Hindawi
Journal of Applied Mathematics
Volume 2019, Article ID 2674697, 20 pages
http://dx.doi.org/10.1155/2019/2674697

Research Article

Mixed Optimal Scheduling Model of Flexible Service System Based on Inverted Triangle

Shipei Hu[1] and Yujun Sun[2]

[1] *Jiyang College, Zhejiang A and F University, 77 Puyang Rd., Zhuji 311800, Zhejiang, China*
[2] *School of Economics and Management, Beijing Forestry University, 35 Qinghua East Rd., Haidian District, Beijing 100083, China*

Correspondence should be addressed to Shipei Hu; 081018024@fudan.edu.cn

Received 4 January 2019; Accepted 27 March 2019; Published 2 June 2019

Academic Editor: Frank Werner

In the study presented in this paper, we built a nonlinear binary integer programming model of a flexible scheduling problem for the Department of Zhejiang Provincial Local Tax Services. One difference between our model and typical ones is that whereas in the latter the number of open windows within each working day is fixed, in our model it is not. We used a variety of integer programming software in an attempt to solve our scheduling model; however, unfortunately we could not find an optimal solution. Thus, we tested all the combinations of different numbers of employees to construct the optimal solution. When we tested our model in the tax office of Lishui City, China, the average waiting time of taxpayers was less than 15 min and the employees working hours were clearly reduced. Thus, a noteworthy improvement in the quality of the service is achieved by the model.

1. Introduction

Together with the change in governmental functions, the reform of the administrative approval system, and the boost of local tax informatization in Zhejiang Province, China, more service items are now provided by the local tax service windows and thus the service quantities have increased rapidly. However, service capabilities have not improved proportionately, because the number of service windows and employees is fixed, which results in a long queue of taxpayers, a decline in the service quality in the tax service halls, and damage to the external image of the local taxation bureau.

The primary cause of the queue is that the production capacity of the windows is insufficient to produce the external inventory and the windows pass the inventory cost of the production process to the taxpayers. In order to alleviate the situation of long queues, at present, the measures taken by the tax service halls management are limited mostly to increasing the input of equipment to reduce the inventory and the taxpayers waiting cost. However, the problems that accompany this simple and crude extension of improvements is that even if there is superfluous investment in various types of equipment and window employees are overequipped, a contradiction still exists between the fixed nature of production capacity and the fluctuation in service demand, which in turn may lead to an increase in the chances of periods occurring when employees are idle and in window service costs. Therefore, the determination of an appropriate arrangement of flexible working hours and window services in tax service halls is the core problem that needs to be solved.

From the perspective of management and service needs, with the aim of handling the feature of great fluctuations in service demand and so that the supply satisfies the taxpayers needs more appropriately, in this study we addressed the dynamic settings of the windows and the flexible working hours of the window employees and attempted to realize the optimal allocation of resources by using a basic queuing model and nonlinear 0-1 integer programming model. The term flexible working hours means that window employees can choose their work place and hours flexibly and autonomously and not according to the same fixed work schedule, on the premise that they have completed their regulated duties. In a flexible service system, the working time and number of window employees are dynamically adjusted by optimizing the personnel combination and implementing flexible work arrangements, according to the changing situation of the flow of taxpayers and the business volume at different time periods.

In order to provide top-quality service to the taxpayers, with the assistance of the taxation bureau of Lishui City we distributed a questionnaire to all the window employees in the city and to nearly two thousand of the taxpayers and collected almost 20000 data items. The analysis of these data items using SPSS software showed that the maximum average waiting time tolerated by taxpayers is 15 min and the average time duration for which window employees can provide high-quality services is one hour. After one hour of efficient working, the psychological and physiological state of the window employees is reflected in different levels of burnout, resulting in a decline in the quality of service and an increase in the work error rate. After the window employees practiced 15 min of stress relaxation, their quality of service returned to normal levels. Therefore, this type of working model is more flexible, can effectively include humanistic care, motivates the window employees, improves window service efficiency and quality of service, and reduces management costs. In general, it has the following advantages.

(1) It enhances the working efficiency of the windows and reduces taxpayers waiting time. After the launch of our integrated management system for tax hall services, a performance evaluation of the window employees is also planned. Work efficiency is the main measure of performance, and the key to improving it is shortening the duration of each business transaction to a reasonable extent. The key factors influencing efficiency are the physical and mental stability of the window employees and the service capacity and level. The implementation of a flexible service system can help distinguish core from noncore work time and ensure that the window employees working in noncore time can rest sufficiently and business training. Thus, they can have sufficient energy to maintain their best working state and thereby their job performance in core work time is improved. The implementation of a flexible service system can also reduce taxpayers waiting time by allowing a sufficient number of windows to be open according to the measured flow of taxpayers and the business volume.

(2) It alleviates the pressure on window employees and meets the requirements related to the background work. The sources of work pressure are various because of the specific characteristics of window work. The first is the fixed schedule, including the same work place and work time throughout the year. The second is the simplistic content; that is, the business operations are relatively mechanical and repetitive. The third is the uniform service standard and the enforcement of strict disciplinary standards, including the employees appearance, and the discipline related to the tax window operation process, service quality, etc. Because they face taxpayers all day, window employees remain in a state of mental pressure for a long time, which may influence their work efficiency, reduce the service level, and more seriously damage the relationship between the body that levies the tax and the taxpayer, causing the employees negative emotions such as anxiety and boredom. Therefore, it is undoubtedly beneficial to establish a comprehensive mechanism for psychological counselling and pressure relief for employees. The establishment of a flexible work system will allow the window employees an appropriate amount of time to receive psychological counselling and to decompress and rest and will effectively reduce the long-term physical fatigue and mental stress. A flexible service system will also allow the employees time to organize the taxpayers archives, record the collected information in the ≪ *Dragon version tax system* ≫ and send the related documents to management, as well as audit tax exemptions, effectively meeting the demands related to the background work.

(3) It promotes the study of window service and communication and improves professional quality. Tax management refinement is improved unceasingly, while the fixed nature of window positions and the monotonous work content may affect the professional improvement of employees. Under the flexible service system, flexible working hours could allow the development of the employees professional study of window service and communication with employees in other business positions, which could convert employees more specialized talents related to tax administration to all-round talents.

(4) It improves the management level of the tax service hall, optimizes the tax payment service, and enhances the satisfaction of taxpayers. The tax service hall is the main bridge and communication link between the two sides; that is, the tax levying body and taxpayer. It is an important window for displaying the image of the tax authorities and officials and providing solutions to tax-related issues, as well as an important platform on which to offer quality tax services. A flexible working schedule can not only solve a series of existing problems, but also embody the people-oriented, advanced service-first management idea, which will thus lead to optimization of the tax service, innovation in service methods, improvement of the management level, demonstration of the spirituality of the employees in the new era, and, to a certain extent, enhancement of the taxpayers satisfaction.

In general, the number of windows in the tax service halls in Lishui City, Zhejiang Province, is excessive relative to the demand. However, there are still a few tax service halls that cannot meet the demand, which affects the service quality. This paper presents the so-called inverted triangle flexible shift model of full- and part-time window employees to solve the problem of an insufficient number of window employees. According to the inverted triangle flexible scheduling model, full-time window employees are assigned more working hours and part-time window employees fewer working hours at the window. The length of the working hours of the window employees is from long to short, and thus the top to bottom composition of the graph is like an inverted triangle. The inverted triangle scheduling model with its combination scheduling of both full- and part-time window employees both ensures the service quality of the windows and reduces their service cost.

The rest of the paper is organized as follows. In Section 2, we review the personnel scheduling literature. In Section 3, we scientifically set the number of open windows at different time periods according to a large data analysis and queuing theory. In Section 4, we describe a flexible scheduling model that meets the needs of the tax department in Zhejiang

Province. In Section 5, we establish a set of iterative algorithms to construct the optimal solution of the flexible scheduling model. In Section 6, we provide an example of flexible scheduling to verify the effectiveness and feasibility of our algorithms. In Section 7, we discuss the potential application of the scheduling model for decision-making in the Zhejiang Provincial Tax Department.

2. Literature Review

In the research field of queuing and flexible scheduling problems, Deutsch et al. [1] presented a successful application of queuing theory to the scheduling of a large bank. Targeting cost optimization, Hammond et al. [2] set up a spreadsheet model based on queuing theory to achieve the required number of service personnel in a bank and verified the effectiveness of the proposed model through simulations. Jones et al. [3] analysed the relationship between the actual and acceptable queue waiting time and validated it in an empirical study. So et al. [4] established a queuing theory model to verify the ability of the dynamic adjustment service function to reduce the length of the service line. Nosek et al. [5] claimed that queuing theory can be used to evaluate the employees working arrangements, working environment and productivity, and the customers waiting time and waiting environment. They applied queuing theory and customer satisfaction to the field of pharmaceutical research and showed that, if managers use queuing theory correctly to make the appropriate decisions, then customers, employees, and managers will all be satisfied. On the basis of queuing theory, Wang et al. [6] presented a fast channel model to improve the efficiency of the bank queuing system. By modifying a greedy algorithm and then using MATLAB to execute the numerical simulation of a multiple optimization model, they obtained the experimental results that a system that includes a fast channel can reduce customers average waiting time in both the regular and fast channel queue. Using the queuing model, Ogunwale et al. [7] conducted a comparative research study on the waiting time of customers in two banks and provided corresponding suggestions.

Moondra [8] set up a linear programming model of bank employee scheduling. Kra-Jewski et al. [9] described a scheduling system for check-decoding personnel and its implementation in large banks. They then assessed its effect on cost savings and other functions of the system. Ernst et al. [10] identified the volatility of customer demands in a day as the source of the main difficulties in bank employee scheduling and that handling this volatility depends on appropriate arrangements for full-time employees and part-time employees. Mabert et al. [11] proposed two types of heuristic scheduling methods to meet the volatility of customer demands by means of using part-time employees. Both methods were aimed to minimize the number of windows and the window employee transfers between branches; the validity of the method was verified by using the actual data of a bank. In particular, Burns et al. [12] introduced an approach for solving multiple-shift manpower scheduling problems by means of an algorithm that constructs an optimal schedule for a large and common class of scheduling problems.

3. Forecast to Meet the Needed Number of Open Windows

3.1. Prediction of the Flow of Customers in the Tax Service Hall. According to an analysis of the historical data provided by several tax service halls of the local taxation bureau of Lishui City, the number of taxpayers obeys Poisson distribution, and in each month the numbers of taxpayers at the beginning, middle, and end of the month may be different. The business in the tax service hall can be divided into two seasons: slack and busy. In addition, there are two types of visitor-flow-rate data of taxpayers: one is called the busy time type and the other the idle time type. The first type occurs usually at the beginning and end of a month and the second in the middle of a month. The average number of taxpayers arriving at the tax service hall at different times (busy and idle) is divided into busy season and slack season separately, where different time refers to, for example, 08:30–09:30, 09:30–10:30, etc. Meanwhile, we can predict the average total number of taxpayers arriving at the tax service hall at different times (busy and idle) in different seasons, separately, according to the autoregressive moving average (ARMA) model.

3.2. Calculation of the Minimum Required Window Number in Different Time Periods According to Queuing Theory. The analysis of all the data collected from the tax service halls showed that the entire service process from the customers arrival to departure obeys the standard queuing model of $M/M/c$. The predicted value of the average arrival rate of the taxpayer per hour can be calculated using the historical data and the average service number per hour can be calculated by dividing the total number of services with the total taxpayers service time.

In the standard queuing theory, c represents the number of open windows in the tax service hall, λ the log-run per hour average arrival rate of the taxpayers, μ the daily average service rate at the windows and L_q the number of the taxpayers in the queue. Then, $\rho = \lambda/c\mu$ denotes the service intensity. According to the queuing theory, the average waiting time [13] until receiving service is

$$\begin{aligned} W_q &= \frac{L_q}{\lambda} = \frac{(c\rho)^c \rho}{c!\,(1-\rho)^2} P_0 \\ &= \frac{(c\rho)^c \rho}{c!\,(1-\rho)^2} \left[\sum_{k=0}^{c-1} \frac{(c\rho)^k}{k!} + \frac{(c\rho)^c \rho}{c!\,(1-\rho)} \right]^{-1}. \end{aligned} \tag{1}$$

According to the complaints of the taxpayers and the data from the responses to the questionnaire, the average waiting time for the taxpayers should not be more than 15 min. The local tax bureau in Lishui City has a requirement, that is, $W_q \leq 0.25$ hr. In addition, on the basis of quality of service assurance, the number of open windows c should be as small as possible. Because c is a positive integer and it is difficult to compute its analytical solution, the calculation is performed according to the following steps.

Step 1. Let $c = \lambda/\mu + 1$, based on the queuing process to achieve the steady state.

Step 2. Use formula (1) to compute W_q. If $W_q \le 0.25$, then the algorithm is terminated and c is the minimum number of open windows required to satisfy the demands; if $W_q > 0.25$, then go to Step 3.

Step 3. Let $c = c + 1$, and then return to Step 2.

Because of the volatility of the number of taxpayers arriving at the tax service hall in different time periods and seasons, the average arrival rate λ also varies. In addition, the service rate μ differs because of the varying business skills of the window employees. The minimum number of open windows that can satisfy the demands in different time periods and seasons can be calculated using the above calculation steps.

4. Staff Scheduling Model

4.1. Parameters and Decision Variables. The opening time of the windows of the tax service hall in Lishui City is 08:30–17:00, divided into the morning shift from 08:30 to 12:00, the noon shift from 12:00 to 14:00, and the afternoon shift from 14:00 to 17:00. To provide high-quality service, each window employee should rest for 15 min after working continuously for 45 min or 1 hr. Let 15 min be one time period; then, the open window time in any one working day can be divided into 27 time periods. 08:30–08:45 is recorded as the first time period and 08:45–09:00 as the second and so on. Then, 11:45–12:00 is recorded as the 14th, 12:00–14:00 as the 15th (consisting of eight times 15 min), and 16:45–17:00 as the 27th time period.

In the tax service hall, there is only one full-time window employee on duty at noon and this window employee does not work from 11:30 to 12:00. Since each employee's working time cannot exceed eight hours in a working day, each employee works for only two of the three shifts in every working day: morning, noon, and afternoon. The difference in the working time of any two full-time window employees is as small as possible in a scheduling cycle. According to the inverted triangle scheduling model, each full-time window employee needs to work at least three and at most four consecutive time periods. Each full-time window employee can be allowed to work for two consecutive time periods only immediately before 12:00 or immediately before 17:00. If a window employee in the morning or afternoon has rested for two consecutive time periods, then he/she will not work again in the morning or afternoon at the window.

The parameters and decision variables of the flexible scheduling model are defined as follows. Let T denote the total days of the scheduling period (usually one month is considered a cycle) and let m and $n - m$ denote the total number of full-time window employees and part-time window employees, respectively. Let c_{jk} denote the minimum number of open windows required to meet the taxpayers service demand for time period k on day j (calculated by the formula (1)); k=1, 5, 9, 13, 16, 20, and 24 correspond to 08:30, 09:30, 10:30, 11:30, 14:00, 15:00, and 16:00, respectively, and $c_{j1} = c_{j2} = c_{j3} = c_{j4}$, $c_{j5} = c_{j6} = c_{j7} = c_{j8}$, $c_{j9} = c_{j10} = c_{j11} = c_{j12}$, $c_{j13} = c_{j14}$, $c_{j15} = 1$, $c_{j16} = c_{j17} = c_{j18} = c_{j19}$, $c_{j20} = c_{j21} = c_{j22} = c_{j23}$, $c_{j24} = c_{j25} = c_{j26} = c_{j27}$. We define also the following decision variables:

$$x_{ijk} = \begin{cases} 1 & \text{if full-time window employee } i \text{ is assigned to work in time period } k \text{ on day } j, \\ & \mathrm{i = 1, 2, \ldots, m;} \\ 0 & \text{otherwise.} \end{cases}$$
$$x_{ijk} = \begin{cases} 1 & \text{if part-time window employee } i \text{ is assigned to work in time period } k \text{ on day } j, \\ & \mathrm{i = m+1, \ldots, n;} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

4.2. Objective Function. In order to meet the requirement that the taxpayers average waiting time does not exceed 15 min, the number of employees is the minimum for the optimal objective; that is, the objective function is

$$\min \quad z = n \tag{3}$$

$$\text{subject to} \quad 2\left(1 - x_{ijk}\right) x_{ij,k+1} \le x_{ij,k+2} + x_{ij,k+3}, \quad \forall i = 1, \ldots, m, \ j = 1, \ldots, T, \ k = 1, \ldots, 11, 16, \ldots, 24,$$
$$2x_{ij1} \le x_{ij2} + x_{ij3}, \quad \forall i = 1, \ldots, m, \ j = 1, \ldots, T \tag{4}$$

$$\sum_{d=k}^{k+4} x_{ijd} \le 4, \quad \forall i = 1, \ldots, m, \ j = 1, \ldots, T, \ k = 1, 2, \ldots, 10, 16, \ldots, 23 \tag{5}$$

$$\begin{aligned}
&x_{ijk}\left(1-x_{ij,k+1}\right)\left(1-x_{ij,k+2}\right) \leq \prod_{n=k+3}^{14}\left(1-x_{ijn}\right), \quad \forall i=1,\ldots,m,\ j=1,\ldots,T,\ k=1,2,\ldots,11 \\
&x_{ijk}\left(1-x_{ij,k+1}\right)\left(1-x_{ij,k+2}\right) \leq \prod_{n=k+3}^{27}\left(1-x_{ijn}\right), \quad \forall i=1,\ldots,m,\ j=1,\ldots,T,\ k=16,\ldots,24
\end{aligned} \tag{6}$$

$$\frac{1}{2}\left(x_{ij13}+x_{ij14}\right) \leq 1-x_{ij15}, \quad \forall i=1,\ldots,m,\ j=1,\ldots,T \tag{7}$$

$$\left[\prod_{k=1}^{14}\left(1-x_{ijk}\right)+\left(1-x_{ij15}\right)+\prod_{k=16}^{27}\left(1-x_{ijk}\right)\right]=2, \quad \forall i=1,\ldots,m,\ j=1,\ldots,T \tag{8}$$

$$x_{ij15} \leq 1-x_{ij,k}, \quad \forall i=1,\ldots,m,\ k=16,\ldots,27 \tag{9}$$

$$\begin{aligned}
&x_{ij14} \leq x_{ij13}, \\
&\left(1-x_{ij12}\right)x_{ij13} \leq x_{ij14}, \\
&\forall i=1,\ldots,m,\ j=1,\ldots,T \\
&\left(1-x_{ij25}\right)x_{ij26} \leq x_{ij27}, \\
&x_{ij27} \leq x_{ij26}, \\
&\forall i=1,\ldots,m,\ j=1,\ldots,T
\end{aligned} \tag{10}$$

$$\begin{aligned}
&\left(1-x_{ij,k}\right)x_{ij,k+1} \leq x_{ij,k+2}, \quad \forall i=m+1,\ldots,n,\ j=1,\ldots,T,\ k=1,2,\ldots,12,16,\ldots,25, \\
&x_{ij1} \leq x_{ij2}, \quad \forall i=m+1,\ldots,n,\ j=1,\ldots,T
\end{aligned} \tag{11}$$

$$\begin{aligned}
&x_{ijk}\left(1-x_{ij,k+1}\right)x_{ij,k+2} \leq x_{ij,k+3}\left[x_{ij,k+4}+\left(1-x_{ij,k+4}\right)\left(1-x_{ij,k+5}\right)\right], \\
&\forall i=m+1,\ldots,n,\ j=1,\ldots,T,\ k=1,2,\ldots,9,16,\ldots,22
\end{aligned} \tag{12}$$

$$\sum_{d=k}^{k+4} x_{ijd} \leq 4, \quad \forall i=m+1,\ldots,n,\ j=1,\ldots,T,\ k=1,2,\ldots,10,16,\ldots,23 \tag{13}$$

$$\begin{aligned}
&x_{ijk}\left(1-x_{ij,k+1}\right)\left(1-x_{ij,k+2}\right) \leq \prod_{n=k+3}^{14}\left(1-x_{ijn}\right), \\
&\forall i=m+1,\ldots,n,\ j=1,\ldots,T,\ k=1,2,\ldots,11, \\
&x_{ijk}\left(1-x_{ij,k+1}\right)\left(1-x_{ij,k+2}\right) \leq \prod_{n=k+3}^{27}\left(1-x_{ijn}\right), \\
&\forall i=m+1,\ldots,n,\ j=1,\ldots,T,\ k=16,\ldots,24
\end{aligned} \tag{14}$$

$$x_{ijk}\left(1-x_{ij,k+1}\right)x_{ij,k+2} \leq x_{ij,k+3}, \quad \forall i=m+1,\ldots,n,\ j=1,\ldots,T,\ k=10,11,23,24 \tag{15}$$

$$\sum_{i=1}^{n} x_{ijk} \geq c_{jk}, \quad \forall j=1,\ldots,T,\ k=1,\ldots,27 \tag{16}$$

$$\sum_{i=1}^{m} x_{ij15}=1, \quad \forall j=1,\ldots,T \tag{17}$$

$$\sum_{j=1}^{T}\sum_{k=1}^{27} x_{ljk}+7\sum_{j=1}^{T} x_{lj15}-\sum_{j=1}^{T}\sum_{k=1}^{27} x_{sjk}-7\sum_{j=1}^{T} x_{sj15} \leq c, \quad \forall l=1,\ldots,m,\ s=1,2,\ldots,m \tag{18}$$

$$x_{ijk} \in \{0,1\}, \quad \forall i=1,\ldots,n,\ j=1,\ldots,T,\ k=1,\ldots,27, \text{ and all variables integer.} \tag{19}$$

The first constraint (4) assures that each full-time window employee works for at least three consecutive time periods (45 min) after he/she has started to work. Constraint (5) assures that each full-time window employee works for at most four consecutive time periods (60 min). Constraint (6) assures that if each full-time window employee has rested for two consecutive time periods in the morning/afternoon, then he/she no longer works at the window in the morning/afternoon. Constraint (7) assures that if a full-time window employee is on duty from 12:00-14:00, then he/she does not work from 11:30 to 12:00; thus, he/she has time to eat lunch and prepare for the noon duty. Constraint (8) assures that each full-time window employee in every working day works only two of the three shifts, morning, noon and afternoon; otherwise, his/her working time is more than eight hours (in violation of labour law). A rule of the tax service hall in Lishui City is that if a full-time window employee is on duty at noon, he/she must also work at the window in the morning. The advantage of the rule is that it makes it convenient for an employee on this duty to take leave, make a business trip, or organize documents in the background in the afternoon. Constraint (9) assures that if a full-time window employee is on duty at noon, then he/she does not work at the window in the afternoon. Constraint (10) assures that each full-time employee is allowed to work for two consecutive time periods before 12:00/17:00. Constraint (11) assures that each part-time window employee has to work for at least two consecutive time periods after he/she has started to work. Constraint (12) assures that if each part-time window employee is on duty for the second time at the window, then he/she has to work for at least two consecutive time periods. Constraint (13) assures that each part-time window employee works for at most four consecutive time periods. Constraint (14) assures that if each part-time window employee has rested for two consecutive time periods in the morning/afternoon, then he/she no longer works at the window in the morning/afternoon. Constraint (15) assures that in the last period of the morning/afternoon shift each part-time window employee is allowed to work only for two consecutive working time periods. Constraint (16) assures that there is a sufficient number of employees to cover the demand for each time period in any working day. In the following algorithm, our optimal solution takes the equality in constraint (16); i.e., under the condition that $\sum_{i=1}^{m}\sum_{j=1}^{T}\sum_{k=1}^{27} x_{ijk} + 7\sum_{i=1}^{m}\sum_{j=1}^{T} x_{ij15} + \sum_{i=m+1}^{n}\sum_{j=1}^{T}\sum_{k=1,k\neq 15}^{27} x_{ijk}$ is the minimum, we obtain the optimal solution. Constraint (17) assures that there is only one full-time window employee on duty at noon on any day j. Constraint (18) assures that the difference in the working hours of any two full-time window employees in one cycle is less than c (nonnegative integer constant) time periods. In order to take fairness among all full-time window employees into account, we usually calculate the best constant c (the difference in the working time of any two full-time window employees is less than or equal to $15c$ min in one cycle).

5. The Algorithm

We decompose the original scheduling model into several submodels and use the descending dimension method to construct the optimal solution of the flexible scheduling model. Before examining the algorithm of the flexible scheduling model, we first repeat the scheduling rules and introduce some scheduling variables. The series of time periods are defined as the vertical direction and the number of windows of each time period is defined as the horizontal direction. When each full-time window employee has started to work, he/she has to work for at least three but not more than four consecutive time periods and then rest for one time period. Each window employee working at the end of the morning/afternoon is allowed to work for two consecutive time periods. When a part-time window employee has started to work, he/she has to work for at least two consecutive time periods. In any fixed working day, there are two variables for any time period. The first variable is the state variable of the time period, which has two values: if the state variable takes 1, it shows that the window employee works at the window and if the state variable takes 0, it shows that the window employee does not work at the window. The second variable is the flag variable of the time period. If the flag variable takes 0, it shows that the working state of the window employee is uncertain and if the flag variable takes 1, it shows that the working state of the window employee is certain. If the flag variable takes 3, it shows that a part-time window employee is working his/her first shift, and there are two time periods from the current state variable to the determined state. (If the flag variable value is 3, it shows that the part-time window employee has worked for two consecutive time periods, and thus, the state of the second time period of the current period is determined.) If the flag variable takes 2, it shows that the state variable has one time period to the determined state. If the flag variable takes 10, it shows that the state variable of the time period previous to the current time period takes 0; the window employee has finished working for consecutive working time periods and has rested for one time period, that is, the first flag variable of the new cycle takes 10 (the state variable of the current time period is uncertain). If the flag variable takes 11, it shows that the state variables of the current time period and the subsequent time period can be assigned only 0. (If the state variables of the two consecutive time periods take 0, the value of the corresponding second flag variable is set to 11; i.e., when the flag variable is set to 11, the window employee no longer works for the remaining time periods of the morning/afternoon.) If the flag variable takes 13, it shows that the current state variable to the determined state has at most two time periods, and if 0 is assigned to the current state variable, then the corresponding flag variable is set to 11, and the state variable of the time period subsequent to the current time period is set to 0 and the corresponding flag variable is set to 11; i.e., the window employee no longer works for the remaining time period of the morning/afternoon, and therefore, the state variable of the time period previous to the current time period is the last working state in the morning/afternoon; if the flag variable takes 13 and 1 is assigned to the current state variable, then the values of the state variable and the flag variable of the time period subsequent to the current time period are set to 0 and 2, respectively. If the flag variable takes 14, it shows that the state variable of the current time period

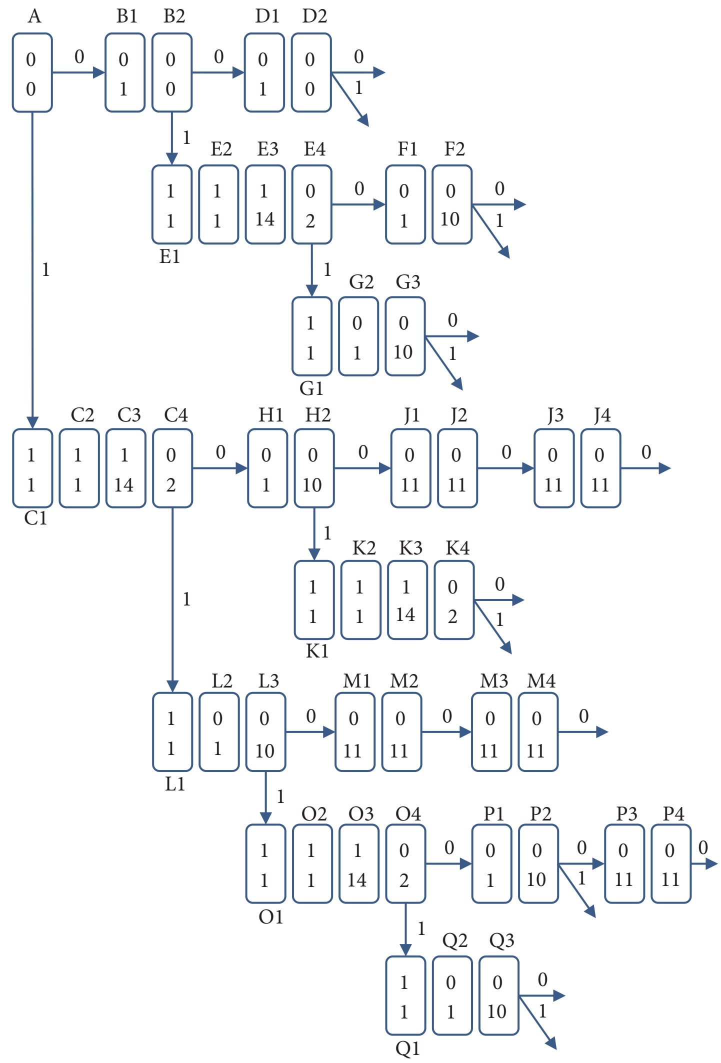

FIGURE 1: Iteration of full-time employee.

does not participate in the horizontal cycle permutation of the combinatorial tuple consisting of 0 and 1. Each full-time window employee's symbol variable is set to green and each part-time window employee's symbol variable is set to red. In order to improve the efficiency of iterative return, we usually set the values of the state variable and the flag variable only of the subsequent two consecutive time periods in the current time period (except the value of flag variable is 14).

Step 1 (initialize variables). The initial state variable and the initial flag variable of each time period of each window employee are all set to 0.

Step 2 (the vertical iterations for flexible scheduling algorithm in the morning). The rule of the iterative scheme is as follows. We first examine the vertical (time periods 1-11) iterative schemes of each full-time window employee. The iterative schemes in Figures 1 and 2 are suitable for time periods 1-11. The digits at the top of all the boxes indicate the values of the state variables of the corresponding time periods, respectively, and the digits at the bottom of all the boxes represent the values of the flag variables of the corresponding time periods, respectively.

First, we introduce the iterative scheme of the state variables and the flag variables in Figure 1 of Appendix A. The algorithm in Figure 1 of Appendix A ensures that constraints (4), (5), and (6) are satisfied. In box A, the initial value of the state variable is 0, and the initial value of the flag variable is 0, which shows that the state variable is not determined. If 0 is assigned to the state variable in box A, which means the value of the state variable in box A has been determined, then the flag variable in box A is set to 1; that is, the values of the state variable and the flag variable in box A are set to corresponding values in box $B1$, respectively. The initial values of the state variable and the flag variable of the time period subsequent to the current time period in box A correspond to the values in box $B2$. If 1 is assigned to the state variable in box $B2$, which means the value of the state variable of the current time period in box $B2$ has been determined,

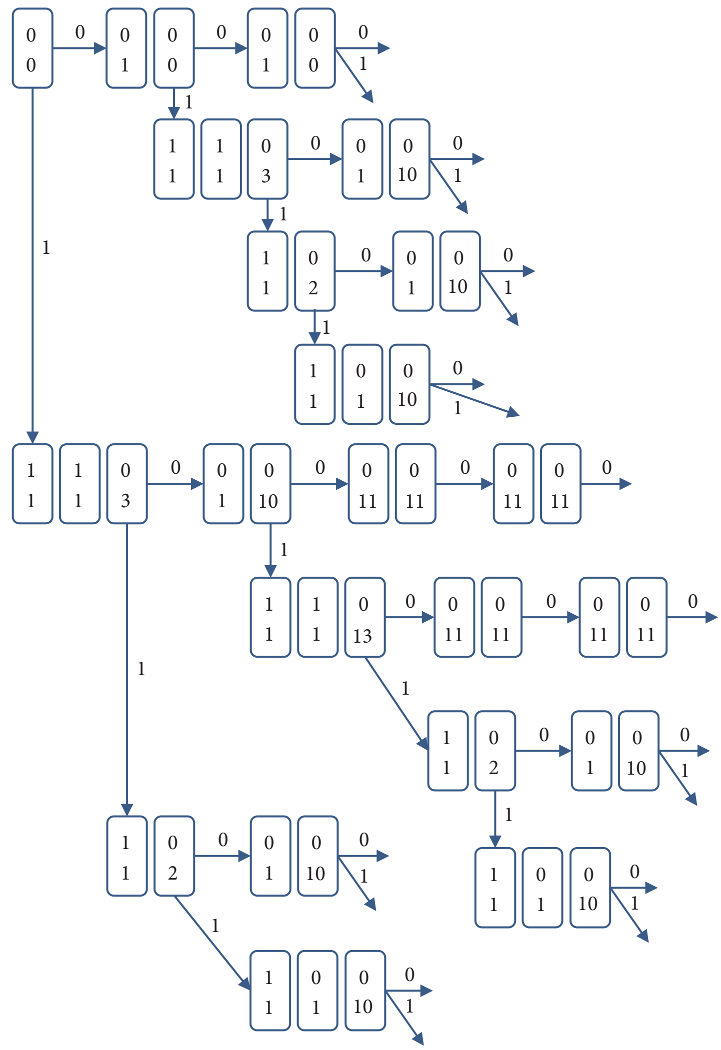

Figure 2: Iteration of part-time employee.

then the corresponding flag variable is set to 1. Since each full-time window employee must work for at least three consecutive time periods, the values of the state variables and the flag variables of the two time periods subsequent to the current period in box $B2$ are respectively 1,1 and 1,14. The values of the state variable and the flag variable of the third time period subsequent to the current time period in box $B2$ are set to 0 and 2, respectively. If 0 is assigned to the state variable in box $E4$ and the state variable of the current time period has been determined, then the flag variable in box $E4$ is set to 1. The values of box $E4$ are changed to the corresponding values of box $F1$; that is, the assignment law from box $E1$ to box $F1$ is divided into the process of working for three consecutive time periods and resting for one time period. The flag variable of the subsequent time period in box $F1$ (i.e., the flag variable in box $F2$) is set to 10 (the end of a working cycle; the initial value of the state variable for the new cycle is uncertain). If 1 is assigned to the state variable in box $E4$, then the flag variable of the current time period in box $E4$ (i.e., the current time period in box $G1$) is set to 1, and a scheduling cycle has ended and the window employee has to rest for one time period; therefore, the values of the state variable and the flag variable of the current time period in box $G2$ are set to 0 and 1, respectively. The values of the state variable and the flag variable of the current time period in box $G3$ are 0 and 10, respectively. The assignment law of the values of the state variables and the flag variables in boxes $C1--H2$ ($C1--L3$) is consistent with the assignment law of the values of the corresponding variables in boxes $E1--F2$ ($E1--G3$). If 0 is assigned to the state variable in box $L3$, we note that the value of the state variable in box $L2$ is 0, and the time periods in box $L2$ and box $L3$ are two adjacent time periods; i.e., the window employee has rested for two consecutive time periods, and therefore, the values of the state variables of the subsequent two time periods in box $L2$ (i.e., the current time periods in box $M1$ and $M2$) are all set to 0 and their corresponding flag variables are all set to 11. The window employee has rested for two consecutive time periods in the morning/afternoon, so he/she no longer needs to work for the remaining time periods in the morning/afternoon; thus, the

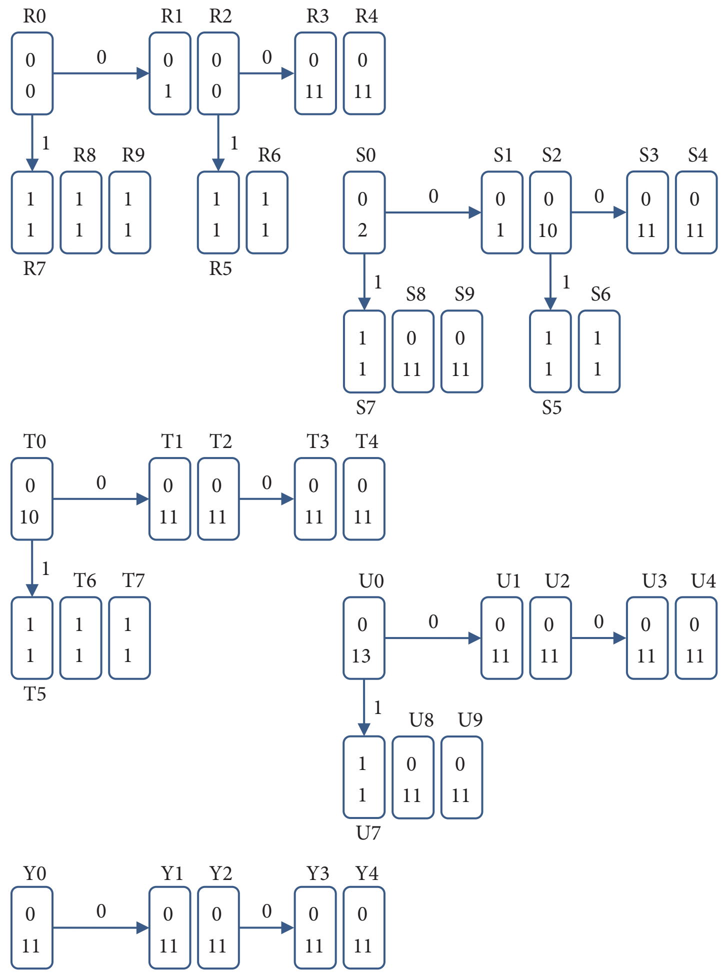

Figure 3: Iteration of time period 12.

state variable in box $M3$ is set only to 0 and the corresponding flag variable is set to 11. If 1 is assigned to the state variable in box $L3$, then the flag variable in box $L3$ is set to 1 and the state variables of the two time periods subsequent to the current time period in box $L3$ are all set to 1, and thus, their corresponding flag variables are set to 1 and 14, respectively; that is, the values of the state variables in boxes $O1--O3$ are all set to 1 and the values of the flag variables in box $O2$ and $O3$ are set to 1 and 14, respectively. Other cases are similar to the above logical reasoning. The scheduling iterative schemes in Figure 2 of Appendix A are similar to those in Figure 1. The algorithm in Figures 1 and 2 of Appendix A guarantees that constraints (11), (12), (13), and (14) are satisfied.

The following algorithm guarantees that constraints (10) and (15) are satisfied. The iterative schemes in Figure 3 of Appendix A are described as follows. When the iterative schemes are executed for time period 12 and the corresponding initial values of the state variable and the flag variable of time period 12 are 0 and 0, respectively, if 1 is assigned to the state variable of time period 12, the window employee must continuously work for time periods 13 and 14, and then, the values of the state variables of time periods 13 and 14 are all set to 1; otherwise, there is a window employee who works only at time period 14, which does not satisfy the flexible scheduling model. If 0 is assigned to the state variable of time period 12, then the state variables of time periods 13 and 14 are set to 0 or 1 at the same time; otherwise, there is a window employee who works only at time period 14, which does not satisfy the flexible scheduling model. When the corresponding values of the state variable and the flag variable of time period 12 are 0 and 2, respectively, and the value of the flag variable of time period 12 appears as 2, this shows that the window employee has been working for three consecutive time periods. If 1 is assigned to the state variable of time period 12, then the state variables of time periods 13 and 14 must be set to 0; otherwise, there is a window employee working only for one time period, which does not meet the flexible scheduling model. If 0 is assigned to the state variable of time period 12, then the state variables of time periods 13 and 14 are taken as 0 or 1 at the same time; the values of the flag variables are determined by the values of the state variables (the assignment law of the flag variable is shown in Figure 3). When the corresponding values of the state variable and the flag variable of time period 12 are 0 and 10, respectively, and the value of the flag variable of time period 12 appears as 10, then the value of the state variable of time period 11 must be set to 0. If 0 is assigned to the state variable of time period 12, the state variables of time periods 11 and 12 have continuously taken value 0, and thus the state variable and the flag variable of time period 12 must be set to 0 and 11, respectively (if the window employee has rested for two consecutive time periods, then he/she no longer works for the remaining time periods of the morning). If 1 is assigned to the state variable of time period 12, the state variables of time periods 13 and 14 must be set to 1; otherwise, there is a window employee who works only at time period 14. When the corresponding values of the state variable and the flag variable of time period 12 are 0 and 13, respectively, and the value of the flag variable of time period 12 appears as 13, the state variables of time periods 10 and 11 have continuously taken value 1. If 0 is assigned to the state variable of time period 12, then the values of the state variables of time periods 13 and 14 must be all set to 0; i.e., when the value of the flag variable is 13, if 0 is assigned to the state variable of time period 12, the state variable of time period 13 must be set to 0, which means the employee's work at the window in the morning has been completed, that is, in his/her last shift of the morning the employee is allowed to work only for two consecutive time periods (an employee working in the noon shift is not allowed to work only for two consecutive time periods). If 1 is assigned to the state variable of time period 12, then the window employee has been working for three consecutive time periods, and thus the state variables of time periods 13 and 14 must all be set to 0; otherwise, there is a window employee who works only at time period 14, which does not satisfy the constraints of the flexible scheduling model. The algorithm in Figure 2 ensures that constraints (11), (12), (14), and (15) are satisfied. Other cases of the iterative schemes in Figure 3 are clear.

The following algorithm guarantees that constraints (10) and (15) are satisfied. When the corresponding values of the state variable and the flag variable of time period 12 are 0 and 3, respectively, the part-time window employee has been working for two consecutive time periods. If 0 is assigned to the state variable of time period 12 (each part-time window employee is allowed to work for two consecutive time periods), then the values of the state variables of time periods 13 and 14 are all set to 0 or 1 at the same time. If 1 is assigned to the state variable of time period 12, then the state variables of time periods 13 and 14 must all be set to 0; otherwise, there is a part-time window employee who only works for one time period in time periods 13/14 (see Figure 4 of Appendix A).

We now study the iterative scheme of time periods 13 and 14 shown in Figure 4. When the state variable and the flag variable of time period 13 take the corresponding values 0 and 13, respectively, it shows that the state variables of time periods 11 and 12 have been continuously taken as value 1. Then, the state variables of time periods 13 and 14 must be set to 0 or 1 at the same time; otherwise, there are some part-time window employees who work only for one time period in time periods 13/14, which does not satisfy the constraints of flexible scheduling model. Other scheduling iterative schemes in Figure 4 are similar to the above discussions.

Step 3 (the horizontal iterations of the flexible scheduling algorithm in the morning). The following algorithm ensures that constraint (16) is satisfied and the objective function (3) reaches the minimum value. Take $C_j = \max(c_{j1}, c_{j2}, \ldots, c_{j27})$, and without loss of generality, let $m = C_j$ be the number of full-time window employees. In order to compute the minimum value of n, the number of part-time window employee is taken first as 1. If this does not satisfy the subsequent iteration, then we add one part-time window employee to participate in the iterative scheduling, until we have constructed the optimal solution. Take the number of the open windows from time period 1 to time period 4, that is, c_{j1}. We select any c_{j1} window employees from the $m + 1$ window employees, and the c_{j1} window employees begin to work at the window for time period 1; i.e., the values of the state variables of the c_{j1} window employees of time period 1 are all set to 1. The assignment law of the state variables from time period 3 to time period 14 is to execute the horizontal cyclic permutation of the all combinatorial tuples of numbers consisting of 0 and 1. For example, we assume that the maximum number of open windows is 3 and let the number of the window employees be 4, which can be the initial value of the iteration. Assume that the number of open windows in time period 1 is 2; then, the total number of the four tuple consisting of two 0 and two 1 is the combinatorial number $\binom{4}{2} = 6$, i.e., the following six tuples: tuple A: 1, 1, 0, 0; tuple B: 1,0, 1, 0; tuple C: 0, 1, 1, 0; tuple D: 1, 0, 0, 1; tuple E: 0, 1, 0, 1; and tuple F: 0, 0, 1, 1. If the symbol variable is green, then the iterative schemes from time period 3 to time period 11 are executed iteratively, as shown in Figure 1; if the symbol variable is red, then the iterative schemes from time period 3 to time period 11 are executed iteratively, as shown in Figure 2. When the value of the flag variable appears as 13, as shown in Figure 3, this iteration is suitable only for the state variable of each part-time window employee. The other iterations in Figures 3 and 4 are appropriate for the state variables of the part-time and full-time window employees from time period 12 to time period 14. The assignment law of the state variables of time period 1 takes tuple A and we perform a vertical iteration according to the iteration rules in Figures 1–4; if the iterative chain does not satisfy a constraint condition of the scheduling model, then tuple A is replaced by tuple B, until all the tuples are used. If all the tuples are used, and the optimal solution is not found, the number of part-time window employees is increased to perform new iterations until the optimal solution is found. The flexible scheduling of the afternoon windows is similar to the flexible scheduling of the morning windows.

Step 4 (the daily flexible scheduling model). The following algorithm ensures that constraints (8), (9), and (17) are satisfied. The numbers of working time periods of the window employees in the morning/afternoon scheduling model are sorted from large to small, combining the morning scheduling model and the afternoon scheduling model to obtain the daily scheduling model, which is called the inverted triangle scheduling model. If it is arranged that a full-part window employee is on duty at

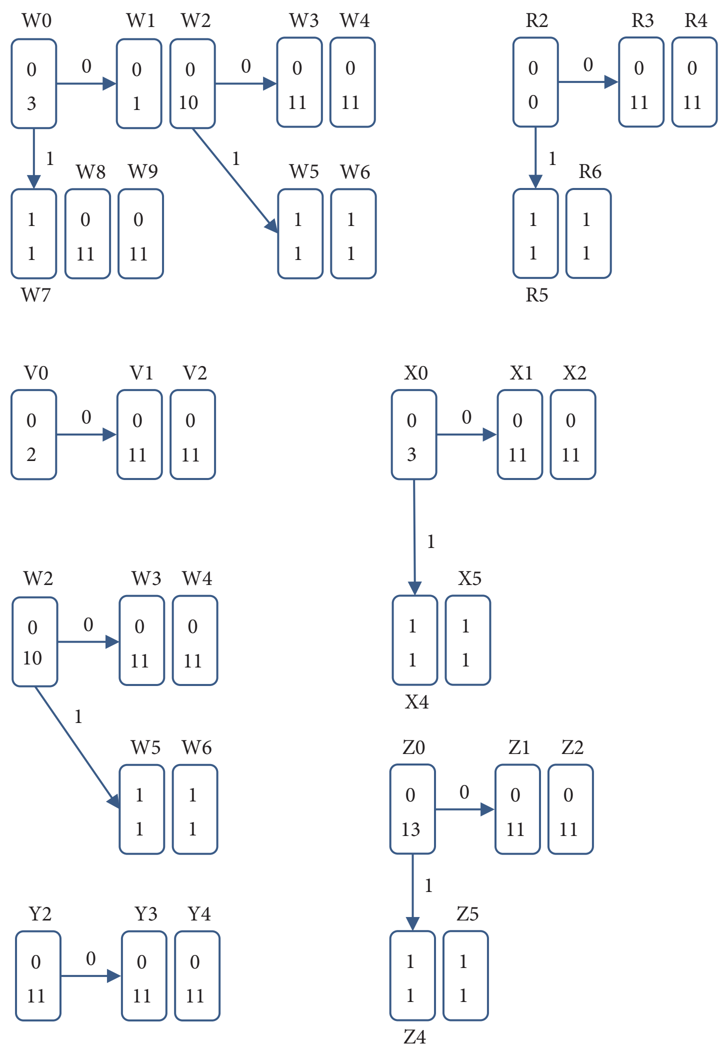

FIGURE 4: Iteration of time period 13.

noon, taking into account the fact that he/she can only be on duty in the morning and cannot be on duty from 11:30 to 12:00, the search to meet these conditions begins from the last employee according to the inverted triangle scheduling model. If a full-time window employee is on duty in the morning and not on duty at the window from 11:30 to 12:00, he/she is assigned to work at noon. If a full-time window employee is also on duty in the afternoon, then the employee's afternoon shift is replaced by the next employee, and so on. If the daily scheduling model has searched all the window employees and the scheduling model does not satisfy these conditions, we add a window employee to finish the afternoon shift.

Step 5. The following algorithm guarantees that constraint (18) is satisfied. According to the above four steps, we can compute the daily working patterns of the window employees, as well as the number of working time periods. Denote by $t_{1j}, t_{2j}, \ldots, t_{mj}$ the first m values for the jth working day, $j = 1, 2, \ldots, T$. We also want the values of the window working time periods of any working day to be sorted from large to small; hence, let $t_{1j} \geq t_{2j}, \ldots, \geq t_{mj}$, $j = 1, \ldots, T$. We define the following 0-1 variables.

$$y_{ilj} = \begin{cases} 1 & \text{if the number of working time periods of the window employee } i \text{ on day } j \text{ is } t_{lj}, \\ & i = 1, 2, \ldots, m, \ \ l = 1, 2, \ldots, m, \ \ j = 1, 2, \ldots, T; \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

Table 1: Number of taxpayers arriving at the tax service hall per time period.

Time period	Third week in January, average number of taxpayers arriving at the tax service hall	Fourth week in January, average number of taxpayers arriving at the tax service hall
08:30–09:30	45	41
09:30–10:30	55	43
10:30–11:30	48	37
11:30–12:00	12	9
12:00–14:00	5	3
14:00–15:00	42	44
15:00–16:00	56	52
16:00–17:00	40	35

Table 2: Number of open windows.

Time period	Number of open windows in first week of January	Number of open windows in second week of January
08:30–9:30	5	5
09:30–10:30	6	5
10:30–11:30	5	4
11:30–12:00	3	3
12:00–14:00	1	1
14:00–15:00	5	5
15:00–16:00	6	6
16:00–17:00	5	4

Objective function is

$$\min c \tag{21}$$

The constraint conditions are as follows:

$$\begin{gathered}\sum_{j=1}^{T}\sum_{l=1}^{m} t_{lj} y_{ilj} - \sum_{j=1}^{T}\sum_{l=1}^{m} t_{lj} y_{slj} \le c, \\ \forall i = 1,\dots,m,\ s = 1,\dots,m, \\ \sum_{l=1}^{m} y_{ilj} = 1, \\ \sum_{l=1}^{m} y_{slj} = 1, \\ \forall i = 1,\dots,m,\ s = 1,\dots,m,\ j = 1,\dots,T.\end{gathered} \tag{22}$$

First, we select $c = 0$. If all the constraint conditions are satisfied, then $c = 0$ is the best constant; if a constraint condition is not satisfied, the value of c is increased by 1 until all the constraint conditions are satisfied, and then, the calculated c is the best constant.

6. Illustrative Example

Clearly, the above model is a nonlinear 0-1 integer programming model, and nonlinear 0-1 integer programming is an NPC problem, without a guarantee that the optimal solution will be calculated in polynomial time. The main difference and difficulty here in the scheduling model are high nonlinearity and complexity, as well as the large number of variables involved in the model, and thus, linear programming software usually cannot be used to obtain the optimal solution to the 0-1 integer programming model established by us.

The source of our data was the second branch of the local taxation bureau in Lishui City. According to the historical data of this branch, we predicted the data of the last two weeks in January 2016. In the past, whether slack season or busy season, the second branch of the local taxation bureau has always opened six windows. In order to save space, we take two weeks as a cycle (see Table 1 in Appendix B).

By using the integrated management platform of the second branch of the local taxation bureau, we can obtain that the average time each window employee takes to handle one taxpayer's business is 5.6 min and the average waiting time for taxpayers is less than 15 min. The number of the open windows in each time period of the first two weeks of January is computed by the $M/M/c$ model of queuing theory (see Table 2 in Appendix B).

The digit 1 represents a window employee who works at the window and the digit 0 represents a window employee who works in the background or rests (see Tables 3 and 4 in Appendix B). In the example, we can see that, as compared with manual scheduling, the flexible scheduling model proposed in this paper has the following advantages:

Table 3: Flexible scheduling of the window personnel.

Day	January 18th, 2016, Monday									
Personnel	A_2	A_1	A_4	A_5	A_3	A_6	A_7	A_8	A_9	Number of windows
08:30–08:45	1	1	1	1	0	1	0	0	0	5
08:45–09:00	1	1	1	1	0	1	0	0	0	5
09:00–09:15	1	1	1	1	0	1	0	0	0	5
09:15–09:30	1	1	0	0	1	0	0	1	1	5
09:30–09:45	0	0	1	1	1	1	0	1	1	6
09:45–10:00	1	1	1	1	1	1	0	0	0	6
10:00–10:15	1	1	1	1	0	1	1	0	0	6
10:15–10:30	1	1	1	1	1	0	1	0	0	6
10:30–10:45	1	1	0	0	1	1	1	0	0	5
10:45–11:00	0	0	1	1	1	1	1	0	0	5
11:00–11:15	1	1	1	1	0	1	0	0	0	5
11:15–11:30	1	1	1	1	1	0	0	0	0	5
11:30–11:45	1	1	0	0	1	0	0	0	0	3
11:45–12:00	1	1	0	0	1	0	0	0	0	3
12:00–14:00	0	0	0	0	0	1	0	0	0	1
14:00–14:15	1	1	1	1	1	0	0	0	0	5
14:15–14:30	1	1	1	1	1	0	0	0	0	5
14:30–14:45	1	1	1	1	1	0	0	0	0	5
14:45–15:00	1	1	0	0	0	0	1	1	1	5
15:00–15:15	0	0	1	1	1	0	1	1	1	6
15:15–15:30	1	1	1	1	1	0	1	0	0	6
15:30–15:45	1	1	1	1	1	0	0	1	0	6
15:45–16:00	1	1	1	1	0	0	1	1	0	6
16:00–16:15	1	1	0	0	1	0	1	1	0	5
16:15–16:30	0	0	1	1	1	0	1	1	0	5
16:30–16:45	1	1	1	1	1	0	0	0	0	5
16:45–17:00	1	1	1	1	1	0	0	0	0	5
Total	22	22	20	20	19	17	10	8	4	
Day	January 19th, 2016, Tuesday									
Personnel	A_2	A_5	A_1	A_4	A_6	A_3	A_7	A_8	A_9	Number of windows
08:30–08:45	1	1	1	1	0	1	0	0	0	5
08:45–09:00	1	1	1	1	0	1	0	0	0	5
09:00–09:15	1	1	1	1	0	1	0	0	0	5
09:15–09:30	1	1	0	0	1	0	0	1	1	5
09:30–09:45	0	0	1	1	1	1	0	1	1	6
09:45–10:00	1	1	1	1	1	1	0	0	0	6
10:00–10:15	1	1	1	1	0	1	1	0	0	6
10:15–10:30	1	1	1	1	1	0	1	0	0	6
10:30–10:45	1	1	0	0	1	1	1	0	0	5
10:45–11:00	0	0	1	1	1	1	1	0	0	5
11:00–11:15	1	1	1	1	0	1	0	0	0	5
11:15–11:30	1	1	1	1	1	0	0	0	0	5
11:30–11:45	1	1	0	0	1	0	0	0	0	3
11:45–12:00	1	1	0	0	1	0	0	0	0	3
12:00–14:00	0	0	0	0	0	1	0	0	0	1
14:00–14:15	1	1	1	1	1	0	0	0	0	5
14:15–14:30	1	1	1	1	1	0	0	0	0	5
14:30–14:45	1	1	1	1	1	0	0	0	0	5
14:45–15:00	1	1	0	0	0	0	1	1	1	5
15:00–15:15	0	0	1	1	1	0	1	1	1	6
15:15–15:30	1	1	1	1	1	0	1	0	0	6

Table 3: Continued.

15:30–15:45	1	1	1	1	1	0	0	1	0	6
15:45–16:00	1	1	1	1	0	0	1	1	0	6
16:00–16:15	1	1	0	0	1	0	1	1	0	5
16:15-16:30	0	0	1	1	1	0	1	1	0	5
16:30–16:45	1	1	1	1	1	0	0	0	0	5
16:45–17:00	1	1	1	1	1	0	0	0	0	5
Total	22	22	20	20	19	17	10	8	4	
Day	January 20th, 2016, Wednesday									
Personnel	A_6	A_4	A_3	A_2	A_1	A_5	A_7	A_8	A_9	Number of windows
08:30–08:45	1	1	1	1	0	1	0	0	0	5
08:45–09:00	1	1	1	1	0	1	0	0	0	5
09:00–09:15	1	1	1	1	0	1	0	0	0	5
09:15–09:30	1	1	0	0	1	0	0	1	1	5
09:30–09:45	0	0	1	1	1	1	0	1	1	6
09:45–10:00	1	1	1	1	1	1	0	0	0	6
10:00–10:15	1	1	1	1	0	1	1	0	0	6
10:15–10:30	1	1	1	1	1	0	1	0	0	6
10:30–10:45	1	1	0	0	1	1	1	0	0	5
10:45–11:00	0	0	1	1	1	1	1	0	0	5
11:00–11:15	1	1	1	1	0	1	0	0	0	5
11:15–11:30	1	1	1	1	1	0	0	0	0	5
11:30–11:45	1	1	0	0	1	0	0	0	0	3
11:45–12:00	1	1	0	0	1	0	0	0	0	3
12:00–14:00	0	0	0	0	0	1	0	0	0	1
14:00–14:15	1	1	1	1	1	0	0	0	0	5
14:15–14:30	1	1	1	1	1	0	0	0	0	5
14:30–14:45	1	1	1	1	1	0	0	0	0	5
14:45–15:00	1	1	0	0	0	0	1	1	1	5
15:00–15:15	0	0	1	1	1	0	1	1	1	6
15:15–15:30	1	1	1	1	1	0	1	0	0	6
15:30–15:45	1	1	1	1	1	0	0	1	0	6
15:45–16:00	1	1	1	1	0	0	1	1	0	6
16:00–16:15	1	1	0	0	1	0	1	1	0	5
16:15–16:30	0	0	1	1	1	0	1	1	0	5
16:30–16:45	1	1	1	1	1	0	0	0	0	5
16:45–17:00	1	1	1	1	1	0	0	0	0	5
Total	22	22	20	20	19	17	10	8	4	
Day	January 21st, 2016, Thursday									
Personnel	A_5	A_6	A_4	A_1	A_3	A_2	A_7	A_8	A_9	Number of windows
08:30–08:45	1	1	1	1	0	1	0	0	0	5
08:45–09:00	1	1	1	1	0	1	0	0	0	5
09:00–09:15	1	1	1	1	0	1	0	0	0	5
09:15–09:30	1	1	0	0	1	0	0	1	1	5
09:30–09:45	0	0	1	1	1	1	0	1	1	6
09:45–10:00	1	1	1	1	1	1	0	0	0	6
10:00–10:15	1	1	1	1	0	1	1	0	0	6
10:15–10:30	1	1	1	1	1	0	1	0	0	6
10:30–10:45	1	1	0	0	1	1	1	0	0	5
10:45–11:00	0	0	1	1	1	1	1	0	0	5
11:00–11:15	1	1	1	1	0	1	0	0	0	5
11:15–11:30	1	1	1	1	1	0	0	0	0	5
11:30–11:45	1	1	0	0	1	0	0	0	0	3

Table 3: Continued.

11:45–12:00	1	1	0	0	1	0	0	0	0	3
12:00–14:00	0	0	0	0	0	1	0	0	0	1
14:00–14:15	1	1	1	1	1	0	0	0	0	5
14:15–14:30	1	1	1	1	1	0	0	0	0	5
14:30–14:45	1	1	1	1	1	0	0	0	0	5
14:45–15:00	1	1	0	0	0	0	1	1	1	5
15:00–15:15	0	0	1	1	1	0	1	1	1	6
15:15–15:30	1	1	1	1	1	0	1	0	0	6
15:30–15:45	1	1	1	1	1	0	0	1	0	6
15:45–16:00	1	1	1	1	0	0	1	1	0	6
16:00–16:15	1	1	0	0	1	0	1	1	0	5
16:15–16:30	0	0	1	1	1	0	1	1	0	5
16:30–16:45	1	1	1	1	1	0	0	0	0	5
16:45–17:00	1	1	1	1	1	0	0	0	0	5
Total	22	22	20	20	19	17	10	8	4	
Day	January 22nd, 2016, Friday									
Personnel	A_1	A_4	A_5	A_6	A_3	A_2	A_7	A_8	A_9	Number of windows
08:30–08:45	1	1	1	1	0	1	0	0	0	5
08:45–09:00	1	1	1	1	0	1	0	0	0	5
09:00–09:15	1	1	1	1	0	1	0	0	0	5
09:15–09:30	1	1	0	0	1	0	0	1	1	5
09:30–09:45	0	0	1	1	1	1	0	1	1	6
09:45–10:00	1	1	1	1	1	1	0	0	0	6
10:00–10:15	1	1	1	1	0	1	1	0	0	6
10:15–10:30	1	1	1	1	1	0	1	0	0	6
10:30–10:45	1	1	0	0	1	1	1	0	0	5
10:45–11:00	0	0	1	1	1	1	1	0	0	5
11:00–11:15	1	1	1	1	0	1	0	0	0	5
11:15–11:30	1	1	1	1	1	0	0	0	0	5
11:30–11:45	1	1	0	0	1	0	0	0	0	3
11:45–12:00	1	1	0	0	1	0	0	0	0	3
12:00–14:00	0	0	0	0	0	1	0	0	0	1
14:00–14:15	1	1	1	1	1	0	0	0	0	5
14:15–14:30	1	1	1	1	1	0	0	0	0	5
14:30–14:45	1	1	1	1	1	0	0	0	0	5
14:45–15:00	1	1	0	0	0	0	1	1	1	5
15:00–15:15	0	0	1	1	1	0	1	1	1	6
15:15–15:30	1	1	1	1	1	0	1	0	0	6
15:30–15:45	1	1	1	1	1	0	0	1	0	6
15:45–16:00	1	1	1	1	0	0	1	1	0	6
16:00–16:15	1	1	0	0	1	0	1	1	0	5
16:15–16:30	0	0	1	1	1	0	1	1	0	5
16:30–16:45	1	1	1	1	1	0	0	0	0	5
16:45–17:00	1	1	1	1	1	0	0	0	0	5
Total	22	22	20	20	19	17	10	8	4	
Day	January 25th, 2016, Monday									
Personnel	A_3	A_2	A_5	A_6	A_4	A_1	A_7	A_8	A_9	Number of windows
08:30–08:45	1	1	1	0	1	0	0	0	1	5
08:45–09:00	1	1	1	0	1	0	0	0	1	5
09:00–09:15	1	1	1	0	1	1	0	0	0	5

Table 3: Continued.

09:15–09:30	1	0	1	1	0	1	1	0	0	5
09:30–09:45	0	1	0	1	1	1	1	0	0	5
09:45–010:00	1	1	1	1	1	0	0	0	0	5
10:00–10:15	1	1	1	0	1	1	0	0	0	5
10:15–10:30	1	0	1	1	1	1	0	0	0	5
10:30–10:45	1	1	0	1	0	1	0	0	0	4
10:45–11:00	0	1	1	1	1	0	0	0	0	4
11:00–11:15	1	1	1	0	1	0	0	0	0	4
11:15–11:30	1	0	1	1	1	0	0	0	0	4
11:30–11:45	1	1	0	1	0	0	0	0	0	3
11:45–12:00	1	1	0	1	0	0	0	0	0	3
12:00–14:00	0	0	0	0	0	1	0	0	0	1
14:00–14:15	1	1	1	1	1	0	0	0	0	5
14:15–14:30	1	1	1	1	1	0	0	0	0	5
14:30–14:45	1	1	1	1	1	0	0	0	0	5
14:45–15:00	1	1	0	0	0	0	1	1	1	5
15:00–15:15	0	0	1	1	1	0	1	1	1	6
15:15–15:30	1	1	1	1	1	0	1	0	0	6
15:30–15:45	1	1	1	1	1	0	0	1	0	6
15:45–16:00	1	1	1	1	0	0	1	1	0	6
16:00–16:15	1	1	0	0	1	0	1	0	0	4
16:15–16:30	0	0	1	1	1	0	1	0	0	4
16:30–16:45	1	1	1	1	0	0	0	0	0	4
16:45–17:00	1	1	1	1	0	0	0	0	0	4
Total	22	21	20	19	18	14	8	4	4	
Day	January 26th, 2016, Tuesday									
Personnel	A_2	A_3	A_4	A_6	A_1	A_5	A_7	A_8	A_9	Number of windows
08:30–08:45	1	1	1	0	1	0	0	0	1	5
08:45–09:00	1	1	1	0	1	0	0	0	1	5
09:00–09:15	1	1	1	0	1	1	0	0	0	5
09:15–09:30	1	0	1	1	0	1	0	1	0	5
09:30–09:45	0	1	0	1	1	1	0	1	0	5
9:45–10:00	1	1	1	1	1	0	0	0	0	5
10:00–10:15	1	1	1	0	1	1	0	0	0	5
10:15–10:30	1	0	1	1	1	1	0	0	0	5
10:30–10:45	1	1	0	1	0	1	0	0	0	4
10:45–11:00	0	1	1	1	1	0	0	0	0	4
11:00–11:15	1	1	1	0	1	0	0	0	0	4
11:15–11:30	1	0	1	1	1	0	0	0	0	4
11:30–11:45	1	1	0	1	0	0	0	0	0	3
11:45–12:00	1	1	0	1	0	0	0	0	0	3
12:00–14:00	0	0	0	0	0	1	0	0	0	1
14:00–14:15	1	1	1	1	1	0	0	0	0	5
14:15–14:30	1	1	1	1	1	0	0	0	0	5
14:30–14:45	1	1	1	1	1	0	0	0	0	5
14:45–15:00	1	1	0	0	0	0	1	1	1	5
15:00–15:15	0	0	1	1	1	0	1	1	1	6
15:15–15:30	1	1	1	1	1	0	1	0	0	6
15:30–15:45	1	1	1	1	1	0	0	1	0	6
15:45–16:00	1	1	1	1	0	0	1	1	0	6
16:00–16:15	1	1	0	0	1	0	1	0	0	4
16:15–16:30	0	0	1	1	1	0	1	0	0	4

Table 3: Continued.

16:30–16:45	1	1	1	1	0	0	0	0	0	4
16:45–17:00	1	1	1	1	0	0	0	0	0	4
Total	22	21	20	19	18	14	6	6	4	
Day	January 27th, 2016, Wednesday									
Personnel	A_3	A_1	A_5	A_6	A_4	A_2	A_7	A_8	A_9	Number of windows
08:30–08:45	1	1	1	0	1	0	0	1	0	5
08:45–09:00	1	1	1	0	1	0	0	1	0	5
09:00–09:15	1	1	1	0	1	1	0	0	0	5
09:15–09:30	1	0	1	1	0	1	0	1	0	5
09:30–09:45	0	1	0	1	1	1	0	1	0	5
09:45–10:00	1	1	1	1	1	0	0	0	0	5
10:00–10:15	1	1	1	0	1	1	0	0	0	5
10:15–10:30	1	0	1	1	1	1	0	0	0	5
10:30–10:45	1	1	0	1	0	1	0	0	0	4
10:45–11:00	0	1	1	1	1	0	0	0	0	4
11:00–11:15	1	1	1	0	1	0	0	0	0	4
11:15–11:30	1	0	1	1	1	0	0	0	0	4
11:30–11:45	1	1	0	1	0	0	0	0	0	3
11:45–12:00	1	1	0	1	0	0	0	0	0	3
12:00–14:00	0	0	0	0	0	1	0	0	0	1
14:00–14:15	1	1	1	1	1	0	0	0	0	5
14:15–14:30	1	1	1	1	1	0	0	0	0	5
14:30–14:45	1	1	1	1	1	0	0	0	0	5
14:45–15:00	1	1	0	0	0	0	1	1	1	5
15:00–15:15	0	0	1	1	1	0	1	1	1	6
15:15–15:30	1	1	1	1	1	0	1	0	0	6
15:30–15:45	1	1	1	1	1	0	0	1	0	6
15:45–16:00	1	1	1	1	0	0	1	1	0	6
16:00–16:15	1	1	0	0	1	0	1	0	0	4
16:15–16:30	0	0	1	1	1	0	1	0	0	4
16:30–16:45	1	1	1	1	0	0	0	0	0	4
16:45–17:00	1	1	1	1	0	0	0	0	0	4
Total	22	21	20	19	18	14	6	8	2	
Day	January 28th, 2016, Thursday									
Personnel	A_3	A_2	A_6	A_1	A_5	A_4	A_7	A_8	A_9	Number of windows
08:30–08:45	1	1	1	0	1	0	1	0	0	5
08:45–09:00	1	1	1	0	1	0	1	0	0	5
09:00–09:15	1	1	1	0	1	1	0	0	0	5
09:15–09:30	1	0	1	1	0	1	0	0	1	5
09:30–09:45	0	1	0	1	1	1	0	0	1	5
09:45–10:00	1	1	1	1	1	0	0	0	0	5
10:00–10:15	1	1	1	0	1	1	0	0	0	5
10:15–10:30	1	0	1	1	1	1	0	0	0	5
10:30–10:45	1	1	0	1	0	1	0	0	0	4
10:45–11:00	0	1	1	1	1	0	0	0	0	4
11:00–11:15	1	1	1	0	1	0	0	0	0	4
11:15–11:30	1	0	1	1	1	0	0	0	0	4
11:30–11:45	1	1	0	1	0	0	0	0	0	3
11:45–12:00	1	1	0	1	0	0	0	0	0	3
12:00–14:00	0	0	0	0	0	1	0	0	0	1
14:00–14:15	1	1	1	1	1	0	0	0	0	5

Table 3: Continued.

14:15–14:30	1	1	1	1	1	0	0	0	0	5
14:30–14:45	1	1	1	1	1	0	0	0	0	5
14:45–15:00	1	1	0	0	0	0	1	1	1	5
15:00–15:15	0	0	1	1	1	0	1	1	1	6
15:15–15:30	1	1	1	1	1	0	1	0	0	6
15:30–15:45	1	1	1	1	1	0	0	1	0	6
15:45–16:00	1	1	1	1	0	0	1	1	0	6
16:00–16:15	1	1	0	0	1	0	1	0	0	4
16:15–16:30	0	0	1	1	1	0	1	0	0	4
16:30–16:45	1	1	1	1	0	0	0	0	0	4
16:45–17:00	1	1	1	1	0	0	0	0	0	4
Total	22	21	20	19	18	14	8	4	4	
Day	January 29th, 2016, Friday									
Personnel	A_5	A_4	A_1	A_2	A_6	A_3	A_7	A_8	A_9	Number of windows
08:30–08:45	1	1	1	0	1	0	0	0	1	5
08:45–09:00	1	1	1	0	1	0	0	0	1	5
09:00–09:15	1	1	1	0	1	1	0	0	0	5
09:15–09:30	1	0	1	1	0	1	1	0	0	5
09:30–09:45	0	1	0	1	1	1	1	0	0	5
09:45–10:00	1	1	1	1	1	0	0	0	0	5
10:00–10:15	1	1	1	0	1	1	0	0	0	5
10:15–10:30	1	0	1	1	1	1	0	0	0	5
10:30–10:45	1	1	0	1	0	1	0	0	0	4
10:45–11:00	0	1	1	1	1	0	0	0	0	4
11:00–11:15	1	1	1	0	1	0	0	0	0	4
11:15–11:30	1	0	1	1	1	0	0	0	0	4
11:30–11:45	1	1	0	1	0	0	0	0	0	3
11:45–12:00	1	1	0	1	0	0	0	0	0	3
12:00–14:00	0	0	0	0	0	1	0	0	0	1
14:00–14:15	1	1	1	1	1	0	0	0	0	5
14:15–14:30	1	1	1	1	1	0	0	0	0	5
14:30–14:45	1	1	1	1	1	0	0	0	0	5
14:45–15:00	1	1	0	0	0	0	1	1	1	5
15:00–15:15	0	0	1	1	1	0	1	1	1	6
15:15–15:30	1	1	1	1	1	0	1	0	0	6
15:30–15:45	1	1	1	1	1	0	0	1	0	6
15:45–16:00	1	1	1	1	0	0	1	1	0	6
16:00–16:15	1	1	0	0	1	0	1	0	0	4
16:15–16:30	0	0	1	1	1	0	1	0	0	4
16:30–16:45	1	1	1	1	0	0	0	0	0	4
16:45–17:00	1	1	1	1	0	0	0	0	0	4
Total	22	21	20	19	18	14	8	4	4	

Table 4: Number of working time periods of each window employee in two weeks.

Personnel	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
Total	195	195	195	195	195	195	86	66	38

Table 5: Performance appraisal of flexible scheduling before and after the pilot (one week).

Pilot unit	Time	Number of the served taxpayers	Business volume	Online time	Acceptance of business	Operating duty cycle	Evaluation satisfaction
Yunhe county	Before the pilot	265	407	15300 min	1477.1 min	9.7	99.55
Yunhe county	After the pilot	375	644	7620 min	2318.9 min	30.4	99.76

(1) In this instance, the number of constraint conditions of the flexible scheduling is greater than 3000 and that of the flexible scheduling variables is greater than 2000. It is considerably better than the manual scheduling, which cannot easily satisfy the basic constraints; let alone obtain the optimal solution.

(2) The taxpayers' waiting time can be effectively controlled by the scientific calculation and scheduling.

(3) The number of window employees can be adjusted according to the dynamic changes in the demand.

(4) The quality of service of the windows can be improved by the window working mode; i.e., no more than 1 hr is worked continuously and it is alternated with a 15 min rest.

(5) The full-time window employees $A_1 - -A_6$, in one cycle, have the same working hours at the window. In the tax service hall, the head of the window employees is responsible for consultation and his/her work mode is A_9. A_7 is assigned to a part-time window employee with a good performance in the integrated management platform and A_8 is assigned to a part-time window employee with a poor performance.

A case study of Yunhe County Local Taxation Bureau in Lishui City, Zhejiang Province, showed that, as can be explained by the advantages of the flexible scheduling, the taxpayers, front-line window employees, and window management personnel benefited from a win-win situation in three areas. The number of served taxpayers and business transactions rose appropriately. The phenomenon of idle waiting of window employees was greatly reduced. The work efficiency and the average service times, as well as the taxpayers' satisfaction, were improved. After the pilot of the flexible scheduling system was conducted, the average waiting time of the taxpayers did not exceed 15 min and the taxpayers were more satisfied with the quality of service. The length of working time of the window employees was greatly reduced from 185000 hours in 2014 to 80000 hours in the same period in 2015, and the ratio of actual acceptance of business hours to the total window open hours increased from 22.67% to 44.45% (see Table 5 in Appendix B).

7. Conclusion

Based on the analysis of the problem of queues in the tax service hall in Lishui City, in this study a flexible window employee scheduling model was established by means of applying queuing theory and nonlinear integer programming. In strict compliance with national labour laws and regulations, and on the premise that it provides better services at the windows, the model can not only optimize the window employees scheduling but also consider the window employees work and rest habits and fulfill leave requirements, with the target of minimizing the employees total working time and reducing the taxpayers waiting time. After one-year operation at the Lishui City local taxation bureau, the flexible scheduling model could achieve a tolerable taxpayer average waiting time in the peak season, thus effectively alleviating the long queuing at the windows. At the same time, the model can also provide decision-making support to tax hall administrators on setting the number of open windows scientifically and reasonably. The algorithm and the model were designed to offer user-friendly scheduling software. In order to automatically obtain the optimized scheduling scheme, administrators need only to input the basic parameters, which facilitates the scheduling of work. The mathematical programming model in this paper was established to find an optimal solution to flexible scheduling. The model facilitates the reasonable arrangement of the window employees work, optimizing the working time of the window employees, saving manpower and scheduling costs, and improving the quality of service of the windows and thus helps improve the taxpayer's satisfaction and enhance the image of the service provided at the windows.

Appendix

A.

See Figures 1, 2, 3, and 4.

B.

See Tables 1, 2, 3, 4, and 5.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Yujun Sun is responsible for the implementation of the algorithm in the paper, and he has made detailed modifications and polishing of the paper.

References

[1] H. Deutsch and V. A. Mabert, "Queuing theory and teller employeeing: A successful application," *Interfaces*, vol. 10, no. 5, pp. 63–67, 1980.

[2] D. Hammond and S. Mahesh, "A simulation and analysis of bank teller manning," in *Proceedings of the 27th Conference on Winter Simulation, WSC'95*, pp. 1077–1080, 1995.

[3] P. Jones and E. Peppiatt, "Managing perceptions of waiting times in service queues," *International Journal of Service Industry Management*, vol. 7, no. 5, pp. 47–61, 1996.

[4] K. C. So and C. S. Tang, "On managing operating capacity to reduce congestion in service systems," *European Journal of Operational Research*, vol. 92, no. 92, pp. 83–98, 2007.

[5] Nosek R. A. and J. P. Wilson, "Queuing theory and customer satisfaction: A review of terminology trends and applications to pharmacy practice," *Hospital Pharmacy Journal*, vol. 36, no. 3, pp. 275–279, 2001.

[6] Y. B. Wang, Q. Cheng, and J. D. Cao, "Optimized M/M/c model and simulation for bank queuing system," *IEEE Software Engineering and Service Sciences*, vol. 16, no. 18, pp. 474–477, 2010.

[7] O. D. Ogunwale and O. A. Olubiyi, "A comparative analysis of waiting time of customers in banks," *Global Journal of Science Frontier Research*, vol. 10, no. 6, pp. 97–99, 2010.

[8] S. Moondra, "An LP model for workforce scheduling for banks," *Journal of Bank Research*, vol. 7, no. 4, pp. 299–301, 1976.

[9] L. J. Krajewski, L. P. Ritzman, and P. McKenzie, "Shift scheduling in banking operations: a case application," *Interfaces*, vol. 10, no. 2, pp. 1–8, 1980.

[10] A. T. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier, "Staff scheduling and rostering: A review of applications methods and models," *European Journal of Operational Research*, vol. 153, no. 1, pp. 3–27, 2004.

[11] V. A. Mabert, "A case study of encoder shift scheduling under uncertainty," *Management Science*, vol. 25, no. 7, pp. 623–631, 1979.

[12] R. N. Burns and G. J. Koop, "Modular approach to optimal multiple-shift manpower scheduling," *Operations Research*, vol. 35, no. 1, pp. 100–110, 1987.

[13] Y. J. Cheng and L. Luo, "A teller scheduling model based on queuing theory and integer programming," *Chinese Journal of Management*, vol. 7, no. 10, pp. 1558–1565, 2010.