ECM2433 - The C Family

Traffic Simulation Report

2022

1 Design Choices

1.1 Assumptions Made

While completing the project, I made the following assumptions:

- It takes a car one time period to cross the lights (leave the queue and reach the other side).
- The traffic lights switch between green and red every light period iterations no matter the state of the queues. Moreover if one queue is empty and the other has cars in, then the lights will not favour the queue with cars in it.
- There are no other delays to cars other than waiting for the lights, meaning when the light is green, if there is a car in the corresponding queue it can always move.
- The traffic lights always begin with being green for the left-hand side queue and red for the right-hand side queue.
- Each iteration of the simulation is one time period.
- It takes the traffic lights one time period to change colour and while the lights are changing colour, no cars can leave or join either queue.

1.2 Design Decisions

When designing the platform I chose to structure the project in three different (pairs of .c and .h) files. 'queue.c' and 'queue.h' are responsible for creating Queue and QNode structures which represent each side of the traffic flow. I did this because it abstracted the complexities of the queues from the rest of the project. 'runOneSimulation.c' and 'runOneSimulation.h' are used to run a single simulation using the runOneSimulation function which carries out the steps described by the specification, and finally 'runSimulations.c' and 'runSimulations.h' contains the main function responsible for taking in the four parameters from the user and using them to run 100 simulations outputting the average result.

Another design choice I made was to use the GNU Scientific Library. This is because, we are looking for accurate statistical modelling in the simulation and therefore we need to use a function which will provide high-quality pseudo-random numbers. I chose to use the 'gsl_ran_flat' function because it uses uniform distribution which is appropriate for getting a random number between 1 and 100 where all the values have an equal probability of being selected.

Along with the structures created to manage a Queue and QNode, I chose to create a structure to handle all the data (the results) of a simulation. This structure has a variable for both the left and right hand side number of vehicles, maximum wait time, average wait time and clearance time. I chose to do this as it is the simplest and most efficient way of returning multiple variables from functions.

A key feature of the simulation is queues, I chose to implement them using linked list, this allowed dynamic memory allocation so the queues would only ever use as much memory as is required of them and also would mean that I can easily free any memory which is no longer required to ensure there are no data leaks and memory wastage.

1.3 Drawbacks of my Design

Due to how I have interpreted the specification and made the assumption that cars cannot join either queue when the lights are changing colour, when the traffic period is increased, the queue related to it will have more cars added in the first 500 iterations due to the lights changing less hence there are more opportunities for cars to join. I believe this means that the simulation does not accurately represent a real like traffic situation as cars should be able to join when the lights change even if they cannot leave. As cars cannot join the queues while the lights change, it means that modifying the traffic light period affects the number of cars which are in each queue which is not accurate to real life.

2 Experimentation

2.1 Experiment Description

The purpose of my experiment is to investigate the effects of changing the traffic light period on the average waiting time of a car. By using an implementation of the algorithm outlined in the specification and following the design choices described in the above section, I have run tests increasing the traffic light period by one each time while keeping all other parameters constant to ensure a fair investigation. I have kept the left and right traffic light time periods the same for each test.

2.2 Results

The following the results of the investigation. Figures one and two show the tables of results when increasing the traffic light time period by one each time and figure three shows a graph of traffic light time period against average waiting time for both queues in the tests.

Results for the Left-Hand Queue

Traffic Light Period	Number of Vehicles	Average Waiting Time	Maximum Waiting Time	Clearance Time
1	126	8	19	12
2	166	8	21	11
3	185	8	21	11
4	199	10	24	14
5	209	10	25	16
6	213	11	26	16
7	217	11	27	16
8	220	12	29	16
9	225	13	32	18
10	227	13	31	19

Figure 1: Results of the left-hand-side queue for varying traffic light period.

Results for the Right-Hand Queue

Traffic Light Period	Number of Vehicles	Average Waiting Time	Maximum Waiting Time	Clearance Time
1	125	7	18	11
2	165	8	20	11
3	184	9	21	11
4	200	10	26	16
5	207	11	25	14
6	213	11	27	14
7	220	12	29	17
8	222	13	26	17
9	225	13	30	17
10	227	13	30	17

Figure 2: Results of the right-hand-side queue for varying traffic light period.

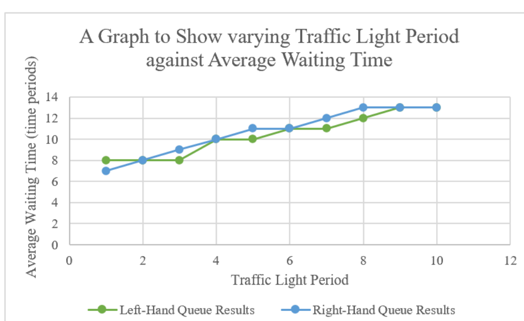

Figure 3: A graph to show the affects of varying the traffic light period against the average waiting time of each queue.

2.3 Conclusion and Analysis

The results seen from figure 3 show that as you increase the traffic light period the average waiting time increases linearly. This is not what we would expect from a similar scenario in real life, what we would expect is that the average waiting time would decrease at to an optimum traffic light period before increasing again. I believe the reason for the results I have obtained is related to the drawbacks of my design described above which describes how when you increase the traffic light period, the number of vehicles increases which will increase the waiting time.

The overall meaning of these results is that, if you have a traffic situation where cars can only join the queue when the lights are not changing the best traffic light period is one, where the lights change every other time period. This is because with a traffic light period of one, the queues are filled with less cars and move more frequently as neither side builds up a large queue.

If I were to complete this investigation again, I would modify the algorithm for running a simulation to allow cars to be able to join the queue when the lights are changing. This would allow the number of cars in each queue to not depend on traffic light period and therefore would create a more accurate simulation real life results.

3 Example Output

The following is an example of the output of running 'runSimulations' with the following parameters: Left Traffic Arrival Rate - 1, Left Traffic Light Period - 50, Right Traffic Arrival Rate - 1 and Right Traffic Light Period - 50.

Therefore the full command used was:

```
$ ./runSimulations 1 50 1 50
```

```
Parameter values :
        from left:
                traffic arrival rate:   50
                traffic light period:   1
        from right:
                traffic arrival rate:   50
                traffic light period:   1
Results (averaged over 100 runs):
        from left:
                number of vehicles :    125
                average waiting time :  7
                maximum waiting time    18
                clearance time :        10
        from right:
                number of vehicles :    124
                average waiting time :  7
                maximum waiting time    17
                clearance time :        9
```

Figure 4: An example output of the project.