

# An Intelligent Model for Solving Manpower Scheduling Problems

## Abstract

The manpower scheduling problem is a critical research field in the resource management area. Based on the existing studies on scheduling problem solutions, this paper transforms the manpower scheduling problem into a combinational optimization problem under multi-constraint conditions from a new perspective. It also uses logical paradigms to build a mathematical model for problem solution and a new multi-dimensional genetic algorithm for solving the model. Moreover, the constraints discussed in this paper basically cover all the requirements of human resource coordination in modern society and are supported by our experiment results. In the discussion part, we compare our model with other heuristic algorithms or linear programming methods and prove that the model proposed in this paper makes a 25.7% increase in efficiency and a 17% increase in accuracy at most.

In addition, to the numerical solution of the manpower scheduling problem, this paper also studies the algorithm for scheduling task list generation and the method of displaying scheduling results. As a result, we not only provide various modifications for the basic algorithm to solve different condition problems but also propose a new algorithm that increases at least 28.91% in time efficiency by comparing with different baseline models.

## 1 Introduction

Member scheduling problem (MSP) is a sub-problem that belongs to scheduling problems (SP). The requirements of the MSP are generally to meet a variety of constraints while arranging flexible work for different employees within a given period and finding out the maximization/minimization of specific objective functions (Pan et al. 2010). As a memorable beginning, Dantzig and Fulkerson (1954) described a mathematical model for solving MSP and obtained the optimal solution using the simplex method for the first time. Since then, not only mathematicians but some scholars with computer science, management science, or even medical background have entered this field. They give different restrictions and mathematical models according to the specific conditions of the manpower scheduling problem, and find some approximate optimal solutions that can be applied in actual production. However, MSP is still a complex

combinatorial optimization problem and an NP-Hard problem(Romero et al. 2016), which means that in the solving process, we need to overcome a number of problems brought by large-scale calculation and optimization.

Therefore, this paper tries to construct a novel mathematical model, that is, under the condition of satisfying a variety of constraints, to propose the optimal objective functions and select the appropriate heuristic algorithms to solve these objective functions. Moreover, we create an optimal generation algorithm that satisfies schedule for employees arrangement.

## 2 Related Work

The problems of manpower scheduling can be divided into single-shift manpower scheduling problem (in the following part this is referred as single-shift MSP problem) and multi-shift manpower scheduling problem (in the following part this is referred as multi-shift MSP problem). The single-shift MSP problem, as its name implies, means that there is only one shift per day for each department, and there is no need to consider the problem brought by rotating of employees. The solution of single-shift MSP is simpler than that of multi-shift MSP and the basis of solving method for multi-shift MSP is also generated from single-shift MSP. Single-shift MSP always has obvious time series features (Kumar et al. 2018) which can help us find solutions. Above all, multi-shift MSP is the essential research area in recent years and we will discuss more about this kind of problems.

To find effective solving method for the problem of personnel scheduling, Sabar, Mon-treuil, and Frayret (2012) design an agent-based algorithm assembly center dynamic environment. Multi-agent refers to the occurrence of multiple 'agents' in the algorithm. There are four kinds of agents: production agent refers to the input that generates the schedule, that is, the producer of the production plan; Workstation agent refers to the agent who sends instructions to the coordination agent based on the production plan. Coordinating agent refers to the agent who arranges the scheduling plan of a site's employees; Employee agency refers to the individual employee and reflects the interests of the employee. Through the cooperation between different agents, the staff of the site can get the right arrangement. At the same time, since the individual interests of the staff are also taken into consideration in the solving process, the satisfaction of the

staff to the task arrangement will also increase. Saber proves that the efficiency of the new algorithm is better than that of the simulated annealing algorithm.

When studying scheduling problems with multi-skill requirements and multi-resource constraints, Zheng, Wang, and Zheng (2017) establish the teacher-learning algorithm (TLBO), combine the resource list with the task list, and propose a solution for task coding and left-shift decoding. Under the framework of TLBO, there are individuals with suitable fitness in the initial population after task coding, which are called teachers, and individuals with low fitness, which are called students. To optimize individual fitness and obtain the optimal solution, TLBO mainly has the following two stages: teacher stage and student stage. In the teacher stage, teachers impart knowledge to the students; in the student stage, students learn from each other to improve their scores or learn from teachers by exchanging different parts. The flowchart of TLBO can be expressed as:

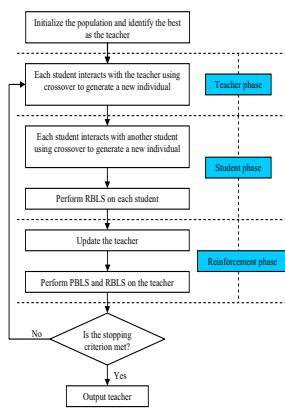


Figure 1: Schematic diagram of solving process (Zheng, Wang, and Zheng 2017)

Recently, Ciancio et al. (2018) propose an integrated approach to solving the personnel scheduling problem (same as manpower scheduling problem), proving the effectiveness of the comprehensive use of the heuristic algorithm. They solve the shift scheduling problem of buses and bus drivers in EU countries by decomposing the problem and applying a variety of heuristic algorithms and then obtained satisfactory results. In addition, these researchers determine initial solution according to the actual situation to minimize the target function group (*time, people*), thus obtaining the solution of the second subproblem. In reality, some EU countries have already applied the scheme.

### 3 Model Establishment

Consider to arrange  $K$  tasks in  $T$  days, while  $K$  tasks are corresponding to  $K$  jobs.  $N_K$  means the number of people in need for  $K$ th job. In total,  $M$  workers are provided to choose. For the given  $L$  constraints,  $\varphi_1, \varphi_2, \dots, \varphi_L$ , our target is to obtain the optimal solution for the objective function  $f(x|\varphi_1, \varphi_2, \dots, \varphi_L)$  in the feasible region, and the specific meaning of  $x$  depends on the situation (Pan et al. 2010).

In this paper we use our mathematical model to solve **min/max/minmax**  $f(x|\varphi_1, \varphi_2, \dots, \varphi_L)$  under certain constraints. In addition, independent variables of the optimal solution  $\varphi_1, \varphi_2, \dots, \varphi_K$  can form one subspace of solution space. Therefore, our model is mainly composed of four parts: constraint conditions, objective functions, solution process and the representation of the solution space.

### 3.1 The Assumptions and Constraints in the Scheduling Problem Solving Model

#### Assumptions

To create the mathematical model for the problem, the following assumptions are necessary:

- Do not take into account the staff's emergency, that is, illness and other similar factors affecting attendance.
- The working time of employees is continuous, the scheduling interval is omitted.
- Employees' job schedules cannot be crossed, that is, one person can only perform one kind of work.
- Working hours of employees are constant.

#### Constraints

To solve or generate **min/max/minmax**  $f(x|\varphi_1, \varphi_2, \dots, \varphi_L)$ , the following constraints are necessary:

As different companies have different business rules and working conditions, the constraints of scheduling can be divided into the following three categories in general:

##### i. Rules of employee availability (Ho and Leung 2010)

These rules represent the constraint rules established by the characteristics of the employee, for example:

- Each employee is not allowed to be on multi-duties  $\varphi_{k1}$ .
- Make sure every job is occupied every day  $\varphi_{k2}$ .
- In a scheduling cycle, the minimum working time of the employee shall not be less than the stipulated lower limit, and the maximum working time shall not exceed the stipulated upper limit. The employee should not work for longer time without permission  $\varphi_{k3}$ .
- The total salary of the employees cannot exceed a certain limit, also cannot fall below a certain limit  $\varphi_{k4}$ .
- There exists upper limit for the total number of employees  $\varphi_{k5}$ .
- Employees should be guaranteed to take time off  $\varphi_{k6}$ .
- ...

##### ii. Rules for working scheduling (Türker and Demiriz 2018)

These rules represent constraint rules resulting from management and business arrangements, for example:

- Urgent tasks must be set as first priority  $\varphi_{y1}$ .
- The total number of people in any jobs is greater than or equal to the minimum number of people required for this specific job, but no more than a certain number of people  $\varphi_{y2}$ .
- Sometimes the company will be asked to schedule in a particular order  $\varphi_{y3}$  (Narasimhan 2000).

- ...

### iii. Other important rules

These rules represent constraints that interfere with the solution space in addition to the above constraints, for example:

- In real life, sometimes we need to carry out a job b shift arrangement is required  $\varphi_{o1}$  (Yingjun and Mingqing 2010).
- Some tasks require the cooperation of different types of workers or different shifts workers  $\varphi_{o2}$  (Su and Liu 2017).
- ...

## 3.2 The Approaches to Establish the Model

### The Combination of Model Condition and Objective Functions

Different industries or companies will choose different constraints. In order to realize the combination of multiple constraints, we consider transforming constraints into different propositions:  $\varphi_k, \varphi_y, \varphi_o$ . These constraints are connected with logical operators to construct the disjunctive and collative paradigms to maximize the range of constraints that can be expressed and covered. Analysis in theory, only consider employees usability rules on the one hand, this combination can have different paradigm  $2^n - 1$ , where  $n$  is the number of constraints, to consider the logical connective  $\wedge, \vee, \neg$ , which is possible to build hundreds of the combination of the constraint condition.

Similarly, for different objective functions, we can reconstruct them into a unified normal form by means of proposition combination. At the present stage of multi-objective optimization problem, the coupling between objective functions is still dominated by  $\wedge$  (intersection). For example, ‘minimum total hours and minimum total salary’ can be expressed as:

$$(T_{total} = \min T_{total}) \wedge (C_{cost} = \min C_{cost}) = True \quad (1)$$

### Determining Constraints and Targets

#### i. Rules of employee availability

We define the arbitrary jobs set  $K$  contains jobs that encoded as  $(1, 2, 3, \dots, k)$ .

For arbitrary job position  $K_j$ , to determine whether the employee  $i$  is on duty that day, the employee of corresponding position can be coded as impact function  $\delta_i^{K_j}$ , the value of this function satisfy:

$$\begin{cases} \delta_i^{K_j} = 1 \\ \delta_i^{K_j} = 0 \end{cases} \quad (2)$$

$\delta_i^{K_j}$  takes 1 means the employee  $i$ , whose job is  $K$ , works on that day, while this function takes 0 to mean he/she is absent. Therefore, the daily working time of employees can be expressed as  $T_K$ :

$$T_K = MOR_K + AFT_K + ENV_K + MID_K \quad (3)$$

Since the encoding method defines the types of employees, the condition  $\varphi_{k1}$  must be satisfied. Now we need to consider the constraint  $\varphi_{k1}$ , which indicates that for employee  $i$

in any position  $K$ , the value of daily impact function  $\delta_i^K$  is not all 0, namely:

$$\varphi_{k2} : \forall j \in K, \sum_{i=1}^{N_j} \delta_i^{K_j} > 0 \quad (4)$$

For constraint  $\varphi_{k3}$ , it is necessary to assume that the lower limit of working time of each post is  $T_l^K$  and the upper limit is  $T_u^K$ , and then define the function to calculate the corresponding working time  $f_1$ . For any jobs  $K$ , the actual working time of employees in that job  $f_1$  can be expressed as:

$$f_1 : \sum_{i=1}^{N_j} T_K \delta_i^{K_j}, \quad j \in (1, 2, 3, \dots, K) \quad (5)$$

Therefore, the expression of constraint  $\varphi_{k3}$  is:

$$\begin{cases} \varphi_{k3} : T_l^{K_j} \leq f_1 \leq T_u^{K_j} \\ j \in (1, 2, 3, \dots, K) \end{cases} \quad (6)$$

For the constraint  $\varphi_{k4}$ , it should be assumed that the lower limit of salary expenditure stipulated by the company is  $C_l^{cost}$  and the upper limit of salary expenditure is  $C_u^{cost}$ . Moreover, since the total salary of employees can be calculated, at the next step we define the function  $f_2$  to calculate the full salary, and the expression is:

$$f_2 : T_{day} \sum_{j=1}^K \sum_{i=1}^{N_j} \delta_i^{K_j} C_{K_j} \quad (7)$$

Therefore, the expression of constraint  $\varphi_{k4}$  is:

$$\begin{cases} \varphi_{k4} : C_l^{cost} \leq f_2 \leq C_u^{cost} \\ j \in (1, 2, 3, \dots, K) \end{cases} \quad (8)$$

As for the constraint  $\varphi_{k5}$ , since it expresses the limit of the total number of employees in each post, we need to sum up the number of employees in each post. Finally, the constraint can be expressed as:

$$\varphi_{k5} : N = \sum_{j=1}^K N_j \leq M \quad (9)$$

When considering the constraint  $\varphi_{k6}$ , which involves employee vacation time, A counter  $Count_{(\varphi=True)}$  for analysis condition  $\varphi$  needs to be created, and the initial value of the counter is assigned to 0. The counting process of the counter can be interpreted as increasing the value of  $Count_{(\varphi=True)}$  by one when  $\varphi = True$ , so this counter can be used to calculate the vacation time. We also define the upper limit of employees’ vacation time as  $T_{rest}$  so that  $\varphi_{k6}$  can be expressed as:

$$\varphi_{k6} : 0 \leq Count_{\delta_i^{K_j}=0} \leq T_{rest} \quad (10)$$

#### ii.Rules for working scheduling

For emergency task constraint  $\varphi_{y1}$ , the general way to deal with it should be to transfer  $\varphi_{y1}$  employees in work and spend certain amount of time  $T_0$  to complete the task. The priority of emergency task should be higher than that

of normal work. We should also consider the mechanism of reward and punishment, these changes can be reflected in the expenditure of wages and determined by  $C_{cost}$ . Now, assuming that the total working time in the original scheduling task is  $T_{total}$ , the constraint condition indicates that some parameters in the scheduling task, namely the total number of assigned employees, the total working time  $T_{total}$  and the wage expenditure  $C$ , have changed. Therefore, the constraint condition can be expressed as:

$$\varphi_{y1} : \begin{cases} N' = N - \alpha \\ T' = T_{total} - T_0 \\ C' = C_{cost} - C_{bonus} + C_{punishment} \end{cases} \quad (11)$$

For constraint  $\varphi_{y2}$ , the minimum number  $N_l^{K_j}$  and the maximum number  $N_u^{K_j}$  on each job  $j$  need to be considered. But in some problems the value of  $N_u^{K_j}$  may not be given, then its value can be determined using the weighted average method, namely:

$$N_u^{K_j} = \frac{N_l^{K_j}}{\sum_{i=1}^K N_l^{K_i}} M \quad (12)$$

Therefore,  $\varphi_{y2}$  can be expressed as:

$$\varphi_{y2} : \forall j \in (1, 2, 3, \dots, K), \quad N_l^{K_j} \leq N_j \leq N_u^{K_j} \quad (13)$$

Regarding the expression of constraint  $\varphi_{y3}$ , it involves the generation algorithm of the shift table. Because of its special diversity (for example, the diversity of order and the diversity of task selection), no universal representation method has been discovered yet. When solving the problem, we can only give a solution for its special case.

### iii. Other important rules

Firstly we study the constraint  $\varphi_{o1}$  that allows jobs shifts (multi-shifts), including job-sharing, shift work and flex-time. In this case, since the four shifts for involved working time can all be arranged with different employees, the working status of the employees needs to be re-determined. Now we define that  $\delta_i^j$  represents the working status of the employee with the number  $i$  under the  $j_{th}$  working hour for every day:

$$\left\{ \begin{array}{l} \delta_i^j = 1 \\ \delta_i^j = 0 \\ s.t. \sum_{j=1}^4 \delta_i^j > 0 \end{array} \right. \quad \text{and} \quad \left\{ \begin{array}{l} j = 1 (\text{working at MOR}) \\ j = 2 (\text{working at AFT}) \\ j = 3 (\text{working at ENV}) \\ j = 4 (\text{working at MID}) \end{array} \right. \quad (14)$$

Under this condition, if we consider two vectors  $\vec{M}_i = (\delta_i^1, \dots, \delta_i^4)$  and  $\vec{N}_m = (MOR_m, \dots, MID_m)$ , the total working time  $T_{total}$  can be expressed as new formula:

$$T_{total} = \sum_{m=1}^K \sum_{i=1}^{N_m} T_i^m \quad \text{where} \quad (15)$$

$$T_i^m = \vec{M}_i \cdot \vec{N}_m$$

Similarly, the function  $f_3$  for calculating the total salary will also change. After updating, the function  $f_3$  can be expressed as:

$$f_3 : \sum_{n=1}^{T_{day}} \sum_{j=1}^K \sum_{i=1}^{N_j} \left( \sum_{m=1}^4 \delta_i^j C_m \right) \quad (16)$$

In this expression,  $C_m$  corresponds to the  $m_{th}$  hour's wage. For the requirement constraint  $\varphi_{o2}$  of the cooperative task, there are two approaches to express it. The first method is to treat it as an individual emergency task ( $\varphi_{y1}$ ), equivalent to transferring a portion of the total number of employees to complete that emergency task; for the second one, considering the low frequency of cooperative tasks (usually with fixed cycles) and a large number of participants, the cooperation problem is separated from the multi-shift scheduling problem, and the design algorithm is optimized separately. This alternative scheme will not have a tremendous negative impact on the stability and accuracy of the model.

Finally, we briefly analyze the objective function. For the objective function in combinatorial optimization, the more commonly used is the total number  $N$  of employees seeking min/max/minimax, the total time  $T_{total}$ , and the total salary expenditure  $C_{cost}$ .

### 3.3 Scheduling Table Generation Algorithm

Considering the automaticity requirement of generating scheduling list and the humanization requirement of labor task assignment, this paper presents a scheduling list generator as an algorithm after obtaining the feasible solution under several constraint conditions. The generator based on random selection of employees can guarantee the relative rationality of average working time. In addition, if the constraints and solutions of the scheduling problem are more complicated, the update function of this generation algorithm will only require simple modification on the logic and selection of the function *Suitable()*.

---

#### Algorithm 1 Scheduling Table Generation Algorithm

---

```

MemberList ← [person1, person2, ..., personn]
Workable ← []
Day ← number
Worktime ← []
i ← 0
while i < 7 do
    count ← 0
    list ← []
    while count <= n do
        man ← random(MemberList)
        if man is Suitable(Workable) then
            list.append(man)
            workable[man] ← workable[man] + 1
            count ← count + 1
        end if
    end while
    Worktime.append(list)
    i ← i + 1
end while

```

---

### 3.4 The Representation of Solution Space

Consider that there are three key identifiers that distinguish different jobs and tasks in scheduling problem: employee identifiers, job identifiers, and time identifiers. Therefore, a solution space can be created, where the X-axis (channel 1) is the employee axis, representing different employees; the

Y-axis (channel 2) is the time axis, which represents the circulation of four shifts in the morning, middle, evening and midnight; the Z-axis (channel 3) is the type of job axis, representing different types of work. For efficiency of solution process, we separate channels corresponding to Z-axis and then input different matrices to the algorithm part and in the next step we combine the different result matrices into the space. Therefore, the process to generate solution space can be explained as:

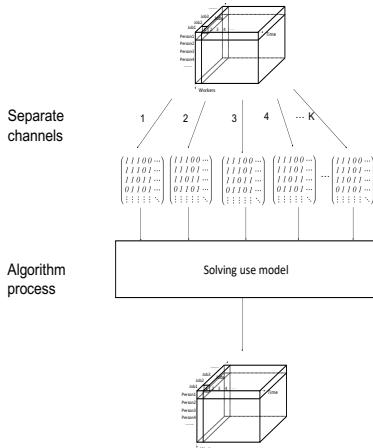


Figure 2: Schematic diagram of solving process

It can be proved that the separated channel matrix is equivalent to the solution space in reality. Moreover, for the elements in channel matrix rows and columns, their definition is shown as following:

$$a_{(i,j)} : \begin{cases} i : \text{rows mean the number of staff} \\ j : \text{columns mean working time} \\ a_{(i,j)} = 1 \quad \text{this worker attends} \\ a_{(i,j)} = 0 \quad \text{this worker does not attend} \end{cases}$$

### 3.5 Genetic Algorithm Implementation

In this paper, inspired by the work finished by Wang, Yalaoui, and Dugardin (2017), two improved genetic algorithms (GA) based on the adaptive multidimensional input are used to solve the scheduling problem.

Single-objective optimization problem refers to the optimization problem with an objective function, while multi-objective optimization problem is a number of different objective functions using different constraints to link the operation of the object. It can be seen that the solution of multi-objective optimization problem is generally more difficult and complex than that of single-objective optimization.

For the representation of constraints, it is necessary to introduce the mechanism of penalty function to combine the constraint conditions with GA in the process of solving, and there are mainly two approaches to express the penalty function: external penalty function method (Shahnazari-Shahrezaei, Tavakkoli-Moghaddam, and Kazemipoor 2013) and internal penalty function method (Du et al. 2007). Both methods are tried in this paper and it is proved that both

methods can be used to solve manpower scheduling problem.

## 4 Experiments and Tests of the Model

The following experiment part is about how to apply the new scheduling model proposed in this paper to solve one in a shopping mall.  $M$  is assigned to 60.

| Code | a       | b     | c     | d          | e          | f       |
|------|---------|-------|-------|------------|------------|---------|
| Type | manager | clerk | guard | salesclerk | tallyclerk | cleaner |

### 4.1 Experiment1: Minimize Total Time under the Basic Constraints

To minimize total time under the constraints  $\varphi_{k1} \wedge \varphi_{k2} \wedge \varphi_{k3} \wedge \varphi_{k4} \wedge \varphi_{k5} \wedge \varphi_{k6}$ , using the preprocess analysis for this problem, the requirement can be expressed as:

$$\begin{cases} \min & T_{\text{total}} \\ \text{s.t.} & \varphi_{k1} \wedge \varphi_{k2} \wedge \varphi_{k3} \wedge \varphi_{k4} \wedge \varphi_{k5} \wedge \varphi_{k6} \end{cases} \quad (17)$$

The initial number of individuals in the population is 100. To compare the efficiency of GA, in this experiment, two different chromosome coding methods are adopted to solve the minimum value. They are binary coding and real & integer mixed coding.

If BG (binary coding) method is adopted, and after 50 generations, min value of this objective function is 5250. For the optimal solution, the values of the independent variables are:

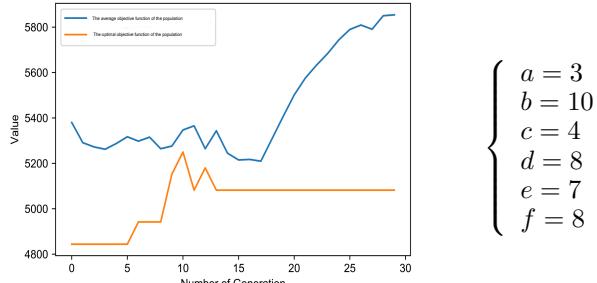


Figure 3: BG coding result

If RI (real number & integer number mixed coding) is adopted, and after 50 generations, the GA will get a minimum value of 4942. For the optimal solution, the values of the independent variables are:

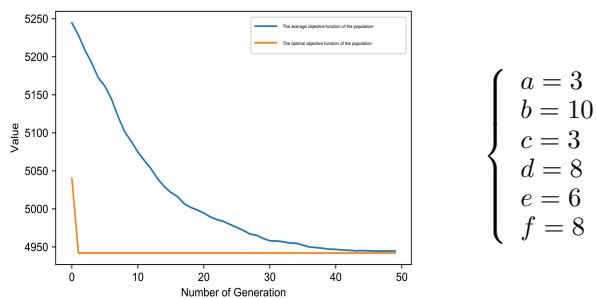


Figure 4: RI coding result

From the experimental results, we can see that the result obtained by using RI coding has a faster convergence speed, and the objective function value is relatively smaller, which shows that RI coding is more suitable for the following solving operations.

After obtaining the number of people needed for the arrangement, the value of the impact function should be determined. At this point, the problem has been transformed into a 0-1integer programming problem, which can be solved by using genetic algorithm again, and the following results can be obtained:

Table 1: Experiment results obtained by GA

| Evaluating times | Time cost       | Optimal function value |
|------------------|-----------------|------------------------|
| <b>8400</b>      | <b>0.07379s</b> | <b>3850</b>            |

We input the results obtained from above GA to the generator algorithm for scheduling table to get solution space  $S_{xyz}$ :

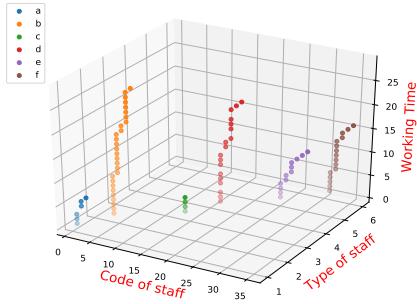


Figure 5: Solution space for experiment1

## 4.2 Experiment2: Work Shifts in the Scheduling Problem

If the work shifts ( $\varphi_{o1}$ ) is allowed, that is, the form of which is generally determined by the number of persons required and the number of shifts required. For example, three positions in five shifts means that five people should be arranged in three positions in a shift cycle. Since the situation has changed, an algorithm can be designed to consider the situation of shifts work. The sequence of employees in the list can be adjusted by the function  $Order()$ . Therefore, the result of the generator algorithm is as follows:

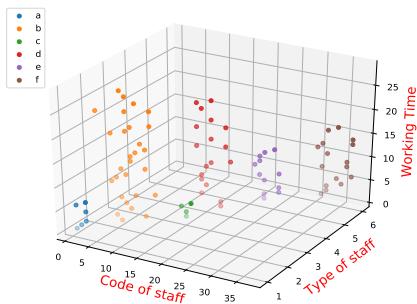


Figure 6: Solution space  $S_{xyz}$  for experiment2

## 4.3 Experiment3: The Solution for the Problem after Complex Combination of Constraints.

The experiment is divided into two parts to explore the possibility of logical paradigms considering the representation of constraints to be joined by logical operators. In the first part, the penalty function is directly linked by logical operators from the point of view of genetic algorithm. For example, the constraint conditions are: each post cannot be interleaved with duty, and each post has someone on duty every day, and the total salary of the employees should be greater than a certain amount; or the employee's working time is within a certain range and the problem of the minimum total working time. The problem can be shown as:

$$\begin{cases} \min & T_{\text{total}} \\ \text{s.t.} & \varphi_{k1} \wedge \varphi_{k2} \wedge (\neg \varphi_{k5}) \vee \varphi_{k3} \end{cases} \quad (18)$$

This graph represents the results obtained using GA:

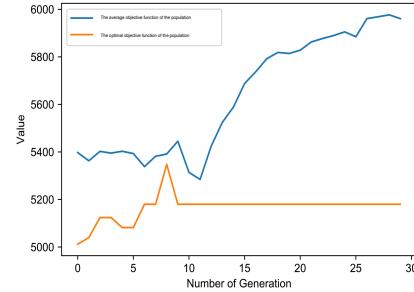


Figure 7: Result for experiment3

## 4.4 Experiment4: Scheduling Problem under Multi-objective Optimization

In reality, different enterprises and different managers will have different demands, which also determine that there are various combinations of distinct objective functions. The scheduling problem of multi-objective combinatorial optimization can still be solved by using the model proposed in this paper, as a GA with elite strategy inspired by other researches (Deb et al. 2002).

The following example shows the application of the algorithm in solving the multi-objective optimization scheduling problem. The objective functions are: Minimizing the number of clerks, guards and tallyclerks  $f_a$  while minimizing the working time  $f_b$ .

$$\begin{cases} \min & f_a : \sum_{i=2,3,5} N_i \\ \min & f_b : \sum_{j=1}^K \sum_{i=1}^{N_j} T_k \delta_i^{K_j} \\ \text{s.t.} & \varphi_{k1} \wedge \varphi_{k2} \wedge \varphi_{k3} \wedge \varphi_{k4} \wedge \varphi_{k5} \wedge \varphi_{k6} \end{cases} \quad (19)$$

After the optimization conditions are substituted in, the following results can be obtained:

Table 2: The parameters obtained by GA

| Time cost      | Optimal value of $f_a$ | Optimal value of $f_b$ |
|----------------|------------------------|------------------------|
| <b>0.1096s</b> | <b>21</b>              | <b>4844</b>            |

## 5 Discussion

In order to verify the efficiency of the new model we proposed, this paper compares some baseline models with the approach proposed in this paper from two aspects. Moreover, the superiority of the new approach proposed in this paper is judged by studying the time cost, the accuracy of solutions, and the rationality of the generated results.

### 5.1 Obtaining the Number of Employees

In order to compare the efficiency of the solution algorithm horizontally, this paper also uses integer programming algorithm (IP), particle swarm optimization (PSO) and simulated annealing algorithm (SA) to complete Experiment 1, and the following results are obtained:

Table 3: Comparison results for different algorithms

| Name of algorithm | Time cost       | Accuracy    | Convergence |
|-------------------|-----------------|-------------|-------------|
| New-GA            | <b>0.19899s</b> | <b>100%</b> | <b>2</b>    |
| IP                | 0.01378s        | 70%         | 1           |
| PSO               | 0.26791s        | 100%        | 3           |
| SA                | 1.07792s        | 83%         | 4           |

Firstly, we explain the meaning of the data in the table. *Runningtime*: this variable refers to the time cost by different algorithms. *Accuracy*: this value is defined as  $\frac{\text{Result of New-GA}}{\text{Result of other algorithms}} * 100\%$ . *Convergence*: it is applied to observe whether the final solution result is stable. First, we find that in time cost part IP performs better than New-GA. However, we believe that this is more related to the size of the data set. When integer programming faces more dimensions and more constraints, the running time will significantly increase. In addition, the integer programming method is usually applied to solve some simple constraints. For complex problems, there is still a significant gap between the result of IP and the real optimal solution. In the comparison of heuristic algorithms, the new genetic algorithm proposed in this paper has apparent advantages, which are not only the high efficiency but also the outstanding performance in the selection of optimal solutions. In the running time, it is 25.7% faster than the particle swarm optimization (PSO); in the accuracy of the running result, it is 17% higher than the simulated annealing algorithm (SA). Therefore, the genetic algorithm proposed in this paper is useful in completing complex arrangements manpower scheduling task, and it is worth further optimization and investigation.

### 5.2 Generator Algorithm of Scheduling Plan

In the comparison of the scheduling table generation algorithm, this paper selects a manpower scheduling algorithm related to the research results of Korf (2002) for comparison, which is adapted by the baseline model (Thane 2017) that has related code collections on Github. It can be used to study the effectiveness of the generation algorithm proposed in our paper. By comparing the completion of the scheduling tasks required by the two algorithms for the two

different time periods of 7 days and 30 days, we can discuss the effective solutions shown in Figure 8&9. We can

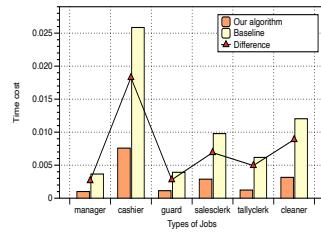


Figure 8: Results for 7 days

determine that from the results, this paper has a significant advantage over the method provided by the baseline model in the optimization of solution efficiency. For the arrangement of different jobs, the efficiency can be increased by up to 40.28%, and at least by 28.91%, which can save human, material and financial resources for various companies. As a result, the mathematical model proposed by this paper is successful and highly efficient.

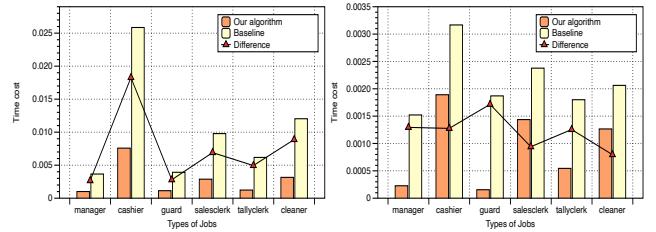
## 6 Conclusions

In this paper, we propose a new model that can be applied to solving single-shift MSP or multi-shift MSP. It is creative and improves the efficiency of enterprise scheduling tasks, thereby optimizing the human resources structure of the enterprise and creating more excellent value. Our work transforms the traditional scheduling problem as a combinatorial optimization problem and identifies two most critical elements, which are the determination of staff allocation and the generation of a specific scheduling arrangement. First of all, because of the complicated situation required by the current scheduling problem, this paper combines different constraints by combining logical paradigms. Subsequently, this paper determines the value of the optimal solution function through the operation results of the improved genetic algorithm to obtain the determination for the number of employees required for different jobs. Finally, by studying the new scheduling generation algorithm, the arrangement of manpower scheduling problems can be generated.

In order to verify the efficiency of the new scheduling model, this paper compares the new model with different baseline models in terms of different aspects, such as efficiency, convergence, and accuracy. The experimental results show that the new model (new algorithm) proposed in this paper performs better than the baseline models, thus affirming the research value of our work.

## 7 Future Work

In the future, our research will focus on the following aspects: (1) Make our model performs better in solving the scheduling problem when discrete emergencies occur. (2) The model proposed in this paper is slightly less efficient than the integer programming method in dealing with small-scale scheduling problems. In the future, we will start from this aspect, try to narrow the gap between the two methods, and finally achieve overtaking inefficiency.



## References

- Ciancio, C.; Laganà, D.; Musmanno, R.; and Santoro, F. 2018. An integrated algorithm for shift scheduling problems for local public transport companies. *Omega* 75: 139–153.
- Dantzig, G. B.; and Fulkerson, D. R. 1954. Minimizing the number of tankers to meet a fixed schedule. *Naval Research Logistics Quarterly* 1(3): 217–222.
- Deb, K.; Pratap, A.; Agarwal, S.; and Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6(2): 182–197.
- Du, X.; Wang, H.; Dang, Q.; Liu, X.; and Huang, Y. 2007. Optimization Algorithm of Initial Orbit Based on Internal Penalty Function Method. In *2007 IEEE International Conference on Systems, Man and Cybernetics*, 3202–3207. IEEE.
- Ho, S. C.; and Leung, J. M. 2010. Solving a manpower scheduling problem for airline catering using metaheuristics. *European Journal of Operational Research* 202(3): 903–921.
- Korf, R. E. 2002. A new algorithm for optimal bin packing. In *Aaai/Iaai*, 731–736.
- Kumar, T. S.; Wang, Z.; Kumar, A.; Rogers, C. M.; and Knoblock, C. A. 2018. Load Scheduling of Simple Temporal Networks Under Dynamic Resource Pricing. In *AAAI*, 6227–6236.
- Narasimhan, R. 2000. An algorithm for multiple shift scheduling of hierarchical workforce on four-day or three-day workweeks. *INFOR: Information Systems and Operational Research* 38(1): 14–32.
- Pan, Q.-K.; Suganthan, P. N.; Chua, T. J.; and Cai, T. 2010. Solving manpower scheduling problem in manufacturing using mixed-integer programming with a two-stage heuristic algorithm. *The International Journal of Advanced Manufacturing Technology* 46(9-12): 1229–1237.
- Romero, P.; Robledo, F.; Bevilacqua, P.; and Delafuente, M. 2016. Man power optimization in large-scale corporations. *Investigación Operacional* 37(2): 173–183.
- Sabar, M.; Montreuil, B.; and Frayret, J.-M. 2012. An agent-based algorithm for personnel shift-scheduling and rescheduling in flexible assembly lines. *Journal of Intelligent Manufacturing* 23(6): 2623–2634.
- Shahnazari-Shahrezaei, P.; Tavakkoli-Moghaddam, R.; and Kazemipoor, H. 2013. Solving a multi-objective multi-skilled manpower scheduling model by a fuzzy goal programming approach. *Applied Mathematical Modelling* 37(7): 5424–5443.
- Su, T.-S.; and Liu, S.-C. 2017. Integrated supporting co-operation model with fuzzy approach for staff scheduling problem in service supply chain. In *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 369–373. IEEE.
- Thane, T. 2017. conference-mgmt. <https://github.com/thanethomson/conference-mgmt>. Online.
- Türker, T.; and Demiriz, A. 2018. An Integrated Approach for Shift Scheduling and Rostering Problems with Break Times for Inbound Call Centers. *Mathematical Problems in Engineering* 2018.
- Wang, X.; Yalaoui, F.; and Dugardin, F. 2017. Genetic algorithms hybridized with the self controlling dominance to solve a multi-objective resource constraint project scheduling problem. In *2017 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, 39–44. IEEE.
- Yingjun, W.; and Mingqing, S. 2010. The External Penalty Function Method for Optimization Design of Reinforced Concrete Underground Penstock. In *2010 International Conference on Computing, Control and Industrial Engineering*, volume 2, 345–347. IEEE.
- Zheng, H.-y.; Wang, L.; and Zheng, X.-l. 2017. Teaching–learning-based optimization algorithm for multi-skill resource constrained project scheduling problem. *Soft Computing* 21(6): 1537–1548.