Physician Scheduling for Emergency Departments Under Time-Varying Demand and Patient Return

Zixiang Wang, Ran Liu, and Zhankun Sun

***Abstract*— Emergency departments (EDs) are facing increasing overcrowding and long patient waiting time, which is mainly caused by the time-varying demand of new and returning patients. In this paper, we focus on scheduling ED physicians to reduce the patient waiting time and the physician working hours. We consider the ED network as a time-varying queuing system with returns and provide an analytical methodology to approximate the system state and patient waiting time of this system. The computation of the system state is based on the pointwise stationary fluid flow approximation method, while we compute the patient waiting time by classifying the patients into groups and individually calculating the waiting time of each group. Because of the nonlinearity of the approximation methods, we propose a linearization technique to formulate the physician scheduling problem as a mixed-integer programming (MIP) model. Since the MIP model is hard to be solved by an optimization solver, a tabu search algorithm is designed. Numerical experiments show that our proposed methods can reasonably approximate the system state and patient waiting time of this complex queueing model. The scheduling computed by the heuristic algorithm can improve the physician schedule without increasing the number of physicians.**

***Note to Practitioners*—This article is motivated by the emergency department of our collaborative hospital in Wuhan, China. The emergency department wishes to use a "flexible shifts" strategy to obtain a better physician scheduling plan. Different from the traditional "three shifts" strategy, the "flexible shifts" strategy has more available shifts and more flexible physician assignments to accommodate the fluctuation of the patient demands. However, the managers generally have difficulty providing high-quality schedules to physicians, since they usually lack the understanding of the impact of the time-varying patient demands with returns. Thus, we propose a set of approaches to solve this problem. Especially, a computational approach for calculating the patient waiting time that considers the stochastic and time-varying arrivals of patients and their returns is proposed. Experiments with hospital's real-life data show the methods proposed in this paper are useful for generating reasonable scheduling plans that can reduce the patient waiting time and system state without increasing the physician numbers.**

***Index Terms*— Emergency departments, time-varying demands, service with returns, physician scheduling, queuing theory.**

I. Introduction

EMERGENCY departments (EDs) in China have been suffered from a shortage of physicians. Due to the physician shortage, overcrowding and long waiting time have become a common problem in hospital emergency departments (EDs) in China. This problem is not unique to China. According to Kennedy *et al.* [1], over half of the patients in the USA report experiencing long waiting times. Similarly, research from France also reports problems like overcrowding in the ED and excessive patient waiting time [2].

There are several underlying reasons for the overcrowding and long waiting time. One reason is that the patient's arrivals are strongly stochastic and time-varying. Unlike outpatient departments, where the appointment mechanism smoothes the patient arrival rate's fluctuation, emergency patients cannot make an appointment and arrive randomly. Moreover, the arrival rate of emergency patients fluctuates dramatically during a day. We use our collaborative hospital in Wuhan, China, as an example. Fig. 1 shows a typical pattern of the arrival rate of emergency patients in one day in 2017. The arrival rate of emergency patients in this hospital is low in the nighttime and fluctuates intensively during the daytime. The arrival rate increases dramatically at 6:00 and reaches the first peak at about 7:00, remains low in the next few hours, and reaches the second peak at about 14:00. The physician scheduling plan in EDs often lacks flexibility and fails to offer adequate physicians in these peak hours, which leads to overcrowding and excessive patient waiting time.

Another critical reason leading to the overcrowding and excessive waiting time is the return of patients. Generally, physician's medical decisions are based on the results of various examinations, such as blood tests, urine tests, etc. Thus, after the first physician consultation, patients usually need to go through a series of medical tests and then return to the physicians to get diagnosed and treated again. In some special cases, patients may need to examine and revisit the physician multiple times. Time-varying and stochastic arrivals of new and returning patients superpose each other, exacerbating the overcrowding and increasing patient waiting time. During

Manuscript received 30 December 2021; revised 2 March 2022; accepted 25 March 2022. Date of publication 14 April 2022; date of current version 6 January 2023. This article was recommended for publication by Associate Editor S. Gao and Editor J. Li upon evaluation of the reviewers' comments. This work was supported by the National Natural Science Foundation of China under Grant 71972133 and Grant 71672112. *(Corresponding author: Ran Liu.)*

Zixiang Wang and Ran Liu are with the Department of Industrial Engineering and Management, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zixiang_wang@sjtu.edu.cn; liuran2009@sjtu.edu.cn).

Zhankun Sun is with the Department of Management Sciences, College of Business, City University of Hong Kong, Hong Kong (e-mail: zhankun.sun@cityu.edu.hk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2022.3163259.

Digital Object Identifier 10.1109/TASE.2022.3163259

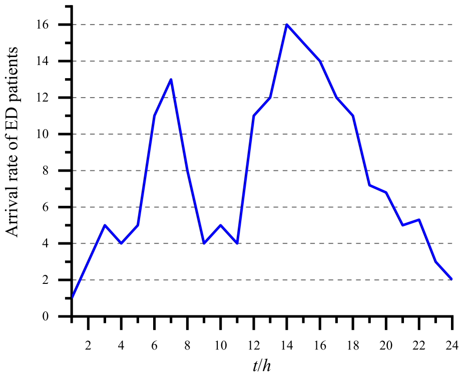

Fig. 1. Hourly ED patient arrival rate in an ED.

TABLE I
THE PHYSICIAN WORKING SHIFTS IN ONE ED

No. of shift	Schedule time
1	8:00-16:00
2	10:00-18:00
3	12:00-20:00
4	14:00-22:00
5	16:00-24:00
6	24:00-8:00

the emergency patient’s medical process, various departments, staff and equipment together form a complex queuing network, which makes the physician scheduling more challenging.

Although the EDs are in a stochastic and complex environment where the number of new patients varies over time and patients often require repetitive services, the planning and scheduling of physicians are carried out intuitively in many hospitals. For example, a widely used scheduling pattern is the ***“three shifts” strategy***, in which only three working shifts are used, namely morning shift, afternoon shift, and night shift, and the only decision is to assign the physicians to these shifts. This strategy is easy to be implemented, but it is hard to cope with the fluctuations of patient demands. To improve emergency services, hospital managers have been devoting increasing attention to the “***flexible shifts***” strategy. Compared with the traditional scheduling method, the “flexible shifts” strategy introduces more available shifts. With more possible working shifts, the physicians’ assignments can become more flexible and the resulting scheduling can better match the time-varying demands of emergency patients, which can reduce the waiting time of patients at different hours and the potential waste of the service capacity in the periods with few patients. Table I shows the available candidate shifts in our cooperative ED.

The “flexible shifts” strategy offers an opportunity for a more flexible physician schedule, yet it presents many difficulties. First, physician scheduling becomes more complicated since there are more shifts available. Second, the time-varying and stochastic nature of new arrivals and reentrants must be carefully considered in the physician scheduling. Third, the patient waiting time is a critical performance metric for emergency service systems. The computation of the patient waiting time with a given scheduling plan is the basis for the physician scheduling optimization. How to effectively evaluate the patient waiting time of such a system remains to be investigated. In this paper, motivated by the physician scheduling problem of our partner hospital in Wuhan, China, we address the weekly ED physician scheduling problem with the “flexible shifts” strategy. We summarize the main contributions of this paper as follows.

First, we develop analytical approaches to approximate the system state and patient waiting time of a time-varying queuing system that serves both newly-arrived and returning patients. The accuracy of the proposed computation method is observed numerically. Second, we design a tabu search (TS) algorithm to effectively solve the physician scheduling problem under the time-varying demands of patients with returns. The effectiveness of the algorithm in real-life scenarios is verified through a set of numerical experiments.

The remaining paper is organized as follows. We give a brief literature overview in section II. In section III, we introduce our queuing model and our system state and waiting time computation methods. In section IV, we develop a MIP model for the physician scheduling problem based on our proposed approximation methods and a pointwise linearization method. Section V describes the design of our TS algorithm. Section VI validates our proposed approximation approaches and conducts numerical experiments. Section VII summarizes in conclusion and identifies further research directions.

II. RELATED LITERATURE

Numerous studies have investigated the physician staffing/scheduling problem. In the existing literature, most researchers use standard mathematical programming techniques, i.e., linear programming, integer programming, and mixed-integer programming, to formulate models for the physician staffing/scheduling problem, while little research uses nonlinear programming, constraint programming, etc. to address the problem [3]. The solution technique also varies among researchers. Exact solution algorithms are a typical solution approach for solving mathematical programming models. Researches using this approach can be generally divided into two sub-groups. Some studies use mathematical optimization solvers such as CPLEX [4] or Gurobi [5], while others design exact solution algorithms, e.g., Branch-and-Cut [6], Branch-and-Price [7], etc., to solve the models. The exact solution algorithms can get the optimal solution of the model, but the approach’s weakness is that it cannot even get a solution of the model with high complexity or large size within a reasonable time, which is the dominant area of heuristics. Heuristics can get a good solution (usually not the optimal solution) within a reasonable computational time. Heuristics such as genetic algorithm [8], simulated annealing [9], variable neighborhood search [10], column generation-based heuristic [11], etc., have been used to solve the physician staffing/scheduling problem. For more reference about physician scheduling, we refer the reader to Erhard *et al.* [3].

Since EDs have to confront stochastic and time-varying demands of new and returning patients, the performance evaluation of non-stationary service systems with returns is the second related topic of our paper. While many researchers have analyzed the performance evaluation methods for non-stationary queuing systems in hospitals [3], only a few researchers have investigated the performance measures of time-varying service systems with returns in healthcare units. Among the different performance evaluation methodologies, simulation is a frequently used method to get the performance metrics of complex ED systems. For example, Ghanes *et al.* [12] use simulation to calculate the length of stay (LOS) and service level of emergency patients. The same performance metrics are also simulated by Guo *et al.* [13]. Vanbrabant *et al.* [14] use simulation to compute the key performance indicators, such as the patient waiting time before getting the first physician's consultation and the patient LOS, of an ED. Kuo [15] uses an ED simulation model to assess the average patient waiting time. Ahmed and Alkhamis [16] use simulation to compute the ED patient's throughputs. Simulation can approximate the exact value of performance metrics of complex ED systems, but the validation and computation of a simulation model are usually time-consuming. For more reference about the simulation, we refer readers to Vanbrabant [17], Kuo *et al.* [18], Ghanes *et al.* [19] and Zeltyn *et al.* [20].

Because of the interaction between different medical processes and the superposition of newly arrived and returning patients, analytical methods to estimate the performances of non-stationary service systems with returns in hospitals are thus far a less investigated topic. In the existing literature, the sample average approximation (SAA) method has been commonly used to evaluate the performance of such a system. For example, EL-Rifai *et al.* [2] use SAA to model the patient waiting time of an ED system. The basic idea of the method is to estimate the mathematical expectations of waiting time by their sample averages. Based on this estimation, a scheduling model with time-varying and stochastic demand of new and returning patients can be approximated by a deterministic optimization problem. Xiao *et al.* [21] also use SAA to compute the patient waiting time in a clinical department with revisits. Zaerpour *et al.* [22] use SAA to calculate the mismatch between the demand of emergency patients and the service productivity of physicians. When the number of samples is enough, SAA can well approximate the performance metrics of a complex queuing system, but increasing the number of samples will increase the size of the model linearly, which makes the model hard to solve. The second alternative is the fluid approximation method, which uses deterministic fluid models to approximate the stochastic models [23]. Whitt [23] reveals that the fluid approximation is particularly useful for the evaluation of the system that is temporarily overloaded. Yom-Tov and Mandelbaum [24] analyze the time-varying modified offered load (MOL) of a system, called the "Erlang-R" system, using a fluid model and designed a physician staffing algorithm based on the "Square Root Staffing Rule" with the offered load as inputs. The "Erlang-R" system is characterized by station 1 with n servers and station 2 with infinite servers. Customers who leave station 1 will either leave the system or go to station 2 with a given possibility, while customers who leave station 2 will return to station 1. Their experiments show that their method is effective to model the offered load of such a system. Chan *et al.* [25] also use the fluid model to investigate the effects of state-dependent service rate and return probability of an "Erlang-R" system. Ingolfsson *et al.* [26] use the fluid approximation method to compare the equilibrium state and the transient behavior of two different queuing systems with state-dependent service time and return probability. The difference between the two models is the time to decide whether a customer returns for services. Queuing theory is also a common tool in the system evaluation of service systems with returns. For instance, Huang *et al.* [27] investigate the policy of patient flow control in an ED from a queueing theory perspective and demonstrate that it is optimal to prioritize returning patients over newly-arrived patients while adhering to their deadlines in emergency department under heavy traffic. Besides, data-driven approaches are also frequently used in the estimation of system performances. For example, Whitt and Zhang [28] study the distribution of patient LOS in an ED based on data-driven models. Stefanini *et al.* [29] also use data-driven models to evaluate the performance indicators such as patient LOS, patient waiting time for treatment, admission rate, etc.

In the existing literature, to our knowledge, the paper by Yom-Tov and Mandelbaum [24] is the only work that considers both the analytical evaluation of the performance of non-stationary service systems with returns and the physician staffing problem. This paper will present our performance evaluation methods of our queuing model and a mathematical formulation to address the physician scheduling problem. Instead of evaluating the offered load as Yom-Tov and Mandelbaum [24], we analyze the system state and patient waiting time, which can intuitively reflect the service system's performances. We propose a new method based on the pointwise stationary fluid flow approximation (PSFFA) to compute the system state of a time-varying queue with returns. Based on the system state computation, we propose a novel method to calculate the patient waiting time, which is used as the basis of the physician scheduling model. Since the model is hard to solve, we propose additionally an effective heuristic algorithm to solve it.

III. Queuing Model in the ED

Since various hospitals may have different ED medical processes, we first summarize the service process of our collaboration ED. Upon the patients' arrival, they are registered and triaged by nurses. Urgent patients (with severe conditions) are immediately sent to the resuscitation room via a separate "acute care" track and resuscitated by paramedics, while non-urgent patients will go to the waiting area of the ED and wait until getting evaluated in the first consultation of physicians. After the first consultation, only a few patients complete the service process: either leave the ED or are admitted to the hospital. In contrast, most patients need to go to medical examination departments for medical tests such as X-rays, blood tests, B-ultrasonic, etc. After getting medical checks, the patients go back to the waiting area and wait to

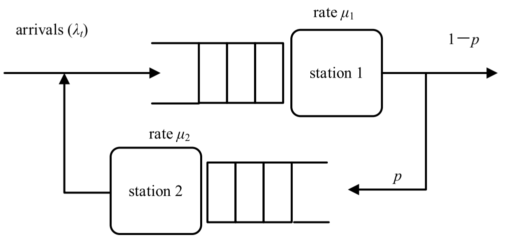

Fig. 2. Queuing network model in the ED.

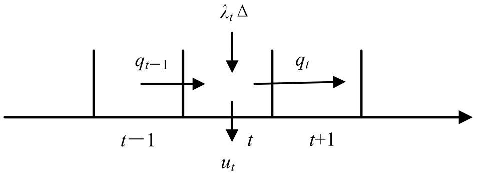

Fig. 3. Fluid equilibrium in an $M_t/M/c_t$ queuing system.

revisit the physician. Note that a patient may be requested medical tests more than once. Finally, a decision is made to release the patient from the ED (home or hospitalized).

According to our observation of the ED, the portion of urgent patients with severe conditions is limited, and the overcrowding is not primarily caused by these urgent patients. Therefore, we focus on the services of the non-urgent patients, whose services can be modeled as a queuing network with two stations, as shown in Fig. 2. We denote the physicians and examination servers "station 1" and "station 2", respectively. The planning horizon is divided into several periods, each of which has a time length of Δ. In each period, the patient arrival rate, the number of physicians and examination servers, and the service rate of servers are assumed unchanged. The return probability, i.e., the probability that a patient needs medical examinations and returns to physicians for further consultation, is p. Patients who leave without being served are not considered.

To simplify the problem, we additionally make the following assumptions in our paper:

A1. In each period t, patients arrive at the physician's according to a Poisson distribution with rate λ_t and are served under a first-come-first-serve (FCFS) rule. This assumption is supported by the data from our partner hospitals when intervals are selected appropriately.

A2. All physicians are homogeneous, i.e., with the same skills and service rate. The service time of physicians is exponentially distributed with service rate μ_1. The assumption of exponential service times is reasonable and common in healthcare settings [2], [24], [30].

A3. All the medical examinations are homogeneous, and the examination time is exponentially distributed with service rate μ_2. In practice, as stated above, there are different types of examinations with varying service times, while patients need various tests. We assume that the service rate μ_2 is the examination's average service rate, i.e., the examination has a stochastic duration with a mean of $1/\mu_2$.

With the above assumptions, our queuing model can be considered as an $M_t/M/c_t$ queuing system with returns. The returning patients make our queuing model different from the classical model and more complicated.

A. System State Computations

We first discuss the computation methods of the system state with a given physician staffing, i.e., assess the expected total number of patients in the system under the condition that the number of servers is known. Our computation methods are built based on PSFFA [31]. We briefly explain the principles of the PSFFA for approximating the system state for an $M_t/M/c_t$ queuing system, then we introduce the system state computation method for our queuing model.

1) $M_t/M/c_t$ Queuing System: According to the fluid model, as shown in Fig. 3, the system state at the beginning of the period t plus the number of patients arrived in the period t equals the system state at the end of the period t plus the number of patients served in the period t. Let u_t represent the expected number of patients served in the period t, q_{t-1} and q_t be the system state at the beginning and the end of the period t, respectively. We have the following equation:

$$q_t + u_t = q_{t-1} + \lambda_t \Delta. \tag{1}$$

Let ρ_t be the average service intensity, then (1) can be expressed as

$$q_t + c_t \mu \rho_t \Delta = q_{t-1} + \lambda_t \Delta. \tag{2}$$

If an $M/M/c$ queue system has constant parameters (λ, μ, c) with $\rho = \lambda/(c\mu) < 1$ and $c > 1$, the queuing system can reach the steady-state, and the steady-state system state can be calculated by

$$l^{M/M/c}(\rho, c) = \frac{\rho^{c+1} c^c}{c!(1-\rho)^2} \pi_0 + \rho c, \tag{3}$$

where

$$\pi_0 = \left[\sum_{i=0}^{c-1} \frac{(\rho c)^i}{i!} + \frac{(\rho c)^c}{c!(1-\rho)} \right]^{-1}. \tag{4}$$

Note that (3) and (4) are used in the steady queuing system [32]. We assume that the queue system can reach steady-state at the end of each period, so we can use the stationary formula with parameters ρ_t and c_t to compute the system state at the end of the period t, i.e., in our approximation approach,

$$q_t \approx l^{M/M/c}(\rho_t, c_t). \tag{5}$$

In PSFFA, ρ_t is not the service intensity $\lambda/(c\mu)$ in the queuing theory, but the average service intensity $(u_t/\Delta)/(c\mu)$ of the service system. Thus, even if the system is overloaded, i.e., $\lambda/(c\mu) > 1$, the average service intensity ρ_t does not exceed 1. Since ρ_t is in the range of [0, 1) and q_t is monotonically increasing and convex function of ρ_t [23], ρ_t can be solved using the bisection method [31]. By substituting the value of ρ_t into (5), we can calculate the system state at the end of the period t. We call this computation the **"APP-length-1"** method.

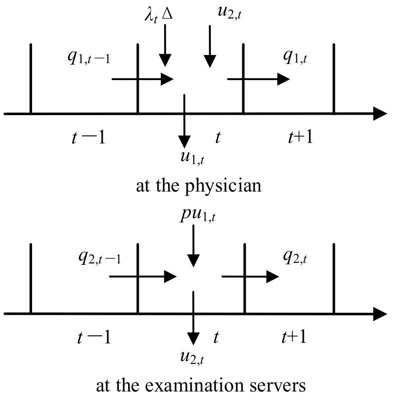

Fig. 4. Fluid equilibrium in our queuing model.

2) Our Queuing Model: We extend the PSFFA to our queuing network. We add the number "1" and "2" in the subscript to represent two stations, while the meaning of symbols remains the same. For example, $q_{1,t}$ and $q_{2,t}$ denote the system state at the end of period t in station 1 and station 2, respectively. Assume that the number of examination servers c_2 is a constant. Note that the patients in the physician's queue consist of the newly arrived and returning patients. In some EDs, returning patients may have higher priority than the new patients, but such priority is not adopted in some other hospitals. In our computation, returning patients' priority is not considered, i.e., all patients are served by an FCFS rule. The fluid equilibrium of our queuing model of period t is shown in Fig 4.

Similar to an $M_t/M/c_t$ queuing system, the fluid balance of our queuing model can be expressed as following (6) and (7):

$$q_{1,t} + c_{1,t}\mu_1\rho_{1,t}\Delta = q_{1,t-1} + \lambda_t\Delta + c_2\mu_2\rho_{2,t}\Delta \quad (6)$$

$$q_{2,t} + c_2\mu_2\rho_{2,t}\Delta = q_{2,t-1} + pc_{1,t}\mu_1\rho_{1,t}\Delta \quad (7)$$

When $c_{1,t}$ and c_2 are given, only two variables $\rho_{1,t}$ and $\rho_{2,t}$ are unknown in (6) and (7), since $q_{1,t}$ and $q_{2,t}$ can be approximated by $l^{M/M/c}(\rho_{1,t}, c_{1,t})$ and $l^{M/M/c}(\rho_{2,t}, c_2)$, respectively. We give an initial value to $\rho_{1,t}$. Substituting $\rho_{1,t}$ into (7), similar to the solution procedure of the "APP-length-1" method, $\rho_{2,t}$ can be computed by the bisection method. In the same way, substituting the computed $\rho_{2,t}$ into (6), $\rho_{1,t}$ can be calculated. This process is repeated until (6) and (7) hold. Then the system states $q_{1,t}$ and $q_{2,t}$ can be obtained. In part A of the appendix, we use a simple numerical example to show the solution process of the equations. Furthermore, in part B of the appendix, we demonstrate that the unique solution of (6) and (7) always exists and can be solved using the proposed bisection method.

The key of the system state computation for an $M_t/M/c_t$ queuing system and our queuing network is the estimation of the average service intensity. We consider three scenarios:

Case 1 (Severely Overloaded System): For a queueing system, if the system is severely overloaded, i.e., the system's service capacity is much smaller than the demand of patients, the system can be considered working continuously ($u_t = c_t\mu\Delta$), and the average service intensity can therefore be approximated by one. In our system, the examination such as blood test, urine test, etc. can be performed simultaneously, so the examination's service capacity is usually large enough to cover the patients' demands. Since the number of physicians is limited and physicians' service capacity may be temporarily inadequate, we define the "severely overloaded system" as that physicians' service capacity is much smaller than the demand of patients. In period $t,$ we introduce a parameter $\bar{\rho}$ when the system is critically overloaded: if $\hat{\rho} = (q_{1,t-1} + \lambda_t)/c_{1,t}\mu_1$is larger than the predefined $\bar{\rho}$, the average service intensity $\rho_{1,t}$ can be approximate to one and $\rho_{2,t}$ is calculated by solving (7). The system state $q_{1,t}$ can be then calculated by

$$q_{1,t} = \max(q_{1,t-1} + \lambda_t\Delta + c_{2,t}\mu_2\rho_{2,t}\Delta - c_{1,t}\mu_1\Delta, 0). \quad (8)$$

Case 2 (Relatively Low Service Intensity): In such a case, the stationary queueing formulas are more appropriate to describe the system, i.e., $\hat{\rho} = (q_{1,t-1} + \lambda_t)/c_{1,t}\mu_1$is smaller than a predefined parameter ρ. In such a case, the intensities of $\rho_{1,t}$ and $\rho_{2,t}$ can be computed from (6) and (7). Then, the system states at each station can also be obtained from the results of such two formulas.

Case 3 (The Service Intensity Is Between the Above Two Cases): I.e., the system is neither overloaded nor with low service intensity. We combine the numerical results of them to calculate the system states.

In summary, the system state in the physician's queue at the end of period t is calculated as follows:

$$q_{1,t} = \begin{cases} \text{the result from (6)} \sim \text{ (7)} & \hat{\rho} < \rho \\ \text{the average of both values} & \rho \le \hat{\rho} \le \bar{\rho} \\ \text{the result from (8)} & \hat{\rho} > \bar{\rho}. \end{cases} \quad (9)$$

where $\hat{\rho} = (q_{1,t-1} + \lambda_t)/c_{1,t}\mu_1$, $\bar{\rho}$ and ρ are two parameters. This computation is referred to as the "**APP-length-2**" method.

B. Waiting Time Computations

Besides the system state, the waiting time is also a critical metric of the system performance. Given the system state at the beginning and the end of a period obtained by the above APP-length-2 method, we try to propose a method to compute the patient waiting time in the period. Similar to Liu and Xie [10], our method divides the patients into three groups: (i) patients who arrived before period t and get served in period t, (ii) patients who arrived and get served in period t, and (iii) patients who get served after period t, who are denoted as the first-, second- and third-group patients, respectively. The waiting times of three groups of patients in period t are referred to as wt_1, wt_2, and wt_3, and the number of three groups of patients in period t is denoted as u_1, u_2, and u_3, respectively. The waiting time of each group of patients is computed individually. The total waiting time W_t in period t can then be calculated by $W_t = wt_1 + wt_2 + wt_3$. Note that W_t is not the total waiting time of patients who arrived in period t but the total waiting time of patients in period t. Because the capacity of examination (such as blood test, urine test, etc.) is usually adequate and the patient waiting time at the examination servers is generally short, we only focus on the patient waiting time in the physicians' queue in each period in this section. We call the waiting time computation the **“APP-time-1”** method.

1) Computation of wt_1: We first propose the waiting time computation method for an $M_t/M/c_t$ queuing system that does not consider the returning patients as the basis. Next, we extend it to our time-varying system with returning patients. For an $M_t/M/c_t$ queuing system, the number of the first-group patients u_1 in period t is dependent on the number of served patients u_t and the initial state of the system q_{t-1} in period t. If $u_t > q_{t-1}$, all q_{t-1} patients get served in the period t, i.e., u_1 equals q_{t-1}. Otherwise, u_1 equals u_t. In summary,$u_1 = \min(q_{t-1}, u_t)$. Assumed that all c servers are not occupied at the beginning of period t, so the first c_t patients do not wait and $(u_1 - c_t)^+$ patients wait in period t, while the $(c_t + n)$-th patient has to wait until n patients leave the system. So the waiting time of the $(c_t + n)$-th patient is equivalent to the occurrence time of the departure of the n-th (first-group) patient. Since the departure of the patient is according to a Poisson process with rate $(c_t\mu)$, the average waiting time for $(c_t + n)$-th patient is $n/(c_t\mu)$. Therefore, the waiting time wt_1 can be computed by

$$wt_1 = \sum_{n=1}^{(u_1-c_t)^+} \frac{n}{c_t\mu} = \frac{((u_1 - c_t)^+ + 1)(u_1 - c_t)^+}{2c_t\mu}.$$

In our model, the computation of wt_1 is more complicated than the $M_t/M/c_t$ model because of the returning patients. As stated in Yom-Tov and Mandelbaum [15], for a stationary $M/M/c$ with returning patients model, two strategies can be used to modify the model to the basic *M/M/c model*: increase the arrival rate to $\lambda/(1-p)$, or reduce the service rate to $(1-p)\mu$. Using the first option as an example, the amplification of the arrival rate is due to the effect of returns: the amplified number of arrivals equals the sum of the number of new arrivals and returning patients, which can be computed by

$$\lambda + \lambda p + \lambda p^2 + \cdots = \lambda \sum_{k=0}^{\infty} p^k = \lambda/(1-p).$$

We compute the amplification for our queuing model in a similar way. Considering that our system cannot reach a steady state and the number of returns in one period is finite, we denote υ as the average return number of one patient in a period, so the modified arrival rate of our model should be

$$\lambda + \lambda p + \lambda p^2 + \cdots \lambda p^{\lfloor \upsilon \rfloor} = \lambda \sum_{k=0}^{\lfloor \upsilon \rfloor} p^k.$$

Since the average service time in station 1 and station 2 are μ_1^{-1} and μ_2^{-1}, so υ equals $(\mu_1^{-1} + \mu_2^{-1})^{-1}$. Correspondingly, the second alternative for our system is to minify the service rate, i.e., a minified service rate μ', computed by $\mu \Big/ \sum_{k=0}^{\lfloor \upsilon \rfloor} p^k$. Because the second alternative turns out to be a superior fit for our queuing model in numerical experiments, we propose methods based on the second strategy to compute the waiting time for our system. Thus, we calculate the patient waiting time wt_1 of period t by

$$wt_1 = \sum_{n=1}^{(u_1-c_{1,t})^+} \frac{n}{c_{1,t}\mu'} = \frac{((u_1 - c_{1,t})^+ + 1)(u_1 - c_{1,t})^+}{2c_{1,t}\mu'},$$

where $\mu' = \mu \Big/ \sum_{k=0}^{\lfloor \upsilon \rfloor} p^k$.

2) Computation of wt_2: The number of the second-group patients u_2 equals $u_t - u_1$. Assumed that a (second-group) patient j arrives in the service system and finds that the system state is q_j (excluding this patient j) and patients leave the system at a rate of $c_{1,t}\mu'$. Then the waiting time of patient j can be calculated by

$$t_j = \begin{cases} 0 & q_j < c_{1,t} - 1 \\ \dfrac{q_j - c_{1,t} + 1}{c_{1,t}\mu'} & otherwise. \end{cases} \tag{10}$$

Thus, wt_2 can be computed by adding up the waiting time of all second-group patients, and the critical issue is to calculate q_j in (10). The q_j computation depends on the change tendency of the number of patients, for example, if the arrival time of the j-th patient τ_j is known and the patients depart at a constant rate of $c_{1,t}\mu'$ before τ_j, then q_j should equal $q_{1,t-1} + (\lambda_t - c_{1,t}\mu')\tau_j$. We discuss the q_j computation in the following three cases. In all three cases, we assume that the arrival time of the first (second-group) patient is zero and the inter-arrival time of patients is $1/\lambda_t$, i.e., $\tau_j = (j-1)/\lambda_t (j = 1, \ldots, \lfloor u_2 \rfloor)$.

Case 1: $\lambda_t \geq c_{1,t}\mu'$. This means the system state is increasing in a period. In this case, we assume that the system state grows at a rate of $(\lambda_t - c_{1,t}\mu')$, so the system state when patient j arrives is

$$q_j = q_{1,t-1} + \left(\lambda_t - c_{1,t}\mu'\right)\tau_j. \tag{11}$$

Case 2: $\lambda_t < c_{1,t}\mu'$ and $q_{1,t-1} > q_{\text{steady}}$, where q_{steady} denotes the steady-state system state of an $M/M/c$ queuing system with constant parameters $(\lambda_t, \mu', c_{1,t})$. This condition means the system state tends to first decrease from $q_{1,t-1}$ to q_{steady} and then keeps stable. We further divide this case into three different subcases. The classification is according to the relationship among $q_{1,t-1}$, q_{steady}, and $c_{1,t}$, which influences the change rate of the system state.

Case 2.1: $q_{1,t-1} > q_{\text{steady}} > c_{1,t}$. This case implies that the initial system state first falls to q_{steady} at a rate of $(c_{1,t}\mu' - \lambda_t)$ and keeps stable at q_{steady} for the rest of time of this period (as shown in Fig. 5). Let t_{steady} be the time needed to reduce the system state from $q_{1,t-1}$ to q_{steady}.

So, t_{steady} can be computed by

$$t_{\text{steady}} = \frac{q_{1,t-1} - q_{\text{steady}}}{c_{1,t}\mu' - \lambda_t}.$$

So the system state q_j is

$$q_j = \begin{cases} q_{1,t-1} + (\lambda_t - c_{1,t}\mu')\tau_j & \tau_j < t_{\text{steady}} \\ q_{\text{steady}} & otherwise. \end{cases}$$

Case 2.2: $q_{1,t-1} > c_{1,t} > q_{\text{steady}}$. This case implies that the initial system state of patients falls to $c_{1,t}$ at a rate of $(c_{1,t}\mu' - \lambda_t)$, then is declined to q_{steady} at a rate of $(q\mu' - \lambda_t)$

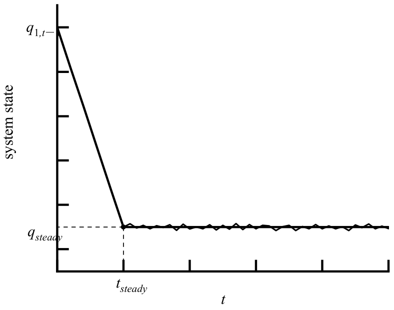

Fig. 5. Change tendency of the system state in case 2.1.

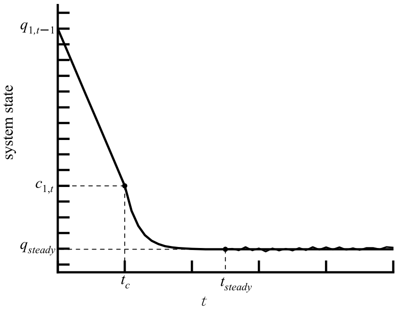

Fig. 6. Change tendency of the system state in case 2.2.

and keeps stable at q_{steady} for the rest of the time (shown in Fig. 6).

Let t_{c} be the time when system state $q_{1,t-1}$ reduces to $c_{1,t}$, which can be computed by

$$t_c = \frac{q_{1,t-1} - c_{1,t}}{c_{1,t}\mu' - \lambda_t}.$$

The dynamics of the reduction of system state q from $c_{1,t}$ to q_{steady} can be expressed as

$$dq\big/dt = \lambda_t - q\mu'.$$

Integrating both sides with respect t,

$$-\ln\left|\lambda_t - q\mu'\right|/\mu' = t + C_0,$$

where C_0 is a constant of integration. Considering the initial state: $q = c_{1,t}$ when $t = t_{\text{c}}$, then:

$$-\ln\left|\lambda_t - q\mu'\right|/\mu' = t - \ln\left|\lambda_t - c_{1,t}\mu'\right|/\mu' - t_c.$$

So, t_{steady} can be computed by

$$t_{\text{steady}} = -(\ln\left|\lambda_t - q_{\text{steady}}\mu'\right| - \ln\left|\lambda_t - c_{1,t}\mu'\right|)/\mu' + t_c.$$

When $t_{\text{c}} < \tau_j < t_{\text{steady}}$, the system state falls at the rate of $(q\mu' - \lambda_t)$, so q_j can be obtained from the following equality

$$-\ln\left|\lambda_t - q_j\mu'\right|/\mu' = \tau_j - \ln\left|\lambda_t - c_{1,t}\mu'\right|/\mu' - t_c. \quad (12)$$

To sum up, the system state q_j can be computed by

$$q_j = \begin{cases} q_{1,t-1} + \left(\lambda_t - c_{1,t}\mu'\right)\tau_j & \tau_j \le t_c \\ q_j\text{obtained from (12)} & t_c < \tau_j < t_{\text{steady}} \\ q_{\text{steady}} & otherwise \end{cases}$$

Case 2.3: $c_{1,t} > q_{1,t-1} > q_{\text{steady}}$. In this case, the system state of patients falls to q_{steady} at a rate of $(q\mu' - \lambda_t)$ and is stable at q_{steady} during the rest time of the period. The process of the reduction of system state is similar to case 2.2 and can be presented by the following differential equation:

$$\frac{dq}{dt} = q\mu' - \lambda_t.$$

Integrating both sides with respect t,

$$-\ln\left|\lambda_t - q\mu'\right|/\mu' = t + C_0,$$

where C_0 is a constant of integration. Considering the initial state: $q = q_{1,t-1}$ when $t = 0$,

$$-\ln\left|\lambda_t - q\mu'\right|/\mu' = t - \ln\left|\lambda_t - q_{1,t-1}\mu'\right|/\mu'.$$

Let t_{steady} be the time needed to reduce the system state from $q_{1,t-1}$ to q_{steady}, so t_{steady} can be computed by

$$t_{\text{steady}} = -(\ln\left|\lambda_t - q_{\text{steady}}\mu'\right| - \ln\left|\lambda_t - q_{1,t-1}\mu'\right|)/\mu'.$$

When $\tau_j < t_{\text{steady}}$, the system state falls at the rate of $(q\mu - \lambda)$, so q_j can be obtained from the following equation

$$-\ln\left|\lambda_t - q_j\mu'\right|/\mu' = \tau_j - \ln\left|\lambda_t - q_{1,t-1}\mu'\right|/\mu'. \quad (13)$$

So, the computation of system state q_j can be summarized as follows:

$$q_j = \begin{cases} q_j\text{obtained from (13)} & \tau_j < t_{\text{steady}} \\ q_{\text{steady}} & otherwise. \end{cases}$$

Case 3: $\lambda_t/(c_{1,t}\mu') < 1$, $q_{1,t-1} \le q_{\text{steady}}$. This condition indicates that the system state tends to first increase from $q_{1,t-1}$ to q_{steady} and then keeps stable. In this case, for the first-arriving second-group patient, $q_j = q_{1,t-1}$. We assume that the change rate of the system state keeps unchanged in each patient inter-arrival time. The change rate of the system state is $(\lambda_t - c_{1,t}\mu')$ if $q_{j-1} > c_{1,t}$, while the change rate is $(\lambda_t - q_{j-1}\mu')$ if $q_{j-1} \le c_{1,t}$. So for other second-group patients, $q_j = q_{j-1} + \left(\lambda_t - \hat{\mu}\right)/\lambda_t$. To sum up, the value of q_j is computed by

$$q_j = \begin{cases} q_{1,t-1} & j = 1 \\ q_{j-1} + \left(\lambda_t - \hat{\mu}\right)/\lambda_t & otherwise, \end{cases}$$

where

$$\hat{\mu} = \begin{cases} c_{1,t}u' & q_{j-1} > c_{1,t} \\ q_{j-1}u' & otherwise. \end{cases}$$

3) Computation of wt_3: The number of the third-group patients u_3 is equivalent to the system state $q_{1,t}$. To calculate wt_3, we first introduce the following theorem.

Theorem 1: $\{N(t),\ n \ge 0\}$ is a Poisson process with parameter λ, within time interval [0,t], $N(t)$ patients arrived, the total expected waiting time of all patients from each arrival to instant t is $\lambda t^2/2$.

Proof: Let $X(t)$ be the expected waiting time of all patients, so

$$X(t) = \sum_{k=1}^{N(t)} (t - \tau_k).$$

According to the law of total expectation,

$$E[X(t)]=E[E[X(t)|N(t)]=E\left[E\left[\sum_{k=1}^{n}(t-\tau_k)|N(t)=n\right]\right].$$

Assume that $(U_{(1)}, U_{(2)}, \ldots, U_{(n)})$ be the order statistics corresponding to independent random variables uniformly distributed on the interval $[0,t]$. According to Theorem 5.2 in [33], "given that $N(t) = n$, the n arrival time $\tau_1, \ldots, \tau_n$ have the same distribution as the order statistics corresponding to independent random variables uniformly distributed on the interval $[0, t]$". So,

$$\begin{aligned}E[\sum_{k=1}^{n}(t-\tau_k)|N(t)=n] &= E[\sum_{k=1}^{n}(t-U_{(k)})]=tn-E[\sum_{k=1}^{n}U_{(k)}]\\ &= tn–E[\sum_{k=1}^{n}U_k]=tn-\frac{tn}{2}=\frac{tn}{2}.\end{aligned}$$

So, $E[X(t)|N(t)] = tN(t)/2, E[X(t)] = tE[N(t)]/2 = \lambda t^2/2$. Q.E.D.

We consider two scenarios for the computation of wt_3.

If $q_{1t} < \lambda_t\Delta$, which indicates that all the third-group patients arrived in the queuing system in period t and the time interval of all these patients' arrival is $[(1-q_{1,t}/(\lambda_t\Delta))\Delta, \Delta]$. So the waiting time wt_3 can be computed according to Theorem 1 by

$$wt_3=\frac{\lambda_t}{2}\left[\Delta-(1-\frac{q_{1,t}}{\lambda_t\Delta})\Delta\right]^2=\frac{\lambda_t}{2}(\frac{q_{1,t}}{\lambda_t})^2=\frac{q_{1,t}^2}{2\lambda_t}.$$

If $q_{1,t} \geq \lambda_t\Delta$, in this case, $\lambda_t\Delta$ third-group patients arrived in the queuing system in period t and $(q_{1,t}-\lambda_t\Delta)$ third-group patients arrived in the queuing system before period t. These $(q_{1,t}-\lambda_t\Delta)$ third-group patients wait for the entire period, while the waiting time of the rest $\lambda_t\Delta$ third-group patients can be computed according to Theorem 1, so the waiting time wt_3 can be calculated by

$$wt_3=(q_{1,t}-\lambda_t\Delta)\Delta+\frac{\lambda_t\Delta^2}{2}=q_{1,t}\Delta-\frac{\lambda_t\Delta^2}{2}.$$

To sum up, the waiting time wt_3 can be computed by

$$wt_3=\begin{cases}\dfrac{q_{1,t}^2}{2\lambda_t} & q_{1,t}<\lambda_t\Delta\\ q_{1,t}\Delta-\dfrac{\lambda_t\Delta^2}{2} & otherwise.\end{cases}$$

IV. Physician Scheduling Model

Based on the waiting time computations method, we formulate the ED physician scheduling problem as a mathematical model. The objective of the scheduling model is to minimize the total waiting time of patients and the total working time of physicians. The ED needs to assign the physicians to different predefined shifts to address the time-varying patient demands.

According to our field survey in our collaboration hospital, the following physician scheduling constraints are considered in our model:

1) Each physician can work only one shift in a day;
2) Each physician's working time cannot be interrupted;
3) The total working time of each physician cannot exceed H hours in a week;
4) Each physician performs no more than $C_{\max}$ and no less than $C_{\min}$ night shift in a week;
5) A physician rests at least 24 hours after completing a night shift;
6) At least one physician is working in each period of a week.

The planning horizon of the scheduling model is cyclic with $|T|$ periods, and each period has a length of Δ. In this paper, Δ is one hour. The total available number of ED physicians is $|K|$. Each day has $|N|$ available shifts, which are predefined by the ED, i.e., the ED predefines the start time and end time of each shift, such as the physician working shifts shown in Table I. The daily shift pattern remains the same, and it is assumed that the night shift is the last shift of a day. We use the binary parameter $r_{n,t}$ to express the working time of each shift. $r_{n,t}$ equals 1 only if period t is in the working hours of shift n, otherwise $r_{n,t}$ equals 0.

Sets and Indices

$i \in \mathcal{I}$: index of physicians, $\mathcal{I} = \{1, 2, \ldots, |K|\}$
$m \in \mathcal{M}$: index of days, $\mathcal{M} = \{1, 2, \ldots, 7\}$
$n \in \mathcal{N}$: index of shifts, $\mathcal{N} = \{1, 2, \ldots, 7|N|\}$
$t \in \mathcal{T}$: index of periods, $\mathcal{T} = \{1, 2, \ldots, |T|\}$

Parameters

$C_{\max}$: maximum number of night shifts assigned to a physician during one week
$C_{\min}$: minimum number of night shifts assigned to a physician during one week
H: maximum number of working periods of a physician during one week
$r_{n,t}$: 1, if period t is in the working hours of shift n; 0, otherwise
α: the parameter to balance two terms of the objective
Δ : length of a period
$|K|$: number of available physicians
$|N|$: total number of one day's shifts
$|T|$: total number of periods

Decision variables

$x_{i,n}$: 1, if physician i is assigned to shift n; 0, otherwise

The physician scheduling model can be formulated as follows:

$$\text{Obj:}\quad \min\ z=\sum_{t=1}^{|T|}W_{1,t}+\alpha\sum_{t=1}^{|T|}c_{1,t}\Delta \tag{14}$$

$$s.t.\quad \sum_{n=m\cdot|N|-(|N|-1)}^{m\cdot|N|}x_{i,n}\leq 1,\quad \forall m\in\mathcal{M},\ i\in\mathcal{I} \tag{15}$$

$$x_{i,m\cdot|N|}+\sum_{n=m\cdot|N|+1}^{(m+1)\cdot|N|}x_{i,n}\leq 1,\quad \forall m\in\mathcal{M}\backslash\{7\},\ i\in\mathcal{I} \tag{16}$$

$$x_{i,7|N|}+\sum_{n=1}^{|N|}x_{i,n}\leq 1,\quad \forall i\in\mathcal{I} \tag{17}$$

$$\sum_{m=1}^{7} x_{i,m\cdot|N|} \leq C_{\max}, \quad \forall i \in \mathcal{I} \tag{18}$$

$$\sum_{m=1}^{7} x_{i,m\cdot|N|} \geq C_{\min}, \quad \forall i \in \mathcal{I} \tag{19}$$

$$\sum_{n=1}^{7|N|}\sum_{t=1}^{|T|} x_{i,n} r_{n,t} \leq H, \quad \forall i \in \mathcal{I} \tag{20}$$

$$c_{1,t} = \sum_{n=1}^{7|N|}\sum_{i=1}^{|K|} x_{i,n} r_{n,t}, \quad \forall t \in \mathcal{T} \tag{21}$$

$$c_{1,t} \geq 1, \quad \forall t \in \mathcal{T} \tag{22}$$

$$q_{1,t} = L_1(c_{1,t}, q_{1,t-1}, q_{2,t-1}, t), \quad \forall t \in \mathcal{T} \tag{23}$$

$$q_{2,t} = L_2(c_{1,t}, q_{1,t-1}, q_{2,t-1}, t), \quad \forall t \in \mathcal{T} \tag{24}$$

$$W_{1,t} = W_1(c_{1,t}, q_{1,t-1}, q_{2,t-1}, t), \quad \forall t \in \mathcal{T} \tag{25}$$

$$q_{1,0} = 0 \tag{26}$$

$$q_{2,0} = 0 \tag{27}$$

The objective function (14) minimizes the total patient waiting time in the physicians' queue and the staffing cost, where α is the parameter to balance two terms of the objective. Constraint (15) ensures that each physician works at most on one shift in one day. Constraint (16) and (17) guarantee that a physician rests at least 24 hours after performing a night shift. Note that constraint (17) ensures that a physician does not work on the first day of a week if the physician selects the night shift on the last day of the week. Constraints (18) and (19) ensure the limitation of the number of night shifts in one week of each physician. Constraint (20) ensures that the total working time of each physician is no more than H hours. Constraint (21) gives the number of physicians in each period. Constraint (22) guarantees that at least one physician is on duty for each period of the week. The function L_1, L_2, and W_1 in constraints (23)~(25) denote our computation approach of $q_{1,t}$, $q_{2,t}$, and $w_{1,t}$, respectively. Constraints (23) and (24) specify the computation of $q_{1,t}$ and $q_{2,t}$. Constraint (25) specifies the computation of $W_{1,t}$. Constraints (26) and (27) provide the initial state of the ED system.

The objective function is highly nonlinear because of the constraints (23)~(25), in which the system state $q_{1,t}$ and $q_{2,t}$, and the waiting time $W_{1,t}$ are computed. Because optimization solvers such as CPLEX or Gurobi cannot be used to solve the nonlinear model, we design a *pointwise technique* to linearize the scheduling model.

The linearization is implemented by assuming that the value of the initial states of station 1 and station 2 are integer multiple of a base constant φ. The linearization is performed with the help of the following parameters and variables:

Sets and Indices

j: index of the system state in station 1, $j \in \{0, 1, 2, \ldots, U/\varphi\}$

k: index of the system state in station 2, $k \in \{0, 1, 2, \ldots, V/\varphi\}$

l: index of shifts, $l \in \{1, 2, \ldots, |K|\}$

Parameters

U: possible maximal system state in station 1

V: possible maximal system state in station 2

φ: base constant of system state

Decision variables

$a_{j,t}$: 1, if $q_{1,t} = j\varphi$; 0, otherwise

$b_{k,t}$: 1, if $q_{2,t} = k\varphi$; 0, otherwise

$y_{l,t}$: 1, if $c_{1,t} = l$; 0, otherwise

$d_{j,k,l,t}$: 1, if $q_{1,t} = j\varphi$, $q_{2,t} = k\varphi$ and $c_{1,t} = l$; 0, otherwise

U/φ and V/φ should be integers. The constraints (23)~(25) can be represented by linear constraints (28)~(38).

$$\sum_{j=0}^{U/\varphi} a_{j,t} = 1 \ \ , \forall t \in \mathcal{T} \tag{28}$$

$$\sum_{j=0}^{U/\varphi} a_{j,t} \cdot j\varphi = q_{1,t}, \quad \forall t \in \mathcal{T} \tag{29}$$

$$\sum_{k=0}^{V/\varphi} b_{k,t} = 1, \quad \forall t \in \mathcal{T} \tag{30}$$

$$\sum_{k=0}^{V/\varphi} b_{k,t} \cdot k\varphi = q_{2,t}, \quad \forall t \in \mathcal{T} \tag{31}$$

$$\sum_{l=1}^{|K|} y_{l,t} = 1, \quad \forall t \in \mathcal{T} \tag{32}$$

$$\sum_{l=1}^{|K|} l \cdot y_{l,t} = c_{1,t}, \quad \forall t \in \mathcal{T} \tag{33}$$

$$\sum_{l=1}^{|K|}\sum_{j=0}^{U/\varphi}\sum_{k=0}^{V/\varphi} d_{j,k,l,t} = 1, \quad \forall t \in \mathcal{T} \tag{34}$$

$$\begin{aligned} d_{j,k,l,t} &\geq a_{j,t} + b_{k,t} + y_{l,t} - 2, \quad \forall t \in \mathcal{T}, \\ &\quad j \in \{1, \ldots, U/\varphi\}, \\ &\quad k \in \{1, \ldots, V/\varphi\}, l \in 1, \ldots, |K|\} \end{aligned} \tag{35}$$

$$\begin{aligned} q_{1,t+1} &= \sum_{l=1}^{|K|}\sum_{j=0}^{U/\varphi}\sum_{k=0}^{V/\varphi} d_{j,k,l,t} L_1(j\varphi, k\varphi, l, t), \\ &\quad \forall t \in \mathcal{T}\backslash\{|T|\} \end{aligned} \tag{36}$$

$$\begin{aligned} q_{2,t+1} &= \sum_{l=1}^{|K|}\sum_{j=0}^{U/\varphi}\sum_{k=0}^{V/\varphi} d_{j,k,l,t} L_2(j\varphi, k\varphi, l, t), \\ &\quad \forall t \in \mathcal{T}\backslash\{|T|\} \end{aligned} \tag{37}$$

$$\begin{aligned} W_{1,t} &= \sum_{l=1}^{|K|}\sum_{j=0}^{U/\varphi}\sum_{k=0}^{V/\varphi} d_{j,k,l,t} \cdot W_1(j\varphi, k\varphi, l, t), \\ &\quad \forall t \in \mathcal{T} \end{aligned} \tag{38}$$

Constraints (28)~(31) ensure that $q_{1,t}$ and $q_{2,t}$ patients are at the beginning of period t in station 1 and station 2. Constraints (32) and (33) ensure that c_t physicians work at the period t. Constraints (34) and (35) ensure that only one combination of $q_{1,t}$, $q_{2,t}$, and c_t is available. $L_1(j\varphi, k\varphi, l, t)$, $L_2(j\varphi, k\varphi, l, t)$ and $W_1(j\varphi, k\varphi, l, t)$ in constraints (36)~(38) give the value of $q_{1,t+1}$, $q_{2,t+1}$, and $W_{1,t}$ when the initial states and the number of physicians of period t are $j\varphi$, $k\varphi$ and l, respectively. Constraints (36) and (37) specify the computation of $q_{1,t}$ and $q_{2,t}$, while constraint (38) specifies the computation of $W_{1,t}$. As an illustration of the pointwise linearization method, we provide an example in part C of the appendix.

With the system state $q_{1,t}$, $q_{2,t}$ and the waiting time $W_{1,t}$ computation being substituted by the linearized constraints (28)~(38), the model can be transformed into a linearized model. Let MIP1 be the linearized physician scheduling model, which can be summarized as follows:

$$\begin{aligned}&\textit{Minimize } (14)\\&\textit{s.t. } (15) \sim (22), (26) \sim (38).\end{aligned}$$

Physician scheduling problems are NP-hard problems and computationally challenging even for small-size instances [11], [34], [35]. Because of the pointwise linearization, the size of the model increases dramatically. We take the second instance (week 2) of part C in section VI (numerical experiments) as an example: We use Gurobi as the mathematical solver and see that the MIP1 model has more than 9,982,700 rows and 11,126,300 columns with about 500 continuous variables and 11,125,800 integer variables. The huge size of the model makes it challenging for the commercial solver to solve the model, which is further shown in section VI. Therefore, we additionally design a heuristic algorithm to solve the physician scheduling problem.

V. Algorithm Design

In this section, we design a TS algorithm to solve the physician scheduling problem. The TS algorithm is a commonly used algorithm to solve discrete optimization problems. Successful examples of the TS algorithm's application in the physician scheduling problem can be found in Niroumandrad and Lahrichi [36] and Liu and Xie [37]. The general structure of the TS algorithm is shown in Fig. 7. The algorithm starts from an initial solution s^0. After that, the tabu list gets initialized, and s^0 is set as the current solution s. In each iteration, the algorithm generates the neighborhood of the current solution s and finds the best neighborhood solution s', which is either non-tabu or satisfies a specific aspiration criterion. Then the algorithm updates the tabu list and sets $s = s'$. The algorithm stops once a predefined stopping criterion is met.

A. Initial Solution, Neighborhood Structure and Tabu Move

The initial solution s^0 is generated by a greedy algorithm. First, we assign a physician to the first shift of the week and then randomly assign each time one physician to the shift that starts from the end of the last working shift, concerning the constraints of physician assignment (e.g., no physician works more than one shift in one day or two successive night shifts). This procedure is repeated until each period of a week has at least one physician working. Then, without violating the assignment constraints, we add the physician one by one to the shift that minimizes the objective function until no improvement is found by assigning a physician to any shift.

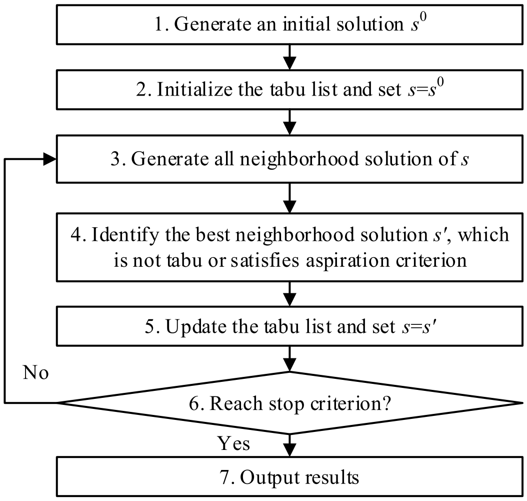
1. Generate an initial solution s^0
2. Initialize the tabu list and set s=s^0
3. Generate all neighborhood solution of s
4. Identify the best neighborhood solution s', which is not tabu or satisfies aspiration criterion
5. Update the tabu list and set s=s'
No
6. Reach stop criterion?
Yes
7. Output results

Fig. 7. Structure of the TS algorithm.

After generating the initial solution, we set it as the first current solution s and generate its neighborhood solutions. All neighborhood solutions of the current solution are generated by following two moves: (1) to assign one physician additionally to one shift; (2) to eliminate one assigned shift of one physician. The neighborhood solution, which satisfies the physician scheduling constraints, will be reserved, and the best feasible neighborhood solution s' will be identified and set as the new current solution.

To avoid the algorithm getting trapped in local optima during iterations, we define tabu moves in the algorithm. The principle of the tabu move is defined as follows: assume that in one iteration, s is the current solution and s' is the best feasible neighborhood solution at the end of the iteration, then the move that makes solution s' back to solution s are forbidden in the following θ iterations.

B. Solution Evaluation

The first trick is to use a rough evaluation. As stated in section III.A.2, the system state under a given staffing plan is computed by the bisection method, which makes the computation time of APP-length-2 highly dependent on the given tolerance of the bisection method: a small tolerance will bring an accurate result with a long computation time. In contrast, a bigger tolerance will cause a less accurate result but with a shorter computation time. Since the APP-time-1 method relies on the system state results computed by APP-length-2, we design two evaluations to accelerate the evaluation process. When calculating the patient waiting time by APP-time-1, we use the evaluation with 10^{-5} absolute errors in APP-length-2 as the "exact" evaluation and the evaluation with 10^{-1} absolute errors as the "semi-exact" evaluation. As shown in Fig. 8, a feasible neighborhood solution will get first semi-exact evaluated. If the objective value of the solution is better than that of the current best neighborhood solution, the solution will then be evaluated exactly. Numerical results show that the result of "semi-exact" evaluation approximates the value of the "exact" evaluation when evaluating the same

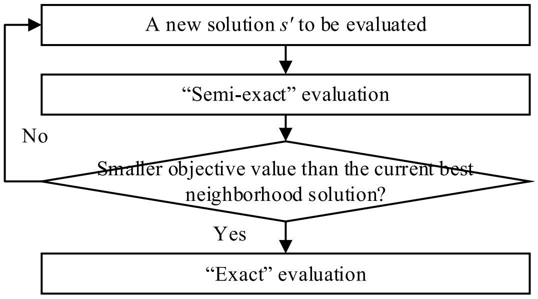
A new solution s' to be evaluated
"Semi-exact" evaluation
No
Smaller objective value than the current best neighborhood solution?
Yes
"Exact" evaluation

Fig. 8. Solution evaluation process.

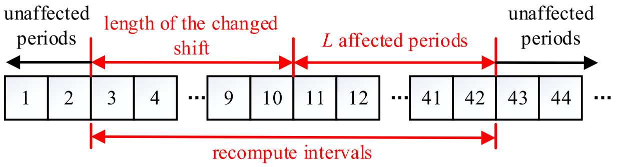

Fig. 9. Computation method in "semi-exact" evaluation.

solution, while the "semi-exact" evaluation is 5-6 times faster than the "exact" evaluation.

The second trick is that in the "semi-exact" evaluation we do not compute the system state and patient waiting time from the first period of a week but the first-changed period. In our algorithm, a neighborhood solution is obtained by changing one shift of one physician. This change has no impact on the system state and waiting time computation before the first-changing period and little impact on the system state and waiting time far from the changing periods. As shown in Fig. 9, we only recompute the system state and waiting time of changing periods and the consecutive L periods after the change. Through experiments, we set L to 32 to guarantee the evaluation precision of the system state and patient waiting time.

C. Aspiration and Stop Criterion

The algorithm uses a simple aspiration criterion. If the objective value of one neighborhood solution generated by a tabu move is better than the best-known solution, the tabu of the solution will be disabled. The stop criterion of the TS algorithm is the number of iterations. The algorithm will stop and output results after η iterations.

VI. Numerical Experiments

In this section, we first implement experiments to validate our proposed approximation methods of the system state and patient waiting time. The results of our approximations and simulations are compared. Next, we present computational experiments designed to assess the performance of the proposed TS algorithm. We compare the solutions of Gurobi 9.0 solver with the scheduling solutions of our TS algorithm. Furthermore, we compare our TS solutions with the real-life physician schedule and another method in the literature. The real-life physician schedule and patient arrival patterns are used in the numerical experiments except for part C, in which we cut the patient arrival rate by half to make it possible to solve the model by a MIP solver. All simulations are performed by simulation software Anylogic 8.5 with 50,000 replications. The TS algorithm is implemented in C++. All algorithms are run on a 3.1 GHz CPU, 512 GB memory computer and Win10 operation system.

TABLE II

Parameters of Experiments to Validate the Approximation Method

Symbol	Value	Explanation
Δ	1	Length of each period
μ_1	10.93	Service rate of station 1
μ_2	2.5	Service rate of station 2
c_2	10	Number of servers of station 2
p	0.55	Probability of patients requiring exams
$\underline{\rho}$	2	Parameter defining the domain of (9)
$\bar{\rho}$	2.5	Parameter defining the domain of (9)

A. System State Approximations vs. Simulation

e first compare system state approximations (APP-length-1 and APP-length-2) with simulation results to check the accuracy of approximation methods. We use real-life data from Wuhan hospital to perform the numerical experiments. Six weeks of operation data are selected. The typical pattern of the daily arrival rates of patients is shown in Fig. 1. The ED takes a four-shift working schedule for the physicians. Four shifts are 8:00-16:00, 9:00-17:00, 17:00-1:00, and 1:00-9:00, while one, two, one, and two physicians are assigned to such four shifts, respectively. Other parameters of the experiments, e.g., the duration of a period and the service rate, are listed in Table II. We use the actual physician staffing to implement the experiments. All experiments begin with null patients in both stations. For each week, the system states at the end of each period are obtained by three methods (APP-length-1 and APP-length-2, and simulation).

Table III presents the numerical results. Columns "APP-length-1", "APP-length-2", and "Simulation" represent the total system state (i.e., the sum of system states at the end of each period) in the physicians' queue computed by two approximations and simulation. Column "Gap$_{1-s}$" gives the percentage deviation between the total system state of APP-length-1 and the simulation results, and "Gap$_{2-s}$" indicates deviation between APP-length-2 and simulation. The deviation between APP-length-1 and simulation is computed by $100\times|\mathit{APP\text{-}length\text{-}}1-\mathit{Simulation}|/\mathit{Simulation}$, while a similar formula is used to calculate Gap$_{2-s}$. To show the hourly system states of different methods, we use week 2 and week 3 as examples, and the results are illustrated in Fig. 10 and 11.

Note that APP-length-1 can only evaluate the system state of an $M_t/M/c_t$ queuing system without returns, so we modify our model to an $M_t/M/c_t$ queuing system when applying the APP-length-1 method. As stated in section III.B.1, the effect of the returning patients can be taken as either the amplification of the patient arrival rate or the minification of the physician service rate [16]. Therefore, we consider our queuing model

TABLE III
TOTAL SYSTEM STATE BY APPROXIMATIONS AND SIMULATION

Instance	APP-length-1	APP-length-2	Simulation	$\text{Gap}_{1\text{-s}}$ (%)	$\text{Gap}_{2\text{-s}}$ (%)
Week 1	827.44	1635.95	1622.59	49.00	0.82
Week 2	608.55	1177.64	1154.87	47.31	1.97
Week 3	651.29	1248.56	1233.55	47.20	1.22
Week 4	1440.13	1954.90	1919.09	24.96	1.87
Week 5	1600.20	2226.44	2209.93	27.59	0.75
Week 6	1391.32	1946.37	1905.43	26.98	2.15
Average	1086.49	1698.31	1674.24	37.17	1.46

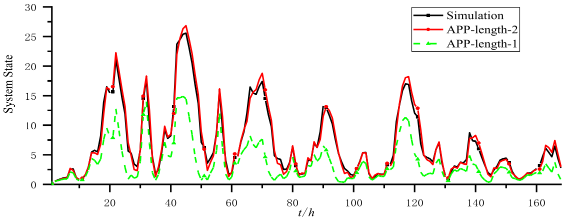

Fig. 10. Hourly system state in the physicians' queue of week 2.

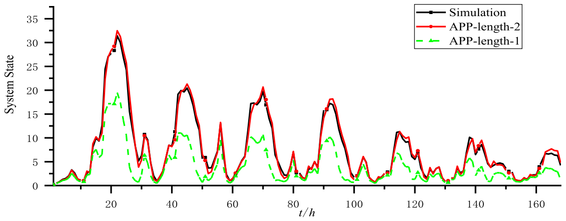

Fig. 11. Hourly system state in the physicians' queue of week 3.

as an $M_t/M/c_t$ queuing system with a minified service rate in the experiments, i.e., the APP-length-1 method evaluates the system state with parameters $(\lambda_t, \mu', c_{1,t})$.

From the results, we find that APP-length-2 is close to the simulation and has higher accuracy than APP-length-1. As shown in Table III, the total system state of APP-length-2 is close to the simulation results for each week's data, with a relative deviation below 3%. For all six weeks of data, the average total system state over six weeks is 1674.24, while the simulation result is 1698.31. The average gap of all six instances is 1.46%. Meanwhile, from Table III we observe that APP-length-2 tends to overestimate the total number of patients slightly, but such overestimates are acceptable. Such results verify that our APP-length-2 can approximate the real system states well. On the contrary, the system state of APP-length-1 is not as good as APP-length-2. With a mean deviation of 37.17%, the APP-length-1 method underestimates the system state, which demonstrates that the APP-length-1 cannot reasonably approximate the system state for our model.

From Fig. 10 and 11, we can further observe that APP-length-2 is very close to the simulation results in terms of the system state and much better than APP-length-1 results. Thus, the APP-length-2 approximation method is adopted as the system state evaluation method in our optimization algorithm.

TABLE IV
TOTAL WEEKLY WAITING TIME BY APP-TIME-1 AND SIMULATION

Instance	APP-time-1	Simulation	Gap (%)
Week 1	1329.09	1360.15	2.28
Week 2	897.53	916.16	2.03
Week 3	978.03	999.83	2.18
Week 4	1671.10	1660.10	0.66
Week 5	1934.33	1943.79	0.49
Week 6	1652.56	1639.26	0.81
Average	1410.44	1419.88	1.41

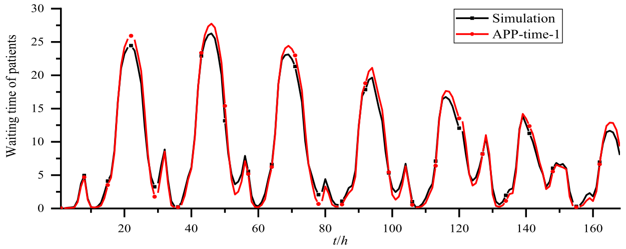

Fig. 12. Hourly waiting time of patients in the physicians' queue of week 1.

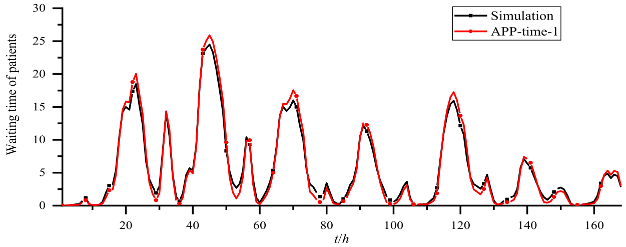

Fig. 13. Hourly waiting time of patients in the physicians' queue of week 2.

B. Waiting Time Approximations vs. Simulation

This subsection compares APP-time-1 with simulation results to present the precision of our waiting time computation method. We also use the real patient arrival rate and the actual hospital staffing to validate APP-time-1. All experiments begin with null patients in both stations. In Table IV, columns "APP-time-1" and "Simulation" are the total waiting time in the physicians' queue of each week (i.e., the sum of waiting time of each period) computed by APP-time-1 and simulation, respectively; column "Gap" gives the percentage difference between the total waiting time of APP-time-1and simulation. Fig. 12 and 13 use week 1 and week 2 as examples to show the hourly patient waiting time in the physicians' queue using APP-time-1 and simulation.

The experiment results show that the total waiting time computed by APP-time-1 is very close to the simulation. The average gap of the total weekly patient waiting time is 1.41%. Fig. 12 and 13 show that APP-time-1 can well approximate the hourly patient waiting time. From the results we can conclude that APP-time-1 can be used as the evaluation method in our TS algorithm.

C. Compare With Commercial MIP Solver

This subsection addresses the physician scheduling acquired by the TS algorithm and Gurobi solver. Due to the model size

TABLE V
PARAMETERS OF NUMERICAL EXPERIMENTS

Parameter	Value	Explanation
$\|K\|$	9	Number of available physicians
α	1	Parameter in objective function
$C_{\max}$	2	Upper bound of the number of night shifts
$C_{\min}$	0	Lower bound of the number of night shifts
H	50	Maximum number of weekly working hours
θ	10	Tabu length of the TS algorithm
η	300	The number of iterations of the TS algorithm
err_semi	10^{-1}	Absolute error in semi-exact evaluations of the TS algorithm
err_exact	10^{-5}	Absolute error in exact evaluations of the TS algorithm

limitation of Gurobi, we reduce the actual patient arrival rate at each period by half to limit the size of the MIP1 model. Based on such reduced arrival rates, we set $\varphi = 1$, $U = 160$, and $V = 40$ in MIP1. Besides the parameters shown in Table II, other parameters used in the numerical experiments are listed in Table V. However, not surprisingly, even for such small-size instances, we find Gurobi still very hard to obtain the optimal solution. So, we set a stopping criterion to Gurobi that it stops once the relative MIP gap is smaller than 3% or the solution time reaches 12 hours.

Table VI shows the comparison between TS and Gurobi solutions. For the TS algorithm, the objective value of formula (14) (column "Objective value"), total patient waiting time in the physician's queue ("Waiting time"), and computation time of the solutions ("CPU") are given. The relative MIP gap ("Rel. gap"), which is the gap of the lower and upper bound computed by Gurobi, is also listed. Column "Gap-1" shows the percentage deviation of the objective value between the TS and Gurobi solution, which is computed by 100×|objective value of TS solution−objective value of Gurobi solution|/objective value of Gurobi solution. With a similar formula, "Gap-2" gives the deviation of patient waiting time between the TS and Gurobi solution. Note that the results of the total patient waiting time in the physicians' queue is obtained by simulation, based on the staffing derived from TS and Gurobi solutions.

Table VI shows that Gurobi can get solutions within a solution time of 12 hours in all six instances, but none of the solutions is optimal, while the TS solutions are much better than Gurobi solutions in both computation time and solution quality. For all six instances, the average computation time of the TS algorithm is 0.53 hours, which is about 95% less than Gurobi when solving the same instance. In terms of the objective function composed of patient waiting time and physician working time, the TS algorithm can get an averagely 44.3% smaller objective value than that of Gurobi. The TS algorithm's superiority is even more obvious when only regarding the patient waiting time: the total waiting time of the TS solution is averagely 74.9% less than the Gurobi solution. These results show that our TS algorithm can get better solutions than Gurobi in our scheduling problem.

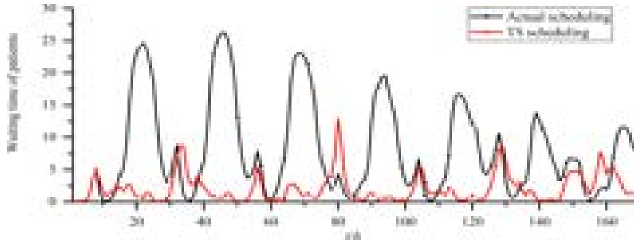

Fig. 14. Hourly patient waiting time of TS solution and actual hospital scheduling of week 1.

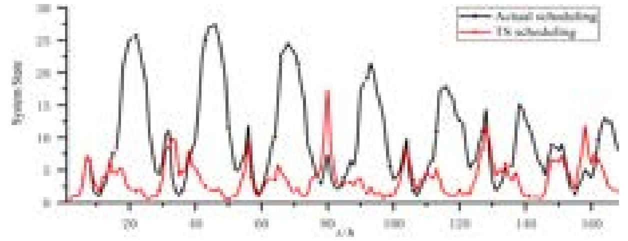

Fig. 15. Hourly system state of TS solution and actual hospital scheduling of week 1.

D. Compare With Actual Hospital Schedule

We compare the physician scheduling of TS solutions and the real-life scheduling used in the hospital. Real-life patient arrival data are used to conduct the experiments. Table VII shows the objective value (column "Objective value") and total patient waiting time in the physician's queue ("Waiting time") of the TS solutions and the real-life scheduling. The computation time ("CPU") of the TS is also shown in Table. The results of total patient waiting time in the physicians' queue are obtained by simulation. Columns "Gap-1" and "Gap-2" show the percentage deviation of the objective value and total waiting time between the TS solution and the actual hospital schedule.

The hourly patient waiting time and system state in the physicians' queue of week 1 of the TS solution and the actual hospital scheduling are shown in Fig. 14 and 15. The hourly number of physicians of the TS solution and the actual scheduling with daily patient arrival patterns are shown in Fig. 16 and 17. We compare the staffing levels because the system state and patient waiting time are determined by the physician staffing levels.

Table VII shows that the TS solutions have smaller objective values and total patient waiting time than the actual scheduling. The average objective value of TS solutions is 52.8% less than that of the actual scheduling. The average total patient waiting time of six instances is 390.3, which is over 70% lower than that of the actual scheduling. The average computation time of our TS algorithm is about 0.6 hours. Among all six instances, week 6 has the longest computation time with 0.67 hours.

The TS scheduling can effectively reduce not only the patient waiting time but also the expected number of patients, as shown in Fig. 14 and 15. Compared with the actual hospital schedule, the TS solution achieves a shorter patient waiting

TABLE VI
COMPARISON BETWEEN TS AND GUROBI SOLUTIONS

Instance	TS Solution			Gurobi Solution			Gap-1 (%)	Gap-2(%)
	Objective value	Waiting time	CPU (h)	Objective value	Waiting time	Rel. gap (%)		
Week 1	376.80	96.80	0.43	682.55	450.55	55.70	44.80	78.52
Week 2	356.34	100.34	0.48	527.97	247.97	59.53	32.51	59.54
Week 3	347.84	99.84	0.57	556.57	356.57	53.55	37.50	72.00
Week 4	384.25	112.25	0.54	732.70	524.70	53.69	47.56	78.61
Week 5	370.72	90.72	0.57	835.98	595.98	53.44	55.65	84.78
Week 6	408.43	136.43	0.58	779.68	571.68	54.32	47.62	76.14
Average	374.06	106.06	0.53	685.91	457.91	55.04	44.27	74.93

TABLE VII
COMPARISON BETWEEN TS SOLUTIONS AND ACTUAL HOSPITAL SCHEDULING

Instance	TS Solution			Actual Scheduling		Gap-1 (%)	Gap-2(%)
	Objective value	Waiting time	CPU (h)	Objective value	Waiting time		
Week 1	744.48	352.48	0.60	1688.63	1352.63	55.91	73.94
Week 2	742.72	350.72	0.56	1250.65	914.65	40.61	61.66
Week 3	604.52	212.52	0.62	1333.01	997.01	54.65	78.68
Week 4	1000.51	592.51	0.59	1910.81	1574.81	47.64	62.38
Week 5	902.37	478.37	0.60	2149.71	1813.71	58.02	73.62
Week 6	771.21	355.21	0.67	1915.98	1579.98	59.75	77.52
Average	794.30	390.30	0.61	1708.13	1372.13	52.76	71.30

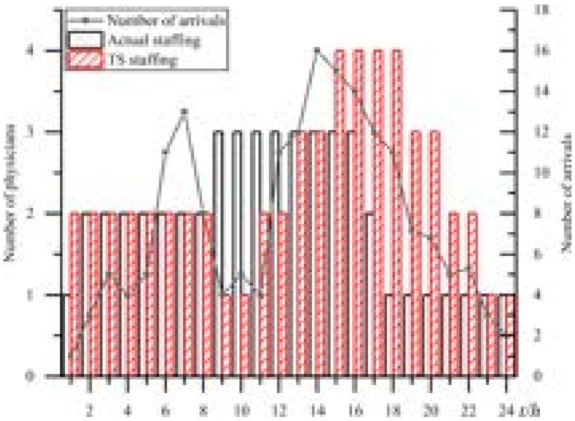

Fig. 16. Patient arrival rate and physician staffing of day 1 of week 1.

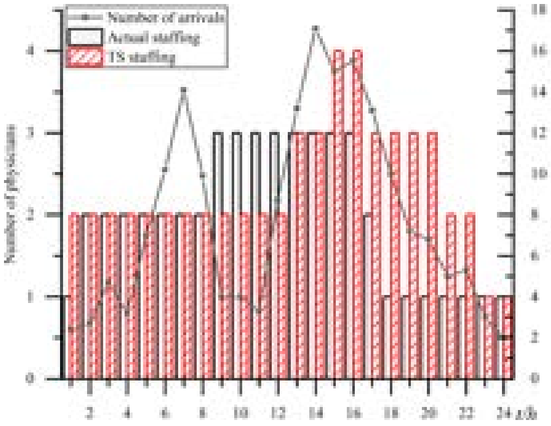

Fig. 17. Patient arrival rate and physician staffing of day 2 of week 1.

time in most periods except for a small number of periods (like periods 9-14, 33-38, 77-84). The maximal hourly waiting time is reduced from 26.3 to 12.9, while the maximal hourly system state is reduced from 27.4 to 17.4. The number of periods where the system state is greater than 15 is reduced from 39 to 1.

Fig. 16 and 17 show that the TS scheduling can better match the fluctuation of the patient arrival rate than the actual hospital scheduling. The TS scheduling increases the number of physicians in periods with high patient arrival rates (such as periods 15 and 16) and decreases the number of physicians in periods with low patient arrival rates (like periods 9, 10, and 11). These physicians' assignments make it possible to reduce the total number of patients and waiting time of patients without increasing the available number of physicians.

E. Compare With the Method in the Literature

Although numerous studies address the physician staffing and scheduling problem, there is little research that addresses the physician scheduling problem under time-varying demands of patients with returns. To show the performance of our method, we compare our TS algorithm with the method in Yom-Tov and Mandelbaum [24].

The queuing model in Yom-Tov and Mandelbaum [24] is similar to ours, i.e., the planning horizon is divided into several periods, and for each period the patient arrives according to a Poisson distribution and returns probably after an exponential delay. Yom-Tov and Mandelbaum [24] study the computational methods of the offered load $R_{1,t}$ and $R_{2,t}$ of their queuing model and propose a staffing method based on the square-root-staffing (SRS) formula: $s_t = R_{1,t} + \beta\sqrt{R_{1,t}}$, where β is a parameter related to the waiting probability of patients in the

TABLE VIII
COMPARISON BETWEEN TS AND ERLANG-R STAFFING

Instance	TS Solution			Erlang-R Staffing			Gap-1 (%)	Gap-2(%)
	Objective value	Waiting time	Labor time	Objective value	Waiting time	Labor time		
Week 1	544.36	127.36	417	647.61	286.61	361	18.97	125.04
Week 2	508.77	115.77	393	624.79	291.79	333	22.80	152.04
Week 3	501.73	116.73	385	612.82	283.82	329	22.14	143.14
Week 4	526.53	126.53	400	617.54	264.54	353	17.28	109.07
Week 5	536.46	122.46	414	628.44	263.44	365	17.15	115.12
Week 6	535.39	125.39	410	658.10	307.10	351	22.92	144.92
Average	525.54	122.37	403.17	631.55	282.88	348.67	20.21	131.56

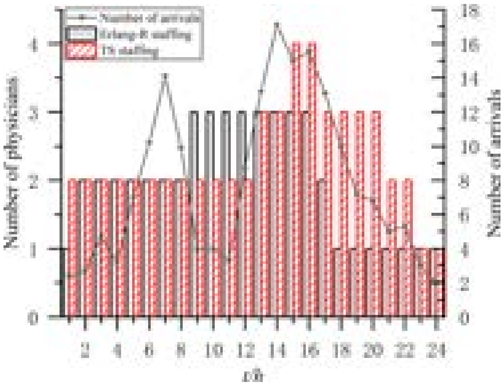

Fig. 18. Patient arrival rate and physician staffing of day 3 of week 1.

physician's queue. The staffing level is obtained by rounding up the computed s_t. Particularly, if the number of physicians of a period is zero, we set the staffing level of the period to one. Evidently, the bigger the parameter β is, the greater the total labor time and the smaller the total patient waiting time is. In this part, we set β to 0.5 to avoid getting a too large or too small staffing level.

Compared with our work, Yom-Tov and Mandelbaum [24] do not consider the physician scheduling constraints. Therefore, considering the difference, we implement the comparison experiments by comparing the staffing plan. The staffing plan of our method is obtained from the TS algorithm by relaxing the scheduling constraints. The minimal number of physicians in each period constraint is the only reserved constraint.

Table VIII presents the objective value, total waiting time and percentage deviation between our TS staffing and the Erlang-R method. The total labor time of both methods is also presented. The deviation of the objective value between the TS solution and the Erlang-R staffing is calculated by 100×|TS−Erlang-R|/TS and shown in column "Gap-1". Column "Gap-2" shows the deviation of the total patient waiting time between both methods. The hourly number of physicians of the TS solution and Erlang-R staffing, as well as the daily patient arrival patterns, are shown in Fig. 18.

From Table VIII, we can see that the TS algorithm can get better staffing than the Erlang-R method in terms of the objective value and the total waiting time. In all six instances, the patient waiting time of the TS solution is considerably shorter than the Erlang-R method. The deviation of the patient waiting time between both methods is surprisingly over 130% on average, although our TS solutions have 54.5 hours (about 13.5%) higher labor time averagely. Besides, compared with the Erlang-R staffing plan, the TS solution reduces the objective value by 20.2% on average. Figure 18 shows that our TS staffing plan can better fluctuate with the time-varying arrival rates of patients, while the Erlang-R staffing plan cannot well follow the fluctuation of the patient arrivals. The results show that our proposed TS algorithm is more effective for solving our physician scheduling problem.

F. Extension to Non-Exponential Service Time

In the above model, the patients' inter-arrival time and service time are assumed exponentially distributed, which is a common assumption in many relative works. However, the data from our partner hospital show that the exponential distribution (especially the service time distribution) assumption is sometimes invalid. We individually use the *degenerate distribution* and *general distribution* to model the service time distributions of both stations. Based on the TS solutions in the previous subsection and the actual hospital scheduling, we obtain the system performance metrics, i.e., the total waiting time of patients, under non-exponential service time by simulation. In our simulation, we generate a distribution pool for the *general distribution* by mixing five different distributions: Erlang distributions, gamma distributions, beta distributions, Pareto distributions and truncated normal distributions. The mean and variance of the five distributions as well as the *degenerate distribution* are all adjusted to the corresponding exponential distributions (i.e., μ_1 and μ_2 in TABLE II). For each patient, the general service time is generated by selecting a random distribution from the distribution pool and creating a corresponding service time. In other words, five distributions are combined to represent the *general distribution* of patient service times in the simulations.

Table IX shows the objective value (column "z") and total waiting time of patients in the physicians' queue ("Waiting time") with TS schedule and actual scheduling when the service time of both stations are degenerately distributed; Table X gives the results with the generally distributed service times. Columns "Gap-1" and "Gap-2" show the percentage deviation of the objective value and total waiting time between the TS solution and the actual hospital scheduling, respectively.

We observe when the service time is degenerately distributed, the TS solution reduces on average 53.1% objective value and 74.2% total waiting time of the actual scheduling. Table X shows that when the service time is generally distributed, the

TABLE IX

Comparison Between TS Solutions and Actual Hospital Scheduling for Degenerate Service Time

Instance	TS Solution		Actual scheduling		Gap-1 (%)	Gap-2 (%)
	z	Waiting time	z	Waiting time		
Week 1	646.2	254.2	1490.2	1154.2	56.6	78.0
Week 2	672.0	280.0	1092.5	756.5	38.5	63.0
Week 3	537.9	145.9	1187.9	851.9	54.7	82.9
Week 4	926.8	518.8	1807.7	1471.7	48.7	64.8
Week 5	823.2	399.2	2026.6	1690.6	59.4	76.4
Week 6	705.7	289.7	1801.1	1465.1	60.8	80.2
Average	718.7	314.7	1567.7	1231.7	53.1	74.2

TABLE X

Comparison Between TS Solutions and Actual Hospital Scheduling for General Service Time

Instance	TS Solution		Actual scheduling		Gap-1 (%)	Gap-2 (%)
	z	Waiting time	z	Waiting time		
Week 1	692.2	1576.1	300.2	1240.1	56.1	75.8
Week 2	704.7	1159.0	312.7	823.0	39.2	62.0
Week 3	569.7	1251.0	177.7	915.0	54.5	80.6
Week 4	958.3	1852.6	550.3	1516.6	48.3	63.7
Week 5	859.0	2077.9	435.0	1741.9	58.7	75.0
Week 6	735.9	1850.7	319.9	1514.7	60.2	78.9
Average	753.3	1627.9	349.3	1291.9	52.8	72.7

objective value and total waiting time decrease correspondingly by 52.8% and 72.7% on average. The results indicate that the TS solutions are still better than the actual scheduling when the service time is non-exponentially distributed.

VII. Conclusion

We address the ED physician scheduling problem with time-varying patient arrivals and returns. To evaluate the system performance of the ED system, we model the ED as a time-varying queuing network system with returns and propose a set of approximation methods to compute the system state and patient waiting time. For the system state computation, we extend the PSFFA method to our queuing model and design different approximation methods for various cases. The calculation of patient waiting time is based on the computed system state. By classifying the patients into three groups, we analyze the computation of each group's patient waiting time and then obtain the total patient waiting time. Based on the approximation methods and a pointwise linearization technique, we formulate a MIP model for the physician scheduling problem and design a TS algorithm to solve the model. Numerical results prove the precision of our proposed approximation methods and the efficiency of our TS algorithm. Results on real-life data show that our TS scheduling can effectively improve the real-life hospital schedule and reduce the patient waiting time.

This work can be extended in several directions. First, many hospitals adopt the monthly scheduling of their physicians. An extension of the planning horizon will make the scheduling problem more complicated because of expanding the scale and adding more scheduling constraints. Second, emergency patients are classified after triage. Even within the non-urgent patient group, some patients will have higher priority because of worse conditions. An extension of our method to cope with the prioritized patients is an interesting research topic.

Appendix

A. Toy Example of "APP-Length-2" Computation Method

To explain the "APP-Length-2" computation method clearly, We take the computation of two periods as an example. Assume that the length of each period is one hour. During each hour, patients arrive according to a Poisson process. The external patient arrival rates λ_1 and λ_2 are 15.6 and 5.1, respectively. In the first period, the number of physicians is two, while one physician works in the second period. The examination department has ten servers working for each period. The service times are exponentially distributed with mean service rate $\mu_1 = 10.93$ and $\mu_2 = 2.5$. The probability that a patient returns is 0.55. Assume that the system state at the initial moment is zero at both stations.

In our computation method, $q_{1,t}$ and $q_{2,t}$ are approximated by $l^{M/M/c}(\rho_{1,t}, 2)$ and $l^{M/M/c}(\rho_{2,t}, 10)$. Therefore, Eq. (6) and (7) for the first period can be rewritten as follows:

$$
\begin{aligned}
l^{M/M/c}(\rho_{1,t}, 2) &+ 2 \times 10.93 \times \rho_{1,t} \times 1\\
&= 0 + 15.6 \times 1 + 10 \times 2.5 \times \rho_{2,t} \times 1\\
l^{M/M/c}(\rho_{2,t}, 10) &+ 10 \times 2.5 \times \rho_{2,t} \times 1\\
&= 0 + 0.55 \times 2 \times 10.93 \times \rho_{1,t} \times 1
\end{aligned}
$$

Using the "bisection method", we can numerically solve $\rho_{1,t}$ and $\rho_{2,t}$, and then compute $q_{1,t}$ and $q_{2,t}$ by substituting $\rho_{1,t}$ and $\rho_{2,t}$ into $l^{M/M/c}(\rho_{1,t}, 2)$ and $l^{M/M/c}(\rho_{2,t}, 10)$, respectively. The results are as follows: $\rho_{1,t} = 0.813$, $\rho_{2,t} = 0.279$, $q_{1,t} = 4.805$, $q_{2,t} = 2.794$ $(t = 1)$.

For the second period, Eq. (6) and (7) can be rewritten as follows:

$$
\begin{aligned}
l^{M/M/c}(\rho_{1,t}, 1) &+ 1 \times 10.93 \times \rho_{1,t} \times 1\\
&= 4.805 + 5.1 \times 1 + 10 \times 2.5 \times \rho_{2,t} \times 1\\
l^{M/M/c}(\rho_{2,t}, 10) &+ 10 \times 2.5 \times \rho_{2,t} \times 1\\
&= 2.794 + 0.55 \times 1 \times 10.93 \times \rho_{1,t} \times 1
\end{aligned}
$$

Similarly, the equations for the second period can also be solved numerically. The results are as follows: $\rho_{1,t} = 0.861$, $\rho_{2,t} = 0.228$, $q_{1,t} = 6.188$, $q_{2,t} = 2.277$ $(t = 2)$. Using the same way, we can calculate the system states $q_{1,t}$ and $q_{2,t}$ from the first period to the last one.

B. Analysis of the Computation Method of "APP-Length-2"

Eq. (6) and (7) can be rewritten from the functional perspectives as follows:

$$C_1 = f_1(\rho_{1,t}) - k_2\rho_{2,t}, \tag{i}$$

$$C_2 = -k_1\rho_{1,t} + f_2(\rho_{2,t}), \tag{ii}$$

where $C_i \geq 0$, and $k_i > 0$ $(i \in \{1, 2\})$. In (i) and (ii), $f_1(\rho_{1,t}) = l^{M/M/c}(\rho_{1,t}, c_{1,t}) + c_{1,t}\mu_1\rho_{1,t}\Delta$, $f_2(\rho_{2,t}) = l^{M/M/c}(\rho_{2,t}, c_2) + c_2\mu_2\rho_{2,t}\Delta$.

Because ρ_t is in the range of [0, 1) and $l^{M/M/c}(\rho_t, c_t)$ is monotonically increasing and convex function of ρ_t, f_i

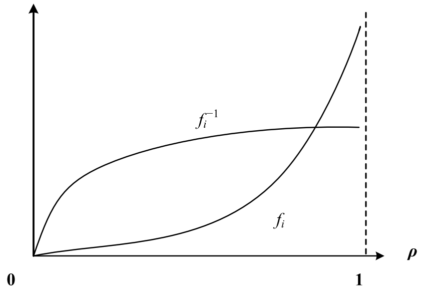

Fig. 19. Graph of $f_1(\rho_{1,t})$ and $k_2 f_2^{-1}(C_2 + k_1\rho_{1,t})$.

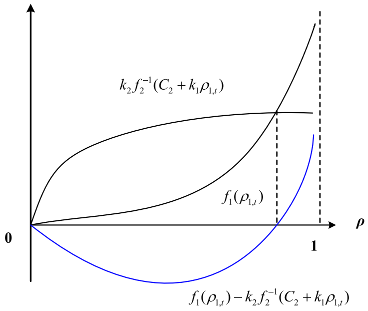

Fig. 20. Graph of f_i and f_i^{-1}.

is also monotonically increasing and convex function of ρ_t. Therefore, f_i is reversible, i.e., f_1^{-1} and f_2^{-1} exist, and it is clear that f_1^{-1} and f_2^{-1} are also monotonically increasing, but concave (as depicted in Fig. 19).

From (ii), we can have

$$\rho_{2,t} = f_2^{-1}(C_2 + k_1\rho_{1,t}). \quad \text{(iii)}$$

Substituting (iii) into (i), we have

$$C_1 + k_2 f_2^{-1}(C_2 + k_1\rho_{1,t}) = f_1(\rho_{1,t}). \quad \text{(iv)}$$

Let $g(\rho_{1,t}) = C_1 + k_2 f_2^{-1}(C_2 + k_1\rho_{1,t})$ and $h(\rho_{1,t}) = f_1(\rho_{1,t})$. Since $\rho_{1,t} \in [0, 1)$, we have

$$g(\rho_{1,t}) \in \Big[C_1 + k_2 f_2^{-1}(C_2), C_1 + k_2 f_2^{-1}(C_2 + k_1)\Big),$$
$$h(\rho_{1,t}) \in [0, +\infty).$$

Because $g(0) > h(0)$ and $g(1^-) < h(1^-)$, there exists exactly one $\rho_{1,t}$ such that (iv) holds. Because of (iii), there exists exactly one $\rho_{2,t}$.Therefore, the unique solution of (6) and (7) always exists. Next, we show that the solution can be solved using the proposed bisection method.

From (iv), C_1 can be rewritten as follows:

$$C_1 = f_1(\rho_{1,t}) - k_2 f_2^{-1}(C_2 + k_1\rho_{1,t}) \geq 0. \quad \text{(v)}$$

The graph of $f_1(\rho_{1,t})$ and $k_2 f_2^{-1}(C_2 + k_1\rho_{1,t})$ over [0, 1) is shown in Fig. 20. From Fig. 20, we can see that $f_1(\rho_{1,t}) - k_2 f_2^{-1}(C_2 + k_1\rho_{1,t})$ is monotonically increasing in the range that $f_1(\rho_{1,t}) \geq k_2 f_2^{-1}(C_2 + k_1\rho_{1,t})$. Therefore, $\rho_{1,t}$ can be solved by the bisection method. In the same way, $\rho_{2,t}$ can also be solved by the bisection method.

TABLE XI
VALUE OF $q_{1,t+1}$

k	$l = 1$			$l = 2$		
	$j = 0$	$j = 1$	$j = 2$	$j = 0$	$j = 1$	$j = 2$
0	0.523	0.725	0.967	0.378	0.482	0.592
1	0.817	1.076	1.380	0.525	0.638	0.760
2	1.192	1.516	1.889	0.686	0.811	0.946

C. Toy Example of the Pointwise Linearization Method

We use the linearization of $q_{1,t+1}$ as an example of our pointwise linearization method. We assume a period t with the length of one hour. During each hour, patients arrive according to a Poisson process with a mean of 2.8. The examination department has ten servers working for each period. The service times are exponentially distributed with mean service rate $\mu_1 = 10.93$ and $\mu_2 = 2.5$. We assume that φ is equal to one and set the maximal system state in stations 1 and 2 to two, i.e., $U = V = 2$, so $j \in \{0, 1, 2\}$, $k \in \{0, 1, 2\}$. Assume that $|K| = 2$, therefore, $l \in \{1, 2\}$. Constraint (36) specifies the value of $q_{1,t+1}$ by multiplying variable $d_{j,k,l,t}$ by the pre-calculated value of $q_{1,t+1}$ with different combinations of j, k, l and t. We calculate the value of $q_{1,t+1}$ using our "APP-Length-2" computation method.

Table XI shows the different values of $q_{1,t+1}$ (the system state of the period's end) with different inputs. For example, for period t, if the MIP solver get the following results: $d_{0,1,2,t} = 1$ while the rest variable $d_{j,k,l,t} = 0$, i.e., the system states at the start of the period $j = 0$, $k = 1$ and the number of physicians $l = 2$. The MIP solver calculates the value of $q_{1,t+1}$ as follows:

$$\begin{aligned} q_{1,t+1} &= 0 \times 0.523 + 0 \times 0.725 + \cdots + 1 \times 0.525 + \cdots \\ &\quad + 0 \times 0.946 \\ &= 0.525. \end{aligned}$$

REFERENCES

[1] J. Kennedy, K. Rhodes, C. A. Walls, and B. R. Asplin, "Access to emergency care: Restricted by long waiting times and cost and coverage concerns," *Ann. Emergency Med.*, vol. 43, no. 5, pp. 567–573, May 2004.

[2] O. El-Rifai, T. Garaix, V. Augusto, and X. Xie, "A stochastic optimization model for shift scheduling in emergency departments," *Health Care Manage. Sci.*, vol. 18, no. 3, pp. 289–302, Sep. 2015.

[3] M. Erhard, J. Schoenfelder, A. Fügener, and J. O. Brunner, "State of the art in physician scheduling," *Eur. J. Oper. Res.*, vol. 265, no. 1, pp. 1–18, Feb. 2018.

[4] J. F. Bard, Z. Shu, and L. Leykum, "A network-based approach for monthly scheduling of residents in primary care clinics," *Oper. Res. Health Care*, vol. 3, no. 4, pp. 200–214, Dec. 2014.

[5] R. Baum, D. Bertsimas, and N. Kallus, "Scheduling, revenue management, and fairness in an academic-hospital radiology division," *Academic Radiol.*, vol. 21, no. 10, pp. 1322–1330, Oct. 2014.

[6] R. Bruni and P. Detti, "A flexible discrete optimization approach to the physician scheduling problem," *Oper. Res. Health Care*, vol. 3, no. 4, pp. 191–199, Dec. 2014.

[7] J. O. Brunner, J. F. Bard, and R. Kolisch, "Midterm scheduling of physicians with flexible shifts using branch and price," *IIE Trans.*, vol. 43, no. 2, pp. 84–109, Nov. 2010.

[8] J. Puente, A. Gómez, I. Fernández, and P. Priore, "Medical doctor rostering problem in a hospital emergency department by means of genetic algorithms," *Comput. Ind. Eng.*, vol. 56, no. 4, pp. 1232–1242, May 2009.

[9] L. Rosocha, S. Vernerova, and R. Verner, “Medical staff scheduling using simulated annealing,” *Qual. Innov. Prosperity*, vol. 19, no. 1, pp. 1–8, Jul. 2015.

[10] R. Liu and X. Xie, “Physician staffing for emergency departments with time-varying demand,” *INFORMS J. Comput.*, vol. 30, no. 3, pp. 588–607, Aug. 2018.

[11] A. Fügener and J. O. Brunner, “Planning for overtime: The value of shift extensions in physician scheduling,” *INFORMS J. Comput.*, vol. 31, no. 4, pp. 732–744, Oct. 2019.

[12] K. Ghanes *et al.*, “Simulation-based optimization of staffing levels in an emergency department,” *Simulation*, vol. 91, no. 10, pp. 942–953, 2015.

[13] H. Guo, S. Gao, K.-L. Tsui, and T. Niu, “Simulation optimization for medical staff configuration at emergency department in Hong Kong,” *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 4, pp. 1655–1665, Oct. 2017.

[14] L. Vanbrabant, K. Braekers, and K. Ramaekers, “Improving emergency department performance by revising the patient–physician assignment process,” *Flexible Services Manuf. J.*, vol. 33, no. 3, pp. 783–845, Sep. 2021.

[15] Y. H. Kuo, “Integrating simulation with simulated annealing for scheduling physicians in an understaffed emergency department,” *HKIE Trans.*, vol. 21, no. 4, pp. 253–261, Oct. 2014.

[16] M. A. Ahmed and T. M. Alkhamis, “Simulation optimization for an emergency department healthcare unit in Kuwait,” *Eur. J. Oper. Res.*, vol. 198, no. 3, pp. 936–942, Nov. 2009.

[17] L. Vanbrabant, “Simulation and optimisation of emergency department operations,” *4OR*, vol. 19, pp. 469–470, Sep. 2021.

[18] Y.-H. Kuo, O. Rado, B. Lupia, J. M. Y. Leung, and C. A. Graham, “Improving the efficiency of a hospital emergency department: A simulation study with indirectly imputed service-time distributions,” *Flexible Services Manuf. J.*, vol. 28, no. 1, pp. 120–147, 2016.

[19] K. Ghanes *et al.*, “A comprehensive simulation modeling of an emergency department: A case study for simulation optimization of staffing levels,” in *Proc. Winter Simulation Conf.*, Dec. 2014, pp. 1421–1432.

[20] S. Zeltyn *et al.*, “Simulation-based models of emergency departments: Operational, tactical, and strategic staffing,” in *Proc. ACM Trans. Modeling Comput. Simulation (TOMACS)*, 2011, vol. 21, no. 4, p. 24.

[21] G. Xiao, M. Dong, J. Li, and L. Sun, “Scheduling routine and call-in clinical appointments with revisits,” *Int. J. Prod. Res.*, vol. 55, no. 6, pp. 1767–1779, Mar. 2017.

[22] F. Zaerpour, M. Bijvank, H. Ouyang, and Z. Sun, “Scheduling of physicians with time-varying productivity levels in emergency departments,” *Prod. Oper. Manage.*, vol. 31, no. 2, pp. 645–667, Feb. 2022, doi: 10.1111/poms.13571.

[23] W. Whitt, “Fluid models for multiserver queues with abandonments,” *Oper. Res.*, vol. 54, no. 1, pp. 37–54, Feb. 2006.

[24] G. B. Yom-Tov and A. Mandelbaum, “Erlang-R: A time-varying queue with reentrant customers, in support of healthcare staffing,” *Manuf. Service Oper. Manage.*, vol. 16, no. 2, pp. 283–299, May 2014.

[25] C. W. Chan, G. Yom-Tov, and G. Escobar, “When to use speedup: An examination of service systems with returns,” *Oper. Res.*, vol. 62, no. 2, pp. 462–482, Apr. 2014.

[26] A. Ingolfsson, E. Almehdawe, A. Pedram, and M. Tran, “Comparison of fluid approximations for service systems with state-dependent service rates and return probabilities,” *Eur. J. Oper. Res.*, vol. 283, no. 2, pp. 562–575, Jun. 2020.

[27] J. Huang, B. Carmeli, and A. Mandelbaum, “Control of patient flow in emergency departments, or multiclass queues with deadlines and feedback,” *Oper. Res.*, vol. 63, no. 4, pp. 892–908, Aug. 2015.

[28] W. Whitt and X. Zhang, “A data-driven model of an emergency department,” *Oper. Res. Health Care*, vol. 12, pp. 1–15, Mar. 2017.

[29] A. Stefanini, D. Aloini, E. Benevento, R. Dulmin, and V. Mininno, “Performance analysis in emergency departments: A data-driven approach,” *Measuring Bus. Excellence*, vol. 22, no. 2, pp. 130–145, Jun. 2018.

[30] L. V. Green, J. Soares, J. F. Giglio, and R. A. Green, “Using queueing theory to increase the effectiveness of emergency department provider staffing,” *Academic Emergency Med.*, vol. 13, no. 1, pp. 61–68, Jan. 2006.

[31] G. Chen, K. Govindan, Z.-Z. Yang, T.-M. Choi, and L. Jiang, “Terminal appointment system design by non-stationary $M(t)/E_k/c(t)$ queueing model and genetic algorithm,” *Int. J. Prod. Econ.*, vol. 146, no. 2, pp. 694–703, Dec. 2013.

[32] F. S. Hillier, *Introduction to Operations Research*. New York, NY, USA: McGraw-Hill, 2012.

[33] S. M. Ross, *Introduction to Probability Models*, 10 ed. Oxford, U.K.: Academic, 2014.

[34] L. Kletzander and N. Musliu, “Solving the general employee scheduling problem,” *Comput. Oper. Res.*, vol. 113, Jan. 2020, Art. no. 104794.

[35] S. Lan, W. Fan, S. Yang, N. Mladenović, and P. M. Pardalos, “Solving a multiple-qualifications physician scheduling problem with multiple types of tasks by dynamic programming and variable neighborhood search,” *J. Oper. Res. Soc.*, pp. 1–16, Aug. 2021, doi: 10.1080/01605682.2021.1954485.

[36] N. Niroumandrad and N. Lahrichi, “A stochastic Tabu search algorithm to align physician schedule with patient flow,” *Health Care Manage. Sci.*, vol. 21, no. 2, pp. 244–258, Jun. 2018.

[37] R. Liu and X. Xie, “Weekly scheduling of emergency department physicians to cope with time-varying demand,” *IISE Trans.*, vol. 53, no. 10, pp. 1109–1123, Oct. 2021.

Zixiang Wang received the B.S. degree in industrial engineering from Tongji University, Shanghai, China, in 2015, and the M.S. degree in industrial engineering from Technische Universität Braunschweig, Brunswick, Germany, in 2018. He is currently pursuing the Ph.D. degree with the Department of Industrial Engineering and Management, Shanghai Jiao Tong University. His research interests focus on exact and heuristic algorithm design, stochastic programming, and the optimization of service systems.

Ran Liu received the Ph.D. degree in industrial engineering from Shanghai Jiao Tong University, Shanghai, China, in 2011. He is an Associate Professor with the Department of Industrial Engineering and Management, Shanghai Jiao Tong University. He has ten years of research experience in areas of healthcare management and applied operations research. His research interests focus on combinational optimization, and algorithm design and analysis.

Zhankun Sun received the bachelor’s degree in industrial engineering from Tsinghua University and the Ph.D. degree in statistics and operations research from The University of North Carolina at Chapel Hill. He is currently an Assistant Professor with the City University of Hong Kong. His research interests include stochastic modeling and optimal control, especially with their applications arising from healthcare operations.