****

# Background

 Café Bistro (the actual business name has been changed for privacy reasons) is a fast-casual restaurant. One of our group members has been working at a Café Bistro for a year. During this year, she noticed that the restaurant builds its schedule out manually for a week at a time. This caused problems when people called out sick or when large orders were placed the night before because managers were unable to shift schedules or efficiently contact the right people to come in and cover shifts. The table in Figure 1 is an example of what the current schedule looks like.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Employee** | **W** | **Th** | **F** | **S** | **S** | **M** | **T** |
| A | Shift 1Shift 2Prod | Shift 1Service | na | na | na | Shift 1Prod | Shift 1Prod |
| B | Shift 3Service | Shift 3Service | na | Shift 2Shift 3Service | na | Shift 3Service | Shift 3Service |
| C | Shift 1Service | Shift 1Service | Shift 1Shift 2Service | na | Shift 1Service | na | na |
| D | Shift 2Service | na | Shift 2Service | Shift 1Service | Shift 3Service | na | na |
| ... | ... | ... | ... | ... | ... | ... | ... |

Figure 1: Café Bistro’s Current Schedule

Employees are trained in one of four trainings: Production, Service, Delivery, and Management (Figure 2). Each of these trainings are tailored to one or many workstations. Most employees are cross-trained to work at different stations. This is outlined in Figure 3. For example, all employee types are cross-trained to work dining.



Figure 2: Training Provided to the employees



Figure 3: Jobs they can perform based on training

The current schedule is based off of employee availability and estimated demand. Since Café Bistro is a 12-hour restaurant and runs three 4 hour shifts each day, employees provide their availability for the shifts they are able to work. When they are first hired, employees submit their weekly availability through an online form and must get manager approval to change it, so an employee’s availability is consistent pretty consistent from week to week with the exception of sick days and requested days off. From there the manager generates a printout of everyone’s availability to create the schedule for the week.

Since this is currently a manual process, there are some challenges that occur when the schedule is created. The challenges faced due to the current scheduling methodology are:

* People dependent over process dependent
* Time taken to Schedule is high
	+ It takes a couple of hours for the manager to figure out who to schedule for what time
* Scope for Error is high
	+ Manual error checking (shift manager must manual check if demand is covered)
	+ Lack of employee schedule crosscheck (people are scheduled outside their availability or are scheduled when they have requested off. The manager cannot easily check if certain people are available to cover shifts)
* Time to reschedule in case of minor changes is high
	+ If an employee calls in sick, the shift manager does not have a way to easily redistribute people who are currently in the store or a way to figure out who is available to call in to cover
* It’s not dynamic
	+ Static schedule (once the week has started, the schedule cannot be edited)
* The process is not standardized

## What Makes This Problem Interesting?

 There are a few things we had to take into consideration when building out our model. Due to constraints that the Café Bistro manager gave to us, we had to consider:

* Cross Trained Employees
	+ Employees can work across different workstations
* Multiple Jobs to perform
	+ There are multiple workstations for which the demand has to be covered
* Dynamic Employee Availability
	+ Every employee has her/his own availability
* Dynamic Employee Requirement
	+ Demand for employees across stations is different both across days and shifts within a day
* What if someone calls in sick
	+ Whom to call in case an employee who is initially scheduled calls in sick

#

# Approach

Our goal was to build up an automated model that met demand while minimizing employee utilization. We recognized that this type of problem required a scheduling model using the Simplex LP. The first approach we considered was a complete overhaul with a week-long model. After presenting this idea to the manager, he mentioned that it is hard to implement a large amount of change in a short period so we should focus on improving the day-to-day schedule.

Our next approach was to transform the current schedule table into decision variables. The most important part is that the decision variables should be binary. Another major thing we had to consider when building the model was that the decision variable had to take into account cross-trained employees.



Figure 4: Model Framework

 Per our discussion with the Café Bistro manager, we recognized that we needed to build a model that created a schedule that met the workstation shift requirement while minimizing employee utilization. As demonstrated in Figure 4, the manager will be able to input workstation shift requirements based on forecasted demand for the day. The model will consider employee availability, employee training, and restaurant policies and labor laws to output employee workstation shift requirements.

Our final model, which will be discussed in depth in the next section, was able to take all these variables into account to minimize employee utilization. Employee utilization is measured by the total amount of shifts worked per day. For instance, if a cross-trained employee of production (cooking, dinning) is assigned to Shift 1 and Shift 2 for dining on Wednesday, it would look like Figure 5.



Figure 5: Example Decision Output

The model we created had to be a dynamic. Since demand and employee availability is subject to change, we needed to create a model that could be easily altered in the moment. For example, if an employee A1 called out sick, then the manager can change their availability, rerun the model, and see who is available to cover the shift.

As with most businesses, Café Bistro is focused on reducing cost. Since all employees (with the exception of managers) are paid at a standard set wage, minimizing the number of shifts scheduled per day equates to minimizing the cost of labor.

# Mathematical Model

Since the money paid to an employee is a constant $12 per shift and $22 per manager and we always needed one manger on floor, minimizing employee utilization would minimize cost. Also, after discussion with the business, they wanted scheduling to be done at a day level.

We need not consider the break time required for each employee as the restaurant had a flexible break time that employees take when it is not crowded.

We wanted to retain the idea of understanding the requirement at different stations for the different shifts and schedule employees based on their training.

We started with streamlining the process by collecting the data for

1. Employees availability
2. Employee training
3. Station requirement
4. Restaurant Policies & Labor Laws

Our deployment of the mathematical model was done at a day level where in the whole model was built for a day, i.e. the manager can input the requirement at different stations for the different shifts in a day. Based on employee availability and training, model will automatically allocate employees to stations in different shifts to cover demand while minimizing total shifts used as the employees are paid on shift basis which in turn means minimization of cost while covering demand

The same model can then be implemented for different days. The things that vary across days is employee availability and station requirement at different shifts. The current model can be replicated to work for all seven days in a week. The manager was happy with the automated approach to scheduling and suggested few improvements that would make the model more user friendly.

Our current model can take care of:

Dynamic demand

Dynamic availability

Cross Station Training

With the above approach, we can setup the following mathematical model.

**Model.**

Parameters:

$$i:employee, i\in \{1,2,…,18\} $$

$$j:station, j\in \{Cooking,Dining,Cash,Delivery,Manager\}⇔j\in \{1,2,3,4,5\}$$

$$k:shift, k\in \{Morning,Lunch,Evening\}⇔k\in \{1,2,3\} $$

$$T\_{i}:employee i is trained in T\_{i}, T\_{i} \in \{Production, Service, Delivery, Manager\}$$

$$\left(i,j\right)\in f, $$

$$where f is the set of all feasible combinations of employee and stations based on training$$

$$A\_{ik}:availability of employee i for shift k$$

$$R\_{jk}:requirement at station j for shift k$$

$$M:maximum allowed utilization, 2 shifts ⇔8 hours $$

Decisions:

$x\_{ijk}:whether to allocate employee i to station j for shift k$, $\left(i,j\right)\in f, k \in \{1,2,3\}$

$$\min\_{ }\sum\_{k}^{}\sum\_{\left(i,j\right)\in f}^{}x\_{ijk}$$

Constraints:

$$\sum\_{j}^{}x\_{ijk}\leq A\_{ik}, k\in \left\{1,2,3\right\}, i\in \{1,2,…,18\} \left(1\right)$$

$$\sum\_{i}^{}x\_{ijk}\geq R\_{jk}, k\in \left\{1,2,3\right\}, j\in \left\{1,2,3,4,5\right\} \left(2\right)$$

$$\sum\_{k}^{}\sum\_{j}^{}x\_{ijk} \leq 2, i\in \{1,2,…,18\} \left(3\right)$$

$$x\_{ijk}\in \left\{0,1\right\}, \left(i,j\right)\in f, k\in \left\{1,2,3\right\} \left(4\right)$$

Notes:

1. The objective function captures the total employee utilization in terms of shifts
2. The Constraint (1) ensures that an employee is scheduled only if s/he is available for that shift. It also helps in ensuring that in each shift, an employee is scheduled to only one station
3. The Constraint (2) ensures that the dynamic requirement at each station for different shifts is met
4. The Constraint (3) ensures that none of the employees are scheduled to work more than 2 shifts (8 hours) on any given day
5. The Constraint (4) ensures that there will be a binary assignment of stations i.e. Destiny either is assigned to a station or she is not

**Optimal Solution. The following is the solution obtained from Excel Solver.**

# Final Excel Model

 Our final model was built in Excel using Solver. Since there are a lot of employees the model has to account for, the model in Figure 4 only shows a portion of the employees for a particular day. The blue box contains employee relevant information. The orange cells are the employee’s availability for the day. The green cells are the shift and station that the employee is assigned to for the day. The red box is the labor hours constraint or Max Utilization Allowed constraint. This makes sure employees do not work more than 2 shifts (or 8 hours) a day. The purple box contains the shift demand requirement constraint for each workstation. The green box contains the objective function which is to minimize employee utilization. On this particular day, 22 shifts were required to meet the demand for the entire day.



...



Figure 4: Final Excel Model

# Future Decisions/Conclusion

## Wins/Improvements

With our new model, we achieved many improvements standardizing and streamlining the Café Bistro scheduling process. If Café Bistro decides to implement this model, the manager will be able to reduce the time it takes to create the schedule. First off, due to the automatization Excel allows, the new scheduling process is now process dependent opposed to people dependent. Secondly, using our model could minimize scheduling error. The model has employee availability built in so it will not allow employees to be scheduled when they are not available. Third, it could let manager make in the moment changes to the schedule. For instance, demand will greatly increase during the holidays (Christmas, Thanksgiving, etc.), but the shift requirement can be adjusted to compensate for the increase in demand. If someone calls in sick, the manager has the ability to adjust the model, which shows which employees are currently available to cover the shift. Last but not least, this model allows the manager to take cross-training employees into account which leads to the minimize of employee utilization by distributing them to different workstations. From all of these improvements above, our model leads to minimize labor cost.

## Next Steps

As per discussion with the Café Bistro manager, our current model could work well for a day-to-day schedule. Although this accomplished our goal, there is still room for improvement. One problem that both the old model and our current model run into is how to add a new employee. If the person adding a new employee to our current solver model does not have a basic knowledge of how solver works, adding an employee could prove very complex. The same goes for removing an employee. As a result, we have the potential to build a combined (week level) model.

In order to expand upon our model, we could build a tool that would act as a dynamic model. This would have a user-friendly interface that makes it easy to add and remove employees and adjust employee availability. It could create flexibility for manager to input new employees or remove former employees without changing the entire work schedule. Building a non-technical interface allows manager to reduce learning costs for scheduling.

Moreover, this model would also allow users to tag a characteristic to a day (such as tagging a day Black Friday, Christmas, Thanksgiving) so the model will automatically take that into account. This tool could also allow for an even more dynamic approach to the employee schedule than our current model. In our current model, adjustments to Thursday will not affect Wednesday or Friday. Depending on how we wanted to expand the model, it could be useful for the model to talk across days. For example, say Friday was Black Friday and Café Bistro expected an increase in demand that weekend. In the new dynamic model, the manager could increase demand for Friday - Sunday and the model could adjust by shifting around the employees currently scheduled to be better equipped to meet demand.

## Conclusion

 Our model has the potential to be a powerful tool for Café Bistro. If given the opportunity, we would like the ability to convert our current model from an Excel model into Python model which could lead to higher efficiency. From the above analysis, we strongly recommend that Café Bistro utilizes our scheduling model. This will help the restaurant minimize labor cost and time wasted on manually creating models each week. Thanks to this project, our team is eager to expand our current knowledge of modeling and explore other possibilities for this model such as if it could work for other kinds of situations.