

Multi-Agent systems - Assignment 3 (Communication)

Emma Machielse, Federico Tavella, Alessandro Tezza

February 24, 2018

1 Overview

During the previous week, we implemented the first version of the transportation system, in which the agents were moving according to a fixed schedule. The firstly created agent was responsible of creating new buses based on the amount of people waiting at the bus stops and the total capacity of the bus fleet. This week, we focused on further improve the first version and on implementing a communication system between agents.

2 Improvements

- not all the buses are following the same fixed schedule. The buses created with an even ID are moving through the schedule that we defined the previous week, whereas the bus created with an odd ID are moving through the **reverse version** of the same schedule;
- every bus picks up **only the passengers for which it is better to move in that specific bus direction**. If this is not the case, the passengers will wait for a bus moving in the opposite direction. In this way, we try to minimize the travelling time of the passengers. The waiting time is not so affected by this choice, thanks to the even distribution of the buses in the two directions;
- we implemented the possibility for a bus of **stopping and waiting**. This is useful when the waiting passengers are not many and the situation is managed by other buses, because we can decrease the travelling costs;
- we implemented some **roles**: every bus can have one or more roles. The roles define the **behaviour** of a bus. In this way, we can easily change the behaviour of a bus by changing its role. At the moment, we defined the following roles:
 - **Coordinator**: a special bus that is allowed to create new buses. At the moment, only the first created bus is a coordinator.
 - **Scout**: a bus that is always moving through the schedule. It is not allowed to stop and wait.
 - **Simple**: a bus with no special aims.

3 Communication

The communication is implemented as follows. Every time a bus reaches a bus stop, it checks the incoming messages. It can retrieve the new messages from the inbox history comparing the ticks of the messages with a **locally stored value** that represents the tick in which the agent checked the messages the last time. The content of a message is a string that represents an action to do or an information request. At the moment, the *coordinator* (that locally keeps track of the created buses and their capacity) can send a “promotion” message to the other buses: every *simple* bus that receive the message will become a *scout*. In this way, the *coordinator* can check for problematic situation using its own local information and manage it ordering to the other buses to never stop (i.e., to be *scouts*). When the danger situation will be managed, the *coordinator* can send other messages to remove the promotion of the buses: then, they will be allowed to stop and wait (i.e., the role will change from *scout* to *simple*).