

階層型意思決定モデルに基づく分散生産スケジューリング方式の提案

伊藤 俊明 岡崎 司 齋藤 信 高瀬 洋史 樋地 正浩 高橋 勉

日立東北ソフトウェア(株)

〒980-0014 仙台市青葉区本町 2-16-10 仙台大同生命ビル 11F

あらまし 多品種少量化が進む製造ラインでは、生産変動が頻発し計画に基づく作業実施が困難となっている。なぜなら、従来の作業指示では、予め立案した作業計画に基づいて作業指示を行い、作業実績は前回の作業計画立案時から新しい作業計画立案時までの実績を一括して反映させるため、生産変動の発生から作業指示までに時間遅れが発生するからである。

そこで、生産変動に即応した作業指示を実施可能とするため、分散スケジューリング技術の要請が高まってきている。本報告では、生産スケジューリング問題を生産活動の意思決定場面に基づきいくつかの小問題に分割し解く分散スケジューリング方式を提案する。

A Decentralized Scheduling Method based on a Hierarchical Decision Making Model

Toshiaki Ito Tsukasa Okazaki Shin Saito Hiroshi Takase
Masahiro Hiji Tsutomu Takahashi

Hitachi Tohoku Software, Ltd.

Sendai Daidoseimei Bldg., 2-16-10 Honcho, Aoba-ku, Sendai, Miyagi 980-0014, Japan

Abstract In today's manufacturing systems, it is difficult to execute orders on schedule, because of frequent production fluctuations. In order to solve the problem, it is necessary to develop the decentralized scheduling technology that is able to accommodate local changes and uncertainty.

In this paper, we propose the decentralized scheduling method base on a hierarchical decision making model. First, we suppose various cases of decision making in a manufacturing system. Next, we show models of these decision making modules using an agent model. Lastly, we describe an example of the decentralized scheduling system base on the model.

1. はじめに

製造業の分野では、生産の多品種少量化に伴い生産の変動要因が増加している。

製造業における従来の生産方式では、旬、週または日毎定期的に一定期間の作業計画を予め立案し、その計画に基づいて作業指示を行っている。製造現場では作業指示に従って作業に着手し、完了すると POP (Point of Production) などの実績収集システムを通して作業実績を入力する。入力された実績は、次の作業計画立案時に一括して計画に反映される。従って、設備の故障などによって発生した生産変動が次の作業指示に反映されるまでには、相当の時間遅れが発生する。このため、生産変動へ即応できる新しい生産方式が望まれている。

この要請に応える技術として分散スケジューリング技術が期待されている[1][2][3]。

分散スケジューリングは、スケジューリング機能の分業モデルによって水平分業モデル、垂直分業モデルに分類される[4]。また、スケジューラを分散させる対象は、設備やラインなどのリソースに分散させる方式とジョブに分散させる方式[5]とが提案されている。実現方式としては、エージェントシステムを用いた方式[6][7][8]、階層モデルの記述能力を持つベトリネットを用いた方式[9]、また競合解消のためにゲーム理論を用いた方式[10]などが提案されている。

筆者らはこれまでに、自律分散型生産管理システム構築を目的に、日程計画などの生産管理システムを構成する要素をエージェントモデルに基づきモデル化する手法を提案してきた[11][12][13]。

本稿では、このモデル化手法を用いて生産システムにおける意思決定をモデル化し、それに基づいて分散スケジューリングを実現する方式を提案する。

本稿の構成は以下の通りである。まず、2章でスケジューリング対象とする生産システムの構成について述べる。3章では、生産システムにおける意思決定とそのモデルについて述べる。4章では、分散スケジューリングシステムを実現するためのエージェントモデルについて述べる。5章では、3章で述べた階層型意思決定モデルを4章で述べたエージェントモデルを用いて表し、その動作例を示す。6章は、本稿のまとめである。

2. 生産システムの構成

まず、分散スケジューリングの対象となる生産システムの構成について述べる。図1に生産システムの構成を示す。生産システムは、ロット、リソース、材料倉庫、生産ショップ、完成品倉庫から構成される。

ロットは、注文に応じた生産量で作業単位を形成し、材料倉庫から生産ショップに投入される。ロットは品種に応じて加工工程順序が決められている。

リソースは、設備やラインなどを総称し、ロットの加工作業を行う。リソースには、同じ加工作業を行う機能を有する代替リソースが1つ以上存在することができる。これら代替リソースの集合をワークセンタと呼ぶ。よって、ロットの加工工程順序はワークセンタの並びで定義できる。

つまり、ロットはワークセンタからワークセンタへ移動し、ワークセンタ内では、1つの代替リソースで加工される。

生産システムでは、このロットの流れの中でいくつかの意思決定が存在する。次にこの意思決定について考察する。

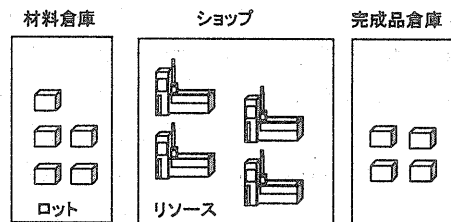


図1 生産システム

3. 階層型意思決定モデル

3.1 生産システムにおける意思決定

生産システムでの意思決定について述べる前に、生産の流れについて考える。リソースは設備と仮定する。

まず、材料倉庫からロットが第1工程に投入される。第1工程にはいくつかの代替設備が存在する。そこで、代替設備の中から実際に作業を行う設備を1つ選び出す。これを設備割り付けと呼ぶ。ロットは、割り付けられた設備で加工が行われ、加工が完了したロットは次の工程に搬送される。次工程でも同様に設備割り付け、加工が行われる。このようにして全工程を完了したロットは完成品倉庫へ搬送される。

以上が生産の流れである。次に、この生産の流れの中で行なわれる意思決定について考える。

まず、材料倉庫から第1工程へのロット投入については、材料倉庫にあるロットを、ロット毎の納期やショップの負荷などを考慮し、どのロットを、いつショップに投入するかを決定する必要がある。

次に、設備割り付けについて考える。前章で工程はワークセンタに置き換えられることを説明した。設備割り付けとは、ワークセンタ内のどの設備にロットを割り付けるかを決定することであり、各設備の負荷や能力を考慮して、ロットを割り付ける設備を決定する

必要がある。

また、実際の設備でのロットの加工について考える。実際の設備では、加工前のロットを一時溜めておく棚などのスペースを持って、ロットをバッファリングしている。なお、加工後のロットについても同様である。これは、作業毎に作業時間が異なるためであり、もしバッファがないと、前工程の作業時間が後工程のそれに対して短い場合、前工程の設備では後工程の加工完了まで作業を停滞させなければならなくなる。これでは、設備の稼働率や生産効率を低下させる原因となってしまうため、バッファが必要となる。

したがって、各設備では、バッファリングしている加工前ロットの中から加工するロットを1つ選び出すこと(これを差し立てと呼ぶ)が必要となる。通常は、到着順、即ち FIFO(First In First Out)優先ルールに従うが、仕掛り待ちの増加や製品リードタイムの増大などの生産変動が発生した場合は、SPT (Shortest Processing Time) 優先など他の優先ルールに従うことも可能である。

このように生産システムでは以下の3つの意思決定が存在する。

- (1) ロットの投入時期の決定
- (2) 設備割り付け
- (3) 作業差し立て

3.2 階層型意思決定モデル

筆者らは、各意思決定を階層的に関連づけた階層型意思決定モデルを提案する。

生産システムの構成をもう一度考えてみる。リソースは生産システムを構成する最小単位である。ワークセンタは、1つ以上のリソースから構成されている。またショップも、1つ以上のリソースから構成されているが、すべてのリソースは必ず1つ以上のワークセンタに所属するため、ショップは、ワークセンタから構成されていると言って問題ない。

よって、生産システムはショップ、ワークセンタ、リソースで階層的に構成されると言える。

次に各意思決定の対象範囲について考える。

ロット投入時期の決定は、ショップに投入されるすべてのロットを対象とする。設備割り付けは、1つのワークセンタに搬送され設備割り付けを待つロットを対象とし、他のワークセンタとは独立に意思決定できる。作業差し立ては、1つのリソースのバッファで加工を待つロットを対象とし、同様に他のリソースとは独立に意思決定できる。

このように生産システムにおける意思決定は、ショップ、ワークセンタ、リソースの順で階層的に決定さ

れ、また同じ層では、意思決定が互いに独立している。

このことから、図2に示すような階層型意思決定モデルを提案する。

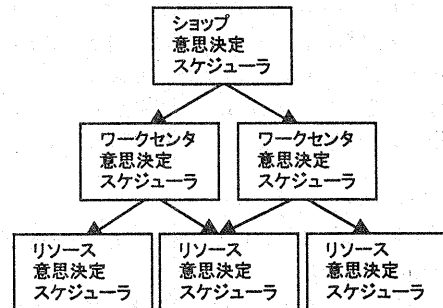


図2 階層型意思決定モデル

このモデルに基づき、各意思決定に対応するスケジューラを分散させることにより、上下の階層間では同期、非同期的に、同一階層では非同期で同時並行的に随時、スケジューリングを実行することが可能である。また、スケジューリング問題の規模が小さくなるため、応答性もよくリアルタイムでの運用が期待できる。

提案する方式では、分散するスケジューラをエージェントモデルに基づく手法でモデル化する。そこで、スケジューラのモデル化について述べる前に、ここで用いるエージェントモデルについて述べる。

4. エージェントモデル

筆者らは、これまでに自律分散型生産管理システム構築を目的として、エージェントモデルに基づき生産システム構成要素をモデル化する手法を提案してきた[11][12][13]。提案するエージェントは、属性、メソッド、インタプリタから構成される実行主体である。エージェントは、他のエージェントとメッセージの送受信により互いに情報を交換することができる。属性は、属性を識別するための名称とその値(属性値)を持ち、そのエージェントの内部状態を保持している。

エージェントは、自身のメソッドまたは他のエージェントからメッセージとして受けとった処理手続きをインタプリタで解釈実行することによって属性値を変更することができる。

属性として特定の属性名を、また特定のメソッドを与えることにより、特定の目的または機能を持ったエージェントを定義することができる。

このようにして、移動エージェントとサービスエージェントを定義する。

4.1 移動エージェント

移動エージェントは、後述するいくつかのサービス

エージェントの間を、サービスを受けながら移動する。

移動エージェントは、属性として、AgentID, To, From, MigrationList を持つ。AgentID は、エージェントを識別するための識別コードである。To は、このエージェントの移動先を表す名称であり、また、From は、このエージェントを生成したサービスエージェントの名称である。MigrationList は、移動エージェントが利用するサービスエージェントの名称のリストである。

4.2 サービスエージェント

サービスエージェントは、前章で述べた各層での意思決定を提供する。

サービスエージェントは、属性として、AgentID, InputBuffer, OutputBuffer を持つ。AgentID は、そのエージェントを識別するための名称であり、すなわちサービス名称である。InputBuffer は、サービスエージェントに移動してきた移動エージェントの AgentID を保持する。OutputBuffer は、サービスを完了した移動エージェントを次の移動先に実際に移動するまで、その AgentID を保持する。これらの属性の他、対応する意思決定サービス毎に任意の属性をもつこともできる。またサービスエージェントは、メソッドとして意思決定のためのメソッドを持つ。

5. 分散スケジューリング方式

前章のモデル化手法を用い、ロットおよび3章で述べた各意思決定スケジューラのエージェントモデルについて述べる。

ロットは、移動エージェントでモデル化できる。MigrationList 属性は、ロットの加工工程順序を保持する。加工工程順序は、後述するワークセンタエージェントの AgentID を加工工程順にリストで定義される。

5.1 ショップエージェント

図3にショップエージェントの構成を示す。ショップエージェントは、InputBuffer, OutputBuffer にそれぞれ次のような属性値を持つ。

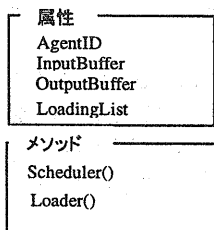


図3 ショップエージェントの構成

InputBuffer の属性値は、材料倉庫のすべてのロッ

トに対応する移動エージェントの AgentID の一覧である。OutputBuffer の属性値は、完成品倉庫のすべてのロット(完成品)に対応する移動エージェントの AgentID の一覧である。

ショップエージェントは、他の属性として LoadingList を持つ。LoadingList の属性値は、InputBuffer に保持される AgentID を持つロットの投入順序を保持する。投入順序は、(AgentID, 投入時刻)の2個組の一覧である。

またショップエージェントの意思決定メソッドは、Scheduler() である。Scheduler() は、InputBuffer に保持される AgentID を持つロットの投入順、投入時刻を決定する。Scheduling() は、処理結果を LoadingList 属性に格納する。さらにショップエージェントは、Loader() をメソッドに持つ。Loader() は、LoadingList 属性に保持された先頭の組の投入時刻にその組の AgentID を持つ移動エージェントの To 属性を MigrationList の先頭の AgentID で更新し移動させる。

5.2 ワークセンタエージェント

図4にワークセンタエージェントの構成を示す。

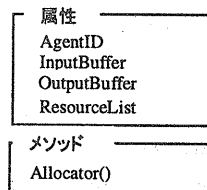


図4 ワークセンタエージェントの構成

ワークセンタエージェントは、InputBuffer, OutputBuffer にそれぞれ次のような属性値を持つ。InputBuffer の属性値は、そのワークセンタの設備割り付けを待っているロットに対応する移動エージェントの AgentID の一覧である。OutputBuffer の属性値は、そのワークセンタのサービスが完了し次ワークセンタの設備割り付けを待つロットに対応する移動エージェントの AgentID の一覧である。

ワークセンタエージェントは、他の属性として ResourceList を持つ。ResourceList は、このワークセンタに所属するリソースエージェントの AgentID の一覧である。

またワークセンタエージェントの意思決定メソッドは、Allocator() である。Allocator() は、InputBuffer に保持される AgentID を持つすべてのロットを ResourceList に保持される AgentID を持つリソースエージェントに割り付ける。あるリソースエージェントに割り付けられた移動エージェントは、その To 属性の値を割り付けられたリソースエージェントの

AgentID に書き換えそのリソースエージェントに移動する。

5.3 リソースエージェント

図5にリソースエージェントの構成を示す。

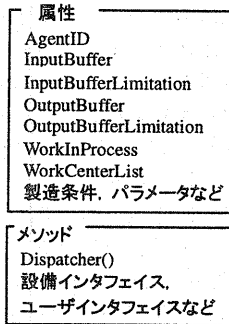


図5 リソースエージェントの構成

リソースエージェントは、InputBuffer, OutputBuffer にそれぞれ次のような属性値を持つ。InputBuffer の属性値は、そのリソースでの加工を待っているロットに対応する移動エージェントのAgentIDの一覧である。OutputBuffer の属性値は、そのリソースでの加工が完了し次ワークセンタの設備割り付けを待つロットに対応する移動エージェントのAgentIDの一覧である。

リソースエージェントは、他の属性としてInputBufferLimitation, OutputBufferLimitation, WorkInProgress, WorkcenterListを持つ。InputBufferLimitation, OutputBufferLimitation は、それぞれInputBuffer, OutputBuffer の物理的な最大容量である。WorkInProgress は、現在加工中のロットに対応する移動エージェントのAgentIDである。WorkCenterList は、このリソースが所属するワークセンタに対応するワークセンタエージェントのAgentIDの一覧である。他に、対応するリソース毎に製造条件やパラメータなどを保持する属性を持つことが可能である。

またリソースエージェントの意思決定メソッドは、Dispatcher()である。Dispatcher()は、InputBuffer に保持されるAgentIDを持つロットの中から1つのロットを取り出す機能を持つ。その取り出し方は、任意のディスパッチングルールに従う。例えば、前出のFIFO 優先ルールの場合、InputBuffer に保持されるAgentID を持つ各移動エージェントの到着時刻の最も早いものを取り出す。

リソースエージェントは他のメソッドとして、設備とのインタフェイス、グラフィカルユーザインタフェイスなどを持つ。これらのメソッドは、自動機であれば、設備へのロットローディング指示、アンロード信

号の受信に、また作業者に対しては、Dispatcher()メソッドの結果に基づく作業指示の表示、作業終了の入力などを実現する。

5.4 動作例

提案するエージェントモデルを用いた分散スケジューリング方式の動作を設備4台の生産ジョブ、工程数が2工程の例を用いて説明する。4台の設備をそれぞれ、M1, M2, M3, M4とする。製品の工程を(WC1, WC2)とする。

ワークセンタ WC1 を {M1, M2}, WC2 を {M3, M4} とすると、ワークセンタエージェント WC1, WC2 のResourceList 属性は、それぞれ(M1, M2), (M3, M4)である。またリソースエージェント M1, M2, M3, M4 のWorkCenterList は、それぞれ WC1, WC1, WC2, WC2 である。

ジョブをSとする。材料倉庫には、ロットL1, L2, L3, L4が保持されている。移動エージェントL1, L2, L3, L4のMigrationList 属性は、(WC1, WC2)であるとする。このとき、ジョブエージェントSのInputBuffer は、(L1, L2, L3, L4)である。

ジョブエージェントは、Scheduling()メソッドでInputBuffer の各ロットに対し投入順序を決定する。ここでは簡単のために、優先ルールを入力順とする。LoadingList 属性は、((L1, t1), (L2, t2), (L3, t3), (L4, t4))とする。但し、 $t1 \leq t2 \leq t3 \leq t4$ である。

時刻t1に、ジョブエージェントSのLoader()メソッドによって、移動エージェントL1のTo属性の属性値がWC1で更新され、L1はワークセンタエージェントWC1に移動する。ジョブエージェントSでは、InputBuffer の属性値が(L2, L3, L4)で更新される。以下同様に、L2, L3, L4が時刻t2, t3, t4にWC1に移動する。

ワークセンタ WC1 では、ロットL1が到着するとInputBuffer にL1が登録される。

今、リソースエージェントM1のInputBuffer は空であるため、リソースエージェントM1は、WorkCenterList に登録されているワークセンタエージェントに割り付け要求メッセージを送る。

ワークセンタ WC1 では、リソースエージェントM1からの割り付け要求メッセージを受信すると、Allocator()メソッドによってInputBuffer に登録されているロットの中から1つのロットを選び出す。この場合、L1が選ばれる。ロットエージェントL1は、To属性を割り付け要求メッセージの送信者であるM1で更新され、リソースエージェントM1に移動する。ワークセンタエージェントWC1のInputBuffer からL1

が削除される。

リソースエージェント M1 では、ロット L1 が到着すると InputBuffer に L1 が登録される。次に、WorkInProgress 属性が空であるため、InputBuffer に登録されているすべてのロットから Dispatcher() メソッドにより作業差し立てが行われる。差し立てられたロットは、WorkInProgress に登録され、一方、InputBuffer から削除される。また、設備とのインタフェースメソッドやグラフィカルユーザインタフェースメソッドにより、設備へのロットローディング指示または作業着手指示が出される。

ロット L1 の加工が完了し、設備とのインタフェースメソッドやグラフィカルユーザインタフェースメソッドにより、加工完了を通知されると、リソースエージェント M1 は、OutputBuffer に L1 を登録する。WorkInProgress は空に更新される。また、リソースエージェント M1 からワークセンタエージェント WC1 に加工完了のメッセージが送信される。

ワークセンタ WC1 は、ロット L1 の加工完了メッセージを受信すると、OutputBuffer に L1 を登録し、ロット L1 の MigrationList に従い次のワークセンタ WC2 にロット L1 を移動させる。

ロットはこのようにしてワークセンタエージェント間を移動し、またワークセンタエージェントの中で、1つのリソースエージェントのサービスを受ける。

この一連の動作がすべてのロット、ショップエージェント、ワークセンタエージェント、リソースエージェントで同時並行的に行われる。

6. まとめ

製造業では、製品ライフサイクルの短命化やユーザーの多様化によって多品種少量化が進んでいる。このような状況で、新しい生産パラダイム、生産管理方式が模索されている。分散スケジューリング技術は、これらの基本技術として期待が大きい。

本稿では、このような要請に応えるために、階層型意思決定モデルに基づく分散スケジューリング方式を提案した。生産システムで発生する意思決定は、ロット投入時期の決定、設備割り付け、作業差し立てであり、これらは生産システムの構成と対応し階層化されていることを示した。提案する方式では、これらの各階層毎の意思決定をエージェントモデルに基づく手法を用いてモデル化し、生産システム上に分散した各サービスエージェントが対応する意思決定を同時並行的に随時行うことにより、分散スケジューリングを実現した。これにより、意思決定が必要な状況毎に随時、適切な意思決定が行われ、生産状況の変化に追従した

生産を可能とする。

参考文献

- [1] 藤本英雄, "生産スケジューリングの動向", 計測と制御, Vol. 33, No. 7, pp. 533-540, 1994
- [2] 黒田充, "生産スケジューリング研究の課題と展望", 生産スケジューリング・シンポジウム'94 講演論文集, pp. 1-13, 1994
- [3] 西岡靖之, "状況対応型スケジューリング問題へのアプローチ", 計測と制御, Vol. 33, No. 7, pp. 571-574, 1994
- [4] 宮下和雄, "分散環境におけるスケジューリング", 生産スケジューリング・シンポジウム'97 講演論文集, pp. 203-208, 1997
- [5] 大井秀人, 沖野教郎, "自律分散スケジューリングにおける協調アルゴリズム", 生産スケジューリング・シンポジウム'95 講演論文集, pp. 291-296, 1995
- [6] 鳩野逸生, 田村坦之, "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. 575-580, 1994
- [7] Khalid K. Kouiss, H. Pierreval, N. Mebarki, "Using multiagent architecture in FMS for dynamic scheduling", Journal of Intelligent Manufacturing, 4, pp. 41-47, 1997
- [8] H. P. Wiendahl, V. Ahrens, "Agent-Based Control of SelfOrganized Production Systems", Annals of the CIRP, Vol. 46/1/1997
- [9] 築山誠, 森一之, 福田豊生, "大規模生産システムの階層分散型スケジューリング", 計測と制御, Vol. 33, No. 7, pp. 581-584, 1994
- [10] 西岡靖之, "自律分散型の生産システムに対するゲーム理論の適用について", 生産スケジューリング・シンポジウム'97 講演論文集, pp. 223-224, 1997
- [11] 樋地正浩, 伊藤俊明, 岡崎司, 高橋勉, "分散協調 CIM のための自律オブジェクトのモデルとその実装", 情報処理学会第 51 回全国大会講演論文集 (I), pp. 335-336, 1995
- [12] 伊藤俊明, 樋地正浩, 岡崎司, 高橋勉, "自律オブジェクトによる動的生産スケジューリング", 情報処理学会第 51 回全国大会講演論文集 (I), pp. 333-334, 1995
- [13] 樋地正浩, 手塚大, "エージェントモデルに基づく生産システム要素のモデル化", 生産スケジューリング・シンポジウム'98 講演論文集, pp. 137-142, 1998