System Modelling:

We are creating all customer threads in our main function,in main function We will get size of set customers by using random generator We will create the total number of threads of that size after that we will Get delay time by using random generator then other set of threads will be created Only after that delay time this whole process will continue till all customer Threads gets created

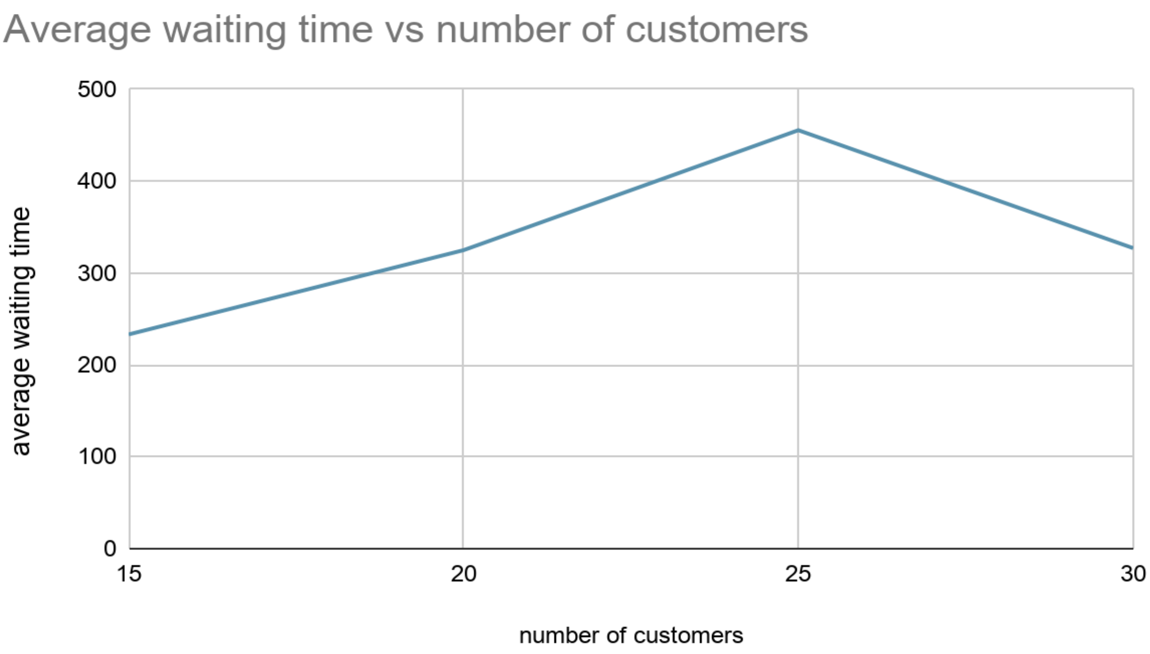
Working of Algorithm and implementation details

```
We will lock mutex so no other process can enter (by using sem_wait(&mutex))
 When one customer thread access request then we will check whether the table is full or not and we will also check whether there are any threads waiting are not
 if either the table is full or any threads are waiting then
                 We will unlock mutex because this process is going to wait   (by using sem_post(&mutex))
                  We will make our current thread also wait(by using sem_wait(&x_sem))
 Else
       We will allow current thread to eat
        We will unlock mutex(by using sem_post(&mutex))
  After eating,our current thread will check whether the table is empty or not
  If the table is empty then
           If there are more process waiting compared to number of tables then
                  We will signal process which are waiting till number of tables is full (by using sem_post(&x_sem))
           Else
                 We will signal all process which are waiting (by using sem_post(&x_sem))
 We will unlock mutex so other process can enter  (by using sem_post(&mutex))
```

We can keep track number of people eating currently and number of people Waiting by using variables,

Note: i am using global array to store times we will not get synchronization issues because each thread will access its own location based on thread id In an array for example thread id 2 will access a[2] and thread id 3 will access a[3]

Analysis of Graph1



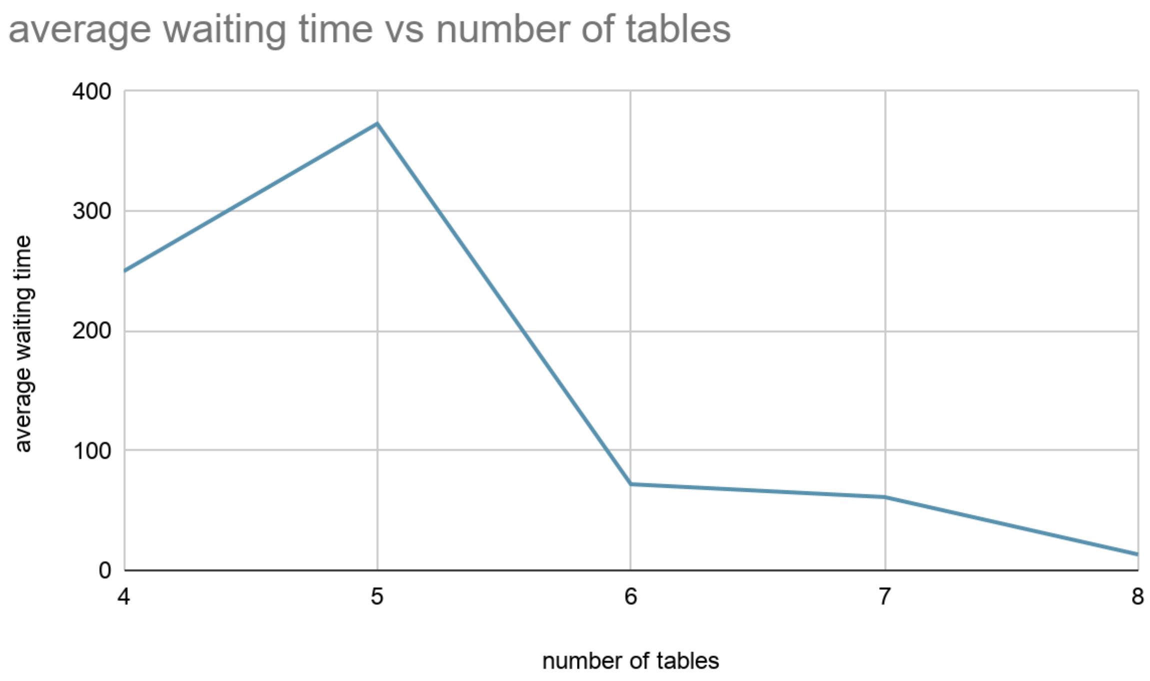
Explanation: Usually when number of customers increases average waiting time should increase because customer will wait more time if there are more number of customers

case 1: if there are more number of customers then average waiting time increases
case 2: if there is more delay between creation of one set of customers(threads) and other set of customers(threads) then average waiting time decreases because if there is more delay in creation of one set of customers then those set of customers need to wait less time compared to Set of customers whose delay is less because They were not created early

From number of customers = 15 to 25 the benefit from case 2 is negligible
or delay may be nearly equal when number of customers =15,20,25
So it is following case 1 that means as number of customers increases average waiting time also increase
From number of customers = 25 to 30
From 25 to 30 average waiting time decreases because the benefit which are getting Form case 1 may gets dominated by case 2 that means if set of threads are created early then they need to wait more time because they are created early so when

number of customers =25 the set of threads may get created so much early compared To when number of tables =30 because of that the case 1 could not able to dominate Case2

Analysis of Graph2



Explanation: Usually when number of tables increases average waiting time should Decrease because customers will wait less time if number of tables increases

case 1: if there are more number of tables then average waiting time decreases
Because customer needs to wait less time

case 2: if there is more delay between creation of one set of customers(threads) and other set of customers(threads) then average waiting time decreases because if there is more delay in creation of one set of customers then those set of customers need to wait less time compared to Set of customers whose delay is less because They were not created early

Note: delay time will be produced by random generator

For number of tables =4 to 5

From 4 to 5 average waiting time increases because the benefit which are getting Form case 1 may gets dominated by case 2 that means if set of threads are created early then they need to wait more time because they are created early so when number of tables =5 the set of threads may get created so much early compared To when number of tables =4 because of that the case 1 could not able to dominate Case2

For number of tables =5 to 6

From 5 to 6 average waiting time decreases so much because at number of tables=6

We may get benefit form both case 1 and case 2 whereas when number of tables=5
We are not getting benefit from case1 because number of tables is less
And we may not even get benefit from case 2 because from 4 to 5 there is increase in graph

For number of tables =6 to 8

From 6 to 7 and 7 to 8,graph is not decreasing so much because benefit from case2
Is negligible or delay may be nearly equal when number of tables =6,7,8
But number of tables are increasing so average waiting time will be less