SAPIENZA
UNIVERSITÀ DI ROMA

SAPIENZA UNIVERSITY OF ROME

MASTER THESIS

# Simulation of Congestion Phenomena on Transit Networks

*Author:*
Lory Michelle
BRESCIANI MIRISTICE
1699496

*Supervisor:*
Chiar.mo Professore
Guido GENTILE

Academic Year 2016/2017

*Per i miei genitori,*
*instancabili motivatori.*

Sapienza University of Rome

# *Abstract*

Department of Civil, Constructional and Environmental Engineering

Master in Transport System Engineering

**Simulation of Congestion Phenomena on Transit Networks**

by Lory Michelle
BRESCIANI MIRISTICE
1699496

This thesis proposes a deterministic static assignment model which can compute UE for large-scale networks in case of passengers mingling at stops and high-frequency or low-reliability transit service, considering also congestion phenomena.
More precisely, the model represents the following phenomena are represented by introducing additional costs (in terms of time) for the affected trip legs: 1) *Overcrowding Congestion* (i.e. passengers discomfort due to overcrowding on-board of a vehicle or at platforms); 2) *Queuing congestion* (i.e. passengers queuing at platforms); 3)*Dwelling Delay* (i.e. passengers waiting on-board of a vehicle, due to limited door capacity and overcrowding).
In addition, the model represents other two congestion phenomena: the *availability of seats* (both for boarding passengers and dwelling passengers) and the *waiting process* at stops. This is achieved by using the concept of strategies and hyperpaths, i.e. plans adopted by the passengers to reach the desired destination at a minimum expected cost.

As the congestion phenomena in transit networks are non-separable (i.e. cost of an arc depends also on the flows of other adjacent arcs), the uniqueness of equilibrium is not guaranteed and the UE cannot be found through convex optimization.
Thus, this thesis model adopts an assignment algorithm which solves the *fixed-point formulation* of UE, based on the circular dependency between arc flows and congested costs, through the following iterative process: 1) computing costs depending on flows; 2) updating users choice according to the shortest hypertree computed though an extension of Dijkstra algorithm; 3) finding the new distribution of flows; 4) using the new distribution of flow to compute the new costs. The iterative process is repeated until UE is found (i.e. flows resulting in the previous iteration converge with the ones computed in the current iteration).
As UE assignment function is not a contraction, to find UE convergence an innovative Reduce Gradient Projection method which performs convergence search over arcs, avoiding the enumeration of hyperarcs (i.e. implicit hyperarcs), was implemented. Relative gap criterion is adopted as stop condition.

The model was implemented in Visual Basic 15 and tested by using text files and the software PTV Visum 17 as I/O source. During the various tests, it was shown that the algorithm converges quickly and with high precision when the single congestion phenomenon is studied, while it takes more than 100 iterations where these phenomena are combined or more than on line is present. However, solution time for medium-size networks is still in the order of seconds.

Finally, the model implemented in this thesis performs static assignment and, thus, it is not suitable for real-time simulation and incident management. However, it can be seen as an early stage for the implementation of a real-time incident management software for public transport networks based on the *Transit Link Transmission Model* (TLTM) proposed by Gentile (2017), which is a fast-macroscopic model that performs a dynamic assignment from the results of a static assignment.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| **BPR** | **B**ureau **P**ublic **R**oads |
| **CSV** | **C**omma **S**eparated **V**alue |
| **IMP** | **IMP**edance |
| **FIFO** | **F**irst **I**n **F**irst **O**ut |
| **FB** | **F**requency-**B**ased |
| **GIS** | **G**eographic **I**nformation **S**ystem |
| **GLTM** | **G**eneral **L**ink **T**ransmission **M**odel |
| **GP** | **G**radient **P**rojection |
| **HB** | **H**eadway-**B**ased |
| **MSA** | **M**etod **S**uccessive **A**varage |
| **NLM** | **N**etwork **L**oading **M**ap |
| **OD** | **O**rigin **D**estination |
| **PJT** | **P**erceived **J**ourney **T**ime |
| **PTV** | **P**lanung **T**ransport **V**erkehr |
| **RCM** | **R**oute **C**hoice **M**odel |
| **RGAP** | **R**educed **GAP** |
| **RGP** | **R**educed **G**radient **P**rojection |
| **TB** | **T**-**B**ased |
| **TDE** | **T**ransportation **D**ata **E**xchange |
| **TLTM** | **T**ransit **Link Transmission Model** |
| **TSB** | **T**ransport **S**ystem-**B**ased procedure |
| **UDA** | **U**ser **D**efined **A**ttributes |
| **UE** | **U**ser **E**quilibrium |
| **VB** | **V**isual **Basic** |
| **WHO** | **W**orld **H**ealth **O**rganization |

# 1 Introduction

According to the WHO/Europe (World Health Organization), European countries are facing conflicting demands for transport policies. On the one hand, transport has a positive key role in the economy, on the other hand, its policies can harm human health and the environment.
In the European region alone, about 100 000 premature adult deaths occur each year due to air pollution, which is mostly caused by emissions from road traffic. In addition to these, road accidents result in about 127 000 deaths and 2.4 million injuries per year, killing more young people aged 5-29 than any other causes. Moreover, transport is the fastest growing source of fossil-fuel $CO_2$ emissions, which is one of the main factors of climate change. Finally, traffic noise and congestion damage health, psychological adjustment, work performances and overall life satisfaction (Dora and Phillips, 2000).

Therefore, one of the greatest challenges cities are facing is the creation of sustainable transportation solutions, where private transport gives way to different means of public transport, which are more sustainable and have a larger capacity.
Governments have been investing huge capitals on public transport, attempting to attract more users by increasing the quality of service and comfort. Nevertheless, this sustainable transport mode must still compete with private transport, as users' experience is influenced negatively by congestion and service reliability.

Indeed, passengers in many countries around the world experience congestion problems on public transport (e.g. buses, underground and trains), which make travelling during peak hours a stressful experience.
Over saturation of vehicles makes access/egress a difficult operation, causing significant delays, and the stress of overcrowding on public transport reduces people's productivity at work. For instance, in London transit network, crowding and delays cost the population around £230 million per year (Oxford Economic Forecasting, 2003).
Overcrowding also has consequences in safety and risk on platforms. For instance, in Rome, the main subway operator (ATAC) sometimes blocks access to escalators and stairs, to allow platforms to be cleared of passengers who failed to board overcrowded trains.

Hence, sustainable urban mobility requires the increase in capacity to compete with private transport, reducing congestion. This can be done through careful planning of the public transport service, where decision-makers need to answer in a rational and transparent way the following key question: "*How big is the total benefit of a proposed investment?*" (Gentile and Noekel, 2016).
To answer this question, different transit assignment models have been introduced in the

last 60 years.

Finally, congestion and overcrowding are not only due to wrong strategic and tactical planning but also to the lack of operational and real-time planning. And yet, a system which can predict the network response to incidents and propose a solution to keep congestion levels acceptable does not yet exist.
For example, it may happen that a bus breaks down and a passenger needs to decide whether to wait for the next bus or to alter their route altogether. In both cases, the incident causes a redistribution of passengers' flows and may cause a congestion increase either on the malfunctioning bus line or to the rest of network, causing overcrowding and delays.
To contain congestion and keep the level of service high, public transport operators should use a system able to predict congestion development and guide users' choices. However, a software fast enough to forecast real-time passengers' congestion in public transport networks while considering a wide range of congestion phenomena, does not exist yet.

This thesis focuses on the simulation of transit networks, including passengers' congestion phenomena, as a tool for strategic and tactical planning. More precisely, a model able to consider the following peculiarities of transit assignment was implemented and tested: a) Transit vehicles have finite capacity and demand may exceed this capacity during peak-hours; b) when timetables are not published, passengers consider multiple routes and choose among them only when the first vehicle arrives; c) passengers' choices also depend on the seat availability of the vehicle.
Simulations were executed implementing the model in Visual Basic 15 (VB "14"), and results were visualized with the software PTV Visum 17 (Visum) of the company *PTV (Planung Transport Verkehr) Group*, which allows GIS-based data management in the field of private and public transport. To simplify the input process, data regarding networks and demand were imported from Visum.
Moreover, to validate simulation results, comparisons between results obtained by the implemented software and Visum transit assignment feature were made.

This thesis can also be seen as an early stage for the implementation of a real-time incident management software for public transport networks. Indeed, outcomes of the implemented model can be used as inputs of the Transit Link Transmission Model (TLTM) introduced by Gentile (2017), which is a fast model able to propagate flows affected by congestion on any network for given route choices.

This paper is organized as follows. Chapter 2 reviews some of the existing approaches to transit assignment modelling and Visum approach, while chapter 3 illustrates the theoretical background of the implemented model. Chapter 4 describes how the model was implemented in using a software, and finally, chapter 5 contains some numerical tests, key findings and future research plans.

# 2 Existing Approaches to Transit Assignment

The main aim of transit assignment is to find the set of routes which will be chosen by users according to the demand and supply of the transit network.
Existing algorithms are based on the so-called *Wardrop's first principle*, which states that, at network equilibrium, the journey time of routes that are used is less or equal to those that would be experienced by a single user taking any of the unused routes (Wardrop and Whitehead, 1952). According to this principle, the so-called *user equilibrium* (UE) is found when no user finds convenient to unilaterally change the path.
Assumptions beyond this principle are that users are rational decision makers and have perfect knowledge of journey cost, which often implies some unrealistic results such as aggregation of choices (i.e. users going from the same node to the same destination will have the tendency to take the same path, even if they come from different origins).
Even if the UE principle results in sub-optimal system performance, it is used by the majority of transit assignment approaches as it represents how individual users realistically behave, with some limitations due to the assumptions introduced before.

Since Dial (1967) presented one of the first transit assignments, different models of public transport networks and transit services have been proposed. They can be usually divided into two main categories: *frequency-based (FB) models* and *schedule-based (SB) models*.

The so-called schedule-based (SB) models are adopted to simulate low-frequency or high-reliability services, which usually have published timetables. These models introduce a diachronic graph to represents every single run of the service, making the models inherently dynamic. Indeed, the assignment results (travel times and passenger flows of transit lines) are always referred to a specific run and to a certain time of the day.
SB models can represent in details passengers' crowding due to vehicle capacities but are not suitable to simulate the delays of vehicles due to alighting and boarding passengers at stops, as the diachronic graph must vary with the flow pattern.
Moreover, this framework cannot be used to model high-frequency services. Indeed, SB models assume that users would choose specific runs and try to synchronize their arrival at stops with the scheduled passage of carries. This is not true for densely connected urban networks, where passengers usually do not consider timetables as they perceive runs of the same transit line as a unitary supply facility.

To simulate high-frequency or low-reliability service, frequency-based (FB) models are usually preferred. These models can simulate services where vehicles are not able to follow timetables and the service is perceived by passengers in terms of probabilistic departure events.
However, also the frequency-based approach is insufficient to simulate all the passengers' congestion phenomena. Indeed, frequency-based models simulate the behaviour of public transport lines without considering single runs and, consequently, they can simulate vehicles delays but cannot consider vehicle capacity constraints.

Another differentiation that can be done among models regards the assignment of passengers' priority in the stop waiting process. In general, when passenger congestion occurs, the queuing protocol followed by travellers is determined by the stop layout. Usually, passengers do mingling in stations and stops with large platforms, while on urban bus networks they tend to respect the boarding priority of anybody who has arrived before them.

The model adopted in this thesis is based on FB approach with passengers mingling at stops. Hereafter, previous works carried out to solve the transit assignment with FB approaches are reviewed, and the transit assignment algorithm used by Visum is illustrated in detail.

## 2.1 Overview of Frequency Based Models

This section overviews the state of the art of the FB UE transit assignment framework and models, starting from a brief description of the early models.

As introduced before, the development of transit assignment models can be traced back to a contribution by Dial (1967). The modeller proposed an adaptation to public transport of the already existing private transport shortest path algorithm. This new algorithm was able to include in the computation of costs the waiting times of passengers at stops, which is the main phenomenon that distinguishes public transport from private one, but it neglected congestion.
De Cea and Fernandez (1989) developed the concept introduced by Dial into an efficient algorithm for large networks.

Chriquì and Robillard (1975) introduced the *common line dilemma*. This phenomenon happens when users going from an origin to a destination can choose among a set of lines, that can have different travel times. In this case, it was observed that passengers apply strategies to choose between boarding the first line approaching the stop or keep waiting for a more convenient service. Chiriquí and Robillard also suggested a heuristic algorithm to find the set of routes that minimize the expected travel time.

To solve the common line dilemma, Nguyen and Pallottino (1988) stated that strategies (or *hyperpaths*) are chosen pre-trip by users, and that the realization of the same travel strategy may change due to events at the stop (which vehicle arrives first). They also proposed a

heuristic method to assign passengers to lines according to the nominal frequencies.
Spiess and Florian (1989) showed that passengers can often significantly reduce their travel time when they consider several paths to their destination, and Cominetti and Correa (2001) used the concept of hyperpath to adapt Dijkstras shortest-path algorithm to a *shortest hyperpath algorithm*.

Spiess and Florian (1989) discussed the extension of their model to the case where link costs are depending on link flows, which often happens in transit networks (e.g. dwelling time at stops depends on passengers boarding and alighting). Their model, however, had three main limitations: 1) cost-function must be continuously and monotonously increasing; 2) all passengers on board the vehicle suffers the same inconvenience, independently of where they boarded; 3) service frequencies are assumed to not be affected by crowding, so that waiting time at stops is only a function of nominal frequencies.
Spiess and Florian suggested that these limitations could be overcome with the *effective frequency approach*, namely the line frequency perceived by the waiting passengers. The idea behind the effective frequency is that the more buses arrive full, the higher the waiting time will be as it will be harder to get onto the vehicle.

De Cea and Fernandez (1993) introduced a model that applied the effective frequency concept. They also proved that the set of attractive lines monotonically increases with congestion and that the set of attractive lines in uncongested situations is a subset of the attractive lines in congested situations. However, in their model, De Cea and Fernández did not consider that a high number of passengers wishing to board will reduce the chance for an individual to board.
According to Cominetti and Correa (2001), other shortcomings of De Cea and Fernández's model were: 1) the functional form used to represent congestion was only justified heuristically and caused an overload of lines; 2) a heuristic method was used to compute common lines between transfer nodes, not guaranteeing the fulfillment of Wardrops first principle (in some congestion situations, cost can be minimized by combining two strategies).

An example for the second shortcoming of De Cea and Fernández's model is the following. Given a network with one OD pair and two direct lines $l_1$ and $l_2$, with $l_1$ faster than $l_2$, passengers can choose between the two following strategies: 1) wait for $l_1$; 2) take whichever line arrives first (waiting for only $l_2$ is not an option as it is more "*expensive*"). In De Cea and Fernandez (1993)'s model assigned all the passengers to strategy number two, but Cominetti and Correa (2001) shown that if in highly congested situations some passengers stick to Strategy 1 as this would reduce overall travel time.

More recent models have added rigour to the analysis and efficiency of the above-introduced algorithms, also addressing the issue of congestion in transit networks and its consequences on discomfort and queuing (e.g. overcrowding discomfort on board, queuing of passengers at stops, provision of information at stops, etc.).
Nowadays, most of the existing FB models include the concept of hyperpaths and assume that passengers mingle at stops (e.g. Leurent et al., 2012). This implies that no priority is satisfied in the waiting process, and, in case of over-saturation, all passengers have the same

probability to board the next attractive arriving carrier.
Alternatively, when assuming FIFO queues in FB models the problem becomes more challenging. In this case, the stability condition usually adopted to find equilibrium is that passengers waiting at a stop consider an attractive set that is never completely saturated (Bouzaïene-Ayari et al., 1998). As congestion increases, more and *"worse"* lines are included in the attractive set and, when all lines are congested, passengers decide to walk even if the extra waiting time due to congestion is short (Trozzi et al., 2013).

The state of the art in transit assignment is summarized in Gentile and Noekel (2016), which provides a collection of recent transit assignment frameworks and models.

The model implemented in this thesis derives from the transit assignment approach introduced by Gentile et al. 2016b. The new approach integrates passengers' congestion in UE modelling framework introducing some coefficients in the disutility function to represent crowding discomfort. Further details will be discussed in chapter 3.

## 2.2 Visum Transit Assignment

This section is a resume of the transit UE assignment algorithm used by Visum. All the section is referenced to PTV Visum 17 Manual (PTV AG, 2017).

Visum provides three types of UE assignment for the public transport case, called *procedures*:

- **Transport system-based (TSB) procedure.** It is based on *"all or nothing"* assignment and provides an approximate *"ideal line network"*, where passengers choose the fastest route without considering any existing line network constraints. It is usually used in the initial planning phase when line service is not yet existing.
- **Headway-Based (HB) procedure.** It is a frequency-based framework, which is ideal for urban networks with high service frequency and low reliability. It is usually used for long-term conceptual planning when timetables are still unknown.
- **Timetable-based (TB) procedure.** It is a schedule-based framework, which is ideal for networks with long headways services and where timetable coordination is important for transfers.

As the purpose of this section is introducing Visum assignment methods to compare them with the model implemented in this thesis, only the FB approach will be analysed.

### 2.2.1 Headway-Based Procedure

For the HB procedure, each line is described by the line route, the run times between line stops, and the headway. In the following sections, the term line is used for the sake of convenience.

The HB assignment procedure of Visum determines the optimum routes and those that are close to optimum, and follows three operational steps:

1. headway calculation;
2. route search and route choice;
3. route loading.

In these procedures, transfer wait times are analysed globally and departures of different lines are independent of each other.
This means that in this type of assignment a timetable coordination is not possible and that results, such as the number of transfers, journey time and ride time, can be estimated with sufficient accuracy only when all lines have short headways.
On the other hand, this procedure is not suited for transit network planning hence long headways occur (i.e. urban area or long-distance transport) as usually providing connections is important.

### Headway Calculation

The headway $\tau$ of a line for HB assignment can be defined in three different ways:

1. as a line attribute (in this case timetables are completely ignored);
2. as the mean headway $\tilde{\tau}^{a,b}$ of each time interval $t = [t_a, t_b)$ in the assignment time interval, according to the number of departures $n$ of the timetable ($\tilde{\tau}^{a,b} = \frac{t_b - t_a}{n}$);
3. as the mean wait time $\tilde{\delta}^{a,b}$ according to the timetable and the set of departures $x_i$ in the time interval $[t_a, t_b)$ ($\tilde{\delta^{a,b}} = \frac{1}{t_b - t_a} \sum_{i=0}^{n} \Delta_i$ [1]).

Each of the three methods can be applied separately by time interval, which means the Visum user can model supplies that vary within the assignment period and do a semi-dynamic assignment (i.e. dividing the study period in several time intervals and computing static network equilibrium for each of them, while also considering the flow propagation between periods).

The second approach is a simplified approximation of the third one. This approximation is acceptable only in case of networks with short headways and sufficiently broad time intervals. Indeed, two main problematics are connected to this approach: 1) the definition is too sensitive to the shifting of individual departures across the interval limits, causing a discontinuity in the results; 2) for passengers that arrive randomly, trips spread evenly throughout the time interval, causing less waiting time than trips that are piled up.
The third approach solves these shortcomings defining the headway of a line in case of random access as double the expected waiting time for the next departure.

The following example illustrates the difference in approximation of the two approaches, considering two lines with the timetables illustrated in table 2.1.

As it can be seen from table 2.1, the distance between departures is the 40 minutes for both lines. Consequently, the headways computed should be the same.

[1] $\Delta_0 = (x_1 - a)^2$, $\Delta_i = (x_{i+1} - x_i)^2$, $\Delta_n = (x_{n+1} - x_n)^2 - (x_{n+1} - x_b)^2$

TABLE 2.1: Example for headway calculation, input timetables

| Line | Departure Times |
|---|---|
| 1 | 05:55, 06:35, 07:15 |
| 2 | 06:05, 06:45, 07:25 |

Applying method 2 to compute the headway as the mean headway of time interval $t = [06:00, 07:00)$, results differ from each other, as they are sensitive to the shifting of departures across the interval limits. Indeed, using formula 2.1 and 2.2 and considering that the number of departures in interval $t$ are respectively 1 and 2 for line 1 and line 2, headway of line 1 results twice the one of line 2.
On the other hand, applying method 3 for the same interval $t$ and lines and computing the headway as mean of the waiting time in the time interval (equation 2.3 and 2.4), results are the same.
Overall results are shown in table 2.2.

TABLE 2.2: Example for headway calculation, results

| Line | Headway Method 2 [min] | Headway Method 3 [min] |
|---|---|---|
| 1 | 60 | 43.33 |
| 2 | 30 | 43.33 |

$$\tilde{\tau}_1^t = \frac{t_b - t_a}{n} = \frac{07{:}00 - 06{:}00}{1} = 1h. \tag{2.1}$$

$$\tilde{\tau}_2^t = \frac{t_b - t_a}{n} = \frac{07{:}00 - 06{:}00}{2} = 30min. \tag{2.2}$$

$$\begin{aligned}\tilde{\delta}_1^t &= \frac{1}{t_b - t_a} \sum_{i=0}^{1} \Delta_i = \\ &= \frac{1}{07{:}00 - 06{:}00} \cdot \left[(06{:}35 - 6{:}00)^2 + ((07{:}15 - 6{:}35)^2 - (7{:}15 - 7{:}00)^2)\right] = 43.33min.\end{aligned} \tag{2.3}$$

$$\begin{aligned}\tilde{\delta}_1^t &= \frac{1}{t_b - t_a} \sum_{i=0}^{2} \Delta_i = \\ &= \frac{1}{07{:}00 - 06{:}00} \cdot \left[(06{:}05 - 6{:}00)^2 + (06{:}45 - 06{:}05)^2 + ((07{:}25 - 6{:}45)^2 - (7{:}25 - 7{:}00)^2)\right] + \\ &\quad + ((07{:}25 - 6{:}45)^2 - (7{:}25 - 7{:}00)^2)\,] = 43.33min.\end{aligned} \tag{2.4}$$

### Route Search and Route Choice

In this step, paths are assessed minimizing their *generalized cost*, computed through the so-called impedance functions. Headway-based assignment procedures in Visum does not take into account capacity constraints.

#### *Impedance function computation*

Impedance functions represent the disutility of taking a specific path. Impedance is composed of times and fares and assigns malus or bonus to specific connection properties through

a combination of user-defined skims. Generally, the lower the impedance of a connection is, the higher its share of transport demand.
The general form of impedance function is represented by the following equation:

$$IMP = PJT \cdot \alpha_{PJT} + n_f \cdot \alpha_f, \tag{2.5}$$

where $PJT$ is the perceived journey time, $\alpha_{PJT}$ is its weight factor, $n_f$ is the number of fare points (or fares) and $alpha_f$ is the fare-based component weight.

The perceived journey time $PJT$, used in equation 2.5, is computed in minutes with the following deterministic formula (journey times, cost, etc. are deterministic):

$$\begin{aligned} PJT\,[MIN] \quad = \quad & t_{IVT} \cdot \alpha_{IVT} \cdot \beta_l + t_{AXT} \cdot \alpha_{AXT} + \\ & + t_{ACT} \cdot \alpha_{ACT} + t_{EGT} \cdot \alpha_{EGT} + t_{WT} \cdot \alpha_{WT} + t_{OWT} \cdot \alpha_{OWT} + \\ & + t_{TWT} \cdot \alpha_{TWT} \cdot \beta_s + NTR \cdot \alpha_{NTR} + p_b + p_{b_{AUX}} + \tilde{d}. \end{aligned} \tag{2.6}$$

In equation 2.6, the first seven factors depend directly on times. More precisely, $t_{IVT}$ is the in-vehicle time, $t_{AXT}$ the aux-ride time, $t_{ACT}$ the access time, $t_{EGT}$ the egress time, $t_{WT}$ the transfer walk time, $t_{OWT}$ the origin wait time, and finally $t_{TWT}$ is the transfer wait time. Factors $\alpha$ are the corresponding general penalty factors, while $\beta_l$ and $\beta_s$ represent weights associated to a specific line $l$ (to model the vol/cap ratio or other aspect of usability) and stop $s$ (to model the users' preference for some stops, e.g. stop close to shops).
Origin and transfer wait time result from the headway of lines and they depend randomly on the relative position of the transfer lines.
The last four terms of the equation 2.6 represent additional penalties. More precisely, $NTR$ and $\alpha_{NTR}$ are the number of transfers and its weight factor, $p_b$ and $p_{b_{AUX}}$are the boarding penalties associated respectively to line and auxiliary transport system [2] and $\tilde{d}$ is the mean delay of the line.

For what concerns the non-temporal component of equation 2.5, the variable $n_f$ represents either the total of all fare points that are traversed along the route or a fare derived from the Visum fare model. Of course, fare must be applied to each path leg separately, so that, for instance, each boarding passenger has to purchase a new ticket.

Passengers make choices in different situations:

1. they are at a stop and must choose which line to board;
2. they are starting their journey (origin zone) and must choose where to board;
3. they are on board of a line and have to choose when to alight;
4. they are choosing between transfer stops, which may be reachable by a foot-path.

These choices can be based on observations or on estimations, depending on the situation.

Generally, Visum models passengers' decisions as a sequence of separate decisions, each of them based either on estimations (discrete choice model) or observations (model 1, 2, 3 and

[2]Auxiliary transport system is e.g. public transport plus car for park and ride

4), as introduced before. More precisely, the result of the decision made on a lower level becomes part of the decision on a higher level, in form of expected remaining travel time (hierarchical structure).

Hereafter, models adopted for these decisions are illustrated.

***Choice models in case of observations***

In this section, choice models for choices based on observations are described. This is the case of boarding at stops (situation 1), or situation 2, 3 and 4 when a suitable infrastructure (e.g. passenger's knowledge of timetables and on-board real-time information system which provides information about departures of other lines at the stop or at close-by stops) is available.
Indeed, FB models usually assume that passengers know line headways and running times, and, thus, in situations 2, 3 and 4, costs can only be estimated when full information about next departures of lines from stops is not available.

Visum offers four different models for decisions based on observations:

1. no information and exponentially distributed headways (passengers face a high level of uncertainty);
2. no information and constant headways;
3. information on the elapsed wait time and constant headways;
4. information on the next departure times of the lines from stops (e.g. passenger information systems at stops);

All models are suitable for boarding decisions (situation 1), but only model 4 can be applied to the other cases (situation 2, 3 and 4). In case of boarding, even if the four models assumed different levels of information, in the end the passenger always chooses one of the different lines, due to observations (e.g. arriving vehicles).
Hereafter, models are explained referring to situation 1, boarding decisions.

Being $L$ the set of available lines, sorted in ascending order according to their remaining journey time $s_l \geq 0$ (expected time needed from the studied node to reach the destination node), the different choice models calculate the optimal set of lines $L^* \subseteq L$, and the demand share $\pi_l \geq 0$ of each line $l \in L^*$. Of course, this computation changes according to the available information.
When users choose any set of lines $L'$, they want to minimize their remaining costs $C_L$, which is a function of the remaining journey time $s$ and the wait time before boarding $W_{L'}$, as shown in the following equation:

$$C_{L'} = W_{L'} + \sum_{l \in L'} (\pi_l \cdot s_l)\,. \tag{2.7}$$

The remaining journey time $s$ is computed through the impedance function explained in equation 2.5.

When no extra information is known and headways are distributed exponentially (model 1), the optimization can be done with two different approaches: 1) using the model introduced by Spiess and Florian (1989) to minimize times, assigning passengers to lines proportionally to the nominal frequencies, without considering explicitly the increase in waiting times induced by congestion; 2) determining the optimal set of lines, and then choosing the first arriving line in this set.
The first approach allows only additive fares and simple coordination of transfers, while the second allows additional choice model settings and more complex definitions in terms of coordination and fares.

Adopting the second approach, the optimal set of lines is computed considering that the expected remaining travel time of a set $L_i$ is given by equation 2.8, where $\lambda_j$ and $s_j$ are respectively line $j \in L_i$ frequency (inverse of headway) and remaining travel time, and that the optimal set of lines $L_{i^*}$ is the one which minimizes $u_{i^*}$. It is proved that the optimal set is $L_{i^*}$, where $i^*$ is the last line in the optimal set [3] and for which equation 2.9 is valid. Equation 2.9 ensures the stop condition: $E(C_{L_i}) > E(C_{L_{i-1}})$.

$$u_i = \frac{1 + \sum_{j=1}^{i} (\lambda_j \cdot s_j)}{\sum_{j=1}^{i} \lambda_j}. \tag{2.8}$$

$$i^* = max\left\{ i : s_i \le u_i - 1 \right\}. \tag{2.9}$$

The share $\pi_i$ of line $i \in L_{i*}$ is equal to the probability that line $i$ depart first and can be computed as function of the frequency as in equation 2.10. This non dependence from remaining travel times in the share definition illustrates the heavily simplified construction of this choice model.

$$\pi_i = \frac{\lambda_i}{\sum_{j \in L_{i*}} \lambda_j}. \tag{2.10}$$

This construction of share $\pi_i$ shows how this choice model is heavily simplified.

In case that no information is available to the passengers and service is characterized by constant headways (model 2), the route choice model follows the same principles of the previous one, where users select the first line arriving from the optimal set. However, the approach to compute the first line arriving is different, as in this case condition 2.9 does not ensure optimality. Indeed, in this case the stop condition $E(C_{L_i}) > E(C_{L_{i-1}})$ is no longer sufficient, when headways are constant as some local optima may occur.
The stop condition adopted in this approach is the following:

$$i^* = argmin_i \left\{ E(C_{L_i}) \right\}. \tag{2.11}$$

[3] Lines are sorted by ascending $s$, as explained before

The shares assigned to the individual lines again correspond with the possibility of arriving first, as shown in the following equation:

$$\pi_i = \lambda_i \cdot \int_0^{\overline{h}} \prod_{j \in L^*, j \neq i} (1 - \lambda_i \cdot w)\, dw, \tag{2.12}$$

where $\overline{h} = min \{h_i\}$ is the minimal occurring headway.
If the timetable in the analyzed network is regular or only slightly irregular, and the passengers do not have any information on departure times, this choice model is more realistic than the model considered before.

When information on elapsed waiting time is available and headways are constant (model 3), passengers can use the information of how long they have been waiting to reduce their expected remaining costs. Indeed, knowing the elapsed time a passenger can ignore potentially earlier arriving time if they are slower than the remaining journey cost of another line. In this case, the optimal set $L^*$ depends on the elapsed wait time and it is no longer constant, which makes determining it with respect to the previous models more difficult.
It is proven that there is an exact point in time $t_j \in I_j = \left(t_j + 1, t_j\right]$ from which onward the remaining journey time of line $j$ is greater than the remaining cost $C_{L_{j-1}}$ of line $l \in L_{j-1}$, which means that $t_j$ is the unique solution of equation 2.13. If the users observe an arrival of a line from $\tau \in I_j$ after wait time $W$, they will board the line, while other lines will be ignored.

$$s_j = E(C_{L_{j-1}} |_{W>t})t. \tag{2.13}$$

As normally a passenger has the information about elapsed time, this model does not imply any strong options. However, the computation of optimal set of lines has to be redone for each instant of time and, thus, determining the optimal set is more difficult than in the previous models.

Finally, when information on departure times is available (model 4) it is not necessary for the user to know the times and headways of all the lines and the optimal strategy can be formulated as follows: "*A passenger boards the line that offers the least remaining costs given the actual departure times*".
As nowadays many places already have information systems at stops which display the next departure times on the basis of real-time operating data, the assumption of passenger knowledge of departure times is not an extremely strict requirement.

Thus, the optimal line set consists of all lines which have the least costs in some timetable positions, as shown in the following equation:

$$i^* = max \left\{i : s_i < min_j \left\{s_j + h_j\right\}\right\}. \tag{2.14}$$

In this case, the calculation of shares is as follows:

$$
\begin{aligned}
\pi_i &= P\left(C_i < C_j\ \forall j\right) = \\
&= \frac{1}{h_i}\int_{s_i}^{s_i+h_i} P\left(C_i < C_j\ \forall j | C_i = x\right) dx = \\
&= \frac{1}{h_i}\int_{s_i}^{s_i+h_i} \prod_{j\neq i} P\left(C_i < C_j|_{C_i=x}\right) dx \\
&= \frac{1}{h_i}\int_{s_i}^{s_i+h_i} \prod_{j\neq i} P\left(C_j > x\right) dx = \\
&= \frac{1}{h_i}\sum_{k=i}^{i^*}\int_{s_k}^{s_{k+1}} \prod_{j=1,\, j\neq i}^{k} P\left(C_j > x\right) dx = \\
&= \frac{1}{h_i}\sum_{k=i}^{i^*}\int_{s_k}^{s_{k+1}} \prod_{j=1,\, j\neq i}^{k} \left(1 - \frac{x - s_j}{h_j}\right) dx.
\end{aligned}
\tag{2.15}
$$

The demonstration of derivation of equation 2.15 is not the subject of this thesis. For the sake of knowledge, it can be found in PTV Visum 17 Manual PTV AG (2017), section 7.9.4.4.
For what concerns situations beside boarding choice where this model can be applied, the model will consider choosing between all the possible transfer lines, which are available after alighting, and the on-board line (also at the beginning of users' trip, at origin zone).

***Choice models in case of estimations***

In this section, choice models for choices based on estimations are described. This is the case of situations 2, 3 and 4 when no additional information beside travel times and headways are available, i.e. when information systems or timetables are not provided.
Such decisions can be modelled in two ways:

1. by a discrete choice model;
2. by an *"all-or-nothing"* decision in favour of the best alternative.

The second case reduces the expected remaining costs but does not reflect the fuzziness of the passengers' behaviour. Thus, normally a discrete choice model is better.

**Route Loading**

To search routes, first of all, the travel demand of an OD pair is entered at the origin zone. According to the choice model adopted, Visum assignment procedure divides (as is the case at all later decision points) the entire demand between all reasonable alternatives. Several alternatives having different headways and impedance may be already available with the choice of the first line.
When splitting demand, stochastic fuzziness becomes involved as all the used lines possess a headway and, thus, the wait time for a line is random. Consequently, a certain percentage of demand can be given to a line which is less attractive (e.g. when passenger information on departures is available, it may happen that the line with positive probability departs so much earlier than qualitatively better alternatives, and this time advantage wins over the higher impedance).

To represent this phenomenon, the route search in the headway-based assignment is not based on shortest path search but creates a directed decision graph for each destination zone, where nodes represent the decision points (i.e. stops with several alternatives) and paths represent the various options to reach the destination zone.
A fundamental assumption of this approach is the Markov's memory-less property of a stochastic process (Markov, 1954). This assumption means that, from each stop, passengers will make their choice for the continuation of their journey on the basis of this probability graph, regardless of how they reached this stop. Consequently, search and choice in the headway-based procedure are organized so that, working backwards from each destination zone, all options are calculated to allow passengers to move from the stops of the network towards the destination zone. Thus, the mean impedance of already-analysed decision points is used for the iterative calculation of the distribution for more distant decision points.

In the course of this search, only routes that are positively assessed by the selected choice model are maintained and then loaded in the decision graph.

**Example for the Headway-Based Assignment**

This section illustrates an example of Visum headway-based assignment. More precisely, user equilibrium in a two-hour time interval is computed for the public transport supply represented in figure 2.1.

The network is composed of 22 different links and two public transport services are operating on them. The first service, called *Bus 1*, is represented with a red line in figure 2.1, while the second, called *Train*, is represented with a white-and-black-striped line. Service timetables for the time interval between 05:30 and 07:30 are listed in table 2.3.

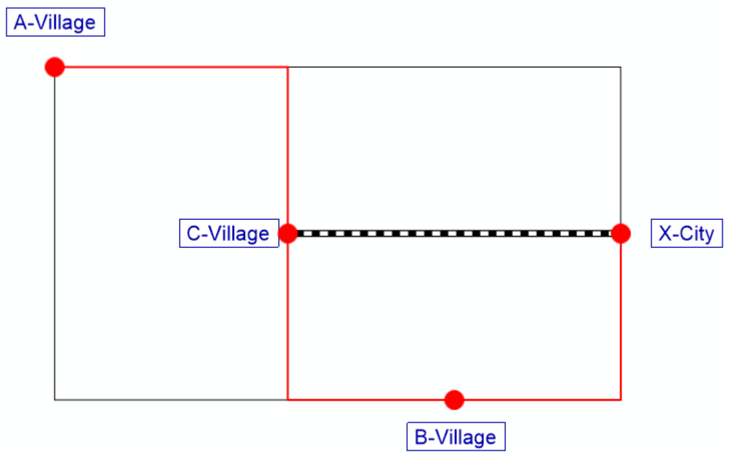

FIGURE 2.1: Example of headway-based assignment, studied public transport supply.

First of all, headways can be calculated applying one of the methods described in section 2.2.1. In this case, the method of the mean headway is chosen and applied to the time interval above mentioned. From the formula of this method (model 2), and being the number of

TABLE 2.3: Example of headway-based assignment, input timetables.

| Line | Departure Times |
|---|---|
| *Bus* | 06:10, 06:55, 07:25 |
| *Train* | 06:25, 07:25 |

departures per time interval respectively 3 for *Bus 1* and 2 for *Train*, the resulting headways are 40 minutes and 60 minutes.

Secondly, route search has to be performed. In this example, the demand is composed of the only OD pair $(A - Village, X - City)$. Assuming the case in which information on departure times is available to passengers, both at stops and on board of *Bus 1* (complete information, the route search step finds two possible routes:

- Route 1: *Bus 1;*
- Route 2: *Train;*

where Route 2 becomes faster than of Route 1 only if no extremely high transfer time penalty is used. Indeed, besides the transfer time penalty, the mean wait time for the train (i.e. 30 minutes) and the complete range of possible wait times are used to choose the path.
In this case, passengers will wait for the train at the transfer stop between 0 and 60 minutes. With a certain probability, the train will arrive only shortly after the bus arrives. However, the probability of obtaining an unfavourable connection in higher than the favourable case, and the majority of the passengers will continue their journey by bus. Thus, thanks to the existing passenger information, each of the two routes receives that portion of demand which corresponds to the chance of being the better of the two options.

To compute this chance, equations 2.5 and 2.6 are used to find the impedance. In this example, the following specific impedance parameters are set:

- $\alpha_{PJT} = 1;$
- $\alpha_{f} = 0;$
- $\alpha_{IVT} = 1;$
- $\beta_{l} = 1;$
- $\alpha_{AXT} = n.d.$[4]$;$
- $\alpha_{ACT} = 1;$
- $\alpha_{EGT} = 1;$
- $\alpha_{WT} = 1;$
- $\alpha_{OWT} = 1;$
- $\alpha_{TWT} = 1;$
- $\beta_{s} = 1;$
- $\alpha_{NTR} = 2min.$

Resulting impedances for a passenger arriving at the railway station on Bus 1 for the remaining route legs, assuming a remaining run time of 33 minutes for *Bus 1* and 16 minutes for *Train*, are listed in table 2.4.

As said before, the decision of which of the two routes is more attractive depends on whether the random variable (impedance of *Train*, called *IMP2*) is greater or smaller than the constant variable (impedance of *Bus 1*, called *IMP1*). As the first (*IMP2*) is uniformly distributed in $rand\,[18, 78)$ minutes, and the second (*IMP1*) is equal to 33 minutes, the probability of choosing route 2 is 0.25 and can be computed through the following formula:

$$P_2 = \frac{IMP1 - min\,(IMP2)}{max\,(IMP2) - min\,(IMP2)} = \frac{33min - 18min}{78min - 18min} = \frac{15min}{60min} = 0.25. \tag{2.16}$$

[4]no auxiliary transport systems are available, thus the parameter setting is not irrelevant

TABLE 2.4: Example of headway-based assignment, resulting impedance calculation.

| | *Bus 1* [min] | *Train* [min] |
|---|---|---|
| Egress and walk time | 0 | 0 |
| Run time | 33 | 16 |
| Transfer wait time | 0 | *rand* $[0, 60)$ |
| Transfer time penalty | 0 | 2 |
| **IMP** | **33** | **rand** $[\mathbf{18}, \mathbf{78})$ |

For example, if the OD demand is 90 trips, 25% of 90 (≈23 passengers) will choose route 2, while the remaining 75% (≈67 passengers) will choose route 1, and the final passengers' volumes on the network are the ones shown in figure 2.2.

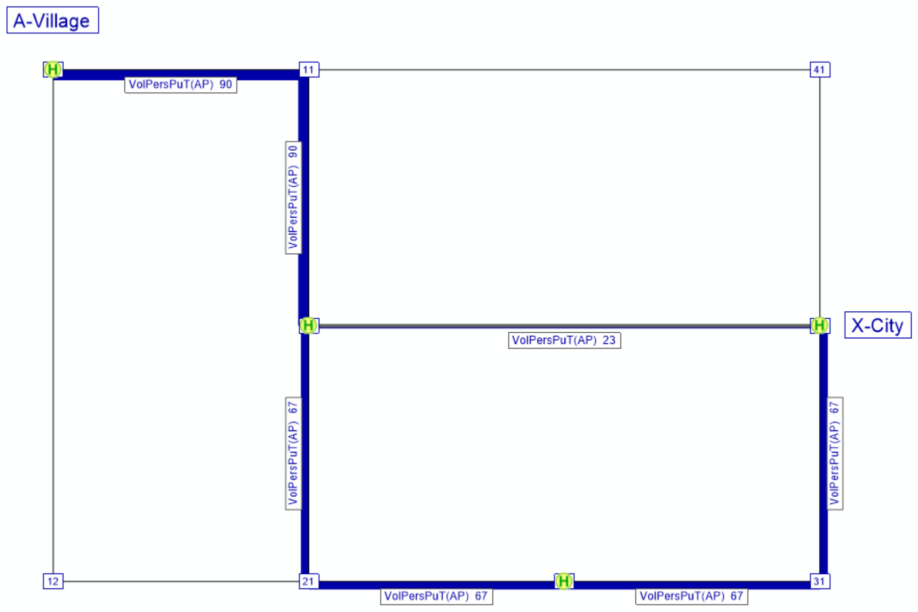

FIGURE 2.2: Example of headway-based assignment, distribution of flows.

With any variation of impedance parameters, the computed flow shares change. Table 2.5 shows changing in flow shares according to the variation of transfer penalty.

TABLE 2.5: Example for headway calculation, change of flow distribution according to transfer penalty.

| Transfer Time Penalty [min] | Share of Route 1 | Share of Route 2 |
|---|---|---|
| 0 | 0.717 | 0.283 |
| 1 | 0.733 | 0.267 |
| 2 | 0.750 | 0.250 |
| 5 | 0.800 | 0.200 |
| 10 | 0.883 | 0.117 |

# 3 Theoretical Background

In transit assignment, passengers' journey can be split into several legs, i.e. reaching the stop, accessing the platform, waiting for the vehicle, boarding the vehicle, travelling on-board of the vehicle, dwelling at stops, alighting the vehicle (also for transfers), etc..
The implemented model represents all these trip phases through a specific graph, which associates an arc and a related cost function to each journey leg.

Cost functions representing arc performances consider the costs strictly linked to the journey leg (such as travel time) and some additional costs, which represent the following congestion discomfort due to vehicle and platform capacity:

1. *Overcrowding Congestion*: passengers discomfort due to overcrowding on-board of a vehicle or at platforms;
2. *Queuing congestion* at platforms for waiting passenger;
3. *Dwelling Delay* for passenger on-board of a vehicle, due to overcrowding and door capacity.

In addition, the model can also represent other congestion phenomena such as the *availability of seats* (both for boarding passengers and dwelling passengers) and the *waiting process* at stops.
To model these last two congestion phenomena, it is assumed that passengers have a strategic behaviour, and the concept of strategies and hyperpaths, introduced by Chriquì and Robillard (1975) and Nguyen and Pallottino (1988), is adopted.

Finally, the model performs *deterministic congested transit assignment* through a UE algorithm which determines search direction and size of the convergence search through a *reduced gradient projection (RGP)* method on *implicit hyperarcs*. Relative gap criterion is adopted as stop condition.

As introduced in chapter 2, deterministic congested transit assignment finds the UE, which consists of the set of routes which will be chosen by users according to demand and supply of the transit network.
UE algorithms are based on Wardrop's first principle (Wardrop and Whitehead, 1952), which states that the journey times in all routes that are used are less or equal to those that would be experienced by a single user on any unused route, so that:

1. any used path $k \in K_{od}$ is shortest;
2. any non-shortest path $k \in K_{od}$ is unused.

This means that equilibrium is found when no user finds convenient to unilaterally change path.
The main assumption behind this principle is that users are rational decision makers and wishes to choose a cost-efficient path (i.e. the path with minimum cost).

In case of uncongested assignment (i.e. arc, costs do not depend on flows), UE can be easily found simply assigning all flows to the *shortest path* (i.e. the path with minimum perceived cost). On the other hand, in case of congested assignment where there is *circular dependency* between costs $c$ and flows $q$ [1], when UE exists[2] it can be found through convex optimization. However, this approach can be adopted only when the uniqueness of equilibrium is guaranteed, which is not the case for transit assignment. Indeed, the sufficient condition for UE uniqueness is that the Jacobian arc cost function is positively defined (Cascetta, 2009), i.e. the travel time of a given link is a positive and increasing function of the flow on that link only, but congestion phenomena in transit networks are non-separable (i.e. cost of an arc depends also on the flows of other adjacent arcs, for example, cost of boarding a line depends on users that are already on the line). Moreover, this dependency is in general not symmetric nor monotonic.
Consequently, more complex formulations for UE are required in transit assignments, such as valid inequalities or *fixed-point formulation*. The latter, illustrated by the flowchart of figure 3.1, is the one adopted in this thesis.

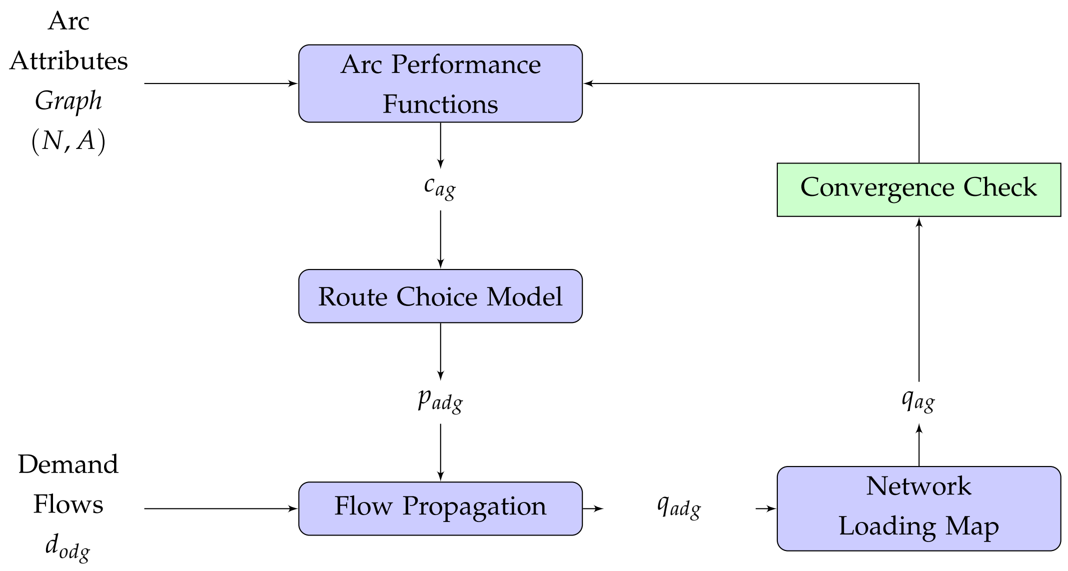

FIGURE 3.1: Static transit assignment, fixed-point formulation.

As shown in figure 3.1, inputs of the process are the travel **demand** $d_{odmg}$, on the demand side, and **arc attributes** (i.e. graph $G = (N, A)$), on the supply side.
First, the algorithm is initialized in the **Arc Performance Functions** block, which computes the *free flow cost* $c^0_{ag}$ (i.e. cost when flows on the arc is null) of each arc $a \in A$. These costs are then used by the **Route Choice Model** block to compute the *conditional probabilities* $p_{adg}$ [3].

[1] $q = \varphi_1(c)$, $c = \varphi_2(q)$, and consequently $q = \varphi_1(\varphi_2(q))$.
[2] The existence of UE is guaranteed by the continuity of the arc performance functions.
[3] Probability that users take arc $a \in A$ conditional on being at its tail node.

Then, the **Flow Propagation** block couples the conditional probabilities $p_{adg}$ with the demand flows $d_{odg}$ for each OD pair and user group $g \in G$ through an *"all-or-nothing"* assignment over the shortest hypertree [4]. The outputs of this block are the *aggregated flows* $q_{adg}$ for each arc $a \in A$, directed to each destination $d \in D$ and for each user group $g \in G$. These flows satisfy Markov's memory-less assumption, introduced in chapter 2 (i.e. they do not carry information about their origin).
Aggregated flows $q_{qdg}$ are then used as input of the **Network Loading Map (NLM)**, which assigns final flows $q_{ag}$ to each link $a \in A$, for each user group $g \in G$.
Finally, flows $q_{ag}$ are used by the **Arc Performance Functions** block to update arc congested costs $c_{ag}$ (i.e. cost depends on arc flow) for each arc $a \in A$ and each user group $g \in G$, and these costs are used by the **Route Choice Model** block to find the routes chosen by passengers and compute the new conditional probabilities $p_{adg}$.
This iterative process is repeated until UE is found, that means the flows $q_{ag}$ resulting in the previous iteration from the **NLM** block are the same of the one computed in the current iteration (*convergence*). This convergence is checked in the **Check Convergence** block.

When studying fixed-point functions that are contractions [5], convergence is obtained at first iteration. However, UE assignment function is not a contraction and, thus, some methods are necessaries to find convergence.

Hereafter, demand modelling, network topology, implemented congestion phenomena, arc performance functions, strategies, route choice model and the methods adopted to find convergence are described in detail.

## 3.1 Demand Modelling

People spend their day doing activities that often are engaged in different locations, thus different trips are produced. Each trip is associated with some disutility due to time usage and comfort, and transport demand must be estimated considering it. More precisely, travel demand is usually segmented in different *groups* $g \in G$ of users, called passengers in public transport simulation, that have different personal characteristics and trip purposes.

In the implemented model, travel demand is expressed as *Origin-Destination (OD) matrices*, usually partitioned according to land, means and time.
For what concerns the land partition, the land is divided into a set $Z$ of *zones*, where each zone represents an area with similar characteristics, such as homogeneity of activities. All the socio-economic activities located in a zone are assumed to be concentrated in one single point, called *centroid*, where trips start and end. For model purposes, two different centroids cannot be associated with the same origin or destination node. Links between centroids and transport network are done through *origin* and *destination* nodes.
From the means point of view, passengers of public transport can perform their trips in a set $M$ of different ways called *modes*. Each mode is a set of transport systems and rules on

[4]Spanning tree $T \subseteq (N, A)$, such that the hyperpath distance from its root to any other vertex is the shortest hyperpath distance.
[5]F contraction: $\forall x, y\ ||F(x) - F(y)|| \leqslant ||x - y||$.

how/where it can be accessed and implies different travel times and costs. In the implemented model, only one mode will be available, which is the *public transport mode (PuT)*. PuT is composed of all the public transport system, such as metro, train, bus, and the walk option.
Finally, demand is partitioned according to the*time interval* $t \in T$ in which the flow of users departs from a certain origin towards a certain destination.
The portion of travel demand representing user group $g \in G$ which goes from origin $o \in Z$ to destination $d \in Z$, with mean $m \in M$ at time $t \in T$ travel demand is denoted as $d^g_{odmt}$.

As in this thesis only the transit service is simulated, the mode is assumed to be public transport and variable $m$ will not be specified anymore. Moreover, the model is static and, consequently, the demand is fixed (i.e. non-dependant from OD skim matrices) and the reference to time will also be omitted.

## 3.2 Network Topology

The model represents transport supply through a directed *graph* $(N, A)$, where $N$ is the set of *nodes* $n$ and $A$ is the set of *links/arcs* $a$. The zone centroids are a subset of these nodes ($Z \subseteq N$). It is assumed that on the graph exists a non-empty set $K_{od}$ of acyclic paths connecting each origin $o \in O \subset Z$ to every destination $d \in D \subset Z$.

The graph is articulated in *sub-networks*, consisting of separate layers of arcs and nodes that are reserved to one specific transport system. Sub-networks are connected by *inter-modal arcs* and, consequently, each mode is identified indirectly by the set of inter-modal arcs that can use. For example, public transport can use connector arcs between centroids and the pedestrian network, and stop arcs between the pedestrian network and the transit lines.

Figure 3.2 represents an example of the supply graph at a given stop $s \in S$, called *transit network*. The transit graph is composed of:

- the pedestrian network (sub-network of the base network, where pedestrians can walk);
- the line network (that represents all possible trip phases for each line);
- the inter-modal arcs that connect centroids to the pedestrian network;
- the inter-modal arcs that connect the pedestrian network to the line network, which is composed of one sub-network for each service line.

Thus, the transit graph is composed of the following types of nodes and links:

- zone centroids $Z$;
- base nodes $N_{base}$ (and its subset connected to stops $B_s$);
- stop nodes $S$;
- line nodes $N_l$;
- origin and destination connector arcs $A_{orig}$, $A_{dest}$;
- base arcs $A_{base}$;

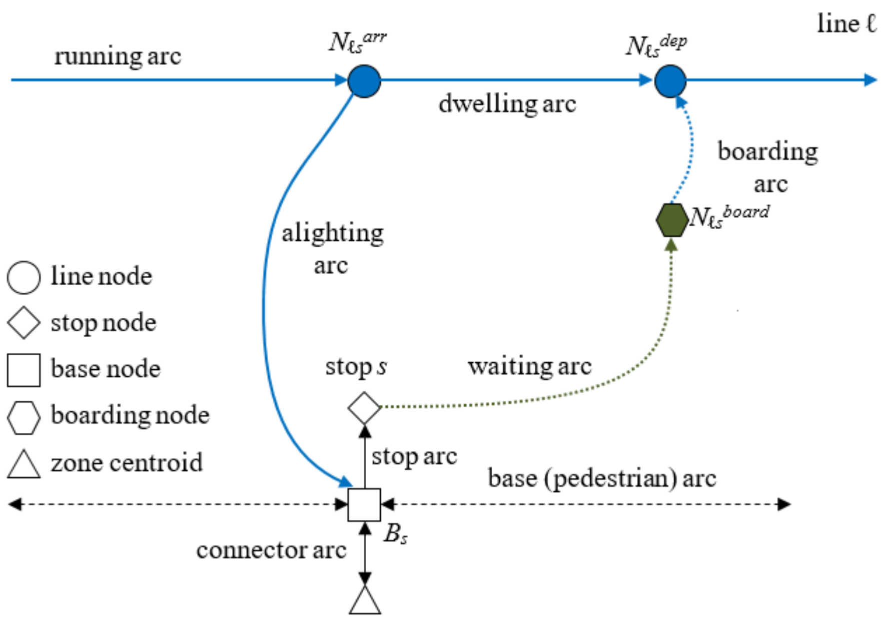

FIGURE 3.2: Topology of the transit graph at a given stop $s \in S$.

- stop arcs $A_{stop}$;
- waiting arcs $A_{board}$;
- boarding/placing arcs $A_{board}$;
- running arcs $A_{run}$;
- dwelling arcs $A_{dwell}$.

It is important to stress out that the described graph is a simplified version of the one adopted in the model. Indeed, the model can differentiate the seating option from the standing one by introducing seat arcs and stand arcs. Consequently, the graph in figure 3.2 should have more arcs. More precisely, each of the boarding/placing arcs, running arcs, dwelling arcs and alighting arcs should be divided into two different arcs, one for seat passengers and one for stand passengers and switch arcs should be added to represent the possibility of sitting during dwelling. Moreover, the implemented model studies also the possibility of not enough seats to board, thus fail-to-board arcs must be introduced as well.
However, in this section only the simplified graph of figure 3.2 will be described for sake of simplicity (indeed, from the topological point of view, having seat and stand arcs is irrelevant).

### 3.2.1 Base Network

The *base network* $(N_{base}, A_{base})$ is a sub-network of $(N, A)$ that represents infrastructures, such as roads and rails.
usepackageEach node $n \in N_{base}$ of the base network has geographic coordinates. Base arcs $a \in A_{base}$ are described by polylines with intermediate points to allow map representation though GIS (Geographic Information System).
Link $a \in A$ in the base network is characterized by the following parameters:

- $l_a^{base}$ length;
- $v_a^{walk}$ walking speed (non-walkable arcs have a null speed);
- $x_a^{walk}$ sidewalk width (non-walkable arcs have a null width).

The *pedestrian network* is the subset of the base network composed of walkable links the nodes between.

### 3.2.2 Line Network

*Line network* is a sub-network of $(N, A)$ that consists of a set $S \in N$ of nodes called *stops*, between which service operates.

Each stop $s \in S$ represents a unique geographic location where passengers can access to transit services (e.g. a platform) and that is associated with a base node $N_s \subset N$. It is assumed that transfers within a single stop take zero walking time.

Stops are served by transit services, which are organized in a set $L$ of *lines*. A line $l \in L$ serves an ordered set of stops $S_l \in S$, with no repetitions (circular lines and side-trips are excluded from the model). Each stop $s_l^- \in S_l$ is linked to the *successive stop* $s_l^+ \in S_l$ in the line by a so-called *line segment*. For visualization requirements, each line segment is associated with an acyclic path on the base network, whose support arcs are denoted *support edges*.

Stops that are usually perceived as one unique network element are grouped in *stop areas*, while similar lines and their reverse are grouped in *line sets*. However, it is assumed that grouping has no consequence for passenger route choice.

The line network has a sub-network for each service line, articulated in boarding, running, dwelling and alighting arcs to represent separate trip phases. To do this, for each stop $s \in S_l$ of line $l \in L$ two nodes, called *arrival node* $N_{ls}^{arr}$ and *departing node* $N_{ls}^{dep}$, are introduced. Moreover, for each stop $s \in S$, a *boarding node* $N_s^{board}$ is added to the graph.

For each link of the line network, variables $l_a \in L$ and $s_a \in S$ are used to indicate respectively the line and the stop associated to it. For $a \in A_{run}$, $s_a$ indicates the stop at the head of the link.

The generic line $l \in L$ is characterized by a strictly positive *running time* $t_{ls}^{run}$ for each line segment $ls = s_l^- s_l^+$ and a non-negative *dwelling time* $t_{ls}^{dwell}$ for each stop $s \in S_l$.
Often the running time is not a direct input but derives instead from more aggregated data sources. In practical the running time of line segment $ls = s_l^- s_l^+$ is the sum of the travel time of its support edges, plus a stop time $t_l^{stop}$ characteristic of the line:

$$t_{ls}^{run} = t_l^{stop} + \sum_{a \in B_{ls}} \frac{l_a}{\mathcal{S}_a}, \tag{3.1}$$

where $\mathcal{S}_a$ is the *commercial speed* along link $a \in B_{ls}$ of segment $ls$, under the assumption all lines using that edge are characterized by the same transport system.

Also, line $l \in L$ is characterized by a strictly positive *alighting time* $t_{ls}^{alight}$ to indicate the time necessary for getting off the vehicle and accessory operations (such as baggage claim) at stop

$s \in S_l$, and a strictly positive *boarding time* $t_l^{board}$, which represent the time for getting on the vehicle (usually negligible) plus the a priory anticipation of passengers in reaching the stop or the safety margin in transfers. Further details can be found in Gentile et al. (2016a).

In schedule-based networks, lines are served by an ordered set of *runs* $R_l$, where a run $r \in R_l$ represents one vehicle serving all stops of its line in order. It is assumed that each run $r \in R_l$ has a schedule with an *arrival time* $\tau_{rs}$ and a *departure time* $\theta_{rs}$ for each stop $s \in S_l$.
When using frequency-based models, only the departure time of each run from the first stop and the scheduled running times between subsequent stops are known, while dwelling times are computed through the simulation (they may depend on flows of alighting and boarding passengers as well as of vehicles occupying the platform).

A generic stop $s \in S$ has the following attributes

- $k_s^{pax}$ platform passenger capacity;
- $k_s^{veh}$ platform vehicle capacity;

while the relevant attributes of a vehicle serving the generic line $l \in L$ are

- $t_l^{min}$ minimum dwelling time;
- $t_l^{doors}$ door manoeuvre time;
- $k_l^{seat}$ seat capacity;
- $k_l^{stand}$ stand capacity;
- $k_l^{alight}$ alight capacity;
- $k_l^{board}$ board capacity.

Each line segment $ls$ of line $l \in L$ at stop $s \in S_l$ is characterized by the *expected headway* $E(h_{ls})$, which represents the time interval between two successive departures of the line from the stop. The expected headway $h_{ls}$ of line $l \in L$ at stop $s \in S_l$ is assumed to be an independent random variable with Erlang distribution, characterized by irregularity parameter $\sigma_{ls}$. Further details can be found in Gentile et al. (2016a).

From the expected headway $E(h_{ls})$, the *frequency* $f_{ls}$ of line segment $ls$ of line $l \in L$ at stop $s \in S_l$, representing the number of run departures from the stop in the studied time interval, is computed as:

$$f_{ls} = \frac{1}{E(h_{ls})}. \tag{3.2}$$

Under the assumption of constant running and dwelling times, the frequency is constant.

Finally, knowing the frequency $f_{ls}$ of line segment $ls$ of line $l \in L$ at the generic stop $s \in S_l$, for each vehicle capacity $k_l$ introduced before (e.g. seat capacity, stand capacity. etc.), the corresponding line segment capacity $K_{ls}$ can be computed as follows:

$$K_{ls} = k_l \cdot f_{ls}. \tag{3.3}$$

## 3.3 Congestion Phenomena

As stated at the beginning of this section, different congestion phenomena can be simulated by the implemented model through the introduction of additional costs on the interested arcs.
Hereafter, congestion phenomena simulated by the model.

### 3.3.1 Overcrowding Congestion

At higher densities, the more passengers are packed, the more likely they perceive this as uncomfortable and stressful. Hence, passengers travelling through crowded vehicles and platforms will be willing to re-route on longer but less-congested routes.

For what concerns passengers on-board, their discomfort increases with in-vehicle loading, which can be measured by the *saturation rate* (i.e. the number of passengers on board divided by the vehicle capacity). This is directly related to the seat availability and the density of standing passengers:

- for low/medium saturation rates, crowding discomfort is due to the lower probability of getting a seat;
- for medium/high saturation rates, crowding discomfort is due to the close physical distance with other passengers;
- for higher saturation rates, crowding discomfort is due to physical contact and pressure of other passengers.

To represent these phenomena, the running travel time for all standing passengers on-board of a vehicle of line $l \in L$ at stop $s \in S_l$ is multiplied by the following BPR-type factor $\mathrm{BPR}_{ls}^{v-crowd}$

$$BPR_{ls}^{v-crowd}\left(q_a\right) = \alpha^{crowd} \cdot \left(\frac{q_a}{K_{ls}^{stand}}\right)^{\beta_l^{v-crowd}} , \tag{3.4}$$

where:

- $q_a$ is the standing flow attempting to use the link;
- $\alpha^{crowd}$ and $\beta_l^{v-crowd}$ are vehicle overcrowding discomfort parameters;
- $K_{ls}^{stand}$ is the stand capacity of line $l$ at stop $s \in S_l$, which can be computed with equation 3.3.

With regards to passengers waiting at platforms, their discomfort due to overcrowding can be modelled in the same way, introducing a multiplication factor to the wait time.
Again, this can be done through a BPR-type $\mathrm{BPR}_s^{p-crowd}$ factor computed for stop $s \in S$ by the following equation

$$BPR_s^{p-crowd}\left(\mathbf{q_a}\right) = \alpha^{crowd} \cdot \left(\frac{\sum\limits_{b \in s^+} \left(q_b \cdot t_b\right)}{k_s^{pax}}\right)^{\beta^{p-crowd}} , \tag{3.5}$$

where:

- $k_s^{pax}$ is the platform capacity of stop $s \in S$;
- $\alpha^{crowd}$ and $\beta^{p-crowd}$ are platform overcrowding discomfort parameters;
- $\mathbf{q_a} = \sum_{b \in s^+} (q_b \cdot t_b)$ yields the expected number of passengers waiting at the stop $s \in S$ ($q_b$ is the sum of the passenger flow for each waiting arc exiting from the stop and $t_b$ is its expected time).

It can be observed that the crowding discomfort for standing passengers depends only on the flow of standing arcs (separable congestion), while the crowding discomfort for passengers at stop depends on several arc flows (non-separable).
Typical values of $\alpha^{crowd}$ and $\beta^{crowd}$ are respectively 1 and 2.

### 3.3.2 Queuing Congestion

When overcrowding is very heavy, and the crush capacity is reached on-board, no further passenger can get on the vehicle. Then, an over-saturation queue of passengers waiting at the stop is formed, increasing the expected waiting time. Clearly, this phenomenon does not affect the passengers that are already on-board, but only the ones willing to board.
The additional wait time due to the lack of space on-board increases not only with the number of passengers wishing to board but also with the number of dwelling passengers that are already on-board (queuing congestion is not separable). The dwelling passengers clearly have a priority on passengers attempting to board with respect to the occupation of the available vehicle space. Moreover, dwelling passengers are not affected by passengers waiting to board, unless discomfort is considered.
Queuing congestion is thus patently non-separable.

Two main modelling approaches are usually adopted to represent crush capacity:

- soft capacity constraints;
- strict capacity constraints.

In the first case, the vehicle capacity can be exceeded by the number of on-board passengers. Congestion affects the cost pattern through the so-called effective frequency (De Cea and Fernandez, 1993), inducing additional impedance on waiting arcs by decrease the nominal frequency of studied line. Then, the route choice model will indirectly tend to lower the on-board flow exceeding the line capacity.
However, relevant capacity violations can result at the equilibrium when no alternative route is available. In the implemented model, capacity violations are allowed only for standing passengers, as it is assumed they can squeeze, contrarily to sitting passengers.

In the second case, the vehicle capacity will never be exceeded by the number of on-board passengers.
Strict capacity constraints can be satisfied in several ways, as described in Gentile et al. (2016b). This thesis adopts the *fail-to-board probability* method (Kurauchi et al., 2003), which

removes the flow in excess from the waiting arc and ideally injects it in the following temporal layer (e.g. *"next hour"*).

The implemented model allows the user to choose which of the two capacity constraints should be adopted, assuming passengers mingling at stops.

**Effective Frequency**

The effective frequency method reduces the nominal frequency $f_{ls}$ of line $l \in L$ at stop $s \in S_l$ according to the vehicle remaining capacity.
The fundamental idea behind this approach is that, when passengers mingle at stops (that is the implemented model case), the probability to succeed in boarding the next desired approaching vehicle decreases on average with the level of on-board congestion. The latter is expressed by the saturation rate of the next line segment (running arc), where the waiting flow and the dwelling flow merge.
Therefore, efficient frequency can be computed as follows

$$f_{ls}^{eff}(q_a) = \frac{f_{ls}}{1 + \alpha^{queue} \cdot \left(\frac{q_a}{K_{ls}^{veh}}\right)^{\beta^{queue}}}, \tag{3.6}$$

where:

- $f_{ls}$ is the nominal frequency of line $l \in L$ at stop $s \in S_l$;
- $q_a$ is the flow of passengers on board of next line segment (running arc), equal to the sum of the dwelling flow $q_d$ and the waiting flow $q_a$;
- $K_{ls}^{veh}$ is the vehicle capacity of line $l \in L$ at stop $s \in S_l$, which can be computed as the sum of the seat capacity $K_{ls}^{seat}$ and the stand capacity $K_{ls}^{stand}$ of line $l \in L$ at stop $s \in S_l$, introduced in section 3.2.2 Line Network;
- $\alpha^{queue}$ and $\beta^{queue}$ are the BPR parameters for queuing congestion, whose typical values are respectively 1 and 4.

The expected wait time at stop $s \in S_l$ is, hence, calculated by applying the same equations that are valid in the uncongested case (that will be introduced in section 3.4.6 Waiting Arcs), whereas the nominal frequency is substituted with the effective frequency

$$t_{ls}^{wait}(q_a) = \frac{h_{ls}}{2} \cdot \frac{\left(1 + \sigma_{ls}^2\right)}{f_{ls}^{eff}}, \tag{3.7}$$

where:

- $f_{ls}^{eff}$ is the effective frequency of line $l \in L$ as stop $s \in S_l$ computed with equation 3.6;
- $h_{ls}$ is the expected headway of line $l \in L$ at stop $s \in S_l$;
- $\sigma_{ls}$ is the *variation coefficient* representing service irregularity of line $l \in L$ at stop $s \in S_l$.

In general, the method of effective frequency may result in travel times that are unrealistically high as static assignment models are not able to reproduce the accumulation capacity of the network. Thus, a practical way of representing queues in the context of static assignment is obtained by coupling the effective frequencies and optimal strategies into a UE model, as it will be explained in section 3.5.1 Waiting Hyperarcs.

**Fail-to-board Probability**

As in the effective frequency method, the fail-to-board probability approach is based on the idea that, when mingles queues occur at the stop, the probability to succeed in boarding the next approaching vehicle depends on the saturation flow of the vehicle and the flow of passengers wishing to board.
Hence, when strict capacities are considered and line $l \in L$ is over-saturated (remaining capacity smaller than demand), some travellers will fail to board.
Fail-to-board probability method represents this phenomenon through the introduction of the so-called fail-to-board arcs, which represent the connection with the following temporal layer (e.g. *"next hour"*). Indeed, this method removes the flow in excess from waiting arcs and, ideally, injects it in next temporal layer of a quasi-dynamic assignment [6]. In practice, as a static assignment is performed, passengers who failed to board are eliminated from the model.

Furthermore, all waiting passengers for line $l \in L$ at stop $s \in S_l$ suffer from an additional cost $c_{lsg}^{fail}$ due to the risk of fail-to-board, which is additional to the temporal cost of waiting for the arrival of the boarded service. This additional cost can be computed for each user group $g \in G$ through the risk-averseness coefficient $\gamma_g^{risk}$ towards abnormal delays (fail-to-board is perceived as a system malfunctioning), as shown in the following equation

$$c_{lsg}^{fail} = \gamma_g^{risk} \cdot \frac{p_a^{fail}}{f_{ls}\left(1 - p_a^{fail}\right)}, \tag{3.8}$$

where:

- $f_{ls}$ is the nominal frequency of line $l \in L$ at stop $s \in S_l$;
- $p_a^{fail}$ is the fail-to-board probability.

The term at the denominator of equation 3.8 can also be a sort of effective frequency and its inverse as a sort of effective expected headway. It coincides with the average additional time that the passenger must wait if fail-to-board occurs for the first approaching vehicle. Thus, the failing cost tends to infinity as the fail to board probability goes to one.
The number of passengers who will accept the risk of failing is a result of the equilibrium mechanism.

[6]In quasi-dynamic models *"a layer sequence of static models is defined, each referred to a time interval, to reproduce some dynamic phenomena, such as queuing."* (Gentile et al., 2016a)

### 3.3.3 Dwelling Delay

Passengers dwelling at stops experience some additional delays caused by difficulty in boarding, because of door capacity. Indeed, smaller is the capacity of the door, higher is the difficulty for passengers to board, as it will be explained in section 3.4.3 Dwelling Arcs.
Moreover, when the standing saturation rate of a vehicle is large (i.e. vehicle overcrowding), the nominal capacity of doors is reduced, and boarding/alighting passengers take more time to access/egress the vehicle.

The following BPR-type function is used to model the additional delay due to overcrowding

$$BPR^{dwell}\left(q^{dwell-stand}\right) = 1 + \alpha^{dwell} \cdot \left(\frac{q^{dwell-stand}}{K_{ls}^{stand}}\right)^{\beta^{dwell}} , \tag{3.9}$$

where:

- $\alpha_{ls}^{dwell}$ and $\beta_{ls}^{dwell}$ are the BPR parameters for dwelling at line $l \in L$ at stop $s \in S_l$;
- $q^{dwell-stand}$ is the flow of people standing on board of the dwelling vehicle;
- $K_{ls}^{stand}$ is the stand capacity of line $l \in L$ at stop $s \in S_l$, which can be computed with equation 3.3.

Typical values of $\alpha^{dwell}$ and $\beta^{dwell}$ are respectively 1 and 2.

## 3.4 Arc Performance Functions

Usually, the generic cost $c_{ag}$ of link $a \in A$ for user of group $g \in G$ is the sum of a *non temporal cost* $c_{ag}^{nt}$ and a *temporal cost*, given by the *discomfort coefficient* $\gamma_{ag}$ representing the *value of time* of the group $g \in G$ multiplied by the *arc travel time* $t_a$, as shown in equation 3.10.

$$c_{ag} = c_{ag}^{nt} + \gamma_{ag} \cdot t_a. \tag{3.10}$$

For what concerns the discomfort coefficient $\gamma_{ag}$ of the group $g \in G$, it can be computed by multiplying the base value of time $\gamma_g^{vot}$ for different weights (discomfort coefficients), that change according to the arc type and arc flow.
Travel time $t_a$ is also function of flows, as it depends on congestion. More precisely, travel time $t_a$ can be described by the following Bureau of Public Roads (BPR) formula

$$t_a\left(q_a\right) = t_a^0 \cdot \left(1 + \alpha\left(\frac{q_a}{K_a}\right)^{\beta}\right), \tag{3.11}$$

where:

- $q_a$ is the flow attempting to use link $a \in A$;
- $t_a^0$ is the *free flow travel time* on link $a \in A$;
- $K_a$ is the *capacity* of link $a \in A$;
- $\alpha$ and $\beta$ are the BPR parameters.

In case of capacity equal to zero, the travel time is assumed to be infinite.
Finally, non-temporal costs depend on different disutility factors, such as monetary costs and users' preferences (length, slope, pollution, etc). An example is the *value of distance* $\gamma_g^{dist}$, which is the costs that user of group $g \in G$ assigns to the travelled unit of distance.

The proposed arc performance model includes many coefficients expressing the attitudes and preferences of the different user groups. One of the best ways to set these coefficient values is by the calibration of a random utility model for route choice, based on surveys (including both revealed and stated preference questions). Examples of calibration techniques can be found in Cascetta (2001).

Hereafter, for each kind of arcs represented in figure 3.4, the model adopted for computing its cost is described. As the computation process is the same, no matter which user group is considered, here after the variable $g$ will not be specified anymore.
Null cost is assigned to stop, fail-to-board, switch seat, switch stand, origin and destination arcs.

### 3.4.1 Walking Arcs

Walking arcs represent passengers walking on sidewalks.
The non-temporal cost $c_a^{nt}$ of walking arc $a \in A_{base}$ can be obtained multiplying the length of the link $l_a$ by the value of distance $\gamma^{dist}$.
For what concerns travel time $t_a$, it coincides with the free flow travel time $t_a^0$ when the arc capacity is large enough to avoid congestion, otherwise a BPR function like the one in equation 3.11 is adopted. The free flow travel $t_a^0$ can be computed as the ratio between the is the link length $l_a$ and the link walking speed $v_a^{walk}$.
Finally, the discomfort coefficient $\gamma_a$ can be computed multiplying the value of time $\gamma^{vot}$ with the *walking discomfort coefficient* $\gamma^{walk}$.

Equation 3.12 shows the general form of equation 3.10 in case of walking arcs.

$$c_a = \gamma^{dist} \cdot l_a + \gamma^{vot} \cdot \gamma^{walk} \cdot \frac{l_a}{v_a^{walk}} \tag{3.12}$$

### 3.4.2 Running Arcs

Running arcs represent passengers sitting or standing on-board of a vehicle travelling from one stop to the successive.
The non-temporal cost $c_a^{nt}$ of running arc $a \in A_{line}$ can be obtained multiplying the length of the link $l_a$ by the value of distance $\gamma^{dist}$, summed to the product of the *kilometric fee* $c_l^{kfee}$ and a possible *fee multiplier* $\gamma^{mfee}$.
For what concerns travel time $t_a$, it is usually assumed to be just the running time of the line segment (i.e. link) introduced in equation 3.1, where the speed of the support edges is assumed to be constant [7].

[7]Transit assignment does not consider vehicle traffic, thus the time needed for the vehicle to travel a link of the base network is constant.

In case timetables exist and are strictly followed, the travel time $t_a$ is computed just as the sum, over all runs $r \in R_l$, of the difference between the head stop arrival time $\tau_{rs_h}$ and the tail stop departure time $\theta_{rs_t}$, divided by the number of runs (equation 3.13).

$$t_a^0 = \frac{\sum\limits_{r \in R_a} \left(\tau_{rs_h} - \theta_{rs_t}\right)}{|R_l|} \tag{3.13}$$

Finally, the discomfort coefficient $\gamma_a$ can be computed as the product of the value of time $\gamma^{vot}$, the *line discomfort coefficient* $\gamma^{line}$, and the *seat discomfort coefficient* $\gamma^{seat}$ (or the *stand discomfort coefficient* $\gamma^{stand}$ in case of standing arcs).
The line discomfort coefficient $\gamma^{line}$ indicates some additional attributes of the line which can influence users' cost perception (e.g. air conditioning, leather seats, etc.) and can be computed through the following equation

$$\gamma^{line} = 1 + \sum_{c \in C^{cl}} \left(\beta_c^{line} \cdot a_{lc}\right), \tag{3.14}$$

where:

- $C^{cl}$ is the set of such attributes;
- $a_{lc}$ is line $l$ attribute;
- $\beta_c$ is the user group's utility coefficient relative to the line attribute $a_{lc}$.

Equation 3.15 shows the general form of equation 3.10 in case of seat running arcs (the one for stand running arcs is similar).

$$c_a = l_a \cdot \left(\gamma^{dist} + \gamma^{mfee} \cdot c_{ls}^{kfee}\right) + \gamma^{vot} \cdot \gamma^{seat} \cdot \gamma^{line} \cdot t_a^0 \tag{3.15}$$

As explained in section 3.3.1 Overcrowding Congestion, the effect of overcrowding must be considered for standing passengers. To this purpose, a BPR-type additional time due to discomfort $t_a^{crowd}$ is added to the link cost $c_a$.
This additional time is computed multiplying the product between free flow travel time $t_a^0$ and the discomfort coefficient $\gamma_a$ by the BPR-type factor introduced in equation 3.4, as follows

$$t_a\left(q_a\right)^{crowd} = t_a^0 \cdot \gamma_a \cdot \alpha^{crowd} \cdot \left(\frac{q_a}{K_a}\right)^{\beta^{crowd}}, \tag{3.16}$$

where:

- $q_a$ is the flow attempting to use the link;
- $t_a^0$ is the free flow travel time on the link;
- $K_a$ is the *capacity* of the line;
- $\alpha^{crowd}$ and $\beta_l^{v-crowd}$ are overcrowding discomfort parameters.

### 3.4.3 Dwelling Arcs

Dwelling arcs represent vehicle dwelling at stops. Dwell time is the period when a vehicle is immobilized at a stop to allow passengers alighting and boarding, and it is composed of a series of processes:

1. doors opening after the vehicle is safely positioned at stops;
2. passenger alighting and boarding;
3. doors remaining open without passenger flow (if necessary);
4. doors closing, safety control, and vehicle departing.

The first and last process are independent of the vehicle loads and is represented by the *door manoeuvre time* $t_l^{doors}$. The second process is associated to the passenger alighting and boarding time $t_a^{dwell}\left(q^{alight}, q^{board}\right)$), that is greater than zero as the doors are assumed to have finite capacity $K_l^{alight}$ and $K_l^{board}$ (as introduced in section 3.3.3 Dwelling Delay). Finally, the third process is done only if the *operational minimum dwelling time* $t_l^{min}$ is not yet passed.
Thus, the travel time $t_a$ corresponds to the maximum between the $t_l^{min}$, if existing, and the sum of the door manoeuvre time $t_l^{doors}$ and the passenger alighting and boarding time ($t_a^{dwell}\left(q^{alight}, q^{board}\right)$).
The passenger alighting and boarding time ($t_a^{dwell}\left(q^{alight}, q^{board}\right)$) depends on non-separable congestion (alighting flow $q^{alight}$ and boarding flow $q^{board}$) and it is computed differently according to whether doors are dedicated or not, as shown in the following equation

$$t_a^{dwell}\left(q^{alight}, q^{board}\right) = \begin{cases} \dfrac{q^{alight}}{K_l^{alight}} + \dfrac{q^{board}}{K_l^{board}} & \text{not dedicated doors} \\ \max\left(\dfrac{q^{alight}}{K_l^{alight} \cdot f_{ls}}, \dfrac{q^{board}}{K_l^{board} \cdot f_{ls}}\right) & \text{dedicated doors} \end{cases}, \tag{3.17}$$

where:

- $q^{alight}$ and $q^{board}$ are respectively the corresponding alighting and boarding flow;
- $K_l^{alight}$ and $K_l^{board}$ are respectively the corresponding alighting and boarding capacity (in terms of flow);
- $f_{ls}$ is the frequency of line $l \in L$ at stop $s \in S_l$.

Moreover, since the capacity of doors can be reduced by the effects of on-board overcrowding (difficulty of moving inside the carrier due to passenger on-board), the final dwelling time due to alighting and boarding flows can be obtained multiplying the result of equation 3.17 by the BPR factor $BPR^{dwell}\left(q^{dwell-stand}\right)$ introduced by equation 3.9.
Finally, the non-temporal cost $c_a^{nt}$ of dwelling arc $a \in A_{line}$ is null, while the discomfort coefficient $\gamma_a$ is just of the value of time $\gamma^{vot}$.

Equation 3.18 shows the general form of equation 3.10 in case of dwelling arcs.

$$c_a = \gamma^{vot} \cdot \max\left(t_l^{min}, t_l^{doors} + t_a^{dwell}\left(q^{alight}, q^{board}\right) \cdot BPR^{dwell}\left(q^{dwell-stand}\right)\right). \tag{3.18}$$

### 3.4.4 Alighting Arcs

Alighting arcs represent users alighting a vehicle at stops.
The non-temporal cost $c_a^{nt}$ of the alighting arc $a \in A_{line}$ is equal to the cost of transfer $c^{tran}$.

This cost of transfer represents different aspect of transfer disutility, not necessarily connected with a measurable delay (e.g. psychological stress of possibly changing line, additional travel time variance induced by the transfer, etc.).
For what concerns the travel time $t_a$, it is assumed to be the alighting time $t_{ls}^{alight}$ of the corresponding line $l \in L$ at the corresponding stop $s \in S_l$ (introduced in section 3.2.2 Line Network).
Finally, the discomfort coefficient $\gamma_a$ is simply of the value of time $\gamma^{vot}$.

Equation 3.19 shows the general form of equation 3.10 for alighting arcs.

$$c_a = c^{tran} + \gamma^{vot} \cdot t_{ls}^{alight} \tag{3.19}$$

### 3.4.5 Placing Arcs

Placing arcs represent users that, once boarded on a vehicle, find their space on it.
The non-temporal cost $c_a^{nt}$ of the placing arc $a \in A_{line}$ can be obtained multiplying the boarding fee $c_{ls}^{bfee}$ of line $l \in L$ at stop $s \in S_l$ by possible fee multiplier $\gamma^{mfee}$.
For what concerns travel time $t_a$, it is usually assumed to be null as it overlaps with the travel time of the corresponding waiting arc and those of running arcs.

Equation 3.20 shows the general form of equation 3.10 for alighting arcs.

$$c_a = \gamma^{mfee} \cdot c_{ls}^{bfee} \tag{3.20}$$

### 3.4.6 Waiting Arcs

Waiting arcs represent passengers waiting to board a specific line. As introduced in section 3.3.2 Queuing Congestion, their cost depends on which kind of capacity constraint type (i.e. soft capacity constraints and strict capacity constraints) is adopted. Hence, a general framework for computing the cost of waiting arcs, which can be applied to both methods, is described in this section.

For what concerns the temporal cost $t_a$ of waiting arc $a$ of line $l \in L$ at stop $s \in S_l$, it is usually composed of the three following terms:

- the expected waiting time $t_{ls}^{wait}$ before boarding the vehicle;
- the additional time $t_{ls}^{plat}$ associated to overcrowding congestion at platforms (introduced in section 3.3.1 Overcrowding Congestion);
- the fail-to-board additional cost $c_{ls}^{fail}$ (introduced in section 3.3.2 Queuing Congestion).

When soft capacity constraints are considered, the effective frequency methods are adopted. Hence, all passengers must go to destination. Thus, the fail-to-board additional cost $c_{ls}^{fail}$ is null and the expected waiting time $t_{ls}^{wait}$ is computed as follows

$$t_{ls}^{wait}(q_a) = t_a^0 \cdot \frac{f_{ls}}{f_{ls}^{eff}} \cdot, \tag{3.21}$$

where:

- $f_{ls}$ is the nominal frequency of line $l \in L$ at stop $s \in S_l$;
- $f_{ls}^{eff}$ is the effective frequency computed with equation 3.6;
- $t_a^0$ is the free flow travel time of line $l \in L$ at stop $s \in S_l$, computed with in different ways according to the type of service.

In case of services characterized by frequency, free flow travel time $t_a^0$ is computed as follows

$$t_a^0 = \frac{h_{ls}}{2} \cdot \left(1 + \sigma^2\right), \tag{3.22}$$

where:

- $h_{ls}$ is the expected headway of line $l \in L$ at stop $s \in S_l$;
- $\sigma_{ls}$ is the *variation coefficient* representing service irregularity of line $l \in L$ at stop $s \in S_l$.

On the other hand, when the service is characterized by a schedule, the following equation 3.23 is applied

$$t_a^0 = min\left(\frac{h_{ls}}{2}, t_l^{board}\right), \tag{3.23}$$

where:

- $h_{ls}$ is the expected headway of line $l \in L$ at stop $s \in S_l$;
- $t_l^{board}$ is the boarding time of line $l \in L$ associated to the safety margin introduced in section 3.2.2 Line Network.

The former formula represents the rational choice of a user, which chooses the most convenient option between waiting at the stop on average for half of the expected headway, or staying at home and waiting at the stop only the time associated to the safety margin.

When fail-to-board is considered, the expected waiting time $t_a^{wait}$ of the waiting arc $a \in A$ can also be computed with equation 3.21, assuming that effective frequency $f_{ls}^{eff}$ is equal to the nominal one ($f_{ls}$). For what concerns the additional cost $c_{ls}^{fail}$ related to fail-to-board, it can be computed with equation 3.8 introduced in section 3.3.2 Queuing Congestion.

Finally, the additional time $t_{ls}^{plat}$ associated to overcrowding congestion at platforms is computed multiplying the waiting time $t_{ls}^{wait}$ by the BPR-type factor of equation 3.5.

Thus, being the discomfort coefficient $\gamma_a$ the product of the value of time $\gamma^{vot}$ and the *wait discomfort coefficient* $\gamma^{wait}$, the generalized cost of waiting arcs becomes

$$\begin{aligned} c_a\left(q_a, q_d\right) &= c_{ls}^{fail}\left(q_a, q_d\right) + \gamma_a \cdot \left(t_{ls}^{wait} + t_{ls}^{plat}\right) = \\ &= \gamma^{risk} \cdot \frac{p_a^{fail}\left(q_a, q_d\right)}{f_{ls}\left(1 - p_a^{fail}\left(q_a, q_d\right)\right)} + \\ &\quad + \gamma^{vot} \cdot \gamma^{wait} \cdot t_{ls}^{wait} \cdot \left(1 + BPR_s^{p-crowd}\left(\mathbf{q_a}\right)\right), \end{aligned} \tag{3.24}$$

where:

- $\gamma^{risk}$, $\gamma^{vot}$ and $\gamma^{wait}$ are respectively the risk averseness coefficient, the value of time for waiting arcs and the waiting discomfort coefficient;
- $p_a^{fail}$ is the fail-to-board probability;
- $f_{ls}$ is the nominal frequency of line $l \in L$ at stop $s \in S_l$;
- $t_{ls}^{wait}(q_a)$ is the waiting time of line $l \in L$ at stop $s \in S_l$, as in equation 3.21;
- $BPR_s^{p-crowd}(\mathbf{q_a})$ is the BPR-type coefficient of equation 3.5 ($\mathbf{q_a}$ is the vector of passenger flow the platform).

## 3.5 Strategies and Hyperpaths

In transport modelling, a strategy can be defined as the plan users adopt to reach the desired destination at a minimum expected cost.
Strategic behaviour usually results in lower costs and, thus, it is adopted by passengers. Indeed, it could be better to board a slower line that is arriving earlier than to wait for a faster line that will arrive later, where "*slow*" and "*fast*" do not refer to the commercial speed of the line but to the expected travel time to reach the destination once boarded the line (which may include further sub-strategies and other lines).
To model travel strategies, it is assumed that passengers choose the strategy pre-trip. Moreover, the concept of *diversion point* is introduced.
As stated by Gentile et al. (2016a), "*diversion points are nodes where users may exploit information acquired along the trip, about variables that are preventively seen as random unknowns, and on this base, can make en-route decisions on how to proceed toward the destination*".
This information may be acquired at the diversion node itself (e.g. a vehicle is approaching the stop, timetables are visible at the stop), or before with modern information system (e.g. timetables are available online).
A travel strategy is then described by an "*iterative sequence of route diversions, starting from the origin, until the destination is reached for each possible combination of events, given the considered options*" (Gentile et al., 2016a).
Two main kinds of strategies exist: strategies that are outcome of events and strategies that depend on random variables.

An example for the first kind is passengers waiting at stops, which are diversion points. Passengers choose their strategy (e.g. take the first bus that brings to destination) at home, setting the subset of the *attractive lines*. Once at the stop, they will choose which line (of the attractive set) to take, because of an event (e.g. vehicle approaching the stop). Furthermore, the journey will follow different routes depending on which event has happened.
Hence, when a strategic behaviour is adopted, passengers' decisions do not consist in which vehicle to board (which is an information that reveals itself en-route), but which strategy to adopt.
For what concerns strategies depending on random variables, examples are passengers boarding a vehicle and trying to seat, or passengers on a crowded platform and trying to board a vehicle. In these cases, the outcome of the strategy depends only on if the passengers are lucky enough to find a seat or a space on-board. Thus, passengers do not continue their

journey in a specific way because of the revelation of information (as in the previous case), but based on estimations.

From a topological point of view, strategies on transit networks are formalized with hyperarcs and hyperpaths. The topological definition of hyperarcs and hyperpaths is described in following lines.

For definition, a *hyperarc* $\breve{a}$ is a non-empty set of diversion arcs exiting from a diversion node $i \in N^{div} \subseteq N$ (i.e. a subset of its forward star $i^+$).
The generic hyperarc $\breve{a} \subseteq i^+$ of the diversion node $i \in N^{div}$ has as tail $\breve{a}^-$ the diversion node itself, while its head $\breve{a}^+$ is the set of nodes $\{a^+ : a \in \breve{a}\}$.
Each branch $a \in \breve{a}$ of a hyperarc $\breve{a} \in H$, being $H$ the set of hyperarcs, is characterized by the following variables:

- the diversion probability $p_{a|\breve{a}}$ of using branch $a$ among all branches $\breve{a}$ of the hyperarc;
- the conditional travel time $t_{a|\breve{a}}$ connected to using branch $a$ as part of the hyperarc $\breve{a}$;
- the conditional cost $c_{a|\breve{a}g}$ connected to using branch $a \in \breve{a}$ as part of the hyperarc $\breve{a} \in H$ for users of group $g \in G$.

For notation consistency, it is intended that if $a \notin \breve{a}$ then $p_{a|\breve{a}} = 0$, $t_{a|\breve{a}} = 0$, $c_{a|\breve{a}g} = 0$.

The generic *hyperpath* $k$ is a "*bush*" of arcs that connects its unique origin $k^-$ to its unique destination $k^+$. Thus, from a topological point of view a hyperpath can be defined as an acyclic sub-graph $(N_k, A_k)$ with:

- one origin;
- one destination;
- one successor arcs for each node, except the destination (which has none) and the diversion nodes (which may have more than one);
- one or more predecessor arcs for each node, except the origin (which has none).

The cost of the generic *hyperpath* $k$ can consequently be defined as the sum of its arc costs and of its hyperarc branch costs, multiplied by the probability of using these arcs when following that route. A path is a hyperpath that does not include diversions.

Figure 3.3 shows an example of hyperpath $k$ from origin $o$ to destination $d$.
From this picture, it can be observed that hyperpath $h$ is composed of the hyperarc $\breve{a} = \{a, b\}$, even if 7 possible hyperarcs are exiting (the diversion node $i \in N^{div}$ (i.e. all the possible combinations of diversion arcs $a$, $b$ and $c$).

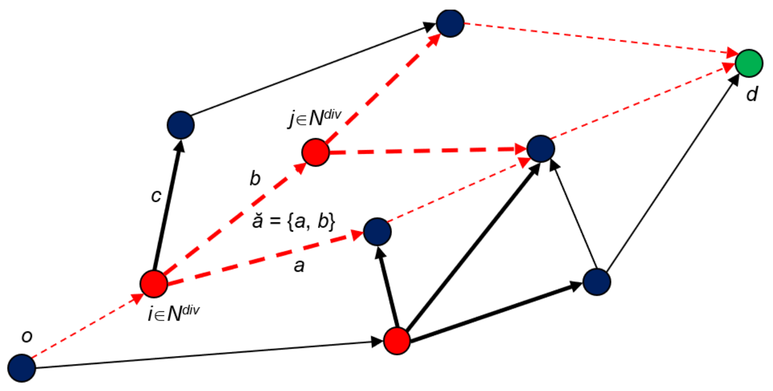

FIGURE 3.3: Example of a hyperpath $k$ from origin $o$ to destination $d$. The hyperpath is depicted in dashed red lines. The diversion nodes are in red. The bold lines are diversion arcs.

When studying hyperarcs, the combined cost $c_{\breve{a}g}$ of hyperarc $\breve{a}$ and its combined travel time $t_{\breve{a}}$ are given respectively by the following equations

$$c_{\breve{a}g} = \sum_{a\in\breve{a}} c_{a|\breve{a}g} \cdot p_{a|\breve{a}} \tag{3.25}$$

$$t_{\breve{a}} = \sum_{a\in\breve{a}} t_{a|\breve{a}} \cdot p_{a|\breve{a}} \tag{3.26}$$

where:

- $c_{a|\breve{a}g}$ is the conditional cost connected to using branch $a \in \breve{a}$ for users of group $g \in G$ (i.e. cost of link $a$);
- $p_{a|\breve{a}}$ is the diversion probability of using branch $a \in \breve{a}$ among all branches $\breve{a}$ of the hyperarc;
- $t_{a|\breve{a}}$ is the conditional travel time connected to using branch $a \in \breve{a}$ (i.e. travel time of link $a$).

Being the generic cost $c_{ag}$ of link $a \in A$ for user of group $g \in G$ is the sum of a *non-temporal cost* $c^{nt}_{ag}$ and a *temporal cost*, and combining equation 3.25, the final form of the combined cost $c_{\breve{a}g}$ of hyperarc $\breve{a}$ is the following:

$$c_{\breve{a}g} = \sum_{a\in\breve{a}} c_{a|\breve{a}g} \cdot p_{a|\breve{a}} = \sum_{a\in\breve{a}} \left(c^{nt}_{ag} + \gamma_{\breve{a}g} \cdot t_{a|\breve{a}}\right) \cdot p_{a|\breve{a}} = \sum_{a\in\breve{a}} c^{nt}_{ag} \cdot p_{a|\breve{a}} + \gamma_{\breve{a}^- g} \cdot t_{\breve{a}}. \tag{3.27}$$

The value of time $\gamma_{\breve{a}g}$ it was assumed to be equal for all diversion arcs exiting the same diversion node $i = \breve{a}^-$.

In conclusion, a strategy can be defined as a hyperpath that connects the origin-destination pair of the trip. Each strategy has an expected cost which corresponds to the cost of a hyperpath.

As the number of hyperarcs and of hyperpaths that can be defined on a network can be huge, although finite, an implicit enumeration of hyperpaths is usually adopted as the explicit enumeration is prohibitive.

### 3.5.1 Hyperarcs of the Model

Figure 3.4 represents the adopted graph for a generic line $l \in L$ and its generic stop $s \in S_l$ when strategies are modelled.
As introduced in section 3.2 Network Topology, this graph is slightly different from the general one explained in figure 3.2. More precisely, the model distinguishes stand and seat passenger flows, by splitting line links and by introducing placing and switching links. Moreover, a fail-to-board arc is added to the graph to simulate passengers that are not able to board any vehicles during the simulation time, as it will explain later.

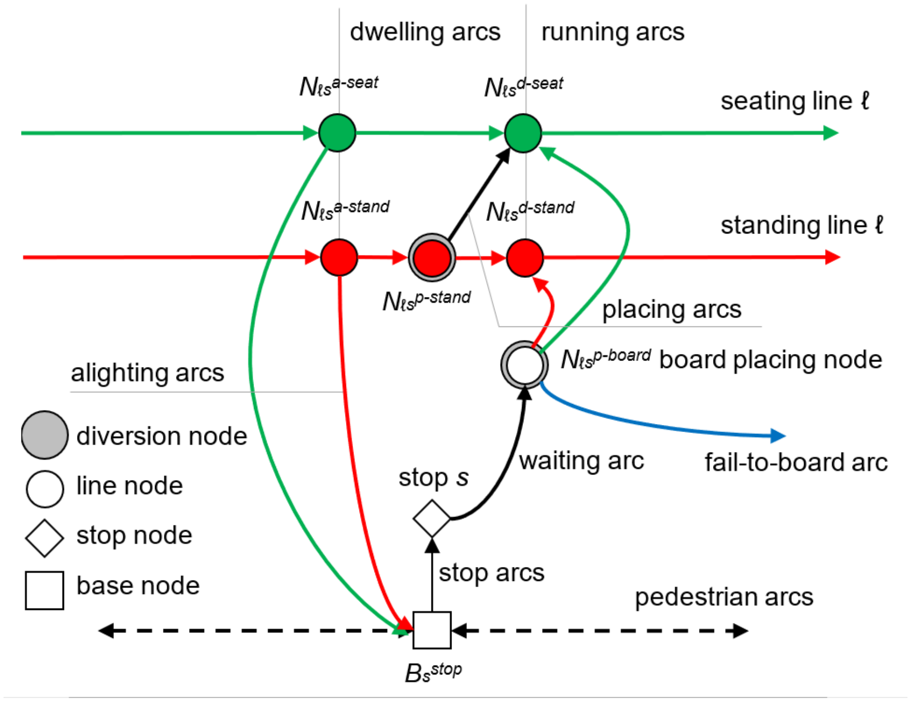

FIGURE 3.4: Example of adopted graph, with seating line (in red), standing line (in green) and fail-to-board arc (in blue).

From figure 3.4 besides the nodes explained in section 3.2 Network Topology, the graph adopted in the model has a new kind of nodes: the *diversion nodes*, introduced to represent strategy and hyperarcs. When more than one line is represented, the generic stop node $s$ becomes a diversion node.

The implemented graph has three types of diversion nodes: *board placing node* $N_{ls}^{p-board}$, *stand arrival placing node* $N_{ls}^{p-stand}$ and *stop node* $s$ (when more than one line serve the stop). Consequently, three types of hyperarcs can be introduced:

- boarding hyperarc;
- dwelling hyperarc;

- waiting hyperarc.

The first and the second hyperarc represent the probability of passengers to seat, while the third one represents the probability of passengers to board the desired line.
Noteworthy, as introduced in section 3.3.2 Queuing Congestion, for seating hyperarcs there is no choice to be made, as the probabilities depend only on a physical random event (in fact there is just one exiting hyperarc). Moreover, for waiting hyperarcs, the choice consists in the determination of the attractive line set, which depends solely on given expected headways and *remaining costs* (i.e. expected cost perceived by users to reach the destination), while the resulting diversion probabilities depend on passenger flows (no choice).

Hereafter, how these hyperarcs are implemented and how they work is described in detail.

**Seating Hyperarcs**

As introduced before, passengers travelling on transit services experience discomfort by standing versus sitting. Passengers are subject to the random process of finding a seat on-board of line $l \in L$ at stop $s \in S_l$ at two different nodes of the graph:

1. the *board placing node* $N_{ls}^{p-board}$, where passengers are boarding the desired vehicle and may find a seat according to the vehicle remaining capacity (i.e. vehicle capacity minus passengers already on-board);
2. *stand arrival placing node* $N_{ls}^{p-stand}$, where passengers standing on-board of a vehicle may find a seat because of sitting passengers alighting from the vehicle.

In the former case it is assumed that passengers who are already on board have priority over the newly boarding passengers in two ways:

1. passengers arriving at a stop sit have guaranteed a seat for the next line segment so that they either alight or remain seated;
2. passengers arriving at a stop standing who do not alight have priority over the passengers newly boarding so that these passengers have a prior chance to occupy any seat that might become vacant thanks to alighting passengers.

The model accomplishes the differentiation of the discomfort experienced by sitting versus standing passengers by explicitly modelling the limited seat availability and the random process of passengers finding a seat as a hyperarc.
Hyperarcs related to board placing node $N_{ls}^{p-board}$ are called *boarding hyperarcs*, while the ones related to stand arrival placing node $N_{ls}^{p-stand}$ are called *dwelling hyperarcs*.

For both kind of hyperarcs, the hyperarc sitting diversion probability, which is called *sit probability*, is simply given by the ratio between supply and demand seat, under the main assumption that all competing passengers have (on average) the same motivation in chasing any free seats. Hence, the diversion probability is anyhow bounded between 0 and 1.
The introduction of fail-to-sit probabilities ensures that the seating capacity of the vehicle is never exceeded.

For the dwelling hyperarc, the following equation is adopted to compute the sitting probability $p_{a|\breve{a}}$ of getting the switch seat arc $a$

$$p_{a|\breve{a}} = \frac{K_{ls}^{seat} - q_d^{seat}}{q_a}, \tag{3.28}$$

where:

- $K_{ls}^{seat}$ is the seat capacity of line $l \in L$ at stop $s \in S_l$, which can be computed with equation 3.3;
- $q_d^{seat}$ is the flow of seat passengers dwelling;
- $q_a$ is the total flow of switching passengers, given by the sum of those who actually are standing $q_a$, and those who will continue their journey sitting $q_a$.

At constant demand the probability of sitting increases with larger remaining capacity $K_{ls}^{seat} - q_d^{seat}$. Moreover, as the probability is bounded between 0 and 1, the diversion probability for stand switching arcs (which is *fail-to-sit probability*) can be computed as the 1's complement of the one computed in equation 3.28.

Boarding hyperarcs have a similar sitting probability computation. In this case, the supply is given by the seating capacity of line $l \in L$ at stop $s \in S_l$ (equation 3.3), reduced by the dwelling passengers $q_d$, while the demand is given by the boarding passengers $q_a$.
For what concerns the fail-to-sit probability, it can be computed as the 1's complement of the sitting probability when the effective frequency method introduced in section 3.3.2 Effective Frequency is adopted to represent queuing congestion.
Otherwise, if the fail-to-board probability method is adopted (section 3.3.2 Fail-to-board Probability), an additional arc is introduced at the boarding node $N_{ls}^{p-board}$ and the hyperarc is composed of three arcs instead of two. Hence, the fail-to-sit probability is computed as the 1's complement of the sitting probability, the fail-to-board one.

For what concerns the expected waiting time of these hyperarcs, it can be computed by equation 3.26. This ensures that the expected cost of reaching the destination when boarding a given line results from the average cost of sitting and standing, weighted by the sitting and fail-to-sit probability, respectively. In turn, the cost of standing includes the possibility of sitting at next stops.
As explained in Gentile et al. (2016b), this model implies that the alighting decision is not predetermined anymore. Indeed, passengers who have obtained a seat might prefer to transfer later, whereas standing passengers are more likely to transfer earlier.

**Waiting Hyperarcs**

As stated before, passengers choose strategies (pre-trip) to minimize their journey time. To this purpose, they select a set of lines (i.e. attractive set), depending on given expected headways and remaining costs, and, once at the stop, take the first vehicle approaching from this set.
The attractive set is computed with a *Greedy algorithm*, which exploits the order of lines in terms of remaining costs as follows:

- starting from an empty set,
- add the lines in increasing order of remaining cost to reach the destination once boarded,
- stop when the remaining cost of the next line is higher than the current value of the expected cost.

To model this strategic behaviour, the *expected wait time* of the first vehicle arrival for any subset of the attractive line set, as well as the probability that each line arrives first, must be computed.
The expected wait time $t_{\breve{a}}$ of hyperarc $\breve{a} \subseteq A_s$ (referred to as the combined wait time) can be computed as follows:

$$t_{\breve{a}} = \frac{1}{\sum\limits_{b \in \breve{a}} f_b}, \tag{3.29}$$

being $f_b$ the frequency of branch $b \in \breve{a}$ and assuming that the line expected headways is independently distributed with an exponential distribution[8].
For what concerns the diversion probability $p_{a|\breve{a}}$that branch $a \in \breve{a}$ corresponds to the first approaching vehicle, it can be computed as follows:

$$p_{a|\breve{a}} = \frac{f_a}{\sum\limits_{b \in \breve{a}} f_b}. \tag{3.30}$$

From equation 3.29 and 3.30, it can be seen that the combined expected time of a waiting hyperarc, as well as the diversion probability of its branches, depends solely on the frequency of its arcs (served lines). Obviously, this frequency corresponds either to the nominal frequency or to the effective frequency, when the effective frequency approach described in section 3.3.2 Effective Frequency is adopted.
The sum of the frequencies of all attractive lines is referred to as the *combined frequency* of the stop.

It is important to stress out that, as congestion increases, more (and hence slower) lines are included in the attractive set. Moreover, if all lines are congested, some passengers would rather walk than continue to wait.
This leads to a stability condition: passengers waiting at a stop would consider an attractive set that is never completely saturated and therefore each of them would be able to board the first arriving vehicle for at least one of the attractive lines.

## 3.6 Route Choice Model

The algorithm adopts an *arc based route choice model (RCM)* that produces arc conditional probabilities $p_{adg}$ for each link $a \in A$, destination $d \in D$ and user group $g \in G$. More precisely, it takes as input the cost of each arc and decides, for each node $i \in N$, the conditional probability of each arc $a \in i^+ \subseteq A$ of its forward star $i^+ \subseteq A$ (*sequential RCM*). As the flow

[8]More general results can be found in Gentile et al. (2016b).

conservation must hold for each node, the sum of the conditional probability over the forward star of each must be equal to 1.

Assumption beyond the sequential application of RCM is that, in case of additive supply models, users reach their destination through a sequence of choices at nodes, where the local alternatives are the arcs of the forward star. These choices depend only on arcs cost to reach destination and if they are connected to the destination or not.
The sequential approach is based on implicit enumeration of routes, which requires the introduction of the following variables:

- $w_{id}$ expected cost perceived by users to reach the destination $d \in D$ from node $i \in N$ (*remaining cost*);
- $q_{id}$ flow of users traversing node $i \in N$ to reach the destination $d \in D$.

Thus, RCM finds the shortest hypertree by computing the remaining cost $w_{id}$ for each node $i \in N$ to reach destination $d \in D$ and computes conditional probabilities $p_{adg}$ for each link $a \in A$, destination $d \in D$ and user group $g \in G$.
Under the assumption that only efficient routes are considered (i.e. paths getting closer to the destination with respect to some fixed cost), RCM computes the remaining cost of each node by processing nodes in *topological order* with respect to the metric chosen to define efficient routes.

RCM computes the conditional probabilities $p_{ad}$ for each arc $a \in A$ and destination $d \in D$ as follows:

- if arc $a \in A$ does not belong to the shortest path $T \in (N, A)$, its diversion probability $p_{ad}$ for destination $d \in D$ is set to zero;
- if arc $a \in T \subseteq A$ belongs to the shortest path $T \in (N, A)$ and it does not belong to any hyperarc $\breve{a} \in H$, its diversion probability $p_{ad}$ for destination $d \in D$ is set to 1;
- if arc $a \in T \subseteq A$ belongs to the shortest path $T \in (N, A)$ and it belongs to hyperarc $\breve{a} \in H$, its conditional probability $p_{ad}$ for destination $d \in D$ is equal to the diversion probability $p_{a|\breve{a}}$ (of taking arc $a \in \breve{a}^+$ as branch of the hyperarc $\breve{a} \in H$).

Hereafter, the dynamic programming method adopted to compute the shortest hypertree for each destination $d \in D$. For sake of simplicity, variable $d$ will be omitted.

### 3.6.1 Shortest Hypertree Computation

The implemented method computes the shortest hypertree through an extension of *Dijkstra Algorithm* (Dijkstra, 1959), which repeatedly applies the following *Bellman update* to every arc $a \in A$, until no further cost improvement is possible (Bellman, 1958):

$$w_{a^-} \leftarrow \min\left(w_{a^-}, w_a\right), \tag{3.31}$$

where:

- $w_{a^-}$ is the current remaining cost of node $a^- \in N$, tail of arc $a \in A$;
- $w_a$ is the remaining cost of node $a^- \in N$, tail of arc $a \in A$, computed with equation 3.32).

Thus, Bellman update checks whether using arc $a \in A$ can improve the current cost to reach the destination from the initial node $a^-$.

The remaining cost $w_i$ for each node $i \in N$ to reach the destination taking arc $b \in i^+$ of its forward star $i^+ \subseteq A$, can be computed as follows

$$w_i = c_b + w_{b^+}, \tag{3.32}$$

where:

- $b^+$ is the head node of arc $b \in i^+$;
- $c_b$ is the cost of arc $b \in i^+$;
- $w_{b^+d}$ is the remaining cost to reach the destination of the head $b^+$ of arc $b \in i^+$.

If the node is not connected to the destination, its perceived cost is infinite.

This concept can be extended to hyperarcs, considering that the remaining cost $w_i$ for each diversion node $i \in N$ to reach the destination by taking hyperarc $\check{b} \in i^+$ of its forward star $i^+$, is equal to:

$$w_i = \frac{c_{\check{b}} + \sum\limits_{b \in \check{b}} p_{b|\check{b}} \cdot w_{b^+}}{\sum\limits_{b \in \check{b}} p_{b|\check{b}}}, \tag{3.33}$$

where:

- $c_{\check{b}}$ is the cost of hyperarc $\check{b} \in i^+$;
- $p_{b|\check{b}}$ is the diversion probability for branch $b \in \check{b}$ of hyperarc $\check{b} \in i^+$;
- $w_{b^+}$ is the remaining costs to reach the destination of head $b^+ \in N$ of branch $b \in \check{b}$ of hyperarc $\check{b} \in i^+$.

To explain the extended Dijkstra Algorithm adopted to compute the shortest hypertree, the normal version of Dijkstra Algorithm to compute the shortest tree [9] is described in the following lines and illustrated by the pseudo-code of algorithm 1.
First of all, the ordered list $L$ of nodes to be visited and the list $B$ of extraction node order are created. Then list $L$ is initialized with the destination, while list $B$ is emptied. Moreover, node remaining costs (called *labels*) are set to infinity for each node $i \in N \setminus D$ and to zero for the destination.
Then, for each node $i \in L$ extracted from list $L$, each arc $a \in i^-$ of its backward star is studied and its label $w_i$ is updated according to Bellman update (equation 3.31).
Whenever a label $w_i$ of node $i \in N$ is updated, the corresponding node $i \in N$ is inserted in list $L$ according to the remaining cost $w_i$ value. If arc costs are not negative, nodes are visited

[9]Spanning tree $T \subseteq (N, A)$, such that the path distance from its root to any other vertex is the shortest path distance.

only once, and the algorithm can adopt a *label setting* approach (i.e. the cost is not updated if the node has already been extracted).
At each successful update of label $w_i$ of node $i \in N$, the algorithm records also the related arc $a \in i^-$ as *successor arc* $s_i$ of node $i \in N$.
When all arcs $a \in i^-$ from the backward star of node $i \in N$ are analysed, node $i \in N$ is added to the list of visited node $B$, which provides the topological order of the nodes.
List $B$ of the node extraction order will be reversed and used by the **Flow Propagation** block of the UE assignment algorithm (figure 3.1).

**Algorithm 1** Shortest Tree Computation Algorithm

```
1: procedure COMPUTE SHORTEST HYPERTREE

2:     Step 1 (initialization):
3:     $L \leftarrow L \cup \{i\}$
4:     $B \leftarrow L = \emptyset$
5:     $w_i \leftarrow \infty\ \forall i \in N; w_d \leftarrow 0$

6:     Step 2 (get the next arc to examine):
7:     $\forall i \in L$ :
8:     while $L \neq \emptyset$ do
9:         $L \leftarrow L - \{i\}$
10:        for all $a \in i^-$ do
11:            Bellman Update: $w_{a^-} \leftarrow \min(w_{a^-}, w_a)$
12:            if Bellman Update succeeded then
13:                $L \leftarrow L + \{a^-\}$
14:                $s_i \leftarrow a$
15:        B ←i
```

When including strategies, Dijkstra Algorithm is modified as it may happen that the remaining cost $w_{id}$ of diversion node $i = \breve{a}^- \in N$ of hyperarc $\breve{a}$ is lower than those of its heads $a^+ \in \breve{a}^+$, which prejudices the label setting approach of the Dijkstra algorithm. Indeed, nodes with lower cost could be extracted after nodes with a higher cost, so that a node already extracted can be further optimized (and can be visited more than once). Moreover, successive arc $s_i$ of the diversion node $i = \breve{a}^- \in N$ of hyperarc $\breve{a}$ cannot be defined, as the head of the hyperarc may consist in more than one arc.
As a consequence of the presence of head costs higher than tail ones, the optimal strategy can involve so-called *absorbing cycles* (e.g. an unlucky boarding passenger unable to seat, who then alights at next stop and walks back to wait again for the line at the previous stop, thus gaining another chance of sitting on-board), which is inconsistent with the definition of hyperpath (i.e. acyclic graph).

To avoid this kind of paradoxes, there are two possible approaches:

1. Force the label setting everywhere including the stop, justifying it through a risk-averse behaviour: passengers do not consider an alternative if it has a chance of ending up to a higher cost.
2. Accept the *label correcting* (i.e. the cost is updated, no matters if the node has already been extracted) except for alighting arcs, where a label setting approach is imposed, and diversion nodes, which are addressed separately.

In this thesis, approach number 2 is adopted. However, this approach is not optimal and may cause difficulties in equilibrium convergence.

The adopted shortest hypertree computation algorithm studies the diversion nodes (i.e. the diversion node is added to list $L$ and its label is updated) only if some conditions are satisfied. More precisely:

- in case of seating hyperarcs, when both heads have already been extracted;
- in case of waiting hyperarcs, every time adding the arc related to the extracted node will decrease the remaining cost of the hyperarc tail (i.e. the studied arc belongs to the attractive line set).

In case of seating hyperarc $\breve{a} \in H$, the remaining cost $w_i$ of its diversion node $i = \breve{a}^- \in N$ can be computed as follows (coupling equation 3.25 with 3.33)

$$w_i = \frac{\sum\limits_{a \in \breve{a}} p_{a|\breve{a}} \cdot (c_a + w_{a^+})}{\sum\limits_{a \in \breve{a}} p_{a|\breve{a}}}, \tag{3.34}$$

where:

- $p_{a|\breve{a}}$ is the diversion probability of taking branch $a \in \breve{a}^+$ of hyperarc $\breve{a} \in H$;
- $c_a$ is the expected cost of branch $a \in \breve{a}^+$ of hyperarc $\breve{a} \in H$;
- $w_{a^+}$ is the remaining cost of node $a^+$, head of the branch $a \in \breve{a}^+$ of hyperarc $\breve{a} \in H$.

For what concerns the remaining cost $w_i$ of the diversion node $i = \check{b}^- \in N$ corresponding to the waiting hyperarc $\check{b} \in H$ can be computed as follows (coupling equation 3.25 with 3.33)

$$w_i = \frac{\gamma_i + \sum\limits_{a \in \check{b}} w_{a^+} \cdot f_a}{\sum\limits_{a \in \breve{a}} f_a}, \tag{3.35}$$

where:

- $\gamma_i$ is the value of time for node $i \in N$;
- $w_{a^+}$ is the remaining cost of node $a^+$, head of the branch $a \in \check{b}^+$ of hyperarc $\check{b} \in H$;
- $f_a$ is the frequency of branch $a \in \check{b}^+$ of hyperarc $\check{b} \in H$.

This is valid only after the assumptions that headways are regular and that diversion arcs have only non-temporal costs.
If another branch $c \notin \check{b}$ is added to the hyperarc (i.e. *Greedy algorithm* to compute attractive line set of section 3.5.1 Waiting Hyperarcs), the new cost remaining cost $w_i$ of the waiting

hyperarc $\check{b} \in H$ becomes:

$$\begin{aligned} w_i\left(\check{b} \cup c\right) &= \frac{\gamma_i + w_{c^+} \cdot f_c + \sum\limits_{a \in \check{b}} w_{a^+} \cdot f_a}{f_c + \sum\limits_{a \in \check{b}} f_a} = \\ &= \frac{w_i\left(\check{b}\right) \cdot \sum\limits_{a \in \check{b}} f_a + w_{c^+} \cdot f_c}{f_c + \sum\limits_{a \in \check{b}} f_a}, \end{aligned} \tag{3.36}$$

where:

- $\gamma_i$ is the value of time for node $i \in N$;
- $w_i\left(\check{b}\right)$ is the remaining cost of the waiting hyperarc $\check{b} \in H$ (computed with equation 3.36);
- $f_a$ is the frequency of branch $a \in \check{b}^+$ of hyperarc $\check{b} \in H$;
- $f_c$ is the frequency of the additional arc $c \in A \notin \check{b}^+$;
- $w_{c^+}$ is the remaining cost of node $c^+$, head of the additional $c \in A$.

## 3.7 Convergence Search

The drawback of fixed point problems with respect to more classical optimization models (e.g. where the objective function is the sum of cost integrals) is the lack of rapidly convergent algorithms, which prevents precise calculations of the equilibrium solutions.

The *Method of Successive Averages (MSA)* is one of the most used convergence methods in UE assignments and converges to UE under Blum theorem conditions (see Cascetta, 2009, Appendix A).
At each iteration $k+1$, MSA compares the average $q^{k+1}$ of all preceding flows exiting the **NLM** block of figure 3.1 (computed through equation 3.37, where $q(c^k)$ are the flows at the current iteration) with the average $q^k$ computed at previous iteration $k$.
MSA usually decides whether to stop as the equilibrium was found or to iterate again, by checking the *stop criterion* represented by the inequality of equation 3.38, where $\varepsilon$ represents a threshold. Convergence is reached when $q^{k+1} = q^k$, that is for small values of the ratio $\lambda$. Thus, decreasing values of $\varepsilon$ ensure better convergence.

$$q^{k+1} = q^k + \frac{1}{k} \cdot \left(q(c^k) - q^k\right) \tag{3.37}$$

$$\gamma = \frac{\left|q^{k+1} - q^k\right|}{\left|q^k\right|} < \varepsilon \tag{3.38}$$

MSA is a very simple method, but it does not give high convergence precision, as it usually stabilizes around a value after some iterations. Moreover, MSA is very slow (in terms of number of iterations), as the step size $1/k$ becomes smaller each iteration.

To overcome MSA drawbacks, this thesis solution algorithm implements a *reduced gradient projection (RGP)* method over implicit hyperarcs.
RGP is a variant of *Gradient Projection (GP)* methods, which consist in iterative optimization algorithms for constrained functions that project the search direction on the active constraints (Rosen, 1960).
The implemented RGP is an innovative one, as it performs convergence search over arcs $a \in A$, avoiding the enumeration of hyperarcs (*RGP on implicit hyperarcs*). The implicit definition of hyperarcs is an important goal achieved, as enumerating them will be really costly in terms of computational time. Indeed, transit networks usually consists in large-scale instances and the total number of possible hyperarcs is usually huge (e.g. considering a small network composed of three stops and three lines serving those stops, there are already 16 hyperarcs: 7 waiting hyperarcs and 9 seating hyperarcs).

In general, GP algorithms for equilibrium assignment find the optimal set of alternatives and flows, solving the convex cost optimization problem (Beckman et al., 1956), where alternatives can be either hyperpaths connecting the OD pairs or arcs exiting a node towards the destinations.
More precisely, in case of explicit enumeration, the following equilibrium scheme which equalizes hyperpath costs transferring flows from all non-shortest hyperpaths to the shortest hyperpath, is adopted:

1. Compute an initial feasible solution of hyperpaths.
2. Compute the descent direction.
3. Find the optimal step size.
4. Update the hyperpath flows and link flows.
5. Compute the shortest hyperpath. If its cost is less than the minimum cost of the active hyperpaths returns to step 2, otherwise, stop.

The adopted RGP method eliminates the probability of the best alternative from the problem and considers as search direction the anti-gradient of the objective function, suitably scaled to apply shifts it in the space of probabilities. In this thesis, a scale factor equal to the alternative cost is adopted. To do this, the solution algorithm considers OD pairs sequentially, in a way similar to the GP algorithm proposed by Florian et al. (2009), based on a Gauss-Seidel decomposition. More precisely, for each OD pair, the adopted RGP method is applied.

Finally, the solution of traffic assignment may be not unique and RGP algorithms may not converge to it. Thus, a stop criterion for the solution algorithm is needed. For RGP methods, stop criteria usually consist in *gap functions*.
Gap functions indicate how far the candidate equilibrium $\mathbf{p} \in S_p$ is from an equilibrium. An example, which is the one adopted in this thesis, is the *relative gap*, shown in equation 3.39, where $\mathbf{p} \in S_p$ is the candidate equilibrium, $\mathbf{x} \in S_p$ is the candidate equilibrium of previous iteration, $\gamma(\mathbf{p})$ is the relative gap, $c(\mathbf{p})$ is the cost of the candidate equilibrium, and $c_g^{min}(\mathbf{p})$ and $c_g^{avg}(\mathbf{p})$ are the minimum and average costs of user group $g \in G$ of the candidate equilibrium $\mathbf{p} \in S_p$.

Given a candidate equilibrium , relative gap yields the best improvement on the sum of average costs that is achievable, shifting choice probabilities to better alternatives.

$$\gamma(\mathbf{p}) = \max\left(1 - \frac{c(\mathbf{p})^T \cdot \mathbf{x}}{c(\mathbf{p})^T \cdot \mathbf{p}} : \mathbf{x} \in S_p\right) = 1 - \frac{\sum\limits_{g \in G} c_g^{min}(\mathbf{p})}{\sum\limits_{g \in G} c_g^{avg}(\mathbf{p})} \in [0,1] \tag{3.39}$$

Value of $\varepsilon$ is chosen according to the purpose of the simulation. Usually, $\varepsilon \approx 10^{-2} \div 10^{-3}$ is adopted for fair check of convergence, while $\varepsilon \approx 10^{-4} \div 10^{-5}$ means good value of simulation convergence and is adopted for reliable scenario comparison. $\gamma(\mathbf{p}) = 0$ means equilibrium is reached.

# 4 Software Description

The implemented VB software follows the algorithm explained in chapter 3. More precisely the software can be described by the flowchart of figure 4.1, where:

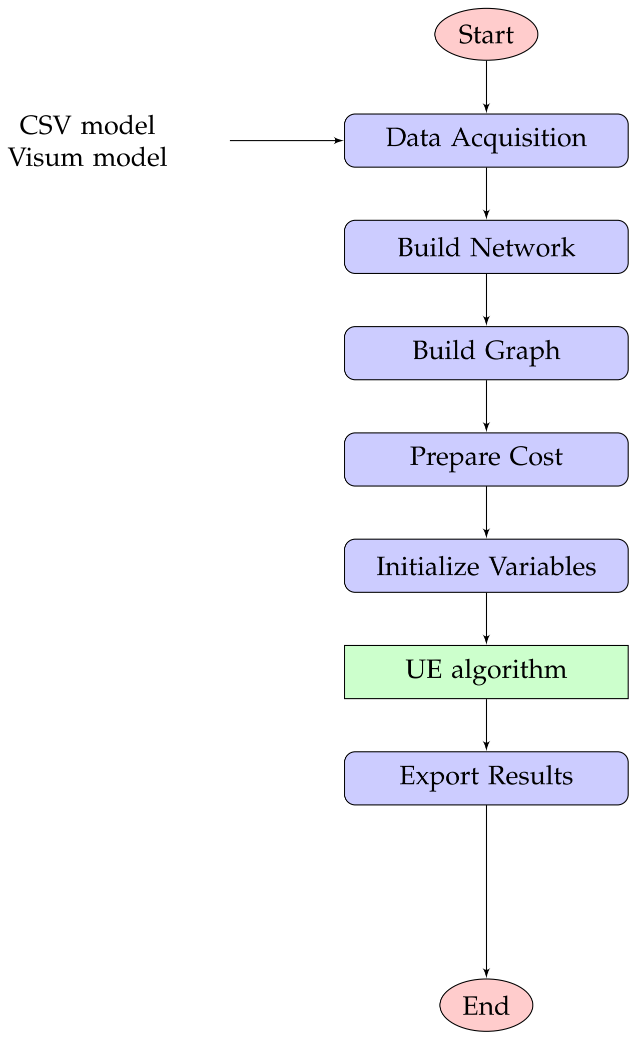

FIGURE 4.1: Flow chart of the implemented VB software.

- **Data Acquisition** block reads demand and supply of the network to study, from input files (CSV and Visum files);
- **Build Network** block populates specific VB structures representing each element of the network;
- **Build Graph** block creates graph $(N, A)$ introduced in chapter 3;

- **Prepare Cost** block sets the variables containing all link and group attributes, such as capacity, value of time, etc.;
- **Initialize Variables** block initializes assignment variables;
- **UE Algorithm** block, composed of may sub-blocks, performs UE algorithm explained in chapter 3;
- **Export Results** block exports results in different forms (eg. CSV file, image file, Visum).

Hereafter, all significant aspects of each diagram block are described.

## 4.1 Data Acquisition

As stated in chapter 3, inputs of the model are the demand $d_{odmg}$, the network topology (i.e. graph $(N, A)$) and their attributes.
The **Data Acquisition** block performs data acquisition either by a CSV file or by a Visum model. CSV files are read with the *StreamReader* class of Visual Basic (VB), while Visum models are imported through the software TDE (Transportation Data Exchange) of PTV SISTeMA (PTV SISTeMA, 2017), that is a library to import, store and export a network model (including road infrastructures, transit services, travel demand and traffic data) from/to different sources (such as databases and text files).
In this thesis, the first type of input was adopted for the implemented software and algorithm testing, while the second one for simulating larger networks.

### 4.1.1 Network Modelling

The implemented VB software takes as input network models of a certain form and converts them in the graph $(N, A)$, explained in section 3.2 Network Topology.
When modelling a network, a user should define the following elements:

- **pedestrian infrastructures:** roads, pathways, etc., where passengers can walk;
- **transit infrastructures:** roads, rail, etc., where transit vehicles can travel;
- **transit services:** lines, stops, platforms, routes, timetables, etc., which define the service itself;
- **transit demand:** OD matrices for passengers, divided in groups, and class characteristics.

When modelling the network through CSV files, the input process can be done by writing a text line for each of the following model elements: links, lines, runs, OD matrix items and user groups (called *groups*). These text lines assume different forms, as follows:

- links: "*link;tail;head;length;func*";
- lines: "*line;node;stop;func*";
- runs: "*runs;line;run;stop;arrt;dept*" [1];
- OD matrix items:"*user;orig;dest;flow;group*";

[1] *arrt* and *dept* are the arrival and the departure time of the run at a specific stop

- user groups: *"group;func"*;

where *func* represent the element function (i.e. the set of element attributes, that will be explained in section 4.1.2 Parameter Setting).
For example, if the modeller wants to define a bus line called *BUS 1*, going from stop *A* at node *1* to stop *B* at node *2*, he or she can write in the CSV file: *"BUS 1;1;A"* and *"BUS 1;2;B"*.
In section A.1 of appendix A, an example of CSV file used to model a simple network composed of three stops and one line is shown.

For what concerns modelling through Visum, the process illustrated in PTV Visum 17 Manual (PTV AG, 2017) should be followed, observing that:

- model nodes are Visum *nodes*;
- model links are Visum *links*;
- model stops are Visum *stop points*;
- model lines are Visum *line routes*;
- model line segments are Visum *line route items*;
- model OD matrices are Visum *matrices*;
- model groups are Visum *demand segments*.

### 4.1.2 Parameter Setting

To simplify the data model and make it more flexible to possible extensions, supply attributes and demand attributes are read from input files and saved in *dictionary* data structures $F$ of VB. Each dictionary represents a collection of keys and values pair of data.
Dictionaries were chosen to store inputs as they allow fast key lookups, as each item of the dictionary is just the combination of a key and a value.

To model input data, the following dictionaries have been introduced in the VB software:

- group dictionary $F_g$ for each group $g \in G$;
- link dictionary $F_a$ for each link $a \in A_{base}$ of the base-network $(N_{base}, A_{base})$;
- stop platform dictionary $F_s$ for each stop platform of stop $s \in S$ of the line network $(N_{line}, A_{line})$;
- line segment dictionary $F_{ls}$ for each line segment after stop $s \in S_l$ of line $l \in L$ of the line network $(N_{line}, A_{line})$.

Table 4.1, 4.2, 4.3 and 4.4 illustrate respectively the parameters saved in link dictionaries, stop platform dictionaries, line segment dictionaries and group dictionaries, and their associated four-letter key. Line segment parameters can also be set generally for the whole line, no matter which stop is considered.
In these tables, "("length-UM), "("time-UM) and "("int-UM), represents the unity of measurement adopted for length, time and time intervals.

TABLE 4.1: Parameters saved in the generic link dictionary $F_a$ and associated keys.

| Parameter | Key | Comment |
|---|---|---|
| $v_a^{walk}$ | *WSPE* | walking speed (length-UM/time-UM) |
| $\mathcal{S}_a$ | *TSPE* | transit speed (length-UM/time-UM) |

TABLE 4.2: Parameters saved in the generic stop dictionary $F_s$ and associated keys.

| Parameter | Key | Comment |
|---|---|---|
| $k_s^{pax}$ | *PCAP* | platform capacity (pax) |
| $\beta_s^{p-crowd}$ | *BSTP* | crowding BPR exponent (coef) |
| $\beta_s$ | *A_SX* | attribute X (if any) |

TABLE 4.3: Parameters saved in the generic line segment dictionary $F_{ls}$ and associated keys.

| Parameter | Key | Comment |
|---|---|---|
| $\mathcal{S}_{ls}$ | *LSPE* | commercial speed (length-UM/time-UM); 0 means TSPE |
| $t_{ls}^{alight}$ | *TALI* | alighting time (int-UM) |
| $t_{ls}^{board}$ | *TBOA* | boarding margin time (int-UM) |
| $h_{ls}$ | *HDWY* | expected headway (int-UM) |
| $\sigma_{ls}$ | *HVAR* | Erlang headway variation coefficient [2] |
| $lc^{kfee}$ | *KFEE* | kilometric fee (€/length-UM) |
| $c_{ls}^{bfee}$ | *BFEE* | boarding fee (€) |
| $K_{ls}^{alight}$ | *BCAP* | boarding door capacity (pax/int-UM) |
| $K_{ls}^{board}$ | *ACAP* | alighting door capacity (pax/int-UM) |
| $a^{ded} \in \{0,1\}$ | *DEDO* | dedicated doors for boarding and alighting (coeff) |
| $t_{ls}^{doors}$ | *DOTI* | door operation time (int-UM) |
| $t_{ls}^{min}$ | *TDWL* | minimum dwell time, after DOTI |
| $h_{ls}^{offs}$ | *OFFS* | offset w.r.t. simulation Initialize if no run is given (int-UM) |
| $k_{ls}^{seat}$ | *SCAP* | seating capacity (pax) |
| $k_{ls}^{stand}$ | *VCAP* | standing capacity (pax) |
| $\beta_{ls}^{v-crowd}$ | *BCOM* | discomfort BPR exponent (coef) |
| $\alpha_{ls}^{queue}$ | *AQUE* | queueing BPR multiplier (coef) |
| $\beta_{ls}^{queue}$ | *BQUE* | queueing BPR exponent (coef); 0 means fail to board |
| $\alpha_{ls}^{dwell}$ | *ADWL* | dwelling BPR multiplier (coef) |
| $\beta_{ls}^{dwell}$ | *BDWL* | dwelling BPR exponent (coef) |
| $\beta_{ls}$ | *A_LX* | attribute X (if any) |

TABLE 4.4: Parameters saved in the generic group dictionary $F_g$ and associated keys.

| Parameter | Key | Comment |
|---|---|---|
| $\gamma_g^{vot}$ | *VOFT* | value of time (€/time-UM) |
| $\gamma_g^{dist}$ | *VOFD* | value of distance (€/length-UM) |
| $\gamma_g^{wait}$ | *WAIT* | waiting discomfort (coef) |
| $\gamma_g^{walk}$ | *WALK* | walking discomfort (coef) |
| $c_g^{tran}$ | *CTRA* | transfer cost (€) |
| $\gamma_g^{stand}$ | *STND* | standing discomfort (coef) |
| $\gamma_g^{seat}$ | *SEAT* | sitting discomfort (coef) |
| $\gamma_g^{mfee}$ | *MFEE* | fee multiplier (coef) |
| $\alpha_g^{crowd}$ | *ACRW* | crowding discomfort BPR multiplier (coef) |
| $\gamma_g^{risk}$ | *RISK* | risk adverseness (coeff) |
| $nl_g^{max}$ | *MNAL* | maximum number of attractive lines (No) |
| $a_{gl}$ | *B_LX* | *A_LX* multiplier (if any) |
| $a_{gs}$ | *B_SX* | *A_SX* multiplier (if any) |

In CSV files, these attributes can be easily defined by writing a line following the form *"func;para;valu;desc"*, where *func* is the function name, *para* is the parameter key, *valu* is the parameter value and *desc* is the parameter description (e.g. *"0;BPED;2;[link] walking BPR exponent (coef)"*).
In section A.1 of appendix A, an example of CSV file used to model the attributes for a simple network composed of three stops and one line is shown.

In Visum, the modeller can define these attributes as user-defined attributes (UDA, see PTV Visum 17 Manual, PTV AG (2017)) of each network object of the network, according to the correspondence illustrated in section 4.1.1 Network Modelling.
Noteworthy, to introduce model user groups it is also necessary to define an additional POI (Point OF Interest), whose UDAs will represent the attributes of the related group.
Finally, looking at the implemented model as a pre-work for dynamic assignment, the UDAs regarding model lines that could depend on the specific time of the day (e.g. expected headway, capacity, etc.) must be set on Visum as UDAs of *time profile* elements.

## 4.2 Build Graph

After **Build Graph** block populates some specific VB structures (eg. lines, stops, nodes, zones, etc.) with the data imported by the **Data Acquisition** block, **Build Graph** block creates the graph $(N, A)$ represented in figure 3.4 by populating VB arrays *nodes* and *links*.
In **Build Graph** block, firstly:

- the links contained in VB structure *links* representing the arcs of the base network are added to VB *links* array;
- the nodes contained in VB structure *nodes* representing the nodes of the base network are added to VB *nodes* array;

and then each link of the line network is added to VB *links* array, and its corresponding nodes are added to VB *nodes* array following the sequence described in next lines.

Stop arcs, connecting the stop node $s$ to the relative base node $B_s^{stop}$, are the first added to VB *links* array. Then, for each line $l \in L$ and for each stop $s \in S_l$ of the line, the block checks whether the stop is a terminal or not.
If stop $s$ is not the first stop of line $l \in L$, the seat alighting arc and the stand alighting arc (connecting respectively the seat arrival node $N_{ls}^{a-seat}$ and the stand arrival node $N_{ls}^{a-stand}$ to the stop-associated base node $B_s^{stop}$) are added to VB *links* array.
If stop $s$ is not a terminal, arcs representing the dwelling leg of the journey are added to VB *links* array, according to the following sequence:

1. the seat dwelling arc connecting the seat arrival node $N_{ls}^{a-seat}$ to the seat departure node $N_{ls}^{d-seat}$;
2. the stand dwelling arc connecting the stand arrival node $N_{ls}^{a-stand}$ to the switch node $N_{ls}^{p-stand}$;
3. the seat switching arc connecting the switch node $N_{ls}^{p-stand}$ to the seat departure node $N_{ls}^{d-seat}$;
4. the stand switching arc connecting the switch node $N_{ls}^{p-stand}$ to the stand departure node $N_{ls}^{d-stand}$.

Then, if stop $s$ is the first stop of line $l$ or any stop excluding the last one, the waiting arc between the stop and the board placing node $N_{ls}^{p-board}$, as well as all the arcs in its forward star are added to VB *links* array. These arcs are:

1. the seat placing arc connecting to the seat departure node $N_{ls}^{d-seat}$;
2. the stand placing arc connecting to the stand departure node $N_{ls}^{d-stand}$;
3. the fail-to-board arc connecting to *"next temporal layer"*, if exists.

Again, for stops that are not the final terminal, running arcs are added to VB *links* array. More precisely:

1. the seat running arc connecting to the seat departure node $N_{ls}^{d-seat}$ to next stop seat arrival node $N_{ls_+}^{a-seat}$;
2. the stand running arc connecting to the stand departure node $N_{ls}^{d-stand}$ to next stop stand arrival node $N_{ls_+}^{a-stand}$;

are added.
Finally, connectors are added to VB *links* array.
Moreover, every time an arc is added, the **Build Graph** block checks whether both heads of this arc are already in VB *nodes* array. If not, it adds the missing ones to the array.

Figure 4.2 shows an example of complete graph $(N, A)$ for network *"OneLine_ThreeStops"* (composed of one line, three stops and three zones) obtained by plotting the VB *links* and *nodes* array after **Build Graph** block.

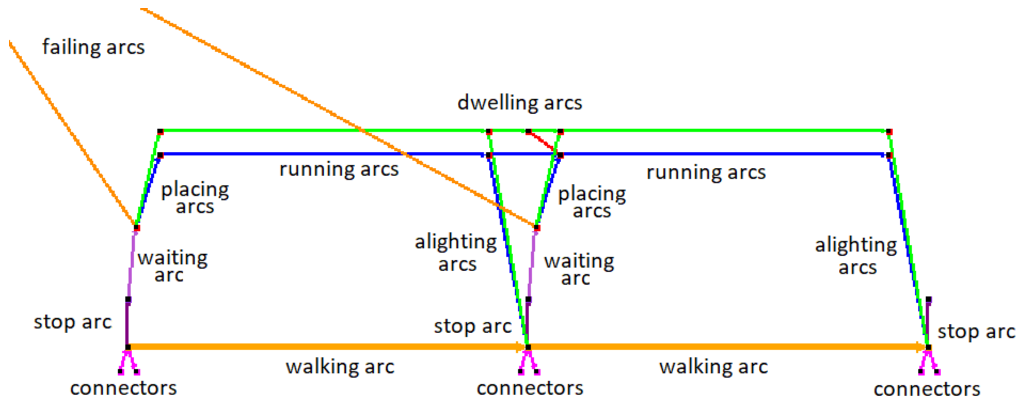

FIGURE 4.2: *OneLine_ThreeStops* graph. Base network is drawn in yellow, seating network in blue, standing network in green, seat switching arc in red.

## 4.3 Prepare Cost

The **Prepare Cost** block sets the VB software variables containing all the attributes connected to links and user groups. More precisely, the variables represented in table 4.5, 4.6 and 4.7 are populated, according to what explained in section 3.4 Arc Performance Functions.

TABLE 4.5: Variables populated related to user group $g \in G$.

| VB Variable | Model Parameter | Comment |
|---|---|---|
| *acrw* | $\alpha_g^{crowd}$ | crowding discomfort BPR multiplier (coef) |
| *risk* | $\gamma_g^{risk}$ | risk adverseness (coef) |
| *mnal* | $nl_g^{max}$ | maximum number (No) of attractive lines |
| *gvot* | $\gamma_g^{vot}$ | value of time (currency/hour) |

TABLE 4.6: Variables populated related to arc $a \in A$.

| VB Variable | Model Parameter | Comment |
|---|---|---|
| *leng* | $l_a$ | length (m) |
| *tim0* | $t_a^0$ | uncongested time (sec) |
| *capa* | $K_a$ | capacity (pax/h) of arc $a \in A$ |
| *abpr* | $\alpha_a^{BPR}$ | BPR coefficient (coef) |
| *aux1* | / | first auxiliary parameter |
| *aux2* | / | second auxiliary parameter |

Noteworthy, variable *capa* is the platform capacity in case arc $a \in A$ is a stop arc, and variable *aux1* and *aux2* represent different things according to the arc type (eg. for waiting arcs *aux1* represents the line expected headway, while for seat dwelling arcs represents whether doors are dedicated or not).

## 4.4 UE Algorithm

The **UE Algorithm** block, composed of may sub-blocks, performs the UE algorithm illustrated in figure 3.1.

TABLE 4.7: Variables populated related to both user group $g \in G$ and arc $a \in A$.

| VB Variable | Model Parameter | Comment |
|---|---|---|
| *avot* | $\gamma_{ag}^{vot}$ | discomfort coefficient of time (coef) |
| *antc* | $c_{ag}^{nt}$ | non-temporal cost (currency) |

One of the main changes of the algorithm implemented in the VB software with respect to what explained in chapter 3 is that link costs are not expressed in monetary terms (eg. Euros) but in temporal terms (eg. seconds). This is done because the currency is not a unit of measurement defined in the International System of Unit (SI), while time is. Indeed, using the currency as the unity of measurement for costs makes the results varying widely depending to the country (eg. in Europe costs will be expressed in Euros, while in the USA in Dollars), while time is officially expressed in seconds.
This change in unit of measurement is simply achieved by dividing the generalized cost $c_{ag}$ of each link $a \in A$ for each user group $g \in G$ for the value of time $\gamma_g^{vot}$ of the user group $g \in G$.

Moreover, when debugging the VB software, some problems have arisen for what concerns passengers choices at dwelling arcs, and, thus, some other changes were introduced with respect to what explained in chapter 3.
More precisely, passengers were sometimes alighting the vehicle at one stop and boarding the same vehicle at the same stop to avoid dwelling costs. Indeed, when no cost of boarding or transfer is introduced, bypassing the dwelling arc resulted more convenient.
However, this is not realistic as, even if passengers alight and board the same vehicle, they still must wait until the vehicle leaves the station and, consequently, paying all the dwelling time.

The problem of bypassed dwelling arcs is easily overcome in dynamic assignment, where assumption of not boarding the same vehicle twice can be implemented as vehicles are explicitly modelled. In static assignment case, fix the problem is more complicated. In the VB software, it was decided to disincentive the bypass of dwelling arcs by adding some extra time to placing and alighting arcs, assuming that the generic passenger is the last one to board and the last one to alight.
As this change in the model is simply a deterrent for dwelling passengers, extra costs were added only at stops that are not terminals. Obviously, this is not consistent with respect to the costs of boarding and alighting passengers, but these extra costs are significantly smaller than the total costs of paths and, thus, they do not affect path choice (except in proximity of dwelling arcs).

For what concerns alighting arcs $a \in A_{line}$ of stops that are not terminals, the extra time is computed as a variation of the dwelling time described in section 3.4.3 Dwelling Arcs, and it is equal to the sum of the following factors:

- doors opening time for line $l \in L$, which is assumed to be half of the door manoeuvre time $t_l^{doors}$;
- an additional delay $t_a^{alight}(q_a)$ due to the capacity of the doors, multiplied by the BPR coefficient $BPR^{dwell}\left(q^{dwell-stand}\right)$ representing vehicle overcrowding (equation 3.9).

The additional delay $t_a^{alight}(q_a)$ can be computed with the following equation, assuming that the generic alighting passenger is the last one to alight and that doors have a limited capacity

$$t_a^{alight}(q_a) = \frac{q_a}{K_a \cdot f_{ls}}, \tag{4.1}$$

where:

- $q_a$ is the alighting flow composed of both sitting and standing passengers;
- $K_a$ is the capacity of alighting arcs (in terms of flows), which represents the capacity of doors;
- $f_{ls}$ is the frequency of line $l \in L$ at the studied stop $s \in S_l$.

Equation 4.1 is a variant of equation 3.17.

Thus, equation 3.19 for alighting arcs of line $l \in L$ at stop $s \in S_l$ becomes the following:

$$c_a = c^{tran} + \gamma^{vot} \cdot \left( t_{ls}^{alight} + f \cdot \left( \frac{t_l^{doors}}{2} + t_a^{alight}(q_a) \cdot BPR^{dwell}\left(q^{dwell-stand}\right) \right) \right), \tag{4.2}$$

where:

- $c^{tran}$ is the transfer cost;
- $\gamma^{vot}$ is the value of time of the studied user group;
- $t_l^{doors}$ is the door manoeuvre time of line $l \in L$;
- $t_a^{alight}(q_a)$ is the additional delay introduced by equation 4.1;
- $BPR^{dwell}\left(q^{dwell-stand}\right)$ is the BPR factor representing vehicle overcrowding and introduced by equation 3.9;
- $f = \{0, 1\}$ is a parameter which indicates whether the stop is a terminal ($f = 1$) or not.

When boarding passengers are considered, the placing arcs $a \in A_{line}$ at stops that are not terminal have the additional time to be added is the sum of the following terms:

- door manoeuvre time $t_l^{doors}$ of line $l \in L$;
- an additional delay $t_a^{place}(q_a, q_b)$ (where $q_b$ is the flow of corresponding alighting arcs) due to the capacity of the doors, multiplied by the BPR coefficient $BPR^{dwell}\left(q^{dwell-stand}\right)$ representing vehicle overcrowding (equation 3.9).

The additional delay $t_a^{place}(q_a, q_b)$ is due to the assumption that doors have limited capacity and can be computed with the following equation, assuming that the generic boarding passenger is the last one to board (and, thus, he or she must suffer all the delay due to boarding and alighting passengers)

$$t_a^{place}(q_a, q_b) = \begin{cases} \dfrac{q_a}{K_a \cdot f_{ls}} + \dfrac{q_b}{K_b \cdot f_{ls}} & \text{not dedicated doors} \\ \max\left\{\dfrac{q_a}{K_a \cdot f_{ls}}, \dfrac{q_b}{K_b \cdot f_{ls}}\right\} & \text{dedicated doors} \end{cases} \tag{4.3}$$

where:

- $q_a$ and $q_b$ are respectively the alighting flow and boarding flow composed of both sitting and standing passengers;
- $K_a$ and $K_b$ are respectively the capacity of alighting arcs and placing arcs in terms of flow (which represent the capacity of doors);
- $f_{ls}$ is the frequency of line $l \in L$ at studied stop $s \in S_l$.

Equation 4.3 is a variant of equation 3.17.
Thus, equation 3.20 for placing arcs of line $l \in L$ at stop $s \in S_l$ becomes the following

$$c_a = \gamma^{mfee} \cdot c_{ls}^{bfee} + f \cdot \left(t_l^{doors} + t_a^{place}(q_a, q_b) \cdot BPR^{dwell}\left(q^{dwell-stand}\right)\right) \tag{4.4}$$

where:

- $\gamma^{mfee}$ is the fee multiplier;
- $c_{ls}^{bfee}$ is the boarding fee of line $l \in L$ at stop $s \in S_l$ ;
- $t_l^{doors}$ is the door manoeuvre time of line $l \in L$;
- $t_a^{place}(q_a, q_b)$ is the additional delay introduced by equation 4.3;
- $BPR^{dwell}\left(q^{dwell-stand}\right)$ is the BPR factor representing vehicle overcrowding and introduced by equation 3.9;
- $f = \{0, 1\}$ is a parameter which indicates whether the stop is a terminal ($f = 1$) or not.

## 4.5 Export Results

Model results consist in the distribution of flows over the network once equilibrium is found and the expected costs of traversing each arc. To analyse the solution, also how the equilibrium was found (*RGAP* trend) and the grade of congestion (i.e. increase of costs due to congestion) are important; thus, these data are exported as well.
The results can be visualized through different ways: 1) CSV file; 2) image file; 3) Visum. In this thesis, the first and the second way was usually adopted for debugging and phenomena study, while the third one was used for large simulations or not linear networks.
Hereafter, the possible outputs are described.

### 4.5.1 Output format - CSV file

For what concerns the CSV file representing flows and costs, it consists of one text line for each arc $a \in A$. These lines follow the form *"arc;arc;tail;head;cap;Flow1;Flow2; ... ; Flow$|G|$; Cost1;Cost2;...;Cost$|G|$"*, where $|G|$ is the number of user groups. Thus, CSV file can represent

in detail the flow and the expected cost of each user group $g \in G$ on each link $a \in A$.
Results about the equilibrium trend are illustrated in another CSV file, where each text line represents an assignment iteration according to the form "*iter,clock,rgap,alfa*", where *clock* is the iteration clock time, and *rgap* and *alfa* are respectively the RGAP and the step size corresponding to the iteration.
An example of CSV files representing assignment results can be found in section A.2 of appendix A.

### 4.5.2 Output format - Image file

Another way of representing results is to draw them on a graph representation. Thus, the image file that shows the results consists of a representation of graph $(N, A)$, where each link width depends on the flow assigned to the link at the equilibrium. More precisely, if the flow on arc $a \in A$ is null at the equilibrium, the arc is not drawn, otherwise the brush size $b$ is computed by the following formula:

$$b = 7 \cdot \frac{q_a}{\max\limits_{b \in A} (q_b)}, \tag{4.5}$$

where $q_a$ is the flow on arc $a \in A$ and $q_b$ is the flow of arc $b \in A$.

Moreover, for each arc, a label indicating the flow and its capacity, if relevant [3], is plotted.
An example of image file representing assignment results can be found in section A.2 of appendix A.

When drawing a representation of the graph $(N, A)$, base network nodes $N_{base}$ are plotted according to their coordinates, if available (i.e. Visum input). Usually, this is not the case of network models imported by CSV files, thus some values must be assigned to the coordinates.
More precisely, when coordinates are not available they are computed sequentially by setting the position of a chosen origin node and drawing all the nodes of its forward star at a certain distance, with a certain angle, assuming that transit lines have linear infrastructure.
Thus, taking node $n_1 \in N_{base}$ as origin node and node $n_2 \in N_{base}$ as its forward star node to be plotted, the coordinates of node $n_2$ are computed with the following formula:

$$x_{n_2} = x_{n_1} + 1000 \cdot l_{n_1,n_2} \cdot \cos\left(c \cdot \frac{\pi}{12}\right), \tag{4.6}$$

$$y_{n_2} = y_{n_1} + 1000 \cdot l_{n_1,n_2} \cdot \sin\left(c \cdot \frac{\pi}{12}\right), \tag{4.7}$$

where $l_{n_1,n_2}$ is the length of the link connecting the two nodes, and $c$ is a counter indicating how many arcs are already drawn from the forward star of node $n_1$.

Furthermore, different colours are adopted according to the type of arcs. More precisely, connectors are plotted in magenta, base network arcs are plotted in yellow, stop arcs are plotted

[3] Capacity is indicated only for those arcs that may produce congestion for lack of capacity (i.e. running arcs and stop arcs).

in purple and line arcs are plotted in the colour indicated by the line name (if available) or in one of the predefined colours (depending on the line plotting order).

### 4.5.3 Output format - Visum

In case inputs are imported from a Visum model, results can also be represented through Visum. More precisely, the VB software creates two files to be imported in the Visum model, one CSV file containing the results in form of UDAs and one XML file containing the graphics of the results. Examples of these files and their outputs can be found in section A.2 of appendix A.

The following UDAs are defined to represent flows and congestion:

- "*PUTUE_WALK*" representing the flow of users (user/h) walking on the UDA-related *link* or *connector*;
- "*PUTUE_SEAT_FLOW_lineName*" representing the flow of users (user/h) sitting on line $lineName$, on the UDA-related *link*;
- "*PUTUE_STAND_FLOW_lineName*" representing the flow of users (user/h) standing on line $lineName$, on the UDA-related *link*;
- "*PUTUE_STAND_CONG_lineName*" representing the congestion level for users standing on line $lineName$, on the UDA-related *link*;
- "*PUTUE_BOARD_FLOW_lineName*" representing the flow of users (user/h) boarding on line $lineName$ at the UDA-related *stop point*;
- "*PUTUE_ALIGHT_FLOW_lineName*" representing the flow of users (user/h) alighting on line $lineName$ at the UDA-related *stop point*;
- "*PUTUE_WAIT_CONG_lineName*" representing the congested time (s) for users waiting line $lineName$ at the UDA-related *stop point*.

Except the first UDA ("*PUTUE_WALK*"), the others are defined for each line in the network.

The congestion level $\kappa_a$ of arc $a \in A_{line}$ is defined as the percentage of cost increase from uncongested network state to UE, as follows

$$\kappa = \frac{\overline{c_a} - \overline{c_a}^0}{\overline{c_a^0}} \cdot 100, \tag{4.8}$$

where:

- $\overline{c_a} = \sum\limits_{g \in G} c_a$ is the sum for each user group $g \in G$ of the UE costs of arc $a \in A_{line}$;
- $\overline{c_a}^0 = \sum\limits_{g \in G} c_a^0$ is the sum for each user group $g \in G$ of the UE costs of arc $a \in A_{line}$.

As stated before, a file indicating the graphics to represent results is also produced.

Results related to link flows are represented as bars on the link, with different colours depending on the flow type and line, and on the level of congestion. More precisely, the walking flow of passengers is represented with a magenta bar, while seating flows are plotted in the colour indicated by the line name (if available) or in one of the predefined colours (depending on the line plotting order).
For what concerns standing flows, they assume different gradation of their related seating flow, depending on the level of congestion. Higher the congestion is, darker the gradation.

Results related to stop flows are represented with a histogram for each stop, where alighting and boarding flows are illustrated by different columns plotted in gradations of the colour indicated by the line name (if available) or in one of the predefined colours (depending on the line plotting order).
For what concerns congested times, they are plotted in the same histogram as columns of line-related colour, whose gradation its darker than corresponding boarding and alighting flow ones. Congested times do not include cost due to overcrowding at platforms and fail-to-board, as they are meant to indicate solely the estimated waiting time for passengers boarding the studied line at the corresponding stop.

# 5 Model Validation and Simulations

The model validation process consisted in reproducing the congestion phenomena described in section 3.3 Congestion Phenomena for small networks, as well as the strategic behaviour of section 3.5 Strategies and Hyperpaths, and analysing them.
More precisely, it was checked whether a specific congestion phenomenon causes changing in the passengers' choices and if these changes are sensible in terms of passengers behaviour. While doing this, the software and the algorithm were tested in terms of flow conservation and cost convergence at UE. Moreover, the satisfaction of the constraints defined by the model, such as the not violation of seating capacity and the priority of seating arcs over standing ones, was also ensured.
Moreover, analyses were conducted about the convergence speed and precision of the implemented convergence method (i.e. RGAP on implicit hyperarcs).
After validating the model, some simulations of larger networks were conducted, and results were analysed and compared with the results of Visum HB assignment procedure, described in section 2.2.1 Headway-Based Procedure

Hereafter, the process followed for validating the model and some relevant simulations are described.
All simulations were conducted over a simulation interval of 1 hour, and by adopting minutes, hours and kilometres as unit of measurements for time intervals, time and length.

## 5.1 Model Validation

To validate the model, the following three simple networks were adopted:

- ***NetworkA***, composed of one line serving two stops and used for testing on-board overcrowding congestion, platform overcrowding congestion, queuing congestion and boarding hyperarcs.
- ***NetworkB***, composed of two lines serving two stops and used for testing waiting hyperarcs.
- ***NetworkC***, composed of one line serving three stops and used for testing dwelling delay due and dwelling hyperarc.

Table 5.1 and 5.2 illustrate CSV inputs related to base and line network for *NetworkA*, table 5.3 and 5.4 the ones related to *NetworkB*, and table 5.5 and 5.6 the ones related to *NetworkC*, while figure 5.1, 5.2 and 5.3 show their graph $(N, A)$. As the travel demand was introduced only for one user group, user group $g \in G$ will not be specified.

Parameter values were changed according to the phenomenon to observe. Table 5.7 shows the standard values adopted if not differently specified. Noteworthy, the following values were assumed in every simulation:

- line $l \in L$ expected headway $h_l$ equal to *6min*;
- sitting discomfort coefficient $\gamma_g^{seat}$ equal to *1*;
- standing discomfort coefficient $\gamma^{stand}$ equal to *1.5*;
- waiting discomfort coefficient $\gamma^{wait}$ equal to *1*;
- walking discomfort coefficient $\gamma^{walk}$ equal to *1*;
- walking speed $v_a^{walk}$ of walking arcs $a\, in A_{base}$ equal to *5Km/h* ;
- commercial speed $\mathcal{S}_a$ of running arc $a \in A_{line}$ equal to *20km/h*.

Finally, RGAP equal to $10^{-9}$ was adopted as general stop criterion, and the maximum number of iterations was set equal to 500. This means that, if convergence is not achieved before 500 iterations, the algorithm is stopped.

TABLE 5.1: *NetworkA*, CSV file of the base network model.

| link | tail | head | leng | func |
|---|---|---|---|---|
| 1 | 1 | 2 | 5 | 0 |

TABLE 5.2: *NetworkA*, CSV file of the line network model.

| line | node | stop | func |
|---|---|---|---|
| Bus1 | 1 | 1 | 0 |
| Bus1 | 2 | 2 | 0 |

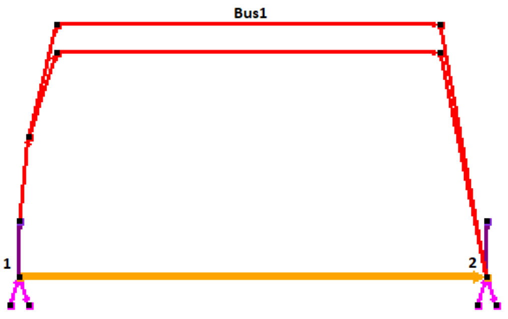

FIGURE 5.1: *NetworkA*, graph $(N, A)$. Fail-to-board arc not plotted.

TABLE 5.3: *NetworkB*, CSV file of the base network model.

| link | tail | head | leng | func |
|---|---|---|---|---|
| 1 | 1 | 2 | 5 | 0 |

TABLE 5.4: *NetworkB*, CSV file of the line network model.

| line | node | stop | func |
|---|---|---|---|
| Bus1 | 1 | 1 | 0 |
| Bus1 | 2 | 2 | 0 |
| Bus2 | 1 | 1 | 0 |
| Bus2 | 2 | 2 | 0 |

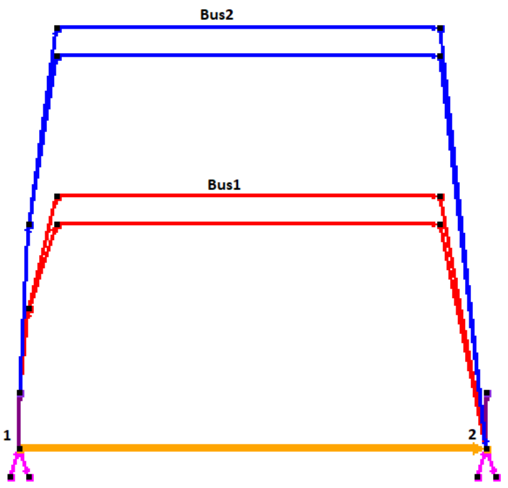

FIGURE 5.2: *NetworkB*, graph $(N, A)$. Fail-to-board arcs not plotted.

TABLE 5.5: *NetworkC*, CSV file of the base network model.

| link | tail | head | leng | func |
|---|---|---|---|---|
| 1 | 1 | 2 | 5 | 0 |
| 2 | 2 | 3 | 5 | 0 |

TABLE 5.6: *NetworkC*, CSV file of the line network model.

| line | node | stop | func |
|---|---|---|---|
| Bus1 | 1 | 1 | 0 |
| Bus1 | 2 | 2 | 0 |
| Bus1 | 3 | 3 | 0 |

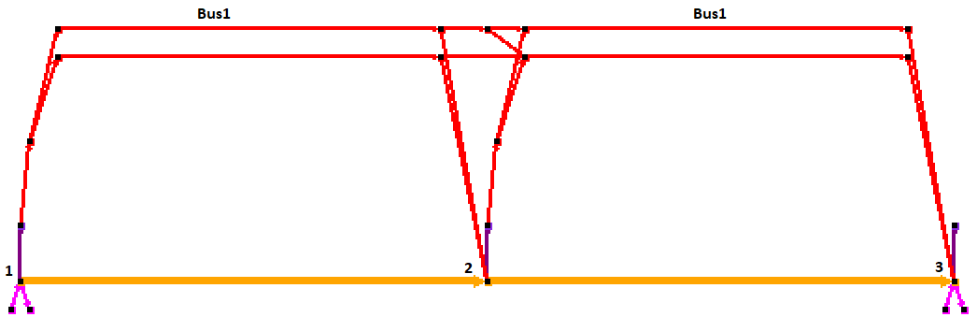

FIGURE 5.3: *NetworkC*, graph $(N, A)$. Fail-to-board arcs not plotted.

TABLE 5.7: Standard parameter settings adopted to study congestion phenomena. Value equal to 100000 represents infinite.

| func | para | valu | desc |
|---|---|---|---|
| 0 | fromLengUnitToMeter | 1000 | |
| 0 | fromTimeUnitToSec | 3600 | |
| 0 | fromIntUnitToSec | 60 | |
| 0 | WSPE | 5 | [link] |
| 0 | TSPE | 20 | [link] |
| 0 | LSPE | 0 | [line-seg] |
| 0 | TALI | 0 | [line-seg] |
| 0 | TBOA | 0 | [line-seg] |
| 0 | HDWY | 6 | [line-seg] |
| 0 | HVAR | 0 | [line-seg] |
| 0 | KFEE | 0 | [line-seg] |
| 0 | BFEE | 0 | [line-seg] |
| 0 | DEDO | 0 | [line-seg] |
| 0 | DOTI | 0 | [line-seg] |
| 0 | TDWL | 0 | [line-seg] |
| 0 | MDWL | 0 | [line-seg] |
| 0 | OFFS | 0 | [line-seg] |
| 0 | VOFD | 0 | [class] |
| 0 | VOFT | 60 | [class] |
| 0 | WAIT | 1 | [class] |
| 0 | WALK | 1 | [class] |
| 0 | CTRA | 0 | [class] |
| 0 | STND | 1.5 | [class] |
| 0 | SEAT | 1 | [class] |
| 0 | MFEE | 0 | [class] |
| 0 | ACRW | 0 | [class] |
| 0 | APED | 0 | [link] |
| 0 | BPED | 2 | [link] |
| 0 | BCAP | 100000 | [line-seg] |
| 0 | ACAP | 100000 | [line-seg] |
| 0 | ADWL | 0 | [line-seg] |
| 0 | BDWL | 2 | [line-seg] |
| 0 | BCOM | 0 | [line-seg] |
| 0 | AQUE | 0 | [line-seg] |
| 0 | BQUE | 6 | [line-seg] |
| 0 | BSTP | 0 | [stop-plat] |
| 0 | RISK | 1 | [class] |
| 0 | MNAL | 10 | [class] |
| 0 | SCAP | 100000 | [line-seg] |
| 0 | VCAP | 100000 | [line-seg] |
| 0 | PCAP | 100000 | [stop-plat] |

### 5.1.1 Test - On-board Overcrowding Congestion

As stated before, on-board overcrowding congestion is tested on *NetworkA*, composed of one line serving two stops. To this purpose, many simulations were conducted, each of them with different amount of travel demand going from node *1* to node *2*.
In these simulations, the crowding BPR multiplier $\alpha^{crowd}$ was set equal to 1, while the on-board crowding BPR exponent $\beta^{v-crowd}$ was set equal to 2. Moreover, to reproduce the congestion phenomenon related to vehicle crowding, line seating capacity was set equal to 0 passengers per hour and line standing capacity was set equal to 40 passengers per hour (which means 400 passengers per hour, being the expected headway of 6 minutes).
The demand was varied from 100 to 1000 passengers per hour, with steps of 100 passengers per hour, and RGAP equal to $10^{-9}$ was adopted as stop criterion.

Figure 5.4 illustrates the trend of expected cost for stand running arc during the assignment algorithm. More precisely, the horizontal axis shows the number of iterations necessaries to find equilibrium, while the vertical axis represents the expected cost of stand running arc for flows exiting the **cost functions** block of figure 3.1. Trends for different travel demand values are shown in different colours.
From this picture, growing travel demand implies growing costs, as expected. Indeed, greater flows on stand running arc cause larger expected costs as consequence of on-board overcrowding congestion. Furthermore, it can also be observed that greater travel demands generally stem in a larger number of iterations necessary to find the equilibrium expected cost.
More precisely, it can be observed that there is a threshold (in this case, demand equal to 500 passengers per hour) below which the equilibrium is found in two iterations, which consist in performing an *"all-or-nothing assignment"* to the shortest uncongested hyper-tree and doing another iteration to check convergence. For demand flows higher than 400 passengers per hour, the walking option becomes competitive with the one of boarding line *BUS1* and, hence, the algorithm looks for the exact distribution of flows that balances out the hyperpath expected costs. Thus, more iterations are needed to converge.

Figure 5.5 shows the trend of RGAP for the different travel demands during the algorithm iterations. RGAP trends for different travel demand values are shown in different colours.
Again, it can be observed that greater demands generally imply harder-to-find solutions and, consequently, slower convergence (in terms of number of iterations). Moreover, this graph highlights the threshold introduced in the previous paragraph (500 passengers per hour) below which the number of iterations necessary to find the convergence is only two. However, there is no evident relation between convergence speed and growing travel demand for flows above the threshold.
From figure 5.5 it can also be seen that, by imposing a less restrictive stop criterion (e.g. RGAP equal to $10^{-5}$ and $10^{-4}$), the convergence speed would have been faster (in terms of number of iterations). More precisely, when the RGAP stop criterion is set equal to $10^{-4}$, maximum 9 iterations are needed by the algorithm, against the 23 necessaries with the RGAP equal to $10^{-9}$. When testing small network, the difference in number of iterations due to the stop criterion is not relevant, but it influences widely the algorithm speed in case of large networks (in terms of number of iterations). Moreover, from the practical point of view,

high convergence precision is not important, as usually precision in the order of magnitude of $10^{-4}$ is enough.

Figure 5.6 shows the equilibrium of expected costs for hyperpaths when the travel demand is equal to 1000 passengers per hour, where the two lines represent the trend of the expected general costs of two hyperpaths. More precisely, the violet one for the walking hyperpath and the red one for the transit line *BUS1* hyperpath.
Noteworthy, as no diversion node is present in the graph (there are only one line and no seating arcs), the equilibrium is between paths and not hyperpaths.
It can also be seen that the walking option is part of the solution only after the second iteration. This is because, as stated before, at first iteration all flows are assigned to the shortest uncongested hyper-tree, i.e. boarding the transit line.
Also, it can be observed that the RGP algorithm corrects the flows with gradual steps.

Figure 5.7 shows the flow of passengers boarding the transit line *BUS1* for each demand flow, where each column is a different demand flow. From this picture, it can be observed that maximum 473 passengers per hour are assigned to the transit line *BUS1*. Thus, all passengers exceeding this value choose the walking option.
Obviously, these results strictly depend on standing capacity value and BPR parameters. For example, changing the value of $\beta^{v-crowd}$ from 2 to 4 reduces the maximum number of boarding passengers to 429 passengers per hour.

Finally, to study more in detail how the algorithm performs when on-board overcrowding congestion is activated, additional simulations were conducted for travel demand equal to 1000 passengers per hour and for varying on-board crowding BPR exponent $\beta^{v-crowd}$ (from 2 to 10). Figure 5.8 shows the value of expected flow boarding the transit service for each $\beta^{v-crowd}$. It can be observed that, for growing $\beta^{v-crowd}$, the number of passengers choosing to take line *BUS1* decreases, as the costs of the stand running arc increases due to congestion. Similar conclusions can be observed by varying the crowding BPR multiplier $\alpha^{crowd}$.

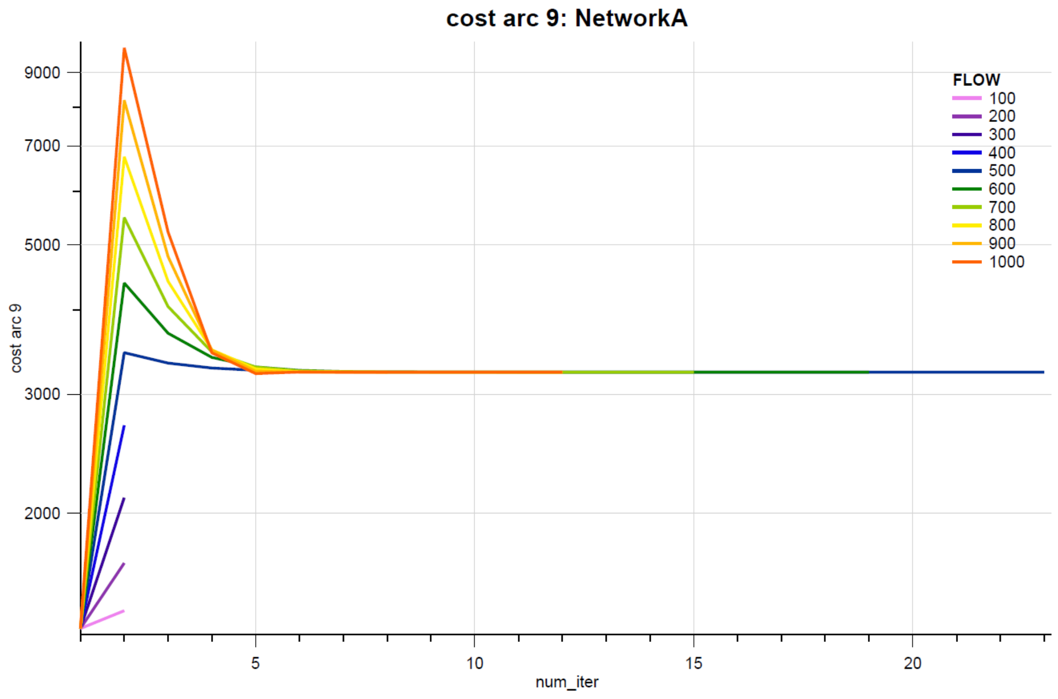

FIGURE 5.4: On-board Overcrowding Congestion test. Expected cost of stand running arc.

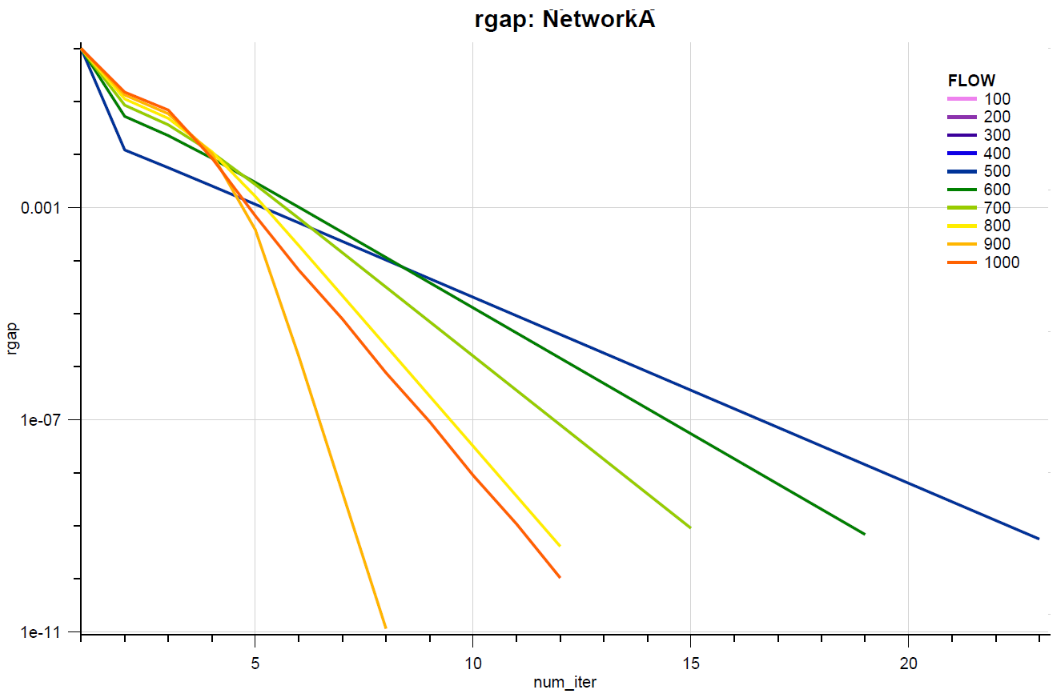

FIGURE 5.5: On-board Overcrowding Congestion test. RGAP of the solution algorithm for $\beta^{v-crowd} = 2$.

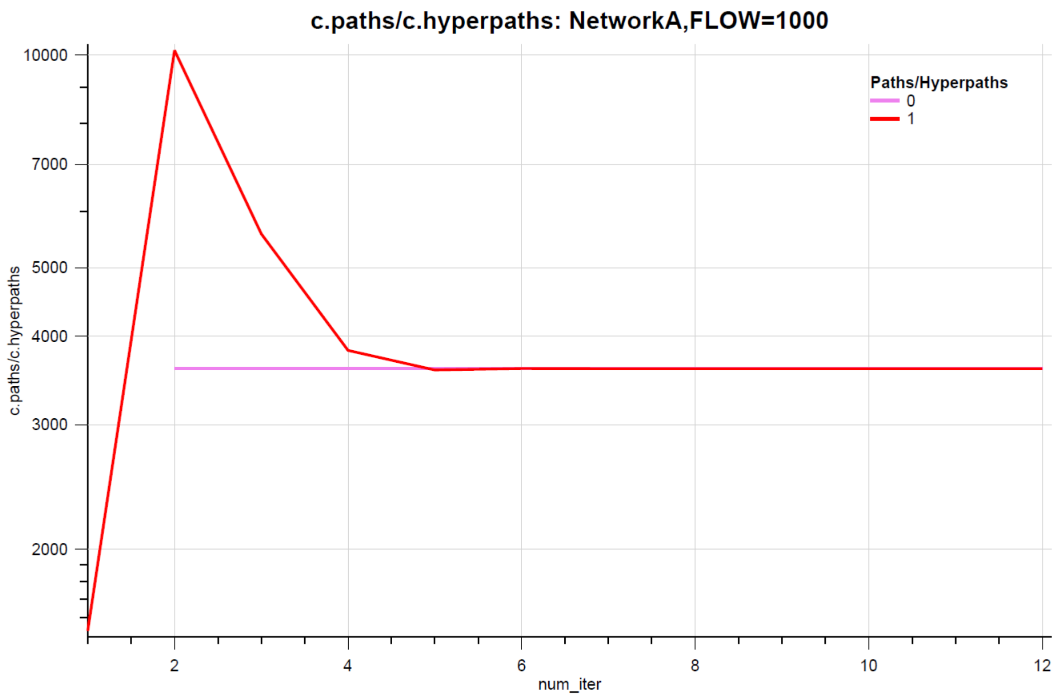

FIGURE 5.6: On-board Overcrowding Congestion test. Hyperpath cost equilibrium: The walking option is represented in violet, while the option of taking line *BUS1* is represented in red.

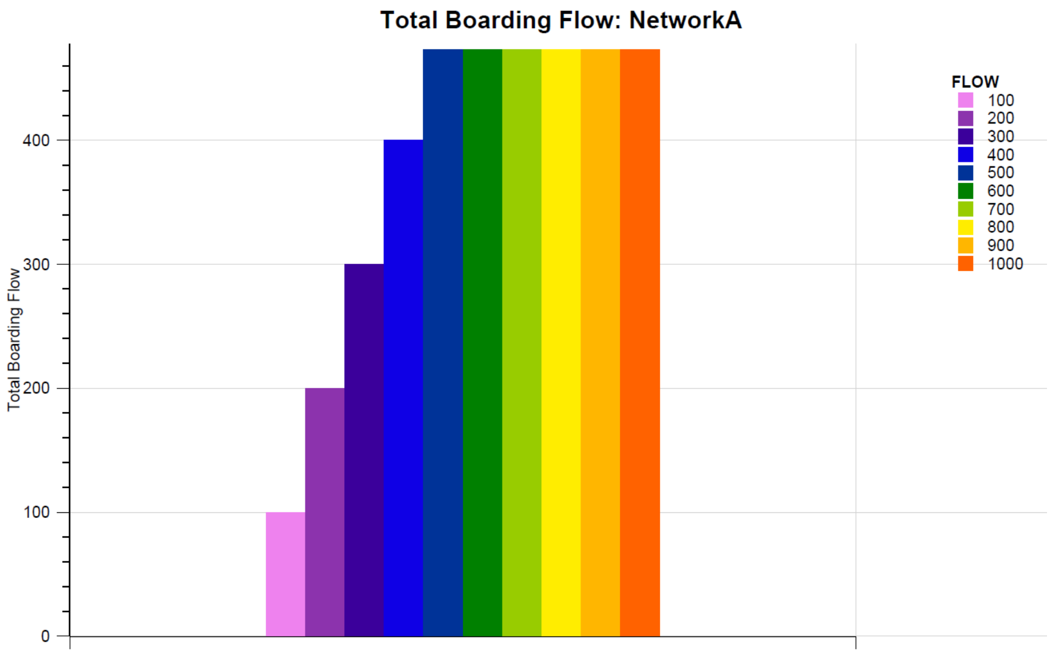

FIGURE 5.7: On-board Overcrowding Congestion test. Expected total boarding flow of line *BUS1* at equilibrium.

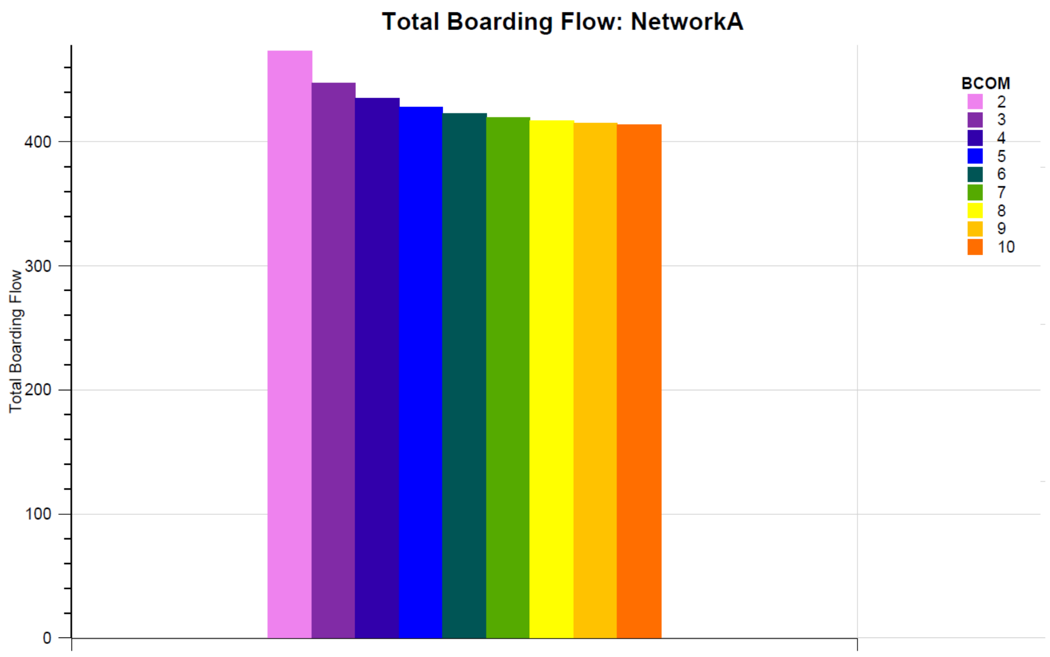

FIGURE 5.8: On-board Overcrowding Congestion test. Expected total boarding flow of line *BUS1* at equilibrium for varying $\beta^{v-crowd}$.

### 5.1.2 Test - Platform Overcrowding Congestion

As stated before, platform overcrowding congestion is tested on *NetworkA*, composed of one line serving two stops. As in on-board overcrowding congestion testing, many simulations were conducted, varying the travel demand (going from node *1* to node *2*) from 100 to 1000 passengers per hour, with steps of 100 passengers per hour.
In these simulations, the crowding BPR multiplier $\alpha^{crowd}$ was set equal to 1, while the platform crowding BPR exponent $\beta^{p-crowd}$ was set equal to 2. Moreover, to reproduce the congestion phenomenon related to platform crowding, the platform capacity $k_s^{pax}$ was set equal to 10 passengers. Finally, RGAP equal to $10^{-9}$ was again adopted as stop criterion.

Figure 5.9 illustrates the trend of expected cost for stand running arc (similarly to figure 5.4), while figure 5.10 shows the trend of RGAP for the different travel demands (similarly to figure 5.4). Both trends are plotted with respect to the assignment algorithm iterations. Trends for different travel demand values are shown in different colours.
From these pictures, growing travel demand implies growing costs, as consequence of congestion, and that there is a threshold below which the equilibrium is found in two iterations (as in the graphs related to on-board overcrowding congestion).
In these specific simulations, the threshold is travel demand equal to 800 passengers per hour, which means that for demand flows higher than 700 passengers per hour the walking option becomes competitive with the one of boarding the line *BUS1* and the algorithm needs more iterations to converge.
Moreover, figure 5.5 shows that, with the stop criterion imposed as RGAP equal to $10^{-4}$, maximum 7 iterations are needed by the algorithm, against the 16 necessaries with stop criterion set to RGAP equal to $10^{-9}$).

Figure 5.11 shows the equilibrium of expected hyperpath costs during the assignment algorithm when the travel demand is equal to 1000 passengers per hour (similarly to figure 5.6). More precisely, the violet one for the walking hyperpath and the red one for the transit line *BUS1* hyperpath.
As in the on-board crowding congestion case, no diversion node is present in the graph (there are only one line and no seating arcs) and the equilibrium is between path costs and not hyperpaths.
Moreover, also in this case, it can be observed that the walking option is part of the solution only after the second iteration.

Figure 5.12 shows the flow of passengers boarding the transit line *BUS1* for each demand flow, where each column is a different demand flow. From this picture, it can be observed that maximum 748 passengers per hour are assigned the transit line *BUS1*. Thus, all the exceeding passengers choose the walking option.
Obviously, these results strictly depend on platform capacity and BPR parameters. For example, changing the value of $\beta^{p-crowd}$ from 2 to 4 varies reduces the maximum number of boarding passengers to 387 passengers per hour.

Comparing these results with the ones of the on-board overcrowding test (section 5.1.1), it can be observed that platform congestion is less relevant to users' decisions than on-board crowding. Indeed, the flow demand after which congestion influences passengers' choices is significantly higher for the platform case than for the on-board case (800 passengers per hour with platform capacity of 10 passengers against 500 passengers per hour with standing vehicle capacity of 40 passengers).
Moreover, if the platform capacity is imposed of the same value of the vehicle standing capacity used in section 5.1.1 (40 passengers), the congestion due to platform overcrowding becomes relevant only for demand flows higher than 2992 passengers per hour. This is because the BPR platform overcrowding congestion term is multiplied for the waiting time, while the BPR on-board overcrowding congestion term is multiplied by the running time, and that waiting time is significantly smaller than running time (360 seconds against 900 seconds).

Finally, as in the case of on-board congestion, additional simulations were conducted for travel demand equal to 1000 passengers per hour and for varying platform crowding BPR exponent $\beta^{p-crowd}$ (from 2 to 10). The purpose of these simulations was studying more in detail how the algorithm performs when platform overcrowding congestion is activated.
Figure 5.13 shows the expected flow boarding the transit service, where each column represents different value of $\beta^{p-crowd}$. It can be observed that, for growing $\beta^{p-crowd}$, the number of passengers choosing the transit line *BUS1* decreases. This is a consequence of the increase of waiting expected costs due to congestion. A similar conclusion can be observed by varying the crowding BPR multiplier $\alpha^{crowd}$.

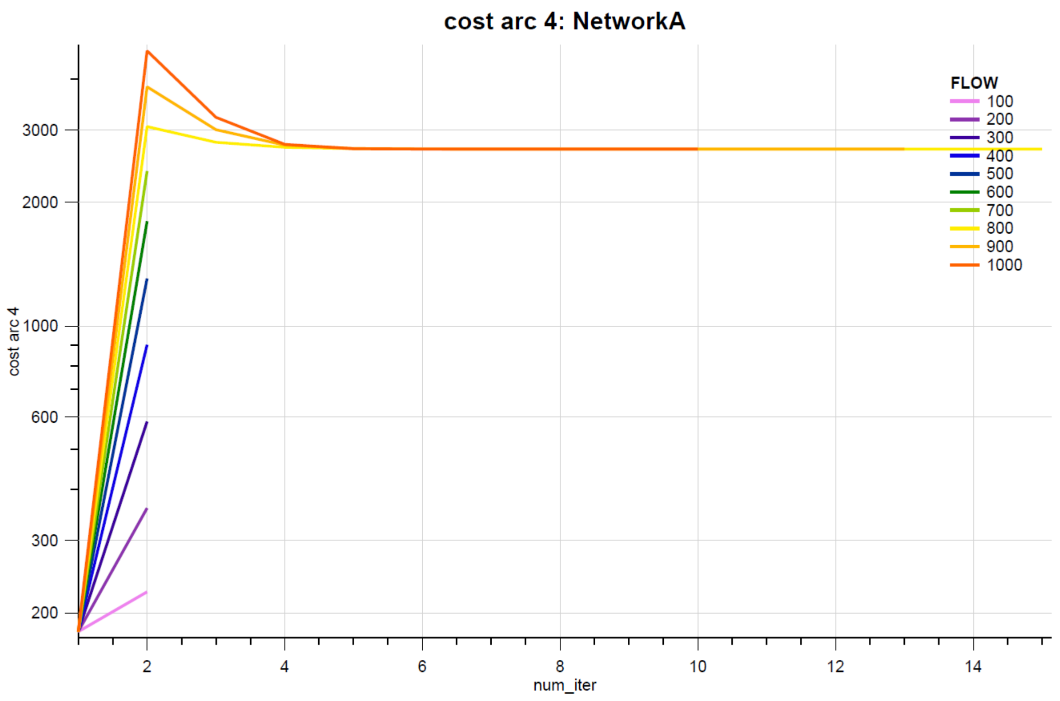

FIGURE 5.9: Platform Overcrowding Congestion test. Expected cost of waiting arc.

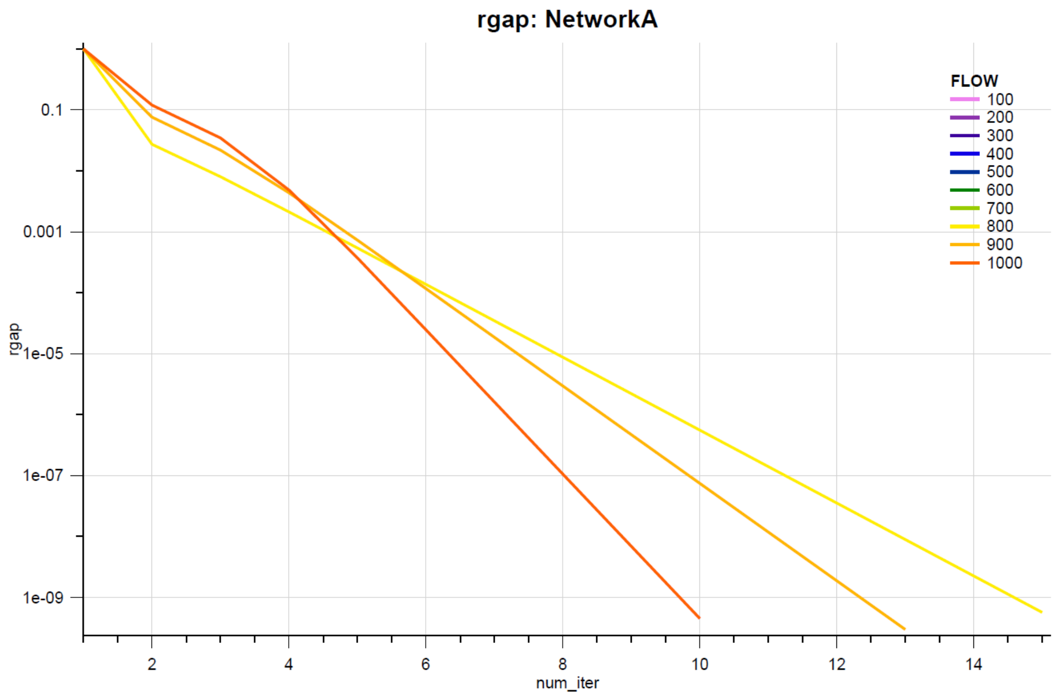

FIGURE 5.10: Platform Overcrowding Congestion test. RGAP of the solution algorithm for $\beta^{p-crowd} = 2$.

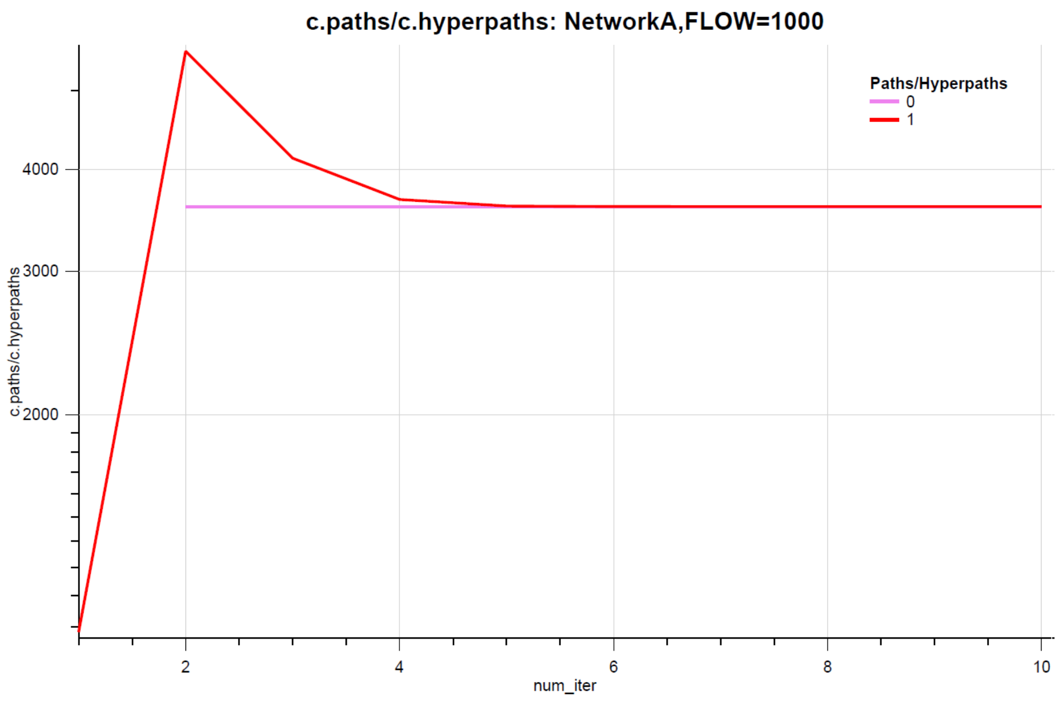

FIGURE 5.11: Platform Overcrowding Congestion test. Hyperpath cost equilibrium: The walking option is represented in violet, while the option of taking line *BUS1* is represented in red.

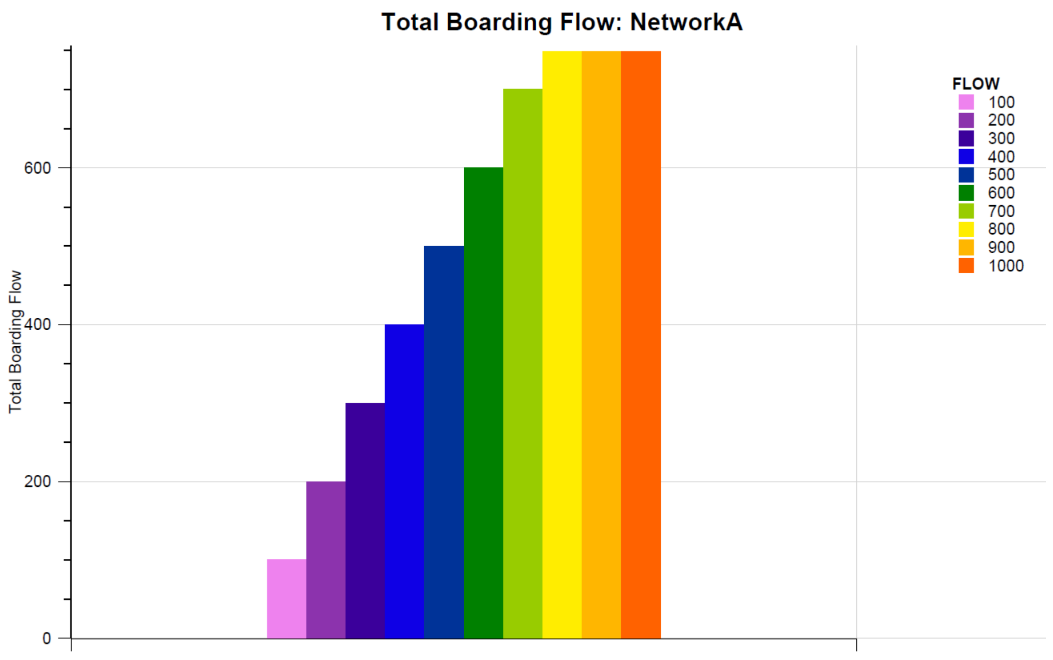

FIGURE 5.12: Platform Overcrowding Congestion test. Expected total boarding flow of line *BUS1* at equilibrium.

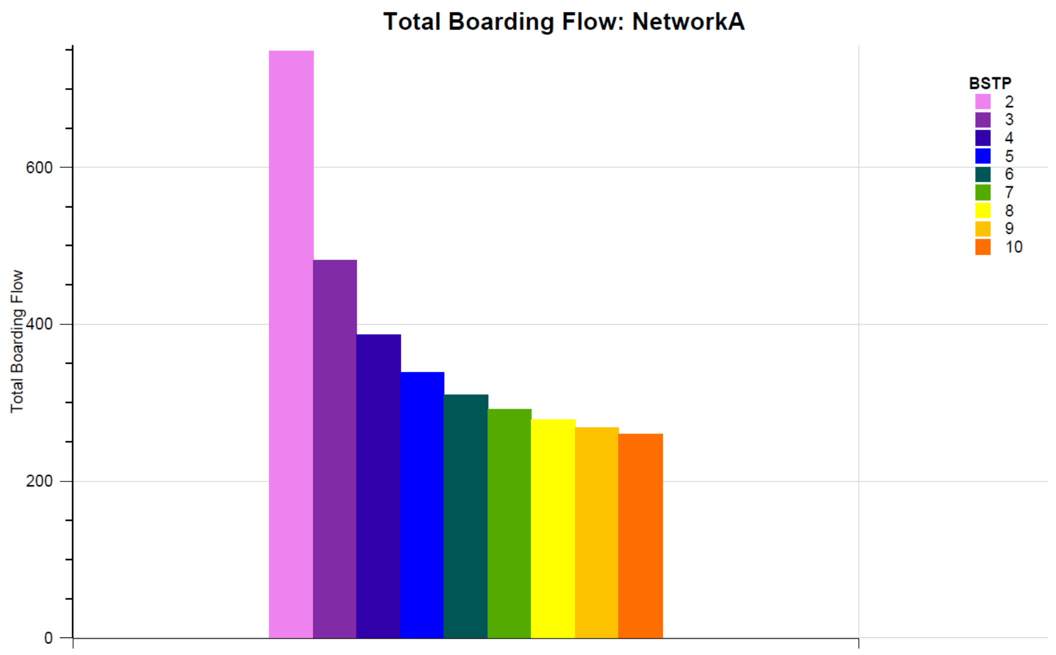

FIGURE 5.13: Platform Overcrowding Congestion test. Expected total boarding flow of line *BUS1* at equilibrium for varying $\beta^{p-crowd}$.

### 5.1.3 Test - Queuing Congestion

As stated before, queuing congestion is tested on *NetworkA*, composed of one line serving two stops. As for on-board and platform overcrowding congestion testing, many simulations were conducted, varying the travel demand (going from node *1* to node 2) from 100 to 1000 passengers per hour, with steps of 100 passengers per hour.
In these simulations, the standing vehicle capacity $k_l^{stand}$ was set equal to 40 passengers (which means 400 passengers per hour, being the expected headway equal to 6 minutes), as in the on-board overcrowding case of section 5.1.1. Moreover, RGAP equal to $10^{-9}$ was again adopted as stop criterion.

As introduced in section 3.3.2 Queuing Congestion, and summarized by equation 3.24, two kinds of queuing congestion are implemented in the algorithm:

- soft capacity constraints reproduced with the effective frequency method;
- strict capacity constraints reproduced with the fail-to-board probability method.

To simulate the first method, the queuing BPR multiplier $\alpha^{queue}$ was set equal to 1, while the platform crowding BPR exponent $\beta^{queue}$ was set equal to 4. On the other hand, to study queuing congestion with the second method, the platform crowding BPR exponent $\beta^{queue}$ was set equal to 0 and the risk adverseness coefficient $\gamma^{risk}$ was set equal to 1.

Figure 5.14 and 5.15 represent the trend of RGAP during the assignment algorithm, respectively for the effective-frequency and the fail-to-board probability method. Trends for different travel demand values are shown in different colours.

As it can be observed, in case of $\beta^{queue} = 4$ the algorithm converges in a way like the over-crowding congestion simulations (figure 5.5 and 5.10), where a threshold exists after which more than two iterations are necessary to find the equilibrium. In this test, the threshold is the demand equal to 700 passengers per hour.
For what concerns the case of $\beta^{queue} = 0$, figure 5.15 shows that the algorithm does not converge, and it is stopped when the maximum number of iterations is reached. This is because, when the fail-to-board probability is considered, some of the demand flow is removed from the network (and theoretically moved to next temporal layer) and, consequently, flow conservation does not hold anymore.

With regards to the expected cost of the waiting arc, its dependence from the demand flow can be seen in figure 5.16 for $\beta^{queue} = 4$ and figure 5.17 for $\beta^{queue} = 0$, which show the expected cost of the waiting arc during the assignment algorithm. Trends for different travel demand values are shown in different colours.
Again, it can be observed as the expected cost of waiting arc increases with the amount of demand, and that the algorithm does not converge in the fail-to-board probability method.

Finally, figure 5.18 and 5.19 show respectively the amount of flow boarding the line and the amount of flow reaching destination for $\beta^{queue} = 4$, while figure 5.20 and 5.21 show the same for $\beta^{queue} = 0$. Each column represents a different demand flow.
It can be observed that, when the effective frequency method is applied, passengers either board the vehicle or walk, while, when the fail-to-board probability method is considered, some of the flow disappear from the network. Moreover, in the effective frequency case, the maximum number of passengers choosing to board the transit line *BUS1* is 642 passengers per hour, while in the fail-to-board case with risk adverseness coefficient $\gamma^{risk}$ equal to 1 all passengers choose to board the transit line *BUS1* (even if not all passengers are able to board during the assignment period).

To study more in detail how the algorithm performs when the fail-to-board probability method is applied, additional simulations were conducted for travel demand equal to 1000 passengers per hour and for varying risk adverseness coefficient (from 1 to 10).
Figure 5.22 shows the RGAP trend during the assignment algorithm for this case. Trends for different travel demand values are shown in different colours.
From this picture, it can be observed that growing risk adverseness coefficient implies better convergence.
Figure 5.23 represents the expected flow reaching the destination in the assignment period, where each column is a different value of the risk adverseness coefficient.
For $\gamma^{risk}$ larger than 3, some passengers start choosing the walking option instead of waiting the "*next temporal layer*" and, thus, the number of flow reaching the destination in the assignment period increase.

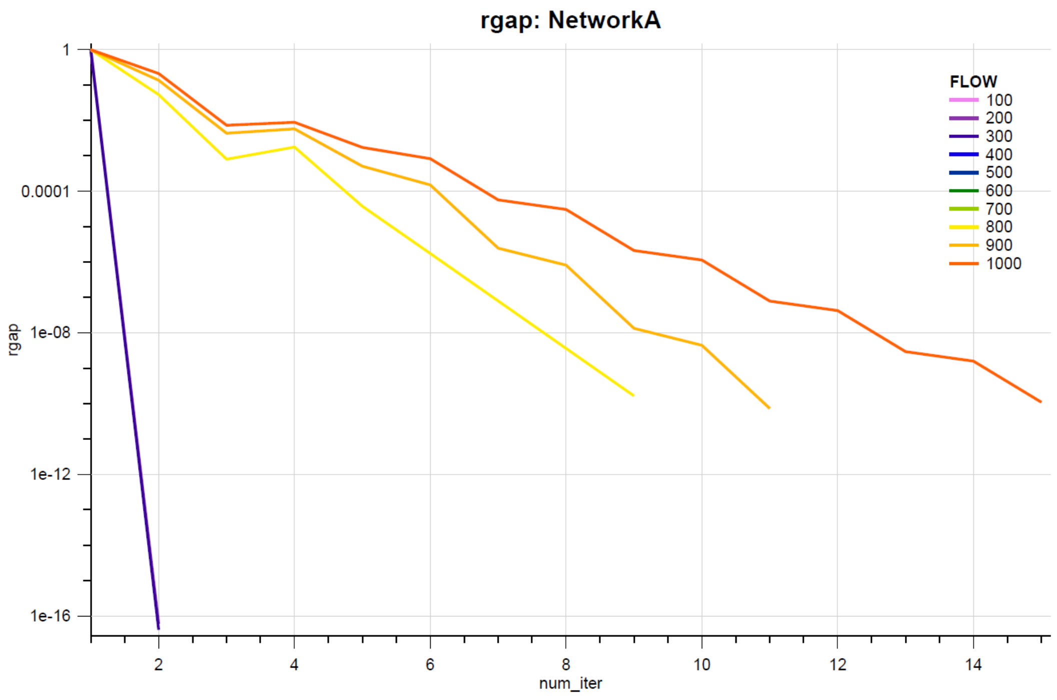

FIGURE 5.14: Queuing Congestion test. Expected total boarding flow of line *BUS1* for $\beta^{queue} = 4$.

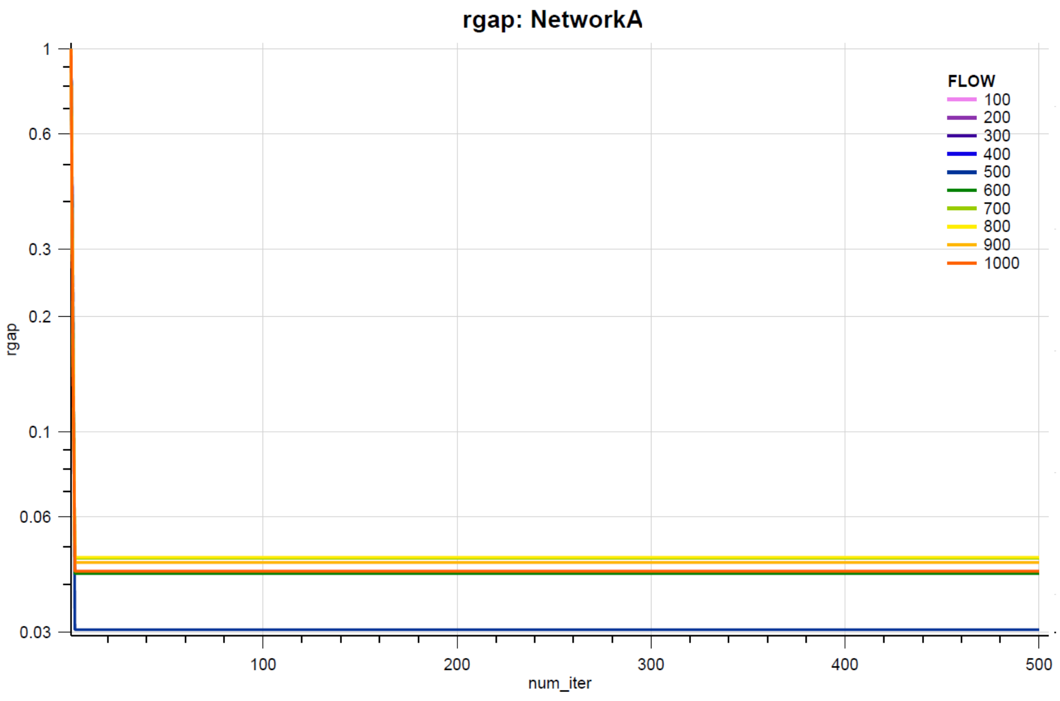

FIGURE 5.15: Queuing Congestion test. Expected total boarding flow of line *BUS1* for $\beta^{queue} = 0$.

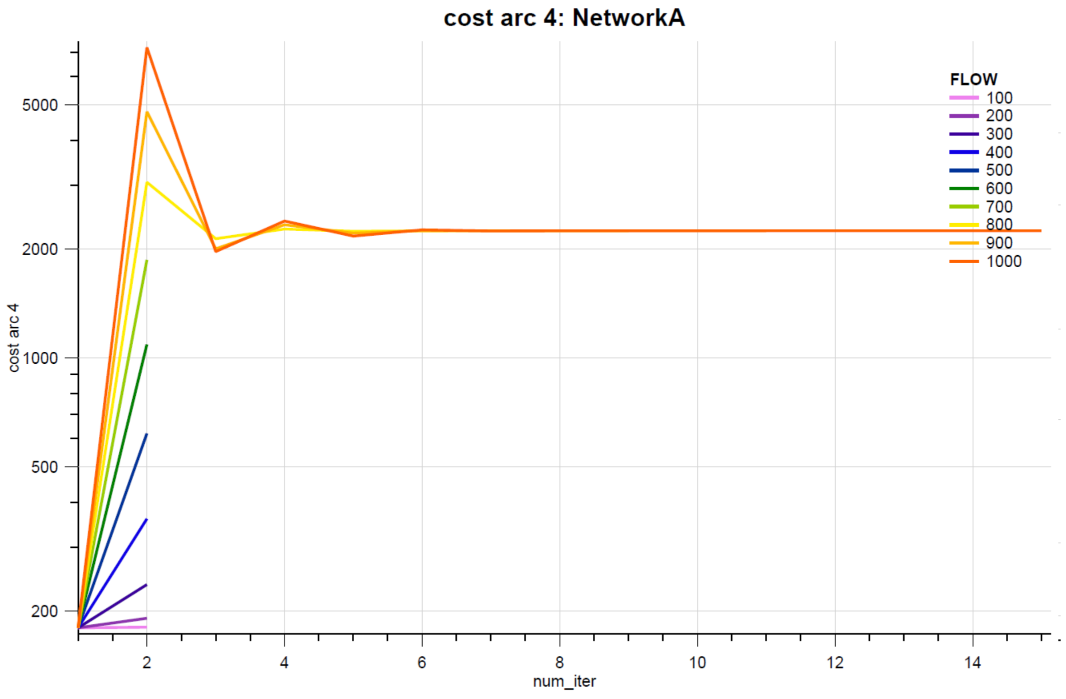

FIGURE 5.16: Queuing Congestion test. Expected cost of waiting arc for $\beta^{queue} = 4$.

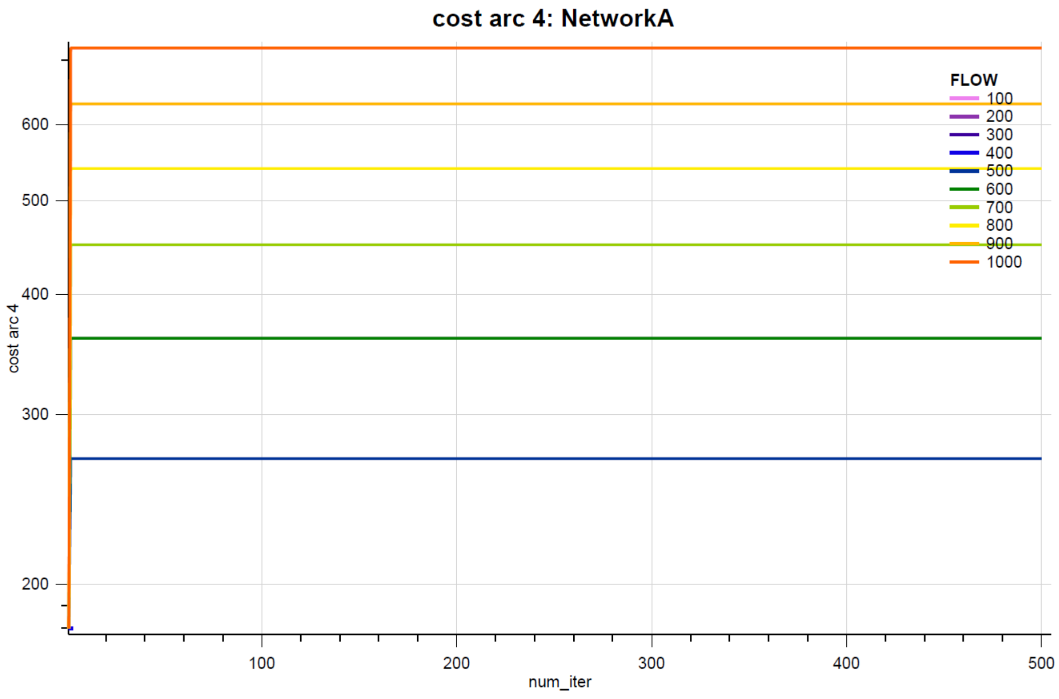

FIGURE 5.17: Queuing Congestion test. Expected cost of waiting arc for $\beta^{queue} = 0$.

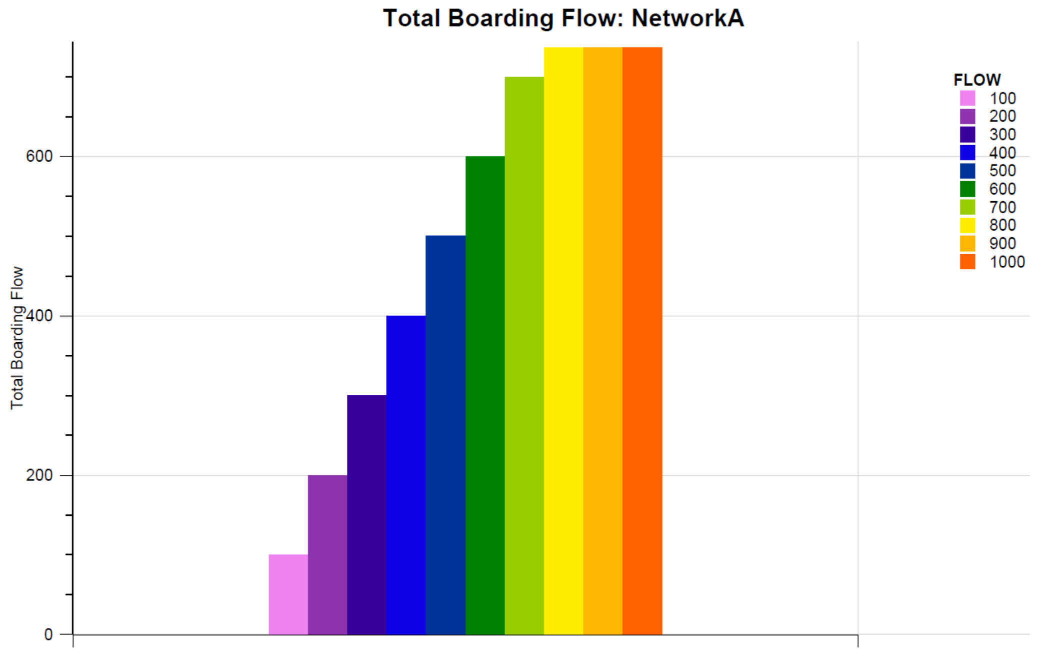

FIGURE 5.18: Queuing Congestion test. Expected total boarding flow of line *BUS1* for $\beta^{queue} = 4$.

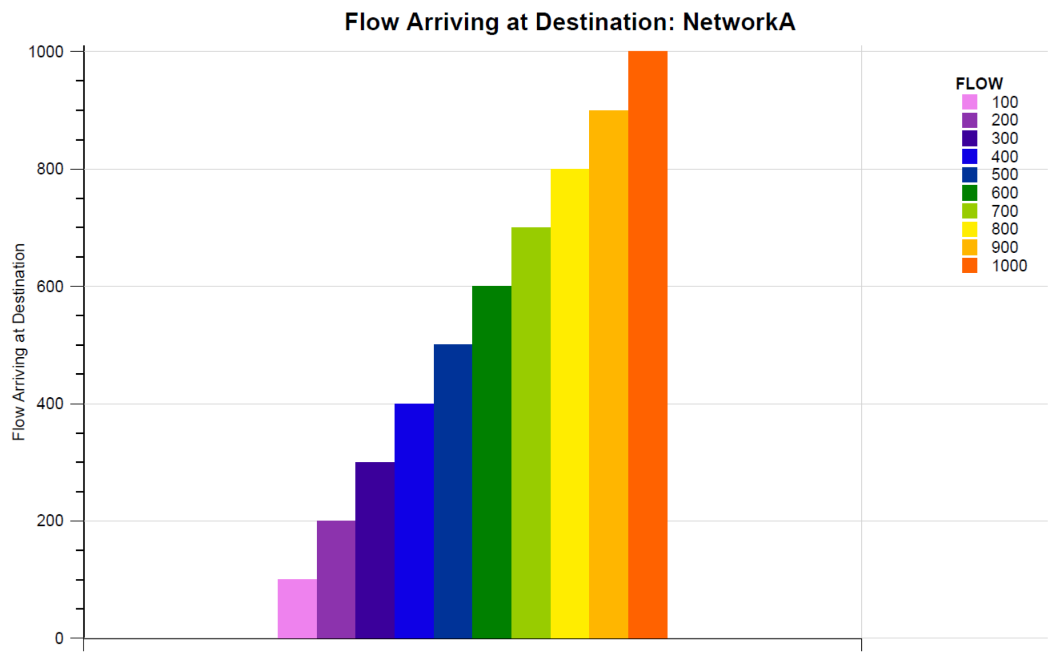

FIGURE 5.19: Queuing Congestion test. Expected flow reaching the destination for $\beta^{queue} = 4$.

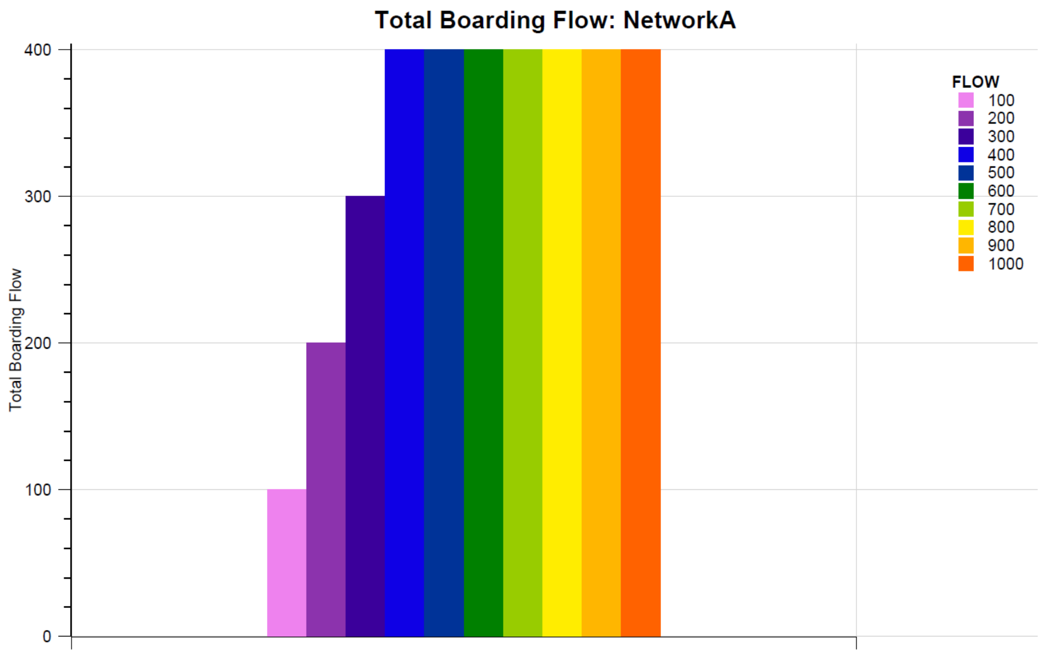

FIGURE 5.20: Queuing Congestion test. Expected total boarding flow of line *BUS1* for $\beta^{queue} = 0$.

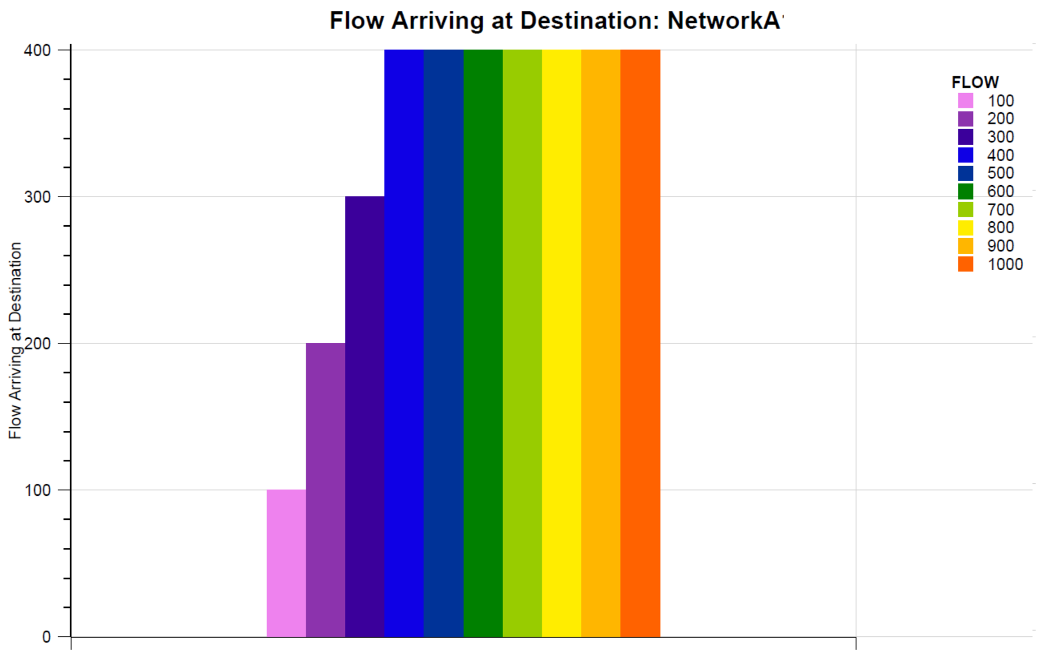

FIGURE 5.21: Queuing Congestion test. Expected flow reaching the destination in the assignment period for $\beta^{queue} = 0$.

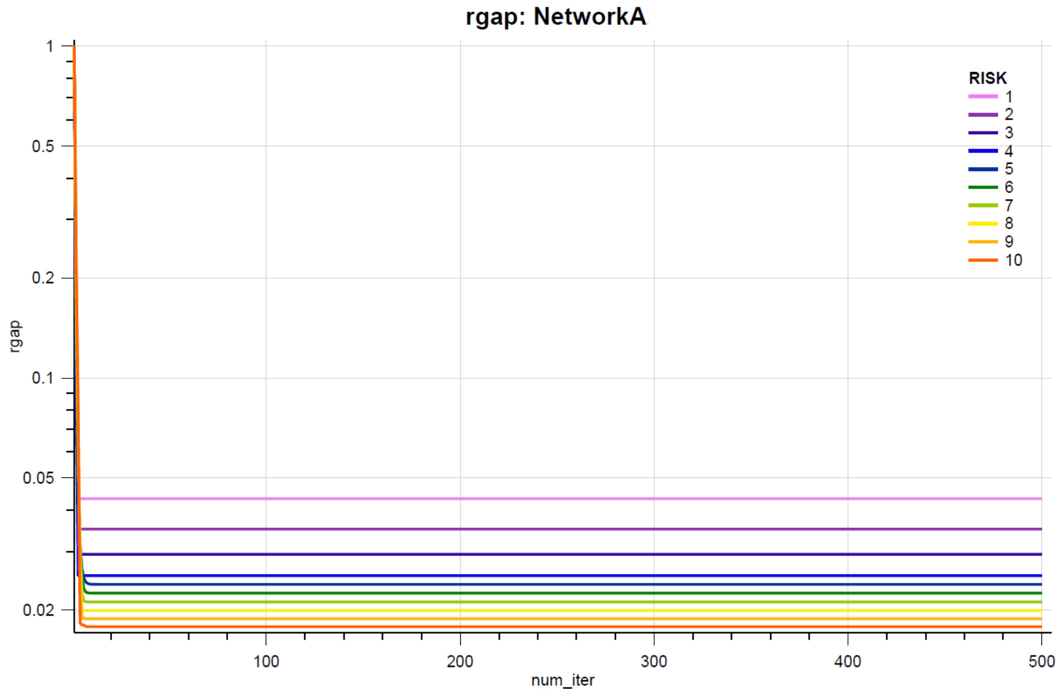

FIGURE 5.22: Queuing Congestion test. RGAP of the solution algorithm for $\beta^{queue} = 0$ and varying $\gamma^{risk}$.

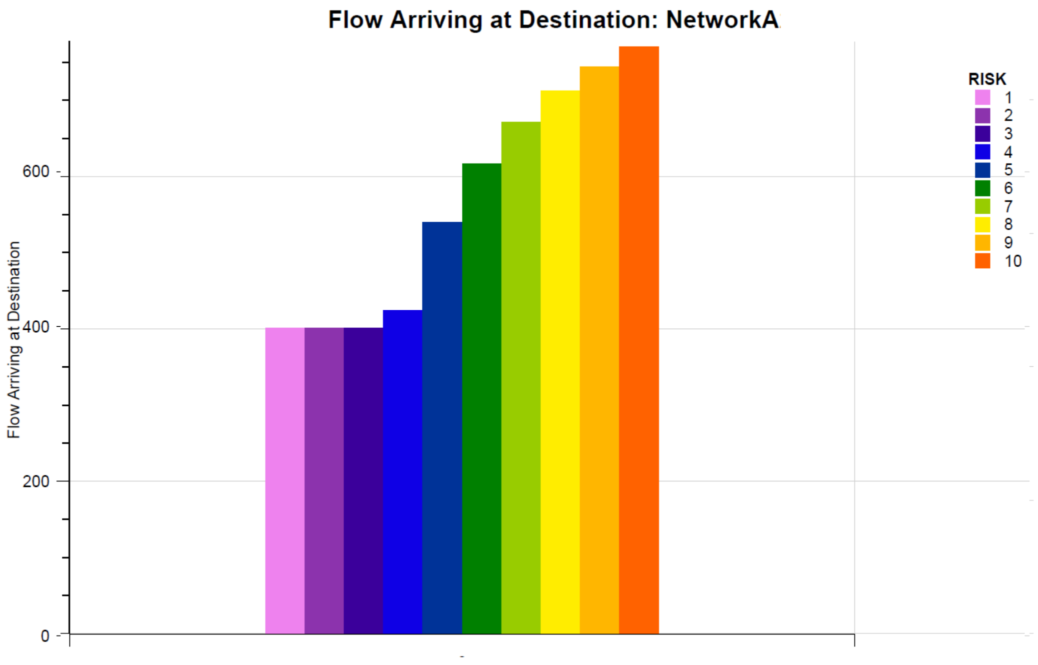

FIGURE 5.23: Queuing Congestion test. Expected flow reaching the destination in the assignment period for $\beta^{queue} = 0$ and varying $\gamma^{risk}$.

### 5.1.4 Test - Dwelling Delay

As stated before, dwelling delay due to congestion is tested on *NetworkC*, composed of one line serving two stops. To this purpose, many simulations were conducted, each of them with different amount of travel demand.
It must be stressed out that this congestion phenomenon is not separable, as it depends both on dwelling and boarding/alighting flows, thus the phenomena cannot be isolated as the ones studied in previous sections.

As the studied congestion phenomenon is related to the vehicle overcrowding, line seating capacity was set equal to 0 passengers per hour and line standing capacity was set equal to 40 passengers per hour (which means 400 passengers per hour, being the expected headway of 6 minutes).
Furthermore, the congestion factor $BPR^{dwell}\left(q^{dwell-stand}\right)$ is introduced in cost computation by multiplying it per the passenger alighting and boarding time $t_a^{dwell}\left(q^{alight}, q^{board}\right)$ due to door finite capacity, introduced in equation 3.17.
In these simulations it was decided to focus on the effects of dwelling congestion related to boarding passengers, thus the door boarding capacity $K_l^{board}$ was limited to 5 passengers per minute. Studying the effects of dwelling congestion related to alighting passengers gives similar results (thus, tests are not described in this thesis).
To study the boarding flow effects, two OD pairs were defined: the first one going from node *1* to node *3* and varied from 100 to 1000 passengers per hour (with steps of 100 passengers per hour), and the second one going from node *2* to node *3* and assumed to get the constant value of 500 passengers per hour.
In these simulations, the dwelling BPR multiplier $\alpha^{dwell}$ was set equal to 1, while the dwelling BPR exponent $\beta^{dwell}$ was set equal to 2.

Figure 5.24 shows the trend of RGAP during the assignment algorithm. Trends for different travel demand values are shown in different colours.
As in all the previous cases involving congestion, it can be observed that there is a threshold after which more than two iterations are needed to find the equilibrium. In this case, the threshold is demand going from node *1* to node *2* equal to 600 passengers per hour.

For what concerns the expected cost of dwelling arc, as well as the expected cost of the placing arc of stop *2*, they are respectively represented in figure 5.25 and 5.26, where trends for different travel demand values are shown in different colours.
From these pictures, it can be observed that the expected costs of these arcs have the same trend. This is since boarding fee is assumed to be null and to the assumptions made in section 4.4 UE Algorithm, where some dwelling cost was added to the placing cost to disincentive the bypass of dwelling arcs.

Finally, figure 5.27 and 5.28 represent the expected total boarding flow of transit line *BUS1*, respectively at stop *1* and at stop *2*. Each column represents a different demand flow.
From these pictures, it can be observed that at stop *1* all passengers board, no matter the congestion.

On the other hand, at stop 2 all passengers choose to board the transit line *BUS1* until a certain value of passengers is present on-board. After this threshold, that is 1100 passengers, some of the network users' start walking to the destination.
Obviously, this threshold strictly depends on BPR coefficient, door boarding capacity and cost of other links.

Noteworthy, the dwelling congestion, in this case, does not influence the number of users already on-board, which can choose between waiting in the dwelling vehicle or alighting at the stop and walking, but it changes the decision of passengers that have still to board the line.
The same happens when the dwelling congestion linked to alighting passengers is studied. In this case, passengers choose to not board the line instead of alighting during the journey and continue walking.
These results are a direct consequence of the assumptions made in section 4.4 UE Algorithm, which increase the cost of alighting and boarding together with the cost of dwelling.

To study more in detail the effect of vehicle overcrowding on dwelling congestion, additional simulations were conducted, assuming both travel demands equal to 1000 passengers per hour and varying the dwelling BPR exponent $\beta^{dwell}$ (from 2 to 10).
For parameters larger than 5, the algorithm becomes in-stable and equilibrium with stop criterion RGAP equal to $10^{-9}$ is not reached in the maximum number of iterations. Thus, the stop criterion was made less restrictive by imposing the minimum RGAP at equilibrium equal to $10^{-4}$.
Figure 5.29 shows the RGAP trend during the assignment algorithm in this case, where different colours represent different $\beta^{dwell}$. Furthermore, figure 5.30 represents the expected flow boarding transit line *BUS1* at stop 2, where each column is a different value of $\beta^{dwell}$.
From these pictures it can be observed that the algorithm gets highly in-stable for growing $\beta^{dwell}$ and that growing congestion means more passengers choosing to walk instead of boarding the line.
Moreover, it can be observed that congestion starts having consequences on the choices of already on-board passengers after a certain value of $\beta^{dwell}$. More precisely, some of them start alighting at stop 2, as shown by figure 5.31, where each column represents a different $\beta^{dwell}$.

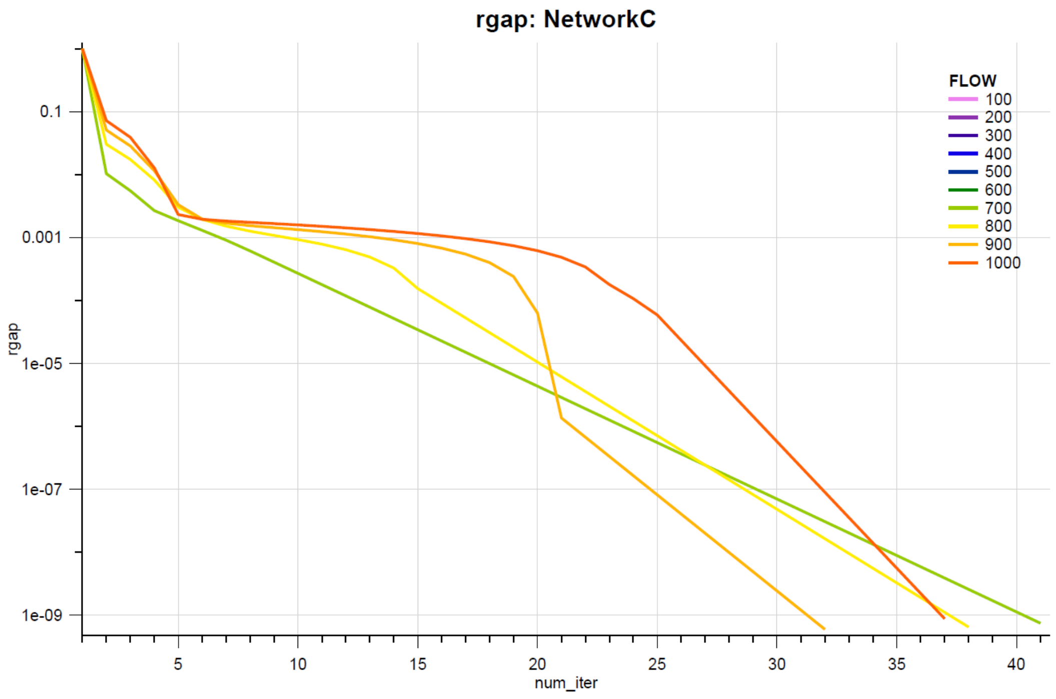

FIGURE 5.24: Dwelling Delay test. RGAP of the solution algorithm.

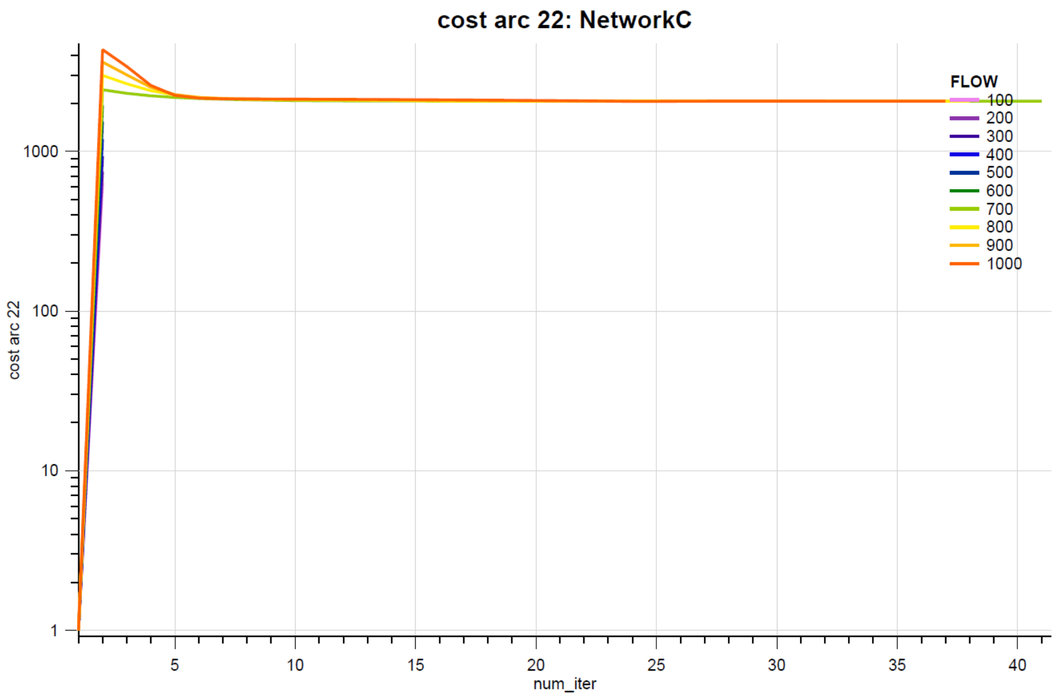

FIGURE 5.25: Dwelling Delay test. Expected cost of stand dwelling arc.

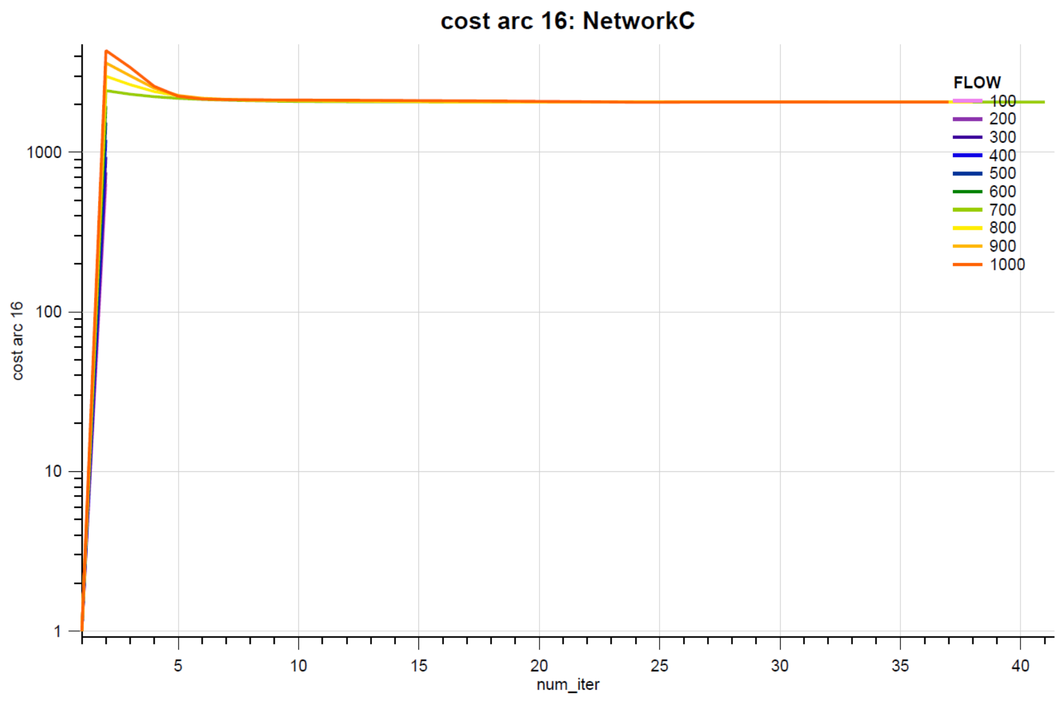

FIGURE 5.26: Dwelling Delay test. Expected cost of stand placing arc.

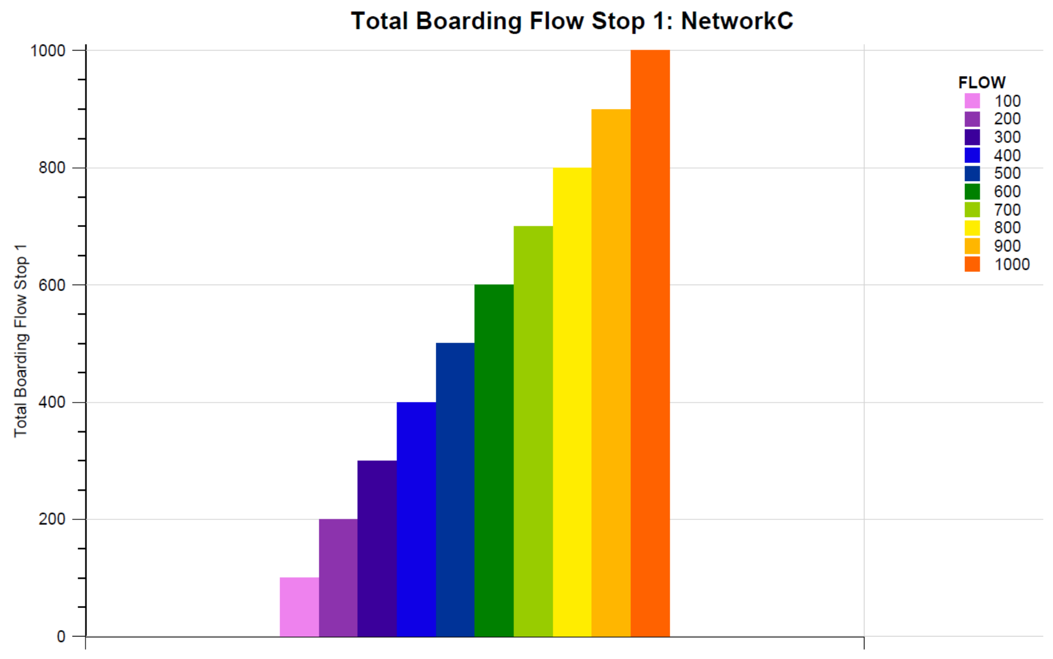

FIGURE 5.27: Dwelling Delay test. Expected total boarding flow of line *BUS1* at stop *1* at equilibrium.

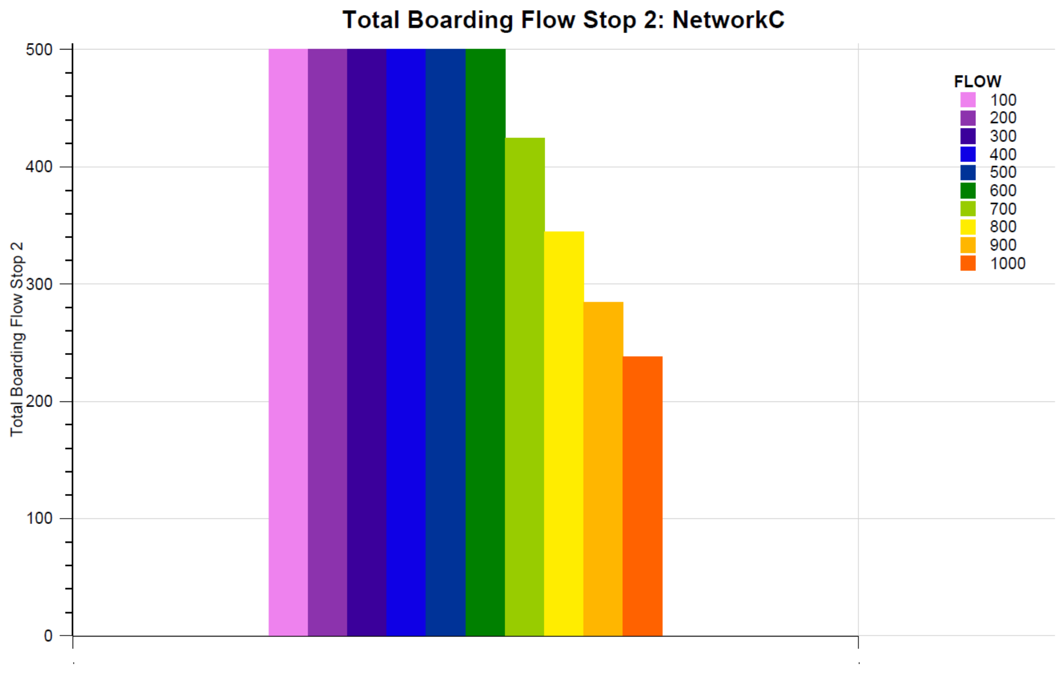

FIGURE 5.28: Dwelling Delay test. Expected total boarding flow of line *BUS1* at stop *2* at equilibrium.

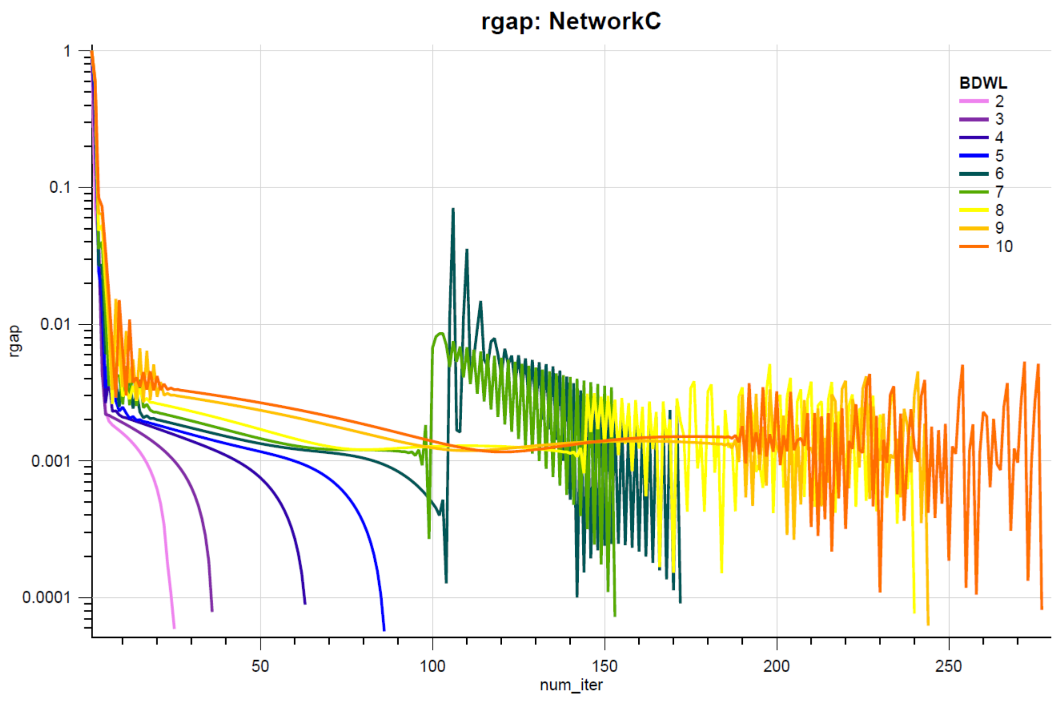

FIGURE 5.29: Dwelling Delay test. RGAP of the solution algorithm for varying $\beta^{dwell}$.

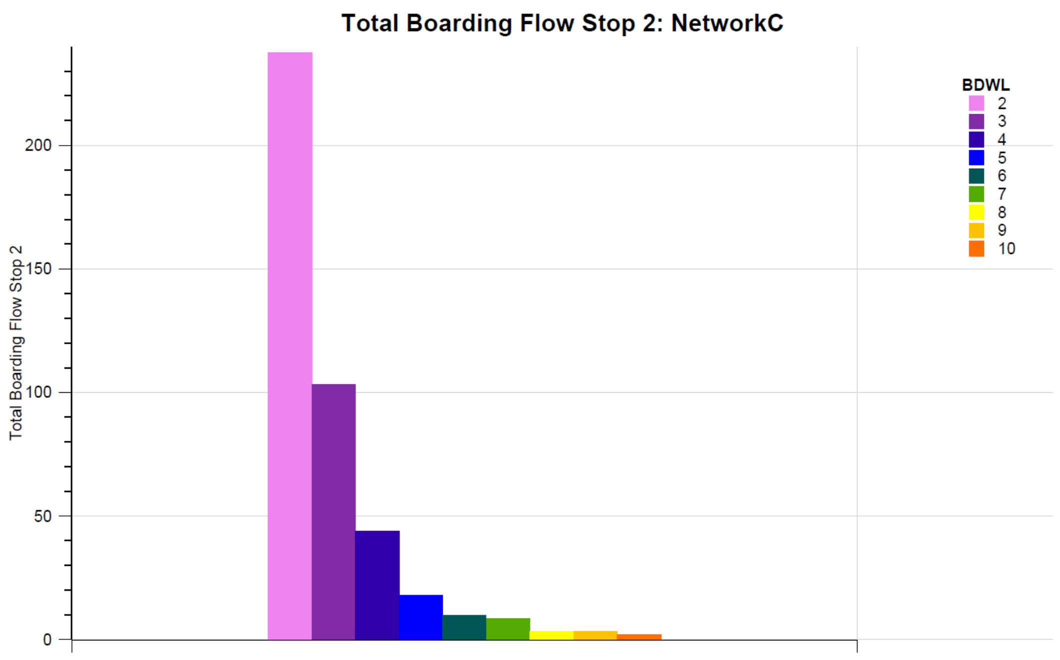

FIGURE 5.30: Dwelling Delay test. Expected total boarding flow of line *BUS1* at stop 2 at equilibrium.

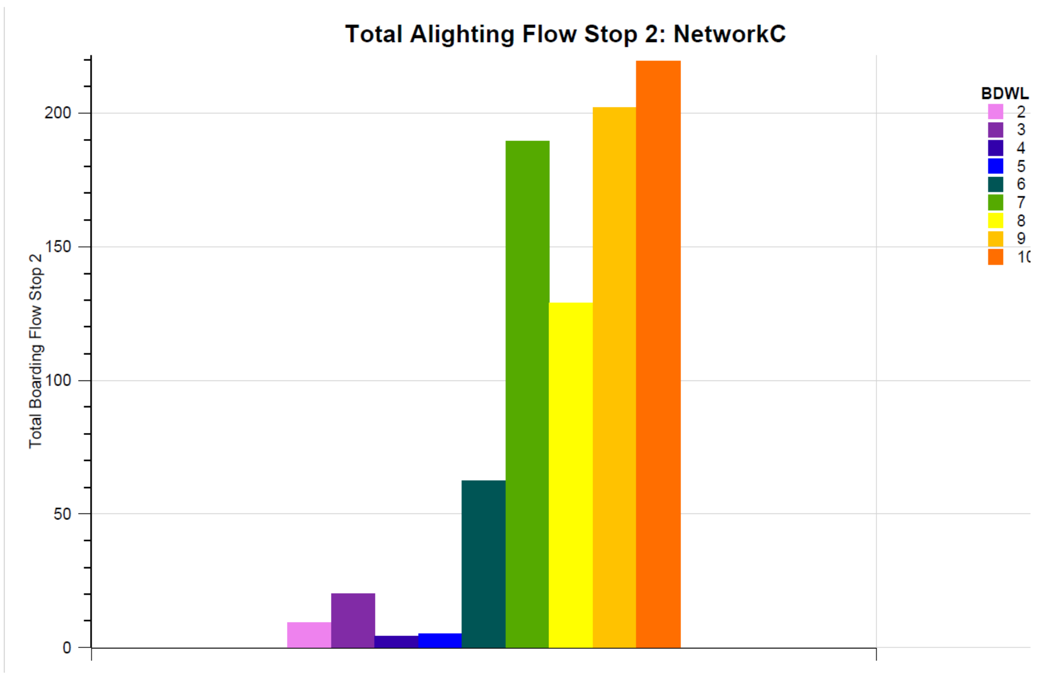

FIGURE 5.31: Dwelling Delay test. Expected total alighting flow of line *BUS1* at stop 2 at equilibrium.

### 5.1.5 Test - Seating Hyperarcs

As stated introduced before, to test the phenomena related to the sitting probability, two different networks were adopted:

- *NetworkA*, composed of one line serving two stops, for the placing hyperarc test;
- *NetworkC*, composed of one line serving three stops, for the dwelling hyperarc test.

All the congestion phenomena not related to seating hyperarcs (eg. on-board, at stops, queuing, dwelling delay) were deactivated. Moreover, RGAP equal to $10^{-9}$ was adopted as stop criterion.

For what concerns the placing hyperarc test, several simulations were carried out, each of them with different amount of travel demand. More precisely, one OD pair going from node *1* to node *2* was introduced and its travel demand was varied from 100 to 1000 passengers per hour (with steps of 100 passengers per hour). Moreover, to isolate the placing hyperarcs phenomena from the phenomena related to the other types of congestion (eg. dwelling delay, overcrowding, etc.), the door capacity was set equal to infinity and all the BPR coefficients were set equal to zero, except for $\beta^{queue}$ that was set higher than zero to perform the assignment with the effective frequency method.
Finally, seating and standing capacity for vehicles of line *BUS1* were set equal to 20 passengers per vehicle (which means 200 passengers per hour, being the expected headway of 6 minutes).

With regards to the dwelling hyperarc test, the same approach, with respect to capacities and BPR coefficients, was adopted. However, in this case, it was not possible to fully isolate the phenomena related to dwelling hyperarcs as having both seating and standing capacity higher than zero activates also the placing hyperarcs.
Again, many simulations were conducted assuming two different OD pairs. The first one, going from node *1* to node *2*, was varied from 100 to 1000 passengers per hour (with steps of 100 passengers per hour), while the second, going from node *1* to node *3*, was kept constant (400 passengers per hour).

Figure 5.32 and 5.33 show the trend of RGAP during the assignment algorithm, respectively for placing and dwelling hyperarc. Trends for different travel demand values are shown in different colours.
From these pictures, it can be observed that when the demand is higher than the seating vehicle capacity (i.e. demand higher than 200 passengers per hour), the algorithm needs at least three iterations to find the equilibrium in case of seating hyperarcs activated. Indeed, the algorithm firstly performs an *"all-or-nothing"* assignment to the shortest hypertree and then corrects the flow distribution according to the new conditional probabilities computed according to the congested costs. If only one seating phenomenon is involved in the equilibrium computation (i.e. placing hyperarc test, figure 5.32), the third iteration is sufficient to check the convergence, otherwise more iterations are necessaries as different combinations of flows can decrease the costs (i.e. dwelling hyperarc test, figure 5.33).

To represents the algorithm behaviour in case of only three iterations (i.e. placing hyperarcs), figure 5.36 and 5.37, representing the algorithm behaviour in case of demand equal to 1000 passengers per hour, were included in this paper. More precisely, figure 5.36 shows how passengers flow assignment changes (during the assignment algorithm), where the violet and red trends represent respectively the passengers sitting and standing on-board of line *BUS1*, while figure 5.37 shows the hyperarcs to which demand flows are assigned (during the assignment algorithm).
From figure 5.36, at first iteration all flows are assigned to the seating arc (violet trend), while at second iterations the seating share is adjusted considering seating vehicle capacity and, consequently, assigning some flow to the standing arc (red trend). Moreover, from figure 5.37 it can be observed that all demand is assigned to the hyperarc representing line *BUS1*. This is because of the non-presence of restrictions on standing vehicle capacity, as either soft constraints or strict constraints are not enforced.

Figure 5.34 and 5.35 represent the sitting probability for each travel demand, respectively at the board placing node of *NetworkA* and at the stand arrival placing node of *NetworkC*. Each demand flow is represented by a different column.
From figure 5.34 it can be observed that for values of demand flows lower than the seating vehicle capacity (200 passengers per hour), the sitting probability is equal to 1. Then, it decreases rapidly with the demand growth.
For what concerns the sitting probability of dwelling hyperarcs, it increases with growing demand. This is because of, when the demand going from node *1* to node *2* increases, less seating capacity is assigned to the demand going from node *1* to node *3*, as some of the capacity is assigned to passengers of the first OD pair. This means that, when passengers alight at stop *2*, a larger number of seats get available and, thus, more passengers dwelling can sit. However, it can also be observed that this increase in number of available seats depends on the probabilistic way of assigning seats (at the placing hyperarc). Indeed, higher the demand for an OD pair is, higher is its sitting probability at the placing hyperarcs.

To study more in detail the effect of seating vehicle capacity $k_{BUS1}^{seat}$, additional simulations were conducted, varying this capacity value from 10 to 100 passengers per vehicle (i.e. 100 to 1000 passengers per hour, being the expected headway equal to 6 minutes). The placing hyperarc test was carried out over *NetworkA*, while the dwelling hyperarc test was carried out over *NetworkC*.

For what concerns the placing hyperarc test, one OD pair demand going from node *1* to node *2* was introduced and kept constant to the value of 1000 passengers per hour. Figure 5.38 shows the RGAP trend during the assignment algorithm of this test (where different colours represent different value of $k_{BUS1}^{seat}$), while figure 5.39 represents the sitting probability at the boar placing diversion node of *NetworkA* (where each column is a different value of $k_{BUS1}^{seat}$). As expected, it can be seen again from these pictures that only three iterations are needed to find the equilibrium and that growing seating capacity means larger sitting probability.

Finally, with respect to the dwelling hyperarc test, the same OD pairs of the test carried out in previous lines were adopted. In these simulations, the demand flow of first OD pair, going

from node *1* to node *2*, was kept constant to 1000 passengers per hour, while the demand flow of the second OD pair, going from node *1* to node *3*, was kept constant to 400 passengers per hour.
Figure 5.40 and 5.41 represent the equivalent figure 5.38 and 5.39 illustrates for the dwelling hyperarc test. Form these pictures it can be observed that growing seating capacity means faster algorithm, as fewer iterations are needed to find the equilibrium. More precisely, for seating capacity higher than 30 passengers per vehicle, the sitting probability is 1, which means that all passengers dwelling at stop *2* find a seat.

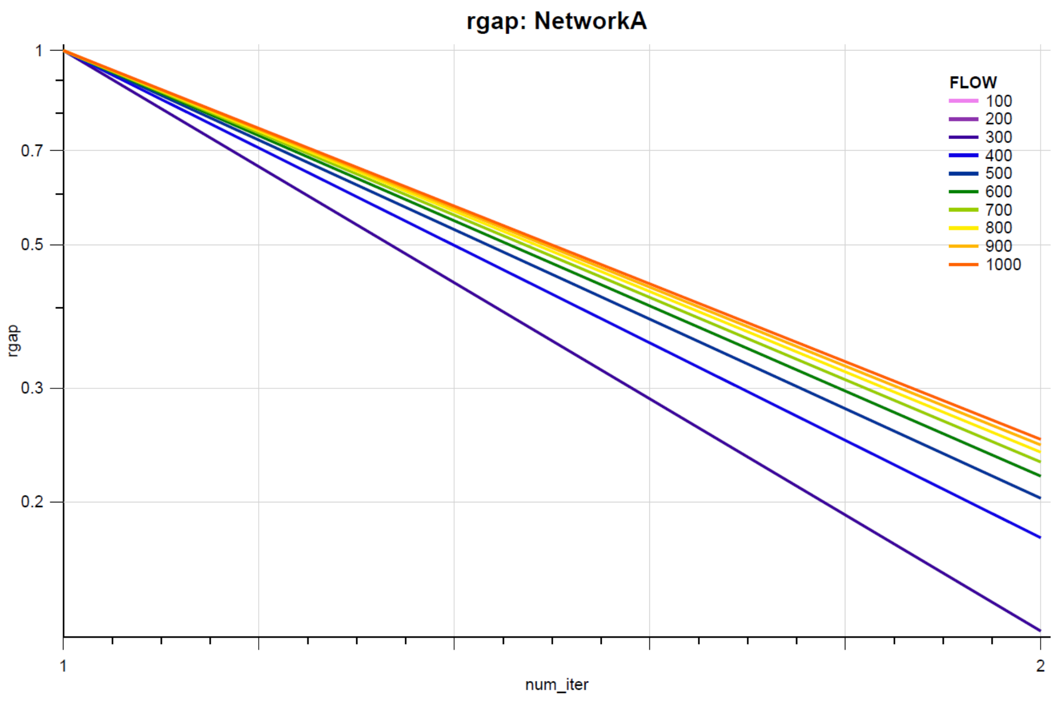

FIGURE 5.32: Seating Hyperarc test, placing hyperarc and effective frequency method. RGAP of the solution algorithm.

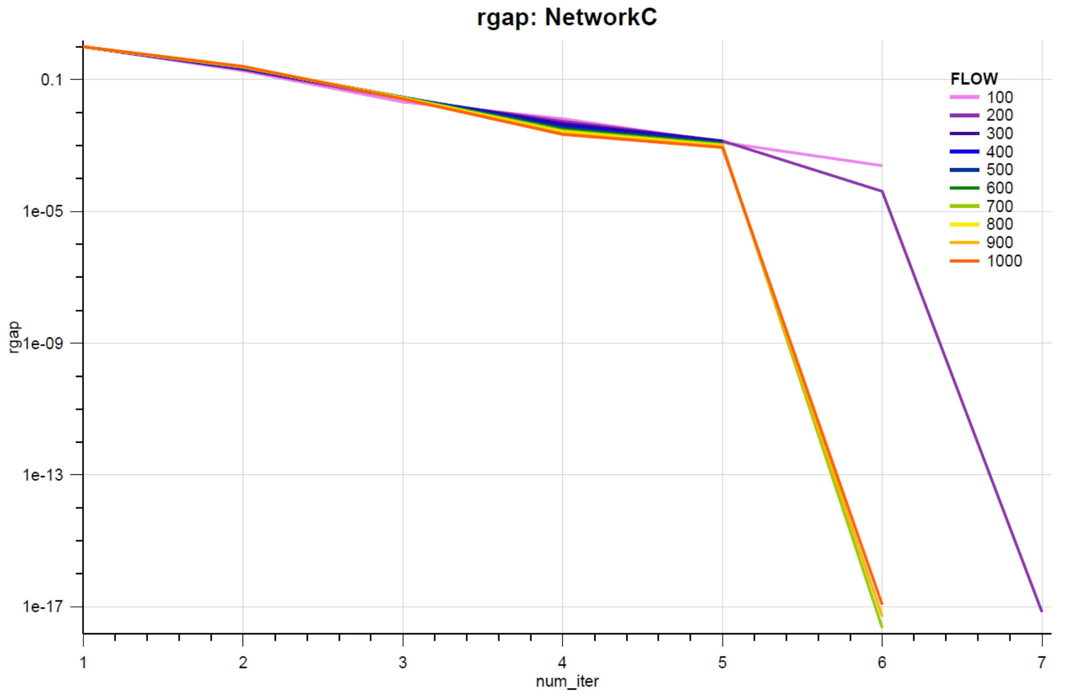

FIGURE 5.33: Seating Hyperarc test, dwelling hyperarc. RGAP of the solution algorithm.

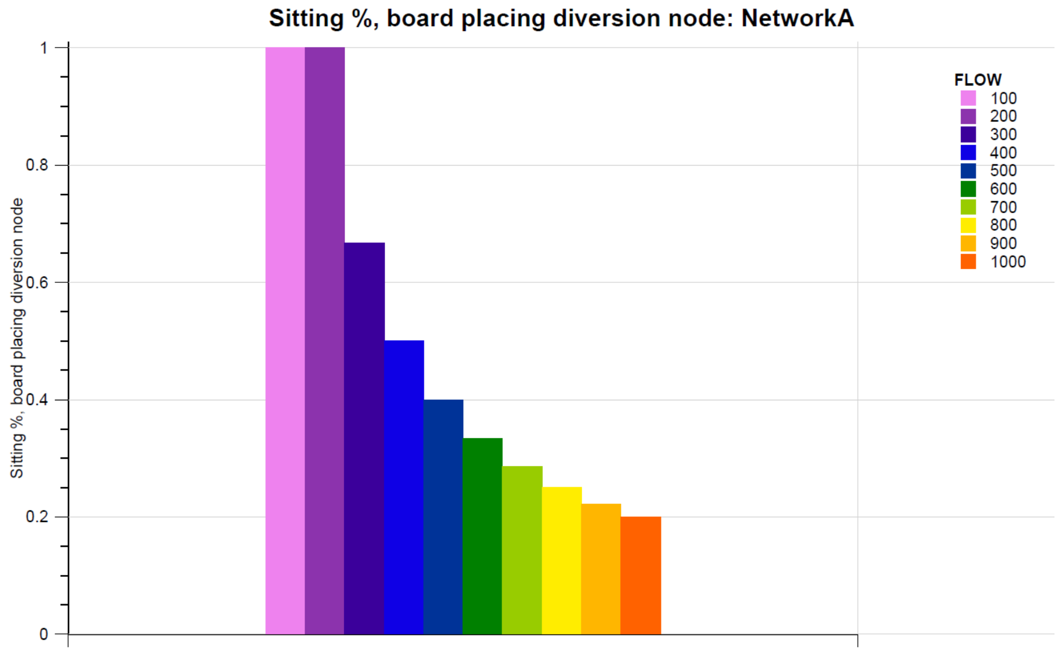

FIGURE 5.34: Seating Hyperarc test, placing hyperarc and effective frequency method. Expected percentage of passengers sitting on-board of line *BUS1* at equilibrium.

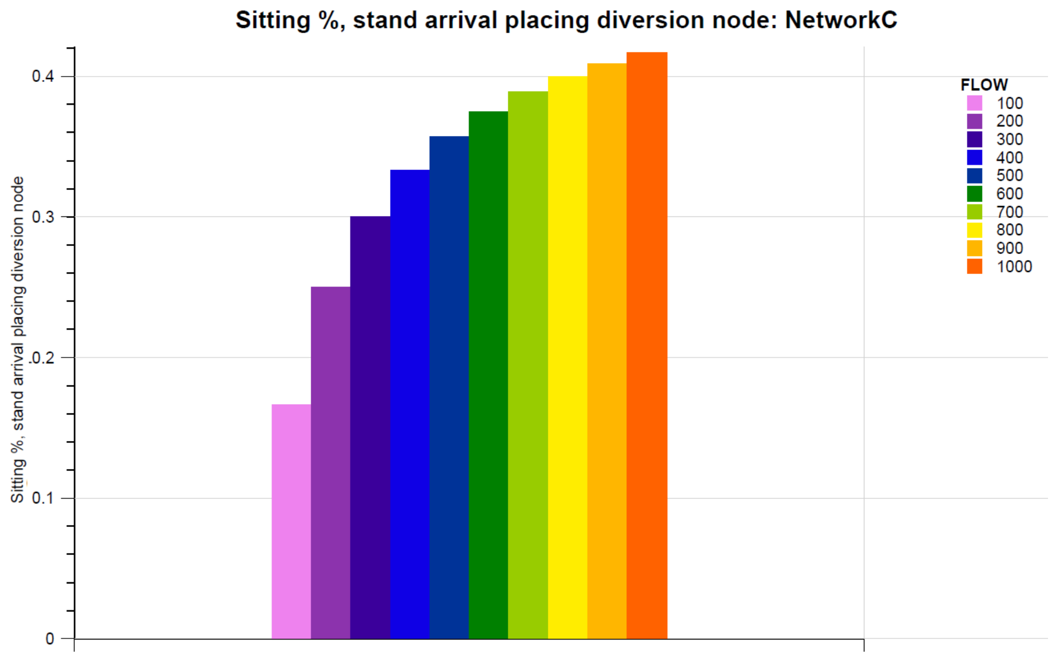

FIGURE 5.35: Seating Hyperarc test, dwelling hyperarc. Expected percentage of passengers sitting on-board of line *BUS1* at equilibrium.

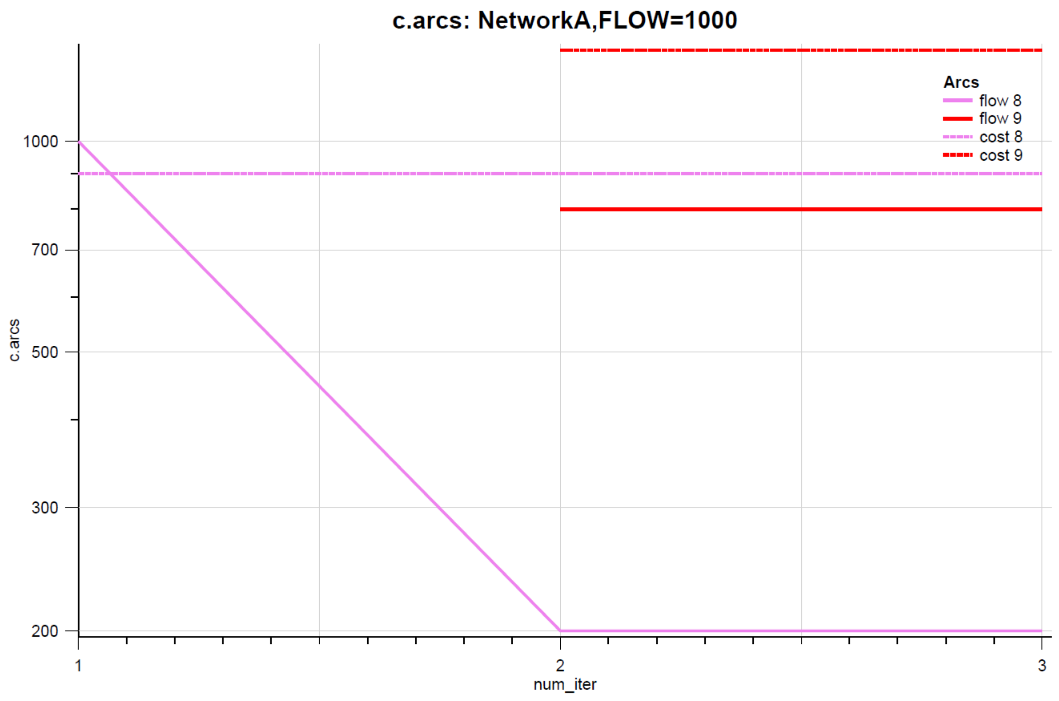

FIGURE 5.36: Seating Hyperarc test, placing hyperarc and effective frequency method. Expected cost and flow of seat running arc (in violet) and stand running arcs (in red).

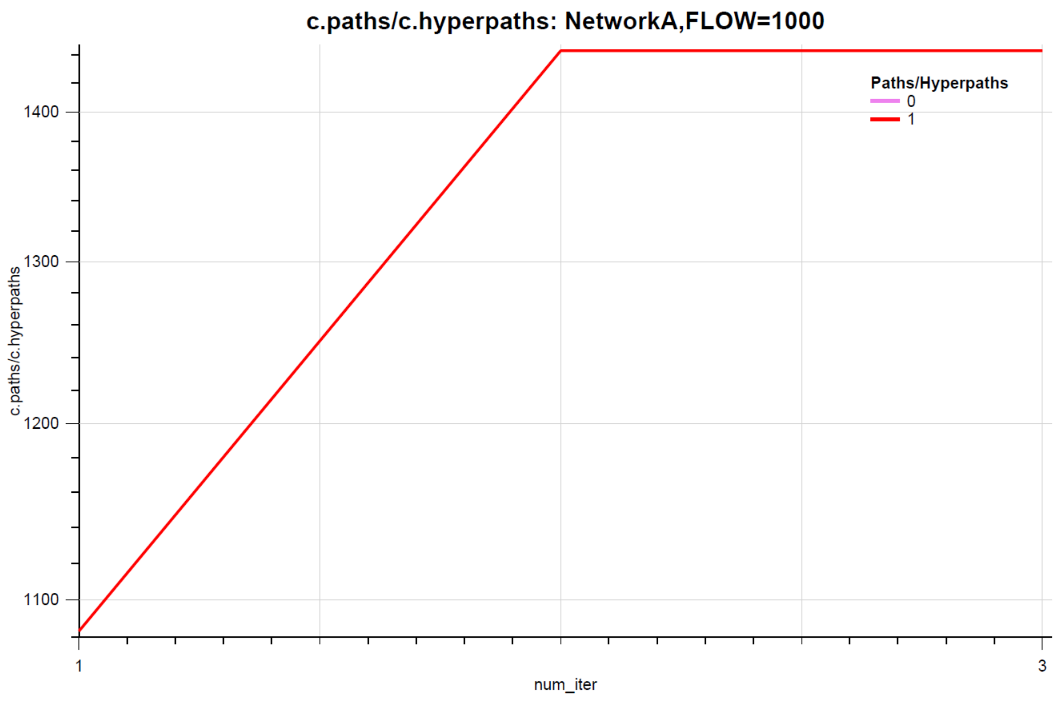

FIGURE 5.37: Seating Hyperarc test, placing hyperarc and effective frequency method. Hyperpath cost equilibrium. The walking option is not represented as its flow is null, while the option of taking line *BUS1* is represented in red.

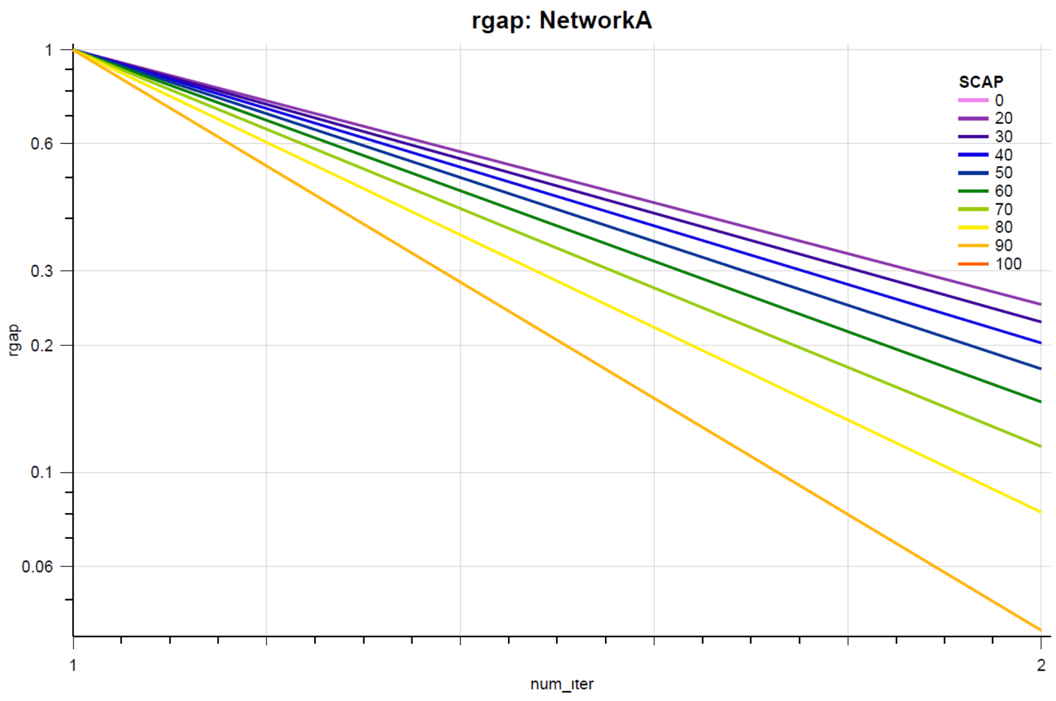

FIGURE 5.38: Seating Hyperarc test, placing hyperarc and effective frequency method. RGAP of the solution algorithm for varying $k_{BUS1}^{seat}$.

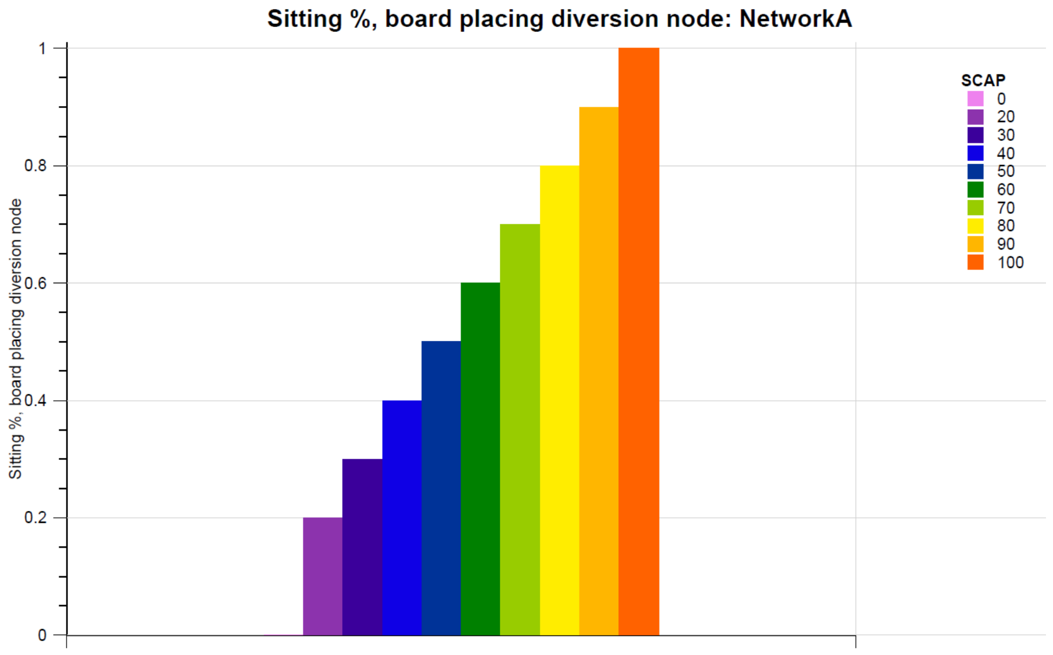

FIGURE 5.39: Seating Hyperarc test, placing hyperarc and effective frequency method. Expected percentage of passengers sitting on-board of line *BUS1* at equilibrium for varying $k^{seat}_{BUS1}$.

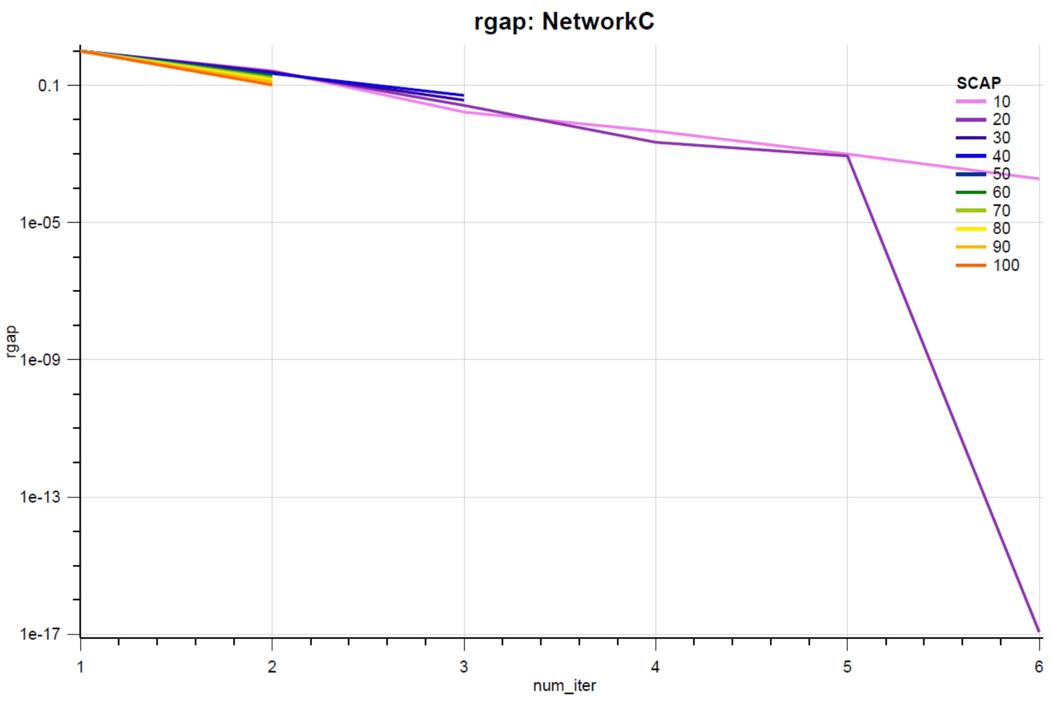

FIGURE 5.40: Seating Hyperarc test, dwelling hyperarc. RGAP of the solution algorithm for varying $k^{seat}_{BUS1}$.

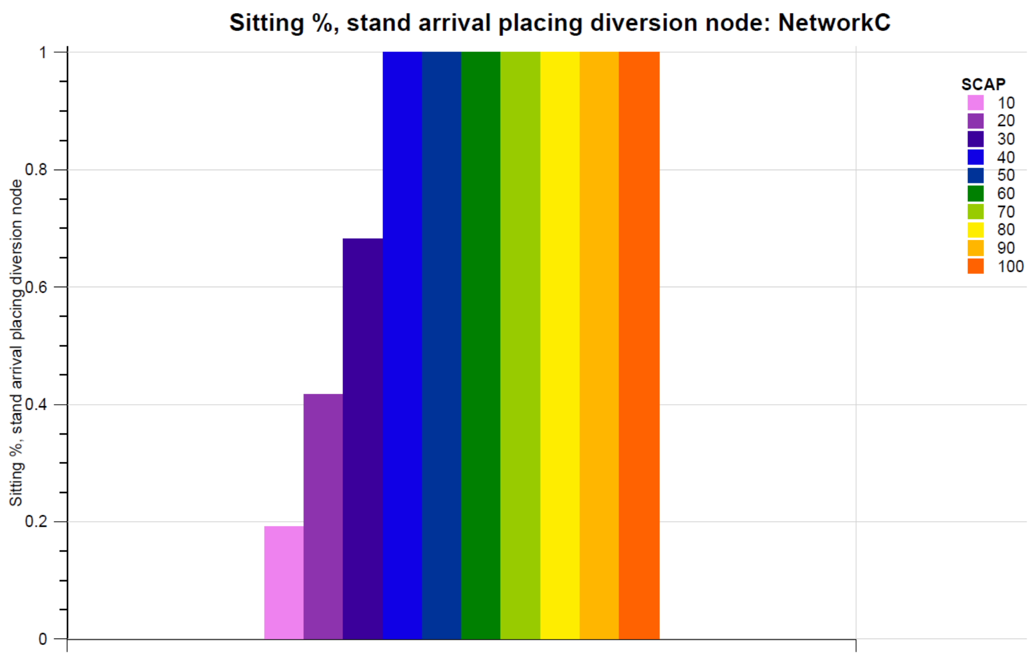

FIGURE 5.41: Seating Hyperarc test, dwelling hyperarc. Expected percentage of passengers sitting on-board of line *BUS1* at equilibrium for varying $k^{seat}_{BUS1}$.

### 5.1.6 Test - Waiting Hyperarcs

As stated before, waiting hyperarcs are tested on *NetworkB*, composed of two lines serving two stops.
To study how diversion probabilities of waiting hyperarc change at the diversion node (i.e. stop *1*), all the other congestion phenomena (eg. crowding, queuing, etc.) were deactivated. Moreover, the effective frequency method was adopted.

To this purpose, many simulations were carried out by setting a constant demand going from node *1* to node *2* (1000 passengers per hour) and the standing vehicle capacity equal to 20 passengers per vehicle. Moreover, the expected headway of line *BUS1* was kept constant to the value of 5 minutes, while the expected headway of line *BUS2* was varied between 1 and 10 minutes.
Figure 5.42 illustrates the trend of algorithm convergence during the assignment algorithm respectively, while figure 5.43 shows the expected percentage of passengers boarding line *BUS1* at equilibrium. Different values of line *BUS2* expected headway are represented in different colours.
Figure 5.42 shows how the equilibrium is found just after the first iteration, thus the equilibrium solution is the "*all-or-nothing*" assignment of uncongested network. This is because the cost of waiting hyperarcs depend solely on the line effective frequencies, as well as the diversion probabilities. Thus, if no other congestion phenomena are considered, the effective frequencies of the two lines are the same and, consequently, the conditional probabilities of the stop node are equal to the diversion probabilities of the waiting hyperarc and the attractive line set is composed of both lines.

Furthermore, as expected the conditional percentage of line *BUS1* increases with the expected headway of line *BUS2* (the headway is inversely proportional to the waiting time). Expected flows are exactly halves at the diversion node only when the headways are the same (identical lines).

To study further in detail the behaviour of waiting hyperarcs, other simulations were run. More precisely, the demand was varied between 100 and 1000 passengers per hour (with steps of 100 passengers per hour), while the two lines were assumed to have the same expected headway and similar speed: 20 kilometres per hour for line *BUS1* and 15 kilometres per hour for line *BUS2*.
Figure 5.44 and 5.45 are the same of figure 5.42 and 5.43 for the case with varying demand.
From figure 5.44 the convergence is reached again in one iteration for flows below 500 passengers per hour, while more than 50 iterations are required otherwise. This is a direct consequence of the effective frequency method adopted, which decreases the nominal frequencies in function of the flow of passengers waiting the line.
More precisely, it can be seen from figure 5.45 that, initially, all flows are assigned to line *BUS1*, which means that the waiting hyperarc *1*, composed of line *BUS1* only, is chosen by passengers.
When the demand starts growing, the effective frequency of line *BUS1* decreases until users find no convenient to take only line *BUS1*. Indeed, being line *BUS1* crowded, they may be able to board line *BUS2* firstly and, thus, decrease their costs (strategies concept). Hence, for values of demand higher than 400 passengers per hour, some passengers start including in their attractive line set also line *BUS2* and, consequently, the new equilibrium is between waiting hyperarc *1* (composed of line *BUS1* solely) and waiting hyperarc *2* (composed of line *BUS1* and line *BUS2*).
Of course, no passenger chooses the hyperarc composed by line *BUS2* solely as it is slower, hence more expensive.
In conclusion, the increase of number of iterations necessaries to find equilibrium is because, for demand larger than 400 passengers per hour, the algorithm looks for an equilibrium among hyperarcs, which is more difficult than the ones studied until now.

Finally, it is important to stress out that some passengers stay stick to the choice of waiting for line *BUS1* (i.e. waiting hyperarc composed only by one branch). This phenomenon can be observed in real-life cases and it is the only way to find UE (i.e. no user finds convenient to change choice unilaterally).

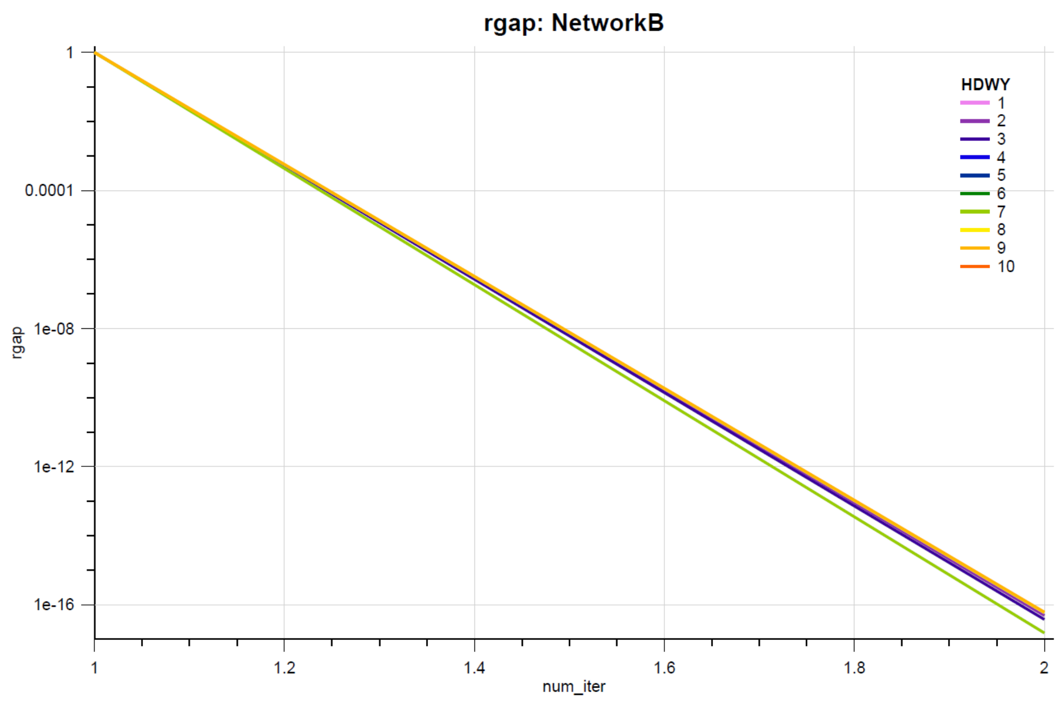

FIGURE 5.42: Waiting Hyperarc test. RGAP of the solution algorithm.

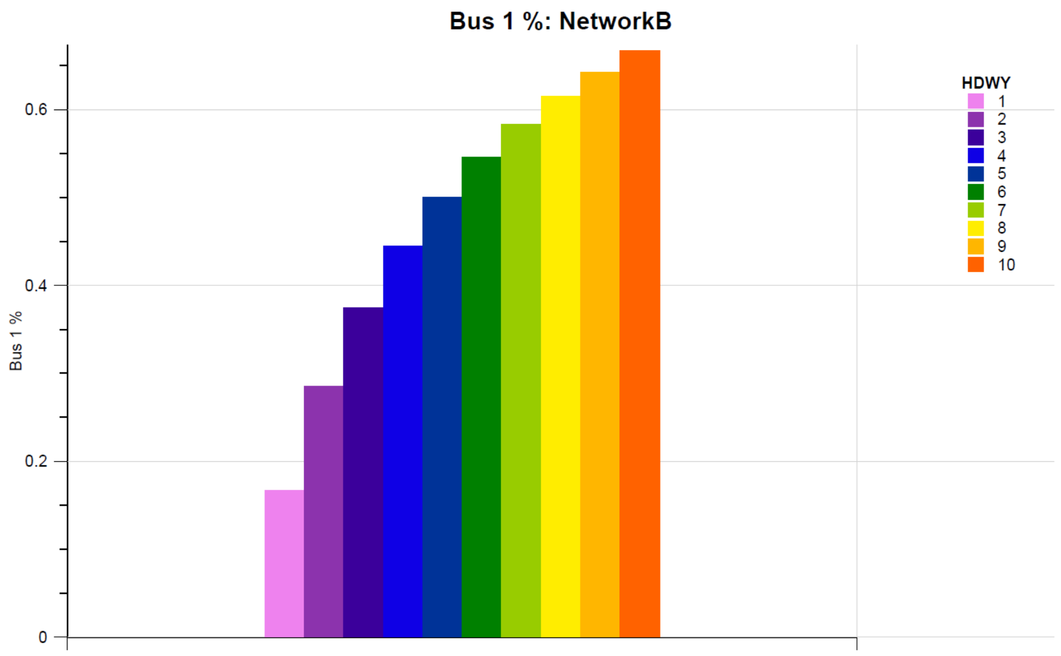

FIGURE 5.43: Waiting Hyperarc test. Expected percentage of passengers choosing line *BUS1* at equilibrium.

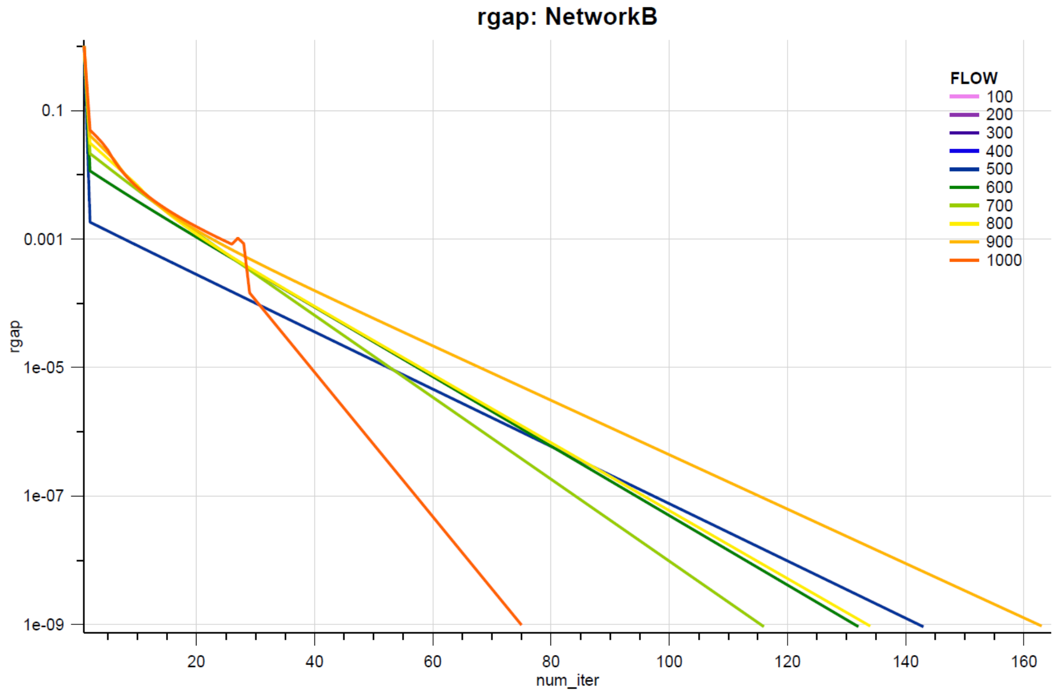

FIGURE 5.44: Waiting Hyperarc test. RGAP of the solution algorithm for varying demand.

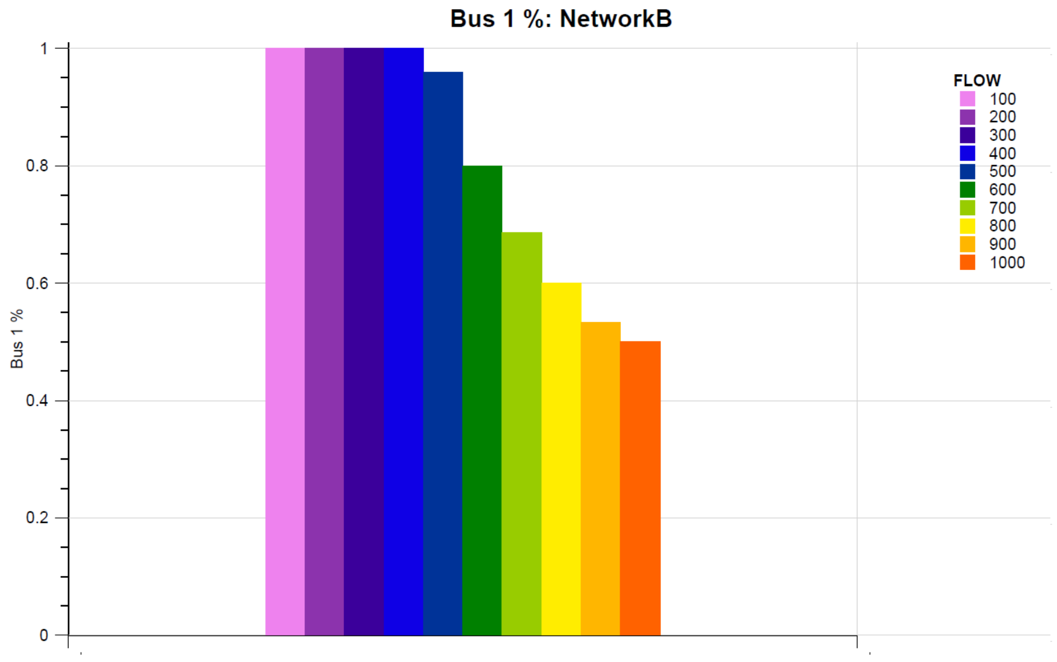

FIGURE 5.45: Waiting Hyperarc test. Expected percentage of passengers choosing line *BUS1* at equilibrium for varying demand.

## 5.2 Visum Example Simulation

In this section, the network introduced in section 2.2.1 Example for the Headway-Based Assignment and represented in figure 2.1 was adopted to perform a simulation of the implemented transit assignment.
The UE was computed over a one-hour time interval of a common day, whose demand consists in one user group following the OD matrix of table 5.8.

| | *A-Village* | *X-City* | *C-Village* | *B-Village* |
|---|---|---|---|---|
| *A-Village* | 0 | 200 | 20 | 0 |
| *X-City* | 200 | 0 | 500 | 200 |
| *C-Village* | 20 | 500 | 0 | 0 |
| *B-Village* | 0 | 200 | 0 | 0 |

TABLE 5.8: Visum Example Simulation, OD matrix of the assignment demand.

For what concerns the network topology, it is described in section 2.2.1 Example for the Headway-Based Assignment.
Parameters illustrated in table 5.9, 5.10, 5.11, 5.12 were adopted to set the assignment.

TABLE 5.9: Visum Example Simulation, parameters related to the links.

| Parameter | Comment | *Pedestrians* | *Bus* | *Train* |
|---|---|---|---|---|
| $v_a^{walk}$ | walking speed (km/h) | 6 | / | / |
| $\mathcal{S}_a$ | transit speed (km/h) | / | 18 | 30 |

TABLE 5.10: Visum Example Simulation, parameters related to the stop platforms.

| Parameter | Comment | *10* and *20* | *30* and *40* |
|---|---|---|---|
| $k_s^{pax}$ | platform capacity (pax) | 30 | 100 |
| $\beta_s^{p-crowd}$ | crowding BPR exponent (coef) | 2 | 2 |

TABLE 5.11: Visum Example Simulation, parameters related to the transit lines *Bus* and *Train*.

| Parameter | Comment | *Bus* | *Train* |
|---|---|---|---|
| $S_{ls}$ | commercial speed (km/time-UM); 0 means TSPE | 0 | 0 |
| $t_{ls}^{alight}$ | alighting time (min) | 0 | 0 |
| $t_{ls}^{board}$ | boarding margin time (min) | 0 | 0 |
| $h_{ls}$ | expected headway (min) | 10 | 20 |
| $\sigma_{ls}$ | Erlang headway variation coefficient | 0 | 0 |
| $lc^{kfee}$ | kilometric fee (€/km) | 0 | 0 |
| $c_{ls}^{bfee}$ | boarding fee (€) | 0 | 0 |
| $K_{ls}^{alight}$ | boarding door capacity (pax/min) | 20 | 20 |
| $K_{ls}^{board}$ | alighting door capacity (pax/min) | 20 | 20 |
| $a^{ded} \in \{0,1\}$ | dedicated doors for boarding and alighting (coeff) | 0 | 0 |
| $t_{ls}^{doors}$ | door operation time (min) | 0.5 | 0.5 |
| $t_{ls}^{min}$ | minimum dwell time, after DOTI | 0 | 2 |
| $h_{ls}^{offs}$ | offset w.r.t. simulation Initialize if no run is given (min) | 5 | 0 |
| $k_{ls}^{seat}$ | seating capacity (pax) | 10 | 50 |
| $k_{ls}^{stand}$ | standing capacity (pax) | 40 | 50 |
| $\beta_{ls}^{v-crowd}$ | discomfort BPR exponent (coef) | 2 | 2 |
| $\alpha_{ls}^{queue}$ | queuing BPR multiplier (coef) | 1 | 1 |
| $\beta_{ls}^{queue}$ | queuing BPR exponent (coef) | 4 | 4 |
| $\alpha_{ls}^{dwell}$ | dwelling BPR multiplier (coef) | 1 | 1 |
| $\beta_{ls}^{dwell}$ | dwelling BPR exponent (coef) | 2 | 2 |

TABLE 5.12: Visum Example Simulation, parameters related to the user group.

| Parameter | Comment | *Group 1* |
|---|---|---|
| $\gamma_g^{vot}$ | value of time (€/h) | 60 |
| $\gamma_g^{dist}$ | value of distance (€/km) | 0 |
| $\gamma_g^{wait}$ | waiting discomfort (coef) | 1 |
| $\gamma_g^{walk}$ | walking discomfort (coef) | 1 |
| $c_g^{tran}$ | transfer cost (€) | 0 |
| $\gamma_g^{stand}$ | standing discomfort (coef) | 1.5 |
| $\gamma_g^{seat}$ | sitting discomfort (coef) | 1 |
| $\gamma_g^{mfee}$ | fee multiplier (coef) | 0 |
| $\alpha_g^{crowd}$ | crowding discomfort BPR multiplier (coef) | 1 |
| $\gamma_g^{risk}$ | risk adverseness (coeff) | 1 |
| $nl_g^{max}$ | maximum number of attractive lines (No) | 10 |

The assignment algorithm needed 116 iterations to converge when for stop criterion RGAP equal to $10^{-4}$.
Figure 5.46 and 5.47 show the results represented in Visum. More precisely, the first one illustrates the flows as bars on links, while the second one illustrates the passenger boarding and alighting at stops as charts on stop points.

It can be observed that, according to the implemented software, the right-bottom part of the network is highly congested, and the supply is not enough to satisfy the demand. Indeed, as shown from the presence of dark colours almost all over the network, all transit services except the *Bus* line connecting *C-Village* and *B-Village* have a standing cost that is at least twice its corresponding uncongested cost.
Moreover, for links connecting *B-Village* and *X-City*, and the other way around, some passengers prefer walking rather than squeezing/waiting the bus.
Finally, even if the cost of waiting the train is both directions is huge (as shown from the dark green and dark yellow columns at stop points *20* and *40* of figure 5.47), passengers choose to take the *Train* lines because they are faster, have more capacity and run over a shorter path than the bus lines connecting the same stops.

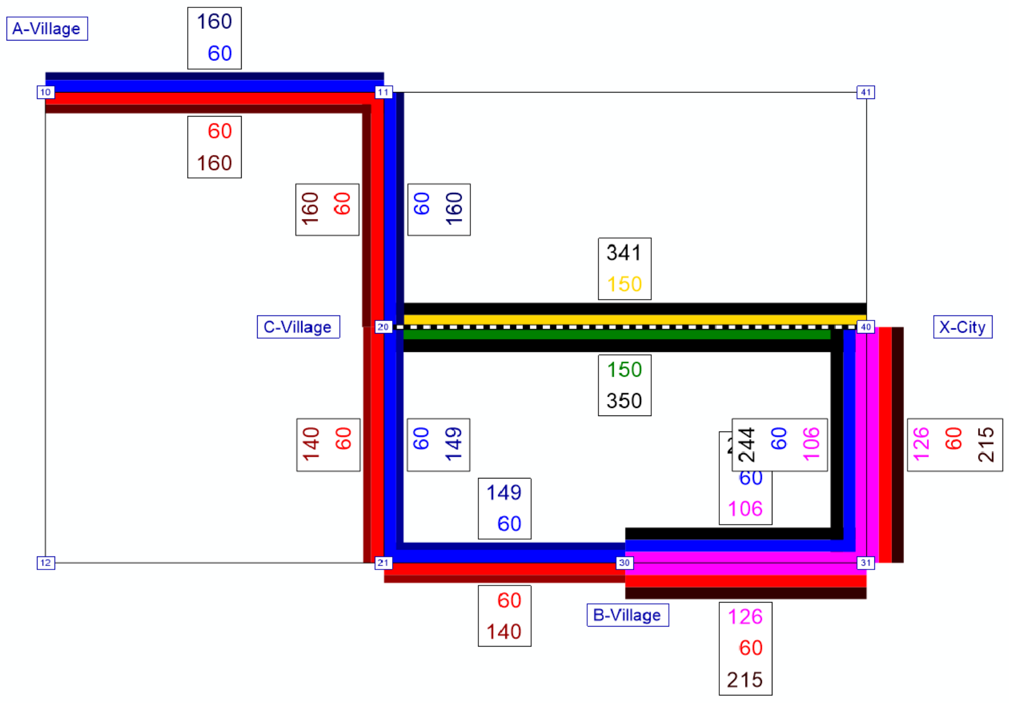

FIGURE 5.46: Visum Example Simulation, implemented software assignment. Network flows represented as bars on links.

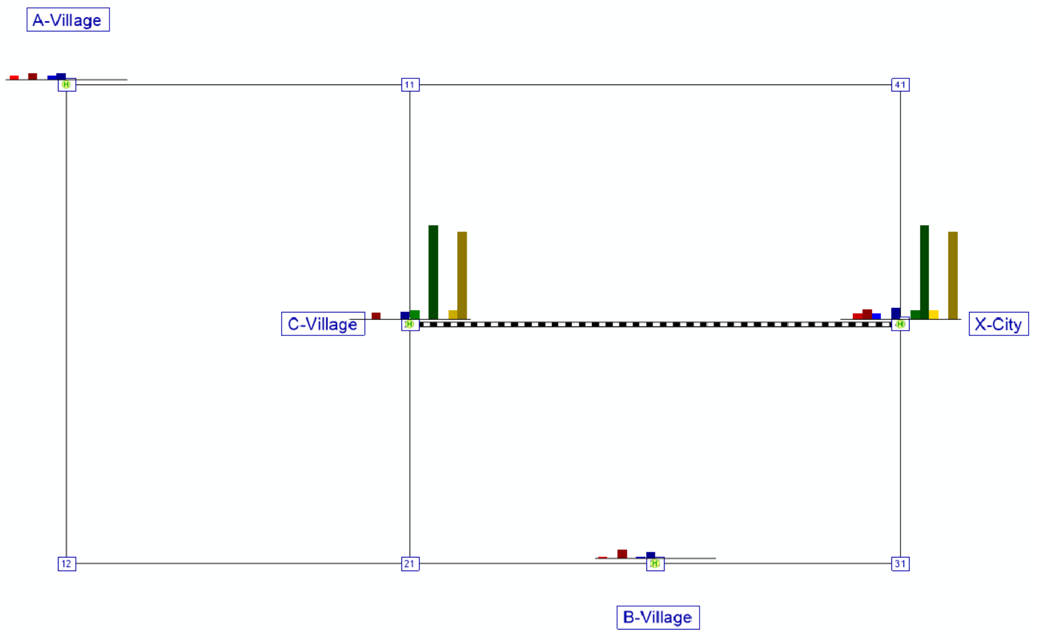

FIGURE 5.47: Visum Example Simulation, implemented software assignment. Boarding and alighting flows represented as charts on stop points.

Finally, Visum HB assignment procedure explained in section 2.2.1 Headway-Based Procedure was executed for the same network, where the headway was obtained by the UDA "*HDWY*" set for the previous assignment.
Figure 5.48 shows the results as bars on links when the following specific impedance parameters were set:

- $\alpha_{PJT} = 1$;
- $\alpha_f = 0$;
- $\alpha_{IVT} = 1$;
- $\beta_l = 1$;
- $\alpha_{AXT} = n.d.^1$;
- $\alpha_{ACT} = 1$;
- $\alpha_{EGT} = 1$;
- $\alpha_{WT} = 1$;
- $\alpha_{OWT} = 1$;
- $\alpha_{TWT} = 1$;
- $\beta_s = 1$;
- $\alpha_{NTR} = 0min.$

Comparing the results of Visum HB assignment procedure illustrated in figure 5.48 with the ones obtained with the implemented software (figure 5.46 and 5.47), it can be seen that Visum assigns all passengers to the transit service (i.e. no passengers walking) and that is expected that 700 passengers per hour will take the train, which is 7 times higher the train lines capacity. This is a consequence of Visum not considering any vehicle capacity restraints during the assignment.
Moreover, according to Visum results, line *Bus* could be completely deleted between *B-Village* and *C-Village* as it is almost unused.

[1]no auxiliary transport systems are available, thus the parameter setting is not irrelevant

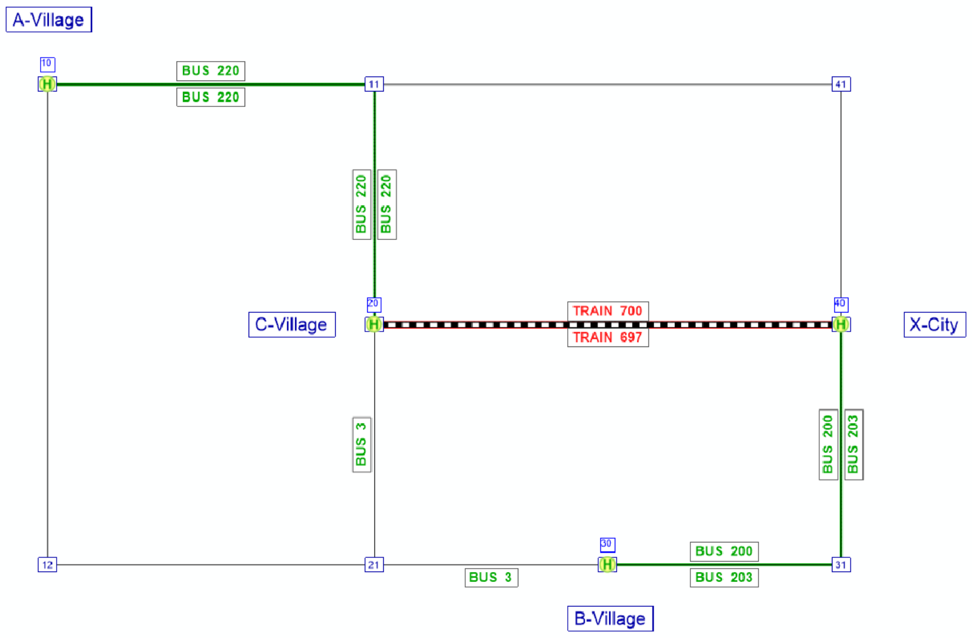

FIGURE 5.48: Visum Example simulation, Visum HB procedure. Network flows represented as bars on links.

An example of strategic planning using the implemented software is the following.
Considering the results shown by figure 5.46 and 5.47, the transport planner may wonder whether changing the bus frequency will relieve the network congestion or not. To this purpose, the transport planner could run a simulation, changing the expected headway of *Bus* from 10 minutes to 5 minutes.
Then, he or she might study the simulation outputs, that are shown in figure 5.49. From this picture, the transport planner could observe that changing the expected headway of *Bus* lines relieves significantly the network congestion. More precisely, it makes more passengers taking the *Bus* lines instead of walking or waiting the *Train* lines.
However, the transport planner may also realize that this solution is more expensive, as it requires 12 working buses per hour instead of just 6. Moreover, *Train* lines are still highly congested, hence, implementing this solution is not increase the level of service significantly.

As a consequence, the transport planner may also wonder whether, instead of changing the expected headway of the *Bus* lines, it would be better to decrease the expected headway of the *Train* lines (from 20 minutes to 10 minutes). To this purpose, the transport planner could run another simulation whose results are shown in figure 5.50.
From this picture he or she could observe that, when *Train* expected headway is equal to 10 minutes, passengers moves their choices from *Bus* lines to *Train* lines, making the network uncongested (with the exception of the *Train* lines). However, as the *Train* lines have higher frequency and, thus, higher capacity, their waiting costs and running costs decrease and, consequently, also congestion is lower with respect to previous cases.
Hence, the transport planner could decide to implement this last option in order to increase the level of service.
It is important to specify that the distribution of flows of this last solution is the same of the one resulting from the Visum HB procedure, even if its expected headway of *Train* lines is

halves.

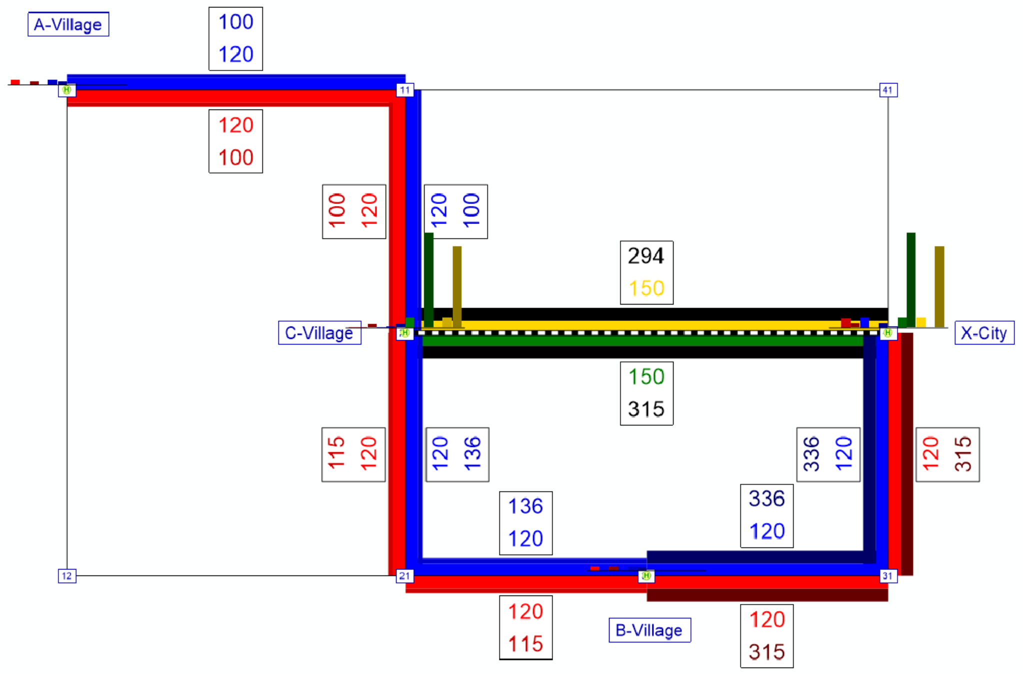

FIGURE 5.49: Visum Example Simulation, implemented software assignment. Network flows represented as bars on links, and boarding and alighting flows represented as charts on stop points for $h_{Bus} = 5min$.

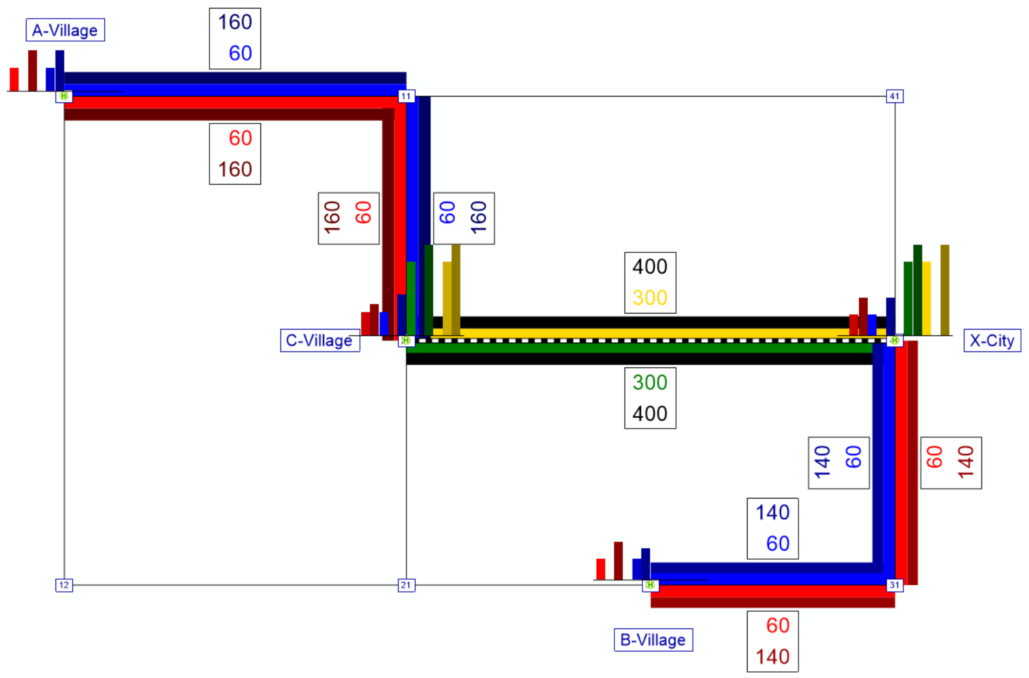

FIGURE 5.50: Visum Example Simulation, implemented software assignment. Network flows represented as bars on links, and boarding and alighting flows represented as charts on stop points for $h_{Train} = 10min$.

# 6 Conclusions and Further Developments

Transport has a positive key role in the economy and, at the same time, its policies can harm human health and the environment. Thus, governments have been investing huge capitals on public transport, attempting to attract more users by increasing the quality of service and comfort. Nevertheless, public transport must still compete with private one, as users' experience is influenced negatively by congestion and service reliability.
The increase of level of service can be achieved by reducing congestion through careful planning of the public transport service. To this purpose, different transit assignment models have been introduced in the last 60 years.

This thesis focuses on the simulation of transit networks, including passengers' congestion phenomena, as a tool for strategic and tactical planning.
More precisely, the thesis proposes a deterministic static assignment model which is able to compute UE for large-scale networks in case of passengers mingling at stops and high-frequency or low-reliability transit service, considering congestion phenomena. To this purpose, passengers' journey is divided into several legs, each of them described by a link. Each link is characterized by a specific cost performance function, that represents the cost of the trip leg in terms of time. This cost is usually composed of a monetary cost and a temporal cost, and the latter may be a function of congestion (depending on the trip leg).

Congestion phenomena such as *Overcrowding Congestion*, i.e. passengers discomfort due to overcrowding on-board of a vehicle or at platforms, *Queuing congestion*, i.e. passengers waiting at platforms, and *Dwelling Delay*, i.e. passengers waiting on-board of a vehicle, due to limited door capacity and overcrowding, are represented in the model by introducing additional costs (in terms of time) for the affected. These congestion costs are computed in different ways according to the studied phenomenon.
More precisely, overcrowding congestion on-board and at platforms is modelled by introducing a BPR factor which multiplies the travel time when flows is equal to zero, dwelling delays is modelled by adding extra dwelling time due to limited door capacity and overcrowding on-board (if the vehicle saturation rate is high, people have difficulties in access and egress the vehicle), and queuing congestion is modelled either by reducing the nominal frequency of the line depending on the number of passengers waiting to board (effective frequency method, where the vehicle standing capacity can be exceeded) or by introducing the high cost probability of boarding the vehicle only in "*next temporal layer*" (fail-to-board probability method, where the vehicle standing capacity cannot be violated).

In addition, the model represents other two congestion phenomena, such as the *availability of seats* (both for boarding passengers and dwelling passengers) and the *waiting process* at stops. This is achieved by using the concept of strategies and hyperpaths, i.e. plans users adopt to reach the desired destination at a minimum expected cost.
More precisely, congestion related to the waiting process is represented through a *waiting hyperarc*, where the strategy is chosen pre-trip (eg. user takes the first vehicle approaching the stop from his or her attractive line set composed of line 1 and line 2) and whose outcome depends on an event (e.g. the vehicle of line 1 approaches the stop). The attractive line set is determined by a *Greedy algorithm* and depends solely on given expected headways and *remaining costs* (i.e. expected cost perceived by users to reach the destination.
For what concerns congestion related to the process of finding a sit, it is represented by a *placing hyperarc* at placing nodes and a *dwelling hyperarcs* at the stand arrival placing node, and whose outcomes do not involve any choice as they depend on random variables (i.e. the passenger is lucky enough to find a sit).

As the congestion phenomena in transit networks are non-separable (i.e. cost of an arc depends also on the flows of other adjacent arcs, for example, cost of boarding a line depends on users that are already on the line), the uniqueness of equilibrium is not guaranteed. Thus, the problem of finding UE for congested transit networks cannot be solved through convex optimization.

This thesis model adopts an assignment algorithm which solves the *fixed-point formulation* of UE, based on the circular dependency between arc flows and congested costs.
Inputs of this assignment algorithm are the travel *demand*, defined by the OD matrix for each user group $g \in G$ and *arc attributes*, which consist in the graph $(N, A)$ representing the transit network according to model requirements.
First, the algorithm is initialized by computing the *free-flow cost* of each arc $a \in A$ (i.e. cost with null flow on the arc) and the corresponding *conditional probabilities* (i.e. probability that users take arc $a \in A$ conditional on being at its tail node). These probabilities are obtained through a sequential approach which computes the shortest hypertree (i.e. spanning tree $T \subseteq (N, A)$, such that the hyperpath distance from its root to any other vertex is the shortest hyperpath distance) through an extended version of Dijkstra algorithm, assuming that users reach their destination through a sequence of choices at nodes, where the local alternatives are the arcs of the forward star. These choices depend only on arcs cost, the remaining cost of their heads and if they are connected to the destination or not.
Then, the conditional probabilities are coupled with the demand flows for each OD pair and user group through an *"all-or-nothing"* assignment over the shortest hypertree to produce the *aggregated flows* for each arc, directed to each destination and for each user group. These flows satisfy Markov's memory-less assumption, i.e. they do not carry information about their origin.
Aggregated flows are then used as input of the *Network Loading Map*, which assigns final flows to each link, for each user group.
Finally, flows are used to update arc congested costs for each arc and each user group, and these updated costs are used again to find the new routes chosen by passengers and the corresponding conditional probabilities.
This iterative process is repeated until UE is found, that means the flows resulting in the

previous iteration from the *NLM* are the same of the one computed in the current iteration (*convergence*).

As UE assignment function is not a contraction, the RGP method on implicit hyperarcs was implemented to find UE convergence by determining search direction and size of the convergence search. Relative gap criterion is adopted as stop condition.
The implemented RGP is an innovative one, as it performs convergence search over arcs, avoiding the enumeration of hyperarcs. The implicit definition of hyperarcs is an important goal achieved, as enumerating them will be really costly in terms of computational time. Indeed, transit networks usually consists in large-scale instances and the total number of possible hyperarcs is usually huge (e.g. considering a small network composed of three stops and three lines serving those stops, there are already 16 hyperarcs: 7 waiting hyperarcs and 9 seating hyperarcs).

Simulations were executed implementing the model in Visual Basic 15, and results were visualized with the software PTV Visum 17 of the company PTV, which allows GIS-based data management in the field of private and public transport. To simplify the input process, data regarding networks and demand were imported from Visum.

To validate simulation results, the various congestion phenomena were tested singularly. More precisely, tests have shown that the algorithm converges in less than 10 iterations and with high precision (RGAP equal to $10^{-9}$) in case of simple congestion phenomena (eg. overcrowding, queuing with effective frequency, sitting hyperarcs). On the other hand, for dwelling delay and waiting hyperarcs more iterations are needed to reach the same convergence precision.
Moreover, in case of queuing congestion with fail-to-board probability, the algorithm does not converge as flow conservation does not hold (i.e. some flows is removed from the network and sent to the "*next temporal layer*").
From these tests it was also observed that all the modelled congestion phenomena behave as expected, except for the dwelling delay. Indeed, this congestion phenomenon starts having consequences on the choice of on-board dwelling passengers only after a certain value of BPR exponent. Before this value, dwelling congestion simply influences boarding passengers, which choose to walk instead of to board the line at the congested stop. This is a direct consequence of the costs introduced in boarding and alighting arcs with the purpose of avoid dwelling arc bypass (i.e. passengers alighting and boarding the same vehicle to avoid paying dwelling costs).

Finally, it can be said that the congestion and overcrowding are not only due to wrong strategic and tactical planning but also to the lack of operational and real-time planning. And yet, a system which can predict the network response to incidents and propose a solution to keep congestion levels acceptable does not yet exist.
To contain congestion and keep the level of service high, public transport operators should use a system able to predict congestion development and guide users' choices. However, a software fast enough to forecast real-time passengers' congestion in public transport networks while considering a wide range of congestion phenomena, does not exist yet.

The model implemented in this thesis is not suitable for real-time simulation and incident management, as it performs a static assignment which implies steady state setting (constant flows and performances during the assignment period), allowing only an average evaluation of network performances during the analysis period. An example is the load of passengers on each line: the implemented model is can evaluate the phenomena, but it may not be satisfactory if the travel demand has a sharp peak.

However, this thesis can also be an early stage for the implementation of a real-time incident management software for public transport networks.
Indeed, outcomes of the implemented model can be used as inputs of the Transit Link Transmission Model (TLTM) shown in figure 6.1, which is a fast-macroscopic model that performs a dynamic assignment on any network from the results of its static.
As it can be seen from figure 6.1, inputs of TLTM are solely demand, supply and service features and the results of a static assignment. Hence, the static assignment performed from the implemented model can be easily turned dynamic by using its inputs and its results as inputs of a TLTM model. More precisely, the fundamental UE outputs for performing dynamic assignment through TLTM are the *splitting rates*, which are conditional probabilities at nodes.
TLTM is based indeed on the Simplified Kinematic Wave Theory(Newell, 1993), which

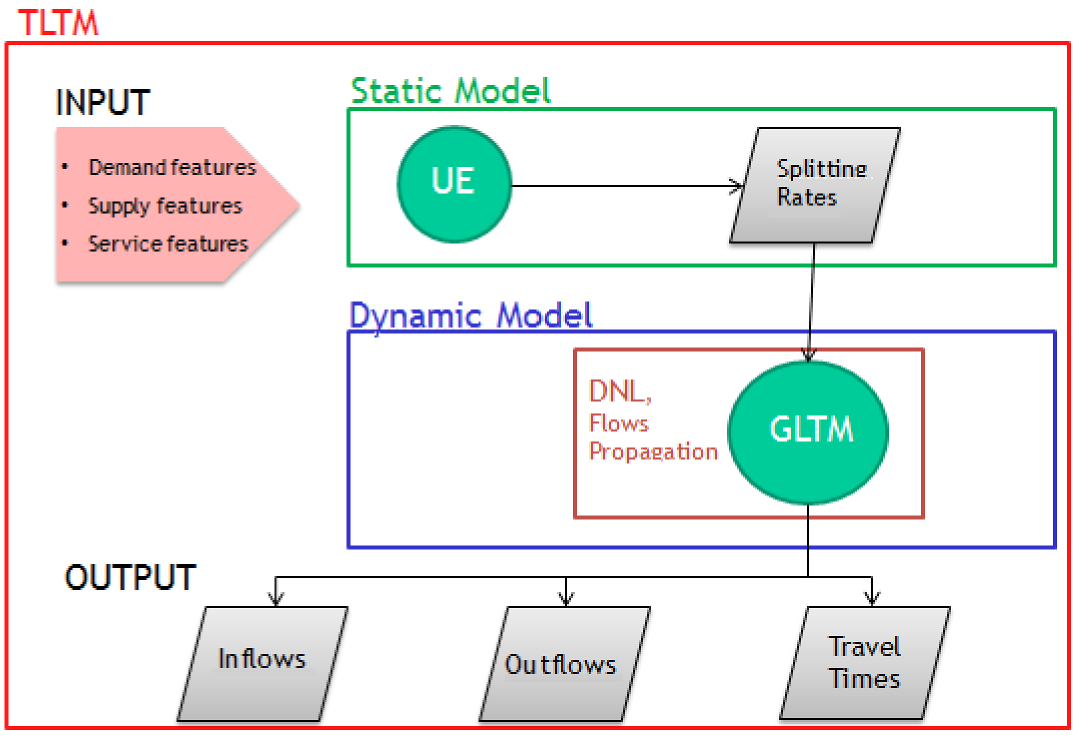

FIGURE 6.1: TLTM structure.

propagates flows jointly according to the splitting rates, and its outputs are inflows and outflows of network links, and their travel times.
TLTM is applied by solving consecutively two sub-models, called *link model* and *node model*, for each element of the network. The link model propagates flow states forward (vehicles

and passengers) and backward (space) on each arc, while the node model solves flow conflicts at nodes.
At current state, TLTM is able to do real-time monitoring and forecasting of the number of users travelling on a certain transit network. Since the forecast of transit network loads in real-time is one of the most requested research topics by cities such as Singapore, where public transport is the lifeblood of the society, the implementation of a TLTM model able to consider all the congestion phenomena described in this thesis is fundamental.
Another important aspect of TLTM is the possibility of real-time forecasting of accidental events, such as service interruptions, strikes, accidents, etc., which, if not adequately forecast and studied, can change significantly the service quality and level of service.

An example of TLTM application is the provision of information regarding available seat on a vehicle. If this information is made available to passengers, together with other information such as traffic and, if necessary, service malfunctions, the efficiency of the transit network as well as its level of service can increase significantly.
In conclusion, TLTM is a valid tool to move demand from private transport to the public one.

# *Acknowledgements*

Innanzitutto, vorrei ringraziare il mio relatore, il prof. Guido Gentile, che in questi mesi di collaborazione ha saputo coinvolgermi e motivarmi con la sua conoscenza e infinita passione.
Inoltre, ringrazio PTV SISTeMa e tutto il suo personale, che mi ha accolto e aiutato durante lo stage e la stesura tesi. Un grazie in particolare va a tutto il gruppo .NET: Agostino, Andrea, Daniele ed Edmondo, per avermi supportato e sopportato durante la mia permanenza.

Grazie alla prof.ssa Foglio, per avermi accompagnato in tutti questi anni e avermi aiutato quando ne avevo bisogno.

Durante luniversità ho vissuto in luoghi diversi, differenti tra loro ma allo stesso modo indimenticabili. E in ognuno di questi luoghi ho saputo trovare una casa, grazie alle persone che mi sono state vicine e, nonostante la lontananza, riescono ad esserlo tuttora.

Grazie ad Alessia, Giusy, Laura, Dario, Cristina, Andrea e Bonnina: senza di voi sarei ancora alla triennale. E a Vale e Chiara, amiche da sempre.

Grazie ad Alessandra, Carola, Martina e Monica, amiche e coinquiline che sanno riempire le mie giornate con risate e dolci.
Grazie a Flavio, Gemma, Lavinia, Letizia, Livia, Nicoletta, Nietta, Noemi e Roberta, che mi hanno insegnato a mettermi in gioco e a vivere Roma e lo spirito romano.

Thanks to Agostino, Alessandra, Davide, Federico, Lorenzo, Marj, Matteo, Nicola and Roberto, my classmates and friends. Thanks to Basak and Vivek, for being always happy and motivating.

Thanks to Myrto and Arek, who taught me that believing in yourself is enough to be successful. Thanks to Helena, my partner in coding and my favourite yellow duck.

Grazie ad Elena, sempre presente in ogni città. Grazie a te ho scoperto che, se si vuole, ci si può sempre incontrare e che viaggiare è uno dei piaceri della vita.

Grazie ad Isabel, che arrossisce facilmente. Grazie per la mia adolescenza: senza di te sarebbe stato tutto più noioso! E ad Abel, fratello "*indisponente*" che trova sempre qualcosa da dire.

Grazie a Vale, amica e confidente, sempre nel mio pensiero e nel mio cuore. Sarei persa senza di te.

Grazie a Davide, mio compagno, che ha imparato ad essere anche il mio migliore amico. Grazie perché mi "*dai ragione*" anche quando non ne ho, perché sai concedermi lo spazio di cui ho bisogno e farmi sorridere quando nientaltro ce la fa. Grazie!

Infine, il ringraziamento più importante va a voi, Papi e Fede, a cui devo questo traguardo. Avete sempre creduto in me, anche quelle volte in cui io stessa non ne ero in grado, motivandomi e spronandomi inesauribilmente. Grazie, perché con la vostra felicità, correttezza e forza mi avete insegnato ad essere la persona che sono.

# References

Beckman, M., McGuire, C., and Winsten, C. (1956). Studies in the economics of transportation. *Yale University Press.*

Bellman, R. (1958). On a routing problem. *Q Appl Math*, 16:87–90.

Bouzaïene-Ayari, B., Gendreau, M., and Nguyen, S. (1998). Passenger assignment in congested transit networks: a historical perspective. In Marcotte, P. and Nguyen, S., editors, *Equilibrium and advanced transportation modelling.*, chapter 3, pages 47–71. Springer.

Cascetta, E. (2001). Estimation of travel demand flows. In Cascetta, E., editor, *Transportation Systems Engineering: Theory and Methods.*, chapter 8, pages 387–439. Springer.

Cascetta, E. (2009). *Transportation Systems Analysis Models and Applications.* Springer.

Chriquì, C. and Robillard, P. (1975). Common bus lines. *Transportation Science*, 9:115–121.

Cominetti, R. and Correa, J. (2001). Common lines and passenger assignment in congested transit networks. *Transportation Science*, 35:250–267.

De Cea, J. and Fernandez, E. (1989). Transit assignment to minimal routes: an efficient new algorithm. *Traffic engineering and control*, 30:491–494.

De Cea, J. and Fernandez, E. (1993). Transit assignment for congested public transport system: An equilibrium model. *Transportation Science*, 27(2):133–147.

Dial, R. B. (1967). Transit pathfinder algorithm. *Highway Research Board*, 205:67–85.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numer Math*, 1:269–271.

Dora, C. and Phillips, M. (2000). Transport, environment and health. *WHO Regional Publications, European Series*, 89.

Florian, M., Florian, D., and Constantin, I. (2009). A new look at projected gradient method for equilibrium assignment. *Transportation Research Record: Journal of the Transportation Research Board*, 2090:10–16.

Gentile, G. (2017). Formulation of the transit link transmission model. *Transportation Research Procedia*, 27:889–896.

Gentile, G., Florian, M., Hamdouch, Y., Cats, O., and Nuzzolo, A. (2016a). The theory of transit assignment: basic modelling frameworks. In Gentile, G. and Noekel, K., editors, *Modelling Public Transport Passenger Flows in the Era of Intelligent Transport System: COST Act ion TU1004 (TransITS).*, chapter 6, pages 287–386. Springer.

Gentile, G. and Noekel, K. (2016). *Modelling Public Transport Passenger Flows in the Era of Intelligent Transport System: COST Act ion TU1004 (TransITS).* Springer.

Gentile, G., Noekel, K., Schmocker, J. D., Trozzi, V., and Chandakass, E. (2016b). The theory of transit assignment: demand and supply phenomena. In Gentile, G. and Noekel, K., editors, *Modelling Public Transport Passenger Flows in the Era of Intelligent Transport System: COST Act ion TU1004 (TransITS).*, chapter 7, pages 387–439. Springer.

Kurauchi, F., Bell, M. G. H., and Schmocker, J. D. (2003). Capacity constrained transit assignment with common lines. *J Math Model Algorithms.*, 24:309–327.

Leurent, F., Chandakas, E., and Poulhès, A. (2012). A passenger traffic assignment model with capacity constraints for transit networks. *Procedia - Social and Behavioral Sciences*, 54:772–784.

Newell, G. (1993). A simplified theory of kinematic waves in highway traffic, part i: general theory; part ii: queuing at freeway bottlenecks; part iii: multi-destination flows. *Transportation Research.*, 27:281–313.

Nguyen, S. and Pallottino, S. (1988). Equilibrium traffic assignment for large scale transit networks. *European Journal of Operational Research*, 37:176–186.

Oxford Economic Forecasting (2003). The economic effects of transport delays on the city of london. *Corporation of the City of London*, pages 26–27.

PTV AG (2017). Ptv visum 17 user manual.

PTV SISTeMA (2017). Tde user manual.

Rosen, J. B. (1960). The gradient projection method for nonlinear programming. part i. linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 8:181–217.

Spiess, H. and Florian, M. (1989). Optimal strategies: A new assignment model for transit networks. *Transportation Research*, 23B(2):83–102.

Trozzi, V., Gentile, G., Bell, M., and Kaparias, I. (2013). A passenger traffic assignment model with capacity constraints for transit networks. *Procedia - Social and Behavioral Sciences*, 80:427–454.

Wardrop, J. G. and Whitehead, J. I. (1952). Correspondence. some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers*, 1:767–768.

WHO/Europe (2017). Transport and health. http://www.euro.who.int/en/health-topics/environment-and-health/, last visited Dec 23, 2017.

# A Software IO methods and examples.

This appendix includes examples of input/output methods for the implemented software.

## A.1 Example of network modelling by CSV files

This section shows the CSV file used to model the network *OneLine_ThreeStops* of figure A.1, composed of:

- two road links of 5km length each;
- a line (*BUS 1*) serving each node (stops coincides with nodes) of the network with a headway-based service of expected headway equal to 6 minutes.

The demand in the assignment period consist on:

- flow of 500 passengers of *Class1* per hour going from zone *Block A* (connected to node 1) to *Work* (connected to node 3);
- flow of 300 passenger of *Class2* per hour going from zone *Block B* (connected to node 2) to *Work* (connected to node 3).

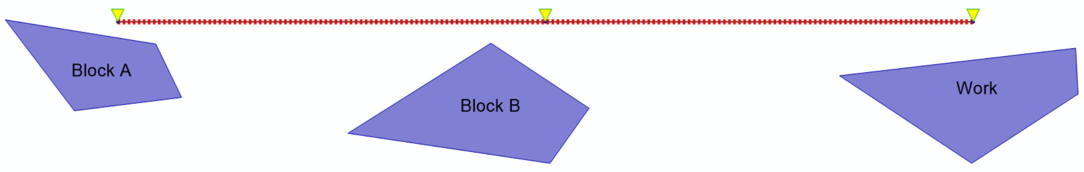

FIGURE A.1: *OneLine_ThreeStops* network. Yellow triangles are stops, dashed red line is the transit line *BUS 1*, purple polygons are demand zones.

Table A.1 and A.2 represent the CSV model of *OneLine_ThreeStops* network topology, table A.3 and A.4 represent the CSV model of the demand, and table A.5 represents the CSV model of demand and supply attributes [1].

Line *BUS 1* has:

- headway $h_{BUS1} = 6\ min$;

[1]Negative *BQUE* means hyperbolic, 0 fail to board. Negative *HVAR* means scheduled. *OFF* defined only when no line is provided

- speed $\mathcal{S}_{BUS1} = 20$ *km/h*;
- seat vehicle capacity $k^{seat}_{BUS1} = 30$ *pax/veh*;
- stand vehicle capacity $k^{stand}_{BUS1} = 50$ *pax/veh*.

All other attributes are negligible or assume the typical values.

For what concerns users, *Class1* differs from *Class2* in value of time $\gamma^{vot}_{g}$ and waiting discomfort coefficient $\gamma^{wait}_{g}$. More precisely:

- $\gamma^{vot}_{Class1} = 60$ *€/hour* $\quad \gamma^{wait}_{Class1} = 1$;
- $\gamma^{vot}_{Class2} = 40$ *€/hour* $\quad \gamma^{wait}_{Class2} = 2$.

All other attributes are the same and assume the typical values.

TABLE A.1: OneLine_ThreeStops network, CSV file of the node topology model. "/" means function 0.

| link | tail | head | leng | func |
|---|---|---|---|---|
| 1 | 1 | 2 | 5 | / |
| 2 | 2 | 3 | 5 | / |

TABLE A.2: OneLine_ThreeStops network, CSV file of the line topology model. "/" means function 0.

| line | node | stop | func |
|---|---|---|---|
| BUS 1 | 1 | 1 | / |
| BUS 1 | 2 | 2 | / |
| BUS 1 | 3 | 3 | / |

TABLE A.3: OneLine_ThreeStops network, CSV file of the OD flow model.

| user | orig | dest | flow | grup |
|---|---|---|---|---|
| 1 | 1 | 3 | 500 | Class1 |
| 1 | 2 | 3 | 300 | Class2 |

TABLE A.4: OneLine_ThreeStops network, CSV file of the user class model.

| grup | func |
|---|---|
| Class1 | Class1 |
| Class2 | Class2 |

TABLE A.5: OneLine_ThreeStops network, CSV file of the network and demand attribute model. 1 and 2 mean Group 1 (user class 1) and Group 2 (user class 2).

| func | para | valu | desc |
|---|---|---|---|
| 0 | WSPE | 5 | [link] walking speed (km/h) |
| 0 | APED | 0 | [link] walking BPR multiplier (coef) |
| 0 | BPED | 2 | [link] walking BPR exponent (coef) |
| 0 | TSPE | 30 | [link] transit speed (km/h) |
| 0 | TALI | 0 | [line-seg] alighting time (min) |
| 0 | TBOA | 0 | [line-seg] boarding margin time (min) |
| 0 | HDWY | 6 | [line-seg] expected headway (min) |
| 0 | KFEE | 0 | [line-seg] kilometric fee (€/km) |
| 0 | BFEE | 0 | [line-seg] boarding fee (€) |
| 0 | BCAP | 100000 | [line-seg] boarding door capacity (pax/min) |
| 0 | ACAP | 100000 | [line-seg] alighting door capacity (pax/min) |
| 0 | DEDO | 0 | [line-seg] dedicated doors (0-1) |
| 0 | DOTI | 0.5 | [line-seg] door operation time (min) |
| 0 | ADWL | 0 | [line-seg] dwelling BPR multiplier (coef) |
| 0 | BDWL | 2 | [line-seg] dwelling BPR exponent (coef) |
| 0 | OFFS | 0 | [line-seg] offset wrt simulation init (min) |
| 0 | SCAP | 30 | [line-seg] sitting capacity (pax) |
| 0 | VCAP | 50 | [line-seg] standing capacity (pax) |
| 0 | BCOM | 2 | [line-seg] discomfort BPR exponent (coef) |
| 0 | AQUE | 1 | [line-seg] queueing BPR multiplier (coef) |
| 0 | BQUE | 6 | [line-seg] queueing BPR exponent (coef) |
| 0 | PCAP | 100000 | [stop-plat] platform capacity (pax) |
| 0 | BSTP | 2 | [stop-plat] crowding BPR exponent (coef) |
| Class1 | VOFT | 60 | [class] value of time (€/h) |
| Class1 | VOFD | 0 | [class] value of distance (€/km) |
| Class1 | WAIT | 1 | [class] waiting discomfort (coef) |
| Class1 | WALK | 1 | [class] walking discomfort (coef) |
| Class1 | CTRA | 0 | [class] transfer cost (€) |
| Class1 | STND | 1.5 | [class] standing discomfort (coef) |
| Class1 | SEAT | 1 | [class] sitting discomfort (coef) |
| Class1 | MFEE | 1 | [class] fee multiplier (coef) |
| Class1 | ACRW | 1 | [class] crowding discomfort BPR multiplier (coef) |
| Class1 | RISK | 1 | [class] risk averseness (coef) |
| Class1 | MNAL | 10 | [class] maximum number of attractive lines (No |
| Class2 | VOFT | 40 | [class] value of time (€/h) |
| Class2 | VOFD | 0 | [class] value of distance (€/km) |
| Class2 | WAIT | 2 | [class] waiting discomfort (coef) |
| Class2 | WALK | 1 | [class] walking discomfort (coef) |
| Class2 | CTRA | 0 | [class] transfer cost ($e$) |
| Class2 | STND | 1.5 | [class] standing discomfort (coef) |
| Class2 | SEAT | 1 | [class] sitting discomfort (coef) |
| Class2 | MFEE | 1 | [class] fee multiplier (coef) |
| Class2 | ACRW | 1 | [class] crowding discomfort BPR multiplier (coef) |
| Class2 | RISK | 1 | [class] risk averseness (coef) |
| Class2 | MNAL | 10 | [class] maximum number of attractive lines (No) |

## A.2 Example of results representation

This section shows the results of the assignment performed on the network *OneLine_Three-Stops*, represented in figure A.1.
Outputs represented through CSV files are shown by table A.6 and table A.7, while figure A.2 shows the flow results on an image where also the capacity of links is shown (when relevant "*flow/capacity*").
Finally, figure A.3 shows the flow results on links (represented by flow bars) and the results for stop points (represented as histogram columns), while listing A.1 and listing A.2 show the CSV files produced by the software to export result-related UDAs and their graphics in Visum.

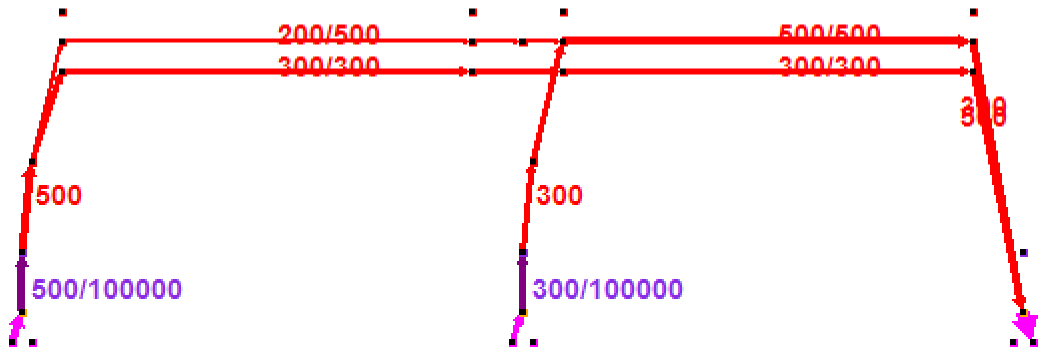

FIGURE A.2: *OneLine_ThreeStops* network, image file representing the flows on the network and capacities (flow/capacity).

TABLE A.6: OneLine_ThreeStops network, CSV file representing assignment results in terms of convergence trend.

| iter | clock | rgap | alfa |
|---|---|---|---|
| 1 | 7:47:19 PM | 1 | 1 |
| 2 | 7:47:19 PM | 0.121398401199613 | 1 |
| 3 | 7:47:19 PM | 3.57099622776313E-17 | 1 |

TABLE A.7: OneLine_ThreeStops network, CSV file representing assignment results in terms of flows and costs.

| Arc | Arc | Tail | Head | Capacity | Flow1 | Flow2 | Cost1 | Cost2 |
|---|---|---|---|---|---|---|---|---|
| a ∈ A | type | node | node | volume | vol | vol | cost | cost |
| index | code | code | code | pax/h | pass/h | pass/h | sec | sec |
| 1 | walk | Walk(1) | Walk(2) | 100000 | 0 | 0 | 3600 | 3600 |
| 2 | walk | Walk(2) | Walk(3) | 100000 | 0 | 0 | 3600 | 3600 |
| 3 | st0p | Walk(1) | Stop(1) | 100000 | 500 | 0 | 1 | 1 |
| 4 | st0p | Walk(2) | Stop(2) | 100000 | 0 | 300 | 1 | 1 |
| 5 | st0p | Walk(3) | Stop(3) | 100000 | 0 | 0 | 1 | 1 |
| 6 | board | Stop(1) | Line(BUS 1) Stop(1) Place | 16667 | 500 | 0 | 191 | 381 |
| 7 | place_seat | Line(BUS 1) Stop(1) Place | Line(BUS 1) Stop(1) DepSeat | 0 | 300 | 0 | 1 | 1 |
| 8 | place_stand | Line(BUS 1) Stop(1) Place | Line(BUS 1) Stop(1) DepStand | 0 | 200 | 0 | 1 | 1 |
| 9 | fail | Line(BUS 1) Stop(1) Place | Line(BUS 1) Stop(1) Fail | 0 | 0 | 0 | 1 | 1 |
| 10 | run_seat | Line(BUS 1) Stop(1) DepSeat | Line(BUS 1) Stop(2) ArrSeat | 300 | 300 | 0 | 900 | 900 |
| 11 | run_stand | Line(BUS 1) Stop(1) DepStand | Line(BUS 1) Stop(2) ArrStand | 500 | 200 | 0 | 1566 | 1566 |
| 12 | run_veh | Line(BUS 1) Stop(1) DepVeh | Line(BUS 1) Stop(2) ArrVeh | 0 | 0 | 0 | 1 | 1 |
| 13 | alight_seat | Line(BUS 1) Stop(2) ArrSeat | Walk(2) | 16667 | 0 | 0 | 15 | 15 |
| 14 | alight_stand | Line(BUS 1) Stop(2) ArrStand | Walk(2) | 16667 | 0 | 0 | 1 | 1 |
| 15 | dwell_seat | Line(BUS 1) Stop(2) ArrSeat | Line(BUS 1) Stop(2) DepSeat | 0 | 300 | 0 | 30 | 30 |
| 16 | dwell_stand | Line(BUS 1) Stop(2) ArrStand | Line(BUS 1) Stop(2) Switch | 0 | 200 | 0 | 30 | 30 |
| 17 | dwell_veh | Line(BUS 1) Stop(2) ArrVeh | Line(BUS 1) Stop(2) DepVeh | 0 | 0 | 0 | 1 | 1 |
| 18 | switch_seat | Line(BUS 1) Stop(2) Switch | Line(BUS 1) Stop(2) DepSeat | 0 | 0 | 0 | 1 | 1 |
| 19 | switch_stand | Line(BUS 1) Stop(2) Switch | Line(BUS 1) Stop(2) DepStand | 0 | 200 | 0 | 1 | 1 |
| 20 | board | Stop(2) | Line(BUS 1) Stop(2) Place | 16667 | 0 | 300 | 360 | 720 |
| 21 | place_seat | Line(BUS 1) Stop(2) Place | Line(BUS 1) Stop(2) DepSeat | 0 | 0 | 0 | 30 | 30 |
| 22 | place_stand | Line(BUS 1) Stop(2) Place | Line(BUS 1) Stop(2) DepStand | 0 | 0 | 300 | 1 | 1 |
| 23 | fail | Line(BUS 1) Stop(2) Place | Line(BUS 1) Stop(2) Fail | 0 | 0 | 0 | 1 | 1 |
| 24 | run_seat | Line(BUS 1) Stop(2) DepSeat | Line(BUS 1) Stop(3) ArrSeat | 300 | 300 | 0 | 900 | 900 |
| 25 | run_stand | Line(BUS 1) Stop(2) DepStand | Line(BUS 1) Stop(3) ArrStand | 500 | 200 | 300 | 2700 | 2700 |
| 26 | run_veh | Line(BUS 1) Stop(2) DepVeh | Line(BUS 1) Stop(3) ArrVeh | 0 | 0 | 0 | 1 | 1 |
| 27 | alight_seat | Line(BUS 1) Stop(3) ArrSeat | Walk(3) | 16667 | 300 | 0 | 1 | 1 |
| 28 | alight_stand | Line(BUS 1) Stop(3) ArrStand | Walk(3) | 16667 | 200 | 300 | 1 | 1 |
| 29 | orig | Orig(1) | Walk(1) | 0 | 500 | 0 | 1 | 1 |
| 30 | dest | Walk(1) | Dest(1) | 0 | 0 | 0 | 1 | 1 |
| 31 | orig | Orig(3) | Walk(3) | 0 | 0 | 0 | 1 | 1 |
| 32 | dest | Walk(3) | Dest(3) | 0 | 500 | 300 | 1 | 1 |
| 33 | orig | Orig(2) | Walk(2) | 0 | 0 | 300 | 1 | 1 |
| 34 | dest | Walk(2) | Dest(2) | 0 | 0 | 0 | 1 | 1 |

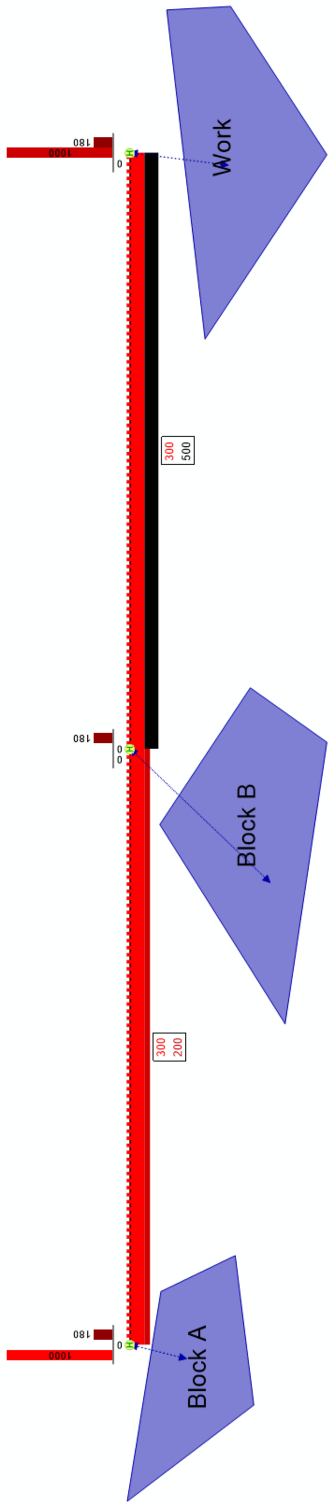

FIGURE A.3: *OneLine_ThreeStops* network, screen-shot of Visum model representing the flows on links, divided in walking flows and seating and standing flows for each line, and the boarding and alighting flows at stops. The congestion level for standing passengers is plotted by different color gradations of standing flow (higher congestion, darker color), while the waiting time at stops due to congestion is plotted as a darker column in the histogram. Flows on connectors are not plotted for space necessity.

```
$LINK:NO;FROMNODENO;TONODENO;PUTUE_WALK_FLOW;PUTUE_WALK_CONG;PUTUE_SEAT_FLOW_1_BUS_1;
    PUTUE_STAND_FLOW_1_BUS_1;PUTUE_STAND_CONG_1_BUS_1
   1;1;2;0;0;300;200;16
   1;2;1;0;NaN;0;0;0
   2;2;3;0;0;300;500;100
   2;3;2;0;NaN;0;0;0

   $CONNECTOR:ZONENO;NODENO;DIRECTION;TSYSSET;PUTUE_WALK_FLOW;PUTUE_WALK_CONG
   1;1;O;;500;0
   1;1;D;;0;0
   2;2;O;;300;0
   2;2;D;;0;0
   3;3;O;;0;0
   3;3;D;;800;0

   $STOPPOINT:NO;STOPAREANO;TSYSSET;NODENO;LINKNO;FROMNODENO;RELPOS;PUTUE_BOARD_FLOW_1_BUS_1;
    PUTUE_ALIGHT_FLOW_1_BUS_1;PUTUE_WAIT_CONG_1_BUS_1
   1;1;B;1;0;0;0;500;0;180
   2;2;B;2;0;0;0;300;0;180
   3;3;B;3;0;0;0;0;800;180
```

LISTING A.1: *OneLine_ThreeStops* network, CSV file representing the UDAs to import in Visum in order to represent results.

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<netEditorGraphicParameters version="2">
  <netEditor2D>
    <layerParameters>
      <layers partial="true">
        <layer draw="true" type="STOPPOINTS" />
      </layers>
    </layerParameters>
    <links>
      <layers partial="true">
        <layer draw="true" type="BARLABELS" />
        <layer draw="false" type="LABELS" />
        <layer draw="true" type="BARS" />
      </layers>
      <bars>
        <general barGap="0" connectBars="false" hideShortBars="true" useCrossValues="false">
          <label avoidOverlapping="false" labelGap="1" labelInsideBar="false"
     labelVisibilityRule="DRAWPOSITIONDEPENDENT" usePolyLineMidPos="false">
            <textFormat drawFrame="true" frameColor="ff000000" lineSpacing="50" textColor="
     ff000000" textColorFromBar="true" textOrientation="VERTICAL" textSize="2.5" transparentText
     ="false">
              <fontStyle bold="false" fontFamilyName="Arial" italic="false" />
            </textFormat>
          </label>
          <objectSelection classificationAttrID="TYPENO" drawOnClosedObjects="false"
     drawOnlyOnActiveMainObjects="false" drawOnlyOnActiveObjects="true" numDecPlaces="0"
     useClassifiedMode="false" useLayerOrder="false" />
          <defaultLinePathBarStyle draw="false">
            <polygonStyle drawBorder="true">
              <fillStyle color="ff00a800" hatchType="SOLIDFILLING" hatchWidth="1.5" />
              <borderStyle color="ff00a800" dashPattern="1" dashWidth="1" width="0.3" />
            </polygonStyle>
          </defaultLinePathBarStyle>
        </general>
        <items>
          <item barType="STANDARD" draw="true" gridColumnWidth="114">
            <scale autoScaleFactor="1" maxDimension="4" maxValue="1000" minValue="0"
     useAutoScale="true" useMinDimension="false" />
            <label labelThreshold="0" multiplyFactor="1" numDecPlaces="0" roundValue="1"
     showTitle="false" showUnit="false" showValue="true" stringFormatType="DEFAULTFMT"
     textContentType="TEXTFROMATTRIBUTEVALUE" textPriority="1" useLabelThreshold="true"
     useMultiplyFactor="false" useUserDefinedTitleText="true" userDefinedTitleText="Walk" />
```

```
            <standardBar classificationAttrID="PUTUE_WALK_CONG" negativeScaleAttrID="PUTUE_WALK_FLOW" numDecPlaces="0" scaleAttrID="PUTUE_WALK_FLOW" useClassifiedMode="true" useLayerOrder="false">
              <uniform draw="true">
                <polygonStyle drawBorder="true">
                  <fillStyle color="FFFF00FF" hatchType="SOLIDFILLING" hatchWidth="0" />
                  <borderStyle color="FFFF00FF" dashPattern="1" dashWidth="1" width="0.3" />
                </polygonStyle>
              </uniform>
              <classified>
                <class hasCustomLegendName="true" layerNo="1" legendName="&amp;lt;= 0" stringValue="*" upperLimit="0">
                  <contents draw="true">
                    <polygonStyle drawBorder="true">
                      <fillStyle color="FFFF00FF" hatchType="SOLIDFILLING" hatchWidth="1.5" />
                      <borderStyle color="FFFF00FF" dashPattern="1" dashWidth="1" width="0.3" />
                    </polygonStyle>
                  </contents>
                </class>
                <class hasCustomLegendName="true" layerNo="1" legendName="&amp;lt;= 20" stringValue="*" upperLimit="20">
                  <contents draw="true">
                    <polygonStyle drawBorder="true">
                      <fillStyle color="FFCC00CC" hatchType="SOLIDFILLING" hatchWidth="1.5" />
                      <borderStyle color="FFCC00CC" dashPattern="1" dashWidth="1" width="0.3" />
                    </polygonStyle>
                  </contents>
                </class>
                <class hasCustomLegendName="true" layerNo="1" legendName="&amp;lt;= 40" stringValue="*" upperLimit="40">
                  <contents draw="true">
                    <polygonStyle drawBorder="true">
                      <fillStyle color="FF990099" hatchType="SOLIDFILLING" hatchWidth="1.5" />
                      <borderStyle color="FF990099" dashPattern="1" dashWidth="1" width="0.3" />
                    </polygonStyle>
                  </contents>
                </class>
                <class hasCustomLegendName="true" layerNo="1" legendName="&amp;lt;= 60" stringValue="*" upperLimit="60">
                  <contents draw="true">
                    <polygonStyle drawBorder="true">
                      <fillStyle color="FF660066" hatchType="SOLIDFILLING" hatchWidth="1.5" />
                      <borderStyle color="FF660066" dashPattern="1" dashWidth="1" width="0.3" />
                    </polygonStyle>
                  </contents>
                </class>
                <class hasCustomLegendName="true" layerNo="1" legendName="&amp;lt;= 80" stringValue="*" upperLimit="80">
                  <contents draw="true">
                    <polygonStyle drawBorder="true">
                      <fillStyle color="FF330033" hatchType="SOLIDFILLING" hatchWidth="1.5" />
                      <borderStyle color="FF330033" dashPattern="1" dashWidth="1" width="0.3" />
                    </polygonStyle>
                  </contents>
                </class>
                <class hasCustomLegendName="true" layerNo="1" legendName="&amp;gt;0.4" stringValue="*" upperLimit="INF">
                  <contents draw="true">
                    <polygonStyle drawBorder="true">
                      <fillStyle color="FF000000" hatchType="SOLIDFILLING" hatchWidth="1.5" />
                      <borderStyle color="FF000000" dashPattern="1" dashWidth="1" width="0.3" />
                    </polygonStyle>
                  </contents>
                </class>
              </classified>
            </standardBar>
          </item>
          <item barType="STANDARD" draw="true" gridColumnWidth="114">
```

```
      <scale autoScaleFactor="1" maxDimension="4" maxValue="1000" minValue="0"
useAutoScale="true" useMinDimension="false" />
      <label labelThreshold="0" multiplyFactor="1" numDecPlaces="0" roundValue="1"
showTitle="false" showUnit="false" showValue="true" stringFormatType="DEFAULTFMT"
textContentType="TEXTFROMATTRIBUTEVALUE" textPriority="1" useLabelThreshold="true"
useMultiplyFactor="false" useUserDefinedTitleText="true" userDefinedTitleText="BUS_1 - Seat
" />
      <standardBar classificationAttrID="" negativeScaleAttrID="PUTUE_SEAT_FLOW_1_BUS_1"
numDecPlaces="0" scaleAttrID="PUTUE_SEAT_FLOW_1_BUS_1" useClassifiedMode="false"
useLayerOrder="false">
        <uniform draw="true">
          <polygonStyle drawBorder="true">
            <fillStyle color="FFFF0000" hatchType="SOLIDFILLING" hatchWidth="0" />
            <borderStyle color="FFFF0000" dashPattern="1" dashWidth="1" width="0.3" />
          </polygonStyle>
        </uniform>
      </standardBar>
    </item>
    <item barType="STANDARD" draw="true" gridColumnWidth="114">
      <scale autoScaleFactor="1" maxDimension="4" maxValue="1000" minValue="0"
useAutoScale="true" useMinDimension="false" />
      <label labelThreshold="0" multiplyFactor="1" numDecPlaces="0" roundValue="1"
showTitle="false" showUnit="false" showValue="true" stringFormatType="DEFAULTFMT"
textContentType="TEXTFROMATTRIBUTEVALUE" textPriority="1" useLabelThreshold="true"
useMultiplyFactor="false" useUserDefinedTitleText="true" userDefinedTitleText="BUS_1 -
Stand" />
      <standardBar classificationAttrID="PUTUE_STAND_CONG_1_BUS_1" negativeScaleAttrID="
PUTUE_STAND_FLOW_1_BUS_1" numDecPlaces="0" scaleAttrID="PUTUE_STAND_FLOW_1_BUS_1"
useClassifiedMode="true" useLayerOrder="false">
        <uniform draw="true">
          <polygonStyle drawBorder="true">
            <fillStyle color="FFFF0000" hatchType="SOLIDFILLING" hatchWidth="0" />
            <borderStyle color="FFFF0000" dashPattern="1" dashWidth="1" width="0.3" />
          </polygonStyle>
        </uniform>
        <classified>
          <class hasCustomLegendName="true" layerNo="1" legendName="&amp;lt;= 0"
stringValue="*" upperLimit="0">
            <contents draw="true">
              <polygonStyle drawBorder="true">
                <fillStyle color="FFFF0000" hatchType="SOLIDFILLING" hatchWidth="1.5" />
                <borderStyle color="FFFF0000" dashPattern="1" dashWidth="1" width="0.3" />
              </polygonStyle>
            </contents>
          </class>
          <class hasCustomLegendName="true" layerNo="1" legendName="&amp;lt;= 20"
stringValue="*" upperLimit="20">
            <contents draw="true">
              <polygonStyle drawBorder="true">
                <fillStyle color="FFCC0000" hatchType="SOLIDFILLING" hatchWidth="1.5" />
                <borderStyle color="FFCC0000" dashPattern="1" dashWidth="1" width="0.3" />
              </polygonStyle>
            </contents>
          </class>
          <class hasCustomLegendName="true" layerNo="1" legendName="&amp;lt;= 40"
stringValue="*" upperLimit="40">
            <contents draw="true">
              <polygonStyle drawBorder="true">
                <fillStyle color="FF990000" hatchType="SOLIDFILLING" hatchWidth="1.5" />
                <borderStyle color="FF990000" dashPattern="1" dashWidth="1" width="0.3" />
              </polygonStyle>
            </contents>
          </class>
          <class hasCustomLegendName="true" layerNo="1" legendName="&amp;lt;= 60"
stringValue="*" upperLimit="60">
            <contents draw="true">
              <polygonStyle drawBorder="true">
                <fillStyle color="FF660000" hatchType="SOLIDFILLING" hatchWidth="1.5" />
                <borderStyle color="FF660000" dashPattern="1" dashWidth="1" width="0.3" />
                </polygonStyle>
              </contents>
            </class>
            <class hasCustomLegendName="true" layerNo="1" legendName="&amp;lt;= 80"
 stringValue="*" upperLimit="80">
              <contents draw="true">
                <polygonStyle drawBorder="true">
                  <fillStyle color="FF330000" hatchType="SOLIDFILLING" hatchWidth="1.5" />
                  <borderStyle color="FF330000" dashPattern="1" dashWidth="1" width="0.3" />
                </polygonStyle>
              </contents>
            </class>
            <class hasCustomLegendName="true" layerNo="1" legendName="&amp;gt;0.4"
 stringValue="*" upperLimit="INF">
              <contents draw="true">
                <polygonStyle drawBorder="true">
                  <fillStyle color="FF000000" hatchType="SOLIDFILLING" hatchWidth="1.5" />
                  <borderStyle color="FF000000" dashPattern="1" dashWidth="1" width="0.3" />
                </polygonStyle>
              </contents>
            </class>
          </classified>
        </standardBar>
      </item>
    </items>
  </bars>
</links>
<connectors>
  <layers partial="true">
    <layer draw="true" type="BARLABELS" />
    <layer draw="false" type="LABELS" />
    <layer draw="true" type="BARS" />
  </layers>
  <bars>
    <general barGap="0" connectBars="false" hideShortBars="true" useCrossValues="false">
      <label avoidOverlapping="false" labelGap="0" labelInsideBar="false"
 labelVisibilityRule="DRAWPOSITIONDEPENDENT" usePolyLineMidPos="false">
        <textFormat drawFrame="true" frameColor="ff000000" lineSpacing="30" textColor="
 ff000000" textColorFromBar="true" textOrientation="VERTICAL" textSize="2.3" transparentText
 ="false">
          <fontStyle bold="false" fontFamilyName="Arial" italic="false" />
        </textFormat>
      </label>
      <objectSelection classificationAttrID="TYPENO" drawOnClosedObjects="false"
 drawOnlyOnActiveMainObjects="false" drawOnlyOnActiveObjects="true" numDecPlaces="0"
 useClassifiedMode="false" useLayerOrder="false" />
    </general>
    <items>
      <item barType="STANDARD" draw="true" gridColumnWidth="120">
        <scale autoScaleFactor="1" maxDimension="4" maxValue="1000" minValue="0"
 useAutoScale="true" useMinDimension="false" />
        <label labelThreshold="0" multiplyFactor="1" numDecPlaces="0" roundValue="1"
 showTitle="false" showUnit="false" showValue="true" stringFormatType="DEFAULTFMT"
 textContentType="TEXTFROMATTRIBUTEVALUE" textPriority="1" useLabelThreshold="true"
 useMultiplyFactor="false" useUserDefinedTitleText="true" userDefinedTitleText="Walk" />
        <standardBar classificationAttrID="PUTUE_WALK_CONG" negativeScaleAttrID="
PUTUE_WALK_FLOW" numDecPlaces="0" scaleAttrID="PUTUE_WALK_FLOW" useClassifiedMode="true"
 useLayerOrder="false">
          <uniform draw="true">
            <polygonStyle drawBorder="true">
              <fillStyle color="FFFF00FF" hatchType="SOLIDFILLING" hatchWidth="0" />
              <borderStyle color="FFFF00FF" dashPattern="1" dashWidth="1" width="0.3" />
            </polygonStyle>
          </uniform>
          <classified>
            <class hasCustomLegendName="true" layerNo="1" legendName="&amp;lt;= 0"
 stringValue="*" upperLimit="0">
              <contents draw="true">
                <polygonStyle drawBorder="true">
```

```
                  <fillStyle color="FFFF00FF" hatchType="SOLIDFILLING" hatchWidth="1.5" />
                  <borderStyle color="FFFF00FF" dashPattern="1" dashWidth="1" width="0.3" />
                </polygonStyle>
              </contents>
            </class>
            <class hasCustomLegendName="true" layerNo="1" legendName="&amp;lt;= 20"
 stringValue="*" upperLimit="20">
              <contents draw="true">
                <polygonStyle drawBorder="true">
                  <fillStyle color="FFCC00CC" hatchType="SOLIDFILLING" hatchWidth="1.5" />
                  <borderStyle color="FFCC00CC" dashPattern="1" dashWidth="1" width="0.3" />
                </polygonStyle>
              </contents>
            </class>
            <class hasCustomLegendName="true" layerNo="1" legendName="&amp;lt;= 40"
 stringValue="*" upperLimit="40">
              <contents draw="true">
                <polygonStyle drawBorder="true">
                  <fillStyle color="FF990099" hatchType="SOLIDFILLING" hatchWidth="1.5" />
                  <borderStyle color="FF990099" dashPattern="1" dashWidth="1" width="0.3" />
                </polygonStyle>
              </contents>
            </class>
            <class hasCustomLegendName="true" layerNo="1" legendName="&amp;lt;= 60"
 stringValue="*" upperLimit="60">
              <contents draw="true">
                <polygonStyle drawBorder="true">
                  <fillStyle color="FF660066" hatchType="SOLIDFILLING" hatchWidth="1.5" />
                  <borderStyle color="FF660066" dashPattern="1" dashWidth="1" width="0.3" />
                </polygonStyle>
              </contents>
            </class>
            <class hasCustomLegendName="true" layerNo="1" legendName="&amp;lt;= 80"
 stringValue="*" upperLimit="80">
              <contents draw="true">
                <polygonStyle drawBorder="true">
                  <fillStyle color="FF330033" hatchType="SOLIDFILLING" hatchWidth="1.5" />
                  <borderStyle color="FF330033" dashPattern="1" dashWidth="1" width="0.3" />
                </polygonStyle>
              </contents>
            </class>
            <class hasCustomLegendName="true" layerNo="1" legendName="&amp;gt;0.4"
 stringValue="*" upperLimit="INF">
              <contents draw="true">
                <polygonStyle drawBorder="true">
                  <fillStyle color="FF000000" hatchType="SOLIDFILLING" hatchWidth="1.5" />
                  <borderStyle color="FF000000" dashPattern="1" dashWidth="1" width="0.3" />
                </polygonStyle>
              </contents>
            </class>
          </classified>
        </standardBar>
      </item>
    </items>
  </bars>
</connectors>
<stopPoints avoidOverlapping="true">
  <display>
    <active classificationAttrID="TYPENO" numDecPlaces="0" useClassifiedMode="false"
 useLayerOrder="false">
      <uniform draw="true" drawChart="true" drawTable="true" drawUntilScale="false"
 drawUntilScaleValue="10000" pointObjectType="POINTOBJ_SYMBOL">
        <symbol drawSymbolChar="false" size="3" symbolChar="" symbolCharColor="ff000000"
 symbolDisplayType="SYMBOL_STOP_GERMAN" />
      </uniform>
    </active>
    <passive>
      <uniform draw="true" drawChart="false" drawTable="false" drawUntilScale="false"
 drawUntilScaleValue="10000" pointObjectType="POINTOBJ_SYMBOL">
```

```
          <symbol drawSymbolChar="false" size="3" symbolChar="" symbolCharColor="ff000000" symbolDisplayType="SYMBOL_CIRCLE">
            <polygonStyle drawBorder="true">
              <fillStyle color="ffa8a8a8" hatchType="SOLIDFILLING" hatchWidth="0" />
              <borderStyle color="ffa8a8a8" dashPattern="1" dashWidth="1" width="0.3" />
            </polygonStyle>
          </symbol>
        </uniform>
      </passive>
      <marked>
        <uniform draw="true" drawChart="false" drawTable="false" drawUntilScale="false" drawUntilScaleValue="10000" pointObjectType="POINTOBJ_TEXT">
          <text attrID="NO" drawFrame="true" frameColor="ffff4040" numDecPlaces="0" roundValue="1" textColor="ffff4040" textSize="1.8" transparent="false">
            <fontStyle bold="false" fontFamilyName="Arial" italic="false" />
          </text>
        </uniform>
      </marked>
      <traversed>
        <uniform draw="true" drawChart="false" drawTable="false" drawUntilScale="false" drawUntilScaleValue="10000" pointObjectType="POINTOBJ_SYMBOL">
          <symbol drawSymbolChar="false" size="2" symbolChar="" symbolCharColor="ff000000" symbolDisplayType="SYMBOL_SQUARE">
            <polygonStyle drawBorder="true">
              <fillStyle color="ffff4040" hatchType="SOLIDFILLING" hatchWidth="0" />
              <borderStyle color="ffff4040" dashPattern="1" dashWidth="1" width="0.3" />
            </polygonStyle>
          </symbol>
        </uniform>
      </traversed>
    </display>
    <table constantColor="ff000000" draw="false" drawFrame="true" frameColor="ff000000" relTextSizeDistance="30" tableAlignment="ADJUSTMENT_TOPCENTER" tableDistance="0" textAlignmentColumn1="ADJUSTMENT_TOPLEFT" textAlignmentColumn2="ADJUSTMENT_TOPRIGHT" textSize="2" transparent="false" useConstantColor="true" />
    <chart autoScaleFactor="1" chartDisplayType="COLUMNS" columnWidth="3" displayNegativeValues="false" draw="true" drawLabel="true" drawTextFrame="false" sameScaleForAllColumns="true" scaleMaxColumnHeight="30" scaleMaxPieArea="1000" scaleMaxValueColumn="1000" scaleMaxValuePie="1000" scaleMinValueColumn="0" scaleMinValuePie="0" textColor="ff000000" textColorFromChartItem="false" textSize="2" useAutoScale="true">
      <fontStyle bold="false" fontFamilyName="Arial" italic="false" />
      <textFrameStyle drawBorder="false">
        <fillStyle color="ffffffff" hatchType="SOLIDFILLING" hatchWidth="1.5" />
      </textFrameStyle>
      <items>
        <item displayAsBaseColumn="false" displayThreshold="0" draw="true" gridColumnWidth="150" numDecPlaces="0" roundValue="1" scaleAttrID="PUTUE_BOARD_FLOW_1_BUS_1" scaleMaxColumnHeight="30" scaleMaxValueColumn="1000" scaleMinValueColumn="0" stringFormatType="DEFAULTFMT" useDisplayThreshold="false">
          <polygonStyle drawBorder="true">
            <fillStyle color="FFFF0000" hatchType="SOLIDFILLING" hatchWidth="0" />
            <borderStyle color="00000000" dashPattern="1" dashWidth="1" width="0.3" />
          </polygonStyle>
          <negativeValuePolygonStyle drawBorder="true">
            <fillStyle color="00000000" hatchType="SOLIDFILLING" hatchWidth="0" />
            <borderStyle color="00000000" dashPattern="1" dashWidth="1" width="0.3" />
          </negativeValuePolygonStyle>
        </item>
        <item displayAsBaseColumn="false" displayThreshold="0" draw="true" gridColumnWidth="150" numDecPlaces="0" roundValue="1" scaleAttrID="PUTUE_ALIGHT_FLOW_1_BUS_1" scaleMaxColumnHeight="30" scaleMaxValueColumn="1000" scaleMinValueColumn="0" stringFormatType="DEFAULTFMT" useDisplayThreshold="false">
          <polygonStyle drawBorder="true">
            <fillStyle color="FFCC0000" hatchType="SOLIDFILLING" hatchWidth="0" />
            <borderStyle color="00000000" dashPattern="1" dashWidth="1" width="0.3" />
          </polygonStyle>
          <negativeValuePolygonStyle drawBorder="true">
            <fillStyle color="00000000" hatchType="SOLIDFILLING" hatchWidth="0" />
            <borderStyle color="00000000" dashPattern="1" dashWidth="1" width="0.3" />
          </negativeValuePolygonStyle>
        </item>
        <item displayAsBaseColumn="false" displayThreshold="0" draw="true" gridColumnWidth="
    150" numDecPlaces="0" roundValue="1" scaleAttrID="PUTUE_WAIT_CONG_1_BUS_1"
    scaleMaxColumnHeight="30" scaleMaxValueColumn="1000" scaleMinValueColumn="0"
    stringFormatType="DEFAULTFMT" useDisplayThreshold="false">
          <polygonStyle drawBorder="true">
            <fillStyle color="FF8F0000" hatchType="SOLIDFILLING" hatchWidth="0" />
            <borderStyle color="00000000" dashPattern="1" dashWidth="1" width="0.3" />
          </polygonStyle>
          <negativeValuePolygonStyle drawBorder="true">
            <fillStyle color="00000000" hatchType="SOLIDFILLING" hatchWidth="0" />
            <borderStyle color="00000000" dashPattern="1" dashWidth="1" width="0.3" />
          </negativeValuePolygonStyle>
        </item>
      </items>
    </chart>
  </stopPoints>
</netEditor2D>
<printParameters>
  <contents drawTileGlueLine="false" expandToPrintPage="false" outputArea="CURRENTVIEW"
    outputSize="FITTOPAGE" oversizeHandling="CUT" printMarkings="true"
    scaleAbsoluteMillimeterSizes="false" scaleFactor="1" tileOverlapping="0" />
  <pageSetup bottomMargin="0" centerHorizontally="true" centerVertically="true" leftMargin="0"
     orientation="AUTOMATIC" rightMargin="0" topMargin="0" />
  <printArea bottomMargin="0.005" fixedAspectRatio="1" leftMargin="−0.005" rightMargin="0.005"
     showPrintAreaOnScreen="false" topMargin="−0.005" useFixedAspectRatio="false" />
</printParameters>
</netEditorGraphicParameters>
```

LISTING A.2: *OneLine_ThreeStops* network, XML file representing the graphics to import in Visum in order to represent results.