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 3 回

動的計画法

岡本 吉央　(電気通信大学)

okamotoy@uec.ac.jp

2024 年 10 月 29 日

最終更新：2024 年 10 月 28 日　21:22

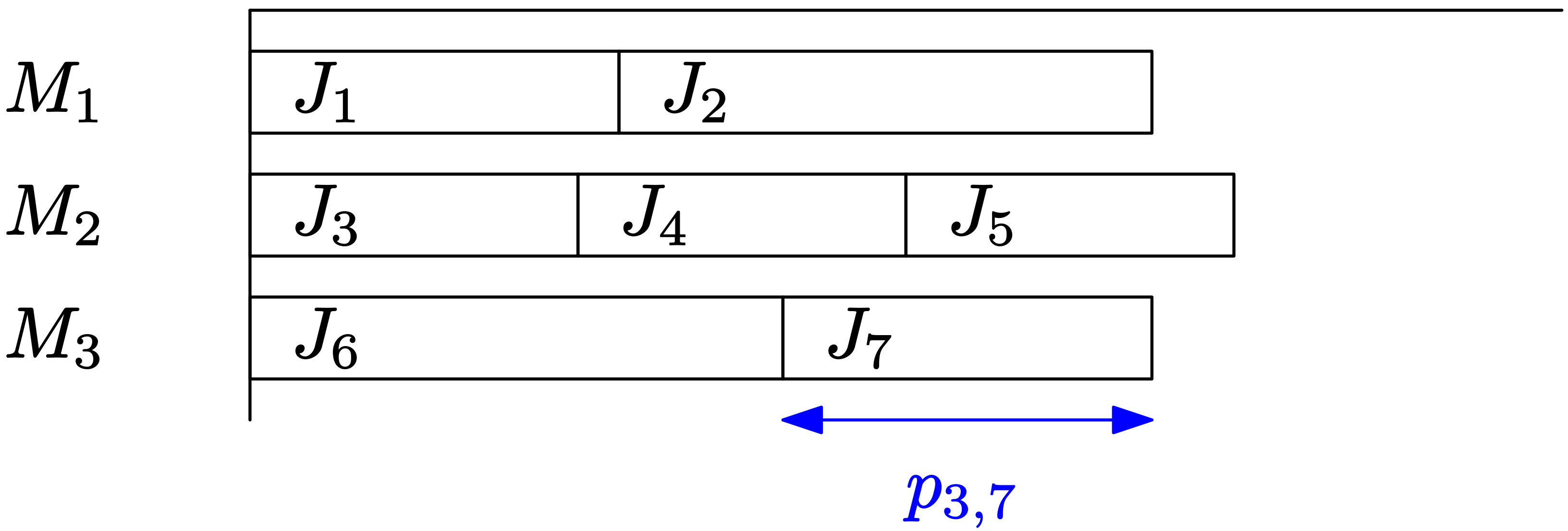
スケジュール (前半)

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)

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

今日の内容

今日の内容

動的計画法 を用いて,
基本的なスケジューリング問題を解く

扱うスケジューリング問題

- $P2 \mid\mid C_{\max}$
- $1 \mid\mid \sum T_j$

動的計画法とは

岩波数学辞典第 4 版 (2007) によると

最適化問題の中には，その過程がいくつもの段階から構成される多段決定問題と見なせるものが数多く存在する．**動的計画法** (dynamic programming) は，多段決定問題を体系的に取り扱う研究分野であり，1950 年以降に R. Bellman が発展させた理論・手法である．動的計画法は，離散最適化問題や組合せ問題に対しても，アルゴリズム設計のパラダイムとしてしばしば使われる．

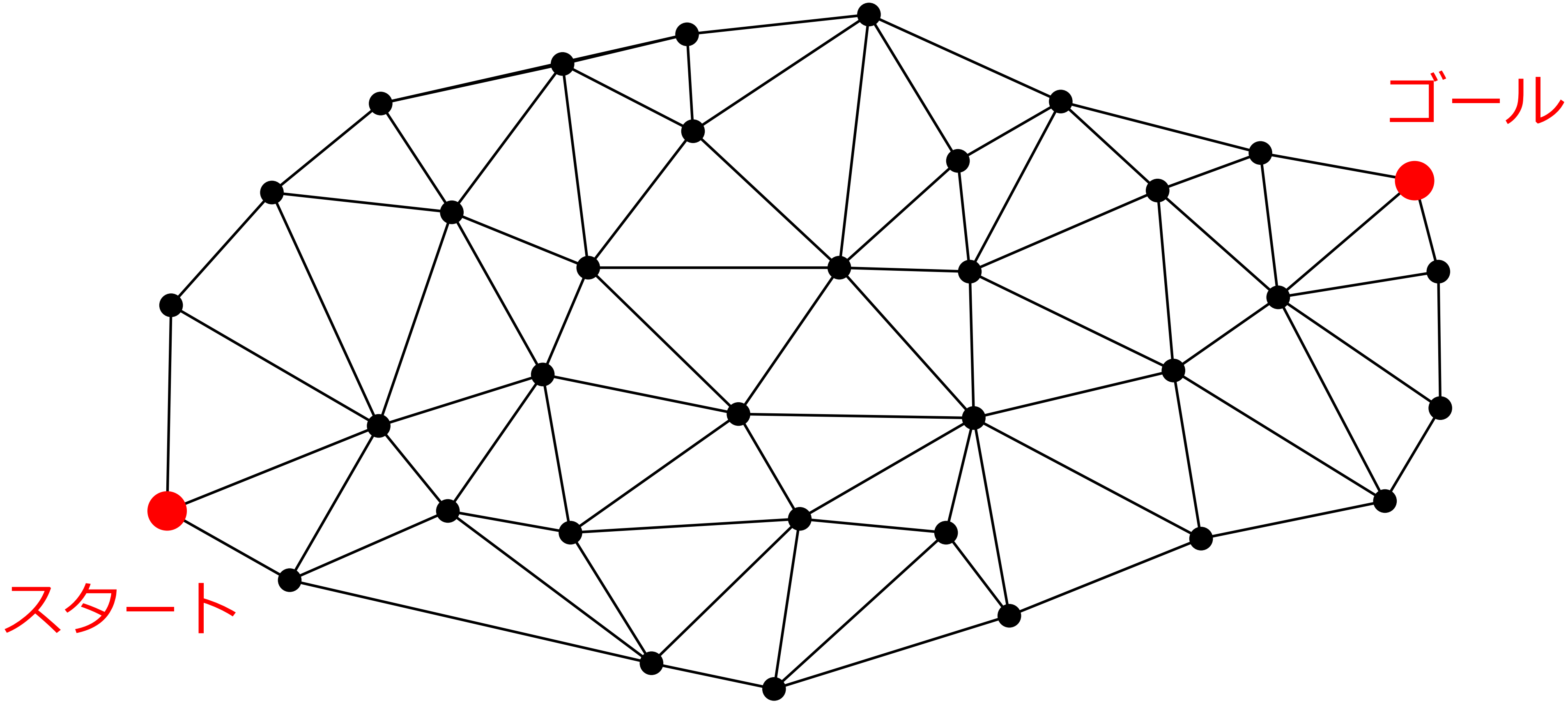
- R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

動的計画法の特徴

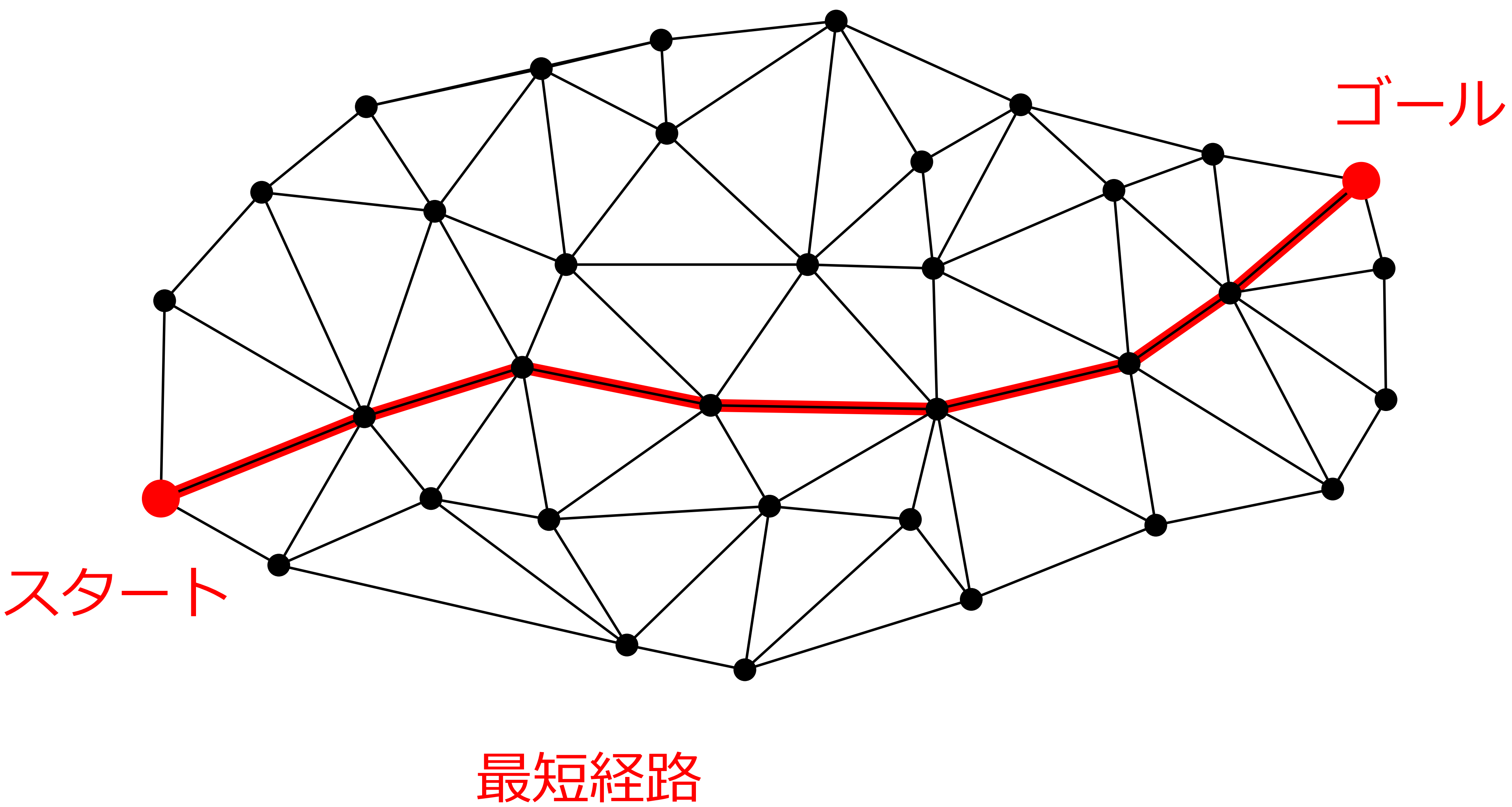
応用数理計画ハンドブック (2002) によると

動的計画の特徴は，意思決定が段階ごとになされる点にある．段階は多くの場合，離散化された時間軸を表すことに用いられる．時間軸を考慮したモデルは，しばしば動的 (dynamic) とよばれ，それが “dynamic” programming のよび名の所以である．初期の動的計画は，段階ごとの部分問題の解の情報を「表」にして保存しておいた．もともと “programming” は表による計算法の意味をもち，それが dynamic “programming” のよび名のもう 1 つの所以である．

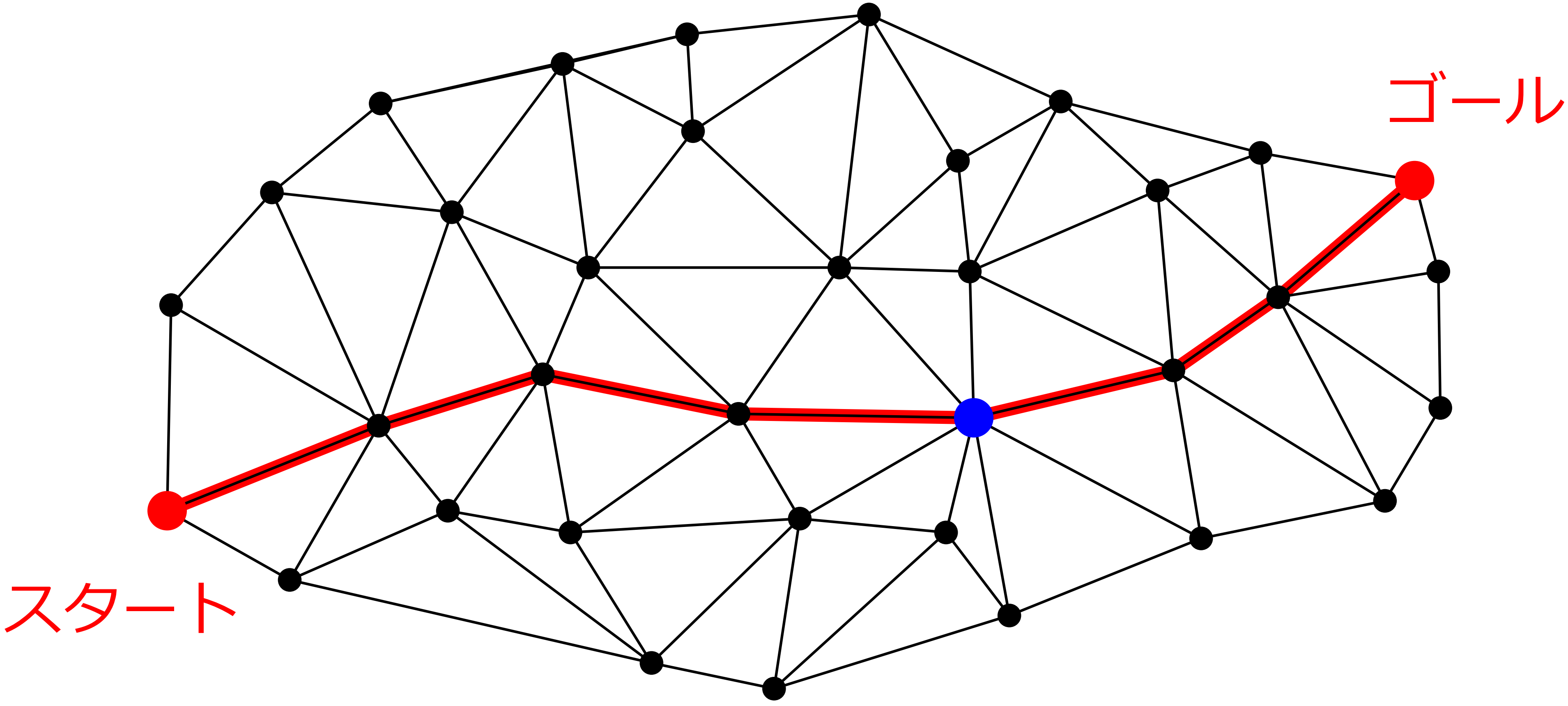
動的計画法の原理



動的計画法の原理

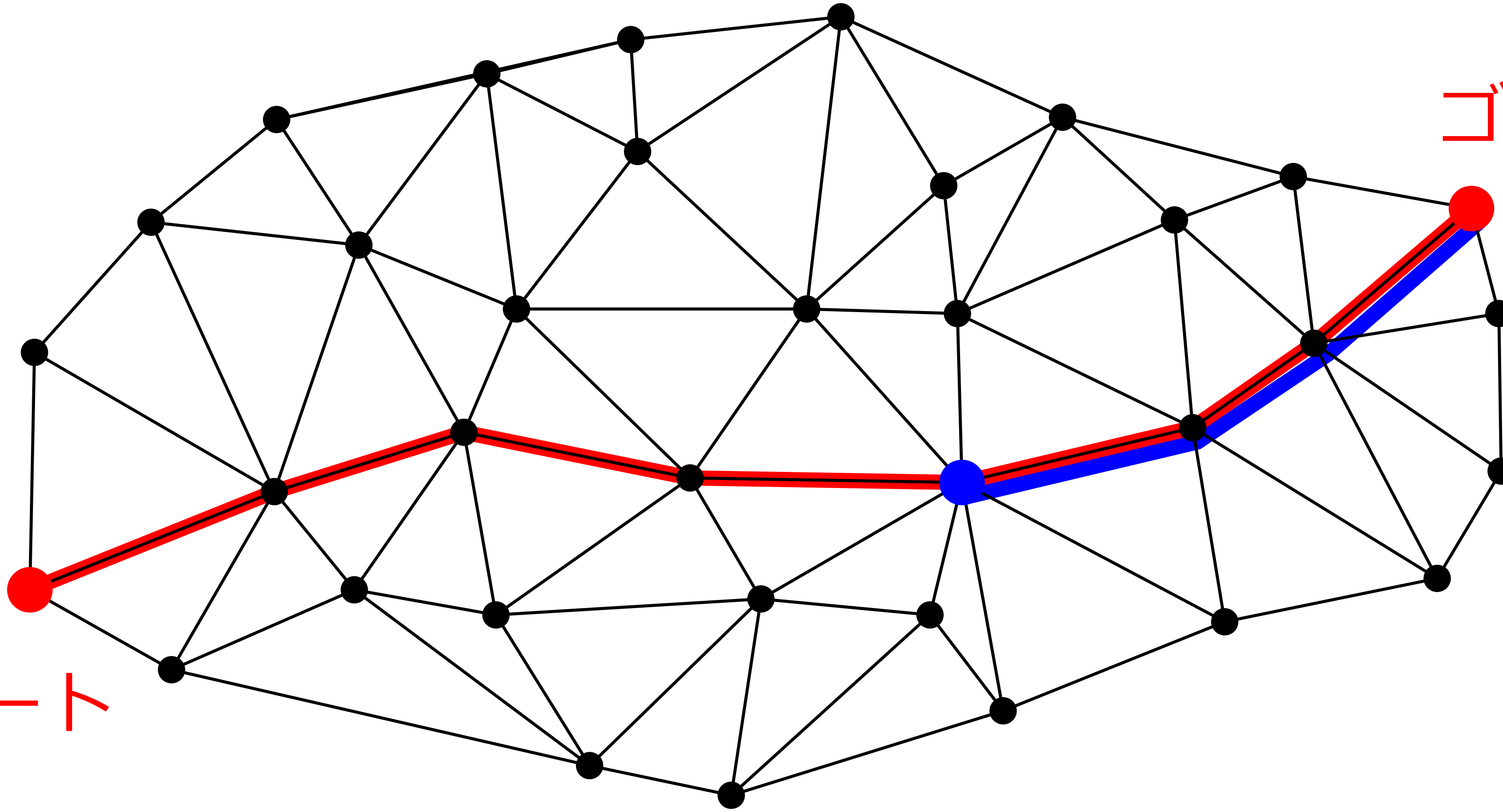


動的計画法の原理



最短経路 の途中の点からの最短経路は

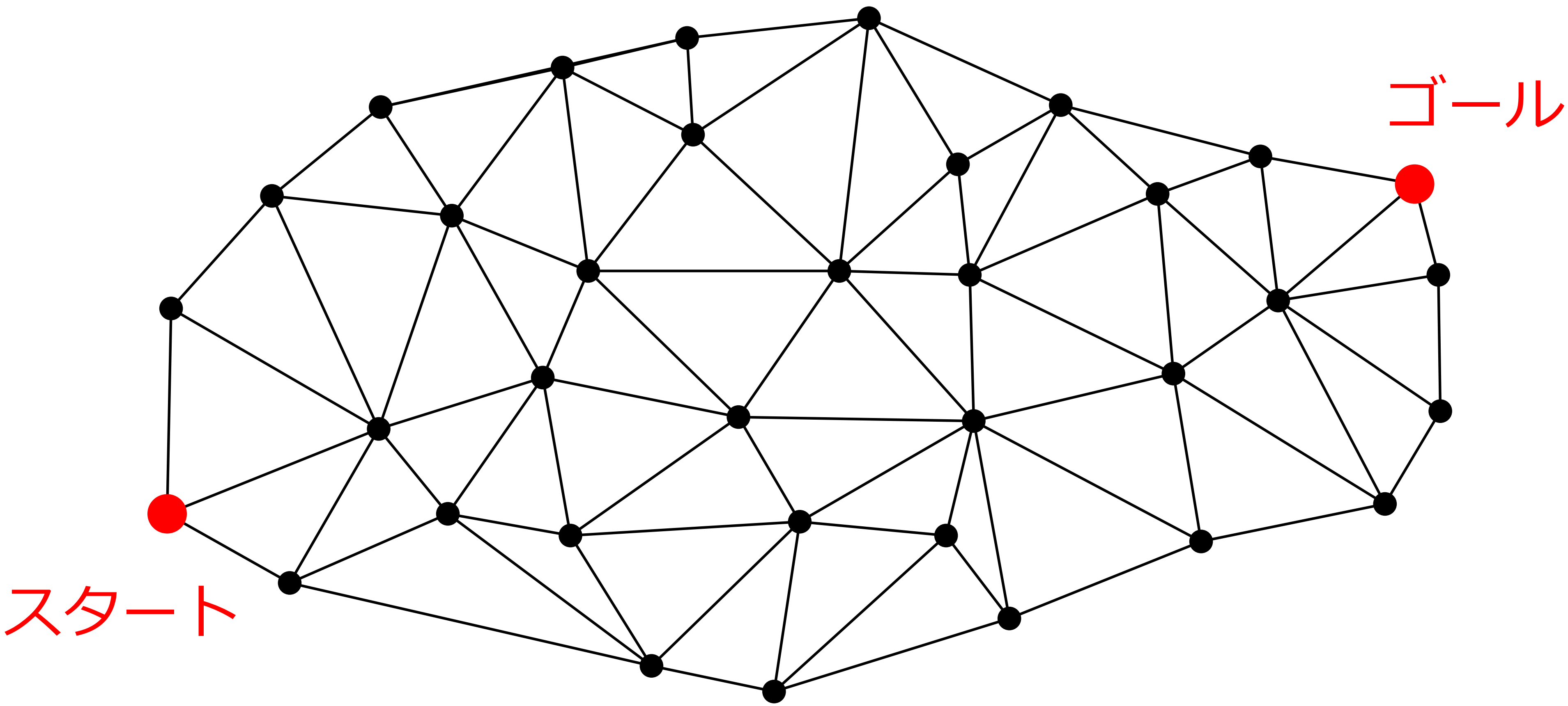
動的計画法の原理



$\boxed{\text{最適性の原理}}$ ： 最短経路の途中の点からの最短経路は
もとの最短経路をたどれば得られる

(principle of optimality)

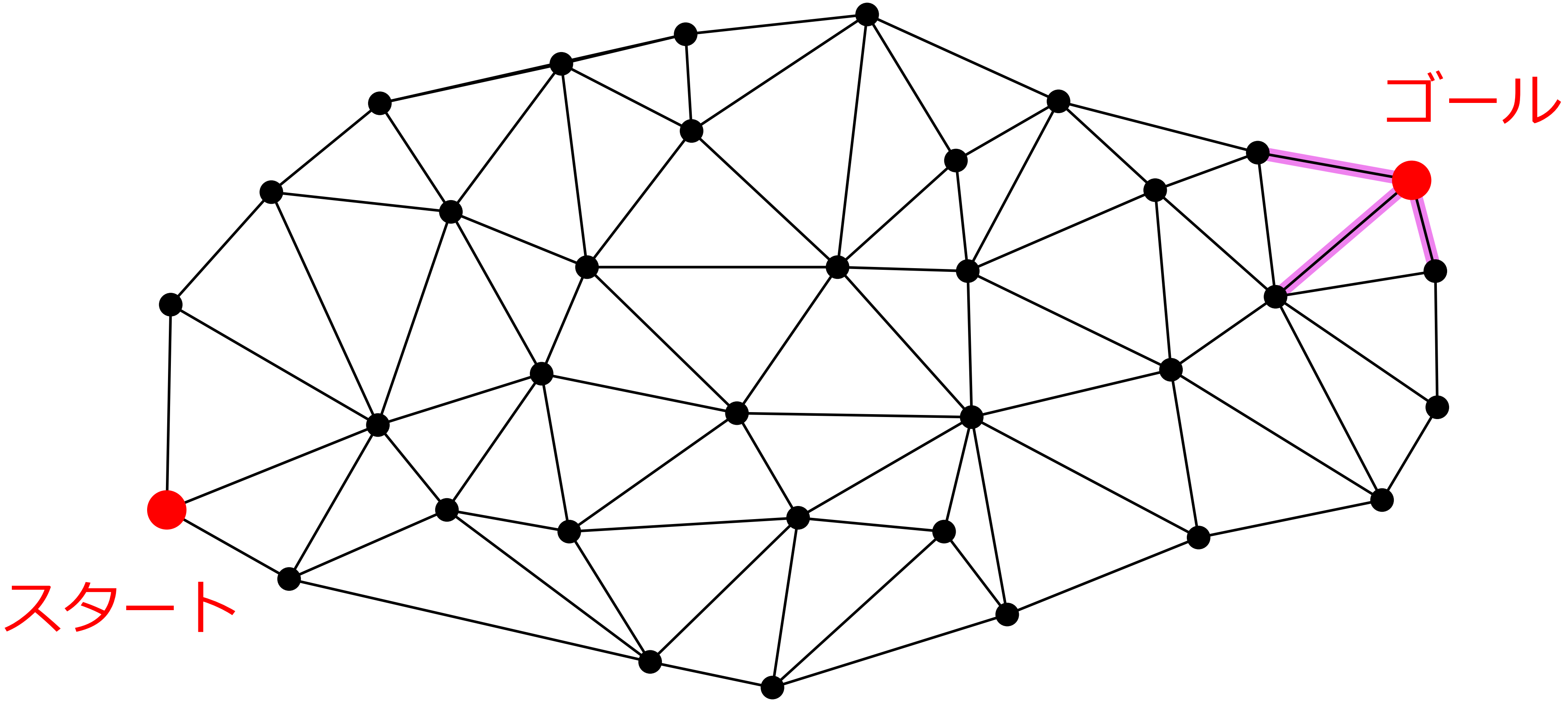
動的計画法の原理：アルゴリズム



最適性の原理 ： 最短経路の途中の点からの最短経路は
もとの最短経路をたどれば得られる

(principle of optimality)

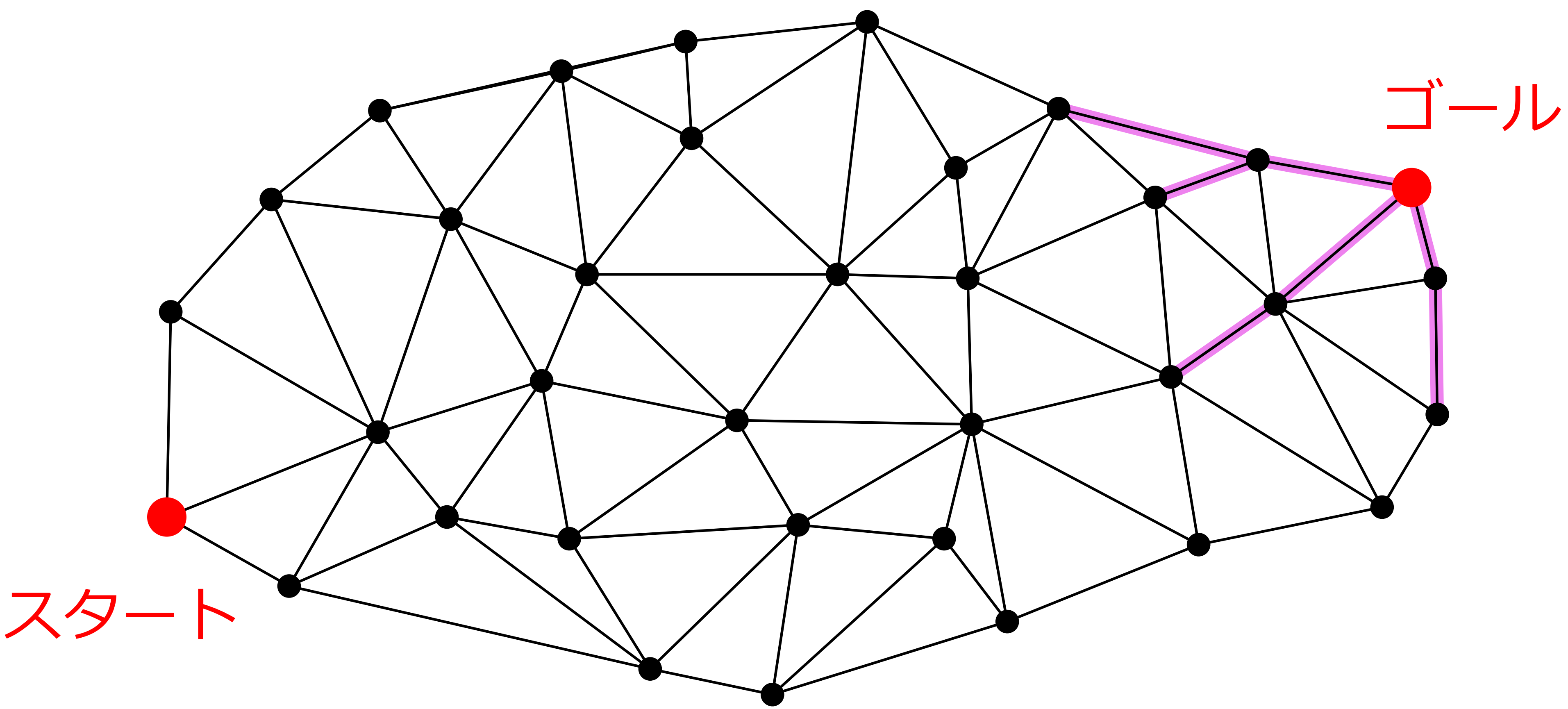
動的計画法の原理：アルゴリズム



最適性の原理 ： 最短経路の途中の点からの最短経路は
もとの最短経路をたどれば得られる

(principle of optimality)

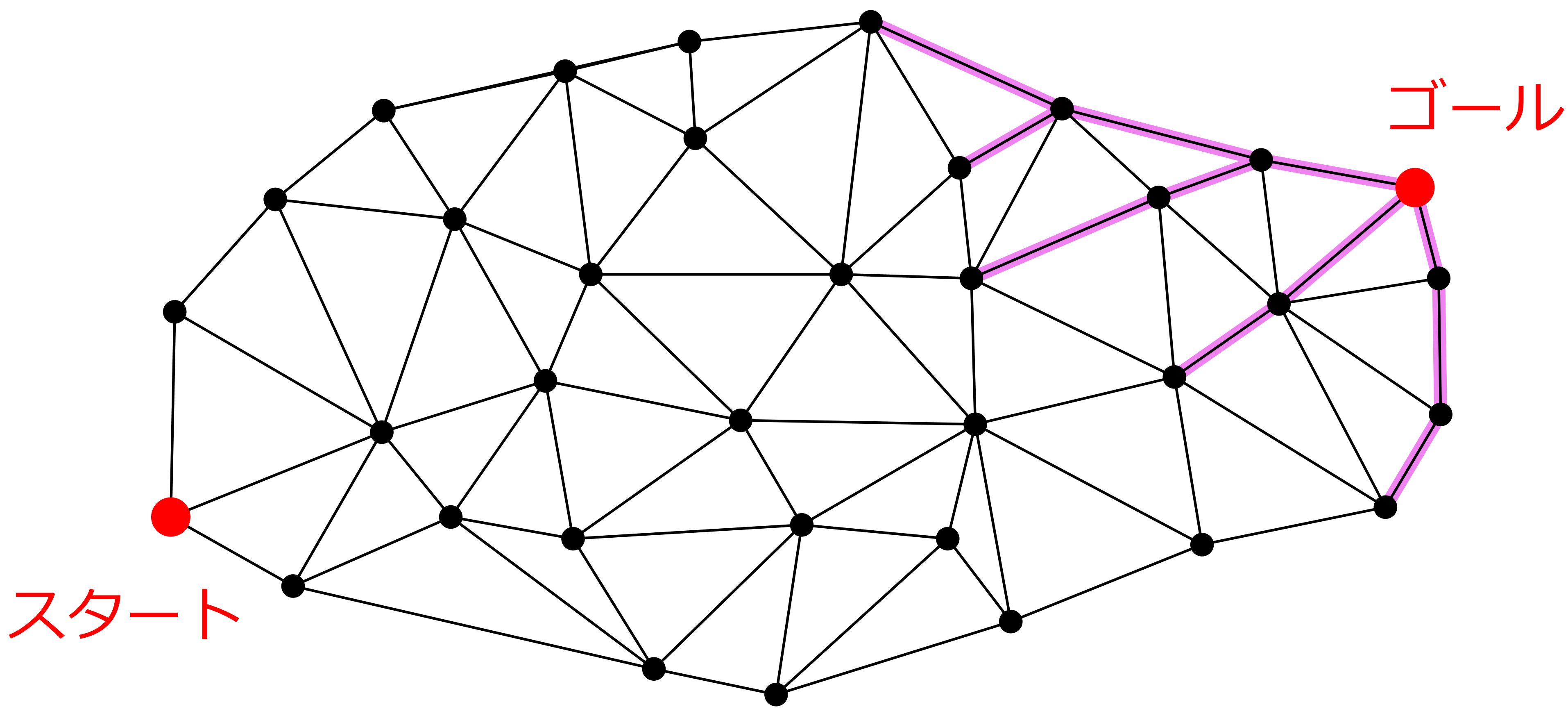
動的計画法の原理：アルゴリズム



最適性の原理 ： 最短経路の途中の点からの最短経路は
もとの最短経路をたどれば得られる

(principle of optimality)

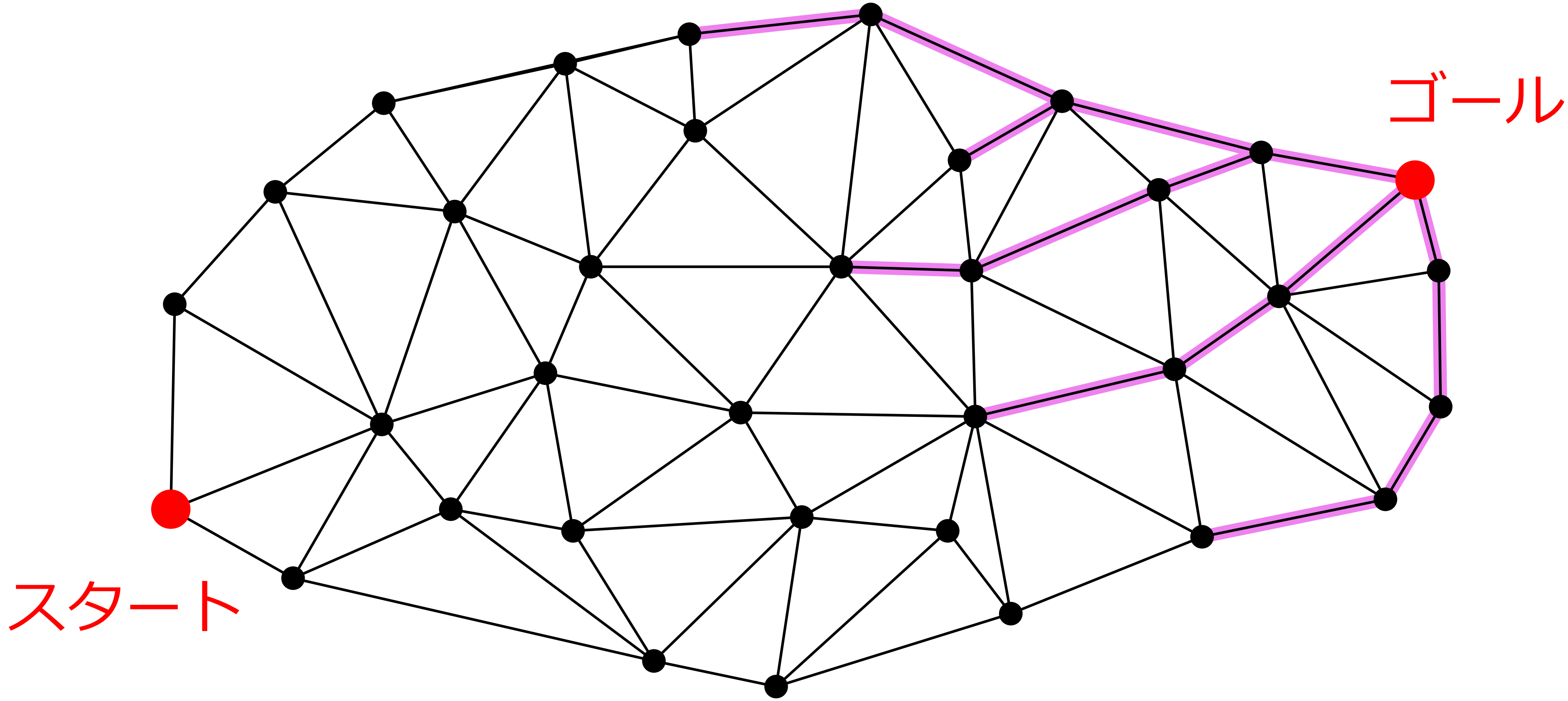
動的計画法の原理：アルゴリズム



最適性の原理 ： 最短経路の途中の点からの最短経路は
もとの最短経路をたどれば得られる

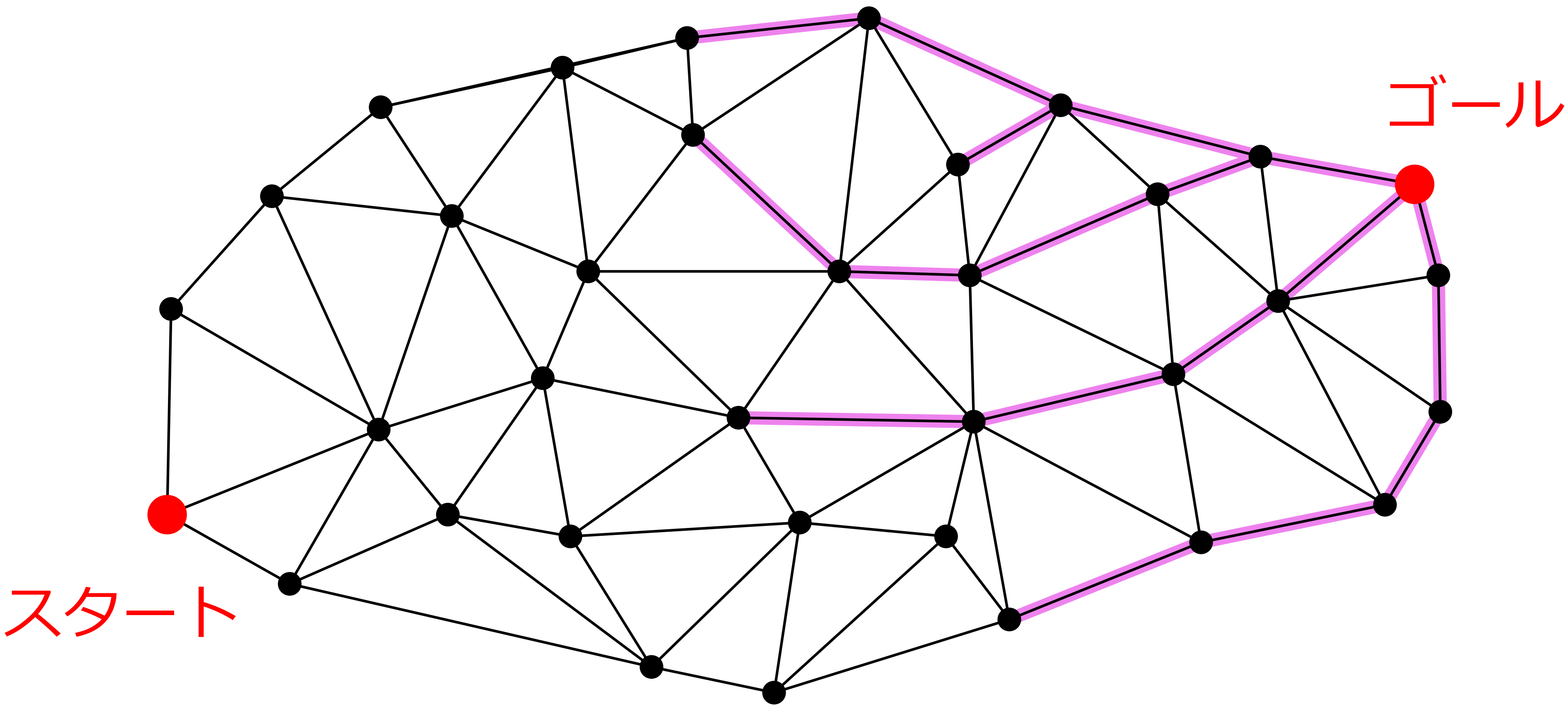
(principle of optimality)

動的計画法の原理：アルゴリズム



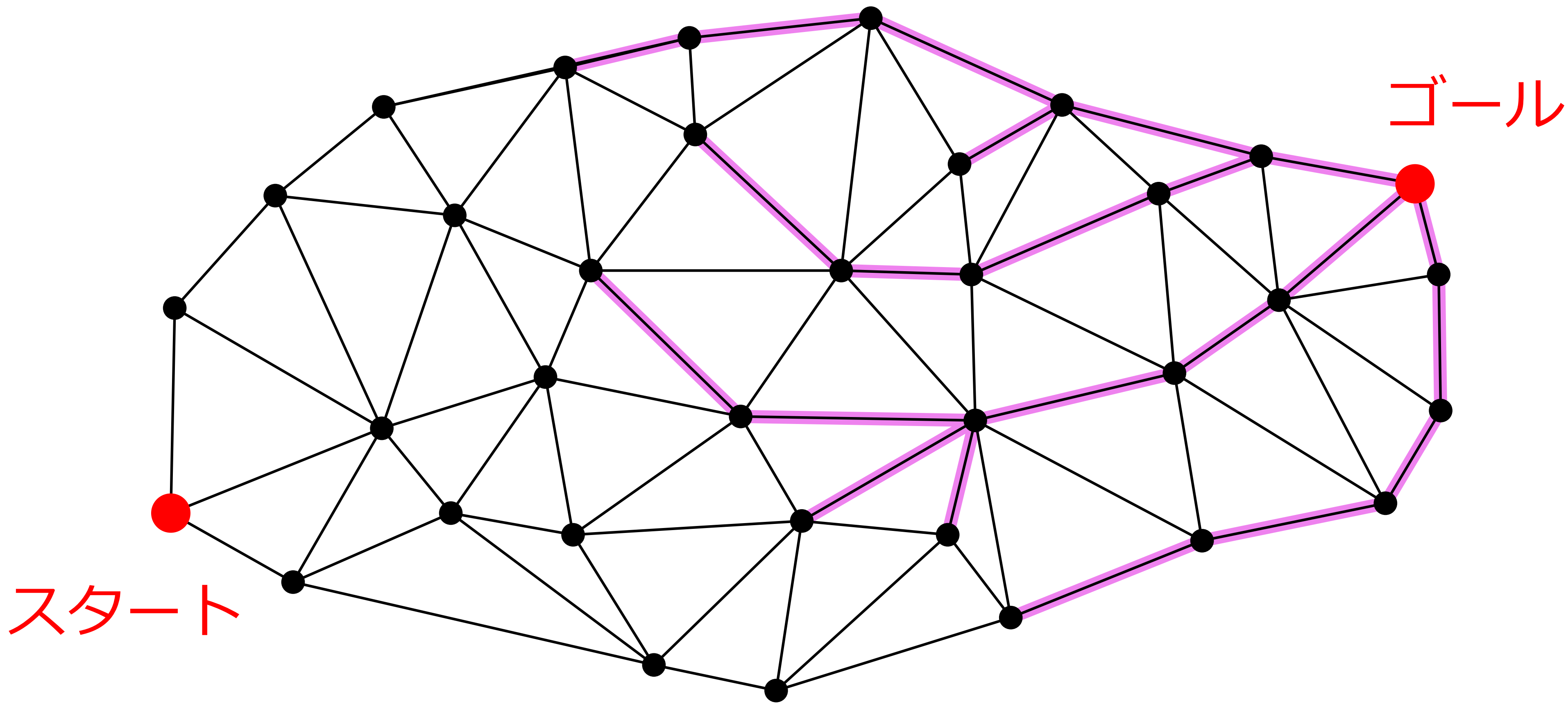
最適性の原理 ： 最短経路の途中の点からの最短経路は
もとの最短経路をたどれば得られる

(principle of optimality)



$\boxed{\text{最適性の原理}}$： 最短経路の途中の点からの最短経路は
もとの最短経路をたどれば得られる

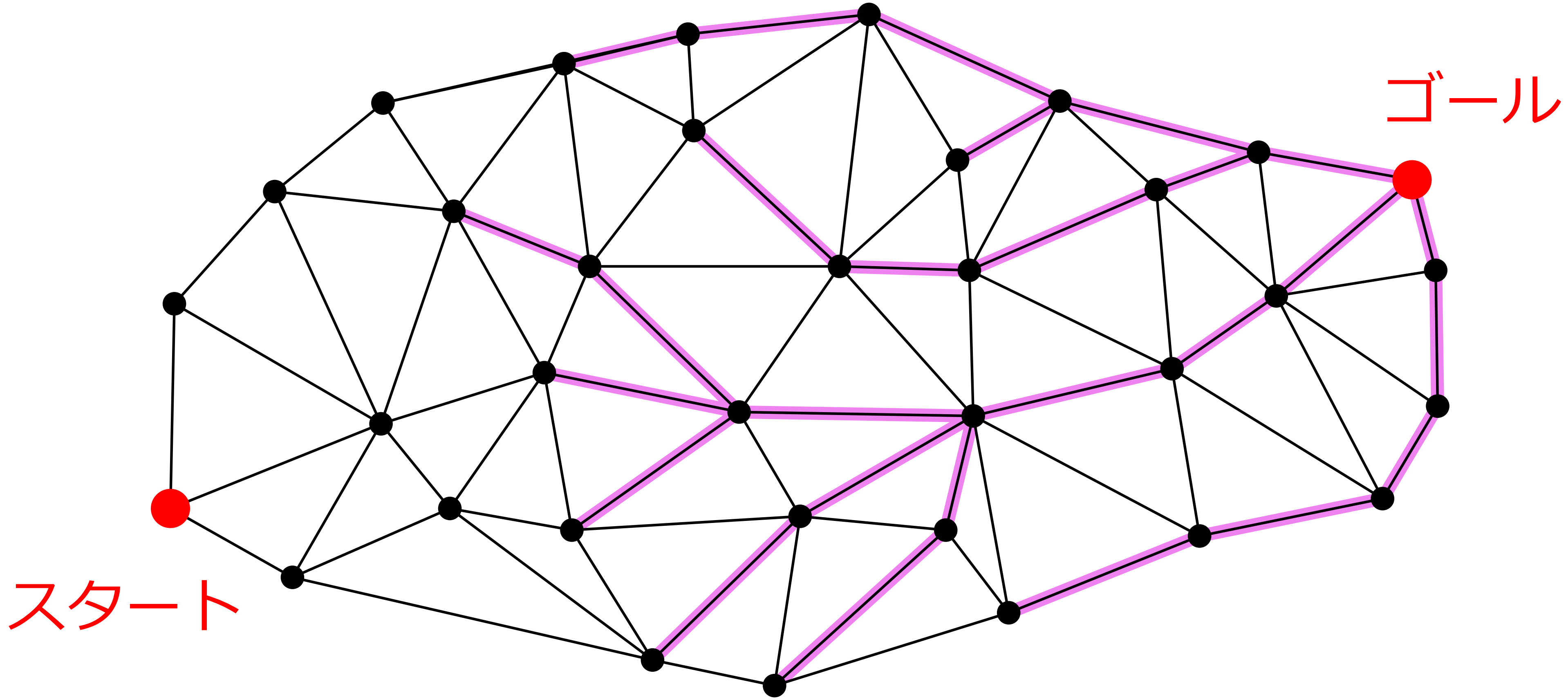
(principle of optimality)



$\boxed{\text{最適性の原理}}$ ： 最短経路 の途中の点からの最短経路は
もとの最短経路をたどれば得られる

(principle of optimality)

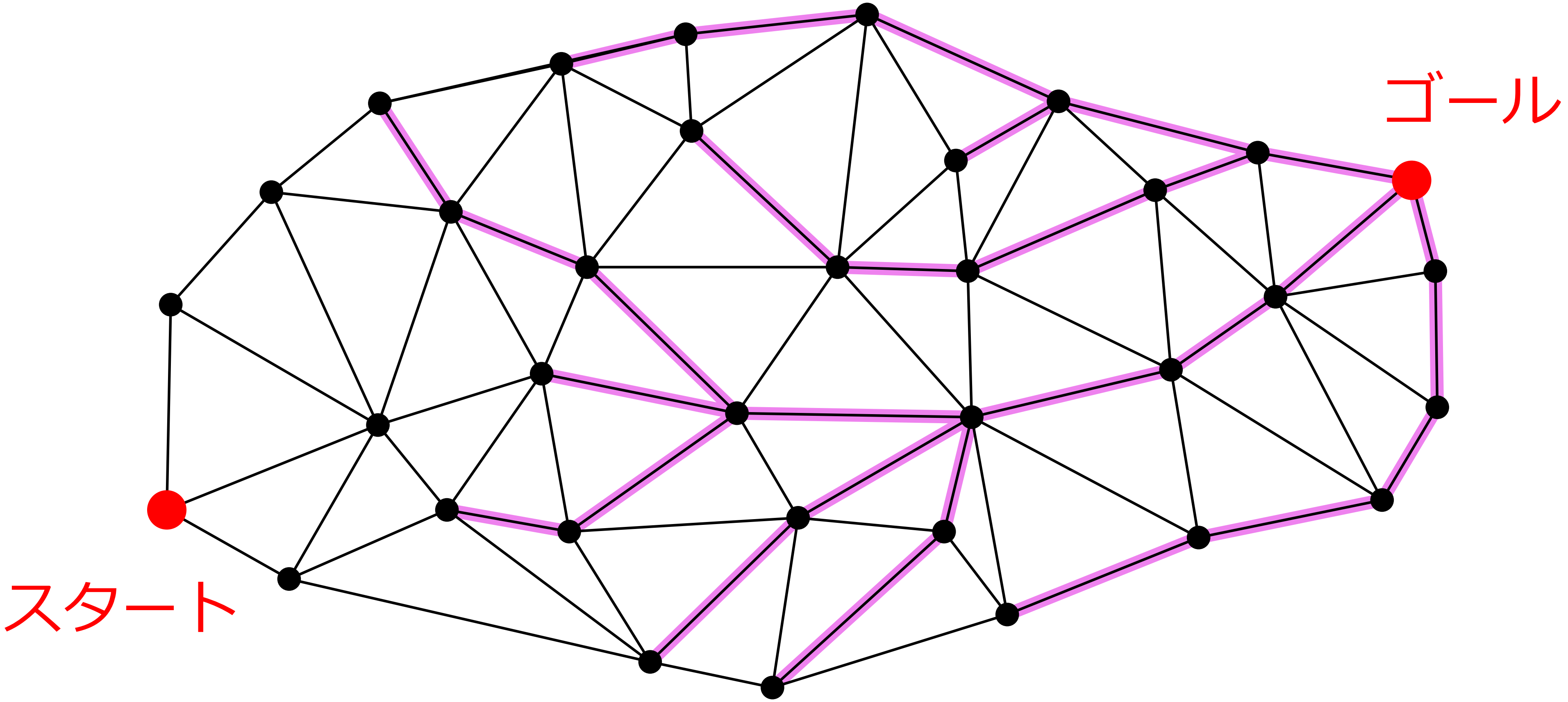
動的計画法の原理：アルゴリズム



最適性の原理 ： 最短経路の途中の点からの最短経路はもとの最短経路をたどれば得られる

(principle of optimality)

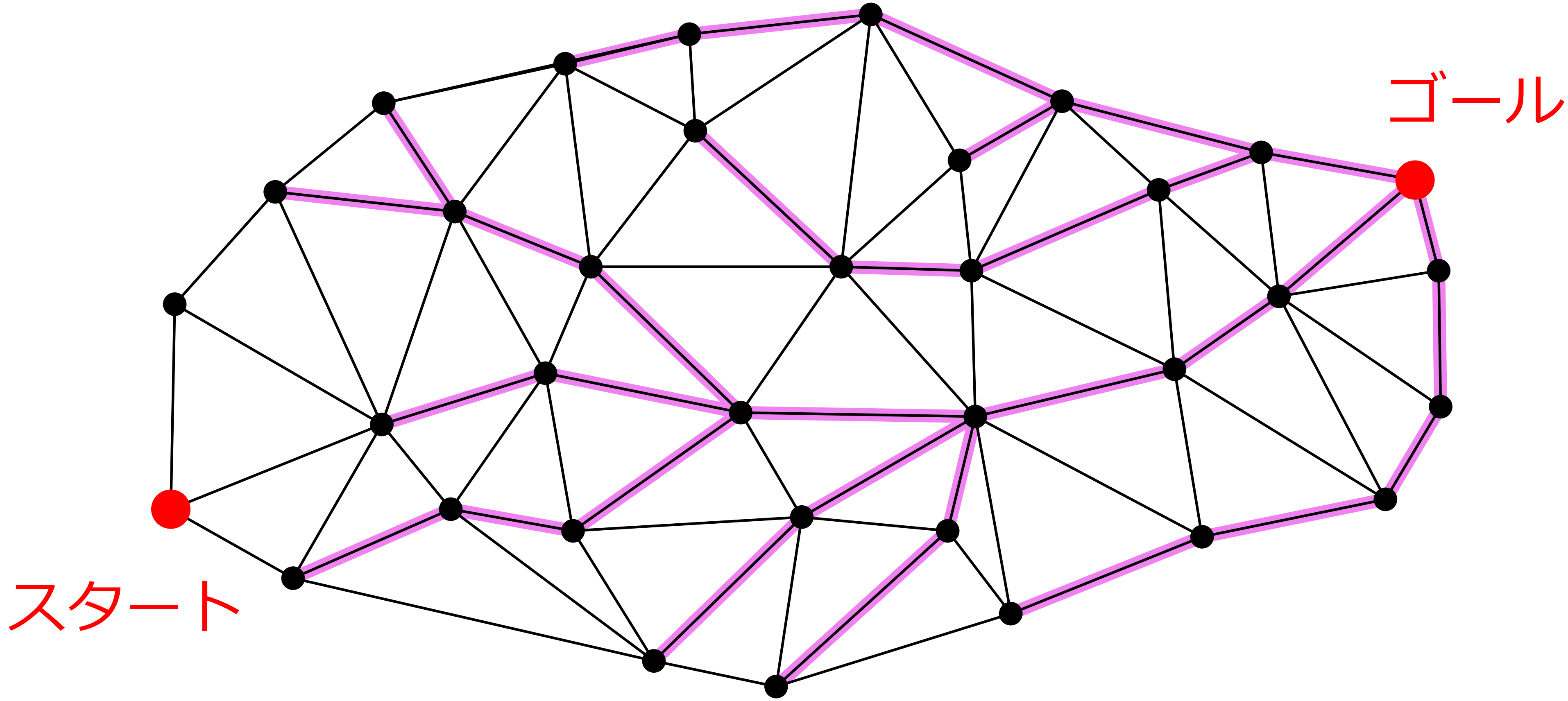
動的計画法の原理：アルゴリズム



最適性の原理 ： 最短経路 の途中の点からの最短経路は
もとの最短経路をたどれば得られる

(principle of optimality)

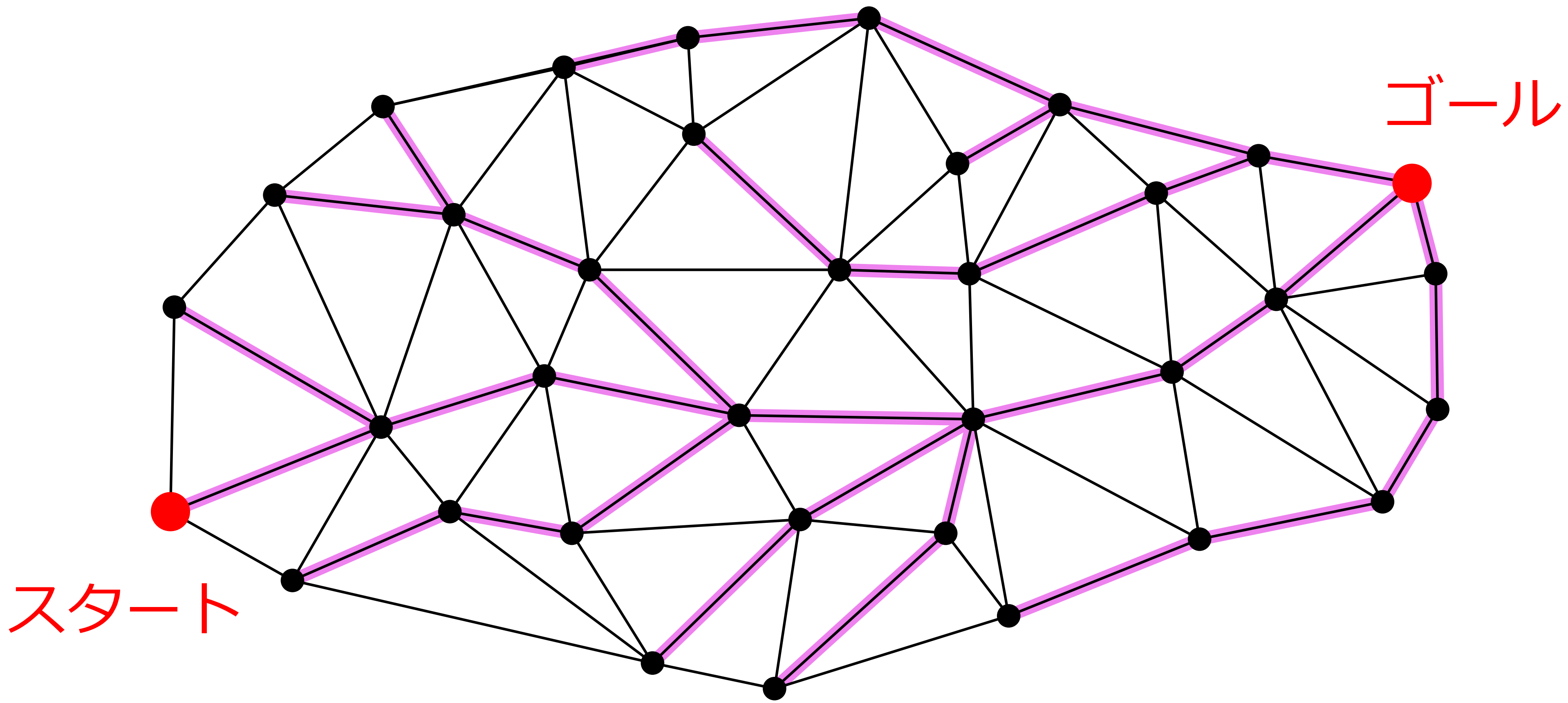
動的計画法の原理：アルゴリズム



最適性の原理 ： 最短経路の途中の点からの最短経路は
もとの最短経路をたどれば得られる

(principle of optimality)

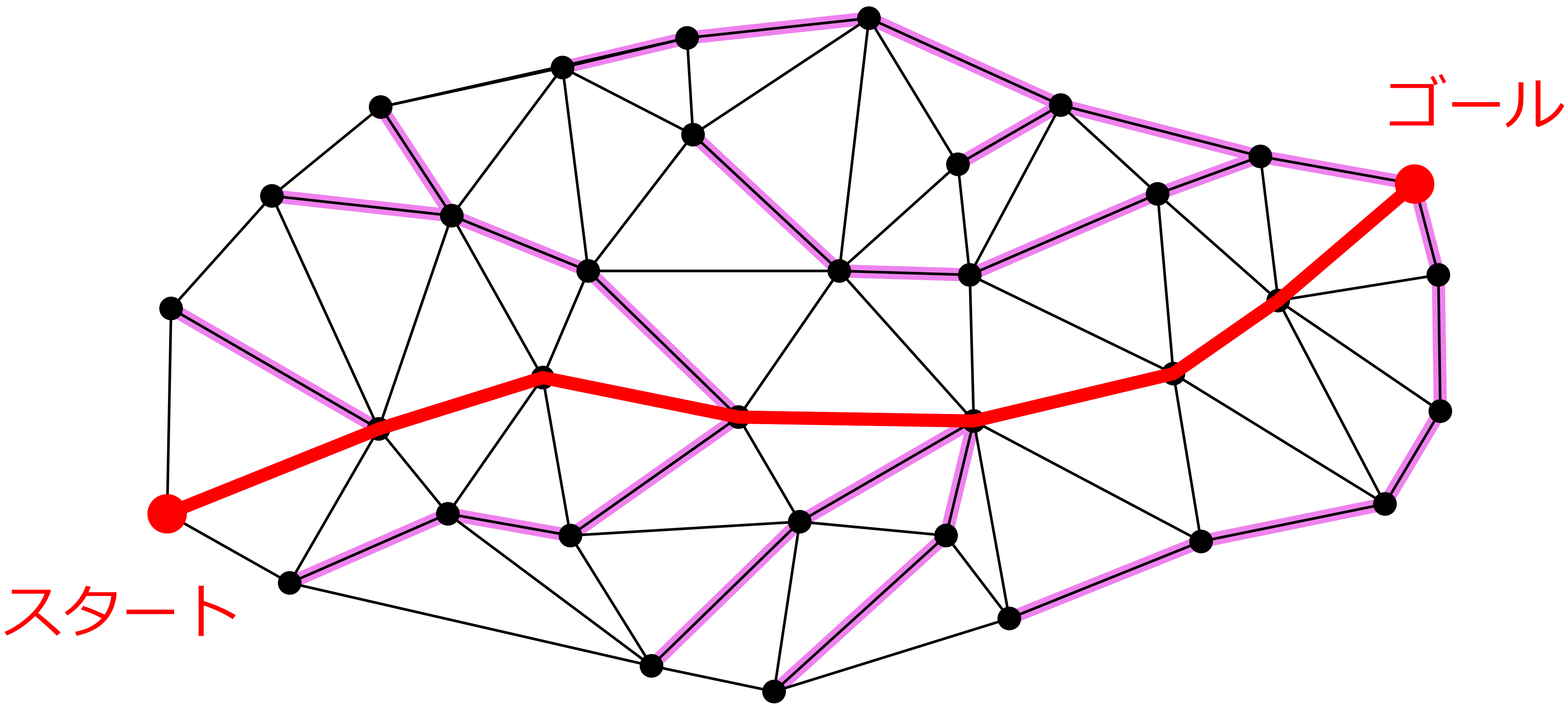
動的計画法の原理：アルゴリズム



最適性の原理 ： 最短経路の途中の点からの最短経路は
もとの最短経路をたどれば得られる

(principle of optimality)

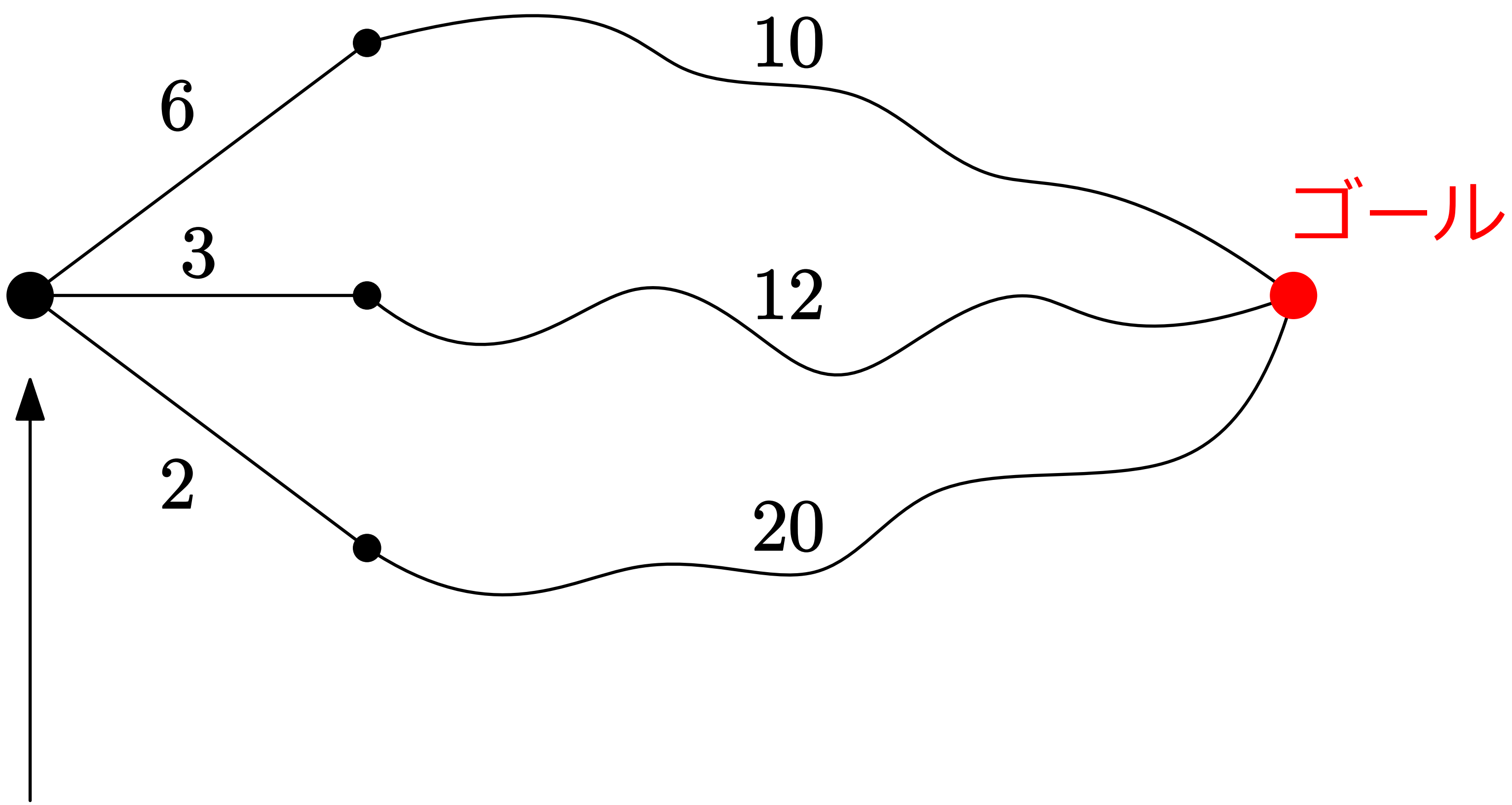
動的計画法の原理：アルゴリズム



$\boxed{\text{最適性の原理}}$ ： 最短経路の途中の点からの最短経路は
もとの最短経路をたどれば得られる

(principle of optimality)

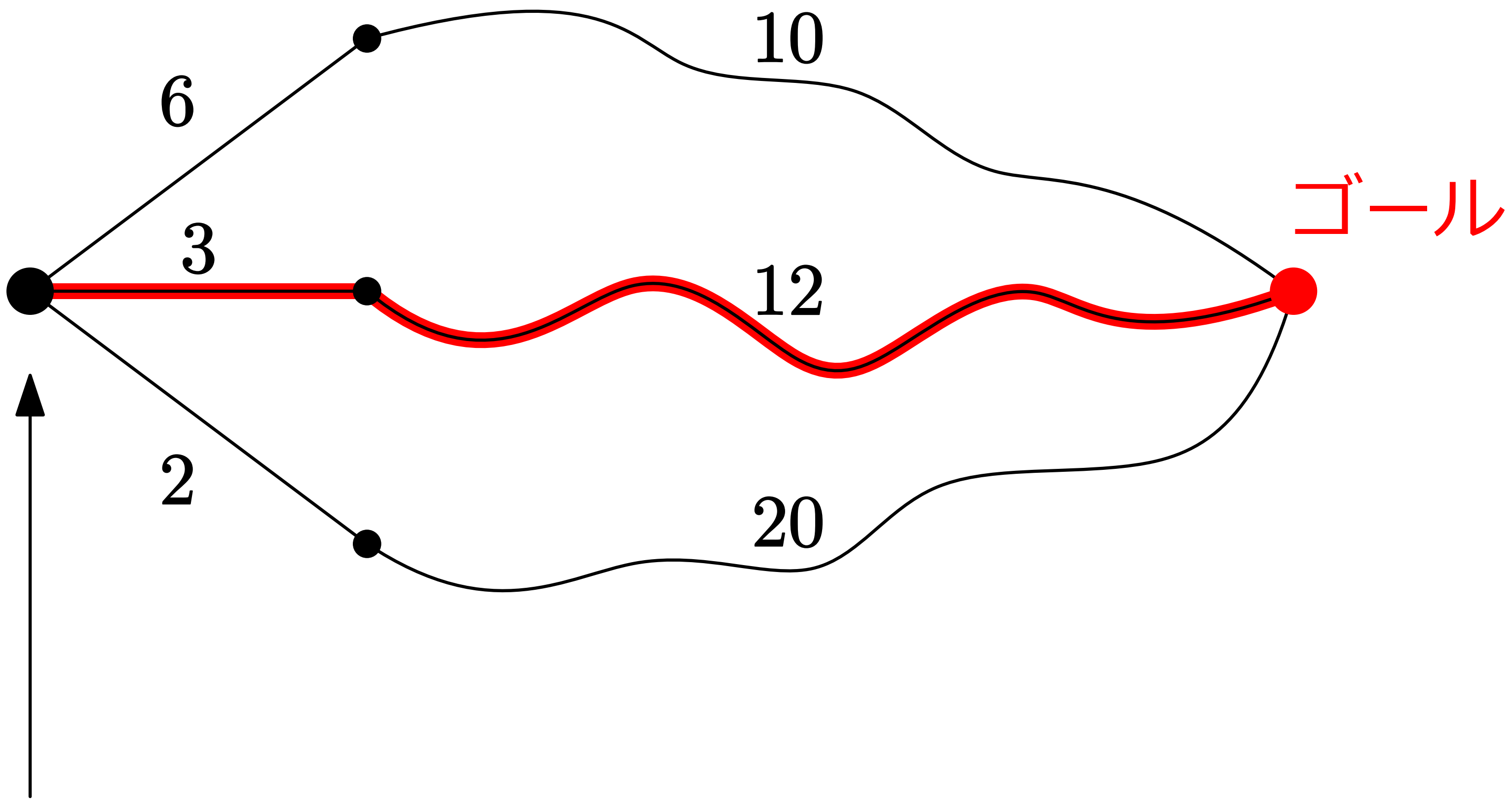
動的計画法の原理：アルゴリズム (続)



ここからゴールへの最短経路の長さ

$$= \min\{6+10, 3+12, 2+20\} = \min\{16, 15, 22\} = 15$$

動的計画法の原理：アルゴリズム (続)



ここからゴールへの最短経路の長さ

$$= \min\{6 + 10, 3 + 12, 2 + 20\} = \min\{16, 15, 22\} = 15$$

本日の内容

1. **$P2 \mid\mid C_{\max}$ と擬多項式時間動的計画法**
2. $1 \mid\mid \sum T_j$ と擬多項式時間動的計画法

$\boxed{\text{注}}$: 今日の授業では，処理時間はすべて整数とする

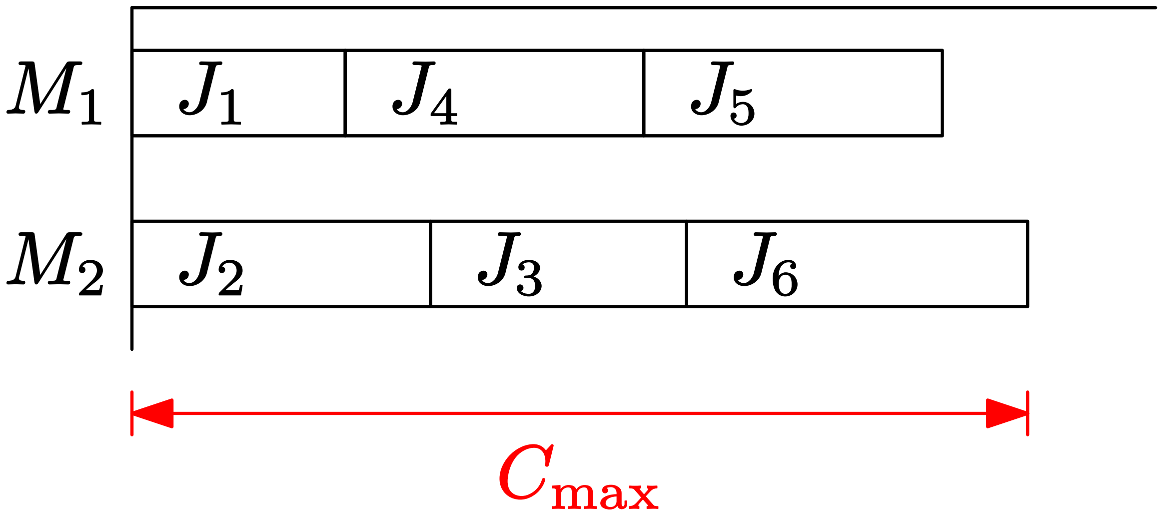
- R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

機械の環境

- 同一並列機械，機械数 $= 2$

最適化する目的

- 最大完了時刻の最小化



$P2 \mid\mid C_{\max}$：「自明」な解法

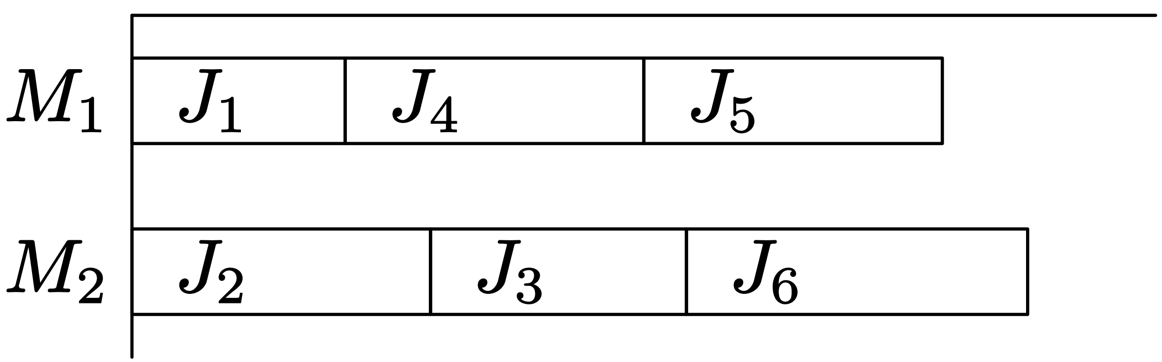
復習：$1 \mid\mid C_{\max}$ の最適解

どのように並べても，最大完了時刻は同じ

$\therefore P2 \mid\mid C_{\max}$ では，割当だけ気にすればよい

$P2 \mid\mid C_{\max}$ の「自明」な解法

すべての割当を考える



割当の総数 $=$ 2^n $\quad\leftarrow$ 膨大

定理 (Bellman ’57)

$P2 \mid\mid C_{\max}$ は $O\left(n \underline{\sum p_j}\right)$ 時間で解ける

処理時間の総和

比較			
	「自明」	$O(2^n n)$	指数時間
	動的計画法	$O(n \sum p_j)$	擬多項式時間 (詳細は次回)

$P2 \mid\mid C_{\max}$: 動的計画法

定理 (Bellman '57)

$P2 \mid\mid C_{\max}$ は $O\left(n \sum p_j\right)$ 時間で解ける

処理時間の総和

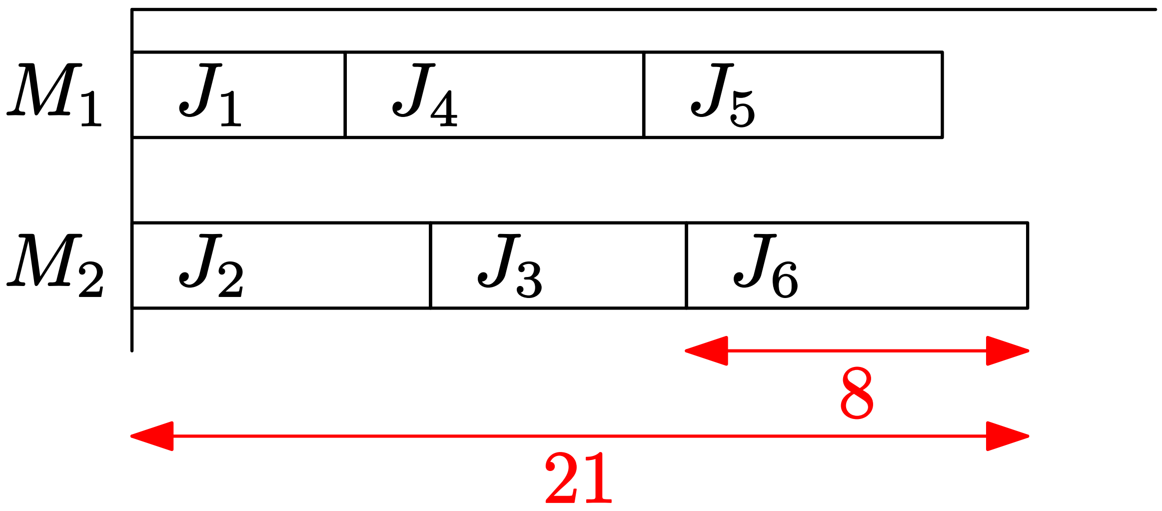
比較			
	「自明」	$O(2^n n)$	指数時間
	動的計画法	$O(n \sum p_j)$	擬多項式時間 (詳細は次回)

動的計画法を考えるときの鍵

1. 最適解の持つ **再帰的な構造** を見出す
2. 上の構造から，**状態** を適切に定義する
3. 状態の間の **再帰式** を立てる

最適解の構造

$\boxed{注}$：割当を決めることだけ 考えればよい

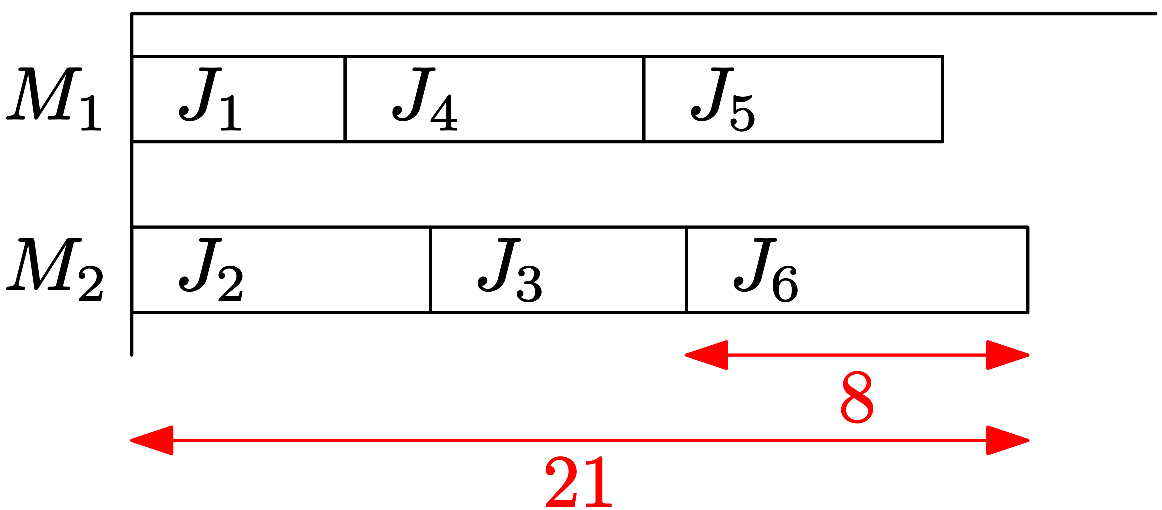


$J_1, \ldots, J_6$ に対する

最適スケジュール

最適解の構造

$\boxed{\text{注}}$：割当を決めることだけ 考えればよい



$J_1, \ldots, J_6$ に対する

最適スケジュール

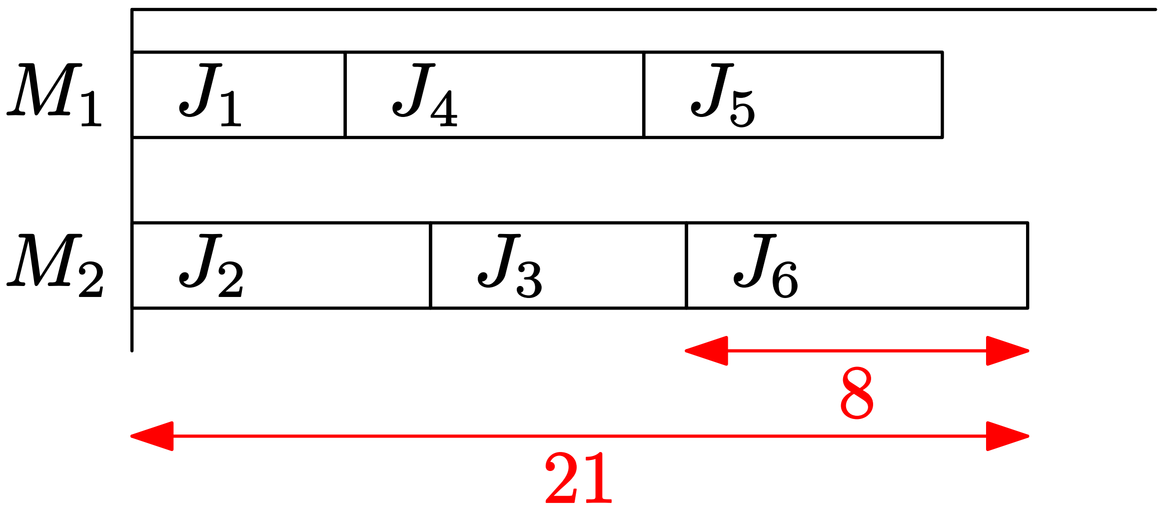
$\uparrow$

$J_1, \ldots, J_5$ に対する

最適スケジュール

最適解の構造

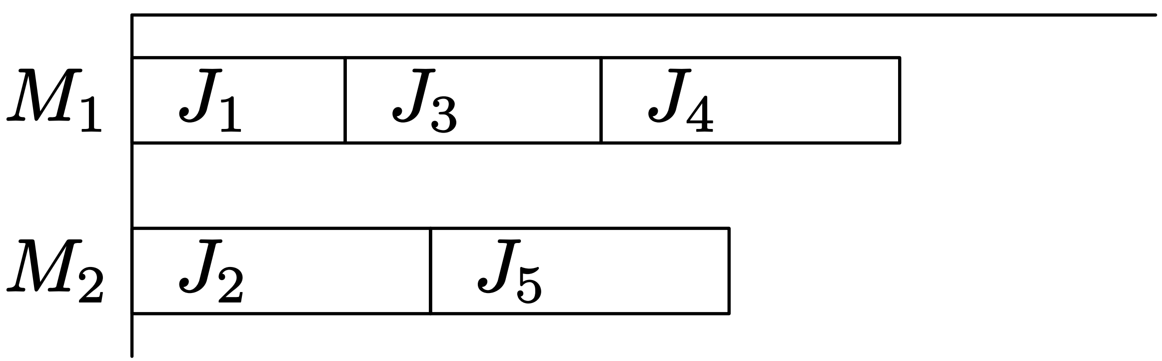
$\boxed{\text{注}}$: 割当を決めることだけ考えればよい



$J_1, \ldots, J_6$ に対する

最適スケジュール

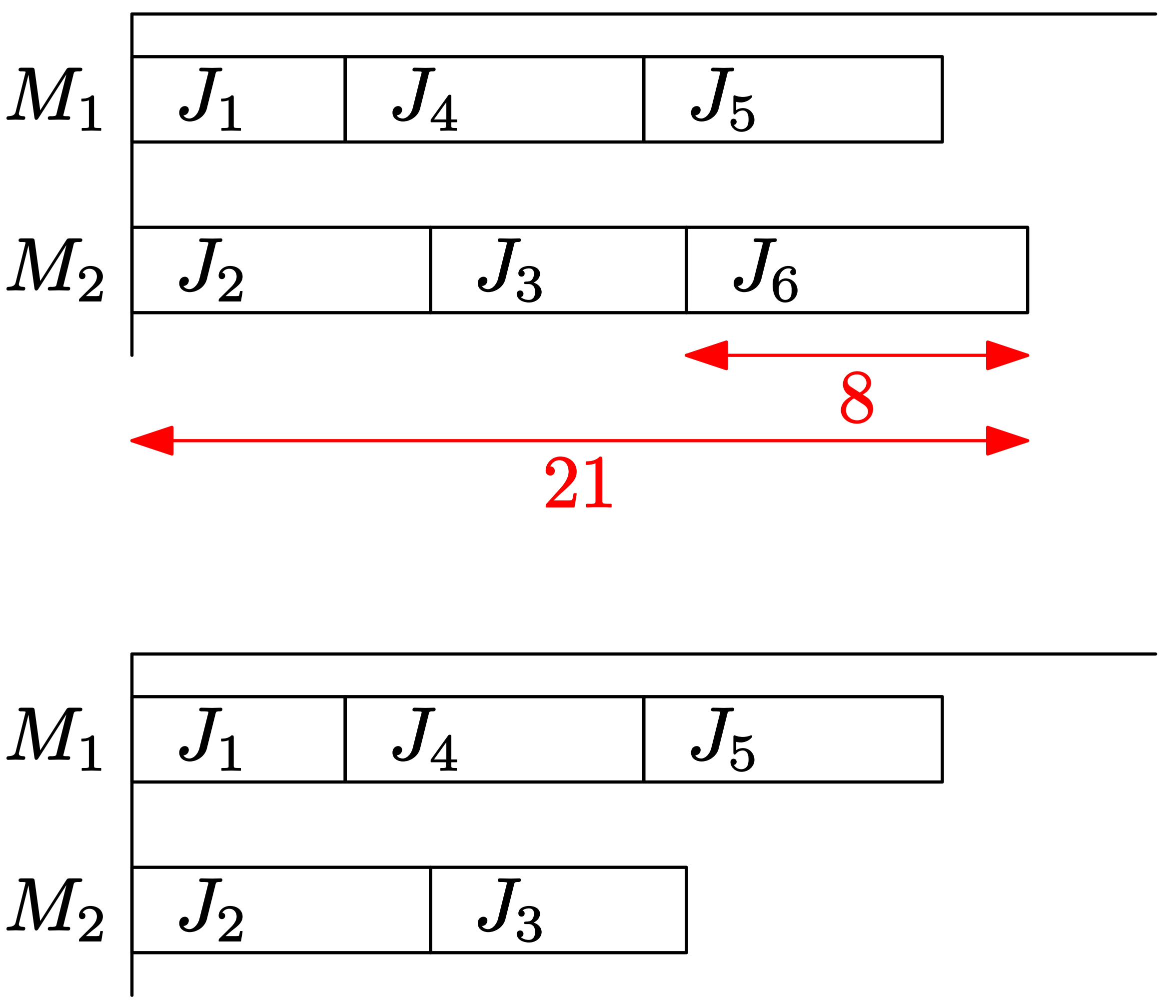
???



$J_1, \ldots, J_5$ に対する

最適スケジュール

$\boxed{\text{注}}$：割当を決めることだけ 考えればよい

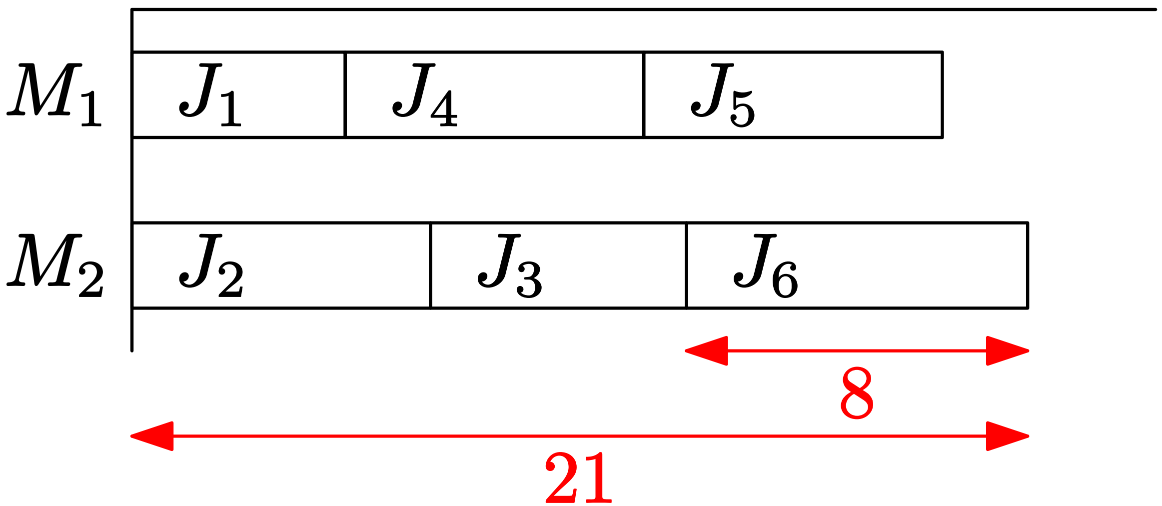


$J_1, \ldots, J_6$ に対する最適スケジュール

$J_1, \ldots, J_5$ に対するスケジュール

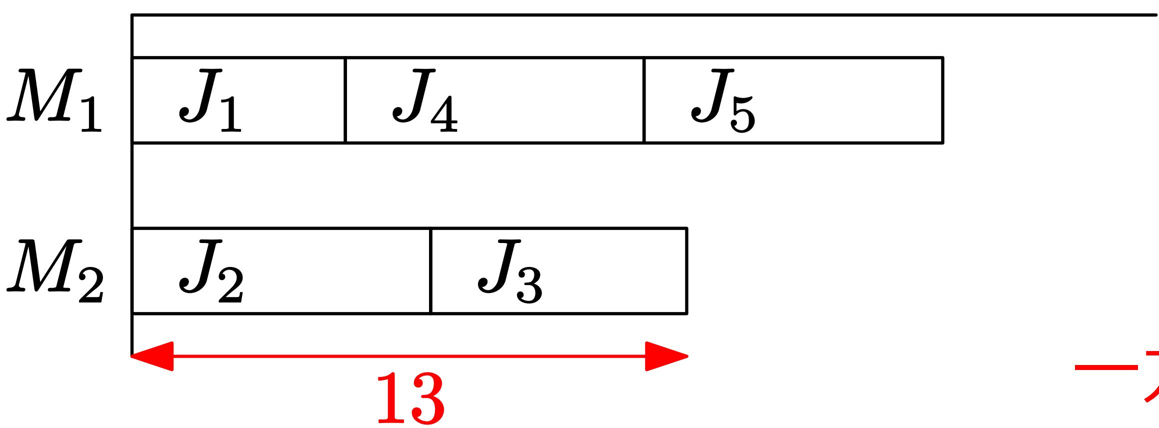
最適解の構造：続

注：割当を決めることだけ考えればよい



$J_1, \ldots, J_6$ に対する

最適スケジュール



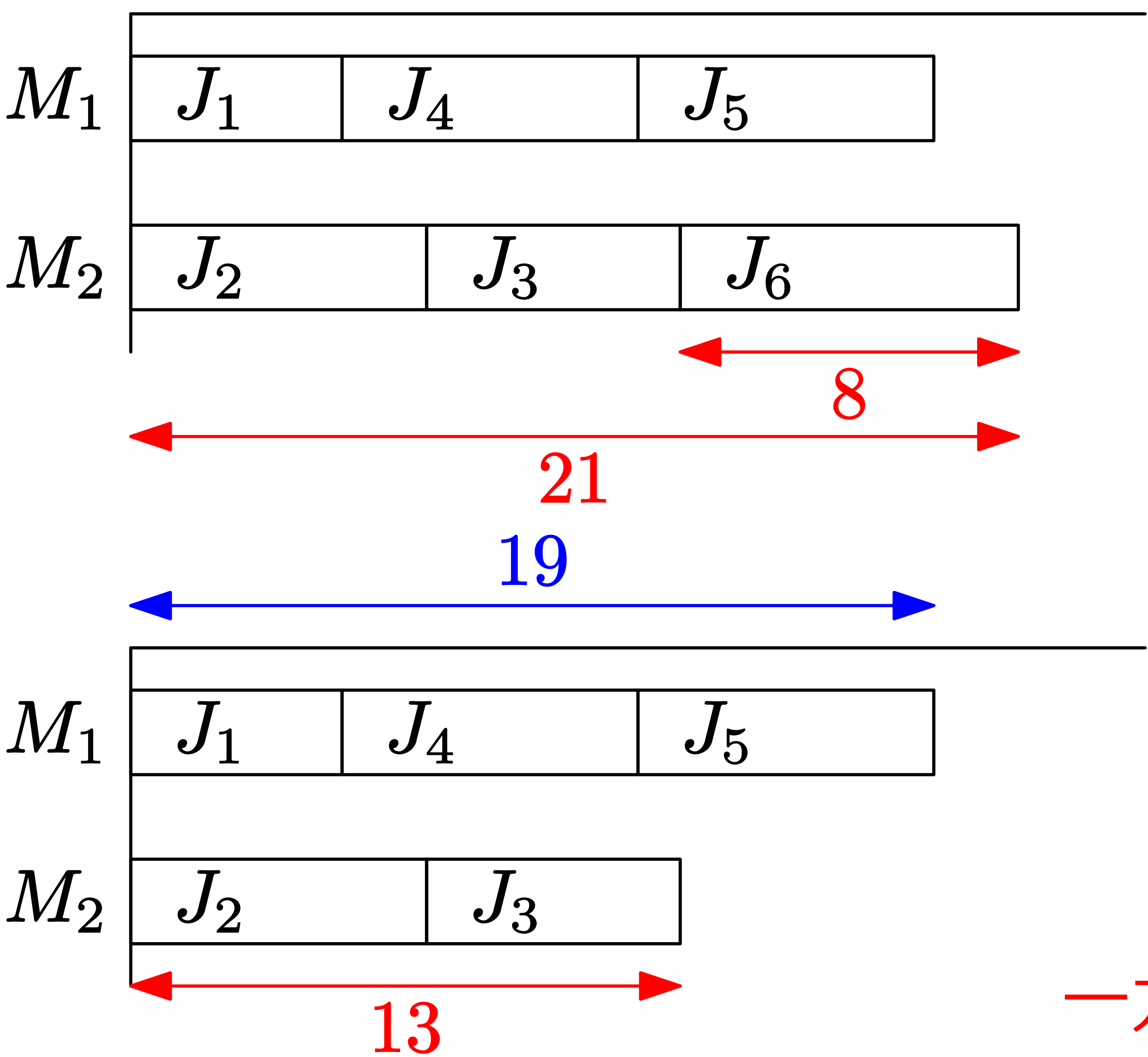
$J_1, \ldots, J_5$ に対する

スケジュール

一方の最終完了時刻が 13

最適解の構造：続

$\boxed{\text{注}}$：割当を決めることだけ 考えればよい



$J_1, \ldots, J_6$ に対する

最適スケジュール

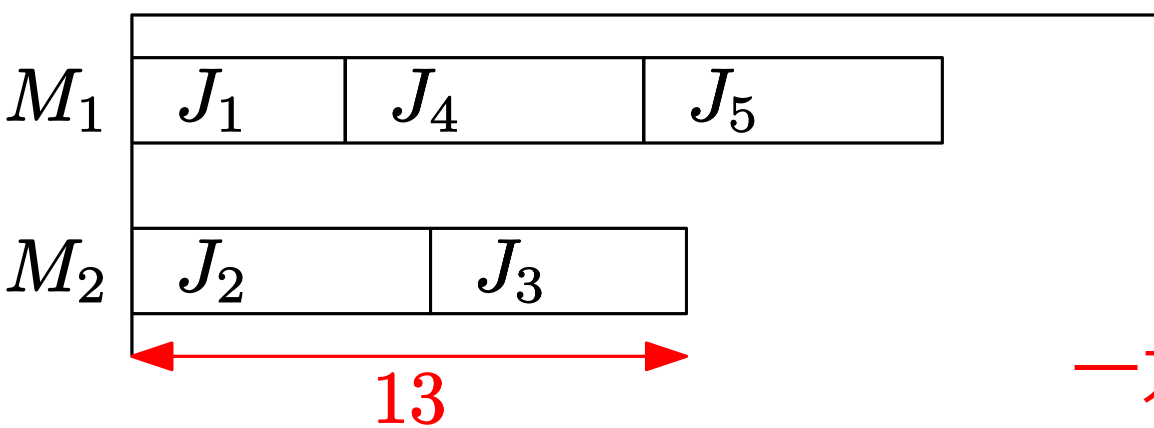
$J_1, \ldots, J_5$ に対する

スケジュール

一方の最終完了時刻が 13

状態の定義

$\boxed{\text{状態}}$: $(j, t) \quad (\forall\, j \in \{0\} \cup [n], \forall\, t \in \{0\} \cup [\sum p_j])$



$J_1, \ldots, J_5$ に対する

スケジュール

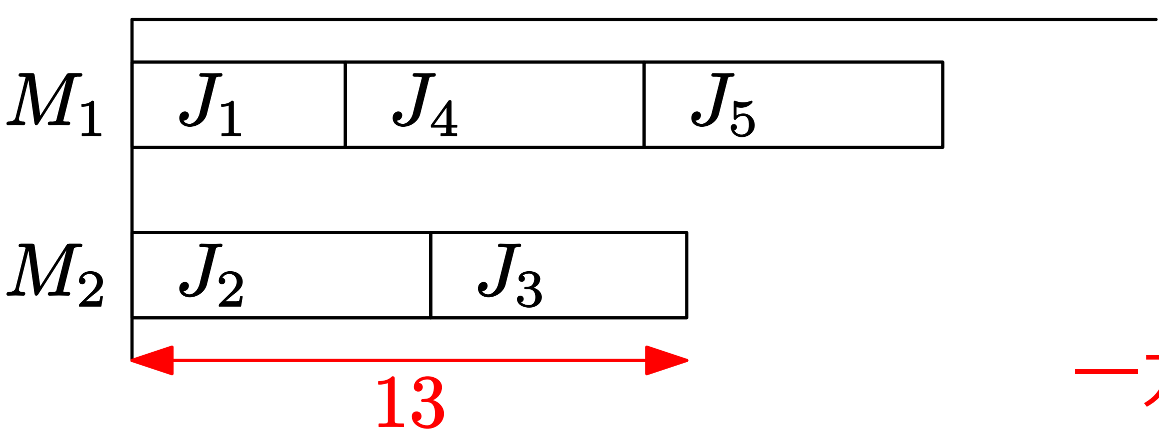
一方の最終完了時刻が 13

状態の定義

$\boxed{\text{状態}}$: $(j, t) \quad (\forall\, j \in \{0\} \cup [n], \forall\, t \in \{0\} \cup [\sum p_j])$

$\boxed{\text{状態の値}}$: $j \in \{0\} \cup [n], t \in \{0\} \cup [\sum p_j]$ に対して

$$f(j,t) = \begin{cases} \textsf{true} & J_1, \ldots, J_j \text{ に対するスケジュールで，} \\ & \text{一方の機械の最大完了時刻が } t \text{ のものが存在,} \\ \textsf{false} & \text{それが非存在} \end{cases}$$



$J_1, \ldots, J_5$ に対するスケジュール

一方の最終完了時刻が 13

再帰式 (Bellman 方程式)

$$f(0,0) = \text{true}$$

$$f(0,t) = \text{false} \quad \forall\, t > 0$$

$\boxed{\text{状態}}$: $(j,t) \quad (\forall\, j \in \{0\} \cup [n], \forall\, t \in \{0\} \cup [\sum p_j])$

$\boxed{\text{状態の値}}$: $j \in \{0\} \cup [n], t \in \{0\} \cup [\sum p_j]$ に対して

$$f(j,t) = \begin{cases} \text{true} & J_1, \ldots, J_j \text{ に対するスケジュールで,} \\ & \text{一方の機械の最大完了時刻が } t \text{ のものが存在,} \\ \text{false} & \text{それが非存在} \end{cases}$$

再帰式 (Bellman 方程式)

$$f(0,0) = \text{true}$$

$$f(0,t) = \text{false} \quad \forall\, t > 0$$

$j \geq 1$ のとき

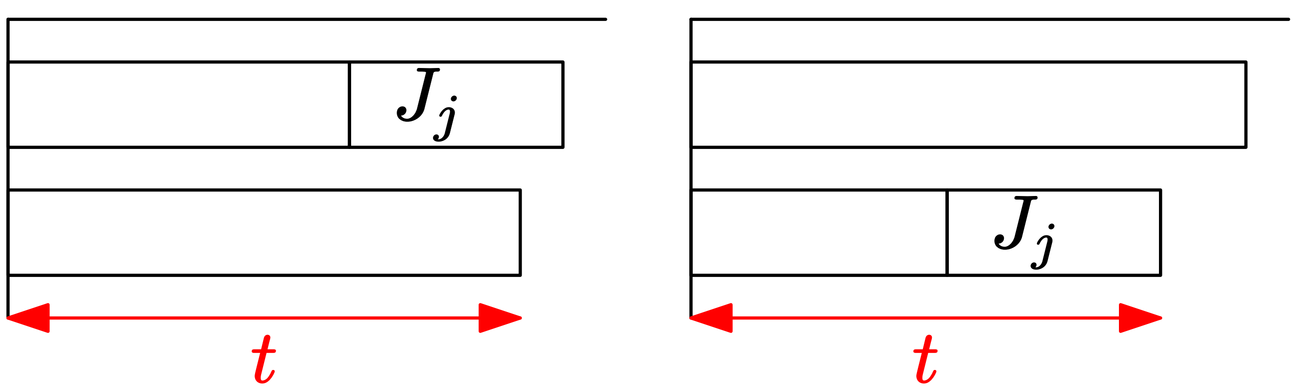
$\boxed{\text{状態}}$: (j,t) $\quad (\forall\, j \in \{0\} \cup [n], \forall\, t \in \{0\} \cup [\sum p_j])$

$\boxed{\text{状態の値}}$: $j \in \{0\} \cup [n], t \in \{0\} \cup [\sum p_j]$ に対して

$$f(j,t) = \begin{cases} \text{true} & J_1, \ldots, J_j \text{ に対するスケジュールで,} \\ & \text{一方の機械の最大完了時刻が } t \text{ のものが存在,} \\ \text{false} & \text{それが非存在} \end{cases}$$

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

論理和「または」



動的計画法の動き：例 1

ポイント ：再帰式は j, t が **小さい方から** 使っていく

$j \backslash t$	0	1	2	3	4	5
0						
1						
2						
3						
4						

J_1
1

J_2
1

J_3
2

J_4
1

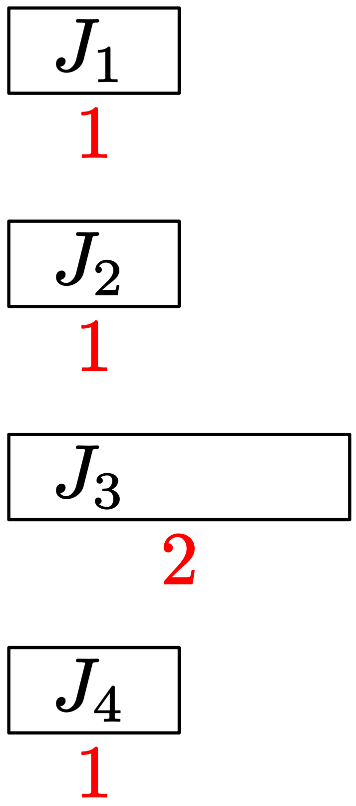
動的計画法の動き：例 1

$\boxed{\text{ポイント}}$：再帰式は j, t が **小さい方から** 使っていく

$$f(0,0) = \mathsf{true}$$

$$f(0,t) = \mathsf{false} \quad \forall\, t > 0$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1						
2						
3						
4						



動的計画法の動き：例1

$\boxed{\text{ポイント}}$：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

j \ t	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T					
2						
3						
4						

J_1: 1

J_2: 1

J_3: 2

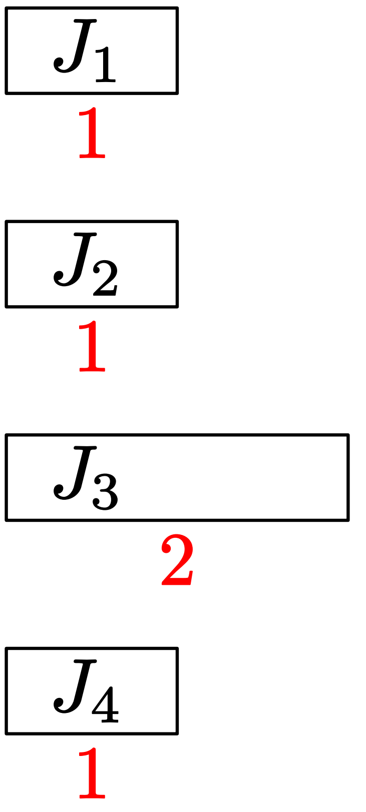
J_4: 1

動的計画法の動き：例 1

ポイント：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \lor f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T				
2						
3						
4						



動的計画法の動き：例 1

ポイント：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F			
2						
3						
4						

J_1: 1

J_2: 1

J_3: 2

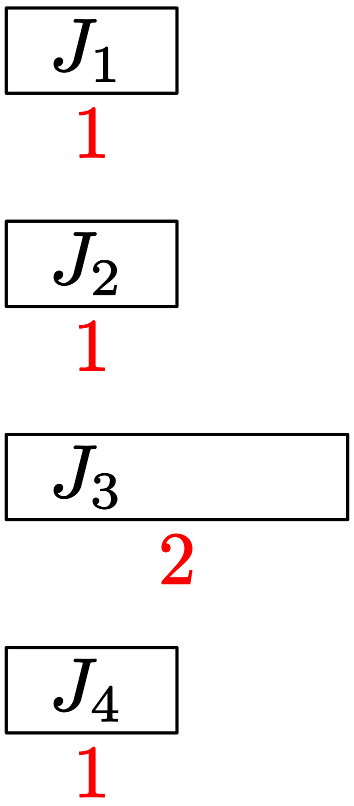
J_4: 1

動的計画法の動き：例1

ポイント：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \ge p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F		
2						
3						
4						

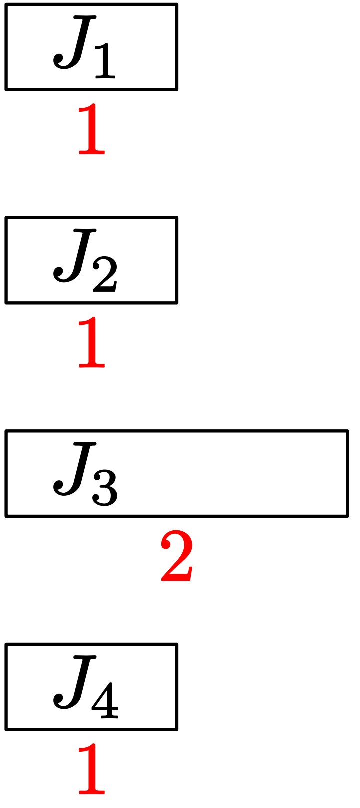


動的計画法の動き：例 1

ポイント：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	
2						
3						
4						

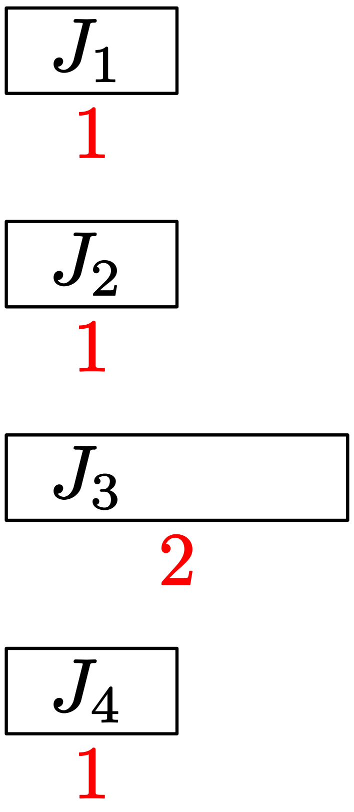


動的計画法の動き：例 1

$\boxed{\text{ポイント}}$：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

j \ t	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2						
3						
4						

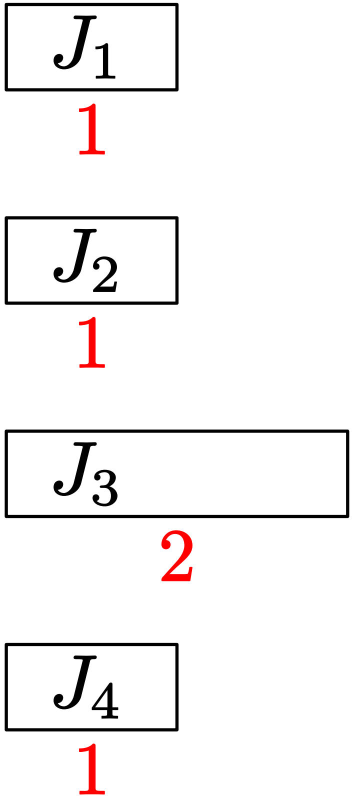


動的計画法の動き：例 1

ポイント ：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2	T					
3						
4						



動的計画法の動き：例 1

$\boxed{\text{ポイント}}$：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2	T	T				
3						
4						

J_1: 1

J_2: 1

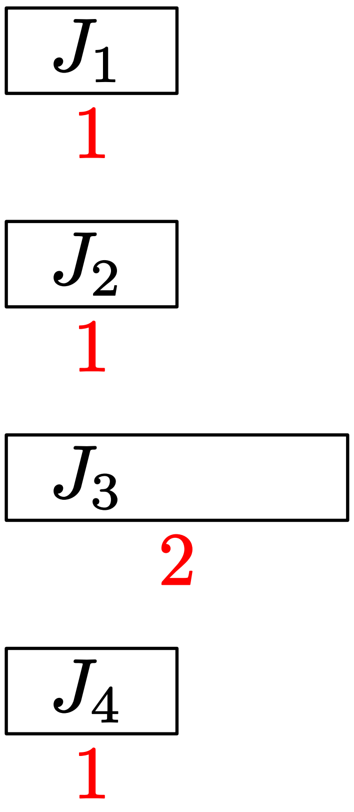
J_3: 2

J_4: 1

ポイント：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \lor f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2	T	T	T			
3						
4						



動的計画法の動き：例 1

ポイント：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2	T	T	T	F		
3						
4						

J_1

1

J_2

1

J_3

2

J_4

1

動的計画法の動き：例 1

$\boxed{\text{ポイント}}$：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2	T	T	T	F	F	
3						
4						

J_1 1

J_2 1

J_3 2

J_4 1

動的計画法の動き：例 1

ポイント：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2	T	T	T	F	F	F
3						
4						

J_1: 1

J_2: 1

J_3: 2

J_4: 1

動的計画法の動き：例 1

$\boxed{\text{ポイント}}$：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2	T	T	T	F	F	F
3	T					
4						

J_1	1
J_2	1
J_3	2
J_4	1

動的計画法の動き：例 1

ポイント：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \lor f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

j \ t	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2	T	T	T	F	F	F
3	T	T				
4						

J_1: 1

J_2: 1

J_3: 2

J_4: 1

動的計画法の動き：例1

$\boxed{\text{ポイント}}$：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

j \ t	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2	T	T	T	F	F	F
3	T	T	T			
4						

J_1: 1

J_2: 1

J_3: 2

J_4: 1

動的計画法の動き：例 1

$\boxed{\text{ポイント}}$：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

j \ t	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2	T	T	T	F	F	F
3	T	T	T	T		
4						

Job	p_j
J_1	1
J_2	1
J_3	2
J_4	1

動的計画法の動き：例 1

$\boxed{\text{ポイント}}$：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2	T	T	T	F	F	F
3	T	T	T	T	T	
4						

J_1: 1

J_2: 1

J_3: 2

J_4: 1

ポイント：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2	T	T	T	F	F	F
3	T	T	T	T	T	F
4						

J_1 1

J_2 1

J_3 2

J_4 1

動的計画法の動き：例1

ポイント：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2	T	T	T	F	F	F
3	T	T	T	T	T	F
4	T					

J_1 1

J_2 1

J_3 2

J_4 1

動的計画法の動き：例 1

ポイント：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2	T	T	T	F	F	F
3	T	T	T	T	T	F
4	T	T				

J_1: 1

J_2: 1

J_3: 2

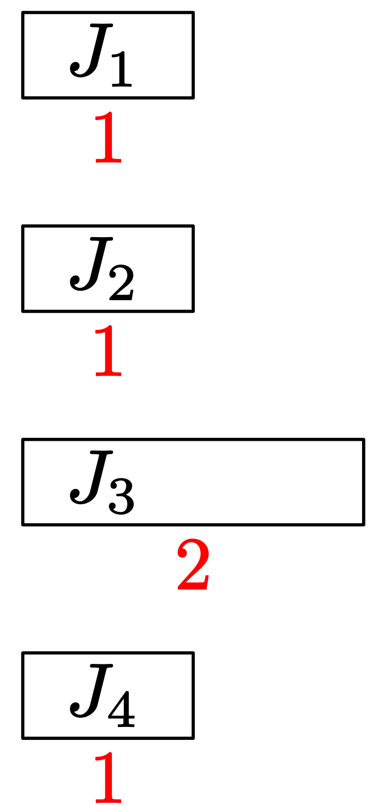
J_4: 1

動的計画法の動き：例1

$\boxed{\text{ポイント}}$：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2	T	T	T	F	F	F
3	T	T	T	T	T	F
4	T	T	T			

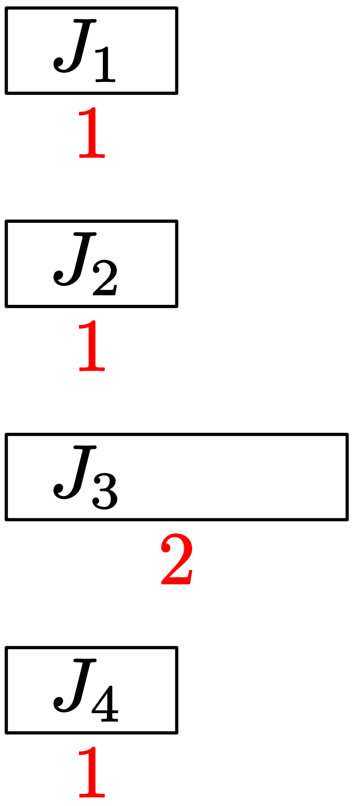


動的計画法の動き：例 1

$\boxed{\text{ポイント}}$：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2	T	T	T	F	F	F
3	T	T	T	T	T	F
4	T	T	T	T		



動的計画法の動き：例 1

ポイント ：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \vee f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2	T	T	T	F	F	F
3	T	T	T	T	T	F
4	T	T	T	T	T	

J_1 1

J_2 1

J_3 2

J_4 1

動的計画法の動き：例 1

$\boxed{\text{ポイント}}$ ：再帰式は j, t が **小さい方から** 使っていく

$$f(j,t) = \begin{cases} f(j-1,t) \lor f(j-1,t-p_j) & (t \geq p_j), \\ f(j-1,t) & (t < p_j) \end{cases}$$

$j \backslash t$	0	1	2	3	4	5
0	T	F	F	F	F	F
1	T	T	F	F	F	F
2	T	T	T	F	F	F
3	T	T	T	T	T	F
4	T	T	T	T	T	T

$\boxed{J_1}$ 1

$\boxed{J_2}$ 1

$\boxed{J_3}$ 2

$\boxed{J_4}$ 1

動的計画法の動き：例 2

$p = (3, 9, 12, 7)$ の場合 $(\sum p_j = 31)$

T F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F

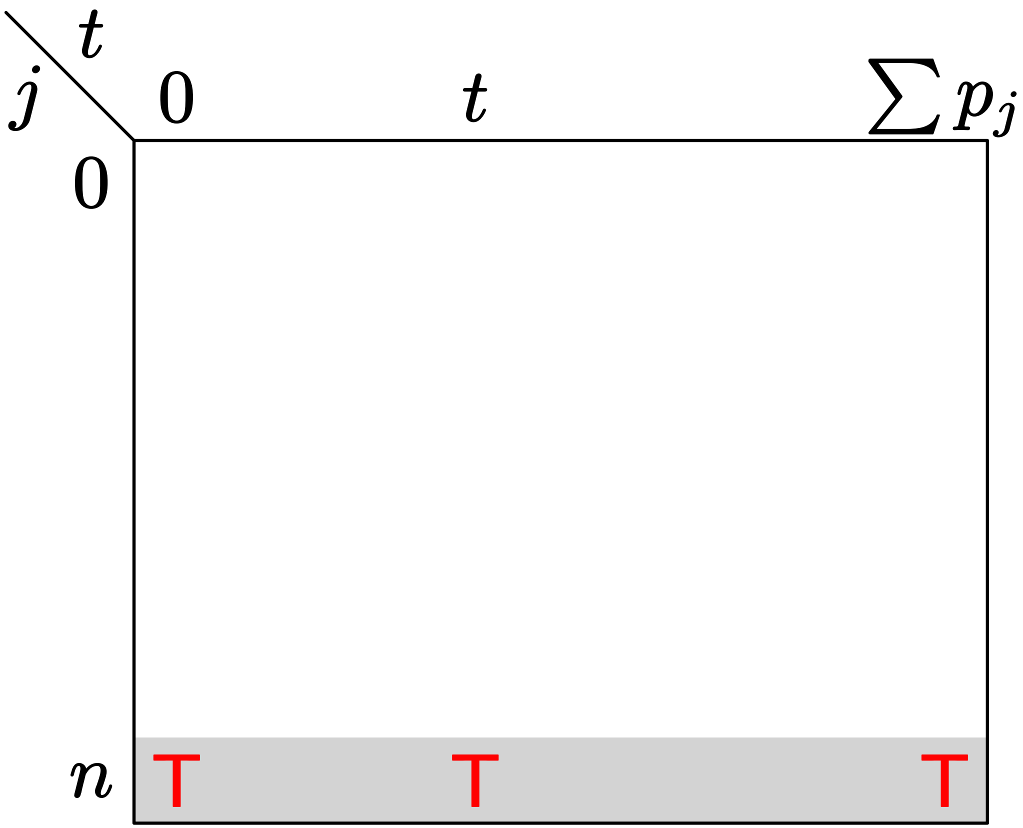
T F F T F F F F F F F F F F F F F F F F F F F F F F F F F F F F

T F F T F F F F F T F F T F F F F F F F F F F F F F F F F F F F

T F F T F F F F F T F F T F F T F F F F F T F F T F F F F F F F

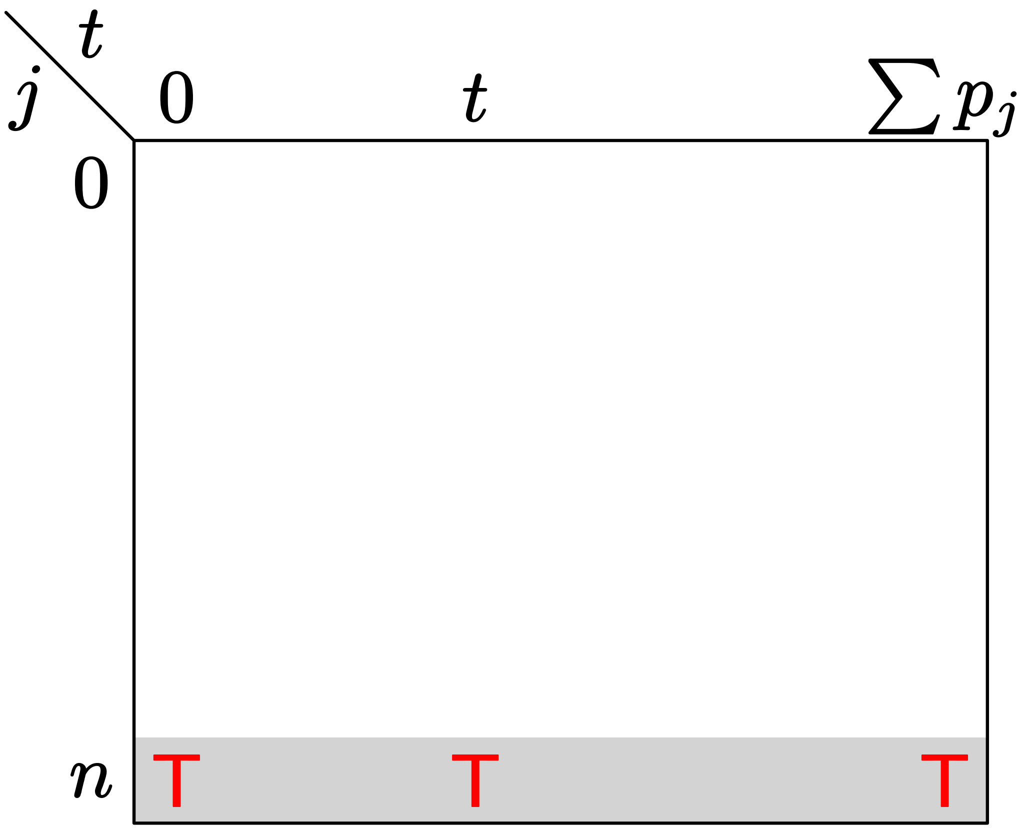
T F F T F F F T F T T F T F F T T F F T F T T F T F F F T F F T

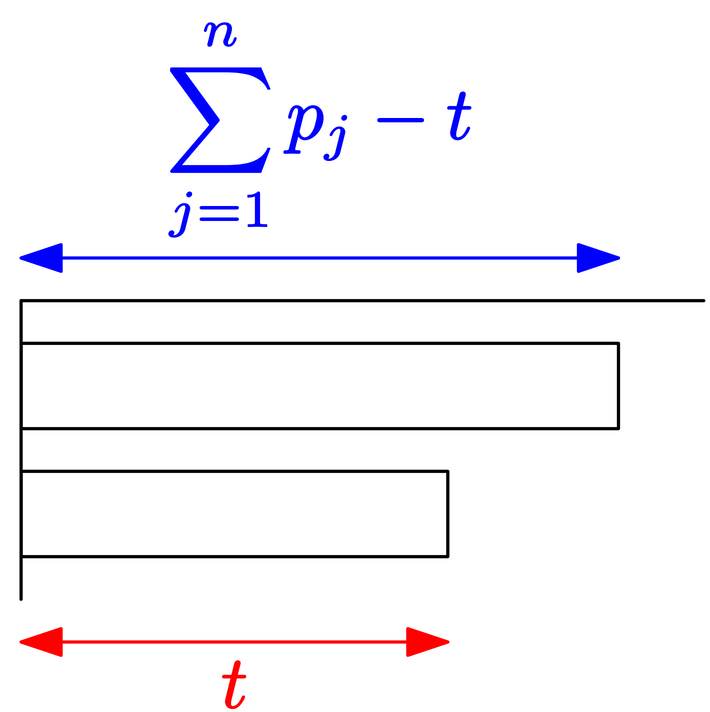
$$\text{最適値} = \min_{t \in \{0\} \cup [\sum p_j]} \left\{ \max\left\{ t, \sum_{j=1}^{n} p_j - t \right\} \;\middle|\; f(n,t) = \textsf{true} \right\}$$



動的計画法：最適値の特定

$$\text{最適値} = \min_{t \in \{0\} \cup [\sum p_j]} \left\{ \max\left\{ t, \sum_{j=1}^{n} p_j - t \right\} \,\middle|\, f(n,t) = \mathsf{true} \right\}$$





動的計画法に基づくアルゴリズム

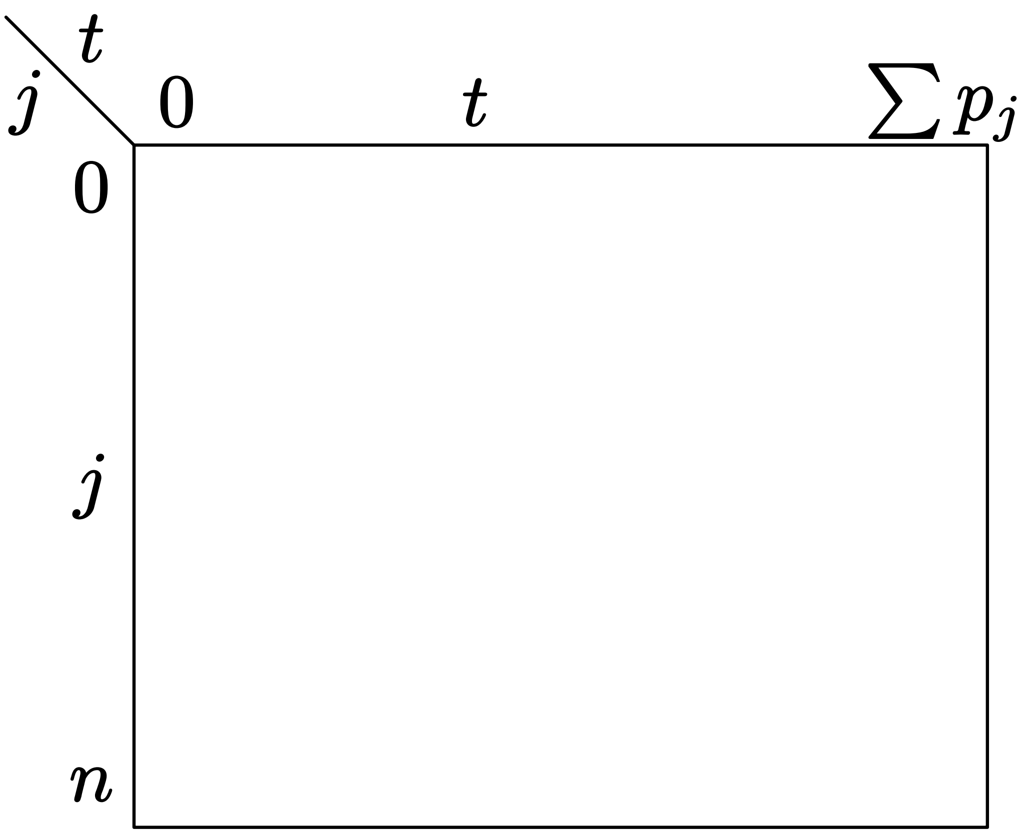
1. 再帰式に基づき，すべての j, t に対して，$f(j, t)$ を計算
2. $f(j, n)$ から最適値を特定

<u>正当性</u>：ここまでの議論から分かる

$P2 \parallel C_{\max}$：計算量 (1)

動的計画法に基づくアルゴリズム

1. 再帰式に基づき，すべての j, t に対して，$f(j, t)$ を計算
2. $f(j, n)$ から最適値を特定



Step 1 の計算量

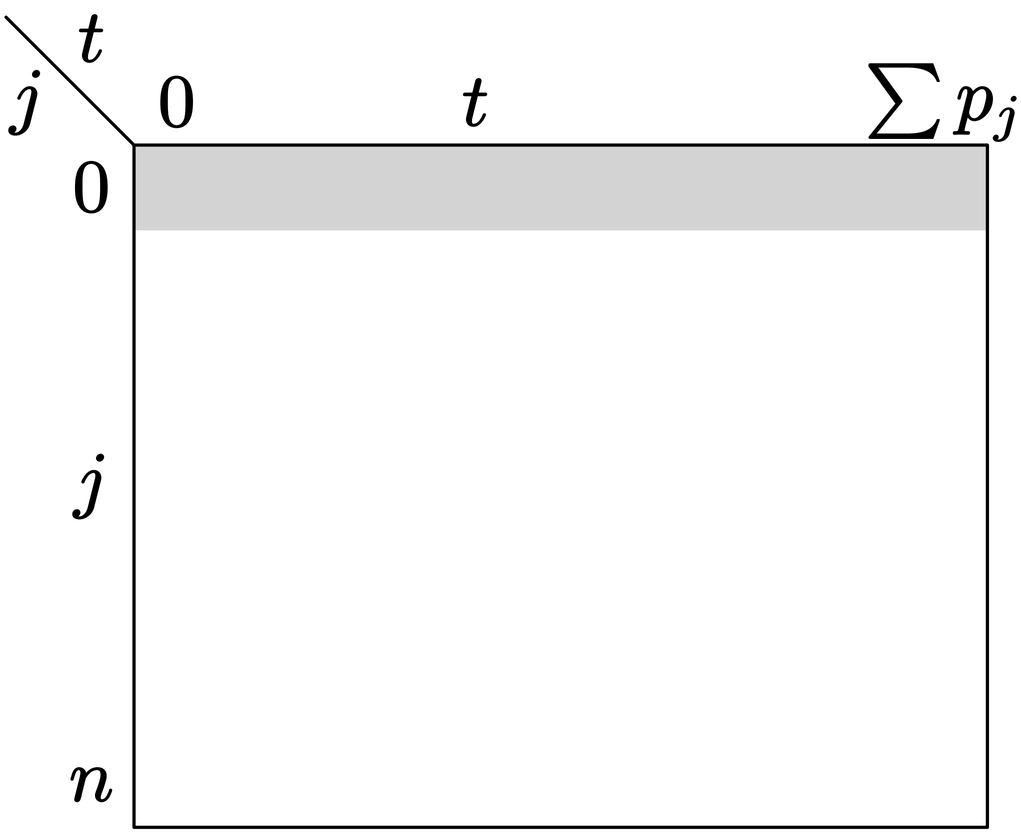
= (マスの数) ×

(1 マスにかかる計算量)

$P2 \mid\mid C_{\max}$：計算量 (1)

動的計画法に基づくアルゴリズム

1. 再帰式に基づき，すべての j, t に対して，$f(j,t)$ を計算
2. $f(j,n)$ から最適値を特定



Step 1 の計算量

$= $ (マスの数) $\times$

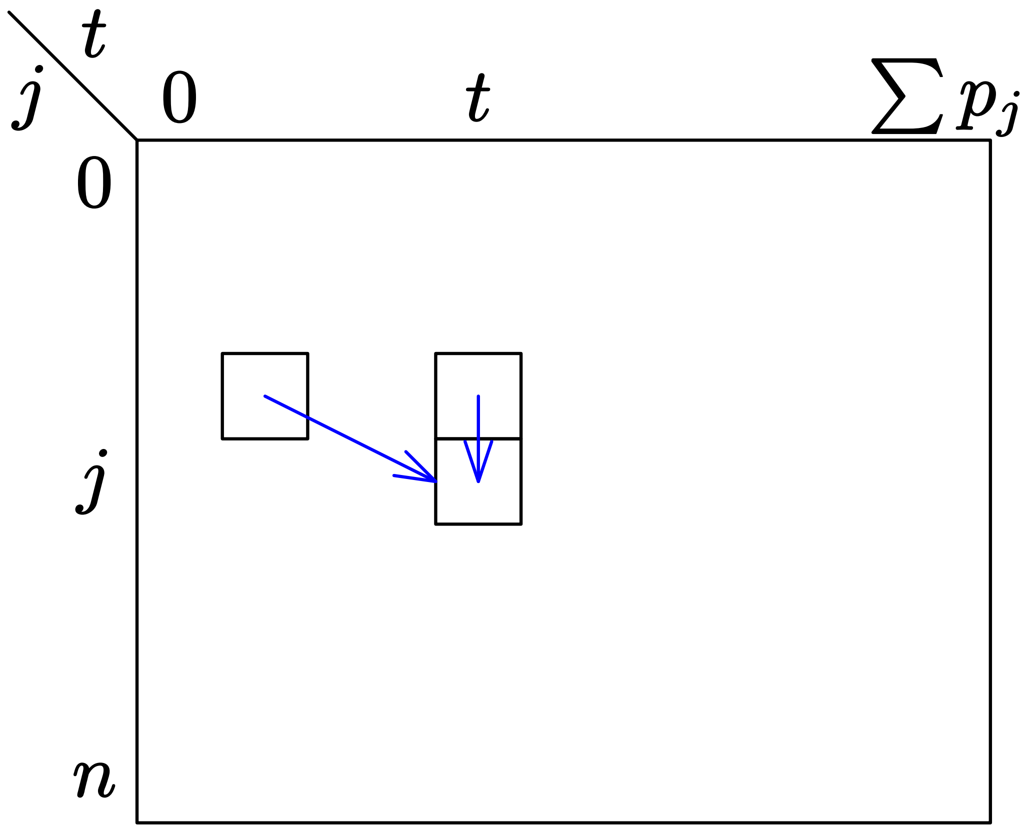
(1 マスにかかる計算量)

$= O(n \sum p_j) \times$

$O(1)$

動的計画法に基づくアルゴリズム

1. 再帰式に基づき，すべての j, t に対して，$f(j,t)$ を計算
2. $f(j,n)$ から最適値を特定



Step 1 の計算量

$=$ (マスの数) $\times$

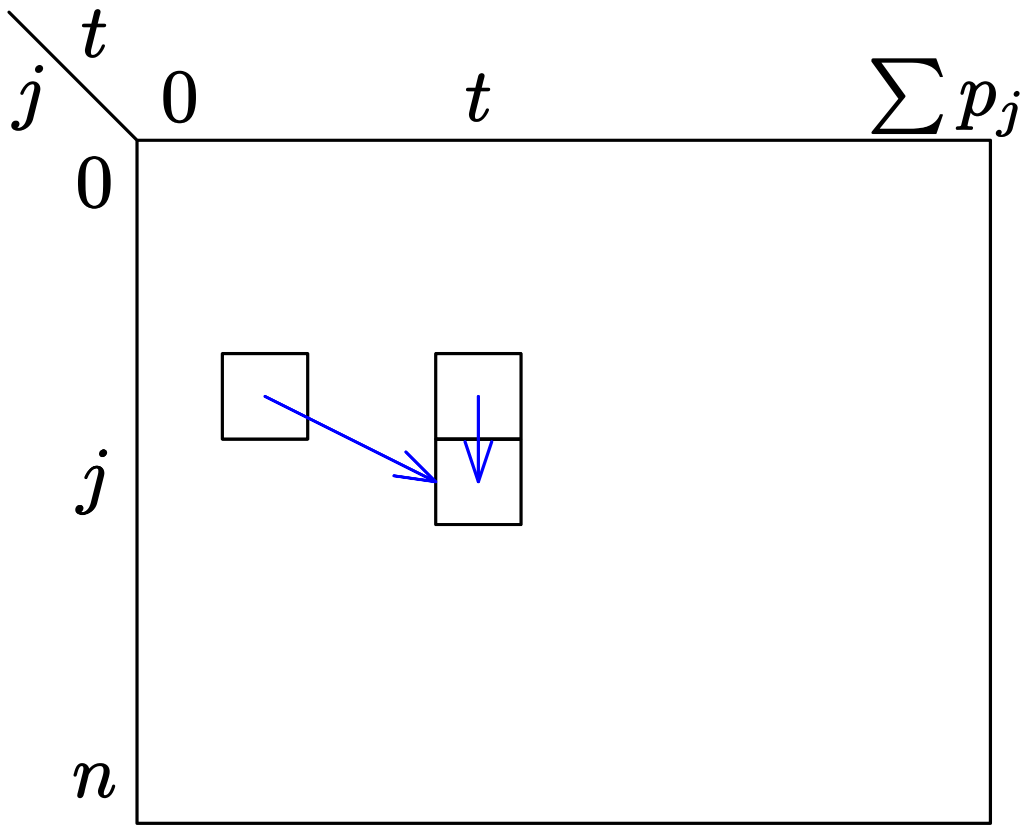
(1 マスにかかる計算量)

$= O(n \sum p_j) \times$

$O(1)$

動的計画法に基づくアルゴリズム

1. 再帰式に基づき，すべての j, t に対して，$f(j,t)$ を計算
2. $f(j,n)$ から最適値を特定



Step 1 の計算量

$= $ (マスの数) $\times$

(1 マスにかかる計算量)

$= O(n \sum p_j) \times$

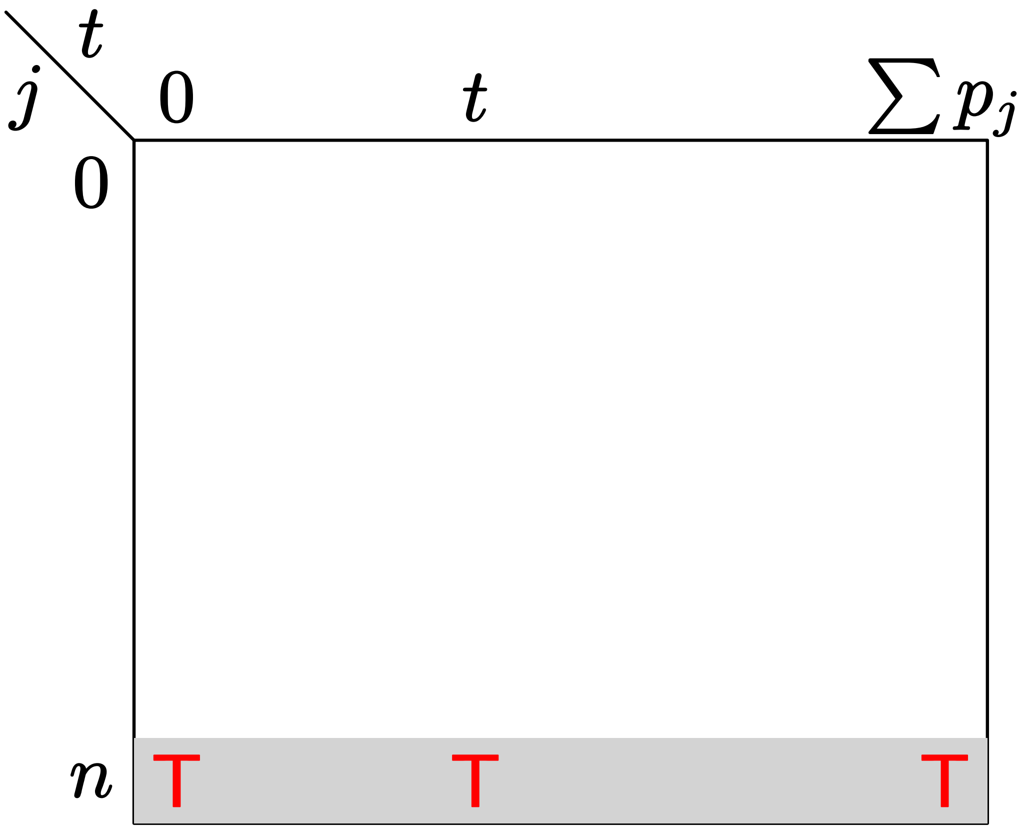
$O(1)$

$= O(n \sum p_j)$

$P2 \,||\, C_{\max}$：計算量 (2)

動的計画法に基づくアルゴリズム

1. 再帰式に基づき，すべての j, t に対して，$f(j,t)$ を計算
2. $f(j,n)$ から最適値を特定



Step 2 の計算量 $= O(\sum p_j)$

$\therefore$ アルゴリズムの計算量

$=$ Step 1 の計算量 $+$

Step 2 の計算量

$= O(n \sum p_j) + O(\sum p_j)$

$= O(n \sum p_j)$

定理 (再掲) (Bellman '57)

$P2 \mid\mid C_{\max}$ は $O\left(n \underbrace{\sum p_j}\right)$ 時間で解ける

処理時間の総和

$\boxed{\text{注}}$：この定理そのものが Bellman ('57) に書かれているわけではない

動的計画法を考えるときの鍵

1. 最適解の持つ **再帰的な構造** を見出す
2. 上の構造から，**状態** を適切に定義する
3. 状態の間の **再帰式** を立てる

1. $P2 \,||\, C_{\max}$ と擬多項式時間動的計画法
2. $1 \,||\, \sum T_j$ **と擬多項式時間動的計画法**

$\boxed{\text{注}}$：今日の授業では，処理時間はすべて整数とする

- E.L. Lawler, A “pseudopolynomial” algorithm for sequencing jobs to minimize total tardiness. *Annals of Discrete Mathematics* 1 (1977) 331–342.

機械の環境

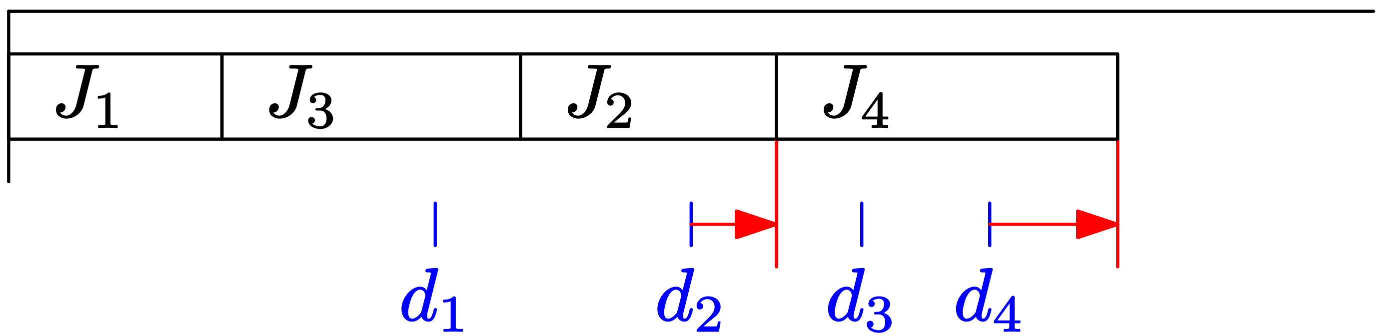
- 一機械

ジョブの特性

- 各ジョブ J_j が処理時間 p_j，納期 d_j を持つ

最適化する目的

- 総納期遅れの最小化 $T_j = \max\{0, C_j - d_j\}$



総納期遅れ最小化：結論から先に

定理：$1 \,||\, \sum T_j$ に対するアルゴリズム　(Lawler '77)

$1 \,||\, \sum T_j$ は $O(n^4 \sum p_j)$ 時間で解ける

注意：**擬多項式時間** の計算量 (詳細は次回)

比較	「自明」	$O(n! \cdot n)$	指数時間
	動的計画法	$O(n^4 \sum p_j)$	擬多項式時間

以下，議論を簡単にするため，すべてのジョブの処理時間は異なるとする

定理：$1 \mid\mid \sum T_j$ に対するアルゴリズム　(Lawler '77)

$1 \mid\mid \sum T_j$ は $O(n^4 \sum p_j)$ 時間で解ける

$\boxed{\text{注意}}$：**擬多項式時間** の計算量 (詳細は次回)

アルゴリズムは **動的計画法** に基づく

動的計画法を考えるときの鍵

1. 最適解の持つ **再帰的な構造** を見出す
2. 上の構造から，**状態** を適切に定義する
3. 状態の間の **再帰式** を立てる

総納期遅れ最小化：最適解の構造

$1 \mid\mid \sum T_j$ では

1つの最適解 が持つ

構造を見出す

$P2 \mid\mid C_{\max}$ では

すべての最適解 が持つ

構造を見出した

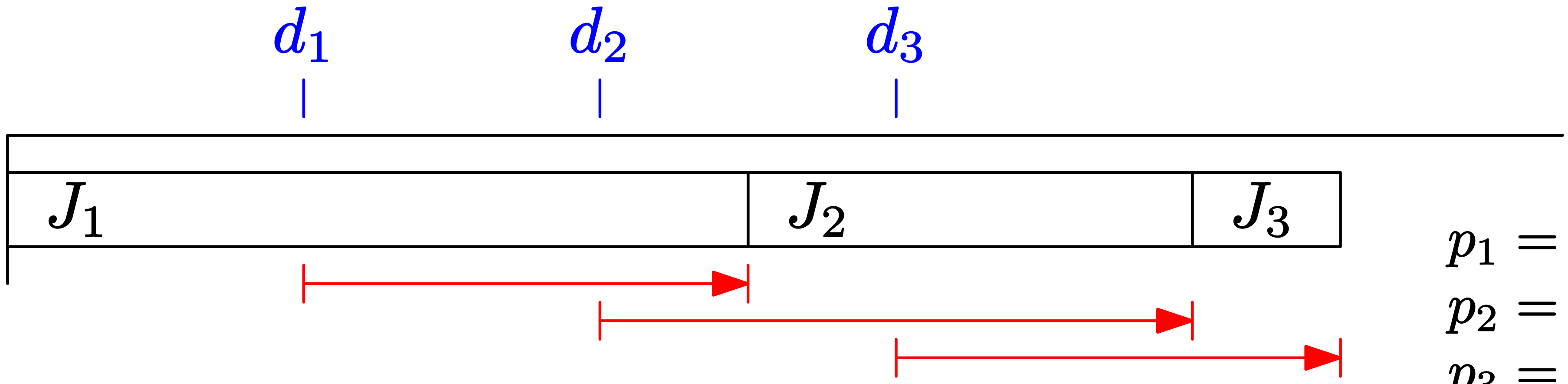
動的計画法を考えるときの鍵

1. 最適解の持つ **再帰的な構造** を見出す
2. 上の構造から，**状態** を適切に定義する
3. 状態の間の **再帰式** を立てる

EDDではダメか？

EDDでは最適スケジュールを得られない場合がある

(最早納期優先規則，earliest due date)

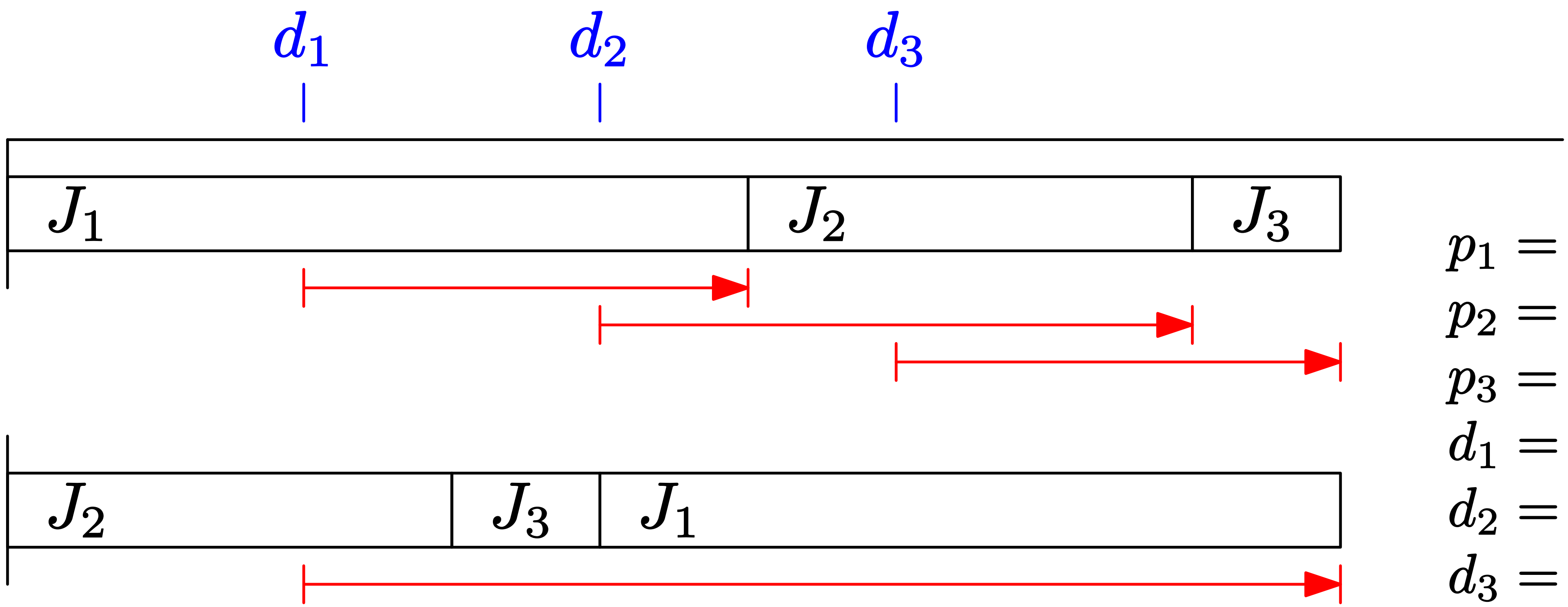


$$p_1 = 20$$
$$p_2 = 12$$
$$p_3 = 4$$
$$d_1 = 8$$
$$d_2 = 16$$
$$d_3 = 24$$

EDD ではダメか？

EDDでは最適スケジュールを得られない場合がある

(最早納期優先規則，earliest due date)

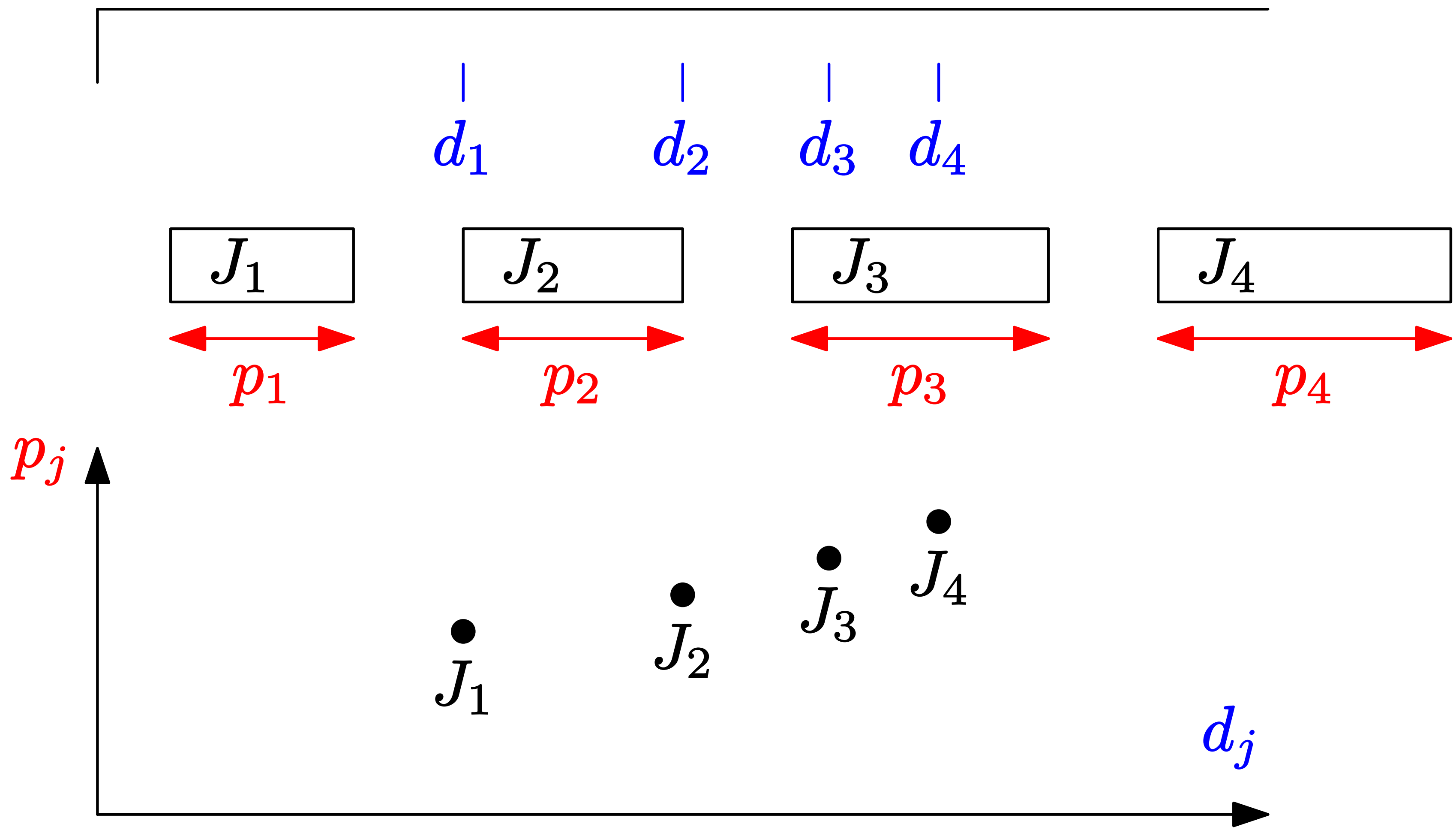


$p_1 = 20$
$p_2 = 12$
$p_3 = 4$
$d_1 = 8$
$d_2 = 16$
$d_3 = 24$

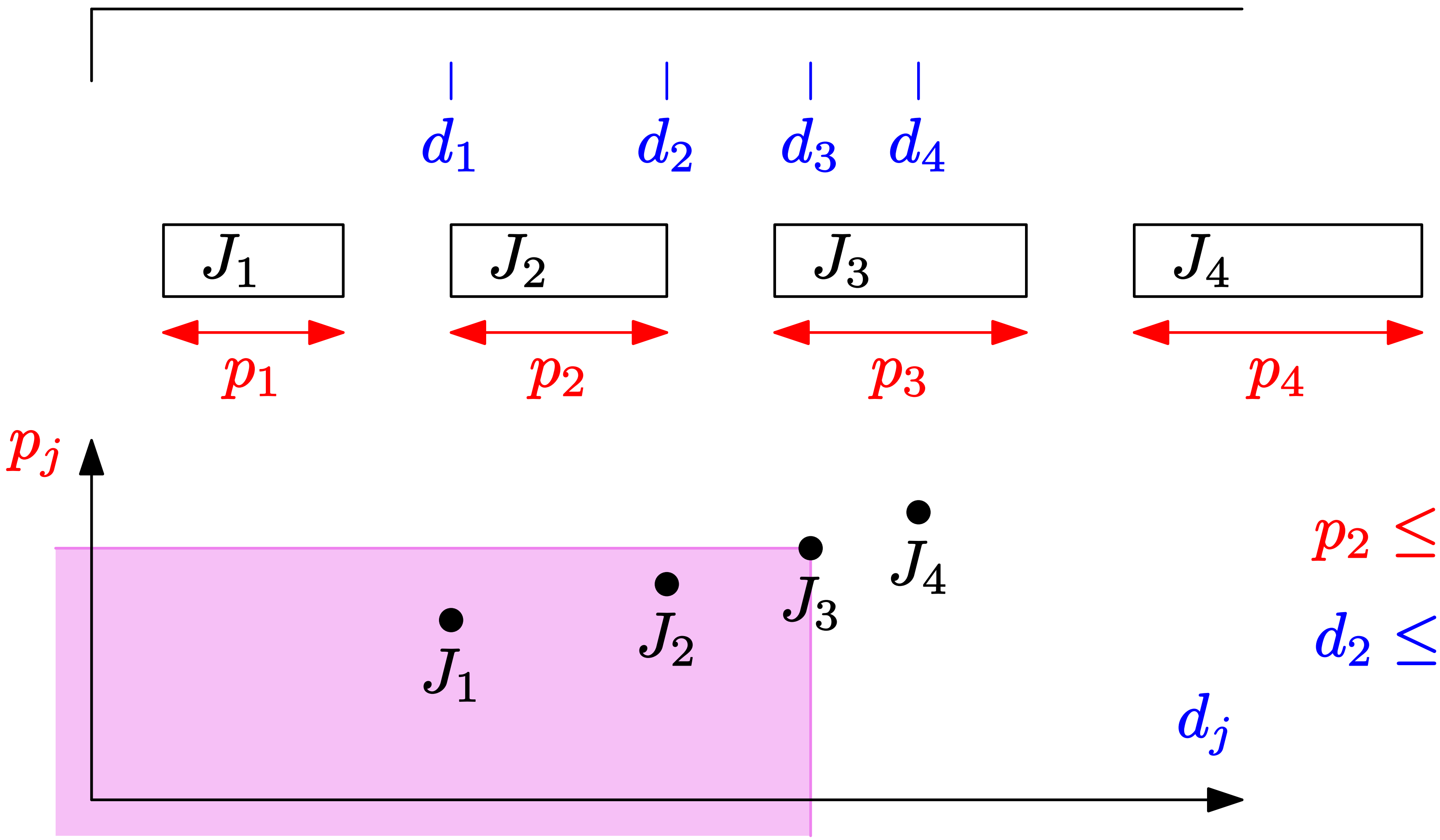
教訓：アルゴリズム設計全般において

うまくいかない場合を考える／気に留める

処理時間と納期のプロット



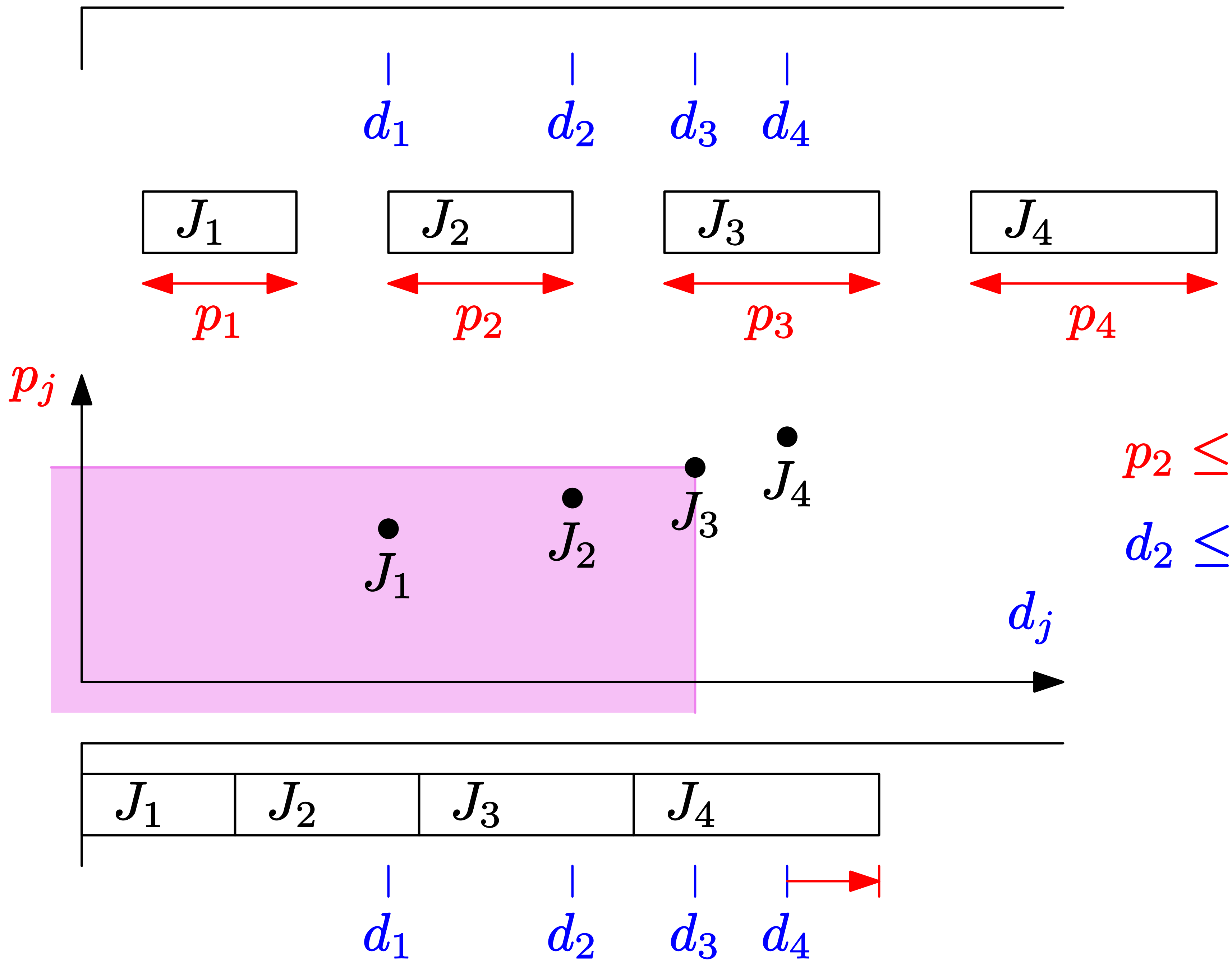
処理時間と納期のプロット



$p_2 \leq p_3$

$d_2 \leq d_3$

処理時間と納期のプロット



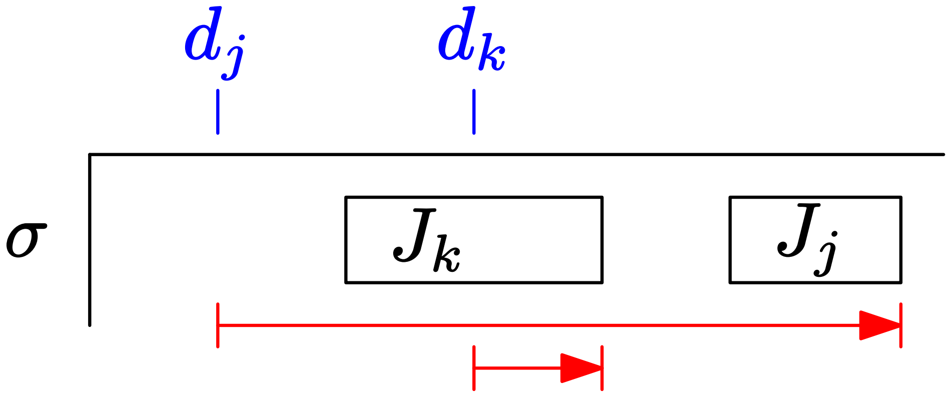
$p_2 \leq p_3$

$d_2 \leq d_3$

補題 A (優越ジョブ)

$1 \mid\mid \sum T_j$ において，$p_j \leq p_k$ かつ $d_j \leq d_k$ であるとき，J_j を J_k より先に処理する最適スケジュールが存在する

証明のイメージ：ジョブ交換論法を使う



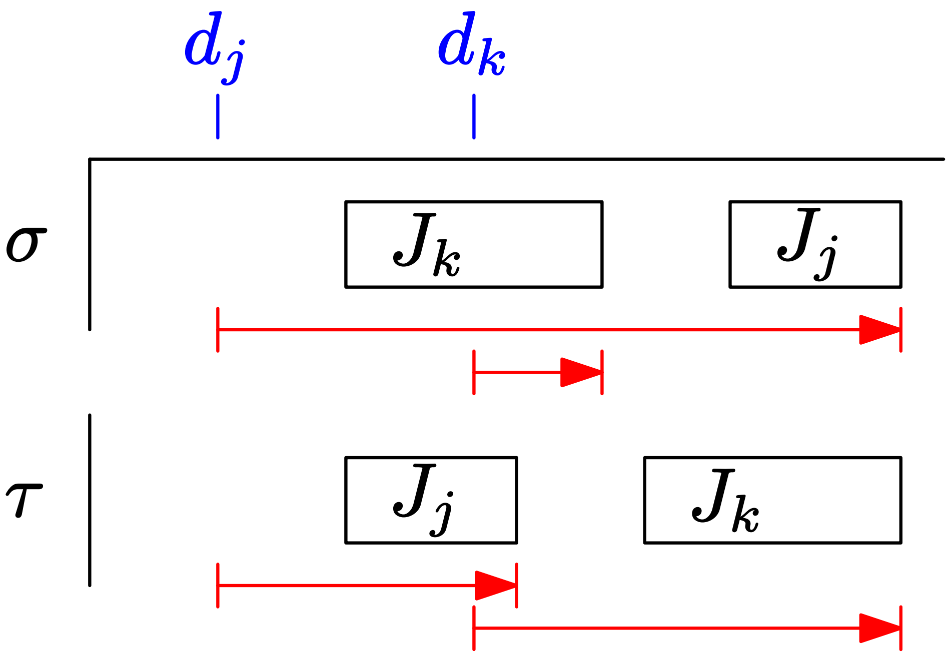
詳細は付録にて

補題 A

補題 A (優越ジョブ)

$1 \,||\, \sum T_j$ において，$p_j \le p_k$ かつ $d_j \le d_k$ であるとき，J_j を J_k より先に処理する最適スケジュールが存在する

証明のイメージ：ジョブ交換論法を使う



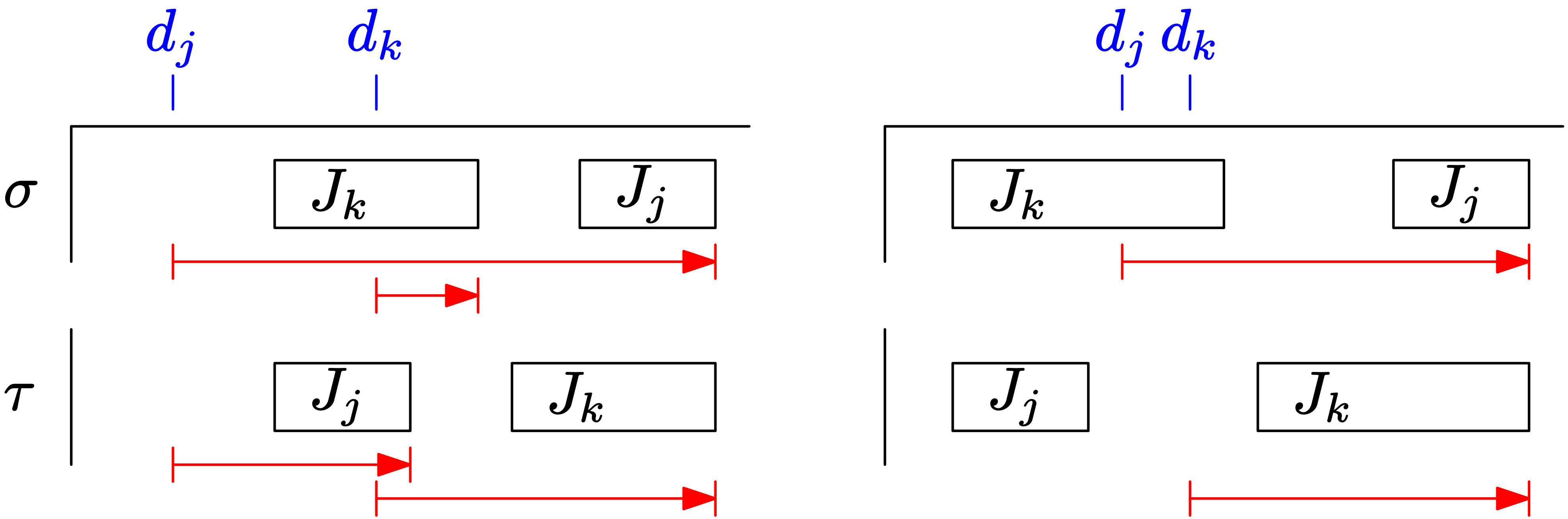
詳細は付録にて

補題 A

補題 A (優越ジョブ)

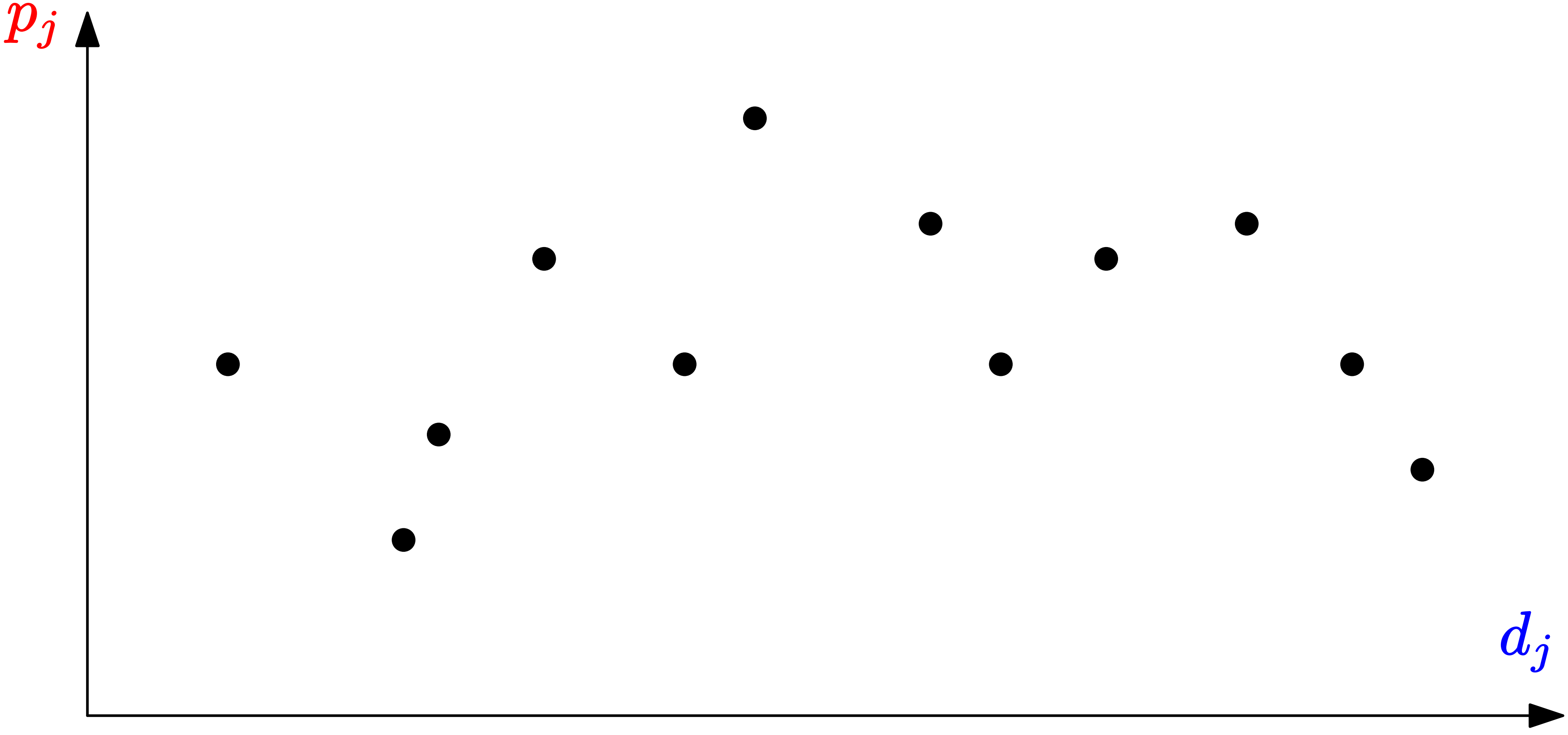
$1 \mid\mid \sum T_j$ において，$p_j \le p_k$ かつ $d_j \le d_k$ であるとき，
J_j を J_k より先に処理する最適スケジュールが存在する

証明のイメージ：ジョブ交換論法を使う

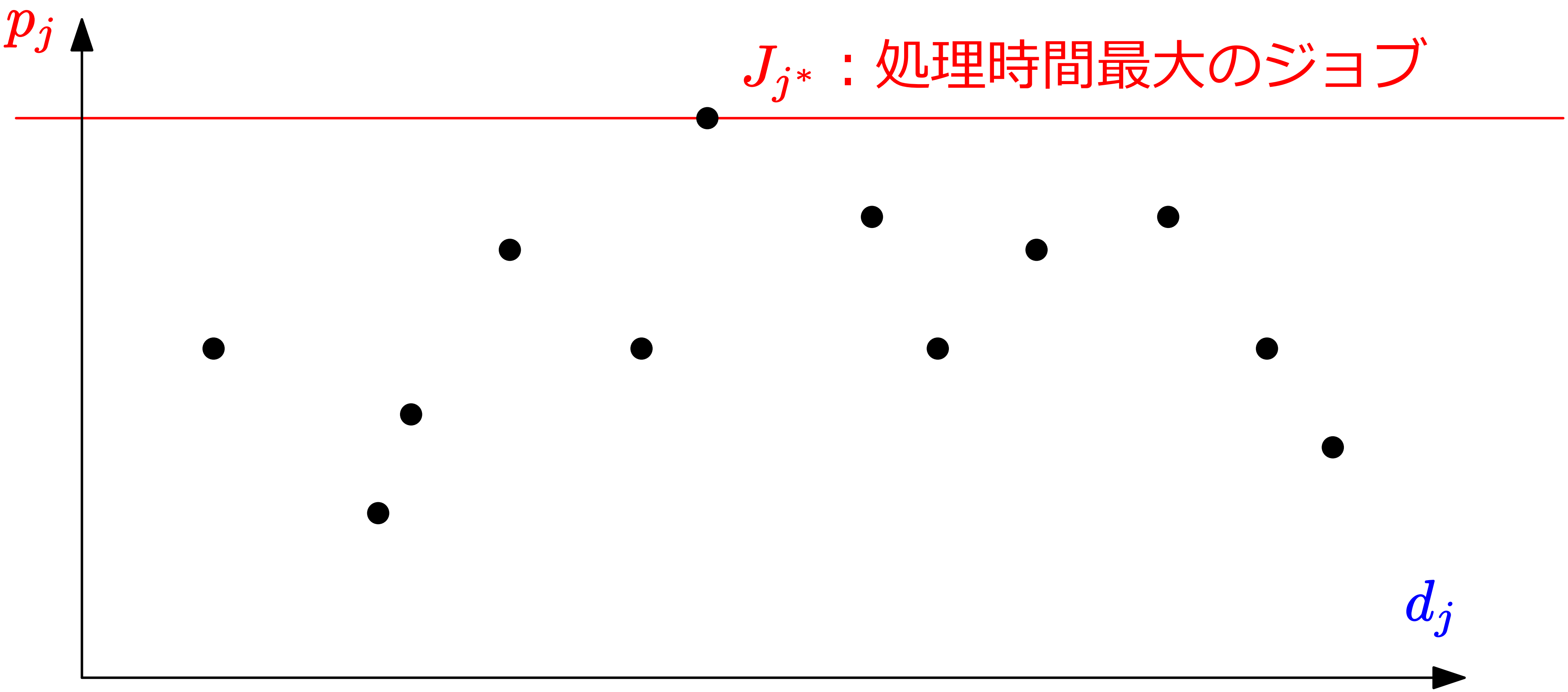


詳細は付録にて

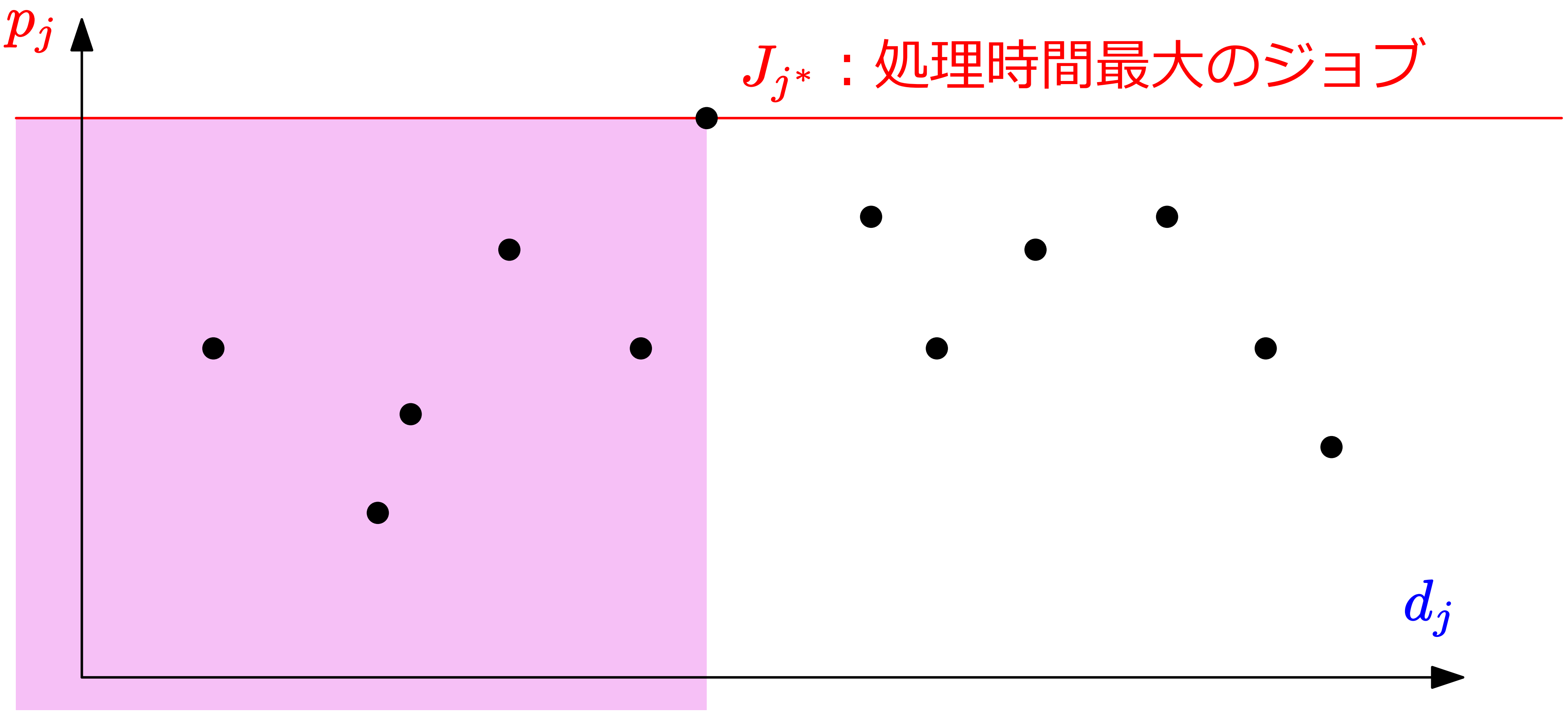
最適スケジュールの処理順



最適スケジュールの処理順



最適スケジュールの処理順



補題 A (優越ジョブ)

$1 \mid\mid \sum T_j$ において，$p_j \leq p_k$ かつ $d_j \leq d_k$ であるとき，J_j を J_k より先に処理する最適スケジュールが存在する

補題 B

仮定：$1 \,||\, \sum T_j$ の設定で，ジョブは EDD 順 $(d_1 \leq \cdots \leq d_n)$

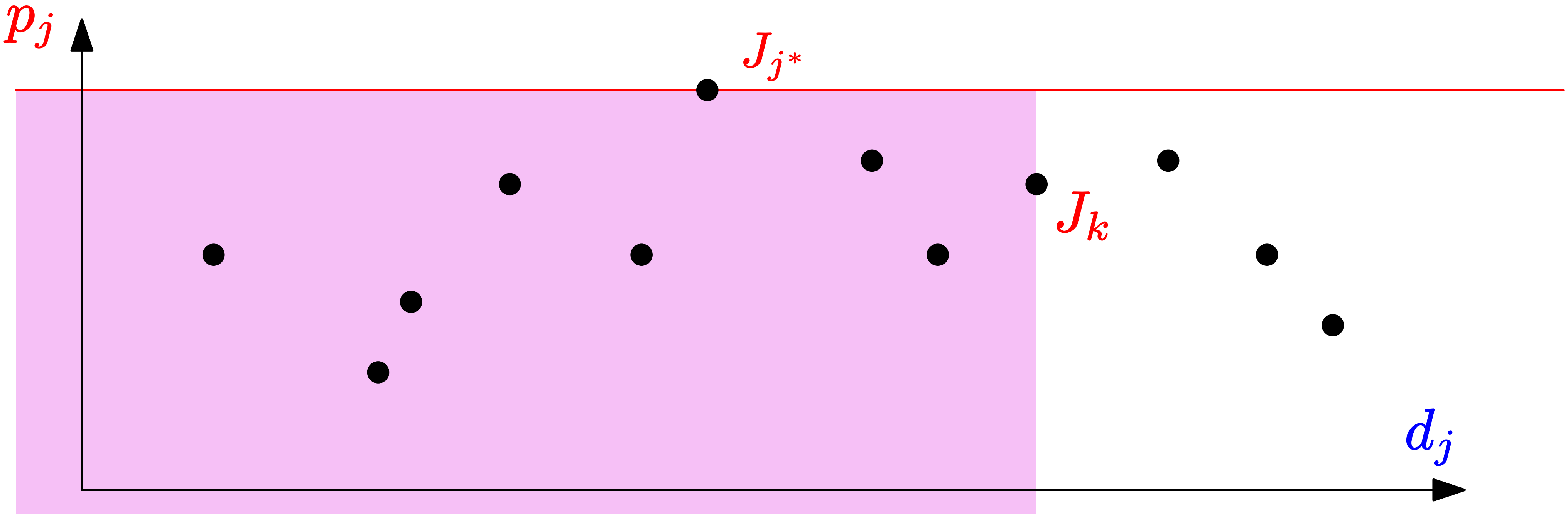
J_{j^*} = 処理時間最大のジョブ

補題 B (処理時間最大ジョブ)

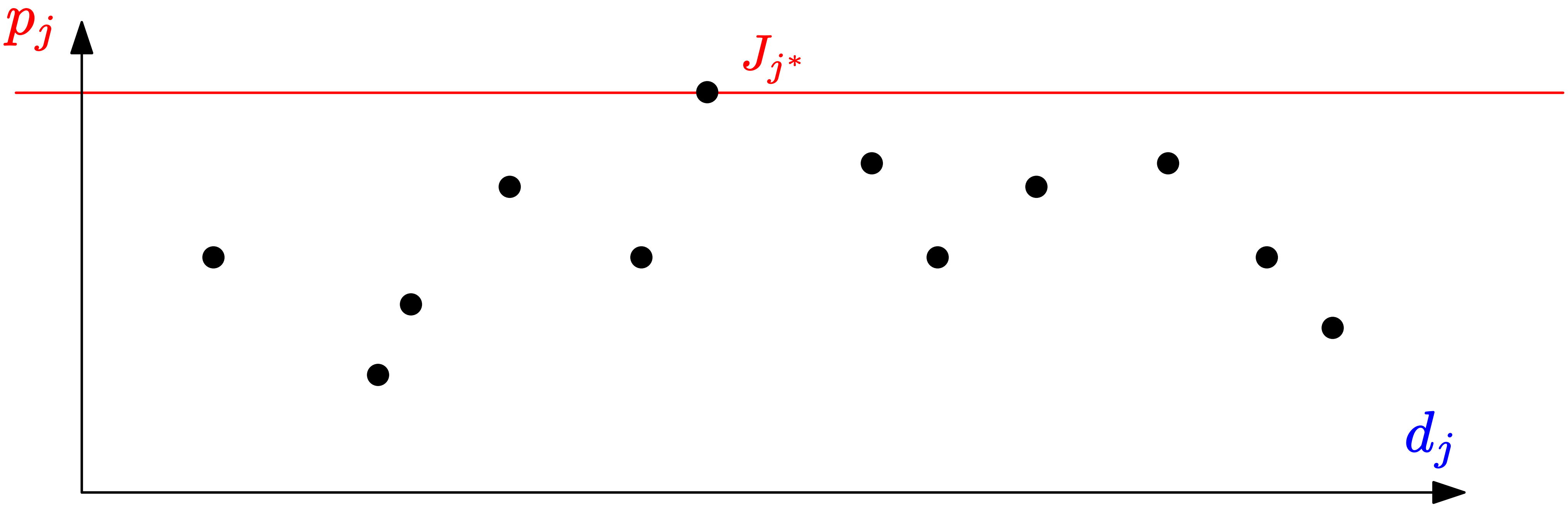
ある k とある最適スケジュールが存在して，そこでは

$j \leq k, j \neq j^*$ のジョブ J_j を J_{j^*} より前に処理し

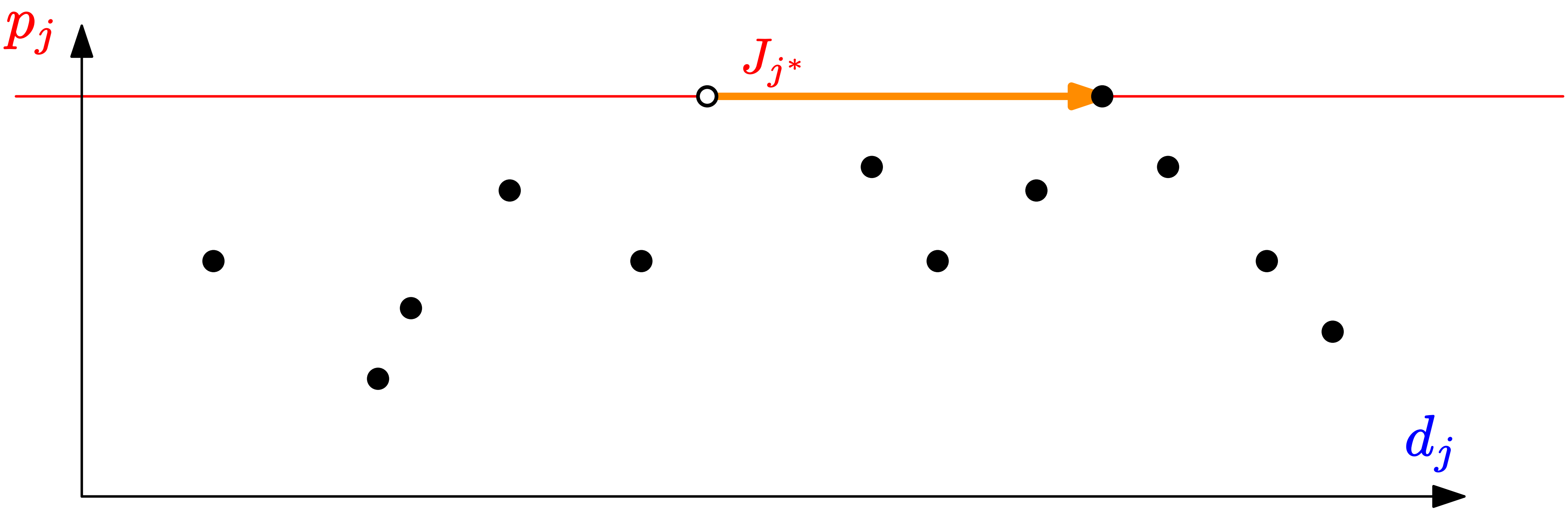
$j > k$ のジョブ J_j を J_{j^*} より後に処理する



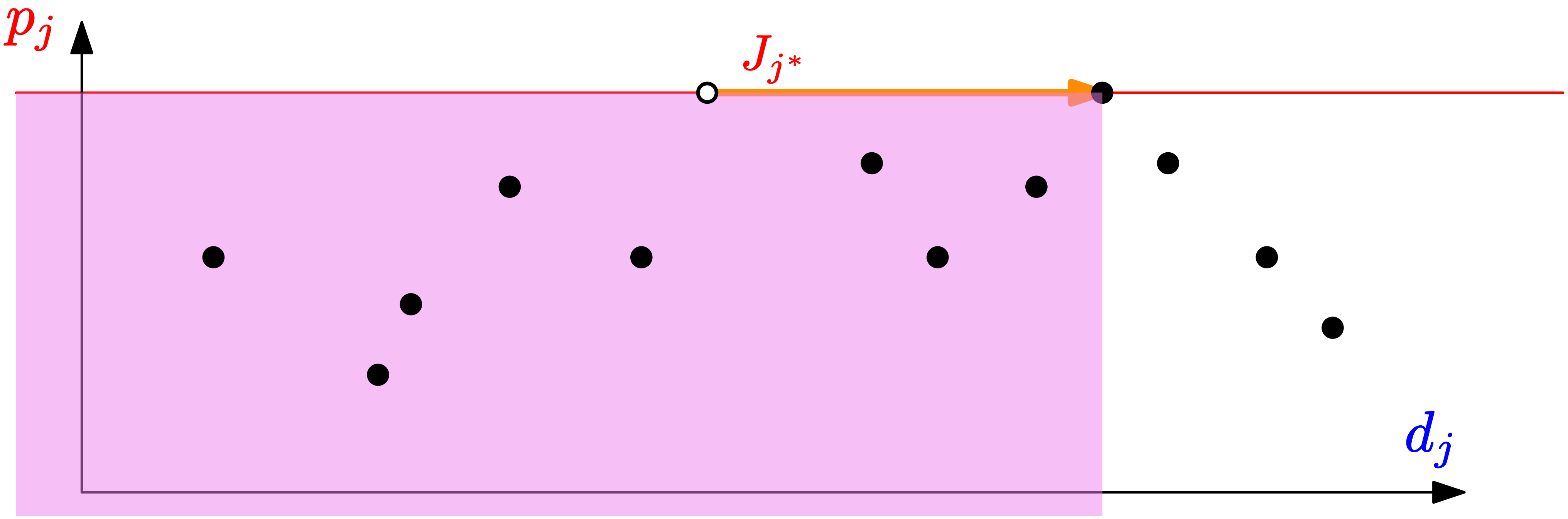
補題 B：直感



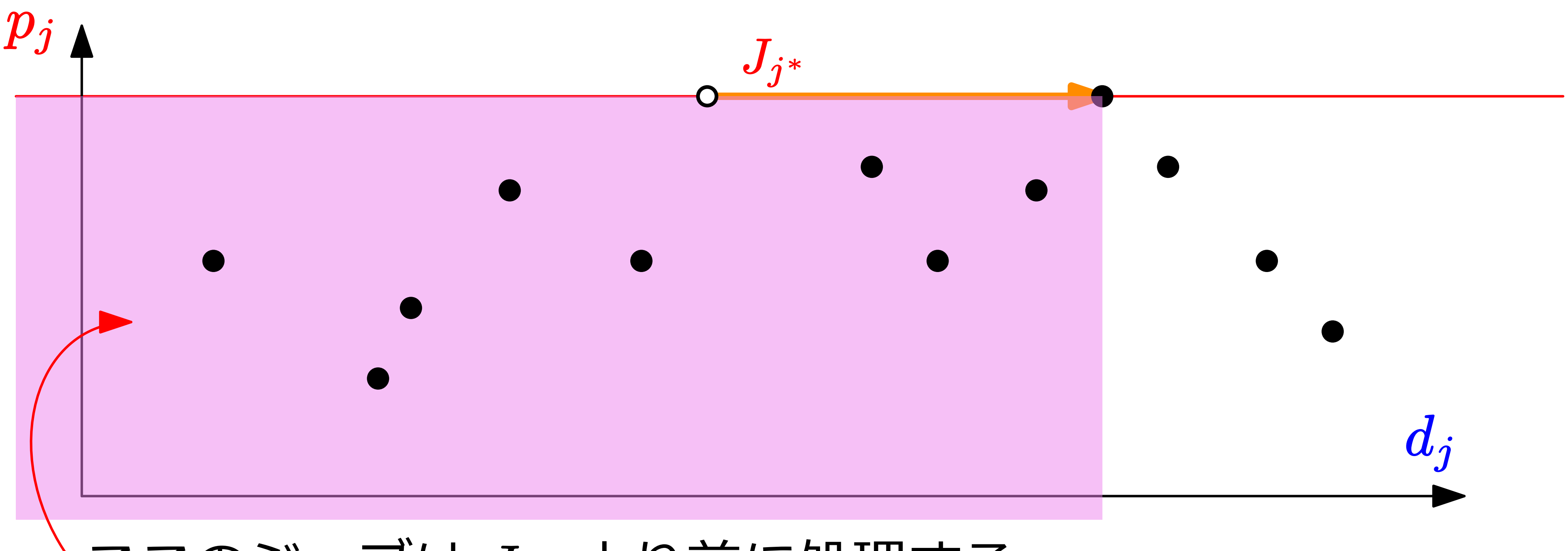
「最適解が変わらない」限り，d_{j^*} を変える



「最適解が変わらない」限り，d_{j^*} を変える



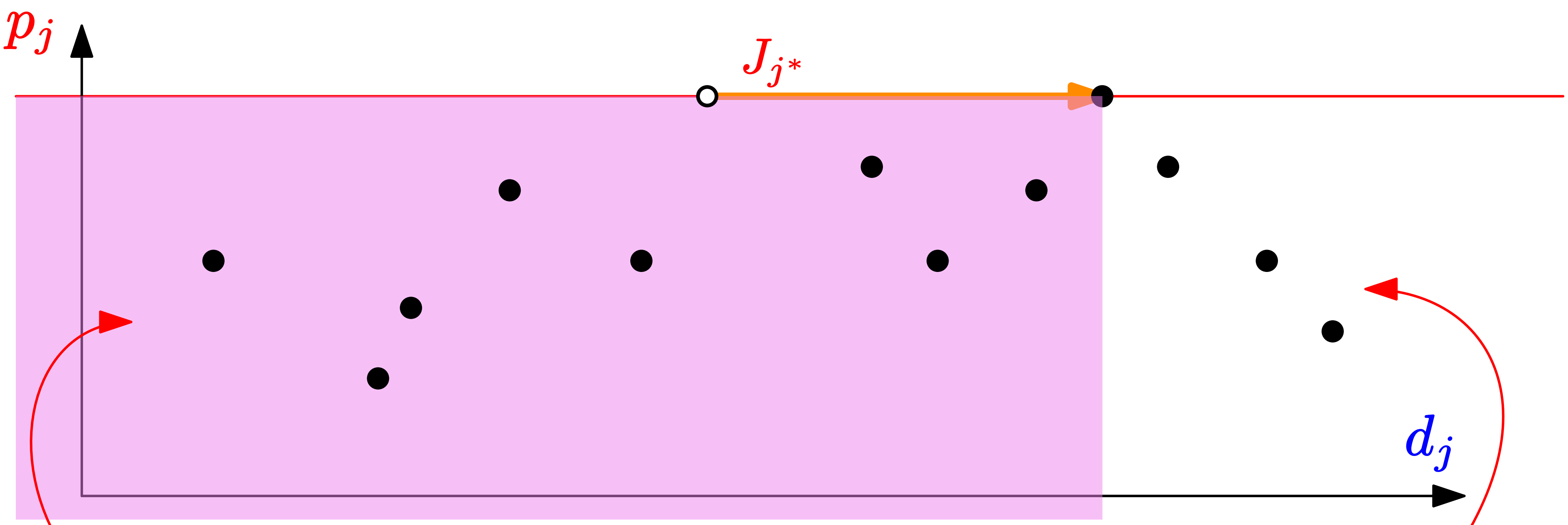
「最適解が変わらない」限り，d_{j^*} を変える



ここのジョブは J_{j^*} より前に処理する

補題 B：直感

「最適解が変わらない」限り，d_{j^*} を変える

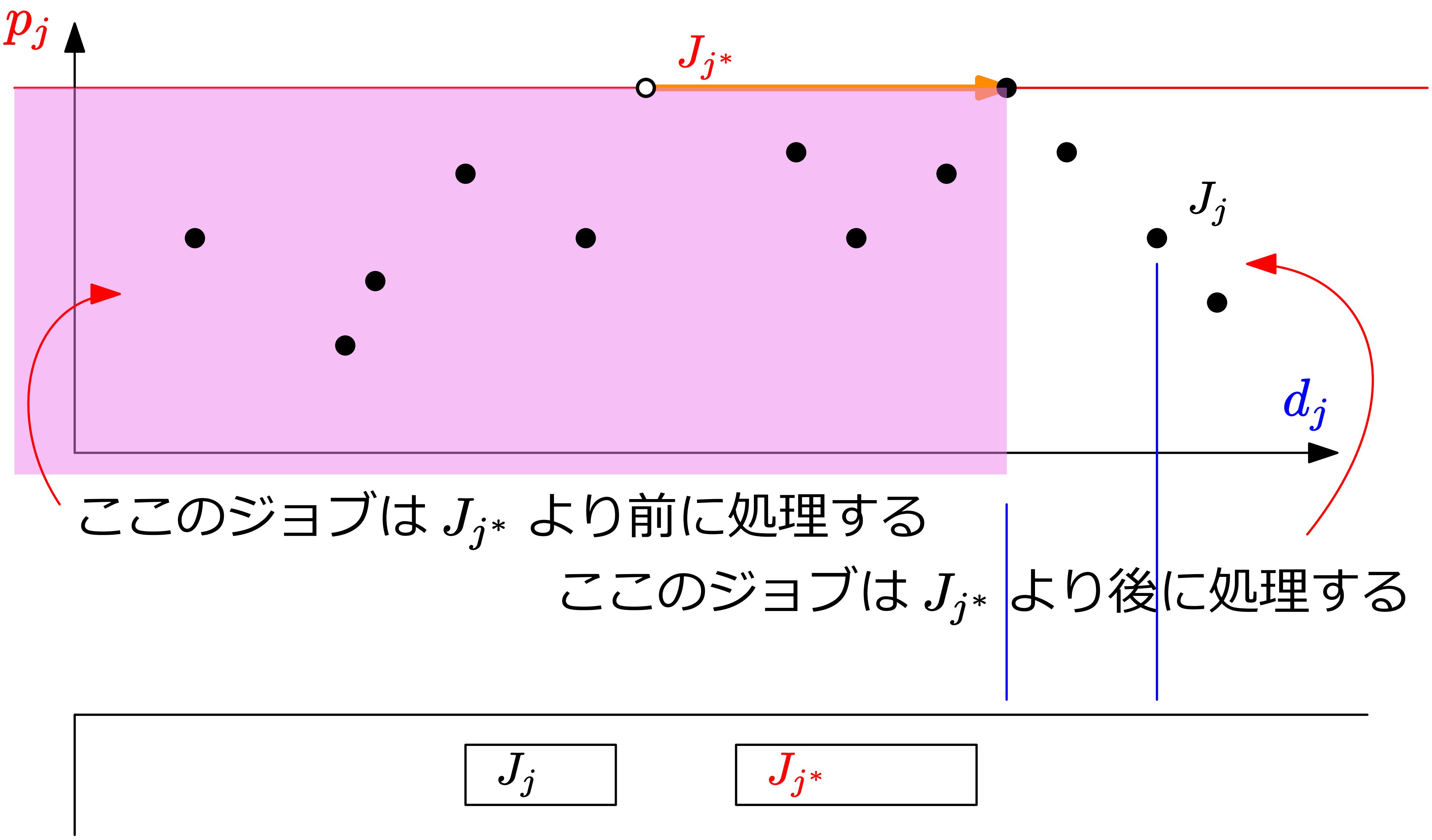


ここのジョブは J_{j^*} より前に処理する

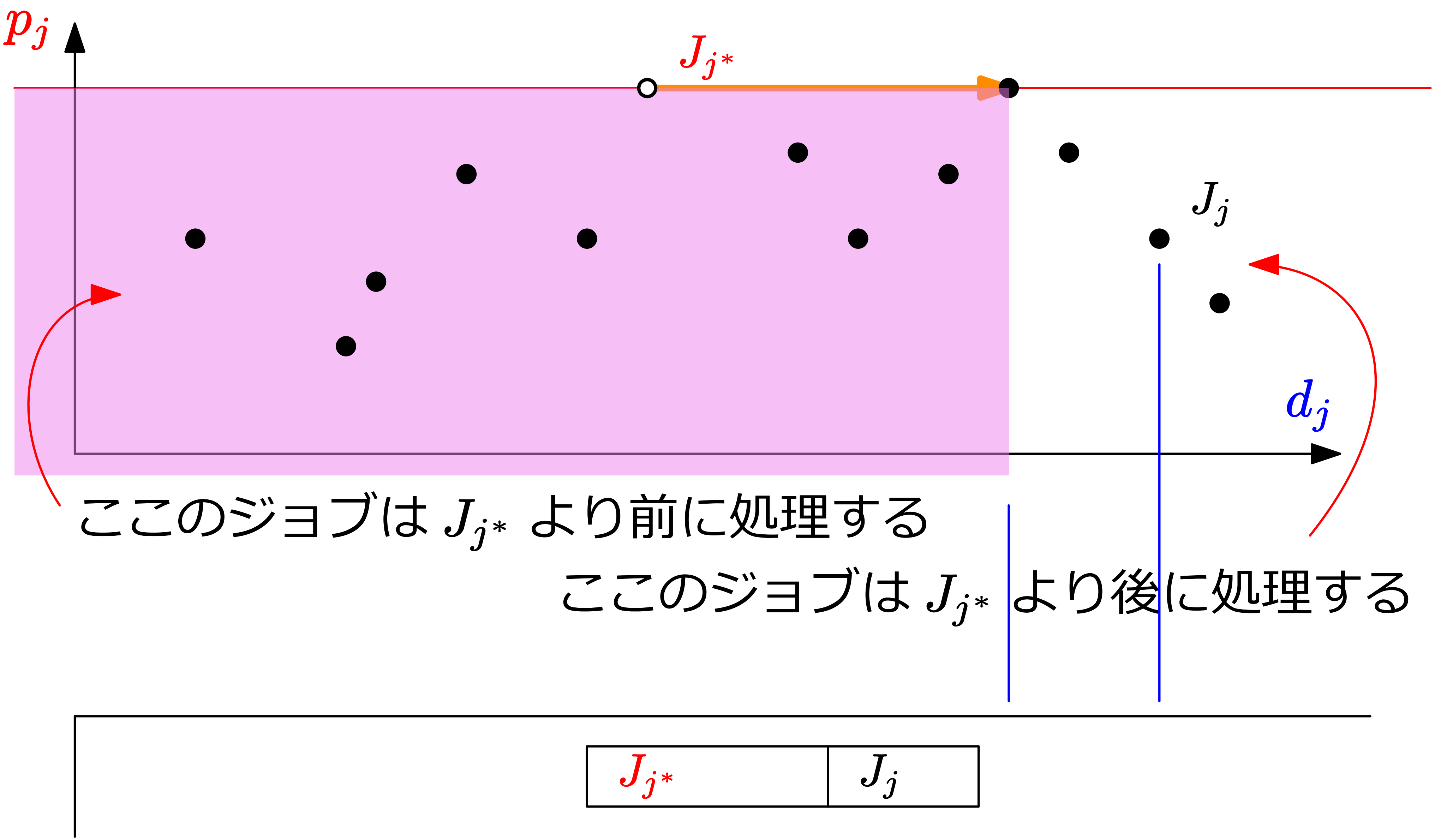
ここのジョブは J_{j^*} より後に処理する

補題 B：直感

「最適解が変わらない」限り，d_{j^*} を変える



「最適解が変わらない」限り，d_{j^*} を変える



補題 B から動的計画法へ

仮定：$1 \,||\, \sum T_j$ の設定で，ジョブは EDD 順 $(d_1 \leq \cdots \leq d_n)$
J_{j^*} = 処理時間最大のジョブ

補題 B (処理時間最大ジョブ) (再掲)

ある k とある最適スケジュールが存在して，そこでは
$j \leq k, j \neq j^*$ のジョブ J_j を J_{j^*} より前に処理し
$j > k$ のジョブ J_j を J_{j^*} より後に処理する

動的計画法を考えるときの鍵

1. 最適解の持つ **再帰的な構造** を見出す
2. 上の構造から，**状態** を適切に定義する
3. 状態の間の **再帰式** を立てる

補題 B から動的計画法へ

仮定：$1 \mid\mid \sum T_j$ の設定で，ジョブは EDD 順 ($d_1 \leq \cdots \leq d_n$)
J_{j^*} = 処理時間最大のジョブ

補題 B (処理時間最大ジョブ) (再掲)

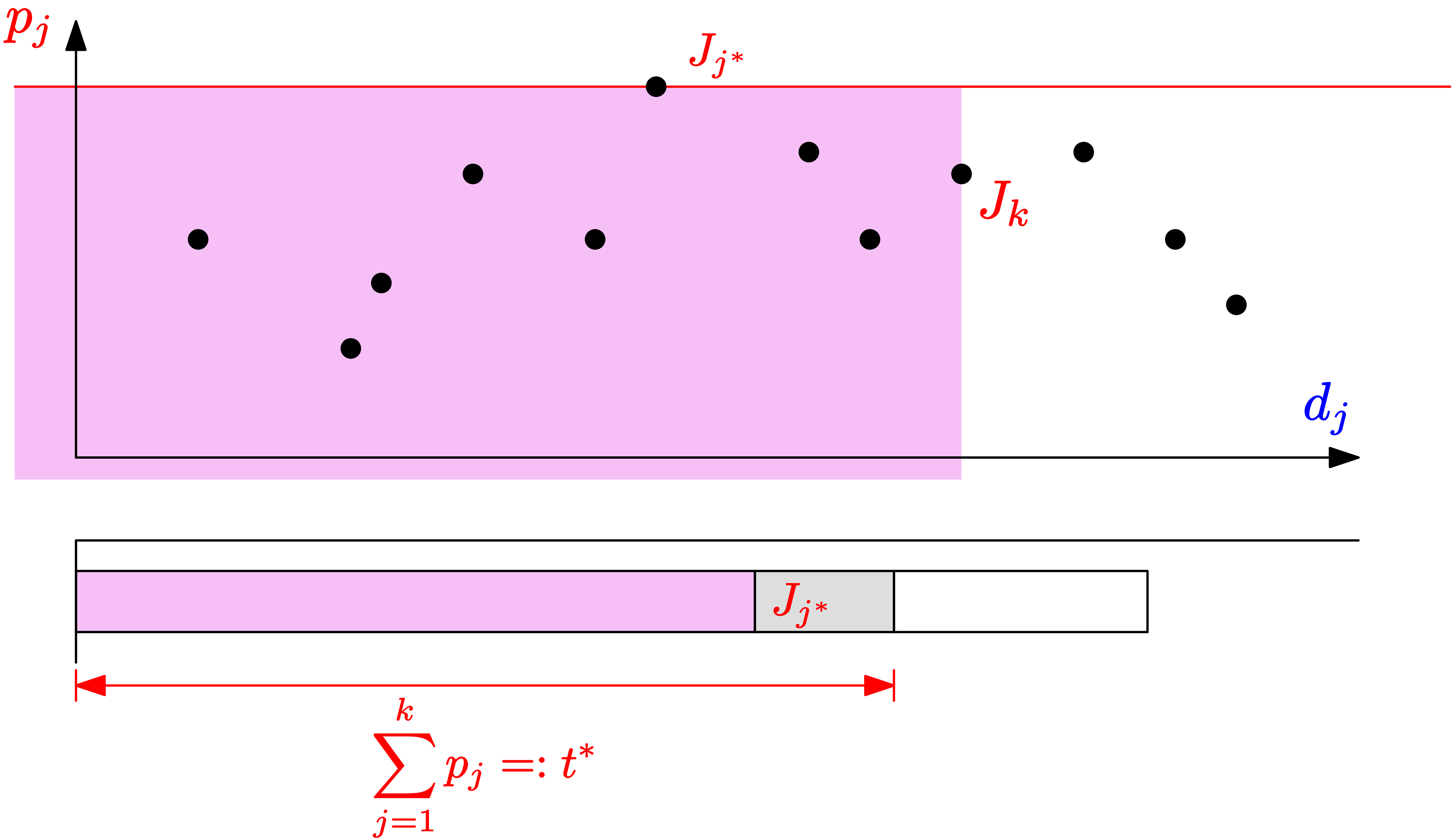
ある k とある最適スケジュールが存在して，そこでは
$j \leq k, j \neq j^*$ のジョブ J_j を J_{j^*} より前に処理し
$j > k$ のジョブ J_j を J_{j^*} より後に処理する

効率よく計算できるか分からない
$\rightsquigarrow$ [解決策] 候補を全部試す

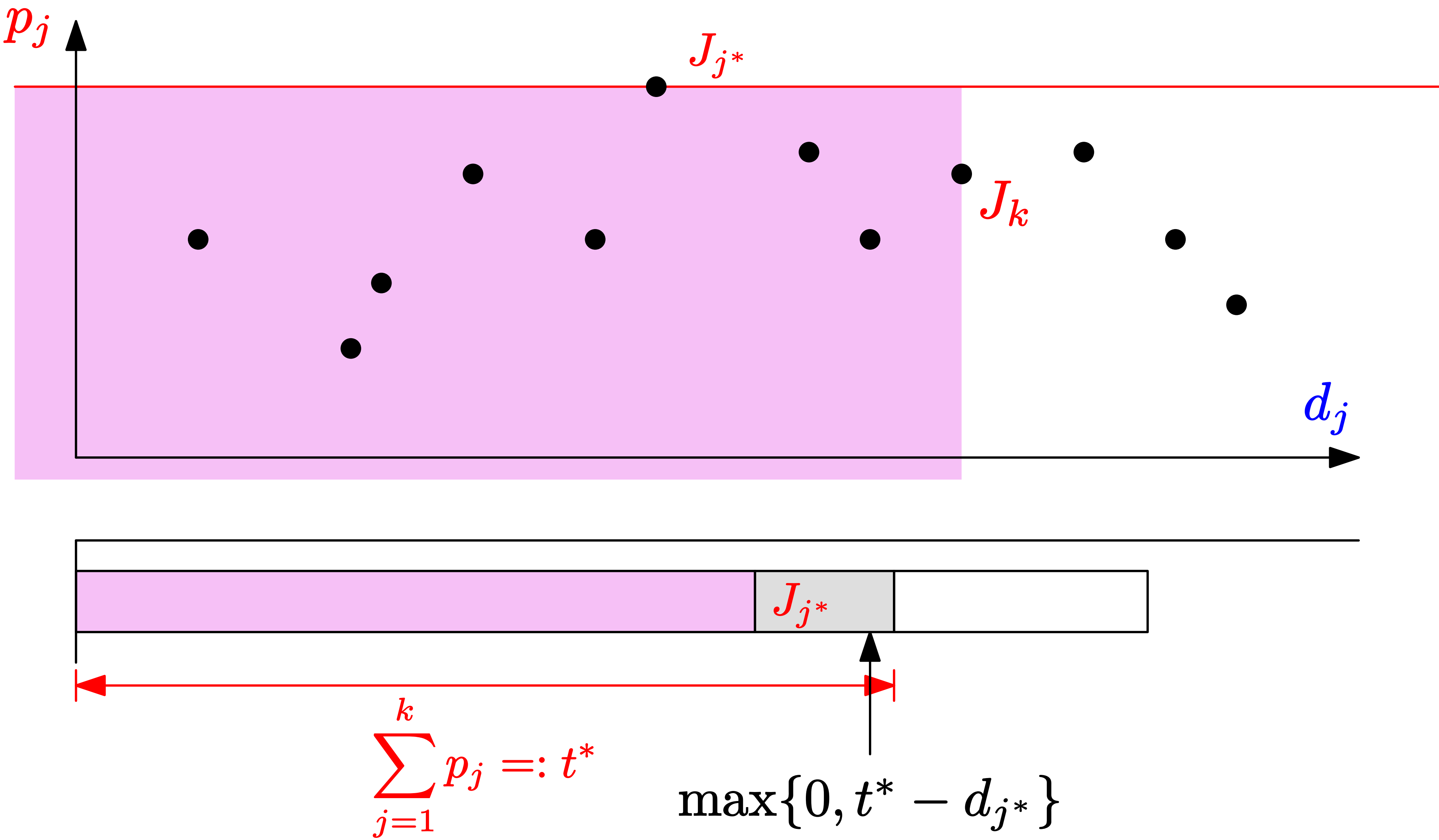
動的計画法を考えるときの鍵

1. 最適解の持つ **再帰的な構造** を見出す
2. 上の構造から，**状態** を適切に定義する
3. 状態の間の **再帰式** を立てる

ある最適解の構造 (1)

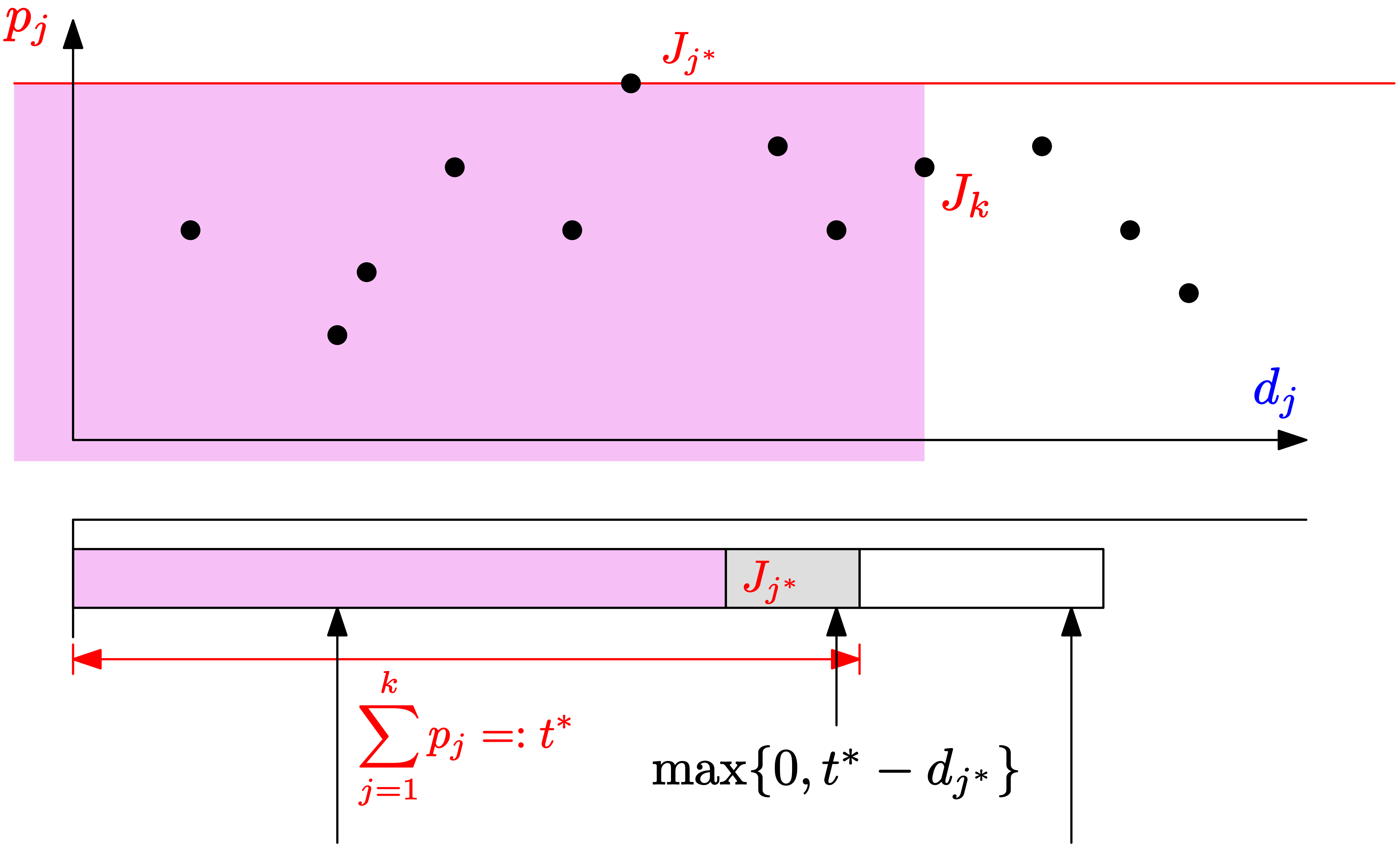


ある最適解の構造 (1)



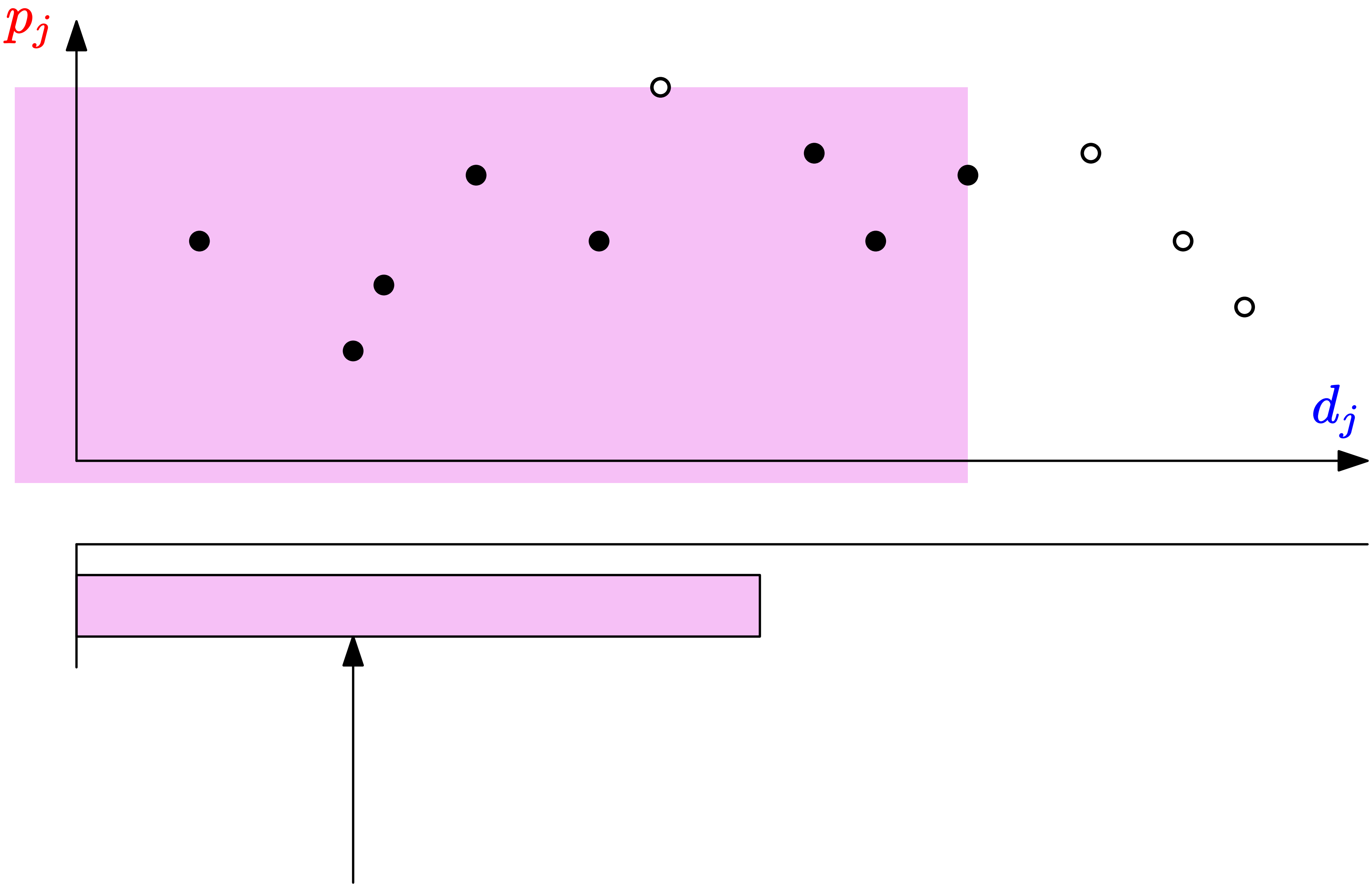
総納期遅れ

ある最適解の構造 (1)



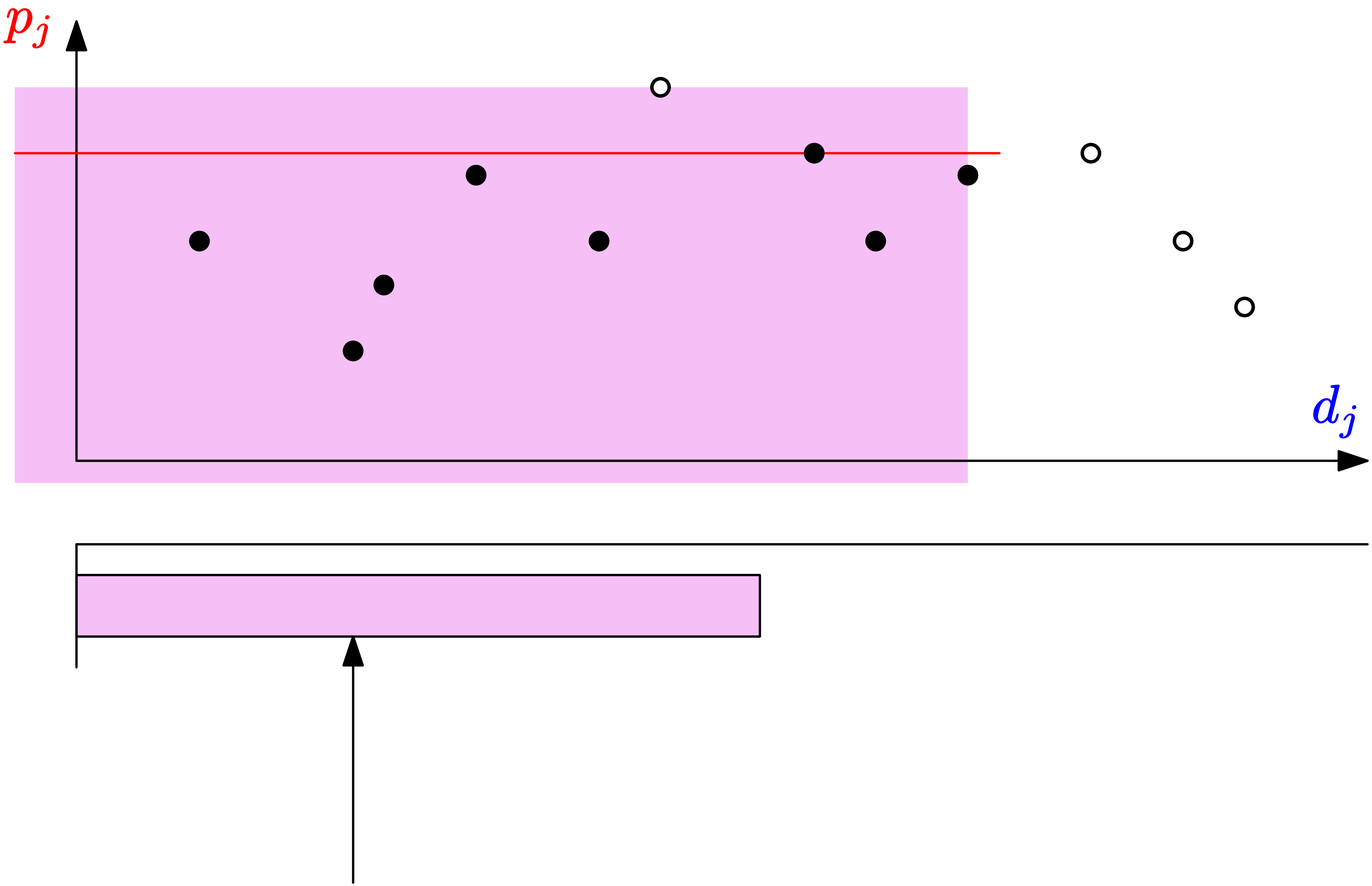
総納期遅れ　再帰的に計算する　再帰的に計算する

ある最適解の構造 (2)



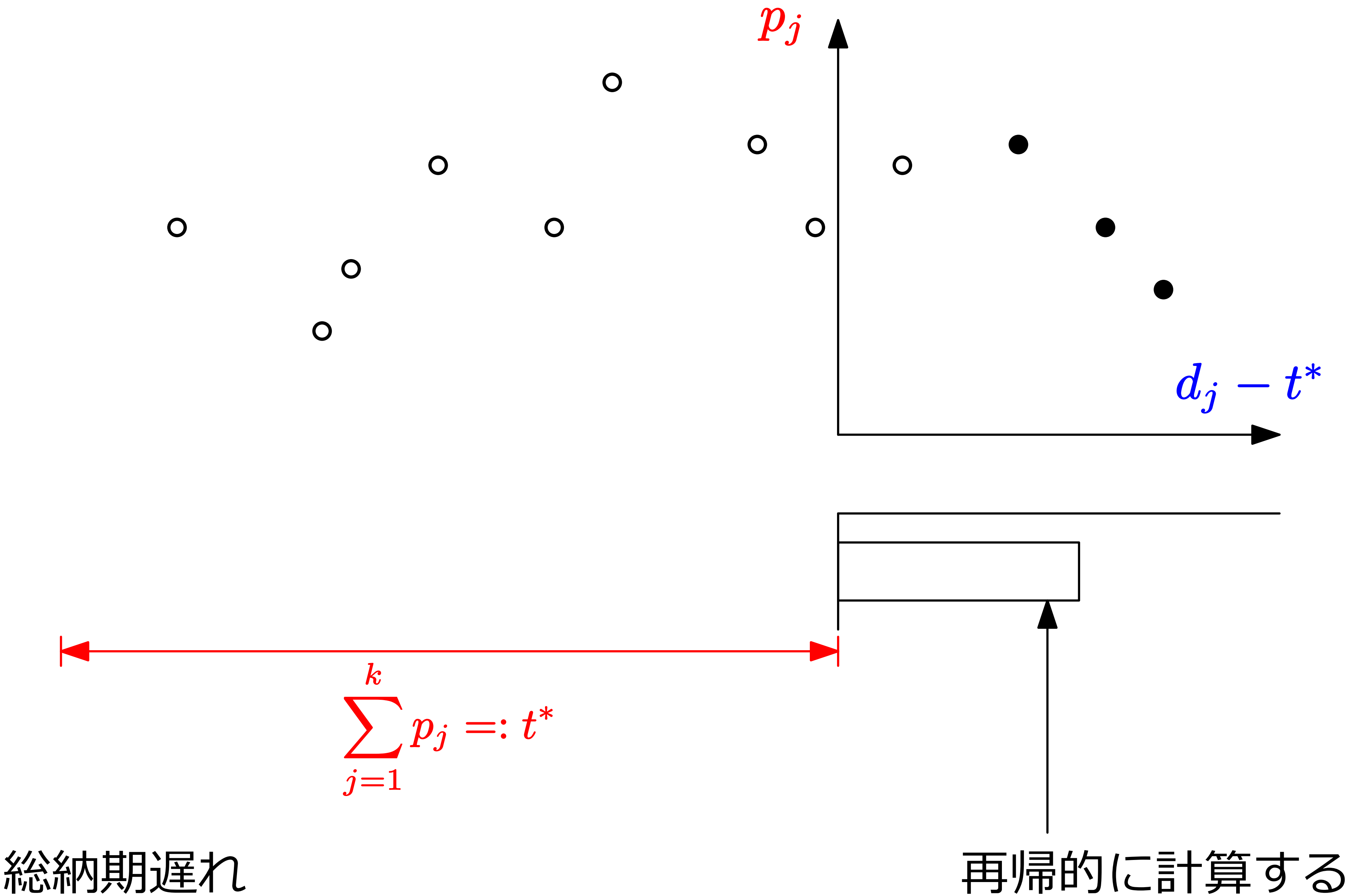
総納期遅れ　再帰的に計算する

ある最適解の構造 (2)



総納期遅れ　再帰的に計算する

ある最適解の構造 (3)



状態

$\boxed{\text{状態}}$: $(j_1, j_2, j_3, t) \in [n+1] \times [n+1] \times [n] \times (\{0\} \cup [\sum p_j])$

補題 B (処理時間最大ジョブ) (再掲)

ある k とある最適スケジュールが存在して，そこでは
$j \le k, j \ne j^*$ のジョブ J_j を J_{j^*} より前に処理し
$j > k$ のジョブ J_j を J_{j^*} より後に処理する

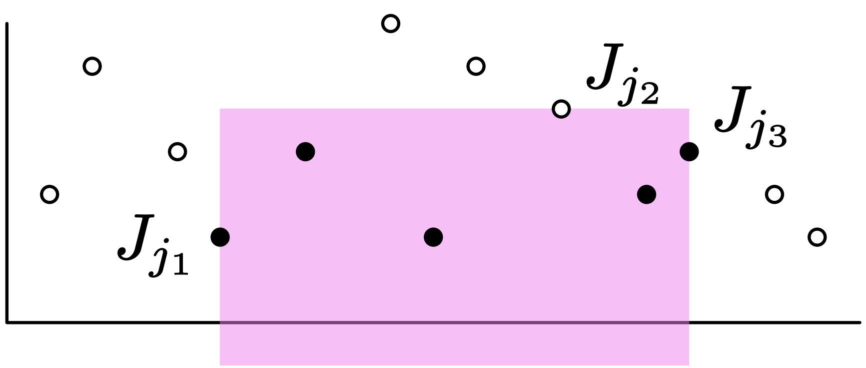
$\boxed{\text{状態の値}}$: $\mathcal{J}(j_1, j_2, j_3) = \{J_j \mid j_1 \le j \le j_3, p_j < p_{j_2}\}$ として

$f(j_1, j_2, j_3, t) =$ スタートを時刻 t とするとき，

$\mathcal{J}(j_1, j_2, j_3)$ を処理するときの

総納期遅れの最小値

$p_{n+1} = +\infty$ とする



再帰式 (Bellman 方程式) (1)

$\mathcal{J}(j_1, j_2, j_3) = \emptyset$ のとき,

$$f(j_1, j_2, j_3, t) = 0$$

$\boxed{\text{状態の値}}$: $\mathcal{J}(j_1, j_2, j_3) = \{J_j \mid j_1 \leq j \leq j_3, p_j < p_{j_2}\}$ として

$f(j_1, j_2, j_3, t) =$ スタートを時刻 t とするとき,
$\mathcal{J}(j_1, j_2, j_3)$ を処理するときの
総納期遅れの最小値

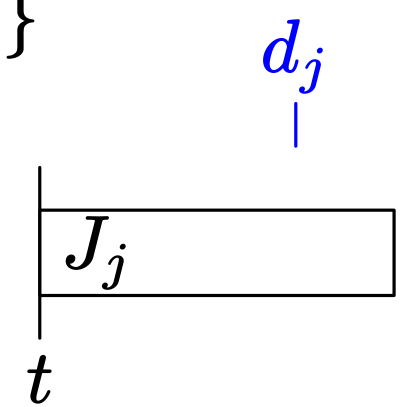
再帰式 (Bellman 方程式) (1)

$\mathcal{J}(j_1, j_2, j_3) = \emptyset$ のとき,

$$f(j_1, j_2, j_3, t) = 0$$

$\mathcal{J}(j_1, j_2, j_3) = \{J_j\}$ のとき,

$$f(j_1, j_2, j_3, t) = \max\{0, t + p_j - d_j\}$$



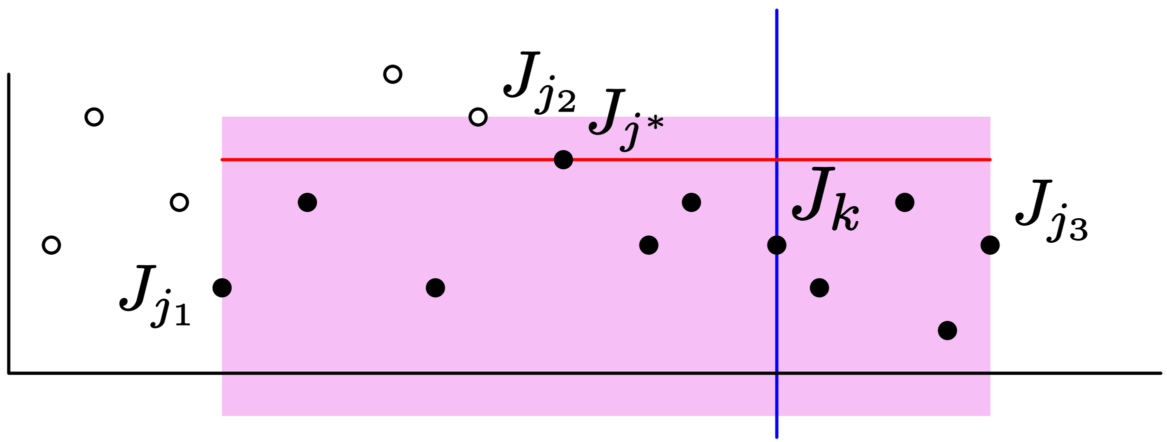
$\boxed{\text{状態の値}}$: $\mathcal{J}(j_1, j_2, j_3) = \{J_j \mid j_1 \leq j \leq j_3, p_j < p_{j_2}\}$ として

$f(j_1, j_2, j_3, t) =$ スタートを時刻 t とするとき,
$\mathcal{J}(j_1, j_2, j_3)$ を処理するときの
総納期遅れの最小値

その他のとき,

$$f(j_1, j_2, j_3, t) = \min_{j^* \le k \le j_3} \left\{ \begin{array}{l} f(j_1, j^*, k, t) \\ + \max\{0, t + t_k^* - d_{j^*}\} \\ + f(k+1, j^*, j_3, t + t_k^*) \end{array} \right\}$$

ただし, $j^* \in \arg\max\{p_j \mid J_j \in \mathcal{J}(j_1, j_2, j_3)\}$



$\boxed{\text{状態の値}}$: $\mathcal{J}(j_1, j_2, j_3) = \{J_j \mid j_1 \le j \le j_3, p_j < p_{j_2}\}$ として

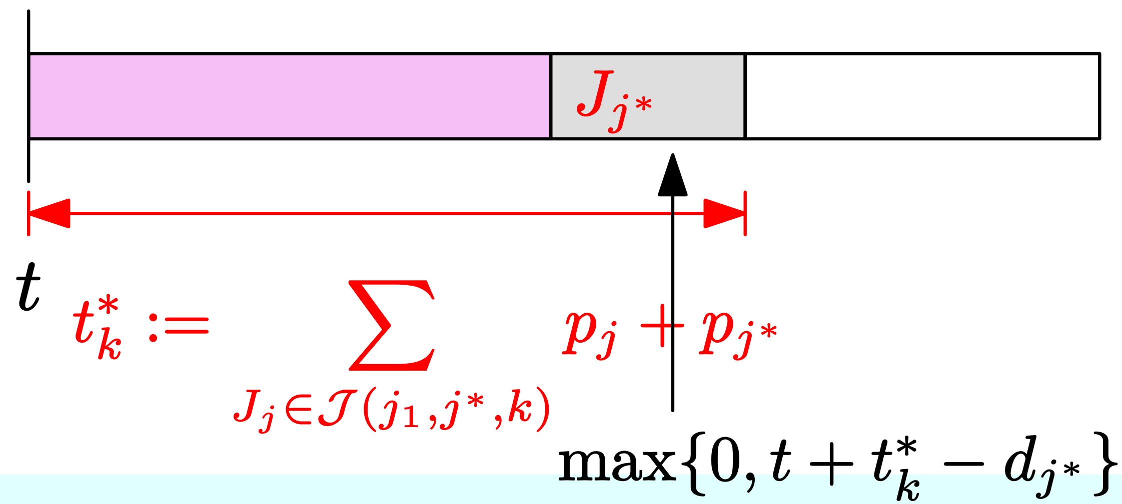
$f(j_1, j_2, j_3, t) =$ スタートを時刻 t とするとき,
$\mathcal{J}(j_1, j_2, j_3)$ を処理するときの
総納期遅れの最小値

再帰式 (Bellman 方程式) (2)

その他のとき，

$$f(j_1, j_2, j_3, t) = \min_{j^* \le k \le j_3} \left\{ \begin{array}{l} f(j_1, j^*, k, t) \\ + \max\{0, t + t_k^* - d_{j^*}\} \\ + f(k+1, j^*, j_3, t + t_k^*) \end{array} \right\}$$

ただし， $j^* \in \arg\max\{p_j \mid J_j \in \mathcal{J}(j_1, j_2, j_3)\}$



$\boxed{\text{状態の値}}$: $\mathcal{J}(j_1, j_2, j_3) = \{J_j \mid j_1 \le j \le j_3, p_j < p_{j_2}\}$ として

$f(j_1, j_2, j_3, t) =$ スタートを時刻 t とするとき，
$\mathcal{J}(j_1, j_2, j_3)$ を処理するときの
総納期遅れの最小値

アルゴリズム

$1 \mid\mid \sum T_j$ に対する動的計画法アルゴリズム

1. 各 j_1, j_2, j_3 に対して，
   $\arg\max\{p_j \mid J_j \in \mathcal{J}(j_1, j_2, j_3)\}$ を計算する
2. 各 j_1, j_2, j_3, t に対する $f(j_1, j_2, j_3, t)$ の値を
   再帰式に基づいて，j_1, j_2, j_3, t が小さい方から定める
3. $f(1, n+1, n, 0)$ を出力する
   (ただし，$j^* \in \arg\max\{p_j \mid j \in [n]\}$)

その他のとき，

$$f(j_1, j_2, j_3, t) = \min_{j^* \le k \le j_3} \left\{ \begin{array}{l} f(j_1, j^*, k, t) \\ + \max\{0, t + t_k^* - d_{j^*}\} \\ + f(k+1, j^*, j_3, t + t_k^*) \end{array} \right\}$$

ただし，$j^* \in \arg\max\{p_j \mid J_j \in \mathcal{J}(j_1, j_2, j_3)\}$

$1 \mid\mid \sum T_j$ に対する動的計画法アルゴリズム

1. 各 j_1, j_2, j_3 に対して，
   $\arg\max\{p_j \mid J_j \in \mathcal{J}(j_1, j_2, j_3)\}$ を計算する
2. 各 j_1, j_2, j_3, t に対する $f(j_1, j_2, j_3, t)$ の値を
   再帰式に基づいて，j_1, j_2, j_3, t が小さい方から定める
3. $f(1, n+1, n, 0)$ を出力する
   (ただし，$j^* \in \arg\max\{p_j \mid j \in [n]\}$)

<u>正当性</u>：f の定義と再帰式から正当性が分かる □

$\boxed{\text{状態の値}}$：$\mathcal{J}(j_1, j_2, j_3) = \{J_j \mid j_1 \le j \le j_3, p_j < p_{j_2}\}$ として

$f(j_1, j_2, j_3, t) =$ スタートを時刻 t とするとき，
$\mathcal{J}(j_1, j_2, j_3)$ を処理するときの
総納期遅れの最小値

$1 \mid\mid \sum T_j$ に対する動的計画法アルゴリズム

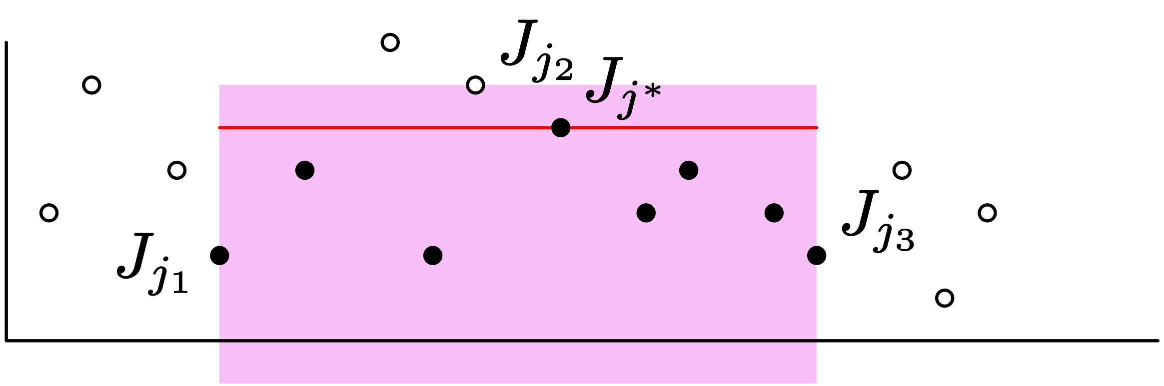
1. 各 j_1, j_2, j_3 に対して，
   $\arg\max\{p_j \mid J_j \in \mathcal{J}(j_1, j_2, j_3)\}$ を計算する
2. 各 j_1, j_2, j_3, t に対する $f(j_1, j_2, j_3, t)$ の値を
   再帰式に基づいて，j_1, j_2, j_3, t が小さい方から定める
3. $f(1, n+1, n, 0)$ を出力する
   (ただし，$j^* \in \arg\max\{p_j \mid j \in [n]\}$)

<u>計算量 0</u>：EDD 順にジョブを並べる $\leadsto$ $O(n \log n)$ 時間

$1 \,||\, \sum T_j$ に対する動的計画法アルゴリズム

1. 各 j_1, j_2, j_3 に対して，
   $\arg\max\{p_j \mid J_j \in \mathcal{J}(j_1, j_2, j_3)\}$ を計算する
2. 各 j_1, j_2, j_3, t に対する $f(j_1, j_2, j_3, t)$ の値を
   再帰式に基づいて，j_1, j_2, j_3, t が小さい方から定める
3. $f(1, n+1, n, 0)$ を出力する
   (ただし，$j^* \in \arg\max\{p_j \mid j \in [n]\}$)

計算量 1：各 j_1, j_2, j_3 に対して，$O(n)$ 時間　$\rightsquigarrow O(n^4)$ 時間



$1 \,||\, \sum T_j$ に対する動的計画法アルゴリズム

1. 各 j_1, j_2, j_3 に対して，
   $\arg\max\{p_j \mid J_j \in \mathcal{J}(j_1, j_2, j_3)\}$ を計算する
2. 各 j_1, j_2, j_3, t に対する $f(j_1, j_2, j_3, t)$ の値を
   再帰式に基づいて，j_1, j_2, j_3, t が小さい方から定める
3. $f(1, n+1, n, 0)$ を出力する
   (ただし，$j^* \in \arg\max\{p_j \mid j \in [n]\}$)

<u>計算量 2</u>：各 j_1, j_2, j_3, t に対して，$O(n)$ 時間

$\rightsquigarrow$ $O(n^4 \sum p_j)$ 時間

$$f(j_1, j_2, j_3, t) = \min_{j^* \le k \le j_3} \left\{ \begin{array}{l} f(j_1, j^*, k, t) \\ + \max\{0, t + t_k^* - d_{j^*}\} \\ + f(k+1, j^*, j_3, t + t_k^*) \end{array} \right\}$$

$1 \,||\, \sum T_j$ に対する動的計画法アルゴリズム

1. 各 j_1, j_2, j_3 に対して，
   $\arg\max\{p_j \mid J_j \in \mathcal{J}(j_1, j_2, j_3)\}$ を計算する
2. 各 j_1, j_2, j_3, t に対する $f(j_1, j_2, j_3, t)$ の値を
   再帰式に基づいて，j_1, j_2, j_3, t が小さい方から定める
3. $f(1, n+1, n, 0)$ を出力する
   (ただし，$j^* \in \arg\max\{p_j \mid j \in [n]\}$)

計算量 3：$O(1)$ 時間

アルゴリズム：計算量 まとめ

$1 \,||\, \sum T_j$ に対する動的計画法アルゴリズム

1. 各 j_1, j_2, j_3 に対して，
   $\arg\max\{p_j \mid J_j \in \mathcal{J}(j_1, j_2, j_3)\}$ を計算する
2. 各 j_1, j_2, j_3, t に対する $f(j_1, j_2, j_3, t)$ の値を
   再帰式に基づいて，j_1, j_2, j_3, t が小さい方から定める
3. $f(1, n+1, n, 0)$ を出力する
   (ただし，$j^* \in \arg\max\{p_j \mid j \in [n]\}$)

計算量：合わせて $O(n^4 \sum p_j)$ 時間 □

0. $O(n \log n)$ 時間
1. $O(n^4)$ 時間
2. $O(n^4 \sum p_j)$ 時間
3. $O(1)$ 時間

総納期遅れ最小化：まとめ

定理：$1 \,||\, \sum T_j$ に対するアルゴリズム　(Lawler '77)

$1 \,||\, \sum T_j$ は $O(n^4 \sum p_j)$ 時間で解ける

$\boxed{\text{注意}}$：**擬多項式時間** の計算量 (詳細は次回)

$\boxed{\text{比較}}$

「自明」	$O(n! \cdot n)$	指数時間
動的計画法	$O(n^4 \sum p_j)$	擬多項式時間

議論を簡単にするため，すべてのジョブの処理時間は異なるとする

まとめ：一機械，ジョブの特性なし

$\boxed{\text{まとめ}}$：$1 \mid\mid \gamma$

γ		計算量	(文献)
$C_{\max}$	最大完了時刻	$O(n)$	
$\sum C_j$	総完了時刻	$O(n \log n)$	(Smith ’56)
$L_{\max}$	最大納期ずれ	$O(n \log n)$	(Jackson ’55)
$\sum T_j$	総納期遅れ	$O(n^4 \sum p_j)$	(Lawler ’77)
		NP 困難	(Du, Leung ’90)
$\sum U_j$	納期遅れジョブ数	$O(n \log n)$	(Moore ’68)

まとめと次回の予告

動的計画法を考えるときの鍵

1. 最適解の持つ **再帰的な構造** を見出す
2. 上の構造から，**状態** を適切に定義する
3. 状態の間の **再帰式** を立てる

教訓：アルゴリズム設計全般において

うまくいかない場合を考える／気に留める

次回の予告

計算複雑性を議論する方法

- $P2 \mid\mid C_{\max}$， $P \mid\mid C_{\max}$
- $1 \mid r_j \mid L_{\max}$

付録

- **補題 A の証明**
- 補題 C の証明
- 補題 B の証明
- 動的計画法アルゴリズムの動作例

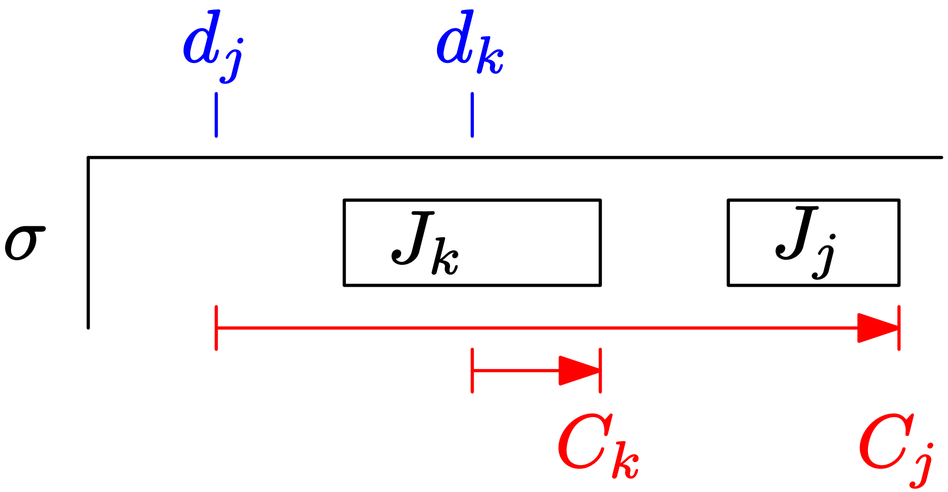
補題 A の証明 (1)

補題 A (優越ジョブ) (再掲)

$1 \,||\, \sum T_j$ において，$p_j \leq p_k$ かつ $d_j \leq d_k$ であるとき，
J_j を J_k より先に処理する最適スケジュールが存在する

証明：σ を最適スケジュール，$p_j \leq p_k$ かつ $d_j \leq d_k$ とする

- σ における J_j, J_k の完了時刻を C_j, C_k とする
- $C_k < C_j$ のとき，d_j, d_k, C_j, C_k の大小順は次のどれか
  1. $C_k < C_j \leq d_j \leq d_k$
  2. $C_k \leq d_j \leq C_j \leq d_k$
  3. $C_k \leq d_j \leq d_k \leq C_j$
  4. $d_j \leq C_k < C_j \leq d_k$
  5. $d_j \leq C_k \leq d_k \leq C_j$
  6. $d_j \leq d_k \leq C_k \leq C_j$



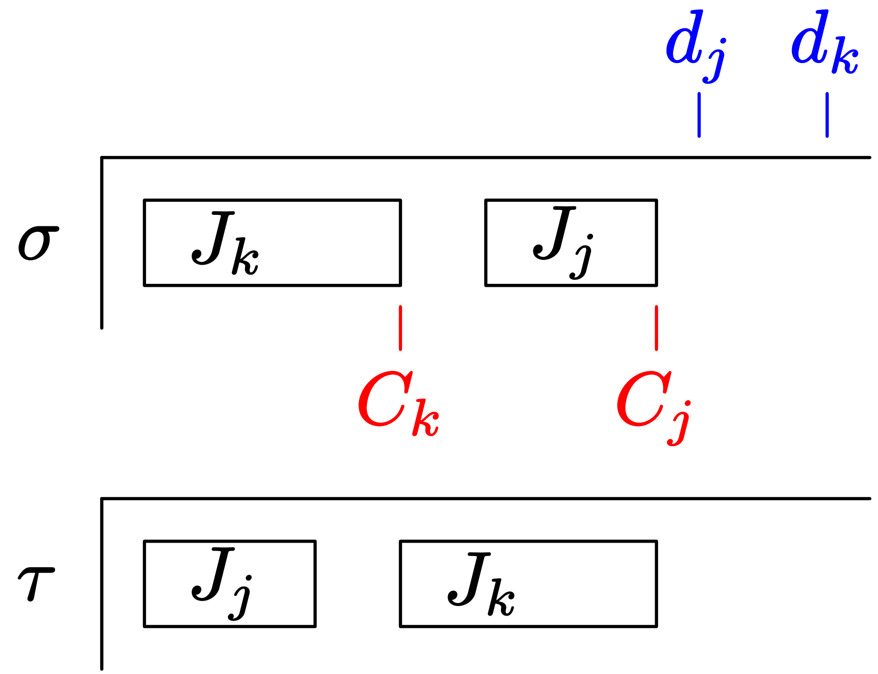
- $\tau = \sigma$ の J_j と J_k の処理順を交換したもの とする

補題 A の証明 (2)

1. $C_k < C_j \leq d_j \leq d_k$ のとき：

$$\sum_{j'} T_{j'}(\sigma) - \sum_{j'} T_{j'}(\tau)$$
$$\geq T_j(\sigma) + T_k(\sigma) - T_j(\tau) - T_k(\tau)$$
$$= 0 + 0 - 0 - 0 = 0$$

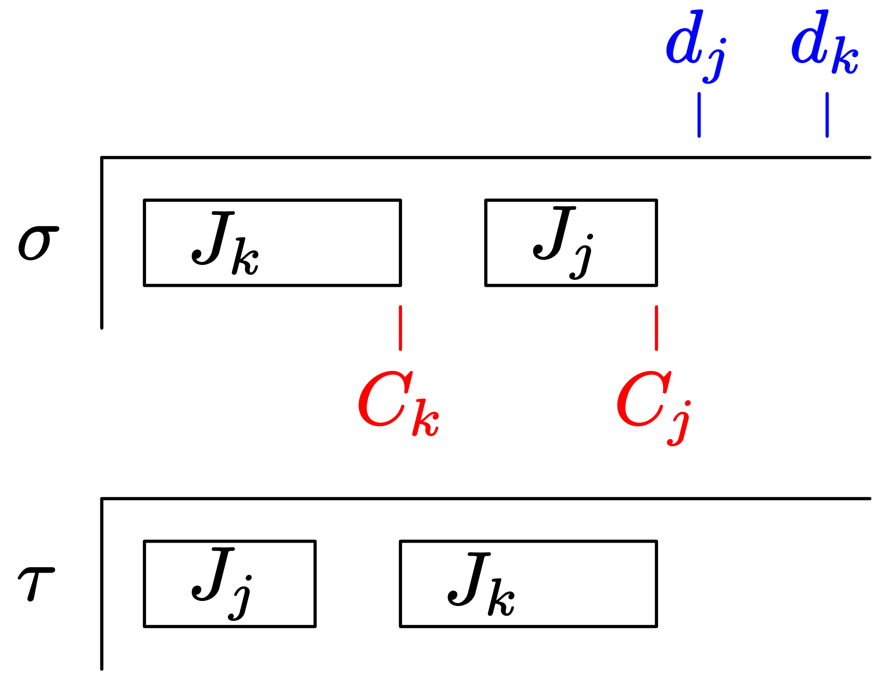
$\therefore \tau$ も最適スケジュール



1. $C_k < C_j \le d_j \le d_k$ のとき：

$$\sum_{j'} T_{j'}(\sigma) - \sum_{j'} T_{j'}(\tau)$$
$$\ge T_j(\sigma) + T_k(\sigma) - T_j(\tau) - T_k(\tau)$$
$$= 0 + 0 - 0 - 0 = 0$$

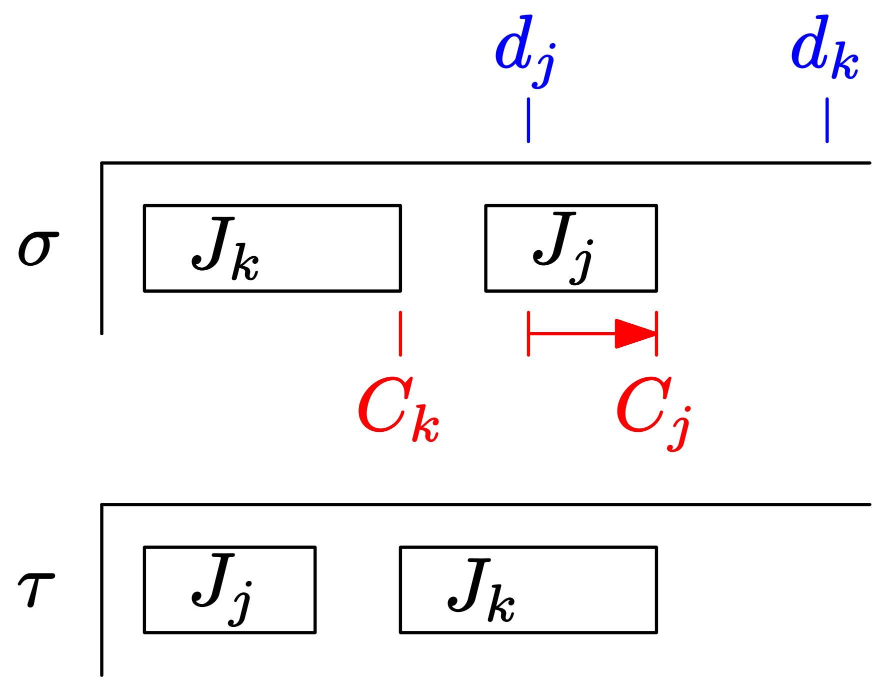
$\therefore \tau$ も最適スケジュール



2. $C_k \le d_j \le C_j \le d_k$ のとき：

$$\sum_{j'} T_{j'}(\sigma) - \sum_{j'} T_{j'}(\tau)$$
$$\ge T_j(\sigma) + T_k(\sigma) - T_j(\tau) - T_k(\tau)$$
$$\ge T_j(\sigma) + 0 - 0 - 0 \ge 0$$

$\therefore \tau$ も最適スケジュール

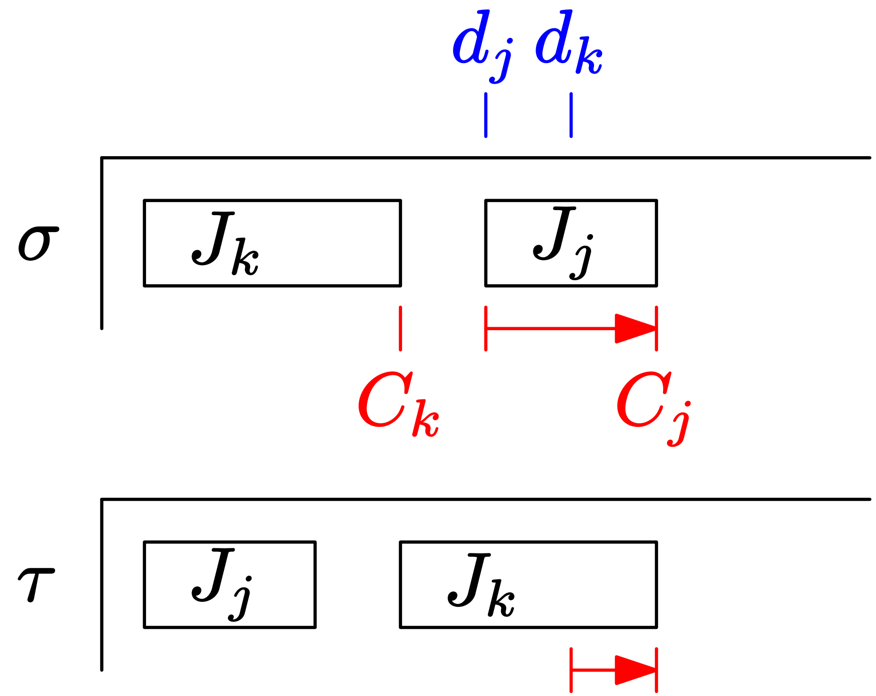


補題 A の証明 (3)

3. $C_k \leq d_j \leq d_k \leq C_j$ のとき：

$$\sum_{j'} T_{j'}(\sigma) - \sum_{j'} T_{j'}(\tau)$$
$$\geq T_j(\sigma) + T_k(\sigma) - T_j(\tau) - T_k(\tau)$$
$$= T_j(\sigma) - T_k(\tau) \geq 0$$

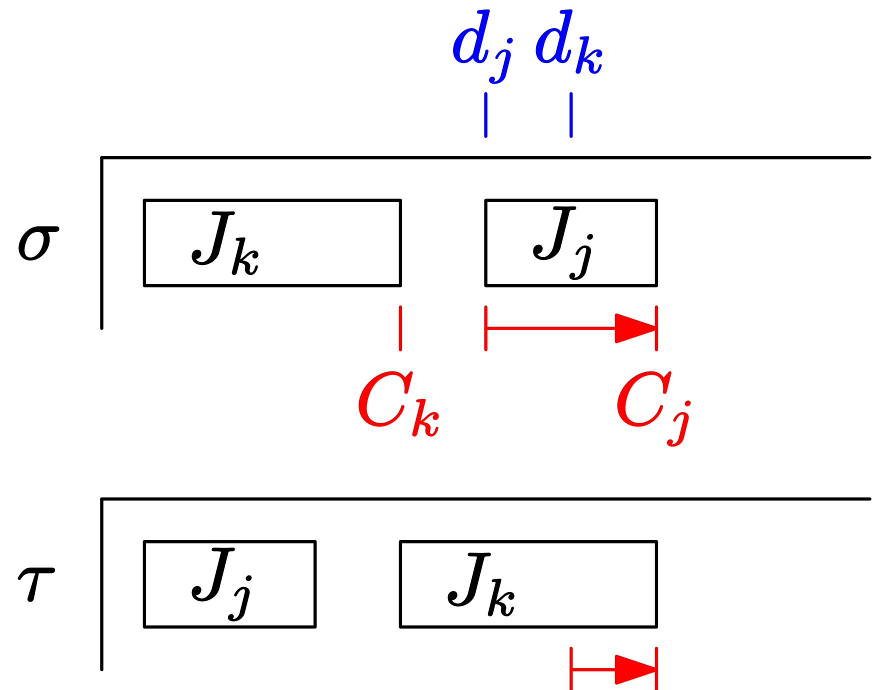
$d_j \leq d_k$



3. $C_k \le d_j \le d_k \le C_j$ のとき：

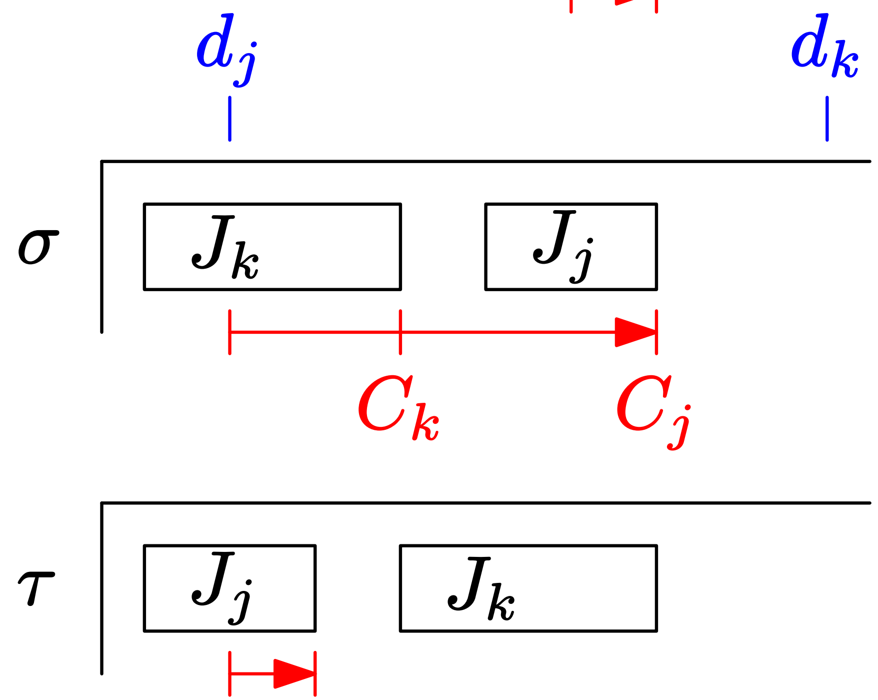
$$\sum_{j'} T_{j'}(\sigma) - \sum_{j'} T_{j'}(\tau)$$
$$\ge T_j(\sigma) + T_k(\sigma) - T_j(\tau) - T_k(\tau)$$
$$= T_j(\sigma) - T_k(\tau) \ge 0$$

$d_j \le d_k$



4. $d_j \le C_k < C_j \le d_k$ のとき：

$$\sum_{j'} T_{j'}(\sigma) - \sum_{j'} T_{j'}(\tau)$$
$$\ge T_j(\sigma) + T_k(\sigma) - T_j(\tau) - T_k(\tau)$$
$$\ge T_j(\sigma) - T_j(\tau) \ge 0$$



5. $d_j \le C_k \le d_k \le C_j$ のとき：

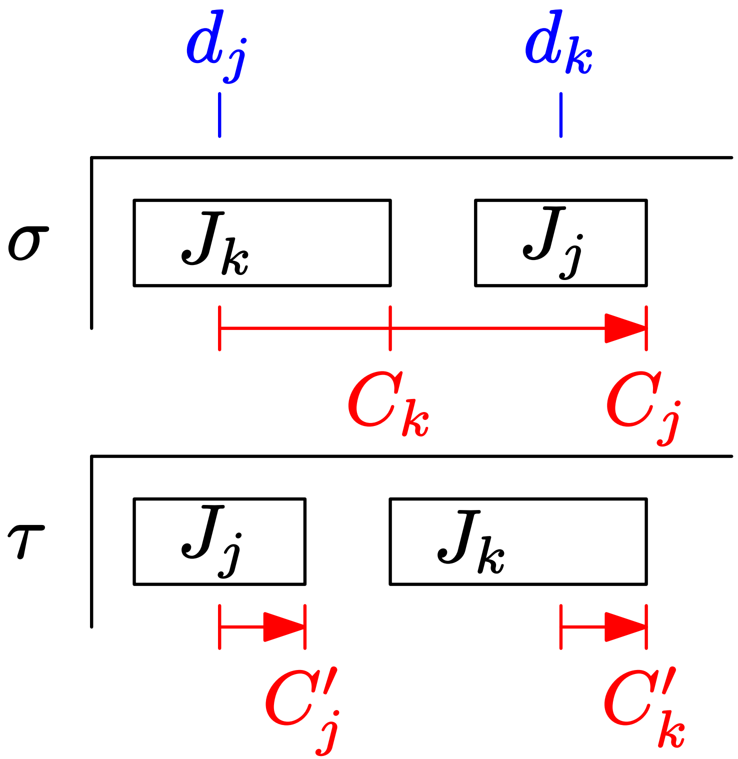
$$\sum_{j'} T_{j'}(\sigma) - \sum_{j'} T_{j'}(\tau)$$

$$\ge T_j(\sigma) + T_k(\sigma) - T_j(\tau) - T_k(\tau)$$

$$= (C_j - d_j) + 0$$

$$- \max\{0, C'_j - d_j\} - (C'_k - d_k)$$

$$\ge \underbrace{}_{p_j \le p_k} - d_j + d_k - \max\{0, C_k - d_j\}$$



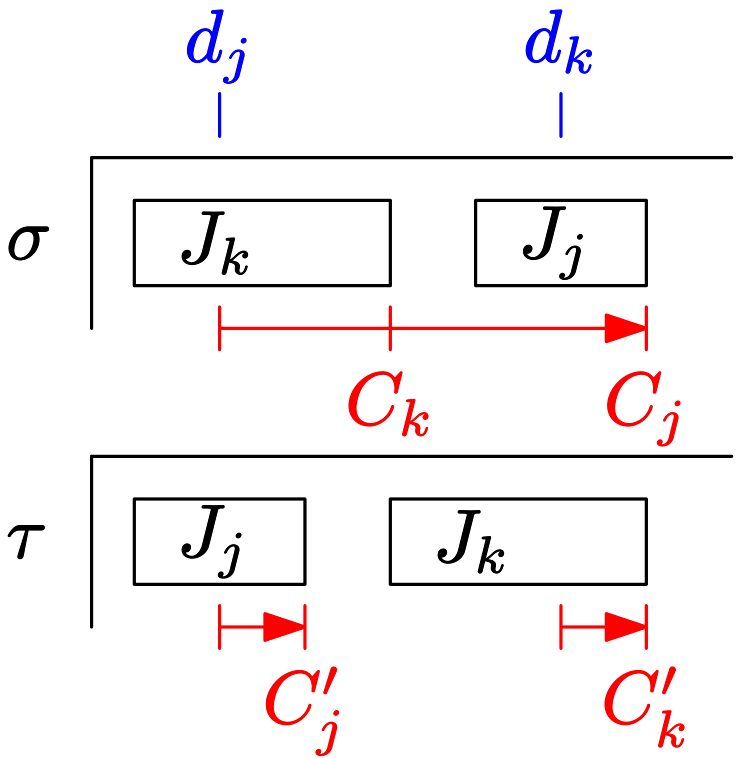
5. $d_j \le C_k \le d_k \le C_j$ のとき：

$$
\begin{aligned}
&\textstyle\sum_{j'} T_{j'}(\sigma) - \sum_{j'} T_{j'}(\tau) \\
&\ge T_j(\sigma) + T_k(\sigma) - T_j(\tau) - T_k(\tau) \\
&= (C_j - d_j) + 0 \\
&\qquad - \max\{0, C'_j - d_j\} - (C'_k - d_k) \\
&\ge -d_j + d_k - \max\{0, C_k - d_j\}
\end{aligned}
$$

$(0 \ge C_k - d_j$ のとき$)$

$$
= -d_j + d_k - 0 \ge 0
$$

$d_j \le d_k$



5. $d_j \leq C_k \leq d_k \leq C_j$ のとき：

$$\textstyle\sum_{j'} T_{j'}(\sigma) - \sum_{j'} T_{j'}(\tau)$$

$$\geq T_j(\sigma) + T_k(\sigma) - T_j(\tau) - T_k(\tau)$$

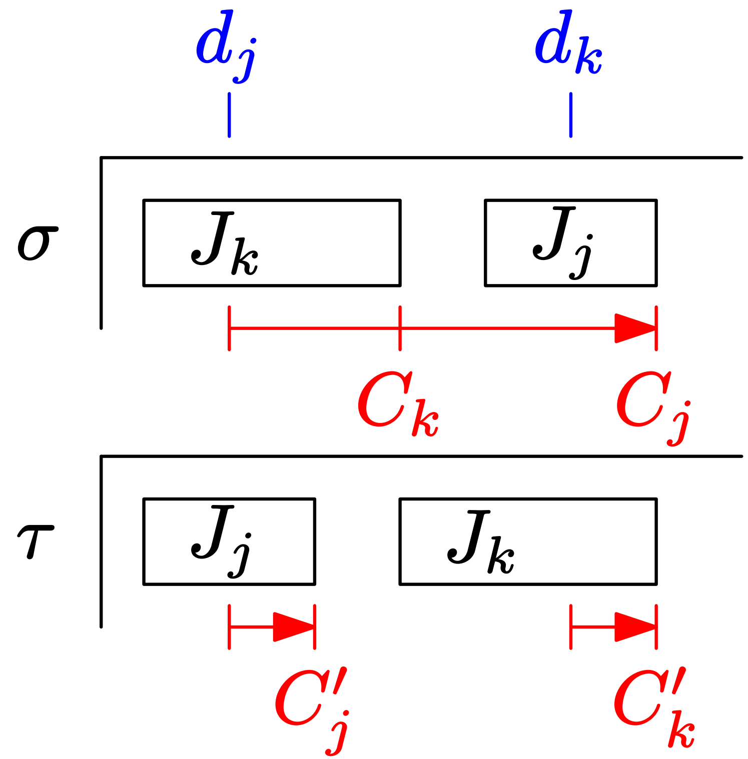
$$= (C_j - d_j) + 0$$

$$\quad - \max\{0, C'_j - d_j\} - (C'_k - d_k)$$

$$\geq \; -d_j + d_k - \max\{0, C_k - d_j\}$$

($0 \leq C_k - d_j$ のとき)

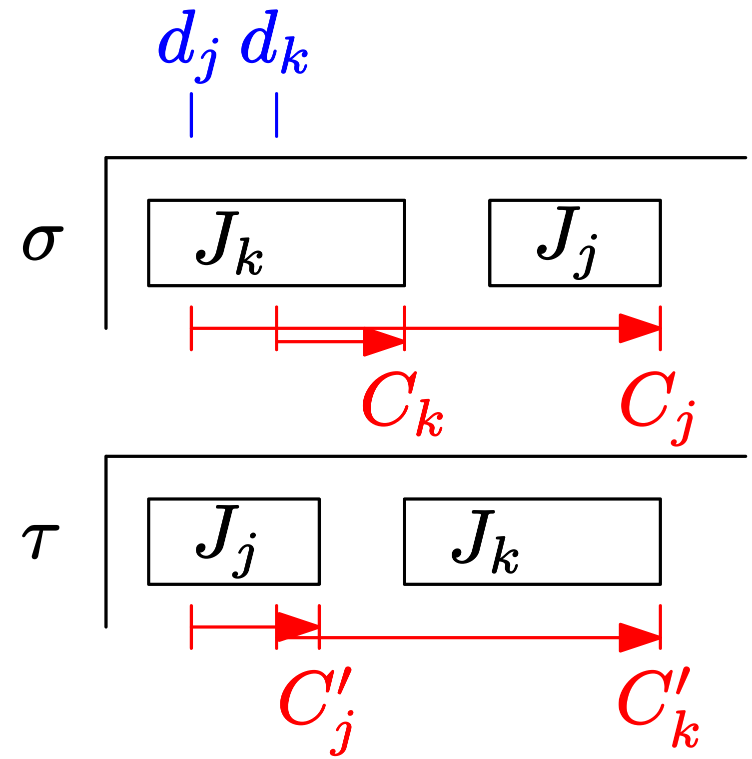
$$= \; -d_j + d_k - (C_k - d_j) = d_k - C_k \geq 0$$



6. $d_j \le d_k \le C_k \le C_j$ のとき：

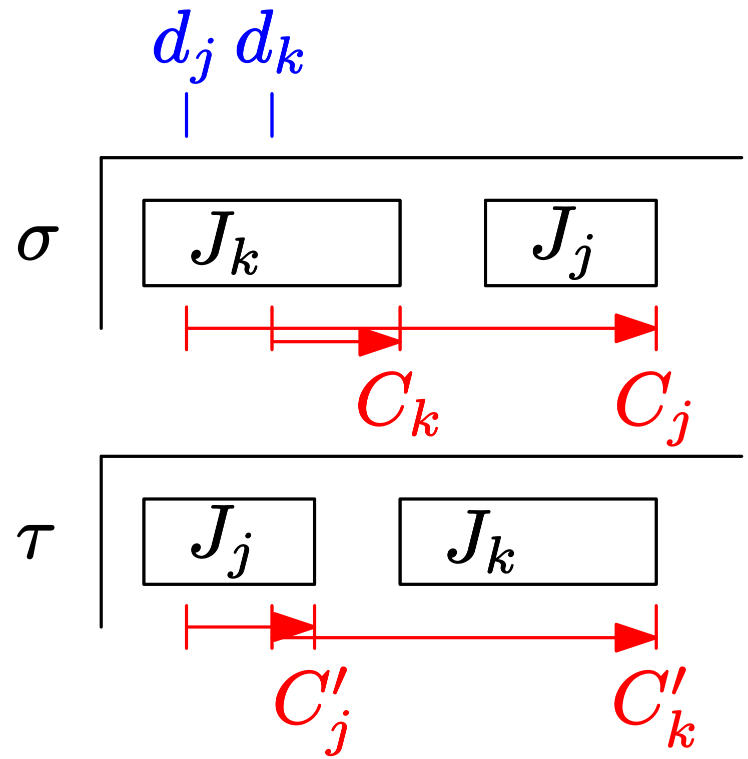
$$\sum_{j'} T_{j'}(\sigma) - \sum_{j'} T_{j'}(\tau)$$
$$\ge T_j(\sigma) + T_k(\sigma) - T_j(\tau) - T_k(\tau)$$
$$= (C_j - d_j) + (C_k - d_k)$$
$$\quad - \max\{0, C'_j - d_j\} - (C'_k - d_k)$$
$$\ge C_k - d_j - \max\{0, C_k - d_j\}$$

$p_j \le p_k$



6. $d_j \leq d_k \leq C_k \leq C_j$ のとき：

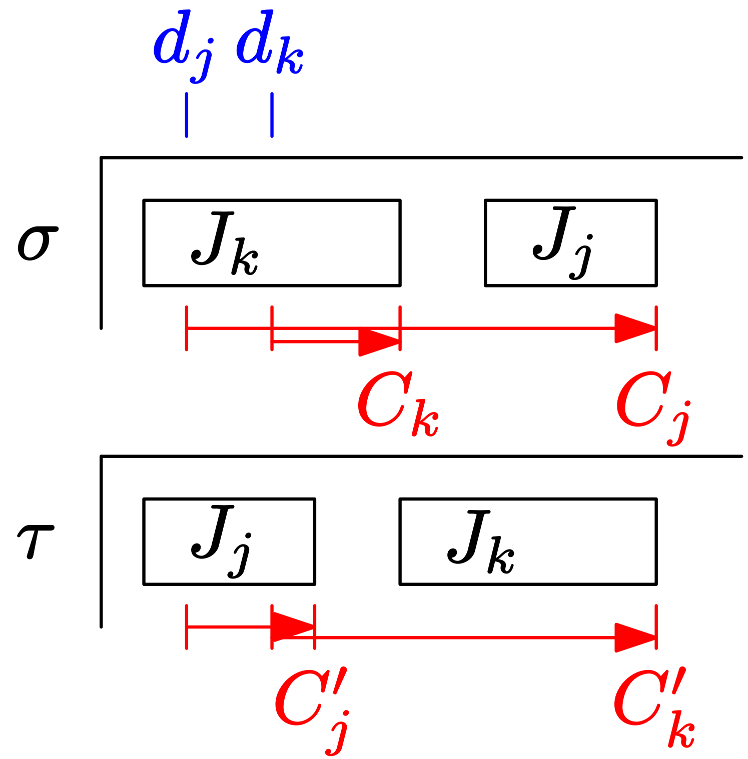
$$
\begin{aligned}
&\textstyle\sum_{j'} T_{j'}(\sigma) - \sum_{j'} T_{j'}(\tau) \\
&\geq T_j(\sigma) + T_k(\sigma) - T_j(\tau) - T_k(\tau) \\
&= (C_j - d_j) + (C_k - d_k) \\
&\quad - \max\{0, C'_j - d_j\} - (C'_k - d_k) \\
&\geq C_k - d_j - \max\{0, C_k - d_j\} \\
&= C_k - d_j - (C_k - d_j) = 0
\end{aligned}
$$



補題 A の証明 (5)

6. $d_j \le d_k \le C_k \le C_j$ のとき：

$$
\begin{aligned}
&\textstyle\sum_{j'} T_{j'}(\sigma) - \sum_{j'} T_{j'}(\tau) \\
&\ge T_j(\sigma) + T_k(\sigma) - T_j(\tau) - T_k(\tau) \\
&= (C_j - d_j) + (C_k - d_k) \\
&\quad - \max\{0, C'_j - d_j\} - (C'_k - d_k) \\
&\ge C_k - d_j - \max\{0, C_k - d_j\} \\
&= C_k - d_j - (C_k - d_j) = 0
\end{aligned}
$$



$\therefore$ すべての場合で τ も最適スケジュール $\square$

付録

- 補題 A の証明
- **補題 C の証明**
- 補題 B の証明
- 動的計画法アルゴリズムの動作例

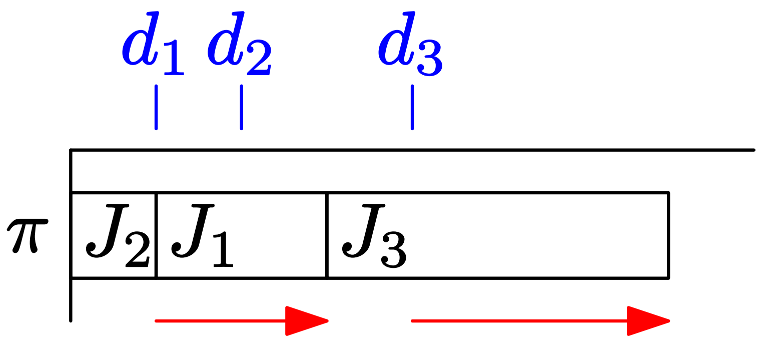
補題 C

補題 B の証明のために，次の補題 C を使う

仮定：$1 \,||\, \sum T_j$ の設定で，ジョブは EDD 順 ($d_1 \leq \cdots \leq d_n$)

補題 C

π : $d_1, \ldots, d_n$ に対する最適スケジュール
C_j : π におけるジョブ J_j の完了時刻
d'_j : $\min\{d_j, C_j\} \leq d'_j \leq \max\{d_j, C_j\}$ を満たす
π' : $d'_1, \ldots, d'_n$ に対する最適スケジュール
$\Rightarrow \pi'$ は $d_1, \ldots, d_n$ に対しても最適スケジュール



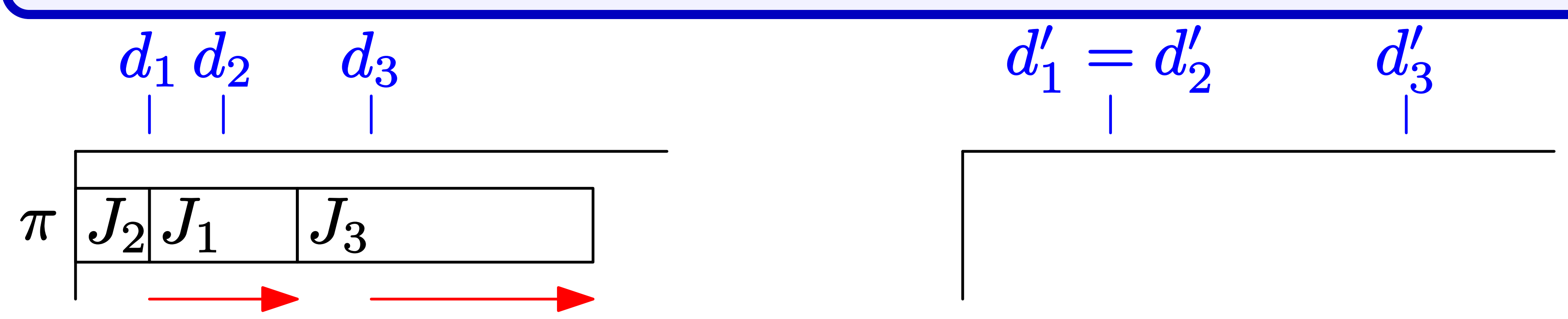
補題 C

補題 B の証明のために，次の補題 C を使う

仮定：$1 \mid\mid \sum T_j$ の設定で，ジョブは EDD 順 ($d_1 \leq \cdots \leq d_n$)

補題 C

π : $d_1, \ldots, d_n$ に対する最適スケジュール
C_j : π におけるジョブ J_j の完了時刻
d'_j : $\min\{d_j, C_j\} \leq d'_j \leq \max\{d_j, C_j\}$ を満たす
π' : $d'_1, \ldots, d'_n$ に対する最適スケジュール
$\Rightarrow$ π' は $d_1, \ldots, d_n$ に対しても最適スケジュール



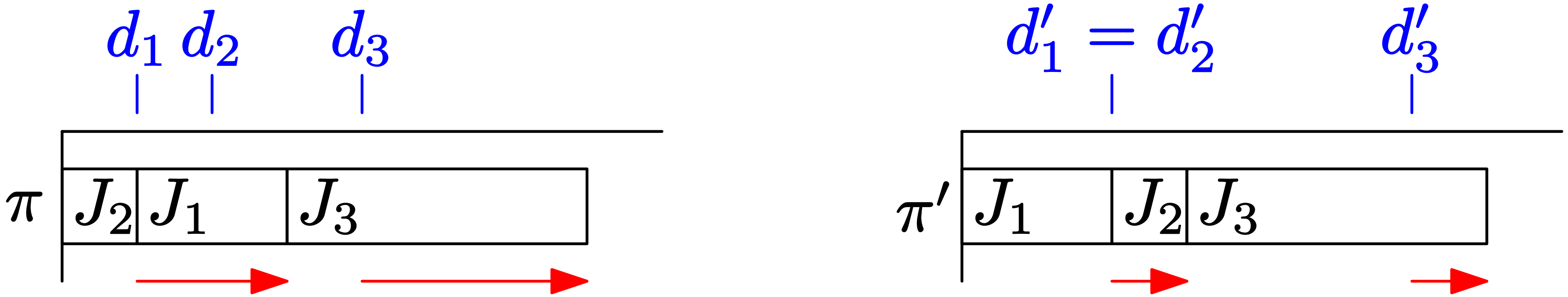
補題 C

補題 B の証明のために，次の補題 C を使う

仮定：$1 \mid\mid \sum T_j$ の設定で，ジョブは EDD 順 ($d_1 \le \cdots \le d_n$)

補題 C

π : $d_1, \ldots, d_n$ に対する最適スケジュール
C_j : π におけるジョブ J_j の完了時刻
d'_j : $\min\{d_j, C_j\} \le d'_j \le \max\{d_j, C_j\}$ を満たす
π' : $d'_1, \ldots, d'_n$ に対する最適スケジュール
$\Rightarrow \pi'$ は $d_1, \ldots, d_n$ に対しても最適スケジュール



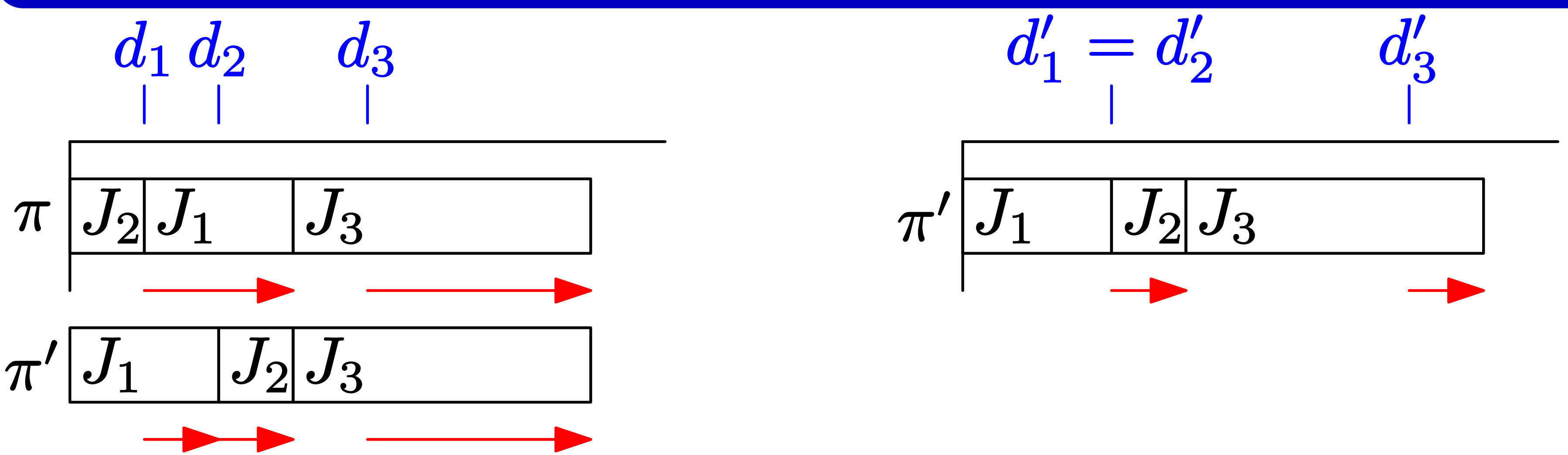
補題 C

補題 B の証明のために，次の補題 C を使う

仮定：$1 \,||\, \sum T_j$ の設定で，ジョブは EDD 順 ($d_1 \leq \cdots \leq d_n$)

補題 C

π : $d_1, \ldots, d_n$ に対する最適スケジュール
C_j : π におけるジョブ J_j の完了時刻
d'_j : $\min\{d_j, C_j\} \leq d'_j \leq \max\{d_j, C_j\}$ を満たす
π' : $d'_1, \ldots, d'_n$ に対する最適スケジュール
$\Rightarrow \pi'$ は $d_1, \ldots, d_n$ に対しても最適スケジュール

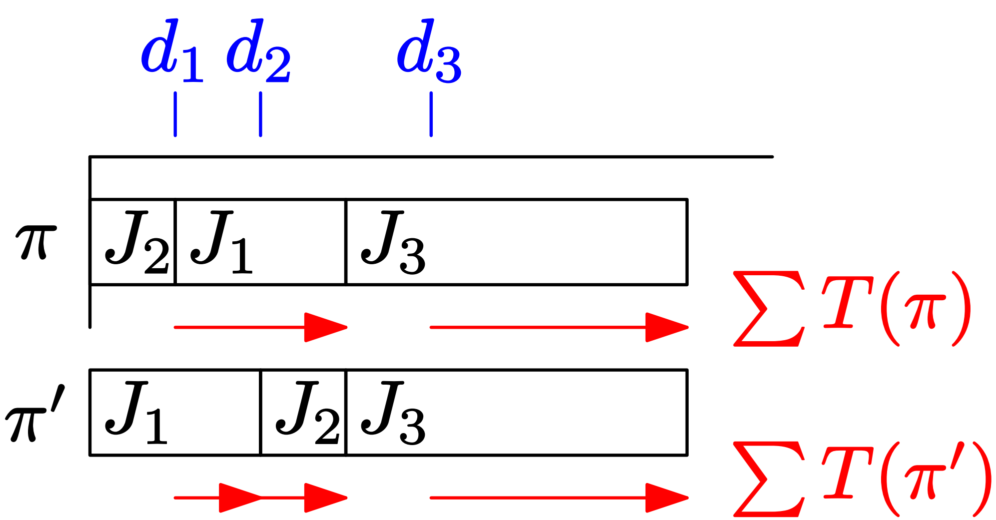


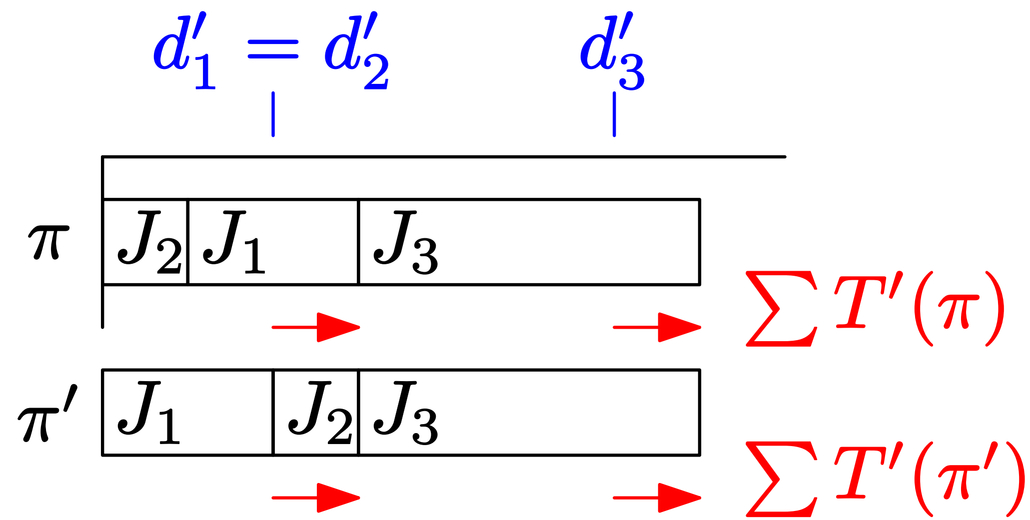
補題 C：証明 (1)

証明：次の記法を使って証明を進めていく

- $\sum T(\pi) = d_1, \ldots, d_n$ に対する π の総納期遅れ
- $\sum T(\pi') = d_1, \ldots, d_n$ に対する π' の総納期遅れ
- $\sum T'(\pi) = d'_1, \ldots, d'_n$ に対する π の総納期遅れ
- $\sum T'(\pi') = d'_1, \ldots, d'_n$ に対する π' の総納期遅れ

証明したい目標：$\sum T(\pi') \leq \sum T(\pi)$





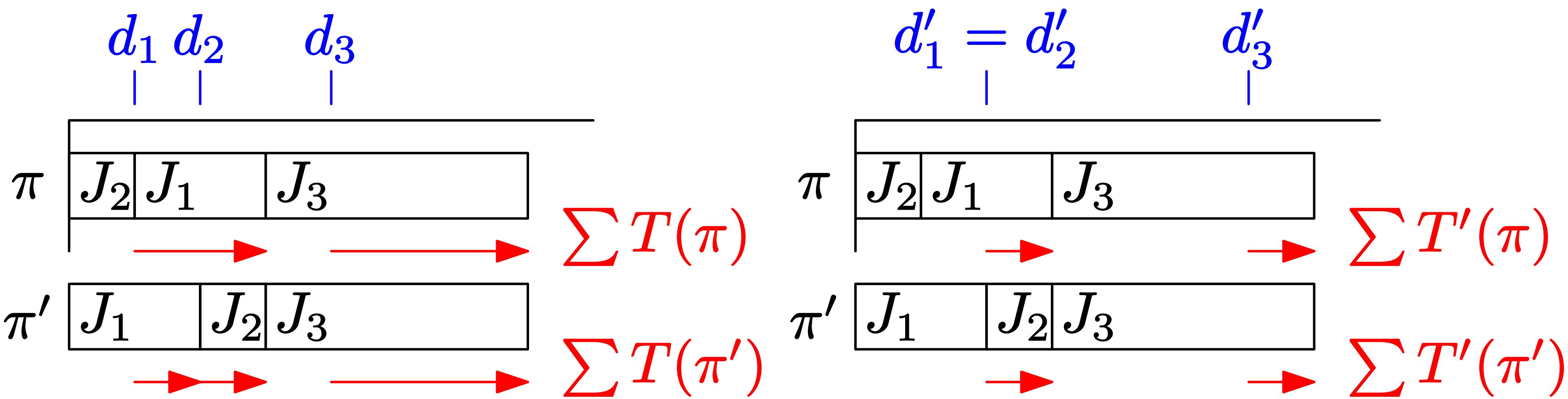
証明したい目標：$\sum T(\pi') \leq \sum T(\pi)$

そのために，次の 2 つを証明する

1. $\sum T(\pi) - \sum T'(\pi) \geq \sum T(\pi') - \sum T'(\pi')$
2. $\sum T'(\pi) \geq \sum T'(\pi')$

この 2 つが証明できれば，次が得られて証明が終わる

$$\sum T(\pi') \leq \sum T(\pi) - \sum T'(\pi) + \sum T'(\pi') \leq T(\pi)$$



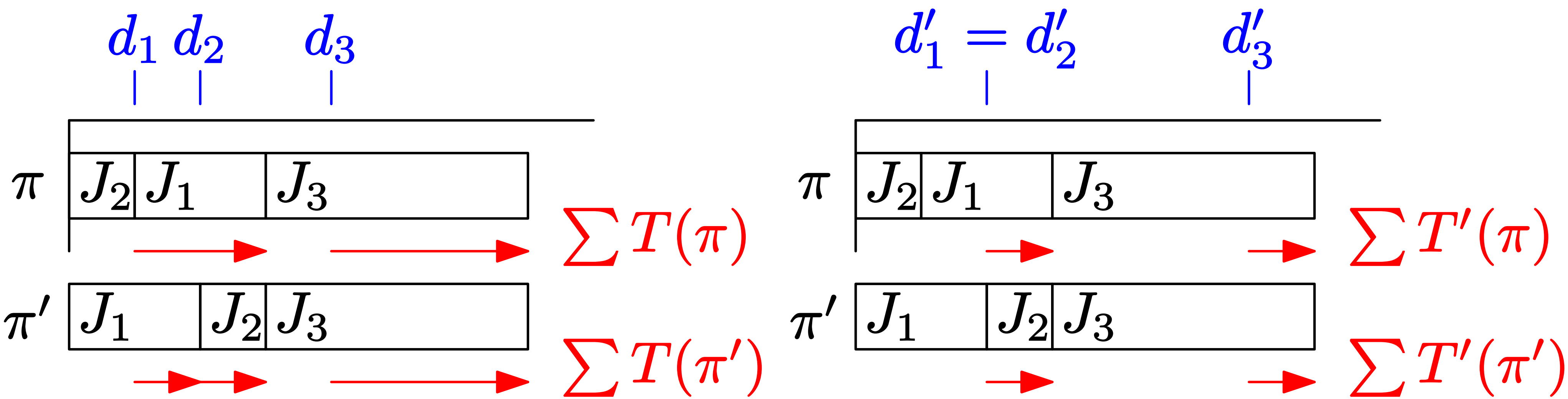
証明したい目標：$\sum T(\pi') \leq \sum T(\pi)$

そのために，次の 2 つを証明する

1. $\sum T(\pi) - \sum T'(\pi) \geq \sum T(\pi') - \sum T'(\pi')$
2. $\sum T'(\pi) \geq \sum T'(\pi')$　$\because d'_1, \ldots, d'_n$ に対して π' が最適

この 2 つが証明できれば，次が得られて証明が終わる

$$\sum T(\pi') \leq \sum T(\pi) - \sum T'(\pi) + \sum T'(\pi') \leq T(\pi)$$



補題 C：証明 (3)

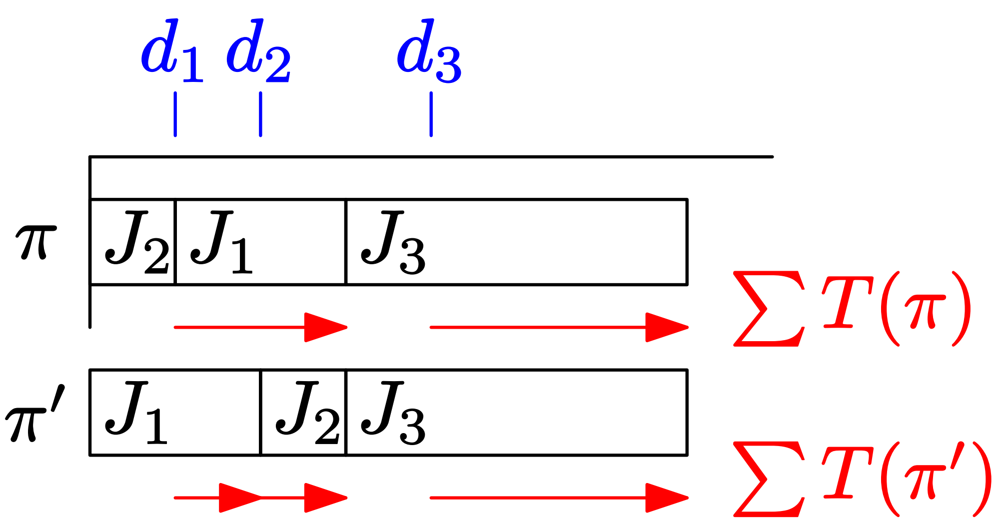
証明したい目標：$\sum T(\pi) - \sum T'(\pi) \geq \sum T(\pi') - \sum T'(\pi')$

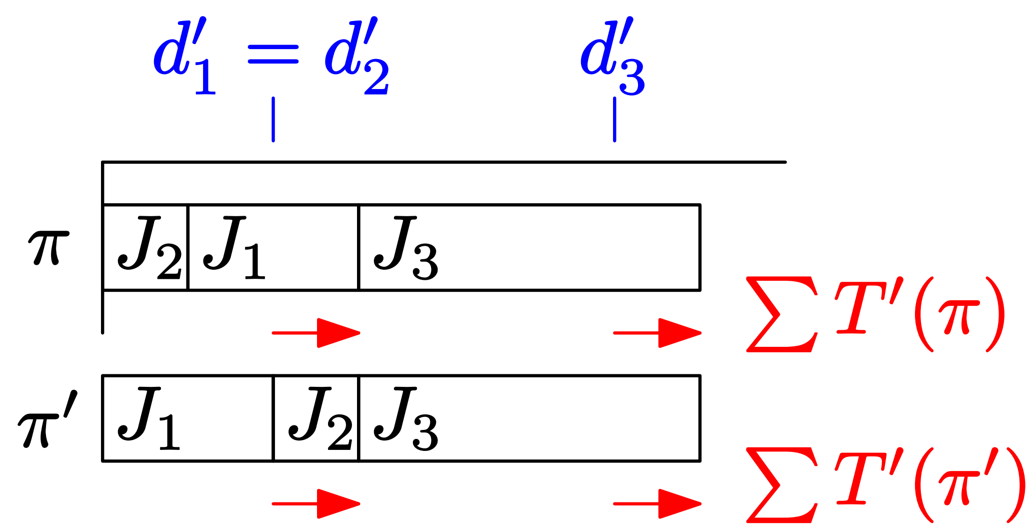
両辺に対する，各ジョブの貢献に着目する

- 仮定 $\Rightarrow \min\{d_j, C_j\} \leq d'_j \leq \max\{d_j, C_j\}$ を満たす

(Case 1) $d_j \leq C_j$ のとき：$d_j \leq d'_j \leq C_j$

- $T_j(\pi) = \max\{0, C_j - d_j\} = C_j - d_j$
- $T'_j(\pi) = \max\{0, C_j - d'_j\} = C_j - d'_j$
- $\therefore T_j(\pi) - T'_j(\pi) = d'_j - d_j$





補題 C：証明 (4)

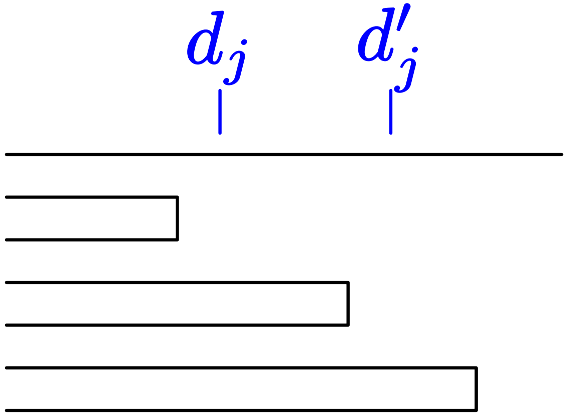
<u>証明したい目標</u>：$\sum T(\pi) - \sum T'(\pi) \geq \sum T(\pi') - \sum T'(\pi')$

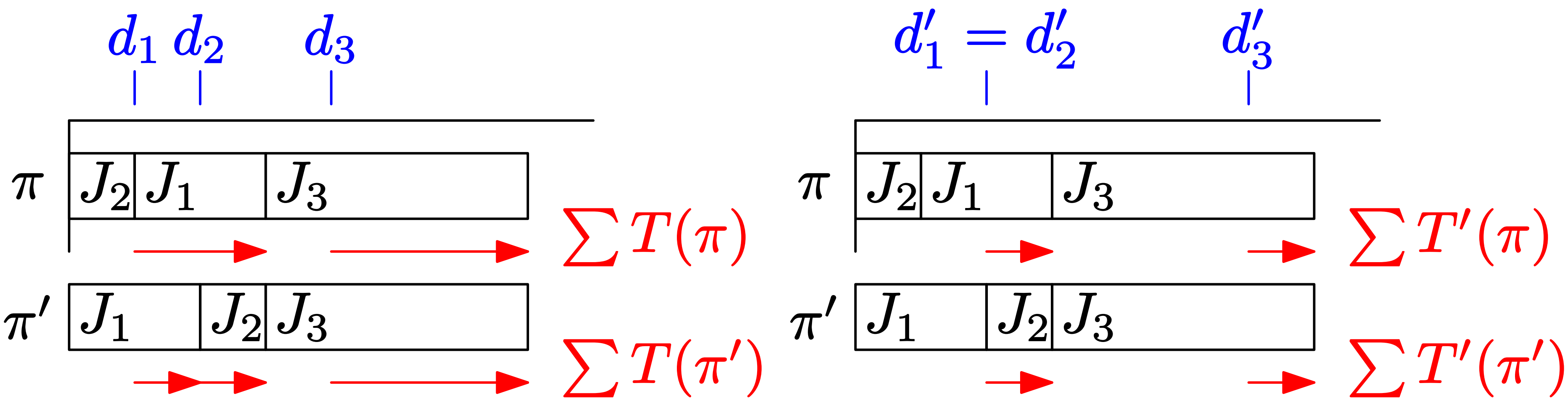
両辺に対する，各ジョブの貢献に着目する

- 仮定 $\Rightarrow \min\{d_j, C_j\} \leq d'_j \leq \max\{d_j, C_j\}$ を満たす

(Case 1) $d_j \leq C_j$ のとき：$d_j \leq d'_j \leq C_j$

- $T_j(\pi') = \max\{0, C_j(\pi') - d_j\}$
- $T'_j(\pi') = \max\{0, C_j(\pi') - d'_j\}$
- $\therefore T_j(\pi') - T'_j(\pi') \leq d'_j - d_j$





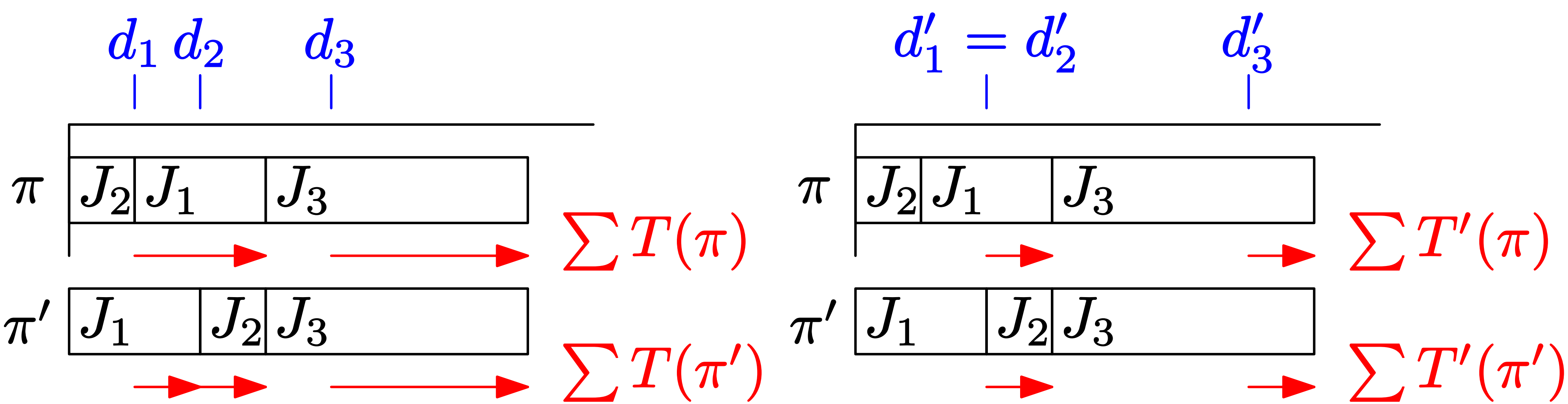
<u>証明したい目標</u>：$\sum T(\pi) - \sum T'(\pi) \geq \sum T(\pi') - \sum T'(\pi')$

両辺に対する，各ジョブの貢献に着目する

- 仮定 $\Rightarrow \min\{d_j, C_j\} \leq d'_j \leq \max\{d_j, C_j\}$ を満たす

(Case 1) $d_j \leq C_j$ のとき：$d_j \leq d'_j \leq C_j$

- したがって，$T_j(\pi) - T'_j(\pi) = d'_j - d_j \geq T_j(\pi') - T'_j(\pi')$



補題 C：証明 (6)

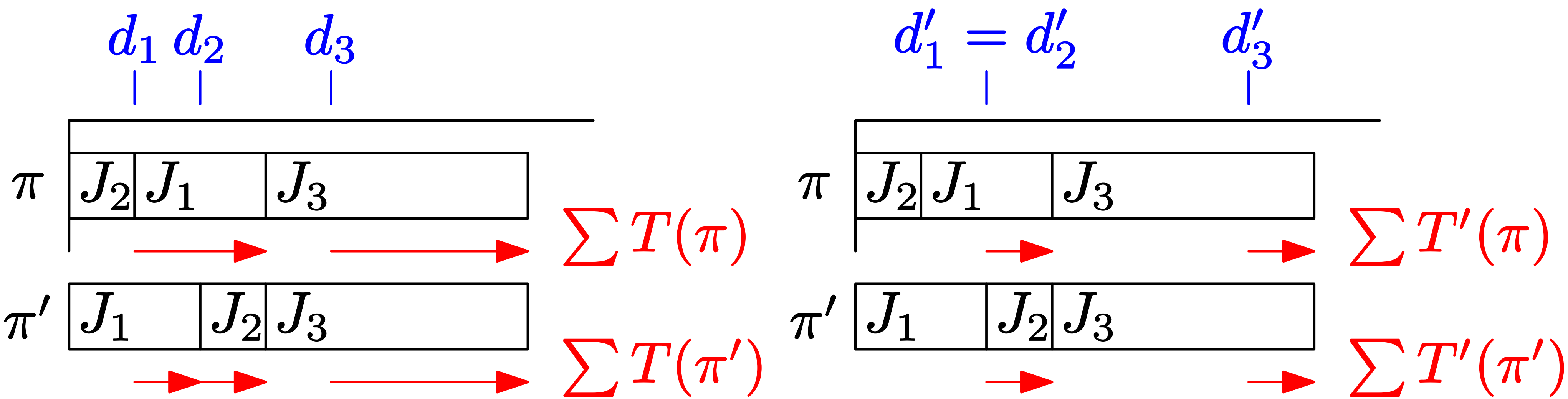
証明したい目標：$\sum T(\pi) - \sum T'(\pi) \geq \sum T(\pi') - \sum T'(\pi')$

両辺に対する，各ジョブの貢献に着目する

- 仮定 $\Rightarrow \min\{d_j, C_j\} \leq d'_j \leq \max\{d_j, C_j\}$ を満たす

(Case 2) $C_j \leq d_j$ のとき：$C_j \leq d'_j \leq d_j$

- $T_j(\pi) = \max\{0, C_j - d_j\} = 0$
- $T'_j(\pi) = \max\{0, C_j - d'_j\} = 0$
- $\therefore T_j(\pi) - T'_j(\pi) = 0$



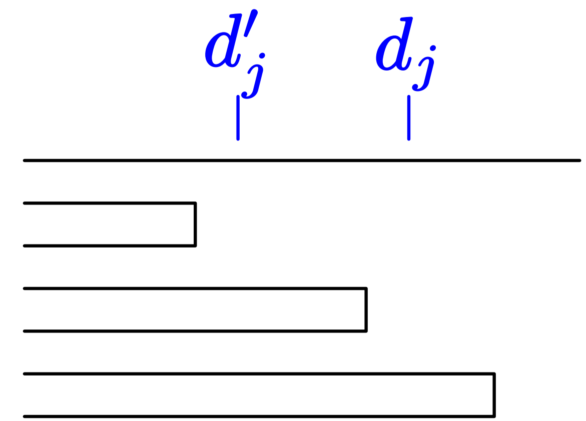
<u>証明したい目標</u>：$\sum T(\pi) - \sum T'(\pi) \geq \sum T(\pi') - \sum T'(\pi')$

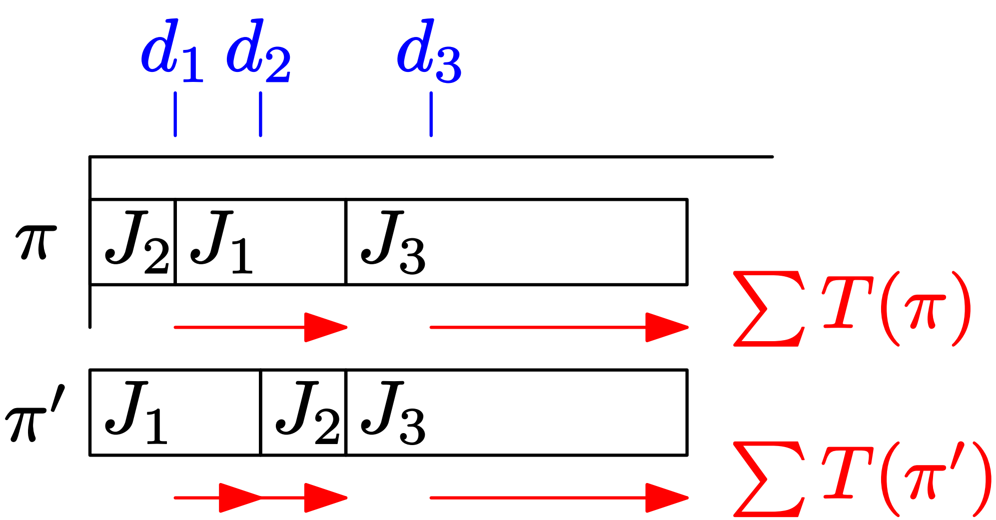
両辺に対する，各ジョブの貢献に着目する

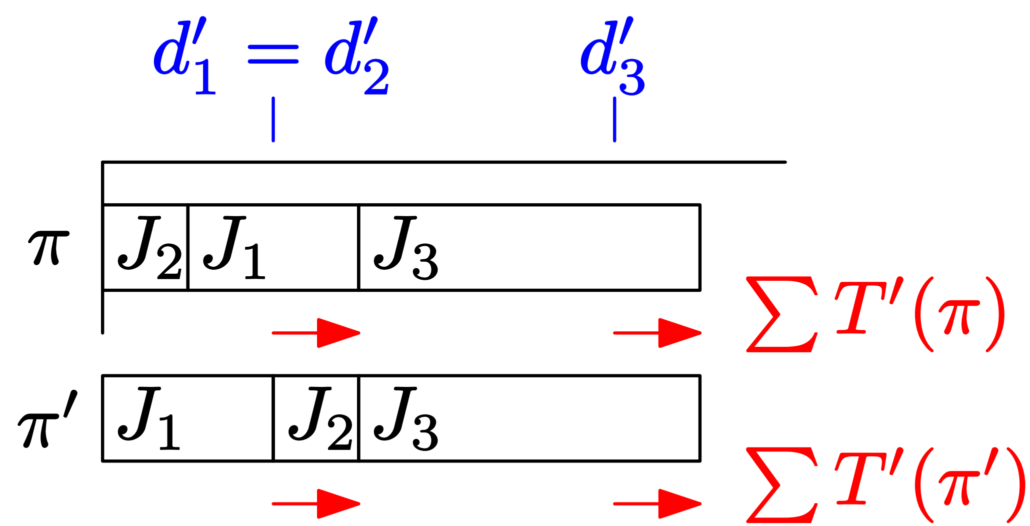
- 仮定 $\Rightarrow \min\{d_j, C_j\} \leq d'_j \leq \max\{d_j, C_j\}$ を満たす

(Case 2) $C_j \leq d_j$ のとき：$C_j \leq d'_j \leq d_j$

- $T_j(\pi') = \max\{0, C'_j - d_j\}$
- $T'_j(\pi') = \max\{0, C'_j - d'_j\}$
- $\therefore T_j(\pi') - T'_j(\pi') \leq d'_j - d_j \leq 0$





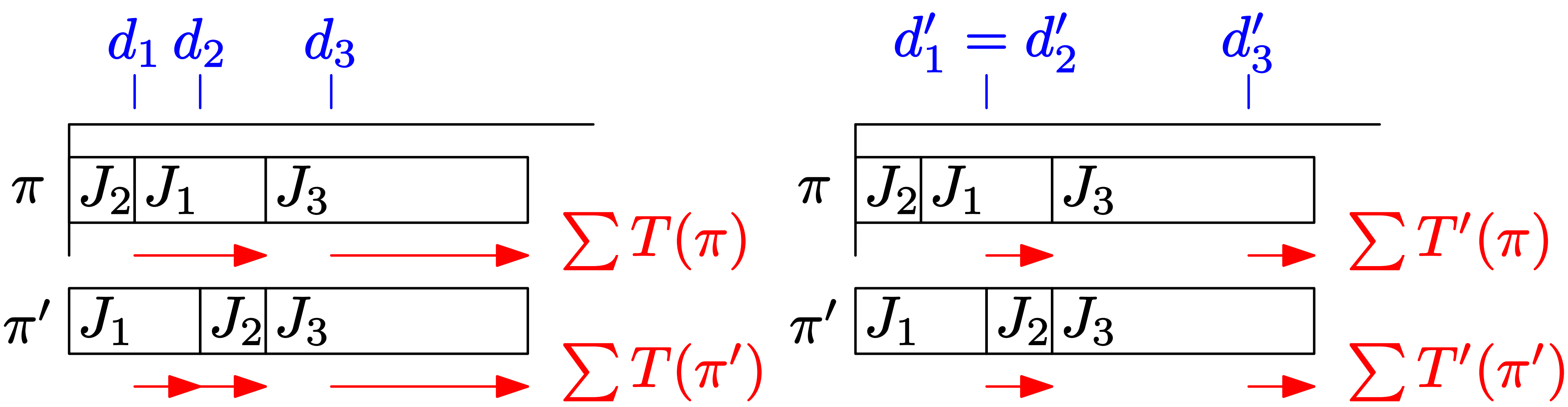


補題 C：証明 (8)

証明したい目標：$\sum T(\pi) - \sum T'(\pi) \geq \sum T(\pi') - \sum T'(\pi')$

両辺に対する，各ジョブの貢献に着目する
- 仮定 $\Rightarrow \min\{d_j, C_j\} \leq d'_j \leq \max\{d_j, C_j\}$ を満たす

(Case 2) $C_j \leq d_j$ のとき：$C_j \leq d'_j \leq d_j$
- したがって，$T_j(\pi) - T'_j(\pi) = 0 \geq T_j(\pi') - T'_j(\pi')$



証明したい目標：$\sum T(\pi) - \sum T'(\pi) \geq \sum T(\pi') - \sum T'(\pi')$

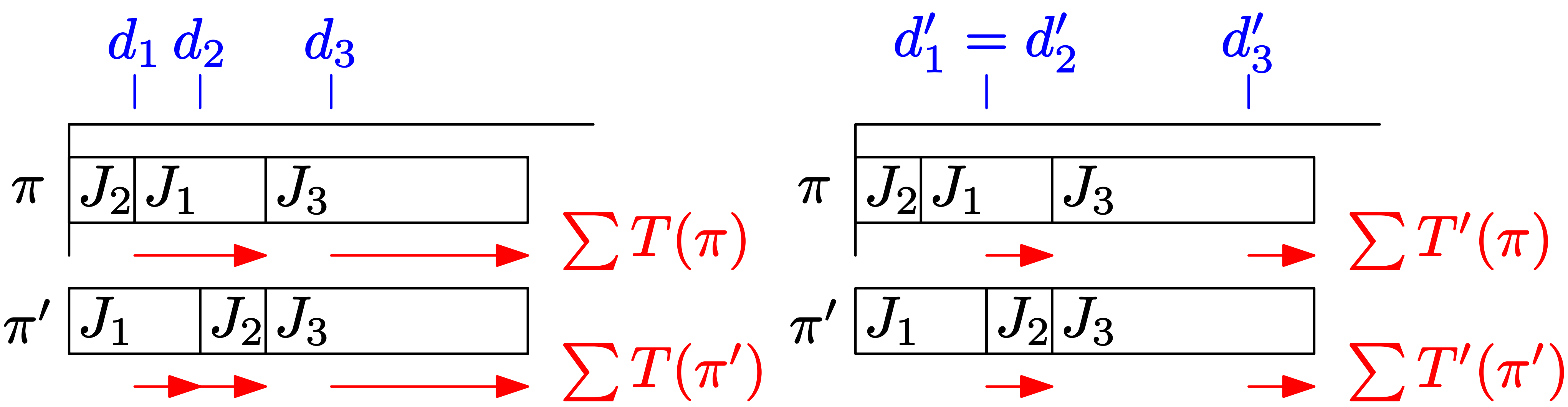
両辺に対する，各ジョブの貢献に着目する

- 仮定 $\Rightarrow \min\{d_j, C_j\} \leq d'_j \leq \max\{d_j, C_j\}$ を満たす

(Case 2) $C_j \leq d_j$ のとき：$C_j \leq d'_j \leq d_j$

- したがって，$T_j(\pi) - T'_j(\pi) = 0 \geq T_j(\pi') - T'_j(\pi')$

$\therefore$ どちらの場合でも，$T_j(\pi) - T'_j(\pi) = 0 \geq T_j(\pi') - T'_j(\pi')$



補題 C：証明 (9)

<u>証明したい目標</u>：$\sum T(\pi) - \sum T'(\pi) \geq \sum T(\pi') - \sum T'(\pi')$

まとめると，

$$\begin{aligned}
\sum T(\pi) - \sum T'(\pi) &= \sum_j T_j(\pi) - \sum_j T'_j(\pi) \\
&\geq \sum_j T_j(\pi') - \sum_j T'_j(\pi') \\
&= \sum T(\pi') - \sum T'(\pi') \qquad \square
\end{aligned}$$

付録

- 補題 A の証明
- 補題 C の証明
- **補題 B の証明**
- 動的計画法アルゴリズムの動作例

補題 B：再掲

仮定：$1 \,||\, \sum T_j$ の設定で，ジョブは EDD 順 $(d_1 \leq \cdots \leq d_n)$
J_{j^*} = 処理時間最大のジョブ

補題 B (処理時間最大ジョブ) (再掲)

ある k とある最適スケジュールが存在して，そこでは
$j \leq k, j \neq j^*$ のジョブ J_j を J_{j^*} より前に処理し
$j > k$ のジョブ J_j を J_{j^*} より後に処理する

補題 C (補題 B の証明に使う) (再掲)

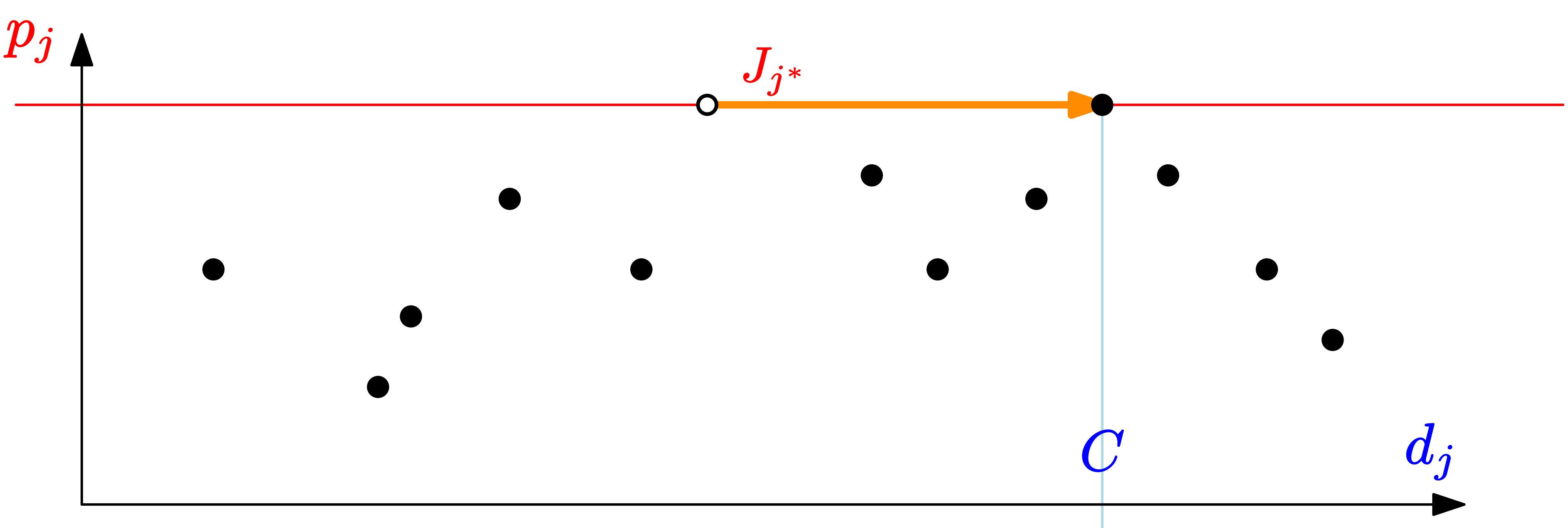
π : $d_1, \ldots, d_n$ に対する最適スケジュール
C_j : π におけるジョブ J_j の完了時刻
d'_j : $\min\{d_j, C_j\} \leq d'_j \leq \max\{d_j, C_j\}$ を満たす
π' : $d'_1, \ldots, d'_n$ に対する最適スケジュール
$\Rightarrow \pi'$ は $d_1, \ldots, d_n$ に対しても最適スケジュール

補題 B：証明 (1)

証明：$C = \max\{C_{j^*}(\pi) \mid \pi$ は最適スケジュール$\}$ とする

- 次のように $d'_1, \ldots, d'_n$ を設定

$$d'_j = \begin{cases} d_j & (j \neq j^*), \\ C & (j = j^*) \end{cases}$$

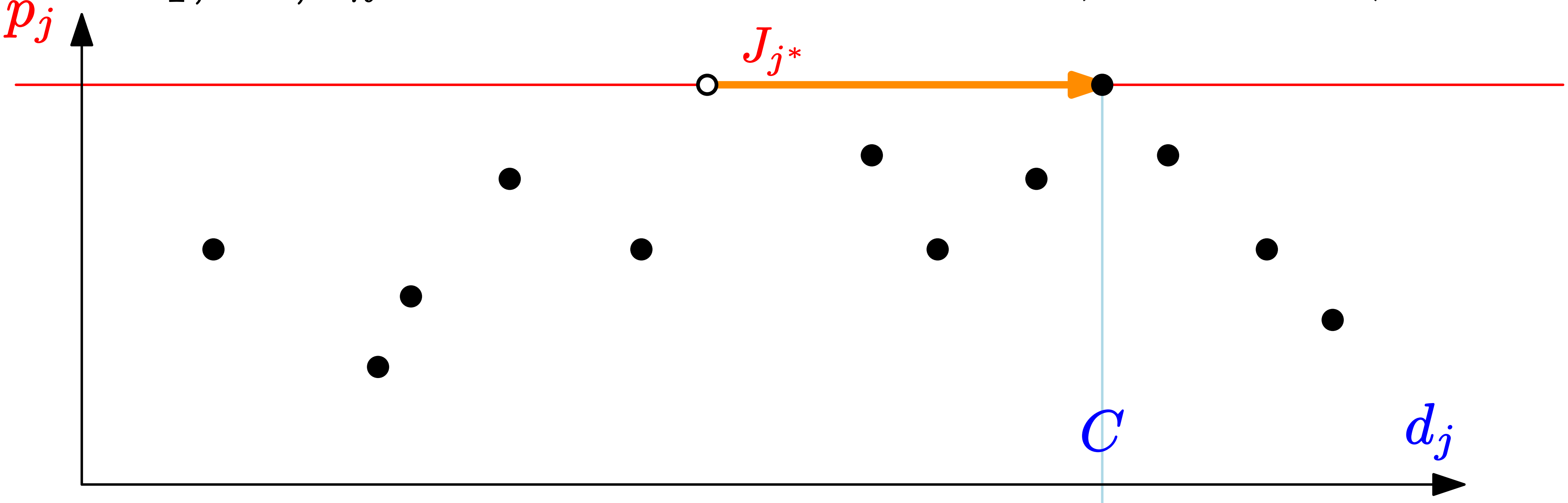


証明：$C = \max\{C_{j^*}(\pi) \mid \pi$ は最適スケジュール$\}$ とする

- 次のように $d'_1, \ldots, d'_n$ を設定

$$d'_j = \begin{cases} d_j & (j \neq j^*), \\ C & (j = j^*) \end{cases}$$

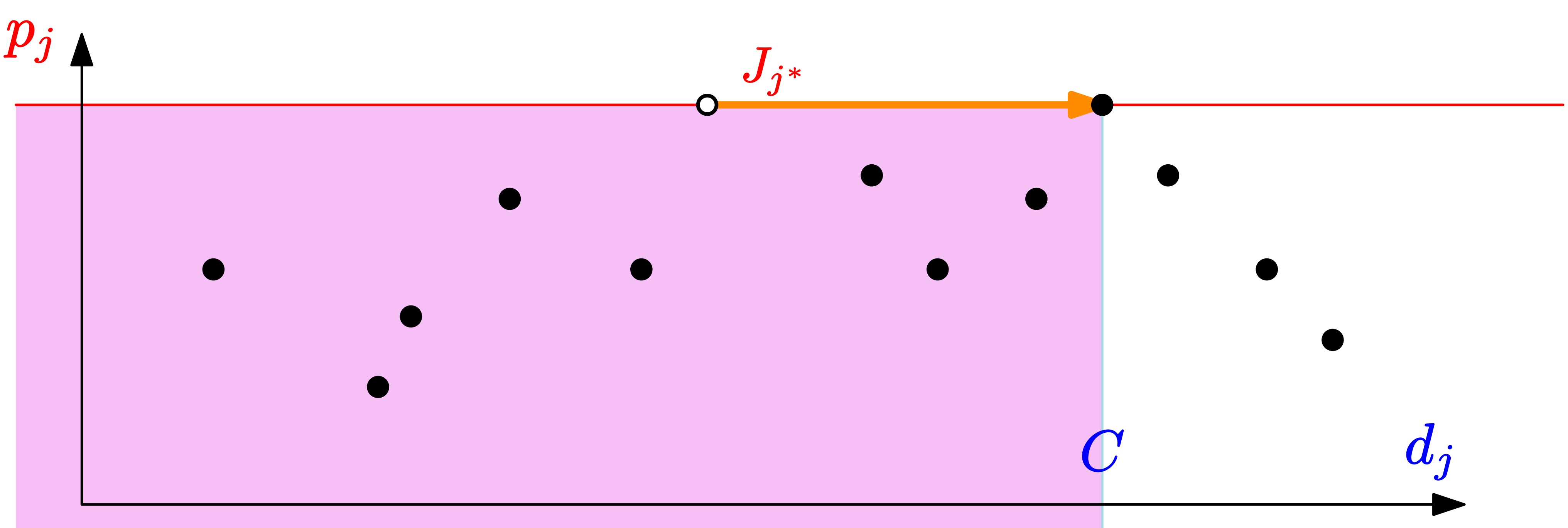
- $d'_1, \ldots, d'_n$ は補題 C の仮定を満たす
- $\therefore d'_1, \ldots, d'_n$ に対する最適スケジュールは, $d_1, \ldots, d_n$ に対する最適スケジュール (補題 C より)



補題 B：証明 (2)

証明 (続き)：

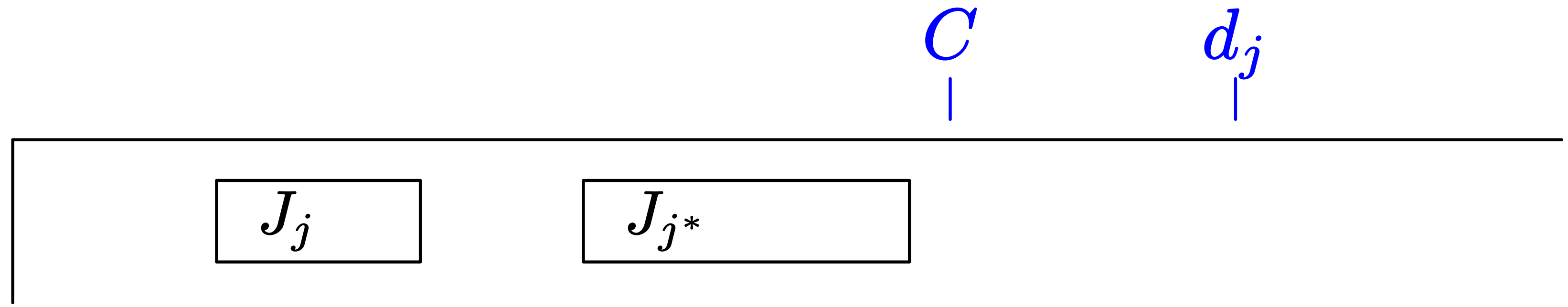
- 補題 A より，$d'_1, \ldots, d'_n$ の最適スケジュール σ で，$d_j < C$ ならば，J_j を J_{j^*} よりも前に処理するものが存在
- $C_{j^*}(\sigma) \leq C = d'_{j^*}$

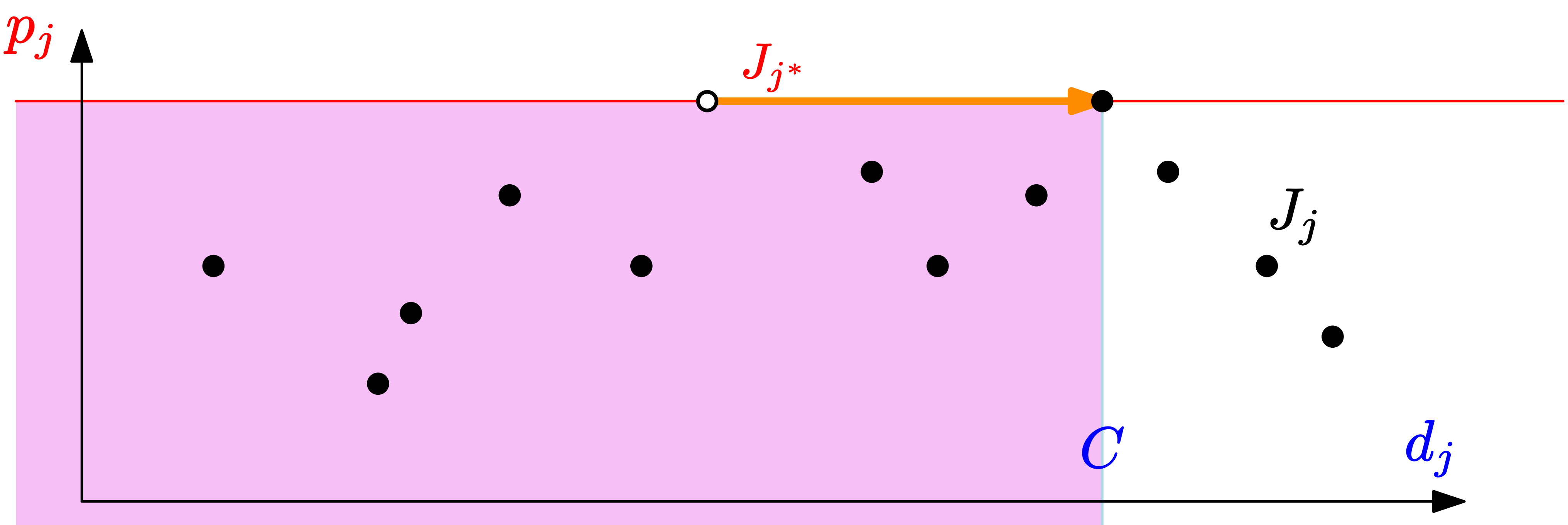


補題 B：証明 (3)

証明 (続き)：

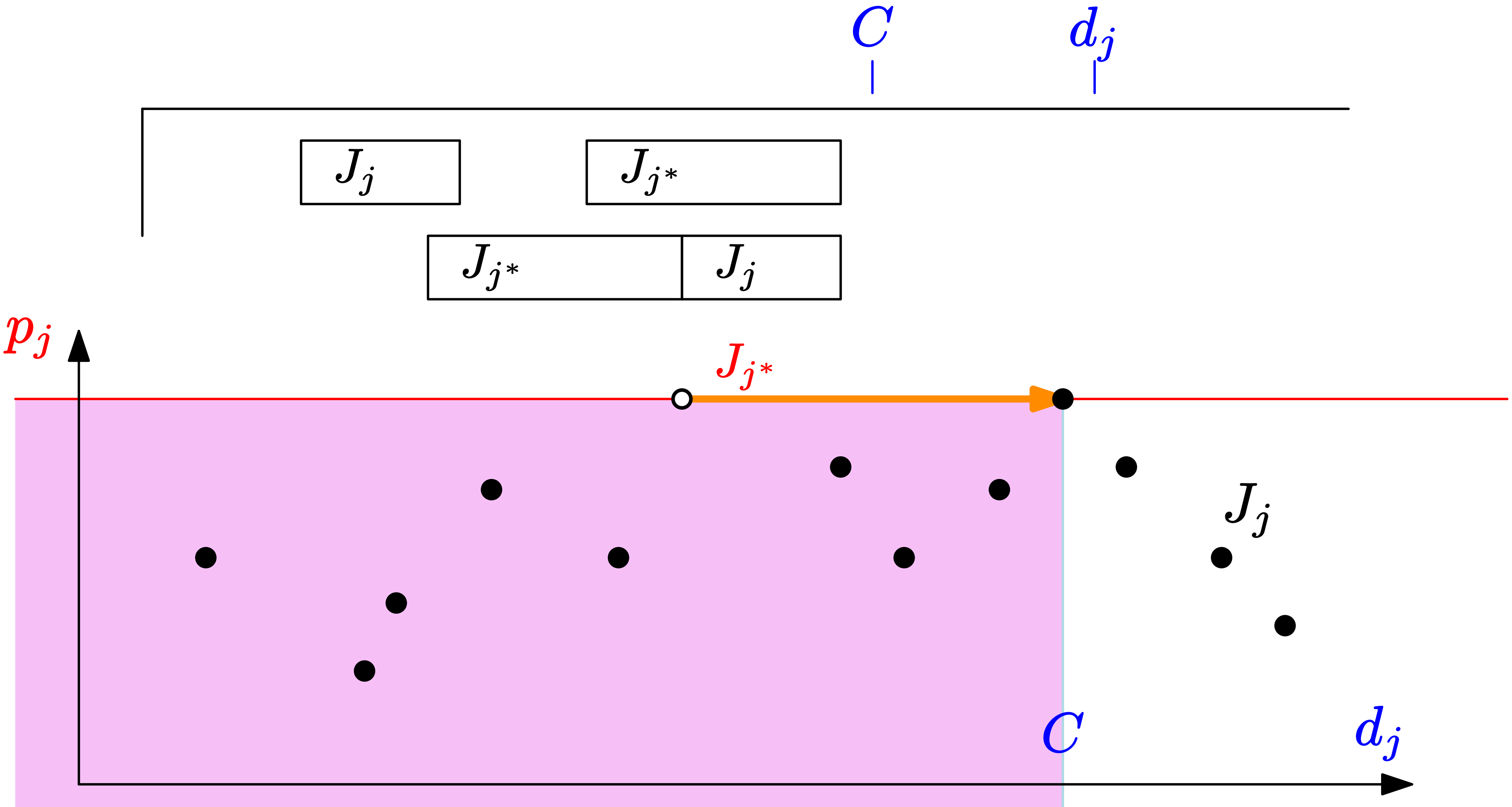
- このとき，σ において
  $C < d_j$ ならば，J_j を J_{j^*} より後に処理してもよい





<u>証明 (続き)</u>：

- このとき，σ において
  $C < d_j$ ならば，J_j を J_{j^*} より後に処理してもよい $\square$

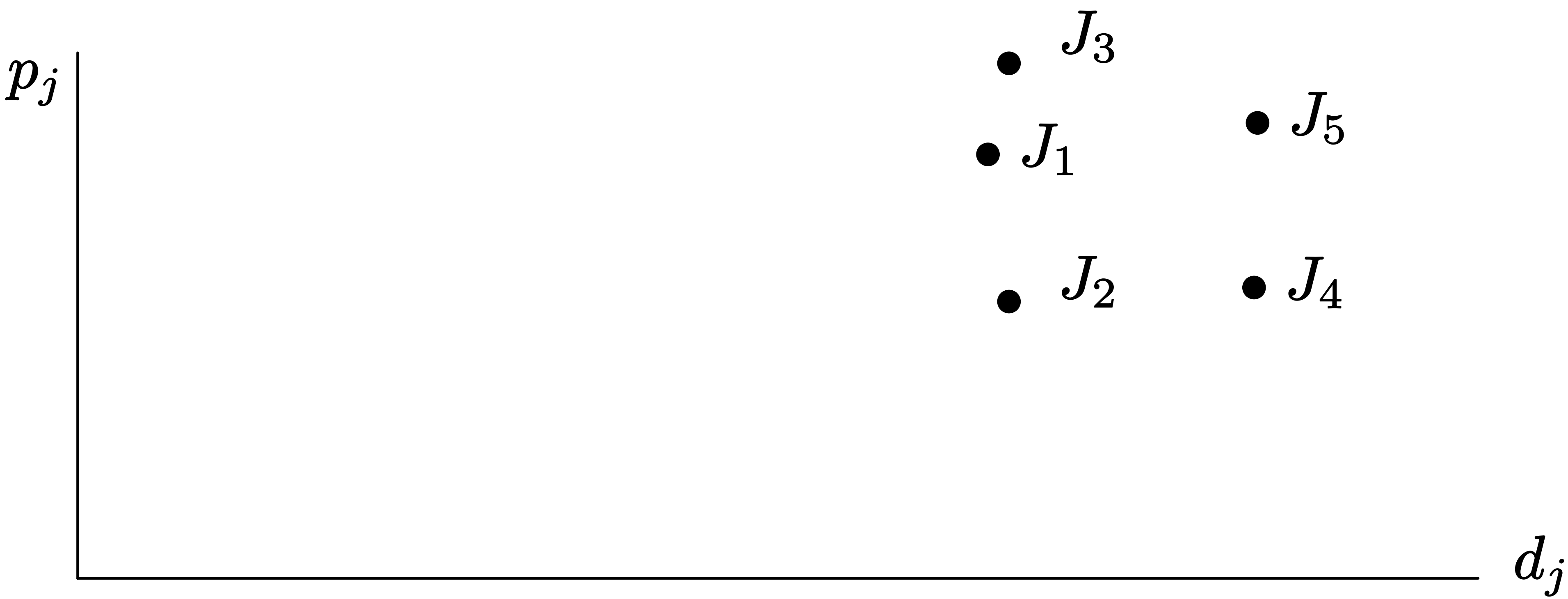


付録

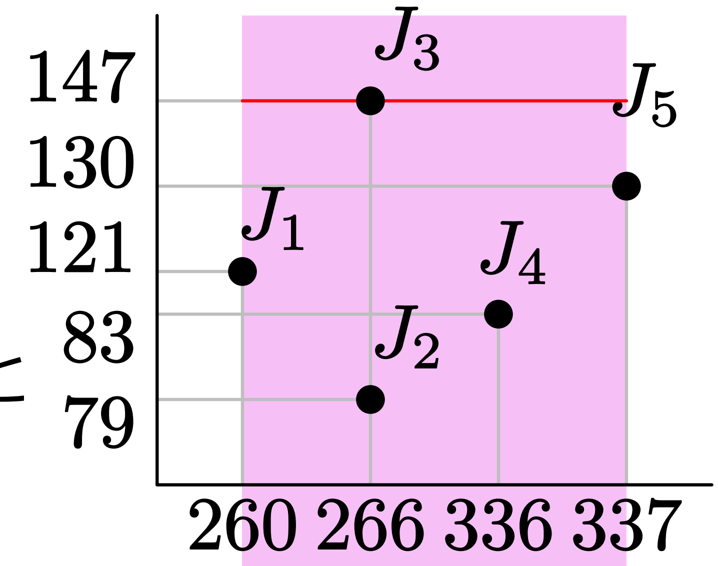
- 補題 A の証明
- 補題 C の証明
- 補題 B の証明
- **動的計画法アルゴリズムの動作例**

例 (Pinedo の本 '16 に載っている)

j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337



j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560

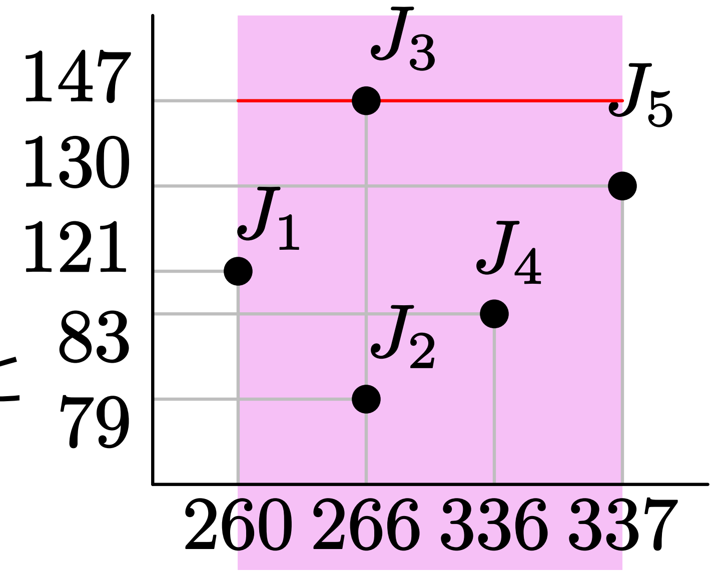


(j_1, j_2, j_3, t) が小さい方から解いていくと大変なので，普通に再帰を解いていく

$$
\begin{aligned}
f(1,6,5,0) = \min\{ & f(1,3,3,0) + \max\{0, 347-266\} + f(4,3,5,347), \\
& f(1,3,4,0) + \max\{0, 430-266\} + f(5,3,5,430), \\
& f(1,3,5,0) + \max\{0, 560-266\} + f(6,3,5,560)\}
\end{aligned}
$$

$$
f(j_1, j_2, j_3, t) = \min_{j^* \le k \le j_3} \left\{ \begin{array}{l} f(j_1, j^*, k, t) \\ + \max\{0, t + t_k^* - d_{j^*}\} \\ + f(k+1, j^*, j_3, t + t_k^*) \end{array} \right\}
$$

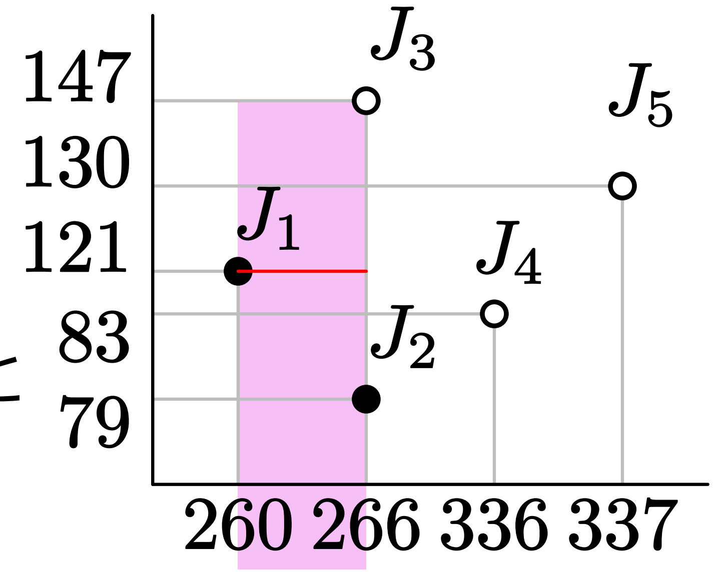
j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



(j_1, j_2, j_3, t) が小さい方から解いていくと大変なので，普通に再帰を解いていく

$$
\begin{aligned}
f(1,6,5,0) = \min\{ & f(1,3,3,0) + \max\{0, 347-266\} + f(4,3,5,347), \\
& f(1,3,4,0) + \max\{0, 430-266\} + f(5,3,5,430), \\
& f(1,3,5,0) + \max\{0, 560-266\} + f(6,3,5,560)\} \\
= \min\{ & f(1,3,3,0) + 81 + f(4,3,5,347), \\
& f(1,3,4,0) + 164 + f(5,3,5,430), \\
& f(1,3,5,0) + 294 + f(6,3,5,560)\}
\end{aligned}
$$

j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560

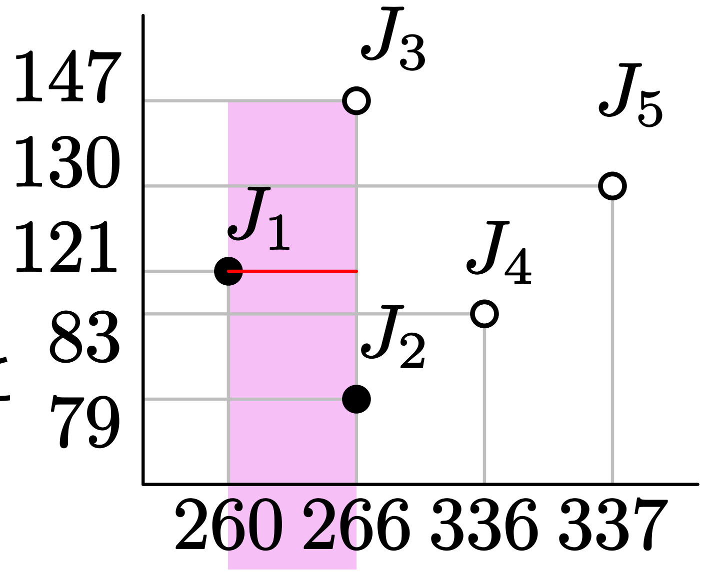


(j_1, j_2, j_3, t) が小さい方から解いていくと大変なので，普通に再帰を解いていく

$$f(1,3,3,0) = \min\{f(1,1,1,0) + \max\{0, 121-260\} + f(2,1,2,121),$$
$$f(1,1,2,0) + \max\{0, 200-266\} + f(3,1,2,200)\}$$

$$f(j_1, j_2, j_3, t) = \min_{j^* \leq k \leq j_3} \left\{ \begin{array}{l} f(j_1, j^*, k, t) \\ + \max\{0, t + t_k^* - d_{j^*}\} \\ + f(k+1, j^*, j_3, t + t_k^*) \end{array} \right\}$$

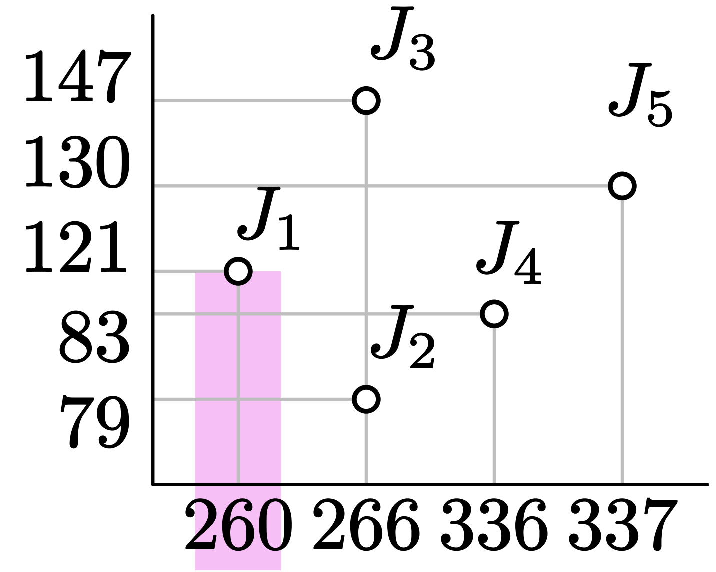
j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



(j_1, j_2, j_3, t) が小さい方から解いていくと大変なので，普通に再帰を解いていく

$$
\begin{aligned}
f(1,3,3,0) &= \min\{f(1,1,1,0) + \max\{0, 121-260\} + f(2,1,2,121), \\
&\qquad\qquad f(1,1,2,0) + \max\{0, 200-266\} + f(3,1,2,200)\} \\
&= \min\{f(1,1,1,0) + 0 + f(2,1,2,121), \\
&\qquad\qquad f(1,1,2,0) + 0 + f(3,1,2,200)\}
\end{aligned}
$$

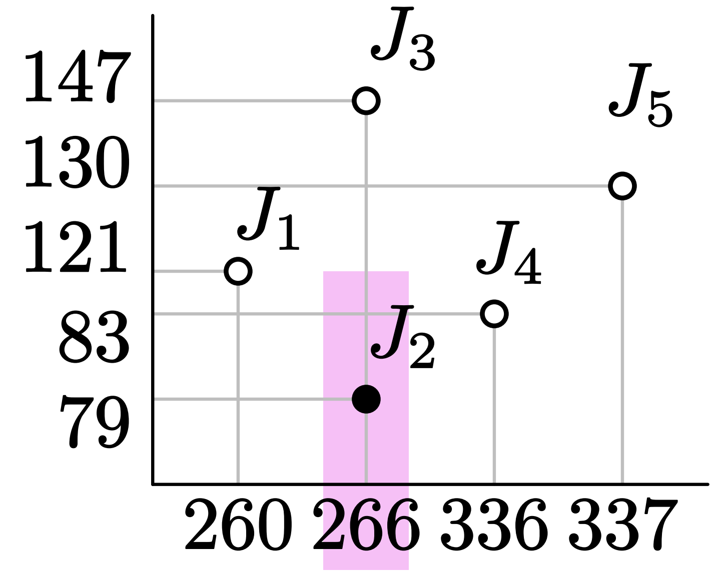
j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



$$f(1,1,1,0) \quad = 0$$

$$f(j_1, j_2, j_3, t) = \min_{j^* \le k \le j_3} \left\{ \begin{array}{l} f(j_1, j^*, k, t) \\ + \max\{0, t + t_k^* - d_{j^*}\} \\ + f(k+1, j^*, j_3, t + t_k^*) \end{array} \right\}$$

j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560

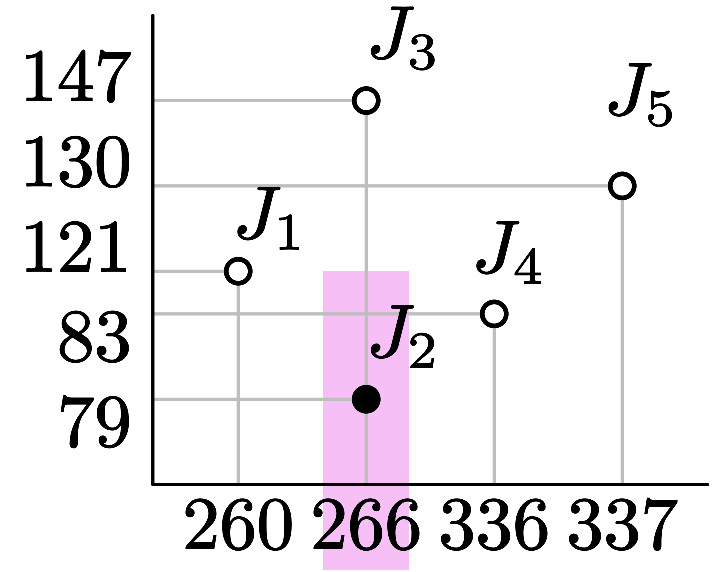


$$f(1,1,1,0) = 0$$

$$f(2,1,2,121) = \max\{0, 121 + 79 - 266\} = 0$$

$$f(j_1, j_2, j_3, t) = \min_{j^* \leq k \leq j_3} \left\{ \begin{array}{l} f(j_1, j^*, k, t) \\ + \max\{0, t + t_k^* - d_{j^*}\} \\ + f(k+1, j^*, j_3, t + t_k^*) \end{array} \right\}$$

j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



$$f(1,1,1,0) = 0$$

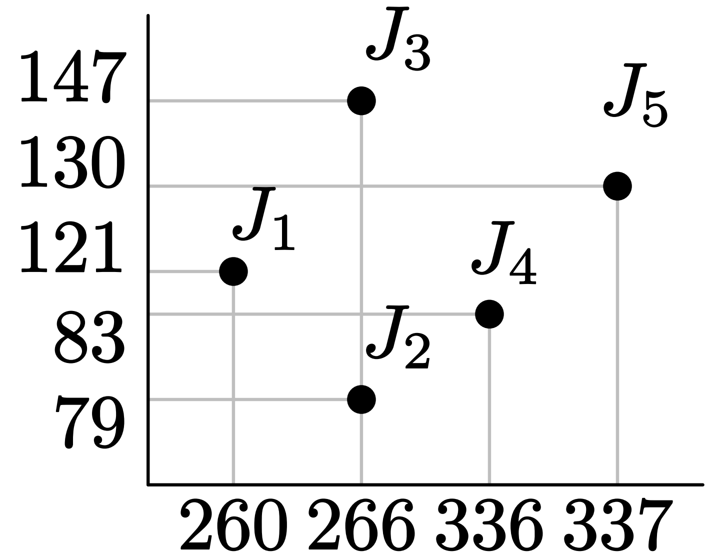
$$f(2,1,2,121) = \max\{0, 121+79-266\} = 0$$

ゆえに,

$$\begin{aligned} f(1,3,3,0) = \min\{ & f(1,1,1,0)+0+f(2,1,2,121), \\ & f(1,1,2,0)+0+f(3,1,2,200)\} \\ = 0 & \end{aligned}$$

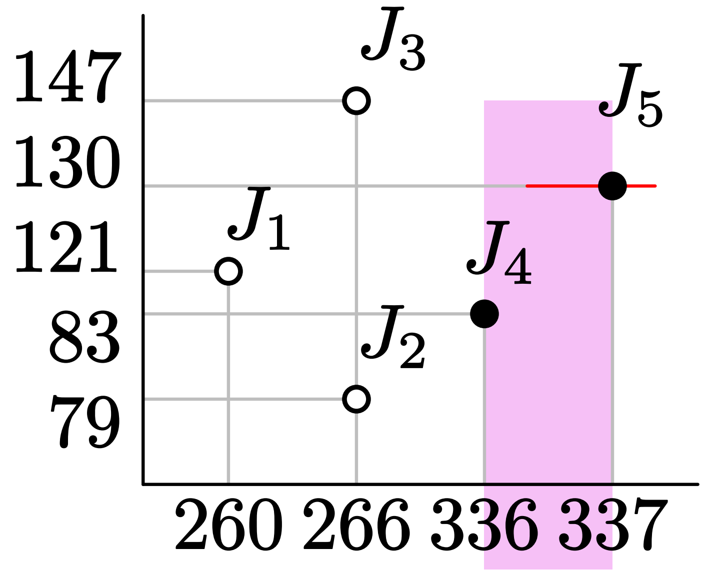
$$f(j_1, j_2, j_3, t) = \min_{j^* \le k \le j_3} \left\{ \begin{array}{l} f(j_1, j^*, k, t) \\ + \max\{0, t + t_k^* - d_{j^*}\} \\ + f(k+1, j^*, j_3, t + t_k^*) \end{array} \right\}$$

j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



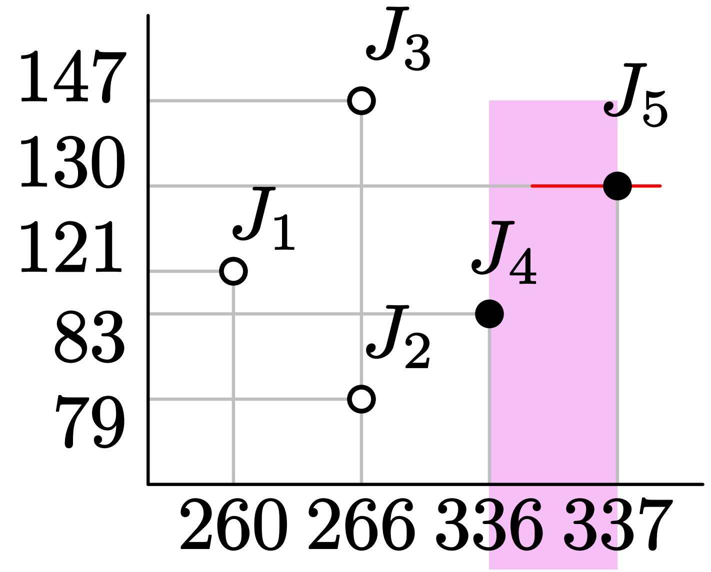
$$
\begin{aligned}
f(1,6,5,0) = \min\{&f(1,3,3,0) + 81 + f(4,3,5,347),\\
&f(1,3,4,0) + 164 + f(5,3,5,430),\\
&f(1,3,5,0) + 294 + f(6,3,5,560)\}\\
= \min\{&0 + 81 + f(4,3,5,347),\\
&f(1,3,4,0) + 164 + f(5,3,5,430),\\
&f(1,3,5,0) + 294 + f(6,3,5,560)\}
\end{aligned}
$$

j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



$$
\begin{aligned}
f(4,3,5,347) &= f(4,5,5,347) + \max\{0, 347 + 213 - 337\} + f(6,5,5,560) \\
&= f(4,5,5,347) + 223 + f(6,5,5,560)
\end{aligned}
$$

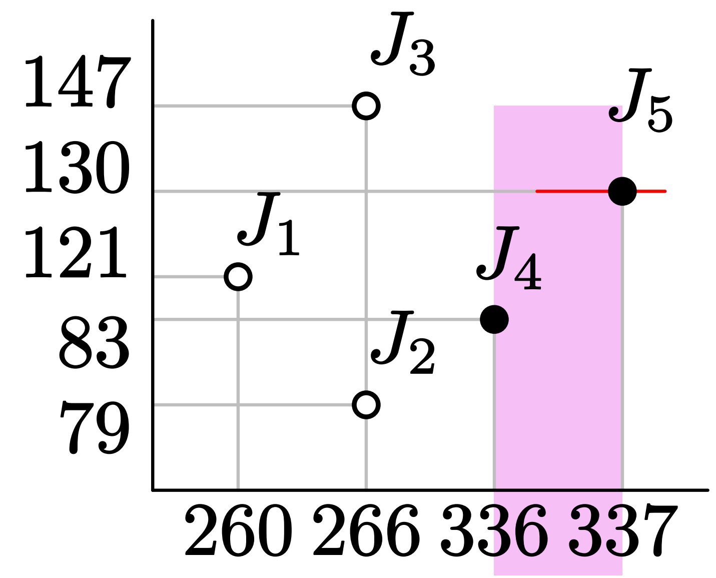
j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



$$\begin{aligned} f(4,3,5,347) &= f(4,5,5,347) + \max\{0, 347+213-337\} + f(6,5,5,560) \\ &= f(4,5,5,347) + 223 + f(6,5,5,560) \end{aligned}$$

$$f(4,5,5,347) = 347 + 83 - 336 = 94$$

j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560

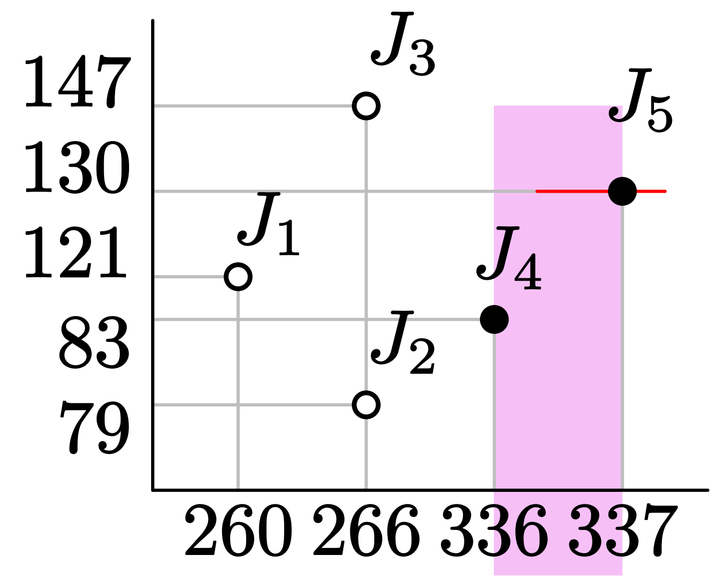


$$
\begin{aligned}
f(4,3,5,347) &= f(4,5,5,347) + \max\{0, 347+213-337\} + f(6,5,5,560) \\
&= f(4,5,5,347) + 223 + f(6,5,5,560)
\end{aligned}
$$

$$f(4,5,5,347) = 347 + 83 - 336 = 94$$

$$f(6,5,5,560) = 0$$

j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560

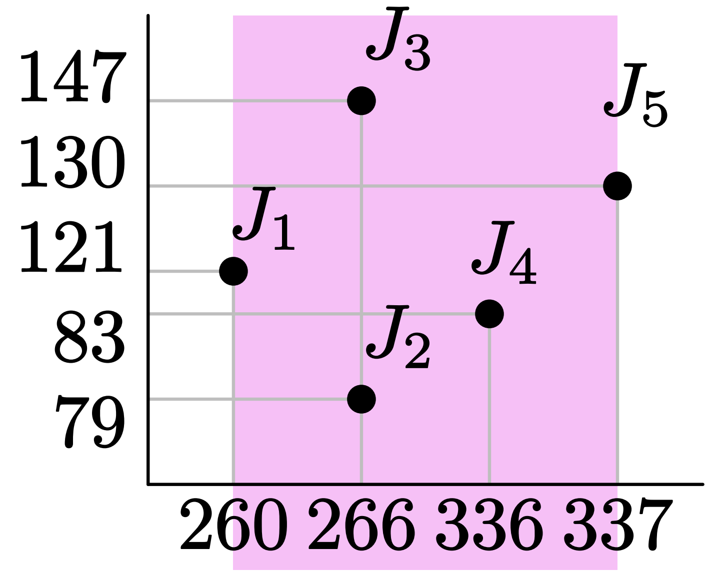


$$
\begin{aligned}
f(4,3,5,347) &= f(4,5,5,347) + \max\{0, 347+213-337\} + f(6,5,5,560) \\
&= f(4,5,5,347) + 223 + f(6,5,5,560) \\
&= 94 + 223 + 0 = 317
\end{aligned}
$$

$$f(4,5,5,347) = 347 + 83 - 336 = 94$$

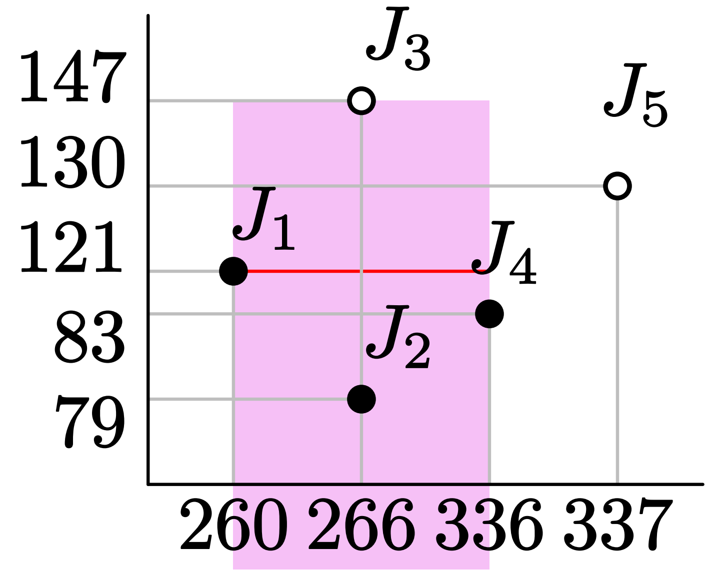
$$f(6,5,5,560) = 0$$

j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



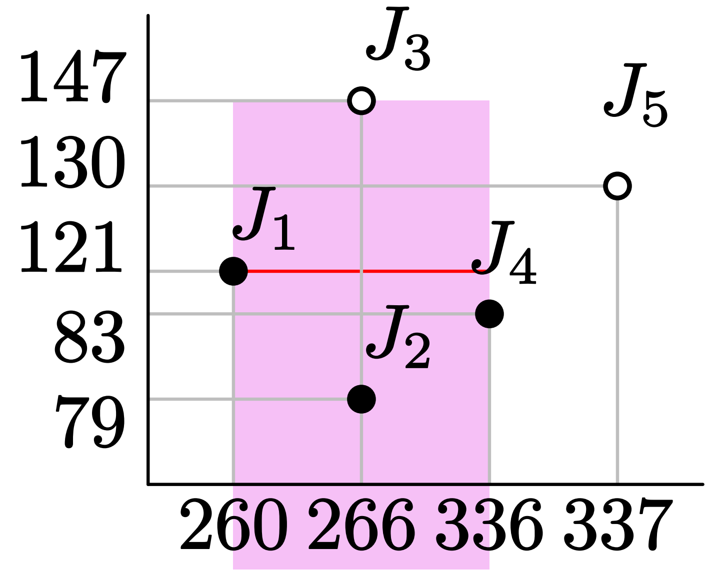
$$
\begin{aligned}
f(1,6,5,0) = \min\{ & f(1,3,3,0) + 81 + f(4,3,5,347), \\
& f(1,3,4,0) + 164 + f(5,3,5,430), \\
& f(1,3,5,0) + 294 + f(6,3,5,560)\} \\
= \min\{ & 0 + 81 + 317, \\
& f(1,3,4,0) + 164 + f(5,3,5,430), \\
& f(1,3,5,0) + 294 + f(6,3,5,560)\}
\end{aligned}
$$

j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



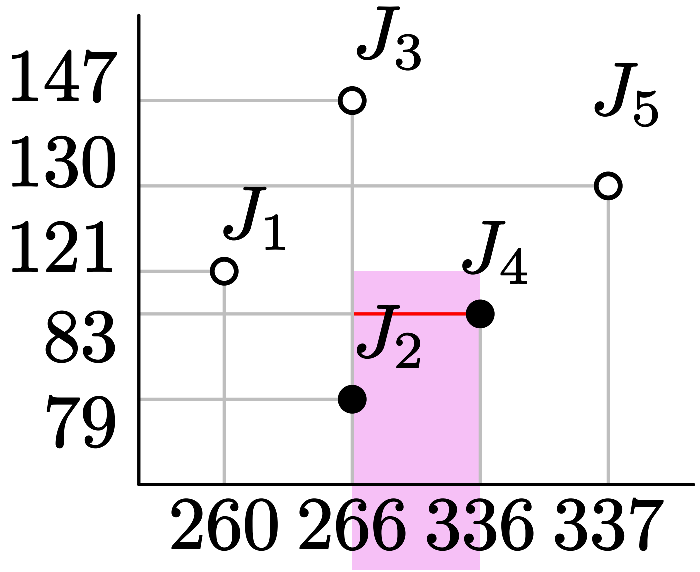
$$f(1,3,4,0) = \min\{f(1,1,1,0) + \max\{0, 121-260\} + f(2,1,4,121),$$
$$f(1,1,2,0) + \max\{0, 200-121\} + f(3,1,4,200),$$
$$f(1,1,3,0) + \max\{0, 200-121\} + f(4,1,4,200),$$
$$f(1,1,4,0) + \max\{0, 283-121\} + f(5,1,4,283)\}$$

j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



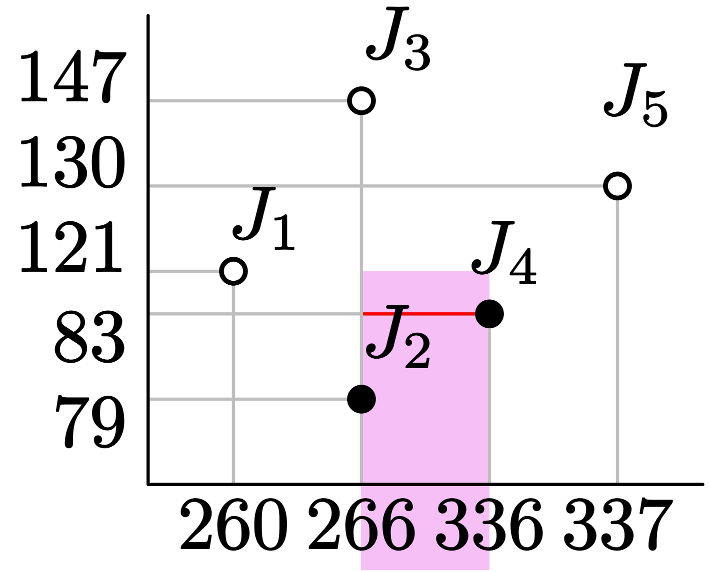
$$
\begin{aligned}
f(1,3,4,0) = \min\{ & f(1,1,1,0) + \max\{0, 121-260\} + f(2,1,4,121), \\
& f(1,1,2,0) + \max\{0, 200-121\} + f(3,1,4,200), \\
& f(1,1,3,0) + \max\{0, 200-121\} + f(4,1,4,200), \\
& f(1,1,4,0) + \max\{0, 283-121\} + f(5,1,4,283)\} \\
= \min\{ & 0 + 0 + f(2,1,4,121), \\
& f(1,1,2,0) + 79 + f(3,1,4,200), \\
& f(1,1,4,0) + 162 + f(5,1,4,283)\}
\end{aligned}
$$

j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



$$
\begin{aligned}
f(2,1,4,121) &= f(2,4,4,121) + \max\{0, 121 + 162 - 336\} + f(5,4,4,284) \\
&= f(2,4,4,121) + 0 + f(5,4,4,284)
\end{aligned}
$$

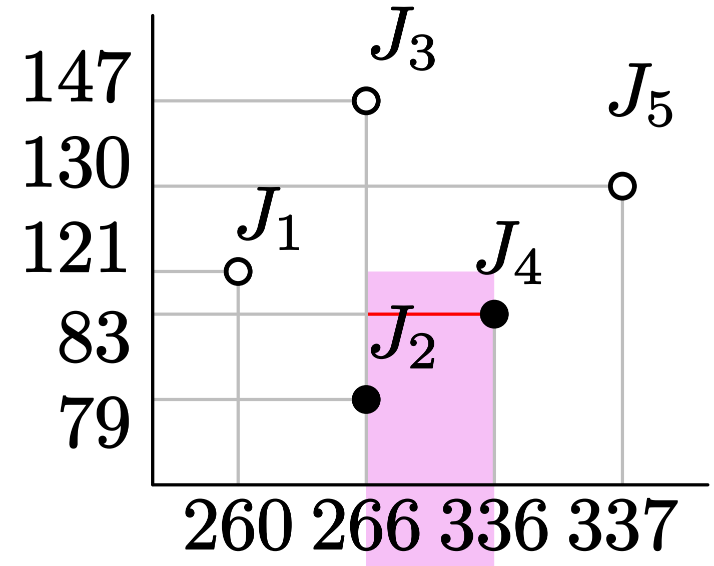
j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



$$
\begin{aligned}
f(2,1,4,121) &= f(2,4,4,121) + \max\{0, 121 + 162 - 336\} + f(5,4,4,284) \\
&= f(2,4,4,121) + 0 + f(5,4,4,284)
\end{aligned}
$$

$$
f(2,4,4,121) = \max\{0, 121 + 79 - 266\} = 0
$$

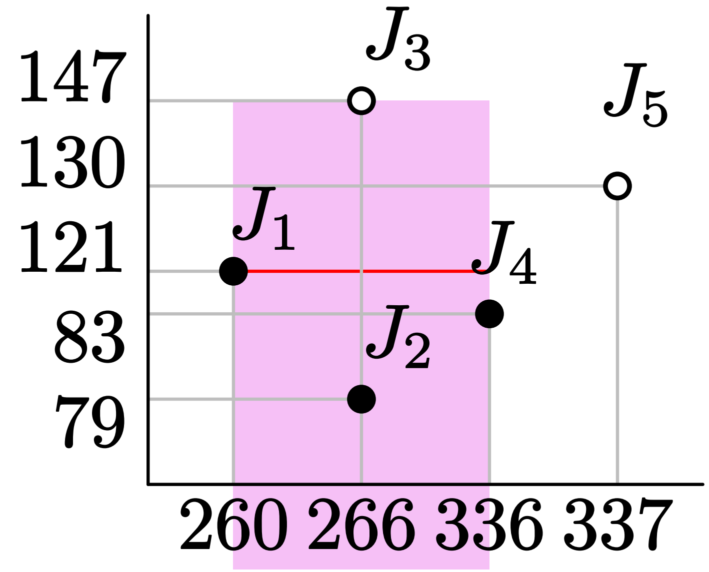
j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



$$
\begin{aligned}
f(2,1,4,121) &= f(2,4,4,121) + \max\{0, 121+162-336\} + f(5,4,4,284) \\
&= f(2,4,4,121) + 0 + f(5,4,4,284) \\
&= 0 + 0 + 0 = 0
\end{aligned}
$$

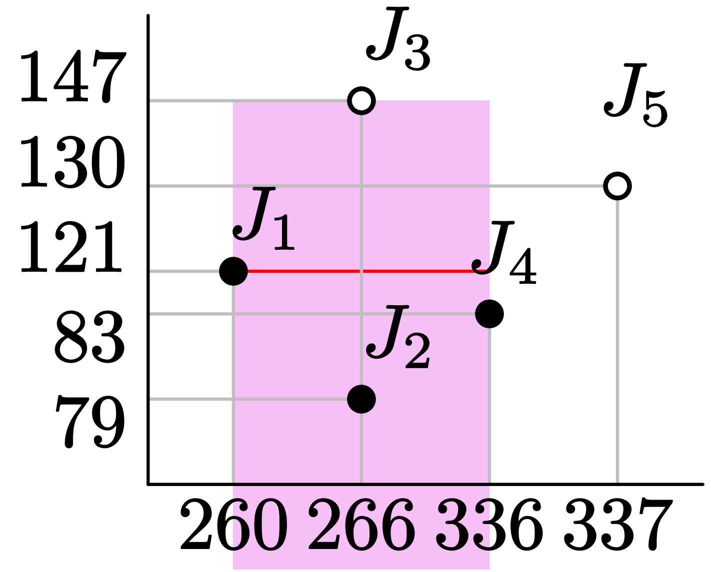
$$f(2,4,4,121) = \max\{0, 121+79-266\} = 0$$

j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



$$
\begin{aligned}
f(1,3,4,0) = \min\{&0 + 0 + f(2,1,4,121),\\
&f(1,1,2,0) + 79 + f(3,1,4,200),\\
&f(1,1,4,0) + 162 + f(5,1,4,283)\}\\
= 0&
\end{aligned}
$$

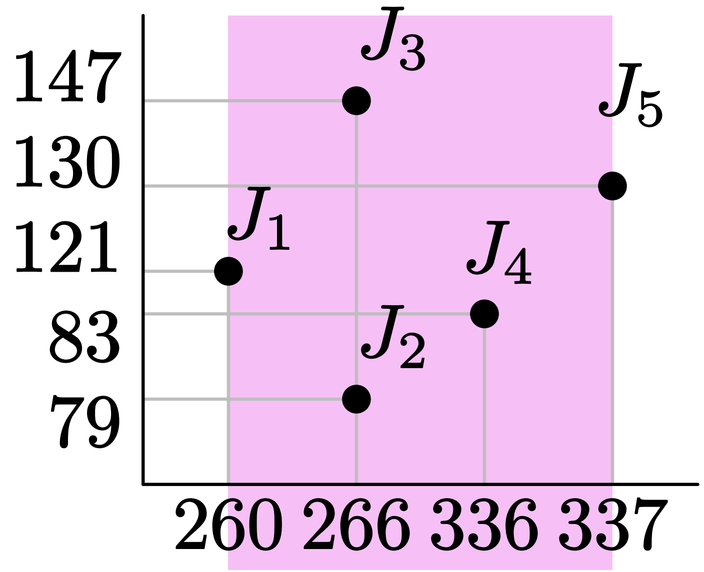
j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



$$
\begin{aligned}
f(1,3,4,0) = \min\{&0+0+f(2,1,4,121),\\
&f(1,1,2,0)+79+f(3,1,4,200),\\
&f(1,1,4,0)+162+f(5,1,4,283)\}\\
= 0&
\end{aligned}
$$

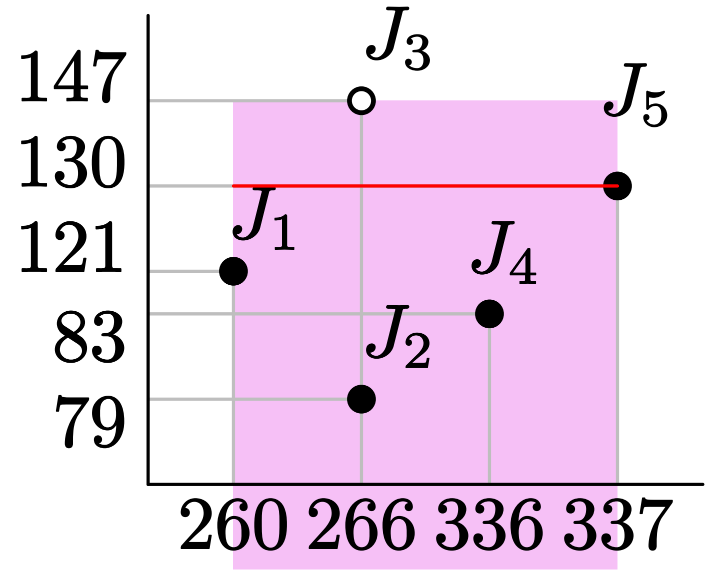
$$f(5,3,5,430) = \max\{0, 430+130-337\} = 223$$

j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



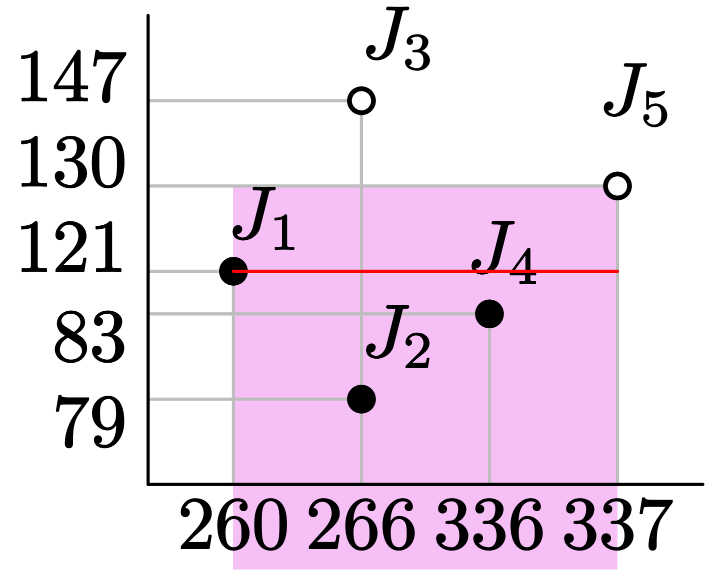
$$
\begin{aligned}
f(1,6,5,0) &= \min\{f(1,3,3,0)+81+f(4,3,5,347),\\
&\qquad\quad f(1,3,4,0)+164+f(5,3,5,430),\\
&\qquad\quad f(1,3,5,0)+294+f(6,3,5,560)\}\\
&= \min\{0+81+317,\\
&\qquad\quad 0+164+223,\\
&\qquad\quad f(1,3,5,0)+294+f(6,3,5,560)\}
\end{aligned}
$$

j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



$$
\begin{aligned}
f(1,3,5,0) &= f(1,5,5,0) + \max\{0, 413 - 337\} + f(6,5,5,560) \\
&= f(1,5,5,0) + 76 + 0
\end{aligned}
$$

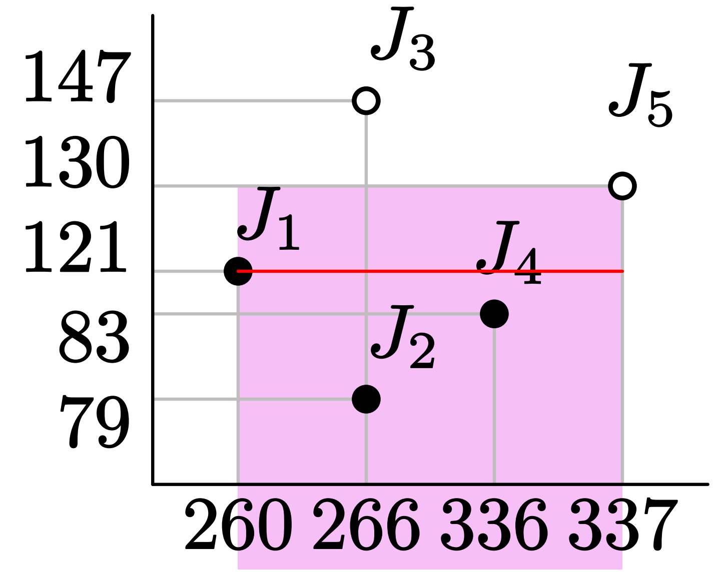
j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



$$
\begin{aligned}
f(1,3,5,0) &= f(1,5,5,0) + \max\{0, 413 - 337\} + f(6,5,5,560) \\
&= f(1,5,5,0) + 76 + 0
\end{aligned}
$$

$$f(1,5,5,0) = f(1,3,4,0) = 0$$

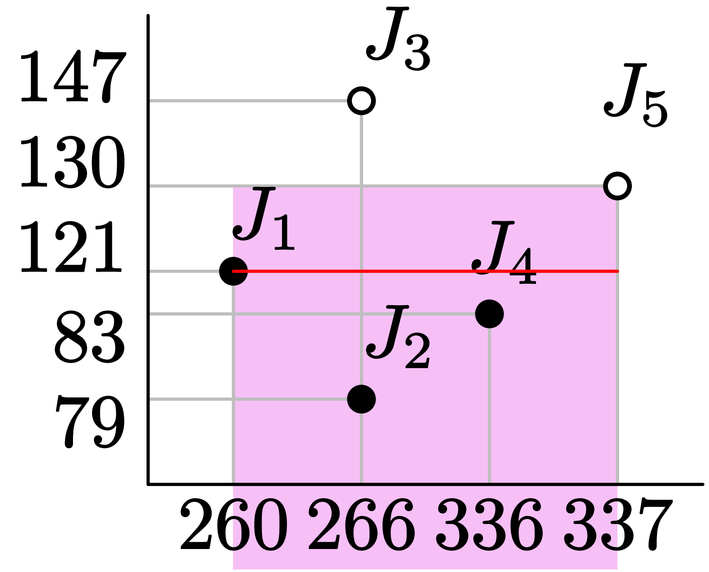
j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



$$f(1,3,5,0) = f(1,5,5,0) + \max\{0, 413 - 337\} + f(6,5,5,560)$$
$$= f(1,5,5,0) + 76 + 0$$
$$= 0 + 76 + 0 = 76$$

$$f(1,5,5,0) = f(1,3,4,0) = 0$$

j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560

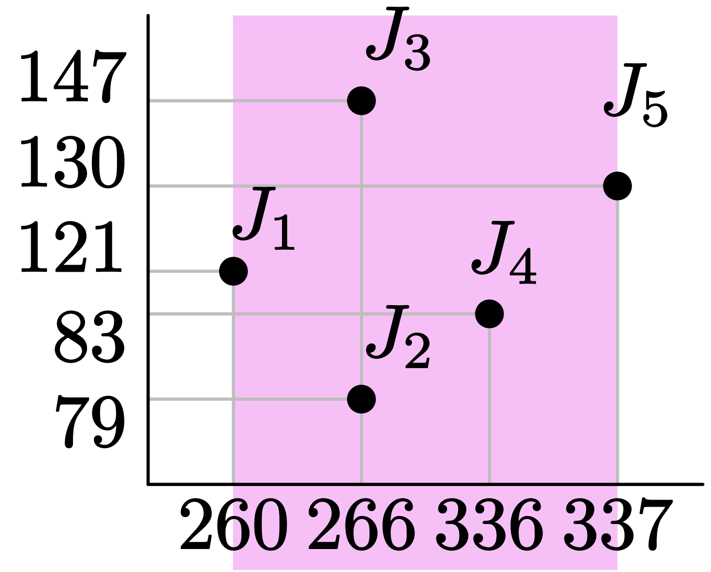


$$
\begin{aligned}
f(1,3,5,0) &= f(1,5,5,0) + \max\{0, 413-337\} + f(6,5,5,560) \\
&= f(1,5,5,0) + 76 + 0 \\
&= 0 + 76 + 0 = 76
\end{aligned}
$$

$$f(1,5,5,0) = f(1,3,4,0) = 0$$

$$f(6,3,5,560) = 0$$

j	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337
	121	200	347	430	560



$$
\begin{aligned}
f(1,6,5,0) = \min\{ & f(1,3,3,0) + 81 + f(4,3,5,347),\\
& f(1,3,4,0) + 164 + f(5,3,5,430),\\
& f(1,3,5,0) + 294 + f(6,3,5,560)\}\\
= \min\{ & 0 + 81 + 317,\\
& 0 + 164 + 223,\\
& 76 + 294 + 0\}\\
= \min\{ & 398, 387, 370\} = 370
\end{aligned}
$$