Modeling the Optimal Number of Lines at an Airport for Various Arrival Rates of Travelers

Katherine Wilsdon
Idaho State University
Pocatello, Idaho
wilskat7@isu.edu

Dr. Isaac Griffith
Idaho State University
Pocatello, Idaho
grifisaa@isu.edu

I. INTRODUCTION

The TSA, Transportation Security Administration, has had difficulty determining the amount of operating costs to allocate towards customer service. Many airports do not have an adequate number of security lines causing the travelers to wait in line for a long period of time. Other airports have too many lines open, which increases the cost to operate. TSA chose to invest in software that will determine the number of lines to keep open depending on the arrival rate of the travelers. A simulator for wait times in a line was created at Bloated Government Contracts, Inc. The inputs of the program were the arrival rate, which is the average number of travelers entering the line at the security checkpoint per minute, and the maximum number of lines that can be open depending on personnel and equipment availability. The output of the program was the average wait times of travelers at the security checkpoint over a 12 hours period for each potential number of lines. The simulation utilized a linked stack, a linked queue, and a linked deque to model when a person enters and leaves the line at the security checkpoint. The hypothesis is that a linked queue is the best linked structure for traveling through a security checkpoint because every traveler will eventually exit the line in the order, first in first out (FIFO).

II. EXPERIMENTAL METHODS

A. *Linked Stack*

A linked stack is a collection of objects that are inserted and removed according to the last in first out model. The linked structure to hold the collection of objects was a doubly linked list. The main functions within the linked stack are: push(e), pop(), and peek(). push(e) puts the new object e onto the top of the stack. pop() removes and returns the object that is on the top of the stack. peek() returns the object on the top of the stack but does not remove the object.

B. *Linked Queue*

A linked queue is a collection of objects that are inserted and removed according to the first in first out model. The linked structure to hold the collection of objects was a doubly linked list. The main functions within the linked stack are: offer(e), poll(), and peek(). offer(e) puts the new object e onto the bottom of the queue. poll() removes and returns the object that is on the top of the queue. peek() returns the object on the top of the queue but does not remove the object.

C. *Linked Deque*

A linked deque is a collection of objects that are inserted and removed from both the top and bottom of the deque. The linked structure to hold the collection of objects was a doubly linked list. The main functions within the linked stack are: offer(e), offerFirst(e), poll(), pollLast(), peek(), and peekLast(). offer(e) puts the new object e onto the bottom of the deque. offerFirst(e) puts the new object e onto the top of the deque. offer(e) and offerFirst(e) were randomly chosen in the simulation. poll() removes and returns the object that is on the top of the deque. pollLast() removes and returns the object that is on the bottom of the deque. poll() and pollLast() were randomly chosen in the simulation. peek() returns the object on the top of the deque but does not remove the object. peekLast() returns the object on the bottom of the deque but does not remove the object.

D. *Simulation*

Each minute, a random number of people arrived at the security checkpoint by calling getRandomNumPeople(). Each person entered the shortest line, one at a time. In the same minute, two people or fewer, if there is less than two people in a line, advanced through the checkpoint from each line and their wait time was recorded. The wait times of the travelers were recorded for 720 minutes, which is 12 hours, and the average wait time for travelers that proceeded through the checkpoint was documented. The people remaining in line after 720 minutes were discarded. The explanation described one iteration for a given number of lines. To replicate the experiment, the simulation was run for 50 iterations with the same parameters and the average, average wait time was calculated. The average, average wait time was reported to a .csv file and the console. The simulations were performed for each possible number of lines for a total of 10 lines and arrival rates ranging from 1 person a minute to 30 people a minute.

III. RESULTS

A. *Linked Queue*

For an arrival rate of 5 travelers per minute using a linked queue, the traveler waited 0 minutes for 3 to 10 lines. With 1 and 2 lines, the traveler waited on average for 215 and 72 minutes, respectively. For an arrival rate of 10 travelers per minute, the traveler waited 0 minutes for 6 to 10 queue lines. The traveler waited on average 287, 215, 143, 72, and 2 minutes for lines lengths of 1 through 5. For an arrival rate of 15 travelers per minute, the traveler waited 0 minutes for 8 to 10 queue lines. With lines length of 1 to 7, the traveler waited on average for 311, 263, 215, 167, 118, 71, and 24 minutes, correspondingly. In Fig. 1 and Fig. 2 show the results for an arrival rate of 18 travelers per minute using a linked queue. The only number of queues that the travel waited 0 minutes was for 10 queue lines.

```
Queue
Average arrival rate: 18
Average wait time using 1 queue(s): 319
Average wait time using 2 queue(s): 279
Average wait time using 3 queue(s): 239
Average wait time using 4 queue(s): 199
Average wait time using 5 queue(s): 159
Average wait time using 6 queue(s): 119
Average wait time using 7 queue(s): 79
Average wait time using 8 queue(s): 39
Average wait time using 9 queue(s): 2
Average wait time using 10 queue(s): 0
```

Fig 1. The average wait time (minutes) for each queue line with an arrival rate of 18 travels per minute.

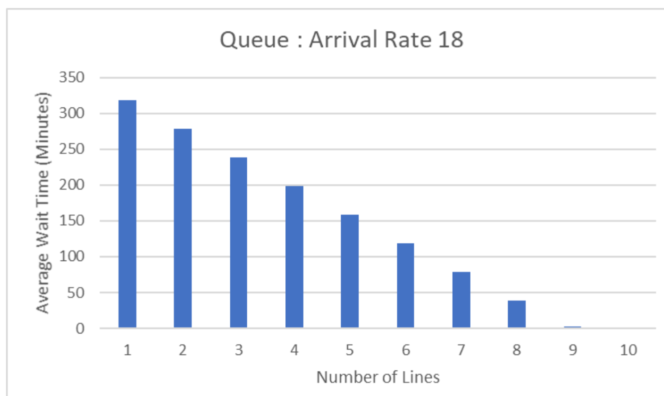

Fig 2. The average wait time (minutes) for each queue line with an arrival rate of 18 travels per minute.

B. *Linked Stack*

For an arrival rate of 5 and 15 travelers per minute using a linked stack, the traveler waited 0 minutes for all 10 lines. For an arrival rate of 10 and 18 travelers per minute, the traveler waited 0 minutes for 9 of 10 lines. The remaining queue had an average wait time of 1 minute, as shown in Fig. 3 and Fig. 4.

Stack
Average arrival rate: 18
Average wait time using 1 stack(s): 0
Average wait time using 2 stack(s): 0
Average wait time using 3 stack(s): 0
Average wait time using 4 stack(s): 0
Average wait time using 5 stack(s): 0
Average wait time using 6 stack(s): 0
Average wait time using 7 stack(s): 0
Average wait time using 8 stack(s): 0
Average wait time using 9 stack(s): 1
Average wait time using 10 stack(s): 0

Fig 3. The average wait time (minutes) for each stack line with an arrival rate of 18 travels per minute.

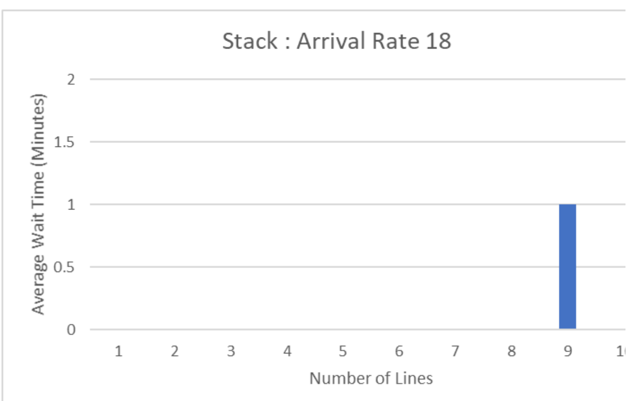

Fig 4. The average wait time (minutes) for each stack line with an arrival rate of 18 travels per minute.

C. *Linked Deque*

For an arrival rate of 5 travelers per minute using a linked deque, the traveler waited 0 minutes for 9 of 10 lines. The remaining deque had an average wait time of 2 minutes. For an arrival rate of 10 travelers per minute, the traveler waited 0 minutes for 8 of 10 lines. The remaining deques had an average wait time of 2 and 3 minutes. For an arrival rate of 15 travelers per minute, the traveler waited 0 minutes for 7 of 10 lines. The remaining deques had an average wait time of 1, 2, and 6 minutes. In Fig. 5 and Fig. 6 show the results for an arrival rate of 18 travelers per minute using a linked deque. There were 3 deques had an average wait time of 1, 4, and 2 minutes.

Deque
Average arrival rate: 18
Average wait time using 1 deque(s): 0
Average wait time using 2 deque(s): 0
Average wait time using 3 deque(s): 0
Average wait time using 4 deque(s): 0
Average wait time using 5 deque(s): 0
Average wait time using 6 deque(s): 0
Average wait time using 7 deque(s): 1
Average wait time using 8 deque(s): 4
Average wait time using 9 deque(s): 2
Average wait time using 10 deque(s): 0

Fig 5. The average wait time (minutes) for each deque line with an arrival rate of 18 travels per minute.

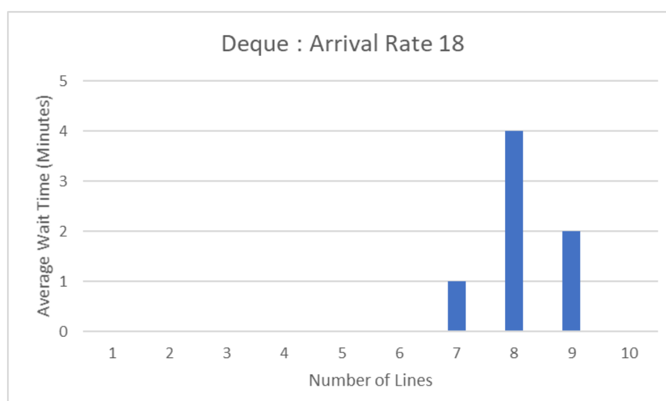

Fig 6. The average wait time (minutes) for each deque line with an arrival rate of 18 travels per minute.

IV. ANALYSIS AND INTERPRETATION

A. *Linked Queue*

The linked queue accurately simulated a traveler waiting in line. The traveler arrived at the security checkpoint and began waiting in line. Each person that arrived before the traveler will go through the check point before the traveler does. The optimal average wait time for travelers was between 0 and 45 minutes. For arrival rates of 5, 10, 15, and 18 travelers per minute, the ideal number of lines were 3, 5, 7, and 8 at 0, 2, 24, and 39 minutes, respectively. Because the linked queue utilized the first in first out model, every traveler who enters the line will ultimately leave the line. Thus, a linked queue in the best linked structure for allowing each traveler to go through the security checkpoint and wait an average of 0 to 45 minutes.

B. *Linked Stack*

The linked stack poorly simulated a traveler waiting in line. On average, 18 travelers arrived at the security checkpoint and were added to the front of the line. The last two travelers that arrived in that minute are able to go through the security checkpoint. The rest of the travelers remain in line. The following minute another 18 travelers arrived at the security checkpoint and were added to the front of the line, etc. The travelers who do not initially go through the security checkpoint in the same minute of arriving will always remain in the line. Since the greatest wait time for a stack was 1 minute for all stack lines with arrival rates of 5, 10, 15, and 18 travelers per minute, the ideal number of lines would be 1 stack line. However, a linked stack is a poor linked structure to simulate traveling through a security checkpoint because some travelers wait in line indefinitely due to the last in first out model.

C. *Linked Deque*

The linked deque poorly simulated a traveler waiting in line. On average, 18 travelers arrived at the security checkpoint. Half of the travelers were added to the front of the line and half of the travelers were added to the back on the line. The first and last travelers in the queue was able to go through the checkpoint 50 percent of the time. Or, both the of first or both of the last travelers in the queue were able to go through the checkpoint, each 25% probability. The rest of the travelers remain in line. Another 18 travelers arrived at the security checkpoint in the next minute, etc. Since the greatest wait time for a deque was 6 minutes for all deque lines with arrival rates of 5, 10, 15, and 18 travelers per minute, the ideal number of lines would be 1 deque line. However, a linked deque is a poor linked structure to simulate traveling through a security checkpoint because some travelers wait in line indefinitely.

V. CONCLUSIONS

The best model to simulate traveling through a security checkpoint is a queue. The queue follows the first in first out model allowing each traveler to ultimately leave the line. For arrival rates of 5, 10, 15, and 18 travelers per minute, the ideal number of lines were 3, 5, 7, and 8 at 0, 2, 24, and 39 minutes, correspondingly. The linked stack and linked queue are poor models to simulate traveling through a security checkpoint. Both the linked stack and the linked deque allows some travelers to wait in line indefinitely. Even though these linked structures had almost no wait time, the linked stack and linked queue should not be

Modeling the Optimal Number of Lines at an Airport for Various Arrival Rates of Travelers

implemented by the Transportation Security Administration.