経営科学（日本オペレーションズ・リサーチ学会邦文機関誌） 第17巻 第6号（1973年11月）

<総 合 報 告>

スケジューリングの展望†

山 本 正 明*

1. はじめに

一般に“スケジューリング”という場合，かなり広範囲な計画問題を含んでいるわけだが，これらの問題に共通した点は「計画対象を完成させるまでに必要な一連の作業群の時間軸上での位置を決めること」にある．この小論での展望の範囲も，時間軸上で位置を決める形の問題に限定したい．

時間軸上での位置を決める場合，普通次の二つの方法が考えられる．

1. 時点計画——すべての作業の開始時刻，終了時刻を与える．
2. 期間計画——時間軸を一定間隔の区間に分け，それぞれの仕事をその区間に割り付ける．

ORの分野でスケジューリングという場合には，時点計画を対象とすることが多いので，この小論の話題も時点計画に片寄るが，実用的な立場からいうと期間計画のほうが役立っているようなので，これについてはスケジューリング理論の実用性を論ずるところで触れたい．

時点計画を考える場合，作業を時間軸上に割り付ける際の多種多様な制約の中から，最も本質的なものをとり出し，できるだけこれを単純化してモデルに構成することが必要であった．このため，いわゆるジョブ・ショップ問題が多くのOR研究者の興味の対象となり，1954年，Johnson [16] が2台の機械でのスケジューリングの論文を発表して以降，かなりの数の論文がこの問題の解決のため提出されている．1967年，Conway, Maxwell, Miller が *Theory of scheduling* [6] を著して，スケジューリングの分野の体系化を試み，約200件の文献リストも添えているので，その頃までの研究成果は一応この本にまとめられている．この小論ではこの本以降の研究にできるだけ焦点を絞って，スケジューリングの分野での発展を展望してみようと思う．

2. ジョブ・ショップ問題

まず最初に，Conway らが単純ジョブ・ショップ過程と呼んだ [6]，最も基本的なモデルについて述べておこう．いま m 台の異なった機械をもつ工場で，n 個の異なった製品を製造する問題を考えることにする（以後 $m \times n$ 問題と呼ぶ）．各製品の製造には順序の定まった複数工程の作業

† 1973年9月4日受理．1973年3月，月例講演会講演要旨．

* 法政大学工学部．

を必要とし，各工程の作業に必要な機械の種類とその所要時間は一定で，あらかじめ与えられているものとする．そこでわれわれが求めたいものは，ある与えられた評価基準に関して最適なガント・チャート，すなわち各作業の時間軸上の位置である．

問題を単純化するために，このモデルでは次の仮定を置いている．

1) 製品に関しては，

1.1) 各製品の工程順序は固定している——代替機械を認めない．

1.2) 各製品はいったん工場にはいれば必ず完成される——キャンセル・不良等を考えない．

1.3) 各製品は一つの機械で加工を始めたら作業中断を考えない．

1.4) 各製品は直線的な加工工程からなる——分解型，組立型の工程を考えない．

1.5) 製品のある工程は，その直前の工程の作業が終了しなければ開始できない——ロットの並行処理は認めない．

1.6) 各機械間（工程間）での製品の機械待ちを認める．

1.7) 各製品は数工程からなり，そのおのおのは1台の機械だけで加工される．

1.8) 各製品はすべての機械で1工程ずつ作業が行なわれる——各製品とも m 工程となる[1]．

2) 機械に関しては，

2.1) 工場に同種の機械は1台しかない．

2.2) 各機械の能力は常に一定である——故障や保守は考えない．

2.3) 各機械は同時にたかだか一つの製品しか加工できない．

3) 加工時間に関しては，

3.1) 加工時間は既知で一定である．

3.2) 加工時間には運搬・取付け・取外しなどの時間が含まれているものと考える．

3.3) 段取がえの時間を考慮するときには，これが製品順序に無関係なものとする．

以上の仮定にはかなり非現実的なものも含まれているが，これらの仮定により問題がきわめて単純化され，定式化が容易になっている．あとに述べるように，これらの仮定を一つはずすごとにまったく新たなジョブ・ショップ問題が発生することになる．

図1は丸印で作業を，矢印で工程順序を表わした3×3のジョブ・ショップ問題を示しているが，この問題に対する一つの解として，図2のようなガント・チャートが書ける．このガント・チャート上では各機械の工程順序は一意的に定まってしまうが，逆に工程順序が決まればガント・チャートが定まるものと考えることができる[2]．よく知られているように，ジョブ・ショップ問題は各機械での製品の順序づけ問題（sequencing）となるのである．このときこれらの順序がとりうる場合の数は有限個であるから，この有限個の解の中から与えられた評価基準に対して

---

1) 仮定1.8）は，問題の形を整えるためだけの形式的なものである．

2) 一般にガント・チャート上で作業の割付け位置を右方へずらすことは常に可能であるが，左方へずらすことはある限界で不可能となる（製品順序が変わるような左移動は考えない）．この限界のスケジュールが与えられた順序に対応するものとする．

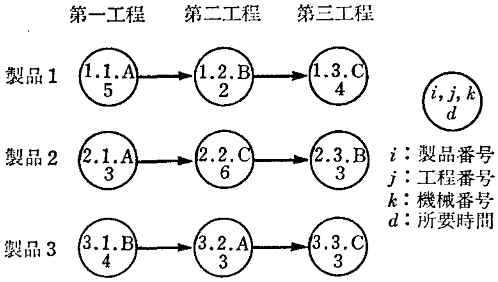

図1　3×3ジョブ・ショップ問題

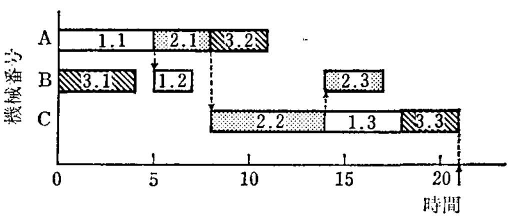

図2　ガント・チャートの一例

最適なものを選び出すことが問題となる．このとき用いる評価基準は，実際の必要にもとづいて与えればよいのだが，一例を上げると，

1) 全作業が完了するまでの総所要時間を最小にする
2) 各製品の工程滞留時間の総和を最小にする
3) 各製品ごとに与えられた納期に対して，納期遅れ時間の総和を最小にする
4) スケジュールに関連する諸コストを最小にする[14]

などが考えられるが，取扱いの容易なこともあってか，1)を用いた例が最も多い．

順序づけ問題の粗野ではあるが最も確実な解法は列挙法であろう．いま$m \times n$問題で，仮定1.8)を考慮に入れると，可能な順序の組合せの数は$(n!)^m$通りとなる．この$(n!)^m$通りのスケジュールをすべて調べあげれば最適解が得られるのは当然であるが，問題はその膨大な数にある3)．いまこの可能なすべての組合せを分類してみると，次のように分けることができる．

- 可能なすべての順序の組合せ
  - 非許容な順序
  - 許容な順序
    - 最適解を含まない順序
    - 最適解を含む順序
      - 非最適順序
      - 最適解を与える順序

非許容な順序というのは，作業の順序間にループが発生してガント・チャートへ割付けることができなくなった場合をいうもので，許容な順序を求める non-numerical な問題については，Akers と Friedman [1]，Marimont [19]，Driscoll と Suyemoto [8] などの報告がある．最適解を含むことが明らかで，要素数の小さい解の部分集合が得られれば，列挙法が改善されることは明らかであるが，Giffler と Thompson [11]は，アクティブ・スケジュールと呼ぶこのような集合を作ることに成功した4)．しかしアクティブ・スケジュールの大きさでも列挙法は実用的には問題となりえず，さらにこれを部分的な列挙ですませるために，Branch-and-bound 法を適用して計算量をへらす努力が行なわれている [4]．

3) たとえば，6 × 6 問題で$(6!)^6 = 1.35 \times 10^{17}$.
4) このようなスケジュール集合は，アクティブ・スケジュールしか得られていない．

順序づけ問題が整数線形計画法で定式化できることはよく知られているが，ジョブ・ショップ問題についても，Bowman [3]，Manne [18]らの定式化が報告されている．この場合，問題になるのは問題の大きさである．比較的コンパクトにまとまるManneの定式化を例にとっても，$m\times n$問題で(0－1)整数変数が$mn(n-1)/2$個，通常の非負の変数mn個，制約条件式$n(m-1)+mn(n-1)+n$個となり，実用的な大きさの問題でこれによってスケジュールを作製するのは絶望的に近い．しかしこれらの定式化が，スケジューリング問題の性質から持ち込まれた特殊な構造をもっていることに着目して，これを利用して計算量を節約する可能性は追求されるべきであろう．Greenberg [13] は，Manneの定式化の(0－1)変数の選択についてのtreeを考えて，Branch-and-bound法による最適解を探索する手順を提案している．

これまでは$m\times n$問題ということで一般的に話を進めてきたが，もっと限定された問題についての研究も数多い．

まず$m=1$の機械1台の問題であるが，この場合とりうる順序の数が$n!$となり問題を簡単化できることから，Smith [28] の論文以降かなりの数の研究が発表されている．仮定1.7) で製品が1工程だけのときは，各製品を機械グループ別に分けて独立に取り扱うことができるから，各グループごとに$m=1$の問題を解けばよい．

全製品の機械の加工順序がすべて等しいような問題は，フロー・ショップと呼ばれている．フロー・ショップ問題もジョブ・ショップ問題の一部と考えられるから，もちろん一般的に取り扱うことができるが，問題の制約がよりきびしいのだから，この点を利用したより効率の良い解法が考えられている．直観的に考えて，すべての加工順序が同じだから逆に各機械での製品の加工順序も同じになるのではないかと考えられる．そうすれば可能な順序は$n!$になり，$m=1$のときと同じになる．厳密にいうとこれは$m\leqq 3$までしか成り立たない[5]．

つぎに述べておきたいことは，ジョブ・ショップのモデルではスケジューリングすべき仕事はあらかじめ全部与えられていて，前から仕掛中の仕事をまったく持たない工場でこれを加工する計画をたてることになっている．しかし実際の工場では製品(注文)がつぎつぎにはいってきて，工場は連続して動いている．Conwayらはこれをダイナミックなモデルと呼んで[6]，前者の静的なモデルと区別している．ジョブ・ショップをダイナミックなモデルとして取り扱おうとすると，注文を連続して到着する客，機械をそれに対するサービス施設と考えた待ち行列のモデルとなる．$m=1$の場合を除けば一般に待ち行列のネットワークを考えなくてはならない．このモデルでは注文はその到着の統計的性質のみが与えられ，個々の注文は識別できないのだから，それを時間軸に割り付けるなどということは問題とならない．ダイナミック・モデルは1で述べたような意味でのスケジューリングの問題にはならない．それはスケジューリングにあたっての一般的規則の検討などに利用できるだけである[6]．たとえば，個々の機械での加工順序を局所的に決めていくディスパッチング・ルールの性能検討などに用いられている．実際の工場がダイナミッ

5) 総所要時間の最小化を目的関数としたとき，[6]の第5章参照．
6) 実際には，すこし複雑になるとシミュレーション手法を利用せざるをえない．

クに動いているといっても，計画を作製するときはその時点でのスタティックな状況から決定されるのだから，状況に応じて敏速に再計画が可能なシステムが問題なのであって，この点では静的なモデルで十分である．

前にも述べたように，単純ジョブ・ショップ過程における仮定を緩めると，それに応じて新しいジョブ・ショップ問題が発生する．そのなかでいくつかの注目すべき研究を紹介しておこう．

仮定 1.3） ……作業中断を許す場合

あとから機械に到着したより緊急度の高い作業を実行するため，加工中の他の作業を一度中断してあとに回すことにより，スケジュールの改善をはかることが許される場合である．Schrage [26]は，次に述べる一般スケジューリング問題の中で，この形の問題を扱っている．

仮定 1.4） ……工程順序をネットワークで表現して，より複雑な工程を一般的に取り扱う場合

仮定 1.4）は部品加工工場での実際の状況の反映であろうが，一方，仮定 1.8）とともに問題をマトリックス表示により簡単に表わすことを可能にしている．しかし，計算機によるネットワークの処理が容易になったこともあり，工程順序をネットワークで表現して問題を取り扱うほうがより広い範囲の問題を一般化して扱うことができ，最近の論文にはこの形のものが多い．この形のモデルは一般スケジューリング問題（general scheduling problem）と呼ばれており，3で述べる．

仮定 1.5） ……ロットの平行処理を認める場合

ロット生産の場合，前の工程で一定時間作業が進むと，その工程が完全に終了しなくても次の工程の作業を開始させる場合がある．すなわち，相並ぶ二つの工程の開始時刻・終了時刻に一定の時間差が要求されるだけで，仮定 1.5）は成り立ない場合である．Mitten [20]，鍋島 [22] らは，このような条件をもつフローショップについてのスケジューリングを検討している．

仮定 1.7）と 2.1） ……複数機械による加工や工場での複数機械の存在を考える場合

単純ジョブ・ショップ過程では，一台一台の機械はすべて異なったものとして識別される．もしこの中に同種の機械があれば，それぞれ識別されるものと考えて，各機械に作業をあらかじめ配分しておけば，単純ジョブ・ショップ過程と同じ形の問題となる[7]．同種の機械台数が 2, 3 台ということなら，単純ジョブ・ショップ問題の考え方をそのまま拡張する余地もあるが，一般的に複数機械問題を取り扱うには，総利用可能機械台数の制約の下で，各作業への機械配分を考慮してスケジュールを作る問題を考える必要が起こる．この形の問題は資源配分問題（resource allocation）と呼ばれており，4で説明する．

仮定 3.3） ……段取がえ時間が製品順序によって影響を受ける場合

機械 1 台だけのとき（$m=1$）は，段取がえ時間の総和を評価基準に考えれば，巡回セールスマン問題として取り扱うことができる．しかし，$m>1$ で製品の工程順序を考慮しなければならな

7）作業を各機械に配分する組合せの数だけのジョブ・ショップ問題が発生するわけだから，計算量はさらに増大する．

い場合は，ジョブ・ショップ形の順序づけ問題も同時に考えねばならず，かなり複雑な問題となる．

3. 一般スケジューリング問題

ジョブ・ショップ問題では，製品の工程として直線的につながるものだけを考えていたが，これを一般化して，作業間の順序関係がネットワークの形で与えられるようにすれば，より広い範囲の問題を取り扱うことができる．図1で示した3×3問題の場合，ダミー作業として開始作業s，終了作業tを考えて，図3のように3作業の始まりと終わりをs，tにまとめれば，ノードが作業，アークが作業間の順序関係を与えるネットワークで表現できる．

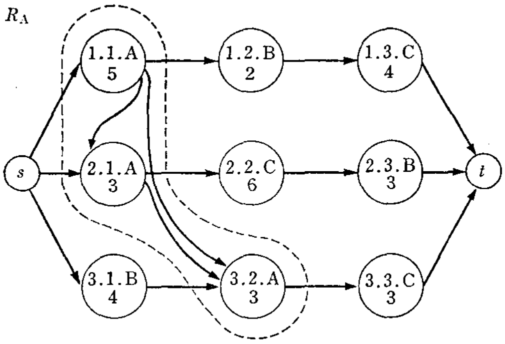

図3 ジョブ・ショップ問題のネットワーク表示
$R_A=\{1.1, 2.1, 3.2\}$
$R_B=\{1.2, 2.3, 3.1\}$
$R_C=\{1.3, 2.2, 3.3\}$

まず山本[31]により一般スケジューリング問題の定式化を示そう．図3で示すような作業のネットワークをグラフGで与える．

(1) $G=(N, A)$

ここでNは作業の集合，Aは作業間の技術的順序関係に対応したアークの集合を意味する．各作業の所要時間d_iはノードiに与えられている[8)]．機械kを必要とする作業の集合を共通資源集合R_kと呼ぶと，R_kに属する任意の二つの作業$i, j\in R_k$の間には，(i, j)か(j, i)かいずれかの順序関係が成り立たねばならない（2の仮定2.1)，2.3)より）．このような資源の制約によって導入される順序関係を資源順序関係と呼ぶことにする．$R_k\ (k=1, 2, \cdots, m)$に属する作業から作られるすべての作業対に資源順序関係が指定されると，資源に関する制約は満足される．このようにして選ばれた資源順序関係の集合をBとすると，

(2) $G=(N, A\cup B)$

にループがなければGは許容スケジュールSを与える．目的関数としてこのネットワークのクリティカルパスの長さλを選ぶと，問題は$S^0=(N, A), R_k, d_i$が与えられているとき，λを最小にするような$S=(N, A\cup B)$を選択する問題となる．

Balas [2] は disjunctive graph という概念を用いて，上で述べた資源順序関係を表わしている．グラフ中の2本のアークが，グラフ中のいかなるパスもこのアークのうちたかだか1本しか通ることができないような場合，disjunctive pair をなすといい，このようなアークを含むグラフを disjunctive graph と呼んでいる．図4は図3と同じ内容を disjunctive graph で表現したも

8) Balas [2] らの disjunctive graph では，所要時間はその作業に対応するノードから始まるすべてのアークに与えている．

ので，点線で示したアークの正逆いずれか一方だけが成り立つことにより，資源順序関係を表わすことができる．

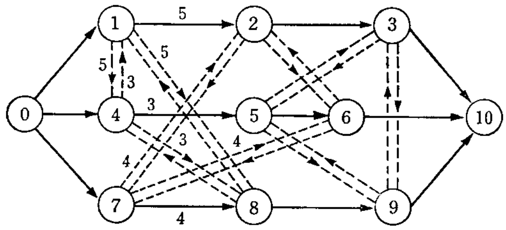

図4 disjunctive graph による表示

近年発表されている一般スケジューリング問題に対する解法は，ほとんどBranch-and-bound法によるスケジュールの部分列挙によるものである．この場合，すべての可能なスケジュールを列挙するために，スケジュールの部分集合を分割していく branching の構成（樹によって表現される）には大きく分けて二つのタイプがある．

1) 同一機械を使う共通資源集合の中のすべての作業対に対して正逆いずれかの順序を指定する．Balas [2] は，すべての disjunctive arc にあらかじめ一定方向の順序を指定した上で，各アークを順次逆転させることによって branching を構成した．山本 [31] は，資源順序関係のはいっていない $G=(N, A)$ から出発して，一定の順序で逐次資源順序関係を導入することにより樹を構成するアルゴリズムを提案している．disjunctive arc の中には，スケジュールにとって制約としてきいてこない redundant なものがかなり含まれていることに着目して，Charlton と Death [5] や鍋島 [23] は，制約として意味のあるノードだけで樹を構成することにより，探索量をへらす方法を発表している．

2) 時間的に早くスケジュール可能なものから割り付けていって，各時点で特定機械を競合する作業集合に対してのみ順序関係を指定していく．資源順序関係の中で実際にはきいてこない redundant なものを排除するには，Giffler と Thompson [11] がアクティブ・スケジュールで考えたように，時間軸に割り付けながらコンフリクトを起こした作業間の順序だけを考えていくのが樹のノード数を最も少なくする．山本[30]は，コンフリクトを起こした作業間に正逆の資源順序対を入れることにより，branching を構成するアルゴリズムを提案している．Florian [10] らは，未割付作業と割付済作業とのcutという概念で同様な樹の構成を試みている．Schrage[25], [26] は，作業集合 N の要素による順列とガント・チャート上への各作業の割付けとの対応関係を定めることにより，最適スケジュールを探索するための樹を構成している．

(2)のほうが樹のノードの数は当然少なくなるが，(1), (2)いずれが有利であるかは，同時に用いるバウンドの効率や計算機のプログラミング上の問題ともからんで，優劣を決める段階にはない．

つぎに Branch-and-bound 法のバウンドの問題であるが，樹のあるノードでの，すなわちスケジュールの部分集合の目的関数の下限（最小化問題のとき）を求めることが必要である．枝分かれの初期の段階でその部分集合の真の下限に近い値を推定することができれば，樹の探索の効率は向上できるが，一般的にいえば，バウンドの性能を上げようとすれば計算量が可速度的に大きくなり，必ずしも全体的には有利でない．クリティカル・パスの長さを目的関数とした問題では，

1) 技術的順序関係によるパスの長さ

2) 各機械ごとの資源順序関係によるパスの長さの推定値

3) 1) と 2) の複合形

がバウンドに用いられており，上に紹介した論文でもほとんど大同小異のものを用いている．むしろ(1)の形の branching を用いた場合には，樹のノードを構成する作業対を選択する順序のほうが総探索量に影響が大きいことが指摘されている [31].

これらのアルゴリズムを用いて実際に問題を解いた経験は，せいぜい 10×10 問題ぐらいまでのものしか報告されていない．表1は Fischer と Thompson が用いたテスト問題 [9] をいくつかの方法で解いた結果を示している．表から明らかなように，現実的な規模の大きさの問題を解くのにはまだかなり距離があるといえよう．そこで，Branch-and-bound 法で最初に見つかった解，あるいは一定時間の探索のうちで最良の解が真の最適解にどのくらい近づいているかを実験的に検討した上で，これらを近似解として利用するのも賢明な方法といえよう．

これまで述べてきた一般スケジューリング問題をいっそう拡張したものとしては，Gorenstein [12] が minimal number of chain の概念を利用して複数機械問題を扱っており，Schrage [26] は，作業の中断を許した場合にも [25] と同じような branching ができるように工夫をして，この形の問題を解いている．

表1 一般スケジューリング問題の計算例2)

			初期解		最適解(または最良解)		
問題	$m \times n$	方法1)	ステップ数3)	λの値	ステップ数	λの値	総ステップ数
1	6 × 6	I	36	55	55	36	186
		II	36	59	233	55	241
		III	—	152	41	55	200*
		IV	90	58	331	55	830
2	10 × 10	I	107	1193	923	1156	2000*
		II	100	1041	100	1041	2000*
		III	-	3394	156	1177	200*
		IV	450	1117	1222	1068	1360*
3	5 × 20	I	100	1306	1539	1302	4000*
		II	100	1252	124	1222	2000*
		III	—	3218	89	1231	200*

* 計算が途中で打ち切られた．したがって最適性は保証できず，それまでに得られた最良解が示されている．

1) 方法 I Schrage [25] のアルゴリズムによる
II Florian [10] のアルゴリズムによる
III Balas [2] のアルゴリズムによる
IV 山本 [31] のアルゴリズムによる

2) I，II，III は文献 [10] にある計算結果を用いてある．

3) ステップ数の意味はアルゴリズムによってまったく異なるので，方法間の比較は意味がない．

4. 資源配分問題

複数機械問題を一般的に取り扱うには，総利用可能機械台数の制約の下で，各作業への機械配分を考慮してスケジューリングを行なう問題を考えねばならない．この種の資源配分問題では，機械台数がかなり多い場合を考えているので，ジョブ・ショップ問題のように，個々の機械内での製品順序を考えると組合せの数が膨大となって得策ではない[9]．この場合は，各スケジュール時刻ごとに必要資源量が総利用可能資源量の制約を満たすような許容スケジュールを考えるのが普通である．

問題の性格を明らかにするため，Pritsker ら [24] が Bowman [3] の定式化を拡張して，（0−1）整数計画法による定式化を試みているので，これを紹介しておこう．いま第 i プロジェクト $(i=1,2,\cdots,I)$ の第 j 作業 $(j=1,2,\cdots,N_j)$ に所要時間 d_{ij}，到着時刻 a_{ij}，第 k 資源 $(k=1,2,\cdots,K)$ の必要量 r_{ijk} が与えられているものとする．プロジェクト内の各作業の技術的順序関係は与えられているから，各作業の最早終了時刻 l_{ij} は計算でき，それによってプロジェクトの最早完了時刻 e_i も決まる．また各プロジェクトの絶体納期 G_i を与えれば，これに対する最遅終了時刻 u_{ij} も計算できる．いま変数として

(3) $x_{ijt}=1$ （時刻 t に作業 ij が完了したとき）

$\quad\quad =0$ （それ以外のとき）

(4) $x_{it}=1$ （プロジェクトのすべての作業が時刻 t までに完了しているとき）

$\quad\quad =0$ （それ以外のとき）

ここで x_{ijt} が意味をもつ時間域は，

(5) $l_{ij} \leqq t \leqq u_{ij}$

であり，x_{it} が意味をもつ時間域は，

(6) $e_i \leqq t \leqq G_i$

である．いま各作業の完了時刻に対しては，

(7) $$\sum_{t=l_{ij}}^{u_{ij}-1} x_{ijt} \leqq 1 \qquad (i=1,2,\cdots,I;\ j=1,2,\cdots,N_i)$$

が，各プロジェクトの完了時刻に対しては

(8) $$x_{it} \leqq (1/N_i)\sum_{j=1}^{N_i}\sum_{q=l_{ij}}^{t-1} x_{ijq} \qquad (i=1,2,\cdots,I;\ t=e_i,e_i+1,\cdots,G_i)$$

が成り立たねばならない．技術的順序関係については，作業 im が作業 in に先行するとき，

(9) $$\sum_{t=l_{im}}^{u_{im}} t x_{imt} + d_{in} \leqq \sum_{t=l_{in}}^{u_{in}} t x_{int}$$

が成り立つ．資源の制約に関しては

9) 機械台数でいうよりは，各作業の必要人数で話したほうがこの問題の性格にはふさわしい．

$$(10)\quad \sum_{i=1}^{I}\sum_{j=1}^{N_i}\sum_{q=t}^{t+d_{ij}-1} r_{ijk}x_{ijq} \leqq R_{kt}\ (k=1,2,\cdots,K;\ t=\min a_{ij},\cdots,\max G_i)$$

が必要である．ここで R_{kt} は時刻 t で資源 k の利用可能量である．目的関数として全プロジェクトの完了時刻を最小にすることを考えると，変数 x_t を導入して，

$$(11)\quad x_t = 1 \quad \text{(時刻 } t \text{ ですべてのプロジェクトが完了しているとき)}$$
$$\qquad = 0 \quad \text{(それ以外のとき)}$$

$$(12)\quad x_t \leqq (1/\sum_{i=1}^{I}N_i)\sum_{i=1}^{I}\sum_{j=1}^{N_i}\sum_{q=l_{ij}}^{t-1} x_{ijq} \qquad (t=\max l_i,\cdots,\max G_i)$$

のとき，

$$(13)\quad Z=\sum_{t=\max l_i}^{\max G_i} x_t \longrightarrow \min$$

となる．Pritsker らの興味は，これらの定式化での変数・制約式の数をへらすことにあり，3プロジェクトと8作業，3資源の例題で(0－1)変数が33個，制約式37本になっている．この程度の問題なら，現在利用できる整数計画法コードでも解くことができるが，一般には整数計画で解くことは無理であろう．

はじめに述べたように，資源配分問題では，各スケジュール時刻での資源に対する許容性を保証できるスケジュールを作ればよいのだから，3で一般スケジューリング問題の branching の二つのタイプを述べたときの(2)の考え方は，資源配分問題に拡張できる可能性がある．山本 [30] の場合は，むしろ資源配分問題の特殊ケースとして一般スケジューリング問題を解いている．

Davis と Heidorn [7] は，作業を1日ずつの直列の作業に分割することによって，資源配分問題がラインバランシング問題と類似の問題になることを示し，ラインバランシング問題の解法を利用したアルゴリズムを発表している．

資源配分問題も，ジョブ・ショップ問題と同様に組合せ型の問題であり，実用的には近似解の利用が避けられない．Wiest [29] は，比較的大きなプロジェクトに対する heuristic な解法を研究している．PERT 系の手法では MANSCHEDULING [27] や RAMPS [17] が資源配分問題のためのプログラムとしてよく知られている．

5. スケジューリング理論の問題点

スケジューリングの理論を実際に使うという立場から検討すると，非常に問題点が多く，OR 学会のスケジューリング応用研究部会の終了報告 [15] をみても，この点についてはかなり悲観的な見方をしているようである．

5・1 問題の大きさのギャップ

前節までにくり返し述べてきたように，スケジューリング問題の最大の障壁はその膨大な計算量であろう．スケジューリング理論によって実際に解ける問題の大きさと，実用上要求される問題の大きさとのギャップをどの程度まで埋めることができるかについての見通しはかなり厳し

い．しかし，実用上は最適解にそれほどこだわる必要はないと考えられるから，より大きい問題を効率よく近似的に解く方法をもっと意識的に追求する必要があるように思う．見方を変えれば，最適解を求めるアルゴリズムの研究も，より良い近似解を考えるためのステップと考えることもできる．また，実際の問題は大きいといっても，なんらかの形の分割が可能で，部分問題ごとの計算をすることによって全体のスケジューリングが可能な場合も考えてみる必要があるだろう．

5・2 時点計画法の実用性

現実のジョブ・ショップでは工程の変動要因が多すぎて10)，すべての作業の開始・終了時刻を決めてしまう時点計画のような“かたい”計画では対応しきれない場合が多い．もしこれをあえて使おうとすれば，非常に短いサイクルでの再スケジューリングが要求される．このために管理費用は増大し，しかも頻繁な計画変更に振り回されて，現場はかえって混乱するおそれがある．現状ではより“ソフトな”期間計画のほうが使いやすい面が多い．しかし工程の変動要因は，その機械化・自動化の進行とともに減少する傾向にあり，時点計画法の必要性は高まるものと考えられる．

期間計画法にも目的と状況に応じて種々のタイプがあるが，本質的にはいくつかの計画区間にそれぞれの仕事を割り付けるある種の配分モデルであり，時点計画のような順序に関する制約は考えないで済むので，問題としては簡単となる11)．

5・3 スケジューリングのシステム化

前項で述べたように，スケジューリングの実用化を考えるとき，管理情報の伝達・処理のスピードアップと結びつかなくては，現実の変動に迅速に対処していくことはできない．一方，コンピュータをベースにした総合的な管理システムを考えた場合，なんらかの時間軸上での計画は必ずたてねばならず，このスケジューリング・システムが全管理システムの中で中心的役割を果たす．したがって効率の良いスケジューリング・システムに対する要求は大きい．このような現実的ニーズがスケジューリングの理論の新しい展開を約束しているものといえよう．

参 考 文 献

[1] Akers, S. B. and J. Friedman, "A non-numerical approach to production scheduling problems," *O. R.*, **3**, 4(1955), 429–442.

[2] Balas, E., "Machine sequencing via disjunctive graphs: an implicit enumeration algorithm," *O. R.*, **17**, 6 (1969), 941–957.

[3] Bowman, E. H., "The schedule-sequencing problem," *O. R.*, **7**, 5 (1959), 621–624.

[4] Brooks, G. H. and C. R. White, "An algorithm for finding optimal or near-optimal solutions to the production scheduling problem," *Journal of IE*, **16**, 1 (1965), 34–40.

[5] Charlton, J. M. and C. C. Death, "A generalized machine-scheduling algorithm," *O. R. Quart.*, **21**, 1 (1970), 127–134.

10) 2の仮定1.2)，2.2)，3.1)が成り立たない．また新たな注文の到着が常に起こる．

11) 一つの製品がいくつかの計画区間にまたがるような場合には，能力の制限を考えれば，やはり順序に関する制約が必要となる．

[6] Conway, R. W., W. L. Maxwell and L. W. Miller, *Theory of scheduling*, Addison-Wesley, 1967 (関根智明監訳, スケジューリングの理論, 日刊工業).
[7] Davis, E. W. and G. E. Heidorn, "An algorithm for optimal project scheduling under multiple resource constraints," *Management Science*, **17**, 12 (1971), B-803-816.
[8] Driscoll, L. C. and L, Suyemoto, "Heuristics for resolution of logical scheduling conflict," *4th Int'l conf. of OR* (1966), F-I—93-129.
[9] Fischer, H. and G. L. Thompson, "Probabilistic Learning Combinations of Local Job-shop Scheduling Rules,"chap. 15 of [21].
[10] Florian, M., P. Trepant and G. Mcmahon, "A implicit enumeration algorithm for the machine sequencing problem,"*Management Science*, **17**, 12 (1971), B—782-792.
[11] Giffler, B. and G. L. Thompson, "Algorithms for solving production scheduling problem," *O. R.*, **8**, 4 (1960), 487-503.
[12] Gorenstein, S. "An algorithm for project (Job) sequencing with resource constraints," *O. R.*, **20**, 4 (1972), 835-850.
[13] Greenberg, H. H.,"A branch-bound solution to the general scheduling problem," *O. R.*, **16**, 2 (1968), 353-361.
[14] Gupta, J. N. D.,"Economic aspects of production scheduling systems," *JORSJ*, **13**, 4 (1971), 169-193.
[15] 原　亨, 下城康世, "スケジューリング・システムにおける問題点", 経営科学, **16**, 4 (1972), 233-235.
[16] Johnson, S. M., "Optimal Two-and-Three-Stage Production Schedules with Setup Times Included," *Nav. Res. Log. Quart.*, **1**, 1 (1954), 61-68.
[17] Lambourn, S., "Resource allocation and multi-project scheduling (RAMPS)-a new tool in planning and control," *Computer Journal*, **5** (1963), 300-304.
[18] Manne, A. S.,"On the job-shop scheduling problem," *O. R.*, **8**, 2 (1960), 219-223.
[19] Marimont, R. B.,"A new method of checking the consistency of precedence matrices," *Journal of A. C. M*, **6**, 2 (1959), 164-171.
[20] Mitten, L. G., "Sequencing n job on two machines with arbitary time lags," *Management Science*, **5**, 3 (1959), 293-298.
[21] Muth, J. F. and G. L. Thompson, eds., *Industrial Scheduling*, Prentice-Hall, 1963 (関根智明訳, インダストリアル・スケジューリング, 竹内書店).
[22] Nabeshima, I., "Sequencing of two machines with start lag and stop lag," *JORSJ*, **5**, 3 (1963).
[23] ————, "General scheduling algorithms with applications to parallel scheduling and multiprogramming scheduling," *JORSJ*, **14**, 2 (1971), 72-99.
[24] Pritsker, A. A. B., L. J. Watters and P. M. Wolfe, "Multiproject scheduling with limited resource: A zero-one programming approach,"*Management Science*, **16**, 1 (1969), 93-108.
[25] Schrage, L.,"Solving resource-constrained network problems by implicit enumeration—Nonpreemptive case," *O. R.*, **18**, 2 (1970), 263-278.
[26] ————, "Solving resource-constrained network problems by implicit enumeration—Preemptive case," *O. R.*, **20** (1972) 668-677.
[27] Shriley, W. W. and I. L. Bowman, 8000, Critical path, and manscheduling, share general program library.
[28] Smith, W. E.,"Various optimizers for single-state production," *Nav. Res. Log. Quart.*, **3**, 1 (1956), 59-66.
[29] Wiest, J. D., "A heuristic model for scheduling large projects with limited resources," *Management Science*, **13** (1967), B-359-377.
[30] 山本正明, "プロジェクト・ネットワーク上での人員機械配分計画法 (第2部)", 経営科学, **11**, 2 (1968), 105-124.
[31] Yamamoto, M.,"An algorithm for general scheduling problem," *JORSJ*, **16**, 2 (1973), 65-77.