ISCTE IUL
REPOSITÓRIO
INSTITUTO UNIVERSITÁRIO DE LISBOA

Repositório ISCTE-IUL

Deposited in *Repositório ISCTE-IUL*:
2026-09-30

Deposited version:
Publisher Version

Peer-review status of attached file:
Peer-reviewed

Citation for published item:
Simões, R., Lopes, A. & Brito e Abreu, F. (2026). A genetic algorithm for pedestrian origin-destination flow estimation. In Filipe Araújo, Eugénio Ribeiro (Ed.), INForum 2025: Atas do 16º Simpósio Nacional de Informática (pp. 188?200). https://doi.org/10.5281/zenodo.19640654

Further information on publisher's website:
10.5281/zenodo.19640654

Publisher's copyright statement:
This is the peer reviewed version of the following article: Simões, R., Lopes, A. & Brito e Abreu, F. (2026). A genetic algorithm for pedestrian origin-destination flow estimation. In Filipe Araújo, Eugénio Ribeiro (Ed.), INForum 2025: Atas do 16º Simpósio Nacional de Informática (pp. 188?200). https://doi.org/10.5281/zenodo.19640654, which has been published in final form at https://dx.doi.org/10.5281/zenodo.19640654. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

Use policy

Creative Commons CC BY 4.0
The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Serviços de Informação e Documentação, Instituto Universitário de Lisboa (ISCTE-IUL)
Av. das Forças Armadas, Edifício II, 1649-026 Lisboa Portugal
Phone: +(351) 217 903 024 | e-mail: administrador.repositorio@iscte-iul.pt
https://repositorio.iscte-iul.pt

A Genetic Algorithm for Pedestrian Origin-Destination Flow Estimation

Rodrigo Simões, Adriano Lopes, and Fernando Brito e Abreu

ISTAR-IUL, Instituto Universitário de Lisboa (Iscte-IUL), Lisbon, Portugal
{rjbss,adriano.lopes,fba}@iscte-iul.pt

Abstract. Urban environments are facing increasing complexity, mainly due to rising populations, vehicle usage, and tourism. This intensifies urban flows, including traffic, daily commuting, and tourist movements, posing challenges for congestion management, environmental sustainability, and overall quality of life. In popular tourist destinations, where urban infrastructure often limits carrying capacity, managing the combined flows of residents and visitors becomes particularly difficult. This can degrade both the visitor experience and the mobility of local communities.
To address these challenges, mobility flows can be modeled using trajectory data, typically obtained via GPS. However, such data is costly and often unavailable. In this study, we leverage a more accessible but less granular dataset from the Lisboa Aberta initiative, which provides aggregated mobile device counts across Lisbon by location and time. While this dataset lacks individual identifiers or trajectory information, we aim to reconstruct pedestrian origin-destination (OD) flows from it.
We frame this as a dynamic OD estimation problem, which is an inherently under-determined optimization task. To address this, we present a genetic algorithm constrained by a simplified pedestrian mobility model. This approach enables us to infer plausible mobility patterns from aggregated data, offering a cost-effective method for urban flow analysis and contributing to more sustainable urban planning.

Keywords: Origin-Destination Flows · Origin-Destination Estimation · Pedestrian Flows · Genetic Algorithm

1 Introduction

Many urban environments are becoming increasingly complex, with growing populations, increased vehicle use, and a rise in the number of tourists. The intensity of urban flows, such as traffic, the pendular movement of residents, and tourist flows between points of interest, is a significant reminder of this. Efficient management of these flows is crucial to prevent congestion, minimize environmental impact, enhance the quality of life for residents and visitors, and promote local sustainability.

In overcrowded scenarios, common in popular tourist destinations, where the urban fabric reduces the carrying capacity [1], managing visitor flows in

combination with flows of locals becomes problematic. This degrades the visitor experience and hinders the mobility of residents and traders in local communities.

These flows can be modeled using individual trajectory data, which can be obtained through technologies such as GPS; however, this approach has a higher cost compared to less granular data [10]. Thanks to the Lisboa Aberta initiative, we have a dataset with mobile device counts for the city of Lisbon. It falls into this 'less granular data' category since it does not have sufficient information so that it is possible to reconstruct mobility flows straightforwardly, as it does not contain individual IDs, pseudonymized or otherwise, nor any trajectory information. Instead, it has aggregated counts of mobile devices for each considered location and date/time. A detailed description of the dataset will be provided in Subsection 4.1; for now, it suffices to note that reconstructing flows from this type of data is not a trivial task. We simplify this problem in two ways: (i) we aim to reconstruct origin-destination flows; (ii) we only consider pedestrian mobility. Thus, our problem is a dynamic origin-destination estimation problem, which is an under-determined optimization problem. We developed a genetic algorithm to address this problem, utilizing a simplified model of pedestrian mobility to constrain the solutions and mitigate the issue of underdetermination.

This paper is structured as follows: Section 2 presents work related to origin-destination flows and pedestrian flow modeling. In Section 3, the genetic algorithm used to estimate origin-destination flows is described in detail. Then, in Section 4, we describe the dataset used and present the results of our proposed solution on that dataset. Finally, Section 5 concludes the paper with a discussion of the results, addressing its limitations and proposing future work.

2 Background & related work

Urban mobility flows can be classified from the perspective of travel mode into traffic flow, public transit flow, crowd (or pedestrian) flow, among others [13]. In this work, we focus on pedestrian flows.

2.1 Modeling pedestrian mobility

Our proposed algorithm uses a custom-made model of pedestrian mobility that assigns probabilities to transitions between regions. Other approaches exist for modeling pedestrian flows, which could, in theory, replace our simplistic model and therefore generate higher-quality solutions. One approach is agent-based modeling (ABM), which enables agents to interact with each other, capturing various effects that arise in real person-to-person interactions that may not be possible to capture through other approaches [12]. A well-known method for modeling individual agents is through the social forces model, where the environment (e.g., other pedestrians or borders) can be viewed as exerting a force on agents, thereby influencing their motivations and, consequently, their behavior [8].

Another approach to modeling pedestrian flow is network-based, where the street network is treated as a graph and movement is based on connectivity and various parameters. One traditional algorithm that has been adapted to street networks is the betweenness index, where the "betweenness" of a street segment captures the estimated foot traffic passing through it [11]. The network-based approach is typically much more efficient than the ABM approach.

2.2 Origin-destination flows

As mentioned in Section 1, we focus on origin-destination (OD) flows. We divide an urban area into distinct non-overlapping regions, and we want to know, for each region, how many pedestrians traveled to each other region. When we group these pedestrian counts in a matrix, where each row represents an origin and each column represents a destination, we obtain an origin-destination (OD) matrix.

Rong et al. [10] list four problems whose goal is to obtain OD matrices: OD prediction, OD construction, OD estimation, and OD forecasting. In OD prediction, the OD matrix is partially filled, and the goal is to find the missing OD flows. In OD construction, no OD flow information is available for constructing the OD matrix. In OD estimation, there is also no OD flow information; however, a set of observations from the real world (e.g., pedestrian counts obtained by a movement sensor) is available, from which the OD flows can be estimated. In OD forecasting, we are given historical OD matrices, and the goal is to obtain new OD matrices for future time periods.

Our goal in this work is to obtain OD matrices based on a dataset that has counts of mobile devices, which we assume are strongly correlated with pedestrian counts. This dataset contains the observations that make our problem an OD estimation problem. It might be relevant to transcribe Rong et al.'s definition of OD estimation (mathematical symbols omitted):

Given a set of observed temporal traffic counts or other relevant observations collected at various locations within a transportation network (e.g., road segments, intersections, or sensor-equipped locations), the objective is to infer the underlying OD flows—that is, the number of trips between different origin and destination pairs—that generated the observed traffic pattern.

This definition is specific to traffic OD flow, as it is the type of flow that has been more extensively studied in the context of OD estimation. One work that utilizes OD estimation for pedestrian flow (albeit in an indoor scenario rather than an urban context) employs techniques developed for traffic flow [3], which highlights the centrality of traffic flows in the context of OD estimation research.

OD flows can also be categorized by their temporal nature. A static OD matrix refers to patterns that maintain a stable structure over a period, such as commuting flows. In contrast, a dynamic OD matrix refers to a time-varying pattern of mobility. In our work, we consider OD matrices based on two sets of observations that are consecutive in time, thus making our problem a dynamic OD estimation (DODE) problem. The most common approach to this problem is to formulate it as a bi-level optimization problem [2], where the inner problem takes a proposed OD matrix as input and solves a Dynamic Traffic Assignment

(DTA) problem. In the outer problem, the matrices are adjusted to minimize the discrepancy between the solution of the inner problem and the traffic observations. Several approaches have been proposed for solving the bi-level optimization problem, including genetic algorithms [5]. Our work's approach also uses a genetic algorithm, although it does not employ a bi-level optimization problem formulation.

3 Methodology

Estimating OD matrices from counts along different regions is an underdetermined problem, as one can generate an infinite number of matrices that fit the observations. This means that something more is required to constrain the optimization process. In addition to the observations, our approach utilizes a pedestrian mobility model. The intuition behind our optimization is that we can fit a matrix to the observations and the mobility model. The accuracy of the generated matrix depends on the accuracy of the mobility model employed.

Our genetic algorithm requires crowding data (observations) for two consecutive time periods. The generated matrix will then contain information on how many people moved from one region to another between the first time period and the second.

Algorithm 1 Genetic Algorithm

1: Initialize population P with $PopSize$ random individuals
2: **while** termination condition not met **do**
3: Evaluate fitness of each individual in P
4: Select parents from P based on fitness
5: Apply crossover to parents to produce offspring
6: Apply mutation to offspring
7: Evaluate fitness of offspring
8: Select individuals for the next generation using steady-state selection (i.e. few individuals are replaced)
9: Update population P with new individuals
10: **end while**
11: Return the best individual found

Algorithm 1 shows our genetic algorithm. We used the PyGAD library in our implementation. The individuals in the population are candidate OD matrices, and our solution is the individual with best fitness after the last generation. For our experiments described in Section 4, we defined $PopSize$ and the maximum number of generations in an ad-hoc way, running the algorithm multiple times on different sets of observations and analyzing the curve of the fitness function across generations. Besides setting a maximum number of generations $MaxGenerations$, we implemented an alternative termination condition: after

each generation, if in the window of the last W generations there is a plateau in the fitness improvement of the best individuals, that is, the distance between the best fitnesses is below a threshold T, then the algorithm is stopped early.

3.1 Initialization

We begin by generating an initial solution for the genetic algorithm. Given N regions, the dimension of our OD matrix is $(N + 1, N + 1)$. The extra line represents people entering from outside the considered regions, and the extra column represents people exiting the considered regions.

Our initialization function assigns a constant value to all elements of the matrix (except one, which we will soon discuss). That value corresponds to the sum of crowding values for the first time period, divided by the number of elements in the matrix.

What about the last element in the matrix, that is, row $N + 1$ and column $N + 1$? This would represent individuals who were not in any of the regions of interest during the first period and subsequently moved outside of the regions of interest into the second time period. Conceptually, this means almost everybody, everywhere, which falls outside the scope of our OD estimation task. Therefore, this value is ignored in the iterative steps of the genetic algorithm (mutation and fitness evaluation). For the purposes of our implementation, we initialize it with 0. Using a null value might be more appropriate, but we use 0 due to the constraints of the libraries used in the implementation.

3.2 Mutation function

For each individual (matrix) in the population, we select one element (excluding the last) and use randomness to modify it. First, we obtain a random value between 0 and 1, which we can call *rand*. Given the chosen element E, we obtain its new value through the following formula:

$$(E + 1) \times rand \times 2$$

This is the same as multiplying $E + 1$ by a random number in the interval $[0, 2]$. We add one to E to ensure this factor is not zero. Otherwise, an element with value zero would be "stuck" at zero even if it were selected multiple times to be mutated. We multiply the random value by two so that there is an equal likelihood of making the value of $E + 1$ smaller or larger.

3.3 Objective function 1: fit to observations

The fitness function of our genetic algorithm is a sum of two objective functions. We want to minimize these functions. Therefore, the fitness function is actually the sum of the two objective functions multiplied by -1, so that by maximizing the fitness we minimize the objective functions.

We will now describe the first of the two objective functions. Its purpose is to guide the optimization process in a way that the resulting matrix is consistent with the observations, that is, the crowding data for the two consecutive time periods.

Let D be the matrix that holds the crowding data. This matrix has dimension $(2, N)$, where N is the number of regions. $D_{1,:}$ holds the data for the first time period, while $D_{2,:}$ holds the data for the second time period. Let M be another matrix. This is the OD matrix, which means its rows represent origins and its columns represent destinations. Recall that its dimension is $(N+1, N+1)$, with the extra line to represent people coming from outside the regions of interest and the extra column to represent the people going to outside the regions of interest. With these definitions, we can express the objective function as follows:

$$\sum_{i=1}^{N} \left(|D_{1,i} - \sum_{j=1}^{N+1} M_{i,j}| + |D_{2,i} - \sum_{j=1}^{N+1} M_{j,i}| \right)$$

The idea is to sum the distances between the actual observations and the hypothetical observations implied by the OD matrix. For example, for region 1, the implied observation for time period 1 is the sum of line 1 of M, while the implied observation for time period 2 is the sum of column 1 of M. The lower this objective function is, the closer M is to the real observations.

3.4 Objective function 2: fit to model

When minimizing the previous function, we obtain an OD matrix that is consistent with the observations. However, many different matrices could fit the observations. Indeed, when running the genetic algorithm without a second objective function, the results vary widely depending on the seed of the random number generator. We need a second objective function that further constrains the range of possible solutions. Our approach is to use a pedestrian mobility model for this second function. We will discuss the mobility model we used later, although other mobility models could also be used for this second objective function. Indeed, all that is required of the model is that it provides us with the probabilities of a person moving from one region to every other region.

What we need, more precisely, is to know what the probability is of a person who was in region R_1 during time period 1 to move to region R_2 during period 2, which we can denote by $p(R_1 \rightarrow R_2)$. We also need to know the reverse of this: given that a person moved to region R_2 in time period 2, we want to know the probability that it came from region R_1: $p(R_2 \leftarrow R_1)$.

Once we have these probabilities, we can construct an objective function that, when minimized, brings the OD matrix M closer to the model. The expression of this function is as follows:

$$\sum_{i=1}^{N}\left(\sum_{j=1}^{N}\left(|M_{i,j}-p(R_i\rightarrow R_j)\sum_{k=1}^{N+1}M_{i,k}|\right)+|M_{i,N+1}-p(R_i\rightarrow R_{out})\sum_{k=1}^{N+1}M_{i,k}|\right)$$
$$+\sum_{j=1}^{N}|M_{N+1,j}-p(R_j\leftarrow R_{out})\sum_{k=1}^{N+1}M_{k,j}|$$

First, we loop through the first N lines, and within this loop, we loop again through the first N columns. For each element, we calculate the distance between that element's value and the probability of the OD flow represented by that element, multiplied by the sum of the element's line in the matrix. If the elements along a line i are proportional to the probabilities of the OD flows that have i as their origin, then the result of this part of the expression will be 0, meaning that the line is perfectly fitted to the model.

At the end of each outer iteration, we then consider the element $M_{i,N+1}$, which is in the last column, representing the OD flow between origin i and locations outside the regions of interest. The idea here is the same, and we use $p(R_i \rightarrow R_{out})$ to mean "probability that a person moved from region i to any location outside the regions of interest."

Finally, outside the loop, we consider the last line of the matrix. We loop through that line, and for each element (except the last), we calculate the distance between its value and the product between the sum of its column and the probability that a person moved into the region represented by that column, knowing that it came from outside the regions of interest.

3.5 Pedestrian mobility model

We have developed an algorithm that provides a probabilistic model of pedestrian mobility, which can be utilized for the second objective function of the genetic algorithm. For a given origin region, our algorithm assigns a probability to each possible destination region. That is the probability that a pedestrian, on a random walk, starting from a random location in the origin region, ends up somewhere in the destination region. More sophisticated mobility models, such as those we mentioned in Subsection 2.1, can be used instead, as long as they provide the required probabilities of moving from one region to another.

Our algorithm utilizes OpenStreetMap geographic data. The different regions of interest are assumed to be represented as polygons rather than as points. Thus, for spatial topologies that use points rather than polygons, such as those based on movement sensors or cameras, the algorithm presented below cannot be used. In particular, for the dataset we used, the regions are 200 by 200 square meter cells in Lisbon; adaptations in one of the algorithm's steps may be required for different topologies.

The crux of our algorithm is to determine the isochrone of the origin cell. Before we present our definition of "isochrone of a cell", we will present the

concept of isochrone of a point. The isochrone of a point is a line that delimits the possible positions that a moving agent can reach in a given time frame starting from that point. We assume that the agent is a pedestrian walking at an average speed of 1.3 meters per second, which is the average speed of pedestrians as stated by [7]. Isochrones can be obtained using OpenStreetMap data through tools such as GraphHopper, which is the one we use.[1]

Here we extend this idea of an isochrone of a point to an isochrone of a cell, that is, a line that delimits all the possible positions that a pedestrian can end up in, starting from somewhere in the given origin cell. We can obtain such a line by merging the isochrones of several points along the edges of the cell. For large cells, this merging may not cover the cell itself, which should be part of the isochrone of the cell; therefore, the isochrones are also merged with the cell.

The choice of the number and placement of points from which to generate isochrones depends on the specific characteristics of the shape and geography of the regions of interest. For topologies different than ours, alternative methods may be required to determine the placement of the points. In our scenario, we begin by placing points at the corners of the cell and the midpoints of the edges, resulting in a total of 8 points. However, some of these points may fall on unwalkable terrain, such as the middle of a road or a building. Therefore, we first apply the carrying capacity algorithm described by [1] to the cell so that we only consider the walkable parts of the edges of the cell. Then, when a point is not walkable, we project it towards the nearest walkable point along its edge.

Once we have the set of points in walkable zones, we determine their isochrone areas and combine them, together with the cell, into a single area, which we call the isochrone area of the cell. This area encompasses all the possible destinations that a pedestrian can reach within a given time frame, walking at an average walking speed. For our case, since we have crowding data for periods of 5 minutes, we set the time frame of the isochrones also to be 5 minutes.

Once we have the isochrone area of a cell, we run the walkable area algorithm on it. Then, we quantify the area of the resulting geometry. Given this, we can calculate the probability of transition from the origin cell to a destination cell by intersecting the walkable area of the isochrone with the destination cell. Then, we quantify the area of the intersection. The area of the intersection divided by the walkable area of the isochrone gives us the final result: the probability of transition from the origin cell to the destination cell.

This can be visualized in Figure 1, where the blue square represents the origin cell, the green square represents a destination cell, and the walkable isochrone area of the origin cell is shaded in brown. The probability of transitioning from the blue square to the green square is given by the area of the intersection of the brown area with the green square, divided by the total area of the polygon isochrone (the brown area).

As seen earlier, our second objective function requires not only the probability of a person going to a destination cell given that they were in a specific origin cell,

[1] OpenTripPlanner is another option for obtaining isochrones. We chose GraphHopper because its executable is slightly smaller than OpenTripPlanner's.



Fig. 1. Walkable isochrone area (brown) of a cell (blue)

but also the inverse of that: the probability of having come from an origin cell given that the person arrived at a specific destination cell. We can use a similar approach with isochrones and walkable areas to determine the probabilities for this inverted model. Given a destination cell, its isochrone area represents all the locations from which a pedestrian can have come. To determine the probability that a pedestrian originated from a specific cell, we calculate the intersection of the origin cell with the isochrone area of the destination cell, calculate its area, and then divide it by the walkable area within the isochrone cell. The reader may have noticed that this algorithm is identical to the one described immediately before, which determines the probability of transition from one cell to another, given a chosen origin cell. Using the notation introduced in Subsection 3.4, we can say that $p(A \rightarrow B) = p(B \leftarrow A)$.

3.6 OD matrix visualization

We have developed a simple web-based tool to visualize OD matrices, shown in Figure 2. This visualization does not use the extra line and column representing pedestrians coming from or to outside the regions of interest. In the figure, the cell with the black border and without an arrow is the selected origin. We are visualizing the row in the OD matrix that corresponds to the selected cell. The colors in the heatmap range from green (the row element with the minimum value) to red (the row element with the maximum value). When we place the cursor over a cell, a tooltip displaying the exact OD flow for that cell is shown. The arrows inside the cells serve to provide a more intuitive understanding of what is being visualized, showing the direction of the OD flows, starting from the selected cell. It is also possible to visualize columns of the OD matrix rather than rows by toggling the switch shown on the top left. When "To" is selected, the selected cell acts as a destination rather than an origin, and the arrows point toward the selected cell rather than from it.



Fig. 2. Visualization of a row of an OD matrix

4 Evaluation

4.1 Dataset

We use a dataset provided by the Lisboa Aberta for evaluation purposes, based on data collected by a mobile network operator. It divides the Lisbon municipality into 3,743 square cells, each measuring 200 meters by 200 meters. For each cell, several metrics related to the counts of detected mobile devices are available. Values below 10 devices are not reported and are instead marked as missing. The data is collected in 5-minute intervals.

The most relevant metrics for flow estimation are: *C1*: number of distinct terminals in the cell; *C3*: number of distinct terminals that remained in the cell; *C5*: number of distinct terminal entries in the grid; *C6*: number of distinct terminal exits in the grid.

It is not clear how these different metrics are related. Considering two consecutive time periods, where $t1_A$ is metric A in the first time period and $t2_A$ is metric A in the second time period, we could expect $t2_{C1} = t1_{C1} + t2_{C5} - t2_{C6}$. However, this relationship does not hold. We decided to use the C1 metric exclusively instead. We also make the simplifying assumption that these device counts represent the pedestrian counts for the purposes of our genetic algorithm, as we do not have a means of grouping these counts by different modes of transport.

In our experiments, we used $N = 25$ cells, organized geographically as a 5×5 grid, comprising 1,000,000 square meters. Thus, our OD matrices are of size $(25 + 1)^2 = 676$.

4.2 Results

We compared different crossover functions, including a custom-built function that, for each row index, selects a row from either parent with equal probability. We tested other crossover functions, built into PyGAD. These functions are: single-point crossover, two-point crossover, scattered crossover, and uniform crossover. Finally, we also tested the algorithm without any crossover. For this evaluation, we used five pairs of consecutive observations, testing each crossover type twice for each pair, with a different seed each time to account for the effects of randomness. Using the notation introduced in Section 3, we set $PopSize = 500$, $MaxGenerations = 1000$, $W = 50$, and $T = 50$. As stated earlier, these values were determined experimentally in an ad-hoc manner. The crossover functions with the best performance (best fitness) were scattered and uniform. These are mathematically equivalent, but scattered crossover had the fastest computation times. Therefore, in the following evaluation, we use scattered crossover.

The ideal way to validate the estimated OD matrices would be to compare them with ground-truth OD matrices. However, the dataset that we use does not contain OD information. Our experimental evaluation employs an alternative approach: since we are using a genetic algorithm to estimate OD matrices, which

relies on randomness, we aim to assess the randomness of the results by comparing the structural similarity of matrices generated for the same observations. We estimated matrices for 20 pairs of consecutive observations. For each pair, we generated 5 matrices using different seeds for pseudo-random generation, and we made $5 \times 4 = 20$ pairwise comparisons between matrices using normalized Levenshtein distance for OD matrices (NLOD) [4]. The average NLOD across all pairs of observations is 0.35, with a minimum NLOD of 0.33 for one pair of observations and a maximum NLOD of 0.38 for another pair of observations. This means that randomness plays a fairly important role in the resulting structure of the OD matrices.

5 Conclusions & future work

We have presented a genetic algorithm for estimating OD pedestrian flow. Another contribution we made is a simple pedestrian mobility model that we use in the algorithm. As stated earlier, our algorithm can be enhanced by replacing our mobility model with a more sophisticated one, provided that the latter provides probabilities of transitions between regions. The enhancement of our mobility model is part of planned future work.

Our approach to the DODE problem differs significantly from existing literature in that we do not formulate it as a bi-level optimization problem, but rather utilize two objective functions within a genetic algorithm. We combine the two functions using summation instead of employing multi-objective optimization techniques, as these methods output a set of solutions rather than a single optimal solution. Although not mentioned in the body of this paper, we did experiment with multi-objective optimization with evolutionary algorithms. We found that selecting the solution with the best value for the first objective function, which approximates the matrix to the observations, yields a solution with fitness values similar to those of our genetic algorithm solution.

Our experimental results show that randomness plays a significant role in the results, despite the incorporation of the mobility model, which attempts to alleviate this problem.

As mentioned previously, obtaining datasets of aggregated counts rather than individual trajectories may be more cost-effective. Another advantage of our method is that the dataset that serves as input to the genetic algorithm is completely anonymized. Having said that, the output OD matrices can constitute a privacy risk in that they may reveal individuals' behaviors when combined with auxiliary information, such as publicly available demographics or social media check-ins [9]. Privacy in spatiotemporal data is an active area of research (see e.g. [6]) and is an important issue that we aim to explore in the future.

Acknowledgments. This work was supported by FCT - Fundação para a Ciência e Tecnologia, I.P., under ISTAR-Iscte projects UIDB/04466/2020, UIDP/04466/2020, and EUROSTIT / 10.54499/2024.07579.IACDC). The computational resources used were provided by the INCD, funded by FCT and FEDER under project GEMINI /

2024.07624.IACDC. We are particularly grateful to INCD's technical team for their timely support of our research needs.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. de Almeida, D.S., Simões, R., e Abreu, F.B., Lopes, A., Boavida-Portugal, I.: A Carrying Capacity Calculator for Pedestrians Using OpenStreetMap Data: Application to Urban Tourism and Public Spaces. In: Proceedings of the ENTER25 e-Tourism Conference. pp. 79–90. Springer Nature (2025). https://doi.org/10.1007/978-3-031-83705-0_7
2. Andersen, N.S., Chiarandini, M., Debrabant, K.: A comparison of different approaches to dynamic origin-destination matrix estimation in urban traffic (2022). https://doi.org/10.48550/arXiv.2202.00099
3. Bauer, D.: Estimating origin–destination-matrices depending on the time of the day from high frequent pedestrian entry and exit counts. IET Intelligent Transport Systems **6**(4), 463–473 (2012). https://doi.org/10.1049/iet-its.2011.0156
4. Behara, K.N., Bhaskar, A., Chung, E.: A novel approach for the structural comparison of origin-destination matrices: Levenshtein distance. Transportation Research Part C: Emerging Technologies **111**, 513–530 (2 2020). https://doi.org/10.1016/j.trc.2020.01.005
5. Bera, S., Rao, K.V.K.: Estimation of origin-destination matrix from traffic counts: the state of the art. Tech. rep. (2011)
6. Bhadani, R.: A survey on differential privacy for spatiotemporal data in transportation research (2024). https://doi.org/10.48550/arXiv.2407.15868
7. Bowman, B.L., Vecellio, R.L.: Pedestrian walking speeds and conflicts at urban median locations. Transportation Research Record pp. 67–73 (1994)
8. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Physical review E **51**(5), 4282 (1995). https://doi.org/10.1103/PhysRevE.51.4282
9. Kim, J.: Distod: A hybrid privacy-preserving and distributed framework for origin–destination matrix computation. Electronics **13**(22) (2024). https://doi.org/10.3390/electronics13224545
10. Rong, C., Ding, J., Li, Y.: An interdisciplinary survey on origin-destination flows modeling: Theory and techniques. ACM Computing Surveys (1 2024). https://doi.org/10.1145/3682058
11. Sevtsuk, A.: Estimating pedestrian flows on street networks: Revisiting the betweenness index. Journal of the American Planning Association **87**, 512–526 (2021). https://doi.org/10.1080/01944363.2020.1864758
12. Sevtsuk, A., Kalvo, R.: Predicting pedestrian flow along city streets: A comparison of route choice estimation approaches in downtown san francisco. International Journal of Sustainable Transportation **16**, 222–236 (2022). https://doi.org/10.1080/15568318.2020.1858377
13. Xie, P., Li, T., Liu, J., Du, S., Yang, X., Zhang, J.: Urban flow prediction from spatiotemporal data using machine learning: A survey. Information Fusion **59**, 1–12 (2020). https://doi.org/10.1016/j.inffus.2020.01.002