

Title	知識工学を導入した物流システム運用制御技術に関する研究
Author(s)	都島, 功
Citation	大阪大学, 1989, 博士論文
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/210">https://hdl.handle.net/11094/210</a>
rights	
Note	

*Osaka University Knowledge Archive : OUKA*

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

Osaka University

知識工学を導入した  
物流システム運用制御技術に関する研究

都 島 功

# 知識工学を導入した 物流システム運用制御技術に関する研究

## 目 次

1. 結 論 .....	1
1.1 研究の動機と目的 .....	1
1.2 従来の研究との関連 .....	4
1.3 研究の概要 .....	7
参考文献 .....	9
2. 問題と解法の記述に知識工学を応用した流れ作業ライン物流制御方式 .....	12
2.1 まえがき .....	12
2.2 制御対象のモデル .....	13
2.3 ルール型制御システム S C D (Station Coordinator) .....	14
2.4 ルールのチェック方法 .....	19
2.5 ビレット精整ラインのルール型制御 .....	20
2.6 ルール型制御方式の評価 .....	25
2.7 まとめ .....	28
参考文献 .....	30
3. 問題記述に知識工学を応用した搬送車群運行制御方式 .....	32
3.1 まえがき .....	32
3.2 運行制御ソフトへの要求 .....	32
3.3 運行制御機能 .....	34
3.4 搬送要求への搬送車割付方式 .....	35
3.5 運行制御システム .....	39
3.6 搬送車群運行制御シミュレータ .....	42
3.7 適用例 .....	45
3.8 まとめ .....	51
参考文献 .....	53
4. 問題記述に知識工学を応用したピッキング効率向上運用方式 .....	54
4.1 まえがき .....	54
4.2 二段階倉庫システムの定式化 .....	55
4.3 二段階倉庫の運用アルゴリズム .....	57
4.4 評価 .....	60
4.5 まとめ .....	65
参考文献 .....	66

5. 解法戦略ノウハウの記述にも知識工学を応用した作業者の作業計画方式	67
5.1 まえがき	67
5.2 資源配分問題の定義	68
5.3 前倒し, 削除機能の実現方式	69
5.4 負荷山崩し結果の評価	72
5.5 実システムへの応用例	75
5.6 まとめ	83
参考文献	84
6. 結 論	85
謝 辞	87

# 第1章 緒論

# 第 1 章 結 論

## 1.1 研究の動機と目的

本論文では、物流システムにおける物の流れの制御、作業計画業務を対象に、物、仕事に対してどの資源（設備、作業員）を割付けるかを決定する運用制御方式について論ずる。

物流システムの効率化がいま多くの企業で進められている。その背景として、物流活動に伴うコストの全コストに占める割合が次第に上昇し、企業利益に重大な影響を及ぼしつつあること、企業間競争の激化に伴って物流活動が顧客サービス向上の重要な手段であること、が認識されてきたことが挙げられる。

近年、顧客要求の多様化に伴い、多品種少量生産化が急速に進行し、物の流れがますます複雑になって来ている。そのため、このような状況下で物流システムにおける運用制御が効率的に実施できることが要求されている。さらに、製品のモデルチェンジによる生産方式、生産設備の変更に伴い、物流システムの運用制御方式に頻繁な変更が発生する。そのため、運用制御方式が容易に変更できることも、効率向上と同様に要求されている。

効率的な運用制御を実現するために、運用制御問題を数理計画問題に定式化し、解を求めるというアプローチが考えられる。しかしながら、実際の運用制御問題では、目的関数、制約条件を数学的に厳密に定義することは、通常、容易ではない。たとえ初期設計段階で定式化できたとしても、生産方式等の変化に伴う運用制御方式の変更がパラメータの値の変更で済むことは稀で、問題の再定式化が必要となり、変更柔軟に対応できない。そのため、通常は、運用制御担当者がノウハウとして持つ、仕事を資源に割り付けるための割付優先規則に基づきヒューリスティックなアルゴリズムを作成し、解を求めるというアプローチが採られている。

アルゴリズムは手続き型の汎用プログラミング言語により運用制御プログラムとして実現される。この場合、ノウハウとそれらの利用順序を規定する手続き処理がプログラム中に混在してしまう。そのため、変更柔軟に対応できない。また、プログラムの作成方法においても、一般にプログラミングの非専門家である運用制御担当者と運用制御の非専門家であるプログラマーが一意的に解釈困難な仕様書を介して協同でプログラムを作成することになるので、仕様書の内容とプログラムの内容に不一致が生じ、変更柔軟に対応できない。

それらの問題点を解決するために、運用制御担当者のノウハウを容易に計算機化でき、且つ、ノウハウの変更に対して柔軟に対応できる知識工学に着目した。

知識工学は、プロダクションシステムに代表されるように、対象の論理を“IF(条件), THEN(結論)”型のルールとして表現し、その論理の利用順序はシステムが用意する推論機構が自動的に決定するもの(非手続き型)である。これにより、論理を利用順序と独立に記述でき、日本語的な表現もそのまま扱うことができるので、ノウハウの計算機化が容易となり、且つ、ノウハウの変更に対する柔軟性が期待できる。

しかしながら、知識工学には処理時間上の問題がある。同じ内容の処理を従来のプログラム方式で行う場合に比べ、通常、10倍以上の処理時間がかかる。しかも、問題によっ

ては試行錯誤的なバックトラックが生じ、実用的な時間で処理できなくなる可能性が大きい。また、知識工学は各種条件に合う仕事や資源あるいは割付優先規則を選択するという条件判定的な問題に適したものであり、解を合成しながら最適なものを見つけていく組合せ問題を含むことになる運用制御問題に応用するには問題が多い。そのため、実用的な時間での求解を実現するには、組合せ計算に向けた数理的なアルゴリズムと組合せる必要がある。ただし、組合せの数の少ない、小規模な問題に対しては、知識工学だけで対応しても処理時間に関し問題が発生しない場合が多い。

本論文では、上述した知識工学の運用制御問題への適用上の問題点を解決するため、物流システムの運用制御問題を変更の多い条件判定的な部分と変更の少ない組合せ計算アルゴリズム部分に分け、知識工学アプローチと数理的アプローチの良さを組合せて利用する方式を提案する。

運用制御問題には多様なタイプがあり、それらを単一の知識工学応用運用制御方式で対応することは難しい。しかし、運用制御問題を、主に知識工学アプローチに適した条件判定的な問題と主に数理的アプローチに適した組合せ的な問題とからなる構造で共通的に捉えることができる。この条件判定的な問題において、問題の記述あるいはどの場合にどの組合せ計算の解法を使うべきかというノウハウ(解法戦略ノウハウと呼ぶ)の記述に変更の柔軟性が求められること、処理時間が組合せの数に依存することから、知識工学の効果的な応用形態は以下の3つに大きく分けられる。

(1) 問題と解法の両者の記述に知識工学を応用する形態

これは、組合せの数の少ない、小規模な問題に適した形態である。

(2) 問題の記述に知識工学を応用し、組合せ計算はアルゴリズムで対応する応用形態

これは、制約条件、評価項目に変更が頻発し、基本的に組合せ計算が一種類の解法で対応でき、組合せの数の多い問題に適した形態である。

(3) 問題の記述だけでなく、解法戦略ノウハウの記述にも知識工学を応用し、各解法戦略に基づく組合せ計算はアルゴリズムで対応する応用形態

これは、複数種類の解法を必要とする複雑な組合せ問題に適した形態である。

以下では、それぞれの応用形態に対し、解決を強く要望されている事例を取り上げ、上述の構造の実現手法について述べる。

第1の、問題と解法の両者の記述に知識工学を応用する形態に適した問題として、流れ作業ライン型の工程での物流制御問題を取り上げる<sup>1)</sup>。この制御問題とは、処理設備の稼働率の向上を目的に、分岐点に到着した物(仕事)を複数個ある処理設備のどれに搬送すべきかを決定することである。搬送すべき処理設備先は処理設備の稼働状況、搬送ラインの物流状況、物により異なる処理内容などの組合せで決まる。ただし、流れ作業ライン型の工程では、物が分岐、合流を繰り返して流れるが、分岐点から各処理装置への搬送ラインは通常一本であるため、運用制御論理はあまり複雑にはならず、大部分をIF-THENルールで記述できる。ここでの課題は知識工学を実時間制御に適用する点にある。実時間制御への適用では、ルール記述されたプログラムが人間の介在が許されない閉ループコントロールの中で動くため、誤った制御信号を出力しないように実稼働前に十分にルールの合理性を

エックが行える方式が必要である。また、実時間制御においては、種々のセンサ・アクチュエータがインターフェースとして使用されるため、それらとの簡易なプロセスインターフェース方式も必要である。そこで、製鉄所のビレットミル精整ラインの物流制御へ知識工学を実適用した事例をもとに、知識工学応用上の実用的な方式、手順、実用面からの解析結果について述べる。なお、これは、実プロセスの実時間制御への我が国初の知識工学応用事例である。

第2の、問題の記述に知識工学を応用し、組合せ計算はアルゴリズムで対応する応用形態に適した問題として、2つの問題を取り上げる。1つは、流れ作業ライン型の工程に相対するジョブショップ型の工程での搬送車群の運行制御問題である。搬送車は各作業ステーションへアクセスできるという自由度の高い搬送手段であり、各作業ステーションからの搬送要求にどの搬送車を割り付けるかという組合せ方により効率が大きく左右される。また、作業ステーションへ物を搬送できるか否かの条件(搬送可否条件)、どの搬送要求から優先的に搬送すべきかの条件(搬送優先条件)は制御対象毎に異なり、これらの条件は頻繁に変更される可能性が高い。そこで、搬送可否条件に基づき多数の搬送要求から搬送可能なものを選択し、選択された搬送要求に対し搬送優先条件に基づき搬送優先度を付けるという条件判定的な問題と、条件の変更に影響されない、優先度付けされた搬送要求への搬送車割り付けのための組合せ問題に分離する方式を提案する。また、搬送車の割り付けについては高効率な汎用的アルゴリズムを提案する。

他の一つの問題は、物流システムの重要な構成要素である自動倉庫において最大の課題であるピッキング効率の向上に関する運用制御問題を取り上げる。ピッキング作業とは、注文に応じてパレットから必要個数の製品を取り出す作業であり、人間に頼らざるを得ず、多くの人手がかかっている。これに対して、注文の処理順序の並び換えによりピッキング効率を向上させる運用制御方式を提案する。全体の注文をある基準、例えば納期上、処理順序を並び換えてよい注文群に分割するという条件判定的な問題の記述にルールを応用した。一方、ピッキング作業者が物の到着待ちにより無作業とならないように、分割された各注文群に対して注文の処理順序を決定する問題はアルゴリズムで対応した。

第3の、解法戦略ノウハウの記述にも知識工学を応用し、各解法戦略に基づく組合せ計算はアルゴリズムで対応する応用形態に適した問題として、作業員に対する作業計画問題を取り上げる。作業計画問題は、確保できる人員内に仕事の負荷を納めるように、各仕事の実施時刻を決定する問題である。この問題に対して実用的な解を求めるには、負荷平滑化のために、各仕事の実施時刻を遅らせたり(仕事の後倒し)、各仕事の実施時刻を早めたり(仕事の前倒し)、場合によっては仕事を削除する必要がある。しかも、それらの負荷平滑化手段を状況に応じて使い分ける必要がある。そこで、どの仕事にどの負荷平滑化手段をどの様な場合に適用すべきかといった負荷平滑化戦略をルールで記述し、ルール記述された戦略に基づく負荷平滑化計算は仕事の後倒し、仕事の前倒し、仕事の削除の3つのアルゴリズムで対応する方式を提案する。なお、アルゴリズムの部分は、仕事の後倒しによる負荷平滑化手段しか持たないPERT/LOADを拡張したものである。この種の作業計画問題は物流に限らず、流通など広い分野で発生する。提案方式をスーパーマーケットの実際の



作業計画業務に適用したので、その内容についても述べる。

## 1.2 従来の研究との関連

物流システムの運用制御問題は教科書的には数理計画問題として解かれるが、実際の問題は固有の条件が多く、汎用的な取り扱いが難しい。ここでは、本論文で取り上げる事例に係るコンベア、無人搬送車、自動倉庫、作業者を対象にした運用制御問題の研究について、この順に研究状況を説明する。

### (1) コンベアを中心とした流れ作業ライン運用制御問題の研究

どの物から順に流すかの最適解を求めるための方法に有名なJohnson氏の解析的な方法がある。これは2台の機械、2工程で、仕事の機械にかかる順序が同一である場合、すなわち流れ作業ラインの場合を扱っている。Lomnicki氏は、機械3台の場合の問題に対し、分岐限定法を使用した解法を提案している。いずれの方法も最適解を求められるが、シンプルな対象しか扱えず、実際の問題に適用できない。

一方、実際の問題に対しては、家長氏が製鋼工場のオンライン物流管制システムを発表している。ここでは作業間の競合などの制約条件を緩めながら、逐次、順序を決定していくヒューリスティックなアルゴリズムを提案している。このように、製鉄所では物流の合理化が盛んに進められており、福村氏も同様なアプローチをとったシステムを発表している。しかしながら、それらは、従来のプログラミング言語により実現されているため、条件の変更に柔軟に対応できないという問題点を持つ。

知識工学の応用例としては、三浦氏が静電容量式差圧発信機の製造ライン(流れ作業ラインが中心)を対象に、対象をネットワークモデルとルールモデルで表現し、そのモデルを使って物流制御方式決定のためのシミュレーションを行うツールを提案している。ネットワークモデルでは設備の基本動作および物の流れなどの設備条件をモデル化する。ルールでは複数の動作が競合する場合、いずれの動作を優先するかといった運用条件をモデル化している。このツールによりシミュレータが早期に開発できたとのことであるが、シミュレーションレベルであり、実時間制御問題への知識工学応用上の課題であるルールの合理性チェック方式、プロセスとのインターフェース方式等について言及されていない。

これに対して、本論文では、実際の実時間制御へ知識工学を応用した事例をもとに、上述のチェック方式、インターフェース方式に関し、実用的な方式を提案しており、実用面からの各種解析結果についても述べている。なお、この事例は、比較的小規模な組合せ問題であるため、問題と解法の両者に知識工学を応用している。

### (2) 搬送車群運行制御問題の研究

この問題には、最短ルート決定問題、搬送要求に即応するために搬送車をどこに待機させるべきかを決定する待機点選択問題、合流点での搬送車進入制御問題、搬送要求への搬送車割当て問題がある。最短ルート決定問題については、Haines氏が混雑度を考慮して動的にルートを決定していくアルゴリズムを提案している。このアルゴリズムの特徴は作業ステーションの番号付けの工夫により部分的な軌道レイアウト情報のみで適切なルートを決定できることにある。待機点選択問題については、安信氏が、無人搬送車システムでは

ないが、シティーカーシステムを対象に、シミュレータを用いて多地点間の多種類の搬送要求パターンと待機点方式の関係を分析している。合流点での搬送車進入制御問題については、既に制御方式が確立されている状況にあり、篠塚氏が閉塞区間制御方式を含め、既存の方式の長所、短所を示している。

しかしながら、以上の3つの運行制御問題については、FAにおける搬送車導入システムの大部分は軌道レイアウトがあまり複雑でないこと、搬送車数は平均5、6台であること、搬送車の移動範囲はあまり広くないことから、その実現方式の違いによって運行制御の効率が大きく変わるものではない。また、変更の柔軟性に関し、問題となる搬送可否条件、搬送優先条件の変更に影響されない。

一方、搬送車割当て問題については、その実現方式によって運行制御の効率が大きく左右される。また、搬送可否条件、搬送優先条件にも係わる。これに対して、藤井氏が、搬送車シミュレータを用い、FIFO、LIFO、最短距離要求優先、最早工程要求優先、最遅工程要求優先の5つの搬送要求選択方法の良し悪しを搬送車数との関係で分析している。しかしながら、取り上げられた運用制御方式が単純すぎるため、実際の問題には適用できない。

知識工学の応用例としては、飯田氏が、搬送車ではなく鉄道車両に対する割当て計画にPrologを適用している。この計画対象は組合せ問題としてはかなりシンプルであるため、問題記述、解法(ディスパッチングルール)の両者をPrologにより記述している。また、一日に一度の計画立案であり、プロセスとのインターフェースもないため、実時間性の問題を考慮しなくてもよい。したがって、本方式を複雑な組合せ問題となる搬送車割当て問題に適用することはできない。

これに対し、本論文では、実システムレベルの搬送車割当て問題に対して、変更が頻発する条件への柔軟な対応と高效率運行の両立を実現するため、問題の記述に知識工学を応用し、組合せ計算はアルゴリズムで対応する方式を提案している。

### (3) ピッキング作業に係る自動倉庫システムの運用制御方式の研究

Hausman氏は、ラックとクレーンからなる自動倉庫(棚が回転しないので、固定倉庫と呼ぶ)を対象に回転率の高い品目ほど入庫口の近くの空き棚を入庫棚とする方式を含め、各種入庫棚決定運用方式に対し、品目構成をパラメータとしたクレーン移動距離算出モデルを提案し、各方式の有効性を評価している。三留氏は、Hausman氏の静的モデルを拡張し、動的に入庫棚を割当てるスケジューリング方式を提案している。また、クレーンの空走行距離最小化を目的に、入庫後、入庫口に戻らず、出庫を実施させる複合サイクル運転運用方式もあり、Hausman氏はこの方式についてもその有効性を評価している。

これらは、固定倉庫を対象に、既に実施されているクレーン運用の各種方式の有効性を入庫時間の面から原理的に評価したものであり、ピッキング効率を向上させるための新しい方式の提案ではない。また、運用方式はアルゴリズムのみで実現しており、実際の問題に適用できるものではない。

知識工学の自動倉庫への応用例としては、藤井氏が、棚の回転する自動倉庫(回転倉庫と呼ぶ)の運用方式について発表している。ここでは、単位時間当りの出庫量最大化を目的に、どの品目(棚)から出庫すべきかの制御則(ディスパッチングルール)をIF-THENル-

ルで表現している。基本的な各種制御則の検討が目的のため、問題と解法の両者をルール記述しても処理時間の問題は発生しない。しかしながら、実際の倉庫において、それらの単純な制御則および回転倉庫だけではピッキング効率を向上させることが困難であると考ええる。

そこで、本論文では、ピッキング効率を高めるには、固定倉庫、回転倉庫単独での入出庫時間の短縮では対応できないと考え、倉庫形態としては固定倉庫と回転倉庫を結合したシステムとし、出庫順序のスケジューリングアルゴリズムの工夫によりピッキング効率向上を図る方式を提案している<sup>9)</sup>。また、実用化を目的とするため、変更の頻発する条件の記述には知識工学を応用している。

#### (4) 作業計画問題の研究

作業計画問題は時間の概念を導入した組合せ問題、すなわちスケジューリング問題となる。スケジューリングについては、(1)で述べたJohnson氏の解析的な方法<sup>3)</sup>の発表以来、多数の研究がある。その中で、組合せ最適化問題として、整数計画法、分岐限界法で定式化するアプローチがある<sup>28)</sup>が、処理時間の面から実用的な規模の問題を扱うことが困難である。これに対して、ディスパッチングルールをヒューリスティックに定めて、対象別に作成したシミュレータでスケジューリングするアプローチがある<sup>29)</sup>。しかしながら、これらの多くものは状況に応じてディスパッチングルールを変えることができないという問題点を持つ。そこで、中村氏は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によってモデル化し、状況にあわせて適用すべきディスパッチングルールをIF-THENルールで記述する方法を発表している<sup>30)</sup>。しかしながら、時間のかかるペトリネットを利用しているため、現実の問題に適用可能とするには高速化処理方式の開発が必要と考える。

一方、数理工学的な手法で実際の計画問題を扱っても処理時間の面で問題のないものにPERT/LOADがある。大場氏、筆者は、処理設備しか扱えないPERT/LOADを拡張し、ストレージも扱えるアルゴリズムを提案<sup>31)</sup>している。これは物流システム建設時における初期の設備容量計画を対象にしているため、運用方式は単純にしている。そのため、考慮できる負荷平滑化手段は仕事の後倒しのみとしている。

スケジューリング問題への知識工学の応用例について、最近いくつか発表されている。Bourne氏が発表しているISISシステム<sup>32)</sup>はその一つである。これは、計画対象となるジョブショップの資源、作業、それらに関する制約条件などをフレーム形式の知識型言語(SRL)によりモデル化する。解法は、制約条件をどれだけ満足するかを随時評価し、それに基づき探索木を枝刈りしながら探索を行う方式である。作業順序の決定ではCPMが用いられている。解の探索において、CPMは探索木の枝刈りの一手段として使われており、探索自体は知識工学の手段によって行なっているため、実際の計画問題を実用的な時間で解くことが困難であると考ええる。Bruno氏はFMSを対象にルール型生産計画システムを発表している<sup>33)</sup>。このシステムでは、数理工学的な手法として、closed queue network を用いて負荷状況を評価しながらスケジューリングを行っているが、ISISと同じ問題点を持つ。戸沢氏はスケジューリングを行う際の基準や制約等をProlog上のプロダクションシステムで記述する製鋼工程スケジューリングシステムSchepplanを発表している<sup>34)</sup>。これは、ベストなスケジュー

ルを作ることを諦め、一部の制約条件を満たす解を一つ出力し、人間がマンマシンの他の制約条件を満たすように修正する方式を採っている。しかし、製鋼工程特有の属性がスケジューリングエンジンに入っており、適用範囲が狭いと考える。Chandrasekaran氏は知識処理用の個別問題向けシェル Generic Task を発表している。ここでは、知識処理システムの対象とする問題を6つのサブ問題(これらをTaskと呼んでいる)に分類し、各Taskに応じたツール(これをシェルと呼ぶ)を開発することを提案している。これは、モデル化の容易化に着目したもので、求解においては試行錯誤を行い、制約条件を満たす解を探すため、実用的な時間で解を求めることが困難であると考えられる。

本論文では、実際のスケジューリング問題を解くには解法そのものを手続き型言語による汎用的なアルゴリズムで実現し、しかも複雑な組合せ問題に対して適切な解を求めるために複数種類の解法アルゴリズムを導入し、変更への柔軟性の確保のためには、問題の記述だけでなく解法戦略ノウハウの記述にも知識工学を応用する方式を提案している。なお、アルゴリズムはPERT/LOADを拡張したものである。

### 1.3 研究の概要

第2章で問題と解法の両者の記述に知識工学を応用する形態に適した問題として、流れ作業ラインの物流制御問題を取り上げる。この問題は、実時間プラント制御への知識工学の応用問題であるため、プロセスとの簡易なインターフェース方式、誤った制御信号を出力しないための実用的なルールの合理性チェック方式などが必要である。プロセスとの簡易なインターフェース方式として、文字列、パラメータを与えるだけでトラッキングテーブルから推論機構のワークテーブルへの設備現況書き込み、推論機構で決定された制御指令の読み出しが可能な方式を提案する。ルールの合理性チェック方式として、人間が介入するが、排他の条件に着目し、条件と結論との相互関係の構造化を図る方法を提案する。また、提案方式が有効であったことを製鉄所のビレット精整ライン制御システムへの実適用を通じて示す。知識工学応用上の手順、実用面からの解析結果(応答性、拡張性、保守性、ソフト開発工数など)についても述べる。

第3章で問題の記述に知識工学を応用し、組合せ計算はアルゴリズムで対応する応用形態に適した問題として、搬送車群運行制御問題を取り上げる。ここでは、搬送要求への搬送車割付問題を条件判定的な問題と組合せ最適化的な問題に分離する構造を提案する。その条件判定的な問題では、搬送車によりステーションへパレットを搬送できるか否かの条件(搬送可否条件)、どの搬送要求から優先的に搬送車を割り付けるべきかの条件(搬送優先条件)などが指定され、それらの条件に基づき搬送可能な優先度付けされた搬送候補を求める。それらの条件は変更が頻発するため、変更の容易なルールにより記述する。一方、分離された組合せ最適化的な問題は、条件判定的な問題で求められた優先度付きの搬送候補に対する搬送車割付の問題である。これに対し、高効率で標準化された搬送車割付アルゴリズムを提案する。提案方式により変更が容易に実現できること、搬送効率が向上できることを例をもとに示す。また、提案方式を組み込んだ搬送車群運行制御シミュレータについても述べる。

第4章では問題の記述に知識工学を応用し、組合せ計算はアルゴリズムで対応する応用形態に適した問題の他の事例として、自動倉庫のピッキング効率向上運用制御問題を取り上げる。ここでは、設備構成として、小容量のピッキング用回転倉庫と、それに対してパレットを供給するための少数のクレーンを持つ固定倉庫(ラック+クレーン)からなる二段階倉庫を考える。ピッキング効率を向上するために、① 事前に分かっている注文内容のデータを利用し、処理順序が近い注文間に共通品目が多くなるように注文処理順序を決定する運用アルゴリズムと、② 決定された注文処理順序のもとで、回転倉庫上には近い将来、再度ピッキングされるパレットが格納されるように、回転倉庫から固定倉庫へ払い出すパレットを決定する運用アルゴリズムを提案する。また、納期条件等から全ての注文に対して上記の運用アルゴリズムは適用できず、それらの条件は対象によって異なり、変更も多いため、ある条件下で処理順序を並び換えてもよい注文群に分割するという条件判定的な問題はルールで記述する。さらに、提案運用方式の有効性をシミュレーションにより、固定倉庫のみのシステムとの比較で示す。

第5章で解法戦略ノウハウの記述にも知識工学を応用し、各解法戦略に基づく組合せ計算はアルゴリズムで対応する応用形態に適した問題として、作業者に対する作業計画問題を取り上げる。ここでは、広い分野で実際の計画業務に適用可能な、PERT/LOADを拡張した資源配分方式を提案する。具体的には、負荷山積みにおいて、各仕事の実施時刻を最適実施時刻に設定する。負荷平滑化手段として、仕事の後倒し、仕事の前倒し、仕事の削除の3つの手段を導入する。どの仕事にどの手段をどの様な場合に適用すべきかといった負荷平滑化戦略は対象ごとに異なり、変更も多いため、ルールで記述する。ルール記述された戦略に基づく負荷平滑化計算は各手段に対応したスケジューリングアルゴリズムで実行する。なお、スケジューリングアルゴリズムの部分は、仕事の後倒しによる負荷平滑化手段しか持たないPERT/LOADを拡張したものである。提案方式をスーパーマーケットの実際の作業計画業務に適用したので、その内容についても述べる。

最後に、第6章で、本研究全般にわたり、その結果の意義と残された問題点についてまとめる。

## 参考文献

- (1) N.J. Nilson: Principles of Artificial Intelligence, Tioga Publishing Co., Palo Alto (1980)
- (2) 都島, 田代, 薦田, 馬場, 高倉: 流れ作業ラインへのルール型制御方式の適用—製鉄所のピレット精整ライン制御への適用—, 計測自動制御学会論文集, Vol.21, No.10, pp107-114 (昭和60年10月)
- (3) 都島, 馬場, 高倉, 田代: ルール型制御方式の実プロセス制御への適用, 第23回SICE学術講演会予稿集, pp273-274 (昭和59年7月)
- (4) 高倉, 都島: FAシステムにおけるAIの応用, (株)トリケップス, pp193-203 (昭和62年2月)
- (5) 都島, 薦田, 岡坂: 知識工学を応用した搬送車群運行制御方式, 計測自動制御学会論文集, Vol.24, No.3, pp300-307 (昭和63年3月)
- (6) I. Tsushima and N. Komoda: Knowledge Engineering Approach to Traffic Control for Automated Guided Vehicles, Proc. of Int. Workshop on Industrial Applications of Machine Intelligences, pp261-265 (1987-2)
- (7) 都島, 薦田, 松本, 原田: プリント基板組立ショップにおけるルール型搬送車群制御方式, 第4回SICE知識工学シンポジウム, pp1-6 (昭和61年3月)
- (8) 都島, 三森, 薦田: 回転倉庫と固定倉庫を結合した自動倉庫システムの運用方式の提案, 電気学会論文誌C, Vol.107, No.9, pp843-849 (昭和62年9月)
- (9) 鈴木: オーダピッキングの種類と手法, オーダピッキングハンドブック, Vol.28, No.10, pp8-16 (昭和56年10月)
- (10) 都島, 薦田, 楠崎, 井下田: 知識工学応用・流通業向ワークスケジューリングシステム, 第5回SICE知識工学シンポジウム予稿集, pp51-54 (昭和62年3月)
- (11) 都島, 薦田, 楠崎, 井下田: 流通業向・ワークスケジューリングシステムにおける作業割付方式, 情報処理学会第34回全国大会, 2G-4, pp2161-2162 (昭和62年3月)
- (12) 都島, 薦田: 負荷山崩し機能を拡張した資源配分手法の提案 (電気学会論文誌Cに投稿中)
- (13) S.M. Johnson: Optimal Two-and-Three-Stage Production Schedules with Set-up Times Included, Nav. Res. Log. Quart.1, No.1, (1954)
- (14) Z.M. Lomnicki: A Branch-and-Bound Algorithms for the Exact Solution of the Three Machine Scheduling Problem, Operational Research, Quart.16, No.1 (1965)
- (15) 家長, 大西, 成合, 田中: 製鋼工場のオンライン物流管制システム, オペレーションズ リサーチ, Vol.72, No.6, pp353-359 (昭和61年6月)
- (16) 福村, 松川, 外岡: 製鋼工場から冷延工場までの一貫スケジューリング問題, オペレーションズ リサーチ, Vol.72, No.6, pp346-352 (昭和61年6月)
- (17) 三浦, 丸山, 島田, 山本: 自動化推進のためのエキスパートシステム開発例, 電気学会論文誌C, Vol.108, No.4, pp238-245 (昭和63年4月)

- (18) C.L.Haines: An Algorithm for Carrier Routing in a Flexible Material-Handling System, IBM J. Develop., Vol.29, No. 4, pp356-362 (1985-7)
- (19) 安部, 鈴木, 原島: シティーカーシステムにおける車両再配置効果のシミュレーションによる検討, 計測自動制御学会論文集, Vol.16, No.4, pp525-532 (昭和55年8月)
- (20) 篠塚, 関: FAにおける無人搬送と貯蔵の自動化, 計測と制御, Vol.22, No.11, pp931-936 (昭和58年11月)
- (21) 藤井, 三森, 河内, 綿谷, 高田: 搬送シミュレータを用いたカート運用法に関する基礎的研究, Proc. of the First Intelligent FA Symposium, JAACE, pp3-4 (昭和62年6月)
- (22) 飯田: PROLOGを用いた車両割当てシミュレーション, 日本シミュレーション学会, pp139-144 (昭和59年11月)
- (23) W.H.Hausman: Optimal Storage Assignment in Automatic Warehousing Systems, Management Science, Vol.22, No.6, pp629-638 (1976-2)
- (24) 三留: ストレージシステムの搬送距離最小化動的スケジューリング方式, 電気学会論文誌C, Vol.104, No.7, pp157-164 (昭和59年7月)
- (25) W.H.Hausman: Storage-Retrieval Interleaving in Automatic Warehousing Systems, Management Science, Vol.23, No.9, pp935-945 (1977-5)
- (26) 藤井, 三森, 綿谷: 水平回転棚の運用法の最適化に関する研究, Proc. of the First Intelligent FA Symposium, JAACE, pp27-30 (昭和62年6月)
- (27) 村松: 生産管理の基礎, 国元書房, pp192-216 (昭和56年3月)
- (28) 橋本: ジョブショップスケジューリング, Computrol, No.11, pp61-69 (昭和60年8月)
- (29) R.W.Conway: Priority Dispatching and Work-in-Process Inventory in a Job Shop, J. Ind. Eng., Vol.16, No.2 (1965)
- (30) 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, pp66-71 (昭和62年1月)
- (31) 刀根: パート講座1, 東洋経済新報社 (昭和52年7月)
- (32) 大場, 都島, 岩本: ストレージを持つ離散システムへのPERT/LOADの拡張アルゴリズム, 電気学会論文誌C, Vol.104, No.3, pp49-56 (昭和59年3月)
- (33) I.Tsushima, S.Mitsumori, M.Ooba, T.Iwamoto and H.Furuyado: An Extension of Resource Allocation Problem, Proc. of the 1980 Int. Conf. on Cybernetics and Society, pp253-257 (1980-12)
- (34) I.Tsushima, S.Mitsumori, K.Akashi, T.Iwamoto and M.Ooba: A Resource Allocation Method and An Identification Method for Discrete Systems, Proc. of the 8th Triennial World Congress of the Int. Federation of Automatic Control, pp3573-3578 (1981-8)
- (35) 都島, 岩本, 大場: 資源配分問題の拡張と物流システムへの応用, 電気学会, システム・制御研究会, SC-80-17, pp11-19, (昭和55年9月)
- (36) D.A.Bourne: Autonomous Manufacturing: Automating the Job-Shop, IEEE Computer

Magazine, Vol.17, No.9, pp76-86 (1984-9)

(37)B.Bruno, A.Elia and P.Laface: A Rule-Based System to Schedule Production, IEEE Computer Magazine, Vol.19, No.7, pp32-40 (1986-7)

(38)戸沢, 沼尾, 森下: 製鋼工程スケジューリング・システム: Scheplan(1), 情報処理学会第36回全国大会, pp1547-1548 (昭和63年3月)

(39)B.Chandrasecaran: Generic Tasks in Knowledge-Based Reasoning, IEEE Expert Magazine, pp23-30 (1986-Fall)



## 第2章 問題と解法の記述に知識工学を応用 した流れ作業ライン物流制御方式

## 第2章 問題と解法の記述に知識工学を応用した流れ作業ライン物流制御方式

### 2.1 まえがき

今日、ユーザ要求は多様化しており、その内容は短期間に変化している。そのため、製品のモデルチェンジに伴う、製造ラインの頻繁な変更への迅速な対応が Factory Automation (FA) システムに強く要求されている。従来、このようなシステムの制御ソフトは、手続き型で汎用のプログラム言語 (FORTRAN 等) により開発されている。しかし、制御論理がプログラム中に分散し、埋もれてしまうため、一度書かれたプログラムから制御論理を読み出すことが作成者でも困難な場合が多い。また、制御知識の利用順序を規定する命令語が知識間を強く結び付けてしまうため、知識の追加、変更、削除が困難となる。さらに、制御論理を考察する生産技術者は一般にソフトの非専門家であるため、プログラマを介しないと制御論理の変更ができない。

最近、前述した問題点を解決するために、知識工学応用のアプローチが現われてきている。これは、プロダクションシステムに代表されるように、対象の論理を“IF (条件) THEN (結論)”型のルールとして表現し、その論理の利用順序はシステムが用意する推論機構が自動的に決定するもの (非手続き型) である。これにより、データ構造を考える必要がなく、論理を利用順序と独立に記述でき、日本語的な表現もそのまま扱うことができるので、生産技術者でもプログラムに組み込まれた論理を直接理解できるようになる。

知識工学の応用例には、研究レベル、実験レベルでは、鉄道ダイヤの乱れを回復させるスケジュールを作る運転整理問題<sup>1)</sup>、電力系統事故時の復旧系統決定の問題<sup>2)</sup>、原子炉等のプラント異常診断の問題<sup>3)</sup>、下水処理プロセスの運用支援問題<sup>4)</sup>などがある。

一方、実用レベルのシステムは、計算機システム構成設計支援システム<sup>5)</sup> R1、石油鉞脈試掘データ解析システム Dipmeter Advisor<sup>7)</sup> 等、極めて少ない。この理由として、実システムでは、推論機構はシステムの一部にしかすぎず、まわりに多くの機能が必要なことが考えられる。たとえば、Dipmeter Advisor に関する報告では、推論機構のコード量は 8%、ルールは 22% と報告されている。従って、知識工学応用システムの実用化のためには、まわりに従来ソフトによる十分なサポートを行うとともに、それらと推論機構との接続の方式を確立する必要がある。しかし、推論機構を含めた知識工学応用システムとしての実現方法については、ほとんど報告されていない。第2の理由としては、推論機構に入れる論理の完全性の問題である。実用システムでは常に正しい結論の出ることが要求される。ルールチェックに関しては、メタ知識を活用することで入力ルールを構文上、意味上、因果関係上で検査し、ルール間の合理性を自動的に保持することを目指す知識獲得システムの研究<sup>8)</sup>がなされているが、ルール間あるいは入力されている知識間の整合性をチェックするレベルであり、論理そのものの良し悪しを論じるものではない。

本章では、問題と解法の両者の記述に知識工学を応用する形態に適した問題として、流れ作業ラインの物流制御問題を取り上げる。実時間制御においては、種々のセンサ・アクチュエータがインターフェースとして使用されるため、それらと推論機構との簡易なイン

ターフェース方式が必要である。また、実時間制御への知識工学の応用では、ルール記述されたプログラムが人間の介在が許されない閉ループコントロールの中で動くため、誤った制御信号を出力しないように実稼動前に十分にルールの合理性チェックが行える方式が必要である。そこで、製鉄所のピレット精整ラインの物流制御に知識工学を実適用した事例をもとに、プロセスとの簡易なインターフェースを考慮した、ルール型制御システムの構成を示し、ルール記述された制御論理の合理性チェックの現実的な方法を提案する。つぎに、ピレット精整ライン制御システムの開発過程の分析から、制御論理の変更、段階的バージョンアップが極めて容易になったこと、応答時間に占める推論機構の処理時間の割合、ソフト開発工数、提案するルールチェック方法が有効であったことなどを、実データにより示す。

## 2.2 制御対象のモデル

FAでは、物の分岐、合流制御や設備への作業割付制御など、各種条件の組合せで制御指令が決まるタイプの制御が多くある。その一例として、1段の流れ作業工程の例を図2.1に示す。本工程では、分岐点に到着したパレットをコンベアで作業ステーションに搬送し、作業完了後、払出しライン(合流点)へ搬送する。各ステーションは複数種類の作業を実施でき、その能力は、設備番号が小さいほど大とする。この工程の分岐制御をルール記述すると、図2.1に示す4ルールになる。なお、このルールの $\langle \min. z \rangle$ は、条件を満足する $z$ のうち、最小の値(能力大の作業ステーション番号)を選択するユーザ手続きである。

図2.1に示した1段の工程を多段につなげた流れ作業工程、また、無人搬送車を物流手段とするジョブショップ

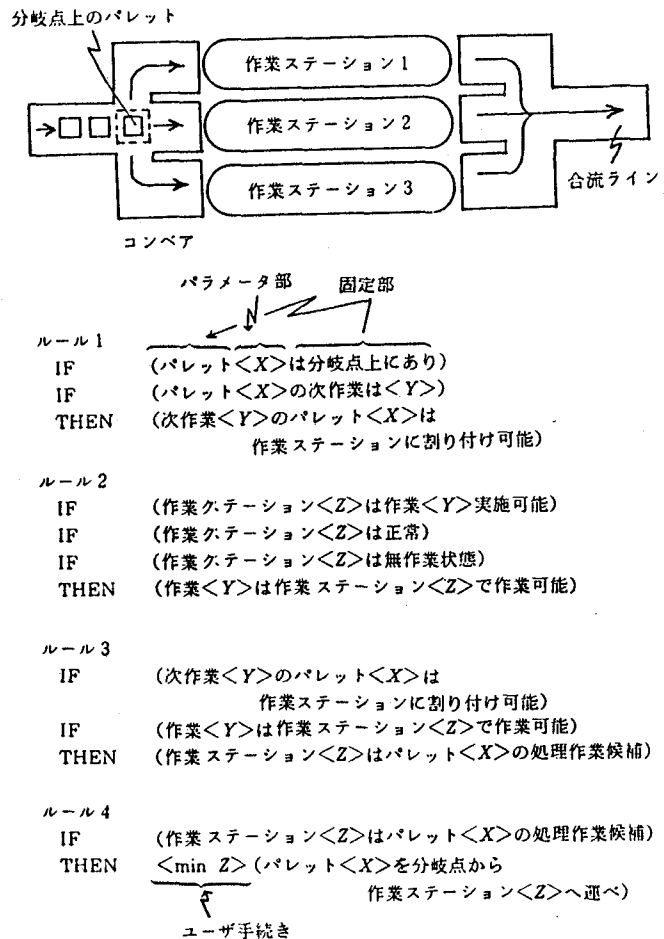


図2.1 単一流れ作業ラインと制御ルール例

型の作業工程の場合も、物流制御、作業割付制御等の多くの制御は条件組合せ型制御となり、それらはルール型制御向きの制御対象である。今回、ルール型制御方式を適用したピレット精整プロセス制御は、典型的な流れ作業工程の物流制御である。

### 2.3 ルール型制御システム S C D (Station Coordinator<sup>14)</sup>-13)

本節では、IF-THENルールの記述形式、ルール入力用画面エディタ、推論機構および推論機構と他ソフトとの関係、変更の容易性を考慮したルール型制御システムとしてのソフト構成およびソフトの作り方について述べる。

#### 2.3.1 IF-THENルール

制御論理を日本語により“IF(条件)  
THEN(中間結論あるいは最終結論(制御指令))”のルールで与える。

IF-THENルールは、ルールの種類を宣言するTYPE部と、条件を記述するIF部と、その条件を満たした場合の結論を記述するTHEN部より構成される。IF部、THEN部は各々複数の条件、結論を記述できる。IF部の複数個の条件はAND条件として取り扱う。条件、結論を記述する文字列は、“(”, “)”

で囲われ、固定部とパラメータ部よりなる。固定部は条件や結論の内容を表わし、パラメータ部は、“<”, “)”でくくり固定部と区別する。また、THEN部には、外部手続きとして組み込まれる、ルールでは記述出来ない(あるいは、記述すると膨大な量になる)演算処理をコールするための外部手続きを指定することができる。IF-THENルールの構文規則を図2.2に示す。なお、同一の状況で使用するルールはルール群としてまとめ、“\*”で始まる文でルール群の宣言を行う。起動するルール群はイベントに対し前もって設定できるほか、推論の途中結果により次に使用するルール群を選択することもできる。

固定部は条件や結論の内容を表わし、パラメータ部は、“<”, “)”でくくり固定部と区別する。また、THEN部には、外部手続きとして組み込まれる、ルールでは記述出来ない(あるいは、記述すると膨大な量になる)演算処理をコールするための外部手続きを指定することができる。IF-THENルールの構文規則を図2.2に示す。なお、同一の状況で使用するルールはルール群としてまとめ、“\*”で始まる文でルール群の宣言を行う。起動するルール群はイベントに対し前もって設定できるほか、推論の途中結果により次に使用するルール群を選択することもできる。

#### 2.3.2 ルール入力用画面エディタ

ルールエディタは、ルールの定義をマンマシンで誤りなく作成、修正できるエディット機能に加え、誤り検出を支援するルール構造チェッカー、ルール動作シミュレータを備えている。

##### (1)エディット機能

エディタの開始画面において、エディットしたいルール群名を指定すると、図2.3に示すようなエディット画面が表示される。本画面において、スクロール等の画面操作、文字列の検索等、処理中のルール群全体を対象とした基本コマンド(表2.1)と、文字列のコピー、削除等のルール編集用の投入した行に対してのみ有効な行コマンド(表2.2)が使用できる。また、文字列(最大70文字)内の文字の削除、挿入等の指定はファンクションキ

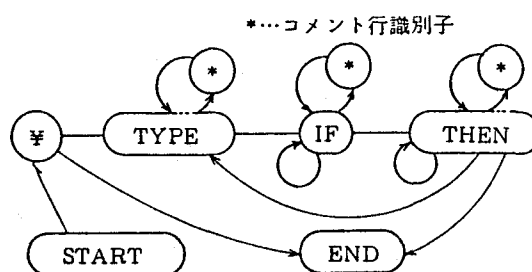


図2.2 ルールの構文規則

ーにより実行できる。入力ガイダンス出力領域には、エディタ側から図2.2の構文規則に従い入力すべき文節(IF部, THEN部等)を示す“IF”, “THEN”等の識別子を出力し、正しい文字列の入力を誘導する。誘導すべき文節が複数許される場合には、前もって設定された優先順位に従って順に識別子が表示される。なお、コメント行は任意の場所で入力可能であるので、誘導は行わず、入力された文字列の先頭が“\*”であることでコメント行と判定し、識別子を“\*”に変更する。入力された文字列は、1行ごとに固定部、パラメータ部に分割され、カッコの抜けやパラメータ長のチェック等の文節単位のシンタックスチェックを行い、エラーの場合はメッセージを出力すると同時にカーソルを該当文字列のエラー検出位置に設定し、その文字列を受け付けない。入力されたルールは、ルール群単位で管理され、ユーティリティ機能を用いてコピー、削除、リネーム等が行える。

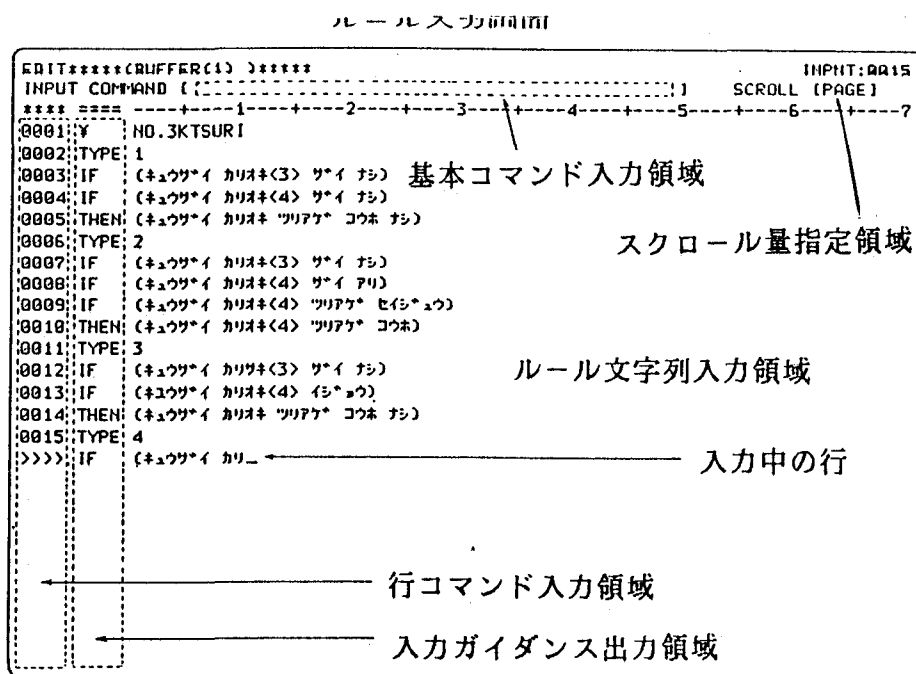


図2.3 エディット画面の構成

(2)ルール構造チェッカー

ルール構造チェッカーは、ルール間の結論、条件のつながりを図示(ルール構造図と呼ぶ)し、ルール作成者に対し表示することで、あるルールの制御指令以外の結論が他のどのルールの条件判定にも用いられないといった誤りの検出を支援するものである。

ルール構造チェッカー出力画面の例を図2.4に示す。ルール構造図は、正方形および円をノードに持つ階層グラフであり、正方形はルール、円はルールに記述された条件、および結論を示す。画面の左から右が推論の進む方向となっている。例えば、図2.4の例でルール05は、条件01と06とが満足されると結論02を下し、さらに、その結論はルール12の

表 2.1 基本コマンド一覧表

コマンド	オペランド	機能
C	文字列1、文字列2	文字列1を文字列2と置き換える
F	文字列	指定された文字列の存在する行を検索する
I	行番号	指定行の直後に入力可能行を挿入する
U		1画面分表示範囲を上方にスクロールする
D		1画面分表示範囲を下方にスクロールする
L	行番号	指定行を画面の先頭に表示する
M	ルール群名	指定ルール群のルールを編集中のルール群の中へコピーする
Q	S, N	編集処理を終了する (S:セーブ、N:非セーブ)

表 2.2 行コマンド一覧表

コマンド	機能
A	本コマンドを入力した行の直後にコピー又は移動をする
B	本コマンドを入力した行の直前にコピー又は移動をする
C	1行をコピーする
CC	複数行をコピーする
D	1行を削除する
DD	複数行を削除する
I	1行を挿入する

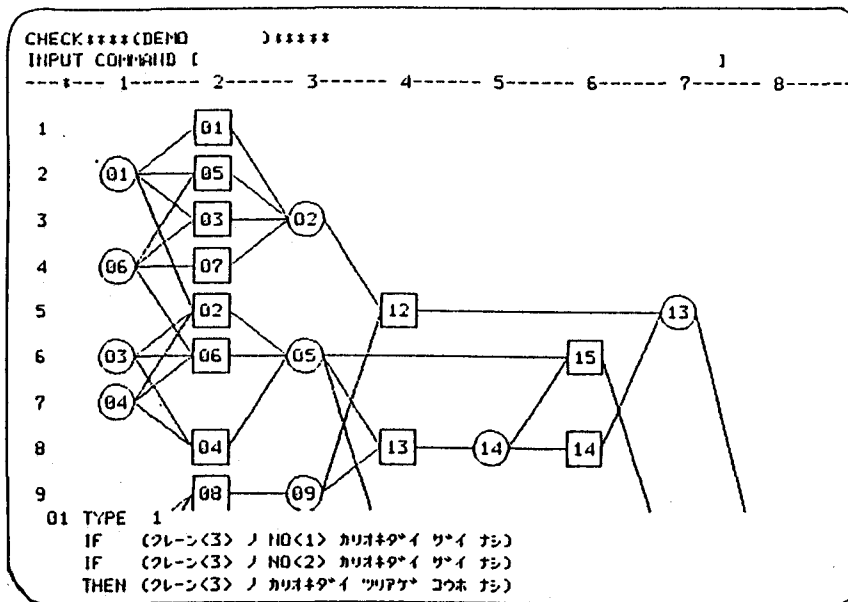


図 2.4 ルール構造チェッカーの画面例

条件判定に用いられる。ユーザは本画面で、一番左端の条件がすべて制御対象状態であるか、パスのつながりに不備、自分の思惑と異なる点は無いか等をチェックしていく。この際、ルール構造図のスクロール、ルール内容の表示等のコマンドが使用できる。

### (3) ルール動作シミュレータ

ルール構造チェッカーでは変数値の誤りをチェックできないので、現実の制御対象状態に対して果たして正しい制御指令が結論されるか否かは完全にはチェックできない。ユーザは、ルール構造チェッカーにてルールの構造上の誤りを検出、修正した後、さらに、ルール動作シミュレータにより変数の記述されたルールに対しても推論が正しく進むかをチェックする。このために、ルール動作シミュレータは、端末からセットされた任意の対象

状態に対して、ルールの推論を実行し、その過程を表示する機能を有している。

### 2.3.3 推論機構

通常のプロダクションシステムでは、ワークメモリ内のすべての文字列とすべてのルールの文字列の一致判定を実施し、一致するものに対し変数値の取り込み操作を行う。このため、ワークメモリに格納されている文字列数、ルール数の増加に伴い、一致判定操作の所要時間が爆発的に増加するという問題がある。これを解決するために、図2.5に示す高速推論機構を開発した。

本推論機構では、IF-THENルールの文字列の固定部は処理的には文字列どうしを区別するための情報でしかなく、パラメータ部の内容が制御指令決定のための情報となる点から、パラメータ部のみをそのまま扱い、固定部を1つのコードに対応させるという文字列の内部形式化を行う。さらに、ワークメモリには、同一コード毎にパラメータ部の情報をポインタで結合して格納する。このことにより、ワークメモリ内の文字列とルールの文字列の一致判定を行うことなく処理に必要な情報のみを取り出すことが可能となり、一致判定に伴う処理時間の爆発的な増加を防ぐことができる。さらに、本機構では、現実の制御では、制御指令の決定に際し判定すべき条件が、システムの状態変化の内容、すなわち、イベントによって限られてくるという性質を利用し、あらかじめルールをイベントの種類に対応したルール群に分類しておく。これにより、必要のないルールに対する変数値の取り込み処理が省かれ、処理速度を向上できる。

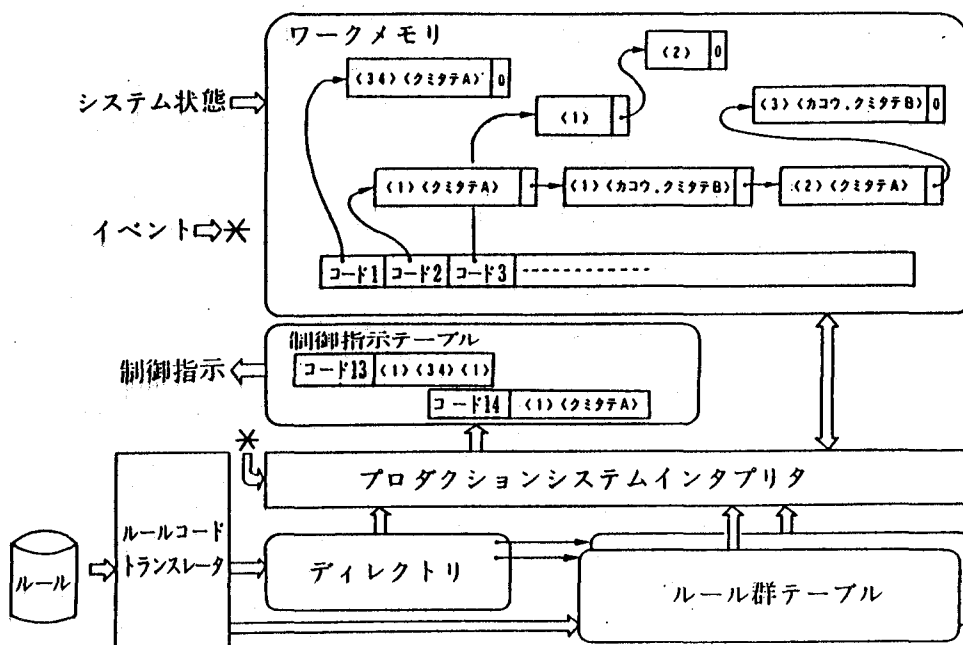


図2.5 高速推論機構

本機構の動作を説明しておく。IF-THENルールは、あらかじめ、ルールコードトランスレータで内部形式に変換されルール群毎にルール群テーブルに格納される。各ルール群の所在とイベントとの対応情報はディレクトリに格納される。この状態でイベントが入力されると、プロダクションインタプリタは、システム状態をワークメモリに取り込み、ディレクトリから使用ルール群を選択し、そのルール群を用いて推論を行う。決定された制御指令は、制御指令テーブルに格納され、外部に出力される。

### 2.3.4 ソフト構成

プロセスからの信号の検出、その信号をもとにしたトラッキング、制御指令から制御信号への変換の3つのプログラムはほぼ、標準化されている。そこで、文字列、パラメータを与えるだけでトラッキングテーブルから推論機構のワークメモリへの設備現況書き込み、推論機構で決定された制御指令の読み出しを実現するインターフェースルーチンを設けた。このルーチンを含め、図2.6により、ルール型制御システムSCDのソフト構成を説明する。

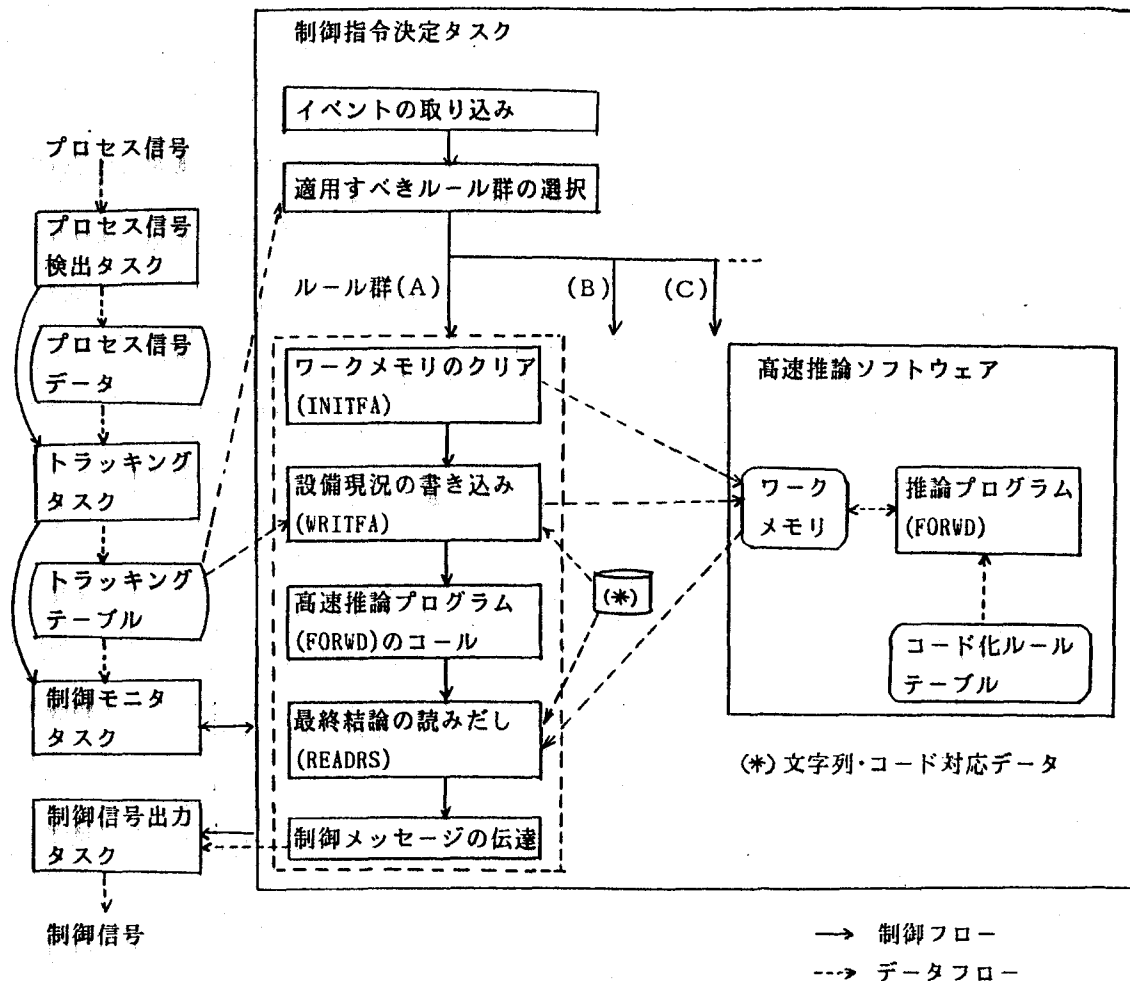


図2.6 ルール型制御システムのソフト構成



制御モニタタスクはトラッキングタスクから起動されると、発生したイベント内容を調べ、制御指令決定タスクへ知らせる。制御指令決定タスクはイベント情報を取り込み、トラッキング情報も考慮して、適用すべきルール群を選択する(使用した推論機構では、推論時間の短縮のため、予めルールをイベントの種類に対応してルール群に分類している)。そして、各ルール群処理では、推論機構のワークメモリをクリアするルーチンINITFA、つぎに設備現況を書き込むルーチンWRITFAがコールされる。WRITFAには、書き込みたい設備現況を示す文字列とその現況の具体的なパラメータ値を引数として与える。高速推論プログラムFORWDで決定された制御指令は、最終結論読み出しルーチンREADRSで読み出され、それは制御信号出力タスクで具体的な制御信号へ変換され、設備へ発信される。なお、READRSには、読み出したい制御指令を示す文字列と変数を与える。

つぎに、ルール型制御システム全体としての変更容易性の実現に関して述べる。実機デバッグでは制御論理の変更は、再コンパイル、再リンケージ等のための時間がなかなか取れず、変更論理を再実行するまでに長時間かかっている。この解決には、データの変更だけで対応可能にすることである。制御論理の修正が、設備現況のワークメモリへの書き込みプログラムの修正に及ばないように、ルール群に関係なく、一括して設備現況の登録を行うソフトの作り方をすることが必要である。

## 2.4 ルールのチェック方法

ルールのチェック方法として、ルール間のつながり図を作成してチェックする静的な方法(2.3.2で述べたルール構造チェッカー)、設定した条件に対して所望の結論が得られるかをシミュレートする動的な方法(2.3.2で述べたルール動作シミュレータ)がある。しかし、これらは、制御論理そのものの良し悪しをチェックできるものではない。

知識工学の適用対象は、ルール記述されたプログラムを人間の介在の許されない閉ループコントロールの中で動かさなければならない実時間制御のため、実稼動前に条件そのものの抜けやルールそのものの抜け等の論理自体のチェックが確実にできる方法が必要であり、人間が介在するが、その要求を満たす、現実的な方法を以下に提案する。

ルールそのままの形では、条件、結論が全て記述されているか、各条件がどの結論と関連しているかが把握しにくく、制御論理の十分なチェックは困難である。この解決には、条件と結論との相互関係の構造化が必要である。そこで、排他の条件を1つの条件項目とし、図2.7の(2)のように、各ルール毎にその条件を該当する条件項目のもとに位置付ける。また、設備の使用法の違いから下記の原則に従いルールを分類し、変数を含む条件項目にすべきか、定数指定の条件項目にすべきかを決定する。

(タイプ1) 同種の設備に共通の使用方法を指定するルール(パラメータ部を変数で指定するルール) …変数を含んだ条件項目で扱ってよい。

(タイプ2) 同種の設備群に対し、個々に使用方法を定める(パラメータ部を定数で指定するルール)が、設備間にその使用に関し優先順序等の相互関係を指定するルール…変数付きの条件項目とすると、優先順序記述のため、1つの条件項目下に複数の条件を設定することになり、これが他の設備番号のルールとの相互関係をわかりにくくする。そのため、

定数指定の条件項目とする。

(タイプ3) 同種の設備群に対し、個々に使用方法を定める(パラメータ部を定数で指定するルール)が、それが各設備固有の条件のみで決まるルール…図2.7のルール1, 2が対応する。ルール自体のパラメータは定数であるが、各設備の使用は、各設備固有の条件で決まるため、タイプ2での問題は発生しない。そのため、変数を含む条件項目とする。

	Condition 1 GRダイレヤ(X); セイジョウ/ イジョウ	Condition 2 GR(X); セイジョウ/ イジョウ	Condition 3 GRダイレヤ(X); ダイアリ/ ダイナシ	Condition 4 ナイタイレダイ; アリ/ ナシ		
F 1 IF (GRダイレヤ(3)セイジョウ) IF (GR(3)セイジョウ) IF (GRダイレヤ(3)ダイ ナシ) THEN (GRダイレヤ(3)レヨウ カノウ)	Rule 1	セイジョウ(X=3)	セイジョウ(X=3)	ダイナシ(X=3)	—	GRダイレヤ(3) レヨウ カノウ
F 2 IF (GRダイレヤ(4)セイジョウ) IF (GR(4)セイジョウ) IF (GRダイレヤ(4)ダイ ナシ) IF (ナイタイレダイ ナシ) THEN (GRダイレヤ(4)レヨウ カノウ)	Rule 2	" (X=4)	" (X=4)	" (X=4)	ナシ	GRダイレヤ(4) レヨウ カノウ
F 3 IF (GRダイレヤ(X)イジョウ) THEN (GRダイレヤ(X)レヨウ フカ)	Rule 3	イジョウ	--	—	—	GRダイレヤ(X) レヨウ フカ
F 4 IF (GR(X)イジョウ) THEN (GRダイレヤ(X)レヨウ フカ)	Rule 4	—	イジョウ	—	—	"
F 5 IF (GRダイレヤ(X)ダイ アリ) THEN (GRダイレヤ(X)レヨウ フカ)	Rule 5	—	—	ダイアリ	—	"
F 6 IF (ナイタイレダイ ナシ) THEN (GRダイレヤ(4) レヨウ フカ)	Rule 6	—	—	—	アリ	GRダイレヤ(4) レヨウ フカ

(1) ルール例

(2) 提案したルールチェック様式例

図2.7 ルールの合理性チェック方式

## 2.5 ビレット精整ラインのルール型制御

### 2.5.1 ビレット精整ライン

ビレット精整ラインでは、圧延後の、10m前後の棒材であるビレットを検査、手入れ(研削)、仕分けし、出荷する(図2.8)。

まず、材を探傷機1で検査する。無傷の材(無手入れ材と呼ぶ)はテーブル(TA)4~10を直進し、仕分けラインへ搬送される。傷のある材(手入れ材と呼ぶ)はテーブル7から移載機3にて台車3あるいは4に移され、グラインダ3あるいは4で研削後、移載機4にてテーブル8に戻される。研削後の材は、探傷機2にて再検査され、傷なしとなった材は仕分けラインへ、まだ、傷のある材は台車4を介してグラインダ4にて再手入れされる。手入れ前の材を給材、手入れ後の材を集材と呼ぶと、移載機3は給材処理だけを、移載機4は集材処理だけを行う。稼働中の移載機3, 4のエリアをゾーン2、ゾーン2の前に同様の設備群が将来増設されるエリアをゾーン1と呼ぶ。なお、移載機的一方が故障時、

同一ゾーン内の他方が給材と集材の両処理を行う。この場合、材は双方向に流れる。仮置き設備では材の円滑な流れを確保するために一時保管を行う。

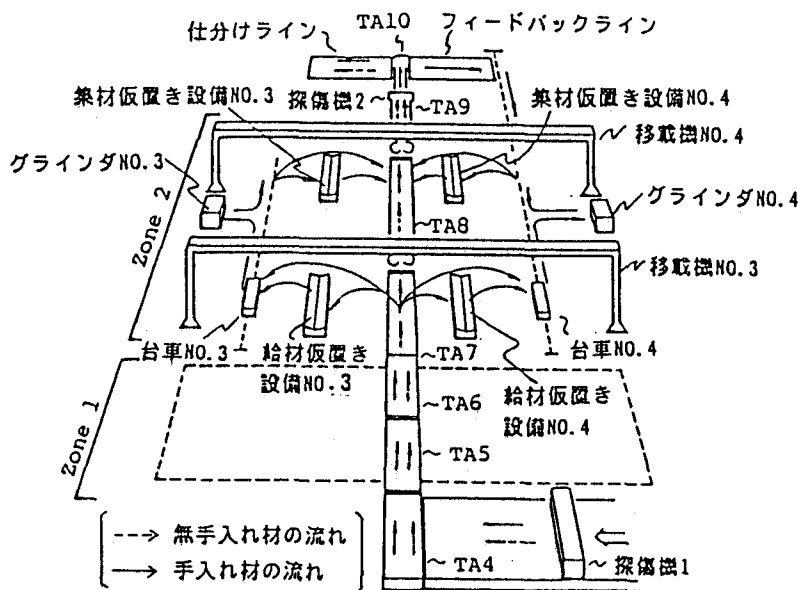


図 2.8 ピレット精整ライン

### 2.5.2 制御ルール

図 2.9 に実稼動直後のルール群の全体構成を示す(ルール数の合計は 260)。以下に、分割の考え方を示す。

- (1)ゾーン間の負荷を平滑化するための材の行き先ゾーン決定，デッドロック防止のための材の投入タイミング制御は投入口でしか実施できないので，投入口制御とゾーン内制御を分ける。
- (2)各移載機制御にはかなり共通機能があるが，制御論理の修正時や移載機の増設時に容易に対応できるようにするため，移載機別に制御を分ける。
- (3)異常時制御は，正常時制御に比べ，使用頻度が非常に少ないため，両制御を分ける。
- (4)正確な研削時間の予測は難しいため，早めに移動先を決め，移動すると，処理待ちが大となる。この解決のため，吊り上げ位置の決定と卸し位置の決定とを分ける。

つぎに，ルール群内のルール分割の考え方を示す。

制御指令は，各種設備の使用可能状況の組合せから決まる。そのため，設備の使用可能状況を抽出する中間結論ルールを設備の種類毎に作る。制御指令となる最終結論ルールは，絞られた中間結論群を条件として作成する。これにより，一度に考慮すべき条件数が限られるため，ルールが容易に作成できる。

上述した考え方を全ルール群に適用したが，紙面の都合から，図 2.10 に示す比較的ルール数の少ない移載機 3 給材卸しルール群のみを説明する。

制御方針は，グラインダ稼働率の向上から，卸せる台車があればそこに卸す(ルール

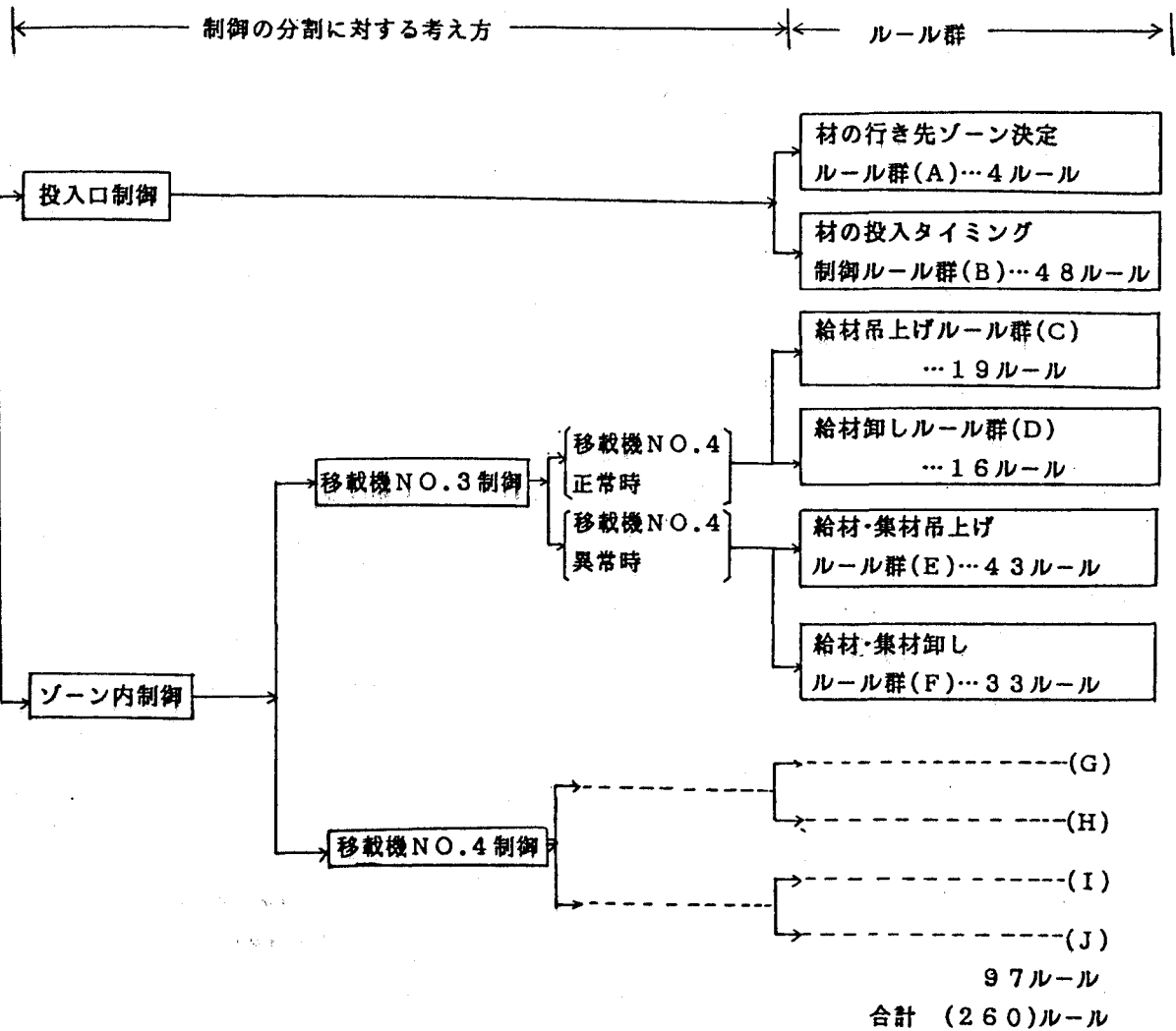


図 2.9 ルール群の全体構成

15)。卸せる台車はないが、仮置き設備に卸せる場合、次材(給材)が既にテーブル7に来ていれば、仮置き設備に卸す(ルール16)。以上の最終結論を出すには、台車に卸せるか、仮置き設備に卸せるか等の中間結論を出すルールが必要であり、それは設備毎に考えればよい。

中間結論1のルール1～6は、台車、グラインダの正常/異常、台車上に材あり/なし、再手入れ材があり/なしの条件から、台車3,4が使用可能か不可かを結論する。中間結論2のルール7～9は、中間結論1を条件に、卸し候補とする台車番号を結論する。グラインダ4の方は再手入れ作業を優先させるため、台車3を4より優先する(ルール7,8)。ただし、両台車が使用不可ならば、次材にて判定する(ルール9)。中間結論3のルール10は、テーブル7に給材があるかどうかを判定する。中間結論4のルール11～13は、仮置き設備3を4より優先することも考慮し、卸し候補とする仮置き設備番号を結論する。中間結論5のルール14は、移載機3が作業中でなく、材を掴んでいることを確認して、卸し材ありを結論する。以上、設備毎に作成したルールの中間結論1～5の組合せが、最

終結論ルール15, 16の結論となる。

\*\*\*\*\*  
 \* No(3) キュウザイ オロシ ルール \*  
 \*\*\*\*\*

\*チュウカン ケツロン 1\*

```
F 1
IF (GRダイジャ<3>セイジョウ)
IF (GR<3>セイジョウ)
IF (GRダイジャ<3>ザイ ナレ)
THEN (GRダイジャ<3>シヨウ カノウ)
F 2
IF (GRダイジャ<4>セイジョウ)
IF (GR<4>セイジョウ)
IF (GRダイジャ<4>ザイ ナレ)
IF (サイタイレザイ ナレ)
THEN (GRダイジャ<4>シヨウ カノウ)
F 3
IF (GRダイジャ<X>イジョウ)
THEN (GRダイジャ<X>シヨウ フカ)
F 4
IF (GR<X>イジョウ)
THEN (GRダイジャ<X>シヨウ フカ)
F 5
IF (GRダイジャ<X>ザイ アリ)
THEN (GRダイジャ<X>シヨウ フカ)
F 6
IF (サイタイレザイ アリ)
THEN (GRダイジャ<4>シヨウ フカ)
```

\*チュウカン ケツロン 2\*

```
F 7
IF (GRダイジャ<3>シヨウ カノウ)
THEN (GRダイジャ<3>オロシ コウホ)
F 8
IF (GRダイジャ<3>シヨウ フカ)
IF (GRダイジャ<4>シヨウ カノウ)
THEN (GRダイジャ<4>オロシ コウホ)
F 9
IF (GRダイジャ<3>シヨウ フカ)
IF (GRダイジャ<4>シヨウ フカ)
THEN (ジザイ ニテ ハンタイ)
```

\*チュウカン ケツロン 3\*

```
F 10
IF (タイレ<7>ザイ アリ)
IF (タイレ<7>イキサキ ゾーン<2>)
THEN (ジザイ アリ)
```

\*チュウカン ケツロン 4\*

```
F 11
IF (キュウザイ カリオキ<3>ザイ ナレ)
IF (キュウザイ カリオキ<3>セイジョウ)
THEN (キュウザイ カリオキ<3>オロシ コウホ)
F 12
IF (キュウザイ カリオキ<3>ザイ アリ)
IF (キュウザイ カリオキ<4>セイジョウ)
IF (キュウザイ カリオキ<4>ザイ ナレ)
THEN (キュウザイ カリオキ<4>オロシ コウホ)
F 13
IF (キュウザイ カリオキ<3>イジョウ)
IF (キュウザイ カリオキ<4>ザイ ナレ)
IF (キュウザイ カリオキ<4>セイジョウ)
THEN (キュウザイ カリオキ<4>オロシ コウホ)
```

\*チュウカン ケツロン 5\*

```
F 14
IF (イサイキ<3>サギョウチュウ OFF)
IF (イサイキ<3>ザイ アリ)
IF (イサイキ<3>セイジョウ)
THEN (オロシ ザイ アリ)
```

\*サイシユウ ケツロン\*

```
R 15
IF (GRダイジャ<X>オロシ コウホ)
IF (オロシ ザイ アリ)
THEN (GRダイジャ<X> オロセ)
R 16
IF (ジザイ ニテ ハンタイ)
IF (ジザイ アリ)
IF (キュウザイ カリオキ<X>オロシ コウホ)
IF (オロシザイ アリ)
THEN (キュウザイ カリオキ<X> オロセ)
```

図 2.10 移載機3の給材卸しルール群

### 2.5.3 制御システムの構成

計算機ハードウェア構成を図2.11に示す。プロセス信号検出, 制御信号出力は16ビットマイコンHIDIC-08Lで, それらを結び付ける制御出力決定(推論機構を含む)等は制御用ミニコンHIDIC-V90/50で行っている。

ソフトウェア構成は2.3で述べた図2.6に従っている。留意した点を以下に示す。

(1)2.3で述べたように, 現地デバッグでの修正がデータの変更だけで済むように, 表2.3に示す, きめ細かい設備現況を, ルール群に関係なく, 一括して推論機構に取り込んだ。

(2)制御指令の方は, マイコンとのインターフェースをシンプルにするため, 必要最小限とした。

(3)ルールの振り分けは, 応答性を高めるため, 各イベントに対し, 多くとも2つのルール群が動けば済むように, 細かい振分けとした。

なお, 図2.6に示したソフトウェア構成において, 推論プログラムとインターフェー

スルーチンは、FORTRAN77により、他のモジュールはSPL(Structured Programming Language)により作られている。

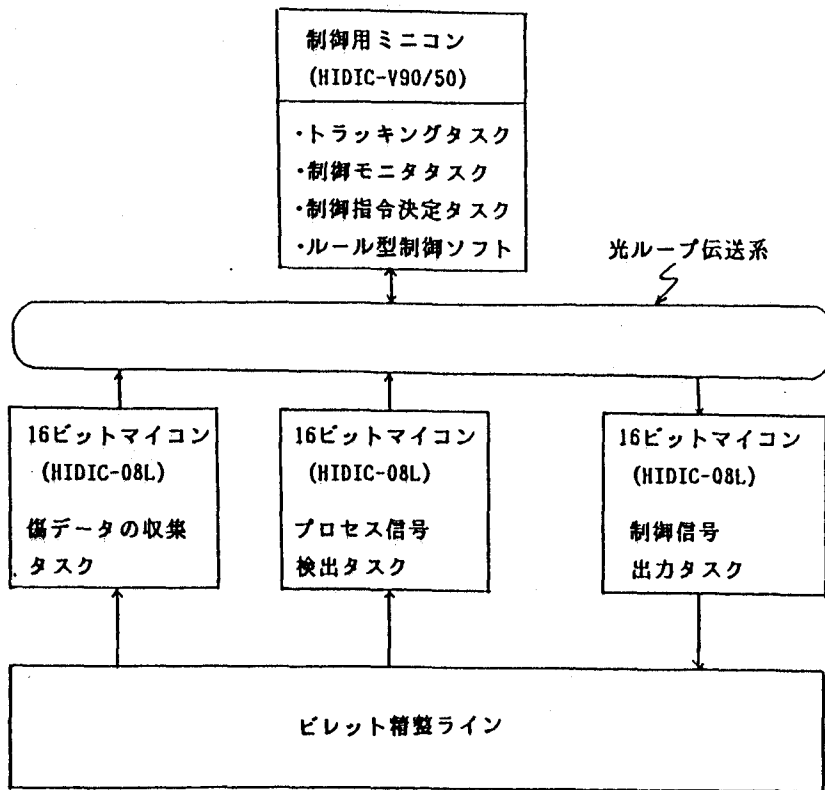


図 2.11 計算機ハードウェア構成

表 2.3 設備現況と制御指令

設備現況 (材の状態を含む)	制御指令
<ul style="list-style-type: none"> <li>・設備</li> <li>  ・位置   ・正常/異常</li> <li>  ・材の有/無</li> <li>  ・作業中/無作業</li> <li>・材</li> <li>  ・位置</li> <li>  ・手入れ材/無手入れ材</li> <li>  ・材のモード 給材/集材 (手入れ材に対して)</li> <li>  ・材の行き先 (ゾーン1, ゾーン2, 直進)</li> <li>  など</li> </ul>	<ul style="list-style-type: none"> <li>・テーブルに対して</li> <li>  ・行き先 (ゾーン1, ゾーン2, 直進)</li> <li>・移載機に対して</li> <li>  ・行き先 (テーブルi/台車i/仮置き設備i)</li> <li>  ・下降指令</li> </ul>

## 2.6 ルール型制御方式の評価

### (1) 応答性

応答時間の目標値は、グラインダ処理効率に影響を与えないこと、材の流れのスピード等を考慮して、最大1 secと設定していた。これに対し、表2.4に示すように、マルチタスク下で、制御指令決定タスクの最大応答時間568 msec(平均は128 msec)の実測値を得て、目標値の達成を確認した。

表2.4 制御指令決定タスクの応答時間

	待ち時間 (a)	処理時間 (b)	応答時間 (a+b)
最大時間 (m sec)	29	539	568
平均時間 (m sec)	11	117	128

表2.5 推論プログラムの処理時間

ルール群名称	ルール数	処理時間/ルール群 (m sec)			処理時間/ルール (m sec)
		最大	最小	平均	
ルール群B	48	98	54	57	1.2
ルール群C	19	34	21	22	1.2
ルール群D	9	14	11	12	1.3
ルール群G	19	30	21	23	1.2
ルール群H	8	16	9	10	1.3

推論プログラムだけの処理時間は、表2.5に示すように、最大のルール群である材の投入タイミング制御ルール群でも、最大98 msec(平均57 msec)の実測結果であった。同一ルールに対する、処理時間のばらつきは、適用時点で設備現況および中間結論の数異なることによる。なお、1ルール当たりの平均時間は1.2 msecであった。

応答時間に占める推論機構の処理時間の割合は、制御指令決定タスクの実行開始から完了までの時間117 msecに対する1ルール群当たりの平均処理時間26 msecの割合と考えることができ、20%強であった(1ルール群とした理由は、各イベントに対し、ほとんど、1ルール群しか動かないようにルール群の振分けを行ったため)。

(2) 拡張性, 保守性

制御プログラムが段階的に開発できたことを示すために、マクロなルール数の時間推移を図2.12を使って説明する。

まず、正常時のルール群を固め、つぎに異常時のルール群を作るという段階的開発方針をとった。正常時のルール群はA, C, D, G, H(図2.9参照)であり、ルールの合計は約60であった。その後、設備の故障時、代替設備による自動運転の可能化を実現するため、ルール群E, F, I, Jを追加し、ルール数は約200になった。しかし、材の投入タイミング制御をしないとデッドロックが発生することが分かり、ルール群Bを加え、ルール数は251になった。以上が現地デバッグ前までのルール数の推移である。

現地デバッグでは、2回のルール修正を行い、現在順調に稼働している。その修正は、

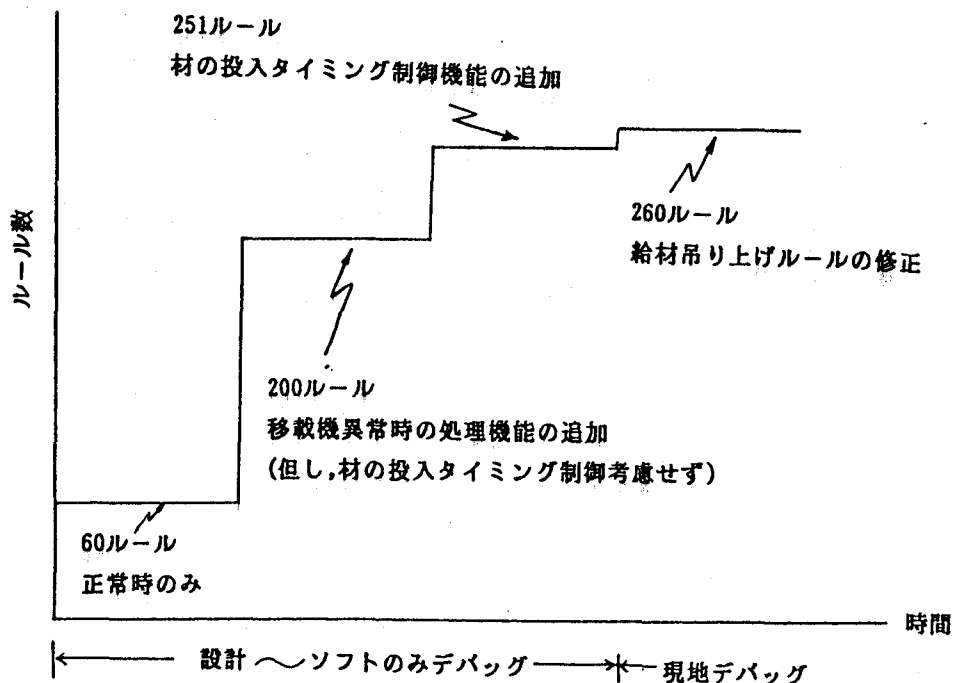


図2.12 ルール数の推移



運転状況を見てルールの不具合が判明したために実施した。その1つの給材吊り上げルール群Cの修正を説明する。修正前のルールでは、テーブル上の材に関し、テーブル7上に吊り上げる材があることのみを条件としていた。しかし、テーブル7に材がなくとも、その上流に吊り上げるべき材があれば、移載機に吊り上げ指示を先に出し、移載機をテーブル7の前で待機させる方が効率的と分かり、ルール修正した。この修正を含め、最終的には260ルールになった。これらの修正は、データのみの変更で済んだ。

以上述べたデータは本プラント建設当初のものであるが、その後、図2.8に示すゾーン1の増設が完了し、現状、総ルール数はユーザの手で528ルールまで拡張されている。この過程でシステム拡張時での本制御方式の柔軟性が、エンドユーザにより立証された。

以上は、ルール数のマクロな推移であり、各フェーズでは、マイクロに変動している。これは、2.4で提案したルールチェック方法によるもので、ルール数の約1割程のルールの抜けや不合理なルールを抽出できた。そのチェック方法では、条件項目そのものの抜け、あるルールに対する条件の抜けが最も多く抽出できたが、条件の組合せが全てを尽くしていないことの判明から、ルールの追加もかなりあった。

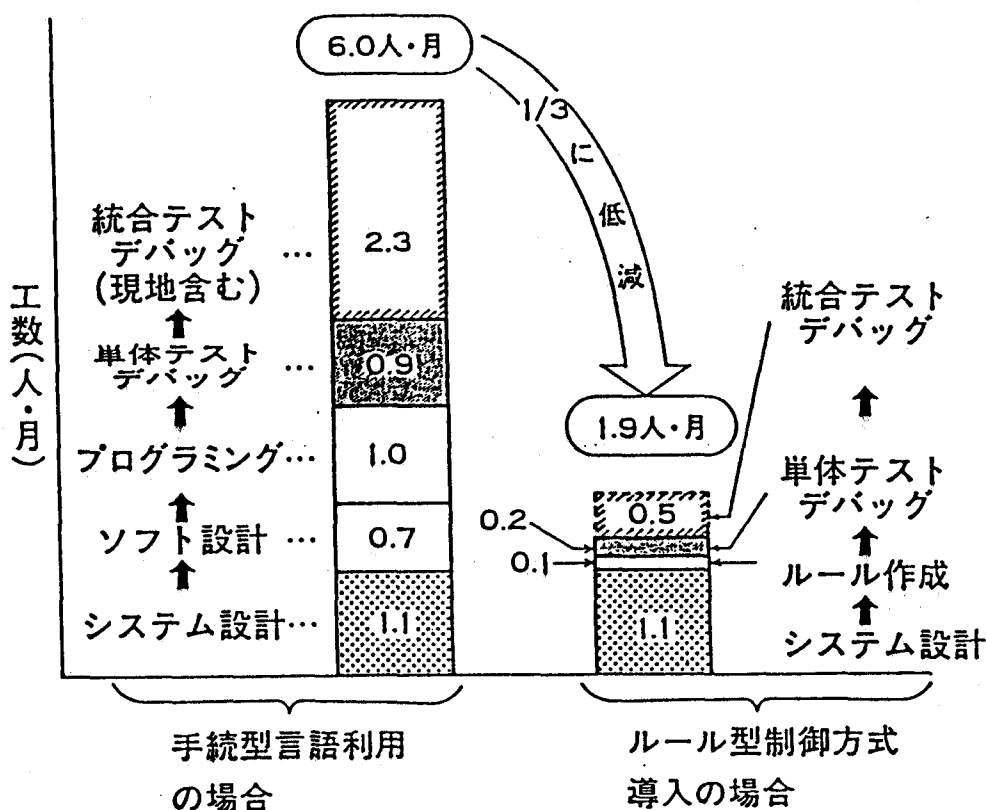


図2.13 ソフト開発工数の低減効果

一方、条件を設定して、推論機構を動かし、所望の結論が得られるかをシミュレートする方法も適用したが、ルールの文字列、単語の不統一のミスをいくつか発見できたのみであった。そのミスの発見が少なかった理由は、シミュレーションする前に、エディタのファインド機能(検索したい文字列を指定すると、その文字列のある箇所を順次示す機能)で大部分をチェック、修正したことによる。

### (3)ソフト開発工数

図2.13に示すように、ソフト開発工数は、従来の手続き型言語を利用した場合に比べ、1/3に低減できることが分かった。なお、従来のソフト開発工数の値は、過去の実績データに基づいた推定値である。この主な要因は、以下の項目と考える。

(i)システム設計段階で決定されるルールが従来のプログラムに対応する。そのため、従来必要であった、システム設計書をもとにした、テーブル設計、処理フロー設計、プログラミングが不要になる。

(ii)プログラム(ルール)がシステム設計段階で高品質化でき、しかも、ルールに問題がある場合でも、コンパイル、リンクなしに論理をデータ修正だけで対応できる。このため、最も工数の要する現地デバッグの工数が大幅に低減できた。

## 2.7 まとめ

第2章では、問題と解法の記述に知識工学を応用する運用制御問題を、流れ作業ラインの物流制御を対象に論じた。明らかにした事項は次の通りである。

(1)本物流制御は実時間制御であり、知識工学の実時間制御への応用では、プロセスとの簡易なインターフェース方式が必要である。これを実現するための方式として、トラッキングテーブルから推論機構のワークテーブルへの設備現況書き込み機能、推論機構で決定された制御指令の読み出し機能に関して、文字列、パラメータを与えるだけでプロセスと推論機構をつなげられるインターフェースルーチンを設ける方式が有効であることを実システムへの適用を通じて示した。

(2)ルール記述されたプログラムを人間の介入の許されない閉ループコントロールの中で動かさなければならない実時間制御では、稼働前に条件そのものの抜けやルールそのものの抜け等の論理自体のチェックが確実にできる方法が必要である。その実用的な方法として、人間が介入するが、排他の条件に着目し、条件と結論との相互関係の構造化を図る方法を提案し、その有効性を実システムへの適用を通じて確認した。

(3)ビレット精整ラインの物流制御への知識工学の適用を通じて、組合せの数が比較的に少ない問題では、問題と解法の両者をルールで記述しても、十分に実時間性を満足することが分かった。また、制御ソフトの開発工数は、従来の手続き型の場合に比べ、約1/3に低減できることが判った。

本ルール型制御方式は、流れ作業ライン制御以外に自動倉庫システムの制御に対しても適用されている<sup>22)</sup>。自動倉庫システムの場合には、スタッカクレーン、入出庫コンベア、昇降機の制御に適用しており、既に20件以上の実システムで稼働中で、制御ソフトの開発工数の低減(従来比1/3)、変更の容易性などの効果を確認している。

なお、制御対象によっては、本章で述べたシステムで実現した568 msecという応答よりも更に高いレスポンスを要求される場合がある。そのような極めて高い応答性を求められるものに対しては、ルール型制御方式をソフトウェアのみで実現すると、IF部と一致する条件データを判定するための処理時間が大きくなり、要求された応答性を満たせない場合が発生する。これを解決するには、その一致判定等のルール処理を専用ハードを用いて飛躍的に向上する必要がある。そこで、並列検索ハード技術を用いるIF部のAND条件演算ハードとルールを並列処理する並列推論システムの研究開発も行った<sup>24)25)</sup>。その結果、従来のソフト処理での処理時間を一桁低減することができることを確認した。

## 参考文献

- (1) 荒屋：ヒューリスティックスを利用した列車運転整理手法，電気学会論文誌C，Vol.103，No.11，pp249-256（昭和58年11月）
- (2) 松本，坂口：知識ベースに基づく電力系統復旧方式の決定方法，電気学会論文誌B，Vol.103，No.3，pp175-182（昭和58年3月）
- (3) 元田，山田：システムの機能に関する知識を用いた原子炉異常診断の試み，計測と制御，Vol.22，No.9，pp791-796（昭和58年9月）
- (4) 栗原，明石，天満，三留，井原：プロダクションシステム応用のプラント異常診断方式，電気学会全国大会，pp1714-1714（昭和58年4月）
- (5) 前田：ルールベースによる下水処理プロセス運用制御支援法，第22回SICE学術講演会予稿集，pp169-170（昭和58年7月）
- (6) J.MacDermott：R1:Rule-based Configurer of Computer Systems, Artificial Intelligence, Vol.19, No.1, pp39-88 (1982)
- (7) R.G.Smith: On the Development of Commercial Expert Systems, AI Magazine, Vol.5, No.3, pp61-74 (1984)
- (8) 渡辺：知識獲得支援システム，計測と制御，Vol.22，No.9，pp746-773（昭和58年9月）
- (9) 都島，田代，薦田，馬場，高倉：流れ作業ラインへのルール型制御方式の適用－製鉄所のピレット精整ライン制御への適用－，計測自動制御学会論文集，Vol.21，No.10，pp107-114（昭和60年10月）
- (10) 都島，馬場，高倉，田代：ルール型制御方式の実プロセス制御への適用，第23回SICE学術講演会予稿集，pp273-274（昭和59年7月）
- (11) 高倉，都島：FAシステムにおけるAIの応用，(株)トリケップス，pp193-203（昭和62年2月）
- (12) K.Baba, J.Yamasaki, T.Nakanishi, H.Kikugawa, T.Takahashi and T.Hujimoto: Development of Process Computer Control System for a New Billet Mill, Proc. of IECON '85, pp557-562 (1985)
- (13) 都島，田代，春名，薦田，松本：FA搬送制御方式の提案とそのルール記述の試み，昭和58年度電気学会東京支部大会予稿集，pp386-387（昭和58年11月）
- (14) 都島，薦田，田代：知識情報処理技術のFAへの応用，情報処理，Vol.25，No.4，pp378-385（昭和59年4月）
- (15) 田代，薦田，都島，松本：ルール型ソフトウェアシステムSCD (Station Coordinator)の開発，情報処理学会論文誌，Vol.27，No.5，pp552-561（昭和61年5月）
- (16) 田代，都島，薦田，松本：離散型システム用ルールベースコントローラ，第2回SICE知識工学シンポジウム，pp115-120（昭和59年3月）
- (17) N.Komoda, K.Kera and T.Kubo: An Autonomous, Decentralized Control System for Factory Automation, IEEE Computer, Vol.17, No.12, pp73-83 (1984)
- (18) T.Tashiro, N.Komoda, I.Tsushima and K.Matsumoto: Advanced Software for

Constraint Combinatorial Control—Rule-based Control Software for Factory Automation—, IEEE Int. Computer Conf. and Exhibition on Computer Aided Technologies Compint 85, pp132-137 (1985-9)

(19) T. Tashiro, N. Komoda, I. Tsushima and K. Matsumoto: Rule-based Control Software for Factory Automation—Its Rule Correctness Check Support and Response-Time Estimation, Proc. of the 4th IFAC/IFIP Symposium on Software for Computer Control SOCOCO'86, pp38-42 (1986-5)

(20) 薦田, 西山, 田代, 松本: ルールベースコントローラ用ルール入力画面エディタ, 情報処理学会第29回全国大会, 5N-2, pp1249-1250 (昭和59年9月)

(21) 田代, 前川, 薦田, 都島: ルール型制御システムSCD用ルール作成支援ソフトウェア, 処理学会第31回全国大会, 5M-1, pp981-982 (昭和60年9月)

(22) K. Okasaka and I. Tsushima: Automated Storage and Retrieval System(AS/RS) Using AI(Artificial Intelligence), Proc. of the 5th Int. Conf. on Automated Guided Vehicle System, pp1-18 (1987-10)

(23) 薦田, 都島, 田代: ルール型制御技術のFA物流システムへの適用, システムと制御, Vol.30, No.5, pp296-302 (昭和61年5月)

(24) 都島, 田代, 薦田: ルール処理の並列化可能個所の分析, 情報処理学会第32回全国大会, 6P-6, pp1543-1544 (昭和61年3月)

(25) 田代, 薦田, 都島: ルールの並列推論装置, 情報処理学会第32回全国大会, 6P-7, pp1545-1546 (昭和61年3月)

### 第3章 問題記述に知識工学を応用した 搬送車群運行制御方式

## 第3章 問題記述に知識工学を応用した 搬送車群運行制御方式

### 3.1 まえがき

消費者要求はますます多様化してきており、それに伴い、多種少量生産化の度合いが高まっている。そのため、生産システムでは物の流れが複雑化し、運用方式、レイアウト、設備は頻繁に変更される。これに対応するため、搬送設備は、従来のコンベアから複雑な軌道の実現、変更の容易な無人搬送車へと移行しつつある。

しかしながら、複数台の搬送車を効率よく運行させることを目的に、搬送要求に搬送車を割り付ける運行制御ソフトの方は変更が困難なままで、しかも効率的な運行が実現されていない。

変更が困難な理由として、従来、運行制御ソフトが手続き型で汎用のプログラミング言語(FORTRANなど)で開発されているため、一度書かれたプログラムから制御論理を読みだすことが作成者でも困難なこと、さらに重要なことは、運行制御問題の構造化が不適切のため、変更の頻発する運行条件の変更が、この条件に基づく搬送車割付アルゴリズムの変更にもまで及ぶことの2点が挙げられる。

効率的な運行が実現されていない理由は、従来の搬送車割付アルゴリズム<sup>1)</sup>では、搬送候補に対して、作業割付待ち搬送車のうち、最も早く到着できる搬送車を割り付けるため、現在作業中の搬送車とその作業を完了後に処理した方が早く処理できる場合を考慮できないことにある。

最近、変更に対する柔軟性の向上を図るため、プロダクションシステムに代表される知識工学応用のアプローチが現われている。知識工学<sup>2)</sup>は、各種条件に合う項目を選択するという条件判定的な問題に対して適したものであり、組合せ最適化的な問題を含む運行制御問題にそのまま適用することはできない。

そこで、本章では、問題記述に知識工学を応用した搬送車群運行制御方式<sup>3)</sup>について論じる。まず、運行制御問題を条件判定的な問題と組合せ最適化的な問題に分離する構造を提案する。その条件判定的な問題では、搬送車により、ステーションへパレットを搬送できるか否かの条件、搬送車の台数が限られているため、どの搬送要求から優先的に搬送車を割り付けるべきかの条件などが指定される。それらの条件は変更が頻発するため、変更の容易なルールにより記述する。一方、分離された組合せ最適化的な問題は、搬送要求に対する搬送車割付の問題であり、これに対し、高効率で標準化された搬送車割付アルゴリズムを提案する。また、以上の考え方に基づく運行制御システムについて述べ、提案方式により変更が容易に実現できること、搬送効率が向上できることを例をもとに説明する。

### 3.2 運行制御ソフトへの要求

はじめに、運行制御ソフトの制御対象に関し、図3.1に示す組立ショップのレイアウト例により説明する。前工程である加工ショップから1製品の組み立てに必要な部品を搭載したパレットが投入コンベアに投入される。投入されたパレットは、製品の種類毎に決

められた組立ステーションへ順々に搬送車により運ばれる(この経路を工程経路と呼ぶ)。組立ステーションには搬送車とのパレットの受け渡しのため、搬入バッファ、搬出バッファが設置されている。全工程経路をパレットが経ると、製品が完成し、それは後工程である検査ショップへの払出しコンベアに搬送される。なお、搬送車の軌道は一方通行であり、その軌道上に作業指示待ちの搬送車が待機するための待機点が設置されている。

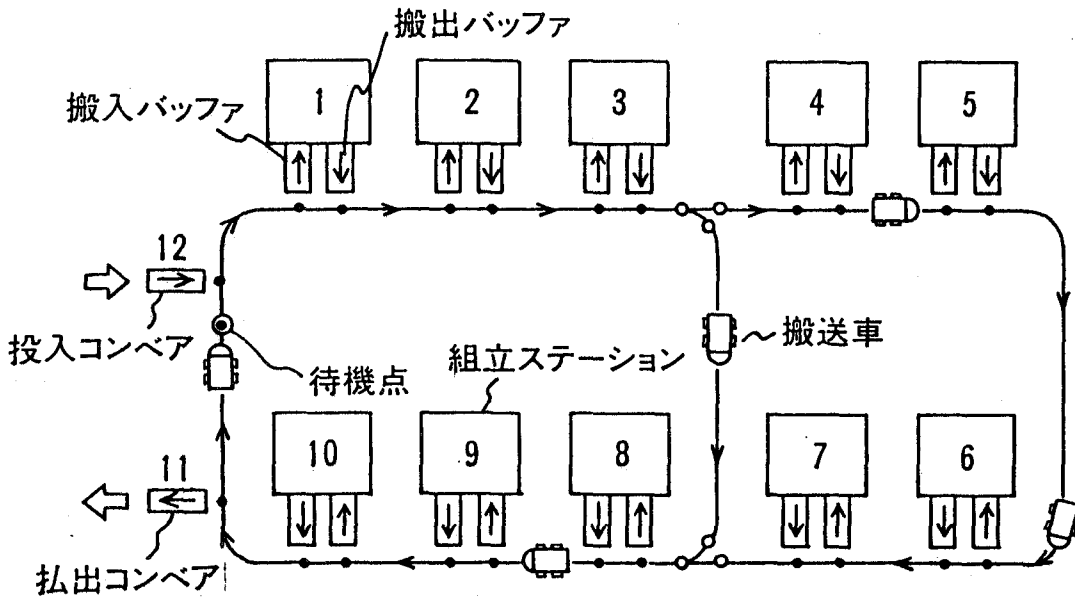


図 3.1 組立ショップのレイアウト例

搬送は生産のための補助機能である。そのため、生産システムからの運行制御に対する要求が運行制御の目的となる。以下に、その要求(運行制御の目的)を示す。

(1) マクロには、作業ステーション全体の稼働率の向上

具体的には、搬入バッファ上のパレットが切れることにより、あるいは搬出バッファ上にパレットが満杯となり、組立て機からパレットを搬出できないことにより作業ステーションが無作業になることの防止である。

(2) ミクロには、生産状況の変化に即した、個々の作業ステーション、個々の製品に対する優先的搬送の実現

例として、故障回復後の作業ステーション、納期条件の厳しい製品などに対する優先的搬送が挙げられる。

(3) 搬送車の無駄な走行時間の低減

上述の運行制御の目的(1),(2)を短時間で達成するため、(3)の実現が図られる必要がある。

また、生産システムから、以下の変更に対し、運行制御ソフトが迅速に修正可能なこと



が要求されている。

(i)既存タイプの設備の容量(搬送車数, 作業ステーション数, 搬入出バッファ容量)の変更

(ii)軌道レイアウトの変更

(iii)工程経路の変更

(iv)新タイプの作業ステーションの追加

(v)運行制御方法の変更

上記の(i)~(iii)は, 運行制御ソフトのパラメータの値の変更だけで対応できる。一方, (iv), (v)では, 運行制御ソフトの内容そのものに変更が及び, パラメータ値の変更では対応できない。

(iv)ではステーションに対し, 搬送車によりパレットを搬入あるいは搬出できるか否かの条件(搬送可否条件と呼ぶ)の変更が発生する。搬送可否条件の具体例としては, 搬入バッファが空いているか否か, 搬出バッファの先頭にパレットが出ているか否か, ステーション内の周辺設備がパレットの搬入出の際に, パレットに衝突しない位置に退避しているか否かなどが挙げられる。このような条件は, ステーション内の設備構成により異なるため, 新たな設備構成のステーションが追加されるたびに, 新たな内容の条件が追加される。

(v)は搬送車がどの搬送要求から優先的にサービスすべきかを規定する条件(搬送優先条件と呼ぶ)の変更が大部分である。搬送優先条件の具体例としては, 無作業になるまでの余裕時間の少ないステーションへの優先搬送, 生産量の落ちている故障回復後のステーションへの優先搬送などが挙げられる。このような条件は, 前述の運行制御の目的(1), (2)の実現に大きく影響する。生産技術者は日々, 新たな搬送優先条件を考案し, それを運行制御ソフトに迅速に組み入れたいという要求を持つ。

### 3.3 運行制御機能

本節では, 3.2で述べた運行制御の目的, 変更に対する柔軟性の向上を実現するため, どの運行制御機能を考察すべきかについて述べる。

運行制御機能として, つぎの4つが考えられる。

#### (1)搬送要求への搬送車割付機能

ある時点において, 複数の搬送要求が発生しているとき, どれを先にサービスすべきか, また, それらに対して, 各所にいる空き搬送車のどれを割り付けるかを決定する機能である。

#### (2)搬送車の待機点選択機能

割り付けるべき搬送要求がない場合, 作業終了した搬送車に対して, 作業指示を待つための待機点を選択する機能である。

#### (3)搬送ルートを選択機能

最短時間で到着するルートを選択する機能である。

#### (4)合流点での搬送車進行制御機能

合流点に複数の搬送車が進入しようとする際、その進入順序を制御する機能である。

搬送要求への搬送車割付機能は、その機能から分かるように、運行制御の目的に大きく係わる。一方、他の3つの機能については、以下に述べるように、Factory Automation (FA)における搬送車導入システムの特徴から、その実現方式の違いによって運行制御の目的の実現を左右するものではない。

FAにおける搬送車導入システムの大部分は、以下の特徴を持つ。

- (i)搬送車の移動範囲はあまり広くない。
- (ii)搬送車数は平均5, 6台である。
- (iii)軌道レイアウトはあまり複雑ではない。

待機点選択機能は、広域の搬送システムでは効果的であるが、特徴(i)により、FAでは近くの待機点を選択する方式で充分である。ルート選択機能については各種の方式が検討されているが、FAでは、特徴(ii)により混雑度を考慮しなければならないほどの高密度運行にならないこと、特徴(iii)により代替ルートに比べてはるかに適切なルートが一意的に決まってしまうことから、あらかじめ、2点間のルートを一意的に与えている。合流点の進行制御機能についても、ルート選択に関する特徴(iii)の議論から、先入先出方式で充分である。

一方、変更に対する柔軟性に関し、問題となる搬送可否条件、搬送優先条件の変更に係わる制御機能は、搬送要求への搬送車割付機能だけである。

したがって、運行制御の目的、変更に対する柔軟性の向上の実現は、搬送要求への搬送車割付方式を対象に議論すればよい。

### 3.4 搬送要求への搬送車割付方式

#### 3.4.1 柔軟性の実現方式

柔軟な搬送車割付ソフトを構築するには、3.2で述べた、内容そのものの変更の生じる搬送可否条件、搬送優先条件を迅速に変更できる方法が必要である。また、それらの変更が他の部分に影響を与えない搬送車割付問題の構造化、言いかえれば、搬送要求に適切な搬送車を割り付けるアルゴリズムをそれらの条件から分離する構造化が必要である。そのように分離できれば一度開発した割付アルゴリズムは標準化できたことになり、上述した条件に伴う改造は不要となる。

まず、割付アルゴリズムの標準化を実現するための搬送車割付問題の構造化について述べる。

アルゴリズムの標準化のための条件は、入力情報が規定された項目だけで定義できることである。割付アルゴリズムへの入力情報は、搬送要求と搬送車状況である。割付の際に必要な搬送車状況は、搬送車の作業状況(作業指示待ち、実走行中など)、現在位置であるが、これらは搬送車固有のものであり、規定された項目で定義できる。しかしながら、搬送要求の方は新しい内容の条件が追加され、標準化のための条件を満たさない。その条件を満足させるために、多様な搬送要求を{(搬送元)、(搬送先)、(搬送優先度)}という搬送一般に共通な搬送要求に加工する。この加工された搬送要求を入力情報とすれば、その

割付アルゴリズムは標準化可能となる。

さらに、搬送可否条件の変更は新しいタイプの作業ステーションの追加時に、搬送優先条件の変更は生産状況に応じて発生するというように独立に発生することから、それぞれの変更がお互いに影響を与えないように搬送要求の加工処理を2段階に分ける。第1段階は、多様な搬送要求から搬送可否条件をチェックし、搬送可能な搬送要求(搬送候補と呼ぶ)を抽出する。この結果の形式も、搬送一般に共通な{(搬送元),(搬送先)}とする。第2段階は、搬送優先条件に基づき、搬送候補に優先度を付け、前述の最終的な搬送要求へ加工する。

つぎに、搬送可否条件、搬送優先条件を迅速に変更できる方法について述べる。搬送の可否は、作業ステーション内の各設備の状態に応じて決定される。搬送優先度は、各ステーションの作業進捗状況、物の属性(納期など)といったシステム状態の組合せによって決定される。すなわち、システムがどのような状態にあるかを判別し、その状態に応じて適切な結論を決定するという条件判別型の論理である。したがって、それらの論理の記述は、プロダクションシステムにおけるIF(条件),THEN(結論)型のルールによる記述が適しており、上述の条件をルール記述できるようにする。

搬送可否条件を考慮して、搬送候補を抽出するルールの一部を図3.2に示す。ルールNo.1では、搬送元になりうる搬出バッファを求める。その条件は、搬出バッファが正常で自動モードであり、そのバッファの先頭にパレットがあり、搬送指示が未割付であると

RULE No.1

IF(一般作業ステーション<x>の搬出バッファは正常で自動モード)  
IF(一般作業ステーション<x>の搬出バッファの先頭にパレットあり)  
IF(一般作業ステーション<x>の搬出バッファは搬送指示未割付け)  
THEN(作業ステーション<x>の搬出バッファは搬出候補)  
:

RULE No.5

IF(一般作業ステーション<x>の搬入バッファは正常で自動モード)  
IF(一般作業ステーション<x>の搬入バッファに受入エリアあり)  
THEN(作業ステーション<x>の搬入バッファは搬入候補)  
:

RULE No.12

IF(作業ステーション<x>の搬出バッファは搬出候補)  
IF(作業ステーション<y>の搬入バッファは搬入候補)  
IF(作業ステーション<x>の搬出バッファの先頭パレットは製品<z>)  
IF(製品<z>の工程経路に作業ステーション<x>から<y>あり)  
THEN(作業ステーション<x>から<y>は搬送候補)  
:

図3.2 搬送候補抽出ルール

いう条件である。ルールNo.5では、搬送先になりうる搬入バッファを、正常で自動モード、かつ受入れエリアありという条件により求める。ルールNo.12では、上記のルールの結論と工程経路データを条件として、搬送候補を抽出する。この結論が{(搬送元),(搬送先)}に相当する。

搬送要求に対する搬送優先条件を考慮し、先のルールで求めた搬送候補に搬送優先度を付けるルールの一部を図3.3に示す。このルールでは、まず、搬入バッファ上のパレットが切れ、作業ステーションが無作業になるまでのパレット搬入余裕時間を、そのバッファ上の製品の処理時間をもとに求める(ルールNo.1)。また、搬出バッファ上にパレットが満杯となり、組立機からパレットを搬出できないことにより、作業ステーションが無作業になるまでのパレット搬出余裕時間を、搬入バッファ上の製品の処理時間などをもとに求める(ルールNo.7)。搬送候補に対する搬送余裕時間は、その搬送元の搬出バッファからの搬出余裕時間と、その搬送先の搬入バッファへの搬入余裕時間との小さい方で決まる(ルールNo.15)。この搬送余裕時間が各優先度に対応した搬送余裕時間区間定数と比較され、搬送候補に搬送優先度が付けられる(ルールNo.18)。

RULE No.1

```
IF(作業ステーション<x>の搬入バッファ上にパレット2個)
IF(作業ステーション<y>の搬入バッファ上の第1番目パレットの処理時間は<y>分)
IF(作業ステーション<x>の搬入バッファ上の第2番目パレットの処理時間は<z>分)
THEN<plus /y,z,t/>(作業ステーション<x>の搬入バッファへの搬入余裕時間は<t>分)
:
```

RULE No.7

```
IF(作業ステーション<x>の搬出バッファ上にパレット3個)
IF(作業ステーション<x>の搬入バッファ上にパレット1個)
IF(作業ステーション<x>の搬入バッファ上の第1番目パレットの処理時間は<y>分)
THEN(作業ステーション<x>の搬出バッファからの搬出余裕時間は<y>分)
:
```

RULE No.15

```
IF(作業ステーション<x>から<y>は搬送候補)
IF(作業ステーション<x>の搬出バッファからの搬出余裕時間は<z>分)
IF(作業ステーション<y>の搬入バッファへの搬入余裕時間は<s>分)
THEN<min /z,s,t/>(作業ステーション<x>から<y>の搬送候補の搬送余裕時間は<t>)
:
```

RULE No.18

```
IF(作業ステーション<x>から<y>の搬送候補の搬送余裕時間は<z>)
IF(搬送余裕時間<z>.LE.<5>分)
THEN(作業ステーション<x>から<y>の搬送候補は優先度<1>)
:
```

図3.3 搬送優先度付けルール

### 3.4.2 搬送車割付アルゴリズム

優先度付けされた搬送候補{(搬送元), (搬送先), (搬送優先度)}が与えられれば, 搬送車割付アルゴリズムが標準化可能であることは, 3.4.1で述べた。本項では, 搬送効率の向上を実現する搬送車割付アルゴリズムを提案する。

従来アルゴリズムでは, まず, 搬送要求への割付候補とする搬送車として, 作業完了して, 次の搬送指示を待っている搬送車群を抽出する。つぎに, 割付候補として抽出された搬送車群の中から搬送要求を最も早く処理できる搬送車を選択し, それを該当の搬送要求に対して割り付ける。この従来アルゴリズムの問題点は, 搬送候補にならない作業中の搬送車がその作業を完了した後に, 該当の搬送要求を処理したほうが早く処理できる場合を考慮できないことにある。

そこで, 従来アルゴリズムの問題点を解決するため, 作業割付待ち搬送車だけでなく, 現在作業中の搬送車も考慮して割り付けるべき搬送車を決定するアルゴリズムとした。ただし, 最も早く到着する搬送車が作業割付待ちのものであれば, 作業を割り付けるが, 作業中のものであればこの時点では割り付けない。というのは, 近い将来, 新たな搬送優先度の高い搬送要求が発生するかもしれないためであり, その搬送車が作業割付待ちになった時点, 搬送要求が発生した時点で, 搬送車への作業割付を毎回実施し, 新たな状況への対応を行う。

提案アルゴリズムのフローチャートを図3.4に示す。まず, 各搬送車の, 各搬送候補

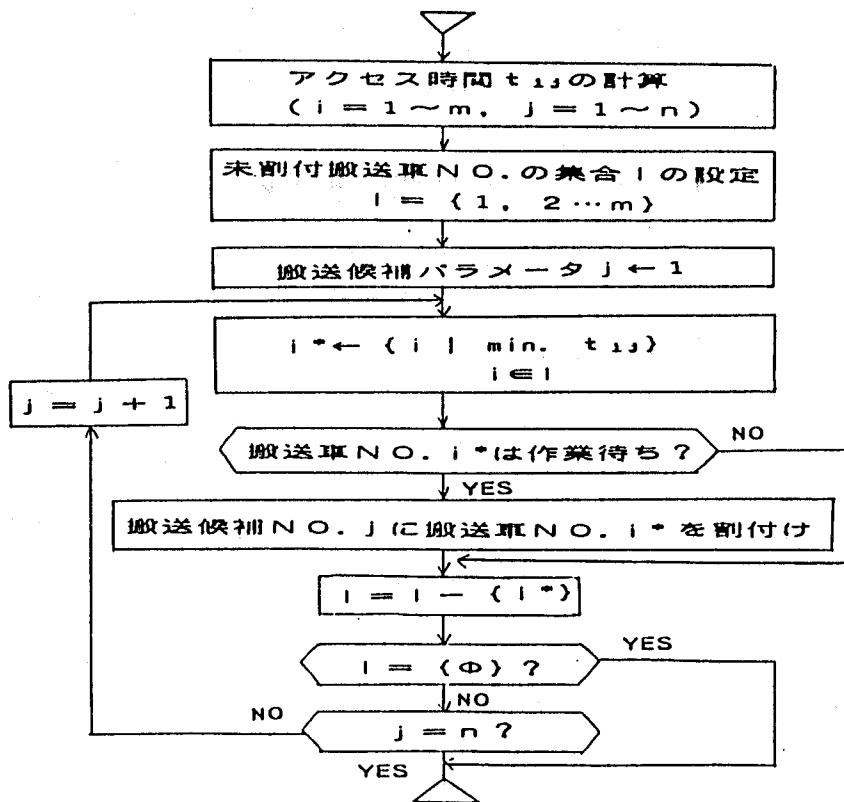


図3.4 提案アルゴリズム

へのアクセス時間 $t_{1j}$ を求める。 $t_{1j}$ は、搬送車No.  $i$  が搬送候補No.  $j$  の搬送元へ到着するのに要する時間を示し、搬送車の現在位置、搬送車の作業状況(作業待ち、パレット搬送中、空走行中、パレット積み卸し中など)、軌道レイアウト、各軌道ブロックの走行にかかる時間、単位積み卸し時間、工程経路データ、搬送候補データから求められる。

つぎに、 $t_{1j}$ をもとに、搬送優先度の高い搬送候補から順に、搬送車を割り付ける。なお、搬送候補No.  $j$  は、 $j$  の値が若いほど搬送優先度が高いとする。まず、各搬送車No. の集合を  $I = \{1, 2, \dots, m\}$  とする。つぎに、搬送候補を示すパラメータ  $j$  に 1 を設定する。以上は、初期値の設定処理である。つぎに、搬送車No. の集合  $I$  のなかで、搬送候補No.  $j$  に最も早く到着できる搬送車No.  $i^*$  を求める ( $i^* = \{i \mid \min_{i \in I} t_{1j}\}$ )。つぎに、搬送車No.  $i^*$  が作業待ちか否かをチェックする。もし、作業待ちであれば、搬送候補No.  $j$  に搬送車No.  $i^*$  を割り付ける。 $i^*$  が作業中であれば、割り付けを行わない。以上で、搬送車No.  $i^*$  の検討は終わったので、 $I$  から  $i^*$  を除く。 $I$  が空集合  $\phi$  に等しい、あるいは全ての搬送要求に対する検討が完了した時点で処理を終了する。

### 3.5 運行制御システム

#### 3.5.1 入力情報の記述形式

##### (1) 各種パラメータ値の記述形式

パラメータの値として入力する情報の主なものに、作業ステーション情報、軌道レイアウト情報、工程経路情報がある。

表3.1は、作業ステーション情報定義方式を示す。この表をもとに、作業ステーショ

表3.1 作業ステーション情報定義方式

作業ステーション NO.	作業ステーション の種別	搬入バッファ		搬入バッファ	
		ノードNO.	容量	ノードNO.	容量
1	一般	1	3	2	3
2	一般	3	3	4	3
:	:	:	:	:	:
:	:	:	:	:	:
11	払出専用	30	10	-	-
12	投入専用	-	-	32	10

ンNo., 作業ステーションの種別, そのステーションに設置されている搬入, 搬出バッファの容量などを定義する。種別には, 搬入, 搬出バッファとも有する一般作業ステーション, 投入専用ステーション(図3.1の投入コンベアに相当), 払出し専用ステーション(図3.1の払出しコンベアに相当)の区別を与える。

表3.2 軌道レイアウト情報定義方式

ノードNO.	ノード種別	次ノード	分岐情報
1	搬入ノード	2	
:	:	:	
:	:	:	
7	分岐開始ノード	8	8,9,10,11,12,13,14, 15,16,17,18,19
:	:	:	
:	:	:	

表3.3 工程経路情報定義方式

品目	搬出ステーションNO. (FROM)	搬入ステーションNO. (TO)
A	1 2	1
	1	3
	3	5
	5	7
	7	1 1
:	:	:
:	:	:
D	1 2	1
	1	3
	3	9
	9	1 1
	:	:
:	:	:

表3.2は、軌道レイアウト情報定義方式を示す。軌道定義は、各ノードから次のノードという個々のアークにより定義する。ノード種別には、搬送車がパレットを積み卸しするために停止する搬入ノード、搬出ノード、分岐点、合流点に相当するノード、作業指示を待つための待機点ノードのいずれかを定義する。分岐情報には、その分岐アークに行く搬送車の行先ノードを定義する。

表3.3は、工程経路情報定義方式を示す。

### (2) ルールの記述形式

図3.2、図3.3に示したように、制御論理は、日本語により、IF(条件),THEN(中間結論あるいは最終結論(制御指令))のルールで与える。IF,THENルールは、ルールの種別を宣言するTYPE部と、条件を記述するIF部と、その条件を満たした場合の結論を記述するTHEN部より構成される。IF部, THEN部はそれぞれ複数個の条件, 結論を記述できる。IF部の複数個の条件は、AND条件として取り扱う。条件, 結論を記述する文字列は、“(”, “)”で囲われ、固定部とパラメータ部よりなる。固定部は、条件や結論の内容を表わし、パラメータ部は“<”, “)”でくくり、固定部と区別する。パラメータ部のなかには、四則演算, 簡単な関数を定義できる。なお、ルールエディタについては、2.3.2で述べたので、ここでは省略する。

### 3.5.2 構成

提案する運行制御ソフトウェアシステムの構成を図3.5に示す。搬送候補の抽出部、搬送候補への搬送優先度付け部は、それぞれ、第2章の2.3で説明したルール型核ソフトSCDを利用して搬送要求を加工する。加工された搬送要求{(搬送元), (搬送先), (搬送優先度)}に対して、搬送車割付アルゴリズムにより、適切な搬送車が割り付けられる。ルールおよび各種テーブルの内容は、エディタにより簡単に確認, 変更ができる。

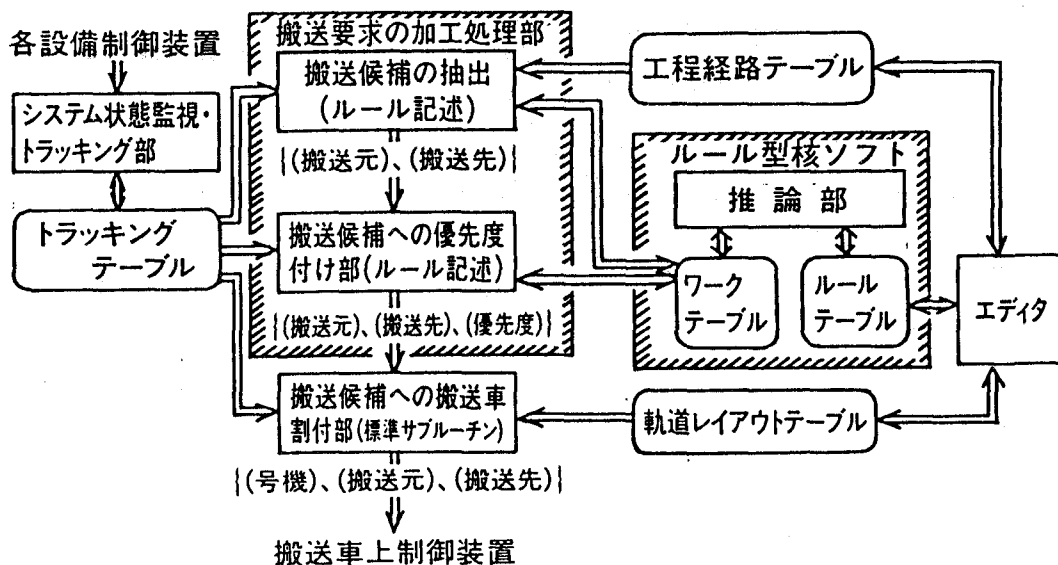


図3.5 運行制御ソフトウェアシステム



### 3.6 搬送車群運行制御シミュレータ

搬送車はコンベアに比べ、運行制御やレイアウトの実現の仕方に自由度があるため、実現方式を十分に検討しないと搬送の目的の未達成や搬送車の過剰投入といった状況が発生する。そのため、搬送車導入のFAシステムの計画に際し、各システムに即した搬送車群運行制御方式、軌道レイアウト、必要な搬送車台数を短時間で決定できるシミュレータが望まれる。また、生産設備の増設等により、運行制御やレイアウトの変更が発生する際にも、上述のシミュレータが要求される。

そこで、以上で述べた運行制御方式を組み込んだシミュレータを開発した。

#### 3.6.1 入力項目と入力形式

シミュレータへの入力項目は、以下の6つである。

- (1)ステーション情報
- (2)工程経路情報
- (3)軌道レイアウト
- (4)搬送車情報(台数, 速度等)
- (5)製品発生情報
- (6)搬送可否条件
- (7)搬送優先条件

上記の(1), (2), (3), (6), (7)は、3.5に述べた運行制御システムの入力情報の記述形式と同じである。(4), (5)はシミュレーション固有の入力情報である。図3.6は製品発生情報定義画面を示す。この画面をもとに、投入ステーション番号とそのステーションから投入される製品のコードを投入順に定義する。

製品発生情報

投入ステーション	製品コード発生順
12	001 002 003 004 005 006 007

削除PF7 上画面PF9 終了PF11  
挿入PF8 下画面PF10 取消PF12

図3.6 製品発生情報定義画面

#### 3.6.2 出力項目と出力形式

出力項目として、改善を示唆できる搬送車稼動状況を出力する。

搬送車に関しては、図3.7に示すように、パレット積み卸し中とパレットを積んで走行中の状態以外の無作業状態として、空で走行している状態と待機点で搬送指示を待っている状態を設け、それぞれの状態の占める割合が出力される。なお、搬送車の詳細な動きをトレースする機能も持っている。

### 搬送車稼働状況

搬送車台数[ 5] 作業ステーション数[12]

状態 ステーションNO	積卸し(%)	実走行(%)	空走行(%)	作業待ち(%)
1	14.1	31.3	40.0	14.6
2	14.3	31.0	41.3	13.4
3	13.0	34.4	42.2	10.4
4	11.9	33.3	43.8	11.0
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
平均	13.5	32.2	42.0	12.3

ステーション稼働状況PF8 上画面PF9 終了PF11  
下画面PF10

図3.7 搬送車稼働状況出力画面

ステーションに関しては、図3.8に示すように、組立作業状態以外の無作業状態として、パレットが搬入されないために組立作業ができない無作業状態(パレット搬入待ち)と、搬出バッファが満杯で組立機からパレットを卸せずに組立作業が出来ない無作業状態(パ

### 作業ステーション稼働状況

生産量[ 103] 搬送車台数[ 5] 作業ステーション数[10]

状態 ステーションNO	稼働(%)	パレット搬入待ち(%)	パレット搬出待ち(%)
1	90	8	2
2	99	1	0
3	89	7	4
4	83	13	4
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮

搬送車稼働状況PF8 上画面PF9 終了PF11  
下画面PF10

図3.8 ステーション稼働状況出力画面

レット搬出待ち)を設け、それぞれの状態の占める割合が出力される。また、組立ショップで生産された製品数も出力される(図3.8参照)。

計画者は、以上に述べた出力項目の結果をもとに、必要ならば運行制御方式、搬送車台数などを変更し、シミュレーションを繰り返し、適切な計画を得る。

### 3.6.3 構成

シミュレータの全体構成を図3.9に示す。

#### (1)入出力部

前述の入出力のための各種エディタを設けている。搬送可否条件、搬送優先条件は、ルールエディタを用いて定義される。

#### (2)運行制御指令決定部

本決定部は、3.3で述べた4つの運行制御機能を実現する。待機点の選択、ルートを選択、合流点の進行制御に対しては、標準モジュールが組み込まれている。一方、搬送要求への搬送車割付けについては、推論機構が、ルール記述された搬送可否条件、搬送優先条件とトラッキング現況をもとに、優先度付けされた搬送候補を抽出する。これらの搬送候補とトラッキング現況をもとに、シミュレータに組み込まれた搬送車割付モジュール(標準化モジュール)が適切な搬送指示を決定する。

#### (3)走行、積降し実行部

この処理部は、運行制御指令決定部から与えられた搬送指示をもとに搬送車システムのモデルを変更し、次のイベントを全体管理部に渡す。閉塞制御のモデル化の方法として、

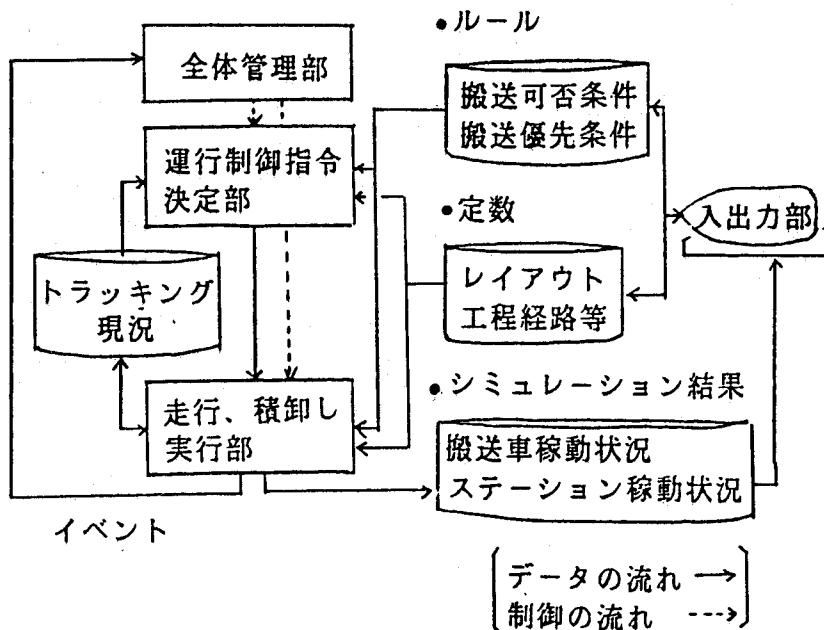


図3.9 シミュレータの全体構成

本シミュレータでは、閉塞エリアにあらかじめ定義される台数の搬送車が入れることにし、次の行き先ノード間に搬送車が入れるか否かを識別する方法を採っている。次の行き先ノード間に搬送車が満杯になっている場合および行き先ノードで搬送車がパレット積降しのために停止している場合には、それらの状態が解消されるイベントが発生するまで搬送車を動かさない。この方法により、搬送車1台毎に閉塞エリアを設ける方法に比べて、シミュレーションの処理時間を大幅に低減できた。

(4) 全体管理部

全体管理部は、イベントを解析し、運行制御指令決定部を起動する。

3.7 適用例

本節では、図3.1に示す製品組立ショップの搬送システムとプリント基板組立ショップに提案方式を適用した例について述べる。

3.7.1 製品組立ショップの搬送システムへの適用

まず、提案方式により条件の追加が容易に実施できることを示すため、特急処理条件の追加について述べる。つぎに、提案した搬送車割付アルゴリズムの効果について述べる。

(1) 制御ルール

初期設計段階のルール数は約80である。そのルール群の構成を図3.10に示す。

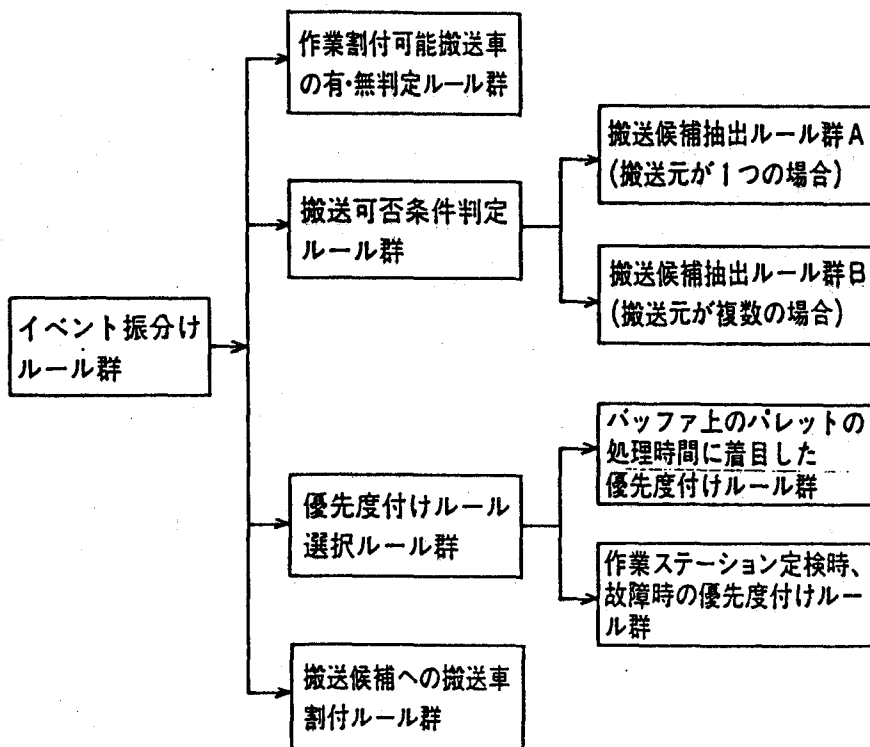


図3.10 ルール群の全体構成

イベント振分けルール群は、搬送車の作業完了、作業ステーションの状態変化などのイベントの発生に従って、どのルール群を起動するかを選択を行う。作業ステーションの状態変化時には、搬送要求に関するルール群の処理の前に、作業割付可能な搬送車があるかないかの判定ルール群を起動する。そのような搬送車がなければ、以降の処理は不要と結論する。あれば、搬送要求に関するルール群を起動する。

搬送可否条件判定ルール群は、搬送先が一意的に決定している搬送要求に対しては、3.4で説明した搬送候補抽出ルール群Aを起動する。複数の搬送先から一つを選択できる搬送要求に対しては、搬送候補抽出ルール群Bを起動する。ルール群Bでは、対象製品の作業時間が短くて済む搬送先作業ステーションを選択するロジックが組み込まれている。

搬送候補への優先度付けルール群は2種類ある。1つは、3.4で説明した、バッファ上のパレットの処理時間に着目した優先度付けルール群である。このルール群では、作業ステーション全体の稼働率が向上するように、優先度付けがなされる。他の1つは、ある作業ステーションの定検時、故障時に適用されるルール群である。そのような状況では、そのステーションで生産する製品量が大幅に低下するため、同種の稼働ステーションの稼働率を他より優先する必要がある、これを実現するのが、作業ステーション定検時、故障時の優先度付けルール群である。優先度付けルール選択ルール群は、以上の2種類のうち、どちらかを選択するルール群である。

## (2) 特急処理の追加

緊急の注文が入り、特定の製品を特急で生産する特急処理機能の追加を考える。

まず、搬送候補が特急処理製品の場合、その搬送優先度を最大にした(優先度←1)。また、通常処理の製品が特急処理を妨げないようにするため、特急製品を処理する作業ステーションからの搬送候補の搬送優先度を2番目に高いものにした(優先度←2)。さらに、搬送候補が通常処理製品で、それが現在、通常処理製品だけを処理する作業ステーションにあり、次工程が特急処理の作業ステーションの場合、その搬送候補の優先度を最低とした(優先度←8)。

通常処理製品のうち、特急製品の処理を妨げないものは、初期設計段階のルールがそのまま使用できる。ただし、優先度は、特急処理が入ったため、2ランクずつ下げ、1～5を3～7にした。

上記の特急処理は図3.11に示すように、初期設計段階でのバッファ上のパレットの処理時間に着目した優先度付けルールに、新規に5つのルールを追加し、既作成ルールの条件の一部と優先度を修正するだけで対応できた。

図3.12は、特急処理を考慮した場合と考慮しなかった場合との特急製品の投入から払出しまでの経過を示したものである。この例では、製品Dが3個分、特急製品として、ステーション1, 3, 9で処理される。特急処理の考慮で、通常2時間で処理されたものが、1時間20分で処理できたことがわかり、特急処理ルールの追加の効果が出ていることがわかる。

表3.4は、作業ステーションの稼働率結果を示したものである。特急処理を考慮すると、考慮しない場合に比べ、作業ステーションの平均稼働率は低くなる。これは、ステー

ション全体の稼働率の向上を犠牲にして、特急処理ステーションへのサービスを最優先したためである。大幅に稼働率が低下している通常処理ステーションがあるが、これは特急処理ステーション行きのパレットの搬出を控えたためである。

#### 5個の新しいルールの追加

↓

```

RULE No.1'
IF(作業ステーション<x>から<y>は搬送候補)
IF(作業ステーション<x>の搬出バッファの先頭パレットは製品<z>)
IF(製品<z>は特急処理要求あり)
THEN(作業ステーション<x>から<y>の搬送候補は優先度<1>)
RULE No.2'
IF(作業ステーション<x>から<y>は搬送候補)
IF(作業ステーション<x>の搬出バッファの先頭パレットは製品<z>)
IF(製品<z>は特急処理要求なし)
THEN(作業ステーション<x>から<y>の搬送候補は通常処理製品)
RULE No.3'
IF(作業ステーション<x>から<y>の搬送候補は通常処理製品)
IF(作業ステーション<x>は特急処理ステーション)
THEN(作業ステーション<x>から<y>の搬送候補は優先度<2>)
RULE No.4'
IF(作業ステーション<x>から<y>の搬送候補は通常処理製品)
IF(作業ステーション<x>は通常処理ステーション)
IF(作業ステーション<y>は特急処理ステーション)
THEN(作業ステーション<x>から<y>の搬送候補は優先度<8>)
RULE No.5'
IF(作業ステーション<x>から<y>の搬送候補は通常処理製品)
IF(作業ステーション<x>は通常処理ステーション)
IF(作業ステーション<y>は通常処理ステーション)
THEN(作業ステーション<x>から<y>の搬送候補は通常優先度付け)

```

⋮

#### 1個の条件の修正

↓

```

RULE No.15
IF(作業ステーション<x>から<y>の搬送候補は通常優先度付け)
IF(作業ステーション<x>の搬出バッファからの搬出余裕時間は<z>分)
IF(作業ステーション<y>の搬入バッファへの搬入余裕時間は<s>分)
THEN<min /z,s,t/>(作業ステーション<x>から<y>の搬送候補の搬送余裕時間は<t>)

```

⋮

図3.11 優先度付けルールの修正

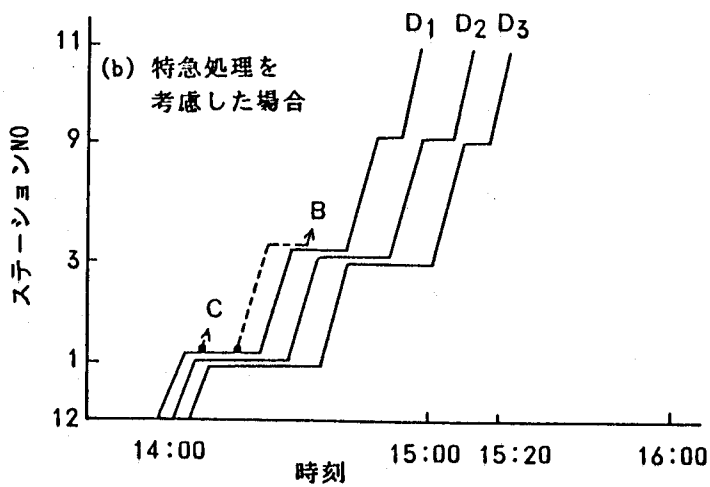
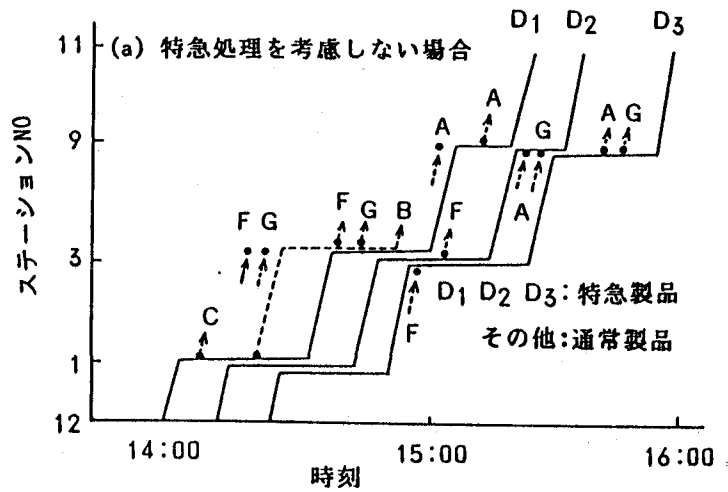


図3.12 特急処理製品の処理経過

表3.4 作業ステーションの稼働率結果

ステーションNO	1	2	3	4	5	6	7	8	9	10	平均
運用種別											
通常運用	90	99	89	83	93	96	86	98	98	88	92
特急処理の追加	93	70	90	70	76	77	86	85	91	86	82

### (3) 搬送車割付アルゴリズムの効果

3.2で述べた運行制御目的の実現の度合は、搬送優先条件の与え方と、搬送車割付アルゴリズムの良し悪しに影響される。提案方式では、搬送優先条件を容易に効果的なものに変更可能であるが、この面まで含めて提案方式を従来方式と評価するのは困難であるため、搬送車割付アルゴリズムに関し、両方式をシミュレーションにより比較評価した。対象としたレイアウトは、図3.1に示すもので、搬送優先条件は両方式とも図3.3に示すものとした。

シミュレーション結果の1例を図3.13に示す。同一のステーションの平均稼働率90%を達成するため、従来方式では9台の搬送車が必要であるが、提案方式では、6台の搬送車で済む。これは、発生している搬送要求にただちに作業待ち搬送車を割り付けるより、その時点で作業中の搬送車が作業終了後にその搬送要求を処理した方が早く処理できるケースが頻繁に発生するためである。

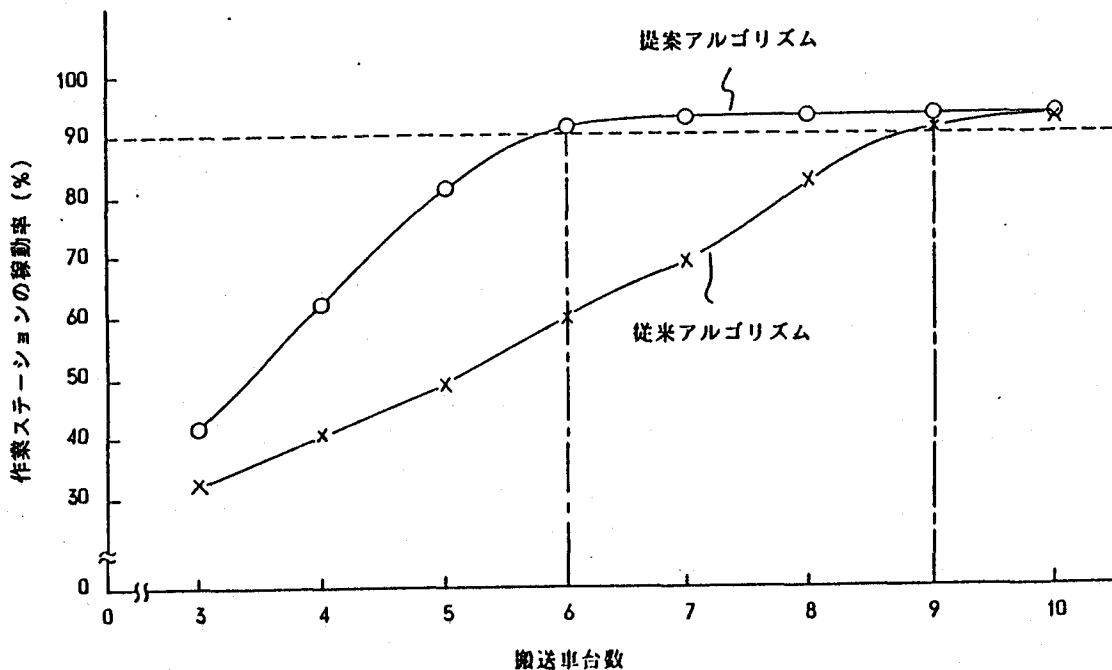


図3.13 提案アルゴリズムのシミュレーション結果

### 3.7.2 プリント基板組立ショップへの適用<sup>11), 12)</sup>

#### (1) 作業の流れと特徴

レイアウトを図3.14に示す。本ショップでは、基板にIC等の素子をインサータにより挿入し、各種の製品パッケージを作成する。まず、素子が未挿入の基板が自動倉庫から出庫され、数十枚の一定単位ごとにマガジンと呼ばれる箱に格納され、投入コンベアに投入される。インサータには、搬送車からマガジンを受け取るためのローダと搬送車へマガジンを渡すためのアンローダが設置されている。それぞれ、数マガジン分を一時保管で



きる容量を持ち、これにより、組立作業と搬送作業とを並行に実施できるようにしている。各インサータでは、マガジンから基板を取り出し、所定の素子を挿入し、マガジンに戻すという作業が行われる。なお、前述の適用対象と異なり、前回と今回の処理パッケージのサイズが異なる場合には、その切り替えのための段取り作業を要する。全工程経路をマガジンが経ると、そのなかの基板は製品として完成される。製品格納のマガジンは、検査ショップへの払出コンベアに搬送される。

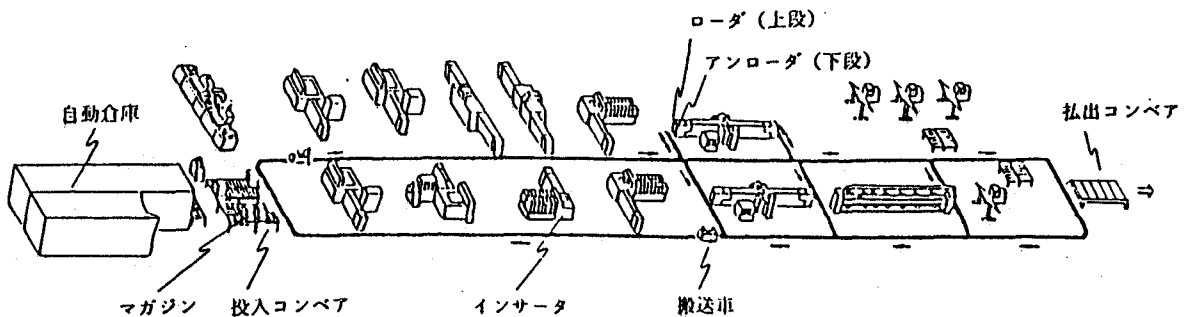


図3.14 プリント基板組立ショップのレイアウト

## (2) 計算機制御システムの構成

本適用対象の計算機構成を図3.15に示す。搬送車群制御は16ビットマイコンで行っている。搬送可否条件の判定に必要な各設備状態、搬送優先度決定に必要な、コンベア等に搭載されているマガジン数は、それぞれの機器コントローラから伝送される。また、各マガジンの次の行き先を判断するための工程経路データは、マガジンを投入コンベアに投入するたびに、各マガジンに対応して付けられているロットNO.をキーにして、ショップ管理コンピュータから送られてくる。マガジンの流れは物の流れが乱れなければ、投入時のロットNO.、搬送車への搬送指示内容、インサータでのマガジン受け渡し情報をもとに、正確にトラッキングできる。しかし、人手でインサータにマガジンを供給し、作業をさせたい場合がたまにあり、この場合、前述のトラッキング方法では実際の物の流れとトラッキング情報が食い違ってしまう。この解決のため、各インサータでは、作業開始時、ロットNO.を品名読み取り機で読み取り、その情報をDNCマイコン等を介して、搬送車群制御用マイコンに伝送している。導入した搬送車はセットポイントコントロールタイプであるため、搬送車群制御用マイコンはホームポジションでのみコントローラ(地上局)を介して搬送車上コントローラと搬送車到着、作業指示等の情報の受け渡しが行える。なお、前述の適用対象に導入された搬送車は、搬送車がどこにいても作業指示等の情報の受け渡しが行える無線タイプである。

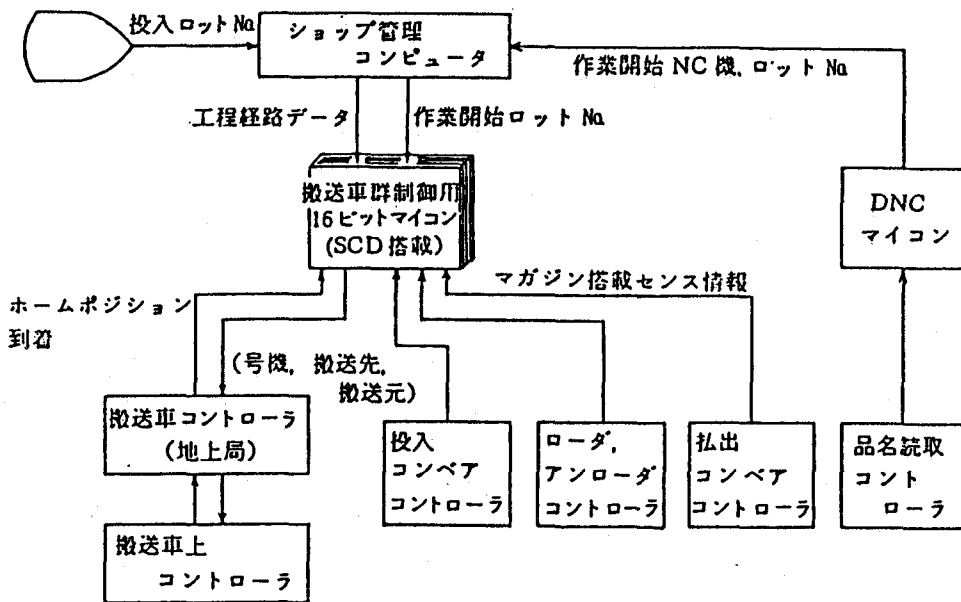


図 3.15 計算機構成

### (3) 制御ルール

制御ルールは約 80 である。ルール群の構成、各ルール群内のルールは前述の適用対象のものと同様である。異なる点はつぎの 2 点である。まず、搬送候補抽出ルール群において、搬送元候補のマガジンの次工程が複数ある場合、インサータでの段取り作業が不要な搬送先を選択する。また、搬送優先度付けルール群では、各マガジンの処理時間まで考慮せず、ローダ、アンローダ上のマガジンの数に基づいて優先度を付ける。

### (4) 搬送車割付アルゴリズム

導入した搬送車はセットポイントコントロールタイプであるため、ホームポジションでのみ作業指示の情報が受け取れる。そのため、インサータの稼働率向上を目的に、搬送車がホームポジションを出発して戻ってくるまでの 1 サイクルで、搬送優先度の高い搬送候補をできるだけ多く搬送可能とする搬送車割付アルゴリズムを開発した。

この適用例についても、前述の適用例と同様に提案方式の有効性を確認している。

## 3.8 まとめ

第 3 章では、問題の記述に知識工学を応用し、組合せ計算はアルゴリズムで対応すべき運用制御問題を、無人搬送車の運行制御問題を対象に論じた。明らかにした事項は次の通りである。

(1) 搬送車の運行制御問題を条件判定的な問題と組合せ最適化的な問題に分離する構造を

提案した。その条件判定的な問題では、パラメータの値の変更では対応できない、作業ステーションへ物を搬送できるか否かの搬送可否条件、どの搬送要求から優先的に搬送すべきかの搬送優先条件などが頻繁に変更されるため、それらの条件はルールによる記述方式とした。一方、分離された組合せ最適化的な問題は、搬送要求に対する搬送車割付の問題であり、ほとんど変更が発生しないため、これに対して、高効率で標準化された搬送車割付アルゴリズムを提案した。

(2)搬送車導入のシステムの計画に際し、各システムに即した運行制御方式、軌道レイアウト、必要な搬送車台数を短時間で決定するために、上記の搬送車群運行制御方式を組み込んだシミュレータを開発した。

(3)提案した運行制御方式により変更が容易にできることを特急処理運用の追加を例に示した。また、提案した搬送車割付アルゴリズムは、従来のアルゴリズムに比べ、効率が向上できること(同じ処理量/時間に対して、必要搬送車台数が少なくて済むこと)をシミュレーションにより示した。

(4)提案方式を組み込んだ運行制御システムは、すでにプリント基板組立<sup>11)</sup>、<sup>12)</sup>などに適用されており、その制御ソフト開発工数の削減、その後のメンテナンスの容易性といった効果が確認されている。

参考文献

- (1)篠塚, 吉谷: 自動倉庫・搬送システム, 計測と制御, Vol.22, No.11 pp317-325 (昭和58年11月)
- (2)N.J.Nilson: Principles of Artificial Intelligence, Tioga Publishing Co., Palo Alto (1980)
- (3)都島, 薦田, 岡坂: 知識工学を応用した搬送車群運行制御方式, 計測自動制御学会論文集, Vol.24, No.3, pp300-307 (昭和63年3月)
- (4)都島, 薦田: 知識型搬送車群運行制御方式, 電気学会システム・制御研究会, SC-86-14, pp21-30 (昭和61年8月)
- (5)I.Tsushima and N.Komoda: Knowledge Engineering Approach to Traffic Control for Automated Guided Vehicles, Proc. of International Workshop on Industrial Applications of Machine Vision and Machine Intelligences, pp261-265 (1987-2)
- (6)K.Okasaka and I.Tsushima: Automated Storage and Retrieval System(AS/RS) Using AI(Artificial Intelligence), Proc. of the 5th Int. Conf. on Automated Guided Vehicle System, pp1-18 (1987-10)
- (7)田代, 薦田, 都島, 松本: ルール型ソフトウェアシステム S C D (Station Coordinator)の開発, 情報処理学会論文誌, Vol.27, No.5, pp552-561 (昭和61年5月)
- (8)田代, 薦田, 都島: ルール記述に基づくシステム制御方式, 計測と制御, Vol.22, No.9, pp786-792 (昭和58年9月)
- (9)都島, 岡坂: F A用搬送車群運行制御シミュレータ, シミュレーション, Vol.7, No.1, pp10-16 (昭和63年3月)
- (10)都島, 岡坂: F A用搬送車群運行制御シミュレータ, 情報処理学会第34回全国大会, 2G-10, pp2173-2174 (昭和62年3月)
- (11)都島, 薦田, 松本, 原田: プリント基板組立ショップにおけるルール型搬送車群制御方式, 第4回SICE知識工学シンポジウム, pp1-6 (昭和61年3月)
- (12)佐川, 小島, 平原, 井上, 都島, 原田: 多品種中・少量プリント板組立におけるファクトリーオートメーション-コンピュータ機器-, 日立評論, Vol.68, No.12, pp945-950 (昭和61年11月)

## 第4章 問題記述に知識工学を応用した ピッキング効率向上運用方式

## 第4章 問題記述に知識工学を応用した ピッキング効率向上運用方式

### 4.1 まえがき

自動倉庫はFactory Automationの重要な構成要素であるが、注文に応じて自動倉庫内に保管されているパレットから必要個数の製品(品目)を取り出すピッキング作業は人手に頼らざるを得ず、しかも多くの人手がかかっている。例えば、ある報告では、倉庫作業の分析の結果、ピッキング作業に要する時間が全作業時間の30~40%に達していることが少なくないと述べている。そのため、ピッキング作業の効率向上が自動倉庫システムの最大の課題となっている。

ピッキングの方式を大きく分けると、2通りになる。第1は、ピッキング作業員(以降、ピッカーと呼ぶ)が徒歩やクレーンに乗って製品のもとまで移動し、ピッキングする方式である。この方式の問題点はピッカーが移動中、ピッカーはピッキングできないことであり、これがピッキング効率を低下させている。

他の1つは、製品の方をピッカーのもとに流し、ピッキングする方式である。この方式を実現するものとして、無人クレーンとラックからなる自動倉庫と、全棚を回転させ、製品をピッカーのもとに搬送する回転倉庫とがある。前者は棚が回転しないため、以降、固定倉庫と呼ぶ。

固定倉庫の問題点は、クレーンによりパレットを出庫するのに時間がかかることである。ピッカーが物の到着待ちにより無作業にならないようにするには、多数のクレーンが必要となり、多大のコストがかかる。一方、回転倉庫は、機構が複雑で、しかも全棚を動かすため、設備コスト、ランニングコストがかかるという問題がある。

そこで、以上の問題点を解決するため、設備構成としては、小容量のピッキング用回転倉庫と、それに対してパレットを供給するための少数のクレーンを持つ固定倉庫からなる二段階倉庫とし、運用方式の工夫によりピッキング効率を向上しようと考えた。

ピッキング効率を向上するには、ピッキングすべき回転倉庫の棚がピッカーの前にくるまでの待ち時間を低減することである。このために、ある注文でピッキングされたパレットが先の注文で再度ピッキングされる場合には、それをアクセスタイムの短い回転倉庫に残しておくという基本的な運用が考えられる。しかし、二段階倉庫では、回転倉庫が小容量のため、そのパレットが再度ピッキングされるまでの時間を低減する必要があり、そのための注文処理順序の並び換え運用アルゴリズムの開発が第1の課題である。

第2の課題は上記の運用アルゴリズムの導入によって発生する課題である。注文処理順序を並び換える場合、納期条件等から全ての注文に対して上記の運用アルゴリズムは適用できず、ある条件下で処理順序を並び換えてもよい注文群に分割する必要がある。それらの条件は対象によって異なり、変更も多いため、変更に対応できる注文分割運用方式の開発が第2の課題である。

本章では、問題記述に知識工学を応用する形態に適した問題として、上述のピッキング効率向上運用問題を取り上げる。注文分割運用問題は条件判定的な問題であり、変更への

柔軟性を要求されるため、ルールで記述する方式を提案する。ピッキング効率を高めるために、分割された注文を対象にその注文内容のデータを利用し、処理順序に近い注文間に共通品目が多くなるように注文処理順序を決定する運用アルゴリズム<sup>(1)</sup>を提案する。また、決定された注文処理順序のもとで、回転倉庫上には近い将来再度ピッキングされるパレットが格納されるように、回転倉庫から固定倉庫へ払い出すパレットを決定する運用アルゴリズムを提案する。さらに、提案運用方式の有効性をシミュレーションにより、固定倉庫のみのシステムとの比較で示す。

## 4.2 二段階倉庫システムの定式化

### 4.2.1 ピッキング作業の説明

二段階倉庫システムの定式化の前に、そこでのピッキング作業を図4.1により説明する。

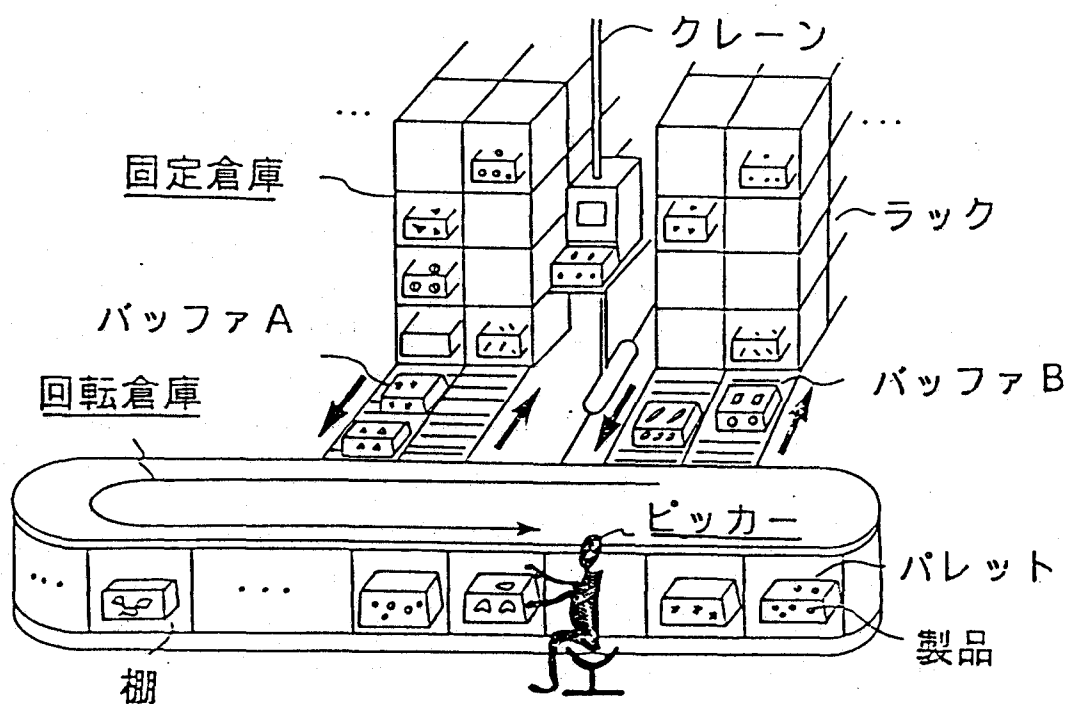


図4.1 二段階倉庫システム

製品は、形状、大きさの統一されたパレットに格納されている。各パレットには、同一品目の製品が多数格納されており、このパレットから注文で要求された品目を要求数量だけ取り出す作業が、ピッキング作業である。ピッキング対象の品目を格納しているパレットは、固定倉庫の棚からクレーンにより、バッファAを介して回転倉庫の棚へ搬送される。ピッカーは移動せず、ピッキング対象の品目を格納している回転倉庫棚(1棚に1パレット

ト格納)がピッカーの前にくると、一定速度で回転している棚からピッキングする。ピッキング不要となったパレットは、回転倉庫の棚からバッファBを介して、クレーンにより固定倉庫の棚へ戻される(再入庫される)。バッファA, Bの機能は、回転倉庫と固定倉庫でのそれぞれのパレット搬送作業を独立に、並列に実施できるようにするためのバッファリング機能である。

各注文では、一般に複数品目に対するピッキングが要求される。1注文内では、どの品目からピッキングしてもよく、1注文の製品をまとめあげればよい。ただし、各注文は1つつつ、順次まとめられる。

ピッカーに対するピッキング指示書は、受注データにもとずき作成される。提案する運用方式では、従来のように受注順に注文を処理するのではなく、注文の処理順序を並び換えるため、以下の点を考慮する必要がある。各注文により要求された製品は受注時点からある日数以内に注文主に配送しなければならないという納期が与えられる。また、特別に、短時間での配送が要求される緊急注文を考慮しなければならない場合もある。さらに、配送先によって、配送時間が異なる。

#### 4.2.2 二段階倉庫の運用問題

本章で扱う二段階倉庫の運用問題を定義する。この問題は、4.1で述べたように、以下の2つの問題に分けられる。

(問題1)受注した注文を各種の配送要求の条件に基づき、その注文群内では注文処理順序を並び換えてよい注文群に分割する問題

(問題2)分割された各注文群に対してピッキング効率の向上のため、注文処理順序の決定と回転倉庫から固定倉庫へ払い出すパレットの決定の問題

問題1は条件判定的な問題であり、指定される条件は対象により異なり、変更も多い。そのため、ルールによる記述方式とした。その記述例を図4.2に示す。この例では、各日の夜に締め切った受注データをもとに、翌日の午前と午後にピッキング処理すべき2つの注文群に分割している。例えば、緊急注文であれば、それは翌日の午前にピッキングする。通常の注文であっても、遠距離の配送先の注文は午前中にピッキングする。

```
IF (注文(X)が緊急注文)
THEN(注文(X)を翌日の午前にピッキング)

IF (注文(X)が通常注文)
IF (注文(X)の配送先は遠距離)
THEN(注文(X)を翌日の午前にピッキング)
:
:
```

図4.2 注文群分割ルール

問題2はスケジューリングアルゴリズムの問題である。この問題は、注文、回転倉庫、固定倉庫の特性パラメータが与えられたもとで、ピッキング効率の向上を目的に、固定倉庫から回転倉庫へのパレットの投入運用に係わる注文の処理順序の決定と、固定倉庫への回転倉庫上のパレットの払出し運用に係わる払出しパレットの決定である。



問題2に関し、以下、ピッキング効率、特性パラメータ、決定項目を述べる。なお、注文処理順序の決定、払出しパレットの決定については、4.3で述べる。

(1)ピッキング効率  $e$

$$e = \frac{\text{(正味のピッキング時間)}}{\text{(ピッカーの拘束時間)}} \dots\dots\dots (4.1)$$

ここで、分子はピッカーが実際にピッキングしている時間である。分母のピッカーの拘束時間は、正味のピッキング時間とピッキングすべき回転倉庫棚がピッカーの前に回ってくるまでの待ち時間との和である。

(2)特性パラメータ

(a)注文特性

$K$  : ピッキング作業計画対象の注文数。

$O^*_k$  for  $k = 1 \sim K$  : ピッキング作業計画対象で、 $k$  番目に受注した注文名。

$O^*_k$  for  $k = 1 \sim K$  : 注文 $O^*_k$ を構成する品目の集合。

$N(O^*_k)$  for  $k = 1 \sim K$  : 注文 $O^*_k$ の品目の種類数。

なお、1品目当たりのピッキング時間は、品目、それぞれの品目に対するピッキング数量によって異なるが、本章では、1品目当たりのピッキング時間が品目、ピッキング数量に関係なく、一定とみなせる範囲の倉庫を対象とする。

(b)回転倉庫特性

$l$  : 回転倉庫の棚数。

$v$  : ピッカーの前を単位時間当たり通過する回転倉庫の棚数(棚数/分)。

(c)固定倉庫特性

$t_s$  : 固定倉庫から回転倉庫への1パレット当たりの出庫時間。

$t_s$ は固定倉庫のクレーン群を1台のクレーンとして捉えた場合のパラメータである。

この $t_s$ は、クレーン台数、クレーン単体性能、固定倉庫の棚数の要素から決まるが、ピッキング作業から見れば、固定倉庫からのパレット出庫時間として捉えればよい。

パレットを回転倉庫へ出庫すると、そのパレットの固定倉庫への戻しがあるため、1パレットを処理するのに要する固定倉庫能力は $2t_s$ となる。

$Z$  : 在庫品目の集合。

$N(Z)$  : 在庫品目の種類数。

(3)決定項目

・ 注文の処理順序 $O_k$  ( $k = 1 \sim K$ )

$O_k$ は $k$ 番目に処理する注文名を意味する。なお、4.4で述べる注文処理順序の決定において使用する属性として、注文 $O_k$ の品目の集合を $O_k$ と定義しておく。

・ 回転倉庫上からの固定倉庫への払出しパレット。

## 4.3 二段階倉庫の運用アルゴリズム

### 4.3.1 運用の考え方

ピッキング効率を高めるには、ピッキングすべき回転倉庫の棚がピッカーの前に回って

くるまでの待ち時間を低減する必要がある。この実現のため、ある注文に対してピッキングしたパレットが、先の注文で再度ピッキングされる場合には、そのパレットをアクセスタイムの長い固定倉庫に戻さず、そのままアクセスタイムの短い回転倉庫に残す。しかし、従来のように、受注時点の早い注文から順にピッキング作業を実施したのでは、ある注文でピッキングされたパレットが再度ピッキングされるのは、かなり先の注文になる可能性が高い。従って、多量のパレットを回転倉庫上に残しておく必要がある。しかし、回転倉庫上のパレットが多くなると、回転待ち時間が長くなり、ピッキング効率が低下する。

上記の問題点を解決するには、再度ピッキングされるパレットの回転倉庫上での滞留時間を低減する必要がある。そこで、事前にわかっている注文データを利用して、処理順序が近い注文間に共通品目が多くなるように、注文の処理順序を決定する。このアルゴリズムの詳細は4.3.2で述べる。

注文処理順序の決定は、回転倉庫へのパレットの投入運用に関するものであるが、運用目的である、再度ピッキングされるパレットの回転倉庫上での滞留時間の短縮には、満杯である回転倉庫上から固定倉庫へのパレットの払出し運用が必要である。これについては回転倉庫上のパレットのうち、最も先の注文で再度ピッキングされるパレットから順に、これから処理しようとしている注文の品目で回転倉庫上にない品目数分を払出しパレットとして選択する方式とする。

#### 4.3.2 注文処理順序の決定アルゴリズム

注文処理順序の決定では、各注文 $O_k$  ( $k=1\sim K$ )に対し、その品目のうち、 $O_k$ 以前に処理される注文 $O_{k-j}$  ( $j=1\sim k-1$ )のどの注文から何種類の品目を注文 $O_k$ のピッキングのために回転倉庫上に残すことができるかを把握する必要がある。このためには単純に注文 $O_k$ と注文 $O_{k-j}$ の共通品目の種類数を求めることでは対応できない。注文 $O_k$ の品目が、注文 $O_k$ 以前に処理される注文 $O_{k-j}$ の複数注文に含まれる場合には、該当の品目は処理順序として注文 $O_k$ に最も近い注文 $O_{k-j}$ に含まれ、それ以外の注文 $O_{k-j}$ には含まれないとしなければならない。このことを考慮した尺度として、以下に示す $p_{k-j,k}$  (以降、注文間の相関と呼ぶ)を導入する。

$$\begin{aligned}
 p_{k-j,k} &= \frac{N(O_{k-1} \cap O_k)}{N(O_k)} && \text{for } j=1, k=2\sim K \\
 &= \frac{N(O_{k-j} \cap \bar{O}_{k-j+1} \cap \dots \cap \bar{O}_{k-1} \cap O_k)}{N(O_k)} && \text{for } j=2\sim k-1, k=3\sim K \\
 &&& \dots\dots\dots (4.2)
 \end{aligned}$$

ここで、 $\bar{O}_j$ は、在庫品目の集合 $Z$ に対する $O_j$ の補集合を意味する。

(4.2)式の分子は、注文 $O_k$ のピッキングのため、それ以前に処理される注文 $O_{k-j}$ から回転倉庫上に残される品目の種類数を示す(図4.3参照)。分母は、各注文 $O_k$ の品目種類数の違いを吸収するためのものである。

さて、 $j$ 注文先の注文のピッキングのために回転倉庫上にパレットを残すと、 $j$ 注文分の処理が終了するまでそのパレットは回転倉庫の棚をピッキングされない状態で占有する。従って、与えられた回転倉庫容量(棚数)のもとで、再度ピッキングされるために残された

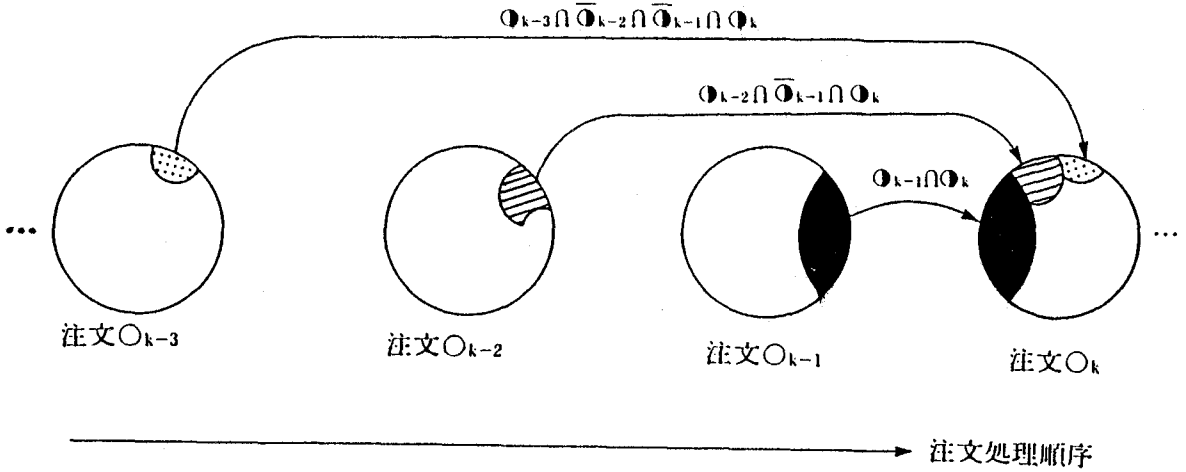


図 4.3 注文間のパレットの推移

パレットの回転倉庫上での滞留時間を低減するためには、 $j$  が小さいほど  $p_{k-j,k}$  の値が大きくなるように注文処理順序を決める必要があり、次式に示す目的関数  $f$  を導入する。

$$f = \sum_{k=2}^K \sum_{j=1}^{k-1} (1/j) p_{k-j,k} \rightarrow \text{maximize} \dots \dots \dots (4.3)$$

ここで、 $p_{k-j,k}$  の係数  $1/j$  は、再度ピッキングされるパレットの回転倉庫上での滞留時間の低減を考慮するためのものである。

(4.3)式の目的関数を最大にする注文処理順序決定問題は、注文をノード、 $p_{ij}$  をアーク上に定義されるコストと見なせば、トラベリングセールスマン問題に似たものとなる。違いは、トラベリングセールスマン問題のように、各アークのコストがその2つのノード  $i, j$  だけで唯一に定義されるだけでなく、ルートとして、(4.2)式に示すように  $i$  と  $j$  の間に他のノードが入る場合のそれぞれに対してコストが定義されなければならない点である。そのため、本問題は、トラベリングセールスマン問題に比べて、はるかに難しい問題である。

そこで、逐次、注文処理順序  $O_k (k = 1 \sim K)$  を決定していく現実的なアルゴリズムを考案した。基本的な考え方は、既に、 $(k-1)$  番目までの注文処理順序を決めたとすると、次に処理順序未決定の注文のなかから、次の目的関数  $g$  を最大にする注文を  $k$  番目に処理すべき注文  $O_k$  とする。

$$g = \sum_{j=1}^K (1/j) p_{k-j,k} \rightarrow \text{maximize} \dots \dots \dots (4.4)$$

なお、本アルゴリズムでは逐次、処理すべき注文を決定していくため、 $g$  は(4.3)式の  $f$  の  $\sum_{k=2}^K$  を除いたものに対応する。また、第1番目に処理すべき注文  $O_1$  の決定の際、既に処理順序が決定された注文がないが、 $O_1$  の処理では、ピッキング開始前の回転倉庫上

の品目がピッキング効率に影響を与えるため、回転倉庫上の品目の集合を注文 $O_0$ と見なす。そのため、(4.4)式の上限は $k$ となる。

以上の考え方に基づく計算手順を以下に示す。

#### ステップ1：前処理

- ・処理順序未決定の注文の集合を $R$ とし、 $R$ に受注時点の早い順に順序付けられた注文の集合 $\{O^*_1, O^*_2, \dots, O^*_k\}$ と注文 $O_0$ を設定する。
- ・決定すべき注文番号 $k$ に1を代入する。

#### ステップ2： $k$ 番目に処理すべき注文名 $O_k$ の決定

- ・ $R$ に含まれる各 $O^*_1$ に対し、 $O^*_1$ を仮に $O_k$ と見なし、 $g_1 = \sum_{j=1}^k (1/j) P_{k-j, k}$ を計算する。 $g_1$ を最大にする $O^*_1$ を $O_k$ に決定する。また、その $O^*_1$ に対応する品目の集合を $O_k$ とする。

#### ステップ3：後処理

- ・ステップ2で $k$ 番目に処理すべき注文と決定した $O^*_1$ を $R$ から削除する。
- ・ $k = K$ ならば、計算を終了する。 $k \neq K$ ならば $k \leftarrow k + 1$ とし、ステップ2へ戻る。

## 4.4 評価

### 4.4.1 評価方法

#### (1) 評価の概要

提案する二段階倉庫運用は、ピッキング効率の向上を目的とした、注文処理順序の決定運用と回転倉庫からのパレットの払出し運用である。それらの運用の有効性評価のため、ピッキング効率を注文、回転倉庫、固定倉庫の特性パラメータとの関係で、以下の三つの倉庫システム間で比較評価する。

(提案システム1)：注文処理順序の決定運用と、パレットの払出し運用を実施するシステム。

(提案システム2)：パレットの払出し運用を実施するシステム。

(固定倉庫のみのシステム)：ピッキングは受注時点の早い注文から順に実施され、ピッキング後のパレットは注文毎に逐一、固定倉庫に戻されるシステム。

提案システム2を設定した理由は、注文処理順序の決定運用とパレットの払出し運用のそれぞれの有効性を評価するためである。

なお、回転倉庫の速度 $v$ は、一品目当たりのピッキング時間を $c$ とすると、ピッカーの前を1棚が通過する間に所定のピッキングが完了する範囲内( $v \leq 1/c$ )でなければならない。本評価においては、最も効率的な $v = 1/c$ で回転するとし、以降で述べるシミュレーション評価により、提案システムのピッキング効率を求める。

一方、固定倉庫のみのシステムにおけるピッキング効率 $e$ は簡単に解析的に求められるので、以下に説明しておく。

固定倉庫のみのシステムでは、ピッキング後のパレットは注文ごとに逐一、固定倉庫に戻される運用が採られるため、1回のピッキング作業をもとに $e$ を計算すればよい。ピッカーのもとへ固定倉庫から1パレットを供給できる時間は、再入庫時間も考え、 $2t_s$ で

ある。クレーン作業とピッカーの作業は独立に実施できるため、 $2t_s$ 時間が1回のピッキング当たりのピッカーの拘束時間となる。一方、1回のピッキング時間は $c$ である。従って、固定倉庫のみのシステムのピッキング効率 $e$ は次式で求められる。

$$e = c / (2 t_s) \dots\dots\dots (4.5)$$

## (2)シミュレーション手順

提案システムのピッキング効率は、解析的に求めることができないため、シミュレーションにより求めた。シミュレーションに当たり、以下の前提を設定した。

- ①注文を構成する品目の種類数は正規分布(平均 $n$ ，標準偏差 $\sigma$ )に従う。
- ②在庫品目のそれぞれが、ピッキング指示される確率は品目の種類に関係なく一定である。

シミュレーションの手順は以下の通りである。

- (i)  $K, N(Z), n, \sigma$  を定義する。
- (ii) 正規分布( $n, \sigma$ )をもとに、初期の各注文 $O^*_k$ の品目の種類数 $N(O^*_k)$  ( $k = 1 \sim K$ ) を決定する。
- (iii) 各注文の品目の集合 $O^*_k$  ( $k = 1 \sim K$ )を、②の前提のもとで決定する。
- (iv) 以上のように決定した注文群に対して、注文処理順序の決定アルゴリズムを適用し、注文処理順序 $O_k$  ( $k = 1 \sim K$ )を決定する。

なお、提案システム2に対しては、上記アルゴリズムを適用せず、 $O_k = O^*_k$ とする。

- (v)  $l, c, t_s$ のそれぞれに値を設定し、注文処理順序 $O_k$  ( $k = 1 \sim K$ )に従い、二段階倉庫の振舞いをシミュレータでシミュレーションし(パレット払出し運用の適用)、ピッキング効率を算出する。

### 4.4.2 シミュレーション結果と考察

主なパラメータを以下に示す。

- (a) 在庫品目の種類数  $N(Z) = 100$
- (b) 品目数/注文  $n, \sigma \dots 6$  ケース
  - (ケース1)  $n = 50, \sigma = 5$
  - (ケース2)  $n = 50, \sigma = 2.5$
  - (ケース3)  $n = 20, \sigma = 2$
  - (ケース4)  $n = 20, \sigma = 1.0$
  - (ケース5)  $n = 10, \sigma = 1$
  - (ケース6)  $n = 10, \sigma = 5$

$n$ については、 $N(Z)$ に対する割合が大、中、小を設定した。 $\sigma$ は、大、小を設定し、それぞれ、 $n$ に対する比率を一定とした。

#### (1) 注文処理順序の決定アルゴリズムの適用による注文間の相関の改善評価

図4.4は、注文処理順序の決定アルゴリズムの適用により並び換えした注文列と並び換え前の注文列に対する注文間の相関 $p_{k-j,k}$ の計算結果例を示す。横軸は、二つの注文の間が処理順序において何注文はなれているかを示し、注文間が $j$ 注文はなれていることを注文間の距離は $j$ という。

図4.4の(1),(2)の結果から、 $n/N(Z)$ の値が小さいほど、並び換え後の $p_{k-j,k}$ は

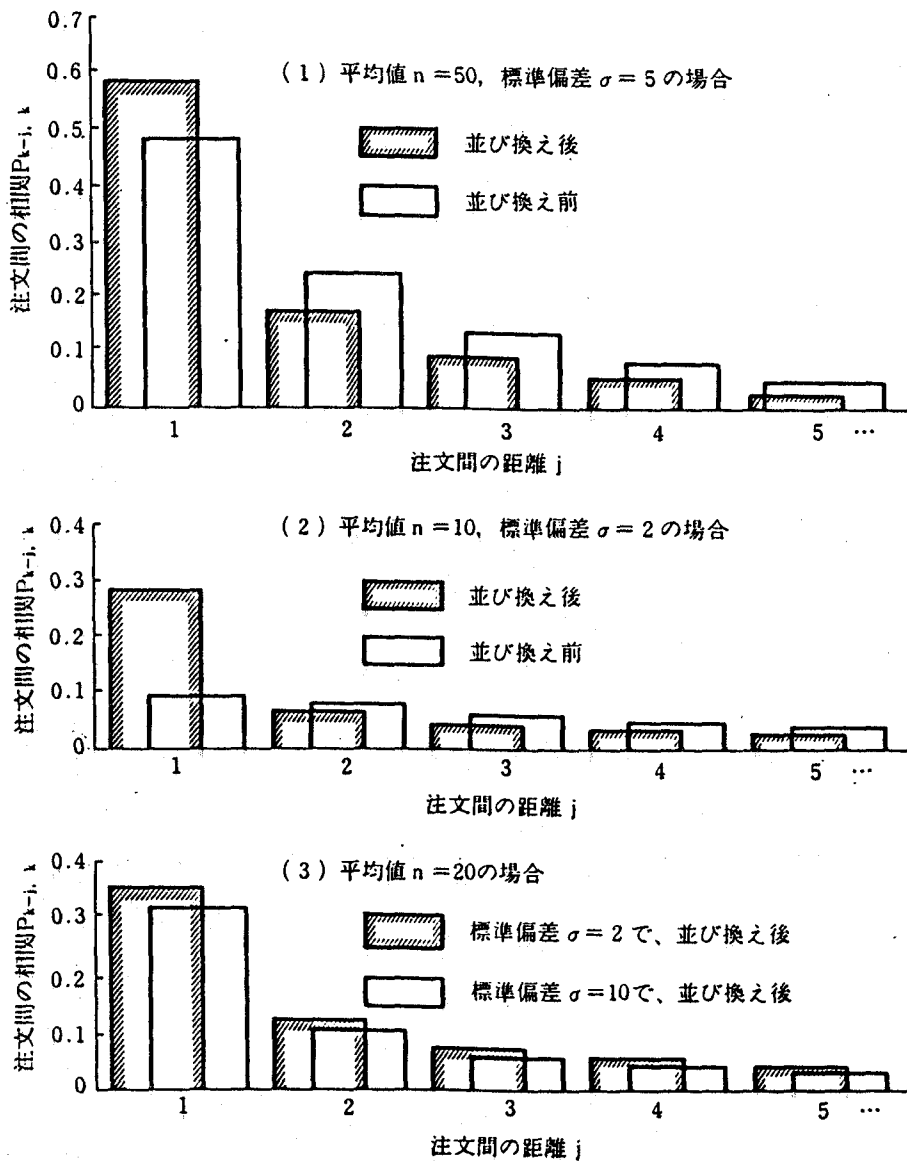


図 4.4 注文間の相関の計算結果

並び換え前の場合に比べ、距離  $j$  の近い注文間に集中することがわかる。これは  $n/N(Z)$  の値が大きい場合には、並び換え前でも、任意に注文間に高い相関が存在するためである。なお、並び換え前の注文列においても、 $p_{k-j,k}$  が注文間の距離  $j$  の増加に伴い減少しているのは、(4.2)式の定義から明らかである。

以上のことから、 $n/N(Z)$  が小さいほど、注文処理順序の決定アルゴリズムが効果的に働くことがわかる。

図 4.4 の(3)は、注文当たりの品目数の平均値  $n$  が同じで、 $\sigma$  が大、小の場合について、 $p_{k-j,k}$  の計算結果を示したものである。この結果から、注文当たりの品目数がかな

りばらについても、注文処理順序の決定アルゴリズムの効果は、大きくは低下しないことがわかる。なお、注文当たりの品目数がばらつくと、全体的に  $p_{k-j,k}$  が低下する理由は、相関をとる注文の品目数の違いから、共通製品をさがす空間が狭められることによると考えられる。

## (2) ピッキング効率の改善評価

図4.5は、提案システム1，提案システム2，固定倉庫のみのシステムに対し、 $n, \sigma, 1$ をパラメータに、ピッキング効率  $e$  と  $c/t_s$  との関係の計算結果を示す。なお、 $\sigma$  の小さいケースのみを示している。

全体的に見て、提案システムでは、固定倉庫のみのシステムに比べて、ピッキング効率が十分に向上することがわかる。例えば、 $c/t_s = 1$  の場合、固定倉庫のみのシステムでは、 $e = 0.5$  であるが、提案システム1の一例( $n = 20, \sigma = 2$  の場合)では、 $e \approx 0.9$  となる。

つぎに詳細を見ると、 $n$  が大きいほど ( $N(Z)$  を一定としているので、 $N(Z)$  が大きいほど)、提案システムの有効性の増すことがわかる。これは、 $n/N(Z)$  が大きいほど、

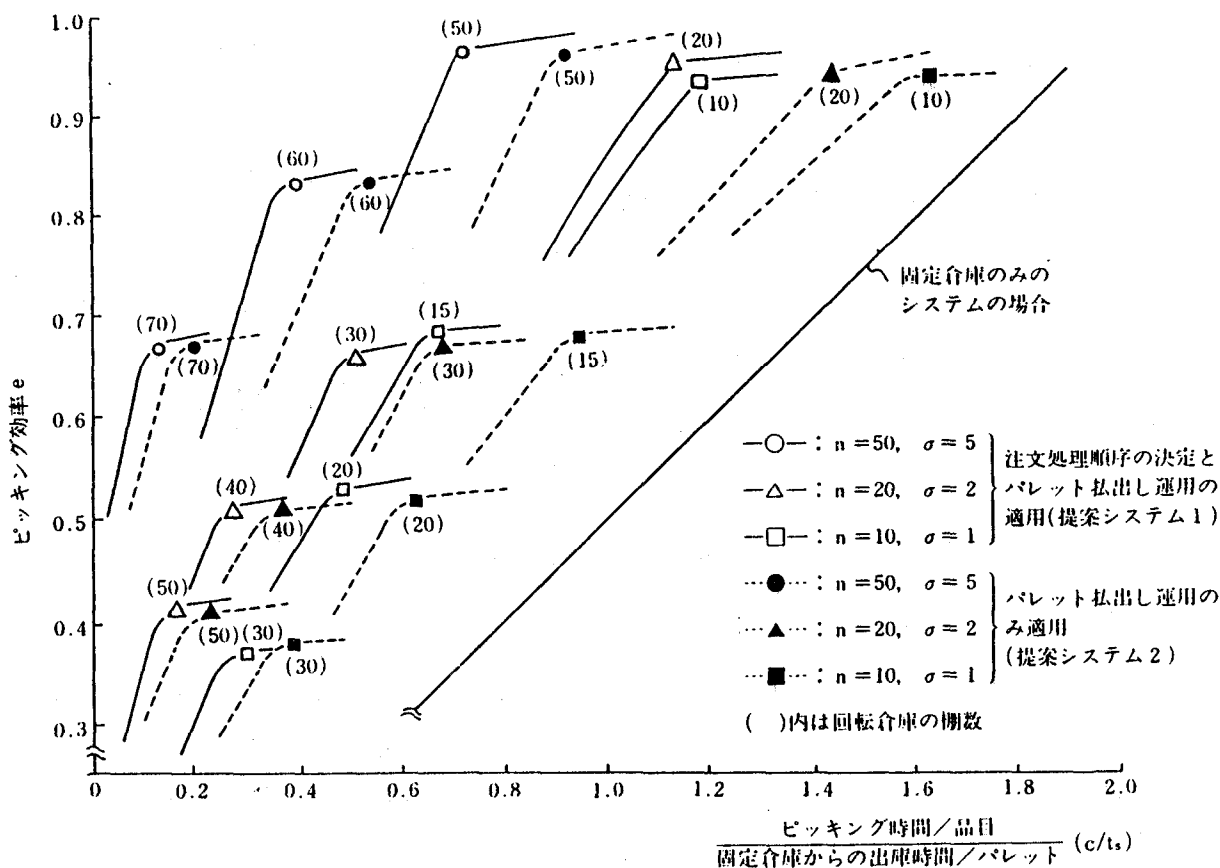


図4.5 ピッキング効率の計算結果1

$pk_{-j,k}$ が高くなるためである。ただし、その有効性の大部分は、パレット払出し運用によるものであることが計算結果からわかる(図4.5の-○-, -●-と固定倉庫のみのシステムのピッキング効率の比較から)。この場合、注文処理順序の決定運用が効果的でない理由は、4.4.2の(1)で述べたので、説明を省略する。

$n/N(Z)$ が小さいと、その値が大きい場合に比べれば、提案システムの有効性が低下するが、注文処理順序の決定運用とパレット払出し運用の両方が効果的に働くことがわかる(図4.5の-□-, -■-と固定倉庫のみのシステムのピッキング効率の比較から)。

図4.6は、提案システムの有効性の、注文当たりの品目の種類数のばらつき $\sigma$ による影響を示す。この結果から、その影響はあまり大きくないことがわかる。

なお、図4.5, 図4.6には、提案システムで実現されたピッキング効率 $e$ の前提となる回転倉庫の棚数 $l$ のパラメータ値を記載してある。 $l$ が一定の場合、 $c/t_s$ の増加( $c$ を一定と考えると、固定倉庫の能力の増加)に伴い、 $e$ は、始めは大きく向上し、ある点( $c^*/t_s^*$ と呼ぶ)以降はわずかしか増加しない。これは、 $c^*/t_s^*$ で回転倉庫の能力と固定倉庫の能力がほぼバランスするためである。 $c/t_s$ が $c^*/t_s^*$ 以降の領域では、固定倉庫の能力が過剰となり、 $c^*/t_s^*$ で実現されるピッキング効率はわずかしか改善されない。一方、 $c/t_s$ が $c^*/t_s^*$ 以前の領域では、固定倉庫のパレット搬送が追いつかず、その領域では $c/t_s$ が増加すると、それに見合う $e$ の向上が図れる。

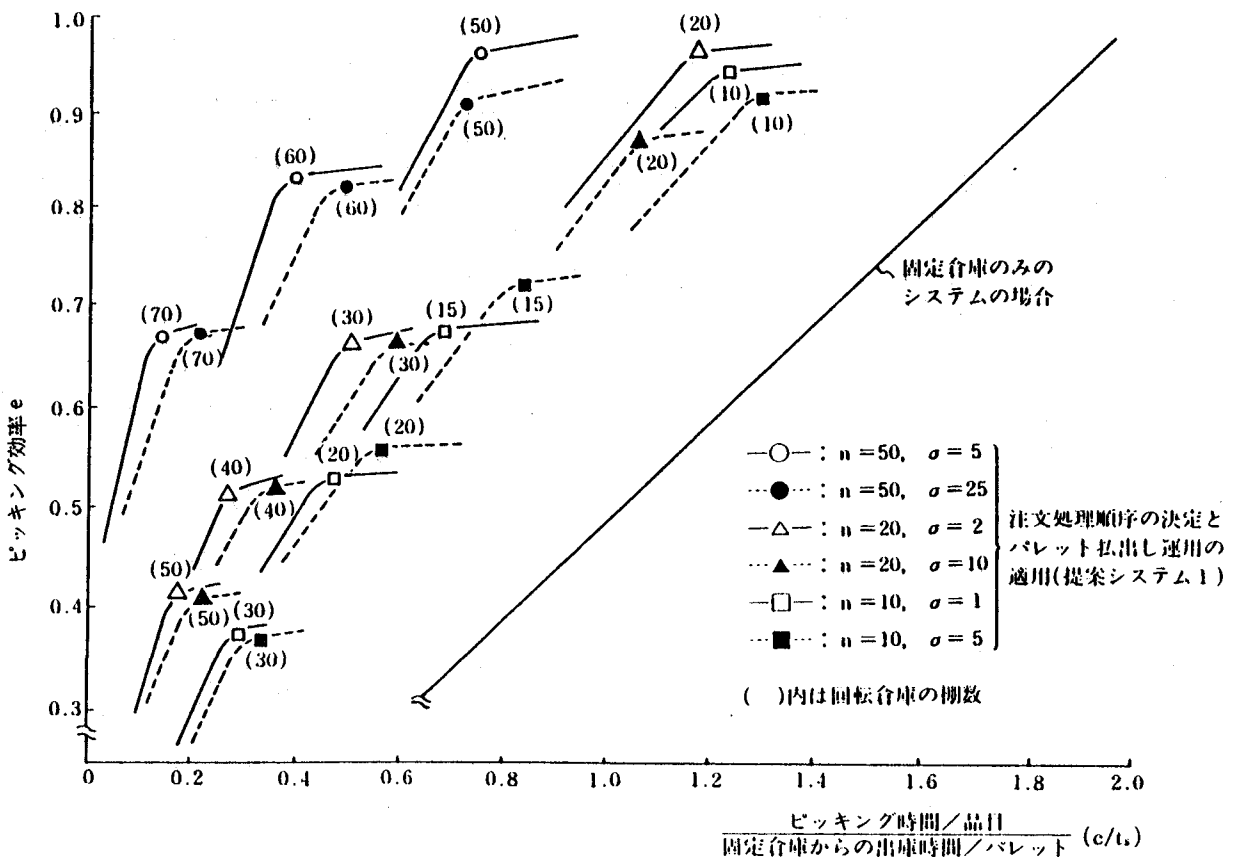


図4.6 ピッキング効率の計算結果2



#### 4.5 まとめ

第4章で明らかにした事項は次の通りである。

問題の記述に知識工学を応用し、組合せ計算はアルゴリズムで対応すべき問題を、ピッキング効率向上運用方式を対象に論じた。

(1)全体の注文をある条件、例えば、納期上、処理順序を並び換えてよい注文群に分割するという条件判定的な問題は対象により異なり、変更も多いのでルールで記述し、ピッキング作業が無作業にならないように、分割された各注文群に対して順序付けする問題はアルゴリズムで対応する方式を提案した。

(2)自動倉庫システムの設備構成としては、小容量のピッキング用回転倉庫と、それに対してパレットを供給するための少数のクレーンを持つ固定倉庫からなる二段階倉庫とし、運用アルゴリズムによりピッキング効率を向上しようと考えた。そのアルゴリズムとして、事前に判っている注文内容のデータを利用し、処理順序が近い注文に共通品目が多くなるように注文処理順序を決定する運用アルゴリズムと、決定された注文処理順序のもとで、回転倉庫上には近い将来再度ピッキングされるパレットが格納されるように、回転倉庫から固定倉庫へ払出すパレットを決定する運用アルゴリズムを提案した。

(3)提案方式によるピッキング効率と、その効率に影響を与える注文、回転倉庫、固定倉庫のパラメータとの関連をシミュレータにより明らかにした。その結果、提案方式により、高いピッキング効率を実現できる見通しを得た。

なお、自動倉庫の問題には、以上で述べた運用制御問題のほかに、各対象に即した倉庫形態、その規模を決定する計画問題がある。倉庫を計画する際には、その倉庫が使用される物流環境の違いに応じて計画されなければならない。そのためには、計画すべき倉庫の環境を調べる必要があり、それは膨大な作業量となる。そこで、必要最小限の環境データだけで多種類の倉庫形態の中から対象に即した倉庫形態を選択し、さらに適切な倉庫規模を決定する計画手法についても研究開発を行なった。

## 参考文献

- ( 1)鈴木：オーダピッキングの種類と手法，オーダピッキングハンドブック，Vol.28，No.10，pp8-16（昭和56年10月）
- ( 2)篠塚，関：F Aにおける無人搬送と貯蔵の自動化，計測と制御，Vol.22，No.11，pp27-36（昭和58年11月）
- ( 3)篠塚，吉谷：自動倉庫・搬送システム，計測と制御，Vol.22，No.11 pp317-325（昭和58年11月）
- ( 4)高橋：立体自動倉庫の動向と課題，日本機械学会誌，Vol.86，No.775，pp68-74（昭和58年6月）
- ( 5)J.A.Tompkins and J.D.Smith：Keys Developping Material Handling System for Automated Factory are Listed，Industrial Engineering，Vol.15，No.9，pp48-54（1983-9）
- ( 6)都島，三森，薦田：回転倉庫と固定倉庫を結合した自動倉庫システムの運用方式の提案，電気学会論文誌C，Vol.107，No.9，pp843-849（昭和62年9月）
- ( 7)都島，薦田：二段階自動倉庫システムの運用方式の提案，昭和61年電気学会全国大会，No.1477，pp1929-1930（昭和61年4月）
- ( 8)S.Mitsumori， T.Sato and I.Tsushima：A New Automatic Warehouse System with High Picking Efficiency，ORSA/TIMS/AIIE Joint National Meeting（1972-11）
- ( 9)西川：最適化，岩波書店，pp127-134（昭和57年9月）
- (10)都島，三森，鍋島，今西：自動倉庫システム設備計画プログラムCAD-WH，日立評論，Vol.59，No.12，pp1029-1034（昭和55年12月）
- (11)都島，園塚，三森：自動倉庫システムの計画技法，昭和50年電気学会全国大会，pp2049-2050（昭和50年4月）

## 第5章 解法戦略ノウハウの記述にも知識工学を応用した作業者の作業計画方式

## 第5章 解法戦略ノウハウの記述にも知識工学を応用した作業者の作業計画方式

### 5.1 まえがき

本章では、解法戦略ノウハウの記述にも知識工学を応用し、各解法戦略に基づく組合せ計算はアルゴリズムで対応する応用形態に適した問題として、作業者に対する作業計画問題を取り上げる。

与えられた仕事を限られた資源(人員)により完成させるためには、各資源に、何時、どのような仕事を実施させるかの作業計画を適切に立案することが重要である。この計画問題は資源配分問題と呼ばれ、その解法として、Petri-netを応用した手法、PERT/LOAD手法などが提案されている。しかしながら、資源配分問題は組合せ問題となるため、実用レベルの規模の問題に対して、実用的な時間で解を求められるのは、PERT/LOADぐらいしかないのが現状である。

PERT/LOADでは、まず、資源容量の制約を考慮せず、PERT図と呼ばれるネットワークで表現された仕事間の実施順序の制約、各仕事の仕事量等をもとに、各仕事を最も早く開始できる最早開始時刻に仮に割り付けた負荷山積み計画を立てる。つぎに、各仕事の実施時刻を、納期及び仕事間の実施順序の制約が守れる範囲で遅らせるという仕事の後倒しを行い、負荷が資源容量内に納まるようにする。これを負荷山崩しという。なお、仕事の後倒しは、仕事の実施順序として遅く実施しなければならない仕事から順に、逐次、実行される。

PERT/LOADを実際の資源配分業務に適用して、以下に示す問題点のあることを見出した。

各仕事には、その仕事にとって最適な実施時刻がある。仕事によっては、最適実施時刻が最遅開始時刻のものもある。例えば、発注業務がその1つであり、できる限り締切ぎりぎり望ましい。全ての仕事が最適な実施時刻に開始できれば理想的であるが、そのような作業計画では、一般に、資源容量の制約を満たせない。そこで、負荷を資源容量内に納めるため、仕事の実施時刻の調整が必要となるが、その際、できる限り少ない仕事の実施時刻の調整で済みます(逆にいえば、できる限り最適実施時刻から開始できる仕事を多くすることが望ましい。また、最適実施時刻が最遅開始時刻である仕事の場合、実施時刻を早めるという仕事の前倒しが必要である。さらに、仕事の内容によって、ある仕事の実施時刻の調整は他の仕事の実施時刻の調整より先にすべきというように、仕事間に実施時刻を調整するための優先度があるので、その優先度の高い仕事だけで調整を行うことが望ましい。

しかしながら、PERT/LOADでは、前述した処理内容からわかるように、最適実施時刻という概念がなく、移動させる仕事の優先順序もない。このため、得られた負荷山崩し結果では、ほとんどの仕事の実施時刻は最適実施時刻と異なってしまう。

また、実際の作業計画業務における負荷山崩しでは、実施時刻の前後の調整だけでなく、場合によっては仕事の削除を実施するが、PERT/LOADではこれらを扱えない。

本章では、以上の問題点を解決する、広い分野で実際の計画業務に適用可能な、PERT/

LOADを拡張した資源配分方式を提案する。具体的には、負荷山積みにおいて、各仕事の実施時刻を最適実施時刻に設定する。また、負荷山崩し手段として、仕事の後倒しだけでなく、仕事の前倒し、削除を導入する。また、負荷山崩しにおいて、実施時刻の調整を行う仕事が少ないと済み、且つ、仕事の内容に即した調整が行えるように、各仕事に対する負荷山崩し手段の適用優先度を導入する。さらに、対象の個別事情に柔軟に対応するため、どの仕事にどの負荷平滑化手段をどのような場合に適用すべきかといった負荷平滑化戦略をルールで記述し、ルール記述された戦略に基づく負荷平滑化計算は各負荷平滑化アルゴリズムで行わせる。

以下、5.2で資源配分問題を定義し、5.3でPERT/LOADの拡張点、アルゴリズムの詳細を示し、5.4で負荷山崩し結果の評価を述べ、5.5で提案方式をスーパーマーケットの作業計画業務に実際に適用した事例について述べる。

## 5.2 資源配分問題の定義

資源配分問題は、一般に資源と仕事を与えられたもとで、ある目的関数と制約とを満たすように、各仕事の実施時刻と実施資源を決定する問題である。提案手法が対象とする資源配分問題を以下に定義する。

### (1) 資源配分問題を定義する変数

#### (a) 資源

資源の種類は1種類とし、確保可能な資源容量(仮りに人員とする)の時間分布を  $r(t)$  for  $t_1 \leq t \leq t_2$  とする。なお、 $t_1 \leq t \leq t_2$  は、資源の使用可能時間帯である。

#### (b) 仕事

仕事全体の最早開始時刻、最遅終了時刻(納期)は、資源の使用可能時間帯に合わせて、それぞれ、 $t_1$ 、 $t_2$  とする。個々の仕事は  $j$  によって区別し、仕事の集合を  $J$  とする。各仕事はつぎの属性を持つものとする。

#### (i) 各仕事からみた最適実施時刻: $t_j^*$ , $j \in J$

$t_j^*$  は、他の仕事、資源容量の制約と無関係に決まる、仕事  $j$  にとって最も望ましい実施時刻とする。

#### (ii) 仕事 $j$ の仕事量: $(p_j, q_j)$ , $j \in J$

$p_j$ : 仕事  $j$  の処理に必要な資源容量(人員)

$q_j$ : 仕事  $j$  の処理時間

#### (iii) 仕事の後倒し、前倒し、削除手段の適用優先度: $u_{jk}$ , $j \in J$ , $k = 1, 2, 3$

$u_{j1}$ ,  $u_{j2}$ ,  $u_{j3}$  をそれぞれ、仕事  $j$  に対する仕事の後倒し、前倒し、削除手段の適用優先度とする。 $u_{jk}$  は非負の整数をとり、その整数値が小さいほど、その手段を早く適用すべきことを示す。ただし、 $u_{jk} = 0$  の場合には、その手段を適用しない。各仕事に対する  $u_{jk}$  の値は、全体の仕事との関連で設定される。異なった仕事に対して、同一の優先度を設定してもよく、1つの仕事に対して、複数の手段の適用を許す。

#### (iv) 仕事の後倒し、前倒し、削除手段の適用指定時間帯: $[a_{jk}^L, a_{jk}^U]$ , $j \in J$ , $k = 1, 2, 3$

仕事の後倒し，前倒し，削除手段の適用範囲を， $a^L_k \leq t \leq a^U_k$  以内と指定するものである ( $k = 1, k = 2, k = 3$  の場合が，後倒し，前倒し，削除のそれぞれの指定時間帯に相当する)。

(v) 仕事間の実施順序の規制

仕事は，それを実施する資源とは無関係に仕事間に実施順序の規制を持っており，それはPERT図で記述できる。この仕事間の実施順序の規制からも，各仕事の後倒し，前倒しが可能な時間帯が限定される。その時間帯は，PERT/TIMEで計算される最早開始時刻  $t^E_j$  から最遅終了時刻  $t^L_j$  までに対応する。したがって，仕事の後倒し，前倒し手段の適用可能な時間帯は， $[a^L_k, a^U_k]$  と  $[t^E_j, t^L_j]$  のANDをとった時間帯である。

(2) 資源配分問題の解を表わす変数

資源配分問題とは，各仕事  $j$  の実施時刻  $\tau_j$  を決める問題である。実施時刻を決めることを割り付けと呼ぶことにする。

(3) 評価尺度

負荷が資源容量内に山崩しされたか否かの判定は，次式で行う。

$$g(t) = r(t) - L^{(J)}(t) \geq 0 \quad \text{for all } t \in [t_1, t_2] \quad \dots\dots\dots (5.1)$$

ただし， $L^{(C)}(t)$  は，集合  $C$  に属する仕事の全てを割り付けた時の時刻  $t$  の負荷とする。なお， $r(t) - L^{(J)}(t) < 0$  なる  $t$  が存在する場合には，以下に示す，資源容量と負荷の差の2乗和  $h$  を最小にするように負荷山崩しをする。

$$h = \sum_{k=t_1}^{t_2} \{r(t) - L^{(J)}(t)\}^2 \rightarrow \min. \quad \dots\dots\dots (5.2)$$

5.3 前倒し，削除機能の実現方式

5.3.1 PERT/LOADアルゴリズムの概要

PERT/LOADアルゴリズムでは，PERT/TIMEを使い，まず，資源容量  $r(t)$  を考慮せず，各仕事の処理時間  $q_j$ ， $j \in J$  と仕事間の実施順序の規制をもとに，各仕事の実施時刻  $\tau_j$  を最早開始時刻  $t^E_j$  に割り付けた負荷山積み計画を立てる。この計画では，資源容量の制約を考慮していないため，一般に，負荷は資源容量内に納まらない。そこで，負荷が資源容量内に納まるように，仕事間の実施順序に規制が守れる範囲内で仕事の後倒しを行う。仕事を後倒ししても仕事間の実施順序の規制を守ることができる余裕時間は自由フロート  $FF_j$ ， $j \in J$  と呼ばれ，PERT/TIMEによって求めることができる。PERT/LOADでは，仕事の実施順序の遅い仕事から順に，自由フロートの範囲内で実施時刻を後倒しし，負荷と資源容量の差の2乗和  $h$  を最小にする時刻に実施時刻を変更する。ある仕事の実施時刻を変更したら，その仕事に先行する仕事の  $FF_j$  を変更する。なお， $FF_j = 0$  の仕事は本処理をスキップする。各仕事に対する後倒しの適用順序について，もう少し詳しく以下に述べる。

仕事  $j$  は，PERT図では，始めノード番号  $n^S_j$  と終りノード番号  $n^E_j$  のペアで一意的に定義できるように表現される。また，仕事の実施順序の遅い仕事とは， $n^E_j$  が大きい仕事となるよう，全ての  $j$  に対して， $n^S_j < n^E_j$  となるように，ノード番号が付けられる。同一

の  $n^E_j$  を持つ仕事が複数個ある場合は、 $n^S_j$  の大きい仕事から順に後倒しする。

全仕事に対して、一通り、後倒し処理がなされると、再度、同様の処理が何回か実施される。h の改善がある値以下になった時点で処理を終了する。

### 5.3.2 負荷山崩し戦略の記述法

負荷山崩し手段をどの作業に、どのような場合に適用すべきかといった負荷山崩し戦略(方針)は、計画対象によって異なり、変更も多い。また、これらの方針はベテラン計画者がノウハウとして持っている。そのため、ノウハウを容易に計算機化でき、且つ、ノウハウの変更に対して柔軟に対応できる知識工学に着目し、それらのノウハウをルールで記述する方式とした。その記述例を図5.1に示す。IF部は、負荷山崩しの条件を示し、その条件は負荷が資源容量内に納まらない場合である。THEN部はその条件を満たした場合の適用すべき負荷山崩し方針を記述する。

[ ] 内はパラメータ部を示し、負荷山崩し対象とする仕事名、負荷山崩し手段の適用指定時間帯( $a^L_{jk}$ ,  $a^U_{jk}$ )、適用すべき負荷山崩し手段、その適用優先度( $u_{jk}$ )を記述する。なお、記述

IF(負荷が資源容量内に納まらない)

THEN(仕事[発注]を[10]時以降[11]時以内で

[前倒し];適用優先度[3])

図5.1 負荷山崩し方針のルール記述例

性を高めるため、パラメータ部以外は固定とした基本文型を用意した。このルールは、専用のルールエディタを用いて容易に確認、修正できるようにした。

### 5.3.3 前倒し、削除機能を実現するためのPERT/LOADアルゴリズムの拡張

(1) 負荷山積み計画では、各仕事の実施時刻  $\tau_j$  を、各作業から見た最適実施時刻  $t_j^{opt}$ ,  $j \in J$  に設定する。

(2) 負荷山崩し手段の適用に関する情報  $u_{jk}$ ,  $[a^L_{jk}, a^U_{jk}]$ ,  $j \in J$ ,  $k = 1, 2, 3$  は外部から指定する。

(3) 負荷山崩しの際、実施時刻の変更が可能な時間を示す自由フロート  $FF_j$  に着目するが、PERT/LOADのフロートは、後倒し用のフロートである。そこで、前倒し用の自由フロート  $LFF_j$  を導入する(従来のフロートは、 $REF_j$  とする)。LFF<sub>j</sub> は、従来のフロートの計算式で時間軸を逆にすれば求められる。

(4) 仕事によっては、できる限り、後倒し、前倒し、削除を実施したくないものがある。これを考慮するため、仕事に対する各負荷山崩し手段の適用優先度  $u_{jk}$  を導入する。そして、負荷山崩しでは、まず、優先度が最も高い(実施時刻の変更を最も許してもよい)仕事群のみで負荷山崩しを行う。その結果、(5.1)式に示した評価尺度  $g(t) \geq 0$  for all  $t \in [t_1, t_2]$  を満たさなければ、 $u_{jk}$  にもとずき負荷山崩し対象の仕事の範囲を段階的に拡大し、(5.2)式に示した評価尺度  $h$  のもとで、負荷山崩しを実施する。評価尺度  $g(t) \geq 0$  for all  $t \in [t_1, t_2]$  を満足した時点で、処理を終了する。

(5) (4)において、優先度の同じ仕事が多数個あった場合、あるいは、優先度の同じ負

荷山崩し手段が複数個あった場合、どの仕事、どの手段から適用するかについて、以下に述べる。後倒し適用の仕事では、PERT/LOADと同じように、終りノード番号  $n^E_j$  の最も大きい仕事を最優先する。なお、 $n^E_j$  が同一の仕事が複数個ある場合には、PERT/LOADと異なり、後倒し用の自由フロート  $RFF_j$  が小さい仕事を優先する。これは、逐次、解を決定する計算法では、一般に、自由度の小さい手を打っていくことが、全体として適切な解を得られるためである。一方、前倒し適用の仕事に対しては、仕事の実施時刻を早めていくため、始めノード番号  $n^S_j$  の最も小さい仕事を最優先する。 $n^S_j$  が同一の仕事が複数個ある場合には、前倒し用の自由フロート  $LFF_j$  が小さい仕事を優先する。なお、削除は極力避けたいため、一般的には、後倒し、前倒しを削除より優先する。前倒しと後倒しのどちらの手段から適用するかは、それぞれの最優先適用対象の仕事のうち、自由フロートの小さい方を選択する。

#### 5.3.4 アルゴリズム

提案アルゴリズムを以下に示す。

① 負荷山積み計画として、各仕事  $j$  の仕事量  $(p_j, q_j)$ ,  $j \in J$  を最適開始時刻  $t^{opt}$  に割り付ける。また、その時の負荷  $L^{(J)}(t)$  を以下に示す(5.3)式により求める。

$$\tau_j := t^{opt}, j \in J$$

$$\left. \begin{aligned} L^{(J)}(t) &:= \sum_{j \in A^{(J)}(t)} p_j \\ A^{(J)}(t) &:= \{j \mid \tau_j \leq t \leq \tau_j + q_j, j \in J\} \end{aligned} \right\} \dots\dots\dots (5.3)$$

$L^{(C)}(t)$  は、集合  $C$  に属する仕事を割り付けた時の時刻  $t$  の負荷を示す。また、 $A^{(J)}(t)$  は、時刻  $t$  に実施される仕事  $j \in J$  を示す。

② もし、(5.1)式に示す  $g(t) \geq 0$  for all  $t \in [t_1, t_2]$  が成立するならば、処理を終了する。そうでなければ、③に進む。

③ 負荷山崩し手段適用による改善評価のため、 $r(t)$ ,  $L^{(J)}(t)$  for  $t \in [t_1, t_2]$  をもとに、(5.3)式により  $h$  を計算し、これを  $h_0$  にとっておく。

$$h_0 := h$$

④ 仕事の実施順序の規制、 $\tau_j, q_j$  for  $j \in J$  をもとに、各仕事の前倒し用自由フロート  $LFF_j$ ,  $j \in J$ , 後倒し用自由フロート  $RFF_j$ ,  $j \in J$  を計算する。

⑤ 負荷山崩し手段の適用優先度パラメータ  $b$  に初期値として 1 を設定する。

⑥ 後倒し、前倒し、削除対象とする仕事名とその仕事に適用すべき負荷山崩し手段名のペアを求め、それを要素とする集合を  $D$  とする。

$$\begin{aligned} D := & \{(j, 1) \mid u_{jk} \leq b, u_{jk} \neq 0, RFF_j > 0\} \\ & \cup \{(j, 2) \mid u_{jk} \leq b, u_{jk} \neq 0, LFF_j > 0\} \\ & \cup \{(j, 3) \mid u_{jk} \leq b, u_{jk} \neq 0\} \end{aligned}$$

⑦  $D$  の集合のなかから、5.3.2の(5)の方針に従い、負荷山崩し対象とする仕事および負



荷山崩し手段を1つ選択する。これを $(j^*, k^*)$ とする。 $k^* = 1$  (後倒し)の場合, ⑧へ。  
 $k^* = 2$  (前倒し)の場合, ⑨へ。 $k^* = 3$  (削除)の場合は, ⑩へ。

⑧後倒しにする仕事 $j^*$ の変更先の実施時刻 $\tau'_{j^*}$ を次式により求める。そして,  $\tau'_{j^*}$ を $\tau_{j^*}$ に代入し, 自由フロートを変更し, ⑩へ。

$$\tau'_{j^*} := \{s \mid \min_{\tau_{j^*} \leq s \leq \tau_{j^*} + RFF_{j^*}} \{r(t) - L^{(j^*)}(t) - L^{(G^*)}(t, s)\}^2\}$$

ただし,  $L^{(G^*)}(t, s)$  は, 仕事 $j^*$ の実施時刻を $s$ とした時の時刻 $t$ での仕事 $j^*$ の負荷であり,  $s \leq t \leq s + q_{j^*}$ の時,  $p_{j^*}$ となり, それ以外は0である。

⑨前倒しにする仕事 $j^*$ の変更先の実施時刻 $\tau'_{j^*}$ を次式により求める。そして,  $\tau'_{j^*}$ を $\tau_{j^*}$ に代入し, 自由フロートを変更し, ⑩へ。

$$\tau'_{j^*} := \{s \mid \min_{\tau_{j^*} - LFF_{j^*} \leq s \leq \tau_{j^*}} \{r(t) - L^{(j^*)}(t) - L^{(G^*)}(t, s)\}^2\}$$

⑩仕事 $j^*$ の削除後の必要資源容量 $p'_{j^*}$ を次式により求める。そして,  $p'_{j^*}$ を $p_{j^*}$ に代入し, ⑩へ。

$$p'_{j^*} := \{v \mid \min_{0 \leq v \leq p_{j^*}} \{r(t) - L^{(j^*)}(t) - M^{(G^*)}(t, v)\}^2\}$$

ただし,  $M^{(G^*)}(t, v)$  は, 仕事 $j^*$ の必要資源容量を $v$ とした時の時刻 $t$ での仕事 $j^*$ の負荷であり,  $\tau_{j^*} \leq t \leq \tau_{j^*} + q_{j^*}$ の時,  $v$ となり, それ以外は0である。

⑪新たな $\tau_{j^*}$ あるいは $p_{j^*}$ をもとに, 式(5.3)により負荷 $L^{(j^*)}(t)$  for  $t \in [t_1, t_2]$ を計算する。もし, (5.1)式に示す $g(t) \geq 0$  for all  $t \in [t_1, t_2]$ が成立するならば, 処理を完了する。そうでなければ,  $D := D - \{(j^*, k^*)\}$ として⑫へ。

⑫ $D \neq \{\phi\}$ なら, ⑦へ。 $D = \{\phi\}$ なら, ⑬へ。

⑬(5.2)式により $h$ を計算し,  $h_0 - h > \epsilon$ なら,  $h_0 := h$ として, ⑥へ。 $h_0 - h \leq \epsilon$ なら, ⑭へ。

$\epsilon$ は, 改善の度合の下限値を示すパラメータ値である。

⑭ $b = \max_{j \in J, k = \{1, 2, 3\}} u_{jk}$ ならば, 終了。 $b < \max_{j \in J, k = \{1, 2, 3\}} u_{jk}$ ならば,  $b := b + 1$ ,

$h_0 := h$ として, ⑥へ。

#### 5.4 負荷山崩し結果の評価

提案方式では, PERT/LOADに比べ, 適切な負荷山崩しが可能であり, しかも, 各仕事にとっての最適実施時刻を守る計画結果の得られることを例を用いて示す。仕事は, ①~⑧の8種類があるとする。仕事①は, 仕事⑤, ⑥に, 仕事⑤は仕事⑥に先行して実施されなければならないという仕事間に実施順序に対する規則があるとする。また, 各仕事にとっての最適実施時刻に仕事を割り付けた負荷山積み結果を図5.2に示す。これは, 提案方式における負荷山積み結果に相当する。一方, PERT/LOADにおける負荷山積み結果を図5.3に示す。図5.2, 5.3の“→”は, その仕事が後倒し可能であることを示す。

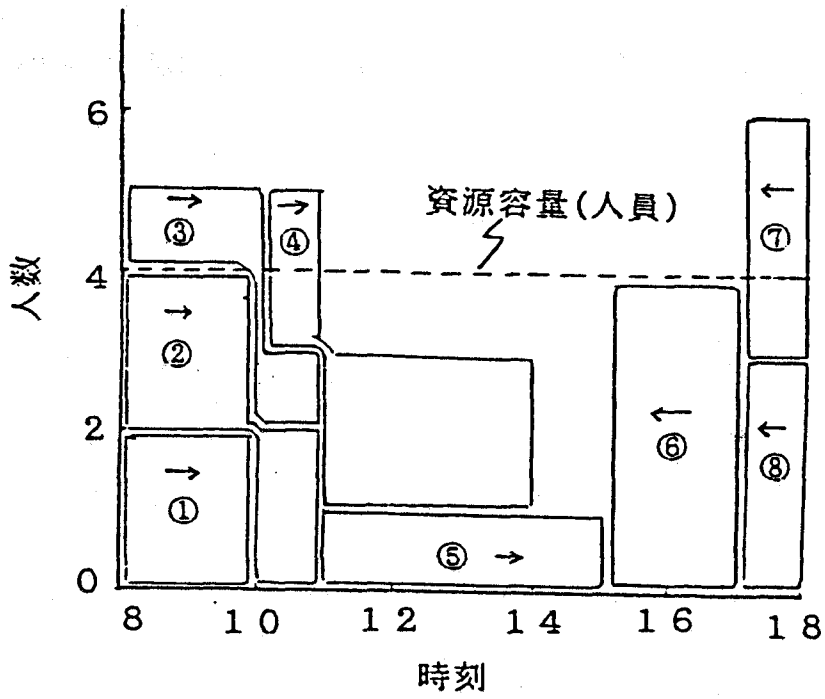


図 5.2 提案方式による負荷山積み結果

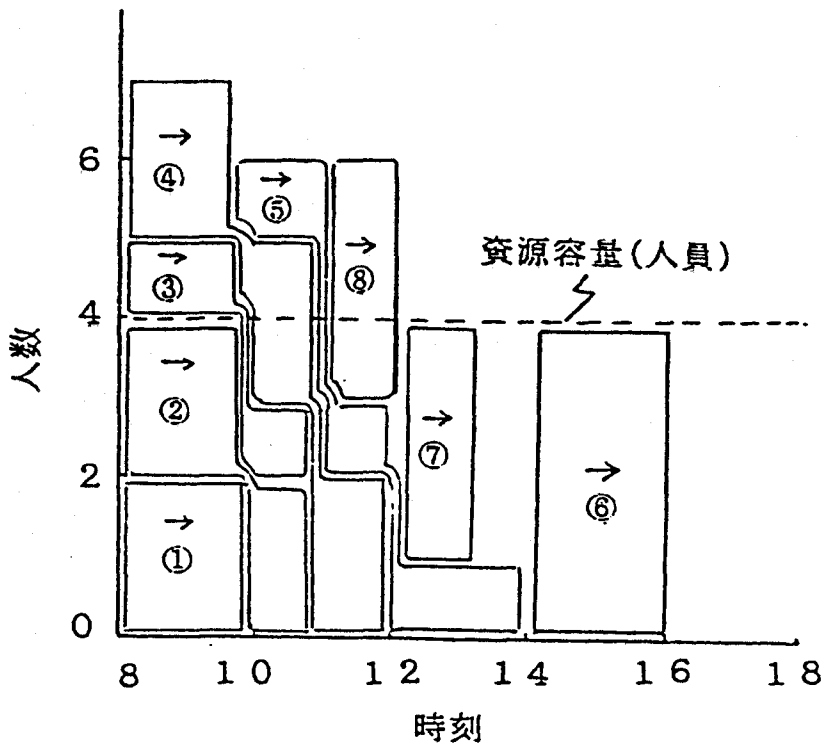


図 5.3 PERT/LOADによる負荷山積み結果

図5.2の“←”は、その仕事が前倒し可能であることを示す(提案方式により前倒し可能な仕事は、⑥、⑦、⑧に相当する)。提案方式において、負荷山崩しの適用に関して指定した情報は以下の通りである。

- ・仕事③；  $[a_{3,1}^L=10, a_{3,1}^U=16], u_{3,1}=1$
- ・仕事⑦；  $[a_{7,1}^L=10, a_{7,1}^U=16], u_{7,2}=2$
- ・仕事②；  $[a_{2,1}^L=10, a_{2,1}^U=16], u_{2,1}=3$

：

提案方式に基づく負荷山崩し結果、PERT/LOADに基づく負荷山崩し結果をそれぞれ、図5.4, 5.5に示す。図中の“→”あるいは“←”は負荷山積み時に割り付けた実施時刻を負荷山崩しで後倒しあるいは前倒ししたものである。提案方式では、負荷を資源容量内に納めることができるが、PERT/LOADではできない。また、提案方式では、負荷山崩しにより、最適実施時刻に実施できなくなった仕事が、③、⑦のみに対して、PERT/LOADの結果では、仕事①のみが最適実施時刻で実施できる。

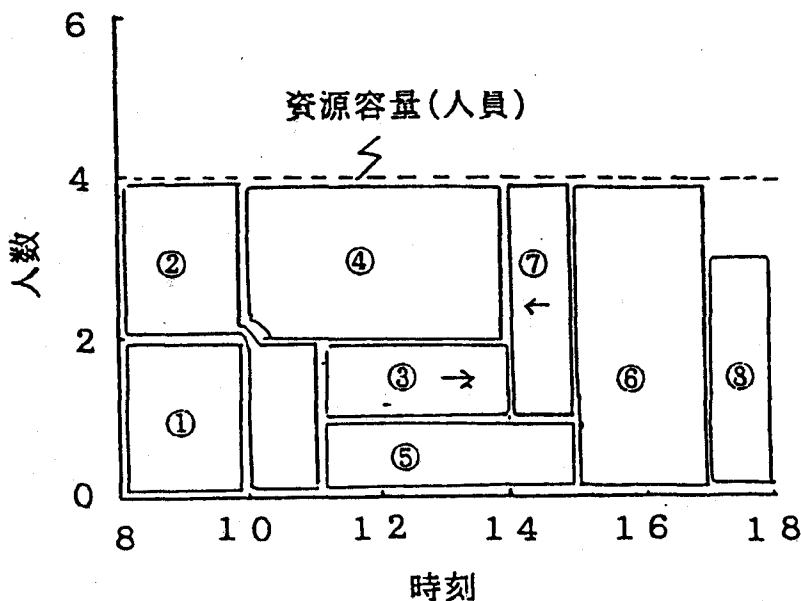


図5.4 提案方式による負荷山崩し結果

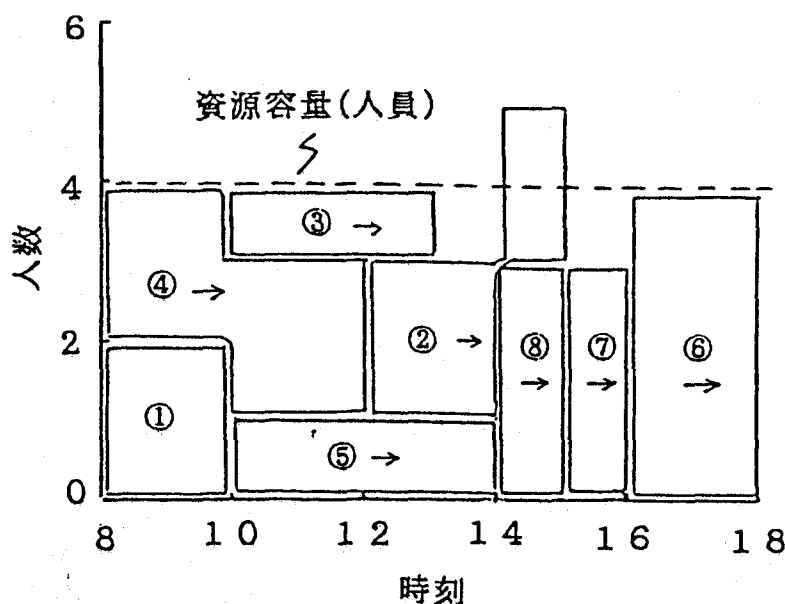


図 5.5 PERT/LOADによる負荷山崩し結果

## 5.5 実システムへの応用例

提案方式を組み込んだ作業計画システムをスーパーマーケットの実際の作業計画業務に適用した。その内容を以下に述べる。

### 5.5.1 流通分野における作業計画システムの必要性

急成長の流通業界においても、消費者構造の変化により、厳しい生残りの時代に入り、多店舗チェーン展開している流通業界（量販店等）では、売上向上、店舗運営効率化のため、POS (Point of Sales) 端末で収集されたデータを有効活用するストアオートメーション化への要求が強まっている。スーパーマーケットなどの量販店において、社員、パートタイム、アルバイトなど複雑な雇用形態の従業員に対して、それぞれに適した作業内容を製造業並にきめ細かく計画することにより、店舗運営の効率化やサービスの向上を図りたいという要求はその一つである。この要求に応えるために、店長のノウハウをルール化し、適正な作業計画を立案する知識工学応用・流通業向作業計画システムを開発し、スーパーマーケットの実際の作業計画業務に適用した。適用対象は以下の通りである。

#### (1) 仕事

仕事は50種類である。また、仕事の処理時間は、15分を最小単位とする。

#### (2) 資源

従業員は社員、パートタイム、アルバイトからなり、全体で約20名である。それぞれの勤務時間帯がまちまちであるため、時間軸上での資源容量は変動する。なお、店舗全体としての勤務時間帯は7:30から22:00までである。

### 5.5.2 作業計画システム

## (1) システム構成

システムの構成を図5.6に示す。明日の条件を設定すると、過去の売上高（POSデータ）、週間勤務予定（勤怠管理データ）、売上高修正ルール、負荷山崩し方針ルールをもとに、売上高予測結果、負荷山崩し結果及び個人別スケジュール表を出力する。POSデータ、勤怠管理データはそれぞれストアコントローラ内のPOSデータ管理部、勤怠管理データ管理部から送られる。ルールは専用のルールエディタを用いて、店長が容易に確認、修正することができるようにした。

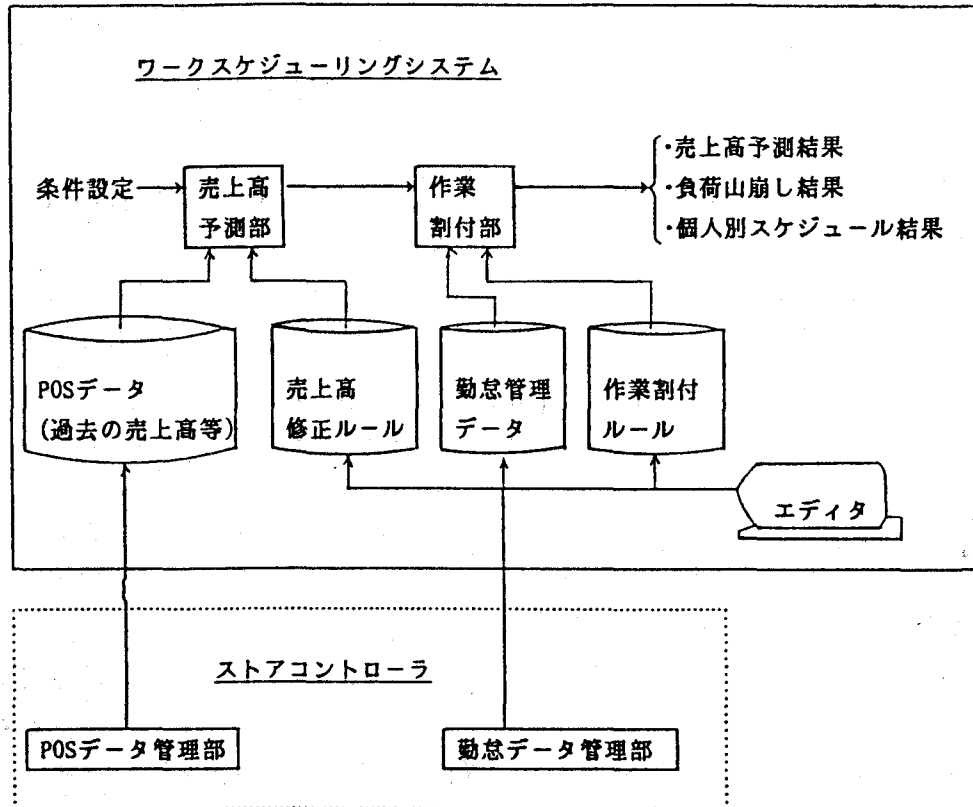


図5.6 システム構成

## (2) ルールの記述内容

明日の作業計画を立案するには、まず、作業量の基本データとなる明日の売上高時間分布を予測する必要がある。その売上高は明日の曜日、天候、催事（運動会、遠足など）、自店や競合他店の特売内容などにより変動する。各変動要因がどの程度の影響を与えるかは、店舗ごとに異なり、店長がノウハウとして持っている。そのため、そのノウハウを売上高修正ルールとしてルール記述することにした（図5.7参照）。明日の売上高時間分布はPOSデータに売上高修正ルールが適用され、求められる。

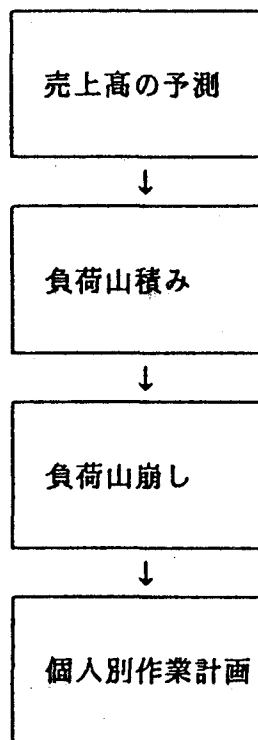
負荷山崩し方針については5.3.2で述べた記述形式でルール化した。

売上高修正ルールと負荷山崩し方針ルールを合せて作成したルール数は約60である。

IF(〔競合他店〕が〔朝市実施〕)  
 THEN(〔10〕時から〔12〕時まで売上高を〔15%減〕)  
 THEN(〔12〕時から〔16〕時まで売上高を〔 5%減〕)

IF(〔天候〕が〔午後:雨〕)  
 THEN(〔午後〕売上高を〔20%減〕)

図 5.7 売上高修正ルール例



### (3) 作業計画立案の流れ

本システムによる作業計画立案の流れを図 5.8 に示す。

店長は明日の曜日、天候、催事(運動会、遠足等)、自店や競合他店の特売内容などを画面(図 5.9)から設定する。システムは過去の部門別、時間帯別の売上高基本データに対して、前述の設定項目である変動要因を加味して、明日の売上高の予測結果を出力する(図 5.10)。

図 5.8 作業計画立案の流れ

つぎに、システムは、売上高の予測値にもとずき、レジ、補充、発注などの負荷を見積り、それぞれの作業に最も適した時刻に仮に割り付ける。割り付けられた負荷山積み結果(図 5.11)が、週間勤務予定データから求められた明日の手持ち人員をオーバする場合には、ルール表現された負荷山崩し方針に基づき、負荷を人員以内に納め、負荷山崩し結果(図 5.12)を出力する。

以上の負荷山崩しにより求められた作業計画は、チーフ、社員、パートタイマ等、職種単位の計画であり、最終的には、個人別の作業計画が必要である。そこで、システムは、前述の作業計画結果をもとに、個人ごとの特性を考慮し、個人別作業計画表を作成、出力する(図 5.13)。

### 5.5.3 適用結果

提案方式を適用した結果、以下のことを確認した。

- (i) 従来、作業計画に専門家が人手で約 2 時間かかっていたが、提案方式を組み込んだ 16 ビットのワークステーション上の作業計画システムでは約 5 分で処理できた。
- (ii) 作業計画の専門家より質の高い作業計画結果を得ることができた。
- (iii) 負荷山崩し方針ルールの変更が容易なため、店舗の個別事情に即した作業計画システムを短時間で構築することができた。

設定  
中止

# 外部条件入力

[1] 売り上げ予測日  
S62年 9月12日 土曜日

[2] 予測日の条件を推定してください。

[天候:午前]	:	[晴れ [雪]	:	[曇]	:	[雨]
[天候:午後]	:	[晴れ [雪]	:	[曇]	:	[雨]
[特売]	:	[朝市]	:	[レジャー]	:	
[他店特売]	:	[朝市]	:	[レジャー]	:	
[催事]	:	[運動会]	:	[遠足]	:	



an

図5.9 条件入力画面



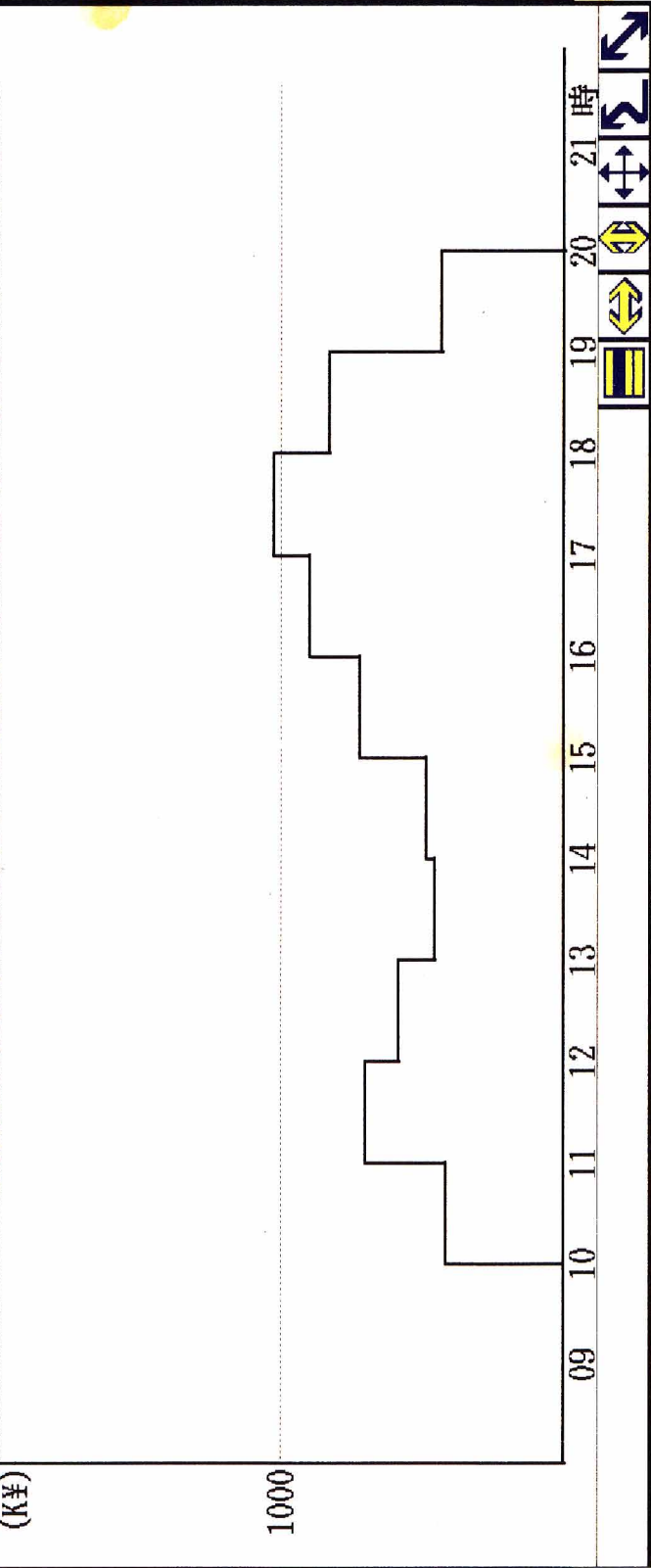
# 売上予測

予測結果をグラフにて出力します  
いずれの予測かを選択して下さい。

売上点数 来客数

金額補正  
点数補正  
客数補正  
設定

金額 (K円) 予測日 S62年09月12日 土曜日 レジ部門



an

図5.10 売上高予測結果画面



作業ルール

# 作業山積み

部門

S62年 9月 12日 土曜日

- : 出勤者

戻り  
山崩し  
追加  
一覧

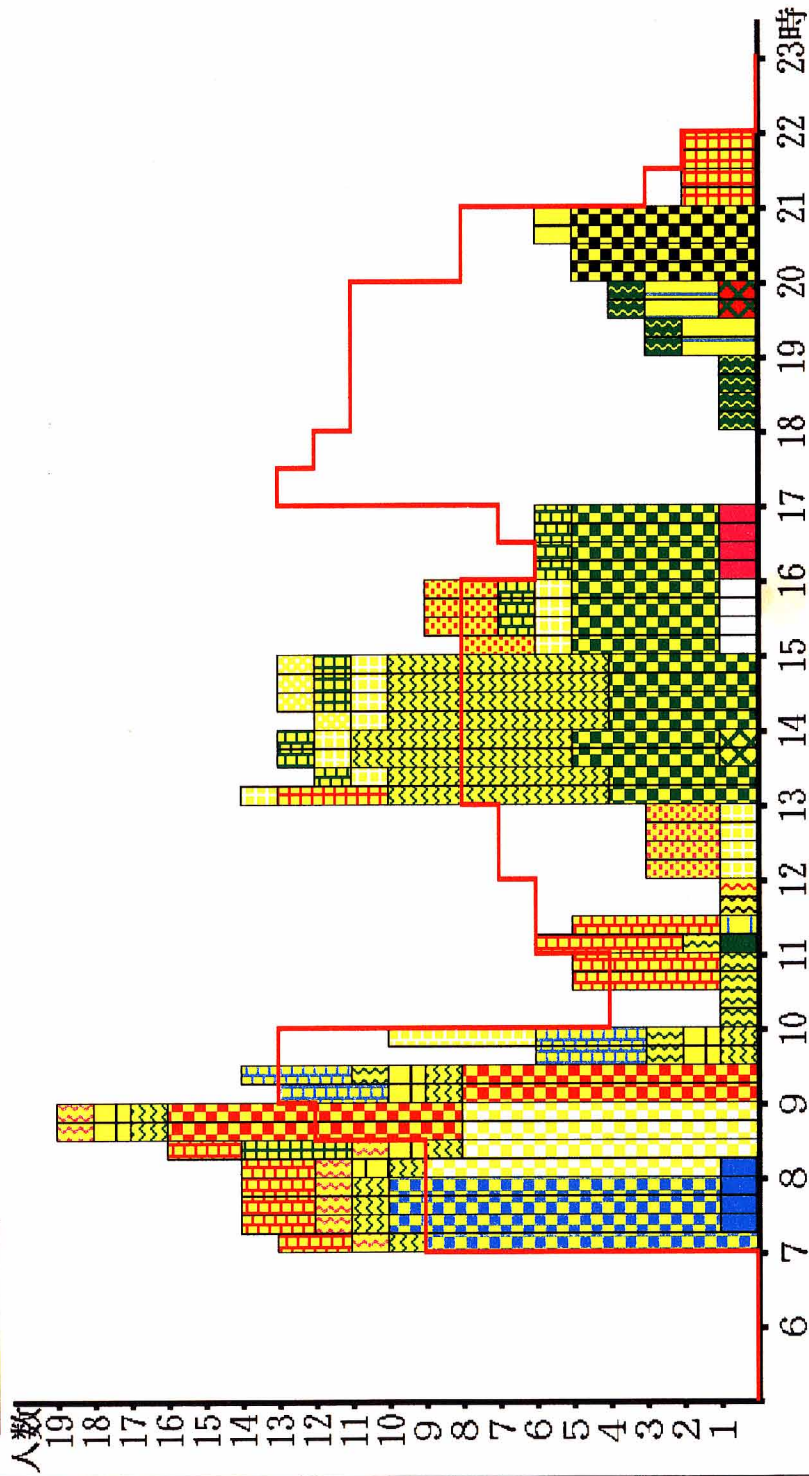


図 5.1.1 負荷山積み結果画面

an

山積み 作業ルール 作業優先順位

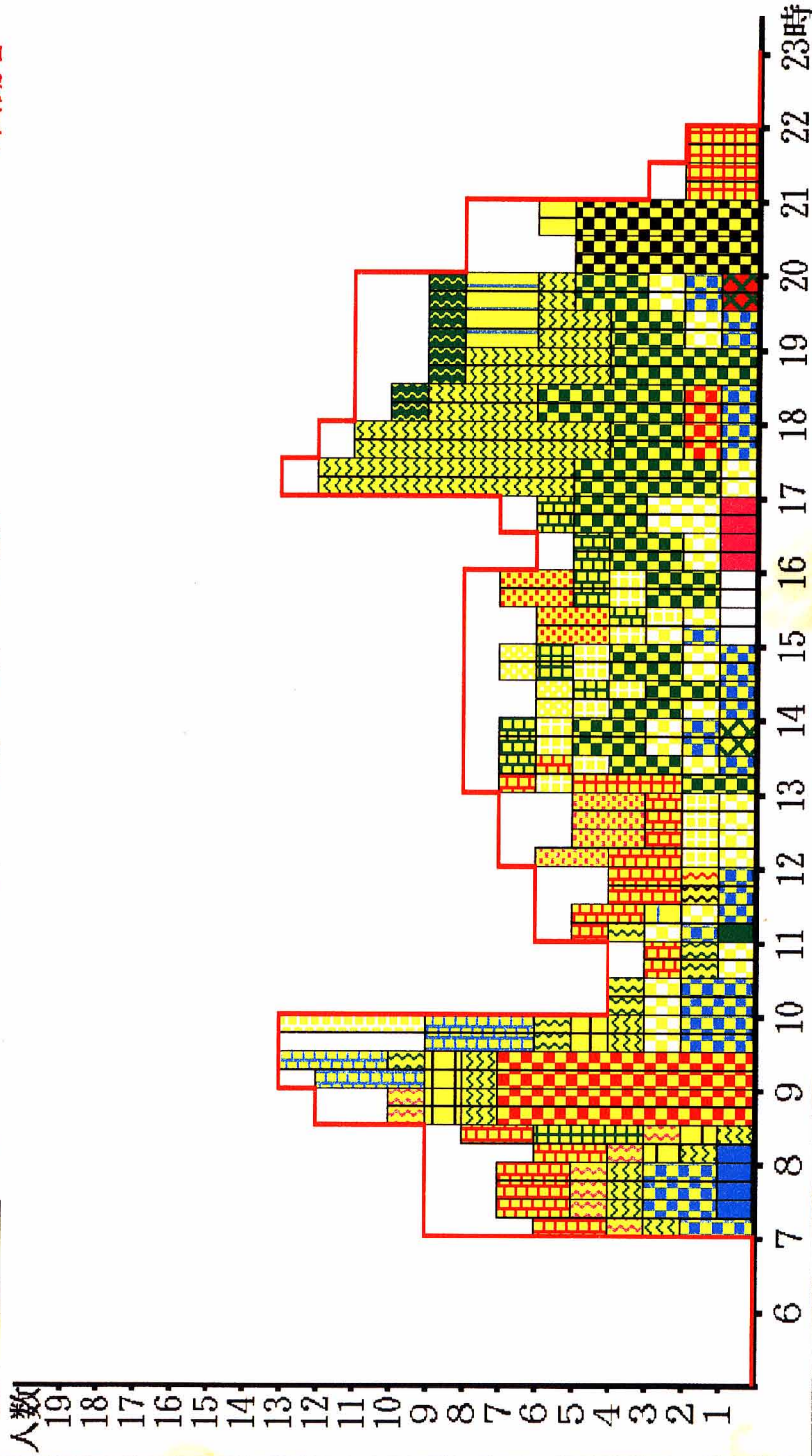
# 作業山崩し

グロサリー 部門

S62年 9月 12日 土曜日

- :: 出勤者

ソフト  
カット  
戻り  
割付け  
一覧



an

図5.12 負荷山崩し結果画面

# 個人別スケジュール表

グロサリー

部門

S62年 9月 12日 土曜日

追加	戻り
ソフト	一覧
カット	リスト
応援	

氏名	区	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
吉沢 泰弘	社						BR在												
横山 和裕	社						定番	定番	定番	定番	定番	定番	定番	定番	定番	定番	定番	定番	定番
藤崎 きよい	定						定番	定番	定番	定番	定番	定番	定番	定番	定番	定番	定番	定番	定番
小松崎 也恵子	定						定番	定番	定番	定番	定番	定番	定番	定番	定番	定番	定番	定番	定番
戸ノ岡 和子	定						定番	定番	定番	定番	定番	定番	定番	定番	定番	定番	定番	定番	定番
斉藤 恵子	準						4発	6BR在	庫出	荷	食事	イ	ト	補	充				
吉原 香代	準						定番	定番	定番	定番	定番	定番	定番	定番	定番	定番	定番	定番	定番
雨貝 繁和	ア																		

売場イ

an

図 5.13 個人別作業計画結果画面

## 5.6 まとめ

第5章では、解法戦略の記述にも知識工学を応用し、各解法戦略に基づく組合せ計算法はアルゴリズムで対応すべき運用制御問題を、作業計画問題を対象に論じた。明らかにした事項は次の通りである。

- (1) PERT/LOADを拡張し、広い分野で実際の作業計画業務に適用可能な、以下の特長を持つ資源配分手法を提案した。負荷山積みにおいて、各仕事の実施時刻を仮りに最適実施時刻に設定する。負荷を資源容量内に納めるための負荷山崩しでは、その手段として、仕事の後倒し(仕事の実施時刻を遅らせる)以外に仕事の前倒し(仕事の実施時刻を早める)、削除を実現し、且つ、それらの手段の適用優先度を導入することによって、実際の業務に即した作業計画を可能とした。また、各負荷山崩し手段をどの仕事に、どのような場合に適用すべきかといった負荷山崩し方針を記述、変更の容易なルールにより指定できるようにしたため、対象の個別事情に即した作業計画システムを短時間で構築することができる。
- (2) 提案方式を組み込んだ作業計画システムを実際のスーパーマーケットの作業計画業務に適用した。この適用を通じて、作業計画結果の質の向上、作業計画時間の短縮などの効果を確認した。

現在、以上の提案方式を更に拡張し、ソフトウェアの開発プロジェクトの作業計画を作成する<sup>13)</sup>方式<sup>12)</sup>についても研究開発を進めている。この作業計画業務では、標準的な作業負荷のもとでの作業計画では納期を満たせない場合、ある仕事に人員を投入して、その仕事の開発期間を短縮することにより、全体の作業計画が納期を満たすようにし、且つ、負荷山崩しを実現する手段が更に必要である。この手段を組み込んだソフトウェア開発プロジェクト計画パイロットシステムを開発し、現在、その有効性の評価のため、実際の計画業務で試用中である。また、処理設備しか扱えないPERT/LOADを拡張して、ストレージも扱えるようにした、物流基地一般に適用可能な汎用的設備計画手法を開発<sup>13)</sup>している<sup>16)</sup>。



## 参考文献

- (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, pp66-71 (昭和62年1月)
- (2)刀根:パート講座1,東洋経済新報社(昭和52年7月)
- (3)都島,薦田:負荷山崩し機能を拡張した資源配分手法の提案(電気学会論文誌Cに投稿中)
- (4)都島,薦田,楠崎,長谷川:流通業向けエキスパートシステム,日立評論,Vol.69, No.3, pp259-262 (昭和62年3月)
- (5)山下,都島,林,朝倉,青山:POSシステムを中心としたストアオートメーションシステム—システム ソフトウェア技術の側面から—,日立評論,Vol.68, No.12, pp945-950 (昭和61年12月)
- (6)都島,薦田,八木,楠崎:知識工学応用スーパー向ワークスケジューリングシステム,情報処理学会第33回全国大会,2L-4, pp1189-1190 (昭和61年22月)
- (7)都島,薦田,楠崎,井下田:知識工学応用・流通業向ワークスケジューリングシステム,第5回SICE知識工学シンポジウム予稿集, pp51-54 (昭和62年3月)
- (8)都島,薦田,楠崎,井下田:流通業向・ワークスケジューリングシステムにおける作業割付方式,情報処理学会第34回全国大会,2G-4, pp2161-2162 (昭和62年3月)
- (9)中田,都島,他:流通業向ワークスケジューリングシステムにおける個人別作業計画方式,情報処理学会第36回全国大会,4J-1, pp2255-2256 (昭和63年3月)
- (10)大場,都島,薦田,山中:知識工学を応用したプロジェクト計画方式,情報処理学会第35回全国大会,7Y-2, pp1131-1132 (昭和62年9月)
- (11)大場,都島,薦田,山中:知識工学を応用したプロジェクト進捗管理方式,情報処理学会第36回全国大会,3N-4, pp1103-1104 (昭和63年3月)
- (12)大場,都島,薦田,山中:プロジェクト管理支援エキスパートシステム,第14回システムシンポジウム, pp215-218 (昭和63年8月)
- (13)大場,都島,岩本:ストレージを持つ離散システムへのPERT/LOADの拡張アルゴリズム,電気学会論文誌C, Vol.104, No.3, pp49-56 (昭和59年3月)
- (14)I.Tsushima, S.Mitsumori, M.Ooba, T.Iwamoto and H.Furuyado: An Extension of Resource Allocation Problem, Proc. of the 1980 Int. Conf. on Cybernetics and Society, pp253-257 (1980-12)
- (15)I.Tsushima, S.Mitsumori, K.Akashi, T.Iwamoto and M.Ooba: A Resource Allocation Method and An Identification Method for Discrete Systems, Proc. of the 8th Triennial World Congress of the Int. Federation of Automatic Control, pp3573-3578 (1981-8)
- (16)都島,岩本,大場:資源配分問題の拡張と物流システムへの応用,電気学会,システム・制御研究会, SC-80-17, pp11-19, (昭和55年9月)

## 第6章 結 論

## 第6章 結 論

本論文では、知識工学の応用形態を3種類の形態で捉え、物流システムの運用制御問題を論じた。

第一の応用形態は、問題と解法の記述に知識工学を応用するもので、事例として流れ作業ラインの物流制御問題を取り上げた。本物流制御は実時間制御であり、知識工学の実時間制御への応用では、プロセスとの簡易なインターフェース方式が必要である。この実現方式として、文字列、パラメータを与えるだけでトラッキングテーブルから推論機構のワークテーブルへの設備現況書き込み、推論機構で決定された制御指令の読み出しを可能とする方式が有効であることを実システムへの適用を通じて示した。また、ルール記述されたプログラムを人間の介在の許されない閉ループコントロールの中で動かさなければならない実時間制御では、実稼動前に条件そのものの抜けやルールそのものの抜け等の論理自体のチェックが確実にできる方法が必要である。その実用的な方法として、人間が介在するが、排他の条件に着目し、条件と結論との相互関係の構造化を図る方法を提案し、その有効性を実システムへの適用を通じて確認した。さらに、ビレット精整ラインへの物流制御への知識工学の適用を通じて、組合せの数が比較的少ない問題では、問題と解法の両者をルールで記述しても、十分に実時間性を満足すること、制御ソフトの開発工数は、従来の手続き型の場合に比べ、約1/3に低減できることが判った。

第2の応用形態は、問題の記述に知識工学を応用し、組合せ計算はアルゴリズムで対応するもので、事例として、無人搬送車の運行制御問題とピッキング効率向上運用問題を取り上げた。まず、無人搬送車の運行制御問題では、その問題を条件判定的な問題と組合せ最適化的な問題に分離する構造を提案した。その条件判定的な問題では、パラメータの値の変更では対応できない、作業ステーションへ物を搬送できるか否かの搬送可否条件、どの搬送要求から優先的に搬送すべきかの搬送優先条件などが頻繁に変更されるため、それらの条件はルールによる記述方式とした。一方、分離された組合せ最適化的な問題は、搬送要求に対する搬送車割付の問題であり、ほとんど変更が発生しないため、これに対して、高効率で標準化された搬送車割付アルゴリズムを提案した。また、提案方式により変更が容易にできることを特急処理運用の追加を例に示した。さらに、提案した搬送車割付アルゴリズムは、従来のアルゴリズムに比べ、効率が向上できること(同じ処理量/時間に対して、必要搬送車台数が少なくて済むこと)をシミュレーションにより示した。提案方式を組み込んだ運行制御システムは、すでにプリント基板組立ショップなどに適用されており、その制御ソフト開発工数の削減、その後のメンテナンスの容易性といった効果が確認されている。

第2の応用形態のもう1つの事例であるピッキング効率向上運用問題に対し、全体の注文のある制約条件、例えば納期条件下で順序を並び換えてよい注文群に分割するという条件判定的な問題は対象により異なり、変更も多いため、ルールで記述する方式を提案した。一方、分割された各注文群に対してピッキング作業が無作業にならないように注文の処理順序を決定する順列組合せ問題はアルゴリズムで対応する方式を提案した。自動倉庫シ

システムを、ピッキング用回転倉庫と、それにパレットを供給するためのクレーンを持つ固定倉庫からなる二段階構成とし、ピッキング効率を向上させるための2種類の運用アルゴリズムを提案した。そのアルゴリズムは、事前に判っている注文内容のデータを利用し、処理順序として近い注文ほど共通品目が多くなるように注文処理順序を決定する運用アルゴリズムと、決定された注文処理順序のもとで、回転倉庫上には近い将来再度ピッキングされるパレットが格納されるように、回転倉庫から固定倉庫へ払出すパレットを決定する運用アルゴリズムである。また、提案アルゴリズムによるピッキング効率と、その効率に影響を与える注文、回転倉庫、固定倉庫のパラメータとの関連をシミュレーションにより明らかにした。その結果、提案方式により高いピッキング効率を実現できる見通しを得た。

第3の応用形態は、解法戦略の記述にも知識工学を応用し、各解法戦略に基づく組合せ計算はアルゴリズムで対応するもので、事例として作業計画問題を取り上げ、PERT/LOADを拡張し、広い分野で実際の作業計画業務に適用可能な、以下の特長を持つ資源配分手法を提案した。負荷山積みにおいて、各仕事の実施時刻を仮りに最適実施時刻に設定する。負荷を資源容量内に納めるための負荷山崩しでは、その手段として、仕事の後倒し(仕事の実施時刻を遅らせる)以外に仕事の前倒し(仕事の実施時刻を早める)、削除を実現し、且つ、それらの手段の適用優先度を導入することによって、実際の業務に即した作業計画を可能とした。また、各負荷山崩し手段をどの仕事に、どのような場合に適用すべきかといった負荷山崩し方針(解法戦略)を容易なルールにより指定できるようにしたため、対象の個別事情に即した作業計画システムを短時間で構築することができる。なお、提案方式を組み込んだ作業計画システムを実際のスーパーマーケットの作業計画業務に適用した。この適用を通じて、作業計画結果の質の向上、作業計画時間の短縮などの効果を確認した。

今後、ますます多種少量生産化が進み、物の流れが複雑になる傾向にある。そのため、各種変更迅速に対応でき、かつ、効率的な物流運用制御の実現を可能とする知識工学と数理工学的アルゴリズムの融合アプローチが一層有効になってくると考える。本論文では、そのアプローチに基づく物流運用制御方式について述べたが、そのアプローチをさらに効果的にするには、今後、以下に示す課題の解決が望まれる。

(1)運用制御ノウハウについてはあいまい性を含むものもあり、本アプローチの適用範囲を更に拡大するにはF U Z Z Yの技術の導入も必要と考える。

(2)物流機能は大きく搬送機能と保管機能に分けられ、搬送に関する問題についてはその問題を知識工学で対応すべき部分と数理工学的アルゴリズムで対応すべき部分に分割する汎用的な方法を3章に示した。しかし、保管に関する問題の汎用的な分割の方法については今後の課題である。

(3)アルゴリズム部分を対象毎に作成するのでは開発に工数がかかる。これを解決するには、広範囲の問題に対応できる、数種のアルゴリズムを用意する必要があり、5章で述べた、負荷平滑化を目的として作業スケジュールを決めるPERTタイプのアルゴリズムと、作業スケジュールが与えられたもとで資源を割り付ける割付問題向きタイプのアルゴリズムで対応できると考えるが、各種事例をもとにこの点を明確にする必要がある。



## 謝 辞

本研究の推進に当って、下記の多くの御理解、御指導、御援助を賜った。

本論文をまとめるに際しては、大阪大学工学部 藤井克彦教授、鈴木胖教授、児玉慎三教授、田村坦之教授より、終始有益かつ懇なる御指導と御鞭達を賜った。

本研究の遂行において、(株)日立製作所 専務取締役 三浦武雄博士、同社・情報事業本部 次長 川崎淳博士、同社・システム開発研究所 所長 堂免信義氏、主管研究長 三巻達夫博士、副所長 春名公一博士、主管研究員 三森定道博士には、研究の機会を与えて頂くとともに、暖かい御指導と御鞭達を賜った。(株)日立製作所 システム開発研究所 部長 中尾和夫博士、同研究所 大森分室長 明石吉三博士、主任研究員 松本邦頭博士、主任研究員 薦田憲久博士には、常に有益なる御指導と御鞭達を賜った。特に、薦田憲久博士の御指導がなくては、本論文は纏められなかったと考える次第で、格別の感謝の意を表すものである。

第2章に係る研究では、川崎製鉄(株)水島製鉄所 システム部部长 斎藤達氏、課長 山崎順次郎氏、掛長 馬場和史氏に提案方式の実システムへの適用の場を与えて頂いた。また、(株)日立製作所 大みか工場 副工場長 出村吉晴氏、主任技師 高倉修一氏には、本研究に関し、有益な御討議を頂き、提案方式の適用に御協力頂いた。

第3章に係る研究では、(株)日立製作所 機電事業本部 プラント建設本部長 鍋島康夫氏、同社・笠戸工場 主任技師 古宿英機氏、主任技師 岡坂建一氏に御指導と御鞭達を賜った。また、(株)日立製作所 神奈川工場 主任技師 小島明夫氏、技師 原田治氏には、提案方式の実システムへの適用に御協力頂いた。

第4章に係る研究では、(株)日立製作所 機電事業本部 プラント建設本部長 鍋島康夫氏、同社・システム事業部 事業部長 油井兄朝博士、部長 橋本友孝氏、主任技師 山室幸之助氏、同社・機電事業本部 副技師長 並木浩氏、主任技師 桐生隆久氏に御指導と御鞭達を賜った。

第5章に係る研究では、(株)カスミ 取締役 畑木義昭氏、石井俊樹氏、皆川渡氏、根本弘氏に実際の業務での問題点について御討論頂き、提案システムを実際の店舗で適用して頂いた。また、(株)日立製作所 大森ソフトウェア工場 副工場長 青山義彦氏、部長 藤松彌壽雄氏、部長 内田正則氏、主任技師 楠崎哲生氏、主任技師 加茂隆氏、主任技師 西本達美氏、技師 井下田正二氏、技師 八木正雄氏、同社・情報事業本部 主任技師 長谷川一行氏に御指導、御鞭達、提案方式の実システムへの適用に御協力頂いた。

また、本研究の遂行において、田代勤氏、大場雅博氏、大場みち子氏、中田英樹氏を始め、(株)日立製作所のシステム開発研究所、大森ソフトウェア工場、大みか工場、笠戸工場、システム事業本部、機電事業本部、神奈川工場の多くの方々のご援助を頂いた。

ここに、以上の方々に深甚な謝意を表する次第である。