Proceedings of the 16th Annual Conference of the Transportation Science Society of the Philippines
19 September 2008, Pasay City, Metro Manila

MODELING MOBILE TRAFFIC AGENTS ON NETWORK SIMULATION

Kardi TEKNOMO
Associate Professor
Department of Information Systems and Computer Science
Ateneo de Manila University
Loyola Heights, Quezon City
Philippines
E-mail: teknomo@gmail.com

Abstract: This article describes a new and novel approach to model dynamic simulation of mobile traffic agents such as pedestrians and cars on a simple network graph. The modeling approach is based on route choice self-organization of multi agents. In contrast to the traditional method of traffic assignment that assigns probability on each route based on a generalized travel cost, our model considers route probability as direct output of the simulation rather than an input to the network. The self-organizing route choice happens as a dynamic feedback loop that optimizes the product between global and local information.

Key Words: Route Choice Self-Organization, Dynamic Traffic Assignment, Multi-Agent Simulation, Macroscopic Pedestrian Simulation, Sink Propagation Values

1. INTRODUCTION

Multi agent traffic simulation is an abstract representation of real world traffic agents into mathematical equation or computer program that might give a new paradigm to evaluate the outcome of various design, control and policy scenarios. Simulation is a great tool as virtual laboratory to experiment with various types of facilities and behavioral rules and what-if scenarios.

A mobile traffic agent or just *agent* in short is defined as an intelligent autonomous discrete character (such as pedestrians, goods, vehicles and mix traffic) that moves on a network graph, lattice grid, continuous space or hybrid environment based on local rules from an origin point to a destination point. The term autonomous indicates that the agents have their own minimal intelligence to sense, to decide, and to react or to adapt independently based on a set of rules and agent's observation of the environment. Each agent has it own properties such as position and speed and many other parameters that affect agent's behaviors.

Agents have limited ability to dynamically sense or gain information from the environment. Through the sensing ability, agents know the existence of other agents and the existence of obstructions and facility such as roadway for vehicle and stair or elevator for pedestrian.

Agent also has minimum intelligence to decide which way to go. There is no requirement for user intervention on agents' decision. The decision is based on agent's observation of the current situation and simple local rules.

The agents have the ability to make decisions which lead them to adapt to their environment. The agents' behavior is affected by the current condition of the environment. If the environment is dynamic such as the possibility to go from red to green light of traffic signals or opening and closing of gates, the agents' decision would be realized as their adaptive ability. In terms of closing doors for pedestrian agents, the agents may even attempt to find another path.

The notion of local rules that is used to decide the path comes from two types of information. The first information is global information of navigation that is transformed into local information through a function that I call as Sink propagation Values (SPV). The global information provides full information regarding the environment while the local information is only available within the vicinity of agent's neighborhood (e.g. sight distance). The second information is local information of interaction with other agents. This information measures the density or how many other agents surround the agent. Both local and global information are stored inside the mind of each agent and updated every time step of the simulation.

Without losing generality, the illustration in this paper mainly use pedestrian agents on a network graph as examples to demonstrate simulation of mobile traffic agents. Pedestrian movement has a much higher degree of freedom than vehicular movement so it makes vehicular traffic a subset of pedestrian traffic in the simulation. For example, pedestrian is able to turn direction almost immediately and the environment that pedestrians move is 3 dimensional space (e.g. using elevator & escalator). Therefore, vehicular movement can be viewt to fgen 009t caase of pedestrian movement model given its many constraints (such as steering wheel and inability to climb surfaces such as stairs). With some modifications, the same model could also be applit tfor any type of agents from pedestrians, goods, vehicles and mixed traffic.

The representation other than network graph has been published elsewhere. Most pedestrian simulation models were done in continuous space (see Helbing and Molnár (1995), Teknomo and Gerilla (2005) and Teknomo (2006)) or lattice grid (see for Blue and Adler (2000), Schadschneider (2001), Kretz and Schreckenberg (2006), Teknomo and Millonig (2007)). Only a few models have been published in network graph (e.g. Lovas (1994) that uses queuing network model). The approach in this paper is somewhat different from Lovas (1994) in that it gives much greater flexibility and intelligence to the agents to move in a dynamic environment, more autonomous based on local rules.

The significance of modeling mobile traffic agents using network graph is two folds. First, compared to ordinary traffic simulation or traffic software, multi agent simulation provide more realistic behavior as we could track the trajectory movement of each agent. Second, compared to dynamic traffic assignment (DTA) model, this model is generally much faster to compute due to many simplifications.

Furthermore, the main contribution of this paper is to reverse the process of traditional method of traffic assignment that assigns probability on each route based on a generalized travel cost. The model in this paper considers route probability as direct output of the route choice self-organization (RCSO) rather than an input to the network.

2. TRADITIONAL TRAFFIC ASSIGNMENT

To be able to appreciate the benefit of route choice self-organization (RCSO), this section summarizes traditional method of traffic assignment as described in Kachroo and Ozbay (1999) and Ortuzar and Willumsen (2001) .

The basic methods of loading in traditional traffic assignment is either all or nothing or multipath. All or nothing is to assign all trips of a designated origin-destination traffic flow (or OD flow in short) to the shortest path. In this sense, the probability of a route r is given by

$$p_{od}^{r} = \begin{cases} 1 & if\ \ r \in \min \text{cost} \\ & \\ 0 & other\ routes \end{cases} \tag{1}$$

Though it is not a realistic load, all or nothing is heavily used in many algorithms and commercial software due to its simplicity and it is fast to compute.

The other common loading type of traffic assignment is multipath assignment or stochastic proportional method where all trips of a designated OD flow is assigned to all possible routes with proportion according to the impedance. The probability of a route r is given by

$$p_{od}^{r} = \frac{\exp(-\lambda c_r)}{\sum_s \exp(-\lambda c_s)} \tag{2}$$

The parameter λ is chosen such that longer route will get smaller probability to be selected. Multipath assignment has a major drawback that coding strategy will effect allocation of flows. It tends to allocate more traffic to dense sections of network with short links compared to sparse sections with longer links.

In both methods of all or nothing and multipath assignment, the three step computation is performed for each OD pair which are to get all possible routes and the costs, to compute the probability of each possible route and to assign OD flow according to the probability along the route. Finally, the flow of each link is obtained by simple summation of all link flows.

The basic methods of traffic assignment above did not consider the existence of physical road width that affects the capacity of the links. To be more realistic, cost flow functions are considered for each link to reflect on link capacity. Incorporating the cost flow function, the basic methods of traffic assignment become capacity restraint traffic assignment which basically finds the equilibrium condition either in terms of user equilibrium or social equilibrium. Under user equilibrium conditions, the traffic flow shall be arranged in congested networks such that all routes between any OD pair have equal and minimum cost while all unused routes have greater or equal costs. Under social equilibrium, the traffic flow should be arranged in congested networks in such a way that the average (or total) cost is minimized.

The computation is iterative such that the travel cost of one loading is utilized for the next iteration with some small percentage of traffic flow at the beginning using all or nothing. There are many algorithms to solve such iteration into convergence and reasonable speed with few notable ones such as Method Successive Average (MSA) and Frank-Wolfe Algorithm.

There is something in common in all above traditional method of traffic assignment as shown in figure 1. In all assignment methods, the probability of each route is computed based on a generalized travel cost and the assigned flow. The generalized travel cost is also computed from the assigned flow. Thus, first we need to assign the OD flow then we compute the generalized travel cost. In turn, we compute route choice probability and use that route choice probability to assign OD flow. The iterative procedure is used to converge the assigned flow.

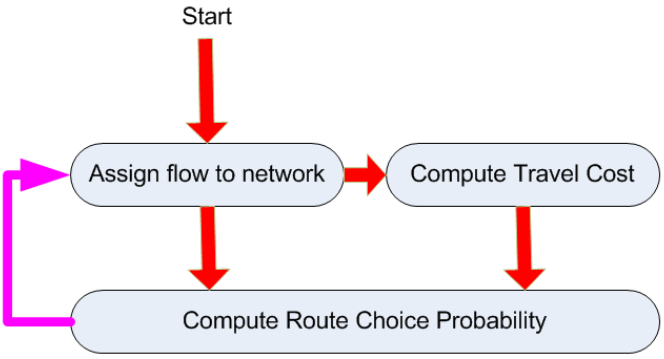


Figure 1. Iterative procedure of traditional traffic assignment

3. ROUTE CHOICE SELF ORGANIZATION

The backbone of the proposed method in this paper is based on route choice self-organization (RCSO) as described in more detail in Teknomo et al (2008). RCSO is a phenomenon that agents autonomously and endogenously, during the simulation, decide to change their plans with respect to route choice. In reality RCSO is observed in crowded scenarios, where people tend to deviate from the usual chosen shortest path in an attempt to avoid congestions.

In contrast to the traditional traffic assignment, the proposed multi agent model does not have both explicit models for route choice or traffic assignment. Figure 2 shows the procedure of the proposed multi agent model which can be contrasted to Figure 1. In the proposed model, the agent is moving in the network naturally, based on the product of local rules of permission, interaction and navigation. Instead of using logical iteration over the flow which has no physical interpretation, the multi agent model ‘iterates’ based on the dynamics of the simulation. In this sense, the iteration is performed by incorporating time as a variable. The route choice and traffic assignment in this case, is an emergent phenomenon from the distributive natural movement of the agents that run in parallel over time. The natural movement of the agents creates spatial dispersion over the network and the spatial dispersion creates queue in some places on the network. As we give flexibility to the agent to select either to go with the queue or to select other paths, the agents which have not been trapped in the queue would sense the queue situation ahead and select dynamically other longer routes to destination or to join the queue. Over time, this kind of individual decisions will fill up the network and produces the loading of traffic assignment.

The route choice of each agent, flow on each link as well as travel distance and travel time can be viewed back as an output of the computation rather than as input to the model.

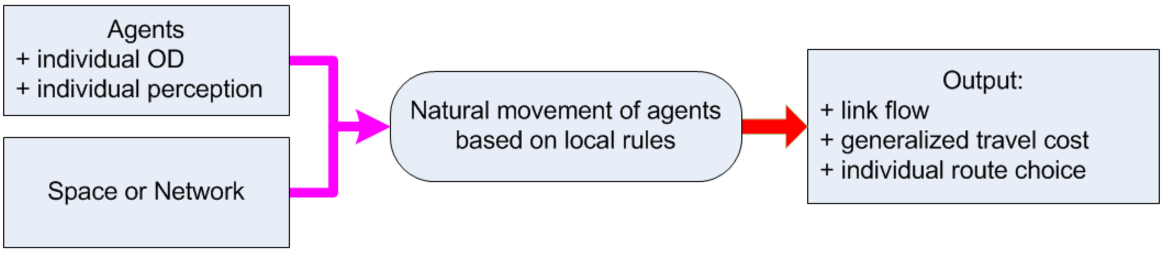


Figure 2. Straightforward procedure of RCSO

The individual route choice to go away from the shortest path happens when the shortest paths are perceived by the individual agents as congested or overcrowded. The notion of congestion here is subjectively decided by the individual agent. What one agent considers as a free flow might be considered as normal traffic by the other agents and a situation where a link is considered as congested might be thought of as overcrowded by other agents.

When the number of agents is relatively low, all the agents will move freely through the shortest path similar to the road traffic situation during early morning. If we increase the number of agents by the dynamic OD flow, the interaction among agents start to play around on certain links of the network with reduction of travel time but the individual route choice remains on the shortest path. As the OD flow increases to morning peak, some links would be considered as congested by some of the individual agents and they start to select alternatives routes.

It is the selection of alternatives route which is modeled in this paper as self-organization phenomena to find optimum ways to go to destination. This phenomenon is an emergence behavior (unsupervised-learning) in which the arrangement arises from the interaction of agents rather than as the result of centralized rule in the model.

4. MODELING MULTIAGENT MOVEMENT

This section describes in mathematical detail the proposed model.

Agent’s movement is directed from the origin vertex to the destination vertex, i.e. only one-directional movement is considered here, although also multi-directional movement could be included with little extra complexity. When reaching a vertex they decide which edge to enter next. This decision is done autonomously based on a set of rules using the agent’s observation of the local environment. In our model, the agent’s sensing ability is limited to only observe the edges’ density and edges’ space capacity adjacent to the vertex they are currently located in. Additionally the agents have complete information of the distance to the exit at each vertex.

The environment where the pedestrians live is a space aggregated into a directed network graph. The three-dimensional network graph is a non-planar multi-graph as it may contain multiple edges. There is no limitation on the number of agents that can be accommodated within a vertex. Each edge in the graph represents real space such as road midblock, rooms, doors, or facilities such as stairs, ramp, elevator, and escalator, etc. Therefore, two main properties of an edge are an *equivalent width* W and an *equivalent length* L. I called these properties as equivalent width and equivalent length instead of simply width and length because they can also be used to represent other impedance factors rather than mere distance. For example, equivalent length can represent dynamic average server time in queuing network. When the edge represents real space of a room, equivalent width and equivalent length can be shorten as width and length of the room.

The notion of capacity is as a direct measurement of physical maximum space against the user's perception of the space rather than logical maximum flow. Space capacity of a link $\overrightarrow{ij}$ is defined as a product of equivalent length $\ell_{\overrightarrow{ij}}$, equivalent width $\omega_{\overrightarrow{ij}}$ and agents' perception on maximum density $\rho_a^{\max}$.

$$c_{\overrightarrow{ij}} = \ell_{\overrightarrow{ij}} \omega_{\overrightarrow{ij}} \rho_a^{\max} \tag{3}$$

We define space capacity based on the product of actual space and perception of the agents. When the agent could bear to stay in a crowded situation, the perception of maximum density would be higher than the agents who cannot stand overcrowded situation. Lower agents' perception on maximum density tends to make the agent to select space with fewer pedestrians.

The agents' movement must be started and ended somewhere in the node of the network graph. A source is a set of origin vertices where the agents start their journey. A sink is a set of destination vertices where agents stop for some activity or final destinations of the agents.

Decision to move from one space to the other space is assumed only take place in vertices. As the agents only use local information, we need to identify that the local information is the information from the current vertex where the agent stay to decide and neighbors of the current vertex. The agent's decision to enter the next edge is guided by three principles which can be called 'permission', 'interaction' and 'navigation'.

Permission is represented by the direction of the link on the network as well as possibility of dynamic open and close node to model green and red traffic signal or open and close gates for pedestrian. Permission value is a binary quantity to indicate whether an agent a is allowed to enter certain vertex v at a specified discrete time t. As we model the environment with network graph, the permission is indicated by the direction of the arrow in each edge. If two vertices are not joined by two way directional edges, the agent is not permitted to move directly between those two vertices.

Interaction on the other hand takes the fact into account that agents attempt to avoid perceived crowded links. Within the model the interaction uses only local information represented by a generalized cost function of the observed density on the edges adjacent to the decision vertex where the agent currently is located. Interaction is represented by a function of edge density. If the edge density is high, the proportion of agents to go to that link is reduced. Suppose the agent position is in current vertex i and let j be the neighbor vertex (for self-loop, $j = i$). And let n_i be the total number of edges out of node i, $\rho_{\overline{ij}}$ is the current density of edge $\overrightarrow{ij}$ and $c_{\overline{ij}}$ is the space capacity of edge $\overrightarrow{ij}$, then the interaction at edge $\overrightarrow{ij}$ is measured as a proportion that an agent who is currently at vertex i will enter edge $\overrightarrow{ij}$. It can be noticed here that similar to the traditional transportation model that employ generalized cost function to adjust the parameters, the length of the edge is hidden as space capacity.

$$I_{\overline{ij}} = 1 - \text{BetaCDF}\left(\frac{\rho_{\overline{ij}}}{c_{\overline{ij}}}; \phi, \varphi\right) \tag{4}$$

Where, $\frac{\rho_{\overline{ij}}}{c_{\overline{ij}}}$ is *edge density ratio* and $\text{BetaCDF}(x;\alpha,\beta)$ is cumulative Beta *Distribution* function which can be computed as $\text{BetaCDF}(x;\alpha,\beta) = \frac{\mathrm{B_x}(\alpha,\beta)}{\mathrm{B}(\alpha,\beta)}$. Nominator $\mathrm{B_x}(\alpha,\beta)$ is Incomplete Beta Function formulated as $\mathrm{B_x}(\alpha,\beta) = \int_0^x g^{\alpha-1}(1-g)^{\beta-1}\,dg$ while the denominator $\mathrm{B}(\alpha,\beta)$ is Beta function formulated similar to Incomplete Beta function with $x=1$, which is $\mathrm{B}(\alpha,\beta) = \int_0^1 g^{\alpha-1}(1-g)^{\beta-1}\,dg$. Range of parameters is $\alpha > 0$ and $\beta > 0$. Linear relationship is a special case when $\alpha = \beta = 1$.

Navigation refers to a notion of distance to the destination point without taking other agents into account. Often navigation uses shortest path methods. With respect to navigation a concept called “sink propagation value” (SPV) is used. Here SPV is a function assigning a value to each vertex that is implementing a general notion of distance from the sink. I derive the name from the fact that the SPV of the vertices are propagating from the sink node into all other connected nodes. Using SPV it is possible to transfer the global information of distance to the sink into local information available at each vertex. Different SPV concepts can be obtained implicitly by using computational methods such as reinforcement-learning (i.e. Q-Learning), smoothing relaxation, Bellman flooding algorithm and the distance transform.

Navigation is represented by a function of Sink Propagation Value (SPV), which is vertex value that guides the navigation of the agent. Instead of selecting the optimum value, we can also make the SPV biased toward slightly second or third-optimal values though a multiplication with a probability. Let v_i and v_j be the sink propagation value of current vertex and neighbor vertex j respectively, we can define *normalized SPV difference* at current vertex as follow

$$z_i = \frac{v_j - v_i}{\max_j\left\{\left(v_j - v_i\right)\right\}}, \quad j \in \Gamma(i) \tag{5}$$

Then, the Navigation is defined as

$$N_i = \text{BetaCDF}\left(z_i; \vartheta, \theta\right) \tag{6}$$

When $z_i = 1$, optimum (i.e. shortest path by the definition of SPV) vertex will be selected. Bias towards sub-optimum value is done through parameter setting between zero and one.

Decision on which edge agent will move is only made at current vertex i. Selected edge $\overrightarrow{ik}$ is given by

$$k = \arg\max_j \left(I_{\overrightarrow{ij}} \cdot N_i\right), \quad j \in \Gamma(i) \tag{7}$$

Current speed v_t is adjusted based on current link density ρ_t at $t = t_{in}$ and speed density relationship for every time step $v_t = f(\rho_t)$. Let $\frac{\rho_{\overrightarrow{ij}}}{\rho_a^{\max}}$ be the *normalized density* based on agents' perception of maximum density, then

$$v_t = 1 - \text{BetaCDF}\left(\frac{\rho_{\overrightarrow{ij}}}{\rho_a^{\max}}; \varsigma, \tau\right) \tag{8}$$

Let t^* be the actual time when the agent goes out of the edge, computed as ceiling of simulation time such that the simulation time is higher than the pre-computed time to exit. The actual time to go out of the edge is given by

$$t^* = \left(\lceil t \rceil \,\middle|\, t \geq t_{out}\right) \tag{9}$$

Thus the macroscopic pedestrian simulation model has six parameters to be calibrated which are ϕ and φ for the interaction, ϑ and θ for navigation, ς and τ for speed-density relationship. Note that the fundamental diagram is the input of the model (through speed-density relationship) rather than the output.

The individual RCSO happens due to balance between navigation and interaction. Navigation criterion makes the agent find shorter route while Interaction criterion tends to make the agent select a longer route by reducing the probability to go to higher density links. The RCSO emanates in relatively dense scenarios where the optimal route in terms of travel distance is abandoned by agents due to their preference to avoid crowded edges, i.e. in a sense the agents anticipate congestion and take the corresponding delays into account. This can be demonstrated using a comparative static analysis analyzing the choice between two alternative routes: At low levels of flow the navigation term will dominate the interaction term. Due to the navigation criterion to select the shorter route, edges which correspond to the shorter route will be filled first. Increasing the flow level, the density on the shorter route will increase making the interaction term more important. Consequently more agents will choose the longer route to avoid regions of high crowd density. Increasing the flow levels even higher up to a level where also the alternative route is congested the navigation again gains importance. This conforms to observations in real world scenarios.

5. NUMERICAL EXAMPLE OF MACROSCOPIC MODEL

This section illustrates the manual computation of RCSO. I give illustration of a simple network with 6 nodes as shown on the left of Figure 3. The distance is given on the link while the space capacity is one for all links. The distance matrix is given as follow

$$\mathbf{D} = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{bmatrix} 0 & 5 & 7 & 0 & 0 & 0 \\ 0 & 0 & 4 & 7 & 0 & 0 \\ 0 & 5 & 0 & 2 & 3 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 6 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \qquad (10)$$

The sink propagation value are computed and shown in the right of Figure 3.

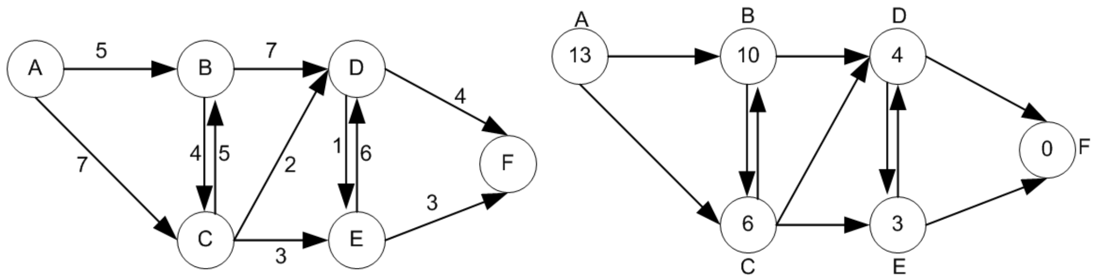


Figure 3. Network example with distance (left) and SPV (right)

For the sake of illustration, we simulate 100 pedestrian agents by varying only two main parameters. Maximum density is 5 pedestrians per square meter and maximum speed is 1.2 m/s. Figure 4 shows some variation on the interaction parameters on the network. All other navigation parameters and parameter of speed-density relationship are set to default value of 1. The figure describe that the relationship of parameter set and flow pattern is not a one-to-one correspondence. In fact, it is not a function that maps a set of parameters to a set of flow pattern. A set of parameters will produce a set of flow pattern, but other set of parameters may also produce the same set of flow pattern. In other words, we can make various parameters set to obtain the same flow pattern. Parameter set is the necessary condition to obtain the flow pattern but it is not the sufficient condition. This results show that we cannot make flow pattern as a criteria of the objective function in parameter calibration. For example, the calibration cannot be done by fitting some flow count at certain edges such that the sum square error of the different dynamic flow count of all edges is minimized. This type of calibration could produce a set of parameters, which are not optimal.

More varied route categories produces more flow to fill edges that are not in the shortest path. Filling the flow of edges can be done by producing more variation of routes. Also, notice that forcing all 100 pedestrian agents to use the shortest path (route 1-3-5-6) will not produce the minimum egress time. In fact, it has very high egress time that may happen in panic situation (512 seconds compare to optimum egress time of 132 seconds or compare normal situation of 262 seconds). This result highlighted the exact reverse to the common belief that shortest path yield to the minimum evacuation time. That common belief may be true if and only if only a minimum number of agents are present on the system. As the number of agents in the system increases, filling through the main routes may yield optimum system in terms of minimum egress time.

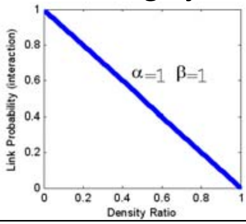
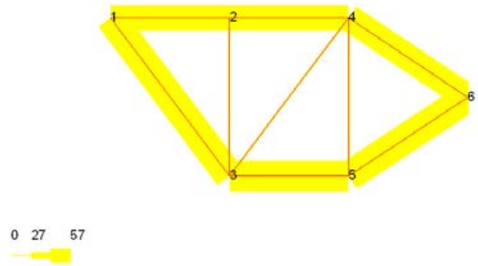
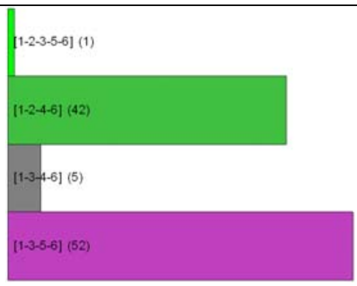
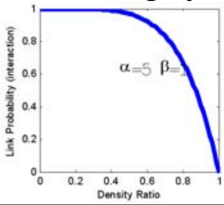
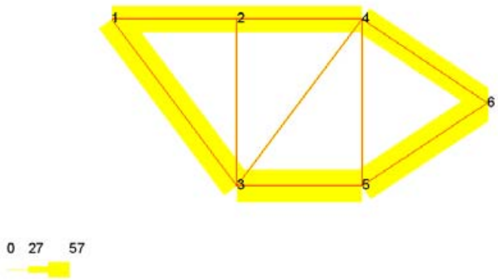
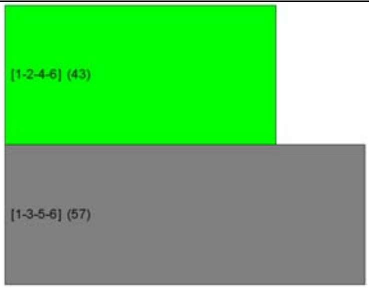
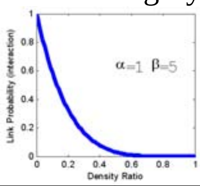
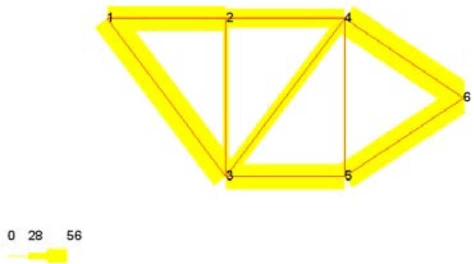
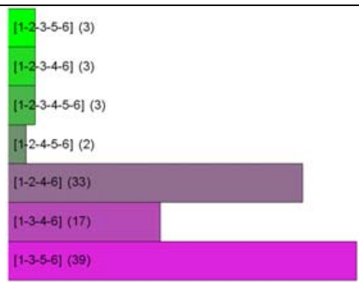
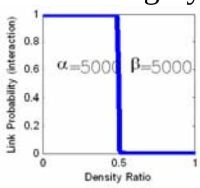
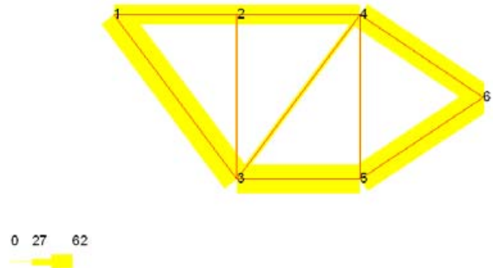
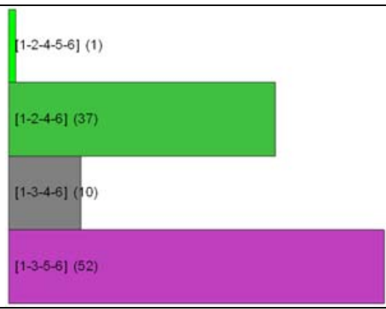
Parameters	**Flow Pattern**	**Route**
$\phi = 1, \varphi = 1$ Egress Time = 262 Route Category: 4 		
$\phi = 5, \varphi = 1$ Egress Time = 262 Route Category: 2 		
$\phi = 1, \varphi = 5$ Egress Time = 255 Route Category: 7 		
$\phi = 5000, \varphi = 5000$ Egress Time = 132 Route Category: 4 		

Figure 4. Effect of varying parameters on Network example

The dynamic node density of the 6 vertices is exhibited in figure 5. It shows that the source vertex density is going down over time and sink vertex is going up over time because no actual limit of density on the node. All other vertices in between do not have density higher than 5 pedestrian agents per square meter at the same time.

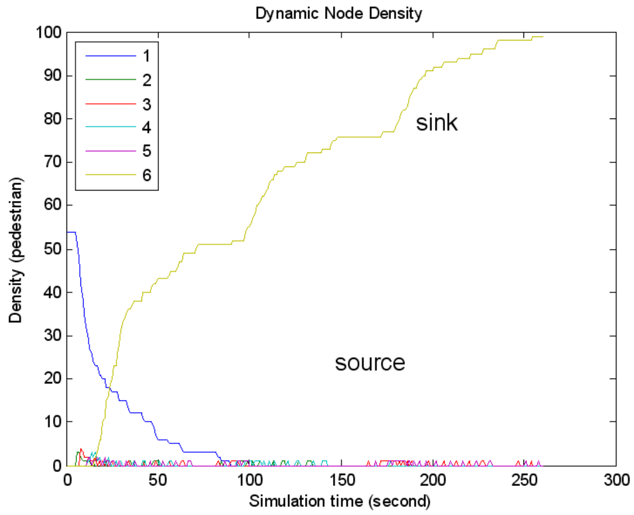


Figure 5. Dynamic node density of Network example

6. CONCLUSIONS

The route choice self-organization can be seen as a new and novel alternative to existing multi-agent dynamic traffic assignment models. The proposed model is much simpler than traditional traffic assignment model that it incorporates natural movement of agents to establish individual route choice. As the result, the model is able to handle dynamic traffic assignment based on route choice self-organization.

The illustration example of pedestrian agents over the network graph indicates that non-unique set of parameters would produce the same set of link flow pattern. Therefore, we cannot use flow pattern as criteria of objective function in parameter calibration.

It was also found out that shortest distance path does not necessarily lead to minimum egress time which is useful for evacuation application. This finding is especially important in the present of panic crowds. It implies that evacuation route should be analyzed further as it is not necessarily the shortest distance as commonly believe by many people.

ACKNOWLEDGEMENT

This research was funded by the Austrian Science Foundation through a Hertha Firnberg research grant.

REFERENCES

Blue, V.J. and J.L. Adler (2000). Cellular Automata Microsimulation of Bidirectional Pedestrian Flows, Transportation Research Board, 1678, 135-141.

Helbing D. and Molnár P. (1995). Social force model for pedestrian dynamics, Physical Review E 51, 4282-4286

Kachroo, P. and Ozbay, K. (1999) Feedback Control Theory for Dynamic Traffic Assignment, Springer-Verlag, London.

Kretz, T. and M. Schreckenberg (2006). F.A.S.T.- Floor field- and Agent-based Simulation Tool, Proceedings of International Symposium of Transport Simulation 2006.

Lovas, G.G. (1994). Modeling and Simulation of Pedestrian Traffic Flow, Transportation Research B, 28, 429-443.

Ortuzar, J.D. and Willumsen, L. G. (2001) Modelling Transport, 3rd Ed. John Wiley and Sons

Schadschneider A, (2001). Cellular automaton approach to pedestrian, dynamics-theory, in Pedestrian and Evacuation Dynamics (M. Schreckenberg, S. D. Sharma, eds.), Springer, Berlin.

Teknomo, K. (2006). Application of microscopic pedestrian simulation model, Transportation Research Part F: Psychology and Behaviour Vol 9 issues 1, p. 15-27.

Teknomo, K. and A. Millonig (2007). A Navigation Algorithm for Pedestrian Simulation in Dynamic Environments, Proceeding of the 11th World Conference on Transport Research (WCTR), University of California, Berkeley, USA.

Teknomo, K. and Gerilla, G. P. (2005) Sensitivity Analysis and Validation of a Multi-Agents Pedestrian Model, Journal of the Eastern Asia Society for Transportation Studies (EASTS), Vol 6, September 2005, p.198-213

Teknomo, K., D. Bauer and T. Matyus (2008). Pedestrian Route Choice Self-Organization, Proceeding of the 3rd International Symposium of Simulation in Transportation, Gold Coast, Australia.