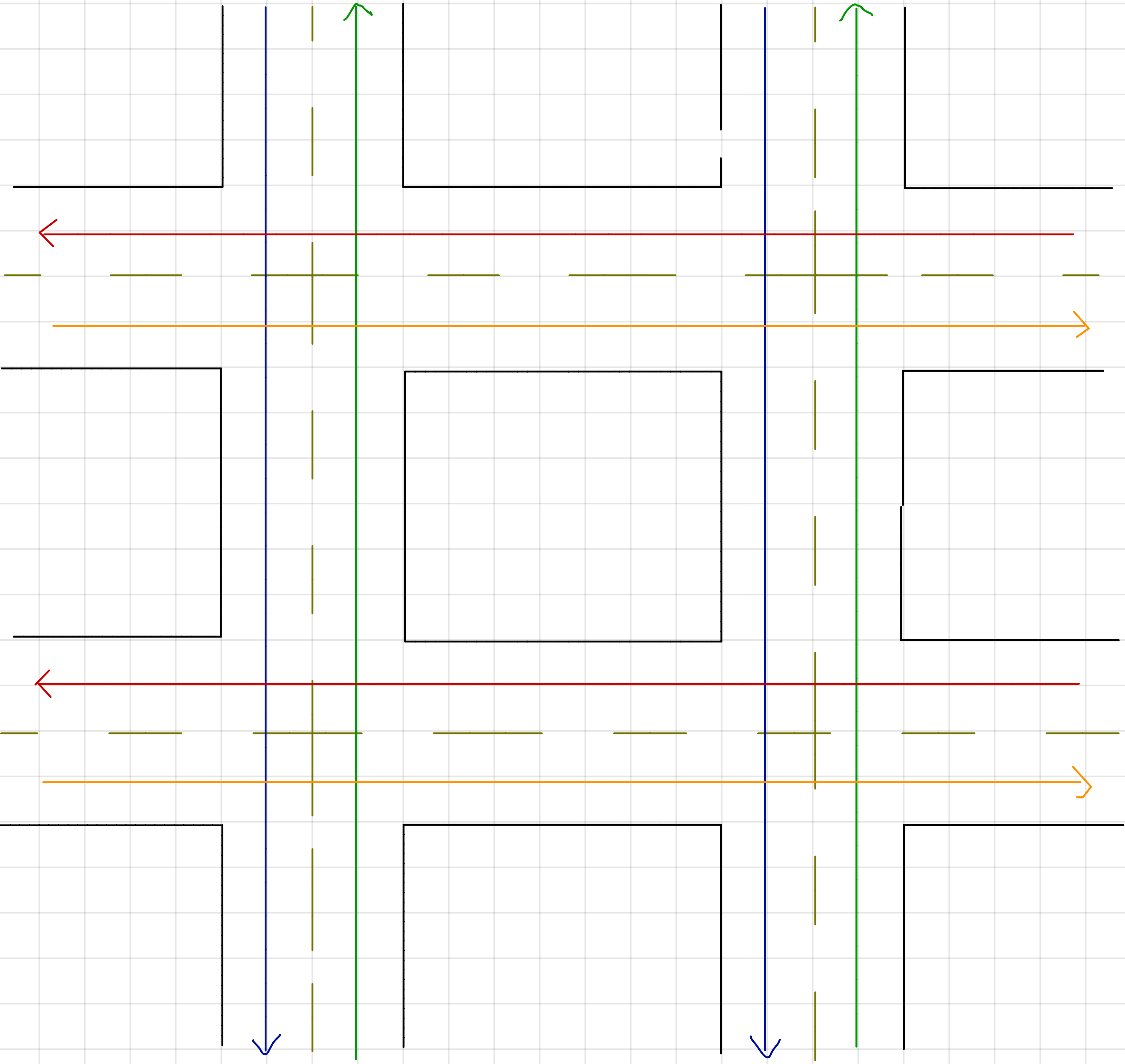
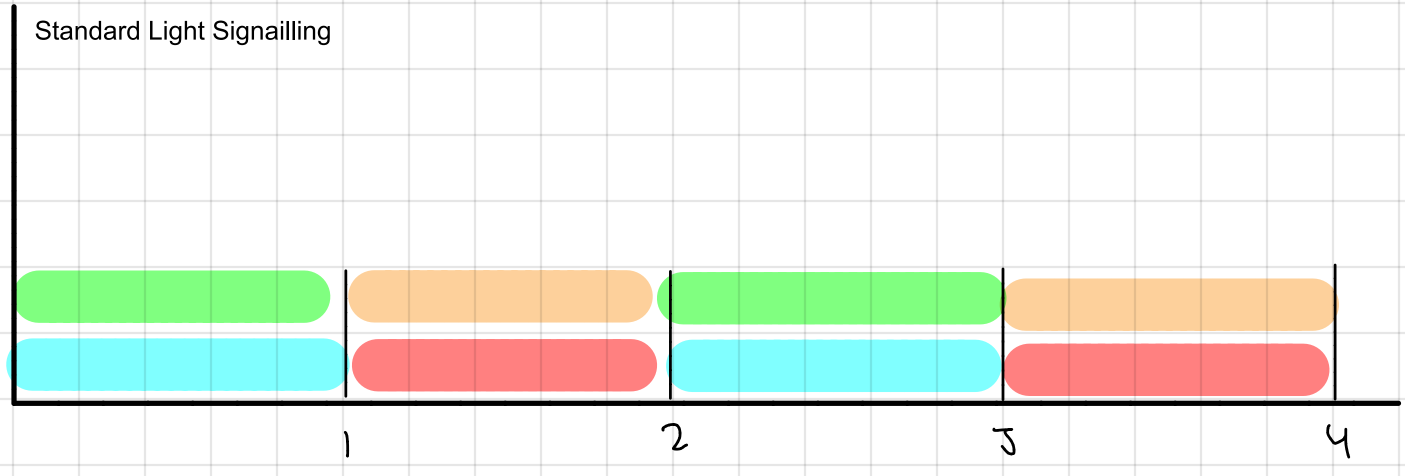
UOTTAWA HACKATHON REASEARCH PAPER ON TRAFFIC ALGORITHM DESIGN

Fozail Ahmad
Collin Barber
KunYi Wang
Abby Leung




Standard Light Signalling
1
2
3
4

In a standard light signalling system each direction gets a time quantum that they get regardless of the use for it.

This is not efficient as it gives time recourses to lanes that do not need them while potentially depriving them from a system that could use it. Also it doesn't account for the fact that start stop traffic is not good for a traffic network and reducing that is better than absolute fairness is current system

This paper will cover the multi step process of a alogorthm that aims to avoid idling lights, that aims to clear paths non stop traffic movement. This algorithm is implemented in a 4 intersection grid can however we scaled out to cover much larger cities.

We are using light sensor on every intersection facing every lane to know how many cars want to go straigh, turn left and turn right, therefore from every standard intersection we get 12 points of data. This would mean that if we were to create a bruise force algorithm that would take the decision tree from every node in every intersection and combine all of the intersections we wold expenetially from 4^intersections and if not larger! This many nodes to computer in real time is not feasible hence we have to reply on heuristic algorithms to solve this problem that come as close as possible to perfect optimization.

There are different levels of utilization for a lane. Straight gong cars use their time such that no one else could make use of the road from adjoining lanes other than opppsite travelling direction cars. Turn cars would allow the diagonally opposite turning cars to go, side cars to potentially go and opposite direction cars to travel. Based on this knowledge we can allocate the time to other lanes dynamically on runtime

There are two parts we are trying to solve. Allocating unused time quanta and over allocating time quanta to reduce traffic stopping.

Part 1: Remove idle time

We design a system that uses a simple sprodic server to allocate time to each lane. Therefore we start off with every lane having as single quanta of time (q) and if there is no car present to see the system will allocate that time to any other lane that has the highest density of cars at the intersection that can use it. This doesn't mean that you loose your time quantum completely, if another car shows within the time quantum it will get the most priority, But if the car is turning we could let other turning still operate. The point of the algorithm is to let a lane get full priority during its time quantum but if its not being used to allocate it accordingly to other lanes.

Algorithm: for straight

If only straight cars, then allow opposite directiion, then allow right-bound traffic to make left turns also. You can manage which one gets proroity based on density of cars in each scenario, where as if one has a 1:3 or higer ratio in density get more priority to go rather than a static allocation list previously stated.

Algorithm: for right turns

You can allow pretty much all traffic except for right bound perpendiucaular traffic. We allocate the next most dense to go since we want to clear the most traffic.

Algorithm: for left turns

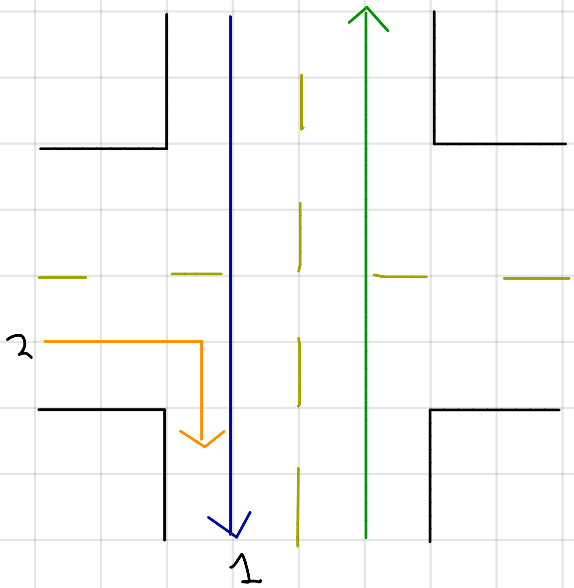
Block oncoming traffic and other left turns, will allow right bound right turns and left bound right turns unequivocally. No priority for any allowed lanes as they can all do it safely.

Algorithm: Complete

Based on all the data coming in for each direction we can apply restrictions for the safet and whatever is allow for each scenario can be implemented according to our policy.

Scenarios

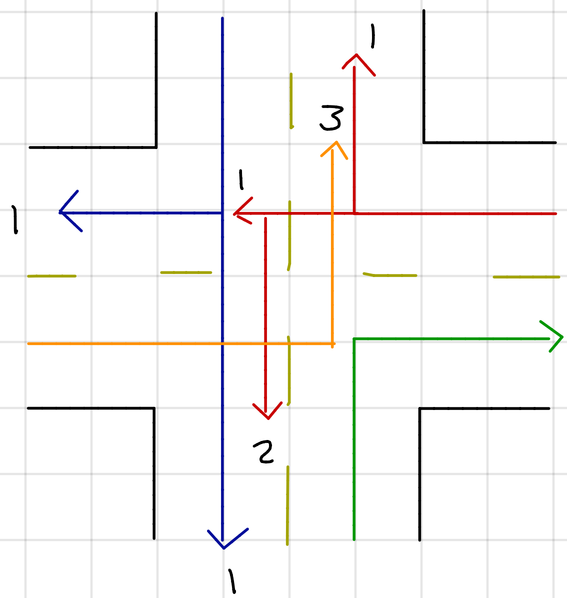
Straight Traffic



else

1:3 traffic ratio gets priority

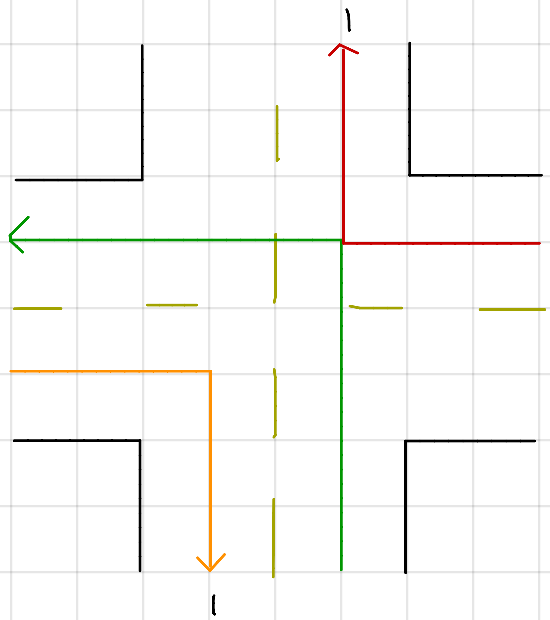
Right turns traffic



Blue gets priority, if 1:3 ratio

Otherwise traffic density rules thoroughput

Left turning traffic



Allows these turns

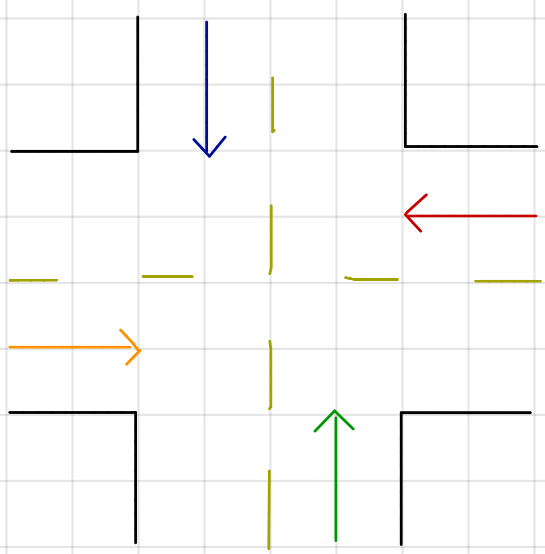
Part 2: Extending time

Current system ensure fairness of the traffic system but fairness created traffic, we want to keep cars moving!

Therefore the second past of our algorithm focuses on giving more time to busier lanes. Deciding how much time to give and in a way so no one lanes gets starved is the subject of many researches since doing so optimally is impossible due to the complexity of the problem.

Algorithm: Extra Quanta

If a particular lanes time quanta is over the algorithm will evaluate the density of traffic. If the ratio is 1:5 then the particular lane will get another quanta, and it can only use it at along as the density ratio between it and all other lanes remains so. This was we prevent starvation.



Keep on providing time until the lane stays exceptionally busy or one more qunta gets wasted.