多目的スケジューリングに関する研究

1999年

毛利 進太郎

目 次

第1章

緒論

1.1 背景と目的

近年，生産現場における製造工程の多様化，高速化によってスケジューリング理論に対する要求は日増しに高まっている．何を生産するかということのみではなくどのように生産を行うのかということが，コストの面さらには環境問題の面からも問われている．一方，スケジューリング理論の応用範囲は製造業のみならず，さまざまな分野，特に計算機のオペレーティングシステムにおける各タスクのスケジューリングや CPU における処理の高速化，並列計算のスケジューリング，さらには医療現場におけるナース・スケジューリングのような人的資源の適正な配分など広範囲に渡っている．

オペレーションズ・リサーチの分野において扱われるスケジューリング問題とは以下のように定義される．

人，設備，機械等があり，それらを用いて処理すべき対象があるときに，前者を機械，後者を仕事またはオペレーションと呼ぶ．各機械を用いて各仕事またはオペレーションを処理すべき時間の割当てがスケジュールである．このときあらかじめ問題ごとに定められた条件を満たすスケジュールを実行可能スケジュールであるという．実行可能であるスケジュールのうち，定められた目的に対して最適であるスケジュールを最適スケジュールといい，この最適スケジュールを求める問題をスケジューリング問題という．

スケジューリング理論で扱うべき場面はますます多岐に渡っており，そこで生じる制約やスケジュールを評価する基準も個々に異なってくる．スケジューリング問題に関する研究は 1950 年代の 2 機械フローショップ問題に対する Johnson の研究 [48] や 1 機械最大納

期ずれ問題に対するJacksonの研究[44]で始まった．しかしながらスケジューリング問題に関する研究の多くは，単純化された機械環境のもとで単一の目的関数を持つ問題を扱っており，現実の生産現場における複雑な要求をすべて満たすようなスケジュールを得るのは難しい．また1970年代におけるCook[25]やKarp[52]による組合せ最適化問題における計算の複雑性に関する研究によってスケジューリング問題の多くは非常に計算が難しい問題のクラスに属することがわかってきた．そこで近年では計算機の処理速度の向上もあり，スケジューリングを必要とする現場では現実的な時間でより良い解を得るように発見的手法によるスケジューリングが行われてきている．一般に生産現場においては単一の目的を単純に最適化するよりは，さまざまな評価基準をバランスよく満たすようなスケジュールが望まれる．このように複数の目的関数を同時に考慮する問題を多目的スケジューリング問題と呼ぶ．

スケジューリングに限らず最適化問題においてこれまでは単一の目的関数を扱うことが多かった．しかし現実において最適化を考えるべき場面の多くは本質的に多くの評価基準を含んでおり，一元的な価値基準で評価を行うことができる場面のほうがむしろまれである．そのように複数の目的を考慮する最適化問題を多目的最適化問題 (Multi Criteria Optimaization Problem) と呼ぶ．しかし複数の評価基準を持つ問題を数学的に取り扱うための研究が行われ始めたのは最近のことである．多目的最適化問題においては，全ての評価基準を最適化するような解が存在することはほとんどありえず，まず何をもって最適な解であるとするかを定義する必要がある．この定義で最も一般的なものが非劣解である．非劣解は多目的最適化問題において2つの実行可能解を比較したときの優越関係を定義し，ある解より優れた解が存在しないときにその解を非劣解とするものである．現場において実際にスケジュールを決定する際には経験的，もしくは発見的な手法を用いてスケジュールを構築することがよく行われるが，このときに各目的関数がどの程度の値であれば実行可能なスケジュールが存在するかを正確に推定することが重要である．よって非劣解がその問題の目的関数に関する実行可能領域においてどのように位置するのかが意思決定を行う際の非常に有益な情報となる．

本研究では上記の観点から，従来単一の目的関数のみを扱ってきたスケジューリング問題に対し複数の目的関数を持つスケジューリングのモデルを扱う．そしてこれらに対し数理的立場から非劣スケジュールを構築する方法を考案する．さらに複数の目的関数間のトレードオフの関係について考察を行う．これらの結果は現実の生産現場などでスケジュー

リングを考察する場合において数理的基礎となり，実際にスケジュールを構築するときに，それらの評価基準間の関係を示しその情報をもとにスケジュールを構築するため有益である．また複数の目的関数を持つ組合せ最適化問題に対する研究の端緒としての意義があると思われる．

1.2 概要

本論文の構成は以下のようになっている．

第 2 章では，準備として本論文の基礎となるスケジューリングのモデルを概観し，多目的最適化における幾つかの手法について述べ，非劣解の定義を行う．

第 3 章では，従来のスケジューリングのモデルの一般化である多機能機械において，最大完了時間と最大納期遅れという 2 つの目的関数を持つオープンショップスケジューリング問題を考え，非劣スケジュールを構成する方法を考案する．また Flexible Manufacturing Systems へ適用するために拡張を行い，さらに再生不可能な資源の消費量，最大完了時間と納期遅れという 3 つの目的関数を持つオープンショップスケジューリング問題への拡張を行う．

第 4 章では，等価並列 2 機械，3 機械の機械環境においてそれぞれの仕事が異なる納期を持つ問題を考える．Sahni によって提案されたアルゴリズムから，これらの問題において実行可能なスケジュールが存在するための条件を新たに導き，この条件より最大完了時間と最大納期遅れの二つの目的関数をもつ問題に対して，目的関数間の関係式を導き，実行可能領域の境界が非劣解になっていることを示し，非劣スケジュールを求める方法を開発する．

第 5 章では，等価並列機械の機械環境において，それぞれの仕事が異なる納期を持ち，かつ資源を割り当てることによって処理時間が変化する問題を考える．4 章において導いた実行可能なスケジュールが存在するための条件から，資源の割り当てに関するいくつかの定理を導き，資源の消費に伴うコストを最小とするような実行可能スケジュールを求めるアルゴリズムを開発する．また資源の消費に伴うコストと最大納期遅れの 2 目的の問題において上記のアルゴリズムで求めた解が非劣解となっていることを示す．

第 6 章では等価並列機械において，各機械の使用に対しその使用時間に依存したコストがかかるスケジューリング問題を考える．機械の利用時間に対し制約がある問題を考え，その時のコストの総和を最小にするアルゴリズムを与える．一方コストの総和に制約があ

る問題も考え，その時の最大完了時間を最小にするアルゴリズムを与える．これらの結果より機械の利用時間と機械の利用に対するコストの総和との関係を求め，機械の利用時間と機械の利用コストの総和という2目的の問題において，上記のアルゴリズムで求めた解が非劣解となっていることを示す．

第7章において本研究の総括を行い．その成果や意義をまとめるとともに，今後の課題について述べる．

第2章

多目的スケジューリング問題

製造業は絶えず市場の変化に対応し，社会的ニーズを満たすようにその生産活動の計画を立てる．これらの生産活動の計画は上層から長期計画，中期計画，短期計画，瞬時計画といった階層に分けることができ，上層から下層に進むにしたがって計画の期間や対象を限定し，より確度の高い情報に基づいて具体的な計画が立てられる．これらの階層の中でスケジューリング理論が扱うべき問題は一般に短期計画に対応し，上位の計画によって決定された生産物を生産ライン，機械，人員などの資源に対し割り当て，具体的に実行できる計画を立案するために用いられる．

適切なスケジュールを立てるべき状況はその業種，生産物などによって異なり，また長期，中期計画よって定められる制約や要求も多種にわたる．これらを包括的に扱うことは困難であり，従来，現場に精通した決定者が経験と勘によってスケジュールを立案することが多く行われてきた．しかしこれらの知識を他者に伝達することは困難であり，また近年のコンピュータやネットワーク技術の発達により，企業活動全般に情報化とシステム化が推し進められた結果，長，中期計画よりの要求に対しそれらを満たすような現場のスケジューリングをそのシステム内において自動的に構成することが求められるようになり，それらを実現するためのスケジューリング理論が重要視されている．

スケジューリング問題に関する数理的研究は 1950 年代の 2 機械フローショップ問題に対する Johnson の研究 [48] や 1 機械最大納期ずれ問題に対する Jackson の研究 [44] で始まり，それ以降さまざまなモデルについて研究が進められてきた．そこで本章では一般的なスケジューリング問題の概略を述べ，さらに多目的最適化問題における意思決定について述べる．

2.1　スケジューリング問題

人，設備，機械等があり，それらを用いて処理すべき対象があるときに，前者を機械，後者を仕事またはオペレーションと呼ぶ．機械は $M_j(j=1,\cdots,m)$，仕事は $J_i(i=1,\cdots,n)$ で表わすとする．仕事 J_iは n_i個のオペレーション $O_{ik}(k=1,\cdots,n_i)$ からなるとし，オペレーション O_{ij}はその処理に p_{ij}時間を要するものとする．もし仕事 J_iが単一のオペレーションからなる (すなわち $n_i=1$) ならば，J_iは O_{i1}を示すものとし，さらにその処理時間を p_iとする．さらに仕事 J_iに関する要素として最早開始時間 r_i，納期 d_i，処理にかかるコストなどが定められている場合がある．それぞれのオペレーション O_{ij}に対し機械集合 $U_{ij}\subseteq\{M_1,\ldots,M_m\}$ が定められおり，O_{ij}は U_{ij}内のいずれかの機械によって処理されるものとする．これまでは全ての U_{ij}がただ一つの要素を持つか，もしくは全ての U_{ij}が全ての機械の集合に等しい場合のみが考えられてきた．前者は単一機械，ショップ型機械などであり，後者は並列機械である．さらに最近はさまざまな工具 (Tool) を使い分けて処理を行う FMS(Flexible Manufacturing Sytems) における問題などを考えるための機械環境の一般化が考えられている．この場合，仕事はいずれの機械においても処理を行うことができるが，定められた工具を必要とする．これをモデル化したものを多機能機械 (Multi-Purpose Machines:MPM) と呼ぶ [17, 18, 50]．また U_{ij}の要素である全ての機械が O_{ij}を処理する際に同時に用いられるようなときも考えられており，そのような問題は多機械処理 (Multiprocessor Task) スケジューリング問題と呼ばれる [5, 7, 12, 26, 36, 57].

ここで各機械上での各仕事またはオペレーションの実行すべき時間区間への割り当てがスケジュールである．このときあらかじめ問題ごとに定められた制約条件を満たすスケジュールが実行可能スケジュールである．実行可能であるようなスケジュールのうち，定められた目的関数を最適にするようなスケジュールを最適スケジュールといい，この最適スケジュールを求める問題をスケジューリング問題という.

スケジューリング問題における評価基準は各仕事 J_iの完了時間 C_iに対する何らかのコスト関数 $f_i(C_i)$ で評価される場合が多い．このような関数 f_iの要素として処理時間 p_iだけではなく，最早開始時間 r_i，納期 d_i，重み w_iなどがよく用いられる．また一般にスケジューリング問題に関する全てのデータは整数であると仮定されることが多い.

Graham らはこれらのモデルを表わすために $\alpha\mid\beta\mid\gamma$ という3つの項による表記を提案している [32]．この表記法では α は機械環境，β は仕事の属性，γ は目的関数という3つの項によってモデルを表記する.

2.1.1 機械環境

機械環境αは$\alpha = \alpha_1\alpha_2$という二つの項で表わされる．$\alpha_1$は機械環境の形態を示し，$\alpha_2$は機械の台数を示している．機械環境は大きく分けて単一の工程がある場合と複数の工程がある場合に分けることができる．単一工程の場合は仕事はただ一つのオペレーションからなり，複数工程の場合は各仕事が複数のオペレーションからなっており，ショップ型と呼ばれる．

単一工程において機械が一台しかない場合は単一機械といい，$\alpha_1 = \phi$と記述する．このときすべての仕事は同一の機械で処理され，各仕事に対する時間の割り当て方のみが問題となる．

単一工程で機械が複数台ある場合は並列機械であるといい，全ての機械は同じ処理を行うことができ，各仕事はいずれの機械でも処理することが可能である．

いずれの仕事も全ての機械で等しい処理時間で処理されるときに，等価並列機械であるといい，$\alpha_1 = P$と記述する．すなわち仕事 J_iの標準処理時間を p_iとし，仕事 J_iを機械 M_jによって処理したときの処理時間を p_{ij}とすると $p_{ij} = p_i$である．

それぞれの機械 M_jに対し処理速度比 s_jが決められており, 仕事の処理時間は機械の処理速度に依存する場合，すなわち機械 M_j上での仕事 J_iの処理時間 p_{ij}が $p_{ij} = p_i/s_j$であるとき一様並列機械といい, $\alpha_1 = Q$ と記述する．

仕事をどの機械で処理するかによって処理時間が異なるとき，非一様並列機械という．すなわち機械 M_jにおいて仕事に依存した速度比 s_{ij}が決められており，機械 M_j上での仕事 J_iの処理時間 p_{ij}は $p_{ij} = p_i/s_{ij}$となる．このとき$\alpha_1 = R$と記述する．

複数工程の場合，各仕事は複数の機械によって処理されるオペレーションからなっており，ショップ型であると呼ばれる．それぞれの仕事 J_iはオペレーションの集合 $O_{i1}, \cdots, O_{in_i}$からなり，それぞれのオペレーションの処理を行う機械が決められており，一般にオペレーション O_{ij}は機械 M_jで処理するとされる．ショップ型は各仕事におけるオペレーションの処理の順序に対する制約によって 3 つの場合に分けられる．

各仕事におけるオペレーションの処理をどのような順序で行ってもよいとき，オープンショップ型であるといい，$\alpha_1 = O$と記述する．

全ての仕事において定められたオペレーションの処理の順序が等しいとき，フローショップ型であるといい，$\alpha_1 = F$と記述する．

各仕事におけるオペレーションの処理の順序がその仕事ごとに定められているとき，ジョ

ブショップ型であるといい，$\alpha_1 = J$と記述する．

もし$\alpha_1 = PMPM$または$\alpha_1 = QMPM$ならば，MPM における等価並列機械もしくは一様並列機械であり，各仕事 J_iは$U_i \subseteq \{M_1, \cdots, M_m\}$ に属するいずれの機械においても処理することができる．

α_2は整数をとり，機械の数を表わしている．もし$\alpha_2 = k$ならば機械の台数が任意であるが固定された値 kを取ることを示し，$\alpha_2 = 1$ ならば単一機械問題である．$\alpha_2 = \phi$ならば機械の台数は任意である．

2.1.2 仕事の属性

仕事の属性には中断可能，各仕事の間の優先順位，最早開始時間，処理時間の制約，納期といったものがあり，$\beta = \beta_1\beta_2\beta_3\beta_4\beta_5$によって表わされる．

もし仕事またはオペレーションが中断可能であるとき，すなわち処理を中断し，それ以降同一の機械または異なる機械で処理を中断したところより再開することが可能であるとき$\beta_1 = pmtn$ と記述する．一般に仕事の処理の中断は複数回にわたっても良い．ただし現場においては仕事の処理の中断は少ないほうが良いとされ，中断回数が制約や目的関数となる場合もある．もし中断が許されない場合$\beta_1 = \phi$となる．

β_2は仕事間の処理の優先順位を表わす [63, 70]．一般に仕事の処理の優先順位は有向グラフ $G = (V, A)$ で表わされる．ここで節点集合 $V = \{1, \cdots, n\}$ は各仕事を表わし，有向枝 $(i, j) \in A$ は仕事 J_iが仕事 J_jより前に処理されなければならないことを示す．優先順位が任意に定められる場合$\beta_2 = prec$ とする．優先順位が特定のグラフで表されるときβ_2はそのグラフの名前となる．たとえば優先順位が木構造として示されるとき，$\beta_2 = tree$ となる．もし優先順位が存在しないとき$\beta_2 = \phi$となる．

$\beta_3 = r_i$のとき各仕事に対し最早開始時間，すなわちその仕事の処理を始めることができる時間が定められているを示している．もし$\beta_3 = \phi$ならば，全ての仕事は時刻 0 から開始可能，すなわち $r_i = 0$ である．

β_4は各仕事の処理に関する特殊な制約などを示す．例えばβ_4が $p_i = 1$ となっているならば全ての仕事の処理時間が単位時間であることを示す [58]．

β_5は納期に関する制約を示す．$\beta_5 = d_i$ならば各仕事は任意の納期を持つ．すなわちその仕事の処理が完了すべき時刻が指定されていることを示す．

2.1.3 目的関数

γには目的関数が記述される．目的関数には仕事の実行時間に関するもの，また仕事の実行にかかる何らかのコストに関するものに大別できる．またその最大値を最小化するもの，総和を最小化するものというようにも分けることができる．今ある実行可能なスケジュールπが存在し，仕事 J_iの処理の完了時間が C_iであるとする．そのとき以下のように表わされる 2 種類の目的関数が考えられる．

$$
\begin{aligned}
f_{\mathrm{max}}(\pi) &= \max\{f_i(C_i) \mid i = 1, \cdots, n\} && (2.1) \\
f_{\mathrm{sum}}(\pi) &= \sum_{i=1}^{n} f_i(C_i) && (2.2)
\end{aligned}
$$

このとき式 (2.1) は各仕事に対するコスト関数の最大値を示している．この形のものには最大完了時間や，最大納期遅れなどがある．式 (2.2) は各仕事に対するコスト関数の総和を示しており，この形のものには総滞留時間などがある．一般的なスケジューリング問題におけるモデル化ではこれらの関数を最小化するものとする．これらの形で最もよく使われる目的関数は以下のものがある．

最大完了時間

すべての仕事が終了する時間を示すものであり，以下の式で表される．

$$f_{\mathrm{max}} = \max\{C_i \mid i = 1, \cdots, n\} \tag{2.3}$$

総滞留時間

各仕事が処理をどれだけ滞留されているか示すものであり，以下の式で表される．

$$f_{\mathrm{sum}} = \sum_{i=1}^{n} C_i \tag{2.4}$$

重み付き総滞留時間

滞留時間に対し各仕事 J_iに重み w_iが与えられているものであり，以下の式で表される．

$$f_{\mathrm{weighted\_sum}} = \sum_{i=1}^{n} w_i C_i \tag{2.5}$$

ここで w_iは各仕事 J_iに定義された重みである．

また各仕事 J_iに対し納期 d_iが設定されているとき，納期と仕事の処理の完了時間とのずれ L_iを

$$L_i = C_i - d_i \tag{2.6}$$

と定義すると，以下の目的関数が考えられる．

納期遅れ仕事数

仕事の処理の完了時刻が納期より遅れてしまう仕事の総数を示し，以下の式で表される．

$$f_{\mathrm{L\_number}} = |\{J_i \mid L_i > 0\}| \tag{2.7}$$

最大納期ずれ

仕事の完了時間が納期より遅れてしまうときの最も大きい遅れであり，以下の式で表される．

$$f_{\mathrm{L\_max}} = \max_{1 \leq i \leq n} L_i \tag{2.8}$$

これらの目的関数は一般に同時に最適化できず，むしろこれらのトレードオフが問題となる．例えば最大完了時間や総滞留時間は工場の生産ライン，すなわち生産者側の要求であり，納期遅れ仕事数や最大納期ずれは顧客側の要求である．生産現場においてはこのように異なる立場からの要求が複数あり，これらにバランス良く応えていくことが必要となる．

2.1.4　資源

生産現場において仕事の処理を行うときには，何らかの資源の制約を受けることが多い．ここでいう資源には人員や工具，空間，資金，エネルギーなどが挙げられる．現実的なスケジュールを考えるためにはこのような資源の制約を考慮する必要があり，またそれ自体が目的関数となることも多い [2, 8, 9, 10, 15, 21, 51]．資源はその属性によって離散量と連続量または再生可能か再生不可能という2つの方法で分類される．

離散量と連続量という分類は資源の単位の属性によるものである．

離散量 (Discrete)

資源が離散量として与えられる．例えば人員，工具等である．

連続量 (Continuous)

資源が連続量として与えられる．電力や燃料などのエネルギー，また資本等も連続量と見なされることが多い．

再生可能か再生不可能かという分類は資源を使用したときの属性によるものである．

再生可能 (Renewable)

再生可能とは一度用いた資源を後で再度利用可能であることであり，人員，工具等がこれにあたる．

再生不可能 (Nonrenewable)

再生不可能とは一度用いた資源は再度利用することが不可能なことであり，資金やエネルギー等がこれにあたる．

これらの資源は仕事の処理を行うときに消費され，スケジューリングを考える上での大きな要素であり，スケジューリング問題においては資源の消費量を制約とするか，もしくは目的関数として考える．

2.2 スケジューリングにおける現状

産業の発展に伴い様々な現場における生産工程も複雑多岐にわたるようになり，スケジューリング理論に対する要請も複雑化してきた．また産業に対する社会の意識が変わり，効率一辺倒の生産工程から労働者の労働環境や環境に与える影響を考慮する必要も出てきている．スケジューリング理論に対する要求には以下のようなことが考えられる．

複雑化した生産工程への対応

嗜好の多様化に伴い生産工程は少品種大量生産から多品種少量生産に移行している．これに伴い生産工程も複雑になっている．このような生産工程を考えるためには新たなモデルが必要であり，またそれに対する解法の開発も必要である．

環境問題への対応

近年，産業の環境に対する影響が非常に大きく問題視されている．企業はこれまでのようにコストの削減の追求のための効率化のみではなく，環境への配慮から省資源，省力化という効率化を考える必要がある．

生産技術の知識の共有化

企業の活動は国際化しつつあり，海外に生産の拠点を持つ企業も多い．そのような生産拠点に対し，生産の効率化と生産物の質の維持を図るためには生産工程の管理の知識を徒弟制度的に習得するような形ではなく，理論として体系化し伝えられるような形にする必要がある．

生産システムの情報化

近年，生産工程に限らず企業活動全体が情報化され，それらはコンピュータによって制御されている．そこで生産工程の管理も経験的な知識ではなく，理論として体系化されコンピュータを用いて処理できるような形にする必要がある．

従来の多くの現場ではスケジューリングはそれに精通した人員によって行われており，それによって得られるスケジュールは十分に実用に足るものであった．しかしこれらの知識，経験を後継者に伝えていくことが難しく，また企業の活動全体が情報システムによって管理されるようになりつつあり，生産技術が発達するにつれ生産の規模は肥大化し，かつ生産の工程もますます複雑になってきており，スケジューリングに対する理論的基盤の必要性がますます高まってきている．そこでスケジューリング理論における手法もさまざまなものが開発されてきており，大きく以下のようなものに分けられる [59]．

構成的アルゴリズム

本来解くべき問題に条件を付加した特殊な問題を構成し，問題の特殊性を利用して最適解を効率よく求めるアルゴリズムをいう [56]．代表的なものに前述の Johnson による n 仕事 2 機械フローショップ問題に対する最適解法などがある．

数理的解法

現実の問題に対して数理的なモデルを構築し，数理的な解法を用いて解を得る．整数計画法，分枝限定法，動的計画法などの手法があり，得られる解の最適性が保証される．しかし現実における複雑な制約をすべてモデル化して解法を構成することが困難であり，また問題の規模の増大につれて計算時間が指数的に増大する．そのため探索範囲を制限したり，計算時間の上限を設定することによって近似解法として利用するというような研究も行われている [14, 19, 20, 41, 78]．

発見的解法

発見的解法は現実の問題が持っている構造や性質を用いて短時間で良好な解を求めるものであり，多くの現場で用いられている．位置的重みづけ法，最大要素時間法，生産比逆数法，目標追跡法，拡張 Johnson 法，最低発注単価法，最低トータルコスト法などがある [60, 76, 96].

メタ・ヒューリスティックス

メタ・ヒューリスティクスとは組合せ最適化問題に対しての発見的解法の枠組みであり，従来の数理的，分析的手法に基づく厳密解法に対し，ある暫定解からより良い解を発見的に探索するための方法論である．主なものに遺伝的アルゴリズム，シミュレーテッド・アニーリング，タブ・サーチなどがあり，さまざまな制約を考慮したモデルに適用することが比較的容易で，探索能力，計算時間の面で有効であることが報告されている [6, 29, 30, 31, 35, 53, 77, 89, 90].

AI を用いた解法

従来スケジューリングを行ってきた熟練者の知識をデータベースとして利用し，AI を用いてスケジューリングを行おうとするものである．複雑な制約を持つ場合に有効な方法であるが，熟練者の知識をデータベース化することが困難である [39].

2.3 多目的最適化

一般に最適化問題と呼ばれる問題の多くは単一の目的関数を扱うことが多い．しかし現実における最適化問題の多くは本質的に多くの評価基準を含んでおり，一元的な価値基準で評価を行うことができる場面のほうがむしろまれである．そのような複数の評価基準に対し最適化を考える問題を多目的最適化問題 (Multi Criteria Optimaization Problem) と呼ぶ．複数の評価基準を持つ問題を数学的にとり扱うための研究が行われ始めたのは比較的最近のことである．多目的最適化問題においては，全ての評価基準を最適化するような解が存在することはまれであり，何をもって最適な解であると定義するかということも問題である．

多目的最適化問題は一般に以下のように定義される．

定義 2.1 $\mathcal{F}$を $\mathbf{R}^n$の閉集合

$$\mathcal{F} = \{x \mid g_i(x) \leq 0; \quad i = 1, \cdots, n\} \tag{2.9}$$

とするとき，$x(x \in R^n)$ についての p 個の関数 $f_k(x)(k = 1, \cdots, p)$ を，$x \in \mathcal{F}$の範囲で最小化せよ.

この問題では一般に全ての目的関数 $f_k(x)$ を同時に最小にすることはできない．むしろこれらの間にトレードオフの関係があることが問題となる．よって多目的最適化問題において解の評価をいかに決定するのかということに対しさまざまな手法が提案されてきた.

2.3.1　2項関係としての選好

まず二つの要素における選好関係を考える．ある要素x, yに対し以下のうちいずれかただ一つの関係が成立するとする.

(1) x はyより良い．これを $x < y$で表わす.

(2) x はyより悪い．これを $x > y$で表わす.

(3) x とyは同じ．これを $x = y$で表わす.

(4) x とyは (1)，(2)，(3) のいずれでもない．これを $x?y$で表わす．この場合 x とyの間の選項関係は不確定または未定である.

今ある最適化問題においてある解が他のすべての解より良いかもしくは同じであるとき，すなわち実行可能解の集合を $\mathcal{X}$としたときに，ある解 x に対し

$$x < y \quad \text{または} \quad x = y \quad \{y \mid y \in \mathcal{X} - \{x\}\} \tag{2.10}$$

が成立するとすると，この問題における最適な解は x であるといえる．単一目的の最適化問題においてその解は式 (2.10) を満たす.

しかし多目的最適化問題の場合，その目的関数の値における選好関係がそれぞれの目的関数によって異なるために最適解を一意に定めることができない．そこで多目的最適化問題に対し最適解に変わる新しい概念として非劣解が Palate によって考えられた [87].

2.3.2 非劣解

n 次元ベクトル$\boldsymbol{a} = (a_1, \cdots, a_n), \boldsymbol{b} = (b_1, \cdots, b_n)$ に対し以下のような関係を定義する

$$\boldsymbol{a} \leq \boldsymbol{b} \quad \Leftrightarrow \quad a_i \leq b_i, \quad i = 1, \cdots, n \tag{2.11}$$

$$\boldsymbol{a} < \boldsymbol{b} \quad \Leftrightarrow \quad a_i < b_i, \quad i = 1, \cdots, n \tag{2.12}$$

ここで

$$\boldsymbol{f}\ (x) = (f_1(x), \cdots, f_n(x)) \tag{2.13}$$

とすると，多目的最適 (最小) 化問題における解の優越関係は次のように定義される．

定義 2.2 $x_1, x_2 \in \mathcal{F}(\mathcal{F}$: 実行可能解集合) とする．

- $\boldsymbol{f}\ (x_1) \leq \boldsymbol{f}\ (x_2)$ かつ$\boldsymbol{f}\ (x_1) \neq \boldsymbol{f}\ (x_2)$ のとき，x_1は x_2に優越するという．
- $\boldsymbol{f}\ (x_1) < \boldsymbol{f}\ (x_2)$ のとき，x_1は x_2に強い意味で (strong sense) 優越するという．

もし x_1が x_2に優越しているならば x_1は x_2より良い解であると考えることができる．したがって多目的最適化問題において他の全ての解に優越されないような解を最適な解として選択することは妥当であると思われる．このことより多目的最適化問題における非劣解は以下の様に定義される．

定義 2.3 非劣解 (nondominated solution)

$x_0 \in \mathcal{F}$とする．

- x_0に強い意味で優越する解 $x \in \mathcal{F}$が存在しないとき，x_0は弱非劣解という．
- x_0に優越する解 $x \in \mathcal{F}$が存在しないとき，x_0は非劣解という．

例 2.4 ある f_1, f_2という 2 つの目的関数を持つ 2 目的最適化 (最小化) 問題において実行可能領域が図 2.1のように示されたとする．ここで解 x_1, x_2に対し

$$f_1(x_1) = f_1(x_2), \quad f_2(x_1) < f_2(x_2) \tag{2.14}$$

が成立しており，x_1が x_2に優越している．

また x_3に対して

$$f_1(x_1) < f_1(x_3), \quad f_2(x_1) < f_2(x_3) \tag{2.15}$$

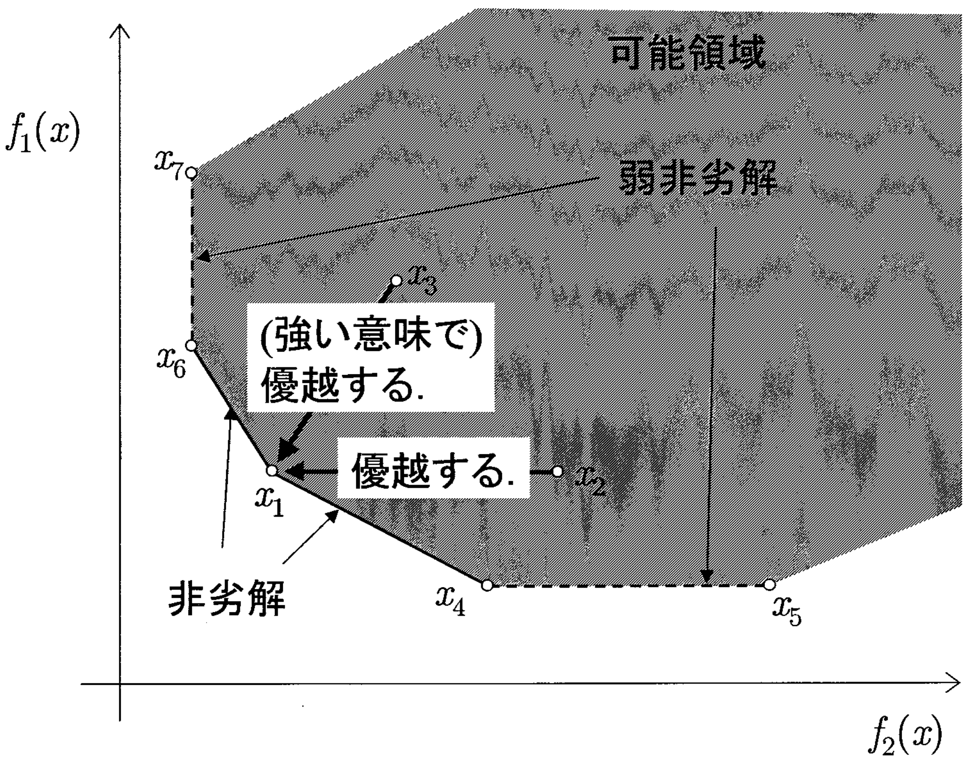

図 2.1　非劣解

が成立しており，x_1は x_3に強い意味で優越している．

x_1と x_4，x_1と x_6，x_4と x_5，x_6と x_7をそれぞれ結ぶ直線は弱非劣解になっており，とくに x_1と x_4，x_1と x_6をそれぞれ結ぶ直線は非劣解になっている．

2.3.3　多目的最適化における意思決定

多目的最適化問題においてすべての評価基準が最適であるような解が存在することはまれであり，一般にはさまざまな評価基準がトレードオフの関係にあるので最適な解を一意に得ることはできない．そこで意思決定の場面において意思決定者の意志ができる限り反映されるような決定を行う手法が考えられてきた [3, 55].

重みパラメータ法

各目的関数 f_iに重み w_iをつけ，その和を新たに目的関数とする．すなわち

$$\sum_{i=1}^{n} w_i f_i(\boldsymbol{x}) \tag{2.16}$$

ただし

$$\begin{aligned} w_k &\geq 0, \quad k=1,\cdots,n & (2.17) \\ \sum_{k=1}^{n} w_k &= 1 & (2.18) \end{aligned}$$

とする．もし実行可能領域が凸の場合はこの方法によって非劣解の 1 つを得ることができる．しかし実行可能領域が非凸の場合は双対ギャップが生じるので得た解が非劣解であることは保証されない．

ε制約法

一つの目的関数 f_lのみを残して，他は定数ε_kを用いて制約としたスカラー最小化問題

$$\min_{x \in \mathcal{F} \cap \varepsilon} f_l(\boldsymbol{x}) \tag{2.19}$$

を考える．ただし

$$\varepsilon = \{x \mid f_k(\boldsymbol{x}) \leq \varepsilon_k, k=1,\cdots,l-1,l+1,\cdots,p\} \tag{2.20}$$

である．このときε_kを変化させて解を繰り返し求めると可能領域が非凸の場合でも，非劣解をすべて求めることができる．

辞書式配列法

目的関数に優先順位をつけ，その優先順位にしたがって解を絞り込んでいく方法である．例えば，f_1のみによって順位を決め，f_1が同じものについては f_2により，さらに同じものには f_3により，というように解を求める方法である．この方法では，明らかに先に採用される目的関数が重視されるので，その優先順位の決め方には意思決定者の選好が反映されることになる．

目標計画法

まず各目的関数の理想値$\hat{f}_k$を決め，f空間上の理想点$\hat{f}$を定める．一般に$\hat{f}$は fの空間の可能領域外にある．次に f空間上の各点について$\hat{f}$からのずれの量を

$$||f(\boldsymbol{x}\ ) - \hat{f}||_q = \left\{ \sum_{k=1}^{p} |f_k(\boldsymbol{x}\ ) - \hat{f}_k|^q \right\}^{1/q} \tag{2.21}$$

と定義する．これをリグレット関数と呼ぶ．そしてリグレット関数を最小にする$\boldsymbol{x}$ を選好解とする．この方法では意思決定者は理想点$\hat{f}$およびパラメータ qをあらかじめ決定する必要がある．

目標到達法

目標計画法の場合と同様に理想点$\hat{f}$を定め，以下の問題を考える．

目的関数

$$r \rightarrow \text{minimize} \tag{2.22}$$

制約条件

$$f(\boldsymbol{x}\ ) - r\boldsymbol{w} \ \leq \hat{f} \tag{2.23}$$

$$x \in \mathcal{F} \tag{2.24}$$

この問題を解くことにより選好解を求める．ここでは rはスカラー，$\boldsymbol{w}$ (>0) は式 (2.18) をみたす一種の重みパラメータである．なおこの方法では理想点$\hat{f}$および方向 $\boldsymbol{w}$ に意思決定者の選好が反映される．

対話的手法

上に述べた方法ではいくつかのパラメータや関数値をあらかじめ与えることによって定められる単一目的最適化問題を解いて選好解を求めるものである．しかしこれらの値は意思決定者自身にとっても未知で，あらかじめ設定することは難しい．またこのパラメータを解の決定過程を通して常に一定にすることにも根拠が無い．そこで繰り返し型の計算によって解の探索途中の各段階で対話的に意思決定者に質問しながら，その選好を取り入れる方法である．

以上，多目的最適化における意思決定について代表的な手法を幾つか述べてきたが，これらの方法においては意思決定者は，あらかじめ各目的関数について優先順位やトレードオフの関係についての予測が必要であり，それらを適切に行わない限り意思決定者の望む解を得ることは必ずしも容易でない．

2.3.4 多目的スケジューリング問題

2.3.2節における非劣解の定義は多目的スケジューリング問題においても適用することができる．ただしスケジューリング問題においては，各目的関数からなるベクトルが非劣解となっていることだけでなく，その目的関数を実現するスケジュールを構築する必要がある．この非劣解と非劣スケジュールの関係を説明するために以下の例を考える．

例 2.5 単一機械 3 仕事最大納期遅れ $L_{\max}$，総滞留時間 C_{sum}最小化問題を考える．各仕事 J_iは表 2.1に示される処理時間 p_iと納期 d_iを持つとする．

表 2.1 各仕事の処理時間，納期

	J_1	J_2	J_3
p_i	1	2	3
d_i	6	5	3

また全ての仕事は時間 0 から処理可能であるとする．単一機械において最大納期遅れと総滞留時間それぞれを最小にするアルゴリズムは知られており，最大納期遅れについては納期の非減少順に処理を行い，総滞留時間については処理時間の非減少順に処理を行えば良い．

S_{ijk}を $\{J_i, J_j, J_k\}$ の順番で仕事を処理するスケジュールであるとし，その時の最大納期遅れを $L_{\max}(S_{ijk})$ とし，総滞留時間を $C_{\mathrm{sum}}(S_{ijk})$ すると，最大納期遅れについては S_{321}が最小となり，その時の目的関数はそれぞれ

$$(L_{\max}(S_{321}), C_{\mathrm{sum}}(S_{321})) = (0, 14) \tag{2.25}$$

となる．また総滞留時間については S_{123}が最小となり，その時の目的関数はそれぞれ

$$(L_{\max}(S_{123}), C_{\mathrm{sum}}(S_{123})) = (1, 10) \tag{2.26}$$

となる．その他のスケジュールについては表 2.2のようになる．

表 2.2　各スケジュールの納期遅れ，総滞留時間

	S_{123}	S_{132}	S_{213}	S_{231}	S_{312}	S_{321}
$L_{\max}$	3	1	3	3	1	0
C_{sum}	10	11	11	13	13	14

このとき$\boldsymbol{f}(x) = (L_{\max}(x), C_{\text{sum}}(x))$ とすると，優越関係は

$$\boldsymbol{f}(S_{123}) \leq \boldsymbol{f}(S_{213}) \tag{2.27}$$
$$\boldsymbol{f}(S_{123}) \leq \boldsymbol{f}(S_{231}) \tag{2.28}$$
$$\boldsymbol{f}(S_{132}) \leq \boldsymbol{f}(S_{213}) \tag{2.29}$$
$$\boldsymbol{f}(S_{132}) < \boldsymbol{f}(S_{231}) \tag{2.30}$$
$$\boldsymbol{f}(S_{132}) \leq \boldsymbol{f}(S_{312}) \tag{2.31}$$
$$\boldsymbol{f}(S_{213}) \leq \boldsymbol{f}(S_{231}) \tag{2.32}$$
$$\boldsymbol{f}(S_{231}) \geq \boldsymbol{f}(S_{312}) \tag{2.33}$$

となり残りは不確定である (図 2.2参照)．弱非劣解は

$$\{\boldsymbol{f}(S_{123}), \boldsymbol{f}(S_{132}), \boldsymbol{f}(S_{213}), \boldsymbol{f}(S_{312}), \boldsymbol{f}(S_{321})\}$$

非劣解は

$$\{\boldsymbol{f}(S_{123}), \boldsymbol{f}(S_{132}), \boldsymbol{f}(S_{321})\}$$

となり，

$$S_{123}, S_{132}, S_{213}, S_{312}, S_{321}$$

は弱非劣スケジュール，

$$S_{123}, S_{132}, S_{321}$$

は非劣スケジュールとなる．

スケジューリング問題の研究は単一の目的関数を持つ問題については数多く行われているが，多目的スケジューリング問題については，あまり行われておらずまた単一機械に対す

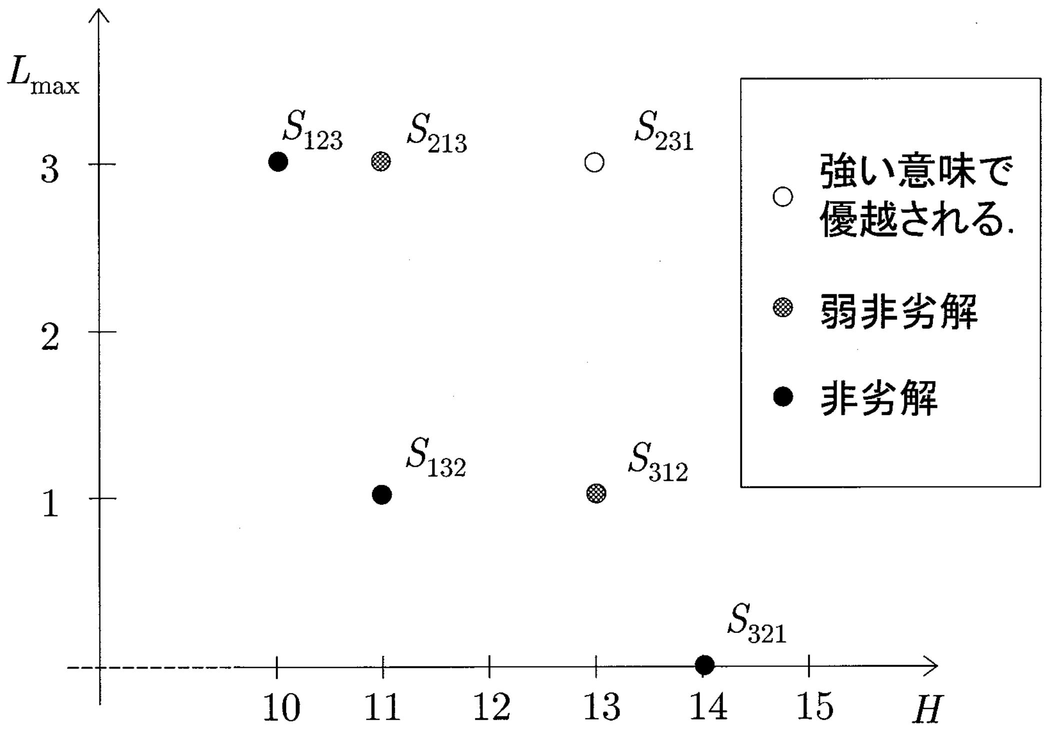

図 2.2 非劣解

るものがほとんどである [4, 27, 28, 34, 37, 38, 73]. 近年になっていくつかの数理的解法の研究が行われ [16, 81, 84, 92], またメタ・ヒューリスティックの各手法, とりわけ遺伝的アルゴリズムを用いた解法が比較的多く発表されている [79, 91].

第 3 章

2 目的 MPM オープンショップスケジューリング問題

3.1　緒言

本章では MPM において最大完了時間と最大納期遅れという二つの目的関数を持つオープンショップスケジューリング問題を扱う．

近年，さまざまな技術の進歩に伴って顧客の要求が多岐に渡るようになり，加えて生産技術が進歩したことによって生産の形態は少品種大量生産から多品種少量生産に移行しつつある．これを実現するためには従来のような一つのオペレーションのみを実行できるだけの機械によるものではなくさまざまなオペレーションに対応できるような機械による生産ラインとしなければならない．このような生産現場でよく用いられる機械環境に FMS(Flexible Manufacturing Systems) がある．

Buzacott と Yao は FMS を以下のように定義している [22].

定義 3.1 *FMS* とは *NC* 工作機械群と加工品の操作をコンピュータによって自動管理し，各製品に要求される多種多様な処理を自動的にかつ自立的に行う統合処理システムである．

ここでいう NC 工作機械とは，取り外し変更可能な工具をもち，その工具を使い分けることによってさまざまな処理を行うことができる機械のことである．この工具は一般に機械間で共有され，同種の工具を使うことができる機械は同じ処理を行うことができる．

FMS におけるスケジューリング問題の研究は多く行われており [33, 67, 75]，多機能機

械(Multi Purpose Machine:MPM)は従来考えられてきた機械環境の一般化としてFMSの研究から考案された.

MPMにおいてはそれぞれの機械はいくつかの異なるタイプの処理を行うことができるとする. たとえばMPMにおけるショップ型スケジューリング 問題は以下の様に定義される. 仕事J_iはn_i個の処理$O_{ij}(j=1,\cdots,n_i)$からなり, またm台の機械$M_k(k=1,\cdots,m)$があり, 各オペレーションO_{ij}は機械集合$U_{ij}\subseteq\{M_1,\cdots,M_m\}$におけるいずれの機械によっても処理できるものとする.

MPMははじめにBruckerとSchlieによって研究された[18]. 彼らは2つの仕事を持つMPMジョブショップ問題を考え, 最大完了時間を最小にする最適スケジュールを構成する多項式時間のアルゴリズムを与えている. またこれらの結果よりJurishによってm機械MPMジョブショップ問題における滞留時間の最大値の下界が与えられている. しかしショップ型のMPM問題においてはわずかな研究しか行われていない[17, 50].

一方LawlerとLabertoulleは非一様並列機械における最大完了時間と最大納期遅れを各々単一の目的関数とする問題について研究し, これらの問題を解くために線形計画問題を定式化している[64]. また彼らの結果を用いてIshiiは2目的非一様機械スケジューリング問題を考察している[42]. 本章ではあらたに最大納期遅れ$L_{\max}$と最大完了時間$C_{\max}$というMPMにおける2目的オープンショップ問題を考え, [42]の結果を拡張することによって線形計画問題を定式化し, それによって得られる最適解の非劣性について考察する. またその最適解より実行可能スケジュールを構築する方法を示す. さらにFMSに対するモデルの拡張を行い, 資源の消費を考慮した3目的の問題への拡張を行う.

はじめに3.2節において我々の考察する問題を与える. そして3.3節において, 線形計画問題への定式化を考え, 3.4節において実行可能スケジュールを構築する方法について述べる. 3.5節において, その結果をFMSに適用する方法について述べる. また3.6節において各機械において処理を行うことによって資源を消費するという環境における資源の消費コストと最大納期遅れ, 最大完了時間という3目的の問題を考える. 最後に結論を述べる.

3.2　2目的MPMオープンショップスケジューリング問題

以下の2目的MPMオープンショップ 問題$OMPM\mid pmtn,d_i\mid C_{\max},L_{\max}$を考える.

- n個の仕事$J_1,\cdots,J_n$がある.

- m 台の機械 $M_1, \cdots, M_m$がある.
- それぞれの仕事 $J_i(i = 1 \cdots n)$ は n_i個のオペレーション $O_{i1}, \cdots, O_{in_i}$からなる.
- それぞれの O_{ij}は機械集合 $U_{ij} \subseteq \{M_1, \cdots, M_m\}$ のうちの 1 つの機械で処理され，その処理時間を p_iとする.
- 任意のオペレーション O_{ij}はどのような順序でも処理することができる.
- 各オペレーションの処理は開始以降，任意の時間で中断することができ，それ以降いつでも，U_{ij}に属するいずれの機械においても再開可能である.
- それぞれの仕事は一度にただ一つの機械でのみ処理することができ，それぞれの機械は一度に一つのオペレーションを処理することができる.
- それぞれの仕事 J_iには納期 d_iが定められている. あるスケジュールにおいて C_iを仕事 J_iの全てのオペレーションが完了した時刻とすると，仕事 J_iの納期遅れは $L_i = C_i - d_i$ となり，最大完了時間は

  $$C_{\max} = \max_i C_i \tag{3.1}$$

  最大納期遅れは

  $$L_{\max} = \max_i L_i \tag{3.2}$$

  と表わすことができる. この最大完了時間と最大納期遅れをできる限り小さくするようなスケジュールを求めることがこの問題の目的である.

この問題においては通常 $C_{\max}$と $L_{\max}$を同時に最適化するような解は存在しない場合が多い. そこでこの問題における非劣解の集合を探すことが目的となる.

3.3 線形計画問題としての定式化

本節では，前節で述べた問題を線形計画問題へ定式化する. まずパラメータ yを導入し，パラメータ yによって最大完了時間 $C_{\max}$に対し $C_{\max} \leq y$という制約を加える. ここで最大完了時間は全ての仕事に対する共通の納期と捉えることができる. そしてその制約の下で最大納期遅れ $L_{\max}$が最小となるような解を求める.

各仕事 J_iの納期 $d_i(i=1,\cdots,n)$ を非減少順に並べ直し，必要ならば添え字番号を付け直し $d'_1 < d'_2 < \cdots < d'_q$とする．ただしここでは同じ納期は一つの納期にまとめるものとし，qは異なる納期の数とする．また p を $d'_l \leq y$が成立するような lの最大値とする．すなわち $d'_p \leq y$かつ $d'_{p+1} > y$とする．また l_iを $d'_l = d_i$となる lの値とする．x_{ij}^{kl}を仕事 J_iの処理 O_{ij}が機械 M_kによって区間 $[d'_{l-1}, d'_l)$ において処理される時間の合計とする．

以下の線形計画問題 $P_1^p(y)$ を考える．

$P_1^p(y)$:

目的関数

$$z \longrightarrow \text{minimize} \tag{3.3}$$

制約条件

$$\sum_{j,k} x_{ij}^{kl} \leq d'_l - d'_{l-1}, \quad l=2,\cdots,p-1, i=1,\cdots,n \tag{3.4}$$

$$\sum_{i,j} x_{ij}^{kl} \leq d'_l - d'_{l-1}, \quad l=2,\cdots p-1, k=1,\cdots m \tag{3.5}$$

$$\sum_{j,k} x_{ij}^{k1} \leq d'_1 + z, \quad i=1,\cdots,n \tag{3.6}$$

$$\sum_{i,j} x_{ij}^{k1} \leq d'_1 + z, \quad k=1,\cdots,m \tag{3.7}$$

$$\sum_{j,k} x_{ij}^{kp+1} \leq y - d'_p, \quad i=1,\cdots,n \tag{3.8}$$

$$\sum_{i,j} x_{ij}^{kp+1} \leq y - d'_p, \quad k=1,\cdots,m \tag{3.9}$$

$$z \geq y - d'_p \tag{3.10}$$

$$\sum_{k,l\leq l_i} x_{ij}^{kl} = p_{ij}, \quad j=1,\cdots,n_i,\ i=1,\cdots,m \tag{3.11}$$

$$x_{ij}^{kl} = 0, \quad \begin{array}{ll} l = l_i \cdots p+1, & i=1,\cdots,n \\ j=1,\cdots,n_i, & k=1,\cdots,m \end{array} \tag{3.12}$$

$$x_{ij}^{kl} = 0, \quad \begin{array}{ll} l = 1 \cdots l_i, & i=1,\cdots,n \\ j=1,\cdots,n_i, & k=\{k \mid M_k \notin U_{ij}\} \end{array} \tag{3.13}$$

$$x_{ij}^{kl} \geq 0, \quad \begin{array}{ll} l = 2 \cdots l_i, & i=1,\cdots,n \\ j=1,\cdots,n_i, & k=1,\cdots,m \end{array} \tag{3.14}$$

ここで zは最大納期遅れを表わす．スケジュールの開始時刻を 0 から$-z$に早め，全ての仕事が納期に間に合うようなスケジュールを組むことができれば，そのスケジュールの開始時刻を 0 としたときに各仕事の処理が納期から遅れる時間は高々zとなる．式 (3.4), (3.5) は区間 $[d'_{l-1}, d'_l)$ においてそれぞれ機械または仕事における処理時間の合計がその区間を超えない制約を示す．同様に式 (3.6)，(3.7) は区間 $[-z, d'_1)$ に対し，式 (3.8)，(3.9) は区間 $[d'_p, y)$ に対し，それぞれの機械または仕事における処理時間の合計がその区間を超えない制約を示す．式 (3.11) が成立することでは全てのオペレーションの処理は終了していることを保証し，(3.12) は納期以降にオペレーションが処理されていないことを保証する．(3.13) はオペレーションがその処理を行うことができる機械において処理されていることを保証する．(3.14) は非負条件である．

この線形計画問題 $P_1^p(y)$ の最適値は前節の問題において $C_{\max} \leq y$としたときの最大納期遅れ $L_{\max}$の最小値となっていることは明らかである．$f_1^p(y)$ を $P_1^p(y)$ における zの最適値であるとする．そのとき以下の定理 3.2，3.3 が成立する．

定理 3.2 $f_1^p(y)$ は区間 $[d'_l, y_l]$ $(l = r, \cdots, q)$ において yに対し凸関数であり，区分的に線形で，かつ yに対し非増加関数である．ただし y_lは $f_1^p(y) = y - d'_l$が成立する最小の yである．

証明

yが増えるにつれて，$P_1^p(y)$ の可能な領域は増えるので $f_1^p(y)$ の非増加性は明らかである．$P_1^p(y)$ の双対問題を考えることによって区分的に線形であることは明らかである．最後に凸性について $(z_1, x_{1ij}^{kl}), (z_2, x_{2ij}^{kl}), (z_\lambda, x_{\lambda ij}^{kl})$ をそれぞれ $P_1^p(y_1), P_1^p(y_2), P_1^p(\lambda y_1 + (1-\lambda)y_2)$ の最適解であるとすると，

$$\begin{aligned} \lambda f_1^p(y_1) + (1-\lambda) f_1^p(y_2) &= \lambda z_1 + (1-\lambda) z_2 \\ &\geq z_\lambda = f_1^p(\lambda y_1 + (1-\lambda) y_2) \qquad (3.15) \end{aligned}$$

となり，z_λの最適性より，

$$\left(\lambda z_1 + (1-\lambda) z_2, \lambda x_{1ij}^{kl} + (1-\lambda) x_{2ij}^{kl}\right) \qquad (3.16)$$

は $P_1^p(\lambda y_1 + (1-\lambda) y_2)$ の可能領域内に存在することによって凸であることが示された．

証明終

定理 3.3 $f_1^p(y), y \in [d'_l, y_l](l = r, \cdots, q)$ の単調増加域と対応するyの変域において非劣スケジュールのスケジュールベクトルは $(y, f_1^p(y))$ の形で与えられる．ここでrは$d'_k \leq C^*_{\max}$を満たす最大のkの値であり，一般性を失わずに$C^*_{\max} > d'_1$を仮定する．

証明

$C_{\max}$が固定されているとき納期遅れのみを考える単一目的の問題となるので$f_1^p(y)$ は $L_{\max}$の最小値となる．区間 $[y_p, d'_{l+1})$ において $z = y - d'_l$が成立する．よってyが増加してもzの最適値は減少することはない．なぜならzは常に区間において基底変数で $(y, f_1^p(y))$ はベクトル $(y_l, f_1^p(y_l))$ に優越する．よって区間$[d'_l, y_l]$ のみを考えればよい．

証明終

定理 3.2，3.3よりyを実行可能領域で動かすことにより前節の2目的問題の非劣解集合を求めることができる．

3.4　実行可能スケジュールの構成

ここでは $P_1^p(y)$ の最適解 x_{ij}^{kl}を用いて実行可能なスケジュールを構成する方法について述べる．

まずそれぞれの区間 $[d'_{l-1}, d'_l)(l = 1, \ldots, p+1)$ について，同じ機械に割り当てられている同じ仕事に属するオペレーションを合わせて1つのオペレーションとする．すなわち仕事$\bar{J}_{ik}^l(i = 1, \ldots, n, k = 1, \ldots, m)$ を考え，仕事$\bar{J}_{ik}^l$の機械 M_k上での処理時間$\bar{p}_{ik}^l$

$$\bar{p}_{ik}^l = \sum_j x_{ij}^{kl}, \quad i = 1, \ldots, n, k = 1, \ldots, m \tag{3.17}$$

とする．こうすることにより，上記の問題はm個のオペレーションを持つn仕事m機械の中断可能なオープンショップ問題 $O \mid pmtn \mid -$に帰着することができる．この問題はLawlerとLabettoulleによりすでに考察されている [65]．具体的には，各区間 $[d'_{l-1}, d'_l)(l = 1, \ldots, p+1)$において，$n \times m$ 行列 T^l を

$$T^l = \begin{bmatrix} \bar{p}_{11}^l & \cdots & \bar{p}_{1m}^l \\ \vdots & \ddots & \vdots \\ \bar{p}_{n1}^l & \cdots & \bar{p}_{nm}^l \end{bmatrix} \tag{3.18}$$

とし，$C^l_{\max}$を

$$C^l_{\max} = d'_l - d'_{l-1} \tag{3.19}$$

とする．

また Y^lを $m \times m$ の対角行列で各要素 y_{kk}が機械 M_kにおける遊休時間の合計であるとする．すなわち，

$$Y^l = \begin{bmatrix} y_{11} & \cdots & \cdots & 0 \\ 0 & y_{22} & 0 & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & y_{mm} \end{bmatrix} \tag{3.20}$$

とする．ただし

$$y_{kk} = C^l_{\max} - \sum_{i=1}^{n} \bar{p}^{kl}_i \tag{3.21}$$

とする．

$m \times (m+n)$ 行列 V^lを $V^l = \left[T^l, Y^l\right]$とし，$V^l = \{v_{ij}\} \quad (1 \leq i \leq m, 1 \leq j \leq m+n)$ とする．集合 U^lを行列 V^lの要素のうち T^lの i 列が$\sum_{k=1}^{m} \bar{p}^l_{ik} = C^l_{\max}$となるような各列から 1 つの要素を取り出し，残りの列，行からちょうど要素が 1 つずつになるように取り出したものの集合とする．U^lは長さ$\delta > 0$ の部分スケジュールを構成するのに用いる．最適なスケジュールはこの部分スケジュールを連結することによって得られる．以下にその手順を示す．

for $l \leftarrow 0$ to p do

begin

  while $C^l_{\max} > 0$ do

  begin

    $\bar{p}^l_{ik}$を計算し，行列 V^lを構成する．

    V^lより U^lを求める．

    $v_{\min} \leftarrow \min\{v_{ij} \in U^l\}$, $v_{\max} \leftarrow \max_j\{\sum v_{ij} \mid v_{ij} \notin U^l, \forall i\}$

    if $C^l_{\max} - v_{\max} \geq v_{\max}$ then

      $\delta \leftarrow v_{\min}$

    else

$\delta \leftarrow C^l_{\max} - v_{\min}$

S $\leftarrow$ S+Construct_schedule(U, δ)

$C^l_{\max} \leftarrow C^l_{\max} - \delta$

for $v_{ij} \in U^l$ do

begin

$v_{ij} \leftarrow v_{ij} - \delta$

end

end

end

集合U^lより実行可能なスケジュールを構成することができる．集合U^lの各要素$\bar{p}^{kl}_i$に対し，機械M_kで仕事J_iを時間$\min\{\bar{p}^{kl}_i, \delta\}$の間だけ処理するものとし，集合$U^l$の要素に対し$\bar{p}^{kl}_i$を$\max\{0, \bar{p}^{kl}_i - \delta\}$に置き換え新たな行列$T^l$を構成し$C^l_{\max} \leftarrow C^l_{\max} - \delta$とする．これを繰り返すことにより実行可能なスケジュールを構成することができる．

3.5 FMSへの適用

FMSは工具群を持つ多機能工作機械からなる統合生産システムとして定義することができる．一般にFMSにおいて用いられるNC工作機械では機械が特定の処理を行うために工具が必要であり，その工具は複数の機械間で交換して用いることができる．またFMSにおいてはNC工作機械は複数台あるのが一般的である．前節におけるMPMモデルはそのモデル内の機械をFMSにおける工具と考えることによってFMSのモデルとみなすことができるが，そのままでは各機械に対する制約が満たされない．このため以下のような問題を考える．

- n個の仕事$J_1, \cdots, J_n$がある．
- m台の機械$M_1, \cdots, M_m$がある．
- t個の工具$T_1, \cdots, T_t$がある．
- それぞれの仕事$J_i (i = 1 \cdots n)$はn_i個のオペレーション$O_{i1}, \cdots, O_{in_i}$からなる．
- それぞれのオペレーションO_{ij}はすべての機械によって処理ができるものとする．

- オペレーション O_{ij}を処理するためには，工具集合 $U_{ij} \subseteq \{T_1, \cdots, T_t\}$ に属する一つの工具が必要であり，その処理に p_{ij}時間かかるものとする．

- 任意のオペレーション O_{ij}はどのような順序でも処理することができる．

- 各オペレーションの処理は開始以降，任意の時間で中断することができ，それ以降いつでも，U_{ij}に属する工具と少なくとも一台の機械が使用可能である限りいずれの機械においても再開可能である．

- それぞれの仕事は一度にただ一つの機械でのみ処理することができ，それぞれの機械は一度に一つのオペレーションを処理することができる．

- それぞれの仕事 J_iには納期 d_iが定められている．

- あるスケジュールにおいて C_iを仕事 J_iの全てのオペレーションが完了した時刻とすると，納期遅れは $L_i = C_i - d_i$となり，最大完了時間は

  $$C_{\max} = \max_i C_i \tag{3.22}$$

  最大納期遅れは

  $$L_{\max} = \max_i L_i \tag{3.23}$$

  と表わすことができる．この最大完了時間と最大納期遅れをできる限り小さくするようなスケジュールを求めることがこの問題の目的である．

この問題を扱うために前節で述べた線形計画問題 $P_1^p(y)$ を拡張する．MPM モデルにおける機械を FMS モデルにおいては工具として扱うことによって工具における実行可能性は保証されるが，その場合各機械における実行可能性が保証されないことになる．そこで $P_1^p(y)$ に機械の実行可能性を保証するための新たな制約式を加えた新しい線形計画問題 $P_2^p(y)$ を定式化する．

前節と同様にパラメータ yを導入し，パラメータ yによって最大完了時間 $C_{\max}$に対し $C_{\max} \leq y$という制約を加える．そしてその制約の下で最大納期遅れ $L_{\max}$が最小となるような解を求める．

各仕事 J_iの納期 $d_i(i = 1, \cdots, n)$ を非減少順に並べ直し，必要ならば添え字番号を付け直し $d'_1 < d'_2 < \cdots < d'_q$とする．ただしここでは同じ納期は一つの納期にまとめるものと

し，qは異なる納期の数とする．pを$d'_l \leq y$が成立するようなlの最大値とし，l_iを$d'_l = d_i$となるlの値とする．x_{ij}^{kl}を仕事J_iの処理O_{ij}が機械M_kによって区間$[d'_{l-1}, d'_l)$において処理される時間の合計とする．次の線形計画問題$P_2^p(y)$を考える．

$P_2^p(y)$:

目的関数

$$z \to \text{minimize} \tag{3.24}$$

制約条件

$$\sum_{j,k} x_{ij}^{kl} \leq d'_l - d'_{l-1}, \quad l = 2, \cdots, p-1, i = 1, \cdots, n \tag{3.25}$$

$$\sum_{i,j} x_{ij}^{kl} \leq d'_l - d'_{l-1}, \quad l = 2, \cdots p-1, k = 1, \cdots m \tag{3.26}$$

$$\sum_{i,j} x_{ij}^{kl} \leq m \times \{d'_l - d'_{l-1}\}, \quad l = 2, \cdots p-1 \tag{3.27}$$

$$\sum_{j,k} x_{ij}^{k1} \leq d'_1 + z, \quad i = 1, \cdots, n \tag{3.28}$$

$$\sum_{i,j} x_{ij}^{k1} \leq d'_1 + z, \quad k = 1, \cdots, m \tag{3.29}$$

$$\sum_{i,j} x_{ij}^{k1} \leq m \times \{d'_1 + z\} \tag{3.30}$$

$$\sum_{j,k} x_{ij}^{kp+1} \leq y - d'_p, \quad i = 1, \cdots, n \tag{3.31}$$

$$\sum_{i,j} x_{ij}^{kp+1} \leq y - d'_p, \quad k = 1, \cdots, m \tag{3.32}$$

$$\sum_{i,j} x_{ij}^{kp+1} \leq m \times \{y - d'_p\} \tag{3.33}$$

$$z \geq y - d'_p \tag{3.34}$$

$$\sum_{k,l \leq l_i} x_{ij}^{kl} = p_{ij}, \quad j = 1, \cdots, n_i, i = 1, \cdots, m \tag{3.35}$$

$$x_{ij}^{kl} = 0, \quad \begin{array}{ll} l = l_i \cdots p+1, & i = 1, \cdots, n \\ j = 1, \cdots, n_i, & k = 1, \cdots, m \end{array} \tag{3.36}$$

$$x_{ij}^{kl} = 0, \quad \begin{array}{ll} l = 1 \cdots l_i, & i = 1, \cdots, n \\ j = 1, \cdots, n_i, & k = \{k \mid M_k \notin U_{ij}\} \end{array} \tag{3.37}$$

$$x_{ij}^{kl} \geq 0, \quad \begin{array}{ll} l = 2 \cdots l_i, & i = 1, \cdots, n \\ j = 1, \cdots, n_i, & k = 1, \cdots, m \end{array} \tag{3.38}$$

各工具についての列は前節の機械と同様であり，実行可能であることを保証するものである．また式 (3.27)，(3.30)，(3.33) によって，各機械での処理が実行可能であることを保証する．すなわち問題 $P_2^p(y)$ の最適解は工具における制約と機械における制約双方を満足する．この時 y を実行可能領域で動かすことによって前節の 2 目的問題の非劣解集合を求めることができる．そして前節と同様にして実行可能スケジュールを構築することが可能である．

3.6 資源の消費を考慮した多目的 MPM オープンショップスケジューリング問題

これまで述べてきたモデルは最大完了時間と最大納期遅れという主にスケジュールの時間に関する最適化である．しかし生産現場においてはオペレーションの処理を行うためにさまざまなエネルギーや資金といった資源を消費し，それらが制約となって現れる場合が多い．資源には離散量か連続量，再生可能か再生不可能かという性質があるが，本節においては連続でかつ再生不可能な資源の消費を考え，最大完了時間，最大納期遅れに加え，資源の消費を最小化するような問題を考える．本節で取り扱う 3 目的オープンショップスケジューリング問題は以下の様なものである．

- n 個の仕事 $J_1, \cdots, J_n$ がある．
- m 台の機械 $M_1, \cdots, M_m$ がある．
- 各仕事 $J_i (i = 1 \cdots n)$ はオペレーション $O_{i1}, \cdots, O_{in_i}$ からなる．
- オペレーション O_{ij} は機械の集合 $U_{ij} \subseteq \{M_1, \cdots, M_m\}$ の内の一台の機械によって p_{ij} 時間で処理される．
- 仕事 J_i のオペレーション O_{ij} を機械 M_k で処理する際に単位時間当たり資源 v_{ij}^k を消費するものとする．
- 任意のオペレーション O_{ij} はどのような順序でも処理することができる．
- 各オペレーションの処理は開始以降，任意の時間で中断することができ，それ以降いつでも，U_{ij} に属するいずれの機械においても再開可能である．

- それぞれの仕事は一度にただ一つの機械でのみ処理することができ，それぞれの機械は一度に一つのオペレーションを処理することができる．
- それぞれの仕事 J_iには納期 d_iが定められている．
- あるスケジュールにおいて C_iを仕事 J_iの全てのオペレーションが完了した時刻とする．そのとき納期遅れは $L_i = C_i - d_i$となり，最大完了時間 $C_{\max}$は

$$C_{\max} = \max_i C_i \tag{3.39}$$

最大納期遅れ $L_{\max}$は

$$L_{\max} = \max_i L_i \tag{3.40}$$

と表わすことができる．また x_{ij}^kを仕事 J_iのオペレーション O_{ij}が機械 M_kで処理される時間を表わすとすると資源の消費の合計 Wは

$$W = \sum_{i=1}^{n}\sum_{j=1}^{n_i}\sum_{k=1}^{m} v_{ij}^k x_{ij}^k \tag{3.41}$$

と表わすことができる．この最大完了時間と最大納期遅れ，資源の消費の合計をできる限り小さくするようなスケジュールを求めることがこの問題の目的である．

上記の問題に対し非劣解を求めるためにパラメータ w, yを導入する．パラメータ wを資源の消費の合計 Wに対する上限とする．すなわち $W \leq w$を制約として加える．またパラメータ yを最大完了時間の上限とする．すなわち $C_{\max} \leq y$を制約とする．その上で各機械への仕事の割り当て時間を解とし，$L_{\max}$を最小化する線形計画問題を導入する．各仕事 J_iの納期 $d_i (i = 1, \cdots, n)$ を非減少順に並べ直し，必要ならば添え字番号を付け直し $d'_1 < d'_2 < \cdots < d'_q$とする．ただしここでの qは異なる納期の数である．また p は $d'_l \leq y$となる最大の lの値とし，l_iは $d'_l = d_i$となる lの値とする．x_{ij}^{kl}を仕事 J_iのオペレーション O_{ij}が機械 M_kで区間 $[d'_{l-1}, d'_l)$ の間に処理される時間の合計とする．

以下の線形計画問題 $P_3^p(w, y)$ を考える．

$P_3^p(w, y)$:

目的関数

$$z \to \text{minimize} \tag{3.42}$$

制約条件

$$\sum_{j,k} x_{ij}^{kl} \le d'_l - d'_{l-1}, \quad \begin{array}{c} l = 2,\ldots,p-1 \\ i = 1,\ldots,n \end{array} \tag{3.43}$$

$$\sum_{i,j} x_{ij}^{kl} \le d'_l - d'_{l-1}, \quad \begin{array}{c} l = 2,\ldots,p-1 \\ k = 1,\ldots,m \end{array} \tag{3.44}$$

$$\sum_{j,k} x_{ij}^{k1} \le d'_1 + z, \quad i = 1,\ldots,n \tag{3.45}$$

$$\sum_{i,j} x_{ij}^{k1} \le d'_1 + z, \quad k = 1,\ldots,m \tag{3.46}$$

$$\sum_{j,k} x_{ij}^{kp+1} \le y - d'_p, \quad i = 1,\ldots,n \tag{3.47}$$

$$\sum_{i,j} x_{ij}^{kp+1} \le y - d'_p, \quad k = 1,\ldots,m \tag{3.48}$$

$$z \ge y - d'_p \tag{3.49}$$

$$\sum_{k,l \le l_i} x_{ij}^{kl} = p_{ij}, \quad j = 1,\ldots,n_i,\ i = 1,\ldots,n \tag{3.50}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n_i}\sum_{k=1}^{m}\sum_{l=1}^{p+1} v_{ij}^k x_{ij}^{kl} \le w \tag{3.51}$$

$$x_{ij}^{kl} = 0, \quad \begin{array}{ll} l = l_i \ldots p+1, & i = 1,\ldots,n \\ j = 1,\ldots,n_i, & k = \{k \mid M_k \notin U_{ij}\} \end{array} \tag{3.52}$$

$$x_{ij}^{kl} \ge 0, \quad \begin{array}{ll} l = 2 \ldots l_i, & i = 1,\ldots,n \\ j = 1,\ldots,n_i, & k = 1,\ldots,m \end{array} \tag{3.53}$$

問題 $P_3^p(w,y)$ を解くことによって資源の消費の合計 W を w 以下とし，かつ $C_{\max} \le y$ を満たす $L_{\max}$ の最小値を得ることができる．このとき $f_3^p(w,y)$ を $P_3^p(w,y)$ における z の最適値であるとすると，以下の定理 3.4，3.5 が成立する．

定理 3.4 $f_3^p(w,y)$ は区間 $[d'_l, y_l]$ $(l = r,\ldots,q)$ において w, y それぞれに対し凸関数であり，区分的に線形で，かつ w, y に対し非増加関数である．ただし y_l は $f_1^p(y) = y - d'_l$ が成立する最小の y である．

証明

定理 3.2と同様にして示すことができる．

証明終

定理 3.5 $f^p(w,y), y \in [d'_l, y_l](l = r, \ldots, q)$ の単調増加域と対応する yの変域において非劣スケジュールのスケジュールベクトルは $(w, y, f^p(w,y))$ の形で与えられる．ここで rは $d'_k \leq C^*_{\max}$を満たす最大の kの値であり，一般性を失わずに $C^*_{\max} > d'_1$を仮定する．

証明

定理3.3と同様にして示すことができる．

証明終

これまでと同様の方法でこの線形計画問題の解から実行可能スケジュールを得ることができる．

3.7　結言

本章ではFMSのモデル化として提案されたMPM機械環境を取り上げ，最大完了時間と最大納期遅れを2つの目的関数とするMPMオープンショップ問題を考察した．そしてこの問題に対し線形計画問題として定式化を行い，この線形計画問題の最適解から非劣スケジュールを構築する方法について示した．さらに実際にFMSに適用するため工具の使用に対する制約を加えたモデル化を行い，前述の結果を拡張することによって非劣スケジュールを求める方法を示した．また再生不可能な資源の消費を考慮し，最大完了時間，最大納期遅れ，資源の消費の3目的の問題についても考察を行った．本章で用いた線形計画問題としての定式化はさらにさまざまな現場の制約を付加することが容易であると考えられる．

第 4 章

2 目的等価並列機械スケジューリング問題

4.1 緒言

近年になって多目的スケジューリング問題についていくつかの研究が行われる様になった [43, 69, 71, 97, 98]．[69] において，2 機械オープンショップにおける最大完了時間と最大納期遅れという 2 目的の問題において単一の最適解が存在するかもしくは非劣解が線分上に存在することが示されている．また McCormick と Pinedo によって総滞留時間と最大完了時間という 2 目的の問題において非劣解の完全なトレードオフ曲線が求められている [71]．本章では 2 機械と 3 機械の等価並列機械において最大完了時間と最大納期遅れという以下の 2 つの目的関数を持つスケジューリング問題を考える．

m 台の等価並列機械において最大完了時間と最大納期遅れ，それぞれの単一目的関数問題についてはすでに研究されている [40, 72]．最大完了時間については McNaughton が以下の公式を示している [72]．ここで p_iは各仕事の処理時間であり，$C^*_{\max}$は最大完了時間の最適値である．

$$C^*_{\max} = \max\left[\max_i p_i,\ \frac{1}{m}\sum_{i=1}^{n} p_i\right] \tag{4.1}$$

最適解に対応する最適スケジュールの構築については Coffmann によって示されている [24]．

最大納期遅れについては Labertoulle がネットワークフロー問題に基づき実行可能スケジュールが存在するか否かを決定するアルゴリズムを示し, また最大納期遅れの最適値 $L^*_{\max}$ についての公式を示している [40]．Sahni は最大納期遅れについての実行可能スケジュールを求める異なるアルゴリズムを与えている [83]．

生産システムにおいて最大完了時間の最小化は生産現場から求められる要求であり，最大納期遅れの最小化は顧客より求められる要求である．これらの目的関数は異なるところから要求されるものであるために，しばしば対立する．そこで意思決定者はどちらか一方の目的関数を最小化するよりもこの二つの目的関数をバランスよく満たすようなスケジュールを立案しなければならない．次節では本章で扱う 2 目的等価並列機械スケジューリング問題について述べる．そして 4.3節においてこの 2 目的問題を解くために Sahni のアルゴリズムを変更し、実行可能なスケジュールが存在するための条件を導出する．4.4節において，2 機械の場合の結果を与え，4.5節においてその結果を 3 機械の場合に拡張する．最後の節においていくつかの結論と今後の課題について述べる．

4.2　2 目的等価並列機械スケジューリング問題

本章では以下の問題を考える．

- n 個の仕事 $J_1, \cdots, J_n$がある．
- 2(または 3) 台の並列機械 $M_1, M_2, (M_3)$ がある．
- それぞれ仕事 J_iには処理時間 p_iと納期 d_iが定められている．ただしここで $p_i < d_i$を仮定する．
- いずれの仕事も中断することが可能であり，直ちにまたは後で同じ機械かもしくは異なる機械で処理を再開できるものとする．
- それぞれ機械は一度に高々一つの仕事しか処理できず，またそれぞれ仕事は一度に高々一つの機械でしか処理されないものとする．
- あるスケジュールにおいて，仕事 J_iに対し，C_iをその仕事が完了した時刻、L_iを納期遅れすなわち $L_i = \max[C_i - d_i, 0]$ を定義すると最大完了時間 $C_{\max}$は

$$C_{\max} = \max_{1 \leq i \leq n} C_i \tag{4.2}$$

　最大納期遅れ $L_{\max}$は

$$L_{\max} = \max_{1 \leq i \leq n} L_i \tag{4.3}$$

　とそれぞれ定義される．

この問題においては最大完了時間 $C_{\max}$ と最大納期遅れを $L_{\max}$ をそれぞれ最小化することが目的である．しかし一般には同時に 2 つの目的関数を最適化するような解は存在しないので、非劣解集合を求めることを目的とする．目的関数値 $(C_{\max}, L_{\max})$ の値の組をもつ実行可能スケジュールが存在するならば $(C_{\max}, L_{\max})$ を実行可能点と呼ぶ．この問題の目的はこの 2 つの目的関数に対し単一の最適解が存在するか，もしくは非劣実行可能点を示す領域を求めることである．

4.3 Sahni のアルゴリズム

本節では Sahni によるアルゴリズムを紹介し，実行可能なスケジュールが存在するための条件を導くために Sahni のアルゴリズムに変更を行う．まず以下の定義を行う．

- $\mathcal{J}$ を n 個の仕事の集合とする．すなわち $\mathcal{J} = \{J_i \mid i = 1, \cdots, n\}$ である．
- $d'_i(0 \leq i \leq k)$ を k 個の異なる納期とする．ただしここで $d_0 = 0$ とする．
- $n_i(1 \leq i \leq k)$ を納期 d'_i を持つ仕事の数であるとする．

明らかに $\sum n_i = n$ であり，$n_i \geq 1(1 \leq i \leq k)$ を仮定する．

Sahni によって提案されたアルゴリズム ONERT では以下の処理を k 回繰り返すことによって n 個の仕事を割り当てる．i 回目の繰り返しにおいて d'_i の納期を持つ仕事が割り当てられる．納期 d'_i を持つ仕事 J_r の各機械への割り当てを以下の様に行う．ただし各仕事は納期の非減少順に添え字がつけられているものとする．すなわち $d_1 < d_2 < \cdots < d_n$ とする．

1. $M_j(i)$ を機械 M_j での仕事 J_i の処理の終了時間とし，$RPT(j)$ を機械 M_j において 0 から $d'_i(1 \leq i \leq n)$ までに仕事の処理が可能な時間であるとする．すなわち $RPT(j) = d'_i - M_j(i-1)$ とする．

2. もし全ての機械について $p_r > RPT(j)$ ならばどの機械で J_r を処理しても納期までに間に合わない．よってアルゴリズムが終了する．

3. もし $RPT(j) \neq 0$ でない全ての機械について $p_r \leq RPT(j)$ ならば RPT が最も小さい機械で J_r を処理する．

4. もし2でも3でもなければ，$p_r \geq RPT(j)$ となる機械の中で最も大きい RPT を持つ機械と $RPT(k) \neq 0$ でない $p_r < RPT(j)$ となる機械の中で最も小さい RPT を持つ機械で処理を行う．

上記の基本ルールに基づいて Sahni はアルゴリズム ONERT を提案している．まず各仕事は納期の非減少順に添え字がつけられているものとする．すなわち $d_1 < d_2 < \cdots < d_n$ とする．$M_j(i)$ を機械 M_j での仕事 J_i の処理の終了時間とする．

アルゴリズム **ONERT**

for $i \leftarrow 1$ to n do

begin

　for $j \leftarrow 1$ to m do

　　$RPT(j) = d_i - M_j(i-1)$

　if 全ての j について $RPT(j) > p_i$ then

　begin

　　$k \leftarrow \{j \mid \min_j\{RPT(j)\}\}$

　　// 仕事 J_i は機械 M_k によって処理される．

　　$M_k(i) \leftarrow M_k(i-1) + p_i$

　　$M_j(i) \leftarrow M_j(i-1), \quad j = 1, \cdots, k-1, k+1, \cdots, m$

　end else

　if $RPT(j) > p_i$ となる j が存在する．

　begin

　　$k \leftarrow \{j \mid RPT(j) > p_i$ かつ $\min_j\{RPT(j)\}\}$

　　$l \leftarrow \{j \mid RPT(j) < p_i$ かつ $\max_j\{RPT(j)\}\}$

　　// 仕事 J_i は機械 M_k, M_l によって処理される．

　　$M_k(i) \leftarrow M_k(i-1) + d_i$

　　$M_l(i) \leftarrow M_l(i-1) + p_i - (d_i - M_k(i-1))$

　　$M_j(i) \leftarrow M_j(i-1), \quad j = 1, \cdots, k-1, k+1, \cdots, l-1, l+1, \cdots, m$

　end else

　if $RPT(j) > p_i$ となる j が存在しない．

　begin

　　　実行可能なスケジュールは存在しない．

　　end

end

ONERT によって実行可能なスケジュールが存在するとき，そのスケジュールを構築することができるが，実行不可能であるときには停止してしまい，それ以上の情報を得ることができない．よって以降の解析を行うために ONERT に基づいて，新たなアルゴリズム ONERT2 を提案する．ONERT2 では $M_j(i)$ を機械 M_jでの仕事 J_iの処理の終了時間としたときに，常に $M_j(i) < M_{j+1}(i)$ が成立する．まず各仕事は納期の非減少順に添え字がつけられているものとする．すなわち $d_1 < d_2 < \cdots < d_n$とする．$M_j(i)$ を機械 M_jでの仕事 J_iの処理の終了時間とする．

アルゴリズム **ONERT2**

for $i \leftarrow 1$ to n do

begin

　　$r_i \leftarrow p_i$

　　for $j \leftarrow 1$ to m do

　　begin

　　　　$RPT(j) = d_i - M_j(i-1)$

　　end

　　if $RPT(1) > r_i$ then

　　begin

　　　　$M_1(i) \leftarrow M_1(i-1) + r_i$ // 仕事 J_iは機械 M_1によって全て処理される．

　　　　$M_j(i) \leftarrow M_j(i-1), \quad j = 2, \cdots, m$

　　　　$r_i \leftarrow 0$

　　end else

　　begin

　　　　$M_1(i) \leftarrow d_i$

　　　　// 機械 M_1は仕事 J_iを $[M_1(i-1), d_i]$ 間処理される．

　　　　$r_i \leftarrow r_i - (d_i - M_1(i-1))$

　　　　while $r_i > 0$ do

```
                begin
                    if RPT(j) < r_i then
                    begin
                        if j < m then
                        begin
                            M_j(i) ← M_{j-1}(i-1)
                            // 機械 M_1は仕事 J_iを [M_j(i-1), M_{j-1}(i-1)] 間処理される
                            r_i ← r_i - (M_{j-1}(i-1) - M_j(i-1))
                        end else
                        begin
                            実行可能なスケジュールは存在しない.
                        end
                    end else
                    begin
                        M_j(i) ← M_j(i-1) + r_i
                        // 機械 M_1は仕事 J_iを [M_j(i-1), M_j(i) + r_i] 間処理される.
                        r_i ← 0
                        M_k(i) ← M_k(i-1),    k = j+1, ..., m
                    end
                end
            end
        end
```

ここでONERTとONERT2が各仕事の処理の完了時間に関して等価であることを示す.

補題 4.1 スケジュールπをONERT2によって構築されたスケジュールとし，$M_k(i)$をスケジュールπにおけるJ_iのM_kにおける処理の完了時間とする．π'をアルゴリズムONERTによって構築されたスケジュールとし，$M'_k(i)$をスケジュールπ'においてそれぞれの機械でJ_iを処理を完了した時間を非減少順にならべたものとする．そのとき以下の式が成立する.

$$M_j(i) = M'_j(i), \quad j = 1, \cdots, m \tag{4.4}$$

証明

もし $i=1$ ならば明らかに命題は成立する.

$M_j(i-1) = M'_j(i-1)$ が成立すると仮定し，$RPT(j) = d_i - M'_j(i-1)$ とする.

1. $p_i \leq d_i - M_1(i-1)$ ならば，

   ONERT2 において J_1 は M_1 で処理される．よって

$$
\begin{aligned}
M_1(i) &= M_1(i-1) + p_i && (4.5)\\
M_j(i) &= M_j(i-1), \quad j = 2, \cdots, m && (4.6)
\end{aligned}
$$

   が成立する．一方 Sahni のアルゴリズムでは J_i は $RPT(j)$ の最も少ない機械 M_j によって処理されるので

$$
\begin{aligned}
M'_1(i) &= M'_1(i-1) + p_i && (4.7)\\
M'_j(i) &= M'_j(i-1), \quad j = 2, \cdots, m && (4.8)
\end{aligned}
$$

   が成立する.

2. $p_i > d_i - M_1(i-1)$ ならば，

   ONERT2 において，ある機械 M_k が存在し，以下の式が成立する.

$$
\begin{aligned}
M_1(i) &= d_i && (4.9)\\
M_j(i-1) &= M_{j-1}(i-1), \quad j = 2, \cdots, k && (4.10)\\
M_k(i) &= M_k(i-1) + p_i - (d_i - M_{k-1}(i-1)) && (4.11)\\
M_j(i) &= M_j(i-1), \quad j = k, \cdots, m && (4.12)
\end{aligned}
$$

   Sahni のアルゴリズムでは $M_{k'}$ を $p_i \geq RPT(j)$ が成立するような全ての機械のうちもっとも $RPT(j)$ が大きい機械とすると $M_{k'+1}$ が $p_i < RPT(j)$ が成立するような全ての機械のうちもっとも $RPT(j)$ が小さい機械となり，J_i は $M_{k'}$ において区間 $(M'_{k'}(i-1), d_i]$ で処理され，$M_{k'+1}$ において区間 $(M'_{k'+1}(i-1), M'_{k'+1}(i-1) + p_i - (d_i - M'_{k'}(i-1))]$ で処理される．すなわち以下の式が成立する.

$$M'_{k'}(i) = d_i \tag{4.13}$$

$$M'_{k'+1}(i-1) = M'_{k'+1}(i-1) + p_i - (d_i - M'_{k'}(i-1)) \tag{4.14}$$

$$M'_j(i) = M'_j(i-1), \quad j = 1, \cdots, k', k'+1 \cdots, m \tag{4.15}$$

式(4.9)-(4.12)と(4.13)-(4.15)より，以下の式が成立.

$$M_1(i) = M'_{k'}(i) \tag{4.16}$$

$$M_j(i) = M'_j(i), \quad j = 2, \cdots, k \tag{4.17}$$

$$M_k(i) = M'_{k'+1}(i) \tag{4.18}$$

$$M_j(i) = M'_j(i), \quad j = k+2, \cdots, m \tag{4.19}$$

よって補題が示された.

証明終

ここで実行可能なスケジュールが存在するための条件を示す.

定理 4.2 ONERT2において，以下の不等式が $i = 1, \cdots, n$ において成立する時またその時のみ実行可能スケジュールが存在する.

$$M_m(i) \leq M_{m-1}(i-1) \tag{4.20}$$

証明

もしある仕事 J_iにおいて $M_m(i) > M_{m-1}(i-1)$ が成立するとき，J_iは区間 $[M_{m-1}(i-1), M_m(i)]$ において機械 M_{m-1}, M_mの両方で処理されることになる. これは制約に反する. よって必要条件は明らかである. 十分条件について補題4.1より，ONERT2はSahniのアルゴリズムと全ての仕事の処理の開始時間と終了時間において等価である. したがって十分条件はSahniのアルゴリズムの妥当性より明らかである.

証明終

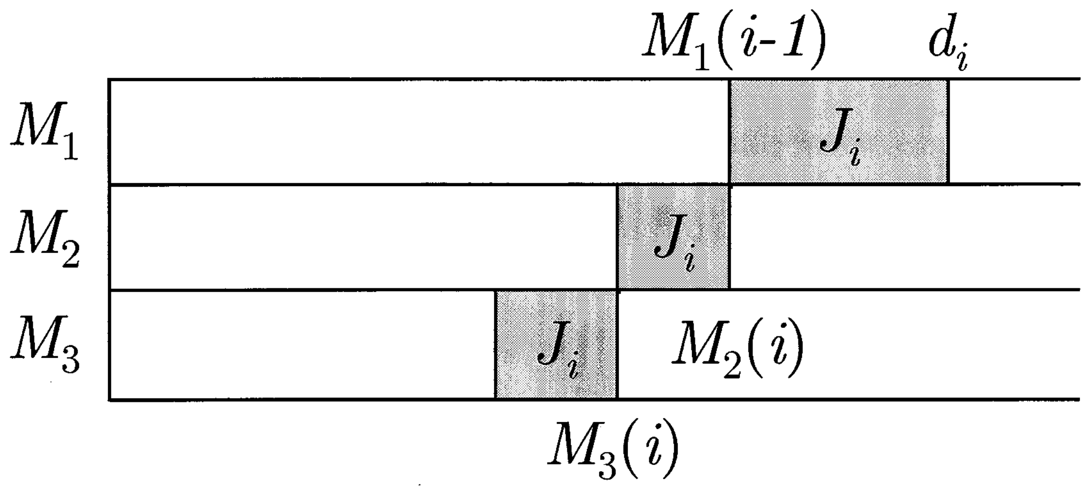

図 4.1　実行可能な場合

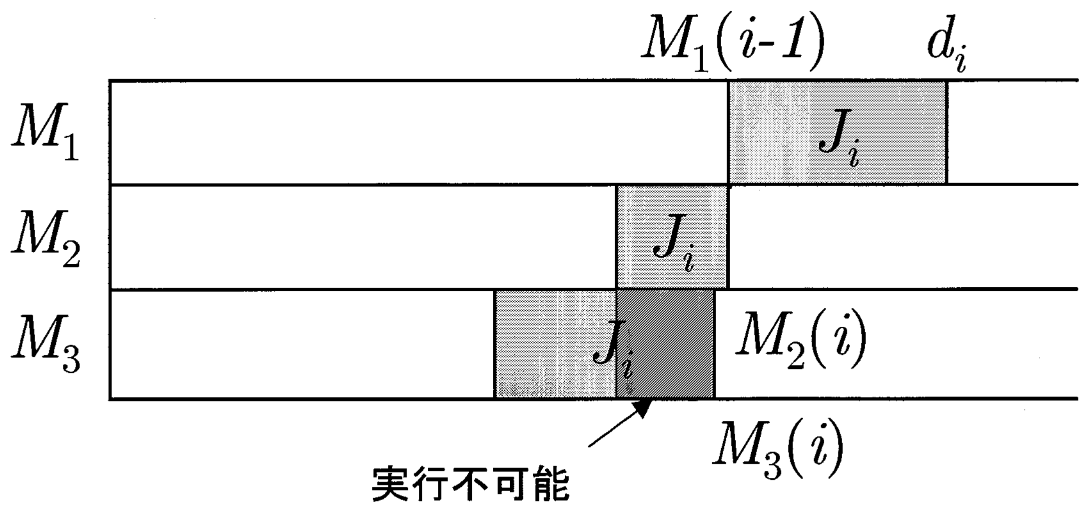

図 4.2　実行不可能な場合

4.4 2機械問題

4.4.1 最大納期遅れ最小化問題

本節では最大納期遅れ $L^*_{\max}$を最小化問題を考える．Sahni によるアルゴリズムは納期遅れがない場合に任意の m 機械上で実行可能なスケジュールを与える．本節では機械が 2 台の場合における Sahni のアルゴリズムを納期に遅れる場合を加えて解析し，最大納期遅れ $L^*_{\max}$についての新しい公式を導出する．

まずそれぞれ仕事は納期の順に添え字が付けられているものとする．すなわち $d_1 \le d_2 \le \cdots \le d_n$とする．$M_1(i)$ と $M_2(i)$ をそれぞれ M_1と M_2における仕事 J_iの処理の完了時間とする．

前節で示したように実行可能スケジュールが存在する必要十分条件は以下の不等式が成立することである．

$$M_2(i) \le M_1(i-1), \quad i = 1, \cdots, n \tag{4.21}$$

ONERT2 を解析することによって以下の漸化式を得る．

$$\begin{aligned} M_1(i) &= \min\left[M_1(i-1) + p_i, d_i\right] && (4.22) \\ M_2(i) &= M_2(i-1) + (p_i - (M_1(i) - M_1(i-1))) && (4.23) \end{aligned}$$

ただし $M_1(0) = 0, M_2(0) = 0$ とする．漸化式を解くことにより，式 (4.22) と式 (4.23) は以下の様に変形できる．

$$\begin{aligned} M_1(i) &= \min_{1 \le k \le i}\left[\sum_{j=1}^{i} p_j, d_k + \sum_{j=k+1}^{i} p_j\right] && (4.24) \\ M_2(i) &= \sum_{j=1}^{i} p_j - M_1(i) && (4.25) \end{aligned}$$

ただし$\sum_{j=i+1}^{i} p_j = 0$ であるとする．

Sahni のアルゴリズムは全ての仕事が納期に間に合うような実行可能スケジュールが存在するときにそのスケジュールを構築することができるが，もし実行可能なスケジュールが存在しないとき，実行不可能な仕事が存在した時点で停止してしまい，その仕事とそれ以降の仕事がどのような納期を与えられれば実行可能になるのかという情報を得ることができない．そこで最大納期遅れの最適値を求めるために最大納期遅れを表わすパラメータ L を導入する．それぞれの納期 d_iを新たに $d'_i = d_i + L$ に置き換え，その最小値を求める

ことにより実行可能スケジュールが存在するような最小の L の値すなわち最大納期遅れの最適値を求めることができる．d_iを d'_iに置き換えることによって式 (4.24) は以下の様に変更される．

$$M_1(i) = \min_{1\le k\le i}\left[\sum_{j=1}^{i} p_j,\ d_k + L + \sum_{j=k+1}^{i} p_j\right] \tag{4.26}$$

式 (4.25) と (4.26) を代入することにより実行可能である条件を示す不等式 (4.21) を以下の様に変形できる．

$$\sum_{j=1}^{i} p_j \le \min_{1\le k\le i}\left[\sum_{j=1}^{i} p_j, d_k + L + \sum_{j=k+1}^{i} p_j\right] + \min_{1\le k\le i-1}\left[\sum_{j=1}^{i-1} p_j, d_k + L + \sum_{j=k+1}^{i-1} p_j\right] \tag{4.27}$$

不等式 (4.27) を各 min 関数を展開することによって以下の 4 つの不等式を得る．

$$\sum_{j=1}^{i} p_j \quad\le\quad \sum_{j=1}^{i} p_j + \sum_{j=1}^{i-1} p_j \tag{4.28}$$

$$\sum_{j=1}^{i} p_j \quad\le\quad \sum_{j=1}^{i} p_j + \min_{1\le k\le i-1}\left[d_k + L + \sum_{j=k+1}^{i-1} p_j\right] \tag{4.29}$$

$$\sum_{j=1}^{i} p_j \quad\le\quad \min_{1\le k\le i}\left[d_k + L + \sum_{j=k+1}^{i} p_j\right] + \sum_{j=1}^{i-1} p_j \tag{4.30}$$

$$\sum_{j=1}^{i} p_j \quad\le\quad \min_{1\le k\le i}\left[d_k + L + \sum_{j=k+1}^{i} p_j\right] + \min_{1\le k\le i-1}\left[d_k + L + \sum_{j=k+1}^{i-1} p_j\right] \tag{4.31}$$

すなわち実行可能である条件を示す不等式 (4.27) は不等式 (4.28)-(4.31) が全て成立する時またその時のみ成立する．

ここで不等式 (4.28) と (4.29) は

$$\sum_{j=1}^{i-1} p_j \quad\ge\quad 0 \tag{4.32}$$

$$\min_{1\le k\le i-1}\left[d_k + L + \sum_{j=1}^{i-1} p_j\right] \quad\ge\quad 0 \tag{4.33}$$

より常に成立することは明らかである．

また不等式 (4.30) を変形することによって以下の式を得る．

$$p_i \le \min\left[\min_{1\le k\le i-1}\left[d_k + \sum_{j=k+1}^{i-1} p_j\right] + p_i, d_i\right] + L \tag{4.34}$$

$p_i \leq d_i$と $L \geq 0$ より，不等式 (4.34) もまた常に成立する．したがって実行可能である条件を示す不等式 (4.27) は不等式 (4.31) と等価である．不等式 (4.31) を変形することによって，以下の不等式を得る．

$$2L \geq \left[\sum_{j=1}^{i} p_j - \min_{1\leq k\leq i}\left[d_k + \sum_{j=k+1}^{i} p_j\right] - \min_{1\leq k\leq i-1}\left[d_k + \sum_{j=k+1}^{i-1} p_j\right]\right] \tag{4.35}$$

実行可能スケジュールが存在するためには，不等式 (4.35) が全ての仕事，すなわち $i = 1, \cdots, n$ において成立する必要がある．よって次式が成立するとき、またそのときに限り実行可能なスケジュールが存在する．

$$L \geq \frac{1}{2}\max_{1\leq i\leq n}\left[\sum_{j=1}^{i} p_j - \min_{1\leq k\leq i}\left[d_k + \sum_{j=k+1}^{i} p_j\right] - \min_{1\leq k\leq i-1}\left[d_k + \sum_{j=k+1}^{i-1} p_j\right]\right] \tag{4.36}$$

以上の議論より以下の定理が成立する．

定理 4.3 納期遅れの最適値 $L^*_{\max}$について以下の式が成立する．

$$L^*_{\max} = \frac{1}{2}\max_{1\leq i\leq n}\left[\sum_{j=1}^{i} p_j - \min_{1\leq k\leq i}\left[d_k + \sum_{j=k+1}^{i} p_j\right] - \min_{1\leq k\leq i-1}\left[d_k + \sum_{j=k+1}^{i-1} p_j\right]\right] \tag{4.37}$$

4.4.2　2 目的問題

この節では，最大完了時間 $C_{\max}$と最大納期遅れ $L_{\max}$という 2 目的の等価並列 2 機械スケジューリング問題を考える．もしある仕事について定められた最大完了時間以降にその仕事の納期があるならば，その仕事は納期ではなく最大完了時間までに処理しなければならない．ゆえに最大完了時間はすべての仕事についての共通の納期として考えることができる．したがって仕事 J_iについての納期を新たに d_i^+を $d_i^+ = \min\left[d_i + L, C_{\max}\right]$ と定義する．ただしここで $C_{\max}$は最大完了時間を示すものとし、式 (4.1) を満たすものとする．

4.4.1節と同様にして，実行可能である条件を示す不等式は以下のように与えられる．

$$\begin{aligned}\sum_{j=1}^{i} p_j \quad &\leq \quad \min_{1\leq k\leq i}\left[\sum_{j=1}^{i} p_j, \min\left[d_k + L, C_{\max}\right] + \sum_{j=k+1}^{i} p_j\right] \\ &+ \quad \min_{1\leq k\leq i-1}\left[\sum_{j=1}^{i-1} p_j, \min\left[d_k + L, C_{\max}\right] + \sum_{j=k+1}^{i-1} p_j\right]\end{aligned} \tag{4.38}$$

不等式 (4.38) を前節と同様に展開することによって以下の不等式を得る．

$$\sum_{j=1}^{i} p_j \quad \leq \quad \sum_{j=1}^{i} p_j + \sum_{j=1}^{i-1} p_j \tag{4.39}$$

$$\sum_{j=1}^{i} p_j \leq \sum_{j=1}^{i} p_j + \min_{1 \leq k \leq i-1} \left[\min\left[d_k + L, C_{\max}\right] + \sum_{j=k+1}^{i-1} p_j \right] \tag{4.40}$$

$$\sum_{j=1}^{i} p_j \leq \min_{1 \leq k \leq i} \left[\min\left[d_k + L, C_{\max}\right] + \sum_{j=k+1}^{i} p_j \right] + \sum_{j=1}^{i-1} p_j \tag{4.41}$$

$$\begin{aligned}\sum_{j=1}^{i} p_j \leq & \min_{1 \leq k \leq i} \left[\min\left[d_k + L, C_{\max}\right] + \sum_{j=k+1}^{i} p_j \right] \\ & + \min_{1 \leq k \leq i-1} \left[\min\left[d_k + L, C_{\max}\right] + \sum_{j=k+1}^{i-1} p_j \right] \end{aligned} \tag{4.42}$$

これらの全ての不等式 (4.39)-(4.42) が成立するとき，またその時のみ実行可能スケジュールが存在する．不等式 (4.28) と (4.29) と同様に，不等式 (4.39) と (4.40) 常に成立することは明らかである．さらに不等式 (4.41) は以下のように 2 つの不等式に展開できる．

$$p_i \leq \min_{1 \leq k \leq i} \left[d_k + L + \sum_{j=k+1}^{i} p_j \right] \tag{4.43}$$

$$p_i \leq C_{\max} + \min_{1 \leq k \leq i} \left[\sum_{j=k+1}^{i} p_j \right] \tag{4.44}$$

ここで $C_{\max} \geq \max_{1 \leq i \leq n} \left[p_i, \frac{1}{2} \sum_{j=1}^{n} p_i \right]$ より不等式 (4.43)，(4.44) は常に成立する．したがって不等式 (4.41) は常に成立する．残りの不等式 (4.42) を展開することによって以下の 4 つの不等式を得る．

$$\sum_{j=1}^{i} p_j \leq C_{\max} + \min_{1 \leq k \leq i} \left[\sum_{j=k+1}^{i} p_j \right] + C_{\max} + \min_{1 \leq k \leq i-1} \left[\sum_{j=k+1}^{i-1} p_j \right] \tag{4.45}$$

$$\sum_{j=1}^{i} p_j \leq C_{\max} + \min_{1 \leq k \leq i} \left[\sum_{j=k+1}^{i} p_j \right] + \min_{1 \leq k \leq i-1} \left[d_k + L + \sum_{j=k+1}^{i-1} p_j \right] \tag{4.46}$$

$$\sum_{j=1}^{i} p_j \leq \min_{1 \leq k \leq i} \left[d_k + L + \sum_{j=k+1}^{i} p_j \right] + C_{\max} + \min_{1 \leq k \leq i-1} \left[\sum_{j=k+1}^{i-1} p_j \right] \tag{4.47}$$

$$\sum_{j=1}^{i} p_j \leq \min_{1 \leq k \leq i} \left[d_k + L + \sum_{j=k+1}^{i} p_j \right] + \min_{1 \leq k \leq i-1} \left[d_k + L + \sum_{j=k+1}^{i-1} p_j \right] \tag{4.48}$$

最大完了時間の最小値の公式 (4.1) より，$\sum_{j=1}^{n} p_j \leq 2C_{\max}$が成立する．よって不等式 (4.45) は常に成立する．また $\min_{1 \leq k \leq i} \left[\sum_{j=k+1}^{i} p_j \right] = 0$ より不等式 (4.46) と (4.47) を変形する

ことによって以下の不等式を得る.

$$L \geq \sum_{j=1}^{i} p_j - C_{\max} - \min_{1 \leq k \leq i-1} \left[d_k + \sum_{j=k+1}^{i-1} p_j \right] \tag{4.49}$$

$$L \geq \sum_{j=1}^{i} p_j - C_{\max} - \min_{1 \leq k \leq i} \left[d_k + \sum_{j=k+1}^{i} p_j \right] \tag{4.50}$$

さらに以下の不等式が成立する.

$$\begin{aligned} \min_{1 \leq k \leq i} \left[d_k + \sum_{j=k+1}^{i} p_j \right] &= \min \left[d_i, \min_{1 \leq k \leq i-1} \left[d_k + \sum_{j=k+1}^{i-1} p_j \right] \right] + p_i \\ &\geq \min_{1 \leq k \leq i-1} \left[d_k + \sum_{j=k+1}^{i-1} p_j \right] \end{aligned} \tag{4.51}$$

よって不等式 (4.50) が成立するとき不等式 (4.49) は常に成立する.

また不等式 (4.48) は $L_{\max}$の単一目的問題の最適値を示す式と同一である.

以上の議論より次の定理を得る.

定理 4.4 最大納期遅れ L について以下の式が成立する.

$$L = \max \left[L^*_{\max}, \max_{1 \leq i \leq n} \left[\sum_{j=1}^{i} p_j - C_{\max} - \min_{1 \leq k \leq i-1} \left[d_k + \sum_{j=k+1}^{i-1} p_j \right] \right] \right] \tag{4.52}$$

ここで $C_{\max}$は最大完了時間を示すパラメータである.

ここで

$$F = \max_{1 \leq i \leq n} \left[\sum_{j=1}^{i} p_j - \min_{1 \leq k \leq i-1} \left[d_k + \sum_{j=1}^{i-1} p_j \right] \right] \tag{4.53}$$

とおくと, もし $L^*_{\max} \geq F - C^*_{\max}$ ならば $(C^*_{\max}, L^*_{\max})$ という単一の最適解が存在する. この場合両方の目的関数は同時に最適化される. そうでなければ $(C^*_{\max}, F - C^*_{\max})$ と $(C^+_{\max}, L^*_{\max})$ を結ぶ線分上に非劣解は存在する (図 4.3). ただし $C^+_{\max}$は $L = L^*_{\max}$と固定したときの最大完了時間の最小値とする. 実際にスケジュールを構築する際にはまず $C_{\max}$または $L_{\max}$のいずれかの値がを決定し, もう一方の目的関数の値を式 (4.52) によって求める. それに対応した実際の実行可能スケジュールはこの式によって得られた最大納期遅れと最大完了時間を用いて ONERT によって求めれば良い.

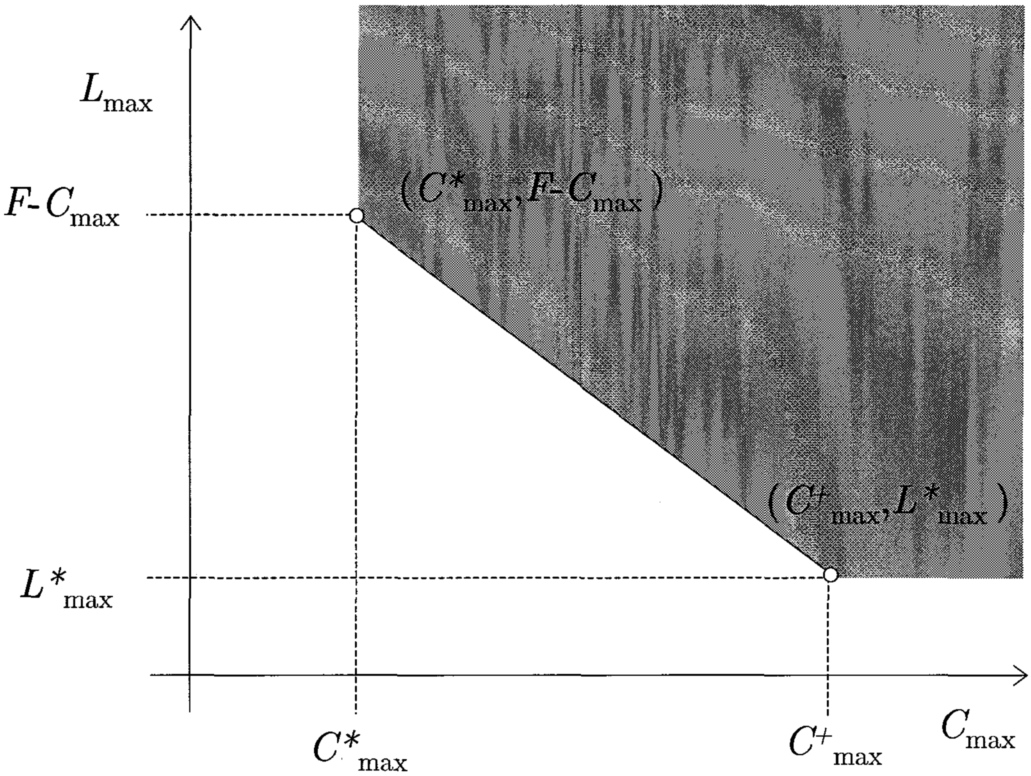

図 4.3 非劣解

4.5　3 機械問題

4.5.1　最大納期遅れ最小化問題

2 機械の場合と同様に Sahni のアルゴリズムの解析から始める．それぞれの仕事はその納期順にインデックスが付けられているものとする．すなわち $d_1 \leq d_2 \leq \cdots \leq d_n$とする．$M_1(i), M_2(i), M_3(i)$ はそれぞれ機械 M_1, M_2, M_3において仕事 J_iの処理の完了時間を表わすとする．

式 (4.20) より、実行可能スケジュールが存在することと以下の式が成立することは等価である．

$$M_3(i) \leq M_2(i-1), \quad i = 1, \cdots, n \tag{4.54}$$

ONERT2 を解析し，納期 d_iを $d_i^+ = d_i + L$ に置き換えることによって以下の式を得る.

$$
\begin{aligned}
M_1(i) &= \min_{1\leq k\leq i}\left[\sum_{j=1}^{i} p_j, d_k + L + \sum_{j=k+1}^{i} p_j\right] && (4.55)\\
M_2(i) &= \min\left[M_1(i-1), M_2(i-1) + (p_i - (M_1(i) - M_1(i-1)))\right] \\
&= \min_{1\leq k\leq i}\left[M_1(k-1) + \left(\sum_{j=k+1}^{i} p_j - (M_1(i) - M_1(k))\right), \sum_{j=1}^{i} p_j - M_1(i)\right] && (4.56)\\
M_3(i) &= \sum_{j=1}^{i} p_j - (M_1(i) + M_2(i)) && (4.57)
\end{aligned}
$$

ただし $M_1(0) = 0, M_2(0) = 0, M_3(0) = 0$ とする.

(4.55)，(4.56)，(4.57) より実行可能である条件を示す不等式 (4.54) は以下のように変形される.

$$
\begin{aligned}
\sum_{j=1}^{i} p_j &\leq M_1(i) + M_2(i) + M_2(i-1) \\
&= \min_{1\leq l\leq i}\left[M_1(l) + M_1(l-1) + \sum_{j=l+1}^{i} p_j, \sum_{j=1}^{i} p_j\right] \\
&\quad + \min_{1\leq l\leq i-1}\left[M_1(l) + M_1(l-1) + \sum_{j=l+1}^{i-1} p_j, \sum_{j=1}^{i-1} p_j\right] - M_1(i-1) && (4.58)
\end{aligned}
$$

ただし$\sum_{j=i+1}^{i} p_j = 0$ とする.

不等式 (4.58) は以下の 4 つの不等式 (4.59)-(4.62) と等価である.

$$
\begin{aligned}
\sum_{j=1}^{i} p_j &\leq \sum_{j=1}^{i} p_j + \sum_{j=1}^{i-1} p_j - \min_{1\leq k\leq i-1}\left[d_k + L + \sum_{j=k+1}^{i-1} p_j\right] && (4.59)\\
\sum_{j=1}^{i} p_j &\leq \sum_{j=1}^{i} p_j + \min_{1\leq l\leq i-1}\left[\min_{1\leq k\leq l}\left[d_k + L + \sum_{j=k+1}^{l} p_j\right] + \min_{1\leq k\leq l-1}\left[d_k + L + \sum_{j=k+1}^{l-1} p_j\right]\right. \\
&\quad \left. + \sum_{j=l+1}^{i-1} p_j\right] - \min_{1\leq k\leq i-1}\left[d_k + L + \sum_{j=k+1}^{i-1} p_j\right] && (4.60)\\
\sum_{j=1}^{i} p_j &\leq \min_{1\leq l\leq i}\left[\min_{1\leq k\leq l}\left[d_k + L + \sum_{j=k+1}^{l} p_j\right] + \min_{1\leq k\leq l-1}\left[d_k + L + \sum_{j=k+1}^{l-1} p_j\right]\right. \\
&\quad \left. + \sum_{j=l+1}^{i} p_j\right] + \sum_{j=1}^{i-1} p_j - \min_{1\leq k\leq i-1}\left[d_k + L + \sum_{j=k+1}^{i-1} p_j\right] && (4.61)\\
\sum_{j=1}^{i} p_j &\leq \min_{1\leq l\leq i}\left[\min_{1\leq k\leq l}\left[d_k + L + \sum_{j=k+1}^{l} p_j\right] + \min_{1\leq k\leq l-1}\left[d_k + L + \sum_{j=k+1}^{l-1} p_j\right] + \sum_{j=l+1}^{i} p_j\right]
\end{aligned}
$$

$$+\min_{1\le l\le i-1}\left[\min_{1\le k\le l}\left[d_k+L+\sum_{j=k+1}^{l}p_j\right]+\min_{1\le k\le l-1}\left[d_k+L+\sum_{j=k+1}^{l-1}p_j\right]+\sum_{j=l+1}^{i-1}p_j\right]$$
$$-\min_{1\le k\le i-1}\left[d_k+L+\sum_{j=k+1}^{i-1}p_j\right] \tag{4.62}$$

ここで

$$\min_{1\le k\le i-1}\left[d_k+L+\sum_{j=k+1}^{i-1}p_j\right]\le\sum_{j=1}^{i-1}p_j \tag{4.63}$$

が成立するので不等式 (4.59) は常に成立．また以下の不等式 (4.64) が成立するので不等式 (4.60) は常に成立する．

$$\min_{1\le k\le i-1}\left[d_k+L+\sum_{j=k+1}^{i-1}p_j\right]\le\min_{1\le l\le i-1}\left[\min_{1\le k\le l}\left[d_k+L+\sum_{j=k+1}^{l}p_j\right]\right.$$
$$\left.+\min_{1\le k\le l-1}\left[d_k+L+\sum_{j=k+1}^{l-1}p_j\right]+\sum_{j=l+1}^{i-1}p_j\right] \tag{4.64}$$

不等式 (4.61) を変形することによって以下の不等式を得る．

$$\min_{1\le k\le i-1}\left[d_k+L+\sum_{j=k+1}^{i-1}p_j\right]+p_i\le\min\left[\min_{1\le k\le i}\left[d_k+L+\sum_{j=k+1}^{i}p_j\right]\right.$$
$$+\min_{1\le k\le i-1}\left[d_k+L+\sum_{j=k+1}^{i-1}p_j\right],\min_{1\le l\le i-1}\left[\min_{1\le k\le l}\left[d_k+L+\sum_{j=k+1}^{l}p_j\right]\right.$$
$$\left.\left.+\min_{1\le k\le l-1}\left[d_k+L+\sum_{j=k+1}^{l-1}p_j\right]+\sum_{j=l+1}^{i-1}p_j\right]+p_i\right] \tag{4.65}$$

ここで以下の不等式 (4.66) と (4.67) が成立することより，不等式 (4.61) は常に成立する．

$$\min_{1\le k\le i-1}\left[d_k+L+\sum_{j=k+1}^{i-1}p_j\right]+p_i\le\min_{1\le k\le i}\left[d_k+L+\sum_{j=k+1}^{i}p_j\right]$$
$$+\min_{1\le k\le i-1}\left[d_k+L+\sum_{j=k+1}^{i-1}p_j\right] \tag{4.66}$$

$$\min_{1\le k\le i-1}\left[d_k+L+\sum_{j=k+1}^{i-1}p_j\right]\le\min_{1\le l\le i-1}\left[\min_{1\le k\le l}\left[d_k+L+\sum_{j=k+1}^{l}p_j\right]\right.$$
$$\left.+\min_{1\le k\le l-1}\left[d_k+L+\sum_{j=k+1}^{l-1}p_j\right]+\sum_{j=l+1}^{i-1}p_j\right] \tag{4.67}$$

したがって不等式 (4.58) は不等式 (4.62) と等価である．ここでもし仕事 J_iが機械 M_3 において処理する必要が無ければ仕事 J_iは明らかに実行可能であり，i について実行可能

である条件を示す不等式は明らかに成立するはずである．もし仕事 J_iが機械 M_3において処理されているならばその時には機械 M_1と M_2においては実行可能な上限まで処理を行っているはずである．すなわち，$M_1(i) = d_i + L$ かつ $M_2(i) = M_1(i-1)$ が成立する．よって以下の不等式が成立する．

$$\min_{1\le l\le i-1}\left[M_1(l) + M_1(l-1) + \sum_{j=l+1}^{i-1} p_j\right] + p_i \ge M_1(i) + M_1(i-1) \tag{4.68}$$

不等式 (4.68) より式 (4.69) が導かれる．

$$\min_{1\le l\le i}\left[M_1(l) + M_1(l-1) + \sum_{j=l+1}^{i} p_j\right] = M_1(i) + M_1(i-1) \tag{4.69}$$

式 (4.69) より不等式 (4.62) は以下のように変形される．

$$\begin{aligned}\sum_{j=1}^{i} p_j \;\le\; & \min_{1\le k\le i}\left[d_k + L + \sum_{j=k+1}^{i} p_j\right] + \min_{1\le l\le i-1}\left[\min_{1\le k\le l}\left[d_k + L + \sum_{j=k+1}^{l} p_j\right]\right.\\ & \left.+ \min_{1\le k\le l-1}\left[d_k + L + \sum_{j=k+1}^{l-1} p_j\right] + \sum_{j=l+1}^{i-1} p_j\right]\end{aligned} \tag{4.70}$$

(4.70) をさらに変形することによって以下の不等式を得る．

$$3L \ge \left[\sum_{j=1}^{i} p_j - M_1'(i) - \min_{1\le l\le i-1}\left[M_1'(l) + M_1'(l-1) + \sum_{j=l+1}^{i-1} p_j\right]\right] \tag{4.71}$$

ただしここで $M_1'(i) = \min\limits_{1\le k\le i}\left[d_k + \sum_{j=k+1}^{i} p_j\right]$ とする．

不等式 (4.71) が $i = 1, \cdots, n$ について成立する時またその時のみ実行可能スケジュールが存在する．

以上の議論より以下の定理を得る．

定理 4.5 最大納期遅れの最適値 $L^*_{\max}$について以下の式が成立する．

$$L^*_{\max} = \frac{1}{3}\max_{1\le i\le n}\left[\sum_{j=1}^{i} p_j - M_1'(i) - \min_{1\le l\le i-1}\left[M_1'(l) + M_1'(l-1) + \sum_{j=l+1}^{i-1} p_j\right]\right] \tag{4.72}$$

4.5.2　2目的問題

2機械の場合と同様に最大納期遅れ最小化問題より最大完了時間 $C_{\max}$と最大納期遅れ $L_{\max}$という2目的の問題を考える．仕事 J_iについての納期 d_i^+を $d_i^+ = \min\left[d_i + L, C_{\max}\right]$

と定義する．式 (4.55) における $d_i + L$ を d_i^+ におきかえることより以下の等式を得る．ただし$\sum_{j=i+1}^{i} p_j = 0$ とする．

$$M_1(i) = \min_{1 \leq k \leq i} \left[\sum_{j=1}^{i} p_j, \min\left[d_k + L, C_{\max}\right] + \sum_{j=k+1}^{i} p_j \right] \tag{4.73}$$

$\min_{1 \leq k \leq i} \left[C_{\max} + \sum_{j=k+1}^{i} p_j \right] = C_{\max}$ であるから，式 (4.73) は以下の式と等価となる．

$$M_1(i) = \min_{1 \leq k \leq i} \left[\sum_{j=1}^{i} p_j, d_k + L + \sum_{j=k+1}^{i} p_j, C_{\max} \right] \tag{4.74}$$

さらに前節と同様にして以下の不等式を得る．

$$\begin{aligned} \sum_{j=1}^{i} p_j \ \leq \ & \min_{1 \leq l \leq i} \left[M_1(l) + M_1(l-1) + \sum_{j=l+1}^{i} p_j, \sum_{j=1}^{i} p_j \right] \\ & + \min_{1 \leq l \leq i-1} \left[M_1(l) + M_1(l-1) + \sum_{j=l+1}^{i-1} p_j, \sum_{j=1}^{i} p_j \right] - M_1(i-1) \end{aligned} \tag{4.75}$$

実行可能である条件を示す不等式 (4.54) は不等式 (4.75) が成立する時またその時のみ成立する．不等式 (4.75) は以下の 不等式 (4.76) と等価となる．

$$\begin{aligned} \sum_{j=1}^{i} p_j \ \leq \ & \min_{1 \leq l \leq i} \left[M_1'(l) + M_1'(l-1) + \sum_{j=l+1}^{i} p_j \right] \\ & + \min_{1 \leq l \leq i-1} \left[M_1'(l) + M_1'(l-1) + \sum_{j=l+1}^{i-1} p_j \right] - M_1'(i-1) \end{aligned} \tag{4.76}$$

ただし $M_1'(i) = \min_{1 \leq k \leq i} \left[d_k + L + \sum_{j=k+1}^{i} p_j, C_{\max} \right]$ である．

式 (4.69) を用いて，不等式 (4.76) を以下のように展開する．

$$\sum_{j=1}^{i} p_j \leq M_1'(i) + \min_{1 \leq l \leq i-1} \left[M_1'(l) + M_1'(l-1) + \sum_{j=l+1}^{i-1} p_j \right] \tag{4.77}$$

ゆえに不等式 (4.70) は以下の 4 つの不等式と等価である．

$$3L \ \geq \ \sum_{j=1}^{i} p_j - M_1''(i) - \min_{1 \leq l \leq i-1} \left[M_1''(l) + M_1''(l-1) + \sum_{j=l+1}^{i-1} p_j \right] \tag{4.78}$$

$$2L + C_{\max} \ \geq \ \sum_{j=1}^{i} p_j - \min_{1 \leq l \leq i-1} \left[M_1''(l) + M_1''(l-1) + \sum_{j=l+1}^{i-1} p_j \right] \tag{4.79}$$

$$L + 2C_{\max} \geq \sum_{j=1}^{i} p_j - \min_{1\leq l \leq i-1}\left[M_1''(l-1) + \sum_{j=l+1}^{i-1} p_j\right] \tag{4.80}$$

$$3C_{\max} \geq \sum_{j=1}^{i} p_j \tag{4.81}$$

ただし

$$M_1''(i) = \min_{1\leq k\leq i}\left[d_k + \textstyle\sum_{j=k+1}^{i} p_j\right]$$

とする.

これらの不等式 (4.78)-(4.81) を満足する領域に実行可能解の点は存在する.

以上の議論より以下の定理を得る.

定理 4.6 最大納期遅れ L について以下の式が成立する.

$$\begin{aligned} L = \max\Bigg[& L^*_{\max}, \sum_{j=1}^{i} p_j - \min_{1\leq l\leq i-1}\left[M_1''(l) + M_1''(l-1) + \sum_{j=l+1}^{i-1} p_j\right] - C_{\max}, \\ & \sum_{j=1}^{i} p_j - \min_{1\leq l\leq i-1}\left[M_1''(l-1) + \sum_{j=l+1}^{i-1} p_j\right] - 2C_{\max}\Bigg] \end{aligned} \tag{4.82}$$

ここで

$$F_1 = \sum_{j=1}^{i} p_j - \min_{1\leq l\leq i-1}\left[M_1''(l) + M_1''(l-1) + \sum_{j=l+1}^{i-1} p_j\right], \tag{4.83}$$

$$F_2 = \sum_{j=1}^{i} p_j - \min_{1\leq l\leq i-1}\left[M_1''(l-1) + \sum_{j=l+1}^{i-1} p_j\right] \tag{4.84}$$

とする．もし $L^*_{\max} \geq F_1 - C^*_{\max}$かつ $L^*_{\max} \geq F_2 - 2C^*_{\max}$ ならば単一の最適解 $(C^*_{\max}, L^*_{\max})$ が存在する．この場合両方の目的関数が同時に最適化される．そうでなければ非劣解点は 2 つの線分上に位置する．$l_1(C_{\max}) = F_1 - C_{\max}, l_2(C_{\max}) = F_2 - 2C_{\max}$とし，$(C', L')$ を l_1 と l_2の交点であるとする．非劣解点は $(C^*_{\max}, F - C^*_{\max})$ と (C', L') を結ぶ線分と (C', L') と $C^+_{\max}, L^*_{\max})$ を結ぶ線分のいずれかに位置する (図 4.4参照)．ここで $C^+_{\max}$は $L = L^*_{\max}$と固定したときの最大完了時間の最小値である.

最大完了時間 $C_{\max}$または最大納期遅れ $L_{\max}$のいずれかの値が与えられたときもう一方の目的関数の値は式 (4.82) によって与えられる.

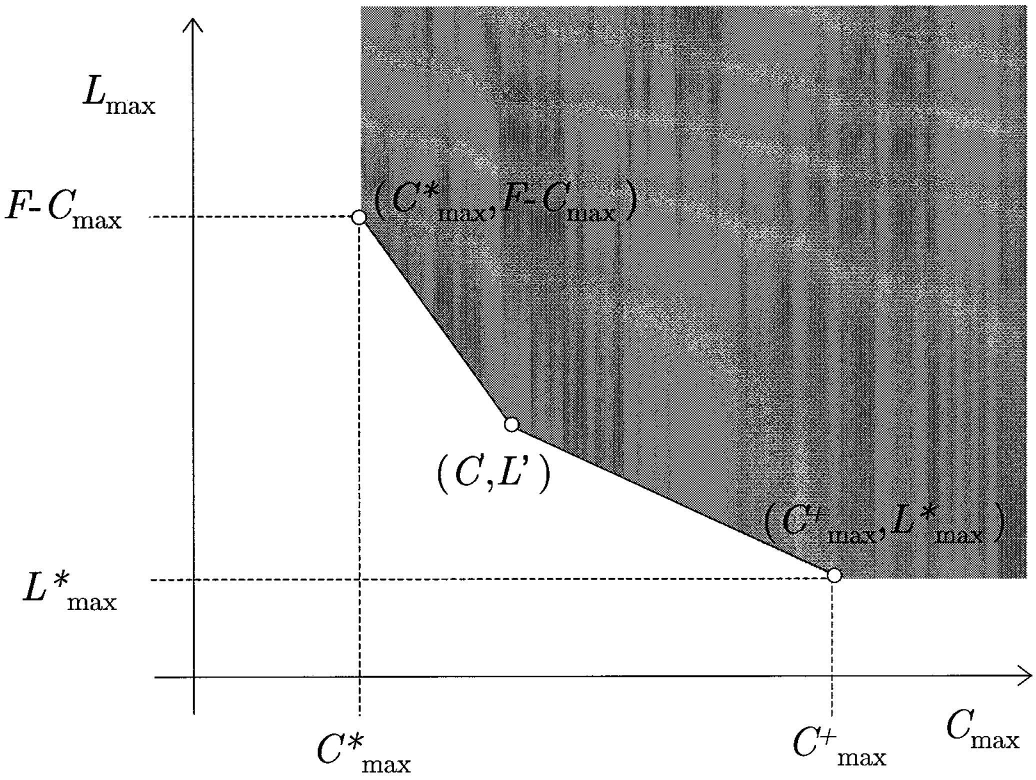

図 4.4 非劣解

4.6 結言

本章では 2 または 3 台の等価並列機械での 2 つの目的関数を持つスケジューリング問題について考察を行った．Sahni による全ての仕事の処理がその納期までに完了するという制約を満たす実行可能スケジュールを構築するアルゴリズムをもとに全ての仕事の処理の完了時間を明示するために新たなアルゴリズムを構築し，そのアルゴリズムを解析することによって各機械における各仕事の処理の完了時間を示す式を導出し，実行可能なスケジュールが存在するための条件を明らかにした．そしてこの条件式より，最大納期遅れを示す式を導出し，その納期を最大完了時間を考慮した修正納期に置き換えることによって最大完了時間と最大納期遅れの 2 つの目的関数の関係を示す式を導出し，2 つの目的関数における実行可能領域と非劣解を示す線分を示した．実行可能スケジュールを構成するた

めには，意思決定者がこれらの結果を用いて2つの目的関数の値を決定し，それらの値を持つようなスケジュールを本論文中で述べたようにそれぞれの納期を元の納期に最大納期遅れを加えたのものとし，それを用いてONERTを適用することにより求めることができる．これらの結果はさらに台数が多い場合にも拡張可能であると思われる．

第 5 章

資源によって変化する処理時間を持つ等価並列機械スケジューリング問題

5.1 緒言

現実の生産現場でのスケジューリング問題は人員や空間，資金，エネルギーといった資源によって制約を受ける．そのような問題において，意思決定者は時間による制約とともに資源による制約も考慮に入れなければならない．また資源の消費が仕事の処理に対しさまざまな影響を与えるような場合が存在する．例えばプロジェクトスケジューリングにおいて，プロジェクトの処理時間が人員や資金といった資源を割り当てることによってより短縮されるような場合である．

処理時間が可変であるようなスケジューリングは多く扱われている [23, 49, 54, 80, 86, 88, 93]．また資源の消費が仕事のパラメータに影響を与えるようなスケジューリング問題は実際の生産現場においては多く見ることができる．例えば製鉄における圧延加工においてインゴットを暖める際には適切な温度まで温度を上昇させるためにガスを消費する．このガスの供給量を増やすことによって処理時間を短縮することができる [99]．このような問題は Vickson によって研究が始められた [94, 95]．しかし大部分の研究は単一機械における問題を扱っている [45, 47, 46]．本研究では資源の消費が仕事の実行時間に影響を与えるような並列機械問題を扱う．ここで扱う問題の詳細を以下に示す．

n 個の仕事と m 台の等価並列機械があるとする．それぞれの仕事 J_i は納期と資源の投入によって変化する処理時間，すなわちその仕事に割り当てられた資源の量の線形関数と

して表わされる処理時間を持つ．この問題の目的は納期を満足し，かつ資源消費に伴うコストを最小化するような資源の割り当てを決定しスケジュールを構築することである．この問題を扱うために前章で述べたSahni による $Pm|d_i, pmtn|-$問題に対するアルゴリズムをもとに新たなアルゴリズムを開発する．

次節では資源によって変化する処理時間を持つ等価並列機械スケジューリング問題について述べる．さらに5.3節において資源の消費に伴うコストを最小とする実行可能スケジュールを与えるアルゴリズムを提案する．5.4節において最大納期遅れと資源消費に伴うコストの二つの目的を持つ問題を考え，これらの二つの目的関数値が実行可能領域で凸集合となっていることを示す．5.5節で結論を述べる．

5.2　資源消費コスト最小化問題

本節では本章の目的である資源消費コスト最小化問題といくつかの定理について述べる．以下の問題を考える．

- n 個の仕事 $J_1, \cdots, J_n$がある．
- $m(\geq 2)$ 台の等価並列機械 $M_1, \cdots, M_m$がある．
- 全ての仕事は時間0から処理が可能である．
- 各仕事 J_iは資源に対する重み付け w_i, 納期 d_i，資源に依存する処理時間

$$p_i = b_i - a_i x_i \tag{5.1}$$

  を持つ．ただし b_iは J_iの標準実行時間であり，x_iは J_iに割り当てられる資源の量であり，$x_i \geq 0$ とする．a_iは J_iが資源によって短縮される実行時間の割合とする．

- J_iへの資源の割り当て量 x_iには上限 u_iがあるとする．ただし $u_i \leq b_i/a_i$が成立するものとし，処理時間が負になることはないものとする．
- それぞれの仕事は任意の時点で中断可能とし，任意の時点で同じ機械または違う機械によって再開可能であるとする．

- この問題の目的は全ての仕事が納期を満足し，かつ資源の消費に伴うコストの合計

$$\sum_{i=1}^{n} w_i x_i \tag{5.2}$$

を最小化するような実行可能スケジュールを求めることである．

この問題を解くアルゴリズムを開発するためにいくつかの解析を行う．一般性を失わずに $d_1 \leq d_2 \leq \cdots \leq d_n$ を仮定する．$M_k(i)$ を仕事 J_i の機械 M_k における処理の完了時間とする．ただし仕事 J_i が機械 M_k において処理が行われない場合，時間 0 の処理を行っているものとする．すなわち $M_k(i) = M_k(i-1)$ が成立するものとする．

前章において実行可能スケジュールが存在するための必要十分条件を得ている．もし不等式 (4.20) が全ての仕事について成立するならば資源の割り当ては不要である．もし不等式 (4.20) がある仕事 J_i について不成立ならば不等式 (4.20) が成立するように資源を割り当てる必要がある．

まずいくつかの定義を行う．

定義 5.1 π_i を ONERT2 によって構築された J_1 から $J_i (i = 1, \cdots, n)$ までの仕事を含む部分スケジュールとする．

定義 5.2 部分スケジュール π_i において，$\delta_{(l,k)}$ を仕事 $J_l (i = 1, \cdots, i)$ について機械 $M_k (k = 1, \cdots, m)$ での処理時間とする．すなわち以下の式で表される．

$$\delta_{(l,k)} = \begin{cases} \min\{p_l, d_l - M_k(l-1)\}, & k = 1 \\ M_k(l) - M_k(l-1), & k > 1 \end{cases} \tag{5.3}$$

さらに仕事 J_l に対し k_l を以下の様に定義する．

$$k_l = \max\{k \mid \delta_{(l,k)} > 0\} \tag{5.4}$$

すなわち k_l は J_l について $\delta_{(l,k)} > 0$ を満たすような機械のインデックスのうち最大のものである．

定義 5.3 J_l について $\mathcal{J}_l$ を以下の条件を満たす仕事の系列 $\{J_{(l,0)}, J_{(l,1)}, \ldots, J_{(l,m-k_l)}\}$ とする．$s = 1, \cdots, m - k_l$ に対し，

$$J_{(l,0)} = J_l \tag{5.5}$$

$$J_{(l,s)} = \min\{J_i \mid \text{仕事 } J_i \text{に対し } k_i > k_l + s \text{ かつ } (l, s-1) < i\} \tag{5.6}$$

すなわち仕事 $J_{(l,s)}$は $k_{(l,s)} > k_l + s$ かつ $(l, s-1) < (l, s)$ を満たす全ての仕事のうち最小のインデックスを持つ仕事である.

定義 5.4 $\overline{\mathcal{J}_l}$について，δ_lを以下のように定義する.

$$\delta_l = \min\left[u_l, \min_{J_{(l,h)} \in \overline{\mathcal{J}_l}} \delta_{(h, k_l+h)}\right] \tag{5.7}$$

例 5.5 図 5.1に示すような 3 機械問題における部分スケジュールπ_6を考える. このとき $\delta_{l,k}(l = 1, \cdots, 6, k = 1, 2, 3)$ とまた $k_l(l = 1, \cdots, 6)$ は表 5.1に示すように与えられる.

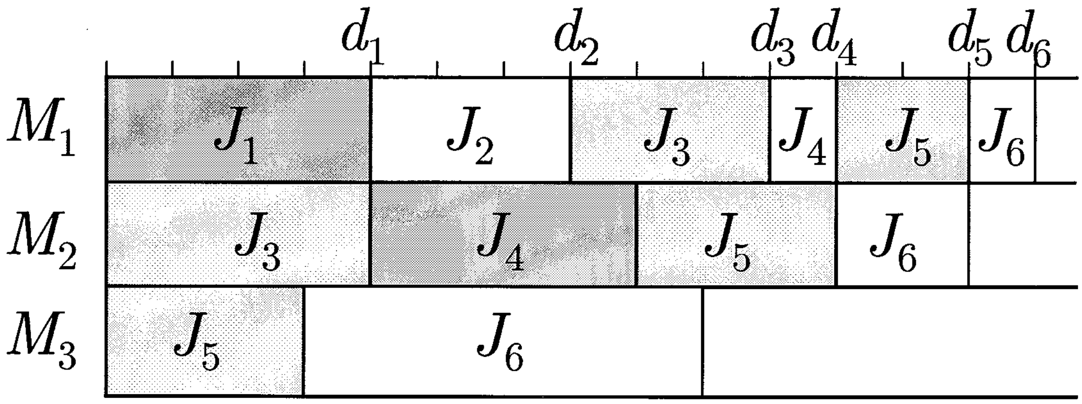

図 5.1 部分スケジュールπ_6

表 5.1 $\delta_{l,k}(l = 1, \cdots, 6, k = 1, 2, 3)$, $k_l(l = 1, \cdots, 6)$

	J_1	J_2	J_3	J_4	J_5	J_6
$\delta_{(l,1)}$	4	3	3	1	2	1
$\delta_{(l,2)}$	0	0	4	4	3	2
$\delta_{(l,3)}$	0	0	0	0	3	6
k_l	1	1	2	2	3	3

このとき$\overline{\mathcal{J}}_l (l = 1, \cdots, n)$ は

$$\begin{aligned}
\overline{\mathcal{J}}_1 &= \{J_1, J_3, J_5\} &(5.8)\\
\overline{\mathcal{J}}_2 &= \{J_2, J_3, J_5\} &(5.9)\\
\overline{\mathcal{J}}_3 &= \{J_3, J_5\} &(5.10)\\
\overline{\mathcal{J}}_4 &= \{J_4, J_5\} &(5.11)\\
\overline{\mathcal{J}}_5 &= \{J_5\} &(5.12)\\
\overline{\mathcal{J}}_6 &= \{J_6\} &(5.13)
\end{aligned}$$

となる．

もし資源を割り当てることなしに実行可能なスケジュールを構築できない場合，すなわち $x_i = 0 (i = 1, \cdots, n)$ であるときにすべての仕事が納期に間に合うような実行可能スケジュールを構築することができず，資源の割り当てが必要になったときに，その割り当てがスケジュールが実行可能になるように，すなわち資源を割り当てることによって式 (4.20) が成立するような割り当て方がされなければならない．ここで資源の割り当て方について以下の性質が成り立つ．

補題 5.6 ある部分スケジュールπにおいて仕事 $J_l (l = 1, \cdots, i)$ の機械 M_kでの仕事の完了時間を $M_k(l)$ とし，またπにおいてある仕事 $J_s (l = 1, \cdots, i-1)$ に資源 x を割り当て，スケジュールを再構築した部分スケジュールをπ'とし，そこでの仕事 $J_l (l = 1, \cdots, i-1)$ の機械 M_kでの仕事の完了時間を $M'_k(l)$ とする．ただし資源の割当量 x は以下を満たすものとする．

$$a_s x < \min\{\delta_{(s,k_s)}, \delta_{(s+1,k_s+1)}\} \tag{5.14}$$

このとき以下の式が成立する．

$k_{s+1} \leq k_s$のとき

$$\begin{aligned}
M'_k(s+1) &= M_k(s+1), \quad k = 1, \ldots, k_s - 1 &(5.15)\\
M'_{k_s}(s+1) &= M_{k_s}(s+1) - a_s x &(5.16)\\
M'_k(s+1) &= M_k(s+1), \quad k = k_s + 1, \ldots, m &(5.17)
\end{aligned}$$

$k_{s+1} > k_s$のとき

$$
\begin{aligned}
M'_k(s+1) &= M_k(s+1), \quad k = 1, \ldots, k_s & (5.18)\\
M'_{k_s+1}(s+1) &= M_{k_s+1}(s+1) - a_s x & (5.19)\\
M'_k(s+1) &= M_k(s+1), \quad k = k_s+2, \ldots, m & (5.20)
\end{aligned}
$$

証明

仮定より以下の式が成立する.

$$
\begin{aligned}
M'_k(s) &= M_k(s), \quad k = 1, \ldots, k_s - 1 & (5.21)\\
M'_{k_s}(s) &= M_{k_s}(s) - a_s x & (5.22)\\
M'_k(s) &= M_k(s), \quad k = k_s + 1, \ldots, m & (5.23)
\end{aligned}
$$

$k_{s+1} \le k_s$のとき成立するときは明らかである.

$k_{s+1} > k_s$のとき部分スケジュールπ'_iにおいて，$\delta'_{(l,k)}$を仕事 $J_l (l = 1, \cdots, i)$ について機械 M_kでの処理時間とする．ここで式 (5.20) が成立することは明らかである.

以下の二つの場合を考える.

$k_{s+1} = k_s + 1$ のとき

明らかに$k < k_s$に対し，式 (5.19) は成立する．$k = k_s$のとき，式 (5.14) より割り当てられる資源の量は高々

$$x < \delta_{(s+1,k_s+1)}/a_s \tag{5.24}$$

なので仕事 J_{s+1}の機械 M_{k_s+1}における処理時間$\delta'_{(l,k)}$は以下を満たす.

$$0 \le \delta'_{(l,k)} \le \delta_{(l,k)} \tag{5.25}$$

よって仕事 J_{s+1}は機械 M_{k_s}において限界まで処理されるので

$$
\begin{aligned}
M'_{k_s}(s+1) &= M_{k_s-1}(s)\\
&= M_{k_s+1}(s) & (5.26)
\end{aligned}
$$

が成立．また仕事 J_{s+1}の機械 M_{k_s+1}での完了時間 $M'_{k_s+1}(s+1)$ は

$$M'_{k_s+1}(s+1) = M_{k_s+1}(s+1) - a_s x \tag{5.27}$$

となる．$k > k_s + 1$ では J_{s+1}は処理は行われておらず，かつ式 (5.23) が成立するので式 (5.20) が成立する．

$k_{s+1} > k_s + 1$ のとき

$k_{s+1} = k_s + 1$ の時と同様に $k \leq k_s$に対し，式 (5.19) は成立する．また仕事 J_{s+1}は機械 M_{k_s}において限界まで処理されるので処理の完了時間 $M'_{k_s+1}(s+1)$ は

$$\begin{aligned} M'_{k_s+1}(s+1) &= M'_{k_s}(s) \\ &= M_{k_s}(s) - a_s x \\ &= M_{k_s+1}(s+1) - a_s x \end{aligned} \tag{5.28}$$

となる．

また仕事 J_{s+1}の機械 M_{k_s}, M_{k_s+1}での部分スケジュールπでの処理時間の合計$\delta_{(s+1,k_s)} + \delta_{(s+1,k_s+1)}$と部分スケジュール$\pi'$での処理時間の合計$\delta'_{(s+1,k_s)} + \delta'_{(s+1,k_s+1)}$に対し

$$\delta_{(s+1,k_s)} + \delta_{(s+1,k_s+1)} = \delta'_{(s+1,k_s)} + \delta'_{(s+1,k_s+1)} \tag{5.29}$$

が成立する．よって $k > k_s + 1$ において式 (5.20) が成立する．

よって補題が成り立つ．

証明終

定理 5.7 部分スケジュールπ_iにおいて $J_l(l = 1, \ldots, i-1)$ が実行可能であり，J_iが実行不可能であるとする．すなわち，$M_m(l) < M_{m-1}(l-1)(l = 1, \ldots, i-1)$ かつ $M_m(i) > M_{m-1}(i-1)$ とする．そのとき J_iを実行可能とするためには処理時間 p_iを減らす，または他の仕事の処理時間を減らすことによって $M_m(i-1)$ の値を減らし，その合計が $M_m(i) - M_{m-1}(i-1)$ 以上になるように資源を割り当てなければならない．

証明

仕事 J_iに資源を割り当てることによって p_iを $M_m(i) - M_{m-1}(i-1)$ 以上減らすことができれば式 (4.20) が成立することは明らかである．また補題5.6より，$k \leq m-1$ の機械 M_kに対し，$M_k(i-1)$ が変化しても式 (4.20) の各項に変化を与えないことは明らかである．

$\delta = M_m(i) - M_{m-1}(i-1)$ とし，部分スケジュールπ_iの仕事 J_lの機械 M_kでの処理の完了時間を $M_k(l)$ とし，部分スケジュールπ_i'を J_iを除くある仕事に資源を割り当てた部分スケジュールとしその仕事 J_lの機械 M_kでの処理の完了時間を $M_k'(l)$ とする．次の二つの場合を考える．

$M_{m-1}'(i-1) = M_{m-1}(i-1) - \delta$のとき

J_iは各機械で限界まで処理されているので

$$M_{m-1}'(i) = M_{m-1}(i) \tag{5.30}$$

$$M_m'(i) = M_m(i) - \delta \tag{5.31}$$

式 (4.20) より，

$$\begin{aligned} M_{m-1}'(i-1) &= M_{m-1}(i-1) - \delta \\ &< M_m(i) - \delta \\ &= M_m'(i) \end{aligned} \tag{5.32}$$

となり，部分スケジュールπ'において式 (4.20) は不成立となる．

$M_m'(i-1) = M_m(i-1) - \delta$のとき

あきらかに

$$M_{m-1}'(i-1) = M_{m-1}(i-1) \tag{5.33}$$

$$M_m'(i) = M_m(i) - \delta \tag{5.34}$$

が成立し，

式 (4.20) より，

$$\begin{aligned} M_{m-1}'(i-1) &= M_{m-1}(i-1) \\ &= M_m(i) - \delta \\ &= M_m'(i) \end{aligned} \tag{5.35}$$

となり，部分スケジュールπ'において式 (4.20) は成立する．

よって定理は成立．

証明終

部分スケジュールπ_iにおいて，$M_j(l)$ を仕事 $J_l(l=1,\cdots,i)$ の M_jでの処理の完了時間とする．またπ'_iを資源 x_lを $J_l(l=1,\cdots,i)$ に割り当て処理時間を$\delta(<\delta_{(l,k_l)})$ だけ減らした後，構築した J_1から J_iまでの部分スケジュールとし，$M'_j(i)$ を部分スケジュールπ'_iにおける J_iの M_jでの仕事の処理の完了時間とする．このとき以下の定理が成り立つ．

定理 5.8 もし J_lについて$\mathcal{J}_l$が存在するならば，

$$M'_m(i) < M_m(i) - \min\left[\delta, \delta_l\right] \tag{5.36}$$

が成立し，一方もし存在しなければ

$$M'_m(i) = M_m(i) \tag{5.37}$$

が成立する．

証明

J_lより前で処理される仕事 $J_h(h<l)$ については明らかに以下の式が成立する．

$$M'_j(h) = M_j(h), \quad j=1,\ldots,m \tag{5.38}$$

また

$$\begin{aligned} M'_j(l) &= M_j(l), \quad j<k_l \text{または} j>k_l &(5.39)\\ M'_{k_l}(l) &= M_{k_l}(l)-\delta &(5.40)\end{aligned}$$

が成立．

J_l以降の仕事 $J_h(h>l)$ に対し，以下を考える．$\delta' < \delta_{(h,k_h)}$かつ $M'_{k_{h-1}}(h-1) = M_{k_{h-1}}(h-1)-\delta'$を仮定する．

1. もし $k_h < k_{h-1}$ならば

$$M'_j(h) = M_j(h), \quad j < k_h \text{または} j > k_h \tag{5.41}$$

$$M'_{k_h}(h) = M_{k_h}(h) - \delta' \tag{5.42}$$

2. もし $k_h \geq k_{h-1}$ならば

$$M'_j(h) = M_j(h), \quad j \leq k_h \text{または} j > k_h + 1 \tag{5.43}$$

$$M'_{k_h+1}(h) = M_{k_h+1}(h) - \delta' \tag{5.44}$$

よって，もし J_lにたいし$\mathcal{J}_l$が存在するならば，$\mathcal{J}_l$の定義より，J_lの処理時間をδ_l減らすことによって$\mathcal{J}_l$におけるそれぞれの仕事 $J_{(l,h)}$において以下の式が成立する．

$$M'_j(h) = M_j(h), \quad j \leq k_{(l,h)} \text{または} j > k_{(l,h)} + 1 \tag{5.45}$$

$$M'_{k_{(l,h)}+1}(h) = M_{k_{(l,h)}+1}(h) - \delta_l \tag{5.46}$$

$\mathcal{J}_l$における最後の仕事は M_mにおける処理の完了時間をδ_l減らす．よって最後の機械における各仕事の完了時間に対し以下の式が成立する．

$$M'_m(h) = M_m(h), \quad h < (l, m - k_l) \tag{5.47}$$

$$M'_m(h) = M_m(h) - \delta, \quad h \geq (l, m - k_l) \tag{5.48}$$

もし$\delta > \delta_l$ならばδがδ_lを超えた時点で再構築したスケジュールにおいて$\mathcal{J}_l$が定義 5.3を満たさなくなる．よって定理が成立．

証明終

もし J_lの処理の完了時間をδ_l減らし，スケジュールを再構築したときに他の$\mathcal{J}_l$が存在するならば，さらに $M_m(i)$ を減らすことが可能である．そうでなければそれ以上減らすことは不可能である．

定理 5.7と定理 5.8より，ONERT2 によって構築されたスケジュールにおいて，もしある仕事 J_iが実行不可能ならば，J_iもしくは定義 5.3を満たす系列を持ち J_iに先行する仕事に対し資源を割り当てなければならない．

5.3 資源消費コスト最小化アルゴリズム

前節の議論より ONERT2 をもとに問題を解くためのアルゴリズムを提案する．このアルゴリズムは前章でのアルゴリズム ONERT2 をもとに納期順に仕事を割り当てていき，資源の割り当て無しに実行可能な仕事は ONERT2 と同様に処理区間を決定し，資源の割り当て無しに処理を実行することが不可能な仕事 J_iが存在したときに，J_iもしくは$\overline{\mathcal{J}_l}$が存在するような仕事 J_lのうち資源の割り当てコストが最も小さいものすなわち w_i/a_iがもっとも小さいものに対し J_iが実行可能となるか，もしくは割り当てることで J_iの実行可能性に寄与するだけ資源を割り当てる．以下に ONERT3 を示す．仕事のリスト $\mathcal{C}$を仕事 J_iを w_i/a_i の非減少順に並べたものとする．

アルゴリズム **ONERT3**

for $i \leftarrow 1$ to n do
begin
    $r_i \leftarrow p_i$
    for $j \leftarrow 1$ to m do
        $RPT(j) = d_i - M_j(i-1)$
    if $RPT(1) > r_i$ then
    begin
        $M_1(i) \leftarrow M_1(i-1) + r_i$
        // 仕事 J_iは機械 M_1によって全て処理される．
        $M_j(i) \leftarrow M_j(i-1), \quad j = 2, \cdots, m$
        $r_i \leftarrow 0$
    end
else
begin
    $M_1(i) \leftarrow d_i$
    // 機械 M_1は仕事 J_iを $[M_1(i-1), d_i]$ 間処理される．
    $r_i \leftarrow r_i - (d_i - M_1(i-1))$
    while $r_i > 0$ and $j \leq m$ do
    begin

```
        if RPT(j) < r_i then
        begin
            M_j(i) ← M_{j-1}(i-1)
            // 機械 M_1は仕事 J_iを [M_j(i-1), M_{j-1}(i-1)] 間処理される.
            r_i ← r_i - (M_{j-1}(i-1) - M_j(i-1))
        end else
        begin
            M_j(i) ← M_j(i-1) + r_i
            // 機械 M_1は仕事 J_iを [M_j(i-1), M_j(i) + r_i] 間処理される.
            r_i ← 0
            M_k(i) ← M_k(i-1),    k = j+1, ..., m
        end
    if r_i > 0 then
    begin
        δ ← r_i - (M_{m-1}(i-1) - M_m(i-1))
        while C = ∅ and δ > 0 do begin
            J_l ← Cの先頭の仕事
            if l = i and x_i < u_i then
            begin
                x_i ← min[u_l, δ/a_l]
                δ ← δ - a_l x_l
                J_lを Cから削除する.
            end
            else if l = i and x_i < u_i then
            begin
                if 𝒥_lが存在するならば begin
                    x_i ← min[u_l, δ/a_l]
                    δ ← δ - a_i x_i
                    J_lを Cから削除する.
                end else
```

```
                                begin
                                    J_l を C から削除する.
                                end
                            end
                        end
                    end
                end
```

例 5.9 以下の 6 仕事，3 機械の問題を考える．表 5.2は各仕事 J_iの処理時間 p_i，納期 d_i，資源の消費に対する単位量あたりのコスト w_i，資源の消費による単位量あたりの処理時間の変化 a_i，資源消費量の上限 u_iを示している．

表 5.2 各仕事のパラメータ

仕事	J_1	J_2	J_3	J_4	J_5	J_6
p_j	4	3	8	8	3	11
d_j	4	7	8	10	13	14
w_j	3	4	5	6	2	1
a_j	1	1	1	1	1	1
u_j	2	2	2	2	1	1

J_1から J_5までは，資源の割り当てなしにスケジュールを構築することができ，J_6は資源の割り当て無しでは実行不可能である (図 5.9参照)．

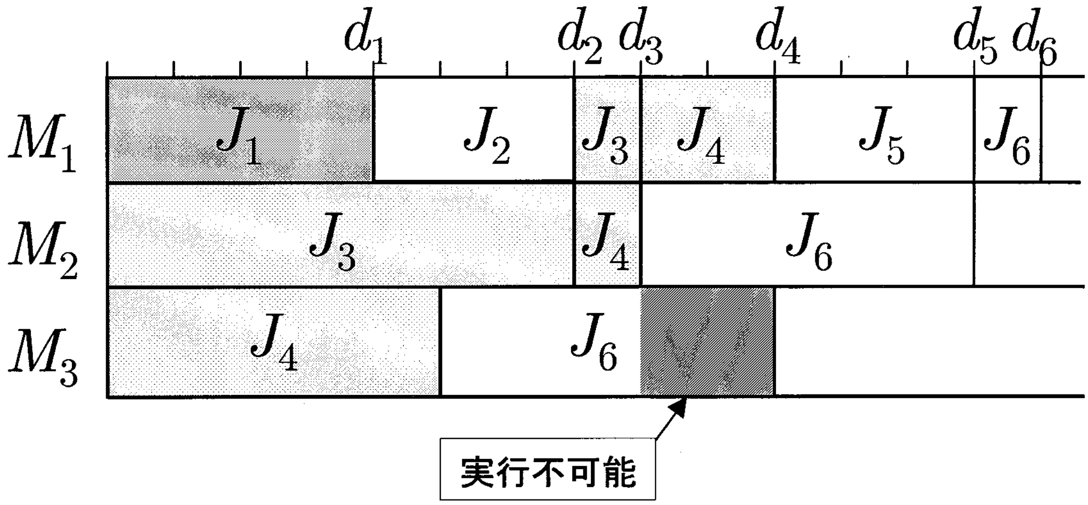

図 5.2 資源の割り当て無し

ここで $\mathcal{C} = \{J_6, J_5, J_1, J_2, J_3, J_4\}$ であり，$\delta = 2$ である．

1. $\mathcal{C}$の先頭は J_6であるので，資源 $x_6 = 1 (= u_6)$ を J_6に割り当て $M_3(6)$ を 1 減らすことができる．

   スケジュールを再構築すると$\delta = 1$ となる．

2. 次の候補は J_5であるが，$\overline{\mathcal{J}_5}$は存在しない．よって J_5に資源は割り当てられない．

3. 次の候補は J_1であり，$\overline{\mathcal{J}_1}$が $\{J_1, J_3, J_4\}$ として存在する．また$\delta_{(1,1)} = 3, \delta_{(3,2)} = 7, \delta_{(4,3)} = 5$, $\delta_1 = 3$, $x_1 = 0 > 2 = u_1$である．よって J_1に資源 $x_1 = 1 = \delta/a_1$を割り当て $M_3(6)$ を 1 減らすことができる．

これにより J_6は資源の割り当てることによって実行可能となる．図 5.9は資源の割り当て後のスケジュールを示している．

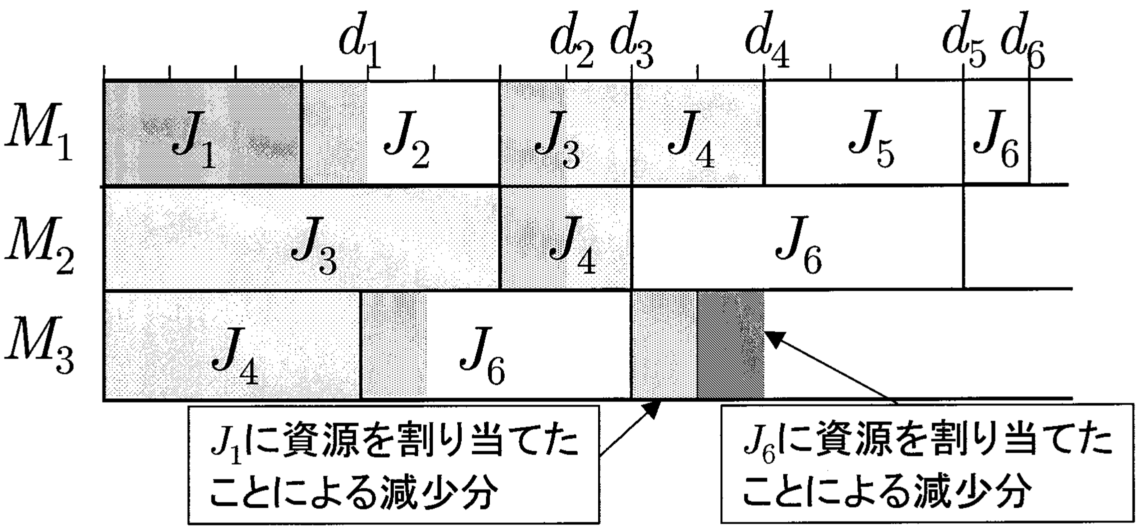

図 5.3 資源の割り当て後

ここでONERT3の正当性を示す.

補題 5.10 π_iを仕事 J_1から J_iまでのある部分スケジュールとし，π_i'をπ_iにおいて資源を$J_l(l = 1, \cdots, i)$ に割り当て，その処理時間を$\delta(< \delta_l)$ だけ減らし，スケジュールを再構築したものとする．また J_hに対し$\delta_h^{\pi_i}$と$\delta_h^{\pi_i'}$をそれぞれπ_iとπ_i'において定義 5.4によって定義された値とする.

$J_h(h = 1, \cdots, i)$ に対し以下の関係が成立する.

$$\delta_h^{\pi_i} - \delta \leq \delta_h^{\pi_i'} \leq \delta_h^{\pi_i} \tag{5.49}$$

証明

J_hに対し$\delta_{(h,k)}^{\pi_i}$と$\delta_{(h,k)}^{\pi_i'}$をそれぞれπ_iとπ_i'において定義 5.2によって定義された値とする.

以下の式が明らかに成立する.

$$\delta_{(l,k)}^{\pi_i'} = \delta_{(l,k)}^{\pi_i}, \quad k \neq k_l \tag{5.50}$$

$$\delta_{(l,k_l)}^{\pi_i'} = \delta_{(l,k_l)}^{\pi_i} - \delta \tag{5.51}$$

定理5.10の証明より，以下の式が成立する．$J_g \in \overline{\mathcal{J}_l}$のとき

$$\delta^{\pi'_i}_{(g,k)} = \delta^{\pi_i}_{(g,k)}, \quad k \leq \overline{k_{(l,g)}} \tag{5.52}$$
$$\delta^{\pi'_i}_{(g,k)} = \delta^{\pi_i}_{(g,k)} - \delta, \quad k > \overline{k_{(l,g)}} \tag{5.53}$$

$J_g \notin \overline{\mathcal{J}_l}$のとき

$$\delta^{\pi'_i}_{(g,k)} = \delta^{\pi_i}_{(g,k)}, \quad k = 1, \cdots, m \tag{5.54}$$

もし$\overline{\mathcal{J}_l} \cap \overline{\mathcal{J}_h} = \emptyset$ならば$\delta^{\pi'_i}_h$は変わらない．そうでなければ定義5.4より，$\delta^{\pi'_i}_h$は高々$\delta$減るだけである．よって定理は成立する．

証明終

定理 5.11 ONERT3は資源消費に伴うコストを最小とするような実行可能スケジュールを構築する．

<u>**証明**</u>

Sahniのアルゴリズムの正当性より，任意の実行可能スケジュールに対し資源の割り当てにおいて等価であり，かつ納期の非減少順に並んでいるスケジュールが存在する．また以下は明らかである．もしJ_iが実行不可能ならばJ_iに先行するような仕事に対して資源は割り当てられなければならず，J_iの後に処理を行う仕事に対し資源を割り当てることによってJ_iを実行可能にすることはできない．

背理法と帰納法によって証明する．

π_iをONERT3によって構築された部分スケジュールとし，別の最適な実行可能スケジュールπ'_iが存在するとする．すなわち$f(\pi'_i) < f(\pi_i)$，ここで$f(\pi_i)$と$f(\pi'_i)$は各スケジュールの資源消費に伴うコストである．以下の場合を考える．

1. $i = 1$ならば，資源の割り当て方は一通りしか存在しないのでπ'_iのようなスケジュールは存在しない．よって仮定は矛盾する．
2. $i = z - 1 (z = 2, \cdots, n)$において定理が成立すると仮定する．$i = z$とし次の2つの場合を考える．

(a) もし J_iがさらなる資源の割り当て無しに実行可能であるならば，帰納法の仮定よりπ'_iのようなスケジュールは存在しないことは明らかである．よって背理法の仮定は矛盾を導く．

(b) もし J_iを実行可能とするためにさらなる資源の割り当てが必要であるならば，x_lと x'_lをそれぞれπ_iとπ'_iにおいて J_iを実行可能とするために J_lに対して割り当てられた資源の量とする．また$\delta_l = a_l x_l$かつ$\delta'_l = a_l x'_l$とすると以下の式が成立する．

$$\sum_{l=1}^{i} \delta_l = \sum_{l=1}^{i} \delta'_l \tag{5.55}$$

π_iとπ'_i資源の割り当て方に違いが存在するはずなので，$\delta_l < \delta'_l$となるような仕事 J_lが存在する．

ONERT3 において w_l/a_lの順に，限界まで資源は割り当てられるので $w_h/a_h < w_l/a_l$であり，π_iにおいて J_hに対する資源の割り当てによって J_lに対する資源の割り当て $x'_l - x_l$が無駄になっている仕事 J_hが存在するはずである．また定理 5.10より，もし J_hに対し資源を割り当て処理時間をδ だけ減らしたときに，J_lの処理時間を減らして実行可能にすることのみに寄与する量の上限は高々δ減るだけである．

したがってπ'_iにおいて，J_hに対し資源を割り当て直すことによって資源消費に伴うコストの総和を減らすことができる．またこれらの操作を繰り返すことによりπ'_iはπ_iと等価になる．よって背理法の仮定は矛盾を導く．

したがって定理は成立.

証明終

5.4 2 目的問題

本節ではこれまでの結果をもとに資源消費に伴うコストと最大納期遅れの 2 つの目的関数を持つ問題を考える．

- n 個の仕事 $J_1, \cdots, J_n$がある．

- $m(\geq 2)$ 台の等価並列機械 $M_1, \cdots, M_m$がある.
- 全ての仕事は時間0から処理が可能である.
- 各仕事 J_iには重み付け w_i，納期 d_i，資源に依存する処理時間

$$p_i = b_i - a_i x_i \tag{5.56}$$

  を持つ．ただし b_iは J_iの標準実行時間であり，x_iは J_iに割り当てられる資源の量，a_iは J_iが資源によって短縮される実行時間の割合とする.
- J_iへの資源の割り当て量 x_iには上限 u_iがあるとする．ただし $u_i \leq b_i/a_i$が成立するものとし，処理時間が負になることはないものとする.
- それぞれの仕事は任意の時点で中断可能とし，任意の時点で同じ機械または違う機械によって再開可能であるとする.
- あるスケジュールにおいて，それぞれ 仕事 J_iに対し，C_iをその仕事が完了した時刻，L_iを納期遅れすなわち $L_i = \max[C_i - d_i, 0]$ を定義すると最大納期遅れ $L_{\max}$は

$$L_{\max} = \max_{1 \leq i \leq n} L_i \tag{5.57}$$

  と定義され，また資源の消費に伴うコストの合計は

$$\sum_{i=1}^{n} w_i x_i \tag{5.58}$$

  となる．この問題の目的はこれらをを最小化するような実行可能スケジュールを求めることである.

4.4節で最大納期遅れ問題を考えたときと同様にパラメータ L を導入し，修正納期 $d'_i = d_i + L$ とし納期と置き換える．このとき L の最大値 $L_{\max}$は Labertoulle によってネットワークフロー問題を帰着し得ることができることが示されている [62]．このとき資源の消費は必要なく，それに伴うコストも0である．またこの L を減らすことによって実行可能なスケジュールが存在しなくなるか，もしくは $L = 0$ になったときの L の最小値を $L_{\min}$とする.

$L(L_{\min} \leq L \leq L_{\max})$ に対し，実行可能なスケジュールにおける資源の消費に伴うコストの最小値を $f(L)$ とすると以下の定理が成立する.

定理 5.12 $f(L)$ は区間 $[L_{\min}, L_{\max}]$ においてLに対し凹関数であり，区分的に線形で，かつ L に対し非増加関数である．

<u>**証明**</u>

L が増えるにしたがって $f(L)$ の可能な領域は増えるので非増加性は明らかである．$L = L_{\max}$ のとき一つ以上の仕事がちょうど修正納期に処理が完了している．これらの仕事の集合を $\mathcal{J}_{critical}$ とする．L を $L_{\max}$ より減らすことを考える．このとき J_i を $\mathcal{J}_{critical}$ のうち最小のインデックスを持つ仕事とするとある $J_l (l \leq i)$ に対し資源が割り当てられているはずである．このとき以下のうち少なくとも一つの場合が生じる

1. 新たに修正納期に処理が完了している仕事 J_s が生じ，$s < l$ である．
2. 仕事 J_l に割り当てられる資源の量 x_l がその上限に達する．すなわち $x_l = u_l$ となる．

これらの場合が生じるまでの間，$f(L)$ の線形性は明らかである．これらいずれかの場合が生じたときに，J_l にさらに資源を割り当てることは無意味であり，新たな仕事 $J_{l'}$ に資源を割り当てなければならない．ここで ONERT3 は $w_t/a_t (1 \leq t \leq n)$ の順に資源を割り当てる候補を探すので $w_l/a_l \leq w_{l'}/a_{l'}$ が成立する．よって $f(L)$ が凹関数であることが示された．

証明終

よって $(L, f(L))$ は区間 $[L_{\min}, L_{\max}]$ において非劣解集合をなし，適当な L を決定し，ONERT3 を納期を修正納期 $d'_i = d_i + L$ に置き換えた問題に適用することによって非劣スケジュールを得ることができる．

5.5 結言

本章では資源を消費することによって仕事の処理時間が短縮されるという性質を持つ等価並列機械問題を考えた．そしてこの問題の定式化を行い，資源の割り当てに関する幾つかの定理を導いた．そしてこの結果をもとに資源消費に伴うコストの総和 $\sum_{i=1}^{n} w_i x_i$ を最小

とするスケジュールを構築するためのアルゴリズムを与えた．また資源の消費と最大納期遅れとの関係について考察を行った．この結果は資源の総量が制約となる問題や資源の総量と最大完了時間または最大納期遅れといった目的関数との2目的問題に拡張可能であると思われる．

第6章

機械使用コストを持つ2目的スケジューリング問題

6.1 緒言

本章では等価並列機械において，各機械の使用に対しその使用時間に依存したコストがかかるスケジューリング問題を考える．従来のスケジューリング問題においてもコストを伴うスケジューリングは数多く考えられている．また各仕事の終了時間や納期ずれに対し異なる重みを与え，その総和を目的関数としているような場合もそれらをコストとして扱っているともいえる．しかしその中で従来の研究のほとんどが各仕事の処理にコストがかかる場合を扱っている [11, 13, 66, 82]．生産の現場においては，機械を利用するためにも燃料費，人件費，整備費用等の様々なコストがかかってくる．これらのコストは仕事をどのように実行するかというよりは，機械をいかに利用し，仕事を処理するかに対するもののほうがふさわしい．

またコストを伴うスケジューリング問題において，コストを単一の目的関数として扱う場合が多いが，単純にコストを最小にするだけでは最大完了時間や納期遅れなどの他の制約や目的関数は考慮されない．現実には最大完了時間などの処理の実行時間における条件も制約となるためにそれらの制約を考え，その上でコストをできる限り低く押さえるようなスケジュールが必要となる．本章では機械使用コストを持つスケジューリング問題を考え，最大完了時間と機械使用コストの2つの目的関数を考える．そして各目的関数を最小とするアルゴリズムを与え，またこの目的関数間の関係を示す式を導出する．

次節においてまず本章で考える問題に対し実行可能解が存在することを保証するためにMasuda 他による研究 [68] について述べる．次に 6.3節において機械の利用時間に対し制約がある問題を考え，その時のコストの総和を最小にするアルゴリズムを与える．さらに 6.4節においてコストの総和に対し制約がある問題についてその時の最大完了時間が最小となるアルゴリズムを与える．これらの結果より 6.5節において機械の利用時間と機械の利用に対するコストの総和の関係を求め，機械の利用時間と機械の利用のコストの総和の 2 目的の問題において，上記のアルゴリズムで求めた解が非劣解となっていることを示し，最後に 6.6節において結論を述べる．

6.2　機械の利用時間に制限がある場合の割り当てアルゴリズム

Masuda 他 [68] によって等価並列機械において機械の利用時間に制限があるときに，実行可能なスケジュールを与えるアルゴリズムはすでに提案されている．本節ではそのアルゴリズムについて紹介する．

- n 個の仕事 $J_1, \cdots, J_n$がある．
- m 台の等価な並列機械 $M_1, \cdots, M_m$がある．
- 仕事 $J_i (i = 1, \cdots, n)$ の処理時間は p_jとする．
- 各機械 M_jの利用時間には制約があり，$[0, d_j]$ $(j = 1, \cdots, m)$ の間利用可能であるとする．
- 各仕事は同時に複数の機械で処理することはできず，各機械は同時に複数の仕事を処理することはできない．

ここで，一般性を失わずに以下を仮定する．

$$p_1 \geq p_2 \geq \cdots \geq p_n \tag{6.1}$$

$$d_1 \geq d_2 \geq \cdots \geq d_m \tag{6.2}$$

実行可能スケジュールは以下のアルゴリズムによって求めることができる．このアルゴリズムは大きく分けて二つの部分で構成される．はじめにある時間において利用可能な機

械の集合に割り当てる仕事の集合$\mathcal{J}$ を決定する．このときそれぞれの仕事がどの機械に割り当てられるかということは決定されない．次に各時間において割り当てられる仕事の集合または機械の集合が変化するまでを一つの区間とし，その区間ごとに仕事の機械への割り当てを決定する．

アルゴリズム **PACK**

$t \leftarrow 0$

for $i = 1$ to n do

  $r_i \leftarrow p_i$

$j \leftarrow m$

$k \leftarrow 0$

while $t < d_1$ do

//区間 $[t_{k-1}, t_k]$ における機械集合に割り当てる仕事集合$\mathcal{J}$を決定する．

begin

  $t_k \leftarrow t$

  $k \leftarrow k + 1$

  $M_k = j$

  $\overline{\mathcal{J}_k} \leftarrow \{J_i \mid r_i = \max_J[r_j]\}$

  if $|\overline{\mathcal{J}_k}| < m$ then

    $\overline{\mathcal{J}_k}$に $\{J_i \mid r_i = \max_{J/\overline{\mathcal{J}_k}}[r_j]\}$ を加える．

  $r_{\min} = \min_{\overline{\mathcal{J}_k}}[r_j]\}$

  $r_{\max} = \max_{J/\overline{\mathcal{J}_k}}[r_j]\}$

  if $m(d_j - t)/|\overline{\mathcal{J}_k}| < (r_{\min} - r_{\max})$

  begin

    // ある機械の利用可能時間が終了する．

    $j \leftarrow j - 1$

    $t \leftarrow d_j$

  end else

  begin

    // $\overline{\mathcal{J}_k}$内の仕事の処理の最小の残り時間が

    // $J/\overline{\mathcal{J}_k}$における仕事の処理の最大の残り時間に等しくなる．

$t \leftarrow t + |\overline{\mathcal{J}}|(r_{\min} - r_{\max})/j$

end

end

for $l = 1$ to k do

begin

if $|\overline{\mathcal{J}_k}| = M_k$

$\overline{\mathcal{J}_k}$の各仕事をそれぞれ各機械に区間 $[t_{l-1}, t_l]$ の間割り当てる．

else

begin

// $|\overline{\mathcal{J}_k}| > M_k$のとき

区間 $[t_{l-1}, t_l]$ をさらに $|\overline{\mathcal{J}_k}|$ この区間に等分割する．

分割された区間において$\overline{\mathcal{J}_k}$のうち M_k個の仕事をそれぞれ

各機械に実行可能になるように割り当てる．

end

end

実行可能なスケジュールの存在性について以下の命題が成立する．

補題 6.1 以下の不等式が成立する時またその時に限り実行可能なスケジュールが存在する．

$$\sum_{i=1}^{j} d_i \geq \sum_{i=1}^{j} p_i, \quad j = 1, \cdots, m-1 \tag{6.3}$$

$$\sum_{i=1}^{m} d_i \geq \sum_{i=1}^{n} p_i \tag{6.4}$$

補題 6.1より，各機械の利用時間を式 (6.3),(6.4) を満たすように決定すれば PACK によって実行可能なスケジュールが構成できる．

6.3 機械の利用時間に制約がある機械使用コスト付きスケジューリング問題

本節では機械の利用時間に制約がある機械使用コスト付き等価並列機械スケジューリング問題について述べる.

- n 個の仕事 $J_1, \cdots, J_n$がある.
- m 台の仕事の処理に関して等価な機械 $M_1, \cdots, M_m$がある.
- 仕事 $J_i (i = 1, \cdots, n)$ の処理時間は p_iとする.
- 各機械の利用の際に単位時間当たり $w_j (j = 1, \cdots, m)$ のコストがかかるものとする.
- 一般性を失わずに以下を仮定する.

$$p_1 \geq p_2 \geq \cdots \geq p_n \tag{6.5}$$

$$w_1 \leq w_2 \leq \cdots \leq w_m \tag{6.6}$$

- Tを全ての仕事の処理時間の総和, すなわち

$$T = \sum_{i=1}^{n} p_i \tag{6.7}$$

  とする.

- 機械の利用時間には制約があり, 全ての機械は $[0, d]$ の間利用可能であるとする.
- 各機械の利用時間を $t_j (j = 1, \cdots, m)$ とすると機械の利用コストの総和 Cは

$$C = \sum_{j=1}^{m} w_j t_j \tag{6.8}$$

  と表わされる.

この問題は以下のように定式化される.

Problem I:

目的関数

$$C \to \text{minimize} \tag{6.9}$$

制約条件

$$\sum_{j=1}^{m} w_j t_j \leq C \tag{6.10}$$

$$t_j \leq d, \quad j = 1, \cdots, m \tag{6.11}$$

この問題は以下のアルゴリズム **MINCOST** によって最適解を得ることができる. このアルゴリズムではコストが安い機械から, 制約の上限まで使用することによってコストを最小とするスケジュールを与えている. ただし $\lceil x \rceil$ は x 以上の最小の整数とする.

アルゴリズム **MINCOST**

$k \leftarrow \lceil T/d \rceil$

$t_j \leftarrow d, \quad j = 1, \cdots, k-1$

$t_k \leftarrow T - (k-1)p_1$

$t_j \leftarrow 0, \quad j = k+1, \cdots, m$

PACK によって実行可能スケジュールを求める.

このときコストの総和 Cは以下のようになる. ただし $\lceil x \rceil$ は x よりも大きい最小の整数とする.

$$\begin{aligned} C &= \sum_{j=1}^{m} w_j t_j \\ &= \left(\sum_{j=1}^{\lceil \frac{T}{d} \rceil - 1} w_j d \right) + w_{\lceil \frac{T}{d} \rceil} \left(\sum_{i=1}^{n} p_i - \left(\left\lceil \frac{T}{d} \right\rceil - 1 \right) d \right) \\ &= \left(\sum_{j=1}^{\lceil \frac{T}{d} \rceil - 1} w_j - w_{\lceil \frac{T}{d} \rceil} \left(\left\lceil \frac{T}{d} \right\rceil - 1 \right) \right) d + w_{\lceil \frac{T}{d} \rceil} \sum_{i=1}^{n} p_i \end{aligned} \tag{6.12}$$

定理 6.2 MINCOST は Problem I の最適スケジュールを構築する.

証明

まずMINCOSTによって構築されるスケジュールが制約条件が満たされていることを示す.

使用可能時間 d の最小値は $\max\{\max_{1\leq j\leq n} t_i, \{\sum_{j=1}^{n} t_i\}/m\}$ である．次の 2 つの場合を考える.

1. $d = t_1$のとき,

$$
\begin{aligned}
t_1 &= p_1 && (6.13)\\
t_1 + t_2 &= 2p_1 > p_1 + p_2 \quad p_1 > p_2 > \cdots > p_n && (6.14)\\
&\vdots && (6.15)\\
\sum_{i=1}^{l} t_i &= lp_1 > \sum_{i=1}^{l} p_i && (6.16)\\
\sum_{i=1}^{k} t_i &= kp_1 \geq T = \sum_{i=1}^{n} p_i && (6.17)
\end{aligned}
$$

で成り立っている.

2. $d = T/m$ のとき,

$$
\begin{aligned}
t_1 &= T/m > p_1 && (6.18)\\
t_1 + t_2 &= 2T/m > t_1 + t_2 \quad p_1 > p_2 > \cdots > p_n && (6.19)\\
&\vdots && (6.20)\\
\sum_{i=1}^{l} t_i &= lT/m > \sum_{i=1}^{l} p_i && (6.21)
\end{aligned}
$$

この条件のときは $k = m$ が成立する．よって

$$\sum_{i=1}^{k} t_i = kT/m = T = \sum_{i=1}^{n} p_1 \qquad (6.22)$$

となり成立.

次に最適性を示す.

πを MINCOST によって構築されたスケジュールとし，コストを $C(\pi)$ とする．また最適なスケジュールπ'が存在し，そのコストが $C(\pi')$ であり，$C(\pi') < C(\pi)$ が成立するものと仮定する．またπ'における機械 M_iの使用可能時間を t'_iとする．

あきらかに

$$\sum_{1=i}^{m} t_i = \sum_{1=i}^{m} t'_i \tag{6.23}$$

が成立する．MINCOST において $i = 1, \cdots, k-1$ の機械 M_iについては制約の上限まで使用している．すなわちπにおいて $M_i = d(i = 1, \cdots, k)$ が成立する．

ここで2つの場合が考えられる．

1. $t'_i < t_i = d$ かつ $i < k$なる機械 M_iが存在する．このとき M_iを $t'_i < t_i$かつ $i < k$が成立する機械のうち最小のインデックスを持つ機械であるとする．式 (6.23) よりπにおいて M_iに $[t'_i, t_i]$ の時間として与えられる使用可能時間はπ'において $M_j(j > i)$ なる機械に与えられているはずである．この使用時間を機械 M_iに移すことによって $w_j > w_i$より $C(\pi')$ を減らすことができる．よってスケジュールπ'の最適性に矛盾する．
2. $t_k < t'_k$が成立する．このとき式 (6.23) よりπにおいて M_kに $[t'_k, t_k]$ の時間として与えられる使用可能時間はπ'において $M_j(j < k)$ なる機械は制約の上限まで使用されているので $M_j(j > k)$ なる機械に与えられているはずである．この使用時間を機械 M_kに移すことによって $w_j > w_k$より $C(\pi')$ を減らすことができる．よってスケジュールπ'の最適性に矛盾する．

よって示された.

証明終

6.4 コストに制約がある機械使用コスト付きスケジューリング問題

本節では機械の利用におけるコストの総和に制約がある機械使用コスト付き等価並列機械スケジューリング問題について述べる.

- n 個の仕事 $J_1, \cdots, J_n$がある.
- m 台の仕事の処理に関して等価な機械 $M_1, \cdots, M_m$がある.
- 仕事 $J_i(i = 1, \cdots, n)$ の処理時間は p_jとする.
- 各機械の利用の際に単位時間当たり $w_j(j = 1, \cdots, m)$ のコストがかかるものとする.
- 一般性を失わずに以下を仮定する.

$$p_1 \geq p_2 \geq \cdots \geq p_n$$

$$w_1 \leq w_2 \leq \cdots \leq w_m$$

- Tを全ての仕事の処理時間の総和，すなわち

$$T = \sum_{i=1}^{n} p_i \tag{6.24}$$

  とする.

- 機械の利用におけるコストの総和 Cには制約があり，$C \leq \mathcal{C}$ ($\mathcal{C}$: 定数) を満たすものとする.

このとき各機械の利用時間を $t_j(j = 1, \cdots, m)$ とするとこの問題は以下のように定式化される.

Problem II:

目的関数

$$d \rightarrow \text{minimize} \tag{6.25}$$

制約条件

$$\sum_{j=1}^{m} w_j t_j \leq \mathcal{C} \tag{6.26}$$

$$t_j \leq d, \quad j = 1, \cdots, m \tag{6.27}$$

この問題は以下のアルゴリズム **MINTIME** によって最適解を得ることができる．このアルゴリズムはまず最大完了時間が最小となるスケジュールを考え，このスケジュールにおいてコストが制約を超える場合に，コストが制約を満足するようにコストが安い機械から仕事の割り当てを増やすことによってコストの制約を満足し，かつ最大完了時間を最小とするスケジュールを得る.

アルゴリズム MINTIME

if $p_1 \geq T/m$ then

begin

$k \leftarrow \lceil T/m \rceil$

$m \leftarrow k$

$C' \leftarrow p_1 \sum_{j=1}^{k-1} w_j + w_k(T-(k-1)p_1)$

$t_j \leftarrow p_1, \quad j=1,\cdots,k-1$

$t_k \leftarrow T-(k-1)p_1$

$t_j \leftarrow 0, \quad j=k+1,\cdots,m$

end else

begin

$t_j \leftarrow T/m, \quad j=1,\cdots,m$

$C' \leftarrow T/m \sum_{j=1}^{m} w_j$

end

if $C' > C$ then

begin

$L \leftarrow (C'-C)/(w_m-(1/(M-1))\sum_{j=1}^{m-1} w_j)$

if $L \leq \delta$ then

begin

$t_j \leftarrow t_j + (L/(m-1)), \quad j=1,\cdots,m-1$

$t_m \leftarrow \delta - L$

end else

begin

$m \leftarrow m-1$

end

end

PACK によって実行可能スケジュールを求める．

定理 6.3 MINTIME は Problem II の最適スケジュールを構築する．

証明

まず MINTIME によって制約条件が満足されていることを示す．MINTIME によって実際に処理を行う機械，すなわち $t_j > 0$ である機械に対し $t_j \geq p_1$かつ $t_j \geq T/m$ が成立することは明らかである．これより補題を満たすので実行可能なスケジュールは存在する．

次に目的関数の最適性の証明をする．$t_1 = \max\{p_1, T/m\}$ のとき，総コストが利用可能コストの上限 C以下なら，$\max_j t_j$が最小となっているのは明らかである．利用可能コストの上限 Cより総コストが大きいならば，$t_1 = \max\{p_1, T/m\}$ となる制約を満たすスケジュールは存在しない．このとき明らかに一番コストがかかる機械 M_kから時間を減らして，この時間を M_1から M_{k-1}の機械に均等に分けることによって最大完了時間を最小にすることができる．

証明終

6.5 機械使用コスト付き 2 目的等価並列機械スケジューリング問題

前述の Problem I，II に対し機械利用におけるコストの総和と機械の利用時間の 2 目的のスケジューリング問題を考える．

- n 個の仕事 $J_1, \cdots, J_n$がある．
- m 台の仕事の処理に関して等価な機械 $M_1, \cdots, M_m$がある．
- 仕事 $J_i (i = 1, \cdots, n)$ の処理時間は p_jとする．
- 各機械の利用の際に単位時間当たり $w_j (j = 1, \cdots, m)$ のコストがかかるものとする．
- 機械の利用コストの総和 C，最大完了時間 d とし，この二つの目的関数を最小化する．

ただし，一般性を失わずに以下を仮定する．

$$p_1 \geq p_2 \geq \cdots \geq p_n$$

$$w_1 \leq w_2 \leq \cdots \leq w_m$$

また $T=\sum_{i=1}^{n} p_i$ とする.

このとき問題は以下のように定式化される.

Problem III:

目的関数

$$
\begin{aligned}
C &\rightarrow \text{minimize} && (6.28)\\
d &\rightarrow \text{minimize} && (6.29)
\end{aligned}
$$

制約条件

$$
\begin{aligned}
\sum_{j=1}^{m} w_j t_j &\leq C && (6.30)\\
t_j &\leq d, \quad j=1,\cdots,m && (6.31)
\end{aligned}
$$

6.3節述べた**MINCOST**の式(6.12)より，コストの総和 C，最大完了時間 d の以下の関係が成立する.

$$
\begin{aligned}
C &= w_{\lceil \frac{T}{d} \rceil} \sum_{i=1}^{n} p_i \\
&+ \left(\sum_{j=1}^{\lceil \frac{T}{d} \rceil - 1} w_j - w_{\lceil \frac{T}{d} \rceil} \left(\left\lceil \frac{T}{d} \right\rceil - 1 \right) \right) d \qquad (6.32)
\end{aligned}
$$

ここで

$$
\begin{aligned}
\alpha_k &= \left(\sum_{j=1}^{k-1} w_j - w_k (k-1) \right) \\
\beta_k &= w_k \sum_{i=1}^{n} p_i
\end{aligned}
$$

とすると，式(6.32)は以下の様に書くことができる.

$$
C = \beta_{\lceil \frac{T}{d} \rceil} + \alpha_{\lceil \frac{T}{d} \rceil} d \qquad (6.33)
$$

また

$$
\begin{aligned}
\alpha_{k+1} &- \alpha_k \\
&= \left(\sum_{j=1}^{k} w_j - w_{k+1} k \right) - \left(\sum_{j=1}^{k-1} w_j - w_k (k-1) \right)
\end{aligned}
$$

$$
\begin{aligned}
&= \left(\sum_{j=1}^{k} w_j - w_{k+1}k\right) - \left(\sum_{j=1}^{k} w_j - w_k k\right) \\
&= -k\left(w_{k+1} - w_k\right) \\
&\leq 0
\end{aligned}
$$

$$
\begin{aligned}
\beta_{k+1} - \beta_k &= w_{k+1}\sum_{i=1}^{n} p_i - w_k \sum_{i=1}^{n} p_i \\
&= (w_{k+1} - w_k)\sum_{i=1}^{n} p_i \\
&\geq 0
\end{aligned}
$$

となり，これより任意の $d_1, d_2 \quad (d_1 < d_2)$ に対し，$\left\lceil \frac{T}{d_1} \right\rceil \geq \left\lceil \frac{T}{d_1} \right\rceil$ より

$$
\begin{aligned}
\alpha_{\lceil \frac{T}{d_1} \rceil} &\leq \alpha_{\lceil \frac{T}{d_2} \rceil} \\
\beta_{\lceil \frac{T}{d_1} \rceil} &\geq \beta_{\lceil \frac{T}{d_2} \rceil}
\end{aligned}
$$

となる．また明らかに任意の d に対し

$$
\begin{aligned}
\alpha_{\lceil \frac{T}{d} \rceil} \leq 0 \\
\beta_{\lceil \frac{T}{d} \rceil} \geq 0
\end{aligned}
$$

が成立する．

また $d \geq T$ のとき，すべての仕事は最もコストの安い機械で処理されるので機械の利用におけるコストの総和は

$$
C = w_1 T \tag{6.34}
$$

となる．一方 $d < \frac{T}{m}$ のとき全ての機械を利用しても制約を満たすように仕事の処理を行うスケジュールは存在しない．

よって実行可能領域は式 (6.34) と式 (6.32) において d を T/m から T まで動かした直線によって囲まれた凸な領域となり，その境界が非劣解となる (図 6.1参照)．

実際のスケジュールを求めるためにはまず式 (6.32) より C, d の値を決定し，その値を用いて **MINCOST** によってスケジュールを構成すればよい．

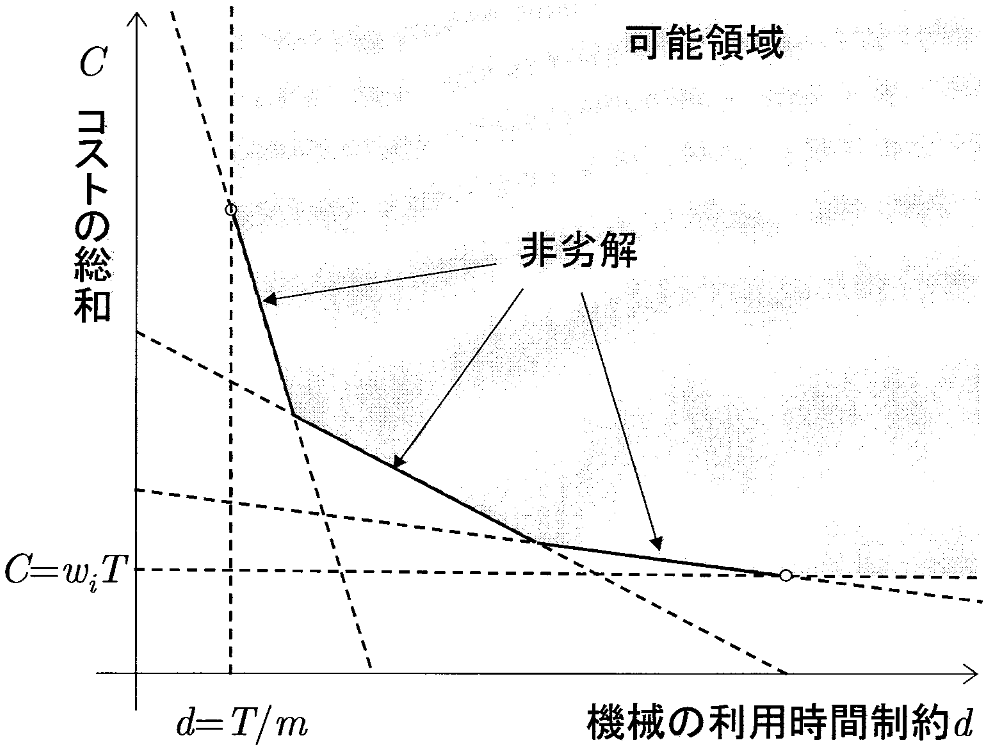

図 6.1　非劣解

6.6　結言

本章では等価並列機械において，従来の仕事の処理についてコストが定められている問題に対し新たに各機械の使用に対しその使用時間に依存したコストがかかるスケジューリング問題を考えた．この問題において総コストと最大完了時間をそれぞれを最小とするスケジュールを構築するためのアルゴリズムを与え，さらにこの問題における総コストと最大完了時間との関係式を求め，この関係式から総コストと最大完了時間という2つの目的関数の下でのスケジューリング問題における非劣解集合となることを示した．ここでのアルゴリズムは比較的単純なものであり，さらに制約を付加した問題へ拡張することが可能であると思われる．また実際のスケジュールを構築するアルゴリズムについても中断回数を減らすなどの改良が可能であると思われる．

第7章

結論

本論文では幾つかの多目的スケジューリング問題に対し数理的解析を行い，各問題について非劣スケジュールを与える方法を考察した．本章では本論文の各研究の総括を行い，またスケジューリング理論の今後の課題を述べる．

7.1 各章のまとめと今後の課題

第3章ではMPM環境においていくつかの特別な線形計画問題を定式化することによって，実際にスケジュールを求める方法を開発した．これはさらに様々な現場に対応したモデルを定式化する際において新たな制約を取りいれることを容易にすると考えられる．しかしFMS等において処理を中断することは容易ではなくスケジュールを構築する際に処理の中断を減らす，もしくは処理の中断にコストがかかるなどのペナルティを課すなどが必要であると思われる．また資源の処理を考慮した3目的の問題においてはその非劣スケジュールをどのように選択するかが大きな課題となる．

第4章では最大完了時間と最大納期遅れの二つの目的関数を最小化するような等価並列機械スケジューリング問題を取り上げた．まず納期制約がある等価並列機械問題において実行可能なスケジュールを求めるSahniのアルゴリズムをもとに新たなアルゴリズムを提案し，その各機械での各仕事の処理の完了時間を示す式と，実行可能なスケジュールが存在するための条件を導出した．この結果より最大納期遅れを示す式を導出し，かつ最大完了時間を考慮した修正納期を導入することによって実行可能であるための条件より二つの目的関数値間の関係を示す式を陽に導出した．同様の方法でさらに機械の台数が多い場

合も解析が可能であると思われる．これは今後考察すべき課題である．

第5章で取り上げた資源によって変化する処理時間を持つ等価並列機械問題では，前章での結果をもとに資源の割り当てに対する幾つかの定理を導き，その定理をもとに資源消費に伴うコストの総和を最小化するアルゴリズムを開発した．また最大納期遅れと資源の消費に伴うコストの総和という2目的の問題における2つの目的関数の最適値の関係を考察した．この結果は資源の総量が制約となる問題や資源の総量と最大完了時間といった多目的問題に拡張可能であると思われる．

第6章では等価並列機械において，各機械の使用する際にその使用時間に依存したコストがかかるスケジューリング問題を考え，総コストと最大完了時間をそれぞれ目的関数とする単一目的問題に対し最適なスケジュールを与えるアルゴリズムを提案した．またこの結果をもとにこの問題における総コストと最大完了時間との関係を示し，この関係式から総コストと最大完了時間という2つの目的関数の下での非劣スケジューリングを求める方法を開発した．ここでのアルゴリズムは比較的単純なものであり，さらに制約条件を付加した問題に対しても拡張が容易であると思われる．また実際のスケジュールを構築するアルゴリズムについても仕事の処理の中断の回数を更に減らすような改良が必要であると思われる．

以上の章を通して本論文では幾つかの多目的スケジューリング問題を考察し，それらの非劣スケジュールを与える方法を開発した．本論文で主に仕事の処理の完了時間に関係する複数の目的関数を取り上げており，これらの結果は現実のスケジュールを必要とする現場において各目的関数のバランスを取るための有益な情報を与えると期待される．特に第4章，第6章において二つの目的関数間の関係が式によって明示されており，これらは従来の多目的最適化問題には見られない画期的な成果であると考える．現場におけるスケジュールを考察する場合にはさらに多くの条件と目的関数を考える必要があるが，本論文の成果はより現実に近い多目的のスケジュールを数理的に考察するための端緒として大きな意義を持つものと考える

7.2　スケジューリング理論の今後の展開

第2章で述べたように急速に企業活動の多様化，情報化が行われることによって生産を管理を自動化することの重要性が増してきている．一方，企業活動の社会性，とりわけ環境に対する配慮などが問われるようになり，生産という活動に要求される事柄も多くなり

つつある．これらの期待に応えるためにスケジューリング理論も，従来の理論的かつ単一目的志向のものから，現実的かつ多様な要求に応えるようなものにならなければならない．

本論文では主に数理的解法を扱ってきたが，数理的解法は最適性の保証と論理の明快さという利点があるが，その計算の複雑さ，また多様な要求を取り入れることが困難であるという面がある．一方，近年コンピュータを利用した近似解法，特にメタ・ヒューリスティックスなどの解法の応用事例が数多く報告されているが，今後，製造業を取り巻く環境が厳しくなることが予想されるためさらに最適に近いスケジュールを組むことが求められる．そこでメタ・ヒューリスティックスなどの解法を大枠とし，その内部で個々の問題に対する数理的アプローチを用いるといった方法で現実の要請に応えうるスケジューリングの手法を生み出すことが急務であると思われる．

また本論文では多目的スケジューリング問題において非劣スケジュールの集合を求めることを目的としてきたが，非劣スケジュールの集合よりどの非劣スケジュールを選択すべきかという問題は未解決であり，企業活動全般の情報化，自動化において今後考えるべき重要な課題である．

謝　辞

本論文は大阪大学大学院工学研究科応用物理学専攻において筆者が在学中に行った多目的スケジューリングに関する研究の成果をまとめたものである．本研究を遂行し，まとめるにあたって終始懇切なご指導，ご助言をいただきました本学大学院工学研究科教授石井博昭先生に深く感謝の意を表すとともに，厚く御礼申し上げます．

本学大学院工学研究科教授 八木厚志先生，河田聡先生，同助教授 小松雅治先生，本学超伝導エレクトロニクス研究センター教授 萩行正憲先生には本論文作成にあたり細部にわたり御指導頂き，また貴重な御助言を頂きましたこと，心より感謝いたします．

神戸学院大学教授 塩出省吾先生，本学大学院講師 齋藤誠慈先生，流通科学大学講師 伊藤健先生には本研究の遂行，及び本論文の作成において，終始適切な御指導御助言をいただきましたことに心より感謝いたします．

本学大学院工学研究科 小出武氏，片桐英樹氏を始めとする石井研究室の皆様には貴重なご意見，及び激励の御言葉を頂きましたことに厚く御礼申し上げます．

最後になりましたが帝塚山大学教授 益田照雄先生には本研究への契機を与えてくださり，永きにわたり御指導御助言を頂きましたことに心よりお礼を申し上げます．

参考文献

[1] 阿部, 永倉, 落合, “知識ベース手法のアプローチによる自動生産計画システムの開発”, 生産スケジューリング・シンポジウム*'97* 講演論文集, (1997), 139-144.

[2] I. Adiri, O. Hamberg, “Openshop Scheduling under Linear Resources Constraints”, *Naval Research Logistics Quarterly*, 45, (1998), 51-66.

[3] C. A. Banna e Costa, L. Ensslin, E. C. Correa, J. C. Vansnick, “Decision Support Systems in Action : Integrated Application in a Multicriteria Decision Aid Process”, *European Journal of Operational Research*, 113, (1999), 315-335.

[4] S. P. Bansal, “Single Machine Scheduling to Minimize Weighted Sum of Completion Times with Secondary Criterion - A Branch and Bound Approach”, *European Journal of Operational Research*, 5, (1980), 177-181.

[5] S. K. Baruah, “The Multiprocessor Scheduling of Precedence Constrained Task Systems in the Presedence of Interprocessor Communication Delay”, *Operations Research*, 46, (1998), 65-72.

[6] M. Ben-Daya, M. Al-Fawzan, “A Tabu Search Approach for the Flow Shop Scheduling Problem”, *European Journal of Operational Research*, 109, (1998), 88-95.

[7] L. Bianco, J. Blazewicz, P. Dell'Olmo, M. Drozdowski, “Linear Algorithms for Preemptive Scheduling of Multiprocessor Tasks Subject to Minimal Lateness”, *Discrete Applied Mathematics*, 72, (1997), 25-46.

[8] J. Blazewicz, J. K. Lenstra, A. H. G. Rinnooy Kan, “Scheduling Subject to Resource Constraints: Classification and Complexity”, *Discrete Applied Mathematics*, 5, (1983), 11-24.

[9] J. Blazewicz, W. Cellary, R. Slowinski, J. Weglarz, *Scheduling under Resource Constraints - Deterministic Models*, J. C. Baltzer, 1986

[10] J. Blazewicz, W. Kubiak, S. Martello, "Algorithms for Minimizing Maximum Lateness with Unit Length Tasks and Resource Constraint", *Discrete Applied Mathematics*, 42, (1993), 123-138.

[11] J. Blazewicz, K. H. Ecker, G. Schimidt, J. Weglarz, *Scheduling in Computer and Manufacturing Systems*, Springer, 1993.

[12] J. Blazewicz, Z. Liu, "Scheduling Multiprocessor Tasks with Chain Constraints", *European Journal of Operational Research*, 94, (1996), 231-241.

[13] J. Blazewicz, K. H. Ecker, E. Pesch, G. Schimidt, J. Weglarz, *Scheduling in Computer and Manufacturing Processes*, Springer, 1996.

[14] J. D. Blocher, S. Chand, "A Forward Branch and Bound Algorithm and Forecast Horizon Results for the Changeover Scheduling Problem", *European Journal of Operational Research*, 91, (1996), 456-470.

[15] D. P. Bovet, M. Petrini, "Evaluation of Scheduling Algorithms for Resources with High Set-up Time", *European Journal of Operational Research*, 5, (1980), 182-192.

[16] P. Brandimarte, "Exploiting Process Plan Flexibility in Production Scheduling: A Multi-objective Approach", *European Journal of Operational Research*, 114, (1999), 59-71.

[17] P. Brucker, *Scheduling Algorithms*, Springer, 1995.

[18] P. Brucker, R. Schlie, "Job-shop Scheduling with Multi-purpose Machines", *Computing*, 45, (1990), 369-375.

[19] P. Brucker, B. Jurisch, B. Sievers, "A Branch and Bound Algorithm for the Job-shop Scheduling Problem", *Discrete Applied Mathematics*, 49, (1994), 107-127.

[20] P. Brucker, J. Hurink, B. Jurisch, B. Wostmann, "A Branch & Bound Algorithm for the Open-shop Problem", *Discrete Applied Mathematics*, 76, (1997), 43-59.

[21] P. Brucker, S. Knust, A. Schoo, O. Thiele, "A Branch & Bound Algorithm for the Resource-constrained Project Scheduling Problem", *Discrete Applied Mathematics*, 76, (1997), 43-59.

[22] J. A. Buzacott, D. D. YAO, "Flexible Manufacturing Systems: A Review of Analytical Models", *Management Science*, 32, (1986), 890-905.

[23] T. C. E. Cheng, Z. Chen, C. Li, B.M.-T. Lin, "Scheduling to Minimize the Total Compression and Late Costs", *Naval Research Logistics Quarterly*, 45, (1988), 67-82.

[24] E. G. Coffman Jr., *Computer and Job Shop Scheduling*, John Wiley & Sons 1976.

[25] S. A. Cook, "The Complexity of Theorem-proving Procedures", *Proceedings of 3rd Annual ACM Symposium on Theory of Computing*,(1971), 151-158.

[26] M. Drozdowski, "Scheduling Multiprocessor Tasks - An Overview", *European Journal of Operational Research*, (1996), 215-230.

[27] L. F. Gelders, P. R. Kleindoefer, "Coordinating Aggregate and Detailed Scheduling in a One Machine Job Shop Part I", *Theory of Operations Research*, 22, (1974), 46-60.

[28] L. F. Gelders, P. R. Kleindoefer, "Coordinating Aggregate and Detailed Scheduling in a One Machine Job Shop Part II", *Theory of Operations Research*, 23, (1975), 312-324.

[29] F. Glover, "Tabu Search I", *ORSA Journal on Computing*, 1, (1989), 190-206.

[30] F. Glover, "Tabu Search II" *ORSA Journal on Computing*, 2, (1989), 4-32.

[31] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

[32] R. E. Graham, E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, "Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey", *Annals of Discrete Mathematics*, 4, (1979), 287-326.

[33] A. Gunasekran, T. Martikainen, P. Yli-Olli, "Flexible Manufacturing Systems: An Investigation for Research and Applications", *European Journal of Operational Research*, 66, (1993), 1-26.

[34] H. Heck, S. Roberts, "A Note on the Extension of a Result on Scheduling with Secondary Criteria", *Naval Research Logistics Quarterly*, 19,(1972), 403-405.

[35] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.

[36] J. A. Hoogeveen, S. L. van de Velde, B. Veltman, "Complexity of Scheduling Multiprocessor Tasks with Prespecified Processor Allocations", *Discrete Applied Mathematics*, 17, (1994), 205-208.

[37] J. A. Hoogeveen, S. L. van de Velde, "Minimizing Total Completion Time and Maximum Cost Simultaneously is Solvable in Polynomial Time", *Journal of Algorithms*, 21, (1996), 415-433.

[38] J. A. Hoogeveen, "Minimizing Maximum Promptness and Maximum Lateness on a Single Machine", *Mathematics of Operations Research*, 21, (1996), 100-114.

[39] 堀, "生産スケジューリング・システム構築のためのオブジェクト指向フレームワーク", 生産スケジューリング・シンポジウム *'96* 講演論文集, (1996), 247-252.

[40] W. A. Horn, "Some Simple Scheduling Algorithms", *Naval Research Logistics Quarterly*, 21, (1974), 177-185.

[41] 伊倉, 名原, "石油精製工程スケジューリングへの数理計画法の応用", オペレーションズ・リサーチ, 40, (1995), 255-260.

[42] H. Ishii, "Multiobjective Scheduling Problems", *Lecture Notes in Economics and Mathematical Systems*, 405 (1992), 386-391.

[43] H.Ishii, M.Tada, T.Nishida, "Bi-criteria Scheduling Problem on Uniform Processors", *Mathematica Japonica*, 35, (1990), 515-519

[44] J. R. Jackson, "Scheduling a Production Line to Minimize Maximum Tardiness", *Research Report*, 43, (1995), Management Science Research Project, University of California at Los Angels.

[45] A. Janiak, "Single Machine Scheduling Problem with a Common Deadline and Resource Dependent Release Dates", *European Journal of Operational Research*, 53, (1991), 317-325.

[46] A. Janiak, "Single Machine Sequencing with Linear Models of Release Dates", *Naval Research Logistics Quarterly*, 45, (1998) 99-113.

[47] A. Janiak, M. Y. Kovalov, "Single Machine Scheduling Subject to Deadlines and Resource Dependent Processing Times", *European Journal of Operational Research*, 94, (1996), 284-291.

[48] S. M. Johnson, "Optimal Two- and Three-stage Production Schedules", *Naval Research Logistic Quarterly*, 1, (1954), 61-68.

[49] J. Jozefowska, J. Weglarz, "Discrete-continuous Scheduling Problems - Mean Completion Time Results", *European Journal of Operational Research*, 94, (1996), 302-309.

[50] B. Jurisch, "Lower Bounds for the Job-shop Scheduling Problem on Multi-purpose Machines", *Discrete Applied Mathematics*, 58, (1995), 145-156.

[51] B. Jurisch, W. Kubiak, "Two-machine Open Shops with Renewable Resources", *Operations Research*, 45, (1995), 544-552.

[52] R. M. Karp, "Reducibility among Combinatorial Problems", in R. E. Miller, J. W. Thatcher(eds.), *Complexity of Computer Computations*, Plenum Press, 1972.

[53] S. Kirkpatrick, C. D. Gellat, M. P. Vecchi, "Optimization by Simulated Annealing", *Science*, 220, (1983), 671-680.

[54] F. Kolahan, M. Liang, "An Adaptive TS Approach to JIT Sequencing with Variable Processing Times and Sequence-dependent Setups", *European Journal of Operational Research*, 109, (1998), 142-159.

[55] P. Korhonen, H. Moskowitz, Jyrki Wallesnius, "Multiple Criteria Decision Support - A Review", *European Journal of Operational Research*, 63, (1992),361-375.

[56] S. A. Kravchenko, "A Polynomial Algorithm for a Two-machine No-wait Job-shop Scheduling Problem", *European Journal of Operational Research*, 106, (1998), 101-107.

[57] M. Kubale, "Preemptive versus Nonpreemptive Scheduling of Biprocessor Tasks on Dedicated Processors", *European Journal of Operational Research*, 94, (1996), 242-251.

[58] W. Kubiak, V. Timkovsky, "Total Completion Time Minimization in Two-machine Job Shops with Unit-time Operations", *European Journal of Operational Research*, 94, (1996), 310-320.

[59] 黒田 "生産スケジューリングの現状と研究動向", 日本ファジィ学会誌, 8, (1996), 784-794.

[60] 黒田, 田部, 園川, 中根, 生産管理, 1989, 朝倉書店.

[61] P. J. M. V. Laarhoven, E. H. L. Aarts, J. K. Lenstra, "Job Shop Scheduling by Simulated Annealing", *Operations Research*, 40, (1992), 113-125.

[62] J. Labertoulle, E. L. Lawler, J. K. Lenstra, A. H. G. Rinnoy Kan, "Preemptive Scheduling of Uniform Machines Subject to Release Dates", in H. R. Pulleyblank(eds.), *Progress in Combinatorial Optimization*, Academic Press, 1984.

[63] E. L. Lawler, "Optimal Sequence of a Single Machine Subject to Precedence Constraints", *Management Science*, 19, (1973), 544-546.

[64] E. L. Lawler, and J. Labertoulle, "On Preemptive Scheduling Problem on Uniform Processors by Linear Programming", *Journal of the Association for Computing Machinery*, 25, (1978), 612-619.

[65] E. L. Lawler, M. G. Luby and V. V. Vazirani, "Scheduling Open Shop with Parallel Machines", *Operations Research Letters*, 1, (1982), 161-164 .

[66] C. Li, E. C. Sewell, T. C. E. Cheng, "Scheduling to Minimize Release-Time Resource Consumption and Tardiness Penalties", *Navel Research Logistics Quarterly*, 42, (1995), 929-966.

[67] S. Lim, Y. Kim, "Capacity Planning for Implementation of Flexible Manufacturing Systems under Budget Restrictions", *European Journal of Operational Research*, 104, (1998), 175-186.

[68] T. Masuda, H. Ishii, T. Nishida, "Scheduling with Machine Available Time Constraints", *TECHNOLOGY REPORTS OF THE OSAKA UNIVERSITY*, 31, No. 1583, (1981).

[69] T. Masuda, H. Ishii, "Two Machine Open Shop Scheduling Problem with Bi-Criteria", *Discrete Applied Mathematics*, 52 (1994), 253-259.

[70] G. B. McMahon, C. Lim, "The Two-machine Flow Shop Problem with Arbitrary Precedence Relations", *European Journal of Operational Research*, 64, (1993), 249-257.

[71] S. T. McCornick, M. L. Pinedo, "Scheduling n Independent Jobs on m Uniform Machines with both Flowtime and Makespan Objectives: A parametric Analysis", *ORSA Journal on Computing*, 7, (1995), 63-77.

[72] R. McNaughton, "Scheduling with Deadlines and Loss Functions", *Management Science*, 6, (1959), 1-12.

[73] S. Miyazaki, "One Machine Scheduling Problem with Dual Criteria", *Journal of Operations Research Society of Japan*, 24, (1981), 37-50.

[74] Z. M. Mohamed, "A Flexible Approach to (Re)configure Flexible Manufacturing Cells", *European Journal of Operational Research*, 95, (1996), 566-576.

[75] Z. M. Mohamed, J. J. Bernardo, "Tool Planning Models for Flexible Manufacturing Systems", *European Journal of Operational Research*, 103, (1997), 497-514.

[76] 門田, 新トヨタシステム, 1991, 講談社.

[77] E. L. Mooney, R. L. Rardin, "Tabu Search for Class of Scheduling Problems", *Annals of Operations Research*, 41, (1993), 253-278.

[78] 永井, 田村, 中川, 谷崎, 中島, "分散環境によるマンマシン協調型スケジューリングシステムの構築", オペレーションズ・リサーチ, 40, (1995), 261-267.

[79] V. N. R. Neppali, C. Chen, J. N. D. Gupta, "Genetic Algorithms for Two-stage Bicriteria Flowshop Problem", *European Journal of Operational Research*, 95, (1996), 356-373.

[80] E. Nowicki, S. A. Zdrzalka, "A Bicriterion Approach to Preemptive Scheduling of Parallel Machines with Controllable Job Processing Time", *Discrete Applied Mathematics*, 63, (1995), 237-256.

[81] C. Rajendran, "Heuristics for Scheduling in Flowshop with Multiple Objectives", *European Journal of Operational Research*, 82, (1995), 540-555.

[82] R. Z. Rios-Mercado, J. F. Bard, "Heuristics for Flow Line Problem with Setup Costs", *European Journal of Operational Research*, 110, (1998), 76-98.

[83] S. Sahni, "Preemptive Scheduling with Due Dates", *Operations Research*, 27,(1979) 925-934.

[84] S. Sayin, S. Karabati, "A Bicriteria Approach to the Two-machine Flow Shop Scheduling Problem", *European Journal of Operational Research*, 113, (1999), 435-449.

[85] F. Sivrikaya-Serifoglu, G. Ulsoy, "A Branch and Bound Algorithm for the Resource-constrained Project Scheduling Problem", *European Journal of Operational Research*, 107, (1998), 272-288.

[86] W. E. Smith, "Various Optimizers for Single-Stage Production", *Naval Research Logistics Quarterly*, 3, (1956), 56-66.

[87] W. A. Stadler, "A Survey of Multicriteria Optimization or the Vector Maximization Problem, I: 1776-1960", *Journal of Optimization Theory and Application*, 69, (1979), 1-52.

[88] V. A. Strusevich, "Two Machine Flow Shop Scheduling Problem with No Wait Process: Controllable Machine Speeds", *Discrete Applied Mathematics*, 59, (1995), 75-86.

[89] 玉置, 森, 荒木, 三島, 大貝, 黒田, "遺伝的アルゴリズムに基づく圧延工程スケジューリング問題の解法", 生産スケジューリングシンポジウム, (1993), 39-44.

[90] 玉置, 田口, 荒木, "メタ戦略によるプラスチック成型工程スケジューリング", 生産スケジューリングシンポジウム'94 講演論文集, (1994), 182-187.

[91] 玉置, 西野, 阿部, "非正規目的関数を含む多目的並列機械型スケジューリング問題に対する遺伝アルゴリズムの構成手法", 生産スケジューリングシンポジウム'98 講演論文集, (1998), 203-208.

[92] G. Vairaktarakis, S. Sahni, "Dual Criteria Preemptive Open-Shop Problems with Minimum Makespan", *Naval Research Logistics Quarterly*, 42, (1995), 103-121.

[93] A. J. Vakharia, B. Catay, "Two Machine Openshop Scheduling with Machine-dependent Processing Times", *Discrete Applied Mathematics*, 73, (1997), 283-288.

[94] R. G. Vickson, "Choosing the Job Sequence and Processing Time to Minimize Total Processing Plus Flow Cost on a Single Machine", *Operations Research*, 28, (1980), 1155-1167.

[95] R. G. Vickson, "Two Single Machine Sequencing Problems Involving Controllable Job Processing Times", *AIIE Transactions*, 2, (1980), 258-262.

[96] C. Wang, C. Chu, J. Proth, "Heuristic Approaches for $n/m/F/\sum C_i$ Scheduling Problems", *European Journal of Operations Research*, 96, (1997), 636-644.

[97] L. N. V. Wassenhove, L. F. Gelder, "Solving a Bi-Criteria Scheduling Problem", *European Journal of Operations Research*, 4, (1980), 42-48.

[98] L. N. V. Wassenhove, K. R. Baker, "A Bicriterion Approach to Time Cost Trade-Offs in Sequencing", *European Journal of Operations Research*, 11, (1982), 48-52.

[99] T. J. Williams, *Analysis and Design of Hierarchical Control Systems: With Special Reference to Steel Plant Operations*, North-Holland, 1986

著者発表論文

(1) Shintaro Mohri, Teruo Masuda, Hiroaki Ishii, “Bi-Criteria Scheduling Problem on 3 Identical Parallel Machines”, *International Journal of Production Economics*, 60, (1999), 526-536.

(2) Shintaro Mohri, Hiroaki Ishii, Teruo Masuda, “Bi-criteria MPM Open Shop Scheduling Problem”, *MATHEMATICA JAPONICA*, 49, (1999), 411-416.

(3) Shintaro Mohri, Teruo Masuda, Hiroaki Ishii, “Parallel Machines Scheduling with Resource Dependent Processing Times” *Journal of Operations Research Society of Japan*, (投稿中).

(4) Shintaro Mohri, Teruo Masuda, Hiroaki Ishii, “Bi-criteria Scheduling Problem with a Cost Depending on Machine Working Time” Journal of Scheduling, (投稿中).