RESEARCH

Estimation of flow trajectories in a multi-lines transportation network

Guillaume Guex[1*], Romain Loup[2] and François Bavaud[1,2]

*Correspondence: gguex@unil.ch
[1]Department of Language and Information Sciences, University of Lausanne, Lausanne, Switzerland
Full list of author information is available at the end of the article

Abstract

Characterizing a public transportation network, such as an urban network with multiple lines, requires the origin-destination trip counts during a given period. Yet, if automatic counting makes the embarkment (boarding) and disembarkment (alighting) counts in vehicles known, it often happens that pedestrian transfers between lines are harder to track, and require costly and invasive devices (e.g., facial recognition system) to be estimated. In this contribution, we propose a method, based on maximum entropy and involving an iterative fitting procedure, which estimates the passenger flow between origins and destinations solely based on embarkment and disembarkment counts. Moreover, this method is flexible enough to provide an adaptable framework in case additional data is known, such as attraction poles between certain nodes in the network, or percentages of transferring passengers between some lines. This method is tested on toy examples, as well as with the data of the public transportation network of the city of Lausanne provided by its Transportation Agency (tl), and gives arguably convincing estimations of the transportation flow.

Keywords: multiline bus network; origin-destination flows; boarding and alighting counts; maximum entropy estimation; iterative proportional fitting

1 Introduction

Transportation networks determine our mobility, require a considerable amount of planning and resources, and elicit much public hopes and critics. They also constitute an endless source of inspiration in formal modeling and optimization, as attested in operations research (classical optimal transportation, maximum flow problem), quantitative geography and spatial econometrics (spatial navigation, multimodality, gravity models for flows), and machine learning (recent developments in regularized optimal transportation, such as color transfer or images interpolation; see e.g. [1]).

This contribution addresses a straightforward, yet central question in public transportation networks: given a network made of many train, bus, subway, or tram lines, how can one estimate the real trips made by the travelers, on the sole basis of the embarkment (boarding) counts and disembarkment (alighting) counts in each vehicle? Although estimating origin-destination flows is a much addressed issue in transportation modeling (see e.g [2] [3] [4] [5] and references therein), the specific problem addressed in this contribution seems, to the best of our knowledge, original.

Pedestrian transfers of travelers between different lines here constitute the missing information, which planners traditionally attempt to estimate using census, or more recently

with costly and invasive devices located at station, such as mobile phone tracking or facial recognition systems. In this article, we take a different approach, which is to produce the optimal estimation of transportation trajectories without additional data, using the principle of maximum entropy. The proposed solution consists of three consecutive steps: a maximum-entropy computation of the trip distributions, obeying marginal constraints and with a given prior; an update of the prior distribution by shrinking the components responsible for transfer overflow; and an update of marginal distributions. This iterative procedure clearly evokes the EM algorithm (see e.g. [6] [7]), with the first step corresponding to the "expectation step" and last two steps to the "maximization step". The first step only is required for solving the single line case, naturally much simpler but yet not trivial, and exhibiting a disembarking probability independent of the embarking stop (Markov property). This method also offers some flexibility, as it is possible to set a prior distribution taking into account attraction or repulsion poles among stops, and to fix hyperparameters limiting the number of transfers between lines.

As case studies, we test the proposed method on toy examples as well as with the data of the public transportation network[1] of the city of Lausanne, in Switzerland. Toy examples offer some kind of validation, as a transportation flow can be entirely set on toy networks and compared to the solution given by our algorithm. The real case scenario with the data from the Lausanne public transportation network demonstrates that this method is applicable on a real transportation network and can yield pertinent insights about traveler habits.

Section 2 introduces the notations and the formalism, as well as the statement of the problem and the proposed solution. Section 3 contains case studies with toy examples and the Lausanne public transportation network. Section 4 concludes the article. Data, codes and extensive results can be found in the GitHub repository of the article[2].

2 Notations and formalism

2.1 Lines, stops and junctions

Consider a *transportation network* made of *lines* numbered $\ell = 1,\dots,q$, of respective lengths (number of stops) l_ℓ. Opposite lines, that is parallel lines running in the back and forth directions are considered as distinct.

The $l = \sum_{\ell=1}^{q} l_\ell$ *stops* constitute the nodes of the transportation network. Each stop $i = 1,\dots,l$ belongs to a single line, and defines a unique next or forward stop $F(i)$ (unless i is the line terminus) and a unique backward stop $B(i)$ (unless i is the line start), both on the same line.

Let S_i denotes the *set of stops which can be reached from stop i outside lines connection* (with, e.g., an acceptable walking distance), excluding i itself. A stop i is referred to as an *isolated stop* if $S_i = \emptyset$, and to as a *junction* otherwise.

2.2 Edges, trips, and the incidence matrix

Two sorts of oriented edges are involved in the transportation network:

- *intra-line edges* $(i,j) = (i,F(i))$ belonging to a single line $\ell(i) = \ell(j)$
- *inter-line* or *transfer edges* (i,j) connecting different lines $\ell(i) \neq \ell(j)$, involving walks from junction i to $j \in S_i$. The *set of transfer edges* is denoted by T.

[1] https://www.t-l.ch/
[2] https://github.com/sliunil/tl_study

A *st-trip*, noted $[s,t]$, consists of entering into the network at stop s, and leaving the network at t, by following the shortest-path (i.e. achieving the minimum distance, minimum time, or minimum cost), supposed unique, leading from s to t.

The succession of edges (i,j) belonging to the st-trip, noted $(i,j) \in [s,t]$, is unique. Define the *edge-trip incidence matrix* as

$$\chi_{ij}^{st} = \begin{cases} 1 & \text{if } (ij) \in [s,t], \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Note that we can also forbid some aberrant trips across the network, for example, trips $[s,t]$ where (s,t) is a transfer edge (travelers making this trip do not actually do not use the transportation network). The *set of permitted trips* across the network is denoted by P, and can be defined following some conditions (see, e.g., section 3.2.1).

2.3 Transportation flows

Let x_{ij} count the *number of travelers using edge* (i,j) in a given period, such as a given hour, day, week or year. The edge flow x_{ij} is denoted by y_{ij} for an intra-line edge (i,j), and by z_{ij} for a transfer edge (i,j). By construction, $x_{ij} = y_{ij} + z_{ij}$, where $y_{ij} z_{ij} = 0$.

Let a_i, respectively b_i, the *number of passengers embarking*, respectively *disembarking* at stop i. By construction,

$$\begin{cases} y_{i,F(i)} = a_i \text{ and } b_i = 0 & \text{if } i \text{ is a line start,} \\ y_{B(i),i} = b_i \text{ and } a_i = 0 & \text{if } i \text{ is a line terminus,} \\ y_{i,F(i)} = y_{B(i),i} + a_i - b_i & \text{otherwise.} \end{cases} \tag{2}$$

Also, **a** and **b** must be consistent, in the sense that $A_{B(i)} \geq B_i$, where A_i (respectively B_i) is the *cumulated number of embarked (resp. disembarked) passengers* on the line under consideration, recursively defined as $A_{F(i)} = A_i + a_i$ (resp. $B_{F(i)} = B_i + b_i$). Moreover, $A_i = B_i$ at a terminal line stop i. This common value yields the total number of passengers transported by the line.

Let the *transportation flow* n_{st} denotes the number of passengers following an st-trip, that is entering the network at s and leaving the network at t by using the shortest-path. One gets from (1)

$$x_{ij} = \sum_{st} \chi_{ij}^{st} n_{st} \tag{3}$$

Among the passengers embarking in i, some transfer from another line, and some others enter into the network:

$$a_i = z_{\bullet i} + n_{i\bullet} \tag{4}$$

where "$\bullet$" denotes the summation over the replaced index, as in $n_{i\bullet} = \sum_{j=1}^{l} n_{ij}$. Similarly, among the passengers disembarking in i, some transfer to another line, and some others leave the network:

$$b_i = z_{i\bullet} + n_{\bullet i} \tag{5}$$

By construction

$$a_\bullet = b_\bullet = z_{\bullet\bullet} + n_{\bullet\bullet}$$

where $n_{\bullet\bullet}$ counts the number of passengers, and $z_{\bullet\bullet}$ counts the number of transfers. $z_{\bullet\bullet}/n_{\bullet\bullet}$ is the average number of transfers per passenger.

As explained in section 2.1, transfers can only occur at junctions, that is $z_{ij} > 0$ implies $(i, j) \in T$. In particular, $z_{ii} = 0$: no traveler is supposed to disembark and re-embark later at the same stop.

2.4 Statement of the problem and solution method

Automatic passenger counters measure the number of passengers entering and leaving lines at each stop [Boyle, 1998], that is **a** and **b**, which provide the basic raw data of the present study, kindly provided by the Lausanne Transportation Agency (tl) for the case study in section 3.3. We will suppose here that this data obeys the necessary consistency conditions for embarkment and disembarkment counts, even if, in real case studies, a rescaling must usually be performed to balance in and out-flows on each lines (given in the appendix).

Intra-line edge flows $\mathbf{Y} = (y_{ij})$ can be determined by (2), but transfer edge flows $\mathbf{Z} = (z_{ij})$ are, here and typically, unknown. The objective is to estimate the $l \times l$ transportation flow matrix $\mathbf{N} = (n_{st})$. Many consistent solutions coexist in general, even for a single line with no transferts (section 2.5). This issue of incompletely observed data can be tackled by the maximum entropy formalism [8], which has often been invoked in transportation modeling research [9] [10].

Let $f_{st} = n_{st}/n_{\bullet\bullet}$ be the *distribution of st-trips* (empirical distribution) and let g_{st} be some prior guess on its shape (theoretical distribution). Assuming some reasonable initial prior g_{st},

(1) we first suppose that the empirical margins $\alpha_s = f_{s\bullet}$ and $\beta_t = f_{\bullet t}$ are known. Then f_{st} can be determined as the maximum entropy solution (section 2.4.1), i.e. as the distribution closest to g_{st} in the Kullback-Leibler divergence sense under the margin constraints, to be calibrated by an iterative fitting inner loop

(2) then (section 2.4.4), the prior is updated to $\tilde{g}_{st}$ by shrinking, if necessary, the priors g_{st}, thus avoiding transfer overflow exceeding the embarking and disembarking counts at each stop. Moreover, an hyperparameter $\theta \in [0, 1]$ is used at this stage in order to control the minimum proportion of passengers entering/leaving the network at each stop.

(3) finally (section 2.4.5), the margins are updated to $\tilde{\alpha}_s$ and $\tilde{\beta}_t$.

With the new prior distribution $\widetilde{g}_{st}$ and the new margin distributions $\widetilde{\alpha}_s$, $\widetilde{\beta}_t$, we can iterate the above steps, until convergence. The only free parameter is θ, whose effect is studied on toy examples in section 3.2.

The above iterative solution method is somewhat reminiscent of the EM algorithm. As a matter of fact, the first step (maximum entropy) exactly correspond to the "expectation step" of the EM algorithm (see e.g. [6] [7]), but steps two and three, aiming at calibrating parameters g_{st}, α_s and β_t, do not follow the maximum likelihood rationale of the "maximisation step". Pseudocode of the algorithm is shown in Algorithm 1.

2.4.1 Maximum entropy estimate of st-trips

As announced, the proportion of st-trips $f_{st} = n_{st}/n_{\bullet\bullet}$ (empirical distribution) will be estimated from some prior guess g_{st} (theoretical distribution) and margin constraints α_s and β_t for f_{st} by maximum entropy, i.e. by solving the problem

$$\begin{aligned} &\min_{\mathbf{f}\in\mathscr{F}} \sum_{st} f_{st} \log \frac{f_{st}}{g_{st}}, \\ &s.t. \sum_{t} f_{st} = \alpha_s, \\ &\quad \sum_{s} f_{st} = \beta_t. \end{aligned} \tag{6}$$

The Lagragian is

$$L = \sum_{st} f_{st} \log \frac{f_{st}}{g_{st}} - \sum_{s} \lambda_s (\alpha_s - \sum_{t} f_{st}) - \sum_{t} \mu_t (\beta_t - \sum_{s} f_{st}),$$

which gives, after deriving and setting to zero,

$$f_{st} = \phi_s \psi_t g_{st} \qquad \text{with } \phi_s := \exp(-1-\lambda_s),\ \psi_t := \exp(-\mu_t). \tag{7}$$

Using constraints in (6), we find

$$\phi_s = \frac{\alpha_s}{\sum_t \psi_t g_{st}}, \qquad \psi_t = \frac{\beta_t}{\sum_s \phi_s g_{st}}, \tag{8}$$

which yields the following *iterative fitting algorithm*: starting with some $\psi_t^{(0)} > 0$, one performs the iteration

$$\phi_s^{(\iota)} = \frac{\alpha_s}{\sum_t \psi_t^{(\iota)} g_{st}}, \qquad \psi_t^{(\iota+1)} = \frac{\beta_t}{\sum_s \phi_s^{(\iota)} g_{st}}, \tag{9}$$

until convergence to ϕ_s and ψ_t obeying (8).[3]

In view of (4) and (5), the postulated margins must satisfy, for each isolated stop i

$$\alpha_i = \frac{a_i}{n_{\bullet\bullet}} \qquad \beta_i = \frac{b_i}{n_{\bullet\bullet}} \tag{10}$$

permitting to determine the total flow as $n_{\bullet\bullet} = \frac{a_i}{\alpha_i}$, or $n_{\bullet\bullet} = \frac{b_i}{\beta_i}$ for any isolated stop i, and thus the st-flow itself as

$$n_{st} = n_{\bullet\bullet} f_{st} = n_{\bullet\bullet} \phi_s \psi_t g_{st} \tag{11}$$

whose plugging into (3) yields the intra-line edge flows $\mathbf{Y} = (y_{ij})$ and the transfer edge flows $\mathbf{Z} = (z_{ij})$. Only the latter is required for subsequent algorithm steps and can be computed with

$$z_{ij} = I((i,j) \in T) \sum_{st} \chi_{ij}^{st} n_{st} \tag{12}$$

[3] with possibly a small quantity $\varepsilon > 0$ added on null components of g_{st}.

where $I(.)$ denotes the 0/1 indicator function.

2.4.2 Initialization of the prior and the margins

The geometry of the network permits to define the set of permitted st-trips across the network denoted by P. The initial prior was chosen as the uniform distribution on admissible paths, that is as

$$g_{st} = \begin{cases} \frac{1}{|P|} & [s,t] \in \mathrm{P}, \\ 0 & \text{otherwise.} \end{cases}$$

The initial margins were chosen to initially match g_{st}, namely $\alpha_s = g_{s\bullet}$ and $\beta_t = g_{\bullet t}$ for all stops. However, g_{st} could be chosen more carefully in case of additional data. As a matter of fact, g_{st} represent *prior attractions* between nodes in equation (7), before margin correction given by ϕ_s and ψ_t and subsequent algorithm steps. An urban planner could choose to increase or decrease some of these values in order to take into account prior knowledge on commuting habits of inhabitants.

2.4.3 Embarkment, disembarkment constraints and hyperparameter θ

After a single iterative fitting step, the resulting transportation flow $\mathbf{N} = (n_{st})$ and transfer flow $\mathbf{Z} = (z_{ij})$ have little chance to fulfill constraints (4) and (5) defined by $\mathbf{a}$ and $\mathbf{b}$, and prior distributions g_{st}, α_s and β_t must be corrected accordingly.

When $z_{\bullet i} > a_i$, or $z_{i\bullet} > b_i$, the found solution typically predicts that there are more passengers entering, respectively exiting, at a stop i than the actual measured quantity. One could be tempted to correct prior distributions in order to have $z_{\bullet i} = a_i$, or $z_{i\bullet} = b_i$, on every problematic nodes i, but the latter solution would mean that all passengers entering (resp. exiting) the line at this stop are transiting, which seems unrealistic in most real life scenarios.

We define here the hyperparameter $\theta \in [0,1]$ as the *minimum proportion of passengers (among a_i and b_i) entering/leaving the **network** at each stop* (in other words, not transferring), that is

$$n_{s\bullet} \geq \theta a_s \qquad n_{\bullet t} \geq \theta b_t \tag{13}$$

or equivalently

$$z_{\bullet s} \leq (1-\theta) a_s \qquad z_{t\bullet} \leq (1-\theta) b_t \tag{14}$$

and updates of distributions will be made accordingly.

Note that, if additional data were known, we could set a particular value of θ for every node, and differing for embarkments and disembarkments. However, without additional information, we will restrain to this simpler case.

2.4.4 Updating the prior distribution

Overflow occurs in transfer edge (i,j) if $z_{i\bullet} > (1-\theta) b_i$ or $z_{\bullet j} > (1-\theta) a_j$. To avoid it, components g_{st} of the prior distribution will be shrinked by a suitable ratio whenever edge

flows $(i,j) \in [s,t]$ exhibit overflow. For any edge (i,j), let us compute the *flow ratio* r_{ij} as

$$r_{ij} = \max\left(1, \frac{z_{i\bullet}}{(1-\theta)b_i}, \frac{z_{\bullet j}}{(1-\theta)a_j}\right) \geq 1 \quad , \tag{15}$$

where $r_{ij} > 1$ denotes an overflow through edge (i,j). For a given origin-destination $[s,t]$, define the *orgin-destination flow ratio* $\bar{r}_{st}$ as the largest r_{ij} among edge flows $(i,j) \in [s,t]$, that is as

$$\bar{r}_{st} = \max_{ij} \chi_{ij}^{st} r_{ij} \geq 1 \quad . \tag{16}$$

By construction, $\bar{r}_{st} > 1$ denotes an overflow on some transfer edge between s and t. To adjust the flow, we shall divide the previous flow by this ratio

$$\widetilde{n}_{st} = \frac{n_{st}}{\bar{r}_{st}} \tag{17}$$

and define the new prior distribution as

$$\widetilde{g}_{st} = \frac{\left(\frac{\widetilde{n}_{st}}{\phi_s \psi_t}\right)}{\sum_{s',t'}\left(\frac{\widetilde{n}_{s',t'}}{\phi_{s'}\psi_{t'}}\right)} \quad . \tag{18}$$

where ϕ_s and ψ_t are the values (8) obtained in the previous maximum entropy step.

2.4.5 Updating the margin distributions

By construction, the corrected flow $\widetilde{n}_{st}$ found in (17) now respects embarkment and disembarkment constraints. We can compute the new transfer flow on edges with

$$\widetilde{z}_{ij} = I((i,j) \in T) \sum_{st} \chi_{ij}^{st} \widetilde{n}_{st} \tag{19}$$

and, with (4) and (5), updating margin distributions is straightforward

$$\widetilde{\alpha}_s = \frac{a_s - \widetilde{z}_{\bullet s}}{\sum_{s'}(a_{s'} - z_{\bullet s'})} \qquad \widetilde{\beta}_t = \frac{b_t - \widetilde{z}_{t\bullet}}{\sum_{t'}(b_{t'} - z_{t'\bullet})} \quad . \tag{20}$$

2.5 Markov property for a single line

A "network" made of a single line contains no transfers, and flow estimates can be obtained at once by the maximum entropy step only.

Let $i = 1, \ldots, l$ enumerate the stops in increasing order, that is $F(i) = i+1$. The initial prior is simply $g_{st} = c\, I(s < t)$ and captures solely the unidirectional nature of trips, where $c = \frac{1}{(l-1)(l-2)}$. The margins of the empirical distribution f_{st}, as well as the total flow, are here known

$$\alpha_s = \frac{a_s}{a_\bullet} \qquad \beta_t = \frac{b_t}{b_\bullet} \qquad n_{\bullet\bullet} = a_\bullet = b_\bullet \quad . \tag{21}$$

Algorithm 1 Computes the transportation flow matrix $\mathbf{N} = (n_{st})$, knowing the edge-trip incidence matrix $\boldsymbol{\chi} = (\chi_{ij}^{st})$, the set of transfer edges T, the set of permitted trips P, the embarking flow $\mathbf{a}$, the disambarking flow $\mathbf{b}$, the index of an isolated source node $\tilde{s}$, and the minimum proportion of passengers entering/leaving the network θ.

$g_{st} \leftarrow I([s,t] \in P)/|P|,\ \forall s,t$ ▷ Initialize the prior distribution
$\alpha_s \leftarrow g_{s\bullet},\ \forall s$ ▷ Initialize the network ingoing distribution
$\beta_t \leftarrow g_{\bullet t},\ \forall t$ ▷ Initialize the network outgoing distribution
$\varepsilon \leftarrow 10^{-40}$ ▷ Fix a small quantity
while $\mathbf{N} = (n_{st})$ has not converge **do** ▷ Main loop
  $\psi_t \leftarrow 1,\ \forall t$
  while $\boldsymbol{\psi} = (\psi_t)$ has not converge **do** ▷ Iterative fitting loop
    $\phi_s \leftarrow \alpha_s/(\sum_t \psi_t g_{st} + \varepsilon),\ \forall s$
    $\psi_t \leftarrow \beta_t/(\sum_s \phi_s g_{st} + \varepsilon),\ \forall t$
  end while
  $n_{st} \leftarrow \frac{a_{\tilde{s}}}{\alpha_{\tilde{s}}} \phi_s \psi_t g_{st},\ \forall s,t$ ▷ Compute the transportation flow
  $z_{ij} \leftarrow I((i,j) \in T) \sum_{st} \chi_{ij}^{st} n_{st},\ \forall i,j$
  $r_{ij} \leftarrow \max\left(1, \frac{z_{i\bullet}}{(1-\theta)b_i}, \frac{z_{\bullet j}}{(1-\theta)a_j}\right),\ \forall i,j$
  $\bar{r}_{st} \leftarrow \max_{ij} \chi_{ij}^{st} r_{ij},\ \forall s,t$
  $\widetilde{n}_{st} \leftarrow \frac{n_{st}}{\bar{r}_{st}},\ \forall s,t$
  $\widetilde{g}_{st} \leftarrow \frac{\left(\frac{\widetilde{n}_{st}}{\phi_s \psi_t + \varepsilon}\right)}{\sum_{s't'}\left(\frac{\widetilde{n}_{s't'}}{\phi_{s'} \psi_{t'} + \varepsilon}\right) + \varepsilon},\ \forall s,t$ ▷ Update the prior distribution
  $\widetilde{z}_{ij} \leftarrow I((i,j) \in T) \sum_{st} \chi_{ij}^{st} \widetilde{n}_{st},\ \forall i,j$
  $\alpha_s \leftarrow \frac{a_s - \widetilde{z}_{\bullet s}}{\sum_{s'}(a_{s'} - \widetilde{z}_{\bullet s'})},\ \forall s$ ▷ Update the network ingoing distribution
  $\beta_t \leftarrow \frac{b_t - \widetilde{z}_{t\bullet}}{\sum_{t'}(b_{t'} - \widetilde{z}_{t'\bullet})},\ \forall t$ ▷ Update the network outgoing distribution
end while
return $\mathbf{N} = (n_{st})$

Following (7) maximum entropy flows are of the form

$$n_{st} = n_{\bullet\bullet}\, c\, I(s < t)\, \phi_s\, \psi_t \tag{22}$$

where (setting $\Psi_s := \sum_{t>s} \psi_t$ and $\Phi_t := \sum_{s<t} c\phi_s$) the constraints (8) equivalently read

$$\phi_s = \frac{\alpha_s}{c \sum_{t>s} \psi_t} = \frac{a_s}{n_{\bullet\bullet}\, c\, \Psi_s} \qquad \psi_t = \frac{\beta_t}{c \sum_{s<t} \phi_s} = \frac{b_t}{n_{\bullet\bullet}\, c\, \Phi_t} \tag{23}$$

to be solved by iterative fitting.

Interestingly enough, the form (22) for the flows is reminiscent of the *gravity flows* of quantitative Geography [9] [10] [11] [12], where ϕ_s is the *push factor*, ψ_t is the *pull factor*, and $I(s<t)$ the *distance deterrence function*. Yet, instead of being symmetric in s,t and decreasing with the distance $|s-t|$, the distance deterrence function is here asymmetric due to the line orientation, but otherwise constant.

This constancy entails the following Markovian behaviour for flows: let m_{st} be the number of travelers embarking at stop s and still inside the line at stop $t > s$, and let ρ_{st} the probability that travelers embarking at s will disembark at t. By (22),

$$m_{st} = \sum_{u \geq t} n_{su} = n_{\bullet\bullet}\, c\, \phi_s \sum_{u \geq t} I(s < u)\, \psi_u = n_{\bullet\bullet}\, c\, \phi_s (\psi_t + \Psi_t)$$

The empirical estimate of ρ_{st} is given by the proportion, among the travelers embarking at s and still present at $t > s$, of travelers disembarking at t, that is

$$\rho_{st} = \frac{n_{st}}{m_{st}} = \frac{n_{\bullet\bullet}\, c\, \phi_s\, \psi_t}{n_{\bullet\bullet}\, c\, \phi_s (\psi_t + \Psi_t)} = \frac{\psi_t}{\psi_t + \Psi_t} \leq 1$$

which depends on t only: it appears that the disembarkment probability $\rho_t = \frac{\psi_t}{\psi_t + \Psi_t}$ at t is *independent* of the embarkment stop s. Said otherwise, a traveler embarking at any stop s (and thus necessarily in the line at $F(s) = s + 1$) experiences the *same disembarkment probability* at each further stop $t > s$.

This Markov property, enjoyed by maximum-entropic flows, contrasts other possible solutions, such as the "first in, first out" (FIFO) flows (homogenizing the traveled distances among users) or the "last in, first out" (LIFO) flows (tending to generate maximally contrasted traveled distances).

3 Case Studies

Case studies are divided in two sections. In the first section, we test the algorithm on toy examples, which are artificial networks where the transportation flow $\mathbf{N}_{\text{ref}} = (n^{\text{ref}}_{st})$ is randomly drawn. These examples enable some kind of validation of the algorithm, as the "real" transportation flow is known and can be compared to the solution $\mathbf{N} = (n_{st})$ given by our method. This setup differs from the second section, which is dedicated to applying the algorithm to the real case of the public transportation network of the city of Lausanne (tl), where embankment and disembarkment flows are measured but the real transportation flow is unknown. This second case study shows that the algorithm is applicable on large, real datasets and can give insights about passengers probable routes in the network.

3.1 Error measurements

In all case studies, we obtain a estimation of the transportation flow with the algorithm, noted $\mathbf{N} = (n_{st})$, starting from the real embarkment flow $\mathbf{a}_{\text{ref}}$ and disembarkment flow $\mathbf{b}_{\text{ref}}$. In toy examples, we also have access to the real transportation flow $\mathbf{N}_{\text{ref}}$. There are two types of dissimilarity measures between the data and the solution proposed by the algorithm: (1) if we have access to $\mathbf{N}_{\text{ref}}$, how much $\mathbf{N}$ differs from it, and (2) how well constraints defined by $\mathbf{a}_{\text{ref}}$ and $\mathbf{b}_{\text{ref}}$ are respected. The first dissimilarity is measured through the *mean transportation error*, denoted by $\text{MTE}(\mathbf{N})$, and computed as

$$\text{MTE}(\mathbf{N}) = \sum_{st} \frac{n^{\text{ref}}_{st}}{n^{\text{ref}}_{\bullet\bullet}} \frac{|n_{st} - n^{\text{ref}}_{st}|}{n^{\text{ref}}_{st}} = \frac{\sum_{st} |n_{st} - n^{\text{ref}}_{st}|}{n^{\text{ref}}_{\bullet\bullet}} \tag{24}$$

and the second one with the *mean margin error*, noted $\text{MME}(\mathbf{N})$, defined as

$$\begin{aligned} \text{MME}(\mathbf{N}) &= \frac{1}{2} \sum_i \frac{a^{\text{ref}}_i}{a^{\text{ref}}_\bullet} \frac{|z_{\bullet i} + n_{i\bullet} - a^{\text{ref}}_i|}{a^{\text{ref}}_i} + \frac{1}{2} \sum_i \frac{b^{\text{ref}}_i}{b^{\text{ref}}_\bullet} \frac{|z_{i\bullet} + n_{\bullet i} - b^{\text{ref}}_i|}{a^{\text{ref}}_i} \\ &= \frac{\sum_i (|z_{\bullet i} + n_{i\bullet} - a^{\text{ref}}_i| + |z_{i\bullet} + n_{\bullet i} - b^{\text{ref}}_i|)}{2 n^{\text{ref}}_{\bullet\bullet}} \end{aligned} \tag{25}$$

where $z_{ij} = I((i,j) \in T) \sum_{st} \chi^{st}_{ij} n_{st}$ is the flow on transfer edges T. Both errors can be interpreted as a weighted mean of error percentages.

Note that, by construction, the MME should be zero when the algorithm converges. However, it can be informative to track down this error along iterations and, in some practical cases where margin constraints are impossible to fulfill, the algorithm convergence criterion is reached with MME > 0.

3.2 Toy Examples

3.2.1 Construction

All constructed toy examples are built following the same approach, which aims at being simple but somewhat realistic. We fix a number of *round trips* $p \geq 2$, each of which is constituted of a forward line and a backward line, for a total of $q = 2p$ lines. Every line has a starting and an ending node, which are isolated nodes, and possesses $p-1$ intermediary nodes which allow transfers to the other round trips, yielding a total of $n = 2p(p+1)$ nodes in the network. Examples of these toy networks can be found in Figure 1.

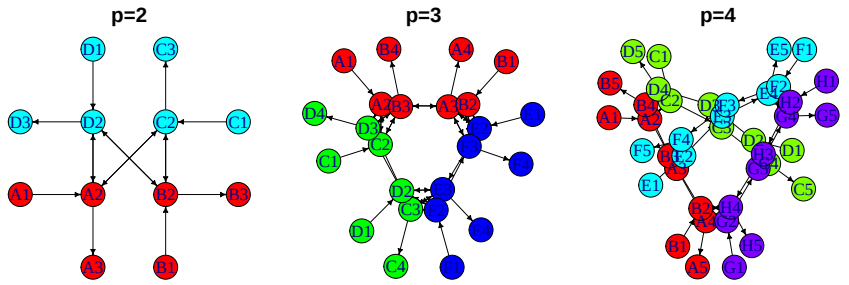

Figure 1: 3 toy examples, with the number of round trips $p \in \{2,3,4\}$. Trips are displayed in the same color, lines possess a unique letter, and each stop correspond to a unique combination of a letter and a number. Clusters of nodes represent positions where transfers between round trips are possible.

In order to be realistic, the permitted *st*-trips set P is constructed by considering all shortest-path between pair of nodes, excluding :

- s and t that are on the same line but with t preceding (or equal to) s in the line order.
- s and t that are on the same round trip but on opposite lines.
- s and t whose shortest-path starts with a transfer edge, ends with a transfer edge, or possesses two (or more) consecutive transfer edges.

A transportation flow $\mathbf{N}_{\text{ref}} = (n^{\text{ref}}_{st})$ is drawn by setting a fixed number of passengers $n^{\text{ref}}_{\bullet\bullet}$, and each passenger is assigned randomly to a (s,t) pair drawn uniformly among P. From this reference transportation flow $\mathbf{N}_{\text{ref}}$, using the edge-trip incidence matrix $\boldsymbol{\chi}$ and equation (3), we can compute flow on edges $\mathbf{X}_{\text{ref}}$ and, in turn, the number of passengers embarking $\mathbf{a}_{\text{ref}}$ and the number of passengers disembarking $\mathbf{b}_{\text{ref}}$ at each stop.

3.2.2 Algorithm iterations

First, we exhibit some algorithm iterations on a toy example with $p = 2$, where 50 passengers were drawn uniformly across the $|P| = 20$ possible *st*-trips. Some iterations of the algorithm with $\theta = 0.001$, along with MTE and MME errors, are shown in Figure 2. On this small example, we can see that the algorithm quickly find an estimation giving a small MME error, but still give a MTE of 0.256. This result is due to the fact that only 50 passengers are drawn, giving a large deviation compared to the optimally found solution which maximizes the entropy.

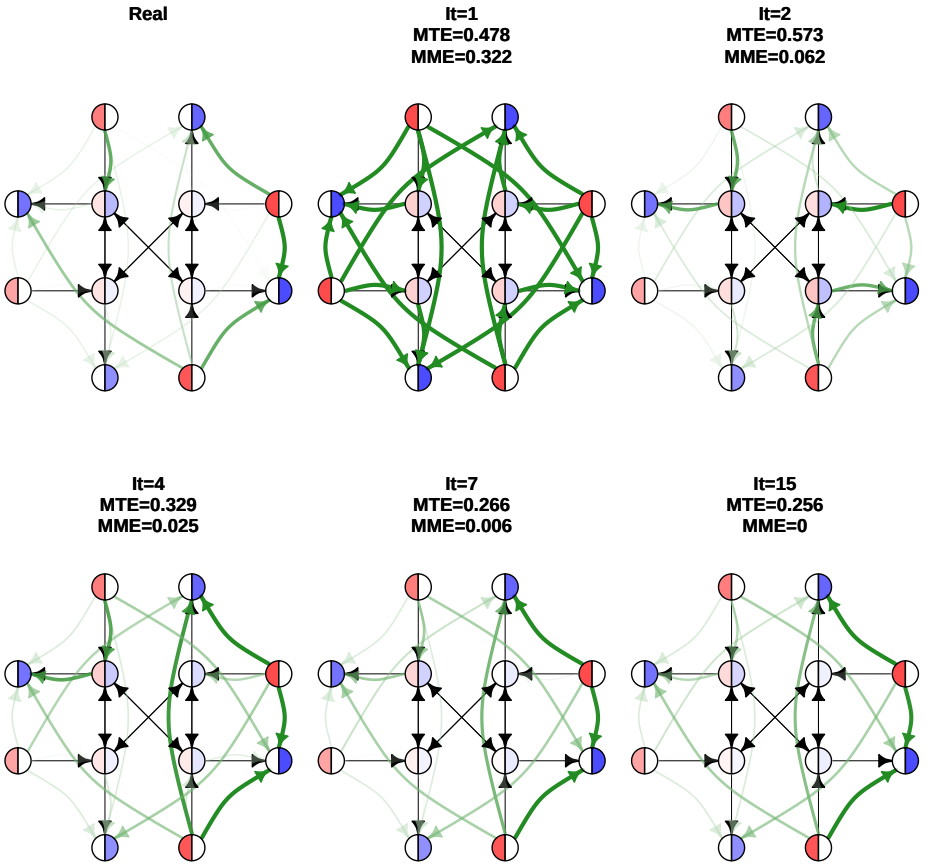

Figure 2: Real transportation flow (green arrows) obtained by randomly drawing 50 passengers on the graph toy example with $p = 2$ round trips, along with iterations $1, 2, 4, 7, 15$ of the algorithm with $\theta = 0.001$. MTE and MME errors are computed, and embarkment and disembarkment counts are represented respectively by the red and blue colors on nodes.

3.2.3 MTE study

The main goal of toy examples, since we have access to the real transportation flow, is to study how the resulting MTE behaves regarding passenger *st*-trips distribution and hyperparameter θ, on different network sizes.

The randomness of the *st*-trips distribution is controlled by the number of drawn passengers and we can see that the algorithm performs better if this number increases, as seen in Figure 3. This behavior can be expected as the method is constructed to find the solution maximizing the entropy while respecting embarkment and disembarkment constraints. In fact, all MTE should converge to 0 eventually, however this rate of convergence seems to be low. As for the hyperparameter study, found in Figure 4, we see that the optimal parameter seems to decrease as the network size increases, reaching value close to 0 for 8 round trips. However, this is not the only factor which helps to calibrate this parameter. As a matter of fact, it seems unrealistic, in some real life scenarios, that all passengers embarking or

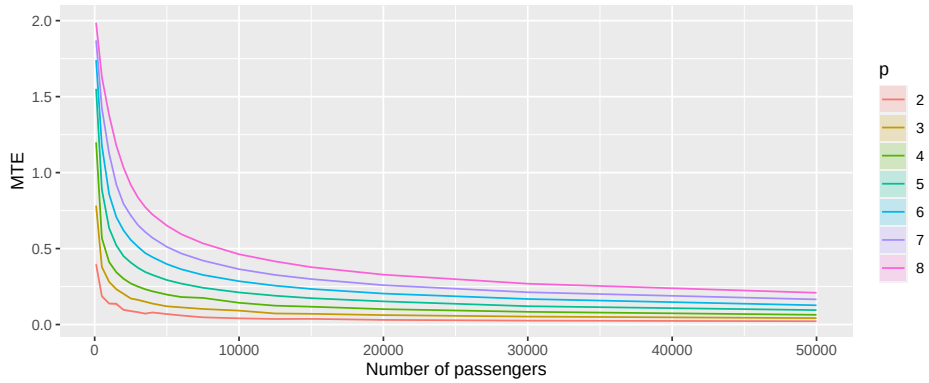

Figure 3: Mean transportation error (MTE) vs the number of randomly drawn passengers, in toy examples with numbers of round trips p varying from 2 to 8. The hyperparameter is set to $\theta = 0.001$ and 10 different draws are performed for each data point.

disembarking at junctions are all transiting, and this value will be kept at 0.1 for our real life study.

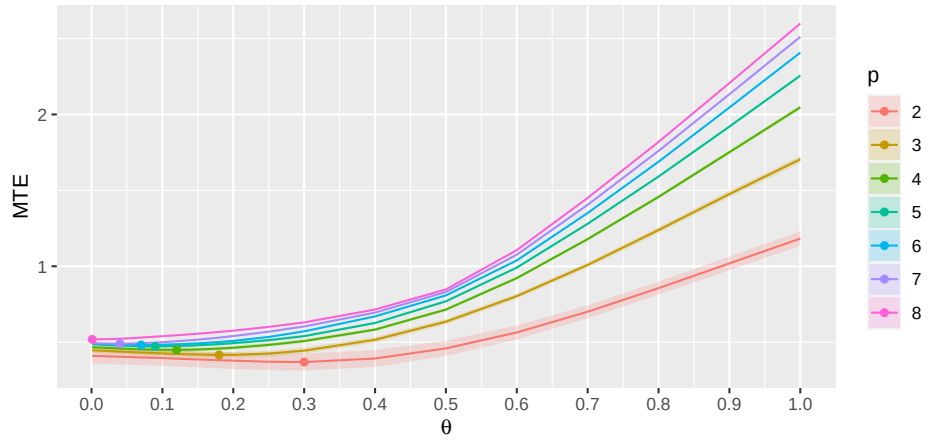

Figure 4: Mean transportation error (MTE) vs hyperparameter θ, in toy examples with numbers of round trips p varying from 2 to 8. The dot on each curve denotes the minimum. The number of drawn passengers is different for every network size and is set to $5 \cdot |P|$, in order to keep an (almost) constant variance. 10 different draws are performed for each data point.

3.3 Real Data

3.3.1 The dataset and algorithm setup

The dataset was constructed from data collected by the public transportation agency in the city of Lausanne (tl), in Switzerland. Automatic passenger counters were used to collect data on the number of passengers entering and leaving buses and subways at each stop. The dataset comprises 44 lines, 1361 stops, and over 115 million passengers per year. Each round trip forms two separate lines, which often have stops that correspond to both

the inbound and outbound directions, but this is not always the case. Some stops may be unique to either the inbound or outbound direction. The initial dataset contained 13 million data rows, which were aggregated for the year 2019, and the passenger data was filtered to include only passengers who embarked and disembarked at each stop. Lines with the most errors were removed, resulting in a final dataset of 35 transportation lines and 1216 stops. The threshold for invalid lines was set when a difference between total embarkment and disembarkment counts differs more than 15% of their means. For the rest all other lines, the correction found in the appendix was applied to ensure consistency conditions.

Line edges were built from the given data, and to obtain the full network structure, we added transfer edges between different lines if the estimated walking time (using Open Trip Planner[4] [13]) was less than 120 seconds. Shortest-paths between all pairs of stops were computed using the breath-first search algorithm [14] implemented in the *igraph*[5] R library [15], thus giving $\boldsymbol{\chi}$. The matrix P, containing permitted st-trips, was built using the same conditions found in the toy example experiments (section 3.2.1) with an additional one. If the estimated pedestrian time between two stops s and t was shorter than the estimated time by taking the public transportation network, this st-trip was also removed from P. The graph structure, consisting in the edge-trip incidence matrix $\boldsymbol{\chi}$, the permitted st-trips set P, and the transfer edges set T, was constructed beforehand testing the method.

The algorithm was run with hyperparameter $\theta = 0.1$. Tracking the mean margin error (MME) showed a rapid decrease (< 0.001 after 13 iterations) but the (conservative) convergence criterion (chosen as when $\sum_{st} |f_{st}^{\text{prev}} - f_{st}| < 10^{-6}$) was reach after 316 iterations, which corresponds to approximately one hour of computing time with a single thread on a AMD Epyc2 7402 with 32GB of RAM.

3.3.2 Results

As the result of our algorithm consists in the (1216×1216) matrix $\mathbf{N} = (n_{st})$, it is difficult to present them succinctly. We chose here to focus on two aspects: (1) mapping the origin and destination profiles of stops with the help of Correspondence Analysis (CA) [16]; and (2) mapping the aggregated transfers between lines.

The first and second factorial dimensions of the destination and origin profiles of stops, obtained with CA applied on $\mathbf{N}$, can be found on Figure 5. We can see that origin and destination coordinates of each stop are quite similar when displayed on the first two factorial axes. This result can be explained if we understand that, in fact, origin and destination profiles are, to some extent, symmetrical: at the beginning of a line, stops have several choices of destinations, but very few origins "point" at them, which is exactly the reverse for stops at the end of a line. Intermediary stops are generally located near the city center, and have a similarly rich origin and destination profiles. Concerning the dimensions, we can see that the first component divides the city stops between west and east, a known dichotomy in the public transportation usage in Lausanne: the east of the city is less well served by public transport and inhabitants often use cars, while the west intensively uses public transports. The second component is harder to interpret, but seems to highlight the radial structure of the transportation network in Lausanne.

Figure 6 maps predicted transfer hubs, obtained by aggregating every transfer counts z_{ij} between lines when they occurs between stops located in the same spatial cluster

[4] https://docs.opentripplanner.org/en/v2.2.0/
[5] https://igraph.org/

(same "superstop" name in the dataset). As expected, most transfers occur in the city center, with the top 5 largest hubs located at St-François (2,132,597 transfers), Lausanne-Gare (2,035,543 transfers), Bel-Air (1,908,168 transfers), Ours (1,057,282 transfers), and Chauderon (969,789 transfers). When taken individually, four of the top 5 transfer counts are predicted between line 1 and line 72 occurring at the train station (Lausanne-Gare). Line 72 is actually a subway line connecting the northern and southern regions of the city, and is by far the most used line in the network. Line 1 is also a highly frequented line and operated toward the western region of the city. It is not surprising to see most transfers occurring between these two lines, as they permit to connect the region with a high density of public transportation network users (the western part of Lausanne), to the main artery of the network (line 72).

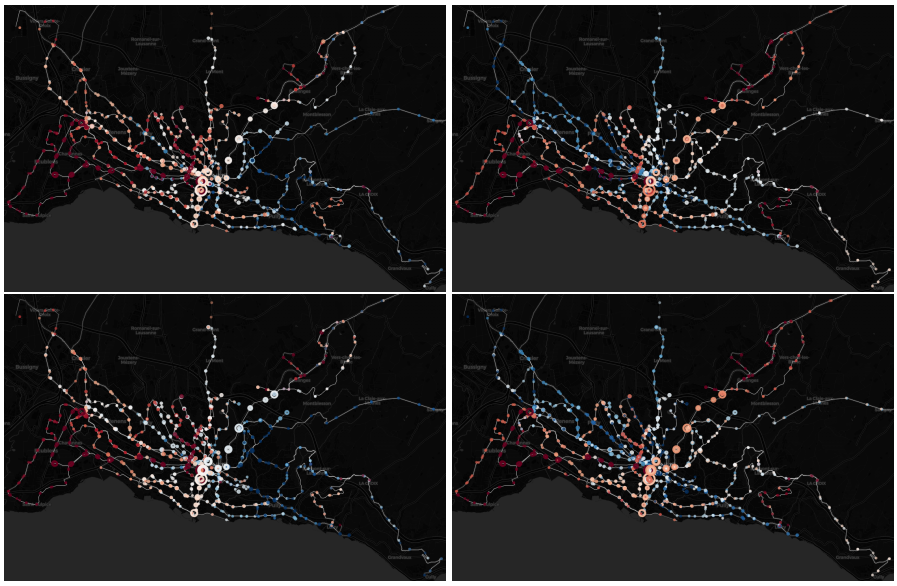

Figure 5: Color representations (positive values in red, negative values in blue) of the first two factorial coordinates of stops, obtained from Correspondence analysis (CA) performed on the transportation matrix **N**. Top row maps stops as "origin profiles" ("where passengers are going"), bottom row maps stops as "destination profile" ("where passengers are coming from"). Left column depicts the first factorial dimension, right column the second factorial dimension. Ring sizes correspond to the number of embarkments (top) and disembarkments (bottom).

4 Conclusion

At first glance, the task of estimating the origin-destination trip counts of public transportation networks on the sole basis of embarkments and disembarkments stop counts might appear as too ambitious. The project is indeed challenging, but we hope to have convinced the reader that meaningful estimates can be obtained by applying a succession of carefully chosen, principled yet flexible steps. Some of the ingredients (maximum entropy, iterative fitting) are familiar in transportation theory, and some others (shrinkage of priors, minimal amount of entering and leaving the network) seem original. Possible theoretical and

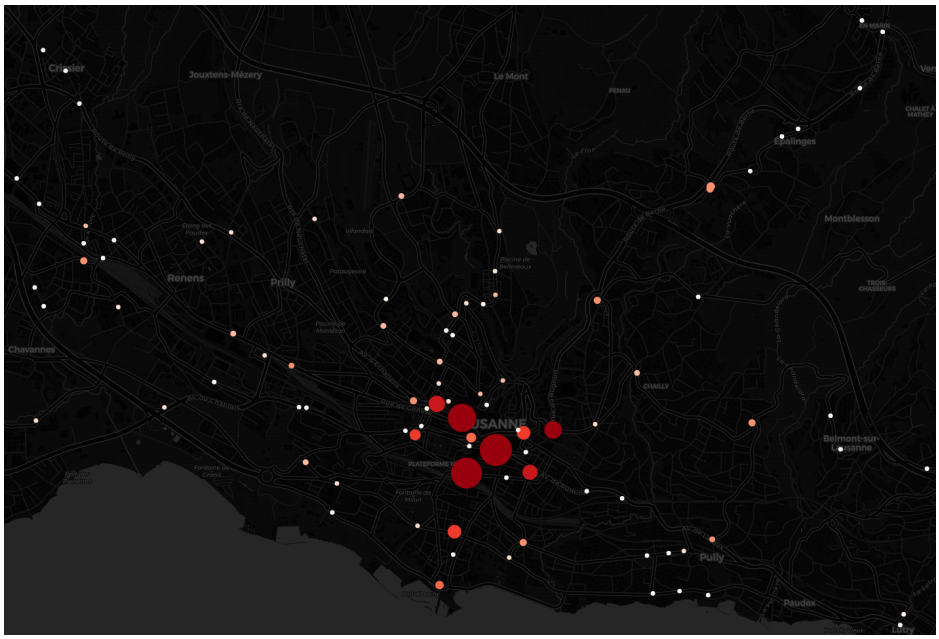

Figure 6: Localization and size of transfer hubs, obtained by aggregating estimated transfer flows between lines z_{ij} in "superstops".

algorithmic improvements are still under investigation. Due to length constraints, the exploitation and interpretation of the estimated flow has been kept to a minimum. Yet, many issues, such as disaggregating data and regimes (week days, week-ends or holidays; time in the day), assessing the centrality of stops and junctions, or characterizing the imbalance between uphill and downhill flows (Lausanne is known for its slopes) are of primary interest for urban planning and transportation geography, and will be developed in a forthcoming work. Also, visualizing the numerous and various quantities of interest, on a spatial map or else, constitutes a challenge in itself, requiring a particular blend of creativity and rigor.

Appendix

Correction of the embarkment and disembarkment counts in a single line

It may happen that, in a line ℓ with stops indexed in order as $1,\dots,l$, raw data $\mathbf{a} = (a_i)$ and $\mathbf{b} = (b_i)$ do not obey the following necessary consistency conditions

$$\begin{aligned} a_l = 0,\ b_1 = 0 & \\ A_{(i-1)} \geq B_i & \qquad \forall i \in \{1,\dots,l-1\} \\ A_l = B_l & \qquad \text{for terminal stop } l \end{aligned}$$

where A_i (respectively B_i) is the cumulated number of embarked (resp. disembarked) passengers on the line at stop i, i.e. $A_i = \sum_{j=1}^{i} a_j$ and $B_i = \sum_{j=1}^{i} b_j$. The first condition is easy to correct (by setting both quantities to 0), and we will assume that they are valid. The last two require an iterative procedure, explained here.

We first identify all stops i where $A_{i-1} < B_i$ and store them, along with stop 1 and l, in a order set S. Let $\text{Prev}(i)$ be the item right before i in the set S. For all $i \in S \setminus \{1\}$, we do :

$$\hat{a}_j = \left(1 - \frac{(A_{(i-1)} - A_{(\text{Prev}(i)-1)}) - (B_i - B_{\text{Prev}(i)})}{(A_{(i-1)} - A_{(\text{Prev}(i)-1)}) + (B_i - B_{\text{Prev}(i)})}\right) a_j \qquad \forall j \in \{\text{Prev}(i), \dots, i-1\}$$

$$\hat{b}_j = \left(1 + \frac{(A_{(i-1)} - A_{(\text{Prev}(i)-1)}) - (B_i - B_{\text{Prev}(i)})}{(A_{(i-1)} - A_{(\text{Prev}(i)-1)}) + (B_i - B_{\text{Prev}(i)})}\right) b_j \qquad \forall j \in \{\text{Prev}(i)+1, \dots, i\}$$

Before the last step, all conditions $A_{(i-1)} \geq B_i$ should be respected except for node l. The last step ensures that $A_l = B_l$. However, as this last step can sometimes lower the embarkment count and increase the disembarkment count on previous nodes (if $A_l > B_l$ before correction), some new nodes can now violate $A_{(i-1)} \geq B_i$. This is why this procedure must be iterated until all stops on the line verify consistency conditions.

Acknowledgements
This paper would not have been possible without the dataset given by the transportation agency of the city of Lausanne (tl). We wish to thank the agency for its kindness and hope this research will help them for future developments.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Department of Language and Information Sciences, University of Lausanne, Lausanne, Switzerland. [2]Institute of Geography and Sustainability, University of Lausanne, Lausanne, Switzerland.

References

1. Peyré, G., Cuturi, M., *et al.*: Computational optimal transport: With applications to data science. Foundations and Trends® in Machine Learning **11**(5-6), 355–607 (2019)
2. Bell, M.G., Lida, Y.: Transportation Network Analysis. Wiley, Chichester (1997)
3. Hazelton, M.L.: Estimation of origin–destination matrices from link flows on uncongested networks. Transportation Research Part B: Methodological **34**(7), 549–566 (2000)
4. Ashok, K., Ben-Akiva, M.E.: Estimation and prediction of time-dependent origin-destination flows with a stochastic mapping to path flows and link flows. Transportation science **36**(2), 184–198 (2002)
5. Cui, A.: Bus passenger origin-destination matrix estimation using automated data collection systems. PhD thesis, Massachusetts Institute of Technology (2006)
6. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the royal statistical society: series B (methodological) **39**(1), 1–22 (1977)
7. Bavaud, F.: Information theory, relative entropy and statistics. In: Sommaruga, G. (ed.) Formal Theories of Information: From Shannon to Semantic Information Theory and General Concepts of Information. LNCS, vol. 5363, pp. 54–78. Springer, Berlin (2009)
8. Jaynes, E.T.: Information theory and statistical mechanics. Physical review **106**(4), 620 (1957)
9. Wilson, A.: A statistical theory of spatial distribution models. Transportation Research **1**(3), 253–269 (1967)
10. Erlander, S., Stewart, N.F.: The Gravity Model in Transportation Analysis: Theory and Extensions vol. 3. VSP, Leiden (1990)
11. Bavaud, F.: The quasi-symmetric side of gravity modelling. Environment and Planning A **34**(1), 61–79 (2002)
12. Thomas-Agnan, C., LeSage, J.P.: In: Fischer, M.M., Nijkamp, P. (eds.) Spatial Econometric OD-Flow Models, pp. 2179–2199. Springer, Berlin, Heidelberg (2021)
13. Malcolm Morgan, Marcus Young, Robin Lovelace, Layik Hama: Opentripplanner for r. Journal of Open Source Software **4**(44), 1926 (2019). doi:10.21105/joss.01926
14. West, D.B., *et al.*: Introduction to Graph Theory vol. 2. Prentice hall Upper Saddle River, ??? (2001)
15. Csardi, G., Nepusz, T.: The igraph software package for complex network research. InterJournal **Complex Systems**, 1695 (2006)
16. Benzécri, J.-P.: Histoire et préhistoire de l'analyse des données. partie v l'analyse des correspondances. Cahiers de l'analyse des données **2**(1), 9–40 (1977)