Aron Hammond
10437215
hammond756@live.nl

Jasper Bakker
10260250
jcsbakker@gmail.com

Jens Dudink
11421479
ja.dudink@gmail.com

1 Competition

1.1 Auctions

This week we augment our multi-agent system for buses by adding a mechanism for competition between buses. We implement an auction system which buses use to ascertain that specific bus stops which do not see as much traffic from buses still get visited often enough. To fully understand this idea, a little background is required. In week 5, we implemented a voting system for buses which allows for flexible route buses, buses whose routes are determined by its passengers, in addition to fixed route buses. This week, these same flexible buses gain the additional role of making sure no single bus stop is overcrowded.
We deem an auction to most accurately reflect the idea of competition, as buses directly bid for a task based on their fitness with respect to the task.

1.2 Implementation

Since information on the number of waiting passengers is available to all buses, we designate the first bus (bus 24) to spawn to be the auctioneer throughout the day. Every 10 ticks, this bus will determine if any specific bus station contains more than 150 waiting passengers. Since one auction every 10 ticks is quite often, we might implement a belief system for all buses to track whether there is an outstanding auction that has not yet been fulfilled (i.e. the bus has not arrived at the station yet), to avoid auctioning the same station twice. Note that a single large, empty bus cannot fulfill the task of relieving the station of all waiting passengers. If there is a station with more than 150 waiting passengers, the auctioneer will send out a request to all flexible buses available, except for those with a capacity of 12 (these are too small to make a difference). These buses then respond to the auctioneer with a bid to go to this bus station to pick up its passengers. The auction we employ is a first-price sealed-bid auction. That is, the highest bidder gets to fulfill the task.

Every flexible bus makes a bid to go the full station based on two parameters: the distance to the relevant bus station, and the capacity of the bus taking into account its current capacity utilization. The bid is then determined by the following formula.

$$\frac{1}{\ln(d) + 1 + \epsilon} \times c \tag{1}$$

where d is the distance to the bus station, ϵ a small offset to prevent division by zero and c is the current net capacity of the bus.

After winning an auction, the winning bus takes the shortest route from its current position to the designated bus station. When the bus which won the auction arrives at its designated station, it picks up as many waiting passengers as possible (after possibly dropping off those passengers which had that station as their final destinations), and functions according to the flexible route bus scheme described in previous weeks. That is, passengers - both waiting and aboard the bus - vote on which route to take, and a majority vote determines the ultimate direction of the bus. By this scheme, the bus ought to take as many passengers as possible with it when it arrives at the auctioned station.

It must be noted that along its way to its designated station, the winning bus functions according to the coordination scheme previously described. That is, passenger which might benefit from going to the next station (more precisely, if the shortest route to their own destination contains the bus' next stop) stay on the bus, while those who do not depart.

1.3 Justification

This system complements our coordination system, flexible bus system and fixed bus system in the following way. As fixed routes cover all edges in the graph, all passengers arrive at their destiny eventually. A problem, however, arises when passengers travel along a fixed bus route that has fewer buses than others, because they only require one or two stops of that route (and thus most buses are empty, resulting in no new buses being added along the route). Competition among flexible buses allow us to counteract these bottlenecks by addressing them as they occur.