

# 食品工場におけるシフトスケジュール作成問題

2016SS074 角 大輝 2016SS092 山下依純

指導教員：鈴木敦夫

## 1 はじめに

本研究では、ある食品製造会社の工場における従業員の出勤ローテーション計画問題を取り扱う。研究内容は、スケジューリング手法を実際問題に適用させるための問題の定式化、システムの設計、実現法である。我々は、この出勤ローテーションを作成するシステムを実際に作成する。

スケジューリングシステムを用いると、従業員のシフトを短時間、かつ効率的に作成することができ、導入する企業も近年増加している [1]。このようなシステムを活用すれば、大変便利であるにも関わらず、いまだに多くの企業ではシステムの導入が行われておらず、スケジュール作成は手作業で行われている。その結果、スケジューリングを行う従業員が必要であり、かつ、必要以上の時間を費やしてしまっているのが現状である。本研究では、実際の現場で発生しているスケジューリング問題を調査し、その解決方法を考えてスケジューリングシステムを作成し、現場で活用する。

具体的には、「工場に勤める従業員の出勤ローテーションスケジューリング問題」について取り扱う。この問題を解決するために、過去の研究 [2] でシステムを開発したが、これらは実用性が低く、現場への導入ができなかった。我々が作成したシステムは、現場で試用でき、実際の現場に導入してスケジュール作成にかかる労力を軽減し、現場の効率を向上させる試みを行っている。

仙敷の研究 [3] によると、現在の工場の生産システムは、手作業のスケジューリングによる業務の効率の悪さと質の両面が問題として存在し、従業員にも大きな負担がかかっている。生産システムの最適化を自動化することで、これらの問題はすべて解決に到り、質の高いスケジュールによって工場全体の業務改善に繋がる。また、本研究では、企業の担当者や打ち合わせを重ね、現場での実用化を目的とした、使いやすく分かりやすいユーザインタフェースにしている。そして、現場で実際に使われる際に不具合はないか担当者や何度も協議し、システムの設計を行った。

## 2 食品工場における要員計画問題の現状

### 2.1 問題の説明

ある食品製造会社の工場では、複数の部署が存在し、部署ごとに従業員の出勤予定である出勤ローテーションを作成している。出勤ローテーションは各部署の担当者が、従業員ごとに存在する以下の情報をもとに作成する。

従業員の労働時間を表す「契約勤務時間情報」、従業員が働ける時間帯を表す「勤務可能時間帯情報」、休日を表す「公休日情報」と「有給情報」、従業員が担当できる勤務を表す「従業員勤務可能情報」、そして社員やアルバイト、派

遣、外国人実習生など従業員ごとの属性を考慮する。その上で各時刻に勤務する従業員とその勤務内容を割り振り、24時間分の出勤ローテーションを1週間分作成する。

この一連の作業を最適化ソフトウェアを用いて計算して自動化し、実際の現場での実用化を目指す。

### 2.2 スケジュールの作成手順

出勤ローテーションの作成段階として、派遣従業員以外の従業員の担当する勤務を決定する第1段階のシステムと派遣従業員の担当する勤務を決定する第2段階のシステムに分けて取り扱う。以降、システムの第1段階を Phase1、システムの第2段階を Phase2 と記述する。

最終的には Phase1 と Phase2 を連動させ、一つのシステムとして構築する。具体的な作成手順については次の章で記述するが、最初に部署ごとに配属されている従業員を各勤務に割り振り、そこで人数が不足した場合に、派遣部署から派遣従業員を割り振り、人数の不足分を補うことで出勤ローテーションを作成する。勤務スケジュールの作成手順をフローチャート図として図1に表す。

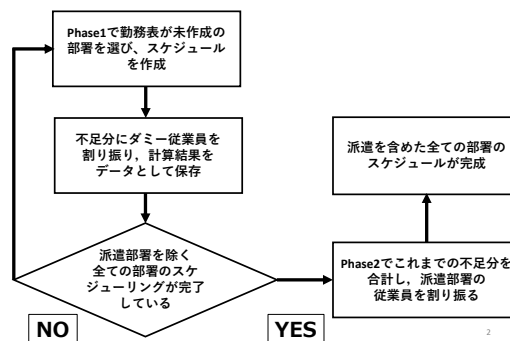


図1 勤務スケジュールの作成手順

### 2.3 その他の問題点

工場内には、出勤ローテーションを作成する上で必要になる部署内の従業員がどの勤務を担当できるかという情報がデータとして管理されていない部署も存在した。そのため、出勤ローテーション作成作業が属人化している問題が生じていた。

また、出勤ローテーションは各部署の担当者が独自に作成するため、部署ごとにレイアウトが統一されていなかった。そのため、スケジュール作成担当者を変更する場合、負担がかかってしまうという問題が生じていた。

### 3 問題解決の方法

#### 3.1 問題へのアプローチ

問題を解決するために、工場の全ての部署で扱えるスケジューリング支援システムを 0-1 整数計画法の問題として定式化することにより作成する。全ての部署で利用できることで、出勤ローテーションのレイアウトが統一され、スケジュール作成担当者が変更された場合でも負担がかかりにくいようにする。

具体的なスケジューリングシステムの手順として、まず Phase1 のシステムで派遣従業員が含まれることを想定してスケジューリングを行う。その際、不足分をダミー従業員として配置させる。このスケジューリングを派遣部署以外の部署が終わるまで繰り返し、派遣部署以外のスケジューリングを全て終了したら Phase1 は終了である。

次に、Phase2 では Phase1 で配置されたダミー従業員の勤務に対する不足分をまとめて時刻ごとの勤務に対する必要人数としてスケジューリングさせ、派遣従業員を割り振る。そうして Phase2 で求めた結果を Phase1 で求めた各部署に配置させて、各部署の最終的な出勤ローテーションとする。このように、システムの構築を Phase1 と Phase2 に分割していることで、計算する問題の規模が縮小され、計算速度と扱いやすさが向上している。

#### 3.2 Phase1 の出勤ローテーションスケジューリングの定式化

Phase1 では、ダミー従業員を交えて、合理的な出勤ローテーションを作成する。そのためには、各制約条件を満たした上で、勤務を行うダミー従業員の人数とダミー従業員が勤務を行う時間数、従業員が希望しない勤務を担当する時間数、従業員が契約勤務時間帯外に勤務を担当する時間数の重み付き和の最小化を図ることが望ましい。

各従業員には 1 日の契約勤務時間の下限値と、週の勤務時間の上下限値が与えられており、これを満たすようにスケジューリングする必要がある。また、勤務ごとに必要な作業人数が設けられており、その勤務内容によってはスキルを必要とするものが存在する。これらの条件をもとに定式化を行っていく。

##### 3.2.1 記号の定義

Phase1 における従業員の出勤ローテーションの定式化を示す。定式化にあたり、以下のように記号を定義する。

- $I$  : 従業員全体の集合  $i \in I, i = 1, 2, \dots, n$
- $I_1$  : ダミー従業員を除いた従業員の集合
- $I_2$  : ダミー従業員の集合,  $I_2 = I \setminus I_1$
- $J$  : 勤務の集合  $j \in J$
- $K$  : 時間の集合  $k \in K$
- $M$  : 従業員が 1 週間に勤務できる日数の集合  $m \in M$
- $K_{im}$  : 従業員  $i$  の  $m$  番目の勤務可能開始時刻の集合  $K_{im} \subset K$

$KW_{imk}$  : 従業員  $i$  の  $m$  番目の勤務可能開始時刻  $k$  と  $k$  以降の契約勤務時間数終了までの時間の和集合  $KW_{imk} \subset K$

##### 定数

- $L_i$  : 従業員  $i$  が 1 週間に勤務する時間数の下限値
- $U_i$  : 従業員  $i$  が 1 週間に勤務する時間数の上限値
- $w_i$  : 従業員  $i$  の契約勤務時間数
- $d_{jk}$  : 勤務  $j$  について、時刻  $k$  に必要な従業員の数
- $p_{ij}$  : 従業員  $i$  が勤務  $j$  を担当する際に発生する負担
- $q_{ik}$  : 従業員  $i$  が時刻  $k$  に勤務を担当した場合に発生する負担

$$a_{ik} = \begin{cases} 1: \text{従業員 } i \text{ が時刻 } k \text{ に勤務可能である} \\ 0: \text{その他の場合} \end{cases}$$

$$b_{ij} = \begin{cases} 1: \text{従業員 } i \text{ が勤務 } j \text{ を担当可能である} \\ 0: \text{その他の場合} \end{cases}$$

$$c_{ijk} = \begin{cases} 1: \text{従業員 } i \text{ が時刻 } k \text{ に勤務 } j \text{ を希望したときには} \\ \text{かならず勤務する} \\ 0: \text{その他の場合} \end{cases}$$

##### 決定変数

$$x_{ijk} = \begin{cases} 1: \text{従業員 } i \text{ が時刻 } k \text{ に勤務 } j \text{ を行う} \\ 0: \text{その他の場合} \end{cases}$$

$$y_{ik} = \begin{cases} 1: \text{従業員 } i \text{ が時刻 } k \text{ に勤務を開始する} \\ 0: \text{その他の場合} \end{cases}$$

#### 3.2.2 定式化

以上の記号を用いて、問題の定式化を行う。

##### 目的関数

$$\min. \quad \alpha \sum_{i \in I_2, j \in J, k \in K} ix_{ijk} + \beta \sum_{i \in I_1, j \in J, k \in K} p_{ij}x_{ijk} + \gamma \sum_{i \in I_1, j \in J, k \in K} q_{ik}x_{ijk} \quad (1)$$

##### 制約条件

$$L_i \leq \sum_{j \in J, k \in K} x_{ijk} \leq U_i, \quad i \in I_1 \quad (2)$$

$$\sum_{i \in I} x_{ijk} \geq d_{jk}, \quad j \in J, k \in K \quad (3)$$

$$x_{ijk} \leq a_{ik}, \quad i \in I_1, j \in J, k \in K \quad (4)$$

$$x_{ijk} \leq b_{ij}, \quad i \in I_1, j \in J, k \in K \quad (5)$$

$$y_{ik} \leq a_{ik}, \quad i \in I_1, k \in K_{im} \quad (6)$$

$$x_{ijk} \geq c_{ijk}, \quad i \in I_1, j \in J, k \in K \quad (7)$$

$$\sum_{k \in K_{im}} y_{ik} = 1, \quad i \in I_1 \quad (8)$$

$$\sum_{j \in J, k' \in KW_{imk}} x_{ijk'} \geq w_i y_{ik}, \quad i \in I_1, k \in K_{im} \quad (9)$$

$$\sum_{j \in J} x_{ijk} \leq 1, \quad i \in I, k \in K \quad (10)$$

$$x_{ijk} \in \{0, 1\}, \quad i \in I, j \in J, k \in K \quad (11)$$

$$y_{ik} \in \{0, 1\}, \quad i \in I, k \in K_{im} \quad (12)$$

### 式の説明

(1) 勤務を行うダミー従業員の人数と勤務を行う時間数、従業員が希望しない勤務を担当する時間数、従業員が契約勤務時間外に勤務を担当する時間数の重み付き和の最小化

(2) 各従業員について、週の出勤時間数を上下限値で指定

(3) 各勤務について、各時刻に必要な人数を割り当てる

(4) 従業員は勤務可能な時間以外には働かない

(5) 従業員は担当可能な勤務のみ行う

(6) 勤務を開始する時間は勤務可能時間内

(7) 希望した時間に希望した勤務をかならず行う

(8) 勤務を開始する時間は1出勤に1回のみ

(9) 従業員は契約時間分連続で勤務する

(10) 従業員が時間ごとに担当できる勤務はたかだか1つ

(11),(12) バイナリ制約

## 3.3 システムの実装

### 3.3.1 システムの概要

本システムは、VBA を用いて Microsoft Excel 上に実装し、最適化計算には Python を使用した。次で述べる手順に従って、従業員のシフトを作成していく。

### 3.3.2 システムを利用したシフト作成手順

#### シフト作成手順

1. 「スケジュール作成月」：シフトを作成する週の初め(月曜日)にあたる年、月、日を選択する。
2. 「作業情報」：シフトを作成する工場、勤務帯(昼夜)、部署を選択する。選択した部署の従業員情報(公休日や契約勤務時間など)についてもここに表示される。
3. 「勤務情報」：各時刻における必要な作業人数を勤務ごとに入力する。また、作業者が業務を担当可能であるか、担当可能である場合の業務を振り分ける優先度についてもここで設定する。
4. 「臨時予定入力」：公休日として設定されている日以外で臨時に業務が行えない従業員がいる場合、ここで設定する。
5. 「仮定期予定」：これまでの情報をもとに、Microsoft Excel の VBA を用いた計算によって従業員ごとに勤務時間のみ割り振られたスケジュール表が作成され、出力される。
6. 「勤務表作成」：「仮定期予定」で作成された表に勤務内容が割り振られたシフト表が出力される。

### 3.3.3 実行結果

出勤ローテーションシステム Phase1 の実行結果の一面を図2に示す。システムの使用方法として、シフト作成手

The screenshot shows a software interface for creating shift schedules. At the top, there are several buttons: 'シフト作成' (Shift Creation), 'シフト印刷' (Shift Printing), 'シフト入力' (Shift Input), 'シフト確認' (Shift Confirmation), and 'シフト終了' (Shift End). Below these buttons is a large grid. The grid has columns representing different tasks or departments and rows representing individual employees. The cells in the grid contain text indicating the assigned shift and task. The interface is designed for data entry and review of shift assignments.

図2 出勤ローテーションシステム Phase1

順の各工程に対応している画面上部のボタンを順に押し歩いていき、スケジューリングに必要な各種情報を入力していくと図2のような勤務表が得られる。

## 4 システム化の実際的な工夫

本研究では、企業の担当者と慎重に打ち合わせを重ね、現場での実用化を目的としている。そのため、使用するソフトウェアやシステムのユーザインターフェースについて様々な要望、指摘からシステムの利便性を向上させるように修正と変更を行った。

### 4.1 ダミー従業員数の設定

システムのプロトタイプでは、勤務人数が不足した場合に振り分けられるダミー従業員数を20人の固定値で考えていた。しかし、担当者との話し合いの中で、振り分けられるダミー従業員、すなわち、派遣社員数はシステムの利用者が任意に決定できる方が良いという要望を受け、ダミー従業員数の設定場所を追加した。

これにより、シフトスケジュール作成の自由度が高まり、人数が少ない部署にも対応できるようになった。

### 4.2 計算速度とソフトウェア

本研究では、Gurobi という最適化ソフトウェアを用いたが、その導入には相応のコストが必要となる。企業での実用化を目的とする中で、無料で利用可能な mypulp という最適化ソフトウェアを用いて計算を行った場合と計算時間を比較し、企業での運用の決定材料として役立てた。

### 4.3 ダミー従業員数の最小化と優先パラメータ

本研究では、ダミー従業員について、担当者との話し合いの中で、勤務を割り当てるダミー従業員の人数は無駄なく最低限であった方が良いという要望を受け、勤務を行うダミー従業員の人数とその時間数の最小化を目的関数の一つとしている。必要な人数分のダミー従業員にだけ勤務が割り振られることで、図3のようにダミー従業員の勤務も通常の従業員と同様に考えることができるようになり、派遣社員を割り振りやすくなった。ダミー従業員の勤務は、勤務情報シートで設定する各勤務の優先度をもとに、優先度の高い勤務から割り振るようにしている。

また、この変更に伴い、優先度を決定していた目的関数の3つの重み  $\alpha, \beta, \gamma$  を調整した。これらは現場担当者の裁量で変更可能である。

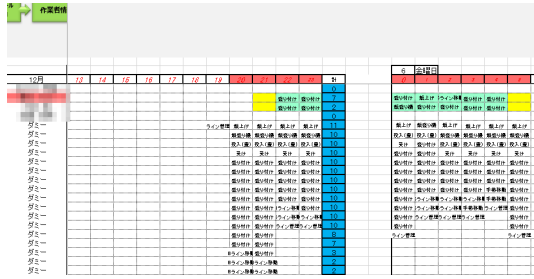


図 3 ダミー従業員の勤務

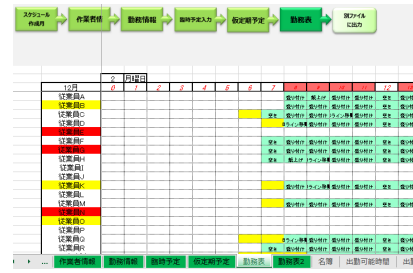


図 4

### 優先パラメータの調整

- 最優先：従業員が契約勤務時間外に勤務を担当する時間数の最小化 ( $\gamma$ )
- 優先：ダミー従業員の勤務する時間数の最小化 ( $\alpha$ )
- 重みなし：従業員が希望しない勤務を担当する時間数の最小化 ( $\beta$ )

### 4.4 勤務情報の変更

勤務表作成前にあらかじめ特定の従業員に対して勤務内容を指定したい場合、仮定期予定画面で各時間帯ごとに操作を行うことができる。この時、勤務日であるが所用により勤務を行わない時間帯の設定と公休日により勤務を行わない時間帯の区別がインターフェース、プログラムともに成されていなかった。そこで、勤務日であるが勤務を行わない時間帯に対して「勤務不可」という名称の勤務を勤務情報のシートに設置し、仮定期予定画面で選択できるようにした。これにより、公休日により勤務を行わない時間帯との区別ができるようになった。

### 4.5 夜勤務のスケジューリング

本工場には昼勤務と夜勤務が存在する。夜勤務のスケジューリングは日をまたぐケースが存在するため、適応させる公休日やシステムのインターフェースについて、担当者との確認を慎重に積み重ねた。その中で、日をまたいで勤務を行う夜勤務従業員と早朝から勤務を行う夜勤務従業員との区別ができるように仕様を変更した。

具体的には、従業員の勤務契約時間、勤務可能時間の登録時に 24 時以降の時間を設定できるようにし、例えば、日をまたいで勤務を終える際には 25 時、早朝から勤務を始める際には 1 時として対応できるようにシステムを改善した。この時、24 時以降はすべて同一の日付、すなわち、曜日はまだ変わっていないものとみなして扱うため、その際に参照する公休日には注意が必要である。

### 4.6 契約条件の自動確認

勤務表作成時に、勤務時間が週の勤務時間の上下限值内という制約条件から外れていないかを自動で確認し、どの従業員がどのような契約条件から外れているのかを視覚的に理解できるようにインターフェースを変更した。

図 4 に示すように、週の出勤時間が契約の上限を超える

場合は名前のセルを黄色に、下限を下回る場合には赤色に自動で塗るようにシステムを作成した。

## 5 おわりに

現在、ある食品製造会社の工場の出勤ローテーションシステムにおける要員計画スケジューリングシステムの作成に取り組んでいる。従業員ごとの契約勤務時間や勤務ごとの必要人数といった、現場における全ての問題を定式化し、最適化ソフトウェア Gurobi を用いて解を求め Microsoft Excel 上にシステムを構築した。

システムについては、担当者と協議を重ねることで、ユーザーインターフェースがシステム利用者にとって扱いやすいレイアウトになるよう設計されている。また、実際の現場では、数年の勤務形態の変更が行われることも考えられ、勤務名や必要人数、契約時間といったシステムの条件の変更が容易に行えるようにしている。

システムを使用することで、手作業で作成するよりも短時間で条件を満たすスケジュールを作成することが期待できる。そうした場合、スケジュール作成業務の負担を減らすだけでなく、時間の短縮や人件費削減に繋がる。また、効率的かつ合理的なスケジュールによって工場働く従業員の負担も軽減することができる。

今後も、担当者と慎重に協議しながら、システムで作成したスケジュールに不都合がないか、そして、追加すべき要素があるか確認していく。さらに Phase2 として Phase1 で勤務人数が不足した場合に割り振られるダミー従業員に対して派遣従業員を効率的、かつ、合理的に割り振るシステムの設計を目指していく。

## 参考文献

- [1] 伊藤邦春, 小林敬和, 塩谷政典, 森純一: 鉄鋼生産プロセスにおける生産計画, スケジューリング技術適用事例. オペレーションズ・リサーチ: 経営の科学, Vol.64 No.12 pp.4-11 (2019)
- [2] 松崎佳人: 『スケジューリングシステム開発についての実際研究』. 南山大学大学院理工学研究科システム数理専攻, 2017 年度修士論文.
- [3] 仙敷瞭汰: 『食品工場内の工程管理システムの開発』. 南山大学大学院理工学研究科システム数理専攻, 2018 年度修士論文.