

# 飲食店におけるアルバイトのシフト管理

2016SS041 三宅藍

指導教員：福嶋雅夫

## 1 はじめに

高校生や大学生などの学生は収入を得るためにアルバイトをしている人が多い。タウンワークマガジンでのアンケート結果のデータ [1] によると、7 割以上が学業のかたわらアルバイトをしている。

アルバイト先が求める日時や曜日、または自分の都合に合わせてスケジュールを決めるタイプの働き方をシフト制という。本研究では、シフト制のアルバイトを対象とし、アルバイト先である店側のコストが最小となるようなシフト管理について考察する。シフト管理とは、複数の時間帯で働くアルバイトのシフトを管理することである。シフト管理と一言と言っても、様々な必要条件 [2] を満たして、人件費を抑えることは簡単なことではない。そのため満たさなければいけないいくつかの制約条件を考慮した上で、店側のコスト削減につながるようなシフト管理をすることを目的とする。

## 2 研究の内容

本研究では、店がアルバイトに支払う給料の合計が最小となるようなシフト管理問題を混合整数計画問題 [3] として定式化する。その際、以下の事柄を考慮する。

アルバイトの時給は一人ひとり違う。また、深夜になると時給は高くなるため、勤務時間帯によっては割増がある。

アルバイトはそれぞれ経験に差があるため、アルバイト一人ひとりに経験値を与える。時給は経験値に依存する。

アルバイトのシフト管理を行うには、日にちや時間帯の様々な組み合わせを考える必要がある。そのため、アルバイトにはいくつかの勤務パターンが存在する。

## 3 定式化

前節で述べた事柄を考慮すると、本研究の問題は以下のように定式化できる。まず定式化に用いる記号、定数および変数を整理する。

記号の定義

$\mathcal{N} := \{1, 2, \dots, N\}$  : アルバイトの集合

$\mathcal{D} := \{1, 2, \dots, D\}$  : 勤務日の集合

$\mathcal{T} := \{1, 2, \dots, T\}$  : 時間帯の集合

$\mathcal{P} := \{1, 2, \dots, P\}$  : 勤務パターンの集合

定数の定義

$e_i \geq 0$ : アルバイト  $i$  の経験値。

$E_t \geq 0$ : 時間帯  $t$  において少なくとも 1 人のアルバイトに必要とされる経験値。

$C_i \geq 0$ : アルバイト  $i$  の時給。

$\beta_t \geq 1$ : 時間帯  $t$  の割増率。

$\delta_{pt} \in \{0, 1\}$ : 勤務パターン  $p$  が時間帯  $t$  を含むと

き 1, 含まないとき 0。

$l_t \geq 0$ : 時間帯  $t$  の時間数。よって勤務パターン  $p$  の勤務時間数は  $\sum_{t=1}^T l_t \delta_{pt}$  となる。

$m_{td}$ : 勤務日  $d$  の時間帯  $t$  に必要なアルバイトの人数。

$\mathcal{A}_i$ : アルバイト  $i$  に割り当て可能なシフトの集合。

$L_i$ : アルバイト  $i$  の期間内での勤務時間の上限。

変数の定義

$x_{idp} \in \{0, 1\}$ : アルバイト  $i$  が勤務日  $d$  に勤務パターン  $p$  で働くとき 1, 働かないとき 0。

以上の記号を用いると、アルバイトの人件費の合計を最小化する問題は次のように表される。

$$\min \sum_{i \in \mathcal{N}} \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}} C_i \left( \sum_{t \in \mathcal{T}} \beta_t l_t \delta_{pt} \right) x_{idp} \quad (1)$$

$$\text{s.t.} \sum_{p \in \mathcal{P}} x_{idp} \leq 1 \quad (i \in \mathcal{N})(d \in \mathcal{D}) \quad (2)$$

$$\sum_{d \in \mathcal{D}} \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}} l_t \delta_{pt} x_{idp} \leq L_i \quad (i \in \mathcal{N}) \quad (3)$$

$$\sum_{i \in \mathcal{N}} \sum_{p \in \mathcal{P}} \delta_{pt} x_{idp} \geq m_{td} \quad (t \in \mathcal{T})(d \in \mathcal{D}) \quad (4)$$

$$y_{td} \geq E_t \quad (t \in \mathcal{T})(d \in \mathcal{D}) \quad (5)$$

$$M(1 - z_{itd}) \geq y_{td} - \sum_{p \in \mathcal{P}} e_i \delta_{pt} x_{idp} \geq 0 \quad (6)$$
$$(i \in \mathcal{N})(t \in \mathcal{T})(d \in \mathcal{D})$$

$$\sum_{i \in \mathcal{N}} z_{itd} = 1 \quad (t \in \mathcal{T})(d \in \mathcal{D}) \quad (7)$$

$$z_{itd} \in \{0, 1\} \quad (i \in \mathcal{N})(t \in \mathcal{T})(d \in \mathcal{D}) \quad (8)$$

$$x_{idp} \in \{0, 1\} \quad (i \in \mathcal{N}) \quad (9)$$

$$x_{idp} = 0 \quad (d, p) \notin \mathcal{A}_i \quad (i \in \mathcal{N}) \quad (10)$$

式 (1) は人件費の総額を表す目的関数であり  $\sum_{t \in \mathcal{T}} \beta_t l_t \delta_{pt}$  は勤務パターン  $p$  が含む時間帯  $t$  の時間数にその時間帯の時給の割増率の  $\beta_t$  を掛けたものの合計を表す。式 (2) は各アルバイトに対して 1 日に割り当てる勤務パターンは高々 1 つであることを表す。式 (3) は、勤務パターン  $p$  の勤務時間数  $\sum_{t=1}^T l_t \delta_{pt}$  を用いて、各アルバイトの期間内での勤務時間の上限を表している。式 (4) は勤務日  $d$  の時間帯  $t$  に最低必要なアルバイトの人数を表す。式 (5)~(7) は、各時間帯  $t$  で最低  $E_t$  の経験値をもつアルバイトが 1 人は必要であるという制約

$$\max_{i \in \mathcal{N}} \sum_{p \in \mathcal{P}} e_i \delta_{pt} x_{idp} \geq E_t \quad (t \in \mathcal{T})(d \in \mathcal{D})$$

において、左辺を新しい連続変数  $y_{td}$  で置き換え、さらに 0-1 変数  $z_{itd}$  と十分大きい正定数  $M$  を用いることにより、この制約式を書き換えたものである。なお、 $y_{td}$  は勤務

日  $d$  の時間帯  $t$  において、勤務しているアルバイトの中で最大の経験値を持つアルバイトの経験値を表す変数であり、 $z_{itd}$  は各アルバイトが最大の経験値を持つかどうかを表わす変数である。最適解において、 $z_{itd} = 1$  となるアルバイト  $i$  が経験値最大のアルバイトとなる。

#### 4 数値実験

実際の問題として、飲食店でのシフト管理について考える。各時間帯において最低必要な経験値や曜日によって最低必要な人数を変化させ、その結果を比較して考察する。数値実験には *Gurobi* を用いた [4]。

まず、基本設定としてアルバイトは 15 人、勤務日は月曜から日曜の 7 日間、時間帯は 3 つ、勤務パターンは 6 つとし、アルバイト  $i$  に対して割り当て可能なシフトの集合  $\mathcal{A}_i$  を与えた。また、各アルバイトに対して時給と経験値も定めた。とくに、経験値は 1~5 の 5 段階とした。アルバイト 1 と 2 は経験値 1、アルバイト 3~7 は経験値 2、アルバイト 8 と 9 は経験値 3、アルバイト 10~12 は経験値 4、アルバイト 13~15 は経験値 5 とした。時間帯 1,2,3 の時間数をそれぞれ 5 時間ずつとし、勤務パターン 1 は時間帯 1 だけを、勤務パターン 2 は時間帯 2 だけを、勤務パターン 3 は時間帯 3 だけを、勤務パターン 4 は時間帯 1,2 を、勤務パターン 5 は時間帯 2,3 を、勤務パターン 6 は時間帯 1,2,3 すべてを含むものとし、時間帯 3 は夜ということで時給を割増させる。各アルバイトに対して、1 週間で働ける上限も与えた。

**例 1.** 必要な人数はすべての曜日で時間帯 1 は 3 人、時間帯 2 と 3 は 4 人ずつとした。少なくとも 1 人に必要な経験値も時間帯 1 は 2、時間帯 2 は 3、時間帯 3 は 4 として計算を行った。そうすると、時給が低く経験値の低いアルバイトのシフトが優先的に組まれることになり、人件費を抑えることができた。その結果を表 1 に示す。カッコ内の数字  $(d, t)$  は  $d$  日目の時間帯  $t$  を表し、各々の欄に書かれた数字はその時間帯に働くアルバイトの番号を表す。

**例 2.** 例 1 では、どの曜日にも必要なアルバイトの人数は同じとしたが、これはあまり現実的ではないので飲食店の忙しい金土日の必要な人数を増やした。そうすると、各アルバイトが 1 週間で働ける上限はきまっているため、時給が低く経験値の低いアルバイトばかりをシフトにくみこむことなく、偏りなくアルバイトのシフトを決めることができた。ただし、人数を増やしたことで、例 1 での人件費と比較すると、コストは上昇した。

**例 3.** さらに、各時間帯において少なくとも 1 人に必要とされる経験値を高く設定して計算を行った。そうすると、経験値の高いアルバイトをシフトに組み込まざるを得なくなるため、例 1 と比較すると人件費は高くなる。しかし、例 2 と比較すると人件費の差はそれほど小さくなく、その上で経験値の高いアルバイトをシフトに組むことができるため、店側としては安心して円滑に営業を進めることができるようなシフトが得られた。

表 1 例 1 の計算結果

(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)
1,2,3	1,2,3,10	2,3,4,10	4,6,11	2,3,4,11	2,3,4,11
(3,1)	(3,2)	(3,3)	(4,1)	(4,2)	(4,3)
1,3,4	2,3,4,11	2,3,4,11	1,2,3	1,2,3,8	1,2,3,11
(5,1)	(5,2)	(5,3)	(6,1)	(6,2)	(6,3)
1,2,3	1,2,3,9	1,2,3,10	1,3,5	1,2,4,8	1,2,4,11
(7,1)	(7,2)	(7,3)			人件費の総額
3,4,5	1,2,4,8	1,2,4,11			430087

表 2 例 2 の計算結果

(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)
1,2,3	1,2,3,10	2,3,4,10	4,6,11	2,3,4,11	2,3,4,11
(3,1)	(3,2)	(3,3)	(4,1)	(4,2)	(4,3)
1,3,4	2,3,4,11	1,2,3,4	1,2,3	1,2,3,8	1,2,3,11
(5,1)	(5,2)	(5,3)	(6,1)	(6,2)	(6,3)
1,2,3	1,2,3,9	1,2,3,4,10	1,5,7,9	1,2,3,4,7,8	1,2,3,4,6,7,11
(7,1)	(7,2)	(7,3)			人件費の総額
4,7,8,9	1,2,3,4,7,8	1,2,3,4,5,6,10			509637

表 3 例 3 の計算結果

(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)
2,3,13	1,2,3,13	1,2,3,13	4,6,14	2,3,4,14	2,3,4,14
(3,1)	(3,2)	(3,3)	(4,1)	(4,2)	(4,3)
1,3,11	2,3,4,11	2,3,4,14	1,2,11	1,2,3,10	1,2,4,13
(5,1)	(5,2)	(5,3)	(6,1)	(6,2)	(6,3)
1,2,10	1,2,4,10	1,2,3,4,13	1,5,6,11	1,2,3,4,5,12	1,2,3,4,5,7,15
(7,1)	(7,2)	(7,3)			人件費の総額
1,5,6,8,12	1,2,3,4,5,10	1,2,3,4,5,7,15			516700

#### 5 まとめ

アルバイトのシフトを組む上で、人件費が最小になるように、つまり店側の利益が最大になるような定式化を行った。様々な数値実験を行い、各時間帯で働くアルバイトの人数や必要とされる経験値を適切に設定することにより、人件費は多少高くなるが、店側が円滑に営業できるようなシフト管理が可能であることが確認できた。

#### 参考文献

- [1] タウンワークマガジン (最終閲覧日:2019 年 9 月 19 日) <https://townwork.net> > baitoSearch
- [2] 飲食業の経費 (最終閲覧日:2019 年 9 月 24 日) [www.kobekaijyo.com/inshokusushi/keihi.html](http://www.kobekaijyo.com/inshokusushi/keihi.html)
- [3] 大野勝久, 逆瀬川浩孝, 中出康一: Excel で学ぶオペレーションズリサーチ, 近代科学社, 2015
- [4] 久保幹雄, J.P ペドロソ, 松村正和, A. レイス, 新しい数理最適化-Python 言語と Gurobi で解く, 近代科学社, 2012 年