



Title	分散型シミュレーションを用いたリアルタイムスケジューリングシステムの評価
Author(s)	鳩野, 逸生
Citation	大阪大学大型計算機センターニュース. 1997, 106, p. 31-39
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/66237">https://hdl.handle.net/11094/66237</a>
rights	
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

## 分散型シミュレーションを用いたリアルタイム スケジューリングシステムの評価

大阪大学大学院基礎工学研究科

鳩野逸生

### 1 はじめに

近年の消費者のニーズの多様化のため、変種変量の生産様式をとらざるを得なくなっている。このような生産様式に対応するための手段として、

1. 生産設備の高度化, 自動化,
2. 情報ネットワークを用いた生産システムの統合的管理, CIM 化,
3. 生物型生産システム [1, 2] など, 次世代自律分散型生産システムの開発,

などがあげられる。特に, (2) の情報ネットワークを用いた生産システムの統合的管理は, 1つの工場内の統合から, 他の工場, 企業内の他部門 (開発, 営業, 購買など) へと広がっている (CIM: Computer Integrated Manufacturing)。現状では, 生産システムの CIM 化においてはインフラストラクチャの整備が先行し, CIM 環境におけるスケジューリング技術の開発は遅れているのが現状であり, CIM 環境下におけるスケジューリング技術の開発が大きな課題となっている。また, (3) の次世代自律分散型生産システムに関しても, 現在開発されているのは概念と簡単なプロトタイプのみであり, スケジューリング手法など詳細な技術の開発は着手したばかりである。本稿では, 上記の問題に対する 1つのアプローチとして開発されている分散型リアルタイムスケジューリングについて述べる。

### 2 分散型リアルタイムスケジューリング

工作機械の故障およびそれに伴う修理時間, 加工時間のばらつきといった生産システム自体に内在する不確実性や, 特急ジョブの割込, 急なジョブのキャンセルといった外的環境の変化がしばしば生じる不確実な状況下にある生産システムでは, 状況が変化するたびにスケジュールを作り直す必要がある通常のオフラインスケジューリング [3] より, 実際に生産を行ないつつリアルタイムに工程の順序付けや工作機械の割当てなどを行なうリアルタイムスケジューリング [4, 5] の方が有効であると思われる。

一般に, 工作機械および搬送車で構成される生産システムにおいてスケジュールをリアルタイムに発生するためには, (1) 次工程における代替機械の選択 (ルーティング), (2)

搬送車における搬送ジョブの決定, (3) 次に加工を行なうジョブの入力バッファからの選択という 3 つの決定をリアルタイムに行なう必要がある。

本稿では, 必要が生じた時点でリアルタイムに以上の 3 つの決定を行ないながら生産を行なうことをリアルタイムスケジューリングと呼ぶことにする。また, リアルタイムスケジューリングを行なう 1 つのスケジューリングプロセスがシステム全体のリアルタイムスケジューリングを行なう場合を集中型リアルタイムスケジューリングシステムと呼ぶ。

一方, 大規模な半導体生産設備など対象とする生産システムが大規模化した場合, 生産管理システムも従来の大型計算機を用いた集中型のアーキテクチャでは一般に対応が困難であり, 生産管理システムは, 階層型あるいは分散型のアーキテクチャが採用される場合が多い。この場合, リアルタイムスケジューリングシステムも分散型システムとして構築する必要がある。本稿では, 分散型システムとして構築されておりかつシステム中に存在する複数のスケジューリングプロセスが協調してスケジューリングを行なうシステムを分散型リアルタイムスケジューリングシステムと呼ぶことにする。さらに, 集中型と分散型の中間として, 階層型あるいは分散型と階層型のハイブリッド型のシステムも本稿におけるリアルタイムスケジューリングの対象とする。様々なアーキテクチャのリアルタイムスケジューリングシステムの例を **Fig. 1**に示す。

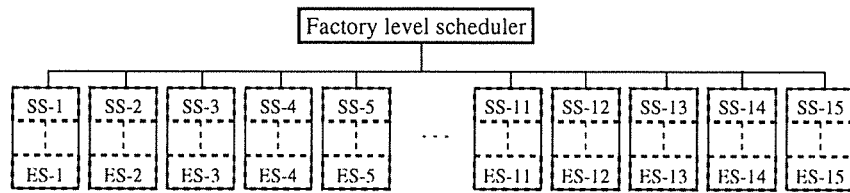
### 3 分散型リアルタイムスケジューリングの問題点

リアルタイムスケジューリングを用いて生産システムを効率に運用するためには, オフラインスケジューリングを解く際には制約として与えられるような条件を満たし, かつスケジューリング目的を達成する必要がある。本稿で取り扱うリアルタイムスケジューリングにおいては, 「次に何をするか」のみをリアルタイム決定するため, (1) デッドロック回避, (2) 工程間の時間的制約, (3) 特急ジョブの処理, などといった必ず守る必要がある重要な制約と, 通常のスケジューリング目的を同時に考慮してスケジューリングを行う必要がある。オフラインスケジューリングにおける探索の場合と異なり, バックトラックを行うことができないため, オフラインスケジューリングと同様な手法 [6] を使うことは実用的でないと思われる。また, 本稿で取り扱うような, 分散型リアルタイムスケジューリングの場合, 各スケジューリングプロセスが独立にスケジューリングを行うため, 各スケジューリングプロセス間に協調動作や他プロセスとの情報交換を行うためのプロトコルを規定しておく必要がある。各スケジューリングプロセス間の通信プロトコルの概要を **Fig. 2**に示す。

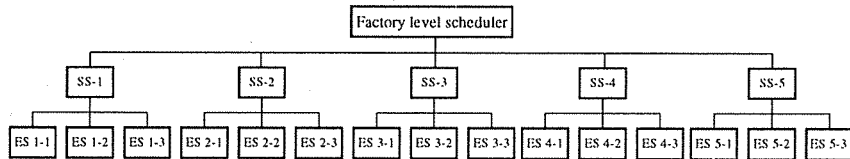
### 4 分散型リアルタイムスケジューリングのための協調プロトコル

分散型リアルタイムスケジューリングを実現するために開発された, 協調プロトコルおよびディスパッチングアルゴリズムを以下に示す。

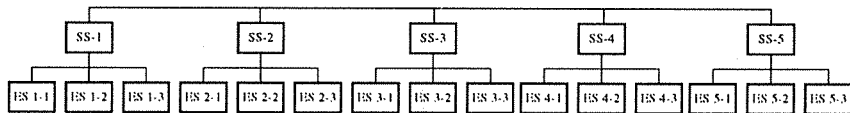
SS: ショップスケジューラ ES: 設備スケジューラ



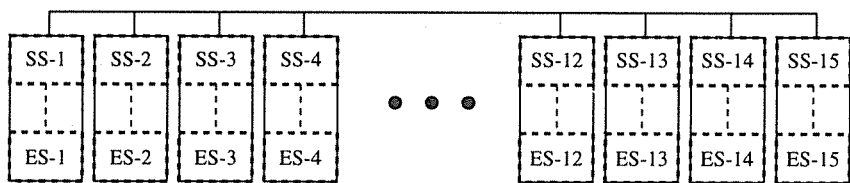
(a) 集中型リアルタイムスケジューリングシステム



(b) 階層型リアルタイムスケジューリングシステム



(c) 分散階層型リアルタイムスケジューリングシステム



(d) 分散型リアルタイムスケジューリングシステム

図 1: 階層型・分散型リアルタイムスケジューリングシステムの例.

- 次の工作機械を選択する (ルーティング) ための協調プロトコル.
- 各工作機械から次に処理するジョブを選択するためのディスパッチングアルゴリズム.
- 各搬送車について、次に搬送すべきジョブの選択のためのディスパッチングアルゴリズム.
- 特急ジョブを処理するための協調プロトコルおよびディスパッチングアルゴリズム.
- 設備故障時のジョブ滞留防止プロトコル.

本稿では、紙数の関係で、次の工作機械あるいは次に処理すべきジョブを選択するための協調プロトコルについてのみ述べる.

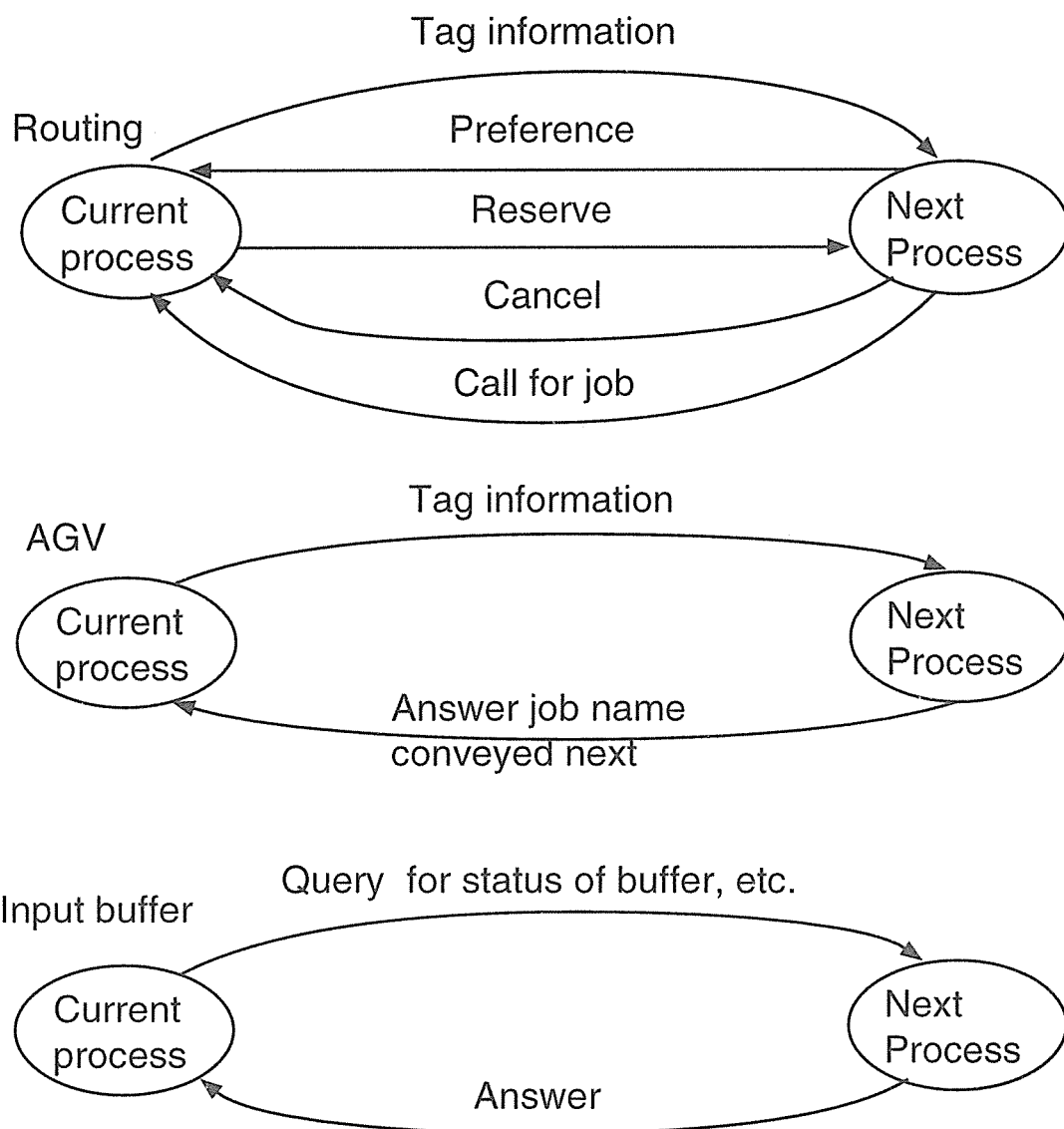


図 2: スケジューリングプロセス間の協調プロトコルの概要.

## 5 ルーティングのための協調プロトコル

ルーティングは、他のスケジューリングプロセスと協調しながら動作する必要があるため、他スケジューリングプロセスとの通信プロトコルを定義する必要がある。本稿では、デッドロック回避、特急ジョブの処理など生産システムを運用する上で重要な機能は、プロトコル自体に埋め込む形で実装している。以下に、ルーティングにおけるスケジューリングプロセス間の通信プロトコルを詳述する。

本システムでは、各機械におけるバッファの空き状態などの管理は、各機械が独立して行なうので、搬送先決定時には空いていたバッファが、搬送先に実際に到着したときすでに空きがない、といったことが起こる可能性がある。このようなことを避けるために、ルーティングされたジョブは、そのルーティング先の入力バッファを予約するようにしている。

さらに、すでに入力バッファの予約を完了しているジョブに対しても、もしそのジョブより緊急度の高いジョブが発生した場合には、予約を解除することによって、緊急度の高いジョブを優先させるといったことに柔軟に対応することができる。本システムにおいて、各特急ジョブではない通常のジョブのルーティングは次のような方法 [5] で行なう。通常ジョブのルーティング：

1. ひとつの工程が完了してその機械の出力バッファに到着したジョブは、全ての次工程の代替機械にタグ情報を提示する。
2. 次工程の代替機械は、そのジョブが来る可能性があるということを“ジョブ依頼可能性リスト”に記録する。
3. 次工程の代替機械のうち、バッファに空きがあるものはそのジョブに対する評価値を、またバッファに空きのないものについては予約解除を行なう場合のペナルティを評価値に乗じたものを最終的な評価値として返す。
4. 各機械の返す評価値をもとに次工程を処理する機械を決定し、その機械の入力バッファを予約する。
5. 決定された次工程の機械が、他のジョブの予約の解除を必要とする場合は、解除されるジョブのすべての代替機械の“バッファ空き待ちリスト”にそのジョブがバッファ空き待ちであることを登録する。
6. 予約が完了したジョブは、全ての次工程の機械に予約完了を通知し、予約が成立した機械以外の機械は、そのジョブをジョブ依頼可能性リストから外す。

予約解除の方法および、どの機械も予約できなかったジョブや予約を解除されたジョブに対する処理については、以下の節で述べる。

## 5.1 予約解除

予約の解除は、あるジョブの次工程の代替機械を評価した結果、予約の解除を必要とする機械が選択された場合に行なわれる。各機械は、ルーティングに対する評価値を要求された際にバッファに空きがない場合、ルーティングの対象となっているジョブ ( $A$  とする) の評価値  $v_A$  と、すでにその機械のバッファを予約している各ジョブの現在の評価値  $v_i$  を比較する。もし  $v_A > v_i$  であるようなジョブ  $i$  が存在すれば、そのジョブ ( $B$  とする) を予約解除ジョブの候補として記録しておく。ジョブ  $A$  の評価値に予約解除を行なうためのペナルティ (パラメータとしてあらかじめ設定しておく) を乗じたものをジョブ  $A$  の最終的な評価値とする。

予約を要求している機械で、ジョブ  $A$  のすべての代替機械の評価値を比較した結果、予約解除を行なう機械が選択されると、解除の候補となっていたジョブ  $B$  の予約を実際に解除し、代わりに現在ルーティング対象となっているジョブ  $A$  が入力バッファの予約を行なう。

ただし、予約解除の対象となるジョブは、予約はしているが実際にはまだ前工程の出力バッファにおいて搬送されるのを待っているジョブに限る。また、予約を解除されたジョブは、どの機械も予約できずにバッファが空くのを待っているジョブと同様に扱う。

## 5.2 バッファ空き待ちジョブに対する処理

すべての代替機械の入力バッファに空きがなく、どの機械も予約できなかったジョブは、次の工程を実行する機械の入力バッファに空きが生じ、予約できる状態になるまで、現在いる出力バッファで待つ。また、予約を解除されたジョブについても同様に予約できるまで待つ。

次工程の候補となっている機械が、このようなバッファ空き待ちジョブに対して行なう処理は次の通りである。

1. 次工程の候補となっている機械のうち入力バッファに空きが生じたものは、“バッファ空き待ちリスト” 中のジョブの中で適当なものを選択し、そのジョブに通知する。
2. 通知を受けたジョブは通知を行なった機械を予約する。

なお、本システムでは、機械の入力バッファに空きが生じた場合、空き待ちジョブがあればそれらのうち納期の最も早いものにその入力バッファを割り当てるようにしている。

## 6 分散型リアルタイムスケジューリングの評価システムの開発

リアルタイムスケジューリングシステムを構築するためには、スケジューラのアーキテクチャやスケジューリングアルゴリズムの評価を事前に十分に行なう必要がある。本稿では、様々なアーキテクチャのリアルタイムスケジューリングシステムのシミュレー

タを構築することを可能とするため、工場層、ショップ層、設備層の3階層に分割し、自由に組み合わせることができるよう設計している。各層における基本的な処理を以下に示す [7]。

**工場層 (工場コントローラ)** 工場層では、主に複数のショップの制御を行なうことで、工場全体管理を行なう。また、各ショップからの情報を収集することで、工場全体の情報を作成する。

**ショップ層 (ショップコントローラ)** ショップ目標に応じて、ショップコントローラに属する複数の工作機械を制御し、ショップ全体の管理を行なう。また、工作機械から情報を収集することで、ショップ情報を作成する。また、工作機械の情報を作成する。

**設備層 (設備コントローラ)** 個々の生産設備の制御を行なう。

以上のような階層構造を記述し、各階層のコントローラの組合せを変更することで、Fig. 1に示すような完全な分散システムから集中システムまでモデル化することができる。

本稿で取り扱う協調・通信プロトコルを離散事象シミュレーションで完全にモデル化することは困難であるため、現実の時間スケールを短縮した仮想生産システムを用いた分散型シミュレーション手法を用いている。

## 7 数値例

ここでは、Fig. 1 (a)~(d) に示すリアルタイムスケジューリングシステムに前述の分散型シミュレーションシステムを適用した例について述べる。ここで、対象とする生産システムは、3台の工作機械で構成された5つのショップからなるものとする。この生産システムで、工程数30のジョブ10種類をそれぞれ3つずつ生産するものとする。

**Fig 3**は、生産システムを構成する工作機械の数を変化させたときの、ショップスケジューラのオペレーション総数の変化を調べたものである。Fig. 3より、分散型、分散階層型は、集中階層型、集中型よりもショップスケジューラが取り扱うオペレーション総数が多くなることが分かる。これは、本スケジューラでは、分散型において、協調プロトコルに契約ネットと用いているためである。そのため、分散度が大きくなると工作機械数に対するショップスケジューラのオペレーション増加率は、集中型に比べて大きくなっている。集中型と階層型では、ショップスケジューラのオペレーション総数は、大きな変化がないことがわかる。また、全てのスケジューリング構成において、工作機械が16台を越えると、工作機械が増加してもショップスケジューラの負荷総数が大きく増えていないことがわかる。これは、工作機械が16台をこえると工作機械が遊休する割合が多くなるため、ショップスケジューラにオペレーション要請をしないからである。このことから、本ジョブを作成するには、工作機械が16台ぐらいが適当であると推測することができる。



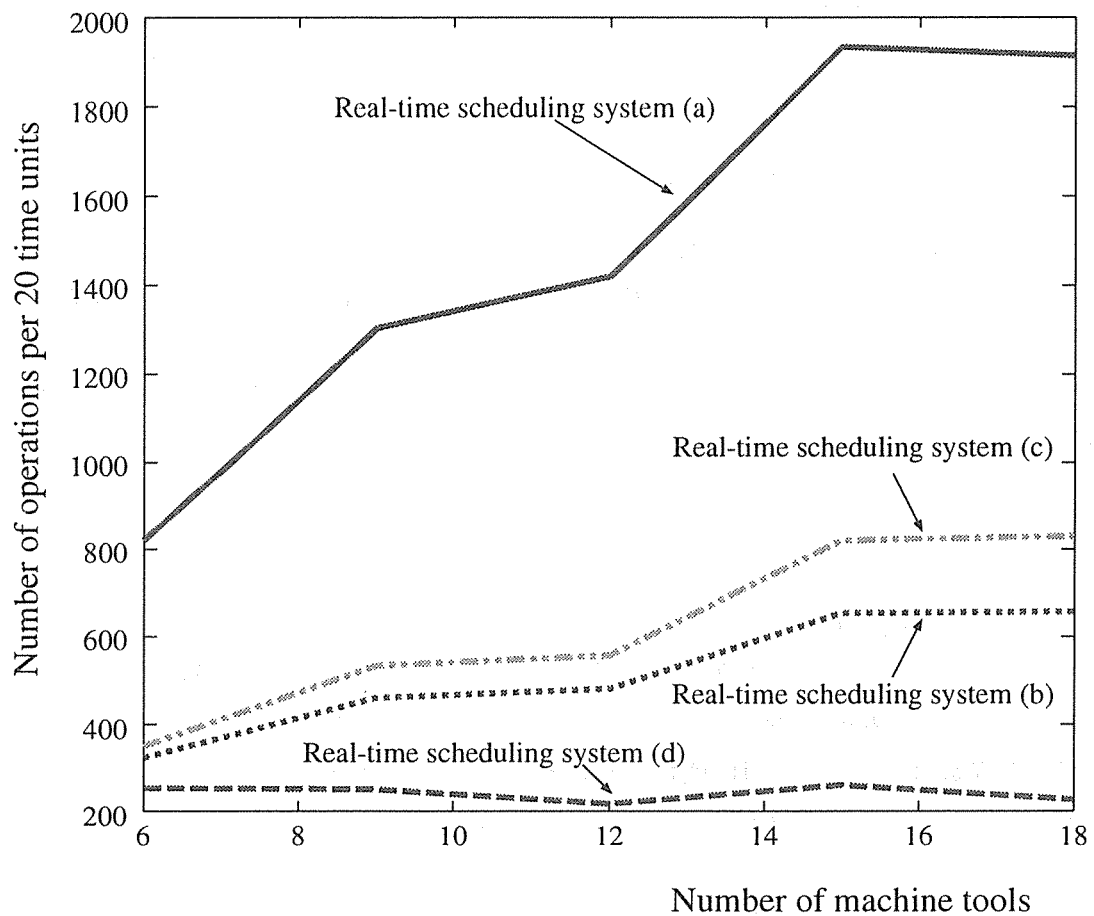


図 3: 20 タイムステップ毎の平均処理回数

## 8 まとめ

本稿では、分散型リアルタイムスケジューリングシステムの構成について述べた後、分散型リアルタイムスケジューリングシステムのための通信プロトコルについて述べた。今後は、シミュレーションを用いて性能の評価を行なうとともに、通信プロトコルの検証および改良を行なう必要がある。また、今回は通信プロトコル内にデッドロック回避、特急ジョブの処理を埋め込んだ形の実装としたが、実際規模の生産システムに分散型リアルタイムスケジューリングシステムを適用する場合、通信プロトコル内に埋め込むことが必要な機能についての考察を行なう必要がある。

さらに、完全分散型あるいは階層型など様々な分散のレベルが考えられるが、生産システムの構成、規模によりどのような形態の分散型システムとすればいいのかについての考察を行ない、分散型リアルタイムスケジューリングに適した生産管理システムのアーキテクチャを開発していく必要がある。

## 参考文献

- [1] N. Okino: A prototyping of bionic manufacturing system, *Proceedings of the ICOOMS '92*, pp. 297–302, 1992.
- [2] K. Ueda: An approach to bionic manufacturing systems based on DNC-Type information, *Proceedings of the ICOOMS '92*, pp. 303–308, 1992.
- [3] K. Baker: *Introduction to Sequencing and Scheduling*, John Wiley and Sons, New York, 1974.
- [4] C. Harmonosky and S. Robohn: Real-time scheduling in computer integrated manufacturing: Review of recent research, *International Journal of Computer Integrated Manufacturing*, Vol. 4, No. 6, pp. 331–340, 1991.
- [5] 鳩野, 田村: 分散型リアルタイム生産スケジューリング, 計測と制御, Vol. 33, No. 7, pp. 571–580, 1994.
- [6] F. Rodammer and K. J. White: A recent survey of production scheduling, *IEEE Transactions of Systems, Man, and Cybernetics*, Vol. 18, No. 6, pp. 841–851, 1988.
- [7] I. Hatono, T. Nishiyama, M. Umano, and H. Tamura: Performance evaluation of distributed real-time scheduling system using distributed production system simulator, In H. T. N. Okino and S. Fujii eds., *Advances in Production Management Systems, Proceedings of 6th IFIP TC5/WG5.7 International Conference APMS'96*, Kyoto, Japan, pp. 531–536, November 4–6, 1996.