複数の納期がある
バッチスケジューリング問題における
納期ずれ総和最小化問題

毛　利　進太郎

概　要

本稿では，食品生産におけるバッチスケジューリング問題について考察する。鮮度が重視される食品を生産する場合，製造から販売までの時間を短くすることが求められる。そこで製造現場では1日に何度か，製品が発注元の要求に応じ出荷される。そのとき出荷する時間までに要求された量の製品の製造を完了していなければならない。本稿ではこのような制約を持つ食品生産の工程を考え，納期内に需要量を超える生産量を確保するという制約を満たし，かつ生産終了から納期までの時間の総和を最小とするバッチスケジュールを求める問題を考察する。

1　はじめに

近年，食に対する関心の高まりなどから，食品生産の現場に対し多くの要求が課せられつつある。また流通の発達により小売店などの需要も多様化し，その需要に応じた生産計画が求められる。例えばパン生産の現場では，生産から時間を経るとパンの味が落ちるため，小売店は需要が多い時間に応じて出荷を要求する。ただし小売店はスーパーからコンビニエンスストアまでその業態は多岐に渡り，それぞれの店舗によって営業時間や製品が多く販売される時間が異なる。パン工場ではこれら小売店の需要に応じて日に数回の出荷を行っている。ここではその納期（出荷）までに要求された量の製品を生産しなければならない。またパン生産の現場では製品は製造するパンの量に合わせて原材料を

混ぜ合わすことから始まり，その原材料の量を1つのバッチとするような生産が行われる。本稿ではこのような状況下での総滞留時間を目的関数とするスケジューリング問題を考える。バッチスケジューリング問題はこれまでも多く研究されている[1，4，6，7]。バッチスケジューリング問題では一般にバッチの処理時間をセットアップ時間とバッチに含まれる仕事の数に比例する処理時間とを加えたものとして定義される。すなわちセットアップ時間を s，1つの仕事の処理時間を p，バッチに含まれる仕事の数を n とするとバッチの処理時間は $s+pn$ と定義される。本稿では納期が存在し，納期までに定められた量の製品の製造を終了しなければならないという制約がある問題を扱う。筆者は[3]においてこの条件下において，総滞留時間を最小とするスケジュールを与える方法を提案した。そこでは実行可能なスケジュールをグラフとして表し，そこで動的計画法に基づくアルゴリズムによってグラフ上のパスとして最適なスケジュールを求められることを示した。本稿ではこの方法を拡張し，各仕事の生産終了から納期までの時間の総和を最小とする方法を提案する。

2　問題の定式化

本稿では以下の記号を用いる。

n：仕事の数

s：バッチのセットアップ時間

p：それぞれの仕事の処理時間

b_j：j 番目に処理されるバッチ

$|b_j|$：j 番目のバッチに含まれる仕事の数

C_j：バッチ b_j に属する仕事の完了時間

n 個の仕事のスケジュール S は $(b_1, \cdots, b_k)$ のベクトルとして定義される。ここで k はバッチの総数であり，バッチ b_j はその添え字の順に処理されるとする。さらに複数の納期が存在し，その納期までに要求された量の仕事を処理しなければならないという制約がある場合を考える。ここで納期集合 $D=$

$\{d_1, d_2, \ldots, d_m\}$，$d_{k-1} < d_k$ とし，それぞれの納期 d_k に対し需要 $|d_k|$ が定められているものとする。

この問題に目的関数は各仕事の完了時間と納期の差の総和である。仕事はどの納期に対して，処理が終了している限り，どのように割り当てても納期ずれの総和は同じである。そこで処理の完了は早いものから早い納期に割り当てるものとする。バッチ数 k のスケジュールの各仕事の終了時間と納期との差の総和 $f(S)$ は以下のようにあらわされる。

$$
\begin{aligned}
f(S) &= \sum_{j=1}^{k} \{(d_j - C_j)\,|b_j|\} \\
&= \sum_{j=1}^{k} d_j |b_j| - \sum_{j=1}^{k} \left\{ \sum_{l=1}^{j} \{s + p\,|b_l|\} \right\} |b_j| \\
&= \sum_{j=1}^{k} d_j |d_j| - \sum_{j=1}^{k} \left\{ \sum_{l=1}^{j} \{s + p\,|b_l|\} \right\} |b_j|
\end{aligned}
$$

この問題の目的は各仕事の終了時間と納期との差 $f(S)$ を最小化するようなスケジュール，すなわち n 個の仕事の分割を決定することである。

本稿では，さらに制約として納期が存在し，それまでに需要を満たす製品数を製造しなければならないという問題を考える。この問題を扱うため以下のグラフ $G(V, E)$ を考える。頂点集合 V，枝集合 E を以下のように定義する。

$V = \{v(i, B)\,|\,$仕事数 $i(i=1, \ldots, n)$，バッチ数 $B(1, \ldots, i)\} \cup v(0, 0)$

$E = \{e(v(i_2, b_2), v(i_1, b_1))\,|\,v(i_2, b_2)$ から $v(i_1, b_1)$ への有向枝，

ただし $b_2 = b_1 + 1$, $i_1 < i_2\}$

また頂点 $v(i, B)$ に対し，仕事数 i，バッチ数 B の完了時間 C を定義する。

$C(v(i, B)) = Bs + ip$

このときバッチ数 B，仕事数 n のバッチスケジュール $S_{(B,n)}$ は $v(n, B)$ から $v(0, 0)$ への有向路として表される。またバッチ b_k は有向枝 $e(v(i_2, b_1+1), v(i_1, b_1))$ として表され，$|b_k| = i_2 - i_1$ となる。

このとき，仕事数 n のバッチスケジュール $S_{(B,n)}$ は頂点 $v(n, B)$ から $v(0, 0)$ への有向路としてあらわされ，バッチ b_k は有向枝 $e(v(i_2, b_1+1),$

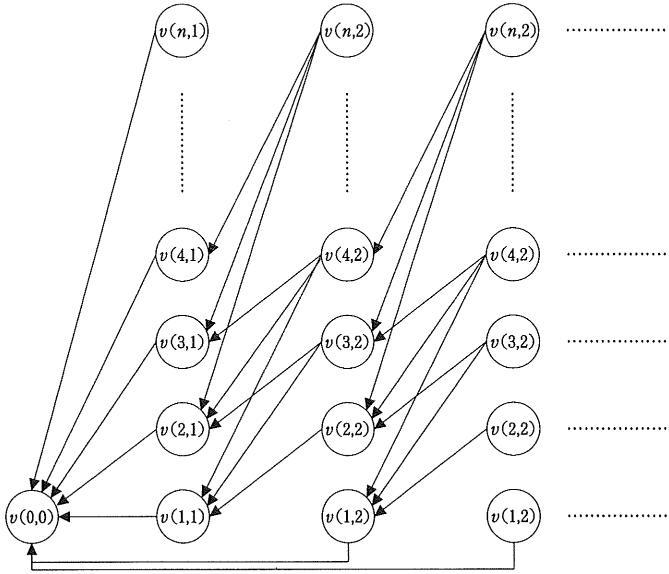

図1．グラフ $G(V, E)$

$v(i_1, b_1)$）として表され，$|b_k|=i_2-i_1$ となる。さらに納期制約を満たすスケジュールを求めるために，各納期 d_i に対しグラフ $G(V, E)$ の頂点集合 E を以下の 4 つ部分集合に分類する。

(1) $V_1=\{v|$ 納期 d_k に対し，$f(v)\leqq d_k$ かつ $i\geqq|d_k|\}$

すなわち納期 d_k に対し納期の時間制約を満たし，かつ需要を満たす頂点の集合

(2) $V_2=\{v|$ 納期 d_k に対し，$f(v)>d_k$ かつ $i<|d_k|\}$

すなわち納期 d_k に対し納期の時間制約を満たさず，需要も満たさない頂点の集合

(3) $V_3=\{v|$ 納期 d_k に対し，$f(v)\leqq d_k$ かつ $i<|d_k|\}$

すなわち納期 d_k に対し納期の時間制約を満たすが，需要を満たさない頂点の集合

(4) $V_4=\{v|$ 納期 d_k に対し，$f(v)>d_k$ かつ $i>|d_k|\}$

すなわち納期 d_k に対し納期の時間制約を満たさないが，需要を満たす頂点の集合

ここであるスケジュール S を考える。Sのグラフ上でのパスが納期 d_k に関する頂点の部分集合 V_1 を含むとき，そのスケジュールSは納期 d_k に関する制約を満たす。Sのグラフ上でのパスが納期 d_k に関する頂点の部分集合 V_2 を含むとき，関数 f は増加関数であることより納期に間に合いかつ需要を満たす頂点はパス上に存在せず制約を満たさない。次にSのグラフ上でのパスが納期 d_k に関する頂点の部分集合 V_3 を含むとき，関数 f は増加関数であることから納期に間に合いかつ需要を満たす頂点は V_3 より前のパス上にのみ存在する。またSのグラフ上でのパスが頂点の部分集合 V_4 を含むとき，関数 f は増加関数であることから納期に間に合いかつ需要を満たす頂点は V_4 以降のパス上にのみ存在する。

以上の議論よりスケジュール S は納期 d_k に対し $\forall v_i \in V_i$ とすると v_2 を始点または終点に含む枝，v_4 から v_3 への枝を含んでいる場合納期の制約を満たさない。また f が増加関数であることから v_1 または v_3 から v_4 への枝は存在しない。

これより実行可能なスケジュールを求めるにはグラフ $G(V, E)$ より

1．頂点部分集合 V_2 と，この頂点を端点に持つ枝集合を削除する。

2．V_4 から V_3 への枝を削除する。

という操作を行ったグラフ $G'(V, E)$ に対し，最適なスケジュールを求めれば

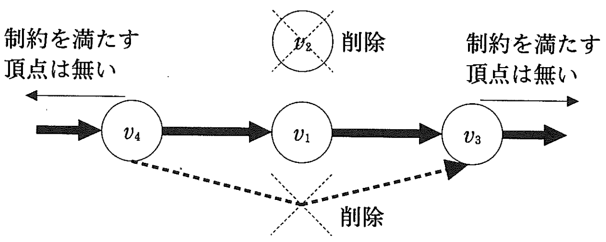

図2．実行可能スケジュールと頂点集合

複数の納期があるバッチスケジューリング問題における納期ずれ総和最小化問題

よい。

問題の目的は目的関数 $f(S)$ を最小化することであるが,

ここで $g(S)=\sum_{j=1}^{k}\left\{\sum_{l=1}^{k}\{s+p|b_l|\}\right\}|b_j|$ とすると,

$$f(S)=\sum_{j=1}^{k}d_j|d_j|-\sum_{j=1}^{k}\left\{\sum_{l=1}^{j}\{s+p|b_l|\}\right\}|b_j|$$

$$=\sum_{j=1}^{k}d_j|d_j|-g(S)$$

となり, $\sum_{j=1}^{k}d_j|d_j|$ は定数であることから, $f(S)$ を最小化する問題は $g(S)$ を最大化する問題に置き換えることができる。このとき, バッチの仕事数, バッチの総数に制約がない場合, 以下の定理が成立する。

定理 1

S_n^* を仕事数 n の最適スケジュールとする。そのとき以下の式が成立する。

$$g(S_n^*)=\max_{|b_1|=1,\ldots,n}\{s+p|b_1|n+f(S_{n-|b_1|}^*)\}$$

ただし S_0^* とし, スケジュール S_n^* のバッチを b_j とし, スケジュール $S_{n-|b_1|}^*$ のバッチを b_j' としたときに $|b_j|=|b_{j-1}'|$ とする。

定理 1 より, 動的計画法によって, 最適スケジュールを求めることができる [2, 5]。

定理 I とグラフ $G(V, E)$ より最適スケジュールを求めるアルゴリズムを次のように構築できる。

これより最適なスケジュールを求めるアルゴリズムは以下のようになる。

AlgorithmI

begin

すべての頂点 v に対し, 完了時間 $C(v)$ を計算する。

for $i=0$ to m begin

納期 d_k に対し

頂点部分集合 V_2 と，この頂点を端点に持つ枝集合を削除する。

V_4 から V_3 への枝を削除する。

end

for $i=0$ to n begin

if ($i=0$) then

$g(S_i)=0$

else begin for $j=1$ to i

頂点 $v(i, j)$ に対し枝 $e(v(i, j), v(i_{j-1}, j-1))$ が存在し，かつ $g(v(i, j))$ が最大となる頂点 $v(i_{j-1}, j-1)$ を選択し，$g(v(i, j))$ を計算する。

end

end

納期制約を満たすバッチ数を B_s とする。

頂点集合 $v(n, j)$ ($j=1, 2, \cdots, B_s$) より $g(v(n, j))$ が最大となる頂点を $v(n, B)$ とし，$v(n, B)$ から $v(0, 0)$ へのパスより各バッチのサイズを決定する。

end

AlgorithmI により，各納期制約を満たし，かつ納期の完了時間との差の総和 $f(v(n, j))$ が最小となる仕事数 n の最適バッチスケジュール $S_{(n, B)}$ を求めることができる。

このとき，仕事数 n の最適バッチスケジュール $S_{(B, n)}$ は総滞留時間 $f(v(n, j))$ が最小となる頂点 $v(n, B)$ から $V(0, 0)$ への有向路としてあらわされ，バッチ b_k は有向枝 $e(v(b_1+1, i_2), (b_1, i_1))$ として表され，$|b_k|=i_2-i_1$ となる。

3 おわりに

本論文では一機械で一種類の仕事を行い総滞留時間を最小にするというバッチスケジューリング問題を考え，スケジュールをグラフとして考えることで，複数の納期と需要の制約がある問題に対し最適なスケジュールを与えるアルゴリズムを示した。本論文では最大完了時間を最小とする問題を扱ったが，食品生産の現場では鮮度が問題となる。そこで納期ずれを考察する必要があると思われる。また生産現場においては多機械，多種類の仕事という環境も多くそれらにたいする研究も今後の課題である。

参考文献

[1] D. Naddef and C. Santos, One-pass batching algorithms for the one-machine problem, *Discrete Applied Mathematics* 21 (1988) 133-145.

[2] 毛利進太郎，1機械バッチスケジューリング問題への動的計画法の適用，神戸学院大学経済学論集，第33巻第3号（2001）41-47

[3] 毛利進太郎，複数の納期があるバッチスケジューリング問題，神戸学院大学経済学論集，第38巻第1，2号（2007）115-123

[4] C. N. Potts and M. Y. Kovalyov, Scheduling with batching: A review, European Journal of *Operational Research* 120 (2000) 228-249.

[5] C. Santos and M. Magazine, Batching in single operation manufacturing systems, *Operations Research Letters* 4(3) (1985) 213-218.

[6] D. F. Shallcross, A polynomial algorithm for a one machine batching problem, *Operations Research Letters* 11 (1992) 213-218.

[7] C. S. Sung and U. G. Joo, Batching to minimize weighted mean flow time on a single machine with batch size restrictions, *Computers and Industrial Engineering* 21(2) (1997) 333-340.