

| | |
|--------------|---|
| Title | フレキシブル生産におけるインテリジェント・スケジューリングに関する研究 |
| Author(s) | 鳩野, 逸生 |
| Citation | 大阪大学, 1994, 博士論文 |
| Version Type | VoR |
| URL | https://doi.org/10.11501/3097497 |
| rights | |
| Note | |

Osaka University Knowledge Archive : OUKA

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

Osaka University

フレキシブル生産における
インテリジェント・スケジューリング
に関する研究

平成 6 年 3 月

鳩 野 逸 生

フレキシブル生産における
インテリジェント・スケジューリング
に関する研究

平成 6 年 3 月

鳩 野 逸 生

目次

| | | |
|-------|----------------------------|----|
| 第1章 | 序論 | 1 |
| 1.1 | 多品種少量生産とスケジューリング問題 | 1 |
| 1.1.1 | 多品種少量生産とフレキシブル生産の概要 | 1 |
| 1.1.2 | 生産スケジューリング問題 | 3 |
| 1.2 | 本論文の構成 | 5 |
| 第2章 | ルールベースを用いたフレキシブル生産スケジューリング | 7 |
| 2.1 | まえがき | 7 |
| 2.2 | 時間ペトリネットによるFMSのモデル化 | 8 |
| 2.3 | スケジューリングへのルールの適用 | 11 |
| 2.4 | 試作システムの概要 | 13 |
| 2.4.1 | FMSモデリングシステム | 14 |
| 2.4.2 | FMSシミュレータ | 15 |
| 2.4.3 | ルールインタプリタ | 15 |
| 2.5 | 総処理終了時間短縮のためのルールベース | 15 |
| 2.5.1 | ルール構築のためのヒューリスティクス | 15 |
| 2.5.2 | 数値例 | 18 |
| 2.6 | ルール修正機能を持ったルールベーススケジューリング | 20 |
| 2.6.1 | ジャストインタイムのためのルールベース | 20 |
| 2.6.2 | ルールの適応 | 23 |
| 2.6.3 | 数値例 | 24 |
| 2.7 | むすび | 25 |
| 第3章 | ファジィ推論を用いたフレキシブル生産スケジューリング | 26 |
| 3.1 | まえがき | 26 |
| 3.2 | 多様なスケジューリング目的の実現のための知識ベース | 26 |
| 3.2.1 | 知識ベースの構造と柔軟さ | 26 |
| 3.2.2 | システム動作の概要 | 28 |
| 3.3 | スケジュール評価モジュール | 29 |
| 3.3.1 | 部分スケジュール評価のための評価項目 | 29 |

| | | |
|------------|--|-----------|
| 3.3.2 | スケジュール評価のためのファジィ推論 | 32 |
| 3.4 | スケジューリング方針決定モジュールにおける推論 | 39 |
| 3.4.1 | OR 演算子における親に対する子の影響度 | 39 |
| 3.4.2 | AND 演算子における親に対する子の影響度 | 39 |
| 3.4.3 | 量限定子における親に対する子の影響度 | 40 |
| 3.4.4 | 先祖のノードに対する影響度 | 40 |
| 3.5 | ディスパッチングモジュールにおける推論 | 41 |
| 3.6 | システム構成 | 42 |
| 3.7 | 数値例 | 43 |
| 3.8 | むすび | 47 |
| 第4章 | フィルター付きビーム探索法におけるヒューリスティクスの自動合成 | 49 |
| 4.1 | まえがき | 49 |
| 4.2 | 搬送車スケジューリングのモデル化 | 50 |
| 4.3 | フィルター付きビーム探索法 | 51 |
| 4.3.1 | スケジュール探索のための優先度評価関数 | 51 |
| 4.3.2 | フィルター付きビーム探索法のアルゴリズム | 53 |
| 4.3.3 | フィルター付きビーム探索法の問題点 | 55 |
| 4.4 | ローカル優先度評価関数の合成 | 55 |
| 4.4.1 | 入力データ分類モジュール | 56 |
| 4.4.2 | 評価関数合成モジュール | 56 |
| 4.4.3 | 優先度評価関数の修正 | 58 |
| 4.5 | 搬送車スケジューリング問題への適用 | 58 |
| 4.5.1 | スケジューリング要求 | 58 |
| 4.5.2 | 分枝限定法との比較 | 61 |
| 4.5.3 | スケジューリング要求と優先度評価関数の関係 | 62 |
| 4.5.4 | フィルター幅とビーム幅に関する考察 | 62 |
| 4.6 | むすび | 66 |

| | | |
|------------|---|------------|
| 第5章 | ルールベースを用いたフレキシブル生産リアルタイムスケジューリング | 67 |
| 5.1 | まえがき | 67 |
| 5.2 | リアルタイム・ルールベース・スケジューリングシステム | 67 |
| 5.2.1 | リアルタイムスケジューリングのためのルールベース | 68 |
| 5.2.2 | ルール構築のためのヒューリスティクス | 68 |
| 5.3 | 階層的確率ペトリネットによる生産システムのモデル化 | 69 |
| 5.3.1 | 階層的確率ペトリネット | 69 |
| 5.3.2 | モデル化における仮定 | 69 |
| 5.4 | 数値例によるルールベースの評価 | 73 |
| 5.4.1 | 数値例の条件 | 73 |
| 5.4.2 | 結果と考察 | 75 |
| 5.5 | むすび | 75 |
| 第6章 | 分散型リアルタイムスケジューリング | 77 |
| 6.1 | まえがき | 77 |
| 6.2 | 分散型スケジューリング | 78 |
| 6.2.1 | 生産システムのモデル化 | 78 |
| 6.2.2 | 分散型スケジューリングシステム | 80 |
| 6.2.3 | システム構成 | 87 |
| 6.3 | スケジューリングに用いる評価構造 | 88 |
| 6.3.1 | 競合解決と評価方法 | 89 |
| 6.3.2 | 各評価項目の定式化 | 93 |
| 6.4 | 数値例 | 96 |
| 6.4.1 | 半導体生産プロセスの概要 | 96 |
| 6.4.2 | モデル化 | 97 |
| 6.4.3 | 本例で用いる生産環境 | 98 |
| 6.4.4 | 出力結果 | 98 |
| 6.5 | むすび | 105 |
| 第7章 | 結論 | 107 |
| 7.1 | オフラインスケジューリング | 107 |
| 7.2 | リアルタイムスケジューリング | 108 |
| 7.3 | 今後の課題 | 109 |
| 謝辞 | | 110 |

| | |
|--------------|-----|
| 参考文献 | 111 |
| 図表リスト | 118 |
| 関連発表論文 | 122 |

第1章 序論

1.1 多品種少量生産とスケジューリング問題

本節では、多品種少量生産およびその問題点と対策のひとつであるフレキシブル生産について述べ、さらにフレキシブル生産を行なう場合に特に重要な問題であるスケジューリング問題について述べる。

1.1.1 多品種少量生産とフレキシブル生産の概要

多品種少量生産とは、ある期間に生産の対象となる品物の種類(仕様、形状、寸法、生産工程、色彩など)が多様化していて、それぞれの生産数量が少ない生産形態である。多品種少量生産を取り扱う工場のことを“ジョブショップ (job shop)”といい、生産の対象となる製品やその処理を“ジョブ”という。機械で処理される加工順序がすべてのジョブについて同じである工場を特に“フローショップ”といい、これはジョブショップの特別な場合と考えることができる。

多品種少量生産の様式は、多くは注文生産形態であるが、最近では見込み生産形態にも見られ、個別生産ないし間欠(ロットまたはバッチ)生産の様式をとる。さらに連続生産でも、仕様の異なるジョブが個別・連続的に流れ作業で作られる混流生産の様式で行なわれることもある。

多品種少量生産の特徴は、次の通りである [1]。

- (1) 生産品目の多様性：生産の対象となる製品の種類が多く、生産数量や納期が多様である。
- (2) 生産工程の多様性：生産の変換過程、すなわち素材から製品への変換の手順が多様で、生産工程が個々のジョブによって異なり、生産工場内でジョブが複雑に交錯する。
- (3) 生産能力の複雑性：多様化製品の需要量次第で生産設備に過不足が生じ、残業・交替制など稼働時間の延長をもたらす、時に人海戦術に頼らなければならないことがある。
- (4) 環境条件の不確実性：受注品の仕様・数量・納期の変更、特急ジョブの発生、外部購入品(素材、部品、外注品)の納入遅れなどが起こりやすい。
- (5) 生産の工程・スケジューリングの困難性：受注品の仕様変更に起因する設計変更、生産工程の計画や複雑な物の流れ、および上述の環境条件の不確実性のために精度の高い生産情報が欠如し、最適スケジューリングが困難である。

(6) 生産の実施と統制の動態性：生産の工程・スケジュールの最適案が不確実なため、生産工場における作業実施が複雑多岐になり、設備故障、作業員欠勤、習熟効果の欠如、不良品の生起などが多発しやすく、したがって綿密な計画をたてるよりもむしろ、経験と勘に頼る現場中心の場当たりの・成行きの管理に陥りやすく、合理的な生産管理が困難である。

このように、多品種少量生産は、いわゆる大量生産よりも、コスト・能率の面で劣っているが、近年の消費者のニーズの多様化のため、この生産様式をとらざるを得なくなってきた。以下に、多品種少量生産に対処するための手法のいくつかについて簡単に述べる。

A. グループ・テクノロジー (GT)

グループテクノロジー (group technology, GT) [1] は“部品群加工”とも呼ばれる。多種類の部品・製品を形状・寸法・加工法などが類似しているものを“グループ”として集約し、この部品のグループをロットとみなすことにより、設計を標準化・合理化し、各グループに適切な生産設備と治工具を割り当て、段取り時間・工程間運搬・加工待ちを減少させ、無秩序に個別的に設計・生産・管理する場合より大きいロット数で、大量生産方式に近い効果を与え、生産性を向上させようとする多品種少量生産技法である。

最近では、GTは生産の計画やスケジューリング、管理にも適用されるようになってきた。これによる最も大きな利点は、段取り時間が大幅に短縮でき、フローショップ・タイプのスケジューリングを考えれば良いので、スケジューリングが楽になる点である。これは、“グループ・スケジューリング”と呼ばれる。

B. ジャスト・イン・タイム (JIT) 生産

ジャスト・イン・タイム (just in time, JIT) 生産 [1] は、“必要な時に必要なものを必要な量だけ生産・供給する”ようにした生産方式であり、作業指示と現場管理に“かんばん”を用いる。そのため、“かんばん方式”とも呼ばれる。また、生産現場で仕掛在庫削減ないし無在庫方式を目指すので、“ストックレス生産”あるいは“ノンストック生産”ともいう。

C. フレキシブル生産

フレキシブル生産 [2] では、柔軟性のある生産設備 (高精度の汎用工作機械) を用い、さらに、平均需要量をまかなうための基本設備と変動需要量に対処するための適量の補助設備を設け、特殊工具や自動工具の利用をはかり、また外注製作も考慮する。

設備面だけでなく、使用する労働力も、多様な作業を能率的・効果的に遂行する高度な熟練労働力を主体にし、高度な計画性を有する頭脳的管理技術力を駆使し、製品仕様の迅速な決定、原

材料の適切な取得, 納期を重視する生産の計画と管理を迅速かつ適切に行ない, 多品種少量生産のもつ柔軟性に対応しようとしている。

本論文で扱う生産形態は, このフレキシブル生産である。

1.1.2 生産スケジューリング問題

生産スケジューリング問題 [3,4] とは, 実際の生産加工において, どの工作機械 (生産設備, ワークステーション) で, いつ, どの作業員が, どの様な作業を行なうのかという, 個々の要素作業を実行するための具体的で詳細な実施プログラムを作成する問題のことである。具体的には, n 個の仕事 (ジョブ) を m 台の工作機械を用いて加工する時, 各ジョブの技術的加工順序と各機械による加工時間が既知のもとで, ある目的関数 (総所要時間, 納期遅れなど) が最小となるように各ジョブを加工する機械の順序を求める (ジョブを機械に割り付ける) 問題である。

スケジューリング問題は, つぎの 4 種類の情報を用いて表現することができる [3].

- (1) 処理すべき仕事とその作業
- (2) 工場に存在する機械の種類と台数
- (3) 割り付け方を制約する条件
- (4) スケジュールを評価する判定基準

(3) の制約条件は, さらに, 製品とそれを処理できる資源との関係や, 注文の納期, 中間仕掛品の置き場の容量や置いておく時間の制限など, 必ず守らなければならない制約 (硬い制約) と, 同時段取り替え多発の禁止, 1 日/1 月あたりの残業時間の制限, 稼働率の向上や平準化などの, 守らないと実行が不可能というわけではないが, できるだけ守ってほしい制約 (軟らかい制約) に分類することができる。また, (4) の判定基準では, トレードオフの関係にあるものが多いため, スケジューリング問題を難しくしている一因となっている。また, スケジューリング問題はスケジューリングを行なう時期と期間によってオフラインスケジューリングとリアルタイムスケジューリングに大別される。オフラインスケジューリングは, 1 日, 1 週間, 1 ヶ月など比較的長期のスケジュールを, 生産を開始する前に求める方法であり, 現在普通にスケジューリングといえばオフラインスケジューリングを意味することがほとんどである。一方, リアルタイムスケジューリングはオンラインスケジューリングとも呼ばれ, 生産を行ないつつ次に行なうべき工程のスケジューリングなど非常に短い範囲のスケジューリングをダイナミックに逐次的に実行する方法である。以下にオフラインスケジューリングおよびリアルタイムスケジューリングの代表的な解法よびそれらの問題点について述べる。

A. オフラインスケジューリングの解法

オフラインスケジューリング問題の解法は、最適解法と近似解法に大別される。最適解法では、問題を組合せ最適化問題や混合整数計画法として定式化し、分枝限定法 [3,4] 辞書探索法 [4,5] および A* アルゴリズム [6-8] のような OR 手法を用いて解くという方法がある。しかし、ジョブショップスケジューリングの場合、 n 個のジョブを m 台の機械を用いて処理するには、 $(m!)^n$ 通りの方法があり、原理的には最適解を有限回の計算で求めることができるが、小規模な問題でも膨大な計算時間が必要であり、実用的な問題を解くことは事実上不可能である。そこで、実際には準最適解を求め、それをスケジューリング問題に対する解とする近似解法 [9] を用いるのが一般的である。スケジューリングの近似解法としては、あるディスパッチングルールを用いて縦型探索を行なう方法 [3]、ビーム探索 [10] などのヒューリスティックアルゴリズム、スケジューリング・シミュレーション法 [4] などがあげられる。近似解法の長所としては比較的短い計算時間でスケジュールが得られるが、スケジューリング目的に合致した解が得られるか否かはそれぞれの問題とアルゴリズムの組合せに依存し、どの組合せを用いれば良いのかを事前に知ることは一般に困難である。

B. リアルタイムスケジューリングの解法

オフラインスケジューリングでは、その性格上、機械の故障や特急ジョブの発生など、予測不可能な事態に対処することは難しい。従って、予測不可能な事態が多発するフレキシブル生産においては、オフラインスケジューリングによる方法は実際の実行に耐えるものとはなりにくい。

オフラインスケジューリングに対して、リアルタイムスケジューリング (オンラインスケジューリング) [11] は、実際に生産を行ないつつ、リアルタイムに工程の順序づけや工作機械の割り当てなどを行なうものである。この方法は、スケジュールを生成するための決定を行なうのに必要な計算量に制限があるため、オフラインスケジューリングの研究に比べて進んでいないのが現状である。しかし、大規模な生産を短期間でしかも著しく変化する生産環境のもとで行なうことが必然となってくるにつれ、実用的なリアルタイムスケジューリング手法の要求が切実なものとなっている。リアルタイムスケジューリング研究の例としては、スケジューリング対象を搬送車スケジューリングまたはジョブのルーティング等に限定した上で、小規模な準最適化問題をリアルタイムで解くことによりスケジューリングを行なったシステム [12] や、対象を離散形状状態方程式でモデル化することにより最適制御理論を適用してスケジューリングを行なった例 [13] などがあげられる。また、文献 [14,15] においてもリアルタイムスケジューリングを行なうためのヒューリスティクスの研究が行なわれている。

これらの方法は、いずれも計算を単一のプロセス上で実行するため、問題の規模にともなって計算量が増加し、大規模な問題を実用的に解くことは困難である。

1.2 本論文の構成

以上に述べた背景のもとで、本論文では実用的かつ従来の方法よりフレキシブルでインテリジェントなスケジューリング手法の開発を目指して行なった研究成果をまとめる。

まず第2章～第4章においては、オフライン・スケジューリングについて議論する。

第2章では、シミュレーションを用いたスケジューリングにおいて、時々刻々の状況に適した優先規則を選択するための方略を if ~ then ~ 形式のルールベースとして構成し、そのルールベースを用いて動的に優先規則を選択しながらシミュレーションを行なうことにより、より効率的なスケジュールを得ることを試みる。また、ジャスト・イン・タイムを目標としたルールベース中のパラメータを自動的にスケジューリング対象の FMS (flexible manufacturing system) に適応させることによりルールベース構築の簡単化を試みる。

第3章では、シミュレーションを用いたスケジューリングにおいて、競合した部品の選択のための知識ベースを、現在までに生成されているスケジュールを評価するスケジュール評価モジュール、今後のスケジューリング方針を決定するスケジューリング方針決定モジュールおよび競合した部品の選択を行なうディスパッチングモジュールに分割し、ファジイ述語を含むファジイ・ルールによって各モジュールにおける知識を記述して知識のモジュール化を行なう。さらにこの中で最もユーザが直感的に理解し易いと考えられるスケジュール評価モジュール中の知識に含まれるルールのメンバーシップ関数を操作することにより、多様なスケジューリング目的に対応可能なスケジューリングシステムを構成することを試みる。

第4章では、ヒューリスティック探索法的一种であるビーム探索法を改良したフィルター付きビーム探索法におけるヒューリスティクスをシステム自身が自動的に獲得し、それを有効に利用する方法を提案し、搬送車スケジューリング問題に適用することにより有効性を確かめる。フィルター付きビーム探索法は、ビーム探索法において計算時間がかかるグローバルな情報に基づく探索の数を、計算時間が短くて済むローカルな情報に基づいて絞り込むことにより探索空間を狭める方法であるが、このローカルな情報を領域専門家の知識によらず、自動的に構成することを試みる。さらに、分枝限定法によって得られた最適解と比較することにより本手法によって得られたヒューリスティックな知識の精度の評価を行なう。

つづいて第5章および第6章においては、リアルタイムスケジューリングについて議論する。

第5章では、不確実性の多い生産システムに対するスケジューリングのアプローチとして、部品のディスパッチングやルーティングにおける部品選択にルールベースによる意思決定を導入することにより、生産を行ないながらスケジュールを作成するリアルタイムスケジューリングを、ある生産目標に対して効率的に行なうことを試みる。また、ルールベースの作成、評価にあたってはシミュレーションを行なう必要がある。本章では工作機械の故障、修理時間および加工時間のばらつきという不確実な事象を含むシステムのモデリングに確率ペトリネットを用いる。

第6章では、現在中央集中型であるスケジューリングシステムを分散環境のもとで実現することにより、より柔軟で信頼性の高いスケジューリングシステムを構成することを試みる。各ジョブそれぞれに与えたスケジューリング要求に応じて、各機械に平等に配置したスケジューリングモジュールが他の機械の状況を参照しながらスケジューリングを行なうことにより、多様なスケジューリング要求に対応でき、しかも故障などの不慮の事態に対しても寛容なシステムを実現することができる。さらに、具体的な実例として、半導体生産プロセスを対象としたスケジューリング問題を取りあげ、本章において提案する分散型リアルタイムスケジューリングの有用性を検証する。

第7章では、本論文の総括を行ない、得られた結論と今後の課題をまとめる。

第2章 ルールベースを用いたフレキシブル 生産スケジューリング

2.1 まえがき

多品種少量生産または多品種中量生産を対象としたフレキシブル生産システム (flexible manufacturing system, FMS) [1] は近年の需要の多様化に対応できる生産設備として注目されている。FMS を効率的に運用するには、ハードウェアだけでなく、生産計画やハードウェアを管理・制御するソフトウェアが必要である [2]。この中の生産計画に含まれる重要な問題の一つに生産スケジューリングがある。スケジューリング問題は組み合わせ最適化問題として定式化できるが、問題が大きくなると計算量が増大し容易に解くことが困難になる。このため近似解法 [3] が用いられることが多い。

一方、スケジューリングを行なう有力な方法としてスケジューリング・シミュレーション法 [1,4,5] がある。これはスケジューリング対象生産設備のシミュレータにより生産状況のシミュレーションを行ない、得られた結果のうち良いものをスケジュールとして用いるものである。シミュレーションにおいて、工作機械または搬送機械に対して次に作業可能なジョブが複数存在する場合や次工程の作業ができる機械または搬送機械が複数台存在する場合がある。このような場合、優先規則あるいはディスパッチングルールと呼ばれる規則を用いて次に作業を行なうべきジョブを選択したり (ディスパッチングという)、次工程を行なう工作機械あるいは搬送機械を選択する (ルーティングという) 必要がある。従って、効率的なスケジュールを得るには、取扱っている問題に適した優先規則を適用する必要がある。しかし、一般には事前に適した優先規則を特定化することは困難である。そのため、いくつかの優先規則についてシミュレーションを行ない、それらの結果のうち最も良いものをスケジュールとして採用することが行なわれる。

シミュレーションによるスケジューリングは必ずしも最適なものになっているという保証はないが、比較的短い時間で結果を得ることができる。しかし、優先規則を固定してシミュレーションを行なった場合、その優先規則が適していない状況にも適用してしまうため、必ずしも効率的なスケジュールを得ることができるとは限らない。そこで本章では、時々刻々の状況に適した優先規則を選択するための方略を if ~ then ~ 形式のルールベースとして構成し、そのルールベースを用いて動的に優先規則を選択しながらシミュレーションを行なうことにより、より効率的

なスケジュールを得ることを試みる [6-10]. なお, FMS のモデル化を行なうために本論文では FMS を離散事象システムとして捉え, 構造が簡単でモデル化能力が高く, ある程度の理論解析も可能であるという利点を持ったペトリネットに時間の概念を導入した時間ペトリネット [11-13] を用いる. また, ジャストイン・タイムを目標としたルールベース中のパラメータを自動的にスケジューリング対象の FMS に適応させることによりルールベース構築の単純化を試みた研究 [14] についても述べる.

2.2 時間ペトリネットによる FMS のモデル化

FMS を一つの工作機械 (搬送車) へのジョブの搬入, 加工 (搬送), ジョブの搬出という一連の離散事象の集まりと見なし, ペトリネットでモデル化する. $P = \{p_1, p_2, \dots, p_n\}$ をプレースの有限集合, $T = \{t_1, t_2, \dots, t_m\}$ をトランジションの有限集合とする. 工作機械, 搬送車とジョブの流れの代表的な関係を Fig. 2.1 に示す 4 つの型で表わす. 例えば t_1 と t_2 はそれぞれ加工開始, 加工終了を表わすトランジションであり, p_1 と p_3 はバッファ, p_2 は加工中, p_4 は加工待ちを表わすプレースである. p_1, p_2, p_3 に存在するトークンがジョブを表現している. FMS のモデル化はこれら 4 つの型の組み合わせにより行なう.

本章で用いる時間ペトリネットには次のような条件を導入している.

- (1) 各プレースにはトークンの数の上限 $L(p_i)$ を次のように与える.
 - p_i が加工中または加工待ちプレース: $L(p_i) = 1$.
 - p_i がバッファプレース: $L(p_i) = \text{バッファ容量}$.
- (2) 各プレースにはトークンの滞在時間の下限値を示す $S(p_i)$ を次のように与える.
 - p_i がバッファプレース: $S(p_i) = 0$.
 - p_i が加工待ちプレース: $S(p_i) = 0$.
 - p_i が加工中プレース: $S(p_i) = \text{加工時間}$.

ここで, バッファ容量と加工時間は正整数で与える. 条件 (2) はペトリネット表現を簡略化するため時間の概念を陽的に導入したものである. $I(t_i), O(t_i)$ をそれぞれトランジション t_i の入力プレース, 出力プレースの集合とし, 時刻 τ におけるマーキングを μ とする. このとき, 時間ペトリネットのトランジションの発火条件を次のように与える.

発火条件: 時刻 τ において, トランジション t_i は次の 4 つの条件が成立するならば発火可能である.

- (1) 任意の $p_j \in I(t_i)$ に対して $\mu_\tau(p_j) > 0$.
- (2) 任意の $p_j \in O(t_i)$ に対して $\mu_\tau(p_j) < L(p_j)$.
- (3) 任意の $p_j \in I(t_i)$ と任意の $k_j = 0, 1, \dots, S(p_j)$ に対して $\mu_{\tau-k_j}(p_j) = \text{一定}$.

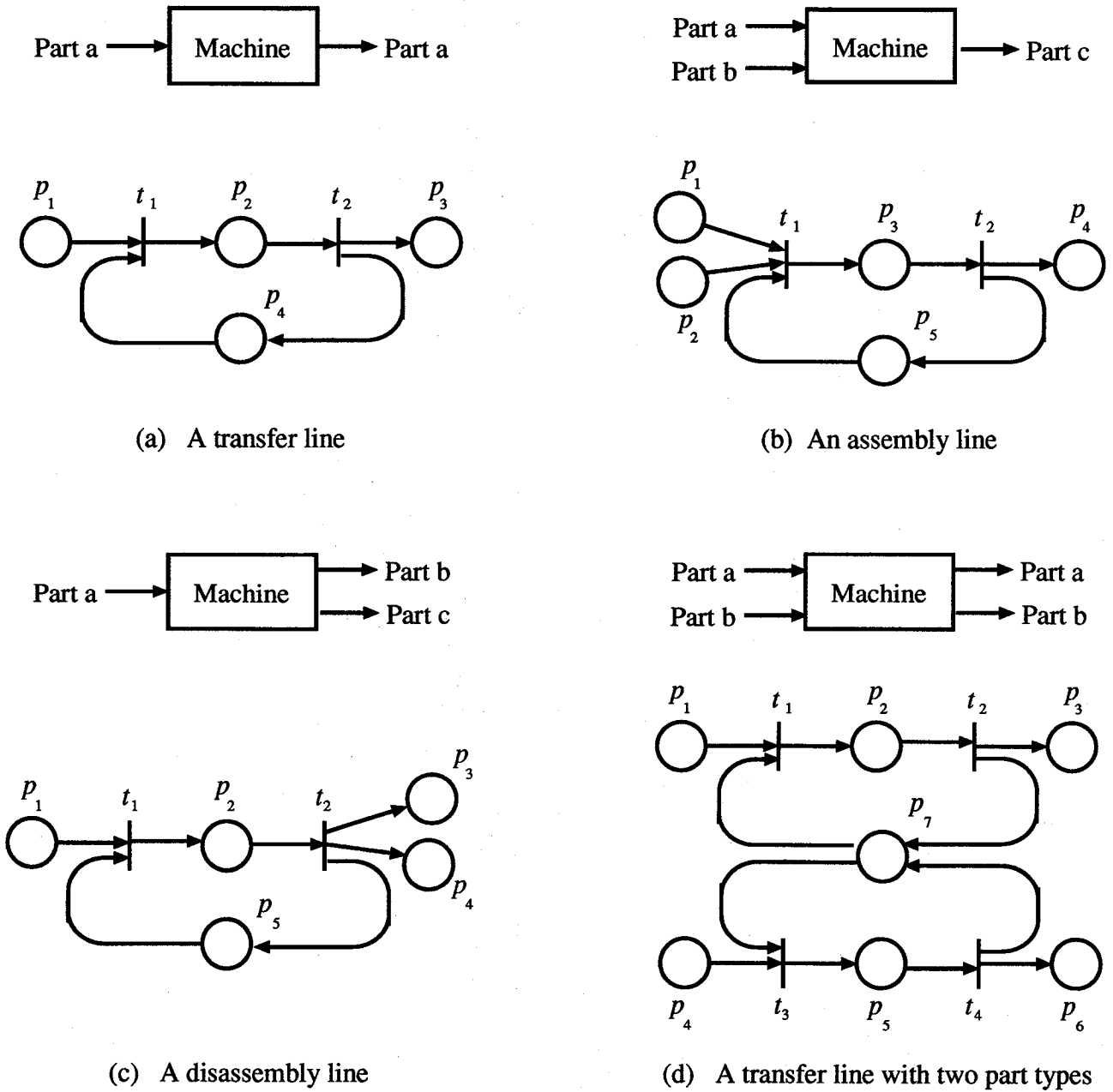
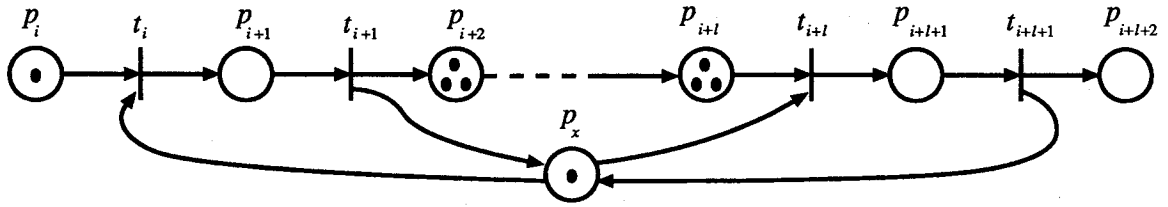
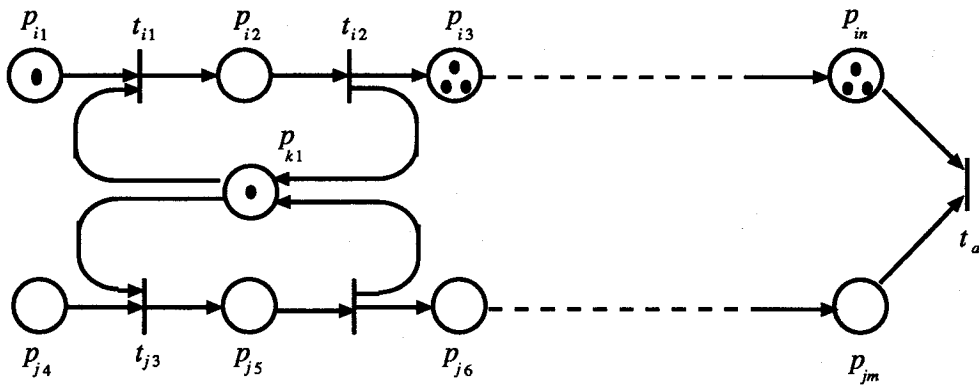


Fig. 2.1: Examples of FMS model [6].



(a) An example of deadlock of a transfer line



(b) An example of deadlock of an assembly line

Fig. 2.2: Examples of deadlocks [15].

ここで(1)~(2)は一般のペトリネットで用いられる発火条件である。(3)はトランジションの入力プレース p_j において以前にトークンが変化した時刻から $S(p_j)$ 以上時間が経過しなければ p_j 中のトークンが変化してはならないことを表わしている。

一般に、Fig. 2.1 (a)~(d) の組み合わせで FMS モデルを構築した場合、デッドロックが発生する場合がある。その代表的な例を Fig. 2.2 に示す。(a) はトランスファーラインで生じるデッドロックの例を表現している。(a) において、プレース $p_{i+2} \sim p_{i+l}$ にそれぞれトークンがトークン数の上限個存在している時、発火可能なトランジション t_i が発火すると、(a) のペトリネットはデッドロックに陥ることが容易に分かる。また、Fig. 2.2 (b) においてプレース $p_{i3} \sim p_{im}$ にそれぞれトークンがトークン数の上限個存在している時、 t_{i1} は (1), (2), (3) の発火条件によれば発火可能であるが、 t_{i1} が発火した後、Fig. 2.2 (b) のペトリネットはデッドロックに陥る。このようにして生じるデッドロックは以下に示すデッドロック回避条件により回避している。

デッドロック回避:

- (1) Fig. 2.2 (a) のような場合, トランジション t_{i1} は, プレース $p_{i3} \sim p_{l-2}$ の中にトークン数が上限より少ないものが少なくとも一つ存在する時のみ発火可能である.
- (2) Fig. 2.2 (b) のような場合, トランジション t_{i1} は, プレース $p_{i3} \sim p_{im}$ の中にトークン数が上限より少ないプレースが少なくとも一つ存在するかまたは $p_{j6} \sim p_{jm}$ の中にトークン数が1以上のプレースが少なくとも1つ存在する時に限り発火可能である.

発火条件が満たされたトランジションは次の規則に従ってマーキングを変化させる.

マーキングの変化: トランジション t_i が時刻 τ で発火すると, 時刻 $\tau+1$ のマーキングは以下のように変化する.

- (1) 任意の $p_j \in I(t_i)$ に対して $\mu_{\tau+1}(p_j) = \mu_{\tau}(p_j) - 1$.
- (2) 任意の $p_j \in O(t_i)$ に対して $\mu_{\tau+1}(p_j) = \mu_{\tau}(p_j) + 1$.

Fig. 2.1 (d) に示すように優先的に加工または搬送されるべきジョブの種類を選択する必要が生じた場合, 優先規則を適用しなければならない. このため, このような競合が生じる状態を以下に定義するような互いに競合するトランジションの集合を同定することにより検出する.

トランジションの競合: 時刻 τ でトランジション t_i が発火すると, 同時刻にトランジション t_j が発火可能でなくなる時, t_i と t_j は競合しているという.

時間ペトリネットモデルにおいては, 検出された互いに競合しているトランジションは, 同一の加工待ちプレースを持つ加工 (搬送) 開始トランジションとなる. ここで, 競合しているトランジションはジョブの種類に対応している. すなわち, 優先させるジョブの種類を決めることにより競合を解消することができる.

2.3 スケジューリングへのルールの適用

シミュレーション中でディスパッチングやルーティングなどの競合が発生した場合, 以下に示す形式のルールに従い優先規則を決定し, それに対応するジョブまたは工作機械をを選択することにより競合の解消を行なう. 本章で用いるルールは次のような形式で与える.

$$\text{if } P \text{ then } R \quad (2.1)$$

ここで条件 P は FMS シミュレータ内のバッファの状態や各ジョブの加工進展状況などを調べるための述語であり, 後で述べる試作システムでは次のようなものを用意している.

true: 常に真となる.

false: 常に偽となる.

not(a): もし a が真ならば偽となり, 偽ならば真となる.

time(a): もし現時刻が a 以降ならば真となり, その他は偽となる.

buffer1(a, b): 現工程より a 工程先のバッファの占有率が $b\%$ 以上のものが存在すれば真となり, その他は偽となる.

buffer2(a, b): 現工程より a 工程先のバッファの占有率がすべて $b\%$ 以上であれば真となり, その他は偽となる.

buffer3(a): 現工程より後にあるバッファのうち, 占有率が $a\%$ 以上のものがあれば真となり, その他は偽となる.

buffer4(b): 現工程より後にあるバッファの占有率がすべて $b\%$ 以上ならば真となり, その他は偽となる.

processing-time(a, b): 現工程から a 工程先の加工時間の最小・最大の比 $\times 100$ が $b\%$ 以上ならば真となり, その他は偽となる.

progress(a): 競合しているジョブの残りの工程数が a 以下のものがあれば真となり, その他は偽となる.

attain: 競合しているジョブの内, 現時刻で一つでも完成しているものがあれば真となり, その他は偽となる.

また, 結果 R は優先規則 [5] かまたはルールのグループ番号である. 現在試作システムで用いている優先規則は次のようなものである.

| | |
|---|---|
| R_1 : 最短加工残り時間優先 (Least Remaining Processing Time, LRPT) | R_5 : 先着優先 (First Come First Served, FCFS) |
| R_2 : 最長加工残り時間優先 | R_6 : バッファ占有率最大優先 |
| R_3 : 最短加工時間優先 (Least Processing Time, LPT) | R_7 : 次工程バッファ占有率最小優先 |
| R_4 : 最長加工時間優先 | R_8 : 目的達成率最小優先 |
| | R_9 : 残り工程数最大優先 |

ルールは結果として次にチェックすべきグループを持つものと結果として優先規則を持つものと2種類あり, ルールベースは一般的に Fig. 2.3 に示すような階層構造を持つ. ただしレベル1以上の階層に属するルールはそれぞれ次にチェックすべきグループ番号を結果として持ちレベル0に属するルールは優先規則番号を結果として持つ. 次にルールの適用について述べる. ルールは Fig. 2.3の enter で示されるグループ1の最初のルールからチェックされ, もし条件が成立すればTで示される矢印の順に, 成立しなければFで示される矢印の順にルールがチェックされる. ここで成立したルールがレベル0に属するルールならば, そのルールが結果として持つ優先規則番号をルール適用の最終結果とする. もし, あるグループに含まれるルールがすべて成立しない場合が生じた時は, 直前にチェックされたグループの中で最後に適用されたルールの次のルールがチェックされる. もしそのグループに該当するルールがなければ, さらに前にチェッ

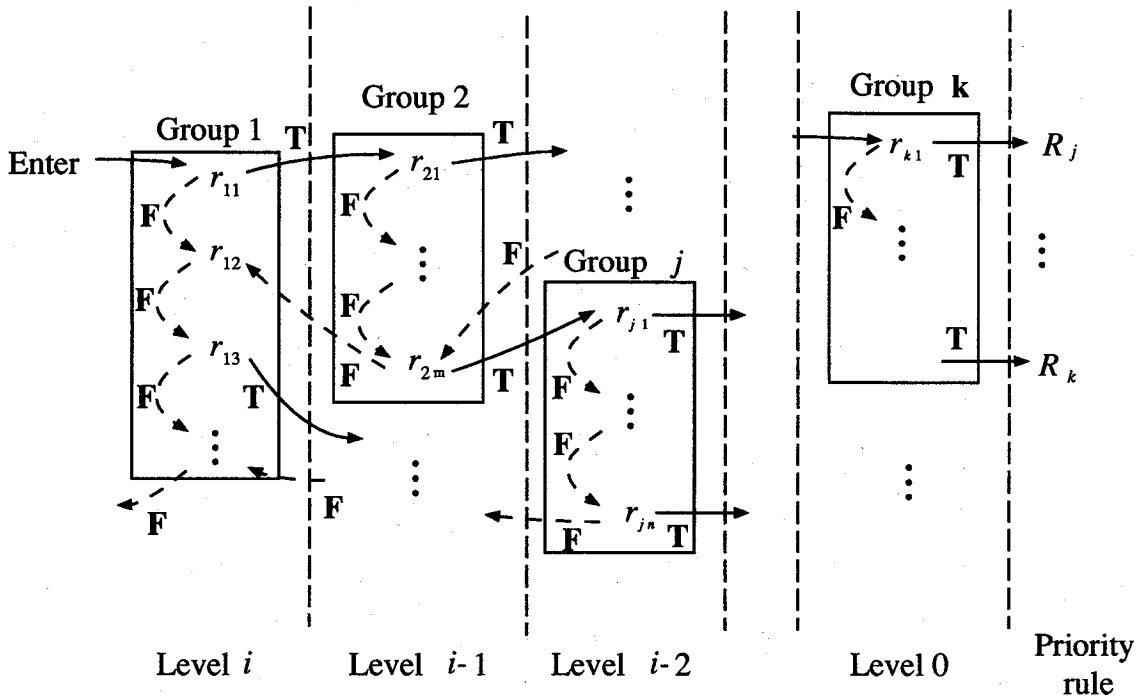


Fig. 2.3: Hierarchical structure of the rule base [6].

クされたグループから該当するルールを探す。もしレベル i のグループにも該当のルールがなければルール全体が不成立となる。シミュレータ側では推論の結果得られた優先規則番号をもとに競合の解消を行なう。

以下にルールの例を示す。

- (1) if *not*(time(2)) then check rules of group2

意味: もし現時刻が 2 以降でないすなわち 1 ならばグループ 2 のルールをチェックせよ。

- (2) if *buffer*2(0, 80) then check rules of group 3

意味: もし現工程のバッファの占有率がすべて 80 % 以上ならばグループ 3 のルールをチェックせよ。

- (3) if *progress*(1) then R_4

意味: もし残り工程が 1 以下のジョブがあれば優先規則 R_4 に従ってジョブの選択を行なえ。

2.4 試作システムの概要

前節で述べたルールを適用してスケジューリングを実現するための試作システムを作成した。このシステムは大きくわけて次の 3 つの部分からなる。

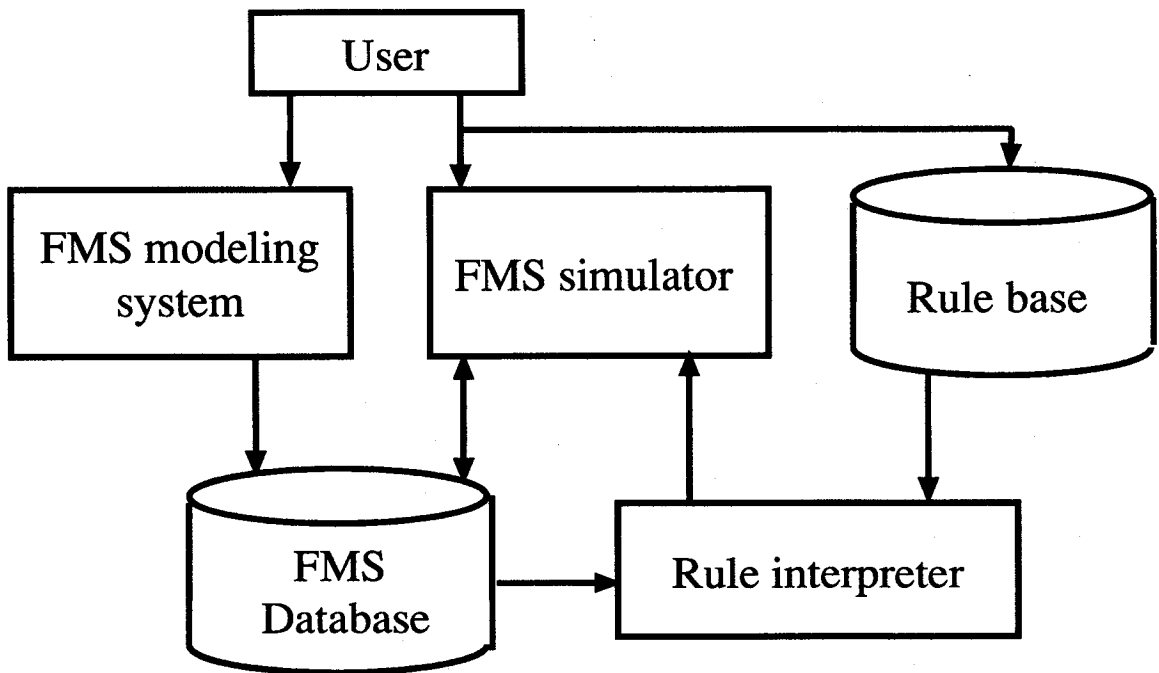


Fig. 2.4: A rule-based scheduling system [6].

(1) FMS モデリングシステム

固定ルートを持つ FMS の加工情報をもとに時間ペトリネットモデルを作成する。

(2) FMS シミュレータ

(1) で作成されたモデルをもとにシミュレーション中に生じる競合を解消しながらシミュレーションを行ない、スケジュールを生成する。

(3) ルールインタプリタ

シミュレーション中に競合が発生した時にシミュレータから呼出され、ルールベース中のルールを適用して、優先規則番号をシミュレータに返す。

このシステムの構成図を Fig. 2.4 に示す。なお、このシステムは CP/M-86 上に C 言語を用いて構成されており、ソースファイルで約 5000 行の大きさを持つ。以下に各部分の構成および機能について述べる。

2.4.1 FMS モデリングシステム

FMS モデリングシステムは、固定ルートを持つ FMS の加工情報をもとに時間ペトリネットモデルを作成する。スケジューリング対象である FMS の時間ペトリネットモデル作成にあたって必要な情報は以下の通りである。

- (1) 加工ジョブ名および番号

- (2) 使用機械名および番号
- (3) 現工程に必要な全てのジョブ名とそれらのジョブが直前に通った工程番号
- (4) 現工程で作られる全てのジョブ名と次に送る予定の工程番号
- (5) 現工程における入力側バッファの容量
- (6) 現工程の工程番号

ここで(3)は組み立て工程を取扱う場合にのみ必要であり、(4)は分解工程を扱う場合にのみ必要となる。FMSモデリングシステムに入力されたデータは時間ペトリネットに変換されメモリ上に展開される。

2.4.2 FMS シミュレータ

FMSシミュレータは、FMSモデリングシステムで入力されたモデルをもとに次のアルゴリズムで競合を解消しながらシミュレーションを行なう。

Step 1: 現時刻 τ において発火可能なトランジションを探す。

Step 2: 発火可能なトランジション中に互いに競合しているものがあれば、ルールを適用して競合を解消する。競合していないトランジションはそのまま発火させる。

Step 3: シミュレーション終了時刻でなければ時刻を1だけ進めてStep 1へもどる。

2.4.3 ルールインタプリタ

ルールインタプリタは、シミュレーション中に競合が発生した時にシミュレータから呼出される。ルールの実行中、必要があればシミュレーションの進行状況をシミュレータから読みだすことができる。ルールの実行の結果得られた優先規則番号は、シミュレータにルールインタプリタの出力として返される。

2.5 総処理終了時間短縮のためのルールベース

本節では、本章で提案するルールベースを用いたスケジューリングの有効性を検証するため、総処理時間短縮を主な目的とするルールベースの作成および優先規則を固定した場合と比較することによる本スケジューリングシステムの性能評価を行なう。

2.5.1 ルール構築のためのヒューリスティクス

ここでは、なるべく各工作機械にかかる負担を分散させ、かつ総処理終了時間を短くし、しかも同じ処理を行なう工程はなるべく連続させることにより段取り替えを低減させるというスケジューリング目的を達成するため、次の様なヒューリスティクスを用いてルールベースを作成した。

- (1) 時刻が1の時は後に処理が多く残っているジョブを先に処理する。
- (2) 現工程に残り工程が1の部品が存在する時は後に加工が多く残っているジョブを先に処理する。
- (3) 現工程の全てのバッファが一杯の時,
 - (a) 更に次工程のバッファも全て一杯の時は、最も処理時間が長いジョブを先に処理する。
 - (b) 次工程のバッファに一杯のものが存在しかつ現工程における各ジョブの処理時間に大差がある場合には次工程のバッファの中で最も占有率が低いジョブを先に処理する。
 - (c) その他の場合は、最も処理時間が短いジョブを先に処理する。
- (4) 次工程の全てのバッファが一杯の時,
 - (a) 現工程における各ジョブの処理時間に大差がある場合には最も長い加工残り時間を持つジョブを先に処理する。
 - (b) 現時刻で加工が全て終了しているジョブが一つでも存在する場合は、最も完成品が少ないジョブを先に処理する。
 - (c) その他の場合は、最も長い加工残り時間をもつジョブを先に処理する。
- (5) 現工程より後の工程のバッファが全て一杯の時,
 - (a) 現工程における各ジョブの処理時間に大差がある場合には、現工程の処理時間が最も長いジョブを先に処理する。
 - (b) その他の場合は、残り工程数が最も多いジョブを先に処理する。
- (6) 現工程より後の工程のバッファの中に一杯のものが存在する時は、もっとも長い加工残り時間をもつジョブを先に処理する。
- (7) その他の場合,
 - (a) 現時刻で処理が全て終了しているジョブが一つでも存在する場合は最も加工残り時間が長いジョブを先に処理する。
 - (b) その他の場合は、もっとも残り工程数が多いジョブを先に処理する。

以上のヒューリスティクスをルールで記述したものを **Fig. 2.5** に示す。各グループ内のルールは優先度が高いと思われる順に配置してある。

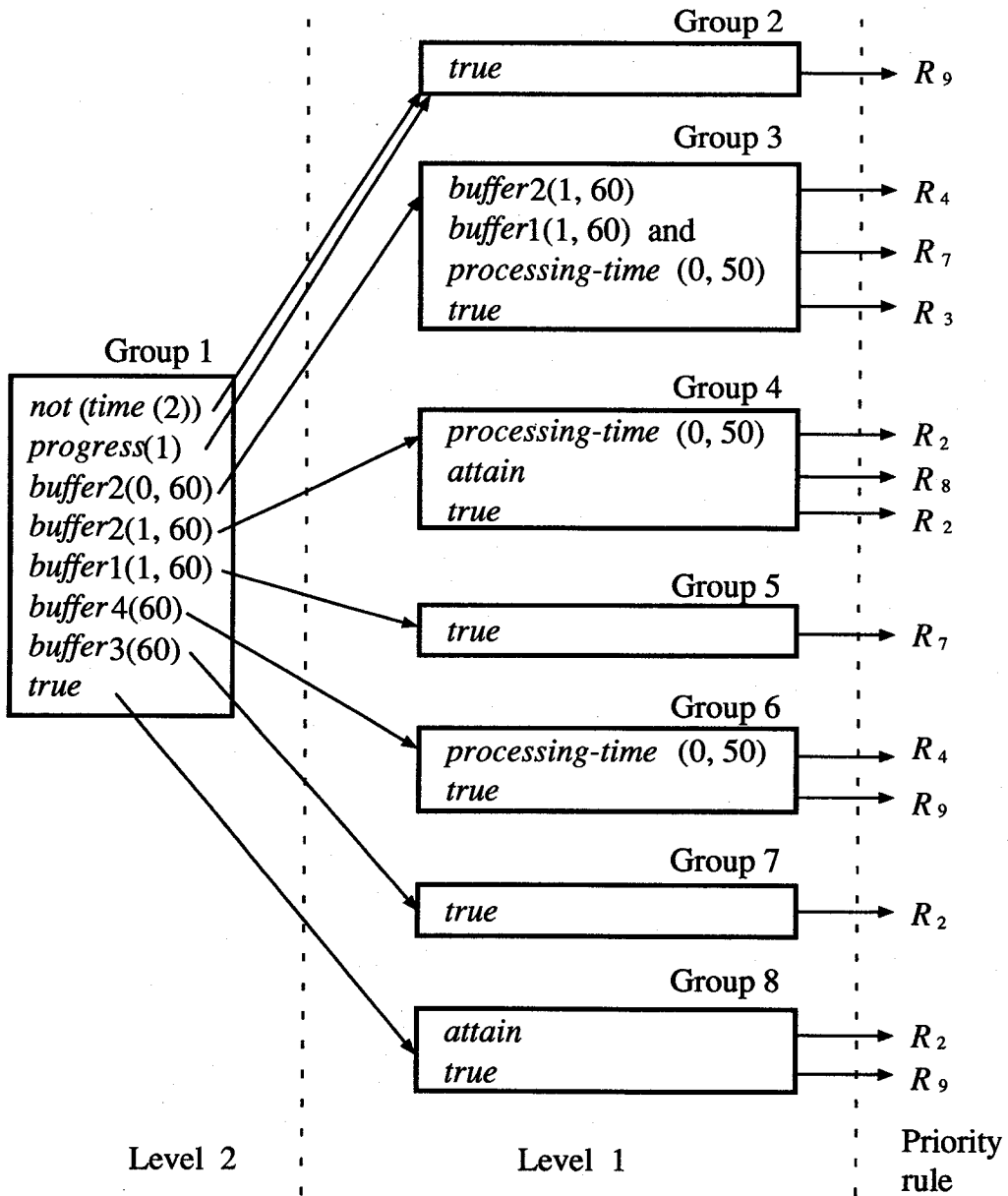


Fig. 2.5: A rule base for minimizing the completion time and the number of set-ups [6].

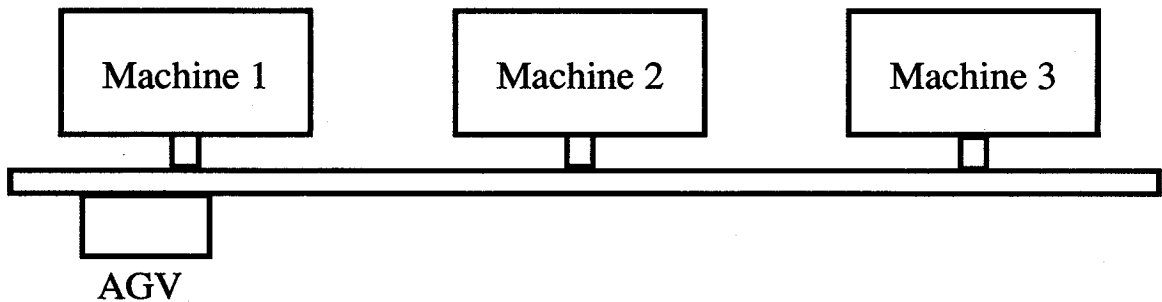


Fig. 2.6: A transfer line [6].

2.5.2 数値例

簡単な数値例として Fig. 2.6 に示すライン型 FMS のスケジューリングを次の加工情報のもとで行なう事を考える。

ジョブ 1: 機械 1 (6 分)—搬送 (1 分)—機械 2 (6 分)

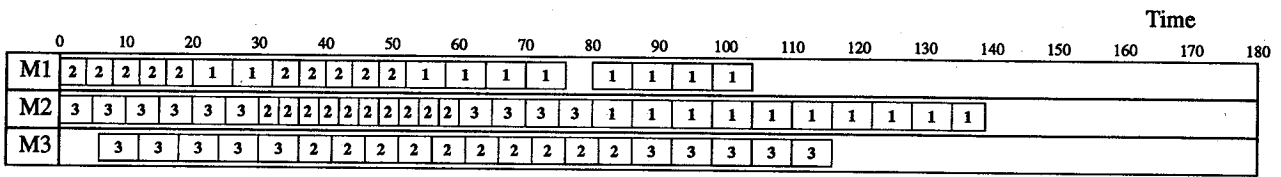
ジョブ 2: 機械 1 (4 分)—搬送 (1 分)—機械 2 (3 分)—搬送 (1 分)—機械 3 (5 分)

ジョブ 3: 機械 2 (5 分)—搬送 (1 分)—機械 3 (6 分)

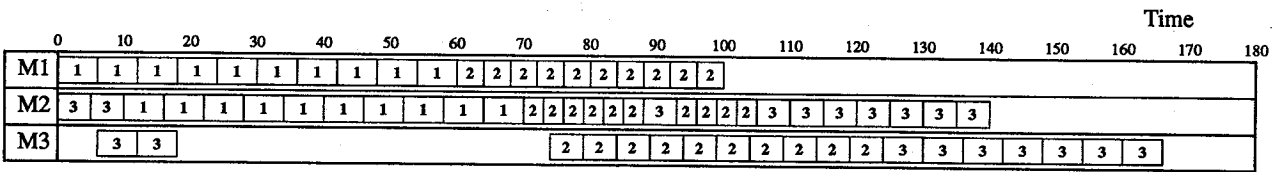
ここで各バッファ容量は 3 とし, 各ジョブを 10 個ずつ投入し, Fig. 2.5 に示すルールベースを適用したものと優先規則を R_2 : (最長加工残り時間優先), R_4 : (最長加工時間優先) および R_5 : (先着優先) に固定したものの 4 種類についてシミュレーションを行なった。

シミュレーション結果を Fig. 2.7 (a)~(d) に示す。ただし (a) は Fig. 2.5 に示すルールベースを用いて動的に優先規則を選択しながらシミュレーションを行なったものであり, (b) は優先規則を R_2 : (最長加工残り時間優先), (c) は R_4 : (最長加工時間優先), (d) は R_5 : (先着優先) に固定してシミュレーションを行なったものである。

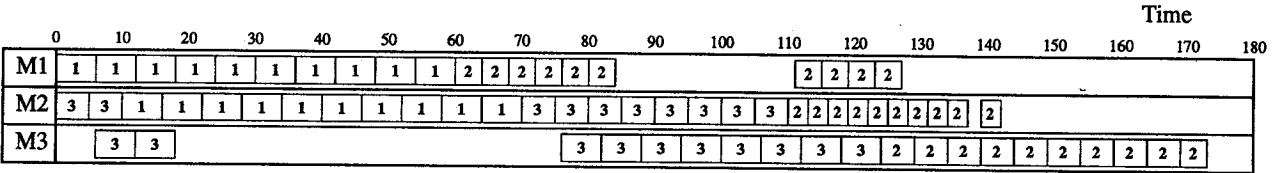
Fig. 2.7 (a) は, 明らかに作業終了時間に対して最適なものとなっていることが分かる。また, 同一機械で同種類のジョブの作業が連続しているため, 段取り替えの少ない効率的なスケジュールになっていることが分かる。一方, 優先規則を R_2 , R_4 に固定した Fig. 2.7 (b) と (c) のスケジュールは (a) に較べて作業終了時間がかなり遅いものとなっている。(d) のスケジュールは作業終了時間の面では (a) と大差はないが, 同一機械で同種類の作業が連続している部分がほとんどないため, 段取り替え時間を考慮すると (a) よりもかなり効率の悪いものとなっていると思われる。なお, 計算には PC9801E(5MHz) を使用した。計算時間は (a)~(d) とともに 3~4 分であった。



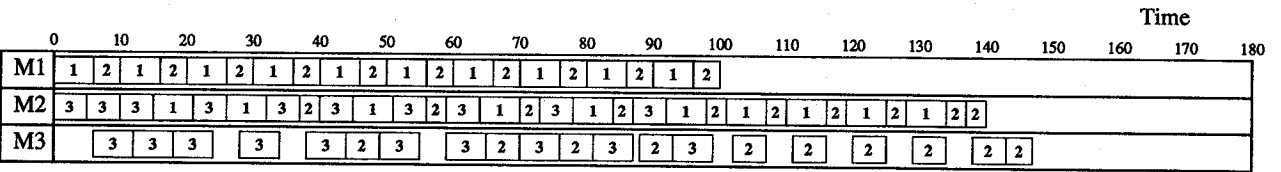
(a) A schedule generated by using the rule base



(b) A schedule generated by using LRPT



(c) A schedule generated by using LPT



(d) A schedule generated by using FCFS

Fig. 2.7: Examples of schedules.

2.6 ルール修正機能を持ったルールベーススケジューリング

一般にルールベース・スケジューリングにおいては、ルールベースの構成によりスケジューリング結果が大きく左右される。しかし、あらゆる加工情報に対して常に適当なスケジュールを発生するルールベースを作成することは困難である。このような問題に対処する1ステップとして、本節ではJIT (Just-in-Time) を目的として構築されたルールベース中のルールのパラメータを、シミュレーション結果を分析することにより修正し、従来の適応機構を持たないルールベースでは十分な性能を得ることができなかつた問題に対しても十分な性能を持つよう適応させることを試みる [14]。

2.6.1 ジャストインタイムのためのルールベース

JITを目的とする場合、納期遅れおよび納期余裕を両方ともできるだけ小さくしなければならないため、“どのジョブも負荷しない”という優先規則を加える必要がある。そのため、シミュレーション中に作業の競合が発生した場合だけでなく、開始可能な作業が存在した場合、常にルールベースを適用する。本論文では、各ジョブについてそれぞれ設定された納期に対して、納期遅れおよび納期余裕が最小となるようなJITを目的としたルールベース [9] を以下のような方針のもとで作成した。

- (1) 加工を開始する必要の生じる時刻までジョブを機械に負荷しない。
- (2) 競合を起こしている全てのジョブについて納期までの余裕がある場合は納期までの余裕が最小と思われるものを先に加工する。
- (3) ジョブがラインに投入された後も納期までの余裕率が大きい場合は次の加工を開始しない。
- (4) 競合を起こしているジョブのうち、最終搬送工程にあるものがあればそのジョブを先に搬送する。
- (5) 次工程の全てのバッファが一杯の場合はどのジョブも搬送しない。また、一杯でない場合はバッファ占有率が最小のジョブを先に搬送する。

以上のようなヒューリスティクスをルールで表現したものを **Fig. 2.8**に示す。ルール記述のために用いた組込述語は11~12頁に示したものおよび以下のものを用いている。

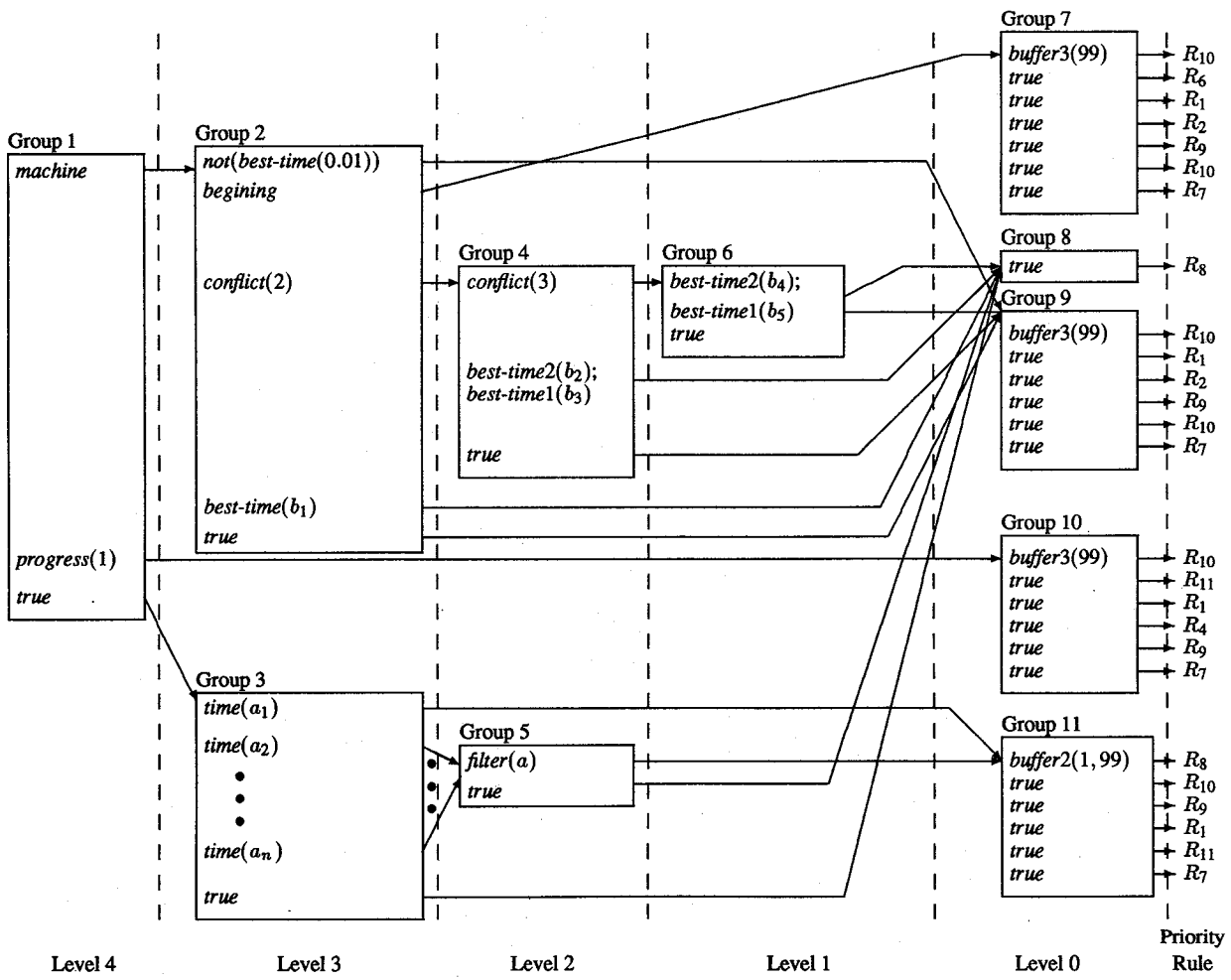


Fig. 2.8: An example of rule base for JIT [15].

progress1(a): 競合しているジョブの残りの工程数が a 以下のものがあれば真となり、
 その他は偽となる。

progress2(a): 残り工程数がすべて a 以下の時のみ真となる。

attain1: 競合しているジョブの内、現時刻で一つでも完成しているものがあれば真となり、
 その他は偽となる。

attain2: 現時刻で全てのジョブが完成品を一つでも持つ時のみ真となる。

worst-time1(a): 納期までの残り時間とそのジョブについての I_{max} の比が a 以上の
 ものがある時のみ真となる。

worst-time2(a): 納期までの残り時間とそのジョブについての I_{max} の比が全て a 以上
 である時のみ真となる。

best-time3(a): 納期までの残り時間とそのジョブについての I_{min} の比が a 以上のもの
 がある時のみ真となる。

best-time1(a): 納期までの残り時間とそのジョブについての I_{min} の比が全て a 以上
 である時のみ真となる。

conflict(a): 競合しているジョブが a 個以上の時のみ真となる。

machine: 競合が工作機械で発生している時のみ真となる。

beginning1: 既に作業開始しているジョブがある時のみ真となる。

beginning2: 全てのジョブが作業開始している時のみ真となる。

ここで、 I_{max} と I_{min} は述語の真偽を検証する時刻において受注個数全てを仕上げるのにかかる
 と予想される最大時間および最小時間の推定値を表し、次式のように表わされる。

$$I_{max} = T \cdot n \quad (2.2)$$

$$I_{min} = T + (n - 1)S \quad (2.3)$$

ただし、

n : 未完成ジョブの数

T : 現工程以降の作業時間の総和

S : 現工程以降の作業時間の最大値

であるものとする。

また、JITのためのルールベースで用いている優先規則を以下に示す。

| | |
|---------------------------|-------------------------|
| R_1 : 最短加工残り時間優先 | R_{11} : 納期順優先 |
| R_2 : 最長加工残り時間優先 (LRPT) | R_{12} : スラック時間最小優先 |
| R_3 : 最短加工時間優先 | R_{13} : 総納期遅れ時間最大優先 |
| R_4 : 最長加工時間優先 (LPT) | R_{14} : 総納期余裕時間最小優先 |
| R_5 : 先着優先 (FCFS) | R_{15} : 最大納期遅れ時間最大優先 |
| R_6 : バッファ占有率最大優先 | R_{16} : 納期遅れ個数最大優先 |
| R_7 : 次工程バッファ占有率最小優先 | R_{17} : 納期余裕率最小優先 |
| R_8 : 目的達成率最小優先 | R_{18} : ランダム |
| R_9 : 残り工程数最大優先 | R_{19} : 負荷しない |
| R_{10} : 残り工程数最大優先 | |

R_8 のスルーputとは完成ジョブの数と投入ジョブの数の比を表わす。 R_{12} のスラック時間とは納期までの残り時間から作業残り時間を引いた時間を意味する。 R_{17} の納期余裕率は納期までの残り時間とそのジョブについての I_{min} の比を表わす。

2.6.2 ルールの適応

JITを目的として本研究で試作したスケジューリングシステムでは、バッファ容量の適正化、機械稼働率の向上、段取り替え回数の低減のための規則の他に、ルールベース中に設定した時刻 T 以前にはジョブをラインに投入しないようなルールを導入している。さらに、加工完了時刻を調整するため、各工程の開始前にそのジョブの加工完了までの予想時間に対する納期までの残り時間の比がある定数 C より大きい場合にはその工程を開始せずに納期が近づくまで加工を休止するようなルールを導入している。定数 T, C はヒューリスティックなパラメータであり、与えられた加工情報により変化させる必要があると予想される。

ここでは、スケジューリング結果を評価して得られる情報のうち加工開始時刻および加工完了時刻をスケジューリング結果から読みとることにより、次のような手順でルール中のヒューリスティックなパラメータ T, C を調整している。

Step 1. T, C と C の増分 ΔC に適当な初期値を与える。

Step 2. シミュレーションを行ない、スケジュールを生成する。

Step 3. 各ジョブ p_i の加工開始時刻 s_i および加工終了時刻 e_i から C の値を $C + \Delta C$ とする。

Step 4. シミュレーションを行ない、スケジュールを生成する。

Step 5. 納期余裕が改善されれば Step 3 へ、その他は Step 6 を実行する。

Step 6. $T = \min(s_i)$ として、さらにスケジュールを求める。

2.6.3 数値例

簡単な数値例としてライン型 FMS の JIT を目的としたスケジューリングを考える。FMS は 3 台の工作機械 (a,b,c) とローディング/アンローディングステーションおよび 1 台の搬送車から構成されているものとする。このような FMS に Table 2.1 に示す加工情報のもとで 3 種類のジョブ (A,B,C) の加工を行なうものとする。このような条件で生成されるスケジュールを、ルールを適応させた場合と適応させなかった場合とを比較した結果を Table 2.2 に示す。

Table 2.2 において、各項目の左の数値が本システムを用いて得たものであり、右がルールの適応を行なわないで得たものである。表からもわかるように、ほとんどの場合においてルールを適応させた方がよりよいスケジュールを得ていることがわかる。特にルールを適応させない場合に十分な結果が得られていない問題 (Case 1, Case 2, Case 6) においてかなり良い結果が得られていることがわかる。

Table 2.1: Processing information [14].

| Case 1 | | | | Case 2 | | | | Case 3 | | | |
|--------|---|---|---|--------|---|---|----|--------|---|---|---|
| | a | b | c | | a | b | c | | a | b | c |
| A | 1 | - | 8 | A | 1 | 6 | 10 | A | 1 | 3 | 5 |
| B | - | 3 | 5 | B | 2 | 9 | 8 | B | - | 6 | 3 |
| C | - | - | 9 | C | - | 8 | 8 | C | 1 | 4 | 4 |

| Case 4 | | | | Case 5 | | | | Case 6 | | | |
|--------|----|----|---|--------|---|---|---|--------|---|---|---|
| | a | b | c | | a | b | c | | a | b | c |
| A | 1 | 10 | - | A | 7 | 4 | - | A | 6 | 6 | - |
| B | 10 | 1 | - | B | - | 4 | 4 | B | 4 | 3 | 5 |
| C | 2 | 2 | 2 | C | 6 | - | 4 | C | - | 5 | 6 |

Table 2.2: Comparison of schedulings [14].

| Case | Total earliness | | Total minimum earliness | | Average machine utilization | |
|------|-----------------|--------------|-------------------------|--------------|-----------------------------|--------------|
| | Adaptive | Non-adaptive | Adaptive | Non-adaptive | Adaptive | Non-adaptive |
| 1 | 460 | 495 | 19 | 55 | .326 | .380 |
| 2 | 559 | 546 | 7 | 58 | .615 | .615 |
| 3 | 130 | 166 | 2 | 5 | .529 | .592 |
| 4 | 165 | 215 | 4 | 2 | .444 | .491 |
| 5 | 190 | 216 | 6 | 6 | .469 | .562 |
| 6 | 340 | 315 | 4 | 29 | .512 | .595 |

2.7 むすび

本章では、シミュレーションを用いたスケジューリングにおいて次に加工するジョブに競合が生じた際、ルールベースを用いてその状況に適した優先規則を動的に選択し競合を解消することにより、より効率的なスケジュールを得ることを試みた。今後は試作システムを拡張して固定ルートを持つ FMS 以外にも適用できるようにするとともに、使用できる優先規則や組込み述語をふやし、さらに大規模な例について性能評価を行なう必要がある。

また、JIT を目的として構築されたルールベース中のルールのパラメータを、シミュレーション結果を分析することにより修正し、適応機構がないルールベースでは十分な性能を得ることができなかった問題に対しても十分な性能を持つよう適応させることを試みた。そして、スケジューリング結果を、従来のルールベースと比較することによりその有用性を評価した。今後の課題として、スケジューリング結果からルールの構造自体を適応させるような学習機能を開発することがあげられる。

第3章 ファジィ推論を用いたフレキシブル 生産スケジューリング

3.1 まえがき

フレキシブル生産システム (FMS) [1] を効率的に運用するためには、FMS の構成、生産対象、生産目標などの変化に柔軟に対応可能なスケジューリングシステムが必要不可欠である。

第2章に述べたように、シミュレーションを用いたスケジューリング [1-3] において部品の競合が発生した場合に、ルールベースを用いることにより時々刻々の状況に適した優先規則を動的に発生させてスケジューリングを行なうシステムが開発されている [4,5]。しかし、FMS の構成、スケジューリング目標などが変化した場合、ルールベース自体を変更する必要がある。この作業を行なうには熟練した技術が必要であり、ユーザには困難な場合が多い。これは、一般にヒューリスティックな知識は非構造的であり、部分的な修正が推論におよぼす影響を予想することが困難なためである。

本章では、シミュレーションを用いたスケジューリングにおいて、競合した部品の選択のための知識ベースを、現在までに生成されているスケジュールを評価するスケジュール評価モジュール、今後のスケジューリング方針を決定するスケジューリング方針決定モジュールおよび競合した部品の選択を行なうディスパッチングモジュールに分割し、ファジィ述語を含むファジィ・ルール [6] によって各モジュールにおける知識を記述して知識のモジュール化を行なう。さらにこの中で最もユーザが直感的に理解し易いと考えられるスケジュール評価モジュール中の知識に含まれるルールのメンバーシップ関数を操作することにより、多様なスケジューリング目的に対応可能なスケジューリングシステムを構成することを試みる [7,8]。

3.2 多様なスケジューリング目的の実現のための知識ベース

3.2.1 知識ベースの構造と柔軟さ

知識ベース・スケジューラを構築する際の重要な問題は、スケジュール評価項目をいかに組み合わせるかということである。これは、知識ベース中のスケジューリング知識の構造、すなわち、スケジューリング意思決定構造をどのようなものにするかということの意味する。具体的

には、スケジューリング知識はそれが言及している評価項目ごとに分類され、その断片化された知識は各々ルールとして表現されるが、このルールをどのような順番で適用してスケジュール解を絞り込んでいくかということが重要である。スケジューラがどのようなスケジュールを発生できるかはこの構造によって決ってしまう。ヒューリスティックな知識は対象依存で ad hoc であり、適用範囲が狭いにもかかわらず、従来の AI スケジューラ [4, 5, 9-13] においてはスケジューリング意思決定構造は、固定的なものにしている。このため、知識ベースが有用であるスケジューリング目的というものが決まっており、その目的に対してしか適用できず、柔軟性に乏しいという問題点がある。例として、Turksen ら [12] のスケジューリング知識は、納期という観点からスケジュール対象となるジョブを選択し、そのジョブに対する代替作業の中から機械稼働率という観点から作業を選択し、次にその作業を行なう機械を選択し、最後に、入力バッファ中でのジョブの位置を選択するという意思決定構造を持つ。最初に納期によってジョブを絞り込んでいることにより、これは納期重視の意思決定構造であると言える。したがって、このスケジューリングシステムは、総所要時間最小、納期遅れ最小などのスケジュール目標にも対応している。しかし、納期を重視しなくてもよい生産においては、納期によってスケジュール解を絞り込んでいくことは無意味であり、別の意思決定構造、例えば機械稼働率重視の意思決定構造（機械稼働率によってスケジュール対象となる機械を絞り込み、更にその機械を使用する作業から対象を絞り込んでいく）を持つスケジューラが必要である。

現実の生産現場では、総所要時間最小とか納期遅れ最小のような単純な生産目的のみでなく、複雑なスケジューリング目的に対応できるスケジューラが求められている。現実的なスケジュール要求の例としては、「機械 M1 の加工速度は早いですが、M1 は故障しがちである。そこで、機械 M1 をなるべく使用しないで、できるだけ総所要時間を小さくするスケジューリングを行なえ」などが考えられる。しかし、固定的な意思決定構造をもつ知識ベースでは、スケジューリング目的ごとに知識ベースを作成する必要があり、対応が困難である。

Bensana ら [10] はディスパッチングルール（スケジューリングのためのヒューリスティクス）ごとにスケジューリング目的に対する重要度という属性をもうけ、この値（ユーザが恣意的に決定）に応じてルールを結合させて意思決定を行なっている。これにより、意思決定構造は重要度に応じて変更され、一つの知識ベースで様々なスケジュール要求に対応することができ、知識ベースのスケジュール要求に対する柔軟性がでてくる。しかし、ここで問題となるのは、ユーザが個々のスケジューリングルールに重要度を直接割り当てることによりスケジュール目的をスケジュールに反映させることである。すなわち、ユーザはスケジュール要求に応じて知識構造そのものを直接変更しなければならないのである。このためには、ユーザは、どのように重要度を設定するとどの様なスケジュールになるかを明確に理解していなければならない。このルール重要度とスケジュールの対応に関する知識は、状況に依存した複雑のものであり、かなりのユーザ負担となることが予想される。

フレキシブル生産システムの効率的運用のためには、スケジュール要求・状況に応じてスケジュール知識の構造を動的に変更するスケジューラが必要である。このようなスケジューラは、柔軟に変更できるスケジュール知識の構造と、スケジュール要求をその知識の変更に反映させるメカニズムを持っていなければならない。

次節で、今回作成したモデルでこれらのスケジューラに対する要求をどの様に実現しているかについて述べる。

3.2.2 システム動作の概要

従来は、ユーザがスケジュール要求に応じてスケジューリング知識を直接変更しなければならなかった。このため、ユーザがスケジューラを深く理解している必要があった。これに対して、今回のスケジューラはユーザが要求を入力するとスケジュール知識がそれに適応するように自動的に変更され、その知識によりスケジューリングを行なう。また、知識の表現にファジィ理論を適用することにより、知識ベースと人とのインターフェイスが言語的表現で行なえるという特徴を持つ。

スケジュール要求に対して柔軟なスケジューラを実現するために次のような知識ベースモデルを構築した。モデル中のスケジュール知識は、動的に変化できるように、ファジィ推論を使って、重要度に応じてルールを結合させて全体的な知識とする構造を持つ。また、スケジュール要求・状況とルール重要度の対応付けを別の知識ベースとして持たせておく。すなわち、今回作成した知識ベーススケジューラは、実際にスケジューリングを行なう知識ベース（スケジューリング知識ベース）とそれに加えて、その知識ベースを状況・要求に適応させる知識ベース（知識構造決定知識ベース）を持つ。本章では、スケジューリングシステムの持つ知識を“重要である”、“適切である”などのファジィ述語を含むファジィルールで構成されたスケジュール評価モジュール、スケジューリング方針決定モジュールおよびディスパッチングモジュールの3つのモジュールに分割している。このうち、スケジュール評価モジュールおよびスケジューリング評価モジュールが知識構造決定知識ベースに相当し、ディスパッチングモジュールがスケジューリング知識ベースに相当する。

FMS シミュレータにおいて、次に加工すべき部品を選択しなければならない状況（ディスパッチング）または加工終了後次に行くべき機械を選択しなければならない状況（ルーティング）が生じた場合、最初にスケジュール評価モジュールが呼び出され、機械の稼働率、納期余裕、各機械における待ち時間の累積値などの評価項目をもとにして現時刻までに得られているスケジュールの評価値を算出する。その評価値に基づいて、スケジューリング方針決定モジュールにおいてスケジュール評価値への各評価項目の影響度を決定する。ディスパッチングモジュールには部品の加工情報とスケジュール評価モジュールの各項目を改善するためのヒューリスティックルールが各項目ごとに集積されており、スケジューリング方針決定モジュールで決定された各評

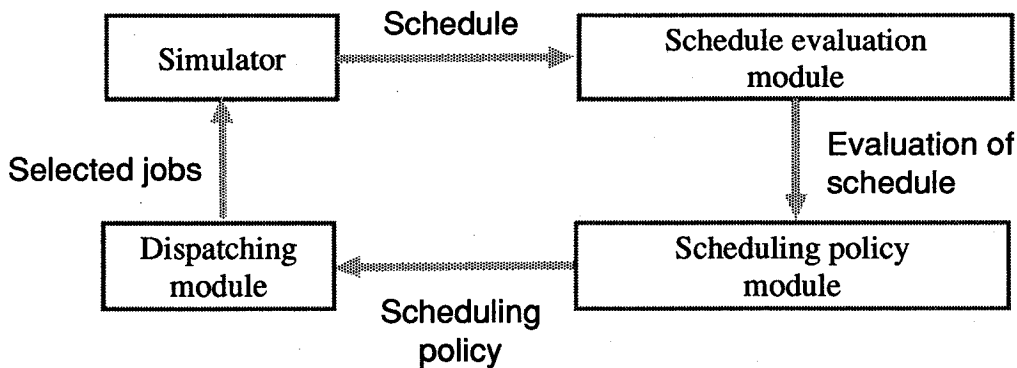


Fig. 3.1: An overview of flexible scheduling [8].

価項目の影響度がそれぞれの評価項目に対応するルール群に重みとして割り当てられる。ディスパッチングモジュールは各ルール群に割り当てられた重みに基づいて次に加工すべき部品または次に行くべき機械を選択する。すなわち、ディスパッチングモジュールの動作はスケジュール評価モジュールにおいてどのようなスケジュールを“良い”ものとするかに依存することになる。本システムでは、スケジュール評価モジュール中のファジイ述語のメンバーシップ関数を調節することにより、容易にユーザの望むスケジュールを生成するようにシステムを適応させることができる。Fig. 3.1にシステム動作の概要を示す。次節以下にそれぞれのモジュールにおける知識表現および推論について述べる。

3.3 スケジュール評価モジュール

スケジュール評価モジュールは、機械稼働率、ある機械に対する待ち時間の予想値を表わす密集度、ロット納期余裕、部品の待ち時間という4つの観点から現時点までに生成されているスケジュールの評価を行なう。

3.3.1 部分スケジュール評価のための評価項目

本システムは、評価項目を用いてスケジュール目的を記述してスケジューラに入力する。評価項目としては、次のものを使用する。

- 機械稼働率
- 機械利用の密集度
- ロット納期余裕時間
- 待ち時間

以下に各項目について説明する。

A. 機械稼働率

現時刻 t までの機械 M_i の稼働率 $U(M_i)$ は次のように定義される.

$$U(M_i) = \frac{\text{機械 } M_i \text{ の現在までの作業時間}}{\text{機械 } M_i \text{ の現在までの作業時間}} \quad (3.1)$$

B. 機械利用の密集度

密集度は、ある機械に対する待ち時間の将来の予想値を意味する. 代替作業が存在しない場合についてまず説明し、代替作業が存在する場合に対して拡張する.

代替作業が存在しない場合:

ある時点まで作業が進んでおりかつ代替作業が存在しないジョブ i の残りの各作業 p_{i1}, \dots, p_{im} が^s, $p_{i1} \rightarrow \dots \rightarrow p_{im}$ の順番で処理されるものとする. また、それぞれの作業の作業時間が^s $T(p_{ij}), j = 1, \dots, m$ で与えられるものとする.

(1) 生産システム中に搬入されている各ジョブ i の各作業 p_{ij} 毎に最早処理開始時刻 ($S_{p_{ij}}$) および最早処理完了時刻 ($F_{p_{ij}}$) を計算する. ここで、ジョブ i の最早処理開始時刻および最早処理完了時刻は、 $S_{p_{i1}}$ をジョブ i の最初の作業の最早処理開始時刻とすると以下のよう
に定義される.

$$\begin{aligned} S_{p_{ij}} &= S_{p_{i1}} + \sum_{n=1}^{m-1} T(p_{in}) \quad (m > 1) \\ E_{p_{ij}} &= S_{p_{i1}} + \sum_{n=1}^m T(p_{in}) \end{aligned} \quad (3.2)$$

すなわち、最早処理開始時刻はジョブ i の作業が最も早く開始可能な時刻を意味し、最早処理完了時刻は待ちが全くないと仮定した場合、処理が最も早く終了する時刻を意味する.

(2) (1) で求めた各最早処理開始時刻と処理終了時刻から次式を用いて各工作機械 M_j の密集度 $C(M_j)$ を求める.

$$\begin{aligned} C(M_j) &= \frac{\text{機械 } M_i \text{ に対する各処理の待ち時間の総和の予測値}}{\text{システム内に存在するジョブに対する機械 } M_i \text{ の負荷量}} \\ &= \frac{\sum_{p \in E_{M_j}} T(p) - (\min_{q \in E_{M_j}} S_q - \max_{r \in E_{M_j}} F_r)}{\sum_{s \in E_{M_j}} T(s)} \end{aligned} \quad (3.3)$$

ただし、 E_{M_j} は、機械 M_j を使用する作業でシステム内に搬入されているものの集合を意味するものとする. (3.3) 式から容易にわかるように E_{M_j} は機械 M_j に対する各作業待ち時間の和の予想値が大きい場合に 1 に近い値をとる. すなわち、 E_{M_j} が 1 に近ければ機械 M_j で各作業は長く待たされる可能性が高いことを意味している.

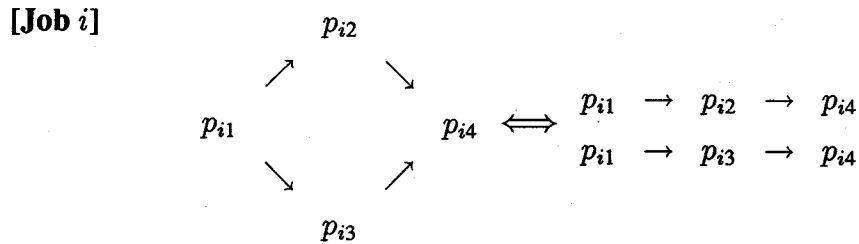


Fig. 3.2: Example of alternative operations of job i .

代替作業が存在する場合:

Fig. 3.2 のようにあるジョブに対して複数のルートが存在するとき、実際にどのルートを通って処理されるか予想がつかないため、代替作業が存在する場合の定義をそのまま適用することができない。本章では、各ルートが平均して同回数使用されると仮定することにより、以下の手続きで密集度を計算する。

- (1) 代替作業が存在するジョブについて可能な全ての作業のルートを生成する。
- (2) (1) で生成された各ルートの各処理毎に最早開始時刻および最早完了時刻を計算する。
- (3) 次式を用いて各機械 M_j 毎の密集度を計算する。

$$\begin{aligned}
 C(M_j) &= \frac{\text{機械 } M_j \text{ に対する各処理の待ち時間の総和の予測値}}{\text{システム内に存在するジョブに対する機械 } M_j \text{ の負荷量}} \\
 &= \frac{\sum_{p \in E_{M_j}} T(p)/d(p) - (\min_{q \in E_{M_j}} S_q - \max_{r \in E_{M_j}} F_r)}{\sum_{s \in E_{M_j}} T(s)/d(s)} \quad (3.4)
 \end{aligned}$$

ただし、 $d(p)$ は、作業 p が含まれるジョブの可能なルート数を表わす。(3.4) 式は、(3.3) 式における各作業の処理時間を各ジョブのルート数で割ることにより、計算における処理時間の重複を避けている。

C. ロット納期余裕

本章では、ジョブ i のロット納期余裕 $L(i)$ を代替ルートを持つジョブの存在を考慮して次式のように定義する。

$$\begin{aligned}
 L(i) &= \frac{\text{納期} - \text{現時刻}}{\text{残り作業時間}} \\
 &= \frac{D_i - t}{\sum_{y \in R_i} T(y)/d(y)} \quad (3.5)
 \end{aligned}$$

ここで, R_i はジョブ i に属する作業でまだ終了していないものの集合を意味する.

D. 待ち時間

本章においては, 作業待ち時間とジョブ待ち時間の2種類の待ち時間を用いる. 作業 p の作業待ち時間 $O(p)$ は各作業が実行可能になってからの経過時間を意味するものとする, ジョブ i のジョブ待ち時間 $J(i)$ は残りの作業の作業待ち時間の和として, 次式のように定義できる.

$$J(i) = \sum_{p \in R_i} O(p) \quad (3.6)$$

3.3.2 スケジュール評価のためのファジイ推論

スケジュール評価モジュール個々の知識は条件命題

$$\begin{aligned} &\text{If } (x_1 \text{ is } A_1) \text{ and/or } (x_2 \text{ is } A_2) \text{ and/or } \dots \\ &\text{and/or } (x_n \text{ is } A_n) \quad \text{then } y \text{ is } C \end{aligned} \quad (3.7)$$

として表現される [6]. ここで, x_1, x_2, \dots, x_n, y は, “稼働率”, “待ち時間” などのスケジュール評価項目, A_1, A_2, \dots, A_n, C は “重要である”, “適切である” などのファジイ述語である. また, 各ファジイ述語間の結合関係が複雑で明確な AND/OR 結合で表現することが困難な場合は, “ほとんど全て”, “少なくとも一つ” などの量限定子を用いて以下のように表現することができる.

$$\text{If } (Q \text{ } B(\{x_1, x_2, \dots, x_n\}) \text{ are } A) \text{ then } y \text{ is } C \quad (3.8)$$

ここで, A, B は “重要である”, “適切である” などのファジイ述語, Q は “ほとんど全て”, “少なくとも一つ” などの量限定子 [14–18] を表わす. 量限定子を用いて記述されたルールの例としては, “もし, ほとんど全ての機械の稼働率が適切でないならば機械稼働率という観点でのスケジュール評価が悪い” があげられる.

各観点における評価は, Fig. 3.3 に示す推論木を用いて統合される. スケジュール評価モジュールにおける推論は, Fig. 3.3 に示すようにシミュレータから稼働率, 納期余裕などの事実を推論木の最下層のノードにそれぞれ入力し, 推論木中に真理値を伝播させることによって推論木中の各評価項目における真理値を求める過程として定式化される. そのため, 各ルール中のファジイ述語のメンバーシップ関数を変化させることにより, どの評価項目を重視した評価を行なうかを操作することができる. 各ルールは, “もし重要な機械の稼働率が高いならばスケジュールの評価を高める” といった直感的に容易に把握できるもので構成されており, システムの内容を詳細に理解しなくても容易に調節することができる. 以下に, 前件部が AND/OR 構造で表現される場合と量限定子で統合される場合ごとに推論法を詳述する.

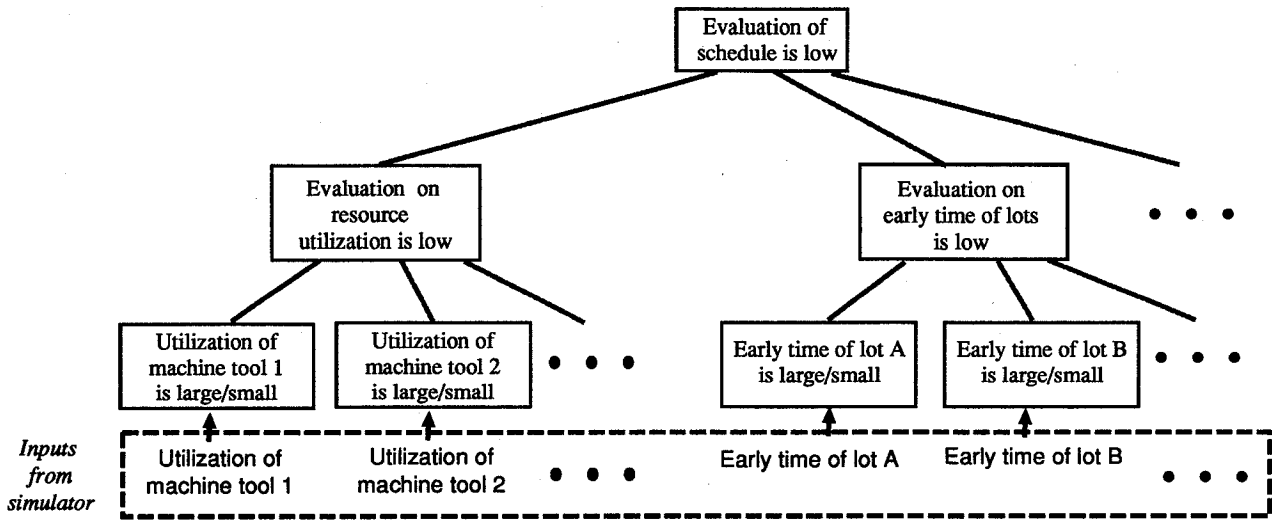


Fig. 3.3: An example of inference tree in a schedule evaluation module [8].

A. AND/OR で表現される推論木における推論

AND/OR 構造で表現される推論木における真理値伝播のプロセスは、ルール前件部の合成命題の真理値を求める真理値の統合過程と、前件命題の真理値から後件命題の真理値を求める条件命題による真理値の伝播過程よりなる。真理値の統合は真理値空間上での論理演算を定めることによって規定される [19].

$$\tau(x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2) = \min(\tau(x_1 \text{ is } A_1), \tau(x_2 \text{ is } A_2))$$

$$\tau(x_1 \text{ is } A_1 \text{ or } x_2 \text{ is } A_2) = \max(\tau(x_1 \text{ is } A_1), \tau(x_2 \text{ is } A_2))$$

ここで、 $\tau(P)$ は命題 P の真理値を表わし、ファジィ述語 P に与えられたメンバーシップ関数により決定される。

条件命題による真理値伝播のプロセスは、次のように定式化される。前件 “ x is A ” の真理値 $\tau(x \text{ is } A)$ と、条件命題 “If x is A then y is B ” (以下では、“ x is $A \rightarrow y$ is B ” と表記する) そのものの真理値 $\tau(x \text{ is } A \rightarrow y \text{ is } B)$ から後件 “ y is B ” の真理値 $\tau(y \text{ is } B)$ を求める。これを数値的真理値を用いたファジィ推論 [20] により実現する。

この問題においては、条件命題 “ x is $A \rightarrow y$ is B ” の定義は含意演算子 I

$$I : [0, 1] \times [0, 1] \rightarrow [0, 1]$$

$$I(\tau(x \text{ is } A), \tau(y \text{ is } B)) = \tau(x \text{ is } A \rightarrow y \text{ is } B) \quad (3.9)$$

によって表現される。条件命題を定義する方法は定まっていない。いろいろな多値論理ごとに条

件命題の定義がなされている。ここでは、Lukasiewicz の論理 [20] における推論則、

$$\tau(x \text{ is } A \rightarrow y \text{ is } B) = I(\tau(x \text{ is } A), \tau(y \text{ is } B)) = (1 - \tau(x \text{ is } A) + \tau(y \text{ is } B)) \wedge 1 \quad (3.10)$$

を用いる。この定義によると、条件命題が真であるということは、後件の真理値は前件の真理値を下回ることがないことを意味する。ここで、条件命題そのものが常に真ではないので

$$\tau(x \text{ is } A \rightarrow y \text{ is } B) \geq 1 - \varepsilon \quad (0.5 \geq \varepsilon \geq 0) \quad (3.11)$$

とする。これを (3.10) 式に代入すると、

$$\tau(y \text{ is } B) \geq \tau(x \text{ is } A) - \varepsilon \quad (3.12)$$

となる。すなわち、前件の真理値 $\tau(x \text{ is } A)$ が後件の真理値 $\tau(y \text{ is } B)$ の下限値となる。そこで、この下限値 $\tau(x \text{ is } A)$ を伝播させていくことによって推論を行なう。

B. 量限定子で表現される推論木における推論

(3.8) 式の前件部は量限定命題 [14, 15, 21, 22] と呼ばれ、一般には、

$$Q \ B \ X \ \text{are} \ A \quad (3.13)$$

と表わされる。ただし、

- Q : 量限定子
- X : 対象集合 $\{x_1, x_2, \dots, x_n\}$
- B : ファジィ述語
- A : ファジィ述語

である。

量限定子 Q は、対象集合 X の B という性質を持つ部分集合について、その要素が持っている特性 A についての情報を統合する。すなわち、個々の要素 $x_i \in X$ についての命題 “If x_i is B then x_i is A ” の真理値を統合して、量限定命題 “ $Q \ B \ X \ \text{are} \ A$ ” の真理値を導出することができる。量限定子 Q はその意味によってタイプ I (絶対量限定子)、II (相対量限定子) の二通りに分類される [6, 14]。量限定命題による情報の統合の方法は、量限定子のもつ意味によって変わってくる。以下にそれぞれの推論法について述べる。

タイプ I: 絶対量限定子

量限定子 Q が *about 3*, *all*, *at least 5* など絶対的な量について述べている場合、 Q は R^+ (非負の実数) または I^+ (非負の整数) 上のファジィ集合として表現することができる。例えば、量限

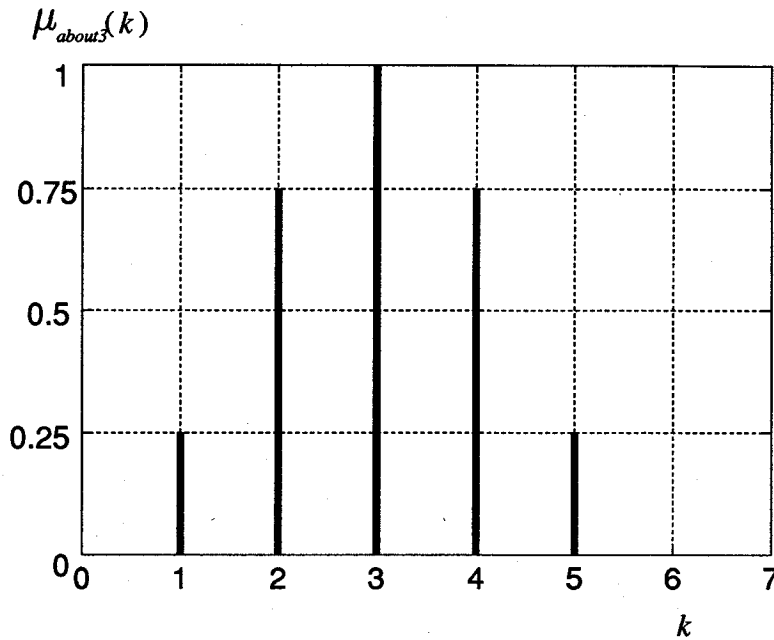


Fig. 3.4: A fuzzy set representing quantifier “about 3.”

定子 $Q = \text{about } 3$ は, Fig. 3.4 のように非負の実数上のファジィ集合として表現される. ただし, $k \in I^+$ である. このように Q をファジィ集合として定義した場合, (3.13) 式で示される量限定命題の真理値は, 次のようになると考えられる.

$$\tau(Q \text{ } B \text{ } X \text{ are } A) = \tau(n \text{ is } Q \mid n \text{ is } r) \quad (3.14)$$

$$= \mu_Q(r) \quad (3.15)$$

ただし, n は R^+ 上の変数であり, r は “ X の要素で, もし B に含まれるならば A にも含まれる” という条件を満たす要素の数を表わす. しかし, B, A はファジィ述語であるため, クリस्पな述語の場合と同じ方法で r を決定することはできない.

ここで, $C \in 2^X$ なる非ファジィ集合 C について以下の様な命題 P_C を考える.

$$P_C = \text{“} Q \text{ } Xs \text{ are } C \text{”} \quad (3.16)$$

命題 P_C の真理値 $\tau(P_C)$ は, 集合 C がクリस्पな集合であることと (3.15) 式から以下の様に求まる.

$$\tau(P_C) = \mu_Q(|C|) \quad (3.17)$$

ただし, $|C|$ は集合 C の要素数を表わすものとする. 一方, 全ての $x_i \in C$ に対して, 命題 P_i^C :

$$P_i^C = \text{“If } x_i \text{ is } B \text{ then } x_i \text{ is } A \text{”} \quad (3.18)$$

が真であるとする、(3.15)式における r の定義により $r = |C|$ となる。したがって、 X の非ファジィな部分集合 C の中で Q と条件 (3.18) を最も満たすものを C' とすると、

$$Q \text{ B } X \text{ are } A = P_{C'} \text{ and } P_1^{C'} \text{ and } P_2^{C'} \dots \tag{3.19}$$

が成立する。したがって、

$$\begin{aligned} \tau(Q \text{ B } X \text{ are } A) &= \tau(P_{C'} \text{ and } P_1^{C'} \text{ and } P_2^{C'} \dots) \\ &= \max_{C \in 2^{X_1}} \{ \tau(P_C \text{ and } P_1^C \text{ and } P_2^C \text{ and } \dots) \} \\ &= \max_{C \in 2^{X_1}} \{ T(\tau(P_C), \tau(P_1^C, P_2^C, \dots)) \} \\ &= \max_{C \in 2^{X_1}} \{ T \tau(P_C), \prod_{x_i \in C} \tau(P_i^C) \} \end{aligned} \tag{3.20}$$

と $\tau(Q \text{ B } X \text{ are } A)$ を計算できる。ただし、 T は任意の三角ノルムを示すものとする。また、命題 P_i^C の真理値は含意演算子 I を用いて以下のように表わすことができる。

$$\tau(P_i^C) = I(\tau(x \text{ is } B \mid x \text{ is } x_i), \tau(x \text{ is } A \mid x \text{ is } x_i)) \tag{3.21}$$

タイプ II : 相対量限定子

量限定子 Q が、almost, most, about half, few など相対的な量について述べている場合、 Q は区間 $[0, 1]$ 上のファジィ集合として表現される。

$$Q \text{ のメンバーシップ関数 } \mu_Q : [0, 1] \rightarrow [0, 1] \tag{3.22}$$

例えば、量限定子 $Q = \text{most}$ は、Fig. 3.5のように区間 $[0, 1]$ 上のファジィ集合として表現される。

この Q の定義によって、(3.13) 式の量限定命題の意味は次のように解釈することができる。

$$\begin{aligned} \tau(Q \text{ B } X \text{ are } A) &= \tau(n \text{ is } Q \mid n \text{ is } r) \\ &= \mu_Q(r) \end{aligned} \tag{3.23}$$

ここで、 n は $[0, 1]$ 上の変数、 r は X の要素中에서도 B に含まれるならば A にも含まれるものの数を B に含まれる要素数で割ったものを示す。

タイプ I の量限定子の場合 (3.15) 式と異なる点は、 r の定義にファジィ集合 B の要素の数 (すなわち、 B の濃度) が含まれている点だけである。ファジィ集合の濃度の定義としては、Zadeh による定義 [23]

$$|B| = \sum_{i=1}^n \mu_B(x_i) \tag{3.24}$$

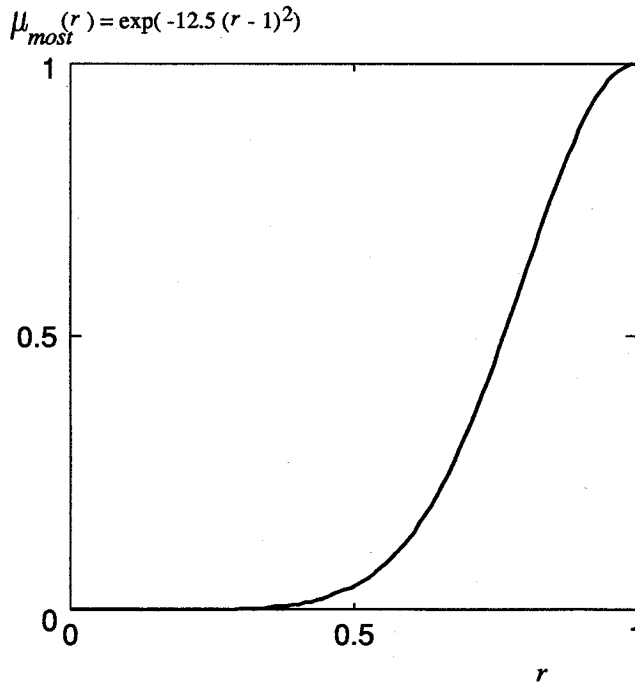


Fig. 3.5: An example of a membership function representing *most*.

を用いる。これに応じて、タイプ I の場合の (3.17) 式が

$$\tau(P_C) = \mu_Q(|C|) \quad (3.25)$$

ただし、

$$\begin{aligned} r &= \min(|C|/|B|, 1) \\ &= \min(|C|/\sum \mu_B(x_i), 1) \end{aligned} \quad (3.26)$$

と変更される。これ以外は、タイプ I と同様である。量限定命題を用いて表現されたルール例を以下に示す。

/* 全体的なスケジュールの評価ルール */

ルール 1301:

If ほとんど (重要な観点 (機械稼働率という観点での評価値 is 高い),
 (機械の密集度という観点での評価値 is 高い),
 (ロット納期余裕という観点での評価値 is 高い),
 (待ち時間という観点での評価値 is 高い))

then (スケジュール評価値 is 高い).

言語的表現: 重要な観点での評価値がほとんどが高いならば, スケジュール評価値は高い.

/* 機械稼働率という観点からのスケジュール評価ルール */

ルール 0501:

If ほとんど (重要な機械稼働率 (機械 m1 の稼働率 is 適切),
 (機械 m2 の稼働率 is 適切),
 (機械 m3 の稼働率 is 適切),
 (機械 m4 の稼働率 is 適切),
 (機械 m5 の稼働率 is 適切))

then (機械稼働率という観点での評価値 is 高い).

言語的表現: 機械稼働率のうち重要なものがほとんど適切ならば, 機械稼働率という観点での評価値は高い.

ルール 0502:

If (機械 m1 の稼働率 is 適切) and (機械 m2 の稼働率 is 適切) and
 (機械 m3 の稼働率 is 適切) and (機械 m4 の稼働率 is 適切) and
 (機械 m5 の稼働率 is 適切)

then (機械稼働率という観点での評価値 is 高い).

言語的表現: 「機械 m1 の稼働率 is 高い」かつ... かつ「機械 m2 の稼働率 is 高い」ならば 機械稼働率という観点での評価値は高い.

/* 部品待ち時間という観点からのスケジュール評価ルール */

/* 部品待ち時間 = 部品の wip 時間 - 被作業時間 */

ルール 0801:

If ほとんど (重要な部品待ち時間
 (部品待ち時間 (ロードされている部品)) is 短い)

then (部品待ち時間という観点での評価値 is 高い).

言語的表現: ロードされている部品のうち, 重要なものはほとんどが待ち時間が短い.

3.4 スケジューリング方針決定モジュールにおける推論

スケジューリング方針決定モジュールは、スケジュール評価モジュールにおける評価状態を入力とし、各評価項目を改善することはどの程度重要であるかを表わす数値(すなわち、スケジューリング方針)を推論する。そして、このスケジューリング方針に基づいてディスパッチングモジュール中の優先規則の重要度を求めて出力する。

スケジューリング方針決定のためには、現時点のスケジュールの問題点を見つけ出す必要がある。すなわち、スケジュールの評価の悪化に対して、どの評価項目の影響が大きいかを同定しなければならない。悪影響を及ぼしている項目を改善することがスケジューリング方針となる。以下に、各評価項目が、全体的な評価にどの程度影響を与えているかを求める手続きを示す。

現在のスケジュールの問題点を同定するためには、推論木の根の真理値に各子孫のノードがどの程度影響しているかを同定する必要がある。本節では、推論木上のあるノード S_0 において、その子孫にあたるノード S_1 上の命題 S_1 の真理値が、命題 S_0 の真理値に対して、どの程度影響しているかを表わす影響度 $C(S_1 \rightarrow S_0) \in [0, 1]$ を導入する。影響度 $C(S_1 \rightarrow S_0) = 0$ のとき、 S_1 の真理値が変わっても S_0 の真理値は変わらないことを示し、 $C(S_1 \rightarrow S_0) = 1$ のとき S_1 の真理値が変わると S_0 の真理値は大きく変わることを示している。

以下に、推論木上の全てのノードの真理値が既知の時、各ノードの影響度の推定値を求める手続きについて述べる。

3.4.1 OR 演算子における親に対する子の影響度

n 個の子ノード (S_1, S_2, \dots, S_n) が or 結合されて、親ノード S_0 を指示している場合を考える。ここで、ノード S_i が表わす命題の真理値を $\tau(S_i)$ とする。

or 結合された子ノードは、真理値の大きい順に親ノードの真理値に影響していると考えられる。したがって、子ノードの真理値そのものによって、影響の度合を表現することができる。この場合の子ノードから親ノードへの影響度を次のように定義する。

$$C(S_i \rightarrow S_0) = \tau(S_i) / \tau_{max}, \quad i = 1, \dots, n \quad (3.27)$$

ただし、 $\tau_{max} = \max\{\tau(S_1), \tau(S_2), \dots, \tau(S_n)\}$ とする。このように、子ノードの真理値を最大のもので割ってやることにより、一番影響を与えている子ノードの影響度が 1 となる。

3.4.2 AND 演算子における親に対する子の影響度

n 個の子ノード (S_1, S_2, \dots, S_n) が and 結合されて、親ノード S_0 を指示している場合を考える。

and 結合された子ノードは、真理値の小さい順に親ノードの真理値に影響していると考えられる。したがって、1 から子ノードの真理値を引いた値によって、影響の度合を表現することがで

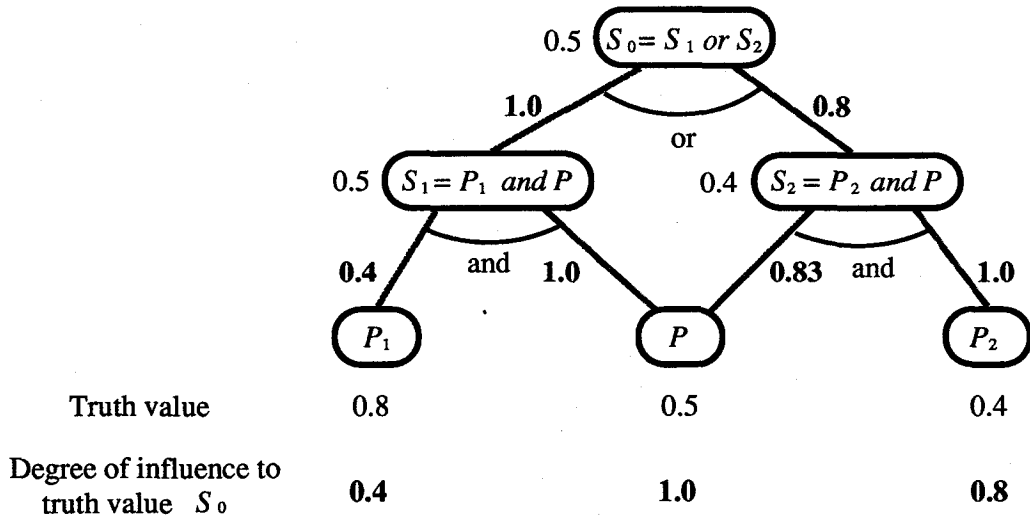


Fig. 3.6: A numerical example of degree of influence [8].

きる。この場合の子ノードから親ノードへの影響度を次のように定義する。

$$C(S_i \rightarrow S_0) = (1 - \tau(S_i)) / (1 - \tau_{min}), \quad i = 1, \dots, n \quad (3.28)$$

ただし, $\tau_{min} = \min\{\tau(S_1), \tau(S_2), \dots, \tau(S_n)\}$ とする。

3.4.3 量限定子における親に対する子の影響度

(3.19) 式によれば, 量限定命題は等価な and/or で表現された命題で表現することが可能である。したがって, 量限定子における影響度は等価な and/or 推論木に変換して求めることができる。

3.4.4 先祖のノードに対する影響度

直接の親に対する影響度は, (3.27), (3.28) 式によって求められる。ここでは, 数レベル隔たった先祖に対する影響度を求める。ノード S_0 からノード S_n まで1つのルート ($S_0 \rightarrow S_1 \rightarrow \dots \rightarrow S_n$) が存在する場合について考える。このとき, S_n の S_0 に対する影響度 $C(S_n \rightarrow S_0)$ は次のように定義される。

$$\begin{aligned} C(S_n \rightarrow S_0) &= C(S_n \rightarrow S_{n-1} \rightarrow \dots \rightarrow S_1 \rightarrow S_0) \\ &= C(S_n \rightarrow S_{n-1}) \cdot C(S_{n-1} \rightarrow S_{n-2}) \cdot \dots \cdot C(S_1 \rightarrow S_0) \end{aligned} \quad (3.29)$$

ノード S_0 からノード S_n まで複数のルートが存在する場合は, 各ルートに対して計算される影響度の最大のを S_0 への影響度とする。Fig. 3.6に影響度計算の例を示す。

Dispatching rules

Rule-1

If process O uses resource M frequently,
then process O has a priority
⋮

Rule-2

If tardiness of job P is large,
then job P has a priority
⋮

Rule-3

⋮
⋮
⋮

Processing information network

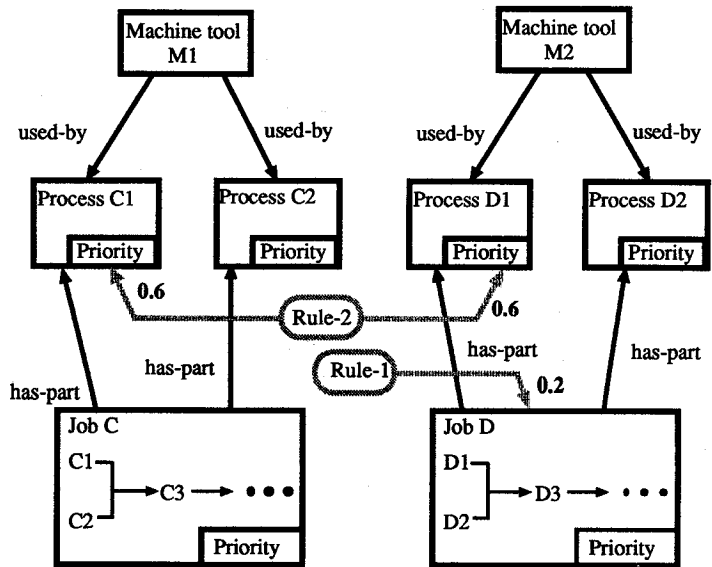


Fig. 3.7: Knowledges in dispatching module [8].

合成命題 $S_0 = (P_1 \text{ and } P) \text{ or } (P_2 \text{ and } P)$ の真理値が 0.5 であることに対して、命題 P_1, P, P_2 の影響度はそれぞれ 0.4, 1.0, 0.8 である。すなわち、命題 S_0 の真理値の改善に対しては、命題 P, P_2, P_1 の順に影響する。

以上のようにして求められた影響度は、以下に示す形式のルール群に適用され、ディスパッチングモジュールの優先規則に割り付けられる。

If スケジュールの問題点 P の影響が大きい
then 優先規則 R を重要度 C で対象 O に適用

以下に例を示す。

If 問題点：“機械 m の稼働率が低い” の影響が大きい
then 優先規則 1 を機械 m に対して重要度 1.0 で適用

3.5 ディスパッチングモジュールにおける推論

ディスパッチングモジュールは Fig. 3.7 に示すように部品がどの機械でどのような順序で加工されるかを意味ネットワークで表現した加工情報ネットワークと、ファジイ述語を用いて記述されたディスパッチングルール群から構成される。Fig. 3.7 において、“ジョブ” は部品の種類を示し、“作業 (Process)” は部品加工における 1 つの工程を示す。ディスパッチングモジュールでは次の

ような手順で推論を行ない、次に加工すべき部品または次に搬送すべき機械の選択を行なう。それぞれのディスパッチングルールは、改善する評価項目、スケジュール方針決定モジュールで決定される重みの2つの属性をもつ。

- (1) それぞれのディスパッチングルールの後件部に3.3節で述べた推論法を用いて真理値を与える。
- (2) それぞれの真理値にスケジューリング方針決定モジュールから与えられた重みを掛け、優先度を求める。
- (3) (2)で求めた値をそれぞれの後件部に対応する加工情報ネットワークのノードに与える。
- (4) 優先度が与えられたノードが“ジョブ”または“機械”を示すノードであれば、そのノードから到達可能な“作業”を表わすノードに優先度を伝播させる。
- (5) “作業”を示すノードに割り当てられた優先度と他のノードから伝播してきた優先度の総和をとり、その“作業”の優先度とする。
- (6) 最も優先度の大きい“作業”を選択し、その“作業”に対応する部品または機械を結論としてシミュレータに伝える。

以下にディスパッチングルールの例を示す。

If 作業 i の機械 m に対する使用量が多い
then 作業 i の優先度を上昇させる。

Fig. 3.7の例では作業 D1 の優先度が 0.8 で最高となり作業 D1 が選択される。

3.6 システム構成

本システムは、FMS シミュレータ、スケジュール評価モジュール、スケジュールリング方針決定モジュールおよびディスパッチングモジュールからなる推論部、およびシミュレータと推論部とのインタフェースから構成される。FMS シミュレータは汎用シミュレーション言語 SLAM II [24]、推論部は LISP、インタフェースは C 言語を用いて記述し、Sun ワークステーション上にインプリメントされている。このようにして今回作成した知識ベーススケジューリングシステムの構成を Fig. 3.8 に示す。

Fig. 3.8において、ルールベース 1 はスケジュール評価モジュールにおいて用いられるもので、ユーザがメンバーシップ関数を調節することによりスケジュールリング目標を変更することができる。ルールベース 2、ルールベース 3 はそれぞれスケジュールリング方針決定モジュール、ディスパッチングモジュールにおいて使用されるルールベースであり、システムのメンテナンスを行なうエキスパートによって保守が行なわれる。

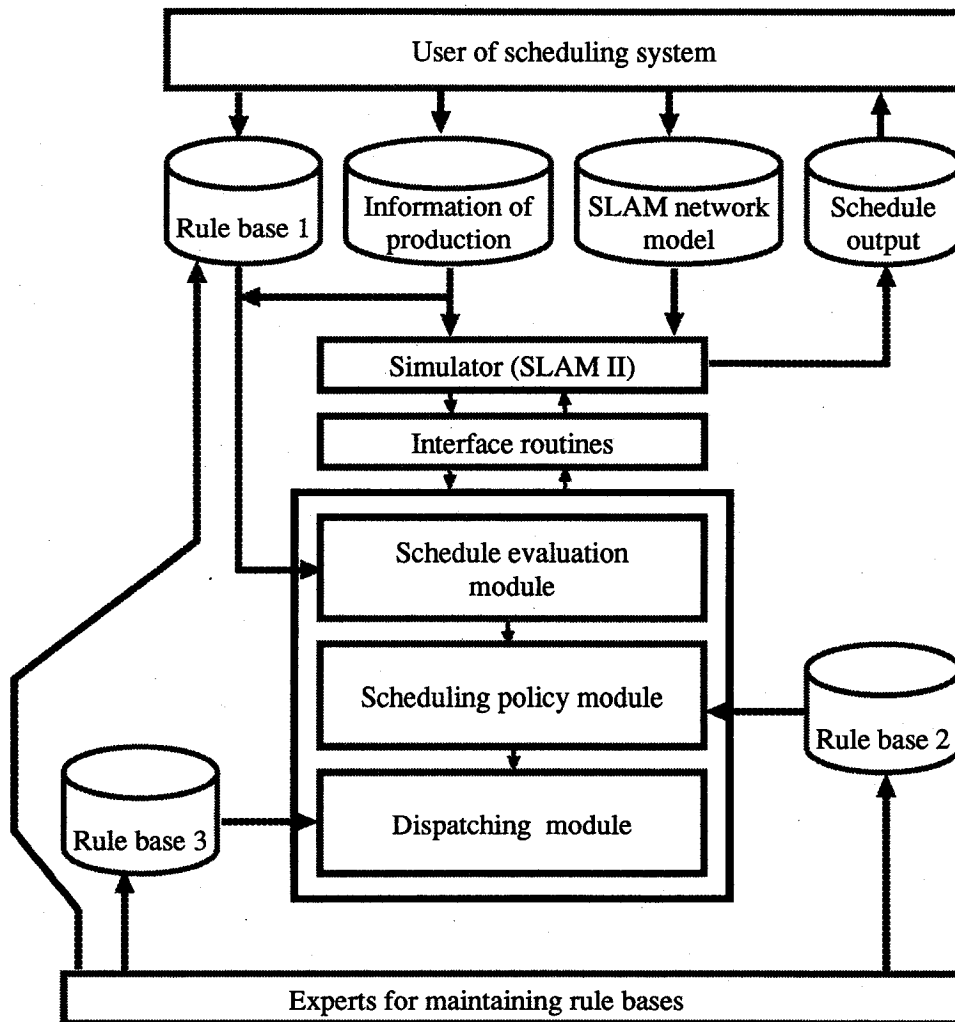


Fig. 3.8: System configuration [8].

3.7 数値例

ここでは、5 台の機械、1 台の搬送車とロード/アンロードマシンで構成された FMS を考える。この FMS において Table 3.1, Table 3.2 に示すような加工情報を持つ部品の加工を考える。各部品 A, B, C は工程の順番に従って加工が行なわれるものとする。Table 3.1 中の括弧内の数字はそれぞれの機械における加工時間を示す。各部品の投入個数は 8 であるものとする。以上のような加工条件を用い、次の 2 つの納期およびスケジューリング戦略でスケジュールを生成した。

納期 1 部品 A-70, 部品 B-55, 部品 C-70

納期 2 部品 A-70, 部品 B-40, 部品 C-70

戦略 1 納期を満たすため、納期余裕という観点を重視する。また、機械の負荷を平滑化して総所用時間を短くすることが必要であるため、稼働率と密集度も重要視する。特に他の機械と

Table 3.1: Processing informations [8].

| | Process 1 | Process 2 | Process 3 |
|-------|----------------|----------------|-----------|
| Job A | M1(6) or M2(5) | M3(6) or M4(4) | – |
| Job B | M1(4) or M2(3) | M3(3) or M4(2) | M5(3) |
| Job C | M3(5) or M4(4) | M5(4) | – |

Table 3.2: Transport time of jobs [8].

| Source | Destination | | | | | |
|--------|-------------|-----|-----|-----|-----|-----|
| | M1 | M2 | M3 | M4 | M5 | L/U |
| M1 | – | – | 0.4 | 0.6 | – | – |
| M2 | – | – | 0.2 | 0.4 | – | – |
| M3 | – | – | – | – | 0.4 | 0.6 |
| M4 | – | – | – | – | 0.2 | 0.4 |
| M5 | – | – | – | – | – | 0.2 |
| L/U | 0.2 | 0.4 | 0.6 | 0.8 | – | – |

比べて負荷量の多い M5 に重みをおく。部品待ち時間についてはあまり重視しない。

戦略2 戦略1に加えて M1 の稼働率を抑えるという要求を付加する。すなわち、戦略1では機械の稼働率が低くなるほど評価が悪くなるよう設定していたが、M1 の稼働率の評価を稼働率が高くなればなるほど悪くなるように設定する。さらに M1 の稼働率を抑える代償として M5 の密集度の重要性を相対的に下げる。

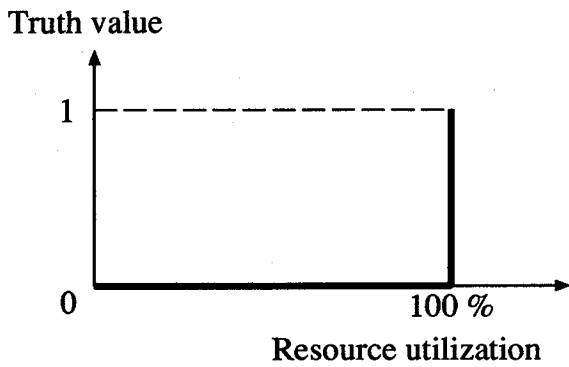
Fig. 3.9 (a), (b) にそれぞれ戦略1, 戦略2における“M1 の稼働率が高い”というファジィ述語の“高い”に対するメンバーシップ関数を示す。(a)での設定は、スケジュール評価モジュール中の

If 機械稼働率が高い機械が存在する

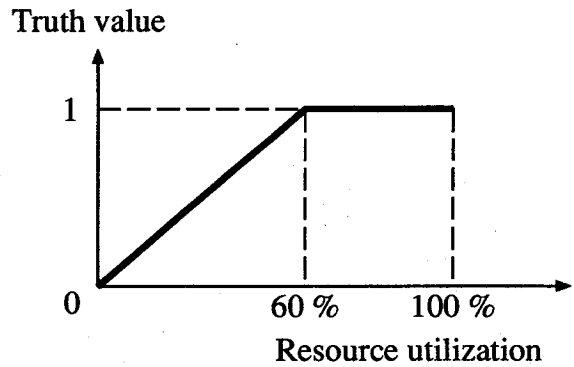
then 機械稼働率という観点からのスケジュールの評価が低い。

というルールが発火しないような設定になっているのに対して、(b)では、上記のルールが発火しやすいような設定になっている。これは、例えば M1 が故障しがちであり、一度故障すると長い修理時間を要するため、使用頻度を落したいという場合に相当する。

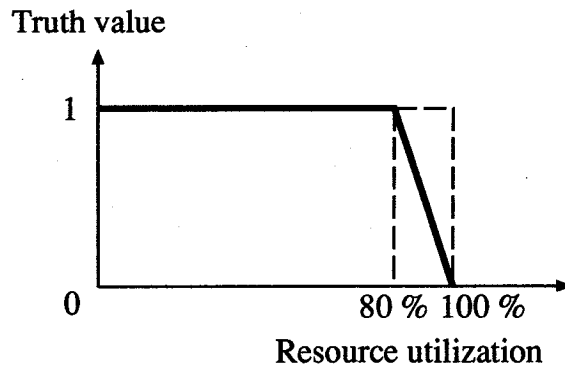
Fig. 3.10 (a), (b) および **Fig. 3.11 (a), (b)** にそれぞれスケジューリング戦略1 および2に基づいて生成したスケジューリング例を示す。Fig. 3.10 (b), Fig. 3.11 (b) のスケジュールでは部品 B に僅かの納期遅れが生じている。しかし、Table 3.3に示すように、Fig. 3.11 (a), (b) では、Fig. 3.10 (a), (b) に比べて M1 の稼働率が低く抑えられて、M2 の稼働率が上昇していることが分かる。また、その他の機械の稼働率は大きく変動していないことが分かる。加えて M3 の稼働率を改善したい場合は、“M3 の稼働率小さい”の“小さい”に対するメンバーシップ関数を、Fig. 3.9 に示すように変更する必要がある。



(a) A membership function of "large" for utilization of M1 on Strategy 1

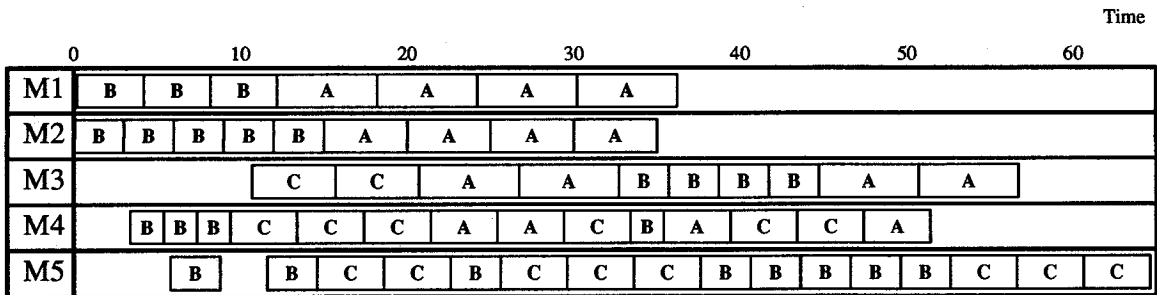


(b) A membership function of "large" for utilization of M1 on Strategy 2

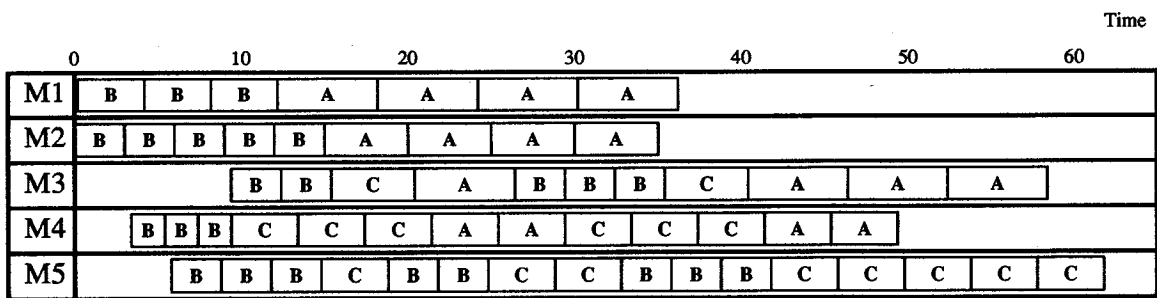


(c) A membership function of "small" for utilization of M3

Fig. 3.9: Examples of membership functions [8].

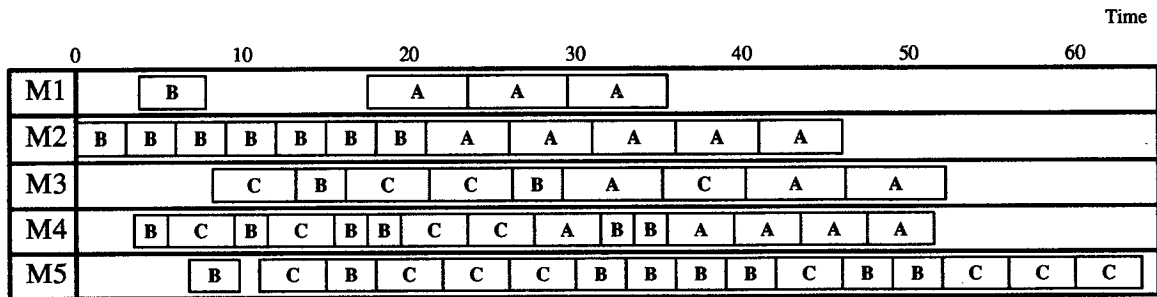


(a) In case of due date 1

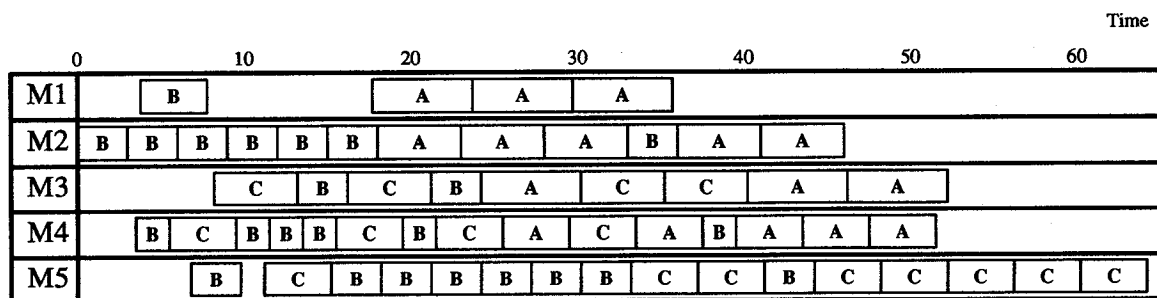


(b) In case of due date 2

Fig. 3.10: Example of schedules based on Strategy 1 [8].



(a) In case of due date 1



(b) In case of due date 2

Fig. 3.11: Example of schedules based on Strategy 2 [8].

Table 3.3: Statistics of the schedules [8].

| | Average resource utilization | | | |
|----|------------------------------|------------|------------|------------|
| | Strategy 1 | | Strategy 2 | |
| | Due date 1 | Due date 2 | Due date 1 | Due date 2 |
| M1 | 54.9 | 57.3 | 33.9 | 33.7 |
| M2 | 53.4 | 55.7 | 70.8 | 70.6 |
| M3 | 70.1 | 78.0 | 67.7 | 67.5 |
| M4 | 73.2 | 73.3 | 73.9 | 73.6 |
| M5 | 85.4 | 89.2 | 86.2 | 85.9 |

3.8 むすび

本章では、シミュレーションを用いたスケジューリングにおいて、競合した部品の選択のための知識ベースを、スケジュール評価モジュール、スケジューリング方針決定モジュールおよびディスプレイパッチングモジュールに分割し、それぞれにおける知識をファジィ・ルールで記述することにより、“できるだけ機械 M1 を使用しないで納期を満たす”といった多様なスケジューリング目的に対応可能なスケジューリングシステムを構成することを試みた。

このモデルに対する試作システムを構築し、数値実験を行なった結果、次のような結論を得た。

- (1) 評価モジュール中のファジィ述語の変更に応じて多様なスケジューリングを行なうことができる。
- (2) 本スケジューラのチューニングは容易に行なえる。すなわち、ユーザは評価項目の値についての定性的な判断に基づいて、評価モジュールのファジィ述語を調節するだけでよく、システム内における処理を熟知している必要はない。
- (3) スケジューリング目的の記述を言語的に行なうことができ、より現実的で複雑なスケジューリングが可能である。

今後の課題としては次のようなことが挙げられる。

- (1) 現在は、機械稼働率、機械利用の密集度、ロット納期余裕、部品待ち時間の4つの評価項目のみを取り扱っているが、スケジューラの良否は評価項目の選択に依存しているので、スケジューラ中の評価項目を充実させる必要がある。
- (2) ユーザはスケジューラに用意されている評価項目の値に対するファジィ述語、「高すぎる」、「低すぎる」などを設定することによってスケジューリング目的を表現するが、各評価項目間にトレードオフの関係があった場合、メンバーシップ関数の同定が困難になると予想される。このようなことを避けるためには、評価項目間の関係を数値実験により同定し、スケジューリング目標設定の際の指針を明確化する必要がある。

- (3) スケジューリングの評価において現時点の状態のみを評価するのではなく、先読みを行ない未来の状態も評価の対象としてスケジューリングの質を向上させる必要がある。現在は、評価項目「密集度」においてのみ先読みの要素が含まれているだけである。
- (4) シミュレータにおける制約のために、スケジューラは開始可能な作業が存在する場合は必ず1つの作業を実行しなければならず、「待ち」という意思決定が行なえない。このため、スケジューラが実現できるスケジューリングがかなり制約されている。そこで、より多様なスケジューリングを実現するためにはシミュレータの機能を強化する必要がある。
- (5) スケジューリング結果、ルール・チューニング情報などに対するユーザインターフェイスを充実する。
- (6) 実システムへの適用実験を行なう。

第4章 フィルター付きビーム探索法における ヒューリスティクスの自動合成

4.1 まえがき

第1章でも述べたように、大規模な問題を対象としたジョブショップ・スケジューリング問題は、最適解を求めることが困難なため、A*アルゴリズム [1-3] や隣接法 (Neighborhood Method) [4] およびビーム探索法 [4] などの発見的方法 (Heuristic Method) を用いた近似解法が用いられる場合が多い。

一般にこれらのジョブショップ・スケジューリング問題のための近似解法によって得られる解の精度は、解法の過程で用いられる対象領域に依存した問題固有の知識 [5-7, 3](Heuristic Knowledge: ヒューリスティクス) に大きく依存する場合がほとんどである。このような知識は、通常個々の問題領域に対して領域専門家が持っているものである。しかし、問題構造が複雑化し、様々な状況が存在する個々の現実環境に対して、それぞれ問題領域理論を作り、そこから情報を得るのは非常に困難である。そこで、種々の問題に対して柔軟に、しかもある程度精度の良いヒューリスティクスを自動的に合成することが期待される。

本章では、ヒューリスティック探索法的一种であるビーム探索法を改良したフィルター付きビーム探索法 [4, 8] におけるヒューリスティクスをシステム自身が自動的に獲得し、それを有効に利用する方法を提案し、搬送車スケジューリング問題に適用することにより有効性を議論する。フィルター付きビーム探索法は、ビーム探索法において多大の計算時間を必要とするグローバルな情報に基づく探索の数を、計算時間が短くて済むローカルな情報に基づいた探索によって絞り込むことにより探索空間を狭める方法であるが、このローカルな情報を領域専門家の知識によらず、自動的に構成することを試みる。さらに、分枝限界法によって得られた最適解と比較することにより本手法によって得られたヒューリスティックな知識の精度の評価を行なう [9]。

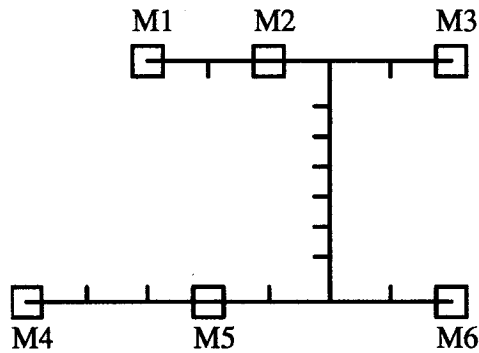


Fig. 4.1: A transport line [9].

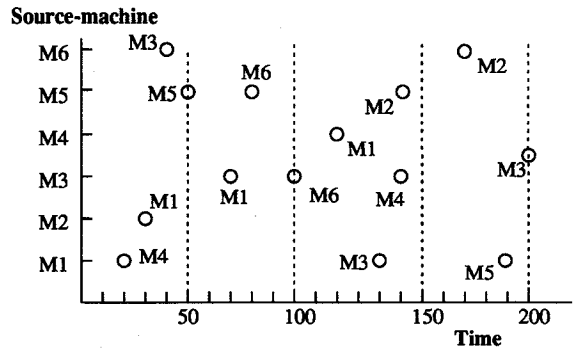


Fig. 4.2: The distribution of requirements [9].

4.2 搬送車スケジューリングのモデル化

製品の製造過程では、工場の各機械に部品などの資材を運搬するため搬送車が用いられている。搬送車のスケジューリングは、搬送要求が出された順に行なわれており、ある時間間隔で最適スケジュールになっているとは限らない。この搬送システムの性能を評価するために一定時間における最適スケジュールを求める必要が生じてくる。搬送車が一台であるような単純な場合には、人間がその問題構造に注目した単純な観点からヒューリスティクスを獲得することができる。しかし、搬送車が同一軌道に複数台あるような実際の現場に近い状況では、一台の時のようにヒューリスティクスを簡単に抽出することは困難である。

本章で取り扱う搬送車スケジューリング問題における搬送車の台数は2台とする。各搬送車は、工場内の1ゾーンにおいて Fig. 4.1のような軌道上を移動し、各ステーション (M1~M6) 上で部品の積み降ろしを行なう。各工程での搬送要求時間間隔は、搬送回数よりある確率分布に従うが、ここでは確定的であるものとし、Fig. 4.2で与えられるものとする。ただし、縦軸は搬送元のステーション名、横軸は搬送要求の発生する時刻、および図中の M1~M6 は搬送先を示すものとする。また、各ステーション間の部品の搬送時間及び回送時間は、ステーション間の軌道上距離に比例するものとする。

本章で扱う搬送車スケジューリング問題の設定において、以下のような仮定および制約条件おくこととする。

仮定：

- 2台の搬送車の走行速度は搬送時、回送時ともに V とする。搬送車は、各ステーションで回避行動をとることができる。

- 搬送車の初期状態は、時刻 0 でステーション M1 とその退避所に各々 1 台ずつあるものとする。
- 部品のロットサイズは、すべて同一であるとする。
- 部品のローディングおよびアンローディングに要する時間は、搬送車及びステーションの種類によらず一定であり、搬送時間に含めるものとする。

制約条件：

- 搬送車同士は、距離 D 以上接近してはならない。
- 一方の搬送車は軌道上で他方を追い越してはならない。
- 衝突回避のための退避及び待機行動を行なってはならない。

スケジューリングにおける目的関数(評価尺度)は、搬送要求時刻から実際に搬送を開始する時刻までの部品の待ち時間の総計とする。これを定式化すると以下ようになる。

$$\text{minimize } \sum_i w_i \quad (4.1)$$

ただし、

$$w_i = s_i - R_i \quad (s_i \geq R_i) \quad (4.2)$$

- w_i : 搬送要求 i の待ち時間
 s_i : 搬送要求 i の実際の搬送開始時刻
 R_i : 搬送要求 i の搬送要求時刻

4.3 フィルター付きビーム探索法

4.3.1 スケジュール探索のための優先度評価関数

一般にスケジュール探索問題は、全スケジュール状態の集合である状態空間の中から選択された解候補(ノード)から、解候補をいくつか選択し、それらの解候補から次の解候補を選択するという手順を繰り返すことによりスケジュールが生成される。

スケジュール探索の近似解法においては、解候補を選択する際に、その解候補がどの程度最適なスケジュールに近いかを示す優先度評価関数を用いる場合が多い。優先度評価関数には、大別して二つある。一つは、探索空間の大局的な視点に基づいたグローバル優先度評価関数であり、いま一つは、探索空間の局所的な視点に基づいたローカル優先度評価関数または優先度ヒューリスティクスと呼ばれるものである。

ローカル優先度評価関数は、現在のノード状態を次の状態に遷移させるときに、現在までのノード状態と次のノードの状態から得られる情報を基にして、その問題に対して得られている領域理論や経験的に得られた知識を用いて構成される。

一方、グローバル優先度評価関数は、多くの場合ローカル優先度評価関数を評価関数とするプロブサーチ（ローカル優先度評価関数を優先規則とする縦型探索）を用いて作成される。あるノード n のグローバル優先度評価関数値を計算するには、まずノード n によって代表される状態（部分的スケジュール状態）から次のレベルのノードをすべて生成させ、それらのノードのローカル優先度評価関数値を各々計算する。計算されたローカル優先度評価関数値に基づいて最も評価値の高いノードを一つだけ選び出す。以下、同様な操作を繰り返し、最終的に到達したノード状態（完全なスケジュール状態）から問題に与えられている目的関数値を計算し、その値をグローバル優先度評価関数値と定義する。したがって、グローバル優先度評価関数は、ローカル優先度評価関数と目的関数によって一意に求められる。

これら二つの評価関数の特徴としては、ローカル優先度評価関数は、探索空間の局所的な情報しか見ていないので、探索に要する実行時間や計算コストを低くすることができるが、これによって得られる最終的なスケジュールの精度は、ローカル優先度評価関数が局所的情報から、どれだけ大局的推定をすることができるか、という優先度評価関数の精度に大きく依存する。

一方、グローバル評価関数は、一度プロブサーチを行ない、得られた解（スケジュール）の目的関数値を計算することによって求めているので、大局的な情報を含んでおり、ローカル優先度評価関数よりも、一般的に最適解を推定する精度が高いと予想される。しかし、現ノードからゴール・ノードまで一度、探索を行なわなければならないので、ローカル優先度評価関数に比べて、探索に要する計算時間やコストがかなり高くなるという欠点がある。

ローカル優先度評価関数のスケジューリング問題に対する具体的な例としては、フローショップ問題に対して Ow らが用いた関数 [4] がある。この関数は、ボトルネック機械がジョブから解放された時刻 t が与えられると以下のように構成される。

$$P_i(t) = \{w_i/p_{im} + S_i^+ \times \exp\{(-\max(0, d_i - f_i) + h(S_i^+ + \phi))/(k(p^- + \phi))\}\} \quad (4.3)$$

ここで、 p_{im} はボトルネック機械 m におけるジョブ i の加工時間、 p^- は全てのジョブのボトルネック機械における加工時間の平均、 S_i^+ はジョブ i によってボトルネック機械で強要された非負の遊休時間、 ϕ は全てのジョブにわたる強要された遊休時間の平均、 h は資源によるパラメーター、 k は進行方向パラメーター、 w_i はジョブ i が納期よりも遅れた場合の加重、 d_i はジョブ i の納期、 f_i はジョブ i の完了時刻を表わす。

4.3.2 フィルター付きビーム探索法のアルゴリズム

フィルター付きビーム探索法 [4] の特徴は、最適解を推定する精度は低いが、実行時間が短くて済み、計算コストが安いという利点を持つローカル優先度評価関数を用いてノードをある程度の数に絞っておき、残ったノードに対して、計算コストは高いが、最適解に対する推定精度が高いグローバル優先度評価関数を用いて更にノードの選択を行なうことによって、計算コストを押さえながら比較的推定精度の高い解を得ることができるということがあげられる。

フィルター付きビーム探索では、まず探索木中の一つのレベルで展開されているすべてのノードに対して、ローカル優先度評価関数値を計算し、この値をもとに、最適解を含む可能性の低いノードを切り捨て、フィルター幅 k に等しい数のノードだけを残す（フィルター操作）。次に、フィルター操作の後に残った k 個のノードに対して、グローバル優先度評価関数を用いて各ノードの最適解を含んでいる可能性に対する評価値を計算する。この評価値に基づいて、ビーム幅 w に等しい数のノードだけを残し、それ以外のノードは切り捨てる。ビーム操作の後に残ったノードを全て展開することによって、次のレベルを構成し、以下同様のフィルター操作とビーム操作を繰り返し、最終的に残ったビーム幅 w に等しい数の可能解（許容スケジュール）の中から、目的関数値が最良のものをこの探索によって得られた解（スケジュール）とする。すなわち、フィルター付きビーム探索ではフィルター幅 k とビーム幅 w を操作することによって、実行時間や計算コストと解の推定精度を操作することができる。以下に、フィルター付きビーム探索のアルゴリズムを示す。

フィルター付きビーム探索法のアルゴリズム:

- Step 1.** スタートノードを *OPEN* と呼ばれる未展開ノードのリストに入れ、スタートノードのグローバル優先度評価関数を計算する。
- Step 2.** もし、*OPEN* が空ならば、失敗として停止する。
- Step 3.** 展開されたノードのうち、グローバル優先度評価関数値が最も良いもの w 個を *OPEN* から選び出す。 (w : ビーム幅) これらのノードを *OPEN* から *CLOSED* と呼ばれる展開されたノードのリストに移す。 *OPEN* に残ったノードは枝刈りすることにより放棄する。
- Step 4.** もし、 w 個のノードのうちのどれかにゴールノードがあれば、成功として停止する。
- Step 5.** w 個のノードを展開し、各ノードに対する子ノードを生成させる。
- Step 6.** 生成された子ノードの各々に対してローカル優先度評価関数を計算し、その中で、最良のもの k 個を残す。 (k : フィルター幅)
- Step 7.** 残った k 個の子ノードに対して、各々グローバル評価関数値を計算し、*OPEN* に追加する。
- Step 8.** Step 2 に戻る。

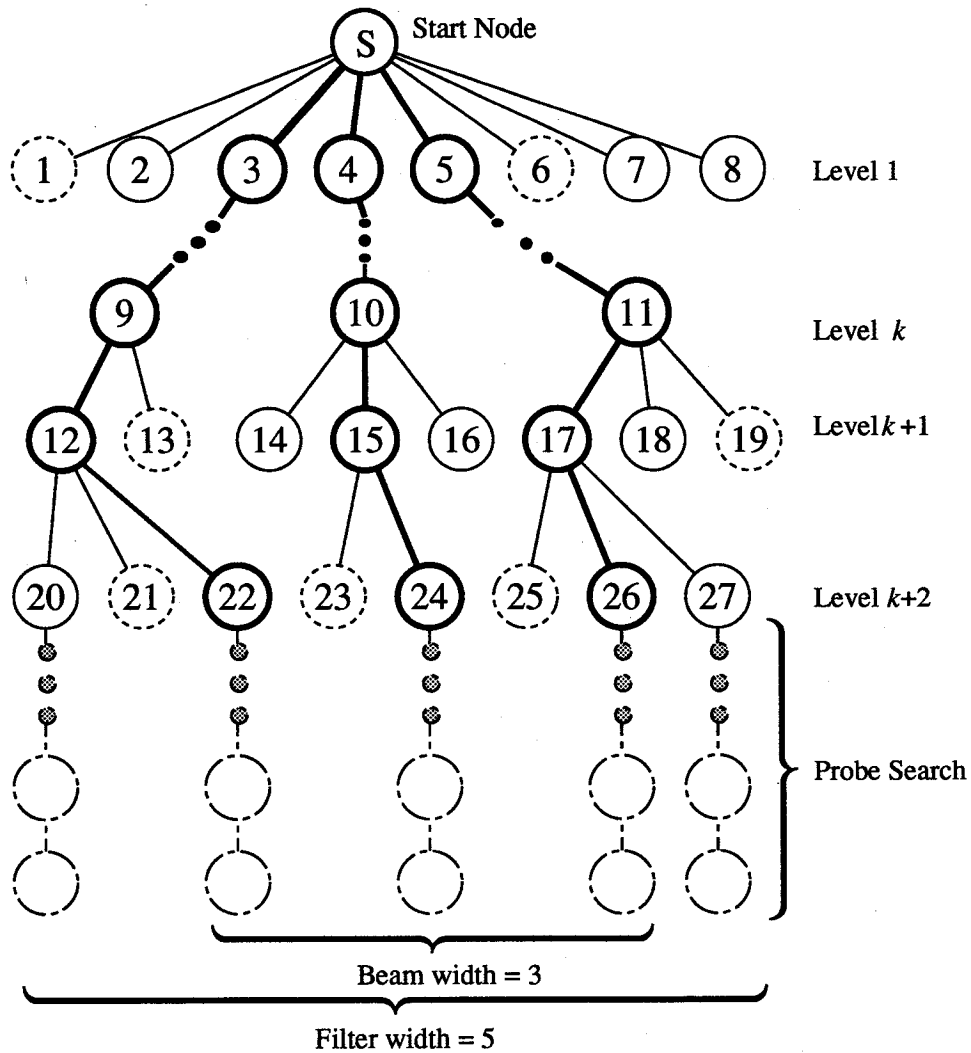


Fig. 4.3: An example of search tree in filtered beam search method [9].

フィルター幅 k に従って、最良の子ノードを選ぶ場合には、各親ノードから生成されたノードの中から、各々最良のノードを一つだけ選んでいくようにする。それは、ある一つの親ノードから生成されたノードのローカル優先度評価関数値と、別の親ノードから生成されたノードのローカル優先度評価関数値とは、親ノードが異なるので、単純に比較することはできないためである。例えば、フィルター幅が $k=5$ で、各親ノードに8つのノードがあるとすると、そのレベルにある40個のノードすべての中から最良のノード5個を選ぶのではなく、各親ノードに継っているノード8個の中から最良のノード1個を選ぶようにしなければならない。フィルター付きビーム探索法における探索木の例を Fig. 4.3に示す。

4.3.3 フィルター付きビーム探索法の問題点

ローカル優先度評価関数の精度が悪いと、良好な解を得るためには、ビーム幅、フィルター幅をかなり大きく設定しなければならず、ほとんど全数探索に近い計算時間を費やしてしまう。従って、できるだけ精度の良いローカル優先度評価関数を選定することがこの手法における鍵となる。しかし、対象とするモデルの構造が複雑であると、精度の良いローカル優先度評価関数を得ることは困難である。

4.4 ローカル優先度評価関数の合成

本節では、フィルター付きビーム探索におけるローカル優先度評価関数を、以下の手順に従って合成する。まず、システムに入力されたスケジューリング問題をいくつかのパターンに分類し、そのパターンに対し、過去に合成されたローカル優先度評価関数があればそれをローカル優先度評価関数の初期値として用い、過去に合成されたものがない場合はひとまず目的関数に含まれる評価項目をもとにして合成する。次に合成されたローカル優先度評価関数を用いてスケジュールを生成し、事前に設定されたスケジュール目的に合致した結果であるか否かを調べ、合致していなければローカル優先度評価関数を修正して再びスケジュールを生成する。スケジュール目的に合致していれば得られたスケジュールを結果として出力し、ローカル優先度評価関数をヒューリスティックベース [5,6] に蓄積する。システム構成を Fig. 4.4に示す。

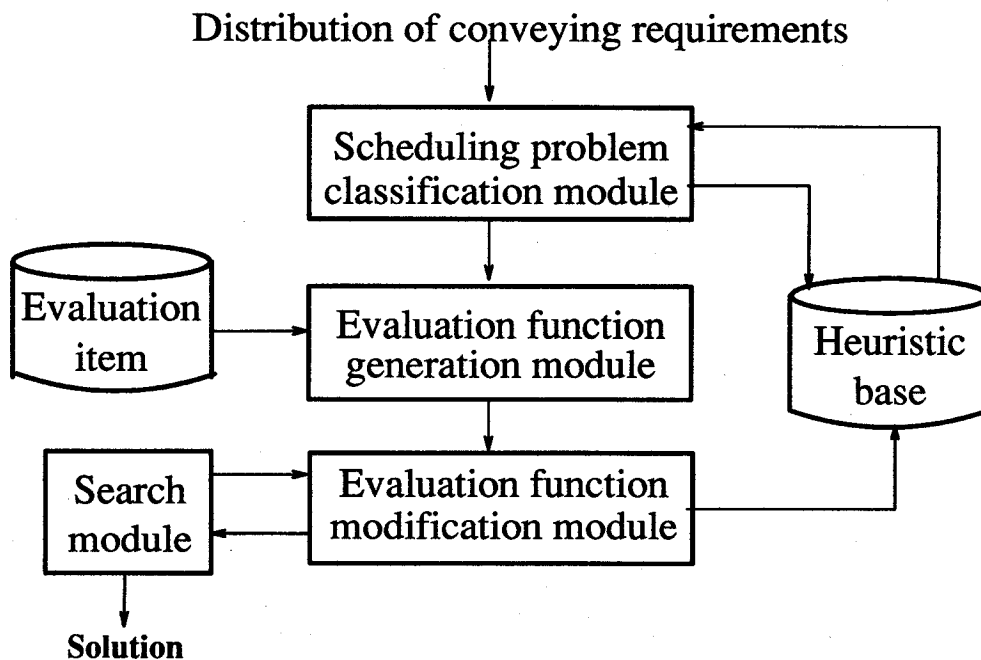


Fig. 4.4: System configuration [9].

| Out-of-service distance | Transporting distance | Task density | | |
|-------------------------|-----------------------|--------------|--------|------|
| low | low | A | J | S |
| low | medium | B | K | T |
| low | high | C | K | U |
| medium | low | D | M | V |
| medium | medium | E | N | W |
| medium | high | F | O | X |
| high | low | G | P | Y |
| high | medium | H | Q | Z |
| high | high | I | R | a |
| | | | medium | |
| | | | low | high |

Fig. 4.5: An example of data classification matrix of task [9].

4.4.1 入力データ分類モジュール

スケジューリング問題を表わすタスクデータを入力し、データの持つ属性(タスクの密集度、搬送距離の平均、回送距離の平均、目的関数の重みなど)に基づいて分類する。それぞれの属性値は入力される可能性のあるデータのおおまかな範囲が既知であるとして、各々を low, medium, high の3値に振り分け、Fig. 4.5に示すデータ分類行列の1要素に割り付ける。例えば、行列中の“A”という要素は、回送距離 low, 搬送距離=low, タスクの密集度=lowであることを示す。ここでのタスクデータは、タスクの搬送要求時刻とタスクの種類(搬送経路)から構成されるものとする。

4.4.2 評価関数合成モジュール

スケジューリング状態の評価項目をもとにして、分類されたクラスの評価関数を合成する。本モジュールに与えられたタスク集合に対して、各々の評価値が最大及び最小であるものを優先してスケジューリングしていった場合の解スケジュールに対する目的関数値を計算し、Table 4.1(a), (b)に示すような表をそれぞれ作成する。表中のA, B, CはFig. 4.5に示す行列の要素をさすものとする。二つの表を比較して、目的関数値の差が小さい項目は、たとえ両者が共によい結果であってもその評価項目の目的関数に対する影響度は小さいと考えられる。従って、目的関数値の差が大きい項目だけが目的関数に影響を及ぼすものであるとして、(4.4)式を用いて各評価項

Table 4.1: An example of degree of influence [9].

(a) Value of evaluation items of schedules using the priority rule “largest value of i -th evaluation item ($i = 1, \dots, 7$)”

| Pattern | Evaluation item | | | | | | |
|---------|-----------------|---|----|----|----|----|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| A | 20 | 7 | 13 | 4 | 11 | 12 | 22 |
| B | 12 | 3 | 26 | 20 | 10 | 14 | 12 |
| C | 21 | 9 | 21 | 13 | 15 | 4 | 2 |

(b) Value of evaluation items of schedules using the priority rule “least value of i -th evaluation item ($i = 1, \dots, 7$)”

| Pattern | Evaluation item | | | | | | |
|---------|-----------------|----|----|----|----|----|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| A | 10 | 9 | 3 | 2 | 5 | 15 | 17 |
| B | 3 | 7 | 34 | 15 | 20 | 11 | 4 |
| C | 21 | 20 | 11 | 32 | 5 | 4 | 8 |

(c) Degree of influence

| Pattern | Evaluation item | | | | | | |
|---------|-----------------|-------|-------|-------|-------|-------|-------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| A | +0.67 | -0.25 | +1.30 | +0.67 | +0.75 | -0.22 | +0.26 |
| B | +1.20 | -0.80 | -0.27 | +0.29 | -0.67 | +0.24 | +1.00 |
| C | +0.00 | -0.76 | +0.63 | -0.42 | +1.00 | +0.00 | -1.20 |

目のスケジュールに及ぼす影響度の初期値を計算する (Table 4.1 (c)). 影響度が各評価項目にわたる影響度の平均値以上であれば、その項目はスケジュールに大きな影響を与える因子であるとし、評価関数を構成するためのプリミティブとして採用する。

影響度:

$$d_i = 2(z_{max_i} - z_{min_i}) / (z_{max_i} + z_{min_i}) \quad (4.4)$$

if $|d_i| \geq \sum_i |d_i| / n$

then 評価項目 i をプリミティブとして採用

ここで、

i : 評価項目 (搬送要求時刻, 搬送要求待ち時間, 遊休時間, 回送時間, 搬送時間, 各搬送車の稼働率, 搬送要求待ち時間の現時刻までの総和; $i = 1, 2, \dots, n$)

z_{max_i} : 評価項目 i について評価値最大タスクを優先してディスパッチングを行なった場合の解スケジュールの目的関数値

z_{min_i} : 評価項目 i について評価値最小タスクを優先してディスパッチングを行なった場合の解スケジュールの目的関数値

(4.4) 式の影響度を用いて (4.5) 式のようにクラス c のタスクパターンに対するノード n (スケジュール状態) の優先度評価関数 $f_c(n)$ を以下のように構成する。

$$f_c(n) = \sum_i |d_i| X_i \quad (4.5)$$

ここで,

$$X_i = \begin{cases} z_{min}/x_i & (\text{評価項目最小値優先の時}) \\ x_i/z_{max} & (\text{評価項目最大値優先の時}) \end{cases}$$

4.4.3 優先度評価関数の修正

前節で合成された優先度評価関数は以下に示すアルゴリズムで修正することにより目的とするスケジュール目的関数に適応したものとする。

Step 1. (4.5) 式で表される評価関数を用いて縦型探索 (ディスパッチング) を行なう。目的関数値が小さい順にそのプリミティブの影響度を $\delta\%$ 増加させていく。この時、解が改善された影響度のみを増加させた値にし、それ以外は増加前の値に戻す。

Step 2. Step 1 とは逆に、目的関数値が大きい順にそのプリミティブの影響度を減少させていく。この時、解が改善された影響度のみを $\delta\%$ 減少させた値にし、それ以外は減少前の値に戻す。

Step 3. 解が改善される影響度がなくなるまで Step 1, 2 を順次繰り返す。

Step 4. 解スケジュールが各プリミティブ単独のものよりも改善されていないならば、そのプリミティブの影響度を目的関数値の大きさに比例して増加させ、Step 1 に戻る。

Step 5. 得られた評価関数でフィルター付きビーム探索を行ない、その結果をスケジュールとして出力する。

ヒューリスティックベースはデータ分類行列の要素毎に過去に合成されたローカル評価関数を蓄えたもので、ヒューリスティックベースに蓄えられたローカル評価関数は次のスケジューリングに対する初期評価関数として使用される。

4.5 搬送車スケジューリング問題への適用

4.5.1 スケジューリング要求

数値例として Fig. 4.1 に示す軌道を持つ搬送車スケジューリング問題に対してローカル優先度評価関数を獲得することを考える。搬送車に対するスケジューリング要求のパターンは、以下に示す5種類のものを用いた。また、それぞれのパターンを Fig. 4.6~4.10 に図示する。

パターン 1: 比較的短い時間に多くの搬送要求が発生。

パターン 2: 比較的搬送時間が長いタスクが多く発生。

パターン 3: タスクの発生間隔が比較的大きく、回送時間が比較的短いタスクが多く発生。

パターン 4: タスクの発生間隔が比較的大きく、搬送時間が比較的短いタスクが多く発生。

パターン 5: 回送時間が比較的小さいタスクが多く発生。

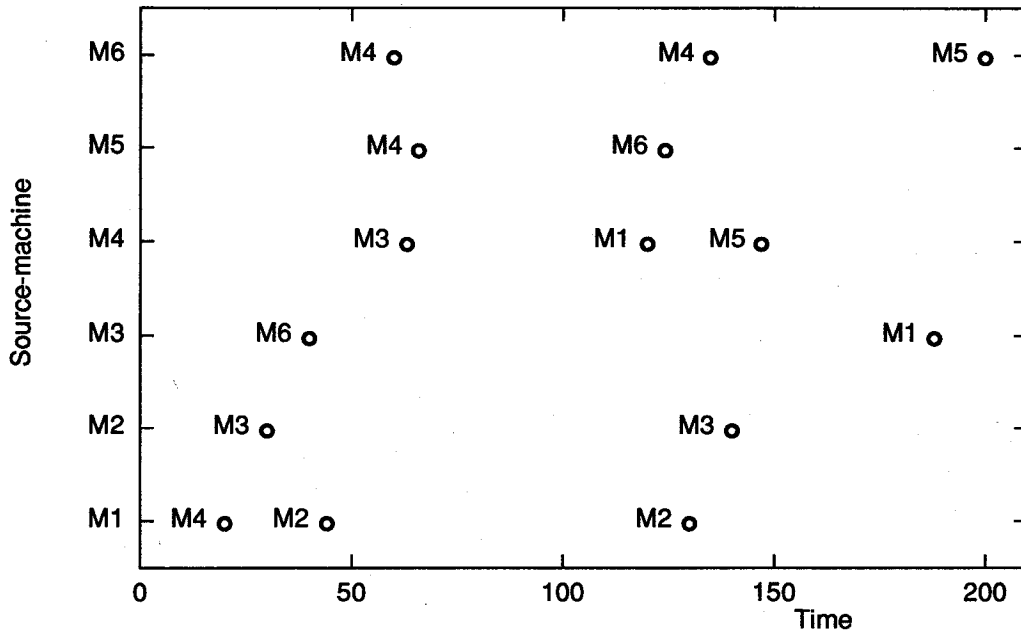


Fig. 4.6: Scheduling requirement 1.

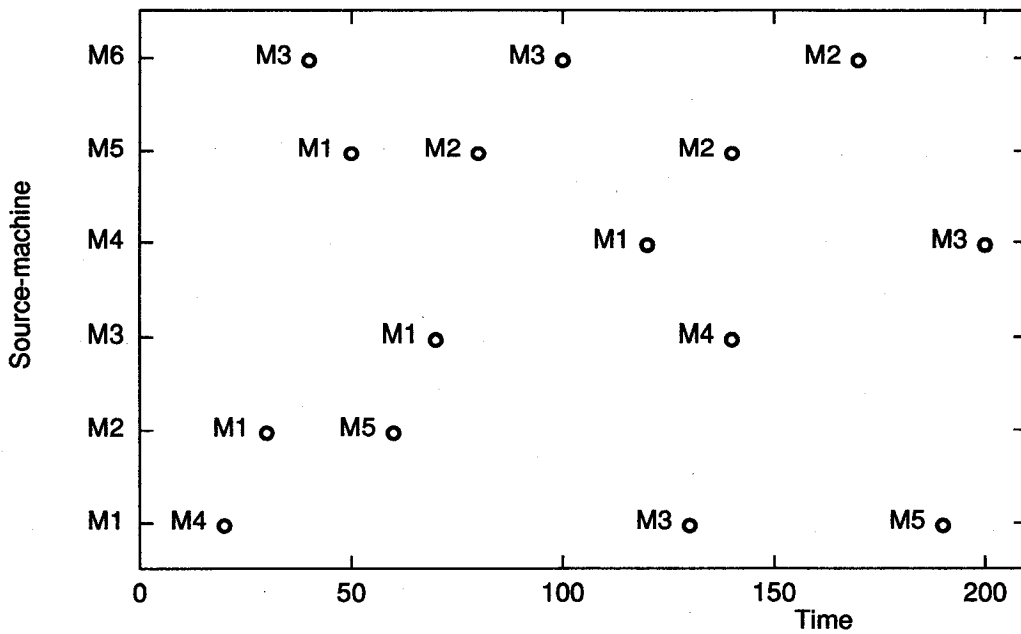


Fig. 4.7: Scheduling requirement 2.

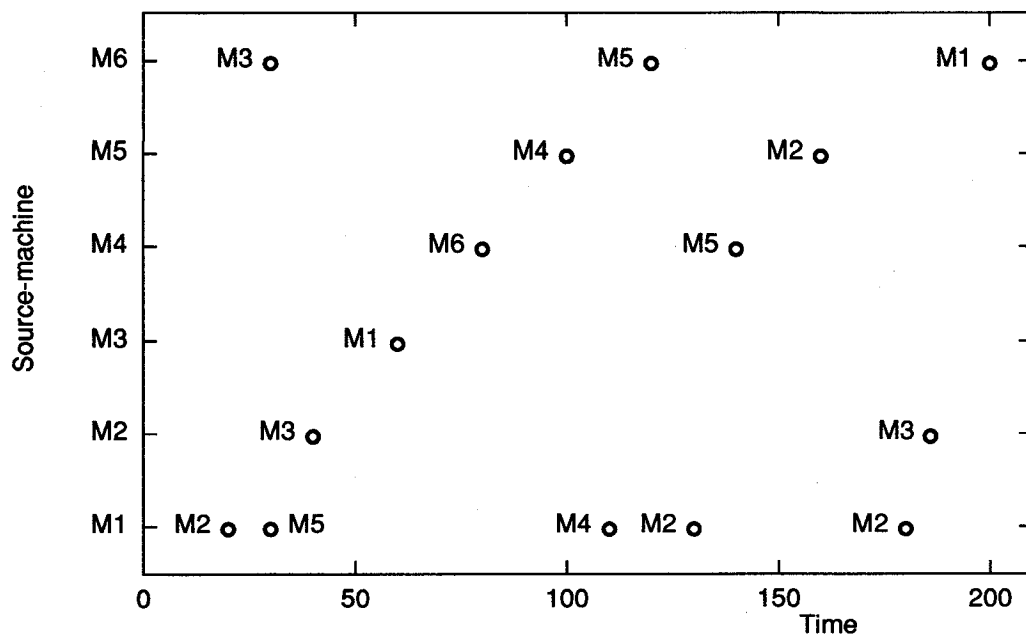


Fig. 4.8: Scheduling requirement 3.

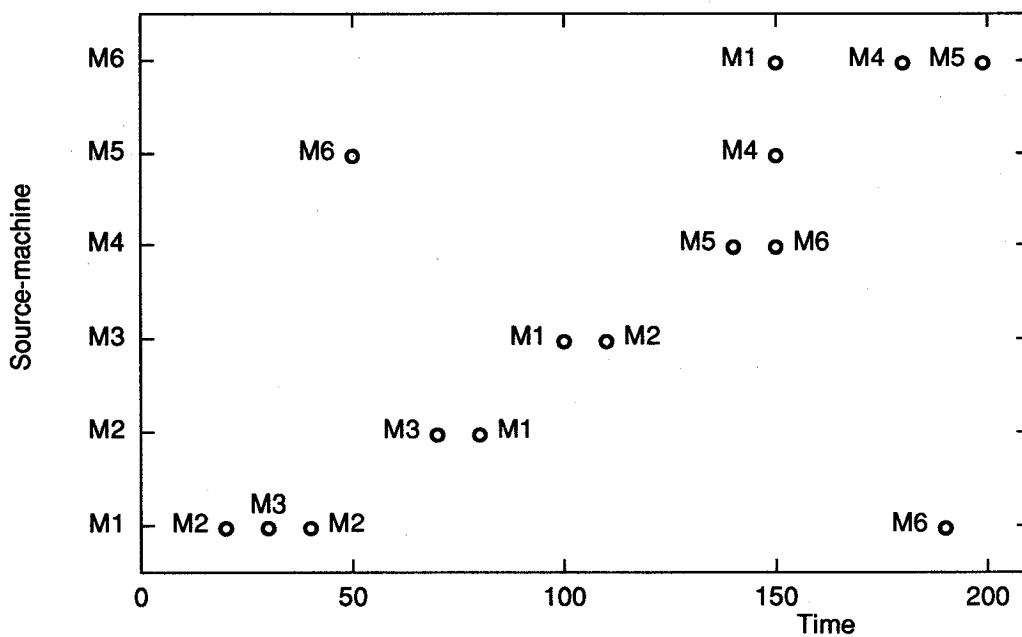


Fig. 4.9: Scheduling requirement 4.

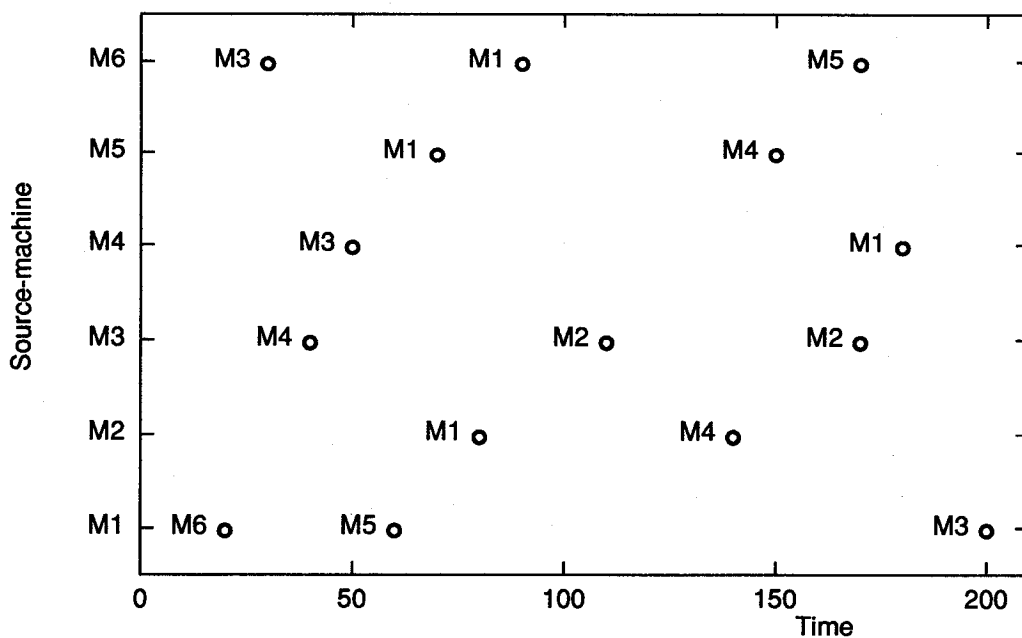


Fig. 4.10: Scheduling requirement 5.

4.5.2 分枝限定法との比較

本数値実験の解を分枝限定法で求めた解と比較した例を **Table 4.2**に示す. 分枝限定法で求めた解は, 数週間計算した結果得られた解のうち最も良い解を示している. その結果, 若干分枝限定法で求めた解と目的関数が離れているものも存在するが, Table 4.2のうち, 本章で提案した方法で計算した解の計算時間が1~2分であることを考慮すると, 概して良好な解が得られていることがわかる. 本数値実験の計算には, SUN Sparc Station 1 を用いた. ただし, フィルター幅とビーム幅はそれぞれ10, 3に設定した.

Table 4.2: Comparison of the performance of schedulings [9].

| Pattern | Value of the objective function | |
|---------|---------------------------------|-----|
| | F-B search | B&B |
| 1 | 278 | 191 |
| 2 | 583 | 538 |
| 3 | 136 | 123 |
| 4 | 78 | 38 |
| 5 | 301 | 285 |

Table 4.3: The coefficients of evaluation items [9].

| Pattern | Evaluation items | | | | | | | |
|---------|------------------|------|------|------|-------|------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | -2.42 | 0.00 | 0.92 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 2 | -3.93 | 0.00 | 0.36 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 3 | -6.26 | 0.00 | 0.47 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 4 | -7.10 | 0.00 | 0.49 | 0.00 | -0.99 | 0.00 | 0.00 | 0.00 |
| 5 | -3.10 | 0.00 | 0.85 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

4.5.3 スケジューリング要求と優先度評価関数の関係

ヒューリスティックベースにおける評価項目の重みはすべて0とした場合に数値実験で得られたローカル優先度評価関数の各評価項目に対する値を Table 4.3に示す。Table 4.3中の評価項目1～8は、それぞれ搬送要求時刻、搬送要求待ち時間、有休時間、回送時間、搬送時間、各搬送車の稼働率および搬送要求待ち時間の現時刻までの総和を意味するものとする。Table 4.3から、本数値実験における搬送車スケジューリングにおいては、搬送要求時刻、および有休時間がスケジューリングを行なうのに重要なファクターであることがわかる。また、パターン4の例では、搬送時間も考慮に入れる必要があることを示している。

4.5.4 フィルター幅とビーム幅に関する考察

フィルター幅を1～14、ビーム幅を1～4の間でそれぞれ変化させた場合のスケジューリング目的関数値と分枝限定法で求めたスケジューリング目的関数値との差の変化を Fig. 4.11～4.15に示す。Fig. 4.11～4.15から、少なくともビーム幅が3以上では目的関数の値はほとんど向上していないことが分かる。また、いずれの場合もフィルター幅が10以上であれば目的関数の値は変化していない。以上のことより、本章で取扱った搬送車スケジューリング問題に対してはフィルター幅とビーム幅は余裕を見てそれぞれ12, 3が適当であると推測される。

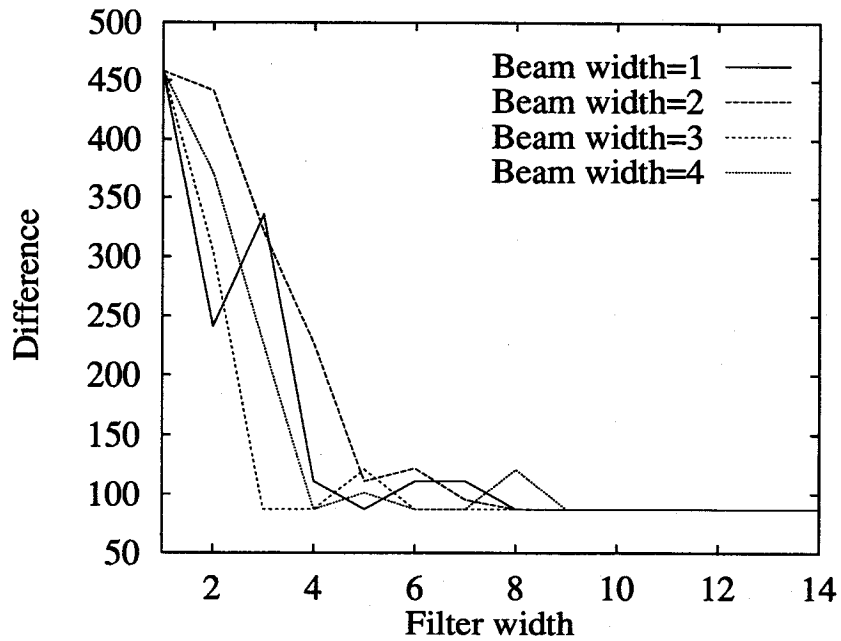


Fig. 4.11: Distribution of the values of the objective function for scheduling requirement 1.

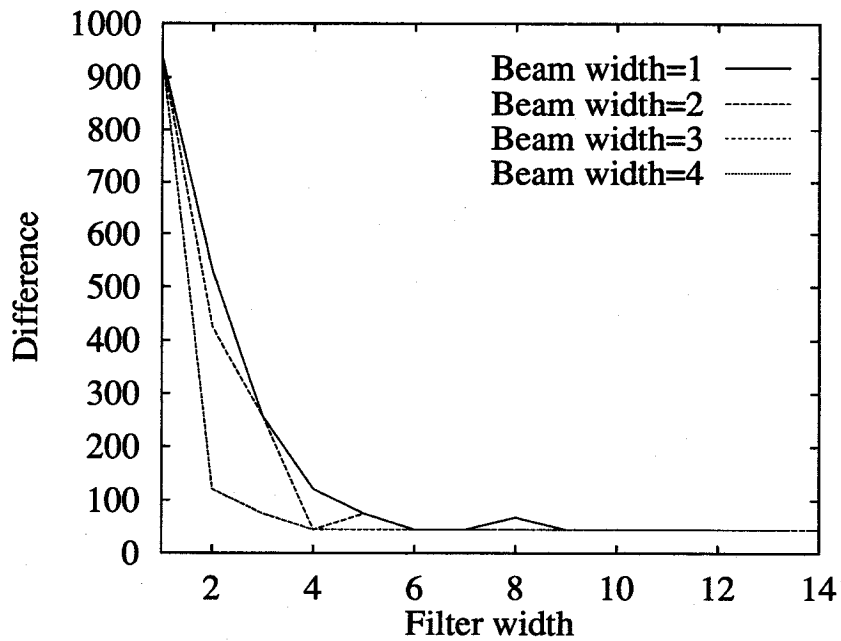


Fig. 4.12: Distribution of the values of the objective function for scheduling requirement 2.

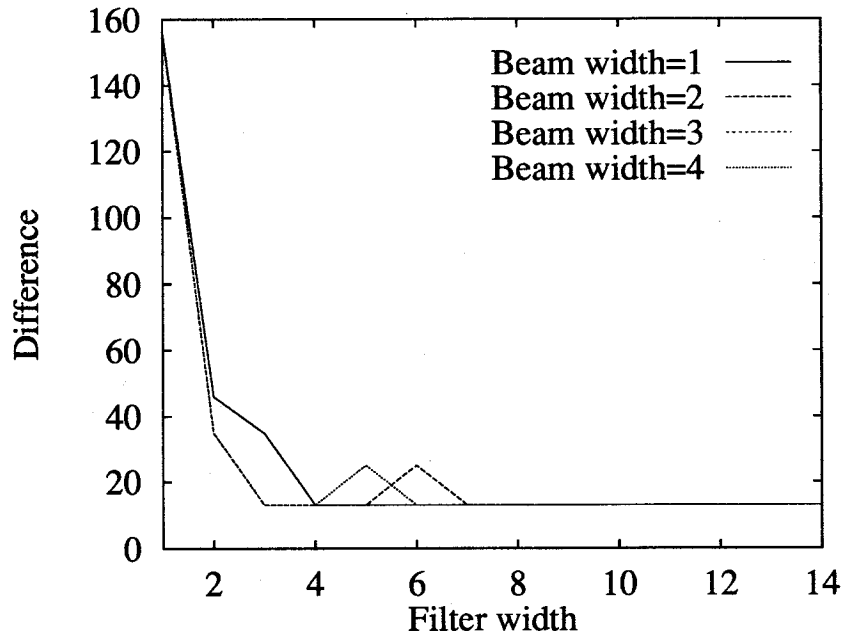


Fig. 4.13: Distribution of the values of the objective function for scheduling requirement 3.

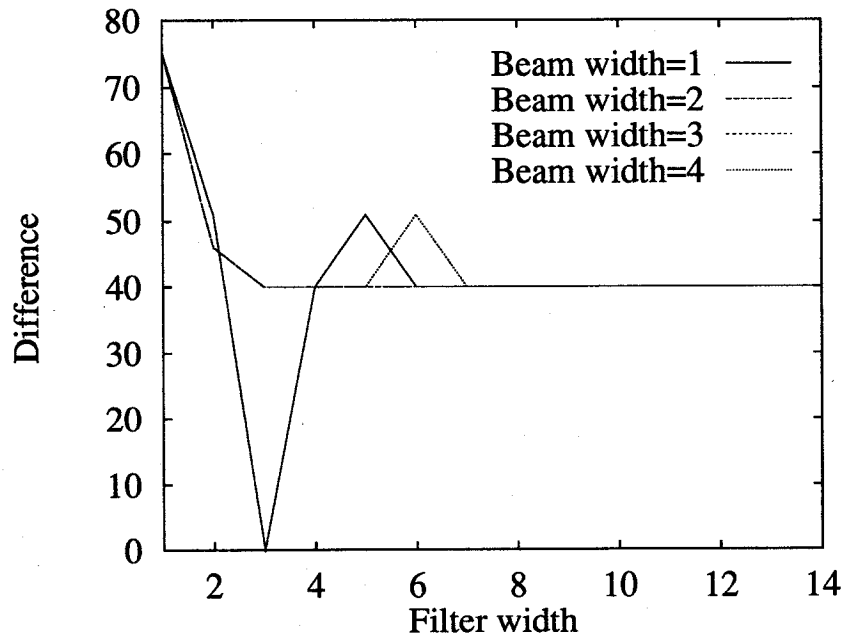


Fig. 4.14: Distribution of the values of the objective function for scheduling requirement 4.

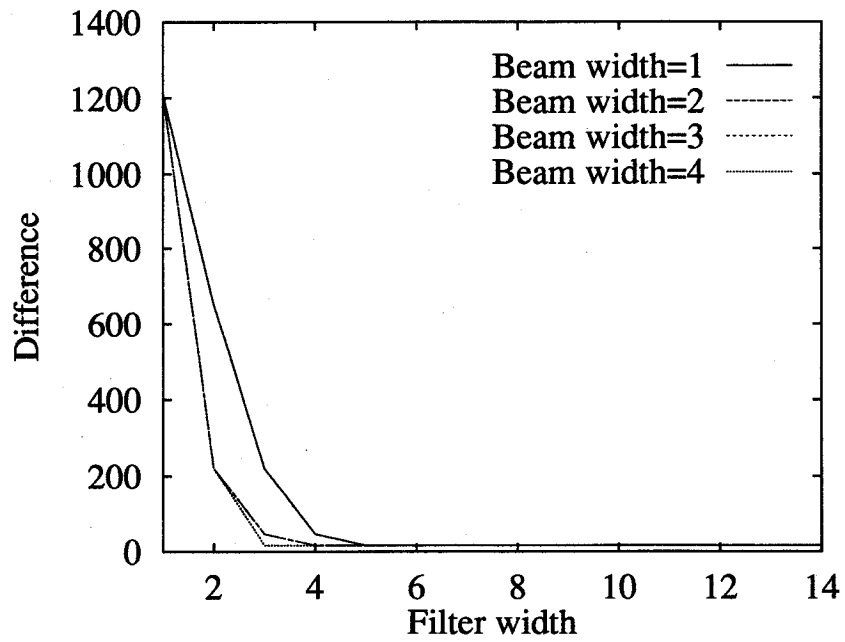


Fig. 4.15: Distribution of the values of the objective function for scheduling requirement 5.

4.6 むすび

本章では、ヒューリスティック探索におけるボトルネックとなっているヒューリスティック知識の獲得をより容易に実現するための枠組みを提案し、搬送車スケジューリング問題の解法に応用することを試みた。ここで提案したシステムは、ヒューリスティック探索の一手法であるフィルター付きビーム探索におけるヒューリスティクス(問題固有の経験的知識)を、探索中のスケジュール状態を評価する評価関数という形で獲得する。システムに入力されたスケジューリング問題をいくつかのパターンに分類し、そのパターンに対して過去に合成された評価関数があればそれを評価関数の初期値として用い、過去に合成されたものがなければ予め用意してある評価項目をもとにして合成する。このようにして合成された評価関数がスケジュール目的に合致した結果であるか否かを調べ、合致していなければ評価関数を修正し、最終的に得られた評価関数を用いて生成されたスケジュールをこのシステムにおけるスケジュール結果として出力する。そのときに使用した評価関数は、ヒューリスティックベースに蓄積し、次回以降のスケジューリングに対して用いる。

本システムの利点は、与えられたスケジューリング問題を上手く分類することによって、複雑な構造の問題に対してでも容易に評価関数を合成および蓄積し、それを逐次的に洗練していくことができる。また、蓄積された評価関数を見ることによって、その関数の属するパターンのスケジューリング問題では、どのようなスケジューリングに対する評価項目が重要であるかを容易に理解することができる。

今回の数値実験によって得られた評価関数を見ることで、搬送車スケジューリング問題に対しては、搬送要求時刻が早く、かつ遊休時間の大きいタスクを優先してスケジューリングすることによって、タスクの待ち時間の総和(目的関数)をほぼ最小化することができることがわかった。

今後の課題としては、大規模なジョブショップスケジューリング問題など種々の問題へ適用してシステムの評価を行なうことにより、評価関数生成のアルゴリズムの改良を行なう必要がある。

第5章 ルールベースを用いたフレキシブル 生産リアルタイムスケジューリング

5.1 まえがき

一般に、フレキシブル生産システム (FMS) [1] における生産は、オフラインで作成された生産スケジュールに従って行なわれる [2]. ところが、実際の生産においては、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、特急ジョブの割込などによりオフラインで作成した生産スケジュール通りに生産することができなくなる場合が多い. この場合、短時間にスケジュールを作成し直さなければならない. しかし、通常オフラインスケジューリングに用いられる分枝限定法などの OR 手法やシミュレーションによるスケジューリングは計算時間がかかりすぎるため、再スケジューリングの目的に適用することは困難である. また、近年スケジューリングを行なうためのエキスパートシステムが開発されている [3,4]. しかし、これらは基本的にオフラインスケジューリングを対象としているため、リアルタイムスケジューリングに適用することは困難である.

本章では、このように不確実性の多い生産システムに対するスケジューリングのアプローチとして、部品のディスパッチングやルーティングにおける部品選択にルールベースによる意思決定を導入することにより、生産を行ないながらスケジュールを作成するリアルタイムスケジューリングをある生産目標に対して効率的に行なうことを試みる [5-7]. また、ルールベースの作成、評価にあたってはシミュレーションを行なう必要がある. 本章では工作機械の故障、修理時間および加工時間のばらつきという不確実な事象を含むシステムのモデリングに確率ペトリネット [8,9] を用いる. 次節以下に本章で提案するリアルタイム・ルールベース・スケジューリングシステムの構成、評価およびルールベースの構築に用いたシミュレーションモデルの構成、数値実験によるルールベースの評価について述べる.

5.2 リアルタイム・ルールベース・スケジューリングシステム

本章で提案するリアルタイムスケジューリングは、部品の加工に際し、ディスパッチングやルーティングに関する意思決定をルールベースによって制御することによりスケジューリングを行なう.

5.2.1 リアルタイムスケジューリングのためのルールベース

上述のディスパッチングやルーティングを行なわなければならない状況を検出した場合に、以下に示すようなルール [10] に従って優先規則を決定し、それに対応する作業または機械を優先的に選択する。

$$\text{if } P \text{ then } R \quad (5.1)$$

ここで、条件 P は FMS モデル内のバッファの状態や各部品の加工進展状況等を調べる述語を表わす。また、結果 R は優先規則 [11] かまたはつぎにチェックするルールのグループ番号を表わす。本章では、リアルタイムスケジューリングにおける納期余裕及び納期遅れの最小化を第一義的な生産目標とするスケジュールを求めることとし、以下に示すような $R_1 \sim R_8$ の優先規則を考えた。また、各工作機械が効率よく稼働し、生産目標を達成することを補助的な目標として $R_9 \sim R_{11}$ を使用した。

- R_1 : 納期順ジョブ優先
- R_2 : スラック時間最小ジョブ優先 (MINSLACK)
- R_2 : 最大納期遅れ時間最大ジョブ優先
- R_4 : 納期遅れ個数最大ジョブ優先
- R_5 : 総納期余裕時間最小ジョブ優先 (MINSPARE)
- R_6 : 納期余裕率最小ジョブ優先
- R_7 : ランダムにジョブを選択
- R_8 : どのジョブも負荷しない
- R_9 : バッファ占有率最大ジョブ優先 (MAXROB)
- R_{10} : 次工程バッファ占有率最小ジョブ優先 (MINROBN)
- R_{11} : 残り工程数最小ジョブ優先

ディスパッチングまたはルーティングが発生した時刻において対象システムから得られる情報により述語 P を構成する。 P は、第2章の 11~12ページおよび 22ページに示す述語の論理積および和で表現される。

5.2.2 ルール構築のためのヒューリスティクス

納期遅れが生じないときは納期余裕を小さくし、納期遅れが生じる場合は、納期遅れが一つの部品に集中せず、納期遅れを小さくおさえようとする目標を設定し、次のような考えに基づいてルールベースを構成した。

- (1) 競合している部品の種類数, 未完成部品数, 納期, 加工残り時間より加工開始時刻を設定し、その時刻までは部品を機械に負荷しない。

- (2) 一旦加工を開始した部品については(1)の条件が成り立たなくても部品を投入する。
- (3) 競合を起こしている部品のすべてに納期余裕があれば、納期までの余裕が一番小さいものを先に負荷する。
- (4) 競合を起こしている部品のうち、すでに納期遅れが発生しているものがあれば、納期の一番早いものを先に負荷する。
- (5) 競合が搬送車で発生している場合は、次工程のバッファ占有率が最小のものを先に搬送する。

各述語はルール中において次のような機能を果たす。*time* は各部品の最初の素材をシステムに投入する時刻を決め、*buffer* はブロッキングを防止する。*progress* は加工の完了した部品を優先的に搬出する。*best-time1* と *best-time2* は加工を開始するか否かを定める。競合している部品数に応じて加工を開始するか否かの決め方が異なり、競合解消は工作機械と搬送車とで考え方が異なるため、*conflict* と *machine* は競合の種類を分類するために用いられる。*beginning* によって一旦、加工を開始した部品を続けて加工し、*filter* によって各部品ごとに設定された加工開始時刻以前はその部品がライン内に投入されるのを防ぐ。本章では、**Fig. 2.8**に示す納期のジャストイン・タイムを目的としたルールベースを用いるものとする。ルールはレベル4のグループ1における最初の条件からチェックされ、もし条件が成立すれば矢印の順に、成立しなければ、同一グループ内の次の条件がチェックされる。また、レベル0で *true* のとき、それに対応する優先規則により優先される部品が選択されるが、同一条件のものが2個以上あるときは、レベル0の次の *true* に対応する優先規則が適用され、選択される部品が一つになるまで繰り返される。

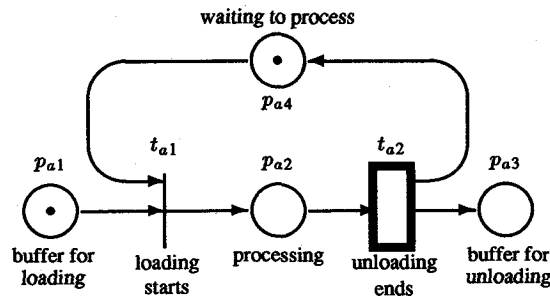
5.3 階層的確率ペトリネットによる生産システムのモデル化

5.3.1 階層的確率ペトリネット

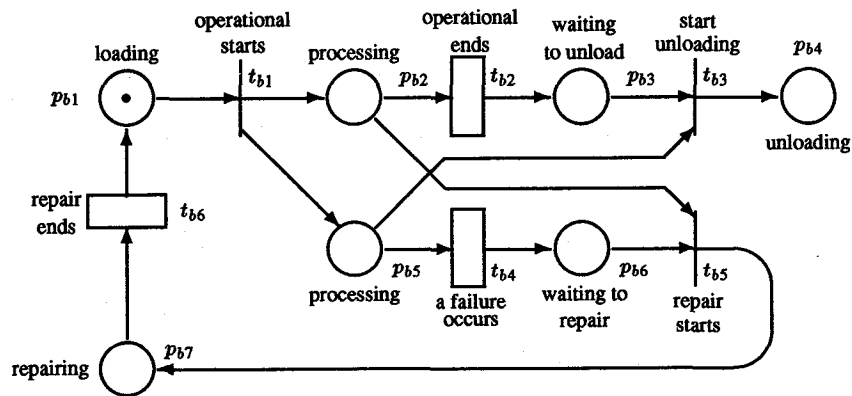
本章で用いる確率ペトリネットには、階層構造を表現するため、発火可能になった場合に、時間0で発火する即時トランジション (*immediate transition*) と、ある確率分布に基づいたランダム時間後に発火する時間トランジション (*timed transition*) の他に、確率ペトリネットのあるプレースからある他のプレースへトークンが移動するのにかかる時間を経過した後に発火する階層化時間トランジション (*hierarchical timed transition*) を導入している。

5.3.2 モデル化における仮定

1つの部品を1台の機械で加工する場合の階層的確率ペトリネットモデルの例を **Fig. 5.1** (a), (b)に示す。**Fig. 5.1**において、即時トランジションおよび時間トランジションは、それぞれ縦線および長方形で表わされるものとする。また、モデル化に際して、つぎの条件を仮定している。



(a) An example of transporting level model
(Discrete-time stochastic Petri net).



(b) An example of processing level model
(Continuous-time Stochastic Petri net).

Fig. 5.1: An example of partitioned stochastic Petri net model [5].

- (1) 工作機械の故障は偶発故障のみを取り扱い, 瞬間故障率 γ は時刻 t にかかわらず一定である.
- (2) 故障が発生した工作機械は修理可能である.
- (3) 故障が発生した機械は直ちに修理作業を行ない, 修理に必要な時間は指数分布に従う.
- (4) 修理完了後直ちに通常作業に入る.
- (5) 部品の実際の加工時間は, 生産計画で決定された加工時間を平均とする正規分布に従う.
- (6) 予防取り替えは考慮しない.

また, 前述の条件を表現するため, 本章における階層的確率ペトリネットには具体的に次のような条件を導入している.

- (1) 各時間トランジションの発火時間はそれぞれ次のような確率分布に従う乱数によって決定する. ただし, γ は瞬間故障率であるものとする.

- t_{b2} : 平均が工程計画で決定された加工時間, 分散が σ^2 の正規分布
- t_{b4} : 平均が $1/\gamma$ の指数分布
- t_{b6} : 平均が平均修理時間 r の指数分布

(2) 階層化時間トランジション t_{a2} の発火時間はトークンが Fig. 5.1 (b) のプレース p_{b1} から p_{b4} まで移動するのに要する時間とする.

(3) 各プレースにはトークンの数の上限 $L(p_i)$ を次のように与える.

- p_i がバッファプレース: $L(p_i) = \text{バッファ容量}$
- その他のプレース: $L(p_i) = 1$

ただし, (1) における発火時間は 0 以上であるとする. 発火時間が負になった場合は 0 とみなす.

工作機械, 搬送車と部品の流れの代表的な関係を物流レベルでモデル化した例を Fig. 5.2 に示す. 本章における FMS の物流レベルのモデル化はこれら 4 つの形の組合せによって行なう. また, 部品加工レベルのモデル化は, 各部品の工程ごとに計画段階の加工時間および予想される加工時間の分散, 使用する工作機械の故障率, 平均修理時間を Fig. 5.1 (b) の確率ペトリネットに与えることにより行なう. さらに物流レベルの確率ペトリネットモデルのシミュレーションは, 簡単のため連続時間系ではなく離散時間系のモデルとみなして行なうものとする.

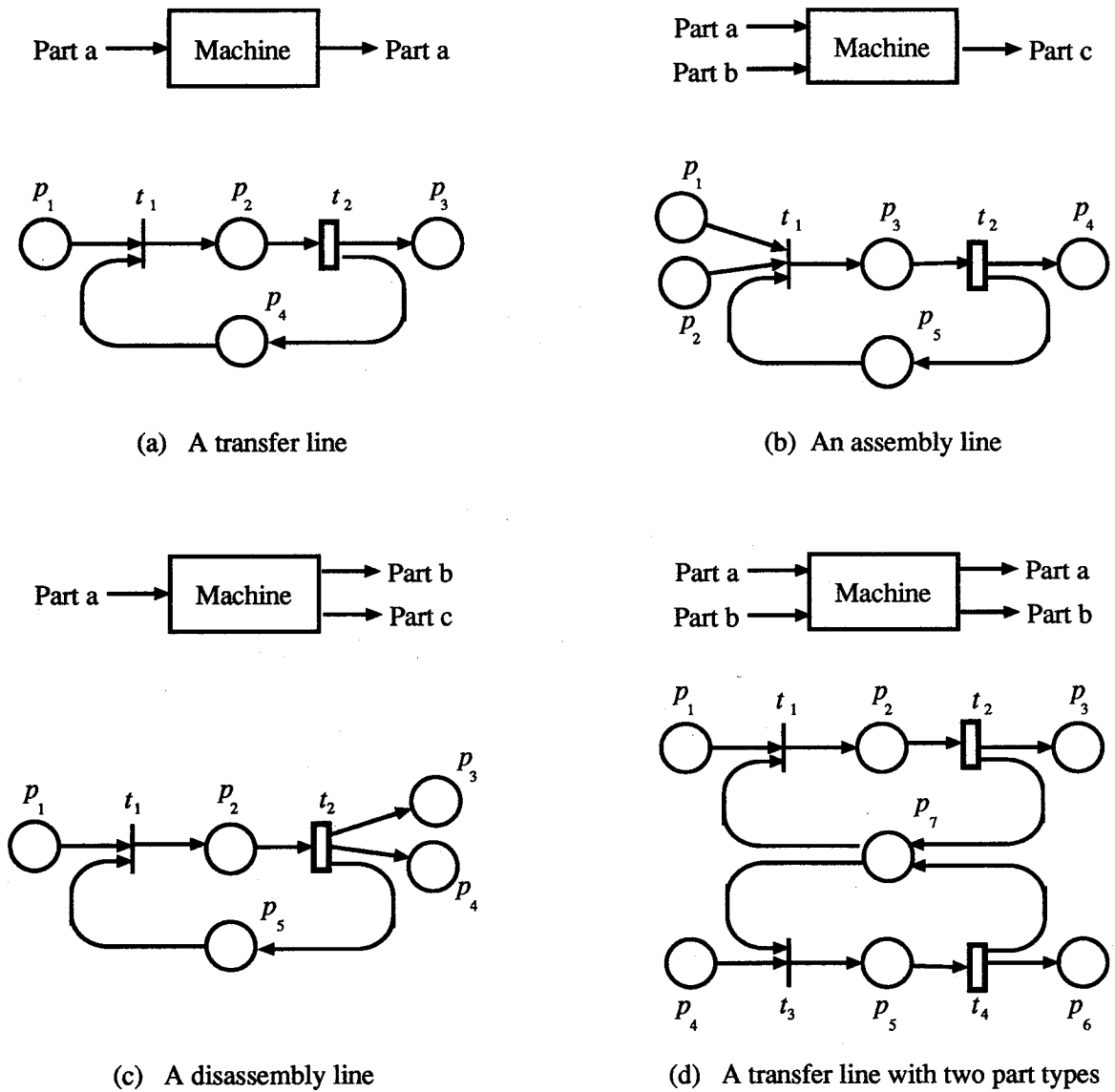


Fig. 5.2: Transporting level models of four basic FMS elements [5].

5.4 数値例によるルールベースの評価

5.4.1 数値例の条件

簡単な数値例としてライン型 FMS のリアルタイムスケジューリングを考える。FMS は 3 台の工作機械 (M_1, M_2, M_3) とローディングステーション, アンローディングステーションおよび 1 台の搬送車から構成されているものとする。各部品 (a, b, c) が加工される順序は以下の通りである。ただし, 括弧内の数字は, 各部品が各工作機械で加工されるときに要する生産計画における加工時間を示す。

部品 a : $M_1(6) \rightarrow M_2(6)$

部品 b : $M_1(4) \rightarrow M_2(3) \rightarrow M_3(5)$

部品 c : $M_1(5) \rightarrow M_3(6)$

各部品が工作機械間を搬送されるのにすべて 1 単位時間かかるとし, 各工程にはそれぞれ容量 5 のバッファが設置されているものとする。各工作機械において, 現在までと異なる種類の部品を次に加工する場合の工具付け替え時間を 1 単位時間と設定する。このような FMS において各部品をそれぞれ 20 個ずつ, 納期を 300 単位時間として生産を行なう例を考える。

ここで, Fig. 2.8 に示すルールベースおよび Table 5.1 に示すディスパッチングルールに従って 5.3 節で述べたシミュレーションモデルを用いてリアルタイムスケジューリングのシミュレーションを行なう。工作機械の故障率, 修理時間, 加工時間のばらつきの各条件は Table 5.2 に示した 13 種類について行なう。また, 搬送機械については故障は発生せず, 搬送における所用時間の分散は 0.7 であるものとする。

Table 5.1: Dispatching rules [7].

| | | |
|-----------|---|--|
| Rule-base | : | Rule-base |
| FCFS | : | First come first served |
| SPT | : | Shortest processing time |
| MINSLACK | : | Minimum slack time |
| MAXROB | : | Maximum rate of occupation for buffers |
| MINROBN | : | Minimum Rate of occupation for buffers in next process |
| MAXLOSS | : | Maximum sum of loss time |
| MINSPPARE | : | Minimum sum of spare time |
| MAXTARD | : | Maximum tardiness |

Table 5.2: Conditions of numerical experiments [7].

| | Failure rate | Variance of processing time | Mean repair time |
|---------|--------------|-----------------------------|------------------|
| Case 1 | 0.000 | 1.00 | 10 |
| Case 2 | 0.001 | 1.00 | 10 |
| Case 3 | 0.005 | 1.00 | 10 |
| Case 4 | 0.010 | 1.00 | 10 |
| Case 5 | 0.015 | 1.00 | 10 |
| Case 6 | 0.005 | 0.00 | 10 |
| Case 7 | 0.005 | 1.00 | 10 |
| Case 8 | 0.005 | 2.00 | 10 |
| Case 9 | 0.005 | 3.00 | 10 |
| Case 10 | 0.005 | 1.00 | 5 |
| Case 11 | 0.005 | 1.00 | 10 |
| Case 12 | 0.005 | 1.00 | 15 |
| Case 13 | 0.005 | 1.00 | 20 |

5.4.2 結果と考察

Fig. 5.3, Fig. 5.4に、以上の条件のもとで100回シミュレーションを行ない、それぞれ加工終了時刻および加工終了時刻の標準偏差を示す。300単位時間という納期は、故障や加工時間のばらつきが生じないと仮定した時、ほぼ最適なスケジューリングが行なわれた場合に達成することのできる納期で、非常に厳しい条件である。ここでの数値実験では、工作機械の故障や加工時間のばらつきが発生することを想定しているため、各実験結果とも設定した納期からは大幅に遅れている。Fig. 5.3, Fig. 5.4からもわかるようにルールベースを用いた方がいずれの条件でも短い加工時間で終了していることがわかる。他のディスパッチングルールに固定した場合でも特定の条件のもとではルールベースを用いた場合と比較して近いものもあるが、他の条件に対しては大幅に劣るという結果を示している。

5.5 むすび

本章では、不確実性の多い生産システムに対するスケジューリングの一方法として、部品のディスパッチングやルーティングにルールベースによる意思決定を導入したリアルタイムスケジューリングを提案した。また、部品加工レベルを階層的構造でとらえた階層的確率ペトリネットを用いて不確実状況下の生産システムをモデル化し、リアルタイムスケジューリングのためのルールベースを評価する数値実験を行なった。その結果、単一のディスパッチングルールを用いたスケジューリングに比べて、本章で提案したリアルタイム・ルールベース・スケジューリングの優位性を確かめることができた。今後の課題として、より大規模な生産に対して本章で述べたリアルタイムスケジューリングを適用することにより本手法の有効性を評価することが挙げられる。

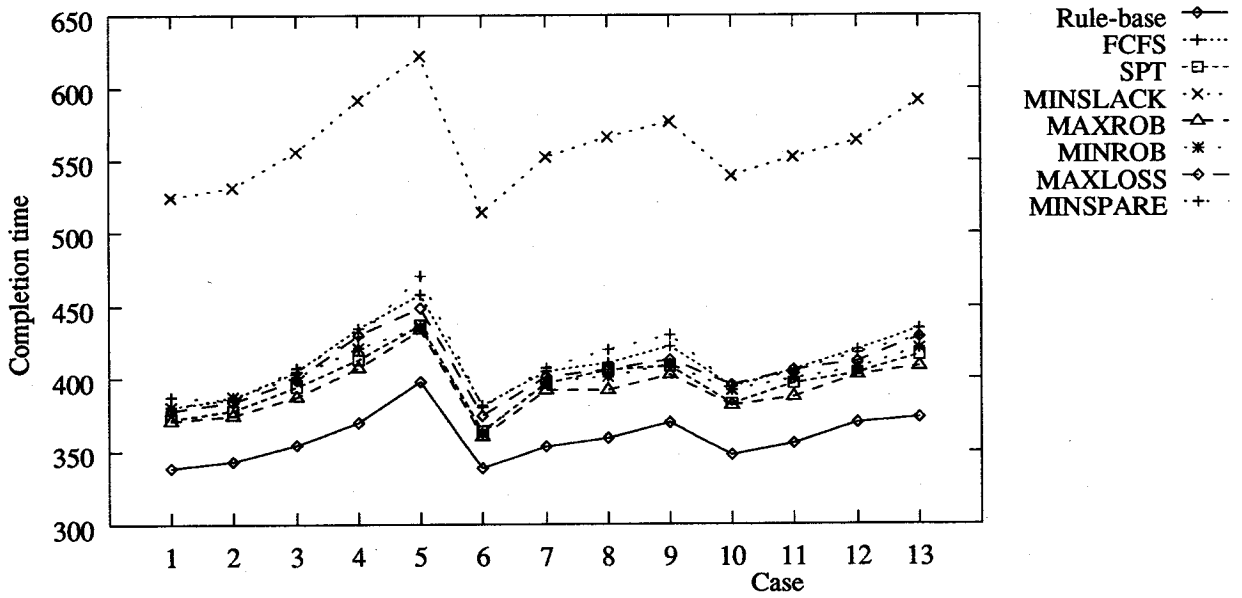


Fig. 5.3: Completion time [7].

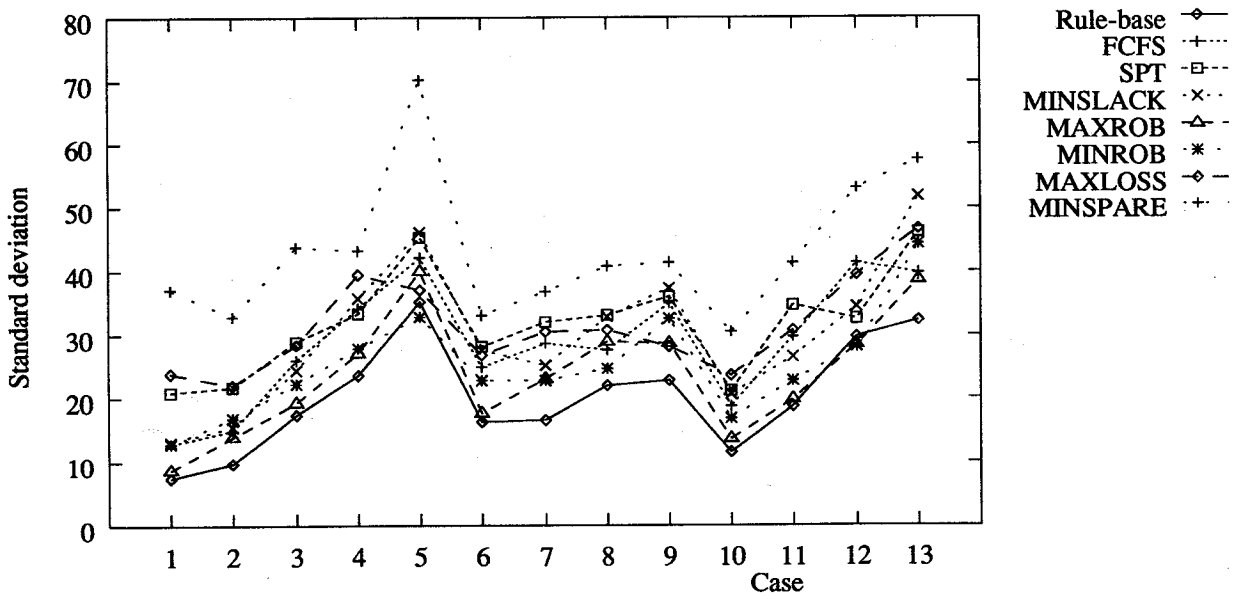


Fig. 5.4: Standard deviations of completion times [7].

第6章 分散型リアルタイムスケジューリング

6.1 まえがき

第1章でも述べたように、多品種少量生産、変種変量生産 [1,2] のように動的に生産環境が変化する生産システムでは、通常の実業システムで用いられるオフラインスケジューリング [3,4] 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づけや工作機械の割当などを行なうリアルタイムスケジューリング (オンラインスケジューリング) [5-7] の方が適していると思われる。しかし、従来のリアルタイムスケジューリングの手法は、システムの全情報を1台の計算機で処理してスケジューリングを行なっていくので、規模が大きくなると計算量も増大し、時間的な制約を満たすことが困難となる。したがって、多品種少量生産の実業システムのように、システムが大規模で、さらに、多様性、柔軟性、信頼性などの要求が高い場合には、従来の集中型システムによる制御よりも、複数のサブシステムに制御機構を分散させた分散型システム [8-11] による制御の方が好ましいと考えられる。

分散型システムの特徴をいくつかあげると、

- **高効率性**：サブシステムは自律的に動作し、並列に実行することができるため、速やかな対応が可能である。
- **信頼性, 耐故障性**：全体を大局的に総括する調整機構は存在しないので、一部の障害は局部的に処理することができ、全システムを停止する必要はない。
- **柔軟性, 多様性**：各サブシステムの協調的な相互作用によって秩序が生成されるので、サブシステム間の協調行動を変化させることによって、様々な目的や環境に対して柔軟に対応することができる。
- **モジュラー性**：システムが複数のサブシステムに分けられているため、拡張、縮小、テスト、保守等を容易に行なうことができる。

などがある。以上のようなことを考慮すると、大規模な多品種少量生産のリアルタイムスケジューリングは分散環境で行なうことが有利であると思われる。

本章では、これまで中央集中型であったスケジューリングシステムを分散環境 [12,13] のもとで実現することにより、より柔軟で信頼性の高いスケジューリングシステムを構成することを試みる [14]。各ジョブそれぞれに与えたスケジューリング要求に応じて、各機械に平等に配置し

たスケジューリングモジュールが他の機械の状況を参照しながらスケジューリングを行なうことにより、多様なスケジューリング要求に対応でき、しかも故障などの不慮の事態に対しても寛容なシステムを実現することができる。

さらに、具体的な実例として、半導体生産プロセスを対象としたスケジューリング問題を取りあげ、本章において提案した分散型リアルタイムスケジューリングの有用性を明らかにする。

6.2 分散型スケジューリング

多品種少量生産におけるユーザへの即応性を高めるためには、システムは柔軟性、耐故障性などを備えていることが必要である。6.1節で述べたように、分散型システムのアーキテクチャを用いれば、このような要求に応え得るシステムが構築できると考えられる。

本節では、多品種少量生産のスケジューリングを目的とした、分散型システムのアーキテクチャを考慮したリアルタイムスケジューリングシステムについて述べる。

6.2.1 生産システムのモデル化

生産工場での各工程は一般に、**Fig. 6.1**に示したように、搬送、加工、搬送のサイクルを繰り返してタスクを実行していく。本論文では、このサイクルを基本に、生産工場におけるジョブの流れを**Fig. 6.2**に示すようにモデル化した。

まず、生産工場に到着したジョブは、(1) システム内に搬入され、(2) 最初の工程を実施する機械の入力バッファで加工されるのを待つ。(3) 加工が終了すると、(4) 出力バッファで搬送されるまで待機し、(5) 搬送が実行されると次の工程を行なう機械の入力バッファに運ばれる。このようにして(2)から(5)を繰り返し、すべての工程が終わると、(6) 完成ジョブはシステムの外に搬出される。

このサイクルの中で、スケジュールを発生させるために決定すべき要素として、ジョブのルーティング、搬送するジョブの決定、次に加工を行なうジョブの入力バッファからの選択の3つがある。

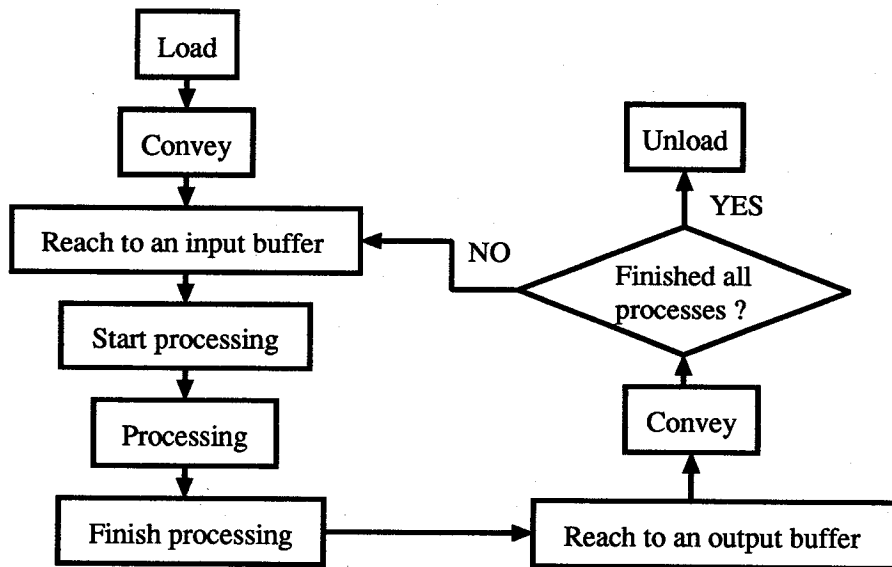


Fig. 6.1: Flow chart of processing in a factory.

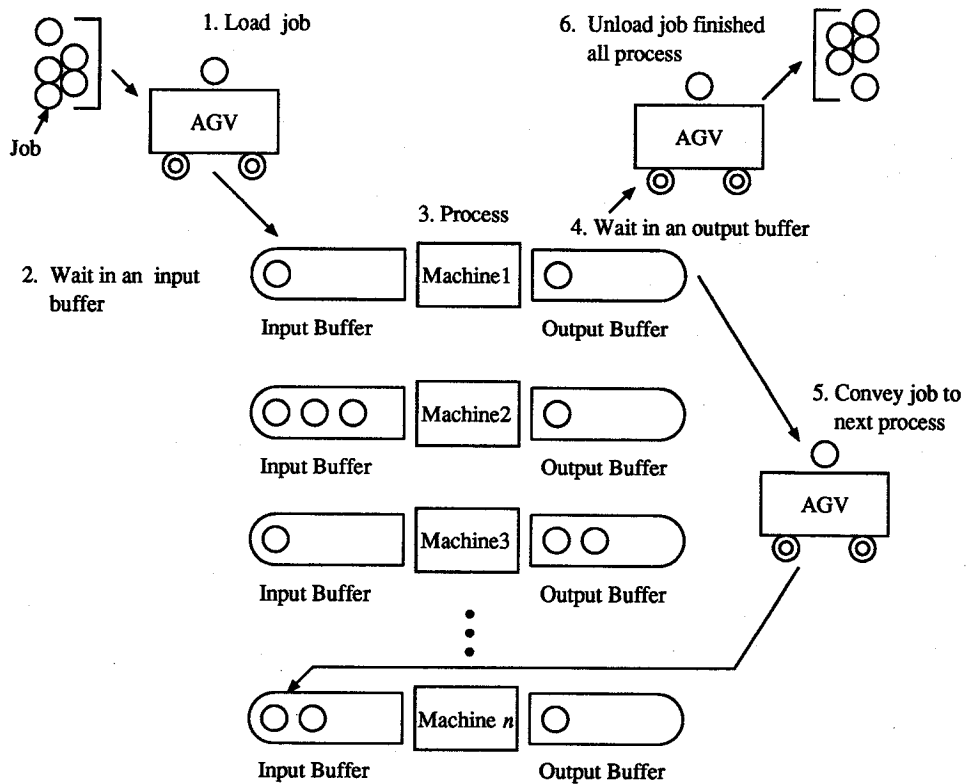


Fig. 6.2: An example of a model of a factory.

ルーティング:

各ジョブの各工程を実行できる代替機械は、一般には複数存在する。ルーティングとは、1つの工程に対して、複数の代替機械が存在する場合に、どの機械を用いて次の加工を行なうかを決定することである。一般には、各代替機械で加工性能は異なるので、スケジューリング目的を満足できるようなスケジュールを作成するためには、その時のシステムの状況に応じて適切なルーティングを行なう必要がある。ルーティングを決定する際に考慮を要する事項としては、次工程の候補となっている加工機械の入力バッファの占有状態、次工程における加工時間等がある。

搬送するジョブの決定:

すべてのジョブは、1つの作業が終了すると、その作業を行なった機械の出力バッファに入り、次の工程に搬送されるまで、そこで待機している。一般には、出力バッファには搬送されるのを待っているジョブが複数個存在する。1度に搬送することのできるジョブは有限個であるので、これらの搬送待ちのジョブからどのジョブを選んで搬送するかを決定する必要がある。搬送ジョブを決定する際に考慮を要する事項としては、現在当該ジョブが占有している出力バッファの占有状態、次工程を行なう機械の入力バッファの占有状態、次工程における待ち時間、搬送コスト等があげられる。

次に加工を行なうジョブの入力バッファからの選択:

搬送されたジョブは、次に加工されるべき加工機械の入力バッファにおいて、加工されるまで待機している。従って、加工機械が現在行なっているタスクを終了し、次の加工が開始可能になった時、どのジョブを選択して加工するかを決定する必要がある。入力バッファから選択するジョブを決定する際に考慮を要する事項としては、ジョブの納期までの余裕、次工程における待ち時間等があげられる。

6.2.2 分散型スケジューリングシステム

分散型システムでは、6.1節で述べたように、システム全体がいくつかのサブシステムの集合として構成されている。本章で作成したスケジューリングシステムでは、各加工機械を1つのサブシステムとみなし、それぞれのサブシステムを、スケジューリングを行なう平等なエージェントとして扱っている。分散型スケジューリングシステム概念図を Fig. 6.3に示す。

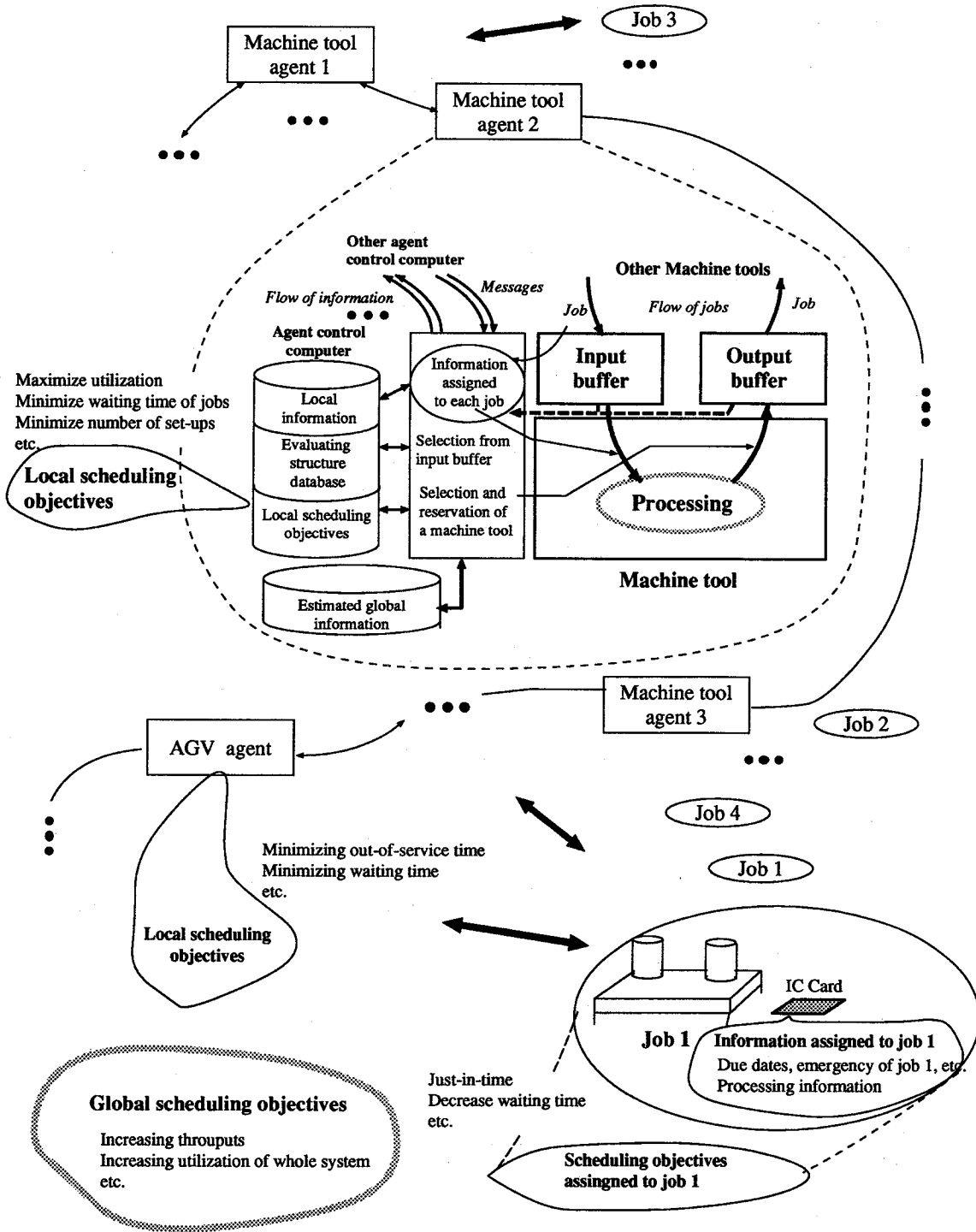


Fig. 6.3: Conceptual configuration of distributed real-time scheduling system.

各加工機械は、それぞれがその機械に関するローカルな情報のみを持つ。他の機械の状態や、システム全体の状態、他の機械にあるジョブに関する情報などは、それぞれの機械が持つローカルな情報を通信によってやりとりし、それをを用いてそれぞれが前節で述べたような競合を解決していくことにより、スケジュールを作成していく。ここでは、本章で作成した分散型スケジューリングシステムにおいて、それぞれの決定がどのような方法で行なわれるかについて述べる。

A. タグ情報

多品種少量生産においては、同時に流れるジョブの使用する加工機械、加工時間、納期などは異なっている。従って、スケジューリングを行なうためには、これらの各ジョブそれぞれによって異なる情報が必要である。本章では、これらの情報を各ジョブひとつひとつに持たせた“タグ”に記録し、エージェントの役割を果たす各機械がこのタグを参照することによってスケジューリングを行なっていく。その概略を Fig. 6.4 に示す。なお、これらの決定の方法の詳細については次節以降で述べる。

タグに持たせる情報は、次に示す通りである。

- 部品名
- ロット番号
- ジョブ経路
- 各タスクについての所要時間
- 現在の工程
- 納期
- 現在までの加工時間の和
- 加工開始時刻
- 納期属性
- 処理履歴
- そのジョブの、スケジューリング決定に用いる各評価項目に対する重要度

次節以降では、ルーティング、搬送するジョブの決定、次に加工するジョブの入力バッファからの選択の3つのスケジューリング決定の際にエージェント間で行なわれる通信について述べる。

B. ルーティングの際のエージェント間の通信

ルーティングの決定は、スケジューリング決定の中でも最も重要でしかも複雑なものである。本システムでは、あるジョブのひとつの工程が完了して、出力バッファに到着すると、その機械が当該ジョブの次工程のすべての代替機械についてそのジョブのタグ情報を送り、その結果各代替機械から返ってきた評価値をもとに次のルーティング先を決定する。

また、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、搬送先に到着した時すでに空きがない、といったことが起

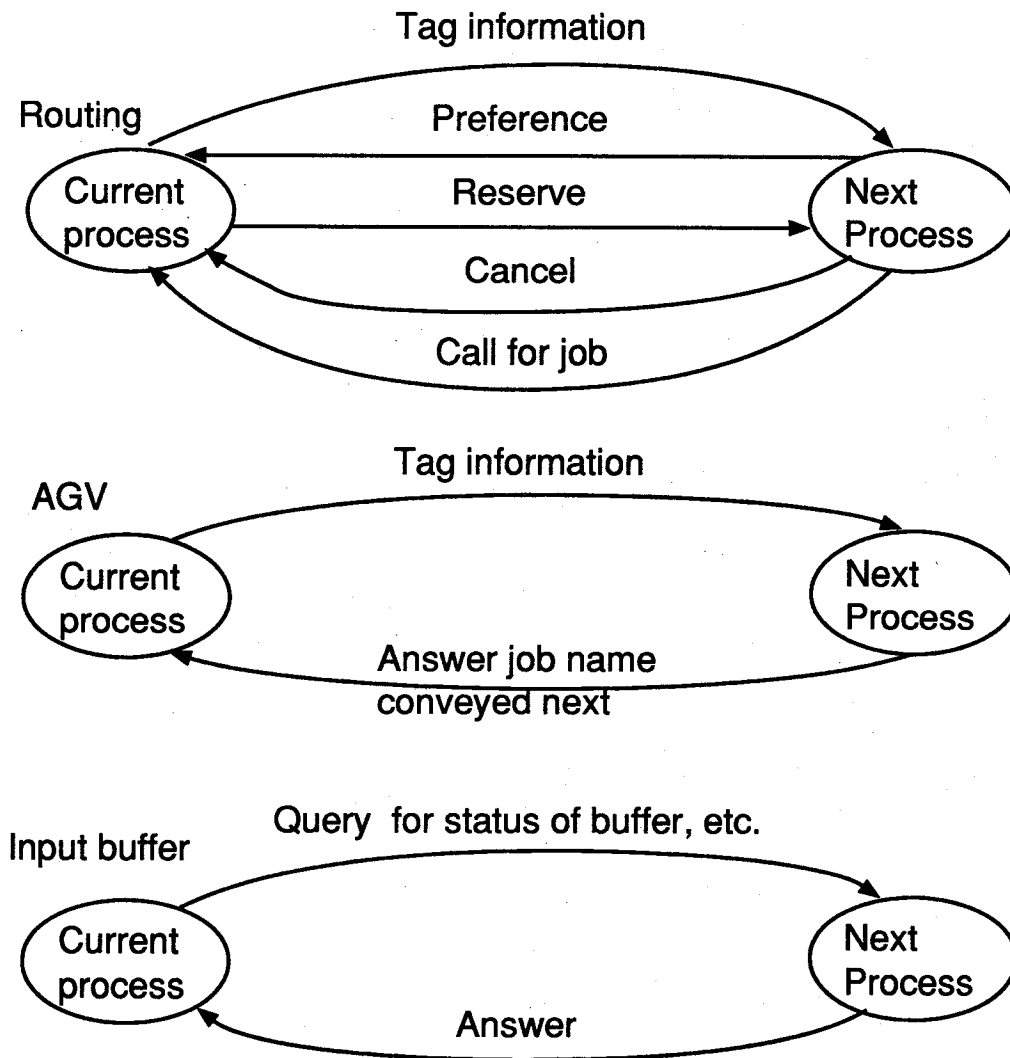


Fig. 6.4: Overview of communications between scheduling agents.

こる可能性がある。そこで、バッファの予約という考え方を導入する。すなわち、ルーティングが決まったジョブは、そのルーティング先の入力バッファを確保するようにする。

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、柔軟に対応することができるようにしている。

ルーティングにおける情報の流れを Fig. 6.5 に示す。

本システムにおいて、各ジョブのルーティングは次のような方法で行なう。

- (1) ひとつの工程が完了したジョブは、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、ということをも“ジョブ依頼可能性リ

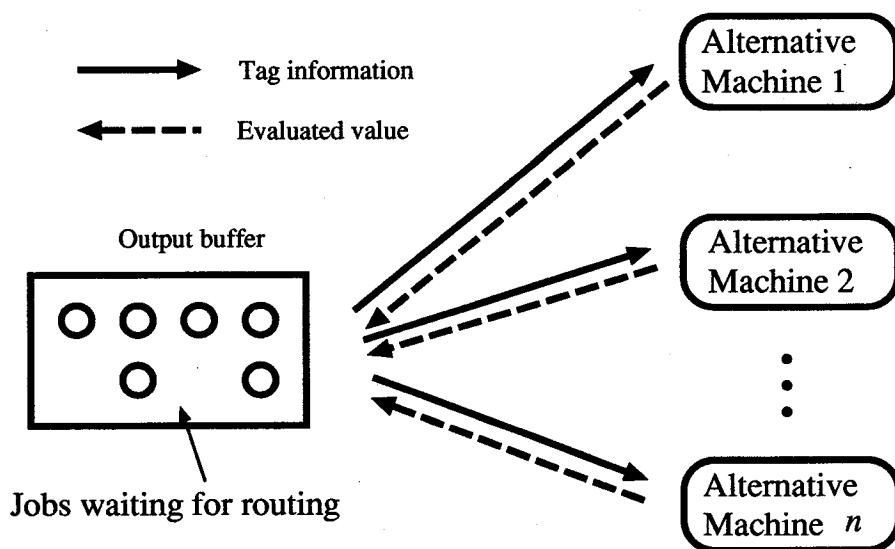


Fig. 6.5: Information flows for routing

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行なう機械を決定し、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、全ての次工程に通知し、予約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、次節以降で述べる。

C. 予約解除の方法

すでにすべての入力バッファが予約されている場合であっても、特急ジョブなど緊急度の高いジョブが発生すれば、入力バッファをそのジョブに優先して割り当てる必要がある。そこで、先に予約をしていたジョブの予約を解除し、新たにその緊急度の高いジョブに予約をさせる。

予約の解除は、あるジョブの次工程の代替機械の評価の結果、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 を比較し、もし $v_A > v_i$ であるようなジョブ i (B とする) が存在すれば、そのジョブを予約解除ジョブの候補として記録しておき、ジョブ A に対する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して返す。

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 に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で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。

D. バッファ空き待ちジョブに対する処理

ひとつの工程が終了し、出力バッファに到着したジョブは、次の工程を実行する機械を選定してそれを予約するが、すべての代替機械の入力バッファがすでに満杯で、どの機械も予約できなかった場合、そのジョブは、次の工程を実行する機械の入力バッファを予約できるまで、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” 中のジョブの中から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他の候補となっている機械の状況を調べ、バッファに空きが生じそうな適当な機械がなければ、通知を行なった機械を予約する。空きが生じそうなもので適当なものがあれば、(1) に戻る。

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。

E. 搬送ジョブの決定の際のエージェント間の通信

次の工程の機械の入力バッファを予約したジョブは、直前の工程を行なった機械の出力バッファで搬送されるのを待つ。搬送されるまで出力バッファはジョブで占有されているため、最悪

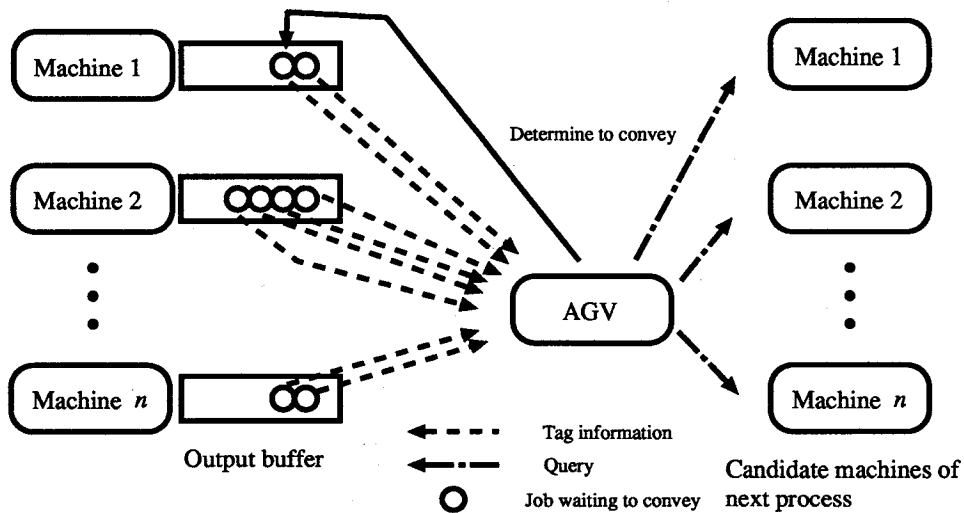


Fig. 6.6: Flow of information for selecting a conveying job.

の場合、出力バッファに空きがないため、その機械による加工を開始できないといったことが起こることもある。従って、搬送するジョブを適切に選択することは重要である。

搬送するジョブの決定における情報の流れを Fig. 6.6 に示す。

本システムにおいて、搬送するジョブの決定は、次のような方法で行なう。

- (1) 搬送車は、各機械の出力バッファ中のジョブで次工程の入力バッファを予約済みのもののタグを参照して、各ジョブの次工程の状況などをもとにそれぞれのジョブを搬送した場合の評価値を求める。
- (2) 評価値の最も高いジョブを、搬送するジョブとし、搬送を実行する。

F. 入力バッファからのジョブ選択の際のエージェント間の通信

入力バッファから次にどのジョブを選んで作業を実行するのかの決定は、その機械の出力バッファの占有状態や、次の工程を行なう機械の入力バッファの占有状態に影響を与える。従って、これらの状況を考慮に入れて次に作業を実行するジョブを選択する必要がある。なお、本システムでは、入力バッファ中にすでに存在しているジョブは、必ずその機械で作業が行なわれると仮定している。

入力バッファからのジョブの選択における情報の流れを Fig. 6.7 に示す。

入力バッファからのジョブの選択の方法は、次に示す通りである。

- (1) 今まで行なっていた工程が終了し、次の処理を行なうことができる機械は、入力バッファ中の全ジョブについて、各ジョブの次工程の状況を参照して、現時点での評価値を求める。
- (2) 評価値の高いジョブを選択し、加工を開始する。

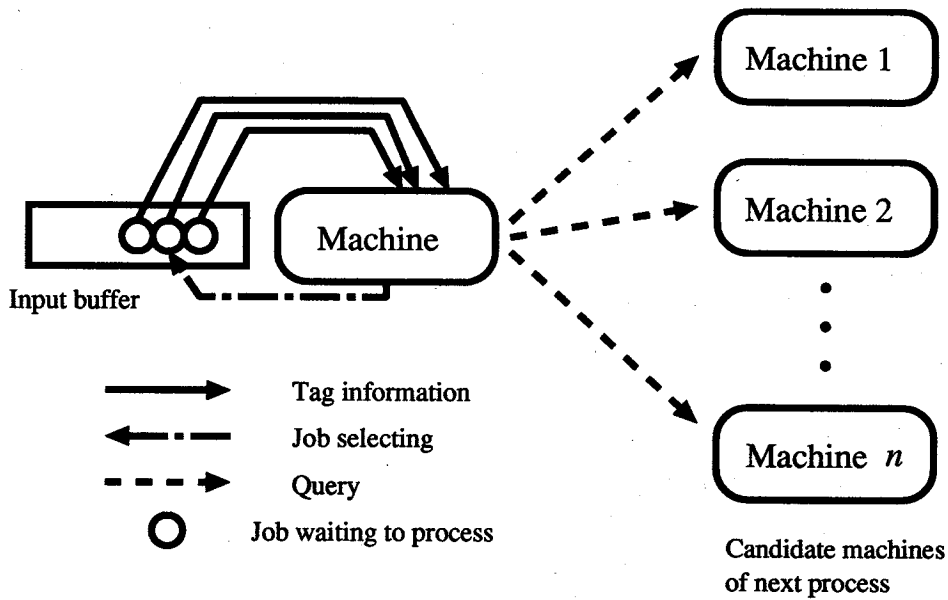


Fig. 6.7: Flow of information for selecting a job from input buffer.

なお、出力バッファに加工を完了したジョブの入る余地がない場合は、次の加工を開始しない。

6.2.3 システム構成

本章で提案した分散環境下におけるスケジューリングシステムの評価を行なうため、本章では、Fig. 6.8に示すような実験システムを構成する。

この実験システムは、各機械に分散したいくつかのスケジューリングプロセスと生産システムのシミュレータよりなる。各スケジューリングプロセスは、各々平等であり、各加工機械の制御用計算機上のプログラムがその役割を果たすことを想定している。

A. シミュレータ

シミュレータは、実際の生産工場を模擬する。本システムでは、シミュレータとして、汎用シミュレーション言語 SLAM II [15, 16] を用いている。シミュレータは、ジョブが入力バッファや出力バッファなど、スケジューリング決定が必要なところに到着すると、SLAM II のユーザ・サブルーチン ALLOC を介して、それぞれに対応するスケジューラが呼び出される。その結果は再びシミュレータに返され、以降のシミュレーションが再開される。なお、SLAM II についての詳細は文献 [16] を参照されたい。

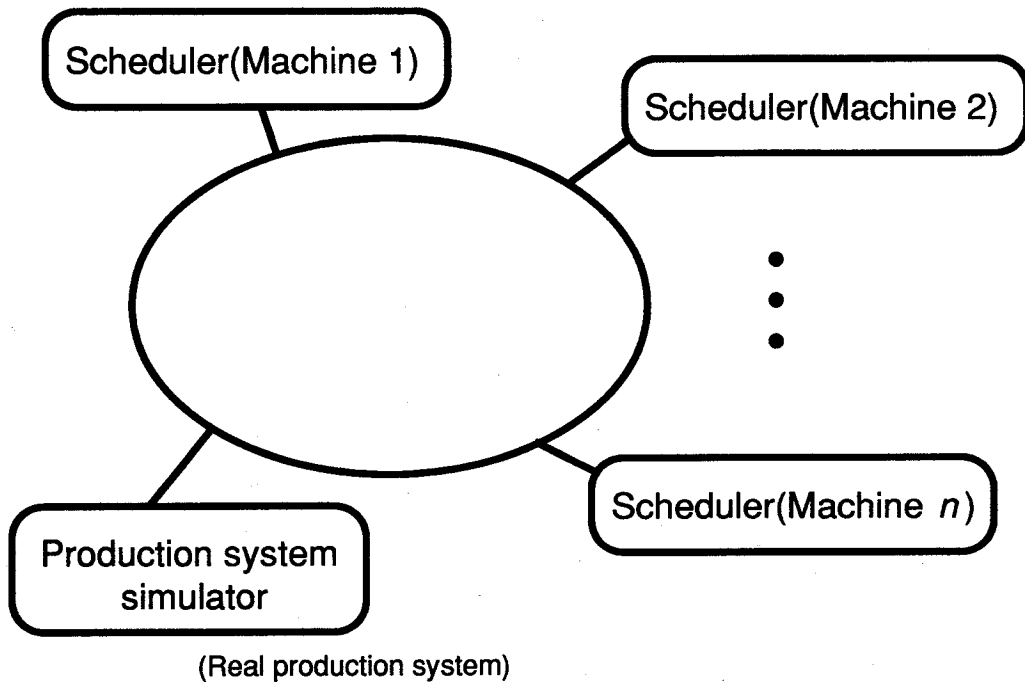


Fig. 6.8: System configuration [14].

B. スケジューラ

スケジューラは、各加工機械毎に、平等なエージェントとして配置する。シミュレータから競合解決の要請があると、関連のあるエージェントが互いにデータをやりとりすることによって現在の状況进行评估し、それに基づいてスケジューリング決定を行ない、競合を解消していく。その結果は、SLAM 言語と Fortran とのインタフェースルーチン ALLOC [16] を介してシミュレータに渡される。

6.3 スケジューリングに用いる評価構造

多品種少量生産のスケジュールの作成においては、一般には、ジョブごとにスケジュール要求は異なっている。したがって、多品種少量生産で要求される柔軟性に応えるためには、これまで述べてきたように、システムを分散型にして柔軟性、耐故障性を増すことに加え、このようなジョブごとに課せられる要求も高いレベルで満足することのできるスケジューリングのための決定機構が望まれる。

本節では、6.2節で述べた分散型スケジューリングシステムにおいて、スケジューリングを行なう時に各エージェントがどのような方法でシステムの状況进行评估し、スケジューリングを行なうのかについて述べる。

6.3.1 競合解決と評価方法

A. グローバルな評価とローカルな評価

一般に、分散型システムでは、システムのグローバルな状態を各エージェントが直接に知ることはできない。したがって、各エージェントがグローバルな情報を知るためには、全エージェントの持つローカルな情報を集めて統合する必要がある。しかし、このためには、高いコストを払って通信を行なう必要があるため、評価を行なう際の項目についても、グローバルな情報に基づくものとローカルな情報に基づくものとに分類し、なおかつグローバルな情報はできるだけ用いないようにすることが望ましい。

本章では、スケジュール発生のために必要な、ルーティングの決定、搬送ジョブの決定、入力バッファから選択するジョブの決定のそれぞれについて、決定を行なうための評価項目を設定し、それらをローカルなものとしてグローバルなものに二分した。さらに、各評価項目についてジョブ毎にそれぞれの重要度を設定することにより、ジョブ毎に異なるスケジューリング要求を満たすことができるようにした。以下に、機械が競合している場合とジョブが競合している場合に分けて、評価値決定までの処理の流れの概要を示す。

機械の競合を解消する場合 (ルーティングの決定):

(1) ジョブのタグ情報を参照

スケジューリング決定を行なおうとしている加工機械は、対象としているジョブのタグ情報を読み込む。

(2) 関連する機械へのローカルな評価の要請

タグ情報から、そのスケジューリング決定に関連する加工機械 (すなわちルーティング先の各代替機械) を求め、それらの加工機械にそのジョブのタグ情報を送る。

(3) ジョブに対するローカルな評価値の算出 (各機械について)

(2) で送られてきたタグ情報を受け取った各代替機械は、そのタグ情報及びその機械の持っているローカルな情報を用いてローカルな評価項目に対する評価値を算出し、送られてきたタグ情報に記録してある各評価項目に対する重要度を係数とする線形和を求め、それをそのジョブに対する各代替機械のグローバルな評価値とする。

(4) 関連する機械からのローカルな評価値の収集

(2) でタグ情報を受け取った機械から、それぞれの機械が算出した、各代替機械についてのローカルな評価値を収集する。

(5) そのジョブに対する総合的な評価値の算出

(3) で求めたローカルな評価値より、それぞれの機械におけるそのジョブに対する総合的な評価値を決める。

(6) 選択する機械の決定

(5) の、各機械についての総合的な評価値を比較し、評価値が最大の機械を、選択する機械として決定する。

ジョブの競合を解消する場合 (搬送するジョブの決定, 入力バッファから選択するジョブの決定):

(1) 各ジョブのタグ情報を参照

ジョブの競合を解決しようとする加工機械は、競合している全てのジョブのタグ情報を読み込む。

(2) グローバルな評価値算出のための情報収集

グローバルな評価値を決定するために、すべての加工機械から情報を収集する。

(3) 各ジョブに対するグローバルな評価値の算出

(2) で収集した情報を用いて、グローバルな評価項目に対する評価値を算出し、さらに各ジョブのスケジューリング要求を反映して設定される、各ジョブのタグ情報に記録してある各評価項目に対する重要度を係数とする線形和を求め、その値を各ジョブに対するグローバルな評価値とする。

(4) 各ジョブの関連する機械へのローカルな評価の要請

各ジョブのタグ情報から、それぞれのジョブについての、そのスケジューリング決定に関連する加工機械を決定し、それらの加工機械にそれぞれのジョブのタグ情報を送る。

(5) 各ジョブに対するローカルな評価値の算出 (各機械について)

(4) で送られてきたタグ情報を受け取った各代替機械は、そのタグ情報及びその機械の持っているローカルな情報を用いて、タグ情報を送ってきたジョブすべてに対して、その機械についてのローカルな評価項目に対する評価値を算出する。つぎに、送られてきたタグ情報に記録してある各評価項目に対する重要度を用いてそれらの線形和を求め、その値をそのジョブに対する、その機械におけるローカルな評価値とする。

(6) 関連する機械からの各ジョブに対するローカルな評価値の収集

(4) で算出した各ジョブの、各機械に対する評価値を、ローカルな評価を要請した機械に返す。

(7) 各ジョブに対する総合的な評価値の算出

各ジョブについて (5) で、各代替機械が返してきたローカルな評価値のうち、最大のものをそのジョブのローカルな評価値とする。これと、(3) で求めたグローバルな評価値より、それぞれのジョブに対する総合的な評価値を決める。

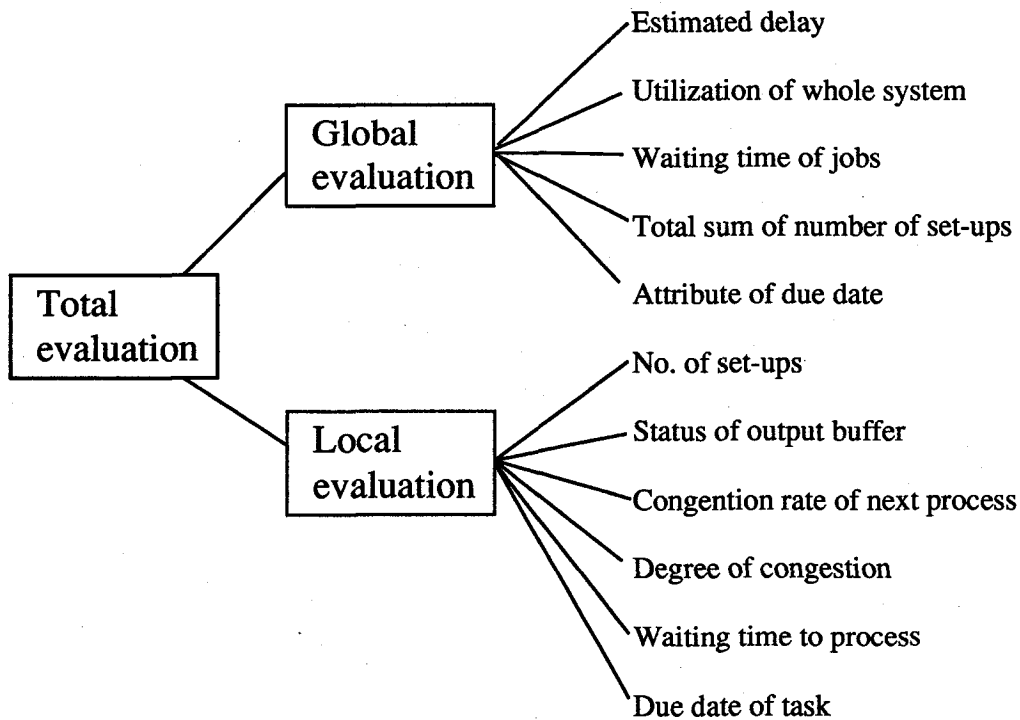


Fig. 6.9: Structure of evaluation for resolving conflicts in routing.

(8) 選択するジョブの決定

(7) の、各ジョブについての総合的な評価値を比較し、評価値が最大のジョブを、選択するジョブとして決定する。

以上のようにして各代替案についての評価値を求め、評価値が最大となる代替案を最終的なスケジュール決定として採用する。

B. ルーティング

ルーティングでは、競合しているのは次工程を行なうことのできる複数の代替機械である。ルーティングのために設定した評価構造を Fig. 6.9 に示す。

加工時間は、各代替機械で異なるのが一般的であり、短い機械を選択する方がジョブのシステム内滞留時間が短くなる。予想納期遅れは、現時点で予想される納期遅れである。バッファ中の同種のジョブの数は、特に段取り替えを少なくしたい時などには考慮することが必要である。出力バッファの占有率は、この値が高い場合、出力バッファが満杯になってしまいその機械が使用不能になるおそれがある。予想作業待ち時間は、その機械における現時点での待ち時間である。待ち時間の長い機械にルーティングを行なうと、最終的に部品のシステム内滞留時間が長くなる。納期属性は、そのジョブの納期の緊急度であり、特急であるほど高い評価値となる。

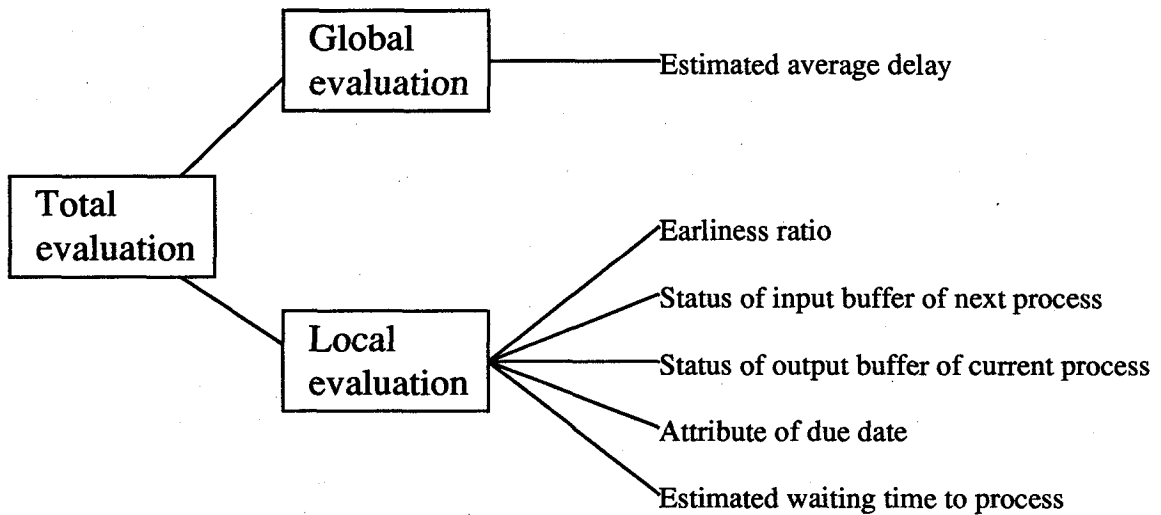


Fig. 6.10: Structure of evaluation for resolving conflicts in selecting a conveying job.

C. 搬送ジョブの決定

搬送ジョブの決定では、競合しているのはジョブである。搬送ジョブの決定のために設定した評価構造を Fig. 6.10 に示す。

グローバルな評価項目である予想平均納期遅れは、そのジョブが全体の納期遅れに大きな影響を与える場合などに考慮することができる。納期余裕率は、そのジョブが納期まで余裕がないものを優先するために考慮する。また、搬送待ちのジョブは出力バッファを占有しているので、出力バッファの占有率が高くなってきた場合は、そのバッファに存在するジョブを早く他の機械に搬送し、出力バッファを解放する必要がある。さらに、搬送先の入力バッファが満杯に近い場合は、そのジョブの搬送を抑止する。ジョブ待ち時間は、ジョブがシステムに投入されてから今までに、バッファで加工を待っていた時間の和であり、待ち時間の長いジョブは、システム内滞留時間を低くするという意味で優先することができる。

D. 入力バッファから選択するジョブの決定

入力バッファから選択するジョブの決定では、競合しているのはジョブである。入力バッファから選択するジョブの決定のために設定した評価構造を Fig. 6.11 に示す。

“全システム稼働率”は、例えば、稼働率の低い時に優先して実行したいジョブがある場合などに必要な評価項目であり、全ジョブ平均待ち時間は、加工時間が非常に長いなど、他のジョブの待ち時間に大きな影響を与えるジョブの実行を制御するために設定した。

ローカルな評価項目としては、“次工程における予想待ち時間”は、例えば、次工程における待ち時間が長い場合には、他のジョブを優先させるといった制御を可能にしている。

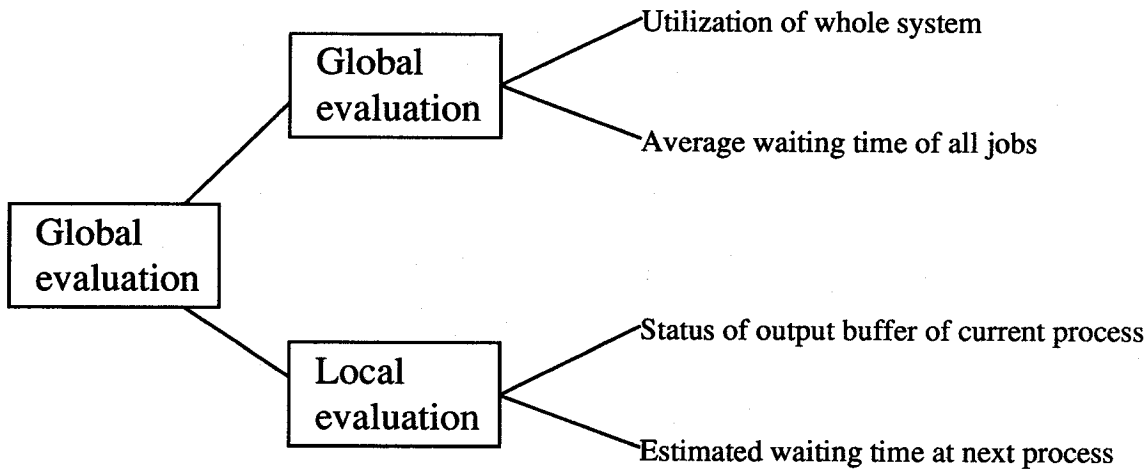


Fig. 6.11: Structure of evaluation for resolving conflicts in selecting a job from an input buffer.

6.3.2 各評価項目の定式化

ここでは、前節で述べた様々なスケジューリング決定において用いられる評価項目を定式化し、さらにそれらの値を決める各パラメータの算出方法について述べる。

A. 変数の説明

各評価項目の定式化の前に、以下の説明に用いる変数の意味について簡単に説明する。

- $t(j, p, i)$: ジョブ j の工程 p のうち、機械 i を用いるものの作業時間
- \bar{t}_{jp} : ジョブ j の工程 p の平均作業時間 $= \sum_i t(j, p, i) / a(j, p)$
- D_j : ジョブ j の現在までに完了した工程の集合
- M : 機械の数
- n : 全ジョブの数
- t : 現在時刻
- t_0 : 加工開始時刻
- s_j : ジョブ j のシステム投入時刻
- $C(M_i, d)$: 機械 i の時刻 d までの段取り替え回数
- e_j : ジョブ j の最早完了時刻
- d_j : ジョブ j の納期
- $a(j, p)$: ジョブ j の工程 p の代替機械の数

- I_i : 機械 i の入力バッファに存在するジョブの集合
 M_{next} : 現在ルーティングを考慮している機械の番号,
 またはルーティング先の機械の番号
 P_{next} : 次の工程
 M_{now} : 現工程の機械の番号
 P_{now} : 現在の工程
 $b_{in}(i)$: 機械 i の占有されている入力バッファの数
 $b_{out}(i)$: 機械 i の占有されている出力バッファの数
 $B_{in}(i)$: 機械 i の入力バッファの総数
 $B_{out}(i)$: 機械 i の出力バッファの総数

B. 各評価項目の定式化

ここでは、前述の評価項目それぞれについて、具体的に定式化を行なう。

グローバルな評価項目:

(1) 全システム稼働率

$$\frac{\text{現在までの全加工時間の和}}{\text{機械の数} \times (\text{現在時刻} - \text{加工開始時刻})} = \frac{\sum_j \sum_{(p,i) \in D_j} t(j,p,i)}{M(t-t_0)}$$

(2) 予想平均納期遅れ

$$\frac{\text{全ジョブの予想納期遅れの和}}{\text{全ジョブ数}} = \frac{\sum_j \max\{0, e_j - d_j\}}{n}$$

(3) 全ジョブ平均待ち時間

$$\frac{\text{現時点までの全ジョブの待ち時間の和}}{\text{全ジョブ数}} = \frac{\sum_j \{(t - s_j) - \sum_{(p,i) \in D_j} t(j,p,i)\}}{n}$$

ローカルな評価項目:

(1) ジョブ待ち時間

$$\begin{aligned} & (\text{現在時刻} - \text{当該ジョブのシステム投入時刻}) - \text{現在時刻までの実加工時間} \\ &= (t - s_j) - \sum_{(p,i) \in D_j} t(j,p,i) \end{aligned}$$

(2) 予想納期遅れ

$$\max\{0, (\text{最早完了時刻} - \text{納期})\} = \max\{0, e_j - d_j\}$$

(3) 段取り替え回数

$$\text{その機械での段取り替え回数} = C(M_{next}, t)$$

(4) 次工程の出力バッファの占有率

$$\frac{\text{次工程の占有されている出力バッファの数}}{\text{次工程の出力バッファの総数}} = \frac{b_{out}(M_{next})}{B_{out}(M_{next})}$$

(5) 現工程機械の加工時間

$$\text{現工程機械の加工時間} = t(j, P_{now}, M_{now}), j \text{は現在考慮しているジョブ}$$

(6) 次工程機械の加工時間

$$\text{次工程機械の加工時間} = t(j, P_{next}, M_{next}), j \text{は現在考慮しているジョブ}$$

(7) 予想作業待ち時間

$$\text{入力バッファ到着から加工開始までの予想待ち時間} = \sum_{j \in I_{M_{next}}} t(j, P_{next}, M_{next})$$

(8) 次工程での予想待ち時間

$$\begin{aligned} & \text{次工程候補機械の入力バッファ中のジョブの加工時間の和} \\ & = \sum_{j \in I_{M_{next}}} t(j, P_{next}, M_{next}) \end{aligned}$$

(9) 納期余裕率

$$\frac{\text{納期までの残り時間}}{\text{各残り工程の平均加工時間の和}} = \frac{d_j - e_j}{\sum_{p \in \bar{D}_j} \bar{t}_{jp}}$$

(10) 現在占有している出力バッファの占有率

$$\frac{\text{現工程の占有されている出力バッファの数}}{\text{現工程の出力バッファの総数}} = \frac{b_{in}(M_{now})}{B_{in}(M_{now})}$$

(11) 次工程の候補機械の入力バッファの占有率

$$\frac{\text{次工程の占有されている入力バッファの数}}{\text{次工程の入力バッファの総数}} = \frac{b_{in}(M_{next})}{B_{in}(M_{next})}$$

(12) 納期属性

各ジョブのタグ情報より得る。

6.4 数値例

本節では、フレキシブル生産の例として半導体生産プロセス [17] を取り上げ、そのリアルタイムスケジューリングを本システムを用いて行ない、その結果を用いて本システムの評価を行なう。

6.4.1 半導体生産プロセスの概要

半導体生産プロセス (ウエハプロセス:ウエハ上に素子を作成するプロセス) は、(1) エピタキシャル成長, (2) 酸化, (3) ホトレジスト加工 (ホトレジスト塗布, 露光, 現像, エッチング), (4) 選択拡散, (5) イオン注入, (6) CVD (chemical vapour deposition), (7) 蒸着等々の複雑な一連のプロセスによって作成される。

半導体生産プロセス中の生産装置は、Fig. 6.12に示すように、複数の装置をひとまとめにしたショップを単位として管理されている。それぞれのショップは、類似した装置を集めて構成されている。ジョブは、各ショップにおいて加工されるまで、ショップごとに設置されるバッファで待

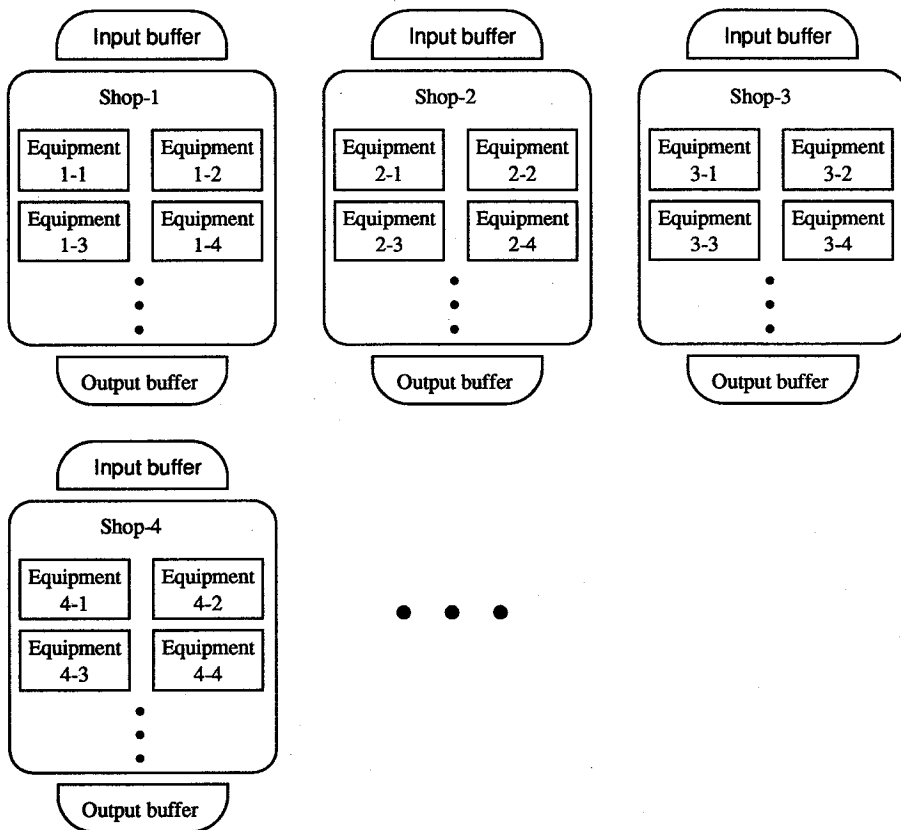


Fig. 6.12: System configuration of production equipments.

機し、ショップ内の適当な装置に割り当てられて加工が行なわれた後、そのショップの出力バッファで次の処理を行なうショップに搬送されるまで待つ。

ウエハプロセスの工程数は少ないもので 20 以上、多いものでは 50 以上におよび、ウエハの加工に入ってからチップ状の IC 片ができるまでに短くて 1 週間、長いときは 6 週間近くの時間がかかる。このように、製造プロセスは、極めて複雑であり、また、流れるものの大きさ自体が小さなものであるため、同時に非常に多品種のジョブを流すことが可能である。したがって、生産システムとしては、非常に大規模で複雑なものとなり、その制御も非常に困難なものとなる。

6.4.2 モデル化

前節では、半導体生産プロセスについて、概要を簡単に説明したが、ここでは、本システムを適用する数値例として、どのようにこれをモデル化したかについて述べる。

上で述べた半導体生産プロセスは、大きく分けて、(1) ホトリソグラフィ、(2) デポジション、(3) 拡散、イオン打ち込み、の 3 種類の工程から構成されると考えられる。そこで、ウエハは、**Fig. 6.13** に示すように、これらの 3 つのプロセスを何度も行き来させることにより、製品として完成すると仮定する。また、各ショップをスケジューリングにおける生産環境の構成単位とみなす。従って、本例では、このショップに各ジョブ (ウエハ) をスケジューリング目的に従って適切に割り当てることを目的とする。

段取り替え時間はジョブによらず一定とし、各プロセスの実行時間に含める。また、プロセス間の搬送は、一度に 1 ジョブのみ行なえ、所要時間は一定であると仮定する。

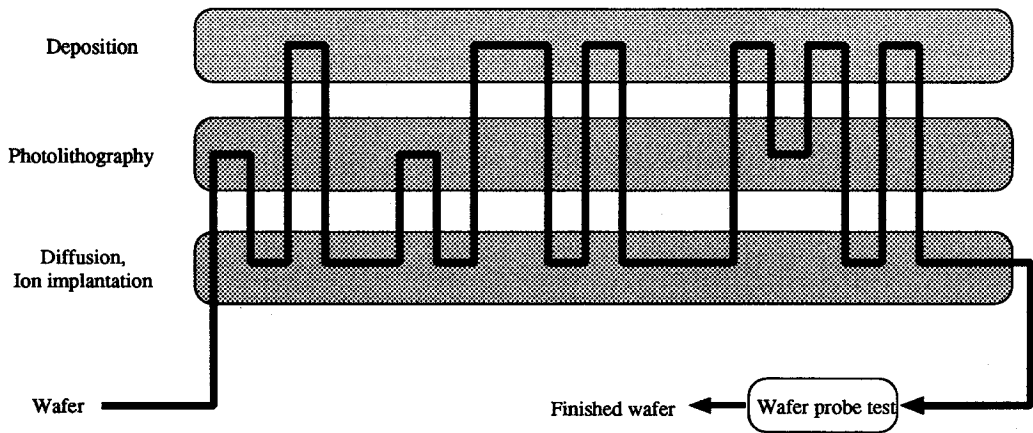


Fig. 6.13: An example of model of semiconductor process.

6.4.3 本例で用いる生産環境

生産設備は、ホトリソグラフィを行なう装置を集めたショップが3 (S1 ~ S3), デポジションを行なう装置を集めたショップが3 (S4 ~ S6), 拡散およびイオン打ち込みを行なう装置を集めたショップが4 (S7 ~ S10) の計 10 のショップ, および 1 台の搬送車によって構成されている。搬送車は一度に 1 つのジョブをショップからショップへ搬送する。

この生産設備で, Table 6.1に示すような 10 種類のジョブ (J1 ~ J10) を処理する。括弧内はそれぞれの処理に要する時間を表わす。ただし, P_1 はホトリソグラフィ, P_2 は拡散およびイオン注入, P_3 はデポジションを表わすものとする。

6.4.4 出力結果

ここでは, 数値例 1 として特急ジョブが発生した場合, 数値例 2 として段取り替えをなるべく低く押えたい場合, 数値例 3 として待ち時間を少なくしたい場合を取り上げる。

A. 数値例 1

ジョブ J1 ~ J9 のロットサイズはすべて 3 とし, 納期は 200.0 に設定する。また, J8 は他のジョブよりも若干優先して作業を行ないたく, 納期属性である特急度を 3 に設定する。なお他のジョブは特急度 0 である。J1 ~ J9 は時刻 0.0 に投入され, その後時刻 70.0 にロットサイズ 3 の特急ジョブ J10(特急度 10) が納期を 200.0 として発生したとする。

ここでは, 比較のために, 特急ジョブが発生しなかった場合を並べて示す。

スケジューリング目的は, “各ジョブの納期を満たす” こととする。これに従って, 各ジョブのスケジューリング決定のための評価項目の重要度は, 加工時間, 予想納期遅れ, 納期余裕率, 各バッファの占有率, 納期属性は 1.0 とし, それ以外は 0.0 とする。

Table 6.1: Processing information of each job

| Job | Process | Shop name(processing time) |
|-----|----------------|---------------------------------------|
| J1 | P ₁ | S1(13.0), S2(10.0), S3(9.0) |
| | P ₂ | S7(12.0), S8(10.0), S9(9.0), S10(8.0) |
| | P ₁ | S1(15.0), S2(13.0), S3(12.0) |
| | P ₂ | S7(8.0), S8(7.0), S9(7.0), S10(6.0) |
| | P ₃ | S4(12.0), S5(11.0), S6(10.0) |
| | P ₂ | S7(7.0), S8(7.0), S9(7.0), S10(7.0) |
| | P ₁ | S1(14.0), S2(9.0), S3(8.0) |
| | P ₂ | S7(12.0), S8(10.0), S9(9.0), S10(8.0) |
| | P ₃ | S4(8.0), S5(8.0), S6(8.0) |

| Job | Process | Shop name(processing time) |
|-----|----------------|-------------------------------------|
| J2 | P ₁ | S1(8.0), S2(7.0), S3(6.0) |
| | P ₂ | S7(5.0), S8(5.0), S9(5.0), S10(5.0) |
| | P ₃ | S4(8.0), S5(8.0), S6(6.0) |
| | P ₂ | S7(9.0), S8(8.0), S9(8.0), S10(7.0) |
| | P ₁ | S1(9.0), S2(8.0), S3(8.0) |
| | P ₃ | S4(12.0), S5(9.0), S6(8.0) |

| Job | Process | Shop name(processing time) |
|-----|----------------|---|
| J3 | P ₁ | S1(14.0), S2(12.0), S3(10.0) |
| | P ₂ | S7(11.0), S8(9.0), S9(8.0), S10(7.0) |
| | P ₁ | S1(15.0), S2(15.0), S3(13.0) |
| | P ₂ | S7(18.0), S8(16.0), S9(15.0), S10(13.0) |
| | P ₃ | S4(8.0), S5(7.0), S6(7.0) |

| Job | Process | Shop name(processing time) |
|-----|----------------|-------------------------------------|
| J4 | P ₁ | S1(8.0), S2(8.0), S3(7.0) |
| | P ₃ | S4(9.0), S5(8.0), S6(8.0) |
| | P ₂ | S7(8.0), S8(7.0), S9(6.0), S10(6.0) |
| | P ₃ | S4(10.0), S5(10.0), S6(8.0) |
| | P ₂ | S7(9.0), S8(9.0), S9(9.0), S10(9.0) |
| | P ₁ | S1(9.0), S2(8.0), S3(7.0) |
| | P ₃ | S4(5.0), S5(5.0), S6(5.0) |

| Job | Process | Shop name(processing time) |
|-----|----------------|--|
| J5 | P ₁ | S1(10.0), S2(9.0), S3(9.0) |
| | P ₂ | S7(13.0), S8(12.0), S9(10.0), S10(8.0) |
| | P ₃ | S4(7.0), S5(6.0), S6(6.0) |
| | P ₂ | S7(8.0), S8(8.0), S9(7.0), S10(7.0) |
| | P ₃ | S4(9.0), S5(9.0), S6(8.0) |

| Job | Process | Shop name(processing time) |
|-----|----------------|--|
| J6 | P ₁ | S1(10.0), S2(9.0), S3(8.0) |
| | P ₂ | S7(12.0), S8(12.0), S9(10.0), S10(8.0) |
| | P ₁ | S1(8.0), S2(7.0), S3(6.0) |
| | P ₃ | S4(15.0), S5(13.0), S6(7.0) |
| | P ₂ | S7(10.0), S8(9.0), S9(8.0), S10(8.0) |
| | P ₃ | S4(10.0), S5(9.0), S6(8.0) |

| Job | Process | Shop name(processing time) |
|-----|----------------|---|
| J7 | P ₁ | S1(12.0), S2(12.0), S3(11.0) |
| | P ₂ | S7(12.0), S8(11.0), S9(10.0), S10(10.0) |
| | P ₁ | S1(14.0), S2(13.0), S3(10.0) |
| | P ₂ | S7(8.0), S8(7.0), S9(7.0), S10(7.0) |
| | P ₁ | S1(9.0), S2(8.0), S3(8.0) |

| Job | Process | Shop name(processing time) |
|-----|----------------|--------------------------------------|
| J8 | P ₁ | S1(11.0), S2(10.0), S3(10.0) |
| | P ₃ | S4(13.0), S5(12.0), S6(12.0) |
| | P ₁ | S1(9.0), S2(8.0), S3(7.0) |
| | P ₂ | S7(10.0), S8(9.0), S9(9.0), S10(8.0) |
| | P ₁ | S1(14.0), S2(14.0), S3(13.0) |
| | P ₃ | S4(10.0), S5(10.0), S6(10.0) |

| Job | Process | Shop name(processing time) |
|-----|----------------|-------------------------------------|
| J9 | P ₁ | S1(9.0), S2(9.0), S3(9.0) |
| | P ₃ | S4(10.0), S5(10.0), S6(10.0) |
| | P ₂ | S7(8.0), S8(8.0), S9(8.0), S10(8.0) |
| | P ₃ | S4(9.0), S5(9.0), S6(9.0) |

| Job | Process | Shop name(processing time) |
|-----|----------------|---|
| J10 | P ₁ | S1(13.0), S2(12.0), S3(12.0) |
| | P ₃ | S4(16.0), S5(15.0), S6(13.0) |
| | P ₁ | S1(11.0), S2(11.0), S3(10.0) |
| | P ₂ | S7(12.0), S8(12.0), S9(10.0), S10(10.0) |
| | P ₁ | S1(8.0), S2(7.0), S3(7.0) |

計算結果:

Fig. 6.14 Result (a) に, 特急ジョブのない場合のシミュレーション結果を示す. この例では, ほとんどのジョブの納期が満たされている. また, ジョブ J8 は優先的に加工を行なっているため, 他のジョブに比べてかなり早く加工が完了していることがわかる.

次に, 同じ状況の下で, 時刻 70.0 にロットサイズ 3 の特急ジョブ J10 を納期を 200.0 として投入した場合の結果を Fig. 6.14 Result (b) に示す. この場合, 特急ジョブである J10 は, 時刻 70.0 に投入されたにもかかわらず, 僅かの納期遅れを生じただけで加工を完了している. しかし, 他のジョブはそのしわ寄せを受け, 特に J1, J2, J7 は大きな納期遅れを生じている. この場合でも, J8 は優先度が高いため, 納期が特急ジョブがない場合とほとんど変わっていない.

Result (a) および (b) の全システム稼働率およびジョブ平均待ち時間を Table 6.2 に示す. 全システム稼働率がそれぞれ約 69% および 65% と低いのは, 本数値例が連続操作の場合ではなく, 生産システムの始動時から考慮しているので, 最初の方はほとんどのジョブが処理を行っていないことが原因であると考えられる.

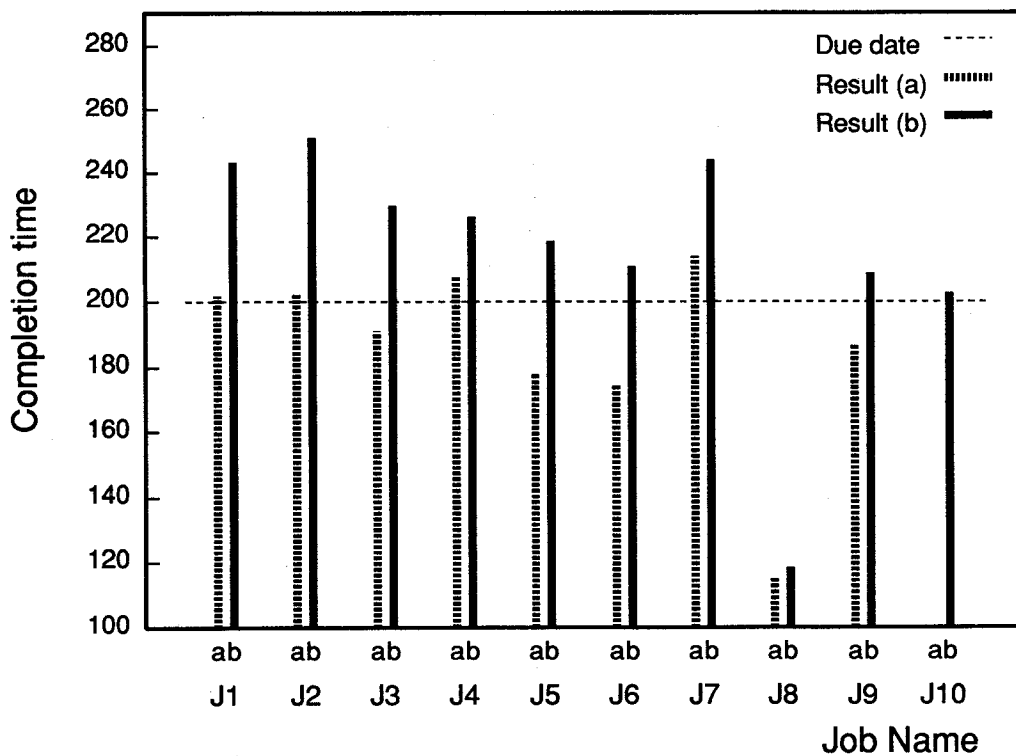


Fig. 6.14: Comparison of schedules depending upon the existence of an express job [14].

Table 6.2: Statistical comparisons on numerical example 1 [14].

| | Result (a) | Result (b) |
|----------------------------------|------------|------------|
| Utilization of whole system | 0.69 | 0.65 |
| Average waiting time of all jobs | 148.1 | 172.1 |

B. 数値例 2

ジョブ J1 ~ J9 の 9 つのジョブを処理する。ジョブ J1 のロットサイズは 6, J2 ~ J9 のロットサイズは 3 とし、各ジョブの納期は、J1 が 250.0, J2 が 210.0, J3 が 220.0, J4 が 250.0, J5 が 180.0, J6 が 200.0, J7 が 250.0, J8 が 230.0, J9 が 200.0 とする。スケジューリング方針は納期重視とする。ただし、J1 は納期を犠牲にしても段取り替え回数をできるだけ少なくする。

この例でも、段取り替え回数を考慮しない場合の結果との比較を行なった。

計算結果:

段取り替えの回数をスケジューリング決定の考慮に入れない場合の結果を Fig. 6.15 Result (a) に示す。この例では、いずれのジョブも段取り替えを考慮しないため、同じ種類のジョブの数および段取り替えの有無の 2 つの評価項目に対する重要度はいずれも 0.0 とした。また、納期重視であるので、数値例 1 と同様、加工時間、予想納期遅れ、納期余裕率、各バッファの占有率、納期属性の重要度は 1.0 とし、それ以外は 0.0 とした。

Result (a) では、ジョブ J1 および J8 を除いて、おおむね納期を満たすことができている。また、段取り替え回数をスケジューリング決定の際の考慮に入れない場合、ジョブ J1 の加工を開始する時に段取り替えが必要回数は 31 回であった。

次に、ジョブ J1 のみ、同じ種類のジョブの数および段取り替えの有無の 2 つの評価項目に対する重要度を 10.0 に変更し、段取り替えを極力少なくするようにした場合の結果を Fig. 6.15 Result (b) に示す。

この例では、上の段取り替えを考慮しない場合と比較して、J1 の段取り替え必要回数が 20 回と 30%程度少なくなっているのがわかる。しかし、J1 は、段取り替えが必要である場合は優先度が低くなるため、納期が段取り替えを考慮しない場合よりも遅れている。逆に、J1 の処理を行なうのに段取り替えが必要である時には他のジョブが優先されるため、J1 以外のほとんどのジョブは、J1 の段取り替えを考慮しない場合よりも加工完了時刻が早くなっている。また、Result (a) および (b) の全システム稼働率および平均ジョブ待ち時間を Table 6.3 に示す。

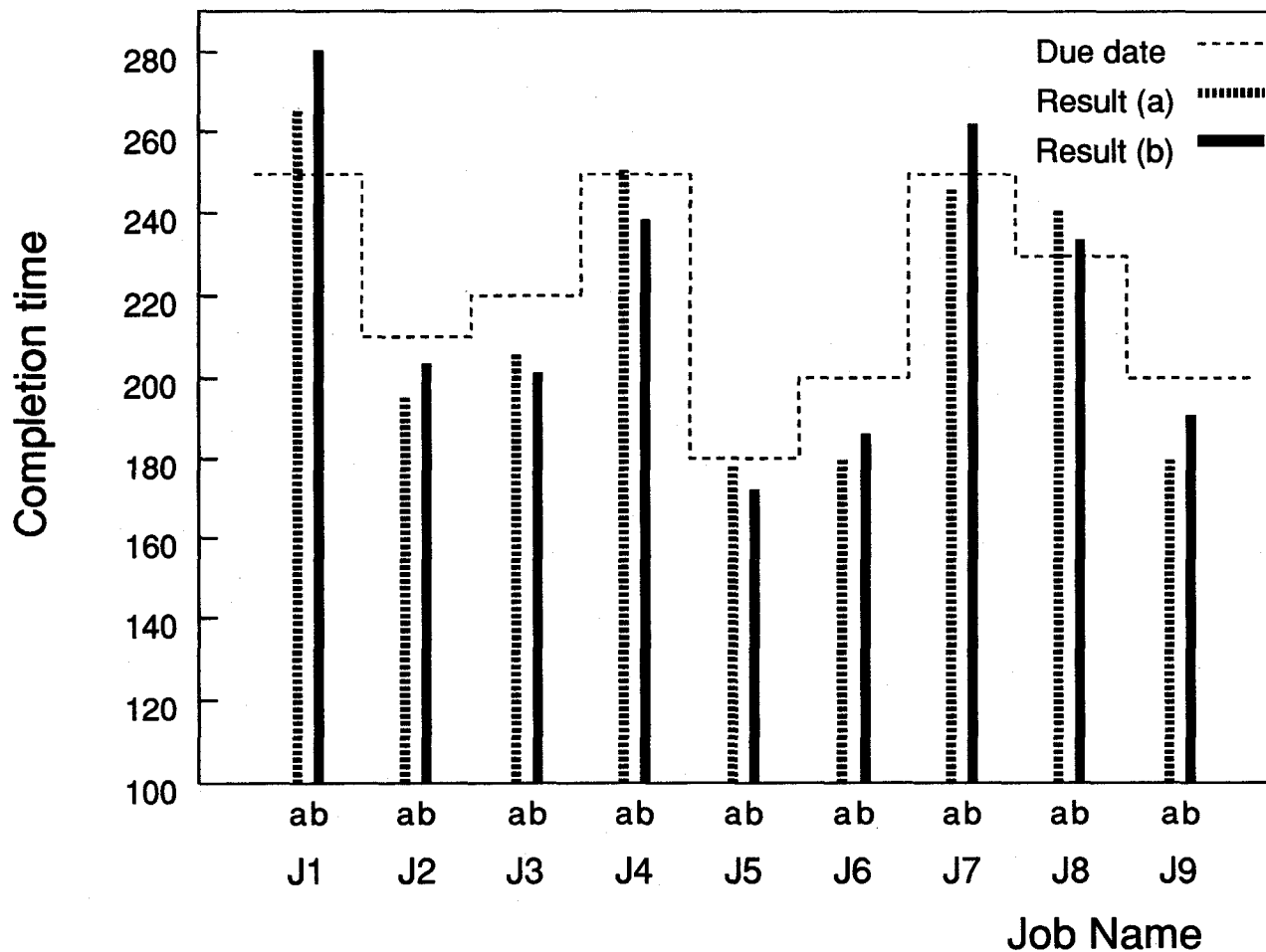


Fig. 6.15: Comparison of schedules depending upon whether the number of set-ups is taken into consideration or not [14].

Table 6.3: Statistical comparison on numerical example 2 [14].

| | Result (a) | Result (b) |
|----------------------------------|------------|------------|
| Utilization of whole system | 0.67 | 0.63 |
| Average waiting time of all jobs | 193.6 | 208.5 |
| No. of set-up | 31 | 20 |

C. 数値例 3

本例では、前例よりも設定を大規模にして計算を行なった。ジョブ J1 ~ J9 の 9 つのジョブを処理する。ジョブ J1 のロットサイズは 15, J2 のロットサイズは 11, J3 のロットサイズは 12, J4 のロットサイズは 15, J5 のロットサイズは 14, J6 のロットサイズは 9, J7 のロットサイズは 11, J8 のロットサイズは 9, J9 ののロットサイズは 8 とする。また、各ジョブの納期は、J1 が 650.0, J2 が 800.0, J3 が 600.0, J4 が 750.0, J5 が 800.0, J6 が 700.0, J7 が 800.0, J8 が 700.0, J9 が 800.0 とする。スケジューリング方針は納期重視とする。ただし、J7 はなるべく待ち時間が短くなるようにする。

同様に、待ち時間の重要度を他のものと同様にした場合の結果との比較を行なった。

計算結果:

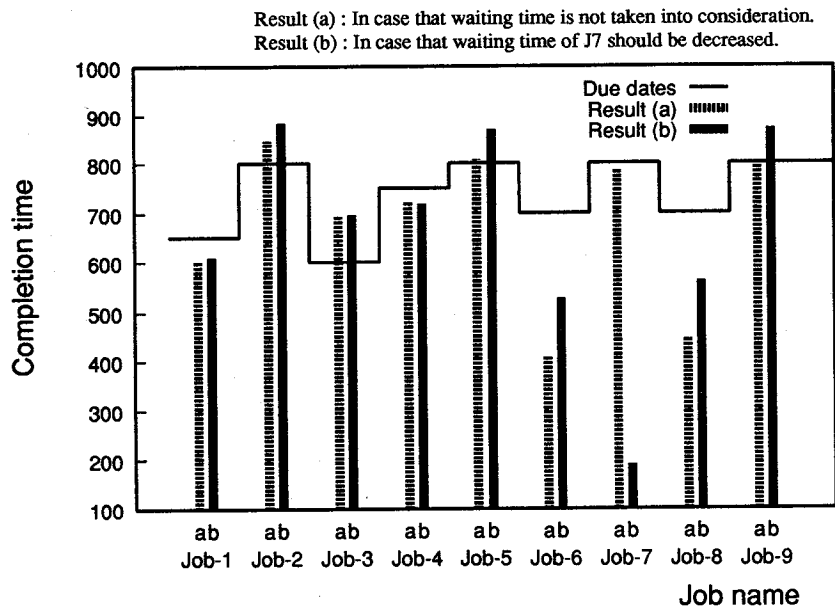
待ち時間を特に考慮に入れない場合の加工完了時刻およびジョブの平均待ち時間のシミュレーション結果を **Fig. 6.16 Result (a)** に示す。

この例では、すべてのジョブが納期重視であるので、加工時間、予想納期遅れ、納期余裕率、各バッファの占有率、納期属性の重要度は 1.0 とし、それ以外は 0.0 とした。

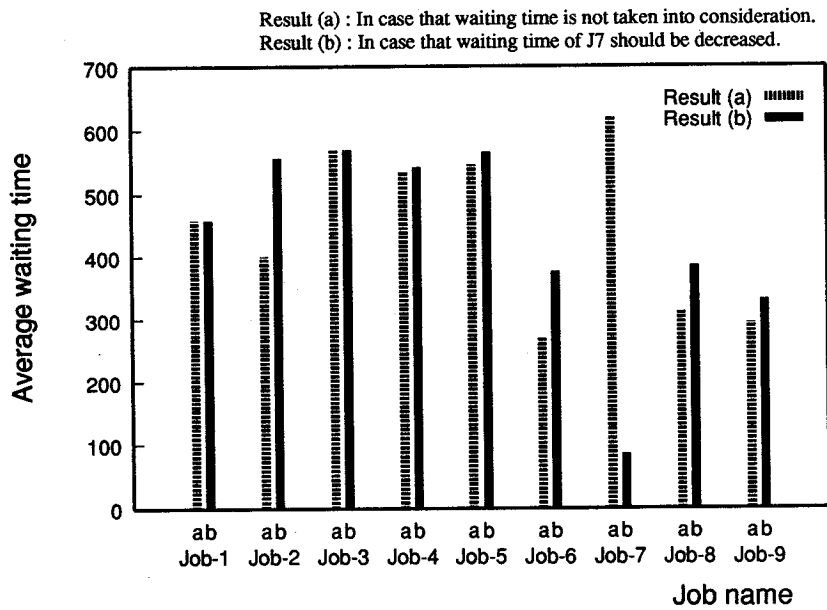
Result (a) では、ジョブ J2, および J3 を除いて、おおむね納期を満たすことができている。

次に、ジョブ J7 のみ、待ち時間が少なくなるような要求があったと想定し、予想作業待ち時間、全ジョブ平均待ち時間、ジョブ待ち時間の各評価項目の重要度を 1.0 から 5.0 に変更した。結果を **Fig. 6.16 Result (b)** に示す。

この例では、納期に関しては、J7 を除いて、待ち時間を考慮しない場合と余り大差がない。J7 は、ルーティングの際は、待ち時間のできるだけ短い機械が選ばれ、また、入力バッファからも優先的に選択されるため、待ち時間を考慮しない場合と比較して、約 86% も待ち時間が短縮され、結果的に納期が非常に早くなっている。また **Table 6.4** から、J7 の待ち時間を優先しても他のジョブの待ち時間にはあまり影響を与えないことがわかる。



(i) Completion time of each job



(ii) Average waiting time of all jobs

Fig. 6.16: Comparison of completion time and average waiting time depending upon whether the waiting time is taken into account or not [14].

Table 6.4: Statistical comparisons on numerical example 3 [14].

| | Result (a) | Result (b) |
|----------------------------------|------------|------------|
| Utilization of whole system | 0.69 | 0.636 |
| Average waiting time of all jobs | 733.3 | 752.8 |

6.5 むすび

本章では、フレキシブル生産の特徴である、柔軟性、不確実性などに対応して、リアルタイムに生産スケジューリングを行なうことのできるスケジューリングシステムを目指して次のような分散型のスケジューリングシステムを提案した。

- (1) 生産システムを構成する各加工機械に、独立した平等なスケジューリングエージェントを配置することにより、スケジューリングのための制御機構を各機械に分散的に配置した。
- (2) 各スケジューリングエージェントがスケジューリングの必要に応じて通信を行なってデータを取り込み、ローカルに決定を行なうことにより、生産システムの構成の変動に対しても柔軟に対応できるようにした。
- (3) スケジューリングのための意思決定は、複数の評価項目を設定し、それらの評価値の線形和をとって、最大の値を持つ代替案を採用するようにした。
- (4) 各評価項目は、ローカルまたはグローバルな評価項目のどちらかに分類した。ローカルな評価項目に対する評価値は各エージェントが持つローカルな情報のみを用いて計算し、グローバルな評価項目に対する評価値は全エージェントから通信を用いて情報を収集して計算した。これらのローカルおよびグローバルな評価値を統合して総合的な評価値とした。
- (5) スケジューリングの際に用いる評価項目の重要度をジョブ毎に設定することにより、スケジューリング目的がジョブによって異なる場合にも、その要求を満たすスケジュールを発生させることを可能にした。

このような構成を持つシステムを試作し、数値実験を行なった結果、以下のような結論を得た。

- (1) 各機械のスケジューリングプロセスをそれぞれの機械上に分散させ、独立したものとして扱うことにより、環境の変化に柔軟に対応できるスケジューリングシステムを構築することができた。
- (2) ジョブそれぞれに、そのジョブについての情報を記録したタグを持たせ、各スケジューリングプロセス間の通信にはこの情報を一元的に用いることにより、各サブシステム間の通信の複雑化を回避することができた。

- (3) ルーティング, 搬送するジョブの決定, 入力バッファから選択するジョブの決定のそれぞれについて, スケジューリングを行なう際に必要な評価項目を定め, 分散環境下におけるグローバルな情報の特殊性を考慮し, それらをグローバルなものと同ローカルなものに明確に分類することにより, 分散環境下でスケジューリングを行なう場合に考慮すべき評価項目を明らかにすることができた.
- (4) タグ情報に, それぞれのジョブに要求されているスケジューリング目的を反映した, 各評価項目の重要度を設定し, それをもとに各スケジューリングプロセスがスケジューリングを行なうことにより, ジョブによって異なるスケジューリング目的を達成することが可能なシステムを構築することができた.
- (5) 評価値の算出に各評価項目の単純な線形和を用いても, スケジューリング目的を反映したスケジューリング結果を得ることができた.

今後の課題としては, 次のようなことがあげられる.

- (1) システムの異常の検出法およびその時の対処法の研究
- (2) スケジューリング決定の基準となる評価項目の一層の充実
- (3) スケジューリング決定のための推論機構の高度化
- (4) 評価項目に対する重要度の学習機能を持たせたチューニングの自動化

また, 本章で設定した“完全に対等なプロセス”による分散システムが常に効率的なわけではなく, 状況によってはある程度の集中化を許した階層型で分散システムを構築した方が効率的な場合もあると考えられる. したがって, 今後の課題として, 実際のシステムに適応する場合, どのようなレベルの分散化が適しているかを推定する手法の開発も行なう必要がある.

第7章 結論

本論文は、多品種少量生産あるいは多品種中量生産を目的としたフレキシブル生産を対象として、実用性に重点をおいて、ヒューリスティックな生産スケジューリングのインテリジェント化に関する研究を行なった結果をまとめたものである。本論文で得られた主な成果をまとめると以下のようなになる。

7.1 オフラインスケジューリング

オフラインスケジューリングに関しては、大別してルールベースに基づいたオフラインスケジューリングおよびスケジューリングのためのヒューリスティクス自動合成に関する研究を行なった。

ルールベースに基づいたオフラインスケジューリング

第2章および第3章では、スケジューリングの近似解法の中で最も実際的なものの一つであるシミュレーションを用いたスケジューリングにおいて、時々刻々の状況に適した優先規則またはジョブを選択するための方略をルールベースとして構成し、そのルールベースを用いて動的に優先規則またはジョブを選択しながらシミュレーションを行なうことにより、インテリジェントなスケジューリングシステムを構築することを試みた。

第2章では、時々刻々の状況に適した優先規則を選択するための方略を if~then~形式のルールベースとして構成することにより効率的なスケジュールを得ることができるスケジューリングシステムを構築した。また、シミュレーション結果を分析することにより、JITを目的として構築されたルールベース中のルールのパラメータを修正する適応機構を持たせて、適応機構がないルールベースと比較した結果、性能の改善が見られた。数値実験を行なった結果、ルールベースに基づいたスケジューリングは、従来の優先規則を固定したシミュレーションを用いたスケジューリングより、少ない労力で効率的な生産スケジュールを得ることができることを確認した。

第3章では、第2章で提案したルールベーススケジューリングにおいて、ルールベースの構築および変更が困難であるという問題点を解決するため、競合した部品選択のための知識ベースを、スケジュール評価モジュール、スケジューリング方針決定モジュールおよびディスパッチングモジュールに分割し、それぞれにおける知識をファジィ・ルールによって記述することにより、

多様なスケジューリング目的に対応可能なスケジューリングシステムを構築した。数値実験を行なった結果、ファジィ推論を用いたスケジューリングシステムは、複雑なスケジューリング目的およびルールベースのチューニングが従来のルールベーススケジューリングシステムより容易であり、より生産現場に即したスケジュールの生成が可能であることを確認した。

スケジューリングのためのヒューリスティクスの自動合成

第4章では、ヒューリスティック探索法のひとつであるフィルター付きビーム探索法において用いられるヒューリスティクスの獲得を容易にするため、スケジューリングシステム自身が探索を繰り返すことによって得られた情報をもとにして、ヒューリスティクスを自動合成するシステムを構築した。開発したスケジューリングシステムを用いて、搬送車スケジューリング問題に関する数値実験を行なった結果、従来、有効なヒューリスティクスが知られていなかったスケジューリング問題に対して、分枝限定法を用いて生成した準最適スケジュールと比較することにより、目的関数(タスクの待ち時間の総和)をほぼ最小化するスケジュールを得ることができていることを確認した。

7.2 リアルタイムスケジューリング

第5章および第6章では、割り込みジョブや工作機械の故障が多く発生するなどの不確実性の多い生産システムを対象としたリアルタイムスケジューリングのインテリジェント化を試みた。一般にリアルタイムスケジューリングでは、オフラインスケジューリングに比べて事前に設定したスケジューリング目的を達成することが困難である。

第5章では、第2章で提案したルールベースをリアルタイムスケジューリングに適用することにより、不確実な生産環境であっても与えられたスケジューリング目的を達成できるようなスケジューリングシステムを開発した。また、部品加工レベルを階層的構造でとらえた階層的確率ペトリネットを用いて不確実状況下の生産システムをモデル化し、リアルタイムスケジューリングのためのルールベースを評価する数値実験を行なった。その結果、優先規則を固定したリアルタイムスケジューリングに比べて効率的なスケジューリングが行なわれる事を確認した。

第6章では、大規模な生産システムを対象にして、分散型リアルタイムスケジューリングシステムを開発した。このスケジューリングシステムでは、スケジューリング目的を含むスケジューリングに必要な情報は、各スケジューリングプロセスおよびジョブに分散して保持させており、大局的な情報は必要に応じて通信によって取得している。そのため、生産システム全体、各工作機械およびジョブという違ったレベルで、各々のスケジューリング目的を設定することができる。仮想的な半導体生産設備のリアルタイムスケジューリングに対する数値実験を行なった結

果,各工作機械およびジョブによって異なったスケジューリング目的を達成することができることを確認した。

7.3 今後の課題

今後の課題としては,次のような問題が残されている。

オフラインスケジューリング:

本論文で提案したルールベーススケジューリングシステムは,スケジューリングのためのルールベースを,スケジューリング目的や対象生産設備の変更に対してチューニングする機能を備えているが,ルールの構造自体はユーザが事前に作成してシステムに与える必要がある。今後は,与えられたスケジューリング目的および対象に適したルール構造を,自動的に獲得するための学習機能の研究を行なう必要がある。スケジューリングのための推論に関しては,現在のシステムではスケジュールを決定する際に,その時点におけるシステムの状態のみに着目して推論している。今後は,過去の履歴および未来状態の予測を取り入れた推論機構を考案していく必要がある。また,現在のシステムは,シミュレーション機能が十分でないため,適用可能な生産形態が限定されている。今後は,シミュレーション機能をさらに高度化させることにより,本システムの適用範囲を拡大する必要がある。

リアルタイムスケジューリング:

本論文で提案したリアルタイムスケジューリングシステムは,ジョブや代替機械のディスパッチングにルールベースまたは生産状態を表わすパラメータの重み付き和を用いている。そのため,ここでもルールベースまたはパラメータの重みを自動的に生成できる学習機能の研究を行なう必要がある。また,リアルタイムスケジューリングを行なうためには,スケジューリングのための情報(知識)をリアルタイムに収集する必要がある。この情報収集機構は,生産システムの制御ソフトウェアに組み込む必要があることを考えると,将来の設備および環境の変化に対して柔軟に対応可能な知識ベース向け情報モデルおよび情報交換プロトコルを構築しておく必要がある。分散型リアルタイムスケジューリングに関しては,実際のシステムに適応する場合,どの程度の分散化が効果的かを推定する手法の開発も行なう必要がある。

謝辞

本研究を行なう機会を与えて頂くとともに、研究遂行に当たり終始懇切な御指導と御鞭撻を賜りました大阪大学基礎工学部システム工学科田村坦之教授に深甚の謝意を表します。また、種々の有益な御教示を賜った大阪大学基礎工学部システム工学科須田信英教授、井口征士教授ならびに馬野元秀助教授に厚く感謝の意を表します。

また、本論文の完成に当たり、御教示を賜りました大阪大学工学部山縣敬一助教授（現広島大学総合科学部教授）および中村豊先生（現筑波大学社会工学系助教授）に心から御礼申し上げます。

さらに、半導体生産設備に関する有益な情報を提供頂きました（株）日立製作所生産技術研究所下社貞夫氏、同武蔵工場久保内講一氏、同半導体設計開発センタ伊藤勝彦氏に心から御礼申し上げます。

また、本論文の作成にあたり御助力頂きました大阪大学基礎工学部システム工学科古川久生先生、大阪大学工学部精密工学科田村研究室ならびに大阪大学基礎工学部システム工学科田村研究室の諸兄に深く感謝致します。

参考文献

第1章

- [1] 人見勝人: 生産システム工学 第2版, 共立出版, 1990.
- [2] 伊東誼, 岩田一明: フレキシブル生産システム, 日刊工業新聞社, 1984.
- [3] 関根智明: スケジュールの理論, 日刊工業新聞社, 1971.
- [4] 鍋島一郎: スケジューリング理論, 森北出版, 1974.
- [5] 茨木俊秀: 組合せ最適化, 産業図書, 1983.
- [6] 白井良明, 辻井潤一: 人工知能, 岩波書店, 1982.
- [7] L. Marshall and G. Alexander: The design analysis and implementation of heuristics, *Management Science*, Vol. 34, No. 3, pp. 263–265, 1988.
- [8] J. Pearl: *Heuristics*, Addison-Wesley, Massachusetts, 1984.
- [9] 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.
- [10] P. Ow and S. Simith: Filtered beam search in scheduling, *International Journal of Production Research*, Vol. 26, No. 1, pp. 35–61, 1988.
- [11] 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.
- [12] A. Agnetis, C. Aribib, M. Lucertini, and F. Nicolo: Part routing in flexible assembly systems, *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 6, pp. 697–705, 1990.
- [13] P. Kumar and T. Seidman: Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems, *IEEE Transactions on Automatic Control*, Vol. 35, No. 3, pp. 289–298, 1990.
- [14] M. Garetti, A. Pozzetti, and A. Bareggi: On-line loading and dispatching in flexible manufacturing systems, *International Journal of Production Research*, Vol. 28, No. 7, pp. 1271–1292, 1990.
- [15] N. Ishii and J. Talavage: A transient-based real-time scheduling algorithm in FMS, *International Journal of Production Research*, Vol. 29, No. 12, pp. 2501–2520, 1991.

第2章

- [1] 伊東誼, 岩田一明: フレキシブル生産システム, 日刊工業新聞社, 1984.
- [2] 岩田: FMS の設計思想, セミナー“FMS とそのソフトウェア”テキスト, 日本自動制御協会, pp. 1-23, 1984.
- [3] 石井: スケジューリングの近似解法, オペレーションズリサーチ, Vol. 31, No. 1, pp. 26-35, 1984.
- [4] 渡辺: FMS のオンラインスケジューリングとその適応制御, セミナー“FMS とそのソフトウェア”テキスト, 日本自動制御協会, pp. 58-71, 1984.
- [5] J. Blackstone, D. Phillips, and G. Hogg: A state-of-the-art survey of dispatching rules for manufacturing job shop operations, *International Journal of Production Research*, Vol. 20, No. 1, pp. 27-45, 1982.
- [6] 中村, 鳩野, 田村: ルールベースに基づいたフレキシブル生産システムのスケジューリング, 計測自動制御学会論文集, Vol. 23, No. 1, pp. 66-71, 1987.
- [7] Y. Nakamura, I. Hatono, Y. Kohara, and H. Tamura: FMS scheduling using timed Petri net and rule base, *Proceedings of USA-Japan Symposium on Flexible Automation*, Minneapolis, pp. 883-890, 1988.
- [8] H. Tamura, K. Yamagata, and I. Hatono: Decision making for flexible manufacturing – OR and/or AI approach in scheduling, *Journal of Mathematical Modeling and Simulation in Systems Analysis*, Vol. 6, No. 5, pp. 363-371, 1989.
- [9] 中村, 小原, 山縣, 田村: ジャストインタイムを目的としたルール表現による FMS スケジューリング, 計測自動制御学会論文集, Vol. 25, No. 2, pp. 104-110, 1989.
- [10] 田村, 山縣, 鳩野: フレキシブル生産スケジューリングのインテリジェント化, オフィス・オートメーション, Vol. 11, No. 1, pp. 45-52, 1990.
- [11] 市川: 事象駆動システムの表現と制御, 計測と制御, Vol. 24, No. 7, pp. 623-632, 1985.
- [12] 児玉, 熊谷: 離散事象システム — ネットモデルによるアプローチ, 計測と制御, Vol. 24, No. 7, pp. 41-50, 1985.
- [13] J. Peterson: *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, Englewood Cliffs, N.J., 1981. (市川, 小林訳: ペトリネット入門, 共立出版, 1984)
- [14] 鳩野, 橋, 山縣, 田村: ルール修正機能を持ったフレキシブル生産システムのルールベース・スケジューリング, 第34回システム制御情報学会研究発表会講演論文集, 京都, pp. 493-494, 1990.

- [15] 鳩野, 山縣, 田村: ルールベースを用いたフレキシブル生産システムのオンライン・スケジューリング, システム制御情報学会学会論文集, Vol. 4, No. 9, pp. 375–381, 1991.

第3章

- [1] 伊東諄, 岩田一明: フレキシブル生産システム, 日刊工業新聞社, 1984.
- [2] J. Blackstone, D. Phillips, and G. Hogg: A state-of-the-art survey of dispatching rules for manufacturing job shop operations, *International Journal of Production Research*, Vol. 20, No. 1, pp. 27–45, 1982.
- [3] 室津, 岩田, 大場, 的場: 代替機械を考慮した大規模ジョブショップスケジューリング問題の一解法, 日本機械学会論文集 (C 編), Vol. 47, No. 418, pp. 803–811, 1981.
- [4] 中村, 鳩野, 田村: ルールベースに基づいたフレキシブル生産システムのスケジューリング, 計測自動制御学会論文集, Vol. 23, No. 1, pp. 66–71, 1987.
- [5] 中村, 小原, 山縣, 田村: ジャストインタイムを目的としたルール表現による FMS スケジューリング, 計測自動制御学会論文集, Vol. 25, No. 2, pp. 104–110, 1989.
- [6] L. Zadeh: A theory of approximate reasoning, *Machine Intelligence*, Vol. 9, pp. 149–194, 1979.
- [7] 鳩野, 鈴鹿, 山縣, 田村: ファジィ推論を用いたフレキシブル生産スケジューリング—多様なスケジューリング目的の実現—, 日本ファジィ学会論文誌, Vol. 3, No. 2, pp. 339–346, 1991.
- [8] I. Hatono, T. Suzuka, M. Umano, and H. Tamura: Towards intelligent scheduling for flexible manufacturing – Application of fuzzy inference to realizing high variety of objectives –, *Proc. Japan-U.S.A. Symp. on Flexible Automation*, San Francisco, pp. 433–441, July 13–15, 1992.
- [9] G. Bel, D. Dubois, H. Farreny, and H. Prade: Towards the use of fuzzy rule-based, In J. Crestin and J. Mcwaters eds., *Software for Discrete Manufacturing*, pp. 525–535, 1986.
- [10] E. Bensana, G. Bel, and D. Dubois: OPAL: A multi-knowledge-based system for industrial job-shop scheduling, *International Journal of Production Research*, Vol. 26, No. 5, pp. 795–819, 1988.
- [11] M. Fox: *Constraint-Directed Search : A Case Study of Job-Shop Scheduling*, PhD thesis, Carnegie-Mellon University, 1983.
- [12] I. Turksen and D. Ulgury: An approximate reasoning based hierarchical scheduling, *Proceedings of the 3rd International Fuzzy Systems Association Congress*, Seattle, pp. 196–199, 1989.

- [13] H. Zimmermann: The use of fuzzy LP and approximate reasoning in production scheduling, *Proceedings of the 3rd International Fuzzy Systems Association Congress*, Seattle, pp. 193–195, 1989.
- [14] R. Yager: Quantified propositions in a linguistic logic, *International Journal of Man-Machine Studies*, Vol. 19, pp. 195–227, 1983.
- [15] R. Yager: Aggregating evidence using quantified statements, *Information Sciences*, Vol. 36, pp. 179–206, 1985.
- [16] R. Yager: Quantifiers in the formulation of multiple objective decision functions, *Information Sciences*, Vol. 31, pp. 107–139, 1985.
- [17] L. Zadeh: The role of fuzzy logic in the management of uncertainty in expert systems, *Fuzzy Sets and Systems*, Vol. 11, pp. 199–227, 1983.
- [18] L. Zadeh: A computational approach to fuzzy quantifiers in natural language, *Computer and Mathematics with Applications*, Vol. 9, pp. 149–184, 1983.
- [19] R. Yager: An approach to inference in approximate reasoning, *International Journal of Man-Machine Studies*, Vol. 13, pp. 323–338, 1980.
- [20] Y. Tsukamoto: An approach to fuzzy reasoning method, In M. Gupta, R. Tagade, and R. Yager eds., *Advance in Fuzzy Set Theory and Applications*, North-Holland, Amsterdam, pp. 137–149, 1979.
- [21] M. Zemankova-Leech and A. Kandel: *Fuzzy Relational Data Bases – A Key to Expert Systems*, TÜV Rheinland, Köln, West Germany, 1984. (向殿政男訳: ファジィ・リレーショナル・データベース – エキスパート・システムへの鍵, 啓学出版, 1987)
- [22] R. Yager: General multiple-objective decision functions and linguistically quantified statements, *International Journal of Man-Machine Studies*, Vol. 21, pp. 389–400, 1984.
- [23] L. Zadeh: PRUF – a meaning representation language for natural languages, *International Journal of Man-Machine Studies*, Vol. 10, pp. 149–184, 1978.
- [24] A. Pritsker: *Introduction to Simulation and SLAM II*, John Wiley and Sons, New York, 1986.

第4章

- [1] 白井良明, 辻井潤一: 人工知能, 岩波書店, 1982.
- [2] L. Marshall and G. Alexander: The design analysis and implementation of heuristics, *Management Science*, Vol. 34, No. 3, pp. 263–265, 1988.

- [3] J. Pearl: *Heuristics*, Addison-Wesley, Massachusetts, 1984.
- [4] P. Ow and S. Simith: Filtered beam search in scheduling, *International Journal of Production Research*, Vol. 26, No. 1, pp. 35–61, 1988.
- [5] 安西祐一郎: 認識と学習, 岩波書店, 1989.
- [6] 大須賀節夫, 佐伯胖: 知識の獲得と学習, オーム社, 1987.
- [7] 國藤, 古川: 学習システム研究の現状と課題, 計測と制御, Vol. 25, No. 9, 1986.
- [8] D. De and A. Lee: Flexible manufacturing system (FMS) scheduling using filtered beam search, *Journal of Intelligent Manufacturing*, Vol. 1, No. 3, pp. 165–183, 1990.
- [9] I. Hatono, M. Tatewaki, M. Umamo, and H. Tamura: Automatic generation of heuristics for filtered beam search and its application to scheduling of automated guided vehicles, *Proc. 1992 Pacific Conf. on Manufacturing*, Sakai, pp. 706–713, 1992.

第5章

- [1] 伊東誼, 岩田一明: フレキシブル生産システム, 日刊工業新聞社, 1984.
- [2] K. Baker: *Introduction to Sequencing and Scheduling*, John Wiley and Sons, New York, 1974.
- [3] M. Fox and S. Simith: ISIS – a knowledge-based system for factory scheduling, *Expert Systems*, Vol. 1, No. 1, pp. 25–49, 1984.
- [4] J. Kanet: Expert system in production scheduling, *European Journal of Operational Research*, Vol. 29, No. 1, pp. 51–59, 1984.
- [5] I. Hatono, K. Yamagata, and H. Tamura: Modeling and on-line scheduling of flexible manufacturing systems using stochastic Petri nets, *IEEE Transactions on Software Engineering*, Vol. 17, No. 2, pp. 126–131, 1992.
- [6] 鳩野, 山縣, 田村: ルールベースを用いたフレキシブル生産システムのオンライン・スケジューリング, システム制御情報学会学会論文集, Vol. 4, No. 9, pp. 375–381, 1991.
- [7] I. Hatono, M. Umamo, and H. Tamura: Knowledge-based on-line scheduling for flexible manufacturing, *Proceedings of the IMACS/SICE International Symposium on Robotics, Mechatronics and Manufacturing Systems*, Kobe, pp. 913–918, 1992.
- [8] M. Marsan, G. Conte, and G. Balbo: A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems, *ACM Computing Surveys*, Vol. 2, No. 2, pp. 93–122, 1984.

- [9] F. Archetti and A. Sciomachen: Representation, analysis and simulation of manufacturing systems by Petri net based models, *Proceedings of IIASA Conference on Discrete Event Systems*, Sopron, Hungary, pp. 162–178, 1987.
- [10] 中村, 鳩野, 田村: ルールベースに基づいたフレキシブル生産システムのスケジューリング, 計測自動制御学会論文集, Vol. 23, No. 1, pp. 66–71, 1987.
- [11] J. Blackstone, D. Phillips, and G. Hogg: A state-of-the-art survey of dispatching rules for manufacturing job shop operations, *International Journal of Production Research*, Vol. 20, No. 1, pp. 27–45, 1982.

第6章

- [1] 人見勝人: 生産システム工学 第2版, 共立出版, 1990.
- [2] 伊東誼, 岩田一明: フレキシブル生産システム, 日刊工業新聞社, 1984.
- [3] 鍋島一郎: スケジューリング理論, 森北出版, 1974.
- [4] 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.
- [5] 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.
- [6] S. Jain, K. Barber, and D. Osterfeld: Expert simulation for on-line scheduling, *Communications of the ACM*, Vol. 33, No. 10, pp. 52–60, 1990.
- [7] I. Hatono, K. Yamagata, and H. Tamura: Modeling and on-line scheduling of flexible manufacturing systems using stochastic Petri nets, *IEEE Transactions on Software Engineering*, Vol. 17, No. 2, pp. 126–131, 1992.
- [8] M. Huhns ed.: *Distributed Artificial Intelligence*, Pitman, London, 1987.
- [9] Y. Demazeau: Decentralized artificial intelligence, In Y. Demazeau, et al. eds., *Decentralized A.I.*, North-Holland, Amsterdam, pp. 3–13, 1990.
- [10] 伊藤: 制御理論から見た自律・分散・協調, 計測と制御, Vol. 26, No. 1, pp. 26–32, 1987.
- [11] 伊藤: 自律分散システムはいかにして構成されるか, 計測と制御, Vol. 29, No. 10, pp. 877–881, 1990.
- [12] H. Parunak: Distributed AI and manufacturing control: Some issues and insights, In Y. Demazeau, et al. eds., *Decentralized A.I.*, North-Holland, Amsterdam, pp. 81–101, 1990.

-
- [13] H. Parunak: Manufacturing experience with the contract net, In M. Huhns ed., *Distributed Artificial Intelligence*, Pitman, London, pp. 923–928, 1987.
- [14] 鳩野, 橋, 馬野, 田村: フレキシブル生産を対象とした分散型リアルタイムスケジューリング, 第1回生産スケジューリングシンポジウム講演論文集, 名古屋, pp. 135–140, 1993.
- [15] A. Pritsker: *Introduction to Simulation and SLAM II*, John Wiley & Sons, New York, 1986.
- [16] 森戸晋, 相沢りえ子: SLAM II によるシステム・シミュレーション, 構造計画研究所, 1986.
- [17] 柳井久義, 永田穰: 集積回路工学 (1), コロナ社, 1983.

図表リスト

図目次

第2章

| | | |
|----------|---|----|
| Fig. 2.1 | Examples of FMS model. | 9 |
| Fig. 2.2 | Examples of deadlocks. | 10 |
| Fig. 2.3 | Hierarchical structure of the rule base. | 13 |
| Fig. 2.4 | A rule-based scheduling system. | 14 |
| Fig. 2.5 | A rule base for minimizing the completion time and the number of set-ups. | 17 |
| Fig. 2.6 | A transfer line. | 18 |
| Fig. 2.7 | Examples of schedules. | 19 |
| Fig. 2.8 | An example of rule base for JIT. | 21 |

第3章

| | | |
|-----------|---|----|
| Fig. 3.1 | An overview of flexible scheduling | 29 |
| Fig. 3.2 | Example of alternative operations of job i | 31 |
| Fig. 3.3 | An example of inference tree in a schedule evaluation module. | 33 |
| Fig. 3.4 | A fuzzy set representing quantifier “ <i>about 3.</i> ” | 35 |
| Fig. 3.5 | An example of a membership function representing <i>most.</i> | 37 |
| Fig. 3.6 | A numerical example of degree of influence | 40 |
| Fig. 3.7 | Knowledges in dispatching module | 41 |
| Fig. 3.8 | System configuration. | 43 |
| Fig. 3.9 | Examples of membership functions. | 45 |
| Fig. 3.10 | Example of schedules based on Strategy 1. | 46 |
| Fig. 3.11 | Example of schedules based on Strategy 2. | 46 |

第 4 章

Fig. 4.1 A transport line. 50

Fig. 4.2 The distribution of requirements. 50

Fig. 4.3 An example of search tree in filtered beam search method 54

Fig. 4.4 System configuration. 55

Fig. 4.5 An example of data classification matrix of task. 56

Fig. 4.6 Scheduling requirement 1. 59

Fig. 4.7 Scheduling requirement 2. 59

Fig. 4.8 Scheduling requirement 3. 60

Fig. 4.9 Scheduling requirement 4. 60

Fig. 4.10 Scheduling requirement 5. 61

Fig. 4.11 Distribution of the values of the objective function for scheduling require-
ment 1. 63

Fig. 4.12 Distribution of the values of the objective function for scheduling require-
ment 2. 63

Fig. 4.13 Distribution of the values of the objective function for scheduling require-
ment 3. 64

Fig. 4.14 Distribution of the values of the objective function for scheduling require-
ment 4. 64

Fig. 4.15 Distribution of the values of the objective function for scheduling require-
ment 5. 65

第 5 章

Fig. 5.1 An example of partitioned stochastic Petri net model 70

Fig. 5.2 Transporting level models of four basic FMS elements. 72

Fig. 5.3 Completion time. 76

Fig. 5.4 Standard deviations of completion times. 76

第6章

| | | |
|-----------|--|-----|
| Fig. 6.1 | Flow chart of processing in a factory. | 79 |
| Fig. 6.2 | An example of a model of a factory. | 79 |
| Fig. 6.3 | Conceptual configuration of distributed real-time scheduling system. | 81 |
| Fig. 6.4 | Overview of communications between scheduling agents. | 83 |
| Fig. 6.5 | Information flows for routing | 84 |
| Fig. 6.6 | Flow of information for selecting a conveying job. | 86 |
| Fig. 6.7 | Flow of information for selecting a job from input buffer. | 87 |
| Fig. 6.8 | System configuration. | 88 |
| Fig. 6.9 | Structure of evaluation for resolving conflicts in routing. | 91 |
| Fig. 6.10 | Structure of evaluation for resolving conflicts in selecting a conveying job. | 92 |
| Fig. 6.11 | Structure of evaluation for resolving conflicts in selecting a job from an input buffer. | 93 |
| Fig. 6.12 | System configuration of production equipments. | 96 |
| Fig. 6.13 | An example of model of semiconductor process. | 98 |
| Fig. 6.14 | Comparison of schedules depending upon the existence of an express job. | 100 |
| Fig. 6.15 | Comparison of schedules depending upon whether the number of set-ups is taken into consideration or not. | 102 |
| Fig. 6.16 | Comparison of completion time and average waiting time depending upon whether the waiting time is taken into account or not. | 104 |

表目次

第2章

| | | |
|-----------|------------------------------------|----|
| Table 2.1 | Processing information. | 24 |
| Table 2.2 | Comparison of schedulings. | 24 |

第3章

| | | |
|-----------|--------------------------------------|----|
| Table 3.1 | Processing informations. | 44 |
| Table 3.2 | Transport time of jobs. | 44 |
| Table 3.3 | Statistics of the schedules. | 47 |

第 4 章

| | | |
|-----------|---|----|
| Table 4.1 | An example of degree of influence. | 57 |
| Table 4.2 | Comparison of the performance of schedulings. | 61 |
| Table 4.3 | The coefficients of evaluation items. | 62 |

第 5 章

| | | |
|-----------|---|----|
| Table 5.1 | Dispatching rules. | 74 |
| Table 5.2 | Conditions of numerical experiments | 74 |

第 6 章

| | | |
|-----------|---|-----|
| Table 6.1 | Processing information of each job | 99 |
| Table 6.2 | Statistical comparisons on numerical example 1. | 101 |
| Table 6.3 | Statistical comparison on numerical example 2. | 102 |
| Table 6.4 | Statistical comparisons on numerical example 3. | 105 |

関連発表論文

- [1] 中村, 鳩野, 田村: ルールベースに基づいたフレキシブル生産システムのスケジューリング, 計測自動制御学会論文集, Vol. 23, No. 1, pp. 66–71, 1987.
- [2] Y. Nakamura, I. Hatono, Y. Kohara, K. Yamagata, and H. Tamura: FMS scheduling using timed Petri net and rule base, *Proc. U.S.A.-Japan Symposium on Flexible Automation*, Minneapolis, pp. 883–890, 1988.
- [3] I. Hatono, K. Yamagata, and H. Tamura: Modeling and on-line scheduling of flexible manufacturing systems using stochastic Petri nets, *IEEE Trans. on Software Engineering*, Vol. 17, No. 2, pp. 126–132, 1991.
- [4] 鳩野, 鈴鹿, 山縣, 田村: ファジィ推論を用いたフレキシブル生産スケジューリング—多様なスケジューリング目的の実現—, 日本ファジィ学会論文誌, Vol. 3, No. 2, pp. 339–346, 1991.
- [5] 鳩野, 山縣, 田村: ルールベースを用いたフレキシブル生産システムのオンライン・スケジューリング, システム制御情報学会論文集, Vol. 4, No. 9, pp. 375–381, 1991.
- [6] I. Hatono, T. Suzuka, M. Umamo, and H. Tamura: Towards intelligent scheduling for flexible manufacturing – Application of fuzzy inference to realizing high variety of objectives –, *Proc. Japan-U.S.A. Symp. on Flexible Automation*, San Francisco, pp. 433–441, July 13–15, 1992.
- [7] I. Hatono, M. Umamo, and H. Tamura: Knowledge-based on-line scheduling for flexible manufacturing, *Proceedings of the IMACS/SICE International Symposium on Robotics, Mechatronics and Manufacturing Systems*, Kobe, pp. 913–918, September 16–20, 1992.
- [8] I. Hatono, M. Tatewaki, M. Umamo, and H. Tamura: Automatic generation of heuristics for filtered beam search and its application to scheduling of automated guided vehicles, *Proc. 1992 Pacific Conf. on Manufacturing*, Sakai, pp. 706–713, November 3–6, 1992.

その他論文

- [1] 鳩野, 上田, 阪田: グループ共同作業支援のためのマルチメディア在席対話システム, 情報処理学会論文誌, Vol. 30, No. 4, pp. 527–535, 1989.
- [2] 田村, 松井, 鳩野: 不確実下の価値関数による公共的リスク評価の方法論, 日本リスク研究学会誌, Vol. 2, No. 1, pp. 89–96, 1990.

- [3] 田村, 寺岡, 鳩野, 山縣: ニューラルネットワークによる巡回セールスマン問題の一解法, システム制御情報学会論文誌, Vol. 4, No. 1, pp. 57-59, 1991.
- [4] 田村, 小林, 鳩野: Dempster-Shafer の確率理論を導入した不確実下の価値関数, システム制御情報学会論文誌, Vol. 4, No. 4, pp. 133-139, 1991.
- [5] 田村, 九津見, 鳩野: 接続水系における実時間予測を併用したファジィ制御方式, システム制御情報学会論文誌, Vol. 4, No. 10, pp. 421-428, 1991.
- [6] H. Tamura, K. Yamamoto, I. Hatono, and M. Umamo: Multiattribute evaluation of flexible manufacturing systems using AHP combined with neural networks, *Proc. Japan-U.S.A. Symp. on Flexible Automation*, San Francisco, pp. 1751-1754, July 13-16, 1992.
- [7] M. Umamo, T. Teraoka, I. Hatono, and H. Tamura: On some attempts for improving learning performance in multi-layered neural networks of back-propagation methods, *Proceedings of the IMACS/SICE International Symposium on Robotics, Mechatronics and Manufacturing Systems*, Kobe, pp. 173-178, September 16-20, 1992.
- [8] 田村, 平原, 鳩野, 馬野: 組合せ最適化問題の近似解法 - 遺伝的アルゴリズムとラグランジュ緩和法を併用したハイブリッド法 -, 計測自動制御学会論文集, Vol. 30, No. 3, pp. 329-336, 1994.

解説

- [1] 鳩野, 田村: 知識処理を用いたフレキシブル生産オンラインスケジューリング, オペレーションズリサーチ, Vol. 37, No. 2, pp. 70-75, 1992.
- [2] I. Hatono and H. Tamura: Knowledge-based on-line scheduling for flexible manufacturing, In M. Jamshidi ed., *Artificial Intelligence: Theory and Applications*, Prentice-Hall. (印刷中)