Computation of location choice of secondary activities in transportation models with cooperative agents

Fabrice Marchal
Computational Laboratory
Swiss Federal Institute of Technology
Hirschengraben 84
8092 Zurich, Switzerland
marchal@inf.ethz.ch

Kai Nagel
Transport Systems Planning and Transport Telematics
Technical University Berlin
Salzufer 17-19 SG 12
10587 Berlin, Germany
nagel@vsp.tu-berlin.de

Abstract

Activity-based models in Transportation Science focus on the description of human trips and activities. We address the modeling of activity location decision for large data sets: given both home and work locations, where do individuals perform so-called secondary activities (e.g shopping)? We propose a model where agents have limited, accurate information about a small subset of the overall spatial environment. Agents are interconnected by a social network through which they can exchange information. This approach has several advantages: a) it can be faster to find suboptimal solutions to build plausible choice sets b) the learning speed of the overall process is governed by the greediness of the exchange and c) it can provide a useful framework to study the propagation of any newly available information.

1. General context

Activity-based models in Transportation Science focus on the description of the organization of human activities in time and space. This organization determines the demand for travel, that is the amount of users that the various transportation systems need to accommodate. It is assumed that the demand for travel is derived from the demand for performing activities at specific locations. Obviously, individuals constantly perform some trade-off between enjoying activities that have a high reward value (for instance working at a company in the downtown area) and the time and budget it takes to get to the specific location of these activities. Various operational models such as URBANSIM **?** are available to describe this trade-off for the choice of the home and work locations. It is essentially assumed that users perform a trade-off between rents, travel costs and wages. However, empirical evidence **?** have shown that a significant amount of traffic is generated for other purposes than commuting, often referred to as *secondary* activities: shopping, leisure, going to social events, etc.

2. Problem statement

Our work intends to model the specific process of the location choice of secondary activities in the case of high resolution data sets. The methodological constraints are that the modeling should be behaviorally sound, compatible with microeconomics foundations and computationally feasible. The temporal dimension (i.e. the scheduling of the activities) is ignored for the time being. We assume that the order of the activities called a *plan* is given (i.e. getting out from home, going to work, working for eight hours, going for shopping at lunch time, etc.). The physical environment is described by two large data sets that typically originate from Geographical Information Systems (GIS): a) the land-use data and b) the transportation system data. The land usage is a raster-type description that includes the information about the nature of each parcel of the studied area (e.g. housing density, number of shops, type of area: rural, commercial, industrial). The transportation system is a vector-type description of the various transportation mode available (e.g. car, rail, bus) as a network with nodes and links. Nowadays, these data have reached a very high resolution: typical land-use cells are 100 meter square and road networks are described down to 10 meter road sections. The long-term goal of our research is to model entire metropolitan areas microscopically by simulating the individual decision of millions of citizens. Therefore, the problem at hand can be stated as follows: how to simulate the selection of the activity locations of $A = 10^6$ citizens in a grid that has $C = 10^5$ cells. Note that the travel times from cell to cell have to be given by external traffic models. Multi-agents traffic assignment models such as those developed by the authors **??** are now able to predict travel time patterns for large-scale data sets. Therefore, this issue will not be considered here. The generation of travel demand for these models is addressed here.

3. Micro-Economics foundation

The standard practice in Transportation Science to approach such problems is to use random utility models (RUM) borrowed from the discrete choice theory of Micro-Economics (**??**). These models assume that individual are maximizing their own utility. For instance, the utility to go shopping at a mall located in cell i for a simple plan (i.e. home - shopping - work) is given by

$$U_i = R_i - C_{hi} - C_{iw} + \mu\epsilon_i = V_i + \mu\epsilon_i$$

, where R_i is the reward associated to shopping at that particular facility that depends on the availability of goods, their prices, etc.; C_{ih} is the travel cost to travel from home to cell i; C_{iw} is the travel cost to travel cell i to $work$; μ is a scale factor and ϵ_i is a random variable that is specific to the individual. The latter random utility part captures all the hidden preferences of a specific user for location i that are not accessible to the modeler. By contrast, V_i is called the deterministic part of the utility. Under the assumption that ϵ_i are i.i.d extreme value distribution of type I, it can be shown that the probability to choose to go shopping at cell k is given by:

$$P(k) = P(U_i < U_k \forall i \neq k) = \frac{\exp(-V_k/\mu)}{\sum_i \exp(-V_i/\mu)}$$

Since a probability greater than zero is assigned to each potential intermediary stop on a cell, this formulation requires a full enumeration of the possibilities on the spatial grid. For trips that count S intermediary stops, the complexity is $O(AC^S)$ which is not feasible in realistic cases. Initially, RUMs are intended for the description of choices between a finite set of alternatives distinguishable by humans (e.g. car brands). But their application to a discretized continuum (i.e. urban space) remains behaviorally questionable. Still, we believe it is fundamental to keep some compatibility with RUMs because a huge amount of the literature has been devoted to developing empirical techniques (e.g. surveys) to calibrate the parameters of those models. Another drawback of RUMs is that they provide only a static representation that does not take into account the temporal dimension of the decision process. RUMs do not model explicitly the learning process performed by human beings and the dynamics due, for instance, to some modification in the land-use patterns.

4. Multi-agent based approach

To overcome these limitations, we propose a multi-agent based simulation where each agent (i.e. each simulated citizen) has only limited, accurate information about N cells ($N \ll C$) called the "memory" of the agent. The intuition is that real humans have limited cognitive abilities and can only consider a small amount of options at the same time. The organization of these options in the mind and the human representation of space is probably far different from "pixels" but we put that problem aside for the time being. Furthermore, we assume that agents are inter-connected by a *social network* through which they can exchange information about their respective subsets. Each agent is socially connected to K acquaintance or "kins". The simulation is iterative and each round has four stages: evaluation, socialization, exchange and exploration.

Evaluation: each agent performs the location choice of the intermediary stops based on his own private information. The choice can be deterministic (the best cells are selected from the memory of the agent) or probabilistic (a RUM is applied to the finite set limited to the memory of the agent). The computing load for building and storing the travel plans is $O(AN^S)$. At this point, the plans are fed in some traffic model which is run to compute the delays incurred due to traffic congestion. These penalties are to be used in the next round of evaluation.

Socialization: social connections are created and delete dynamically. The deletion mechanism is a simple exponential decay. The creation mechanism is a spatial re-inforcement reminiscent of pheromones in ant colonies optimization **?**.

Exchange: for each of its social connection, an agent has the opportunity to exchange a piece of information. A cell is picked up randomly from the agent's memory and the other agent is informed about it. The exchange is bidirectional and the outcome of the exchange is described by the learning mechanism below.

Exploration: agents have the possibility to explore neighboring cells to those that they visit. This stage is mainly intended to recover potential information loss in the other stages, thus relieving the implementation from checking that no cells is lost from the global knowledge of all the agents. Obviously $O(A)$ operations are required.

5. Learning mechanism

The memory of an agent is represented on Figure **??**: a first buffer contains the information about locations that are either close to home or close to work (e.g. a small circular area). The second buffer called "elite" buffer corresponds to location that have a high score values and the third buffer contains "vague" information about cells that have poor score values. When an agent inform another agent during the exchange stage, a cell is picked randomly from the three buffers of memory of the informer. At that point, the informed agent evaluates how this new cell information can potentially improve his/her plan score. This implies to evaluate the replacement of any intermediary stop by the new cell: $O(N^{(S-1)})$ operations are needed. If the score is better than the worst solution of the elite buffer, the new cell is promoted to that buffer and it is sorted: $O(K \ln K)$ operations are needed. If the cell does not improve one of the elite plans, the cell information replace a previous cell from the vague buffer. This has two consequences: Firstly, agents keep information that is not relevant to themselves but that might be to others in the future, hence they adopt a *cooperative* behavior that is not supervised. Secondly, the information in the vague buffer can be erased and lost forever, which is most likely except for cells with the lowest utility. The exploration phase allows still to recover them. So far, the computation load of a single round is $O(AN^{(S-1)}) + O(AK \ln K) + O(C)$ which is feasible for reasonable assumptions ($S \leq 3, N < 50, K < 50$). However, the number of iterations is still to be determined. Note that the learning speed of the overall process depends on the *greediness* of the exchange which is a function of the ratio between the sizes of the elite and the vague buffers.

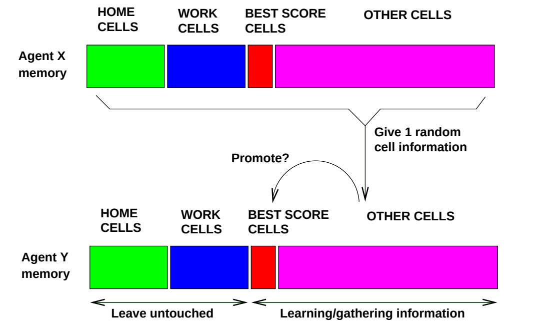

Figure 1. Agent memory organization and learning

6. Social network dynamics

Social links disappear following an exponential decay law at the end of each iteration. Initially, the social network is a random graph of degree K and each social connection has the same decay time. When two agents perform activities at the same location, two situations can occur: 1) if there is a social connection between them, the strength of the connection is reinforced and its decay time increases 2) if not, a new social connection is created. This task requires $O(AK)$ operations for the enumeration of the social links and $O(C)$ operations to browse the cells for detecting interactions between agents. However, the detection of the existence of a social link between two given agents would require $O(AK)$ operations for each cell to browse the connections. An alternative solution would be to store the connections of a given agent in a hash table. Both solutions are costly in memory or computation time. For these reasons, we adopt a slightly different implementation which is equivalent statistically. Every connection has the same decay time but multiple connections can exist between two given agents. When an agent visits a cell, the cell keeps a pointer to the agent until a second one visits the cell. A new connection is then created. A hash table of visited cells needs to be maintained, which is $O(\log(C))$ costly. At the end of each iteration, connections are randomly deleted so that the total number of connections in the system remains constant. Agents visit cells selected randomly from their elite buffer.

7. Implementation issues

The code of the simulation has been written in Java. Input and output files use the XML file format which is well suited to variable length content such as the description of the individual plans with multiple stops. The goal being to simulate 10^6 agents on a single CPU, some performance concerns have to be taken into account.

Cells: each grid cell is stored as an individual object. With 10^5 cells, it is not crucial to store the cell attributes as plain arrays. This allows to keep cell characteristics private and to have cell references. Since cells have to be often compared during the learning process to determine if a new cell is already known to an agent, it is far more efficient to use the equal operator == than the default Java *equals()* method. This is valid as long as the cells are not dynamically allocated or cloned once the simulation starts. Cells have to keep references to agents that visit them (see the socialization stage). The average number of visits per cell is small ($O(SA/C)$) so that we can allow for the overhead of a dynamic container (e.g. vector).

Agents: it is tempting to have a dynamic container steadily increasing in size for the agent memory. However, that would completely ruin the performances and is not compatible with our assumption that only a limited number of simultaneous options can be memorized. Each cell is referenced on average by $O(AN/C)$ so that the initial coverage is sufficient to ensure that there is not any information missing about the environment.

Random numbers: a typical bottleneck of this kind of simulation is the computation of random numbers. A priori, $O(AK)$ random numbers have to be computed for each single iteration of the information exchange stage. This quickly becomes prohibitive and can be avoided by using two integer random seeds at the beginning of the exchange stage. One is used to pick a cell from the informer agent, the other to replace a cell in the memory of the informed agent. These two pointers can be simply incremented from one social link to the other since there is no correlation between social links and they are accessed in a *a priori* random order.

8. Results

The simulation is tested on a real-world example for the Zurich region for which have available a high resolution transportation network and land-use raster (see Figure **??**). The area covers approximately a 50x50 kilometer square area where about one million inhabitants are living. The land-use utility values R_i are generated based on census data. Random plans with 1 or 2 intermediary stops are generated for 10^6 agents that are distributed on the area according to job and housing densities. The home to work pairs are computed using an external model written by one of the author (see **?** for the computation of the rent values presented in Figure **??**). The initial social network that connects them is a random graph. Obviously, this is not realistic but we intend to evaluate only the computational feasibility in this preliminary work.

Figure **??** presents the evolution of the sum of the scores of all the agents during the iterative process. It can be seen that the process converges in a few dozens of iterations but that the choices are not optimal since the utility does not reach the maximum value obtained with a full enumeration of the alternatives. This is due to the fact that some information is lost in the process and slow to be recovered. Still, the value of the plateau is high enough to ensure plausible strategies for the plans. Note that the size of the memory of the agent only slightly affect the convergence properties.

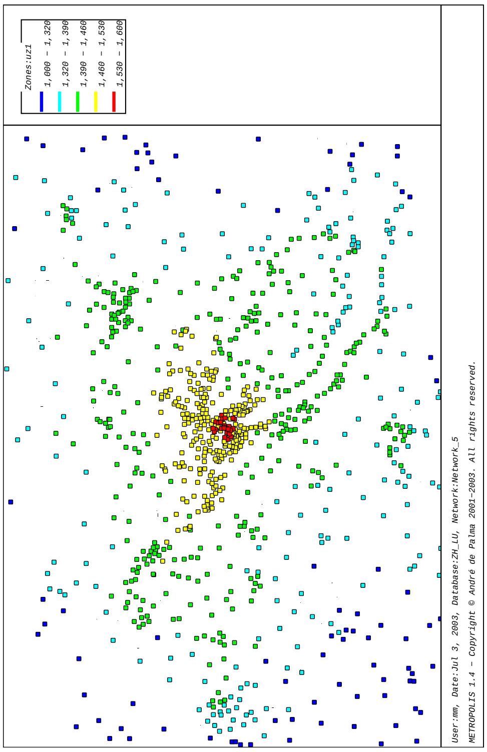
Zones:uz1
1,000 - 1,320
1,320 - 1,390
1,390 - 1,460
1,460 - 1,530
1,530 - 1,600
User:mm, Date:Jul 3, 2003, Database:ZH_LU, Network:Network_5
METROPOLIS 1.4 - Copyright © André de Palma 2001-2003. All rights reserved.

Figure 2. Zurich area: transportation network and rent values

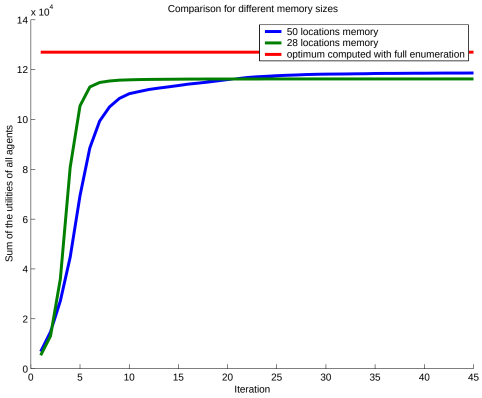

Figure 3. Performance of the multi-agent simulation

Figure **??** illustrates the spatial adaptation process for a single agent with a two-stop plan: home-work-leisure-shopping. On the first iteration, leisure ($L1$) and shopping ($S1$) are performed at the home place because the agent ignores good locations to perform these activities. In the second iteration, he/she learns that the location $L2 - S2$ is a good location for one of the two activities, hence making the extra trip distance worth it. On the third iteration, the agent discover that the area around $S3$ has a high utility for shopping. During the fourth and fifth iterations, the agent keeps shopping close to that area and only optimize the leisure location ($L3 \to L4 \to L5$).

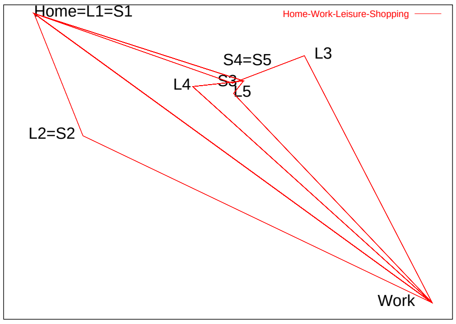

Figure 4. Adaptation of the location for a two-stop plan

Figure **??** shows the evolution of the distribution of social connections in the system. The initial condition is a random graph of degree $K = 20$. On the first iteration, the distribution is rouhgly a $N(20, 1)$ distribution. Progressively, the distributions shifts toward the left. In the end, we have a self-sustained distribution that can be approximated by a $N(13, 4)$ distribution.

All the experiments were done on a computer equipped with a Intel Pentium4 clocked at 2.5Ghz. The typical simulation performance for 100 iterations of a system with 10^6 agents is below one hour of CPU time. This is for plans that have only one or two intermediary stops. In term of memory requirement, about 400 Mbytes of RAM are needed. Obviously, the simulation of larger systems and more sophisticated plans with more than two stops will require to distribute the workload on several computers.

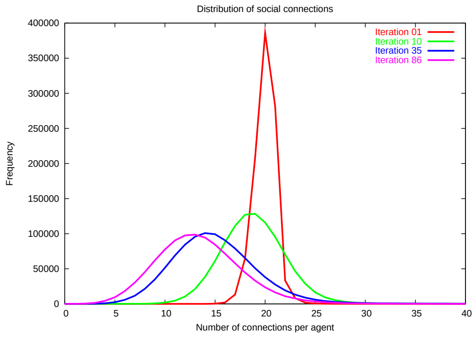

Figure 5. Distribution of social connections

9. Conclusions

This preliminary work has shown that a multi-agent based approach to the location of secondary activities is technically feasible and behaviorally plausible for high resolution data sets. Many aspects still need to be addressed to improve the realism of the model. Nevertheless, the fact that agents cooperate in even some simplistic way yields an important gain in term of computation workload. This has to be compared with the standard practice in Transportation Science where it is typically assumed that users are in the situation of a non-cooperative Nash equilibrium.

10. Acknowledgments

The author express their thanks to Prof. K. Axhausen and PTV AG for providing them with the data for the Zurich area. Credits go also to A. Altenhoff who wrote a first implementation of the simulation. Computer resources for the simulations were provided by the Computational Laboratory (CoLab) at ETHZ.