Agent Systems Group 50 Week 6

Tom Scheffers & Joris Mollinga & Ruben Polak

March 18, 2018

Last week we developed the collaborative spawning of buses, optimized parameters and implemented the gain based pick up scheme. This week we focused on the following things:

1 Improved gain based pickup

We removed the other pickup methods as the gain based pickup method was superior to the rest. Furthermore, there were some issues with the methods, so we fixed the following things:

- We made the gain threshold higher when bus was fuller, so we become more picky as it is more busy in the system.
- If the gain = 1 (shortest route to goal is equal to a part of the route), we drop off at the latest stop with gain 1 (probably the destination).
- We pick passengers up with the highest gain first, so we order the list by gain and then start picking up passengers until we are full.

2 Gain based bidding on passengers

Now we have a set threshold (say 30%) for picking up passengers, but when we have a passenger waiting at the current stop which has a gain of 50% when travelling with this bus, we pick him up. But when one tick later a bus arrives which has free-space which has a gain of 70%, it would be more efficient is this bus takes on the passenger.
Therefore, before a bus reaches the stop, it will already have to know which passengers it will take and which ones it cannot. This is why we developed a protocol in which a bus will bid on passengers before reaching the next stop. It will develop a bidding list of passengers_id and bid (equals to gain), this list contains the X highest gains on that stop for this bus where X is the capacity of the bus plus the number of passengers dropped off at the next stop.
When a bus reaches a station, it will look at the gain for each passenger. When this gain exceeds all the bids of other buses, the passenger is taken on, else it is left on the platform. We must make sure that we send the message to all buses in front of us on the same edge.

3 Stop driving buses in downtime

As the amount of buses gets very large due to high demand, there are a lot of useless buses in the afternoon. If these buses keep driving around, they keep costing money. We developed a protocol where buses stop driving when they are useless.

4 New dijkstra algorithm

The previous implemented Dijkstra algorithm had some strange errors with bus_stop 0, so we implemented a new version using the Network extension. We discovered these errors by looking at the passenger gain (from last week: $gain = (d_{old} - d_{new})/d_{travelled}$. Sometimes this gain was $gain > 1$, which is impossible by definition. Although we regret not implementing our own Dijkstra algorithm, this version is much faster and stable than our own version.