

# 食品工場における生産スケジュールの最適化

2014SS008 江川奈那 2014SS055 中西加奈

指導教員：佐々木美裕

## 1 はじめに

本研究では、コンビニエンスストアの弁当等の生産を行う工場の生産スケジュールリングの最適化を考える。対象とする工場では受注した弁当やおにぎりの生産、各コンビニエンスストアへの配送を行っている。対象とする工場では現在、廃棄率の高さが問題になっている。そこで、廃棄率を下げ、さらに利益を上げるために、生産スケジュールの最適化を行う。これにより、効率の良い生産スケジュールや、なるべく人手のかからない作業手順を自動で作成することが可能になり、労務管理の簡易化(残業時間の削減・組織の柔軟性向上)と生産性の向上が期待できる。

## 2 問題の説明

### 2.1 現状

対象とする工場は25の生産ラインを有し、1日に1便、2便、3便の3回に分けて生産と配送を行っている。各便の生産量は、見込みデータと呼ばれる受注量予測データをもとに仮決定し、生産が完了する前に送られてくる確定データに基づいて確定する。しかし、1便の確定データは、生産が完了したあとにしか送られないため、見込みデータとの差分は生産完了後に再度人員を割いて生産しなければならない場合がある。また、対象とする工場には、食材の加熱処理を行う加熱部門、様々な種類のごはんを炊く炊飯部門、食品の冷却から加熱後の加工など、盛り付けの直前までの作業を行う仕込み部門、調理された商品を盛り付ける盛り付け部門の大きく分けて4つの部門がある。部門ごとに2つの資料をもとに生産スケジュールを表したタイムテーブルを手作業で作成している。そのため、膨大な時間を費やさなければならないこと、そして工場全体として、最適な生産スケジュールを確実に作成できているかわからないことが大きな問題として考えられる。

### 2.2 解決策

本研究では、はじめに、各商品の注文量と出荷時刻の情報をもとに商品ごとのタイムテーブルを作成し、その結果をもとに、各部門のタイムテーブルへと展開する方法を用いる。この手法を用いることにより、各部門の生産スケジュールを考慮した工場全体の生産スケジュールの作成を目指す。

## 3 PERT/CPMを用いた解法

### 3.1 概要

PERT/CPM(program evaluation and review technique/critical path method)を用いて最早時刻と最遅時刻を求め、さらに、プロジェクトを遅らせないために重要

な作業のつながりとなるクリティカルパスを明らかにして所要時間の短縮を図る。

### 3.2 PERT図の作成

#### 3.2.1 ダミー作業

ダミー作業とは、作業時間0の作業であり、実際に作業を行うわけではないが、アローダイアグラムを用いて作業順序を正しく表すために必要となるものである。図1は、任意の2つの作業*i*と作業*j*に関連したダミー作業が必要かどうかを判定するアルゴリズムの概要を示している。

以下では、図1に示した各パターンについて、具体的なダミー作業の追加手順について説明する。

#### パターン1(図2)

作業*i*の終点から作業*j*の終点へダミー作業を追加する。

#### パターン2(図3)

共通の先行作業の終点から作業*i*, *j*それぞれの始点へダミー作業を追加する。

#### パターン3(図4)

作業*i*の始点から作業*j*の始点へダミー作業を追加する。

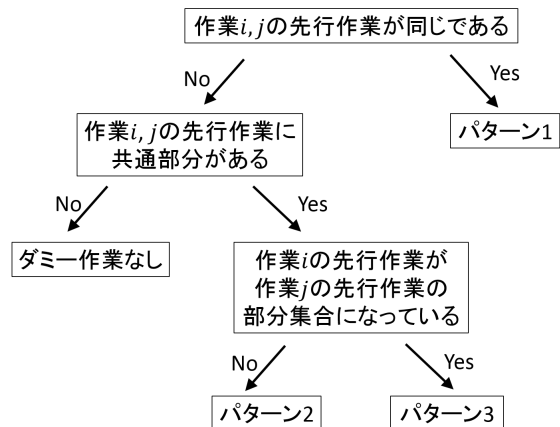


図1 ダミー作業の判定

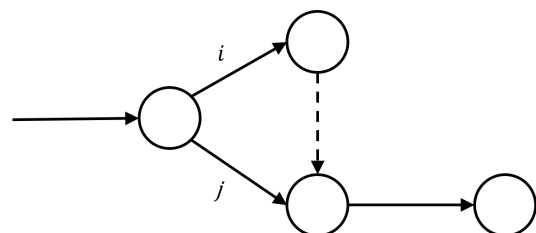


図2 ダミー作業の追加(パターン1)

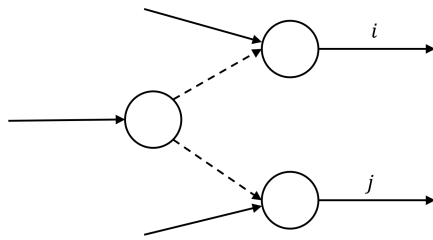


図3 ダミー作業の追加 (パターン2)

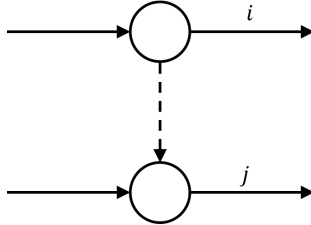


図4 ダミー作業の追加 (パターン3)

ここでは2つの作業のみに着目しているが、ダミー作業が追加されなくなるまでこの手順を繰り返すことによって、追加すべきダミー作業を全て追加することができる。

### 3.3 トポロジカル順

アローダイアグラムを作成するためには、各作業の始点と終点に適切な番号付けを行う必要がある。ここでは、番号付けのルールとしてトポロジカル順を用いる。その順序は、先行作業がないプロジェクト開始点を1番とし、その後は、番号付けされた点から出ている矢印を除いた状態で先行作業のない結合点に番号を振るといった繰り返しで得られる。途中で先行作業のない結合点が複数出てくるときは、どれを先に番号付けてもかまわない [3]。

本研究では各作業の始点と終点にそれぞれ番号付けを行うプログラムを作成した。アルゴリズムの説明のために、以下の記号を定義する。

入力

$n$ : 作業数

$D_i$ : 作業  $i$  の作業名

$P_i$ : 作業  $i$  の先行作業の集合

出力

$S_i$ : 作業  $i$  の始点

$T_i$ : 作業  $i$  の終点

アルゴリズムは以下の通りである。

ステップ0  $i := 1, j := 1, S_i := 0$ .

- (1)  $P_i = \phi$  ならば  $S_i := j$ .
- (2)  $i := i + 1, i \leq n$  ならばステップ0へ。そうでなければステップ1へ。

ステップ1  $i := 1, k := 1$ .

- (1)  $S_i = 1$  かつ  $P_k \supseteq D_i$  ならば  $P_k := P_k \setminus D_i$ .
- (2)  $k := k + 1, k \leq n$  ならば (1) へ。

- (3)  $i := i + 1, k = 1, i \leq n$  ならば (1) へ。そうでなければステップ2へ。

ステップ2  $i := 1, j := j + 1, k := 1$ .

- (1)  $S_i \neq 0$  または  $P_i \neq \phi$  ならば (6) へ。
- (2)  $S_i := j$ .
- (3)  $P_k \supseteq D_i$  ならば  $P_k := P_k \setminus D_i$ .
- (4)  $k := k + 1, k \leq n$  ならば (3) へ。
- (5)  $i := 1, j := j + 1, k := 1$  として (2) へ。
- (6)  $i := i + 1, i \leq n$  ならば (2) へ。そうでなければステップ3へ。

ステップ3  $i := 1, k := 1, T_i = 0$ .

- (1)  $P_k \supseteq D_i$  ならば  $T_i = S_k$ .
- (2)  $k := k + 1, k \leq n$  ならば (1) へ。
- (3)  $i := i + 1, k := 1, i \leq n$  ならば (1) へ。そうでなければステップ4へ。

ステップ4  $i := 1$ .

- (1)  $T_i = 0$  ならば  $T_i := j + 1$ .
- (2)  $i := i + 1, i \leq n$  ならば (1) へ。そうでなければ終了する。

### 3.4 作成したプログラムについて

#### 3.4.1 インターフェイス

PERT/CPM のプログラムは、Python を用いて実装した。Python には数多くのモジュールが存在し、これを用いることによってプログラムを簡潔に書けることが利点である。本研究では xlwings モジュールを用いて Microsoft Excel と Python の連携を行い、Excel 上から Python のプログラムを実行し、結果を Excel に出力する。xlwings モジュールを用いることにより、Python と Excel 双方から Excel の操作が可能になる。これにより、今まで VBA を使ってプログラムしていた Excel 上のデータ処理を容易に Python プログラムに書き換えることができる [2]。

#### 3.4.2 出力までの流れ

PERT/CPM のプログラムでは作業に関するデータを手動で入力し、最早時刻・最遅時刻とクリティカルパスを出力する。

はじめに、「入力データ」シートに作業番号、作業名、作業時間、先行作業数、先行作業を所定の列に入力する (図5)。

入力後、「実行」ボタンを押すと「最早時刻・最遅時刻」シートに各作業の最早時刻・最遅時刻と全ての作業完了時刻が出力される。「クリティカルパス」シートには、この商品におけるクリティカルパスが出力される。実行して得られたクリティカルパス上の作業に、コストや人員を割くことで最も効率よく全作業完了時刻を早めることができる。

作業番号	作業	作業時間	先行作業数	先行作業
0	仕分け	25	1	1
1	奥のこみすびとおかずセット	40	13	2, 14, 26, 28, 32, 34
2	奥のこみすび40部成型【作/作分】	160	2	3, 13
3	冷凍仕上げ調整	30	1	4
4	凍りつけ調整	10	3	5, 6
5	計量	5	0	
6	凍りかけ汁【作/作分】	1	0	
7	凍り着【作/作分】	15	1	8
8	凍り着【作/作分】	15	3	9, 10
9	凍りつけ水溜り【作/作分】	3	0	
10	計量	3	0	
11	凍りかけ汁【作/作分】	30	1	12
12	計量	3	0	
13	No.29 横切り解凍1000【作/作分】	0	0	
14	凍り着凍り【作/作分/作分】	15	1	15
15	凍り着凍り【作/作分/作分】	30	2	16
16	凍り着凍り【作/作分/作分】	5	2	17
17	冷凍凍り【作/作分】	0	0	
18	計量	5	0	
19	計量	5	0	
20	凍り着凍り【作/作分/作分】	25	1	21

図5 入力画面

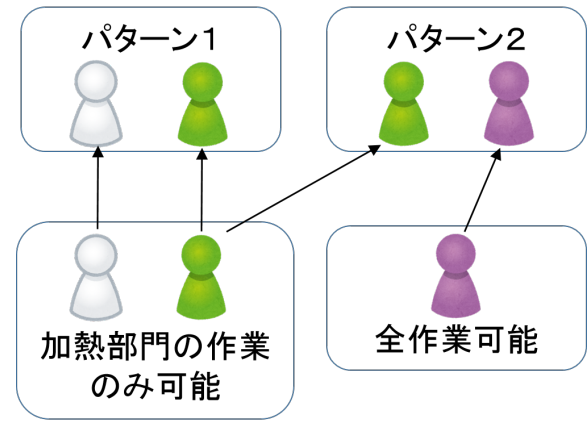


図6 作業割当のパターン

## 4 RCPSP としてのモデル化

### 4.1 概要

RCPSP を用いて資源制約を考慮したうえで、最適な作業順序を求めるプログラムを作成する。また、出力結果と MicroStrategy を用いてガントチャートを作成する。

### 4.2 RCPSP

資源制約付きプロジェクトスケジューリング問題のことであり、英語表記 (resource constrained project scheduling problem) の頭文字をとり、RCPSP と呼ぶ。これは、資源に関する条件がある中で、各作業の開始時刻および終了時刻を決定する問題である。なお、スケジュールの良さを比較するための指標として、作業完了時刻や納期からの遅れなどが挙げられる [5]。プロジェクトスケジューリングにおいて、複数の作業に共通して使用される設備や人員を資源と呼ぶ。本研究では工場が持つ機器や従業員が資源となる。

### 4.3 OptSeq

OptSeq は、メタヒューリスティックを基とし作成されたスケジューリング最適化ソルバーであり、Python のモジュールとして提供されている。様々なタイプの資源制約の記述が可能であり、柔軟性に富んだ最適化ソルバーであり、実務における様々な条件が追加されたスケジューリング問題に対して、短時間で良好な解を探索することができる [1]。

### 4.4 MicroStrategy

MicroStrategy とは、組織で管理するさまざまなデータを組み合わせ、必要なデータを引き出し、レポート、ドキュメント、ダッシュボードで可視化する BI プラットフォームである。MicroStrategy を用いることで、ガントチャートを簡単に出力することができる [4]。

### 4.5 作成したプログラムについて

#### 4.5.1 作業者の能力を考慮したスケジュール表

作業者を能力にあった作業に割り当て、自分の担当する作業を把握できるように、スケジュール表を作成する。こ

の作業者の能力を考慮したスケジュール表を作成するために、OptSeq のモードを用いて資源を定義する。

モードとは作業の処理方法のことを表す。各作業は少なくとも 1 つのモードを持ち、そのうちのいずれかを選択し処理する。このモードの機能を用いることによって、資源制約を定義し、各作業に設定することで資源を作業に制約として追加することができる。

また、モードは細かい資源制約の設定を行うことも可能とする。RCPSP において、一般的に人を資源として定義すると、人という 1 つのグループとして定義され、作業者全員が同じ人として扱われる。しかし、実際には人はそれぞれ異なった能力を持っており、1 つの人というグループで定義してしまうと、各作業者を区別して作業に割り当てることができない。これを解決するために、人を能力ごとのグループに分類して、各作業者が異なった能力を持つことを表現し、その分類を基にスケジューリングする。このように、作業者を能力別に分類すると作業に作業者を割り当てる際に、いくつかのパターンが生じる。例えば、加熱の作業を行うために 2 人作業者が必要なとき、加熱部門の作業だけを行える作業者を 2 人を割り当てる場合と、加熱部門の作業だけが行える作業者を 1 人と、全作業を行える作業者を 1 人を割り当てる場合がある (図 6)。このようにいくつかの割り当てるパターンをもつ場合、モードを用いて各作業に全ての割り当てパターンを設定していくことにより、能力に合った作業者の割り当てを行うことができる。

#### 4.5.2 出力までの流れ

RCPSP のプログラムでは手動で作業に関するデータを入力し、作業開始時刻を出力する。

はじめに「入力データ」シートに作業番号、作業名、部門、作業時間、作業人数、資源の数、使用機械、先行作業数、先行作業を各列に入力する (図 7)。

「資源一覧」シートに、工場が持つ機器と数、各作業者の作業可能な部門には 1、そうでない部門には 0 を入力する (図 8)。

入力後、「実行」ボタンを押すと「結果 (全体)」シートに

は各作業の開始時刻・完了時刻、「結果(加熱)」シートには加熱部門における各作業の開始時刻・完了時刻、「結果(炊飯)」シートには炊飯部門における各作業の開始時刻・完了時刻が出力される(図9)。

実行結果と MicroStrategy を用いることによって、ガントチャートを出力することができる(図11)。

「能力別スケジュール表」ボタンを押すと「能力別スケジュール表」シートに作業者を能力ごとに分け、能力ごとにどれだけの人が使われているかが出力される(図10)。

実際に工場で生産されている「きのこおむすび」と「幕の内弁当」の加熱・炊飯部門を取り出したものを例として実行した結果を図9~11に示す。



図10 能力別スケジュール表出力結果

実行		能力別スケジュール表						
作業番号	作業名	部門	作業時間	作業人数	資源の数	使用機械	先行作業数	先行作業
1	冷)まいたけ御飯	炊飯	30	1	1	炊飯真空冷却	1	2
2	まいたけ御飯	炊飯	10	3	1	炊飯新ライン	3	3
3	計量	炊飯	5	1	0		0	
4	舞茸煮汁(キノコセト)	炊飯	1	1	0		0	
5	舞茸煮(キノコセト)	加熱	15	1	1	釜150	3	6
6	*まいたけ水煮MO-1	加熱	3	1	0		0	
7	計量	加熱	3	1	0		0	
8	鶏かつだし汁(キノコセト)	加熱	30	1	1	釜150	1	9
9	計量	加熱	3	1	0		0	
11	鶏もも唐揚げ(イカセト)	加熱	30	2	1	フライヤー3	2	12
12	鶏肉漬込(イカセト)	加熱	5	1	0		1	14
14	計量	加熱	5	1	0		0	
15	計量	加熱	5	1	0		1	22
16	▲ホーケーゼンホイール	加熱	10	1	1	釜100	1	17
17	計量	加熱	5	1	0		0	
18	冷)弁当用白飯	炊飯	10	2	1	炊飯真空冷却	1	19
19	弁当用白飯	炊飯	5	1	1	炊飯新ライン	1	45
20	幕の内ひじき煮(430)	加熱	1	1	1	釜200	6	21
21	白絞油BL10(バラロ)	加熱	5	1	0		0	

図7 入力画面

資源の名前	数	名前	加熱	炊飯
人	5	A	0	1
炊飯真空冷却	1	B	1	0
炊飯新ライン	1	C	1	1
釜150	1	D	1	0
フライヤー3	1	E	0	1
釜100	1			
釜200	1			
鍋1	1			
ジェットオープン1上段	1			
パッチ式フライヤー	1			

図8 資源一覧

作業番号(全体)	作業名(全体)	作業開始時刻	作業完了時刻
0	切れ目クワンナーR13g計量(1pc=76ヶ)	15	17
1	▲切目ウインナー油調【共通】	19	34
2	粉体調味料をまとめて行う	5	20
3	液体調味料をまとめてはかる	0	5
4	粉体調味料をまとめて行う	33	38
5	液体調味料をまとめてはかる	0	5
6	冷凍鶏もも18-23g【次便】	23	28
7	鶏肉漬込(イカセト/キノコ/430)	63	73
8	鶏もも唐揚げ【イカセト/キノコ/430】	106	111
9	粉体調味料をまとめて行う	8	23
10	液体調味料をまとめてはかる	21	31
11	さば(定塩骨取)4カット【イカセト/430】	10	20
12	さば漬込【イカセト/430幕】	31	71
13	▲さばの醤油焼き【イカセト/430】	101	106
14	▲ポテトコロッシュ油調【共通】	20	21
15	粉体調味料をまとめてする	0	15
16	液体調味料をまとめてする	17	19
17	▲ハンバーグたれ【430幕】	123	125
18	液体調味料をまとめてする	6	8
19	▲ごま和え調味料【430幕】	121	123
20	粉体調味料をまとめてする	43	58

図9 作業開始時刻・完了時刻出力結果

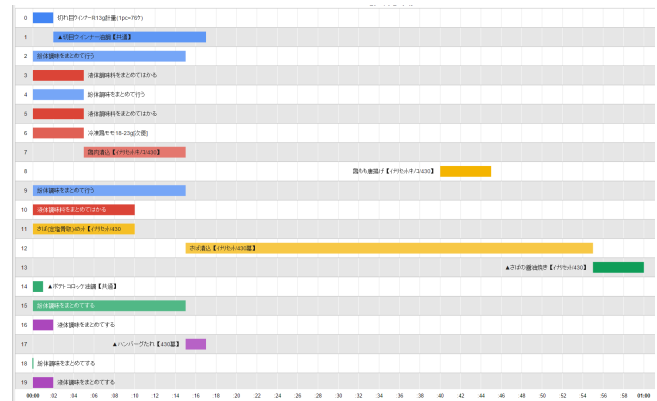


図11 MicroStrategy を用いたガントチャート出力結果

## 5 おわりに

本研究では、ある製品における最早時刻・最遅時刻、クリティカルパスを求めるプログラム、資源制約を追加したうえで生産スケジュールを作成するプログラムの2つを作成した。クリティカルパスを算出することにより、どの作業にコストをさけば作業全体にかかる時間を短縮できるかが分かる。生産スケジュールを作成するプログラムの実用にはまだ改善が必要である。細かい機器の制約を追加することにより、より現実的なプログラムとなる。今後は人員の割当を自動で可能にすることが目標である。

## 参考文献

- [1] 有江禎晶・久保幹雄・橋本英樹・野々部宏司:『実務的な資源制約付きスケジューリング問題に対する OptSeq の適用』。スケジューリング・シンポジウム 2017 講演論文集, pp. 94-99, 2017.
- [2] 久保幹雄・小林和博・斎藤努・並木誠・橋本英樹:『Python 言語によるビジネスアナリティクス 実務家のための最適化・統計解析・機械学習』。近代科学社, 東京, 2016.
- [3] 松井泰子・根本俊男・宇野毅明:『入門オペレーションズ・リサーチ』。東海大学出版会, 神奈川, 2008.
- [4] マイクロストラテジー・ジャパンホームページ。 <https://www.microstrategy.com/jp>
- [5] 野々部宏司・茨木俊秀:『資源制約付きスケジューリング問題の定式化と近似解法』。数理解析研究所講究録, 1120 巻, pp. 88-97, 1999.