


Relaxation–discretization algorithm for spatially constrained secondary location assignment

Journal Article**Author(s):**

Hörl, Sebastian; Axhausen, Kay W. 

Publication date:

2023

Permanent link:

<https://doi.org/10.3929/ethz-b-000508284>

Rights / license:

[Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International](#)

Originally published in:

Transportmetrica A: Transport Science 19(2), <https://doi.org/10.1080/23249935.2021.1982068>

Relaxation–discretization algorithm for spatially constrained secondary location assignment

Sebastian Hörl ^{a,b} and Kay W. Axhausen ^a

^aInstitute for Transport Planning and Systems, ETH Zurich, Zürich, Switzerland; ^bInstitut de Recherche Technologique SystemX, Palaiseau, France

ABSTRACT

Agent-based transport models demand that the daily activity patterns of artificial agents are described in great detail. While choice models for residential locations or workplaces exist, only few approaches are available to find locations for highly constrained secondary activities such as grocery shopping or recreation at high resolution. The paper describes a data-driven approach of assigning viable locations to such secondary locations while maintaining consistency with homes, workplaces and other fixed points in an artificial traveler'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.

ARTICLE HISTORY

Received 12 April 2021

Accepted 11 September 2021

KEYWORDS



Transport; simulation; secondary; location; assignment; data-driven; synthetic; population

1. Introduction

In recent years, agent-based transport models have gained large interest, not only from researchers but also from practitioners. 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 a considerable amount of work. Contrary to more aggregate approaches, the attributes, intentions and interactions between many individual travelers need to be modeled. While for many dimensions useful data exists, such as census data to determine home locations of agents, commuter matrices to assign workplaces, or household travel surveys (HTS) to describe daily mobility schedules, there are still gaps. One major unknown are usually locations of secondary activities, e.g. where people go shopping, engage in leisure or eat. A reason for that is that such choices are much richer and more detailed than residential or work place choices for which standard approaches and detailed data are available.

Literature on *residential* location choice is vast and mostly related to discrete choice modeling (McFadden 1978; Guo and Bhat 2007; Sener, Pendyala, and Bhat 2011; Schirmer, van Eggermond, and Axhausen 2014). Likewise, models such as the gravity model have

CONTACT Sebastian Hörl  sebastian.horl@irt-systemx.fr, hoerl.sebastian@gmail.com  Institute for Transport Planning and Systems, ETH Zurich, Stefano-Franscini-Platz 5, Zürich CH-8093, Switzerland

© 2021 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group

This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited, and is not altered, transformed, or built upon in any way.

emerged as standard procedures for assigning work or education locations that resemble well daily commuting patterns (Sá, Florax, and Rietveld 2007; Filippo et al. 2012; Masucci et al. 2013; Yang et al. 2014). Also, models have been presented for capacitated work location choice (e.g. Vitins, Erath, and Axhausen 2016). Unfortunately, these approaches are difficult to apply to *secondary* locations. Those often ‘fill’ gaps between the *primary* home, work and education activities of people. Therefore, they are much more constrained in terms of reachability and, at the same time, depend highly on individual taste variations, with hundreds or thousands of potential alternatives. For instance, Kusumastuti et al. (2010) perform a structured survey to qualitatively reveal the stages and preferences constituting the choice-making process for leisure shopping activities.

Some discrete choice models have been proposed that give insight into the choice behavior for certain, very specific activity types given certain attributes of locations or zones. For instance, Erath et al. (2007) develop a discrete choice model explaining which attributes of potential shopping destinations influence the location choice of the people, and Pozsgay and Bhat (2001) look at *home-based* recreational activities with aggregated zones as potential destinations. Hence, both approaches do not provide sufficient detail to select discrete locations from a given set of alternatives.

To achieve this level of detail, a range of studies makes use of the geographic concept of space-time prisms. In the relevant studies, first, a set of viable secondary activity locations is generated. The approach considers network travel times and makes sure that only locations are selected that can be reached under consideration of time constraints for the departure and arrival at surrounding fixed activities and opening times of the secondary activity locations. After that, choice models are used to select a specific location from the obtained choice set (Yoon et al. 2012; Justen, Martínez, and Cortés 2013). Ma and Klein (2018) extend the approach by making use of specific heuristics for location choice preferences using Bayesian networks. These approaches are merely used in the context of geographic studies, where the reachability and accessibility of locations, given current traffic conditions, is of interest. They have in common, that information on travel times at the relevant times of day must be available. It is hence difficult to use these models as a preparatory stage for traffic simulation as the travel times usually are the very outcome of those simulations.

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 widespread 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 modeling. For instance, ALBATROSS (Arentze et al. 2000; Arentze and Timmermans 2007) and TASHA (Miller and Roorda 2003) each apply different strategies of implementing location choices into their activity scheduling frameworks by different heuristic means of reducing the available choice set and fulfilling travel time constraints as in the space-time prism approach.

In the context of the agent-based transport simulation framework MATSim (Horni, Nagel, and Axhausen 2016) efforts have been pushed to put location choice into its evolutionary model where agents make random decisions (such as selecting a location nearby an existing choice) and keep promising updates while discarding those that are not beneficial.

Marchal and Nagel (2005) consider a limited agent memory of known facilities for secondary locations, while Horni et al. (2009) explore the use of the concept of ‘frozen randomness’ which applies constant error terms to the attractiveness of each possible secondary activity location. Again, the approach solves the problem of secondary location choice by defining limited search spaces to cope with the vast amount of options. Unfortunately, it is part of a computationally heavy agent-based simulation framework. Hence, it is useful for refining agent decisions, but not for generating location choices from scratch.

In recent years, data-driven approaches have emerged that create synthetic human mobility traces from phone data. While those approaches represent well the general movement patterns of the share of the population which are customers of a specific provider, they may not be representative for the whole population. Furthermore, while providing spatial detail, they commonly do not provide additional information such as activity types along the mobility traces or sociodemographic attributes of the travelers (Jiang et al. 2016; Anda, Ordonez Medina, and Axhausen 2021).

To summarize, existing approaches related to secondary location choice either require *a priori* information which is not available in the context of transport simulation (mainly travel times) or requires to estimate detailed choice models, which is a time-consuming and very case-specific process. For agent-based transport simulations, which may be able to adaptively refine location decisions, useful starting solutions are necessary.

In this paper, we describe a new approach for finding viable locations for secondary activities based on Euclidean distances as a universally available information from household travel surveys and similar data sets. Contrary to existing approaches, which have focused on explaining people’s choice behavior and replicating those choices, the proposed method is data-driven in the sense that we replicate reference distance distributions and their correlations to trip attributes. The algorithm is neither dependent on travel times (which are often the very outcomes of the transport simulations for which the location choice is supposed to be performed in the first place), nor on the modeling of sophisticated decision processes. Furthermore, it allows to select from a large set of discrete locations. The algorithm, hence, strives to provide a starting solution of secondary locations for agent-based simulation models which approximates the spatial structure of real activity chains by imitating existing correlations, but without giving a causal explanation for those structures. The selected locations can be refined using more behavioral approaches.

The remaining part of the paper is structured as follows. First, we describe our method in detail. Afterwards, we present results for two large-scale agent-based simulation models of Switzerland and Île-de-France. Finally, we provide a discussion including pathways for future research, followed by concluding remarks.

2. 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 modeling would have a fixed home location for each agent and its workplace may be known from a separate commuting destination model. In such a case, it

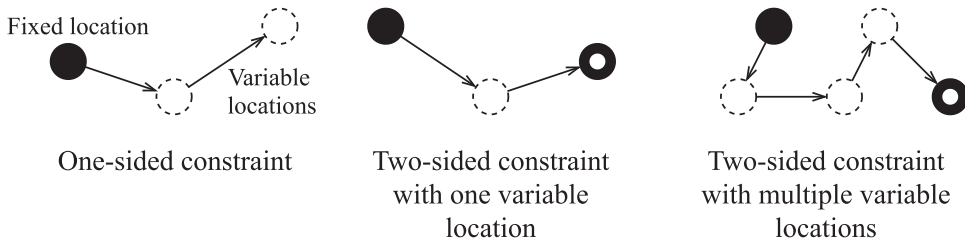


Figure 1. One-sided constraint and two-sided constraint assignment problems.

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 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 locations are generally known upfront, e.g. as a list of all shops in a 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 transport mode 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 result of the relaxation problem has a 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.

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

2.1. 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 pre-defined 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 1.

Algorithm 1 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

$\alpha \sim U(0, 2\pi)$

$(x_i, y_i) = (r \cos(\alpha) + x_{i-1}, r \sin(\alpha) + y_{i-1})$

Continue

Return $(x_1, y_1), \dots, (x_n, y_n)$

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 .

These examples show one component of our proposed relaxation algorithm: Given a list of distances (which we regard further 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 thought 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

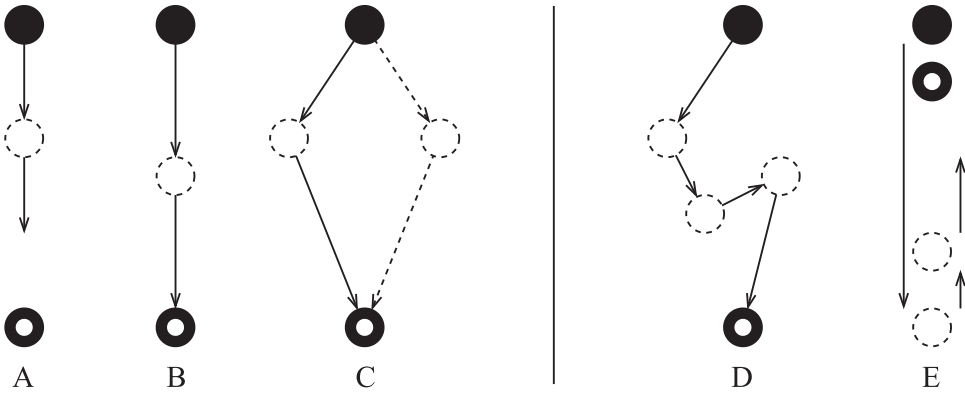


Figure 2. Possible solutions to the relaxation problem.

lateral deviation from that straight line is sampled for each activity and applied to the initial location. After that, 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 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:

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

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

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

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 equilibrium state, while a high γ tends to lead to oscillations with points making large jumps in space. Note that in equilibrium 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 2.

In Figure 3, the procedure is visualized. In the presented case, currently one variable activity in a chain is chosen (at location p) in a chain of two fixed activities (filled circles) and two variable activities (dashed circles). The left neighbor of p is shown in red, and the right neighbor is shown in blue, with current positions p_L and p_R . The large dashed circles show the *desired* distances d_L and d_R between the two neighbors and the selected activity, respectively. For L , p is closer than the desired distance. Hence, a displacement towards the outer circle Δp_L is calculated. Compared to the desired distance towards R , the selected activity

Algorithm 2 Force-based relaxation solver**Input:**Fixed locations $p_0 = (x_0, y_0)$ and $p_N = (x_N, y_N)$ Reference distances d_0, \dots, d_{N-1} **Initial locations:** $c = \|p_0 - p_n\|$ (Direct distance) $u = (p_n - p_0)/c$ (Normed direction vector) $p_i = p_0 + u \cdot (i/n) \quad \forall i \in \{1, \dots, N-1\}$ **Lateral displacement:** $q = (u_x, -u_y)$ (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**Return** $Converged, (p_1, \dots, p_n)$

is too far away. Hence, a displacement towards R is calculated as Δp_R . The overall displacement of the selected activity in this iteration is then $\Delta p = \Delta p_L + \Delta p_R$. After this calculation, the selected activity is moved to the newly calculated position which lies in the direction of the crossing point of the large dashed circles. Afterwards, the point currently denoted as p_L will be treated the same way, with the spot in gray being its left neighbor and the currently selected activity being its right neighbor. Iteration by iteration, both points will then move towards their ideal position to replicate the three desired distances (the ones between the fixed activities and their direct variable neighbors, and the one between the two variable activities).

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 a feasible solution. In the case of more than one variable activity, the force model is used.

2.2. 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 two fixed activities and two variable 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 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

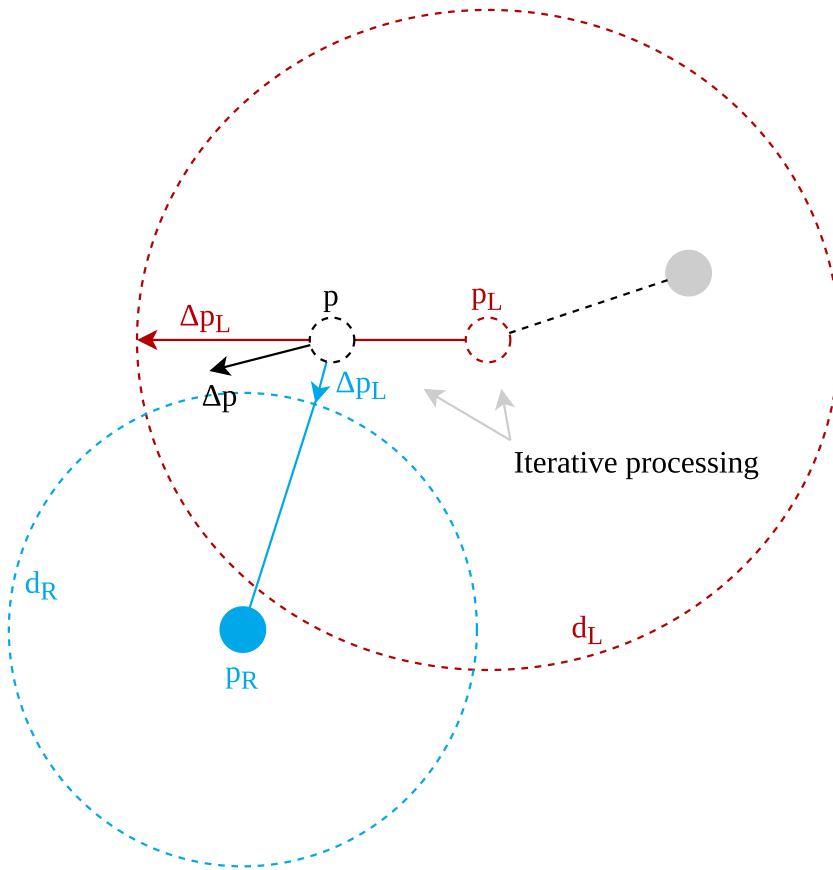


Figure 3. Visualisation of the force model.

relaxation problem must therefore be generalized to:

$$d_i \leq c + \sum_{i \neq j} d_j \quad \forall i \quad (3)$$

The condition says that no distance d_i can be larger than the sum of all other distances, plus 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 fulfill these conditions. While more intelligent sampling approaches could be used in the future, we use the straight-forward scheme in Algorithm 3. There, we sample N distances, check whether they fulfill the condition of Equation 3, and, if not, repeat the sampling.

2.3. 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).

Algorithm 3 Feasible distance chain sampler

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

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. For the purpose of the case study presented below, the simple approach yielded sufficiently satisfying results, but we provide further options and pathways for improving the discretization step in the discussion section.

The discretization 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

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

as the absolute discretization error for each trip i . Based on the trip characteristics, we can define a desired upper bound $\bar{\delta}_i$ for each trip i . Only if then $\delta_i \leq \bar{\delta}_i \forall i$ 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 here 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.

The upper bounds $\bar{\delta}_i$ need to be defined by the modeler. They will strongly depend on what information is available on the activities to be connected and the connecting trips. For instance, in the case study below, the transport mode of a connecting trip will be known. In that case, we will use a smaller upper bound for walk trips than for car trips. It would equally possible to define a generic threshold based on modeling experience or the level of detail of the underlying data sets.

2.4. Summary

The individual components of our approach are linked as described in Algorithm 4. First, feasible distances d_i are sampled (or nearly feasible ones if the algorithm has not converged); second, the relaxation problem is solved, which yields locations in Euclidean space p_i as well as information on whether the algorithm has converged; third, the locations are discretized to l_i given the Euclidean-space locations and desired distances. Also, the discretization step yields whether the discretized locations fall well into the defined requirements.

To complete the algorithm, an objective is calculated which quantifies the aptness of the current solution. Here, we define $J = \max(\delta_i)$, i.e. the maximum deviation between the distance between two activities and their desired distance. This means the better the whole process has assigned locations, the better the objective will be. If all distance requirements

are met perfectly, the objective will be zero. Also, we define that the overall algorithm has converged once the three sub-stages have completed successfully as well.

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. The current best solution is then updated either if we have found an objective that is better than before or if we have found a converged solution. In that case, the algorithm is aborted and the solution is returned. In case, the current iteration has not converged in all stages, the sequence is repeated. The algorithm proceeds iteratively until a valid solution is found or until a maximum number of iterations has been reached.

Algorithm 4 Assignment Problem Solver

Input: AssignmentProblem

Initialize: BestSolution = Null, $J^* = \infty$

Do:

$C_F, d_i = \mathbf{SampleFeasibleDistances}$ (AssignmentProblem)

$C_R, p_i = \mathbf{SolveRelaxationProblem}$ (AssignmentProblem, d_i)

$C_D, l_i, \delta_i = \mathbf{SolveDiscretizationProblem}$ (AssignmentProblem, d_i, p_i)

$J = \max_i\{\delta_i\}$

$Converged = C_F \wedge C_R \wedge C_D$

If $J < J^* \vee Converged$ **Then:**

$J^* = J$

BestSolution = l_i

End If

Until $Converged$ **Or** maximum number of iterations is reached

Return $Converged, BestSolution$

Figure 4 summarizes the relaxation–discretization algorithm. In state (a), a whole activity chain of an artificial traveler is shown. The traveler 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 equilibrium state, the activity locations are discretized in step (c). For both activities, a number of candidates are 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 discretized distance from home to the shopping activity is longer than the sampled

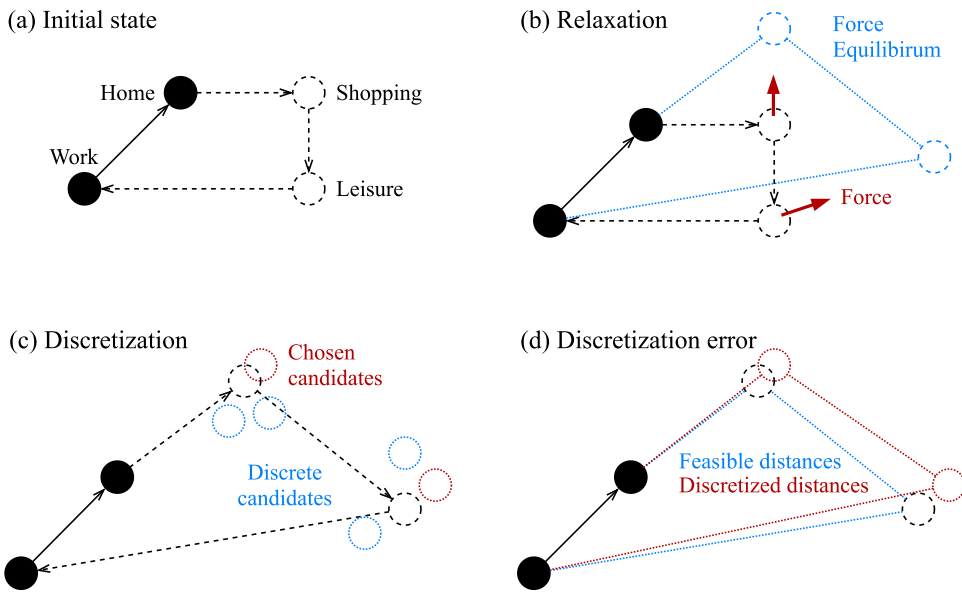


Figure 4. Summary of relaxation–discretization assignment problem.

distance. The algorithm would now determine whether the deviations are too large and continue with the next iteration if necessary.

3. 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.

3.1. Case studies

We consider two synthetic travel demand models, reflecting households, persons, and their daily activity chains – one for the region of Île-de-France (Hörl, Balac, and Axhausen 2019; Hörl and Balac 2021) and one for Switzerland (Balac et al. 2019; Hörl, Becker, and Axhausen 2021). They are intended for the use in MATSim (Horni, Nagel, and Axhausen 2016), a framework for agent-based transport simulation where the movements of people, which are defined in detail using the travel demand data sets, can be simulated in detail. 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 HTS 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 which exist as public and proprietary data for Île-de-France and Switzerland, respectively. What remains then is to find locations for all non-primary activities, i.e. shopping, leisure and others.

Based on the synthetic populations, we search for primary activities (home, work, education) and cut the activity chains such that we arrive at assignment problems that start with one primary activity, followed by one or multiple secondary activities which have not been assigned a location yet, followed by another primary activity. As the activity chains are generated by attaching chains from the HTS to a set of synthetic persons, also the *observed* transport mode between two activities and the *observed* travel time are known. Furthermore, the type of each activity is noted down. The models distinguish between *home*, *work*, *education*, *shopping*, *leisure*, and *other*, which is a common level of granularity for synthetic populations which are intended to be used as input to the agent-based transport modeling framework MATSim. Also, these activity types roughly represent the location information that is available from official data for the use cases. An extracted assignment problem consists hence of the locations of the enveloping primary activities, the number of secondary activities in between, the transport modes and travel times of the trips between all involved activities and the type of each activity.

Furthermore, distributional information on travel times and distances is available from the respective HTS, and a set of discrete locations by activity is available in each case based on the respective enterprise census. Details on the processing of the data sets in the two use cases are given in the references provided above.

The assignment problem for these 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 HTS. At the same time, we want to make sure that distances between synthetic activities make sense given the travel mode and 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 is 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 a strong influence on the mode choice behavior, which is the focus of those simulations.

3.2. Location assignment process

In line with the requirements above, we first track distance distributions by travel mode and time 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 5. 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 5 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

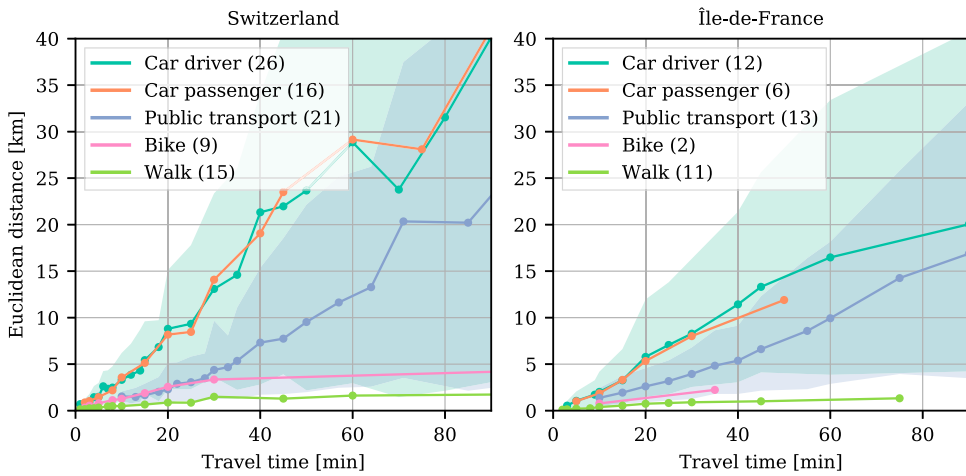


Figure 5. Input distributions to the location assignment algorithm. For all transport modes, the *mean distance* is shown for travel time bins. The points indicate the upper bound of each bin. For driving a car and public transport the range between the 10% and 90% percentile of the respective distributions in each bin is indicated as a shaded area.

the ‘car driver’ graph for the travel time bin between 30 and 40 min. For these travel times, a distance distribution exists which has a mean of around 19 km.

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 of whether to report an activity in between vary between different HTS). 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 means of transport and initial survey travel time is known for each trip. Based on these two values, a distance distribution is selected from the data presented in Figure 5, and distance observations are sampled for all trips. This way trips by bike receive different distances than trips by public transport, for instance.

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
 - Structural chains of activities with
 - activity types
 - connecting transport modes and expected travel time
- *Force model*
 - Lateral deviation: $\mathcal{N}(0, \sigma = 10 \text{ m})$
 - Displacement factor $\gamma = 0.1$
 - Convergence threshold: $T = 10 \text{ m}$

- *Maximum iterations*
 - Feasible distance sampler: 1000
 - Force model: 1000
 - Assignment solver: 1000
- *Maximum discretization errors $\bar{\delta}$*
 - Car driver, car passenger, public transport: 200 m
 - Walk, bike: 100 m

The last parameters have a strong influence on the model performance. If they are very low, many solutions will not be accepted as the discretized distances will be too far off from the freely generated distances. Hence, the algorithm will need to perform an increasing number of iterations to find a solution, or, when the iteration limit is reached, mark an increasing number of problems as unresolved. This is especially the case if the available discrete locations are sparsely distributed. If the thresholds are too high, an increasing number of problems will be solved with few iterations, but discretization will increasingly blur the dependency of the sampled distance on the input characteristics of the trips (e.g. the relation between distance and modal travel time, in this case).

The values of 100 and 200 m have been chosen based on experience and after applying the algorithm multiple times on the presented data sets during the development process. Usually, below these thresholds, we recognize a drop of resolved assignment problems and a strong increase in runtime. A value beyond seems too rough as we assume that a distance of 100 m would not make a large difference in finding a feasible destination for the active modes, and 200 m seems far enough to acknowledge that people may need to find a parking spot or use a specific transit stop to reach the actual destination. In any case, the values are arbitrarily defined and merely control the number of attempts that the algorithm will perform to find a valid assignment. In another implementation, relative thresholds could be used as well.

3.3. 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 of the assigned activity chains. In fact, preliminary experiments have shown that the algorithm tends to skew the distance distribution. This can be explained by the constrained way in which feasible distances are sampled (see Section 2.2). Equation (3) evaluated for two distances gives $|d_1 - d_2| \leq c$. Here, d_1 and d_2 have been sampled from their respective trip-dependent distance distributions and c is the direct distance between the framing activities. Hence, with increasing values for both distributions, the probability decreases that the constraint can be fulfilled. Furthermore, it depends on how often samples from different trip types are evaluated in combination. Conceptually, a distribution with a focus on shorter distances will be skewed towards longer distances, while a distribution with focus on longer distances will be skewed towards shorter ones if evaluated in combination. A solution to overcome the skewing effect is to reweight the distances in all distributions. How to perform this reweighting efficiently and in a mathematically exact way is a complex problem and could provide an interesting pathway for future research.

Table 1. Reweighting factors for the input distance distributions.

	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

For our practical use case, we define a method to reweigh the input distributions based on skew factors that can be calibrated by the user. Hence, as input to the algorithm, we do not use the exact input distributions as shown in Figure 5, but we perform a resampling of the data points according to the following scheme. 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 ($\sum_i f(d_i) = 1$). We then perform a linear reweighing 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 reweighing 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. They have been found by manually setting the values, comparing the model outputs as in the following section, and repeating the process until an acceptable fit was found. In future adaptations of the algorithm, this process may be automated.

3.4. Results

The location assignment model was run with the parameters and input as specified above. Figure 6 shows the resulting distance distribution in comparison to reference data from the HTS. 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–Smirnov 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 right part of Figure 6. Because of the binning of the data, the plot of the reference data appears as a step function for short distances.

Figure 7 shows the mean, median and 90% quantile of mode-independent 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 number 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 of the performance of the algorithm. With the convergence rate presented in Table 2 we obtain a good match in distance distributions. It will be interesting to explore how changing the convergence thresholds would affect the precision and runtime of the algorithm.

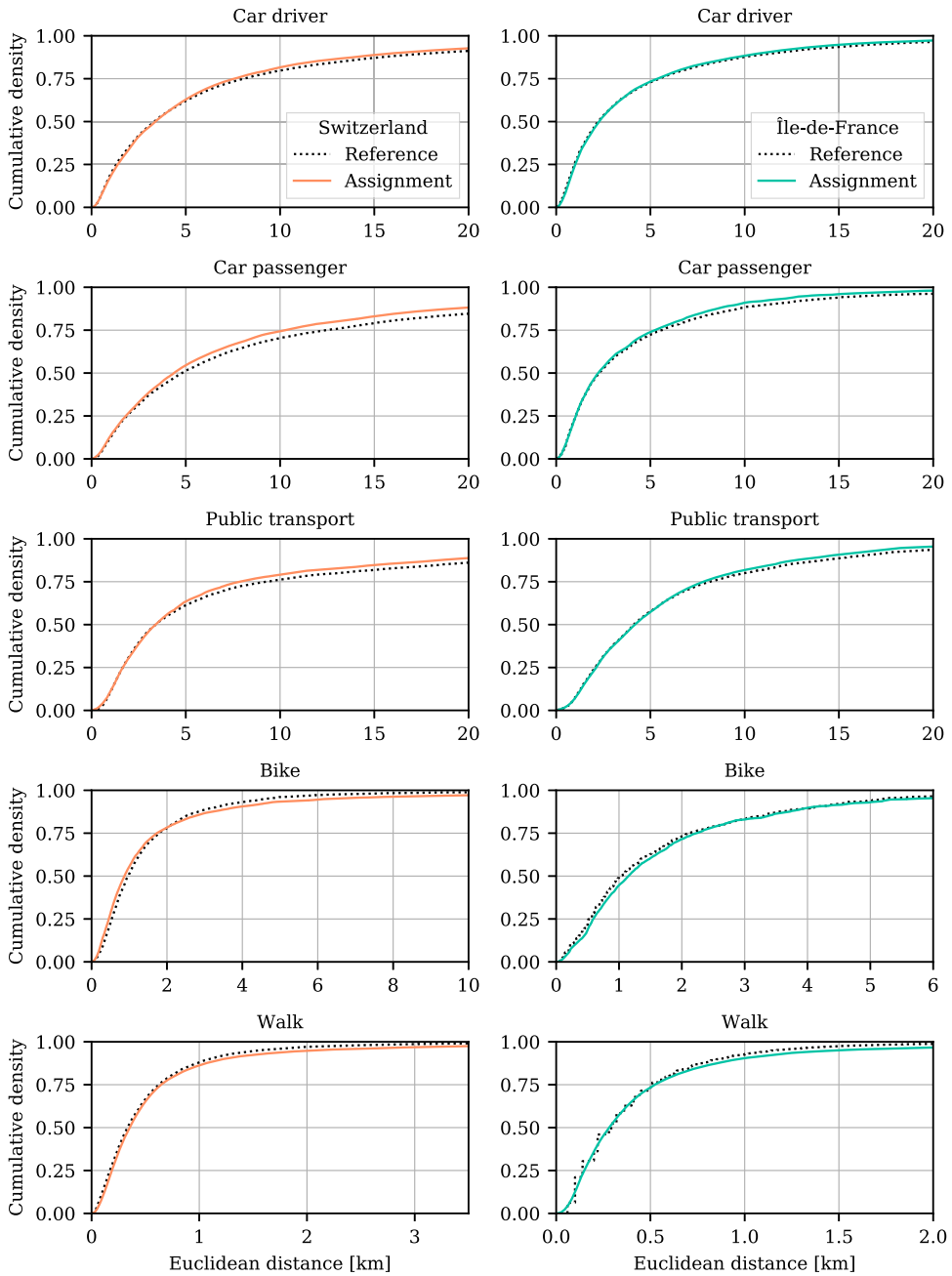


Figure 6. Comparison of assignment results with HTS data in terms of Euclidean distance distributions by mode.

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.

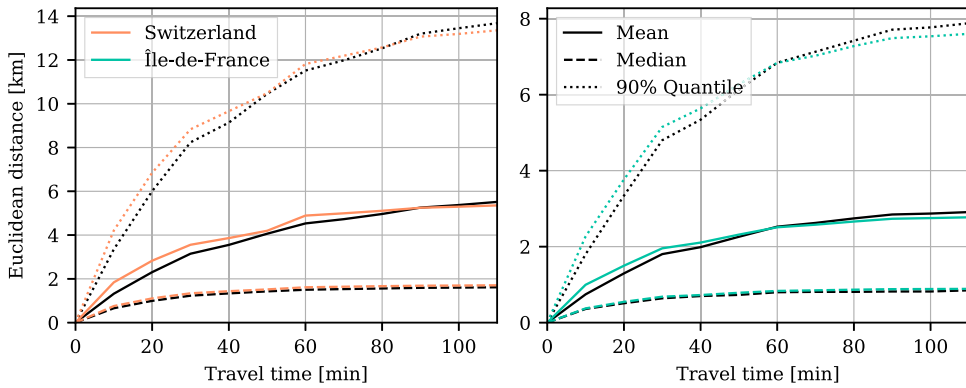


Figure 7. Comparison of Euclidean distance distribution for specific travel time bins by mean, median and 90% centile. The reference data is given in black.

Table 2. Key metrics for performance and convergence of the algorithm.

	Switzerland	Île-de-France
<i>Performance</i>		
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
<i>Convergence</i>		
Feasible distance sampler	99.3%	98.7%
Relaxation	93.2%	92.4%
Discretization	98.3%	97.2%
Assignment	92.5%	91.0%
<i>Errors</i>		
Mean discretization error	92 m	89 m
Mean excess error	19 m	29 m

4. 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 algorithm a data-driven 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 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. Given these data sets, which are usually easy to obtain, researchers and practitioners can set up the code in a couple of minutes and the runtimes we report in Table 2 for

fairly large agent populations give an idea of the expected run time. Note that only a very limited calibration effort is needed, and no models need to be estimated prior to applying the algorithm. This is a major difference compared to elaborate methods around the concept of space-time prisms and potential path areas, where initial input on travel times is needed.

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 attractiveness 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 attractiveness measure into the discretization process will be an interesting task for the future. Another interesting aspect that goes beyond a simple sense of attractiveness 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 simpler 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. Furthermore, including network travel times for routing would then lead us back to potential path areas for which the present approach could be regarded as a sampling approach.

5. Conclusion

In conclusion, we have presented a novel location assignment algorithm that, based on limited information on the chain structure and Euclidean distances between activities can provide a starting solution for more evolved modeling approaches. Contrary to discrete choice models, gravity models or even more advanced modeling techniques, the presented approach requires only very limited calibration effort as it does not aim to recover behaviorally correct decisions but rather to reconstruct activity locations that provide a high level of spatial integrity in terms of distances between activities. It is, therefore, highly useful to prepare input data for agent-based or activity-based transport models. The algorithm 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 straightforward, we give a non-comprehensive list of potential improvements that can be made to the basic version that is presented in this paper.

Acknowledgments

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 (BFS 2016), the national household survey (BFS 2017), the enterprise census (BFS 2019), and the national household travel survey (BFS and ARE 2018). 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 (INSEE 2018) and enterprise census (INSEE 2019). It further draws from the regional household travel survey for Île-de-France conducted by OMNIL, DRIEA and STIF (IDFm, OMNIL, and DRIEA 2013). The authors would like to thank the anonymous reviewers for their invaluable comments and feedback.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Sebastian Hörl  <http://orcid.org/0000-0002-9018-432X>

Kay W. Axhausen  <http://orcid.org/0000-0003-3331-1318>

References

- Anda, Cuauhtemoc, Sergio A. Ordonez Medina, and Kay W. Axhausen. 2021. "Synthesising Digital Twin Travellers: Individual Travel Demand From Aggregated Mobile Phone Data." *Transportation Research Part C: Emerging Technologies* 128: 103118.
- 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.
- Arentze, T., and H. Timmermans. 2007. "A Multi-Agent Activity-Based Model of Facility Location Choice and Use." *disP* 170: 0.
- Balac, Milos, Felix Becker, Francesco Ciari, and Kay W. Axhausen. 2019. "Modeling Competing Free-floating Carsharing Operators: A Case Study for Zurich, Switzerland." *Transportation Research: Part C* 98: 101–117.
- BFS. 2016. "Population and Households Statistics." Technical Report. Swiss Federal Office of Statistics (BFS), Neuchâtel.
- BFS. 2017. "Structural Survey." Technical Report. Swiss Federal Office of Statistics (BFS), Neuchâtel.
- BFS. 2019. "Statistik der Unternehmensstruktur." Technical Report. Swiss Federal Office of Statistics (BFS), Neuchâtel.
- BFS and ARE. 2018. "Mikrozensus Mobilität und Verkehr." Technical Report. Swiss Federal Office of Statistics (BFS) and Federal Office for Spatial Development (ARE), Neuchâtel.
- Erath, A., N. Frank, R. Lademann, and K. W. Axhausen. 2007. "The Impact of Travel Time Savings on Shopping Location Choice or How Far Do people Go To Shop Cheaply?" In *14th International Conference on Recent Advances in Retailing and Service Science*, San Francisco.
- Filippo, S., M. C. González, A. Maritan, and A.-L. Barabási. 2012. "A Universal Model for Mobility and Migration Patterns." *Nature* 484: 96–100.
- Guo, J. Y., and C. R. Bhat. 2007. "Operationalizing the Concept of Neighborhood: Application to Residential Location Choice Analysis." *Journal of Transport Geography* 15: 31–45.
- Hörl, Sebastian, and Milos Balac. 2021. "Synthetic Population and Travel Demand for Paris and Île-de-France Based on Open and Publicly Available Data." *Transportation Research Part C: Emerging Technologies* 130: 103291.

- Hörl, Sebastian, Milos Balac, and Kay W. Axhausen. 2019. "Dynamic demand estimation for an AMoD system in Paris." *IEEE Intelligent Vehicles Symposium 2019*, Paris.
- Hörl, Sebastian, Felix Becker, and Kay W. Axhausen. 2021. "Simulation of Price, Customer Behaviour and System Impact for a Cost-covering Automated Taxi System in Zurich." *Transportation Research Part C: Emerging Technologies* 123: 102974.
- Horni, Andreas, Kai Nagel, and Kay W. Axhausen, eds. 2016. *The Multi-Agent Transport Simulation MATSim*. London: Ubiquity.
- Horni, A., D. M. Scott, M. Balmer, and K. W. Axhausen. 2009. "Location Choice Modeling for Shopping and Leisure Activities with MATSim: Combining Microsimulation and Time Geography." *Transportation Research Record* 2135: 87–95.
- IDFm, OMNIL, and DRIEA. 2013. "Enquête globale de transport." Accessed Mar 1 2020.
- INSEE. 2018. Individus localisés au canton-ou-ville en 2015.
- INSEE. 2019. Base permanente des équipements.
- Jiang, Shan, Yingxiang Yang, Siddharth Gupta, Daniele Veneziano, Shounak Athavale, and Marta C. González. 2016. "The TimeGeo Modeling Framework for Urban Mobility Without Travel Surveys." *Proceedings of the National Academy of Sciences* 113 (37): 5370–5378.
- Justen, Andreas, Francisco J. Martínez, and Cristián E. Cortés. 2013. "The Use of Space-time Constraints for the Selection of Discretionary Activity Locations." *Journal of Transport Geography* 33: 146–152.
- Kusumastuti, Diana, Els Hannes, Davy Janssens, Geert Wets, and Benedict G. C. Dellaert. 2010. "Scrutinizing Individuals' Leisure-shopping Travel Decisions to Appraise Activity-based Models of Travel Demand." *Transportation* 37 (4): 647–661.
- Ma, Tai-Yu, and Sylvain Klein. 2018. "Bayesian Networks for Constrained Location Choice Modeling Using Structural Restrictions and Model Averaging." *European Journal of Transport and Infrastructure Research* 18 (1): 91–111.
- 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.
- Masucci, A. P., 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: 0.
- McFadden, D. 1978. "Modeling the Choice of Residential Location." In *Spatial Interaction Theory and Planning Models*, edited by A. Karlqvist, L. Lundqvist, F. Snickars, and J. Weibull, 75–96. Amsterdam: North Holland.
- Miller, E. J., and M. J. Roorda. 2003. "A Prototype Model of Household Activity/Travel Scheduling." *Transportation Research Record* 1831: 114–121.
- Pozsgay, M. A., and C. R. 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: 47–54.
- Sá, C., R. J. G. M. 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.
- Schirmer, P., M. A. B. van Eggermond, and K. W. Axhausen. 2014. "The Role of Location in Residential Location Choice Models: A Review of Literature." *Journal of Transport and Land Use* 7: 3–21.
- Sener, I. N., R. M. Pendyala, and C. R. 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.
- Vitins, B. J., A. Erath, and K. W. 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.
- Yang, Y., C. Herrera, N. Eagle, and M. C. González. 2014. "Limits of Predictability in Commuting Flows in the Absence of Data for Calibration." *Scientific Reports* 4: 5662.
- Yoon, Seo Youn, Kathleen Deutsch, Yali Chen, and Konstadinos G. Goulias. 2012. "Feasibility of Using Time-space Prism to Represent Available Opportunities and Choice Sets for Destination Choice Models in the Context of Dynamic Urban Environments." *Transportation* 39 (4): 807–823.