バッファ容量を考慮したスケジューリング問題のモデル化手法

玉 置 久*・西 川 禎 一*

Modeling of Job-shop Scheduling Problem with In-process Buffer Capacity

Hisashi TAMAKI* and Yoshikazu NISHIKAWA*

In recent years, as the diversification of demand and the progress of production technology have brought about a wider variety of products, the necessity of scheduling and inventory control is increasing. This paper deals with a job-shop scheduling problem with consideration of the capacities of in-process buffers, and proposes three kinds of methods of modeling the problem. First, the problem is formulated mathematically into a mixed integer programming (MIP) problem, and referring to the MIP, a way of modeling the problem into a disjunctive graph, is presented. Once the problem with in-process buffers is represented by the disjunctive graph, it can be solved effectively by using a conventional algorithm. e.g., a branch-and-bound method, for the ordinary job-shop scheduling problem. Second, the problem is described by another type of the MIP and is modeled using an extended timed Petri net which is useful as a tool for various kinds of simulation. Third, the problem is modeled by using a Gantt chart which is often used to represent a schedule graphically. This model is suitable for developing heuristic methods of scheduling.

Key Words: job-shop scheduling, in-process buffer, modeling, mixed-integer programming, disjunctive graph, timed Petri net

1. はじめに

従来から、生産スケジューリング問題に対する研究が盛んに行われており、生産工程の様々な形態と仕事の種々の特性に応じて、またスケジュールの良さを表わす目的関数の種類に応じて、各種の問題が定義されている。そして近年におけるコンピュータの著しい高速化・大容量化を背景として、それらの問題に対する多くの解法(アルゴリズム)が提案されている^{1)~3)}。しかし、いずれにしても従来の生産スケジューリング問題では、各種の在庫について陽に考慮することはせず、もっぱら生産の効率化(工期の短縮や生産量の増大)のみに注目するものであった。

ところが、現実の生産工程を見れば、原料在庫、半製品(仕掛り品)在庫および製品(完成品)在庫のいずれもが、工程のスケジュールに大きな影響を与えていること

は明らかである。すなわち、生産と在庫は密接に関連しており、それらの計画は本来一体として取り扱われるべき性質のものである。本論文では、生産工程の各段階で仕掛り品在庫を貯える設備(バッファと呼ぶ)を陽に考慮したジョブショップ型のスケジューリング問題について考察する。

在庫管理について見れば、過剰な在庫は大きな金利負担などの損失を伴うので、これを極力減らす方向で合理化を図るいくつかの方式が提案されている^{4),5)}。たとえば、発注点法と定期発注法に代表される伝統的な方式や、カンバン方式、MRP (Material Requirements Planning)のように生産システムの柔軟性を強調するための方式などがある。しかし、生産工程内の各段階における仕掛り品在庫のように、自給自足的な在庫を考慮してスケジューリングを行おうとする場合には、これらの在庫管理方式をそのまま適用することはできない。

以下、本論文では、まず仕掛り品バッファ容量を考慮したジョブショップ・スケジューリング問題を定義するとともに、問題を混合整数計画問題として定式化する。つぎに、この問題に対して分枝限定法などの組合せ最適

* 京都大学工学部 京都市左京区吉田本町

* Faculty of Engineering, Kyoto University, Kyoto
(Received May 2, 1994)
(Revised August 26, 1994)

化手法の適用を可能にするため、選択グラフ⁶⁾、タイム・ペトリネット^{7),8)} およびガントチャートに基づく3種のモデル化手法を提案する。

2. 問題の記述

2.1 ジョブショップ・スケジューリング問題

ジョブショップ型の生産工程におけるスケジューリング問題(以下,単にスケジューリング問題)を以下のように定義する。なお,仕事とは物品などの生産加工に必要な一つ以上の作業の集まりのことで,物品に注目すれば,仕事の処理は物品の加工と見ることができる。また,機械とは人や設備の総称である。

n 個の仕事 J_j ($j=1, \dots, n$) が m 台の機械 M_i ($i=1, \dots, m$) で処理されるものとする。各機械は同時に二つ以上の仕事を処理できないものとする。各仕事 J_j は n_j 個の要素作業 O_u ($u=N_{j-1}+1, \dots, N_j$; $N_j = \sum_{k=1}^j n_k$; $N_0=0$) からなっていて, これら n_j 個の作業 O_u は u の増加順に, 機械 M_{μ_u} (μ_u は作業 O_u が処理される機械の番号を表わす) 上で, 分割なしに所要の時間 p_u で処理されるものとする。さらに, 仕事間にはいくつかの先行関係が存在する。このとき問題は, あるスケジューリング評価基準, たとえばスケジューリングの最大完了時間 $C_{\max}$ が最小となるように, 機械上で各作業が処理される順序, すなわち時間区間を決定することである。

スケジューリング問題は, 作業を節点に, 作業間の先行関係制約を有向弧に対応させた有向グラフで表現できる。このとき各節点には, その節点を表わす作業が処理される機械およびその処理時間を付記しておく。Fig. 1 に, 有向グラフで表わした問題の一例を示しておく。

スケジューリング問題において, 最適スケジュールに近くある程度良いスケジュールの集合として, つぎの3種類が知られている⁹⁾

- セミアクティブ・スケジュール: 各機械上で作業の処理順序を変更することなしには, どの作業の処理開始をも早めることができないスケジュール。
- アクティブ・スケジュール: 各機械上で作業の処

Fig. 1 An example of the job-shop scheduling problem. The symbol below a node indicates the machine, and the numeral above a node indicates the processing time

理順序を変更してもよいとして, どの作業も, ほかの作業の開始時刻を遅らせることなしには, その処理開始を早めることができないスケジュール。

- 遅れなしスケジュール: アクティブ・スケジュールのうちで, いずれかの作業を開始できる時刻に, どの機械も遊休状態にないスケジュール。

これら3種のスケジュール集合の包含関係は,

遅れなし $\subseteq$ アクティブ $\subseteq$ セミアクティブ

のようになる。さらに, 最大完了時間 $C_{\max}$ など, 作業の完了時刻のそれぞれについての非減少関数を目的関数として選んだ場合には,

最適スケジュールが必ずしも遅れなしスケジュールであるとは限らず, アクティブ・スケジュールであって遅れなしスケジュールでない場合がある; 逆にいうと, アクティブ・スケジュールの中に, 最適スケジュールが必ず含まれる,

ということが重要な性質として導かれている⁹⁾。

2.2 スケジューリングと在庫

ジョブショップ型の生産工程では, 外部から入った原材料が加工を施されて半製品となり, それらがさらに幾段かにわたって加工されて出荷対象となる最終製品が生産される。すなわち, 原材料および幾種かの仕掛り品(半製品)が加工されながら, 工程内を流れているという見方ができる。しかし, これらの物の流れは連続的ではなく, 原材料が納入されてから加工が開始されるまでの間, 各加工段階における仕掛り品の処理と処理の間, および最終製品が生産されたのち出荷されるまでの間には, 物の流れに停滞が生じる。このような流れの停滞を受け止めて緩衝の役目をするのが各種の在庫である。特に, 加工工程の各段階で仕掛り品の在庫を適切にもつことによって, 後続する作業を遅滞なく開始し, 生産完了時間の増加を防止することができる。

そこで, ジョブショップ型生産工程における在庫を,

- 原料在庫: 原材料が納入されてから先頭作業(先行作業のない作業)を開始するまでの間の在庫,
- 仕掛り品在庫(半製品在庫): 先行作業が完了してから後続の作業を開始するまでの間の半製品の在庫,
- 製品在庫: 最終作業(後続作業のない作業)が完了してから最終製品を出荷するまでの間の在庫,

の3種に分類し, ここでは特に, bの仕掛り品在庫に注目する。通常, スケジューリングの目的は在庫を削減して生産性を向上させることとされている。しかし, 上で述べたような在庫の効用を積極的に活用するために, 工程内の各段階に適当な容量のバッファ(仕掛り品在庫を置くための設備)を置き, このバッファを有効に利用してス

Fig. 2 An example of the job-shop scheduling problem with in-process buffers
Nodes ○ and ● correspond to operations and buffers, respectively. The numeral above each node ● indicates the capacity of the buffer

ケジューリングを行えば、生産性をより向上させることが期待できる。

2.3 バッファ容量を考慮したスケジューリング問題

従来のスケジューリング問題は、与えられたすべての仕事の処理をそれぞれ1回ずつ行えば完了するような生産工程を対象としてきた。しかし、実際の生産工程の多くでは、仕事は複数回繰り返して処理され、その間に仕掛け品在庫が補充され、また消費されている。したがって、従来型(1回処理型)のスケジューリング問題によって仕掛け品在庫の影響(効果)を調べることはできない。

以下では、ジョブショップ型の生産工程において各作業の直後に仕掛け品を一時的に保持するためのバッファを設け、その作業の処理が完了するとただちに仕掛け品をバッファに入れ、後続の作業を開始するときに、このバッファから一つ取り出すものとする。なお、ここでは、作業間にそれぞれ一つずつバッファが存在する(すなわち、作業を処理することによって生じる半製品の種類ごとに専用のバッファがある)と仮定しておく。そして、このバッファの容量を有限とし、所要数(複数)の最終製品をまとめて生産する場合のスケジューリングを考える。さらに、スケジュールの良さを表わす目的関数は上記の所要数を生産するのに必要な時間(最大完了時間) $C_{\max}$ とし、それを最小化する問題をバッファ容量を考慮したジョブショップ・スケジューリング問題(以下、単にバッファを考慮したスケジューリング問題)と定義する。有向グラフで表わした問題の一例を Fig. 2 に示しておく。

3. 選択グラフ・モデル

3.1 選択グラフ

まず、本論文で用いる選択グラフ(disjunctive graph: DGR)を簡単に説明する⁹⁾。DGRの構成要素は、節点、有向弧および選択弧対(互いに逆方向きの有向弧の対)であり、それぞれはつぎのような意味をもつ。

- 節点：作業(ただし、計算の便宜をはかるため、仮の出発節点と仮の終了節点を追加しておく)。
- 有向弧：あらかじめ定められている作業間の先行関

(a) Disjunctive graph

(b) Example of a feasible schedule

Fig. 3 Disjunctive graph to Fig. 1.

The dummy initial node and the dummy final node are indicated by the label 0 and *, respectively. The disjunctive arc is represented by a dashed arc

係制約。

- 選択弧対：機械条件を満たすために定めなければならない作業間の先行関係。

従来型のスケジューリング問題の場合には、そのグラフ表現(Fig. 1)に、同じ機械で処理される作業間を結ぶ選択弧対を追加することによって、Fig. 3(a)のようにDGRを構成できる。なお、有向弧および選択弧には、その始点に対応する作業の処理時間が重みとして付加される。さらに、このDGR(Fig. 3(a))において、選択弧対から一方の選択弧を消去すること、すなわち対応する2作業間の処理順序を決めることを選択弧対の解消というが、すべての選択弧対を解消して得られるグラフがFig. 3(b)のように閉路を含まなければ、これは一つの実行可能スケジュールを表わしている。このとき、節点0から節点*および節点0から節点 u までのクリティカル・パスの長さが、それぞれ最大完了時間 $C_{\max}$ および作業 O_u の処理開始時刻を表わす^(注1)。

3.2 混合整数計画問題への定式化(その1)

2.3節で定義されたバッファを考慮したスケジューリング問題において、まず各機械上での作業順序を決定変数に選り、混合整数計画問題(mixed-integer programming problem: MIP)として定式化する。MIPを混合整数計画法で解くことは実際には有効でないが、問題を明確に表現できる唯一の方法であり、3.2節では、この

(注1) 節点 x から節点 y に至るパスについて、そのパス上にある重みの和をパスの長さ、これが最大となるようなパスを節点 x から y までのクリティカル・パスとそれぞれ定義する。

MIP をもとに選択グラフ(DGR)モデルを構築する.

$$\begin{aligned} &\text{minimize} \\ &t = t(y) \end{aligned} \quad (1)$$

subject to

$$t_{ik} - t_{i,k-1} \geq p_i \quad (i=1, \dots, N_n; k=1, \dots, D) \quad (2)$$

$$t_{jk} - t_{ik} \geq p_i \quad ((i, j) \in \mathcal{P}; k=1, \dots, D) \quad (3)$$

$$t_{jk} - t_{i,k+b_{ij}} \leq 0 \quad ((i, j) \in \mathcal{P}; k=1, \dots, D-b_{ij}) \quad (4)$$

$$(M+p_i)y_{ik_{ij}k_j} + (t_{ik_i} - t_{jk_j}) \geq p_j \quad (5a)$$

$$(M+p_i)(1-y_{ik_{ij}k_j}) + (t_{jk_j} - t_{ik_i}) \geq p_i \quad (5b)$$

$$(\mu_i = \mu_j; i < j; k_i=1, \dots, D; k_j = \alpha_{ij}, \dots, \beta_{ij})$$

$$t \geq t_{iD} + p_i \quad (i \in \mathcal{F}) \quad (6)$$

$$t_{ik} \geq 0 \quad (i=1, \dots, N_n; k=1, \dots, D) \quad (7)$$

$$y_{ik_{ij}k_j} \in \{0, 1\} \quad (8)$$

ただし,

p_i : 作業 O_i の処理時間,

μ_i : 作業 O_i の処理される機械番号,

b_{ij} : 作業 O_i と O_j の間のバッファ B_{ij} の容量,

M : 十分大きな正定数 ($\geq \max |t_i - t_j|$),

D : 最終製品の所要数(各作業の必要処理回数),

$\mathcal{P}$: 先行関係制約 ($O_i \rightarrow O_j$ のとき $(i, j) \in \mathcal{P}$),

$\mathcal{F}$: 最終作業(後続作業のない作業)の集合,

t : 最大完了時間 $C_{\max}$,

t_{ik} : 作業 O_i の k 回目の処理開始時刻,

$y_{ik_{ij}k_j}$: 作業対 O_i, O_j ($\mu_i = \mu_j$) の処理順序を決める決定変数; M_{μ_i} ($=M_{\mu_j}$) 上で, O_i の k_i 回目の処理が O_j の k_j 回目に行先するとき 1, そうでないとき 0,

とする. (5)式の添字については, $(i, j) \in \mathcal{P}$ のときは,

$$\alpha_{ij} = \max\{1, k_i - b_{ij} - \lfloor p_j/p_i \rfloor\}, \quad (9a)$$

$$\beta_{ij} = \max\{1, k_i - 1\} \quad (9b)$$

とし ($\lfloor x \rfloor$ は x 以下の最大の整数), 一方 $(i, j) \notin \mathcal{P}$ のときは,

$$\alpha_{ij} = 1, \beta_{ij} = D \quad (10)$$

とする.

ここで, 制約条件(2)~(7)について説明しておく.

(2)式は, 作業 O_i の $(k-1)$ 回目の処理が k 回目の処理に先行することを表わす. (3)式は, 作業 O_i が作業 O_j に先行する場合, O_i の k 回目の処理が O_j の k 回目の処理に先行するという条件である. つぎの(4)式はバッファ容量の制約を表わすものである. O_i が O_j に先行する場合, O_i により生産される半製品がバッファ容量分 b_{ij} まで作り置きできることに対応して, O_j の k 回目の処理開始が O_i の $(k+b_{ij})$ 回目の処理開始よりも前であればよいことを表わす. ただし, (4)式では O_i の開始時

点でバッファ B_{ij} の空きを調べていることになる.

(5)式は機械条件を表わす. すなわち, 機械は同時に一つの作業しか処理できないので, 同じ機械を使用する作業対 O_i, O_j の間にはその処理順序が定められなければならないことを表わす. 2 作業 O_i, O_j の間に先行制約が存在するときは, 制約条件(2)および(3)を考慮して添字 k_j の上限および下限を, それぞれ(9)式のように定める. また, 先行制約がない場合には, (10)式により定める. (9)式は, 2 作業 O_i, O_j 間に先行関係制約が存在する場合, 作業 O_i により生産される半製品を O_i と O_j の間にあるバッファ容量分までしか作り置きできないことを考慮して添字を限定し, 不必要な制約式を除去することを目的とするものである. さらに(6)式は, $C_{\max}$ が作業終了時刻の最大値で与えられることを表わしている. (7)式は, 作業の開始時刻が非負となることを保証するためのものである.

3.3 モデルの記述

ここでは, 3.1 節の MIP に基づいて問題を DGR により表現する. これにより, スケジューリング問題に対する厳密解法のうちで最も有効とされている DGR 上での分枝限定法を, バッファを考慮した場合にもそのまま適用することができる.

先に定義したバッファを考慮した問題では各作業が複数回処理されるので, まず最初に各回の処理(以下, 作業単位)に対応して節点を設ける. 節点番号は便宜的に作業番号を表わす数字にその処理回数を添字として付加し, たとえば, 1_2 (O_1 の 2 回目の処理)のように表わす. さらに, 仮想の出発節点 0 と終了節点 * を追加する.

つぎに, 制約条件(2), (3), (4), (6)および(7)に対応する有向弧を付け加える. まず, (2)式に対応する節点 i_{k-1} と i_k とを重み p_i をもつ有向弧で結ぶ. (3)式に対しては, $(i, j) \in \mathcal{P}$ のときに, 節点 i_k と j_k とを重み p_i をもつ有向弧で結ぶ. さらに, (4)式に対応して, $(i, j) \in \mathcal{P}$ であれば節点 j_k と $i_{k+b_{ij}}$ とを重み 0 をもつ有向弧で結ぶ. (6)式については, $i \in \mathcal{F}$ であれば, 節点 i_D と節点 * とを重み p_i をもつ有向弧で結ぶ. (7)式に対しては, 仮想節点 0 と先頭作業 O_i の 1 回目の処理を表わす節点 i_1 とを重み 0 をもつ有向弧で結ぶ.

最後に, 制約条件(5)に対応する選択弧対を付け加える. すなわち, $\mu_i = \mu_j$ であるすべての節点対 i_k, j_l について, i_k から j_l あるいは j_l から i_k に向かう有向弧で結ばれたパスがない場合に, これら 2 節点 i_k, j_l 間を選択弧対で結ぶ. 各選択弧には, その起点となる節点が表わす作業の処理時間を重みとして付加する.

一例として, Fig. 2 の問題を DGR で表現すると Fig. 4 となる. ただし, 最終製品所要数 $D=3$ とした. バッ

Fig. 4 Disjunctive graph to Fig. 2.

Most of the disjunctive arcs are omitted in order to make the graph simple

ファを考慮したスケジューリング問題のDGR表現において作業単位をあらためて作業とみなすと、従来型のスケジューリング問題とまったく同形となる。したがって、バッファを考慮したスケジューリング問題に対して、これまでに提案されているDGR上でのアクティブ・スケジュール生成手順に基づく種々の解法を、そのまま適用できる。

4. タイム・ペトリネット・モデル

一般に、バッファを考慮したスケジューリング問題に対して最適スケジュール(最適解)を求めたい場合には、アクティブ・スケジュール集合での探索を基本とする解法を構成しなければならない⁹⁾。3章でのDGRモデルは、このような分枝限定法の適用を可能にするものである。しかし、ある程度良いスケジュールでよい場合には、探索範囲を遅れなしスケジュール集合に限定することによって、計算時間を短縮できる。さらに、遅れなしスケジュール集合内での最適なスケジュールは大域的な最適スケジュールであることが多く、最適でない場合でも、最適にかなり近いスケジュールであることがわかっていく⁹⁾。以下では、まずバッファを考慮したスケジューリング問題を3章とは異なる形の混合整数計画問題(MIP)に定式化する。つぎにこの定式化を基にして、タイム・ペトリネット(timed Petri net: TPN)⁹⁾で問題をモデル化する。なお、TPNモデル上でのシミュレーションによって得られるスケジュールは、遅れなしスケジュールに限定される。

4.1 混合整数計画問題への定式化(その2)

作業の開始時刻を決定変数として、MIPに定式化する。なお、時間を離散化しておき、単位時間を期と呼ぶこととする。

minimize

$$\sum_{t'=1}^T w_{t'} \quad (11)$$

subject to

$$x_{i,*,T+1} = D \quad (i \in \mathcal{F}) \quad (12)$$

$$x_{i,j,t+1} = x_{ijt} - y_{j,t+1} + y_{i,t-p_i+1} \quad ((i,j) \in \mathcal{P}; t=1, \dots, T) \quad (13)$$

$$y_{it} = u_{itt} = \dots = u_{i,t,t+p_i-1} \quad (i=1, \dots, N_n; t=1, \dots, T-p_i+1) \quad (14)$$

$$\sum_{t=1}^{N_n} \sum_{\mu=c}^T u_{itt'} \leq 1 \quad (c=1, \dots, m; t'=1, \dots, T) \quad (15)$$

$$y_{it} \in \{0, 1\} \quad (i=1, \dots, N_n; t=1, \dots, T) \quad (16)$$

$$0 \leq x_{ijt} \leq b_{ij} \quad ((i,j) \in \mathcal{P}; t=1, \dots, T+1) \quad (17)$$

$$w_{t'} \geq \sum_{i=1}^T u_{itt'} \quad (i=1, \dots, N_n; t'=1, \dots, T) \quad (18)$$

$$w_1 \geq w_2 \geq \dots \geq w_T \quad (19)$$

$$u_{itt'} \in \{0, 1\} \quad (i=1, \dots, N_n; t, t'=1, \dots, T) \quad (20)$$

ただし、3.1節での記号に加えて、

T : $C_{\max}$ の上限値(あらかじめ設定)、

x_{ijt} : バッファ B_{ij} の第 t 期の半製品在庫数、

y_{it} : O_i の開始時刻(期)を表わす決定変数; O_i の処理が第 t 期に始められるとき 1, そうでなければ 0,

$u_{itt'}$: 第 t 期に始められた作業 O_i が第 t' 期に行われるとき 1, そうでなければ 0,

とする。なお、最終作業 O_i ($i \in \mathcal{F}$) の直後のバッファを B_{i*} と記すことにし、このとき $(i, *) \in \mathcal{P}$ とする。

ここで、制約条件(12)~(15)および(17)について説明しておく。(12)式は、第 $(T+1)$ 期に最終製品在庫数がその所要数 D に等しくなることを表わす。(13)式は、バッファ B_{ij} の第 $(t+1)$ 期の在庫数を、作業 O_j が第 $(t+1)$ 期に開始される場合は第 t 期の在庫数から 1 減じ、さらに作業 O_i が第 t 期に完了する場合には 1 加えた値とすることを表わす。(14)式は、第 t 期に始められた作業 O_i の処理が p_i 期間連続して行われるという条件である。(15)式は、各期においてそれぞれの機械上でたかだか一つの作業しか行われなことを表わす。(17)式は、バッファ B_{ij} の在庫数が非負でかつその容量を超えないという条件である。

さらに、目的関数(11)で最小化される $w_{t'}$ は、

$$w_{t'} = \max \left\{ \sum_{i=1}^T u_{itt'} \right\} \quad (21 a)$$

$$= \begin{cases} 0, & \sum_{i=1}^{N_n} \sum_{t=1}^T u_{itt'} = 0 \text{ のとき} \\ 1, & \text{そうでないとき} \end{cases} \quad (21 b)$$

と表わすことができ、 $\sum_{t'} w_{t'}$ を最小化することは、1 台以上の機械が処理を行っている期間をできるだけ少なく

すること, いい換えると, すべての機械が休んでいる期間をできるだけ多くすることに対応する。さらに(19)式により, 遊休期間は後のほうに集中する。この遊休期間を除くと $C_{\max}$ が最小となるスケジュールが得られるので, (11)~(20)式で表わされる MIP は, $C_{\max}$ 最小化問題と本質的に等価であるといえる。

4.2 モデルの記述

4.1 節のように時間を離散化して MIP に定式化すると, バッファを考慮したスケジューリング問題は, 半製品到着, 処理, (処理の施された)半製品の送り出しという, 一連のいわゆる離散的な事象が繰返し発生する離散事象システムにおける問題とみなすことができる。この離散事象システムを表現する有力な手法として, ペトリネット(Petri net: PN)⁷⁾, あるいは PN に時間の概念を導入したタイム・ペトリネット(TPN)⁸⁾がある。以下では, TPN を用いてバッファを考慮したスケジューリング問題を表現する手法を示す。

(1) タイム・ペトリネット(TPN)

純粋な PN では時間経過を表現することができないので, 時間に依存する要素を含むシステムを表現する場合には PN を拡張する必要がある。TPN は PN のトランジション(TR)に発火継続時間をもたせることによって, システムの時間依存性を表現できるようにしたものである。なお, TPN のグラフ表現においては, 発火継続時間をもつ TR を “ $\boxed{}$ ” で表示するものとし, 通常発火継続時間が 0 の TR (“ $|$ ” で表示する)と区別する。

(2) TPN の拡張

TPN の TR を作業に, プレース(PL)をバッファにそれぞれ対応させることにより, バッファを考慮したスケジューリング問題を TPN で表現することができる。このとき, TR の発火で作業の処理を, PL(バッファPL)内のトークン数でバッファ内の半製品の数を表わすことになる。バッファごとにさらにもう一つの PL(バッファ容量 PL)を設け, そのトークン数でバッファの空き容量を表現する場合には,

- ある作業が開始される時点で, 直前のバッファから半製品が取り出されて空き容量が一つ増えることに対応して, TR が発火を始める瞬間に直前のバッファ容量 PL にトークンを送る,
- その作業の処理が完了してはじめて半製品ができることに対応して, TR が発火を終える時点でバッファ PL にトークンを送る,

という二つの要求を満足しなければならない^(注2)。すなわち, ある動作が行われるとき, その開始時点においてある条件を成立させ, 終了時点でまた別の条件を成立させるということである。

通常の TPN では, このようにシステムの状態を 2 段階に変化させる動作を一つの TR で表わすことはできない。そこで, TR に 2 種類の出力プレースをもたせて TPN を拡張した拡張 TPN を提案する。Fig. 5(a)に拡張 TPN の単位モデルを示すが, 黒丸を介して接続される出力 PL ⑥には TR が発火する時点でトークンが送られ, 白丸を介して接続される出力 PL ⑤には発火が終了する(発火してから τ だけ経過した)時点でトークンが送られる。

2 種類の TR, すなわち発火に単位時間要する TR (1-TR)と即時に発火が終了する TR (0-TR)を定義すると, Fig. 5(a)は等価的に Fig. 5(b)のように表現できる。ただしある時刻において, まず発火可能な 0-TR を発火させてから, つぎに 1-TR を発火させて時間を進めるというように, TR の発火に優先関係を設けておく。この等価表現(b)を用いると TR や PL の数は増えるが, トークンに付随する時間の概念を省略できるので, 解析やシミュレーションを行う場合には有効である。一方(a)の表現法は図を簡略化できるという利点がある。

(3) 拡張 TPN によるモデル化

(a) A basic model of extended timed Petri net.

A circle and a box indicate a place and a transition, respectively. The transition can be fired if there are more than one tokens in the place A, and the firing must continue for τ periods. One token is to be put in the place B just when the firing of the transition starts, and when the firing is finished, one token is to be put in the place C

(b) An equivalent representation of the basic model (a).

Defining two kinds of transitions, i.e., 0-transition “ $|$ ” whose firing is to be finished just when the firing starts and 1-transition “ $\boxed{}$ ” whose firing is to continue for one period, a basic model (a) can be equivalently represented by (b)

Fig. 5 An example of the extended timed Petri net

(注2) バッファ容量 PL を省略し, バッファ PL にトークン数の上限を設けることによってこれらの要求を満たすことができるが, 本論文では, 分枝限定法の構成を容易にするために以下に述べる方法で TPN を拡張した。

Table 1 Comparison between MIP and extended TPN

Extended TPN model	MIP representation
Buffer PL	Constraint (13)
Capacity PL	Constraint (17)
Machine PL	Constraint (15)
Material PL and Demand PL	Constraint (12)
Natural firing	Constraints (14) and (19)
Firing of 0-TR	Variable $y=1$
Firing of 1-TR	Variable $u=1$

Fig. 6 Extended timed Petri net to Fig. 2

4.1節のMIPにおける制約条件と、拡張TPNのPLおよびTRとを**Table 1**のように対応づけることにより、問題を拡張TPNで表現する。一例として、Fig. 2の問題を拡張TPNで表現すると**Fig. 6**のようになる。

このTPNモデル上でシミュレーションを進める際には、原則として、0-TRおよび1-TRを自然発火させる。すなわち、時間を単調に進めながら、ある時刻においてまず発火可能な0-TRを発火させ(処理開始可能作業を始めることに対応)、つぎに1-TRを発火させる(その時刻に開始される作業および処理中の作業を1単位時間実行することに対応)ということを繰り返す(自然発火規則)。この自然発火規則に基づいて生成されるスケジュールは、遅れなしスケジュールに限定されるので、分枝限定法などを適用することにより、最適(あるいは最適に近い)スケジュールが比較的高速に求められる。

5. ガントチャート・モデル

3章のDGRモデルおよび4章のTPNモデルは、ある評価基準を最小あるいは最大にするスケジュールを数理計画的手法によって求めるためのものであり、スケジュールの生成される過程を把握しにくい。したがって、これらのモデルはヒューリスティックな解法を構成する場合には適さない。そこで、バッファを考慮した問題に対してガントチャート(Gantt chart: GTC)に基づいたモデル化手法を示す。

Fig. 7 An example of the earliest processing time slot and the standard time slot.

Among the operations O_4 , O_6 and O_7 which are possible to start processing, the operation O_6 has the minimum earliest finish time. Hence, the standard time slot is the earliest processing time slot of O_6 .

ガントチャートとは、横軸に時間、縦軸に機械をとって、各作業が処理される時間区間を図示したものである。GTCモデルでは、ガントチャート上に作業を一つずつ割り付けていくことにより、すなわち、作業の処理時間区間を必要処理回数分だけ順に決めていくことによりスケジュールを作成する。まず、すでにその処理時間区間が決定している作業を実行したと仮定し、必要処理回数に達していない作業について、

- 直前の(それぞれの)バッファに半製品が一つ以上入られる時刻の最大値 t_1 ,
- 直後のバッファに空きができる時刻 t_2 ,
- 使用する機械が空く時刻 t_3

を計算する。ただし、それぞれの時刻が未定の場合は ∞ とする。さらに、 $\max(t_1, t_2, t_3) < \infty$ となる作業を処理開始可能作業と定義する。

すべての処理開始可能作業について、まずその最早開始可能時刻($\max(t_1, t_2, t_3)$)と最早終了可能時刻、すなわち最早処理可能時間区間(以下単に可能区間)を求める。そして、処理開始可能作業の中から最早終了可能時刻の最も早いものを選び出し、その作業を候補作業、その可能区間を基準区間と呼ぶ(**Fig. 7**)。

候補作業以外の処理開始可能作業の可能区間が基準区間と重なっていない場合は、候補作業の処理をその可能区間(基準区間)に行う(ことを決定する)。重なっていれば、何らかの方法で一つの作業を選んでそれを処理する。このような作業の処理時間区間の決定をすべての作業の必要処理回数が満たされるまで繰り返すと、一つのアクティブ・スケジュールが得られる。

ここで、DGR, TPN および GTC を用いた3種類のモデルにおけるスケジュール生成過程を図示すると、**Fig. 8**のようになる。DGRモデルでは、あらかじめすべての作業が割り付けられているが、同一機械上での作業の処理時間区間に重なりがあり実行可能ではない。これらの

(a) Disjunctive graph model.

First, all operations are assigned to the machines according to the earliest processing time slot of operations. Then, each conflict between operations is settled successively

(b) Timed Petri net model.

At every time step, the operations which are possible to start processing is assigned to the machine successively

(c) Gantt-chart model.

Each operation is assigned to the machine successively

Fig. 8 Process of generating a schedule.

デルでは、作業が一つずつ順に割り付けられることによって、実行可能スケジュールが得られる。ただし、TPNモデルでは(離散化された)時間の一つずつ進めながら処理開始(割付け)可能作業が調べられるのに対して^(注3)、GTCモデルでは、開始可能時刻に関係なく処理開始可能作業が抽出される。

6. む す び

本論文では、ジョブショップ型の生産工程における仕掛り品在庫バッファの容量を考慮したスケジューリング問題を定義するとともに、混合整数計画問題への定式化を通して、選択グラフ、タイム・ペトリネットおよびガントチャートによる3種のモデル化手法を提案した。また新たに、拡張タイム・ペトリネットの概念を導入した。これらのモデルによって、組合せ最適化のための分枝限定法などの手法の適用が可能となることを示した。

参 考 文 献

- 1) E. L. Lawler, J. K. Lenstra and A. H. G. Rinnooy Kan : Recent Development in Deterministic Sequencing and Scheduling: A Survey, *Deterministic and Stochastic Scheduling*, 35/73 (1982)
- 2) K. P. White, Jr. : Advances in the Theory and Practice of Production Scheduling, Control and Dynamic Systems **37**, Advances in Industrial Systems, 115/157 (1990)
- 3) J. Blazewicz, K. Ecker, G. Schmidt and J. Weglarz : Scheduling in Computer and Manufacturing Systems, Springer-Verlag (1993)
- 4) 水野幸男 : 在庫管理入門, 日科技連 (1974)
- 5) 黒田, 田部, 圓川, 中根 : 生産管理, 朝倉 (1989)
- 6) 鍋島一郎 : スケジューリング理論, 森北 (1974)
- 7) J. L. Peterson : Petri Net Theory and the Modeling of Systems, Prentice Hall (1981)
- 8) 離散事象システム研究専門委員会編, ペトリネットとその応用, 計測自動制御学会 (1992)
- 9) 今野, 鈴木 : 整数計画法と組合せ最適化, 日科技連 (1982)

[著 者 紹 介]

玉 置 久 (正会員)
(Vol. 31, No. 5 参照)

西 川 禎 一 (正会員)
(Vol. 31, No. 5 参照)

重なりを順に解消していくことによって、実行可能スケジュールが得られる。一方、TPNモデルおよびGTCモデル

(注3) 時間ごとに処理開始可能作業を調べていくという条件が付加されることにより、TPNモデルに基づいて得られるスケジュールは遅れなしスケジュールに制限される。なお、DGRモデルおよびGTCモデルで得られるスケジュールはアクティブ・スケジュールである。