Optimization of Library Staffing Schedule

by Ayesha Sareen, Sanketh Gadhe and Sai Bhargava

MS. Business Analytics

University of Cincinnati, Spring 2023

Abstract

The optimal scheduling of library staff is an essential task to ensure that the library functions smoothly and provides high-quality services to its users. This optimization project aims to schedule the staff of a library in an efficient and effective manner, ensuring that minimum staffing requirements are met, and workload is evenly distributed among staff members. The project considers various constraints, such as the maximum number of hours each staff member can work per day and per week, ensuring that staff members have adequate breaks and cannot work during times they are not available.

Overall, this project demonstrates the usefulness of mathematical modeling and optimization techniques in solving real-world problems such as staff scheduling in a library. The model can be adapted and applied to other contexts with similar requirements for staff scheduling.

Table of Contents

Contents

1. Introduction

Managing staffing levels in libraries can be challenging, especially during peak periods when the number of visitors increases. Hiring too few staff members can lead to overcrowding and frustrated patrons, while hiring too many staff members can result in unnecessary expenses. To optimize staffing levels and minimize costs, we need to develop an effective scheduling system that ensures the right number of staff members are available at the right time.

In this project, we aim to solve the problem of optimal staff scheduling for a library with a specific set of constraints. Our objective is to minimize the total cost of staffing while ensuring that the library has sufficient staff members available during peak hours.

We will develop a linear programming model to optimize the staffing levels, taking into account the constraints related to staff availability, working hours, and mandatory breaks. Our model will ensure that all staff members are available during peak periods and that they do not exceed their maximum working hours or take breaks simultaneously. By using this model, we aim to find the optimal staff scheduling solution that will help the library to operate efficiently and cost-effectively.

2. Literature review

Optimal staff scheduling has been a long-standing problem in various fields, including healthcare, transportation, and retail. In recent years, several researchers have focused on developing efficient scheduling algorithms to optimize staffing levels in libraries. In this literature review, we highlight some of the key findings and approaches used in previous studies related to optimal staff scheduling for libraries.

One common approach to staff scheduling in libraries is to use mathematical models to determine the optimal number of staff members required during different hours of operation. Various mathematical models have been proposed, including linear programming, integer programming, and simulation-based models. These models take into account the staffing requirements based on factors such as visitor traffic, shelf maintenance, and circulation desk duties.

In a study by a group of graduate students at Portland State University, a mixed integer programming model was proposed for optimizing staff scheduling in the university library. The model considers staff availability, workloads, and preferences to minimize the surplus hours and maximize the staff preferences while meeting the required service level. The results of the study showed that the proposed model could effectively optimize staff scheduling and achieve objectives.

Overall, there are several studies that demonstrate the various mathematical, heuristic, and simulation-based models can be used to optimize staffing levels in libraries. These models consider the constraints related to staff availability, working hours, and mandatory breaks to ensure that the library operates efficiently and cost-effectively while meeting the required service level.

3. Problem definition

Efficient staffing is crucial to ensure the smooth functioning of library operations. The library has n librarians and m shelvers. The library is open from Monday to Sunday for 12 hours each day. Each staff member can work for at most 8 hours each day and a maximum of 40 hours in a week. After every 4 hours of work, each staff member must take a mandatory 1-hour break. All staff members cannot be on break at the same time. The library requires a specific number of librarians and shelvers at different times of the day. The hourly cost of librarians is \$P, and the hourly cost of shelvers is \$Q.

The objective of this project is to create a scheduling optimization model to minimize the total cost of staffing while ensuring that the required number of librarians and shelvers are available at all times during library operations. The project will involve formulating the optimization problem as a linear programming model with decision variables and constraints.

4. Mathematical model

Objective: Minimizing total cost of staffing

$$Minimize\left\{P * \sum_{h=1}^{12}\sum_{d=1}^{7}\sum_{i=1}^{n}(L_{i,h,d}) + Q * \sum_{h=1}^{12}\sum_{d=1}^{7}\sum_{j=1}^{m}(S_{j,h,d})\right\}$$

Decision variables:

$L_{i,h,d}$: 1 if Librarian i works on day d and hour h, else 0
where $i \in [1, n]$, $h \in [1, 12]$, d $\in [1, 7]$
$S_{j,h,d}$: 1 if Shelver j works on day d and hour h, else 0
where $j \in [1, \mathrm{m}]$, $h \in [1, 12]$, d $\in [1, 7]$

We can formulate the problem as a set of linear constraints as follows:

1. Total hours worked by each staff member should not exceed 40 hours per week:

$$\sum_{h=1}^{12}\sum_{d=1}^{7}(L_{i,h,d}) \leq 40 \; for \; 1 \leq i \leq n$$

$$\sum_{h=1}^{12}\sum_{d=1}^{7}(S_{j,h,d}) \leq 40 \; for \; 1 \leq j \leq m$$

2. Each staff member cannot work for more than 8 hours in a day:

$$\sum_{h=1}^{12}(L_{i,h,d}) \leq 8 \; for \; 1 \leq i \leq n, 1 \leq d \leq 7$$

$$\sum_{h=1}^{12}(S_{j,h,d}) \leq 8 \; for \; 1 \leq j \leq m, 1 \leq d \leq 7$$

3. After every 4 hours of work, 1 hour break is mandatory for each staff member:

$$\sum(L_{i,h,d} + L_{i,h+1,d} + L_{i,h+2,d} + L_{i,h+3,d}) \leq 4 - L_{i,h+4,d}$$

$$for \; 1 \leq i \leq n, 1 \leq h \leq 8, 1 \leq d \leq 7$$

$$\sum (S_{j,h,d} + S_{j,h+1,d} + S_{j,h+2,d} + S_{j,h+3,d}) \leq 4 - S_{j,h+4,d}$$

$$for\ 1 \leq j \leq m, 1 \leq h \leq 8, 1 \leq d \leq 7$$

4. All staff members should not be on break at the same time:

$$\sum_{i=1}^{n} (L_{i,h,d}) + \sum_{j=1}^{m} (S_{j,h,d}) \geq 1\ for\ \ 1 \leq h \leq 12, 1 \leq d \leq 7$$

5. Computational Results

Using the PuLP library in Python, we have implemented the above problem on synthetic data for the purpose of this project.

Code:

```
# Import PuLP library
from pulp import *

# Define the problem as a minimization problem
prob = LpProblem("Staffing Problem", LpMinimize)

# Define the decision variables
n = 3 # number of librarians
m = 2 # number of shelvers
P = 25 # cost of one librarian per hour
Q = 20 # cost of one shelver per hour

L = LpVariable.dicts("Librarian", [(i, h, d) for i in range(1, n+1) for h in range(1, 13) for d in
range(1, 8)], 0, 1, LpInteger)
S = LpVariable.dicts("Shelver", [(j, h, d) for j in range(1, m+1) for h in range(1, 13) for d in
range(1, 8)], 0, 1, LpInteger)

# Define the objective function
prob += P * lpSum([L[(i, h, d)] for i in range(1, n+1) for h in range(1, 13) for d in range(1, 8)])
+ Q * lpSum([S[(j, h, d)] for j in range(1, m+1) for h in range(1, 13) for d in range(1, 8)])

# Add the constraints
for i in range(1, n+1):
    prob += lpSum([L[(i, h, d)] for h in range(1, 13) for d in range(1, 8)]) <= 40
    for d in range(1, 8):
        prob += lpSum([L[(i, h, d)] for h in range(1, 13)]) <= 8

for j in range(1, m+1):
    prob += lpSum([S[(j, h, d)] for h in range(1, 13) for d in range(1, 8)]) <= 40
    for d in range(1, 8):
        prob += lpSum([S[(j, h, d)] for h in range(1, 13)]) <= 8

for i in range(1, n+1):
    for d in range(1, 8):
        for h in range(1, 9):
```

```
            prob += lpSum([L[(i, hh, d)] for hh in range(h, h+4)]) <= 4 - L[(i, h+4, d)]

for j in range(1, m+1):
    for d in range(1, 8):
        for h in range(1, 9):
            prob += lpSum([S[(j, hh, d)] for hh in range(h, h+4)]) <= 4 - S[(j, h+4, d)]

for h in range(1, 13):
    for d in range(1, 8):
        prob += lpSum([L[(i, h, d)] for i in range(1, n+1)]) + lpSum([S[(j, h, d)] for j in range(1,
m+1)]) >= 1

# Solve the problem
prob.solve()

# Resource allocation
for v in prob.variables():
    if v.varValue > 0:
        print(v.name, "=", v.varValue)

# Print the status of the problem
print(f"status: {prob.status}, {LpStatus[prob.status]}")

# Optimal value
print("Total Cost = ",f" {prob.objective.value()}")
```

Output:

```
Librarian_(1,_1,_3) = 1.0
Librarian_(2,_1,_7) = 1.0
Librarian_(2,_5,_3) = 1.0
Librarian_(3,_1,_4) = 1.0
Shelver_(1,_1,_5) = 1.0
Shelver_(1,_10,_2) = 1.0
Shelver_(1,_10,_3) = 1.0
Shelver_(1,_10,_6) = 1.0
Shelver_(1,_10,_7) = 1.0
Shelver_(1,_11,_4) = 1.0
Shelver_(1,_11,_5) = 1.0
Shelver_(1,_12,_2) = 1.0
Shelver_(1,_12,_3) = 1.0
Shelver_(1,_12,_6) = 1.0
Shelver_(1,_2,_1) = 1.0
Shelver_(1,_2,_3) = 1.0
Shelver_(1,_2,_4) = 1.0
Shelver_(1,_2,_5) = 1.0
Shelver_(1,_2,_6) = 1.0
```

```
Shelver_(1,_3,_1) = 1.0
Shelver_(1,_4,_2) = 1.0
Shelver_(1,_4,_4) = 1.0
Shelver_(1,_4,_5) = 1.0
Shelver_(1,_4,_6) = 1.0
Shelver_(1,_5,_2) = 1.0
Shelver_(1,_5,_5) = 1.0
Shelver_(1,_5,_6) = 1.0
Shelver_(1,_5,_7) = 1.0
Shelver_(1,_6,_1) = 1.0
Shelver_(1,_6,_3) = 1.0
Shelver_(1,_6,_4) = 1.0
Shelver_(1,_6,_5) = 1.0
Shelver_(1,_6,_6) = 1.0
Shelver_(1,_6,_7) = 1.0
Shelver_(1,_7,_2) = 1.0
Shelver_(1,_7,_7) = 1.0
Shelver_(1,_8,_1) = 1.0
Shelver_(1,_8,_2) = 1.0
Shelver_(1,_8,_3) = 1.0
Shelver_(1,_8,_5) = 1.0
Shelver_(1,_8,_6) = 1.0
Shelver_(1,_8,_7) = 1.0
Shelver_(1,_9,_1) = 1.0
Shelver_(1,_9,_6) = 1.0
Shelver_(2,_1,_1) = 1.0
Shelver_(2,_1,_2) = 1.0
Shelver_(2,_1,_6) = 1.0
Shelver_(2,_10,_1) = 1.0
Shelver_(2,_10,_4) = 1.0
Shelver_(2,_10,_5) = 1.0
Shelver_(2,_11,_1) = 1.0
Shelver_(2,_11,_2) = 1.0
Shelver_(2,_11,_3) = 1.0
Shelver_(2,_11,_6) = 1.0
Shelver_(2,_11,_7) = 1.0
Shelver_(2,_12,_1) = 1.0
Shelver_(2,_12,_4) = 1.0
Shelver_(2,_12,_5) = 1.0
Shelver_(2,_12,_7) = 1.0
Shelver_(2,_2,_2) = 1.0
Shelver_(2,_2,_7) = 1.0
Shelver_(2,_3,_2) = 1.0
Shelver_(2,_3,_3) = 1.0
Shelver_(2,_3,_4) = 1.0
Shelver_(2,_3,_5) = 1.0
Shelver_(2,_3,_6) = 1.0
Shelver_(2,_3,_7) = 1.0
Shelver_(2,_4,_1) = 1.0
Shelver_(2,_4,_3) = 1.0
Shelver_(2,_4,_7) = 1.0
Shelver_(2,_5,_1) = 1.0
Shelver_(2,_5,_4) = 1.0
Shelver_(2,_6,_2) = 1.0
Shelver_(2,_7,_1) = 1.0
Shelver_(2,_7,_3) = 1.0
Shelver_(2,_7,_4) = 1.0
Shelver_(2,_7,_5) = 1.0
Shelver_(2,_7,_6) = 1.0
```

```
Shelver_(2,_8,_4) = 1.0
Shelver_(2,_9,_2) = 1.0
Shelver_(2,_9,_3) = 1.0
Shelver_(2,_9,_4) = 1.0
Shelver_(2,_9,_5) = 1.0
Shelver_(2,_9,_7) = 1.0
status: 1, Optimal
Total Cost =   1700.0
```

6. Conclusion

In this paper, we have developed a linear programming optimization model to minimize the total cost of staffing for a library with a certain number of librarians and shelvers.

We have successfully formulated the objective function and constraints as linear programming equations with summation notation. We have used binary decision variables to represent the presence or absence of each staff member at each hour of each day. The optimization model can be solved using a linear programming solver to determine the optimal staffing schedule that minimizes the total cost of staffing while meeting all the constraints.

Further improvements can be made to the model, such as considering the skill levels of different staff members and their availability on different days of the week, to better reflect real-world staffing needs. Overall, this optimization problem provides an excellent example of how linear programming can be applied to solve practical staffing problems in various industries.

References

Gupta, Aayushi; Babu, Anju; Vrushabhendra, Lipishree; Purwar, Shivani; and Doosa, Shravankumar, "Shift Scheduling Optimization for PSU Library" (2018). Engineering and Technology Management Student Projects. 2105.

https://machinelearninggeek.com/solving-staff-scheduling-problem-using-linear-programming/

https://machinelearninggeek.com/solving-linear-programming-using-python-pulp/

https://towardsdatascience.com/linear-programming-and-discrete-optimization-with-python-using-pulp-449f3c5f6e99

https://realpython.com/linear-programming-python/

https://www.analyticsvidhya.com/blog/2022/03/linear-programming-discrete-optimization-with-pulp/