Agent-based traffic assignment: going from trips to behavioral travelers

K. Nagel[a]

[a] Transport Systems Planning and Transport Telematics Group (VSP), TU Berlin, Germany
Correspondence address: nagel@vsp.tu-berlin.de

G. Flötteröd[b]

[b] Transport and Mobility Laboratory (TRANSP-OR), EPF Lausanne, Switzerland
Correspondence address: gunnar.floetteroed@epfl.ch

November 9, 2009

1 Introduction

Despite of the substantial progress made in activity-based demand modeling and computational simulation techniques, most dynamic traffic assignment (DTA) models still suffer from overly simplified behavioral representations. There are two major issues:

1. Consistently incorporated choice dimensions rarely go beyond route choice.
2. Decision protocols rarely go beyond stochastic equilbrium models.

Most DTA models take time-dependent origin/destination (OD) matrices as inputs and equilibrate time-dependent route flows [Peeta and Ziliaskopoulos, 2001]. Yet, they ignore the feedback of changing network conditions on higher–level choice dimensions such as departure time choice, mode choice, activity schedule choice, and such. It appears natural to extend the feedback to all choice dimensions, which also requires to consistently account for them at the assignment level.

One essential aspect of introducing more behavior into DTA models is to account for the dynamic constraints subject to which real travelers make their decisions: For example, going somewhere by car is likely to imply that later travel is done by car as well, or going shopping during a lunch break renders later shopping trips unnecessary. However, as long as the demand representation is in terms of temporally at most loosely coupled OD matrices, it discards much of the structure of real travel decisions.

Having said this, the question arises how to implement and simulate a demand model that properly accounts for general choice dimensions and constraints. A specification in terms of analytical equations exhibits desirable formal properties and enables the application of sound mathematical solution procedures [see, e.g., the supernetworks approach in Sheffi, 1985, Nagurney and Dong, 2002]. However, if the structure of the behavioral model is to be properly accounted for, the dimension of the problem to be solved increases vastly. Mathematically, this can be accounted for by increasing the number of commodities in the macroscopic model, which accounts for the likelwise increased degree of heterogeneity in the population that is naturally revealed if the demand model becomes more detailed. However, because of the combinatorial nature of all possible choices a traveler faces during a single day, the number of commodities quickly becomes computationally intractable.

At this point, microsimulation naturally enters the picture. Observing that the solution of a DTA model that comprises a large number of commodities is in fact a choice distribution over all of these commodities, rather than a vector of deterministic expectation values, Monte Carlo techniques for the realization of this distribution come to mind. Assuming without loss of generality the most disaggregate case where every single traveler constitutes one commodity, the micro-simulation of individual travel behavior can be re-interpreted as a Monte Carlo technique to draw from the underlying distributions. That is, while the microsimulation of individual behavior has an intuitive meaning, it maintains a mathematically consistent interpretation.

The transition from multi-commodity flows to microsimulation faces a symmetrical development in the field of random utility modeling: The classical multinomial logit model allows to estimate different coefficients for different segments of the population of decision makers, but the granularity of this segmentation is limited [Ben-Akiva and Lerman, 1985]. Random coefficient models overcome this confinement in that they allow for a whole distribution of behavioral parameters. However, since random coeffient models have difficult mathematical forms, their evaluation and estimation is conducted based on Monte Carlo simulation [Train, 2003].

Even the micro-simulation approach comes with a substantial computational burden. The simulation of millions of individual travelers on a detailed network requires a careful balance between modeling precision and computational efficiency, and it requires to incorporate substantial computer science and software design knowledge in order to implement operational simulation systems. By now, most of the computational problems can be considered to be solved at least at a basic level of modeling sophistication even for large-scale scenarios, and the most critical research question has become to move these solutions into a more consistent modeling framework while maintaining their favorable computational properties.

This paper starts in Sec. 2 with a discussion of how the iterative solution procedure of congested assignment models can be re-interpreted as a behavioral learning loop. This includes, as important elements, the move from continuous traffic streams to discrete individual travellers, and the inclusion of additional choice dimensions beyond route choice.

Sec. 3 then concentrates on how these concepts can be implemented in a microscopic, behaviorally-oriented ("agent-based") simulation. Most of the text concentrates on what we call *agent-based stochastic user equilibrium (SUE)*; here, the SUE formulation is traced back to its origins in that the simulated travelers are assumed to have a choice set consisting of several alternatives, and that, in every iteration, they make a deliberate, probabilistic choice from this set. It is noted that this has useful parallels with co-evolutionary computation, and in consequence algorithms and methods from that area can be used to address the agent-based SUE problem.

A regular challenge with agent-based simulations is how to set the microscopic rules such that the macroscopic outcome (sometimes called "emergence") corresponds to known or desired behavior. Sec. 4 demonstrates how this challenge can be addressed in the area of behavioral traffic simulation.

All developments in this paper assume within-day dynamic behavior, i.e. a development of the traffic and behavioral patterns along the time-of-day axis. The typical equilibrium interpretation will, however, assume that there is no within-day *replanning.* Since this is clearly an important behavioral dimension, Sec. 5 will investigate some of its consequences. The paper finishes with a conclusion in Sec. 6.

2 Equilibrium models and day-to-day replanning

The traffic assignment problem, no matter if macroscopic or microscopic, static or dynamic, trip-based or agent-based, is to identify a situation in which travel demand and travel supply (network conditions) are consistent with each other. The travel demand results from a demand model that reacts to the conditions in the network, and the network conditions are the output of a supply model (network loading model) that takes travel demand as its input. That is, a solution of the traffic assignment problem describes an equilibrium between travel demand and travel supply.

The arguably most intuitive mathematical formulation of this problem is in terms of a fixed point: Find a demand pattern that generates network conditions that in turn cause the same demand pattern to re-appear. This formulation is operationally important because it motivates a straightforward way of calculating an equilbrium by alternately evaluating the demand model and the supply model. If these iterations stabilize then a fixed point is attained that solves the traffic assignment problem.

In the following, an increasingly comprehensive specification of the traffic assignment problem is given that starts from the classical static user equilibrium model and ends with a fully dynamic model that captures arbitrary travel demand dimensions at the individual level. Computationally, the iterative fixed point solution procedure is carried throughout the entire development. Not by chance, this solution method also has a behavioral interpretation as a model of day-to-day replanning.

We start by considering route assignment only. The generalization towards further choice dimensions will turn to be a natural generalization of the route assignment problem.

2.1 Static traffic assignment

Consider a network of nodes and links, where some or all of the nodes are demand origins, denoted by o, and/or demand destinations, denoted by d. The constant demand q^{od} in origin/destination (OD) relation od splits up among a set of routes K^{od}. Denote the flow on route $k \in K^{od}$ by r_k^{od}, where $\sum_{k \in K^{od}} r_k^{od} = q^{od}$.

Most route assignment models either specify a User (Nash, Wardrop) equilibrium (UE) or a stochastic user equilibrium (SUE). A UE postulates that r_k^{od} is zero for every route k of non-minimal cost [Wardrop, 1952]:

$$c(k) = \min_{s \in K^{od}} c(s) \quad \Rightarrow \quad r_k^{od} \geq 0 \tag{1}$$

$$c(k) > \min_{s \in K^{od}} c(s) \quad \Rightarrow \quad r_k^{od} = 0 \tag{2}$$

where $c(k)$ is the cost (typically delay) on route k.

An alternative, often-used approach is to distribute the demand onto the routes such that a SUE is achieved where users have different perceptions of route cost and every user takes the route of *perceived* minimal cost [Daganzo and Sheffi, 1977]. Mathematically, this means that the route flows fulfill some distribution

$$r_k^{od} = P_k^{od}(c(x(\{r_k^{od}\}))) \cdot q^{od} \tag{3}$$

where the route splits P_k^{od} are a function of the network costs $c(x)$, which depend on the network conditions x, which in turn depend on all route flows $\{r_k^{od}\}$.

In either case, the model needs to be solved iteratively, which typically involves the following steps [Sheffi, 1985]:

Algorithm 1 Macroscopic and static route assignment

1. **Initial conditions:** Compute some initial routes (e.g., best path on empty network for every OD pair).
2. **Iterations:** Repeat the following many times.
   (a) **Network loading:** Load the demand on the network along its routes and obtain network delays (congestion).
   (b) **Choice set generation:** Compute new routes based on the network delays.
   (c) **Choice:** Distribute the demand between the routes based on the network delays.

Considering the network loading to be more on the "physical" side of the system, the behaviorally relevant steps are choice set generation and choice [Bowman and Ben-Akiva, 1998].

Choice set generation: Often, the new routes are best paths based on the last iteration ("best reply" choice set generation). The routes are generated within the iterations because an a priori enumeration of all possible routes is computationally infeasible.

Choice: Usually, demand is shifted among the routes in a way that improves consistency with the route choice model, assuming in the simplest case constant network delays: In a UE, the flow on the currently best routes is increased at the cost of the other route flows ("best reply" choice), whereas for a SUE the flows are shifted towards the desired route choice distribution [often a version of multinomial logit, e.g., Dial, 1971, Cascetta et al., 1996, Ben-Akiva and Bierlaire, 1999]. For stability reasons, this shift is typically realized in some gradual way that dampens the dynamics of the iterations. See below for more discussion on convergence issues.

The **iterations** are repeated until some stopping criterion is fulfilled that indicates that a fixed point is attained. In the best reply situation, the fixed point implies that no shift between routes takes place, i.e., what comes out as the best reply to the previous iteration is either the same or at least of the same performance as what was used in the previous iteration. Since in this situation no OD pair can unilaterally improve by switching routes, this means that the system is at a Nash equilbrium [e.g., Hofbauer and Sigmund, 1998]. In the SUE situation, the fixed point means that a route flow pattern $\{r_k^{od}\}$ is found that leads to exactly those network conditions the travelers perceived when choosing their routes, giving nobody an incentive to re-route.

Behavioral dimensions beyond route choice that can be captured by a static model are destination choice and elasticity in the demand. However, no technical generality is lost when discussing only route choice because both additional choice dimensions can be rephrased as generalized routing problems on an extended network ["supernetwork"; see, e.g., Sheffi, 1985, Nagurney and Dong, 2002].

2.2 Dynamic traffic assignment

As is well known, the above also works for *dynamic* traffic assignment [DTA; see Peeta and Ziliaskopoulos, 2001], where both the demand and the network conditions are now time-dependent and the time-dependent travel times in the network now define a physically meaningful progression of a demand unit through the network.

The structure of the algorithm does not change. The individual steps now look as follows:

Algorithm 2 Macroscopic and dynamic route assignment

1. **Initial conditions:** Compute some initial routing (e.g., best path on empty network for every OD pair and departure time).
2. **Iterations:** Repeat the following many times.
   (a) **Network loading:** Load all demand items on the network according to their departure times, let them follow their routes, and obtain network delays (congestion).
   (b) **Choice set generation:** Compute new routes based on the network delays.
   (c) **Choice:** Distribute the demand between the routes based on the network delays.

Once more, *if* the new routes are best replies (i.e., best paths based on the last iteration), *if* demand is shifted towards these new routes, and *if* these iterations reach a fixed point, then this is a dynamic UE since the best reply dynamics means that no traveler can unilaterally deviate to a better route. The SUE interpretation carries over in a similar way.

Destination choice and elasticity in the demand apply naturally to the dynamic case as well. Beyond this, the dynamic setting also enables the modeling of departure time choice. Again, the sole consideration of route choice does at least technically not constitute a limitation because departure time choice can be translated into route choice in a time-expanded version of the original network [van der Zijpp and Lindveld, 2001].

2.3 Individual travelers

Both in the static and in the dynamic case, it is possible to re-interpret the algorithm in terms of individual travelers. In the static case, for every OD pair one needs to assume a steady (= constant) flow of travelers that enter the network at the origin at a constant rate, corresponding to that OD flow. A solution to the static assignment problem corresponds to the distribution of the different travelers onto possibly different paths. In the dynamic case, one needs to generate the appropriate number of travelers for every OD pair and every time slot, and distribute them across the time slot. From then on, the triple (origin, destination, departure time) is fixed for every simulated traveler, and its goal is to find an appropriate path. Arguably, in the dynamic case this re-interpretation is behaviorally more plausible.

In a trip-based context, there are two major motivations to go from continuous flows to individual travelers:

- Traffic flow dynamics in complex network infrastructures are difficult to model in terms of continous flows but are relatively straightforward to simulate at the level of individual vehicles [TSS Transport Simulation Systems, accessed 2009, MITSIM, Quadstone Paramics Ltd., accessed 2009, PTV AG, accessed 2009]. Disaggregating an OD matrix into individual trip-makers allows to assign one vehicle to every trip-maker in the microscopic traffic flow simulation.

- As mentioned in the introduction, it is computationally inefficient to capture demand heterogeneity through a large number of commodity flows, whereas the sampling of trip-makers with different characteristics is fairly straightforward. For example, every vehicle can be given an individual route to its individual destination.

For a finite population of heterogenous travelers, every single traveler constitutes an integer commodity, and the **choice** step hence needs to be changed from "gradually shift the route flows towards something that is consistent with the behavioral model" into "for a *fraction* of travelers, assign a *single* behaviorally plausible route to each of these travelers". The gradual shift that helps to stabilize the iterations in the continuous assignment carries over here to an equally stabilizing "inert shift" in that not all travelers change their routes at once. This is a consistent reformulation: If one reduces the traveler size to $\varepsilon \to 0$ and increases the number of travelers by a factor of $1/\varepsilon$, a 10% chance of changing routes in the disaggregate case carries over to shifting 10% of all flows to new routes in the aggregate case ("**continuous limit**").

Apart from this, the iterations do not look much different from what has been said before:

Algorithm 3 Microscopic and dynamic route assignment

1. **Initial conditions:** Compute some initial routing (e.g., best path on empty network for every traveler).
2. **Iterations:** Repeat the following many times.
   (a) **Network loading:** Load all travelers on the network according to their departure times, let them follow their routes, and obtain network delays (congestion).
   (b) **Choice set generation:** Compute new routes based on the network delays.
   (c) **Choice:** Assign every traveler to a route (which can be the previously chosen one) based on the network delays.

The notions of UE and SUE carry over to the disaggregate case if the notion of an OD pair (or a commodity) is replaced by that of an individual **particle** (= microscopic traveler).

A **particle UE** may be defined as a system state where no particle can unilaterally improve itself. This definition is consistent with definitions in game theory, which normally start from the discrete problem. It should be noted, however, that this makes the problem combinatorial, which means that even a problem that had a unique solution in its continuous version may have a large number of solutions in its discrete version. That is, the particle UE is deliberately not searching for, say, an integer approximation of the continuous solution. This is structurally similar to the situation that linear programming jumps to being NP-hard when the variables are required to be integers.

As is well known, there may be situations where mixed strategy equilibria exist; these are equilibria where the participants draw between different fixed strategies randomly. This implies that the opponents need to interpret the outcome of the game probabilistically: Even if they themselves play fixed strategies, they need to maximize some expectation value.

For a **particle SUE**, the continuous limit assumption of the macroscopic model is discarded in that the choice fractions $P_k^{od}(c(x(\{r_k^{od}\})))$ in (3) are now interpreted as individual-level choice probabilities. This implies that the route flows r_k^{od} are now integer random variables, and consequently the cost structure based on which the individual choices are made becomes probabilistic as well [Balijepalli et al., 2007, Cascetta and Cantarella, 1991, Cascetta, 1989].

A particle SUE is defined as a system state where travelers draw routes from a stationary choice distribution such that the resulting distribution of traffic conditions re-generates that choice distribution.

An operational specification of a particle SUE results if one assumes that travelers filter out the random fluctuations in what they observe and base their decisions only on the average route costs:

$$P_n(k) = P_n\Big(k|E\{c(x(\{r_k^{od}\}))\}\Big) \tag{4}$$

where $P_n(k)$ now is the probability that trip-maker n selects route k and $E\{\cdot\}$ denotes the expectation.

The resulting route flows r_k^{od} represent not only the mean network conditions but also their variability due to the individual-level route sampling. Alternatively, one could use the particles merely as a discretization scheme of continuous OD flows and distribute them as closely as possible to the macroscopic average flow rates [e.g., Zhang et al., 2008]. The latter approach, however, does not lend itself to the subsequently developed behavioral model type.

No new behavioral dimensions are added when going from commodity flows to particles. However, the microscopic approach allows to simulate greater behavioral variability within the given choice dimensions because it circumvents the computational difficulties of tracking a large number of commodity flows.

2.4 Stochastic network loading

The network loading can be deterministic or stochastic. With deterministic network loading, given time-dependent route inflows, one obtains one corresponding vector of network costs. With stochastic network loading, given the same input, one obtains a *distribution* of vectors of network costs.

The macroscopic SUE approach of Section 2.1 assumes a distribution of choices but converts choice probabilities into choice fractions before starting the network loading. That is, one effectively does $NetworkLoading(E\{Choices\})$. It is, however, by no means clear that this is the same as $E\{NetworkLoading(Choices)\}$; in fact, with a non-linear network loading, even when it is deterministic, the two are different [Cascetta, 1989]. Any Monte Carlo simulation of the particle SUE makes this problem explicit: If, at the choice level, one generates draws from the choice distribution, it effectively makes sense to *first* perform the network loading and *then* do the averaging, rather than the other way around. This is especially true if day-to-day replanning is modeled where the draws from the choice distribution have a behavioral interpretation as the actual choices of the trip makers in a given day.

This, however, makes the output from the network loading effectively stochastic since the input to the network loading is stochastic. In consequence, any behavioral model that uses the traffic conditions as input needs to deal with the issue that these inputs are stochastic. For that reason, using a stochastic instead of a deterministic network loading makes little additional difference. Being able to make the network loading stochastic makes the implementation of certain network loading models simpler. In particular, randomness is a method to resolve fractional behavior in a model with discrete particles.

With stochastic network loading, additional aspects of the iterative dynamics need to be defined. For example, a "best reply" could be against the last stochastic realization or against some average.

2.5 Extending the route assignment loop to other choice dimensions

Given the above behavioral interpretation, it is now straightforward to extend the assignment loop to other choice dimensions. For example, the "best reply" can include optimal departure times [e.g. de Palma and Marchal, 2002, Ettema et al., 2003] or optimal mode choice. This becomes easiest to interpret (and, in our view, most powerful in practice) if one moves from the concept of "trips" to daily plans.

One way to denote daily plans is using an XML notation [XML]:

```
<plan>
   <activity type="home" location="123" endtime="07:23:45" ... />
   <activity type="work" location="..." endtime="..." ... />
   <activity type="shop" ... />
   ...
</plan>
```

This implies that the structure of the DTA in terms of the triple (origin, departure time, destination) is maintained. But different from the DTA, all activities are chained together.

This widens the behavioral modeling scope dramatically in that all choice dimensions of a daily travel plan can now be jointly equilibrated. This increases the number of degrees of freedom that need to be modeled, but it also brings a set of natural constraints along, which again reduce the solution space. Most noteably, the destination of one trip must be the origin of the subsequent trip of an individual, and a traveler must arrive before she/he departs. Also, constraints such as Hägerstrand's space-time prisms [Hägerstrand, 1970] are automatically enforced when the simulated travelers eventually need to return to their starting locations.

There is not much of a conceptual difference between the network loading of a route-based and a plan-based model.

The notion of a particle (S)UE can now be naturally extended to synthetic travelers (agents) that execute complete plans.

An **agent-based UE** implies individual travelers (Sec. 2.3), additional choice dimensions (Sec. 2.5), and possibly stochastic network loading (Sec. 2.4). Corresponding to the particle UE, it is defined as a system state where no agent can unilaterally improve its plan.

An **agent-based SUE** implies individual travelers (Sec. 2.3), additional choice dimensions (Sec. 2.5), and normally stochastic network loading (Sec. 2.4). Corresponding to the particle SUE, it is defined as a system state where agents draw from a stationary choice distribution such that the resulting distribution of traffic conditions re-generates that choice distribution.

If the iterations aim at an agent-based UE, then choice set generation and choice should implement a "best reply" logic in that in some sense optimal plans are calculated and assigned to the agents. This alone is by no means an easy task. The disaggregate counterpiece of a SUE implies that every agent considers a whole choice set of (possibly suboptimal) plans and selects one of these plans probabilistically, which can lead to huge data structures. Section 3 gives examples of how to deal with these difficulties.

Summarizing, we have now arrived at a dynamic DTA specification that accounts for arbitrary behavioral dimensions. Since the presentation was mostly intuitive, the introductory note on the statistical meaning of a disaggregate simulation system should be recalled at this point: It is possible to interpret the agent-based simulation as a Monte-Carlo solution procedure for a probabilistic model of plan choice behavior. However, this specification is not given explicitly but results rather implicitly in the agent-based approach from the interactions of the various sub-models.

3 Agent-based simulation

The conceptual validity of the agent-based traffic assignment model is fairly intuitive. However, since it comes with a substantial computational burden of solving the model, it brings along entirely new challenges on the simulation side.

On the demand side, there is in particular the combinatorial number of choice alternatives that needs to be accounted for. For example, random utility models rely on an a-priori enumeration of a choice set that is representative for the options every single traveler considers when making a choice [Ben-Akiva and Lerman, 1985]. This choice set is huge in the case of an agent-based simulation [Bowman and Ben-Akiva, 1998]. While there are sampling-based approaches to the modeling of large choice sets that aim at reducing this computational burden, they have not yet been carried over to the modeling of all-day-plan choices [Ben-Akiva and Lerman, 1985, Frejinger et al., 2009].

As long as household interactions are not accounted for, the demand modeling problem can be decomposed by agent once the network conditions are given, which is of great computational advantage. The supply model, on the other hand, deals with congestion, which is by definition a result of the physical interactions of all travelers. Modeling large urban areas requires to deal with millions of travelers, and an operational supply simulation must hence be able to load all of these travelers with reasonable computation time on the network.

The following sections describe how these problems can be resolved. The presentation draws heavily from the design of the MATSim simulation system [Raney and Nagel, 2006, MATSIM www page, accessed 2009], in which most of the the outlined procedures have been implemented and tested.

3.1 Agent-based UE; one plan per traveler

The simulation of an agent-based UE is possible by the following implementation of the behavioral elements.

Choice set generation: For every agent, generate what would have been best in the previous iteration. This does not only concern the route, but all considered choice dimensions, e.g., departure times and/or mode choice.

Choice: Switch to the new plan with a certain probability.

The choice set generation implements a "best reply" dynamic. This now requires to identify an optimal all-day plan for given network conditions. While the calculation of time-dependent shortest paths for UE route assignment is computationally manageable, the identification of optimal plans is far more difficult [Recker, 2001]. This constitutes an important technical motivation to switch to an agent-based SUE, where optimality is not required (see below).

Even in the manageable cases of, e.g., shortest paths, any best reply computation is an approximation. Time-dependent routing algorithms need to know every link's travel time as a function of the link entrance time. In computational practice, this information exists only in some average and interpolated way. For that reason, such computations become more robust if the performance of plans is directly taken from the network loading instead of relying on the prediction of the best reply computation, and an agent sticks with a new plan only if it *performs* better than its previous plan [Raney and Nagel, 2004]. However, in order to keep the run times maneagable, in computational practice multiple agents need to make such trial-and-error moves simultaneously. This is, therefore, not an exact best reply algorithm.

For the choice, a useful approach is to make the switching rate from the current to the best reply solution roughly proportional to the expected improvement. A possible approach is

$$P(old \to new) \propto \max[0, e^{\beta\,(S_{new} - S_{old})} - 1] \ . \tag{5}$$

where S_{new} and S_{old} are the (expected) scores of the new and the old plan, respectively. (Section 3.2.4 gives an example of how a scoring function for all-day plans could look like.) For a small difference between S_{old} and S_{new}, this can be linearly approximated by

$$e^{\beta\,(S_{new} - S_{old})} - 1 \approx \beta\,(S_{new} - S_{old}) \ , \tag{6}$$

which essentially means that $P(old \to new)$ is proportional to the magnitude of the improvement. Note how the decreasing switching *fraction* of the continuous case is replaced by a decreasing switching *rate (= probability).*

Clearly, any fixed point of such iterations is a UE since at the fixed point no switching takes place, meaning that the best reply plan has the same score as the already existing plan. The stability of the fixed point depends on the slope of the switching rate at the fixed point, in the above formulation on the β: All other things equal, making β smaller makes the fixed point more stable, but slows down convergence. These observations do not only hold in transporation [e.g., Watling and Hazelton, 2003], but quite generally in the area of "evolutionary games and dynamical systems" [Hofbauer and Sigmund, 1998]. In addition, in the context of traffic assignment, the existence of physical queues that allows for spillback across many links has been shown to be an apparently inevitable source of multiple Nash equilibria [Daganzo, 1998].

Alternatively, some MSA ("method of sucessive averages")-like scheme may be used [Liu et al., 2007]. A disadvantage is that, with MSA, the switching rate does not depend on the magnitude of the expected improvement, which possibly means slow(er) convergence. An advantage of MSA is that one does not need to find out a good value for the proportionality factor (β in the above example).

Yet another approach would be to use a "gap" function that measures the distance of the current assignment from an equilibrium and to infer the switching rate from the requirement that this function needs to be minimized [Lu et al., 2009, Zhang et al., 2008]. However, we are not aware of any operational gap function that applies to all-day plans.

The major criticism of the agent-based UE is its lack of behavioral realism. In a UE, every agent is assumed to react with a best response according to a model of its objectives, which implies that real travelers are able to compute best responses despite of their combinatorial nature and high dimension [Bowman and Ben-Akiva, 1998]. Furthermore, like for a pure route assignment, it is reasonable to assume that (i) the behavioral objective is imperfectly modeled and that (ii) explorative travel behavior leads to more or less random variations in what real travelers do. While (ii) explicitly introduces stochasticity, (i) calls for it as a representation of the imprecisions in the behavioral model.

These considerations do not only lead naturally to the agent-based SUE; they also motivate an additional behavioral component that captures the explorative learning of real travelers. Similarly to the symmetry between day-to-day replanning and the iterative solution of the traffic assigment problem, an explorative learning algorithm can be motivated either as a model of real learning or as a computational method to solve a stochastic assignment problem. The following section presents a possible implementation of such an algorithm.

3.2 Agent-based SUE; multiple plans per traveler

In order to put the proposed method for the simulation of an agent-based SUE into a somewhat broader perspective, the problem is phrased in terms of discrete choice theory [Ben-Akiva and

Lerman, 1985].

Denote by $P_n(i|C_n)$ the probability that agent n selects plan i from its choice set C_n of available plans. The analyst's possible uncertainty about the choice set motivates a stochastic specification $P_n(C_n)$ of this set. Combining these elements, one obtains the following choice distribution per agent [Manski, 1977]:

$$P_n(i) = \sum_{C_n} P_n(i|C_n) P_n(C_n). \tag{7}$$

An evaluation of this model is computationally very challenging because the sum runs over all possible subsets of the universal choice set. However, from a simulation perspective, it is sufficient to generate draws from (7). This requires two steps: First, to draw a choice set C_n for every agent n from $P_n(C_n)$, and second to make draws from $P_n(i|C_n)$ conditional on these choice sets. An additional difficulty with this procedure are the interactions among the agents through the network conditions, which do not only couple all choices but also require to build all choice sets simultaneously.

A possible implementation is to approach every traveler's daily planning problem as a population-based search algorithm. Such a search algorithm maintains a *collection* (= population) of possible solutions to a problem instance, and obtains better solutions via the evolution of that collection. This is a typical machine-learning [e.g. Russel and Norvig, 1995] approach; the best-known population-based search algorithms (also called **evolutionary algorithms**) are **genetic algorithms** [e.g., Goldberg, 1989].

It is important to note that "population" here refers to the collection of solutions for a single individual. There is also the population of travelers. Every individual uses a population-based algorithm in order to "co-evolve" in the population of all travelers [also see Balmer, 2007].

A **population-based search algorithm** typically works as follows:

Algorithm 4 Population-based search

---

1. **Initiation:** Generate a collection of candidate solutions for a problem instance.
2. **Iterations:** Repeat the following many times.
   (a) **Scoring:** Evaluate every candidate solution's "score" or "fitness".
   (b) **Selection:** Decrease the occurences of "bad" solutions. There are many ways how this can be done.
   (c) **Construction of new solutions:** Construct new solutions and add them to the collection of candidate solutions.

Regarding the construction of new solutions, two operators are often used in genetic algorithms: **Mutation** – which takes a candidate solution and performs small modifications to it; and **crossover** – which takes *two* candidate solutions and constructs a new one from those. Since mutation takes one existing solution and crossover takes two, it makes sense to also move in the opposite direction and define an operator that takes zero solutions as input, i.e., generates **solutions from scratch** – a "best-reply to last iteration" would, for example, be such an operator.

For travel behavior, solutions correspond to plans. In the XML notation from Sec. 2.5, this may look as follows:

```
<person id="321" age="25" income="60000" ... >
```

```
  <plan score="123.4">
    <activity type="home" location="123" endtime="07:23:45" ... />
    <leg mode="car">
      <route>...</route>
    </leg>
    <activity type="work" location="..." endtime="..." ... />
    <leg mode="car">
      <route>...</route>
    </leg>
    <activity type="shop" ... />
    ...
  </plan>
  <plan score="134.5">
    ...
  </plan>
</person>
```

Congruent with what has been said before, we typically have a situation where multiple travelers evolve simultaneously. That is, we have a *population of persons* where every person has a *population of plans*. The result is a co-evolutionary dynamic, where every individual person evolves according to a population-based **co-evolutionary algorithm**. The overall approach reads as follows [see, e.g., Hraber et al., 1994, Arthur, 1994, for a similar approaches]:

Algorithm 5 Co-evolutionary, population-based search

---

1. **Initiation:** Generate at least one plan for every person.
2. **Iterations:** Repeat the following many times.
   (a) **Selection/Choice:** Select, for every person, one of the plans.
   (b) **Scoring:** Obtain a score for every person's selected plan. This is done by executing all selected plans simultaneously in a simulation, and attaching some performance measure to each executed plan. Clearly, what was the network loading before has now evolved to a full-fledged person-based simulation of daily activities. See Sec. 3.2.4 for more detail on the scoring.
   (c) **Generation of new plans (innovation)/Choice set generation:** For some of the persons, generate new plans, for example as "best replies" or as mutations of existing plans (e.g. small departure time changes).

Note that this approach is really quite congruent with the SUE approach: Every person has a collection of plans, which may be interpreted as the choice set. As in SUE, the choice set may be generated while the iterations run or before the iterations start. And every person selects between the plans, where one can attach to every plan a score-based probability to be selected, in the end similar to Eq. (3). Clearly, a relevant research topic in this regards is to specify an evolutionary dynamic that can be shown to converge to choice sets that are generated consistently with the requirements of discrete choice theory.

The following subsections give examples for the different elements of this approach.

3.2.1 Selection (choice)

A possible choice algorithm is the following: For persons with unscored plans, select an unscored plan. For all other persons, select between existing plans with some SUE model, e.g. a logit model, i.e.,

$$P(i) = \frac{e^{\beta S_i}}{\sum_j e^{\beta S_j}} \tag{8}$$

where S_i is the score of plan i and β models the travelers' ability to distinguish between plans of different scores.

In practice, we have found that it is much better to not use Eq. (8) directly, but rather use a switching process that *converges* towards Eq. (8). This can, for example, be achieved by using a switching probability from i to j of

$$T(i \to j) = \gamma \, e^{\beta(S_j - S_i)/2} \ , \tag{9}$$

where i is the previous plan, j is a randomly selected plan from the same person, and γ is a proportionality constant that needs to be small enough so that the expression is never larger than one (since it denotes a probability). This works because the logit model (8) fulfills the detailed balance condition

$$P(i) \, T(i \to j) = P(j) \, T(j \to i) \tag{10}$$

for these $T(i \to j)$ [e.g., Ross, 2006].[1]

The "switching approach" has additional advantages, including the following:

- Eq. (9) can be behaviorally interpreted as the probability of switching from plan i to plan j. Plausibly, this probability increases with the magnitude of the improvement.

  For certain applications, one might desire a more involved approach, e.g., an *expected* score of j which then initiates the switch.

- One could replace Eq. (9) by a threshold-based dynamics, i.e. a switch to a better solution will only take place if the improvement is above a certain threshold. The disadvantage is that one loses some of the mathematical interpretation, but the advantage is that it may be more consistent with some discussion in project appraisal where it is said that small improvements may not lead to a change in behavior.

Although we have not done so systematically in past work, it is no problem to include formulations such as path-size logit [Ben-Akiva and Bierlaire, 1999] into the choice probabilities.

3.2.2 Score convergence

The assumption that the scores eventually converge to some constant value intuitively means that the scores cannot display spontaneous reactive behavior to a certain iteration. For example, it might be possible that a particular iteration displays "network breakdown" [Rieser and Nagel, 2008]. Converged scores would not trigger a next-day reaction to that breakdown. In practice, this can be achieved by averaging the scores over many iterations, which bears some similarity with ficticious play [Monderer and Shapley, 1996, Garcia et al., 2000]. Once more, MSA is an option, with the same advantages and disadvantages as discussed before. An alternative is to us a small **learning rate** α in

$$S_i^{new} = (1 - \alpha) \, S_i^{old} + \alpha \, \tilde{S}_i \ , \tag{11}$$

[1] Assume that, after a number of iterations, there is no more innovation, i.e., the choice set for every agent is fixed, and that the scores are updated by MSA. Upon convergence of the iterations, all agents draw their plans from a fixed choice set based on constant score expectations, cf. (4). This means that all agents make their choices independently (and that all interactions are captured in the scores). The switching logic (9) then defines an ergodic Markovian process, which converges to the unique steady state probabilities (8).

where S_i^{new} and S_i^{old} are the agent's memorized scores for option i, and $\tilde{S}_i$ is the most recent actual performance with that option. The issue, in the end, is the same as with the stable-vs-unstable fixed points: If the system is well-behaved (corresponding to a stable fixed point), it will converge benignly to constant scores and thus to the detailed balance solution. If the system is not well-behaved, one can still force it to such a solution with MSA, but the meaning of this is less clear.

As stated before, stochastic network loading makes no additional conceptual difference, since there is already stochasticity caused by the choice behavior.

3.2.3 Innovation (choice set generation)

So far, this has left open the question concerning the choice set *generation*, i.e., the part that generates new plans or modifies existing ones.

One computationally simple technique that does not require a choice set enumeration is to simulate randomly disturbed link costs and to run best response based on these costs. This, however, can yield unrealistic results if one does not get the correlation structure of the noise right.

An alternative is to calculate separate best responses after every network loading. Since the process is stochastic, this will generate different solutions from iteration to iteration. An advantage is that the correlations will be generated by the simulation – and are, thus, presumably realistic. A disadvantage is that there is currently little or no understanding how this relates to the noise specifications in random utility modeling.

Beyond that, there is really a myriad of different algorithms that could be used here. Besides the earlier-mentioned "mutation" [Balmer et al., 2005] or "crossover" [Charypar and Nagel, 2005, Meister et al., 2006], there are also many possibilities for constructive algorihms, such as "agent-based" construction [Zhu et al., 2008]. One attractive option, clearly, is to use a regular activity-based demand generation code [e.g., Bowman et al., 1998, Miller and Roorda, 2003] although our experience is that this may not be as simple as it seems [Rieser et al., 2007] since in practice activity-based models are often constructed with OD matrices in mind.

3.2.4 Adjusting the "improvement function" from shortest time to generalized utility functions

This paper takes the inductive approach of arguing that one can make the network assignment loop more general by including additional choice dimensions beyond routing. Clearly, for this to work the computation of the scoring needs to take the effects of these additional choice dimensions into account [also see Balmer, 2007]. Given evolutionary game theory, it is quite obvious how to do that: One has to extend the cost function that is used for routing to a general scoring function for complete daily plans.

That is, the performance of a daily plan needs to be scored. An established method to estimate scoring functions for different alternatives is random utility theory [e.g. Ben-Akiva and Lerman, 1985], which is why in the following, "scoring" will be replaced by "utility". For a utility function for daily plans, the following arguments may serve as starting points:

- A heuristic approach, consistent with wide-spread assumptions about travel behavior, is to give positive rewards to performing an activity and negative rewards to travelling.

- For the activities, one should select functions where the marginal reward of doing an activity decreases over time.

- Without additional effects, such as opening times or time-varying congestion, the marginal utilities of all performed activities should be the same.

MATSim has, in the past years, gained some experience with the approach described in the following paragraphs.

Total utility The total score of a plan is computed as the sum of individual contributions:

$$U_{total} = \sum_{i=1}^{m} U_{perf,i} + \sum_{i=1}^{m} U_{late,i} + \sum_{i=1}^{m} U_{tr,i} \ , \tag{12}$$

where U_{total} is the total utility for a given plan; m is the number of activities, which equals the number of trips (the first and the last activity—both of the same type and at the same location—are counted as one); $U_{perf,i}$ is the (positive) utility earned for performing activity i; $U_{late,i}$ is the (negative) utility earned for arriving late to activity i; and $U_{tr,i}$ is the (negative) utility earned for traveling during trip i.

Utility of performing an activity A logarithmic form is used for the positive utility earned by performing an activity:

$$U_{perf,i}(t_{perf,i}) = \beta_{perf} \cdot t_{*,i} \cdot \ln\left(\frac{t_{perf,i}}{t_{0,i}}\right) \tag{13}$$

where t_{perf} is the actual performed duration of the activity, t_* is the "typical" duration of an activity, and β_{perf} is the marginal utility of an activity at its typical duration:

$$U'_{perf,i}(x)|_{x=t_{*,i}} = \beta_{perf} \cdot t_{*,i} \cdot \frac{1}{x}\Big|_{x=t_{*,i}} = \beta_{perf} \ . \tag{14}$$

β_{perf} is the same for all activities since in equilibrium all activities at their typical duration need to have the same marginal utility.

At this point, both β_{perf} and $t_{*,i}$ are fixed because they are effectively only one free parameter which is split into its two components in order to simplify the interpretation.

$t_{0,i}$ is a scaling parameter. Since $\ln(t_{perf}/t_0) = \ln(t_{perf}) - \ln(t_0)$, its effect is that of shifting the curve up and down, and thus determining when it crosses the zero line. Although interpretations for this come to mind (such as a minimum duration of an activity, below which it should not be performed), we have found that this does not work well in practice. The practical problem is that one needs to control the *curvature*, i.e. the second derivative, at the typical duration, because it is the *change* of the marginal utility that determines the slack of an activity when the dayplan comes under pressure. The second derivative at the typical duration is:

$$U''_{perf,i}(x)|_{x=t_{*,i}} = \beta_{perf} \cdot t_{*,i} \cdot \frac{-1}{x^2}\Big|_{x=t_{*,i}} = -\frac{\beta_{perf}}{t_{*,i}} \ . \tag{15}$$

Since both β_{perf} and $t_{*,i}$ are already fix, it turns out that, with Eq. (13), one cannot separately control the curvature at the typical duration. At the same time, the remaining free parameter, $t_{0,i}$, controls aspects that seem comparatively less important.

Overall, a better form of the utility function needs to be found, with a functional form that minimally allows to control the first and second derivative at the typical duration. See, e.g., work by Joh [Joh et al., 2003] or by Feil [Feil et al., 2009], although we have not yet detected direct control over the second derivative in these works.

Disutility of traveling The (dis)utility of traveling is uniformly assumed as:

$$U_{tr,i} = \beta_{tr} \cdot t_{tr,i} \ , \tag{16}$$

where β_{tr} is the marginal utility (in Euro/h) for travel, and $t_{tr,i}$ is the number of hours spent traveling during trip i. β_{tr} is usually negative. Clearly, it is no problem to use other forms.

Disutility of schedule delay The (dis)utility of being late is uniformly assumed as:

$$U_{late,i} = \beta_{late} \cdot t_{late,i} \ , \tag{17}$$

where β_{late} is the marginal utility (in Euro/h) for being late, and $t_{late,i}$ is the number of hours late to activity i. β_{late} is usually negative. Once more, clearly it is no problem to use other forms.

In principle, arriving early or leaving early could also be punished. There is, however, no immediate need to punish early arrival, since waiting times are already indirectly punished by foregoing the reward that could be accumulated by doing an activity instead (opportunity cost of time). In consequence, the effective (dis)utility of waiting is already $-\beta_{perf}\, t_{*,i}/t_{perf,i} \approx -\beta_{perf}$. Similarly, that opportunity cost has to be added to the time spent traveling, arriving at an effective (dis)utility of traveling of $\beta_{tr} - \beta_{perf}\, t_{*,i}/t_{perf,i} \approx \beta_{tr} - \beta_{perf}$, where, again, β_{tr} typically is negative.

No opportunity cost needs to be added to late arrivals, because the late arrival time is spent somewhere else. In consequence, the effective (dis)utility of arriving late remains at β_{late}. – These values, β_{perf}, $\beta_{perf} - \beta_{tr}$, and $-\beta_{late}$, are the values that correspond to the parameters of the Vickrey model [Arnott et al., 1990].

3.3 Network modeling

As stated in the introduction, including additional choice dimenensions into the iterations has much to gain if this leads to additional consistency. This implies that a "behavioral" network loading model should not just take trips from a time-dependent OD matrix but should rather model the complete execution of a plan such as in Sec. 3.2.

For this, the behavioral network model needs to be microscopic, i.e., it should follow every individual agent and every individual vehicle. In addition, it should maintain the integrity of those entities, i.e., agents should only be able to depart from an activity if they are actually there (i.e., after they have shown up), and vehicles should only be able to depart from a location if they are actually there. In addition, the feedback from the network loading should be microscopic, i.e., for every agent and every vehicle there should be a trace of their actions.

In practice, it has turned out that using so-called **events** is a good means of transmitting such performance information. For the plan in Sec. 3.2, events may include the following (again using XML as a language):

```
<event type="endAct" time="07:23:45" personId="321" .../>
<event type="departure" mode="car" time="07:23:45" vehId="..." ... />
<event type="lvLink" time="..." vehId="..." .../>
<event type="enterLink" time="..." ... />
...
<event type="arrival" time="..." ... />
<event type="beginAct" actType="work" time="..." ... />
```

This is an elegant method to couple the network loading module with other modules [see, e.g. Ferber, 1999, Naumov, 2006, Mast et al., 2009].

Note that the term "microscopic" refers to the **resolution** of the model (every synthetic traveler is individually resolved), while at the same time the **fidelity** of the model can be very much reduced. The fastest implementations use simple store-and-forward mechanisms for their links [Simão and Powell, 1992, Gawron, 1998a,b, Bottom, 2000]. Although simple, such models obey, per link, flow capacity limits, speed limits, and storage limits. Since flow capacity and storage limits together cause spillback, such models are able to model physical queues. The speed of the backwards traveling kinematic congestion wave is too fast compared to reality [Simon and Nagel, 1999], but this can be corrected [Charypar, 2008]. The approach can be implemented in parallel [Cetin et al., 2003], in an event-driven way [Charypar et al., 2007a], and those two approaches can be combined [Charypar et al., 2007b]. Even without parallelization, this can be set up so that a full day of all of Switzerland (8 million inhabitants) can be simulated in about 1.5 hours [Waraich et al., 2009]; with parallel hardware, linear speed-up can be achieved [Cetin et al., 2003, Charypar et al., 2007b], dividing the runtime by the number of available CPUs.

4 Behavioral calibration

When going from aggregate OD matrices to individual agents, one also goes from smooth equations to stochastic, often rule-based systems. This complicates the mathematical perspective on the model, which arguably is a major reason for the ongoing success of the behaviorally simple yet mathematically tractable traditional assignment procedures. However, although the mathematics of iterated DTA simulations are different, they are not impossible to do Nagel et al. [1998].

Take for example the calibration of the demand. Traditionally, the four-step model would generate an OD matrix, which then is calibrated from traffic counts by approximate yet statistically motivated techniques [e.g., Cascetta and Nguyen, 1988, Cascetta et al., 1993, Ben-Akiva et al., 1998]. Even disaggregate DTA simulations such as DynaMIT or DYNASMART aggregate their individual-level demand representation into OD matrices before adjusting it to available traffic counts [Ashok, 1996, Antoniou, 2004, Zhou, 2004]. This, however, is not necessary if one carries over the mathematics of the calibration to a fully disaggregate perspective, which we demonstrate in this section.[2]

Consider the familiar problem of estimating path flows (i.e., trips) between a set of OD pairs from traffic counts [Bell et al., 1997, 1996, Sherali et al., 1994, 2003, Nie and Lee, 2002, Nie et al., 2005]. It typically is solved by the minimization of some distance measure between simulated volumes and measured traffic counts, where additional assumptions (typically a prior OD matrix) are necessary in order to resolve the ubiquitous underdetermination of the problem. In order to carry these techniques over to the calibration of an agent-based demand, one essentially needs to resolve three problems, the first two of which have both been addressed in earlier parts of this article:

1. Agent-based demand calibration deals with all-day plans, not with separate trips. → Going formally from trips to plans is straightforward if one considers plans as generalized paths on a time-expanded network.

2. Agents are integer entities and no continues streams. → Relating integer agents and continuous commodity streams requires to (i) consider every agent as a single commodity

[2] As stated in the introduction, random utility modeling went through a similar development, going from closed-form models with fairly specific assumptions to simulation-based models with a substantially wider scope.

and to (ii) observe many realizations of the agent behavior in order to again obtain a continous limit of the behavior.

3. What is the agent-based counterpiece of a prior OD matrix? → One needs to observe that agent-based simulation is possible even without any traffic counts. In a Bayesian framework, the simulation system alone provides a complete behavioral prior, which then is updated based on the traffic counts.

It is possible to walk formally through these observations and to end up with an operational formulation of the agent-based demand calibration problem [Flötteröd, 2008, 2009]:

Assume that, in a given iteration of the simulation, agent n faces the problem of drawing a plan i from its plan choice set, and assume that this choice behavior follows some choice distribution $P_n(i)$, which may be given either explicitly or through a procedural decision protocol. Now there is a set y of traffic counts that are obtained on arbitrary links and at arbitrary times in the network. In a Bayesian sense, this information can be added to the agent's behavior by specifying its posterior choice distribution

$$P_n(i|y) \sim p(y|i)P_n(i) \tag{18}$$

where $p(y|i)$ is the likelihood of the measurements y given that the agent chooses plan i. More generally, one can state the agent-based calibration problem as the problem of making all agents in the population draw jointly from their behavioral posteriors instead of their priors, which are already implemented in the plain simulation.

Although this problem can be tackled in some generality, we present, *only for illustration*, a specific solution that relies on the following assumptions:

- The plan choice model is multinomial logit.
- The traffic counts are independently normally distributed.
- The network is only lightly congested.[3]

In this case, it is possible to arrive at the following solution to the problem:

$$P_n(i|y) \sim \exp\left(V_n(i) + \sum_{at \in i} \frac{y_a(t) - \bar{q}_a(t)}{\sigma_a^2(t)}\right). \tag{19}$$

where $V_n(i)$ is the systematic utility agent n assigns to plan i, $at \in i$ indicates that plan i implies to cross the sensor on link a in time step t, $y_a(t)$ is the measured traffic count on link a in time step t, $\bar{q}_a(t)$ is the average simulated volume on link a in time step t, and $\sigma_a^2(t)$ is the variance of the normal likelihood of the respective measurement.

Intuitively, this algorithm works like a controller that steers the agents towards a reasonable fulfillment of the measurements: For any sensor-equipped link, the utility addend is positive if the measured flow is higher than the simulated flow such that the utility of plans that cross this link is increased. Vice versa, if the measured flow is lower than the simulated flow, the according utility addend is negative such that plans that cross this link are penalized.

This approach has proven very powerful in practice, and first results for a large real-world application are available [Flötteröd et al., 2009]. That is, the agent-based approach has caught

[3] This assumption allows to treat the network loading as a linear mapping. In congested conditions, a more involved linearization of the network loading is necessary. However, this can be realized in a computationally very efficient way [Flötteröd and Bierlaire, 2009].

up with the traditional assignment procedures with respect to the calibration of the demand, and the methodological progress that made this possible is likely to carry over to other fields where, up to now, traditional methods had an edge over microsimulations because of their better understood mathematics.

5 Within-day replanning

Up to this point, only equilibrium models were considered, where every traveler is essentially assumed to have full and predictive information about the network conditions from many previous days of experience. The iterative fixed point solution procedure for the calculation of such an equilibrium clearly can be regarded as an explicit implementation of this learning procedure, although it can be equally motivated from a purely computational point of view.

Equilibrium models apply on large time scales to stable transportation systems, but they are inadequate to capture any transients in the system's day-to-day dynamics that result from the learning process of not yet well-informed travelers that only in the limit leads to the postulated equilibrium. Another important aspect of these transients is that travelers encounter situations that are different from what they expect, which can trigger spontaneous replanning within a given day.

From a modeling perspective, within-day replanning is a relatively recent field. Structurally, it does not differ from any other demand model, only that the choices are conditioned on attributes of the situation that do not necessarily reflect that situation but rather the traveler's belief abut it. This, however, implies that the replanning process and the whole-day network loading of an assignment are now intertwined.

5.1 Day-to-day vs. within-day models

The two most important temporal dimensions in DTA are within-day dynamics and day-to-day dynamics. Within-day dynamics refer to the temporal variability of demand and network conditions for a given day. Day-to-day dynamics capture the evolution of the system over many days. Doubly-dynamic models capture both aspects in one model.

The vast majority of **within-day dynamic** assignment models constrain themselves to traffic flow dynamics but do not account for **within-day replanning**. This is reasonable for a stable transportation system in which all travelers experience more or less what they have expected when planning their trips a priori. If, however, there are unforseeable events, then there will be within-day replanning that needs to be modeled.

The defining element of **day-to-day dynamics** is the **day-to-day replanning** of travelers, which is driven by a learning process of exploring the transportation system day by day, collecting information, and (re)considering travel decisions based on this information. Day-to-day models are not specific to DTA and can also be linked with static network models that go without within-day dynamics [Cascetta, 2001, 1989]. For day-to-day models, the notion of an equilibrium needs to be replaced by that of a limiting system behavior after a large number of days, which corresponds to a stationary distribution in case of stochastic dynamics or to a fixed point if the system is deterministic and well-behaved [Watling and Hazelton, 2003].

A **doubly-dynamic assignment** combines a day-to-day replanning logic with a dynamic within-day network model, but it does not necessarily capture within-day replanning [Balijepalli et al., 2007, Cascetta and Cantarella, 1991].

A **doubly-dynamic replanning model** is one where the replanning itself does not only happen on a day-to-day basis but also within-day. This type of model has received little attention in the literature, arguably because it calls for substantial modeling efforts. However, considering that one main advantage of day-to-day assignment models is their ability to model transients in the network evolution during which travelers are imperfectly informed, it is plausible to assume that this imperfect information also implies that travelers replan spontaneously based on en-trip gathered information.

5.2 Inserting the within-day replanning into the day-to-day assignment

Within-day replanning requires to specify the initial information an agent posesses when starting its day. Adopting a black & white perspective on the problem this can either be information about the expected network conditions or a set of alternative plans for the given day.

Replanning only based on expected network conditions is computationally convenient because of the *relatively* small amount of information that needs to be stored per agent (and, in addition, this information could be shared by more than one agent). Replanning only in terms of the selection of a priori specified plans suffers from the need to define the plans such that they overlap at well-defined switching points, which may be numerous and hence call for a large number of plans.

A more general perspective would be to steer the within-day replanning by a *strategy* [Axhausen, 1988, 1990], which would be defined in terms of a mapping of network conditions on (to be) executed plans. This comprises the previous approaches in that both expected network conditions and a priori generated choice sets constitute parameterizations of this strategy.

Whatever the strategy, the result of a simulated day are the executed plan of every traveler, the experienced network conditions, and some assessment of the plan's performance. This is exactly the information that needs to be fed into the learning model of a day-to-day replanning system. The day-to-day replanning model, however, now needs to be generalized into a model that defines a *strategy* for the next day. The arguably simplest approach would be here to keep the decision protocol of the strategy fixed over the days and to update only the expected network conditions on which it is based.

5.3 Direct simulation

Computationally, a direct simulation of within-day replanning itself, is, at least structurally, less involved than an equilibrium model. The reason is that it can be solved forwards through simulated time, which is straightforward to implement at least in principle.

A corresponding, naive implementation in a time-stepped simulation would be to not only go through all network elements during a time step, but also through all simulated travelers, to compute their perceptions based on the information that they receive (own observations, radio broadcast, specific messages), and from that to compute their decisions.

In addition, if one attaches the behavioral algorithms directly to the agent, this becomes rather easy to parallelize and in consequence may run extremely fast. The disadvantage would be less modularity of software design, and possible effects of so-called race conditions, where different program threads run at different speeds depending on the state of the parallel computer, e.g., by loads from other processes.

5.4 Emulating within-day replanning by iterative simulation

Despite of the stronger coupling between network loading and within-day replanning, the model can still be solved in an iterative manner that is consistent with the fixed point simulation approach for the equilibrium model considered so far. It still is possible to iterate between a demand simulator and a supply simulator in a setting where every agent chooses a whole plan before the network loading, and where the whole plan is executed without replanning during the network loading.

The difference between an equilibrium model and a within-day model is that an equilibrium demand model can utilize all information from the most recent network loading(s), whereas a within-day demand model generates every elementary decision of a plan only based on such information that could have actually been gathered up to the according point in simulated time. That is, when an agent replans in the iterative within-day demand model, it still builds its plan incrementally along the time axis, where the information on which it bases its elementary decisions is "revealed" as time increases.

An important plus of the iterative solution is that it re-establishes the separation of replanning and network loading, which helps to deal with the simulation system as a whole. For example, an operational solution to the self-consistent route guidance problem, which only makes sense in non-equilbrium conditions, is only available for an indirect formulation of the within-day replanning problem [Bottom, 2000, Bottom et al., 1999]. Similarly, the real-time tracking of travel behavior from traffic counts is enabled by an iterative solution of the within-day assignment model [Flötteröd, 2008].

6 Conclusion

This paper investigates how behavioral considerations can be integrated into the modeling of network dynamics. Starting from regular route assignment, the paper points out that one can extend the iterative solution procedure of static or dynamic traffic assignment to include additional behavioral dimensions such as time adaptation, mode choice or secondary activity location choice. This is somewhat similar to the so-called supernetworks approach, but argues from the viewpoint of the iterative solution procedure rather than from the viewpoint of the problem definition.

In order to address the combinatorial explosion of the commodities caused by the expansion of the choice dimensions, it is suggested to move to individual particles. This allows an interpretation of the solution procedure as behavioral day-to-day learning, but maintains a connection to the SUE definition by interpreting the synthetic travellers' behavior as random draws from individual choice sets. In that latter interpretation, the iterative solution procedure becomes a Monte Carlo simulation that samples from the population's choice distribution.

A major part of the paper discusses simulation/computer implementation issues. From the definition given above, progress can be made by using methods from machine learning and co-evolutionary search algorithms. The SUE problem of random selection between different alternatives can be cast as a so-called population-based optimization algorithm where every synthetic traveler randomly selects between the different members of the population of possible solutions. At the same time, the *population of the travelers* co-evolves towards a stationary distribution of choices.

A separate section discusses how such an approach can be calibrated to real-world measurements in the same way as this is possible with calibration procedures in more conventional assignment. It turns out that it is possible to use a Bayesian interpretation of the choice behavior in order to systematically modify the choices of the simulated agents according to the measurements.

Overall, it has been clear for some time now that it is possible to simulate large transportation systems microscopically, including many learning iterations with choice dimensions beyond route choice, and this paper describes some of the necessary methods and techniques. Future research will have to fill the gap between the computationally efficient yet behaviorally simplified approaches that have by now been demonstrated to be applicable to large real-world scenarios and the far more sophisticated yet less operational behavioral models proposed in the travel demand literature.

7 Acknowledgements

This paper derives much from the MATSim project, which is a collaboration between many people and institutions. For the issues investigated in this paper, we benefited from discussions with M. Bierlaire, A. Borning, D. Fox, P. Henry, D. Layton, F. Marchal, F. Martinez, and P. Waddell.

References

C. Antoniou. *On-line Calibration for Dynamic Traffic Assignment*. PhD thesis, Massachusetts Institute of Technology, 2004.

R. Arnott, A. de Palma, and R. Lindsey. Economics of a bottleneck. *Journal of Urban Economics*, 27(1): 111–130, 1990.

B. Arthur. Inductive reasoning, bounded rationality, and the bar problem. *American Economic Review (Papers and Proceedings)*, 84:406–411, 1994.

K. Ashok. *Estimation and Prediction of Time-Dependent Origin-Destination Flows*. PhD thesis, Massachusetts Institute of Technology, 1996.

K.W. Axhausen. A simultaneous simulation of activity chains. In P.M. Jones, editor, *New Approaches in Dynamic and Activity-based Approaches to Travel Analysis*, pages 206–225. Avebury, Aldershot, 1990.

K.W. Axhausen. *Eine ereignisorientierte Simulation von Aktivitätenketten zur Parkstandswahl*. PhD thesis, Universität Karlsruhe, Germany, 1988.

N.C. Balijepalli, D.P. Watling, and R. Liu. Doubly dynamic traffic assignment – simulation modeling framework and experimental results. *Transportation Research Record*, 2029:39–48, 2007.

M. Balmer. *Travel demand modeling for multi-agent transport simulations: Algorithms and systems*. PhD thesis, Swiss Federal Institute of Technology (ETH) Zürich, Switzerland, 2007.

M. Balmer, B. Raney, and K. Nagel. Adjustment of activity timing and duration in an agent-based traffic flow simulation. In H.J.P. Timmermans, editor, *Progress in activity-based analysis*, pages 91–114. Elsevier, Oxford, UK, 2005.

M.G.H. Bell, W.H.K. Lam, and Y. Iida. A time-dependent multi-class path flow estimator. In J.-B. Lesort, editor, *Proceedings of the 13th International Symposium on Transportation and Traffic Theory*, pages 173–193, Lyon, France, July 1996. Pergamon.

M.G.H. Bell, C.M. Shield, F. Busch, and G. Kruse. A stochastic user equilibrium path flow estimator. *Transportation Research Part C*, 5(3/4):197–210, 1997.

M. Ben-Akiva and M. Bierlaire. Discrete choice methods and their applications to short-term travel decisions. In R. Hall, editor, *Handbook of Transportation Science*, pages 5–34. Kluwer, 1999.

M. Ben-Akiva, M. Bierlaire, H. Koutsopoulos, and R. Mishalani. DynaMIT: a simulation-based system for traffic prediction. Delft, The Netherlands, February 1998. DACCORD Short Term Forecasting Workshop.

M.E. Ben-Akiva and S.R. Lerman. *Discrete Choice Analysis*. MIT Press series in transportation studies. The MIT Press, 1985.

J. Bottom. *Consistent anticipatory route guidance*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2000.

J. Bottom, M. Ben-Akiva, M. Bierlaire, I. Chabini, H. Koutsopoulos, and Q. Yang. Investigation of route guidance generation issues by simulation with DynaMIT. In A. Ceder, editor, *Proceedings of the 14th International Symposium on Transportation and Traffic Theory*, pages 577–600. Pergamon, Jerusalem, Israel, July 1999.

J.L. Bowman and M.E. Ben-Akiva. Activity based travel demand model systems. In P. Marcotte and S. Nguyen, editors, *Equilibrium and advanced transportation modelling*, pages 27–46. Kluwer, 1998.

J.L. Bowman, M. Bradley, Y. Shiftan, T.K. Lawton, and M. Ben-Akiva. Demonstration of an activity-based model for Portland. In *World Transport Research: Selected Proceedings of the 8th World Conference on Transport Research 1998*, volume 3, pages 171–184. Elsevier, Oxford, 1998.

E. Cascetta. A stochastic process approach to the analysis of temporal dynamics in transportation networks. *Transportation Research Part B*, 23(1):1–17, 1989.

E. Cascetta. *Transportation Systems Engineering: Theory and Methods*. Kluwer Academic Publishers, 2001.

E. Cascetta and G.E. Cantarella. A day-to-day and within-day dynamic stochastic assignment model. *Transportation Research Part A*, 25(5):277–291, 1991.

E. Cascetta and S. Nguyen. A unified framework for estimating or updating origin/destination matrices from traffic counts. *Transportation Research Part B*, 22(6):437–455, 1988.

E. Cascetta, D. Inaudi, and G. Marquis. Dynamic estimators of origin-destination matrices using traffic counts. *Transportation Science*, 27:363–373, 1993.

E. Cascetta, A. Nuzzolo, F. Russo, and A. Vitetta. A modified logit route choice model overcoming path overlapping problems. Specification and some calibration results for interurban networks. In J.-B. Lesort, editor, *Proceedings of the 13th International Symposium on Transportation and Traffic Theory*, pages 697–711, Lyon, France, July 1996. Pergamon.

N. Cetin, A. Burri, and K. Nagel. A large-scale agent-based traffic microsimulation based on queue model. In *Proceedings of Swiss Transport Research Conference (STRC)*, Monte Verita, CH, 2003. See www.strc.ch. Earlier version, with inferior performance values: Transportation Research Board Annual Meeting 2003 paper number 03-4272.

D. Charypar. *Efficient algorithms for the microsimulation of travel behavior in very large scenarios*. PhD thesis, Swiss Federal Institute of Technology (ETH) Zürich, Switzerland, 2008.

D. Charypar and K. Nagel. Generating complete all-day activity plans with genetic algorithms. *Transportation*, 32(4):369–397, 2005.

D. Charypar, K.W. Axhausen, and K. Nagel. Event-driven queue-based traffic flow microsimulation. *Transportation Research Record*, 2003:35–40, 2007a. doi: 10.3141/2003-05.

D. Charypar, K.W. Axhausen, and K. Nagel. An event-driven parallel queue-based microsimulation for large scale traffic scenarios. In *Proceedings of the World Conference on Transport Research*, Berkeley, CA, 2007b.

C.F. Daganzo. Queue spillovers in transportation networks with a route choice. *Transportation Science*, 32(1): 3–11, 1998.

C.F. Daganzo and Y. Sheffi. On stochastic models of traffic assignment. *Transportation Science*, 11(3):253–274, 1977.

A. de Palma and F. Marchal. Real case applications of the fully dynamic METROPOLIS tool-box: An advocacy for large-scale mesoscopic transportation systems. *Networks and Spatial Economics*, 2(4):347–369, 2002.

R.B. Dial. A probabilistic multipath traffic assignment model which obviates the need for path enumeration. *Transportation Research*, 5:83111, 1971.

D. Ettema, G. Tamminga, H. Timmermans, and T. Arentze. A micro-simulation model system of departure time and route choice under travel time uncertainty. In *Proceedings of the meeting of the International Association for Travel Behavior Research (IATBR)*, Lucerne, Switzerland, 2003. See www.ivt.baug.ethz.ch.

M. Feil, M. Balmer, and K.W. Axhausen. New approaches to generating comprehensive all-day activity-travel schedules. Arbeitsbericht Verkehrs- und Raumplanung 575, IVT, ETH Zurich, Zurich, Switzerland, 2009. URL `www.ivt.ethz.ch`.

J. Ferber. *Multi-agent systems. An Introduction to distributed artificial intelligence*. Addison-Wesley, 1999.

G. Flötteröd. Cadyts – a free calibration tool for dynamic traffic simulations. In *Proceedings of the 9th Swiss Transport Research Conference*, Monte Verita/Ascona, September 2009.

G. Flötteröd. *Traffic State Estimation with Multi-Agent Simulations*. PhD thesis, Berlin Institute of Technology, Berlin, Germany, 2008.

G. Flötteröd and M. Bierlaire. Improved estimation of travel demand from traffic counts by a new linearization of the network loading map. In *Proceedings of the European Transport Conference*, The Netherlands, October 2009.

G. Flötteröd, Y. Chen, M. Rieser, and K. Nagel. Behavioral calibration of a large-scale travel behavior microsimulation. In *Proceedings of 12th International Conference on Travel Behaviour Research*, Jaipur, India, December 2009.

E. Frejinger, M. Bierlaire, and M. Ben-Akiva. Sampling of alternatives for route choice modeling. *Transportation Research Part B*, 43(10):984–994, 2009.

A. Garcia, D. Reaume, and R.L. Smith. Fictitious play for finding system optimal routings in dynamic traffic networks. *Transportation Research Part B*, 34(2):147–156, 2000.

C. Gawron. An iterative algorithm to determine the dynamic user equilibrium in a traffic simulation model. *International Journal of Modern Physics C*, 9(3):393–407, 1998a.

C. Gawron. *Simulation-based traffic assignment*. PhD thesis, University of Cologne, Cologne, Germany, 1998b. URL `www.zaik.uni-koeln.de/AFS/publications/theses.html`.

D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

T. Hägerstrand. What about people in Regional Science? *Papers in Regional Science*, 24:6–21, 1970. doi: 10.1007/BF01936872.

J. Hofbauer and K. Sigmund. *Evolutionary games and replicator dynamics*. Cambridge University Press, 1998.

P. Hraber, T. Jones, and S. Forrest. The ecology of Echo. *Artificial Life*, 3(3):165–190, 1994.

C.H. Joh, T. Arentze, and H. Timmermanns. Estimating non-linear utility functions of time use in the context of an activity schedule adaptation model. In *Proceedings of the meeting of the International Association for Travel Behavior Research (IATBR)*, Lucerne, Switzerland, 2003. See www.ivt.baug.ethz.ch.

H.X. Liu, X. He, and B. He. Method of successive weighted averages (MSWA) and self-regulated averaging schemes for solving stochastic user equilibrium problem. *Networks and Spatial Economics*, 2007. to appear.

C.-C. Lu, H.S. Mahmassani, and X. Zhou. Equivalent gap function-based reformulation and solution algorithm for the dynamic user equilibrium problem. *Transportation Research Part B: Methodological*, 43(3):345 – 364, 2009. doi: DOI:10.1016/j.trb.2008.07.005.

C. Manski. The structure of random utility models. *Theory and Decision*, 8:229–254, 1977.

F. Mast, K. Lunde, P. Biggel, L. Kieble, and C. Schlegel. Simulating communication in a satellite-based toll collection system. In *OMNeT++*, 2009.

MATSIM www page. MultiAgent Transport SIMulation. http://matsim.org/, accessed 2009. URL `www.matsim.org`.

K. Meister, M. Balmer, K. Axhausen, and K. Nagel. planomat: A comprehensive scheduler for a large-scale multi-agent transportation simulation. In *Proceedings of the meeting of the International Association for Travel Behavior Research (IATBR)*, Kyoto, Japan, 2006. See www.iatbr.org.

E.J. Miller and M.J. Roorda. A prototype model of household activity/travel scheduling. *Transportation Research Record*, 1831:114–121, 2003.

MITSIM. MITSIM web site. http://mit.edu/its/mitsimlab.html, accessed 2009.

D. Monderer and L.S. Shapley. Fictitious play property for games with identical interests. *Journal of Economic Theory*, 68(1):258–265, 1996.

K. Nagel, M. Rickert, P.M. Simon, and M. Pieck. The dynamics of iterated transportation simulations. In *Proceedings of the 3rd Triennial Symposium on Transportation Analysis*, San Juan, Puerto Rico, 1998.

A. Nagurney and J. Dong. *Supernetworks: Decision-Making for the Information Age*. Edward Elgar, London, 2002.

V. Naumov. *Routing in large vehicular ad hoc networks*. PhD thesis, Swiss Federal Institute of Technology (ETH) Zurich, 2006.

Y. Nie and D.-H. Lee. An uncoupled method for the equilibrium-based linear path flow estimator for origin-destination trip matrices. *Transportation Research Record*, 1783:72–79, 2002.

Y. Nie, H.M. Zhang, and W.W. Recker. Inferring origin-destination trip matrices with a decoupled GLS path flow estimator. *Transportation Research Part B*, 39(6):497–518, 2005.

S. Peeta and A.K. Ziliaskopoulos. Foundations of dynamic traffic assignment: the past, the present and the future. *Networks and Spatial Economics*, 1(3/4):233–265, 2001.

PTV AG. VISSIM web site. http://www.ptvag.com/, accessed 2009.

Quadstone Paramics Ltd. Paramics web site. http://www.paramics-online.com, accessed 2009.

B. Raney and K. Nagel. Iterative route planning for large-scale modular transportation simulations. *Future Generation Computer Systems*, 20(7):1101–1118, 2004.

B. Raney and K. Nagel. An improved framework for large-scale multi-agent simulations of travel behavior. In P. Rietveld, B. Jourquin, and K. Westin, editors, *Towards better performing European Transportation Systems*, pages 305–347. Routledge, 2006.

W.W. Recker. A bridge between travel demand modeling and activity-based travel analysis. *Transportation Research Part B*, 35(5):481–506, 2001.

M. Rieser and K. Nagel. Network breakdown "at the edge of chaos" in multi-agent traffic simulations. *European Journal of Physics*, 63(3):321–327, June 2008. doi: 10.1140/epjb/e2008-00153-6.

M. Rieser, K. Nagel, U. Beuck, M. Balmer, and J. Rümenapp. Truly agent-oriented coupling of an activity-based demand generation with a multi-agent traffic simulation. *Transportation Research Record*, 2021:10–17, 2007. doi: 10.3141/2021-02.

S.M. Ross. *Simulation*. Elsevier, fourth edition, 2006.

S.J. Russel and P. Norvig. *Artificial intelligence: A modern approach*. Series in Artificial Intelligence. Prentice Hall, 1995.

Y. Sheffi. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice-Hall, 1985.

H.D. Sherali, R. Sivanandan, and A.G. Hobeika. A linear programming approach for synthesizing origin-destination trip tables from link traffic volumes. *Transportation Research Part B*, 28(3):213–233, 1994.

H.D. Sherali, A.Narayan, and R. Sivanandan. Estimation of origin-destination trip-tables based on a partial set of traffic link volumes. *Transportation Research Part B*, 37(9):815–836, 2003.

H.P. Simão and W.B. Powell. Numerical methods for simulating transient, stochastic queueing networks. *Transportation Science*, 26:296–311, 1992.

P.M. Simon and K. Nagel. Simple queueing model applied to the city of Portland. Paper 99-0861, Transportation Research Board Annual Meeting, Washington, D.C., 1999.

K.E. Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, 2003.

TSS Transport Simulation Systems. AIMSUN web site. http://www.aimsun.com, accessed 2009.

N.J. van der Zijpp and C.D.R. Lindveld. Estimation of origin-destination demand for dynamic assignment with simultaneous route and departure time choice. *Transportation Research Record*, 1771:75–82, 2001.

R. Waraich, D. Charypar, M. Balmer, and K. Axhausen. Performance improvements for large scale traffic simulation in matsim. Technical report, ETH Zürich, IVT, 2009.

J.G. Wardrop. Some theoretical aspects of road traffic research. *Proceedings of the Institute of Civil Engineers*, 1:325–378, 1952.

D. Watling and M.L. Hazelton. The dynamics and equilibria of day-to-day assignment models. *Networks and Spatial Economics*, 3(3):349–370, 2003.

XML. Extensible Markup Language XML web site. http://www.w3.org/XML/, accessed 2009.

K. Zhang, H.S. Mahmassani, and C.-C. Lu. Probit-based time-dependent stochstic user equilibrium traffic assignment model. *Transportation Research Record*, 2085:86–94, 2008.

X. Zhou. *Dynamic Origin-Destination Demand Estimation and Prediction for Off–Line and On–Line Dynamic Traffic Assignment Operation*. PhD thesis, University of Maryland, College Park, 2004.

S. Zhu, D.M. Levinson, and L. Zhang. Agent-based route choice with learning and exchange of information. Paper 08-2152, Transportation Research Board Annual Meeting, Washington, D.C., 2008.