

## Research Article

# Outpatient Appointment Scheduling with Variable Interappointment Times

**Song Foh Chew**

*Department of Mathematics and Statistics, Southern Illinois University Edwardsville, Edwardsville, IL 62026, USA*

Correspondence should be addressed to Song Foh Chew, [schew@siue.edu](mailto:schew@siue.edu)

Received 30 March 2011; Revised 9 July 2011; Accepted 14 July 2011

Academic Editor: Farouk Yalaoui

Copyright © 2011 Song Foh Chew. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Healthcare currently consumes 17% of the U.S. Gross Domestic Product and is expected to reach 20% within the coming decade. Confronted with such high costs, sharp demand, and limited capacity, many hospitals now are vying for shorter lengths of stay and are transferring services from inpatient to outpatient facilities. This paper seeks to develop a methodology for constructing effective outpatient appointment scheduling systems. The objective of these appointment systems is to minimize the average total cost function describing total costs incurred by patient waiting and by staff idle time and overtime. In the paper, we will establish that the average total cost function exhibits a unimodal curve. The lowest point of the curve essentially means the lowest average total cost. We will next develop a simulation-based heuristic algorithm for finding an outpatient schedule near the lowest point. In the paper, we present numerical examples using the heuristic based upon a set of predetermined unit costs. Specifically, we find the near optimal interappointment times for schedules, where there are two and three patients in each block, respectively. The current work does not consider possible no shows and walk-ins. Future work will undertake these issues.

## 1. Introduction

Healthcare currently consumes 17% of the U.S. Gross Domestic Product and is expected to reach 20% within the coming decade [1]. It is not uncommon to find numerous patients waiting long periods of time to be serviced at a clinic or hospital. As the demand of outpatient facilities increases, more attention is also being directed to waiting times of patients, and doctors and medical personnel.

To address long waiting times for both parties, many clinics are turning to open access scheduling. This form of scheduling allows patients to make their appointments a day or two before the actual appointment [2]. Not having to have patients book their appointments weeks in advance gives all patients the opportunity to see the doctor as soon as necessary.

Using open access scheduling allows clinics the ability to minimize the cost of both the patient and the clinic. Unfortunately, waiting times for all involved are still apparent [3]. This paper seeks to develop effective outpatient appointment scheduling systems for outpatient clinics.

The remainder of the paper has been organized as follows. Section 2 defines and formulates the problem that we endeavor to solve. Section 3 provides a brief review of research conducted in outpatient appointment scheduling. We develop a heuristic algorithm for our problem in Section 4, while we justify the validity of the algorithm in Section 5. Numerical examples using the algorithm are included in Section 6. Finally, Section 7 offers concluding remarks.

## 2. Statement of Problem

Outpatient clinical scheduling is essentially a problem of assigning appointment times to patients who seek none-emergency medical attention. This scheduling occurs when a patient calls a clinic. Typically, a receptionist will answer the call and schedule the appointment for the patient based on their specifications.

Figure 1 illustrates a generic outpatient clinical schedule. In looking at the figure, we can gather that a day in a clinic

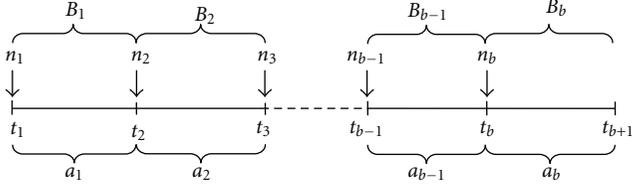


FIGURE 1: An outpatient clinical schedule with  $b$  blocks.

is divided into  $b$  blocks of equal time interval. A certain number  $n_p$  of patients will be scheduled to a block  $B_p$ . Each patient in the block is expected to arrive on time at the given appointment time  $t_p$ . In an ideal setting, the doctor would see every patient promptly and punctually during the interappointment time  $a_p$  for the block. We note that each patient has a random service time.

Along with the placement of patients throughout the day, we must consider the length of each patient's service time. There are 2 scenarios we must consider: (1) the total service time in a block ends before the next appointment time, the start time of the next block, or (2) the total service time takes longer than expected and overlaps with the next block. Scenario (1) gives rise to the doctor idle time at which they wait for the next patients to arrive. Scenario (2) produces overtime for the doctor while pushing patients in blocks  $B_p$  into  $B_{p+1}$ . Both scenarios are conflicting and provide much to be considered when scheduling appointments. The best way to combat this dilemma is to assign an adequate cost to each factor and to minimize the expected total cost. We define the expected total cost as

$$E(C) = c_w E(W) + c_d E(D) + c_v E(V), \quad (1)$$

where  $E(W)$ ,  $E(D)$ , and  $E(V)$  are expected patient waiting time, doctor, and staff idle time and overtime, respectively and  $c_w$ ,  $c_d$ , and  $c_v$  are the respective unit costs.

We define  $E(W)$ ,  $E(D)$ , and  $E(V)$  following the definitions presented in [4]. Let  $i$  denote a patient, and assume that each patient arrives promptly at their appointment time,  $t_i$ . Now, let  $b_i$ ,  $s_i$ , and  $e_i$ , respectively, be the time at which service starts, the length of service time, and the time at which service ends for patient  $i$ . Let us assume the first patient's appointment time and service start time to be 0; that is,  $t_1 = b_1 = 0$ . For each patient  $i$  after the first, we can say that their service start time is the maximum of their appointment time and the service end time of the previous patient. Therefore, we have  $b_i = \max(t_i, e_{i-1})$  and  $e_i = b_i + s_i$ . We can find the waiting time of patient  $i$ ,  $w_i$ , by considering their service start time  $b_i$  minus their appointment time  $t_i$ ; hence,  $w_i = b_i - t_i$ , where  $w_1 = 0$ .

The doctor and staff are considered idle while waiting for the next patient to arrive after serving a patient. Thus, the doctor and staff idle time before serving patient  $i$  is  $d_i = \max(0, t_i - e_{i-1})$ , where  $d_1 = 0$ .

When there is a fixed number of patients scheduled for a day, say  $N$ , the total patient waiting time and the total doctor and staff idle time are  $W = \sum_{i=1}^N w_i$ , and  $D = \sum_{i=1}^N d_i$ , respectively. We now define the doctor and staff overtime; it

is the time the medical staff has to serve after the end of a day; so, the overtime is  $V = \max(0, e_N - t_{b+1})$ .

Note that  $e_N$  is the end time of the last patient, and that  $t_{b+1}$  is the end time of a day. In this paper, service times are assumed random; so  $W$ ,  $D$ , and  $V$  are also random. We use  $E(W)$ ,  $E(D)$ , and  $E(V)$  to denote the expectation of the total patient waiting time, the total doctor and staff idle time, and the total doctor and staff overtime, respectively. We will use the averages of  $h$  observations of  $W$ ,  $D$ , and  $V$  to approximate the respective expectations. Thus, the expected total cost of (1) is estimated by the following average total cost:

$$\bar{C} = c_w \bar{W} + c_d \bar{D} + c_v \bar{V}, \quad (2)$$

where  $\bar{W} = (1/h) \sum_{k=1}^h W_k$ ,  $\bar{D} = (1/h) \sum_{k=1}^h D_k$ ,  $\bar{V} = (1/h) \sum_{k=1}^h V_k$ . Later in the paper, we will set  $h = 1000$  for our numerical examples.

Suppose that the clinic has good estimates of service times and of the number  $N$  of patients per day from historical data, that the number  $b$  of blocks for a clinical schedule is preset, and that  $n_p = N/b$ , assuming an integer, for every  $p$ . Then, this paper specifically seeks to solve the following problem:

Find interappointment times  $a_p$ 's so as to

$$\text{minimize } E(C) \quad (3)$$

subject to  $a_p > 0$  for every  $p$ .

That is, given that the number of patients and the number of blocks are known and that patients are evenly distributed into each block, the interappointment times are to be determined in order to minimize the expected total cost. We will develop a simulation-based heuristic algorithm for finding an outpatient schedule leading to a near optimal solution; we present numerical examples using the heuristic.

### 3. Literature Review

There has been a great deal of research conducted on outpatient appointment scheduling methodologies over the past decades. In the seminal work, Bailey [5] develops an outpatient appointment scheduling system, called an individual appointment system. In this system,  $n_p = 1$  for every  $p$ ; that is, there is exactly 1 patient for each of the  $b$  blocks, and hence, there are  $N = b$  patients in a day. Further, Bailey conveniently chooses all interappointment times to be  $\mu$ , the average time the physician spends with a patient; thus,  $a_p = \mu$  for every  $p$  in their system. The choice of  $\mu$  seems reasonable with regards to minimizing  $E(C)$  of (1). A natural extension of this work would be that not every  $n_p$  is equal to 1 [6]. The resulting systems are called block appointment systems.

There are 2 types of block appointment systems. One is equal-block appointment systems, where every  $n_p$  is the same; the other is variable-block appointment systems, where not every  $n_p$  is equal. These systems are studied in [7–12]. Specifically, White and Pike [7] and Soriano [8] look into equal-block systems. While White and Pike address

```

Input:  $N, b, n_p (p = 1 \dots b), c_w, c_d, c_v$ , service time distributions, and  $h$ .
Output:  $a_p (p = 1 \dots b)$ .
Initialization:  $a_p = 1$  for  $p = 1 \dots b$ ;
    Evaluate  $\bar{C}$ ;
     $OldMinExpectedTotalCost = \bar{C}$ .
Begin
Step 1 For  $p = 1 \dots b$ 
     $a_p = a_p + 1$ ;
    Evaluate  $\bar{C}$ ;
     $ExpectedTotalCost(p) = \bar{C}$ ;
     $a_p = a_p - 1$ ;
End For
Step 2 Find  $p$  such that  $ExpectedTotalCost(p)$  is the minimum;
     $a_p = a_p + 1$ ;
     $NewMinExpectedTotalCost = ExpectedTotalCost(p)$ .
Step 3 If  $NewMinExpectedTotalCost < OldMinExpectedTotalCost$ 
     $OldMinExpectedTotalCost = NewMinExpectedTotalCost$ ;
    Go to Step 1;
else
     $a_p = a_p - 1$ ;
    Return  $a_p (p = 1 \dots b)$ ;
End If
End
    
```

ALGORITHM 1: Simulation-based heuristic algorithm.

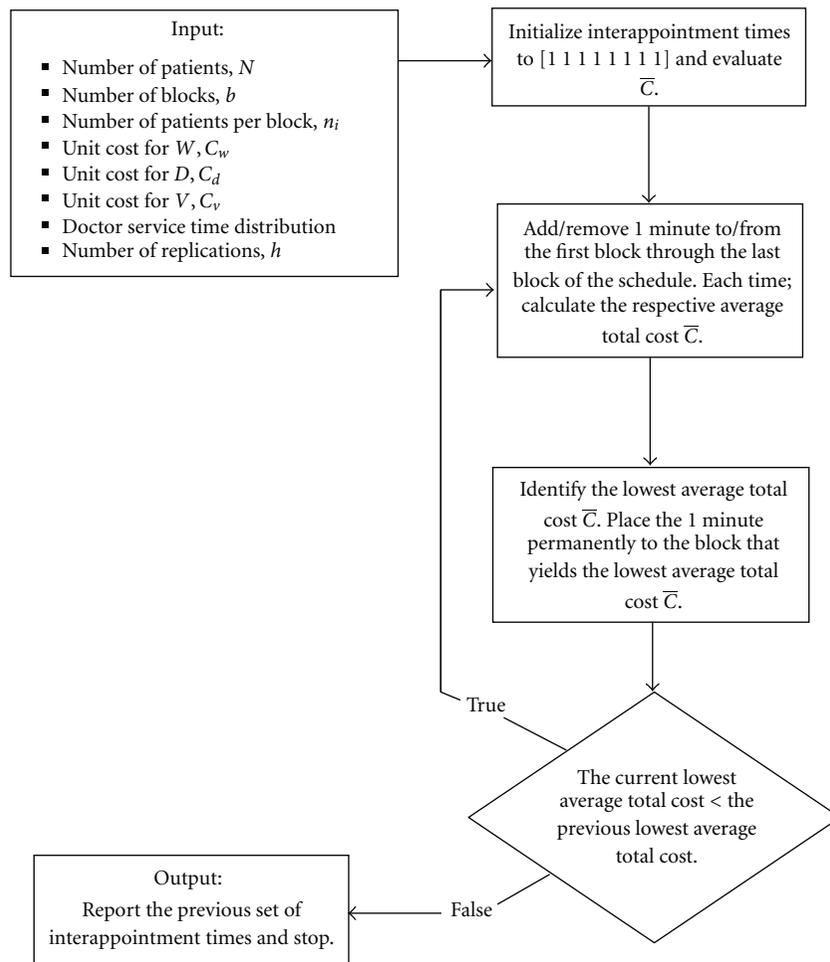


FIGURE 2: Flowchart of simulation-based heuristic algorithm.

setting  $n_p = d > 1$  and  $a_p = d\mu$  for every  $p$ , Soriano particularly investigates into assuming  $n_p = 2$  and  $a_p = 2\mu$  for every  $i$ . In both,  $b = N/d$  for a given  $N$  and a chosen  $d$ . On the other hand, Fries and Marathe [9], Vanden et al. [10], Liu et al. [11], and Muthuraman and Lawley [12] develop variable-block systems. To this end, Fries and Marath, Vanden et al., and Liu et al. apply a notion of dynamic programming to determine  $n_p$ . Muthuraman and Lawley construct an overbooking model to find  $n_p$ . For their approaches,  $b$  and  $a_p$  must be prespecified for a given  $N$ . The performances of these systems, equal- and variable-block, are varying according to  $N$  and random nature of the times spent between the physician and patients. These researchers demonstrate that the performances almost always outshine that of individual appointment systems. Note that a system of these types has a constant interappointment time; that is, every  $a_p$  is equal. As it turns out, varying  $a_p$  brings about powerful systems called variable-interappointment-time appointment systems.

Variable-interappointment-time appointment systems often outperform other systems. Ho and Lau [4] and Ho et al. [13] discovered that patient waiting tends to tremendously prolong towards the end of the day, because more and more patients overflow from earlier blocks into later blocks. In fact, this confirms a common perception that patients with late appointments in the day tend to wait much longer than do patients with early appointments. As such, Ho and Lau, and Ho et al. develop systems in which  $a_p$  is to be augmented towards the end of the schedule so as to reduce patient overflow. Supposing that  $n_p = N/b$  for every  $p$  with given  $N$  and  $b$ , they essentially let  $a_p = \mu + ik\sigma$ , where  $k$  is a constant and  $\sigma$  is the variability of the times spent between the physician and patients. Such choices as  $k = 0.15, 0.25, 0.3$ , and  $0.5$  are experimented and these systems are found to significantly outperform all the aforementioned systems. Note that  $a_p$  for these systems is arbitrarily or at best conveniently determined based upon the choices of  $k$ . As Cayirli and Veral [14] and Gupta and Denton [15] point out, there has virtually been no rigorous research into approaches and methodologies for finding interappointment times. Ultimately motivated by the above understanding, this paper seeks to rigorously and precisely determine  $a_p$  for every  $p$  so as to further unleash the power of variable-interappointment-time appointment systems. The next section will develop an algorithm for constructing variable-interappointment-time appointment systems.

#### 4. Simulation-Based Heuristic Algorithm

This section describes the way we heuristically determine interappointment times for a clinical schedule so as to minimize the expected total cost. Suppose that there are a total of  $N$  patients in a given schedule and that there are  $b$  blocks into which we want to distribute these  $N$  patients. We note that these  $N$  patients are evenly distributed throughout the  $b$  blocks; that is,  $n_1 = n_2 = \dots = n_b$  such that  $N = \sum_{i=1}^b n_i$ . We want to find the interappointment times  $a_p$  ( $p = 1, 2, \dots, b$ ) for the schedule in order to minimize (1).

To start, there is 1 minute given to each block. This is done because it is not reasonable to have a block with zero minutes, since there are  $n_p$  ( $n_p > 0$ ) patients in each block. In this paper, we assume that there are 8 blocks ( $b = 8$ ) in a schedule. Therefore, our initial interappointment times are [1 1 1 1 1 1 1 1].

We now add 1 minute to the first block of the initial interappointment times [1 1 1 1 1 1 1 1], yielding [2 1 1 1 1 1 1 1]. We then generate a set of random service times,  $S_1 = \{S_{11}, S_{12}, \dots, S_{1N}\}$ . This paper assumes that service times are exponentially distributed with a mean of 10 minutes. With  $S_1$ , we compute  $W_1$ ,  $D_1$ , and  $V_1$  for the schedule with interappointment times [2 1 1 1 1 1 1 1]. After that, we generate another set of random service times,  $S_2 = \{S_{21}, S_{22}, \dots, S_{2N}\}$ , where  $S_2 \neq S_1$ . With  $S_2$ ,  $W_2$ ,  $D_2$ , and  $V_2$  are computed for the same schedule. In fact, this process is repeated 1000 times; the number 1000 is conveniently chosen for simulation purposes. We now have  $\{W_1, W_2, \dots, W_{1000}\}$ ,  $\{D_1, D_2, \dots, D_{1000}\}$ , and  $\{V_1, V_2, \dots, V_{1000}\}$  corresponding to  $\{S_1, S_2, \dots, S_{1000}\}$ , respectively, where all service times are random. We then obtain  $\bar{W} = (1/(1000))\sum_{i=1}^{1000} w_i$ ,  $\bar{D} = (1/(1000))\sum_{i=1}^{1000} D_i$  and  $\bar{V} = (1/(1000))\sum_{i=1}^{1000} V_i$  and use  $\bar{C} = c_w\bar{W} + c_d\bar{D} + c_v\bar{V}$  to estimate the expected total cost  $E(C)$  for the schedule with interappointment times [2 1 1 1 1 1 1 1]. Note that the number  $h$  of observations for the above estimation is set to 1000.

This 1 minute is next removed from the first block and added to the second block. We then have interappointment times [1 2 1 1 1 1 1 1] for our schedule. Repeating the above computation process, the average total cost  $\bar{C}$  is calculated for the schedule with interappointment times [1 2 1 1 1 1 1 1]. This continues on as shown below.

$$\begin{aligned}
 & [2 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \\
 & [1 \ 2 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \\
 & [1 \ 1 \ 2 \ 1 \ 1 \ 1 \ 1 \ 1] \\
 & [1 \ 1 \ 1 \ 2 \ 1 \ 1 \ 1 \ 1] \\
 & [1 \ 1 \ 1 \ 1 \ 2 \ 1 \ 1 \ 1] \\
 & [1 \ 1 \ 1 \ 1 \ 1 \ 2 \ 1 \ 1] \\
 & [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 2 \ 1] \\
 & [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 2]
 \end{aligned} \tag{4}$$

After going through the above procedure, we will obtain 8 average total costs  $\bar{C}$ . The smallest average total cost  $\bar{C}$  is then identified. Whichever row from the above has this lowest value will have the 1 minute placed permanently into that block. For example, suppose that the third row gives the lowest average total cost  $\bar{C}$ , we then will place the 1 minute permanently in the third block. As a result, we have

updated our interappointment times from [1 1 1 1 1 1 1] to [1 1 2 1 1 1 1]. We will now make [1 1 2 1 1 1 1] our new initial interappointment times. We will repeat the above procedure with [1 1 2 1 1 1 1] being the initial interappointment times for our schedule by adding/removing 1 minute to/from the first block through the last block. After this procedure, 8 average total cost  $\bar{C}$  will be obtained. The lowest value of  $\bar{C}$  is identified and then the 1 minute is placed permanently in the block corresponding to the lowest  $\bar{C}$ . For example, we may this time obtain interappointment times, say, [1 1 2 1 2 1 1]. We now make [1 1 2 1 2 1 1] our new initial interappointment times.

The above procedure for adding/removing 1 minute to/from a block is repeated multiple times to increase the interappointment times for the schedule progressively. As we may expect, the average total cost  $\bar{C}$  for the schedule would be high when the interappointment times are short. This is true in that the average waiting time  $\bar{W}$  and the average staff overtime  $\bar{V}$  would be large, but the average doctor and staff idle time  $\bar{D}$  would tend to be zero, when the interappointment times are short. Thus, the average total cost  $\bar{C}$  would be decreased as we progressively place 1 minute permanently in a block to effectively increase the respective interappointment time. However, the average total cost would increase when the interappointment times are becoming too large. This is reasonable in that when the interappointment times become too large, the average doctor and staff idle time would become large, but the average patient waiting time would become a constant, and the average doctor and staff overtime would become zero. We would stop the procedure for adding/removing 1 minute to/from a block as soon as the average total cost starts to increase. We will then return the previous set of interappointment times to be an approximation to the optimal interappointment times for the schedule. We will in Section 5 demonstrate that the average total cost is a unimodal function. As a consequence, a local minimum is in fact a global minimum. We now summarize the above-discussed procedure as Algorithm 1.

Figure 2 presents a flowchart illustrating the logical flow of the above algorithm; the algorithm has been coded using MATLAB. We will justify the validity of the algorithm in the following section.

## 5. Justification of Simulation-Based Heuristic Algorithm

This section justifies the validity of the simulation-based heuristic algorithm. Suppose that we have generated a set of service times,  $S = \{S_1^1, S_2^1, \dots, S_{n_1}^1; S_1^2, S_2^2, \dots, S_{n_2}^2; \dots; S_1^b, S_2^b, \dots, S_{n_b}^b\}$ . This means that we have a total of  $N$  ( $N = n_1 + n_2 + \dots + n_b$ ) patients for a day divided into  $b$  blocks. There are  $n_i$  patients in block  $B_i$  and service time  $S_j^i$ , assuming integer-valued in minutes, is the time spent between the doctor and patient  $j$  in block  $B_i$ , with  $i = 1, 2, \dots, b$ .

**Theorem 1.** *If interappointment time  $a_i \geq \sum_{j=1}^{n_i} S_j^i$  for all  $i = 1, 2, \dots, b$ , then the total patient waiting time is  $TW =$*

*$\sum_{i=1}^b \sum_{k=1}^{n_i} \sum_{j=1}^{k-1} S_j^i$ . In addition,  $TW$  is the smallest possible total patient waiting time.*

*Proof.* We have  $a_i \geq \sum_{j=1}^{n_i} S_j^i$  with  $i = 1, 2, \dots, b$ . This means that each interappointment time is longer than or equal to the sum of the service times of all the patients in each block. This in turn says that the first patient in each block does not have to wait. However, the subsequent patients will have to wait until every patient before them has been seen by the doctor. Thus, we have what is seen in the following.

| Patient                         | Patient waiting time                  |
|---------------------------------|---------------------------------------|
| 1st patient in block $B_1$      | 0                                     |
| 2nd patient in block $B_1$      | $S_1^1$                               |
| 3rd patient in block $B_1$      | $S_1^1 + S_2^1$                       |
| $\vdots$                        | $\vdots$                              |
| $n_1$ th patient in block $B_1$ | $S_1^1 + S_2^1 + \dots + S_{n_1-1}^1$ |

| Patient                         | Patient waiting time                  |
|---------------------------------|---------------------------------------|
| 1st patient in block $B_2$      | 0                                     |
| 2nd patient in block $B_2$      | $S_1^2$                               |
| 3rd patient in block $B_2$      | $S_1^2 + S_2^2$                       |
| $\vdots$                        | $\vdots$                              |
| $n_2$ th patient in block $B_2$ | $S_1^2 + S_2^2 + \dots + S_{n_2-1}^2$ |

| Patient                         | Patient waiting time                  |
|---------------------------------|---------------------------------------|
| 1st patient in block $B_b$      | 0                                     |
| 2nd patient in block $B_b$      | $S_1^b$                               |
| 3rd patient in block $B_b$      | $S_1^b + S_2^b$                       |
| $\vdots$                        | $\vdots$                              |
| $n_b$ th patient in block $B_b$ | $S_1^b + S_2^b + \dots + S_{n_b-1}^b$ |

It can now easily be seen that the total patient waiting time is  $TW = \sum_{i=1}^b \sum_{k=1}^{n_i} \sum_{j=1}^{k-1} S_j^i$ . It is also clear, from the above discussion, that  $TW$  is the smallest possible total patient waiting time.  $\square$

The above theorem asserts that, given a realization of service times, the total wait time for all patients in a day is at least  $TW$ . We will use the least total patient wait time to establish that the average total cost is unimodal in the following theorem.

**Theorem 2.** *The average total cost  $\bar{C}$  exhibits a unimodal curve as we carry out the simulation-based heuristic algorithm as discussed in Section 4.*

*Proof.* Assume that the lowest average total cost  $\bar{C}'$  is obtained after a round of adding/removing 1 minute to/from the first block through the last block of a schedule. Now, perform another round of adding/removing 1 minute to/from the first block through the last block; suppose that we obtain the lowest average total cost  $\bar{C}''$  this time. Let us assume that  $\bar{C}'' > \bar{C}'$ . This means that we have just crossed the lowest point of the curve. We now need to show that any lowest average total cost we obtain by further adding/removing 1 minute to/from the first block through the last block will only increase.

TABLE 1: Results for [2 2 2 2 2 2 2 2].

| Unit Costs<br>( $c_w, c_d, c_v$ ) | Number of minutes per block |       |       |       |       |       |       |       | Minimum<br>average total<br>cost |
|-----------------------------------|-----------------------------|-------|-------|-------|-------|-------|-------|-------|----------------------------------|
|                                   | $B_1$                       | $B_2$ | $B_3$ | $B_4$ | $B_5$ | $B_6$ | $B_7$ | $B_8$ |                                  |
| (1, 1, 1,)                        | 29                          | 34    | 33    | 33    | 33    | 33    | 31    | 21    | 232.37                           |
| (1, 1, 50)                        | 28                          | 35    | 32    | 34    | 34    | 34    | 36    | 73    | 281.66                           |
| (1, 1, 100)                       | 28                          | 35    | 32    | 34    | 34    | 35    | 37    | 80    | 287.92                           |
| (1, 50, 1)                        | 4                           | 14    | 14    | 19    | 17    | 17    | 17    | 12    | 725.83                           |
| (1, 50, 50)                       | 5                           | 16    | 15    | 20    | 21    | 19    | 28    | 42    | 2050.92                          |
| (1, 50, 100)                      | 6                           | 18    | 16    | 20    | 21    | 20    | 32    | 51    | 2676.11                          |
| (1, 100, 1)                       | 3                           | 11    | 13    | 16    | 15    | 18    | 16    | 13    | 795.55                           |
| (1, 100, 50)                      | 4                           | 13    | 14    | 20    | 16    | 18    | 26    | 39    | 2518.50                          |
| (1, 100, 100)                     | 4                           | 14    | 13    | 20    | 19    | 19    | 29    | 46    | 3599.97                          |
| (50, 1, 1)                        | 72                          | 69    | 68    | 69    | 63    | 66    | 66    | 18    | 4601.77                          |
| (50, 1, 50)                       | 72                          | 69    | 68    | 69    | 63    | 66    | 66    | 64    | 4646.25                          |
| (50, 1, 100)                      | 72                          | 69    | 68    | 69    | 63    | 66    | 66    | 73    | 4653.42                          |
| (50, 50, 1)                       | 29                          | 37    | 33    | 33    | 33    | 32    | 29    | 3     | 10993.76                         |
| (50, 50, 50)                      | 29                          | 34    | 33    | 33    | 33    | 33    | 31    | 21    | 11618.70                         |
| (50, 50, 100)                     | 29                          | 34    | 33    | 32    | 34    | 33    | 32    | 28    | 11956.80                         |
| (50, 100, 1)                      | 23                          | 30    | 28    | 29    | 29    | 27    | 24    | 3     | 14216.78                         |
| (50, 100, 50)                     | 23                          | 30    | 29    | 29    | 29    | 28    | 26    | 17    | 15119.40                         |
| (50, 100, 100)                    | 23                          | 31    | 28    | 29    | 30    | 28    | 27    | 23    | 15695.75                         |
| (100, 1, 1)                       | 78                          | 76    | 75    | 75    | 67    | 68    | 80    | 18    | 8848.68                          |
| (100, 1, 50)                      | 78                          | 75    | 74    | 75    | 67    | 68    | 79    | 64    | 8893.30                          |
| (100, 1, 100)                     | 78                          | 75    | 74    | 75    | 67    | 68    | 79    | 73    | 8900.47                          |
| (100, 50, 1)                      | 37                          | 39    | 38    | 37    | 39    | 37    | 34    | 3     | 17202.50                         |
| (100, 50, 50)                     | 37                          | 39    | 38    | 38    | 39    | 37    | 35    | 19    | 17770.05                         |
| (100, 50, 100)                    | 37                          | 39    | 38    | 38    | 39    | 37    | 36    | 26    | 18076.10                         |
| (100, 100, 1)                     | 29                          | 34    | 33    | 33    | 33    | 32    | 29    | 3     | 21964.71                         |
| (100, 100, 50)                    | 39                          | 34    | 32    | 33    | 34    | 32    | 30    | 15    | 22743.15                         |
| (100, 100,<br>100)                | 29                          | 34    | 33    | 33    | 33    | 33    | 31    | 21    | 23237.40                         |

Number of patients per block, [2 2 2 2 2 2 2 2]

Note that since  $\bar{C}'' > \bar{C}'$ , each average total cost we obtain from the second round of adding/removing 1 minute to/from each block is greater than  $\bar{C}'$ . This means that adding 1 minute to a block will only increase the average total cost. This further means that the total patient waiting time has already reached the minimum value  $TW$  given in Theorem 1, that the doctor and staff overtime has reached zero, and that the doctor and staff idle time has begun to increase from zero. Thus, from now on, we have  $\bar{C} = c_w \bar{W} + c_d \bar{D} + c_v 0 = c_w \bar{W} + c_d \bar{D}$ , which will only increase, since adding 1 minute to any block will only increase  $\bar{D}$ ; however,  $\bar{W}$  ( $\bar{W} = TW$ ) stays constant.  $\square$

## 6. Numerical Examples and Discussion

This section investigates into the way the 3 unit costs affect the interappointment times in a schedule. We assume that the number  $b$  of blocks in a schedule is 8. Each unit

cost may take on 1 of 3 possible values, 1, 50, and 100; there are a total of 27 combinations of these values. We consider the schedule associated with all the unit costs being 1 as the base-case schedule. We look also at the effects in interappointment times from 2 different numbers  $N$  of patients in a day; that is,  $N = 16$  with [2 2 2 2 2 2 2 2] and  $N = 24$  with [3 3 3 3 3 3 3 3]. We note that for all the numerical examples, we employ exponential service times with a mean of 10 minutes and the number  $h$  of observations is 1000. This configuration is chosen for convenience; one may pick other configurations of choice. We present the results in 2 tables. Specifically, Table 1 contains the results of various unit costs, where there are 2 patients in each block; Table 2, that of various unit costs where each block holds 3 patients.

To begin, we first look at the unit costs associated with the doctor and staff overtime, leaving the other 2 costs intact. If there is a high cost to have overtime at the end of the day, then we would want to minimize the overtime there is. Overtime is the time left at the end of the day after the last patient is

TABLE 2: Results for [3 3 3 3 3 3 3 3].

| Unit Costs<br>( $c_w, c_d, c_v$ ) | Number of minutes per block |       |       |       |       |       |       |       | Minimum<br>average total<br>cost |
|-----------------------------------|-----------------------------|-------|-------|-------|-------|-------|-------|-------|----------------------------------|
|                                   | $B_1$                       | $B_2$ | $B_3$ | $B_4$ | $B_5$ | $B_6$ | $B_7$ | $B_8$ |                                  |
| (1, 1, 1,)                        | 44                          | 51    | 50    | 50    | 49    | 48    | 48    | 32    | 445.45                           |
| (1, 1, 50)                        | 44                          | 52    | 50    | 51    | 53    | 48    | 51    | 81    | 506.65                           |
| (1, 1, 100)                       | 41                          | 51    | 49    | 51    | 52    | 48    | 53    | 92    | 519.79                           |
| (1, 50, 1)                        | 12                          | 25    | 28    | 26    | 28    | 25    | 26    | 15    | 1327.52                          |
| (1, 50, 50)                       | 13                          | 26    | 31    | 29    | 31    | 30    | 39    | 54    | 2918.67                          |
| (1, 50, 100)                      | 13                          | 27    | 33    | 30    | 32    | 34    | 43    | 61    | 3649.89                          |
| (1, 100, 1)                       | 10                          | 22    | 24    | 26    | 27    | 24    | 25    | 15    | 1465.32                          |
| (1, 100, 50)                      | 10                          | 24    | 27    | 27    | 30    | 28    | 38    | 48    | 3684.46                          |
| (1, 100, 100)                     | 11                          | 24    | 29    | 28    | 31    | 29    | 41    | 57    | 4923.40                          |
| (50, 1, 1)                        | 89                          | 91    | 97    | 91    | 91    | 86    | 94    | 29    | 12967.97                         |
| (50, 1, 50)                       | 89                          | 91    | 97    | 91    | 91    | 86    | 94    | 74    | 13015.88                         |
| (50, 1, 100)                      | 89                          | 91    | 97    | 91    | 91    | 86    | 97    | 89    | 13027.42                         |
| (50, 50, 1)                       | 44                          | 51    | 50    | 50    | 49    | 48    | 46    | 7     | 22065.71                         |
| (50, 50, 50)                      | 44                          | 51    | 50    | 50    | 49    | 48    | 48    | 32    | 22772.70                         |
| (50, 50, 100)                     | 14                          | 51    | 50    | 50    | 49    | 48    | 49    | 39    | 23119.40                         |
| (50, 100, 1)                      | 38                          | 46    | 44    | 45    | 43    | 43    | 38    | 6     | 27129.99                         |
| (50, 100, 50)                     | 38                          | 46    | 44    | 46    | 42    | 43    | 41    | 27    | 28144.30                         |
| (50, 100, 100)                    | 38                          | 46    | 44    | 46    | 42    | 44    | 42    | 33    | 28749.40                         |
| (100, 1, 1)                       | 95                          | 94    | 106   | 95    | 95    | 90    | 104   | 29    | 25482.78                         |
| (100, 1, 50)                      | 95                          | 94    | 106   | 95    | 95    | 90    | 104   | 74    | 25530.22                         |
| (100, 1, 100)                     | 95                          | 94    | 106   | 95    | 95    | 90    | 104   | 89    | 25541.32                         |
| (100, 50, 1)                      | 53                          | 58    | 56    | 56    | 57    | 54    | 54    | 6     | 37042.86                         |
| (100, 50, 50)                     | 53                          | 58    | 56    | 56    | 57    | 54    | 55    | 31    | 37695.45                         |
| (100, 50, 100)                    | 53                          | 58    | 56    | 56    | 57    | 54    | 55    | 38    | 38014.45                         |
| (100, 100, 1)                     | 44                          | 51    | 50    | 50    | 49    | 48    | 46    | 6     | 44102.34                         |
| (100, 100, 50)                    | 44                          | 51    | 50    | 50    | 49    | 48    | 47    | 25    | 45011.40                         |
| (100, 100,<br>100)                | 44                          | 51    | 50    | 50    | 49    | 48    | 48    | 32    | 44545.40                         |

Number of patients per block, [3 3 3 3 3 3 3 3]

through seeing the doctor. Thus, if the unit cost of overtime is increased, the first 7 blocks would not be affected, but the last block would be significantly longer.

Again leaving the other two alone, we look at just the unit cost for the doctor and staff idle time. If the unit cost for idle time is adjusted, particularly increased, this means that there is more of an emphasis placed on the amount of idle time there is in a given block. If this unit cost is increased, then to keep the expected total cost at a minimum there needs to be as little to no idle time as possible. This would cause the blocks to be shorter.

Lastly is the unit cost of the patient waiting time. This comes from the fact that the second patient in a block is always going to have to wait on the first patient. What we are concerned with is the prospect of a patient having a long service time. If this happens and there is an overflow into the next block, then the first patient of the next block now has a waiting time. If overtime and idle time are not

issues, we might be able to preclude the above-mentioned scenario from occurring. In this case, all the blocks will be exceptionally long except for the last one, which will be average.

From Tables 1 and 2, we can see how the minimum expected total cost changes as the unit costs change. In retrospect, we said that looking at only the overtime but leaving the other 2 unit costs unchanged, the blocks would be similar to our base-case schedule, where  $(c_w, c_d, c_v) = (1, 1, 1)$ . This is evident in the first 3 rows of Table 1, we can see that the first 7 columns are very similar. Column 8, however, differs considerably across the 3 rows.

Moving now, we look at the unit cost of idle time. We predicted that if the unit cost of the idle time increased, the blocks would be shorter. To verify this, we are going to look at the rows, where  $(c_w, c_d, c_v) = (1, 50, 1)$  and  $(c_w, c_d, c_v) = (1, 100, 1)$ . For these 2 rows, all the blocks are substantially shorter than that of our base case.

Lastly we hypothesized that if the unit cost of patient waiting time was increased, then the interappointment times would be increased, all except for the last block. To see these changes, we need only look at the rows where  $(c_w, c_d, c_v) = (50, 1, 1)$  and  $(c_w, c_d, c_v) = (100, 1, 1)$ . Looking at these rows in Table 1, we see that

$$\begin{aligned} (1, 1, 1): & 29 \ 34 \ 33 \ 33 \ 33 \ 33 \ 31 \ 21 \ 232.374 \\ (50, 1, 1): & 72 \ 69 \ 68 \ 69 \ 63 \ 66 \ 66 \ 18 \ 4601.767 \\ (100, 1, 1): & 78 \ 76 \ 75 \ 75 \ 67 \ 68 \ 80 \ 18 \ 8848.679. \end{aligned} \quad (5)$$

It is easily seen that by increasing the unit cost of patient waiting time, the length of the blocks are significantly increased.

Looking also at all the rest of the rows, these show different choices of the 3 values of 1, 50, and 100. Combinations of these lead to a whole host of different interappointment times, as well as a collection of minimum expected total costs. We further observe that, for example, the row for which  $(c_w, c_d, c_v) = (1, 100, 100)$ . Both the unit costs of idle time and overtime have been increased, respectively. Since there is a high cost associated with the idle time, the blocks must be shorter; since there is a high unit cost for the overtime, the last block must be as long as necessary so as not to have any overtime. Looking at this row, this is what is transpiring to the blocks. There is 1 other observation which needs to be made. The unit costs,  $(c_w, c_d, c_v) = (1, 1, 1)$ ,  $(50, 50, 50)$ ,  $(100, 100, 100)$ , yield the same interappointment times. However, the respective expected total costs differ. The expected total cost is going to be effected in such a way that the expected total cost for  $(c_w, c_d, c_v) = (50, 50, 50)$  is greater than that of  $(c_w, c_d, c_v) = (1, 1, 1)$  by a factor of 50; the expected total cost for  $(c_w, c_d, c_v) = (100, 100, 100)$  is greater than that of  $(c_w, c_d, c_v) = (1, 1, 1)$  by a factor of 100. The above can be seen in Table 1.

In addition to finding various interappointment times for a day of 8 blocks such that  $N = 16$  with 2 patients in each block, we run the same 27 combinations of unit costs for a day such that  $N = 24$  with 3 patients in each block. The reason for this is to see the differences in expected total costs as well as to see changes in interappointment times which come with adding more patients into a day. Table 2 contains these expected total costs as well as the interappointment times associated with the 27 combinations. Similar observations can be made from Table 2 as are deduced from Table 1.

## 7. Conclusion

The objective of this paper was to find how long to make each block in a schedule in order to minimize the expected total cost. A simulation-based heuristic algorithm was constructed so as to determine approximately optimal interappointment times. We justified the validity of the heuristic. We provided a number of numerical examples based upon a set of predetermined unit costs using the heuristic. Specifically, we found the approximately optimal interappointment times for schedules with two patients per block and for schedules

with three patients per block, respectively. We compared and contrasted the results of these two types of schedules. One direction for future work is to look into determining optimal interappointment times under the consideration of patient no-shows and walk-ins, which can be easily incorporated into our simulation-based approach.

## Acknowledgment

This project was supported in part by the Summer Research Fellowship of the Southern Illinois University Edwardsville, Edwardsville, USA.

## References

- [1] National health care expenditures projections: 2009–2019. Centers for Medicare and Office of the Actuary Medicaid Services, USA.
- [2] C. D. O'Hare and J. Corlett, "The outcomes of open-access scheduling," *Family Practice Management*, vol. 11, no. 2, pp. 35–38, 2004.
- [3] P. Proctor, W. Compton, J. Grossman, and G. Fanjiang, Eds., *Building a Better Delivery System: A New Engineering/Health Care Partnership*, National Academies Press, 2005.
- [4] C. Ho and H. Lau, "Minimizing total cost in scheduling outpatient appointments," *Management Science*, vol. 38, no. 12, pp. 1750–1764, 1992.
- [5] N. Bailey, "A study of queues and appointment systems in hospital outpatient departments with special reference to waiting times," *Journal of the Royal Statistical Society*, vol. 14, pp. 185–199, 1952.
- [6] S. Sickinger and R. Kolisch, "The performance of a generalized Bailey-Welch rule for outpatient appointment scheduling under inpatient and emergency demand," *Health Care Management Science*, vol. 12, no. 4, pp. 408–419, 2009.
- [7] M. White and M. Pike, "Appointment systems in outpatient's clinics and the effect of patients' unpunctuality," *Medical Care*, vol. 2, pp. 133–145, 1964.
- [8] A. Soriano, "Comparison of two scheduling systems," *Operations Research*, vol. 14, pp. 388–397, 1966.
- [9] B. E. Fries and V. P. Marathe, "Determination of optimal variable-sized multiple-block appointment systems," *Operations Research*, vol. 29, no. 2, pp. 324–345, 1981.
- [10] B. Vanden, P. Dietz, and J. R. Simeoni, "Scheduling customer arrivals to a stochastic service system," *Naval Research Logistics*, vol. 46, no. 5, pp. 549–559, 1999.
- [11] N. Liu, S. Ziya, and V. G. Kulkarni, "Dynamic scheduling of outpatient appointments under patient no-shows and cancellations," *Manufacturing and Service Operations Management*, vol. 12, no. 2, pp. 347–364, 2010.
- [12] K. Muthuraman and M. Lawley, "A stochastic overbooking model for outpatient clinical scheduling with no-shows," *IIE Transactions*, vol. 40, no. 9, pp. 820–837, 2008.
- [13] C. J. Ho, H. S. Lau, and J. Li, "Introducing variable-interval appointment scheduling rules in service systems," *International Journal of Operations and Production Management*, vol. 15, no. 6, pp. 59–69, 1995.
- [14] T. Cayirli and E. Veral, "Outpatient scheduling in health care: a review of literature," *Production and Operations Management*, vol. 12, no. 4, pp. 519–549, 2003.

- [15] D. Gupta and B. Denton, "Appointment scheduling in health care: challenges and opportunities," *IIE Transactions*, vol. 40, no. 9, pp. 800–819, 2008.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

