2 工程フローショップ・モデルにおける利益最大化スケジューリング*

栢木紀哉^{*1}, 柳川佳也^{*2}, 宮崎茂次^{*2}

Maximize Profit in Two-Stage Flow Shop Scheduling

Noriya KAYAKI, Yoshinari YANAGAWA and Shigeji MIYAZAKI^{*3}^{*3} Department of Systems Engineering, Faculty of Engineering, Okayama University,
3-1-1 Tsushima-naka, Okayama-city, Okayama, 700-8530 Japan

This paper presents a scheduling method to maximize total profit as a performance measure in a two-stage flow shop model. The profit is defined as the difference between the revenue and the cost of production, and a scheduling method to maximize total profit is proposed using the Branch and Bound method aiming at maximizing revenue and minimizing costs. This paper compares the optimal schedule which maximize total profit with the optimal schedule which minimize total flow time searched by Johnson's theory. This paper also shows the usefulness and some characteristics of the proposed method through various numerical experiments.

Key Words: Production Planning, Production Management, Process Planning, Flow Shop Scheduling, Maximize Total Profit, Branch and Bound Method

1. 緒 言

生産工程では, これまで製品の多様化, 納期短縮化に対応するため, さまざまな自動工作機械が導入されてきた。また, 生産工程におけるスケジューリングの研究として, ジョブショップ・スケジューリングにおいて, 納期遅れペナルティコスト最小化を評価尺度とし, 木探索を用いた近似解法を提案した研究⁽¹⁾や組立作業と段取作業を伴うフローショップ・モデルにおいて, 分岐限界法と擬似動的計画法を利用した, ジョブ完成時刻の平均を最小化するスケジューリングの研究⁽²⁾, シミュレーテッド・アニーリング(SA)法の冷却スケジュールを改善し, m 機械 n ジョブのフローショップ・モデルについて, 最大滞留時間最小化スケジューリング法を提案した研究⁽³⁾など, 稼働率に関する時間を評価尺度とした研究が数多く行われてきた。生産現場において, 生産のサイクルタイムをより小さくし, 生産効率を高めることは利益を増大するために重

要である。

しかし, 現在のように, 顧客の要求が多様化した状況で利益を最大化するためには, ジョブの投入順序, 段取時間, 加工時間, 材料費, 製造コスト, 在庫コストなどのさまざまな要素を考慮した総合的な評価が必要となってくる。また, 昨今のエレクトロニクス製造サービス(EMS)⁽⁴⁾の発展をかんがみした場合, 受注に応じていかに効率的にジョブを加工するかという従来の方法ではなく, 何をどのようにして受注し, 生産するかということを考える必要性が生じている。つまり, ジョブの生産コストを考えるだけでなく, ジョブ1個当たりの利益を考慮し, 時間の経過によるジョブ価値の変化をも考慮した利益の分析が必要となってきた⁽⁵⁾。そのため, 従来の時間最小化スケジュールが, 生産現場の求める利益の最大化につながるとは, 明確にはいえない状況となっている。

本論文では, 段取作業工程を含む2工程フローショップ・モデルに対して, 生産の利益部分をモデル化し, 利益の最大化を評価尺度として提案したスケジューリングの最適解法を提案する。受注生産での加工作業を前提とし, 時間の経過によるジョブの価値変化を考慮した生産の利益最大化を考える。最適解法として分岐限界法を取上げ, アルゴリズムをC言語でプログラム

* 原稿受付 2001年6月27日。

^{*1} 宮城大学事業構想学部(☎981-3298 宮城県黒川郡大和町学苑1)。^{*2} 正員, 岡山大学工学部(☎700-8530 岡山市津島中3-1-1)。
E-mail: miyazaki@et6500.mech.okayama-u.ac.jp

化して数値実験を用い、提案する方法について有用性を検討する。

2. スケジューリング・モデル

本論文では、図1に示すような2工程フローショップ・モデルを用いて解析を行う。このモデルでは、部品入力ステーションから各ジョブが投入され、段取作業と機械作業の2工程を終えて出荷される。第1作業工程である段取作業の段取完了時刻に応じて、第2作業工程である機械作業部分で、i) 工程に遊休時間が発生する場合、ii) ジョブの加工待ち時間が発生する場合、iii) 2つの工程で連続して作業が行える場合の3とおりの状態が起こりうる。そこで、本論文で用いるモデルに対し、それぞれの状態について考慮しながらスケジューリングを行う。

2.1 前提条件 本論文で取扱うモデルでの前提条件を以下に示す。

- (1) n 個のジョブ $J_i (i=1, 2, \dots, n)$ は、二つの工程で投入順に処理され出荷される。
- (2) 各ジョブ到着時に各購入価格が発生し、出荷時に各売却価格が発生する。
- (3) 製造コストと在庫コストは、加工計画完了時刻に清算される。
- (4) 各ジョブの納期については、考慮しない。
- (5) 各ジョブの価格およびコストは、加工計画完了時刻での将来価値に換算して考える。
- (6) 作業工程数は2工程あるものとし、第 j 番めの工程を $M_j (j=1, 2)$ として表す。
- (7) 第1作業工程を段取作業工程とし、第2作業工程を機械作業工程とする。
- (8) 加工計画開始時刻には、すべてのジョブが到着しており、加工開始時刻まで在庫され、加工が終了し次第出荷される。
- (9) 工程 M_2 で $J_{[i-1]}$ の加工作業完了後、 $J_{[i]}$ が前工程で作業中である場合に発生する遊休時間 $W_{[i],2}$ は、 $J_{[i]}$ の M_2 における滞留時間に含めて考える。

2.2 記号 本論文で用いる記号を以下に示す。

n : ジョブ数

J_i : ジョブ番号 $i (i=1, 2, \dots, n)$

M_j : 第 j 番めの作業工程 ($j=1, 2$)

Fig. 1 Two-stage flowshop model with set-up process

p_i : J_i の購入価格

s_i : J_i の売却価格

$t_{i,j}$: J_i の M_j での作業時間

mc : 機械作業工程における単位時間当たりの製造コスト

c_i : J_i の単位時間当たりの在庫コスト

$J_{[i]}$: 時間軸前方から第 i 番めに順序付けされたジョブ ($i=1, 2, \dots, n$)

$J_{[i],j}$: M_j において時間軸前方から第 i 番めに順序付けされたジョブ ($i=1, 2, \dots, n$)

$P_{[i]}$: 順序付けされたジョブ $J_{[i]}$ の購入価格

$S_{[i]}$: 順序付けされたジョブ $J_{[i]}$ の売却価格

$T_{[i],j}$: 順序付けされたジョブ $J_{[i]}$ の M_j での作業時間

$C_{[i]}$: 順序付けされたジョブ $J_{[i]}$ の単位時間当たりの在庫コスト

$Q_{[i],2}$: M_2 で作業を開始するまでの $J_{[i]}$ の待ち時間

$W_{[i],2}$: $J_{[i]}$ の作業を開始するまでの M_2 の遊休時間

r : 単位時間当たりの金利

$F_{\max}$: スケジュールの最大滞留時間

R : スケジューリング結果により発生する収入

E : スケジューリング結果により発生する支出

M : スケジューリング結果により発生する利益

PM_i : 利益に関する J_i の評価値

$UB(N_i^q)$: 分岐限界法の上界値

TS : 分岐限界法の評価値

a : 上界式で順序付けされたジョブの個数

$TIME(S)$: 上界式で順序付けされたジョブの最大滞留時間

$TIME(U)$: 上界式で順序付けされていないジョブの最大滞留時間

$J \in J_{[i]}$: 上界式で順序付けされたジョブ $J_{[i]}$ に含まれるジョブ

$J \in \bar{J}_{[i]}$: 上界式で順序付けされたジョブ $J_{[i]}$ に含まれないジョブ

N_i : 分岐限界法の生成された最新ノード

N_i^q : 分岐対象ノード

l : 分岐レベル

3. モデルの定式化

フローショップ・モデルにおける作業工程では、工程間でのジョブの処理において、工程の遊休時間とジョブの待ち時間が発生する。定式化は、遊休時間と待ち時間を考慮して行う。

3.1 遊休時間と待ち時間の計算 スケジューリングによって発生する、工程の遊休時間 $W_{[i],2}$ とジョ

4. 解 法

フローショップ・モデルの利益 M を最大にする最適スケジュールを導出する方法として、最新ノード探索法による分岐限界法を用いる。最新ノード探索法とは、分岐されたばかりのノードで最大の上界値をもつノードを分岐対象ノードとして設定する方法である。

4.1 初期試行解導出アルゴリズム 4.3 節で述べるアルゴリズムの探索数を削減するため、単一工程モデルの利益最大化定理⁽⁶⁾を利用し、初期試行解を求める。

本論文では、定理を利用した式(9)を用いてジョブごとの PM_i を計算し、 PM_i の大きい順に順序付けてスケジューリングを行い、得られた利益 M を暫定解として初期試行解の評価値 TS に設定する。初期試行解導出アルゴリズムを以下に示す。

[初期試行解導出アルゴリズム]

[Step 1] 式(9)を用い、各ジョブの PM_i を求める。

$$PM_i = \frac{S_i r + C_i}{t_{i,1} + t_{i,2}} \quad (i=1, 2, \dots, n) \quad (9)$$

すべてのジョブについて PM_i の値を求め、Step 2へ。

[Step 2] 得られた PM_i の大きい順に、時間軸前方からすべてのジョブを割り付けたスケジュールを作成する。得られた利益 M を、利益最大化スケジュールの初期試行解の評価値とする。

[Step 3] 終了。

4.2 上界式 以下に示す式(10)を用いて利益 M に対する上界値 $UB(N^0)$ を求め、探索手順を決定する。

$$\begin{aligned} UB(N^0) = & \sum_{j \in J_{in}} S_{[j]} \left[1 + r \left\{ TIME(S) + TIME(U) \right. \right. \\ & \left. \left. - \sum_{k=1}^j (W_{[k],2} + T_{[k],2}) \right\} \right] \\ & + \sum_{i=1}^{n-a} s_i \left[1 + r \left\{ TIME(S) \right. \right. \\ & \left. \left. + TIME(U) - \sum_{j \in J_{in}} (W_{[k],2} + T_{[k],2}) \right\} \right] \\ & - \sum_{i=1}^n p_i \{ 1 + r (TIME(S) + TIME(U)) \} \\ & - mc \sum_{i=1}^n t_{i,2} - \sum_{j=2}^n C_{[j]} \left(\sum_{k=1}^{j-1} T_{[k],1} \right) \\ & - \sum_{j \in J_{in}} C_{[j]} Q_{[j],2} \quad \dots \dots \dots (10) \end{aligned}$$

式(10)で $TIME(S)$ は、各ノードで、順序付けされたジョブ $J_{[i]}$ 集合の最大滞留時間を、 $TIME(U)$ は、順序付けされていないジョブ $\bar{J}_{[i]}$ 集合の最大滞留時

間を表している(図3参照)。ここで、

$$DC = \sum_{j \in J_{in}} S_{[j]} + \sum_{j \in J_{in}}^{n-a} S_i - \sum_{i=1}^n p_i \quad \dots \dots \dots (11)$$

として、式(10)を変形すると、

$$\begin{aligned} UB(N^0) = & DC(1 + r TIME(S) + r TIME(U)) \\ & - r \left\{ \sum_{j \in J_{in}} S_{[j]} \sum_{k=1}^j (W_{[k],2} + T_{[k],2}) \right. \\ & \left. + \sum_{i=1}^{n-a} S_i \sum_{j \in J_{in}} (W_{[j],2} + T_{[j],2}) \right\} \\ & - \left\{ mc \sum_{i=1}^n t_{i,2} + \sum_{j=2}^n C_{[j]} \left(\sum_{k=1}^{j-1} T_{[k],1} \right) + \sum_{j \in J_{in}} C_{[j]} Q_{[j],2} \right\} \\ & \dots \dots \dots (12) \end{aligned}$$

となる。式(12)において、順序付けされていないジョブの最大滞留時間 $TIME(U)$ 、および $\sum_{j=2}^n C_{[j]} \left(\sum_{k=1}^{j-1} T_{[k],1} \right)$ 以外の値は既知である。したがって、まず以下の三つの場合に分けて $TIME(U)$ の値を計算し、次に、 $\sum_{j=2}^n C_{[j]} \left(\sum_{k=1}^{j-1} T_{[k],1} \right)$ の最小値を求めることで上界値 $UB(N^0)$ を算出する。

(1) $DC > 0$ のとき

$TIME(U)$ の値を最大にすれば、上界値 $UB(N^0)$ が最大となる。つまり、 $TIME(U)$ を、順序付けされていないジョブ集合における最大滞留時間の最大値とする。

(2) $DC < 0$ のとき

$TIME(U)$ の値を最小にすれば、上界値 $UB(N^0)$ が最大となる。つまり、 $TIME(U)$ を、順序付けされていないジョブ集合における最大滞留時間の最小値とする。

(3) $DC = 0$ のとき

$TIME(U)$ の値によって上界値 $UB(N^0)$ が変わることはないので、 $TIME(U)$ は任意の値でよい。

次に、上界値を計算する際の、順序付けされていないジョブの処理順序については、式(10)の第5項、

$$\sum_{j=2}^n C_{[j]} \left(\sum_{k=1}^{j-1} T_{[k],1} \right) \quad \dots \dots \dots (13)$$

に着目し、式(13)の値が最小となるよう時間軸前方からジョブを割り付けて処理する。これは、第1作業工

Fig. 3 $TIME(S)$ and $TIME(U)$

程での総滞留時間最小化問題における、総在庫コスト最小化問題⁽⁷⁾として解くことができる。したがって、順序付けされていないジョブ $J_{[i]}$ について、

$$CS_i = \frac{c_i}{t_{i,1}} \quad (i=1, 2, \dots, n) \quad \dots\dots\dots (14)$$

として、 CS_i の大きい順に時間軸前方からジョブを順序付けたスケジュールを作成し、上界値 $UB(N_i^0)$ に貢献させる。

4.3 利益最大化スケジュール導出アルゴリズム

利益 M を最大化するスケジュールを計算する際、4.2 節で示した上界式(10)を用い、最新ノード探索手順による分岐限界法を用いる。以下に、分岐限界法のアルゴリズムを示す。

[分岐限界法のアルゴリズム]

[Step 1] $l=0, N_l^0=\phi, TS=-\infty$ とする。

[Step 2] 各ジョブで PM_i を計算し、 PM_i の大きい順にジョブを割り付ける。得られたジョブ投入順序でのスケジューリングの利益 M を初期試行解の評価値とし、 $TS=M$ として設定する。

[Step 3] 分岐対象ノード N_l^0 を分岐し、 $l=l+1$ とする。作成された $n-l+1$ 個の最新ノード N_l について、上界値 $UB(N_l^0)$ を計算する。

[Step 4] 最新ノードで最大の上界値をもつノードを分岐対象ノード N_l^0 として選択する。上界値の等しいノードが複数存在する場合には、 $J \in J_{[i]}$ の中で最後に順序付けされたジョブの PM_i が大きいノードを優先して探索する。 $l=n$ ならば、Step 5 へ、 $l < n$ かつ $UB(N_l^0) \geq TS$ ならば Step 3 へ、 $l < n$ かつ $UB(N_l^0) < TS$ ならば Step 6 へ。

[Step 5] $UB(N_l^0) \geq TS$ ならば、 $TS=UB(N_l^0)$ として Step 6 へ。 $UB(N_l^0) < TS$ ならば、 N_l^0 を消去して Step 6 へ。

[Step 6] 分岐対象ノード N_l^0 に最も近く、かつ TS より大きい上界値をもつノードがあれば、その中で最大の上界値をもつノードを新たに分岐対象ノード N_l^0 とする。 l を対象ノードの分岐レベルに設定し、最新ノードを消去して Step 3 へ。対象ノードが存在しない場合は、Step 7 へ。

[Step 7] 現時点の試行解を最適解とする。

[Step 8] 終了。

5. 数 値 例

5.1 ジョブデータ 上記のアルゴリズムを説明するために、ジョブ数5として問題を解く。数値例のジョブデータを表1に示す。この場合、単位時間当たりの金利 $r \approx 0.000007$ (年金利6%)とする。なお、計

算によって得られた値は、小数点第4位を四捨五入し、第3位までで表す。

5.2 初期試行解の導出 分岐限界法のノード数を減らすため、4.1 節の式(9)を用いてジョブごとの PM_i を計算し、初期試行解を求める。その手順を示す。

[Step 1] 表1の五つのジョブについて、 PM_i を計算する。 PM_i の値を表2に示す。

[Step 2] PM_i の値の大きい順に時間軸前方からすべてのジョブを順序付けて、スケジュールを作成する。初期試行解のスケジュール、利益と最大滞留時間を表3に示す。

得られたスケジュールを初期試行解の評価値 TS として設定する。評価値 $TS=5952.1$ (1000 yen) となる。

5.3 分岐限界法による最適解の導出 図4は、数値例の分岐図である。数値例では、初期ノードとして5個のノードが作成される。

ここで、図4のノード作成番号(1)のノードに関して、上界値の計算方法を具体的に示す。このノードでは、ジョブ J_1 のみが順序付けられている場合の上界値を求めている。

まず、上界値の計算に必要な $TIME(S)$, $TIME(U)$ を計算する。 $TIME(S)$ は、順序付けされたジョブの最大滞留時間となるので、 $TIME(S)=24+12=36$ となる。 $TIME(U)$ は、順序付けされていないジョブの

Table 1 Job data

Job	J_1	J_2	J_3	J_4	J_5
Set-up time (hour)	24	10	23	15	27
Machining time (hour)	12	27	24	16	7
Purchase price (1,000yen)	369	768	196	767	813
Sale price (1,000yen)	1,096	2,590	485	3,489	1,284
Machining cost (yen/hour)	427				
Inventory cost (yen/hour)	189	735	412	143	351

Table 2 PM_i values

Job	J_1	J_2	J_3	J_4	J_5
PM_i	0.006	0.020	0.009	0.005	0.011

Table 3 Initial feasible solution

Job sequence	$J_2 \rightarrow J_5 \rightarrow J_3 \rightarrow J_1 \rightarrow J_4$
Profit (1,000yen)	5,952.1
Total flow time (hour)	115

Fig. 4 The Branch and Bound tree of Numerical example

最大滞留時間となる。ここで、 $DC > 0$ となるので、 $TIME(U)$ は、最大滞留時間の最大値とする。したがって、順序付けされていないジョブを $J_5 \rightarrow J_2 \rightarrow J_3 \rightarrow J_4$ と順序付けた場合の最大滞留時間となり、 $TIME(U) = 117$ となる。

次に、上界値を計算する際の順序付けされていないジョブの処理順序を決定する。各ジョブで式(14)に従って、 CS_i を求める。各ジョブの CS_i の値を表 4 に示す。

したがって、ノード(1)での上界値 $UB(N_1)$ を計算するためのジョブ投入順序は、 $J_1 \rightarrow J_3 \rightarrow J_2 \rightarrow J_5 \rightarrow J_4$ となり、 $UB(N_1) = 5,934.7$ (1 000 yen) が得られる。ここで、ノードで得られた上界値が初期試行解の評価値よりも大きい場合には、分岐対象ノードとしてそのノード以下の探索を行う。上界値が初期試行解の評価値よりも小さい場合は、そのノード以下の探索を行わない。この場合は 5,934.7 (1 000 yen)、初期試行解の評価値が 5,952.1 (1 000 yen) より $5,934.7 > 5,952.1$ となり、このノード以下の探索は行わない。

以下同様の手順で上界値を求め、探索を行った結果得られた最適スケジュールを表 5 に示す。利益の最適値は、5,953.9 (1 000 yen) である。

今回の数値例から得られた実験結果では、探索回数は、ノードの発生回数より 21 回、探索率は、探索回数を完全列挙の場合に発生するノード

Table 4 CS_i value

Job	J_2	J_3	J_4	J_5
CS_i	73.500	17.913	9.533	13.000

Table 5 Optimum solution to maximize total profit

Job sequence	$J_2 \Rightarrow J_3 \Rightarrow J_5 \Rightarrow J_4 \Rightarrow J_1$
Profit (1,000yen)	5,953.9
Total flow time (hour)	111

Table 6 Optimum solution to minimize total flow time

Job sequence	$J_2 \Rightarrow J_4 \Rightarrow J_3 \Rightarrow J_1 \Rightarrow J_5$
Profit (1,000yen)	5,944.0
Total flow time (hour)	106

$\left(\sum_{n=1}^5 \frac{5!}{(n-1)!} = 325 \right)$ で割った値より、 $(21/325) \times 100 = 6.5(\%)$ となる。

次に、時間を評価尺度とした、最大滞留時間最小化評価尺度での最適解を表 6 に示す。得られた利益は、5,944.0 (1 000 yen) である。これは、Johnson アルゴリズム⁽⁸⁾⁽⁹⁾に従ってスケジューリングを行っている。利益最大化評価尺度での最適値と、最大滞留時間最小化評価尺度での最適値との利益差は、9 836 yen であ

った。数値例では、ジョブ数が5と少ないため利益差は小さいが、生産工程の能率化に有効な最大滞留時間最小化スケジューリングにより得られる利益よりも優れた結果となっている。また、数値例の全スケジュールより得られる利益の平均値は、5914.0(1000 yen)となり、提案法との利益差は、39.8(1000 yen)であった。

6. 数 値 実 験

6・1 数値実験結果 従来の時間を評価尺度とした最大滞留時間最小化と、提案する利益最大化を評価尺度とした二つのスケジュール間にどのような差が生

じるのかを調べるため、いくつかの数値実験を行い、以下の結果が得られた。実験に用いたジョブの各数値は、一様乱数によって与え、各100回ランダムにジョブデータを設定してその平均値を調べた。

ジョブ数を10として、年間の金利を0.0(%)から20.0(%)の範囲で1.0ずつ変化させ、実験を行った結果を図5に示す。図5は、金利の変化と利益差(利益最大化スケジュールの利益-最大滞留時間最小化スケジュールの利益)の関係を示している。作業時間 $t_{i,j}$ は、 $1 \leq t_{i,j} < 50$ の整数とし、売却価格 s_i は購入価格 p_i に各コストと作業時間を考慮して与えた。また、製造コスト mc 、在庫コスト c_i については、2000(yen)未

Fig. 5 Interest rate and profit difference

Fig. 8 Job number and profit difference

Fig. 6 Width of process time and profit difference

Fig. 9 Job number and search rate

Fig. 7 Process time ratio and profit difference

Fig. 10 Job number and computation time

満の整数とした。

次に、ジョブ数を10、金利を年間2.5(%)として、乱数によって発生させるジョブの作業時間 $t_{i,j}$ の最大値 y を $0 < y \leq 100$ の範囲で10ずつ変化させ、実験を行った結果を図6に示す。図6は、作業時間最大値の変化と利益差の関係を示している。作業時間の幅が大き (y の値が大き) くなれば、各ジョブの作業時間 $t_{i,j}$ のばらつきが大きくなることになる。

ジョブの段取作業時間と機械作業時間の比率についても変化させ、実験を行った。機械作業時間 $t_{i,2}$ に対する段取作業時間 $t_{i,1}$ の比率を x として、

$$t_{i,1} = t_{i,2} \times x \quad \dots\dots\dots (15)$$

で表し、 $0.1 \leq x \leq 10.0$ の範囲で変化させた場合の実験結果を図7に示す。図7は、作業時間比率の変化と利益差の関係を示している。

また、金利を年間2.5%としてジョブ数を変化させ、実験を行った結果を図8~10に示す。図8は、ジョブ数の変化と利益差の関係を示している。図9、10は、それぞれ、ジョブ数の変化と探索率の関係、ジョブ数の変化と計算時間の関係を示している。なお、CPUはPentium III 500 MHzでC言語を用いてプログラム化した。

6.2 考察 数値実験によって、利益を最大化する最適スケジュールと、従来の方法である最大滞留時間最小化を評価尺度としたスケジュールとの比較を行った結果、すべての実験結果において提案する方法が、滞留時間最小化スケジューリングよりも利益最大化に対して優れていることがわかった。

図5より、金利と利益差(利益最大化スケジュールの利益-最大滞留時間最小化スケジュールの利益)は、比例関係にあることがわかる。

図6より、ジョブの処理時間の幅と利益差は、ジョブの処理時間のばらつきが大きくなれば、利益差が大きくなる傾向にあることがわかる。ジョブが多品種で処理時間に差がある場合、提案する方法が利益の増加に対して有効であることを示している。

図7からは、ジョブの段取作業時間が機械作業時間より長くなれば、長さの割合に比例して利益差が大きくなることがわかる。図8から、ジョブ数と利益差は比例関係になっていることがわかる。また、図9より、ジョブ数が増加すれば、探索率が小さくなることがわかる。図8、9の結果は、提案する方法が、現在の多品種少量生産工程でのジョブの加工作業スケジュールに有効であることを示している。

図10より、ジョブ数が増加すれば、計算時間が増加

することがわかる。特にジョブ数20以上になってからの計算時間が、飛躍的に増加している。ただ、完全列挙法と比較すると大幅な計算時間の短縮化が図れており、提案する方法は、ジョブ数が20までの規模の問題であれば、十分実用に耐えうる有効なものであると考えられる。

数値実験の結果、提案する方法は、現在のように顧客の志向が多様化した多品種少量生産において、有効であることがわかった。

7. 結 言

本論文では、フローショップ・モデルにおける利益最大化のスケジューリング法を提案した。フローショップ・モデルの利益部分をモデル化して定式化を行い、分岐限界法により最適スケジューリングの解法を提案した。また、提案法のアルゴリズムをプログラム化して、数値例による評価を行った。その結果として、従来の時間を評価尺度としたスケジュールよりも利益に関してより良い解を得ることができた。利益差は、ジョブ加工作業時間や価格のばらつきが大きくなるほど広がる傾向にあり、現在の多品種生産が要求される生産工程においては、滞留時間最小化スケジューリングが必ずしも利益を最大化するとはいえないことがわかった。

文 献

- (1) 浅野誠・畑中圭介・太田宏、納期遅れペナルティコスト最小化のジョブショップスケジューリングにおける近似解法、日本経営工学会論文誌、51-3 (2000)、245-253。
- (2) 横山雅夫、組立作業と段取作業を伴う複合フロー・ショップ・スケジューリング、日本経営工学会論文誌、51-4 (2000)、293-305。
- (3) Ishibuchi, H., Misaki, S. and Tanaka, H., Modified simulated annealing algorithms for the flow shop sequencing problem, *European J. Operational Research*, Vol. 81 (1995), 388-398.
- (4) 稲垣公夫、EMS戦略-企業価値を高める製造アウトソーシング、(2001)、4-7、ダイヤモンド社。
- (5) アーサー・アンダーセン、e生産革命、(2000)、104-110、東洋経済新報社。
- (6) 栢木紀哉・柳川佳也・宮崎茂次、単一工程モデルでの利益最大化生産スケジューリング、システム制御情報学会、15-4 (2002)、203-212。
- (7) 世古口謙二・宮崎茂次・太田宏、段取り時間を考慮した製品在庫コスト最小化のスケジューリング、日本経営工学会誌、40-2 (1989)、90-96。
- (8) Conway, R. W., Macwell, W. L. and Miller, L. W., 関根智明 (監訳)、スケジューリングの理論、(1971)、118-126、日刊工業新聞社。
- (9) Johnson, S. M., Optimal two-and three-stage production schedules with setup times included, *Naval Research Logistics Quarterly* 1(1954), 61-68.