

# 非正社員を主力とするスタッフスケジューリングにおける モデル化と支援システムの構築

徳永 拓真<sup>1,†1,a)</sup> 田中 勇真<sup>1,b)</sup> 小林 隆文<sup>2,c)</sup> 沓水 佑樹<sup>2,d)</sup> 池上 敦子<sup>1,e)</sup>

受付日 2015年2月5日, 再受付日 2015年3月9日,  
採録日 2015年4月25日

**概要:** 飲食店や販売店等の非正規雇用のスタッフが主力である現場ではスタッフの勤務可能な時間帯が限られているため, 個人を意識したスケジューリングが必要である. また, 現場によって営業時間や働くシフトの長さ, 業務の数や勤務ルールが異なるため, どの現場にも適用できるモデルの構築が難しい. これに対し, 本研究では現場ごとに異なる状況や個々のスタッフの都合を1日単位のスケジュールで表現することを考える. これらの単日スケジュールを組み合わせることで, 現場の違いやスタッフの都合を考慮できるモデルを提案し, それに基づくスケジューリング支援システムを構築する. また, 過去の勤務表からその傾向を観察し, その傾向をスケジューリングに取り込む方法を検討する.

**キーワード:** スタッフスケジューリング, モデル化, スケジューリング支援システム, シフトスケジューリング, 意思決定

## Development of a Staff Scheduling Model for Part-time Employees and a Support System

TAKUMA TOKUNAGA<sup>1,†1,a)</sup> YUMA TANAKA<sup>1,b)</sup> TAKAFUMI KOBAYASHI<sup>2,c)</sup> YUKI KUTSUMIZU<sup>2,d)</sup>  
ATSUKO IKEGAMI<sup>1,e)</sup>

Received: February 5, 2015, Revised: March 9, 2015,  
Accepted: April 25, 2015

**Abstract:** Scheduling staff in the service industry often takes considerable time and resources. Developing all-around software is also hard due to various constraints which depend on the type of the service and the management policy. The constraints of this scheduling problem involve such factors as the skill level of a team, balancing workload among staff and consideration of staff's preferences. In this paper, we propose a mathematical programming model for this problem, which can deal with differences among the type of the service and the management policy as well as consideration of staff's preferences. We also show a prototype of the scheduling support system which is developed based on the model. We then discuss a way to represent constraints or goals which are considered unconsciously, in our model.

**Keywords:** staff scheduling, modeling, scheduling support system, shift scheduling, decision science

### 1. はじめに

本研究では, ある期間を対象に, 各日の業務にスタッフ

を割り当てて, 勤務表を作成するスタッフスケジューリング問題を扱う. 特にシフトの管理方法が「スタッフ申請型」とよばれる非正規雇用のスタッフが主力となる現場を対象とする. このような現場では, 各スタッフの勤務可能時間帯が限られており, 各スタッフの能力を考慮しながら各時間帯の各業務に必要な人数を揃えるよう勤務表を作成しなければならない [1]. 図 1 に, ある飲食店の勤務表の一部を示す. 1 行目に時間帯, 2 行目以降の各行に, その日に勤務する各スタッフの勤務時間帯をチャートで示している (各時間帯に行う業務まで決める場合には, 業務の記号が書き込まれる).

<sup>1</sup> 成蹊大学  
Seikei University, Musashino, Tokyo 180-8633, Japan

<sup>2</sup> リクルートジョブズ  
Recruit Jobs Co., Ltd., Chuo, Tokyo 104-8227, Japan

<sup>†1</sup> 現在, 富士通システムズ・イースト  
Presently with Fujitsu Systems East Ltd.

a) toku@cleo.ci.seikei.ac.jp

b) ytanaka@st.seikei.ac.jp

c) takafumi.kobayashi@r.recruit.co.jp

d) kutsumizu@r.recruit.co.jp

e) atsuko@st.seikei.ac.jp

スタッフスケジューリングについては1950年代頃から非常に多くの研究がなされている [2], [3], [4]. 文献で紹介されているモデルでは、各日の各時間帯の必要人数を揃えるスタッフ数最小の勤務表を人を特定しないで作り、あとからその要素である1人分のスケジュールをスタッフに割り当てている。したがって、個々のスタッフの都合や希望まで考えたモデルとなっていない [5].

また、勤務表作成を支援するための市販システムも存在する (2章で示すアンケート調査結果では7.6%の現場で利用・試用経験があった) が、実績管理には使われていてもスケジューリング機能は使われていないことが多い。使われない理由 (問題点) としては、

1. 現場の勤務ルールを考慮できないといった、利用されているモデルの問題
  2. 与えた条件に対し、良解を与えないことや求解時間が遅いといった、利用されているアルゴリズムの問題
  3. 作成者が思うような制約条件をすべて入力しようとすると入力に時間がかかってしまうという問題
  4. 作成者が潜在的に考慮している条件が存在し、制約条件として明確に扱えていないという問題
- 等があげられ、望むような勤務表が得られない場合がある

図 1 ある飲食店の勤務表の例

Fig. 1 Example of a shift schedule for a restaurant.

からだと考えられる。

一方、近年、数理最適化技術の進歩は著しく、数理最適化問題を解くための汎用ソルバは、かつて現実的な時間で解くことが不可能だと思われていた問題に対しても高速に解を与えることが可能になってきた [6].

本研究では、既存研究のモデルで個々のスタッフの都合や希望を考慮できない点や、市販システムにおける問題点 1 に着目し、スタッフごとの勤務可能性や希望を考慮するとともに、販売店や飲食店等、1日の勤務のルールやタイムスロットの幅 (スケジューリングにおける最小考慮時間単位) 等が異なる多くの現場に対応できる汎用的なモデルを提案する。そして、そのモデルを基に、数理最適化汎用ソルバを利用した意思決定支援システムの構築を行うことで問題点 2 を解決する。さらに、過去の勤務表における特徴や傾向を観察し、それらをスケジューリングに取り込むことを試み、問題点 3 や問題点 4 に対する解決方法を検討する。

## 2. 現場調査

複数の現場 (店舗) の勤務表作成における共通部分と違い、作成時に考慮していることを (陽に考慮していることだけでも) 把握するために、現場における聞き取り調査を続けるとともに、勤務表作成者を対象に 2013 年 3 月にアンケート調査を行った (回答数: 515). 表 1 は作成者が勤務表作成時にどのようなことをどの程度考慮しているのかを表したものである。

表の 1 列目に示した考慮項目は、複数の現場に対する予備調査で得た内容であり、数値は各考慮項目に対し、「絶対に考慮」「できれば考慮」「気にしない」と回答した割合 (%) を示している。「絶対に考慮」の列の数値につけた下線は、30%以上の回答があったことを示している。この結果では、ベテランの配置、スタッフの働く時間量や時間帯が重視されていることが分かる。

聞き取り調査の結果とアンケート調査の結果をあわせると、各時間帯において業務ごとにスキルレベルを考慮しながらスタッフ数を揃えることはどの店舗でも共通なものであった。つまり、店舗による勤務表の違いは、その要素と

表 1 アンケートの調査結果 (回答パーセンテージ)

Table 1 Questionnaire results (%).

考慮項目	絶対に考慮	できれば考慮	気にしない
ベテランを必ず配置	<u>31.1</u>	47.0	21.9
スタッフの能力	28.5	56.5	15.0
スタッフ間の人間関係	10.9	50.7	38.4
人件費	28.2	53.2	18.6
勤務時間の長短	23.9	55.5	20.6
連続勤務時間	<u>44.5</u>	43.3	12.2
公平さ	<u>33.2</u>	51.7	15.1
希望時間・曜日	<u>42.9</u>	50.7	6.4

なるスタッフの1日における働き方の違い（1日分のスケジュールの違い）にあると考えられる。たとえば、各日各スタッフの勤務する時間帯だけを決定する場合もあれば、行う業務まで決定する場合もある。また、1つの業務をどのくらいの長さで行うことが許されているのか、1日に行える業務の数、そして、スケジュールの最小時間単位が1時間であるか、30分であるか、15分であるか等、現場ごとに様々である。本研究では、1日分のスケジュールの違いを以下のように整理した。

1. 始業時刻と終業時刻
2. タイムスロット（最小考慮時間単位）
3. 1日の勤務時間の下限と上限
4. 業務種類数と内容
5. 各業務の始業時刻と終業時刻
6. 各業務の1日の勤務時間の下限と上限
7. 各業務で考慮すべきスキルレベルの数
8. 1業務を（他業務をはさんで）分割して行えるか否か
9. 1業務を分割して行える場合、1回の勤務時間の下限
10. 1日に行える業務数の上限
11. 1日に一緒に行えない業務の組合せ
12. 連続して行えない業務の組合せ

1 スタッフにとっての単日スケジュールを、現場のこれらの違いにあわせて作成できれば、それらを組み合わせることで、その現場にあった勤務表作成が可能になる。また、これらの違いに加えて、各スタッフの各日の都合にあわせたスケジュールに絞り込むことことで、個々のスタッフを考慮した勤務表作成ができる。

以上のことから、本研究では、店舗間の勤務表作成の違いを「スタッフの1日における働き方の違い」ととらえ、現場ごとの働き方にあわせた実行可能な単日スケジュールを利用することで、これまで扱いが困難とされてきた「店舗間の違い」をパラメータとして表現し、定式化には陽に表さないことを考える。

### 3. 提案モデル

本研究では、各スタッフの各日に対する実行可能な単日スケジュール（個別単日スケジュールとよぶ）を、前章にあげた項目1から12と対象スタッフの可能業務と勤務可能時間帯に従ってあらかじめ全列挙する。そして、各スタッフの各日においてたかだか1つの個別単日スケジュールを選択し、それらを組み合わせて人数を揃える定式化を考える。個別単日スケジュールは現場の営業形態や勤務ルールに合わせて様々なものを設定できるが、対象スタッフの勤務可能時間帯や可能業務にあわせて絞り込み、長さも1日分なので、その数が膨大になることはなく、列挙も比較的容易である\*1。図2、図3に、個別単日スケジュールの例を示す。

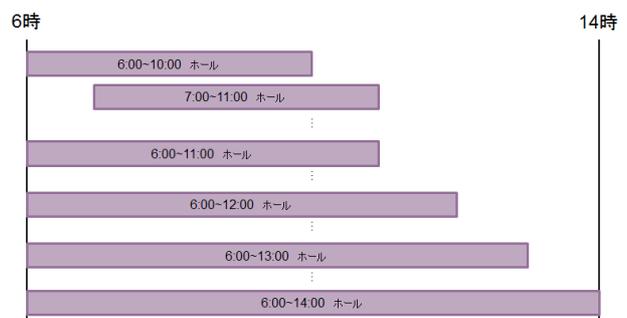


図2 6時から14時まで勤務可能なスタッフの1業務で構成される個別単日スケジュールの例

Fig. 2 Examples of one-day schedules (consisting of a single task) for staff who can work from 6:00 to 14:00.

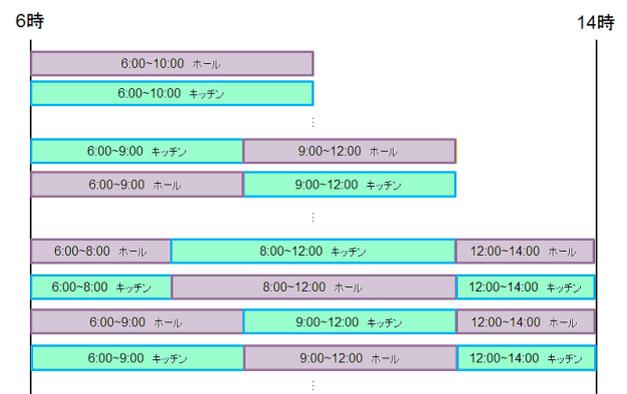


図3 6時から14時まで勤務可能なスタッフの2業務で構成される個別単日スケジュールの例

Fig. 3 Examples of one-day schedules (consisting of two tasks) for staff who can work from 6:00 to 14:00.

本研究では、意思決定変数として、スタッフ*i*の日*j*に個別単日スケジュール*p*を採用するとき1、そうでないとき0となる $x_{ijp}$ を利用する。

#### 定式化のための記号説明

$M$ : スタッフの集合

$D$ : 日にちの集合

$H_j, j \in D$ : 日*j*のタイムスロットの集合

$K$ : 業務の集合

$P_{ij}, i \in M, j \in D$ : スタッフ*i*の日*j*の個別単日スケジュールの集合（各個別単日スケジュールは $\rho_{ijphk}, i \in M, j \in D, p \in P_{ij}, h \in H_j, k \in K$ で表す。スタッフ*i*の日*j*の個別単日スケジュール*p*のタイムスロット*h*が業務*k*であるとき $\rho_{ijphk}$ は1、そうでなければ0）

$e_{ijp}, j \in D, i \in M, p \in P_{ij}$ : スタッフ*i*の日*j*の個別単日スケジュール*p*の勤務時間量

$R_k, k \in K$ : 業務*k*で考慮すべきスキルレベルの集合

\*1 1 スタッフの単日に対し、勤務開始時間と終了時間を変えながら、その時間帯に発生する業務をその業務勤務時間の上下限を守るように割り当て、対象現場で可能性のあるスケジュール（ポテンシャルスケジュール）をすべて列挙する。そして、その中から対象スタッフの可能な業務と各日の可能時間帯にあわせて選択する。

$\delta_{ikr}, i \in M, k \in K, r \in R_k$ : スタッフ  $i$  の業務  $k$  のスキルレベルが  $r$  であるか否かを示す (スキルレベルが  $r$  なら 1, そうでなければ 0)

$a_{jhkr}, b_{jhkr}, j \in D, h \in H_j, k \in K, r \in R_k$ : 日  $j$  のタイムスロット  $h$  の業務  $k$  におけるスキルレベル  $r$  のスタッフの勤務人数の下限と上限

$l_i, u_i, i \in M$ : スタッフ  $i$  の総勤務時間の下限と上限

$c_j, j \in D$ : 日  $j$  に勤務できるスタッフ数の上限

変数

$x_{ijp}, i \in M, j \in D, p \in P_{ij}$ : スタッフ  $i$  の日  $j$  に個別単日スケジュール  $p$  を採用するとき 1, そうでないとき 0 となる意思決定変数

$\alpha_{jhkr}^-, \alpha_{jhkr}^+, j \in D, h \in H_j, k \in K, r \in R_k$ : 日  $j$  のタイムスロット  $h$  の業務  $k$  におけるスキルレベル  $r$  のスタッフの勤務人数のそれぞれ不足分と過剰分を表す変数

$\beta_i^-, \beta_i^+, i \in M$ : スタッフ  $i$  の総勤務時間のそれぞれ不足分と過剰分を表す変数

以下に定式化を示す.

定式化

minimize

$$\sum_{j \in D} \sum_{h \in H_j} \sum_{k \in K} \sum_{r \in R_k} (w_{jhkr}^- \alpha_{jhkr}^- + w_{jhkr}^+ \alpha_{jhkr}^+) + \sum_{i \in M} (w_i^- \beta_i^- + w_i^+ \beta_i^+) \quad (1)$$

subject to

$$\sum_{p \in P_{ij}} x_{ijp} \leq 1 \quad i \in M, j \in D \quad (2)$$

$$a_{jhkr} - \alpha_{jhkr}^- \leq \sum_{i \in M} \sum_{p \in P_{ij}} \delta_{ikr} \rho_{ijphk} x_{ijp} \leq b_{jhkr} + \alpha_{jhkr}^+ \quad j \in D, h \in H_j, k \in K, r \in R_k \quad (3)$$

$$l_i - \beta_i^- \leq \sum_{j \in D} \sum_{p \in P_{ij}} e_{ijp} x_{ijp} \leq u_i + \beta_i^+ \quad i \in M \quad (4)$$

$$\sum_{i \in M} \sum_{p \in P_{ij}} x_{ijp} \leq c_j \quad j \in D \quad (5)$$

$$x_{ijp} = 0 \text{ or } 1 \quad i \in M, j \in D, p \in P_{ij} \quad (6)$$

$$\alpha_{jhkr}^-, \alpha_{jhkr}^+ \geq 0 \quad j \in D, h \in H_j, k \in K, r \in R_k \quad (7)$$

$$\beta_i^-, \beta_i^+ \geq 0 \quad i \in M \quad (8)$$

ここで, 目的関数の  $w_{jhkr}^-, w_{jhkr}^+, w_i^-, w_i^+$  は, それぞれ変数  $\alpha_{jhkr}^-, \alpha_{jhkr}^+, \beta_i^-, \beta_i^+$  が値を持つことに対するペナルティコスト, つまり, 勤務人数の過不足や勤務時間の過不足に対するペナルティコストである.

各式の意味は以下のとおりである.

- (1) 各日の各タイムスロットの各業務において各スキルレベルのスタッフ勤務人数の過不足の加重和と, 各スタッフの総勤務時間の過不足の加重和の総和を最小化する.
- (2) スタッフ  $i$  の日  $j$  に選択される個別単日スケジュールはただか 1 つとする.

(3) 日  $j$  のタイムスロット  $h$  の業務  $k$  におけるスキルレベル  $r$  のスタッフの勤務人数の上下限を守ろうとする.

(4) 各スタッフの総勤務時間の上下限を守ろうとする.

(5) 各日の勤務スタッフ合計人数の上限を守る.

(6), (7), (8) 各変数の値域を表す.

現場ごと, スタッフごとに異なる 1 日のスケジュールをすべて列挙して (集合  $P_{ij}$  を設定して) 利用するモデルを構築することで, 営業時間や業務の内容もまったく違う多くの現場に適用可能であると考えられる.

表 1 にあげた考慮項目との対応を考えると, 「連続勤務時間」「希望時間・曜日」については, 個別単日スケジュール作成の際に反映できる. 「ベテランを必ず配置」「スタッフの能力」は, スキルレベルの設定と勤務人数の上下限, つまり制約式 (3) で考慮できる (4 割近くが「気にしない」と回答している「人間関係」も, 必要であれば, 組合せを避けるスタッフに疑似的なスキルレベルを設定し, そのスキルレベルからの勤務人数の上限を 1 にするといった工夫ができる). 「勤務時間の長短」は, 制約式 (4) で考慮できる. また, 「公平さ」では働く量に関する公平さが主に求められることから, 「勤務時間の長短」として制約式 (4) で考慮できる.

#### 4. 計算実験

提案した定式化に基づいて, ある飲食チェーンの店舗を対象に期間 1 カ月の勤務表作成を行う計算実験を 2012 年 10 月から 2013 年 3 月までの 6 カ月分行った. スタッフ数は 16 人~18 人 (月によって変動), 始業時刻と終業時刻はそれぞれ 14 時と 24 時, 業務の種類は 3 種類 (常勤業務, ホール, キッチン), 各業務においてスキルレベルは 1 種類 (スキルレベル分けされていない), 1 スタッフが 1 日に行うことができる業務は 1 種類, 各スタッフの総勤務時間の下限と上限はそれぞれ 10 時間と 180 時間, 1 日に勤務可能な人数の上限は 9 人, タイムスロットの幅は 30 分である, 例として, 10 月の各スタッフの可能業務, 勤務可能時間帯と各日の個別単日スケジュールの数 (それぞれ 6 日分), 各日各タイムスロットの各業務に必要な人数 (1 日分) を付録に付ける.

数理最適化汎用ソルバとして, IBM CPLEX Ver.12.5.0.0 [7] を利用して求解した. 計算環境は Intel(R) Core(TM) i7-2600 CPU 3.40 GHz×2 である.

目的関数における  $w_{jhkr}^-, w_{jhkr}^+, w_i^-, w_i^+$  の値をすべて 1 に設定した場合の実験について報告する. 表 2 に 6 カ月分の勤務表作成の結果を示す.

表 2 では, 2 列目に各日各タイムスロットの各業務におけるスタッフ不足数  $\alpha_{jhkr}^-$  の総和, 3 列目にスタッフ過剰数  $\alpha_{jhkr}^+$  の総和, 4 列目に各スタッフの総勤務時間の不足  $\beta_i^-$  の総和, 5 列目に勤務時間の過剰  $\beta_i^+$  の総和, 6 列目に求解時間を示している.

いずれの月でも与えた目的関数に対する最適解を1秒弱で得ることができた。しかし、どの月も、勤務人数不足が起きていること、スタッフにとって勤務時間過剰が起きていることから、スタッフの可能時間帯に偏りがあることやスタッフ数自体が足りない可能性があることが分かる。

現場で作成された勤務表を観察したところ、どの月においても、勤務人数の過剰はなく、総勤務時間量の過剰も提案モデルの勤務表と同じ値になっていた。表3は、現場で作成された勤務表と提案モデルによって得られた勤務表を、勤務人数の不足の総和と比較するとともに、各日各スタッフの各タイムスロットの勤務内容が現場で作成された勤務表と一致した割合（一致率）を示す。

10月～2月の計算実験では現場の勤務表よりも勤務人数不足を減らすことに成功している。一般に、現場では、この人数不足に対してスタッフが希望申請していない時間帯に勤務依頼をすることで対応しているが、ここで示した結果は、その調整の負荷を減らすことができる。また、各スタッフの各日の各タイムスロットにおける業務は、現場の勤務表と90%以上一致していたため、提案モデルは、勤務表の質を上げながらも現場の勤務表作成を実現できているといえる。

対象店舗と同じ飲食チェーンの他の店舗でも同じ実験を6カ月分行ったが、同様な結果が得られた（6カ月中5カ月で、勤務人数不足を軽減できた。各月の現場の勤務表との一致率も90%以上であった）。これら以外にも、菓子販売店の勤務表作成では、勤務人数不足を減らすとともに、現場の勤務表作成担当者に「店舗にあった勤務表である」と評価された。

表2 各月の勤務人数過不足，総勤務時間過不足，求解時間  
Table 2 Staff surplus/shortage, surplus/shortage of total working hours, and time to find solution.

対象月	$\sum \alpha_{jhr}^-$	$\sum \alpha_{jhr}^+$	$\sum \beta_i^-$	$\sum \beta_i^+$	求解時間
10月	7	0	0	3	0.85秒
11月	7	0	0	20	0.53秒
12月	16	0	0	34	0.93秒
1月	258	0	0	30	0.94秒
2月	29	0	0	6	0.69秒
3月	6	0	0	0	0.72秒

表3 現場で作成された勤務表との不足人数の比較と一致率  
Table 3 Comparison between actual schedules and our generated schedules: staff shortages and rate of agreement.

対象月	現場の勤務表	提案モデルの勤務表	一致率
10月	12	7	94.1%
11月	8	7	93.6%
12月	22	16	93.4%
1月	277	258	92.5%
2月	43	29	93.6%
3月	6	6	93.8%

提案モデルが目指す勤務表を作成できたので、次のステップでは、多くの勤務表作成を行うことでモデルの汎用性を確認するため提案モデルに基づく勤務表作成支援システムを構築する。

### 5. 支援システムの構築

現場で提案モデルを評価するため、勤務表作成のための意思決定支援システムのプロトタイプを構築した。

支援システムでは、Microsoft Office Excelを使用する。勤務表作成に必要なデータを入力し、数理最適化汎用ソルバを利用して求解する。そして、得られた解を勤務表として表示する。支援システムの作業の流れを図とともに以下に示す（図4）。

- (1) 基本情報を入力する
  - 店舗基本情報  
始業時刻，終業時刻，タイムスロット幅，業務の種類とその時間帯（始業時刻，終業時刻）とスキルレベルの種類。1スタッフの1日の勤務時間（上下限），各業務の勤務時間（上下限），他業務を間にはさむこと（業務の分割）が可能か否か，可能な場合には1回の業務勤務時間の下限，1日に行える業務数の上限，1日に一緒に行えない業務，連続できない業務。
  - スタッフ基本情報  
スタッフの名前（ID），期間内の総勤務時間（上下限），担当可能業務とスキルレベル。
- (2) ポテンシャルスケジュールを生成する  
店舗情報に基づき，その店舗で可能性のある1人分の単日スケジュールをすべて列挙する。
- (3) 対象期間における業務詳細情報の入力
  - 必要スタッフ数  
各日各タイムスロット各業務における各スキルレベルのスタッフ必要数（上下限）。
- (4) 対象期間におけるスタッフ詳細情報の入力
  - 各スタッフの勤務可能日，可能時間帯
- (5) 個別単日スケジュールを生成する  
各スタッフの各日について勤務可能時間帯と可能業務を考慮して実行可能スケジュールを列挙する（ポテン

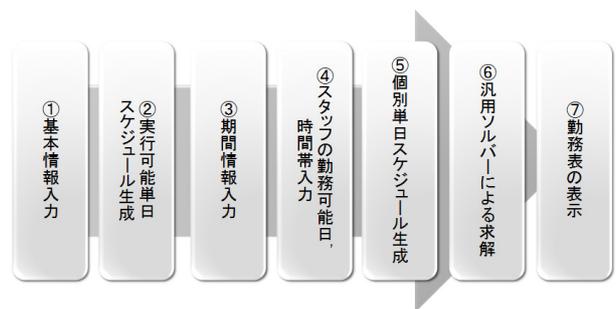


図4 支援システムにおける勤務表作成の流れ  
Fig. 4 Flow of scheduling when using the support system.

シャルスケジュールから条件にあうものを選択する).

(6) 数理最適化汎用ソルバを利用して求解する

提案モデル (定式化) に基づき汎用ソルバで求解する (汎用ソルバとしては, Numerical Optimizer [8] を利用する).

(7) 勤務表を表示する

得られた解 ( $x_{ijp}$  の値) を勤務表の形で表示する. 各日の勤務時間量を示した表と, 各日のスケジュールの詳細をチャートで示したものを表示する.

支援システム利用の際には, 定式化の目的関数における重みづけは, デフォルト値設定しておけば意識せずに勤務表作成できるが, 得られた勤務表に違和感があった場合には, 重みづけをチューニングすることもできる.

構築した支援システムは, 今後, 4章で対象にした店舗の勤務表作成に利用される予定である.

## 6. 過去の勤務表の観察と傾向の反映

構築した支援システムをより有効に活用するため, 作成者が潜在的に考慮している条件, 評価尺度をいかすことを考え, 過去の勤務表を観察した. たとえば, 「対象スタッフが夕方から夜にかけて勤務していることが多い」等の傾向があれば, それを反映することを目指す. 過去の傾向を勤務表作成の評価尺度に加えることで, 作成者が望むような勤務表作成の手助け (1章であげた問題点4の解決) になる可能性, また, 支援システムの問題点となっている「データ入力量の多さ」をデフォルト値等の自動設定により削減 (問題点3の解決) できる可能性がある. そこで, 4章の計算実験で対象とした店舗で作成された5カ月分の勤務表の傾向を観察し, 以下の3つに着目した.

1. 一緒に勤務する傾向があるスタッフのペア
2. 各スタッフに採用される傾向があるスケジュール
3. 各曜日各タイムスロットの各業務の平均スタッフ数

このような傾向を数値化ができれば, 3章の提案モデルに取り込むことが可能だと考えた. 以下に項目1, 2に対する数値化例を示し, それを取り込む修正モデルを示す. 項目3については, データ入力量削減についての今後の課題について述べる.

### 【1】一緒に勤務する傾向があるスタッフのペア

各業務で対象スタッフが一緒に勤務しているタイムスロット数を調べ, 期間内で発生するその業務の総タイムスロット数で割ることで, スタッフ同士が一緒に勤務している割合  $w_{kii'}$  ( $k \in K, i, i' \in M, i \neq i'$ ) を算出した. ここでは, 一緒に勤務している時間が長いほど組ませたいペアであると仮定して評価することにした. この割合が高いペアと一緒に勤務する時間をできる限り多くするような仕組みを作るため, 修正モデルでは, スタッフ  $i$  と  $i'$  ( $i \neq i'$ ) が日  $j$  のタイムスロット  $h$  の業務  $k$  において一緒に勤務する

とき1, そうでないとき0となるような変数  $y_{jhkii'}$  ( $j \in D, h \in H_j, k \in K, i, i' \in M, i \neq i'$ ) を導入した. 修正モデルでは, 元の定式化に次に示す式 (9) と式 (10) を加え, 目的関数を式 (11) に変更する.

追加する制約【1】

$$\sum_{r \in R_k} \sum_{p \in P_{ij}} \delta_{ikr} \rho_{ijphk} x_{ijp} \geq y_{jhkii'} \quad j \in D, h \in H_j, k \in K, i, i' \in M, i \neq i' \quad (9)$$

$$\sum_{r \in R_k} \sum_{p \in P_{i'j}} \delta_{i'kr} \rho_{i'jphk} x_{i'jp} \geq y_{jhkii'} \quad j \in D, h \in H_j, k \in K, i, i' \in M, i \neq i' \quad (10)$$

新しい目的関数【1】

$$\sum_{j \in D} \sum_{h \in H_j} \sum_{k \in K} \sum_{r \in R_k} (w_{jhkr}^- \alpha_{jhkr}^- + w_{jhkr}^+ \alpha_{jhkr}^+) + \sum_{i \in M} (w_i^- \beta_i^- + w_i^+ \beta_i^+) - \sum_{j \in D} \sum_{h \in H_j} \sum_{k \in K} \sum_{i, i' \in M, i \neq i'} w_{kii'}^y y_{jhkii'} \quad (11)$$

ここで  $w_{kii'}^y$  は, 業務  $k$  でスタッフ  $i$  とスタッフ  $i'$  ( $i \neq i'$ ) が一緒に勤務するときの評価値として利用されている. 式 (9) の左辺は, スタッフ  $i$  が日  $j$  のタイムスロット  $h$  で業務  $k$  を行うとき1になるが, そうでないとき0になるので  $y_{jhkii'}$  も0になる. 同様に, 式 (10) の左辺は, スタッフ  $i'$  が日  $j$  のタイムスロット  $h$  で業務  $k$  を行うとき1になるが, そうでないときは0になるので  $y_{jhkii'}$  も0になる. つまり, これら2つの式では, 日  $j$  のタイムスロット  $h$  の業務  $k$  でスタッフ  $i$  と  $i'$  が一緒に勤務するときのみ  $y_{jhkii'}$  が1になることを許す. そして, 目的関数式 (11) では, 元の目的関数の式 (1) を最小化するだけでなく,  $w_{kii'}^y$  の値が大きいスタッフ  $i$  と  $i'$  が一緒に勤務する時間の合計を最大化する\*2.

### 【2】各スタッフに採用される傾向があるスケジュール

各スタッフが各タイムスロットで各業務を行った割合  $q_{ihk}$  を算出した. そして, 多く採用されているタイムスロットが対象スタッフにとって好ましい時間帯であると仮定し, スタッフ  $i$  の日  $j$  の個別単日スケジュール  $p$  の好ましさの得点  $w_{ijp}^x$  ( $i \in M, j \in D, p \in P_{ij}$ ) として, 以下の2通りの値を考えた.

- (A) 各タイムスロットの業務  $k$  に対して, その業務を行った割合  $q_{ihk}$  の合計値 ( $\sum_{h \in H_j} \sum_{k \in K} \rho_{ijphk} q_{ihk}$ )
- (B) (A) の値を勤務スロット数で割った値

修正モデルでは, 目的関数を式 (12) に修正し, 元の目的関数の式 (1) を最小化するだけでなく, 各スタッフの各日

\*2 ここでは, 一緒に勤務している時間が長いほど組ませたいペアであると仮定して評価したが, 逆の要望があった場合には, 式 (11) の最後の項の前の  $-$  を  $+$  に変更して評価することも考えられる.

表 4 結果の比較 (過不足数, 現場の勤務表との一致度, 求解時間)  
**Table 4** Results (staff shortages, rate of agreement, and time to find solution).

モデル	$\sum \alpha_{jhkr}^-$	$\sum \alpha_{jhkr}^+$	$\sum \beta_i^-$	$\sum \beta_i^+$	一致率	求解時間
提案モデル	6	0	0	0	93.8 %	0.72 秒
<b>[1]</b>	6	0	0	0	95.0 %	539.89 秒
<b>[2]</b> (A)-1	6	34	0	0	95.3 %	0.75 秒
<b>[2]</b> (A)-10	6	0	0	0	95.4 %	0.75 秒
<b>[2]</b> (A)-100	6	0	0	0	95.5 %	0.75 秒
<b>[2]</b> (A)-1000	6	0	0	0	95.6 %	0.75 秒
<b>[2]</b> (B)-1	6	0	0	0	94.4 %	0.76 秒
<b>[2]</b> (B)-10	6	0	0	0	95.4 %	0.75 秒
<b>[2]</b> (B)-100	6	0	0	0	95.5 %	0.75 秒
<b>[2]</b> (B)-1000	6	0	0	0	95.6 %	0.75 秒

に対して評価値の高い個別単日スケジュールを多く選ぶように設定した。

新しい目的関数 **[2]**

$$\begin{aligned}
 &\text{minimize} \\
 &\sum_{j \in D} \sum_{h \in H_j} \sum_{k \in K} \sum_{r \in R_k} (w_{jhkr}^- \alpha_{jhkr}^- + w_{jhkr}^+ \alpha_{jhkr}^+) \\
 &\quad + \sum_{i \in M} (w_i^- \beta_i^- + w_i^+ \beta_i^+) \\
 &\quad - \sum_{i \in M} \sum_{j \in D} \sum_{p \in P_{ij}} w_{ijp}^x x_{ijp} \quad (12)
 \end{aligned}$$

4章で扱った店舗を対象に, 観察した5カ月分の結果(評価尺度)を他の月の勤務表に反映することを目指し, **[1]**, **[2]**の修正モデルに基づく勤務表を作成した。そして, 4章で得られた勤務表と比較した。

**[1]**では, 新しく設定した  $w_{kii'}^y$  の値に対し, 元の目的関数に存在した重みづけを1000と設定し, **[2]**については,  $w_{ijp}^x$  の値に対し, 元の目的関数に存在した重みづけを1, 10, 100, 1000の4種類で計算実験を行った。表4は各計算実験ごとに各日の各タイムスロットの各業務における勤務人数の不足数と過剰数  $\alpha_{jhkr}^-$ ,  $\alpha_{jhkr}^+$  それぞれの総和, 各スタッフの総勤務時間の不足と過剰  $\beta_i^-$ ,  $\beta_i^+$  それぞれの総和, そして, 現場の勤務表との一致率, 求解時間を比較したものである。

各傾向を目的関数に取り込むことで, 3章の提案モデルで得られた解よりも一致率が上がっていることを確認した。

なお, モデル **[1]** で求解時間が約9分と長くなっている理由を探るため求解の過程を観察した。最適解自体は暫定解として約4分で得られたが, 下界とのギャップが値が小さいながら埋まらず, 最適解であるという保証を得るために後半の5分を要していた。本実験では,  $w_{kii'}^y$  の値に対し, 元の目的関数に存在した重みづけを1000と設定したが, この値を小さくするとさらに求解時間が長くなることが観察された。複数の評価尺度の間で解の優劣がつきにくく, 最適値に近い目的関数値を持つ解が多く存在する状態, タイトな下界が得にくい状態であったと推測される。この状態は, 与えられた問題例や重みづけの設定により起こりうるので, 実用にあたっては求解時間に上限を設定す

る工夫が必要である。

**[3]** 各曜日各タイムスロットの各業務の平均スタッフ数

各タイムスロットの各業務の勤務人数を数え, 週の数で割ることで各タイムスロットの各業務の平均勤務人数を各曜日について算出した。本稿では, この値に対する計算実験については省略するが, この値を必要勤務人数の上下限値のデフォルト値設定に利用できれば, データ入力量の削減につながると考えられる。各業務の必要人数は, 月によって変動するため, 単なる平均ではなく, 時期的重みづけ(最近の情報を優先する, 同じ季節の情報を優先する等)と具体的なデフォルト値の設定方法を今後の課題としたい。

7. おわりに

本研究では, 多くの現場に対して適用可能なスケジューリングモデルを構築するために, 各現場の勤務表作成における違いである1日の働き方について着目した。各現場にあわせた実行可能な単日スケジュールを利用することで, 人数を揃えるだけでなく, 各スタッフの希望を考慮できるモデルを提案した。このモデルに基づき実際の店舗を対象に計算実験を行い, 高速に勤務表を作成することができた。そして, Microsoft Office Excelで簡単にデータを入出力でき, 提案モデルを基に汎用ソルバを利用できる支援システムを構築した。必要データを入力することで, 短い時間(十数秒)で勤務表を得ることができるようになると同時に, スタッフの勤務時間量や勤務人数が妥当かどうかの検討もできるようになった。

謝辞 本研究は, JSPS 科研費 26350435, 25350434 の助成を受けたものです。

参考文献

- [1] 繁野麻衣子, 池上敦子: スタッフスケジューリング, サービスサイエンスことはじめ, 高木英明(編), pp.177-210, 筑波大学出版会(2014).
- [2] Ernst, A.T., Jiang, H., Krishnamoorthy, M., Owens, B. and Sier, D.: Annotated Bibliography of Personnel Scheduling and Rostering, *Annals of Operations Research*, Vol.127, pp.21-144 (2004).
- [3] Ernst, A.T., Jiang, H., Krishnamoorthy, M. and Sier, D.: Staff Scheduling and Rostering: A review of Applications, Methods and Models, *European Journal of Operational Research*, Vol.153, pp.3-27 (2004).
- [4] Asgeirsson, E.I. and Siguroardottir G.L.: Near-Optimal MIP Solutions for Preference Based Self-Scheduling, *Annals of Operations Research*, Vol.215, pp.222-241 (2014).
- [5] Bard, J.F., Binici, C. and deSilva, A.H.: Staff Scheduling at the United States Postal Service, *Computers & Operations Research*, Vol.30, pp.745-771 (2003).
- [6] 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.183-189 (2012).
- [7] IBM ILOG CPLEX, available from <http://www-03.ibm.com/software/products/ja/ibmilogeple>.
- [8] NTT データ数理システム, Numerical Optimizer, available from <http://www.msi.co.jp/nuopt/>.

付 録

A.1 勤務表作成に必要なデータの例

表 A.1 スタッフの可能業務 (16 人分)

Table A.1 Tasks possible for each staff member (16 people).

スタッフ ID	常勤業務	ホール	キッチン
1	●		
2		●	
3		●	
4		●	
5		●	
6		●	
7		●	
8		●	
9		●	
10		●	
11		●	
12		●	
13			●
14			●
15			●
16			●

表 A.2 各スタッフの勤務希望時間帯 (31 日のうちの 6 日分)

Table A.2 Availability of each staff member (6 of 31 days).

スタッフ ID	10月1日	10月2日	10月3日	10月4日	10月5日	10月6日
1	14:00 24:00		14:00 24:00	14:00 24:00	14:00 24:00	
2			18:00 23:00			17:00 23:00
3	17:00 23:00		17:00 23:00			
4		18:00 24:00		17:00 24:00		14:00 23:00
5	17:00 24:00		17:00 24:00	17:00 24:00	18:30 24:00	14:00 24:00
6		18:00 24:00				18:30 24:00
7	19:00 24:00					
8			17:00 22:00	17:00 22:00	17:00 22:00	17:00 22:00
9	17:00 21:30	17:30 21:30	17:00 21:30		18:00 21:30	16:30 21:30
10	17:00 21:30	17:00 21:30	17:00 21:30	17:00 21:30	17:00 21:30	
11	17:00 24:00	17:00 24:00		17:00 24:00		
12	17:00 24:00	17:00 24:00			17:00 24:00	17:00 24:00
13						18:00 24:00
14				17:00 23:00		
15	17:00 24:00	17:00 24:00	17:00 24:00	17:00 24:00		14:00 24:00
16			17:00 24:00	17:00 24:00		

表 A.3 個別単日スケジュール数 (31 日のうちの 6 日分)

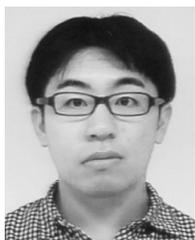
Table A.3 Number of one-day schedules (6 of 31 days).

スタッフ ID	10月1日	10月2日	10月3日	10月4日	10月5日	10月6日
1	1	0	1	120	120	0
2	0	0	21	0	0	36
3	36	0	36	0	0	0
4	0	36	0	55	0	105
5	55	0	55	55	0	136
6	0	36	0	28	0	28
7	21	0	0	0	0	0
8	0	0	21	21	21	21
9	15	10	15	0	6	21
10	15	15	15	15	15	0
11	55	55	0	55	0	0
12	55	55	0	0	21	55
13	0	0	0	0	0	28
14	0	0	0	28	0	0
15	45	45	45	0	0	120
16	0	0	45	45	0	0

表 A.4 ある 1 日の各タイムスロットの各業務の勤務人数の上下限 (上下限が等しい例)

Table A.4 Upper and lower bounds for the number of staff in each time slot (here, the bounds match).

	14:00	14:30	15:00	15:30	16:00	16:30	17:00	17:30	18:00	18:30	19:00	19:30	20:00	20:30	21:00	21:30	22:00	22:30	23:00	23:30
常勤業務人数上限	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
常勤業務人数下限	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ホール人数上限	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	1	1	1	1	1
ホール人数下限	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	1	1	1	1	1
キッチン人数上限	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
キッチン人数下限	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1



徳永 拓真

2013年成蹊大学工学部情報科学科卒業。2015年同大学大学院理工学研究科博士前期課程修了。現実問題のモデル化に興味を持ち、本研究を進めた。日本OR学会学生会員。



田中 勇真 (正会員)

2007年愛知工業大学卒業。2012年名古屋大学大学院情報科学研究科博士後期課程修了。博士(情報科学)。2012年より成蹊大学助教。組合せ最適化研究に従事。2009年FIT論文賞, 2012年情報処理学会全国大会大会奨励賞。



小林 隆文

2006年(現)現株式会社リクルートジョブズ入社。事業開発室所属。日本の雇用における課題解決のための新規事業開発に従事。



沓水 佑樹

2011年(現)株式会社リクルートジョブズ入社。事業開発室所属。日本の雇用における課題解決のための新規事業開発に従事。



池上 敦子 (正会員)

立教大学理学部数学科卒業。成蹊大学助手、講師、准教授を経て、2009年同大学教授。現実問題のモデル化に興味を持ち組合せ最適化研究に従事。博士(工学)。日本OR学会事例研究奨励賞ならびに事例研究賞、日本人間工学会研究奨励賞、スケジューリング学会技術賞。スケジューリング学会理事、横幹連合理事、日本OR学会理事を経験。