並列機械モデルにおける
最大待ち時間最小化問題の計算論的分析

天本 祐希

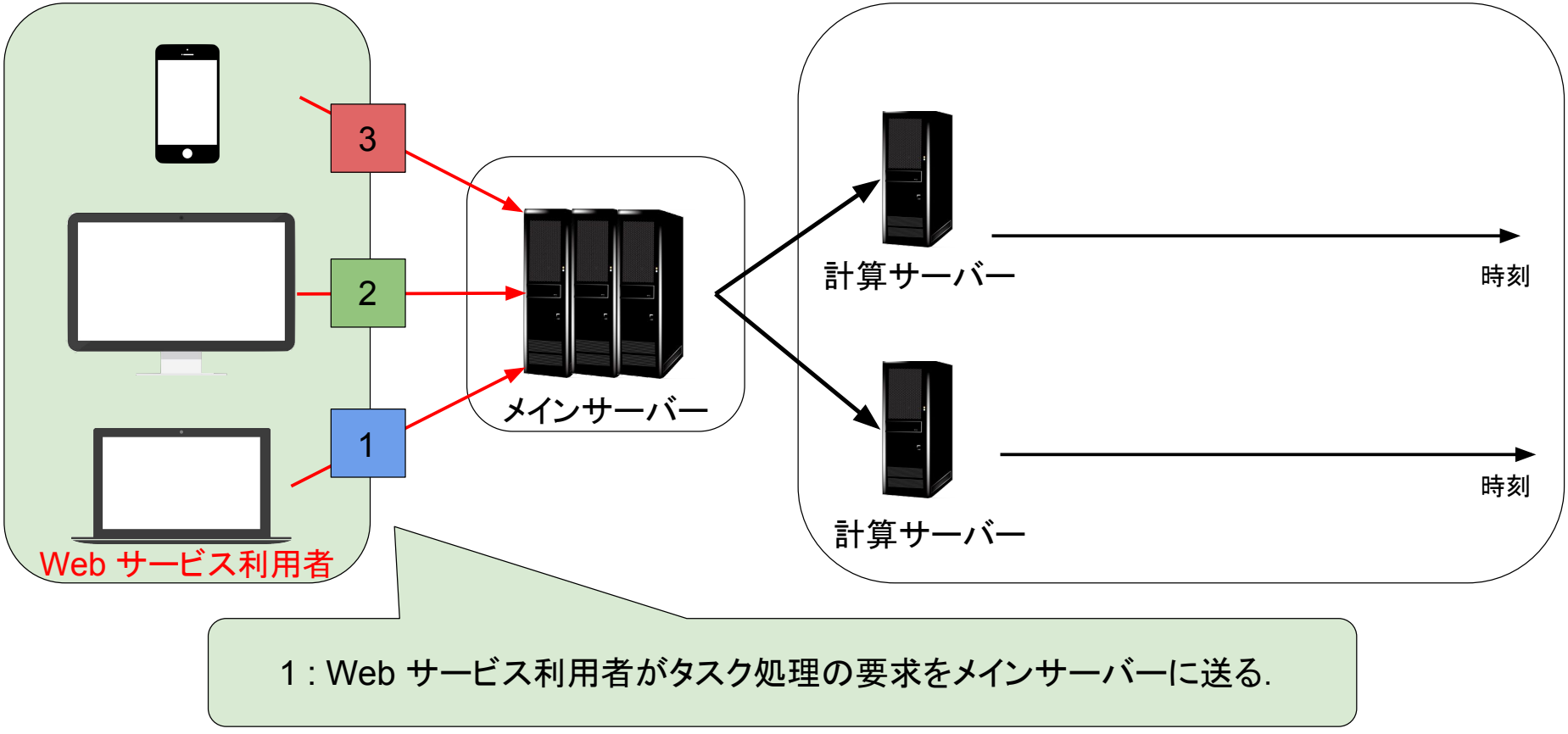
宋研究室

January 25, 2018

目次

目次

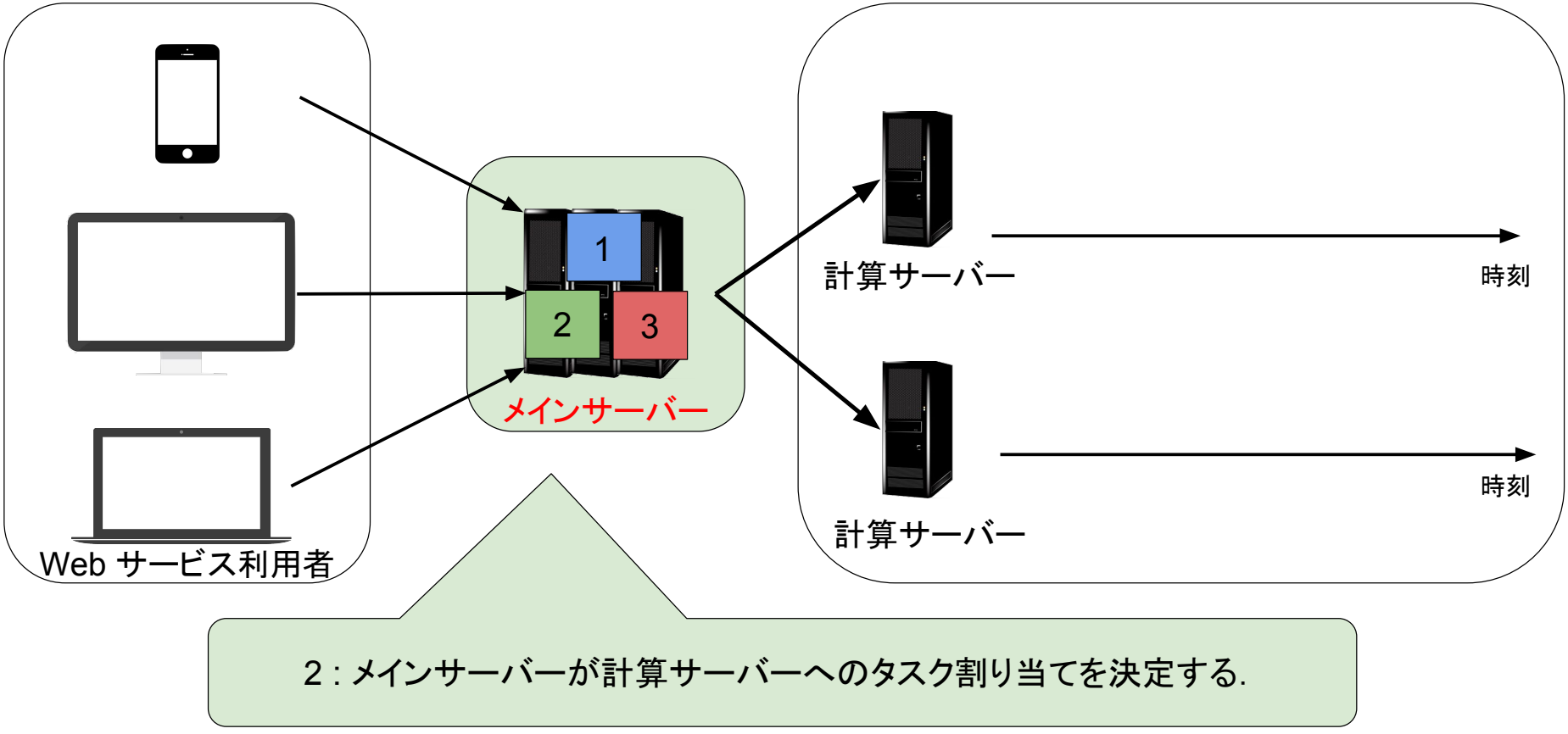
研究背景：Web サービスにおける処理の流れと問題



問題

Web サービスを運用している会社では，運用しているサービスの応答が遅いと，顧客離れやクレームの被害を受けることがある．そのため，計算サーバーの応答の早さは重要である．

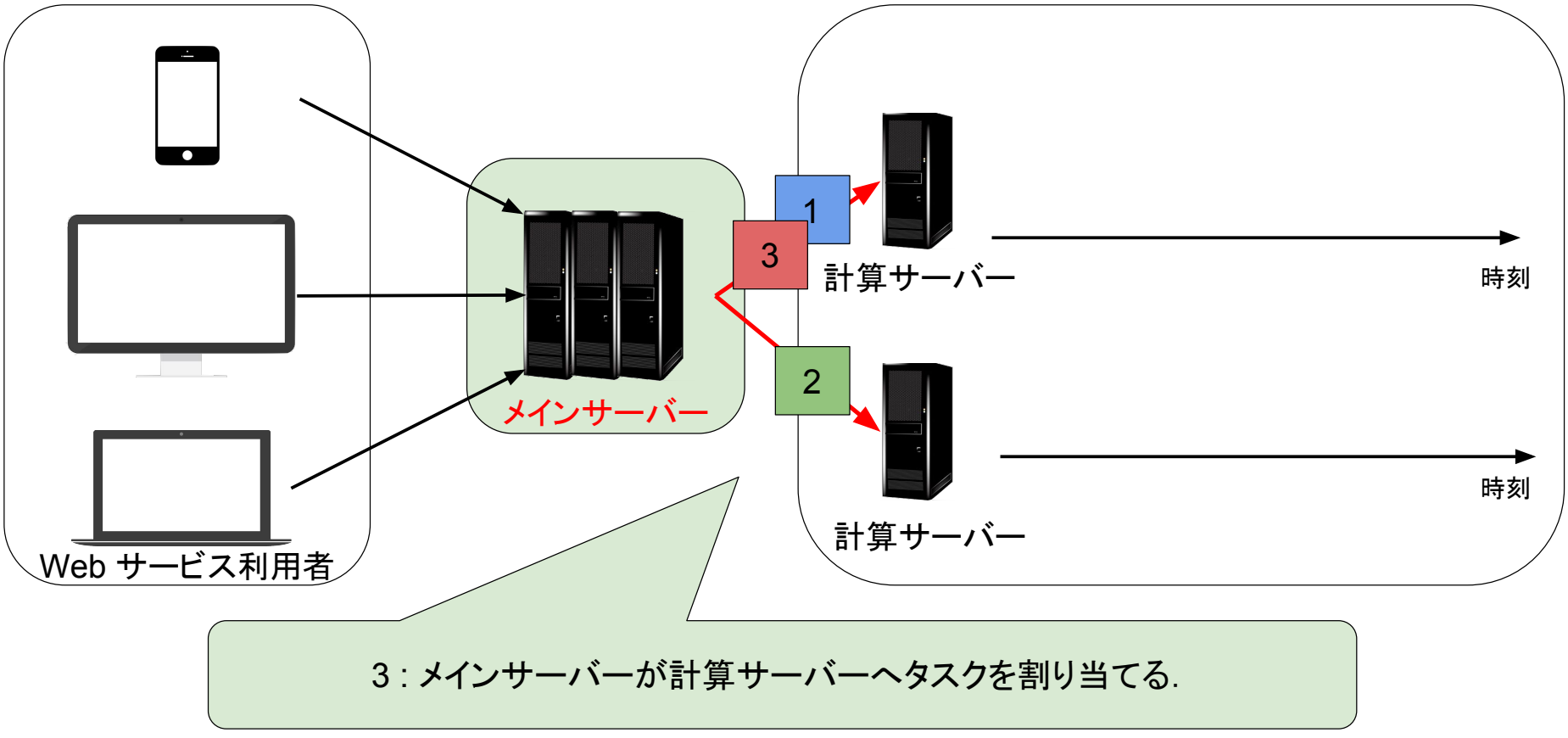
研究背景：Web サービスにおける処理の流れと問題



問題

Web サービスを運用している会社では，運用しているサービスの応答が遅いと，顧客離れやクレームの被害を受けることがある．そのため，計算サーバーの応答の早さは重要である．

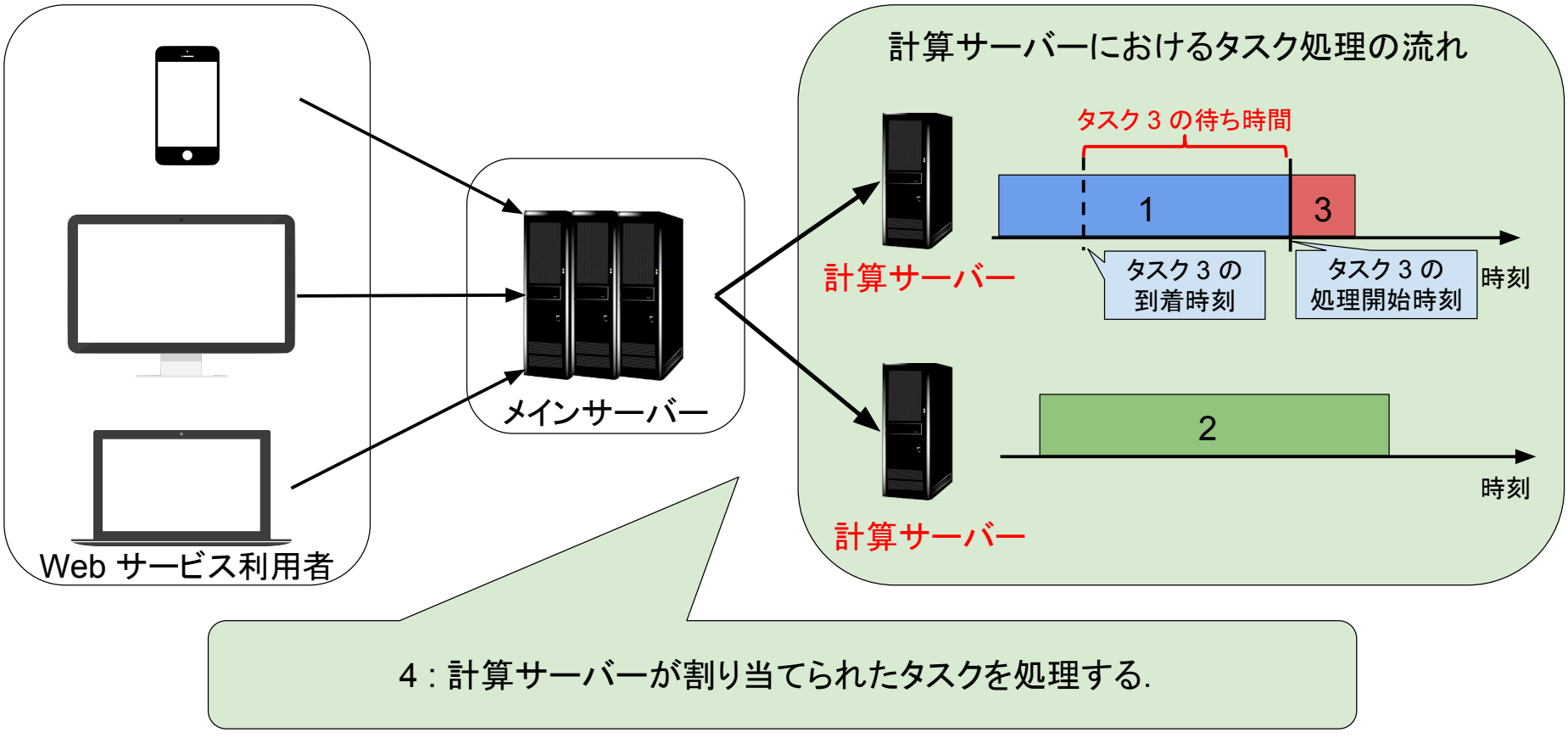
研究背景：Web サービスにおける処理の流れと問題



問題

Web サービスを運用している会社では，運用しているサービスの応答が遅いと，顧客離れやクレームの被害を受けることがある．そのため，計算サーバーの応答の早さは重要である．

研究背景：Web サービスにおける処理の流れと問題



問題

Web サービスを運用している会社では，運用しているサービスの応答が遅いと，顧客離れやクレームの被害を受けることがある．そのため，計算サーバーの応答の早さは重要である．

研究背景：スケジューリング問題との対応

Web サービスの問題を解決する方法の 1 つとして，タスクが到着してから，処理されるまでの時間を短くする方法が挙げられる．計算サーバーにおける各時間は以下に対応する．つまり，計算サーバーへのタスク割り当ては，スケジューリング問題として捉えることができる．

- タスクの到着時刻とは，処理開始可能時刻に対応する．
- タスクが到着してから，タスクの処理が開始されるまでの時間とは，待ち時間に対応する．

最大待ち時間最小化問題（SWT）は，処理開始可能時刻を制約とし，最大の待ち時間の最小化を目的とするスケジューリング問題として捉えることができる．

研究目的

機械モデルおよび機械数に着目し，問題の難しさに影響を与える特徴を明らかにする．また，SWT の計算複雑さに基づいて，解法の提案を行う．

目次

定式化：最大待ち時間最小化問題

機械モデル

ジョブを処理する機械の台数について

- 1 つの機械で処理を行う単一機械モデル，
- 複数の機械で処理を行う並列機械モデル，

に分類される．本研究では，並列機械モデルの中で，すべての機械の性能が等しい同一並列機械モデルとジョブと機械の組み合わせによって処理時間が異なる無関連並列機械モデルを扱う．

以下，無関連並列機械モデルにおける SWT を定式化する．

入力：$\mathcal{I} = (\mathcal{J}, \mathcal{M}, r, p)$

- ジョブの集合 $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$
- 無関連機械の集合 $\mathcal{M} = \{M_1, M_2, \ldots, M_m\}$
- ジョブの処理開始可能時刻を返す関数 $r : \mathcal{J} \to \mathbb{N}$
- ジョブの処理時間を返す関数 $p : \mathcal{J} \times \mathcal{M} \to \mathbb{N}$

定式化：最大待ち時間最小化問題

解：スケジュール (A, s)

以下の条件を満たす $A : \mathcal{J} \to \mathcal{M}$ と $s : \mathcal{J} \to \mathbb{N}$ の 2 項組 (A, s).

- $\forall J \in \mathcal{J}\left[s(J) \geq r(J)\right]$
  - 各ジョブは処理開始可能時刻以降に処理を開始する.
- $\forall J, J' \in \mathcal{J}\ \Big[\left[J \neq J' \wedge A(J) = A(J')\right] \Rightarrow$
  $[s(J), s(J) + p(J, A(J))) \cap [s(J'), s(J') + p(J', A(J'))) = \emptyset\Big]$
  - 各機械は同時に複数のジョブを処理しない.
  - 各ジョブの処理を開始すると，完了するまで中断しない.

目的関数：最大待ち時間 $W_{\max}$

- $\varphi(A, s) = \max_{J \in \mathcal{J}} \{s(J) - r(J)\}$
  - すべてのジョブの中で，ジョブの処理開始可能時刻と処理開始時刻の差が最大のもの.

- 最大待ち時間 $W_{\max}$ の最小化を目的とするスケジューリング問題

目次

定式化：最大待ち時間最小化問題（決定問題）

ここで，SWT を決定問題として定義する．

決定問題

決定問題は，インスタンスと問題によって定義され，判定として Yes または No のいずれかを持つ．

SWT の決定問題

インスタンス： SWT のインスタンス $\mathcal{I}$ と待ち時間 w の 2 項組 $(\mathcal{I}, w)$

問題： SWT の実行可能なスケジュール (A, s) のうち以下の条件を満たすスケジュール (A, s) が存在するか？

- $\max\{s(J) - r(J) \mid J \in \mathcal{J}\} \leq w$
  - ジョブの処理開始可能時刻からその処理を開始するまでの待ち時間は w 以下．

目次

定式化：3-SATISFIABILITY

3-SATISFIABILITY (3-SAT) は決定問題の 1 つで，この問題は NP 完全であることが知られている．

3-SATISFIABILITY

インスタンス： (X, H)

- ブール型の変数集合 $X = \{x_1, x_2, \ldots, x_n\}$
  - リテラルの集合 $L_X = X \cup \{\bar{x} \mid x \in X\}$
    - X と X の各要素の論理否定をとった変数の集合
- 3 つのリテラルからなる節 h の集合 H
  - $H \subseteq 2^{L_X}$ s.t. $\forall h \in H\big[|h| = 3\big]$

問題： 以下を満たす真理値割り当て $f : X \to \{0, 1\}$ が存在するか？

$$\bigwedge_{h \in H} \left(\bigvee_{x \in h} f(x) \vee \bigvee_{\bar{x} \in h} \neg f(x) \right) = 1$$

例えば，$X = \{x_1, x_2, x_3\}$，$H = \big\{\{x_1, x_2, \bar{x}_3\}, \{\bar{x}_1, \bar{x}_2, \bar{x}_3\}\big\}$ のとき，$f(x_1) = 1, f(x_2) = 0, f(x_3) = 0$ が存在するので，判定は Yes となる．

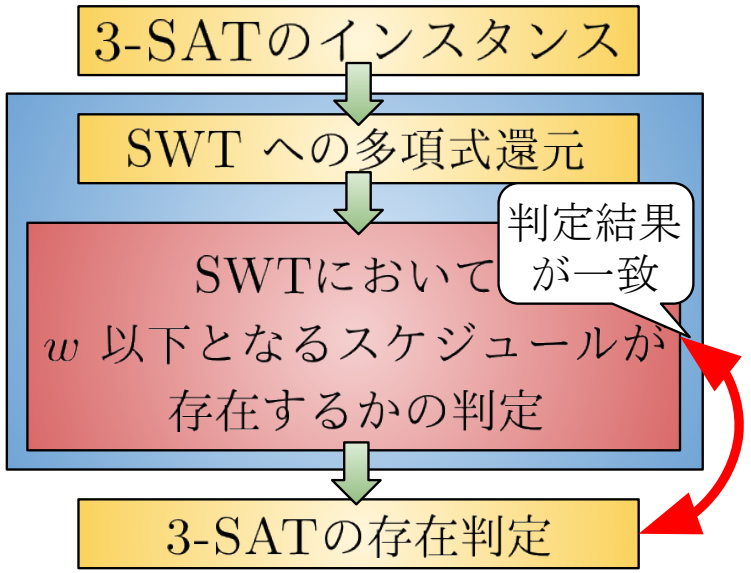
目次

多項式還元とは：還元の流れ

NP 完全

ある問題が NP 完全であることを示すには，以下が成立することを示す．

- 問題が NP に属す．
- 問題が NP 困難である．



本研究では以下が成立することを示す．

1. スケジュールにおける最大待ち時間が w 以下であるかの判定が多項式時間で可能である．
2. 3-SAT のインスタンスから SWT のインスタンスが多項式時間で構成可能であり，3-SAT の判定結果と構成したインスタンスに対する w 以下となるスケジュールの存在判定結果が一致する．

目次

問題の分析：既存のスケジューリング問題との対応

既存のスケジューリング問題との共通部分

- $w = 0$ のとき，処理開始可能時刻ちょうどで処理を開始しなければならない．
  - JIT ジョブスケジューリング問題と対応する．
- SWT において，納期 $=$ (処理開始可能時刻 $+$ 処理時間 $+ w$ ）と定義できる．
  - 処理開始可能時刻と納期付きスケジューリング問題と対応する．

既存のスケジューリング問題との差分

- SWT は，JIT ジョブスケジューリング問題の拡張問題として捉えることができる．
- SWT は，処理開始可能時刻と納期付きスケジューリング問題の部分問題として捉えることができる．

- JIT ジョブスケジューリング問題 $\subseteq$ SWT
- SWT $\subseteq$ 処理開始可能時刻と納期付きスケジューリング問題

問題の分析：既存のスケジューリング問題の計算複雑さ

本研究では，JIT ジョブスケジューリング問題の 1 つである, JIT ジョブ荷重和最大化問題（SJIT）と 処理開始可能時刻，納期付きスケジューリング問題の 1 つである処理開始可能時刻付き最大遅れ時間最小化問題（SRTD）を SWT と対応付けた.

既存のスケジューリング問題の計算複雑さ

- SJIT は，3-SAT からの還元により，無関連並列機械モデルにおいて，機械数が入力の一部の場合，強 NP 困難である.
- SRTD は，3-PARTITION からの還元により，単一機械モデルにおいて，強 NP 困難である.

目次

目次

研究成果：問題の計算複雑さ

成果 1

無関連並列機械モデルにおいて，機械数が入力の一部の場合，SWT が NP 完全であることを明らかにした．

インスタンスの変換

3-SAT のインスタンス を $X = \{x_1, \ldots, x_n\}$ と $H = \{h_1, \ldots, h_\lambda\}$ からなる 2 項組 (X, H) とする．(X, H) に基づき表記を導入する．

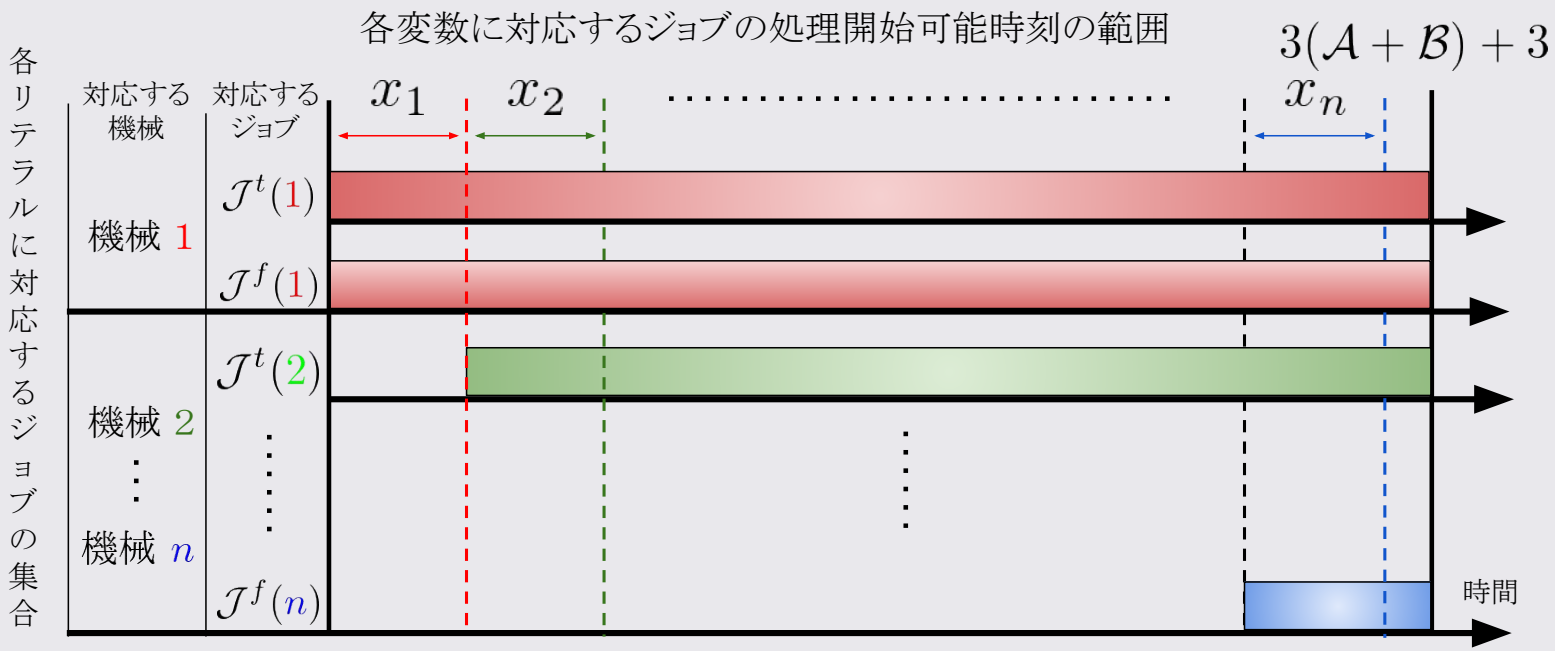
- $\forall i \in \{1, \ldots, n\},\ \alpha_i = |\{h \in H \mid x_i \in h\}|$
  - H において，x_i が現れる回数を表す自然数．
- $\forall i \in \{1, \ldots, n\},\ \beta_i = |\{h \in H \mid \bar{x}_i \in h\}|$
  - H において，$\bar{x}_i$ が現れる回数を表す自然数．
- $\mathcal{A} = \sum_{i \in \{1,\ldots,n\}} \alpha_i$，$\mathcal{B} = \sum_{i \in \{1,\ldots,n\}} \beta_i$

$2(\mathcal{A} + \mathcal{B}) + \lambda$ 個のジョブ $\mathcal{J}$ を $n + \lambda$ 台の無関連機械 $\mathcal{M}$ で処理する．
ただし，$\mathcal{J} = \{\mathcal{J}^t, \mathcal{J}^f, \mathcal{J}^d\}$ s.t. $|\mathcal{J}^t| = |\mathcal{J}^f| = \mathcal{A} + \mathcal{B}$，$|\mathcal{J}^d| = \lambda$ とする．

研究成果：問題の計算複雑さ

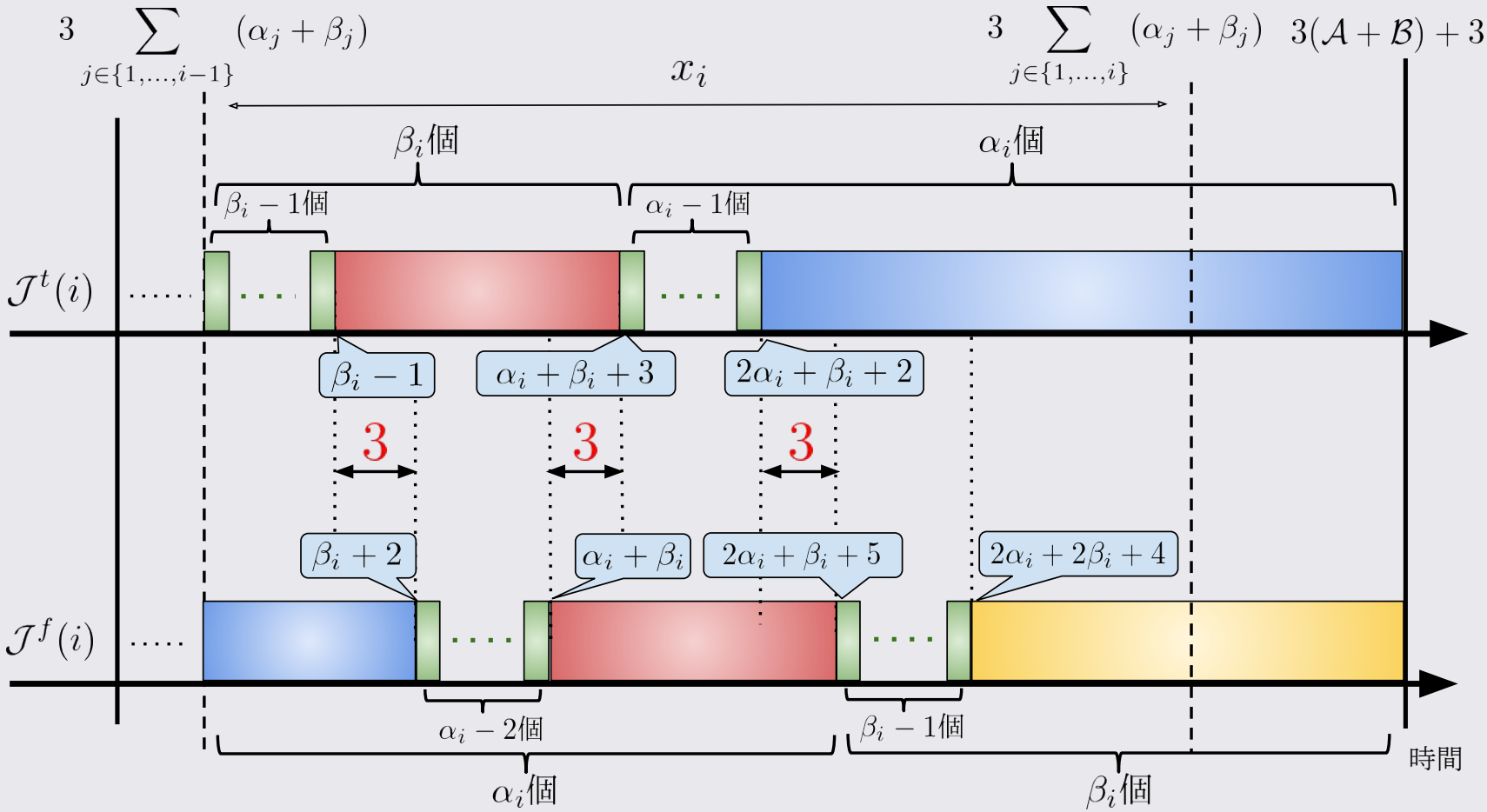
各 $i \in \{1, \ldots, n\}$ において，$\alpha_i \geq 3$，$\beta_i \geq 3$ のとき，次のように SWT のインスタンスを設定する．

各リテラルに対応するジョブの処理開始可能時刻の範囲



- 各リテラルに対応するジョブの集合における各ジョブの処理開始可能時刻を，対応する機械ごとに範囲をずらして設定している．
- 各リテラルに対応するジョブの集合の中で，処理開始可能時刻が一番遅いジョブの完了時間は，$3(\mathcal{A}+\mathcal{B})+3$ と設定している．

機械に対応するジョブを割り当てたときのスケジュール



- x_i と $\bar{x}_i$ に対応するジョブの集合 $\mathcal{J}^t(i)$ と $\mathcal{J}^f(i)$ におけるジョブ間の処理開始可能時刻の差を 3 に設定している．

研究成果：問題の計算複雑さ

3-SAT の判定結果と，SWT における最大待ち時間が 2 以下となるスケジュールの存在判定結果が一致することを示した.

還元のポイント

- 3-SAT の判定結果が Yes のとき，SWT における最大待ち時間は 2 以下になる .
- 3-SAT の判定結果が No のとき，SWT における最大待ち時間は 3 以上になる .

無関連並列機械モデルにおいて機械数が入力の一部の場合，SWT が NP 困難である限り，1.5 未満の定数近似アルゴリズムは存在しない.

計算理論的成果のまとめ

SWT が NP 困難であることを明らかにした.

- NP 困難な問題に対する効率的な解法は発見されていない.

上記の成果に基づき，SWT に対する解法の提案とその分析を行う.

目次

研究成果：ヒューリスティックの提案

成果 2

同一並列機械モデルにおける SWT に対して，ヒューリスティックの開発と解法に対する実験的評価を行った．

貪欲アルゴリズムに基づいた解法

入力：$I = (\mathcal{J}, \mathcal{M}, r, p, C)$

出力：スケジュールの集合 $\mathcal{S}$．

Step 1. $\mathcal{J}$ を処理開始可能時刻の昇順でソートする．

Step 2. 各 $1 \leq i \leq n$ における J_i について以下の処理を繰り返す．

Step 2.1. 最小完了時刻を持つスケジュール集合の要素 $S_{M_a} \in \left\{ \arg\min_{M \in \mathcal{M}} C(S_M) \right\}$ を 1 つ求める．ここで，$S_{M_a} := S_{M_a} \cup J_i$ とする．

Step 3. $\mathcal{S} = \{S_M \mid M \in \mathcal{M}\}$ として，$\mathcal{S}$ を出力する．

研究成果：ヒューリスティックの精度について

ヒューリスティックから得られた解の最大待ち時間と厳密解法から得られた解の最大待ち時間の比に着目して比較を行った.

例：2 台の同一機械と以下のインスタンスにおけるスケジュール

最適なスケジュール $S_1, S_2 \in \mathcal{S}$ は,

- $S_1 : (J_1, J_3, J_5)$, $S_2 : (J_2, J_4, J_6)$.
- $\mathcal{S}$ における最大待ち時間は 0.

最適でないスケジュール $S'_1, S'_2 \in \mathcal{S}'$ は,

- $S'_1 : (J_1, J_3, J_6)$, $S'_2 : (J_2, J_4, J_5)$.
- $\mathcal{S}'$ における最大待ち時間は 3.

$\mathcal{J}$	$r(J)$	$p(J)$
J_1	2	8
J_2	3	7
J_3	11	11
J_4	16	10
J_5	23	10
J_6	26	11

$$\frac{\text{ヒューリスティックから得られた解の最大待ち時間}}{\text{厳密解法から得られた解の最大待ち時間}} = \infty$$

本研究で提案したヒューリスティックは，上記の比が定数となることが保証されている.

研究成果：ヒューリスティックの実験的評価

- 実験回数：約 300 万回
- 縦軸：$\dfrac{\text{ヒューリスティックから得られた解の最大待ち時間}}{\text{厳密解法から得られた解の最大待ち時間}}$
- 横軸：

研究成果：厳密解法の提案

成果 3

同一並列機械モデルにおける SWT に対して，厳密解法の開発と計算時間の分析を行った．

分割生成アルゴリズムの改良

- 分割の要素数 = 機械数 となる分割のみ生成するように改良．
  - 考慮する分割の数を減らすことができる．

分枝限定法アルゴリズムの改良

- 部分問題に対する多項式アルゴリズムの概念を導入した．
  - 列挙する実行可能解を減らすことができる．

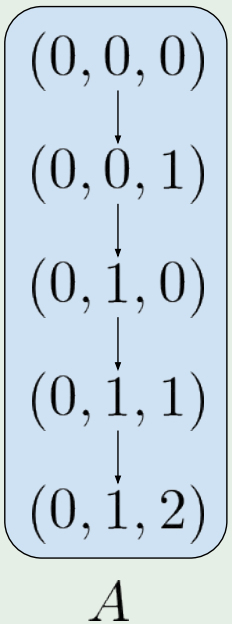
上記以外の改良

- 各機械におけるスケジュールのコストの降順で，探索を始める．
  - 各分割における探索初期段階で，探索の中断を判定できる．

研究成果：分割生成アルゴリズムの改良

同一並列機械モデルにおいて，ジョブ J_1 を機械 1 に割り当てるのと，機械 2 に割り当てることは同じである．したがって，ジョブをどの機械に割り当てるかではなく，どのジョブと同じ機械に割り当てるかを考える必要がある．つまり，同一並列機械モデルにおけるジョブの機械への割り当ては，ジョブの分割として捉えることができる．

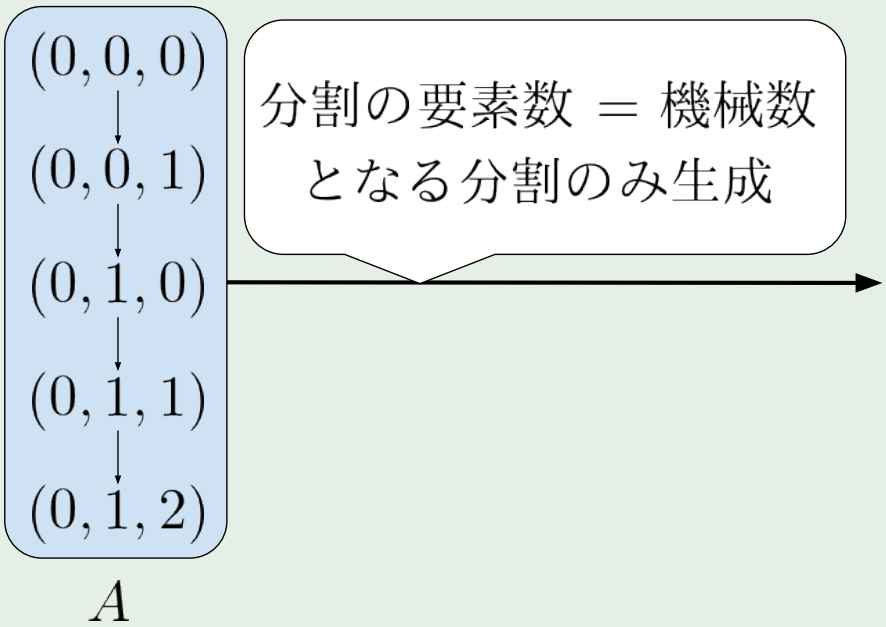
例：ジョブ数 3，同一機械数 2 における分割



研究成果：分割生成アルゴリズムの改良

同一並列機械モデルにおいて，ジョブ J_1 を機械 1 に割り当てるのと，機械 2 に割り当てることは同じである．したがって，ジョブをどの機械に割り当てるかではなく，どのジョブと同じ機械に割り当てるかを考える必要がある．つまり，同一並列機械モデルにおけるジョブの機械への割り当ては，ジョブの分割として捉えることができる．

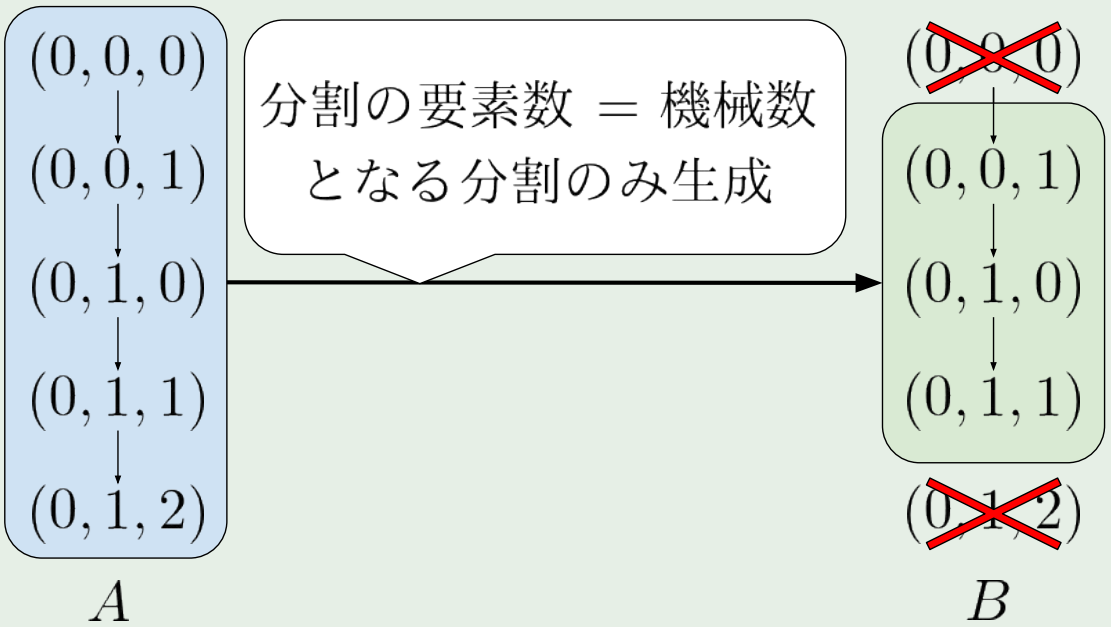
例：ジョブ数 3，同一機械数 2 における分割



研究成果：分割生成アルゴリズムの改良

同一並列機械モデルにおいて，ジョブ J_1 を機械 1 に割り当てるのと，機械 2 に割り当てることは同じである．したがって，ジョブをどの機械に割り当てるかではなく，どのジョブと同じ機械に割り当てるかを考える必要がある．つまり，同一並列機械モデルにおけるジョブの機械への割り当ては，ジョブの分割として捉えることができる．

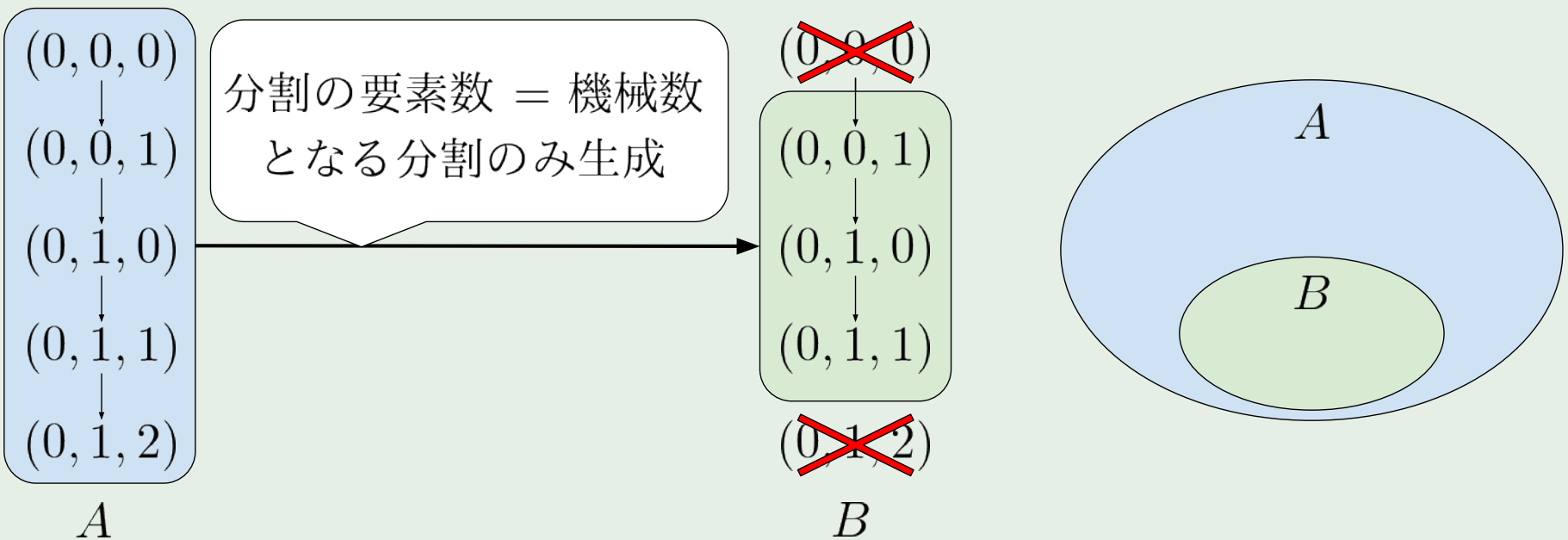
例：ジョブ数 3，同一機械数 2 における分割



研究成果：分割生成アルゴリズムの改良

同一並列機械モデルにおいて，ジョブ J_1 を機械 1 に割り当てるのと，機械 2 に割り当てることは同じである．したがって，ジョブをどの機械に割り当てるかではなく，どのジョブと同じ機械に割り当てるかを考える必要がある．つまり，同一並列機械モデルにおけるジョブの機械への割り当ては，ジョブの分割として捉えることができる．

例：ジョブ数 3，同一機械数 2 における分割



研究成果：その他の改良

分割生成アルゴリズムと分枝限定法アルゴリズムの改良下部分を用いて，さらに計算効率を向上させるために，以下の改良を加えた．

例：以下の分割のとき，順列を生成する機械順の違い

ジョブ数 5，同一機械数 3 において，分割が $(0, 1, 2, 1, 1)$ のとき，通常版と改良版における順列を生成する機械順は以下の通り．

通常版　機械 0 から順に処理を行う．$0 \to 1 \to 2$.

改良版　各機械に割り当てられたジョブ集合におけるコストの下限の降順で処理を行う．$0 \to 2 \to 1$.

上記改良の正当性

上記改良により，コストの下限が悪いスケジュールを生成した機械から順に，その機械におけるジョブ集合の順列を生成することで，目的関数に与える影響の大きいスケジュールを早い段階で生成できる．

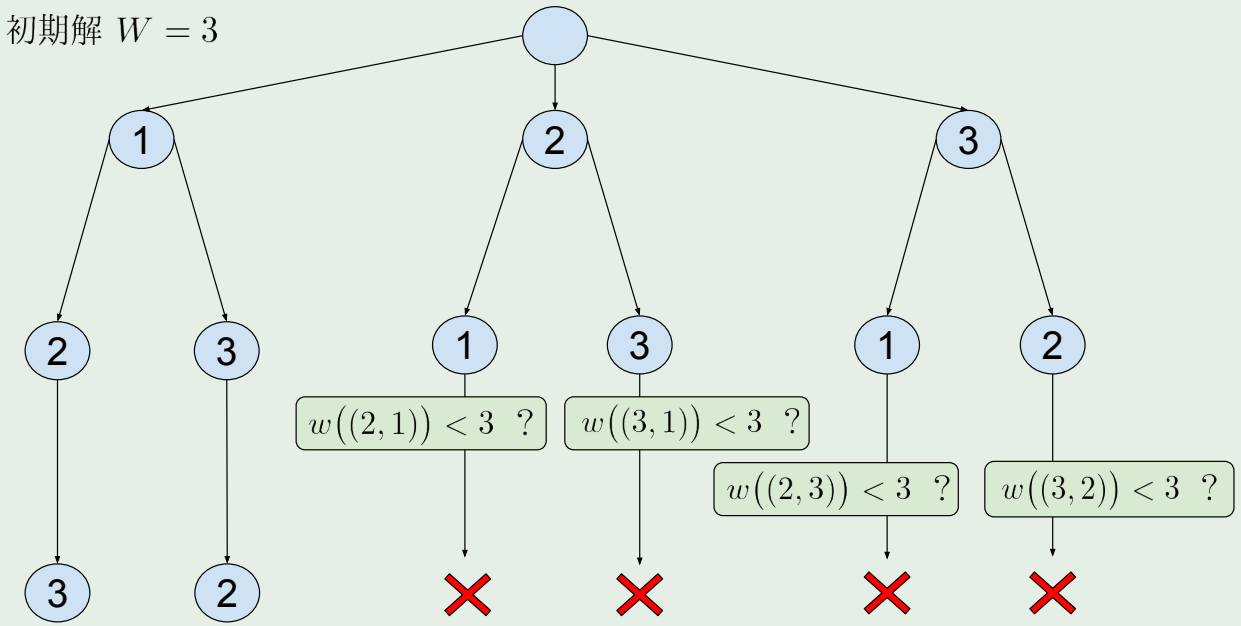
分割生成アルゴリズムの改良による計算時間の評価

分割生成アルゴリズムの改良前と，改良後の計算時間の比較を行った.

- 実験回数：約 $3,000$ 回
- 縦軸：計算時間
- 横軸：

分枝限定法アルゴリズムの改良

例：以下のインスタンスにおけるスケジュール生成

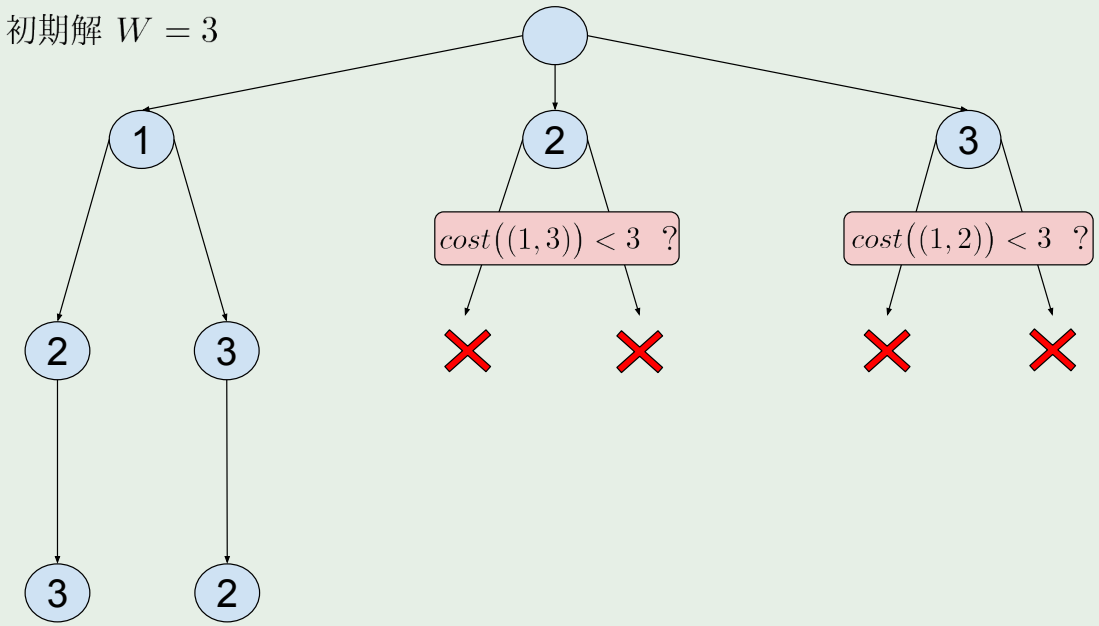


$\mathcal{J}$	$r(J)$	$p(J)$
J_1	2	4
J_2	5	7
J_3	10	15

- それまでに割り当てたジョブのスケジュールにおける最大待ち時間がそれまでの最良の解 W より良い値か？
- 割り当てていない残りのジョブを最適にスケジュールしたときの最大待ち時間がそれまでの最良の解 W より良い値か？
  - SWT は，処理時間が一定のとき，処理開始可能時刻の昇順で処理することで最適解が求まる．

分枝限定法アルゴリズムの改良

例：以下のインスタンスにおけるスケジュール生成



$\mathcal{J}$	$r(J)$	$p(J)$
J_1	2	4
J_2	5	7
J_3	10	15

- それまでに割り当てたジョブのスケジュールにおける最大待ち時間がそれまでの最良の解 W より良い値か？
- 割り当てていない残りのジョブを最適にスケジュールしたときの最大待ち時間がそれまでの最良の解 W より良い値か？
  - SWT は，処理時間が一定のとき，処理開始可能時刻の昇順で処理することで最適解が求まる．

分枝限定法アルゴリズムの改良による計算時間の評価

分枝限定法アルゴリズムの改良前と，改良後の計算時間の比較を行った.

- 実験回数：約 $3,000$ 回
- 縦軸：計算時間
- 横軸：

目次

まとめと今後の課題

まとめ

- 無関連並列機械モデルにおいて，機械数が入力の一部の場合，SWT の NP 完全性を明らかにした．
  - 1.5 未満の定数近似ができないことを明らかにした．
- SWT に対するヒューリスティックと厳密解法の開発した．
- 実験的評価を行い，対象とする環境におけるヒューリスティックの有効性，厳密解法の有用性を示した．

今後の課題

- 無関連並列機械モデル以外の機械モデルにおける SWT の計算複雑さを明らかにし，問題の難しさに影響を与える特徴を明らかにする．
- 3-SAT からの還元手法を工夫して，3-SAT の存在判定が Yes のとき，No のときの SWT の最大待ち時間の比を調整する．
  - 上記の比を大きくすることができれば，その比未満の定数近似アルゴリズムが存在しないことを保証できる．
- 上記の結果に基づき，近似アルゴリズムを開発する．

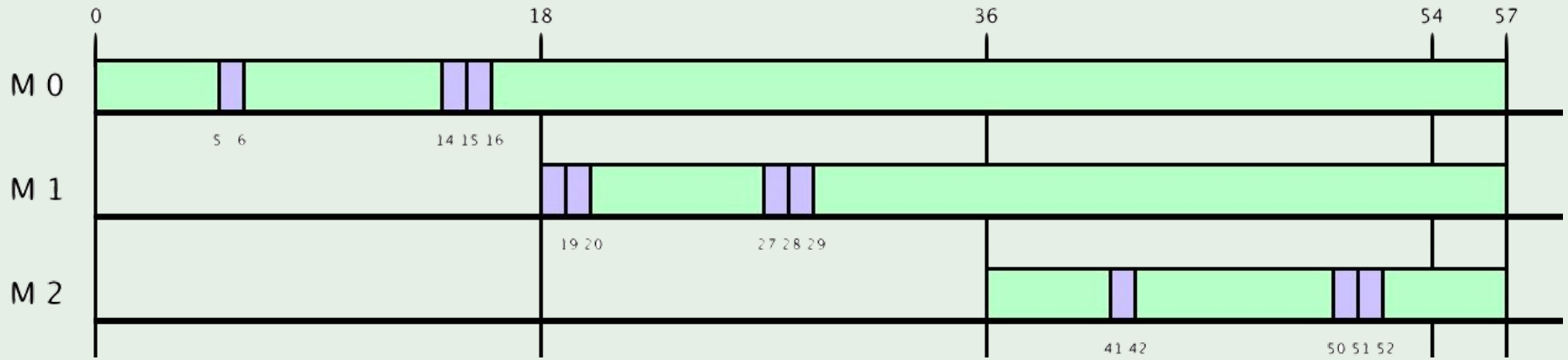
付録：多項式還元の例（3-SAT の判定結果が **Yes** のとき）

例：各変数に対応する機械におけるスケジュール

$X = \{x_1, x_2, x_3\}$, $H = \{\{x_1, x_2, \bar{x}_3\}, \{x_1, x_2, x_3\}, \{x_1, \bar{x}_2, \bar{x}_3\}, \{\bar{x}_1, x_2, x_3\}, \{\bar{x}_1, \bar{x}_2, x_3\}, \{\bar{x}_1, \bar{x}_2, \bar{x}_3\}\}$ の 2 項組 (X, H) が 3-SAT のインスタンスとして与えられたとき，前のスライドで説明した方法により，SWT のインスタンス $I_{(X,H)}$ を作る．

- (X, H) において，H を充足する真理値割り当ての 1 つ $f(x_1) = 1$, $f(x_2) = 0$, $f(x_3) = 1$ が存在する．

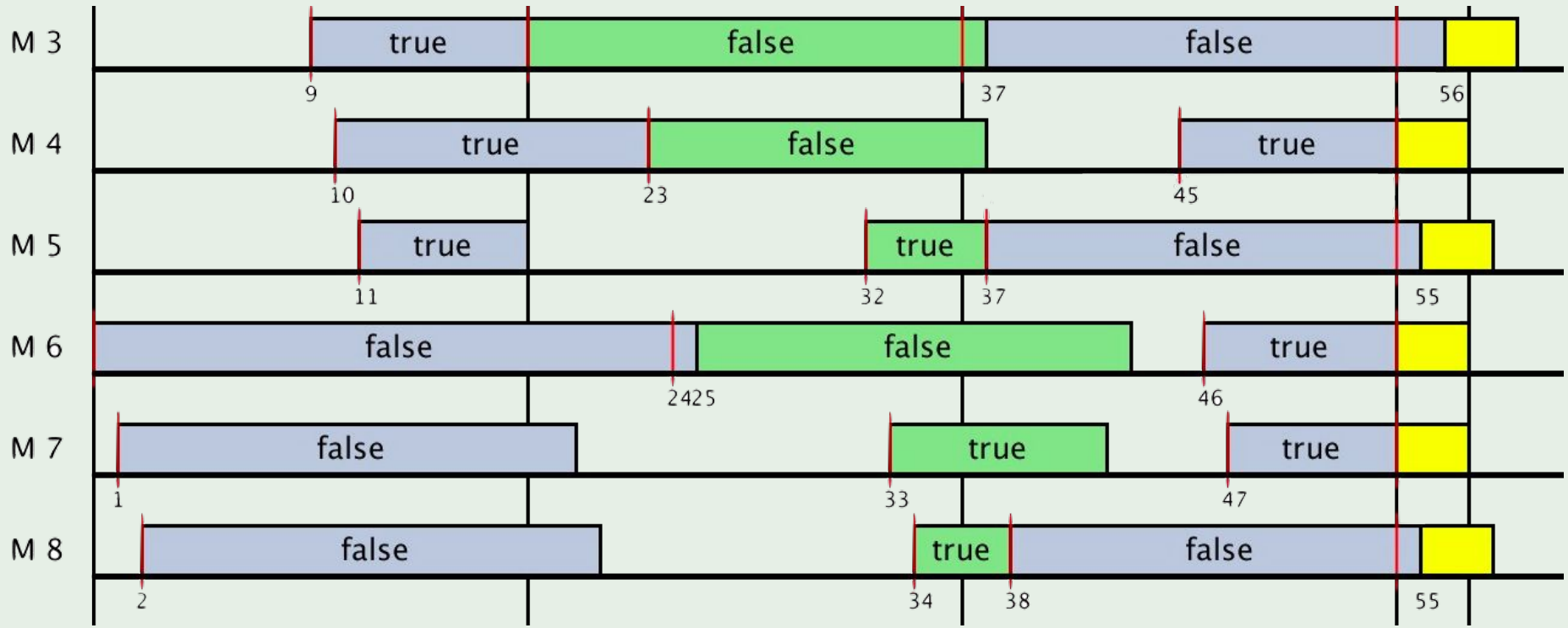
各 $i \in \{0, 1, 2\}$ における機械 i 上のスケジュールは以下の通り．



付録：多項式還元の例（3-SAT の判定結果が **Yes** のとき）

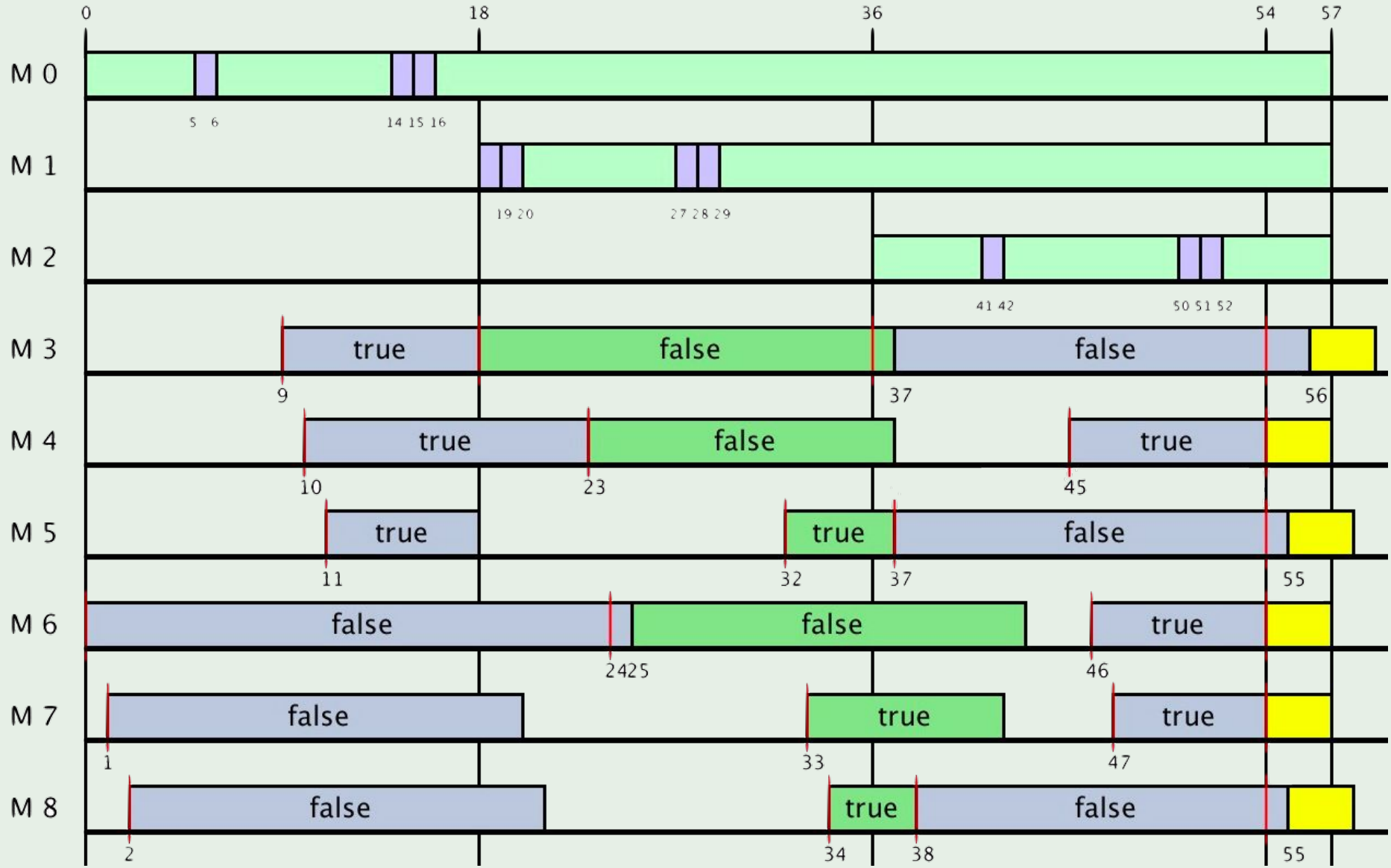
例：各節に対応する機械におけるスケジュール

H を充足する真理値割り当てが存在することから，各節におけるリテラルのなかで，少なくとも 1 つ 1 (true) に割り当てられたジョブが存在する．
したがって，各 $k \in \{3, \ldots, 8\}$ における機械 k に割り当てられた最後のジョブの待ち時間は高々 2 である．以下の通り．



付録：多項式還元の例（3-SAT の判定結果が **Yes** のとき）

例：すべての機械におけるスケジュール



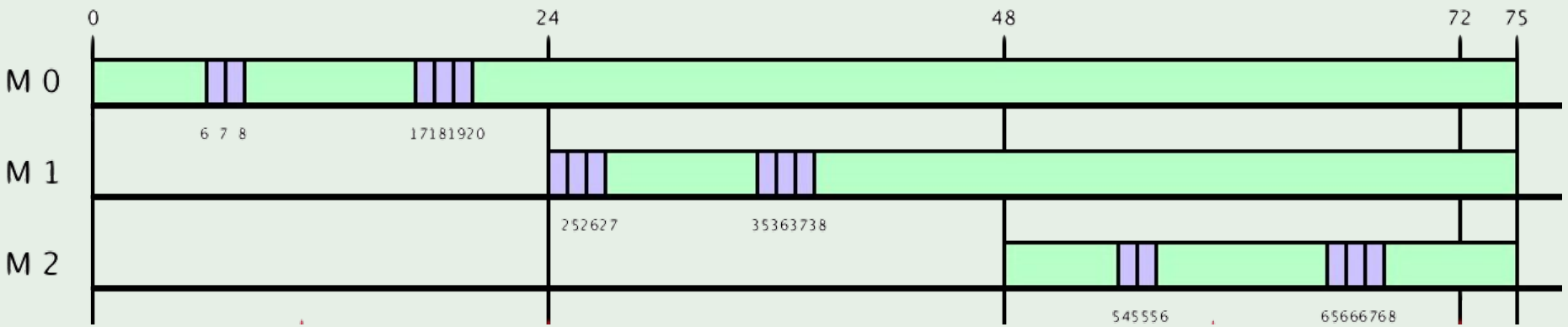
付録：多項式還元の例（3-SAT の判定結果が **No** のとき）

例：各変数に対応する機械におけるスケジュール

$X = \{x_1, x_2, x_3\}$，$H = \{\{x_1, x_2, x_3\}, \{\bar{x}_1, \bar{x}_2, \bar{x}_3\}, \{x_1, \bar{x}_2, \bar{x}_3\}, \{\bar{x}_1, x_2, \bar{x}_3\}, \{\bar{x}_1, \bar{x}_2, x_3\}, \{\bar{x}_1, x_2, x_3\}, \{x_1, \bar{x}_2, x_3\}, \{x_1, x_2, \bar{x}_3\}\}$ の 2 項組 (X, H) が 3-SAT のインスタンスとして与えられたとき，前のスライドで説明した方法により，SWT のインスタンス $I_{(X,H)}$ を作る．

- (X, H) において，H を充足する真理値割り当ては存在しない．
- ここで，$f(x_1) = 1$，$f(x_2) = 1$，$f(x_3) = 1$ としたとき，

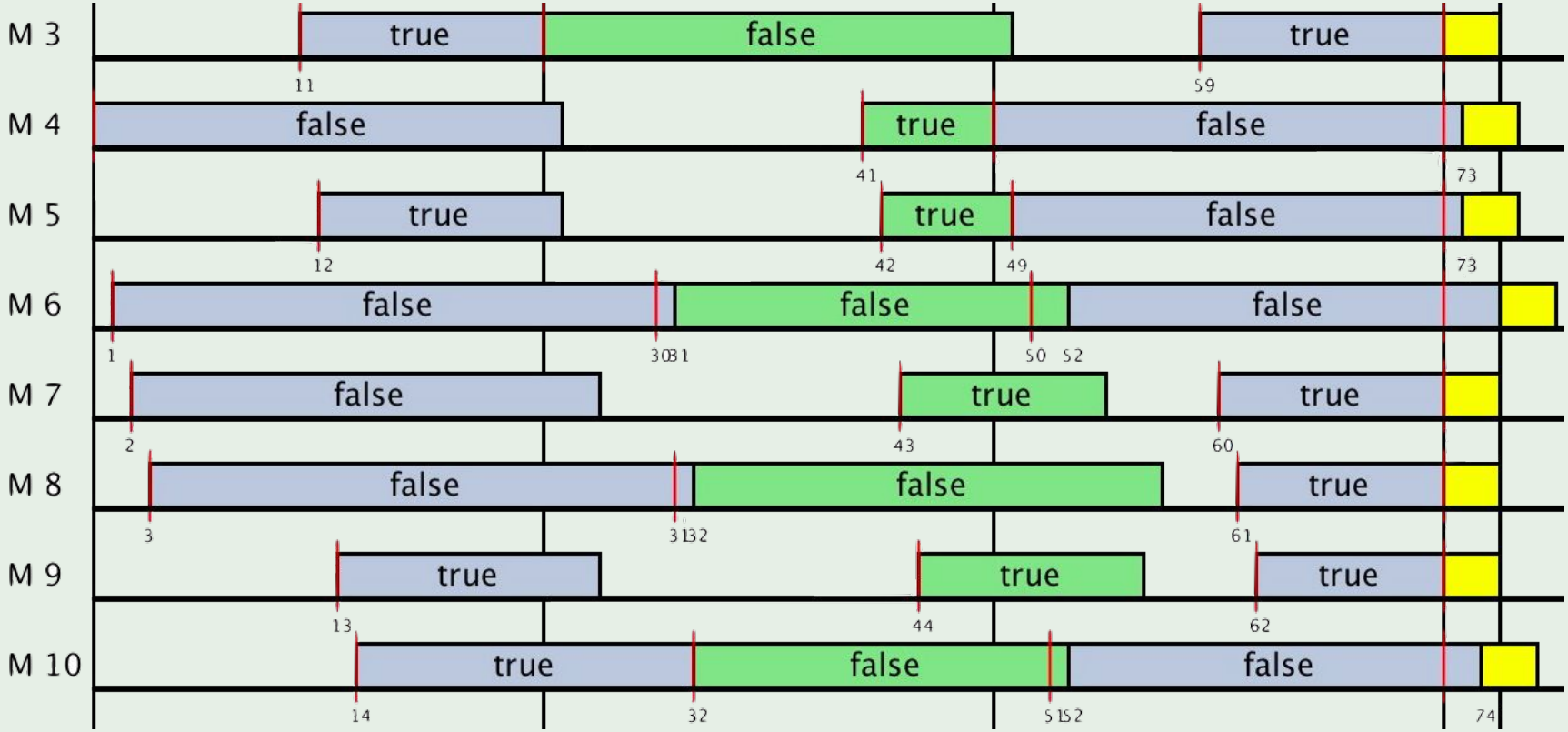
各 $i \in \{0, 1, 2\}$ における機械 i 上のスケジュールは以下の通り．



付録：多項式還元の例（3-SAT の判定結果が **No** のとき）

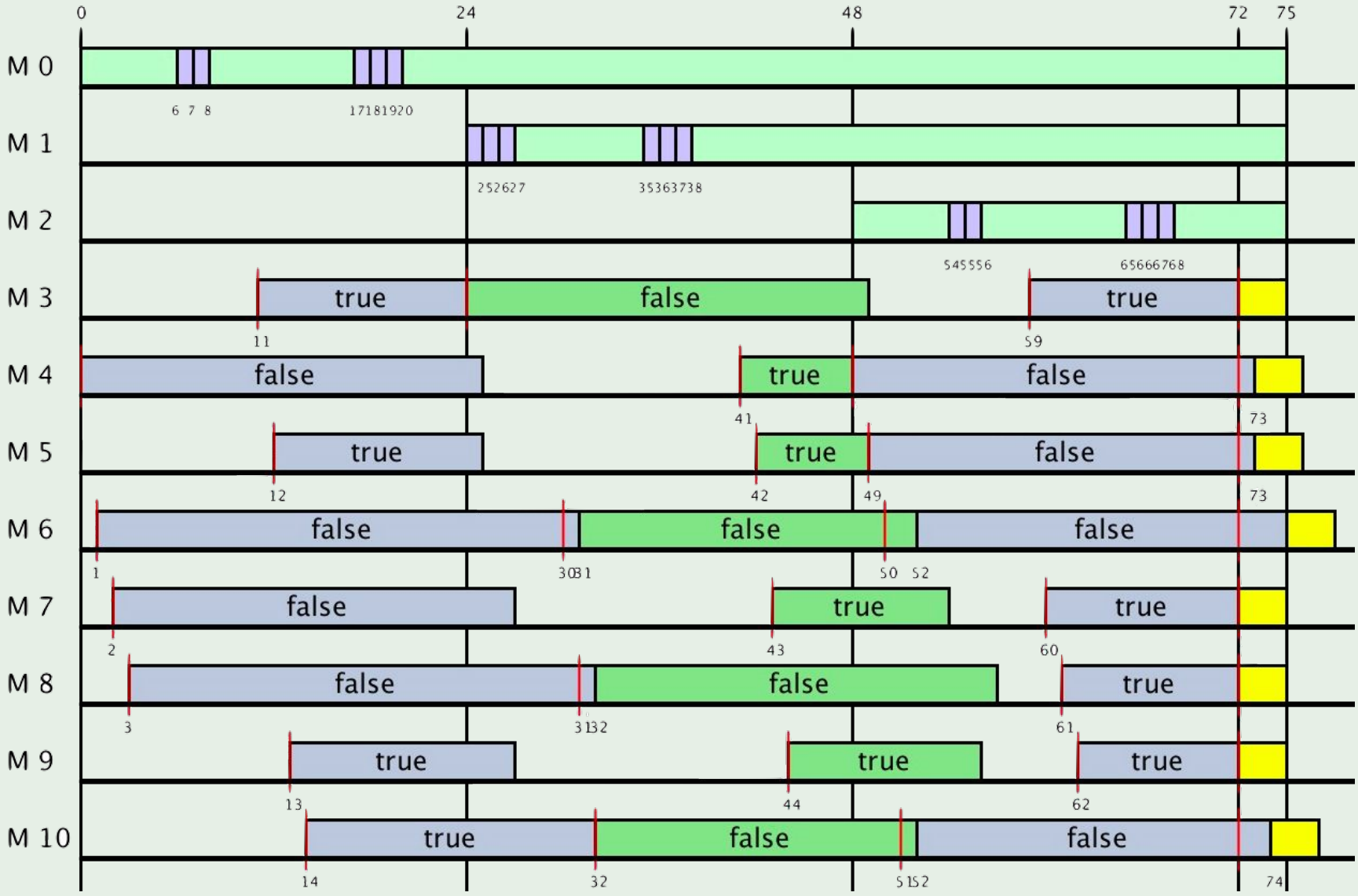
例：各節に対応する機械におけるスケジュール

H を充足する真理値割り当てが存在しないことから，各節におけるリテラルのなかで，すべて 0 (false) に割り当てられたジョブが割り当てられた機械が少なくとも 1 つ存在する．
したがって，各 $k \in \{3, \ldots, 10\}$ における機械 k に割り当てられた最後のジョブの待ち時間が 3 以上となる機械が存在する．以下の通り．



付録：多項式還元の例（3-SAT の判定結果が **No** のとき）

例：すべての機械におけるスケジュール



付録：多項式還元の例（特殊なケース）

特殊なケースにおけるインスタンスの作り方

例えば，$X = \{x_1, x_2, x_3\}$，$H = \{\{x_1, x_2, x_3\}, \{\bar{x}_1, \bar{x}_2, \bar{x}_3\}\}$ のとき，

- 各 $i \in \{1, 2, 3\}$ に対して，α_i，β_i は 3 以下である．
- このとき，各 $i \in \{1, 2, 3\}$ に対して，α_i，β_i は 3 以上となるように，ダミーの節を作る．以下の通り．

$$H = \{\{x_1, x_2, x_3\}, \{x_1, x_2, x_3\}, \{x_1, x_2, x_3\}, \{\bar{x}_1, \bar{x}_2, \bar{x}_3\}, \{\bar{x}_1, \bar{x}_2, \bar{x}_3\}, \{\bar{x}_1, \bar{x}_2, \bar{x}_3\}\}$$

- 作成したダミー節は，本来 H の要素である節と同じ節を複数個用意したに過ぎない．そのため，真理値割り当てに関係なく，判定結果にも影響しない．

上記のように，ダミー節を含んだ節の集合を作り，その後の SWT のインスタンスへの変換する．インスタンスの変換方法は，前のスライドで記述した通り．

付録：3-SAT のインスタンスにおける注意点

3-SAT のインスタンスにおいて除くケース

H に現れるリテラルが各変数，もしくは変数の論理否定をとったもののいずれかしか現れないかつ，そのリテラルがすべての節に現れる場合，H を充足する真理値割り当ては常に存在するためこの場合を除く．

例：3-SAT のインスタンスにおいて除くケース

$X = \{x_1, x_2, x_3\}$，$H = \{\{x_1, x_2, \bar{x}_3\}, \{\bar{x}_1, \bar{x}_2, \bar{x}_3\}\}$ のとき，変数 x_3 に対応するリテラルは，$\bar{x}_3$ しか H に現れない．さらに，すべての節に $\bar{x}_3$ が現れる．このとき，$f(x_3) = 0$ とすることで，H を充足する真理値割り当てが常に存在する．

付録：ヒューリスティックの精度の保証の証明

精度の保証の証明

あるインスタンスに対する，最適解から得られた最大待ち時間が 0 のとき，各機械 $M \in \mathcal{M}$ におけるジョブは以下を満たす．

$$\forall J, J' \in \mathcal{J}_M \Big[J \neq J' \Rightarrow \big[r(J), r(J) + p(J)\big) \cap \big[r(J'), r(J') + p(J')\big) = \emptyset \Big]$$

つまり，各機械 $M \in \mathcal{M}$ におけるジョブは処理開始可能時刻の昇順でスケジュールされているかつ，各ジョブの処理時間の区間は重ならない．
したがって，最適解から得られる最大待ち時間が 0 となるインスタンスに対して，処理開始可能時刻の昇順で，すぐに処理開始可能な機械に割り当てることで最適なスケジュールを作ることができる．

付録：分割生成アルゴリズムの改良部分

表記の導入

- ジョブ数 n，機械数 m
- 要素数 n の配列 b は分割を表す．
- 要素数 n の配列 $b_{\max}$ は 各 $1 \leq i \leq n$ に対して，i 番目の要素は，それまでのジョブが割り当てられた機械の最大数を格納している．つまり，$b_{\max}[i] = \max_{1 \leq j \leq i-1} b[j]$

分割生成アルゴリズムの改良部分

入力： $I = (b, b_{\max}, m)$

出力： 分割 b

Step 1. 各 $1 \leq i \leq n$ に対して，以下の処理を繰り返す．

Step 1.1. $b_{\max}[n-i] \geq m-i$ のとき，処理を終了する．

Step 1.2. $b[n-i] := m-i$，$b_{\max}[n-i] := m-i$ とする．

Step 2. b を出力する．

付録：分割生成アルゴリズムの改良部分

例：$b = (0, 1, 0, 1)$，$b_{\max} = (0, 1, 1, 1)$，$m = 3$ のとき

Step 1. $b_{\max}(0, 1, 1, 1)$ より，$1 < m - 1$.

Step 2. したがって，$b = (0, 1, 0, 2)$，$b_{\max} = (0, 1, 1, 2)$ とする.

Step 3. $b_{\max}(0, 1, 1, 2)$ より，$1 \geq m - 2$.

Step 4. b を出力する.

分枝限定法アルゴリズムの改良部分

入力： $I = (\mathcal{J}', S)$

出力： スケジュール S.

Step 1. 各 $1 \leq i \leq |\mathcal{J}'|$ における J'_i について，以下の処理を繰り返す.

Step 1.1. $p(J'_i) = \min_{j \in \{1,\ldots,|\mathcal{J}'|\}} p(J'_j)$ とする.

Step 1.2. $S := S \cup J'_i$ とする.

Step 2. S を出力する.

- Step 1. で $\mathcal{J}'$ におけるすべてのジョブの処理時間を $\mathcal{J}'$ の中の最小の処理時間に設定している.
- 上記の操作により，$\mathcal{J}'$ におけるジョブの本来の処理時間は，設定した処理時間以上である.
- このとき，処理開始可能時刻順で処理して得られるスケジュールにおける最大待ち時間は，$\mathcal{J}'$ におけるコストの下限である.