A generalization of the MATSim simulation process to simulate the influence of social contacts spatial distribution on mobility behavior

Author(s):
Dubernet, Thibaut Jean Pierre; Axhausen, Kay W.

Publication Date:
2015

Permanent Link:
https://doi.org/10.3929/ethz-b-000100092

Rights / License:
In Copyright - Non-Commercial Use Permitted
A Generalization of the MATSim simulation process to Simulate the influence of Social Contacts Spatial Distribution on Mobility Behavior

Thibaut Dubernet
Kay W. Axhausen

March 2015
A Generalization of the MATSim simulation process to Simulate the influence of Social Contacts Spatial Distribution on Mobility Behavior

Thibaut Dubernet
Institute for Transport Planning and Systems
(IVT)
ETH Zurich
CH-8093 Zurich
phone: +41-44-633 68 65
fax: +41-44-633 10 57
thibaut.dubernet@ivt.baug.ethz.ch

Kay W. Axhausen
Institute for Transport Planning and Systems
(IVT)
ETH Zurich
CH-8093 Zurich
phone: +41-44-633 39 43
fax: +41-44-633 10 57
axhausen@ivt.baug.ethz.ch

March 2015

Abstract

In their daily life, individuals are frequently involved in joint decision making — situations where several individuals have to agree on the actions they will undertake to achieve a joint outcome. Examples in the context of mobility behavior include intra-household task allocation, intra-household vehicle allocation, choice of the time and venue for a dinner with friends or traveling together in the same private vehicle.

In addition to being necessary to predict joint travel and car occupancy, it has been hypothesized that considering explicitly this kind of joint decision process for the case of leisure activities planning might help to improve the forecasts for the choice of the leisure destination, due to the often social nature of such activities (Axhausen, 2007). Evidences of the influence of social contacts meetings on activity durations, trip length, or type of activities could be found (Stauffacher et al., 2005; Carrasco and Habib, 2009; Habib and Carrasco, 2011; Moore et al., 2013). This has motivated the inclusion of social networks in simulation frameworks. For instance, (Hackney, 2009) experimented with the co-evolution of social networks and travel patterns in MATSim — without joint decision processes. Other examples include joint decision making (Han et al., 2011; Ronald et al., 2012; Ma et al., 2011; 2012).

This research aims at including social behavior — including joint decision making and coordination — in a multi-agent transport simulation in a meaningful way.

To this end, the MATSim simulation framework has been extended. MATSim provides a co-evolutionary algorithm to search for a user equilibrium over daily plans. The process was modified to shift the solution concept: instead of searching an individual-based equilibrium, it searches a state without blocking coalitions: a state where no group of agents can all improve
the utility they get from their day by changing their plans *together*, including the possibility of joint trips or activities.

Of course, adding the possibility to perform joint activities, including joint location choice, increases a lot the size of the search space. Different possible techniques to cope with this were experimented, and are discussed, in terms of a tradeoff between computational performance, quality of the results and code maintainability.
1 Introduction

In recent years, the importance of social processes to understand and model mobility behavior became more and more obvious.


The study of other kinds of social contacts is more limited, most probably due to the difficulty to collect reliable data. Several studies however focused on the influence of social contacts, and joint activities, on travel behavior (Carrasco et al.; 2008; Arentze et al.; 2012; Carrasco and Habib; 2009; Habib and Carrasco; 2011; Moore et al.; 2013; Deutsch and Goulias; 2013). Such studies reveal significant influences of spatial distribution of social contacts, or of the number of participants in a joint activity, on the traveling behavior of the participants. This is particularly important for leisure purpose trips, which represent an increasing share of the trips performed in western countries (Schlich et al.; 2004; Axhausen; 2005).

This short paper presents the inclusion of such joint decisions into a simulation framework, MATSim, with a focus on the computational challenges at hand. After a presentation of the software framework, results are outlined, both in terms of behavioral realism and computational effort.

2 Simulation Framework

The current section presents a simulation framework for the simulation of joint decisions for mobility behavior forecasting. After presenting the basic model, computational and practical challenges are detailed, along with their solutions.

This section gives a brief overview of the theoretical underpinnings of the simulation framework. More details can be found in Dubernet and Axhausen (2012, 2013b,a) or Dubernet and Axhausen (2014).

Game theory, as a theoretical framework to represent competition, has been used in many forms
in transportation research. One of the earlier examples, and probably one of the most influential, is the Wardrop equilibrium condition in traffic assignment (Wardrop, 1952), which is simply a Nash equilibrium of a specific congestion game. This equilibrium notion has then widely spread in transportation research in general, and traffic assignment in particular, and doing an exhaustive review is not the purpose of this paper.

2.1 Joint Decision and Game Theory

Although the outcome of any game is a decision "joint" in some way (the decision of a player depends on the decisions of the other players), this work uses a more restrictive definition of what is a joint decision.

A joint decision, as we understand it here, is a set of interlinked decisions by several players, requiring the usage of explicit coordination, or binding agreements. Including such possibility in a game theoretic framework requires a specific solution concept.

This can be illustrated by a classical game, called the House Allocation Problem (Schummer and Vohra, 2007). This game consists of \( n \) players and \( n \) houses. Moreover, each player has its individual ordering of the houses, from the most preferred to the least preferred, and players prefer being allocated alone to any house rather than in the same house as somebody else. The strategy of a player is the house chosen to live in.

An interesting feature of this game is that any one-to-one allocation of players to houses is a Nash Equilibrium: no player can improve its payoff by unilaterally changing its strategy, as it would require choosing an occupied house. This result however contradicts basic intuition about the stability of such an allocation. In this particular case, a more realistic solution concept is the Absence of Blocking Coalition: given a one-to-one allocation of houses to players, a blocking coalition is a set of players which could all be better off by re-allocating their houses among themselves.

It is to be noted that both solution concepts correspond to rational agents, i.e agents having a preference ordering over outcomes. What differentiates both solution concepts is the degree of communication which is hypothesized: in a Nash Equilibrium, for a given player, the strategies of the others are taken as given; in an Absence of Blocking Coalition, players have the possibility to "negotiate" a change of strategy with other players, which will be accepted only if all agents in the negotiation are better off after the re-allocation. In this work, we consider that a Nash Equilibrium corresponds to individual decisions only, whereas the blocking coalition concept allows what we name joint decisions.
2.2 A Solution Concept for the *Daily Planning Game*: the Absence of Improving Coalition

Given those remarks, a solution concept for the "daily planning game", including the possibility of binding agreements, is adopted: the *Absence of Improving Coalition* concept. Given an allocation of daily plans to individuals, an improving coalition is a set of social contacts that could all be better off by *simultaneously* changing their daily plan. One can think of a group of friends switching from individual dinners at home to a joint dinner in a restaurant.

2.3 Simulation Framework

The MATSim framework provides a *Controller* to build and configure co-evolutionary algorithms, where agents each optimize their plan given the (evolving) state of the transport system.

The basic modeling idea is that individuals associate a utility value to their day, which increases with the time spend performing activities and decreases with the time spent traveling. Different parameters can be used for different modes or activity types, using the functional form from Charypar and Nagel (2005). Travel time is influenced by other agents via congestion. Co-evolutionary algorithms are particularly well suited for this kind of problems (Ficici, 2004; Popovici et al., 2012).

Figure 1: The MATSim iterative process

The co-evolutionary algorithm used by MATSim to solve this problem is an emulation of a learning process, suggested by the essentially dynamicsolution concept (Nagel and Marchal, 2006). Using the formalization of the problem above and iterative learning analogy, the specification of the algorithm is quite natural: each agent will perform an evolutionary algorithm to optimize its own daily plan, the fitness of which will be evaluated by executing all daily plans.
on the network to evaluate the resulting state of the transportation system. The steps of this process are represented on Fig. [1]

The first step is the specification of an Initial demand. All agents have an initial daily plan, which will serve as a starting point for the iterative improvement process. Some characteristics of the plans are left untouched during the simulation, and should therefore come from data or external model. This is typically the case of long term decisions, such as home and work locations, or decisions involving a larger time frame than a single day (e.g. do the weekly shopping or not).

The second step is the Mobility simulation. Plans of all agents are executed concurrently, to allow estimating the influence of the plans of the agents on each other. This step typically uses a queue simulation to simulate car traffic, which gives estimates of the congested travel time. Simulation of bus delays due to congestion and bus bunching can also be included. Together with the next step, this step constitutes the evaluation stage of the co-evolutionary algorithm.

Then comes Scoring. The information from the simulation is used to estimate the score of each individual plan. This information typically takes the form of travel times and time spent performing activities; experiments also included information such as facility crowding (Horni et al., 2009). The functional form is the one used by Charypar and Nagel (2005). It uses a linear disutility of travel time, and a logarithmic utility of time passed performing activities. Different parameters can be defined for each mode/activity type.

This gives the score from a single interaction. The fitness of the daily plan (entity of the algorithm) can then be updated, as $(1 - \alpha) f_{old} + \alpha f_{new}$, with $\alpha \in [0, 1]$ being the learning rate. The lowest the learning rate, the more the fitness of a plan will be close to an average fitness over the evaluated interactions. While this is consistent with the hypothesis that individuals react to the expected state of the transport system, most applications use a learning rate of 1, which results in more reactive agents, and thus faster convergence.

The last step is Replanning. This step actually groups two of the important components of co-evolutionary algorithms: (a) selection of the interactions for evaluation, and (b) application of the evolutionary operators (selection and mutation). To do so, part of the agents select a past plan based on the experienced score, following a Logit selection probability. This will have two consequences: (a) the state of the transport system, used for evaluation, will only evolve slowly from iteration to iteration, giving the time to the agents to adapt, and (b) those plans will be re-evaluated, given the new plans of the other agents. The other agents copy and mutate one of their past plans. What kind of mutation is performed determines which alternative plans will be tried out by the agent.

Those steps are iterated until a stationary state is reached, and the state of the system in this
stationary state is taken as a result.

Given this general framework, to be able to implement an algorithm searching for states without blocking coalitions, one needs a way to represent the influence of explicit coordination on the utility of a daily plan. This is solved by including joint plans constraints. A joint plan is a set of individual plans executed simultaneously. Different copies of the same individual plan can be part of different joint plans — for instance an agent might go to a given restaurant alone, with members of its household or with a group of friends. The score of the different copies will take into account the influence of the joint plan to which it pertains. Those joint plan constraints are included using heuristic rules, applied after mutation operators are applied, and are classified as strong or weak constraints — weak constraints are considered when selecting plans for execution, but are allowed to be broken when merely selecting plans for mutation. They are then part of the evolution process. In the current application, the heuristic rules consist in joining newly created plans with joint trips (strong) or with leisure activities at the same location at the same time (weak).

To allow handling joint plans, replanning needs to be performed for groups of agents: agents are handled with all agents with whom they have a joint plan, plus some social contacts with whom new joint plans can be created, chosen randomly among the social contacts.

For each group, two actions are then possible. For most groups, an allocation of existing plans, fulfilling the joint plans constraints, is selected for execution. Based on plan scores, randomized by adding an extreme value distributed error term, an allocation without improving coalitions is searched for by an algorithm inspired by the "Top Trading Cycle" algorithm used for the House Allocation Problem (Schummer and Vohra, 2007).

For the other groups, a plan allocation is selected and copied. The copied plans then undertake mutation, to make the agents explore new alternative joint plans. What kind of mutation is performed determines which alternative plans will be tried out by the agent. The modules used in this study are:

- Departure time mutation
- Subtour mode mutation and re-routing
- Joint trip insertion/removal and re-routing
- Swap two random activities and re-routing
- Choose new leisure location for a group of social contacts

Agents have a limited memory size, keeping at most 3 plans per joint plan composition, and 10 plans in total. If this limit is exceeded, one should keep the plans which have the highest probability to create improving coalitions, that is, to be preferred to the other plans in the agent’s
memory. To this end, a lexicographic ordering is used: the process removes the joint plan which maximizes the number of individual plans which are the worst of the agents’ memories. If several joint plans have the same number of worst plans, the process chooses among them the joint plan which maximizes the number of second worst plans, and so on until the "worst" joint plan is unique. When the overall maximum number of plans in the memory of an agent is reached, the worst individual plan for this agent is removed along with plans of other agents of the same joint plan. Each agents keeps at least one plan not part of a joint plan, as there may otherwise not be any state without blocking coalitions. Agents are parsed in random order, to avoid the emergence of "dictators" over iterations, whose worst plan would always be removed, even if it is the only "bad" plan of a joint plan.

Though those selection operators seem to be in accordance with the chosen solution concept, it is difficult, if not impossible, to prove that the process will actually converge towards the state searched. As noted by Ficici et al. (2005), when they perform a theoretical analysis of different selection methods in a co-evolutionary context, "Co-evolutionary dynamics are notoriously complex. To focus our attention on selection dynamics, we will use a simple evolutionary game-theoretic framework to eliminate confounding factors such as those related to genetic variation, noisy evaluation, and finite population size". Those "confounding factors" can however not be eliminated from an actual implementation of a co-evolutionary algorithm, and rigorously proving that a given algorithm actually implements a specific solution concept is very tedious, if not impossible.

With iterations, agents build a choice set of daily plans that becomes better and better given the actions of the other agents. However, the presence of a large portion of agents with plans resulting from random mutation creates noise, not only for the analyst looking at the output of the simulation, but for the agents themselves when they compute the score of their plans. To solve this issue, when the system reaches a stable state, agents stop performing mutation, and only select plans from their memory for 100 iterations, using the absence of improving coalition with randomized scores. This ensures that the selected plans are the result of a behavioral model, rather than the result of random mutation operators.

2.4 A Note on Performance

At the core of the co-evolutionary approach is the notion of evaluating interactions — which is done by performing a relatively expensive mobility simulation step here. In the basic case, where agents only influence each other by aggregate effects (congestion or crowding), a successful approach is to let agents optimize their plans considering those average values fixed. Several approaches were used in the past:
• **best-response** modules, that implement an optimisation algorithm on their own, and optimize the plans of part of the agents according to the state of the system in the last iteration (Meister et al., 2005a; Feil, 2010). A drawback of those approached is however the difficulty to maintain those modules and keep them modular.

• **pseudo-simulation**, where the mobility simulation periodically switches between “actual” simulation and “pseudo” simulation, where travel times come from the previous iterations (Fourie et al., 2013). This method allows speedups by allowing high parallelism, and allows to modify the plans of all the agents without introducing noise into the evaluation. Agents are *de facto* independent, and the pseudo-simulation is a totally separate software element.

However, both these approaches cannot be used as such here, because score does not only depend of aggregate effects, but also on a few selected individual plans: for a car passenger, whether the driver decides to pick him/her up or not changes everything. Due to the highest variety in possible solution concepts and the highest complexity of the problem, best-response is even more challenging. This paper demonstrate a pseudo-simulation like approach, where the agents can however still interact. This builds on the modular architecture of the mobility simulator, replacing only the “*engine*” for vehicular travel (the queue simulation). This allows to minimize the implementation and maintenance effort, and reduces greatly the risk of inconsistency, as new simulation elements — waiting of co-travelers, identification of time passed with social contacts, managing usage of joint vehicle resources — only need to be implemented once.

### 3 Results

This section focuses mainly on the computational performance of the framework, and how the proposed approach helps to tackle the computation time problem. The simulations are run on a Zurich 10% synthetic population, with two synthetic social networks generated using the approach from Arentze et al. (2013): one with a too high geographical distance between friends, and one more realistic.

Three settings are compared:

- The “classical” setting, where detailed mobility simulation is performed in every iteration
- Two variants of the pseudo-simulation approach:
  - iterating through the links of car routes to re-compute travel times given the observed travel times
  - “teleporting” agents according to their last experienced travel time, or the travel time
estimated for routing if it is the first execution.

The pseudo simulation runs alternate between pseudo simulation (only with mutated plans) and queue simulation (only with plans from the memory) every 4 iterations. The normal runs include more 70% of mutated plans in every iteration — some with only minor modifications.

Table 1 shows the average time per iteration in the different runs. First of all, both pseudo simulation settings bring improvements in terms of travel times, with the “teleported” setting resulting in almost a halved computation time. The improvements come mainly from a better usage of the replanning, which takes most of the iteration time.

Table 1: Performance statistics

<table>
<thead>
<tr>
<th>Setting</th>
<th>Average Iteration Duration (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Queue Sim.</td>
</tr>
<tr>
<td>no Pseudo Sim.</td>
<td>10.9</td>
</tr>
<tr>
<td>“Teleported” Pseudo Sim</td>
<td>3.2</td>
</tr>
<tr>
<td>“Re-estimated” Pseudo Sim</td>
<td>3.76</td>
</tr>
</tbody>
</table>

After 500 iterations, the agents only execute plans from their memory, removing the noise from innovation. Going against the hypothesis that the high mutation rate makes interaction evaluation noisy, the final average executed scores are slightly higher for the run without pseudo simulation, indicating that it actually allows to evaluate interactions better.

As for the results quality, and the capability of the framework to represent the influence of social contact spatial distribution on the trip characteristics, Fig. 2 presents the traveled distribution per mode. The realism of the social network improves a lot the realism of the traveled distances for joint travel, in particular for the “driver” mode: drivers perform much less detours when social contacts are properly located.

Fig. 3 shows the distance distribution per Origin/Destination activity type pair. The geography of the social network has here only a minimal influence on the travelled distribution.

4 Conclusion

The geography of social contacts is assumed to be an important factor influencing daily mobility, as previous studies showed that leisure activities are mainly performed for social purposes.
Figure 2: Travel Distance Distribution per Mode

Figure 3: Travel Distance Distribution per Purpose
However, current forecasting tools fail to represent this kind of phenomenon.

This paper described an approach to simulate joint leisure travel, based on the MATSim framework. Results for the Zurich area showed that it can represent the influence of social contact distribution on the characteristics of travel, and how the computation time could be reduced.

5 References


