MULTIMODAL SHORTEST HYPERPATHS CONSIDERING CROWDING IN TRANSIT VEHICLES

David López
Grupo de Investigación en Ingeniería de Transporte y Logística, Universidad Nacional Autónoma de México, Mexico
Angélica Lozano
Grupo de Investigación en Ingeniería de Transporte y Logística, Universidad Nacional Autónoma de México, Mexico

ABSTRACT

In rush hours, the onboard crowd level within transit vehicles is a problem of any large city. At the expense of time savings, some users will avoid riding crowded lines if they consider that transit vehicles do not have enough personal space. This paper presents a hypergraph model and an algorithm to find multimodal shortest hyperpaths considering the user constraints on the sequence of boarded modes and their preferences of onboard crowding levels. It is assumed that transit inter-arrivals are random. A penalty for riding transit vehicles is defined to model how the users perceive the onboard crowd levels. This penalty depends on the onboard crowd levels and the seating capacity of the vehicles. A state-automaton is used to model the user constraints on the sequence of boarded modes. To find the shortest hyperpaths, the user selects their origin, destination, the maximum number of modal transfers, and the onboard crowd level threshold. A use case on a sample multimodal network is presented.

1. INTRODUCTION

Saturated transit networks are a common issue in any large city. Whenever possible, some users will avoid large crowds or wait at stops for less crowded vehicles. Transit users not only want to reduce their travel time, other factors like cost, transfers, and occupancy also affect their route choice (Raveau et al., 2014). Haywood et al. (2017) pointed out that traveling in crowded vehicles affects users' sanity by decreasing their perception of security and increasing their anxiety and stress.

Expanding the fleet or transit routes may mitigate the effects of crowded vehicles, but these solutions are expensive or may take several years to carry out. An affordable solution to the onboard crowd problem is to display at the stops the crowd information of the incoming vehicles. This information leads to a shift in the passenger distribution among consecutive vehicles, thus reducing the onboard crowd levels (Zhang et al., 2017). Preston et al. (2017) found that it is unlikely that advanced information on onboard crowding will reduce crowd levels during peak hours; however, crowding information will reduce the users' stress and improves the passenger experience while traveling.

Since the onboard crowd information within transit vehicles is an important point to consider for some users, this paper presents a label-correcting algorithm to find multimodal shortest hyperpaths in a multimodal transport system where:

- The user restrict the sequence of boarded modes
- Selects the maximum onboard crowd levels they would like to face during their trip

To find the shortest paths and extension of the algorithm by Lozano & Storchi (2002) is developed. Since transit users experience the onboard crowd levels differently (Whelan & Crockett ,2009), a penalty is defined to model the user perception of the onboard crowd levels. It is considered that the transit modes have random-arrivals, i.e., the arrival-rate at stops is random and the waiting times can only be estimated based on particular distribution.

The subsequent sections are organized as follows. In Section 0, the related work is presented. Section 0 presents a brief introduction to hypergraph notation. Section 0 shows a hypergraph model of a multimodal transport system where: (1) the transit has random-arrivals, (2) the sequence of modes is restricted, and (3) the onboard crowd levels are considered. Section 0 presents the algorithm for finding the shortest hyperpaths bound by modal transfers and onboard crowd levels. An example of usage is shown in Section 0, and in Section 0, the conclusions are highlighted.

2. RELATED WORK

Assessing how users perceive the onboard crowding levels has been studied by conducting surveys. In 2009, Whelan & Crockett (2009) conducted a study in the UK trains to know how passengers perceive time while traveling in crowded vehicles. The authors conclude that standing passengers perceive longer travel times than seated passengers. The authors also found out that the perceived travel time of standing passenger is equal to the travel time multiplied by a scalar which depends on the onboard crowding. Let call this scalar the *riding penalty*. Qin (2014) uses the results of Whelan & Crockett (2009) and proposes three non-linear functions to estimate the *riding penalties*. Other authors like Batarce et al. (2016), Haywood et al. (2017), Tirachini et al. (2017), Bansal et al. (2019), and Márquez et al. (2019) also conducted surveys to estimate the *riding penalty*. As expected, few similarities are found in the mentioned studies as the *riding penalty* estimation depends on the network geometry, the infrastructure, and the idiosyncrasy of passengers. In the presented research, a non-linear function proposed by Qin (2014) is adapted to estimate the *riding penalty*.

Finding shortest paths in transit networks considering onboard crowd levels helps users find routes that accommodate to their personal preferences. Nuzzolo et al. (2015) and Comi & Nuzzolo (2016) developed tailored utility functions to model travel costs in scheduled transit systems. By using surveys and individual preference learning processes, the utility functions are defined and calibrated to model the user preferences on arrival times, onboard crowding levels, and total transfers. Rajapaksha et al. (2017) leverage the use of ubiquitous data sources to propose a framework that could be used to find the shortest paths with low onboard crowding levels. Katona et al. (2017) propose an ant-colony algorithm to find multimodal shortest paths where onboard crowd levels are considered. Unlike other works where the transit is schedule-based, the presented work aims to develop a routing algorithm and hypergraph model for a multimodal transport network where the transit has random-arrivals, such that the hyperpath algorithm considers:

- The user preferences on onboard crowd levels
- The user constraints on the sequence on board modes
- The user bounds of modal transfers.

In the presented paper, the crowd levels are modeled with a penalty that grows as more people get on the vehicles, and according to the reviewed research, this is how people perceive the crowd as the vehicle fills up. The penalty can be calibrated for a particular user or situation (commuting vs. leisure trip), but the calibration procedure is out of the paper's scope. Although the works mentioned in the last paragraph consider the onboard crowding levels for the path computation, the authors fail to present a model of how they estimate the crowding levels.

3. PRELIMINARIES

A directed hypergraph is a pair $H = (V, E)$, where V is the set of nodes and E is the set of arcs and hyperarcs. An arc $a \in E$ is defined as a pair of nodes $a = (i, j)$ where $\{i, j\} \subset V$. A hyperarc $e \in E$ is defined as the pair $e = \big(t(e), h(e)\big)$ where $t(e) \subset V$ is the set of *tail* nodes and $h(e) \subset V$ is the set of *head* nodes (Voloshin, 2009). A hyperarc $e = \big(i, h(e)\big)$ is a *support* hyperarc and a hyperarc $e' = \big(i, h'(e)\big) : h'(e) \subseteq h(e)$ is a *contained* hyperarc (Lozano & Storchi, 2002). Fig shows a directed hypergraph where $e_1' = (1,7)$ is a *contained* hyperarc of the *support* hyperarc $e_1 = (1, \{1,7\})$.

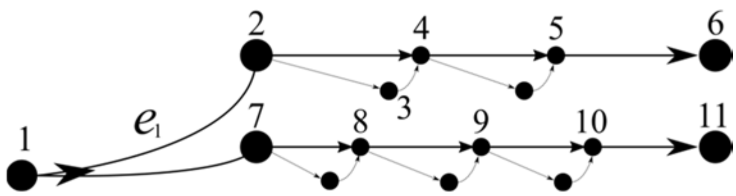

Figure 1: Representation of a directed hypergraph

Let $B(j)$ be a set of arcs $a \in E$ and *contained* hyperarcs $e' \in E : |h'(e)| = 1$ entering to the node j, i.e., $B(j) = \{a = (i,j) \in E$ and $e' = (i',j) \in E : \{i,i',j\} \subset V\}$ (López & Lozano, 2020). For example, in the hypergraph shown in figure 1 $B(4) = \{a = (2,4), e' = (3,4)\}$. A path r_{id}, that connects an origin o and a destination d is a sequence of nodes, arcs and hyperarcs. A hyperpath p_{id} is the minimum set of acyclic paths r_{id}, such that the destination d is connected to any node that belongs to r_{id} (Nguyen & Pallottino, 1989). A sub-hyperpath $p_{id}(l)$ from the node i to the node d associated to the line l is compose by a *contained* hyperarc $e' = (i,j)$ and a set of consecutive arcs $\{(j,j_1), (j_1,j_2) \ldots (j_{n-1},d)\}$, such that the *contained* hyperarc and all the arcs are associated to a unique transit line. In figure 1 the sub-hyperpath $p_{1,6}(l) = \{(1,2), (2,4), (4,5), (5,6)\}$.

4. HYPERGRAPH MODEL

Let H be a hypergraph that models a multimodal transportation system, such all transit modes have random-arrivals. The hypergraph $H = (V, E, M)$ where, V is the set of nodes, E is the set of arcs and hyperarcs and $M = \{M1, M2, M3, M4\}$ is the set of modes such that:

- $M1 \coloneqq$ Pedestrian mode
- $M2 \coloneqq$ Bicycle mode
- $M3 \coloneqq$ Non-rail transit modes
- $M4 \coloneqq$ Rail transit modes
- $M5 \coloneqq$ Modal transfers

The arcs $(i,j) \in E$ model the pedestrian, bicycle and modal transfers networks. The travel time of an arc $(i,j) \in E$ is defined as τ_{ij}. Two consecutive stops of the same transit line are modeled with one *contained* hyperarc $e' = (i,j)$ and two arcs (j,k) and (j,m), where e' represents boarding a transit vehicle at stop i, (j,k) represents riding the vehicle from stop i to stop k and descending at stop k and (j,m) represents riding the vehicle from stop i to stop k and not descending at the stop k.

Since the inter-arrival of transit vehicles is random, the waiting time at stops depends on the transit lines the user is willing to board to get to the destination. Let $e = \big(i, h(e)\big)$ be a *support* hyperarc representing the set of lines stopping at i. Each *contained* hyperarc of e is associated with one of the lines stopping at i. Let L_i be the set of transit lines stopping at i associated with the *support* hyperarc $e = \big(i, h(e)\big)$. Given an origin and a destination, it is defined the *attractive set* of lines, $L_i^{e^b} \subseteq L_i$, such that at stop i, the user is willing to board the first arriving line in $L_i^{e^b}$ to reach the destination. The *attractive set* defines the concept of *boarding* hyperarc $e^b = \Big(i, h(e^b)\Big) : h(e^b) \subseteq h(e)$ where each *contained* hyperarc of e^b is associated with one line in the *attractive set* (Lozano & Storchi, 2002).

Next are defined four concepts for the lines contained in the *attractive set* (Spiess & Florian, 1989):

- φ_{ij}^{l} is the arrival-rate of the line l associated with the *contained* hyperarc $e' = (i,j)$
- ϕ_{e^b} is the arrival-rate of the lines contained in the *attractive set*. It is defined as $\phi_{e^b} = \sum_{j\in h^b(e)} \varphi_{ij}^{l}$ such that, $l \in L_i^{e^b}$ and $e_l' = (i,j)$ is a *contained* hyperarc of the *boarding* hyperarc e^b
- $\omega_{e^b} = 1/\phi_{e^b}$ is the waiting time of the lines in the *attractive set*
- $\pi_{e^b,l} = \varphi_{ij}^{l}/\phi_{e^b}$ is the probability of boarding the line l in the *attractive set*

The user constraints on the sequence of transport modes are modeled with a state automaton and are described as follows. Lozano & Storchi (2002) defined the *viability* of a hyperpath for knowing which combinations of modes are admissible according to the constraints on the sequence of boarded modes. A hyperpath is *viable* if all the sub-hyperpaths satisfy the constraints on the sequence of boarded modes.

The combination of boarded modes in a hyperpath is indicated with the *state* of the hyperpath. In this paper, it is assumed that the bicycle and rail modes are constrained, i.e., if one of these modes is taken and then left, then the mode cannot be retaken in any other part of the trip. Since the hypergraph H has two constrained modes, eight combinations of modes are *viable*, and eight *states* can be associated with the hyperpaths.

The *deterministic finite state automaton* (DFA) shown in figure 2 defines the sequence of *viable states* and is used to know which hyperpaths are *viable*. The DFA is defined by the graph $A = (S, M, \gamma, 0, F)$ where: $S = \{0,1,2,3, \dots 8\}$ is the set of *states*, $M = \{M1, M2, M3, M4, M5\}$ is the set of modes, 0 is the initial *state*, and $F = \{1,4,6,8\}$ is the set of *states* for finishing trips. All trips end by foot so all *states* in F include the pedestrian mode (see figure 2). Finally, the function $\gamma(m, m', s) = s' : \gamma: M \times M \times S \rightarrow s$, models the arcs of the DFA, i.e., the tuple (s, s') is an arc in the DFA if it is possible to transfer from the mode m to the mode m' while maintaining the constraints on the sequence of boarded modes.

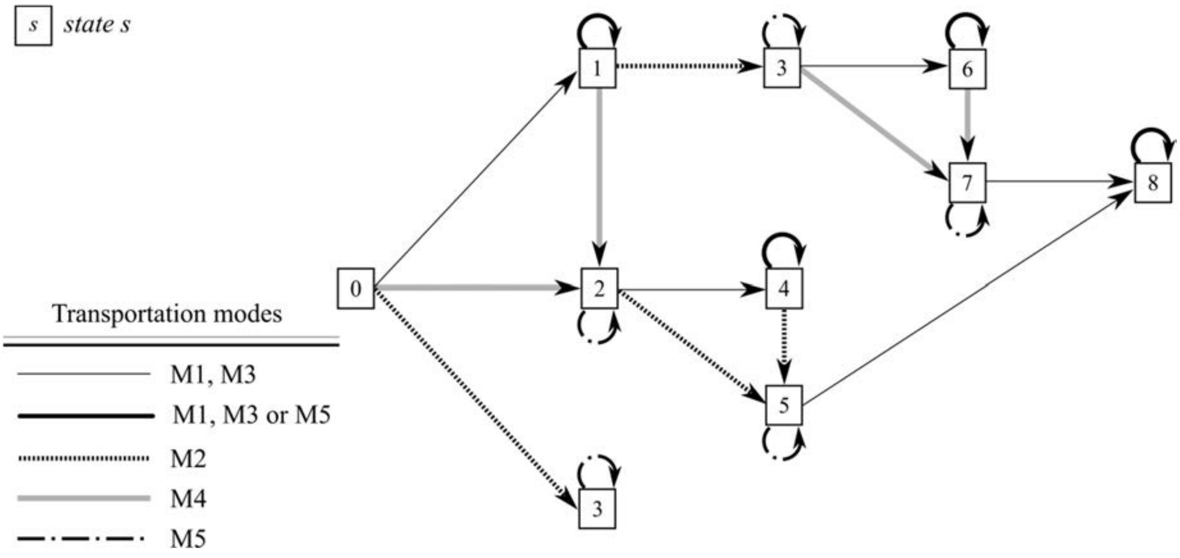

Figure 2: DFA graph with bicycle and rail transit mode restrictions

For each pair of *states* s and s', such that the modes accepted in s are also accepted in s', it is possible to define a relationship to know which *state dominates* the other. This relationship defines an order between any pair of hyperpaths for knowing which hyperpath has a better chance to grow. Artigues et al. (2013) defines the *dominance* between *states* as follows:

Definition 1. *Let s and s' be two states such that all modes accepted in s are also accepted in s'. Define $s \ll s'$, s' dominates s, if for any modes $m \in M$ and $m' \in M$, such that m is a feasible mode for s, one of the following conditions holds:*

- $\delta(m, m', s') = \emptyset$,
- $\delta(m, m', s') = \delta(m, m', s)$
- $\delta(m, m', s) = s$ and $\delta(m, m', s') = s'$.

With the Definition 1 the set of *states* dominating s, $PS_s = \{s' \in S : s \ll s'\}$, is obtained. The *hyper transition* of *states* is defined to get the resulting *state* when a boarding hyperarc e concatenates two hyperpaths. For example, if $e = (i, h(e) = \{j, k\})$ concatenates the hyperpaths $p_{jd}^{s_x}$ and $p_{kd}^{s_y}$, then *hyper transition* of the *states* is $s_x \circ s_y = s_e$, such that s_e indicates the combination of modes used in $p_{jd}^{s_x}$ and $p_{kd}^{s_y}$ (Lozano & Storchi, 2002).

Let p_{id}^{s} be a hyperpath from i to d with *state* s. The *expected travel time* of p_{id}^{s} (Lozano & Storchi, 2002) is recursively defined in Equation 1.

$$\lambda_{id}^{s} = \left\{\left[\tau_{ij} + \lambda_{jd}^{s'} \text{ if } (i, j) \text{ is an arc}\right] or\ [\omega_{e^b} + \sum_{j \in h(e^b)} \pi_{e^b, l} \cdot \lambda_{jd}^{s'} \text{ if } e^b \text{ is a } boarding \text{ hyperarc}]\right\} \quad (1)$$

The *penalty for riding* a transit vehicle models the user perception of the onboard crowding levels. It is assumed that:

- No penalty is given if the user travels seated or there is enough personal space in the vehicle.
- A penalty is given if the personal space in the vehicle is limited.
- The user will not board vehicles at full capacity.

The *penalty for riding* a transit vehicle is an exponential function that depends on the seating capacity and the people on board, this penalty is described as follows. Let $e' = (i, j)$ be a *contained* hyperarc and let (j, k) and (j, m) two arcs. The *contained* hyperarc e' represents boarding a transit vehicle at stop i. The arc (j, k) represents riding the vehicle from stop i to stop k and descending at stop k. The arc (j, m) represents riding the vehicle from stop i to stop k and continuing the trip.

Let $\theta_{jk}(l) = f_{jk}(l)/s(l)$ be the *seat load factor* of the arc (j, k) associated to the line l, such that $f_{jk}(l)$ is the average passenger flow of the arc (j, k) associated to the line l and $s(l)$ is the number of seats of a line l vehicle. Note that $\theta_{jk}(l) = \theta_{jm}(l)$ since both arcs (j, k) and (j, m) represent riding a transit vehicle of the line l from stop i to stop k.

The *seat load factor* may change from one stop to another. For example, consider the sub-hyperpath $p_{id}(l) = \big(e = (i, j), (j, j_1), (j_1, j_2), (j_2, j_3), (j_3, d)\big)$, such that $\{i, k_1, k_2, k_3, d\}$ are consecutive stops of l, see figure 3. A user boarding at stop i and going to stop d, will experience different levels of crowd during the trip, hence the *seat load factor* might change for consecutive arcs, i.e., $\theta_{jj_1}(l) \neq \theta_{j_1j_2}(l) \neq \theta_{j_2j_3}(l) \neq \theta_{j_3d}(l)$. An example of such phenomenon is represented in figure 3 where the passengers getting on and off the vehicle are represented with the red and green pawns, respectively, and the user going from i to d is the black pawn. For instance in figure 3 $\theta_{j_1j_2}(l) \neq \theta_{j_2j_3}(l)$, because from stop j_1 to stop j_2 there are three passengers in the vehicle while from stop j_2 to stop j_3 there are two.

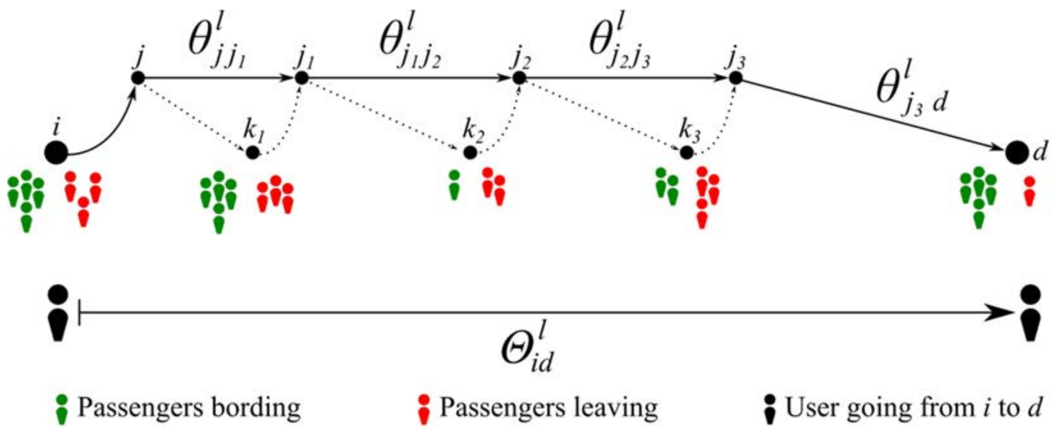

Figure 3: Seat load factor of the line l from the stop i to the stop d

To estimate the *seat load factor* variations of the sub-hyperpath $p_{id}(l)$, in Equation 2 is recursively defined the line *seat load factor* $\Theta_{id}(l)$ of the sub-hyperpath $p_{id}(l)$ as the incremental weighted average of the arc *seat load factors*.

$$\Theta_{id}(l) = \left\{ \left[(\tau_{ij}\left(\theta_{ij}(l) - \Theta_{jd}(l)\right))/T_{id}(l) + \Theta_{jd}(l) \text{ if } (i,j) \text{ is an arc} \right] \ or\ [\Theta_{jd}(l) \text{ if } (i,j) \text{ is a } contained \text{ hyperarc}] \right\} \quad (2)$$

such that, $T_{id}(l)$ is the in-vehicle travel time of the sub-hyperpath $p_{id}(l)$.

To capture the dispersion of the arc *seat load factors* of $p_{id}(l)$, in Equation 3 is recursively defined the standard deviation of the arc *seat load factor*, $\sigma_{id}(l)$, of the sub-hyperpath $p_{id}(l)$.

$$\sigma_{id}(l) = \left\{ \left[\sqrt{((\left(\sigma_{jd}(l)\right)^2 + \tau_{ij}\left(\theta_{ij}(l) - \Theta_{jd}(l)\right)\left(\theta_{ij}(l) - \Theta_{jd}(l)\right))/T_{id}(l))} \text{ if } (i,j) \text{ is an arc} \right], \ or\ [\sigma_{id}(l) \text{ if } (i,j) \text{ is a } contained \text{ hyperarc}] \right\} \quad (3)$$

Users are penalized for riding transit vehicles according to their preferences on onboard crowding levels. For example, some users might like to travel seated all the time, so they are penalized if $\Theta_{id}(l) > 1$. Two bounds are set; one is for know when a user is not penalized for riding transit vehicles, and the other is for know when a user will not board a transit vehicle. Let $\underline{\delta}$ be the *lower seat load factor*, such that if $\Theta_{id}(l) < \underline{\delta}$ the user is not penalized for riding the line l from the stop i to d. Let $\overline{\delta}$ be the *upper seat load factor*, such that if $\Theta_{id}(l) > \overline{\delta}$ the user will not ride the line l from the stop i to d. In Equation 4 is defined the *penalty for riding* the sub-hyperpath $p_{id}(l)$ (Qin, 2014).

$$r_{id}(l) = \{[0 \text{ if } \Theta^l_{id} < \underline{\delta}],\ [T_{id}(l)/(1 + e^{1000(1-\Theta_{id}(l))}) + \beta T_{id}(l) e^{4(\Theta_{id}(l) - \underline{\delta})} \text{ if } \underline{\delta} \leq \Theta^l_{id} \leq \overline{\delta}],\ [\infty \text{ if } \Theta^l_{id} > \overline{\delta}]\} \quad (4)$$

such that, $\beta \in [0, \infty\}$ is a parameter that models the user haste, bigger β's means the penalties are more considerable for ridding crowded buses. In contrast, with smaller β's, the user is less penalized. For $\Theta_{id}(l) > \overline{\delta}$ the penalty for riding the sub-hyperpath $p_{id}(l)$ is infinity, meaning that users will not ride the line l from node i to node d.

Figure 4 shows the plot of $r_{id}(l)$ over $\Theta_{id}(l)$ such that $T_{id}(l) = 1$, $\underline{\delta} = 1.4$, $\overline{\delta} = 2.5$ and $\beta \in \{0, 0.25, 0.5, 0.75, 1\}$.

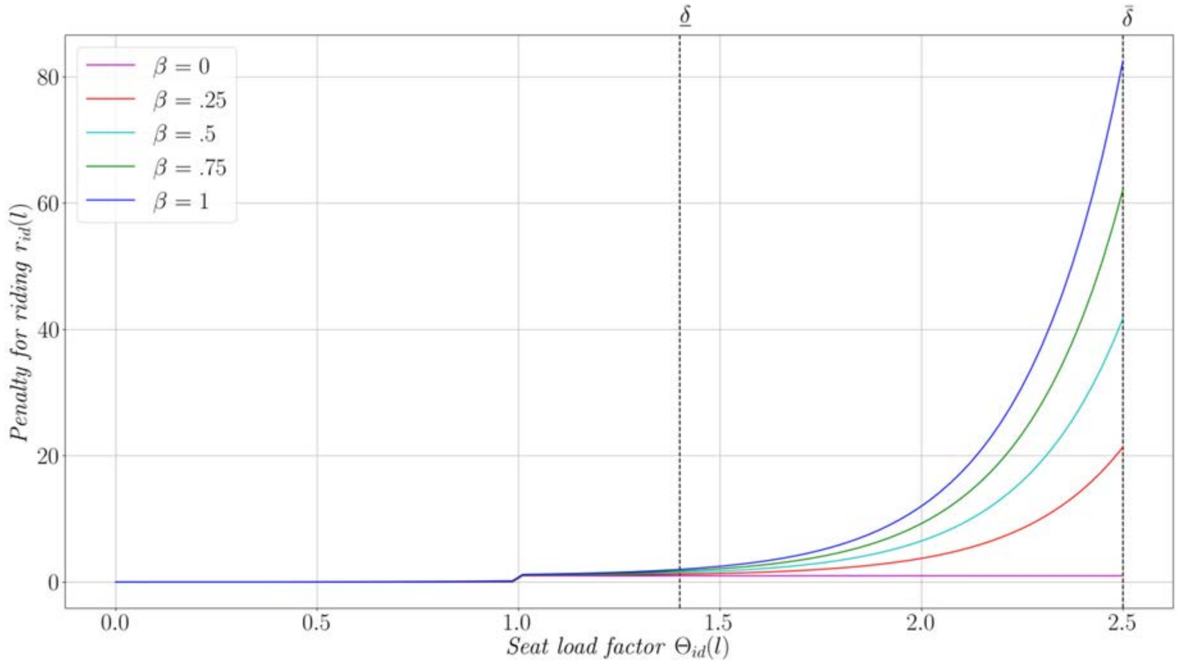

Figure 4: Penalty for riding with different $\boldsymbol{\beta}$ parameters

As shown in figure 4, $r_{id}(l)$ is close to zero when $\beta = 0$ (see magenta line), i.e., impatient or hasty users are not penalized for riding crowded transit since they value more their time and will ride any line that takes them fast to their destination. While with bigger β's, the penalty for riding transit grows exponentially, in this case, users are highly penalized, and therefore less crowded lines are preferred.

To compute the *penalty for riding* the hyperpath p_{id}^{s}, the *expected seat load* factor of p_{id}^{s} is recursively defined in Equation 5.

$$\Theta_{id}^{s} = \left\{ \left[(\mathrm{T}_{id}(l) \left(\theta_{id}(l) - \Theta_{jd}^{s'}(-l)\right)) / \mathrm{T}_{id}^{s} + \Theta_{jd}^{s'}(-l) \text{ if } (i,j) \text{ is an arc} \right],\ \left[\sum_{j\in h(e^b)} \pi_{e^b,l} \cdot \Theta_{jd}^{s} \text{ if } e^b \text{ is an hyperarc}\right] \right\} \quad (5)$$

such that, T_{id}^{s} is the in-vehicle travel time of p_{id}^{s}, $\Theta_{jd}^{s'}(-l)$ is *expected seat load factor* of the hyperpath $p_{jd}^{s'} \setminus p_{jd}(l)$, i.e., the sub-hyperpath $p_{jd}(l)$ is not included in the computation of $\Theta_{jd}^{s'}(-l)$.

The standard deviation of the *expected seat load factor* of $p_{id}(t)$ is recursively defined in Equation 6.

$$\Theta_{id}^{s} = \left\{ \left[\sqrt{(((\sigma_{jd}^{s'}(-l))^{2} + \mathrm{T}_{id}(l) \left(\Theta_{id}(l) - \Theta_{jd}^{s'}(-l)\right) (\Theta_{id}(l) - \Theta_{id}^{s})) / \mathrm{T}_{id}^{s})} \text{ if } (i,j) \text{ is an arc} \right], \left[\sum_{j\in h(e^b)} \pi_{e^b,l} \cdot \sigma_{jd}^{s'} \text{ if } e^b \text{ is an hyperarc}\right] \right\} \quad (6)$$

such that, $\sigma_{jd}^{s'}(-l)$ is *expected seat load factor* of the hyperpath $p_{jd}^{s'} \setminus p_{jd}(l)$.

Finally, the *penalty for riding* the hyperpath p_{id}^{s} is defined in Equation 7.

$$r_{id}^{s} = \{[0 \text{ if } \Theta_{id}^{s} < \underline{\delta}],\ [\mathrm{T}_{id}^{s}/(1+e^{1000(1-\Theta_{id}^{s})}) +\ \beta \mathrm{T}_{id}^{s} e^{4(\Theta_{id}^{s}-\underline{\delta})} \text{ if } \underline{\delta} < \Theta_{id}^{s} \leq \overline{\delta}],$$
$$[\infty \text{ if } \Theta_{id}^{s} > \overline{\delta}]\} \qquad (7)$$

5. SHORTEST HYPERPATH ALGORITHM

A label-correcting algorithm is presented for finding the shortest hyperpaths. The algorithm finds a set of hyperpaths with minimum *expected travel time,* bounded by modal transfers and *seat load factors*. The pseudo-code of the algorithm is shown in Section 5.1. Before starting the shortest hyperpath computation, users select their origin $o \in V$, destination $d \in V$, maximum modal transfers w_{max}, and *seat load factors* bounds ($\underline{\delta}$ and $\overline{\delta}$). The interval from $\underline{\delta}$ to $\overline{\delta}$ is divided into two equal parts by δ' to obtain the *load factor levels* $\delta_1 = \underline{\delta}$, $\delta_2 = \delta'$ and $\delta_3 = \overline{\delta}$. The set P_i is created for storing all *viable* hyperpaths starting at node i.

The sets of node-*state* pairs P_{now}, P_{next}, and Q are created for looping through the arcs in $B(j)$. On the first iteration $P_{now} = \{[d, 0]\}$ and $P_{next} = \emptyset$ and $Q = \emptyset$. Storing elements in P_{now}, P_{next} and Q are for finding the shortest hyperpaths in incremental order with respect to *load factor levels* and modal transfers. The shortest hyperpaths with no modal transfers and *load factor level* below δ_1 are found in the first round. Then, the algorithm finds the shortest hyperpaths with no modal transfers and *load factor level* below δ_2. The hyperpaths with no modal transfers and *load factor level* below δ_3 are found on the next round. After all the *load factor levels* are scanned, the number of modal transfers w is incremented by one unit, and the algorithm finds the shortest hyperpaths with one modal transfer and *load factor level* below δ_1; the process is repeated for δ_2 and δ_3. The algorithm ends when $w > w_{max}$ or when all *load factor levels* are scanned for each modal transfer, i.e., $P_{now} = \emptyset$.

All labels of the algorithm are initialized as follows:

- $\lambda_{id}^{s} = \infty, w_{id}^{s} = 0\ \forall\ i \in V$ and $s \in S$
- $C_{e^b}^{*} = \infty, \Phi_{e^b}^{*} = \infty, h_{e^b}^{*} = \emptyset\ \ \forall\ e^b \in E$
- $\lambda_{dd}^{0} = 0; w = 0; \delta = \delta_1\ ; P_{now} = \{[d, 0]\}; P_{next} = \emptyset; \mathrm{Q} = \emptyset$
- $P_i = \emptyset\ \forall\ i \in V \setminus \{d\}; P_d = \{p_{dd}^{0}\}$

If $P_{now} \neq \emptyset$, $\delta \leq \overline{\delta}$ and $w \leq w_{max}$, get $(i, j) \in B(j)$ and check if the last *expected travel time* of p_{jd}^{s} $(last(p_{jd}^{s}))$ is greater than the actual *expected travel time* of p_{jd}^{s}. This condition eliminates *dominated* hyperpaths by checking if in previous loops a fastest hyperpath was found.

The sub-procedures *getLambda* and *getHyperLambda* concatenates arcs and *contained* hyperarcs, respectively. The procedure *getLambda* concatenates (i,j) to hyperpath p_{jd}^{s} as follows:

- If $\Theta_{jd}^{s} > \overline{\delta}$ or $\theta_{ij}(l) > \overline{\delta}$ the concatenation is canceled because of the user restriction on *seat load factors*.
- With the function *getStateAndTransfers*, get the *state* s' that results from concatenating (i,j) with the hyperpath p_{jd}^{s} (see the DFA shown infigure 2). If the concatenation produces more than w_{max} modal transfers, the hyperpath is not *viable* and the procedure terminates
- Check if $\tau_{ij} + \lambda_{jd}^{s}$ is lower than $\lambda_{id}^{s'} : p_{id}^{s'} \in P_i$. If the condition does not hold, the new hyperpath is no generated because it is *dominated* by $p_{id}^{s'}$. Otherwise, continue to the next step
- Check if $\tau_{ij} + \lambda_{jd}^{s}$ is lower than $\lambda_{id}^{s''} : p_{id}^{s''} \in P_i$ for at least one $s'' \in PS_s$. If the condition does not hold, the new hyperpath is no generated because it is *dominated* by at least one $p_{id}^{s''} \in P_i : s'' \in PS_s$. Otherwise, continue to the next step.
- Concatenate (i,j) with the hyperpath p_{jd}^{s} to generate a *viable* hyperpath $p_{id}^{s'}$ and add $p_{id}^{s'}$ to P_i
- If the mode of (i,j) is $M2$ or $M3$. Compute $\Theta_{id}(l)$, $\sigma_{id}(l)$, $r_{id}(l)$, $\Theta_{id}^{s'}$, $\sigma_{id}^{s'}$, and $r_{id}^{s'}$ with Equations (2-7). If the mode of (i,j) is different from $M2$ or $M3$, then $\theta_{id}(l) = \theta_{jd}(l)$, $\sigma_{id}(l) = \sigma_{jd}(l)$, $r_{id}(l) = r_{jd}(l)$, $\Theta_{id}^{s'} = \Theta_{jd}^{s}$, $\sigma_{id}^{s'} = \sigma_{jd}^{s}$, and $r_{id}^{s'} = r_{jd}^{s}$
- If the mode of (i,j) is a modal transfer, add the pair $[i,s']$ to Q, unless it is already in there. If the mode of (i,j) is not a modal transfer consider the following cases:

    - If $\theta_{ij}^{l} \leq \delta$ add $[i,s']$ to P_{now}, unless it is already in there
    - If $\theta_{ij}^{l} > \delta$ add $[i,s']$ to P_{next}, unless it is already in there

The procedure *getHyperLambda* concatenates the *contained* hyperarc $e' = (i,j)$ to hyperpath p_{jd}^{s} as follows:

1. If $\Theta_{jd}^{s} > \overline{\delta}$, the concatenation is canceled because the restrictions on *seat load factors*
2. Check if $p_{id}^{\hat{s}} \in P_i$, such that $e^b = \{i, h^b(e)\}$ is the last hyperarc of $p_{id}^{\hat{s}}$. If $p_{id}^{\hat{s}} \neq \emptyset$ then $s_e = s \circ \hat{s}$, otherwise $s_e = s$
3. If $\lambda_{jd}^{s} \geq \lambda_{id}^{\hat{s}}$ the hyperpath p_{jd}^{s} is *dominated* by $p_{id}^{\hat{s}}$ and the concatenation does not proceed. Otherwise, continue to the next step

4. Let $e^b = (i, h^b(e) \cup \{j\})$, compute $\Phi^*_{e^b}$ as the sum of the average arrival-rates of all the lines associated to e^b. If $\Phi^*_{e^b} = \varphi_{ij}$ continue to step 4.1, otherwise continue to step 4.2
   4.1. Add the estimated waiting time of the *contained* hyperarc (i,j) to the *expected travel time* of the hyperpath p^s_{jd}, $C^*_{e^b} = \frac{1}{\varphi^l_{ij}} + \lambda^s_{jd}$. Since (i,j) is a *contained* hyperarc and p^s_{jd} is a hyperpath starting at i, then the *expected seat load factor* and the *penalty for riding* are exported inherited from the hyperpath p^s_{jd}, i.e., $\Theta^*_{id} = \Theta^s_{jd}$ and $r^*_{id} = r^s_{jd}$.
   4.2. Compute $C^*_{e^b}$ as the incremental weighted average of $C^*_{e^b}$ and λ^s_{jd}. In this case, the algorithm is concatenating the hyperpaths p^s_{jd} and $p^{\hat{s}}_{id}$ with the *contained* hyperarc (i,j). Hence is necessary to compute (in Section 5.1, see the sub-procedure *getIncHPenalty*):
      4.2.1. $\Theta^*_{e^b}$ as the incremental weighted average of $\Theta^*_{e^b}$ and Θ^s_{jd}
      4.2.2. $\sigma^*_{e^b}$ as the incremental weighted average of $\sigma^*_{e^b}$ and σ^s_{jd}
      4.2.3. $r^*_{e^b}$ with the Equation 7
5. If $C^*_{e^b} \leq \lambda^{s_e}_{id} : \lambda^{s_e}_{id} \in P_i, s_e = s \circ \hat{s}$, and $\Theta^*_{e^b} \leq \overline{\delta}$, get the set PS_{s_e} and continue to the next step, otherwise the new hyperpath is dominated by $p^{s_e}_{id}$ or the new hyperpath is restricted by the user *upper load factor*.
6. If $C^*_{e^b} \leq \lambda^{s''}_{id}$ for at least one $s'' \in PS_{s_e}$, continue to the next step. Otherwise, the new hyperpath is not generated because it is *dominated* by $p^{s''}_{id}$
7. Concatenate $e' = (i,j)$ with the hyperpath p^s_{jd} (and with the hyperpath $p^{\hat{s}}_{id}$, if exists) to produce a new *viable* hyperpath $p^{s_e}_{id}$
8. Make $r^*_{e^b} = r^{s_e}_{id}$, $\Theta^*_{e^b} = \Theta^{s_e}_{id}$, $r_{jd}(l) = r_{id}(l)$ and $\Theta_{jd}(l) = \Theta_{id}(l)$. Note that the penalty for riding the sub-hyperpath $p_{id}(l)$ and the *seat load factor* of $p_{id}(l)$ are inherited from the sub-hyperpath $p_{jd}(l)$ because (i,j) is a *contained* hyperarc.
9. Add $[i, s_e]$ to P_{now}, unless it is already in there.

When the algorithm ends, it generates a solution set, P_i, of *viable* hyperpaths. To obtain the *Pareto-Optimal* set of solutions, select from P_i the non-dominated hyperpaths regarding the *expected travel time*, the modal transfers, and the *penalty for riding*. From the *Pareto-Optimal* set, the users choose the hyperpath they prefer according to their personal views, such as the number of transfers or onboard crowding conditions of the ride. The *expected seat load factor* standard deviation, σ^s_{id}, can be used for comparing two hyperpaths in terms of the passengers onboard. For instance, if two hyperpaths have similar penalties but one of them has a smaller σ, a user may prefer the hyperpath with the small σ since the onboard crowd will be more stable during the trip.

5.1. Shortest hyperpaths algorithm pseudo-code

Procedure. *Combined Real-Time Shortest Hyperpaths*

$\lambda_{id}^{s} = \infty, w_{id}^{s} = 0 \ \forall \ i \in V$ and $s \in S$

$C_{e^b}^{*} = \infty, \Phi_{e^b}^{*} = \infty, h_{e^b}^{*} = \emptyset \ \ \forall \ e^b \in E$

$\lambda_{dd}^{0} = 0; w = 0; \delta = \underline{\delta}; P_{now} = \{[d, 0]\}; P_{next} = \emptyset; \mathrm{Q} = \emptyset$

$P_i = \emptyset \ \forall \ i \in V \setminus d; P_d = \{p_{dd}^{0}\}$

WHILE $w \leq w_{max}$ AND $P_{now} \neq \emptyset$

    WHILE $\delta \leq \overline{\delta}$ AND $P_{now} \neq \emptyset$

        WHILE $P_{now} \neq \emptyset$

            SELECT $[j, s]$ FROM P_{now}

                $P_{now} = P_{now} \setminus [j, s]$

                FOR $(i, j) \in B(j)$

                    IF $\lambda_{jd}^{s} < last(\lambda_{jd}^{s})$

                            $last(\lambda_{jd}^{s}) = \lambda_{jd}^{s}$

                        IF (i, j) is an arc

                            $\mathrm{get}Lambda(s, (i, j), \lambda_{jd}^{s})$

                        IF (i, j) is a *contained* hyperarc

                            $getHyperLambda(s, (i, j), \lambda_{jd}^{s})$

            $P_{now} = P_{next}; P_{next} = \emptyset; \delta = \delta + \delta'$

        $P_{now} = Q; Q = \emptyset; w = w + 1; c = \underline{\delta}$

Sub-procedure. $getLambda(s_x, (i, j), \lambda_{jd}^{s_x})$

IF $\Theta_{jd}^{s} > \overline{\delta}$ OR $\theta_{ij}(l) > \overline{\delta}$

    BREAK

$s, transfers = getStateAndTransfers(s_x, (i, j))$

IF $s \neq 0$ AND $\tau_{ij} + \lambda_{jd}^{s_x} \leq \lambda_{id}^{s}$

    $PS_s = preferredStates(s)$

ELSE

    $PS_s = \emptyset$

FOR $s_y \in PS_s$

    IF $\tau_{ij} + \lambda_{jd}^{s_x} \leq \lambda_{id}^{s_y}$

        $\lambda_{id}^{s} = \tau_{ij} + \lambda_{jd}^{s_x}$

        $w_{id}^{s} = w_{jd}^{s_x} + transfers$

        IF $(i, j) \in \{M2, M3\}$

            $r_{id}(l), \Theta_{id}(l) = getLPenalty\left((i, j), p_{jd}^{s_x}\right)$

            $r_{id}^{s}, \Theta_{id}^{s} = getHPenalty\left((i, j), p_{jd}^{s_x}\right)$

        ELSE

$r_{id}(l), \Theta_{id}(l), r_{id}^{s}, \Theta_{id}^{s} = r_{jd}(l), \Theta_{jd}(l), r_{jd}^{s_x}, \Theta_{jd}^{s_x}$

IF $[i, s] \notin P_{now}$ AND $transfers = 0$ AND $\theta_{ij}(l) \leq \delta$

$P_{now} = P_{now} \cup [i, s]$

IF $[i, s] \notin P_{next}$ AND $transfers = 0$ AND $\theta_{ij}(l) > \delta$

$P_{next} = P_{next} \cup [i, s]$

IF $[i, s] \notin Q$ AND $transfers = 1$

$Q = Q \cup [i, s]$

BREAK

Sub-procedure. $getHyperLambda\left(s_x, (i, j), p_{jd}^{s_x}\right)$

IF $\Theta_{jd}^{s_x} > \overline{\delta}$

BREAK

IF $p_{id}^{s} \neq \emptyset$

$s_e = getHyperState(s_x, s)$

ELSE

$s_e = s_x$

IF $\lambda_{jd}^{s_x} < \lambda_{id}^{s}$

$\Phi_{e^b}^{*} = \Phi_{e^b}^{*} + \varphi_{ij}^{l}$

IF $\Phi_{e^b}^{*} = \varphi_{ij}^{l}$

$C_{e^b}^{*} = \frac{1}{\varphi_{ij}^{l}} + \lambda_{jd}^{s_x}$

$R_{e^b}^{*}, \Theta_{e^b}^{*} = R_{jd}^{s_x}, \Theta_{jd}^{s_x}$

ELSE

$C_{e^b}^{*} = C_{e^b}^{*} - \frac{\left(C_{e^b}^{*} - \lambda_{jd}^{s_x}\right)\varphi_{ij}^{l}}{\Phi_{e^b}^{*}}$

$r_{e^b}^{*}, \Theta_{e^b}^{*} = getIncHPenalty\left((i, j), p_{jd}^{s_x}, p_{id}^{s}\right)$

IF $C_{e^b}^{*} \leq \lambda_{id}^{s_e}$ AND $\Theta_{e^b}^{*} \leq \overline{\delta}$

$PS_{s_e} = preferredStates(s_e)$

ELSE

$PS_{s_e} = \emptyset$

FOR $s_y \in PS_{s_e}$

IF $C_{e^b}^{*} \leq \lambda_{id}^{s_y}$

$\lambda_{id}^{s_e} = C_{e^b}^{*}$

$h_{e^b}^{*} = h_{e^b}^{*} \cup \{j\}$

$w_{id}^{s_e} = \max\left\{w_{jd}^{s_x}, w_{id}^{s}\right\}$

$r_{id}^{s_e}, \Theta_{id}^{s_e} = r_{e^b}^{*}, \Theta_{e^b}^{*}$

$r_{id}(l), \Theta_{id}(l) = r_{jd}(l), \Theta_{jd}(l)$

IF $[i, s_e] \notin P_{now}$

$P_{now} = P_{now} \cup [i, s_e]$
BREAK

Sub-procedure. $getLPenalty\left((i,j), p_{jd}^{s_x}\right)$

$\Theta_{id}(l) = \frac{\tau_{ij}\left(\theta_{ij}(l) - \Theta_{jd}(l)\right)}{T_{id}(l)} + \Theta_{jd}(l)$

$T_{id}(l) = \tau_{ij} + T_{jd}(l)$

IF $\Theta_{id}(l) \leq \underline{\delta}$

$r_{id}(l) = 0$

ELSE

$r_{id}(l) = \frac{T_{id}(l)}{1+e^{1000\left(1-\Theta_{id}(l)\right)}} + \beta T_{id}(l) e^{4\left(\Theta_{id}(l) - \underline{\delta}\right)}$

Sub-procedure. $getHPenalty\left((i,j), p_{jd}^{s_x}\right)$

$\Theta_{id}^{s} = \frac{\mathrm{T}_{id}(l)\left(\theta_{id}(l) - \Theta_{jd}^{s_x}(-l)\right)}{\mathrm{T}_{id}^{s}} + \Theta_{jd}^{s_x}(-l)$

$\mathrm{T}_{id}^{s} = \tau_{ij} + \mathrm{T}_{jd}^{s_x}$

IF $\Theta_{id}^{s} \leq \underline{\delta}$

$r_{id}^{s} = 0$

ELSE

$r_{id}^{s} = \frac{\mathrm{T}_{id}^{s}}{1+e^{1000\left(1-\Theta_{id}^{s}\right)}} + \beta \mathrm{T}_{id}^{s} e^{4\left(\Theta_{id}^{s} - \underline{\delta}\right)}$

Sub-procedure. $getIncHPenalty\left((i,j), p_{jd}^{s_x}, p_{id}^{s}\right)$

$\Theta_{e^b}^{*} = \Theta_{e^b}^{*} + \frac{\left(\Theta_{e^b}^{*} - \Theta_{jd}^{s_x}\right)\varphi_{ij}}{\Phi_{e^b}^{*}}$

$\mathrm{T}_{e^b}^{*} = \mathrm{T}_{e^b}^{*} + \frac{\left(\mathrm{T}_{e^b}^{*} - \mathrm{T}_{jd}^{s_x}\right)\varphi_{ij}}{\Phi_{e^b}^{*}}$

IF $\Theta_{e^b}^{*} \leq \underline{\delta}$

$r_{e^b}^{*} = 0$

ELSE

$r_{e^b}^{*} = \frac{\mathrm{T}_{e^b}^{*}}{1+e^{1000\left(1-\Theta_{e^b}^{*}\right)}} + \beta \mathrm{T}_{e^b}^{*} e^{4\left(\Theta_{e^b}^{*} - \underline{\delta}\right)}$

6. EXAMPLE

Figure 5 shows a multimodal hypergraph composed of three modes; pedestrian ($M1$), buses ($M2$), and private bicycles ($M3$). The mode $M5$ is the modal transfer arcs. The pairs $\left(\tau_{ij}, \theta_{ij}(l)\right)$ on top of the arcs are the arc's travel time and *seat load factor*. On the top of the *contained* hyperarcs is the associated average arrival-rate of the bus line.

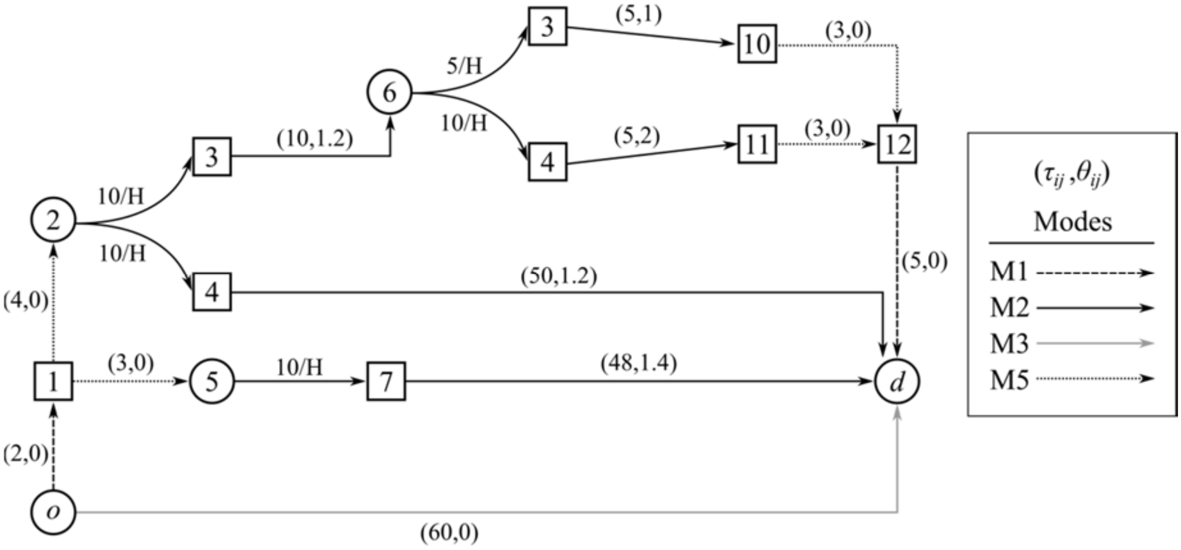

Figure 5: Example hypergraph with three modes and viable hyperpaths

Figure 6 shows the results of computing the shortest hyperpaths from the node o to the node d with a maximum of three modal transfers such that, $\underline{\delta} = 1$, $\overline{\delta} = 1.4$, and $\beta = 1$. The vector $(\lambda_{id}^{s}, w_{id}^{s}, s, r_{id}^{s})$ near the nodes indicates the *expected travel time*, the maximum modal transfers, the *state*, and the *penalty for riding* of the *viable* hyperpath p_{id}^{s}. For example, the vector $(25,1,1,7.5)$ on top of node 6 is associated with the *viable* hyperpath p_{6d}^{1}, such that $\lambda_{6d}^{1} = 25$, $w_{6d}^{1} = 1$, $s = 1$ and $r_{6d}^{1} = 7.5$. The *viable* hyperpath p_{6d}^{1} is composed of the *contained* hyperarc $(6,3)$ and the arcs $(3,10)$, $(10,12)$, and $(12, d)$. Since $\theta_{4,11} = 2$ and $\overline{\delta} = 1.4$, the *contained* hyperarc $(6,4)$ and its consecutive arcs are not in p_{6d}^{1} as the user prefers not to board vehicles with such onboard crowd levels. The *Pareto-Optimal* set of solutions are the starred vectors on top of node o.

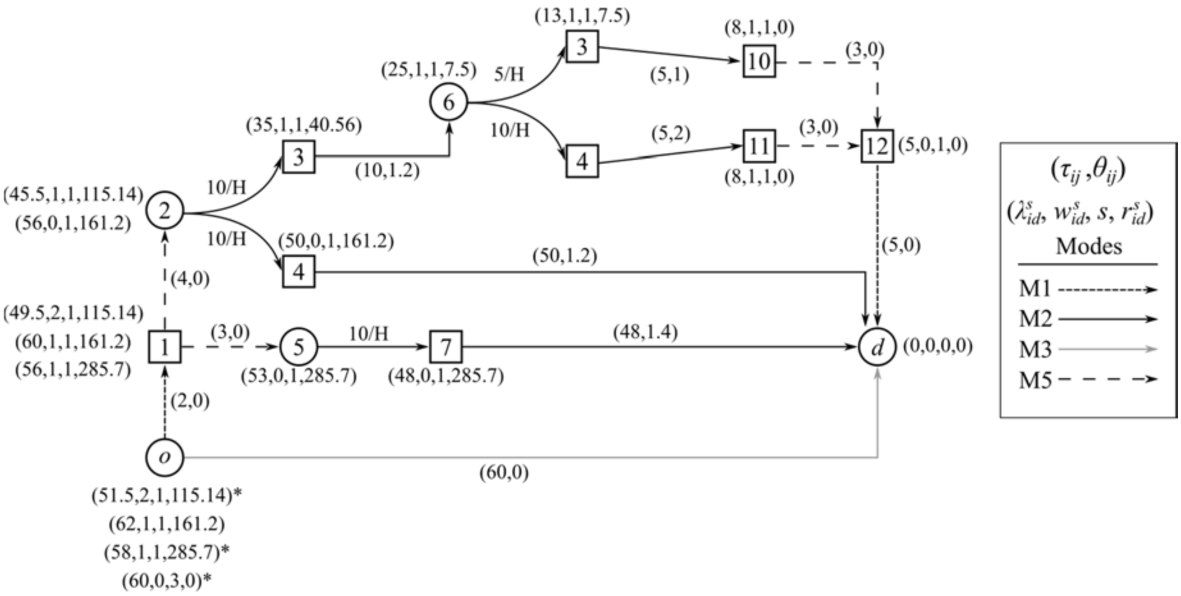

Figure 6: Solution set of viable hyperpaths

Figure 7 shows the three shortest hyperpaths in the Pareto-Optimal set. In the first solution, the user rides a bicycle for 60min to get to the destination. Solution 2 is 58 minutes long; to reach the destination, the user walks to location 1 and then goes to stop 5 for boarding a bus with high onboard crowd levels. Solution 3 is the fastest solution with the lowest onboard crowd levels. However, this solution has the most modal transfers. In solution 3, the user walks to location 1 and then at stops 2 boards the first arriving bus to reach the destination. The decision on selecting the path depends on the user inclinations. Take a long trip by bike to avoid buses (solution 1). Ride a single bus with high onboard crowd levels (solution 2). Alternatively, make many modal transfers to ride buses with low onboard crowd levels (solution 3).

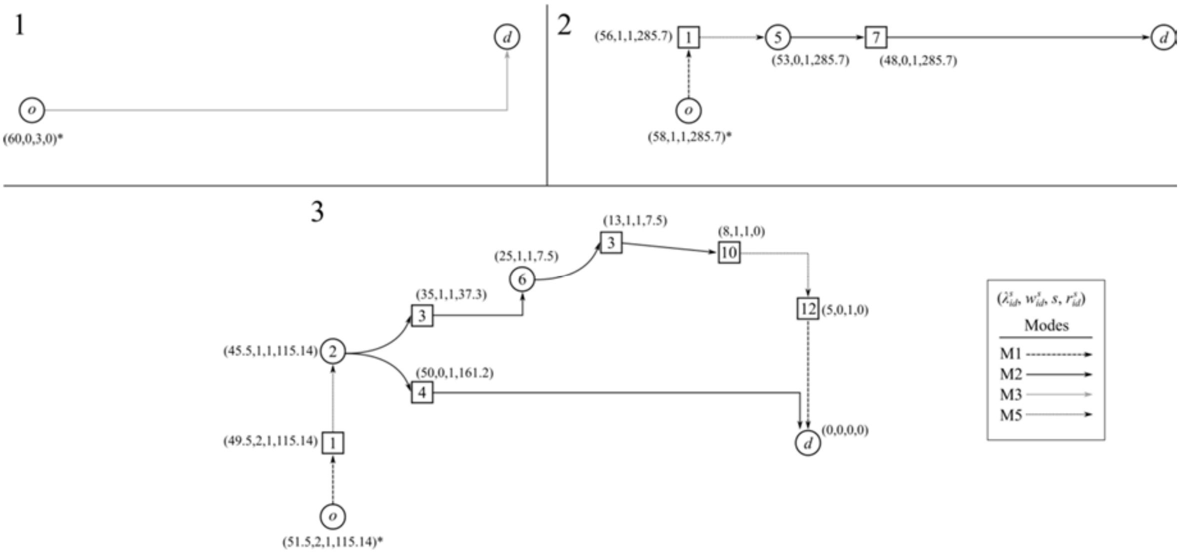

Figure 7: Shortest hyperpath in the Pareto-Optimal set

7. CONCLUSIONS

The presented paper describes a hypergraph model and routing algorithm to find the shortest hyperpaths in a multimodal network where the transit has random-arrivals and are considered the user constraints on the sequence of boarded modes and onboard crowd levels. The *states* and *seat load factors* transform the 'traditional' route planners (that only consider generic factors such as time, cost, or transfers) into personal trip assistants with tailored routes. Nowadays, the enormous amounts of accessible transportation-data set an ideal ground for developing tailored routing algorithms to user habits or idiosyncrasies. In light of the COVID19 pandemic, the presented model and algorithm could be used to find the paths with the lowest levels of onboard crowd and thus reduce the risk of contagion. However, for this tool to be valuable and accurate for COIVD19 applications, real-time information on onboard crowd levels is required.

REFERENCES

ARTIGUES, C., HUGUET, M.J., GUEYE, F., SCHETTINI, F., DEZOU, L., (2013). State-based accelerations and bidirectional search for bi-objective multi-modal shortest paths. Transportation Research Part C: Emerging Technologies 27, 233–259.

BANSAL, P., HURTUBIA, R., TIRACHINI, A., DAZIANO, R.A., (2019). Flexible estimates of heterogeneity in crowding valuation in the New York City subway. Journal of Choice Modelling 31, 124–140.

BATARCE, M., MUÑOZ, J.C., ORTÚZAR, J. DE D., (2016). Valuing crowding in public transport: Implications for cost-benefit analysis. Transportation Research Part A: Policy and Practice 91, 358–378.

COMI, A., NUZZOLO, A., (2016). Individual utility-based path suggestions in transit trip planners. IET Intelligent Transport Systems 10, 219–226.

HAYWOOD, L., KONING, M., MONCHAMBERT, G., (2017). Crowding in public transport: Who cares and why? Transportation Research Part A: Policy and Practice 100, 215–227.

KATONA, G., JUHÁSZ, J., LÉNÁRT, B., (2017). Travel habit based multimodal route planning. Transportation Research Procedia 27, 301–308.

LÓPEZ, D., LOZANO, A., (2020). Shortest hyperpaths in a multimodal hypergraph with real-time information on some transit lines. Transportation Research Part A: Policy and Practice 137, 541–559.

LOZANO, A., STORCHI, G., (2002). Shortest viable hyperpath in multimodal networks. Transportation Research Part B: Methodological 36, 853–874.

MÁRQUEZ, L., ALFONSO A, J. V., POVEDA, J.C., (2019). In-vehicle crowding: Integrating tangible attributes, attitudes, and perceptions in a choice context between BRT and metro. Transportation Research Part A: Policy and Practice 130, 452–465.

NGUYEN, S., PALLOTTINO, S., (1989). Hyperpaths and shortest hyperpaths, in: Simeone, B. (Ed.), Combinatorial Optimization, Lecture Notes in Mathematics. Springer Berlin Heidelberg, pp. 258–271.

NUZZOLO, A., CRISALLI, U., COMI, A., ROSATI, L., (2015). Individual Behavioural Models for Personal Transit Pre-trip Planners. Transportation Research Procedia 5, 30–43.

PRESTON, J., PRITCHARD, J., WATERSON, B., (2017). Train Overcrowding. Transportation Research Record: Journal of the Transportation Research Board 2649, 1–8.

QIN, F., (2014). Investigating the In-Vehicle Crowding Cost Functions for Public Transit Modes. Mathematical Problems in Engineering 2014, 1–13.

RAJAPAKSHA, P., FARAHBAKHSH, R., NATHANAIL, E., CRESPI, N., (2017). ITrip, a framework to enhance urban mobility by leveraging various data sources. Transportation Research Procedia 24, 113–122.

RAVEAU, S., GUO, Z., MUÑOZ, J.C., WILSON, N.H.M., (2014). A behavioural comparison of route choice on metro networks: Time, transfers, crowding, topology and socio-demographics. Transportation Research Part A: Policy and Practice 66, 185–195. TIRACHINI, A., HURTUBIA, R., DEKKER, T., DAZIANO, R.A., (2017). Estimation of crowding discomfort in public transport: Results from Santiago de Chile. Transportation Research Part A: Policy and Practice 103, 311–326.

VOLOSHIN, V.I., (2009). Introduction to Graph and Hypergraph Theory. Nova Science Publishers, New York.

WHELAN, G.A., CROCKETT, J., (2009). An Investigation of the Willingness to Pay to Reduce Rail Overcrowding, in: First International Conference on Choice Modelling. Harrogate, England, p. 16.

ZHANG, Y., JENELIUS, E., KOTTENHOFF, K., (2017). Impact of real-time crowding information: a Stockholm metro pilot study. Public Transport 9, 483–499