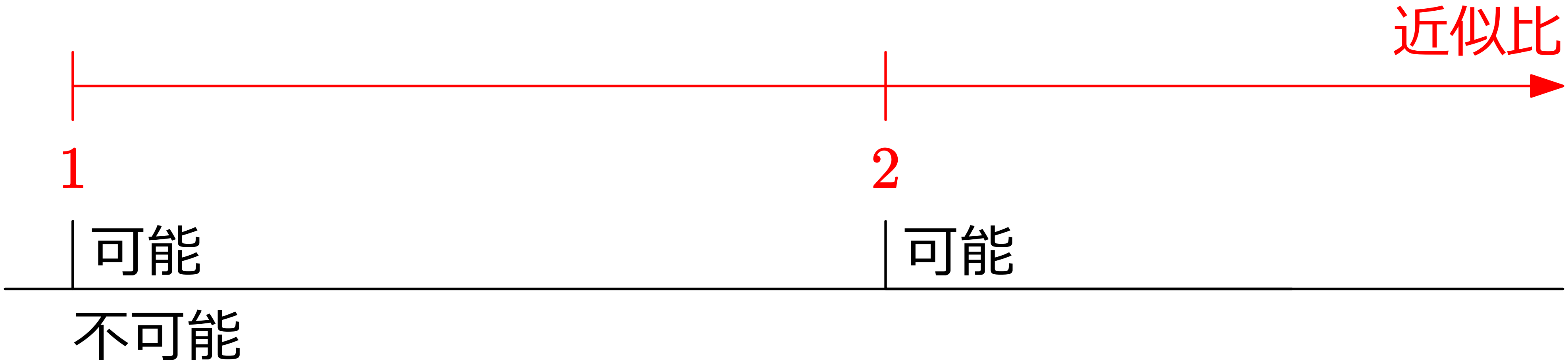
$Om \mid\mid C_{\max}$：多項式時間近似可能性

$\mathrm{P} \neq \mathrm{NP}$ の仮定の下で



$Om \mid\mid C_{\max}$：多項式時間近似可能性

P $\neq$ NP の仮定の下で



(Sevastianov, Woeginger ’98)

$1+\varepsilon$ 近似アルゴリズム？

$\varepsilon > 0$ は定数

定理 (Sevastianov, Woeginger '98)

問題 $Om \mid\mid C_{\max}$ に対して，
$O(m(m/\varepsilon)^{2^{m/\varepsilon}} + mn\log n)$ 時間 $1+\varepsilon$ 近似アルゴリズムが存在する

$Om \mid\mid C_{\max}$ において，m は定数なので，

$$O(m(m/\varepsilon)^{2^{m/\varepsilon}} + mn\log n) = O(n\log n)$$

多項式時間近似スキーム

定義：多項式時間近似スキーム

問題 $\mathcal{P}$ に対する **多項式時間近似スキーム** とは,
次のようなアルゴリズム
　入力：$\varepsilon > 0$
　出力：$\mathcal{P}$ の多項式時間 $1+\varepsilon$ 近似アルゴリズム

polynomial-time approximation scheme (PTAS)　ピータス

ε

$\downarrow$

$\boxed{Om \mid\mid C_{\max} \text{ の PTAS}}$

$\downarrow$

$Om \mid\mid C_{\max}$ の
多項式時間
$1+\varepsilon$ 近似アルゴリズム

定義：多項式時間近似スキーム

問題 $\mathcal{P}$ に対する **多項式時間近似スキーム** とは,
次のようなアルゴリズム
　入力：$\varepsilon > 0$
　出力：$\mathcal{P}$ の多項式時間 $1+\varepsilon$ 近似アルゴリズム

polynomial-time approximation scheme (PTAS)　ピータス

ε

$Om \mid\mid C_{\max}$ の PTAS

$Om \mid\mid C_{\max}$ の入力

$Om \mid\mid C_{\max}$ の多項式時間 $1+\varepsilon$ 近似アルゴリズム

$Om \mid\mid C_{\max}$ の出力

次回，次々回の予告

課題

- どのように，「P $\neq$ NP という仮定の下で，多項式時間 α 近似アルゴリズムが存在しない」ことを証明できるのか？
- どのように，「多項式時間近似スキーム」を設計できるのか？

次回と次々回で，この課題に取り組む