A Framework for a Multimodal Transportation Network: an Agent-Based Model Approach

Nuno Monteiroa, Rosaldo Rossettib, Pedro Camposa, Zafeiris Kokkinogenisb

aPorto University, Economics Department, Porto, Portugal
bPorto University, Engineering Department, Porto, Portugal

Abstract

Mobility and commuting in metropolitan areas are very expensive, highly polluted and time wasting. The Four Step Model (FSM) is the key model to analyze a Transportation Network. However, being the FSM a combination of several models, combining them in one model have rarely been applied. To deal with this problem an Agent-Based Model (ABM) is proposed. An ABM uses the metaphor of autonomous agents and so, they can be a handful tool for combining different models in one. Therefore, this model can be used as a tool for simulation and integrate the FSM in one model. Here we present the preliminary results of this approach.

© 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/).
Selection and peer-review under responsibility of Technische Universität München

Keywords: Multimodal Transportation; Multimodal Network Design Problem; Agent-Based Models; Intelligent Transportation Systems; Four Step Model; Transport Analysis

1. Introduction

There are different motivations behind this work. The first one is the application of Agent-Base models, a nascent type of models in urban studies. The second is that research in Intelligent Transportation System and Future Cities has been increasing in the last years.

Our research problem is that commuting in metropolitan areas is very expensive, highly polluted, time wasting and there is no real car alternative. It is expensive and polluted due to the large sparse cities that make the commuting longer and expensive. There is no real car alternative because public transportation lack in quality and time management to reach all people needs.

* E-mail address: 120414020@fep.up.pt
To analyze the transportation network we adopted the Four Step Model (FSM), described in McNally (2000). The FSM is a framework model developed that functions like an iteration model with four steps Trip Generation, Trip Distribution, Mode Choice and Route Choice.

To solve the research problem we propose a simulation model of the FSM using an Agent Based Model (ABM). Although combined models integrating all of the four stages were developed, they rarely applied in practice. Therefore, this model act as a tool for simulation and prediction interactions between infrastructures changes, public transportation investments, and endogenous traffic effects in a daily basis.

We aim at creating dynamic Origin-Destination Matrices (O-D matrices) and Multi-Modal Transport Network (MMTN). Studying dynamic O-D matrices can provide the FSM with real-time demand that reflects the actual traffic situation and MMTN combines several transportation methods.

To solve the problem we divided the methodological approach in three phases. First, we develop a methodological framework for ABMs regarding the FSM and the MMTN. This framework is built with the ODD protocol (Overview, Design concepts and Details protocol) in Grimm (2006). The ODD is a generic format and a standard structure by which all ABMs. In the second phase, one must combine the different models used to analyze a transportation and expand the network. Finally, we analyze and calibrate the model that must be simple to understand so that it can be useful for decision process.

2. Agent-Based Models

Models are simplifications of reality. They are theoretical abstractions that represent systems. Batty (2009) identify and highlight essential features crucial to move from theory and to applications.

Agent-Based Simulation uses the metaphor of autonomous agents and Multi-Agent Systems as the basic model conceptualization. They are an evolution of Cellular Automata (CA). This means that a model consists of interacting agents situated in a simulated environment thus and agents may correspond to cities, blocks, platoons, households, individual travelers (drivers), vehicles, sensors, traffic signals, etc. In addition, elements of the environment may be agents as highlighted in Portugali (2000).

One reason for the popularity of agents and Multi-Agent Systems are the advances in computers, which are more distributed, open, large, and heterogeneous, Bazzan (2013). Managing interactions among autonomous entities with increasing interdependencies has been one of the biggest motivations for distributed artificial intelligence and for MAS.

Traffic simulation represents a prominent application for modelling and simulation in Bazzan (2013). It supports complex urban and transport planning but their utility depends on adequate calibration, verification, and validation.

Calibration provides values for unknown parameters. Verification and validation means the correctness of model construction and the truthfulness of a model with respect to its problem domain, respectively. In other words, “verification means building the system right, and validation means building the right system” in Parker (2003).

These three aspects motivated the ODD protocol, Grimm (2006). The primary purpose of ODD is to make writing and reading model descriptions easier and more efficient.

At table one, we present the seven elements of the original and updated ODD protocol, as seen in Grimm (2010).

Overview

The first phase, Purpose, defines that every model has to start from a clear question, problem, or hypothesis.

The next one is the Entities, State Variables, and Scales. An entity is a distinct object that behaves as a unit, and defines a set of attributes that can contain numerical or behavioral strategies, Huse, et al. (2002). A state variable traces how the entity changes over time. Scales describes a spatial or temporal variable that explains the amount of space and time represented in the simulation.

The Process Overview and Scheduling defines the order and names of the model’s processes. This is relevant due to different scheduling process can have a very large effect in the model outputs, Bigbee, et al. (2006) and Caron-Lormier (2008).
### Table 1. The Seven Elements of the original and updated ODD protocol in Grimm (2010)

<table>
<thead>
<tr>
<th>Overview</th>
<th>Elements of the original ODD protocol</th>
<th>Elements of the updated ODD protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Purpose</td>
<td>1. Purpose</td>
<td>1. Purpose</td>
</tr>
<tr>
<td>2. State variables and scales</td>
<td>2. Entities, state variables, and scales</td>
<td></td>
</tr>
<tr>
<td>4. Design concepts</td>
<td>Basic principles</td>
<td></td>
</tr>
<tr>
<td>Emergence</td>
<td>Emergence</td>
<td></td>
</tr>
<tr>
<td>Adaptation</td>
<td>Adaptation</td>
<td></td>
</tr>
<tr>
<td>Fitness</td>
<td>Objectives</td>
<td></td>
</tr>
<tr>
<td>Prediction</td>
<td>Learning</td>
<td></td>
</tr>
<tr>
<td>Sensing</td>
<td>Sensing</td>
<td></td>
</tr>
<tr>
<td>Interaction</td>
<td>Interaction</td>
<td></td>
</tr>
<tr>
<td>Stochasticity</td>
<td>Stochasticity</td>
<td></td>
</tr>
<tr>
<td>Collectives</td>
<td>Collectives</td>
<td></td>
</tr>
<tr>
<td>Observation</td>
<td>Observation</td>
<td></td>
</tr>
<tr>
<td>5. Initialization</td>
<td>5. Initialization</td>
<td></td>
</tr>
<tr>
<td>6. Input</td>
<td>6. Input data</td>
<td></td>
</tr>
<tr>
<td>7. Submodels</td>
<td>7. Submodel</td>
<td></td>
</tr>
</tbody>
</table>

### Design Concepts

The Design Concepts may be crucial to interpreting the output of a model. Therefore, they are included in ODD to make sure that important model design decisions made and that a reader is aware of those decisions, Railsback (2001).

The Basic Principles are defined as the general concepts, theories, hypotheses, or modeling approaches that are under the model.

The Emergence is what varies in complex of individuals or their environment change. The Adaptation rules what kind of decisions the agents must have in response to changes in their environment. Objectives defines the agents' success criteria before the model itself. Learning many agents change their trait over time as consequence of their experience, so the way but be explicit. The Prediction is how the agents can predict the future experiences is they learn new things in the present. Sensing is what the state variables can feel and with the new information, how they can communicate it to other agents. Interaction is what the agents encounter and affect other agents, and how they can deal with those encounters. Stochasticity defines the processes calculated in a random way. If the individuals' agents can form aggregations or form Collectives, they must be well explained and represented. For last, the Observation is what data we will use to perform the model.

### Details

The initialization defines the initial state of the model as well as the initial parameters. To define the input data, one must describe the source and the some characteristics.

The last part of the ODD description is the Submodels. Agent-based modeling is new and lacks a firm foundation of theory and established methods, the ODD protocol reinforces that descriptions must include appropriate levels of explanation and justification for the design decisions they illustrate.
3. The Four Step Model and Mathematical Implementation

3.1. Four Step Model

The FSM is a framework model developed that functions like an iteration model with four steps, as seen in McNally (2000) and Ortuzar (2001).

1. Trip Generation, where total daily travel is loaded in the model system;
2. Trip Distribution, where a destination choice model is made that generates a trip matrix;
3. Mode Choice factors the trip tables to produce mode trip tables and
4. Route Choice, allocates trips between an origin and destination by a particular mode to a route.

Fig. 1. The Four Step Model in McNally (2000)

Trip Generation

The first stage of the FSM is the total daily travel in the model system, at the household and at zonal level, for various trip purposes. The first stage also deals with the transformation of activity-based to trip-based, and simultaneously divides each trip into a production and an attraction, to prevent network performance measures from influencing the frequency of travel.

The models that define this separation estimates the productions $P^p_i$ and attractions $A^p_j$ for each trip type (purpose) $p$:

$$\begin{align*}
P^p_i (A) &= f^p_i (\text{activity system characteristics}) \\
A^p_j (A) &= f^p_j (\text{activity system characteristics})
\end{align*}$$

Where $P^p_i$ is the total trip productions generated for trip type $p$ for analysis unit $i$ and $A^p_j$ is the total trip attractions for trip type $p$ for analysis unit $j$. Essentially, trips can be modeled at different levels being the zonal, household, or person level the most common for trip attractions.

Trip Distribution

The trip distribution model is essentially a destination choice model and generates a trip matrix, represented at figure 2. The notation used for each trip purpose utilized in the trip generation model as a function of activity system attributes is $T_{ij}$, through the generated productions $P_i$ and attractions $A_j$, and network attributes.

The general form of the trip distribution model as the second step of the FSM is the gravity model:

$$T_{ij} = a_i, b_j P_i, A_j, f(t_{ij})$$

$$\begin{align*}
a_i &= \left[\sum_j b_j P_i, A_j, f(t_{ij})\right]^{-1} \\
b_i &= \left[\sum_j a_i P_i, f(t_{ij})\right]^{-1}
\end{align*}$$
The parameter $f(t_{ij})$ represents the function of the network level of service.

Table 2. O-D matrices notations in McNally (2000)

<table>
<thead>
<tr>
<th>Zones</th>
<th>1</th>
<th>2</th>
<th>...</th>
<th>$j$</th>
<th>...</th>
<th>$n$</th>
<th>Productions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$T_{11}$</td>
<td>$T_{12}$</td>
<td>...</td>
<td>$T_{1j}$</td>
<td>...</td>
<td>$T_{1n}$</td>
<td>$P_1$</td>
</tr>
<tr>
<td>2</td>
<td>$T_{21}$</td>
<td>$T_{22}$</td>
<td>...</td>
<td>$T_{2j}$</td>
<td>...</td>
<td>$T_{2n}$</td>
<td>$P_2$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$i$</td>
<td>$T_{i1}$</td>
<td>$T_{i2}$</td>
<td>...</td>
<td>$T_{ij}$</td>
<td>...</td>
<td>$T_{in}$</td>
<td>$P_i$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$n$</td>
<td>$T_{n1}$</td>
<td>$T_{n2}$</td>
<td>...</td>
<td>$T_{nj}$</td>
<td>...</td>
<td>$T_{nn}$</td>
<td>$P_n$</td>
</tr>
<tr>
<td>Attractions</td>
<td>$A_1$</td>
<td>$A_2$</td>
<td>...</td>
<td>$A_j$</td>
<td>...</td>
<td>$A_n$</td>
<td>$T$</td>
</tr>
</tbody>
</table>

Fig. 2. Framework for the DTA (Dynamic Traffic Assignment) models in Mahmassani, et al. (2007)

The fig 2. shows a framework for the DTA (Dynamic Traffic Assignment) models. (Mahmassani, et al., 2007). In this framework, the main base for accurate and robust demand estimation and prediction for real-time Dynamic Traffic Assignment (DTA) works upon three main aspects. It incorporates regular demand information into the real-time demand prediction process. It recognizes and stores possible structural changes in demand patterns under several different conditions. For last, it updates the a priori estimate to the regular pattern using new real-time estimation results and traffic observations, Mahmassani, et al. (2007) and Ben-Akiva (2000). Therefore, the main base follow as equation 6.

$$\text{true demand} = \text{regular pattern} + \text{structural deviations} + \text{random fluctuations}$$

**Mode Choice**

Mode choice factors the trip tables from trip distribution to produce mode-specific trip tables. If the proportion of trips by other modes is small, it utilizes a simplified person trip tables to allow the development of vehicle trip tables. Therefore, ignores trips by other modes.

The road network combines several modes of transportation. For that reason multimodal transportation problem appeared. Travelers need improved means to access information on alternative transport modes and to solve problems affecting their journeys.
Route Choice

Route choice is an equilibration of demand and performance. Modal O-D trip matrices are loaded on the modal networks usually under the assumption of user equilibrium where all paths utilized for a given O-D pair have equal measures.

Several works use a market metaphor to find the solution for this problem. Some works use a reservation-based approach to cross networks of intersections, Vasirani (2009) and, Vasirani (2011). Others a market method for optimal single intersections, Schepperle (2007) and Schepperle (2009) or based on intersections but in this case based on the traffic signals, Balan (2006).

3.2. Mathematical Implementation for Network Representation

3.2.1. Network Representation

The mathematical definition of a network is a set of nodes, vertices or points and a set of links connecting those nodes, in Sheffi (1985).

Figure 3 shows a network including five nodes connected by 11 links. Each link in this network is associated with a direction of flow. For example, link 11 represents flow from node 3 to node 2, while link 10 represents the reverse flow, 2 to 3.

The transportation planning process for urban areas uses a partition of an area into traffic zones. A node represents each traffic zone. After, the desired movements over an urban network expresses in terms of an Origin-Destination matrix.

Travel time on urban context is an increasing function of flow. Each network link is typically associated with some impedance. The delay of a travelling vehicle is null when the impedance is also null. As the flow increases, the travel time increases since the number of cars along the link increases, in Sheffi (1985).

A stable condition reached only when no traveler can improve his travel time by unilaterally changing routes. This is the characterization of the user-equilibrium (UE) condition, in Beckmann (1956).

The approach for solving large problems uses the equivalent minimization method. The solutions bases on the behavioral assumption that each motorist travels on the path that minimizes the travel time t from origin to destination.

3.2.2. Network functions

Each O-D pair \( r - s \) is connected by a set of paths (routes) through the network \( \mathcal{K}_{rs} \) where \( r \in \mathcal{R} \) and \( s \in \mathcal{P} \), so the O-D matrix is denoted by \( q \) with \( q_{rs} \).

Let \( x \) and \( t \) represent the flow and travel time, respectively, on link \( a \) (where \( a \in \mathcal{A} \)). Therefore, \( t_a(x_a) \) is the link performance function. Let \( f_k^{rs} \) and \( e_k^{rs} \) represent the flow and travel time, respectively, on path \( k \) connecting origin \( r \) and destination \( s \) (\( k \in \mathcal{K}_{rs} \)).
\[ c^r_k = \sum_a t_a \cdot \delta^r_{a,k} \quad \forall k \in K_{rs}, \quad \forall r \in R, \quad \forall s \in A \] (7)

Where \( \delta^r_{a,k} = 1 \) if link \( a \) is a part of path \( k \) connecting O-D pair \( r-s \), and \( \delta^r_{a,k} = 0 \) otherwise. Link flow expresses as follows.

\[ x_a = \sum_r \sum_s \sum_k f^r_k \cdot \delta^r_{a,k} \quad \forall a \in A \] (8)

Equations 7 and 8 defines the path-arc incidence relationships.

The equilibrium assignment problem is to find the link flows \( x_a \), that satisfy the user-equilibrium criterion when all the Origin-Destination entries \( q_{rs} \), have been appropriately assigned. Solving the following mathematical program obtains link-flow pattern:

\[ \min z(x) = \sum_a f^x_a \int_0^{t_a} t_a(\omega)d\omega \] (9)

Subject to:

\[ \sum_k f^r_k = q_{rs} \quad \forall r, s \] (10)

\[ f^r_k \geq 0 \quad \forall k, r, s \] (11)

The definitional constraints are also part of the program.

\[ x_a = \sum_r \sum_s \sum_k f^r_k \cdot \delta^r_{a,k} \quad \forall a \in A \] (12)

Equation 10 represents a set of flow conservation constraints that the flow on all paths connecting each O-D pair has to equal the O-D trip rate and equation 11 is required to ensure that the solution of the program will be physically meaningful with no negativity path flow.

The link relationship with the capacity and the volume expresses in a function called the BPR function (Bureau of Public Roads (1964)). This function works as follows

\[ S_a(v_a) = t^0_a \left[ 1 + \alpha \left( \frac{v_a}{c'_a} \right)^\beta \right] \] (13)

At equation 13, \( S_a(v_a) \) is the average travel time for a vehicle on link \( a \), \( t^0_a \) is the free-flow time, and \( c'_a \) is the practical capacity of the link \( a \). This practical capacity means that the links never reach their maximum capacity, but rather they have a maximum possible flow through.

4. Framework for Agent-Based Intelligent Transportation Model

A project for methodological framework for an ABM in Transportation Analysis is given. A framework is a real or conceptual structure intended to serve as a support or guide for the building of something. The name is Agent Based Intelligent Transportation Model (ABITM) and it consists in the following major steps.

4.1. The Structure

The structure must follow the updated version of the ODD protocol. This acts as an underlying code structure being easier for new models to be built-in or to expand the current model.

4.2. Tools

This model works under the NetLogo environment.
4.3. The General Framework

Fig. 4. The ABITM methodology (Own Elaboration)

The major framework is the work of Manheim/Florian Transportation System Analysis Framework. The combined ABTIM methodology appear in figure 4. This approach combine the different transportation models used to analyze the transportation metropolitan transportation system. Therefore, this model act as a tool for simulation and prediction interactions between infrastructures changes, public transportation investments, and endogenous traffic effects in a daily basis.

4.4. Models

In this Framework, a combination of transportation models is proposed. The one underlying the model is the FSM as is highlighted in the literature review. The others can be seen detailed in table 3.

Table 3. ABITM Models (Own elaboration)

<table>
<thead>
<tr>
<th>Variables</th>
<th>Control / Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: Activity System</td>
<td>Defines the Activity System within the city.</td>
</tr>
<tr>
<td>L: Location Procedure</td>
<td>Location Procedure gives the location and the situation of each agent travelling in the network.</td>
</tr>
<tr>
<td>T: Transportation System</td>
<td>The Transportation System is the network the model will work on.</td>
</tr>
<tr>
<td>S: Supply Procedure</td>
<td>The Supply Procedure is the characteristics of that Transportation System.</td>
</tr>
<tr>
<td>F: Network Flow</td>
<td>The Network Flow is the output of the simulation Here we can extract information to adjust the model.</td>
</tr>
<tr>
<td>TG: Trip Generation</td>
<td>The Trip Generation will decompose the Activity System in trips that will be assignment in the next step.</td>
</tr>
<tr>
<td>TD: Trip Distribution</td>
<td>Here all the trips generated in the last will by distributed in all the nodes of the network.</td>
</tr>
<tr>
<td>MC: Mode Choice</td>
<td>The Mode Choice will analyze for each agent is preference and assign a transportation mode.</td>
</tr>
<tr>
<td>RC: Route Choice</td>
<td>The Route Choice will give the agent a route / a path to follow and the agent will go under that path.</td>
</tr>
</tbody>
</table>
4.5. Flow and Techniques

The first step is the input data. Depending on the simulation, the model can be loaded with real databases, to create the network and the population that will generate the trips. Moreover, other input data can be loaded, for instance costs, pollution.

The location procedure is the activity system input. As said before the activity system is everything else except the transportation system (e.g. weather, pollution...). The location procedure is the output of the network flow, which is the last step of the model iteration. This tries to simulate changes in people location or options done due to the network state. The transportation system works in a similar way of the activity system.

The four functions must work in a sequential way. To reflect the reality in each step an equilibration must be achieve in order to the simulation go another step. The agents obtain an equilibrium when the agents do not need to change their present status, meaning, their utility reaches an optimum. If an agent need to change the TD, the TG is already in equilibrium but the MC and the RC need a new equilibrium.

The output data is going to reflect, for each iteration, the present network state. Data extracted is in accordance with the input data. For example, in using this model to predict changes in pollution with an investment in a new type of bus, the output must be the updated input data.

The feedback parameters A, B, C and D serves as an input for the TG, TD, MC and RC. Those parameters work with another output from the Network Flow state of the last iteration. This works as a day-to-day updating and a historical dynamic demand. The Feedback parameters work in updating the FSM micro-models with past information from the Network Flows.

5. Implementation

5.1. Model description

We present our agent-based model in accordance with the ODD (Overview, Design concepts, and Details) protocol.

5.1.1. Purpose

A simulation model of the FSM using an ABM is proposed. In this implementation, we developed a daily dynamics (24-hour).

5.1.2. Entities, state variables and scales

In this implementation, the agents are drivers. The state variables that characterizes the drivers are to-node-car (the node each agent is going to), from-node-car (the node each agent comes from), current-node (the node the agent is), who (which identifies the agent number), drivers_ratio (splits the agents strategies), velocity_delta (random number to increase speed) and v (agents velocity). Each agent can have two strategies: the quickest way (yellow_drivers) or the fastest way (red_drivers).

The model is a 30 x 20 patches world in which each patch representing 20 pixels. Each tick represents one-minute simulation and so 1440 ticks represents a 24-hour process. Each patch represents 1000 meters.

5.1.3. Process overview and scheduling

This model works on a sequential way. According to the FSM, an agent must follow the four steps (Trip Generation -> Trip Distribution -> Mode Choice -> Route Choice) in a sequential way. First the agents' travels are loaded and distributed in origin nodes (for simplicity, only one origin is considered). After the agents choose their travel mode (for simplicity, only one mode is considered). The last step the agent follow a route / path in the network until reach the destination. During this process, the agents' can change their behavior according to the network conditions.

The model procedure scheduling has seven steps. The first one willing-to-travel, setups the agents' will to travel during a 24-hour process. Here, we defined the 24-hour procedure with a Poisson distribution. It expresses the probability of a given number of events occurring in a fixed interval of time. The parameters for the Poisson distribution changes over time and those, by definition, predicts the degree of spread around a known average rate of
occurrence. In the model, the parameter `inter-arrival-time` defines the parameter for the Poisson distribution. In the input data section it shows how the parameter was setup.

The Reproduce procedure ensures that new agents' have the same drivers' characteristics. Count-turtles-on-links is a procedure that counts all the agents' in each link. The next procedure, `create_link_volumes`, uses the last procedure to update the behavior of each agent. At this procedure, drivers can increase or decrease their speed according to the present network status.

The last procedure is just to ensure the report of the main important variables. Here the network calculates the total drivers (`calculate_total_drivers`) loaded in the system, the time to travel (`calculate_time_travel_drivers`) for all the agents and for last the average travel time (`calculate_average_travel_time`) for each agent.

5.1.4. Design concepts

a) Emergence
   The main output is the average speed and the agents' travel time.

b) Adaptation
   This model does not use adaptation.

c) Objectives
   Agents' objectives are different, depending on `red_drivers` or `yellow_drivers`. Yellow drivers tries to achieve the shortest path and the `red_drivers` the fastest one.

d) Learning
   This model does not use learning. In the future, learning will be in the model to simulate a day-to-day dynamics. Then the agents can adapt they routes to their experiences and create true Dynamic O-D matrices.

e) Prediction
   This model does not use prediction.

f) Sensing
   Agents' perceive their status and the agents around them, so agents can adapt their velocity to the current network condition.

g) Interaction
   This model does not use interaction. In the future interaction will be in the model, so agents can have vehicle-to-vehicle communication to predict the present road condition. For now, the agents can only see the condition in the next node but, with interaction, they can receive information from all network to they can change their present behavior.

h) Stochasticity
   Several stochastic processes are used in the model. As said before, this model demonstrates the potential of a future implementation. Because this model is not a real problem the type (`drivers_ratio`), number (`num-drivers`), behavior (`drivers_ratio`) and time (`inter-arrival-time`) are modelled using a stochastic model. The `drivers_ratio` parameter divides the agent creation between `yellow_agents` and `red_agents`.

i) Observation
   This simulation produces several results. The time to travel, the number of agents created and the average travel time are one of possible output. In the simulation, several variables come out like, links volume, the average speed and the agents' type.

5.1.5. Initialization

Seven nodes plus one origin and one destination defines this network. Each nodes creates links to that connects to other nodes. For a question of simplicity we just consider that each node only have one directed link to other node. A total of ten links and seven nodes represents this network, as the figure 3 shows.
With this links and nodes, four paths connects the network. We divided the links in two, CCL (City Center Links) and HL (Highway Links). The CCL includes the \text{link}_1 \text{ until } \text{link}_5. \text{The HL comprises the link}_6 \text{ to links}_10.

1. First path - (\text{link}_6 \to \text{link}_7) \text{ and } (\text{origin} \to \text{node} 4 \to \text{destination}) \to \text{length } 25.61 \text{ patches}
2. Second path - (\text{link}_1 \to \text{link}_2 \to \text{link}_3 \to \text{link}_4) \text{ and } (\text{origin} \to \text{node}1 \to \text{node} 2 \to \text{node} 3 \to \text{destination}) \to \text{length } 20 \text{ patches}
3. Third path - (\text{link}_1 \to \text{link}_2 \to \text{link}_5 \to \text{link}_7) \text{ and } (\text{origin} \to \text{node}1 \to \text{node}2 \to \text{node} 4 \to \text{destination}) \to \text{length } 30.81 \text{ patches}
4. Fourth path - (\text{link}_8 \to \text{link}_9 \to \text{link}_10) \text{ and } (\text{origin} \to \text{node}5 \to \text{node}6 \to \text{destination}) \to \text{length } 23.04 \text{ patches}

With the setup button the model automatically setups several variables. \text{velocity}_\text{delta} = 0.10, \text{drivers}_\text{ratio} = 0.5, \text{inter-arrival-time} = 0.10 \text{ and } \text{num-drivers} = 1.0.

5.1.6. Input data

The \text{inter-arrival-time} is the parameter for the Poisson distribution as said before. Since we do not have data, the parameters for the Poisson distribution work as follows in table 4. This way we intend to represent the morning and afternoon high demand.

<table>
<thead>
<tr>
<th>Hour</th>
<th>Ticks</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:00 - 06:59</td>
<td>0 - 419</td>
<td>10</td>
</tr>
<tr>
<td>07:00 - 10:59</td>
<td>420 - 559</td>
<td>2</td>
</tr>
<tr>
<td>11:00 - 14:59</td>
<td>560 - 899</td>
<td>10</td>
</tr>
<tr>
<td>15:00 - 17:59</td>
<td>900 - 1079</td>
<td>10</td>
</tr>
<tr>
<td>18:00 - 20:59</td>
<td>1080 - 1259</td>
<td>2</td>
</tr>
<tr>
<td>21:00 - 23:59</td>
<td>1260 - 1440</td>
<td>10</td>
</tr>
</tbody>
</table>

The agent velocity was develop based in the BPR function. This means that the average travel time for a vehicle on link \( a \), is going to be a function of their status. Each agent have their velocity adjusted to link volumes, as table 5 shows.
The maximum capacity for the CCL is 20 and for the HL is 25 drivers. To explain the table 4, let us imagine an agent is driver is in a CCL. In the next link, there are four cars. The velocity the agent will adapt is 0.6, which corresponds to 1,667 minutes each 1000m. Therefore, the agent travels at a velocity between 36km/h to 42km/h. To give randomness to the agent velocity, each agent is set with a velocity delta, which is a sum to their velocity of 0.1 or 0.2 depending on link type.

5.1.7. Submodels

This model has two submodels. One is the model that updates the velocity of each driver. It relates the maximum capacity for each link with the current-flow. The other submodel is the route/path choice. It works as follows: an agent when reaches a node calculates the distance to the next node or the time to the next node. If the node is the destination, the agent exists the simulation, if not it will travel along the link until the next node.

This implementation tries to analyze the framework and model foundations. With that, we run three basic experiments to analyze the model behavior and if the agents behave as expected.

5.2. First Experiment

In the first experiment the standard parameters of the model configuration is used. The standard parameters are drivers_ratio = 0.5, velocity_delta = 0.10, num-drivers = 1.0.

### Table 5. Relationship between ticks/time (Own production)

<table>
<thead>
<tr>
<th>Volume</th>
<th>V</th>
<th>VD</th>
<th>Vel Range</th>
<th>Volume</th>
<th>V</th>
<th>VD</th>
<th>Vel Range</th>
<th>Volume</th>
<th>V</th>
<th>VD</th>
<th>Vel Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>0</td>
<td>0.1</td>
<td>0km/h - 6km/h</td>
<td>12</td>
<td>0.5</td>
<td>0.1</td>
<td>30km/h - 36km/h</td>
<td>25</td>
<td>0.2</td>
<td>0.2</td>
<td>12km/h - 24km/h</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>0.1</td>
<td>0km/h - 6km/h</td>
<td>11</td>
<td>0.6</td>
<td>0.1</td>
<td>36km/h - 42km/h</td>
<td>24</td>
<td>0.2</td>
<td>0.2</td>
<td>12km/h - 24km/h</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>0.1</td>
<td>0km/h - 6km/h</td>
<td>10</td>
<td>0.7</td>
<td>0.1</td>
<td>42km/h - 48km/h</td>
<td>23</td>
<td>0.2</td>
<td>0.2</td>
<td>18km/h - 24km/h</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0.1</td>
<td>0km/h - 6km/h</td>
<td>9</td>
<td>0.8</td>
<td>0.1</td>
<td>30km/h - 36km/h</td>
<td>22</td>
<td>0.3</td>
<td>0.2</td>
<td>18km/h - 34km/h</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0.1</td>
<td>0km/h - 6km/h</td>
<td>8</td>
<td>0.9</td>
<td>0.1</td>
<td>24km/h - 30km/h</td>
<td>21</td>
<td>0.3</td>
<td>0.2</td>
<td>18km/h - 30km/h</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0.1</td>
<td>6km/h - 12km/h</td>
<td>7</td>
<td>0.4</td>
<td>0.1</td>
<td>30km/h - 36km/h</td>
<td>20</td>
<td>0.3</td>
<td>0.2</td>
<td>18km/h - 30km/h</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0.1</td>
<td>6km/h - 12km/h</td>
<td>6</td>
<td>0.5</td>
<td>0.1</td>
<td>36km/h - 42km/h</td>
<td>19</td>
<td>0.4</td>
<td>0.2</td>
<td>24km/h - 36km/h</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0.1</td>
<td>18km/h - 24km/h</td>
<td>5</td>
<td>0.6</td>
<td>0.1</td>
<td>36km/h - 42km/h</td>
<td>18</td>
<td>0.4</td>
<td>0.2</td>
<td>24km/h - 36km/h</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0.1</td>
<td>18km/h - 24km/h</td>
<td>4</td>
<td>0.7</td>
<td>0.1</td>
<td>42km/h - 48km/h</td>
<td>17</td>
<td>0.4</td>
<td>0.2</td>
<td>24km/h - 36km/h</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0.1</td>
<td>18km/h - 24km/h</td>
<td>3</td>
<td>0.8</td>
<td>0.1</td>
<td>30km/h - 36km/h</td>
<td>16</td>
<td>0.5</td>
<td>0.2</td>
<td>30km/h - 42km/h</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0.1</td>
<td>18km/h - 24km/h</td>
<td>2</td>
<td>0.9</td>
<td>0.1</td>
<td>24km/h - 30km/h</td>
<td>15</td>
<td>0.5</td>
<td>0.2</td>
<td>30km/h - 42km/h</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0.1</td>
<td>18km/h - 24km/h</td>
<td>1</td>
<td>0.4</td>
<td>0.1</td>
<td>24km/h - 30km/h</td>
<td>14</td>
<td>0.5</td>
<td>0.2</td>
<td>30km/h - 42km/h</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0.1</td>
<td>18km/h - 24km/h</td>
<td>0</td>
<td>0.5</td>
<td>0.1</td>
<td>30km/h - 42km/h</td>
<td>13</td>
<td>0.5</td>
<td>0.2</td>
<td>30km/h - 42km/h</td>
</tr>
</tbody>
</table>

![Fig. 6. NetLogo Output - Average Velocity vs Total Drivers for experiment one (Own production)](image)
For 197 drivers, 97 red_drivers and 100 yellow_drivers, a total travel of 6553 ticks with a 33.27 minutes average travel time was record. In the process the figure 6, shows an inverse relationship between the average velocity and the current drivers in the network (The red line (current drivers) divides by 10, which permits a better scale for analyses).

5.3. Second Experiment

In this second experiment, we will run only yellow_drivers and everything else constant. Yellow_drivers search only for shortest path, and in the case they will use of the second and third path. The parameters are drivers_ratio = 1.0, velocity_delta = 0.10, num-drivers = 1.0.

For 186 drivers, a total travel time of 6127 ticks and an average 32.94 minutes travel time is record. To take 32.94 minutes for a 20km/patches way, the agents traveled at an average speed of 36.43km/h.

5.4. Third Experiment

In the third experiment, we will run only agents with the quickest path strategy. The parameters are drivers_ratio = 0.0, velocity_delta = 0.10, num-drivers = 1.0.

For 170 drivers, a total travel time of 6458 ticks and an average 35.48 minutes travel time is record. To take 35.48 minutes for a 25.6km/patches way, the agents traveled at an average speed of 43.31km/h.

In this simulation, all the agents only use the upper "highway route" for simplicity. Although the travel time is higher, the agents travel at higher speed.
5.5. Fourth Experiment

In the fourth experiment, we will run only agents with the shortest path strategy, so they only travel in the City Center Links. The parameters are drivers_ratio = 1.0, velocity_delta = 0.10, num-drivers = 5.0.

![NetLogo Output - Average Velocity vs Total Drivers for experiment four (Own production)](image1)

In this experiment we put the num-drivers = 5.0, which means for each iteration and agents creation, the total agents will be five. With this, we reach the maximum capacity of the network and the agents velocity reach the minimum of almost zero. What we can observed is that, when the agents started to exist the network the curve at figure 10, started to grow again, implying a velocity grow.

6. Conclusion

The first results points to the model usefulness. The main model objective, in the first stage, is to prove the agents can change their status and adapt to current traffic conditions. More, the agents can also have different strategies to reach their goal.

In the simulation presented, we can find that agents can make decisions and change their behavior according to the present network status and thus, in a day-to-day dynamics they will predict and change routes according to their experiences.

With the simulation we can predict, what will be the agents demand in the following day, and what will be expected velocity for the next day. These variables will be important to implement the learning process.
7. Discussion and future work

An agent-based model approach can contribute around the design and control of intelligent transportation systems (ITS) and ultimately make our cities smart. The FSM is the primary tool (McNally, 2000) for forecasting future demand and performance of a transportation system. In this model, in a simple network example, we show the different individual strategies can influence the travel time in a metropolitan network.

In the Dynamic O-D part, the framework assumes a fundamental part in the FSM. It shows a logical framework, which combines the FSM with day-to-day dynamics forces, the structural changes and some random changes. This is an important topic for this work, which aims to predict and estimate a day-to-day traffic analysis.

In the Multi-Model choice the agents can opt for as many modes they want. The implementation is ready to adapt to that situation but at this stage, the focus is only at private transportation mode.

The next steps we will adopt learning, multi-modal network and expand the network. Learning will work as a variable that will store the agent's experiences so they can change their present behavior. The multi-modal network, we will create agents that will act as bus and trains. Finally, we will expand the network, by creating an option to read city maps and to develop a network on top of that.

References


Vasirani & Ossowski (2011) A computational market for distributed control of urban road traffic systems. IEEE, 12(2).