

Title	スケジューリング問題の解法に関する研究
Author(s)	川口, 剛
Citation	大阪大学, 1981, 博士論文
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/1057">https://hdl.handle.net/11094/1057</a>
rights	
Note	

*Osaka University Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

Osaka University

スケジューリング問題の解法  
に関する研究

昭和 55年 12月

川 口 剛

## 内 容 梗 概

本論文は、著者が大阪大学工学研究科（通信工学専攻）在学中に中西研究室において行った研究のうち、スケジューリング問題の解法に関する研究をまとめたものである。

第1章では、本研究の意義及び概要を述べるとともに、解法の良さを測る尺度として時間計算量及び近似解の最悪誤差を定義し、スケジューリング問題に対する解法をこれらの尺度を用いて評価した従来の結果を概説している。

第2章では、リソース条件をもたない一般的なスケジューリング問題に対する厳密解法を提案し、この解法が既存の解法と比較して計算効率の面で優れていることを実験的に確かめている。

第3章では、重み付き平均滞留時間最小化問題に対する重み／処理時間 非増加順リストスケジューリングに1つの操作を加えた解法を評価し、プロセッサの平均終了時刻を越える処理時間をもつジョブが存在しないとき、この解法の最悪誤差の上限値が  $\frac{1}{4}$  で与えられることを示し、この解法の良さを理論的に裏付けている。

第4章では、最悪の場合の評価から良さが保証される近似解法が知られていない問題の中で、多数のジョブが複数のリソースを同時に要求するとき、同時に処理できるジョブの数を最大にする問題に対する1つのヒューリスティックな解法を確率的に評価し、この解法の良さを理論的に確かめている。

第5章では、本研究で得られた結果をまとめるとともに、残された問題点について記述している。

以上の各章を構成している研究内容は、すべて、電子通信学会論文誌、電子通信学会回路とシステム研究会、電子通信学会全国大会において発表されたものである。

## 関 連 発 表 論 文

- (1) 川口, 中野, 中西, “重み付き平均滞留時間最小化スケジューリング問題の解法”, 信学技報, *CST* 78-138 (1979-1)
- (2) 川口, 中野, 中西, “重み付き平均滞留時間最小化スケジューリング問題” 信学論 *D*, 62-D, 6, *p.* 363 (1979-6)
- (3) 川口, 中野, 中西, “あるスケジューリング法の最悪の場合の評価”, 信学技報, *CST* 79-121 (1979-11)
- (4) 川口, 中野, 中西, “あるスケジューリング法の最悪の場合の評価”, 信学論 *D*, 63-D, 3, *p.* 225 (1980-3)
- (5) 川口, 中野, 中西, 新見, “*WMFT* 問題に対する近似解法”, 信学会総合全国大会, 32 (1980-3)
- (6) 川口, 中野, 中西, 竹山, “節点独立集合問題に対する近似解法の高率的评价”, 信学技法, *CAS* 80-45 (1980-7)
- (7) 川口, 中野, 中西, 竹山, “節点独立集合問題に対する近似解法の高率的评价(2)”, 信学技法, *CAS* 80-97, (1980-11)

# 目 次

第1章 緒 論	1
1.1 スケジューリング問題	1
1.2 解法の時間計算量	5
1.2.1 時間計算量と領域計算量	5
1.2.2 $NP$ 完全問題	6
1.2.3 スケジューリング問題に対する解法の時間計算量	9
1.3 近似解法の最悪の場合の評価	12
1.3.1 近似解法の最悪誤差	12
1.3.2 スケジューリング問題に対する近似解法の最悪の場合の評価	13
第2章 リソース条件をもたない問題に対する厳密解法	16
2.1 緒 言	16
2.2 諸 定 義	16
2.3 整数計画問題への定式化	18
2.4 解 法	26
2.4.1 ラグランジュ乗数法の適用	26
2.4.2 山登り法	27
2.4.3 分枝限定法	31
2.5 重み付き平均滞留時間最小化問題に適用した実験結果	33
2.5.1 実験結果	33
2.5.2 Fisherによる解法との比較	35
2.6 結 言	36
第3章 重み付き平均滞留時間最小化問題に対するリストスケジューリングの最悪の場合の評価	37

3.1	緒言	37
3.2	各ジョブの単位時間当りのコスト対処理時間比がすべて等しい場合	38
3.2.1	任意のスケジューリング	38
3.2.2	処理時間による優先順序を用いたスケジューリング	47
3.3	各ジョブの単位時間当りのコスト対処理時間比が異なる場合	50
3.4	結言	56
第4章	リソース条件をもつ問題に対する近似解法の評価	59
4.1	緒言	59
4.2	最悪の場合の評価による従来の結果	59
4.3	節点独立集合問題に対する近似解法の確率的評価	62
4.3.1	問題の定式化と近似解法	62
4.3.2	平均線度一定モデル	64
4.3.3	解法Ⅱによる解の値	70
4.3.4	解法Ⅰと解法Ⅱの比較	80
4.3.5	スケジューリング問題との関係	83
4.4	結言	85
第5章	結論	87
	謝辞	90
	参考文献	91

## 諸 記 号

本論文を通して用いる記号を次のように定義する。

$\equiv$	: 定義によって等しい
$\max(a, \beta)$	: $a, \beta$ のうち大きい方の値
$\max_{k \in J} a_k$ $\max\{a_k \mid k \in J\}$	} : 集合 $J$ に属す要素 $k$ についての $a_k$ の最大値
$\max a_k$	: 集合 $J$ が明らかな場合, 集合 $J$ に属す要素 $k$ についての $a_k$ の最大値
$\min(a, \beta)$	: $a, \beta$ のうち小さい方の値
$\min_{k \in J} a_k$	: 集合 $J$ に属す要素 $k$ についての $a_k$ の最小値
$\sum_{k=1}^n a_k$	: $a_1 \cdots a_n$ の総和
$\sum_{k \in J} a_k$	: 集合 $J$ に属す要素 $k$ についての $a_k$ の総和
$\sum a_k$	: 集合 $J$ が明らかな場合, 集合 $J$ に属す要素 $k$ についての $a_k$ の総和
$ J $	: 集合 $J$ に属す要素の数
$\lfloor a \rfloor$	: $a$ 以下の最大の整数
$\lceil a \rceil$	: $a$ 以上の最小の整数
$k \in J$	: $k$ が集合 $J$ に属すことを表わす
$k \notin J$	: $k$ が集合 $J$ に属さないことを表わす
$\forall k \in J$	: $k$ が集合 $J$ に属す任意の要素であることを表わす
$\phi$	: 空集合
$\bigcup_{k=1}^n J_k$	: 集合 $J_1 \cdots J_n$ の和集合
$J - I$	: $J, I$ が集合であるとき, $J$ と $I$ の差集合を表わす
$\log a$	: 2 を底とする対数
$\ln a$	: 自然対数
$\ \tilde{a}\ $	: ベクトル $\tilde{a}$ の 2 乗ノルム

上記以外の記号については, その記号を用いる時点で定義する。

# 第 1 章 緒 論

## 1.1 スケジューリング問題

プロジェクトの日程計画，工場における作業計画，データ処理システムでのジョブ処理計画などに見られるように，ジョブの順序付けやリソースの割当ては，ほとんどすべての生産システム，サービスシステムにおいて出くわす課題であり，ふるくからスケジューリング問題として考究されてきている。

スケジューリング問題は，以下に示す，問題情報を定める変数，スケジュールを表現する変数，変数の間の関係を記述する制約条件及びスケジュールの良さを表わす評価関数によって規定される。

### 1) 問題情報を定める変数

ジョブの到着時刻，処理時間，重み，納期，リソース要求量，  
各時刻に遊休状態にあるプロセッサ<sup>(注1)</sup>の数，  
各時刻に利用できるリソース<sup>(注2)</sup>の種類，数，容量など。

### 2) スケジュールを表現する変数

ジョブの開始時刻，完了時刻，各時刻に処理されるジョブ集合など。

### 3) 制約条件

ジョブの間の先行関係，プロセッサ条件<sup>(注3)</sup>，リソース条件<sup>(注4)</sup>，ジョブの  
先取権<sup>(注5)</sup>など。

### 4) 評価関数

最大完了時刻，最大納期遅れ，（重み付き）平均滞留時間，（重み付き）  
平均納期遅れ，（重み付き）納期遅れジョブ数など。

---

(注1) ジョブを処理する装置であり，各時刻において1つのジョブは1台のプロセッサを専有することが仮定される。

(注2) ジョブを処理するために必要なプロセッサ以外の付加的リソース。

(注3) 各時刻に処理されるジョブの数は，プロセッサの数を越えないという条件。

(注4) 各時刻に処理されるジョブのリソース要求量の総和がリソース容量を越えないという条件。

(注5) 処理を開始されたジョブが完了するまで中断できないとき，ジョブは先取権をもつという。一方，他のジョブの割込みを許すとき，先取権をもたないという。



従って、これらの問題要素をどのように規定するかによって、スケジューリング問題は種々の形に定式化される。ある形のもの(比較的簡単な場合)については、既に、最適スケジュールを見い出す解法が知られているが、やや複雑な形になると、最適スケジュールを求めることが極めて困難な問題となり、多くの考究課題をもっている。

まず、スケジューリング問題は、ジョブの到着のしかたによって、静的問題と動的問題に分類される<sup>(1)</sup>。前者は一定個数のジョブが同時又は既知の到着時刻にシステムに到着することを仮定する問題であり、後者は無限の時間にわたってジョブが断続的に到着し、しかも到着時刻については統計的な情報しか得られない場合を仮定する問題である。

プロジェクトの日程計画、工場における作業計画、バッチ処理システムでのジョブ処理計画など、その定式化によって静的問題に帰着される例は少なくない。又、動的問題において新たなジョブ集合が到着するまでの1区間のモデルとして静的問題を考え、各区間で良いスケジュールを得ることによって、全体として良いスケジュールを得ることが期待できる。例えば、1台のプロセッサで連続して到着するジョブを処理する場合、新たなジョブの到着時点でシステム内のジョブに最小残り処理時間規則を適用することによって、平均滞留時間を最小にするスケジュールが得られる<sup>(1)</sup>。

本研究は、ジョブの間に先行関係が存在しない静的スケジューリング問題に関する研究であり、一般的な静的問題に対する厳密解法を提案するとともに、2つの静的問題に対するヒューリスティックな解法を理論的に評価し、その解法の良さを確かめる。

静的スケジューリング問題に対する解法の研究は、従来、次の2つの立場から行われてきた。

- (i) 厳密解法又は近似解法の考案。
- (ii) 解法の良さの理論的評価——解法が問題を解くのに要す時間、及び、近似解の最適解に対する誤差の理論的評価。

特に(ii)は問題の規模や投資される資本に応じた解法を選択するために重要で

あり、限られた実験結果に基づき評価と比較して、解法の良さを保証する立場から、意志決定者やスケジューリング手順以後の計画作成者に対して強い説得力を与えることになる<sup>(2)</sup>。(ii)に関する研究から、静的スケジューリング問題は、多項式時間計算量をもつ解法が現在までに見つかっている問題とそうでない問題に分類され、後者の多くは  $NP$  完全問題と呼ばれる問題集合に属することが知られている。 $NP$  完全な問題に対する厳密解法としては、動的計画法や分枝限定法を用いた解法があり<sup>(3)</sup>、小規模な問題に対して最適解を得るための有効な解法となることが実験的に確かめられている<sup>(注1)</sup>。

第2章では、リソース条件をもたない一般的なスケジューリング問題に対する厳密解法を考察している。まず、問題を0-1整数計画問題に定式化し、この定式化の下でラグランジュ関数の値を最適解の値に近づける山登り法を、次いで、山登り法によって得られる値を下限として利用する分枝限定法を提案する。最後に、この解法を実験的に調べ、この解法が既存の解法<sup>(4)</sup>と比較して計算効率の面で優れていることを確かめている。

問題の規模が大きくなると、厳密解を求めることは困難であり、近似解法に依らざるを得ない。通常よく用いられる近似解法にリストスケジューリング法がある<sup>(3)</sup>。この解法は、ジョブを規定する変数値をもとに各ジョブに優先順位を与え、各時刻でプロセッサが遊休状態になり次第、最も高い優先順位をもつジョブから順にプロセッサに割当てて行く解法である。静的問題に対するリストスケジューリング法を理論的に評価する試みもなされてきているが、主に最大完了時刻を評価関数とする問題に対して行われているに過ぎない。

第3章では、異なる重み<sup>(注2)</sup>をもつジョブ集合を $m$  ( $\geq 2$ ) 台のプロセッサで処理する、重み付き平均滞留時間最小化問題に対して通常よく用いられる重み/処理時間 非増加順リストスケジューリング<sup>(5)</sup>を最悪の場合について評価して

---

(注1)  $NP$  完全な問題に対する厳密解法の理論的評価からは絶望的な結果しか得られておらず、この種の解法に対しては、残念ながら、実験的評価法を用いざるを得ない。

(注2) ジョブがシステム内に滞留することによってシステムが受ける負荷、又は、需要者が被るコスト。

いる。又、この解法の時間計算量に影響を与えない1つの操作を加えたリストスケジューリングを提案し、最悪の場合を理論的に評価し、この解法の良さを理論的に裏付けている。重み付き平均滞留時間は、重み付き待ち時間、重み付き納期ずれ、及び、各時刻にシステムに存在する重み付きジョブ数の平均値と等価な評価関数である<sup>(3)</sup>。従って、第3章で提案する解法は、システムの負荷を軽減し、需要者へのサービス効率を高めるという両面において有効な解法といえる。

スケジューリング問題には、既存の近似解法に対する最悪の場合の評価からは、絶望的な結果しか得られない問題も少なくない。リソース条件をもつほとんどのスケジューリング問題は、この種の問題となることが知られている<sup>(3)</sup>。この種の問題に対しては、問題情報を定める変数が任意の値をもつ場合に有効な近似解法を求めることは難かしいが、変数がある分布に従う場合に有効な近似解法を求めることは可能であるかもしれない。変数が統計的情報をもつ場合を仮定する近似解法の評価は、最悪の場合の評価に対して、確率的评价と呼ばれており<sup>(6)</sup>、実際の分布を仮定して得られる結果は、解法の良さを保証する立場からも重要である。

第4章では、多数のジョブが複数のリソースを同時に要求するとき、同時に処理できるジョブの数を最大にする問題をグラフの節点独立集合問題に定式化し、大規模な問題に対して通常よく用いられるヒューリスティックな解法<sup>(7)</sup>を、ランダムグラフを用いて評価し、この解法の良さを理論的に確かめている。この問題の解は、リソース条件をもつ一般的なスケジューリング問題<sup>(8)</sup>、プログラム並列処理における変数割当て問題<sup>(9)</sup>、診断時刻表作成問題<sup>(10)</sup>などにおいて応用される。この問題のランダムグラフを用いた解析は、ジョブの間の競合（リソースの奪合い）が、それぞれ独立に等しい確率で生じる場合に相当し、第4章で得られる結果は、同じ問題要素をもつ動的問題を解析する上からも重要である。

以上が、本研究の意義及び概要である。本研究の詳細を述べる前に、1.2節及び1.3節では、静的スケジューリング問題に対する解法の良さを測る尺

度を定義し、この尺度を用いた従来の結果を概説する。

## 1.2 解法の時間計算量

問題を電子計算機を用いて解くとき、その問題を解く解法が必要とする演算時間及びメモリスペースの理論値は、各々、時間計算量及び領域計算量と呼ばれる<sup>(11)</sup>。スケジューリング問題を含めた多くの組合せ問題は、問題の規模の多項式で表現される時間計算量をもつ解法が現在までに見つかっている問題とそうでない問題に分類され、後者の多くは  $NP$  完全問題と呼ばれる問題集合に属することが知られている<sup>(11),(12)</sup>。

本節では、まず、文献(11)に従って、時間計算量、領域計算量及び  $NP$  完全問題を定義し、次いで、各ジョブが先行関係をもたないスケジューリング問題について現在までに得られている解法の時間計算量を示す。

### 1.2.1 時間計算量と領域計算量

通常の電子計算機の理想的モデルとして、ランダムアクセス機械  $RAM$  がある<sup>(11)</sup>。 $RAM$  は1つの累算機、読み取り専用入力テープ、書き込み専用出力テープ、メモリ(レジスタ)ならびにプログラムで構成され、 $RAM$  に対する命令語としては、通常の電子計算機に見られる機械語命令と同種の命令語が想定される。又、問題を  $RAM$  の入力テープ上に与えられる入力データの集合と考え、1つの入力データに対する入力テープの長さを問題のサイズ<sup>(注)</sup>と呼ぶ。

〔定義1.1〕<sup>(11)</sup> 解法  $A$  の時間計算量 (time complexity) 及び領域計算量 (space complexity) とは、 $A$  を  $RAM$  で実行した際、その  $RAM$  プログラムが必要とする演算時間  $t_A$  及びメモリスペース  $s_A$  を問題のサイズ  $n$  の関数として表現したものであり、各々、 $t_A(n)$  及び  $s_A(n)$  で表わす。

〔定義1.2〕<sup>(11)</sup> 問題のサイズが  $n$  であるすべての入力データに対する時間計算量及び領域計算量の最大値を、各々、最大時間計算量 (worst case time complexity) 及び最大領域計算量 (worst case space complexity) と

---

(注) 問題のサイズの与え方としては、別の方法もあり、これについては後に述べる。

呼ぶ<sup>(注1)</sup>。

一方、解法を RAM プログラムで記述することは極めて手間のかかる作業であり、又、RAM プログラムにどのような命令語を準備するかによって解法の効率が影響を受けることになる。このことを解決するために、通常、ALGOL 60 に準じた文法をもつ片言アルゴルと呼ばれる高水準な言語を用いて解法が記述される。又、問題のサイズとしては、問題情報を規定する変数値の中で、時間（領域）計算量に影響を与える変数値が用いられる。例えば、グラフに関する問題では、節点数や枝数を問題のサイズとし、スケジューリング問題では、ジョブ数、プロセッサ数、ジョブの処理時間などを問題のサイズとする。更に、時間（領域）計算量としては、問題のサイズ  $n$  を大きくしていったときの極限におけるふるまいを表わす、漸近的時間（領域）計算量が用いられ、 $O(\cdot)$  の記法で表現される。すなわち、時間計算量が  $O(f(n))$  であるというのは、十分大きなサイズ  $n$  をもつすべての入力データに対して、 $t_A(n) \leq c f(n)$  ( $c$ ; 定数) が成り立つことである。このような問題のサイズ及び評価関数を用いることにより、片言アルゴルで記述された解法の時間（領域）計算量  $O(f(n))$  は、RAM プログラムにおいても成立する<sup>(注2)</sup> ことが容易に確かめられる。

### 1.2.2 NP 完全問題

計算機械のモデルとしてチューリング機械を用い、NP 完全問題が定義される<sup>(11)</sup>。

まず、与えられた問題  $P$  を適当な符号化によって、言語の認識問題に変換する。  $P$  が、ある条件を満たす解の存在を問う決定問題である場合には、この変

---

(注1) 最大時間（領域）計算量に対して、平均時間（領域）計算量があり、すべての入力データに対する時間（領域）計算量の平均値として定義される。しかし、一般に、問題に対する入力データの分布を求めることは困難であり、通常、その解析の容易さから、計算量として最大計算量が用いられる。

(注2) 片言アルゴルでは、手続き（procedure）の再帰呼出し（recursive call）が許されている。従って、再帰呼出しのある手続きを RAM プログラムに変換した場合に必要なスタックなどは、解法の記述に陽には表われない。このような点にのみ注意し、そのようなスタック操作も評価の中に考慮することによって、この結果が得られる。

換は直接的に行われる。一方、 $P$  が最適化問題である場合には、 $\log k$  (但し、 $k$  はすべての可能な解の値の数) 個の決定問題に変換できる。解の値の数が、たかだか  $c^n$  ( $c$  はある定数、 $n$  は問題のサイズ) であるとき、決定問題が  $NP$  完全ならば、それに対応する最適化問題も  $NP$  完全であり、その逆も成り立つことが容易に示される。

従って、以下では、 $NP$  完全の定義を言語の認識問題を用いて行う。

〔定義 1.3〕<sup>(11)</sup> 決定性チューリング機械  $DTM$  が時間計算量  $T(n)$  であるというのは、長さ  $n$  のすべての入力列に対して、たかだか  $T(n)$  回の動作で受理状態に行く列が存在することをいう。又、 $DTM$  の動作回数が  $T(n)$  に限定されるとき、この  $DTM$  を  $T(n)$  時間限定  $DTM$  と呼ぶ。

非決定性チューリング機械  $NDTM$  についても同様に定義される。

〔定義 1.4〕 多項式時間限定  $DTM$  によって受理される言語の集合を  $\mathcal{P}$ 、多項式時間限定  $NDTM$  によって受理される言語の集合を  $\mathcal{NP}$  で表わす。

〔定義 1.5〕<sup>(11)</sup> 言語  $L_0$  が  $NP$  完全 (非決定性多項式時間完全) というのは、次の (i)、(ii) が成り立つことである。

(i)  $L_0$  は  $\mathcal{NP}$  に属す。

(ii)  $\mathcal{NP}$  のすべての言語  $L$  が  $L_0$  に多項式変換可能である。但し、 $L$  が  $L_0$  に多項式変換可能というのは、 $L$  の任意の文字列  $w$  を  $L_0$  の文字列  $w_0$  に変換する多項式時間限定  $DTM$  が存在し、 $w$  が  $L$  に属す必要十分条件は  $w_0$  が  $L_0$  に属すことである。

与えられた問題が  $NP$  完全であることを示すのは、さほど難しいことではない。というのは、一度、ある問題  $L_0$  が  $NP$  完全であることが示されると、新しい問題  $L'_0$  が  $NP$  完全であることを証明するには、 $L'_0$  が  $\mathcal{NP}$  に属していて、 $L_0$  が  $L'_0$  に多項式変換可能であることを示せばよいからである。

後で述べる符号化の方法のみに注意すれば、多項式時間限定  $DTM$  の動作を、 $RAM$  を用いて多項式時間計算量で模倣することは容易である。しかし、残念ながら現在までのところ、多項式時間限定  $NDTM$  の動作を多項式時間計算量で模倣する  $RAM$  プログラムは見つかっておらず、従って現在までのところ、

$NP$  完全な問題を解く多項式時間解法は見い出されていない。又、定義 1.5 (ii) から、 $NP$  完全である 1 つの問題を解く多項式時間解法が存在すれば、その解法によって、 $NP$  完全である他のすべての問題が多項式時間計算量で解かれることになる。

与えられた問題が  $\mathcal{P}$  に属するか、 $NP$  完全であるかは、問題を言語に変換するとき用いられる符号化の方法によって異なることに注意する必要がある。符号化の方法としては、通常、各数値を 2 進法で表現する方法 (binary notation) や、数値の大きさを 1 の連鎖で表わす方法 (unary notation) が用いられる<sup>(13)</sup>。特に、前者の符号化のもとでのみ  $NP$  完全である問題は弱  $NP$  完全と呼ばれ、後者の符号化のもとでも  $NP$  完全である問題は 強  $NP$  完全 と呼ばれる<sup>(13)</sup>。

例えば、次のスケジューリング問題を考える。

[SCH] 整数の処理時間  $\tau_k$  ( $k=1 \dots n$ ) をもつ独立なジョブ集合  $J = \{1 \dots n\}$  を、2 台のプロセッサで処理するとき、最大完了時刻が  $K$  以下のスケジュールが存在するか。

SCH は、RAM によって  $O(n \sum_{k=1}^n \tau_k)$  の時間計算量で解かれる。SCH を定義する変数値を 1-連鎖を用いて符号化するとき、文字列の長さは  $\sum_{k=1}^n \tau_k$  となり、SCH は  $\mathcal{P}$  に属することになる。一方、変数値を 2 進法で符号化するとき、文字列の長さは  $\sum_{k=1}^n \lceil \log \tau_k \rceil$  となり、 $n \sum_{k=1}^n \tau_k$  は文字列の長さに対して指数関数となる。通常の電子計算機を想定する場合には、2 進法による符号化がより現実的といえ、この場合には SCH は  $NP$  完全となる。

$NP$  完全の定義を、2 進法による符号化に限定すれば、弱  $NP$  完全な問題は、問題を定義する変数値を 1-連鎖で表現したときの長さが問題のサイズの多項式でおさえられるときに  $\mathcal{P}$  に属し、そうでないとき  $NP$  完全となる。先の問題 SCH において、処理時間  $\tau_k$  ( $k=1 \dots n$ ) が  $0 < \tau_k \leq K(n)$  ( $K(n)$  は  $n$  の多項式) に限定された問題を考えると、この問題は、問題のサイズ  $n$  の多項式時間計算量  $O(n^2 K(n))$  で解けることになる。混乱を避けるためには、問題を定義する変数値が任意の値であるか、制限された値であるかを明記することが必要である。

又、解法の時間計算量を評価する立場からは、変数値が上記のように制限されたときのみ多項式時間計算量で問題を解く解法は、擬似多項式時間解法と呼ばれる<sup>(14)</sup>。

### 1.2.3 スケジューリング問題に対する解法の時間計算量

各ジョブが先行関係をもたないスケジューリング問題に対して、著者の知る限り、最も優れた時間計算量をもつ解法の時間計算量を表1.1に示す。

表1.1において、 $\tau_k, c_k, d_k$ は、各々、ジョブの処理時間、単位時間当りのコスト、納期を表わし、 $m$ はプロセッサの数を表わす。又、 $f_k$ はスケジュールによるジョブの完了時刻を表わし、 $t_k \equiv \max(f_k - d_k, 0)$ はジョブの納期遅れを表わす。更に、 $\delta_k$ は、 $f_k > d_k$ のとき1、 $f_k < d_k$ のとき0となる関数を表わす。

なお、表1.1のすべての問題において、ジョブの到着時刻は等しく、ジョブは先取権をもつことを仮定している。又、問題5以外のすべての問題において、プロセッサは同一なプロセッサであることを仮定している。

表1.1 各ジョブが先行関係をもたないスケジューリング問題に対する解法の時間計算量

問題番号	評価関数	$c_k$	$\tau_k$	$d_k$	$m$	時間計算量	解法
1	$\max f_k$	(注1) ×	(注2) $E$	×	(注3) —	$O(n)$	(i)
2	$\max f_k$	×	—	×	$\geq 2$ の定数	弱NP完全	
3	$\max f_k$	×	—	×	—	強NP完全	
4	$\sum c_k f_k$	$E$	—	×	—	$O(n \log n)$	(ii)
5	$\sum c_k f_k$	$E$	$\tau_{k_j} (j=1 \dots m)$ が異なる		—	$O(n^3)$	(iii)
6	$\sum c_k f_k$	—	—	×	1	$O(n \log n)$	(iv)
7	$\sum c_k f_k$	—	$E$	×	—	$O(n \log n)$	(v)
8	$\sum c_k f_k$	—	—	×	$\geq 2$ の定数	弱NP完全	
9	$\sum c_k f_k$	—	—	×	—	強NP完全	



問題番号	評価関数	$c_k$	$\tau_k$	$d_k$	$m$	時間計算量	解法
10	$\max t_k$	×	—	—	1	$O(n \log n)$	(vi)
11	$\max t_k$	×	$E$	—	—	$O(n \log n)$	(vi)
12	$\sum c_k t_k$	$E$	—	$E$	1	$O(n \log n)$	(ii)
13	$\sum c_k t_k$	$E$	$E$	—	—	$O(n \log n)$	(vi)
14	$\sum c_k t_k$	$E$	—	—	1	$O_{pen}$ <sup>(注4)</sup>	
15	$\sum c_k t_k$	—	$E$	—	1	$O(n^3)$	(vii)
16	$\sum c_k t_k$	—	—	—	1	強NP完全	
17	$\sum c_k \delta_k$	$E$	—	—	1	$O(n^2)$	(viii)
18	$\sum c_k \delta_k$	$E$	—	$E$	1	$O(n \log n)$	(ii)
19	$\sum c_k \delta_k$	—	—	—	1	弱NP完全	

(注1) 対応する変数値が評価関数に無関係であることを表わす。

(注2) 対応する変数値がすべてのジョブについて等しいことを表わす。

(注3) 対応する変数値が任意の値をもつことを表わす。

(注4) 多項式時間解法もNP完全であることも知られていない問題を表わす。

表 1.1 に示した時間計算量をもつ解法は、次の通りである。

- (i) 任意のリストスケジューリング
- (ii)  $\tau_k$  の非減少順リストスケジューリング<sup>(3),(2)</sup>
- (iii) 最大流量問題に定式化する解法<sup>(3)</sup>
- (iv)  $c_k/\tau_k$  の非増加順リストスケジューリング<sup>(5)</sup>
- (v)  $c_k$  の非増加順リストスケジューリング
- (vi)  $d_k$  の非減少順リストスケジューリング<sup>(2)</sup>
- (vii) 2部グラフマッチング問題に定式化する解法
- (viii) 文献(15)参照

又、問題 2, 8, 19 が NP 完全であることは、partition 問題がこれらの問題に多項式変換可能であることから示され<sup>(16),(12)</sup>、これらの問題が弱 NP 完全であることは、以下に述べる擬似多項式時間解法の存在から分かる。更に、問題 3, 9, 16 が強 NP 完全であることは、3-partition 問題がこれらの問

題に多項式変換可能であることから示される<sup>(17),(18)</sup>.

表 1.1 で,  $NP$  完全又は  $Open$  として示された問題の中で, 幾つかの問題に対しては, ある条件の下で最適スケジュールを求める多項式時間解法が知られている. 以下, 各問題ごとにこの解法を示す.

問題 2, 8: 擬似多項式時間解法として動的計画法を用いた解法<sup>(19)</sup>があり,

$O(n(\sum \tau_k)^{m-1})$  の時間計算量で最適スケジュールを求めることができる.  $\sum \tau_k$  及び  $m$  が小さい場合にはこの解法は有効である.

問題 14: 解法 (vi) で得られるスケジュールが 2 つ以上のジョブに納期遅れを起こさないとき, 又は, 解法 (ii) で得られるスケジュールがすべてのジョブに納期遅れを起こすとき, 各々の解法によって得られるスケジュールは最適スケジュールである<sup>(2)</sup>. 又, 擬似多項式時間解法として, 問題 2, 8 で述べた解法に類似した解法が文献 (17) に示されている.

問題 16: 任意のスケジュールによってすべてのジョブが納期遅れを起こすとき, 解法 (iv) によって得られるスケジュールは最適スケジュールである<sup>(5)</sup>.

問題 19: すべてのジョブ  $i, j$  の間に,  $\tau_i < \tau_j$  なら  $c_i \geq c_j$ , の関係が成り立つ場合について, 多項式時間解法がある<sup>(20)</sup>. 又, 動的計画法を用いて  $O(n \cdot \min(\sum c_k, \sum \tau_k, \max d_k))$  で解ける<sup>(19)</sup>.

表 1.1 では, 各ジョブが先取権をもつ問題に対して, その解法の時間計算量を示した. 一般に, 各ジョブが先取権をもたない場合には, 各ジョブを分割 (*split*) することによって, 最適スケジュールを求めることが容易になる.

例えば, 問題 2, 3 は, すべてのジョブの処理時間を単位時間 1 に分割することによって問題 1 に定式化でき, 容易に最適スケジュールを求めることができる. ただ, 第 3 章で考察する問題 8, 9 に対する最適スケジュールは, このような分割をもたないスケジュールの中に存在することが知られている<sup>(5)</sup>.

従って問題 8, 9 はジョブが先取権をもたない場合でも, やはり,  $NP$  完全である.

又, 各ジョブが先取権をもたない問題に対しては, できるだけ少ない分割数で最適スケジュールを求めることが, 実際的な立場から重要であり, この種の研究として文献 (21), (22) などがあるが, まだ多くの考究課題を残している.

又、ジョブの到着時刻  $a_k$  が異なる場合については、 $m=1$ で、且つ、すべてのジョブ  $i, j$  の間に、 $a_i \leq a_j$  なら  $d_i \leq d_j$  の関係が成り立つときのみ、 $\sum \delta_k$  を最小にする  $O(n^2)$  の解法<sup>(23)</sup>、及び、 $\max t_k$  を最小にする  $O(n \log n)$  の解法<sup>(24)</sup> が知られているが、 $a_k=0$  である場合と比較して、最適スケジュールを求めることが更に困難になる。

### 1.3 近似解法の最悪の場合の評価

前節で、多くのスケジューリング問題は  $NP$  完全であることを示した。これらの問題に対して実際に解を求めるという立場にたてば、最適解でないまでも、最適解の値に近い値をもつ解を求めることが必要となる。

本節では、文献(7)に従って、近似解法の良さを測る尺度として最悪誤差を定義し、表1.1に示した  $NP$  完全なスケジューリング問題に対する近似解法の最悪誤差を示す。

#### 1.3.1 近似解法の最悪誤差

前節と同様に、問題  $P$  を  $RAM$  に対する入力データの集合と考え、これを  $\mathcal{I}$  で表わす。又、 $\mathcal{I}$  に属す任意の  $I$  に解法  $A$  を適用して得られる解の値を  $Z_A(I)$ 、 $I$  に対する最適解の値を  $Z^*(I)$  とし、 $\epsilon_A(I)$  を次のように定義する。

$$\epsilon_A(I) = \frac{Z_A(I) - Z^*(I)}{Z^*(I)} \quad (P \text{ が最小化問題のとき})$$

$$\epsilon_A(I) = \frac{Z^*(I) - Z_A(I)}{Z^*(I)} \quad (P \text{ が最大化問題のとき})$$

〔定義1.6〕  $\mathcal{I}$  に属す、すべての  $I$  に対する  $\epsilon_A(I)$  の最大値  $\epsilon_A$  を、解法  $A$  の最悪誤差と呼ぶ。又、 $\epsilon_A \leq r$  を満たす  $r$  が存在するとき、この  $r$  を最悪誤差の上限値と呼び、特に、 $\epsilon_A(I) = r$  となる  $I \in \mathcal{I}$  が存在するとき、 $r$  を最良の上限値と呼ぶ。

最悪誤差  $\epsilon_A$  を用いた近似解法の評価は、最悪の場合の評価と呼ばれ、 $\epsilon_A \leq r$  を満たす  $r$  が十分小さな値であるとき、すべての入力データに対して解法  $A$  の良さを保証することになる。

1.3.2 スケジューリング問題に対する近似解法の最悪の場合の評価

表 1.1 に示した NP 完全なスケジューリング問題に対する最悪誤差の上限值を以下に示す。

問題 2, 3 に対する近似解法について, 次の結果が得られている。

(i) 任意のリストスケジューリング  $L$  を用いるとき,

$$\epsilon_L \leq (m-1) \max \tau_k / \sum \tau_k^{(3)} \quad (1.1)$$

が成り立ち, 右辺は最良の上限值である。

(ii)  $\tau_k$  の非増加順リストスケジューリング  $LPT$  を用いるとき,

$$\epsilon_{LPT} \leq \frac{1}{3} (1 - 1/m)^{(25)} \quad (1.2)$$

が成り立ち, 右辺は最良の上限值である。

(iii)  $\tau_k$  の非増加順に選ばれた  $k$  個のジョブに対して最適スケジュールを求め, 残りのジョブに任意のリストスケジューリングを適用する解法  $L'$  を用いて, 次式が成り立つ。

$$\epsilon_{L'} \leq \frac{1 - 1/m}{1 + k/m} \quad (25) \quad (1.3)$$

又,  $k$  が  $m$  の倍数のとき, 右辺は最良の上限值である。式 (1.3) の右辺は  $k \geq 2m$  のとき, 式 (1.2) の右辺よりも小さな値となる。一方,  $L'$  の時間計算量は  $O(k/n)$  であり, この解法は  $m$  が小さな値をもつ場合に有効といえる。

(iv) 動的計画法に区間分割 (*interval partitioning*) と呼ばれる操作<sup>(注)</sup>を加えた解法  $DIP$  を用い,

$$\epsilon_{DIP} \leq \delta \quad (1.4)$$

を得るために要す時間計算量は,  $m = 2$  のときで  $O(n^2/\delta)$ ,  $m \geq 3$  のときで  $O(n(n^2/\delta)^{m-1})$  となることが文献 (19) に示されている。この解法は, 特に

---

(注)  $[0, \sum \tau_k]$  を幾つかの区間に分割し, 同じ区間内の時刻を同一時刻と見なすことによって, 動的計画法の各段階における部分問題の数を減らす操作。

$m = 2$  の場合に有効な解法といえる。

又、問題 8, 9 に対する近似解法について、次の結果が得られている。

(V)  $c_k/\tau_k$  の非増加順リストスケジューリング  $SWPT$  に対して

$$\varepsilon_{SWPT} < 1/2 (1 - 1/m) \quad (1.5)$$

が成り立つ。<sup>(26)</sup>  $Eastman$  は、 $m = 1$  の場合の最適解  $Z_1^*$  を用いて、 $m \geq 2$  の場合の最適解  $Z^*$  の下限値及び  $SWPT$  による解  $Z_{SWPT}$  の上限値が、各々、次式で与えられることを示している。

$$Z^* \geq \frac{1}{m} Z_1^* + \frac{m-1}{2m} Z_n \quad (1.6)$$

$$Z_{SWPT} < \frac{1}{m} Z_1^* + \frac{m-1}{m} Z_n \quad (1.7)$$

但し、 $Z_n$  は次式で定義される値である。

$$Z_n \equiv \sum c_k \tau_k \quad (1.8)$$

$Z^* \geq Z_n$  であることを用いて、式 (1.5) が得られる。

第 3 章では、 $\sum \tau_k/m$  の値を越える処理時間をもつジョブが存在しないとき、 $SWPT$  に 1 つの操作を加えた解法によって、式 (1.5) の右辺の値が  $1/4 (1 - Z_n/2Z^*)$  に改善でき、しかも、この値が最良の上限値となることを見出ししている。

(vi) 特に  $m = 2$  の場合には、(iv) で述べた解法  $DIP$  を用いて、 $O(n^2/\delta)$  の時間計算量で

$$\varepsilon_{DIP} \leq \delta \quad (1.9)$$

が得られることが文献 (19) に示されている。

(vii) すべてのジョブについて  $c_k/\tau_k$  の値が等しいとき、 $LPT$  を用いて、

$$\varepsilon_{LPT} \leq 25/24 (1 - Z_n / 2Z^*) \quad (1.10)$$

が成り立つことが文献(27)に示されている。但し、 $Z_n$  及び  $Z^*$  は (V) で定義された値を表わす。

なお、納期を含む評価関数をもつ問題や、各ジョブの到着時刻が異なる問題に対しては、現在までのところ有効な近似解法は知られていない。

## 第2章 リソース条件をもたない問題に対する 厳密解法

### 2.1 緒 言

本章では、リソース条件をもたないスケジューリング問題に対して、最適スケジュールを求めるために有効な解法を提案する。提案する解法は、最近、巡回セールスマン問題<sup>(28),(29)</sup>、多品種流量問題<sup>(30)</sup>、整数計画問題<sup>(31),(32),(33)</sup>、スケジューリング問題<sup>(4),(34)</sup>などに対して検討されている双対アプローチを、本問題に対して適用することから導出したものである。その意味で、方法的には、Fisher<sup>(4)</sup>の解法と類似しているが、解法の中心的な役割を果たす“relaxation”の方法において新しい方法を取入れ、計算効率を優れたものにしていく。

まず、問題を0-1整数計画問題に定式化し、この定式化の下でラグランジュ関数の値を最適スケジュールの値に近づける山登り法を提案する。次いで、山登り法によって得られる値を下限として利用する分枝限定法を提案する。最後に、重み付き平均滞留時間を評価関数とする問題を用いてこの解法を実験的に調べ、この解法が既存の解法より計算効率の面で優れていることを確かめる。又、山登り法によって得られる下限値は、短時間で求まるだけでなく、それが最適スケジュールの値に対し十分よい近似値となることを確かめる。

### 2.2 諸 定 義

まず、本章で考えるスケジューリング問題を定義する。以下、このスケジューリング問題を G S C H問題と呼ぶ。

〔定義2.1〕(G S C H問題)

- (i) ジョブ  $J = \{k \mid k \dots n\}$  は、整数値をもつ処理時間  $\tau_k$ 、単位時間当りのコスト  $c_k$  及び納期  $d_k$  によって規定される。
- (ii)  $m$  台の同一なプロセッサが与えられる。但し、 $m$  は任意の値をもつ。
- (iii) すべてのジョブは  $t = 0$  で処理可能である。(ジョブの間に先行関係がなく、すべてのジョブの到着時刻が等しいことを意味する。)

(IV) すべてのジョブは先取権をもつ。

(V) 評価関数  $\sum_{k=1}^n g_k(f_k)$  を最小にするスケジュールを求める。但し、 $f_k (k=1 \dots n)$  はスケジュールによるジョブ  $k$  の完了時刻を表わし、 $g_k(f_k)$  ( $k=1 \dots n$ ) は  $f_k$  に関する単調非減少関数である。

表 1.1 に示した

$$\sum_{k=1}^n c_k f_k \quad (2.1)$$

$$\sum_{k=1}^n \max(f_k - d_k, 0) \quad (2.2)$$

$$\sum_{k=1}^n c_k \max(f_k - d_k, 0) \quad (2.3)$$

$$\sum_{k=1}^n \delta(f_k - d_k) \quad (2.4)$$

$$\sum_{k=1}^n c_k \delta(f_k - d_k) \quad (2.5)$$

などは、すべて  $\sum_{k=1}^n g_k(f_k)$  の形で表現できる。但し、 $\delta(f_k - d_k)$  は  $f_k \leq d_k$  のとき 0、 $f_k > d_k$  のとき 1 となる関数を表わす。

〔定義 2.2〕 GSCH 問題において、各ジョブは先取権をもつことから、

$$x_k \equiv f_k - \tau_k + 1 \quad (k=1 \dots n) \quad (2.6)$$

とおき、以下、この  $x_k$  をジョブ  $k$  の開始時刻と呼ぶ。又、 $x_k$  を要素とする  $n$  次元ベクトル  $\tilde{x} \equiv (x_1 \dots x_n)$  でスケジュールを表わす。更に、定義 2.1 (V) の  $g_k(f_k)$  を  $w_k(x_k)$  とおき、GSCH 問題の評価関数を  $\sum_{k=1}^n w_k(x_k)$  で表わす。

式 (2.6) で定義される  $x_k$  は、ジョブの開始時刻として通常用いられる値より 1 だけ大きい値であることを注意しておく。すなわち、 $x_k = t$  はジョブ  $k$  が  $t-1$  の時刻にプロセッサに割当てられることを意味する。

〔定義 2.3〕 プロセッサ条件を満たす (各時刻に処理されるジョブの数が  $m$  以下である) スケジュール  $\tilde{x}$  を可能スケジュールと呼ぶ。

GSCH 問題においては、各ジョブの処理時間は整数であることから、スケジュール  $\tilde{x}$  によって  $[t-1, t]$  (但し  $t$  は整数) の区間に処理されるジョブは

$$\max(1, t - \tau_k + 1) \leq x_k \leq t$$

なる  $x_k$  をもつジョブと一致する。このジョブの数を  $G_t(\tilde{x})$  で表わすとき、可



能スケジュール $\tilde{x}$ は、すべての時刻 $t$ に対して、

$$G_t(\tilde{x}) \leq m$$

を満たすスケジュールである。

〔定義 2.4〕 式(2.1)～(2.5)の評価関数をもつ問題を、それぞれ、WMFT問題、MT問題、WMT問題、NTJ問題およびWNTJ問題と呼ぶ。

### 2.3 整数計画問題への定式化

まず、GSC H問題の最適スケジュールについていえることを定理の形で示しておく。

〔定理 2.1〕 GSC H問題に対する1つの可能スケジュール $\tilde{x} \equiv (x_1 \cdots x_n)$ が遊休時間をもつとき、

$$\sum_{k=1}^n w_k(x'_k) \leq \sum_{k=1}^n w_k(x_k) \quad (2.7)$$

を満たす、遊休時間をもたない可能スケジュール $\tilde{x}'$ が必ず存在する。特にWMFT問題に対しては、式(2.7)で等号を除いた式が成り立つ。

(証明) GSC H問題における $w_k(x_k)$ は $x_k$ に関する単調非減少関数であり、特に、WMFT問題における $w_k(x_k)$ は単調増加関数である。又、各ジョブは任意の時刻に処理可能であることから、遊休時間をもつプロセッサにおいて、その後続くジョブの開始時刻を遊休時間の値だけ早めて得られるスケジュールは可能スケジュールである。以上のことから、定理の結果は明らかである。

(証明終)

定理 2.1の結果から、GSC H問題においては、遊休時間をもたない可能スケジュールの中に必ず最適スケジュールが存在し、特にWMFT問題においては、すべての最適スケジュールは遊休時間をもたないことが分かる。WMFT問題以外の問題においても、遊休時間をもたないスケジュールの中から最適スケジュールを求めることによって、次節で述べる解法の計算効率を高めることができる。

遊休時間をもたないスケジュールに対して次の定理が成り立つことが、文献(3)に示されている。

〔定理 2.2〕  $T_1, T$  を次式で表わされる値とする。

$$T_1 = \lfloor \sum \tau_k / m \rfloor - \max \tau_k \quad (2.8)$$

$$T = \lceil \{ \sum \tau_k + (m-1) \max \tau_k \} / m \rceil \quad (2.9)$$

このとき、遊休時間をもたない可能スケジュール  $\pi$  に対して、次の (i), (ii) が成り立つ。

- (i)  $T_1$  より早い時刻に処理を終るプロセッサは存在しない。
- (ii) 最大完了時刻は  $T$  以下である。

(定理終)

又、ジョブを規定する変数値の間に成り立つ関係を用いて、最適スケジュールが満たすべき条件を導出する。この条件は、解法の効率を高めるために利用できる。

〔定理 2.3〕 ジョブ  $i$  とジョブ  $j$  の間に、 $\tau_i \leq \tau_j$  且つ  $c_i > c_j$  なる関係が存在すれば、 $WMFT$  問題の最適スケジュール  $\tilde{x}^* \equiv (x_1^* \dots x_n^*)$  において、 $x_i^* \leq x_j^*$  が成り立つ。

(証明) 次の (i), (ii) の場合に分けて証明を行う。

- (i)  $i, j$  が同じプロセッサで処理される場合。

仮定により、 $c_i / \tau_i > c_j / \tau_j$  である。一方、 $m=1$  の場合には、ジョブを  $c_k / \tau_k$  の非増加順に処理することによって評価関数の値を最小にできる。このことから、 $i, j$  が同じプロセッサで処理される場合には  $x_i^* < x_j^*$  が成り立つ。

- (ii)  $i, j$  が異なるプロセッサで処理される場合。

$\tau_i \leq \tau_j$  且つ  $c_i > c_j$  を満たす2つのジョブが  $\tilde{x}^*$  において、 $x_i^* > x_j^*$  であったと仮定する。 $i, j$  を処理するプロセッサを各々  $m_i, m_j$  とし、 $m_i$  において  $i$  の後に処理されるジョブの集合を  $A$ 、 $m_j$  において  $j$  の後に処理されるジョブの集合を  $B$  とする。更に、 $C_A \equiv \sum_{k \in A} c_k$ 、 $C_B \equiv \sum_{k \in B} c_k$  とおく。ジョブ  $i, j$

及びジョブ集合  $A, B$  の間の関係を図 2.1 に示す.

$\tilde{x}^*$  に対して, 図 2.2, 図 2.3 で示されるスケジュール  $\tilde{x}^1, \tilde{x}^2$  を考える.  
但し,  $\delta \equiv \tau_j - \tau_i$  である.

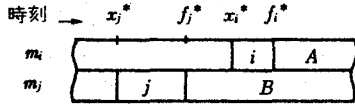


図 2.1 スケジュール  $\tilde{x}^*$  のガントチャート

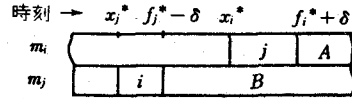


図 2.2 スケジュール  $\tilde{x}^1$  のガントチャート

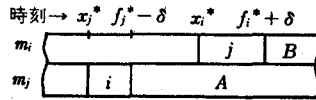


図 2.3 スケジュール  $\tilde{x}^2$  のガントチャート

$\tilde{x}^1$  と  $\tilde{x}^*$  の間には, 式 (2.10) の関係がある.

$$\left. \begin{aligned} x_j^1 &= x_i^*, & x_i^1 &= x_j^* \\ x_k^1 &= x_k^* + \delta & (k \in A) \\ x_k^1 &= x_k^* - \delta & (k \in B) \\ x_k^1 &= x_k^* & (k \notin A \cup B \cup \{i\} \cup \{j\}) \end{aligned} \right\} \quad (2.10)$$

従って,

$$WMFT(\tilde{x}) \equiv \sum_{k=1}^n c_k (x_k + \tau_k - 1) \quad (2.11)$$

とおけば, 次式が成り立つ.

$$WMFT(\tilde{x}^*) - WMFT(\tilde{x}^1) = (c_i - c_j)(x_i^* - x_j^*) + \delta(C_A - C_B) \quad (2.12)$$

一方、 $\tilde{x}^2$ と $\tilde{x}^*$ の間には、式(2.13)の関係がある。

$$\left. \begin{aligned} x_j^2 &= x_i^* , & x_i^* &= x_j^* \\ x_k^2 &= x_k^* + x_j^* - x_i^* & (k \in A) \\ x_k^2 &= x_k^* - x_j^* + x_i^* & (k \in B) \\ x_k^2 &= x_k^* & (k \notin A \cup B \cup \{i\} \cup \{j\}) \end{aligned} \right\} \quad (2.13)$$

従って、次式が成り立つ。

$$\begin{aligned} WMFT(\tilde{x}^*) - WMFT(\tilde{x}^2) \\ = (c_i - c_j)(x_i^* - x_j^*) + (C_A - C_B)(x_i^* - x_j^*) \end{aligned} \quad (2.14)$$

$C_B \geq C_A$  の場合、式(2.12)より、

$$WMFT(\tilde{x}^*) > WMFT(\tilde{x}^1) \quad (2.15)$$

$C_B < C_A$  の場合、式(2.14)より、

$$WMFT(\tilde{x}^*) > WMFT(\tilde{x}^2) \quad (2.16)$$

となって、いずれの場合にも、 $\tilde{x}^*$ より小さい $WMFT$ の値をもつスケジュールが存在する。これは、 $\tilde{x}^*$ が最適スケジュールであることに矛盾する。従って、 $i, j$ が異なるプロセッサで処理される場合についても、 $x_i^* \leq x_j^*$ が成立する。

(証明終)

又、 $MT$ 問題、 $NTJ$ 問題に対して次の定理が成り立つ。

〔定理2.4〕  $MT$ 問題、 $NTJ$ 問題においてジョブ $i, j$ の間に、 $d_i > d_j$ 且つ $\tau_i = \tau_j$ なる関係が成り立つとする。このとき、 $x_j \leq x_i$ を満たすスケジュールの中に最適スケジュールが必ず存在する。

(証明)  $NTJ$ 問題に対しても同様に証明できるので、ここでは、 $MT$ 問題に対して証明を行なう。

すべての最適スケジュール  $\tilde{x}^* \equiv (x_1^* \cdots x_n^*)$  が  $x_j^* > x_i^*$  を満たすと仮定し、矛盾を導く。

この  $\tilde{x}^*$  に対して、式 (2.17) で与えられるスケジュール  $\tilde{x}' \equiv (x'_1 \cdots x'_n)$  を考える。

$$\left. \begin{aligned} x'_i &= x_j^* & , & & x'_j &= x_i^* \\ x'_k &= x_k & (k \neq i, j) \end{aligned} \right\} \quad (2.17)$$

又、

$$MT(\tilde{x}) \equiv \sum_{k=1}^n \max(x_k + \tau_k - d_k - 1, 0) \quad (2.18)$$

とおけば、式 (2.17) から

$$\begin{aligned} \Delta &\equiv MT(\tilde{x}^*) - MT(\tilde{x}') \\ &= \max(x_i^* + \tau - d_i, 0) + \max(x_i^* + \tau - d_j, 0) \\ &\quad - \max(x_j^* + \tau - d_i, 0) - \max(x_j^* + \tau - d_j, 0) \end{aligned} \quad (2.19)$$

が得られる。但し、 $\tau \equiv \tau_i - 1 = \tau_j - 1$  である。

式 (2.19) の第 1 項～第 4 項の絶対値を各々、 $A_1 \sim A_4$  で表わせば、 $d_i > d_j$  且つ  $x_i^* < x_j^*$  の関係から、

$$A_1 \leq A_3, A_4 \leq A_2 \quad (2.20)$$

が成り立つ。式 (2.20) から  $A_l (1 \leq l \leq 4)$  について 6 通りの零又は正の場合分けを行えばよいことになり、これらすべての場合に対して  $\Delta \geq 0$  が容易に導ける。

以上から、常に、 $MT(\tilde{x}^*) \geq MT(\tilde{x}')$  が成り立ち、 $\tilde{x}'_i > \tilde{x}'_j$  であることから、すべての最適スケジュール  $\tilde{x}^*$  が  $x_j^* > x_i^*$  を満たすという仮定に矛盾する。従って、 $x_j \leq x_i$  を満たすスケジュールの中に必ず最適スケジュールが存在する。  
(証明終)

定理 3.4 を,  $\tilde{x}^*$  が最適スケジュールであるための必要条件の形で表わすとき, 次の系が得られる.

〔系 2.1〕  $MT$  問題,  $NTJ$  問題において, ジョブ  $i, j$  の間に  $d_i > d_j$  且つ  $\tau_i = \tau_j$  なる関係が成り立つとする. このとき, 最適スケジュール  $\tilde{x}^* \equiv (x_1^* \dots x_n^*)$  に対して次のことがいえる. 但し,  $\tau \equiv \tau_i - 1 = \tau_j - 1$  である.

$$x_i^* < d_i - \tau \quad \text{ならば} \quad x_j^* \leq d_i - \tau \quad (2.21)$$

$$x_i^* < d_j - \tau \quad \text{ならば} \quad x_j^* \leq d_j - \tau \quad (2.22)$$

(証明) 式 (2.21), (2.22) が成り立たないと仮定して, 式 (2.17) で与えられるスケジュール  $\tilde{x}'$  と  $\tilde{x}^*$  の間に  $MT(\tilde{x}') > MT(\tilde{x}^*)$  の関係が成り立つことが容易に導ける.

(証明終)

定理 2.3, 定理 2.4 (又は系 2.1) から, 各ジョブ  $k$  に対して,

$$S(k) \equiv \{j \mid x_k^* \leq x_j^*\} \quad (2.23)$$

$$P(k) \equiv \{j \mid x_j^* \leq x_k^*\}$$

なる 2 種類のリストをつくることができ, これらのリストは,  $x_k$  に対する下限  $a_k$  及び上限  $b_k$  の決定に使える. 2.4.3 項で述べる分枝限定法では, 幾つかのジョブの開始時刻が固定された部分問題を解くことになるが, 式 (2.23) のリストは, これらの部分問題を解く上で有効である.

以上のことを用いて,  $GSCH$  問題は次のような整数計画問題に定式化できる.

$$\min_{\tilde{x}} \sum_{k=1}^n w_k(x_k) \quad (2.24)$$

$$G_t(\tilde{x}) \quad \begin{cases} = m & (1 \leq t \leq T_1) \\ \leq m & (T_1 + 1 \leq t \leq T) \end{cases} \quad (2.25)$$

$$a_k \leq x_k \leq b_k \quad (1 \leq k \leq n) \quad (2.26)$$

$$x_k \text{ は整数} \quad (2.27)$$

但し、 $G_t(\tilde{x})$  は、スケジュール  $\tilde{x}$  によって区間  $[t-1, t]$  に要求されるプロセッサの数を表わす。

式 (2.24) ~ (2.27) における各ジョブの開始時刻  $x_k$  は 0-1 変数  $x_{kt}$  ( $a_k \leq t \leq b_k$ ) を用いることによって、次のように書換えられる。

$$x_k = \sum_{t=a_k}^{b_k} t \cdot x_{kt} \quad (1 \leq k \leq n) \quad (2.28)$$

但し、

$$\sum_{t=a_k}^{b_k} x_{kt} = 1 \quad (1 \leq k \leq n) \quad (2.29)$$

$$x_{kt} = 0 \text{ 又は } 1$$

$$(1 \leq k \leq n, a_k \leq t \leq b_k) \quad (2.30)$$

又、 $x_{kt}$  を要素とするベクトル  $(x_{1a_1} \cdots x_{1b_1} \cdots x_{na_n} \cdots x_{nb_n})$  を  $\tilde{x}'$  とおく。

更に、 $x_{kt}$  の関数  $w_{kt}(x_{kt})$  を用いて、 $w_k(x_k)$  を次式で表わす。

$$w_k(x_k) = \sum_{t=a_k}^{b_k} w_{kt}(x_{kt}) \quad (2.31)$$

$WMFT$  問題、 $WMT$  問題 ( $MT$  問題)、 $WNTJ$  問題 ( $NTJ$  問題) に対して  $w_{kt}(x_{kt})$  は、それぞれ、式 (2.32)、(2.33)、(2.34) で与えられる。

$$w_{kt}(x_{kt}) = (t-1) c_k x_{kt} \quad (2.32)$$

$$w_{kt}(x_{kt}) = \begin{cases} c_k (t + \tau_k - d_k - 1) x_{kt} & (t + \tau_k - 1 > d_k \text{ のとき}) \\ 0 & (t + \tau_k - 1 \leq d_k \text{ のとき}) \end{cases} \quad (2.33)$$

$$w_{kt}(x_{kt}) = \begin{cases} c_k \cdot x_{kt} & (t + \tau_k - 1 > d_k \text{ のとき}) \\ 0 & (t + \tau_k - 1 \leq d_k \text{ のとき}) \end{cases} \quad (2.34)$$

$x_{kt}$  及び  $w_{kt}(x_{kt})$  を用いて、式 (2.24) ~ (2.27) は次のように書換えられる。

$$\min_{\tilde{x}} \sum_{k=1}^n \sum_{t=a_k}^{b_k} w_{kt}(x_{kt}) \quad (2.35)$$

$$\sum_{k=1}^n \sum_{\tau \in A_{kt}} x_{k\tau} \begin{cases} = m & (1 \leq t \leq T_1) \\ \leq m & (T_1+1 \leq t \leq T) \end{cases} \quad (2.36)$$

$$\sum_{t=a_k}^{b_k} x_{kt} = 1 \quad (1 \leq k \leq n) \quad (2.37)$$

$$x_{kt} = 0 \text{ 又は } 1 \quad (1 \leq k \leq n, a_k \leq t \leq b_k) \quad (2.38)$$

但し、式 (2.36) における  $A_{kt}$  ( $1 \leq k \leq n, 1 \leq t \leq T$ ) は、式 (2.39) で示される時刻の集合を表わす。

$$A_{kt} = \{ \tau \mid t - \tau_k + 1 \leq \tau \leq t \text{ 且つ } a_k \leq \tau \leq b_k \} \quad (2.39)$$

〔定義 2.6〕 式 (2.25) を満たすスケジュール  $\tilde{x}$  を アクティブスケジュール と呼ぶ。従ってアクティブスケジュールは、 $T_1$  以前の時刻においてすべてのプロセッサがジョブを割当てられている可能スケジュールである。又、アクティブスケジュールに属す最適スケジュールを以下、単に 最適スケジュール と呼ぶ。

$W M F T$  問題の最適スケジュールはすべてアクティブスケジュールであり、又、それ以外の  $G S C H$  問題においても、アクティブスケジュールの中から最適スケジュールを選ぶことによって、次節で述べる解法の計算効率を高めることができる。これは、 $G_i(\tilde{x}) \leq m$  に対応するラグランジュ乗数  $u_i$  には  $u_i \geq 0$  の条件が課せられるのに対して、 $G_i(\tilde{x}) = m$  に対応するラグランジュ乗数は任意の値をもつことができるためである。



## 2.4 解法

### 2.4.1 ラグランジュ乗数法の適用

式(2.24)～(2.27)で示される問題に、ラグランジュ乗数法を適用することを考えてみる。式(2.25)に対するラグランジュ乗数を  $\tilde{u} \equiv (u_1 \dots u_T)$  とし、ラグランジュ関数を、

$$L(\tilde{u}) \equiv \min_{\tilde{x}} \left\{ \sum_{k=1}^n w_k(x_k) + \sum_{t=1}^T u_t (G_t(\tilde{x}) - m) \right\} \quad (2.40)$$

で与える。

$T_1 + 1 \leq t \leq T$  に対して、 $u_t \geq 0$  であれば、

$$\sum_{i=1}^n u_i (G_i(\tilde{x}^*) - m) \leq 0 \quad (2.41)$$

であることから、原問題の最適解  $z^* \equiv \sum_{k=1}^n w_k(x_k^*)$  と  $L(\tilde{u})$  の間には

$$z^* \geq L(\tilde{u}) \quad (2.42)$$

の関係が成り立つ。

一方、式(2.35)～(2.38)で示される問題に対してラグランジュ関数は、

$$L(\tilde{u}) = \min_{\tilde{x}'} \left\{ \sum_{k=1}^n \sum_{t=a_k}^{b_k} w_{kt}(x_{kt}) + \sum_{t=1}^n u_t \left( \sum_{k=1}^n \sum_{\tau \in A_{kt}} x_{k\tau} - m \right) \right\} \quad (2.43)$$

で与えられる。適当な計算の後、式(2.43)は

$$L(\tilde{u}) = \min_{\tilde{x}'} \left\{ \sum_{k=1}^n \sum_{t=a_k}^{b_k} \pi_{kt} x_{kt} - m \sum_{t=1}^T u_t \right\} \quad (2.44)$$

但し、

$$\pi_{k_i} \equiv w_{k_i}(x_{k_i}) + \sum_{\tau=i}^{i+\tau_k-1} u_{\tau} \quad \left( \begin{array}{l} 1 \leq k \leq n \\ a_k \leq i \leq b_k \end{array} \right) \quad (2.45)$$

となる。

式(2.37), (2.38)を制約条件とする式(2.44)の解は, 各 $k$ に対して

$$\pi_k \equiv \min \{ \pi_{k_i} \mid a_k \leq i \leq b_k \} \quad (2.46)$$

を求め,  $\pi_k$ を与える $i$ に対して $x_{k_i}=1$ , その他の $i$ に対して $x_{k_i}=0$ と置くことよって得られる。式(2.46)を与える $i$ が唯一に決まる場合,  $\tilde{x}$ に対する $\tilde{x}$ がアクティブスケジュールなら, 式(2.42)から, それは最適スケジュールとなる。 $\tilde{x}$ がアクティブスケジュールでない場合, 又は式(2.46)を与える $i$ が唯一に決定されない場合, 別の方法をとる必要があるが, これについては2.4.3項で述べる。

#### 2.4.2 山登り法

前節で示したことから分かるように, 解法としてまず問題となるのは,  $\tilde{u}$ の値を変化させて $L(\tilde{u})$ の値を原問題の最適解 $s^*$ に近づけることである。文献(4)の解法では, 式(2.35), (2.36)で定義される問題に対する双対問題を考え, この双対問題をシンプレックス法を用いて解く方法が取られるが, 本解法では, シンプレックス法に代わる手法として, 山登り法を用いる。以下, この手法を具体的に示す山登りの方向とステップ幅について検討する。

式(2.25)に対応して, 次の関数を定義する。

$$\phi_i(\tilde{x}) \equiv \begin{cases} G_i(\tilde{x}) - m & (1 \leq i \leq T_1) \\ \max(0, G_i(\tilde{x}) - m) & (T_1 + 1 \leq i \leq T) \end{cases} \quad (2.47)$$

$\phi_i(\tilde{x})$ を成分とする $T$ 次元ベクトル $(\phi_1(\tilde{x}) \cdots \phi_T(\tilde{x}))$ を $\tilde{\phi}(\tilde{x})$ とおけば, 次の定理が成り立つ。

〔定理2.5〕 次式で与えられる $\tilde{u}$ の系列 $\{\tilde{u}^l\}$  ( $l \geq 1$ )を考える。但し,  $T$ 次元零ベクトルを $\tilde{0}$ で表わす。

$$\left. \begin{aligned} \tilde{u}^1 &= \tilde{0} \\ \tilde{u}^{l+1} &= \tilde{u}^l + \lambda \tilde{\phi}(\tilde{x}) \end{aligned} \right\} \quad (2.48)$$

$\tilde{u} \in \{\tilde{u}^l\}$  に対して、 $L(\tilde{u})$  を最大にする  $\tilde{u}$  を  $\tilde{u}^* \equiv (u_1^* \cdots u_T^*)$ 、 $\tilde{u}^*$  に対する  $L(\tilde{u})$  の値を  $L(\tilde{u}^*)$  とすれば、 $\lambda > 0$  のとき、 $z^* \equiv \sum_{k=1}^* w_k(x_k^*)$  に対して、

$$L(\tilde{u}^*) \leq z^* \quad (2.49)$$

が成り立つ。

(証明) 式(2.47)より  $T_1 + 1 \leq i \leq T$  に対して  $\phi_i(\tilde{x}) \geq 0$  である。従って、 $\lambda > 0$  なら、 $u_i^l$  ( $T_1 + 1 \leq i \leq T$ ) は単調非減少な系列となる。一方、 $\tilde{u}^1 = \tilde{0}$  であることから、

$$u_i^* \geq 0 \quad (T_1 + 1 \leq i \leq T) \quad (2.50)$$

となり、 $u_i^*$  ( $1 \leq i \leq T$ ) に対して式(2.42)が成り立つ。よって、式(2.49)が導ける。

(証明終)

定理2.5で述べた  $\tilde{u}$  の系列  $\{\tilde{u}^l\}$  が  $\tilde{u}^*$  に収束するためには、 $\lambda$  にある条件が必要である。 $\lambda$  に対する条件を定理2.6に示すが、これに先立って、次の補題を証明しておく。

[補題2.1]  $\{\tilde{u}^l\}$  を式(2.48)で与えられる  $\tilde{u}$  の系列とする。 $\tilde{u}^l$ 、 $\tilde{u}^* \in \{\tilde{u}^l\}$  に対するラグランジュ関数の値を各々、 $L(\tilde{u}^l)$  及び  $L(\tilde{u}^*)$  とすれば、 $u_i^* \geq u_i^l$  ( $T_1 + 1 \leq i \leq T$ ) のとき

$${}^t(\tilde{u}^* - \tilde{u}^l) \tilde{\phi}(\tilde{x}^l) \geq L(\tilde{u}^*) - L(\tilde{u}^l) \quad (2.51)$$

が成り立つ。但し、 ${}^t(\tilde{u}^* - \tilde{u}^l)$  は  $\tilde{u}^* - \tilde{u}^l$  の転置を表わし、 $\tilde{x}^l$  は  $L(\tilde{u}^l)$  を与える  $\tilde{x}$  を表わす。

(証明)  $L(\tilde{u}^l)$  の定義から、

$$L(\tilde{u}^l) = \sum_{k=1}^n w_k(x_k^l) + \sum_{t=1}^T u_t^l (G_t(\tilde{x}^l) - m) \quad (2.52)$$

であり、更に  $L(\tilde{u}^*)$  を与える  $\tilde{x}$  を  $\tilde{x}^*$  で表わせば、

$$L(\tilde{u}^*) = \sum_{k=1}^n w_k(x_k^*) + \sum_{t=1}^T u_t^* (G_t(\tilde{x}^*) - m) \quad (2.53)$$

である。一方、

$$\delta \equiv \sum_{k=1}^n w_k(x_k^l) + \sum_{t=1}^T u_t^l \phi_t(\tilde{x}^l) \quad (2.54)$$

とおけば、式(2.47)を用いて

$$\delta = L(\tilde{u}^l) + \sum_{t \in T_l} u_t^l (m - G_t(\tilde{x}^l)) \quad (2.55)$$

が得られる。但し、式(2.55)の  $T_l$  は、次式で示される時刻の集合を表わす。

$$T_l \equiv \{t \mid G_t(\tilde{x}^l) < m \text{ 且つ } T_1 + 1 \leq t \leq T\} \quad (2.56)$$

一方、 $L(\tilde{u}^*)$  は  $\tilde{u}^*$  に関する最小値であることから、次式が成り立つ。

$$L(\tilde{u}^*) \leq \sum_{k=1}^n w_k(x_k^l) + \sum_{t=1}^T u_t^* (G_t(\tilde{x}^l) - m) \quad (2.57)$$

更に、

$$\rho \equiv \sum_{k=1}^n w_k(x_k^l) + \sum_{t=1}^T u_t^* \phi_t(\tilde{x}^l) \quad (2.58)$$

とおけば、式(2.47)を用いて、

$$\begin{aligned} \rho &= \sum_{k=1}^n w_k(x_k^l) + \sum_{t=1}^T u_t^*(G_t(\tilde{x}^l) - m) \\ &+ \sum_{t \in T_l} u_t^*(m - G_t(\tilde{x}^l)) \end{aligned} \quad (2.59)$$

が得られる。

式(2.54), (2.58)を用いて,

$$\rho - \delta = {}^t(\tilde{u}^* - \tilde{u}^l) \tilde{\phi}(\tilde{x}^l) \quad (2.60)$$

が得られ, 式(2.55), (2.57), (2.59)及び  $u_t^* \geq u_t^l (t \in T_l)$  を用いて,

$$\rho - \delta \geq L(\tilde{u}^*) - L(\tilde{u}^l) \quad (2.61)$$

が得られる。式(2.60), (2.61)より式(2.51)が導ける。

(証明終)

〔定理 2.6〕 式(2.52)で与えられる  $\tilde{u}$  の系列  $\{\tilde{u}^l\}$  に対するラグランジュ関数の最大値を  $L(\tilde{u}^*)$  とすると,

$$0 < \lambda < \frac{2(L(\tilde{u}^*) - L(\tilde{u}^l))}{\|\tilde{\phi}(\tilde{x}^l)\|^2} \quad (2.62)$$

ならば,

$$\|\tilde{u}^* - \tilde{u}^{l+1}\| < \|\tilde{u}^* - \tilde{u}^l\| \quad (2.63)$$

が成り立つ。

(証明) まず,

$$\begin{aligned} \|\tilde{u}^* - \tilde{u}^{l+1}\|^2 &= \|\tilde{u}^* - (\tilde{u}^l + \lambda \tilde{\phi}(\tilde{x}^l))\|^2 \\ &= \|\tilde{u}^* - \tilde{u}^l\|^2 - 2\lambda {}^t(\tilde{u}^* - \tilde{u}^l) \tilde{\phi}(\tilde{x}^l) + \lambda^2 \|\tilde{\phi}(\tilde{x}^l)\|^2 \end{aligned}$$

$$= \|\tilde{u}^* - \tilde{u}^l\|^2 + \lambda \{ \lambda \|\tilde{\phi}(\tilde{x}^l)\|^2 - 2^l (\tilde{u}^* - \tilde{u}^l) \tilde{\phi}(\tilde{x}^l) \} \quad (2.64)$$

であり、 $L(\tilde{u}^*) \geq L(\tilde{u}^l)$  であることから、補題 2.1 を用いて、上式は

$$\leq \|\tilde{u}^* - \tilde{u}^l\|^2 + \lambda \{ \lambda \|\tilde{\phi}(\tilde{x}^l)\|^2 - 2(L(\tilde{u}^*) - L(\tilde{u}^l)) \} \quad (2.65)$$

となる。更に、式(2.62)から、第2項は負となり、結局

$$\|\tilde{u}^* - \tilde{u}^{l+1}\|^2 < \|\tilde{u}^* - \tilde{u}^l\|^2 \quad (2.66)$$

が得られ、式(2.63)が導ける。

(証明終)

定理 2.5 及び定理 2.6 は、 $L(\tilde{u}^l)$  を  $L(\tilde{u}^*)$  に近づける有効な山登りの方向及びそのステップ幅を与えている。

### 2.4.3 分枝限定法

山登り法によって得られる  $\tilde{u}$  に対して、式(2.46)の  $\pi_k$  を与える  $i$  が唯一に決まり、しかも  $\tilde{x}$  がアクティブスケジュールであるとき、 $\tilde{x}$  は最適スケジュールである。 $\pi_k$  を与える  $i$  が唯一に決まらないとき、又は、 $\tilde{x}$  がアクティブスケジュールでないとき、本質的には、しらみつぶし的な方法に依らざるを得ない。ここでは、2.4.2項で述べた山登り法を、下限を得るために利用する分枝限定法を提案する。この場合、下限としては正確な  $L(\tilde{u}^*)$  の値は必要でなく、 $L(\tilde{u}^*)$  に十分近い値でよい。以下、提案する分枝限定法の分枝規則及び限定規則を示す。

◀分枝規則▶ 式(2.46)に従って各  $k$  に対して  $\pi_k$  を求める。 $\pi_k$  を与える  $i_k$  ( $a_k \leq i_k \leq b_k$ ) が唯一に決定できない場合は、それらの中で最小の  $i_k$  を選ぶ。各  $k$  に対して2番目に小さな  $\pi_{k,i}$  を求め、これを  $\pi_k'$  で表わす。

$\pi_k - \pi_k'$  の非増加順にジョブ  $k$  を並べ換え、この順序に従って、開始時刻  $x_k$  を  $i_k$  に固定して行く。それ以上固定するとアクティブスケジュールが得られなくなるまでこの操作を繰り返す。この操作によって得られるジョブの系列を

$k_1 \dots k_r$  とし、固定されたその開始時刻を各々  $t_{k_1} \dots t_{k_r}$  とする。このとき、次のような  $r+1$  個の互いに排反な部分問題  $s$  ( $1 \leq s \leq r+1$ ) が生成できる。

[部分問題  $s$ ] ( $1 \leq s \leq r+1$ ) ジョブ  $k_i$  ( $1 \leq i \leq s-1$ ) の開始時刻を

$$x_{k_i} = t_{k_i} \quad (1 \leq i \leq s-1) \quad (2.67)$$

と固定し、

$$x_{k_s} \neq t_{k_s} \quad (2.68)$$

としてできる問題。

特に、部分問題  $r+1$  は、式 (2.67) のみで定義される。

一方、部分問題  $s$  で開始時刻を固定されないジョブの開始時刻  $x_k^s$  は、

$$x_k^s \in \{t \mid a_k^s \leq t \leq b_k^s, t \in T_{s,k}\} \quad (2.69)$$

によって与えられる。但し、 $a_k^s$  及び  $b_k^s$  は定理 2.3, 定理 2.4 (又は系 2.1) から得られるリスト  $S(k)$ ,  $P(k)$  を用いて、

$$a_k^s \equiv \max \{t_{k_i} \mid k_i \in S(k), 1 \leq i \leq s-1\} \quad (2.70)$$

$$b_k^s \equiv \min \{t_{k_i} \mid k_i \in P(k), 1 \leq i \leq s-1\} \quad (2.71)$$

によって与えられる。一方、 $T_{s,k}$  は、式 (2.67) の結果得られる時刻  $\tau$  におけるジョブの数を  $G_\tau^s(\tilde{x})$  として、次式で与えられる。

$$T_{s,k} \equiv \{t \mid t \leq \tau \leq t + \tau_k - 1 \text{ に対して } G_\tau^s(\tilde{x}) = m\} \quad (2.72)$$

又、各部分問題は番号の小さい順に、従って、最後に生成されたものから順に解かれるものとする。

《限定規則》 次の (i) ~ (iii) で示される限定規則を用いる。

(i) 部分問題が生成されたとき、定理 2.3, 定理 2.4 (又は系 2.1) の条件を満たさないジョブが存在するならば、その部分問題は取除かれる。

(ii) 各部分問題に対して下限  $L(\tilde{u})$  を求める途中で、現在までに見つかったアクティブスケジュールの最小値  $s^*$  に対して、 $s^* \leq L(\tilde{u})$  となれば、その部分

問題は取除かれる。

(iii) 下限を求める途中で、 $\tilde{x}$ がアクティブスケジュールとなれば、 $z^*$ の値を $L(\tilde{u})$ に更新し、その部分問題は取除かれる。

## 2.5 重み付き平均滞留時間最小化問題に適用した実験結果

### 2.5.1 実験結果

本節では、2.4節で述べた解法をWMT問題に適用した実験結果を示す。

ジョブの数 $n$ 、プロセッサの数 $m$ をパラメータとして適当な数の問題を生成し、各問題での $\tau_k$ 及び $c_k$  ( $1 \leq k \leq n$ )は乱数を用いてランダムに与えた。実験に当たって取った具体的な処理の幾つかは次のとおりである。

(i)  $c_k/\tau_k$ の非増加順にジョブを並べ換え、リストスケジューリングを行なう。これによって得られる解の値を初期解とする。

(ii)  $L(\tilde{u})$ の値を改善する $\tilde{u}$ を、

$$\left. \begin{aligned} \tilde{u}^1 &= \tilde{0} \\ \tilde{u}^{l+1} &= \tilde{u}^l + t_w \frac{(Z^* - L(\tilde{u}^l))}{\|\tilde{\phi}(\tilde{x}^l)\|^2} \tilde{\phi}(\tilde{x}^l) \end{aligned} \right\} \quad (2.73)$$

によって求める。但し、 $z^*$ は現在までに見つかったアクティブスケジュールの最小値を表わす。

式(2.62)を見れば、

$$t_w (z^* - L(\tilde{u}^l)) \leq 2(L(\tilde{u}^*) - L(\tilde{u}^l)) \quad (2.74)$$

でなければならないので、 $L(\tilde{u}^l)$ の値が $z^*$ に近づくにつれて $t_w$ の値を減少させることが必要である。本実験では、5回の繰り返しごとに $L(\tilde{u})$ の最大値 $L_0(\tilde{u})$ を求め、次の5回の繰り返しで $L_0(\tilde{u})$ の値が増加しない場合、 $t_w$ を $t_w/2$ とおき、式(2.73)の下式を繰り返す。この操作を適当な回数繰り返すことによって得られる $L_0(\tilde{u})$ を分枝限定法における下限として用いた。

(iii) 各部分問題を生成する段階で、開始時刻が固定されていないジョブを



$c_k / \tau_k$  の非増加順にリストスケジューリングすることによって、最適スケジュールをヒューリスティックに求める方法を加えた。

表 2.1 に実験結果を示す。

表 2.1 実験結果

問題 番号	$n$	$m$	$T$	$T_1$	初期解	下限値 $LB$	最適解 $z^*$	$LB$ に要す 計算時間 (単位秒)	$z^*$ に要す 計算時間 (単位秒)
1	20	7	13	4	308	302	302	1.00	4
2	20	7	14	5	411	401	401	0.99	6
3	20	7	12	3	229	217	218	0.59	5
4	20	7	13	4	293	284	285	0.77	9
5	20	4	18	10	713	706	707	2.86	74
6	20	4	20	12	956	948	948	4.78	22
7	20	4	16	8	524	520	522	2.26	14
8	20	4	18	10	696	687	688	3.61	87
9	30	10	13	4	488	480	482	0.65	77
10	30	10	13	4	461	452	453	2.25	22
11	30	10	16	5	648	640	641	3.35	143
12	30	10	13	4	418	402	404	1.59	10
13	30	6	19	10	1,083	1,081	1,082	5.45	33
14	30	6	19	10	963	957	959	5.69	17
15	30	6	19	10	1,097	1,086	1,086	6.42	379
16	30	6	19	10	1,083	1,073	1,074	6.18	462

- (1) 各ジョブの処理時間  $\tau_k$  は 1 を下限, 5 を上限とする整数集合上の一様分布としてふった。
- (2)  $r_k$  を 1.5 を下限, 3.5 を上限とする実数集合上の一様分布とし,  $c_k = \lceil \tau_k \cdot r_k \rceil$  によって重みを与えた。
- (3) 評価関数の値は  $\sum c_k f_k$  ではなく,  $\sum c_k (x_k - 1)$  を用いている。
- (4) 下限値 ( $LB$ ) は, 原問題に対して得られる下限  $L(\tilde{u})$  の値を表わす。
- (5) 計算時間は, 問題 1~7 に対しては *NEAC 2200 700* システムを用いたときの値であり, 問題 8~16 に対しては, *MELCOM 70* を用いて得られる結果を前者の計算機に換算した値である。

表 2.1 の結果から見て, 本章で提案した解法は, 同じ  $n$  及び  $\sum \tau_k$  をもつ問題の中でも,  $m$  が大きい問題に対して有効であるといえる。又, 山登り法によって得られる下限値は, 最適解  $z^*$  に対する十分良い近似値となることが分かる。このことから,  $z^*$  の値のみ必要で  $\tilde{x}^*$  を知る必要のない場合, 例えば, 近似解法を評価する場合など, 本解法によって得られる下限値を  $z^*$  の代わりに用い

ることが十分可能である．近傍探索法によって得られる解の値を，この値を用いて評価した結果が文献(35)，(36)に示されている．

### 2.5.2 Fisherによる解法との比較

本章で提案した解法を，文献(4)による解法と実験的に比較してみたのが表2.2である．表2.2に示すように，実験的したすべての問題に対して，本解法は，文献(4)による解法より数倍から10数倍速い時間で最適スケジュールを求めることができた．これは，下限を求める計算時間の差異によって説明できる．すなわち，文献(4)では，式(2.35)，(2.36)の双対問題をシンプレックス法を用いて解くことから， $u^l$  から  $u^{l+1}$  を求めるために  $O(nT(n+T))$  の計算時間を要すが，一方，本解法では， $O(nT)$  の計算時間で済む．

表2.2 文献(4)による解法と本解法を比較する実験結果

問題 番号	最適解 $z^*$	文献(4)による解法			本解法		
		下限値 LB	LBに要す 計算時間 (単位秒)	$z^*$ に要す 計算時間 (単位秒)	下限値 LB	LBに要す 計算時間 (単位秒)	$z^*$ に要す 計算時間 (単位秒)
1	302	302	17	40	302	1.00	4
2	401	401	23	102	401	0.99	6
3	218	218	20	42	217	0.59	5
4	285	285	12	23	284	0.77	9
5	707	707	88	231	706	2.86	74
6	948	948	83	479	948	4.78	22
7	522	522	38	95	520	2.26	14

- (1) 各問題は表2.1における問題1～7に対応している．
- (2) 計算機はNEAC 2200 700システムを用いた．

なお，表2.2のすべての問題に対して，文献(4)の解法による下限値が常に  $z^*$  の値に一致したのは興味深い．この下限値が  $z^*$  と一致することを保証するために，表2.2の実験では分枝限定法を用いたが，分枝限定法に依らないもっとも有効な方法があるかもしれない．但し，そのような方法が見つかった場合でも，本解法より速い時間で  $z^*$  を得ることはできないことが表2.2よりいえる．

又，分枝限定法において，各部分問題を最後に生成されたものから順に解いて行くとき，提案した解法に要すメモリスペースは  $O(nT)$  であり，文献(4)の解法では  $O(nT(n+T))$  である．従って，本解法はこの点でも優れていることが

分かる。

## 2.6 結 言

本章で提案した解法は、特に  $m$  が大きい場合、 $G S C H$  問題に対して有効な厳密解法となることが分った。本解法では、記算機メモリの都合上、分枝限定法の分枝規則として、最後に生成された部分問題から先に解いて行く方法を用いたが、最小の下限值をもつ部分問題から先に解いて行く方法を用いれば、解法は更に効率良いものとなろう。

又、提案した山登り法は、極めて短時間に最適解  $z^*$  に対する下限値を求め、しかもこの値は  $z^*$  に十分近い値をもつ。このことから、 $z^*$  の値のみが必要な場合、例えば、近似解法を評価する場合などにも、この山登り法は有効である。

ただ、本解法を用いた場合、同じ  $n$  及び  $\sum \tau_k$  をもつ問題でも  $m$  の小さい程、難しい問題となる。これを解決するためには  $G S C H$  問題の性質を利用したもっと別な定式化（スケジュール長  $T$  に依らない定式化）を考えることが必要である。特に、 $W M F T$  問題に対して、別な定式化を用いた解法が *Sahni*<sup>(19)</sup> によって提案されており、この解法は逆に  $m$  が小さい問題に対して有効である。実験的には、 $m \geq 4$  の場合に、本解法が文献(19)の解法より優れていることを確かめている<sup>(36)</sup>。

又、ジョブの間に先行関係が存在する問題やリソース条件をもつ問題に対してラグランジュ乗数法を適用するとき、ラグランジュ関数の値を原問題の最適解の値に近づけるラグランジュ乗数を求めるためには、現在までのところ、シンプレックス法を用いらざるを得ない。その意味で、提案した山登り法は、問題の性質を有効に利用した手法ともいえるが、この種の問題に対してシンプレックス法に代わる有効な手法を考える上でも、1つの手がかりを与えるものと考えている。

### 第3章 重み付き平均滞留時間最小化問題に対する リストスケジューリングの最悪の場合の評価

#### 3.1 緒言

処理時間  $\tau_k$  及び単位時間当りのコスト  $c_k$  によって規定される独立なジョブ集合  $J \equiv \{1 \cdots n\}$  を,  $m$  台のプロセッサで処理するとし, 重み付き平均滞留時間  $\sum_{k \in J} c_k f_k$  を最小にするスケジュールを求める問題を考える. 但し,  $f_k$  ( $k=1 \cdots n$ ) は, ジョブ  $k$  の完了時刻を表わす. この問題に対して, *McNaughton*<sup>(5)</sup> は,  $c_k/\tau_k$  の非増加順にジョブに優先順位を与えるリストスケジューリングを提案し,  $m=1$  のとき, この解法によって最適スケジュールが得られることを示している.

又,  $m \geq 2$  の問題に対する *McNaughton* の解法の最悪誤差が,  $1/2(1-1/m)$  以下におさえられることは, 既に, 式 (1.5) に示した.

本章では, 解法の時間計算量に影響を与えない1つの操作を *McNaughton* の解法に加えた解法を提案し,  $\sum_{k \in J} \tau_k/m$  を越える処理時間をもつジョブが存在しないとき, この解法の最悪誤差が  $1/4$  以下におさえられることを示す.

又, *McNaughton* の解法の最悪誤差を, ジョブの処理時間を用いて評価し, その上限値が次式で与えられることを示す.

$$|J_I| \leq m \text{ ならば } \frac{1}{4} (m \cdot \max_{k \in J} \tau_k / \sum_{k \in J} \tau_k)^2$$

$$|J_I| > m \text{ ならば}$$

$$\frac{1}{4} \left\{ m \cdot \max_{k \in J} (\max_{k \in J} \tau_k / \sum_{k \in J} \tau_k, \max_{k \in J_I} \tau_k / \sum_{k \in J_I} \tau_k) \right\}^2$$

但し,  $J_I$  は,  $J$  において最大の  $c_k/\tau_k$  をもつジョブ集合を表わす.

3.3節で, 任意の  $c_k$  をもつジョブ集合に対して上記の結果を証明するが, そのための準備として, 3.2節では,  $c_k/\tau_k$  の値がすべて等しいジョブについて考察する.

3.2 各ジョブの単位時間当りのコスト対処理時間比がすべて等しい場合  
まず、次の定義を行なう。

〔定義3.1〕 3.1節で述べた問題に対する最適スケジュールは、リスト  
スケジュールの中に含まれる。(定理2.1参照)本章では以下、リストス  
ケジュールを単にスケジュールと呼ぶ。

〔定義3.2〕 スケジュール $S$ による重み付き平均滞留時間 $Z(S) = \sum_{k \in J} c_k f_k$   
 $c_k f_k$ に対して、

$$U(S) \equiv Z(S) - \frac{1}{2} \sum_{k \in J} c_k \tau_k \quad (3.1)$$

を定義する。以下、この値を $U$ の値と呼ぶ。

本節では、 $J$ において $r_k \equiv c_k / \tau_k$ の値がすべて等しい場合について考察し、  
任意のスケジュール $S$ と最適スケジュール $S^*$ の間に、

$$\frac{U(S) - U(S^*)}{U(S^*)} \leq \frac{1}{4} (m \cdot \max_{k \in J} \tau_k / \sum_{k \in J} \tau_k)^2 \quad (3.2)$$

の関係が成り立つことを示す。又、処理時間による1つの優先順序を用いて得  
られるスケジュール $S'$ と $S^*$ の間に、

$$\frac{U(S') - U(S^*)}{U(S^*)} \leq \frac{1}{4} \quad (3.3)$$

の関係が成り立つことを示す。

### 3.2.1 任意のスケジューリング

$J$ において $r_k \equiv c_k / \tau_k$ の値がすべて等しいとき、任意のスケジュール $S$ に  
よる重み付き平均滞留時間 $Z(S)$ は、定理3.1によって与えられることが文  
献(26)に示されている。

〔定理3.1〕  $r_k = a$  ( $k=1 \dots n$ )であるジョブ集合を $m$ 台のプロセッサ  
で処理するとき、任意のスケジュール $S$ に対して、

$$Z(S) = \frac{1}{2} \alpha \sum_{i=1}^m u_i^2 + \frac{1}{2} Z_n \quad (3.4)$$

が成り立つ。但し、 $u_i (i=1 \dots m)$  は  $S$  による各プロセッサの終了時刻を表わし、

$$Z_n \equiv \sum_{k=1}^n c_k \tau_k = \alpha \sum_{k=1}^n \tau_k^2 \quad (3.5)$$

である。

(定理終)

式 (3.4) から、任意のスケジュール  $S$  に対する  $U$  の値は、

$$U(S) = \frac{1}{2} \alpha \sum_{i=1}^m u_i^2 \quad (3.6)$$

で与えられる。

又、 $S$  による各プロセッサの終了時刻に対して、次の定義を行う。

〔定義 3.3〕 各プロセッサの終了時刻  $u_i (i=1 \dots m)$  に対して

$$b \equiv \sum_{k=1}^n \tau_k / m \quad (3.7)$$

を用い、

$$\epsilon_i \equiv u_i - b \quad (3.8)$$

なる  $\epsilon_i (i=1 \dots m)$  を定義する。

$\sum_{i=1}^m u_i = mb$  であることから、 $\epsilon_i$  の間には

$$\sum_{i=1}^m \epsilon_i = 0 \quad (3.9)$$

の関係が成り立つ。

この  $\epsilon_i$  に対して、 $\sum_{i=1}^m \epsilon_i^2 \leq m (\max_{k \in J} \tau_k)^2 / 4$  が成り立つことを定理 3.2 に示すが、まず、この定理を証明するために必要な補題を述べておく。

〔補題 3.1〕 変数  $\epsilon_i (i=1 \dots m)$  が次の (1), (2) の条件を満たすものとする。

$$(1) \quad 0 \leq \varepsilon_i \leq a \quad (i=1 \cdots m)$$

$$(2) \quad \sum_{i=1}^m \varepsilon_i = q_m a + \gamma_m$$

但し、 $q_m$  は  $q_m < m$  なる整数であり、 $0 \leq \gamma_m < a$  である。

このとき、すべての自然数  $m$  に対して

$$\max \sum_{i=1}^m \varepsilon_i^2 = q_m a^2 + \gamma_m^2 \quad (3.10)$$

となる。

(証明) まず、 $\sum_{i=1}^m \varepsilon_i^2 \leq q_m a^2 + \gamma_m^2$  を証明する。これが  $m$  の場合に成り立つと仮定して、

$$\sum_{i=1}^{m+1} \varepsilon_i = q_{m+1} a + \gamma_{m+1} \quad (3.11)$$

( $q_{m+1}$  は  $q_{m+1} < m+1$  なる整数、 $0 \leq \gamma_{m+1} < a$ )

のとき、

$$\sum_{i=1}^{m+1} \varepsilon_i^2 \leq q_{m+1} a^2 + \gamma_{m+1}^2 \quad (3.12)$$

となることを示す。これを、次の (i)、(ii) の場合に分けて考える。

(i)  $\gamma_{m+1} < \varepsilon_{m+1} \leq a$  のとき、

式 (3.11) より

$$\sum_{i=1}^m \varepsilon_i = (q_{m+1} - 1)a + (\gamma_{m+1} + a - \varepsilon_{m+1}) \quad (3.13)$$

が得られ、 $q_{m+1} - 1 < m$  であることから、 $m$  の場合の仮定を用いて、次式が成り立つ。

$$\begin{aligned} \sum_{i=1}^{m+1} \varepsilon_i^2 &\leq (q_{m+1} - 1)a^2 + (\gamma_{m+1} + a - \varepsilon_{m+1})^2 + \varepsilon_{m+1}^2 \\ &= q_{m+1} a^2 + \gamma_{m+1}^2 + 2(\varepsilon_{m+1} - a)(\varepsilon_{m+1} - \gamma_{m+1}) \end{aligned} \quad (3.14)$$

第3項は非正であることから、式 (3.12) が成り立つ。

(ii)  $0 \leq \varepsilon_{m+1} \leq \gamma_{m+1}$  のとき、

式 (3.11) より

$$\sum_{i=1}^m \varepsilon_i = q_{m+1} a + (\gamma_{m+1} - \varepsilon_{m+1}) \quad (3.15)$$

が得られ、 $q_{m+1} = m$  のとき  $\sum_{i=1}^m \varepsilon_i^2 = ma^2$  であることから、

$$\sum_{i=1}^{m+1} \varepsilon_i^2 \leq ma^2 + \varepsilon_{m+1}^2 \leq q_{m+1} a^2 + \gamma_{m+1}^2 \quad (3.16)$$

がいえる。一方、 $q_{m+1} < m$  のとき、 $m$  の場合の仮定より

$$\sum_{i=1}^m \varepsilon_i^2 \leq q_{m+1} a^2 + (\gamma_{m+1} - \varepsilon_{m+1})^2 \quad (3.17)$$

である。従って、

$$\begin{aligned} \sum_{i=1}^{m+1} \varepsilon_i^2 &\leq q_{m+1} a^2 + (\gamma_{m+1} - \varepsilon_{m+1})^2 + \varepsilon_{m+1}^2 \\ &= q_{m+1} a^2 + \gamma_{m+1}^2 + 2\varepsilon_{m+1}(\varepsilon_{m+1} - \gamma_{m+1}) \end{aligned} \quad (3.18)$$

となり、第3項は非正であることから、式(3.12)が成り立つ。

以上(i)、(ii)により、すべての自然数 $m$ に対して、 $\sum_{i=1}^m \varepsilon_i^2 \leq q_m a^2 + \gamma_m^2$ が成り立つ。

一方、

$$\varepsilon_i = \begin{cases} a & (1 \leq i \leq q_m) \\ \gamma_m & (i = q_m + 1) \\ 0 & (q_m + 2 \leq i \leq m) \end{cases} \quad (3.19)$$

とおくことによって、 $\sum_{i=1}^m \varepsilon_i^2 = q_m a^2 + \gamma_m^2$ が得られる。

(証明終)

[補題3.2]  $0 \leq \varepsilon_i \leq a$ なる変数 $\varepsilon_i (i=1 \dots m)$ において、 $\sum_{i=1}^m \varepsilon_i = t_1$ なる条件の下での $\sum_{i=1}^m \varepsilon_i^2$ の最大値 $\Delta 1$ と、 $\sum_{i=1}^m \varepsilon_i = t_2$ なる条件の下での $\sum_{i=1}^m \varepsilon_i^2$ の最大値 $\Delta 2$ の間には、

$$t_1 > t_2 \text{ ならば } \Delta 1 > \Delta 2 \quad (3.20)$$



の関係が成り立つ。

(証明)  $q_l = \lfloor t_l/a \rfloor$  ( $l=1, 2$ )とおき  $t_l = q_l a + r_l$  とすると, 補題 3.1 を適用して  $d_l = q_l a^2 + r_l^2$  ( $l=1, 2$ ) が得られる。

(i)  $q_1 = q_2$  ならば, 仮定より  $r_1 > r_2$  であるから,  $d_1 > d_2$  がいえる。

(ii)  $q_1 > q_2$  ならば,

$$d_2 \leq (q_1 - 1)a^2 + r_1^2 \leq q_1 a^2 < d_1 \quad (3.21)$$

から,  $d_1 > d_2$  が導ける。

(証明終)

[定理 3.2] 任意のスケジュール  $S$  による各プロセッサの終了時刻を  $u_i = b + \varepsilon_i$  ( $i=1 \dots m$ ) とするとき,

$$\sum_{i=1}^m \varepsilon_i^2 \leq \frac{m}{4} \tau^2 \quad (3.22)$$

が成り立つ。但し,  $\tau = \max_{k \in J} \tau_k$  である。

(証明) 定義 3.1 によって,  $S$  による各プロセッサの終了時刻の間には

$$|u_i - u_j| \leq \tau \quad (i, j=1 \dots m) \quad (3.23)$$

の関係が成り立つ。これに  $\varepsilon_i = u_i - b$  を代入することによって,

$$|\varepsilon_i - \varepsilon_j| \leq \tau \quad (i, j=1 \dots m) \quad (3.24)$$

が得られる。一方, 各  $\varepsilon_i$  の間には, 式 (3.9) に示した関係が成り立つ。

$$\sum_{i=1}^m \varepsilon_i = 0 \quad (3.25)$$

式 (3.24), (3.25) の条件の下で,  $\sum_{i=1}^m \varepsilon_i^2$  の最大値は,  $m$  の偶数, 奇数に応じて

(A)  $m$  が偶数の場合;  $m$  個の  $\varepsilon_i$  のうち,  $m/2$  個を  $\varepsilon_i = m\tau/2$  とし, 残り

の  $m/2$  個を  $\varepsilon_i = -m\tau/2$  とするとき,

(B)  $m$  が奇数の場合 ;  $m$  個の  $\varepsilon_i$  のうち,  $(m-1)/2$  個を  $\varepsilon_i = (m+1)\tau/2$

残りの  $(m+1)/2$  個を  $\varepsilon_i = -(m-1)\tau/2$  とするとき,

に得られることになるが, これをまず,  $m$  が偶数の場合について示す.

$S$  による  $\varepsilon_i$  ( $i=1 \dots m$ ) を, その正負によって 2 つの集合に分け,  $I^+ \equiv \{i \mid \varepsilon_i > 0\}$  及び  $I^- \equiv \{i \mid \varepsilon_i \leq 0\}$  とする. 又,  $|I^+| = m/2 - q$  と仮定する. ここで  $q$  の値としては, 式 (3.24), (3.25) 及び関数  $\sum_{i=1}^m \varepsilon_i^2$  の形から,  $0 \leq q \leq m/2$  を考えれば十分である. 更に,  $\varepsilon_1 \geq \dots \geq \varepsilon_m$  と仮定しても一般性を失わない. この仮定の下で, 次のことがいえる.

——  $|I^+| = m/2 - q$  ( $m$ ; 偶数) のとき

$$\sum_{i=1}^m \varepsilon_i^2 \leq (m/4 - q^2/m) \tau^2 \quad (3.26)$$

であり,

$$\varepsilon_i = \begin{cases} (1/2 + q/m) \tau & (1 \leq i \leq m/2 - q) \\ (-1/2 + q/m) \tau & (m/2 - q + 1 \leq i \leq m) \end{cases} \quad (3.27)$$

のとき, 式 (3.26) において等号が成り立つ. ——

まず, これを証明する. 記述を容易にするため, 時刻を  $\tau$  で正規化して扱い,

$$\varepsilon^+ \equiv \frac{1}{2} + \frac{q}{m}, \quad \varepsilon^- \equiv -\frac{1}{2} + \frac{q}{m} \quad (3.28)$$

とおく. このとき  $\forall i \in I^+, \forall j \in I^-$  に対して  $\varepsilon_i - \varepsilon_j \leq 1 = \varepsilon^+ - \varepsilon^-$  の関係が成り立つことから,  $\varepsilon_i > \varepsilon^+$  である  $i \in I^+$  及び  $\varepsilon_j < \varepsilon^-$  である  $j \in I^-$  が同時に存在することはない. そこで次の (i), (ii), (iii) の場合に分けて考える.

(i)  $0 < \varepsilon_i \leq \varepsilon^+ (\forall i \in I^+)$  且つ  $\varepsilon^- \leq \varepsilon_j \leq 0 (\forall j \in I^-)$  が成り立つ場合.

明らかに

$$\varepsilon_i = \begin{cases} \varepsilon^+ (\forall i \in I^+) \\ \varepsilon^- (\forall i \in I^-) \end{cases} \quad (3.29)$$

とおくことによつて  $\sum_{i=1}^m \varepsilon_i^2$  を最大にできる。この最大値を  $v_1$  とおけば、式 (3.28) を用いて、

$$v_1 = m \varepsilon^+ (-\varepsilon^-) \quad (3.30)$$

となる。

(ii)  $\varepsilon_i > \varepsilon^+$  である  $i$  が存在する場合。

これらの  $\varepsilon_i$  の中で最大のものを  $\varepsilon_s$  とし  $\varepsilon_s = \varepsilon^+ + \delta$  ( $\delta > 0$ ) とおく。このとき  $|\varepsilon_s - \varepsilon_i| \leq 1$  ( $\forall i \in I^-$ ) でなければならないから、 $\varepsilon^+ - \varepsilon^- = 1$  であることを用いて  $\varepsilon_i \geq \varepsilon^- + \delta$  ( $\forall i \in I^-$ ) となる。

まず、 $\varepsilon_i = \varepsilon^- + \delta$  ( $\forall i \in I^-$ ) である場合には、次式が得られる。

$$\sum_{i \in I^-} |\varepsilon_i| = (m/2 + q)(-\varepsilon^- - \delta) \quad (3.31)$$

一方、ある  $r \in I^-$  に対して  $\varepsilon_r > \varepsilon^- + \delta$  と仮定すれば、 $|\varepsilon_r| < -\varepsilon^- - \delta$  であることから  $\sum_{i \in I^-} |\varepsilon_i|$  の値は、式 (3.31) に示した値よりも減少する。従つて、補題 3.2 の結果を用いて

$$\sum_{i=1}^m \varepsilon_i^2 = \sum_{i \in I^+} \varepsilon_i^2 + \sum_{i \in I^-} \varepsilon_i^2 \quad (3.32)$$

の値は  $\varepsilon_i = \varepsilon^- + \delta$  ( $\forall i \in I^-$ ) で最大となる。このとき、式 (3.31) から次式が得られる。

$$\sum_{i \in I^+} \varepsilon_i = -m\varepsilon^+ (\varepsilon^- + \delta) \quad (3.33)$$

式 (3.33) 及び  $0 < \varepsilon_i \leq \varepsilon^+ + \delta$  ( $\forall i \in I^+$ ) の条件下では、

$$\sum_{i \in I^+} \varepsilon_i^2 \leq -m\varepsilon^+ (\varepsilon^- + \delta)(\varepsilon^+ + \delta) \quad (3.34)$$

が成り立つ。一方、 $\varepsilon_i = \varepsilon^- + \delta$  ( $\forall i \in I^-$ ) であることから、

$$\sum_{i \in I^-} \varepsilon_i^2 = \left(\frac{m}{2} + q\right) (\varepsilon^- + \delta)^2 = m\varepsilon^+ (\varepsilon^- + \delta)^2 \quad (3.35)$$

であり、式 (3.34), (3.35) によつて、

$$\sum_{i=1}^m \varepsilon_i^2 \leq m\varepsilon^- (-\varepsilon^- - \delta) \quad (3.36)$$

が得られる。

式(3.36)は(ii)の条件の下での $\sum_{i=1}^m \varepsilon_i^2$ の上限値を与えているが、いま、この右辺を $v_2$ とおき、式(3.30)に示された $v_1$ の値と比較してみると、

$$\varepsilon^+ > 0, \quad -\varepsilon^- > -\varepsilon^- - \delta > 0$$

であることから

$$v_1 > v_2 \tag{3.37}$$

が成り立つ。このことから、(ii)の条件の下での $\sum_{i=1}^m \varepsilon_i^2$ の最大値は、(i)の条件の下での最大値より減少することが分かる。

(iii)  $\varepsilon_i < \varepsilon^-$ である $i \in I^-$ が存在する場合。

(ii)の場合と同様な議論によって、 $\sum_{i=1}^m \varepsilon_i^2$ の最大値は、(i)の条件の下での最大値より減少することが分かる。

以上の(i)、(ii)、(iii)より、式(3.26)が成り立つこと、及び、式(3.27)に示された $\varepsilon_i$ によって上限値が得られることが導けた。

更に、式(3.26)及び $0 \leq q \leq m/2$ から、 $m$ が偶数である場合、

$$\sum_{i=1}^m \varepsilon_i^2 \leq m\tau^2/4 \tag{3.38}$$

となり、(A)によって上限値が得られることが分かる。

$m$ が奇数である場合についても同様に、

$$\sum_{i=1}^m \varepsilon_i^2 \leq (m^2 - 1)\tau^2/4m \tag{3.39}$$

となり、(B)によって上限値が得られることが導け、 $m$ の任意の値に対して式(3.22)が成り立つことが証明できる。

(証明終)

更に、定理3.1、定理3.2を用いて次の定理が成り立つ。

〔定理3.3〕  $r_k = a$  ( $k=1 \dots n$ )であるジョブ集合に対する任意のスケジュール及び最適スケジュールを、各々、 $S$ 及び $S^*$ とする。このとき、 $U$ の値について次式が成り立つ。

$$\frac{U(S) - U(S^*)}{U(S^*)} \leq \frac{1}{4} \left( \frac{\tau}{b} \right)^2 \quad (3.40)$$

但し,

$$\tau \equiv \max \tau_k, \quad b \equiv \sum_{k=1}^n \tau_k / m$$

である.

(証明)  $S$  による各プロセッサの終了時刻を  $u_i \equiv \epsilon_i + b$  ( $i=1 \dots m$ ) とすると, 定理 3.1 及び  $\sum_{i=1}^m \epsilon_i = 0$  を用いて

$$\frac{2U(S)}{\alpha} = \sum_{i=1}^m u_i^2 = m b^2 + \sum_{i=1}^m \epsilon_i^2 \quad (3.41)$$

となり, 更に, 定理 3.2 を用いて

$$\frac{2U(S)}{\alpha} \leq m b^2 \left( 1 + \frac{\tau^2}{4 b^2} \right) \quad (3.42)$$

が得られる. 一方,  $S^*$  による各プロセッサの終了時刻を  $u_i^*$  ( $i=1 \dots m$ ) とすれば, 式 (3.41) より,

$$\frac{2U(S^*)}{\alpha} = \sum_{i=1}^m (u_i^*)^2 \geq m b^2 \quad (3.43)$$

が成り立つ. 式 (3.42), (3.43) より式 (3.40) が得られる.

(証明終)

定理 3.3 から, 次の系が容易に導ける.

[系 3.1]  $r_k = \alpha$  ( $k=1 \dots n$ ) であるジョブ集合において, すべてのジョブの処理時間が  $\sum_{k=1}^m \tau_k / m$  以下であるとき, 任意のスケジュール  $S$  と最適スケジュール  $S^*$  の間には, 次式が成り立つ.

$$\frac{U(S) - U(S^*)}{U(S^*)} \leq \frac{1}{4} \quad (3.44)$$

(系 終)

### 3.2.2 処理時間による優先順序を用いたスケジューリング

前項系1では、 $\tau_k \leq b \equiv \sum_{k=1}^n \tau_k / m$  ( $k=1 \dots n$ ) を条件とした形で1つの結果を示した。本項では、 $\tau_k > b$  であるジョブ  $k$  が存在する場合を問題にする。

まず、次のスケジューリングを提案する。

《スケジューリング  $A$ 》

ステップ1: (初期設定)  $J_0 = \phi$ ,  $j=1$

ステップ2:  $j \geq 1$  に対して

$$\sigma_j = \sum_{l=0}^{j-1} |J_l|, \quad b_j = \frac{1}{m - \sigma_j} \sum_{k=\sigma_j+1}^n \tau_k$$

とし、

$J_j = \{k | \tau_k > b_j, \sigma_j + 1 \leq k \leq n\}$  とする。  $J_j = \phi$  ならステップ3へ。  $J_j \neq \phi$  ならば  $J_j$  の中のジョブをプロセッサ  $i$  ( $\sigma_j + 1 \leq i \leq \sigma_j + |J_j|$ ) の先頭に割当てる。  $j=j+1$  としてステップ2を繰り返す。

ステップ3: 残りのジョブを任意の順序でスケジュールする。

スケジューリング  $A$  でステップ2の繰り返し数を  $t$  とすると  $\sum_{j=1}^{t-1} |J_j| < m$  であることから、スケジューリング  $A$  の時間計算量は、たかだか  $O(\min(n \log n, nm))$  である。

スケジューリング  $A$  を適用して得られるスケジュールを  $S_A$  で表わすとき、最適スケジュール  $S^*$  との間には、次に示すような関係が成り立つ。

〔定理 3.4〕  $r_k = \alpha$  ( $k=1 \dots n$ ) なるジョブ集合にスケジューリング  $A$  を適用したとき、ステップ2の繰り返し数を  $t$  とし、

$$\sigma_t \equiv \sum_{j=0}^{t-1} |J_j|, \quad J_T \equiv \cup_{j=0}^{t-1} J_j,$$

$$b_t \equiv \frac{1}{m - \sigma_t} \sum_{k=\sigma_t+1}^n \tau_k$$

とおけば、 $S_A$  と  $S^*$  の間には次の関係が成り立つ。

$$\frac{U(S^*) - U(S_A)}{U(S^*)} \leq \frac{1}{4} \left\{ \frac{1}{(m - \sigma_t) b_t^2} \sum_{k \in J_T} \tau_k^2 + 1 \right\}^{-1} \quad (3.45)$$

(証明)  $S_A$  によって最後に処理を開始するジョブ  $n$  の開始時刻  $x_n$  は

$$x_n \leq \frac{1}{m - \sigma_t} \sum_{k=\sigma_t+1}^{n-1} \tau_k < \frac{1}{m - \sigma_t} \sum_{k=\sigma_t+1}^n \tau_k \equiv b_t \quad (3.46)$$

となる。 $J_T$  に属すジョブが処理されるプロセッサ  $i$  ( $1 \leq i \leq |J_T|$ ) の終了時刻  $u_i$  は  $u_i > b_t$  であることから、これらのプロセッサには 1 個のジョブしか割当てられない。これに対して、 $S^*$  においても同じことがいえることを示してみる。

すべてのジョブ  $k$  に対して  $c_k / \tau_k$  の値が同じであることから、同じプロセッサに割当てられたジョブの順序を入れ換えることによって重み付き平均滞留時間の値は変わらない。そこで、 $S^*$  においては、同じプロセッサに割当てられたジョブは処理時間の非増加順に並んでいるものとする。いま、あるプロセッサ  $x$  で  $s \in J_r$  ( $1 \leq r \leq t-1$ ) なるジョブ  $s$  の後に処理されるジョブ  $s'$   $\in \cup_{j=1}^{r-1} J_j$  が存在すると仮定すると、 $\sigma_r = \sum_{j=0}^{r-1} |J_j|$  に対して、 $s \in \{\sigma_r + 1 \dots n\}$  で且つ

$$\frac{1}{m - \sigma_t} \sum_{k=\sigma_r+1}^n \tau_k \equiv b_r < \tau_s \quad (3.47)$$

であることから、プロセッサ  $\sigma_r + 1 \dots m$  の中に  $u_y < b_r$  なる終了時刻をもつ

プロセッサ  $v$  が存在する。この  $v$  に  $s'$  を割当てることによって、より小さい評価関数をもつスケジュールが得られることになる。これは  $S^*$  が最適スケジュールであるという仮定に矛盾し、従って、 $S^*$  においても  $J_T$  に属すジョブはただ 1 つだけプロセッサに割当てられることになる。

結局、 $J_T$  に属すジョブは、 $S_A$  と  $S^*$  の両者において、プロセッサ  $1 \cdots \sigma_i$  に 1 個ずつ割当てられる。そこで残りのジョブ  $k$  ( $k = \sigma_i + 1 \cdots n$ ) を  $m - \sigma_i$  個のプロセッサにスケジュールして得られる  $U$  の値を  $U'(S_A)$ 、最適スケジュールのジョブ  $k$  ( $k = \sigma_i + 1 \cdots n$ ) に対する  $U$  の値を  $U(S^*)$  とすれば、すべての  $k$  ( $k = \sigma_i + 1 \cdots n$ ) に対して

$$\tau_k \leq b_i \equiv \frac{1}{m - \sigma_i} \sum_{k=\sigma_i+1}^n \tau_k \quad (3.48)$$

であることから、ジョブ集合  $\{\sigma_i + 1 \cdots n\}$  に式 (3.42) を適用して

$$\frac{2}{\alpha} U'(S_A) \leq \frac{5}{4} (m - \sigma_i) b_i^2 \quad (3.49)$$

が得られ、これから

$$\frac{2}{\alpha} U(S_A) \leq \sum_{k \in J_T} \tau_k^2 + \frac{5}{4} (m - \sigma_i) b_i^2 \quad (3.50)$$

を得る。一方、ジョブ集合  $\{\sigma_i + 1 \cdots n\}$  に式 (3.43) を適用して

$$\frac{2}{\alpha} U(S^*) \geq (m - \sigma_i) b_i^2 \quad (3.51)$$

が得られ、これから、

$$\frac{2}{\alpha} U(S^*) \geq \sum_{k \in J_T} \tau_k^2 + (m - \sigma_i) b_i^2 \quad (3.52)$$



を得る。式(3.50), (3.52)から, 式(3.45)が導ける。

(証明終)

定理3.4から次の系が容易に導ける。

[系3.2]  $r_k = \alpha$ なるジョブ集合において,  $S_A$ 及び $S^*$ に対する $U$ の値の間には, 式(3.44)と同じ形の式

$$\frac{U(S_A) - U(S^*)}{U(S^*)} \leq \frac{1}{4} \quad (3.53)$$

が成り立つ。

(系 終)

### 3.3 各ジョブの単位時間当りのコスト対処理時間比が異なる場合

本節では, 前節で得た結果を使って, ジョブ集合 $J = \{1 \cdots n\}$ に属すジョブ $k$ の $r_k \equiv c_k / \tau_k$ の値が異なる場合を考察する。これはMcNaughtonの解法によって得られるスケジュールを評価することを意味するが, ここではMcNaughtonの解法に, 更に1つの操作を加えて得られるスケジュールについて評価する。

まず, 次のスケジューリングを考える。

《スケジューリングB》

ステップ1: (初期設定)ジョブ集合 $J = \{1 \cdots n\}$ において $r_1 \geq \cdots \geq r_n$ とする。

ステップ2:  $\alpha_I = \max \{r_k \mid k=1 \cdots n\}$

$J_I = \{k \mid r_k = \alpha_I, k=1 \cdots n\}$ とする。

(i)  $|J_I| \leq m$ ならば $J_I$ の中のジョブを任意の順序でスケジュールする。

(ii)  $|J_I| > m$ ならば $J_I$ の中のジョブに対して3.2.2項で述べたスケジューリングAを実行する。

ステップ3:  $J - J_I$  中のジョブを番号の小さい順にスケジューリングする。

スケジューリング  $B$  におけるステップ2が, *McNaughton* の解法に新たに加えられた操作である。ステップ2の処理に要す時間計算量は, たかだか  $O(|J_I| \min(\log |J_I|, m))$  であることから, スケジューリング  $B$  に要す時間計算量は  $O(n \log n)$  であり, *McNaughton* の解法のそれに等しい。

スケジューリング  $B$  によって得られるスケジュールを  $S_B$  で表わし, 最適スケジュール  $S^*$  との間に成り立つ関係を定理3.5に示す。

〔定理3.5〕  $r_k = c_k / \tau_k$  の値が異なるジョブ集合  $J = \{1 \dots n\}$  において,  $b = \sum_{k=1}^n \tau_k / m$  を越える処理時間をもつジョブが存在しないとき,  $S_B$  と  $S^*$  の間には,

$$\frac{U(S_B) - U(S^*)}{U(S^*)} \leq \frac{1}{4} \quad (3.54)$$

すなわち,

$$U(S_B) \leq 5U(S^*)/4 \quad (3.55)$$

の関係が成り立つ。又, 式(3.54)の右辺の値は最良の上限値である。

(証明)  $J$  に属す, すべての  $k$  ( $k=1 \dots n$ ) について  $r_k$  の値が等しいとき, 式(3.55)が成り立つことは, 式(3.44)に示した。ここでは,  $r_k$  ( $k=1 \dots n$ ) の異なる値の数  $\nu$  に関して帰納法を用いることにより, 定理を証明する。 $\nu = h-1$  のとき式(3.55)が成り立つと仮定して,  $\nu = h$  の場合にも, やはり, 式(3.55)が成り立つことを示す。

さて,  $J$  は, 各ジョブ  $k$  が  $r_k$  ( $k=1 \dots n$ ) の非増加順に並び,  $r_1 = \dots = r_q = \alpha_I$  で,  $r_{q+1} \dots r_n$  の中に  $h-1$  個の異なる組が存在するようなジョブ集合とする。この  $J$  に対して, 各ジョブの単位時間当りのコスト  $c'_k$  が次式で与えられるジョブ集合  $J'$  を考える。

$$\begin{aligned} c'_k &= \varepsilon c_k & (1 \leq k \leq q) \\ c'_k &= c_k & (q+1 \leq k \leq n) \end{aligned} \quad (3.56)$$

但し,

$$\varepsilon \equiv r_{q+1} / r_q < 1 \quad (3.57)$$

である.

$S_B$  及び  $S^*$  の  $J'$  に対する  $U$  の値を各々,  $U'(S_B)$  及び  $U'(S^*)$  で表わす.  $J'$  においては,  $r_k$  の異なる組が  $h-1$  個であるから,  $\nu = h-1$  における仮定を用いて,

$$U'(S_B) \leq 5U'(S^*) / 4 \quad (3.58)$$

が得られる.

ここで  $U'(S_B)$  を, 最初の  $q$  個のジョブに関する量  $x$  と残りの  $n-q$  個のジョブに関する量  $y$  の2つの量に分け

$$U'(S_B) = x + y \quad (3.59)$$

とし, 又,  $U'(S^*)$  についても同様に

$$U'(S^*) = x^* + y^* \quad (3.60)$$

とおく. このとき,  $x$  と  $x^*$  について,

$$x \leq 5x^* / 4 \quad (3.61)$$

が成り立つことが, 次のようにして導ける.

(i)  $m \geq q$  の場合.

$S_B$  においては最初の  $q$  個のジョブがすべてプロセッサの先頭に割り当てられることから,

$$x \leq x^* < 5x^* / 4 \quad (3.62)$$

となる.

(ii)  $m < q$  の場合.

(ii-1) 最初の  $q$  個のジョブ  $k$  がすべて  $\tau_k \leq b_q \equiv \sum_{k=1}^q \tau_k / m$  を満たすと

とき，式 (3.42) を適用して，

$$x \leq \frac{1}{2} \alpha_I \left\{ \frac{5}{4} m b_q^2 \right\} \quad (3.63)$$

が得られる．一方，式 (3.43) を適用して，

$$x^* \geq \alpha_I m b_q^2 / 2 \quad (3.64)$$

であるから， $x \leq 5x^*/4$  が成り立つ．

(ii-2) 最初の  $q$  個のジョブの中に， $\tau_k > b_q$  なるジョブ  $k$  が存在するとき，スケジューリング  $A$  が実行される．スケジューリング  $A$  のステップ 2 の繰り返し数を  $l$  とし，各繰り返しにおいて得られるジョブ集合の系列を  $J_j$  ( $j = 1 \dots l-1$ ) とする．又， $J_T \equiv \bigcup_{j=1}^{l-1} J_j$  とおくと，ジョブ集合  $\{1 \dots q\}$  に対して式 (3.50) を適用して

$$\frac{2}{\alpha_I} x \leq \sum_{k \in J_T} \tau_k^2 + \frac{5}{4} \cdot \frac{(\sum_{k=1}^q \tau_k - \sum_{k \in J_T} \tau_k)^2}{m - |J_T|} \quad (3.65)$$

が得られる．一方，ジョブ集合  $\{1 \dots q\}$  のみのスケジュールを考え，この最適スケジュールを  $x_0$  とすると，明らかに，

$$x^* \geq x_0 \quad (3.66)$$

であり，更に，式 (3.52) を適用して

$$\frac{2}{\alpha_I} x_0 \geq \sum_{k \in J_T} \tau_k^2 + \frac{(\sum_{k=1}^q \tau_k - \sum_{k \in J_T} \tau_k)^2}{m - |J_T|} \quad (3.67)$$

が得られる．式 (3.65) ~ (3.67) から  $x \leq 5x^*/4$  がいえる．

議論をジョブ集合  $J$  に移して， $S_B$  及び  $S^*$  の  $J$  に対する  $U$  の値を，各々， $U(S_B)$ ， $U(S^*)$  とする．式 (3.59)，(3.60) に対応して

$$\left. \begin{aligned} U(S_B) &= x/\varepsilon + \psi \\ U(S^*) &= x/\varepsilon + \psi^* \end{aligned} \right\} \quad (3.68)$$

が得られる。まず、 $y \leq 5y^*/4$ であれば、式(3.61)を用いて式(3.55)がいえ、証明は完了する。

次に、

$$y > 5y^*/4 \quad (3.69)$$

のときには、式(3.61)、(3.69)より

$$x^*y > xy^* \quad (3.70)$$

がいえ、これから $\varepsilon < 1$ であることを用いて

$$(1-\varepsilon)x^*y > (1-\varepsilon)xy^* \quad (3.71)$$

が得られる。更に、式(3.59)、(3.60)、(3.68)、(3.71)を用いて

$$\begin{aligned} \varepsilon U'(S_B)U(S^*) &= (x+y)(x^*+\varepsilon y^*) \\ &> (x^*+y^*)(x+\varepsilon y) = \varepsilon U'(S^*)U(S_B) \end{aligned} \quad (3.72)$$

が導ける。従って、 $\varepsilon > 0$ であることから

$$U'(S_B)U(S^*) > U'(S^*)U(S_B) \quad (3.73)$$

であり、これに式(3.58)を適用して

$$5U'(S^*)U(S^*)/4 \geq U'(S_B)U(S^*) > U'(S^*) > U'(S^*)U(S_B) \quad (3.74)$$

となり、式(3.55)に示した。

$$U(S_B) \leq 5U(S^*)/4 \quad (3.75)$$

が導ける。

又、式(3.54)の上限値は、次の $\tau_k$ 、 $c_k$ によって規定されるジョブ集合 $J = \{1 \cdots 3m\}$ において、 $\delta \rightarrow 0$ のとき得られる。

$$\tau_k = \begin{cases} \tau & (k=1 \cdots 2m) \\ 2\tau & (k=2m+1 \cdots 3m) \end{cases} \quad (3.76)$$

$$c_k = \begin{cases} \tau + \delta & (k=1 \cdots 2m) \\ 2\tau & (k=2m+1 \cdots 3m) \end{cases} \quad (3.77)$$

但し、プロセッサの数  $m$  は偶数であるとする。

(証明終)

更に、定理 3.3 から、定理 3.5 の証明に用いた手順と同様な手順を用いることにより、McNaughton の解法によって得られるスケジュール  $S_M$  と最適スケジュール  $S^*$  の間に成り立つ関係を導くことができる。

〔定理 3.6〕 ジョブ集合  $J = \{1 \cdots n\}$  において、異なる  $r_k$  のうち最大のものを  $\alpha_I$  とし、 $J_I = \{k \mid r_k = \alpha_I, k=1 \cdots n\}$  とする。このとき、 $S_M$  と  $S^*$  の間には次の関係が成り立つ。

(i)  $|J_I| \leq m$  ならば、

$$\frac{U(S_M) - U(S^*)}{U(S^*)} \leq \frac{1}{4} \left( \frac{\tau}{b} \right)^2 \quad (3.78)$$

但し、

$\tau \equiv \max\{\tau_k \mid k \in J\}$  ,  $b \equiv \sum_{k \in J} \tau_k / m$  である。

(ii)  $|J_I| > m$  ならば

$$\frac{U(S_M) - U(S^*)}{U(S^*)} \leq \frac{1}{4} \left\{ \max \left( \frac{\tau}{b}, \frac{\tau_I}{b_I} \right) \right\}^2 \quad (3.79)$$

但し、

$\tau_I \equiv \max\{\tau_k \mid k \in J_I\}$  ,  $b_I \equiv \sum_{k \in J_I} \tau_k / m$  である。

(定理終)

定理 3.5, 定理 3.6 では、 $U$  の値を用いて、 $S_B$  及び  $S_M$  の最悪誤差の上限值を導出した。

$Z(S_B)$  と  $Z(S^*)$  に対しては、式 (3.1) から、

$$\frac{Z(S_B) - Z(S^*)}{Z(S^*)} \leq \frac{1}{4} \left( 1 - \frac{\sum_{k=1}^n c_k \tau_k}{2Z(S^*)} \right) \quad (3.80)$$

が得られる。式 (3.80) は定理 3.5 に示したように最良の上限値であるが、 $Z(S^*)$  や  $\sum_{k=1}^n c_k \tau_k$  の値に依らない最良の上限値を求めることは 今後に残された課題である。ただ、式 (3.80) の右辺の値が  $1/6$  となる例の存在は確かめている。

又、 $Z(S_M)$  に対しても、式 (3.78), (3.79) の右辺に  $1 - \sum_{k=1}^n c_k \tau_k / 2Z(S^*)$  を乗じた形で、最悪誤差の上限値が与えられる。式 (3.78), (3.79) より、 $|J_I| \leq m$  で且つ  $\tau/b$  の値が 1 に比べて十分小さい場合、又は、 $|J_I| > m$  で且つ  $\max(\tau/b, \tau_I/b_I)$  の値が 1 に比べて十分小さい場合、 $U(S_M)$  は  $U(S^*)$  に近い値となり、このことから、 $Z(S_M)$  は  $Z(S^*)$  に対して 良い近似を与えることが分かる。特に、 $m$  に比べて  $n$  が大きく、且つ、 $\tau_k$  ( $k=1 \dots n$ ) が一様に分布するようなジョブ集合に対しては、McNaughton の解法は良い近似解を与えることになり、これは、文献 (36) の実験結果を理論的に裏付ける結果である。文献 (36) では、( $n=50, m=5, 1 \leq \tau_k \leq 10$ ) で最大 0.4%、( $n=50, m=10, 1 \leq \tau_k \leq 20$ ) で最大 1% の誤差内で解が得られることが実験結果として報告されている。一方、式 (3.78) から得られる値は、それぞれ、0.8% 及び 3.6% である。

### 3.4 結 言

本章では、重み付き平均滞留時間最小化問題に対する、McNaughton の解法に 1 つの操作を付け加えた解法を最悪の場合について評価した。(この解法の時間計算量は、McNaughton の解法の時計算量に等しい。) これによって、 $\sum_{k=1}^n \tau_k / m$  を越える処理時間をもつジョブが存在しないという条件付きであるが、文献 (26) の解析から導かれる上限値を  $1/4$  にまで改善することができた。上に述べた条件は、又、すべてのジョブの処理時間が、プロセッサの平

均終了時刻を越えないともいいかえることができ、少なくとも、すべてのプロセッサに 2 個以上のジョブが割当てられる場合には、この条件は満たされることから、この解法は大規模な問題を解くための有効な解法といえる。

又、*McNaughton* の解法の最悪誤差に対して、式 (3.78) 及び式 (3.79) を導出した。これらの式は、各問題ごとに、ジョブの処理時間の値から最悪誤差の上限値を知ることができるという点で有効である。



## 第4章 リソース条件をもつ問題に対する 近似解法の評価

### 4.1 緒言

本章では、ジョブを処理するためにプロセッサ以外の付加的リソースが必要であり、しかもリソースの容量に制限があるモデルを考察する。このモデルにおいては、各時刻に処理されるジョブのリソース要求量の総和が、リソース容量を越えないスケジュールを求めることが必要であり、この条件は、プロセッサ条件に対して、リソース条件と呼ばれる。

リソース条件をもつほとんどのスケジューリング問題は  $NP$  完全な問題となり、更に、次節で述べる特別な場合を除いて、最悪の場合の評価から良さが保証される近似解法も知られていない。

この種の問題に対しては、問題を定義する変数が任意の値をもつ場合に有効な近似解法を見つけることは難しいが、変数がある分布に従う場合に有効な近似解法を見つけることは可能であるかもしれない。問題を定義する変数が統計的情報をもつ場合を仮定する近似解法の評価は、最悪の場合の評価に対して、確率的評価と呼ばれる。<sup>(6)</sup>

4.3 節では、最悪の場合の評価から良さが保証される近似解法が知られていない問題の中で、多数のジョブが複数のリソースを要求するとき、同時に処理できるジョブの数を最大にする問題に対するヒューリスティックな解法を確率的に評価する。この問題に対する解は、リソース条件をもつ一般的なスケジューリング問題<sup>(8)</sup>、プログラム並列処理における変数割当て問題<sup>(9)</sup>、診断時刻表作成問題<sup>(10)</sup>などを解くために応用され、この問題に対して有効な近似解法を考えることは重要である。

### 4.2 特別な問題に対する近似解法の実例の評価

従来、リソース条件をもつスケジューリング問題の中で、最大完了時刻を評価関数とする問題に対して、幾つかの近似解法が提案され、最悪の場合の評価

が行なわれてきた。本節では、これらの近似解法の最悪誤差の上限値を示す。

まず、以下で用いる記号を次のように定義する。

- $J \equiv \{1 \cdots n\}$  : 独立なジョブからなるジョブ集合
- $m$  : プロセッサの数
- $R \equiv \{R_1 \cdots R_s\}$  : リソースの集合
- $\tau_k$  : ジョブ  $k$  の処理時間
- $r_{kl}$  : ジョブ  $k$  がリソース  $R_l$  を要求する量
- $f_k$  : スケジュールによるジョブ  $k$  の完了時刻
- $B_l$  : リソース  $R_l$  の容量
- $\varepsilon_A$  : 解法  $A$  による最悪誤差

$\max f_k$  を評価関数とする問題に対して、次の (i) ~ (V) の結果が得られている。

以下の記述において、 $\tau_k=1$  ( $k=1 \cdots n$ ) は、すべてのジョブの処理時間が等しいことを意味する。この条件は、又、異なる  $\tau_k$  ( $k=1 \cdots n$ ) をもつジョブ集合において、各ジョブが先取権をもたないという条件に相当する。

又、 $m \geq n$  は、プロセッサ条件が存在しない場合を意味する。すなわち、プロセッサに相当する処理装置（各ジョブを処理するために必ず必要であり、しかも、各時刻で1つのジョブによって専有される装置）が存在しない場合や、リソース条件のみを満たすスケジュールにおいて、各時刻で同時に処理されるジョブの数の最大値が  $m$  以下である場合を意味する。

(i)  $\tau_k$  ( $k=1 \cdots n$ ),  $r_{kl}$  ( $k=1 \cdots n, l=1 \cdots s$ ),  $m, s$  が任意の値をもつ問題に対しては、任意のリストスケジューリング  $L$  によって

$$\varepsilon_L \leq \min \left\{ \frac{m-1}{2}, s+1 - \frac{2s+1}{m} \right\} \quad (4.1)$$

が得られる<sup>(37)</sup>。従って、 $m, s$  の値が小さな値をもつときのみ、この解法は有効である。

(ii)  $m=2$ ,  $\tau_k=1$  ( $k=1 \cdots n$ ) で、 $r_{kl}$  ( $k=1 \cdots n, l=1 \cdots s$ ) 及び  $s$  が任意の値をもつ問題は、グラフの枝被覆問題に定式化することによって、最適スケジュールを求めることができる<sup>(3)</sup>。定式化を含めたこの解法の時間計算量

は、 $O(\max(n^3, ns))$ である。

$m \geq 3$  の場合は、 $s=1$  でも  $NP$  完全な問題となる。ただ、 $m$  が小さな値をもつ問題に対して、式(4.1)の右辺の値を更に改善する近似解法が知られており、これを(iii)に示す。

(iii)  $m \geq 3$ ,  $\tau_k = 1 (k=1 \cdots n)$  で、 $r_{kl} (k=1 \cdots n, l=1 \cdots s)$  及び  $s$  が任意の値をもつ問題は、集合被覆問題に定式化でき、4.3節で述べる解法と類似した解法  $SC$  によって、

$$\epsilon_{sc} \leq \sum_{h=2}^m 1/h \quad (4.2)$$

が得られる<sup>(7)</sup>。この解法の、定式化も含めた時間計算量は  $O(n^m s)$  であり、この解法は、 $m$  が小さな値をもつ場合にのみ有効な解法といえる。

(iv)  $m \geq n$ ,  $\tau_k = 1 (k=1 \cdots n)$  及び  $s=1$  で、 $r_{kl} (k=1 \cdots n)$  が任意の値をもつ問題は、*Bin packing* 問題と呼ばれる。 $(r_{kl} (k=1 \cdots n))$  を容量  $B_l$  のビンにつめるとき、最小限必要なビンの個数は、 $\max f_k$  の最小値と一致する。) この問題に対しては、

$$\epsilon_{FFD} \leq 11/9 + 4/Z^* \quad (4.3)$$

となる解法  $FFD$ <sup>(38)</sup> が提案されており、この解法の時間計算量は  $O(n \log n)$  である。但し、 $Z^*$  は最適解の値を表わす。式(4.3)の上限値は、式(4.1)の右辺に  $s=1$  を代入して得られる値より改善されることが分かる。

(v)  $m \geq n$ ,  $\tau_k = 1 (k=1 \cdots n)$  で  $s$  が任意の値をもつ問題に対しては、 $r_{kl} = 0$  又は  $B_l (k=1 \cdots n, l=1 \cdots s)$  の場合でも、最悪誤差の上限値が問題のサイズ ( $n$  や  $s$ ) に依存しない値で与えられる近似解法は知られていない。ただ、この問題を、4.3節で述べる方法を用いて、グラフの節点彩色問題に定式化することにより、幾つかの近似解法を考えることができる<sup>(7)</sup>。

### 4.3 節点独立集合問題に対する近似解法の確率的評価

#### 4.3.1 問題の定式化と近似解法

本節では、 $m \geq n$ 、 $r_{kl} = 0$  又は  $B_l$  ( $k=1 \dots n, l=1 \dots s$ ) で、 $s$  が任意の値をもつとき、同時に処理できるジョブの数を最大にする問題を考察する。

この問題に対する解は、4.1節で述べたようなスケジューリング問題を解くために応用され、又、スケジューリング問題以外の多くの問題を解くためにも応用される<sup>(39)</sup>。

又、以下に述べる、この問題に対するヒューリスティックな解法は、前節(V)に示した問題を解くために利用でき<sup>(7),(40)</sup>、更に(V)のモデルで、他の評価関数をもつ問題や、 $\tau_k$  ( $k=1 \dots n$ ) が異なる問題への適用も期待できる。(  $\tau_k$  が異なる場合には、あるジョブが処理を完了する時刻ごとに、この解法を適用すればよい。 )

この問題は、次のように、グラフの節点独立集合問題に定式化できる。(この問題に対して、節点独立集合問題への定式化を用いる解法より優れた解法は存在し得ないことを4.3.5項に示す。この意味で2つの問題は同じ問題と考えることができる。)

$X = J$  を節点集合とし

$$\Gamma = \{ (i, j) \mid \text{ある } l (l=1 \dots s) \\ \text{に対して } r_{il} + r_{jl} > B_k \} \quad (4.4)$$

で定義される  $\Gamma$  を枝集合とするグラフ  $G \equiv (X, \Gamma)$  を考える。このとき、 $G$  における  $X$  の部分集合の中で、隣接節点を一對も含まない節点集合  $I$  は、同時に処理できるジョブ集合を表わすことになる。

このような節点集合  $I$  は 節点独立集合 と呼ばれ、節点数最大の節点独立集合を求める問題は、節点独立集合問題 と呼ばれる。この問題は、グラフに関する代表的な問題の1つであり、ふるくから種々の解法が提案されているが、現在までのところ最も優れた解法と考えられている Tarjan-Trojanowski の解法<sup>(41)</sup>でも、その時間計算量は  $O(2^{n/3})$  であり、更に、この問題は NP 完全な問題であることが知られている<sup>(12)</sup>。ただ、特別な構造をもつグラフに対しては、最適

解を求めるための多項式時間解法が存在する<sup>(42),(43),(44)</sup>

又、節点独立集合問題に対して、次のようなヒューリスティックな解法が提案されている<sup>(6)</sup>

節点  $v$  ( $v=1 \cdots n$ ) の隣接節点の集合を  $\Gamma(v)$  で表わす。

<解法 I>

ステップ 1 : (初期設定)

$$I \leftarrow \phi, V \leftarrow \{1 \cdots n\}$$

ステップ 2 :  $V$  から最小番号をもつ節点  $v$  を選び、

$$I \leftarrow I \cup \{v\}, V \leftarrow V - \{v\} - \Gamma(v) \text{ とする。}$$

ステップ 3 :  $V \neq \phi$  なら、ステップ 2 へ。

$$V = \phi \text{ なら、} I \text{ を解とする。}$$

又、解法 I を修正した解法として、次の解法 II, III がある。

<解法 II>

$\Gamma(v)$  の非減少順に節点に番号を付け、解法 I を実行する。

<解法 III>

解法 I のステップ 2 を次のように修正する。

ステップ 2 :  $V$  によって誘導される部分グラフにおいて、最小線度をもつ節

$$点 v \text{ を選び、} I \leftarrow I \cup \{v\}, V \leftarrow V - \{v\} - \Gamma(v) \text{ とする。}$$

上記の解法 II 及び III は、各節点間に各々独立に枝が存在するグラフに対して有効な解法となることが、実験的に確かめられている<sup>(45)</sup>

一方、*Grimmett*<sup>(40)</sup> や *Karp*<sup>(6)</sup> は、解法 I をランダムグラフを用いて確率的に評価し、節点数  $n$  が十分大きなグラフに対して、次の結果を示している。

(i) 各節点間に  $p(n) = p$  (定数) の確率で枝が存在するランダムグラフに対して、

$$\frac{\beta^*}{\beta^I} \leq 2 + \epsilon \quad (4.5)$$

が成り立つ<sup>(40)</sup>。但し、 $\beta^*$  及び  $\beta^I$  は、各々、最適解の上限値及び解法 I によって得られる解の値を表わし、 $\epsilon > 0$  は任意に小さな値を表わす。

(ii) 各節点間に  $p(n) = c/(n-1)$  ( $c$ ; 定数) の確率で枝が存在するランダムグラフに対して,

$$\frac{\beta^*}{\beta^1} \leq (1+\epsilon) \gamma(c) \quad (4.6)$$

が成り立つ<sup>(6)</sup>。但し,  $\gamma(c)$  は  $c$  に関する単調増加関数であり,  $c=1100$  で約 1.6 の値をもつ。

本節では, 上記の解法 II を,  $p(n) = c/(n-1)$  であるランダムグラフ (平均線度一定モデル) を用いて確率的に評価し, この解法によって, 式 (4.6) の右辺の値が更に改善できることを確かめる。又,  $c$  が小さい程, 改善度が大きくなることを確かめる。

#### 4.3.2 平均線度一定モデル

〔定義 4.1〕<sup>(6)</sup> 節点の数が  $n$  で, 各節点間に, それぞれ独立に,  $p(n)$  の確率で枝が存在するグラフの集合を  $n$  節点ランダムグラフ と呼ぶ。特に,  $p(n) = c/(n-1)$  ( $c$ ; 定数) である  $n$  節点ランダムグラフを 平均線度一定モデル と呼び, 定数  $c$  を 平均線度 と呼ぶ。

又, 本節で用いる記号を次のように定義する。

〔記号の定義〕

$G_{n,c}$  : 平均線度  $c$  をもつ  $n$  節点ランダムグラフを表わす。

$P(k; c)$  : 母数  $c$  のポアソン分布の確率密度関数を表わす。すなわち,

$$P(k; c) = c^k \exp(-c) / k! \text{ である。}$$

$o(f(n))$  :  $f(n)$  で割った値が,  $n \rightarrow \infty$  で, 0 に収束する値を表わす。

$\theta(f(n))$  :  $f(n)$  で割った値が,  $n \rightarrow \infty$  で, ある定数  $C$  ( $C \neq 0$ ) に収束する値を表わす。

解法 II を  $G_{n,c}$  に適用して得られる解の値を導出するための準備として, ここでは,  $G_{n,c}$  に属すグラフについて, 次の (i) ~ (iii) が成り立つことを示す。

— 線度  $k$  をもつ節点の集合を  $X_k$ ,  $X_k$  に属す節点の個数を  $|X_k|$  とすると,

$$(i) \quad |X_k| = P(k; c) n + o(n) \quad (4.7)$$

である。

(ii) 任意の  $\varepsilon > 0$  に対して,

$$\sum_{k=l}^u |X_k| \geq (1-\varepsilon)n \quad (4.8)$$

となる有限値<sup>(注)</sup>  $l$  及び  $u$  が存在する。すなわち、 $\varepsilon$  の誤差を許せば、 $k < l$  及び  $k > u$  である線度  $k$  をもつ節点は無視できる。

(iii)  $v \in X_i, w \in X_j$  なる 2 節点  $v, w$  の間に枝が存在する確率  $p_{ij}$  は

$$p_{ij} = ij/c(n-1) + o(1/n)$$

で与えられる。————

上記の (i) ~ (iii) は、以下に述べる命題 1 ~ 4 の結果をまとめたものである。

[ 命題 1 ]  $G_{n,c}$  に属すグラフにおいて、各節点の線度を表わす確率変数を  $D_v (v=1 \cdots n)$  とすると、 $D_v$  の確率密度関数  $P_r(D_v=k)$  は

$$P_r(D_v=k) = P(k; c) + o(1) \quad (4.9)$$

である。すなわち、 $D_v (v=1 \cdots n)$  は  $n \rightarrow \infty$  のとき、母数  $c$  のポアソン分布に従う。

(証明) 各節点間には、各々独立に  $p(n) = c/(n-1)$  の確率で枝が存在することから、 $D_v (v=1 \cdots n)$  は、

$$P_r(D_v=k) = \binom{n-1}{k} p(n)^k (1-p(n))^{n-k-1} \quad (4.10)$$

なる二項分布に従う。 $p(n) = c/(n-1)$  であることを用いれば、 $n \rightarrow \infty$  のとき

$$P_r(D_v=k) \rightarrow P(k; c) \quad (4.11)$$

が得られる<sup>(46)</sup>。

(証明終)

[ 命題 2 ]  $G_{n,c}$  に属すグラフにおいて、線度  $k (k; \text{有限値})$  をもつ節点の個数を表わす確率変数を  $N_k$  とすると次式が成り立つ。

(注)  $n$  に依存しない値を以下有限値と呼ぶ。

$$N_k = P(k; c) n + o(n) \quad (4.12)$$

(証明)  $N_k$  の平均値  $E(N_k)$ , 分散  $V(N_k)$  が,

$$E(N_k) = P(k; c) n + o(n) \quad (4.13)$$

$$V(N_k) = \theta(n) \quad (4.14)$$

となることを示す。上式が示されれば, チェビシエフの不等式<sup>(注)</sup>を用いて,

$$N_k = P(k; c) n + o(n) \quad (4.15)$$

が得られる。

式(4.13), (4.14)を示すために, 次式で定義される確率変数  $Y_v$  ( $v=1 \cdots n$ ) を導入する。

$$Y_v = \begin{cases} 1 & (\text{節点 } v \text{ の線度が } k \text{ のとき}) \\ 0 & (\text{節点 } v \text{ の線度が } k \text{ 以外のとき}) \end{cases} \quad (4.16)$$

この  $Y_v$  ( $v=1 \cdots n$ ) を用いて,  $E(N_k)$  は

$$E(N_k) = E(\sum_{v=1}^n Y_v) = \sum_{v=1}^n E(Y_v) \quad (4.17)$$

で与えられる。又, すべての  $v$  ( $v=1 \cdots n$ ) に対して,

$$\begin{aligned} E(Y_v) &= P_r(Y_v=1) = P_r(D_v=k) \\ &= \binom{n-1}{k} p^k (1-p)^{n-k-1} \end{aligned} \quad (4.18)$$

但し,

$$p = c/(n-1) \quad (4.19)$$

が成り立つから, 式(4.17)及び式(4.9)を用いて, 式(4.13)が得られる。

次に, 分散  $V(N_k)$  を求める。まず,

$$\begin{aligned} E(N_k^2) &= E((\sum_{v=1}^n Y_v)^2) \\ &= \sum_{v=1}^n E(Y_v^2) + \sum_{v \neq w} E(Y_v Y_w) \\ &= E(N_k) + \sum_{v \neq w} E(Y_v Y_w) \end{aligned} \quad (4.20)$$

(注) 確率変数  $X$  の平均値を  $E(X)$ , 分散を  $V(X)$  とするとき, 任意の  $\epsilon$  に対して

$$P_r(|X - E(X)| \geq \epsilon \sqrt{V(X)}) \leq 1/\epsilon^2 \text{ が成り立つ。}$$



が得られる。又、 $Y_v Y_w$  は

$$Y_v Y_w = \begin{cases} 1 & (\text{節点 } v, w \text{ の線度がともに } k \text{ の場合}) \\ 0 & (\text{それ以外の場合}) \end{cases} \quad (4.21)$$

であることから、すべての  $v, w$  ( $v, w=1 \cdots n, v \neq w$ ) に対して、

$$\begin{aligned} E(Y_v Y_w) &= P_r(D_v = k, D_w = k) \\ &= P_r(D_v = k, D_w = k \mid (v, w) \in \Gamma) p \\ &\quad + P_r(D_v = k, D_w = k \mid (v, w) \notin \Gamma) (1-p) \end{aligned} \quad (4.22)$$

が成り立つ。但し、 $(v, w) \in \Gamma$  は、枝  $(v, w)$  が存在する事象を表わし、 $p$  は式 (4.19) に示された値である。

式 (4.22) において、

$$\begin{aligned} &P_r(D_v = k, D_w = k \mid (v, w) \in \Gamma) \\ &= \left\{ \binom{n-2}{k-1} p^{k-1} (1-p)^{n-2-(k-1)} \right\}^2 \end{aligned} \quad (4.23)$$

$$\begin{aligned} &P_r(D_v = k, D_w = k \mid (v, w) \notin \Gamma) \\ &= \left\{ \binom{n-2}{k} p^k (1-p)^{n-2-k} \right\}^2 \end{aligned} \quad (4.24)$$

であるから、

$$\begin{aligned} E(Y_v Y_w) &= \binom{n-2}{k-1}^2 p^{2k-1} (1-p)^{2n-2k-2} \\ &\quad + \binom{n-2}{k}^2 p^{2k} (1-p)^{2n-2k-3} \end{aligned} \quad (4.25)$$

が得られる。一方、式 (4.17)、(4.18) を用いて、

$$\frac{1}{n^2} E(N_k)^2 = \binom{n-1}{k}^2 p^{2k} (1-p)^{2n-2k-2} \quad (4.26)$$

が得られ、式 (4.25)、(4.26)、(4.19) を用いて

$$\frac{n^2 E(Y_v Y_w)}{E(N_k)^2} = \frac{k^2}{c(n-1)} + \frac{n-k-1}{n-c-1} \quad (4.27)$$

が得られる。従って、

$$\begin{aligned} \sum_{v \neq w} E(Y_v Y_w) &= n(n-1) E(Y_v Y_w) \\ &= \frac{(n-1)}{n} E(N_k)^2 \left\{ \frac{k^2}{c(n-1)} + \frac{n-k-1}{n-c-1} \right\} \end{aligned} \quad (4.28)$$

となる。

式(4.20), (4.28)から

$$\begin{aligned} V(N_k) &= E(N_k^2) - E(N_k)^2 \\ &= E(N_k) + E(N_k)^2 \left\{ \frac{k^2}{cn} + \frac{(c-k-1)n+k+1}{n(n-c-1)} \right\} \end{aligned} \quad (4.29)$$

が得られ,  $k$  が有限値であること, 及び式(4.13)から, 式(4.14)が得られる。

(証明終)

[命題 4.3] ポアソン分布  $P(k; c)$  に対しては, 任意の  $\varepsilon > 0$  に対して

$$\sum_{k=l}^u P(k; c) \geq 1 - \varepsilon \quad (4.30)$$

となる有限値  $l(\varepsilon)$ ,  $u(\varepsilon)$  が存在する。

(証明)  $P(k; c)$  に対しては, 次式が成り立つ。

$$\sum_{k=0}^a P(k; c) \leq (c/a)^a \exp(a-c) \quad (a \leq c) \quad (4.31)$$

$$\sum_{k=a}^{n-1} P(k; c) \leq (c/a)^a \exp(a-c) \quad (a \geq c) \quad (4.32)$$

上式は, 二項分布に関する文献(47)の結果から容易に導ける。式(4.31), (4.32)の右辺は  $[0, c]$  及び  $[c, n-1]$  において単調に 0 に近づくことから, 任意の  $\varepsilon > 0$  に対して  $\varepsilon/2$  以下となる  $a$  の値が存在する。この  $a$  を, 各々,  $l(\varepsilon)$ ,  $u(\varepsilon)$  とおけば, 式(4.30)が得られる。

(証明終)

[命題 4]  $G_{n,c}$  に属すグラフにおいて, 線度  $i$  及び  $j$  ( $i, j$  は有限値)

をもつ節点の集合を、各々、 $X_i$  及び  $X_j$  とする。このとき、 $v \in X_i, w \in X_j$  なる 2 節点  $v, w$  の間に枝が存在する確率  $p_{ij}$  は、

$$p_{ij} = ij/c(n-1) + o(1/n) \quad (4.33)$$

で与えられる。

(証明) 枝  $(v, w)$  が存在する事象を  $(v, w) \in \Gamma$  で表わし、  
 $p \equiv c/(n-1)$  とおく。

ベイズ則を用いて

$$\begin{aligned} p_{ij} &\equiv P_r((v, w) \in \Gamma \mid D_v = i, D_w = j) \\ &= \frac{P_r(D_v = i, D_w = j \mid (v, w) \in \Gamma)}{P_r(D_v = i, D_w = j)} p \end{aligned} \quad (4.34)$$

が得られる。ここで、

$$\begin{aligned} &P_r(D_v = i, D_w = j) \\ &= P_r(D_v = i, D_w = j \mid (v, w) \in \Gamma) p \\ &\quad + P_r(D_v = i, D_w = j \mid (v, w) \notin \Gamma) (1-p) \end{aligned} \quad (4.35)$$

である。

$$\begin{aligned} &P_r(D_v = i, D_w = j \mid (v, w) \in \Gamma) \\ &= \binom{n-2}{i-1} \binom{n-2}{j-1} p^{i+j-2} (1-p)^{2n-i-j-2} \end{aligned} \quad (4.36)$$

$$\begin{aligned} &P_r(D_v = i, D_w = j \mid (v, w) \notin \Gamma) \\ &= \binom{n-2}{i} \binom{n-2}{j} p^{i+j} (1-p)^{2n-i-j-4} \end{aligned} \quad (4.37)$$

及び、 $p = c/(n-1)$  であることを用い、

$$p_{ij} = \frac{ij(n-c-1)}{ij(n-c-1) + c(n-i-1)(n-j-1)} \quad (4.38)$$

が得られる。上式から  $n \rightarrow \infty$  のとき

$$\left\{ p_{ij} - \frac{ij}{c(n-1)} \right\} / \frac{1}{n} \rightarrow 0 \quad (4.39)$$

が得られる。

(証明終)

### 4.3.3 解法Ⅱによる解の値

解法Ⅱは、4.3.1項で述べたように、節点の線度を考慮したヒューリスティックな解法である。ここでは、以下の記述を容易にするため、改めて、解法Ⅱを次の形で表現しておく。

与えられたグラフ  $G = (X, \Gamma)$  に対して、 $\Gamma(v)$  を節点  $v$  ( $v=1 \dots n$ ) の隣接節点の集合とし、 $l, u$  を線度の最小値及び最大値とする。

〈解法Ⅱ〉

ステップ1:  $X = \{1 \dots n\}$  を線度  $k$  によって類別し、 $X_k$  を線度  $k$  をもつ節点の集合とする。  $k \leftarrow l$ .

ステップ2:  $|X_k| = 0$  ならステップ4へ。

ステップ3:  $X_k$  から最小番号をもつ節点  $v$  を選び、

$$I_k \leftarrow I_k \cup \{v\}, X_k \leftarrow X_k - \{v\} - \Gamma(v)$$

$$X_i \leftarrow X_i - \Gamma(v) \quad (i > k)$$

としてステップ2へ。

ステップ4:  $k = u$  なら  $I = \bigcup_{k=l}^u I_k$  が求める節点独立集合である。

$k < u$  なら、 $k \leftarrow k + 1$  としてステップ2へ。

上記の解法Ⅱを、平均線度  $c$  をもつ  $n$  節点ランダムグラフ  $G_{n,c}$  に適用して得られる解の値に対して、次の定理4.1が成り立つ。

まず、文献(48)に従って、以下のことを定義しておく。

〔定義4.2〕  $P_r(G_{n,c} : Q)$  を、 $G_{n,c}$  に属すグラフが性質  $Q$  をもつ確率とする。

$$“ n \rightarrow \infty \text{ で } P_r(G_{n,c} : Q) \rightarrow 1 ”$$

が成り立つとき、

“平均線度  $c$  をもつほとんどすべてのグラフ (*almost every graph*) が性質  $Q$  をもつ” といひ、

“性質  $Q$  (a.e.)”

の書式を用いる。

〔定理4.1〕 解法Ⅱを  $G_{n,c}$  に適用して得られる解の値を表わす確率変数

を  $\beta^{\text{II}}(n, c)$  とすると, 任意の  $\varepsilon > 0$  に対して次式が成り立つ.

$$(\beta^{\text{II}}(c) - \varepsilon)n < \beta^{\text{II}}(n, c) < (\beta^{\text{II}}(c) + \varepsilon)n \quad (a.e.) \quad (4.40)$$

$$\beta^{\text{II}}(c) = \sum_{k=l}^u a_k \quad (4.41)$$

但し,  $l$  及び  $u$  は式 (4.30) に示した値であり, 又,  $a_k$  ( $k=l \cdots u$ ) は, 次の漸化式によって得られる値である.

$$a_l = \frac{c}{l^2} \ln \left\{ 1 + \frac{l^2}{c} P(l; c) \right\} \quad (4.42)$$

$$a_k = \frac{c}{k^2} \ln \left\{ 1 + \frac{k^2}{c} P(k; c) \exp \left( -\frac{k}{c} \sum_{i=l}^{k-1} i a_i \right) \right\} \quad (k > l) \quad (4.43)$$

定理 4.1 は, 次の補題 4.1, 補題 4.2 を用いれば, 容易に証明できる. 記述を容易にするため, 空集合  $X_{l-1}$  を定義しておく.

[補題 4.1] 解法 II のステップ 2 ~ 4 を  $X_{l-1} \cdots X_{k-1}$  に対して実行した後 (すなわち,  $k-1$  以下の線度をもつ節点から独立集合を生成した後) の  $|X_k|$  を表わす確率を  $N_{k, k-1}$  とし, 更に,  $X_k$  に対して実行した後の  $|I_k|$  を表わす確率変数を  $\beta_k(n)$  とする. このとき,

$$N_{k, k-1} = \xi_{k, k-1} n + o(n) \quad (\xi_{k, k-1} \text{ は正の有限値}) \quad (4.44)$$

ならば, 任意に小さくできる  $\varepsilon_k > 0$  に対して,

$$(a_k - \varepsilon_k)n < \beta_k < (a_k + \varepsilon_k)n \quad (a.e.) \quad (4.45)$$

但し,

$$a_k = \frac{c}{k^2} \ln \left( 1 + \frac{k^2}{c} \xi_{k, k-1} \right) \quad (4.46)$$

が成り立つ.

[補題 4.2] 解法 II のステップ 2 ~ 4 を  $X_{l-1} \cdots X_{k-1}$  に対して実行した

後の  $|X_i| (i \geq k)$  を表わす確率変数を  $N_{i, k-1} (i \geq k)$  とする。更に、 $X_k$  に対して実行した後、 $|I_k|$  及び  $|X_i| (i \geq k+1)$  を表わす確率変数を、各々、 $\beta_k(n)$  及び  $N_{i, k} (i \geq k+1)$  とする。このとき、次のことがいえる。

$$N_{i, k-1} = \xi_{i, k-1} n + o(n) \quad (i \geq k) \quad (\xi_{i, k-1} \text{ は正の有限値}) \quad (4.47)$$

且つ

$$\beta_k(n) = a_k n + o(n) \quad (a_k \text{ は正の有限値}) \quad (4.48)$$

ならば、

$$N_{i, k} = \xi_{i, k} n + o(n) \quad (i \geq k+1) \quad (4.49)$$

但し、

$$\xi_{i, k} = \xi_{i, k-1} \exp \left( -\frac{ik}{c} a_k \right) \quad (4.50)$$

が成り立つ。

まず、補題 4.1, 補題 4.2 が成り立てば、定理 4.1 が成り立つことを示す。

(定理 4.1 の証明)

式 (4.12) から、

$$N_{k, l-1} = P(k; c) n + o(n) \quad (k = l \cdots u) \quad (4.51)$$

がいえ、補題 4.1 の式 (4.44) の仮定が満足される。よって  $k = l$  に対して解法 II が適用された後、任意に小さくできる  $\epsilon_l > 0$  に対して、

$$(a_l - \epsilon_l) n < \beta_l < (a_l + \epsilon_l) n \quad (a. e. e.) \quad (4.52)$$

但し、

$$a_l = \frac{c}{l^2} \ln \left\{ 1 + \frac{l^2}{c} p(l; c) \right\} \quad (4.53)$$

が成り立つ。新たに,

$$a_l = \frac{c}{l^2} \ln \left\{ 1 + \frac{l^2}{c} P(k; c) \right\} \pm \varepsilon_l \quad (4.54)$$

とおけば, 式(4.51), (4.52)によって補題4.2の仮定が満足され,  $k > l$  に対して

$$N_{k,l} = \left\{ P(k; c) \exp \left( -\frac{kl}{c} a_l \right) \right\} n + o(n) \quad (4.55)$$

が成り立つ。上の議論は  $k > l$  の場合についても繰り返すことができ, 結局

$$\xi_{k,l-1} = P(k; c) \quad (k > l-1) \quad (4.56)$$

$$\xi_{k,i} = \xi_{k,i-1} \exp \left( -\frac{ki}{c} a_i \right) \quad (k > i > l-1) \quad (4.57)$$

$$a_k = \frac{c}{k^2} \ln \left( 1 + \frac{k^2}{c} \xi_{k,k-1} \right) \pm \varepsilon_k \quad (k > l-1) \quad (4.58)$$

なる漸化式が得られる。

式(4.56), (4.57)から

$$\xi_{k,k-1} = P(k; c) \exp \left( -\frac{k}{c} \sum_{i=l}^{k-1} i a_i \right) \quad (4.59)$$

が得られ, これを式(4.58)に代入して次式が得られる。

$$a_k = \frac{c}{k^2} \ln \left\{ 1 + \frac{k^2}{c} P(k; c) \exp \left( -\frac{k}{c} \sum_{i=l}^{k-1} i a_i \right) \right\} \pm \varepsilon_k \quad (k > l) \quad (4.60)$$

各  $\varepsilon_k$  は  $a_k$  に無関係に任意に小さくできる値であり, 又, 式(4.56)~(4.58)

の漸化式は、 $u-l+1$  ( $u, l$ は有限値) 回繰り返されるだけであることから、定理 4.1 の結果を得る。

(定理 4.1 証明終)

従って、以下、補題 4.1 及び補題 4.2 を証明することにする。

まず、次のことを注意しておく。解法 II においては、各  $X_k$  から番号の小さい順に節点を選ばれる。ランダムグラフの定義から、このことは、線度  $k$  をもつ節点の中からランダムに節点を選ばれることに相当する。又、 $X_k$  に属す節点は、すべて等しい確率  $p_{i,k}$  ( $i=1 \cdots k-1$ ) で  $I_i$  ( $i=1 \cdots k-1$ ) の節点に隣接することから、 $I_i$  ( $i=1 \cdots k-1$ ) が生成されるとき、 $X_k$  の中のすべての節点は等しい確率  $p_{i,k}$  ( $i=1 \cdots k-1$ ) で  $X_k$  から除去され、等しい確率  $1-p_{i,k}$  で  $X_k$  に残る。

更に、 $X_{k,k-1}$  に解法 II を適用するとき、 $I_k$  に属す節点はランダムに選ばれることから、 $X_{k,k-1}$  の中のすべての節点は、等しい確率  $p_{kk}$  で  $X_{k,k-1}$  から除去され、等しい確率  $1-p_{kk}$  で  $X_{k,k-1}$  に残ることになる。

(補題 4.1 の証明)

記述を容易にするため、補題 4.1 の証明の範囲内で、次のように記号を再定義する。

$$X_0 \equiv X_{k,k-1}, \quad N_0 \equiv N_{k,k-1}, \quad \xi_0 \equiv \xi_{k,k-1} \quad (4.61)$$

又、 $\lim_{n \rightarrow \infty} A/n \leq \lim_{n \rightarrow \infty} B/n$  の意味で、

$$A \leq_n B \quad (4.62)$$

の記号を用いる。

まず、 $X_0$  から  $\alpha \equiv \delta n$  ( $\delta$  は正の有限値) 個の節点が  $I_k$  に加えられた段階での  $X_k$  を  $X_1$  とし、 $|X_1|$  を表わす確率変数を  $N_1$  とする。又、 $r \geq 1$  に対して、 $X_{r-1}$  から  $\alpha$  個の節点が  $I_k$  に加えられた段階での  $X_k$  を  $X_r$  とし、 $|X_r|$  を表わす確率変数を  $N_r$  とする。

このとき、次のことが成り立つ。



$$\sigma_{r-1} = \theta(n), \quad \sigma_{r-1} q^\alpha - \alpha = \theta(n) \quad (4.63)$$

なる  $r \geq 1$  に対して

$$\sigma_{r-1} \leq_n N_{r-1} \quad \text{ならば} \quad \sigma_{r-1} q^\alpha - \alpha \leq_n N_r \quad (4.64)$$

又,

$$\rho_{r-1} = \theta(n), \quad (\rho_{r-1} - \alpha) q^\alpha = \theta(n) \quad (4.65)$$

なる  $r \geq 1$  に対して

$$N_{r-1} \leq_n \rho_{r-1} \quad \text{ならば} \quad N_r \leq_n (\rho_{r-1} - \alpha) q^\alpha \quad (4.66)$$

但し,

$$q = 1 - p_{kk} \quad (4.67)$$

である.

式(4.44)の仮定によって

$$\sigma_0 \leq_n N_0 \leq_n \rho_0 \quad (4.68)$$

但し,

$$\left. \begin{aligned} \sigma_0 &= \xi_0 n + o(n) \\ \rho_0 &= \xi_0 n + o(n) \end{aligned} \right\} \quad (4.69)$$

がいえるから,  $r \geq 1$  に対して, 式(4.64), (4.66)が成り立つことを示せばよい. まず, (i)で式(4.64)を, 次いで(ii)で式(4.66)を示す.

(i)  $N_{r-1} = \rho_{r-1} = \theta(n)$ が与えられたときの  $N_r$ の上限値  $\rho_r$ を求める.

この条件の下で,  $N_r \geq x$ である確率  $P_r(N_r \geq x)$ について, 次式が成り立つ.

$P_r(N_r \geq x) = P_r(X_{r-1}$ の中に  $\alpha$ 個の節点からなる独立集合が少なくとも

1個存在し、且つ、残りの  $\rho_{r-1} - \alpha$  個の節点の中に  $\alpha$  個のすべての節点に隣接しない節点が  $x$  個以上存在する)

$\leq P_r (X_{r-1}$  の中で、 $\alpha$  個のすべての節点に隣接しない接点があり、残りの  $\rho_{r-1} - \alpha$  個の中に  $x$  個以上存在する)

$$= \sum_{i \geq x} \binom{\rho_{r-1} - \alpha}{i} q^\alpha (1 - q^\alpha)^{\rho_{r-1} - \alpha - i} \quad (4.70)$$

式(4.70)の最後の式に従う確率変数を  $U_r$  とすると、 $U_r$  の平均値及び分散は、次式で与えられる。

$$E(U_r) = (\rho_{r-1} - \alpha) q^\alpha \quad (4.71)$$

$$V(U_r) = (\rho_{r-1} - \alpha) q^\alpha (1 - q^\alpha) \quad (4.72)$$

上式に、 $\alpha = \delta n$ 、 $\rho_{r-1} = \theta(n)$  及び

$$q \equiv 1 - \rho_{kk} = 1 - k^2/c(n-1) + o(1/n) \quad (4.73)$$

を代入し、チェビシェフの不等式を用いれば、 $(\rho_{r-1} - \alpha) q^\alpha = \theta(n)$  である  $r$  の値に対して、

$$U_r = (\rho_{r-1} - \alpha) q^\alpha + o(n) \quad (4.74)$$

が成り立つ。よって、式(4.70)から、

$$N_r \leq_n \rho_r \equiv (\rho_{r-1} - \alpha) q^\alpha \quad (4.75)$$

が得られる。

(ii)  $N_{r-1} = \sigma_{r-1} = \theta(n)$  が与えられたときの  $N_r$  の下限値  $\sigma_r$  を求める。

この条件の下で、 $N_r \geq x$  である確率  $P_r(N_r \geq x)$  について、次式が成り立つ。

$P_r(N_r \geq x) = \sum_{i \geq x} P_r(X_{r-1}$  に属す  $\alpha + i$  個の節点の中で少なくとも  $\alpha$  個は独立集合であり、残りの  $i$  個の節点は、この  $\alpha$  個のすべての節点に隣接

しない)

$$\begin{aligned} &\geq q^{\frac{\alpha}{2}(\alpha-1)} \sum_{i \geq x} \binom{\sigma_{r-1}}{\alpha+i} q^{\alpha i} (1 - q^{\alpha})^{\sigma_{r-1} - \alpha - i} \\ &\geq \sum_{i \geq x} \binom{\sigma_{r-1}}{\alpha+i} \left(\frac{q^{\alpha}}{q}\right)^{\alpha+i} (1 - q^{\alpha})^{\sigma_{r-1} - \alpha - i} \end{aligned} \quad (4.76)$$

式(4.76)の最後の式に従う確率変数を  $L_r$  とすると、式(4.70)から式(4.74)を導いた場合と同様にして、 $\sigma_{r-1} q^{\alpha} - \alpha = \theta(n)$  である  $r$  の値に対して、

$$L_r = \sigma_{r-1} q^{\alpha} - \alpha + o(n) \quad (4.77)$$

が成り立つ。よって、式(4.76)より

$$N_{r, n} \geq \sigma_r \equiv \sigma_{r-1} q^{\alpha} - \alpha \quad (4.78)$$

が得られる。

以上、(i) 及び (ii) によって、式(4.64) 及び式(4.66) が示された。

式(4.64), (4.66) から次のことがいえる。

$\rho_r$  及び  $\sigma_r$  を各々、

$$\rho_r = (\rho_{r-1} - \alpha) q^{\alpha} \quad (4.79)$$

$$\sigma_r = \sigma_{r-1} q^{\alpha} - \alpha \quad (4.80)$$

$$\rho_0 = \theta(n), \quad \sigma_0 = \theta(n) \quad (4.81)$$

の漸化式を解いて得られる値とすると、 $\rho_r = \theta(n)$ ,  $\sigma_r = \theta(n)$  である  $r \geq 1$  に対して

$$N_r \leq_n \rho_r \quad (4.82)$$

$$N_{r_n} \geq \sigma_r \quad (4.83)$$

が成り立つ。

式(4.79)及び式(4.80)を解いて

$$\rho_r = \left( \rho_0 + \frac{\alpha q^\alpha}{1-q} \right) q^{\alpha r} - \frac{\alpha q^\alpha}{1-q} \quad (4.84)$$

$$\sigma_r = \left( \sigma_0 + \frac{\alpha}{1-q} \right) q^{\alpha r} - \frac{\alpha}{1-q} \quad (4.85)$$

が得られる。又、 $r_u$  及び  $r_l$  を次式で与えられる値とする。

$$r_u = \left\lceil \frac{c}{k^2 \delta} \ln \left\{ 1 + \frac{\xi_0}{\delta} \left( \exp \left( \frac{k^2}{c} \delta \right) - 1 \right) \right\} \right\rceil \quad (4.86)$$

$$r_l = \left\lfloor \frac{c}{k^2 \delta} \ln \left\{ 1 + \frac{\xi_0}{\delta} \left( 1 - \exp \left( -\frac{k^2}{c} \delta \right) \right) \right\} \right\rfloor \quad (4.87)$$

式(4.84), (4.85)に  $\alpha = \delta n$ ,  $\rho_0 = \xi_0 n + o(n)$ ,  $\sigma_0 = \xi_0 n + o(n)$  及び  $q = 1 - k^2/c(n-1) + o(1/n)$  を代入することにより,

$$\rho_{r_u} \leq_n 0, \quad \sigma_{r_l} \geq 0 \quad (4.88)$$

が得られる。従って、式(4.82), (4.83)から

$$N_{r_u} \leq_n 0, \quad N_{r_l} \geq 0 \quad (4.89)$$

が得られ、結局、 $|I_k|$  を表わす確率変数  $\beta_k(n)$  は次式を満たす。

$$r_l \alpha \leq \beta_k(n) \leq r_u \alpha \quad (a.e.) \quad (4.90)$$

上式に、式(4.86), (4.87)を代入することにより

$$\beta_k(n) \leq \left[ \frac{c}{k^2} \ln \left\{ 1 + \frac{\xi_0}{\delta} \left( \exp \left( \frac{k^2}{c} \delta \right) - 1 \right) \right\} + \delta \right] n \quad (a.e.) \quad (4.91)$$

$$\beta_k(n) \geq \left[ \frac{c}{k^2} \ln \left\{ 1 + \frac{\xi_0}{\sigma} \left( 1 - \exp \left( -\frac{k^2}{c} \delta \right) \right) \right\} - \delta \right] \quad (a.e.) \quad (4.92)$$

が得られる。  $x \equiv k^2 \delta / c$  とおき、  $x \rightarrow 0$  のとき  $(\exp(x) - 1) / x \rightarrow 1$ 、且つ、 $(1 - \exp(-x)) / x \rightarrow 1$  であることを用いれば、

$$(a_k - \varepsilon_k) n < \beta_k(n) < (a_k + \varepsilon_k) n \quad (a.e.) \quad (4.93)$$

が得られる。但し、

$$a_k = \frac{c}{k^2} \ln \left( 1 + \frac{k^2}{c} \xi_0 \right) \quad (4.94)$$

であり、  $\varepsilon_k$  は  $\delta \rightarrow 0$  のとき単調に 0 に近づく値を表わす。

(補題 4.1 証明終)

(補題 4.2 の証明)

$P_r(N_{i,k} = x) \quad (i \geq k+1)$  を、

$$N_{i,k-1} = \xi'_{i,k-1} \equiv \xi_{i,k-1} n + o(n) \quad (4.95)$$

$$\beta_k(n) = a'_k \equiv a_k n + o(n) \quad (4.96)$$

の条件の下に、  $N_{i,k} = x \quad (i \geq k+1)$  となる確率とする。補題 4.1 の証明の前に述べたように、  $I_k$  に節点を加えられるごとに、  $X_{i,k-1} \quad (i \geq k)$  に属す節点は、すべて等しい確率  $p_{ki} \quad (i \geq k)$  で  $X_{i,k-1}$  から除去され、等しい確率  $1 - p_{ki} \quad (i \geq k)$  で  $X_{i,k-1}$  に残ることになる。このことから、次式が得られる。

$P_r(N_{i,k} = x) = P_r(X_{i,k-1}$  の中で  $I_k$  に加えられた  $a'_k$  個の節点に隣接しない節点が  $x$  個存在する)

$$= \binom{\xi'_{i,k-1}}{x} q_{ki}^{a'_k x} \left( 1 - q_{ki}^{a'_k} \right)^{\xi'_{i,k-1} - x} \quad (4.97)$$

但し、式(4.33)から、

$$q_{ki} \equiv 1 - p_{ki} = 1 - ki/c(n-1) + o(1/n) \quad (4.98)$$

である。式(4.95), (4.96), (4.97)から、 $N_{i,k}$  ( $i \geq k+1$ )の平均値及び分散は

$$E(N_{i,k}) = \left[ \xi_{i,k-1} \exp\left(-\frac{ik}{c} a_k\right) \right] n + o(n) \quad (4.99)$$

$$V(N_{i,k}) = \theta(n) \quad (4.100)$$

で与えられる。従って、チェビシエフの不等式を用いて、

$$N_{i,k-1} = \left[ \xi_{i,k-1} \exp\left(-\frac{ik}{c} a_k\right) \right] n + o(n) \quad (4.101)$$

が得られる。

(補題4.2証明終)

#### 4.3.4 解法Iと解法IIの比較

前項では、 $G_{n,c}$ に解法IIを適用して得られる解の値を表わす確率変数 $\beta^{\text{II}}(n, c)$ が、任意の $\varepsilon > 0$ に対して

$$(\beta^{\text{II}}(c) - \varepsilon)n < \beta^{\text{II}}(n, c) < (\beta^{\text{II}}(c) + \varepsilon)n \quad (a.e.) \quad (4.102)$$

$$\beta^{\text{II}}(c) = \sum_{k=l}^u a_k \quad (4.103)$$

となることを示した。但し、 $l$ 及び $u$ は、式(4.30)に示した値であり、また $a_k$  ( $k=l \cdots u$ )は、式(4.42), (4.43)の漸化式を解いて得られる値である。

一方、解法Iによる解の値及び最適解に対する上限値を表わす確率変数を、各々、 $\beta^{\text{I}}(n, c)$ 及び $\beta^*(n, c)$ とすると、次式が成り立つ<sup>(6)</sup>。

$$\left( \frac{\ln(c+1)}{c} - \varepsilon \right) n < \beta^I(n, c) < \left( \frac{\ln(c+1)}{c} + \varepsilon \right) n$$

(a. e.) (4.104)

$$\beta^*(n, c) \leq (1 + \varepsilon)\alpha \quad (\text{a. e.}) \quad (4.105)$$

但し、 $\varepsilon > 0$  は任意に小さな値を表わし、又、 $\alpha$  は次式を満たす値である。

$$-\alpha \ln \alpha - (1 - \alpha) \ln(1 - \alpha) = c\alpha^2/2 \quad (4.106)$$

$\ln(c+1)/c$  及び式(4.106) から得られる  $\alpha$  の値を、各々、 $\beta^I(c)$  及び  $\beta^*(c)$  で表わし、 $c$  を変数としたときの  $\beta^I(c)$ 、 $\beta^{II}(c)$  及び  $\beta^*(c)$  の値を図 4.1 に示す。但し、式(4.107) を用いて  $\beta^{II}(c)$  を計算する場合、式(4.30) における  $\varepsilon$  の値を  $\beta^{II}(c)$  に対して十分小さな値に定め、 $l$  及び  $u$  の値を決定している。

図 4.1 から、解法 II による解が解法 I による解よりも良い解となることが確かめられる。

更に、各  $c$  に対する  $\beta^{II}(c)/\beta^I(c)$ 、 $\beta^*(c)/\beta^I(c)$  及び  $\beta^*(c)/\beta^{II}(c)$  の値を図 4.2 に示す。

図 4.2 から、平均線度  $c$  が小さいグラフほど、線度の非減少順に節点を選ぶことによる解の改善度が大きいことが確かめられる。これは、 $c$  が大きいグラフほど、 $c$  に近い線度をもつ節点の数が増す(各線度をもつ節点の個数は、式(4.12) で与えられ、母数  $c$  のポアソン分布の変動係数は  $1/\sqrt{c}$  で与えられる)ことから予想できる結果である。

図 4.1 では、 $c \leq 1000$  に対して  $\beta^{II}(c)$  の値を求めたが、更に大きな  $c$  に対して、 $\beta^{II}(c)$  の値を求めることは可能である。大きな  $c$  の値に対して、 $\varepsilon_0 \beta^{II}(c)$  の誤差内で  $\beta^{II}(c)$  を求めるために必要な式(4.43) の漸化式の数は、たかだか、 $2\sqrt{2c \ln(c/\varepsilon_0)}$  であることが示される<sup>(注)</sup>。

(注) 脚注の説明は次頁へ

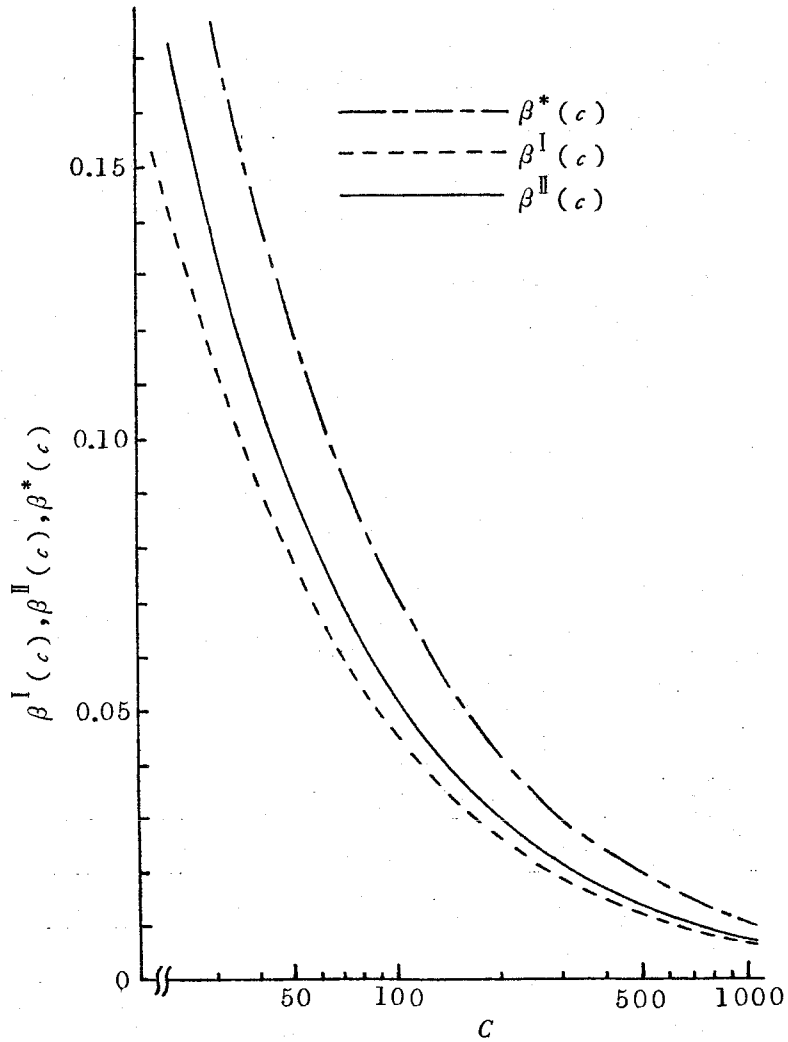


図 4.1  $\beta^I(c)$ ,  $\beta^{II}(c)$ ,  $\beta^*(c)$  の値

(注)  $c$  が大きな値をもつとき、ポアソン分布は正規分布近似できる。更に、

$$\int_z^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) dz < \exp\left(-\frac{z^2}{2}\right) \quad \left(z = \frac{k-c}{\sqrt{c}}\right)$$

$$\int_{-\infty}^z \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) dz < \exp\left(-\frac{z^2}{2}\right) \quad \left(z = \frac{c-k}{\sqrt{c}}\right)$$

であり、 $\beta^{II}(c) > \frac{1}{c}$  であることを用いて、この結果が得られる。



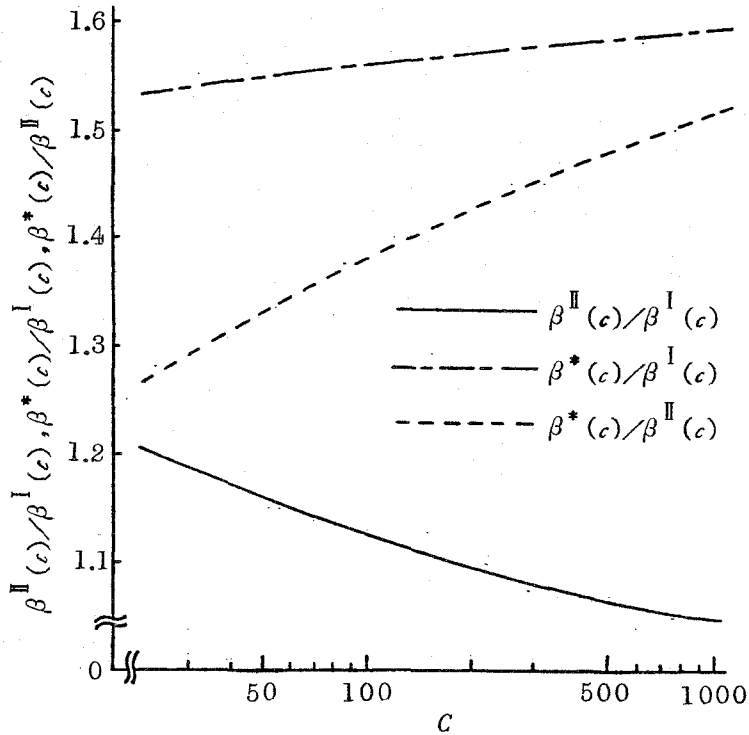


図 4.2  $\beta^{\text{II}}(c)/\beta^{\text{I}}(c)$ ,  $\beta^*(c)/\beta^{\text{I}}(c)$ ,  $\beta^*(c)/\beta^{\text{II}}(c)$  の値

#### 4.3.5 スケジューリング問題との関係

本節では、 $m \geq n$  及び  $r_{kl} = 0$  又は  $B_l$  ( $k=1 \dots n$ ,  $l=1 \dots s$ ) で、 $s$  が任意の値をもつとき、同時に処理できるジョブの数を最大にする問題を、グラフの節点独立集合問題に定式化し、実用的な近似解法をランダムグラフを用いて確率的に評価した。

上記のスケジューリング問題を解く解法としては、節点独立集合問題への定式化を用いる解法より優れた解法は存在し得ないことを以下に示す。(この意味で2つの問題は同じ問題と考えることができる。)記述を容易にするため、以下、上記のスケジューリング問題を  $SC H$  と呼び、節点独立集合問題を  $V P$  と呼ぶ。又、 $SC H$  を 4.3.1 項に述べた方法を用いて  $V P$  に定式化し、この  $V P$  に解法  $A$  を適用する解法を  $A(SC H)$  で表わす。

各ジョブと各節点は1対1に対応することから、 $VP$ に対する解法 $A$ の最悪誤差は、 $SCH$ に解法 $A(SCH)$ を適用したときの最悪誤差と一致する。この最悪誤差を $\epsilon_A$ で表わす。

いま、解法 $A$ は最悪誤差が $\epsilon_A$ 以下となる最小の時間計算量をもつ解法であると仮定し、その時間計算量を $O(f_A(n))$ で表わす。このとき、 $A(SCH)$ の時間計算量は $O(\max(ns, f_A(n)))$ となる。 $VP$ への定式化を用いず、最悪誤差が $\epsilon_A$ 以下となる $SCH$ に対する解法は $O(\max(ns, f_A(n)))$ より小さい時間計算量をもつことはないことが次のように示される。

まず $SCH$ の解法は少なくとも $O(ns)$ の時間計算量をもつことに注意しよう。従って $\max(ns, f_A(n)) = f_A(n)$ の場合を考えれば十分である。

任意のグラフ $G \equiv (X, \Gamma)$ に対して

$$J \equiv X \quad (4.107)$$

$$R \equiv \{R_{ij} \mid (i, j) \in \Gamma\} \quad (4.108)$$

なるジョブ集合 $J$ 、及びリソース集合 $R$ をつくる。又、ジョブ $i$ がリソース $R_{ij}$ を要求する量 $r_i(ij)$ 、及び、リソース $R_{ij}$ の容量 $B_{ij}$ を

$$r_i(ij) = 1 \quad (\forall i \in J, R_{ij} \in R) \quad (4.109)$$

$$B_{ij} = 1 \quad (\forall R_{ij} \in R) \quad (4.110)$$

とおく。式(4.107)～(4.110)で定義されるモデルにおいて、同時に処理できるジョブ集合はグラフ $G$ の節点独立集合と一致する。従って、いま、 $SCH$ に対する解法 $B$ が $O(f_A(n))$ より小さい時間計算量で、最悪誤差 $\epsilon_A$ 以下の解を得ると仮定すると、 $f_A(n) > ns$ であることから上記の定式化に要す時間は無視でき、結局、解法 $B$ は $O(f_A(n))$ より小さい時間計算量で $VP$ を解くことになる。これは解法 $A$ が最小の時間計算量をもつ解法であることに矛盾し、従ってこのような解法 $B$ は存在し得ない。

ランダムグラフは、 $SCH$ の立場からは、ジョブの間の競合(リソースの奪合い)が、それぞれ独立に等しい確率で生じるモデルに相当し、図4.1及び

図 4.2 の結果から、特に、競合数の少ない場合に、解法 II は有効な解法であるといえる。なお、VP への定式化も含めた時間計算量は、解法 I で  $O(ns)$ 、解法 II で  $O(\max(ns, n \log n))$  である。

又、図 4.1 及び図 4.2 に示した結果は、 $n$  が十分大きい場合を仮定して得られたものであるが、解の平均値のみを考える場合には、そのまま利用できる。又、4.3.2 項、4.3.3 項で得られた式において、 $o(n)$  や  $\varepsilon$  を更に厳密な  $n$  の関数で記述することにより、 $n$  が小さい場合の解の値の分散を理論的に導出することは可能であるが、その解析は今後に残されている。実験的には、 $n = 300$  で乱数を用いてランダムに発生されたグラフに対する近似解の値が、理論値と数%の誤差内で一致することを確かめている<sup>(45)</sup>。

#### 4.4 結 言

本章では、リソース条件をもつスケジューリング問題の中で、多数のジョブが複数のリソースを同時に要求するとき、同時に処理できるジョブの数を最大にする問題を、グラフの節点独立集合問題に定式化し、2つのヒューリスティックな解法をランダムグラフを用いて確率的に評価した。

まず、線度の非減少順に節点を選び得られる解の値（確率変数）が、“ほとんどすべてのグラフ”で、平均線度のみによって一意的に定まることを示した。

次に、この値を、ランダムに節点を選び得られる解の値と比較し、線度の非減少順に節点を選ぶことによって解の値が改善されることを理論的に確かめた。又、特に、平均線度が小さなグラフほど、改善度が大きくなることを確かめた。以上の結果は、従来の実験結果を理論的に裏付ける結果である。

4.3.1 項で述べた解法 I ~ III を含め、この問題に対して現在までに提案されているすべての近似解法は、最悪の場合の評価からは絶望的な結果しか得られていない。特に、解法 I ~ III に対しても、実験結果からは、その解の良さに差が認められるにもかかわらず、最悪の場合の評価からは、同程度に悪いという結果しか得られていない。最悪の場合の評価から良さが保証される近似解法が知られていない問題については、少なくとも、ある分布に従う入力データに

対して有効な近似解法を見い出すことは重要であり、図 4.2 の結果は、解法 II が、 $n \geq c$  なる節点数  $n$  をもつランダムグラフに対して有効な近似解法となることを示している。

ランダムグラフは、上記のスケジューリング問題の立場からは、ジョブの間の競合が各々独立に等しい確率で生じる入力データの集合に相当し、本章で得られた結果は、その集合に属す“ほとんどすべての入力データ”に対して適用できるという点で重要である。ただ、ジョブの数が少ない場合について解の値の分散を理論的に導出することは今後に残された課題である。

## 第 5 章 結 論

本研究では、ジョブの間に先行関係が存在しない静的スケジューリング問題に対して、次の (i), (ii) の立場から考察を行なった。

- (i) 小規模な問題に対して最適スケジュールを求める。
- (ii) 大規模な問題に対して実用的な近似解法を理論的に評価する。

本研究で得られた結果と残された問題点を、以下にまとめる。

第 2 章では、リソース条件をもたない一般的な静的スケジューリング問題に対する厳密解法を提案し、この解法が既存の解法と比較して計算効率の面で優れていることを、実験的に確かめた。又、この解法の中心的役割を果たす山登り法は、極めて短時間に最適解に対する下限値を求め、しかも、この下限値は最適解に対して十分良い近似を与える。このことから、最適解の下限値のみが必要な場合（例えば、近似解法を評価する場合など）にも、この山登り法は有効である。この山登り法は、リソース条件をもつ問題に対してそのまま適用することはできないが、シンプレックス法に代わる有効な手法を見出す上で、1つの手がかりを与えるものと考えている。

第 3 章では、重み付き平均滞留時間最小化問題に対する、重み/処理時間非増加順リストスケジューリングを最悪の場合について評価し、この解法の最悪誤差の上限値が、

$$|J_I| \leq m \text{ ならば } \frac{1}{4} \left( \frac{m \cdot \max \tau_k}{\sum \tau_k} \right)^2$$

$$|J_I| > m \text{ ならば } \frac{1}{4} \left\{ m \cdot \max \left( \frac{\max \tau_k}{\sum \tau_k}, \frac{\max_{k \in J_I} \tau_k}{\sum_{k \in J_I} \tau_k} \right) \right\}^2$$

で与えられることを示した。但し、 $J_I$  は  $c_k/\tau_k$  が最大の値をもつジョブ集合を表わす。上式は、各問題ごとに、ジョブの処理時間の値から最悪誤差の上限値を知ることができるという点で有効であり、特に、 $m$  に比べて  $n$  が大きく  $\tau_k$

が一様に分布する問題に対して，この解法が最適解に近い解を得るといふ従来の実験結果を理論的に裏付けている。

更に，この解法の時間計算量に影響を与えない1つの操作を加えたリストスケジューリングを評価し， $\sum \tau_i/m$  を越える処理時間をもつジョブが存在しないとき，この解法の最悪誤差の上限値が， $1/4$  で与えられることを示した。上記の条件は，又，すべてのジョブの処理時間がプロセッサの平均終了時刻を越えないといいかえることもでき，少なくとも，すべてのプロセッサに2個以上のジョブが割当てられる場合にはこの条件は満たされることから，この解法は大規模な問題を解くための有効な解法であるといえる。ただ，任意の処理時間をもつジョブ集合に対して，最悪誤差の上限値が $1/4$  で与えられるように解法を改良することは，今後に残された課題である。又，納期を含む評価関数をもつ問題に対しては，有効な近似解法は知られていない。この種の問題に対して一般的に有効な近似解法を見つけることは難しいが，納期や処理時間の値に制約が加えられた問題に対して有効な近似解法を見い出すことは期待できる。

第4章では，リソース条件をもつスケジューリング問題の中で，多数のジョブが複数のリソースを同時に要求するとき，同時に処理できるジョブの数を最大にする問題を，グラフの節点独立集合問題に定式化し，この問題に対する2つのヒューリスティックな解法をランダムグラフを用いて確率的に評価し，比較した。

まず，線度の非減少順に節点を選び得られる解の値（確率変数）が，“ほとんどすべてのグラフ”で，平均線度 $c$ のみによって一意的に定まることを示した。次に，この値を，ランダムに節点を選び得られる解の値と比較し，線度の非減少順に節点を選ぶことによって解の値が改善されることを理論的に確かめた。又，特に，平均線度が小さなグラフほど，改善度が大きくなることを確かめた。更に，この解の値と最適解の値との相対誤差の上限値は $c = 50$  で約0.33， $c = 500$  で約0.48の値をもつ $c$ に関する単調増加関数となり，ランダムグラフに対する近似解法としてこの解法が有効な解法であることが確かめられた。第4章で得られる解の理論値は，十分大きな節点数をもつグラフに

対して得られる値であるが、解の平均値のみを考える場合には、任意の節点数をもつグラフに対して適用できる。各式における  $o(n)$  や  $\varepsilon$  を、更に厳密な  $n$  の関数で記述することにより、任意の節点数をもつグラフに対して解の値の分散を導出することも可能であるが、その解析は今後に残されている。

ランダムグラフは、上記のスケジューリング問題の立場からは、ジョブの間の競合が、独立に、等しい確率で生じるすべての入力データの集合に相当し、第4章で得られる結果は、その集合に属す、“ほとんどすべての入力データ”に対して適用できるという点で重要である。

近似解法の最悪の場合の評価は、すべての入力データに対して、その解法の良さを保証するという点で重要であるが、一方、第4章で考察した問題のように、最悪の場合の評価から良さが保証される近似解法が知られていない問題に対しては、入力データの情報をもとに有効な近似解法を選択せざるを得ない。第4章で用いた確率的評価は、入力データが統計的情報をもつ場合に対して有効な近似解法を見い出す上で重要である。

## 謝 辞

本研究の全過程を通じて、終始御懇切な御指導、御激励を載いた大阪大学工学部中西義郎教授に衷心より感謝の意を表する。

大阪大学大学院在学中には、通信工学一般に関して御指導、御教示を載いた大阪大学工学部板倉清保名誉教授、熊谷信昭教授、滑川敏彦教授、手塚慶一教授に深謝する。

又、本研究遂行にあたって、種々有益な御討論、御助言を載いた大阪大学工学部中野秀男助手に深謝する。

更に、日頃御討論載いた大阪大学工学部岡田博美助手をはじめとする中西研究室の諸兄に厚く御礼申し上げる。



## 文 献

- (1) 関根監訳, "スケジューリングの理論", 日刊工業新聞社(1971)  
—— R. W. Conway, W. L. Maxwell and L. W. Miller,  
"Theory of scheduling", Addison Wesley (1967)
- (2) K. R. Barker, "Introduction to sequencing and scheduling", John Wiley (1974)
- (3) E. G. Coffman, "Computer and job-shop scheduling theory", John Wiley (1976)
- (4) M. L. Fisher, "Optimal solution of scheduling problems using Lagrange multipliers: Part I", Oper. Res., 21, p.1114 (1973)
- (5) R. McNaughton, "Scheduling with deadlines and loss functions", Manage. Sci., 6, p.1 (1959)
- (6) R. M. Karp, "The probabilistic analysis of some combinatorial search algorithm", Algorithms and Complexity (edited by J. F. Traub), p.1, Academic Press (1976)
- (7) D. S. Johnson, "Approximation algorithms for combinatorial problems", J. of Comp. and Sys. Sci., 9, p.256 (1974)
- (8) E. Balas, "Project scheduling with resource constraints", Applications of Mathematical Programming Techniques (edited by E. D. Beale), p.185, English Univ. Press London (1970)
- (9) D. D. Chamberlin, "The single assignment approach to parallel processing", Proc. of AFIPS Conf., 39, p.263 (1971)
- (10) D. C. Wood, "A technique for colouring a graph applica-

- ble to large scale timetabling problems", *The Computer Journal*, 12, p.317 (1969)
- (11) A.V.Aho, J.E.Hopcroft and J.D.Ullman, "The design and analysis of computer algorithms", Addison Wesley (1974)
- (12) R.M.Karp, "Reducibility among combinatorial problems", *Complexity of Computer Computations* (edited by R.E. Miller and J.W.Tatcher), Plenum Press, N.Y., p.85 (1972)
- (13) M.R.Garey and D.S.Johnson, "Strong NP-Completeness results :Motivation, examples and implications", *J. of ACM*, 25, p,499 (1978)
- (14) S.K.Sahni, "General techniques for combinatorial approximation", *Oper. Res.*, 25, p.920 (1977)
- (15) J. M. Moore, "An n job, one machine sequencing algorithm for minimizing the number of tardy jobs", *Manege. Sci.*, 15, p.102 (1968)
- (16) J. Bruno, E.G.Coffman and R.Sethi, "Scheduling independent tasks to reduce mean finishing time", *Comm. of ACM*, 17, p.382 (1974)
- (17) E.L.Lawler, "A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness", *Ann. Discrete Math.*, 1, p.331 (1977)
- (18) 川口, 中野, 中西, "重み付き平均滞留時間最小化スケジューリング問題の解法", *信学技報*, CST 78-138 (1979-1)
- (19) S.K.Sahni, "Algorithms for scheduling independent tasks", *J. of ACM*, 23, p.116 (1976)
- (20) E.L.Lawler, "Sequencing to minimize the weighted

- number of tardy jobs", *Rev. Franc. Automat. Informa. Rech. Oper.* 10, p. 27 (1976)
- (21) T. Gonzalez, "Preemptive scheduling of uniform process-  
or systems", *J. of ACM*, 25, p.92 (1978)
- (22) C. Martel, "Generalized network flows and application  
to multiprocessor scheduling", Ph.D. thesis, Univ.  
California Berkeley (1979)
- (23) H. Kise, T. Ibaraki and H. Mine, "A solvable case of  
the one-machine scheduling problem with ready and due  
times", *Oper. Res.*, 26, p.121 (1978)
- (24) S. Lakshminarayan, R. Lakshmanan, R.L. Papineau and  
R. Rochette, "Optimal single-machine scheduling with  
earliness and tardiness penalties", *Oper. Res.*, 26,  
p.1079 (1978)
- (25) R.L. Graham, "Bounds on multiprocessing timing anomal-  
ies", *SIAM J. Appl. Math.*, 17, p.416 (1969)
- (26) W.L. Eastman, S. Even and I.M. Issacs, "Bounds for the  
optimal scheduling of n jobs on m processors", *Manege.  
Sci.*, 11, p.268 (1964)
- (27) A.K. Chandra and C.K. Wong, "Worst case analysis of a  
placement algorithm related to storage allocation",  
*SIAM J. Comp.*, 4, p.249 (1975)
- (28) M. Held and R.M. Karp, "The traveling salesman problem  
and minimum spanning trees: Part I", *Oper. Res.*, 18,  
p.1138 (1970)
- (29) M. Held and R.M. Karp, "The traveling salesman problem  
and minimum spanning trees: Part II", *Math. Program.*,  
1, p.6 (1971)

- (30) J. Kennington and M. Shalaby, "An efficient subgradient procedure for minimal cost multicommodity flow problems", *Manege. Sci.*, 23, p.994 (1977)
- (31) M. Held, P. Wolfe and H. Crowder, "Validation of subgradient optimization", *Math. Program.*, 6, p.62 (1974)
- (32) M. L. Fisher and J. F. Shapiro, "Constructive duality in integer programming", *SIAM J. Appl. Math.*, 27, p.31 (1974)
- (33) M. L. Fisher, W. D. Northup and J. F. Shapiro, "Using duality to solve discrete optimization problems: Theory and computational experience", *Math. Program. Study*, 3, p.56 (1975)
- (34) M. L. Fisher, "A dual algorithm for the one-machine scheduling problem", *Math. Program.*, 11, p.229 (1976)
- (35) 川口, 中野, 中西, 新見, "WMFT問題に対する近似解法", 信学会総合全国大会, 32 (1980)
- (36) 新見, "スケジューリング問題の解法に関する研究", 大阪大学工学部卒業論文 (1979).
- (37) M. R. Garey and R. L. Graham, "Bounds for multiprocessing scheduling with resource constraints", *SIAM J. Comp.*, 4, p.187 (1975)
- (38) D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey and R. L. Graham, "Worst case performance bounds for simple one-dimensional packing algorithm", *SIAM J. Comp.*, 3, p.299 (1974)
- (39) N. Christofides, "Graph theory: An algorithmic approach", Academic Press (1975)
- (40) G. R. Grimmett and C. J. H. MacDiarmid, "On colouring

- random graphs", Math. Proc. Camb. Phil. Soc., 77, p.313  
(1975)
- (41) R. E. Tarjan and A. E. Trojanowski, "Finding a maximum independent set", SIAM J. Comp., 6, p.537 (1977)
- (42) F. Gavril, "Algorithms for minimum coloring, maximum clique, minimum covering by cliques and maximum independent set of a chordal graph", SIAM J. Comp., 1, p.180 (1972)
- (43) F. Gavril "Algorithms on circular-arc graphs", Network, 4, p.357 (1974)
- (44) M. C. Golumbic, "The complexity of comparability graph recognition and coloring", Computing, 18, p.199 (1977)
- (45) 松山, "節点独立集合問題の近似解法に関する研究", 大阪大学工学部卒業論文 (1980)
- (46) 森田, "統計数理入門", 日本評論社 (1968)
- (47) R. M. Karp, "A patching algorithm for the non-symmetric traveling salesman problem", SIAM J. Comp., 8, p.561 (1979)
- (48) B. Bollobas, "Graph theory "(Graduate texts in mathematics 63), Springer-Verlag N.Y. (1979)