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 1 回

スケジューリング問題の分類

岡本 吉央　(電気通信大学)

okamotoy@uec.ac.jp

2024 年 10 月 1 日

最終更新：2024 年 9 月 29 日　20:17

講義概要

概要

離散最適化における１つのトピックを集中的に学ぶ

今年度のトピック

ジョブ・スケジューリングのアルゴリズム

特に，1950 年代から 1990 年代までの **基礎的内容** を扱う

実践ではなく，**理論 (数理)** を扱う

本講義の目標

次の 2 つができるようになること

目標 1

基本的なアルゴリズム設計技法 を使って
基本的なスケジューリング問題 を解けるようになる

目標 2

問題の数学的構造がアルゴリズム設計指針を与える
という重要な考え方を実践する

メッセージ：数学的な考え方からアルゴリズムが得られる

解きやすい問題と解きにくい問題

多くの組合せ最適化問題は次のどちらかに分類される

解きやすい問題

- 最小全域木問題
- 最大マッチング問題
- 最大流問題
- ...

多項式時間アルゴリズムで解ける

解きにくい問題

- 巡回セールスマン問題
- 最小 Steiner 木問題
- ナップサック問題
- ...

NP 困難である

解きやすい問題と解きにくい問題

多くの組合せ最適化問題は次のどちらかに分類される

解きやすい問題

- 最小全域木問題
- 最大マッチング問題
- 最大流問題
- ...

多項式時間アルゴリズムで解ける

解きにくい問題

- 巡回セールスマン問題
- 最小 Steiner 木問題
- ナップサック問題
- ...

NP 困難である

問 1：どのスケジューリング問題が解きやすい/にくいのか？

問 2：この違いは何に起因するのか？

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)

授業の進め方：基本情報

教員

岡本吉央 (おかもと よしお)

- 居室：西 4-206
- E-mail：okamotoy@uec.ac.jp

講義 Web ページ

http://dopal.cs.uec.ac.jp/okamotoy/lect/2024/sched/

- 講義スライド
- コメント回答
- レポート出題
- その他

Google Classroom

内部シラバスのコードを見て，各自が登録

授業の進め方：各回の進行 (予定)

10:40	授業開始
11:20 頃	休憩 (3 分程度)
	授業再開
	授業終了
12:10	コメント投稿
13:00	コメント受付終了

授業の進め方：各回の進行 (予定)

10:40	授業開始	
11:20 頃	休憩 (3 分程度)	
	授業再開	
	授業終了	
12:10	コメント投稿	
13:00	コメント受付終了	

Google Classroom のフォームから投稿

- 質問・疑問
- 誤植の指摘
- 感想
- 要望
- 文句
- 雑談
- など何でも

(匿名で収集)

授業の進め方：評価

評価方法

2 回のレポート提出 **のみ** による

- 1 回 50 点満点

成績

$$\text{素点} = \text{レポート 1 の得点} + \text{レポート 2 の得点}$$

授業の考え方：数学と証明

注意点 1

この授業では，数学を使い，数学的な証明もする

$\boxed{\text{理由}}$

- 授業の内容の正しさが **検証** できるべき
- 証明という **技法** を身につけるべき

$\leadsto$ この授業では **批判的思考**，**論理的思考** の体得 も目指す

授業の考え方：プログラミング

注意点 2

この授業では，プログラミングを行わない

理由

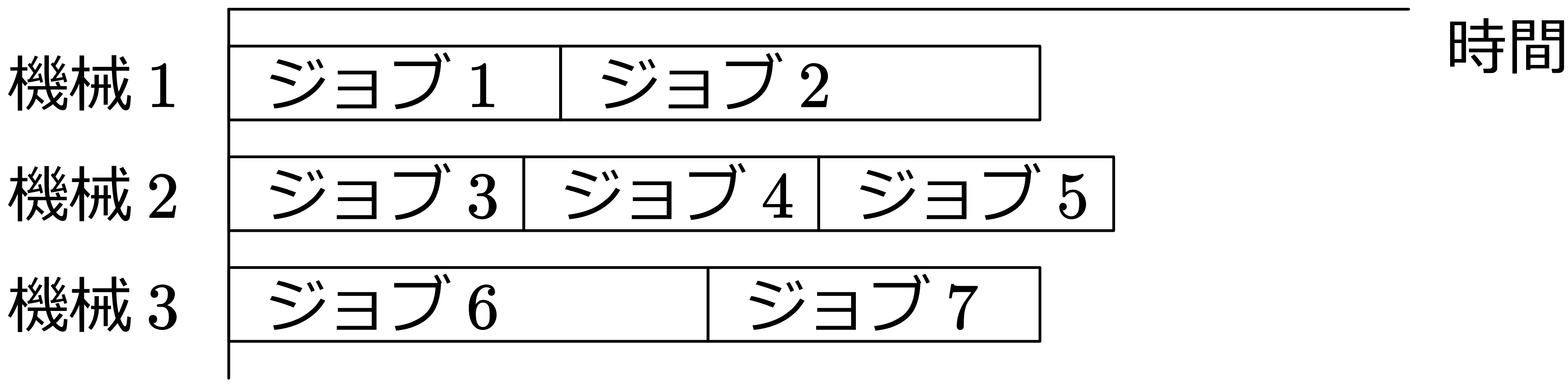
- 教員がまじめにプログラミングをしたことがない
  (∴ 教員が授業で扱えない)
- 受講生自身が自主的にプログラミングをすればよい

$\rightsquigarrow$ 授業の内容から **自由に はみ出ていく** ことを推奨する

本日の内容

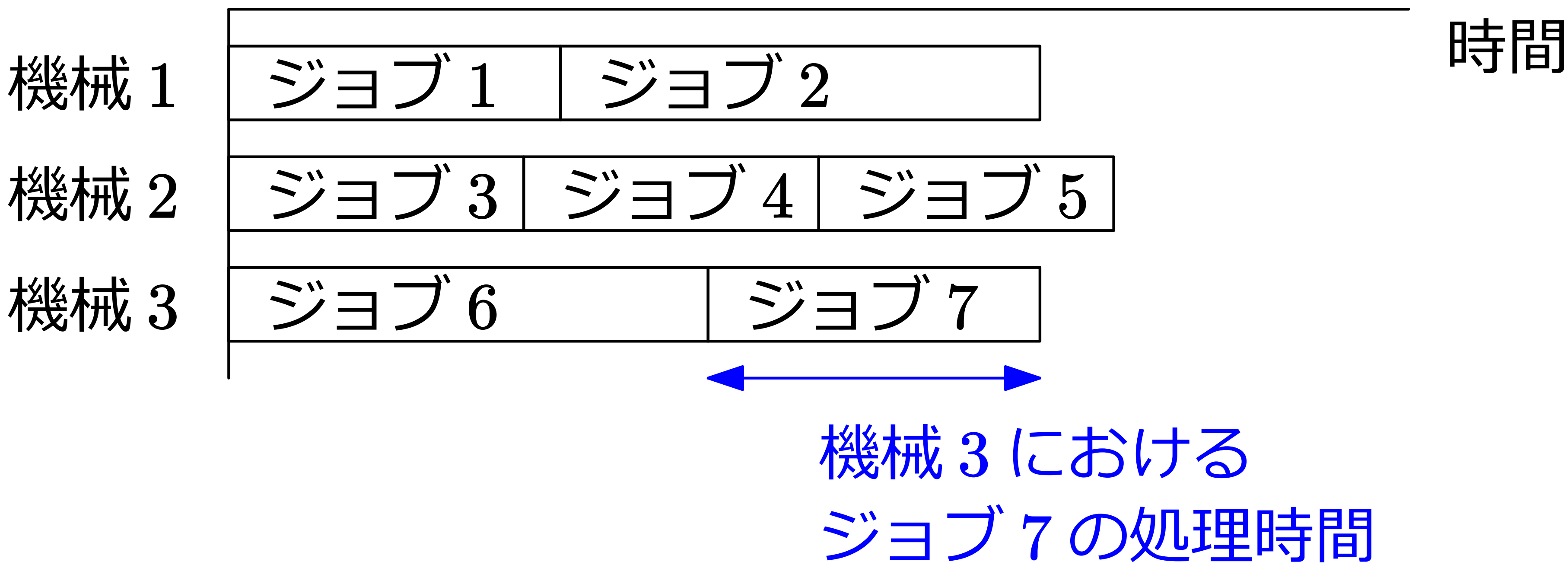
1. **ジョブ・スケジューリングとは？**
2. 3 つ組記法
3. 最適解と最適値

Gantt チャート：例



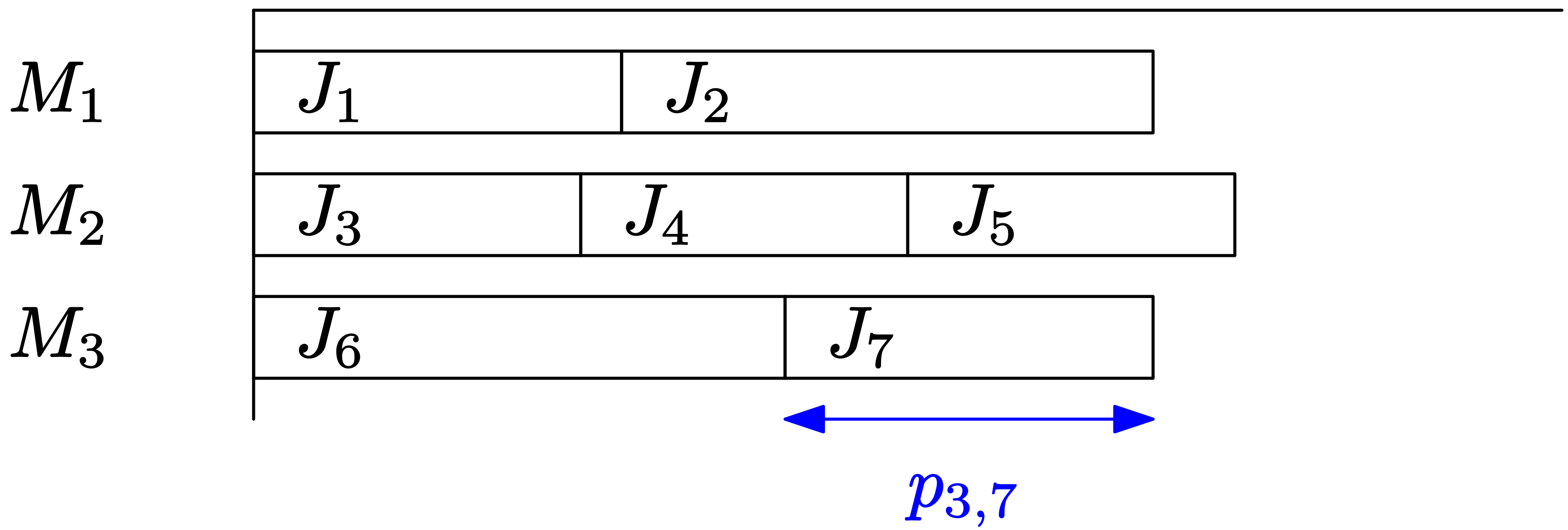
注：これと違うものを Gantt チャートと呼ぶことも多い

Gantt チャート：例



注：これと違うものを Gantt チャートと呼ぶことも多い

ジョブ・スケジューリングの構成要素



- 機械の集合 $\mathcal{M} = \{M_1, M_2, \ldots, M_m\}$
  添字 $i \in \{1, 2, \ldots, m\} =: [m]$
- ジョブの集合 (仕事) $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$
  添字 $j \in \{1, 2, \ldots, n\} =: [n]$
- 処理時間 $p \in \mathbb{Q}_{\geq 0}^{m \times n}$ (非負有理数の行列)
  $p_{ij} = M_i$ における J_j の処理時間

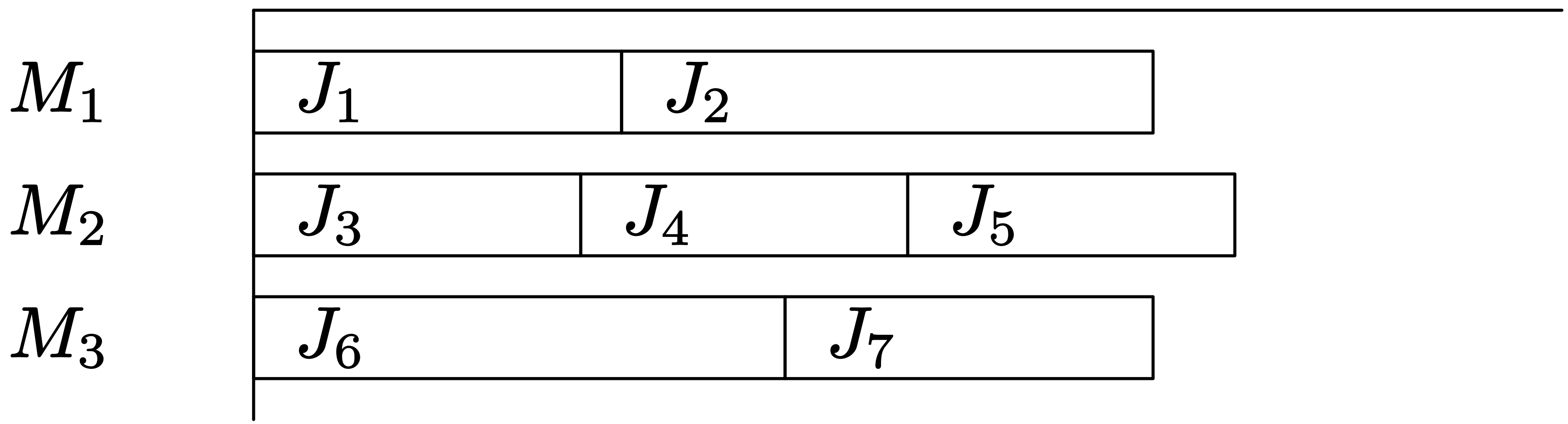
構成要素：処理時間

$$
\begin{array}{c|ccccccc|}
 & J_1 & J_2 & J_3 & J_4 & J_5 & J_6 & J_7 \\
\hline
M_1 & 9 & 13 & 8 & 9 & 9 & 20 & 12 \\
M_2 & 10 & 13 & 8 & 8 & 8 & 15 & 10 \\
M_3 & 12 & 15 & 8 & 10 & 10 & 13 & \boxed{9} \\
\hline
\end{array}
$$

$p_{3,7}$

- 機械の集合　$\mathcal{M} = \{M_1, M_2, \ldots, M_m\}$
  添字 $i \in \{1, 2, \ldots, m\} =: [m]$
- ジョブの集合 (仕事)　$\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$
  添字 $j \in \{1, 2, \ldots, n\} =: [n]$
- 処理時間　$p \in \mathbb{Q}_{\geq 0}^{m \times n}$ (非負有理数の行列)
  $p_{ij} = M_i$ における J_j の処理時間

スケジュール = 割当 + 順序付け



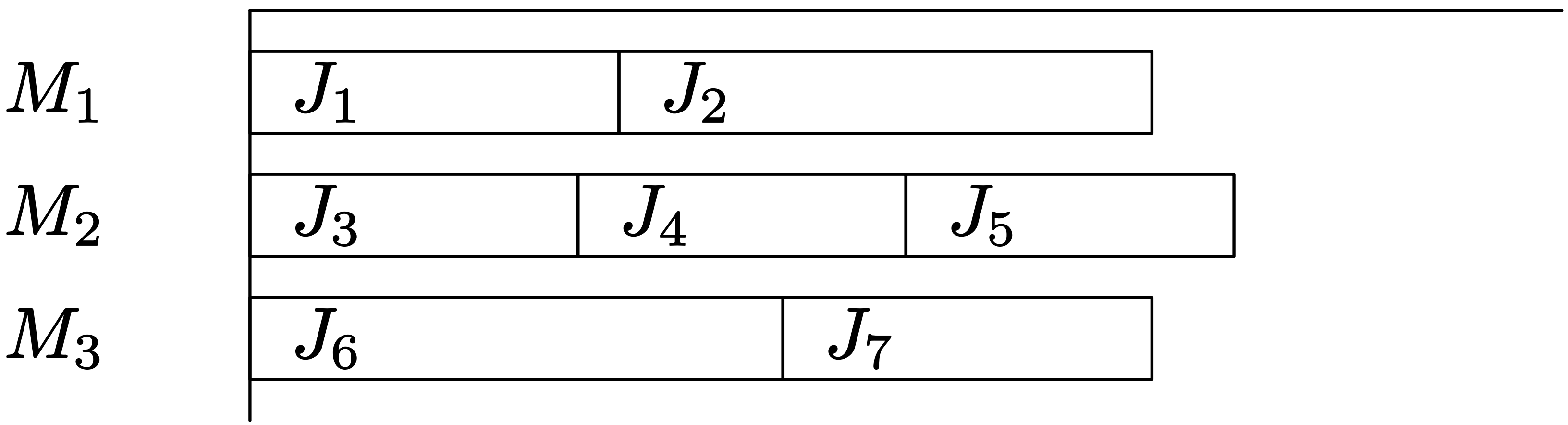
定義 (非形式) : スケジュール (schedule)

スケジュール とは次の 2 つを定めたもの

- **割当** (assignment)
  各ジョブを処理する機械
- **順序付け** (sequencing)
  各ジョブの処理を開始する時刻

注：この定義から外れるスケジュールもある

スケジューリングにおける制約



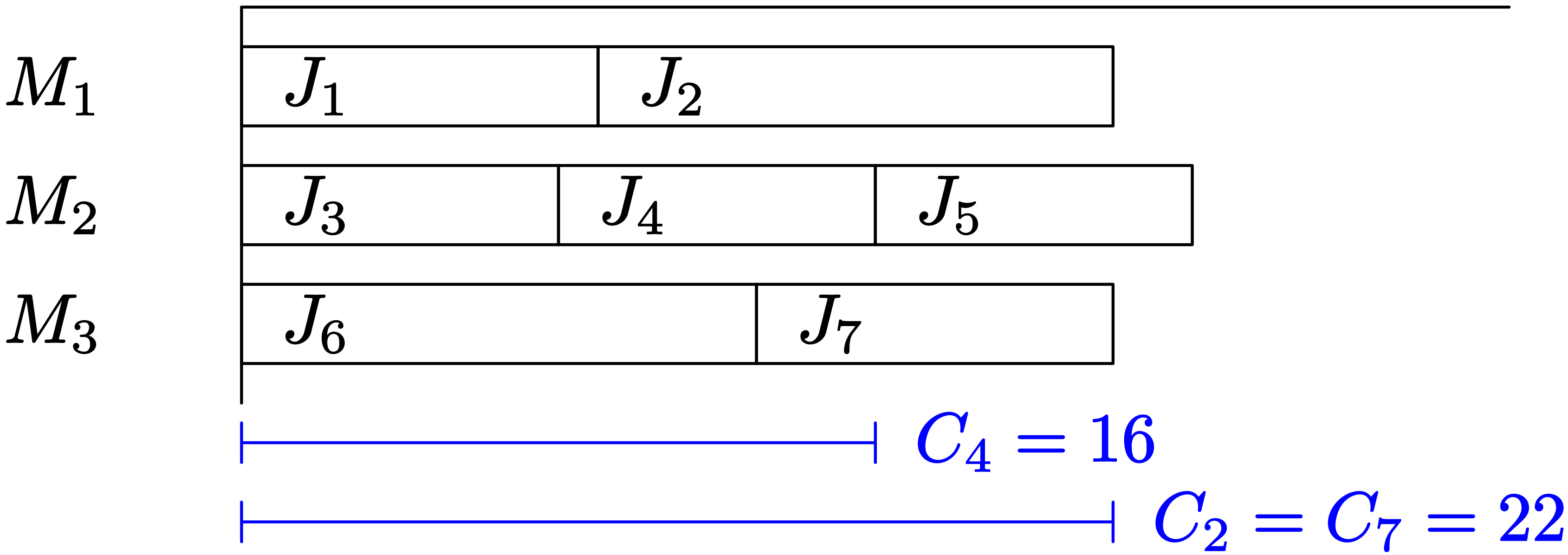
スケジューリングにおける制約

スケジュールは次の 2 つを満たさなくてはならない

- 任意の時刻において，
  どのジョブも高々 1 つの機械で処理される
- 任意の時刻において，
  どの機械も高々 1 つのジョブを処理する

注：この制約を満たさないスケジュールを考えることもある

完了時刻



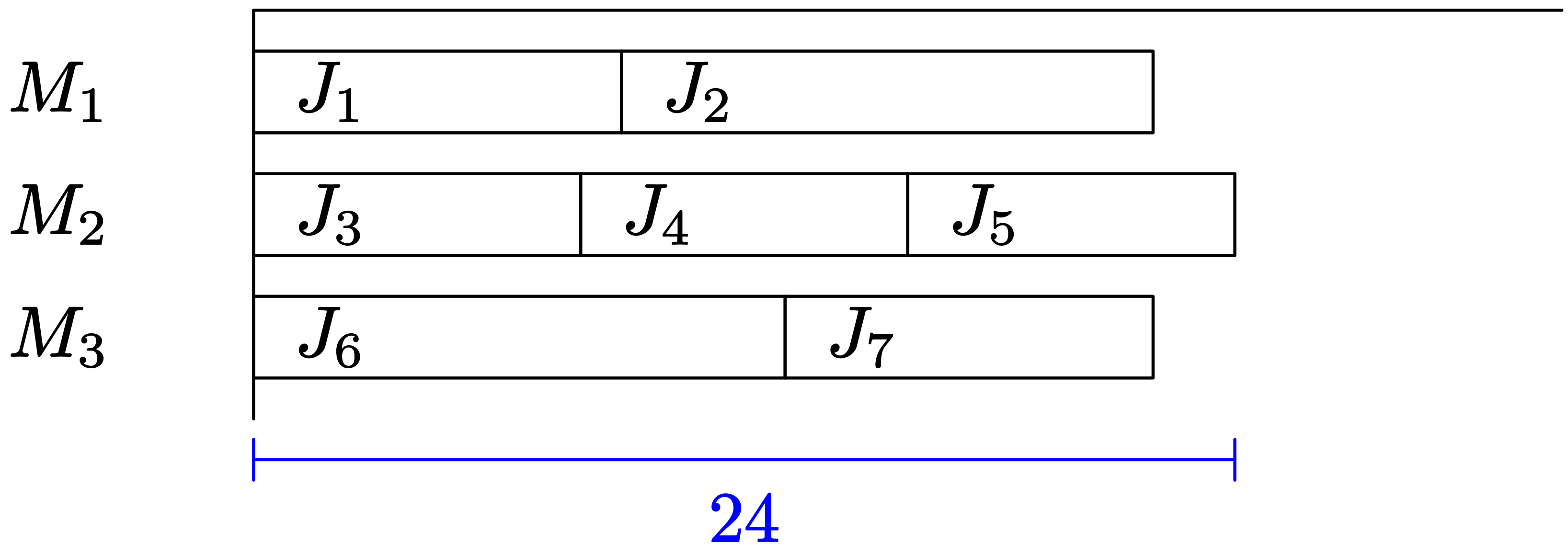
スケジュールが決まる $\longrightarrow$ 各ジョブの完了時刻が決まる

記法：完了時刻 (completion time)

スケジュール σ に対して

$C_j(\sigma) =$ スケジュール σ におけるジョブ J_j の完了時刻

目的：最大完了時刻最小化



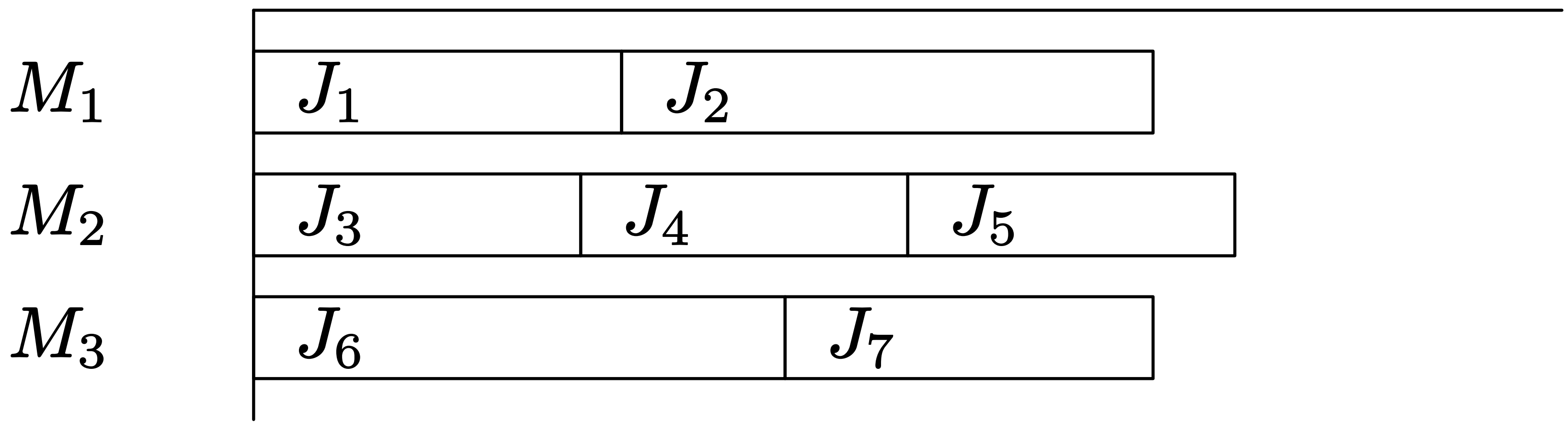
典型的な目的：最大完了時刻最小化

次の **最大完了時刻** (makespan) を最小にするような
スケジュール σ を求めたい

$$\max\{C_1(\sigma), C_2(\sigma), \ldots, C_n(\sigma)\}$$

他の目的を考えることもよくある (後述)

スケジューリング



定義 (非形式)：スケジューリング (scheduling)

スケジューリング は，次のどちらかを意味する
- よいスケジュールを見つける問題
- よいスケジュールを見つけるアルゴリズム

「よい」は文脈・設定に依存する

1. ジョブ・スケジューリングとは？
2. **3 つ組記法**
3. 最適解と最適値

- R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics* 5 (1979) 287–326.

スケジューリングの 3 つ組記法

ジョブ・スケジューリングの問題には多くの変種がある

$\rightarrow$ 次の **3 つ組記法** (three-field notation) がよく使われる

(Graham, Lawler, Lenstra, Rinnooy Kan ’79)

ジョブ・スケジューリングの 3 つ組記法

ジョブ・スケジューリングの問題 1 つを次の記法で表す

$$\alpha \mid \beta \mid \gamma$$

ここで，各部分は次を表す

- α は 機械の環境 (machine environment)
- β は ジョブの特性 (job characteristics)
- γ は 最適化する目的 (optimality criterion)

機械の環境：一機械

機械の環境：一機械 (single-machine)

$\alpha = 1$：機械が 1 台

	J_1	J_2	J_3	J_4	J_5	J_6	J_7
M_1	9	13	8	9	9	20	12

機械の環境：同一並列機械

機械の環境：同一並列機械 (identical parallel machines)

$\alpha = Pm$:

- 機械が m 台 $(m \geq 1)$
- ジョブの処理時刻が機械に依存しない

	J_1	J_2	J_3	J_4	J_5	J_6	J_7
M_1	9	13	8	9	9	20	12
M_2	9	13	8	9	9	20	12
M_3	9	13	8	9	9	20	12

$$\forall\, j \in [n], \exists\, p_j \in \mathbb{Q}_{\geq 0}, \forall\, i \in [m], p_{ij} = p_j$$

機械の環境：一様並列機械 (uniform parallel machines)

$\alpha = Qm$：

- 機械が m 台 $(m \geq 1)$
- 各機械 M_i のスピード $s_i > 0$ が決まっていて，ジョブの処理時間が $1/s_i$ に比例する

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	s_i
M_1	16	8	8	4	12	20	24	1
M_2	8	4	4	2	6	10	12	2
M_3	4	2	2	1	3	5	6	4
p_j	16	8	8	4	12	20	24	

$$\forall\, j \in [n],\ \exists\, p_j \in \mathbb{Q}_{\geq 0},\ \forall\, i \in [m],\ \exists\, s_i > 0,\ p_{ij} = p_j / s_i$$

機械の環境：無関係並列機械 (unrelated parallel machines)

$\alpha = Rm$：
- 機械が m 台 $(m \geq 1)$

	J_1	J_2	J_3	J_4	J_5	J_6	J_7
M_1	9	13	8	9	9	20	12
M_2	10	13	8	8	8	15	10
M_3	12	15	8	10	10	13	9

機械の環境：まとめ

- 1：一機械
- Pm：同一並列機械　(機械数 m)
- Qm：一様並列機械　(機械数 m)
- Rm：無関係並列機械　(機械数 m)

- P：同一並列機械　(機械数の指定なし (可変))
- Q：一様並列機械　(機械数の指定なし (可変))
- R：無関係並列機械　(機械数の指定なし (可変))

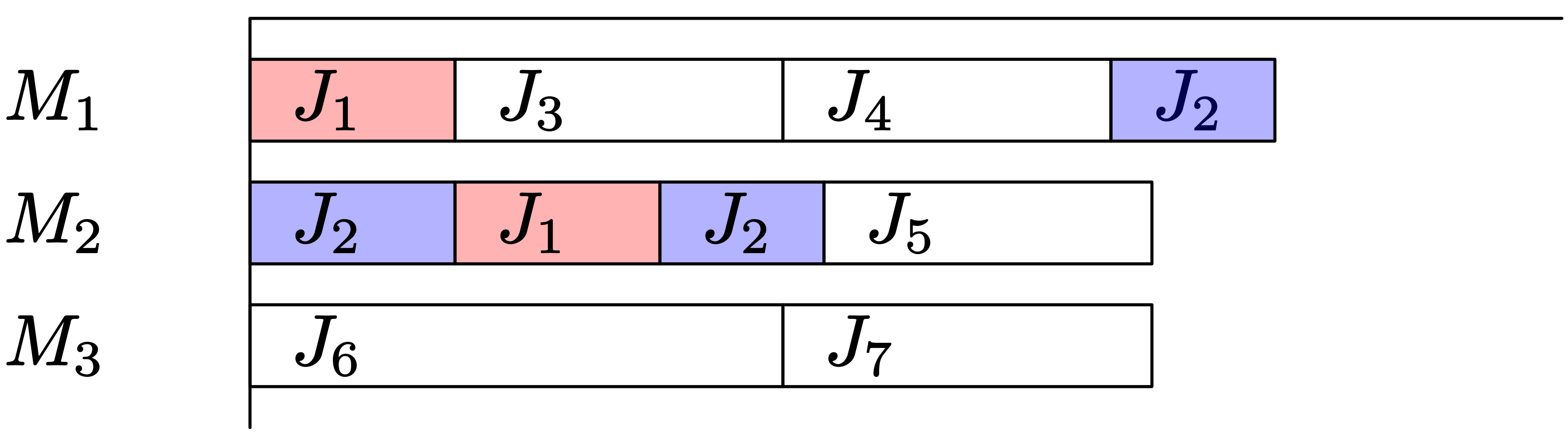
後半で，Jm, Fm, Om が登場する予定

β にはジョブのいろいろな特性をカンマで並べる

ジョブの特性の１つ：分割可能性

- **分割不可能** (non-preemptive) **スケジューリング**とは処理を開始したジョブを完了まで中断できないもの
- **分割可能** (preemptive) **スケジューリング**とは分割不可能ではないこと

分割可能であるとき，β に「pmtn」と書く　(preemption)
「prmp」と書くことも



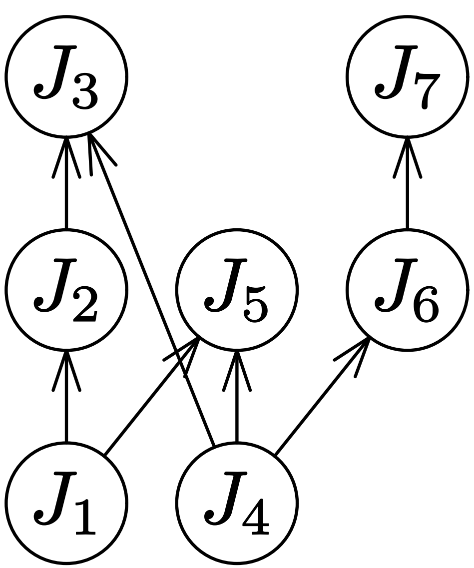
ジョブの特性：先行制約

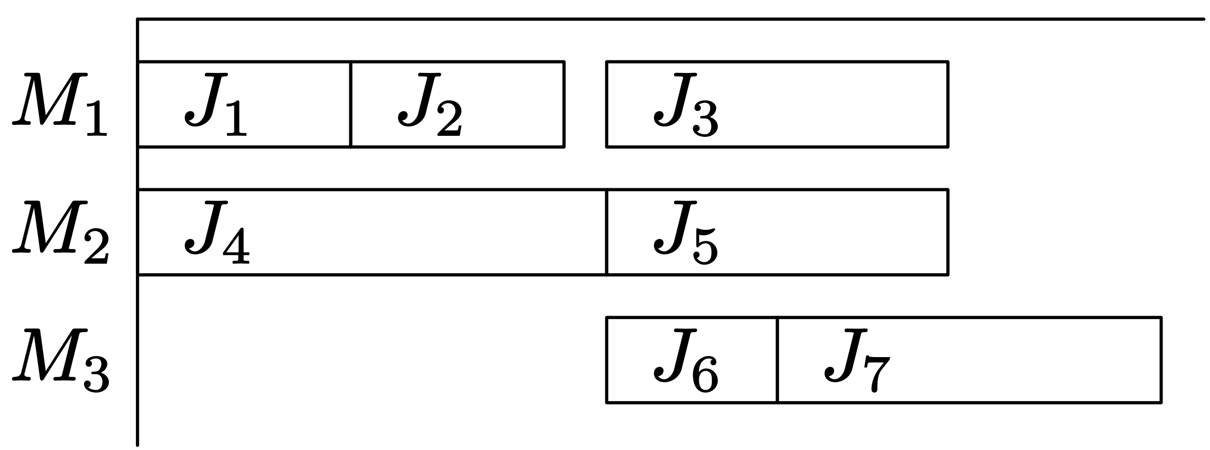
β にはジョブのいろいろな特性をカンマで並べる

ジョブの特性の１つ：先行制約 (precedence constraint)

ジョブの集合 $\mathcal{J}$ 上の半順序 $\to$ を使って
J_j **の処理が完了しないと** $J_{j'}$ **の処理を開始できない**
ことを $J_j \to J_{j'}$ で表す

先行制約があるとき，β に「prec」と書く　　(precedence)





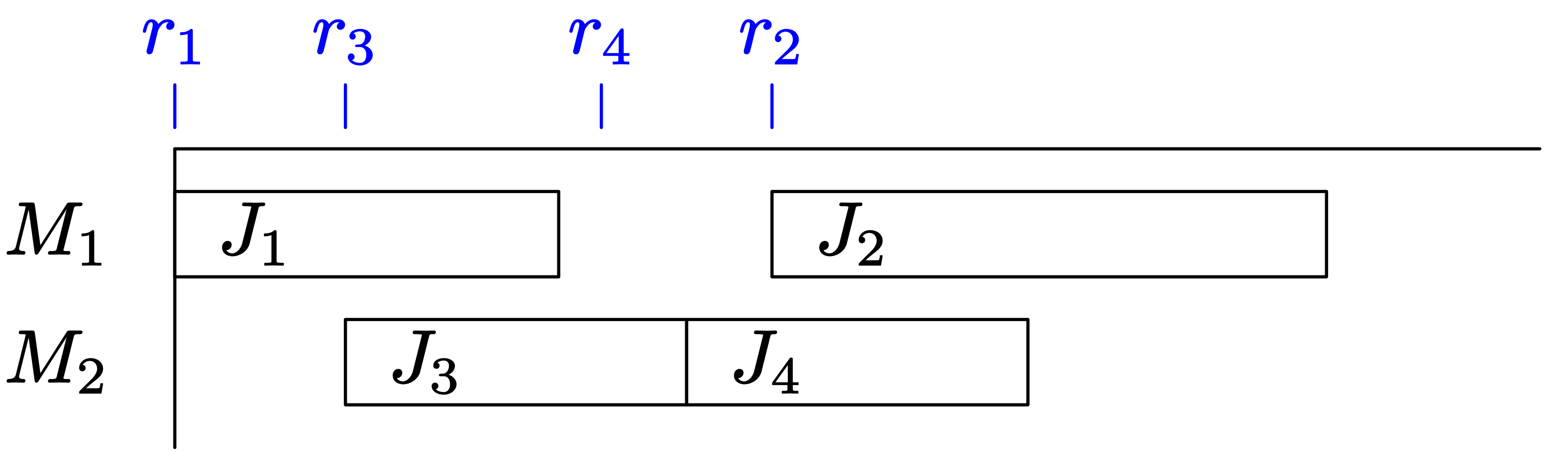
ジョブの特性：到着時刻

β にはジョブのいろいろな特性をカンマで並べる

ジョブの特性の１つ：到着時刻 (release date)

ジョブ J_j の **到着時刻** $r_j \geq 0$ が設定されているとき
時刻 r_j にならないと J_j の処理は開始できない

到着時刻があるとき，β に「r_j」と書く



最適化する目的：まとめ

γ に最適化する目的を書く

最適化する目的：代表的なもの

次の最小化

$C_{\max}$	最大完了時刻	(makespan)
$\sum C_j$	総完了時刻	(total completion time)
$L_{\max}$	最大納期ずれ	(maximum lateness)
$\sum T_j$	総納期遅れ	(total tardiness)
$\sum U_j$	納期遅れジョブ数	(the number of late jobs)

- 他の目的を考えることもある
- 重み和を考えることもある $\leftarrow$ 今後登場 $(\sum w_j C_j, \sum w_j T_j, \sum w_j U_j)$

最適化する目的：最大完了時刻

γ に最適化する目的を書く

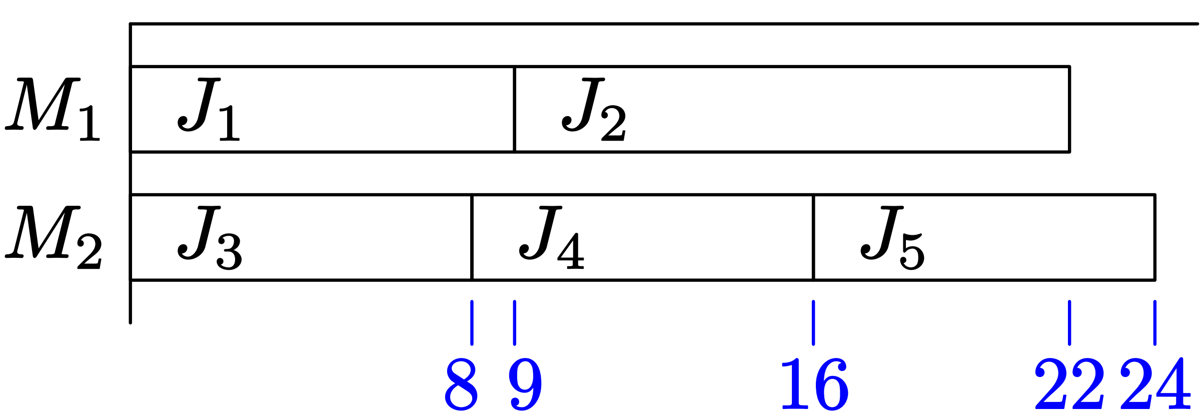
最適化する目的：最大完了時刻 (総所要時間)

$\gamma = C_{\max}$

- 次を最小化

$$C_{\max}(\sigma) = \max\{C_1(\sigma), C_2(\sigma), \ldots, C_n(\sigma)\}$$

- ただし，各 $j \in [n]$ に対して，
  $C_j(\sigma) =$ スケジュール σ における J_j の完了時刻



$C_{\max} = 24$

最適化する目的：総完了時刻

γ に最適化する目的を書く

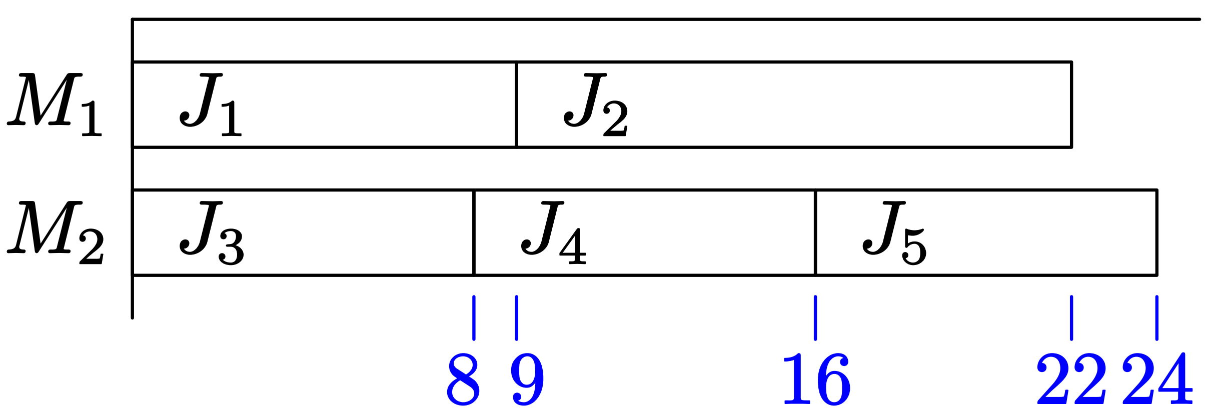
最適化する目的：総完了時刻

$\gamma = \sum C_j$

- 次を最小化

$$\sum C_j(\sigma) = C_1(\sigma) + C_2(\sigma) + \cdots + C_n(\sigma)$$

- ただし，各 $j \in [n]$ に対して，
  $C_j(\sigma) =$ スケジュール σ における J_j の完了時刻



$\sum C_j = 79$

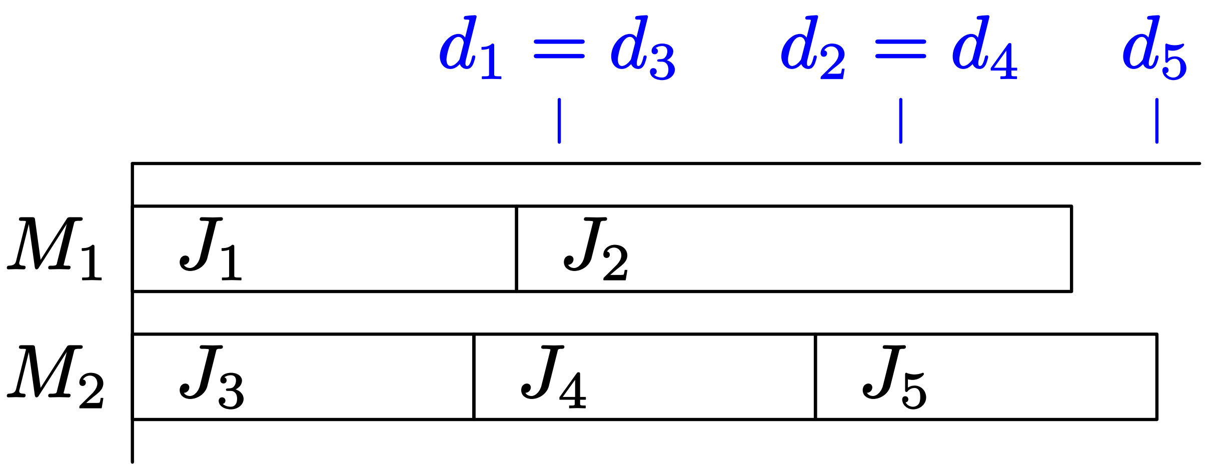
準備：納期

$L_{\max}, \sum T_j, \sum U_j$ では，ジョブの納期を考える

ジョブの納期 (due date)

各ジョブ J_j に対して **納期** $d_j \geq 0$ が
入力の一部として与えられる場合がある

- 気持ち：$C_j(\sigma) \leq d_j$ であるようにしたい



J_2 が遅れジョブ，

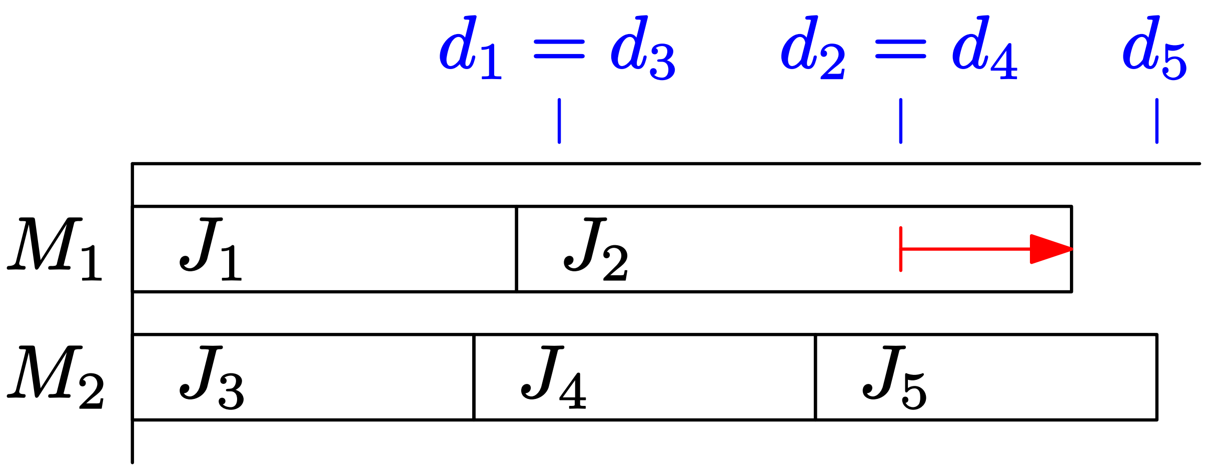
準備：納期

$L_{\max}, \sum T_j, \sum U_j$ では，ジョブの納期を考える

ジョブの納期 (due date)

各ジョブ J_j に対して **納期** $d_j \geq 0$ が
入力の一部として与えられる場合がある

- 気持ち：$C_j(\sigma) \leq d_j$ であるようにしたい



J_2 が遅れジョブ，納期ずれ $=$ 納期遅れ $= 4$

γ に最適化する目的を書く

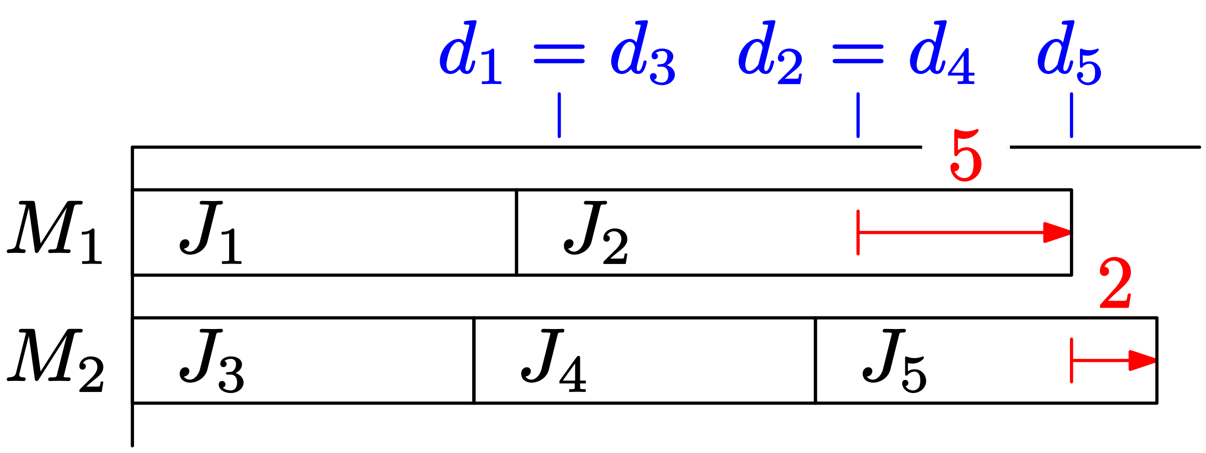
最適化する目的：最大納期ずれ

$\gamma = L_{\max}$

- 次を最小化

$$L_{\max}(\sigma) = \max\{L_1(\sigma), L_2(\sigma), \dots, L_n(\sigma)\}$$

- ただし，各 $j \in [n]$ に対して，$L_j(\sigma) = C_j(\sigma) - d_j$ (σ における J_j の**納期ずれ** (lateness))



$L_{\max} = 5$

γ に最適化する目的を書く

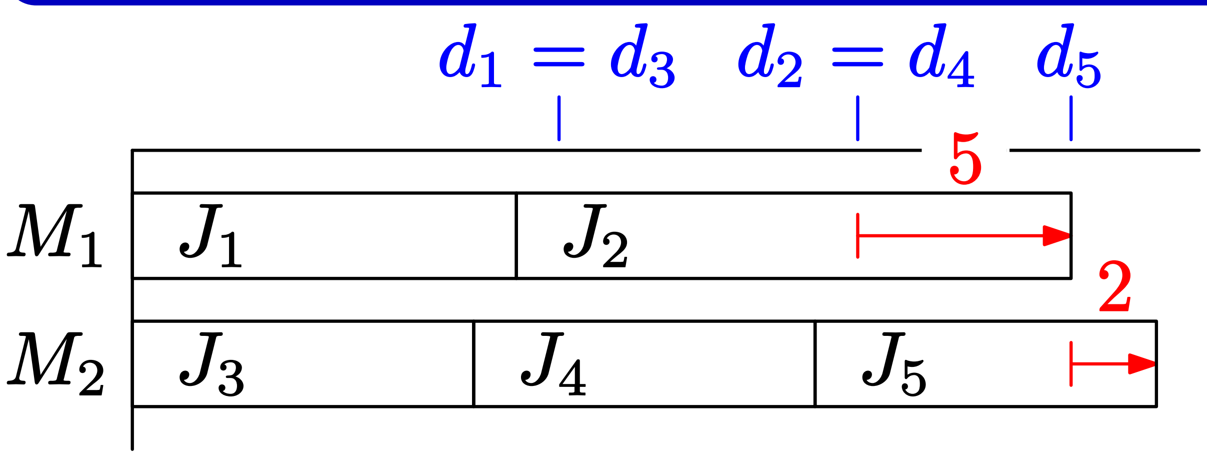
最適化する目的：総納期遅れ

$\gamma = \sum T_j$

- 次を最小化

$$\sum T_j(\sigma) = T_1(\sigma) + T_2(\sigma) + \cdots + T_n(\sigma)$$

- 各 $j \in [n]$ に対して，$T_j(\sigma) = \max\{0, C_j(\sigma) - d_j\}$
  (σ における J_j の**納期遅れ** (tardiness))



$\sum T_j = 7$

最適化する目的：納期遅れジョブ数

γ に最適化する目的を書く

最適化する目的：納期遅れジョブ数

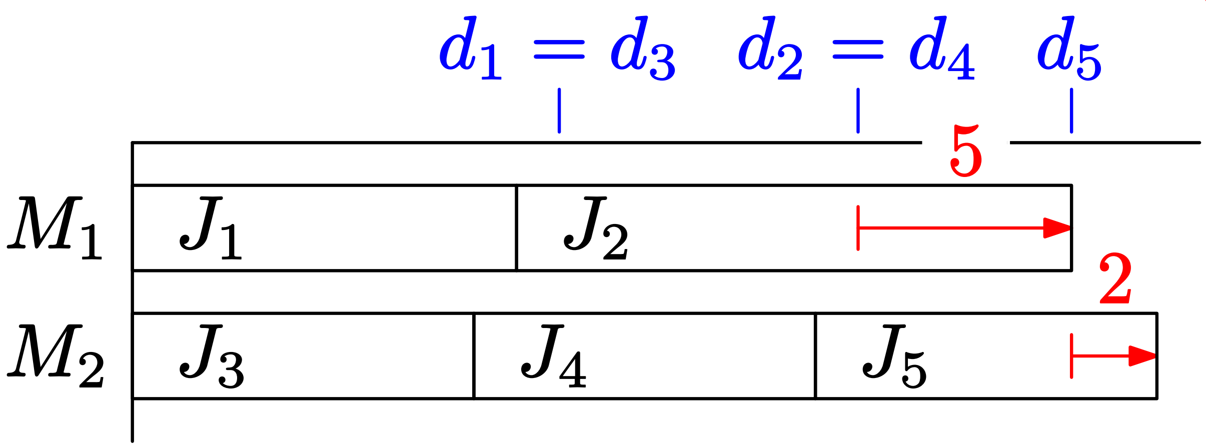
$\gamma = \sum U_j$

- 次を最小化

$$\sum U_j(\sigma) = U_1(\sigma) + U_2(\sigma) + \cdots + U_n(\sigma)$$

- 各 $j \in [n]$ に対して，$U_j(\sigma) = \underline{[C_j(\sigma) > d_j]}$

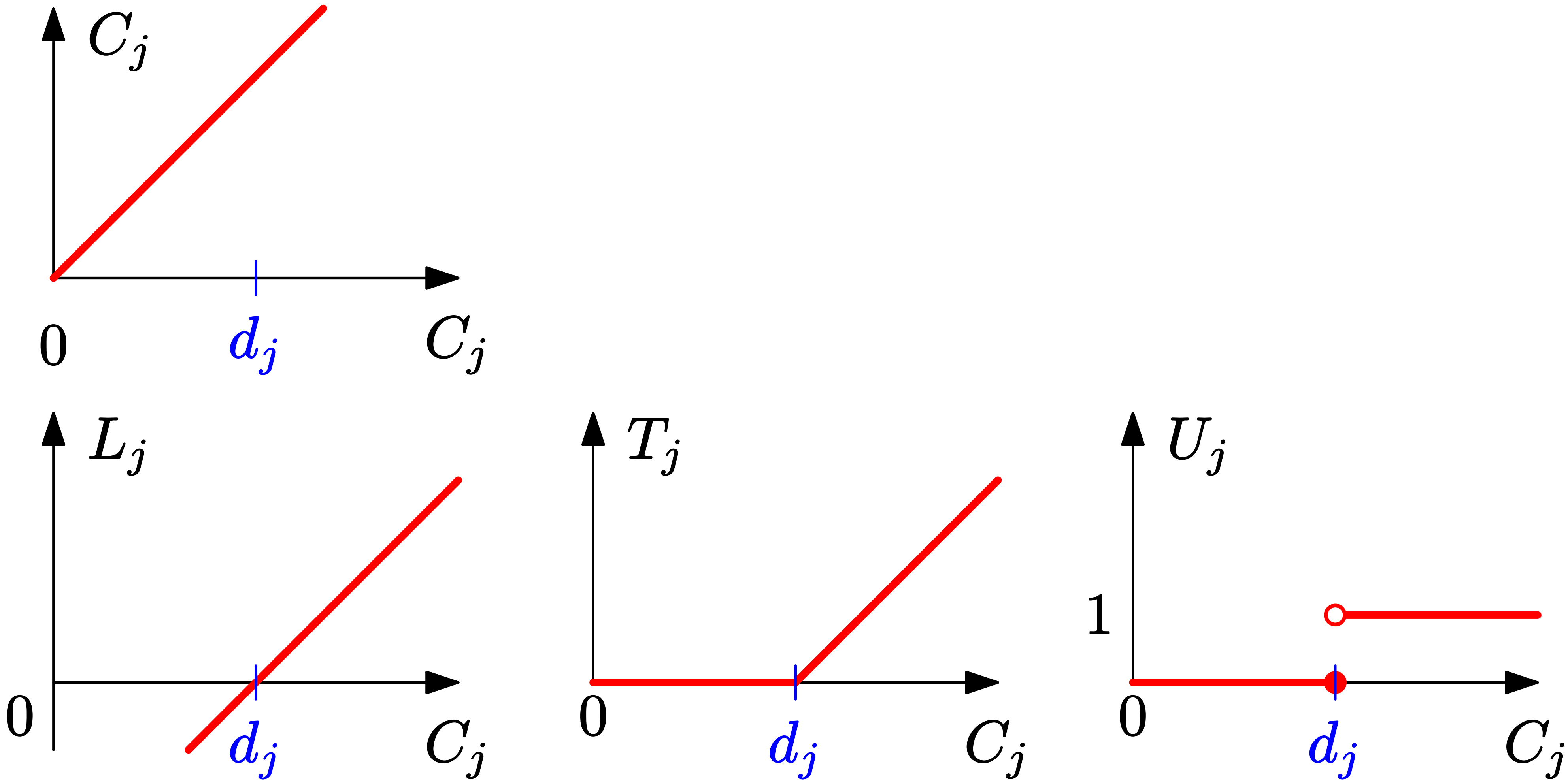
Iverson の括弧



$$[X] = \begin{cases} 0 & (X \text{ が偽}) \\ 1 & (X \text{ が真}) \end{cases}$$

$$\sum U_j = 2$$

目的の比較



$L_j = C_j - d_j$ $\quad$ $T_j = \max\{0, C_j - d_j\}$ $\quad$ $U_j = [C_j > d_j]$

γ に最適化する目的を書く

最適化する目的：代表的なもの

次の最小化

$C_{\max}$	最大完了時刻	(makespan)
$\sum C_j$	総完了時刻	(total completion time)
$L_{\max}$	最大納期ずれ	(maximum lateness)
$\sum T_j$	総納期遅れ	(total tardiness)
$\sum U_j$	納期遅れジョブ数	(the number of late jobs)

- 他の目的を考えることもある
- 重み和を考えることもある $\leftarrow$ 今後登場 $(\sum w_j C_j, \sum w_j T_j, \sum w_j U_j)$

3 つ組記法：例

3 つ組記法の例 (1)

問題 $P2 \,||\, \sum T_j$

- 同一並列機械，機械数 2
- 総納期遅れの最小化

3 つ組記法の例 (2)

問題 $1 \mid r_j, \mathsf{prec} \mid \sum U_j$

- 一機械
- 到着時刻あり，先行制約あり
- 納期遅れジョブ数の最小化

この授業が扱うスケジューリング問題

特に断らないかぎり，次を仮定する

- **すべてのデータは解く前に与えられる**
  (機械の環境，ジョブの特性，最適化する目的，
  処理時間，納期，到着時刻，先行制約，...)
  (↔ オンライン・スケジューリング)
- **不確実性はない** (↔ 確率的スケジューリング)
- **最適化する目的は一つに限る** (↔ 多目的最適化)
- **処理時間は処理順や処理開始時刻に依存しない**
  (↔ 巡回セールスマン問題)

本日の内容

1. ジョブ・スケジューリングとは？
2. 3 つ組記法
3. **最適解と最適値**

最適解 (最適スケジュール)

例えば，最大完了時刻最小化 $C_{\max}$ の場合を考える

定義：最適解 (最適スケジュール)

スケジュール σ^* が **最適** (optimal) であるとは，
任意のスケジュール σ に対して，次が成り立つこと

$$C_{\max}(\sigma^*) \leq C_{\max}(\sigma)$$

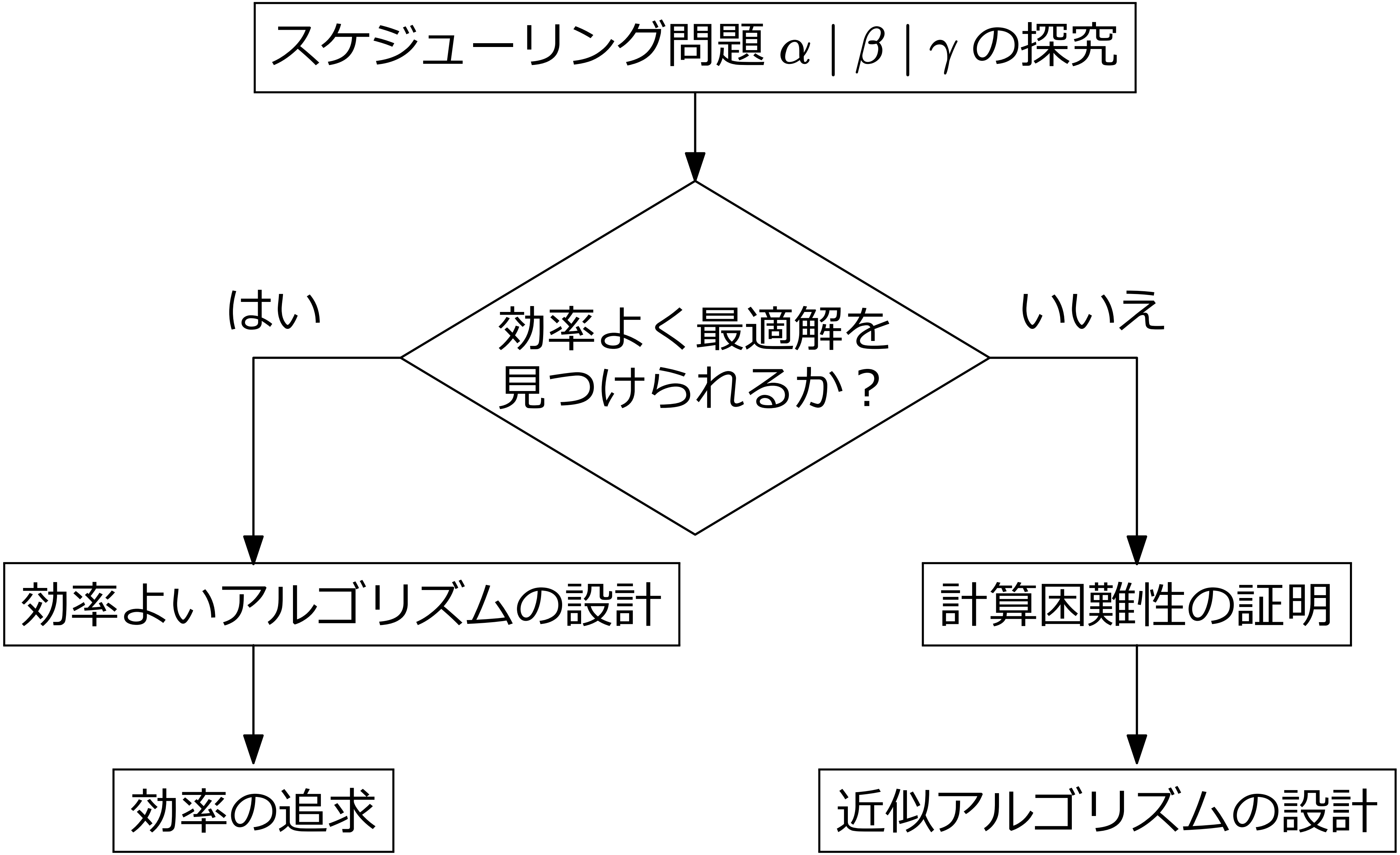
最適スケジュール σ^* に対して，$C_{\max}(\sigma^*)$ を **最適値** と呼ぶ (optimal value)

(他の目的でも同様)

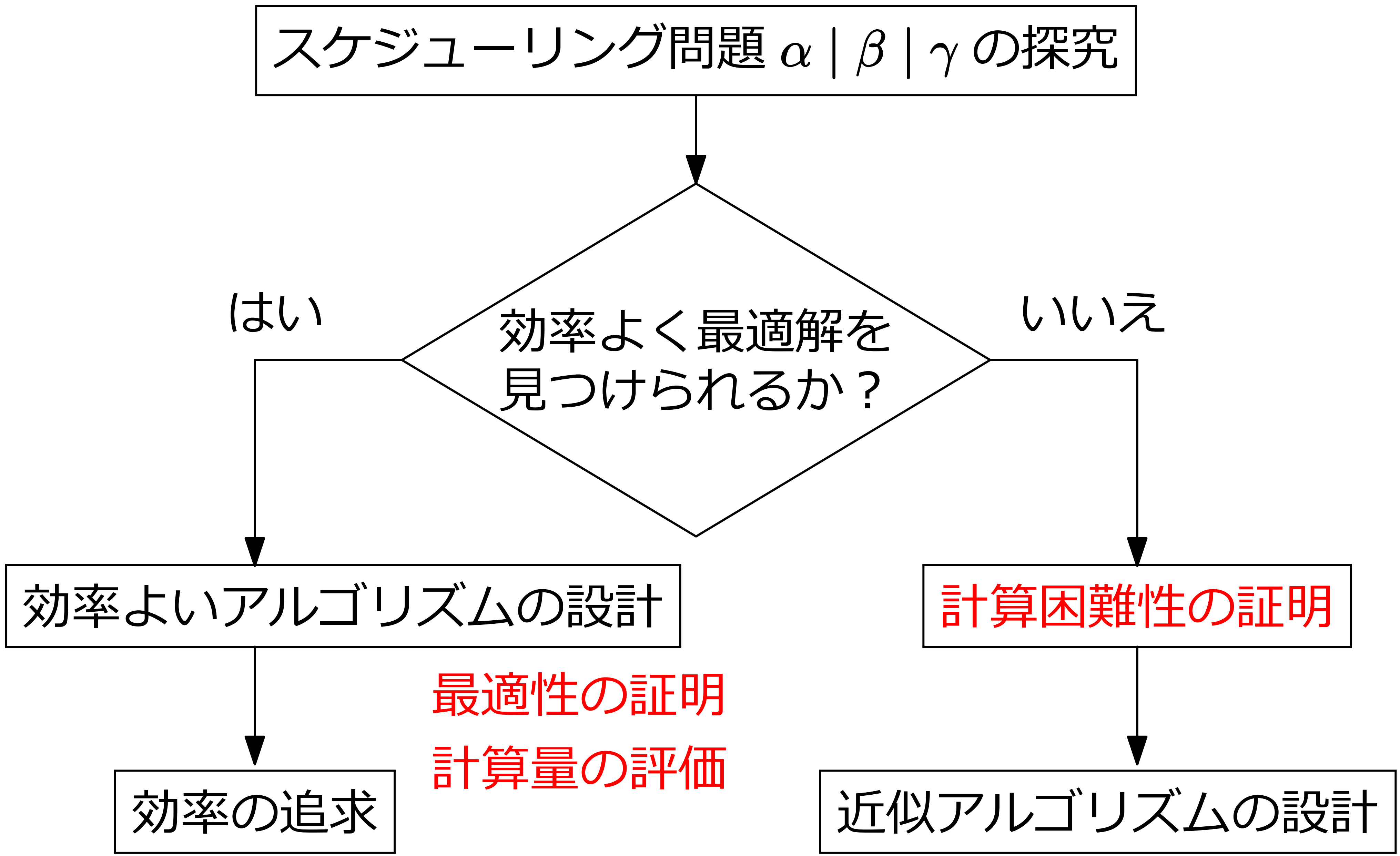
重要な注意：最適化全般において

最適解 と **最適値** という用語を区別して使う

この講義の目標


スケジューリング問題 $\alpha \mid \beta \mid \gamma$ の探究
効率よく最適解を見つけられるか？
はい
いいえ
効率よいアルゴリズムの設計
計算困難性の証明
効率の追求
近似アルゴリズムの設計

この講義の目標


スケジューリング問題 $\alpha \mid \beta \mid \gamma$ の探究
効率よく最適解を見つけられるか？
はい
いいえ
効率よいアルゴリズムの設計
計算困難性の証明
最適性の証明
計算量の評価
効率の追求
近似アルゴリズムの設計

まとめと予告

まとめ

- ジョブ・スケジューリングの構成要素
- 3 つ組記法 $\alpha \mid \beta \mid \gamma$
  - α：機械の環境
  - β：ジョブの特性
  - γ：最適化する目的

次回の予告

効率よく解ける一機械スケジューリング問題

- アルゴリズム設計技法：整列 (sorting)
- 証明技法：ジョブ交換論法