Relaxation-discretization algorithm for spatially constrained secondary location assignment

Author(s):
Hörl, Sebastian; Axhausen, Kay W.

Publication date:
2020-01

Permanent link:
https://doi.org/10.3929/ethz-b-000378016

Relaxation-discretization algorithm for spatially constrained secondary location assignment

Date of submission: 2019-08-01

Sebastian Hörl
IVT, ETH Zürich, 8093 Zürich, Switzerland
phone: +41-44-633-38-01
email: sebastian.hoerl@ivt.baug.ethz.ch
orcid: 0000-0002-9018-432X

Kay W. Axhausen
IVT, ETH Zürich, 8093 Zürich, Switzerland
phone: +41-44-633-39-43
email: axhausen@ivt.baug.ethz.ch
orcid: 0000-0003-3331-1318

Words: 7000 words + 2 tables = 7500 word equivalents

ABSTRACT

Agent-based transport models demand that the daily activity patterns of artificial agents are described in great detail. While established choice models for residential locations or work places exist, only few approaches are available to find locations for highly constrained secondary activities such as grocery shopping or recreation. The paper describes a novel data-driven approach of assigning viable locations to such secondary locations while maintaining consistency with homes, work places and other fixed points in an artificial traveller's daily plan. Two use cases for Switzerland and Île-de-France are presented which show that the algorithm is able to assign locations while providing realistic distance distributions that are consistent with mode-specific travel times.

INTRODUCTION

In recent years, agent-based transport models have gained large interest, not only from researchers, but also from practicioners. Main drivers of this development are cheap computing power which allows for large-scale simulations with millions of agents and an ever growing amount of transport data.

Still, setting up agent-based transport models involves considerable amounts of work. Contrary to more aggregate approaches, the attributes, intentions and interaction between a large number of indiviudal travellers need to be modelled. While for many dimensions useful data exists, such as census data to determine home locations of agents, commuter matrices to assign work places, or household travel surveys to describe daily mobility schedules, there are still gaps. One major unknown are usually locations of secondary activities, i.e. where people go shopping, engage in leisure or eat. A reason for that is that such choices are much more rich and detailed than residential or work place choices, which can often be derived from macroeconomic principles.

Literature on *residential* location choice is vast and mostly related to discrete choice modelling *(1–4)*. Likewise, models such as the gravity model have emerged as standard procedures for assigning work or education locations that resemble well daily commuting patterns *(5–8)*. Also, models such as *(9)* have been presented for capacitated work location choice. Unfortunately, these approaches are difficult to apply to *secondary* locations, because they often "fill" gaps between the *primary* home, work and education activities of people. Therefore, they are much more constrained in terms of time to reach those locations, but depend highly on individual taste variations, and are among hundred and thousands of alternatives.

Some discrete choice models have been proposed that give insight into the choice behaviour for certain, very specific activity types given certain attributes of locations or zones. There are examples for shopping activities *(10)* and recreational activities *(11)*.

The problem of secondary location choice seems to be a challenge that is inherent to agent- and activity-based models, because often not only peak hour commuter traffic is considered, but whole day mobility patterns. Furthermore, discrete locations are considered rather than aggregate zones. Since such models have only gained wide-speard interest in recent years, literature on secondary location choice is scarce and no standard approach has emerged so far. Yet, a search for *secondary location choice* or *destination choice* yields a number of various approaches that are linked to activity-based modelling. For instance, ALBATROSS *(12, 13)* and TASHA *(14)* each apply different strategies of implementing location choices into their activity scheduling frameworks by different heuristic means of reducing the available choice set.

In the context of the agent-based transport simulation framework MATSim *(15)* efforts have been pushed to put location choice into its evolutionary model of learning promising daily plan alternatives. *(16)* consider a limited agent memory of known facilities for secondary locations, while *(17)* explore the use of the concept of "frozen randomness" which applies constant error terms to the attractivity of each possible secondary activity location. Again, the approaches tries to solve the problem secondary location choice by defining limited search spaces to cope with the vast amount of options.

In this paper we describe a new approach for finding viable locations for secondary activities that is data-driven instead of trying to establish a behavioural or structural choice model. In our use case we consider daily activity chains with fixed primary activities and variable secondary activities in between. We seek to assign locations to those secondary activities from a predefined set of discrete locations such that an acceptable fit with a reference distance distribution is achieved. Furthermore, we require that expected transport modes and travel times, which are known a priori from the activity chains, are consistent with the distances that emerge from newly assigned secondary activity locations. This way a good starting solution for the full mobility simulation is provided.

The remaining part of the paper is structured as follows. First, we describe our method in detail. By that we try to formalize the approach in a rather generic way and show path ways for future research and a potentially more closed form treatment of the procedure. Afterwards, we present results for two large-scale agent-based simulation models of Switzerland and Île-de-France, followed by a discussion of our approach and concluding remarks.

METHOD

The algorithm that is presented in the following section operates on chains of activities which are connected by trips. Some activities already have a location in space assigned. We define those as *fixed activities*. The algorithm has the purpose to find sensible locations for all other activities, which we call *variable activities*. For instance, a typical activity chain in agent-based transport modelling would have a fixed home location for each agent and its work place may be known from a separate commuting destination model. In such a case it remains to determine where an agent would perform secondary activities such as shopping or leisure.

The distinction between fixed and variable activities allows us to split up a whole activity chain into smaller *assignment problems*, which can be classified into two types. The first type is a *one-sided constraint problem* as is shown in Figure 1 on the left. These problems appear generally at the start and end of an activity chain, for instance, when an agent comes home on Monday from a weekend leisure activity. Note that most transport models even specify that agents need to start and end their activity chain at home. In those cases the one-sided constraint problem is not relevant.

The second assignment problem type is the *two-sided constraint problem*. This problem is the main focus of this work and is defined by two fixed activity locations with an arbitrary number of variable activities between them. The task is then to find locations for those variable activities such that certain criteria are met. Our criteria, which are detailed below, make sure that the algorithm produces realistic distance distributions.

In any case, the *assignment problem* does not only consist of finding *continuous* locations in Euclidean space for all variable activities, but to select candidates from a given set of *discrete locations*. Such discrete location are generally known upfront, e.g. as a list of all shops in city. Furthermore, the assignment process may rely on additional information about the activities in the chain and attributes on their connecting trips. This way, a certain type of activity may demand that it is assigned to a discrete location where such an activity can be performed. Likewise, a known mode of transport on a certain trip may restrict the distance between two activities.

To solve the assignment problem, we propose a two-step algorithm. In the first step, the *relaxation problem* is solved. Its purpose is to find viable locations for all variable activities in continuous Euclidean space. Afterwards, the *discretization problem* is solved in the second step. There, candidates are chosen from the set of discrete locations and assigned to the variable activities. The

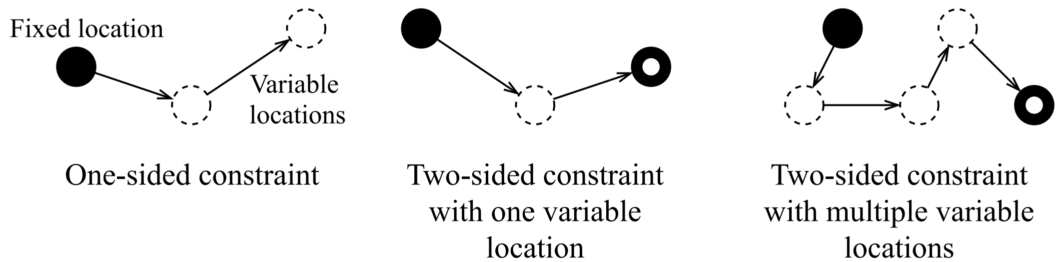

FIGURE 1 One-sided constraint and two-sided constraint assignment problems

result of the relaxation problem has strong influence on this choice process. Finally, a convergence metric tests whether the algorithm should start again with the relaxation phase or can terminate for a certain assignment problem.

More specifically, each round of relaxation and discretization should yield a certain objective value $J(\cdot)$. This objective value is then used to determine whether the algorithm can termine. It also ends if a certain number of iterations has been reached. In this case the solution with the best objective value so far is returned. The procedure of solving the assignment problem is summarized in Algorithm 1.

ALGORITHM 1: Assignment Problem Solver

```
Input: AssignmentProblem
Initialize: BestSolution = Null
Do:
    RelaxedSolution = SolveRelaxationProblem(AssignmentProblem)
    DiscretizedSolution = SolveDiscretizationProblem(AssignmentProblem, RelaxedSolution)

    If J(RelaxedSolution, DiscretizedSolution) < J(BestSolution) Then:
        BestSolution = (RelaxedSolution, DiscretizedSolution)
    End If
Until Converged Or maximum number of iteratons is reached
Return BestSolution
```

There are multiple ways of how the two partial problems can be solved. The following sections detail the implementation in this research.

Relaxation problem

While the discretization phase in this paper is rather seen as a way to "correct" continuous locations to the set of discrete locations, the relaxation solver is the heart of the algorithm. At this stage, our aim is to choose locations for all variable activities in an assignment problem such that we recover a given distance distribution from reference data. In this specific case, we only consider Euclidean distances.

In the case of the one-side constraint assignment problem (see Figure 1) we apply a simple algorithm that constructs a chain of locations around the only fixed one. First, we sample a random angle around the fixed location. Then we sample a distance from the predefined distance distribution. Knowing these two values, the location of the first variable activity is completely specified. If there is another variable activity, we can repeat the procedure but take the previously defined location as the starting point. We call this process the *angular solver* to the one-side constrained assignment problem. It is shown systematically in Algorithm 2.

ALGORITHM 2: Angular relaxation solver

Input: Fixed location (x_0, y_0)
Initialize: $i = 1$
While $i \leq$ Number of variable activities n
  $r \sim$ Distance distribution
  $\alpha \sim U(0, 2\pi)$
  $(x_i, y_i) = (r\cos(\alpha) + x_{i-1}, r\sin(\alpha) + x_{i-1})$
Continue
Return $((x_1, y_1), ..., (x_n, y_n))$

The relaxation problem is more interesting in the two-side constrained case. First, assume that only one variable activity is framed by two fixed ones. Let c define their direct Euclidean distance. Further, assume that two distances (d_1, d_2) have been sampled. Such a case is shown in Figure 2 on the left. In example A, the condition $d_1 + d_2 < c$ is true, i.e. given these two distances there is no feasible solution to the problem of placing the variable activity in such a way that it has distance d_1 to the first fixed activity and distance d_2 to the second fixed activity. The special case $d_1 + d_2 = c$ is shown in example B. There, *one* solution exists to the problem, which is to place the variable activity on a straight line between the fixed ones such that the distances match. Increasing distances even more, we arrive in example C, where $d_1 + d_2 > c$ is true. In that case *two* solutions exist, which can be mirrored at the straight line connecting the fixed activities. The exact locations can be obtained geometrically by intersecting two circles around the fixed activities with the respective radii d_1 and d_2.

Theses example show one component of our proposed relaxation algorithm: Given a list of distances (which we regard futher below) we want to place variable activities in such a way that the Euclidean distance between their locations matches the sampled reference distances. This implies that there is no "gap" in the chain.

How does the problem look like with more than one variable activity? Such a case is presented as example D in Figure 2. It is easy to imagine that all dashed points can be moved around in space almost freely while still maintaining all the correct distances. Only, one needs to "pull" or "push" other points to do so. This tought directly leads to the solution algorithm in this case, where we apply a force model. First, all variable activities are put on a straight line between the fixed activities, according to their order. Then, a small lateral deviation from that straight line is sampled for each activity and applied to the initial location. Then, a force model is run over multiple iterations. In this model, we loop through all the variable activities and calculate their current distances to their neighbors. If a distance is longer than the reference distance d_i the current point is moved towards the neighbor, if it is shorter than expected, the point is moved away from the neighbor. The

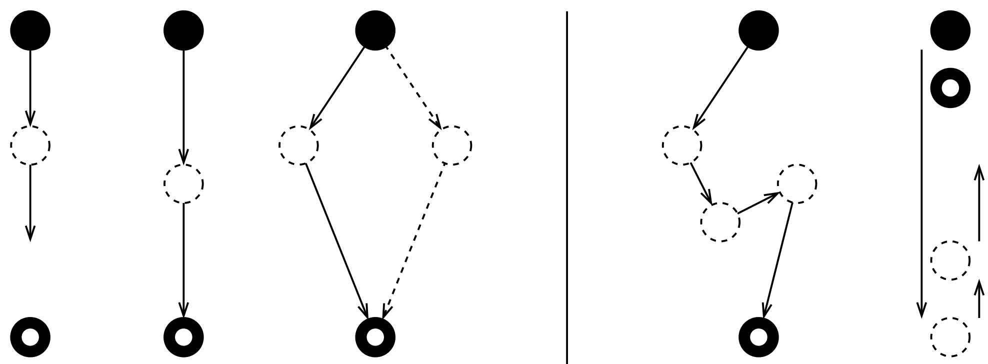

FIGURE 2 Possible solutions to the relaxation problem

displacement Δp is calculated along the direction vectors to the neighbors with p being the current location in Euclidean space, p' being the neighbor and d the reference distance:

$$\Delta p'(p, p', d) = \gamma \cdot (\|p - p'\| - d) \cdot \frac{(p - p')}{\|p - p'\|} \tag{1}$$

With p_L being the left neighbor and p_R being the right neighbor the total displacement is then

$$\Delta p = f(\cdot) = \Delta p'(p, p_L, d_L) + \Delta p'(p, p_R, d_R) \tag{2}$$

The parameter γ is a learning factor that determines how strongly the force model is evolving. A low γ leads to slow convergence (i.e. more iterations) to the equilibirum state, while a high γ tends to lead to oscillations with points making large jumps in space. Note that in equilibirum the distance between the observed distance and d vanishes and therefore no displacement takes place. Generally, this state is only achieved exactly after an infinite number of iterations. Therefore, we define a threshold value T. The algorithm then finishes as soon as all differences between expected and observed distances fall below T or a maximum number of iterations is reached. The full procedure is shown in Algorithm 3.

It is now defined how we solve the relaxation problem: In the case of one variable activity, the solution does not exist, is unique or chosen at random between the two mirrored options. Note that the implemented algorithm will still try to find a best guess solution (e.g. placing the location directly between the two fixed activities) while reporting that it did not converge if there is not feasible solution. In the case of more than one variable activity, the force model is used.

Feasible distances

In the previous section it already has been pointed out that given two distances d_1 and d_2 the relaxation problem is infeasible if their sum is smaller than the Euclidean distance between the fixed activities. This criterion can be generalized to more than one variable activity. Consider a chain of

ALGORITHM 3: Force-based relaxation solver

Input:
Fixed locations $p_0 = (x_0, y_0)$ and $p_N = (x_N, y_N)$
Reference distances $d_0, ..., d_{N-1}$

Initial locations:
$c = ||p_0 - p_n||$ (Direct distance)
$u = (p_n - p_0)/c$ (Normed direction vector)
$p_i = p_0 + u \cdot (i/n) \quad \forall i \in \{1, ..., N-1\}$

Lateral deplacement:
$q = (u_x, -u_y)$ (Normal vector)
$p_i = p_i + q \cdot e_i$ with $e_i \sim \mathcal{N}(0, \sigma)$ for all i

Do (Force model)
  $p_i = p_i + f(p_i, p_{i-1}, p_{i+1}, d_{i-1}, d_i)$ for all i
  Converged = $||p_{i+1} - p_i|| \leq d_i$ for all i
Until *Converged* **Or** maximum iterations reached

two fixed activities and two vairable ones as in Figure 2, example E. In this case the first distance is quite long, such that the next variable location must be far away from the the fixed point. However, the two other distances are so short, that they cannot cover the whole way back to the second fixed location. The feasibility condition for the relaxation problem must therefore be generalized to:

$$\sum_{i \neq j} d_i - c \geq d_i \quad \forall i \tag{3}$$

The condition says that no distance d_i can be larger than the sum of all other distances, minus the direct distance between the fixed points, which can be interpreted as the slack of the distance chain. Even before the relaxation algorithm can be run as stated above we therefore need to make sure that the provided distances fullfil these conditions. While more intelligent sampling approaches could be used in the future, we use the straight-forward scheme in Algorithm 4. There, we sample N distances, check whether they fulfill the condition of Equation 3, and, if not, repeat the sampling. Note that this process may skew the generated distance distribution.

ALGORITHM 4: Feasible distance chain sampler

Input: Distance distribution $\mathcal{D}$
Do (Force model)
  $d_i \sim \mathcal{D}$ for all i
  Converged = $\sum_{i \neq j} d_i - c \geq d_i$ for all i
Until *Converged* **Or** maximum iterations reached

Discretization problem and convergence

The discretization problem can be solved in many ways. Here, we decide to use the arguably simplest approach. Given a sampled chain of locations from the relaxation solver, we find the closest discrete location in terms of Euclidean distance, which fulfills certain criteria (for instance, it should be compatible with the respective activity type).

More elaborate approaches would be possible, such as finding the M closest discrete locations and sampling from them, or sampling from candidates within a specified radius around the relaxed location. Also, such sampling approaches could be extended to make use of attractivity measures for certain discrete locations.

Finally, the convergence objective can be defined in many ways. In this research, we determine convergence by comparing the reference distances from the relaxed solution with those in the discretized solution. As before, let p_i be the relaxed locations (with p_0 and p_N as the fixed ones). Let l_i be the discretized locations in Euclidean space. We can then define

$$\delta_i = |||p_{i+1} - p_i|| - ||l_{i+1} - l_i||| \tag{4}$$

as the absolute discretization error for each trip i. Based on the trip characteristics we can define a desired upper bound $\overline{\delta_i}$ for each trip i. Only if then $\delta_i \leq \overline{\delta_i} \quad \forall i$ we say that the discretization problem is converged. If not, new discrete locations can be sampled until convergence is achieved or the maximum number of iterations is reached. Note that in the discretization approach presented above there is no need yet for performing more than one iteration, because given a set of relaxed locations the result will always be the same.

Finally, we can define the objective for the upper-level assignment problem solver. In our current approach, we simply define $J = \max(\delta_i)$. This way, even if the whole algorithm may not converge perfectly, we always yield the solution with the smallest maximum deviation. For the whole assignment problem we define convergence when *all* parts, feasible distance sampler, relaxation model, discretization solver, have converged.

Summary

Figure 3 summarizes the relaxation-discretization algorithm. In state (a) a whole activity chain of an artificial traveller is shown. The traveller starts at home, goes to a shopping activity, and then to a leisure activity. Afterwards he goes to work and back home. Locations are already known for home and work, but not for the two other activities.

As the next step, feasible distances are sampled from a predefined distribution. The lengths of the blue dotted lines in (b) represent those distances. Note that initially the distance between the variable activities are smaller than the sampled ones. Therefore, the force model moves the activity locations until they reside in the blue equilibrium state.

Given the equilibirum state, the activity locations are discretized in step (c). For both activities a number of candidates is available from which the closest one is chosen. Finally, in (d), we can look at the relaxed locations and their respective discretized versions and check how their connecting distances compare to each other. Clearly, there is a discretization error for both trips, e.g. the

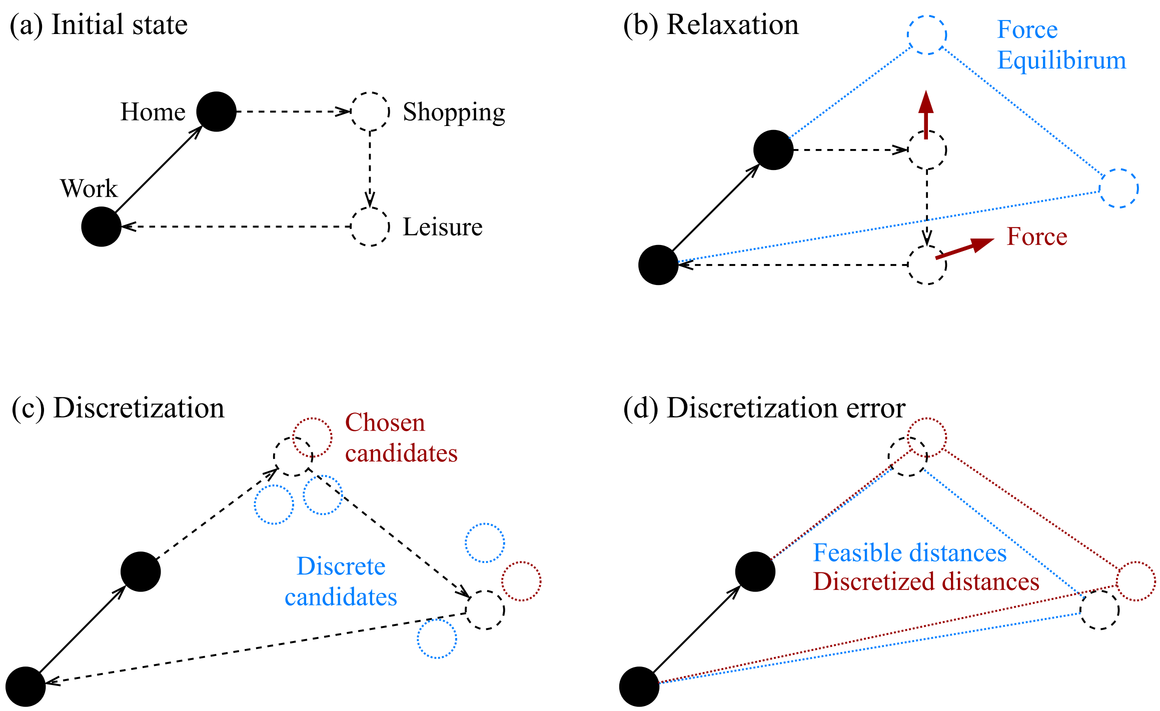

FIGURE 3 Summary of relaxation-discretization assignment problem

discretized distance from home to the shopping activity is longer than the sampled distance. The algorithm would now determine whether the deviations are too large and continue with the next iteration if neccesary.

EXPERIMENTS

The algorithm has successfully been applied to the synthesis of various populations for agent-based transport simulations. The following sections show two of the existing use cases. In each case - for the whole country of Switzerland and for the region of Île-de-France around Paris - similar data sets are used, which we first introduce briefly. Afterwards, we give some background on the respective simulation models and detail which data is relevant to the location assignment process. Finally, we report results on the respective use cases.

Agent-based transport models of Switzerland and Île-de-France

We consider two agent-based transport models, one for the region of Île-de-France *(upcoming, see 18)* and one for the wohle country of Switzerland *(19, 20)*. Both models are based on *eqasim*[1], which is a novel combination of the agent-based transport simulation framework *MATSim (15)* and a flexible extension that makes it possible to use discrete mode choice models *(21, 22)*. Furthermore, *eqasim* features a couple of tools which ease the development of input data for large-scale agent-based

[1]http://www.eqasim.org

transport models. Each of the two use cases has its own data pipeline, but the process is very similar. First, census data is used to synthesize an artificial population that resembles well the sociodemographic structure of the region. Second, the respective household travel survey is used to attach an activity chain to each of the synthetic persons, based on a number of predefined person and household attributes. While the home location of agents is known from the census data in both cases, activity locations for work and education are assigned based on known OD matrices. What remains then is to find locations for all non-primary activities, i.e. shopping, leisure and others. For those activities we have a set of discrete locations in both use cases. They are derived by the respective enterprise census.

The assignment problem for these *eqasim* models is defined as follows: We seek to find locations for secondary activities such that the overall distribution of distances matches well what we observe in the respective household travel survey (HTS). At the same time time, we want to make sure that distances between synthetic activities make sense given the mode of transport and travel time in the initial activity chains that are attached to the agents. Also, activities should only take place at locations where a viable discrete location exists.

Note that this is only an initial assignment. *MATSim* and the *eqasim* framework are used later on to simulate this synthetic population. Then, agents are able to make new mode decisions dynamically given the traffic conditions. In that sense, we seek to establish a credible starting solution for the dynamic simulation. Since location choice is not (yet) part of our simulation, the initial assignment must be of high quality as the generated distance distribution has strong influence on the mode choice behaviour, which is the focus of those simulations.

Location assignment process

In line with the requirements above we first track distance distributions by transport mode and travel bins in both use cases. We consider all trips in the respective HTS that do *not* solely connect fixed activity types (home, work, education). As the next step, for each mode, we define travel time bins by segmenting the distribution into N quantiles such that each quantile contains at least 400 samples. The result is shown in Figure 4. In the case of Switzerland, we arrive at 26 travel time bins for the "car driver" transport mode. Each of those bins then represents a distribution of Euclidean distances and Figure 4 shows their mean. For the "car driver" and "public transport" modes also the area between the 10% and 90% percentiles is shown in the background. As an example for reading the plot one can look at the "car driver" graph for the travel time bin between 30min and 40min. For these travel times a distance distribution exists which has a mean of around 19km.

Note that distributions of Euclidean distances are considered. This means that also for long travel times rather short distances can be observed. Reasons for that can be "loops" where people have reported that they just went for a round trip (and definitions whether to report an activity in between vary between different household travel surveys). Especially for Switzerland winding mountain roads may also explain rather short distances for long travel times.

In the location assignment algorithm the distributions are used as follows. When sampling feasible distances for an assignment problem, the transport mode and initial HTS travel time is known for each trip. Based on these two values a distance distribution is selected from the data presented in Figure 4, and distance observations are sampled for all trips. This way trips by bike

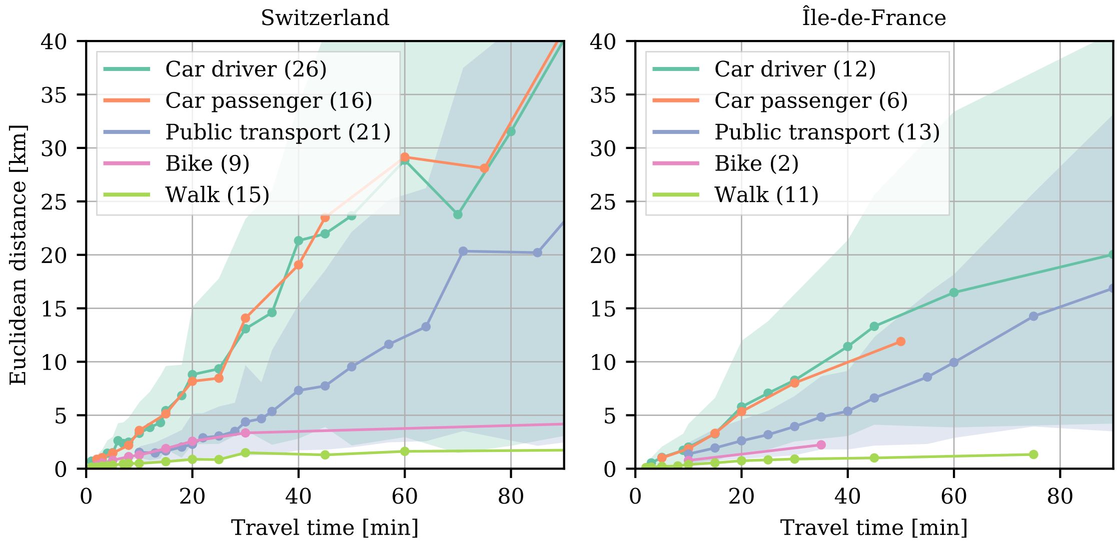

FIGURE 4 Input distributions to the location assignment algorithm. For all transport modes the mean is shown; for driving a car and public transport the area between the 10% and 90% percentile is indicated.

receive different distances than trips by public transport, for instance.

The activity chains also provide information on the type of each activity. It is divided into "shopping", "leisure" and "other". For each of these categories, the respective data sets provide distinct sets of discrete locations. Therefore, if an activity with type "shopping" is discretized, only compatible discrete locations are considered.

In the standard form of the algorithm, which is used actively in our model development, we use the following inputs and parameters:

- **Data**
  - Distance distributions by mode and travel time
  - Discrete locations by activity type
- **Force model**
  - Lateral deviation: $\mathcal{N}(0, \sigma = 10m)$
  - Displacement factor $\gamma = 0.1$
  - Convergence threshold: $T = 10m$
- **Maximum iterations**
  - Feasible distance sampler: 1000
  - Force model: 1000
  - Assignment solver: 1000
- **Maximum discretization errors $\overline{\delta}$**
  - Car driver, car passenger, public transport: $200m$
  - Walk, bike: $100m$

Especially the last parameters have strong influence on the model performance. In the usual case, we define that the discretized distances should not deviate by more than $200m$ or $100m$, respectively,

	Car driver	Car passenger	Public transport	Bike	Walk
Switzerland	0.8	1.0	1.0	0.0	0.0
Île-de-France	0.0	0.1	0.5	0.0	-0.5

TABLE 1 Reweighting factors for the input distance distributions.

from the relaxed solution.

Resampling of input distributions

In terms of model calibration, the two input data sets represent our degrees of freedom. Especially the input distribution can heavily affect the distance distribution in the assigned activity chains. In fact, experiments have shown that the algorithm tends to skew the distance distribution towards shorter distances. This can most likely be explained by the constrained way in which feasible distances are sampled and is a pathway for future research. For practical use, we do not use the exact input distribution as shown in Figure 4, but perform a resampling of the data points.

Let $d_i < d_{i+1}$ be the ordered distance samples in any of the mode and travel time bins and let $f(d_i)$ be their normed weight. We then perform a linear reweighting according to

$$f'(d_i) = \begin{cases} f(d_i) \cdot (1 + \alpha \cdot (i/N)) & \text{if } \alpha \geq 0 \\ f(d_i) \cdot (1 + |\alpha| \cdot [1 - (i/N)]) & \text{else} \end{cases} \quad (5)$$

Afterwards, the weights are normalized again. Later, they are used when sampling feasible distances. Note that if the reweighting factor $\alpha \geq 0$ we oversample long distances, and when $\alpha < 0$ we focus on short distances. The values for the experiments in the paper at hand are documented in Table 1.

Results

The location assignment model was run with the parameters and input as specified above. Figure 5 shows the resulting distance distribution in comparison to reference data from the household travel surveys. After resampling we get a very good fit for all modes of transport. Note that the reference data is sometimes too coarse to make a more analytical comparison in the sense of a Kolmogorov-Smirnoff test, or similar, feasible. For instance, the data for Île-de-France shows heavy rounding of short distances, as can be seen in the lower left part of Figure 5.

Figure 6 shows the mean, median and 90% quantile of mode-indepdendent distributions of Euclidean distances by travel time bin. Note that the travel times in the assignment cases come from the activity chains of the agents while the Euclidean distances are derived from the discrete locations that have been assigned in the location assignment process. We see that, as expected from the sampling, the distance distributions match well the reference values.

In Table 2 we provide some key metrics for the algorithm. Considering the large amount of problems that need to be solved, the algorithm runs fairly quickly. It is possible to reassign a whole agent population in a matter of few hours. We yet have to perform a detailed analysis on the performance of the algorithm. With the convergence rate presented in Table 2 we obtain a good

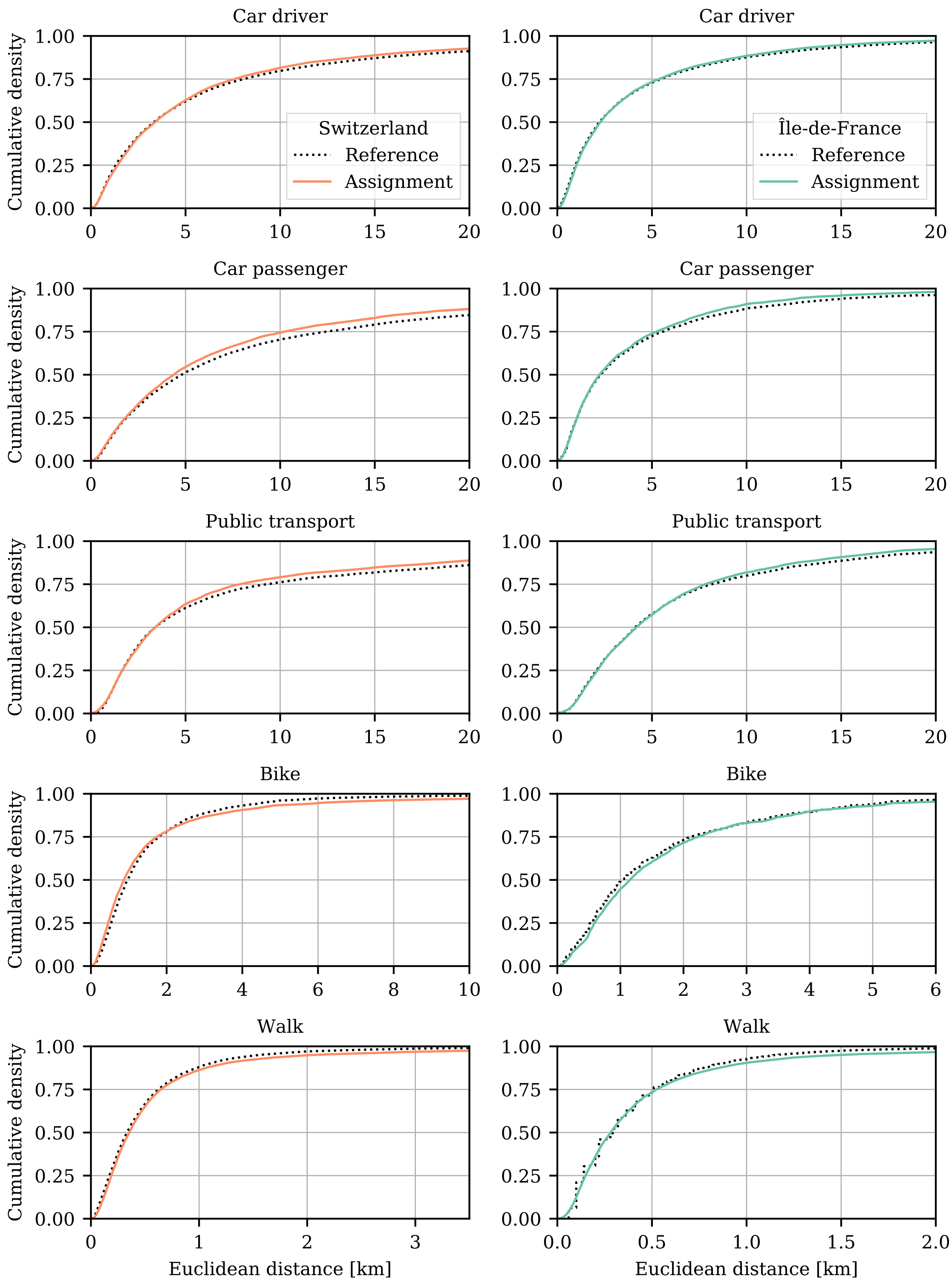

FIGURE 5 Comparison of assignment results with HTS data in terms of Euclidean distance distributions by mode.

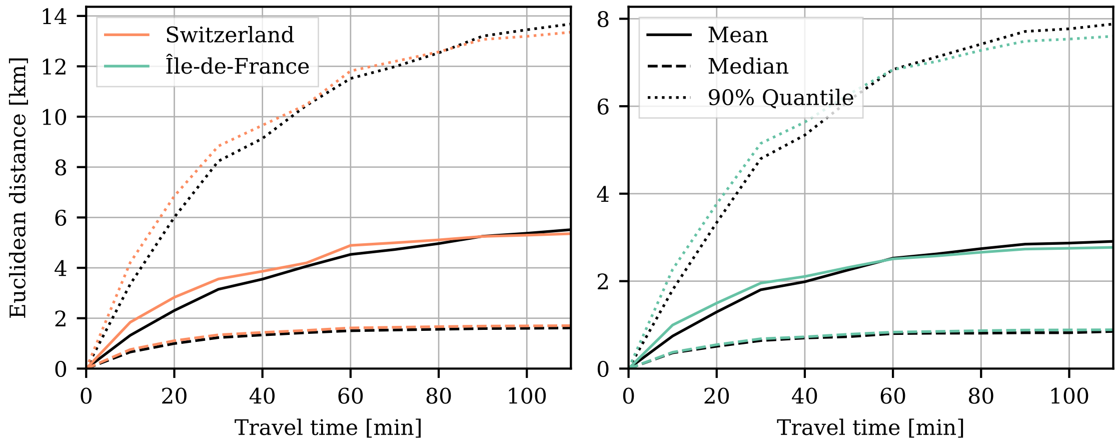

FIGURE 6 Comparison of Euclidean distance distribution for specific travel time bins by mean, median and 90% centile. The reference data is given in black.

	Switzerland	Île-de-France
Performance		
Runtime	170 min	400 min
Agents	8 million	13 million
Assignment problems	8,135,921	13,718,250
Average trips per problem	2.3	2.35
Convergence		
Feasible distance sampler	99.3%	98.7%
Relaxation	93.2%	92.4%
Discretization	98.3%	97.2%
Assignment	92.5%	91.0%
Errors		
Mean discretization error	92 m	89 m
Mean excess error	19 m	29 m

TABLE 2 Key metrics for performance and convergence of the algorithm

match in distance distributions. It will be interesting to explore how changing the convergence thresholds would affect precision and runtime of the algorithm. The lower part of Table 2 shows the resulting errors. On average, our discretization error is less than 100 m. The excess error describes the distance that exceeds the defined distance thresholds. With a value of less than 30 m this indicates that the algorithm not always converges, but if it does not, the maximum deviation is only 30 m on average.

DISCUSSION

To start the discussion about our algorithm it needs to be pointed out that the algorithm is considered mainly data-driven in the sense that it does not try to uncover the underlying process of choosing activity locations. This is the big difference to existing activity-based models where often choice models are applied to make decisions. Therefore, we consider the alogrithm a location *assignment* approach, rather than a location *choice* process.

Therefore, we do not get any deeper insight from our algorithm on *why* people go to certain locations. We only reproduce the distances that that can be observed. While this can be seen as a big drawback of the presented algorithm, we need to state that the foremost objective of developing it was to find an easy and practical way of assigning secondary locations such that they can serve as input to an agent-based transport simulation. In that sense, the algorithm performs well. In fact, the only inputs it needs are the assignment problems (or whole activity chains), the reference distance distributions, and a list of discrete locations. While the code (which is available open source) currently operates on the respective data structures of the MATSim framework, a version that solely acts on generic CSV data is planned. Given these data sets, which are usually easy to obtain, researchers and practicioners can set up the code in a couple of minutes and the runtimes we report in Table 2 for fairly large agent populations show that results can be obtained rather quickly. Note that only very limited calibration effort is needed and no models need to be estimated prior to applying the algorithm.

Yet, there are multiple points how the algorithm can be improved. The most important future step we see is to verify spatial consistency. Our experiments with Switzerland and Île-de-France have shown that realistic distance distributions emerge not only globally, but also in comparison between rural and urban regions. A potential reason for that is that the constraints that are imposed by the fixed and discrete locations automatically lead to distance distributions that are spatially context-dependent. However, a more rigorous spatial validation would be interesting in the future. Also, comparing the reference and synthesized joint distribution of sequential trip lengths will be an interesting analysis.

Furthermore, there is reason to believe that secondary locations are distributed rather evenly within their respective spatial context. In our current approach we do not consider attractivity levels for discrete locations or their surrounding neighborhoods. In that sense large shopping malls are not assigned more frequently than smaller shops. Therefore, implementing an attractivity measure into the discretization process will be an interesting task for the future. Another intersting aspect that goes beyond a simple sense of attractivity is the capacity of discrete locations. Applying the whole algorithm in an iterative fashion or tracking occupancy rates during runtime could be two possible ways forward in that direction.

A last drawback we want to mention is that the current setup makes heavy use of Euclidean distances. One can actually think of using routed (maybe even congested) network distances at several points in the algorithm. The most complicated idea would probably be to replace the force-based relaxation process by one that meanders the network to find "network-relaxed" locations. This could maybe even happen in a two-step process where the force model gives a first starting solution. A more simple approach would be to integrate network distances into the assignment objective. Then, one could perform a routing only after all discrete locations have been assigned. One could compare them to sampled network distances that were fed into the force model, maybe with a certain factor that translates roughly between network and Euclidean distance.

CONCLUSION

In conclusion, we have presented a novel location assignment algorithm that is able to produce an agent population with realistic secondary activity locations. It has low demand on input data that needs to be prepared a priori and it shows good run times on fairly large simulation scenarios. While the general algorithm structure is straight-forward we give a non-compehensive list of potential improvements that can be made to the very basic version that is presented in this paper.

While we show that the algorithm yields good fit and performance for two fairly large-scale agent-based transport models for Swizerland and Île-de-France, it has already been applied to other cities such as Sao Paulo. By providing the algorithm open source[2] we hope to see more applications of the algorithm, potentially with many creative extensions, and also outside of the MATSim ecosystem.

ACKNOWLEDGEMENTS

The model of Switzerland that is used in this paper is based on data from the Federal Statistical Office of Switzerland. Specifically, it draws from the population census data *(23)*, the national household survey *(24)*, the enterprise census *(25)*, and the national household travel survey *(26)*.

The model of Île-de-France is based on data from the National Institute of Statistics and Economic Studies in France, namely their population census data *(27)* and enterprise census *(28)*. It further draws from the regional household travel survey for Île-de-France conducted by OMNIL, DRIEA and STIF *(29)*.

AUTHOR CONTRIBUTION

The authors confirm contribution to the paper as follows: Sebastian Hörl conducted the study and prepared the report. Prof Kay. W. Axhausen gave feedback on the results, and made this resarch possible at the Institute for Transport Planning and Systems at ETH Zurich. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

1. Schirmer, P., M. B. van Eggermond and K. Axhausen (2014) The role of location in residential location choice models: A review of literature, *Journal of Transport and Land Use*, **7**, 3–21.

2. Guo, J. and C. Bhat (2007) Operationalizing the concept of neighborhood: Application to residential location choice analysis, *Journal of Transport Geography*, **15**, 31–45.

3. Sener, I., R. Pendyala and C. Bhat (2011) Accommodating spatial correlation across choice alternatives in discrete choice models: an application to modeling residential location choice behavior, *Journal of Transport Geography*, **19**, 294–303.

4. McFadden, D. (1978) Modeling the choice of residential location, in A. Karlqvist, L. Lundqvist, F. Snickars and J. Weibull (eds.) *Spatial Interaction Theory and Planning Models*, 75–96, North Holland, Amsterdam.

[2]https://github.com/eqasim-org/eqasim-java

5. Sá, C., R. Florax and P. Rietveld (2007) Determinants of the regional demand for higher education in the netherlands: A gravity model approach, *Regional Studies*, **38**, 375–392.

6. Filippo, S., M. González, A. Maritan and A.-L. Barabási (2012) A universal model for mobility and migration patterns, *Nature*, **484**, 96.

7. Masucci, A., J. Serras, A. Johansson and M. Batty (2013) Gravity versus radiation models: On the importance of scale and heterogeneity in commuting flows, *Physical Review E*, **88**.

8. Yang, Y., C. Herrera, N. Eagle and M. González (2014) Limits of predictability in commuting flows in the absence of data for calibration, *Scientific Reports*, **4**, 5662.

9. Vitins, B., A. Erath and K. Axhausen (2016) Integration of a capacity-constrained workplace choice model: Recent developments and applications with an agent-based simulation in singapore, *Transportation Research Record*, **2564**, 1–13.

10. Erath, A., N. Frank, R. Lademann and K. Axhausen (2007) The impact of travel time savings on shopping location choice or how far do people go to shop cheaply?, paper presented at the *14th International Conference on Recent Advances in Retailing and Service Science*, San Francisco.

11. Pozsgay, M. and C. Bhat (2001) Destination choice modeling for home-based recreational trips: Analysis and implications for land-use, transportation, and air quality planning, *Transportation Research Record*, **1777**.

12. Arentze, T., F. Hofman, H. van Mourik and H. Timmermans (2000) Albatross: Multiagent, rule-based model of activity pattern decisions, *Transportation Research Record*, **1706**, 136–144.

13. Arentze, T. and H. Timmermans (2007) A multi-agent activity-based model of facility location choice and use, *disP*, **170**.

14. Miller, E. and M. Roorda (2003) A prototype model of household activity/travel scheduling, *Transportation Research Record*, **1831**.

15. Horni, A., K. Nagel and K. W. Axhausen (eds.) (2016) *The Multi-Agent Transport Simulation MATSim*, Ubiquity, London.

16. Marchal, F. and K. Nagel (2005) Modeling location choice of secondary activities with a social network of cooperative agents, *Transportation Research Record*, **1935**, 141–146.

17. Horni, A., D. Scott, M. Balmer and K. Axhausen (2009) Location choice modeling for shopping and leisure activities with matsim: Combining microsimulation and time geography, *Transportation Research Record*, **2135**, 87–95.

18. Hörl, S., M. Balac and K. W. Axhausen (2019) Dynamic demand estimation for an amod system in paris, *IEEE Intelligent Vehicles Symposium 2019*.

19. Hörl, S., F. Becker, T. Dubernet and K. W. Axhausen (2019) Induced demand by autonomous vehicles: An assessment, *Project report SVI 2016/001, FB 1650*, SVI.

20. Balac, M., F. Becker, F. Ciari and K. W. Axhausen (2019) Modeling competing free-floating carsharing operators - a case study for zurich, switzerland, *Transportation Research: Part C*, **98**, 101–117.

21. Hörl, S., M. Balac and K. W. Axhausen (2018) A first look at bridging discrete choice modeling and agent-based microsimulation in matsim, *Procedia Computer Science*, **130**, 900 – 907.

22. Hörl, S., M. Balac and K. W. Axhausen (2019) Pairing discrete mode choice models and agent-based transport simulation with MATSim, *98th Annual Meeting of the Transportation Research Board*.

23. Federal Statistical Office (2018) Statistik der Bevölkerung und der Haushalte (STATPOP).

24. Federal Statistical Office (2018) Strukturerhebung.

25. Federal Statistical Office (2018) Statistik der Unternehmensstruktur (STATENT).

26. Federal Statistical Office, Federal Office for Spatial Development (2017) Verkehrsverhalten der Bevölkerung. Ergebnisse des Mikrozensus Mobilität und Verkehr 2015.

27. National Institute of Statistics and Economic Studies (2018) Recensement de la population.

28. National Institute of Statistics and Economic Studies (2018) Base permanente des équipements.

29. OMNIL, DRIEA, STIF (2018) Enquête globale de transport.