

# 食品工場内の工程管理システムの開発

M2017SS006 仙敷瞭汰

指導教員：鈴木敦夫

## 1 はじめに

本研究では、スケジューリングシステムを実際問題に適用するための問題の定式化、システムの設計、実現法についての研究を行う。事例として、コンビニエンスストア向けに弁当や総菜を製造・納品しているある食品会社の生産工程計画問題を取り扱う。

近年、一部の企業では、最先端の情報システムを活用し、業務の効率化やコスト削減が図られている [4, 5]。これまでも、オペレーションズ・リサーチの手法を用いた様々な問題の解決によって、業務の効率は高くなっている。しかし、まだ多くの企業ではそれらの手法が浸透していない。そのため、生産スケジューリングや従業員の出勤スケジューリングなど、未だに手作業で行われているスケジューリング問題も数多く残っている。これらのスケジューリングは手作業で作成しているため、手間や時間がかかっていることや、スケジューリングが条件を満たしていないといった問題があり、業務の効率化と質の両面で問題がある。そこで、本研究では、実際の現場で発生している工程計画に関する問題の解決に取り組み、成果をシステムとして実現する。

具体的には、ある食品工場の生産工程で生じているスケジューリングの問題について取り扱う。この問題を解決するために、最適化の手法を用いた問題の定式化とシステムの開発を行う。

本研究では、開発するシステムの現場での実用化を目的とした。そのためにシステムのユーザーインターフェースについて、各システムの使用者と打ち合わせを重ねた。作成したシステムが実際に使われる際に不具合はないか何度も協議し、システムの設計をおこなった。例えば、システムのインターフェースについては、現在、現場で使用されている生産工程のタイムテーブルに近いものにした。そうすることで、システムの利用者が新しいシステムに移行する際に抵抗を感じないようにした。このように、システムを現場で実用化することを目標としてスケジューリングシステムの開発を行う。

## 2 食品工場における生産計画問題

### 2.1 生産工程のタイムテーブルについて

本研究ではある食品製造会社との共同研究である。この会社は、コンビニエンスストアなどに弁当、総菜、調理パン、デザートを企画、販売をしている企業である。中部地方を中心に工場を6つ展開しており、工場は24時間365日年中無休で稼働している。本研究ではこの会社の工場の一つである富士山工場の生産工程を表す、タイムテーブルについて取り扱う。

この食品工場には複数の部署が存在し、部署の中でさらに担当する作業によって複数の係が存在している。そして、係ごとに担当作業のタイムテーブルを作成している。

タイムテーブルは、工場で生産する弁当や総菜の量を生産計画をもとに作成される。生産計画は1週間前に、1週間分まとめて決定される。そのため、タイムテーブルは毎週1週間分作成されている。

タイムテーブルは各係の担当者が、以下の条件をもとに作成する。

- その作業に必要な作業人数
- その作業の前後の作業の兼合い
- 機器の制約

といった様々な条件を考慮した上で、各時刻に割り当て可能な作業を割り振り、タイムテーブルを作成している。

### 2.2 タイムテーブルの作成方法の現状と問題点について

現在のタイムテーブルの作成手順は、タイムテーブル作成担当者が、工場の生産計画をもとに各作業の作業人数や作業時間を計算し、作業の前後関係を考慮した上で割り当て可能な作業を割り振り、タイムテーブルを作成している。

タイムテーブルを作成する上で必要になる各作業にかかる時間や作業の前後関係という情報がタイムテーブル作成担当者の頭の中にしか存在せず、データとして管理されていない。そのため、タイムテーブル作成作業が属人化している問題が生じていた。また、タイムテーブル作成担当者を変更する場合、新しい担当者は過去のタイムテーブルを参考にして、各作業の情報を読み取る必要があるため、負担がかかってしまうという問題も生じていた。

### 2.3 問題解決の方法について

先で述べた問題を解決するために、タイムテーブルのスケジューリングを0-1整数計画問題として定式化する。そして、短時間で効率よくスケジューリングを行う自動化システムを数理最適化ソフトウェアを用いて実現し、実際の現場に導入することを最終目標とする。

具体的には、工場全体の生産計画をスケジューリングするシステムを作成する。タイムテーブル作成担当者を変更した場合でも負担がかかりにくいようにする。

また、作成するスケジューリングシステムについて、一度に全ての作業工程のスケジューリングを行うのではなく、係別にスケジューリングを行い段階的に分けてスケジューリングを行うシステムを作成する。一度に工場内の全ての作業をスケジューリングする場合、問題が大規模になってしまいスケジューリングに多大の時間が掛かってしまう。そのため、スケジュールを何度も修正し最終的なス

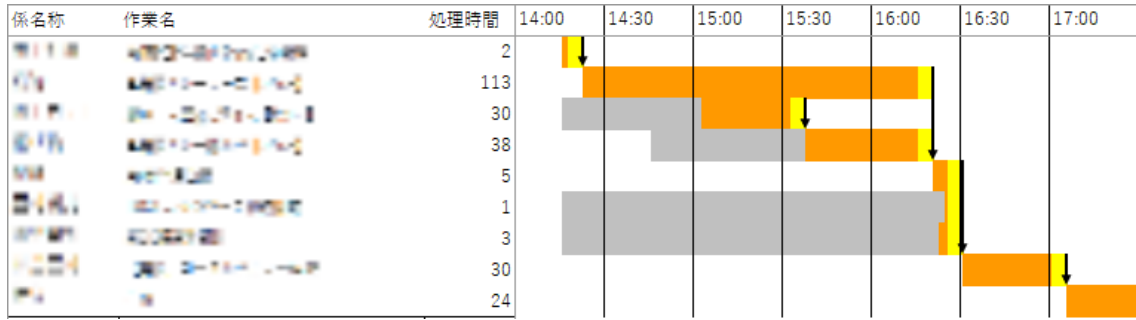


図1 生産工程のガントチャート

スケジュールを確定していくという方法がとりづらくなり、システムが実用的でなくなる。このことから、問題を分割し、段階的に解いていくシステムを作成する。係毎にスケジューリングを行う。

### 3 食品工場におけるタイムテーブルの自動作成システム

#### 3.1 生産工程の可視化

まず、タイムテーブルの作成方法を説明する前に生産工程の可視化について示す。生産工程の全体を把握するために、各製品の作業手順と出荷予定時刻から、PERT/CPMの手法を用いて、作業の順序関係を明らかにし、同時にシステム上に表示して、工程全体を可視化した。

PERT/CPMでは通常はPERT図を用いるが、工場の現場ではガントチャートが使い慣れているということだったので、ガントチャートを活用した。図1がその一例である。

PERT/CPMでは各部署、各係の作業について、最早開始時刻と最遅開始時刻がわかる。また、クリティカルパス上の作業も明確になる。

図2, 3は現在、現場で使用されている生産工程のタイムテーブルの一例である。

フライヤー	揚げ物					
	指示数	時刻表	人数	開始時間	終了時間	所要時間
揚げ物	7	17:10	1	17:10	17:17	7分
揚げ物	6	17:53	1	17:53	18:59	7分
揚げ物	51	18:20	2	18:20	19:11	6分
揚げ物	17	19:17	2	19:17	19:34	6分
揚げ物	28	19:40	3	19:40	20:08	3分
揚げ物	23	20:11	2	20:11	20:34	6分
揚げ物	50	20:40	2	20:40	21:30	6分
揚げ物	50	20:40	2	20:40	21:30	6分

図3 揚げ物係のタイムテーブル

#### 3.2 盛付係の生産計画問題の定式化

作業の割り当てを行うため、0-1 整数計画問題とし定式化する。問題を小規模にするため、3段階に分けて次のように定式化を行う。

step1 各盛付ラインの総作業時間の最大値の最小化

step2 使用する盛付ライン数の最小化

step3 盛付ライン全体の作業時間の最小化

以上の順序に問題を解くことで盛付係のタイムテーブル定式化を行う。

##### 3.2.1 記号の定義

step1を定式化するにあたり、添字集合と変数、そして定数を以下のように記号で定める。

添字集合

$L$  : 盛付ラインの集合  $l \in L$

$S$  : 作業の集合  $s \in S$

定数

$M_s$  : 作業  $s$  の作業時間

$$O_{sl} = \begin{cases} 1: \text{作業 } s \text{ が盛付ライン } l \text{ で可能} \\ 0: \text{上記以外} \end{cases}$$

決定変数

$$x_{sl} = \begin{cases} 1: \text{作業 } s \text{ を盛付ライン } l \text{ で行う} \\ 0: \text{上記以外} \end{cases}$$

炊飯計画 チェック表					新ライン		オペレーター		
3便	品番	品名	入力	調整	品名	浸漬終了時間	炊飯終了時間	冷却終了時間	トッピング
1	0	1	1	0	米	2 : 11	3 : 1	3 : 21	
2	3or4	41	49.83	3	米	2 : 47	3 : 37	3 : 57	0.00
3	2	71	66	3	米	3 : 34	4 : 24	4 : 44	0.00
4	3or4	51	18.22	3	米	3 : 48	4 : 38	4 : 58	4.30
5	3or4	18	14.61	6	米	4 : 2	4 : 52	5 : 12	5.00
6	3or4	178	8.83	3	米	4 : 10	5 : 0	5 : 20	0.00
7	3or4	75	0.92	6	米	4 : 15	5 : 5	5 : 25	7.40
8	3or4	7	0.92	3	米	4 : 18	5 : 8	5 : 28	7.25
9	3or4	21	4.83	8	米	4 : 27	5 : 17	5 : 37	6.35
10	3or4	11	3	8	米	4 : 34	5 : 24	5 : 44	7.00
11	3or4	15	2.61	8	米	4 : 41	5 : 31	5 : 51	8.30
12	3or4	78	8	8	米	4 : 52	5 : 42	6 : 2	8.00
13	3or4	91	4.92	6	米	4 : 59	5 : 49	6 : 9	8.45
14	2	97	1.22	6	米	5 : 4	5 : 54	6 : 14	9.05
15	3or4	94	7.22	6	米	5 : 13	6 : 3	6 : 23	6.00

図2 炊飯係のタイムテーブル

### 3.2.2 定式化

#### 目的関数

$$\text{Min. } Z = \max_{l \in L} \sum_{s \in S} M_s x_{sl} \quad (1)$$

#### 制約式

$$\sum_{l \in L} x_{sl} = 1 \quad (s \in S) \quad (2)$$

$$x_{sl} \leq O_{sl} \quad (s \in S, l \in L) \quad (3)$$

$$x_{sl} \in \{0, 1\} \quad (s \in S, l \in L) \quad (4)$$

#### 式の説明

- (1) 各盛付ラインの総作業時間の最大値の最小化
- (2) 各作業をいずれか1つの盛付ラインで行う
- (3) 各作業が使用可能な盛付ラインである
- (4) バイナリ制約

### 3.2.3 記号の定義

step2 を定式化するにあたり, step1 の記号に加え, 変数と定数を以下のように記号で定める.

#### 定数

$Z$ : 総作業時間の上限

#### 決定変数

$$y_l = \begin{cases} 1: \text{盛付ライン } l \text{ を使用する} \\ 0: \text{上記以外} \end{cases}$$

### 3.2.4 定式化

#### 目的関数

$$\text{Min. } \sum_{l \in L} y_l \quad (5)$$

#### 制約式

$$\sum_{l \in L} x_{sl} = 1 \quad (s \in S) \quad (6)$$

$$\sum_{s \in S} M_s x_{sl} \leq Z \quad (l \in L) \quad (6)$$

$$x_{sl} \leq O_{sl} \quad (s \in S, l \in L) \quad (7)$$

$$x_{sl} \leq y_l \quad (s \in S, l \in L) \quad (7)$$

$$x_{sl} \in \{0, 1\} \quad (s \in S, l \in L) \quad (8)$$

$$y_l \in \{0, 1\} \quad (l \in L) \quad (8)$$

#### 式の説明

- (5) 使用する盛付ライン数の最小化
- (6) 各盛付ラインの総作業時間の上限
- (7) 各作業が使用可能な盛付ラインである
- (8) バイナリ制約

### 3.2.5 記号の定義

step2 より, 作業  $s$  がどの盛付ライン  $l$  に振り分けるのかを決定する. step3 を定式化するにあたり, step2 の記号に加え, 添字集合と変数, そして定数を以下のように記号で

定める.

#### 添字集合

$S_l$ : 盛付ライン  $l$  の作業の集合  $s \in S_l$

$T$ : 時刻の集合  $t \in T$

$T = \{1, 2, \dots, n\}$  ※  $n$  は十分大きい自然数

#### 定数

$N_s$ : 作業  $s$  の作業員数

$P_t$ : 時刻  $t$  の勤務人数

#### 決定変数

$$z_{st} = \begin{cases} 1: \text{作業 } s \text{ を時刻 } t \text{ で行う} \\ 0: \text{上記以外} \end{cases}$$

### 3.2.6 定式化

すべての  $l \in L$  について以下の問題を解く.

#### 目的関数

$$\text{Min. } \max_{s \in S} \sum_{t \in T} t z_{st} \quad (9)$$

#### 制約式

$$\sum_{s \in S} N_s z_{st} \leq P_t \quad (t \in T) \quad (10)$$

$$\sum_{t \in T} z_{st} \geq M_s \quad (s \in S) \quad (11)$$

$$\sum_{s \in S_l} z_{st} \leq 1 \quad (t \in T) \quad (12)$$

$$\sum_{t' \in T \setminus \{1, 2, \dots, t + M_s - 1\}} z_{st'} \leq M_s (1 - z_{st}) \quad (s \in S, t \in T) \quad (13)$$

$$\sum_{s \in S_l, t' \in T \setminus \{1, 2, \dots, t + A - 1\}} z_{st'} \leq A (1 - z_{st}) \quad (t \in T, A = \sum_{s \in S_l} M_s) \quad (14)$$

$$z_{st} \in \{0, 1\} \quad (s \in S, t \in T) \quad (15)$$

#### 式の説明

- (9) 最後の作業終了時刻の最小化
- (10) 各時刻で作業員数が勤務人数以下である
- (11) 各作業を作業時間以上行う
- (12) 各盛付ラインで同時に複数の作業は出来ない
- (13) 各作業を連続で行う
- (14) 各盛付ラインを連続で行う
- (15) バイナリ制約

以上の定式化により, 盛付係のタイムテーブルを作成する.

他の加熱係や, 揚げ物係等の定式化についても, 同様の考え方で現在考察中である.

### 3.2.7 問題規模と計算結果

この会社の過去の生産実績をもとに, 試作したシステムを用いて盛付係のタイムテーブルを作成した. 以下に, スケジューリング問題の規模と計算結果を示す.

計算に使用した計算環境は以下の通りである。

OS Microsoft Windows 10

CPU Intel core i7-7700

RAM 16.0GB

計算に使用したデータは以下の通りである。

商品数 41 商品

食数 40,948 食

製造ライン 14 ライン

以上のデータをもとに計算した結果は以下の通りである。

step1	step2	step3
変数 575	変数 588	変数 16,400
制約式 629	制約式 643	制約式 44,441
時間 0.235 秒	時間 0.179 秒	時間 計測不能

step3 については、計算時間が1日以上かかったので断念した。step2 までのタイムテーブルは図4の通りである。step3 については、貪欲算法で解いた結果を図5に示す。貪欲算法での計算時間は0.359秒であった。

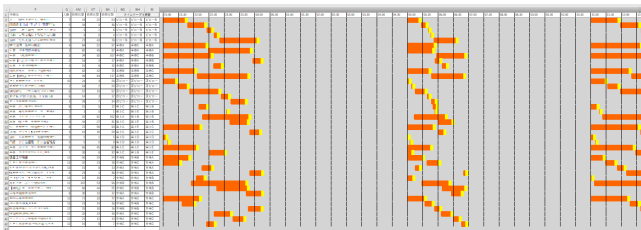


図4 step2 まで解いた盛付係のガントチャート

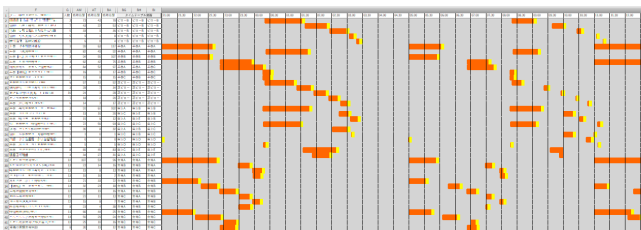


図5 貪欲算法で解いた盛付係のガントチャート

## 4 おわりに

現在、ある食品会社の工程管理システムの開発に取り組んだ。この問題について定式化を行い、最適化ソフトウェア mypulp を用いて解を求め、Microsoft Office Excel 上にシステムを試作した。一部の問題については、貪欲算法で解を求めた。

作成するシステムについて、担当者と協議を重ねることでシステム利用者にとって扱いやすいものになるように設計した。また、実際の現場では無理のあるスケジューリングを出すこともある。そのような場合を考え、タイムテーブルの修正を容易行えるようにした。

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、空いた時間を別の業務に費やすことで企業の利益につなげることができる。また、質の高いスケジュールによって食品工場で従業員の負担を軽減することができる。

今後は、システムを利用者に試用してもらい、手作業によるスケジュールと大きく異なる点はないか、システムで作成したスケジュールに不都合はないか確認していく。そして、そこから得られた知見をもとに、スケジュール作成者が意図した結果が得られるような定式化とシステムの改良を行っていく予定である。

最終的には、「工程管理システム」で生産工程全体をスケジューリングし、その生産工程に必要な作業員の割り当てを行うシステム「出勤ローテーション支援システム」[1]と連携して機能するシステムを構築していく予定である。

## 参考文献

- [1] 松崎佳人, 『スケジューリングシステム開発についての実験的研究』, 南山大学理工学研究科システム数理専攻, 2017年度修士論文, 2018.
- [2] 宮代隆平, 『整数計画ソルバー入門』, オペレーションズ・リサーチ, Vol.57, pp.187-189, (2012).
- [3] 杉原弘章, 『マルチ PERT』, 静岡学術出版, 静岡, 2008.
- [4] 株式会社 構造計画研究所, 『シフトスケジューリングソリューション KKE/ShiftMaster』, <http://solution.kke.co.jp/bpr/solutions/shift-scheduling.html>, 2019年2月13日確認
- [5] 株式会社 NTT データ 数理システム, 『東日本旅客鉄道株式会社様 JINJRE (勤務)』, [http://www.msi.co.jp/nuopt/solution/shift/case\\_jr.html](http://www.msi.co.jp/nuopt/solution/shift/case_jr.html), 2019年2月13日確認