Transportation System for Amsterdam

Team 47: Radu Sibechi, Nichita Diaconu, Bevilacqua André

April 2, 2018

ABSTRACT

In this project we implement an autonomous public transportation system that makes use of intelligent agents. Those agents have beliefs and intentions and seek to carry out their own tasks while maximizing their personal utility. As new agents are injected into the system, we discuss how they can leverage on communication protocols, coordination and competition strategies in order to improve the overall system efficiency.

1 Introduction

The aim of the project is to build a transportation system in which buses pick up and drop off passengers autonomously while maximizing the score obtained with regards to the performance criteria. This system can be seen as the combination of multiple problems from the fields of logistics and planning, where coordination plays an important role in order to ensure that the most convenient routes are being used in order to increase efficiency. To design an efficient system we had to tackle several issues, such as bottlenecks during busy times, which happens when an increasing amount of passengers are waiting to reach their destination.

In relation to the system we implemented, a set of performance criteria was given, which we always had in mind when designing our solution. These revolve around the minimization of:

- the passengers travel time, where travel time is defined as the difference in the number of ticks between the moments when a passenger has been picked up and delivered to its destination;
- the total number of messages sent during one full cycle of the simulation;
- the total costs. Money is spent on the lease cost of purchasing new vehicles; furthermore, bus spend money when traveling, as cost per patch.

The given map of Amsterdam is represented as an undirected, not fully connected graph with 24 vertices (stops) and edges between some of them, which indicate the permitted routes. In our approach, we decided to further divide the map in 4 adjacent cyclic sub-graphs (Fig. 1), where each sub-graph represents a bus' route and each node a bus stop. This way, a bus which is currently operating on a given line will be able to decide, thanks to the use of a routing table, if he wants to pick up passengers currently waiting at the stop or not based on their destination.

Among the main practical challenges, we first have to make sure that all passengers are actually delivered to their destinations, so that no penalties will be accrued for poor dispatchment of travelers. On the other hand, travel time can be controlled by ensuring that passengers are not dropped off to bus stops that are too far from their intended destinations. Furthermore, the costs of adding new agents has to be managed and when possible weighted against the current capacity and the number of waiting passengers.

Figure 1: Re-designed map of Amsterdam showing the four sub-graphs.

2 System architecture

In our implementation we decided to make use of fixed bus routes (Fig 1). We divided the original graph into 4 adjacent cyclic subgraphs (blue, red, green , black), where each sub-graph represents a bus route and the nodes in the sub-graph represent bus stops. It is possible to move from one bus route to another through one of the shared bus stops (orange circles in our figure: Evertsenstraat, Museumplein, Hendrikkade).

Each bus stop is assigned to one bus route. This works fine for all bus stops except for the shared ones, as they belong to multiple bus routes. We solved this issue by treating those as self-contained subgraphs. Table 1 shows our routing table, which is given to each bus; the columns are the destination of the passengers while the rows represent the location of a incoming bus (color of the bus route it is assigned to).

Source\Dest	blue	red	green	black	Es	Mp	Hk
blue	D	Es	Mp	Hk	D	D	D
red	Es	D	Mp	Mp	D	D	Mp
green	Mp	Mp	D	Mp	Mp	D	Mp
black	Hk	Hk	Hk	D	Hk	Hk	D

Table 1 Es = Evertenstraat, Mp = Museumplein, Hk = Hendrikkade, D = destination

2.1 Basic architecture of agents

2.1.1 Hybrid Agents

While traveling within the network, agents are designed so that they will follow a defined protocol of actions that they are meant to respect. In particular, when arriving at a bus stop, agents will:

- drop off passengers, based on their destination and the routing table;
- pick-up new passengers, based on their destination and the routing table;
- decide on next stop, based on what bus route it is assigned to.

Given their ability to perform actions as a result of external information, agents are designed to be **reactive**, as we found this to be the fastest way to translate environmental information into symbolic, while also minimizing the time taken to formulate an appropriate response of actions. Decisions upon acting or not are driven by logical rules in an effort to limit the number of bad decision taken by agents.

Agents are designed to be **deliberative** when they take into account their desires to decide which intentions to act on and when coordinating their actions with other agents (voting, bidding). Agents use practical reasoning to choose between competing reactions to the environment. The BDI framework that is described below filters different actions based on intentions, which are invoked according to an agent's desire. An agent then has a commitment to these actions and a plan is made to carry them out.

Deliberative behaviors include:

- setting desire to be benevolent or selfish;
- voting;
- bidding;
- deciding next stop (only applicable when a bus has to switch bus routes, we will go more in depth into this in the Bargaining section).

Belief, Desire, Intention framework

Beliefs These are observations regarding the environment, such as: the level of used capacity in a bus, destination of passengers that are on-board, adjacent bus stops, number of travelers waiting at bus stops.

Desires Each agent has two desires:

- reduce its load (self-interested);
- help others first (benevolent).

An agent is benevolent if less than 60% of its passenger capacity is occupied and self-interested otherwise.

Intentions

- vote
- bid
- pick up passengers
- drop off passengers
- choose next location selfishly (according to bus route)
- choose next location selflessly (if switching bus lines)
- kick off passengers early (if switching bus lines)

To accomplish desire (1), an agent will sequentially go through the bus stops on its bus route and drop off passengers. To accomplish desire (2), benevolent agents check for messages from "Turtle X" indicating that a bargain is in progress and will participate in order to reallocate tasks between themselves and help other agents in need. Agents use blind commitment with regard to carrying out intentions, as they drop commitments when the intentions are no longer achievable. Winning benevolent agents of the bargain will drop off their passengers early at the next bus station. Furthermore, they will abandon their commitment to pick up passengers at that stop and will select the next location selflessly, in order to reach the new bus lane.

2.2 Turtle X

We make use of a special bus called "Turtle X", which operates within the blue sub-graph in the map. This bus line is of particular importance given that it intersects with all other subgraphs and since it includes the "Central Station" stop from where new buses are injected into the system. "Turtle X" has special properties and acts as a global coordinator for the other agents. It plays a fundamental role in initiating voting, initiating bargaining, collecting messages coming from the network, evaluating them and translating them into new actions and/or decisions. Therefore the "Turtle X" will find its main use within the coordination, voting and bargaining protocols.

3 Coordination

3.1 Coalitions

Cooperation takes place where agents coordinate their actions to increase utility of all members of the coalition. Coalitions are stable, or have a non-empty core, when no member of any coalition has an incentive to break away from their coalition in order to join another. An agent has an incentive to break away from their coalition, if they would gain more utility by becoming a member of another coalition, and therefore is dependent both on the surplus created by the coalition and how this is divided amongst members.

In our implementation, coalitions are represented by dividing the map into different areas (bus routes), such that buses in each area to form a coalition, in which they do not move outside their territories. Benevolent agents have an incentive to break away from their coalition and join another one in order to both gain more utility and to also help other coalitions.

3.2 Communication

Messaging is the main tool used to equip agents with communication features and it has been implemented with the aim of improving the overall cost-efficiency of the network by helping controlling the total number of vehicles in operation at a given moment. Message understanding (ontology) is based on deciphering the content of the title of the message being sent and involves predefined agents variables.

The messages can take the following form:

- Turtle X to buses
  - ["bid_for_line" line_number] To ask a bus to bid for the passengers on a given line.
  - ["vote"] To ask a bus to vote.
  - ["bid_won_go_to" line_nr] To conform winner of auction and tell it its newly assigned line.
  - ["new_bus_instructions" new_bus_color] To tell a new bus what's its assigned color is.

- Buses to Turtle X
  - ["initiated" bus_type] To tell Turtle X that you have been newly initiated and you have type bus_type and that you need to be assigned a color.
  - ["positive vote"]

- ["negative vote"]
- ["bid" free_capacity] To place a bid with the free_capacity value

3.3 Group decisions

Voting is used for group decisions where agents may not want the same outcome. Agents try to take an action that leads to the best possible outcome for themselves. Agents have no reason to misrepresent their votes or vote strategically, as they do not have knowledge of how other agents might vote and votes are cast simultaneously.

At each tick, "Turtle X" checks how many waiting passengers are on a given line, defined as demand. If the demand on a line is more than 50% of the total line capacity, a vote is initiated in which all buses on that line participate. The voting procedure used is Plurality with two candidates (simple majority election):

- Yes, signifying that help is needed
- No, signifying that help is not needed

Each bus on the line has one vote, and votes deliberately based on their percentage of free capacity (social choice function). Since there are only two candidates, this voting scheme satisfies both the weak pareto efficiency and the monotonicity, which is desirable. Other reason for choosing this voting scheme include the speed with which it can be calculated and that it requires only one round of voting. "Turtle X" collects all the answers and acts upon the result. If the outcome is that no help is needed, nothing happens. If the outcome is that help is indeed needed, a bidding process is initiated as described in the section below.

3.4 Competition and allocation of tasks

The logic behind adding an auction scheme is that of having an embedded structure for task sharing. The role of auctions are that of allocating resources, which are scarce in nature, to the highest bidder, i.e. the agent that submits the highest valuation for the resource in question. Negotiation for task sharing is a good addition to a scheme were agents, who are able to self-evaluate the impacts of the actions they are carrying out, may find that re-distributing those tasks among themselves would increase the overall efficiency of the system rather than keep working on their own assignments. In short, redistribution of tasks is worked out by means of sending individual bids for a chance to obtain more work.

In our system an auction is initiated when the result of the voting initiated by "Turtle X" is positive. **Benevolent** agents have an incentive to send their bids to obtain more passengers therefore using up more of their available space while also helping lines in need. The type of auction that we have used is based on the First-Price sealed bid auction type with some modifications.

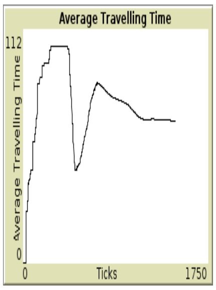
As mentioned above, a vote is initiated only if the demand on a line is more than 50% of the line capacity. A vote is successful if more than 50% of the buses voted yes, which means that more than 50% of the buses have less than 50% capacity. As such the free capacity on the line is less than 25% of the demand. We approximated the value to be at 25% and as such request additional back up for the rest of the 75% of the demand. Over the course of a single round, each agent will bid for more passengers by evaluating how much free space they currently have. The winners of the auction are the first N agents whose bids can handle together the 75% of the demand. If all the winners of the action cannot handle together this amount, then new buses are added.

The main issue with First-price sealed bids is that the best strategy for an agent is to bid less than its true valuation, which has no known solution. In our case we bid with the lower bid of the current free capacity in the bus, while our true value is the whole capacity (as passengers will be dropped before line rellocation). Moreover, this is also in accordance with the collective goal of wasting as little time of the passengers as possible. As a result, the top N buses with the highest free capacity are reassigned a new route.

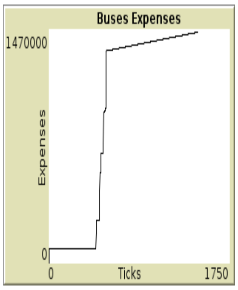
4 Results and Further work

The results of our system per days can be seen below

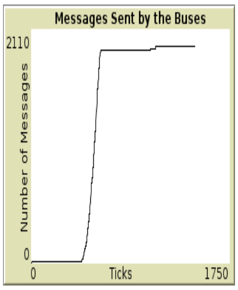
Day 3



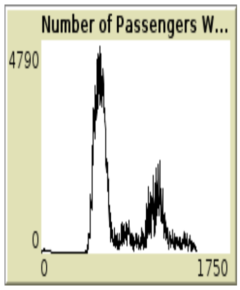
(a) Final average travel time: 69.8



(b) Buses expenses: 4.694.265

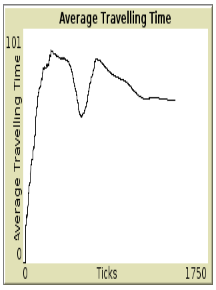


(c) Number of messages: 4329

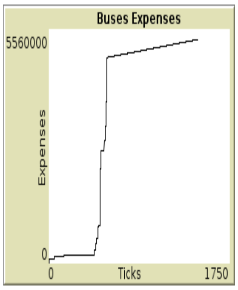


(d) Number of passengers waiting

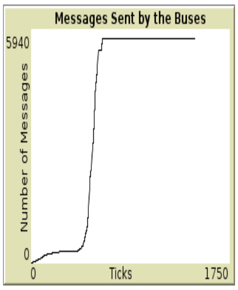
Day 2



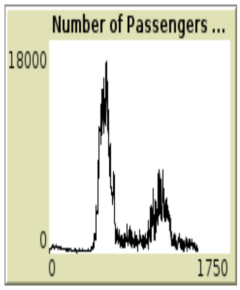
(a) Final average travel time: 72.858



(b) Buses expenses: 5.323.343

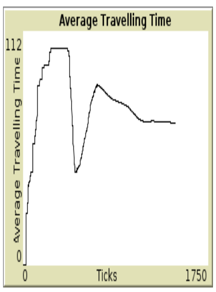


(c) Number of messages: 5695

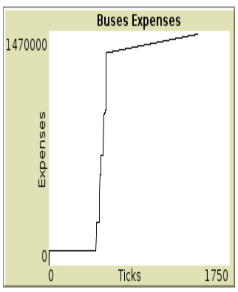


(d) Number of passengers waiting

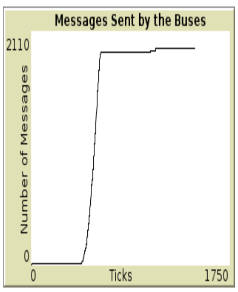
Day 1



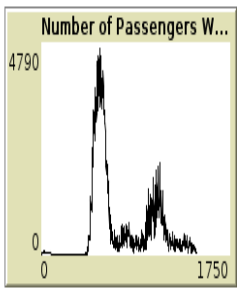
(a) Final average travel time: 71



(b) Buses expenses: 1.497.525



(c) Number of messages: 1951



(d) Number of passengers waiting

The efficiency of the solution would be strengthened by more empirical evidence regarding performance. This is particularly relevant in a number of areas:

- how full a bus should be in order to vote yes / no for adding a new bus

- what percentage of the line capacity should the demand be in order for "Turtle X" to trigger a vote for asking for help
- what percentage of the demand is required as help for a bus route

5 Bibliography

An Introduction to Multi-agent systems, Michael Wooldridge 2009:

- section 2.1 and 2.2: Ch 1 - 5, Agents, Intelligence, Reasoning
- section 3.1 and 3.2: Ch. 6 - 7, Multi-agent interactions and agreement
- section 3.3: Ch. 8 - 12, Strategy functions, Plurality voting
- section 3.4: Ch. 14, First-price sealed-bid auctions