

Title	ルール・マイニング型生産スケジューリング方式に関する研究
Author(s)	東崎, 秀行
Citation	
Issue Date	1999-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1284
Rights	
Description	Supervisor: 藤田 政之, 情報科学研究科, 修士

修 士 論 文

ルール・マイニング型生産スケジューリング方式 に関する研究

指導教官 藤田政之 助教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

東崎秀行

1999 年 2 月 15 日

目 次

1	はじめに	1
1.1	研究の背景と目的	1
1.2	本論文の構成	3
2	準備	4
2.1	生産システムのモデル	4
2.1.1	使用する変数	4
2.1.2	評価基準	5
2.1.3	ディスパッチング・ルール	6
2.2	データ・マイニング	7
2.3	アプリアリアルゴリズム	9
3	ルール・マイニングの導入	17
3.1	シミュレーションモデルの基本仕様	17
3.2	ルール・マイニング適用への準備	19
3.2.1	「同等」の導入	19
3.2.2	「同等」の評価	20
3.2.3	問題の設定	22
4	ルール・マイニングの適用例	25
5	ルール・マイニングの評価	30
5.1	ジョンソン則との比較	
	—2マシン2工程フローショップ問題—	30

5.1.1	ジョンソン則	30
5.1.2	シミュレーションモデルの基本仕様と出力	31
5.2	5 マシン 5 工程ジョブショップ問題	34
5.2.1	シミュレーションモデルの基本仕様	34
5.2.2	複合ルールへの候補とシミュレーション結果	35
5.3	複合ルールの重み付けへのルール・マイニング法の適用	36
5.4	ルール・マイニング法の適用 - その 1 -	40
5.5	ルール・マイニング法の適用 - その 2 -	42
6	おわりに	49
6.1	まとめ	49
6.2	今後の展望	50

目 次

2.1	モデル概念	5
2.2	アイテム集合，トランザクションの関係	10
2.3	二つの入力値を満たす相関ルールの導出	15
2.4	アプリアリアルゴリズムのインターフェースと入出力例	16
3.1	1マシン，2ジョブ，2ディスパッチング・ルールの場合	21
3.2	相関ルール出力例	24
4.1	トランザクション	27
4.2	最大納期遅れサポートグラフ ($\beta = 20$)	29
5.1	ジョンソン則での投入順序例と工程処理例	32
5.2	最大滞留時間サポートグラフ ($\beta = 0$)	33
5.3	ARENA によるシミュレーション実行例	37
5.4	滞留時間，納期遅れ時間からトランザクションの作成	39
5.5	トランザクションの内容	39
5.6	組み合わせ (2,2,2,2,2) の尺度 β, γ とサポート値の関係	43
5.7	新しい尺度 δ の指す領域	45
5.8	二つの評価基準の「同等」の意	45
5.9	平均納期遅れ，平均滞留時間の差異分布図 原点：(3,3,3,3,3)	48

表 目 次

2.1	アルゴリズムで用いる集合	11
2.2	アプリアリアルゴリズムの実行例	14
3.1	2 マシンジョブショップモデルの工程設計	18
3.2	2 マシンジョブショップモデルの工程処理時間	18
4.1	納期係数 α 毎の最大納期遅れサポートデータ ($\beta = 20$)	28
5.1	各々の相関ルールのサポート値 (小数点以下 3 位切り捨て)	38
5.2	重み係数毎の評価基準の出力	40
5.3	使用する複合ルール	41
5.4	組み合わせルール数	47
5.5	(5.5.2) 式の使用, 未使用の同等の数	47

第 1 章

はじめに

1.1 研究の背景と目的

近年，製造企業がさらに成長，発展を遂げていくためには，多様化していく顧客の要求に応えるため，タイムリーに，そして低いコストで提供できる生産システムが求められている．そして文献 [1][4] によれば，生産活動が現実に行われている場においては，その生産の効率化および質的向上を目的とするスケジューリング業務が何らかの形で存在している．生産形態が単純で規模が小さいときには，たとえ組織の中で業務が明示的な形で存在していないように見えても，現場統括者の頭の中では，より効率的な生産のためのやりくり / 手配計画が日常的に描かれスケジューリング業務は存在している．一方，生産形態が多品種化などにより複雑化し，規模が大きくなったときには計画立案を組織的に行うことになる．ゆえに，生産システムを論ずる場合には生産スケジューリングを抜きにしては考えられない．

昨今，品種の整理統合化が図られてきているとはいえ，スケジューリング業務担当者が取り組まなくてはならないスケジューリングの規模は依然として規模が大きく，

- 工数：スケジュールをたてるのに多大な工数がかかる
- スケジュールの質：業務のアウトプットであるスケジュールの質が満足水準に達しがたい．

という問題をかかえている．

このような問題解決のために生産スケジューリングのシステム化が求められ、そのシステム化にあたって使われている技法としては、

1. 最適解 / 近似解指向の O R 法
2. シミュレーション法
3. A I 法・知識獲得法

などに分類される [1]。このうち、知識獲得法については、イリノイ大学の Shaw らが開発した PDS[2]、パデュー大学の Chaturvedi らが開発した GDCA[3] などが存在する。

しかし現状の生産スケジューリングは、生産システムの規模が大きくなるにつれ計算時間が指数関数的に増大するという問題を含んでいる [4]。ゆえに実用的な生産スケジューリング方式として、ヒューリスティック・ルール選択またはディスパッチング・ルール選択と呼ばれるアルゴリズム [5][6] が種々考案されていて、そのアルゴリズムは広く使われている。

それらのアルゴリズムを適用する際の問題点は、生産システムの工程すべてに単一のディスパッチング・ルールを適用する場合の評価基準には経験則が存在しているが、生産システムの工程毎、もしくは工程グループ毎に異なるディスパッチング・ルールを適用する場合の評価基準には経験則が存在せず、試行錯誤的手法を用いてディスパッチング・ルールを選択していることである。

データ・マイニングは知識獲得およびデータベース研究の中のひとつの分野であり、マイニング (mining : 採鉱) という名前の通り、大量のデータから何かしらのパターンを掘り当てることを意味する。大量のデータより導出されるパターンが相関 (association) ルールと呼ばれ、 $X \implies Y$ のように表される。この場合の X は条件部、 Y は結論部と呼ばれる。この相関ルールを導出するために Agrawal[7] らが用いたアプリアリアルゴリズムを適用する。

相関ルールの価値の尺度は 2 種類あり、確信度 (confidence) と呼ばれる物と、サポート (support) と呼ばれる物である。確信度は、条件部と結論部を同時に満たすトランザクションが、条件部を満たすトランザクション中での占める割合を、サポートは、全トランザクションに対して、条件部と結論部を同時に満たすトランザクションが含まれる割合を表す。このデータ・マイニングの相関ルール導出の手法を、生産スケジューリングに適用する。

本研究では、このディスパッチング・ルール適用に指針が存在しないことに着目し、指針を与えるためにルール・マイニングという手法を用いる。このルール・マイニング手法は、その基礎となるデータ・マイニングと「同等」という概念とを併せ持った手法である。ルール・マイニング法を用い、複数の評価基準に対してのディスパッチング・ルール選択に定量的な基準の提案を試みる。

手順として、生産シミュレーターを用いて生産工程をモデル化し、ジョブと呼ばれる生産加工品を投入、そしてシミュレーション実行時に、各生産工程に数々のディスパッチング・ルールを適用させる。評価基準と各工程に適用させたディスパッチング・ルールを出力項目として、データ・マイニングの適用対象となるトランザクションを作成する。

トランザクションを作成する際にディスパッチング・ルールが性能的に、いわゆる「似ている」ということを表現する「同等」という概念を導入する。「同等」を導入したトランザクションから、評価基準とディスパッチング・ルール間の相関ルールを抽出する手法をルール・マイニング法と呼び、本研究の特色となる。

ルール・マイニング法を用い、ディスパッチング・ルール選択に対して指針を与えること、そして評価指標の異なる部門、例えば製造部門で評価基準として用いられことの多い滞留時間や、生産管理部門で評価基準として用いられことの多い納期遅れなど、立場の異なる部門間の評価を改良すること、つまり評価基準が互いに相反する場合に、どのようなディスパッチング・ルールが有効であるかの指針を与える基礎的な研究が本研究の目的である。

1.2 本論文の構成

本論文の構成は次のとおりである。この後の2章では、準備として生産システムに用いる変数、評価基準、ディスパッチング・ルールを説明し、相関ルールを導出するために用いるデータ・マイニングとアプリアリアルゴリズムについて説明する。3章では、ルール・マイニング法適用の準備として、生産シミュレーションモデルの基本仕様、それにルール・マイニング適用への準備として導入した「同等」について述べ、4章にルール・マイニングの適用例、5章にルール・マイニング法の評価を二つのシミュレーションモデルに対して行い、同等を改良する。最後に6章にて本研究内容をまとめ、今後の展望について述べる。

第 2 章

準備

本章では，生産システムで用いられる変数，ディスパッチング・ルール，評価基準，そしてデータ・マイニングのアルゴリズムについて述べる．

2.1 生産システムのモデル

2.1.1 使用する変数

図 2.1 に示されているように， n 個のジョブの集合を $J = \{J_i\}, i = 1, 2, \dots, n$ とし，製造工程を構成している m 工程の集合を $M = \{M_j\}, j = 1, 2, \dots, m$ とする．このとき，ジョブ J_i は，指定された n_i 個の工程で処理を受ける．このような工程指示集合を $O = \{O_{ij}\}, i = 1, 2, \dots, n, j = 1, 2, \dots, n_i$ とし，ジョブ J_i の未処理工程の集合を S_i ， S_i の要素数を $|S_i|$ と表す．そしてシステムの現在時刻を $tnow$ とする．

ジョブ J_i の属性値は，納期 d_i ，生産システムへの到着時刻 e_i ，工程処理集合 $O = \{O_{ij}\}$ で指示される工程処理時間 p_{ij} ，同じく $O = \{O_{ij}\}$ にて指示される全工程を終了した時刻 f_i ，スラック（納期余裕）時間 $d_i - tnow - \sum_{j \in S_i} p_{ij}$ ，納期までの切迫時間 $d_i - tnow$ とする．

以上から下記を表すことが出来る．

- ジョブ J_i の滞留時間： $F_i = f_i - e_i$

ジョブ J_i が投入されてから全工程終了までの生産システム内滞在時間．

- ジョブ J_i の納期遅れ： $T_i = \max(0, L_i)$ ，ただし $L_i = f_i - d_i$

ジョブ J_i の納期を基準にして遅れて全工程終了した場合の時間差．

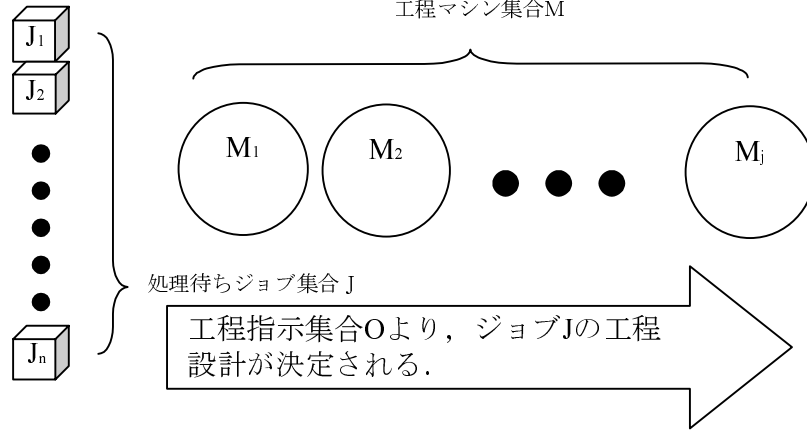


図 2.1: モデル概念

「納期遅れ」は純粹の納期遅れ (tardiness) のみを、「納期ずれ」は納期より早く全工程終了した場合を負の納期遅れと考えた納期ずれ (lateness) を表す。

2.1.2 評価基準

生産システムにおける代表的な評価基準 [13] を次にあげる。

1. 総処理時間 (メイクスパン)

最初の投入ジョブの処理開始時点から最終の投入ジョブの全工程終了時点までの総時間。静的なスケジューリング (全ジョブの生産システムへの到着時刻 $e_i = 0, i = 1, 2, \dots, n$) ならば, 総処理時間と最大滞留時間は同値となる。

2. 最大滞留時間 (F_{max})

$$F_{max} = \max_{1 \leq i \leq n} F_i \quad (2.1.1)$$

3. 平均滞留時間 ($\bar{F}$)

$$\bar{F} = \frac{1}{n} \sum_{i=1}^n F_i \quad (2.1.2)$$

4. 最大納期遅れ (T_{max})

$$T_{max} = \max_{1 \leq i \leq n} T_i \quad (2.1.3)$$

5. 平均納期遅れ ($\bar{T}$)

$$\bar{T} = \frac{1}{n} \sum_{i=1}^n T_i \quad (2.1.4)$$

6. 最大納期ずれ (L_{max})

$$L_{max} = \max_{1 \leq i \leq n} L_i \quad (2.1.5)$$

7. 平均納期ずれ ($\bar{L}$)

$$\bar{L} = \frac{1}{n} \sum_{i=1}^n L_i \quad (2.1.6)$$

8. 納期遅れジョブ数 (N_t)

$$N_t = \sum_{i=1}^n \delta_i \quad (2.1.7)$$

$$\delta_i = \begin{cases} 0 : T_i = 0 \text{ の場合} \\ 1 : T_i > 0 \text{ の場合} \end{cases}$$

2.1.3 ディスパッチング・ルール

Z_{ij} はジョブを投入する際の優先度インデックスで, Z_{ij} の値が小さいほど投入優先度は高いことを意味する [13] .

1. 最小作業時間順 (Shortest Processing Time, 以下 SPT)

$$Z_{ij} = p_{ij}$$

2. 最早納期順 (Earliest Due Date, 以下 EDD または D.DATE)

$$Z_{ij} = d_i$$

3. スラック時間順 (以下 SLACK)

$$Z_{ij} = d_i - t_{now} - \sum_{j \in S_i} p_{ij}$$

4. 到着時刻順 (先入れ先出し) (First Come First Served, 以下 FCFS)

$$Z_{ij} = e_i$$

5. S/OPN (Slack/OPeration Numbers : SLACK 時間を残り工程数で割って得た Z_{ij} の最小値を持つジョブを優先)

$$Z_{ij} = (d_i - tnow - \sum_{j \in S_i} p_{ij}) / |S_i|$$

6. S/RPT (Slack/Remaining Processing Time : SLACK 時間を残り処理時間で割って得た Z_{ij} の最小値を持つジョブを優先)

$$Z_{ij} = (d_i - tnow - \sum_{j \in S_i} p_{ij}) / \sum_{j \in S_i} p_{ij}$$

7. D/OPN (Due date/OPeration Numbers : 納期までの切迫時間を残り工程数で割って得た Z_{ij} の最小値を持つジョブを優先)

$$Z_{ij} = (d_i - tnow) / |S_i|$$

8. D/RPT (Due date/Remaining Processing Time : 納期までの切迫時間を残り処理時間で割って得た Z_{ij} の最小値を持つジョブを優先)

$$Z_{ij} = (d_i - tnow) / \sum_{j \in S_i} p_{ij}$$

9. 複合ルール 複数のディスパッチング・ルールを重み付けして用いる .

$$\begin{aligned} Z_{ij} &= a * [Dispatching Rule 1 \text{ の場合の } Z_{ij}] + \\ &\quad (1 - a) * [Dispatching Rule 2 \text{ の場合の } Z_{ij}] \quad (2.1.8) \\ &\quad 0 \leq a \leq 1 \end{aligned}$$

2.2 データ・マイニング

データ・マイニングは知識獲得およびデータベース研究の中のひとつの分野である . マイニング (mining : 採鉱) という名前の通り , 大量のデータから何かしらのパターン知識などをより速く掘り当てることを意味する [8] .

現在、バーコードやクレジットカードなどデータ収集の大幅な進歩と、記憶装置の劇的な低価格化により、情報収集はたやすい作業になり、巨大なデータ（数ギガからテラバイト）が既に存在する。特に、従来は敬遠されがちであった時間情報（タイムスタンプ）の入った履歴データの収集も可能になってきている。このようなデータベースから、データの属性間に成り立つ規則を発掘し、営業戦略を立案する上で役に立つ、規則・法則のたぐいを得たいという自然な欲求が生まれる。

例として、数十の属性を含むデータベース、例えば銀行における数百万顧客のデータベースを考える。属性としては、当座取引有無・公共料金引落有無等の2値属性や、血液型等の離散属性、普通預金残高・年齢・取引期間等の数値属性があるとする。いま「先月発売した金融商品 A が売れている顧客層を明日の朝までに特定したい」とする。いま条件 X を満たす顧客は高い確度で金融商品 A を購入するというルールを

$$X \implies (\text{金融商品 } A = \text{yes})$$

と記述する。この場合の X を条件部（金融商品 $A = \text{yes}$ ）を結論部と呼ぶ。このように導出されるルールを相関ルール（association rule）と呼ぶ。

相関ルールの価値測定法は2種類あり、条件部を満たすデータが結論部も満たす割合を確信度（confidence）と呼び、条件部と結論部を同時に満たすデータの全データに対する割合を相関ルールのサポート（support）と呼ぶ。価値測定法の例として、バーコードで大量のデータを収集できる POS システムを考える。この場合に発掘できた相関ルールは、

$$\text{目玉商品 } A \implies \text{日用品 } B$$

であったとする。この相関ルールの確信度の意味する内容は、目玉商品 A を購入した顧客のうち、何%が日用品 B を購入するかを示しており、仮に確信度が 100%だとすると、目玉商品 A の購入顧客全員が日用品 B をも購入するといえる。サポートが意味する内容は、目玉商品 A と日用品 B を同時に購入した顧客は、全顧客のうちの何%であるかを示している。

価値をはかる上でサポートはある程度高いことが望ましい。例えばサポートが 0.01% の相関ルールは 0.01% の顧客にしか当てはまらない出番の少ない相関ルールであり、あまり価値がないかもしれない。極端に小さなサポート値のルールを排除し、有効と考えられる相関ルール数を得る方法としては、出力する相関ルールのサポートの最小値（最小サポートと呼ぶ）を設定するのが有効である。すなわち、少なくとも最小サポートを満たすよ

うなすべてのパターンを発掘することは有意義なパターンを得るための第一歩と考えられる。

2.3 アプリアリアルゴリズム

前節で紹介したシステムを構築するために，Agrawalらが用いた相関ルールを発見するためのアルゴリズムがアプリアリアルゴリズム [7] である。

アイテム集合 $I = \{i_1, i_2, \dots, i_m\}$ を考える． D はトランザクション集合で，個々のトランザクション T は， $T \subseteq I$ なるアイテムの集合である．そして個々のトランザクションには，ユニークなトランザクション ID (TID) を割り当てる．

この時に導出される相関ルールは，次の形式で表される．

$$X \implies Y; \quad X \subset I, Y \subset I, X \cap Y = \phi \quad (2.3.1)$$

図 2.2 に (2.3.1) 式の関係を表す．図中の T_x, T_y は，アイテム X, Y を含んでいるトランザクション集合である．

アイテム集合 X を含む D のトランザクションの $c\%$ がアイテム集合 Y も含むのであれば，相関ルール $X \implies Y$ は確信度 (confidence) c を持つといい， D に対して， X, Y の両方をカバーするトランザクションが $s\%$ 存在するとき，相関ルール $X \implies Y$ はサポート (support) s を持つという．

$$\text{サポート} : \frac{\text{アイテム集合 } X, Y \text{ を同時に含むトランザクション数}}{\text{全トランザクション数}} \quad (2.3.2)$$

$$\text{確信度} : \frac{\text{アイテム集合 } X, Y \text{ を同時に含むトランザクション数}}{\text{アイテム集合 } X \text{ を含むトランザクション数}} \quad (2.3.3)$$

(2.3.1) 式のサポートと確信度は，図 2.2 の数字で示された領域で，それぞれ，

$$\begin{aligned} \text{サポート} &= \frac{3}{1 + 2 + 3 + 4} \\ \text{確信度} &= \frac{3}{1 + 3} \end{aligned}$$

となる．

二つの入力値 (最小確信度値，最小サポート値) を満たす相関ルールを導出するアルゴリズムを具体例を挙げて示す．対象となるトランザクション集合は，表 2.2 の Database (Transaction)

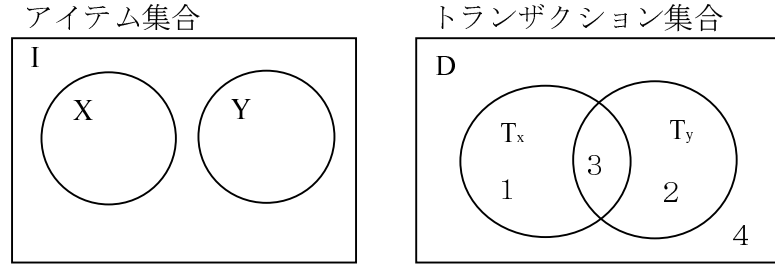


図 2.2: アイテム集合，トランザクションの関係

と記されているものを用い，最小サポート値を 50%，確信度の条件を無視するために最小確信度は 0%とする．そしてアルゴリズムでは，表 2.1 に示してある集合を用いる．実行例に用いている $L_k, C_k, \bar{C}_k$ の集合は，表 2.2 に示してある．

実行例：

1. Database(Transaction) より L_1 を作成する．

最小サポートと全トランザクション数から，最小サポート数 $minsup$ を求めておく．トランザクション数が 4，最初サポートが 50%であるから $minsup=2$ となる．トランザクションから，1ヶのアイテムを持つアイテム集合と，そのアイテムがトランザクション内に存在している数を数え上げ， $minsup$ を満たすものでラージアイテム集合 L_1 を作成する．

2. Database(Transaction) より $\bar{C}_1$ を作成する．

$\bar{C}_1$ は，Database(Transaction) そのものとなる．

3. L_1 から C_2 を作成する．

L_1 のアイテム集合 $\{\{1\}, \{2\}, \{3\}, \{5\}\}$ から，メンバーが 2ヶのアイテムを持つ集合 C_2 を作成するときの組み合わせは， $\{\{1\ 2\}, \{1\ 3\}, \{1\ 5\}, \{2\ 3\}, \{2\ 5\}, \{3\ 5\}\}$ となる．

4. C_2 から $\bar{C}_2$ を作成する．

C_2 メンバーのアイテム集合が， $\bar{C}_1$ のどの TID を持つメンバーから構成されているかを検索し，その時の TID ごとにアイテム集合を作成する．かつ，その作成時に C_2 メンバーのアイテム集合が $\bar{C}_1$ 内に存在している個数を数え上げる．例えば， $\{1\ 3\}$

$k\text{-itemset}$	アイテム集合, k 個のアイテムを持つ
L_k	ラージ k アイテム集合 この集合は k 個のアイテムを持つ集合で, 各々のメンバーは最小サポートを満たし, 2ヶの要素 1. アイテム集合 2. サポートカウント を持つ
C_k	ラージ k アイテム集合への候補集合 各々のメンバーは L_k と同様に 2ヶの要素 1. アイテム集合 2. サポートカウント を持つ この候補集合から最小サポートを満たすメンバーで L_k が作成される.
$\bar{C}_k$	トランザクション ID を持つアイテム集合 このメンバーは要素としてサポートカウントを持たず, 同一のアイテム集合であっても, トランザクション ID が異なれば別々のメンバーとなる. 各々のメンバーは 2ヶの要素 1. トランザクション集合 2. アイテム集合 を持つ

表 2.1: アルゴリズムで用いる集合

の場合は, TID 100 : $\{\{1\}, \{3\}, \{4\}\}$ と TID 300 : $\{\{1\}, \{2\}, \{3\}, \{5\}\}$ に存在しているので, サポートカウントは 2 となる.

5. C_2 から L_2 を作成する.

C_2 のメンバーの中で, $minsup$ を満たすもので L_2 を構成する. 例えば, $\{1\ 2\}$ はサポートカウントが 1 であり, $minsup$ を満たさないので L_2 の構成メンバーにはなれない.

6. L_2 から C_3 を作成する.

L_2 のアイテム集合 $\{\{1\ 3\}, \{2\ 3\}, \{2\ 5\}, \{3\ 5\}\}$ から, メンバーが 3ヶのアイテムを持つ集合 C_3 を作成するときの組み合わせは, $\{2\ 3\ 5\}$ のみとなる.

(補足) L_2 から C_3 の作成手順:

(a) $p, q \in L_2$ を用意する. このなかでのアイテムは昇順になっている. 例として $p = \{2\ 3\}, q = \{2\ 5\}$ とする.

- (b) $p.item_1 = q.item_1, p.item_2 < q.item_2$ を満たすならば, $\{p.item_1 p.item_2 q.item_2\}$ を求める次候補集合 C_3 に追加する. なお $item_1, item_2$ は, 昇順に並んだ個々のアイテムを表す. 例の $p = \{2\ 3\}, q = \{2\ 5\}$ はこの条件を満たすので, $\{2\ 3\ 5\}$ が C_3 に追加される.
- (c) 追加された候補メンバーから任意のアイテムを 1ヶ取り除いたものが, L_2 に含まれているかをチェックする. 含まれていなければ, C_3 から取り除く. 例では, $\{2\ 3\ 5\}$ から $\{2\ 3\}, \{2\ 5\}, \{3\ 5\}$ の 3ヶをチェックしなければならない. この 3ヶのアイテム集合は L_2 に含まれているので, 取り除かれることは無い.
- (d) p, q を入れ替えて同様の操作を行う. 最終的に残るのは $\{2\ 3\ 5\}$ のみとなる. L_1 から C_2 を作成する場合も同様の手順をふむのであるが, L_1 のアイテム集合のアイテム数が 1 であるために, そこから作成されるすべての組み合わせが C_2 となる.

7. C_3 から $\bar{C}_3$ を作成する.

C_3 メンバーのアイテム集合が, $\bar{C}_2$ のどの TID を持つメンバーから構成されているかを検索し, その時の TID ごとにアイテム集合を作成する. かつ, その作成時に C_3 メンバーのアイテム集合のアイテムを 1ヶ欠落させたものが $\bar{C}_2$ 内に存在している個数を数え上げる. 例えば, $\{2\ 3\ 5\}$ の場合は, $\{2\ 3\}, \{2\ 5\}, \{3\ 5\}$ が TID 200 と TID 300 に存在しているので, $\bar{C}_3$ のメンバーは TID 200 : $\{\{2\ 3\}, \{2\ 5\}, \{3\ 5\}\}$, TID 300 : $\{\{2\ 3\}, \{2\ 5\}, \{3\ 5\}\}$, C_3 のサポートカウントは 2 となる.

8. C_3 から L_3 を作成する.

C_3 の唯一のメンバー $\{2\ 3\ 5\}$ は $minsup$ を満たすので L_3 を構成する.

9. 同様の手順で $C_4, \bar{C}_4, L_4$ を作成しようとしてもメンバーが存在しない. よって, これで終了となり, L_1, L_2, L_3 が最小サポートを満たすラージアイテム集合となる. このラージアイテム集合より, 最小確信度を満たすものを導出すると図 2.4 に示されている相関ルールとなる.

ここでの実行例で示した表 2.2 の各集合の作成順序は,

$$\begin{aligned} Transaction &\rightarrow L_1 \rightarrow \bar{C}_1 \rightarrow C_2 \rightarrow \bar{C}_2 \rightarrow L_2 \rightarrow C_3 \rightarrow \bar{C}_3 \rightarrow L_3 \rightarrow \\ &C_4(no\ member) \rightarrow \bar{C}_4(no\ member) \rightarrow L_4(no\ member) \rightarrow fin. \end{aligned}$$

となる．図 2.3 に実行例で用いたアルゴリズムを示す．

図 2.4 の実行結果について，POS の具体例を挙げて説明する．入力例の TID は顧客毎につけられた顧客 ID，Items のそれぞれの数値は，商品 $A, B, \dots, E$ と考える．つまり，入力例の TID が 100 のトランザクションが示す内容は，顧客 ID:100 の顧客が商品 A, C, D を購入したことを示す．

実行結果の相関ルール群は，最小サポート 50% を満たしているので，顧客全体の少なくとも 50% には当てはまるルールである．相関ルール $\{1, \} \rightarrow \{3, \} : 2/2$ は，

商品 A を購入する顧客は商品 C も購入する．

というルールを表現しており，右隣に示している数値 $2/2$ は確信度の値を示している．確信度の値が $2/2$ ，すなわち 100% を示しているので，「商品 A を購入する顧客は，必ず商品 C も購入する」と言える．

相関ルール $\{2, \} \rightarrow \{3, 5, \} : 2/3$ と $\{2, 3, \} \rightarrow \{5, \} : 2/2$ の違いを述べる．共に商品 B, C, E 購入するというルールであるが，確信度の値の違いにより，「商品 B を購入する顧客は，商品 C, E を $2/3$ の確立で購入する」と「商品 B, C を購入する顧客は，必ず商品 E を購入する」の違いが発生する．

Database(Transaction)		L_1		$\bar{C}_1$	
TID	Items	Itemset	Support	TID	Set-of-Itemsets
100	1 3 4	{1}	2	100	{{1},{3},{4}}
200	2 3 5	{2}	3	200	{{2},{3},{5}}
300	1 2 3 5	{3}	3	300	{{1},{2},{3},{5}}
400	2 5	{5}	3	400	{{2},{5}}

C_2		L_2		$\bar{C}_2$	
Itemset	Support	Itemset	Support	TID	Set-of-Itemsets
{1 2}	1	{1 3}	2	100	{{1 3}}
{1 3}	2	{2 3}	2	200	{{2 3},{2 5},{3 5}}
{1 5}	1	{2 5}	3	300	
{2 3}	2	{3 5}	2	400	
{2 5}	3				
{3 5}	2				

C_3		$\bar{C}_3$		L_3	
Itemset	Support	TID	Set-of-Itemsets	Itemset	Support
{2 3 5}	2	200	{{2 3 5}}	{2 3 5}	2
		300			

表 2.2: アプリアリアルゴリズムの実行例

```

 $L_1 = \{ \text{large 1-itemsets} \};$ 
 $\bar{C}_1 = \text{database } D;$ 
for(  $k = 2; L_{k-1} \neq 0; k++$  ) do begin
   $C_k = \text{apriori-gen}(L_{k-1})$  // 候補
   $\bar{C}_k = 0;$ 
  forall entries  $t \in \bar{C}_{k-1}$  do begin;
     $C_t = \{c \in C_k \mid (c - c[k]) \in t.\text{set-of-itemsets} \wedge (c - c[k-1]) \in t.\text{set-of-itemsets}\};$ 
    forall candidates  $c \in C_t$  do;
       $c.\text{count}++;$ 
    if( $C_t \neq 0$ ) then  $\bar{C}_k += \langle t.\text{TID}, C_t \rangle;$ 
  end;
   $L_k = \{c \in C_k \mid c.\text{count} \geq \text{minsup}\};$ 
end;
Answer =  $\cup_k L_k;$ 

```

図 2.3: 二つの入力値を満たす相関ルールの導出

図 2.3 の補足：

apriori-gen() の動作は，アイテム集合 L_{k-1} から次のアイテム集合への候補集合 C_k を作成する．この候補集合作成時には最小サポート値は考慮されない．

1. $p, q \in L_{k-1}$ を用意する．このなかでのアイテムは昇順になっている．
2. $p.item_1 = q.item_1, \dots, p.item_{k-2} = q.item_{k-2}, p.item_{k-1} < q.item_{k-1}$ を満たすならば， $\{p.item_1 \dots p.item_{k-2} q.item_{k-1}\}$ を求める次候補集合 C_k に追加する．なお $item_1, \dots, item_{k-2}, item_{k-1}$ は，昇順に並んだ個々のアイテムを表す． $p, q \in L_{k-1}$ を全て列挙してこの手順を繰り返す．
3. 追加された候補メンバーから任意のアイテムを 1ヶ取り除いたものが， L_{k-1} に含まれているかをチェックする．含まれていなければ， C_k から取り除く．最終的に残るものが次のアイテム集合への候補集合 C_k となる．



入力例：

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

実行結果（抜粋）:

```

Minimum Support (%) : 50
Minimum Confidence (%) : 0
{1,} ---> {3,} : 2/2
{3,} ---> {1,} : 2/3
{2,} ---> {3,} : 2/3
{3,} ---> {2,} : 2/3
...
{2,} ---> {3,5,} : 2/3
{3,} ---> {2,5,} : 2/3
{5,} ---> {2,3,} : 2/3
{2,3,} ---> {5,} : 2/2
{2,5,} ---> {3,} : 2/3
{3,5,} ---> {2,} : 2/2

```

図 2.4: アプリオリアルゴリズムのインターフェースと入出力例

第 3 章

ルール・マイニングの導入

この章では，生産シミュレーションソフト *System Modeling Corp.* の *ARENA*[12] を用いてのデータ収集と，ルール・マイニング適用に向けての「同等」という概念 [14] の導入を説明する．

3.1 シミュレーションモデルの基本仕様

本章で用いたシミュレーションモデルを次に示す．

1. ショップ内には 2 台の生産設備があり，各ジョブは表 3.1 に示すように，工程数が 1，2 の二通りで，ジョブの工程割り当て，すなわち工程設計 ID はランダムに割り当てられる．
2. 生産の対象ジョブを 20ヶ発生させ，そのジョブのショップ到着時刻はすべてゼロ ($e_i = 0; \forall i$) である．
3. 各工程の処理時間は，図 3.2 で示されたものを用いる．図 3.2 の $unif(A, B)$ は区間 $[A, B]$ の一様乱数を表している．
4. 納期の設定方法は総処理時間に納期係数 α を乗じ，バラツキを与えるため，試行錯誤により 0.7 ~ 1.3 の一様乱数 $unif(0.7, 1.3)$ を乗じた．総処理時間は，工程設計 ID 3,4 ならば両処理時間を加算，1,2 ならば第 1 工程のみとする．

$$\alpha * \text{総処理時間} * unif(0.7, 1.3) \quad (3.1.1)$$

工程設計 ID	工程順序
1	マシン 1
2	マシン 2
3	マシン 1 → マシン 2
4	マシン 2 → マシン 1

表 3.1: 2 マシンジョブショップモデルの工程設計

工程	処理時間
第 1 工程	$unif(70, 400)$
第 2 工程	$unif(300, 800)$

表 3.2: 2 マシンジョブショップモデルの工程処理時間

5. 各マシンは、シミュレーション実行中は常に使用可能であり、故障などの要因は考えない。

以上のジョブショップモデル問題で、各マシンで適用するディスパッチング・ルールとして次の集合 Z を考える。

$$Z = \{EDD, SLACK, SPT\} \quad (3.1.2)$$

この 3 種類のディスパッチング・ルールをそれぞれのマシンに適用する。すなわち、このジョブショップモデル内では、9 種類のディスパッチング・ルールの組み合わせが存在する。以後、例えばマシン 1 に EDD 、マシン 2 に SPT を適用させた場合を $EDD\&SPT$ と表記する。

9 種類の組み合わせのすべてについて、 n ジョブを処理することを考える。このシミュレーションモデルの生産対象ジョブは 20 ケなので $n = 20$ となり、これが 1 回のシミュレーション実行単位となる。ここで、各ディスパッチング・ルールの組み合わせについて、同一の属性値を持つ n ジョブを使用する。このようなシミュレーションを、それぞれ異なる n ジョブの集合（一様乱数の乱数系列に依存）に対して、一様乱数 $unif()$ の乱数系列を変更させながら、1000 回のシミュレーションを行った。

3.2 ルール・マイニング適用への準備

3.2.1 「同等」の導入

製造工程に，あるディスパッチング・ルール X を適用したときの評価基準を P^X と書くものとする．例えば，ディスパッチング・ルール EDD を適用したときの最大納期遅れは， T_{max}^{EDD} と記述されることになる．ここで X は，必ずしも単一のディスパッチング・ルールとは限らず，複数のディスパッチング・ルールの組み合わせの場合もある．

(定義) [14]

ディスパッチング・ルール X は，ある与えられた実数 (尺度) $\beta > 0$ に対して，

$$P^X - \beta \leq P^Y \quad (3.2.1)$$

を満たすならば，評価基準 P に関して，ディスパッチング・ルール Y と， β において同等であるという．このとき，

$$X = \frac{Y}{(\beta, P)} \quad (3.2.2)$$

と書くことにする． □

例えば，ディスパッチング・ルール EDD と SPT を考える．このとき，ある与えられた実数 $\beta > 0$ ，平均滞留時間 ($\bar{F}$) に対して，

$$\bar{F}^{EDD} - \beta \leq \bar{F}^{SPT}$$

を満たすならば，ディスパッチング・ルール EDD と SPT は， β において同等であるという．このとき，

$$EDD = \frac{SPT}{(\beta, \bar{F})}$$

と書く．

ここで注意すべきこととして，この定義ではあるいくつかのディスパッチング・ルールに関して，それらがいわゆる「似ていること」を知るために，評価基準値 P^Y が「 X と Y を測る基準となる尺度」として必要であるということを主張している．すなわち，何らかの手段により，尺度 P^Y の値を求める必要がある．さらに， P^Y と P^X の算定に際しては，同一のジョブの集まりを用いることが必要である．

3.2.2 「同等」の評価

図 3.1 に示されている相反するディスパッチング・ルールの簡単な例を示す．文献 [13] より，1 マシンの静的スケジューリング問題に SPT ルールを適用すると平均滞留時間 $\bar{F}$ を最小化し， EDD ルールを適用すると最大納期遅れ T_{max} を最小化することが知られている．

図 3.1 の 1 マシン，2 ジョブ，2 ディスパッチング・ルールの場合で，ジョブ A の処理時間は 5，納期は 7，ジョブ B の処理時間は 3，納期は 10 であるようなジョブについて考える．このとき， EDD を適用するとジョブ A ，次にジョブ B という順序で工程へ投入することになり，この結果として，

$$T_{max}^{EDD} = 0, \quad \bar{F}^{EDD} = 6.5$$

となる．逆に， SPT を適用するとジョブ B ，次にジョブ A となり，結果的に，

$$T_{max}^{SPT} = 1, \quad \bar{F}^{SPT} = 5.5$$

となる．

このような場合に，もし納期遅れが許されないのであれば， EDD を適用した投入順序とせざるを得ない．ところが，納期遅れに対して，多少の遅れを許せる場合には， SPT を適用して， EDD を適用した場合に比べて平均滞留時間の改善を図ることが可能となる．すなわち， $\beta = 1$ と設定し，最大納期遅れを評価基準として考えると，

$$SPT = \frac{EDD}{(\beta = 1, T_{max})}$$

という結果を得る．これは，定義より「 $\beta = 1$ において SPT と EDD は同等」であり， $\beta = 1$ ということを考慮すると最大納期遅れを犠牲にすることなしに，この問題に SPT を適用し，平均滞留時間を 1.0 だけ改善できるということを意味している．

従来の手法では，情報不足により EDD を適用せざるを得なかった状況において，定量的な基準尺度 ($\beta = 1$) を持って，実は SPT を適用することも可能であると考えられることができる．

以上のような考え方を発展させ，前節にて述べた製造工程に適用可能であるディスパッチング・ルールの組み合わせを求める手段について考察していく．

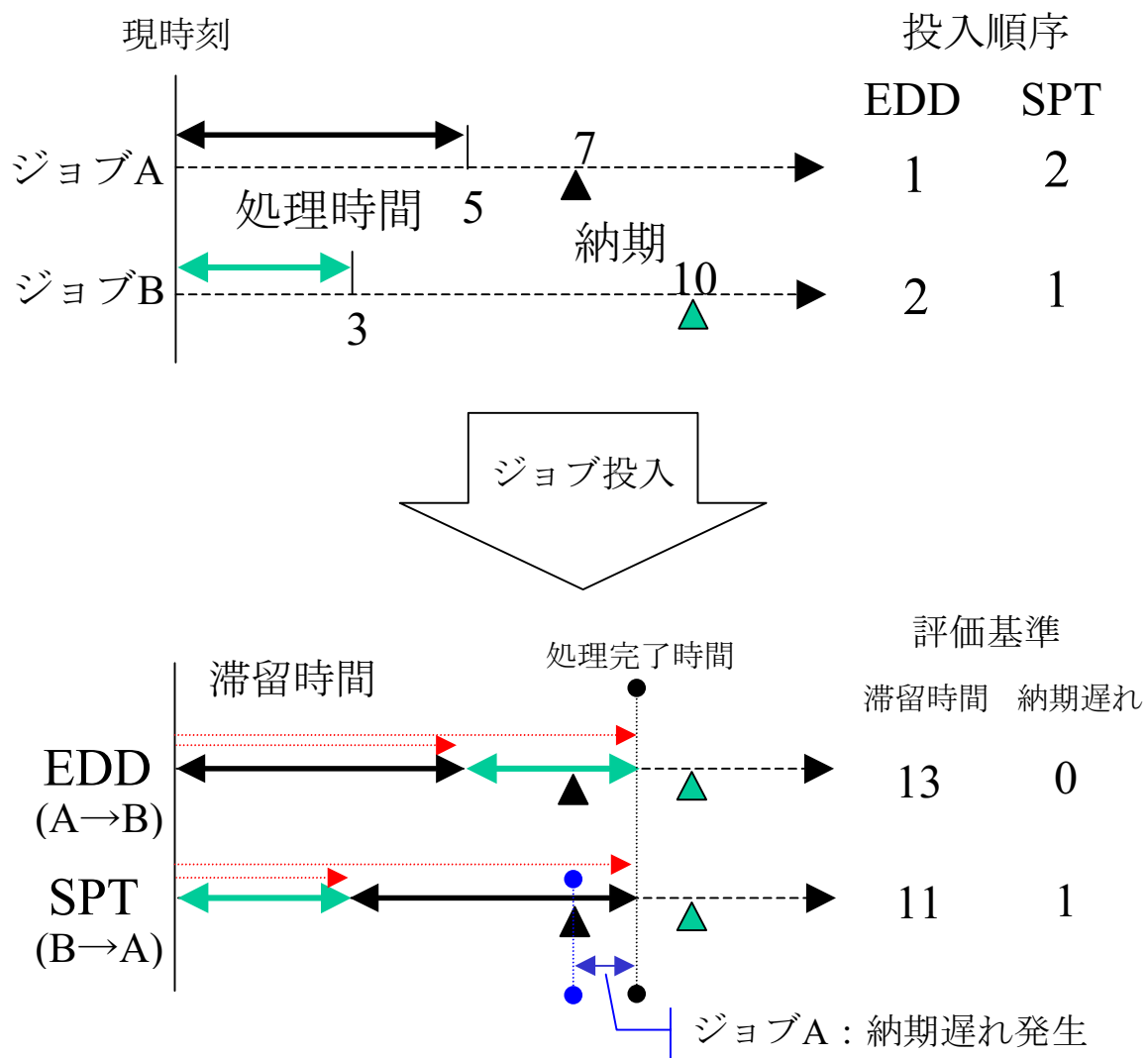


図 3.1: 1 マシン , 2 ジョブ , 2 ディスパッチング・ルールの場合

3.2.3 問題の設定

3.1 節で示した 2 マシンの場合のシミュレーション単位において，次のように定義される集合 D_e を求める．

$$D_e = \{z \in Z \times Z \mid z = \frac{e}{(\beta, T_{max})}\} \quad (3.2.3)$$

ここで Z は (3.1.2) 式にて与えられ， $e \in Z \times Z$ は，9 種類のディスパッチング・ルールの組み合わせ中で，最大納期遅れ T_{max} を最も小さくしたディスパッチング・ルールの組み合わせとする．また， β はある与えられた実数とする．同様に，各シミュレーションにおいて，次のように定義される集合 D_s を求める．

$$D_s = \{z \in Z \times Z \mid z = \frac{s}{(\gamma, \bar{F})}\} \quad (3.2.4)$$

ここで $s \in Z \times Z$ は，9 種類のディスパッチング・ルールの中で，平均滞留時間 $\bar{F}$ を最も小さくしたディスパッチング・ルールの組み合わせとする．また， γ は与えられた実数とする．

これより，シミュレーションの実行回数である N 個の D_e, D_s を得る．これらのシミュレーション結果は，例えば D_e として 1 組のディスパッチング・ルールだけを含むもの，2 組のディスパッチング・ルールを含むものなど，(3.1.1) 式の納期係数 α ，同等の尺度 β ， γ に応じて，最高 9 組すべてのディスパッチング・ルールを含むことが考えられる．

ここでの興味は，9 組のディスパッチング・ルールの組が，どのようなパターンで出現したかということにある．このようなパターンとして，例えば，

$$T_{max} \implies \{EDD\&EDD, EDD\&SLACK\} \quad (3.2.5)$$

などが考えられる．これは，最大納期遅れに関して，ディスパッチング・ルールの組 $EDD\&EDD$ と $EDD\&SLACK$ が有効であるであることを示している．

(3.2.5) 式で表されているルールが，2.3 節で示されている「相関ルール」と呼ばれるものであり，アプリアリ・アルゴリズム [7] の抽出結果として与えられることになる．

相関ルールを抽出するため，アイテム集合を作成する．そのアイテム集合はシミュレーションで考えている評価基準，および 9 組のディスパッチング・ルールのそれぞれを要素とする．すなわち，アイテム集合 I は，

$$I = \{T_{max}, \bar{F}, EDD\&EDD, \dots, SPT\&SPT\} \quad (3.2.6)$$

となる．

トランザクション集合は 2 種類の集合を用意する．つまり，評価基準 T_{max} を，関連したシミュレーション結果 D_e に紐付けたもの，および評価基準 $\bar{F}$ を，関連したシミュレーション結果 D_s に紐付けたもの，例えば，

$$\bar{D}_e = \{T_{max}, EDD\&EDD, EDD\&SPT\} \quad (3.2.7)$$

$$\bar{D}_s = \{\bar{F}, SPT\&SPT\} \quad (3.2.8)$$

などが，トランザクションになり，各々のトランザクションは

$$\bar{D}_e \subseteq I, \bar{D}_s \subseteq I$$

となる．

このとき，(2.3.2) 式，(2.3.3) 式で示されているサポート，確信度は，(3.2.5) 式で表されるような相関ルールを例に， T_{max} を条件部 X ， $\{EDD\&EDD, EDD\&SLACK\}$ を結論部 Y ，シミュレーション実行回数 N を用いて，

$$\text{サポート} : \frac{\bar{D}_e \text{ において } X \cap Y \text{ が出現した数}}{N} \quad (3.2.9)$$

$$\text{確信度} : \frac{\bar{D}_e \text{ において } X \cap Y \text{ が出現した数}}{\bar{D}_e \text{ において } X \text{ が出現した数}} \quad (3.2.10)$$

または，

$$\text{サポート} : \frac{\bar{D}_s \text{ において } X \cap Y \text{ が出現した数}}{N} \quad (3.2.11)$$

$$\text{確信度} : \frac{\bar{D}_s \text{ において } X \cap Y \text{ が出現した数}}{\bar{D}_s \text{ において } X \text{ が出現した数}} \quad (3.2.12)$$

となる．

(3.2.6) 式のアイテム集合，(3.2.7) 式および (3.2.8) 式のトランザクション集合を用い，(3.2.9) 式から (3.2.12) 式のサポート，確信度の入力項目を満たす相関ルールを導出すると，図 3.2 で示しているように，

$$\{\text{ディスパッチング・ルール}\} \implies \{\text{ディスパッチング・ルール}\}$$

$$\{\text{ディスパッチング・ルール}\} \implies \{\text{評価基準}\}$$

$$\{\text{評価基準}\} \implies \{\text{ディスパッチング・ルール}\}$$

$$\{\text{評価基準}\} \implies \{\text{評価基準}\}$$

$$\begin{array}{ccc}
\{EDD\&EDD\} & \Longrightarrow & \{EDD\&SPT\} \\
\{EDD\&EDD, SPT\&SPT\} & \Longrightarrow & \{SLACK\&EDD, SLACK\&SLACK\} \\
\{SLACK\&SPT\} & \Longrightarrow & \{T_{max}\} \\
\{T_{max}\} & \Longrightarrow & \{SLACK\&SPT\} \\
\{\bar{F}\} & \Longrightarrow & \{T_{max}\} \\
& \vdots & \\
& \vdots &
\end{array}$$

図 3.2: 相関ルール出力例

となる．

本研究において現状として意味のある相関ルールは，(3.2.5) 式に示したようなもの，すなわち

$$\{\text{評価基準}\} \Longrightarrow \{\text{ディスパッチング・ルール}\}$$

であり，(2.3.1) 式の相関ルール「 $X \Longrightarrow Y$ 」に置き換えると，

$$\begin{aligned}
X &= \{T_{max}\} \text{ または } X = \{\bar{F}\} \\
Y &= \{EDD\&EDD, \dots, SPT\&SPT\}
\end{aligned}$$

である．

本研究では，本章で述べてきた手法，すなわちデータ・マイニング適用時に「同等」という概念を用いて相関ルールを導出する手法を「ルール・マイニング」手法と呼ぶ．

第 4 章

ルール・マイニングの適用例

この章では、3 章で示したシミュレーションモデルの出力値にルール・マイニングを適用させる場合の相関ルールとサポート値に関して考察する。

サポート値に対する考察

3.1 節にて示されているシミュレーションモデルを、(3.1.1) 式で示されている納期設定式の納期係数 $\alpha = 0.4$ 、シミュレーション実行回数 $N = 1$ にて実行する。その実行結果の各納期遅れは、

$$\begin{aligned} & (EDD\&EDD, EDD\&SLACK, EDD\&SPT, \\ & SLACK\&EDD, SLACK\&SLACK, SLACK\&SPT, \\ & SPT\&EDD, SPT\&SLACK, SPT\&SPT) \\ & = (1517, 1517, 1272, 1517, 1517, 1687, 2528, 2528, 2208) \end{aligned}$$

となった。この結果において、最大納期遅れの最小値は $EDD\&SPT$ を適用したときに 1272 であり、 $\beta = 20$ として同等を導入した場合のトランザクション $\bar{D}_e$ は、

$$\bar{D}_e = \{T_{max}, EDD\&SPT\} \quad (4.1.1)$$

となる。この $\bar{D}_e$ の表記を次のように変形する。

$$\bar{D}_e = \{1, 0, 0, 1, 0, 0, 0, 0, 0\} \quad (4.1.2)$$

各々の要素は図 4.1 で示す．第 1 要素は，トランザクション $\bar{D}_e$ の場合に「最大納期遅れ」，トランザクション $\bar{D}_s$ の場合に「平均滞留時間」を示す．

(4.1.2) 式の右辺第 1 要素の 1 は T_{max} について考えていることを示し，第 2 要素以降は各ディスパッチング・ルールの組み合わせに対応している．当然のことではあるが，(4.1.2) 式は自分自身と同等であるという結果を示している．すなわち，4 番目の 1 は， $EDD\&SPT$ というディスパッチング・ルールの組み合わせが選択されたことを示す．0 は選択されなかったという意味である．例えば $\beta = 300$ とすれば，

$$\bar{D}_e = \{1, 1, 1, 1, 1, 1, 1, 0, 0, 0\}$$

となり，最初の 6 つのディスパッチング・ルールの組み合わせは， T_{max} に関して， $\beta = 300$ のもとで，同等となっていることを示している．

本研究は，このようにして得たトランザクションに対して，適当なサポートおよび確信度を指定して相関ルールを求めることになる．ここでは，このような考え方の妥当性を検証するために，3.1 節で示しているシミュレーション実行単位で，(3.1.1) 式の納期係数 α を 0.4 ~ 1.2 まで 0.1 刻みで変更していき，それぞれの納期設定で一様乱数の乱数系列を変更させながら $N(= 1000)$ 回行い，出力されたデータ群でのサポートの変化状況を観測する．つまり表 4.1，図 4.2 に示している最大納期遅れ T_{max} を評価基準としたサポートについて検討する．

本来は，同等を導入したトランザクションに最小サポート，最小確信度を入力項目としてデータ・マイニングを適用し相関ルールを導出するのであるが，入力する最小サポート値と導出される相関ルール群の関係が判明していないために，本来とは逆の手順を用いてサポート値と相関ルールとの関係を見る．つまり，同等の尺度 β を固定したトランザクションより導出されるであろう相関ルールのサポート値を計算して，評価指標値であるサポートの変化を見る．

表 4.1 から分かるように，各マシンに EDD 又は $SLACK$ を適用した場合のサポートが常に上位に位置している．例えば，(3.1.1) 式の納期係数 $\alpha = 0.4$ ，同等の尺度 $\beta = 20$ において $SLACK\&SLACK$ のサポートは 0.702 である．すなわち，ここでの全シミュレーション・データの 7 割が，この組み合わせのディスパッチング・ルールが良いと結論していることと解釈できる．このことはまた，(3.1.1) 式で $\alpha = 0.4$ ，同等の尺度 $\beta = 20$ ，さらに最小サポート = 0.7 と設定すると，

$$T_{max} \implies \{SLACK\&SLACK\} \quad (4.1.3)$$

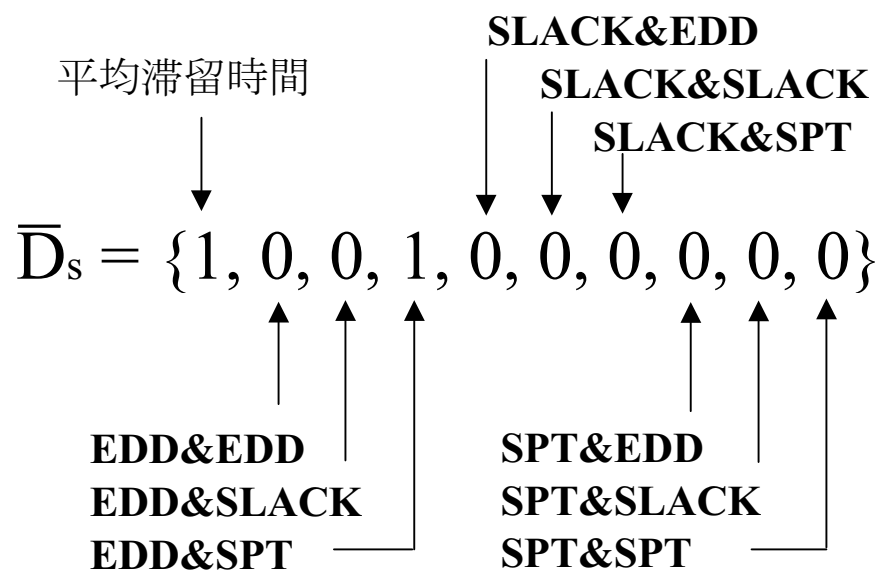
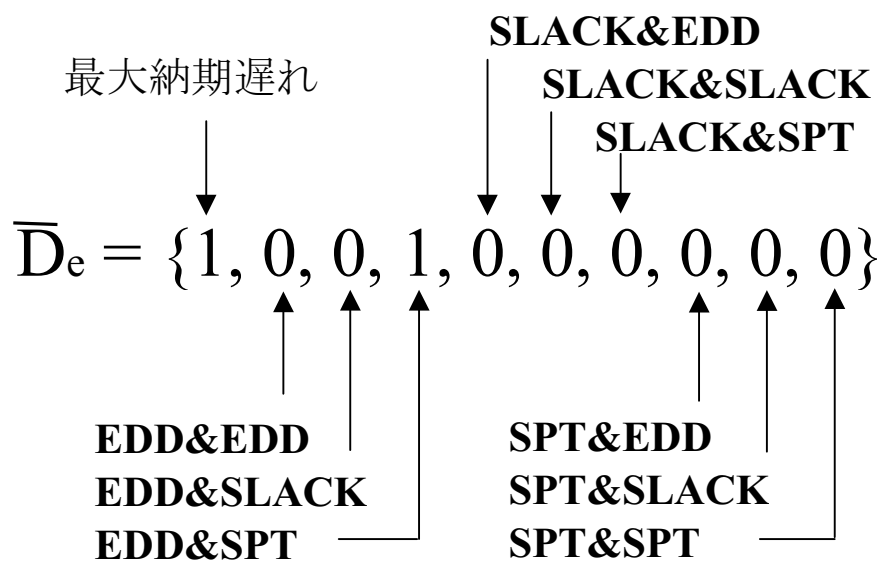


図 4.1: トランザクション

Rule	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2
EDD&EDD	0.424	0.554	0.755	0.921	0.986	0.999	1.0	1.0	1.0
EDD&SLACK	0.554	0.664	0.814	0.944	0.989	0.999	1.0	1.0	1.0
EDD&SPT	0.228	0.198	0.227	0.401	0.63	0.779	0.887	0.945	0.981
SLACK&EDD	0.579	0.668	0.817	0.953	0.993	0.999	1.0	1.0	1.0
SLACK&SLACK	0.702	0.788	0.871	0.975	0.996	1.0	1.0	1.0	1.0
SLACK&SPT	0.368	0.283	0.279	0.438	0.637	0.782	0.887	0.944	0.981
SPT&EDD	0.256	0.184	0.216	0.397	0.6	0.765	0.874	0.941	0.971
SPT&SLACK	0.366	0.267	0.256	0.418	0.604	0.763	0.872	0.941	0.97
SPT&SPT	0.086	0.038	0.019	0.08	0.271	0.518	0.739	0.878	0.947

表 4.1: 納期係数 α 毎の最大納期遅れサポートデータ ($\beta = 20$)

という相関ルールを得るということができると解釈してもよい。

また $\beta = 20$ で固定し，納期係数を $\alpha = 1.1$ と上げれば，相関ルール

$$\begin{aligned}
T_{max} \implies \{ & EDD\&EDD, EDD\&SLACK, EDD\&SPT, \\
& SLACK\&EDD, SLACK\&SLACK, SLACK\&SPT, \\
& SPT\&EDD, SPT\&SLACK, SPT\&SPT \}
\end{aligned} \tag{4.1.4}$$

のサポート値がサポート > 0.8 となることが分かる．つまり，ルール・マイニング適用時に入力項目である最小サポートを 80% とすれば，(4.1.4) 式の相関ルールが導出される．

そして， $\beta = 20$ ，最小サポート 70% とし，納期係数 α を $0.4 \rightarrow 1.1$ と変化させたときに相関ルールが (4.1.3) 式から (4.1.4) 式へ変化することより，今まで経験則より知られていた「納期の設定を緩やかにすると，納期がファクターとなる評価基準が，ディスパッチング・ルールに依存しなくなる」ことが，数学的解析を用いることなく，評価基準のひとつ「最大納期遅れ」に関しては，評価指標値であるサポートを用いて経験則と同様のことが言える．

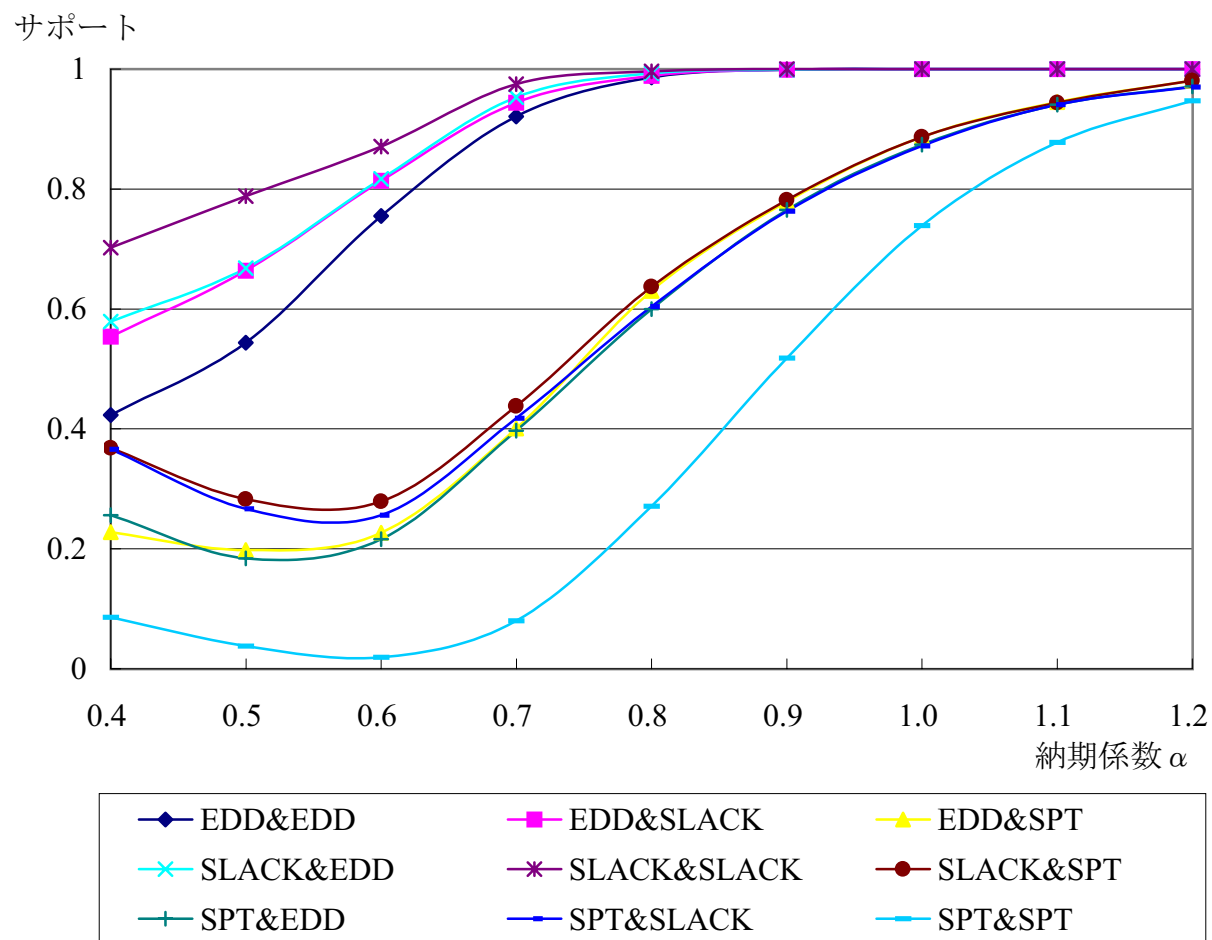


図 4.2: 最大納期遅れサポートグラフ ($\beta = 20$)

第 5 章

ルール・マイニングの評価

この章では，今まで述べてきたルール・マイニング法の評価をするために，2 マシン 2 工程フローショップ問題，5 マシン 5 工程ジョブショップ問題，そして後者のモデルで吟味されている複合ルール問題に適用を試みる．

5.1 ジョンソン則との比較

—2 マシン 2 工程フローショップ問題—

フローショップでは直列に m 台の異なるマシンが並び， n 個のジョブを最初の工程から順次処理するシステムである．一般に， m 段の操作を経て完了する．すなわち，全ジョブの機械にかかる生産順序が同じである流れ作業の生産形式である．

5.1.1 ジョンソン則

メイクスパンの最小化を目的とする 2 マシンの場合のフローショップスケジューリング問題はジョンソンによって厳密に解かれていて，この種の問題をジョンソン問題 [4][13] という．

ジョンソン則のアルゴリズム：（図 5.1 に示す）

ステップ 1：投入順序が決定されていないジョブリストの中から，処理時間の最小のものを見つける．対象となる処理時間は第 1 工程，第 2 工程の隔ては無く，二つ以上該当等する場合は任意のものを選択する．

ステップ 2： 選択された処理時間が，第 1 工程に關与するものであれば，そのジョブを投入順序の初めに順序付けし，それが第 2 工程に關与するものであれば，投入順序の終わりに順序付けする．そして，そのジョブをジョブリストから取り除く．

ステップ 3： リストが空であれば，最適ジョブ順序が得られ，その順序通りに処理開始となる．そうでなければ，ステップ 1 へ返る．

5.1.2 シミュレーションモデルの基本仕様と出力

1. フローショップモデルである．ショップ内には 2 台の生産設備があり，各ジョブは工程数が 2 で，工程設計は表 3.1 で示してある工程設計 ID=3 のみを用いる．
2. 生産の対象ジョブを 20 ヶ発生させ，そのジョブはショップに時刻ゼロ ($e_i; i = 1, 2, \dots, 20$) で全て到着している．
3. 各工程の処理時間は，表 3.2 の中の第 1 工程処理時間のみを用い，各ジョブの納期設定方法は (3.1.1) 式を用いる．
4. 各マシンは，シミュレーション実行中は常に使用可能であり，故障などの要因は考えない．
5. 各マシンで適用するディスパッチング・ルール集合 Z は次の集合とする．

$$Z = \{EDD, SLACK, SPT, Johnson\}$$

すなわち，今までの 9 種類のディスパッチング・ルールの組み合わせにジョンソン則 (ルール) を加えた 10 種類のディスパッチング・ルールが存在する．ただし，ジョンソン則は 2 工程を対象としており，単一のマシンに適用する事は出来ないもので，今までの表記にあるような， $EDD \& Johnson$ という組み合わせは存在しない．

6. シミュレーションの実行単位は，同一の属性値を持った 20 ジョブに対し，10 種類のディスパッチング・ルールをすべて適用した場合である．このようなシミュレーション実行単位で，各々異なる 20 ジョブの集合に対して 1000 回実施する．
7. 出力項目に評価基準の最大滞留時間 F_{max} を加える．全ジョブがショップに時刻ゼロで到着しているので， $F_{max} = \text{メイクスパン}$ となる．

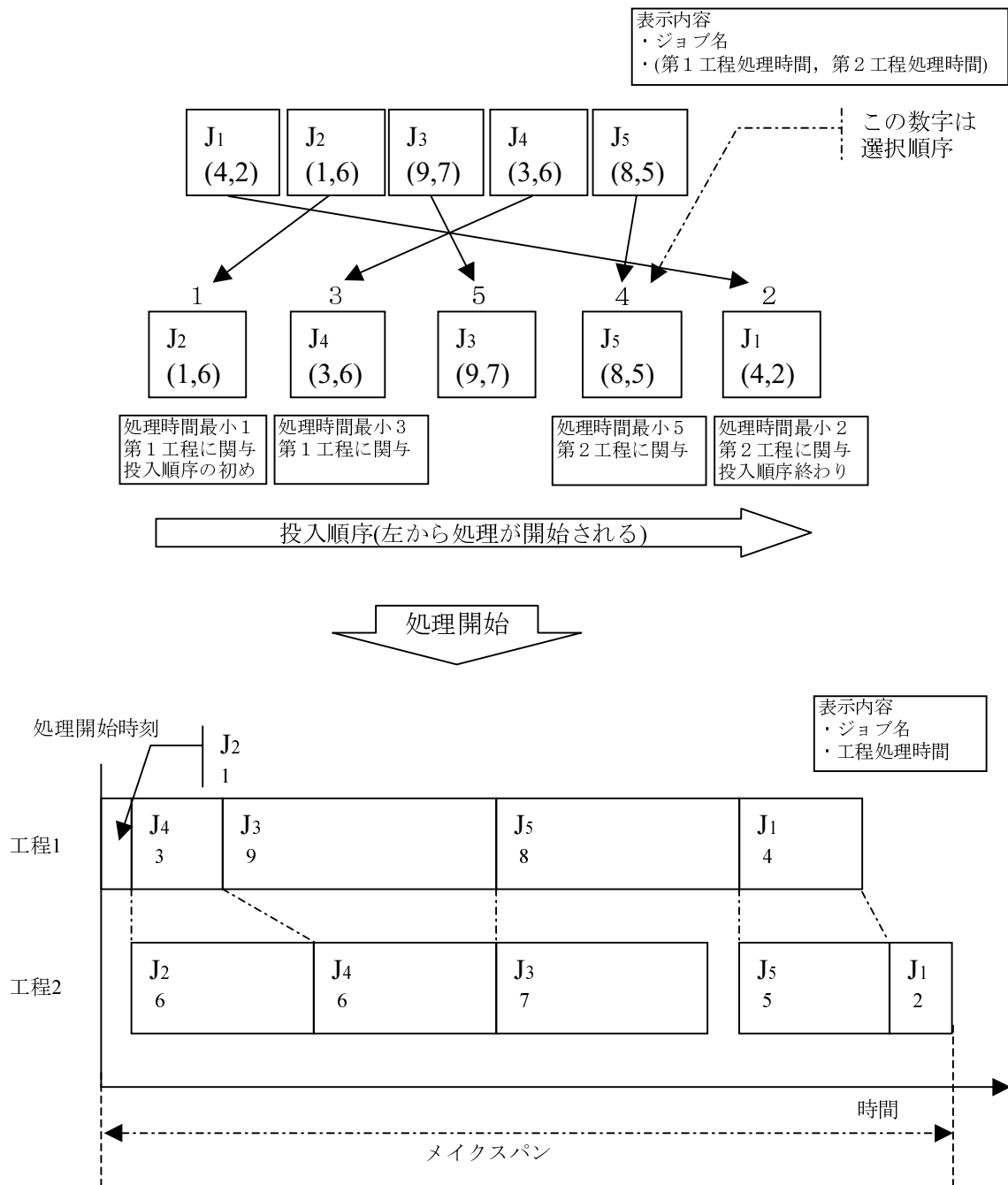


図 5.1: ジョンソン則での投入順序例と工程処理例

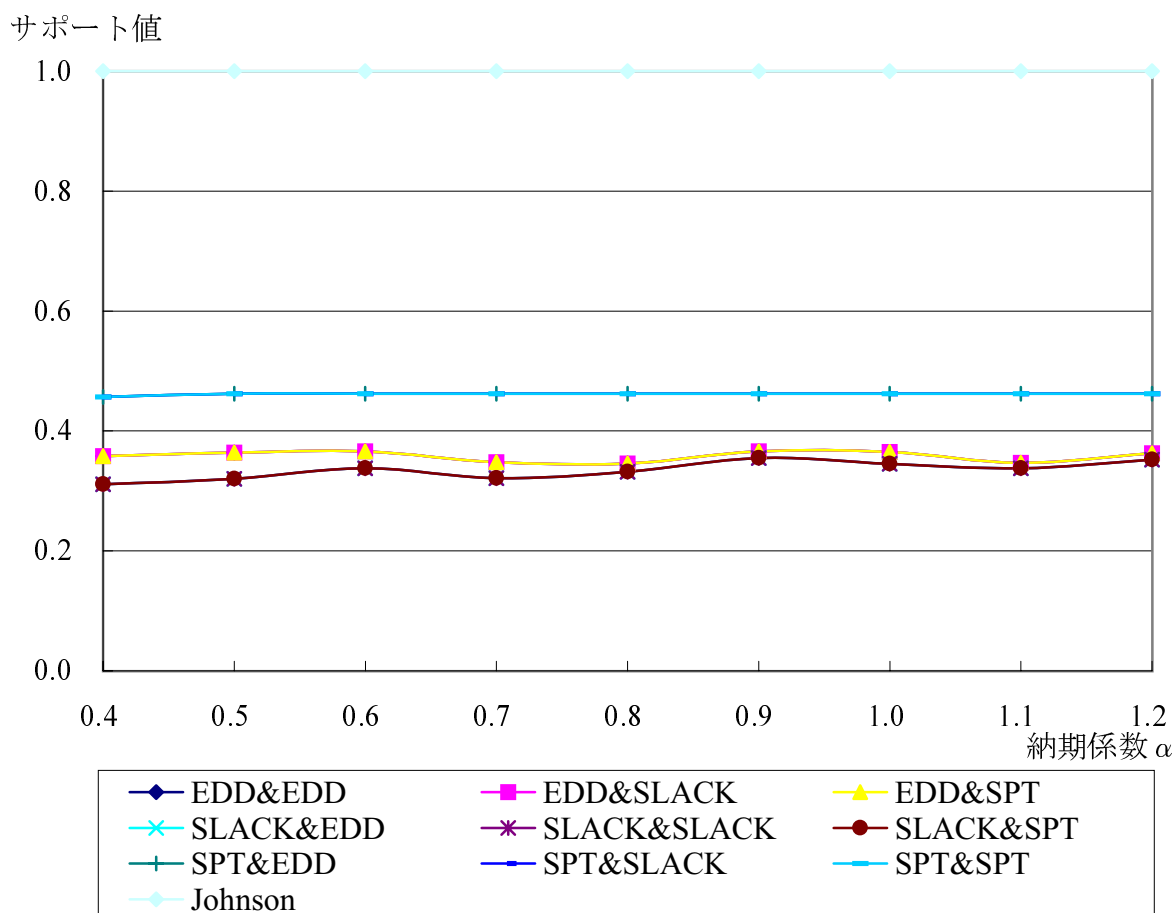


図 5.2: 最大滞留時間サポートグラフ ($\beta = 0$)

この基本仕様でシミュレーションを実行し，4章にて述べた手法を用い，その結果を図 5.2 に示す．

図 5.2 は，(4.1.2) 式と同様の考えにより，左辺第 1 フラグがメイクスパンを指すと考え，また，新たに 11 番目のフラグを追加し，そのフラグをジョンソン則用のフラグとすれば，

$$\{1, *, *, *, *, *, *, *, *, *, 1\} \quad (5.1.1)$$

がすべてのトランザクションにおいて成り立っていることを意味している．ここで $*$ は任意の 1,0 を示しているものとする．すなわち，全データがメイクスパンに関してジョンソン則を支持しているので，サポート = 1.0，同等の尺度 $\beta = 0$ で以下の相関ルールが導出

される．

$$F_{max} \implies \{Johnson\}$$

この時の β の値が 0 であることから，今回のシミュレーション出力群でのメイクスパンに関する最適解は，ジョンソン則である，と言え，ジョンソンの証明とは別のアプローチから，ジョンソン則の有効性が観測される．

図 5.2 では，納期係数 α を変更しても，各ディスパッチング・ルールの組み合わせに対し，サポート値の変化がほとんど見られない．これは，2 マシンフローショップであるために，シミュレーションモデル内のジョブ選択が，第 1 工程のマシン 1 のディスパッチング・ルールに依存してしまい，*Johnson* , *EDD* , *SLACK* , *SPT* の 4 種のグラフに収束してしまうためである．

5.2 5 マシン 5 工程ジョブショップ問題

5 マシン 5 工程ジョブショップモデルは文献 [11] のモデルを用いる．福井らは，複数のディスパッチング・ルールをモデルに適用し，そのシミュレーション結果の評価基準を比較検討して，複合ルールへの候補を示している．その比較検討している部分に，本研究の特色であるルール・マイニングを適用し，その適用結果を考察する．

5.2.1 シミュレーションモデルの基本仕様

1. ショップ内には 5 台の生産設備があり，各ジョブは処理工程数が 5 で，それらのすべてを通過する．またジョブの工程順序は，ランダムである．
2. 全マシンのディスパッチング・ルールは同じものを使用する．そのディスパッチング・ルールはシミュレーションに際し，入力項目である．
3. 生産の対象ジョブがショップに断続的に到着する動的モデルであり，ジョブの到着時間間隔は指数分布に従う．
4. 各工程の処理時間は，1 ~ 9 の一様分布に従う乱数で与えられ，段取り時間は処理時間に含まれる．

5. 納期は，総処理時間に 3～8 の一様分布に従う乱数を乗じて決定される．
6. 各マシンは，シミュレーション実行中は常に使用可能であり，故障などの要因は考えない．

以上のようなジョブショップスケジューリング問題でディスパッチング・ルールを $D.DATE, SLACK$ を中心にして $SPT, FCFS$, 複合ルールを種々試み，その結果を考察する．その後 CONWAY の総処理時間の 9 倍を納期と設定する TWK (Total-WorK due dates) 法 [13] と比較検討するために，納期設定法を TWK 法に変更する．

シミュレーションモデルの負荷率 (ρ) は以下の様に設定される．

$$\rho = \frac{\text{ジョブの各工程での加工時間の平均}}{\text{ジョブのショップへの到着時間間隔の平均}} \quad (5.2.1)$$

(5.2.1) 式より

$$\begin{aligned} & \text{ジョブのショップへの到着時間間隔の平均} \\ &= \frac{\text{ジョブの各工程での加工時間の平均}}{\rho} \end{aligned} \quad (5.2.2)$$

となる．この (5.2.2) 式から，指数分布にて与えられるジョブの到着時間間隔の平均が求まる．すなわち，負荷率 (ρ) がシミュレーションモデルの入力項目のひとつになる．

上記のようなシミュレーションモデルに対して，ジョブ 10,000ヶ投入を 1 回のシミュレーション単位として種々のディスパッチング・ルールで実行し，負荷率を 60～90% と変化させる．そのシミュレーションの出力項目は，以下の 6 つとする．

$$\begin{array}{ll} \text{平均滞留時間} (\bar{F}) & \text{滞留時間の標準偏差} (\sigma_{F_i}) \\ \text{平均納期遅れ} (\bar{T}) & \text{最大納期遅れ} (T_{max}) \\ \text{納期遅れ個数} (Nt) & \text{総納期遅れ時間} (\sum T_i) \end{array}$$

5.2.2 複合ルールへの候補とシミュレーション結果

文献 [11] によれば，ディスパッチング・ルールを $D.DATE, SLACK, SPT, FCFS$ と適用し，負荷率を 60,70,80,90% としたシミュレーションの出力結果 ($\bar{F}, \sigma_{F_i}, \bar{T}, T_{max}, Nt, \sum T_i$) より，納期をファクターとした評価基準に対して，負荷率 90% の Nt の場合のみ $SLACK$ ルールが $D.DATE$ ルールよりも良い結果を与えているが，その他の場合では $D.DATE$ が他の 3 種のディスパッチング・ルールより良い結果を与えていると報告している．

この結果から福井らは、*SLACK* ルールから派生したディスパッチング・ルール *S/OPN* より *D.DATE* ルールから派生したディスパッチング・ルール *D/OPN* がより良好で、複合ルール組み合わせとして CONWAY[13] が示した *SPT* と *S/OPN* より、*SPT* と *D/OPN* が良いのではないかと考察を与えた。その考察内容を確認するために、納期設定法を TWK 法に変更し、(2.1.8) 式の重み係数 a を 0.1 から 0.9 まで 0.1 刻みに変更する *SPT* と *D/OPN* の複合ルールと CONWAY の示した複合ルール $0.5(SPT) + 0.5(S/OPN)$ を適用し、負荷率 90% にてシミュレーションを実行した。シミュレーション実行結果より、CONWAY の提案した $0.5(SPT) + 0.5(S/OPN)$ よりも、文献 [11] で提案した *SPT* と *D/OPN* の複合ルールが納期をファクターとした評価基準に対して優位にあり、そして平均滞留時間と分散に対しても改善が見られた。

文献 [11] では、複合ルールの重み係数を変更していくなかで、重み付けを *SPT* に (0.7 ~ 0.8)、*D/OPN* に (0.3 ~ 0.2) とする複合ディスパッチング・ルール、すなわち、

$$0.7(SPT) + 0.3(D/OPN) \quad (5.2.3)$$

$$0.8(SPT) + 0.2(D/OPN) \quad (5.2.4)$$

が他の組み合わせに比べて良好な結果であることが報告されている。

5.3 複合ルールの重み付けへのルール・マイニング法の適用

文献 [11] での重み係数を段階的に変更しながら、各々重み付けされた複合ルールの評価基準を比較しての発見手法に、ルール・マイニング法の適用を考察する。ルール・マイニング法は、同等を導入したトランザクションに最小サポート、最小確信度を入力項目としてデータ・マイニングを適用し相関ルールを導出するのであるが、入力する最小サポート値と導出される相関ルール群の関係が判明していないために、本来とは逆の手順を用いてサポート値と相関ルールとの関係を見る。つまり、同等の尺度 β を固定したトランザクションより導出されるであろう相関ルールのサポート値を計算して、評価指標値であるサポートの変化をみる。複合ルールへの候補は、5.2.2 節にて示されている、*SPT*、*D/OPN* を用いる。

この場合のシミュレーションモデルは 5.2.1 節の基本仕様に従う。しかし、出力項目は滞留時間と納期ずれを用い、負荷率は 90% のみ、納期設定は TWK 法を用いる。そして 5 マシンすべてに *SPT*、*D/OPN* を適用させ、ディスパッチング・ルール毎に 10,000 ジョブ

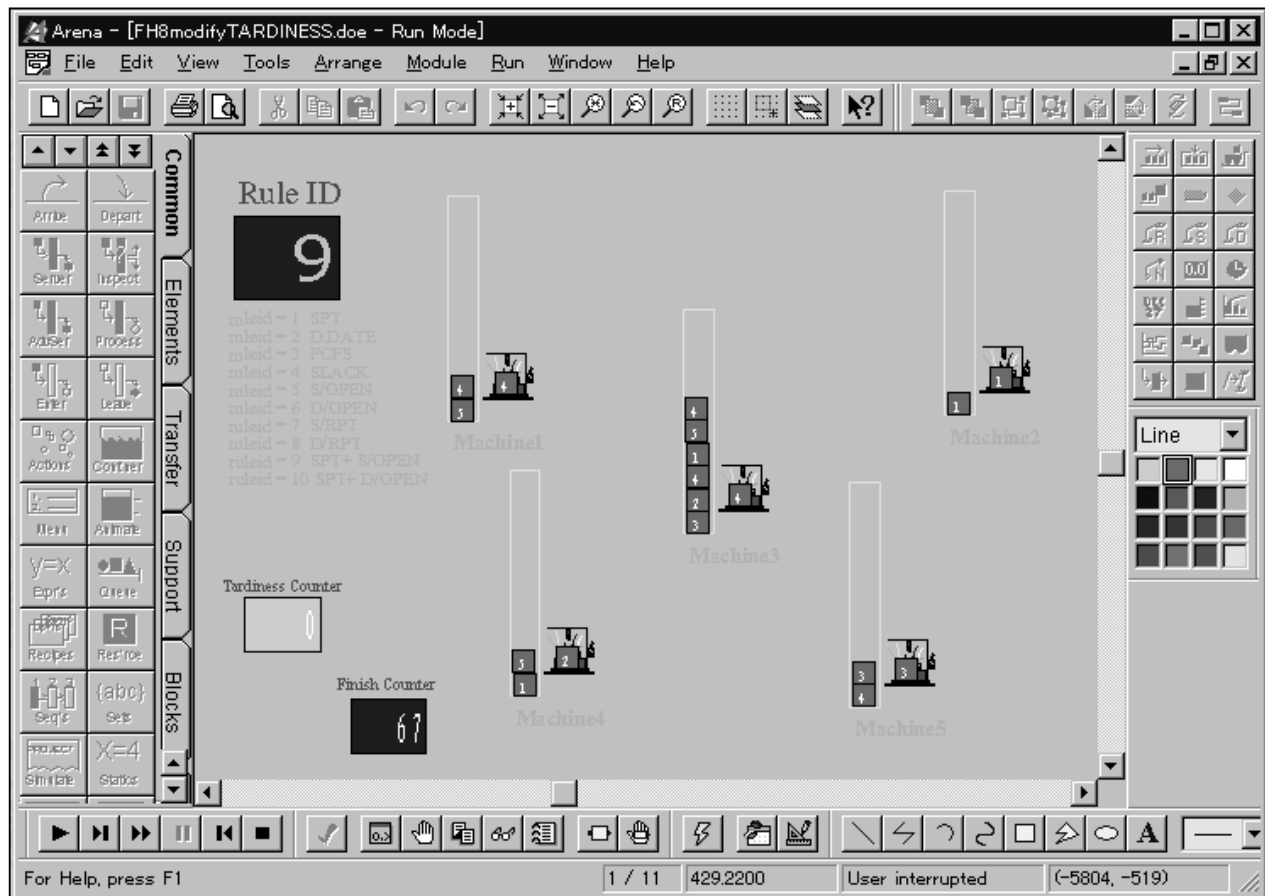


図 5.3: ARENA によるシミュレーション実行例

を投入する．10,000 ジョブはそれぞれユニークな ID を持ち，シミュレーションモデルが使用するディスパッチング・ルールが異なっても，同一 ID を持つジョブは同一属性，同一属性値を持つ．シミュレーションの実行例を図 5.3 に示す．出力はディスパッチング・ルール毎の各々ジョブの滞留時間，納期ずれであり，図 5.4 で示すトランザクション作成のために外部記録する．

二つのディスパッチング・ルールを適用した際の滞留時間，納期遅れを記録したファイルよりデータ・マイニングを適用するために，トランザクションを作成する．その場合の各フラグの内容を図 5.5 に示す．

SPT , *D/OPN* を示す項目は，各々の出力項目である滞留時間，納期ずれに関して，優位な結果を示したディスパッチング・ルールに対しフラグを立てる．なお，両ディスパッ

相関ルール	サポート値
滞留時間 $\Rightarrow SPT$	0.75
滞留時間 $\Rightarrow D/OPN$	0.25
納期ずれ $\Rightarrow SPT$	0.76
納期ずれ $\Rightarrow D/OPN$	0.24

表 5.1: 各々の相関ルールのサポート値 (小数点以下 3 位切り捨て)

チング・ルールの数値が同一ならば、両フラグを立てた。

図 5.5 のようなトランザクション集合から導出される相関ルールは、

滞留時間 $\Rightarrow SPT$

滞留時間 $\Rightarrow D/OPN$

納期ずれ $\Rightarrow SPT$

納期ずれ $\Rightarrow D/OPN$

となり、それぞれのサポート値を表 5.1 に示す。

ここでサポート値が示している内容を考えると、納期ずれに関しては全体の 76% が SPT を優位、24% が D/OPN を優位と示され、滞留時間に関しては全体の 75% が SPT を優位、25% が D/OPN を優位と示されている。このことから、

$$0.75(SPT) + 0.25(D/OPN)$$

$$0.76(SPT) + 0.24(D/OPN)$$

の複合ルールが良いのではないかと考えられる。

例として、重み係数を 0.4 ~ 0.9 の 0.1 刻みでの出力と、0.75 の場合の出力を表 5.2 に示す。重み係数が大きくなれば、 SPT ルールにより重みをおく複合ルールになる。

表 5.2 より、 T_{max} に関しては重み係数 0.75 の結果は、同じく 0.7, 0.8 の場合よりも良くない結果となったが、その他の項目をみる限りでは、重み係数 0.75 の結果は良好と思われる。よって、この複合ルールの重み係数の割り振り方法を 0.1 刻みにすると、前節にて述べているように、

$$0.7(SPT) + 0.3(D/OPN) \tag{5.3.1}$$

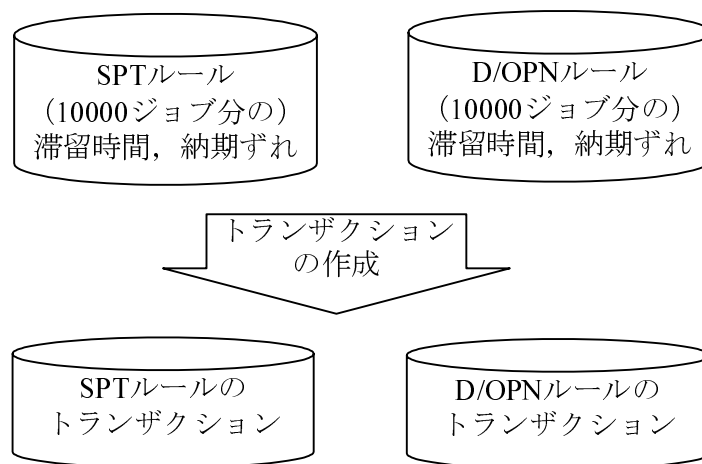


図 5.4: 滞留時間，納期遅れ時間からトランザクションの作成

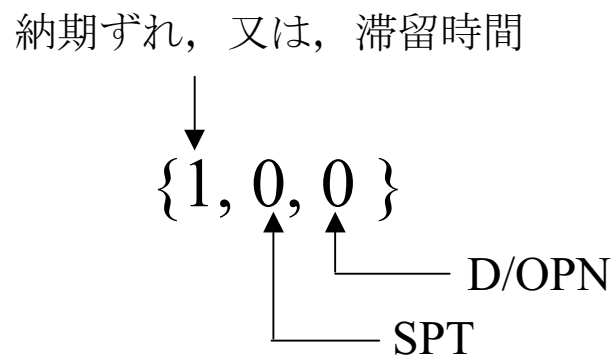


図 5.5: トランザクションの内容

重み係数	$\bar{F}$	σ_{F_i}	$\bar{T}$	T_{max}	Nt	$\sum T_i$
0.4	112.3	54.9	2.64	87.8	230	2635.3
0.5	110.8	55.5	3.18	45.2	308	3184.6
0.6	110.2	55.4	2.30	50.4	261	2291.1
0.7	106.9	55.1	2.42	51.1	231	2415.2
0.75	104.6	55.1	2.20	90.4	188	2202.9
0.8	102.2	55.5	2.44	58.0	216	2442.4
0.9	97.9	57.7	6.97	94.0	414	6970.8

表 5.2: 重み係数毎の評価基準の出力

$$0.8(SPT) + 0.2(D/OPN) \quad (5.3.2)$$

の複合ルールを示すようになる．

すなわち，段階的に重み係数を変更していきながら，シミュレーションを行い，それぞれの出力結果を見比べて重み係数を決定する方法とは，別のアプローチで重み係数を求める事が出来ることを示した．

5.4 ルール・マイニング法の適用 - その 1 -

前節にて示されたシミュレーションモデルでは，モデル内のマシンすべてが同一のディスパッチング・ルールを用いるものであったが，これをマシン毎に用いるディスパッチング・ルールを個別に設定できるように変更する．この場合のディスパッチング・ルールの候補は，すべて (5.3.1) 式，(5.3.2) 式に示されているような複合ルールとする．文献 [11] より， SPT と D/OPN の複合ルール係数の重み付けに対し， SPT に重みをおくルールが， D/OPN に重みをおくルールよりも良好な結果を残していることから，その複合ルールの候補は，表 5.3 の SPT に重みを掛けている 4 種を用いる．

5 マシンにそれぞれ 4 種のディスパッチング・ルールを適応させるのだから，ディスパッチング・ルールの組み合わせは 1024 通りになる．この 1024 通りの組み合わせを，(5.4.1) 式のように表記する．

$$(1, 2, 3, 2, 1) \quad (5.4.1)$$

ルール ID	ディスパッチングルール
1	$0.6(\text{SPT}) + 0.4(\text{D}/\text{OPN})$
2	$0.7(\text{SPT}) + 0.3(\text{D}/\text{OPN})$
3	$0.8(\text{SPT}) + 0.2(\text{D}/\text{OPN})$
4	$0.9(\text{SPT}) + 0.1(\text{D}/\text{OPN})$

表 5.3: 使用する複合ルール

これは，第 1 桁（最左桁）がマシン 1 を示し，そこに記入されている数値は，表 5.3 に示されているルール ID に相当する．(5.4.1) 式では，マシン 1 とマシン 5 がルール ID1 を，マシン 2 とマシン 4 がルール ID2 を，マシン 3 がルール ID3 を用いることを表している．

シミュレーションモデルにジョブを 1000 ケ投入し，各々のディスパッチング・ルール 1024 組にてシミュレーションを行う．これを 1 回のシミュレーション実行単位として，工程毎の処理時間等を設定するための乱数系列を 100 ケ用意し，それぞれシミュレーションを行う．つまり，属性値が異なるジョブを 100 セット（1 セットは 1000 ジョブ）用意して，各々のジョブセット毎にシミュレーションを実行するため，シミュレーション実行回数は 100 回となる．そしてシミュレーション実行単位毎に，相反する評価基準である平均納期遅れと平均滞留時間を出力する．その他のシミュレーションモデルの仕様は，5.2.1 節の基本仕様に従う．100 セットのジョブをシミュレーションした出力結果にルール・マイニング法を適用して，二つの評価基準を満足する相関ルールを考察する．

同等を適用する際の平均納期遅れ ($\bar{T}$) に関する尺度を β とし，平均滞留時間 ($\bar{F}$) に関する尺度を γ とする．つまり，(3.2.3) 式，(3.2.4) 式に当てはめると， T_{max} を $\bar{T}$ に変更， β, γ はそのまま尺度として与えられる．このときの $e \in Z \times Z \times Z \times Z \times Z$ ， $s \in Z \times Z \times Z \times Z \times Z$ は，表 5.3 に示されているディスパッチング・ルールに対して，シミュレーション実行単位毎の 1024 通りの組み合わせ中で，評価基準 $\bar{T}$ ， $\bar{F}$ の最小値を示したルールの組み合わせとなる．

$$\begin{aligned}
D_e &= \{z \in Z \times Z \times Z \times Z \times Z \mid z = \frac{e}{(\beta, \bar{T})}\} \\
D_s &= \{z \in Z \times Z \times Z \times Z \times Z \mid z = \frac{s}{(\gamma, \bar{F})}\} \\
Z &= \{0.6(\text{SPT}) + 0.4(\text{D}/\text{OPN}), 0.7(\text{SPT}) + 0.3(\text{D}/\text{OPN}), \\
&\quad 0.8(\text{SPT}) + 0.2(\text{D}/\text{OPN}), 0.9(\text{SPT}) + 0.1(\text{D}/\text{OPN})\}
\end{aligned}$$

$$0.8(SPT) + 0.2(D/OPN), 0.9(SPT) + 0.1(D/OPN)\}$$

例として，文献[11]で良好と示されたディスパッチング・ルールの内，(5.2.3)式で示されているルールの場合，つまり5マシンすべてに，複合ルール $0.7(SPT) + 0.3(D/OPN)$ を適用したことを意味する組み合わせ (2,2,2,2,2) の場合について，同等の二つの尺度とサポート値の関係を図 5.6 に示す．

通常のルール・マイニング法適用手段とは逆手順を用い，本来は相関ルールを導出するための入力項目である尺度 β, γ と，評価指標値であるサポートとの関係を調べる．

相関ルール

$$\{\bar{T}, \bar{F}\} \implies \{(2, 2, 2, 2, 2)\} \quad (5.4.2)$$

が，最小サポート 0.8 で導出されるためには，尺度 β, γ は図 5.6 においてサポート値 ≥ 0.8 の領域を満足しなくてはならない．その領域を満足する尺度 β, γ は

$$\begin{aligned} (\beta, \gamma) \geq & (0.2, 17), (0.3, 16), (0.4, 15), \\ & (0.6, 14), (0.8, 13) \end{aligned} \quad (5.4.3)$$

となる．すなわち，最小サポート 0.8，(5.4.3) 式を満たす同等の尺度 β, γ の条件で，(5.4.2) 式に示される相関ルールが導出される．

(5.4.2) 式の相関ルールが導出されれば (2,2,2,2,2) のディスパッチング・ルールは，その場合の尺度 β, γ のもとに，相反する評価基準である $\bar{T}, \bar{F}$ に対して，サポート 0.8 で良好な結果を出すと言える．

5.5 ルール・マイニング法の適用 - その 2 -

同等における評価基準の次数を，1 から n へ上げることを論じるための準備として次数 2 の場合を考察する．同等を評価するために基準となる評価基準，例えば $\bar{T}$ や $\bar{F}$ の同等を論じる場合の基準値と，同等の対象物との差異を取り，図 5.7 のような，それぞれ縦軸，横軸で表した散布図を考える．

図 5.7 のなかで正方形で囲まれた部分が，評価基準毎に同等の尺度 β, γ を用いた場合の同等の領域となる．しかし，二つの評価基準に対して，より同等の候補数を絞った形態であり，同等を一元管理するために新たな尺度として，評価基準の差異の二乗平均を指す δ (図 5.7 では原点との距離) を導入する．

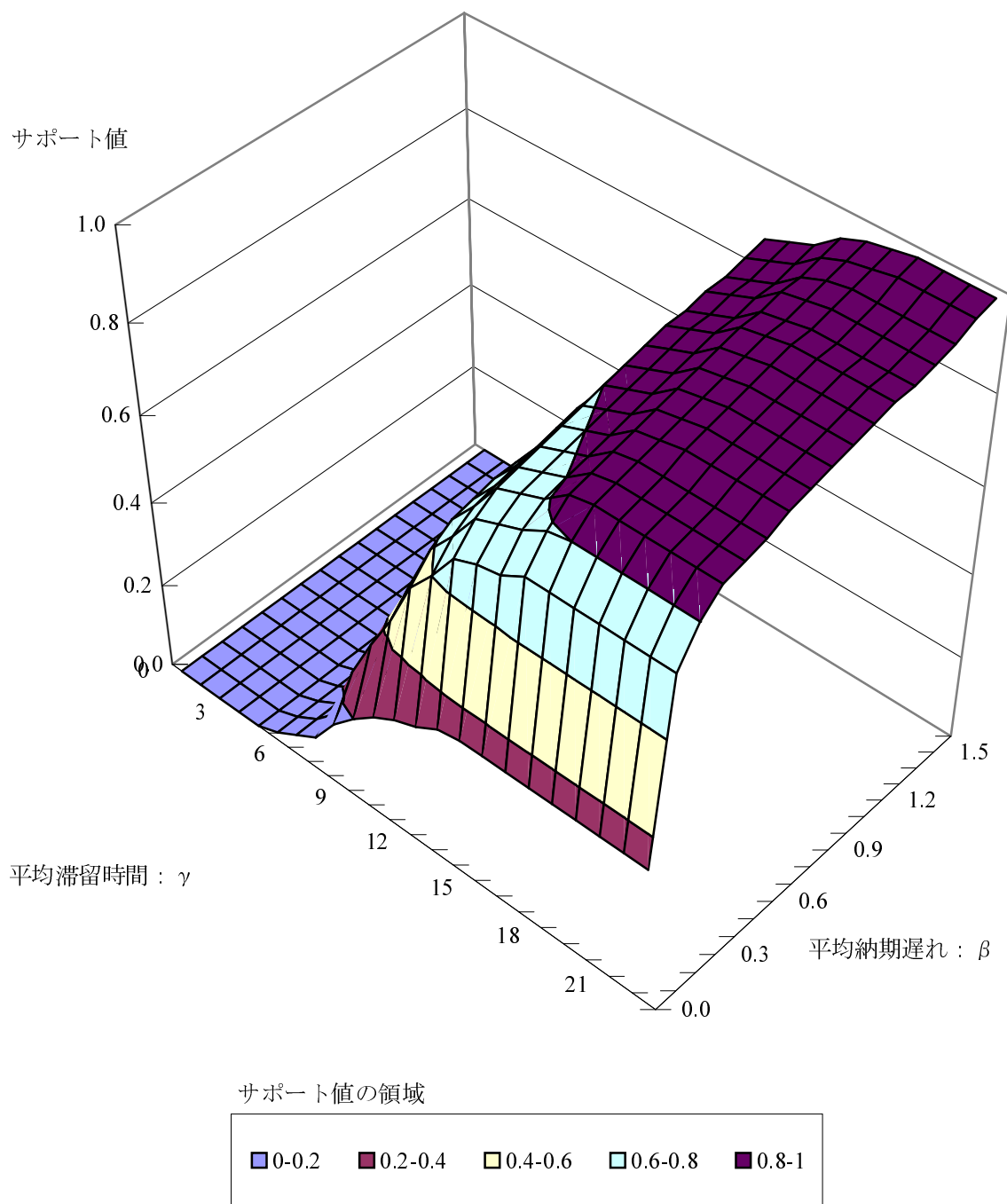


図 5.6: 組み合わせ (2,2,2,2,2) の尺度 β γ とサポート値の関係

δ の導入により，今までは β, γ の 2 枚の尺度を用いなければならなかったが，1 枚の尺度を考えるだけでよく，しかも同等の候補数を絞った形態になる．しかし評価基準によっては，時間を対象にしているものや，ジョブ個数を対象にしているもの，設備の稼働率など，同等を導入する際に与える尺度のスケールが異なるものが考えられ，二乗平均という概念では，複数の評価基準に対しての同等を導入するには少々無理が感じられる．

そこで評価基準のスケールを考慮でき，図 5.8 で示している二乗平均ではなく二つの尺度の 2 倍をそれぞれ長軸，短軸とする楕円で囲まれた物を「同等」とする考えを論じてみる．

図 5.8 にて示している概略図が，新しく二つの尺度を用いた「同等」の概念である．このモデルは先に述べた β, γ を用いてはいるが，「同等」の範囲が異なる．いままで二つの尺度を用いて論じてきた「同等」とは，図 5.8 に示された長方形に当たる部分である．この長方形で示された部分は，単に一つの尺度を用いての「同等」の融合であり，複数の評価基準を目指しているとは言いにくい．そこでより「同等」の候補を絞った形態であり，しかも二つの尺度の大きさを考慮した楕円という範囲形状が，より洗練された「同等」といえる．次に，より多数の評価基準について考えるために，二つの評価基準から三つの評価基準 P, Q, R について拡張する．そのために新規に尺度 a, b, c を用意する．

同等の定義中で用いた (3.2.1) 式を変形して，

$$\frac{P^X - P^Y}{a} \leq 1$$

とする．この式が他の評価基準 Q, R についてもいえるから，

$$\frac{Q^X - Q^Y}{b} \leq 1 \quad \frac{R^X - R^Y}{c} \leq 1$$

となる．左辺，右辺同士を 2 乗して加算すれば

$$\left(\frac{P^X - P^Y}{a} \right)^2 + \left(\frac{Q^X - Q^Y}{b} \right)^2 + \left(\frac{R^X - R^Y}{c} \right)^2 \leq 3 \quad (5.5.1)$$

となり，これは三次元空間の楕円体そのものとなる．

二つの評価基準に「同等」を導入する際には「同等」の範囲が楕円，三つの評価基準に「同等」を導入する際には「同等」の範囲が楕円体となることから，ディスパッチング・ルール X は，与えられた n 枚の尺度 $s = \{s_1, s_2, \dots, s_n\}$ と評価基準 $P = \{P_1, P_2, \dots, P_n\}$ に対して

$$\left(\frac{P_1^X - P_1^Y}{s_1} \right)^2 + \left(\frac{P_2^X - P_2^Y}{s_2} \right)^2 + \dots + \left(\frac{P_n^X - P_n^Y}{s_n} \right)^2 \leq n \quad (5.5.2)$$

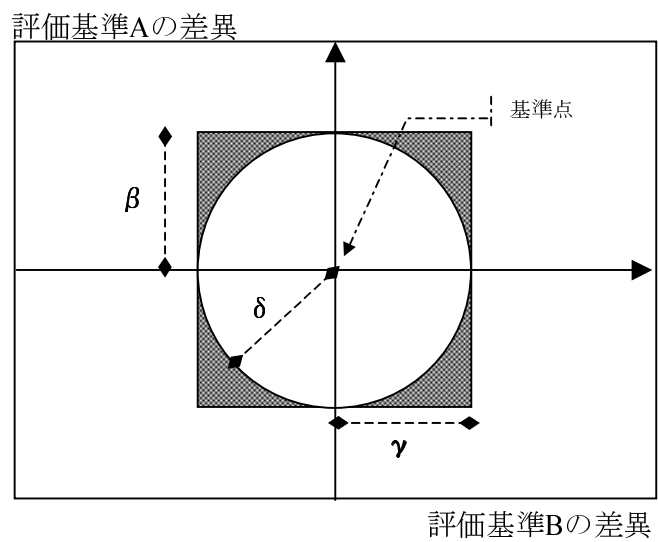


図 5.7: 新しい尺度 δ の指す領域

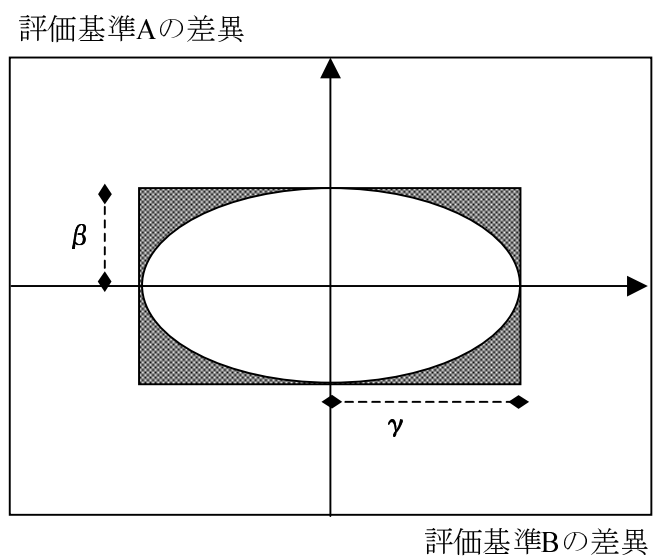


図 5.8: 二つの評価基準の「同等」の意

を満たすならば，評価基準 P に関して，ディスパッチング・ルール Y と，尺度 s において同等であるといえる．

この改良された同等を用いたルール・マイニング法により導出される相関ルールから，複数の評価基準に対して良好な結果を出すような，ディスパッチング・ルールやディスパッチング・ルールの組み合わせを求めることが出来る．

例として，文献 [11] で納期遅れに関し優位と示されている複合ルール (5.2.3) 式，(5.2.4) 式を基準に考え，そしてより優位となりうる複合ルールの組み合わせについて考える．このうち，具体例として使用する複合ルールは (5.2.4) 式の

$$0.8(SPT) + 0.2(D/OPN)$$

を用いる．

トランザクション集合，シミュレーションモデルに使用する複合ルール，評価基準，同等の尺度などは，前節と同様に

$$\begin{aligned} D_e &= \{z \in Z \times Z \times Z \times Z \times Z \mid z = \frac{e}{(\beta, \bar{T})}\} \\ D_s &= \{z \in Z \times Z \times Z \times Z \times Z \mid z = \frac{s}{(\gamma, \bar{F})}\} \\ Z &= \{0.6(SPT) + 0.4(D/OPN), 0.7(SPT) + 0.3(D/OPN), \\ &\quad 0.8(SPT) + 0.2(D/OPN), 0.9(SPT) + 0.1(D/OPN)\} \end{aligned}$$

となり，この時の e, s は，(5.2.4) 式の複合ルールを 5 マシンすべてに適用したとき，つまり (5.4.1) 式の表現を用いると，(3,3,3,3,3) の場合の平均納期遅れと平均滞留時間を指し，

$$\begin{aligned} e &= \text{組み合わせ (3, 3, 3, 3, 3) の } \bar{T} \\ s &= \text{組み合わせ (3, 3, 3, 3, 3) の } \bar{F} \end{aligned}$$

となる．

表 5.3 に示されている複合ルールを，5 マシンに各々適用して得た 1024 通りの平均納期遅れ，平均滞留時間と，(3,3,3,3,3) の平均納期遅れ，平均滞留時間，すなわち e, s との差異を取り，それぞれ縦軸，横軸で表した散布図を図 5.9 に示す．

図 5.9 の原点は，(5.2.4) 式で示されている複合ルールである (3,3,3,3,3) を示しており，その原点との差異を表しているグラフ上で，第 3 象限に存在している複合ルールの組み合わせ，例えば (3,4,4,2,1) などは，平均納期遅れと平均滞留時間において，(3,3,3,3,3) より良い結果を示している．

尺度	「同等」なる組み合わせルール数
β, γ 使用	72
δ 使用	50

表 5.4: 組み合わせルール数

(5.5.2) 式	「同等」なる組み合わせルール数
使用	46
未使用	68

表 5.5: (5.5.2) 式の使用，未使用の同等の数

文献 [11] や前々節にて，原点：(3,3,3,3,3) が表している複合ルールの組み合わせが良いとの結果を得ている．この原点：(3,3,3,3,3) と「同等」なものを考えるために， δ を用いた場合と， β, γ を用いた場合とを比較し，各々の「同等」な組み合わせ数を表 5.4 に表す．この時の δ は，原点との距離が最も離れている点 (-0.9,-0.46) が原点と同等，すなわち複合ルールの組み合わせ (3,4,4,2,1) が (3,3,3,3,3) と同等であると考え，その点までの距離

$$\delta = \sqrt{(-0.9)^2 + (-0.46)^2} = 1.01 \dots$$

を用いる．

尺度 δ を用いた同等の範囲は図 5.9 において円に囲まれた部分であり，以前の二つの尺度 β, γ を用いた同等の範囲は正方形で囲まれた部分である．表 5.4 から二つの評価基準（平均納期遅れ，平均滞留時間）に対して，より候補を絞った形態といえる．しかし，それぞれのスケールが異なるため，同等の尺度のスケールも異なれば，(5.5.2) 式を用いての同等を用いなければならない．

例えば， $(\beta, \gamma) = (0.5, 2.0)$ の場合に，(5.5.2) 式を用いた場合と，用いない場合のそれぞれの同等の数は，表 5.5 のようになり，(5.5.2) 式の有用性が判明する．そして (5.5.2) 式を用いることで，複数の評価基準を満たすディスパッチング・ルール，もしくはディスパッチング・ルールの組み合わせを示す相関ルールが導出される．

平均納期遅れの差異

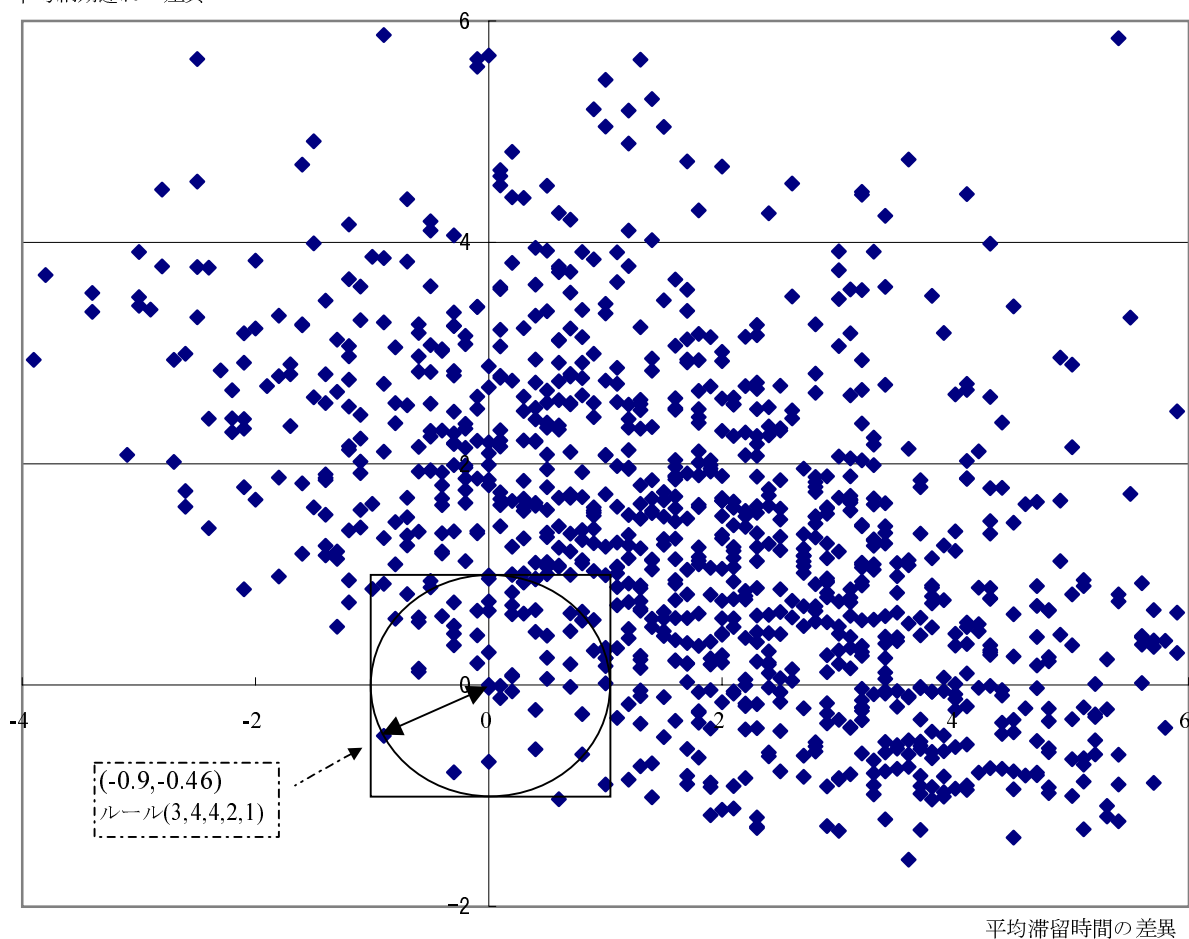


図 5.9: 平均納期遅れ，平均滞留時間の差異分布図 原点： $(3,3,3,3,3)$

第 6 章

おわりに

6.1 まとめ

本稿でのまとめとして主要なものをまとめる．

- ディスパッチング・ルールが「似ている」ことへ「同等」という定義を与え，その定義に沿って，シミュレーションにより出力した評価基準とディスパッチング・ルールで構成された出力データ群に，大規模データからの知識発掘手法のひとつであるデータ・マイニングを適用し，評価基準とディスパッチング・ルール間の相関ルールを抽出できることを示した．そして，この相関ルール抽出方法をルール・マイニング法と名付けた．この相関ルール抽出法は，ディスパッチング・ルール間の「同等」の定義は簡単ではあるが，有益であるということを示している．そして，相関ルールを意味のあるものにするために，条件部に評価基準を含めた．
- 抽出された相関ルールの有効性を，ルール・マイニング法適用時の入力項目であるサポート値を用いて定量的に示す事ができる．すなわち，そのサポート値を用いることで，試行錯誤的なディスパッチング・ルール適用手法に対して，定量的な指針により，ディスパッチング・ルールを適用できる．
- 評価基準にある尺度を用いて「似ている」ディスパッチング・ルールを抽出するために定義した「同等」を，複数の評価基準に適応させた． n 個の評価基準に「同等」を適応させるためには n 個の尺度を用いるのではあるが，その場合に各々の評価基準のスケールを加味出来るように改善した．

6.2 今後の展望

本研究で有意義と判断した相関ルールは，

$$\{\text{評価基準}\} \implies \{\text{ディスパッチング・ルール}\}$$

であるが，その他の相関ルールにも何か意義を見いだせる可能性があり，今後も考察していきたい．

複数の評価基準を対象にする「同等」の尺度の改良，そして生産シミュレーションとルール・マイニング手法の融合した形での生産スケジューラーへのアプローチを今後の課題としたい．

謝辞

本研究を進めるにあたり，適切な御指導，御助言を頂きました藤田助教授に深く感謝いたします．

知識科学研究科 吉田武稔助教授には，本研究について多大な御指導と御助言をしていただき，私を導いて下さいました．ここに感謝の意を表し，心よりお礼申し上げます．

さらに，望山助手をはじめ，藤田研究室の皆様には種々な面でお世話になりました．ここに深い感謝の意を表します．

最後に，私を快く大学院に留学させて頂き，貴重な機会を与えて頂いたうえ，生活面でも全面的に支援下さいましたコマツソフト株式会社に心より感謝いたします．

1999 年仲春 情報科学棟にて
東崎 秀行

参考文献

- [1] 井上 一郎, 冬木 正彦, “ノウハウ活性化シミュレーション法”に基づく生産スケジューリング業務支援, オペレーションズ・リサーチ, vol40, No.3, pp159-164(1995)
- [2] M.J.Shaw, A Pattern-Directed Approach to Flexible Manufacturing: A Framework for Intelligent Scheduling, Learning and Control, International Journal of Flexible Manufacturing System, 2, 121/144 (1989)
- [3] A.B.Chaturvedi, FMS Scheduling and Control: Learning to Achieve Multiple Goals, Expert Systems with Applications, 6-3, 267/286 (1993)
- [4] 人見 勝人, 新・生産管理工学, コロナ社, 1997
- [5] S. S. Panwalker and Wafik Iskander, “A Survey of Scheduling Rules”, OPERATIONS RESEARCH, Vol.25, No.1, Jan-Feb, 1997
- [6] John H. Blackstone, Jr, Don T. Phillips and Gary L.Hogg, “A state-of-the-art survey of dispatching rules for manufacturing job shop operations”, INT. J. PROD. RES., Vol.20, No.1, 27-45, 1982
- [7] Rakesh Agrawal and Ramakrishnan Srikant, “Fast Algorithms for Mining Association Rules”, 20th VLDB Conference Santiago, Chile, 1994
- [8] 福田 剛志, 森本 康彦, 森下 真一, 徳山 豪, 特別論説 情報処理最前線: データマイニングの最新動向 - 巨大データからの知識発見技術 - 情報処理, Vol. 37, No. 7, pages 597-603, July 1996, 情報処理学会
- [9] 藤本 英雄, 生産スケジューリングの動向, 計測と制御 第33巻 第7号 1994年7月

- [10] 黄 東明, 圓川 隆夫, 秋庭 雅夫 成功例に基づく平均納期遅れ基準のスケジューリング問題におけるディスパッチング・ルール生成に関する研究, 日本経営工学会誌 vol.41 No.6(1991)
- [11] 福井 哉, 橋本 文雄, 納期遅れ最小化のためのディスパッチングルールに関する研究, 日本経営工学会誌 vol.46 No.4(1995)
- [12] C.D. ペグデン, R.E. シャノン, R.P. サドウスキー, 高桑宗右エ門 対訳, 生産システムシミュレーション, コロナ社, 1993
- [13] R.Conway, W.Maxwell, and L.Miller, Theory of Scheduling, Addison-Wesley (1967).
- [14] 吉田武稔, 東崎秀行, 生産スケジューリングルール間の相関についての考察, システム / 情報合同シンポジウム '98, 51/56, 1998

研究実績

- [1] 吉田武稔, 東崎秀行, 生産スケジューリングルール間の相関についての考察, システム / 情報合同シンポジウム '98, 51/56, 1998
- [2] T.Yoshida and H.Touzaki, A study on association among dispatching rules in manufacturing scheduling problems, 7th IEEE International Conference on Emerging Technologies and Factory Automation, Barcelona, Spain, October 18-22, 1999.(submitted)