**Description of Current Work**

The Picture 1 shows the visualization of the current work of the traffic simulator.

|  |
| --- |
| * Yellow rectangles represent taxis. * Blue rectangles represent cars on the right lanes. * Red rectangles represent cars on the left lanes. * Pink rectangle represents the called taxi. * Yellow star represents the crashed car * Green lines represent the major roads. If a road is longer than 200 meters (current setting), we mark it a major road. If a road is connecting two major roads, we also mark it a major road. |

Picture 1. The visualization of the traffic simulator.

**Simulation Setting**

* Initial number of cars including the crashed car: 500
* Initial number of taxis: 20
* Lambda for the Poisson arrival for general roads: 0.00002
* Lambda for the Poisson arrival for major roads: 0.00002\*3
* The time between
* All random function is fixed by random seeds.
* Road speed limit: 60 km/hr.
* When a car crashed, the speed limit of the road will be set to 10 km/hr.
* Every line on the map has two roads (two directions) and every road has two lanes.
* Initially generate 80% cars on the major roads.
* The crash event will happen after 5 minutes when the simulation starts.
* The random function in the simulator will be fixed by given seeds. In other words, if two experiments’ settings are the same, their results will be the same.

**Simulation Process**

* Every car will have a destination and it will find a routing based on the traffic situation when the destination is assigned.
* When a car arrives its destination, it will be removed from the map.
* New cars will be added into the map according to a Poisson arrival process (the lambdas are given above).
* Every taxi also has its own destination.
* Before the crash event happens, when a taxi arrives its destination, a new destination will be assigned to that taxi.

For the simulation, when a crash happens (we now fixed the crash location), the simulator will assign the quickest taxi (measured by estimated arrival time) for the crash event and call rest of the taxis to go to the same location as well. The taxis will update their routing every 30 seconds in response to the changes in traffic (the program will find the quickest routing according to the 5-minutes average speed of each road). We also compared the result with that the taxis don't update their routing periodically to see whether the information can help minimized the response time.

|  |
| --- |
|  |

Figure 1.

The experiment results show that with the average speed and updating the navigation periodically, the fastest taxi we found in the beginning will not necessarily arrive first. One possible reason I thought is that the past information (average speed or even a pattern of the traffic) is not enough to guarantee