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 4 回

NP 困難性と計算量の分類

岡本 吉央　(電気通信大学)

okamotoy@uec.ac.jp

2024 年 11 月 5 日

最終更新：2024 年 11 月 5 日　13:01

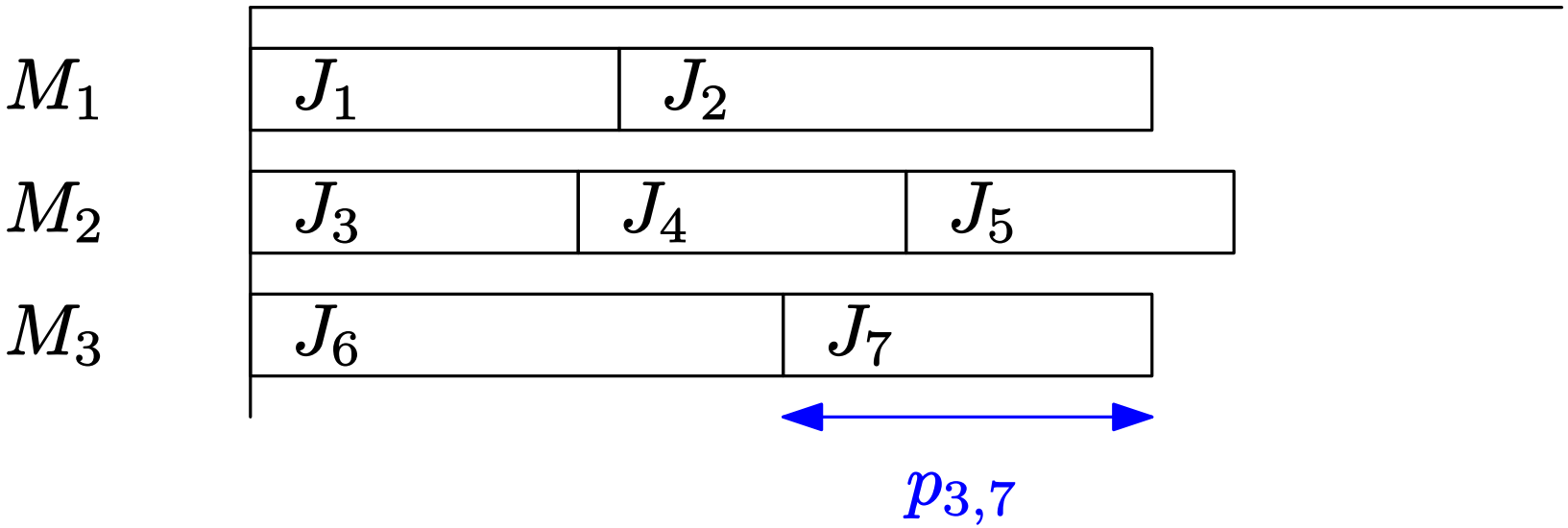
スケジュール (前半)

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・目標

アルゴリズムの計算量に関する基礎概念を身に付ける

- 問題，インスタンス
- 多項式時間アルゴリズム，擬多項式時間アルゴリズム
- 弱 NP 困難性，強 NP 困難性

NP 困難性の証明をできるようになる

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

本日の内容

1. **アルゴリズムと計算量の基本的な考え方**
2. NP 困難性の基本的な考え方
3. ジョブ・スケジューリング問題の NP 困難性証明の例

- 岡本吉央，アルゴリズムを勉強する前に読む 12 章，数学セミナー 2023 年 4 月号～2024 年 3 月号.

アルゴリズム

アルゴリズムには入力と出力がある

アルゴリズムが入力を処理した結果が出力である

アルゴリズム

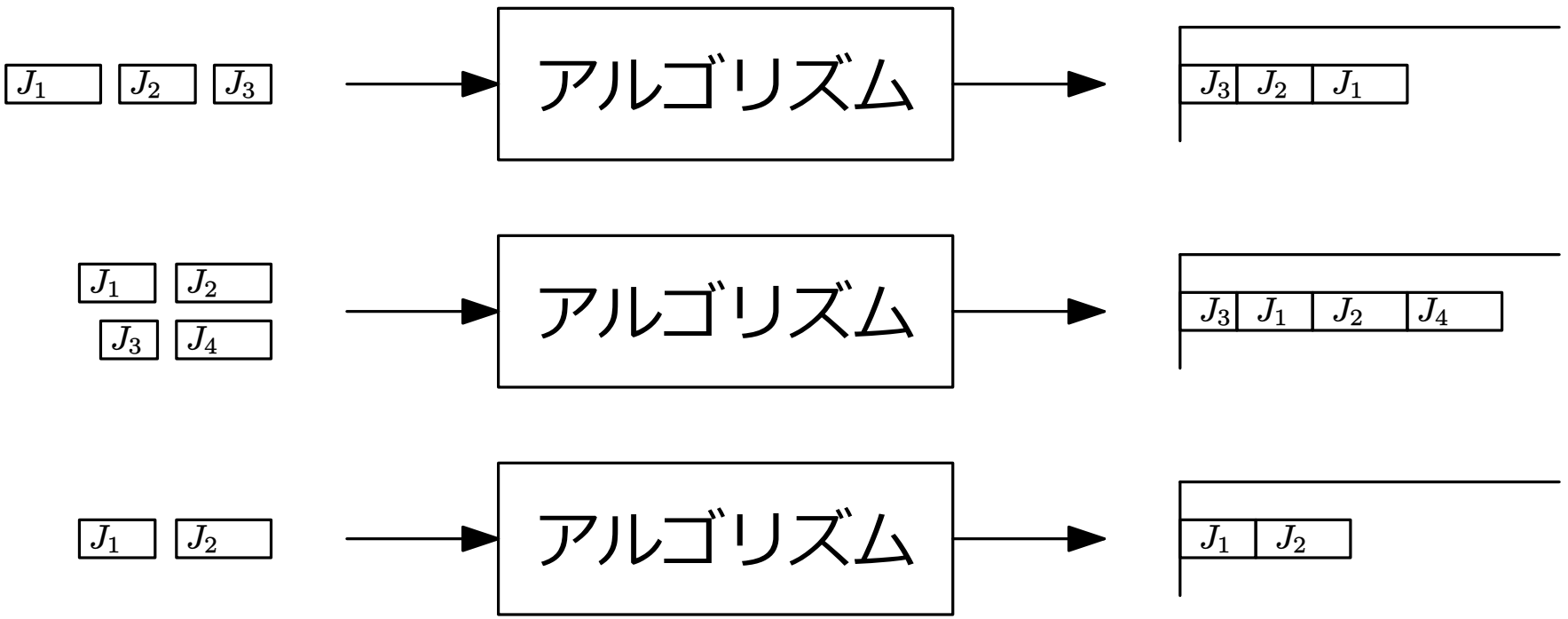
アルゴリズムには入力と出力がある

アルゴリズムが入力を処理した結果が出力である

問題 $1 \mid\mid \sum C_j$ の場合

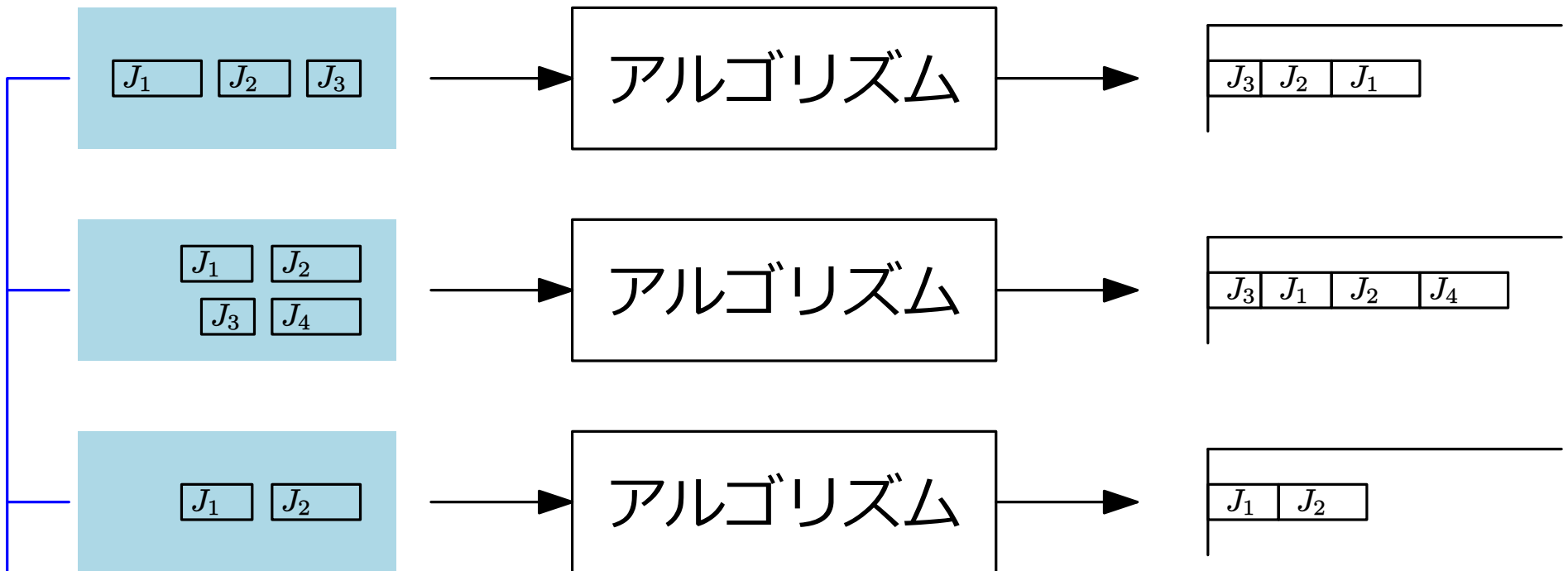
スケジューリングのアルゴリズム

問題 $1 \mid\mid \sum C_j$ の場合



スケジューリングのアルゴリズム

問題 $1 \mid\mid \sum C_j$ の場合



問題 $1 \mid\mid \sum C_j$ のインスタンス

注意 (アルゴリズム全般に関する)

問題 と **インスタンス** を区別する

スケジューリング問題のインスタンス

ジョブ・スケジューリングの問題

ジョブ・スケジューリングの問題の１つは，
3 つ組記法 $\alpha \mid \beta \mid \gamma$ で表される

ジョブ・スケジューリングの問題のインスタンス

問題 $\alpha \mid \beta \mid \gamma$ のインスタンスの１つは，
その問題を解くアルゴリズムの入力で表される

注意：問題によって，入力 (インスタンス) の形式は異なる

インスタンスの例 (1)

問題 $1 \mid\mid \sum C_j$ のインスタンス

$n \in \mathbb{Z}_{\geq 0}$, $p_1, p_2, \ldots, p_n \in \mathbb{Q}_{\geq 0}$

意味：n はジョブ数

p_j はジョブ J_j の処理時間 ($j \in [n]$)

インスタンスの例 (1)

問題 $1 \mid\mid \sum C_j$ のインスタンス

$n \in \mathbb{Z}_{\geq 0}$, $p_1, p_2, \ldots, p_n \in \mathbb{Q}_{\geq 0}$

意味：n はジョブ数

p_j はジョブ J_j の処理時間 $(j \in [n])$

問題 $R \mid\mid L_{\max}$ のインスタンス

$n \in \mathbb{Z}_{\geq 0}$, $m \in \mathbb{Z}_{\geq 0}$,
$p_{i,j} \in \mathbb{Q}_{\geq 0}$, $d_j \in \mathbb{Q}_{\geq 0}$ $(i \in [m], j \in [n])$

意味：n はジョブ数，m は機械数

$p_{i,j}$ は機械 M_i におけるジョブ J_j の処理時間

d_j はジョブ J_j の納期 $(i \in [m], j \in [n])$

インスタンスの例 (2)

問題 $P \mid\mid C_{\max}$ のインスタンス

$n \in \mathbb{Z}_{\geq 0}$, $m \in \mathbb{Z}_{\geq 0}$, $p_j \in \mathbb{Q}_{\geq 0}$ $(j \in [n])$

意味：n はジョブ数，m は機械数

p_j はジョブ J_j の処理時間 $(j \in [n])$

問題 $Pm \mid\mid C_{\max}$ のインスタンス

$n \in \mathbb{Z}_{\geq 0}$, $p_j \in \mathbb{Q}_{\geq 0}$ $(j \in [n])$

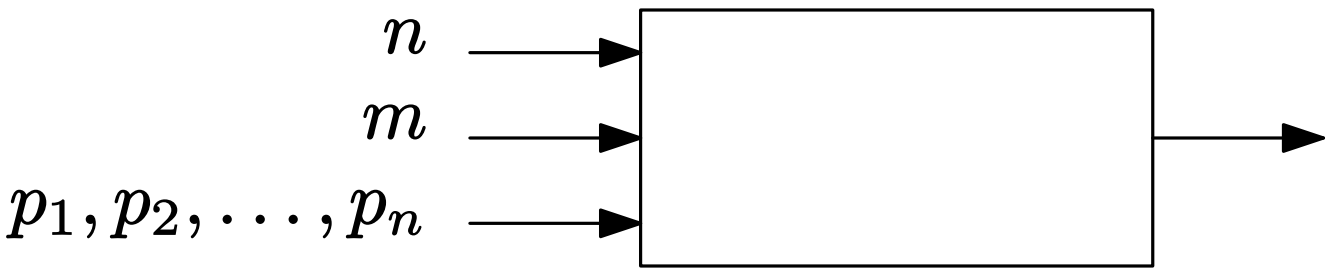
意味：n はジョブ数

p_j はジョブ J_j の処理時間 $(j \in [n])$

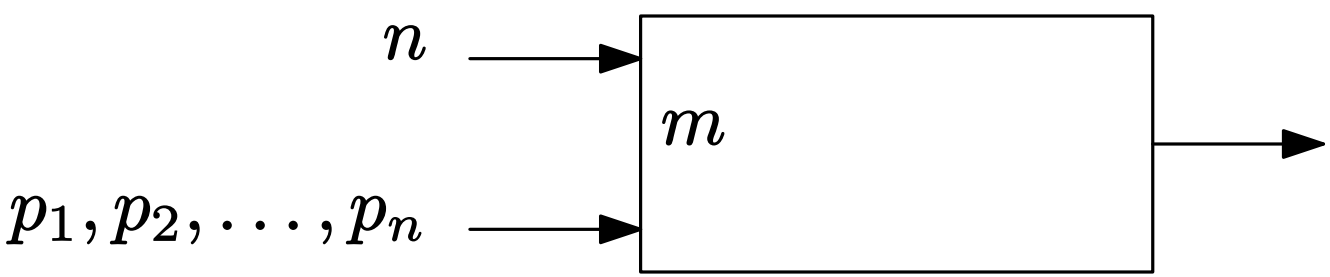
注意：m は定数 (入力の一部ではない)

インスタンスの例 (2) 続き

$P \mid\mid C_{\max}$ を解く
アルゴリズム



$Pm \mid\mid C_{\max}$ を解く
アルゴリズム



Word RAM：採用する計算モデル

注意 (アルゴリズム全般において)

アルゴリズムを記述するためには
計算モデル を指定する必要がある

この講義では，**word RAM** を使う (Fredman, Williad '90)

- word $\rightsquigarrow$ 語を扱える
- RAM $\rightsquigarrow$ ランダム・アクセス機械
  番地指定でメモリへアクセスできる

仮定
- 入力数値は 1 語に収まる

1 ステップでできること
- 語に対する算術演算，論理演算，比較演算，ビット演算
- 番地指定によるメモリアクセス，条件分岐，出力 (停止)

$\rightsquigarrow$「普通のプログラミング」でできることは大体できると思ってよい

アルゴリズムの計算量

定義 (非形式) : アルゴリズムの計算量

問題 $\alpha \mid \beta \mid \gamma$ を解くアルゴリズムの計算量とは,
問題 $\alpha \mid \beta \mid \gamma$ の各インスタンス I に対して,
出力を行うまでに費やすステップ数

- ステップ数はインスタンスに依存する

例 : $1 \mid\mid \sum C_j$ に対する SPT 法の計算量が $O(n \log n)$ とは

$O(n \log n)$ ステップ

(オーダーが高々 $n \log n$ ステップ)

インスタンスのサイズ

スケジューリング問題のインスタンス

- ジョブ数 n
- 機械数 m
- 処理時間 $p_{i,j}$
- 納期 d_j
- 到着時刻 r_j
- $\ldots$

インスタンスのサイズ

スケジューリング問題のインスタンス　　サイズ

- ジョブ数 n
- 機械数 m
- 処理時間 $p_{i,j}$
- 納期 d_j
- 到着時刻 r_j
- $\ldots$

- $O(\log p_{i,j})$
- $O(\log d_j)$
- $O(\log r_j)$

計算量の測り方

計算量は次をパラメータとして算出する

- 入力となる集合の要素数：n，m，$\ldots$
- 入力となる数値：$p_{i,j}$，d_j，r_j，$\ldots$

多項式時間

代表記法：N = 集合の要素数，P = 数値

アルゴリズムの計算量の概念

アルゴリズムの計算量が { **強多項式時間** / **弱多項式時間** / **擬多項式時間** } とは，

任意のインスタンスに対して，

計算量が高々 { N / $N, \log P$ / N, P } の多項式であること

注：強多項式時間 $\Rightarrow$ 弱多項式時間 $\Rightarrow$ 擬多項式時間

多項式時間　　多項式時間ではない

多項式時間

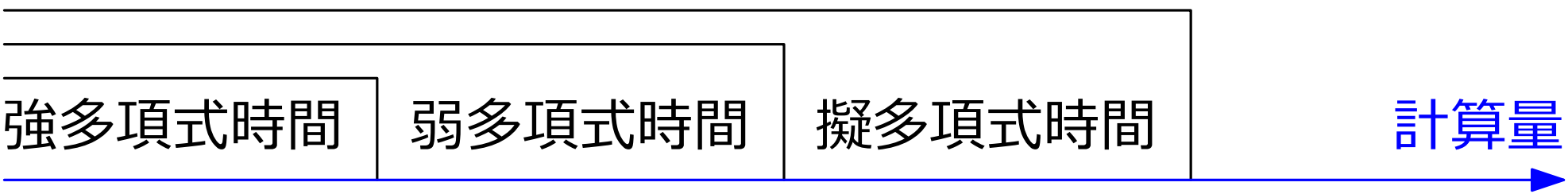
代表記法：N = 集合の要素数，P = 数値

アルゴリズムの計算量の概念

アルゴリズムの計算量が { **強多項式時間** / **弱多項式時間** / **擬多項式時間** } とは，

任意のインスタンスに対して，

計算量が高々 { N / $N, \log P$ / N, P } の多項式であること



多項式時間：例

- $O(n \log n)$ 強多項式時間
- $O(n^2 \log \sum p_j)$ 弱多項式時間
- $O(n^4 \sum p_j)$ 擬多項式時間

アルゴリズムの計算量の概念

アルゴリズムの計算量が { **強多項式時間** / **弱多項式時間** / **擬多項式時間** } とは，

任意のインスタンスに対して，

計算量が高々 { N / $N, \log P$ / N, P } の多項式であること

まとめ：一機械，ジョブの特性なし

まとめ：$1 \mid\mid \gamma$

γ		計算量	
$C_{\max}$	最大完了時刻	$O(n)$	強多項式時間
$\sum C_j$	総完了時刻	$O(n \log n)$	強多項式時間
$L_{\max}$	最大納期ずれ	$O(n \log n)$	強多項式時間
$\sum T_j$	総納期遅れ	$O(n^4 \sum p_j)$	擬多項式時間
$\sum U_j$	納期遅れジョブ数	$O(n \log n)$	強多項式時間

本日の内容

1. アルゴリズムと計算量の基本的な考え方
2. **NP 困難性の基本的な考え方**
3. ジョブ・スケジューリング問題の NP 困難性証明の例

- R. K. Karp, Reducibility Among Combinatorial Problems. In R. E. Miller, J. W. Thatcher (eds.). *Complexity of Computer Computations*. Plenum, 1972, pp. 85–103.
- M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

多項式時間で解けないとき？

前回の結果 (再掲)

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

擬多項式時間 (多項式時間ではない)

Q $1 \mid\mid \sum T_j$ は多項式時間で解けるか？

A 分からない (未解決問題)

多項式時間で解けないとき？

前回の結果 (再掲)

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

擬多項式時間 (多項式時間ではない)

Q $1 \mid\mid \sum T_j$ は多項式時間で解けるか？

A 分からない (未解決問題)

アルゴリズム界の現状

問題が **NP 困難** であることを証明して
多項式時間で解けないことの傍証としている

NP 困難性の使用法

ここでは，NP 困難性の考え方を応用するだけなので，
定義はせずに，その使い方のみを説明する

用語法：NP 困難

NP 困難 (NP-hard) という用語は，問題に対して使う

NP 困難性の使用法

ここでは，NP 困難性の考え方を応用するだけなので，
定義はせずに，その使い方のみを説明する

用語法：NP 困難

NP 困難 (NP-hard) という用語は，問題に対して使う

事実：NP 困難性の使い方

問題 $\alpha \mid \beta \mid \gamma$ が NP 困難，かつ，$\mathrm{P} \neq \mathrm{NP}$
$\Rightarrow$ 問題 $\alpha \mid \beta \mid \gamma$ は多項式時間で解けない

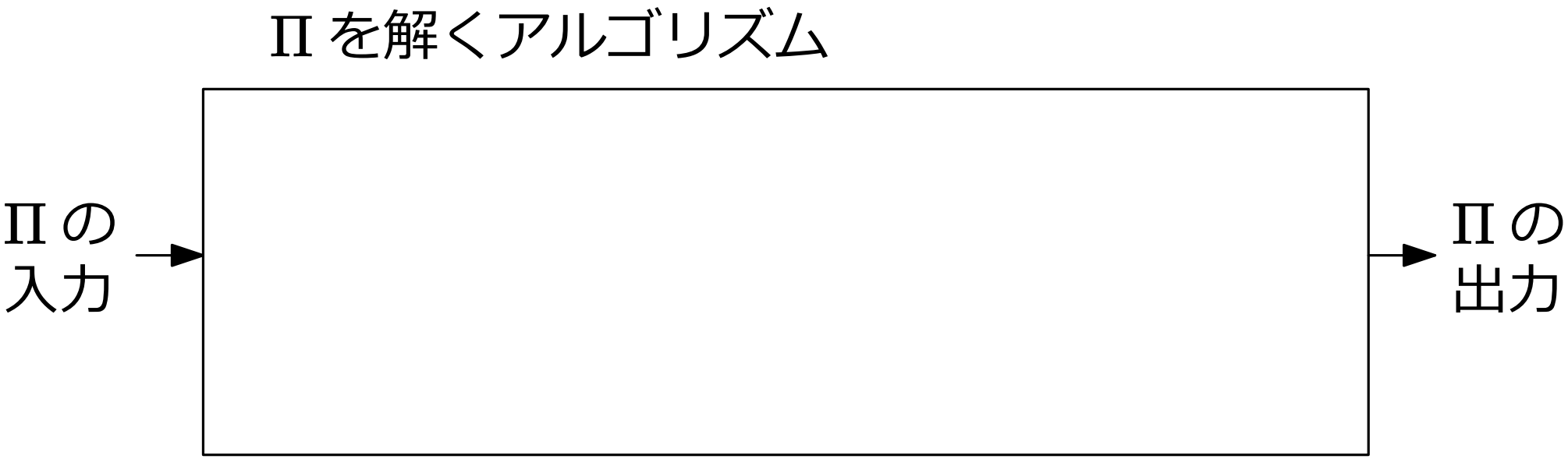
未解決問題：$\mathrm{P} \neq \mathrm{NP}$

NP 困難性の証明法

NP 困難性の証明法

問題 $\alpha \mid \beta \mid \gamma$ が NP 困難だと証明したいとき

1. 既に NP 困難だと知られている問題 Π を持ってくる
2. Π を $\alpha \mid \beta \mid \gamma$ に多項式時間で **帰着** する

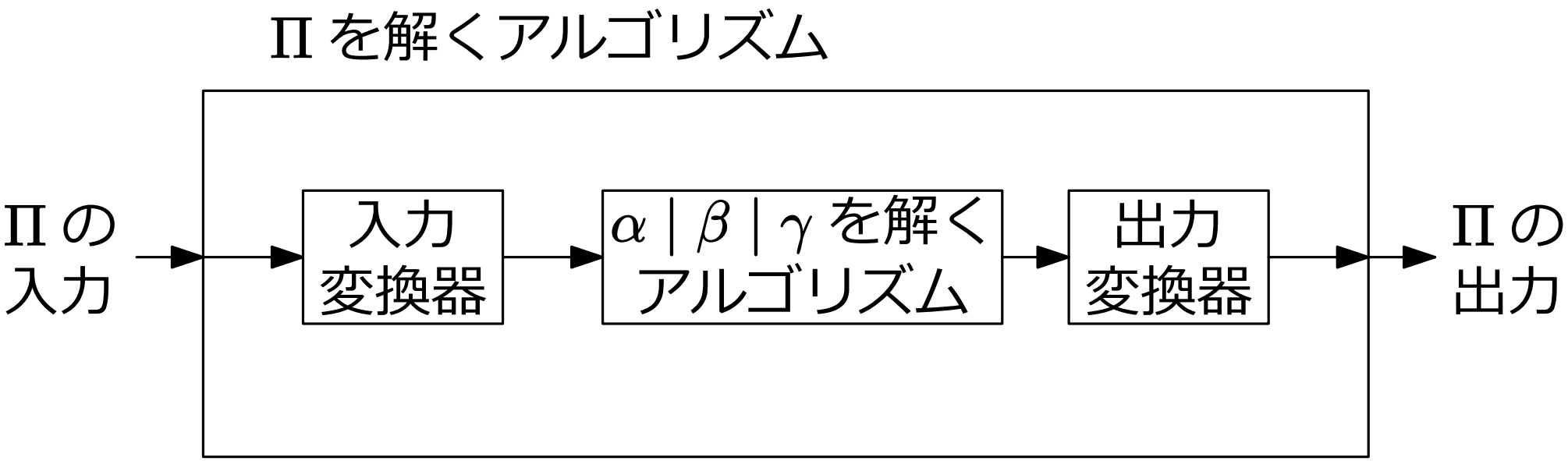


NP 困難性の証明法

NP 困難性の証明法

問題 $\alpha \mid \beta \mid \gamma$ が NP 困難だと証明したいとき

1. 既に NP 困難だと知られている問題 Π を持ってくる
2. Π を $\alpha \mid \beta \mid \gamma$ に多項式時間で **帰着** する

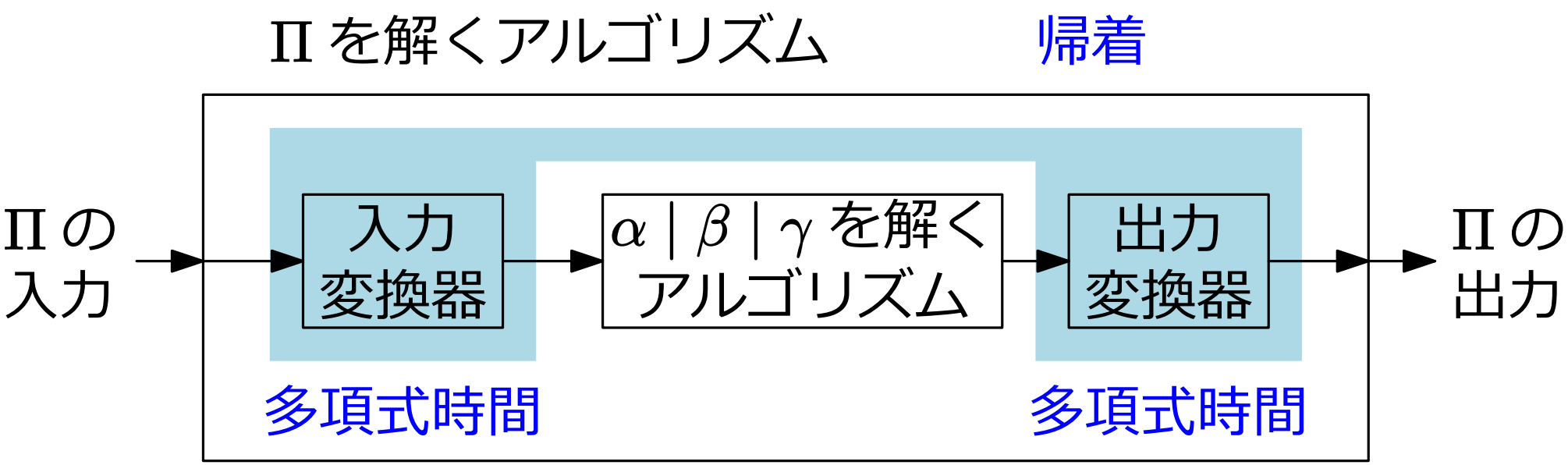


NP 困難性の証明法

NP 困難性の証明法

問題 $\alpha \mid \beta \mid \gamma$ が NP 困難だと証明したいとき

1. 既に NP 困難だと知られている問題 Π を持ってくる
2. Π を $\alpha \mid \beta \mid \gamma$ に多項式時間で **帰着** する

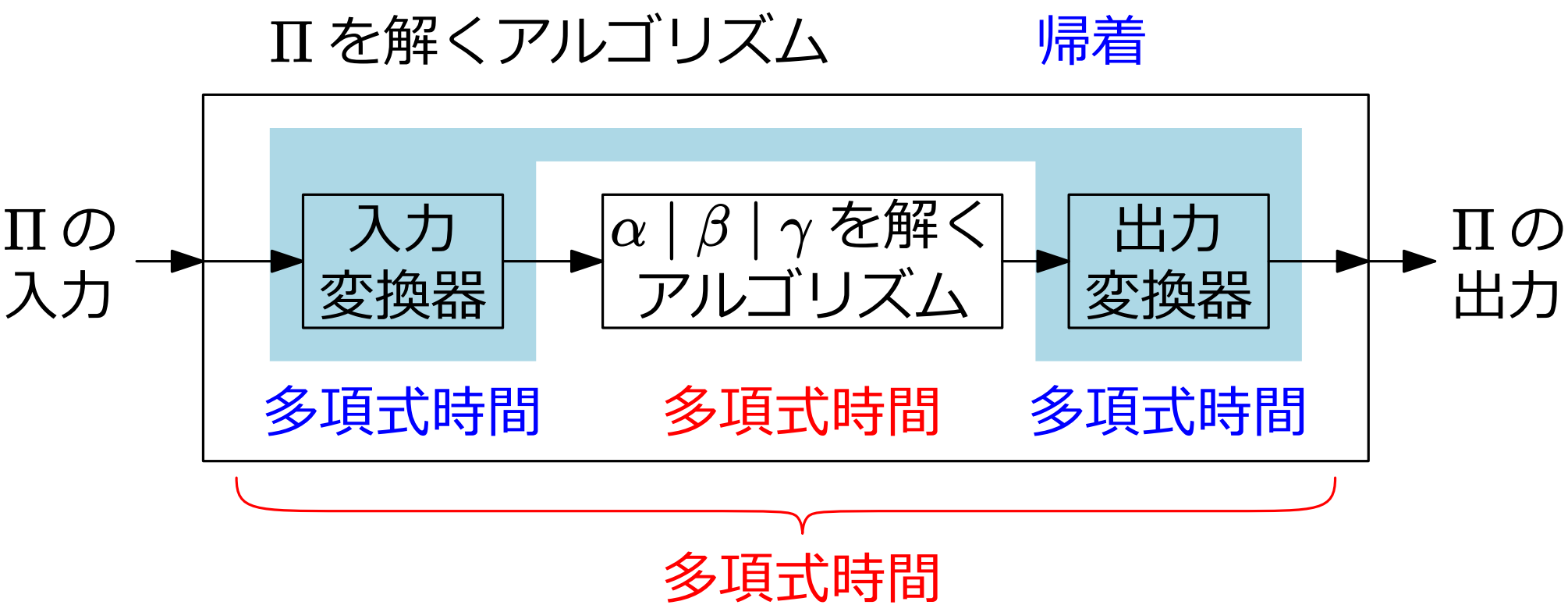


NP 困難性の証明法

NP 困難性の証明法

問題 $\alpha \mid \beta \mid \gamma$ が NP 困難だと証明したいとき

1. 既に NP 困難だと知られている問題 Π を持ってくる
2. Π を $\alpha \mid \beta \mid \gamma$ に多項式時間で **帰着** する



NP 困難性の証明法：帰結

NP 困難性の証明法

問題 $\alpha \mid \beta \mid \gamma$ が NP 困難だと証明したいとき

1. 既に NP 困難だと知られている問題 Π を持ってくる
2. Π を $\alpha \mid \beta \mid \gamma$ に多項式時間で **帰着** する

NP 困難性の帰結

Π が多項式時間で解けない

$\Rightarrow \alpha \mid \beta \mid \gamma$ も多項式時間で解けない

NP 困難だと知られている問題

たくさんあるが，この授業では，Karp の 21 問題を使う

(Karp ’72)

- SAT
- 01-IP
- クリーク
- 集合充填
- 点被覆
- 集合被覆
- 帰還点集合
- 帰還弧集合
- 有向ハミルトン閉路
- 無向ハミルトン閉路
- 3-SAT
- 染色数
- クリーク被覆
- 厳密被覆
- 横断集合
- シュタイナー木
- 3 次元マッチング
- ナップサック
- ジョブ並び替え
- 分割
- 最大カット

スケジューリングでは，次の問題もよく使う

(Garey, Johnson ’75)

- 3 分割

NP 困難だと知られている問題

たくさんあるが，この授業では，Karp の 21 問題を使う

(Karp '72)

- SAT
- 01-IP
- クリーク
- 集合充填
- 点被覆
- 集合被覆
- 帰還点集合
- 帰還弧集合
- 有向ハミルトン閉路
- 無向ハミルトン閉路
- 3-SAT
- 染色数
- クリーク被覆
- 厳密被覆
- 横断集合
- シュタイナー木
- 3 次元マッチング
- ナップサック
- ジョブ並び替え $\leftarrow 1 \mid\mid \sum w_j U_j$
- 分割
- 最大カット

スケジューリングでは，次の問題もよく使う

(Garey, Johnson '75)

- 3 分割

NP 困難だと知られている問題

たくさんあるが，この授業では，Karp の 21 問題を使う

(Karp '72)

- SAT
- 01-IP
- クリーク
- 集合充填
- 点被覆
- 集合被覆
- 帰還点集合
- 帰還弧集合
- 有向ハミルトン閉路
- 無向ハミルトン閉路
- 3-SAT
- 染色数
- クリーク被覆
- 厳密被覆
- 横断集合
- シュタイナー木
- 3 次元マッチング
- ナップサック
- ジョブ並び替え $\leftarrow 1 \mid\mid \sum w_j U_j$
- 分割
- 最大カット

スケジューリングでは，次の問題もよく使う

(Garey, Johnson '75)

- 3 分割

分割問題

定義：分割問題 (partition problem)

正整数 $a_1, a_2, \ldots, a_n$ が与えられたとき，
それらを同じ和の 2 つのグループに分けられるか？

例：$1, 1, 2, 2, 3, 4, 6, 6, 7, 8$

分割問題

定義：分割問題 (partition problem)

正整数 $a_1, a_2, \ldots, a_n$ が与えられたとき，
それらを同じ和の 2 つのグループに分けられるか？

例：$1, 1, 2, 2, 3, 4, 6, 6, 7, 8$

- $1 + 1 + 2 + 2 + 6 + 8 = 20$
- $3 + 4 + 6 + 7 = 20$

できる

分割問題

定義：分割問題 (partition problem)

正整数 $a_1, a_2, \ldots, a_n$ が与えられたとき，
それらを同じ和の 2 つのグループに分けられるか？

例：$1, 1, 2, 2, 3, 4, 6, 6, 7, 8$

- $1 + 1 + 2 + 2 + 6 + 8 = 20$
- $3 + 4 + 6 + 7 = 20$

できる

例：$1, 1, 2, 3, 4, 5, 6, 6, 7, 10$

できない

分割問題

定義：分割問題 (partition problem)

正整数 $a_1, a_2, \ldots, a_n$ が与えられたとき，
それらを同じ和の 2 つのグループに分けられるか？

例：$1, 1, 2, 2, 3, 4, 6, 6, 7, 8$

- $1 + 1 + 2 + 2 + 6 + 8 = 20$
- $3 + 4 + 6 + 7 = 20$

できる

例：$1, 1, 2, 3, 4, 5, 6, 6, 7, 10$

できない

事実 (Karp '72)

分割問題は 弱 NP 困難

3 分割問題

定義：3 分割問題 (3-partition problem)

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

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

それらを同じ和の n 個のグループに分けられるか？

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

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

できる

3 分割問題

定義：3 分割問題 (3-partition problem)

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

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

それらを同じ和の n 個のグループに分けられるか？

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

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

できる

事実 (Garey, Johnson '75)

3 分割問題 は強 NP 困難

弱 NP 困難と強 NP 困難

代表記法：N = 集合の要素数，P = 数値

事実：強 NP 困難性の使い方

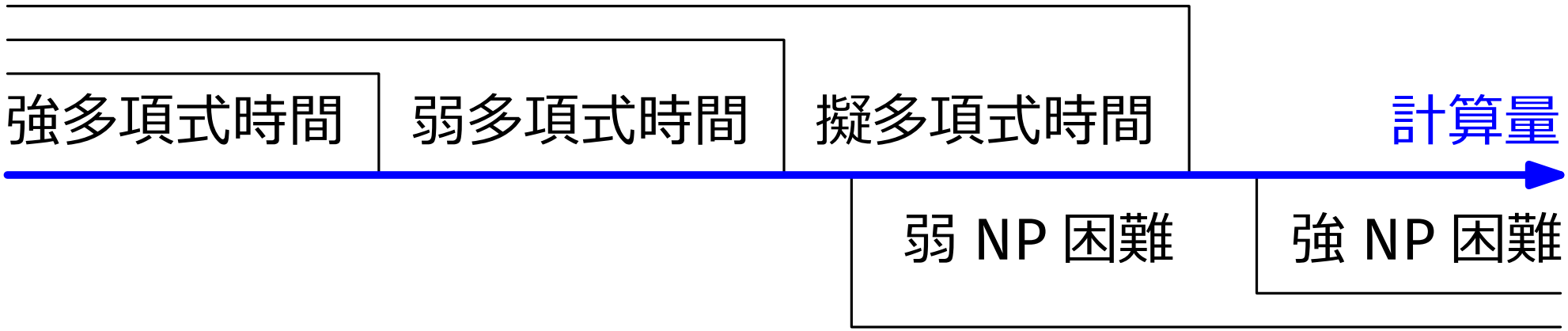
問題 $\alpha \mid \beta \mid \gamma$ が強 NP 困難，かつ，P $\neq$ NP
$\Rightarrow \alpha \mid \beta \mid \gamma$ は N, P に関する多項式時間で解けない
（$\alpha \mid \beta \mid \gamma$ を解く擬多項式時間アルゴリズムはない）

事実：弱 NP 困難性の使い方

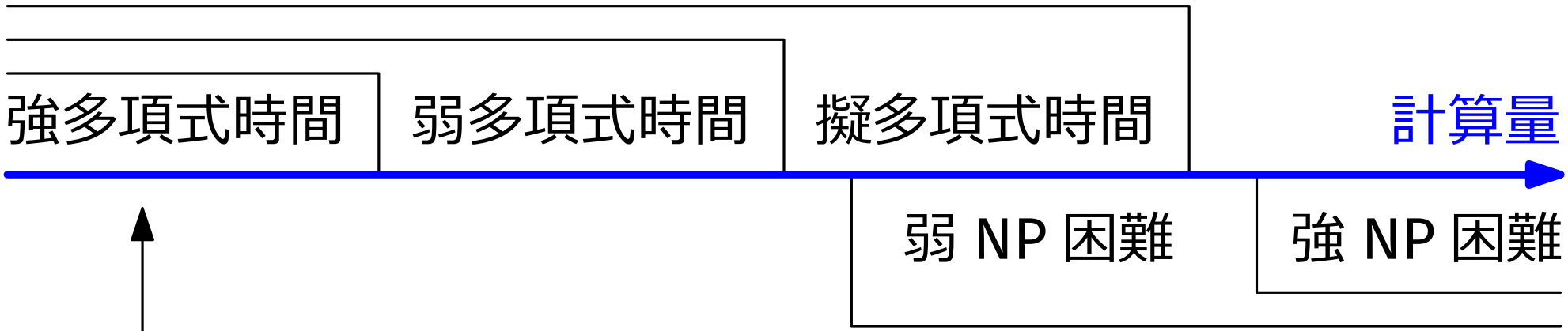
問題 $\alpha \mid \beta \mid \gamma$ が弱 NP 困難，かつ，P $\neq$ NP
$\Rightarrow \alpha \mid \beta \mid \gamma$ は $N, \log P$ に関する多項式時間で解けない
（$\alpha \mid \beta \mid \gamma$ を解く弱多項式時間アルゴリズムはない）

- 弱 NP 困難 = (普通の意味での) NP 困難
- 強 NP 困難 $\Rightarrow$ 弱 NP 困難

多項式時間と NP 困難



多項式時間と NP 困難



$1 \mid\mid C_{\max}$

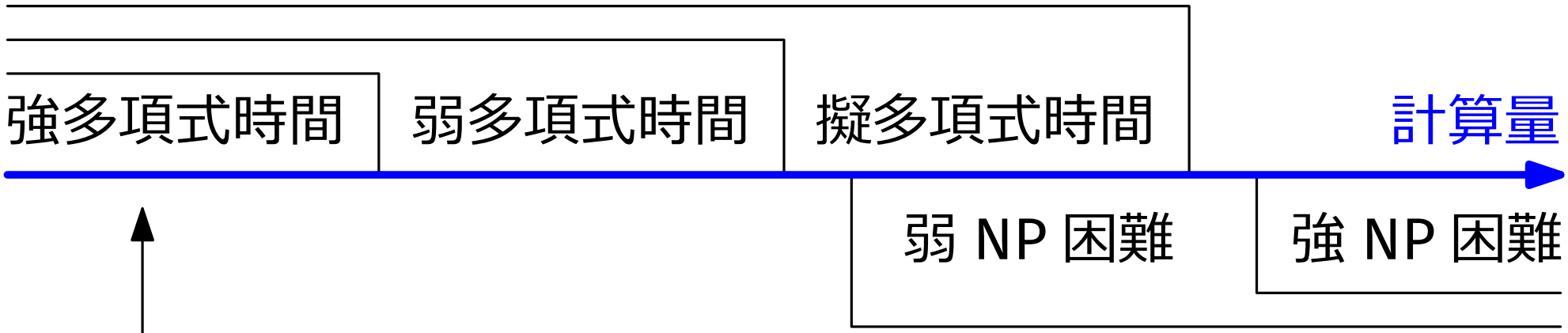
$1 \mid\mid \sum C_j$

$1 \mid\mid L_{\max}$

$1 \mid\mid \sum U_j$

$1 \mid\mid \sum T_j$

多項式時間と NP 困難



$1 \mid\mid C_{\max}$
$1 \mid\mid \sum C_j$
$1 \mid\mid L_{\max}$
$1 \mid\mid \sum U_j$

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

$P \mid\mid C_{\max}$
$1 \mid r_j \mid L_{\max}$

本日の内容

1. アルゴリズムと計算量の基本的な考え方
2. NP 困難性の基本的な考え方
3. **ジョブ・スケジューリング問題の NP 困難性証明の例**

- J. K. Lenstra, A. H. G. Rinnooy Kan, P. Brucker, Complexity of machine scheduling problems. *Annals of Discrete Mathematics* 1 (1977) pp. 343–362.

教訓 (アルゴリズム全般において)

NP 困難性の証明は，アルゴリズム設計者が行うもの

紹介する NP 困難性証明

定理 (Lenstra, Rinnooy Kan, Brucker '77)

1. $P2 \,||\, C_{\max}$ は 弱 NP 困難
2. $P \,||\, C_{\max}$ は 強 NP 困難
3. $1 \,|\, r_j \,|\, L_{\max}$ は 強 NP 困難

注：任意の整数 $m \geq 2$ に対して，$Pm \,||\, C_{\max}$ も弱 NP 困難

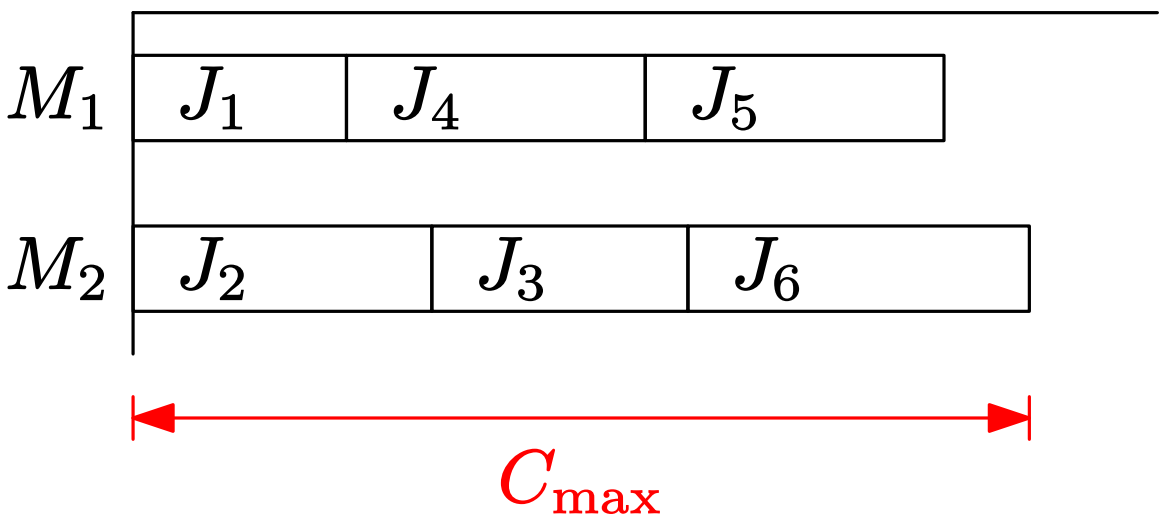
(詳細は付録)

機械の環境

- 同一並列機械，機械数 $= 2$

最適化する目的

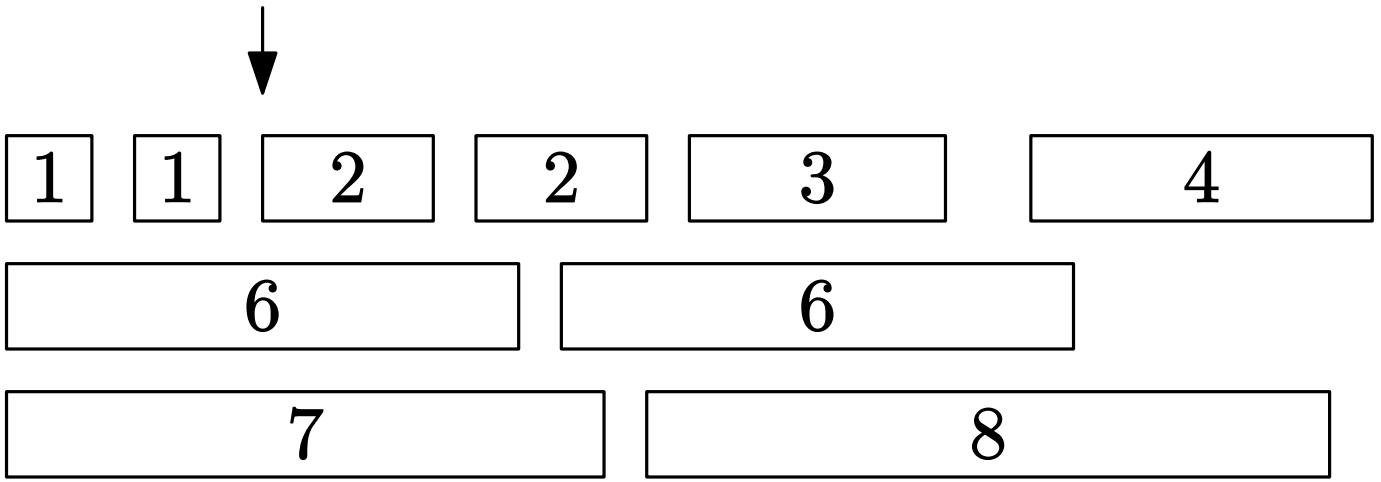
- 最大完了時刻の最小化



$P2 \mid\mid C_{\max}$ の弱 NP 困難性 (1)

証明：分割問題を帰着する

入力の変換： $1, 1, 2, 2, 3, 4, 6, 6, 7, 8$

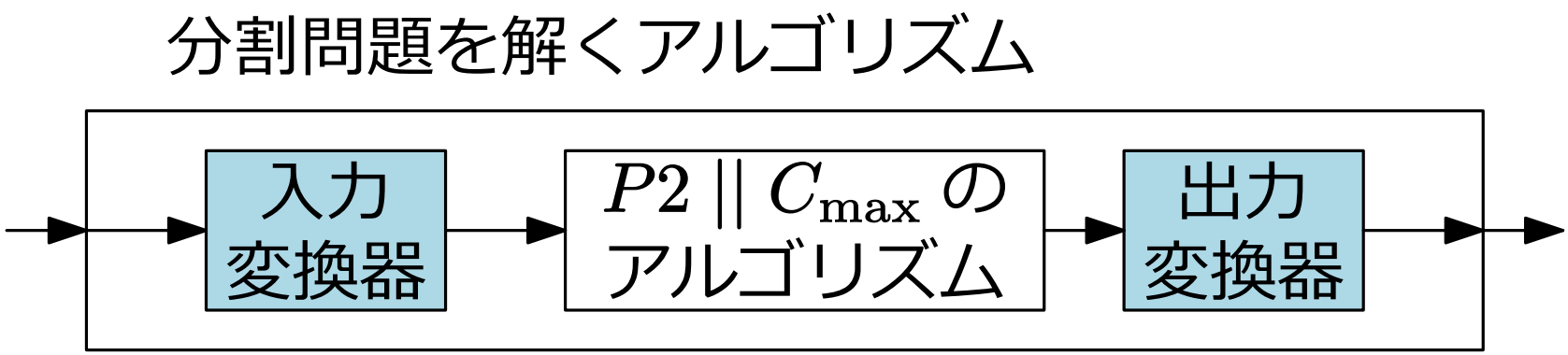


分割問題を解くアルゴリズム

$P2 \mid\mid C_{\max}$ の弱 NP 困難性 (1)

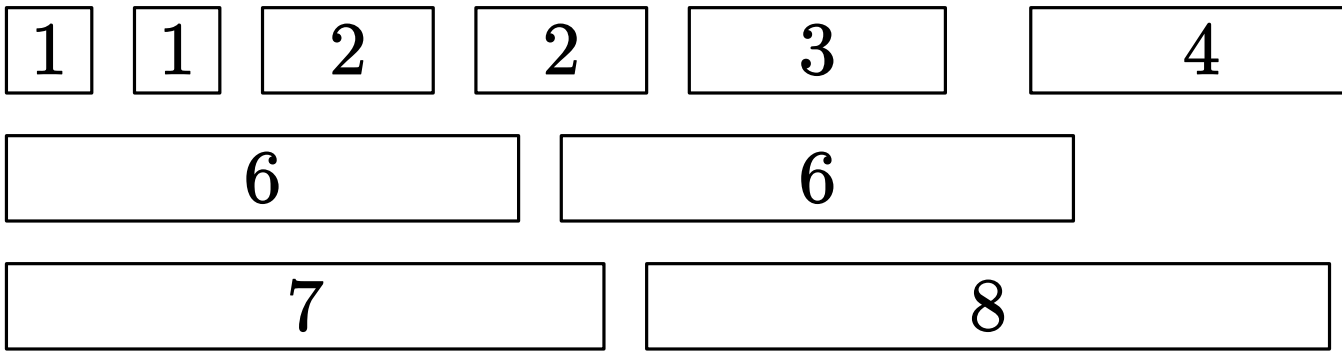
証明：分割問題を帰着する

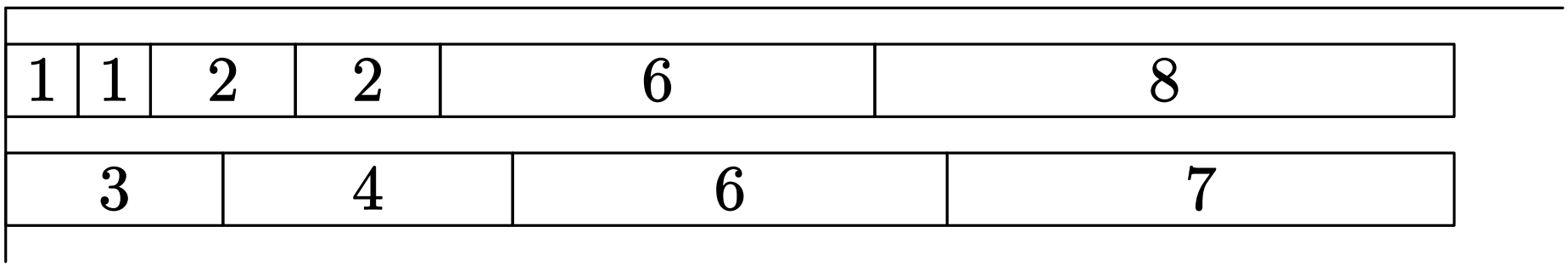
入力の変換： n 個の整数 $a_1, a_2, \ldots, a_n$ に対して，
n 個のジョブ $J_1, J_2, \ldots, J_n$ を作って，
$p_j = a_j$ $(j \in [n])$ とする



$P2 \mid\mid C_{\max}$ の弱 NP 困難性 (2)

証明：分割問題を帰着する



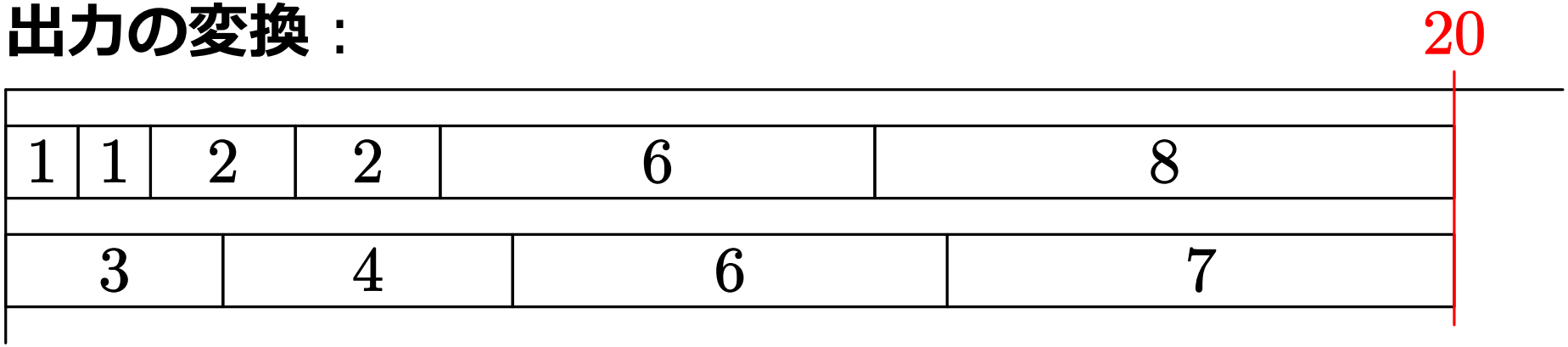


分割問題を解くアルゴリズム

$P2 \mid\mid C_{\max}$ の弱 NP 困難性 (3)

証明：分割問題を帰着する

出力の変換：



$$1+1+2+2+6+8 = 20$$

$$\rightarrow$$

$$3+4+6+7 = 20$$

分割問題を解くアルゴリズム

$P2 \mid\mid C_{\max}$ の弱 NP 困難性 (3)

証明：分割問題を帰着する

出力の変換： $P2 \mid\mid C_{\max}$ のインスタンスとして

最適値 $= \frac{1}{2}\sum_{i=1}^{n} a_i$ $\Rightarrow$ 分割問題の答えは「できる」

最適値 $\neq \frac{1}{2}\sum_{i=1}^{n} a_i$ $\Rightarrow$ 分割問題の答えは「できない」

(本来は，この変換が正しいことを証明しないといけない) □

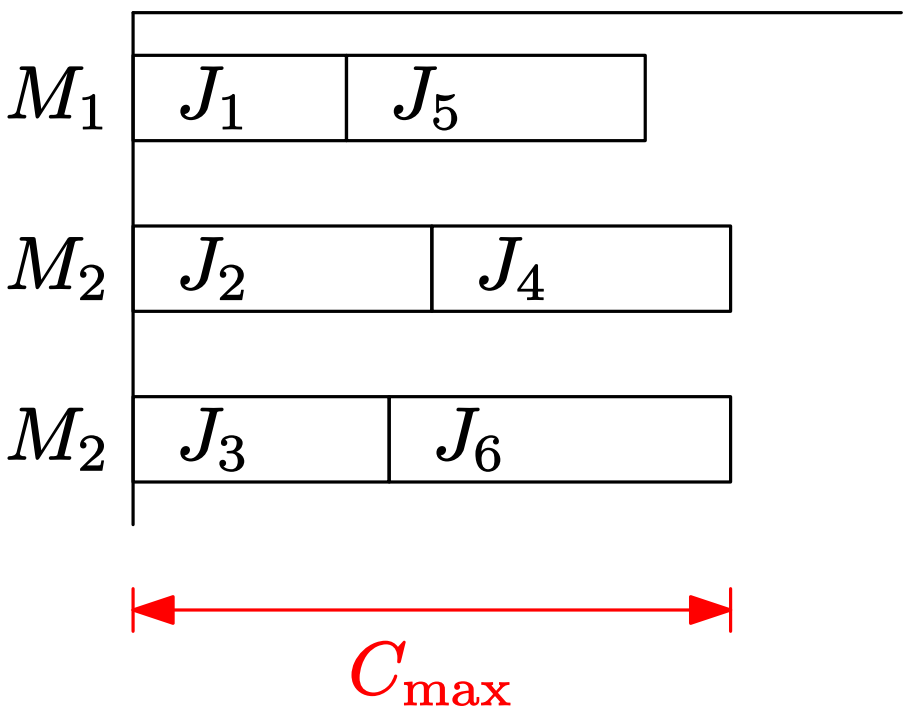
分割問題を解くアルゴリズム

機械の環境

- 同一並列機械，機械数は可変 (入力の一部)

最適化する目的

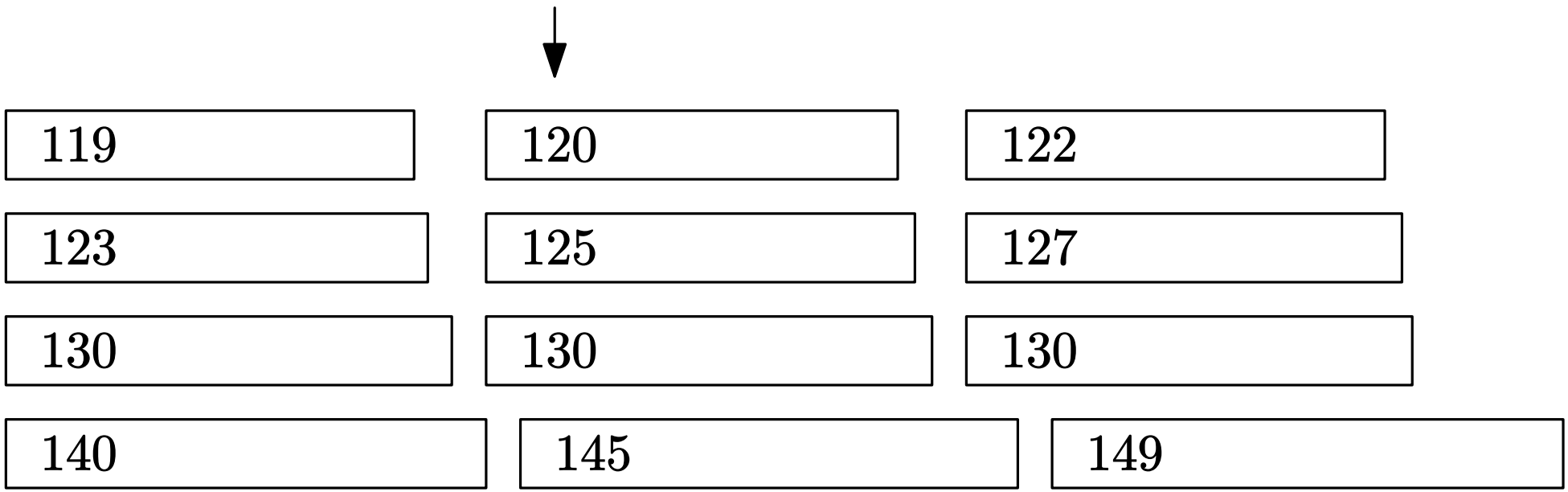
- 最大完了時刻の最小化



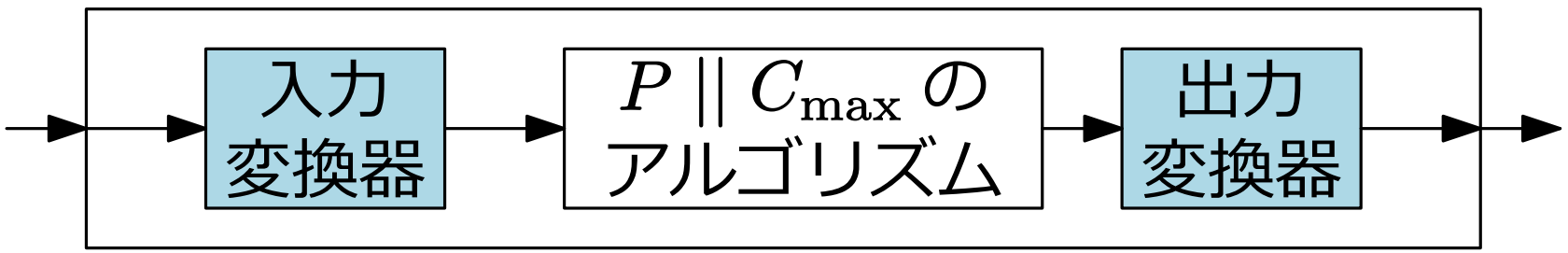
$P \mid\mid C_{\max}$ の強 NP 困難性 (1)

証明：3 分割問題を帰着する

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



3 分割問題を解くアルゴリズム

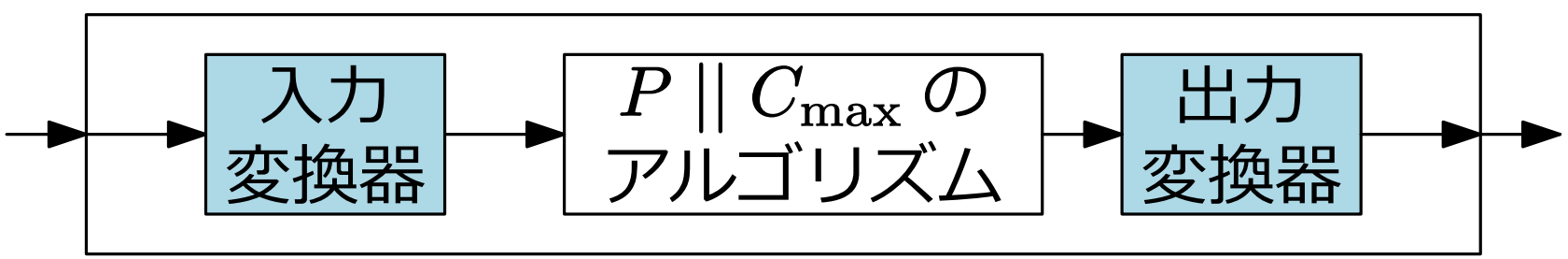


$P \mid\mid C_{\max}$ の強 NP 困難性 (1)

証明：3 分割問題を帰着する

入力の変換： $3n$ 個の整数 $a_1, a_2, \ldots, a_{3n}$ に対して，
$3n$ 個のジョブ $J_1, J_2, \ldots, J_{3n}$ を作って，
$p_j = a_j$ $(j \in [3n])$ とする
機械数 m を n とする

3 分割問題を解くアルゴリズム

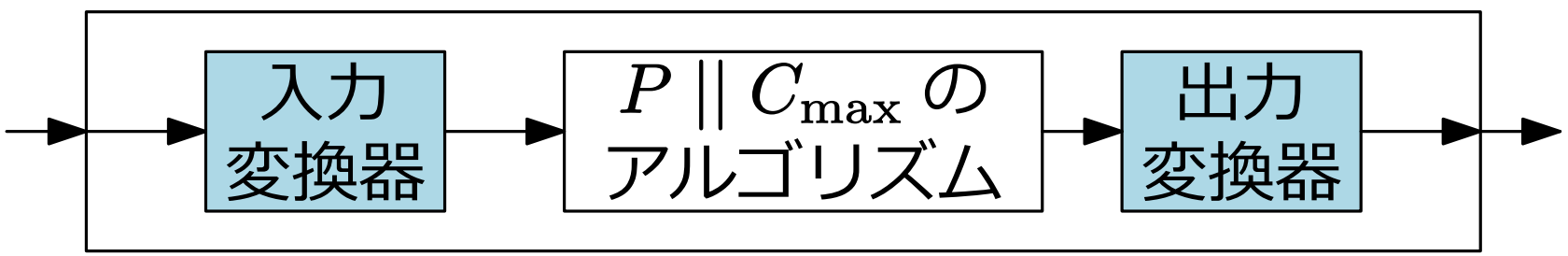


$P \mid\mid C_{\max}$ の強 NP 困難性 (2)

証明：3 分割問題を帰着する

120	125	145
123	127	140
119	122	149
130	130	130

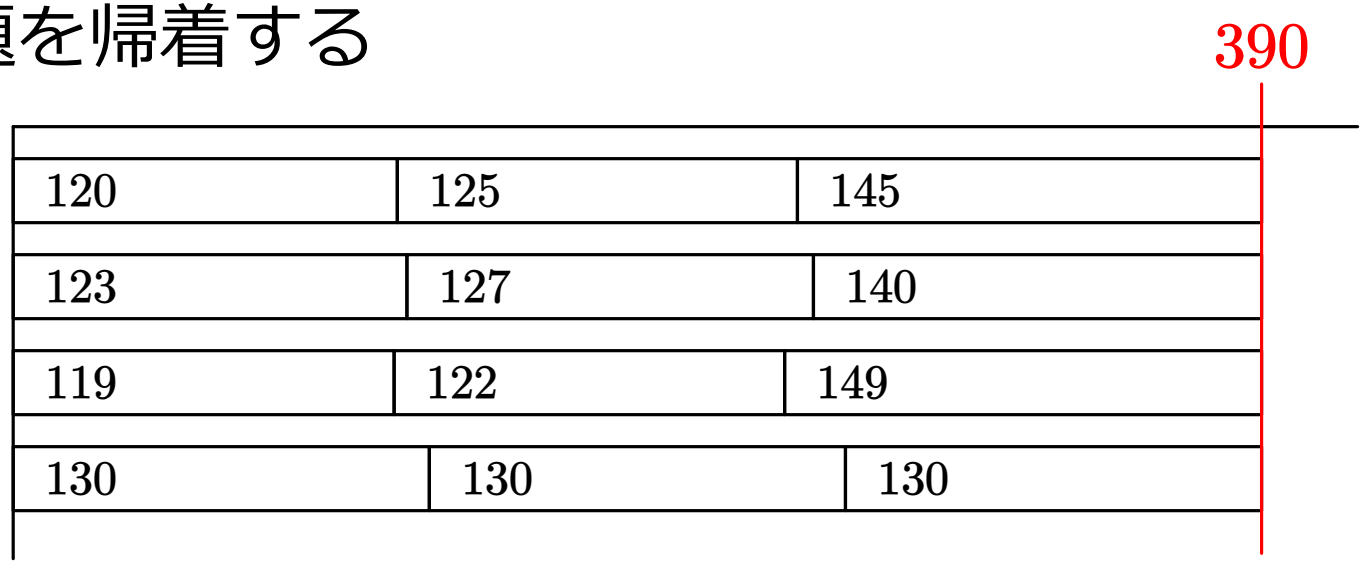
3 分割問題を解くアルゴリズム



$P \mid\mid C_{\max}$ の強 NP 困難性 (3)

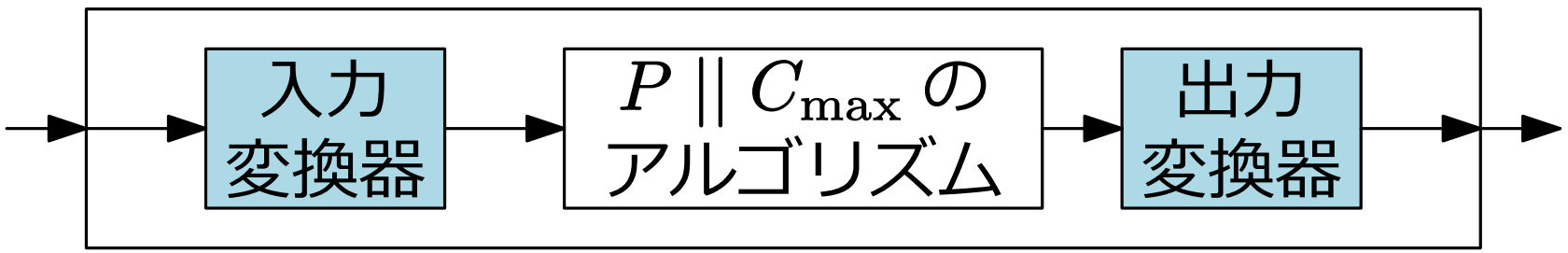
証明：3 分割問題を帰着する

出力の変換：



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

3 分割問題を解くアルゴリズム



$P \mid\mid C_{\max}$ の強 NP 困難性 (3)

証明：3 分割問題を帰着する

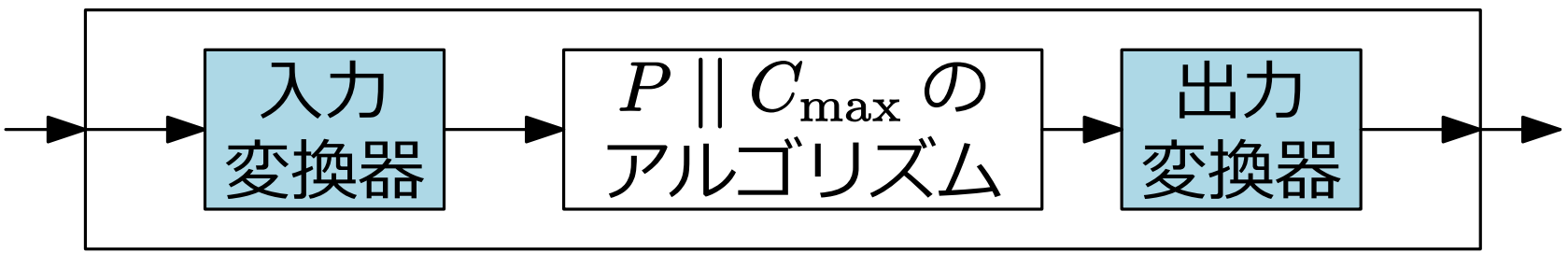
出力の変換： $P \mid\mid C_{\max}$ のインスタンスとして

$$\text{最適値} = \frac{1}{n}\sum_{j=1}^{3n} a_j \Rightarrow \text{3 分割問題の答えは「できる」}$$

$$\text{最適値} \neq \frac{1}{n}\sum_{j=1}^{3n} a_j \Rightarrow \text{3 分割問題の答えは「できない」}$$

(本来は，この変換が正しいことを証明しないといけない) □

3 分割問題を解くアルゴリズム



機械の環境

- 1 機械

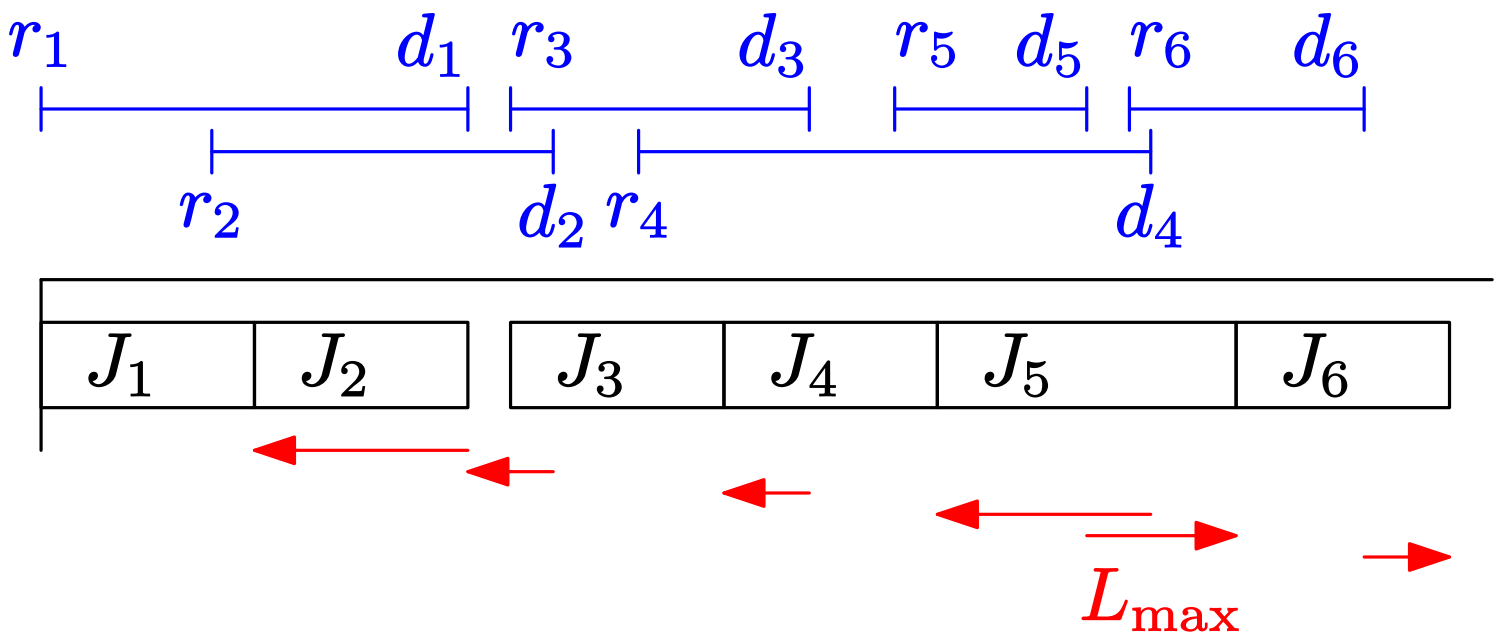
ジョブの特性

- 各ジョブ J_j に到着時刻 r_j，納期 d_j

最適化する目的

- 最大納期ずれの最小化

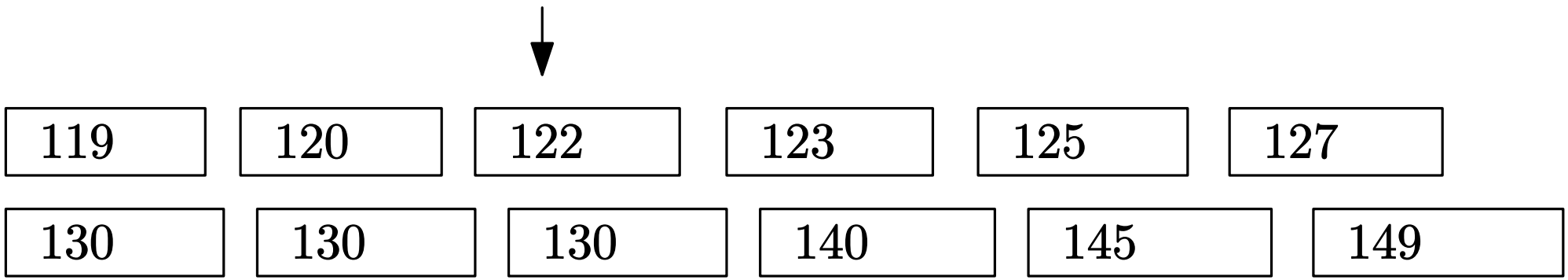
$$L_j = C_j - d_j$$



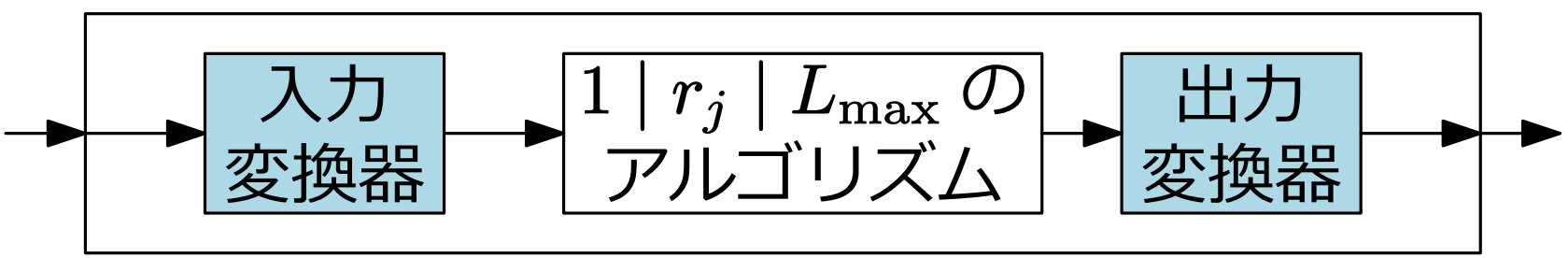
$1 \mid r_j \mid L_{\max}$ の強 NP 困難性 (1)

証明：3 分割問題を帰着する

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



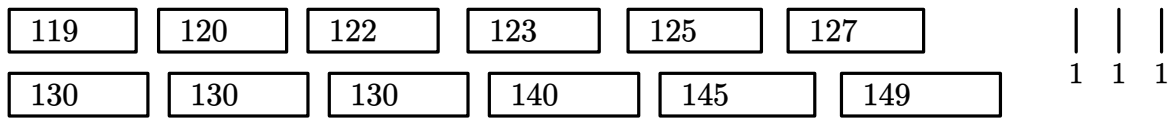
3 分割問題を解くアルゴリズム



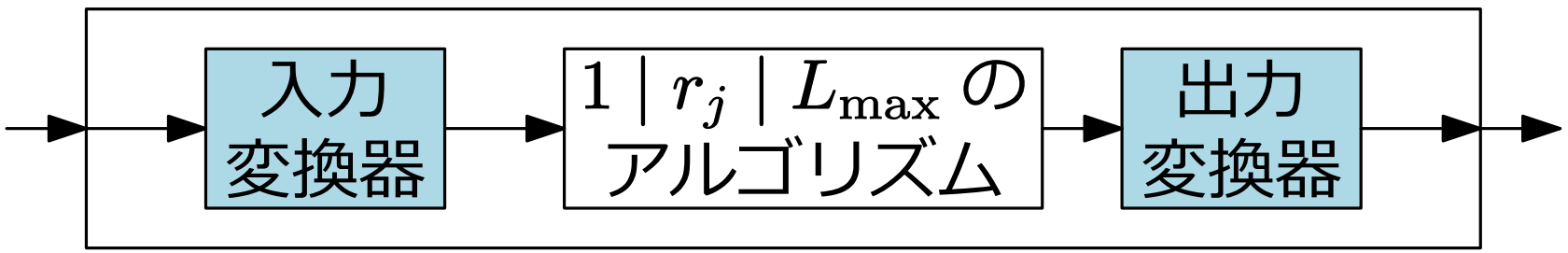
$1 \mid r_j \mid L_{\max}$ の強 NP 困難性 (2)

証明：3 分割問題を帰着する

入力の変換：



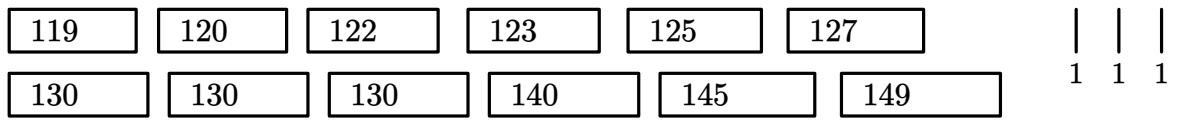
3 分割問題を解くアルゴリズム

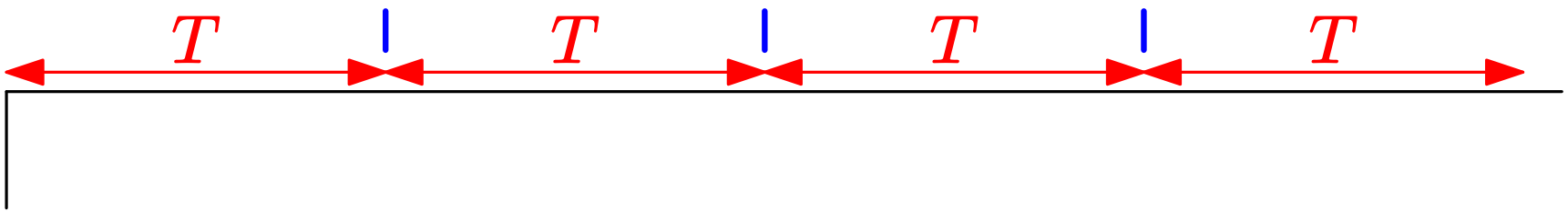


$1 \mid r_j \mid L_{\max}$ の強 NP 困難性 (2)

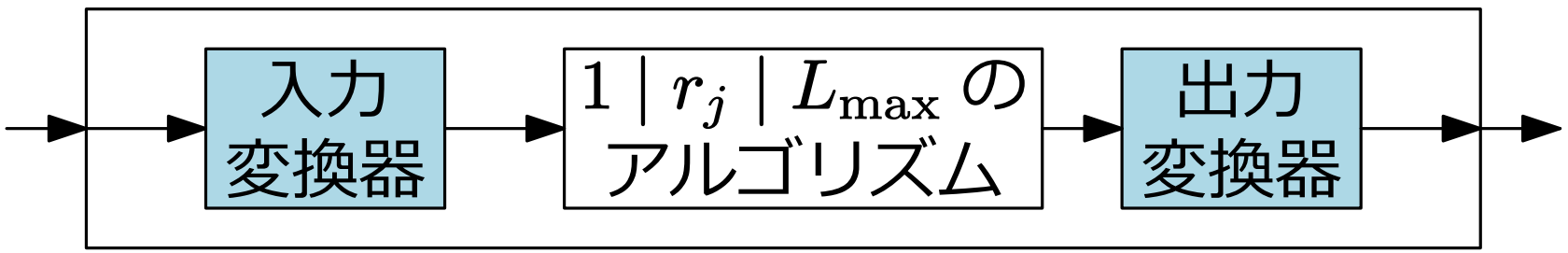
証明：3 分割問題を帰着する

入力の変換：





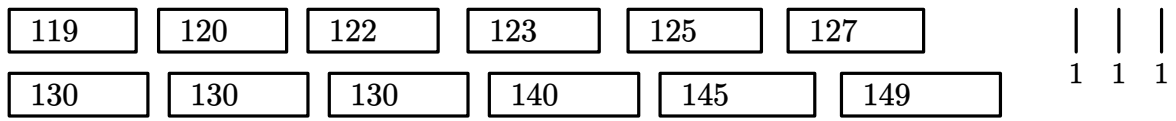
3 分割問題を解くアルゴリズム

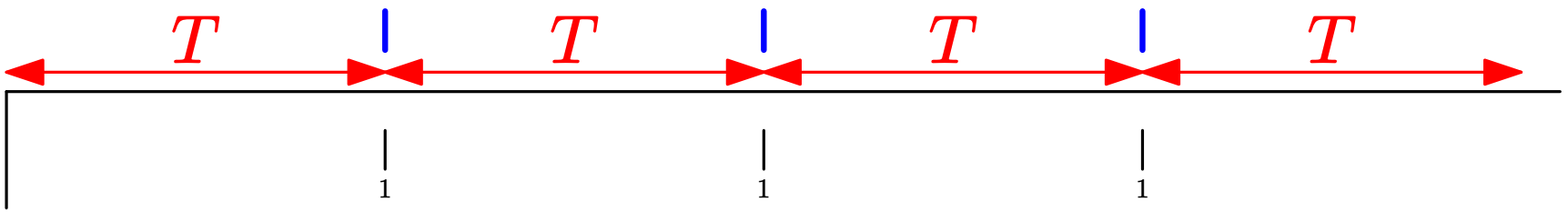


$1 \mid r_j \mid L_{\max}$ の強 NP 困難性 (2)

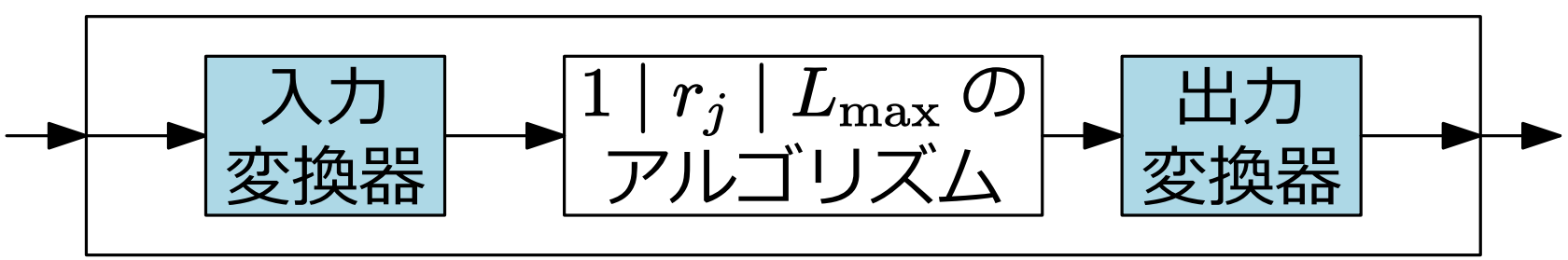
証明：3 分割問題を帰着する

入力の変換：





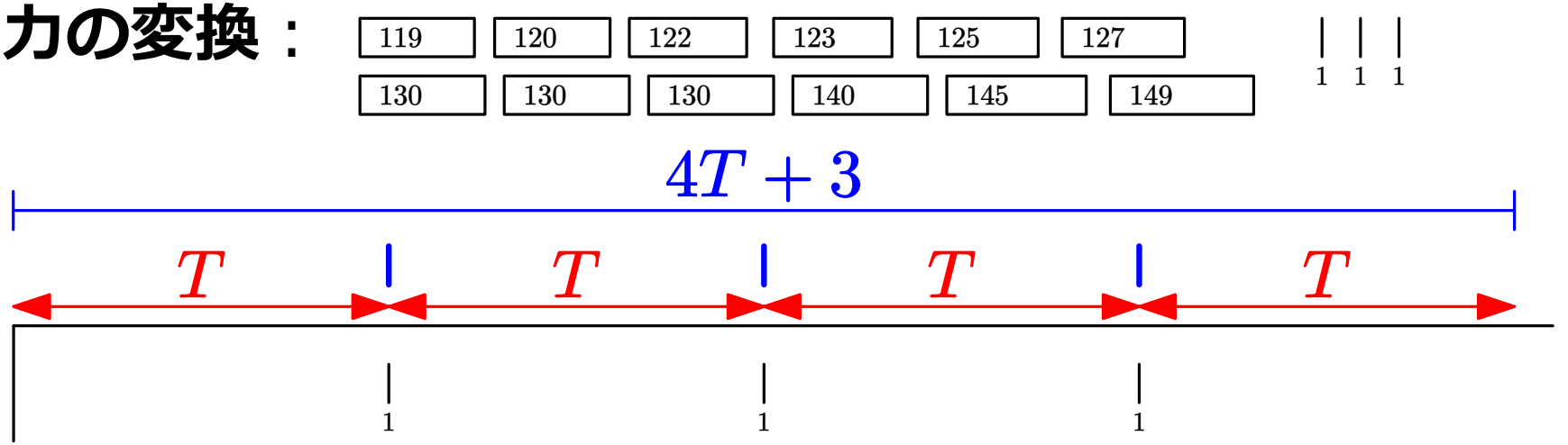
3 分割問題を解くアルゴリズム



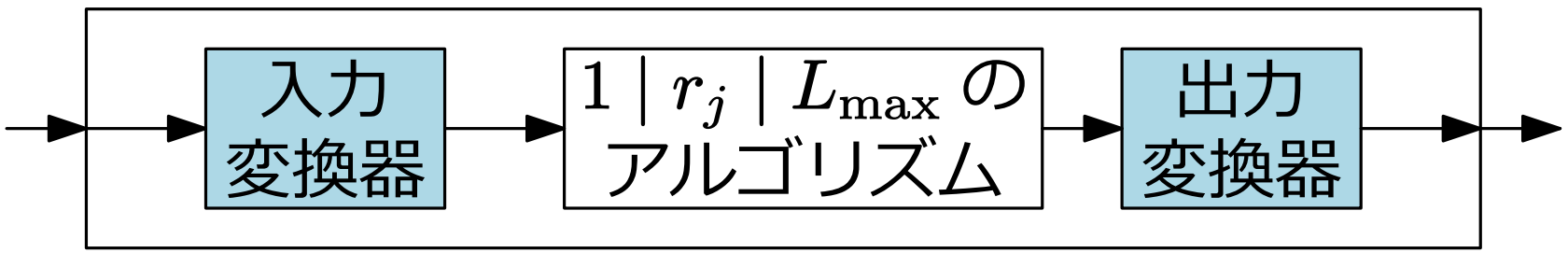
$1 \mid r_j \mid L_{\max}$ の強 NP 困難性 (2)

証明：3 分割問題を帰着する

入力の変換：



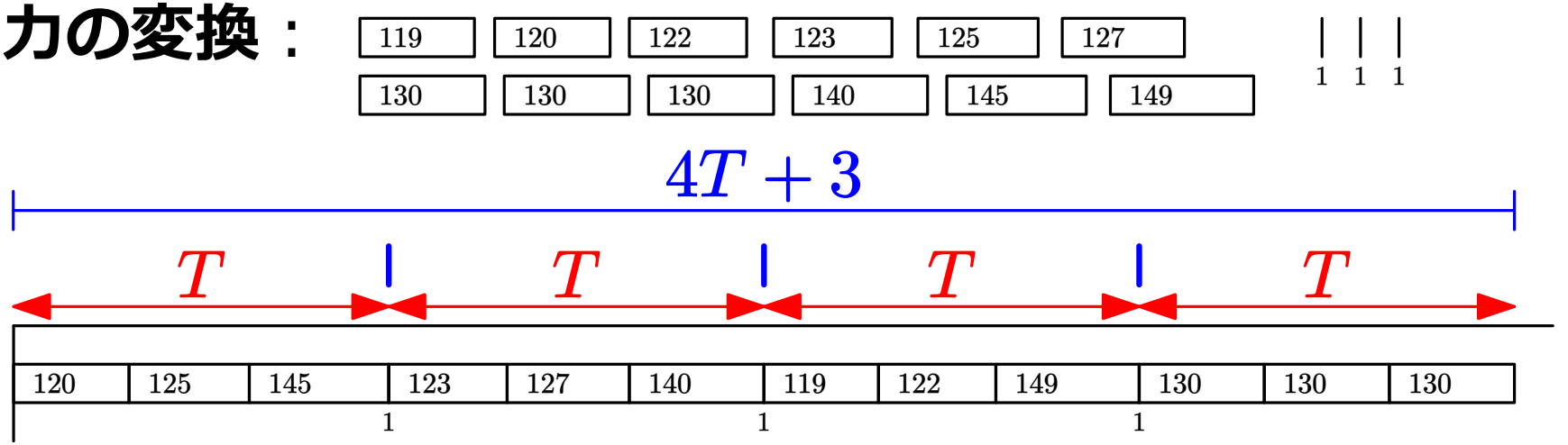
3 分割問題を解くアルゴリズム



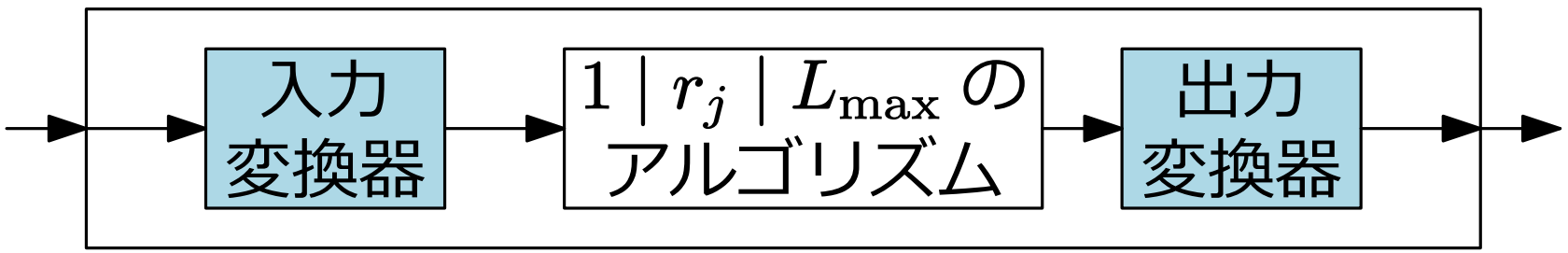
$1 \mid r_j \mid L_{\max}$ の強 NP 困難性 (2)

証明：3 分割問題を帰着する

入力の変換：



3 分割問題を解くアルゴリズム



$1 \mid r_j \mid L_{\max}$ の強 NP 困難性 (3)

証明：3 分割問題を帰着する

入力の変換：

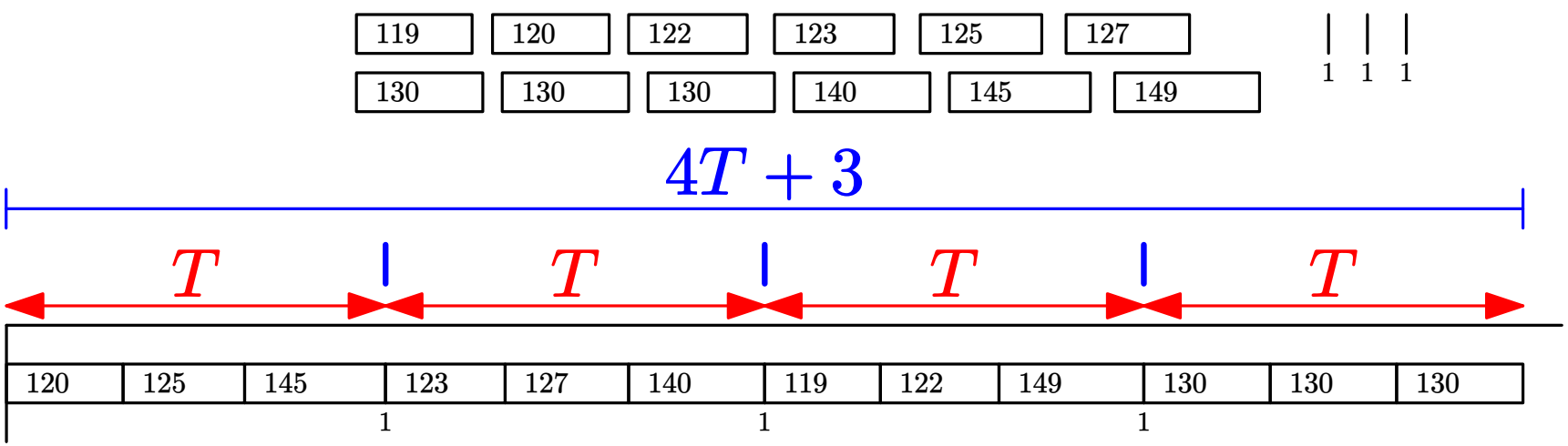
$3n$ 個の整数 $a_1, a_2, \ldots, a_{3n}$ に対して，(ここで，$T = \frac{1}{n}\sum_{j=1}^{3n} a_j$)

$3n$ 個のジョブ $J_1, J_2, \ldots, J_{3n}$ を作って，

$p_j = a_j$，$r_j = 0$，$d_j = n(T+1) - 1$ とする ($j \in [3n]$)

他に，$n-1$ 個のジョブ $J_{3n+1}, J_{3n+2}, \ldots, J_{3n+(n-1)}$ を作って，

$p_{3n+j} = 1$，$r_{3n+j} = j(T+1) - 1$，$d_{3n+j} = j(T+1)$ とする ($j \in [n-1]$)



$1 \mid r_j \mid L_{\max}$ の強 NP 困難性 (4)

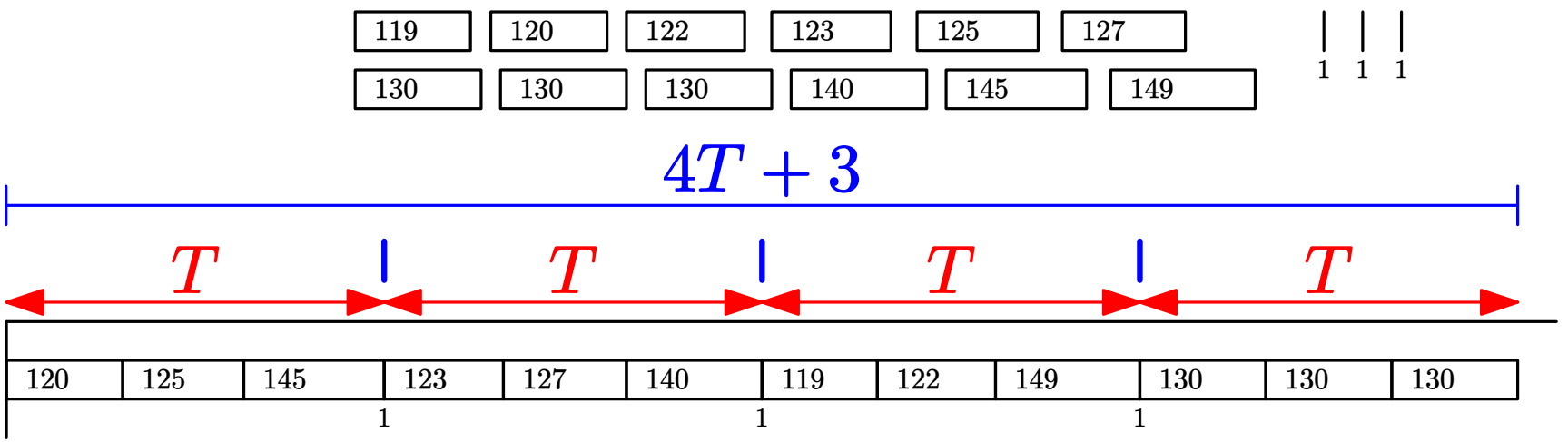
証明：3 分割問題を帰着する

出力の変換： $1 \mid r_j \mid L_{\max}$ のインスタンスとして

最適値 ≤ 0 $\Rightarrow$ 3 分割問題の答えは「できる」

最適値 > 0 $\Rightarrow$ 3 分割問題の答えは「できない」

(本来は，この変換が正しいことを証明しないといけない) □



紹介した NP 困難性証明

定理 (Lenstra, Rinnooy Kan, Brucker '77)

1. $P2 \,||\, C_{\max}$ は 弱 NP 困難
2. $P \,||\, C_{\max}$ は 強 NP 困難
3. $1 \,|\, r_j \,|\, L_{\max}$ は 強 NP 困難

考え方

弱 NP 困難性の証明 $\rightsquigarrow$ 分割問題を帰着

$\rightsquigarrow$ 容量 $\frac{1}{2}\sum_{i=1}^{n} a_i$ の箱を 2 つ作る

強 NP 困難性の証明 $\rightsquigarrow$ 3 分割問題を帰着

$\rightsquigarrow$ 容量 $\frac{1}{n}\sum_{i=1}^{3n} a_i$ の箱を n 個作る

まとめと次回の予告

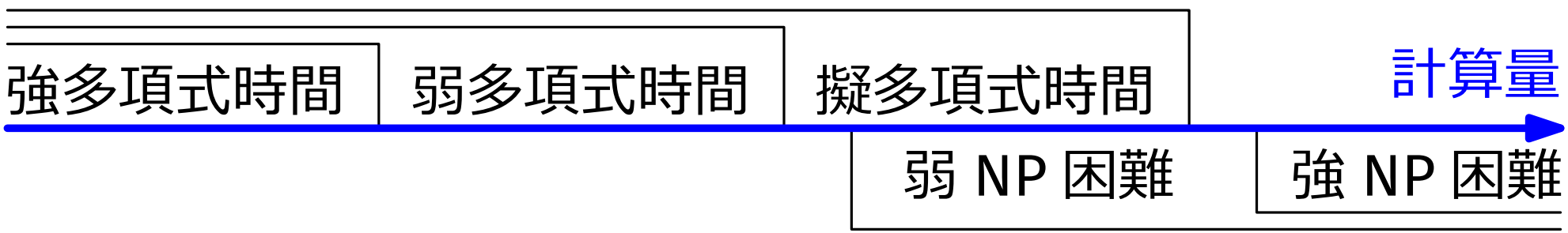
今日の内容・目標

アルゴリズムの計算量に関する基礎概念を身に付ける

- 多項式時間アルゴリズム，擬多項式時間アルゴリズム
- 弱 NP 困難性，強 NP 困難性

NP 困難性の証明をできるようになる

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



次回の予告

ジョブ・スケジューリング問題の間の計算複雑性の関係

付録

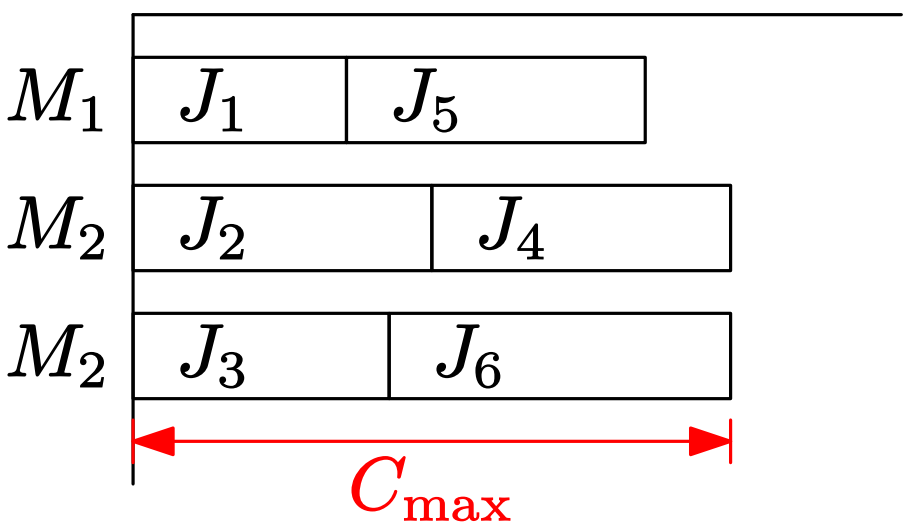
1. $Pm \mid\mid C_{\max}$ の弱 NP 困難性

定理

$m \geq 2$ は整数
- $Pm \mid\mid C_{\max}$ は 弱 NP 困難

機械の環境
- 同一並列機械，機械数 $= m$ (固定)

最適化する目的
- 最大完了時刻の最小化

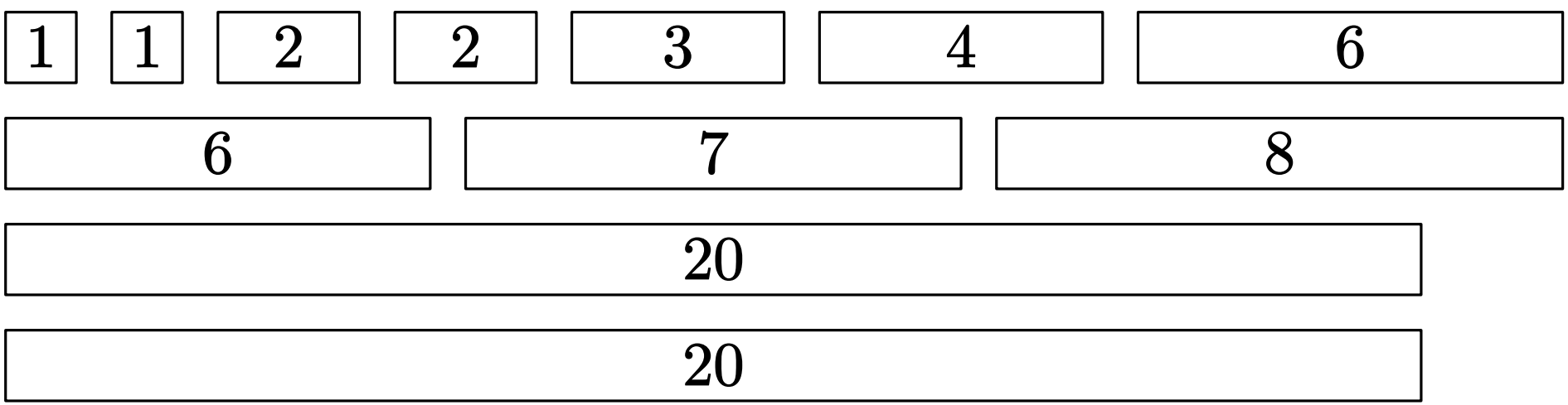


$Pm \mid\mid C_{\max}$ の弱 NP 困難性 (1)

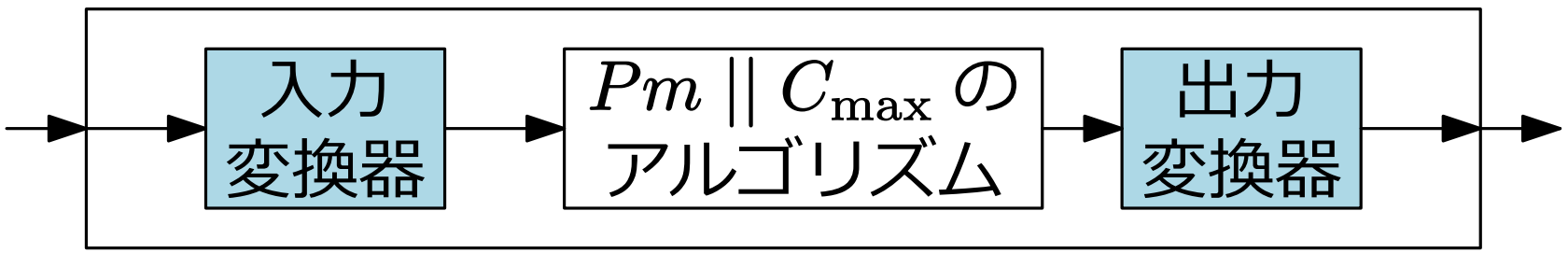
証明：分割問題を帰着する

入力の変換： $1, 1, 2, 2, 3, 4, 6, 6, 7, 8$

↓ ($m = 4$ のとき)



分割問題を解くアルゴリズム



$Pm \mid\mid C_{\max}$ の弱 NP 困難性 (1)

証明：分割問題を帰着する

入力の変換：n 個の整数 $a_1, a_2, \ldots, a_n$ に対して，
n 個のジョブ $J_1, J_2, \ldots, J_n$ を作って，
$p_j = a_j$ $(j \in [n])$ とする

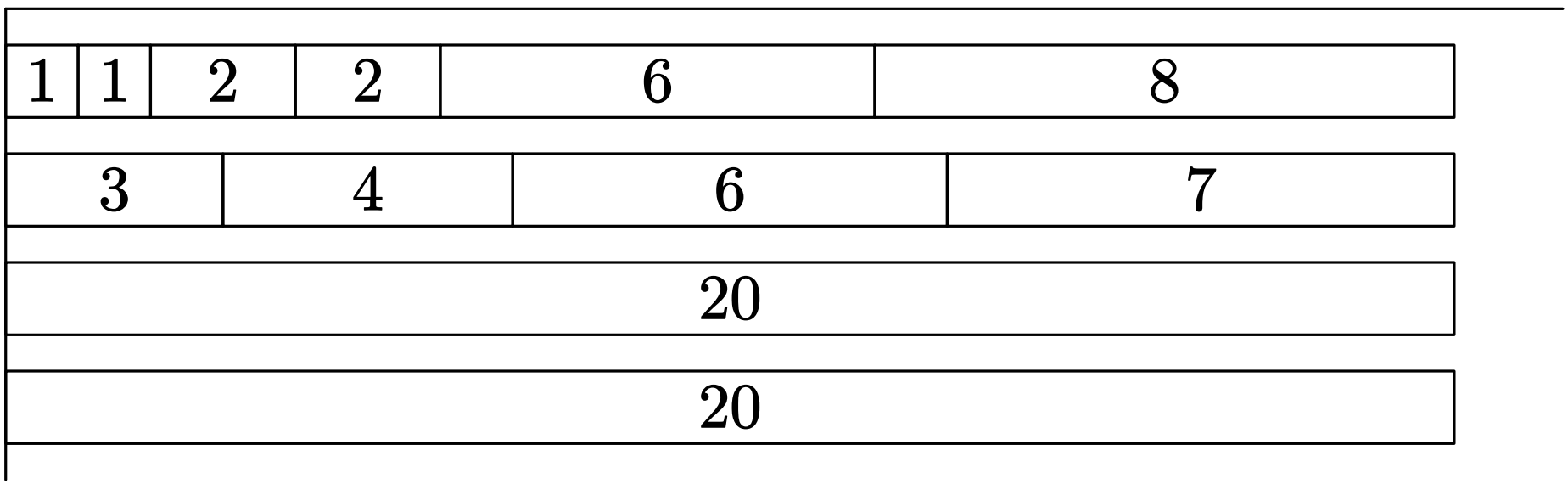
他に，$m-2$ 個のジョブ $J_{n+1}, \ldots, J_{n+m-2}$ を作って，

$$p_{n+j} = \frac{1}{2}\sum_{i=1}^{n} a_i \ (j \in [m-2]) \text{ とする}$$

分割問題を解くアルゴリズム

$Pm \mid\mid C_{\max}$ の弱 NP 困難性 (2)

証明：分割問題を帰着する



分割問題を解くアルゴリズム

$Pm \mid\mid C_{\max}$ の弱 NP 困難性 (3)

証明：分割問題を帰着する

出力の変換： $Pm \mid\mid C_{\max}$ のインスタンスとして

最適値 $= \frac{1}{2}\sum_{i=1}^{n} a_i$ $\Rightarrow$ 分割問題の答えは「できる」

最適値 $\neq \frac{1}{2}\sum_{i=1}^{n} a_i$ $\Rightarrow$ 分割問題の答えは「できない」

(本来は，この変換が正しいことを証明しないといけない) □

分割問題を解くアルゴリズム