

フードコートにおけるアルバイト従業員の勤務シフト作成に関する研究

日野 駿 (沼田 一道 教授, 松浦 隆文 助教)

1. はじめに

現在、飲食店の多くは、その従業員の多くをアルバイトとして雇用している。一般的にアルバイト従業員は、勤務希望時間を自由に提出し、シフト作成者は、その範囲内(拡張する場合もあるが)で勤務シフトを作成している。その際、勤務シフトは、全営業時間帯について、店舗を運営できる最低限の人数を確保していなければならない。

アルバイト従業員のシフト作成は、常勤の従業員を対象とする毎日のローテーションを中心としたスケジューリング[1]と異なり、時間帯ごとに異なる必要人数を確保する様に人員を割り当てることが中心となる。このとき、1単位時間ごとにアルバイトを割り当てることはできず、一定単位数以上の連続勤務を割り当てることになるが、その結果、人員の過剰が生ずる可能性がある。そこで、必要人員を満たしながら、無駄な割り当てを極力減らすことが問題となる。

アルバイト従業員の勤務シフトを扱った研究は多数あるが[2] [3]、多くは、2種類程度の仕事に対して、各アルバイト従業員の作業の可能性を考慮し、勤務時間帯を決定するものである。ここでは、配置された従業員が勤務時間帯中に仕事(職種)が変わることは考慮されていない。

本研究では、仕事の種類が多数ある「フードコート」におけるアルバイト従業員の勤務シフトスケジュールを考える。ここでは、勤務時間中に仕事(店舗)が変わることが可能であるとして運営されている。

2. 問題

2. 1. 現状

フードコートとは、屋台の様な形式で飲食店を出店し、飲食スペースを共有する、セルフサービス形式の飲食施設の事である(図 1)。本研究で扱うフードコート A は、施設全体を一社で経営し、各店舗で働いている従業員を一括雇用している。従業員は雇用された際に、本籍店舗を与えられる。勤務シフトは、基本的にそれにしたがって作成されているが、他店舗の人員不足など、必要に応じて他店舗にヘルプとして勤務する事がある。例えば、5時間出勤するアルバイトが店舗 a で3時間、店舗 b でヘルプとして2時間勤務するなどのパターンがあり、店舗(業務)間の移動回数も含めるとパターンは多数ある。

現在、フードコート A でのシフトの作成は、各店舗の担当者が店舗ごとに行っている。このとき、ある店舗で人員不足の時間帯が見つければ、人員過剰となっている他店舗の従業員から勤務可能な人を探し、ヘルプ勤務として補充している。ヘルプ勤務が必要となった場合、誰が勤務可能で、かつ出勤できるのか、その度に確認しなければならない。そのためシフト作成に、非常に時間がかかっている。

2. 2. 問題の整理

本研究では、ヘルプ勤務を前提として勤務シフトを作成し、余分な割り当てをできるだけ少なくする勤務シフトを考える。これまで、各店舗で別々に作成していたシフト作成をフードコート全体で一



図 1: フードコートの営業形態

括して行い、従業員の出勤時間帯と勤務配置(ある時間にどの店舗で働くか)まで作成することを主眼に置く。この問題を最適化問題として定式化し、求解実験を行う。求解の際は、モデルに取り込めなかった条件(休憩の時間の考慮等)を入力データの変更で補正することを試み、その有効性を吟味する。

問題を解く上で、まず営業時間を、1時間=1コマとなる時間帯に分割する。次に、全従業員に対して、個人データとして、本籍店舗であるスペシャリスト店舗(業務)と、ヘルプ可能店舗(業務)を与える。加えて、アルバイト従業員は勤務希望の時間帯を提出する。次に、各店舗(業務)に対して、全ての時間帯に営業に必要な人数と必要スペシャリスト人数をそれぞれ設定する。従業員の勤務時間は、予め設定する最短(最長)勤務時間に従う。また、6時間以上連続で勤務する場合、間に休憩を取ることが法律で定められているため、休憩を取らせる。その間に時給は発生しない。本研究では、休憩が必要となるような時間帯を予測し、休憩する従業員の代わりに勤務する従業員を、代替人員としてデータである各店舗の必要人員に加えている。

問題を扱う上で、詳細な条件を以下のように設定する。

(1)従業員は勤務希望の時間帯以外に働かない。(2)店舗や業務間の移動には、時間はかからない。(3)全従業員の時給は一律同額とする。(4)従業員は、勤務配置を希望できない。(5)従業員の休憩時間に、時給は発生しない。(6)従業員は、同時に2つ以上の配置に入ることはない。

3. 定式化

前節で示したアルバイトの勤務スケジュールを決定する問題を数理計画問題として定式化

<記号・データの定義>

- i ($i = 1 \sim n$): 店舗(業務)番号
- j ($j = 1 \sim m$): 従業員番号
- t ($t = 0 \sim T$): 時間帯
- L, U : 最短(最長)勤務時間
- r_{it} : 時間帯 t に店舗 i に必要な従業員数
- d_{it} : 時間帯 t に店舗 i に必要なスペシャリストの人数
- $s_{ij} \in \{0,1\}$: 従業員 j が店舗 i のスペシャトならば1, そうでなければ0とする
- $h_{ij} \in \{0,1\}$: 従業員 j が店舗 i をヘルプ可能ならば1, そうでなければ0とする
- $w_{jt} \in \{0,1\}$: 従業員 j が時刻 t に勤務可能ならば1, そうでなければ0とする

<決定変数>

$x_{jtk} \begin{cases} 0: \text{従業員}j \text{を, 時間帯}t \text{から}k \text{時間連続で働かせない} \\ 1: \text{従業員}j \text{を, 時間帯}t \text{から}k \text{時間連続で働かせる} \end{cases}$

$y_{ijt} \begin{cases} 0: \text{従業員}j \text{が, 店舗}i, \text{ 時間帯}t \text{において働かない} \\ 1: \text{従業員}j \text{が, 店舗}i, \text{ 時間帯}t \text{において働く} \end{cases}$

<目的関数>

営業時間中の人時(出勤した従業員の勤務時間の総計)を最小にする。

<最小化>
$$z = \sum_{j=1}^m \sum_{k=L}^U \sum_{t=0}^{f-k+1} kx_{jtk} \quad (1)$$

<制約条件>

$$\sum_{k=L}^U \sum_{p=0}^{k-1} x_{j t-p k} \geq y_{ijt} \quad (\forall i, \forall j, \forall t) \quad (2)$$

$$\sum_{i=1}^n y_{ijt} \geq \sum_{k=L}^U \sum_{p=0}^{k-1} x_{j t-p k} \quad (\forall j, \forall t) \quad (3)$$

$$\sum_{k=L}^U \sum_{t=0}^{T-k+1} x_{jtk} \leq 1 \quad (\forall j) \quad (4)$$

$$\sum_{j=1}^m y_{ijt} \geq r_{it} \quad (\forall i, \forall t) \quad (5)$$

$$\sum_{j=1}^m s_{ij} y_{ijt} \geq d_{it} \quad (\forall t) \quad (6)$$

$$\sum_{i=1}^n y_{ijt} \leq w_{jt} \quad (\forall j, \forall t) \quad (7)$$

$$y_{ijt} \leq h_{ij} \quad (\forall i, \forall j, \forall t) \quad (8)$$

$$\sum_{i=1}^n y_{ijt} \leq 1 \quad (\forall j, \forall t) \quad (9)$$

(1)式は、従業員の延べ勤務時間を最小にする目的関数である。(2)式は、従業員の勤務開始時間帯と勤務時間についての制約である。(3)式は、従業員 j が時間帯 t から k 時間働くならば、いずれかの店舗で働いていることを表す。(4)式は、決められた1日の勤務可能時間を超えて働けないことを示す。(5),(6)式は、必要最低人数と必要最低スペシャリスト人数を満たすようにする制約である。(7)式は前提条件の(2)を意味する。(8)式は、勤務可能な店舗のみで勤務できることを表す。(9)式は、同時に2つ以上の店舗で働くことができないことを表す。

4. 求解実験

本研究では、フードコートAの現状をもとに5つのデータを作成した。全てのデータにおいて、店舗数を($n = 7$)、従業員数を150名($m = 150$)、営業時間を17時間($T = 16$)とする。 r_{it} と d_{it} は営業実態を基に設定し、従業員に関するデータである s_{ij} と h_{ij} は、実際の従業員データを参考にランダムに設定した。各従業員は1つの本籍業務に加えて、2つ程度のヘルプ可能業務が与えられるような確率で乱数を発生させた。 w_{it} も同様にランダムに発生させ、最短(最長)勤務時間は、それぞれ $L = 3, U = 8$ とする。休憩する従業員の代わりに勤務する従業員数は、各店舗4名ずつとし $4 \times 7 = 28$ 人とする。

3節で提起した数理モデルとデータファイルを用いて、汎用ソルバGurobi 5.6.0によって解を求めた。使用した計算機はMac OS X 10.9.1, プロセッサ 1.8GHz Intel core i7, メモリ 4GBである。

5. 結果と考察

数値実験の結果、長くても500[s]程度で厳密解を求めることができた。実店舗では、1週間のシフト作成に3~4時間程度かかっている。提案解法では、1日分のスケジュールを10分弱で作成できるため、1時間程度で1週間分を作成できる。

5種類のデータに対して求解した結果、人員の過剰な割り当ては発生しなかった。しかし、各データにおいて、店舗間でヘルプ勤務を禁止($h_{ij} = s_{ij}$)し求解すると、一部のデータでは実行可能解が得られなかった。つまり、ヘルプ勤務を可能とすることで、営業時間中の人員確保が可能になっていると言える。

また、休憩が必要となる6時間~8時間勤務($k = 6, 7, 8$)の従業員数は13~15名程度であり、長時間勤務をする従業員は、開店作業や閉店作業の時間近辺に勤務していることが多かった。そこで、営業時間中の3コマ目(r_{i3})と14コマ目(r_{i14})の従業員を増やしたデータ r_{it} を作成し直し(増やした従業員数が休憩中の代替人員として働く)、休憩の時に代替となる増員人時を $2 \times 7 = 14$ となるように設定して再実験

を行った。

再実験においても、5種類のデータに対して過剰な割り当ては発生せず、休憩が必要となる従業員数は14名程度であった。代替勤務として r_{it} に加えた14人時という想定から大きく外れる数値ではなく、休憩中の代替人員の確保したスケジュールを作成することができた。

ソルバによって得られた解の一部を、図2に示す。図2の横軸は時間帯、縦軸は従業員番号を表す。マスの中に番号は勤務店舗を表す。

得られたスケジュール結果を見ると、本籍勤務とヘルプ勤務を短時間で繰り返すことがあった。これは、制約条件に移動回数を制限する制約を含まなかったためと考えられる。勤務配置の移動回数の上限を設定

する制約を加えれば、解決できると考える。しかし、この制約を加えると、過剰な人員割り当てが発生する原因ともなる。また、長時間勤務する場合などには、移動回数が多いことで勤務が単調にならない等のメリットもあるため、制限の仕方や上限の設定方法は、慎重に設定する必要があると考える。

6. まとめと今後の課題

本研究では、ある店舗で人員不足の時間帯が見つければ、人員過剰となっている他店舗の従業員から勤務可能な人を探し、ヘルプ勤務として補充が可能なアルバイト従業員の勤務シフトと配置問題を数理計画モデルとして定式化し、汎用ソルバを用いて厳密解を求めた。実店舗の営業時間、各時間帯での必要人員数、アルバイト従業員数に則したデータを作成し、数値実験を実行し、モデルの有効性を示すことができた。

本研究では、休憩する従業員の代わりとして、必要人員 r_{it} の特定の時間帯に、休憩する従業員の代わりとなる従業員を増員した。しかし、長時間勤務する従業員がいつ発生するかを、完全に予測することは難しく、今回のモデルでは不確実性が残る。

今回提案した数理計画モデルは、一営業日のシフト作成であったため、従業員の勤務時間や出勤頻度の公平性については言及していない。しかし、通常シフトスケジュールを行う場合、1週間や一ヶ月など一定期間中のシフトスケジュールを作成する。勤務の公平性を考慮し、長期間のシフトスケジュールが可能なモデルに拡張することは今後の課題である。

参考文献

- [1] 池上敦子, 丹羽明, 大倉元宏(1996), 我が国におけるナース・スケジューリング問題, オペレーションズ・リサーチ, 41巻, pp.436-442.
- [2] 中島 啓介(2002), 「ファーストフード店における準社員勤務時間帯割り付け問題」, 平成十三年度卒業研究抄録集, 東京理科大学工学部第二部経営工学科, pp.233-236.
- [3] 矢川陽一郎(2011), 「ファーストフードA店におけるアルバイトのシフトスケジュール作成支援ソフトの試作」, 平成二十二年度卒業抄録集, 東京理科大学工学部第二部経営工学科, pp.61-64.
- [4] 林 晴比古(2005), 新訂 新C言語入門 ビギナー編, 第5刷, ソフトバンクパブリッシング, 322pp.

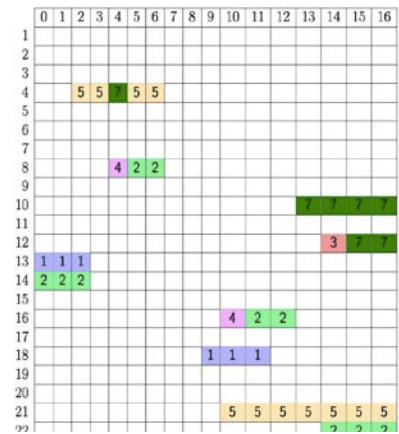


図 2 : 実行結果の抜粋