Legenda

X={…} -> x is equal to …

Init()

For each T={simulation step}

* Qi={Queue of vehicles to be served for lane i}
* B={ Highest bid value known by all vehicles at time T}

For each Qi

For each v={vehicle} which is head(Qi)

* Try reservation()

If reservation NOT OK

Go to next time step

If reservation OK

Remove from Qi, then make v leave the crossing

Init():

For each el of Vt={list of all vehicles spawned at time T}

El.wallet = random.randint(beg, end)

reservation(vID, laneID, T):

Conflicts in my lane?

Yes -> bid()

No -> set tls(laneID) = G

bid(vID, laneID, T):

Problem: bidding can be asynchronous

Idea: vID checks the B value for s time steps.

Wait s time steps

B == (vID last offer) ?

Yes -> vID won, set tls(laneID) = G

NO -> vID lost, set tls(laneID) = r

NOTA:

* Piu votazioni? -> serve S per garantire sincronia su B??
  + Puo essere necessario un tempo max per effettuare le votazioni?
    - Potrebbe essere controllabile definendo lo “step” di valuta col quale si puo biddare es. bid di 5 in 5, 10 in 10, ecc
      * Meccanismo per evitare che ci sia troppo delay per stabilire il vincitore
* Solo una controfferta per ogni veicolo ->

controllore lavora con certa finestra temporale -> da calcolare, misure pratiche

tw 1 sec, si apre l asta, chi riesce a biddare e considerato, si dedica un po di tempo x i calcoli(dentro alla w)

di chi e riuscito a biddare, calcolo il vincitore

no controfferte!!!

chi vince passa, poi gli altri passano in funzione sua!!!

fare i calcoli prima di dare lo step a sumo ma ponendosi dei vincoli di tempo!! timer dopo cui il processo scade

finestra x accettare i bid la fissiamo noi

il tempo di calcolo chiusa la finestra di bid e guardo in coda le bid che ho -> adesso lo calcoliamo, poi lo fisseremo

GRADI

* 1 veicolo alla volta
* con condizioni di passaggio
* tassellatura