AIM:

To find optimal number of charging piles for Fast Charging Stations (FCS).

NOMENCLEATURE

p_n	Steady State probability of n customers in the system
l	Arrival Rate (number of customers per time period)
l_{eff}	Effective arrival rate
l_n	Arrival rate, given n customers in the system
m	Service Rate (number of customers per time period)
m_n	Departure Rate, given n customers in the system
ρ	Server utilization
L_s	Long-run time average number of customers in system
L_q	Long-run time average number of customers in queue
W_s	Long-run average time spent in system per customer
W_q	Long-run average time spent in queue per customer
N	Number of customers in the system
$\bar{c}$	Number of busy servers
d_c	Density of EV (number of EV/km^2)
r	Radius of service station (km)
z	The distance that has been travelled before recharging (km)
C_{ev}	Capacity of battery (KWh)
SOC	State of Charge
h	No. of hours
C^{ele}	Electricity consumed (KWh/km)
$r_{charging}$	Electricity charging speed (KW)

TYPICAL VALUES OF PARAMETERS:

Parameters	Values	Parameters	Values
r	1.24km	c_{ev}	30kWh
d_c	150 EVs/km^2	$r_{charging}$	40kW
z	50km	h	16h
C^{ele}	0.2kWh/km	SOC	0.3

METHODOLOGY:

In this present work, the optimum number of charging piles for FCS is carried out using Aspiration level model. The capacity of FCS is assumed to be infinite. The average arrival rate at FCS is 6.90 EVs/h and average service rate is 2 EVs/h.Aspiration level model evaluates suitable service level for queuing systems. It considers two conflicting parameters: (1) The average time spent by EV in system (W), (2) The idleness percentage of charging piles (X). The solution is to find the acceptable range of c (charging piles) such that: $W \leq \gamma$ and $X \leq \delta\%$. γ and $\delta\%$ are entered by the user. Equations (1)-(8) evaluates the queue measures for infinite system capacity queuing model.

$$l_n = l \text{ , } n \geq 0 \tag{1}$$

$$\mu_n = \begin{cases} nm, & n<c \\ cm, & n \geq c \end{cases} \tag{2}$$

Where, n takes integer values starting from zero. In this case the queueing measures are given as:

$$p_n = \begin{cases} \frac{\rho^n}{n!} p_0, & n<c \\ \frac{\rho^n}{c!c^{n-c}}\ p_0, & n \geq c \end{cases} \tag{3}$$

$$p_0 = \left\{ \sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c}{c!} \left(\frac{1}{1-\frac{\rho}{c}} \right) \right\}^{-1} \tag{4}$$

Where $\left(\frac{\rho}{c}\right) < 1$

$$L_q = \frac{\rho^{c+1}}{(c-1)(c-\rho)^2} p_0 \tag{5}$$

$$L = L_s + \rho \tag{6}$$

$$W = \frac{1}{m} \tag{7}$$

$$W_q = W - \frac{1}{m} \tag{8}$$

The inputs which the user has to enter are explained below:

Input 1-> Average arrival rate of FCS and Average service rate of FCS (should be positive)

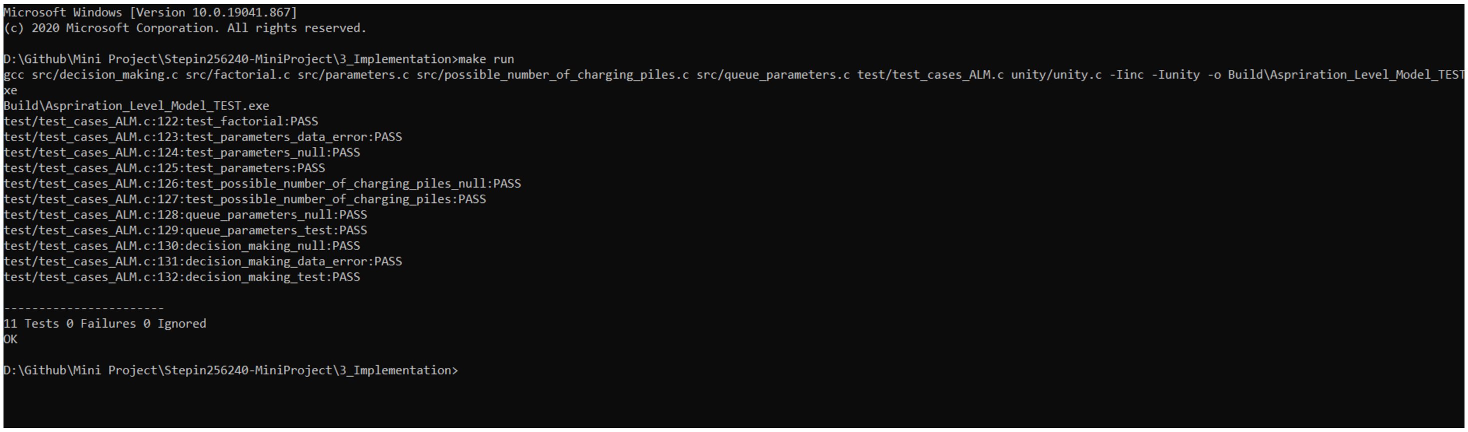
Input 2-> Maximum number of charging piles (should be positive)

Input 3-> Threshold on Average time spent by EV in FCS (should be positive, enter in minutes) and Threshold on percentage idleness of charging piles (should be positive)

The minimum and maximum optimal charging piles are the output of the program. If the optimal charging piles are not found the program returns -1 and -1.

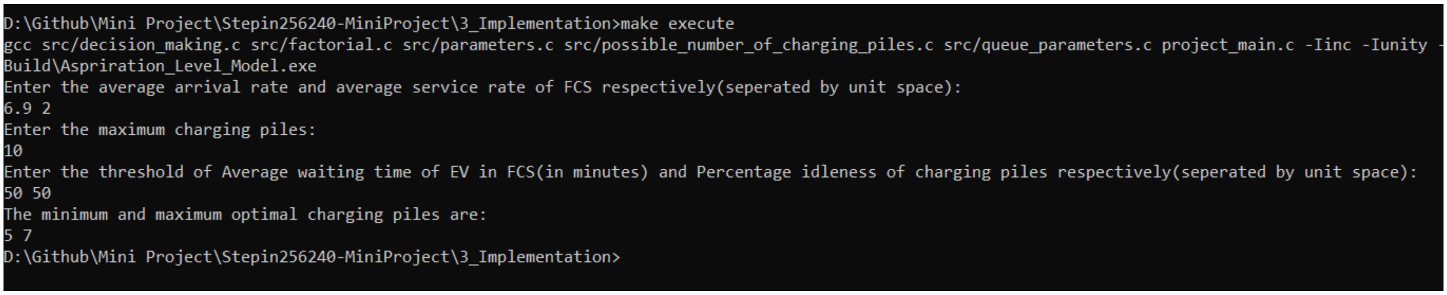
OUTPUTS:

Test Cases:



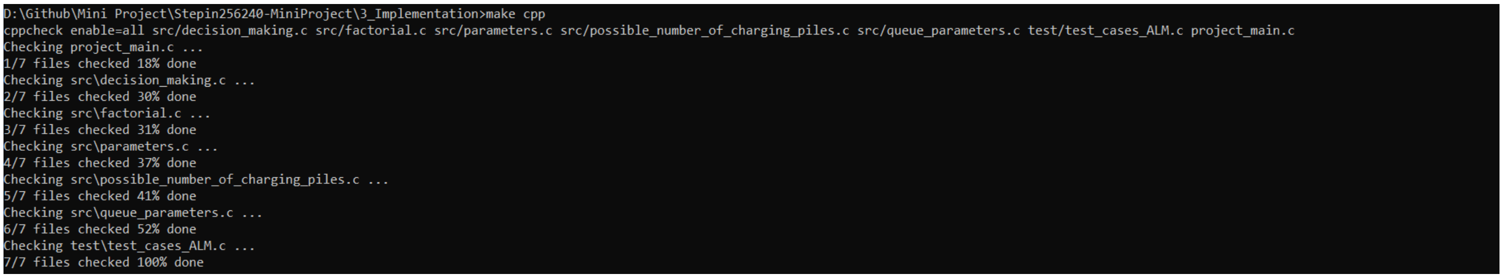
```
Microsoft Windows [Version 10.0.19041.867]
(c) 2020 Microsoft Corporation. All rights reserved.

D:\Github\Mini Project\Stepin256240-MiniProject\3_Implementation>make run
gcc src/decision_making.c src/factorial.c src/parameters.c src/possible_number_of_charging_piles.c src/queue_parameters.c test/test_cases_ALM.c unity/unity.c -Iinc -Iunity -o Build\Aspriration_Level_Model_TEST
xe
Build\Aspriration_Level_Model_TEST.exe
test/test_cases_ALM.c:122:test_factorial:PASS
test/test_cases_ALM.c:123:test_parameters_data_error:PASS
test/test_cases_ALM.c:124:test_parameters_null:PASS
test/test_cases_ALM.c:125:test_parameters:PASS
test/test_cases_ALM.c:126:test_possible_number_of_charging_piles_null:PASS
test/test_cases_ALM.c:127:test_possible_number_of_charging_piles:PASS
test/test_cases_ALM.c:128:queue_parameters_null:PASS
test/test_cases_ALM.c:129:queue_parameters_test:PASS
test/test_cases_ALM.c:130:decision_making_null:PASS
test/test_cases_ALM.c:131:decision_making_data_error:PASS
test/test_cases_ALM.c:132:decision_making_test:PASS

-----------------------
11 Tests 0 Failures 0 Ignored
OK

D:\Github\Mini Project\Stepin256240-MiniProject\3_Implementation>
```

Build:



```
D:\Github\Mini Project\Stepin256240-MiniProject\3_Implementation>make execute
gcc src/decision_making.c src/factorial.c src/parameters.c src/possible_number_of_charging_piles.c src/queue_parameters.c project_main.c -Iinc -Iunity -
Build\Aspriration_Level_Model.exe
Enter the average arrival rate and average service rate of FCS respectively(seperated by unit space):
6.9 2
Enter the maximum charging piles:
10
Enter the threshold of Average waiting time of EV in FCS(in minutes) and Percentage idleness of charging piles respectively(seperated by unit space):
50 50
The minimum and maximum optimal charging piles are:
5 7
D:\Github\Mini Project\Stepin256240-MiniProject\3_Implementation>
```

Coverage:



```
D:\Github\Mini Project\Stepin256240-MiniProject\3_Implementation>make cpp
cppcheck enable=all src/decision_making.c src/factorial.c src/parameters.c src/possible_number_of_charging_piles.c src/queue_parameters.c test/test_cases_ALM.c project_main.c
Checking project_main.c ...
1/7 files checked 18% done
Checking src\decision_making.c ...
2/7 files checked 30% done
Checking src\factorial.c ...
3/7 files checked 31% done
Checking src\parameters.c ...
4/7 files checked 37% done
Checking src\possible_number_of_charging_piles.c ...
5/7 files checked 41% done
Checking src\queue_parameters.c ...
6/7 files checked 52% done
Checking test\test_cases_ALM.c ...
7/7 files checked 100% done
```

Result:

The execution to find the optimal charging piles for the FCS has been completed successfully using C program.