Multi-Agent Transportation System

Laura Ruis
University of Amsterdam
10158006

Fredie Haver
University of Amsterdam
10185453

Tomáš Fábry
University of Amsterdam

1 Assignment 1

A multi-agent system for public transport in Amsterdam is developed in the NetLogo environment. The intelligent agents in the system are buses with the objective of taking all passengers to their desired destination as time and cost efficient as possible. This paper will describe the strategies chosen and assumptions made during the design of the system. In the first section the model of the system will be described. In the second section the problem is described and in the third section a first approach to solving the problem is discussed.

1.1 The Model

The city of Amsterdam is modeled as an undirected graph, where vertices correspond to bus stops and edges indicate possible direct connections between these bus stops. As mentioned before, the intelligence of the system is contained in the buses that travel between bus stops as efficient as possible. Furthermore, the system contains passengers modeled with a unique passenger ID and a desired destination. There are twenty-three bus stops and the graph used to model the public transport network is not fully connected, meaning there is not a direct edge between all the vertices. The edges are labeled with a number denoting the distance between two stops. The vertices have some number of passengers waiting at each time point, and keep track of the number of passengers already arrived there. There are three types of buses, denoting the maximum amount of passengers they can travel with. The buses have access to the information of bus stops, giving them the possibility to observe the number of passengers waiting at each stop. The simulation takes place for one day, where one minute corresponds to one tick in NetLogo.

1.2 The Problem

The objective is taking passengers to their desired destination as quick and cost efficient as possible. This gives rise to several problems, which will be attempted to solve in the coming weeks. First of all, a bus needs to determine its next destination. In deciding a next destination, the bus can either travel with a fixed schedule, or find some intelligent way to pick a next destination. Second of all, the bus needs to find the shortest path to its next destination. These two decisions can be interconnected, in the sense that the shortest path might not always be the desired one. If for example by taking a small detour the bus can pick up some more passengers that need to go to the same destination, this might make up for the extra travel time. The bus will probably not be able to take all passengers to their destinations on its own, so it needs to decide when to add new buses to the system and what type of buses to add. For this end it can make use of the total number of passengers waiting in the entire system and of its own capacity. If the bus is constantly leaving passengers on the side of the road, it might be efficient to add a new vehicle to the system.

1.3 A First Approach

As a first implementation, the system is designed without any form of intelligence or communication between agents. The buses all drive according to the same fixed schedule, where all bus stops get visited at least once. If a bus arrives at a stop, it first lets out the passengers that have this stop as

their destination, it then takes as many passengers as it can by checking its capacity and continues on its route. The system is initialized with one agent, and this agent is labeled the 'boss'. This means that this agent is in charge of adding new vehicles to the system if necessary. This bus adds new vehicles whenever there are more than sixty passengers waiting in the entire system and when the last time it added a new bus is at least sixty minutes (or ticks) ago. This is implemented in the NetLogo environment by giving the bus a local variable that counts down from sixty and gets reset to sixty whenever a new bus is added to the system. For now, the only bus type added is the largest bus, namely type 3. In future versions of the system, some smart way of deciding what bus type to add should be implemented. The route to drive is implemented in NetLogo as a simple list of destinations. Each bus owns a local variable that is incremented each time it visits a stop and that starts from zero when it finished the route.