青 山 学 院 大 学 理 工 学 研 究 科
マ ネ ジ メ ン ト テ ク ノ ロ ジ ー コ ー ス

修 士 論 文

納期ズレ幅の導入による JIT スケジューリング問題の緩和とその解法

2 0 1 2 年 度

川俣 友佳
35611155

指導教員　宋 少秋　印

目 次

第1章　序論

1.1　研究背景

一般に，製造業では受注生産の商品を，商品の納入期限 (納期) に間に合うように，製造する．企業は，顧客からの信頼を維持するため，極力納期遅れが生じないように商品の生産計画を立てる．また，製造に必要な部品の在庫を多く持つことは，保管にかかる費用や，部品の廃棄による損失のリスクなどを要因として好ましくない．企業の多くは，商品の受注後に，製造に必要とする部品を調達している．商品製造の処理の開始は，必要な部品の到着後となるため，部品の到着時刻は，商品製造の処理の開始可能時刻となる．処理開始可能時刻付き遅れジョブ荷重和最小化問題とは，処理開始可能時刻を制約とし，納期遅れによる不利益の最小化を目的とするスケジューリング問題である．この問題は，ジョブを処理する機械が 1 台の場合でも既に，強 **NP** 困難であることが知られている (Garey & Johnson [3])．以降，処理開始可能時刻付き遅れジョブ荷重和最小化問題 (**S**equencing with **R**elease **T**imes and **D**ue **D**ates) を， SRTDD と表記する

SRTDD において，各ジョブは，処理開始可能時刻以降に処理を開始する．処理開始可能時刻を制約として設けることは，完了時刻に制約を設けることと等価である．本研究で導入する納期ズレ幅とは，各ジョブについて，納期よりどれくらい手前で処理を完了することを許すか，という制約を定めるパラメータであり， SRTDD は納期ズレ幅に基づいて表記できる． SRTDD では，ジョブ毎に異なる納期ズレ幅をゆるし，その値は任意である．

SRTDD の処理開始可能時刻に制約を最も弱くすると， SRTDD は遅れジョブ荷重和最小化問題に対応する．この問題は，ジョブを処理する機械が 1 台の場合，弱 **NP** 困難であることが知られており (Karp [6])，疑似多項式時間解法が提案されている (Lawler & Moore [10])．納期ズレ幅に基づいて問題を表すと，各ジョブの納期ズレ幅は，納期ズレ幅が取りうる値の最大値をとる．以降，遅れジョブ荷重和最小化問題 (**S**equencing to minimize the weighted number of **T**ardy jobs) を， ST と表記する．

ジャストインタイム (JIT) 生産方式では，“必要なものを，必要なときに，必要なだ

け” 調達し，製造することで，在庫コストの低減を図る．JIT 生産方式に基づき，納期丁度に処理を完了するようにジョブをスケジュールする問題は， JIT スケジューリング問題として知られている． SRTDD の処理開始可能時刻の制約を最も強くすると，SRTDD は JIT スケジューリング問題に対応し，問題の目的は，納期丁度に処理を完了するジョブ (JIT ジョブ) の荷重和の最大化である．この問題は，ジョブを処理する機械が 1 台の場合，多項式時間解法を持つ (Lann & Mosheiov [9])． 納期ズレ幅に基づいて問題を表すと，各ジョブの納期ズレ幅は，納期ズレ幅がとりうる値の最小値， 0 をとる．以降， JIT ジョブの荷重和の最大化を目的とする JIT スケジューリング問題 (**S**equencing to maximize the weighted number of **JIT** jobs) を， SJIT と表記する．

納期ズレ幅を軸にとって SRTDD の従来研究を調査したところ，問題の多くは未だ計算複雑さが未解決であることが明らかとなった．

1.2 研究目的

SRTDD は強 **NP** 困難であることが知られているが，納期ズレ幅がとる値によっては，効率的に最適な解が求まる．しかし，効率的に最適な解が求まる場合と，手におえない場合の境界は未だ明らかでなく， SRTDD の難しさの本質は現れていない．

本研究の目的は，納期ズレ幅の値が SRTDD の計算複雑さに与える影響を明らかにすることである．納期ズレ幅に基づいて解明にあたるが，計算複雑さが未だ明らかでない問題の中でも，より現実的な問題について優先的に扱う．

1.3 研究成果

本研究では， SRTDD について納期ズレ幅を導入し，納期ズレ幅に特徴付けられる以下のような研究成果を得た．

SRTDD は，強 **NP** 困難として知られており，任意の問題に対する効率的な厳密解法は望めない．入力に制限を加えることで，疑似多項式時間，多項式時間で解くことができることが示されているが (Lawler [12], Baptiste [2])，問題に対する近似解法は未だ開発されていない．本研究では， SRTDD に対し，並列機械モデルにおける近似解を求める $\mathrm{O}(n^4 \log n)$ 時間の解法を開発した．開発した近似解法の特徴として，その近似率は，入力の納期ズレ幅に基づく．これは，納期ズレ幅を導入することで， SRTDD におけるジョブの完了時刻と納期との関係により，ジョブ間の関係が明らかとなったことによって得た

成果である．また，開発した近似解法に対するタイトな例を示すことで，開発した近似解法の近似率はこれ以上改良できないことを示した．

JIT 生産方式では，処理開始可能時刻丁度に部品を調達すること，納期丁度に商品の製造を完了することを徹底することで，在庫コストの削減を図るが，“丁度に” という制限の厳しさから厳密な実現は難しく，現実では若干の時間的余裕をもって，調達，製造していると考えられる．本研究では， SJIT を納期ズレ幅に基づいて緩和し，緩和した問題の定式化を行った． SJIT の緩和問題に対して，並列機械モデルにおける最適解を求める $\mathrm{O}(mn^{m+1}\max_i\{\alpha_i\}^m)$ 時間の解法を開発した．これは，SJIT の緩和問題に対する疑似多項式時間解法であり，機械数 m が定数，かつ，各納期ズレ幅 α_i がジョブ数 n の多項式であるとき，多項式時間解法となる．この成果は， JIT 生産方式を採用している企業での応用が期待される．

SRTDD に対する近似解法の開発と， SJIT の緩和問題に対する厳密解法の開発により，SRTDD のほとんどの入力に対し，本研究で開発したいずれかの解法が適用できる．

任意の納期ズレ幅を許す SRTDD は，強 **NP** 困難である．この証明を見直し，納期ズレ幅の取りうる値の範囲にある制限を加えたとしても， SRTDD は依然，強 **NP** 困難であることを示した．

1.4 章構成

第 2 章では， SRTDD の定式化を行う．また，本研究の特徴である納期ズレ幅を導入し，納期ズレ幅に基づいて SRTDD を表記する．

第 3 章では， SRTDD, ST, SJIT それぞれに対する従来研究の成果を，計算複雑さの観点，解法の開発に用いられる手法の観点から，それぞれまとめる．また，納期ズレ幅の値に着目してそれらの成果を整理し， SRTDD のうち，問題の計算複雑さが明らかでない部分について考察する．

第 4 章から第 6 章にかけて，本研究の成果を述べる．第 4 章では，まず， SRTDD に対して開発した近似解法について述べる． SJIT は，各納期ズレ幅の値を 0 とする SRTDD であり， SJIT において納期以前に処理を完了するジョブの完了時刻は納期と等しい．また， SRTDD における実行可能なスケジュールには，各ジョブの完了時刻が納期と等しいスケジュールが存在する．この実行可能なスケジュール間の関係により，近似解法を開発した．開発した近似解法の特徴として，解法の近似率は，入力となる納期ズレ幅に基づく．

第 5 章では，納期ズレ幅に基づいて緩和した SJIT に対する厳密解法について述べる．まず，納期ズレ幅に基づく緩和に関連して，ジョブの処理順序について述べ， SJIT を緩和した問題の定式化を行う．動的計画法に基づいて開発した厳密解法について述べ， SJIT を緩和した問題について，計算複雑さをまとめる．

第 6 章では，納期ズレ幅の値を制限した SRTDD について，そのような SRTDD もまた，強 **NP** 困難であることを示す．第 7 章では，結論として，本研究の研究成果と今後の課題を述べる．

第2章　問題の定式化

2.1　処理開始可能時刻付き遅れジョブ荷重和最小化問題

処理開始可能時刻付き遅れジョブ荷重和最小化問題 (SRTDD) とは，各ジョブの処理開始可能時刻を制約とし，納期遅れによる不利益の合計の最小化を目的とするスケジューリング問題である．以下の前提のもと，

- 各機械は，あるジョブの処理の完了と同時に異なるジョブの処理を開始できるが，複数のジョブを同時に処理することはない
- 各ジョブの処理は任意の 1 機械の処理で完了する
- 各ジョブは中断されることなく処理される

処理開始可能時刻の制約を満たすスケジュールのうち，納期までに処理を完了しないジョブ (遅れジョブ) の荷重和を最小とするスケジュールを求める．

各ジョブを処理する機械の性能によって，機械モデルは次のように分類される．まず，ジョブを処理する機械の台数について，一つの機械で処理する単一機械モデルと，複数の機械で処理を行う並列機械モデルに分類される．並列機械モデルはさらに，全ての機械の性能が等しい同一並列機械モデル，機械ごとに処理速度が異なる一様並列機械モデル，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に分類される．

以下では，同一並列機械モデルについて， SRTDD を定式化する．各機械がジョブの処理を開始できる時刻を 0 とする．

n 個のジョブ $J_1, \ldots, J_n$ を m 台の同一機械 $M_1, \ldots, M_m$ で処理する．

入力は，各ジョブ J_i ($i \in \{1, \ldots, n\}$) の，処理時間 p_i ，処理開始可能時刻 r_i ，納期 d_i ，荷重 w_i であり，それぞれの値は自然数である．各ジョブの処理時間をまとめて $P = (p_1, \ldots, p_n) \in \mathbf{N}^n$ ，処理開始可能時刻をまとめて $R = (r_1, \ldots, r_n) \in \mathbf{N}^n$ ，納期をまとめて $D = (d_1, \ldots, d_n) \in \mathbf{N}^n$ ，荷重をまとめて $W = (w_1, \ldots, w_n) \in \mathbf{N}^n$ と表し，入力は 4 項組， (P, R, D, W) と表す．

問題の前提に基づき，スケジュールを定式化する．スケジュールは 2 項組 (A, C) であり，スケジュールによって，各ジョブを，どの機械で，いつ処理をするかを決める．

$A = (A_1, \ldots, A_m)$ はジョブの機械への割り当てを表す m 次元ベクトルで，各 $A_j \subseteq \{1, \ldots, n\}$ は機械 M_j に割り当てるジョブの集合を表す．各ジョブは複数の機械で処理されないことから，任意の異なる $j, j' \in \{1, \ldots, m\}$ について A_j と $A_{j'}$ は互いに素である ($A_j \cap A_{j'} = \emptyset$)．また，各ジョブは何れかの機械で処理されることから，$\cup_{j=1}^{m} A_j = \{1, \ldots, n\}$ である．

$C : \{1, \ldots, n\} \to \mathbf{N}$ はジョブの添え字について自然数を返す関数であり，各 $C(i)$ はジョブ J_i の完了時刻を表す．

実行可能性： 以下の制約を満たすスケジュール (A, C) は実行可能である．

任意の機械 M_j ($j \in \{1, \ldots, m\}$) について， M_j に割り当てられた任意の異なる二つのジョブ J_i, J_k ($j, k \in A_j$ かつ $i \neq k$) は同時に処理されない (図 2.1).

$$[C(i) - p_i, C(i)) \cap [C(k) - p_k, C(k)) = \emptyset.$$

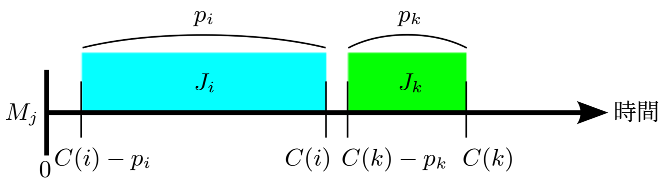

図 2.1: 処理区間

各ジョブ J_i ($i \in \{1, \ldots, n\}$) は，処理開始可能時刻以降に処理を開始する (図 2.2).

$$C(i) \geq r_i + p_i \quad (\Leftrightarrow \quad C(i) - p_i \geq r_i).$$

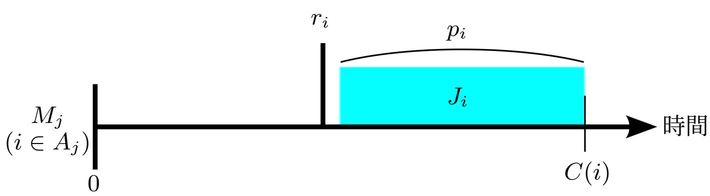

図 2.2: 処理開始可能時刻

SRTDD では，実行可能なスケジュールについて，納期までに処理を完了しなかったジョブ（遅れジョブ）の荷重和の最小化を目的とし，これは，納期までに処理を完了したジョ

ブの荷重和の最大化と等価である．この問題は，実行可能なスケジュールにおいて，ジョブが納期までに処理を完了するか否かに着目するもので，完了時刻と納期の差に応じた利益の変化，不利益の変化はない．

目的： 実行可能なスケジュール (A, C) のうち，納期以前に処理を完了するジョブの荷重和，

$$\varphi(A, C) = \sum_{i \in \mathcal{Q}(A,C)} w_i,$$

を最大とするスケジュールを求める．ただし， $\mathcal{Q}(A, C)$ はスケジュール (A, C) において納期以前に処理を完了するジョブの集合である ($\mathcal{Q}(A, C) = \{i \in \{1, \ldots, n\} \mid C(i) \leq d_i\}$)．

ここで，n 個のジョブのうち，処理開始可能時刻 r_i と同時に処理を開始したとしても，納期 d_i 以前に処理を完了することはないジョブ J_i が存在すると仮定する ($r_i > d_i - p_i$)．前述の目的関数に関する議論より，ジョブ J_i が処理開始可能時刻以降のどの時刻に処理を開始したとしても，目的関数の値に影響しない．このことから，$r_i > d_i - p_i$ を満たす各ジョブは， $r_i \leq d_i - p_i$ を満たす各ジョブの処理完了後に任意の機械で，任意の順に処理するものとする．以降，一般性を失うことなく，各処理開始可能時刻 r_i は， $r_i \leq d_i - p_i$ を満たすものとする．

特に，各ジョブの処理開始可能時刻の制約が最も弱い場合，つまり r_i が各機械の稼働開始時刻 0 と等しい場合 ($r_i = 0$)，各ジョブは時刻 0 以降の任意の時刻に処理を開始することが可能であり， SRTDD は遅れジョブ荷重和最小化問題 (ST) に対応する．また，入力 (P, R, D, W) において，処理開始可能時刻の制約が最も強い場合，つまり r_i が $r_i = d_i - p_i$ を満たす場合，実行可能なスケジュール (A, C) において納期以前に処理を完了するジョブ J_i は ($i \in \mathcal{Q}(A, C)$)，納期と同時に処理を完了し ($C(i) = d_i$)， SRTDD は JIT ジョブ荷重和最大化問題 (SJIT) に対応する．

2.2 納期ズレ幅の導入

任意の実行可能なスケジュール (A, C) において， $\mathcal{Q}(A, C)$ の要素である各ジョブ J_i は，処理開始可能時刻以降に処理を開始し，納期以前に処理を完了する (図 2.3(a))．処理開始可能時刻を制約として設けることは，完了時刻に制約を設けることと等価である．納期以前に処理を完了するジョブ J_i の完了時刻に着目すると， J_i の完了時刻は $C(i) \in [r_i + p_i, d_i]$ を満たし， J_i の完了時刻と納期とのズレは，高々$d_i - (r_i + p_i)$ である (図 2.3(b))．

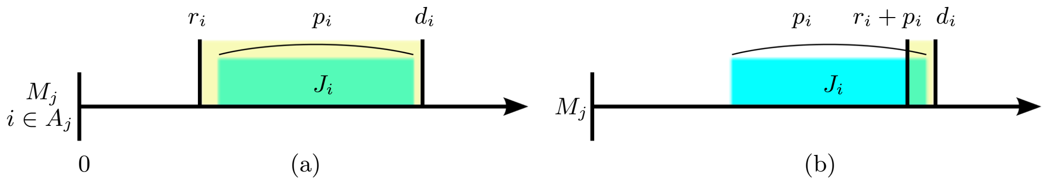

図 2.3: 納期までに処理を完了するジョブ

納期ズレ幅とは，与えられた処理開始可能時刻に基づき，各ジョブが納期以前に処理を完了する際，どれだけ手前で処理を完了することを許すか，という制約を定めるパラメータである．各ジョブの納期ズレ幅 α_i は

$$\alpha_i = d_i - (r_i + p_i)$$

であり，各ジョブの完了時刻は以下を満たす.

$$C(i) \geq d_i - \alpha_i$$

処理開始可能時刻の前提と (各 r_i は $r_i \leq d_i - p_i$ を満たす)，各処理開始可能時刻が自然数であることから，各 α_i は $\alpha_i \leq d_i - p_i$ を満たす自然数である．

以降， SRTDD を納期ズレ幅に基づいて表し，入力は 4 項組， (P, α, D, W) とする．ただし， $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbf{N}^n$ は非負の n 次元ベクトルであり，各 $i \in \{1, \ldots, n\}$ について $\alpha_i \leq d_i - p_i$ を満たす.

特に，各 α_i が $\alpha_i = d_i - p_i$ を満たす場合，問題は ST に対応し，各 α_i が $\alpha_i = 0$ を満たす場合，問題は SJIT に対応する．

第3章　従来研究

ここでは，処理開始可能時刻付き遅れジョブ荷重和最小化問題 (SRTDD) と，遅れジョブ荷重和最小化問題 (ST)， JIT ジョブ荷重和最大化問題 (SJIT) について，従来研究の成果と，解法の開発手法を紹介する．

第 3.4 節では，納期ズレ幅の値が SRTDD の難しさに与える影響に着目し，納期ズレ幅を軸にとり， SRTDD の計算複雑さに関する従来研究をまとめる．

3.1　処理開始可能時刻付き遅れジョブ荷重和最小化問題

この問題は 3 -PARTITION からの還元により，強 **NP** 困難であることが証明されている (Garey & Johnson [3])． 3 -PARTITION の定義は以下である．

Instance: 要素数が $3h$ である集合 $X = \{x_1, \ldots, x_{3h}\}$ ，分ける幅を表す正の整数 $B \in \mathbf{Z}^+$ と，各 $x \in X$ のサイズ $s(x) \in \mathbf{Z}^+$ ．ただし，サイズは次を満たす．各 $x \in X$ について， $B/4 < s(x) < B/2$ であり，かつ $\sum_{x \in X} s(x) = hB$ である．

Question: 各 $1 \leq i \leq h$ について， $\sum_{x \in X_i} s(x) = B$ を満たす X の h-分割， $X_1, \ldots, X_h$ が存在するか？ただし，s の定義より，前述の X の h -分割において，各 X_i の要素数は $|X_i| = 3$ を満たす．

$X = \{x_1, \ldots, x_{3h}\}$ ， $B \in \mathbf{Z}^+$ ， $s : X \to \mathrm{Z}^+$ の 3 項組 (X, B, s) を任意の 3 -PARTITION の入力とする．

(X, B, s) に基づき， SRTDD の入力を以下のように設定する． $1, \ldots, 4h - 1$ のジョブを，単一機械で処理する．各ジョブの処理時間，納期ズレ幅，納期を以下のように設定する．

$$p_i = \begin{cases} s(x_i) & \text{if } 1 \leq i \leq 3h, \\ 1 & \text{if } 3h + 1 \leq i \leq 4h - 1. \end{cases}$$

$$d_i = \begin{cases} hB + h - 1 & \text{if } 1 \leq i \leq 3h, \\ (i-3h)B + i - 3h & \text{if } 3h+1 \leq i \leq 4h-1. \end{cases}$$

$$\alpha_i = \begin{cases} d_i - p_i & \text{if } 1 \leq i \leq 3h, \\ 0 & \text{if } 3h+1 \leq i \leq 4h-1. \end{cases}$$

各ジョブの重み w_i を 1 とする．明らかに各 $i \in \{1, \ldots, 4h-1\}$ について，納期ズレ幅 α_i は， $\alpha_i \leq d_i - p_i$ を満たす自然数であることから， (P, α, D, W) は SRTDD の入力である．以降 (X, B, s) に基づいて前述のように設定した (P, α, D, W) を区別して $I_{(X,B,s)}$ と表す ($I_{(X,B,s)} = (P, \alpha, D, W)$)．

また，単一機械で処理を行うことから，任意の実行可能なスケジュールにおいて， $A = (A_1) = (\{1, \ldots, n\})$ である．この項では，スケジュール (A, C) を簡略化して C と表す．

以下，各 $1 \leq i \leq h$ について， $\sum_{x \in X_i} s(x) = B$ を満たす X の h-分割， $X_1, \ldots, X_h$ が存在することと，$I_{(X,B,s)}$ を入力とする SRTDD に対して $\varphi(C) \geq 4h-1$ を満たす実行可能なスケジュール C が存在することが，同値であることを示す．

補題 1 各 $1 \leq i \leq h$ について， $\sum_{x \in X_i} s(x) = B$ を満たす X の h-分割， $X_1, \ldots, X_h$ が存在するならば，$I_{(X,B,s)}$ を入力とする *SRTDD* に対して $\varphi(A, C) \geq 4h-1$ を満たす実行可能なスケジュール C が存在する．

証明 (X, B, s) を任意の 3 -PARTITION の入力とし， $X_1, \ldots, X_h$ を，各 $1 \leq i \leq h$ について， $\sum_{x \in X_i} s(x) = B$ を満たす X の h-分割とする．

$I_{(X,B,s)}$ について，各 i ($3h-1 \leq i \leq 4h-1$) の完了時刻を納期とする (各 $3h+1 \leq i \leq 4h-1$ について， $C(i) = d_i = (i-3h)B + i - 3h$)．各 i ($3h+1 \leq i \leq 4h-1$) の処理時間は 1 であることから， $[0, \max_i\{d_i\} = hB + h - 1]$ の区間は，それらの $h-1$ 個のジョブによって，h 個に分断され，それぞれの長さは B である (図 3.1)．

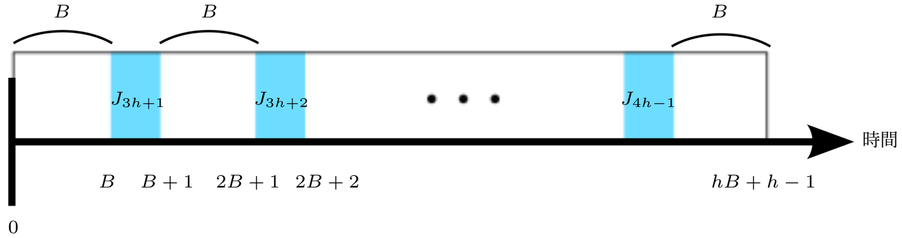

図 3.1: ジョブの配置による区間の分割

$X_1,\ldots,X_h$ について，各集合の各要素について以下のように完了時刻を設定する．各 X_i ($1 \leq i \leq h$) について， X_i の各要素を i 個目の区間，$[(i-1)B+i-1, iB+i-1]$ におさまるように完了時刻を設定することを考える．各 X_i は $\sum_{x \in X_i} s(x) = B$ を満たし，各 $x \in X$ に対応するジョブの処理時間は $p_x = s(x)$ であることから， X_i の各要素は i 個目の区間 $[(i-1)B+i-1, iB+i-1]$ ではみ出すことなく処理できる．ここでは，$X_i = \{x_{i1}, x_{i2}, x_{i3}\}$ と表し，完了時刻を次のように設定する．$C(i1) = (i-1)B+i-1+p_{i1}$, $C(i2) = C(i1) + p_{i2}$ ， $C(i3) = C(i2) + p_{i3}$.

以上のように設定したスケジュール C は，明らかに実行可能である．また， $3h+1,\ldots,4h-1$ の各ジョブは納期丁度に処理を完了し， $1,\ldots,3h$ の各ジョブもまた，納期以前に処理を完了することから (各 $x \in X$ の納期は $hB+h-1$ である)， $\varphi(C) = 4h-1$ であり， C は $\varphi(C) \geq 4h-1$ を満たす実行可能なスケジュールである． ■

補題 2 $I_{(X,B,s)}$ を入力とする *SRTDD* に対して $\varphi(C) \geq 4h-1$ を満たす実行可能なスケジュール C が存在するならば，各 $1 \leq i \leq h$ について， $\sum_{x \in X_i} s(x) = B$ を満たす X の h-分割， $X_1,\ldots,X_h$ が存在する．

証明 (X,B,s) を任意の 3 -PARTITION の入力とし，C を $I_{(X,B,s)}$ について $\varphi(C) \geq 4h-1$ を満たす実行可能なスケジュールとする．

ジョブ数が $4h-1$ であることから，全てのジョブは C において納期以前に処理を完了する．ここで， $3h+1,\ldots,4h-1$ の各ジョブの納期ズレ幅は 0 であることから，$3h+1,\ldots,4h-1$ の各ジョブは納期ちょうどに処理を完了する． C の実行可能性より，各 $x \in X$ は $3h+1,\ldots,4h-1$ のジョブによって区切られたいずれかの区間に重なることなく処理されており，処理時間の合計は納期と等しい．

$$\begin{aligned}\sum_{i \in \{1,\ldots,4h-1\}} p_i &= \sum_{i \in \{1,\ldots,3h\}} p_i + \sum_{i \in \{3h+1,\ldots,4h-1\}} p_i \\ &= \sum_{i \in \{1,\ldots,3h\}} s(x_i) + h - 1 \\ &= hB + h - 1 \\ &= \max_i \{d_i\}.\end{aligned}$$

このことから，それぞれの区間で処理されるジョブの処理時間の合計は，区間の幅 B と等しい．したがって，各 X_i を $1,\ldots,3h$ のジョブうち， i 番目の区間で処理されるジョブの集合とすると ($X_i = \{x_j \in X \mid (i-1)B+i-1 < C(j) \leq iB+i-1\}$)， $X_1,\ldots,X_h$

は，各 X_i が $\sum_{x_j \in X_i} p_j = \sum_{x_j \in X_i} s(x_j) = B$ を満たす， X の h -分割である． ■

補題 1 と補題 2 によって，各 $1 \le i \le h$ について， $\sum_{x \in X_i} s(x) = B$ を満たす X の h-分割， $X_1, \ldots, X_h$ が存在することと，$I_{(X,B,s)}$ を入力とする SRTDD に対して $\varphi(C) \ge 4h-1$ を満たす実行可能なスケジュール C が存在することは，同値である．

3 -PARTITION の入力， (X, B, s) に基づく $I_{(X,B,s)}$ の生成に要する計算時間は，明らかに多項式時間であり，入力 $I_{(X,B,s)}$ の長さは，入力 (X, B, s) の長さに関する多項式で表すことができる．したがって，以下の定理が成り立つ．

定理 3 *[3] SRTDD* は，強 **NP** 困難である．

この還元により， SRTDD は，単一機械モデルにおいて強 **NP** 困難であることから，SRTDD はどの機械モデルについても強 **NP** 困難である．

強 **NP** 困難である問題は，疑似多項式時間解法を望めない．しかし，入力を制限した SRTDD に関しては，いくつかの効率的な解法が提案されている．Lawler [12] は，単一機械モデルにおける SRTDD について，入力を以下のように制限した問題について，

$$r_1 \le r_2 \le \cdots \le r_n \quad \Leftrightarrow \quad d_1 \le d_2 \le \cdots \le d_n.$$

問題が疑似多項式時間解法を持つことを示し，その計算時間は $\mathrm{O}(nW)$ である．また，入力を同様に制限した問題について，納期以前に処理を完了するジョブ数の最大化を目的とする問題は，多項式時間解法を持つ (Kise & Ibaraki [8])．

Baptiste [2] は，単一機械モデルにおいて，各ジョブの処理時間が等しい場合 ($p_i = p$)，問題が多項式時間解法を持つことを示し，その計算時間は $\mathrm{O}(n^7)$ である．

動的計画法

SRTDD のうち，入力を制限した問題に対する厳密解法は，動的計画法に基づく．動的計画法とは，元の問題を部分問題に分け，小さな部分問題から少しずつ問題を広げつつ，それぞれの部分問題を解くことで，元の問題の最適解を求める手法である．それぞれの部分問題の解は，それまでの部分問題の解を用いて求める．ここでは， Lawler [12] の開発した解法について，動的計画法における部分問題の定義と，部分問題の解の関係について紹介する．

単一機械モデルを扱うことから，任意の実行可能なスケジュールにおいて，$A = (A_1) = (\{1, \ldots, n\})$ である．以降，スケジュールを簡略化し，C で表す．

C を任意の実行可能なスケジュールとする．納期ズレ幅が 0 よりも大きいあるジョブが存在する場合，納期以前に処理を完了するジョブの集合を同じくする， C とは異なる実行可能なスケジュール C' が存在する場合がある ($\mathcal{Q}(C) = \mathcal{Q}(C')$)．納期以前に処理を完了するジョブの集合を S とし ($S = \mathcal{Q}(C)$)， S について，以下のように完了時刻 C_S を設定する．

$$\begin{aligned} C_S(\sigma(1)) &= \max\{d_{\sigma(1)} - \alpha_{\sigma(1)}, p_{\sigma(1)}\}, \\ C_S(\sigma(i)) &= \max\{d_{\sigma(i)} - \alpha_{\sigma(i)}, C_S(\sigma(i-1))\} \quad \text{各 } 1 < i \leq |S|. \end{aligned}$$

ただし， $\sigma : \{1, \ldots, |S|\} \to S$ は S について添え字の昇順を与える関数である ($\sigma(1) < \cdots < \sigma(|S|)$ かつ $\sum_{i=1}^{|S|}\{\sigma(i)\} = S$)．明らかに， C_S もまた，実行可能である．さらに， C の実行可能性より，$d_{\sigma(i)} - \alpha_{\sigma(i)} \leq C_S(\sigma(i)) \leq C(\sigma(i)) \leq d_{\sigma(i)}$ であり，各 $i \in S$ は C_S においてもまた，納期以前に処理を完了する．

このことから， SRTDD は， $S = \mathcal{Q}(C_S)$ を満たす $S \subseteq \{1, \ldots, n\}$ のうち， $\varphi(C_S)$ を最大とする S を求める問題と等価である．

Lawler の解法において，扱われる部分問題は以下である．ジョブ $i \in \{1, \ldots, n\}$ と $w \in \{1, \ldots, \sum_i w_i\}$ について，

(i, w) **-部分問題:** $S = \mathcal{Q}(C_S)$ であり，かつ $\varphi(C_S) \geq w$ である，ジョブの部分集合 $S \subseteq \{1, \ldots, i\}$ について，S の要素であるジョブの C_S における完了時刻の最大値 $\max_{i \in S}\{C_S(i)\}$ を最小とする $\hat{S}$ を求める ($\max_{i \in \hat{S}}\{C_{\hat{S}}(i)\} = \min_S \{\max_{i \in S}\{C_S(i)\}\}$).

ここで， $L(i, w)$ を， $\hat{S}$ について， $\hat{S}$ の要素であるジョブの $C_{\hat{S}}$ における完了時刻の最大値とする ($L(i, w) = \max_{i \in \hat{S}}\{C_{\hat{S}}(i)\}$)． $L(i, w)$ とそれまでの部分問題における L との関係は以下である．

$$L(i, w) = \min \begin{cases} L(i-1, w), \\ \max\{d_i - \alpha_i, L(i-1, w - w_i) + p_i\} \end{cases}$$

Lawler は，上記の部分問題の解の関係に基づき，各 $i \in \{1, \ldots, n\}$ と各 $w \in \{1, \ldots, \sum_i w_i\}$ について順に部分問題を解くことて，問題の最適解が求まることを示した．

3.2 遅れジョブ荷重和最小化問題

遅れジョブ荷重和最小化問題 (ST) は，分割問題からの還元により，単一機械モデルについて，弱 **NP** 困難であることが証明されている (Karp [6])．分割問題の定義は以下である．

Instance: 有限集合 $X = \{x_1, \ldots, x_{|X|}\}$ と各 $x_i \in X$ のサイズ $s(x_i) \in \mathrm{Z}^+$.

Question: $\sum_{x \in X'} s(x) = \sum_{x \notin X'} s(x)$ を満たす $X' \subseteq X$ が存在するか？

$X = \{x_1, \ldots, x_{|X|}\}$ と $s : X \to \mathrm{Z}^+$ の 2 項組，(X, s) を任意の分割問題の入力とする．

(X, s) に基づき，以下のように ST の入力を設定する．$|X|$ 個のジョブ $1, \ldots, |X|$ を単一機械で処理する．各ジョブの処理時間，納期ズレ幅，納期，荷重を次のように設定する．各 $x_i \in X$ について，$p_i = w_i = s(x_i)$ ，$\alpha_i = \frac{1}{2} \sum_{x_j \in X} s(x_j) - s(x_i)$ ，$d_i = \frac{1}{2} \sum_{x_j \in X} s(x_j)$.

各 $i \in \{1, \ldots, |X|\}$ について，納期ズレ幅 α_i は，$\alpha_i = d_i - p_i$ を満たす自然数であることから，上記の (P, α, D, W) は ST の入力である．以降，(X, s) に基づいて前述のように設定した (P, α, D, W) を区別して $I_{(X,s)}$ と表す ($I_{(X,s)} = (P, \alpha, D, W)$).

また，単一機械で処理を行うことから，任意の実行可能なスケジュールにおいて，$A = (A_1) = (\{1, \ldots, n\})$ である．この項では，スケジュール (A, C) を簡略化して C と表す．

以下，$\sum_{x_i \in X'} s(x_i) = \sum_{x_i \notin X'} s(x_i)$ を満たす $X' \subseteq X$ が存在することと，$I_{(X,s)}$ を入力とする ST ついて，$\varphi(C) \geq \frac{1}{2} \sum_{x_i \in X} s(x_i)$ を満たす実行可能なスケジュール C が存在することが，同値であることを示す．

補題 4 $\sum_{x_i \in X'} s(x_i) = \sum_{x_i \notin X'} s(x_i)$ を満たす $X' \subseteq X$ が存在するならば，$I_{(X,s)}$ を入力とする *ST* について，$\varphi(C) \geq \frac{1}{2} \sum_{x_i \in X} s(x_i)$ を満たす実行可能なスケジュール C が存在する．

証明 (X, s) を任意の分割問題の入力とし，X' を $\sum_{x_i \in X'} s(x_i) = \sum_{x_i \notin X'} s(x_i)$ を満たす X の部分集合とする．

X' は $\sum_{x_i \in X'} s(x_i) = \sum_{x_i \notin X'} s(x_i)$ を満たすことから，$\sum_{x_i \in X'} s(x_i) = \frac{1}{2} \sum_{x_i \in X} s(x_i)$ である．各 $i \in \{1, \ldots, |X|\}$ の処理時間が $p_i = s(x_i)$ であることから，X' の全ての要素は $[0, \frac{1}{2} \sum_{x_i \in X} s(x_i)]$ の区間内で，処理区間が重ならないように処理できる．各ジョブ $i \in \{1, \ldots, |X|\}$ の納期は $d_i = \frac{1}{2} \sum_{x_j \in X} s(x_j)$ であることから，X' の全ての要素が納期以前に処理を完了する実行可能なスケジュールが存在する．

以下，X' の全ての要素が納期以前に処理を完了する実行可能なスケジュールのうちの一つを示す．$\sigma : \{1, \ldots, |X'|\} \to X'$ を X' の要素の添え字の昇順を与える関数とする．つまり，各 $i \in \{1, \ldots, |X'| - 1\}$ について，$\sigma(i) < \sigma(i+1)$ であり，$\bigcup_{i \in \{1, \ldots, |X'|\}} \{x_{\sigma(i)}\} = X'$ である．各 $\sigma(i)$ の完了時刻を以下のように設定する．

$$C(\sigma(i)) = \begin{cases} p_{\sigma(i)} & \text{if } i = 1, \\ C(\sigma(i-1)) + p_{\sigma(i)} & \text{if } 1 < i \leq |X'|. \end{cases}$$

X の要素のうち， X' に含まれない要素の集合を $\hat{X}$ とする ($\hat{X} = X \setminus X'$)． $\hat{\sigma} : \{1, \ldots, |\hat{X}|\} \to \hat{X}$ を $\hat{X}$ の要素の添え字の昇順を与える関数とする．つまり，各 $i \in \{1, \ldots, |\hat{X}| - 1\}$ について， $\hat{\sigma}(i) < \hat{\sigma}(i+1)$ であり， $\bigcup_{i \in \{1, \ldots, |\hat{X}|\}} \{x_{\hat{\sigma}(i)}\} = \hat{X}$ である．各 $\hat{\sigma}(i)$ の完了時刻を以下のように設定する．

$$C(\hat{\sigma}(i)) = \begin{cases} d_{\hat{\sigma}(i)} + p_{\hat{\sigma}(i)} & \text{if } i = 1, \\ C(\hat{\sigma}(i-1)) + p_{\hat{\sigma}(i)} & \text{if } 1 < i \leq |\hat{X}|. \end{cases}$$

明らかに C は実行可能であり， X' のすべての要素は納期以前に処理を完了し， $\hat{X}$ の全ての要素は遅れジョブである．各 $x_i \in X$ の荷重は $w_i = s(x_i)$ であることから， C は $\varphi(C) = \sum_{x_i \in X'} w_i = \sum_{x_i \in X'} s(x_i) = \frac{1}{2} \sum_{x_i \in X} s(x_i)$ を満たす． ■

補題 5 $I_{(X,s)}$ を入力とする ST について， $\varphi(C) \geq \frac{1}{2} \sum_{x_i \in X} s(x_i)$ を満たす実行可能なスケジュール C が存在するならば，$\sum_{x_i \in X'} s(x_i) = \sum_{x_i \notin X'} s(x_i)$ を満たす $X' \subseteq X$ が存在する．

証明 (X, s) を任意の分割問題の入力とし， C を， $I_{(X,s)}$ を入力とする ST について $\varphi(C) \geq \frac{1}{2} \sum_{x_i \in X} s(x_i)$ を満たす実行可能なスケジュールとする．

$I_{(X,s)}$ の定義より，各 $i \in \{1, \ldots, |X|\}$ の荷重は処理時間と等しいので ($w_i = p_i$)，C において納期以前に処理を完了するジョブの処理時間の合計は $\sum_{i \in \mathcal{Q}(C)} p_i = \varphi(C) \geq \frac{1}{2} \sum_{x_i \in X} s(x_i)$ である．各ジョブ $i \in \{1, \ldots, |X|\}$ の納期が $d_i = \frac{1}{2} \sum_{x_j \in X} s(x_j)$ であることと， C の実行可能性より，C において納期以前に処理を完了するジョブの処理時間の合計は $\sum_{i \in \mathcal{Q}(C)} p_i = \frac{1}{2} \sum_{x_i \in X} s(x_i)$ である．

よって， X' をスケジュール C において納期以前に処理を完了するジョブに対応する x_i の集合とすると ($X' = \{x_i \mid i \in \mathcal{Q}(C)\}$)，スケジュール C においてジョブの重複はないことから， X' は $\sum_{x_i \in X'} = \frac{1}{2} \sum_{x_i \in X} s(x_i)$ であり，$\sum_{x_i \in X'} s(x_i) = \sum_{x_i \notin X'} s(x_i)$ を満たす． ■

補題 4 と補題 5 より，$\sum_{x_i \in X'} s(x_i) = \sum_{x_i \notin X'} s(x_i)$ を満たす $X' \subseteq X$ が存在することと，$I_{(X,s)}$ を入力とする ST に対し $\varphi(C) \geq \frac{1}{2} \sum_{x_i \in X} s(x_i)$ を満たす実行可能なスケジュール C が存在することは，同値である．

分割問題の入力， (X, s) に基づく $I_{(X,s)}$ の生成に要する計算時間は，明らかに多項式時間であり，入力 $I_{(X,s)}$ の長さは，入力 (X, s) の長さに関する多項式で表すことができる．したがって以下の定理が成り立つ．

定理 6 *[6] ST は，単一機械モデルにおいて，弱* **NP** *困難である．*

また，単一機械モデルにおいて，遅れジョブ数の最小化を目的とする問題は，多項式時間解法をもち，計算時間は $\mathrm{O}(n\log n)$ である (Moore [14])． Lawler [11] は，問題の入力において，処理時間の昇順が荷重の降順である場合 ($r_i \leq r_j \Leftrightarrow w_i \geq w_j$)， Moore [14] の開発した多項式時間解法を用いて， ST を多項式時間で解くことができることを示した．

また，同一並列機械モデルについて，機械数が定数の場合，同じく分割問題からの還元により， ST は弱 **NP** 困難であることが示されている (Garey & Johnson [4])．問題に対する疑似多項式時間解法が提案されており，計算時間は $\mathrm{O}(n(\sum_i p_i)^m)$ である (Akker & Hoogeveen [16])．また，機械数が入力の一部の場合，この問題は 3 -PARTITION からの還元により，強 **NP** 困難であることが知られている (Garey & Johnson [4])．

ここで，解法の開発に用いられた手法に着目すると， ST について開発された解法の多くは，動的計画法に基づく．単一機械モデルにおいて Lawler と Moore [10] の開発した疑似多項式時間解法，同一並列機械モデルにおいて， Akker と Hoogeveen [16] の開発した疑似多項式時間解法は，動的計画法に基づく．また，単一機械モデル，重みなしの問題について， Moore & Hodgson's Algorithm としてよく知られている多項式解法も，動的計画法に基づく (Moore [14])

3.3 JIT スケジューリング問題

JIT スケジューリング問題のうち， JIT ジョブの荷重和最大化を目的とする問題 (JIT ジョブ荷重和最大化問題： SJIT) は，単一機械モデル，同一並列機械モデルについて，それぞれ多項式時間解法が開発されている．単一機械モデルについて，Lann と Mosheiov [9] の開発した多項式時間解法の計算時間は，$\mathrm{O}(n^2)$ であり，同一並列機械モデルについて，Hiraishi, Levner と Vlach [5]) が開発した多項式時間解法の計算時間は， $\mathrm{O}(n^4\log n)$ である．

無関連並列機械については，機械数が定数の場合，問題が多項式時間解法を持つことが示されており，計算時間は $\mathrm{O}(mn^{m+1})$ である (Sung & Vlach [15])．このことから，一様並列機械についてもまた，機械数が定数の場合，問題は多項式時間解法を持つ．

また，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** 困難であることが示されている (Sung & Vlach [15], 付録 A)．一様並列機械モデルについて，機械数が入力の一部の場合，問題の計算複雑さは明らかでない．

ここで，解法の開発に用いられた手法に着目すると， SJIT について開発された問題のうち，動的計画法に基づく解法は，以下である．単一機械モデルにおいて Lann と Mosheiov [9] が開発した多項式時間解法, 無関連並列機械モデルにおいて Sung と Vlach [15] の開発した多項式解法 (m は定数，計算時間は $\mathrm{O}(mn^{m+1})$) は，動的計画法に基づく．

最小費用フロー問題への帰着

SJIT のうち，同一並列機械モデルにおいて， Hiraishi, Levner と Vlach [5] が開発した解法では，最小費用フロー問題への帰着によって，最適解を求める．

ジョブ数 n と機械数 m について， $n \leq m$ の場合，各ジョブを各機械に高々一つ割り当てるような任意の (A, C) について (各 $j \in \{1, \ldots, m\}$ について $|A_j| \leq 1$)， (A, C) は明らかに最適なスケジュールである．このことからこの節では， $n > m$ の場合について扱う．

$I = (P, \alpha, D, W)$ を任意の SJIT の入力とする． I に基づき，以下のように有向グラフ $G_I = (V_I, E_I)$ を設定する．

- V_I は $2n + 2$ 個の要素からなる頂点集合で， $V_I = \{s, a_1, \ldots, a_n, b_1 \ldots, b_n, t\}$ である．特に， s は始点を表し，t は終点を表す．各 $i \in \{1, \ldots, n\}$ について， a_i と b_i はジョブ i に対応する頂点である．

- E_I は有向辺の集合であり，以下の要素からなる．

  - 始点 s について， s から 各 a_i ($i \in \{1, \ldots, n\}$) に向かう辺 (s, a_i) .
  - 各 $i \in \{1, \ldots, n\}$ について， a_i から b_i に向かう辺 (a_i, b_i) .
  - 終点 t について，各 b_i ($i \in \{1, \ldots, n\}$) から t に向かう辺 (b_i, t) .
  - $d_k \geq d_i + p_i$ を満たす任意の異なる二つの $i, k \in \{1, \ldots, n\}$ ($i \neq k$) について， b_i から a_k に向かう辺 (b_i, a_k) .

SJIT において，納期以前に処理を完了する各ジョブは，納期丁度に処理を完了する．異なる二つの $i, k \in \{1, \ldots, n\}$ ($i \neq k$) について， $(b_i, a_k) \in E_I$ であるとは， J_i と J_k は同じ機械 j で納期以前に処理を完了するジョブとしてスケジュールすることができることを表し， $d_k \geq d_i + p_i$ を満たす．

$f : E_I \to \{0, 1\}$ を，グラフ G_I における始点 s から終点 t への $0 - 1$ フローとし，その流量を m とする． $n > m$ であることから，このようなフロー f は必ず存在する．た

とえば，各 $i \in \{1,\ldots,m\}$ について，$f(s,a_i)=1$ ，$f(a_i,b_i)=1$ ，$f(b_i,t)=1$ とし，E_I の他の辺 (u,v) について，$f(u,v)=0$ とした f は，s から t への流量 m の $0-1$ フローの一つである (図 3.2).

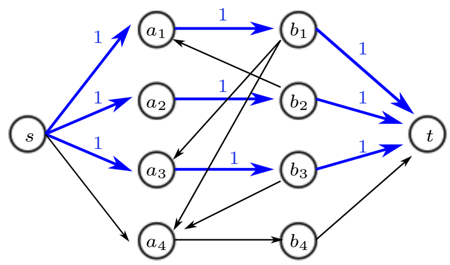

図 3.2: $n=4$ ，$m=3$ の場合の，流量 m の $0-1$ フローの例

G_I の定義より，任意の $I=(P,\alpha,D,W)$ について G_I はサイクルを含まない．このことから，G_I における任意の流量 m の $0-1$ フロー f は，以下を満たす $f_1,\ldots,f_m$ に分割することができる．$f=\sum_{j=1}^{m} f_j$ であり，

- 各 f_j ($j \in \{1,\ldots,m\}$) の流量は 1 である，
- 各 f_j において値が 1 である有向辺によって s から t へのパスが一意に決まる，
- 各 $a_1,\ldots,a_n,b_1,\ldots,b_n$ は高々一つのパスに現れる．さらに，あるパスに a_i が現れるならば，b_i もまた，同じパスに現れる．この性質は，各 a_i について a_i の出次数は 1 であり，b_i の入次数が 1 であることから導かれる．

これらの事実より，各 f_j によって一意に決まる s から t へのパスについて，パスに現れる頂点に対応するジョブの集合は，機械 j において JIT ジョブとして割り当てられたジョブの集合とみなすことができる．f_j によって一意に決まる m 個のパスにおいて何れのパスにも表れないジョブは，各 JIT ジョブの処理の完了後，任意の機械，任意の順で処理するものとする．

ここで，$G_I=(V_I,E_I)$ における辺集合 E_I に対して，以下のようにコスト関数を定義する．各 $(u,v)\in E_I$ について，

$$c_I(u,v)=\begin{cases} w_i & \text{if } u=a_i \text{ and } v=b_i, \\ 0 & \text{otherwise,} \end{cases}$$

以上の議論より，SJIT は，(G_I,c_I) を入力とし，s から t への流量 m の $0-1$ フロー f のうち, 以下の関数の値を最大化する f を求める最小費用フロー問題に対応し，その計

算時間は $O(n^4 \log n)$ である (Ahuja, Magnanti, & Orlin [1]).

$$\sum_{(u,v)\in E_I} c(u,v)f(u,v).$$

3.4 まとめ

SRTDD では，各納期ズレ幅は $0 \le \alpha_i \le d_i - p_i$ を満たす． SRTDD は 3 -PARTITION からの還元により，強 **NP** 困難であることが知られている．第3.1節より，3 -PARTITION からの還元において，各ジョブの納期ズレ幅は， $\alpha_i = d_i - p_i$ と， $\alpha_i = 0$ の何れかの値をとり， 3 -PARTITION の入力 (X, B, s) に基づいて設定した $I_{(X,B,s)}$ において，納期ズレ幅の上界は $d_i - p_i$ である． ST では，各納期ズレ幅は等しく $d_i - p_i$ である．また，SJIT では，各納期ズレ幅は等しく 0 であり，納期ズレ幅の上界は 0 である．

以下では，納期ズレ幅の上界に着目し， SRTDD に対する従来研究の成果をまとめる．表 3.1 は，単一機械モデルについてまとめたものであり，表 3.2 は同一並列機械モデルについてまとめたものである（ただし，機械数 m は定数である）．表中の多項式，疑似多項式は，それぞれ，問題が多項式時間解法を持つこと，問題が疑似多項式時間解法を持つことを表す．表中の×は，対応する問題が存在しないことを表す．

<table>
<tr><th rowspan="2">SRTDD</th><th colspan="3">納期ズレ幅の上界</th></tr>
<tr><th>0</th><th>$\cdots$</th><th>$d_i - p_i$</th></tr>
<tr><td>各 α_i が異なってもよい</td><td>×</td><td rowspan="2">OPEN</td><td>強 NP 困難 [3]</td></tr>
<tr><td>各 α_i が等しい</td><td>SJIT
多項式 [9]</td><td>ST
弱 NP 困難 [6]
疑似多項式 [10]</td></tr>
</table>

表 3.1: 単一機械モデルにおける SRTDD の従来研究の成果

<table>
<tr><th rowspan="2">SRTDD</th><th colspan="3">納期ズレ幅の上界</th></tr>
<tr><th>0</th><th>$\cdots$</th><th>$d_i - p_i$</th></tr>
<tr><td>各 α_i が異なってもよい</td><td>×</td><td rowspan="2">OPEN</td><td>強 NP 困難 [3]</td></tr>
<tr><td>各 α_i が等しい</td><td>SJIT
多項式 [5]</td><td>ST
弱 NP 困難 [4]
疑似多項式 [16]</td></tr>
</table>

表 3.2: 同一並列機械モデルにおける SRTDD の従来研究の成果 (m :定数)

納期ズレ幅の観点から従来研究を調査したが，表の計算複雑さが明らかでない部分に関する従来研究は存在しなかった． SRTDD において，納期ズレ幅の上界が $d_i - p_i$ 未満で

あり， 0 よりも大きい各問題については，問題の計算複雑さが未解決であり，効率的に精度の保障された解を求める解法は，未だ開発されていない．

全ての納期ズレ幅が等しく 0 である問題， JIT スケジューリング問題は， JIT 生産方式に基づく． JIT 生産方式では， “必要なものを，必要なときに，必要なだけ” 調達，製造する．処理開始可能時刻丁度に部品を調達すること，納期丁度に商品の製造を完了することを徹底することで在庫コストの削減を図る．しかし， “丁度に” という制限の厳しさのため， JIT 生産方式の厳密な実現は難しく，現実には各納期に若干の時間的余裕を持たせて製造していると考えられる．

SRTDD は，強 **NP** 困難である問題を内包するため，任意の入力に対する効率的な厳密解法は望めない．しかし，納期ズレ幅に着目して従来研究の成果を整理すると，多くの問題は計算複雑さが未解決であることが明らかとなった．本研究では，納期ズレ幅に基づいて， SRTDD の計算複雑さの解明にあたる．計算複雑さが未解決である問題の中でも，納期ズレ幅に基づいて JIT スケジューリング問題を緩和した問題を優先的に扱う (第 5 章)．

第4章　近似解法

処理開始可能時刻付き遅れジョブ荷重和最小化問題 (SRTDD) は，単一機械モデルにおいて既に強 **NP** 困難であり (第 3.1 節)，問題に対する効率的な厳密解法は望めない．単一機械モデル，並列機械モデルともに，問題に対する近似解法は未だ提案されておらず，問題の多くは，精度の保障された解を求める効率的な解法を持たない状況である．

本研究では，SRTDD に対し，並列機械モデルにおける近似解法を開発した．その近似率については，入力の納期ズレ幅に基づく評価を与えた．また，開発した近似解法に対するタイトな例を示した．

4.1　実行可能なスケジュール

第 3.1 節より， SRTDD は強 **NP** 困難であることが知られている．単一機械モデルについて，入力に制限を加えた問題については，効率的な解法が提案されている．同一並列機械モデルに関しては，いくつかのヒューリスティックは提案されていが (Marc and Philippe [13])，その評価は実験的なものである．強 **NP** 困難である問題は，効率的な厳密解法が望めない．単一機械モデル，並列機械モデルともに，問題のほとんどは，精度の保障された解の効率的な解法が提案されていない状況である．

本研究では，SRTDD に対して近似解法を開発した．近似解法の議論に先立ち，SRTDD と， SJIT の関係に着目する．

SJIT は，SRTDD の一つである．SRTDD において，各納期ズレ幅 α_i が $0 \leq \alpha_i \leq d_i - p_i$ を満たすことに対し， SJIT では，各納期ズレ幅は 0 と等しい． (P, α, D, W) を SRTDD の入力とする． (P, α, D, W) について，各納期ズレ幅を 0 とした 4 項組， (P, α^0, D, W) (つまり，各 $i \in \{1, \ldots, n\}$ について $\alpha_i^0 = 0$) は明らかに SJIT の入力の一つである．さらに， SRTDD の実行可能なスケジュールにおいて，納期以前に処理を完了する各ジョブ i の完了時刻は $C(i) \in [d_i - \alpha_i, d_i]$ を満たすことから，以下の議論が成り立つ． (P, α^0, D, W) を入力とする SJIT における任意の実行可能なスケジュールは， (P, α, D, W) を入力とする SRTDD における実行可能解の一つである．

4.2　開発した解法

第 4.1 節の議論より， SRTDD の入力 (P, α, D, W) について， (P, α^0, D, W) を入力とする SJIT における実行可能なスケジュールは， SRTDD における実行可能なスケジュールの一つである．

APPROXIMATE を SRTDD の入力 (P, α, D, W) について実行可能解なスケジュールの一つを返すアルゴリズムとする．

APPROXIMATE

Input： SRTDD の入力 (P, α, D, W)

Output： 実行可能なスケジュール (A, C)

Step 1. (P, α, D, W) について各納期ズレ幅を 0 とした 4 項組を， (P, α^0, D, W) とする．

Step 2. (P, α^0, D, W) について，第 3.3 節で紹介した SJIT に対する厳密解法に基づき，(P, α^0, D, W) を入力とする SJIT における最適なスケジュール (A, C) を求める．

Step 3. (A, C) を出力する．

第 3.3 節より，APPROXIMATE の計算時間は $\mathrm{O}(n^4 \log n)$ である．

SRTDD の任意の実行可能なスケジュールにおいて，各ジョブの完了時刻は $C(i) \geq d_i - \alpha_i$ を満たす．納期ズレ幅の値に着目すると，それらのジョブの完了時刻を納期に揃えるための条件が見えてくる．

本研究では， SJIT との関係と，納期ズレ幅の値に着目することによって， SRTDD に対して APPROXIMATE が出力する実行可能なスケジュールの精度を明らかにした．

定理 7 (P, α, D, W) を入力とする *SRTDD* における最適なスケジュールを (A^α, C^α) とし，APPROXIMATE (P, α, D, W) によって得られる解を (A^0, C^0) とすると，

$$\frac{\varphi(A^0, C^0)}{\varphi(A^\alpha, C^\alpha)} \geq \frac{1}{k+1},$$

ただし， k は $(k-1)\min_{i'}\{p_{i'}\} < \max_{i'}\{\alpha_{i'}\} \leq k \min_{i'}\{p_{i'}\}$ を満たす正整数である．

証明 (P, α, D, W) を任意の SRTDD の入力とすし，k を $(k-1)\min_{i'}\{p_{i'}\} < \max_{i'}\{\alpha_{i'}\} \leq k\min_{i'}\{p_{i'}\}$ を満たす正整数とする． (A^α, C^α) を問題に対する最適なスケジュールとし，(A^0, C^0) を APPROXIMATE (P, α, D, W) によって得られるスケジュールとする．

(A^α, C^α) ついて，以下の手順によってスケジュール (A, C) を生成する．

Input : (A^α, C^α)

Output : スケジュール (A, C)

Step 1. 各 E_j ($j \in \{1, \ldots, m\}$) を，(A^α, C^α) において機械 M_j に割り当てられたジョブのうち，納期以前に処理を完了するジョブの集合とする．つまり，各 $j \in \{1, \ldots, m\}$ について，

$$E_j = \{i \in A_j^\alpha \mid i \in \mathcal{Q}(A^\alpha, C^\alpha)\}.$$

Step 2. 各 $j \in \{1, \ldots, m\}$ について， E_j の各要素の完了時刻の昇順を与える関数を $\pi_j : \{1, \ldots, |E_j|\} \rightarrow E_j$ とする．つまり，各 $1 \leq i < \hat{i} \leq |E_j|$ について，$C^\alpha(\pi_j(i)) \leq C^\alpha(\pi_j(\hat{i}))$ であり，

$$\cup_{i \in \{1, \ldots, |E_j|\}}\{\pi_j(i)\} = E_j,$$

Step 3. 各 $j \in \{1, \ldots, m\}$ について E_j の要素を，完了時刻の昇順で k 個置きにとった，$k+1$ 個のジョブの集合を $F_0^j, \ldots, F_k^j$ とする (図 4.1)．つまり，各 $i \in \{1, \ldots, |E_j|\}$ について，

$$\pi_j(i) \in F_\ell^j \quad \Leftrightarrow \quad i \pmod{k+1} = \ell.$$

Step 4. 各 $j \in \{1, \ldots, m\}$ について， $F_0^j, \ldots, F_k^j$ のうち，ジョブの荷重和が最大である F_u^j をみつけ，

$$\sum_{i \in F_u^j} w_i = \max\left\{\sum_{i \in F_\ell^j} w_i \mid \ell \in \{0, 1, \ldots, k\}\right\},$$

$E_j := F_u^j$ とする．

Step 5. 各 $j \in \{1, \ldots, m\}$ について， $A_j := E_j$ とし，各 $i \in A_j$ について $C(i) = d_i$ とする．各 $i \in \{1, \ldots, n\} \setminus \{E_j \mid j \in \{1, \ldots, m\}$ は，$\max\{d_i \mid i \in \{1, \ldots, n\}\}$ 以降に任意の時刻，任意の順で処理されるものとし，同一機械に割り当てられた複数のジョブの処理区間が重複しないようにスケジュールする．

Step 6. (A, C) を出力する．

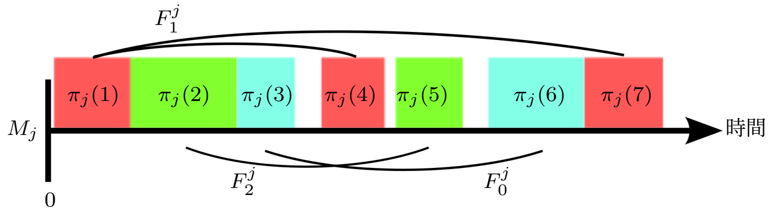

図 4.1: $k=2$ の場合の F^j

上記の手順によって得られたスケジュール (A,C) が実行可能であることを示す． Step 5. より，各ジョブの完了時刻は納期以降であることから，各ジョブは実行可能性の条件，$C(i) \geq d_i - \alpha_i$ を満たす．

次に，ジョブの処理区間の重複について議論する．Step 5. において，いずれの E_j にも含まれない各ジョブは，最大納期の時刻以降に，同一機械に割り当てられた複数のジョブの処理区間が重複しないようにスケジュールされる．ここでは， Step 5. において完了時刻を納期に設定される各ジョブについて，議論する．各機械 $j \in \{1,\ldots,m\}$ について，(A,C) において機械 j に割り当てられたジョブのうち，完了時刻が納期と等しい各ジョブ i は ($i \in A_j$ かつ $C(i) = d_i$)，スケジュール (A^α, C^α) について $i \in \mathcal{Q}(A^\alpha, C^\alpha)$ であり，$i \in A_j^\alpha$ である． (A,C) において j に割り当てられたジョブのうち，完了時刻が納期と等しい任意の異なる二つのジョブ i と $\hat{i}$ について考える ($i, \hat{i} \in A_j$ ， $i \neq \hat{i}$ ，かつ $C(i) = d_i$ であり $C(\hat{i}) = d_{\hat{i}}$)． Step 3. と Step 4. により， (A^α, C^α) において，ジョブ i とジョブ $\hat{i}$ の間には，少なくとも k 個のジョブが存在する (図 4.1)．また， (A^α, C^α) の実行可能性より， $C(i) \geq C^\alpha(i)$ であり， $C(\hat{i}) \geq C^\alpha(\hat{i})$ である．各ジョブの納期ズレ幅が $\alpha_i \leq k \min_{i'}\{p_{i'}\}$ を満たし，i と $\hat{i}$ の間には少なくとも k 個のジョブが存在することから， Step 5. において各ジョブの完了時刻を納期にずらしたとしても i と $\hat{i}$ の処理区間は重複しない．よって， (A,C) は実行可能である．

Step 4. と Step 5. より，スケジュール (A,C) において納期以前に処理を完了するジョブの完了時刻は納期と等しい．このことから， (A,C) は (P, α^0, D, W) を入力とする SJIT の実行可能解の一つであり， SJIT に対する最適なスケジュール (A^0, C^0) について以下を満たす．

$$\varphi(A,C) \leq \varphi(A^0, C^0). \tag{4.1}$$

Step 4. より，各 $j \in \{1,\ldots,m\}$ について， (A,C) において納期以前に処理を完了するジョブの荷重和と (A^α, C^α) において納期以前に処理を完了するジョブの荷重和は，以下の関係を満たす．

$$\sum_{i \in \mathcal{Q}(A,C): i \in A_j} w_i \geq \frac{1}{k+1} \sum_{i \in \mathcal{Q}(A^\alpha, C^\alpha): i \in A_j} w_i.$$

このことから，

$$
\begin{aligned}
\varphi(A,C) &= \sum_{i\in\mathcal{Q}(A,C)} w_i \\
&= \sum_{j\in\{1,\ldots,m\}} \left(\sum_{i\in\mathcal{Q}(A,C):i\in A_j} w_i \right) \\
&\geq \sum_{j\in\{1,\ldots,m\}} \left(\frac{1}{k+1} \sum_{i\in\mathcal{Q}(A^\alpha,C^\alpha):i\in A_j} w_i \right) \\
&= \frac{1}{k+1} \sum_{j\in\{1,\ldots,m\}} \left(\sum_{i\in\mathcal{Q}(A^\alpha,C^\alpha):i\in A_j} w_i \right) \\
&= \frac{1}{k+1}\varphi(A^\alpha,C^\alpha). \qquad (4.2)
\end{aligned}
$$

式 (4.1) と式 (4.2) より以下が成り立ち，

$$\frac{1}{k+1}\varphi(A^\alpha,C^\alpha) \leq \varphi(A,C) \leq \varphi(A^0,C^0),$$

よって，JIT (P,α,D,W) によって得られる解 (A^0,C^0) は以下を満たす.

$$\frac{\varphi(A^0,C^0)}{\varphi(A^\alpha,C^\alpha)} \geq \frac{1}{k+1}.$$

■

よって，APPROXIMATE は SRTDD に対する近似解法である．任意の SRTDD の入力と，対応する正整数 k に対して同様の議論が成り立つため，APPROXIMATE は，納期ズレ幅の最大値が $0 < \max_{i'}\{\alpha_{i'}\} \leq \min_{i'}\{p_{i'}\}$ を満たす SRTDD， $\cdots$， $(k-1)\min_{i'}\{p_{i'}\} < \max_{i'}\{\alpha_{i'}\} \leq k\min_{i'}\{p_{i'}\}$ を満たす SRTDD それぞれに対する，定数近似解法である.

近似解法のタイトな例

ここでは，開発した近似解法におけるタイトな例を示すことで，開発した近似解法の近似率は，これ以上改良できないことを示す.

k を任意の正整数とする．与えられた k について，次のような $(P,\alpha,D,W)_k$ を入力とする SRTDD について考える．m 台の同一機械で， $m(k+1)$ 個のジョブ， $J_1,\ldots,J_{m(k+1)}$ を処理する．ジョブの処理時間を等しく 1 とし，各ジョブの納期ズレ幅を等しく k ，納期を等しく $k+1$ ，荷重を等しく 1 とする (各 $i\in\{1,\ldots,m(k+1)\}$ について， $p_i=1$ ， $\alpha_i=k$ ， $d_i=k+1$ ， $w_i=1$)．このような (P,α,D,W) は，最小処理時間が 1 であ

り，最大納期ズレ幅が k であることから， $(k-1)\min_i\{p_i\} < \max_i\{\alpha_i\} \leq k\min_i\{p_i\}$ を満たす．

$(P,\alpha,D,W)_k$ を入力とする SRTDD について， (A^α, C^α) を次のように設定する．各 $j \in \{1,\ldots,m\}$ について， $A^\alpha_j = \{(j-1)(k+1)+1,\ldots,j(k+1)\}$ とし，各 $j \in \{1,\ldots,m\}$ に割り当てられた各ジョブ $(j-1)(k+1)+i$ (ただし， $i \in \{1,\ldots,k+1\}$) の完了時刻を $C^\alpha\Big((j-1)(k+1)+i\Big) = i$ とする．各 $j \in \{1,\ldots,m\}$ について，各ジョブの処理時間が 1 であることから，機械 j に割り当てられた $k+1$ 個のジョブの処理区間は重複しない．また，各ジョブの納期ズレ幅は $\alpha_i = d_i - p_i = k$ であり，各ジョブの完了時刻は $C^\alpha(i) \geq 1$ であることから，各ジョブの完了時刻は $C^\alpha(i) \geq d_i - p_i$ を満たす．このことから，(A^α, C^α) は実行可能である．また，各ジョブの納期は $k+1$ であることから， (A^α, C^α) において全てのジョブが納期以前に処理を完了する． (A^α, C^α) は最適なスケジュールの一つであり，目的関数の値は， $\varphi(A^\alpha, C^\alpha) = m(k+1)$ である．

ここで， $(P,\alpha^0,D,W)_k$ を入力とする SJIT について， (A,C) を次のように設定する．各 $j \in \{1,\ldots,m\}$ について， $A_j = \{(j-1)(k+1)+1,\ldots,j(k+1)\}$ とし，各 $j \in \{1,\ldots,m\}$ に割り当てられたジョブのうち，添え字が最大であるジョブの完了時刻を $C\Big(j(k+1)\Big) = d_{j(k+1)} = k+1$ とし， j に割り当てられたそのほかの各 $(j-1)(k+1)+i$ (ただし $i \in \{1,\ldots,k\}$) について，完了時刻を $C\Big((j-1)(k+1)+i\Big) = k+1+i$ とする．各ジョブの処理時間が 1 であることから， (A,C) においてジョブの処理区間は重複しない．また，各ジョブ $i \in \{1,\ldots,m(k+1)\}$ の納期ズレ幅が 0 であり，各ジョブの完了時刻が $C(i) \geq k+1$ であることから，各ジョブは $C(i) \geq d_i - p_i$ を満たす．以上より， (A,C) は実行可能であり，目的関数の値は $\varphi(A,C) = m$ である．全ての納期が等しく $k+1$ であることから，実行可能なスケジュールにおいて納期丁度に処理を完了するジョブは高々 m ジョブであることから， (A,C) は最適なスケジュールの一つである．APPROXIMATE がかえすスケジュールを (A^0, C^0) とすると， (A^0, C^0) は $(P,\alpha^0,D,W)_k$ における最適なスケジュールの一つであり，目的関数の値は $\varphi(A^0, C^0) = m$ である．

$(P,\alpha,D,W)_k$ を入力とする SRTDD の最適解 (A^α, C^α) と，$(P,\alpha^0,D,W)_k$ を入力とする SJIT の最適解 (A^0, C^0) について，以下が満たされるため，与えられた k について，$(P,\alpha,D,W)_k$ は開発した近似解法におけるタイトな例である．

$$\frac{\varphi(A^0,C^0)}{\varphi(A^\alpha,C^\alpha)} = \frac{m}{m(k+1)} = \frac{1}{k+1}.$$

任意の正整数 k に対して前述の議論が成り立つため，開発した近似解法の近似率をよりよくすることはできない．

第5章　JIT スケジューリング問題の緩和

JIT 生産方式では，製造に合わせて部品を調達すること，納期丁度に商品の製造を完了することを徹底することで，在庫コストの削減を図る．しかし，“丁度に” という制約の厳しさのため， JIT 生産方式の厳密な実現は難しく，現実には若干の時間的余裕を持たせて製造しているものと考えられる．

本研究では， JIT ジョブ荷重和最大化問題 (SJIT) を納期ズレ幅に基づいて緩和した SJIT を定式化し，問題に対する厳密解法を開発した．第 5.1 節では，納期ズレ幅に基づく緩和に関連して， SRTDD におけるジョブの処理順序と納期ズレ幅の関係について述べ， SJIT の緩和問題を定式化する．

第 5.2 節では，納期ズレ幅に基づいて緩和した SJIT に対して開発した厳密解法について述べる．厳密解法の開発により， SRTDD は，入力が満たす条件によって，多項式時間解法，疑似多項式時間解法を持つことを示した．第 5.3 節では， SJIT の緩和問題の計算複雑さをまとめる．

5.1　ジョブの処理順序

厳密解法の開発においてよく用いられる発想の一つに，ジョブの処理順序を納期昇順とする，という発想がある．第 3 章で紹介した従来研究においても，入力を制限した SRTDD , 遅れジョブ荷重和最小化問題 (ST) について，ジョブの処理順序を納期昇順とする発想に基づいて開発された厳密解法が多く存在する ([8], [10], [12], [14], [16])． ST では，各ジョブを任意に処理できるため，納期以前に処理を完了するジョブを同じくするスケジュールのうち，各ジョブの処理順序が納期昇順となるスケジュールが必ず存在する．また， JIT スケジューリング問題では，納期以前に完了するジョブの完了時刻は納期と等しく，各 JIT ジョブは納期昇順で処理される． JIT スケジューリング問題に対する厳密解法においてもまた，ジョブの処理順序を納期昇順とする発想が用いられている．

ジョブの処理順序を納期昇順とする発想に基づき， SRTDD の最適なスケジュールについて考察する．納期ズレ幅が 0 よりも大きい場合，処理順序を納期昇順とすることで，

問題に対する最適なスケジュールが求まるとは限らない．例として，次を入力とする問題を考える． 3 つのジョブ J_1, J_2, J_3 を単一機械で処理する．それぞれの処理時間，納期ズレ幅，納期，荷重は以下である．

	p_i	α_i	d_i	w_i
J_1	4	3	5	2
J_2	3	3	12	2
J_3	5	4	13	1

処理順序を納期昇順とするスケジュールに限定すると，ジョブ J_2 と ジョブ J_3 の両方を納期以前に処理することはできない． J_2 の完了時刻の下界 ($C(2) \geq d_2 - \alpha_2 = 9$) に J_3 の処理時間を足した値が J_3 の納期よりも大きく ($9 + p_3 = 14 > d_3 = 13$)， J_2 をできるだけ手前で処理したとしても， J_3 の完了時刻が納期を超えるためである．このことから，処理時間を納期昇順とするスケジュールに限定すると，限定したスケジュールのうち荷重和が最大となるスケジュールの一つは，$A = (A_1) = (\{1, 2, 3\})$ ，$C(1) = 5$ ，$C(2) = 12$ ， $C(3) = 18$ である．明らかに (A, C) は実行可能であり，納期以前に処理を完了するジョブの集合は $\mathcal{Q}(A, C) = \{1, 2\}$ であることから，(A, C) における目的関数の値は，$\varphi(A, C) = w_1 + w_2 = 4$ である．ここで，処理順序を納期昇順とするスケジュールに限定せずに問題のスケジュールを考える．スケジュール (A^*, C^*) を， $A^* = (A_1^*) = (\{1, 2, 3\})$ ，$C^*(1) = 4$ ， $C^*(2) = 9$, $C^*(3) = 12$ とする．明らかに (A^*, C^*) は実行可能であり，納期以前に処理を完了するジョブの集合は $\mathcal{Q}(A^*, C^*) = \{1, 2, 3\}$ であることから， (A^*, C^*) における目的関数の値は，$\varphi(A^*, C^*) = w_1 + w_2 + w_3 = 5$ である．よって，$\varphi(A^*, C^*) > \varphi(A, C)$ であり，処理順序を限定したスケジュールにおける荷重和最大なスケジュールは，SRTDD の最適なスケジュールであるとは限らない．

Kawamata, Morita と Sung [7] による共同研究によって，任意の実行可能なスケジュールにおいて処理順序が納期昇順であることと，納期ズレ幅が以下の条件をみたすことが同値であることを示した．

$$\alpha_i \leq \min_{i'}\{p_{i'}\}.$$

補題 8 (A, C) を，各納期ズレ幅 α_i が $\alpha_i \leq \min_{i'}\{p_{i'}\}$ を満たす *SRTDD* における任意の実行可能なスケジュールとする．

任意の機械 $j \in \{1, \ldots, m\}$ と，機械 j に割り当てられたジョブのうち，納期以前に処理を完了する任意の異なるジョブ i, k について *(* $i, k \in A_j$ ， $i, k \in \mathcal{Q}(A, C)$ かつ $i \neq k$ *)*,

- 処理順序は納期昇順である *(* $C(i) \leq C(k) \Rightarrow d_i \leq d_k$ *)*.

特に， $d_i = d_k$ ならば $p_i \geq p_k$ であり， $d_i = d_k$ かつ $p_i = p_k$ ならば $\alpha_i \geq \alpha_k$ である．

証明 (A, C) を実行可能なスケジュールとし， $j \in \{1, \ldots, m\}$ を任意の機械とする．

i, k を機械 j に割り当てられたジョブのうち ($i, k \in A_j$)， $C(i) \leq C(k)$ を満たす任意の異なる納期以前に処理を完了するジョブとする ($i, k \in \mathcal{Q}(A, C)$ かつ $i \neq k$)． i が納期以前に処理を完了することから， i の完了時刻は以下を満たす．

$$d_i - \alpha_i \leq C(i) \leq d_i.$$

(A, C) の実行可能性より，ジョブ k はジョブ i の処理完了後に処理を開始するため，

$$d_i - \alpha_i \leq C(i) \leq C(k) - p_k.$$

ジョブ k もまた，納期以前に処理を完了することから， $C(k) \leq d_k$ であり，

$$d_i - \alpha_i \leq C(i) \leq C(k) - p_k \leq d_k - p_k.$$

このことから， $d_i - \alpha_i \leq d_k - p_k$ であり，

$$d_k - d_i \geq p_k - \alpha_i.$$

納期ズレ幅の前提より， $\alpha_i \leq \min_{i'}\{p_{i'}\} \leq p_k$ であることから，

$$d_k - d_i \geq p_k - \alpha_i \geq 0. \tag{5.1}$$

したがって $d_i \leq d_k$ である．

さらに， i, k が $d_i = d_k$ を満たす場合，(5.1) より， $p_k - \alpha_i = 0$ である．各納期ズレ幅は最小の処理時間以下であることから，

$$p_k = \alpha_i \leq \min_{i'}\{p_{i'}\} \leq p_i.$$

よって， $p_i \geq p_k$ である．

さらに， $p_k - \alpha_i = 0$ であることから， $\alpha_i = p_k = \min_{i'}\{p_{i'}\}$ である．納期ズレ幅の前提より，

$$\alpha_k \leq \min_{i'}\{p_{i'}\} = \alpha_i.$$

■

補題 8 により，各納期ズレ幅 α_i が $\alpha_i \leq \min_{i'}\{p_{i'}\}$ を満たす場合，任意の実行可能なスケジュールにおいて，ジョブの処理順序は納期昇順であることが明らかとなった．このことから，ジョブの処理順序は納期昇順であるスケジュールについて，荷重和が最大となるスケジュールを求めることで，最適解が求まる．

補題 8 により，納期ズレ幅 α_i が $\alpha_i \leq \min_{i'}\{p_{i'}\}$ を満たす任意の SRTDD について，任意の実行可能なスケジュールにおけるジョブの処理順序が納期昇順であることを示した．ここで， $\alpha_i > \min_{i'}\{p_{i'}\}$ を満たすあるジョブ i が存在する場合，任意の実行可能なスケジュールにおいてジョブの処理順序が納期昇順であるとは限らないことを示す． n 個のジョブを単一機械で処理するものとし (ただし， n は 2 以上の整数である)，次のように設定した (P, α, D, W) について考える．全てのジョブの処理時間は 1 とし，ジョブ i をのぞく $n-1$ 個のジョブの納期ズレ幅を 0 とする．また，ジョブ i の納期ズレ幅 α_i を $\alpha_i = \min_{i'}\{p_{i'}\} + k$ とする (ただし， k は正整数である)．ジョブ i をのぞく $n-1$ 個のジョブの納期を正整数 d とし，ジョブ i の納期を $d_i = d + 1$ とする．ここで，ジョブ i の完了時刻を $d-1$ とし，i を除く $n-1$ 個のジョブから任意の一つを選び，完了時刻を d とする．残りの $n-2$ 個のジョブは，$d+1$ 以降，任意の順で，処理区間が重複しないように完了時刻を設定する．明らかに，このようなスケジュールは実行可能であり，納期以前に処理を完了するジョブの処理順序は，納期昇順ではない．

納期ズレ幅の値が $\alpha_i > \min_{i'}\{p_{i'}\}$ を満たすジョブ i が存在する場合，任意の実行可能なスケジュールにおいて処理順序が納期昇順であるとは限らない．また，冒頭で示した反例より，そのような問題について処理順序を納期昇順とするスケジュールに限定したとしても，問題に対する最適なスケジュールが求まるとは限らない (ただし， ST ，つまり，各納期ズレ幅 α_i が $\alpha_i = d_i - p_i$ である場合をのぞく)．

前述の議論より，本研究ではジョブの処理順序を納期昇順とする発想に基づき，納期ズレ幅が $\alpha_i \leq \min_{i'}\{p_{i'}\}$ を満たす問題について扱う．補題 8 に基づき，各ジョブの添え字を，納期昇順を優先し，処理時間降順，納期ズレ幅降順の順に優先して添え字をふりなおす．以降，

$$d_1 \leq d_2 \leq \cdots \leq d_n = \max_i\{d_i\}$$

とし，$d_i = d_{i+1}$ を満たす各ジョブ $1 \leq i < n$ について， $p_i \geq p_{i+1}$ とし， $d_i = d_{i+1}$ かつ $p_i = p_{i+1}$ を満たす各ジョブ $1 \leq i < n$ について， $\alpha_i \geq \alpha_{i+1}$ とする．このような添え字の振り直しには整列アルゴリズムを用い，その計算時間は $\mathrm{O}(n \log n)$ である．

疑似 JIT ジョブ荷重和最大化問題

JIT 生産方式では，“必要なものを，必要なときに，必要なだけ” 調達し，製造する．商品の製造に合わせて部品を調達すること，納期丁度に商品の製造を完了することを徹底することで在庫コストの削減を図る．自社内での徹底はもちろん，下請けの企業に対してもまた，納入期限に関する厳しい要求をするものであり， JIT 生産方式の厳密な実現は難しい．また，部品の納入の遅延が以降の工程に与える影響が大きい，などの問題点も挙げられている．これらのことから，現実には各納期に若干の時間的余裕を持たせて，製造していると考えられる．

本研究では， JIT スケジューリング問題における，“納期丁度に” という制約を，納期ズレ幅に基づいて緩和した問題を扱う．共同研究で導出した条件に基づき，納期ズレ幅の上界を 0 から最小の処理時間に緩和し，納期以前に処理を完了するジョブの荷重和最大化を目的とする問題を，疑似 JIT ジョブ荷重和最大化問題と呼ぶ．つまり，疑似 JIT ジョブ荷重和最大化問題とは，各納期ズレ幅 α_i が以下を満たす SRTDD である．

$$\alpha_i \leq \min_{i'}\{p_{i'}\}.$$

以降，疑似 JIT ジョブ荷重和最大化問題において，納期以前に処理を完了するジョブ，つまり，実行可能なスケジュール (A, C) において $C(i) \in [d_i - \alpha_i, d_i]$ を満たすジョブを疑似 JIT ジョブと呼ぶ．以降，疑似 JIT ジョブ荷重和最大化問題 (**S**equencing to maximize the weighted number of **P**seudo **JIT** jobs) を， SPJIT と表記する．

現実において，納期ズレ幅の上界が最小処理時間である，という条件は，保管費用の観点において有用である．例として， JIT 生産方式を目指している元請けと下請けの関係にある複数の企業の，下流の企業について考える．下流企業は，納期遅れによって上流に与える影響が大きく，余裕をもって各部品を製造していると仮定する．しかし，全体として JIT 生産方式を目指しているため，製造した各商品を納期に発送する．この例において，納期ズレ幅が最小の処理時間を超える場合，ある時刻において発送待ちの商品が 2 つ以上存在する可能性がある．機械の台数が多ければ多いほど，発送待ちの商品の数も増えるため，商品の保管費用がかさみ，好ましくない．納期ズレ幅の上界を最小の処理時間とする場合，ある時刻において発送待ちの商品は，高々 2 つである．発送待ちの商品が 2 つになる時刻が存在するとしても，その場合，それら 2 つの商品の納期は等しく，新しく発送待ちとなる 2 つ目の商品の製造の完了とともに， 2 つの商品は発送される．並列機械についても，機械数を m とすると，ある時刻で発送待ちである商品数は高々 $2m$ である．

JIT ジョブ荷重和最大化問題 (SJIT) は， SPJIT において各納期ズレ幅が 0 である問題に対応する．第 3.3 節より， SJIT は，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** 困難であることから (Sung & Vlach [15]，付録 A)， SPJIT もまた，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． SJIT では，各 JIT ジョブの完了時刻は一意に決まる．しかし， SPJIT では，異なる二つのジョブを疑似 JIT ジョブとして処理できるか否かは，それまでのスケジュールによる影響を受けるため， SJIT の厳密解法は， SPJIT の厳密解法には成り得ない．

本研究では， SRTDD について，計算複雑さが明らかではない問題のうち，SPJIT を優先的に扱い，機械モデルは，同一並列機械モデルを扱う．

5.2 厳密解法の開発

この節では，第 5.1 節で提案した SPJIT に対して開発した厳密解法について述べる．

第 5.1 節で示したジョブの処理順序に関する性質に基づき，この節では，一般性を失うことなく，実行可能なスケジュールのうち，特徴づけたスケジュールを扱う．厳密解法の提案に先立ち，スケジュールの特徴付けとその記述について述べる．

スケジュールの記述

(A, C) を任意の実行可能なスケジュールとする．納期ズレ幅が 0 よりも大きいあるジョブが存在する場合，割り当てと，疑似 JIT ジョブの集合を同じくする， (A, C) とは異なる実行可能なスケジュール (A, C') が存在する場合がある ($\mathcal{Q}(A, C) = \mathcal{Q}(A, C')$)．目的関数の議論より，疑似 JIT ジョブでない各ジョブは，目的関数の値に影響を与えない．以降，疑似 JIT ジョブでない各ジョブは，各疑似 JIT ジョブの処理の完了後，任意の機械，任意の順で処理するものとする．

以下では，疑似 JIT ジョブに着目し，疑似 JIT ジョブの割り当てと，疑似 JIT ジョブの集合を同じくする複数の実行可能なスケジュールに関して議論する．

(A, C) を実行可能なスケジュールとする．割り当て A について，各 Q_j ($j \in \{1, \ldots, m\}$) を，機械 j に割り当てられた各ジョブ $i \in A_j$ のうち，疑似 JIT ジョブであるジョブの集合とし ($Q_j = \{i \in A_j \mid i \in \mathcal{Q}(A, C)\}$)，まとめて $Q = (Q_1, \ldots, Q_m)$ と表す．補題 8 より，同じ機械に割り当てられた疑似 JIT ジョブの添え字の昇順は， (A, C) における完了時刻の昇順を与える．各 Q_j ($j \in \{1, \ldots, m\}$) について， $\sigma_j^Q : \{1, \ldots, |Q_j|\} \to Q_j$ を Q_j

の要素の昇順を表す関数とする．つまり，$\{\sigma_j^Q(i) \mid i \in \{1, \ldots, |Q_j|\} = Q_j$ であり，かつ

$$\sigma_j^Q(1) < \cdots < \sigma_j^Q(|Q_j|).$$

各 Q_j について，$\sigma_j^Q(i)$ ($i \in Q_j$) は，Q_j の要素の添え字の昇順において，i 番目の要素を表す.

(A, C) における疑似 JIT ジョブの集合の集まり Q と，Q について添え字の昇順を表す σ^Q について，実行可能性を守りながら各ジョブを手前に詰めた際の完了時刻を C^Q とする．任意の $j \in \{1, 2, \ldots, m\}$ と任意の $i \in \{1, 2, \ldots, |Q_j|\}$ について，

- $i = 1$ の場合，$C_Q(\sigma_j^Q(1)) = \max\{d_{\sigma_j^Q(1)} - \gamma_{\sigma_j^Q(1)}, p_{\sigma_j^Q(1)}\}$,
- $1 < i \leq |Q_j|$ の場合，$C_Q(\sigma_j^Q(i)) = \max\{d_{\sigma_j^Q(i)} - \gamma_{\sigma_j^Q(i)}, C_Q(\sigma_j^Q(i-1)) + p_{\sigma_j^Q(i)}\}$.

明らかに (Q, C_Q) は実行可能なスケジュールである．さらに，(A, C) の実行可能性より，(A, C) における各疑似 JIT ジョブは，(Q, C_Q) においてもまた，疑似 JIT ジョブである．以降，$Q = (Q_1, \ldots, Q_m) \subseteq \{1, \ldots, n\}^m$ のうち，Q における各 Q_j が互いに素である Q を (各 $1 \leq j < j' \leq m$ について $Q_j \cap Q_{j'} = \emptyset$)，部分割り当てと呼ぶ．また，C_Q の定義より，任意の部分割り当て Q について，(Q, C_Q) は実行可能なスケジュールである.

任意の実行可能なスケジュール (A, C) について，(A, C) における疑似 JIT ジョブの割り当てを Q とする実行可能なスケジュール (Q, C_Q) が存在することから，前述のように特徴付けたスケジュールのみを扱うとしても，一般性を失うことはない．以上の議論より，SPJIT は，部分割り当て Q について，何れかの Q_j に現れるジョブが全て疑似 JIT ジョブである Q のうち ($\bigcup_{j=1}^m \{Q_j\} = \mathcal{Q}(Q, C_Q)$)，疑似 JIT ジョブの荷重和 $\varphi(Q, C_Q)$ を最大とする Q を求める問題と等価である．以降この章では，そのような部分割り当て Q を求める.

5.2.1 動的計画法に基づくアプローチ

J_0 をダミージョブとし，J_0 についての各入力は 0 とする ($p_0 = \alpha_0 = d_0 = w_0 = 0$).

$v \in (\{0, \ldots, n\} \times \{0, \ldots, d_n\})^m$ を各機械の制限を集めた n 次元ベクトルとする．各 $v_j = (u_j, t_j)$ ($j \in \{1, \ldots, m\}$) は機械 j が扱うジョブと処理区間の制限を表す．スケジュール (Q, C_Q) において機械 j が制限 v を満たすとは，機械 j に割り当てるジョブは，集合 $\{1, \ldots, u_j\}$ の要素であり，機械 j がジョブを処理する区間が $[0, t_j]$ であることを表す ($Q_j \subseteq \{1, \ldots, u_j\}$ ，$\max_i\{C_Q(i) \mid i \in Q_j\} \leq t_j$).

$v \in (\{0,\ldots,n\} \times \{0,\ldots,d_n\})^m$ を任意の制限とする．スケジュール (Q, C_Q) において，すべての機械 j が制限 v を満たし，かつ Q に現れるすべてのジョブが (Q, C_Q) において疑似 JIT ジョブである部分割り当て Q を v *-bounded* と呼ぶ．つまり，v*-bounded* である Q は，各 $j \in \{1,\ldots,m\}$ について，$Q_j \subseteq \{1,\ldots,u_j\}$ かつ $\max_{i\in Q_j}\{C_Q(i)\} \le t_j$ であり，$\bigcup_{j=1}^{m}\{Q_j\} = \mathcal{Q}(Q, C_Q)$ を満たす (図 5.1)．

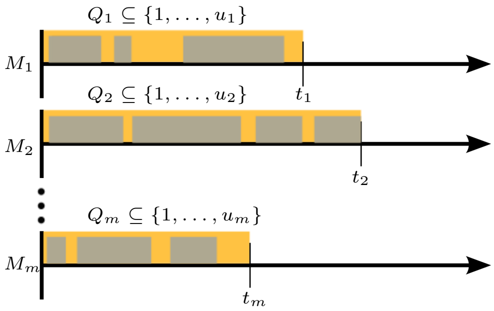

図 5.1: v-bounded である Q

また，v*-bounded* である Q のうち，(Q, C_Q) における疑似 JIT ジョブの荷重和 ($\varphi(Q, C_Q)$) が最大である Q を v*-optimal* と呼び，Q^v と表す．以降，任意の v について，(Q^v, C_Q^v) における疑似 JIT ジョブの荷重和を $\textsc{opt}(v)$ と表す ($\textsc{opt}(v) = \varphi(Q^v, C_Q^v)$)．

本研究で開発した疑似多項式時間解法は，動的計画法に基づく．動的計画法とは，元の問題を部分問題に分け，小さな部分問題から少しずつ問題を広げつつ，それぞれの部分問題を解くことで，元の問題の最適解を求める手法である．それぞれの部分問題の解は，それまでの部分問題の解を用いて求める．

v-部分問題 与えられた $v \in (\{0,\ldots,n\} \times \{0,\ldots,d_n\})^m$ について，v*-optimal* である部分割り当て Q を求める．

特に，各 $v_j = (u_j, t_j)$ について，$u_j = n$ かつ $t_j = d_n = \max_i\{d_i\}$ である場合，v-部分問題の解は，元の問題の最適解に対応する．

正当性の証明

動的計画法では，部分問題の解を，それまでの部分問題の解を用いて解く．動的計画法に基づくアルゴリズムの正当性の論点は，それまでの部分問題の解を用いることで対象とする部分問題の解が正しく求まるのかどうか，である．以下では，任意の v-部分問題の解 Q^v について，それまでの部分問題の解との関係を示す．

$v \in (\{0,\ldots,n\} \times \{0,\ldots,d_n\})^m$ について，各機械に割り当てるジョブの制限のうち，最大の添え字を $\ell(v)$ とする ($\ell(v) = \max_j\{u_j\}$)．与えられた任意の v について， ジョブ $\ell(v)$ を割り当てることができ ($u_j = \ell(v)$)，かつ $\ell(v)$ を処理区間の制限を満たしながら処理することができる ($t_j \geq \max\{d_{\ell(v)} - \alpha_{\ell(v)}, p_{\ell(v)}\}$) 機械の集合を $L(v)$ とする ($L(v) = \{j \in \{1,\ldots,m\} \mid u_j = \ell(v), t_j \geq \max\{d_{\ell(v)} - \alpha_{\ell(v)}, p_{\ell(v)}\}$).

以下，任意の部分問題の解が，それまでの部分問題の解を用いることで正しく求まることを示す． $v \in (\{0,\ldots,n\} \times \{0,\ldots,d_n\})^m$ を任意の制限とし， v-*optimal*である割り当てを Q^v とする．

$\ell(v) = 0$ **の場合** $\ell(v) = \max_j\{u_j\} = 0$ であることから， $\ell(v) = 0$ を満たす各 v について， $Q^v = (\emptyset,\ldots,\emptyset)$ である．

$\ell(v) > 0$ **の場合** 各機械 j の扱うジョブの制限， u_j に着目する．部分問題の解において，$\ell(v)$ が何れの機械にも割り当てられない部分問題を v^0 -部分問題とする．つまり，各 $j \in \{1,\ldots,m\}$ について， $v_j^0 = (u_j^0, t_j^0) = (\min\{u_j, \ell(v)-1\}, t_j)$ である (図 5.2(a))．ここで， Q^v において，ジョブ $\ell(v)$ がいずれの機械 j にも割り当てられていないと仮定する (各 $j \in \{1,\ldots,m\}$ について $\ell(v) \notin Q_j^v$)． v^0-*optimal* である Q^{v^0} について， $Q^v = Q^{v^0}$ である．

$\ell(v) \in Q_j^v$ を満たすある $j \in \{1,\ldots,m\}$ が存在すると仮定する．スケジュール (Q^v, C_Q^v) における $\ell(v)$ の完了時刻を x とする ($x = C_Q^v(\ell(v))$)． v-*bounded* の定義より， $t_j \geq x$ であり， $t_j \geq \max\{d_{\ell(v)} - \alpha_{\ell(v)}, p_{\ell(v)}\}$ であることから， Q^v において $\ell(v)$ を割り当てた機械 j は，明らかに $j \in L(v)$ を満たす．ここで， $\ell(v)$ を処理しうる各機械 $k \in L(v)$ について， $\ell(v)$ を機械 k に完了時刻 $\min\{t_k^0, d_j\}$ として割り当てることを考える．機械 k の処理区間の制限を，ジョブ $\ell(v)$ の処理の邪魔にならないように $\min\{t_k^0, d_j\} - p_{\ell(v)}$ としたベクトルを v^k とする．つまり，各 $j \in \{1,\ldots,m\}$ について， $j = k$ ならば， $v_j^k = (u_j^k, t_j^k) = (u_j^0, \min\{t_j^0, d_j\} - p_{\ell(v)})$ であり，そうでなければ， $v_j^k = v_j^0$ である (図 5.2 (b))．

Q^v について， $\ell(v)$ が割り当てられた機械を k とする ($\ell(v) \in Q_k^v$)． Q^v と k について，次のように Q を設定する．各 $j \in \{1,\ldots,m\}$ について， $j = k$ ならば $Q_j = Q_j^v \setminus \{\ell(v)\}$ とし，そうでなければ $Q_j = Q_j^v$ とする．ジョブの処理順序の性質 (補題 8) より， $\ell(v) \in Q_k^v$ であるスケジュールにおいて，$\{\ell(v)+1,\ldots,n\}$ のジョブが $\ell(v)$ よりも手前に機械 k に現れることはなく， v の定義より， k 以外の機械に

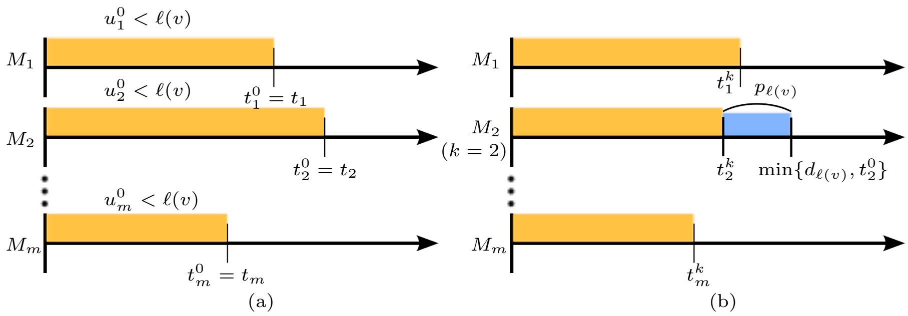

図 5.2: v について，v^0 と， $k = 2$ とした v^k ($k \in L(v)$)

割り当てられたジョブは $\{1, \ldots, \ell(v) - 1\}$ の要素である．よって， Q^v が v-*optimal* であることから， Q は v^k-*optimal*であり， $\textsc{opt}(v) = \textsc{opt}(v^k) + w_{\ell(v)}$ である．

以上の議論より， v-部分問題の解 Q^v は， $\ell(v') < \ell(v)$ を満たし，各 j について $t'_j \leq t_j$ を満たす各 v' に対する， v'-部分問題の解を用いて求めることができる．

解法

$\textsc{optimal}(v)$ を，与えられた n 次元ベクトル v について， v -*optimal*である Q^v を求めるアルゴリズムとする．

OPTIMAL

Instance n 次元ベクトル v

Output v -*optimal*である Q^v

- $\ell(v) = 0$ の場合

  $\ell(v)$ の定義より，各 $j \in \{1, \ldots, m\}$ について $u_j = 0$ である．各 $j \in \{1, \ldots, m\}$ と $d_0 - \alpha_0 \leq t \leq d_0$ について， $Q_j^v(u_j, t_j) = Q_j^v(0, 0) = \emptyset$.

- $\ell(v) > 0$ の場合

  $\gamma_0 = \textsc{opt}(v^0)$ とする．

  各 $k \in L(v)$ について，機械 k にジョブ $\ell(v)$ を完了時刻を $\min\{d_{\ell(v)}, t_k^0\}$ 以下として割り当てたスケジュールの荷重和を γ_k とする (つまり， $\gamma_k = \textsc{opt}(v^k) + w_{(\ell(v))}$ ，図 5.2(b))． v において $\ell(v)$ をどの機械にも割り当てないスケジュールの荷重和 γ_0 とこれらの γ_k ($k \in L(v)$) を比べ，値が最大値となる $k' \in \{0\} \cup L(v)$ について， $\textsc{opt}(v) = \gamma_{k'}$ とする．

k' に基づき，v -部分問題の解を以下のように設定する．

– $k' = 0$ である場合

  $\ell(v)$ を何れの機械にも割り当てない．各 $j \in \{1, \ldots, m\}$ について，$Q_j^v := Q_j^{v^0}$

– $k' \neq 0$ である場合

  $\ell(v)$ を機械 k' に割り当てる．各 $j \in \{1, \ldots, m\}$ について，$j = k'$ ならば $Q_j^v := Q_j^{v^{k'}} \cup \{\ell(v)\}$ とし，$j \neq k'$ ならば $Q_j^v := Q_j^{v^{k'}}$ とする．

各 $j \in \{1, \ldots, m\}$ について $u_j = n$ かつ $t_j = d_n$ を満たす v について，v-*optimal*である Q^v（元の問題の最適解）を求めるアルゴリズムは，以下である．

Input: SPJIT の入力 (P, α, D, W)

Output: 各 $v_j = (u_j, t_j)$ について，$u_j = n$ かつ $t_j = d_n$ を満たす v について，v -*optimal*である部分割り当て Q

Step 1. $h = n$ となるまで，$h = 0$ から順に次を考える．各 h について，$g = d_n = \max_i\{d_i\}$ となるまで，$g = 0$ から順に次を考える．

- 与えられた h と g について，$\ell(v) = h$ かつ $\max_j\{t_j\} = g$ を満たすすべての v について OPTIMAL(v) によって Q^v を求める．

Step 2. 各 $j \in \{1, \ldots, m\}$ について，$u_j = n$ かつ $t_j = d_n$ を満たす v について，Q^v を出力する．

計算時間

まず，OPTIMAL(v) の計算時間を示す．与えられた v について，$\ell(v) = 0$ の場合，計算時間は各 Q_j に対して $Q_j = \emptyset$ とする処理のため，$\mathrm{O}(m)$ である．以下，$\ell(v) > 0$ と仮定する．$L(v)$ の計算時間は $\mathrm{O}(m)$ であり，v^0 と各 v^k の計算時間は，$\mathrm{O}(m)$ である．各 Q_j^v の設定における計算時間は，$\mathrm{O}(mn)$ である．以上より，$\ell(v) = 0$ の場合の計算時間 $\mathrm{O}(m)$ と，$\ell(v) > 0$ の場合の計算時間 $\mathrm{O}(m + m + mn) = \mathrm{O}(mn)$ を比較し，OPTIMAL(v) の計算時間は，$\mathrm{O}(mn)$ である．

元の問題の最適解を求めるアルゴリズムについて，$\ell(v) = h$ となるような u_j の組み合わせは高々 n^m 通りである．また，与えられた g について，各 t_j が $t_j \leq g$ を満たす t_j の組み合わせは高々 ${d_n}^m$ 通りである．OPTIMAL(v) の計算時間は $\mathrm{O}(mn)$ であること

から，Step 1. の計算時間は $\mathrm{O}(mn \cdot n^m \cdot d_n \cdot {d_n}^m) = \mathrm{O}(mn^{m+1}{d_n}^m)$ である．Step 2. について， Q^v の出力における計算時間は， $\mathrm{O}(mn)$ である．よって，この解法の計算時間は， $\mathrm{O}(mn^{m+1}{d_n}^m + mn) = \mathrm{O}(mn^{m+1}{d_n}^m)$ である．

5.2.2 本研究の提案解法

第 5.2.1 節では， SPJIT に対する動的計画法に基づくアプローチの正当性と，その計算時間を示した．ここでは，動的計画法において扱う部分問題について，納期ズレ幅に基づく性質に着目し，扱う部分問題を一般性を失うことなく制限し，本研究の提案解法とする．

部分問題の解と納期ズレ幅

動的計画法において v-部分問題の解は，それまでの部分問題の解を利用して， $\ell(v)$ を候補の機械 $k \in L(v)$ に割り当てたスケジュールと，何れの機械にも割り当てないスケジュールを比較することで得られる．この際， (Q, C_Q) において $\ell(v)$ を疑似 JIT ジョブとなるように各機械への割り当てることから，機械 j の処理区間の幅を表す t_j の値によっては，結論が明らかである場合がある．ここでは， $\ell(v)$ と t_j の関係をもとに，各部分問題の解について議論する．

$v \in (\{0, \ldots, n\} \times \{0, \ldots, d_n\})^m$ を任意の制約とする．ある機械 x ($x \in \{1, \ldots, m\}$) の t_x の値に基づき，部分問題の解について考察する．

- $t_x < \max\{d_{u_x} - \alpha_{u_x}, p_{u_x}\}$ の場合

  v-*optimal* である Q^v について， $\bigcup_{j=1}^m \{Q^v\} = \mathcal{Q}(Q^v, C_Q^v)$ であることから， $u_x \notin Q_x^v$ である ($Q_x^v \subseteq \{1, \ldots, u_x - 1\}$)．機械 M_x において割り当てるジョブを $\{1, \ldots, u_x - 1\}$ と制限した v' について，

  $$v'_j = (u'_j, t'_j) = \begin{cases} (u_j - 1, t_j) & \text{if } j = x, \\ (u_j, t_j) & \text{otherwise.} \end{cases}$$

  v-*optimal* である Q^v と， v'-*optimal* である $Q^{v'}$ は等しい．

- $t_x \geq d_{u_x} - \alpha_{u_x}$ の場合

  v-*optimal* である Q^v について， $Q_j^v \subseteq \{1, \ldots, u_j\}$ であり， Q_j^v の各要素 $i \in Q_j^v$ の納期 d_i は d_{u_j} 以下である ($d_i \leq d_{u_j}$)． (Q^v, C_Q^v) の実行可能性と $\bigcup_{j=1}^m \{Q_j^v\} = \mathcal{Q}(Q^v, C_Q^v)$

であることから，各 $i \in Q_j^v$ の (Q^v, C_Q^v) における完了時刻は， $C_Q^v(i) \leq d_{u_j}$ を満たす．このことより， v における機械 x の処理可能区間の幅 t_x について，各ジョブ $i \in Q_x^v$ の完了時刻は， t_x 以下であり，かつ d_{u_x} 以下である ($\max\{C_Q(i) \mid i \in Q_x^v\} \leq \min\{t_x, d_{u_x}\}$)．機械 x において処理可能区間の幅の制限を $\min\{t_x, d_{u_x}\}$ とした v' について，

$$v'_j = (u'_j, t'_j) = \begin{cases} (u_j, \min\{t_j, d_{u_j}\}) & \text{if } j = x, \\ (u_j, t_j) & \text{otherwise.} \end{cases}$$

v-*optimal*である Q^v と， v'-*optimal*である $Q^{v'}$ は等しい (図 5.3)．

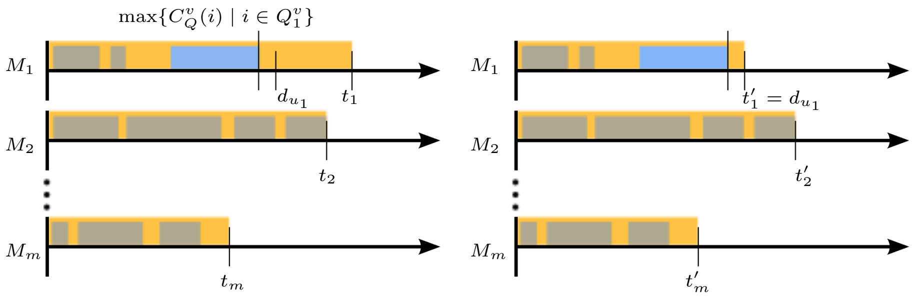

図 5.3: v について， $t_1 > d_{u_1}$ である場合の (Q^v, C_Q^v) と $(Q^{v'}, C_Q^{v'})$

任意の機械について，以上の議論が成り立つ．このことから，元の問題の最適解を求めるアルゴリズムにおいて， g の取りうる値の範囲を $\max\{d_h - \alpha_h, p_h\} \leq g \leq d_h$ とし，与えられた h と g について， $\ell(v) = h$ であり，かつ各 $j \in \{1, \ldots, m\}$ が $t_j \geq \max\{d_h - \alpha_h, p_h\}$, $\max_j\{t_j\} = g$ を満たす v についてのみ，部分問題の解を求めれば十分である．

以下では，求める部分問題を上記のように限定することに合わせて， v^0-部分問題と， v^k-部分問題について， v^0 と v^k の記述について議論する．

任意のジョブの添え字 $i \in \{1, \ldots, n\}$ と $0 \leq t \leq d_n = \max_{i'}\{d_{i'}\}$ について，$b(i, t)$ を $\{1, \ldots, i\}$ のジョブのうち，完了時刻 t 以下で処理されうるジョブのうち，最大の添え字をもつジョブを返す関数とする ($b(i, t) = \max\{i' \in \{1, \ldots, i\} \mid \max\{d_{i'} - \alpha_{i'}, p_{i'}\} \leq t\}$)．前述の部分問題の解に関する議論より，任意の v について，各機械 j における $v_j = (u_j, t_j)$ について，以下のように v'_j を定義した v' について，

$$v'_j = (u'_j, t'_j) = (b(u_j, t_j), \min\{t_j, d_{b(u_j, t_j)}\})$$

$Q^v = Q^{v'}$ である． b を用いて， v^0 と v^k を以下のように再定義する．

任意の $v \in (\{0, \ldots, n\} \times \{0, \ldots, d_n\})^m$ と各 $j \in \{1, \ldots, m\}$ について，
$v_j^0 = (b(\min\{u_j, \ell(v) - 1\}, t_j), \min\{t_j, d_{b(\min\{u_j, \ell(v)-1\}, t_j)}\})$ とする．

任意の $v \in (\{0, \ldots, n\} \times \{0, \ldots, d_n\})^m$ と各 $j \in \{1, \ldots, m\}$ ，各 $k \in L(v)$ について，

- $j = k$ の場合

  $d_{\ell(v)} - \alpha_{\ell(v)} \leq t_j \leq d_{\ell(v)}$ であることから，ジョブ $\ell(v)$ を時刻 t_j を完了時刻として機械 k に割り当てる場合について考える． a を $\{1, \ldots, \ell(v)-1\}$ のうち $t_j - p_{\ell(v)}$ 以前に処理を完了されうるジョブのうち最大の添え字とする ($a = b(\ell(v) - 1, t_j - p_{\ell(v)})$)． a について， $v_j^k = (a, \min\{t_j - p_{\ell(v)}, d_a\})$ ．

- $j \neq k$ の場合

  $v_j^k = (b(\min\{\ell(v) - 1, u_j\}, t_j), t_j)$ ．

部分問題の解に関する前述の議論に基づき，

- 各 h について， g の取りうる値の範囲を $\max\{d_h - \alpha_h, p_h\} \leq g \leq d_h$ とする．
- OPTIMAL(v) は，再定義した v^0 と v^k に基づき，関数 b を用いてそれまでの部分問題を表す．

とし， 以下の厳密解法を提案する．

提案解法

Input: SPJIT の入力 (P, α, D, W)

Output: 各 $v_j = (u_j, t_j)$ について， $u_j = n$ かつ $t_j = d_n = \max_i\{d_i\}$ を満たす v について， v -*optimal*である部分割り当て Q

Step 1. $h = n$ となるまで， $h = 0$ から順に次を考える．各 h について， $g = d_h$ となるまで， $g = \max\{d_h - \alpha_h, p_h\}$ から順に次を考える．

- 与えられた h と g について， $\ell(v) = h$ ，各 $j \in \{1, \ldots, m\}$ について $t_j \geq \max\{d_h - \alpha_h, p_h\}$ ，かつ $\max_j\{t_j\} = g$ を満たす，すべての v について OPTIMAL(v) を用いて Q^v を求める．

Step 2. 各 $j \in \{1, \ldots, m\}$ について， $u_j = n$ かつ $t_j = d_n$ を満たす v について，Q^v を出力する．

計算時間

ここでは，本研究で提案する解法の計算時間を示す．

まず，n 次元ベクトル v が与えられたとして，$\textsc{optimal}(v)$ の計算時間を示す．与えられた v について，$\ell(v)=0$ の場合，各 Q_j に対して $Q_j=\emptyset$ とする処理のため，計算時間は $\mathrm{O}(m)$ である．以下，$\ell(v)>0$ と仮定する．$L(v)$ の計算時間は $\mathrm{O}(m)$ であり，v^0 と各 v^k の計算時間は，$b(i,t)$ によるジョブの探索を含め，$\mathrm{O}(mn)$ である．各 Q_j^v の設定における計算時間は，$\mathrm{O}(mn)$ である．以上より，$\mathrm{O}(m)$ と $\mathrm{O}(m+mn+mn)=\mathrm{O}(mn)$ を比較し，$\textsc{optimal}(v)$ の計算時間は，$\mathrm{O}(mn)$ である．

もとの問題の最適解を求めるアルゴリズムについて，$\ell(v)=h$ となるような u_j の組み合わせは高々 n^m 通りである．また，与えられた h と与えられた $\max\{d_h-\alpha_h,p_h\}\le g\le d_h$ について，各 t_j が $t_j\ge\max\{d_h-\alpha_h,p_h\}$ を満たし，かつ $\max_j\{t_j\}=g$ を満たす t_j の組み合わせは高々 $\max_i\{\alpha_i\}^m$ 通りである．$\textsc{optimal}(v)$ の計算時間が $\mathrm{O}(mn)$ であることから，Step 1. の計算時間は $\mathrm{O}(mn\cdot n^m\max_i\{\alpha_i\}^m)=\mathrm{O}(mn^{m+1}\max_i\{\alpha_i\}^m)$．Step 2. について，$Q^v$ の出力における計算時間は，$\mathrm{O}(mn)$ である．よって，提案解法の計算時間は，$\mathrm{O}(n^{m+1}m\max_i\{\alpha_i\}^m+mn)=\mathrm{O}(mn^{m+1}\max_i\{\alpha_i\}^m)$ である．

5.3 計算複雑さのまとめ

第 5.2 節では，動的計画法に基づき，厳密解法を開発した．開発では，納期ズレ幅に基づき，動的計画法において解を求める部分問題を，正当性を保ちながら限定し，開発した解法の計算時間は $\mathrm{O}(mn^{m+1}\max_i\{\alpha_i\}^m)$ である．厳密解法の提案と，その計算時間により，以下の定理が導ける．

定理 9 m が定数である場合，*SPJIT* は，疑似多項式時間解法を持つ．

さらに，提案する解法の計算時間が納期ズレ幅の上界に依存することから，以下の定理が導ける．

定理 10 m が定数であり，納期ズレ幅の最大値 $\max_i\{\alpha_i\}$ が n の多項式で表せる場合，*SPJIT* は，多項式時間解法を持つ．

ここまでの研究成果により，m が定数であり，納期ズレ幅の上界が最小の処理時間である SRTDD は，この章で提案した厳密解法によって，疑似多項式時間解法を持ち，加

えて，納期ズレ幅の上界を n の多項式で表せる場合，多項式時間解法を持つ．これらの条件を満たさない SRTDD については，入力の納期ズレ幅の値に応じた近似解が多項式時間で求まる (第 4)．したがって， SRTDD のほとんどの入力に対し，本研究で開発したいずれかの解法が適用できる．

m が定数であり，納期ズレ幅の上界が $\min_{i'}\{p_{i'}\}$ である場合，問題が疑似多項式時間解法を持つことは明らかとなったが，問題の計算複雑さは未解決である．しかし，上界の値， $\min_{i'}\{p_{i'}\}$ に関連して， SRTDD の計算複雑さについて，成果を得た．第 6 章では，$\min_{i'}\{p_{i'}\}$ に関連する SRTDD について示した計算複雑さの成果を述べる．

第6章　処理開始可能時刻付き遅れジョブ荷重和最小化問題の計算複雑さ

処理開始可能時刻付き遅れジョブ荷重和最小化問題 (SRTDD) は，強 **NP** 困難であることが知られている．本研究では，任意の納期ズレ幅を許す SRTDD に対する強 **NP** 困難の証明を見直し，納期ズレ幅の取りうる値の範囲にある制限を加えたとしても， SRTDD は依然，強 **NP** 困難であることを示した.

また，納期ズレ幅の取りうる値の範囲の制限は，第 5 章に関連するものである．第 6.1 節では，この章で扱う問題とともに，第 5 章との関係を述べる．第 6.2 節では，この章で扱う問題が強 **NP** 困難であることを示す.

6.1　納期ズレ幅の制限

第 3 章より， SRTDD は納期ズレ幅の上界が $d_i - p_i$ 未満，下界が 0 よりも大きい各問題については，問題の計算複雑さが明らかでない．これらのうち，第 5 章で述べた本研究の研究成果より，各納期ズレ幅 α_i が $\alpha_i \le \min_{i'}\{p_{i'}\}$ を満たす問題は，入力が満たす条件によって，多項式時間，疑似多項式時間で最適解が求まる．納期ズレ幅の上界を $\min_{i'}\{p_{i'}\}$ とする問題は，機械数が定数の場合，高々疑似多項式時間で最適解が求まることから，強 **NP** 困難には成り得ない．厳密解法の開発過程で，納期ズレ幅の上界を $\min_{i'}\{p_{i'}\}$ とする問題は，弱 **NP** 困難であることが予想されたが，未だ明らかでない.

上界を $\min_{i'}\{p_{i'}\}$ とする問題の計算複雑さの解明には至らなかったが，下界を $\min_{i'}\{p_{i'}\}$ とする問題について，計算複雑さを解明した．この章では，納期ズレ幅の下界を $\min_{i'}\{p_{i'}\}$ とする SRTDD について，本研究の成果を述べる.

6.2　計算複雑さの証明

納期ズレ幅の下界を $\min_{i'}\{p_{i'}\}$ とする SRTDD のうち，納期ズレ幅がある性質を満たす問題もまた，強 **NP** 困難であることを示した.

SRTDD のうち，入力 (P, α, D, W) について，納期ズレ幅が次を満たす問題を扱う．任意の $i \in \{1, \ldots, n\}$ について以下を満たす正整数 k が存在する．

$$k \min_{i'}\{p_{i'}\} \leq \alpha_i. \tag{6.1}$$

3 -PARTITION からの還元によって，この問題が強 **NP** 困難であることを示す． 3 -PARTITION の定義は以下である．

Instance: 要素数が $3h$ である集合 $X = \{x_1, \ldots, x_{3h}\}$ ，分ける幅を表す正の整数 $B \in \mathbf{Z}^+$ と，各 $x \in X$ のサイズ $s(x) \in \mathbf{Z}^+$ ．ただし，サイズは次を満たす．各 $x \in X$ について， $B/4 < s(x) < B/2$ であり，かつ $\sum_{x \in X} s(x) = hB$ である．

Question: 各 $1 \leq i \leq h$ について， $\sum_{x \in X_i} s(x) = B$ を満たす X の h-分割， $X_1, \ldots, X_h$ が存在するか？ただし，s の定義より，前述の X の h -分割において，各 X_i の要素数は $|X_i| = 3$ を満たす．

$X = \{x_1, \ldots, x_{3h}\}$ ， $B \in \mathbf{Z}^+$ と，$s : X \to \mathrm{Z}^+$ の 3 項組， (X, B, s) を任意の 3 -PARTITION の入力とする．

ここで， k を任意の正整数とする． (X, B, s) に基づき，与えられた k について SRTDD の入力を以下のように設定する． $J_1, \ldots, J_{4h-1}$ の $4h-1$ 個のジョブを，単一機械で処理する．各ジョブの処理時間，納期ズレ幅，納期を以下のように設定する．

$$p_i = \begin{cases} (k+1) \cdot s(x_i) & \text{if } 1 \leq i \leq 3h, \\ 1 & \text{if } 3h+1 \leq i \leq 4h-1. \end{cases}$$

$$d_i = \begin{cases} h(k+1)B + h - 1 & \text{if } 1 \leq i \leq 3h, \\ (i-3h)(k+1)B + i - 3h & \text{if } 3h+1 \leq i \leq 4h-1. \end{cases}$$

$$\alpha_i = \begin{cases} d_i - p_i & \text{if } 1 \leq i \leq 3h, \\ k & \text{if } 3h+1 \leq i \leq 4h-1. \end{cases}$$

各ジョブの重み w_i を 1 とする． (X, B, s) の定義より， $\min_{i'}\{p_{i'}\} = 1$ であり，各ジョブの納期ズレ幅は明らかに $k \min_{i'}\{p_{i'}\} \leq \alpha_i \leq d_i - p_i$ を満たし，このように設定した (P, α, D, W) は， k について式 (6.1) を満たす SRTDD の入力である．以降，与えられた k と (X, B, s) に基づいて前述のように設定した (P, α, D, W) を区別して $I^k_{(X,B,s)}$ と表す ($I^k_{(X,B,s)} = (P, \alpha, D, W)$)．

また，単一機械で処理を行うことから，この項ではスケジュール (A, C) を簡略化して C と表す．

以下，各 $1 \leq i \leq h$ について $\sum_{x \in X_i} s(x) = B$ を満たす X の h -分割， $X_1, \ldots, X_h$ が存在することと， $I^k_{(X,B,s)}$ を入力とする SRTDD に対して $\varphi(C) \geq 4h-1$ を満たす実行可能なスケジュール C が存在することが同値であることを示す．

補題 11 各 $1 \leq i \leq h$ について $\sum_{x \in X_i} s(x) = B$ を満たす X の h -分割， $X_1, \ldots, X_h$ が存在するならば， $I^k_{(X,B,s)}$ を入力とする *SRTDD* に対して $\varphi(C) \geq 4h-1$ を満たす実行可能なスケジュール C が存在する．

証明 (X, B, s) を任意の 3 -PARTITION の入力とし， $X_1, \ldots, X_h$ を各 $1 \leq i \leq h$ について $\sum_{x \in X_i} = B$ を満たす X の h -分割とする．

$I^k_{(X,B,s)}$ について，各 $3h+1 \leq i \leq 4h-1$ の完了時刻を，各 i の納期とする (各 $3h+1 \leq i \leq 4h-1$ について $C(i) = d_i$)．これらのジョブの処理時間は 1 であることから，$[0, \max_i\{d_i\} = h(k+1)B + h - 1]$ の区間は， $h-1$ 個のジョブによって h 個の区間に分断され，それぞれの長さは $(k+1)B$ である (図 6.1)．

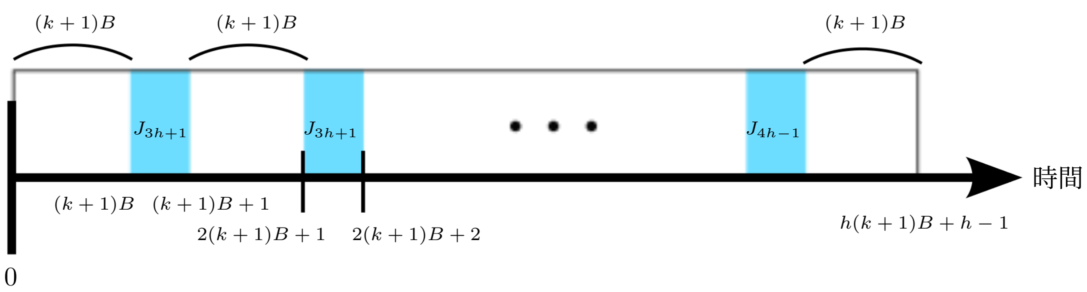

図 6.1: ジョブの配置による区間の分割

$X_1, \ldots, X_h$ について，各集合の各要素について以下のように完了時刻を設定する．各 X_i ($1 \leq i \leq h$) について， X_i の各要素を i 個目の区間， $[(i-1)(k+1)B + i - 1, i(k+1)B + i - 1]$ におさまるように完了時刻を設定することを考える．各 $1 \leq i \leq 3h$ について，$p_i = (k+1)s(x_i)$ であり， 各 X_i は $\sum_{x \in X_i} s(x) = B$ を満たすことから，各 $X_i = \{x_{i1}, x_{i2}, x_{i3}\}$ について $p_{i1} + p_{i2} + p_{i3} = (k+1)B$ である．区間の幅が $(k+1)B$ であることから， X_i の各要素は i 個目の区間で実行可能性を満たしつつ，処理することができる．ここでは，各 $X_i = \{x_{i1}, x_{i2}, x_{i3}\}$ について次のように完了時刻を設定する．
$C(i1) = (i-1)(k+1)B + i - 1 + p_{i1}$ ， $C(i2) = C(i1) + p_{i2}$ ， $C(i3) = C(i2) + p_{i3}$ ．

以上のように設定したスケジュール C は明らかに実行可能である．また，各ジョブ $1 \leq i \leq 4h-1$ は納期以前に処理を完了することから，$\varphi(C) = 4h-1$ であり， C は

$\varphi(C) \geq 4h-1$ を満たす実行可能なスケジュールである． ■

補題 12 $I^k_{(X,B,s)}$ を入力とする *SRTDD* に対して $\varphi(C) \geq 4h-1$ を満たす実行可能なスケジュール C が存在するならば，各 $1 \leq i \leq h$ について $\sum_{x \in X_i} s(x) = B$ を満たす X の h -分割， $X_1, \ldots, X_h$ が存在する．

証明 (X, B, s) を任意の 3 -PARTITION の入力とし，C を $I^k_{(X,B,s)}$ について $\varphi(C) \geq 4h-1$ を満たす実行可能なスケジュールとする．

ジョブ数が $4h-1$ であることから，すべてのジョブは C において納期以前に処理を完了する．ここで， $3h+1, \ldots, 4h-1$ のジョブに着目する．$I^k_{(X,B,s)}$ の定義より，各ジョブ i ($3h+1 \leq i \leq 4h-1$) はそれぞれ異なる納期を持ち，これらの $h-1$ 個のジョブによって，$[0, \max_i\{d_i\} = h(k+1)B + h - 1]$ の区間は分断される． $h-1$ 個のジョブがすべて納期以前に処理を完了する，任意の実行可能なスケジュールにおいて，処理時間と納期ズレ幅，納期の定義より， $3h+i$ と $3h+i+1$ ($1 \leq i < h-1$) によって切り取られる区間の幅は $(k+1)B - k$ 以上 $(k+1)B + k$ 以下の値をとる (図 6.2)．また，時刻 0 とジョブ $3h+1$ によって切り取られる区間の幅は $(k+1)B - k$ 以上 $(k+1)B$ 以下であり，ジョブ $4h-1$ と時刻 $\max_i\{d_i\}$ によって区切られる区間の幅は $(k+1)B$ 以上 $(k+1)B+k$ 以下である．このことから， $h-1$ 個のジョブによって分断された h 個のすべての区間について，区間の幅は $(k+1)B-k$ 以上 $(k+1)B+k$ 以下である．

ここで，$\varphi(C) \geq 4h-1$ を満たす実行可能なスケジュール C において，$3h+1, \ldots, 4h-1$ のジョブによって分断される h 個の区間のうち，区間の幅が $(k+1)B$ と異なるある区間が存在すると仮定し，その幅を $(k+1)B+e$ とする (e は $-k \leq e \leq k$ を満たす整数であり， $e \neq 0$ である)． C において前述の区間で処理されるジョブの処理時間の合計を $\mathcal{P}$ とすると， C の実行可能性より， $(k+1)B + e \geq \mathcal{P}$ である．すべてのジョブの処理時間が $k+1$ の倍数であり， e は $-k \leq e \leq k$ を満たす 0 以外の整数であることから， $(k+1)B + e > \mathcal{P}$ であり，区間において機械がジョブを処理しない時間が存在する．$I^k_{(X,B,s)}$ において，最大納期の値と処理時間の合計が等しいことから， $\varphi(C) \geq 4h-1$ を満たす実行可能なスケジュールにおいて，各ジョブは途切れることなく処理される．ある区間において，機械がジョブを処理しない時間が存在することは，$\varphi(C) \geq 4h-1$ を満たす C の実行可能性に矛盾する．よって，$\varphi(C) \geq 4h-1$ を満たす実行可能なスケジュール C において，$3h+1, \ldots, 4h-1$ のジョブによって分断される h 個の区間の幅は等し

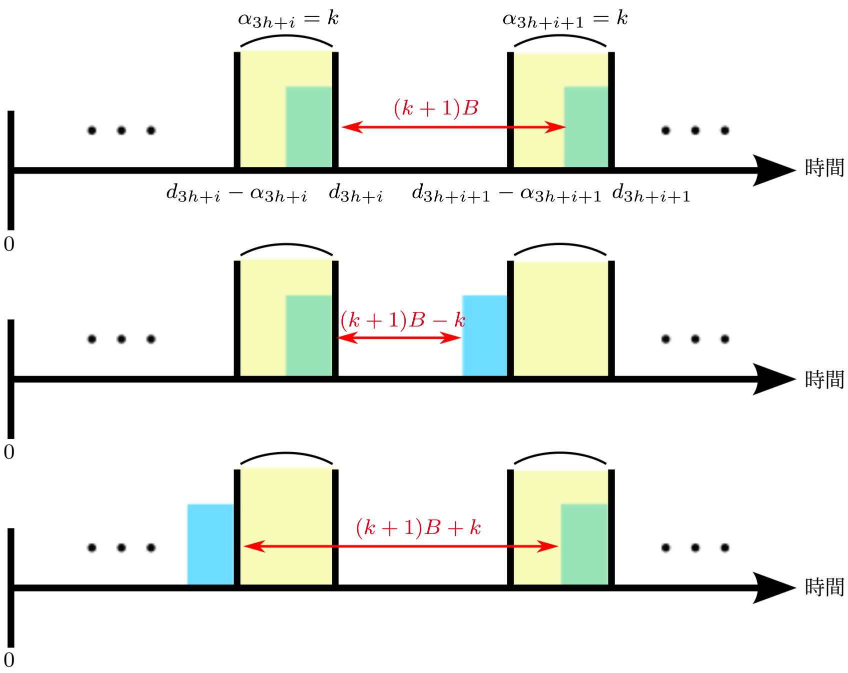

図 6.2: $3h+i$ と $3h+i+1$ によって分断された区間の幅の，上界と下界

く $(k+1)B$ である．また， s の定義より ($B/4 < s(x_i) < B/2$)，各区間で処理されるジョブ数は 3 である．

各 X_i を i 番目の区間で処理されるジョブの集合とする ($X_i = \{j \in \{1,\ldots,3h\} \mid (i-1)(k+1)B+i-1 < C(i) \leq iB+i-1\}$)，各 X_i について $\sum_{j\in X_i} p_j = (k+1)\sum_{j\in X_i} s_j = (k+1)B$ であることから， $X_1,\ldots,X_h$ は $\sum_{j\in X_i} s_j = B$ を満たす X の分割である． ■

補題 11 と補題 12 より， 各 $1 \leq i \leq h$ について $\sum_{x\in X_i} s(x) = B$ を満たす X の h -分割, $X_1,\ldots,X_h$ が存在することと， $I^k_{(X,B,s)}$ を入力とする SRTDD に対して $\varphi(C) \geq 4h-1$ を満たす実行可能なスケジュール C が存在することは，同値である．

3 -PARTITION の入力， (X,B,s) に基づく $I^k_{(X,B,s)}$ の生成に要する計算時間は，明らかに多項式時間であり，入力 $I^k_{(X,B,s)}$ の長さは，(X,B,s) の長さに関する多項式で表すことができる．したがって以下の定理が成り立つ．

定理 13 *SRTDD* のうち，入力 (P,α,D,W) について，納期ズレ幅が次を満たす場合，問題は強 **NP** 困難である．任意の $i \in \{1,\ldots,n\}$ について以下を満たす正整数 k が存在する．

$$k \min_{i'}\{p_{i'}\} \leq \alpha_i.$$

特に，任意の $i \in \{1,\ldots,n\}$ について $k\min_{i'}\{p_{i'}\} \leq \alpha_i$ を満たす正整数 k が 1 である場合，納期ズレ幅の下界を $\min_{i'}\{p_{i'}\}$ とする SRTDD である．下界を $\min_{i'}\{p_{i'}\}$ とする

問題は，強 **NP** 困難であることを示した．また，下界を $\min_{i'}\{p_{i'}\}$ とする問題が内包する問題のうち，下界を $2\min_{i'}\{p_{i'}\}$ ， $\ldots$ ， $k\min_{i'}\{p_{i'}\}$ とする各問題もまた，強 **NP** 困難であることを示した．

しかし，問題の 3 -PARTITION からの還元において， k の値が大きくなるに従い， X の要素に対応する各ジョブの処理時間，納期の値は大きくなる． k の値が大きくなるに従い，小さくなっているように見える納期ズレ幅の上界と下界の差は，それら X の要素に対応する各ジョブについては顕著に広がり，問題の本質的な難しさは未だ現れていない．SRTDD の難しさの要因の解明は，今後の課題の一つである．

また，納期ズレ幅の下界， $\min_{i'}\{p_{i'}\}$ を，上界 $d_i - p_i$ に近づけていくことで SRTDD と遅れジョブ荷重和最小化問題との，計算複雑さの境界が明らかとなるものと考える．また，納期ズレ幅の上界， $d_i - p_i$ を下界，$\min_{i'}\{p_{i'}\}$ に近づけていくことで，納期ズレ幅の上界を $\min_{i'}\{p_{i'}\}$ とする問題の計算複雑さの解明につながるものと考える．

第7章　結論

7.1　研究成果

処理開始可能時刻付き遅れジョブ荷重和最小化問題 (SRTDD) とは，各ジョブの処理開始可能時刻を制約とし，納期以前に処理を完了するジョブの荷重和の最大化を目的としたスケジューリング問題である．処理開始可能時刻の制約を最も弱めた SRTDD は遅れジョブ荷重和最小化問題 (ST) に対応し，処理開始可能時刻の制約を最も強めた SRTDD は JIT ジョブ荷重和最大化問題 (SJIT) に対応する．それぞれの問題，それぞれの機械モデルについて，研究が進められている (第 3 章).

納期ズレ幅の導入 処理開始可能時刻を制約として設けることは，完了時刻に制約を設けることと等価である．本研究で導入した納期ズレ幅とは，各ジョブについて，納期よりどれくらい手前で処理を完了することを許すか，を表すパラメータであり，SRTDD もまた，納期ズレ幅に基づいて表記できる．SRTDD の各納期ズレ幅の値がその上界と等しい場合，ST に対応し，各納期ズレ幅の値がその下界と等しい場合，SJIT に対応する．

SRTDD は，強 **NP** 困難であることが知られているが (Garey & Johnson [3])，納期ズレ幅がとる値によっては， ST, SJIT に対応し，効率的に最適解を求めることができる．納期ズレ幅の導入により，SRTDD の問題の難しさの一部分は，納期ズレ幅に特徴付けられることが明らかとなった．

本研究では，導入した納期ズレ幅を特徴とする以下の研究成果を得た．

近似解法 強 **NP** 困難であり，かつ，これまで近似解法が開発されていない SRTDD に対し，並列機械モデルにおける近似解法を開発した．開発した近似解法の特徴として，近似率は，納期ズレ幅の値に基づく．納期ズレ幅の最大値が次を満たす SRTDD それぞれの問題に対して， $0 < \max_{i'}\{\alpha_{i'}\} \leq \min_{i'}\{p_{i'}\}, \cdots, (k-1)\min_{i'}\{p_{i'}\} < \max_{i'}\{\alpha_{i'}\} \leq k\min_{i'}\{p_{i'}\}$ ，計算時間を $\mathrm{O}(n^4 \log n)$ とする定数近似解法を開発した．k は任意の正整数であるため，任意の SRTDD の入力は，上記の何れかを満たす．

(P, α, D, W) を任意の SRTDD の入力とし，k を与えられた (P, α, D, W) について，$(k-1)\min_{i'}\{p_{i'}\} < \max_{i'}\{\alpha_{i'}\} \leq k\min_{i'}\{p_{i'}\}$ を満たす正整数とする．SRTDD における最適解を (A^α, C^α)，開発した解法が返す解を (A^0, C^0) とすると，開発した近似解法の近似率は以下である．

$$\frac{\varphi(A^0, C^0)}{\varphi(A^\alpha, C^\alpha)} \geq \frac{1}{k+1}.$$

また，タイトな問題例をあげ，開発した近似解法の近似率は，これ以上改良できないことを示した．

JIT スケジューリングの緩和 本研究では，JIT スケジューリング問題を納期ズレ幅に基づいて緩和した問題，疑似 JIT ジョブ荷重和最大化問題 (SPJIT) に対して，厳密解法を開発した．疑似 JIT ジョブ荷重和最大化問題 (SPJIT) とは，納期ズレ幅の上界を最小の処理時間 ($\min_{i'}\{p_{i'}\}$) とする SRTDD である．

SPJIT に対して開発した解法は動的計画法に基づく．納期ズレ幅に基づき，各部分問題の解の関係を分析し，計算時間を，$\mathrm{O}(mn^{m+1}\max_i\{\alpha_i\}^m)$ とする解法を開発した．機械数 m が定数の場合，問題は，疑似多項式時間解法を持つ．また，解法の計算時間は，納期ズレ幅 α_i に依存し，機械数 m が定数であり，かつ，各納期ズレ幅が n の多項式で表せる場合，開発した解法は，SPJIT に対する多項式時間解法となる．

解法の計算時間が，疑似多項式時間となる場合，多項式時間となる場合の境目は未だ明確でない．しかし，多項式時間となる場合の納期ズレ幅の条件が満たされない場合，つまり各納期ズレ幅が指数である場合，に対応する現実の状況は稀であると思われ，現実の問題の多くは，多項式時間で最適解が求まるものと考える．

特に，単一機械モデルについて，各納期ズレ幅が $\alpha_i < \min_{i'}\{p_{i'}\}$ を満たす SRTDD が疑似多項式時間解法を持つことは，成果の一部分として Chezh-Japan Seminar on Data Analysis and Decision Making under Uncertainty において，On Maximizing Weighted Number of Pseudo Just-In-Tme Jobs in Single Machine Environment として発表されている (Kawamata, Morita, & Sung [7])．

計算複雑さ SPJIT に対する厳密解法の開発により，納期ズレ幅の上界を最小処理時間とする問題は，機械数が定数の場合，高々疑似多項式時間で最適解が求まるため，強 **NP** 困難には成り得ない．納期ズレ幅の上界を最小処理時間とする問題に対する解

法の開発過程において，問題が弱 **NP** 困難であることが予想されたが，未解決である．しかし，関連して，納期ズレ幅の下界を最小処理時間とする問題について，計算複雑さに関して次の成果を得た．

k を任意の正整数とする．与えられた k について，任意の $i \in \{1,\ldots,n\}$ が，$k\min_{i'}\{p'_i\} \le \alpha_i$ を満たす SRTDD は，強 **NP** 困難であることを示した．つまり，納期ズレ幅の下界を $\min_{i'}\{p_{i'}\}$ とする SRTDD ， $\ldots$ ， $k\min_{i'}\{p_{i'}\}$ とする SRTDD ，それぞれは，依然，強 **NP** 困難である．

まとめ SRTDD の問題のほとんどの入力に対し，本研究で開発した何れかの解法が適用できる．また，納期ズレ幅を特徴とするこれらの成果は， SRTDD のうち計算複雑さが未解決であった問題に対する成果であり，SRTDD の難しさの本質を解明する足掛かりになるものと期待する．

表 7.1 は，機械数 m が定数の場合の，同一並列機械モデルにおける SRTDD の研究成果をまとめたものである．表中の多項式，疑似多項式は，それぞれ，問題が多項式時間解法を持つこと，問題が疑似多項式時間解法を持つことを表す．表中の×は，対応する問題がないことを表す．また，単一機械モデルは，機械数を 1 とする同一並列機械と等価であり，本研究の成果の一部として含まれる．

<table>
<tr><th rowspan="2">SRTDD
(近似解法)</th><th colspan="5">納期ズレ幅の上界</th></tr>
<tr><th>0</th><th colspan="2">$\min_{i'}\{p_{i'}\}$</th><th>$\cdots$</th><th>$d_i - p_i$</th></tr>
<tr><td rowspan="3">α_i が
異なってもよい</td><td rowspan="3">×</td><td colspan="2">SPJIT</td><td rowspan="4">OPEN</td><td rowspan="3">強 NP 困難 [3]
(下界：0)
強 NP 困難
(下界：$\min_{i'}\{p_{i'}\}$)</td></tr>
<tr><td>n の多項式</td><td>入力の一部</td></tr>
<tr><td rowspan="2">多項式</td><td rowspan="2">疑似多項式</td></tr>
<tr><td>α_i が等しい</td><td>SJIT
多項式 [5]</td><td>ST
弱 NP 困難 [4]</td></tr>
</table>

表 7.1: 同一並列機械モデルにおける SRTDD の研究成果 (m :定数)

7.2 今後の課題

近似解法 本研究で開発した近似解法の近似率は，納期ズレ幅の値に依存し，納期ズレ幅の最大値が最小の処理時間に対して大きい値をとるほど，近似率は悪くなる．また，納期ズレ幅の値のばらつきが大きい入力について，近似率が理論値と比べて大きく

なる可能性がある． SJIT に基づくより良い近似解法の開発は，今後の課題の一つである．

SPJIT に関して 機械数 m が入力の一部の場合，本研究で開発した解法は，問題の効率的な解法には成り得ず，問題の計算複雑さは未だ明らかでない．納期ズレ幅の上界が最小の処理時間であり，機械数 m が入力の一部である問題の，計算複雑さの証明は今後の課題の一つである．

機械数が定数の場合，疑似多項式時間解法の開発により，問題が強 **NP** 困難に成り得ないことが明らかとなった．しかし，問題の計算複雑さは未だ解決はされていない．機械数が定数の場合もまた，計算複雑さの証明が求められる．

本研究の提案解法によって，納期ズレ幅の上界が n の多項式で表せる場合，問題は多項式時間解法を持つ．しかし，この条件のみに着目すると，ほとんどの問題は多項式時間であり，多項式時間，疑似多項式時間の境界は未だ不明確である．計算時間の境界を明らかとする条件の導出が求められる．

SRTDD の計算複雑さ 納期ズレ幅の下界を $\min_{i'}\{p_{i'}\}$， $2\min_{i'}\{p_{i'}\}$， $\ldots$， $k\min_{i'}\{p'_i\}$ とする問題は，それぞれ，強 **NP** 困難であることを示した．しかし，3-PARTITION からの還元において，正整数 k の値が大きくなるに従い， X の要素に対応する各ジョブの処理時間，納期の値は大きくなる． k の値が大きくなるに従って小さくなっているように見える，納期ズレ幅の上界と下界の差は，それら X の要素に対応する各ジョブについては顕著に広がり， SRTDD の問題の本質的な難しさは未だ現れていない．

また，下界を $d_i - p_i$ に近づけていくことで， ST との間の計算複雑さの境界が，上界を $\min_i\{p_i\}$ に近づけていくことで， SPJIT との間の計算複雑さの境界が明らかとなるものと考える．問題の難しさの境界に対する納期ズレ幅に基づく特徴付けは，今後の課題である．

参考文献

[1] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B.. Network Flows – Theory, Algorithms, and Applications., Englewood Cliffs, NJ:Prentice–hall, 1993.

[2] Baptiste, P.. Polynomial Time Algorithm for Minimizing the Weighted Number of Late Jobs on a Single Machine with Equal Processing Times, Journal of Scheduling, 2, pp.245–252, 1999.

[3] Garey, M. R., and Johnson, D. S.. Two-Processor Scheduling with Start-Times and Deadlines, SIAM Journal on Computing, 6, pp.416–426, 1977.

[4] Garey, M. R., and Johnson, D. S.. Computers and Intractability: A Guide to the Theory of NP-Completness, W. H. Freeman and Company, New York, 1979.

[5] Hiraishi, K., Levner, E., and Vlach, M.. Scheduling of parallel identical machines to maximize the weighted number of just-in-time jobs, Computers & Operations Research, 29(7), pp.841–848, 2002.

[6] Karp, R.M.. Reducibility among Combinatorial Problems, in R. E. Miller, and J. W. Thatcher (eds.), Complexity of Computer Computations, Plenum Press, New York, pp.85–103, 1972.

[7] Kawamata, Y., Morita, M., and Sung, S. C.. On Maximizing Weighted Number of Pseudo Just-In-Time Jobs in Single Machine Environment, Czech-Japan Seminar on Data Analysis and Decision Making in Service Science 15, pp.88–91, 2012.

[8] Kise, H., Ibaraki, T., and Mine, H.. A soluvable case of the one-machine scheduling problem with ready and due dates, Oper. Res. 26, pp.121–126, 1978.

[9] Lann, A., and Mosheiov, G.. Single Machine Scheduling to Minimize the Number of Early and Tardy Jobs, Computers & Operations Research 23, pp.765–781, 1996.

[10] Lawler, E. L., and Moore, J. M.. A Functional Equation and its Application to Resource Allocation and Sequencing Prolems, Management Science, 16, pp.77–84, 1969.

[11] Lawler, E. L.. Sequencing to Minimize the Weighted NUmber of Late Jobs, RAIRO Recherche Operationnel, 10, pp.27–33, 1976.

[12] Lawler, E. L.. A Dynamic Programming Algorithm for Preemptive Scheduling of a Single Machine to Minimize the Number of Late Jobs, Annals of Operations Research, 26, pp.125–133, 1990.

[13] Marc, S., and Philippe, T.. Heuristics and Metaheuristics for a Parallel Machine Scheduling Problem: a Computational Evaluation, Research Report 01-2-SP, 2001.

[14] Moore, J. M.. An n Job, One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs, Manag. Sci. 15, pp.102–109, 1968.

[15] Sung, S. C., and Vlach, M.. Maximizing Weighted Number of Just-In-Time Jobs on Unrelated Parallel Machines, Jurnal of Scheduling, 8, pp.453–460, 2005.

[16] v. d. Akker, J. M., and Hogeveen, J. A.. Minimizing the Number of Tardy Jobs, in J.Y-T. Leung (eds.), Handbook of Scheduling: Algorithms, Models, and Performance Analysis, A CRC Press Company, New York, Chapter 12, 2004.

付 録A　計算複雑さ

JIT ジョブ荷重和最大化問題 (SJIT) は，処理開始可能時刻付き遅れジョブ荷重和最小化問題 (SRTDD) のうち，各納期ズレ幅を 0 とする問題である．第 5 章で扱った，納期ズレ幅の上界を最小の処理時間とする SRTDD もまた， SJIT を内包する．ここでは，SJIT に対する従来研究の成果のうち，問題の計算複雑さに関する成果を紹介する．

JIT ジョブ荷重和最大化問題

無関連並列機械モデルにおいて，処理時間はジョブと機械の組み合わせによって異なる値をとる．この章では，ジョブ $i \in \{1, \ldots, n\}$ の機械 $j \in \{1, \ldots, m\}$ における処理時間を $p_{i,j}$ と表し，まとめて $n \times m$ 行列 $P = [p_{ij}]$ で表す．また，以降 JIT ジョブ荷重和最大化問題において納期に処理を完了する各ジョブを JIT ジョブと呼ぶ.

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，この問題は 3 -SAT 問題からの還元により，強 NP 困難であることが証明されている.

3 -SAT 問題

Instance: ブール型変数の集合 X と， X 上の 3 つのリテラルからなる集合の集合 H (各 $h \in H$ は $|h| = 3$ を満たす).

Question: H を充足する真理値割り当て $f : X \to \{0, 1\}$ ，つまり,

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

を満たす f が存在するか？

ブール型変数の集合 $X = \{x_1, \ldots, x_\beta\}$ と， X 上の 3 つのリテラルからなる集合 $H = \{h_1, \ldots, h_\lambda\}$ の 2 項組， (X, H) を任意の 3-SAT 問題の入力とする.

議論に先立ち，いくつかの表記を導入する．各 $k \in \{1, \ldots, \beta\}$ について， γ_k を C において x_k が現れる回数を表す自然数とし， γ を γ_k ($k \in \{1, \ldots, \beta\}$) の最大値に 1 を加えた値とする ($\gamma = \max_{k \in \{1, \ldots, \beta\}} \{\gamma_k\} + 1$).

(X,H) に基づき，以下のように JIT ジョブ荷重和最大化問題の入力を設定する． $2\beta\gamma$ 個のジョブを $\lambda+\beta$ 台の無関連機械で処理する．各ジョブの処理時間，納期ズレ幅，納期，荷重を以下のように設定する．

納期ズレ幅の設定

各ジョブ $i \in \{1,\ldots,2\beta\gamma\}$ の納期ズレ幅 α_i を 0 とする．

納期の設定

$\{1,\ldots,\beta\gamma\}$ の $\beta\gamma$ 個のジョブの納期を設定する．添え字の昇順で γ 個ずつとった β 個のグループについて，機械を順に対応させる ($k \in \{1,\ldots,\beta\}$ 番目のグループに対応する機械は k である)．各グループ $k \in \{1,\ldots,\beta\}$ の γ 番目のジョブの納期を $2\gamma-1$ ， ℓ 番目のジョブ (ただし $\ell \neq \gamma$) の納期を ℓ とする．つまり，各 $k \in \{1,\ldots,\beta\}$ と $\ell \in \{1,\ldots,\gamma\}$ について，

$$d_{(k-1)\gamma+\ell} = \begin{cases} \ell & \text{if } \ell < \gamma, \\ 2\gamma-1 & \text{if } \ell = \gamma. \end{cases}$$

$\{\beta\gamma+1,\ldots,2\beta\gamma\}$ の $\beta\gamma$ 個のジョブについても同様に，添え字の昇順に基づき β 個のグループに分ける．各グループ $k \in \{1,\ldots,\beta\}$ の γ 番目のジョブの納期を γ ， ℓ 番目のジョブ (ただし $\ell \neq \gamma$) の納期を $\gamma+\ell$ とする．つまり，各 $k \in \{1,\ldots,\beta\}$ と $\ell \in \{1,\ldots,\gamma\}$ について，

$$d_{(\beta+k-1)\gamma+\ell} = \begin{cases} \gamma+\ell & \text{if } \ell < \gamma, \\ \gamma & \text{if } \ell = \gamma. \end{cases}$$

荷重和の設定

前述と同様に，各ジョブを納期昇順に基づき 2β 個のグループに分ける．グループ $k \in \{1,\ldots,\beta\}$ と，グループ $k+\beta$ について，グループの γ 番目のジョブの荷重を γ とし，それ以外のジョブの荷重を 1 とする．つまり，各 $k \in \{1,\ldots,\beta\}$ と $\ell \in \{1,\ldots,\gamma\}$ について，

$$w_{(k-1)\gamma+\ell}v = w_{(\beta+k-1)\gamma+\ell} = \begin{cases} 1 & \text{if } \ell < \gamma, \\ \gamma & \text{if } \ell = \gamma. \end{cases}$$

処理時間の設定

前述と同様に，各ジョブを納期昇順に基づき 2β 個のグループに分ける．グループ $k \in \{1,\ldots,\beta\}$ に機械を k を対応させ，グループ $k+\beta$ にも同様に機械 k を対応させる．各グ

ループの各ジョブについて， グループに対応した機械以外の各機械における処理時間を 2γ とする．また，各グループの γ 番目のジョブの，グループに対応する機械における処理時間を γ とし， γ 番目のジョブをのぞいた各ジョブの，グループに対応する機械における処理時間を 1 とする．つまり，各 $j \in \{1,\dots,\beta\}$ ， $k \in \{1,\dots,\beta\}$ ， $\ell \in \{1,\dots,\gamma\}$ について，

$$p_{(k-1)\gamma+\ell,j} = p_{(\beta+k-1)\gamma+\ell,j} = \begin{cases} 1 & \text{if } \ell < \gamma \text{ and } j = k, \\ \gamma & \text{if } \ell = \gamma \text{ and } j = k, \\ 2\gamma & \text{otherwise.} \end{cases}$$

次に，残りの各機械 $j \in \{\beta+1,\dots,\beta+\lambda\}$ について，各ジョブの処理時間を設定する．各グループ $k \in \{1,\dots,\beta\}$ について， x_k を対応させ，グループ $k+\beta$ について $\bar{x}_k$ を対応させる．各グループ $k \in \{1,\dots,\beta\}$ について， $\ell \in \{1,\dots,\gamma-1\}$ 番目の各ジョブについて， $x_k \in h_{j-\beta}$ であれば $d_{(k-1)\gamma+\ell}$ とし，そうでない場合は 2γ とする．また， γ 番目のジョブの処理時間は 2γ とする．各グループ $k+\beta$ ($k \in \{1,\dots,\beta\}$) についても同様に， $\ell \in \{1,\dots,\gamma-1\}$ 番目の各ジョブについて， $\bar{x}_k \in h_{j-\beta}$ であれば $d_{(\beta+k-1)\gamma+\ell}$ とし，そうでない場合は 2γ とする．また， γ 番目のジョブの処理時間は 2γ とする．つまり，各 $j \in \{\beta+1,\dots,\beta+\lambda\}$ と $k \in \{1,\dots,\beta\}$ ， $\ell \in \{1,\dots,\gamma\}$ について，

$$p_{(k-1)\gamma+\ell,j} = \begin{cases} d_{(k-1)\gamma+\ell} & \text{if } \ell < \gamma \text{ and } x_k \in h_{j-\beta}, \\ 2\gamma & \text{otherwise,} \end{cases}$$

$$p_{(\beta+k-1)\gamma+\ell,j} = \begin{cases} d_{(\beta+k-1)\gamma+\ell} & \text{if } \ell < \gamma \text{ and } \bar{x}_k \in h_{j-\beta}, \\ 2\gamma & \text{otherwise.} \end{cases}$$

このように設定した (P,α,D,W) は JIT ジョブ荷重和最大化問題の入力である．以降， (X,H) に基づいて前述のように設定した (P,α,D,W) を区別して $I_{(X,H)}$ と表す ($I_{(X,H)} = (P,\alpha,D,W)$).

以下， H を充足する真理値割り当て $f : X \to \{0,1\}$ が存在することと， $I_{(X,H)}$ を入力とする JIT ジョブ荷重和最大化問題に対して $\varphi(A,C) \geq \beta(2\gamma-1)+\lambda$ を満たす実行可能なスケジュール (A,C) が存在することが，同値であることを示す.

補題 14 H を充足する真理値割り当て $f : X \to \{0,1\}$ が存在するならば， $I_{(X,H)}$ を入力とする *JIT* ジョブ荷重和最大化問題に対して $\varphi(A,C) \geq \beta(2\gamma-1)+\lambda$ を満たす実行可能なスケジュール (A,C) が存在する．

証明 (X, H) を任意の 3 -SAT 問題の入力とし， $f : X \to \{0, 1\}$ を H を充足する心理地割り当てとする．つまり， f は以下を満たす．

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

各 $k \in \{1, \ldots, \beta\}$ について， $f(x_k) = 1$ ならばグループ $\beta + k$ の全てのジョブを，グループ $\beta + k$ に対応する機械 k に割り当て，そうでなければグループ k の全てのジョブを，グループ k に対応する機械 k に割り当てる．また，機械 k に割り当てた各ジョブの完了時刻を納期とする．つまり， $f(x_k) = 1$ ならば，各 $\ell \in \{1, \ldots, \gamma\}$ について $A_k := A_k \cup \{(\beta + k - 1)\gamma + \ell\}$ かつ $C((\beta + k - 1)\gamma + \ell) = d_{(\beta+k-1)\gamma+\ell}$ ， $f(x_k) = 0$ ならば，各 $\ell \in \{1, \ldots, \gamma\}$ について $A_k := A_k \cup \{(k - 1)\gamma + \ell\}$ かつ $C((k - 1)\gamma + \ell) = d_{(k-1)\gamma+\ell}$ ． $I_{(X,H)}$ の定義より，ここまでのスケジュールにおいて，機械 $1, \ldots, \beta$ に割り当てたジョブの処理時間は重複せず (図 A.1)，完了時刻は納期であることから，$k \in \{1, \ldots, \beta\}$ に割り当てたジョブは JIT ジョブである．よって，ここまでのスケジュール (A, C) における JIT ジョブの荷重和は，

$$\varphi(A, C) = \sum_{k \in \{1,\ldots,\beta\}} \left(\sum_{i \in \mathcal{Q}(A,C) : i \in A_k} w_i \right) = \sum_{k \in \{1,\ldots,\beta\}} (2\gamma - 1) = \beta(2\gamma - 1).$$

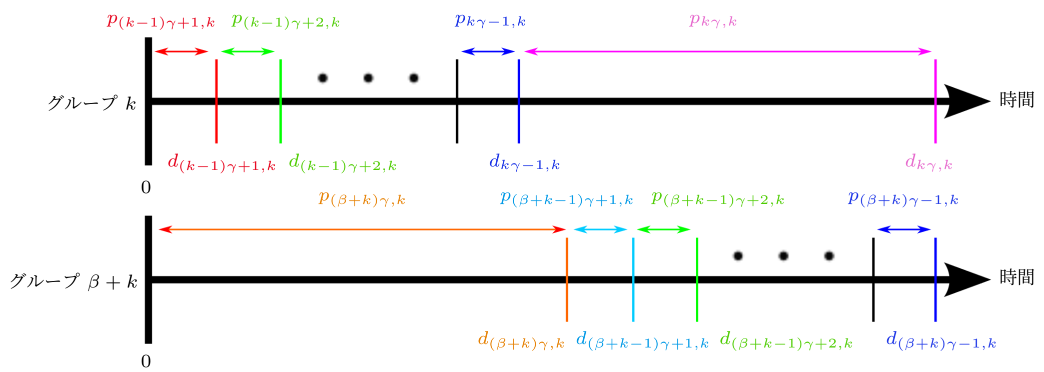

図 A.1: 機械 $k \in \{1, \ldots, \beta\}$ に対応するグループ k とグループ $\beta + k$ の各ジョブ

各 $j \in \{1, \ldots, \lambda\}$ について， $f(x_k) = 1$ を満たす $x_k \in h_j$ か， $f(x_k) = 0$ を満たす $\bar{x}_k \in h_j$ をひとつ見つける． f が H を充足することから，各 $j \in \{1, \ldots, \lambda\}$ について，上記を満たすリテラルは必ず存在する．

与えられた $j \in \{1, \ldots, \lambda\}$ について， $f(x_k) = 1$ を満たす $x_k \in h_j$ を見つけたと仮定する．グループ k のジョブのうち，いずれの機械にも割り当てられていない $\ell \in \{1, \ldots, \gamma - 1\}$ を見つける．グループ k のジョブは， $f(x_k) = 1$ であることから，$\{1, \ldots, \beta\}$ のいずれの機械にも割り当てられておらず，また，各リテラルが現れる回数は高々 $\gamma - 1$ であること

から，そのような ℓ 番目のジョブは必ず存在する． $A_{j+\beta} := A_{j+\beta} \cup \{(k-1)\gamma + \ell\}$ とし，$C((k-1)\gamma + \ell) = d_{(k-1)\gamma+\ell}$ とする．

次に，与えられた $j \in \{1, \ldots, \lambda\}$ について， $f(x_k) = 0$ を満たす $\bar{x}_k \in h_j$ を見つけたと仮定する．グループ $\beta + k$ のジョブのうち，いずれの機械にも割り当てられていない $\ell \in \{1, \ldots, \gamma\}$ を見つける．先ほどと同様の議論で，そのような ℓ は必ず存在する．$A_{j+\beta} := A_{j+\beta} \cup \{(\beta + k - 1)\gamma + \ell\} = j + \beta$ とし， $C((\beta + k - 1)\gamma + \ell) = d_{(\beta+k-1)\gamma+\ell}$ とする．

各グループの $\ell \in \{1, \ldots, \gamma - 1\}$ 番目のジョブの荷重が 1 であることから，各 $j \in \{1, \ldots, \lambda\}$ について j に割り当てられた JIT ジョブの荷重和は,

$$\sum_{i \in \mathcal{Q}(A,C):A(i)=j+\beta} w_i = 1.$$

ここまでのスケジュール (A, C) においていずれの機械にも割り当てられていないジョブを， $\max\{d_i \mid i \in \{1, \ldots, 2\beta\gamma\}$ のあと，任意の順，任意の機械で処理するものとする．

明らかにスケジュール (A, C) は実行可能であり，以下を満たす.

$$\varphi(A, C) = \sum_{j \in \{1,\ldots,\beta\}} \sum_{i \in \mathcal{Q}(A,C):i \in A_j} w_i + \sum_{j \in \{\beta+1,\ldots,\lambda+\beta\}} \sum_{i \in \mathcal{Q}(A,C):i \in A_j} w_i = \beta(2\gamma - 1) + \lambda.$$

■

ここで， $I_{(X,H)}$ を入力とする JIT ジョブ荷重和最大化問題における実行可能なスケジュールについて，いくつかの性質を示す.

(A, C) を $I_{(X,H)}$ を入力とする JIT ジョブ荷重和最大化問題における任意の実行可能なスケジュールとする．実行可能性の制約により，各ジョブは機械の稼働開始時刻以降に処理を開始することから，各 JIT ジョブ $i \in \mathcal{Q}(A, C)$ について， $C(i) - p_i = d_i - p_i \geq 0$ である．このことから， $p_{i,j} \geq 2\gamma$ かつ $d_i \leq 2\gamma - 1$ を満たす $i \in \{1, \ldots, n\}$ と $j \in \{1, \ldots, m\}$ について， $i \in \mathcal{Q}(A, C)$ のとき， $i \notin A_j$ である．したがって， $I_{(X,H)}$ の定義より，各 $k \in \{1, \ldots, \beta\}$ について,

- $i \in \mathcal{Q}(A, C)$ かつ $i \in A_k$ ならば，$(k-1)\gamma + 1 \leq i \leq k\gamma$ または $(\beta + k - 1)\gamma + 1 \leq i \leq (\beta + k)\gamma$ である．

- $k\gamma \in \mathcal{Q}(A, C)$ ならば $k\gamma \in A_k$ であり，$(\beta + k)\gamma \in \mathcal{Q}(A, C)$ ならば $(\beta + k)\gamma \in A_k$ である．

実行可能性の制約より，同じ機械に割り当てられた複数のジョブの処理は重複しないことから，各 $k \in \{1,\dots,\beta\}$ について，

- $k\gamma \in \mathcal{Q}(A,C)$ ならば，$i \in A_k$ である任意の $i \in \mathcal{Q}(A,C)$ について，i は $(k-1)\gamma - 1 \leq k\gamma$ を満たす (図 A.1).
- $(\beta+k)\gamma \in \mathcal{Q}(A,C)$ ならば， $i \in A_k$ である任意の $i \in \mathcal{Q}(A,C)$ について， i は $(\beta+k-1)\gamma - 1 \leq i \leq (\beta+k)\gamma$ を満たす.

以上の議論より，各 $k \in \{1,\dots,\beta\}$ について， k に割り当てられた JIT ジョブの荷重和は以下を満たし，

$$\sum_{i \in \mathcal{Q}(A,C): i \in A_k} w_i \leq 2\gamma - 1, \tag{A.1}$$

以下のいずれかを満たす場合にのみ， $\sum_{i \in \mathcal{Q}(A,C): i \in A_k} w_i = 2\gamma - 1$ となる.

$$\{i \in \mathcal{Q}(A,C) \mid i \in A_k\} = \{(k-1)\gamma+1, (k-1)\gamma+2, \dots, k\gamma\}, \tag{A.2}$$

$$\{i \in \mathcal{Q}(A,C) \mid i \in A_k\} = \{(\beta+k-1)\gamma+1, (\beta+k-1)\gamma+2, \dots, (\beta+k)\gamma\}. \tag{A.3}$$

さらに，各 $j \in \{\beta+1,\dots,\beta+\lambda\}$ について，各ジョブ $i \in \{1,\dots,2\beta\gamma\}$ の処理時間は $p_{ij} \in \{d_i, 2\gamma\}$ であり，機械 j に割り当てられるジョブは高々 1 つである ($|\{i \in \mathcal{Q}(A,C) \mid i \in A_j\}| \leq 1$). 既に述べたように，各 $k \in \{1,\dots,\beta\}$ について， $k\gamma \in \mathcal{Q}(A,C)$ ならば $k\gamma \in A_k$ であり， $(\beta+k)\gamma \in \mathcal{Q}(A,C)$ ならば， $(\beta+k)\gamma \in A_k$ である．これらのジョブをのぞいた各ジョブの荷重は 1 であることから，各 $j \in \{\beta+1,\dots,\lambda+\beta\}$ について，

$$\sum_{i \in \mathcal{Q}(A,C): i \in A_j} w_i \leq 1. \tag{A.4}$$

補題 15 $I_{(X,H)}$ を入力とする *JIT* ジョブ荷重和最大化問題に対して $\varphi(A,C) \geq \beta(2\gamma - 1) + \lambda$ を満たす実行可能なスケジュール (A,C) が存在するならば， H を充足する真理値割り当て $f : X \to \{0,1\}$ が存在する.

証明 (A,C) を $I_{(X,H)}$ を入力とする JIT ジョブ荷重和最大化問題に対して $\varphi(A,C) \geq \beta(2\gamma-1)+\lambda$ を満たす実行可能なスケジュールとする．式 (15) と式 (A.4) より $\varphi(A,C) \leq \beta(2\gamma-1)+\lambda$ であり，このことから，$\varphi(A,C) = \beta(2\gamma-1)+\lambda$ である．各機械 $k \in \{1,\dots,\beta\}$ について，式 (A.2) または式 (A.3) が満たされ，各 $j \in \{\beta+1,\dots,\beta+\lambda\}$ について，$|\{i \in \mathcal{Q}(A,C) \mid i \in A_j\}| = 1$ である． (A,C) に基づき， $f : H \to \{0,1\}$ を以下のように設定する．各 $k \in \{1,\dots,\beta\}$ について，式 (A.3) が満たされるならば $f(x_k) = 1$ とし，そ

うでなければ $f(x_k)=0$ とする．明らかに，各 $j \in \{\beta+1,\ldots,\beta+\lambda\}$ について j に割り当てられたただ一つの JIT ジョブ $i \in \mathcal{Q}(A,C)$ が存在し， $i \le \beta\gamma$ ならば $f(x_k)=1$ を満たすある $x_k \in h_{j-\beta}$ が存在し， $i > \beta\gamma$ ならば $f(x_k)=0$ を満たすある $\bar{x}_k \in h_{j-\beta}$ が存在する．したがって， f は H を充足する真理値割り当てである． ■

補題 15 と補題 A.4 より， H を充足する真理値割り当て $f : X \to \{0,1\}$ が存在することと， $I_{(X,H)}$ を入力とする JIT ジョブ荷重和最大化問題に対して $\varphi(A,C) \ge \beta(2\gamma-1)+\lambda$ を満たす実行可能なスケジュール (A,C) が存在することは，同値である．

3 -SAT 問題の入力 (X,H) に基づく $I_{(X,H)}$ の生成に要する計算時間は，明らかに多項式時間であり，入力 $I_{(X,H)}$ の長さは， (X,H) の長さに関する多項式で表すことができる．したがって，以下の定理が成り立つ．

定理 16 *[15]* 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， *JIT* ジョブ荷重和最大化問題は強 *NP* 困難である．

謝辞

本論文をまとめるにあたり，暖かい激励，ご鞭撻を頂いた宋少秋教授に，心より，感謝いたします．終始熱心なご指導を頂き，ありがとうございました．また，論文をご精読いただきました大内紀知准教授，日吉久礎准教授に，深謝いたします．

本研究を進めるにあたり，弛まず応援してくださった両親，共に学んだ学友に，ここに感謝いたします．ありがとうございました．

2013 年 1 月 31 日 川俣 友佳

修士論文

納期ズレ幅の導入による

川俣友佳

JITスケジューリング問題の緩和とその解法