CSR Scheduling Plan Proposal

B06705040 廖郁華 B05303124 鄭亦辰 T08501105 Furkan Bashan B05303028 蔡孟燐

I. Our Plan

One of the major challenges in running a business or an organization is CSR scheduling. Considerations such as employees' well-being and customer satisfaction are some factors that are essential yet challenging to fulfill. In the proposed CSR scheduling plan, we have come up with our optimal solution plan using excel and python in solving the problem, following some essential criteria. Given requirements of manager and senior limitation are being fulfilled to the best of our ability to ensure operational safety. On top of that, employee shift and day-off requests are being considered into the final scheduling as the well-being of an employee is an important factor to a company's success. Summing up all of the factors mentioned above, and with the help of online tools, we believe that the CSR scheduling plan can increase the company's efficiency and effectiveness greatly.

II. Our Outcome

We organized our outcome into the following points:

1. Scheduled daily shifts according to our examination of the four shifts (morning, afternoon, night, leave) and the amount of employees demanded during the 30 min period. We made sure that the scheduled number of employees meets the demanded number, that is, equal to or greater than the demanded number.
2. A total amount of five randomly repeated patterns were examined in the demand table (column). We examined the five patterns individually and arrange the shifts for each pattern separately so that it is optimized (arranged according to the daily shift scheduling mentioned in 1)

CSR Scheduling Plan Method

B06705040 廖郁華 B05303124 鄭亦辰 T08501105 Furkan Bashan B05303028 蔡孟潾

I. Our Methods

We have used the following methods to reach our goal and fulfill the plan:

A. Most Optimized Number of Employee

We used the solver function on excel to complete this part. The objective in finding the solution to this step was to minimize total employee working for each day, and variables within this problem which changes the total employee number per day are the amount of employees assigned to each shift. Thus our first approach is to apply the constraint of *total employee assigned per day* must be greater than or equal to total employee *demanded per day*. If rows are defined as shifts *{ 1, 2, 3 … ,n }* and columns are defined as half-hour sections, then *column sum* will be representing the total number of employees working within that time interval. On the other hand, *row sum* represents *work hour times employee number in that shift*. Our second approach is to add the constraint of *workers' number appointed to a shift* must be an integer. When we provide all the conditions to the solver function, it then in turn provides us with the result of the most optimized shift assignments for five pattern.

B. Shift Distribution Among Employees

After we have the most optimized number of employees, these shifts can be distributed amongst employees by using a python algorithm. In the program, we first define a 14*31 matrix represent the lack shift for each day according to the five pattern we found from the first step.

Then we first deal with the sheet in the answer sheet. Every shift requests and leave requests are manually set into program. Other not defined shift will be set to initial value -1, defined for unarranged shifts on the employee schedule.

C. Other Restrictions/Conditions

Other restrictions are being written into the program as additional conditions as well. Weekly night and afternoon shift restrictions are defined with a function called rule(). We are going to assign the shift to every employee, we will first assigned shift 14, which is day off, and then night shift, afternoon shift, finally morning shift. The priority of assigning shift is due to the limitation extent of each shift. If the limitation is more strict, we will first deal with that shift. When we assigning the shift, we will use rule function to check if this shift can be assigned to this employee.

For manager limit, to make it simple, two random managers (in this case ID=72 and 84) are used to fulfil the rule. So they are also pre-assigned to 5 specific dates. In order to follow the senior limit rule, employees with more than 2 years of work experience were taken as potential seniors. As we know, all these seniority required shifts are one of the specific patterns *(1, 2, 3, 4 or 5)*, minimum employee number required for these is also known. After dates are specified by patterns that potential senior ID's can be defined on it. After running the program, shifts from 1 to 14 are then assigned to each employee for every day of the month. (*i* is being assigned to one of *1, 2, ..., 14*) The Algorithm prints the schedule we need as an end result.

D. Check Accuracy

Lastly, we checked the accuracy of the schedule using excel's counting function, including whether every employee has 8 days-off per month, and if the summation of each day fulfills the required demand or not.