設備割付処理の並列化によるSynPIXのMRP高速化

Accelerate MRP Calculation of SynPIX using Parallel Resource Allocation

製造業では，事業のグローバル化により PSI（生産・販売・在庫）の多拠点・多段階にまたがる SCM が求められ，MRP で対象とするデータ量が膨大になっている。その一方で，グローバル市場の急激な変化に応じた迅速な意思決定が求められ，MRP の高速化が望まれている。そこで，自社製品の生産計画システム SynPIX のグローバル展開に向けて性能改善を行った。大規模データを対象とする MRP の高速化を実現するために設備割付処理の並列化の実装方法が課題となる。本研究では，生産可能設備の重複の有無に基づき並列に実行できるタスクを定義し，さらにタスク間の実行順序を表す有向グラフを生成して並列化する手法を考案した。評価実験により提案手法の有効性を確認した。

黄　双全　Huang Shuangquan
宗形　聡　Munakata Satoshi
岡崎　司　Okazaki Tsukasa
手塚　大　Tezuka Masaru
海老名　拓　EbinaTaku

１．はじめに

近年, 事業環境の変化や製造コストの削減などにより，製造業の海外進出が加速度的に進行している。生産・販売・在庫(PSI）のグローバル化により多拠点・多段階にまたがるサプライチェーン・マネジメント(SCM)が求められ，資材所要量計算(MRP)の取り扱いデータ量が膨大になっていきている。一方，グローバル市場の需要変化が激しい中，飛び込み受注やそのキャンセル，あるいは納期や数量の変更などが絶え間なく発生する。このような状況に応じて迅速な意思決定を支援するため, MRP の高速化が望まれている [1,2]。

実務で利用される MRP では，設備が有限能力であることを前提に実行可能な計画立案を保証するため，部品表(BOM)のローレベル毎に所要量計算を行うと同時に設備割付も行う [3,4]。従来の MRP は，所要量計算を品目単位で並列化することで高速化されてきた [5,6]。しかし，生産オーダ（品目をいつまでにいくつ生産するか）の設備割付は逐次的に処理されているため，さらなる高速化が困難であった。

本稿では，ディスパッチング法による設備割付処理の並列化手法として，生産可能設備の重複の有無に基づき並列に実行できるタスクを定義し，さらにタスク間の実行順序を表す有向グラフを生成することで並列化する手法を考案した。

２．MRP 計算過程と設備割付問題

実務で利用される MRP の計算過程は総所要量計画，正味所要量計算，ロットまとめおよび設備割付から構成される [4,7]。図 1 には MRP の計算過程を示す。

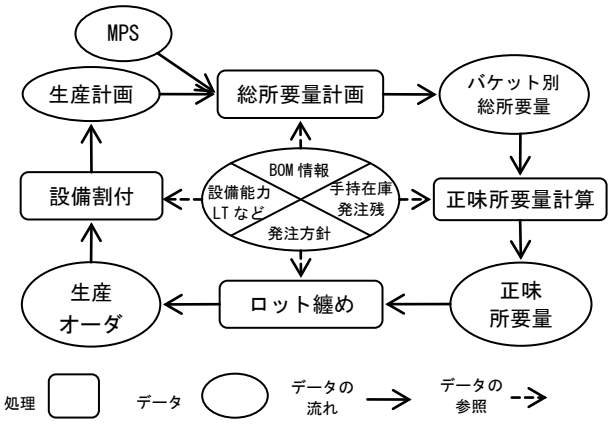


図１　MRP の計算過程

総所要量計画は，基準生産計画(MPS)または親品目の従属需要（親品目の生産計画数×構成数）からバケット毎の総所要量を求める処理である。正味所要量計算は，総所要量から手持在庫と発注残を引いて正味所要量を計

算する処理である。ロットまとめは，定量発注方式や定期発注方式などを用いて生産オーダ（オーダ）を求める処理である。設備割付は，品目の生産量と納期を指定したオーダに対し，オーダの最早完了日（EFT）と最遅完了日（LFT）の間で品目を生産可能な設備群の中からいつどの設備でいくつ作るのか決定し，生産計画を立案する処理である。ここで，LFT は納期－運送 LT で，EFT は LFT－割付可能期間＋1 で設定される。本論文では，生産を開始して最早に完成する日から最遅に完成する日までの期間をオーダの割付可能期間とし，生産された品目が生産設備から発送されて所定の場所まで運ばれる運送時間を運送 LT と記述する。

設備割付処理では満たさなければならない条件は下記の 3 点である。

- オーダ毎に指定された EFT と LFT の間にしか割り付けないこと
- オーダで指定された品目の生産可能な設備にしか割り付けないこと
- 生産計画量が生産可能設備の能力(あるいは単位時間に品目を生産できる数量)を超えないこと

複数のオーダを生産可能設備に割り付けるとき，設備割付の最適化問題は式(1)から(6)までのような線形計画問題で表すことができる。

$$\text{目的関数} \quad \sum_{o\in O}\sum_{r\in R}\sum_{t\in T} w_{ort}x_{ort} \quad \rightarrow \text{最大} \tag{1}$$

$$\text{制約条件} \quad \sum_{r\in R}\sum_{t\in T} x_{ort} \le q(o) \quad \forall o \in \boldsymbol{O} \tag{2}$$

$$\sum_{o\in O} S_{p(o)r}x_{ort} \le C_{rt} \quad \forall r \in \boldsymbol{R}, \forall t \in \boldsymbol{T} \tag{3}$$

$$x_{ort} = 0 \quad t \notin T(o), \forall o \in \boldsymbol{O} \tag{4}$$

$$x_{ort} = 0 \quad r \notin R(p(o), t), \forall o \in \boldsymbol{O} \tag{5}$$

$$x_{ort} \ge 0 \quad \forall o \in \boldsymbol{O}, \forall r \in \boldsymbol{R}, \forall t \in \boldsymbol{T} \tag{6}$$

ここで，$\boldsymbol{P}$，$\boldsymbol{O}$，$\boldsymbol{R}$，$\boldsymbol{T}$はそれぞれ品目の集合，割付対象となるオーダの集合，割付先となる生産設備の集合，立案期間のバケットの集合を表す。$p(o) \in \boldsymbol{P}$，$q(o)$，$EFT(o) \in \boldsymbol{T}$，$LFT(o) \in \boldsymbol{T}$，$T(o) \subseteq \boldsymbol{T}$，はそれぞれオーダ$o \in \boldsymbol{O}$の生産品目，要求数量，最早完了日，最遅完了日，割付可能なバケットの集合$\{t \in \boldsymbol{T} | EFT(o) \le t \le LFT(o)\}$を表す。$R(p,\mathrm{t}) \subseteq \boldsymbol{R}$は品目$p \in \boldsymbol{P}$のバケット$t \in \boldsymbol{T}$での生産可能設備の集合を表す。$x_{ort}$はオーダ$o$に対してバケット$t \in \boldsymbol{T}$に設備$r \in \boldsymbol{R}$で生産する計画量を表し，設備割付問題の決定変数となる。w_{ort}はオーダoの生産計画x_{ort}の重みを表し，オーダの利益などを考慮したオーダの優先順位，生産コストなどを考慮した設備の優先順位，在庫保管コストなどを考慮したバケット間の優先順位などに基づいて設定される。C_{rt}とS_{pr}はそれぞれバケット$t \in \boldsymbol{T}$で設備$r \in \boldsymbol{R}$の生産能力，設備$r \in \boldsymbol{R}$で品目$p \in \boldsymbol{P}$を生産する際に単位あたりの消費能力を表す。$q(o)$, x_{ort}, C_{rt}, S_{pr}は生産するものの種類によって決まる。例えば，個数を単位とする部品などでは整数，液体などでは実数として使う。

また，目的関数の式(1)はすべての生産オーダに対する生産計画の利益を最大化することを表す。また，未割付数量総数は$\sum_{o\in O}(q(o) - \sum_{r\in R}\sum_{t\in T} x_{ort})$となる。制約条件の式(2)から(6)まではそれぞれオーダoに対する生産計画量が要求数量を超えないこと，バケットtに設備rでの生産計画量の消費能力がバケットtで設備rの生産能力を超えないこと， EFT と LFT の間に割り付けること，生産計画を生産可能設備にしか立案しないこと，生産計画量が非負であることを表す。

３．ディスパッチング法による設備割付処理の課題

3.1 ディスパッチング法による設備割付処理

設備割付の最適化問題は NP 困難な問題であることが知られており，実用的な時間で目的関数(1)を最大化することは難しい[3,8]。したがって，実務ではまず制約条件(2)から(6)までを満たす解を求めることが必要とされる。そのための手法の一つがディスパッチング法である。なお，制約条件を満たす解は複数存在するが，ディスパッチング法での解の選択手法を「ディスパッチングルール」と呼ぶ。ただし，ディスパッチング法は生産計画の実行可能解（すなわち制約条件を満たす解）を求める方法であり，立案された生産計画で目的関数(1)が最大となることが保証できない。

3.2 設備能力が無限大の設備割付処理の課題

式(2)から(6)までの制約条件のうち式(3)の制約を取り除いて緩和した問題を解くのが「能力無限割付法」である。設備の能力残数がいつも無限大のため任意のオーダoに対して実行可能解は直接に$x_{o\tilde{r}\tilde{t}} = q(o)$と設定することが可能である。ここで，$\tilde{t}$と$\tilde{r}$はそれぞれ$EFT(o)$と$LFT(o)$の間の優先順位が最も高いバケット，$R(p(o), \tilde{t})$の優先順位が最も高い設備である。そのため，設備割付処理は単純に各オーダoに対応するバケット$\tilde{t}$を求める LT 計算処理として考慮できる。すると，設備割付処理は図１の他の三つの処理と同じく品目間で独立して行うこと

ができる。この特徴を用いて，所要量計算を品目単位で並列化することで MRP の高速化を図る方法がある。しかし，実際には設備の能力が有限であり，単純な LT 計算処理だけで実行可能な計画立案が困難である。

3.3 設備能力が有限の設備割付処理の並列化課題

ディスパッチングルールで定めたオーダの順位に沿って割付処理を行い，同時に割り付けられた分で設備の能力残数を更新する。つまり，複数の品目を同一の設備で生産可能の場合には，優先順位の高いオーダの割付結果(生産計画)は優先度の低いオーダの割付処理に影響を与える。そのため，設備割付処理はオーダ単位または品目単位で並列化することが困難である。

図 2 は品目P_1のオーダO_1と品目P_2のオーダO_2を割り付ける例を示す。品目P_1と品目P_2の生産可能設備をそれぞれR_1とR_2，R_2とR_3とする。納期等を考慮して仮にオーダO_1を先に割付けるとすると，オーダO_1の割付処理が終わるまで設備R_2の能力残数は確定しないため，オーダO_2の割付処理はオーダO_1の処理が完了するまで開始できない。オーダO_2を先に割付ける場合も同様である。このように,複数品目の間で生産可能設備が重複する場合には,設備の競合が発生するためオーダの割付は逐次的な処理に限定される。これが設備割付の並列化を困難とする大きな要因である。

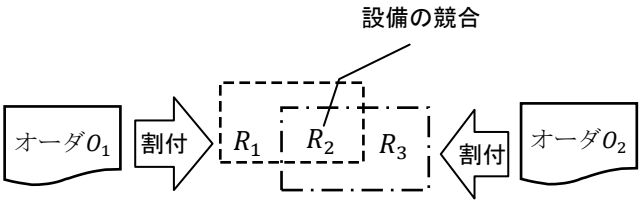


図 2　生産可能設備の重複による設備競合

4．ディスパッチング法による設備割付処理の並列化手法の考案

本章では，ディスパッチング法による設備割付処理の並列化手法を提案する。4.1 節で生産可能設備の重複の有無に基づいた並列化手法を考案し， 4.2 節で提案手法で発生するオーバヘッドを抑制するアプローチについて述べる。

4.1 生産可能設備の重複の有無に基づいた並列化

複数の品目の生産可能設備に重複がある場合には，各品目のオーダの設備割付処理が逐次的になる。しかし逆に，生産可能設備がお互いに重複しない品目のオーダ同士間での設備割付処理は並列化できる。そこで，生産可能設備の重複の有無に基づいた並列化手法を考案する。

4.1.1 並列処理単位となるタスクの定義

設備割付処理を並列化するにあたって，並列割付処理単位となるタスクとその実行順序を決定する必要がある。そこで，タスクとその順序関係を表現する方法として有向グラフを用いた。本節では，並列割付処理単位となるタスクを定義し，さらにタスク間の実行順序を規定する有向グラフを生成する。

並列割付処理単位となるタスクの定義を説明する前に，立案品目の集合$\boldsymbol{P}$の最分割を導入する。まずは式(7)から(9)までを用いてバケットtでの集合$\boldsymbol{P}$の分割を定義する。

$$\boldsymbol{P} = \bigcup_{1 \le i \le n} \Omega_{ti} \tag{7}$$

$$\Psi_{ti} = R(\Omega_{ti}) = \bigcup_{p \in \Omega_{ti}} R(p,t) \tag{8}$$

$$\Psi_{ti} \cap \Psi_{tj} = \emptyset, \quad i \neq j, 1 \le i,j \le n \tag{9}$$

ここで，$\{\Omega_{t1}, \Omega_{t2}, \Omega_{t3}, \cdots, \Omega_{tn}\}$が品目集合$\boldsymbol{P}$のバケット$t$での一つの分割を表し，$n$がその分割の要素(品目群)の数である。式(8)と式(9)はそれぞれ，品目群Ω_{ti}に属する品目をバケットtでの生産可能な設備の集合(設備群Ψ_{it}), 異なる設備群の積集合が空集合である（つまり，品目群の間で生産可能設備が重複しない）ことを示す。品目集合$\boldsymbol{P}$の式(7)から(9)までを満たすすべての分割から構成した集合を$\boldsymbol{\Gamma}_t$とすると，$\boldsymbol{\Gamma}_t$での「細分」関係が半順序である [9]。品目集合$\boldsymbol{P}$が有限のため，集合$\boldsymbol{\Gamma}_t$の下界（或いは最も細分化されている分割）が唯一に存在する。本稿では，その下界を集合$\boldsymbol{P}$の最分割と呼び，バケットtでの集合$\boldsymbol{P}$の最分割を求める手順は図 3 に示す。

s1：品目集合$\boldsymbol{P}$から品目リスト LM を生成．整数 N←リスト LM の要素の数，インデックス変数 k←0．**s2** へ．

s2：品目群を格納するリスト LMG を生成．LMG←∅で初期化．**s3** へ．

s3：$k <$ Nのとき，品目リスト LM から品目 LM[k]を取り出し，k=$k+1$．**s4** へ．　$k =$ Nのとき，**s9** へ．

s4：設備リスト LR←品目 LM[k]のtでの生産可能設備．**s5** へ．

s5：LMG が∅の場合，**s7** へ．LMG が∅でない場合，**s6** へ．

s6：LMG 内の品目群に対応する生産可能設備群が LR と重複する設備があるかどうかを判断し，あれば，**s8** へ．なければ，**s7** へ．

s7：品目群{LM[k]}を LMG へ新規要素として追加．**s2** へ．

s8：重複設備がある品目群Θ_1，…，Θ_mの和集合を$\cup_{1 \le j \le m} \Theta_j$にし，$\Theta_1$，…，$\Theta_m$の削除と{LM[k]}∪($\cup_{1 \le j \le m} \Theta_j$)の追加で LMG を更新．**s2** へ．

s9：終了．

図 3　品目グルーピングのアルゴリズム

s3 から **s8** までの処理では，処理済みの品目の集合に対して品目群を格納する LMG がいつも最分割であるため，**s9** での LMG の品目群は集合$\boldsymbol{P}$の最分割であることが分かる。

続いて，バケットtでの上記の手法を用いて生成した品目群Ω_{ti}に基づき，品目群と一対一となる割付処理のタスクT_{ti}を構成する。タスクT_{ti}では，割付先となる生産可能設備は品目群Ω_{ti}を紐付けられる設備群Ψ_{ti}にする。割付対象となるオーダは各オーダの EFT と LFT に基づき構成され，オーダ群Φ_{ti}は式(10)で求められる。

$$\Phi_{ti} = \{o | p(o) \in \Omega_{ti}, t \in T(o), o \in \boldsymbol{O}\} \tag{10}$$

一方，タスクT_{ti}での処理は割付処理の時間範囲により，バケットtだけで生産計画を求める処理(処理方法Ⅰ)と，オーダの EFT と LFT の間で生産計画を求める処理（処理方法Ⅱ）を二つ定義できる。二つの処理方法に対して，タスクT_{ti}で求められる生産計画の集合X_{ti}はそれぞれ式(11)，(12)で表すことができる。

$$X_{ti} = \{x_{\bar{o}\bar{r}\bar{t}} | \forall \bar{o} \in \Phi_{ti}, \forall \bar{r} \in \Psi_{ti}, \bar{t} = t\} \tag{11}$$

$$X_{ti} = \{x_{\bar{o}\bar{r}\bar{t}} | \forall \bar{o} \in \Phi_{ti}, \forall \bar{r} \in \Psi_{ti}, \forall \bar{t} \in T(\bar{o})\} \tag{12}$$

さらに，タスク間の実行順序を表す有向グラフを生成する。タスクが同じバケットに属する場合には，設備が重複しないためタスク間での並列実行が可能である。しかし，異なるバケットに属するタスク間で依存性が発生する場合，同時に並列実行できない。この依存性はタスクT_{ti}の処理方法によって異なる。処理方法Ⅰの場合は，オーダ群間でオーダの重複の有無だけにより決まる。一方，処理方法Ⅱの場合は，バケットt以外でも割付可能のために設備群間での設備の重複の有無を考慮することも必要となる。

オーダが重複したような依存性が発生するタスク間に実行順序を表す有向辺を持たせる。有向辺の向きは割付方式がフォワードかバックワードかによって決まる。例えば，割付方式がフォワードの場合，時間順の早いバケットの方の優先度が高く，バケット$t(1 \le t \le N-1)$のタスクから$t+1$のタスクへ向かう。隣接するバケットtと$t+1$に属するタスクT_{ti}と$T_{t+1,j}$に有向辺を持たせる条件として，処理方法Ⅰの場合は式(13)が，処理方法Ⅱの場合は式(13)と(14)が同時に成り立つことである。

$$\Phi_{ti} \cap \Phi_{t+1,j} \neq \emptyset \tag{13}$$

$$\Psi_{ti} \cap \Psi_{t+1,j} \neq \emptyset \tag{14}$$

式(13), (14)はそれぞれオーダ群間でオーダが重複すると，設備群間で生産可能設備が重複することを表す。

図 4 のデータを例として生産可能設備の重複の有無に基づいたタスクの定義および有向グラフの生成を説明する。図 4 の上は立案期間を 1 から 4 とした場合の各品目の生産可能設備を示している。例えば，品目P_1の生産可能設備はR_1とR_2であり，設備R_1は 1 から 3 の全体にわたり品目P_1を生産可能で，設備R_2は 1 から 4 で品目P_1を生産可能である。図 4 の下は各品目のオーダを示している。例えば，オーダO_1は品目P_1のオーダで，EFT が 3， LFT が 4 となる。

立案期間	1	2	3	4
品目P_1	R_1 R_2			R_2
品目P_2	R_2 R_3		R_3 R_4	
品目P_3	R_4 R_5		R_4	
品目P_4	R_5			

オーダ	品目	EFT	LFT
オーダO_1	P_1	3	4
オーダO_2	P_2	1	2
オーダO_3	P_2	2	3
オーダO_4	P_3	1	2
オーダO_5	P_4	4	4

図 4　品目の生産可能設備（上）とオーダ（下）

バケット毎に生産可能設備の重複の有無に基づいて生成した品目群は図 5 に示す。例えば，バケット 1 では，$\Omega_{11} = \{P_1, P_2\}$と$\Omega_{12} = \{P_3, P_4\}$は生産可能設備が重複しない品目群となる。

立案期間	1	2	3	4
品目P_1	Ω_{11}	Ω_{21}	Ω_{31}	Ω_{41}
品目P_2			Ω_{32}	Ω_{42}
品目P_3	Ω_{12}	Ω_{22}		
品目P_4			Ω_{33}	Ω_{43}

図 5　設備の重複の有無に基づいた品目グルーピング

続いて，バケット毎に品目群と一対一となるタスクは図 6 のように定義する。例えば，タスクT_{11}のオーダ群と生産可能設備群はそれぞれ$\{O_2\}$，$\{R_1, R_2, R_3\}$である。

バケット	品目群	タスク	オーダ群 Φ_{it}	設備群Ψ_{it}
1	Ω_{11}	T_{11}	$\{O_2\}$	$\{R_1 \quad R_2 \quad R_3\}$
	Ω_{12}	T_{12}	$\{O_4\}$	$\{R_4 \quad R_5\}$
2	Ω_{21}	T_{21}	$\{O_2 \quad O_3\}$	$\{R_1 \quad R_2 \quad R_3\}$
	Ω_{22}	T_{22}	$\{O_4\}$	$\{R_4 \quad R_5\}$
3	Ω_{31}	T_{31}	$\{O_1\}$	$\{R_1 \quad R_2\}$
	Ω_{32}	T_{32}	$\{O_3\}$	$\{R_3 \quad R_4\}$
	Ω_{33}	T_{33}	$\emptyset$	$\{R_5\}$
4	Ω_{41}	T_{41}	$\{O_1\}$	$\{R_2\}$
	Ω_{42}	T_{42}	$\emptyset$	$\{R_3 \quad R_4\}$
	Ω_{43}	T_{43}	$\{O_5\}$	$\{R_5\}$

図 6　タスクのデータ情報

最後に，タスクの実行順序を規定する有向グラフを生成し，処理方法 I の場合の有向グラフを図 7 に示す。例えば，T_{11}とT_{21}はオーダO_2を共有しているため，T_{11}の実行が終了するまでオーダO_2の未割付部分が分からず，T_{21}の実行は開始できない。このような場合，T_{11}からT_{21}に向かう辺を持たせ，T_{11}とT_{21}をそれぞれ親タスク，子タスクと呼ぶ。

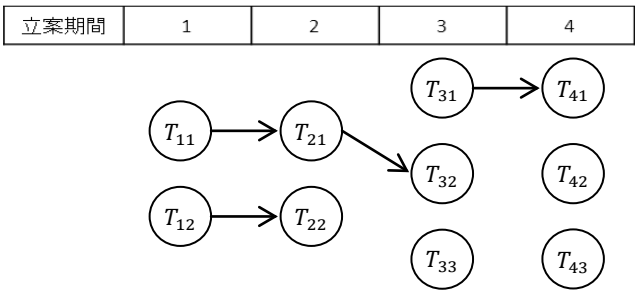


図 7　タスクの有向グラフ

また，バケット毎に品目をグルーピングする処理から有向グラフを生成するまでのすべての処理時間を初期化オーバヘッドと呼ぶことにする。

以上の手法により，並列実行処理が可能なタスクを構築できる。同じバケットに属するタスク（例えば，T_{11}とT_{12}）は並列実行が可能である。異なるバケットに属するタスクについては，辺で規定された順序関係を満たしたときは並列実行できる。例えば，図 7 でT_{11}の実行が終了すればT_{12}の実行終了を待たずにT_{21}の実行を開始できる。

4. 1. 2 タスクの並列実行の同期制御

4.1.1 節に説明したようにタスク間に依存性が発生しているため，タスクの CPU のコアへの割り当ては動的割り当てを利用する。一つの制御コアと複数のワーカコアから構成されるシステムの場合，タスクの有向グラフを用いた動的割り当て手順は図 8 になる。

s1：親タスクを持たないタスク（図 6 では，T_{11}，T_{12}，T_{31}，T_{33}，T_{42}，T_{43}）を用いて実行可能タスクキューQを初期化する．s2 へ．
s2：有向グラフのすべての節点タスクが実行完了であれば，s6 へ．そうでなければ，s3 へ．
s3：Qから先頭タスクを取得し，空いているワーカコアへ割り当てる．s4 へ．
s4：タスクを実行し，実行結果を制御コアへ返す．s5 へ．
s5：オーダの未割付情報を対応する子タスクへ受け継がせ，すべての子タスクの実行可能状態を更新する．実行可能となった子タスクがあったら，Qに追加する．s2 へ．
s6：終了．

図 8　タスクの動的割り当て制御手順

ここで，s4 がワーカコアで並列に，その他は制御コアで逐次に実行される。動的割り当て方式を利用することで，同時に実行可能なタスク数(Q内の実行可能タスク数)がコア数より多ければ，スケジューリングに柔軟性を持たせ，ロートバランスを上手く取ることが可能となる。それにより，コアの遊休時間が少なく，高い並列度での並列実行が可能である。

一方，タスクのワーカコアへ割り当てや，オーダの未割付情報を子タスクへ受け継がせるなどのオーバヘッドが発生するが，これらのオーバヘッドを制御オーバヘッドと呼ぶことにする。タスクの計算量が充分に粗く，並列実行全体からみて制御オーバヘッドが少なければ，高い並列度で並列実行できる。それに対して，タスクの計算量が少なく，制御オーバヘッドの割合が増大すると，設備割付処理が並列化できても並列度が高いと限らない。

4. 2 初期化オーバヘッドと制御オーバヘッドの抑制

前節の提案手法では，タスクの有向グラフを生成するために初期化オーバヘッド(4.1.1 節)が，タスクを並列に実行させるために制御オーバヘッド(4.1.2 節)が発生する。図 9 に従来の逐次処理，考案した並列手法の総時間の構成を示す。

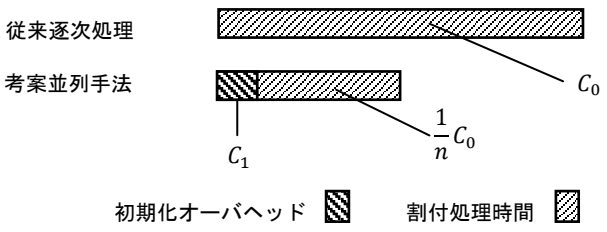


図 9　従来手法と提案手法の処理時間構成

従来の逐次処理では割付処理時間だけから構成される。一方，提案手法では，タスクの有向グラフを生成するための初期化オーバヘッドと，有向グラフを用いた並列割付処理時間から構成される。並列割付処理の並列度をnとすると，提案手法の並列割付処理時間は逐次処理の$1/n$となる。そして，提案手法の高速化率$(S: Speedup)$は不等式(15)で表される [10]。

$$S \le \mathrm{S}_{max} = \frac{1}{\frac{1}{n} + \frac{C_1}{C_0}} \tag{15}$$

ここで，C_0とC_1はそれぞれ従来の逐次処理の処理時間，提案手法による初期化オーバヘッドである。

不等式(15)から，提案手法の高速化率Sは，タスクの有向グラフの生成による初期化オーバヘッドC_1，有向グラフを用いた並列割付処理の並列度nに制限されていることがわかる。立案期間が長い或いはバケット数が多い場合，タスクの構成時間も有向グラフの生成時間も増大するため，初期化オーバヘッドの割合が大きくなる。さらに，生成したタスクが増えることで，タスクのワーカコアへの割り当て回数や，割り付け残ったオーダを子タスクへ受け継がせる回数が多くなり，制御オーバヘッドも大きくなる。制御オーバヘッドが大きくなると，並列度も小さくなることで,提案手法の高速化率が 1 を下回り，従来の逐次処理より時間がかかる恐れがある。

したがって，提案手法の高速化率Sを高めるために，初期化オーバヘッドC_1を小さくすることと，並列度nを大きくするに制御オーバヘッドを小さくすることが必要とされる。そこで，立案期間のバケットマージングと有向グラフの生成処理の並列化を利用してこれらのオーバヘッドの抑制をはかる。

4.2.1 立案期間のバケットマージングによる初期化オーバヘッドと制御オーバヘッドの低減

バケットをマージングする方法としては，同じパケット数で隣接するバケットをマージングする方法や，各設備の生産可能期間の開始や終了の最頻値を用いたマージング法を利用する。バケットをマージングすることで，従来バケット別で品目をグルーピングすることから，複数のバケットから纏められた小期間に渡って品目のグルーピングすることになる。それにより，立案期間が変わらない場合，品目をグルーピングする処理回数が少なくなり，タスクの有向グラフを生成するための初期化オーバヘッドを短縮することが可能となる。

4.1.1 節と同様に，複数バケットから纏められた小期間$\boldsymbol{V} \subseteq \boldsymbol{T}$に渡って設備の重複の有無により品目の集合$\boldsymbol{P}$の分割を式(16)から(18)で定義する。

$$\boldsymbol{P} = \bigcup_{1 \le i \le n} \Omega_{Vi} \tag{16}$$

$$\Psi_{Vi} = \bigcup_{t \in \boldsymbol{V}} R(\Omega_{Vi}, t) = \bigcup_{t \in \boldsymbol{V}} \bigcup_{p \in \Omega_{Vi}} R(p, t) \tag{17}$$

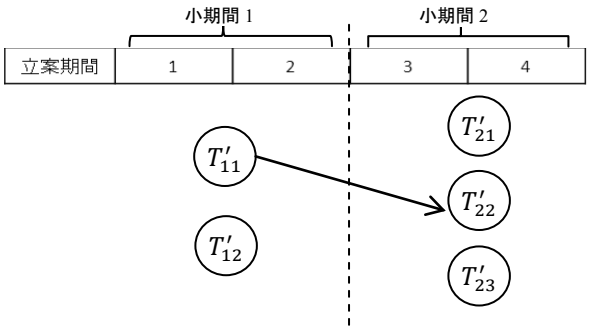
$$\Psi_{Vi} \cap \Psi_{Vj} = \emptyset, \quad i \ne j, 1 \le i, j \le n \tag{18}$$

ここで，小期間$\boldsymbol{V}$に渡って集合$\boldsymbol{P}$の一番に細分されている最分割を$\{\Omega_{V1}, \Omega_{V2}, \cdots, \Omega_{Vn_V}\}$と記す。そして，下記の定理が成り立つ。

定理 1. $\forall \mathbf{V_1}, \boldsymbol{V_2} \subseteq \boldsymbol{T}$に対して，$\mathbf{V_1} \subseteq \boldsymbol{V_2}$のとき，$\mathbf{V_1}$に渡る品目集合$\boldsymbol{P}$の最分割$\{\Omega_{V_1 1}, \Omega_{V_1 2}, \cdots, \Omega_{V_1 n_1}\}$が$\mathbf{V_2}$に渡る集合$\boldsymbol{P}$の最分割$\{\Omega_{V_2 1}, \Omega_{V_2 2}, \cdots, \Omega_{V_2 n_2}\}$の細分である。つまり，$n_2 \le n_1$。

上記の定理は背理法を用いて簡単に証明できるため，ここで証明過程を省略する。定理はバケットをマージングすることで有向グラフの階層を減らすと同時に各階層のタスク数を増やさず，タスクの総数を減らせることを保証できる。そのため，並列割付処理の同期制御時間を短縮することを期待できる。

図 4 の例では，バケット 1 と 2 は小期間 1 に，バケット 3 と 4 は小期間 2 にマージングした場合のタスクの有向グラフ(処理方法 I の場合)は図 10 に示す。



小期間	タスク	オーダ群	設備群
小期間 1	T'_{11}	$\{O_2 O_3\}$	$\{R_1 \quad R_2 \quad R_3\}$
	T'_{12}	$\{O_4\}$	$\{R_4 \quad R_5\}$
小期間 2	T'_{21}	$\{O_1\}$	$\{R_1 \quad R_2\}$
	T'_{22}	$\{O_3\}$	$\{R_3 \quad R_4\}$
	T'_{23}	$\{O_5\}$	$\{R_5\}$

図 10　タスクの有向グラフ（上）とタスクの詳細（下）

このようにすることで,4.1 節の手法で並列処理単位となるタスクに対し，バケットをマージングし，処理量を粗くすることでタスクの数を減らせるようになった。言い換えると，同時に実行できるタスク数がコア数より多

い場合には，本手法を用いて制御オーバヘッドを抑制することで設備割付の並列度を向上することが可能である。

バケットをマージングすることで，各階層のタスク数が減るため，同時に実行できるタスク数がコア数を下回る恐れがある。それにより，コアの遊休時間が増加し，変えて並列度が下がってしまう。そのため，バケットをマージングする前提条件は同時に実行できるタスク数がコア数より多いことである。

4.2.2 有向グラフの生成処理の並列化による初期化オーバヘッドの低減

4.1 節の提案手法の初期化オーバヘッドを抑制するために，タスクの有向グラフを生成するまでの処理を並列化する。ここで，二つの処理について並列化する。

(1)小期間毎にオーダ群を生成する処理を並列化する。例えば，図 7 に示したバケット 1 のタスク（T_{11}とT_{12}）を生成する処理と，バケット 2 のオーダ群（T_{21}とT_{22}）を生成する処理とを並列化する。

(2)隣接する小期間のオーダ群間に有向辺を生成する処理を並列化する。例えば，図 7 に示したバケット 1 のタスクとバケット 2 内のタスクとの間の有向辺を生成する処理と，バケット 2 のタスクとバケット 3 内のタスクとの間の有向辺を生成する処理とを並列化する。

５．評価実験

5.1 評価概要

4 章で述べた並列化提案手法の有効性および汎用性を検証するため，二つの実験データを用いて性能評価を行った。表 1 に実験データの詳細を示す。

表１　実験データ

データ	品目	設備	生産可能設備情報数	オーダ
データ 1	6465	53	16046	46377
データ 2	6366	534	15605	41276

データ 1 は 46377 個のオーダに対して 53 個の生産設備を持ち，一般的な MRP のデータ規模を想定したものである。データ 2 は 41276 個のオーダに対して 534 個の生産設備をもち，グローバル環境下での大規模データを想定したものである。立案期間は 90 日にした。

二つのデータに対して，立案期間のバケットを N 個の小期間にマージングしてから小期間毎に生産可能設備の有無に基づいてタスクの有向グラフ（処理方法Ⅱ）を生成し，一つのタスクの処理を一つのコアに割当る並列処理を行った。バケットのマージング手法として，各設備の生産可能期間の開始や終了の最頻値による方法を用いた。また，実験で用いた PC は OS が Windows Server 2003 Enterprisex64 Edition Service Park 2 で，CPU が Dual-Core AMD Opteron(tm) Processor 2222 SE 3GHz (物理 2 コア×2CPU)で，メモリが 32GB である。

5.2 評価結果

評価結果を図 11 と 12 に示す。提案手法の性能は割付処理の総時間（初期化時間＋割付処理時間）で評価した。従来手法は逐次的に設備割付処理を実行した結果である。また，提案手法の処理時間の傾向を読み取るため，データ 1 とデータ 2 を用いた実験では，実験した小期間数の範囲はそれぞれ 8，12 までにした。

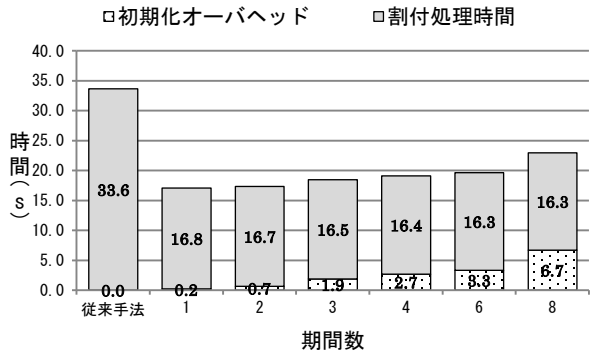


図 11　データ 1 を用いた評価結果

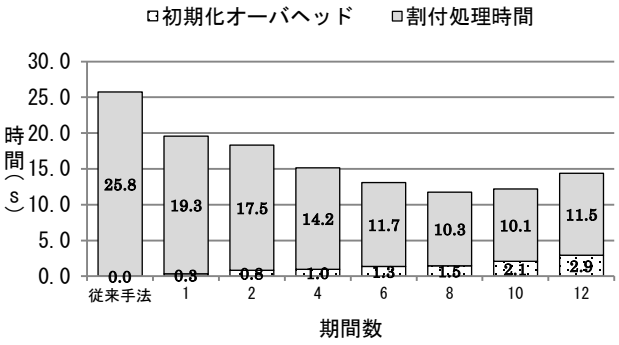


図 12　データ 2 を用いた評価結果

データ 1 とデータ 2 を用いた実験では，提案手法の小期間数 N がそれぞれ 1，8 のときに最も高性能で，割付処理の総時間は従来手法の 1/2 となった。データ 1 を用いた場合，N が 1 から 8 の間に増加しても割付処理時間が変わらない一方で，初期化時間が線形近似的に増加している。データ 2 を用いた場合は，N が 1 から 12 の間に割付処理時間が N=10 のときに最小となったが，初期化オーバヘッドの増加により，N=8 のときよりも割付処理の総時間がかかっている。

データ 1 の評価結果より，設備数が少ない一般規模デ

ータを対象とした MRP では，小期間が 1 にするあるいは立案期間にわたってタスクを生成するときに提案手法がもっとも効果的であることが分かった。一方，データ 2 の評価結果より，グローバルな多拠点・多段階による大規模データを対象とした MRP では，適切な小期間数を選ぶことで提案手法が有効であるといえる。

６．今後の事業展開方針

本稿で述べた設備割付の並列化技術は生産計画システムのスケーラビリティを向上させる。従来は取り扱えなかったデータ量に対しても適切なハードウェアを選択することで，ユーザは意思決定のための生産計画シミュレーションを高速に，限られた時間で何度も実行することが可能となる。処理性能の向上は MRP 逆展開による部材シミュレーション機能など新たな機能の実現にもつながる。今後，本稿で述べた並列化技術を核に，性能向上，機能拡張と順次実施し，グローバル展開する企業への SynPIX 展開を推進していく。

７．おわりに

本研究は設備の有限能力を考慮した MRP の高速化を目的として実施した。ディスパッチング法による設備割付処理の並列化手法として，生産可能設備の重複の有無に基づき品目をグルーピングして，生成した品目群と対応するタスクを定義し，さらにタスクの実行順序を規定する有向グラフを生成するという並列化手法を考案した。また，立案期間のバケットをマージングすることとタスクの有向グラフの生成処理を並列化することを用いて提案手法によるオーバヘッドを抑制し，高速化率の向上をはかった。

最後に，実験により提案手法の有効性を示した。特にデータ 2 を用いた評価結果により，グローバルな多拠点･多段階による大規模データを対象とした MRP では，バケットをマージングするときに小期間数を適切に選ぶことで提案手法がディスパッチング法による設備割付処理の並列化手法として有効であることがわかった。今後の課題として，生産可能設備などのデータの特性から小期間数を自動的に決定する仕組みを検討していきたい。

参考文献

1) 海老名拓，他：SynPIX によるグローバル PSI の実現，日立 TO 技報 18 号 (2012)
2) 黄双全，他：設備割付の並列化による MRP の高速化手法，情報処理学会第 75 回全国大会(2013)
3) Wallace J.Hopp，他： Factory Physics，McGraw Hill Higher Education，3 Edition，2008
4) 今野和幸，他：ハイテク家電業界向け生産計画システム，オフィス･オートメーション 27(4), pp. 50-55 (2007)
5) 築島隆尋，他：並列 MRP システムにおける同期管理方式の開発，電気学会論文誌 C 巻，Vol.122-C，pp.1209-1217 (2002)
6) 築島隆尋，他：並列 MRP システムにおける負荷分散方式の開発，電気学会論文誌 C 巻，Vol.123-C，pp.1847-1857(2003)
7) 鳥羽登：SE のための MRP，日刊工業新聞社，1995
8) 佐藤知一：革新的生産スケジューリング入門―“時間の悩み”を解く手法，日本能率協会マネジメントセンター，2000
9) 松村英之:集合論入門 (基礎数学シリーズ), 朝倉書店, 2005
10) Clay. Breshears：並行コンピューティング技法(千住治郎訳)，オライリー・ジャパン，2011

黄　双全　　2010 年入社
研究開発部
在庫管理，生産計画，シミュレーション技術の研究，開発
shuangquan.huang.dc@hitachi-solutions.com

宗形　聡 2003 年入社
研究開発部
統計・数理的アプローチによる業務分析，意思決定支援技術の研究開発
satoshi.munakata.tu@hitachi-solutions.com

岡崎　司　　1985 年入社
研究開発部
生産計画，組み込みソフトウェアの研究，開発
tsukasa.okazaki.ty@hitachi-solutions.com

手塚　大　　1994 年入社
研究開発部
意思決定，リスク分析，最適化技術の研究，開発
masaru.tezuka.fd@hitachi-solutions.com

海老名　拓　　2000 年入社
PSI ソリューション第一グループ
製造業向け生産計画システムのシステムエンジニアリング
taku.ebina.zc@hitachi-solutions.com