Labor Optimization Under Uncertainty Of Workers' Absenteeism And Workforce Operation & Management System Design

by

Chuan Cheng

**A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Engineering
(Industrial and Systems Engineering)
in the University of Michigan-Dearborn
2018**

Master's Thesis Committee:

**Associate Professor Jian Hu, Chair
Professor Armen Zakarian
Assistant Professor Xi Chen**

Acknowledgements

I wish to express my deep sense of gratitude to Dr. Jian Hu for his rousing inspiration and patient guidance for this research. I am also very much thankful to GDI&A of Ford Motor Company for their support on this project. Last but not least, I want to extend my appreciation to those who could not mentioned here but give me a lot of inspirations behind the curtain.

Table of Contents

List of Tables

List of Figures

Abstract

Workforce planning is a major concern in manufacturing business for running the plants smoothly and efficiently, particularly with the challenge of workers' absenteeism. Unforeseen labor shortage because of the worker absenteeism severely hinders the maintenance of production lines and negatively impact product quality. Business managers often resort to over-hiring; however, in the long run, overstaffing leads to labor waste and excessive cost. This research project designs a workforce operations and management system, with the aim to most efficiently utilize labor effort to meet the production demand under the uncertainty of workers' absenteeism. It accomplishes the following tasks: (i) provides an optimal policy of daily labor force assignment; (ii) recommends a cross-training strategy to improve employees' versatilities; (iii) makes long-term workforce planning to find an optimal quantity of employees for minimizing staff cost and controlling the risk that staff headcounts cannot meet production demand duo to absenteeism. Task (i) is achieved by developing linear programming assignment and adjustment models. The assignment model allocates show-up employees to appropriate jobs considering their individual skills and preferences, whereas the adjustment model optimally and dynamically adjusts job assignment on account of employees' coming late or leaving early. In Task (ii), a two-stage stochastic cross-training optimization model is proposed to select trainees and decide which jobs they should be trained for. Task (iii) incorporates the models developed in Tasks (i) and (ii) in a dynamic optimization model to decide staffing levels for a long-time horizon with the help of the prediction of absenteeism rate.

Chapter 1: Introduction

1.1 Introduction of Absenteeism and Workforce Planning

"To provide the right (required) number of the right (qualified) personnel at the right (specified) time at the minimum cost" (Wang, J. 2005) is always the final goal for any workforce planning system. There are various operational research techniques that been developed to model such problem, and new techniques are needed to update new models and make them more realistic.

For most plants, the workforce demand not only depends on current conditions, but the expectation about the future level of exogenous factors or uncertain factors as well. Exogenous factors like the hiring cost or firing policy may generate adjustment cost for the company so that the plants are supposed to use a dynamic model instead of independent short-term models without any connection to each other.

Uncertainty of absenteeism is another important factor should be considered in the workforce planning model. In 2017 the U.S. Department of Labor (DOL) estimated that almost 3 percent of an employer's workforce was absent on any given day. The price of dealing with absenteeism problems is very high considering the temporary alternative labor cost and the quality of product. It causes an increase in manpower to meet staffing needs, a loss in revenue that is a result of not meeting project schedules, an underutilization of capital investments (e.g., tools and equipment), an interruption of work flow and task accomplishment, a need for increased overtime, and employee fatigue (The Business Roundtable 1982). Some organizations with little flexibility would pay a lot of extra money to hire many more labors than exactly needed, while most organizations are still looking for a balance between paying for buffer labors and taking the risk of labor shortage.

In this paper, a dataset of four-year department level absenteeism rate is tested to follow beta distributions. To be more specific, we find even for different departments in one plant, the absenteeism rates of each department fit different beta distributions very well. Then we apply the distributions to represent the uncertainty of absenteeism rates.

Based on the distribution, we develop a two-stage stochastic programming model to help decide the staff level of the next several months which is usually decided by a deterministic model. With the help of a manufacturing assembly plants, we build the workforce planning model of the constraints on their real rules.

Some rules are for temporary workers that they are only allowed to work at most 3 days per week when legacy workers are working 4 days and the total number of temporary workers should be no more than a fixed upper limit related to total number of legacy worker. Some rules are for the total number of workers needed which is called require to operate (RTO). We will further discuss them in Chapter 2, 3 and 4.

1.2 Introduction of Workforce Operation and Management system

Workforce planning is always much more complicated than a final result telling the manager how many workers are needed. Our goal in this project is to design a system that can help plant from their daily operation to long-term management. The structure of the system is in the following figure.

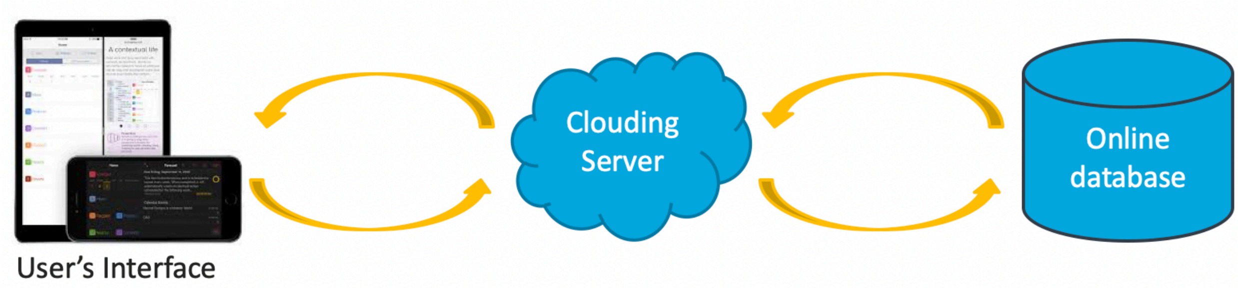

Figure 1.1: Architecture of Workforce Operation and Management system

As we can see, the yellow arrows are the data flow as well as the command flow. The raw data produced by the plant like the daily attendance will be uploaded to the clouding sever with a request for certain type of suggestion. The clouding server based on several optimization models will use the fresh uploaded data as well as the existing employees' information to return the optimal decisions to the users and also store the updated data in the online database. User's interface at the plants' end will allow information exchange between the clouding server and the plants.

1.3 Introduction of Skill Matrix, Assignment and Cross-Training

Skill matrix is the ability or skill level of a worker for one particular job. A worker is usually trained for multiple jobs in the same area, which gives higher flexibility of assignment. One simple fact is that for a well-trained team, the productivity will be as high as expected once the workers showing up to work are more than the required jobs. However, from the other side, if all workers are single functional, some jobs may not have qualified operators even there are some surplus workers for the other jobs.

In this paper, we build a linear programming model of assignment based on the skill matrix and daily attendance. This model is a supplement of workforce planning model telling the plant manager how many "right" people they need to meet the requirement. We further build several mixed integer programming models (MIP) which have more specific applications than the basic assignment model.

Also, with the historical absenteeism rate of each worker, we build a stochastic programming cross-training model to help decide which worker should be trained to which job so that the overall efficiency could be best improved. Since the training process is usually up to 3 days for one of most jobs according to the plant manager, our focus on this cross-training model is not scheduling a training process, but the outcome of training.

1.4 Structure of models

So far, we are still modifying all assignment models as well as the cross-training model to get closer to the reality. In our future work, these models could be combined with long-term workforce planning model to get a more accurate suggested staffing level. The relationship among models and the future structure will be presented in the following chart.

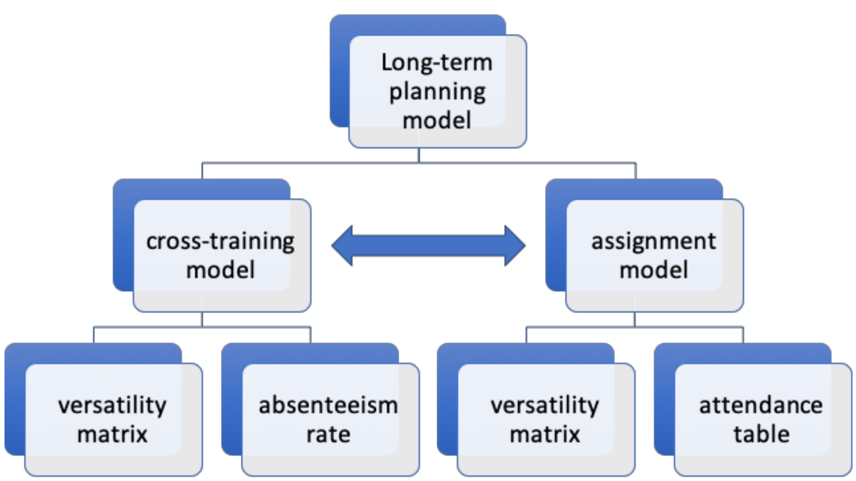

Figure 1.2: Models Structure

As we can see from the chart, the cross-training model and assignment together help make up the slave stage of long-term planning model. From one direction, cross-training model can provide the optimal updated skill matrix to best improve the efficiency of daily assignment; from the opposite direction, the assignment model can serve as the operational level constraints of the cross-training model.

1.5 Literature Review

Operations research models applied to workforce planning have existed for really long time. Holt et al. (1960) provides one of the earliest examples of applying mathematical programming to model employee staffing. Similar mathematical-programming approaches to workforce planning are provided by Lippman et al. (1967), Orrbeck et al. (1968), Ebert (1976), and others. These are all deterministic models which make simply and straight forward connection to reality. Grinold and Stanford (1974) develop the first deterministic dynamic-programming model with linear costs and constraints in this field, while Mehlmann (1980) incorporates stochasticity by modeling employee transitions as a Markov chain. More recently, Gans and Zhou (2002) develop a Markov Decision Process (MDP) model which represents the stochastic nature of demand uncertainty caused by employees' learning and turnover on jobs. The theoretical results show that a "hiring-up-to" policy, which is similar to "order-up-to" policy in inventory models, is optimal, but their model does not include the effect of temporary absences. Pinker and Larson (2003) use dynamic programming to study the use of contingent labor when demand is uncertain. Guerry, M. A. (2014) introduce discrete-time 2-stated Markov chain model to solve the embeddable problem as well as the inverse problem in manpower planning, based on the concept of m-th root probability matrix.

Hopp and Van Oyen (2004) describe how cross-training benefits the organizations in cost, time, quality, and variety. They also indicate that cross-training is ‘broadly applicable, powerful, and also highly complex’. Nembhard (2007) addresses the application of workforce cross-training in manufacturing industries.

Chapter 2: Long-term workforce planning Model

This chapter is organized as follows. Section 1 introduces the background of long-term workforce planning and the problems in reality that this model may help to solve. Section 2 describes a two-stage stochastic programming model about long-term staff level planning. Section 3 gives a case study of an automotive assembly plant with the help of this model.

2.1 Introduction

To provide more favorable working conditions and other benefits, plants are always willing to give their employees reasonable number of vacation and FMLA (Family and Medical Leave). According to the U.S Department of Labor, FMLA is designed to help employees balance their worker and family responsibility by allowing them to take up to 12 weeks of unpaid, job-protected leave per year for certain family and medical reasons.

A worker could be absent with FMLA in any working days without noticing the supervisor in advance, which leads to high uncertainty of the number of workers show up to work. The most common way to deal with such uncertainty is to maintain a buffer beyond the RTO (required to operate), or borrow some worker from other department who has surplus workers if needed.

Our model considers the distributions of daily absenteeism rate of different type of worker, department, month and shift, to maintain dynamic staff level which ensure a certain probability of show-up worker satisfying RTO and a higher probability of satisfying another indicator line under RTO. The monthly adjustment cost (hiring and firing) combined with the salary cost will be minimized to help save money for the plants.

2.2 Long-term Workforce Planning Model

We built a two-stage stochastic dynamic programming model to help decide the staff level in the next several months, considering the absenteeism rates and costs for hiring, working, sending home or temporarily lay-off of both legacy and temporary workers. Also, a very important rule of keeping a required portion of legacy and temporary workers is applied in the model.

c_d^f and c_d^t are the cost for salary for legacy and temporary workers at shift d. c_k^{TLO} is the cost if a worker (legacy) is temporarily lay-off for the entire month k when the plant is predicted to run over staffed for the next month with current on-roll workers. The plant would pay the *TLO* cost rather than pay the go-home cost for each day in such over-staffed case. c_h^f and c_h^t are the hiring cost for any extra legacy or temporary workers that the plant need to hire not from the very beginning. $Q_{k,m}$ is a set of all the shifts in month *k* week m. *AR* is the average absenteeism rate of the past year used in calculating the ratio of max temporary workers. *LL* is the number of long-term leave workers used in calculating the max number of temporary workers. *SL* and *SLT* is the stress level for meeting RTO or part of RTO, which means the plant should have higher probability of having required number of workers than these two fixed bounds. *TOL* is the shortage tolerance of department which indicates the department can sometimes run under RTO. Tw_k^f is total number of legacy workers from other departments which are not the target of this optimization. *RTC* is a fixed ratio of total temporary workers could be used in our target department. $S_{i,d}^f(\omega)$ and $S_{i,d}^t(\omega)$ are the stochastic absenteeism rate for legacy or temporary workers at area i shift d in random sample ω.

x_d^i and y_d^i are the decision variables of the numbers of legacy or temporary workers assigned to area i at shift d. w_k^f and w_k^t are the total numbers of legacy or temporary workers available for month k. θ_k and l_k are the numbers of laying-off, temporarily for legacy workers and permanently without any cost for temporary workers.

Table 2.1: Parameters Definition

Parameter	**Definition**
c_d^f	cost of legacy worker at shift d
c_d^t	cost of temporary workers at shift d
c_k^{TLO}	monthly cost of temporary lay-off
c_h^f	hiring cost of a Legacy worker
c_h^t	hiring cost of a temporary worker
$Q_{k,m}$	a set of all the shifts in month k week m
AR	average absenteeism rate used in calculating the ratio of max temporary worker
LL	number of long − term leave worker
SL	stress level
SLT	stress level under some tolerance
TOL	shortage tolerance for the department
Tw_k^f	total fixed legacy worker from other departments not in this optimization
RTC	a fixed temporary workers ratio for certain department
$S_{i,d}^f(\omega)$	the stochastic absenteeism rate for legacy worker at area i shift d in random sample ω
$S_{i,d}^t(\omega)$	the stochastic absenteeism rate for temporary workers at area i shift d in random sample ω

Decision variables:

x_d^i	number of legacy workers assigned to area i shift d
y_d^i	number of temporary workers assigned to area i shift d
w_k^f	number of legacy workers available for month k
w_k^t	number of temporary workers available for month k
h_k^f	new − hiring legacy workers at month k
h_k^t	new − hiring temporary workers at month k
θ_k	number of temperary lay − off of legacy at month k
l_k	number of lay − off of temporary workers at month k

In the following model, the first stage describes the relationships between each two months next to each other. $J_k(I_k, w_k^f, w_k^t)$ is the total cost from month k to the end of period that we are considering, including the salary cost of month k, the hiring and firing cost from month k to the next month k+1 and the expectation of total cost from month k+1. $J_k(I_k, w_k^f, w_k^t)$ is minimized in (2.1). (2.2) is the constraint which makes the number of legacy workers available for month k+1 should equal to that number at month k add hiring number and minus the TLO planned for month k+1. Since temporary workers can be laid off without any cost at the end of each month while the TLO for legacy is planned at the beginning of each month, (2.3) for temporary workers is slightly different from (2.2) for legacy worker that the lay-off number calculate the number of current month *k*.

The second stage considers several constraints of scheduling within a month, which means the total number of legacy workers and temporary workers should be treated as an input from the first stage. Since the absenteeism rate for legacy or temporary workers should follow beta distributions rather than a fixed value, we use (2.5) to present the probability of the total number of show-up workers greater than required number of workers should be greater than *SL*. Similarly, we have (2.6) for the probability constraint where a higher probability is required for lower RTO with the given tolerance. (2.7) is the ratio constraint that the upper limit of total temporary workers in some departments should depend on the total number of legacy worker, according to the rules of plants to ensure the reliability of the organization as well as protect the benefits of workers' union. (2.8) represents another rule to those temporary workers that each temporary worker can only work for 3 days per week. (2.9) and (2.10) means the plants can only make assignment for those available workers, and cannot have any extra workers in the middle of each month. And of course, both legacy and temporary workers' number should be non-negative integer. However, we only require positive real numbers in (2.11) for faster solving the problem, and then rounded the result to get an approximation.

$\mathbf{stage\ 1\ (\ long-term\ planning)}$:

$$J_k(I_k, w_k^f, w_k^t) = \min E\left\{f_k(I_k, w_k^f, w_k^t) + c_h^f \cdot h_k^f + c_h^t \cdot h_k^t + c_k^{TLO} \cdot \theta_k + J_{k+1}(I_{k+1}, w_{k+1}^f, w_{k+1}^t)\right\} \quad (2.1)$$

$$s.t. \qquad w_{k+1}^f = w_k^f + h_{k+1}^f - \theta_{k+1} \quad (2.2)$$

$$w_{k+1}^t = w_k^t + h_{k+1}^t - l_k \quad (2.3)$$

$\mathbf{stage\ 2\ (\ monthly\ scheduling)}$:

$$f_k(I_k, w_k^f, w_k^t) = \sum_{d \in D} \sum_{i \in I} (c_i^f \cdot x_d^i + c_i^t \cdot y_d^i) \quad (2.4)$$

$$s.t. \qquad Pr\left\{x_d^i \cdot \left(1 - S_{i,d}^f(\omega)\right) + y_d^i \cdot \left(1 - S_{i,d}^t(\omega)\right) \geq RTO_{i,d} \forall i\right\} \geq SL, \forall d \quad (2.5)$$

$$Pr\left\{x_d^i \cdot \left(1 - S_{i,d}^f(\omega)\right) + y_d^i \cdot \left(1 - S_{i,d}^t(\omega)\right) \geq RTO_{i,d} * TOL, \forall i\right\} \geq SLT, \forall d \tag{2.6}$$

$$\left(w_k^f + Tw_k^f\right) \cdot AR \cdot \frac{4}{3} + LL \cdot \frac{4}{3} \geq \frac{w_k^t}{RTC} \tag{2.7}$$

$$\sum_{d \in Q_{k,m}} \sum_{i \in I} y_d^i \leq 3 \times w_k^t, \quad \forall\, m \tag{2.8}$$

$$\sum_{d \in D} \sum_{i \in I} x_d^i \leq w_k^f \tag{2.9}$$

$$\sum_{d \in D} \sum_{i \in I} y_d^i \leq w_k^t \tag{2.10}$$

$$x_d,\ y_d\ \in\ R_+ \tag{2.11}$$

2.3 Case study of the long-term planning model

A manufacturing assembly plant provides the past several years' absenteeism. The absenteeism varies in month which is related with weather and vacation, and it also varies in department and shift. In the following Figure 2.1, we present 4 density plots of legacy workers' daily absenteeism rate of 2 selected departments and 2 different shifts in the same month. Absenteeism rate is calculated by the number of absent workers in the department at the shift divided by the total number of workers supposed to show up there. The red lines are the Beta distribution fits of the data.

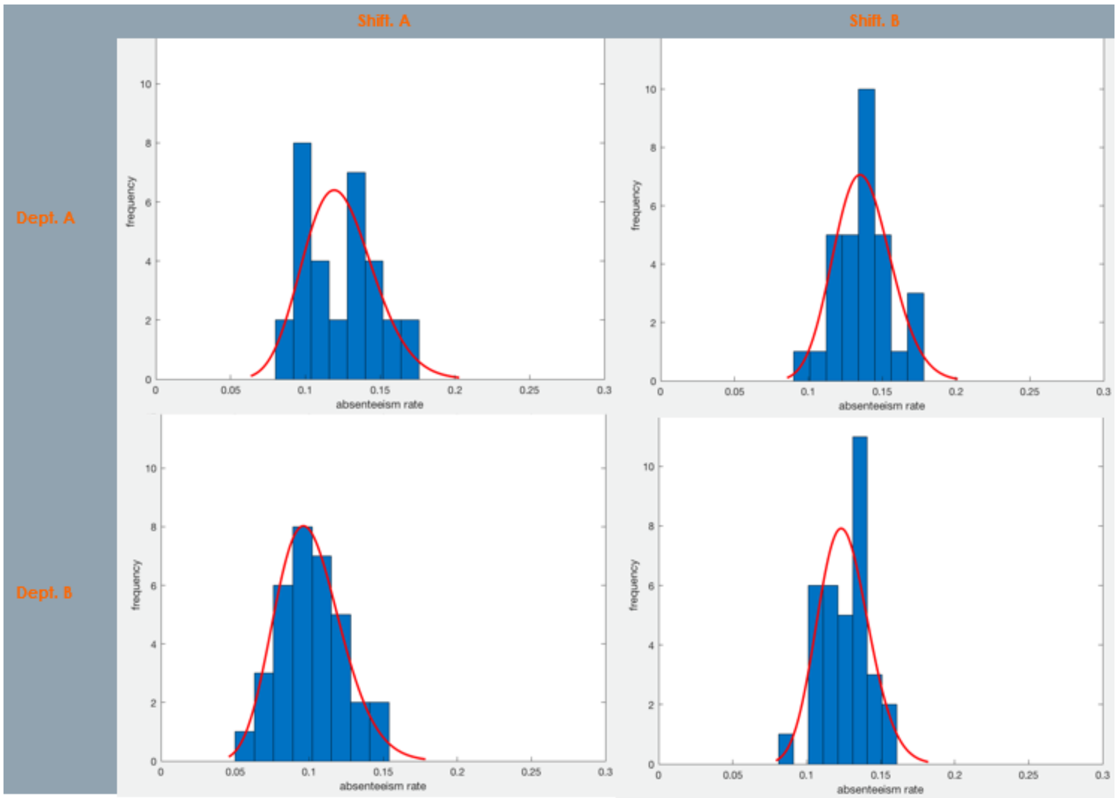

Figure 2.1: Density plots of absenteeism rate

As we can see, all four density plots are different from each other. We can easily tell the absenteeism rate of department B is lower than department A. Then we check the wellness of these 4 different beta distributions to predict the rest of data points we didn't use in fitting in the following Q-Q plot.

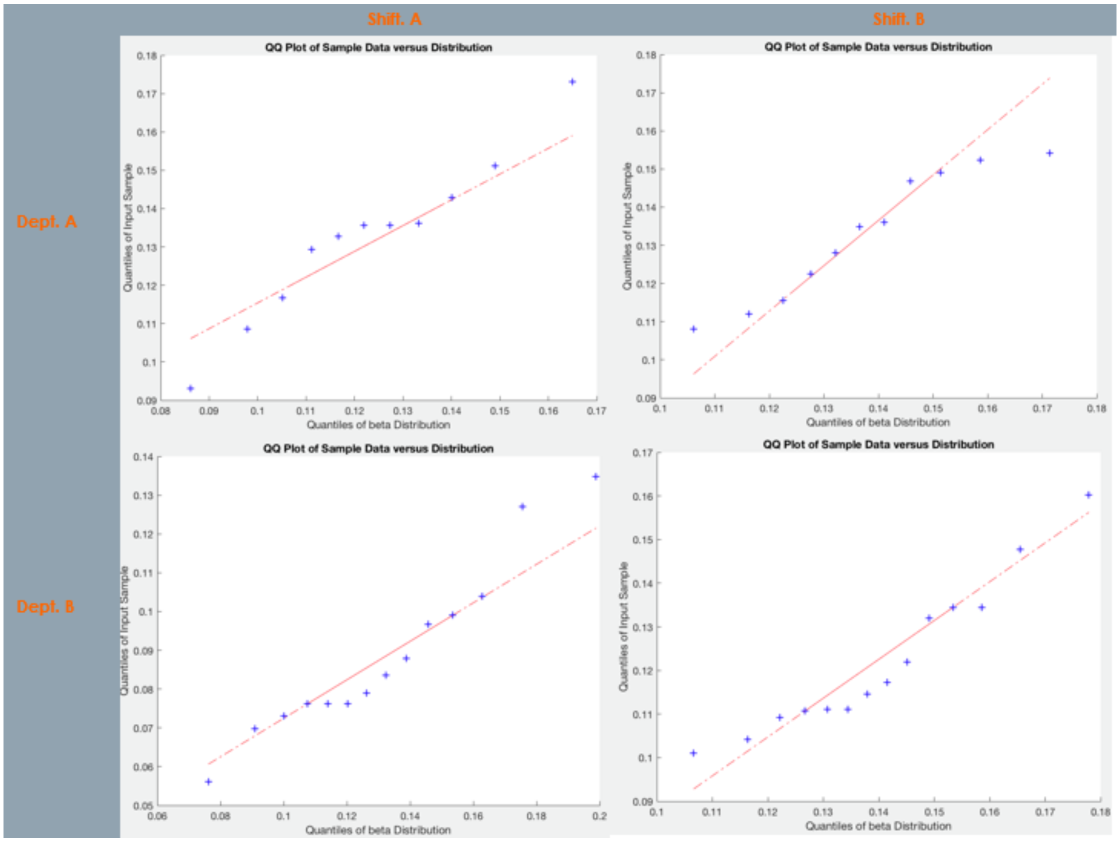

Figure 2.2: Q-Q plots of beta distributions to the rest sample points

Blue dots are very close the central red lines, which means all the distributions can predict very well. And we can use those beta distributions to represent the absenteeism rate of legacy workers.

As for the temporary workers, the absenteeism rate doesn't seem to follow any beta distribution because of two main reasons. The first reason is that temporary workers have less vacation and FMLA, so we find their absenteeism rate is much lower than the legacy workers' and sometimes even goes to 0. The second reason is the number of temporary workers in each department is under a hundred which makes the rate more discrete. Thus we have two options of distributions for temporary workers' absenteeism rate.

Option 1 is a mixed exponential distribution, where we give a mass probability to absenteeism rate equal to 0 and an exponential distribution to rate greater than 0.

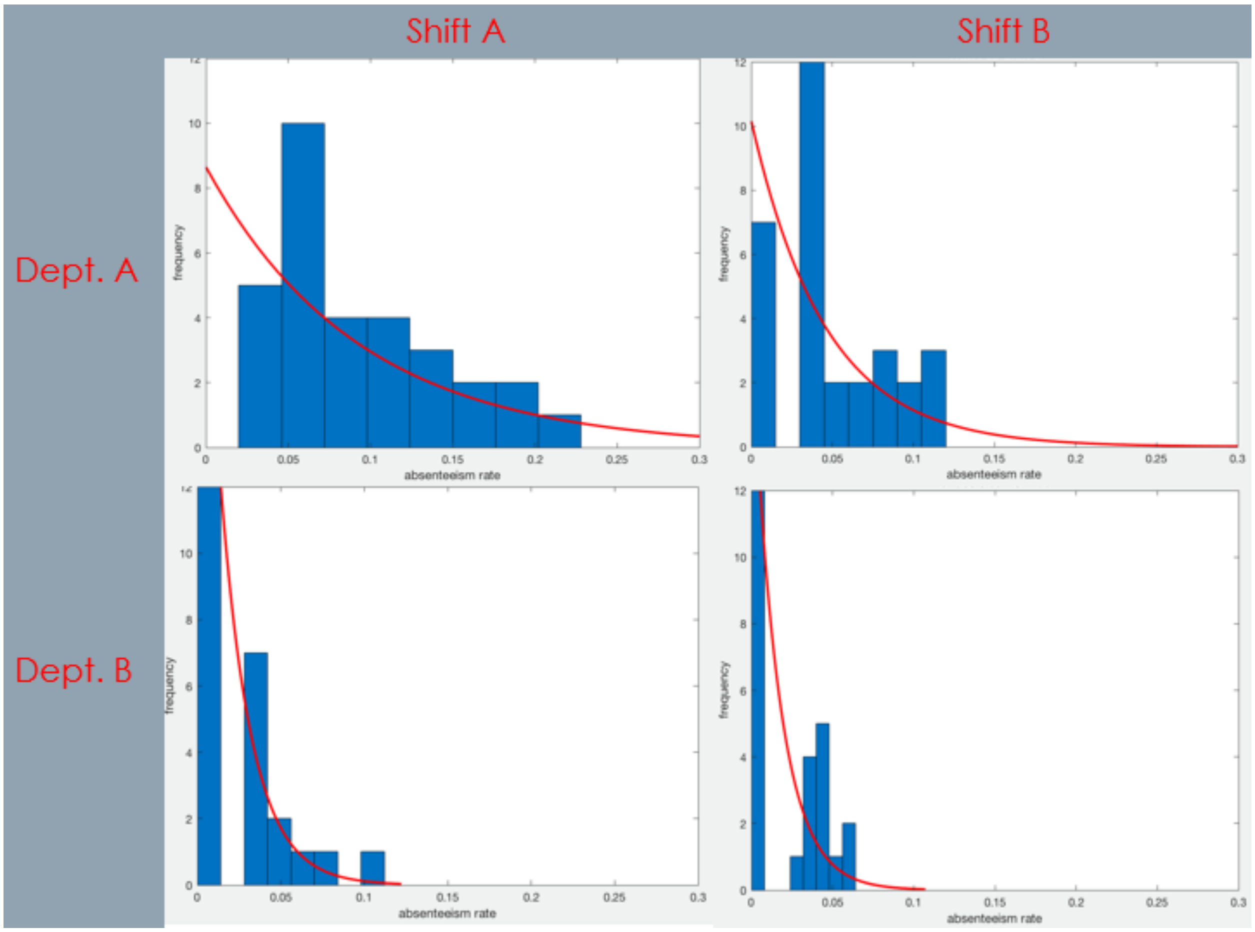

Figure 2.3: Mixed exponential distribution of temporary workers' abs. rate

Option 2 is a beta-binomial distribution. We assume the prior distribution of the probability of whether an employee will come or not to follow a beta distribution, and the number of absent worker will follow the binomial distribution with this beta-distributed probability.

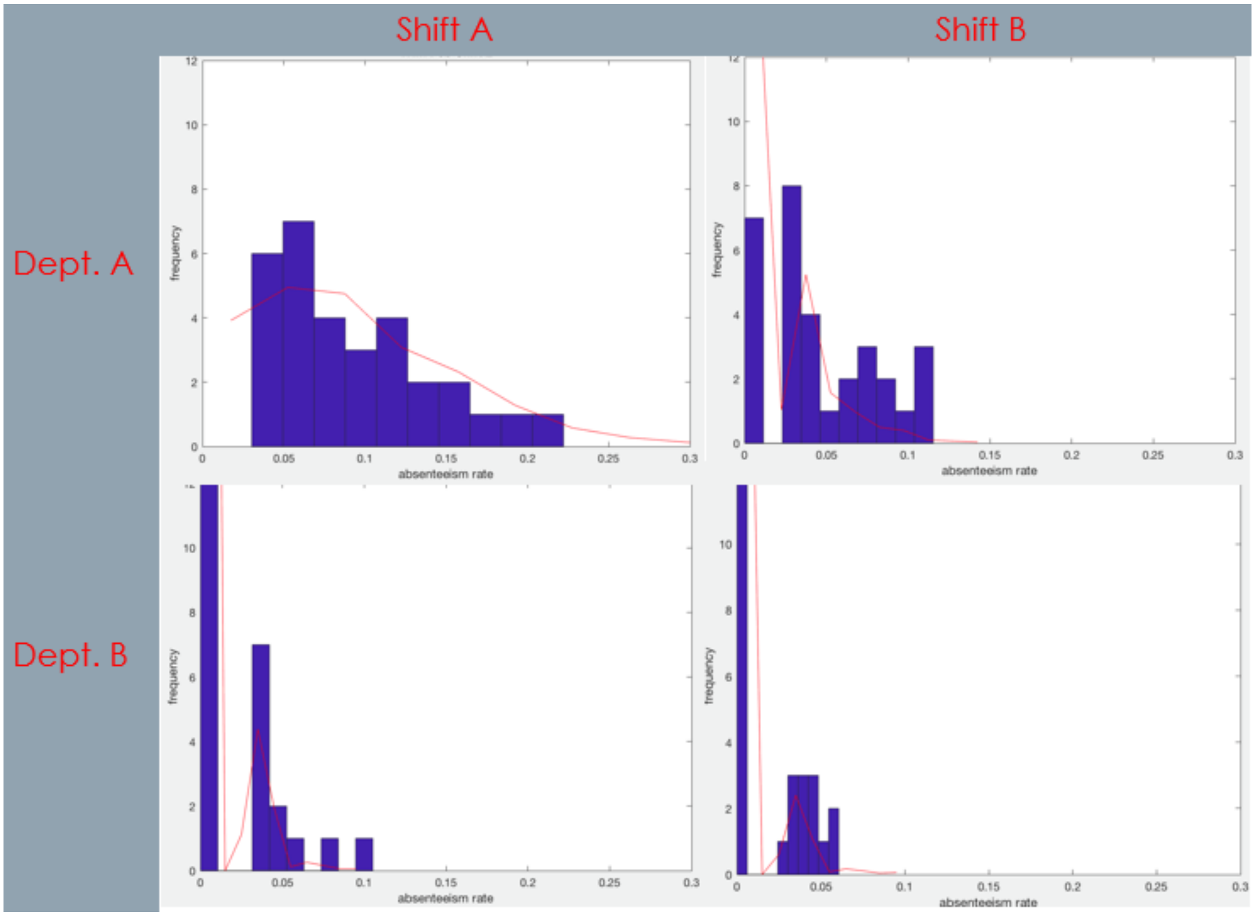

Figure 2.4: beta-binomial distribution of temporary workers’ abs. rate

We can find the mixed exponential distribution approximates the envelop of temporary workers’ absenteeism rate which is assumed to be continuous, while the beta-binomial distribution better fit the number of daily absentee temporary workers. However, we still choose option 1 to finish the model at current stage because option 2 extremely increases the complexity of solution algorithm.

We use the cost and the stress level requirement (probability to satisfy the required number) provided by the manufacturing assembly plant, and Monte Carlo simulation to solve the stochastic model in last section. In the following figure, we present the performance of our staff level compared to their current staff level of two shift and two department for two months next to each other.

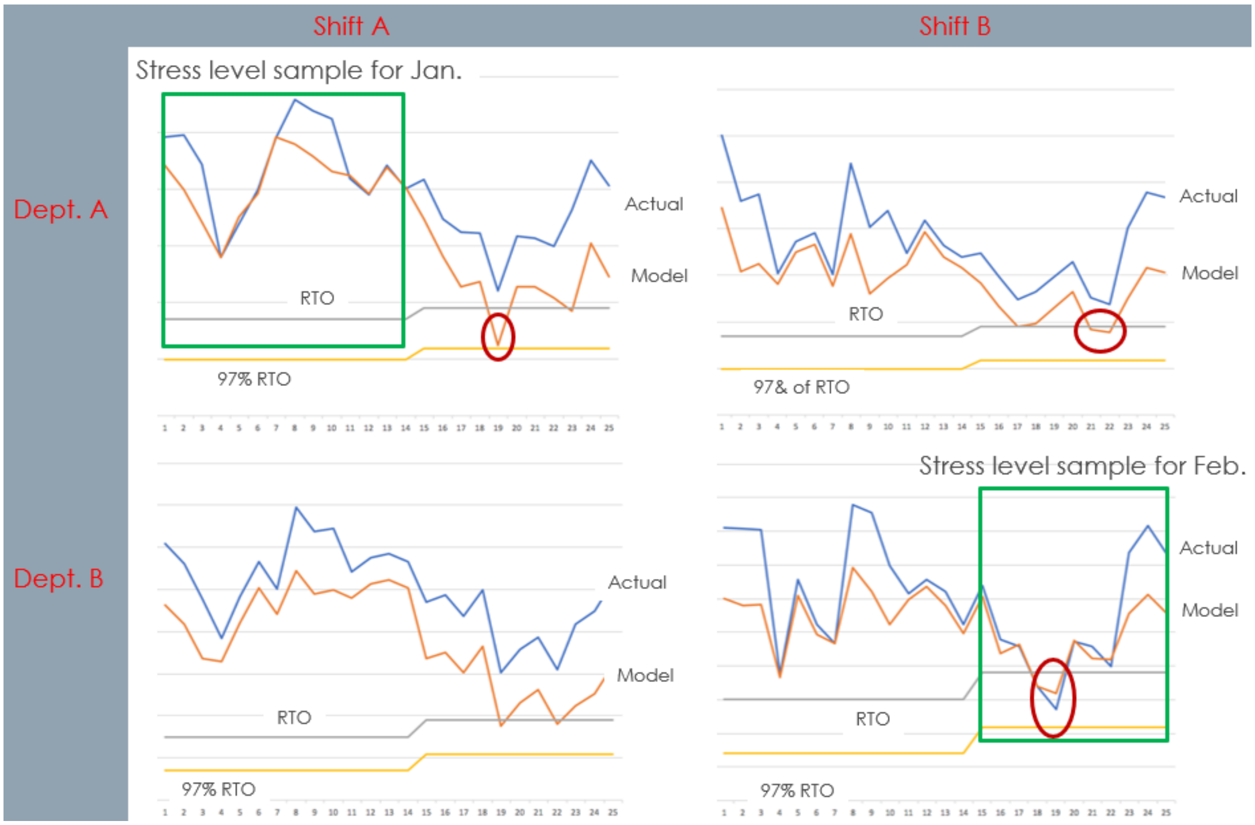

Figure 2.5: Model performance using real abs. rate in Jan. and Feb.

In Figure 2.5, x axis represents the real working day in Jan. and Feb. in this plant, while y axis is the number of workers. RTO differs by department, shift and month, and the stress level differs only by month for productivity purpose. Comparing the Model line and the Actual line which are show-up worker number calculated by the on-roll number of workers multiple by one minus the real daily absenteeism rate, we find the model staff level performance as good as the actual staff level. By balancing the temporary worker and legacy worker, our staff plan saves the plant 2.1% cost in these two months just for these two department and two shifts. That’s expected to be millions of dollars saving when expanded to the whole plant for a year.

Chapter 3: Assignment Models

This chapter is organized as follows. Section 1 describes several different scenarios where request different assigning or adjusting logic. Section 2 introduces the related LP or MIP models of scenarios in section 1. Section 3 gives an example of basic assignment model.

3.1 Introduction

With the long-term planning model in Section 2.2, we can get an answer to the question "how many workers that the plants need". However, even in the same department, not every worker is trained to do every job. In order to getting the "right people" not only "right number of people", we need to develop the assignment model to choose the best combination. After several hours' operation, the plants always find some workers leave in the middle of the day for some reason while others who were absent at the beginning may come later. If the changes are significant to make some adjustment, but the team leaders don't want to shuffle everyone by rerunning the assignment model, then an adjustment model will be needed based on current assignment as well as the updated attendance.

In the daily operation, each department and even each DROT (working team) may have different concerns of assigning which requires different models. We present several models we have already made in the following section.

3.2 Assignment models

$V_{i,j}$ is the skill matrix that has been introduced in Chapter 1. Here we treat $V_{i,j}$ as a binary value for either worker i can finish job j by a single person or cannot. ω_i is the original daily attendance table, while $\overline{\omega}_i$ is the updated one for adjustment model. A is the maximum number of moves that the adjustment model is allowed to make. $M_{i,j}$ is the preference table where the value is smaller for higher priority combination, and huge or infinite value when $V_{i,j}$=0. $y_{i,j}$ is the decision variable that worker i is assigned to job j in the assignment model, and also an input parameter in the adjustment model as the current assignment needing some change. $x_{i,j}$ is the decision variable of updated assignment after adjustment.

Table 3.1: Parameters Definition

Parameter	Definition
$V_{i,j}$	versatitlity matrix. 1 if worker i is able to do job j, 0 if not able.
ω_i	attendance matrix. 1 if worker i shows up, 0 if absent.
$M_{i,j}$	preference table. smaller value for higher priority; huge value when $V_{i,j}$=0.
$\overline{\omega}_i$	updated attendance matrix. 1 if worker i shows up
A	maximum number of moves that the adjustment model is allowed

Decision variables:

$y_{i,j}$	current assignment matrix. 1 if worker i actually works at job j
$x_{i,j}$	updated assignment matrix. 1 if worker i actually works at job j

In the assignment mode, the objective function is to maximize the total number of single-occupied jobs which means the least labor cost and highest efficiency of using people. (3.2) represents the constraint that one worker can fully occupied one job if trained. (3.3) means that the team leader can only make assignment to those who attend to work. (3.4) means one job only need one trained worker. (3.5) is the non-negative constraint which is supposed to be a non-negative integer. However, $y_{i,j}$ can be proved to be either 0 or 1 even if we don't force it to be a binary or integer variable in the model.

Assignment Model:

$$Max \sum_{i \in I} \sum_{j \in J} y_{i,j} \tag{3.1}$$

$$s.t. \qquad y_{i,j} \leq V_{i,j}, \forall\, i, \forall\, j \tag{3.2}$$

$$\sum_{j \in J} y_{i,j} \leq \omega_i\,, \forall\, i \tag{3.3}$$

$$\sum_{i \in I} y_{i,j} \leq 1, \forall\, j \tag{3.4}$$

$$y_{i,j} \geq 0, \forall\, i, \forall\, j \tag{3.5}$$

Once some adjustments are needed, we can input the $y_{i,j}$ and the updated attendance table to the following adjustment model. (3.6) to (3.9) are the same constraints as (3.1) to (3.4). (3.10) is the movement allowance constraint that the new assignment cannot be too different from the current assignment. As you can see, this constraint has a plus function which is non-linear for better comprehensive. We will linearize this constraint after the model.

Adjustment Model:

$$Max \sum_{i \in I} \sum_{j \in J} x_{i,j} \tag{3.6}$$

$s.t.$

$$s_{i,j} \leq V_{i,j}, \forall\, i, \forall\, j \tag{3.7}$$

$$\sum_{j \in J} y_{i,j} \leq \overline{\omega}_i\,, \forall\, i \tag{3.8}$$

$$\sum_{i \in I} x_{i,j} \leq 1, \forall\, j \tag{3.9}$$

$$\sum_{i \in I} \sum_{j \in J} \left(x_{i,j} - y_{i,j}\right)_+ \leq A \tag{3.10}$$

$$x_{i,j} \geq 0, \forall\, i, \forall\, j \tag{3.11}$$

We introduce a new binary decision variable $z_{i,j}$ to linearize (3.10) by replacing it with (3.12) and (3.13). Though we didn't request $x_{i,j}$ and $y_{i,j}$ to be binary, they actually can only be 0 or 1. If $x_{i,j}$ is greater than $y_{i,j}$, $z_{i,j}$ equals to 1; otherwise $z_{i,j}$ equals to 0 which means not counted as movement.

$$x_{i,j} - y_{i,j} \leq z_{i,j}\,, \forall\, i, \forall\, j \tag{3.12}$$

$$\sum_{i \in I} \sum_{j \in J} z_{i,j} \leq A \tag{3.13}$$

The rules of count the movement are as followed. Job A and Job B represent different jobs, absent means the worker is absent at that time, and Double-up means the workers attend to work but without assigning to any job all by themselves.

Table 3.2: Movement counting rules

Previous assignment	**Current assignment**	**Count on moves**
Job A	Job A	0
Job A	Job B	1
Job A	Absent	0
Absent	Job A	1
Absent	Double-up	0
Double-up	Job A	1
Double-up	Double-up	0
Double-up	Absent	0

Besides the traditional assignment and adjustment models, another model will take the preference into consideration which is called priority-based assignment model.

This model requires to run the assignment model first, and record the optimal value of (3.1) as K. And it's used in (3.18) to keep the preferred assignment has the same efficiency as the optimal one which reaches the maximum number of single-occupied jobs. By minimizing the objective value in (3.14), the final output of $y_{i,j}$ will be the most preferred assignment among those that gives the highest efficiency.

Priority − based Assignment Model:

$$Min \sum_{i \in I} \sum_{j \in J} M_{i,j} \times y_{i,j} \tag{3.14}$$

$s.t.$

$$y_{i,j} \leq V_{i,j}, \forall\, i, \forall\, j \tag{3.15}$$

$$\sum_{j \in J} y_{i,j} \leq \omega_i\ , \forall\, i \tag{3.16}$$

$$\sum_{i \in I} y_{i,j} \leq 1, \forall\, j \tag{3.17}$$

$$\sum_{i \in I} \sum_{j \in J} y_{i,j} = K \tag{3.18}$$

$$y_{i,j}\ \geq 0, \forall\, i, \forall\, j \tag{3.19}$$

From the opposite direction, some DROTs are non-rotate in assigning, which means their first goal is to keep their original assignment and then achieve the maximum number of single-assigned jobs for the rest of workers and jobs. We exchange the order of the priority based assignment model above by first maximizing the priority, and then run the assignment model with the additional constraint ensuring the assignment can reach the maximized priority.

3.3 Case study of assignment model

Here is an example of the basic assignment model and some result. We can first check the original skill matrix in the following table (workers' name and jobs' title have been hidden for privacy purpose).

Table 3.3: Original skill matrix

employee no.	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
E1	4	4	4	4	4	4	4	4	4	4	4	4	2	4
E2	4	4	0	0	0	0	4	4	4	4	4	0	0	0
E3	4	4	4	4	4	4	4	4	0	0	0	4	0	4
E4	4	4	4	4	4	4	4	4	4	4	4	4	0	4
E5	0	0	0	4	0	0	0	0	0	0	0	0	0	0
E6	0	0	0	0	0	4	0	0	0	0	0	0	0	0
E7	0	1	2	4	0	0	0	0	0	1	0	0	0	0
E8	0	0	0	0	0	0	0	4	4	0	0	0	0	0
E9	0	0	0	0	0	0	0	0	0	4	0	0	0	0
E10	0	0	0	0	4	0	0	0	0	0	0	0	0	0
E11	0	3	0	0	0	0	0	4	4	0	1	0	0	0
E12*	0	3	0	0	0	1	0	0	0	0	1	0	0	0
E13	0	0	0	0	0	0	0	0	0	0	0	1	0	0
E14	0	0	4	0	0	0	0	0	0	0	0	0	0	0
E15	0	0	0	0	0	0	0	0	0	4	4	0	4	0
E16	4	4	4	4	4	4	4	4	4	4	4	4	4	4

Each row is a skill set for a worker, and each column represent one job. The 0-4 levels are the training level for each worker at each job. The worker has never been trained for the job when the training level is 0; under training process when level 1 or 2 and well-trained for level 3 and 4. In another word, one worker can individually finish the task of one job only if there is 3 boxes or 4. Based on this rule, we convert Table 3.3 to a binary table of whether the worker can be single-assigned or not in Table 3.4.

Table 3.4: Converted skill matrix

employee no.	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
E1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
E2	1	1	0	0	0	0	1	1	1	1	1	0	0	0
E3	1	1	1	1	1	1	1	1	0	0	0	1	0	1
E4	1	1	1	1	1	1	1	1	1	1	1	1	0	1
E5	0	0	0	1	0	0	0	0	0	0	0	0	0	0
E6	0	0	0	0	0	1	0	0	0	0	0	0	0	0
E7	0	0	0	1	0	0	0	0	0	0	0	0	0	0
E8	0	0	0	0	0	0	0	1	1	0	0	0	0	0
E9	0	0	0	0	0	0	0	0	0	1	0	0	0	0
E10	0	0	0	0	1	0	0	0	0	0	0	0	0	0
E11	0	1	0	0	0	0	0	1	1	0	0	0	0	0
E12*	0	1	0	0	0	0	0	0	0	0	0	0	0	0
E13	0	0	0	0	0	0	0	0	0	0	0	1	0	0
E14	0	0	1	0	0	0	0	0	0	0	0	0	0	0
E15	0	0	0	0	0	0	0	0	0	1	1	0	1	0
E16	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Those specially colored columns and rows are critical jobs and temporary workers which could have different weight to the optimal solution. In this example, we will only make use of the converted binary parameters and a randomly picked t to show how assignment works. The output of assignment model is in Table 3.5.

Table 3.5: Output of assignment model

	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
E1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
E2	0	0	0	0	0	0	1	0	0	0	0	0	0	0
E3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E4	0	0	0	0	0	0	0	0	0	0	1	0	0	0
E5	0	0	0	1	0	0	0	0	0	0	0	0	0	0
E6	0	0	0	0	0	1	0	0	0	0	0	0	0	0
E7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E8	0	0	0	0	0	0	0	1	0	0	0	0	0	0
E9	0	0	0	0	0	0	0	0	0	1	0	0	0	0
E10	0	0	0	0	1	0	0	0	0	0	0	0	0	0
E11	0	0	0	0	0	0	0	0	1	0	0	0	0	0
E12*	0	1	0	0	0	0	0	0	0	0	0	0	0	0
E13	0	0	0	0	0	0	0	0	0	0	0	1	0	0
E14	0	0	1	0	0	0	0	0	0	0	0	0	0	0
E15	0	0	0	0	0	0	0	0	0	0	0	0	1	0
E16	0	0	0	0	0	0	0	0	0	0	0	0	0	1

All cells with value 1 have been colored in red, which means we assign that worker in the row to the job in column. For example, the red 1 in the first column means the assignment model suggests to ask 1st worker to do 1st job. From the vertical direction, we can find all jobs have been assigned one worker to do.

After the production line runs for a while, E1 may leave the job because of sickness or some other personal affairs. The team leader definitely doesn't want to reshuffle the whole line, so the adjustment model will suggest the following new assignment table for the least movement of workers who has already been assigned to some job.

Table 3.6: Output of adjustment model

	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
E1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E2	0	0	0	0	0	0	1	0	0	0	0	0	0	0
E3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
E4	0	0	0	0	0	0	0	0	0	0	1	0	0	0
E5	0	0	0	1	0	0	0	0	0	0	0	0	0	0
E6	0	0	0	0	0	1	0	0	0	0	0	0	0	0
E7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E8	0	0	0	0	0	0	0	1	0	0	0	0	0	0
E9	0	0	0	0	0	0	0	0	0	1	0	0	0	0
E10	0	0	0	0	1	0	0	0	0	0	0	0	0	0
E11	0	0	0	0	0	0	0	0	1	0	0	0	0	0
E12*	0	1	0	0	0	0	0	0	0	0	0	0	0	0
E13	0	0	0	0	0	0	0	0	0	0	0	1	0	0
E14	0	0	1	0	0	0	0	0	0	0	0	0	0	0
E15	0	0	0	0	0	0	0	0	0	0	0	0	1	0
E16	0	0	0	0	0	0	0	0	0	0	0	0	0	1

We can find that the only change is moving E3 to cover W1, and all other assignments remain the same after E1 leaves.

When the team leader has some preference, for example E4 to do W3 and E5 to do W4, the priority-based assignment model can give the result of optimal assignment regarding the preference.

Table 3.7: Output of priority-based model

	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
E1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
E2	0	0	0	0	0	0	1	0	0	0	0	0	0	0
E3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
E4	0	0	1	0	0	0	0	0	0	0	0	0	0	0
E5	0	0	0	1	0	0	0	0	0	0	0	0	0	0
E6	0	0	0	0	0	1	0	0	0	0	0	0	0	0
E7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E8	0	0	0	0	0	0	0	1	0	0	0	0	0	0
E9	0	0	0	0	0	0	0	0	0	1	0	0	0	0
E10	0	0	0	0	1	0	0	0	0	0	0	0	0	0
E11	0	0	0	0	0	0	0	0	1	0	0	0	0	0
E12*	0	1	0	0	0	0	0	0	0	0	0	0	0	0
E13	0	0	0	0	0	0	0	0	0	0	0	1	0	0
E14	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E15	0	0	0	0	0	0	0	0	0	0	0	0	1	0
E16	0	0	0	0	0	0	0	0	0	0	0	0	0	1

W3 and W4 have the priority operators E4 and E5, which makes the solution slightly different from basic assignment but still keep the same value of object function.

Chapter 4: Cross-training Model

This chapter is organized as follows. Section 1 describes the motivation of cross-training. Section 2 introduces the two-stage stochastic programming cross-training model. Section 3 gives a case study of this model.

4.1 Introduction

In an ideal situation, the plant will fully train every worker to every job if there's no training cost. Transparently, the training takes time and have some effects on the productivity, so we are going to build the following cross-training model to best improve the efficiency of assignment based on current limitation of training.

4.2 Cross-training Model

$V_{i,j}$ is the current skill matrix which indicates whether worker i is able to do job j. $W(\omega)$ is the stochastic attendance table of random sample ω. The value will equal to 1 if the worker i show up to work in this scenario, 0 if absent. B is the cross-training limitation where we assume the plants only have a fixed number of opportunity to train workers to jobs for B times. If a worker is trained for two different jobs, it counts 2 at B; and if two different workers are trained to do one job, it also counts 2. $y_{i,j}(\omega)$ is the decision variables of assignment table for each random sample. And $u_{i,j}$ is the decision variables of updated skill matrix after cross-training.

Table 4.1: Parameters Definition

Parameter	Definition
$V_{i,j}$	current versatitlity matrix. 1 if worker i is able to do job j, 0 if not able.
$W(\omega)$	stochastic attendance matrix. 1 if worker i shows up, 0 if absent.
B	maximum number of training (1 worker to 1 job count 1)

Decision variables:

$y_{i,j}(\omega)$	assignment matrix based on the attendance. 1 if worker i actually works at job j
$u_{i,j}$	updated assignment matrix. 1 if worker i actually works at job j

The objective of cross-training model is to maximize the expectation of jobs could be optimal single-assigned. (4.2) is the cross-training limitation that the plants only have a fixed number of opportunity to train workers. (4.3) means workers can only gain new skills but never forget what they have been trained. (4.6) represents the constraint that one worker can fully occupied one job if trained. However, the skill matrix is no more the original one, we use the updated skill matrix $u_{i,j}$ here to show the improvement after the cross-training. (4.7) is the attendance constraint applied in every scenario ω. (4.8) means only one trained worker is needed for each job. (4.4) and (4.9) are the binary constraints.

stage 1 (**training**):

$$Max\ E\left(Max \sum_{i\in I}\sum_{j\in J} y_{i,j}(\omega)\right) \tag{4.1}$$

$$s.t. \quad \sum_{i\in I}\sum_{j\in J}(u_{i,j} - V_{i,j}) \leq B \tag{4.2}$$

$$u_{i,j} \geq V_{i,j}, \forall\, i, \forall\, j \tag{4.3}$$

$$u_{i,j}\ is\ binary, \forall\, i, \forall\, j \tag{4.4}$$

stage 2 (**assigning**):

$$Max \sum_{i \in I} \sum_{j \in J} y_{i,j}(\omega) \tag{4.5}$$

$s.t.$

$$y_{i,j}(\omega) \leq u_{i,j}, \forall\, i, \forall\, j \tag{4.6}$$

$$\sum_{j \in J} y_{i,j}(\omega) \leq W(\omega)\ , \forall\, i \tag{4.7}$$

$$\sum_{i \in I} y_{i,j}(\omega) \leq 1, \forall\, j \tag{4.8}$$

$$y_{i,j}(\omega)\ are\ binary, \forall\, i, \forall\, j \tag{4.9}$$

4.3 Case Study of cross-training model

We selected a very stressful line which contains 3 DROTs that can cross-train their workers to each other's job. The current skill matrix is in the following table.

Table 4.2: Current skill matrix

	J01	J02	J03	J04	J05	J06	J07	J08	J09	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19	J20	J21	J22	J23	J24	J25	J26	J27	J28	J29	J30	J31	J32	J33	J34	J35	J36	J37	J38	J39	J40	J41	J42
E1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E4	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E5	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E6	0	1	0	0	0	1	0	0	0	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	1	0	1	1	1	1	0	0	1	1	0	0	0	0	0	0
E7	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E8	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E9	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E10	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E11	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E12	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E13	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E14	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E15	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E16	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E17	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E18	0	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E20	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E24	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E25	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E27	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E31	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0
E42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
E43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
E44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0
E45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
E46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0
E47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0
E48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1	1
E49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
E50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
E51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
E52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
E53	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1
E54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0
E55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
E56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0

In the rows are employees' number, while the jobs are in each column. 0 and 1 mean able to do or not trained to do. With the historical data of absenteeism in this line, we have the following training plan and the performance based on 100,000 times simulation.

Table 4.3: Training plan and performance

# of training	# of short/short days	# of short/total days
0 (original)	3.16	2.72

1 (E40 to J28)	3.01	2.43
2 (E27 to J08)	2.96	2.29
3 (E13 to J09)	2.90	2.17
4 (E17 to J26)	2.89	2.06
5 (E2 to J17)	2.85	1.97
All (fully trained)	2.79	1.71

We can find obvious reduce of the shortage number as more workers are cross-trained. Once we are provided with the cross-training cost or shortage cost, we can help decide the possible budget to cross train the workers in the future research.

Chapter 5: System Architecture

In this chapter, the structure of workforce operation and management system will be introduced. All related models are coded in several programming languages for different purpose of application. The usage of basic functions will be explained in details to help further modify the models.

5.1 User's interface

A user's interface has been developed in QlikView to help the plants upload their daily operational data to the database, and make adjustment to the suggested assignment. We can see the current assignment and the jobs that not been occupied in the selected group, either at area level, department level or assembly line level. Attendance information will be automatically read in through the badge-in system.

By selecting the job in need to be modified, the user could be routed to the othrt page in. The user will be able to check all the workers are qualified to this job as well as all the free-worker that have not been assigned to any job. For those who are not trained to the job, a double-up policy allows two of them to fill in one position. Selection will be uploaded to the clouding sever to updating the lists so that users in all devices can see the same information.

5.2 Clouding server

The clouding server is built with the three types of model introduced in Chapter 2, 3 and 4. The assignment model has been coded in R with 3 major functions, while long-term planning model and cross-training model are still programmed in both MATLAB and OPL (CPLEX studio).

When the main entrance received the data and request from any user's end, it will pass the data to the online database and also read data from it based user's request through data process module. After solving the problem in a certain model, the output will be sent to both database and user.

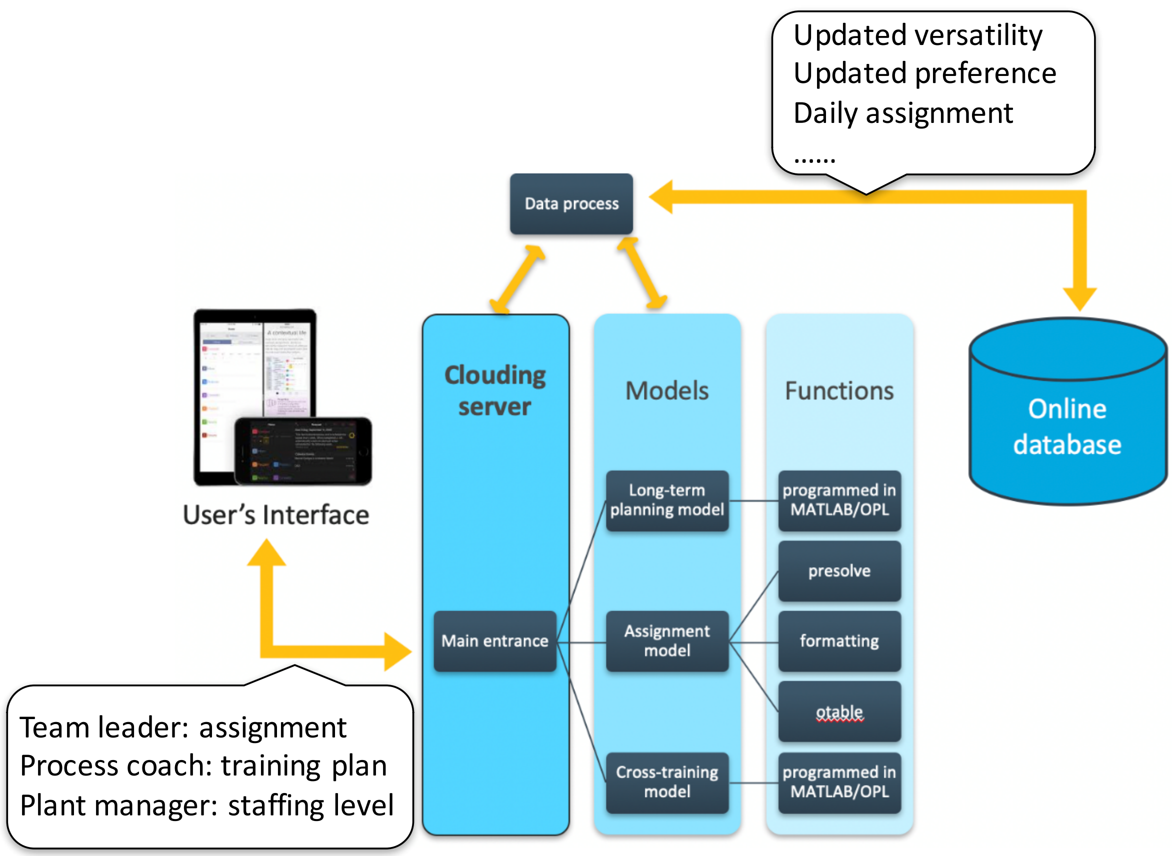

Figure 5.1: System design

5.3 Functions in R

The workforce operation and management system will be eventually based on R. R is a language and environment for statistical computing and graphics which is available as free software. We currently provide the following basic functions to help program the assignment-related models.

Table 5.1: Input/output of R functions

Function	Input	Output
presolve	Skill matrix Attendance table Preference table (optional) Borrow table (optional)	All possible combination of assignment (one trained worker is assigned to one job)
formatting	All possible combination of assignment from function presolve	Standard tables for optimization function in R
otable	Output of optimization function in R Skill matrix Attendance table Preference table (optional) Borrow table (optional)	Assignment table Updated skill matrix

Chapter 6: Conclusion and Discussion

In this project, a workforce operation and management system has been designed to help the plant make use of their employees more efficiently. We develop linear programming assignment and adjustment models which allocate show-up employees to appropriate jobs considering their individual skills and preferences, whereas the adjustment model optimally and dynamically adjust job assignment on account of employees' coming late or leaving early. A two-stage stochastic cross-training optimization model is proposed to select trainees and decide which jobs they should be trained for. We incorporate the models developed in the two types of models in a dynamic optimization model for decide staffing levels during a long-time horizon with the help of the prediction of absenteeism rate.

We're still working on enriching the constrains to make models closer to reality, and combine models together to find global optimal values. Standardized programming will also be applied in the operation and management system when the models are clear.

References

[1] Easton, F. F. (2014). Service completion estimates for cross□trained workforce schedules under uncertain attendance and demand. Production and Operations Management, 23(4), 660-675.

[2] Wang, J. (2005). A review of operations research applications in workforce planning and potential modeling of military training (No. DSTO-TR-1688). DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION SALISBURY (AUSTRALIA) SYSTEMS SCIENCES LAB.

[3] Bureau of Labor Statistics. (2015). Labor force statistics from the current population survey.

[4] Stolletz, R. (2010). Operational workforce planning for counters airports. Transportation Research Part E: Logistics and Transportation Review, 46(3), 414-425.

[5] Bordoloi S, Matsuo H (2001), 'Human Resource Planning inKnowledge-intensive Operations: A Model for Learning With Stochastic Turnover', European Journal of Operational Research, Vol. 130, pp. 169-189.

[6] Castley R (1996), 'Policy-focused Approach to Manpower Planning', International Journal of Manpower, Vol. 17 (3), pp. 15-24.

[7] Bard, J. F., Morton, D. P., & Wang, Y. M. (2007). Workforce planning at USPS mail processing and distribution centers using stochastic optimization. Annals of Operations Research, 155(1), 51-78.

[8] Gans, N., Shen, H., Zhou, Y. P., Korolev, N., McCord, A., & Ristock, H. (2015). Parametric forecasting and stochastic programming models for call-center workforce scheduling. Manufacturing & Service Operations Management, 17(4), 571-588.

[9] Leung, S. C., Wu, Y., & Lai, K. K. (2006). A stochastic programming approach for multi-site aggregate production planning. Journal of the Operational Research Society, 57(2), 123-132.

[10] Punnakitikashem, P., Rosenberger, J. M., & Behan, D. B. (2008). Stochastic programming for nurse assignment. Computational Optimization and Applications, 40(3), 321-34

Appendix

- **R code:**

```
presolve <- function(ver,pre,att,bor) {
  njob <- ncol(ver)
  nworker <- nrow(ver)
  if (nrow(att) != nworker) stop("attendance table and versatility matrix have different
size")
  if (nrow(pre) != nworker) stop("preference table and versatility matrix have different
size")
  names(pre)[2] <- "job"
  names(ver)[1] <- "worker"
  names(att)[1] <- "worker"
  pre$job <- gsub("#", ".", pre$job)
  pre$job <- gsub("-", ".", pre$job)
  pre$job <- gsub(" ", ".", pre$job)
  bor$job <- gsub("#", ".", bor$job)
  bor$job <- gsub("-", ".", bor$job)
  bor$job <- gsub(" ", ".", bor$job)
  temp0 <- rbind(pre,bor)
  temp1 <- filter(temp0, job != 0)
  countpre <- unique(temp1$job)
  if (length(countpre) != nrow(temp1)) stop ("there're duplicated prefer jobs in preference
or borrow")
  # pre-solve
  v0 <- merge(ver,att,by="worker")
  v0 <- filter(v0,attendance == 1)
  v0 <- subset(v0, select = -c(attendance) )
  v <- melt(v0,id="worker")
  names(v)[2] <- "job"
  names(v)[3] <- "value"
  potvar <- filter(v,value == 1)
  temp2 <-merge(x = potvar, y = temp1, by = "worker", all.x = TRUE)
  temp2 <- filter(temp2, is.na(job.y))
  names(temp2)[2] <- "job"
  temp3 <-merge(x = temp2, y = temp1, by = "job", all.x = TRUE)
  temp3 <- filter(temp3, is.na(worker.y))
  names(temp3)[2] <- "worker"
  potvar <- subset(temp3, select = c(worker,job,value))
  potvar$varnum <- paste("var",row(potvar[1]))
  return(potvar)
  }

  # formatting
  formatting <- function(potvar) {
  const.mat1 <- acast(potvar,varnum ~ job)
  const.mat2 <- acast(potvar,varnum ~ worker)
  const.mat <- cbind(const.mat1,const.mat2)
  const.mat[is.na(const.mat)] <- 0
```

```
const.mat <- t(const.mat)
  nvar <- ncol(const.mat)
  nconst <- nrow(const.mat)
  obj <- rep(1,nvar)
  const.dir <- rep("<=",nconst)
  const.rhs <- rep(1,nconst)
  optinput <-
list("obj"=obj,"const.mat"=const.mat,"const.dir"=const.dir,"const.rhs"=const.rhs)
return(optinput)
  }
  # Solve Optimization Problem

  #optimalassign <- lp(direction="max" , obj, const.mat, const.dir, const.rhs)

  ## Outputs ---

  # assignment table
  otable <- function(optimalassign,optinput,potvar,ver, pre, att, bor){
  njob <- ncol(ver)
  nworker <- nrow(ver)
  if (nrow(att) != nworker) stop("attendance table and versatility matrix have different
size")
  if (nrow(pre) != nworker) stop("preference table and versatility matrix have different
size")
  names(pre)[2] <- "job"
  names(ver)[1] <- "worker"
  names(att)[1] <- "worker"
  pre$job <- gsub("#", ".", pre$job)
  pre$job <- gsub("-", ".", pre$job)
  pre$job <- gsub(" ", ".", pre$job)
  bor$job <- gsub("#", ".", bor$job)
  bor$job <- gsub("-", ".", bor$job)
  bor$job <- gsub(" ", ".", bor$job)
  temp0 <- rbind(pre,bor)
  temp1 <- filter(temp0, job != 0)
  optinput$const.mat[1,] <- optimalassign$solution
  assignment <- head(optinput$const.mat,1)
  row.names(assignment)[1] <- "value"
  assignment <- as.data.frame(t(assignment))
  assignment <- cbind(assignment,"varnum"=row.names(assignment))
  assignment <- filter(assignment, assignment$value==1)
  assignment <- merge(assignment,potvar,by="varnum")
  assignment <- subset(assignment, select=c("worker","job"))
  assignment <- rbind(assignment,temp1)
  temp4 <- merge(x = att, y = assignment, by = "worker", all.x = TRUE)
  temp4 <- filter(temp4,temp4$attendance == 1 & is.na(temp4$job))
  temp4 <- subset(temp4, select=c("worker","job"))
  freeworker <- rbind(temp4,filter(temp1, job == 0))
  upver <- melt(ver,id="worker")
  names(upver)[2] <- "job"
  bor <- cbind(bor,1)
  names(bor)[3] <- "value"
  bor <- filter(bor,job != 0)
  upver <- rbind(upver,bor)
  upver <- cast(upver, worker ~ job, sum)
```

```
  # updated versatility matrix

  fval <- optimalassign$objval+nrow(temp1)

output <- list("assignment"=assignment,"updatedv"=upver, "workerpool"=freeworker)
return(output)
  }
```

An example of making use of the functions above to code the basic assignment model is listed as follow (skill matrix, preference, attendance and borrow worker have been read):

```
# run optimization
potvar <- presolve(ver, pre, att, bor)
optinput <- formatting(potvar)
optimalassign <- lp(direction="max" , optinput$obj, optinput$const.mat, optinput$const.dir,
optinput$const.rhs)
output <- otable(optimalassign,optinput,potvar,ver, pre, att, bor)

# check outcome
assignment_table <- output$assignment
updated_versatility <- output$updatedv
surplus_worker <- output$workerpool
```

- **Matlab code:**

Table A: Input/output of Matlab functions

Function	Input	Output
sche	Skill matrix Attendance table Priority table (optional)	Value of the objective function Assignment table
ct	Current skill matrix Absenteeism rate for each worker Total allowance for cross-training	Updated skill matrix
mcs	Absenteeism rate for each worker Number of samples	Monte Carlo simulated samples

```
function [val,schedule]=sche(vmatrix,att,f)
```

```
[nworker,njob] = size(vmatrix);
nx = nworker*njob;

%objective function min -sum
%f=-ones(nx,1);
A=zeros(nx+nworker+njob,nx);
%constraint1 x<=vmatrix
for i = 1:nx
    A(i,i)=1;
    b(i)=vmatrix(ceil(i/njob),i-njob*(ceil(i/njob)-1));
end
%constraint2 one worker only do one job
for i=1:nworker
    A(i+nx,(i-1)*njob+1:i*njob)=ones(1,njob);
    b(i+nx)=att(i);
end
%constraint3 one job only need one worker
for i=1:njob
    A(i+nx+nworker,i:njob:(nworker-1)*njob+i)=ones(1,nworker);
    b(i+nx+nworker)=1;
end
intcon = 1:nx;
lb = zeros(1,nx);
ub = ones(1,nx);
[x,fval] = intlinprog(f,intcon,A,b,[],[],lb,ub);
val=-fval;
for i=1:nworker
    for j=1:njob
        schedule(i,j)=x((i-1)*njob+j);
    end
end
end

function updatev=ct(vmatrix,ncrosstrain)
nsim=30;
[nworker,njob] = size(vmatrix);
% random sample attendance
r=rand(nworker,nsim);
for i=1:nworker
    for s= 1:nsim
        if r(i,s)<prob(i)
            att(i,s)=0;
        else
            att(i,s)=1;
        end
    end
end
% settings
ops = sdpsettings('solver','cplex');
% decision variables
x = binvar(nworker,njob*(nsim+1),'full');
% objective function
obj1 = -sum(sum(x(1:nworker,njob+1:njob*(nsim+1))));
%%%%%%%%%%%% set constraints %%%%%%%%%%%%%%%%%%
% number of training
```

```
constr1 = [sum(sum(x(1:nworker,1:njob)))-sum(sum(vmatrix(1:nworker,1:njob))) <=
ncrosstrain];
% training from 0 to 1
for i = 1:nworker
    for j = 1:njob
        constr1 = [constr1;x(i,j) >= vmatrix(i,j)];
    end
end
% assignment constraints
for s = 1:nsim
    for i = 1:nworker
        for j = 1:njob
            constr1 = [constr1;x(i,j+s*njob) <= x(i,j)];
            constr1 = [constr1;sum(x(1:nworker,j+s*njob)) <= 1];
        end
        constr1 = [constr1;sum(x(i,1+s*njob:(1+s)*njob)) <= att(i,s)];
    end
end
% solve
outputILP = optimize(constr1,obj1,ops)
solution_ILP = value(x);
find(solution_ILP==1);
for i = 1:nworker
        for j = 1:njob
            t(i,j)=solution_ILP(i,j)-vmatrix(i,j);
        end
end
end

function simatt=mcs(prob,nsim)
nworker = length(prob);
r=rand(nworker,nsim);
for i=1:nworker
    for s= 1:nsim
        if r(i,s)<prob(i)
            att(i,s)=0;
        else
            att(i,s)=1;
        end
    end
end
simatt=att;
end
```

- **OPL (CPLEX) code:**

```
//parameters
 int s =...;// number of shift

 range shift=1..s;

 int k =...;
 range day_whole =1..k;

 int k1 =...;
 range day = 1..k1;
```

```
 float rto1[shift]=...;
 float rto2[shift]=...;
 float cx=...;
 float cy=...;
 float abslegacy[shift][day_whole]=...;
 float abstpt[shift][day_whole]=...;
 float absln[shift][day_whole]=...;
 float abstn[shift][day_whole]=...;
 int otherlegacy=...;
 float ar=...;
 float ll=...;
 float ratio=...;//TPT(trim+chassis)/TPT(total)=50%
 float sl1=...;//safety level of 0 tolerance
 float slt1=...;//safety level of 3% tolerance
 float sl2=...;//safety level of 0 tolerance
 float slt2=...;//safety level of 3% tolerance

 //variables
 dvar float+ x[shift];
 dvar float+ y[shift];
 dvar float+ yn[shift];
 dvar float+ tpt[shift];
 //dvar float+ unmet1[shift][day];
 //dvar float+ unmet2[shift][day];
 dvar int  unsafe1[shift][day] in 0..1;
 dvar int  unsafe2[shift][day] in 0..1;
 //dvar float+ unmet1n[shift][day];
// dvar float+ unmet2n[shift][day];
 dvar int  unsafe1n[shift][day] in 0..1;
 dvar int  unsafe2n[shift][day] in 0..1;

 //cost expression
 dexpr float costx=sum(i in shift) (x[i]*cx);
 dexpr float costy=sum(i in shift) (tpt[i]*cy);


 execute CPX_PARAM {
  cplex.epgap= 0.04;
}

 minimize (costx+costy);

 subject to {

       forall (i in shift, j in day)
         full_rto:
              x[i]*(1-abslegacy[i][j])+y[i]*(1-abstpt[i][j]) >= rto1[i]*(1-unsafe1[i][j]);

       forall (i in shift, j in day)
         full_rton:
       x[i]*(1-absln[i][j])+yn[i]*(1-abstn[i][j]) >= rto2[i]*(1-unsafe1n[i][j]);


       forall (i in shift)
         pr1:
         sum(j in day) (unsafe1[i][j])<=k1*(1-sl1);
```

```
      forall (i in shift)
        pr1n:
        sum(j in day) (unsafe1n[i][j])<=k1*(1-sl2);

   forall (i in shift, j in day)
        part_rto:
        x[i]*(1-abslegacy[i][j])+y[i]*(1-abstpt[i][j]) >= rto1[i]*0.97*(1-unsafe2[i][j]);

      forall (i in shift, j in day)
        part_rton:
        x[i]*(1-absln[i][j])+yn[i]*(1-abstn[i][j]) >= rto2[i]*0.97*(1-unsafe2n[i][j]);

      forall (i in shift)
        pr2:
        sum(j in day) (unsafe2[i][j])<=k1*(1-slt1);

      forall (i in shift)
        pr2n:
        sum(j in day) (unsafe2n[i][j])<=k1*(1-slt2);

      forall (i in shift, j in day)
        cut1:
        unsafe2[i][j] <= unsafe1[i][j];

      forall (i in shift, j in day)
        cut2:
        unsafe2n[i][j] <= unsafe1n[i][j];


   forall (i in shift, j in day)
        tpt_week_cons:
        4*y[i]<=3*tpt[i];

      forall (i in shift)
        maxTPTn:
        yn[i]<=tpt[i];

      forall (i in shift)
        maxTPT:
        y[i]<=tpt[i];

   job_ratio_cons:
      ar*(sum(i in shift)(x[i])+otherlegacy)*4/3+ll*4/3>=sum(i in shift)(tpt[i])/ratio;
}
```