**Employee Scheduling**

Weiyu Peng, Ya Ling Su, Venkata Praveen Naradasu

EIN6905 Data Analytics for Social Good

April 19, 2022

Content

1. **Introduction ………………………………………………………………………………….3**
2. **Data ………………………………………………………………………………………………3**
3. **Data processing……………………………………………………………………………..4**

3.1 Collect information & Generate Data………………………………………………….4

3.2 Model \_1 (greedy method with the multi-step iteration)………………………..5

3.3 Model \_2 (One Step with mixed-integer programming matching model)……8

1. **Finding and Discussion…………………………………………………………………10**

4.1 Model \_1 Pros & Cons……………………………………………………………………….10

4.2 Model \_2 Pros & Cons……………………………………………………………………….10

4.3 Conclusion……………………………………………………………………………………11

1. **Future Analysis…………………………………………………………………………….12**
2. **Introduction**

Some companies or restaurants need to arrange the work schedule and set the number of workers in need for different work areas. We know this is irregular shift work but also the basic condition for a company operation. On the employees' side, they only offer their suitable working hours, or they are willing to work day or time. Therefore, the whole process will appear some problems like some of the employees can't satisfy the min work time requirement, some of the specific time don't have worker work or don't have enough workers to fulfill this work area. This will cause the company to lose potential profit and the employees’ loyalty, in these problems.

Hence, we want to design a mechanism to satisfy two sides (the company side and the employee). How to fill the company operation demand and also the employee can pick their willing to work hours at the same time? But the core value still is how min the total cost for the company, because no company will be willing to satisfy employees' demand to increase their cost, which will decrease the company's profit. We will set some constraints, such as employees needing to conform to the min working hour per week, different work areas having min and max worker demand, etc.

In this analysis, we hope this mechanism can assist the company to reduce labor costs and fill the company operation demand without causing the company loss.

**2. Data**

We generate the data used by random normal distribution to do this analysis, so there isn't any specific dataset.

***Working Time***

From 8 am to 8 pm on workdays, 12 hours per day (total 6-time shifts)

From 8 am to 6 pm on weekends, 10 hours per day (total 5-time shifts)

By dividing the whole working hours into two-hour time periods as working time, we have 6 optional work time on workdays and 5 optional work time on weekends.

***Employees’ preference***

Every employee has his own preference for work time. (Generate 10 preferences per person)

***Constraints***

There must be enough workers at each work hour.

Every employee should work at least 10 hours(if possible) but less than 20 hours per week.

***Assumption***

At least 5 people work each work time

**3. Data processing**

***Collect information & Generate Data***

We ask a few friends about their preferred work times when they are doing a part-time job trying to know what day or the time shift they are willing to do the part-time job. After the summary of the information, we collected from our friends, we find that people like to work from Tuesday to Friday, but Saturday, Sunday, and Monday are not like that. Because Saturday and Sunday are weekends, most people like to have fun rather than work, for Monday, people are still not willing to work. For the time shifts, first, we separate 2 hours as a time shift and then assume the working time on the weekday is from 8 am to 8 pm have 6-time shifts, but at the weekends is different, the working time is from 8 am to 6 pm only have 5-time shifts. We find that this tendency (workday and the time shift) is similar to a normal distribution. The resulting table is shown in Table 1. Hence, we used the random normal distribution to generate the days and the work time shifts as ***day\_stats***and ***time\_stats*** to do further analysis.

After generating data, we got the ***day\_stats*** and ***time\_stats***, next step is to generate the preference data for the candidates because each worker has 10 preference work time shifts with the day, so all the candidates will have their preference working list. The resulting table is shown in Table 2.

Chart, histogram

Description automatically generated一張含有 文字 的圖片

自動產生的描述

Table 1. Random Normal Distribution Table 2. Candidate for 10 preference list (example)

***Model \_1 (Greedy method with the multi-step iteration)***

In the first model, we used gurobi MIP function to set a mixed integer programming matching model fit with the weight (means when the worker’s preference time shift is matched then the worker has more probability match with the next time shift) to get the single max matching and then used the greedy method with the multi-step iteration to remove the candidate already matching the time shift to see if each time shift is satisfied for 5 people and how many candidates are not hired in this matching system. The mechanism of model\_1 is shown in Table 3.

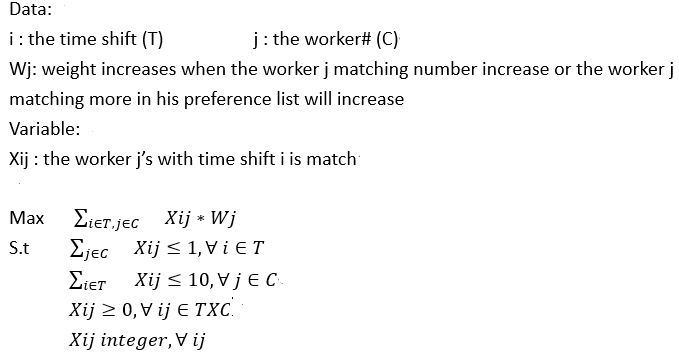


Table 3. The mechanism of model\_1

The result of the model\_1, we can see each time just matches one candidate with one option, so we use the iteration way to do the 100 times to see each time shift matching status, if this time shift is already fulfilled 5 people, it will return None and remove the time shift matching from the candidate's preference list to see each candidate left how many preference lists or how many candidates are not in the matching system.

At first, we initialized the simulation with 20 workers to run the model. We find that there are at least half of the work time shifts are not filled up. For our purpose, we want to minimize the labor cost and to fulfill each time shift with 5 people at the same time, because if each time shift can't have at least 5 people, it means the company needs to hire more people which will increase the cost. So, we try to change the number of workers to see the different results of the min of workers, un-hire workers number, and the left of the time shift number and then draw the graphs to see what is different between these data.

We change the number of workers to 20, 40, 80, 100, 150, 200 then stop because when the worker number = 200, the model already fulfills all the constraints so we can't add more workers to the model. We find when we change the worker to 80, the min labor is 33 but after the worker number increases to 150, 200, the demand for min labor decreases. The resulting graph is shown in Table 4. Hence, based on our model, we can know that no matter how many workers are put into the system, the company's max demand for workers is around 33 workers. At the same time, we also calculated the unhired workers, and we got a similar result as Table 4. The resulting graph is shown in Table 5. Based on our model, we hope to complete all the time shifts to fill up 5 people in each time shift. Also, we hope each time shift can satisfy 5 people and no time shift is unsatisfied, so we see that when we put the worker number to 80, the rest of the time shift will = 0 which means all the time shifts are already filled up with 5 people. The resulting graph is shown in Table 6.

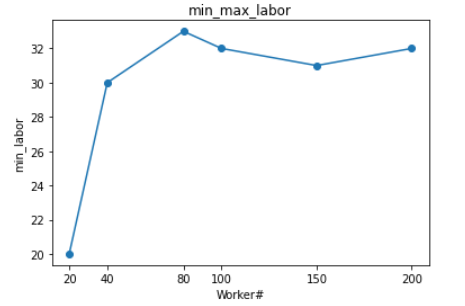
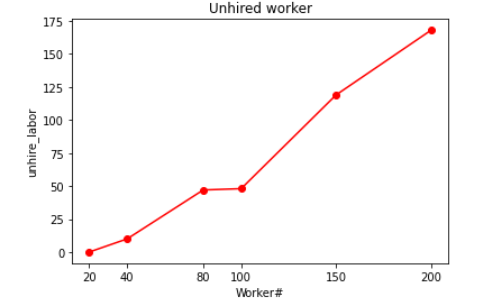
 

Table 4. Min\_Max\_Labor Table 5. Unhired Worker Number

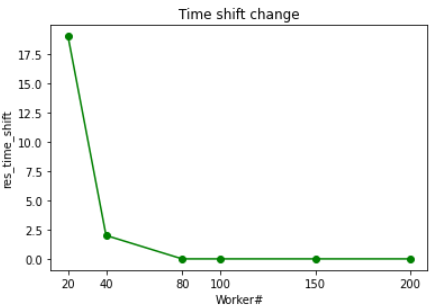
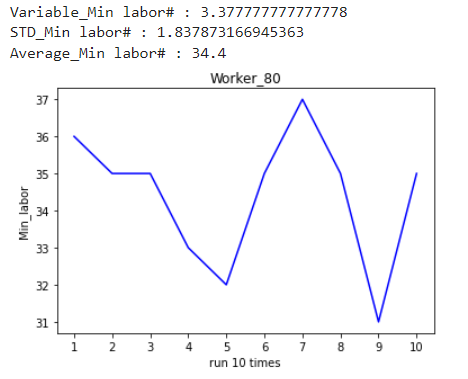


Table 6. The Rest of Work Timeshift

Following these tables and the above information, we have a conclusion when the worker= 80 will have the max labor demand and also satisfy all the work time shifts, so we run for 10 times to see the different data. Also, we add another condition is that if the candidate is in the matching system, each candidate should fulfill the min working hour of 10 hours, so we can have the percentage of candidates who are not satisfied with the min 10 hours requirement. Therefore, we have the average of min labor is 34.4, the standard deviation is 1.83, the variable is 3.37, and the percentage of not satisfied with the min 10 hours of candidates is around 40% which is a little bit high. We can say that there are 40% of workers in the hiring system who do not fill the min 10 hours requirement. The resulting graph and information are shown in Table 7 & Table 8.

一張含有 文字 的圖片

自動產生的描述

Table 7. The Rest of Work Time shift Table 8. Run 10 times data of 80 worker

***Model \_2 (One Step with mixed-integer programming matching model)***

In the second model, we used a simpler way to do it that set a mixed integer programming matching model, just using the gurobi package to build the model without the iteration method to obtain the objective of min labor, we can also calculate how many candidates are not in the hired system and all the time shifts should strictly fulfill the 5 people requirement.

The mechanism of the objective is to min the labor number in the matching system. First, we set the two variables as Xij is an integer and Zj is binary, for the constraint1 is for each time shift at least have 5 people to satisfy the work period demand, constraint 2 is for each candidate can't exceed the 10 preference list, and the constraint 3 is if the candidates i who participated in the match were assigned a value of 1, and those who were not hired were assigned a value of 0. The mechanism of model\_2 is shown in Table 9.

一張含有 文字 的圖片

自動產生的描述

Table 9. The mechanism of model\_2

The result of the model\_2, It can be easier to get the number of time shifts that each candidate can be selected and all the time shifts fill the restraint (5 people) without using any greedy way or iteration method. The mip result is shown in Table 10.

一張含有 桌 的圖片

自動產生的描述

Table 10. The mip result

Although the model\_2 is simpler it needs strictly to follow the constraints, which means if the sample size is too small, the MIP function can’t work. For example, if we put the number of workers between 20 and 40, we find that the model can’t work. Hence for the model\_2, we start the number of workers at 80, 100, 150 then stop because when the worker number = 200, the model already fulfills all the constraints so we can't add more workers to the model. We find when we change the worker to 80, the min labor is 22 but after the worker number increases to 150, the demand for min labor decreases. The resulting graph is shown in Table 11. Hence, based on themodel\_2, we can know that whatever how many workers are put into the system, the company's max demand for workers is around 22 workers. The relation between the un-hire of candidates and the worker number graph is shown in Table 12.

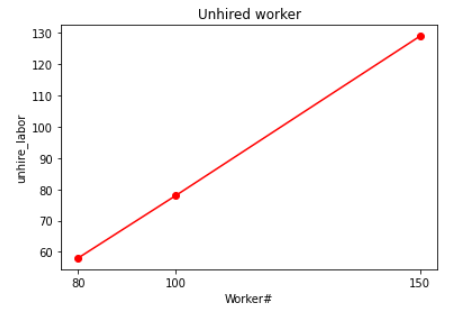
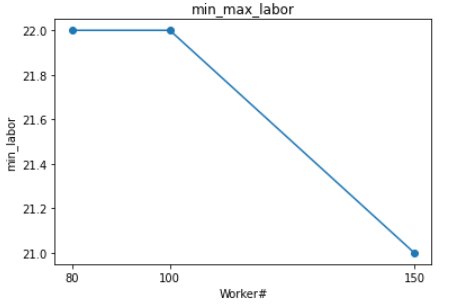
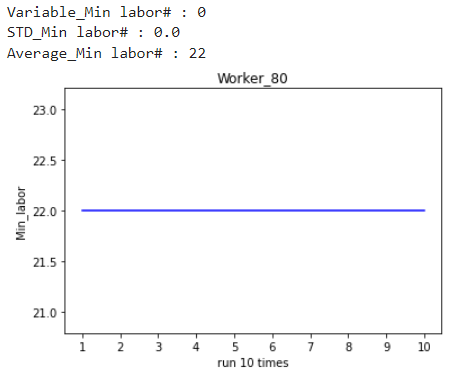


Table 11. Min\_Max\_Labor Table 12. Unhired Worker Number

Following these tables and the above information, we find that when the worker = 80 will have the max labor demand and satisfy all the work time shifts. Therefore, we run 10 times to see if there are any changes under model\_2. The run 10 times result is shown in Table 14. You can see that the min labor, the number of un-hire workers do not change and the percentage of not satisfied with the min 10 hours of candidates is 0 which means when the candidates enter the hire system, there are not any workers fulfilling the min 10 working hour. Also, we calculate that the average of the min labor is 22 because running 10 times of min labor is the same which causes the standard deviation and the variable equal to 0. The resulting graph is shown in Table 13.

一張含有 文字 的圖片

自動產生的描述

Table 13. The Rest of Work Time shift Table 14. Run 10 times data of 80 worker

**4. Finding and Discussion**

For our analysis, we design two different mechanisms, so we want to see what is different between these two models and try to find a better model for future analysis.

***Model \_1 (Greedy method with the multi-step iteration)***

**Feature:** Strong interpretability & logic and does not belong to Black box testing or White box testing.

**Pros:** The model is more flexible. In the real world, no one can determine whether the actual situation can just meet the expected constraints. Hence this method can help to approach the restriction requirement as closely as possible when meeting different circumstances (such as the number of workers in this project). Although it is hard to make it perfect, at least we can do our best to use the limited resources to satisfy the constraints. The company objective is often changed and needs to be dynamically adjusted according to the actual conditions.

**Cons:** 1. Multiple repeated steps make the process more complicated. 2. Because the model contains the greedy process which means when the data reaches a certain scale, it will appear local optimal not the global optimal, but the result of our model is still not bad. 3. The model has multiple repeated steps which means it can only approximate the constraints of the mechanism as much as possible rather than strictly adhering to some assumed prerequisite restrictions.

***Model \_2 (One Step with mixed integer programming matching model)***

**Feature:** Briefer and clearer, also not so complicated.

**Pros:** The model is more considered the global optimal objective, so when the worker number reaches a certain number, model\_2 will be better than model\_1 not stuck in the local optimal. Theoretically, when the sample becomes bigger, model\_2 can be easier to obtain the global optimal than model\_1.

**Cons:** The existing condition must strictly meet the constraints of the objective function (already consider the concept of the flexible constraint), otherwise, the model can't be executed. It means this model is only suitable when the sample size meets the ideal constraint of the mechanism.

***Conclusion***

Compared to Model\_1 and Model\_2 results, first, we find that when the model reaches the max labor even if you add more workers the max labor will be fixed or decreased. In the graph, model\_2 always hired lower workers than model\_1. The resulting graph is shown in Table 15. Second, when we used the maximin number of labors that runs for 10 times, the number of minimum labors has a slight change in the model\_1 but in the modle\_2 is always constant, which means the model\_1 is always found the local optimal not the global optimal but the model\_2 is found the global optimal. The resulting graph is shown in Table 16. Based on the above information, model\_2 is the more appropriate model than model\_1 for our project

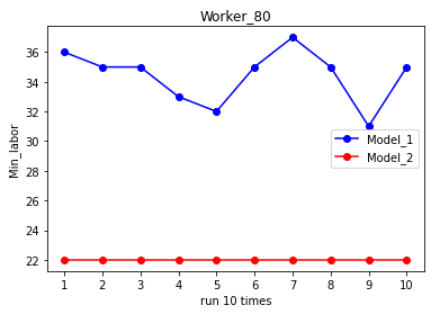
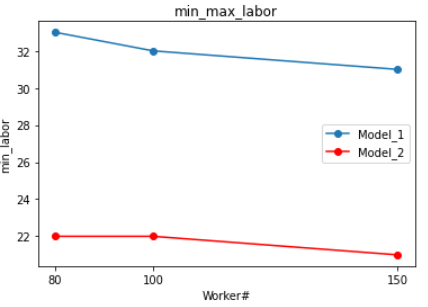


Table 15. Min\_labor between two models Table 16. Run 10 times between two models (labor=80)

However, it's not mean that the model\_1 is not good, the actual outcome is close. So based on the different situations or company restraints, picking the appropriate mechanism to build the model will be better. For model\_1, apply when the quantities of supply and demand are very close to each other, only have a small sample size and the final objective is not clear (it does not need to fully meet the final expected goal and constraint). For model\_2, it is more appropriate when the supply should exceed the demand, having a huge sample size. Also needs to follow the existing condition and must strictly meet the constraints of the objective function.

**5. Future Analysis**

Model\_1 and Model\_2 can apply to different situations or company restraints. For this time, maybe because our data sample is too small,which causes the two models don't have a significant difference. Hence, we believe that future analysis could continue to improve, such as expanding the number of samples, improving the round-robin rule, adding more constraints, increasing the weight, etc. In order to have a deeper understanding of the model that fits the different company's requirements and is efficient in assigning the worker for the time shifts approach to the minimum labor cost for the company target.