

ある種のスケジューリング問題に対する アルゴリズム†

三 根 久*
茨 木 俊 秀*
木 瀬 洋**

1. はじめに

ここではジョブ・ショップ・スケジューリングの特殊な場合について考察する [8]. すなわち、処理されねばならない n 種の仕事がある. 各仕事は準備, 加工および整理工程からなり, かつ, その順序で処理される. ここで準備工程は, 資材の発注などの加工工程の前に必要な工程である. 加工工程では資材を製品に加工する. 整理工程は製品の発送などの加工後に必要な工程である.

すべての仕事の加工工程は1台の機械で処理されねばならない. 他方, 準備工程には, それぞれ1工程に1台ずつ機械が与えられているので, それぞれの準備工程が互いに干渉することはない. 整理工程についても同様である. この場合, 時刻0から開始してすべての仕事の終了時刻 (make-span) を最小にするように, 加工機械上での n 種の仕事の最適順序を求める問題を考える.

簡単な例によって, この問題が単純でないことを示す.

3種の仕事 i, j, k がある. それぞれの準備, 加工および整理工程の処理時間を表1に示す. この例の実行可能な解は, $3! = 6$ 通りの順序であるが, それらの中で最小終了時刻をもたらす順序 (最適解) を見いだそう. 直感的に有効であると考えられる既存の3種の手法を適用する. 準備時間の短い仕事から先に処理

表1 仕事の例

仕 事	準備時間	加工時間	整理時間
i	10	4	11
j	15	6	12
k	20	1	18

する規則 (SPT Rule) に従ったときの仕事の順序は $i < j < k$ である. そのときの終了時刻は 40. 整理時間の長い仕事から先に処理する規則 (LPT Rule) に従ったときの仕事の順序は $k < j < i$. そのときの終了時刻は 42 である. 最後に2台の機械を持つフロー・ショップに対する Johnson [6]

† 1973年10月31日受理. 1973年4月8日, 春季研究発表会要旨.

* 京都大学工学部.

** 京都工芸繊維大学工芸学部機械工学科.

1) $i < j$ は i が j に先行することを表わす. 明らかに, $i < j$ で $j < k$ ならば, $i < k$.

の方法と類似の規則を考える。まず、全仕事の準備時間および（加工＋整理）時間の中で最も小さい値に対する仕事を選ぶ。もし、その値が準備時間であれば、その仕事を最初にする。もし、その値が（加工＋整理）時間であれば、その仕事を最後にする。残った仕事に対し、すべての仕事の順序が決まるまで上の手法をくり返す。この方法を適用したときの仕事の処理順序は $i < j < k$ である。

しかしながら、上の3手法で得られた順序はいずれも最適ではない。この例の最適順序は $i < k < j$ で、そのときの終了時刻は 39 である。

本文では、この種の問題に対して分枝限定法 (Branch-and-bound method) に基づくアルゴリズムを提案する。分枝限定法は比較的規模の小さいジョブ・ショップ・スケジューリング問題 [3] や行商人問題 [4] に対して非常に有効な方法であるが、計算時間が問題の規模に対して指数関数的に増大する傾向がある [11]。この欠点を緩和するため、本文ではこの種の問題の最適解に対する若干の定理を示し、最適解の探索範囲の縮小を試みる。そして最後に、提案したアルゴリズムの有効性を調べるために行なった数値実験の結果を示す。

2. Disjunctive Graph

上述の問題を P と記す。 P をグラフ表示するため、次に disjunctive graph [1] G を導入する。

仕事 $i=1, 2, \dots, n$ の集合を N と記す。

$$(1) \quad N = \{1, 2, \dots, n\}$$

G はすべての仕事の開始と終了を表わす節点 s と t および仕事 $i \in N$ の加工開始時点を表わす節点 i の集合（仕事 i と一対一対応するので、これも N で表わす）と次に示すアークの集合からなる。

(i) 仕事 i の準備工程を表わす長さ $a_i (\geq 0)$ のアーク $(s, i)^2$ 。 a_i は準備時間である。

(ii) N に属する任意の2節点 i と j に対する disjunctive pair。すなわち、disjunctive arc (i, j) と disjunctive arc (j, i) 。それらの長さはそれぞれ仕事 i の加工時間 $b_i (\geq 0)$ と仕事 j の加工時間 $b_j (\geq 0)$ である。加工工程の順序づけを行なう場合には、各 disjunctive pair のどちらか一方が選ばれ、他方が除かれる。そのとき、disjunctive arc (i, j) が選ばれれば、 $i < j$ 、また disjunctive arc (j, i) が選ばれれば、 $j < i$ を意味する。

(iii) 仕事 $i \in N$ の加工＋整理を表わすアーク (i, t) 。アーク (i, t) の長さを c_i と記すとき、

$$(2) \quad c_i = b_i + c_i (\geq c_i)$$

ただし、 $c_i (\geq 0)$ は整理時間である。

(iv) アーク (s, t) 。アーク (s, t) の長さを d と記すとき、

$$(3) \quad d = 0$$

アーク (s, t) はここでは直接意味を持たないが、3節の部分問題で必要となる。

註2) 以下では常に、問題を表わすグラフの始節点を s 、終節点を t で表わすものとする。

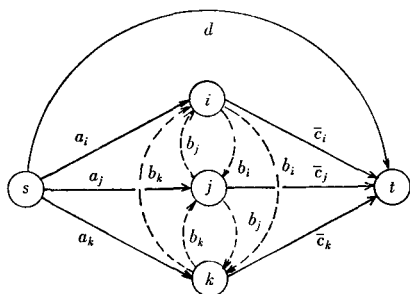


図1 Disjunctive graph G

の一方を選ぶ問題（以下、これを問題Gと記す）に帰着する。

証明： 任意の加工工程の順序, $q_1 < q_2 < \dots < q_n$ を,

$$(4) S_q = [q_1, q_2, \dots, q_n], \quad q_1, q_2, \dots, q_n \in N$$

で表わす。 S_q と閉路を持たない scheduled graph が一対一対応し、かつ、 S_q に対する make-span が scheduled graph のクリテカル・パス長に一致することを示せば、命題1が証明される。

S_q に対し、disjunctive pair $\{(q_i, q_i), (q_j, q_i)\}$ の一方を、 $q_i < q_j$ ならば、 (q_i, q_i) 、 $q_j < q_i$ ならば、 (q_j, q_i) にしたがって選ぶ。 その結果、閉路をもたない scheduled graph が得られる。 なぜならば、もし、閉路

$$\{(q_i, q_{i+1}), (q_{i+1}, q_{i+2}), \dots, (q_{i+j}, q_{i+j+1}), (q_{i+j+1}, q_i)\}$$

があれば、上述の定義より、

$$q_i < q_{i+1} < \dots < q_{i+j+1}, \quad \text{かつ、} \quad q_i > q_{i+j+1}$$

となり、これは S_q の順序と矛盾する。

逆に、 S_q から得られた閉路のない scheduled graph において disjunctive pair $\{(q_i, q_j), (q_j, q_i)\}$ のうち、もし、 (q_i, q_j) が選ばれているならば、 $q_i < q_j$ 、 (q_j, q_i) が選ばれているならば、 $q_j < q_i$ によって N 上に順序をつければ、明らかに元の順序 S_q が得られる。 このように S_q と scheduled graph は一対一対応する。ところで、scheduled graph の各アークは工程の先行関係を表わすから、そのクリテカル・パス長が make-span を表わすことはよく知られている [10]。 (証明終り)

補題1： S_q に対する scheduled graph のクリテカル・パス長を $f(S_q)$ と記すとき、

$$(5) f(S_q) = \max_{k=1 \sim n, j=1 \sim k} (a_{q_j} + \sum_{l=j}^{k-1} b_{q_l} + \bar{c}_{q_k}, d)$$

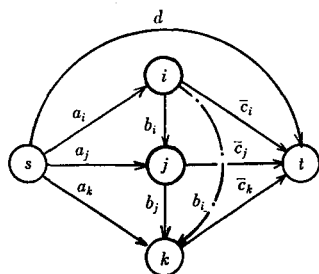


図2 Scheduled graph $S_q = [i, j, k]$

補題1は、図2の scheduled graph で、一点鎖線で示されるようなアークを含む s から t へのパスがクリテカル・パスにならないことを示している。このことはアーク (i, k) の長さの定め方より明らかである。

S_q の集合を、

$$(6) S = \{S_1, S_2, \dots, S_c\}, \quad c = n!$$

と記すとき、問題Pは、

図1は、表1の例に対する disjunctive graph G である。

disjunctive graph G において各 disjunctive pair のどちらか一方を選び、他方を除いて得られるグラフを disjunctive graph に対して scheduled graph と呼ぶ。

命題1： 問題Pは、その結果得られる scheduled graph が閉路を持たず、かつ、そのクリテカル・パスが最短になるように、Gの各 disjunctive pair

$$(7) f^* = f(S^*) = \min_{S_q \in S} f(S_q)$$

与えられる最適順序 S^* と最短クリテカル・パス f^* を求めることに帰着する.

3. 部分問題

3節と4節では分枝限定法に必要な部分問題とその下限を述べるにとどめ, 分枝限定法の一般原理 [9] は省略する.

本文の分枝限定法では, 加工機械上で処理される仕事を最初から順次定めていく方針をとる. その結果, たとえば図4のような探索図を得る. 探索図の節点には, 固定された最初の l 個の仕事の下で残りの仕事の順序を定める部分問題が対応し, 次のように定義される.

加工機械上で処理される最初の $(l-1)$ 個 ($0 \leq l-1 < n$) の仕事を, 処理順序にしたがって並べ, F' と記す. すなわち,

$$(8) F' = [q_1, q_2, \dots, q_{(l-1)}], \quad q_1, q_2, \dots, q_{(l-1)} \in N$$

は, $q_1 < q_2 < \dots < q_{(l-1)}$ および任意の仕事 $p \in F'$ に対して $q_{(l-1)} < p$ を意味する. ただし, $p \in F'$ は F' を集合 $\{q_1, q_2, \dots, q_{(l-1)}\}$ と見なすとき, p は F' の元でないことを示す. また, F' の位数を $|F'|$ と記す.

$$|F'| = (l-1)$$

任意の F' に対し, disjunctive graph $G(F')$ が次のように定義される.

- (a) $G(F')$ は節点の集合 $\{s\} \cup \{N-F'\} \cup \{t\}$ を有する.
- (b) $G(F')$ は次に示すアークを持つ.
 - (i) 長さ $a'_i (\geq 0)$ のアーク (s, i) , $i \in N-F'$.
 - (ii) $N-F'$ に属する任意の節点 i と j に対する disjunctive pair $\{(i, j), (j, i)\}$. 前者の長さは b_i , 後者の長さは b_j である.
 - (iii) 長さ \bar{c}_i のアーク (i, t) , $i \in N-F'$.
 - (iv) 長さ $d' (\geq 0)$ のアーク (s, t) .

任意の F

$$F = [F', q_l] = [q_1, q_2, \dots, q_{(l-1)}, q_l], \quad q_l \in N-F'$$

に対する disjunctive graph $G(F)$ は, $G(F')$ から次のように得られる.

- (i) disjunctive pair $\{(i, q_l), (q_l, i)\}$, $i \in N-F$ を除く.
 - (ii) アーク (s, i) , $i \in N-F$ の長さを
- $$(9) a_i = \max\{a'_i, a_{q_l} + b_{q_l}\}$$

に変換する.

- (iii) アーク (s, q_l) とアーク (q_l, t) を除く.
- (iv) アーク (s, t) の長さを,

$$(10) d = \max\{d', a_{q_l} + \bar{c}_{q_l}\}$$

に変換する.

(v) 節点 q_l を除く.

図3は disjunctive graph $G(F)$ の1例である.

$G(F)$ 上で各 disjunctive pair のどちらか一方の disjunctive arc を選び, 他方を除いて得られるグラフを F に対する scheduled graph と呼ぶ.

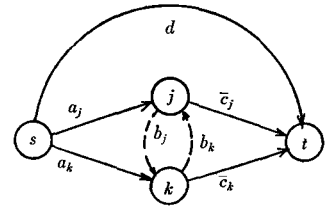


図3 Disjunctive graph $G(F)$

$$\begin{aligned}
 l &= 1, q_1 = i, F' = \phi, F = [i] \\
 a_j &= \max\{a_j, a_i + b_i\} \\
 a_k &= \max\{a_k, a_i + b_i\} \\
 d &= \max\{d, a_i + c_i\}
 \end{aligned}$$

命題2: 任意の F に対し, 残りの仕事に順序をつけ, 全仕事の終了時刻を最小にする問題を部分問題と呼び, $P(F)$ と記す. このとき, $P(F)$ は, F に対する閉路のない scheduled graph のクリテカル・パスを最短にするように, disjunctive graph $G(F)$ 上で disjunctive arc を選ぶ問題 (これを問題 $G(F)$ と略す) に帰着する.

証明: 帰納法で証明する. $F = \phi$ の場合は命題1より明らかである.

$$F' = [q_1, q_2, \dots, q_{(l-1)}], l \geq 2$$

のとき, 問題 $P(F')$ と問題 $G(F')$ が等価であると仮定し,

$$F = [q_1, q_2, \dots, q_{(l-1)}, q_l]$$

に対する問題 $P(F)$ と問題 $G(F)$ が等価であることを示す. そのために, まず, グラフ $G(F')$ から, disjunctive arc (i, q_l) , $i \in N - F$ を除いて得られるグラフを $G'(F)$ とし, 問題 $G'(F)$ (グラフ $G'(F)$ においてその結果得られる scheduled graph のクリテカル・パスが最短になるように disjunctive pair の一方を選ぶ問題) と問題 $P(F)$ が等価であることに注意しよう. 以下, 問題 $G(F)$ と問題 $G'(F)$ が, 残りの仕事の同じ順序づけに対して, 同じクリテカル・パス長を持つという意味で等価であることを示し, 問題 $G(F)$ と問題 $P(F)$ の等価性を示す.

問題 $G'(F)$ を解き, 得られた scheduled graph のクリテカル・パスとして次の場合を考える.

(i) (s, q_l, t) の場合³⁾

(10)式による d の定め方より, $d = a_{q_l} + \bar{c}_{q_l}$. したがって, (s, t) が $G(F)$ のクリテカル・パスとなり, その長さは変化しない.

(ii) $(s, q_l, \alpha, \dots, \beta, t)$, $\alpha, \dots, \beta \in N - F$ の場合

(9)式による a_i の定め方より, $a_\alpha = a_{q_l} + b_{q_l}$.

したがって, $(s, \alpha, \dots, \beta, t)$ が $G(F)$ のクリテカル・パスとなり, その長さは変化しない.

(iii) q_l を経由しない場合

明らかに $G(F)$ も同じクリテカル・パスを与える.

したがって, 問題 $G'(F)$ と問題 $G(F)$ は等価である.

(証明終り)

4. 下限の計算

ここでは部分問題 $P(F)$ の最適値の下限を disjunctive graph $G(F)$ に基づいて定義する.

$G(F)$ において, 節点 s から $N - F$ に属する任意の節点 P を経て t に至るパスの長さは $(a_p + \bar{c}_p)$

3) (s, q_l, t) は節点 s を出発して, 節点 q_l を通り, 節点 t に至るパスを表わす. 以下同じ.

であるから、これらの最大値、

$$(11) \quad Z_1(F) = \max_{p \in N-F} \{a_p + \bar{c}_p\}$$

は問題 $G(F)$ の下限である。同様に、

$$(12) \quad Z_2(F) = \min_{p \in N-F} a_p + \sum_{p \in N-F} b_p + \min_{p \in N-F} c_p$$

も問題 $G(F)$ の下限である。よって、

$$(13) \quad Z(F) = \max \{d, Z_1(F), Z_2(F)\}$$

は、部分問題 $P(F)$ の最適値の下限を与える。

$Z(F)$ は分枝限定法で要求される次の性質を満たしていることは明らかである[7].

$$(14) \quad Z(F') \leq Z(F)$$

$$(15) \quad Z(S_q) = f(S_q)$$

5. 最適順序に関する定理

ここでは分枝限定法を進めていく上で、探索範囲をへらすことに関する定理を示す。

定理 1: グラフ $G(F)$ 上で、節点 $i, j \in N-F$ について、

$$(16) \quad \begin{cases} a_i \leq a_j \\ c_i \geq c_j \end{cases}$$

が成り立つとき、次のことが証明される。

(i) 仕事 j が仕事 i に直接先行する順序以外に $P(F)$ の最適順序が存在する。

(ii) 次の条件式(17)~(20)のいずれかが成立すれば、仕事 i が仕事 j に先行する $P(F)$ の最適順序が存在する。

$$(17) \quad b_i = b_j$$

$$(18) \quad \begin{cases} a_i + b_i \geq a_k \\ b_j + c_j \geq c_k, \forall k \in N-F \end{cases}$$

$$(19) \quad \begin{cases} a_i + b_i \geq a_k, \forall k \in N-F \\ b_i \leq b_j \end{cases}$$

$$(20) \quad \begin{cases} b_i \geq b_j \\ b_j + c_j \geq c_k, \forall k \in N-F \end{cases}$$

証明: 仕事 j が仕事 i に先行する任意の順序を、 $S_q = [\dots, \alpha, \dots, j, \dots, \beta, \dots, i, \dots, \gamma, \dots], \dots, \alpha, \dots, \beta, \dots, \gamma, \dots \in N-F$, とする。ここで α は j に先行する任意の仕事、 β は j に後続し、 i に先行する任意の仕事、また、 γ は i に後続する任意の仕事を表わす。 S_q に対して i と j を逆にした順序を、 $S_p = [\dots, \alpha, \dots, i, \dots, \beta, \dots, j, \dots, \gamma, \dots]$ で表わす。このとき、

$$(21) \quad f(S_p) \leq f(S_q)$$

を証明すればよい。ただし、 $f(S_p)$ は $P(F)$ において S_p という順序を用いたときの全仕事の終了時刻を表わす。命題 2 によって $f(S_p)$ は、 S_p に対する $G(F)$ の scheduled graph のクリテカル・

表 2 定理 1 の証明

$S_p = \alpha, \dots, i, \dots, \beta, \dots, j, \dots, \gamma$		$S_q = \alpha, \dots, j, \dots, \beta, \dots, i, \dots, \gamma$	
パス	パス長	パス	パス長
1 s, α, \dots, i, t	$a_\alpha + b_\alpha + \dots + b_i + c_i$	$s, \alpha, \dots, j, \dots, \beta, \dots, i, t$	$a_\alpha + b_\alpha + \dots + b_j + \dots + b_\beta + \dots + b_i + c_i$
2 $s, \alpha, \dots, i, \dots, \beta, t$	$a_\alpha + b_\alpha + \dots + b_i + \dots + b_\beta + c_i$		
3 $s, \alpha, \dots, i, \dots, \beta, \dots, j, t$	$a_\alpha + b_\alpha + \dots + b_i + \dots + b_\beta + \dots + b_j + c_j$	$s, \alpha, \dots, j, \dots, \beta, \dots, i, t$	$a_\alpha + b_\alpha + \dots + b_j + \dots + b_\beta + \dots + b_i + c_i$
4 $s, \alpha, \dots, i, \dots, \beta, \dots, j, \dots, \gamma, t$	$a_\alpha + b_\alpha + \dots + b_i + \dots + b_\beta + \dots + b_j + \dots + b_\gamma + c_\gamma$	$s, \alpha, \dots, j, \dots, \beta, \dots, i, \dots, \gamma, t$	$a_\alpha + b_\alpha + \dots + b_j + \dots + b_\beta + \dots + b_i + \dots + b_\gamma + c_\gamma$
5 s, i, \dots, β, t	$a_i + b_i + \dots + b_\beta + c_\beta$		
6 $s, i, \dots, \beta, \dots, j, t$	$a_i + b_i + \dots + b_\beta + \dots + b_j + c_j$	$s, j, \dots, \beta, \dots, i, t$	$a_j + b_j + \dots + b_\beta + \dots + b_i + c_i$
7 $s, i, \dots, \beta, \dots, j, \dots, \gamma, t$	$a_i + b_i + \dots + b_\beta + \dots + b_j + \dots + b_\gamma + c_\gamma$	$s, j, \dots, \beta, \dots, i, \dots, \gamma, t$	$a_j + b_j + \dots + b_\beta + \dots + b_i + \dots + b_\gamma + c_\gamma$
8 s, β, \dots, j, t	$a_\beta + b_\beta + \dots + b_j + c_j$		
9 $s, \beta, \dots, j, \dots, \gamma, t$	$a_\beta + b_\beta + \dots + b_j + \dots + b_\gamma + c_\gamma$		
10 s, j, \dots, γ, t	$a_j + b_j + \dots + b_\gamma + c_\gamma$	$s, j, \dots, \beta, \dots, i, \dots, \gamma, t$	$a_j + b_j + \dots + b_\beta + \dots + b_i + \dots + b_\gamma + c_\gamma$

パス長である。よって S_p に対する scheduled graph でクリテカル・パスになりうるすべてのパスのそれぞれ (u と記す) に対して, S_q に対する scheduled graph で s から t に至るパスのうち, 少なくとも一つ u より短くないもの (それは S_q のクリテカル・パスより長くない) があれば, (9) 式が証明できる。このとき, 明らかに節点 i と j のいずれも通らないパスは考慮する必要がないから, 節点 i と j (あるいは, その一方) を通るパスのみを考える。また $G(F)$ に補題 1 を適用すると, S_p に対する $G(F)$ の scheduled graph の考慮すべきパスのすべては表 2 の S_p 欄のパス 1~10 で尽くされる。ここで, たとえばパス 2 は s から, S_p において i に先行するいくつかの仕事 α , 仕事 i および S_p において i と j の間に位置するいくつかの仕事 β を経由し, t に至るパスを示す。またパス長 ($a_\alpha + b_\alpha + \dots + b_i + \dots + b_\beta + c_\beta$) は ($a_\alpha + \sum_{k \leq \beta} b_k + c_\beta$) を表わす。まず, (10) 式が成り立つとき, S_p のパス 1, 3, 4, 6, 7, および 10 に対しては, それより短くない S_q のパスが存在し, それぞれ S_q 欄に示されている。

(i) の証明: 仕事 j が仕事 i に直接先行することは S_q において j と i の間に位置する β が存在しないことを意味する。同様にして S_p において i と j の間に仕事はない。よってパス 2 はパス 1 に等しく, パス 5 と 8 はそれぞれ同じパスが S_q にもあり, またパス 9 はパス 10 に等しい。した

表3 定理1の証明

S_p のパス	S_p のパス
2	$s, \alpha, \dots, j, \dots, \beta, t$
5	s, j, \dots, β, t
8	s, β, \dots, i, t
9	$s, \beta, \dots, i, \dots, \gamma, t$

がって、パス1~10すべてに対して、それより短くないパスが S_q に対する scheduled graph に存在し、よって(2)式が証明された。

(17)式についての証明：(17)式が成立すれば、表3に示すように、 S_p のパス2, 5, 8, 9に対して、それらより短くない S_q のパスが存在するので、表2と併せ用いる

と、(2)式が証明される。

(18)式についての証明： S_p のパス2は、(18)式よりパス3より長くないので、 S_p に対応する scheduled graph のクリテカル・パスになりえない。よってパス2を考慮する必要はない。パス5, 8, 9についても同様である。よって(2)式が証明される。

(19)式についての証明： $b_i \leq b_j$ から S_p のパス2は S_q のパス (s, j, \dots, β, t) より長くない。また、パス5は S_q のパス (s, j, \dots, β, t) より長くない。さらに $a_i + b_i \geq a_k, \forall k \in N-F$ よりパス8, 9はクリテカル・パスにならない。

(20)式についての証明： $N-F$ に属する任意の k に対して、 $b_j + c_j \geq c_k$ が成り立つから、パス2, 5はクリテカル・パスにならない。 $b_i \geq b_j$ よりパス8, 9はそれぞれ、 S_q のパス (s, β, \dots, i, t) 、 $(s, \beta, \dots, i, \dots, \gamma, t)$ より長くない。 (証明終り)

定理2： グラフ $G(F)$ 上で、任意の節点 $k, l \in N-F$ について、

$$(2) \quad \begin{cases} a_k + b_k \geq a_l \\ b_k + c_k \geq c_l \end{cases} \quad \forall k, \forall l \in N-F$$

が成り立つとする。このとき、以下で求められる仕事 i が $(|F|+1)$ 番目に、仕事 j が n 番目になり、 $\{N-F\}$ の残りの仕事の順序が任意であるような $P(F)$ の最適順序が存在する： $a_p = \min_{k \in N-F} a_k$, $c_q = \min_{k \in N-F} c_k$ とする。

(i) $p \neq q$ ならば、 $i=p, j=q$ である。

(ii) $p=q$ ならば、 $a_r = \min_{k \in N-F - \{p\}} a_k$, $c_s = \min_{k \in N-F - \{p\}} c_k$ とする。 $a_p + c_s \leq a_r + c_p$ ならば、 $i=r, j=s$ 。そうでなければ、 $i=r, j=p$ 。

証明：(2)式より順序 $[\alpha, \dots, \beta, \dots, \gamma]$ に対する $G(F)$ の scheduled graph のクリテカル・パス長は $(a_\alpha + \sum_{\beta \in N-F} b_\beta + c_\gamma)$ である。ここで $\sum_{\beta \in N-F} b_\beta$ は順序に関係なく一定であるから、最短クリテカル・パスは、

$$\min_{\alpha, \gamma \in N-F} (a_\alpha + c_\gamma) + \sum_{k \in N-F} b_k$$

で与えられる。明らかに、定理2で定義された仕事 i と j は $\min_{\alpha, \gamma \in N-F} (a_\alpha + c_\gamma)$ をとる。

(証明終り)

定理3： グラフ $G(F)$ 上ですべての節点 k, l について、

$$(23) \quad a_k + b_k \geq a_i, \quad \forall k, \quad \forall i \in N - F$$

が成立するとき、節点 $i, j \in N - F$ について、

$$(24) \quad \begin{cases} c_j + b_j \geq c_k, & \forall k \in N - F - \{i\} \\ c_i \geq c_j \end{cases}$$

が満たされるならば、仕事 i が仕事 j に先行するような $P(F)$ の最適順序が存在する。

証明：(23)式によって表 2 で S_b のクリテカル・パスになりうるのはパス 1～4 だけである。さらに $c_j + b_j \geq c_k$ より、 S_b のクリテカル・パスは 3 と 4 に限られる。パス 4 と同じ長さの S_q のパスは無条件に存在し、 $c_i \geq c_j$ からパス 3 は S_q のパス $(s, \alpha, \dots, j, \dots, \beta, \dots, i, t)$ より長くない。

(証明終了)

定理 4： グラフ $G(F)$ 上で節点 $j \in N - F$ が、

$$(25) \quad a_j \geq \min_{k \in N - F - \{j\}} (a_k + b_k)$$

を満たすならば、仕事 j が残りの全仕事 $N - F - \{j\}$ に先行する順序以外に最適順序が存在する。

$$\text{証明：} \quad a_i + b_i = \min_{k \in N - F - \{j\}} (a_k + b_k)$$

を満たす仕事を i とする。仕事 j が $\{N - F\}$ の残りの全仕事に先行する任意の順序 $[j, \dots, \alpha, \dots, i, \dots, \beta]$ に対して、順序 $[i, j, \dots, \alpha, \dots, \beta]$ を考えると、仕事 i の結果、仕事 j の加工開始は遅れないから、後者の順序による scheduled graph のクリテカル・パスは前者のそれよりも長くない。

(証明終了)

定理 5： グラフ $G(F)$ 上で節点 $i \in N - F$ が、

$$(26) \quad a_i + b_i \geq \max_{k \in N - F} a_k$$

を満たすならば、部分問題 $P([F, i])$ において、 $N - F - \{i\}$ に属する仕事 k をアーク (k, i) の長さの非増加の順に並べた最適順序が存在する。

証明：disjunctive graph $G([F, i])$ を $G(F)$ から作ると、任意の節点 $k \in N - F - \{i\}$ に対するアーク (s, k) の長さは、(9)式に(26)式を適用して、

$$(27) \quad \max \{a_k, a_i + b_i\} = a_i + b_i.$$

(27)式はすべての $k \in N - F - \{i\}$ に対して一定であるから、定理 1 の (i) において、すべての a_i が同じ値になることに相当し、 c_j の値によって順序が定まり、その結果定理 5 を証明することができる。

(証明終了)

6. アルゴリズム

ここで用いられる分枝限定法による探索は、図 4 のような探索図 (tree) で表わすことができる。図において節点 0 は問題 P に、その他の節点は部分問題 (3 節参照) に対応する。

各節点を区別するために、生成順序にしたがって節点番号をつける。すなわち、節点 π は π 番目に生成された部分問題である。節点 π から逆に巡って節点 0 に至るとき通るパス (太線) は一意的に決まり、そのパスのアークの数 l を節点 π のレベル $L(\pi)$ という。このとき通る節点を π

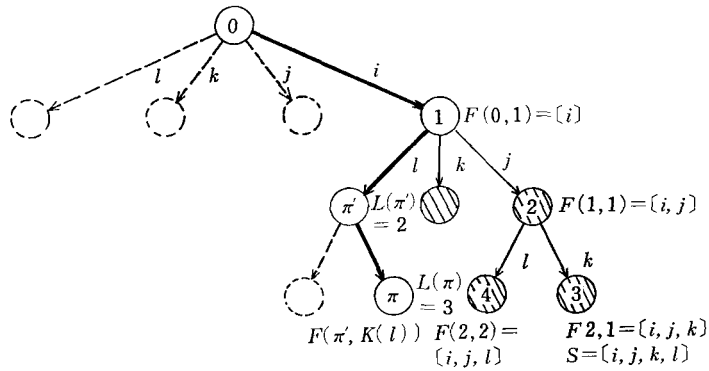


図4 探索図
 $N = \{i, j, k, l\}$

の祖先という。また、 π の祖先で π に直接結ばれている節点 π' を π の親 $V(\pi)$ (あるいは π は $V(\pi)$ の子) という。

節点 π' のレベル $L(\pi') = l - 1$ ，対応する部分問題を $P(F')$ ， $F' = [q_1, q_2, \dots, q_{l-1}]$ とするとき， π' は $|N - F'|$ 個の子に分枝される。 π' の子 π に対応する部分問題は $P(F)$ ， $F = [F', q_l]$ ， $q_l \in N - F'$ と書くことができる。

π' の子はグラフ $G(F')$ のアーク (s, i) ， $i \in N - F'$ の長さ a_i' の非減少の順序で生成される。すなわち， $u, v \in N - F'$ に対して $a_u' < a_v'$ ならば，部分問題 $P([F', u])$ は部分問題 $P([F', v])$ よりも早く生成される。この順序にしたがった π' の子 π が π' の何番目の子であることを示すために，その順番を $K(L(\pi'))$ と記すとき， π で固定されている仕事の順序は一意的に決まり，それを $F = F(\pi', k)$ ， $k = K(L(\pi'))$ と記す。また，節点 π で新しく固定された仕事 q_l は $F_l(\pi', k)$ と書くことができる。

なお，本アルゴリズムで各部分問題は，実際にその問題を考察するとき初めて生成するという手順をとる。また探索法としていわゆる線型探索法 (Linear search) [5] を用いる。すなわち，結論の出た節点 (その最適解が得られている，現在の解 (= 暫定解) よりよい値を与える可能性がない，など) を終端節点と呼ぶ。たとえば，図4において π を探索している時点で， π の祖先の右側の黒丸の節点は終端節点と見なされる。また，左側の破線の節点は， π の時点でまだ生成されていない節点である。

もし， π が終端節点であるならば，線型探索にしたがって次に生成される節点は，まだ生成されていない子を持つ π の祖先のうち最も π に近い節点を親とし，祖先を結ぶパスに最も近い節点である。もし，この過程で上に相当する節点がなければ，問題 P は解けたことになり，そのときの暫定解が最適解である。

もし， π についてなら結論が得られなければ， π の最初の子が生成され，それに対する部分問題を考察することになる。

結局，各時点における探索図が状態を示すには，考察中の節点 π とその祖先が，各レベルで何番目に生成されたかを示せばよい。すなわち， $K(0), \dots, K(L(\pi))$ を記憶し，かつ，新しい節点が生産されるごとに，これらを正確な値に更新する。 $K(0)$ が n 以上になれば， P のすべての可

能な順序を尽くしたことになる。

次に上述の記号に基づいてアルゴリズムの詳細を示す。

初期値の認定

手順 1: レベル $l := 0$, 節点数 $m := 0$, 節点番号 $\pi := 0$, $K(0) := 0$, $F := F(0, 0) := \phi$, 暫定解の値 $f := \infty$, $K(1) := 0$ とする。 $Z(\phi)$ を計算して手順 2 へ行く。

部分問題のテスト

手順 2: $K(l) = |N - F|$ ならば, 手順 19 へ行く。 そうでなければ, 手順 3 へ行く。

定理 1 および定理 4 のテスト

手順 3: $K(l) := K(l) + 1$, $q_{l+1} := F_{l+1}(\pi, K(l))$ とする。 $l \neq 0$, $a_{q_{l+1}} \leq a_{q_l}$ かつ, $c_{q_{l+1}} \geq c_{q_l}$ ならば, 手順 2 へもどる (定理 1 (i)). そうでなければ, $k := 0$ として手順 4 へ進む。

手順 4: $k := k + 1$ とする。 $k = K(l)$ ならば, 手順 8 へ行く。 そうでなければ, 手順 5 へ進む。

手順 5: $l := F_{l+1}(\pi, k)$ とする。 $a_{q_{l+1}} \geq a_i + b_i$ ならば, 手順 2 へもどる (定理 4). そうでなければ, 手順 6 へ進む。

手順 6: $c_i < c_{q_{l+1}}$ ならば, 手順 4 へもどる (定理 1 適用不能). そうでなければ, 手順 7 へ進む。

手順 7: 次の条件,

$$(i) \quad b_i = b_{q_{l+1}}$$

$$(ii) \quad a_i + b_i \geq \max_{p \in N - F} a_p \text{ で } b_i < b_{q_{l+1}}$$

$$(iii) \quad b_{q_{l+1}} + c_{q_{l+1}} \geq \max_{p \in N - F} c_p \text{ で } b_i < b_{q_{l+1}}$$

$$(iv) \quad a_i + b_i \geq \max_{p \in N - F} a_p \text{ で } b_{q_{l+1}} + c_{q_{l+1}} \geq \max_{p \in N - F} c_p$$

のいずれかが成立すれば, 手順 2 へもどる (定理 1 (ii)). そうでなければ, 手順 4 へもどる。

定理 3 のテスト

手順 8: $\min_{p \in N - F} (a_p + b_p) < \max_{p \in N - F} a_p$ ならば, 手順 11 へ行く (定理 3 適用不能). そうでなければ, $h := 0$ として手順 9 へ進む。

手順 9: $h := h + 1$ とする。 $h > |N - F|$ ならば, 手順 11 へ行く (定理 3 適用不能). そうでなければ, 手順 10 へ進む。

手順 10: $i := F_{l+1}(\pi, h)$ とする。 $i \neq q_{l+1}$, $b_{q_{l+1}} + c_{q_{l+1}} \geq \max_{p \in N - F - \{i\}} c_p$ かつ, $c_{q_i} \geq c_{q_{l+1}}$ ならば, 手順 2 へもどる (定理 3). そうでなければ, 手順 9 へもどる。

定理 5 のテスト

手順 11: $a_{q_{l+1}} + b_{q_{l+1}} < \max_{p \in N - F} a_p$ ならば, 手順 12 へ進む (定理 5 適用不能). そうでなければ, $S_q := [F, q_{l+1}, q_{l+2}, \dots, q_n]$ とし, $f(S_q)$ を計算する。 ただし, $[q_{l+2}, \dots, q_n]$ は $N - F - \{q_{l+1}\}$ の仕事を c_i の非減少順に並べたものである (定理 5). 手順 17 へ行く。

部分問題の生成と下限の計算

手順 12 : $F := F(\pi, K(l))$, $m := m+1$, $V(m) := \pi$, $\pi := m$, $l := l+1$ として $G(F)$ を生成し, $Z(F)$ を計算する. $l=n-1$ ならば, $S_q := [F, q_n]$, $q_n = N-F$, そして $f(S_q) := Z(F)$ として手順 17 へ行く. そうでなければ, 手順 13 へ進む.

下限値テスト

手順 13 : $f \leq Z(F)$ ならば, 手順 19 へ行く. そうでなければ, 手順 14 へ進む.

定理 2 のテスト

手順 14 : $\min_{p \in N-F} (a_p + b_p) \geq \max_{p \in N-F} a_p$ かつ, $\min_{p \in N-F} (b_p + c_p) \geq \max_{p \in N-F} c_p$ ならば, 手順 15 へ進む. そうでなければ, $K(l) := 0$ として手順 2 へもどる.

手順 15 : $a_p := \min_{r \in N-F} a_r$, $c_q := \min_{r \in N-F} c_r$ とする. $p \neq q$ ならば, $i := p$, $j := q$ として手順 16 へ進む. そうでなければ, $a_r := \min_{v \in N-F - \{p\}} a_v$, $c_s := \min_{v \in N-F - \{q\}} c_v$ とする. $a_p + c_s \leq a_r + c_p$ ならば, $i := p$, $j := s$ とし, $a_p + c_s > a_r + c_p$ ならば, $i := r$, $j := p$ として手順 16 へ進む.

手順 16 : $S_q := [F, i, q_{l+2}, \dots, q_{n-1}, j]$ とし, $f(S_q)$ を計算する. ただし, $[q_{l+2}, \dots, q_{n-1}]$ は $N-F - \{i, j\}$ の任意の順序. $K(l) := |N-F|$ とし, 手順 17 へ進む (定理 2).

暫定解の更新

手順 17 : $f > f(S_q)$ ならば, $\bar{S} := S_q$, $f^* := f(S_q)$ とする. $l \neq 0$ ならば, 手順 19 へ行く. $l=0$ ならば, 手順 18 へ進む.

計算終了

手順 18 : $S^* := \bar{S}$, $f^* := f^*$ として計算終了.

バック・トラック

手順 19 : 節点 $V(\pi)$ の下限値が f より小さくないならば, $l=0$ のとき, 手順 18 へ, $l>0$ のとき, $l := l-1$, $\pi := V(\pi)$ として手順 19 へもどる. $V(\pi)$ の下限値が f より小さいならば, $\pi := V(\pi)$, $l := l-1$, $F := F(V(\pi), K(l))$ として手順 2 へもどる.

7. 数値実験

ここでは, 準備時間 a_i , 加工時間 b_i および整理時間 c_i の大小関係と仕事の数 n を変えた問題群についてアルゴリズムの能率を調べた結果を示す.

表 4 にその結果の一部を示す. 表の第 1 列は問題の型を表わす. すなわち, 問題 I では, $a_i = 1 \sim 100$, $b_i = 1 \sim 1,000$, $c_i = 1 \sim 100$ の範囲の一樣乱数を与えて, b_i が a_i , c_i より平均的に大きな値をとるようにした. 問題 II では a_i, b_i, c_i に同程度の大きさの一樣乱数を与え, 問題 III では b_i が a_i, c_i よりも平均的に小さくなるように, それぞれに一樣乱数を与えた. また, 表の第 2 列に示すように, 各型の問題に対して仕事の数 n を $10 \sim 50$ の範囲で変えた. さらに, 各問題の型と n に対して a_i, b_i, c_i の数値を変えた問題を解いた. 表の第 3 列以後の結果は, *印⁴⁾を除いて, す

4) *印は最適解の得られた問題についての結果を表わす.

表4 数値実験の結果

問題の型	仕事の数 n	部分問題 の数	計算時間 (Sec)	棄却された順序の割合 $R\%$				
				定理 1	定理 2	定理 3	定理 4	定理 5
I $a_i=1\sim$ 100 $b_i=1\sim$ 1000 $c_i=1\sim$ 100	10	1.4	0.134	1.33	42.0	6.22	30.0	2.45
	20	1.0	0.249	32.0	25.0	1.00	1.00	8.00
	30	1.0	0.383	17.3	0	0.66	0	4.66
	40	1.2	0.523	0	20.0	0	0	1.50
	50	1.2	0.850	0	0	0	0	1.60
II $a_i=1\sim$ 1000 $b_i=1\sim$ 1000 $c_i=1\sim$ 1000	10	1.8	0.139	0	0	2.78×10^{-2}	0	4.47
	20	2.6	0.274	3.00	0	1.05×10^{-1}	10.0	2.44
	30	1.8	0.367	0	0	0	0	1.38
	40	2.6	0.564	0	0	0	0	2.64×10^{-2}
	50	2.6	0.849	0	0	0	0	2.46×10^{-2}
III $a_i=100\sim$ 1000 $b_i=1\sim$ 100 $c_i=100\sim$ 1000	10	16.2	0.192	0	0	0	20.0	0
	20	33.4	0.363	5.28×10^{-14}	0	0	2.92×10^{-11}	2.66×10^{-14}
	*30	1600	11.4	4.00×10^{-21}	0	0	7.60×10^{-21}	3.70×10^{-23}
	*40	17	0.605	9.54×10^{-24}	0	0	0	6.34×10^{-24}

べてそれらの平均値である。なお、計算時間が1分を過ぎると、無条件に打ち切るようにした。

表の第3列は、アルゴリズムが終了するまでに生成された部分問題の数の平均値を、第4列は平均計算時間を表わす（使用した計算機は京都大学大型計算機センターのFACOM 230-60、コードはFORTRANである。コンパイルする時間は計算時間に含まれていない）。表の第5列は、5節の各定理によって棄却された仕事の順序の個数の割合の平均値である。すなわち、それを R と記すと、

(8) $R = (\text{棄却された順序の個数}/n!) \times 100(\%)$ で定義される。もし、レベル $l (=0, 1, \dots, n-1)$ の部分問題が棄却されたとすると、 $(n-l)!$ ⁵⁾通りの順序が探索の対象から除かれる。(8)式の分子は、探索の各時点で適用されたそれらの合計である。なお、 R の全定理についての合計を100(%)からさし引いた残りは、下限の性質によって棄却された順序の割合である。なぜならば、実際に生成した順序の個数は $n!$ にくらべて無視できる小ささである。

表4の結果から見て、アルゴリズムは問題I, IIに対してきわめて効果的である。まず、問題Iに対しては各定理が有効に働いていることがわかる。このことは各定理の成立条件から当然であり、とくに、 $\max_i a_i \leq \min_i b_i$ で $\max_i c_i \leq \min_i b_i$ の極端な場合、定理2から問題の最適解は直接的に得られる。なお、 $a_i=1\sim 100$, $b_i=1\sim 1000$, $c_i=1\sim 1000$ の場合も問題Iと同様の結果が得

5) 厳密には、 $(n-l)! - 1$ 。

られた。

問題Ⅱについては、問題Ⅰにくらべて下限の効果がきわめて高い。このことは、探索レベルの小さい段階で定理5が成り立ち、そしてこれによって順序に対応する目的値が問題Pの下限 $Z(\phi)$ に一致することによる。このような結果は、 $a_i=100\sim 1000$, $b_i=100\sim 1000$, $c_i=1\sim 100$ の場合についても同様な結果が得られた。

問題Ⅲでは、前2者の場合と明らかに異なる。すなわち、計算時間が1分以内で5問題のすべてが解けたのは、 $n=20$ までで、 $n=30$, $n=40$ の場合は1問題だけ解け、 $n=50$ では5問題とも保証された最適解を見いだすことができなかった。

表5は1分で計算を打ち切られた残りの問題の探索状態を示す⁶⁾。表の第7列の最適値への近似度は、それを D と記すと、

$$\textcircled{29} D = \{(\bar{f} - Z(\phi)) / Z(\phi)\} \times 100 \quad (\%)$$

で定義されている。ここで、 \bar{f} は計算時間1分で打ち切られたときの暫定解の値、 $Z(\phi)$ は問題Pの下限値である。厳密には D は \bar{f} の f^* からのへだたりの上限である。 D の結果から判断して、 \bar{f} は、最適解の保証はないけれども、最適解にかなり近いといえる。最適解に近い暫定解の得られる一つの理由は、各定理による R の効果である。なぜなら、 R の数値は非常に小さいにもかかわらず、これらによって除かれる順序の個数は、実際に生成された順序の個数にくらべると天文学的に大である。したがって、実質的にはかなりの広い範囲を探索していると考えられるからである。なお、 $a_i=100\sim 1000$, $b_i=1\sim 100$, $c_i=1\sim 100$ についても、問題Ⅲと同様な結果が得られた。

表5 計算時間1分での探索状態 問題Ⅲ

仕事の数 n	部分問題の数	棄却された順序の割合 R %				最適値への近 似度 D %
		定理 1	定理 3	定理 4	定理 5	
30	3882	2.64×10^{-10}	6.28×10^{-22}	2.82×10^{-11}	1.46×10^{-18}	6.8
40	1872	5.36×10^{-20}	3.54×10^{-26}	3.32×10^{-20}	8.78×10^{-22}	3.9
50	1678	8.64×10^{-27}	0	1.18×10^{-27}	3.38×10^{-17}	5.1

以上、数値実験の結果を要約すると、

- (i) 問題Ⅰ, Ⅱについては、大きな n に対してもほぼ直接的に最適解が得られる。
- (ii) 問題Ⅲについて、大きい n に対して最適解を短い時間で求めることは困難であるが、最適解に近い近似解が得られる。
- (iii) 問題の難易度は、 a_i と b_i の大小関係によって変わり、 c_i の影響をあまり受けない。

8. 結 び

本論文は、ジョブ・ショップ・スケジューリングの特殊な型の問題を分枝限定法で解くことについて論議した。本来、この種の問題は分枝限定法で解かざるをえない局面が多いが、定理1～

6) 明らかに、定理2は問題Ⅲに対して有効でないので、アルゴリズムから除いた。

5の条件を満たす部分問題についてはアルゴリズムの能率化がはかられる。したがって、数値的に示したように、本アルゴリズムが有効に作用する問題が数多くあることが確かめられた。

ここで取り上げたモデルは、現実の実例に基づいて得られたもので、特殊なモデルという面もないではないが、単純なモデルだけに、適用できる事例が広く存在すると期待している。今後、さらに他の目的関数 (Mean flow time, Tardiness など) [2] についても考察し、問題の適用範囲を広げたい。

最後に、筆者の一人 (木瀬) は、日頃、ご指導いただいている京都工芸繊維大学の宇野 稔教授に、この場をかりて深謝の意を表したい。

なお、本研究の一部は文部省科学研究費によるものである。

参 考 文 献

- [1] Balas, E., "Machine Sequencing via Disjunctive Graphs: An Implicit Enumeration Algorithm," *Operations Research*, **17** (1969), 941-957.
- [2] Conway, R., W. Maxwell and L. Miller, "*Theory of Scheduling*," Addison-Wesley, 1967.
- [3] Florian, M., P. Trepant and G. McMahon, "An Implicit Enumeration Algorithm for the Machine Sequencing Problem," *Management Sci.*, **17** (1971), B782-792.
- [4] Held, M. and R. M. Karp, "The Travelling Salesman Problem and Minimum Spanning Trees: Part II," *Math. Progr.*, **1** (1971), 6-25.
- [5] 茨木俊秀, "整数計画法Ⅲ", オペレーションズ・リサーチ, **11** (1970), 51-64.
- [6] Johnson, S. M., "Optimal Two-and-Three- Stage Production Schedules with Setup Times Included," *Nav. Res. Log. Quart.*, **1** (1971).
- [7] Lawler, E. L. and D. E. Wood, "Branch-and-Bound: A Survey," *Operations Research*, **14** (1966), 699-719.
- [8] 三根 久, 茨木俊秀, 木瀬 洋, "1台の機械に対するある種の順序付問題について", OR学会春季研究発表会 (1973).
- [9] Mitten, L. G., "Branch-and-Bound Method, General Formulation and Properties," *Operations Research*, **18**, 1 (1966), 699-719.
- [10] 関根智明, "PERT-CPM入門", 日科技連, 1965.
- [11] 宇野 稔, 木瀬 洋, "分枝限定法によるジョブ・ショップ・スケジューリング", OR学会春季研究発表会 (1972).