Heuristic Methods for Calculating Dynamic Traffic Assignment

Mostafa Ameli

N° d'ordre NNT: 2019LYSET009

THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON

Opérée au sein de
l'École Nationale des Travaux Publics de l'État

École Doctorale N° 162
MEGA (Mécanique, Energétique, Génie Civil et Acoustique)

Spécialité / discipline de doctorat : Génie Civil

Méthodes heuristiques pour le calcul d'affectation dynamique du trafic

---

Soutenue publiquement le 31 octobre 2019 par :

Mostafa AMELI

Devant le jury composé de :

Hani MAHMASSANI	Full Prof. (Northwestern University)	Rapporteur
Francesco VITI	Associate Prof. (Université du Luxembourg)	Rapporteur
Salima HASSAS	Prof. HDR (Université Claude Bernard Lyon 1)	Présidente du jury
Hans VAN LINT	Full Prof. (TU Delft)	Examinateur
Ludovic LECLERCQ	Prof. HDR (Univ. Lyon, ENTPE, IFSTTAR)	Directeur de thèse
Jean-Patrick LEBACQUE	Ingénieur Général de MTES (Univ. Paris-Est, IFSTTAR)	Co-directeur de thèse

Thèse préparée au
LICIT (Laboratoire Ingénierie Circulation Transports) , Lyon, France
&
GRETTIA (Geńie des Reśeaux de Transport Terrestres et Informatique Avanceé), Paris, France

N° d'ordre NNT: 2019LYSET009

A THESIS OF THE UNIVERSITY OF LYON

Prepared at
École Nationale des Travaux Publics de l'État

Doctoral school No. 162
MEGA (Mechanics, Energy, Civil Engineering and Acoustics)

To obtain the graduation of
PhD in Civil Engineering

Heuristic Methods for Calculating Dynamic Traffic Assignment

Defended on October 31, 2019 by:

Mostafa AMELI

In front of the following examination committee:

Hani MAHMASSANI	Full Prof. (Northwestern University)	Reviewer
Francesco VITI	Associate Prof. (Université du Luxembourg)	Reviewer
Salima HASSAS	Prof. HDR (Université Claude Bernard Lyon 1)	Committee chair
Hans VAN LINT	Full Prof. (TU Delft)	Examiner
Ludovic LECLERCQ	Prof. HDR (Univ. Lyon, ENTPE, IFSTTAR)	Supervisor
Jean-Patrick LEBACQUE	General Engineer at MTES (Univ. Paris-Est, IFSTTAR)	Supervisor

Thèse préparée au
LICIT (Laboratoire Ingénierie Circulation Transports) , Lyon, France
&
GRETTIA (Geńie des Reśeaux de Transport Terrestres et Informatique Avanceé), Paris, France

"Learn from yesterday, live for today, hope for tomorrow. The important thing is not to stop questioning."

— Albert Einstein

Theoretical physicist

Abstract

Transport systems are dynamically characterized not only by nonlinear interactions between the different components but also by feedback loops between the state of the network and the decisions of users. In particular, network congestion affects both the distribution of local demand by modifying route choices and overall multimodal demand. Depending on the conditions of the network, they may decide to change for example their transportation mode. Several equilibria can be defined for transportation systems. The user equilibrium corresponds to the situation where each user is allowed to behave selfishly and to minimize his own travel costs. The system optimum corresponds to a situation where the total transport cost of all the users is minimum.

In this context, the study aims to calculate route flow patterns in a network considering different equilibrium conditions and study the network equilibrium in a dynamic setting. The study focuses on traffic models capable of representing large-scale urban traffic dynamics. Three main issues are addressed. First, fast heuristic and meta-heuristic methods are developed to determine equilibria with different types of traffic patterns. Secondly, the existence and uniqueness of user equilibria are studied. When there is no uniqueness, the relationship between multiple equilibria is examined. Moreover, the impact of network history is analyzed. Thirdly, a new approach is developed to analyze the network equilibrium as a function of the level of demand. This approach compares user and system optimums and aims to design control strategies in order to move the user equilibrium situation towards the system optimum.

Résumé

Les systèmes de transport sont caractérisés de manière dynamique non seulement par des interactions non linéaires entre les différents composants, mais également par des boucles de rétroaction entre l'état du réseau et les décisions des utilisateurs. En particulier, la congestion du réseau impacte à la fois la répartition de la demande locale en modifiant les choix d'itinéraire et la demande multimodale globale. Selon les conditions du réseau, ils peuvent décider de changer, par exemple, leur mode de transport. Plusieurs équilibres peuvent être définis pour les systèmes de transport. L'équilibre de l'utilisateur correspond à la situation dans laquelle chaque utilisateur est autorisé à se comporter de manière égoïste et à minimiser ses propres frais de déplacement. L'optimum du système correspond à une situation où le coût total du transport de tous les utilisateurs est minimal.

Dans ce contexte, l'étude vise à calculer les modèles de flux d'itinéraires dans un réseau prenant en compte différentes conditions d'équilibre et à étudier l'équilibre du réseau dans un contexte dynamique. L'étude se concentre sur des modèles de trafic capables de représenter une dynamique du trafic urbain à grande échelle. Trois sujets principaux sont abordés. Premièrement, des méthodes heuristiques et méta-heuristiques rapides sont développées pour déterminer les équilibres avec différents types de trafic. Deuxièmement, l'existence et l'unicité des équilibres d'utilisateurs sont étudiées. Lorsqu'il n'y a pas d'unicité, la relation entre des équilibres multiples est examinée. De plus, l'impact de l'historique du réseau est analysé. Troisièmement, une nouvelle approche est développée pour analyser l'équilibre du réseau en fonction du niveau de la demande. Cette approche compare les optima des utilisateurs et du système et vise à concevoir des stratégies de contrôle afin de déplacer la situation d'équilibre de l'utilisateur vers l'optimum du système.

ACKNOWLEDGEMENTS

In the beginning, I would say thanks to my supervisors Prof. Ludovic Leclercq, deputy director of the LICIT lab, and Prof. Jean-Patrick Lebacque, director of the GRETTIA lab, for continuous support of my PhD study, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having better advisors and mentors for my PhD study. I truly enjoyed working in a research environment that stimulate original thinking and initiative, which they created.

I would like to also express my deep and warm acknowledgments to Prof. Hani Mahmassani from Northwestern University, and Prof. Francesco Viti from University of Luxembourg, who both thoroughly reviewed my thesis and raised detailed comments that improved this revised version of the manuscript. I am also very thankful to all the other members of the examination committee: Prof. Van Lint from TU Delft and Prof. Salima Hassas from Université Claude Bernard Lyon 1, who chaired the committee.

Moreover, these years spent at the GRETTIA and LICIT labs wouldn't have been so exciting without the help and support of all my colleagues that become my friend during this PhD. To Milad, Younes, Florian, Demeng, Haroun, Mohammad Hassan, Moncef, Navid, Adel, Abdullah, Amine, Cyril and Xavier at GRETTIA Humberto, Jean, Nicolas, Delphine, Ruiwei, Mahendra, Anna, Manon, Cyril, Omar and Carlos at LICIT; many thanks for having spent very good times together! It was a real pleasure to work with you. I am very grateful to all the other lab members as well: Régine, Neila, Latifa, Zoï, Simon, Habib, Mahdi, Nadir, Allou, Etienne and Olivier at Ifsttar Marne-la-Vallee, and Nour-Eddin, Christine, Aurélien, Angelo, Andres at Ifsttar Bron.

I have special thanks to Cécile Becarie, Olivier Tonck, Clement Picq, Guilhem Mariotte and Sergio Filipe Assuncao Batista for stimulating discussions and their great helps. I also thank Sonia Cenille (ENTPE), Anne-Christine Demanny (LICIT), Marie-Laure Poiret (GRETTIA) the three secretaries of the labs and Mustapha Tendjaoui for their respective help in many and diverse administrative tasks.

Last but not least, I would like to thank my parents Faranak and Hassan, my parents-in-law Nasrin and Dariush and my sister-in-law Negar. I consider myself nothing without

them. They gave me enough moral support, encouragement and motivation to accomplish the personal goals. I also thank my dear cousin Mohammad Reza and my friend Farhad for their support. And Finally, I would like to say a heartfelt thanks to my wife Negin for always believing in me and encouraging me, for her feedback on my research and for always being so supportive of my work.

This PhD thesis received funding from the European Research Council (ERC) entitled MAGnUM: Multiscale and Multimodal Traffic Modeling Approach for Sustainable Management of Urban Mobility (under the European Union's Horizon 2020 research and innovation program: grant agreement No. 646592).

Table of Contents

List of Figures

List of Tables

1. General introduction

1.1 Background

1.1.1 Urban transportation systems

Over the past several decades, urban areas are expanding, and traffic conditions have gone dramatically worse, particularly in large metropolitan areas. The urban transportation system has a direct impact on the growth rate of business and on the life quality in the urban area. Road congestion in Europe costs approximately over 110 billion euros (about 1% of the GDP) per year (Christidis *et al.*, 2012), and this cost is expected to rise by 50 percent from 2013 to 2030 (Hilber & Palmer, 2014). The CEBR research institute estimates that in 2013, direct costs (fuel consumption and time wasted) and indirect costs (loss of productivity in business) due to congestion accounted for over 20 billion euros in France, and for over 124 billion dollars in the USA. Billions of man-hours are lost in congestion of the transportation system. For instance, each user of the urban transportation system in the UK, Germany, France, and the USA spends, on average, three days in gridlock every year in metropolitan areas (CEBR, 2014).

Moreover, in both developed and developing countries, urban travel demand is continuously growing while the transport infrastructure and services have lagged far behind. Besides, by increasing the price of land and rental rates in city centers, people are forced to move to the city peripherals thus daily commuting increases (Gendron-Carrier *et al.*, 2018). Therefore, studying the urban transportation system in order to reduce its level of traffic congestion is clearly of utter importance. Transportation management has a key role to play in this context, but there are many problems in urban transportation planning and management (Narayanaswami, 2016). In short-mid- term, optimizing the use of existing transportation modes, and infrastructures is probably an effective management policy to improve traffic conditions in the urban environment.

1.1.2 Traffic planning model

In this context, analyzing and forecasting mobility in a given network is vital for traffic managers. The transportation system is very complex, wherein many factors (e.g., traveler's path choice, and transport mode costs) influence the performance of the system. Every agent in the system looks for optimizing his/her own objective(s). Traffic planning models

attempt to represent the transportation system as a function of agents' decision and existing transport infrastructures. The classical traffic planning and estimation process contains four main steps with the following goals in a given traffic network (Patriksson, 2015):

1. Trip generation: estimate the number of users that travel between an origin region and a destination region.
2. Trip distribution: estimate the origin and destination points, i. e., build an Origin-Destination (OD) matrix by aggregated trip numbers from the previous step.
3. Mode split (choice): estimate the demand for transportation modes (e. g., private cars, and public transportation).
4. Traffic assignment: assign a given set of trips to a possible path (set of route and modes).

Each step of the traffic estimation process has a huge body of literature. This PhD thesis focuses on the fourth step. The traffic assignment concludes the traffic estimation process, and the result is an estimate of the state of the network based on a traffic model (Guevara *et al.*, 2011).

1.1.3 Traffic assignment

The traffic assignment problem is like a game; each player wants to win. The travelers (users) and the transportation system are the players of this game. Solving this game means to find an equilibrium if it exists Myerson (1999). The traffic assignment problem, from the early 1980s forward, have been greatly influenced by Wardrop's first and second principles Wardrop (1952) that were stated in a traffic environment. Wardrop's first principle, also known as the user optimum principle, specifies that travelers are viewed as Nash agents competing on a network for resources (e. g., road and mode capacity). The user equilibrium corresponds to the selfish situation when no user can reduce its own travel costs. Wardrop's second principle describes the system optimum conditions: the system optimum is reached when the total travel cost of all users is minimal. In Wardrop's words:

- the User Equilibrium (UE), i.e., Wardrop's first principle, is expressed as: "the journey times in all routes actually used are equal and less than those which would be experienced by a single vehicle on any unused route."
- the System optimum (SO), i.e., Wardrop's second principle, is described as the network state which "minimizes the total travel time spent in the network."

These two equilibrium principles can be extended to take into account generalized travel costs, including monetary costs (e.g., tolls and fuel consumption costs) instead of travel time. Mathematically, the UE conditions can be expressed by the following conditions:

$$f_p \geq 0 \quad ; \forall\, p, w \tag{1.1}$$

$$c_w^* \overset{def}{=} \min_{p \in w} \left(c_p\right) \quad ; \forall\, w \tag{1.2}$$

$$f_p(c_p - c_w^*) = 0 \quad ; \forall\, p, w \tag{1.3}$$

where f_p denotes the flow on path p between OD pair w; c_p represents the generalized travel cost (or travel time) of path p between OD pair w; and c_w^* is the minimum generalized travel cost path (shortest path) between OD pair w. Equation 1.1 ensures the non-negativity of the path flow. The shortest path results from Equation 1.2 for an OD pair w, it must have the minimum cost c_w^*. Finally, Equation 1.3 requires that if path p is used (i.e.$f_p > 0$), the cost on this path should be equal to the minimum travel time. If the cost variable (c_p and c_w^*) are replaced by marginal travel cost [1], the equilibrium under constraints 1.1- 1.3 is the SO equilibrium (Sbayti *et al.*, 2007).

Figure 1.1 presents the traffic assignment as a function. The input is a traffic network and OD matrix, and the output is the optimal solution of the assignment problem under an equilibrium principle. In other words, solving the traffic assignment refers to calculating the path flow distribution of the network equilibrium. As is shown in Figure 1.1, solving the assignment problem requires two steps: 1) path identification and 2) optimization.

1. Path identification requires to estimate travel costs of possible paths between OD pairs. A traffic model is used to calculate the needed variables, and expresses the dynamics of the network as a mathematical model.

2. The second step, which is the main focus of this study, aims to calculate an equilibrium path flow distribution based on the traffic model. Note that this optimization step depends on the nature of the traffic model, analytical or numerical.

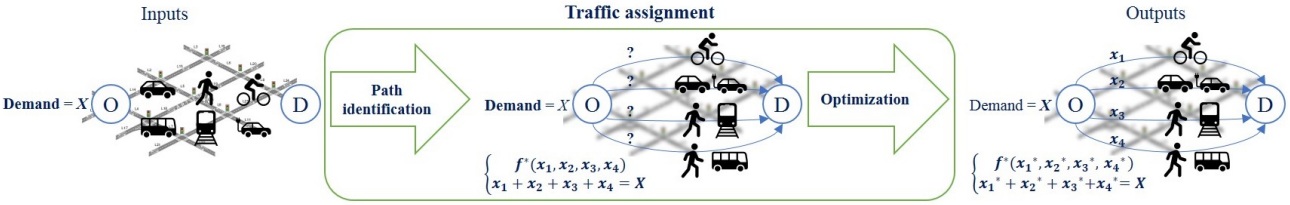


FIGURE 1.1 – Traffic assignment

When the OD matrix and the link flows are assumed to be time-independent, the problem simplifies as a Static Traffic Assignment (STA). On the other hand, if time dependence is considered, the problem becomes a Dynamic traffic assignment (DTA) (Sheffi, 1985). A small but growing literature exists on filling the gap between STA and DTA models. Quasi-dynamic traffic assignment models are in fact, a type of static model, but they are considered more capable (Bliemer *et al.*, 2012). The distinction is based on the fact that a quasi-dynamic model is often assumed to be capacity constrained (Bliemer *et al.*, 2014).

The DTA is much more complex than static assignment, both computationally and conceptually. Thus STA models are often used by practitioners for planning purposes. Indeed, although link flows are never truly time-independent, the static assignment may constitute a good approximation, describing the stationary limit of the traffic flow pattern over not too long time intervals. However, for congested networks in a peak period, this "stationary

[1] Marginal travel cost: the changes in travel cost of one unit of flow on a (time-dependent) path caused by one unit of flow on another (time-dependent) path. (Qian & Zhang, 2011)

limit" is not realistic. Among the significant shortcomings of static traffic assignment, let us mention the following:

- the failure to describe congestion correctly and to estimate correctly congestion costs on links and intersections;
- failure to take into account capacity constraints and spillbacks;
- STA allocates constant flows to paths, therefore, flows on routes sharing a common arc of the network always interact, whereas drivers on different routes might use a common link at different times;
- over-estimation of link flows.

This study focuses on DTA. Nevertheless STA will be used in some instances in order to simplify the presentation of the new proposed methodologies in traffic assignment. These will then be adapted to DTA.

1.1.4 Dynamic traffic assignment (DTA)

DTA models aim to capture the dynamic relationships between paths, time, and network characteristics (Levin *et al.*, 2014b). Regarding the history of mathematical models for DTA, there are several review papers in the literature. The first review paper by Mahmassani *et al.* (1991) discusses the DTA and traffic simulation models. Cascetta & Cantarella (1993) focuses on the DTA models developed before 1991. After a brief review of Huapu *et al.* (1996) in 1996, most of the DTA papers published before 2000 were examined by Hoogendoorn & Bovy (2001) and Peeta & Ziliaskopoulos (2001). Szeto & Lo (2005a) and Mun (2007) examine the properties of the DTA problem with different types of traffic flow models. Szeto & Lo (2005c) address the impact of considering spatial queues on the DTA problem. The same authors also provide a classification of existing DTA models in the literature (Szeto & Wong, 2012). Pel *et al.* (2012) did a review on travel behavior modelling in simulation-based DTA models. A recent review has been carried out by Wang *et al.* (2018), summarizing and examining the recent methodological advances of DTA models in practice.

Since the 1970s, DTA models were used to analyze both long-term and short-term planning issues (Han *et al.*, 2015b). DTA models have the unique capability of capturing within-day and day-to-day evolution of traffic network dynamics. DTA research consists of wide ranges of model sizes using various model types (e.g., microscopic and macroscopic). DTA models can be applied (Szeto & Lo, 2006a):

- To networks with different sizes and resolutions.
- To short-term and future long-range plans.
- To conclude and tune the travel demand estimation process.
- To conduct operational analysis on design improvements.
- For various time periods and time intervals.

DTA models can represent any given traffic network even though the challenge is the computation cost of the equilibrium calculation. According to the level of details that is provided by traffic flow model in order to represent the network's characteristics (e.g., size and loading) and dynamics (e.g., mode cost), the computation cost of DTA models can be different. Figure 1.2 presents a brief overview of how the computation cost of finding a network equilibrium is changed based on their characteristics and the input network. The DTA models are classified regarding their operating scale (vehicle level, link level, or network level). Moreover, the scale of the input network has a direct impact on the calculation time of the equilibrium.

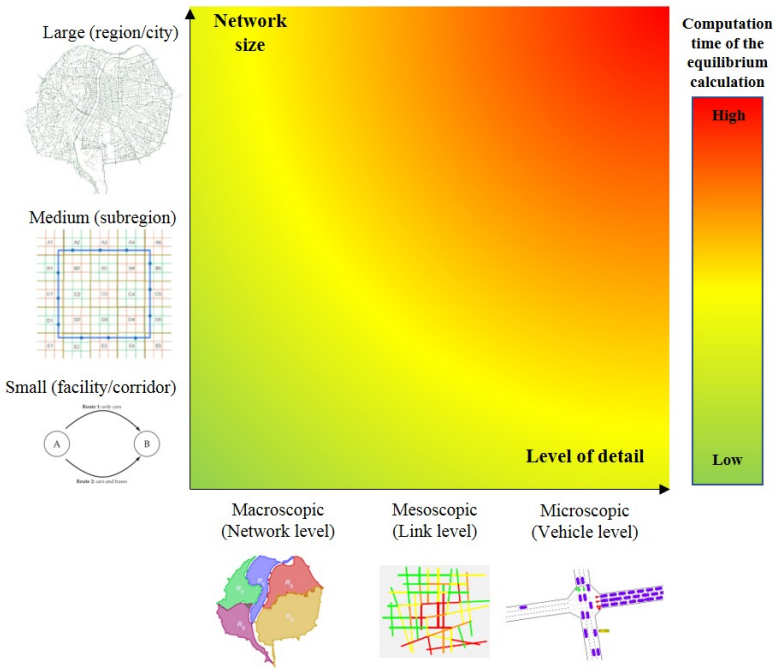


FIGURE 1.2 – The impact of DTA model's detail level and network size on the computation time of calculating a network equilibrium

There are two approaches to solve DTA problems: analytical approach and simulation-based approach. The analytical approach, (reviewed by Boyce *et al.* (2001)), is very accurate but can only be applied in practice to small or medium networks with few ODs. Because we need to consider all path cost functions per OD and also take all the impacts of travel paths and modes into account (Szeto & Wong, 2012). In a large-scale network when we have more and more paths per OD pairs and also a large number of ODs, the problem becomes almost intractable analytically because of multiple flow exchanges at nodes. For this reason, this study uses the simulation-based approach in order to address the question of DTA in large-scale networks.

1.2 Simulation-based DTA

1.2.1 Overview

Simulation-based DTA is an effective tool for analyzing transportation systems for both operational and planning purposes (Schreiter *et al.*, 2012). Dynamic Network Loading (DNL)

is the combination of DTA with a traffic simulator that calculates network states and travel times (Yu *et al.*, 2008). In other words, DTA models depend on a network performance module, which is called DNL. The DNL sub-problem aims at describing and predicting the spatial-temporal evolution of traffic flows on a network by introducing appropriate dynamics to flow propagation and travel delays on a network level (Chong & Osorio, 2017). The DNL operator usually is not available in closed form because it severely complicates equilibrium calculation (Song *et al.*, 2017). The outputs of the DNL model and their computation time are heavily affected by the types of the chosen link model and node model. As is shown in Figure 1.2, based on the level-of-detail with their presentation of traffic, network loading models are typically classified into macroscopic, mesoscopic, and microscopic (Rakha & Tawfik, 2009).

Form another point of view, traffic simulators can be divided into two classes: Flow-based models, which consider the flow of each path and Trip-based models, which define how many travelers take each path. Macroscopic traffic flow models fall into the first category, while microscopic models belong to the second. In other words, the flow-based models have a continuous solution space while trip-based ones have a discrete solution space (Ramadurai & Ukkusuri, 2011). The flow-based approach is good to simplify the problem and optimization process in order to take the advantages of continuous models. The macroscopic approach and flow-based models usually fast in equilibrium calculation as the path flow discipline is more flexible (flows are not necessarily equivalent to vehicle units), but without adding integrality constraints, they are less realistic for OD pairs with low demand as vehicles are split into parts in practice. In this study, we decide to focus on the trip-based approach in which each vehicle is reproduced individually. Microscopic traffic simulators are now widely used for operational studies, and we have chosen to focus on DTA performance for this kind of model. Trip-based DNL attempts to assign particle-discretized time-dependent origin/destination flows in a dynamic network equilibrium framework (Jayakrishnan & Rindt, 1999).

Regarding the real-world application, there are several computer packages in the literature. Jeihani (2007) reviewed the DTA models used in some well-known computer packages such as VISSIM (Fellendorf, 1994), TRANSIMS (Smith *et al.*, 1995), PARAMICS (Cameron & Duncan, 1996), DYNASMART (Hawas *et al.*, 1998, Mahmassani, 2001, Mahmassani & Abdelghany, 2002), DynaMIT (Ben-Akiva *et al.*, 1998), CONTRAM (Taylor, 2003). MATSIM (Cetin, 2005) is also well-known in the literature. For the detailed mathematical formulations and implementations of DTA models or components, readers are referred to the wealth of literature listed above.

1.2.2 Symuvia

In this work, we use the Symuvia[2] platform for trip-based dynamic simulation in order to calculate travel costs in the network for any given path flow distributions for all OD pairs. Symuvia gives access to the position, speed, and acceleration of each vehicle (user) on the network. Symuvia contains a microscopic simulator based on the Lagrangian resolution of the LWR (Lighthill Whitham Richards) model (Leclercq *et al.*, 2008) which is the conservation law concerning traffic density. Vehicle movements at the microscopic scale are gov-

[2]Note that Symuvia is an open-source computer package that will be made available starting winter 2020. https://www.licit.ifsttar.fr/linstitut/cosys/laboratoires/licit-ifsttar/plateformes/symuvia/

erned by a set of rules, including car-following modeling (Leclercq, 2007a,b), lane-changes by using a macroscopic theory of vehicle lane-changing inside microscopic models (Laval & Leclercq, 2008) and specific movements based on noise modeling at intersections (Chevallier & Leclercq, 2007). This car-following law has been further extended to account for all the features of urban traffic: multi-class (Leclercq & Laval, 2009), signalized and unsignalized intersections (Chevallier & Leclercq, 2009a), roundabouts (Chevallier & Leclercq, 2009b). The simulation time-step is equal to 1 second, and we retrieve the travel time information at the link and node level every 1 minute.

This study developed a new module called "SymuMaster" which is implemented in Symuvia platform in order to implement solution algorithms to calculate and analyze the network equilibrium.

1.2.3 SymuMaster

SymuMaster is a portable command module to execute the simulation-based DTA solution methods. This module is shown in Figure 1.3. SymuMaster can be connected to different simulators and demand models. All the information about travel costs is stored in a network graph and optimizer directly work on a network graph and send the optimization output to assignment command. Then SymuMaster provides the inputs for (a) simulation run(s) and sends it to the simulator (Symuvia in this study). The users' routes are determined by the DTA model and the rolling horizon technique (Mahmassani, 2001) which determine the path flow distribution based on a prediction period and an assignment period (Mahmassani, 1998).

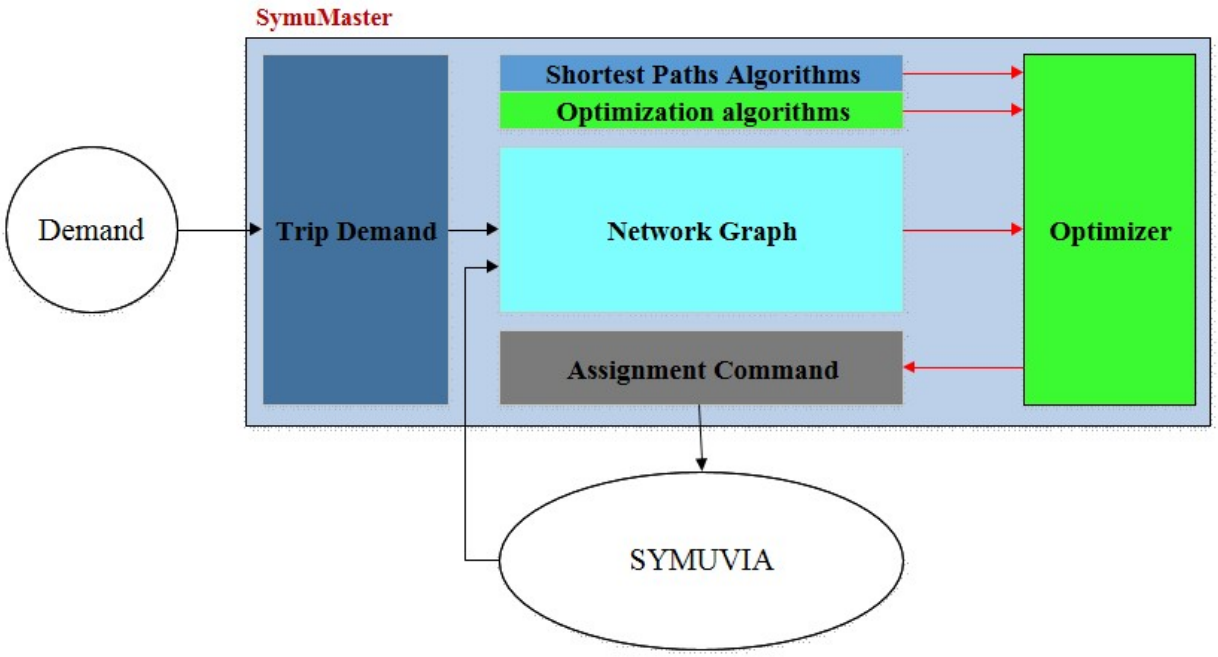


FIGURE 1.3 – SymuMaster

This study focuses on the optimizer box (Green boxes in Figure 1.3), which determines the assignment pattern for each simulation step during the optimization process.

1.3 Equilibrium calculation

Finding an equilibrium through simulation (when no closed-form analytical solution is available) typically involves a solution scheme that relies on an iterative procedure. Figure 1.4 presents the simulation-based process to solve the DTA problem. The optimization process starts with reading the network and the demand profile, and then it considers an initial assignment pattern as an input for the problem. After, the traffic simulator, simulates the users travel in the traffic network according to the inputs. The goal is to update the user travel times based on the inflow of paths in the network. The quality of the solution is measured in "End condition," and if it does not meet the condition, the equilibration process is executed.

Same as traffic assignment Figure 1.1, the DTA solver also contains two main steps: (i) the optimization phase in order to find the optimal assignment pattern; (ii) the reassignment phase to create an assignment pattern for the next simulation. For the first step, several studies have proposed multiple path selection models by considering the time and dynamics of the network, e.g., (Ziliaskopoulos & Mahmassani, 1993, Jayakrishnan *et al.*, 1994, Ziliaskopoulos & Mahmassani, 1996, Ding *et al.*, 2008, Sun *et al.*, 2017, Xie *et al.*, 2018). The second step depends on the user behavior rule we want to adopt, which leads to different definitions of the network equilibrium (e.g., UE or SO).

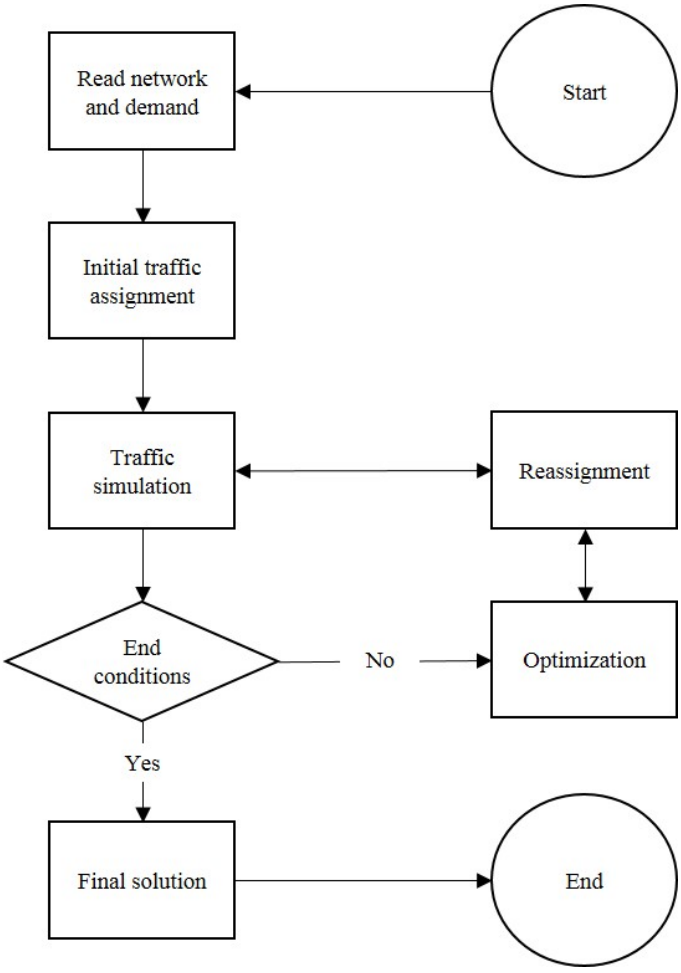


FIGURE 1.4 – General framework to solve simulation-based DTA

Implementing UE discipline as the most straightforward behavioral rule for DNL is far from trivial (Lin *et al.*, 2011). In small networks with less complexity, the DTA process works well converging to equilibrium. As the scale of a model grows in size, complexity, and congestion, the DTA process becomes more difficult. A key cause of degraded performance is the result of queuing that occurs in the model and which may reflect real-world conditions (Foytik *et al.*, 2017).

1.3.1 Column generation approach

As mentioned before, in large-scale DTA problems, there are three costly steps in terms of computation in simulation-based DTA models: traffic simulation, shortest path discovery, and optimization. According to state of the art, it appears that the most advanced framework for solving the simulation-based DTA problem as a black-box optimization problem is based on column generation approach.

The idea of the column generation approach is based on the generation of the set of paths to which users may be assigned. Classical approach starts from an empty set and augment it every iteration as required. Larsson & Patriksson (1992) suggested that a more efficient path set can be created if column-generation principles are applied. The column generation approach contains two loops: outer loop and inner loop. The outer loop is responsible for path discovery while the inner loops implement the path flow optimization for a given path set. Note that the classic approach executes both steps in one top loop. In large-scale network problems, it is extremely costly to keep the data of all possible paths between each OD pair, so, column generation approach definitely reduces memory-requirements Levin *et al.* (2014b). Lu *et al.* (2009) implement and examine the column generation approach in DTA context and show that it not only reduces memory-requirements but also outperforms other algorithms in convergence with a designed swapping algorithm embedded in the inner loop. Next section reviews the most common swapping algorithms in the literature.

1.3.2 Swapping algorithms

STA inspires almost all swapping algorithms for DTA. We can classify all existing solution methods to exact and heuristic algorithms. In the static context, first, we find the cheapest path (e.g., using Dijkstra's algorithm) at the beginning of each iteration and shift a portion of the demand to the (newly) shortest path which is called reassignment process. The optimal portion of demand to shift can be determined analytically when the cost function is link additive and strictly concave. The exact algorithms can be classified into three categories:

- The Path Equilibration (PE) (Dafermos, 1969).
- The Frank-Wolfe algorithm(Frank & Wolfe, 1956) and it's extensions (e.g., LeBlanc *et al.* (1975), Mitradjieva & Lindberg (2013).
- The origin-based or destination-based algorithms: Algorithm B or bush algorithm (Dial, 2006), Linear User Cost Equilibrium (LUCE) algorithm (Gentile, 2012), Nie's algorithm (Nie, 2010), and The original Physarum-inspired algorithm (Xu *et al.*, 2018)

While in practice, reassignment is often based on heuristics. The reason to resort to heuristics is based on the fact that the simulator needs to know the path flow distribution in

order to predict the travel time accurately while the DTA process requires this information to estimate the path flow distribution (Bekhor *et al.*, 2009). Mathematically, this problem corresponds to a fixed-point search, which requires an iterative solution method to converge. Transforming the DTA problem into a fixed-point problem allow using a large number of algorithms. The main idea stems from the theory of fixed-point re-statement (Xu, 2002). Since one run of the traffic simulator is computationally expensive, in particular for a large-scale network, in the field of transportation, it is essential to use an efficient algorithm to solve the fixed-point problem.

In the context of DTA, Method of Successive Average (MSA) algorithm remains by far the most widely used solution method in simulation-based DTA (Peque *et al.*, 2018). The simplicity of the MSA implementation and the non-requirement of derivative information are the main reasons for its widespread use (Sbayti *et al.*, 2007). The MSA algorithm shifts a predetermined fraction of users to shortest path(s). The predetermined fraction is called step size, and it is equal to $\frac{1}{i+1}$ for the MSA algorithm, where i is the iteration index and start from zero. Sbayti *et al.* (2007) and Mahut *et al.* (2008) observed that using the MSA step size is a source of inefficiencies because some OD pairs may be closer to convergence than others in intermediate iterations of the process and also later assignment intervals are typically further away from convergence. Based on this indication, several approaches are proposed to improve the determination of the step size (e.g., Sbayti *et al.* (2007), Mahut *et al.* (2008), Florian *et al.* (2008), Liu *et al.* (2009)).

Another category of heuristic methods is gradient projection and reduction methods. The analytical principles of gradient projection algorithm are presented for STA by (Bertsekas & Gafni, 1983) and for DTA models by (Szeto & Lo, 2006a). The gradient-based methods are working in an optimization problem with a differentiable objective function, which is not the case in simulation-based DTA models. Lu *et al.* (2009) propose a reformulation of DTA problem based on a gap function (The difference between path travel time and shortest path of corresponded OD pair). This development provides the basis for applying gradient-based heuristics in many studies (e.g., (Zhou *et al.*, 2008, Verbas *et al.*, 2015, 2016a,b, Halat *et al.*, 2016)). The step size is defined per path, and the gap is used as the gradient direction for determining the step size. Lu *et al.* (2009) embedded the gap-based step size in a column generation framework (section 1.3.1) and showed that it outperformed MSA in several experiments conducted on small- and medium-size networks. In opposite, Tong & Wong (2010) compare the convergence of the gradient-based procedure to that of MSA on a small network under different demand scenarios. Their results do not show significant differences among methodologies, although the gradient-based approach is observed to lead to a slightly lower gap in less congested scenarios. Mounce & Carey (2011) and Han *et al.* (2019) proposed new formulas based on the projection method, but their work is based on a continuous DTA.

Almost all the mentioned studies propose a new solution algorithm based on MSA or gradient projection algorithm and compare their proposed algorithm to the classical MSA or gradient projection algorithm, but there is no study that show which swapping algorithm is the best algorithm for large-scale application. Moreover, most of the algorithms are developed based on designed DTA models while several algorithms exist for other optimization problems (e.g., vehicle routing problem) that can be applied to simulation-based DTA.

1.4 Network equilibrium analysis

Analyzing the properties of a network equilibrium (e.g., uniqueness and stability) can help to have a better view about network state and the effect of any variation in the network. The issue of unicity for UE has long been a subject of concern in the literature on traffic assignment problems (Beckmann *et al.* (1956); Daganzo (1985); Mounce & Smith (2007); Iryo & Smith (2018)). Much research has been performed on the unicity of STA solutions with several assumptions and limitations on the traffic network model (Netter (1972); Dafermos (1982); Nagurney (1984); Wynter (2001); Wie *et al.* (2002); Florian & Morosan (2014); Sun *et al.* (2014)). For DTA models, the conditions of unicity have been appropriately reviewed by Iryo (2013). In practice, the strong mathematical assumptions for unicity (e.g., FIFO or monotonicity) simply do not hold (Boyles *et al.*, 2013). Therefore, the existence of multiple equilibria can be expected mathematically for real test cases (Levin *et al.*, 2014b), but we could not find any study that shows the multiple equilibria with variation at the equilibrium state or measuring the range of multiple equilibria in practice. There are few analytical studies (Iryo, 2011, Iryo & Smith, 2018) that showed the existence of multiple equilibria in small networks in DTA context. Most of the studies focused on check the uniqueness of equilibrium and not use this property to improve the system.

DTA problems, once successfully solved, may serve as decision-support tools for traffic management; it can be used for generating traffic control strategies. While DTA models are efficient tools to represent the network sate, evaluating control strategies on simulation-based DTA can predict how effective proposed strategies are (Mahmassani, 2001). The goal of control strategies is to improve network performance. In other words, improve the objective functions of the system which may affect the utility function of users. To do so, first, we need to analyze the current state of the network and compare with an ideal situation that the system seeks. The equilibrium analysis has received extensive attention in the literature and applications in transportation planning (Yildirimoglu & Ramezani, 2019).

In this context, comparing the network equilibrium of the user selfish behavior (UE) can be compared to an optimal network state that the system would choose (SO) (Samaranayake *et al.*, 2018). There are several indicators to measure the difference between these two network states. One of the common indicators is Price of Anarchy (PoA) (Roughgarden, 2005) which is generally defined as the ratio of the worst social cost induced by selfish behavior (UE) to the optimal social cost (SO) (Youn *et al.*, 2008). Based on the review of (Van Essen *et al.*, 2016), most of the studies in this direction focused on assessing PoA based on the users' route choice behavior. Knowing the PoA is essential, but it is even more valuable to discover a proper method or strategy to reduce it. We can change the PoA by modifying the underlying network structure. For instance, closing roads to car traffic is relatively easy to implement and is, moreover, equally effective for users. One might expect that closing roads only leads to increased congestion. However, contrary to common intuition, Braess's paradox suggests that road closures can sometimes reduce travel delays (Braess *et al.*, 2005). Braess's paradox exists because the SO and user optimum react in different ways to changes in the network (Youn *et al.*, 2008). Therefore, there is room to design new frameworks to improve the system performance based on PoA analysis and Braess's paradox.

1.5 Research objectives and major contributions

1.5.1 Research questions

A huge body of literature has been dedicated to dynamic traffic assignment in the past for multiple kind of applications. From this literature review, it appears that only simulation-based approaches can face the dimensionality of large-scale urban networks. What is noticeable in latter case is that (i) most solution methods are based on the same concept, i.e., a fixed-point problem solved with MSA inspired methods and (ii) few comparisons have been proposed to test the efficient of the existing algorithms and provide guidance to users about the best settings. The last question is paramount for large-scale networks as computation times usually count in hours or even days, therefore, slight improvements may be valuable. Furthermore, exploring solutions methods from other field of optimization may also lead to breakthrough in computational times. This deserves to be tested.

Applying DTA simulation to new operational problems is also an interesting research direction. In particular, it worth noticing that as lot of DTA research papers focus on the mathematical formulation, strong assumptions are often stated to ensure unicity of the solutions. However, in practice it is now well-known that multimodal cost functions, signalized intersections that break FIFO rules and other classical features of urban networks basically violate such assumptions and may then lead to multiple equilibrium. Interestingly this has not been extensively studied in the literature except the few works from (Iryo & Smith, 2017, Iryo, 2015, 2011, 2013) because (i) the problem is then usually untracktable analytically except for very simple toy networks and (ii) simulation-based approaches are often initialized with the same starting point (all-or-nothing discipline), which results in a single solution in practice.

In this thesis, we propose to address the following research questions in order to (i) improve existing solution methods and provide clear guidance about which methods should be used depending on the network configurations and (ii) address new DTA problems inspired by real urban network settings. All these concerns are summed up in the following list of questions related to simulation-based DTA problem:

- A lot of effort has been paid in providing a new mathematical representation to solve the DTA problem, but in a sense, it is also essential to have an idea about the efficiency of different computational methods and the effective factors on the performance of solution algorithms. To do so, the question is which solution algorithm is efficient to find the DTA solution by considering the size and loading of the network? Is there a way to improve the existing methods to find a good quality solution in terms of optimality and feasible computation time for large-scale?

- The simulation-based DTA problem is a kind of NP-hard problem, where the application of the exact solution methods is infeasible at large-scale networks. Several studies have been done in order to design heuristic methods to solve DTA problems. All of the methods work iteratively and in series. They prefer to exploit a solution that satisfies the equilibrium conditions than exploring the solution space for the optimal solution. Is there a way to design a new generation of solution methods to do more exploration of the solution space? Do we have a better solution in terms of closeness to the optimal solution by exploring more the solution space?

- The unicity has been supposed to be a rare feature of the DTA solution in multimodal context, but a lot of studies actually show that the multimodal DTA model violates the unicity conditions and not really investigate the existing of the multiple solutions and how different they are in terms of path flow distribution. Do we have really different equilibrium solutions in theory and practice? How can we account these multiple equilibria in simulation-based DTA?

- In the real multimodal networks, the network design can be changed in the long term, e.g., several new transportation facilities are added to the system. There are few studies on day-to-day DTA models by considering the evolution of the network. However, none of them investigate the nonunicity and history of the network together. Does the network converge to different equilibria with different network history and the same final network design? In other words, is the current network situation sufficient to grasp the real user distribution inside the network?

- Improving the performance of the transportation system is still a matter of debate. The users are looking for user equilibrium while the system wants to achieve the system optimum. Many studies have been done to calculate the path flow distribution of both equilibrium disciplines by DTA models. But, analyzing the output of DTA models in order to be used for improving the transportation system is missing in the literature. What is the next step after finding the equilibrium? How can we analyze the network equilibrium? Can we design a strategy to move the system from user equilibrium toward the system optimum?

1.5.2 Thesis outline

The objective of this PhD thesis is to address the aforementioned research questions. The questions can be classified into two categories. The first category encompasses the questions concerning cross-comparison of the existing algorithms and developing an efficient framework to find the network equilibrium for simulation-based DTA models. The second category comprises the innovation in analyzing the equilibrium solution: investigating the multiple equilibria, the impact of the network design history on the network equilibrium, develop a framework to analyze the user equilibrium and system optimum solutions and finally design a strategy to improve the network performance. The thesis outline is built upon this classification. Thus, the manuscript is divided into two main parts: the first one focuses on the calculation of equilibrium in DTA problem and aims to improve the DTA calculation algorithms, the second one uses the algorithm to solve new DTA problems and innovates in analyzing the DTA output. In other words, the answer of the first category of questions provides us an efficient tool to compute the equilibrium, and then we use this tool to calculate fast the equilibrium in order to investigate the questions of the second category. The thesis outline is illustrated in Figure 1.5.

In the first part, Chapter 2 focuses on improving the existing solution algorithms in the literature for finding the user equilibrium considering trip-based dynamic network loading. In other words, the goal of this chapter is to optimize existing optimization algorithms. The drawbacks of existing methods are highlighted, and several solutions are proposed to overcome them and speed up the convergence. A significant contribution of this chapter is the full benchmark of all algorithms for different network size and level of saturation. A

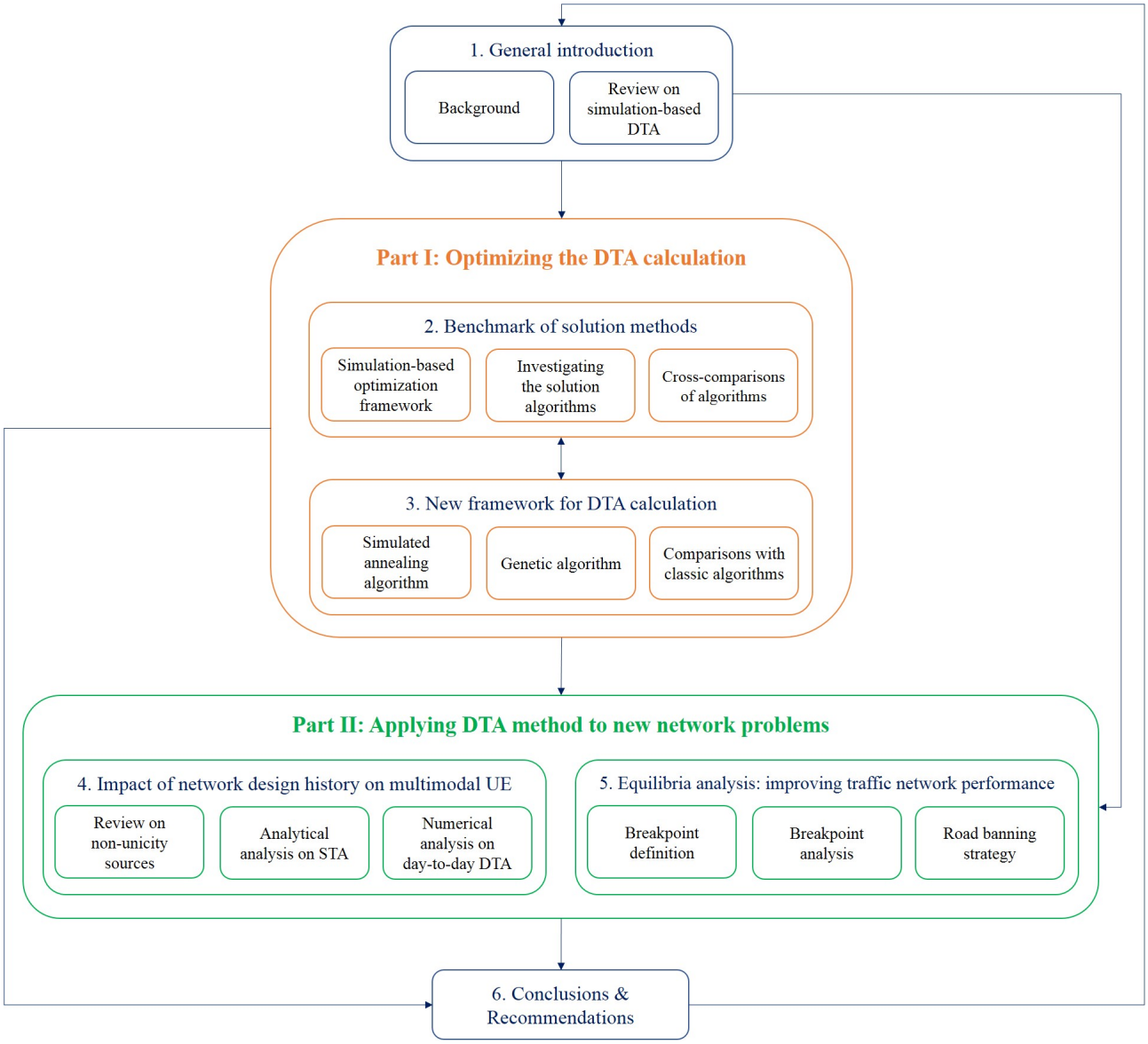


FIGURE 1.5 – Visualisation of the thesis outline

comprehensive comparison of the performance of the algorithms based on the quality of solutions and computation time is conducted.

Chapter 3 introduces a new branch of optimization algorithms for simulation-based DTA models. First, the study highlights the drawbacks of serial algorithms. Second, a new framework based on the parallel computation is proposed for solving the DTA problem. Two parallelized meta-heuristic approaches are applied to solve the network equilibrium problem: the first derived from the simulated annealing framework and the second from that of the genetic algorithm. The results show that meta-heuristic algorithms dominate classical methods in a large-scale dynamic test case.

In the second part, Chapter 4 discusses several reasons for non-unicity in multimodal user equilibrium. The study proves that network design history has an impact on the final equilibrium in static multimodal user equilibrium. Then the investigation moves to dynamic context. Multiple equilibria are observed for a dynamic multimodal user equilibrium when

changing the network design history. It is shown that the self-organization of the system leads to different network performances depending on the history of the network. This means that the study of the current network situation may not be sufficient to grasp the real user distribution inside the network. In other words, a unique UE calculation with the current network setting may lead to an equilibrium other than that resulting from the different steps corresponding to the network history. Moreover, the system can substantially reduce total travel time for some specific network history scenarios.

Chapter 5 considers static and dynamic traffic assignment to study the impact of different demand levels on three equilibria (User Equilibrium, System Optimum, and Boundary Rational User Equilibrium). It defines the concept of breakpoint as a demand level where we observe a change in the active path set of one equilibrium. The study attempts to find the breakpoints and to investigate the possibility to use breakpoint information and apply road banning strategy in order to move the system from one equilibrium to another.

Finally, the general conclusions and recommendations for future studies are presented in the last chapter of this manuscript.

This research is the result of the collaboration between two laboratories of the French institute of science and technology for transport, development and networks (**IFSTTAR**): GRETTIA (Engineering of transportation networks and advanced computing) and LICIT (Transport and Traffic Engineering Lab). The PhD candidate mainly works at GRETTIA laboratory in Paris. However, the implementation of all methods in practice is done with collaboration between the PhD candidate and the members of LICIT laboratory in Lyon. All the research in this PhD is part of the ERC project **MAGnUM** held by Prof. L. Leclercq (Multiscale and Multimodal Traffic Modeling Approach for Sustainable Management of Urban Mobility). The proposed solution methods in the thesis have been implemented in the different simulators from the **MAGnUM** project and are used by other team members for their own research.

Most of the chapters in this manuscript are updated versions of papers published or submitted in peer-reviewed journals or conference proceedings. They all consist of original works first-authored by the PhD candidate.

1.5.3 Publication list

Journal papers

- **Ameli, M.**, Lebacque, J. P. & Leclercq, L. (2019). Simulation-based dynamic traffic assignment: meta-heuristic solution methods with parallel computing. *Transportation Science*, (under first round of review).

- **Ameli, M.**, Lebacque, J. P. & Leclercq, L. (2019). Non-unicity of day-to-day multimodal user equilibrium: the network design history effect. *Transportation Research Part B: Methodological*, (under first round of review).

- **Ameli, M.**, Lebacque, J. P. & Leclercq, L. (2019). Cross-comparison of convergence algorithms to solve trip-based dynamic traffic assignment problems. *Computer-Aided Civil and Infrastructure Engineering*, (in press).

- **Ameli, M.**, Lebacque, J. P. & Leclercq, L. (2020). Improving traffic network performance with road banning strategy: a simulation approach comparing user equi-

librium and system optimum. *Simulation Modelling Practice and Theory*, 99, 101995, doi:10.1016/j.simpat.2019.101995.

Book chapter

- **Ameli, M.**, Lebacque, J. P. & Leclercq, L. (2017). Multi-Attribute, Multi-Class, Trip-Based, Multi-Modal Traffic Network Equilibrium Model: Application to Large-Scale Network, *Traffic and Granular Flow '17*, pages 487–495, Springer, doi:10.1007/978-3-030-11440-4_53 , ISBN: 978-3-030-11440-4.

Peer-reviewed conference proceedings

- **Ameli, M.**, Lebacque, J. P., Delhoum Y. & Leclercq, L. (2019). Simulation-based user equilibrium: improving the fixed-point solution methods, In *Transportation Research Board 98th Annual Meeting*, Washington DC, USA
- **Ameli, M.**, Lebacque, J. P. & Leclercq, L. (2018). Day-to-day multimodal dynamic traffic assignment: Impacts of the learning process in case of non-unique solutions, In *The 7th International Symposium on Dynamic Traffic Assignment (DTA 2018)*, Hong Kong, Chinese special administrative region.
- **Ameli, M.**, Lebacque, J. P. & Leclercq, L. (2018). User equilibrium vs. System optimum: an analysis based on demand level breakpoints, In *Transportation Research Board 97th Annual Meeting*, Washington DC, USA

International conference presentations

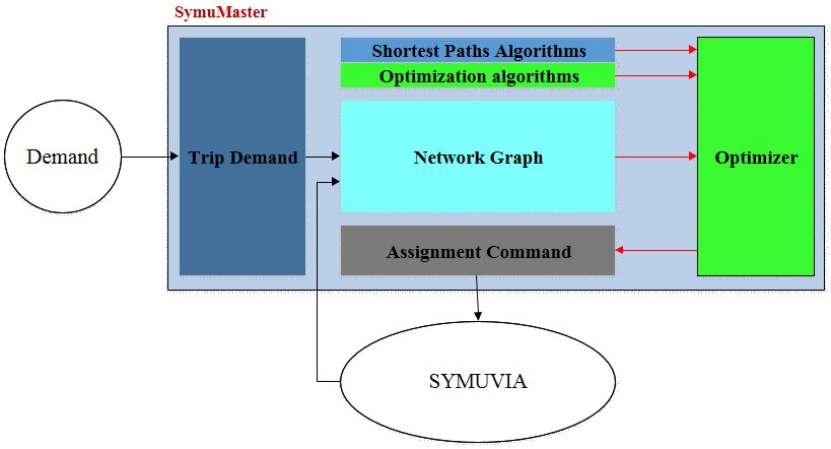
- **Ameli, M.**, Lebacque, J. P. & Leclercq, L. (2019). Applying meta-heuristic algorithms with a parallel computation framework to simulation-based dynamic traffic assignment, *the 8th Symposium of the European Association for Research in Transportation (hEART 2019)*, Budapest, Hungary.
- **Ameli, M.**, Lebacque, J. P. & Leclercq, L. (2019). Applying Meta-heuristic Algorithm with parallel computation framework to simulation-based Dynamic Traffic Assignment, *the 10th Triennial Symposium on Transportation Analysis (TRISTAN X)*, Hamilton Island, Australia.
- **Ameli, M.**, Lebacque, J. P. & Leclercq, L. (2018). Impacts of the network design history on day-to-day multimodal dynamic traffic assignment, *The 7th Symposium of the European Association for Research in Transportation (hEART 2018)*, Athens, Greece.
- **Ameli, M.**, Lebacque, J. P. & Leclercq, L. (2017). Multi-attribute, Multi-class, Trip-Based, Multi-modal Traffic Network Equilibrium Model, *The 12th Conference on Traffic and Granular Flow (TGF 2017)*, Washington, DC, USA.

Working papers

- **Ameli, M.**, Lebacque, J. P. & Leclercq, L. (2019). Computational Method for Multi-modal Multi-class Traffic Network Equilibrium Model: A Review based on Simulation Test Case, in progress.

Part I

Optimize the DTA calculation



Introduction

This first part focuses on the user equilibrium calculation. The goal is to provide an efficient tool for computing the network equilibrium for simulation-based DTA models. Then we use this tool for the equilibrium analysis in the second part.

In the present part, the investigations on solution algorithms are divided into two different studies. Chapter 2 focuses on improving the existing solution algorithms in the literature for finding the user equilibrium considering trip-based dynamic network loading. Benchmark of all existing and proposed algorithms for different network size and level of saturation allow addressing the question of which algorithm is more efficient than other ones. This investigation gives a good background for designing the new framework presented in chapter 3. Two parallelized meta-heuristic approaches are applied to solve the network equilibrium problem: the first one derived from the simulated annealing framework and the second one from that of the genetic algorithm.

Contents

2. Benchmark of solution methods

Solving a dynamic traffic assignment problem in a transportation network is a computational challenge. This chapter first reviews the different algorithms in the literature used to calculate numerically the User Equilibrium (UE) related to dynamic network loading. Most of them are based on iterative methods to solve a fixed-point problem. Two elements must be computed: the path set and the optimal path flow distribution between all origin-destination pairs. In a generic framework, these two steps are referred to as the outer and the inner loops, respectively.

This chapter aims to assess the computational performance of the inner loop methods that calculate the path flow distribution for different network settings (mainly network size and demand levels). Several improvements are also proposed to speed up convergence: four new swapping algorithms and two new methods for the step size initialization used in each descent iteration. All these extensions are compared with existing methods by numerical experiments. The impact of the network size and saturation level of demand on the performance of different components of the solution algorithm is also evaluated. Finally, the best algorithms and settings are identified for all network sizes, with particular attention being given to the largest scale.

This chapter is an updated version of the paper:

> Ameli, M., Lebacque, J. P. & Leclercq, L. (2019). Cross-comparison of fixed-point algorithms to numerically solve dynamic user equilibrium problem. *Computer-Aided Civil and Infrastructure Engineering*, (In press).

2.1 Notations for this chapter

TABLE 2.1 – Specific notations in this chapter

Notation	Definition [units]
W	Origin-Destination (OD) pairs, subset of origin $\times$ destination nodes, $W \subset N \times N$.
w	index of OD pair, $w \in W$.
$P_{w,\tau}$	set of paths for w in departure time interval τ.
$P^*_{w,\tau}$	set of shortest paths for w in departure time interval τ.
p	index of path, $p \in P_{w,\tau}$.
p^*	index of shortest path, $p^* \in P^*_{w,\tau}$.
D_w	total demand for w pair.
$Tr_{w,\tau}$	list of trips which travel for w in departure time interval τ.
$Tr_{p,\tau}$	list of trips which travel for w on path p in departure time interval τ, $Tr_{p,\tau} \subset Tr_{w,\tau}$.
tr	index of trip, $tr \in Tr_{w,\tau}$.
$C_{tr,p,\tau}$	experienced travel cost of trip tr on path p in departure time τ.
$C^*_{w,\tau}$	minimum experienced travel cost for w in departure time interval τ.
$\hat{C}_{p,\tau}$	mean travel cost of trips on path p in departure time τ.
$\hat{C}^*_{w,\tau}$	mean travel cost of trips on minimum cost path(s) of OD pair w in departure time τ.
$n(A)$	cardinality of a set A.

2.2 Motivations

Dynamic Traffic Assignment (DTA) refers to the process of: (i) identifying the relevant paths between all Origin-Destination (OD) pairs in a transportation network, and (ii) determining the path flow distribution, considering the total OD flow demand and the time evolution of traffic states inside the network. For the first step, many researches have proposed multiple path selection models by considering the time and dynamics of the network, e.g., (Jayakrishnan *et al.*, 1994, Mahmassani, 2001, Xie *et al.*, 2018). The second step depends on the user behavior rule we want to adopt, which leads to different definitions of the network equilibrium. The best known is User Equilibrium (UE) when all users try to minimize their own travel time selfishly. It corresponds to Wardrop's first principle (Wardrop, 1952), where users are assumed to be perfectly rational and have perfect information on the network's status (Miaou *et al.*, 1999), i.e. the predicted travel time on all the relevant alternatives is known at the beginning of all the users trips (Ng & Waller, 2012). Implementing this simple behavioral rule for Dynamic Network Loading (DNL) is far from trivial (Lin *et al.*, 2011). DNL is the combination of DTA with a traffic simulator that calculates network states and travel times (Yu *et al.*, 2008). The critical issue is that the simulator needs to know the path flow distribution in order to predict the travel time accurately while the DTA process requires this information to estimate the path flow distribution (Bekhor *et al.*, 2009). Mathematically, this problem corresponds to a fixed-point search, which requires an iterative solution method to converge. Transforming the DTA problem into a fixed-point problem allow using a large number of algorithms. The main idea stems from the theory of fixed-point re-statement (Xu, 2002). Since one run of the traffic simulator is computationally expensive, in particular for a large-scale network, in the field of transportation it is essential to use an efficient algorithm to solve the fixed-point problem.

Multiple algorithms have been proposed in the literature to solve this problem. The analytical approach, e.g., (Wang *et al.*, 2018), is very accurate but can only be applied in practice to small or medium networks with few ODs. Several studies proposed exact decomposition techniques (Mehrabipour *et al.*, 2019) and meta-modeling (Osorio & Bierlaire, 2013) to reduce the computational complexity of the traffic assignment problems. However, congestion patterns are almost intractable analytically due to multiple non-linear interactions inside the network (Taale & Pel, 2015). Simulation-based approaches can match any given network, but obviously the simulation time increases with the number of nodes/links and vehicles inside the network. Here, we consider the simulator as a black box to make this study compatible with any existing traffic simulation software. We then focus on the solution methods.

The general principle of the iterative method to solve the fixed-point problem is to reassign a fraction of the users at each step. The algorithm usually reassigns the part of the users who have chosen a non-optimal path because the travel time estimation was misleading (Sancho *et al.*, 2015). The critical issue is to reach a given level of convergence while minimizing the number of iterations.

The method of successive average (MSA) is the best known algorithm for solving the fixed-point problem. It was presented for the first time by Robbins & Monro (1951). The MSA is still widely used in simulation-based DTA, because it is simple to implement and does not require the derivative information of the flow cost function (Nagel & Flötteröd, 2016). The MSA updates the path flow by using the descent direction and a predetermined

step size. The descent direction (y_p^i) of the iteration (i) for path p is extracted from the auxiliary path assignments obtained by All-or-nothing discipline, i.e., everyone is put on the active shortest path. Consequently, in iteration i, the MSA algorithm swaps a fraction σ^i of users to the shortest path(s) from each non-shortest path (Equation 2.1). Mathematically, a fraction σ^i of the total number of users on non-shortest paths is added to the shortest path(s).

$$n_{p,w}^{i+1} = n_{p,w}^i + \sigma_{MSA}^i (y_{p,w}^i - n_{p,w}^i) \tag{2.1}$$

where $n_{p,w}^i$ is the number of trips for OD pair w which travel on path p in iteration i at each time step. σ_{MSA}^i denotes the step size of the MSA algorithm. The MSA step size satisfies the following (Sheffi, 1985):

$$\sum_{i=1}^{\infty} \left(\sigma^i\right)^2 < \infty \tag{2.2}$$

$$\sum_{i=1}^{\infty} \sigma^i = \infty \tag{2.3}$$

The classic MSA uses one over the iteration index plus one as a step size (Equation 2.4) to ensure the algorithm converges. The step size can be defined with respect to Equations 2.2 and 2.3.

$$\sigma_{MSA}^i = \frac{1}{i+1} \tag{2.4}$$

The first drawback of the MSA is that it swaps a fixed number of users from all non-shortest paths to the shortest one without considering the actual travel time on these paths; it does not consider the gap between the shortest and other paths. First, Sbayti *et al.* (2007) implemented the classic MSA method in trip-based DTA, using the random selection technique in view to reducing memory requirements. Second, they attempted to overcome the first drawback by proposing a criterion-based selection that ranks the users based on experienced travel time. They showed that on a real network, both methodologies were observed to converge, and that the criterion-based technique also produced a better solution than MSA in terms of closeness to optimal. However, by increasing the number of users in a large-scale network, , ranking them based on travel cost is a computationally costly approach. Moreover, Sbayti *et al.* (2007) uses the same predetermined step size as the MSA method.

The second drawback is about predetermining the step size. The step size rule pushes the process to stabilize. There may be a risk of stabilizing before reaching the optimal solution. The step size has a direct impact on the number of iterations (computation time) and convergence speed. There is no exact method for determining the step size in the literature (Szeto & Lo, 2005b, Huang & Lam, 2002). The step size does not guarantee the quality of the solution at the end, which may not be the actual UE (Levin *et al.*, 2014b). One of the goals of this chapter is to improve the performance of the optimization process by proposing new methods for step size determination in the simulation-based optimization framework. This chapter considers common fixed-point algorithms in the literature (e.g., the MSA algorithm and its extensions) and attempts to overcome the drawbacks of existing methods to improve the performance of the solution algorithm for simulation-based DTA problems.

The layout of this chapter is as follows: the next section, Problem statement, provides a discussion on the mathematical conditions for the UE solution. It also presents the two

indicators that will be used to assess the algorithm's performance. The benchmark of the solution algorithm for finding the UE is presented in the Methodology section. The improvements made to the solution algorithm with the new swapping algorithms are presented in the section Investigating the solution algorithm. The experimental design is presented in the section Numerical experiment. The results obtained are discussed in the section Results. Finally, we provide concluding remarks in the Discussion section.

2.3 Problem statement

Traffic simulators can be divided into two classes: Flow-based models, which consider the flow of each path and Trip-based models, which define how many travelers take each path. Macroscopic traffic flow models fall into the first category while microscopic models belong to the second. In other words, the flow-based models have a continuous solution space while trip-based ones have a discrete solution space (Ramadurai & Ukkusuri, 2011). Both kinds of simulators can be coupled with MSA to perform DNL. The macroscopic approach and flow-based models usually converge faster as the path flow discipline is more flexible (flows are not necessarily equivalent to vehicle units), but without adding integrality constraints, they are less realistic for OD pairs with low demand as vehicles are split into parts in practice. As mentioned earlier, we decide to focus on the trip-based approach in which each vehicle is reproduced individually. Microscopic traffic simulators are now widely used for operational studies and we have chosen to focus on DTA performance for this kind of model. Trip-based DNL attempts to assign particle-discretized time-dependent OD flows in a dynamic network equilibrium framework (Jayakrishnan & Rindt, 1999).

2.3.1 Mathematical formula for UE

Let us consider a network $G(N, A)$ with a finite set of nodes N and a finite set of directed links A. The demand is given and time-dependent. The period of interest (planning horizon) of duration H is discretized into a set of small time intervals indexed by τ ($\tau \in T = \{\tau_0, \tau_0 + \eta, \tau_0 + 2\eta, ..., \tau_0 + M\eta\}$ and $\tau_0 + M\eta = H$). η is the duration of the time intervals. In an interval τ, travel times and traffic conditions are estimated on average and are assumed constant for the DTA. Note that the departure time of users are fixed in this study. In the sequel, the minimum cost path is considered as the shortest path. According to the definition in Table 2.1, we have:

$$\hat{C}_{p,\tau} = \frac{\sum_{tr \in Tr_{p,\tau}} C_{tr,p,\tau}}{n(Tr_{p,\tau})} \quad ;\forall p \in P_{w,\tau},\ \tau \in T \tag{2.5}$$

$$\hat{C}^*_{w,\tau} = \frac{\sum_{p^* \in P^*_{w,\tau}} \sum_{tr \in Tr_{p^*,\tau}} C_{tr,p^*,\tau}}{n(Tr_{p^*,\tau})} \quad ;\forall w \in W,\ \tau \in T \tag{2.6}$$

Equation 2.5 presents the calculation of mean travel cost of path p and Equation 2.6 is the same presentation for the shortest path p^*. For each OD pair $w \in W$ and for all paths $p \in P_w$, the dynamic traffic network equilibrium conditions with given travel demand and the users' departure time for the aforementioned traffic network equilibrium problem are:

$$\hat{C}_{p,\tau} - \hat{C}^*_{w,\tau} \geq 0 \quad ;\forall w \in W,\ p \in P_{w,\tau},\ \tau \in T \tag{2.7}$$

$$n(Tr_{p,\tau})(\hat{C}_{p,\tau} - \hat{C}^*_{w,\tau}) = 0 \quad ;\forall w \in W,\ p \in P_{w,\tau},\ \tau \in T \tag{2.8}$$

$$n(Tr_{p,\tau}) \geq 0 \quad ;\forall p \in P_{w,\tau},\ \tau \in T \tag{2.9}$$

According to Constraint 2.7, the shortest path p^* has the minimum travel cost for the related OD pair. Equation 2.8 indicates that all users travel on shortest path with minimum travel cost at UE state and the flow of paths cannot be negative, according to Constraint 2.9.

Lu *et al.* (2009) extended the work of Smith, 1993, and reformulated the problem as a non-linear problem in order to minimize the gap function. The gap function is defined as the gap between average path travel time and average shortest path travel time. Consequently, the solution to this fixed-point problem is equivalent to finding the solution to the following variational inequality:

$$\sum_{w\in W}\sum_{\tau=1}^{T}\sum_{p\in P_{w,\tau}} {C^i_{w,p,\tau}}^* \left[n(Tr_{w,p,\tau}) - n({Tr_{w,p,\tau}}^*)\right] \geq 0 \tag{2.10}$$

where $n({Tr_{w,p,\tau}}^*)$ is the optimal number of trips from OD pair w that are assigned to path p at departure time τ and $n(Tr_{w,p,\tau}), n({Tr_{w,p,\tau}}^*) \in \mathcal{H}$ satisfy the equilibrium. $\mathcal{H}$ denotes the flow constraints based on D_w. In Equation 2.10, both $n(Tr_{w,p,\tau})$ and $n(Tr^*_{w,p,\tau})$ are decision variables and hence the gap function is a function of both variables. Before presenting the solution algorithm, we need to present the convergence indicators used in this study to evaluate the quality of the solution for the trip-based dynamic network user equilibrium.

2.3.2 Convergence quality

In the trip-based DTA problem, the goal is to minimize the left side of the Equation 2.8 for all paths and OD pairs. In other words, finding the UE situation is equivalent to minimizing the delay of each user compared to the optimal option of the associated OD pair (shortest path) in the network. By this definition we can define a quality indicator for solutions which is calculated as the average delay of each user (Janson, 1991):

$$AGap = \frac{\sum_{w\in W}\sum_{\tau=1}^{T}\sum_{p\in P(w,\tau)}\sum_{tr\in Tr_{p,\tau}} (C_{tr,p,\tau} - C^*_{tr,w,\tau})}{\sum_{w\in W}\sum_{\tau=1}^{T} n(Tr_{w,\tau})} \tag{2.11}$$

Note that this formula uses the optimal experienced travel time to calculate the gap rather than the path travel time. The $AGap$ is zero for the perfect UE path flow distribution, so the best optimization algorithm obtains minimum $AGap$. Equation 2.11 has physical meaning for measuring the distance between solution and UE. The second indicator is a characterized assignment violation, i. e. users that are assigned on (a) non-optimal path(s). The violation indicator is calculated by the following steps:

1. Calculate the user violation: it is defined by considering the gap of each user (UV^{tr}_w):

$$UV^{tr}_w = \begin{cases} 1; & \text{if } \frac{C_{tr,w} - C^*_w}{C^*_w} \geq \epsilon \\ 0; & \text{o.w.} \end{cases} \tag{2.12}$$

where $C_{tr,w}$ denotes the experienced travel time of the user tr who travels for OD pair w and C^*_w denotes the shortest path of OD pair w. If the gap between the user perceived travel time and the shortest path travel time is bigger than ϵ of the shortest path travel time, the user is in violation. $\epsilon = 0$ means perfect UE, but in practice with trip-based simulation it is more appropriate to set up a margin to count users that are miss-assigned.

2. Compute the OD violation: the OD pair w is in violation when more than ϵ' of the users on w are in violation. The function ODV_w defines the OD violation:

$$ODV_w = \begin{cases} 1; & \text{if } \frac{\sum_{i \in I_w} UV^i_w}{n(Tr_w)} \geq \epsilon' \\ 0; & \text{o.w.} \end{cases} \tag{2.13}$$

where Tr_w denotes the set of users, who travel for OD pair w.

3. The violation indicator of network G is the share of ODs in violation. The second indicator for the quality of solution (Violation) is defined as follows:

$$V(G) = \frac{\sum_{w \in W} ODV_w}{n(W)} \tag{2.14}$$

The value of ϵ and ϵ' are fixed at 10% in this study to evaluate the quality of the solution from a perspective different from $AGap$. Note that, similar to $AGap$, the perfect UE means $V(G) = 0$ with $\epsilon = \epsilon' = 0$.

$AGap$ is a continuous indicator that gives us the average convergence rate of the solution in a continuous way without parameter(s). As the simulation is never perfect, so we need to monitor the users at the OD level, so it is important also to define convergence indicators based on a threshold(s) in order to account for the limitation of the simulation framework. The two values for ϵ and ϵ' are chosen to define the criterion in order to measure how many ODs are in violation with corresponding to the convergence rate ($AGap$). Please note that we do not use $V(G)$ as an indicator in the convergence process and ranking the solution algorithm.

2.4 Methodology

In large-scale DTA problems, there are three costly steps in terms of computation in simulation-based DTA models: traffic simulation, shortest path discovery, and optimization. Here, we focus on the optimization step. According to the state of the art, it appears that the most advanced framework for solving the simulation-based DTA problem as a black-box optimization problem is that proposed by Lu *et al.* (2009) with a clear decomposition between outer and inner loops. The outer loop is responsible for path discovery while the inner loops implement the path flow optimization for a given path set. The classic approach executes both steps in one top loop. In large-scale network problems, it is extremely costly to keep the data of all possible paths between each OD pair. With Lu *et al.* (2009)'s framework, the simulator simply keeps the feasible paths discovered by the outer loops.

The outer loop is mainly used for path discovery. Therefore, to improve the computation time of the outer loop, the question is mainly to find a more efficient shortest path algorithm, which is not in the scope of this study. The goal of this chapter is to improve the inner loop. The solution algorithm is presented in Figure 2.1 and is detailed in the following:

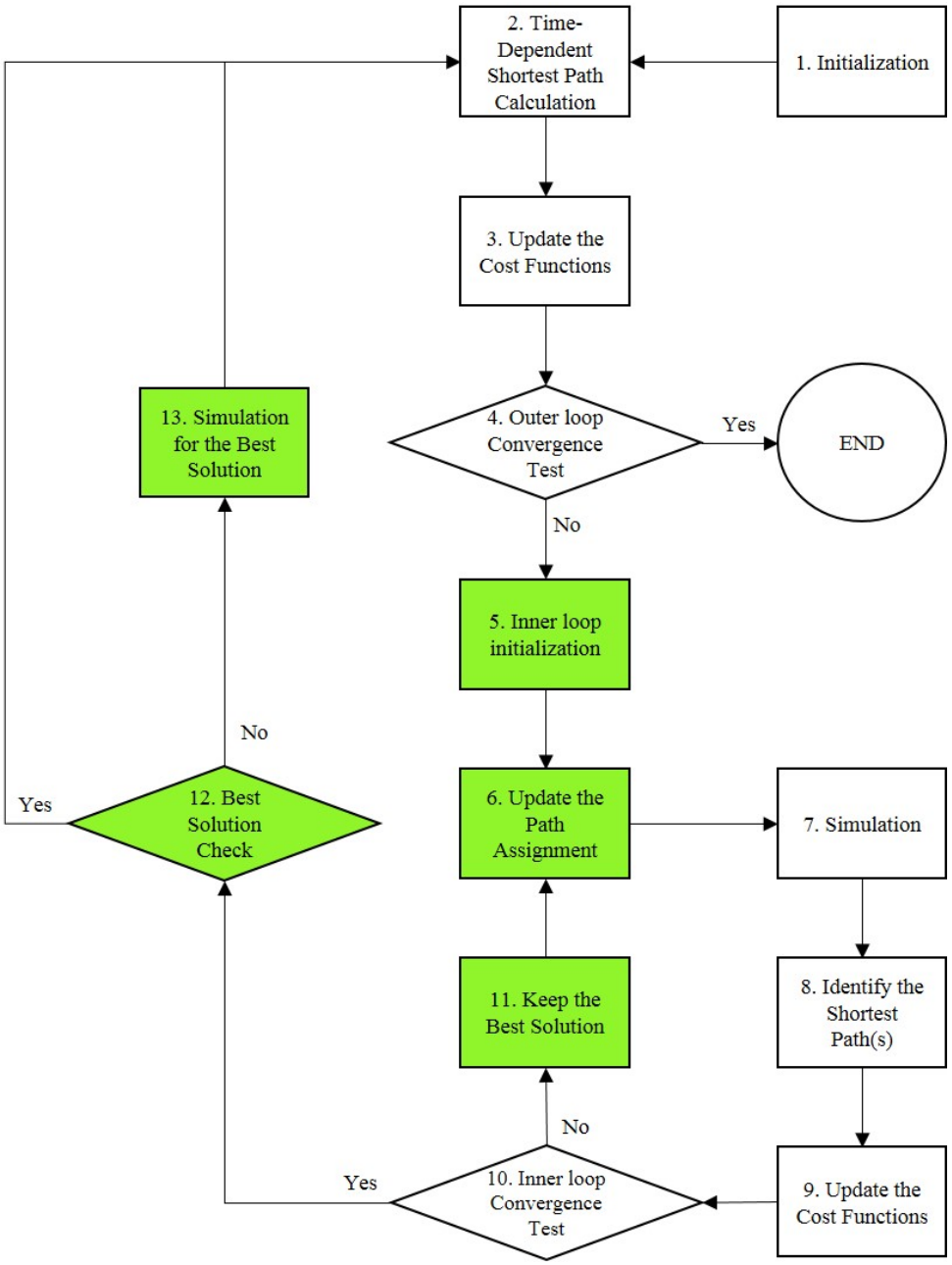


FIGURE 2.1 – Solution algorithm for trip-based dynamic network equilibrium

Step 1. **Initialization**: Load the network and the OD matrix, with real data or simple assignment models (e.g. All-or-nothing algorithm).

Step 2. **Time-dependent shortest path Calculation**: Calculate the time dependent shortest path(s) for each OD pair w.

Step 3. **Update the cost functions**: Read the information from the simulator output and update the cost of paths (links) on the network graph and update the quality indicators.

Step 4. **Outer loop convergence test**: Check the stop conditions:

a. Maximum number of outer loop iterations (j_{max}) is reached **OR**

b. No new shortest path for adding to path set **AND** good solution quality: $AGap^j \leq \lambda$; where $AGap^j$ is the $AGap$ of outer loop iteration j ($\forall j \in \{1, ..., j_{max}\}$). λ is a small and fixed value. Otherwise go to the next step.

If it has converged, the process is stopped (END). Note that the path set is a set of paths that contains used path(s) ($n(Tr_{p,\tau}) > 0$) and shortest path(s) which have been used or not ($n(Tr^*_{p,\tau}) \geq 0$) for all ODs which come from step 2.

Step 5. **Inner loop Initialization**: Load the path flow distribution of Step 1 by (Lu *et al.*, 2009) or other initialization methods (this study) in order to generate the initial assignment to start the inner loop.

Step 6. **Update the path assignment**: Swapping trips from a path to another(s) based on an optimization method in order to load the flow to the different path candidates.

Step 7. **Simulation**: Command the simulator to simulate the new assignment pattern provided by Step 6.

Step 8. **Identify the shortest path(s)**: The simulator returns the experienced travel time of all the users on different ODs. The shortest path travel time can be changed, and it is possible that we have another shortest path from the path set based on the simulation results. Note that we have a fixed path set in this step and we do not need to use the shortest path algorithm.

Step 9. **Update the cost function**: Update the network data from the simulation in Step 3.

Step 10. **Inner loop convergence test**: Check the stop conditions:

a. Maximum number of inner loop iterations (i_{max}) is reached **OR**

b. small enough variation in solution quality:

$$\frac{|AGap^i - AGap^{i-1}|}{AGap^{i-1}} \leq \Lambda \quad ; \forall i \in \{1, ..., i_{max}\} \tag{2.15}$$

where $AGap^i$ is the quality of the solution of inner loop iteration i and Λ is a fixed threshold ($\Lambda = 1\%$) for comparing the relative $AGap$.

At the end of each inner loop, if there is insufficient variation in solution quality, we converge and go to step 12. Otherwise, we continue to Step 11.

Step 11. **Keep the best solution**: Compare the solution quality ($AGap^i$) of the current inner loop iteration (i) with the best solution of the current outer loop ($AGap^j_{min}$). Note that if we are in the first inner loop iteration ($i = 1$), we consider the $AGap$ of Step 4 as the initial best and compare it with the current solution. If the solution has better

quality, we replace the best solution by the current solution. Otherwise, the best solution is kept. The solution contains the path assignment pattern and solution's $AGap$. Afterward, we continue iterating in the inner loop by going back to Step 6.

Step 12. **Best solution check**: The goal of this step is to ensure that our last solution is the best solution of the current outer loop. We compare the last solution with the current best solution from Step 11. If the last solution is the best ($AGap^i \leq AGap^j_{min}$), go to the outer loop to explore the network and find (a) new shortest path(s) (Step 2). Otherwise, a final simulation is executed in the next step for the best assignment pattern which is not the last one we obtain.

Step 13. **Simulation of the best solution**: simulate the traffic state (like Step 7) with the best assignment pattern of the inner loop provided by Step 11 to move to the next time period or the next outer loop iteration (Step 2).

2.5 Investigating the solution algorithm

In this section, we focus on improving the green boxes (in Figure 2.1) located in or added to the inner loop. They correspond to steps 5, 6, and the three new steps by this study (Steps 11-13).

2.5.1 Keeping the best solution over inner loop

Among all these steps, the first visible improvement is to keep the best assignment for the inner loop. Note that in the inner loop convergence test (Step 10), the stop conditions are the maximum number of iterations and variations of the gap, so the last iteration solution is not necessarily the best solution of the inner loop. Therefore, it is obvious that this modification improves the quality of the solution when the algorithm goes back to the outer loop. This will be even better when using trip-based approaches because their discrete nature makes them less stable and we will therefore have more cases where the solution of the last iteration is not the best over the current inner loop.

The present study adds this modification into the solution algorithm (Figure 2.1), by considering three new steps (Steps 11-13). In Step 11, the algorithm saves the assignment pattern and the value of $AGap$ for the best solution. Moreover, Step 11 compares the best solution with the current solution of the inner loop and updates the best solution. Then the best solution of Step 11 is compared with the final solution of the inner loop at Step 12. At the end of the inner loop, if the best solution is not the last one, Step 13 runs another simulation to provide the simulation results. Otherwise, the algorithm sends the last path flow distribution to the outer loop. Indeed, the cost of keeping the best solution is at worst one more simulation per outer loop.

2.5.2 Swapping algorithms

Swapping algorithms corresponds to Step 6 in Figure 2.1, where the solution algorithm updates the path assignment. The MSA algorithm (Equation 2.1) is the first swapping algorithm that we consider in this study. Here, we benchmark different algorithms that exist

in the literature and also propose new ones. Table 2.2 gives a complete summary of all the algorithms with their references.

2.5.2.1 MSA Ranking

As mentioned in Section 2.2, the first drawback of the MSA algorithm for DTA application is that the MSA does not consider the travel time of the non-shortest path and simply swaps a fixed fraction of users to the shortest path. In other words, the MSA does not take into account the quality of the path in terms of travel time. To overcome this drawback of the MSA algorithm, there is an extension of the MSA in the literature called MSA ranking (Sbayti *et al.*, 2007). The idea of MSA ranking is that the users are first ranked by experienced travel time then a maximum number of users with long experienced travel time are swapped to the shortest(s) path, based on the MSA algorithm's step size. The maximum number of swaps NS^i_{max} is observed when no users have been previously assigned to the shortest path.

$$NS^i_{max} = \sigma^i_{MSA}.D_w \tag{2.16}$$

The advantage of this algorithm is that it swaps users from the most expensive paths to the shortest path so the direction of solution searching can be improved to obtain a good solution in terms of quality for the trip-based UE problem. On the other hand, with a large number of users traveling between many ODs by many possible paths, ranking the users is a costly process in a large-scale problem. However, it is a good reference when taking all the MSA-based algorithms into account because it usually provides the best solutions (low $AGap$ and Violation values).

2.5.2.2 Projection-based algorithms

Here, we adapt two swapping algorithms with an extension of the MSA-based formula.
1- The within-day fixed point algorithm is a projection-based algorithm designed for non-linear fixed-points of non-expanding maps. We adapt this algorithm as a Projection method (PM) to the trip-based DTA problem. The validation of the algorithm for the fixed-point problem is well defined in Halpern (1967). The swapping algorithm of the Projection method is based on the transformation of the cost to the flow by a constant (α), which is the time step size of the algorithm. From the standpoint of application, the unit of α is $\frac{time}{flow}$ and measures users' sensitivity to travel costs. The swapping algorithm for trip-based DTA is as follows:

$$NS^p_{PM} = min\ \left\{n(Tr_p),\ \alpha.(\hat{C}_{p,\tau} - \hat{C}_{w,\tau})\right\} \tag{2.17}$$

where $\hat{C}_{w,\tau}$ denotes the mean travel time of all paths of OD pair w in time interval τ. At every iteration, the algorithm attempts to swap users from the path with longer travel time compared to mean travel time to the shortest path and other low-cost paths. Note that α is determined in the light of the recent study of (Han *et al.*, 2019) in which this algorithm is applied to a flow-based DTA problem. Note that in Equation 2.17, we can have a negative number for swapping, meaning that users should be added to this path. It also gives an indication of how many should be added. This only concerns paths whose path travel time is below the mean travel time ($\hat{C}_{w,\tau}$).
2- Friesz *et al.* (2011) extended the Projection algorithm by using a common method for speeding up the convergence in Hilbert space when solving the fixed-point problem. The

idea is to retrieve the initial solution at each iteration by using a weighted coefficient based on a second step size for calculating the Projection algorithm solution. Let us consider z^i as the assignment pattern in iteration i of the Projection algorithm. The solution of this algorithm, labelled as Projection Initialization (PI) algorithm, will be:

$$z^i_{PI} = \sigma^i_{PI} z^0 + (1 - \sigma^i_{PI}) z^i \tag{2.18}$$

where z^0 is the initial flow pattern at the beginning of the process and σ^i_{PI} is the step size determined by the following formula:

$$\sigma^i_{PI} = (\frac{1}{1+i})^q \tag{2.19}$$

where q is a parameter to be designated and which must satisfy $0 < q < 1$.
3- The third algorithm in this category is developed based on the Projection Initialization algorithm. The same procedure is applied for the MSA algorithm to evaluate the impact of sticking to the initial solution on the optimization process. Therefore, the flow formula of the Initialization MSA (IMSA) algorithm is as follows:

$$z^i_{IMSA} = \sigma^i_{PI} z^0 + (1 - \sigma^i_{PI}) z^i{}_{MSA} \tag{2.20}$$

where z^i in Equation 2.18 is replaced by z^i_{MSA}, which is the solution of the MSA algorithm in iteration i. As with the Projection initialization algorithm, σ^i_{PI} is determined by Equation 2.19.

2.5.2.3 Gap-based algorithm and Gap-based normalized algorithm

Lu *et al.* (2009) proposed the gap-based step size and proved that it satisfies the step size conditions. Here, we first introduce Gap-based algorithm based on gap-based step size and flow-based approach then present the implementation of Lu *et al.* (2009) for trip-based approach. The volume of swapping is proportional to the gap (the difference between the non-shortest path and shortest path cost) over the path cost multiplied by MSA step size:

$$\sigma^i_{GB} = \frac{\hat{C}_{p,\tau} - \hat{C}^*_{w,\tau}}{\hat{C}_{p,\tau}} . \rho^j_{MSA} \tag{2.21}$$

$$\rho^j_{MSA} = \begin{cases} \sigma^j_{MSA}; & \text{if} \quad i = 0 \\ 1; & \text{o.w.} \end{cases} \tag{2.22}$$

We recall that j is the outer loop iteration index. This algorithm solves the problem of sorting and also circumvents the first drawback of the MSA formula (Section 2.2). However, it also uses the step-size which can induce the convergence of the algorithm to a non-optimal solution. Moreover, the multiplication of the gap indicator and the MSA step size can provide a small step size for this algorithm. Here we normalize the gap indicator to provide a relative fraction for the step size which gives the algorithm more flexibility in swapping at each iteration. The algorithm can swap more or fewer users from path p with respect to the gap of other paths of OD pair w:

$$\sigma^i_{GBN} = \frac{\hat{C}_{p,\tau} - \hat{C}^*_{w,\tau}}{\sum_{p \in P(w,\tau)} (\hat{C}_{p,\tau} - \hat{C}^*_{w,\tau})} . \rho^i_{MSA} \tag{2.23}$$

This swapping algorithm is applied for each OD pair. This algorithm attempts to normalize the gap indicator of the Gap-Based algorithm (GB) which is called the Gap-based normalization (GBN) algorithm in this study.

2.5.2.4 Probabilistic algorithm

The Probabilistic algorithm is based on the work of Lu *et al.* (2009). We first introduced this algorithm in Ameli *et al.* (2017) for solving the multi-class multimodal equilibrium, but we never investigated it in detail. This algorithm calculates for each user the probability of swapping (SP_{tr}) by Equation 2.24. Then the Bernoulli trial is implemented in simulation for a user tr in order to decide to swap or not according to the result of the trial.

$$SP_{tr} = P(tr_{swap} = 1) = \frac{C_{tr,\tau} - C^*_{w,\tau}}{C_{tr,\tau}} \tag{2.24}$$

where tr_{swap} denotes the binary swap decision variable. This algorithm is the only one which does not use step size. Moreover, it avoids the ranking process and saves computation time. The Probabilistic algorithm can be considered as an implementation of the Gap-based algorithm to a trip-based approach if we adjust the probability of reducing the impact of the descent step. However, here we basically relax the step size. The Probabilistic algorithm is totally flexible when searching the solution space based on the probabilistic process.

2.5.2.5 Hybrid algorithms

Hybrid algorithms are different combinations of Gap-based, Probabilistic and MSA algorithms for each individual step of the calculation.
1- Halat *et al.* (2016) applied a hybrid algorithm for a dynamic activity-based model. The algorithm is a similar to the Probabilistic method as it adds a step size to Equation 2.24. This algorithm is called Step size Probabilistic (SSP), and calculates the swap probability by the following formula:

$$SP^{SSP}_{tr} = \frac{C_{tr,\tau} - C^*_{w,\tau}}{C_{tr,\tau}}.\sigma^i_{MSA} \tag{2.25}$$

The first hybrid algorithm uses a random number and compares it with SP^{SSP}_{tr} to make the swap decision for each user.
2- Lu *et al.* (2009) introduced the probabilistic algorithm for each user on each path to swap more users with high travel cost without ranking. The algorithm is called Gap-based Probabilistic (GBP) because the number of swapping user is determined by Gap-based step size. For instance, for path p, the number of swapping users is determined by Equation 2.21 and the users are selected by Equation 2.24 for swapping. Verbas *et al.* (2015) applied this algorithm algorithm for transit network assignment problems and Verbas *et al.* (2016a) showed that this hybrid algorithm obtains better solutions than the MSA algorithm in large-scale transit assignment problems.
3- The third hybrid algorithm is the Boost-up Gap-based (BGB) algorithm and is proposed by this study. The idea is to boost step size of the Gap-based algorithm. For path p, we multiply the number of swaps by a fraction of the swap number of the Gap-based and MSA

algorithms:

$$NS^{p}_{BGB} = min \left\{ n(Tr_p),\ [\sigma^{i}_{GB}.n(Tr_p)].\frac{\sigma^{i}_{GB}}{\sigma^{i}_{MSA}} \right\} \tag{2.26}$$

This section presents different alternatives for Step 6 of the solution algorithm.

2.5.3 Inner loop initialization

The solving process in the inner loop should start with an initial path flow distribution to first estimate travel times by simulation (Step 5 in Figure 2.1). The usual approach is to assign the total demand to the shortest path(s), using free-flow travel times (All-or-nothing initial assignment). In Step 5, at the beginning of each outer loop before entering the inner loop, the assignment pattern is initialized and reset to the assignment pattern in Step 1 (Lu *et al.*, 2009). We investigate an alternative approach for the assignment pattern initialization.

Keeping the assignment pattern approach removes Step 5 from the solution algorithm. Consequently, the algorithm starts the outer loop j with the optimal solution from the previous outer loop $(j-1)$. Obviously, this will be very efficient for solving static situations, but we want to investigate its performance with dynamic loading.

2.5.4 Initial step size selection

The third investigation is the definition of the descent step size. The initial step size (σ^1) defines how many users can be swapped during the first iteration. It is the largest step size during the inner loop and determines the exploration domain of the solution space. In two-level simulation-based methodology (Lu *et al.*, 2009), the initial step size of the first inner loop of the outer loop j $(\sigma^{1,j})$ is calculated by the iteration counter of the outer loop (j):

$$\sigma^{1,j}_{Initial} = \frac{1}{j+1} \tag{2.27}$$

This setting improves the speed of convergence because increasing j decreases $\sigma^{1,j}_{Initial}$, so the largest number of swaps for the current inner loops in outer loop $j+1$ begins with a smaller value in comparison with outer loop j. On the other hand, increasing j reduces the exploration domain of the inner loop. In order to overcome this drawback, this study proposes two new approaches to set up the step size.

2.5.4.1 Re-initializing the step size

The idea of re-initializing the step size (Reset) method is to reset the step size by the inner loop iteration index:

$$\sigma^{i,j}_{Reset} = \frac{1}{i+1} \tag{2.28}$$

This approach at the beginning of each outer loop starts the optimization with $\sigma^{1,j}_{Reset} = \frac{1}{2}$ to have more flexibility for searching the solution space. In other words, this approach can increase the maximum number of swaps at each iteration in comparison to the initial approach $(\sigma^{1,j}_{Reset} \geq \sigma^{1,j}_{Initial})$.

2.5.4.2 Smart step size

Here we design an approach that uses adaptive step size for each OD pair w. First, all the inner loops are initiated with the same step size whatever the OD pair. At the end of the first iteration ($i = 1$), the OD gap for OD pair w is calculated as follows:

$$Gap_w^i = \sum_{p \in P(w,\tau)} [n(Tr_{w,p}).(\hat{C}_p - \hat{C}_w^*)] \tag{2.29}$$

Then, we keep the step size and run the second loop ($i = 2$). At the end of the second loop, we update the OD gap and compare it with the previous OD gap. If the OD gap is improved, we keep the step size to swap the same fraction of users for a possibly better solution, otherwise we decrease the OD step size to decrease the number of swaps. The step size for OD pair w at inner loop iteration i is:

$$\sigma_w^i = \begin{cases} \frac{\sigma_w^{i-1}}{\sigma_w^{i-1}+1}; & \text{if} \quad Gap_w^i \geq Gap_w^{i-1} \\ \sigma_w^{i-1}; & \text{o.w.} \end{cases} ; \quad \forall i \in \{2, ..., i_{max}\}, \quad \sigma^1 = \tfrac{1}{2} \tag{2.30}$$

Equation 2.30 adapts the step size for each OD pair depending on how the quality of the solution is improved. This method mimics the Newton−Raphson method in numerical analysis (Ypma, 1995). It has been proven to be very efficient for continuous problems, but this is the first time it it is used in the context of DTA and for the discrete formulation (trip-based model).

To conclude this section, we present all the methods considered in this chapter in Table 2.2. Before comparing the methods, we need to present the dynamic trip-based simulator and test cases.

2.6 Numerical experiments

In order to conduct all experiments, we use Symuvia platform, including the trip-based simulator (Section 1.2.2) and the command module: SymuMaster (Section 1.2.3). All the algorithms and methods are implemented in SymuMaster. A configuration file is designed wherein all the settings for an experiment can be selected. The configuration file works like a menu where the swapping algorithms, initial step size selection, and inner loop selection can be selected from the list of implemented modules. In addition, the parameters of an experiment (e.g., λ, Λ, i_{max}, j_{max}) and some algorithms (e.g., Projection method) are fixed in the configuration file. At the beginning of each experiment, SymuMaster is calibrated based on the configuration file. As mentioned in Section 1.3, the network file and demand profiles are the other inputs for the simulation-based DTA. The network file includes all the features of the network, e.g., nodes, links, signalization, available transportation modes. Note that trip-based approach means that the demand profile includes the information of each traveler (user) in the network. The user's information contains, at least, origin, destination, departure time, and the set of transportation modes that the user can choose for his/her travel.

This chapter considers three networks with different topologies to investigate if the network size influences the algorithm settings. Note that all the networks in this chapter are

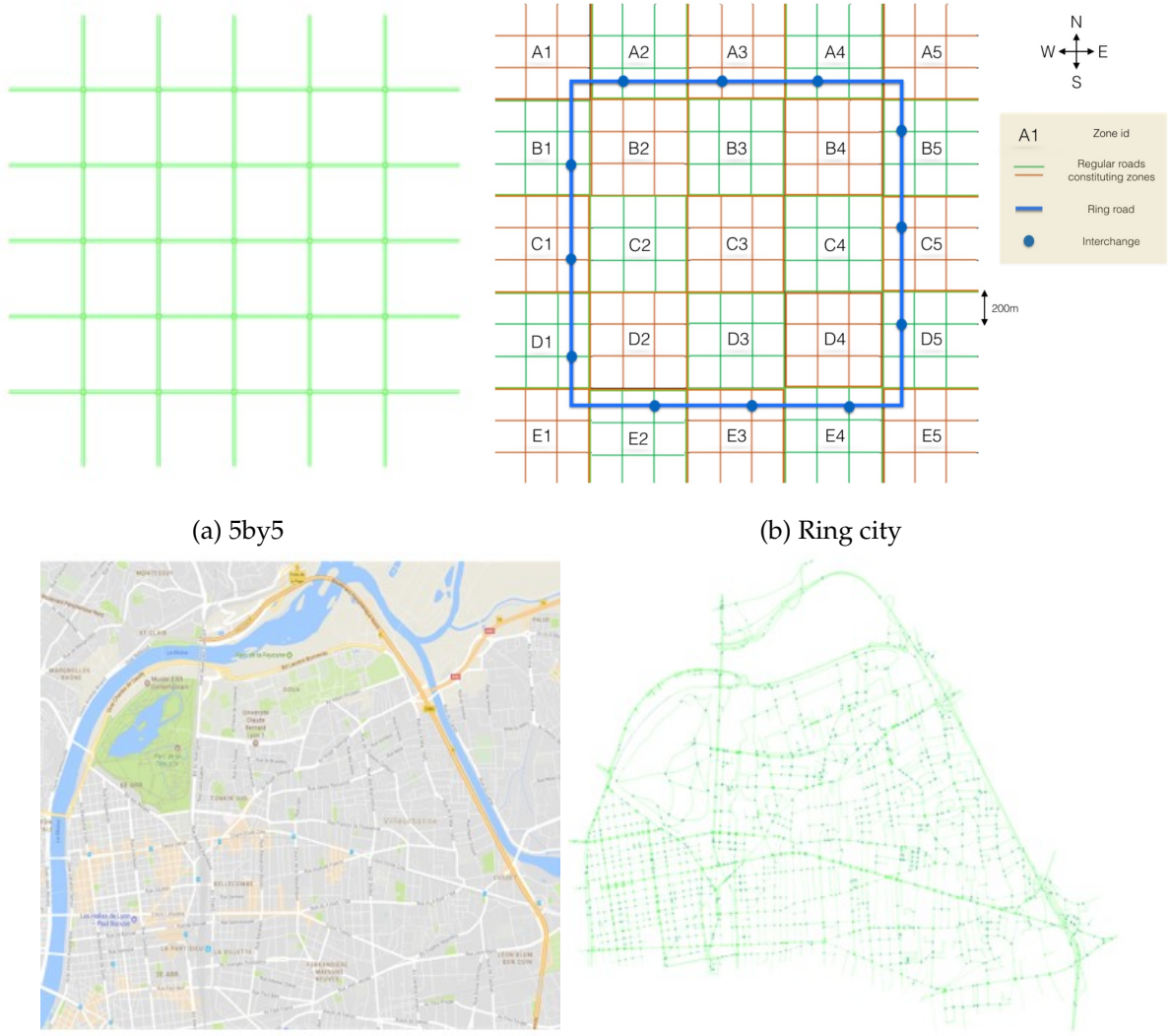
TABLE 2.2 – All the methods in the inner loop considered in this chapter

Method	Abbreviation	Reference
Swapping algorithms (Step 6)		
MSA	MSA	Robbins & Monro (1951)
MSA ranking	MSAR	Sbayti *et al.* (2007)
Projection method	PM	**Adapted** Halpern (1967)
Projection Initialization	PI	**Adapted** Friesz *et al.* (2011)
Initialization MSA	IMSA	**New**
Gap-based	GB	Lu *et al.* (2009) (Flow-based)
Gap-based Normalized	GBN	**New**
Probabilistic	Prob.	**Ameli *et al.* (2017)**
Step size Probabilistic	SSP	Halat *et al.* (2016)
Gap-based Probabilistic	GBP	Lu *et al.* (2009) Verbas *et al.* (2015)
Boost up Gap-based	BGB	**New**
Inner loop initialization (Step 5)		
Default method	All-or-nothing	Lu *et al.* (2009)
Keep the assignment	Keep solution	**New**
Initial step size selection (Step 6)		
Default method	Initial	Lu *et al.* (2009)
Re-initializing the step size	Reset	**New**
Smart step size	Smart	**New**

mono-modal, i.e., with car-traffic only. A 5 × 5 grid network (5by5) is used for the smallest scale network, see Figure 2.2(a). All the intersections are signalized and the green and red light duration is set to 30 seconds. The simulation period is 2 hours for 19 origins and 16 destinations. The medium-scale network exemplifies a Manhattan type city with a ring road (Ring city), see Figure 2.2(b). This network corresponds to 14×14 two-way regular roads with a speed limit of 50km/h. These roads delimit blocks that are grouped 3 by 3 to form 5×5 zones. A two-way ring road with a speed limit of 90km/h has 12 interchanges with peripheral zones. All the intersections are signalized except the interchanges with the ring road and the green and red light duration is set to 30 seconds. Ring city is simulated for 50 minutes with 26 origins and 24 destinations.

The large-scale network of this chapter is the network of two French cities: Lyon 6e + Villeurbanne (Lyon6V). This network has 1,883 Nodes, 3,383 Links, 94 Origins, 227 Des-

tinations. All the signalized intersections in the real field have been implemented in the simulator with their actual signal timing. It is illustrated in Figures 2.2(c) and 2.2(d). The network is loaded with travelers of all ODs with given departure times in order to represent 2.5 hours of the network with the demand of three levels of saturation based on the study of (Krug *et al.*, 2019).



(a) 5by5

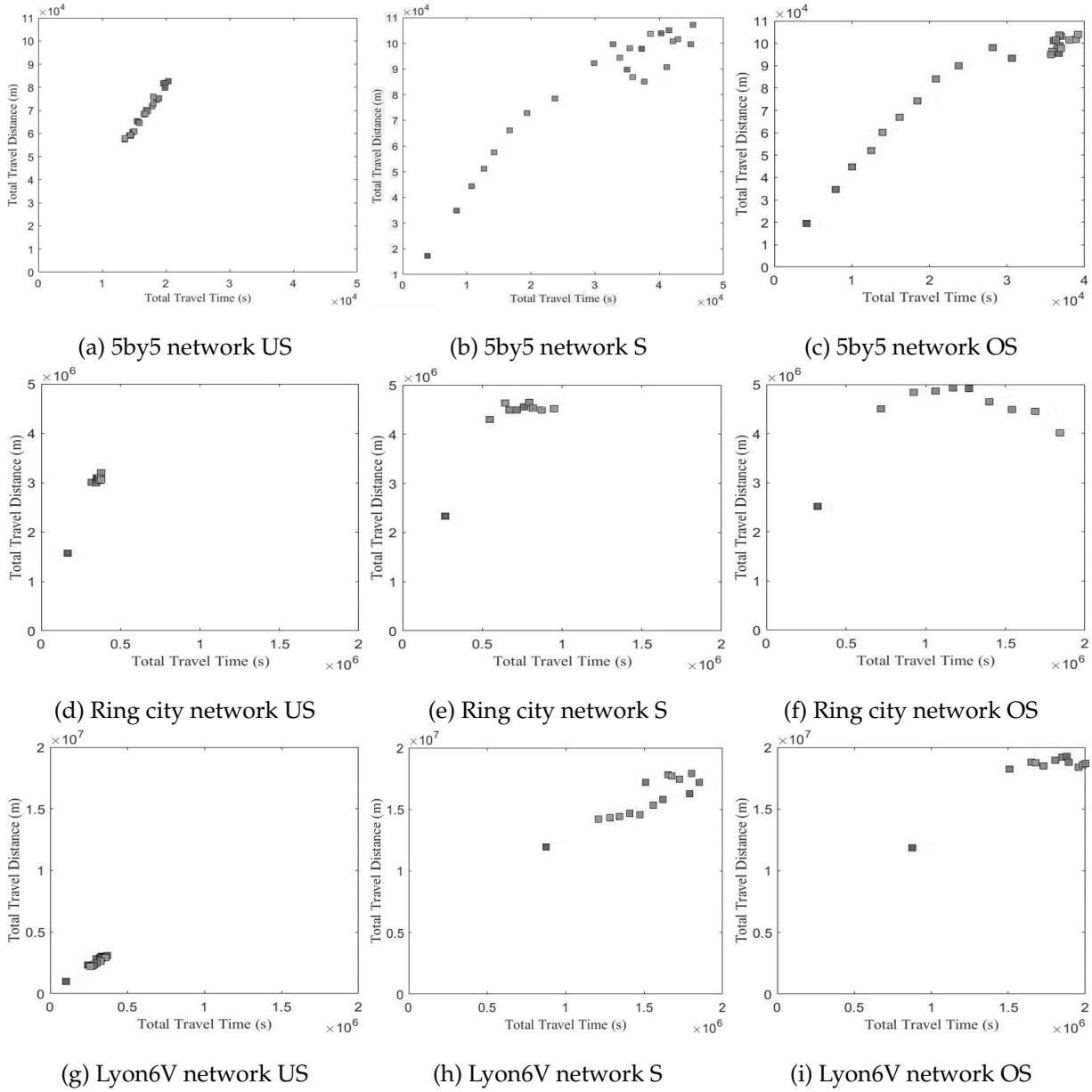
(b) Ring city

(c) ©Google

(d) Lyon6V

FIGURE 2.2 – The three traffic networks of this chapter. (a): Small-scale network. (b): medium-scale network. (c) and (d): Large-scale network.

The Macroscopic Fundamental Diagram (MFD) shows the rapid evolution and gives a synthetic overview of network states. It usually distinguishes three situations. First, the curve increases from (0, 0) and traffic states remain under-saturated when demand is light. This is referred to as the Under Saturation (US) scenario. Travel production, which is equivalent to the total travel distance for a given period of time, stabilizes while the accumulation (or total travel time) still increases, i.e., the network progresses to maximal capacity and then it quickly becomes unloaded. This corresponds to the Saturation (S) scenario when we reach to the maximal capacity of the network and then unload it. Finally, production



(a) 5by5 network US (b) 5by5 network S (c) 5by5 network OS

(d) Ring city network US (e) Ring city network S (f) Ring city network OS

(g) Lyon6V network US (h) Lyon6V network S (i) Lyon6V network OS

FIGURE 2.3 – The macroscopic fundamental diagram of 9 demand scenarios of the three traffic networks. There are 3 different saturation levels per network: Under Saturation (US), Saturation (S) and Over Saturation (OS).

decreases when the total travel time continues increasing. This is the Over Saturation (OS) scenario where the network is heavily congested and remains at the maximum capacity of the network for a long time. We tune the demand to observe all three levels of saturation in numerical experiments to assess the impact of network loading on the performance of the fixed-point algorithm. Table 2.3 presents the total demand for all the test cases of this chapter. Note that demand is constant, but the network state evolves dynamically due to spreading congestion in all three scenarios. The MFDs of each demand scenario for all the networks are presented in Figure 2.3. The points in the figures represent the state of the network at each successive 5-min time period. The MFD of the US scenarios shows the evolution of the state of the networks with almost free flow travel time during the simulation

for each network (Figures 2.3(a), 2.3(d) and 2.3(g)). The saturated MFD diagrams for each network (Figures 2.3(b), 2.3(e) and 2.3(h)) show that the S demand scenarios put the states of the networks in saturation level during the simulation. Finally, the OS scenario moves the network states to a heavily congested situation in which the saturation level is very high (Figures 2.3(c), 2.3(f) and 2.3(i)).

TABLE 2.3 – Total demand for all test cases

Level of saturation	5by5	Ring city	Lyon6V
Under Saturation (US)	3,520	12,000	20,000
Saturation (S)	5,197	19,000	54,190
Over Saturation (OS)	8,100	22,500	100,000

2.7 Results

All the experiments are first initiated with the All-or-nothing assignment algorithm (see Step 1 in Figure 2.1). In this study, we impose a limit on the maximum number of iterations and compare the final solutions obtained by the different algorithms. This is to contain the computational cost and to bring the experiments closer to usual practice in which traffic engineers try to control maximum computation times. Each scenario is executed for five outer loops ($j_{max} = 5$). This means that users have to finally choose between a minimum of six paths (the first path comes from Step 1) for each OD pair. Note that the maximum number of the outer loop iterations is not the limitation. At one point, increasing the number of discovered paths will simply add close alternatives that do not improve the solution. The inner loops run for a maximum of ten iterations ($i_{max} = 10$) for a small network, twenty for Ring city ($i_{max} = 20$), and thirty for a large-scale network ($i_{max} = 30$). We restrict the number of inner loop iterations for each network to obtain a complete comparison of the performances of the algorithms in the same settings. The different values of i_{max} are determined as a function of the size of the network, based on the studies of Halat *et al.* (2016), Verbas *et al.* (2016a). All the experiments are conducted on a 64-bit personal computer with a 12-core central processing unit of 2.10 GHz speed, and a memory of 128 GB.

The experiments are conducted in three stages. First, we compare all the swapping algorithms. Then the best swapping algorithms are chosen for the second stage. The initialization methods are tested in the second stage. Finally, the best combinations of swapping algorithms and initialization methods are considered for the third stage. The step size methods are examined in the third stage.

2.7.1 Comparison of swapping algorithms

The first stage of the numerical experiments entails finding the best swapping algorithm. We track $AGap$ and Violation indicators to assess the performance of each swapping algorithm. The final values are presented in Table 3 for the nine scenarios. Metal colors highlight the top three algorithms (Gold = first, Silver = second, Bronze = third) for each scenario. Moreover, Figure 2.4 presents a bar chart of the computation times. Note that the computation time of each algorithm includes the complete process, e.g., the simulation and shortest path calculations.

TABLE 2.4 – Results of numerical experiments for eleven swapping algorithms [*AGap* (second)]

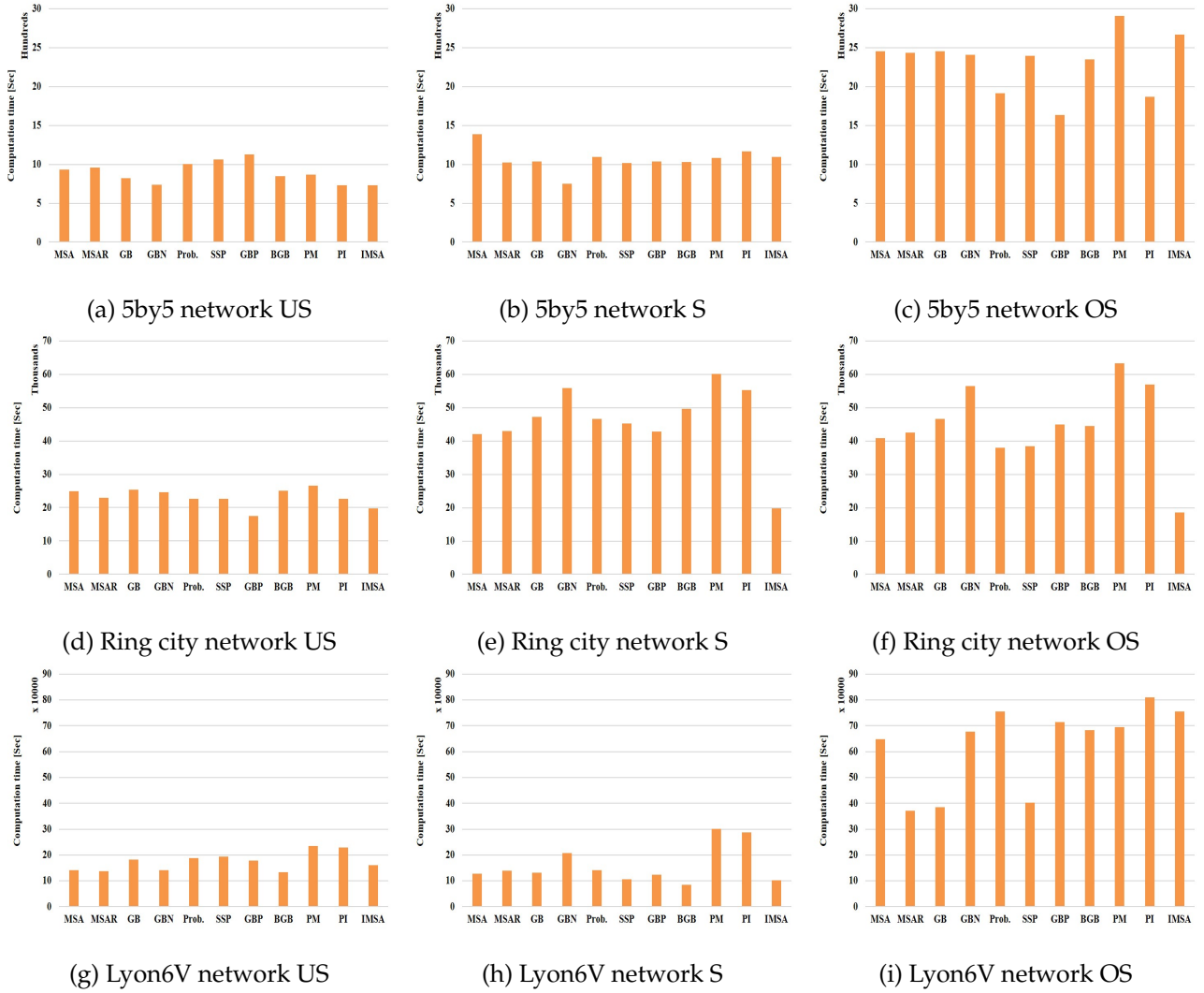
Network	/	algorithm	MSA	MSAR	GB	GBN	Prob.	SSP	GBP	BGB	PM	PI	IMSA
5by5	US	AGap	0.91	0.17	1.10	0.91	0.17	0.88	0.73	0.91	0.48	0.17	1.47
		Violation	0.05	0.05	0.08	0.05	0.05	0.08	0.07	0.06	0.06	0.05	0.05
	S	AGap	11.22	13.34	82.65	11.22	3.50	39.18	22.36	64.05	37.79	112.09	80.19
		Violation	0.13	0.13	0.16	0.13	0.13	0.18	0.16	0.18	0.18	0.16	0.17
	OS	AGap	0.70	0.85	1.18	0.70	0.93	1.44	2.37	0.71	1.05	1.56	1.19
		Violation	0.04	0.09	0.08	0.04	0.05	0.07	0.09	0.08	0.08	0.08	0.08
Ring city	US	AGap	5.79	2.99	8.66	4.30	2.74	5.47	3.02	5.13	4.41	3.27	38.13
		Violation	0.13	0.06	0.13	0.11	0.06	0.12	0.08	0.10	0.10	0.10	0.24
	S	AGap	30.96	12.81	18.31	16.35	8.33	40.13	12.33	24.72	200.71	206.70	205.24
		Violation	0.17	0.16	0.21	0.15	0.16	0.19	0.16	0.22	0.36	0.38	0.36
	OS	AGap	50.06	20.81	47.73	49.35	16.16	36.16	48.38	26.00	230.71	189.04	225.24
		Violation	0.26	0.16	0.17	0.26	0.14	0.21	0.20	0.18	0.36	0.32	0.33
Lyon 6V	US	AGap	109.47	18.78	37.01	41.09	12.30	16.15	13.59	35.47	20.01	41.09	39.75
		Violation	0.32	0.15	0.20	0.24	0.12	0.14	0.13	0.21	0.19	0.24	0.24
	S	AGap	79.89	39.54	237.35	29.14	24.72	47.73	121.45	62.98	1157.21	27.83	798.47
		Violation	0.25	0.24	0.35	0.16	0.13	0.25	0.26	0.25	0.35	0.15	0.43
	OS	AGap	180.38	233.44	183.83	80.79	106.83	339.87	153.00	240.95	108.45	73.43	47.18
		Violation	0.22	0.28	0.23	0.17	0.19	0.30	0.34	0.35	0.27	0.15	0.14

Regarding the uniqueness of the UE, this is the case for the DNL problem when users have the same characteristic (homogeneous demand) and travel time functions are increasing (Iryo & Smith, 2017, Ameli *et al.*, 2018). Monotonicity exists at the link level for the traffic simulator of this study (Leclercq, 2007b) but not at the node because most of the intersections are signalized. Therefore, we cannot claim, on the basis of the literature, that we have unicity; however, we check that we have a similar solution in terms of path flow distribution whatever the algorithm used in each case. In particular, in the small-scale network (5by5), MSA ranking, Probabilistic, and Projection initialization algorithms lead to optimal values for the US scenario (Table 2.4). Furthermore, the path flow distributions are equal, possibly providing the optimal UE for this scenario.

According to Figure 2.4(a), the Projection initialization algorithm converges fastest to the best solution. The Initialization MSA algorithm also converges quickly, but it does not converge to the best solution. In the saturation scenario, the best solution is obtained by Probabilistic algorithm, but Gap-based normalization algorithm is the fastest algorithm to converge (Figure 2.4(b)) while the $AGap$ by this algorithm is ranked second among all the algorithms. The MSA algorithm provides the same gap as the Gap-based normalization algorithm and a smaller gap than the MSA ranking; however, the Computation Time (CT) of this algorithm is the longest. In the saturation scenario, the results provided by the Gap-based, Boost-up Gap-based, Projection initialization, and Initialization MSA algorithms are poor in comparison with the best solution. In the over-saturation scenario, the Gap-based normalization and Boost-up Gap-based algorithms perform well in terms of solution quality and CT. Note that MSA and MSA ranking also provides a high-quality solution, but the CTs are higher than for Gap-based normalization and Boost-up Gap-based algorithms (Figure 2.4(c)).

Table 2.4 shows that the Probabilistic algorithm provides the best solutions in all the scenarios of the medium-scale network (Ring city). For the US scenario, the Gap-based Probabilistic algorithm converges fast (Figure 4(d)) and also to a good solution, with a $AGap$ only 0.28 second longer than the best solution. The MSA ranking and Initialization MSA algorithms provide solutions close to the best algorithm with good CT in comparison to other algorithms. In the S scenario, the Probabilistic algorithm is significantly better than the others in terms of solution quality. Initialization MSA algorithm converges fast (Figure 4(e)) but to a bad solution, which is more than 25 times bigger than the best solution. The MSA ranking algorithm is the third best algorithm; also the CT is better than the best algorithm. As with the saturation scenario of Ring city, the Initialization MSA algorithms converge rapidly in the OS scenario (Figure 4(f)), but the solutions are poor in comparison with the top three algorithms: Probabilistic, MSA ranking and Boost-up Gap-based.

In the large-scale network (Lyon6V), The results show that many algorithms with low CTs converge to solutions far from the best solution. For instance, in the OS scenario, the MSA ranking, Gap-based and Step size Probabilistic algorithms converge faster than the others (Figure 2.4(i)), but the $AGap$ and Violation are very poor in comparison with the best solution. The same observation can be made for the MSA, Gap-based, Gap-based Probabilistic, and Boost-up Gap-based algorithms in the saturation scenario and for the Gap-based Normalized in the US scenario. The top three algorithms in the OS scenario (Projection initialization, Initialization MSA, and Gap-based Normalized) perform significantly better than the others. In the OS scenario, no algorithms except the Initialization MSA can find solutions with a $AGap$ less than a minute. The CTs are also long for two top algorithms.



(a) 5by5 network US (b) 5by5 network S (c) 5by5 network OS

(d) Ring city network US (e) Ring city network S (f) Ring city network OS

(g) Lyon6V network US (h) Lyon6V network S (i) Lyon6V network OS

FIGURE 2.4 – The computation time bar chart of 9 demand scenarios of the three traffic networks. There are 3 different saturation levels per network: Under Saturation (US), Saturation (S) and Over Saturation (OS)

According to the results, the saturation level has an impact on the computational performance of the swapping algorithms because the solution space of the fixed-point problem is more complex and the intermediate results of the inner loop are not stable. Moreover, the results in Table 2.4 show that the performance of the swapping algorithms depends not only on the saturation level but also on the scale of the traffic network. For instance, the MSA method works well in small-scale, but by increasing the size of the traffic network, it obtains poor results for $AGap$ and Violation in comparison with the other algorithms. It is noteworthy that the MSA algorithm is one of the worst algorithms on the large-scale, particularly for the US scenario. The reasons for this performance of the drawbacks of MSA algorithms, but its CT is comparable with other algorithms in the US and S scenario. The MSA ranking algorithm is not among the top three algorithms in the large-scale. However, because of the ranking process, it provides good enough solution compared to the best algorithm except in the OS scenario, where the MSA ranking algorithm converges faster than the others but to a bad solution.

The Gap-based algorithm does not appear in the top three algorithms for all the scenarios, but it obtains solutions with small $AGap$ for all the US demand scenarios. In these test cases, when increasing the level of network saturation, the Gap-based algorithm cannot converge to a high-quality solution. The Gap-based Normalized algorithm performs better than the Gap-based algorithm in most of cases. The normalization techniques improve the performance of the Gap-based method, especially in the saturation and OS scenarios. Generally, the Probabilistic algorithm is the best algorithm in the US and S scenarios of all the networks and also it also provides good performance for the OS scenario in the small and the medium test cases. It should be recalled that the Probabilistic algorithm is the only step size free algorithm of this study. It can explore the solution space without limitation, which makes it more robust than other algorithms. However, in the large-scale and OS scenario, the CT of Probabilistic algorithm is very high, and it could not provide a good solution compared to the best algorithm. It means that this algorithm does not fully cover the solution space under the determined computation budget.

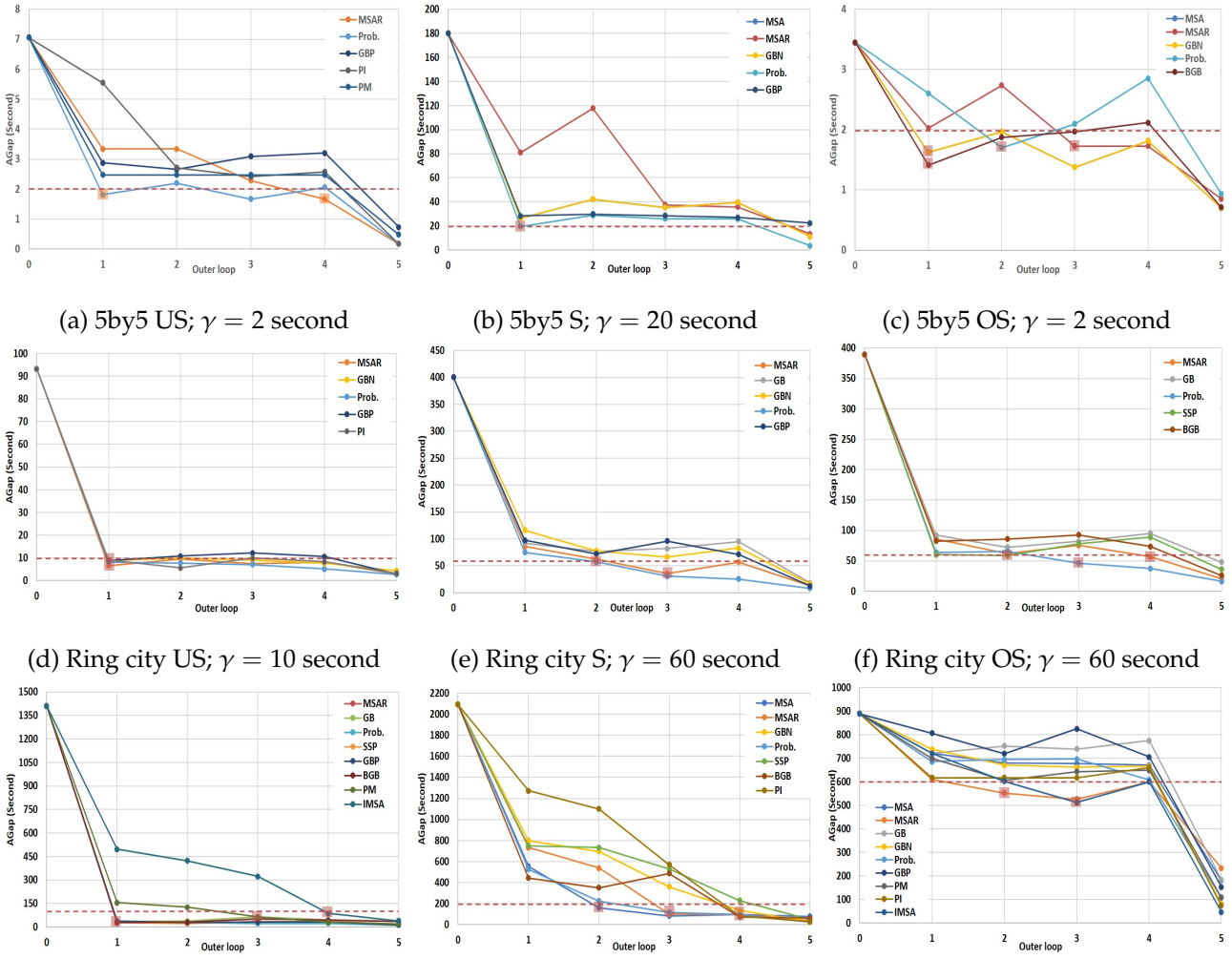
The hybrid algorithms (Section 2.5.2.5) work better than MSA and Gap-based algorithms in the large-scale and US scenario, but they are dominated by Probabilistic algorithm because the step size still limits them. The Step size Probabilistic algorithm, which is a combination of MSA and Gap-based algorithms, provides better results than both methods in the medium- and the large-scale networks with US and saturation levels of demand. When the saturation level of the network increase, the Gap-based Probabilistic algorithm cannot provide a good solution. The Step size Probabilistic and Gap-based Probabilistic algorithms are dominated by the Probabilistic algorithm in all test cases. The Boost-up Gap-based algorithm, which used the boost-up step size performs better in OS scenarios where there is a large number of trips to optimize.

The projection-based algorithms (Section 2.5.2.2) are good for small scale. Their CT is increased significantly by the saturation level of the network. In addition, they have a high CT in the large-scale network compared to other algorithms. The Projection initialization algorithm does not work well in saturation and OS scenarios in all networks. On the other hand, the Initialization MSA algorithm provides good results in Lyon6V network, particularly in OS scenario. The Projection method works better in the small network and US scenarios, but it is not a fast algorithm for trip-based DTA.

Figure 2.5 presents the convergence pattern of swapping algorithms. Top five algorithms in terms of the $AGap$ indicator for each saturation level of small- and medium-scale networks are presented in Figures 2.5(a)- 2.5(f). For the large-scale network, we present more swapping algorithm convergence pattern (Figures 2.5(g)- 2.5(i)) to analyze more algorithms.

According to Figure 2.5, the saturation level has an impact on the scale of $AGap$ at each level of the outer loop during the optimization process. In the US scenario, the value decreases suddenly for the first outer loop but for more saturated scenarios adding the new shortest path changes $AGap$ scale of users, particularly in the large-scale network. As shown in Figure 2.5(i), the gap value slumps for the outer loop five, where users have six alternatives per OD to choose.

In addition to the quality of the final solution, converging with the less number of outer loops $(j < 5$ in this study$)$ to a solution with good quality is important in practice. In other words, the algorithm with a minimum number of outer loops and good solution in terms of $AGap$ can be more efficient. Note that the solution with good quality means the $AGap$ of the solution is below a predetermined satisfaction threshold. Here, we consider a given



(a) 5by5 US; $\gamma = 2$ second (b) 5by5 S; $\gamma = 20$ second (c) 5by5 OS; $\gamma = 2$ second

(d) Ring city US; $\gamma = 10$ second (e) Ring city S; $\gamma = 60$ second (f) Ring city OS; $\gamma = 60$ second

(g) Lyon6V US; $\gamma = 100$ second (h) Lyon6V S; $\gamma = 200$ second (i) Lyon6V OS; $\gamma = 600$ second

FIGURE 2.5 – Convergence patterns for the swapping algorithms of all scenarios. There are 3 different saturation levels per each network: Under Saturation (US), Saturation (S) and Over Saturation (OS). γ denotes the satisfaction threshold.

satisfaction threshold (γ) for the quality of the solution at outer loop iteration j ($AGap_j$) in order to investigate which algorithms can reach faster this threshold ($AGap_j \leq \gamma$). The values of γ for US and S scenarios are approximately equal to $\lceil 2 \times AGap^* \rceil + 1$ where $AGap^*$ denotes the best quality of the final solution obtained by swapping algorithm in second. For the OS scenarios in large-scale, we set the γ to 9 minutes based on the initial $AGap$ ($\approx$ 15 min) which is obtained by the All-or-nothing algorithm. Figure 2.5 presents the γ values for each scenario. We can see in this figure, the first time that each algorithm is positioned below γ threshold before the last iteration and then determine the minimal number of the outer loop to guaranty a given level of performance. The Probabilistic algorithm can converge faster in the small and medium networks according to the γ value. For instance, about the Ring city network saturated scenario, if we want to reach the level of $AGap_j = 1$ minute ($\gamma = 60$), we need to do only one outer loop iteration with the Probabilistic algorithm. In the large-scale considering the γ threshold helps to save CT. In the large-scale network, the MSA algorithm for the saturated scenario, and the MSA ranking algorithm for OS scenario

are much faster than others and need only two outer loops to reach the threshold.

All our analyses give good hints about which are the best swapping algorithms. As we discussed in Section 2.5, there are different components in the optimization process, which can help the swapping algorithm to find a better solution. Consequently, the swapping algorithms with good quality indicators ($AGap$ and Violation) for the final solution and CT in each scenario (all the golden cells in Table 2.4) are chosen for the next stage of the numerical experiment. We also consider all algorithms which obtain a solution with a similar value for $AGap$ and Violation to the best algorithms solution for the next stage. For more investigation in the large-scale, we choose at the swapping algorithms including all colored cells in Table 2.4 and algorithms with $AGap_j \leq \gamma;\ \forall j < j_{max}$ in Figure 2.5 for each saturation level. In other words, we also consider the swapping algorithms which converge to the γ before the last outer loop. Table 2.5 presents which algorithms are selected in each scenario for further investigation. MSA ranking and Probabilistic algorithms are suitable for all scenarios. MSA algorithm is efficient for small network, and Projection method and Step size Probabilistic algorithm are efficient for the large-scale network. Gap-based Normalized and Boost-up Gap-based algorithms provide good results in more saturated scenarios, while Gap-based Probabilistic algorithm is good for US scenarios and larger networks.

2.7.2 Initialization methods

The second stage of the numerical experiments corresponds to Step 5 in the solution algorithm (Figure 2.1). The methodology improvements to this step are discussed in Section 2.5.3 Now, an alternative initialization method (Keep solution; Table 2.2) for the inner loop is applied for all numerical experiments. The results for combinations of swapping algorithms and initialization methods are presented in Tables 2.6 and 2.7. The goal is to compare the Keep solution method to the initial method (All-or-nothing). The first three combinations of swapping algorithms and initialization methods are highlighted by medal colors (Gold = first, Silver = second, Bronze = third). Table 2.6 presents the quality indicators of final solutions for initialization methods in all networks. About 5by5 network, in the US scenario, it is shown that Keep solution initialization obtains the optimal solution for all best

TABLE 2.5 – Best swapping algorithms

Network	5by5			Ring city			Lyon 6V		
Scenario	US	S	OS	US	S	OS	US	S	OS
MSA		✓	✓						
MSAR	✓	✓	✓	✓	✓	✓	✓	✓	✓
GB									
GBN		✓	✓					✓	✓
Prob.	✓	✓		✓	✓	✓	✓	✓	✓
SSP							✓	✓	
GBP				✓	✓		✓		
BGB			✓			✓			
PM							✓	✓	✓
PI	✓								✓
IMSA									✓

three algorithms. Also, in S and OS scenarios, this method improves the performance of all swapping algorithms in comparison with the same swapping algorithm and All-or-nothing initialization. The results on the small network show that the Probabilistic and Keep solution combination provides the best solution in all saturation levels.

TABLE 2.6 – Results of initialization methods [*AGap* (second)]

Network	Scenario		US		S		OS	
	Algo. / Method		AGap	Violation	AGap	Violation	AGap	Violation
5by5	MSA	AoN	-		11.22	0.13	0.70	0.04
		Keep			5.60	0.21	0.68	0.03
	MSAR	AoN	0.17	0.05	13.34	0.13	0.85	0.09
		Keep	0.17	0.05	7.78	0.15	1.10	0.05
	GBN	AoN	-		11.22	0.13	0.70	0.04
		Keep			6.48	0.21	0.46	0.04
	Prob.	AoN	0.17	0.05	3.50	0.13	-	
		Keep	0.17	0.05	0.86	0.10		
	BGB	AoN	-		-		0.71	0.08
		Keep					0.50	0.01
	PI	AoN	0.17	0.05	-		-	
		Keep	0.17	0.05				
Ring city	MSAR	AoN	2.99	0.06	12.81	0.16	20.81	0.16
		Keep	1.89	0.08	12.34	0.17	15.25	0.12
	Prob.	AoN	2.74	0.06	8.33	0.16	16.16	0.14
		Keep	1.47	0.05	5.07	0.09	6.35	0.10
	GBP	AoN	3.02	0.08	12.33	0.16	-	
		Keep	1.53	0.05	6.15	0.13		
	BGB	AoN	-		-		26.00	0.22
		Keep					10.48	0.14
Lyon 6V	MSAR	AoN	18.78	0.15	39.54	0.24	233.44	0.28
		Keep	13.51	0.13	16.42	0.08	162.60	0.28
	GBN	AoN	-		29.14	0.16	80.79	0.17
		Keep			22.47	0.11	69.77	0.15
	Prob.	AoN	12.30	0.12	24.72	0.13	106.83	0.19
		Keep	5.96	0.09	10.59	0.06	92.83	0.19
	SSP	AoN	16.15	0.14	47.73	0.25	-	
		Keep	6.03	0.09	13.58	0.08		
	GBP	AoN	13.59	0.13	-		-	
		Keep	6.79	0.09				
	PM	AoN	20.01	0.19	27.83	0.15	108.45	0.27
		Keep	7.33	0.09	21.58	0.11	101.49	0.19
	PI	AoN	-		-		73.43	0.17
		Keep					62.60	0.15
	IMSA	AoN	-		-		47.18	0.14
		Keep					40.89	0.15

In Ring city network, same as 5by5, the Keep solution method improves the performance

TABLE 2.7 – Computation time of initialization methods (second)

Network		5by5			Ring city			Lyon 6V		
Scenario		US	S	OS	US	S	OS	US	S	OS
MSA	AoN	-	1384	2451	-	-	-	-	-	-
	Keep		713	1184						
MSAR	AoN	956	1021	2430	22962	42973	42447	137527	139232	371695
	Keep	730	624	2306	18130	39262	39418	154138	132750	333751
GBN	AoN	-	754	2406	-	-	-	-	206912	677682
	Keep		739	1965					197719	556489
Prob.	AoN	1003	1093	-	22586	46536	38015	187394	141885	754640
	Keep	595	664		15431	28517	36462	138046	111598	291486
SSP	AoN	-	-	-	-	-	-	194636	111824	-
	Keep							135540	106705	
GBP	AoN	-	-	-	17541	42839	-	177725	-	-
	Keep				23140	36431		121653		
BGB	AoN	-	-	2349	-	-	44461	-	-	-
	Keep			1899			27358			
PM	AoN	-	-	-	-	-	-	234809	287847	693828
	Keep							155722	179288	653254
PI	AoN	731	-	-	-	-	-	-	-	809785
	Keep	492								741501
IMSA	AoN	-	-	-	-	-	-	-	-	754350
	Keep									671548

of swapping algorithms in particular for the Probabilistic method. The differences between All-or-nothing and Keep solution violation value of two closed $AGap$ solution in Table 2.6, box of MSA ranking and S scenario, shows the different direction of searching by swapping algorithm and the impact of the starting point at the beginning of each outer loop on the final result of optimization methods.

In the large-scale test case, the impact of the initialization is significant. It means that the quality of the solution by keep solution method is always better than All-or-nothing with a considerable difference of $AGap$, e.g., the Keep solution method obtains a better solution (more than 50% lower $AGap$) for Step size Probabilistic and Probabilistic algorithms in US and saturation levels.

Table 2.7 presents the CT of all experiments in this stage. First, the results prove that the initialization method has an impact on CT. Second, the Keep solution method which is indicated by "Keep" in the Table 2.7 improves the speed of convergence for all swapping algorithms in comparison with the All-or-nothing method which is the default method in the literature. Third, the combination of Probabilistic and Keep solution method is faster than other methods in most of the cases as we can observe that this method is always in the top two fastest methods in all scenarios (gold and silver cells in Table 2.7).

In order to have a better comparison between the performance of different initialization methods, the inner loop convergence patterns of MSA ranking and Probabilistic algorithms with both initialization methods are presented in Figure 2.6. The results correspond to the saturation scenario for the large-scale network. The default setting, which is the "All-or-

nothing" method, starts at each outer loop with the same assignment pattern. It works like as handbrake at the beginning of each outer loop and forces the algorithm to start from the assignment with a high $AGap$. Therefore, the swapping algorithm should attempt to come back to the optimal region by several iterations.

It is obvious that "All-or-nothing" method needs more iteration (CT) compared to the Keep solution method to find the new optimal solution for each outer loop. The Keep solution method converges faster than the other one and also provides a better solution (Table 2.6). For instance, in Figure 2.6, The Keep solution method prevents the MSA ranking method from spending several iterations in order to come back to the lower range of $AGap$ and also helps this swapping algorithm to converge faster than the initial version. Note that the first outer loop is the same for both methods and then from the second one "MSA ranking + Keep solution" continue with previous path flow distribution, but the initial version of the MSA ranking starts the second outer loop with the All-or-nothing assignment.

Moreover, Figure 2.6 shows the flexibility of probabilistic algorithm to search the solution space. The probabilistic algorithm moves in a larger range of $AGap$ during each outer loop, and it does not improve the solution sequentially except when the probability of swapping (Equation 2.24) is very low for all users, i.e., the algorithm is close to the optimal solution.

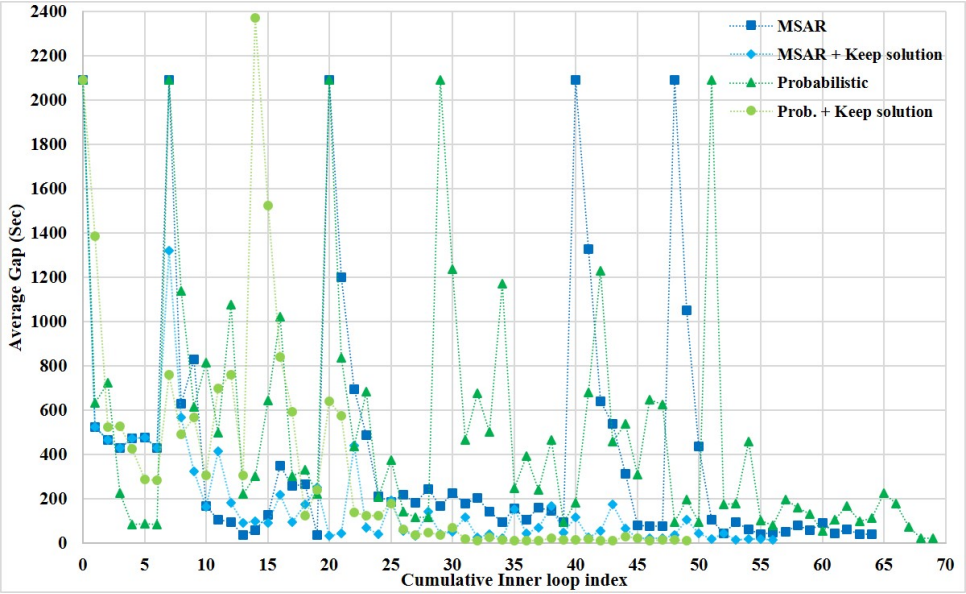


FIGURE 2.6 – Convergence patterns of the inner loops with initialization methods for Saturation (S) scenario on Lyon6V network. [default is All-or-nothing initialization]

2.7.3 Step size methods

In the previous stage, we presented and discussed the results of the initialization method, i.e., Step 5 in Figure 2.1. Here, we will present and discuss the results for the step size methods (Step 6). For the final stage, we select the best methods (colored in Tables 2.6 and 2.7) except the Probabilistic algorithm. The Probabilistic method is excluded simply because it does not have a step size (Section 2.5.2.4). For methods with step size, the one or two best combinations in all test cases are chosen. The new methods for step size initialization are compared with the initial method (Section 2.5.4). Note that we fix the seed for probability

functions in the Step size Probabilistic and Gap-based Probabilistic in order to compare the impact of step sizes. The results of applying the three methods on the three networks are presented in Tables 2.8 and 2.9. Medal colors highlight the top three combinations for each scenario.

TABLE 2.8 – Results of initial step size methods [*AGap* (second)]

Network	Scenario		US		S		OS	
	Algo. / Method		AGap	Violation	AGap	Violation	AGap	Violation
5by5	MSA	Initial			5.60	0.21		
		Reset	-		2.36	0.19	-	
		Smart			3.77	0.18		
	MSAR	Initial	0.17	0.05				
		Reset	0.17	0.05	-		-	
		Smart	0.17	0.05				
	GBN	Initial					0.46	0.04
		Reset	-		-		0.62	0.03
		Smart					0.45	0.03
	BGB	Initial					0.50	0.08
		Reset	-		-		0.45	0.02
		Smart					0.69	0.04
	PI	Initial	0.17	0.05				
		Reset	1.83	0.07	-		-	
		Smart	0.17	0.05				
Ring city	MSAR	Initial	1.89	0.08	12.34	0.17	15.25	0.12
		Reset	1.47	0.05	9.57	0.15	9.57	0.10
		Smart	1.71	0.08	11.25	0.18	12.38	0.12
	GBP	Initial	1.53	0.05	6.15	0.13		
		Reset	1.53	0.05	5.73	0.09	-	
		Smart	1.47	0.05	5.49	0.09		
	BGB	Initial					10.48	0.14
		Reset	-		-		21.38	0.19
		Smart					11.31	0.11
Lyon 6V	SSP	Initial	6.03	0.09	13.58	0.08		
		Reset	5.96	0.09	11.44	0.08	-	
		Smart	12.13	0.12	17.75	0.09		
	GBP	Initial	6.79	0.09				
		Reset	6.50	0.10	-		-	
		Smart	6.42	0.09				
	IMSA	Initial					40.89	0.15
		Reset	-		-		35.49	0.14
		Smart					34.10	0.14

Table 2.8 presents the quality indicators for the final solution of all experiments for all networks. Smart step size improves the performance of the MSA and Gap-based Normalized algorithms. The Reset method improves the MSA for the small network. For the medium-scale network (Ring city), the results show that the Reset method improves the MSA rank-

TABLE 2.9 – Computation time of initial step size methods (second)

Network		5by5			Ring city			Lyon 6V		
Scenario		US	S	OS	US	S	OS	US	S	OS
MSA	Initial	-	713	-	-	-	-	-	-	-
	Reset		337							
	Smart		705							
MSAR	Initial	730	-	-	18130	39262	39418	-	-	-
	Reset	2430			9086	34439	35562			
	Smart	3381			11033	24550	19506			
GBN	Initial	-	-	1965	-	-	-	-	-	-
	Reset			2337						
	Smart			2219						
SSP	Initial	-	-	-	-	-	-	135540	106705	-
	Reset							81498	101141	
	Smart							103695	89258	
GBP	Initial	-	-	-	23140	36431	-	121653	-	-
	Reset				3878	37892		107142		
	Smart				13927	25805		92989		
BGB	Initial	-	-	1899	-	-	27358	-	-	-
	Reset			2796			26114			
	Smart			2747			27557			
PI	Initial	492	-	-	-	-	-	-	-	-
	Reset	1908								
	Smart	3270								
IMSA	Initial	-	-	-	-	-	-	-	-	671548
	Reset									571493
	Smart									485980

ing algorithm, and the Smart method works well with Gap-based Probabilistic algorithm (Table 2.8). In the Boost-up Gap-based algorithm, the step size of this method is already modified at each inner loop by boost-up techniques. The result shows that applying the alternative step size methods cannot improve the performance of this algorithm.

The results in Table 2.8 for the large-scale show that the Smart method improves the solution of Gap-based Probabilistic and Initialization MSA algorithms compared to the Initial method which is the default setting for step size initialization and Reset method. Same as Ring city network, the Reset method improves the performance of Step size Probabilistic algorithm. The CT for all experiments of this stage is presented in Table 2.9. The Smart and Reset methods converge slowly in the small network, but they converge faster than the Initial method in the large-scale network particularly the Smart method for Initialization MSA and oversaturated scenario compared to other methods with which we can save a minimum of one day of computation.

2.8 Discussion

This chapter focused on improving solution algorithms in the literature for finding user equilibrium, by considering trip-based dynamic network loading. We compared the performance of the solution algorithms (see Table 2.2 to create a synthesis of the main algorithms and methods in this chapter), based on the quality of solutions and computation times. Table 2.10 presents the best configuration of the solution algorithm that makes the best compromise between quality and computation time for all network size and saturation levels. If the focus is only on the quality of the solution, more than one configuration can be used in most of the cases. However, the computation time is very important, particularly for the large-scale network. For instance, in the large-scale and oversaturated scenario, Probabilistic algorithm cannot provide a better solution than Initialization MSA algorithm, but it is more than two days faster than Initialization MSA algorithm in computation time (Table 2.7). The combination of Probabilistic approach (without step size) and Keep solution initialization appears in most of the cases in Table 2.10 as the best algorithm. It is not necessarily always the best in terms of quality and speed, but it is the one that is most likely to obtain the best solution in all scenarios and can be considered as the most robust alternative.

This chapter first showeds that the network size and saturation level has an impact on the performance of solution algorithms to solve the DTA problem. The analysis shows that the classic algorithms (e.g., MSA) exhibit good performance in the small-scale network, but they do not provide a good solution in the large-scale network. The computational cost of the MSA algorithm is prohibitive for the large-scale network. On the contrary, one of the hybrid algorithms, Step size Probabilistic (see Section 2.5.2.5), worked faster in the large-scale network (Table 2.10). The MSA ranking is efficient for small- and medium- scale networks, but it cannot provide good results for the large-scale network. Moreover, the results show that some algorithm such as Gap-based algorithm, and Projection method are dominated by other algorithms for all scenarios.

Second, this study shows that the initial assignment and step size at the beginning of the outer loop have a significant impact on the final solution and convergence speed of the algorithm. An alternative method is proposed to initialize the assignment pattern at the beginning of the outer loop (Section 2.5.3). According to the results, the Keep solution method improves the performance of all swapping algorithms compared to the recent methodology in the literature. Table 2.10 shows that all the best configuration of the solution algorithm includes Keep solution method. This initialization method also speeds up the swapping algorithms convergence process (Table 2.7).

Third, this study proposes two new methods for the initialization of the step size (Section 2.5.4). The step size ensures that the algorithm converges and it has a direct impact on the speed of convergence. However, the step size cannot guarantee the quality of the final solution. The two new methods for step size provide a better solution with a combination of different swap formulas than the classic method in the literature. The algorithms based on MSA are improved by the Reset step size method. Besides, the Smart method improves the algorithm based on the gap function and projection method. The new step size methods speed up the convergence of all algorithms, especially in the large-scale (Table 2.9). For the next chapter, we are seeking meta-heuristic algorithms for the DTA problem in order to speed up the optimization process with parallel computation. In addition, designing the framework to predetermine the computation budget based on network size, topology, and

saturation level is a topic of interest.

TABLE 2.10 – Best algorithms and settings with respect to the network size and loading

Network	Saturation level	Best algorithm	Initialization		Step size methods			Ranking for		Best compromise
			AoN	Keep	Initial	Reset	Smart	Quality	Speed	
5by5	US	MSAR		✓	✓	✓	✓	1	3	
		Prob.		✓				1	2	
		PI		✓	✓			1	1	✓
	S	MSA		✓		✓		2	1	✓
		Prob.		✓				1	3	
	OS	MSA		✓				4	1	✓
		GBN		✓			✓	1	4	
		BGB		✓		✓		1	6	
Ring city	US	MSAR		✓		✓		1	2	
		Prob.		✓				1	3	
		GBP		✓		✓		1	1	✓
	S	Prob.		✓				1	1	✓
	OS	MSAR		✓		✓		2	1	✓
		Prob.		✓				1	5	
Lyon 6V	US	Prob.		✓				1	2	
		SSP		✓		✓		1	1	✓
	S	Prob.		✓				1	4	
		SSP		✓			✓	2	1	✓
	OS	Prob.		✓				9	1	✓
		IMSA		✓			✓	1	4	

3. New framework for DTA calculation

Based on the investigation in the previous chapter, we notice that most existing algorithms are based on path-swapping descent direction methods. From the computational standpoint, the main drawback of these methods is that they cannot be parallelized. This is because the existing algorithms need to know the results of the last iteration to determine the next best path flow for the next iteration. Thus, their performance depends on the choice of a single initial solution and on the intermediate solutions, which means they exploit the way that the solutions satisfy the equilibrium conditions. But they do not explore sufficiently the solution space for the optimal solution. More specifically, the goal of this chapter is to overcome the drawbacks of serial algorithms by using meta-heuristic algorithms known to be parallelizable and that have never been applied to the simulation-based DTA problem.

In this chapter, first, we conclude our review on the drawbacks and advantages of existing simulation-based algorithms in the literature. Second, the two new frameworks for DTA calculation are proposed and applied to the solution algorithm. Third, the full benchmark of new algorithms and several common algorithms in the literature for a large-scale network are provided. Finally, the best solution algorithms are highlighted.

This chapter is an updated version of the paper:

> Ameli, M., Lebacque, J. P. & Leclercq, L. (2019). Simulation-based dynamic traffic assignment : meta-heuristic solution methods with parallel computing. *Transportation Science*, (under first round of review).

3.1 Notations for this chapter

TABLE 3.1 – Specific notations in this chapter

Notation	Definition [units]
W	Origin-Destination (OD) pairs, subset of origin $\times$ destination nodes, $W \subset N \times N$.
P_w	Set of paths for w.
P_w^*	Set of shortest (i.e. minimum travel time) paths for w.
w	Index of OD pair, $w \in W$.
D_w	Total demand for w pair.
Tr_w	List of trips that travel for w.
Tr_p	List of trips that travel for w on path p, $Tr_p \subset Tr_w$.
p	Index of path, $p \in P_w$.
p^*	Index of shortest path, $p^* \in P_w^*$.
tr	Index of trip, $tr \in Tr_w$.
π_w	Cardinality of a set Tr_w: number of users traveling for w.
π_p	Cardinality of a set Tr_p: number of users on path p.
$TT_{tr,p}$	Experienced travel time of trip tr on path p.
TT_w^*	Minimum experienced travel time for w.
$\hat{TT}_p$	Mean travel time of trips on path p.
$\hat{TT}_w^*$	Mean travel time of trips on shortest path(s) of OD pair w.

3.2 Motivations

The aim of simulation-based DTA is to calculate the dynamic path flow distribution for all the Origin-Destination (OD) pairs in the traffic network depending on one equilibrium rule (Yang *et al.*, 2017, Ben-Akiva *et al.*, 2012). The well-known equilibrium rule is user equilibrium (UE), wherein all users experience minimum travel time (Wardrop, 1952). Recall that the simulation-based DTA contains two procedures: (i) a simulation-based dynamic network loading model to calculate experienced path travel times by considering the traffic dynamics for a given path flow pattern; and (ii) an algorithm for finding the UE solution (Marcotte & Nguyen, 1998). Here, we focus on the second procedure.

The mathematical foundations of the DTA problem are recalled in Friesz & Han (2018). Much research has shown that the DTA problem can be represented as a fixed-point problem (Wang *et al.*, 2018). To solve the fixed-point problem, a path-swapping descent direction method is used to reassign a fraction of the users at each step (Sheffi, 1985). The reassignment process is monitored to check whether the solution is improved or not. In other words, the algorithm consists in reassigning the share of the users who have chosen a non-optimal path to a more efficient alternative at each iteration of the equilibrium calculation (Levin *et al.*, 2014a). The foundation of all iterative algorithms is based on starting from the initial solution and updating the path flow distribution for iteration i using the following formula (Friesz, 2010):

$$z^i = (1 - \beta_i)z^{i-1} + \beta_i f[z^{i-1}] \tag{3.1}$$

where z^i is the path flow distribution of the iteration i, $f[z^i]$ is the descent direction and β_i is the step (descent) size of iteration i for the fixed-point algorithm. At each iteration, Equation 3.1 reassigns the users in order to move the traffic network toward the UE. Based on the review papers of Szeto & Lo (2006b) and Wang *et al.* (2018) almost all of the solution methods for DTA models used Equation 3.1 to find the network equilibrium. Many works can be found in the literature dedicated to finding the best β_i and $f[z^i]$ to improve the efficiency of the algorithms (see e.g., Nguyen & Dupuis, 1984, Drissi-Kaïtouni & Hameda-Benchekroun, 1992, Akamatsu, 2001, Bar-Gera, 2002, Dial, 2006, Gentile, 2016, Perederieieva *et al.*, 2015, Seshadri & Srinivasan, 2017, Xie *et al.*, 2018, Galligari & Sciandrone, 2017, Raadsen *et al.*, 2019). However, it is not possible to guarantee that fixed point algorithms converge towards the optimal solution (Ben-Akiva *et al.*, 2012) and there is no exact method for determining the step size (β_i) (Szeto & Lo, 2005b, Levin *et al.*, 2014b). All the algorithms based on Equation 3.1 mainly have two drawbacks when equilibrium is sought for large-scale networks:

1. The calculation should be done sequentially: the algorithms need to know the last iteration results to determine the next best path flow for the next iteration. Therefore, all the steps are in series because we need information (travel time) from the last simulation run.

2. The reassignment decision is taken only at the OD level and independently at each step: the algorithms do not consider the effect of shared links between OD path sets in the reassignment process. Intersections between OD flows are only taken into account when running the simulation to derive travel time.

To explain the drawbacks explicitly, let us consider the method of successive average (MSA) by Robbins & Monro (1951). Section 2.2 presented the MSA algorithm as the classical

method to solve the traffic assignment problem (Nagel & Flötteröd, 2016) and widely used in theory and application for DTA problems (Mounce & Carey, 2015). The $f[z^{i-1}]$ of the MSA algorithm in Equation 3.1 is extracted from the auxiliary path assignments obtained by the All-or-nothing procedure, i.e., everyone is placed on the shortest path and $\beta_i = \frac{1}{i+1}$. Consequently, in iteration i, the MSA algorithm tries to improve the path flow distribution by swapping a fraction $\frac{1}{i+1}$ of users to the shortest path(s) from each non-shortest path. Then, one simulation is launched based on the updated path flow distribution. We have to wait until the simulation run is finished to know the new link travel times and adjust the path flow distribution to be tested in the next iteration accordingly. This is the serial process of the MSA algorithm, which limits the solution space exploration and computational process.

Moreover, the MSA algorithm performs the reassignment process for each OD independently without considering that some OD pairs are connected because they share certain links and nodes. For instance, if we have a set of shared links between two OD pairs that are heavily congested, the algorithm will reduce the flow of the paths containing these shared links for all the OD pairs whereas reducing only the flow of a few OD pairs would have been sufficient. Therefore, we may trigger a high compensation of heavily congested paths. The MSA algorithm not only makes no provision to take into account the correlations between the OD assignment and the travel time, but also there is no accurate definition for $f[z^{i-1}]$ to consider this effect (Flötteröd, 2018).

All the works in the literature have the aforementioned limitations and perform the calculation in series. In this chapter, we explore a completely new area for overcoming the drawbacks of serial algorithms using meta-heuristic algorithms. Meta-heuristic algorithms are known to be parallelizable (Fonseca & Fleming, 1995). Traffic simulation, particularly micro-simulation, can be viewed as a complex system for which meta-heuristic algorithms are expected to be well-adapted because they are stochastic methods designed to search the solution space of complex and computationally costly problems (Yun & Park, 2006). We can better explore the solution space and will also run several simulations in parallel for certain path flow assumptions and take the decision on what the next exploration of the solution space should be. This overcomes not only the first drawback but also makes the algorithm capable of starting the optimization with different starting points at the same time. Moreover, a new layer of optimization is added to the algorithm to take into account the correlations between OD pairs through shared links.

The meta-heuristic algorithms are mainly applied to mathematical models that have a well-defined objective function or functions, and in some cases, constraints. With a large-scale simulation-based framework, we use the simulator as a black-box to calculate the objective function, making it necessary to run trials for optimization, e.g., the MSA algorithm is a trial and error process with the descent method. To the best of our knowledge, no study in the literature has yet applied a meta-heuristic algorithm directly to find the UE for simulation-based DTA models. This may be because it is difficult to handle the variables which in this case are path flows.

Meta-heuristics algorithms can be classified into two categories: single solution and population-based (Talbi, 2009). The single solution methods start with an initial solution and apply a process to improve the candidate solution in order to achieve the best solution by following a trajectory in the solutions space. The second class is population-based; the purpose of the methods of this class is to improve a set of solutions (population) by applying a specific process. This study proposes two new solution methods based on two categories

of meta-heuristic algorithms. The new extension of the Simulated Annealing (SA) method from the first category and the adaptive Genetic Algorithm (GA) from the second category are developed to solve the trip-based network equilibrium problem. The algorithms are generally developed to solve traffic assignments with parallel computation to consider more than one path flow distribution per iteration. It is possible that with a stochastic approach, more simulations must be run in order to explore the solution space compared to descent methods. However, with parallel simulation, the algorithm can counterbalance the number of simulation runs by finally reaching the optimum more quickly. Also, at the end of the optimization process, it is expected to achieve better solutions in terms of quality and closeness to the optimal solution because the algorithm explores the solution space more efficiently.

The layout of this chapter is as follows: the next section, Section 3.3, presents a discussion on the mathematical conditions for UE solutions. It also presents the two indicators we use to assess algorithmic performance. The simulation-based framework and two meta-heuristic algorithms are presented in Section 3.4. The numerical experiments are presented in Section 3.5. The results obtained are discussed in Section 3.6. Finally, we present concluding remarks in Section 3.7.

3.3 Problem statement

Depending on the kind of simulator used to perform the network loading and to determine travel times, the demand from origins to destinations can either be expressed as a continuous flow or units of vehicles. The flow-based approach usually corresponds to dynamic macroscopic models, while the trip-based approach is widely implemented in microscopic models (Ramadurai & Ukkusuri, 2011). The latter approach is certainly more realistic for reproducing traffic flows but it is also more challenging when deriving UE because OD flow should always correspond to integer numbers during the convergence process (Jordan *et al.*, 2017). The trip-based approach is used in this study to address the real large-scale DTA problem. In this section, we present the conditions of dynamic UE for the DTA model and the indicators used to determine the proximity of solutions to UE in the simulation-based framework.

Now, we present a short summary of the mathematical model for the trip-based DTA calculation. For more details, please refer to Section 2.3.1. Consider a network $G(N, A)$ with a finite set of nodes N and a finite set of directed links A. The demand is given and time-dependent for each OD pair. The period of interest (planning horizon) of duration H is discretized into a set of small time intervals indexed by τ. In an interval of τ, the traffic conditions are assumed constant for the DTA, i.e. travel times are averaged at the path level over each time interval. Therefore, all the time-dependent variables of the model are indexed by τ. To simplify the equations, we present the model for each departure time interval of τ. In Table 2.1 we introduce the notations of all the symbols and variables used in this chapter.

According to the definition, we have:

$$\hat{TT}_p = \frac{\sum_{tr \in Tr_p} TT_{tr,p}}{\pi_p} \quad ; \forall p \in P_w \tag{3.2}$$

$$\hat{TT}^*_w = \frac{\sum_{tr \in Tr_{p^*}} TT_{tr,p^*}}{\pi_{p^*}} \quad ; \forall p^* \in P^*_w \tag{3.3}$$

The network user equilibrium conditions with predefined travel demand and the users' departure times are (Peeta & Ziliaskopoulos, 2001):

$$\begin{cases} \hat{TT}_p - \hat{TT}^*_w \geq 0 \quad ; \forall w \in W,\ p \in P_w \\ \pi_p(\hat{TT}_p - \hat{TT}^*_w) = 0 \quad ; \forall w \in W,\ p \in P_w \\ \pi_p \geq 0 \quad ; \forall p \in P_w \end{cases} \tag{3.4}$$

Lu *et al.* (2009) reformulated the problem as a non-linear problem to minimize the gap function. The gap function is defined as the gap between the average path travel time and the average shortest path travel time. Therefore, the solution to this fixed-point problem is equivalent to finding the solution to the following variational inequality:

$$\sum_{w \in W} \sum_{p \in P_w} TT_{w,p}{}^* \left[\pi_w - \pi_{p^*}\right] \geq 0 \tag{3.5}$$

where π_{p^*} is the optimal number of trips on path p and $\pi_w, \pi_{p^*} \in \mathcal{H}$ satisfy the equilibrium. $\mathcal{H}$ denotes the flow constraints based on D_w. As mentioned before, finding the optimal solution for a large-scale DTA problem is hard to achieve, so indicators are required to measure the distance between the solutions and the optimal UE. The two convergence indicators of this chapter are presented in Section 2.3.2

3.4 Methodology

Determining the UE path flow distribution requires two main steps: (i) identifying the time-dependent feasible paths between ODs, and (ii) finding the optimal path flow with respect to demand and network dynamics. The first step refers to solving a time-dependent shortest path algorithm, which is a computationally expensive process in a large-scale network (Srinivasan *et al.*, 2018). One of the advanced frameworks in the literature express the solution algorithm in two loops: the outer loop to find the shortest path and the inner loop to find the optimal path flow distribution (Lu *et al.*, 2009). The main advantage of this framework is that it attempts to find the UE path flow distribution with a minimum number of running a time-dependent shortest path algorithm. Figure 3.1 presents the optimization framework of this chapter. Here, we focus on the inner loop, wherein the reassignment process is embedded. The green box in Figure 3.1 presents the classical inner loop structure. For the details of outer loop steps (step 1 to 4), readers can refer to Section 2.4.

The inner loop process starts with a single initial solution, which is generated in step 5. The optimization algorithm updates the path assignment based on the current state of the network (step 6). The reassignment process is executed based on the swapping algorithm (Equation 3.1 in classic approach) and a new path flow pattern is generated to be sent to the simulator to calculate the experienced travel time. Then one simulation runs in step 7, and the shortest path or paths is/are identified based on the simulation results of step 8. The solution quality indicators are calculated in step 9. Step 10 checks the convergence. The inner loop converges if the $AGap$ is unchanged or if the maximum number of iterations is reached. If the process has not converged, step 11 keeps track of the best solution obtained by current inner loop iterations; otherwise, the final solution is checked by step 12 to ensure that it is the best solution based on $AGap$. If the last iteration solution is the best, the inner loop is finished; otherwise, we run one more simulation for the best solution from step 11.

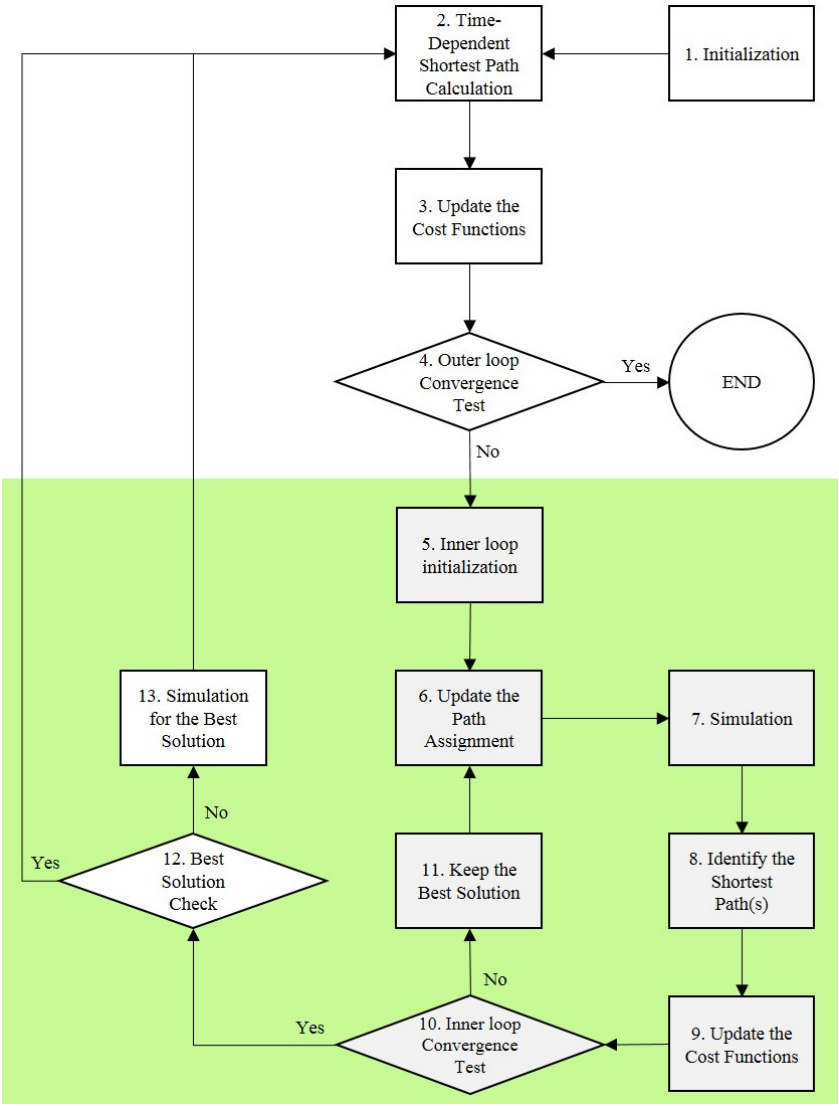


FIGURE 3.1 – Solution algorithm for trip-based dynamic network equilibrium

As mentioned before, the optimization process is executed in series. The meta-heuristic algorithms redesign the inner loop structure.

Before presenting the meta-heuristic algorithms, we recall from the previous chapter the three most common swapping algorithms in the literature for a large-scale DTA problem. They are considered in this chapter as benchmarks for demonstrating the efficiency of the new meta-heuristic algorithms. The swapping algorithm is embedded in step 6. The first algorithm is MSA (introduced in Section 2.2) which is the most common algorithm used in the literature (Foytik *et al.*, 2017). The second method is an extension of the MSA algorithm by Sbayti *et al.* (2007), called MSA ranking (MSAR). The MSAR algorithm was introduced in Section 2.5.2.1. Summarily, the MSAR algorithm ranks the users by the experienced travel time then swaps a quantity $\frac{1}{i+1}D_w$ of travelers with the longest experienced travel time to the shortest path. The design of the third method is based on the expected travel time reduction. This algorithm is called gap-based algorithm in this chapter. Lu *et al.* (2009) showed numerically that the gap-based algorithm obtains a better solution for UE than the MSA algorithm for the large-scale network. The gap-based algorithm was introduced in Section 2.5.2.3.

Briefly, it uses the same formula as Equation 3.1 and same $f[z^{i-1}]$ as MSA algorithm with $\beta_i = \frac{\hat{TT}_p - \hat{TT}^*_w}{\hat{TT}_p}$. We will now present the implementation of the meta-heuristic algorithms.

3.4.1 Simulated Annealing (SA) method

The SA algorithm is inspired by annealing in metallurgy. The basic simulated annealing algorithm is presented in Kirkpatrick *et al.* (1983). This study redesigns and adapts the classic SA to a simulation-based traffic assignment. Figure 3.2 presents the SA algorithm of this study.

The algorithm starts with an initial solution generated randomly. For solution S, the total gap $TGap(s)$ between the users' travel time and the shortest path travel time (Equation 3.6) is considered as the energy of the solution. The next solution is generated with respect to the current one based on the temperature (T) of the current iteration. The current phase of the iteration depends on the temperature of the process. Inspired by the physics of matter, this study distinguishes three different methods to generate a neighbor solution, gas, liquid, and solid; these methods represent the states of matter in nature. When the temperature is high ($T > \alpha$ where α denotes the boiling temperature), the *gas method* is applied. When running the SA algorithm, by decreasing the temperature, the algorithm enters the liquid phase ($\alpha > T > \alpha'$ where α' denotes the melting temperature) and then the *liquid method* is applied. When the temperature is quite low ($T < \alpha'$), the *solid method* is applied.

$$TGap(s) = \sum_{w \in W} \sum_{p \in P_w} \sum_{tr \in Tr_p} (TT_{tr,p} - TT^*_{tr,w}) \tag{3.6}$$

In the gas phase, we explore the solution space without limitation of any step size (β_i). Therefore, the candidates for the neighbor solution correspond to a random path flow distribution with respect to the demand value for each OD pair (feasible OD-assignment). The *Gas method* generates one solution as a neighbor. In the liquid phase, we target the exploration of the solutions space randomly and also apply step size methods. First, we apply a randomizing process on the current solution and obtain the first neighbor solution. Then we optimize it by applying the MSA to obtain the second solution and the Gap-Based method to obtain the third solution. The *liquid method* generates three candidates. In the solid phase, we execute the same process as in the liquid phase but without randomization. This means the two solutions are generated based on the current solution (Figure 3.2).

Afterwards, the algorithm runs parallel simulations (a maximum of three simulations) to update the network based on new different path flows obtained from the previous step. For a new solution or solutions s', the total gap $TGap(s')$ is calculated and corresponds to the energy of the solution (E) compared to the current solution s.

The last step consists in making a decision on accepting one of the best new solutions based on $TGap$ compared to the current solution of the algorithm. The acceptance decision is made by the Bernoulli trial.

$$AP_s = P(S'_{\text{accepted}} = 1) = e^{\frac{-\nabla E}{T}} \tag{3.7}$$

where AP_s denotes the probability of accepting solution s', S'_{accepted} denotes the binary decision variable and $\nabla E = TGap(s') - TGap(s)$. If the solution s' is accepted, the current

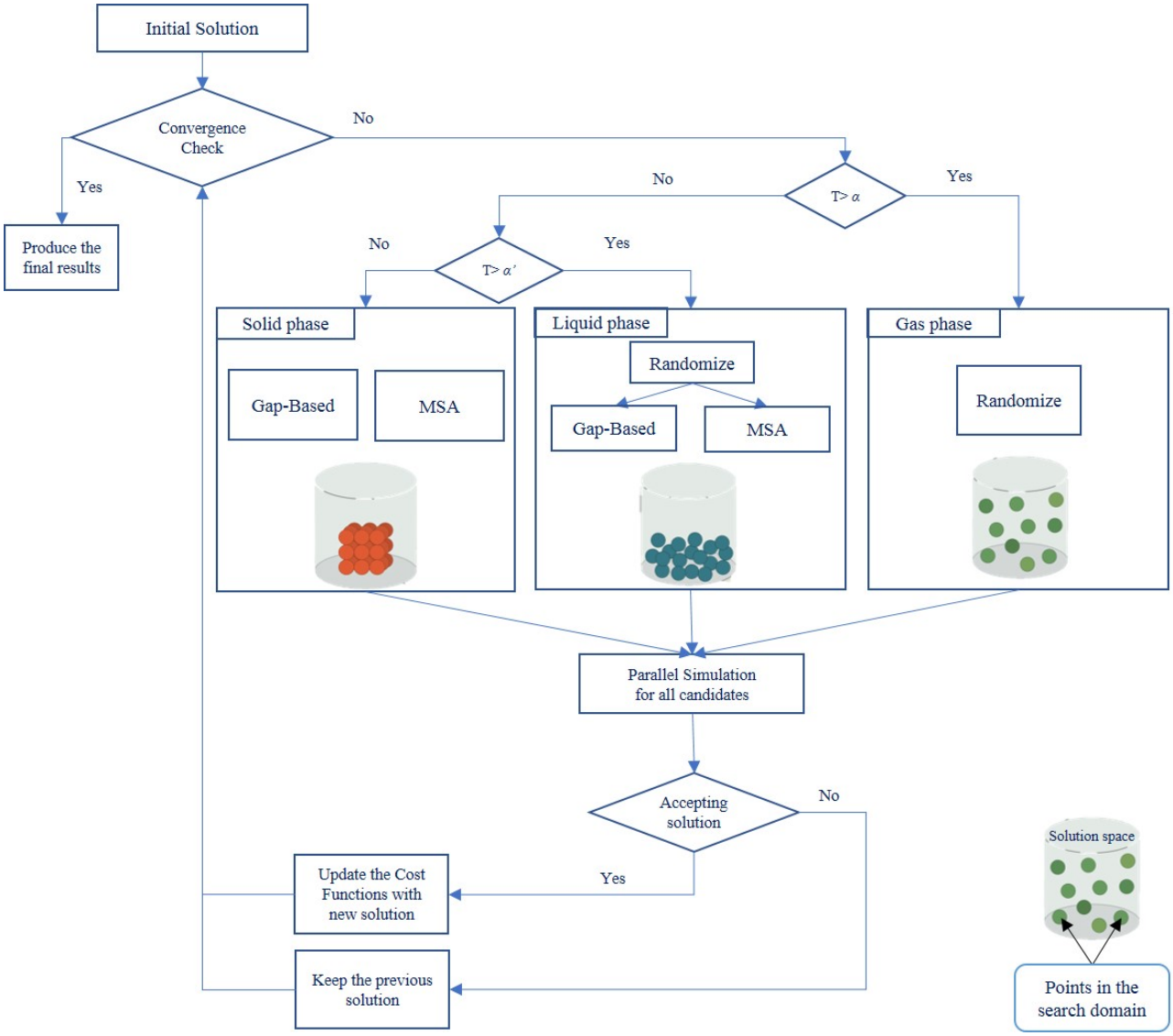


FIGURE 3.2 – SA solution algorithm flowchart

solution s is replaced by s'. At the end of each iteration the temperature is decreased by the following formula:

$$T = \frac{T_0}{ln(i+1)} \tag{3.8}$$

where T_0 denotes the initial temperature. It should be remembered that i denotes the iteration index. In this study, the starting temperature is set to $T_0 = 3000$. The quality of the solution is evaluated in the convergence check step, which is similar to step 10 in Figure 3.1.

The SA algorithm considers more than one solution per iteration in the liquid and solid phases. In addition, the exploration and exploitation degrees are changed based on T. The algorithm explores more in the gas phase and liquid phase when the temperature is high and exploits more in the solid phase when the T is low.

3.4.2 Genetic Algorithm (GA)

The genetic algorithm was first proposed by Holland (1992); it is inspired by natural genetic variation and natural selection; selection, crossover and mutation are the main operators of this approach. In the genetic algorithm, we use the terms chromosomes and genes to refer to the different segments of an individual. In our implementation for the genetic algorithm, we consider our solution and the $TGap$ as an individual (DNA) with a fitness value, our ODs-assignment as chromosomes, and the path flows as the genes. Figure 3.3 illustrates the solution structure for GA. A gene is identified by a path (p) and contains the path flow value (π_p). A chromosome is the OD assignment, identified by w, which contains the genes of all the corresponding paths ($p \in P_w$). A DNA is the full set of chromosomes that constitutes one individual solution. Finally, the set of individuals constitutes a population.

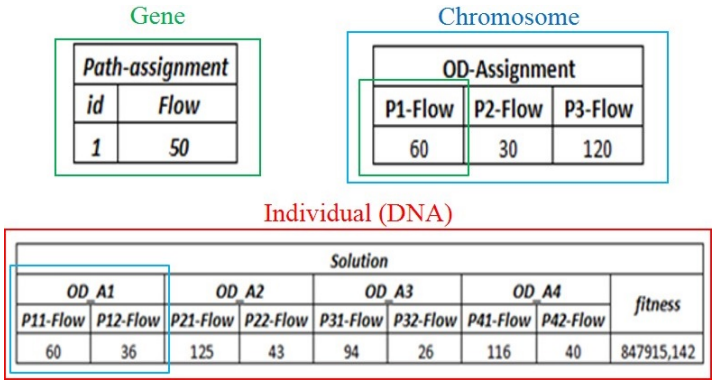


FIGURE 3.3 – Solution structure in the GA case

Figure 3.4 presents the application of GA to the DTA problem. The GA process starts by generating the initial population (initial set of individuals). In this study, we generate a random population. This study designs a two-layered GA process to search the solution space by changing the path flows in the inner GA and take into account the correlations between OD pairs by considering a different combination of OD assignment in GA-operators. In other words, the classical fixed-point algorithms plus a random method is applied in the inner GA, and the GA operators in one upper level generate different combinations of OD assignments to improve the population. The steps of GA applied to traffic assignment are as follows:

- Selection: we use a random selection based on the crossover rate (Cr) and population size (PS) to compute the number of selected solutions for the crossover process:

$$SS = PS \times Cr \tag{3.9}$$

- Crossover: we apply a non-uniform crossover by using a bit-vector mask method (Maini *et al.*, 1994). We select two different solutions (parents) from the set of selected solutions; we apply the crossover between each pair of solutions. As a result, we will have new solutions; the two new solutions will have a part of each parent.

- Mutation: we apply the mutation operator for a set of selected solutions; by replacing one OD assignment (Chromosome) of the solution by another chromosome from another solution, this operator aims to increase the quality of the worst solution. The

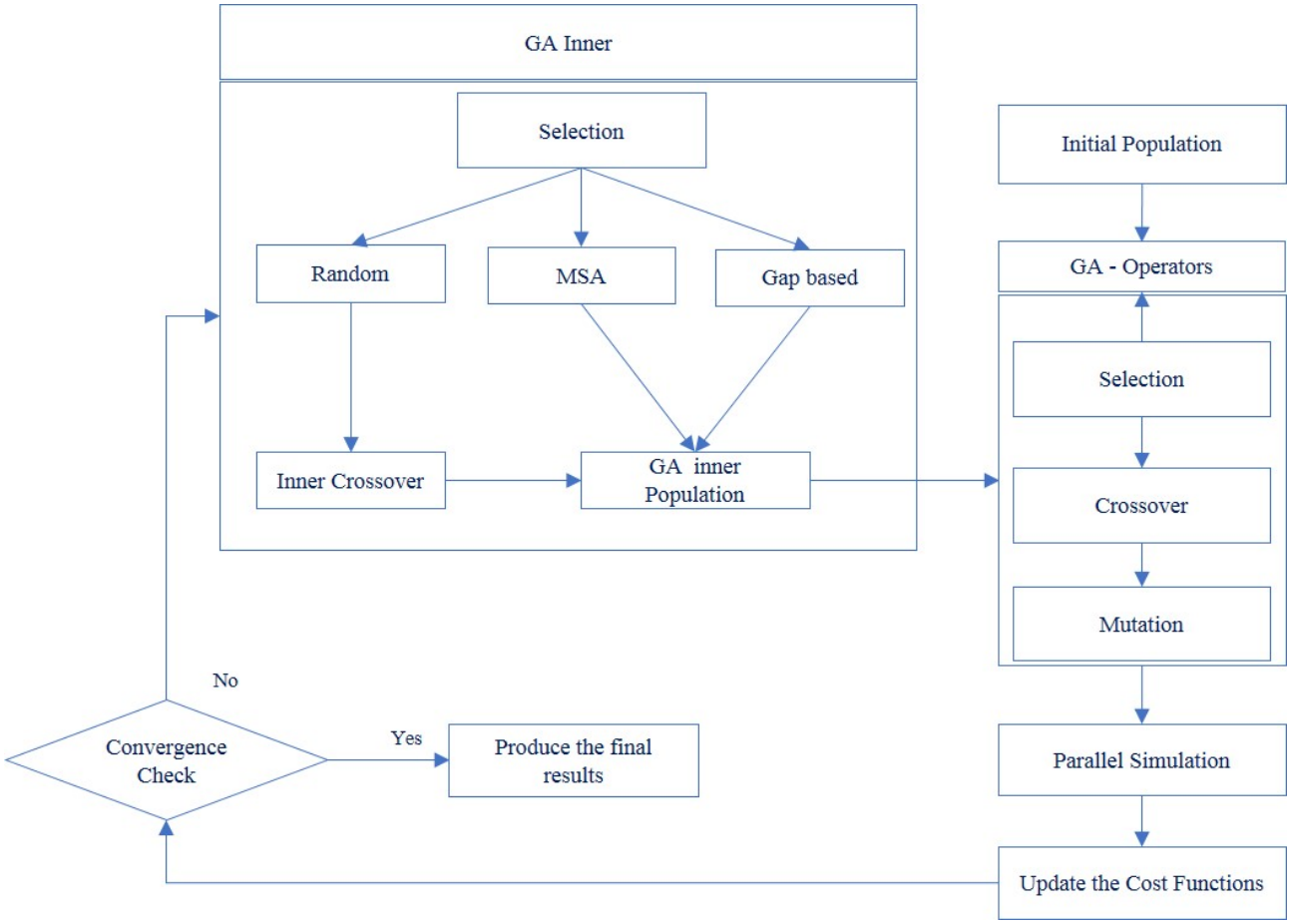


FIGURE 3.4 – GA solution algorithm flowchart

possibility of the mutation for one chromosome is calculated based on the quality of the chromosome:

$$MP_w = \frac{TGap_w(s)}{TGap(s)} \tag{3.10}$$

where $TGap_w(s)$ denotes the total gap of the OD assignment w in solution s.

- Parallel simulation: all the new solutions obtained from the previous steps are simulated in parallel to calculate the fitness function, which is the $TGap$ in this study.
- Replacement: after applying the different GA operators and parallel simulation, the size of the evaluated population set is increased. To keep it as a fixed value (PS), we apply the selection operator, and we keep the best solutions as a replacement strategy.
- Convergence check: the algorithm converges when the maximum number of iterations is reached, or when the algorithm tends to stagnation. In order to check the stagnation, we use a "Stagnation Factor (SF)" indicator: when SF tends to zero, our process tends to stagnation, which means the quality of the population solution has not changed. The Stagnation Factor is presented as follows:

$$SF = 1 - \frac{TGap^i_{\text{Max}}}{TGap^{(i-1)}_{\text{Max}}} \tag{3.11}$$

Where $TGap^{i}_{\text{Max}}$ denotes the total gap of the worst quality DNA (high $TGap$) in the population of iteration i. If the algorithm does not converge, the iteration index is increased, and the inner layer of GA is applied to search the solution space by gene modifications inside the chromosomes of the selected solutions.

- Ga Inner: as mentioned before, the Initial Population, Selection, Crossover, Mutation and Replacement are the basics of GA. In this study, we extend the GA by adding a new operator, called "GA Inner". The purpose of this operator is to create diversity in the current population (diversity inside the OD assignment). This approach is applied using three different methods: the MSA, the Gap-based methods, and the adaptive random method.

  - Adaptive random method: the foundation of this method is based on the genetic algorithm. We consider the selection and crossover operators for this method.
    * Inner Selection: select a set of solutions from the main population. The worst solutions that have a large $TGap$ are selected because the aim of this selection is **to increase the quality of the population** by improving the solutions.
    * Inner Crossover: We consider the OD assignment (Chromosome) as an individual and the flow of each path (Gen) as a chromosome in order to apply the crossover in the same way as in the previous layer of the optimizer. By applying the crossover, we risk having a non-feasible OD assignment with respect to the demand constraint.To solve this problem, we use the following process to keep only the feasible solutions:
      · Step 1: put zero for the flow of the worst path (wp) which has the maximum travel time.
      · Step 2: apply the crossover on the other paths of the current OD.
      · Step 3: compute R, the difference between total flow of the current chromosome and the demand level of OD pair w:

$$R = \sum_{i \neq wp} \pi_i - D_w \tag{3.12}$$

        If $R \leq 0$ then we put the rest of the flow on the worst path ($X_{wo} = R$); otherwise we reject this chromosome.
    * The MSA assignment method and Gap-based method (introduced in Section 3.4) are also applied to the chromosomes of the selected solutions in GA inner.

- The new solutions are injected into the main population and the algorithm iterates while GA converges.

Note that based on the early study of Srinivas & Patnaik (1994), we set the crossover rate set to $Cr = 0.5$, and the mutation rate is fixed to 0.4. The PS is set to 10 individuals in this study.

3.5 Numerical experiments

The Symuvia platform, including the trip-based simulator (Section 1.2.2) and the command module: SymuMaster (Section 1.2.3) is used in order to compare the solution algorithms. Note that in this chapter, we consider a prediction period of 20 minutes and an assignment period of 15 minutes (Mahmassani, 1998).

The real network of Lyon 6e + Villeurbanne (Figure 3.5) is considered as the large-scale test case. This network has 1,883 Nodes, 3,383 Links, 94 Origins, 227 Destinations. The network is loaded with 47,341 travelers of all ODs with given departure times in order to represent the two morning peak hours of the network between 7:30 to 9:30. The demand profile comes from the study of (Krug *et al.*, 2019). The dynamic loading represents the saturation state of the network. To show a quick and synthetic overview of the network state, we plot the Macroscopic Fundamental Diagrams (MFD) of the network in Figure 3.6(a). This diagram represents the overall evolution of the traffic conditions in the network by plotting the total travel distance vs. the total travel time. Note that total travel distance is proportional to the mean network flow while total travel time is equivalent to vehicle accumulation. Figure 3.6(b) shows the time and evolution of the mean speed over the full network.

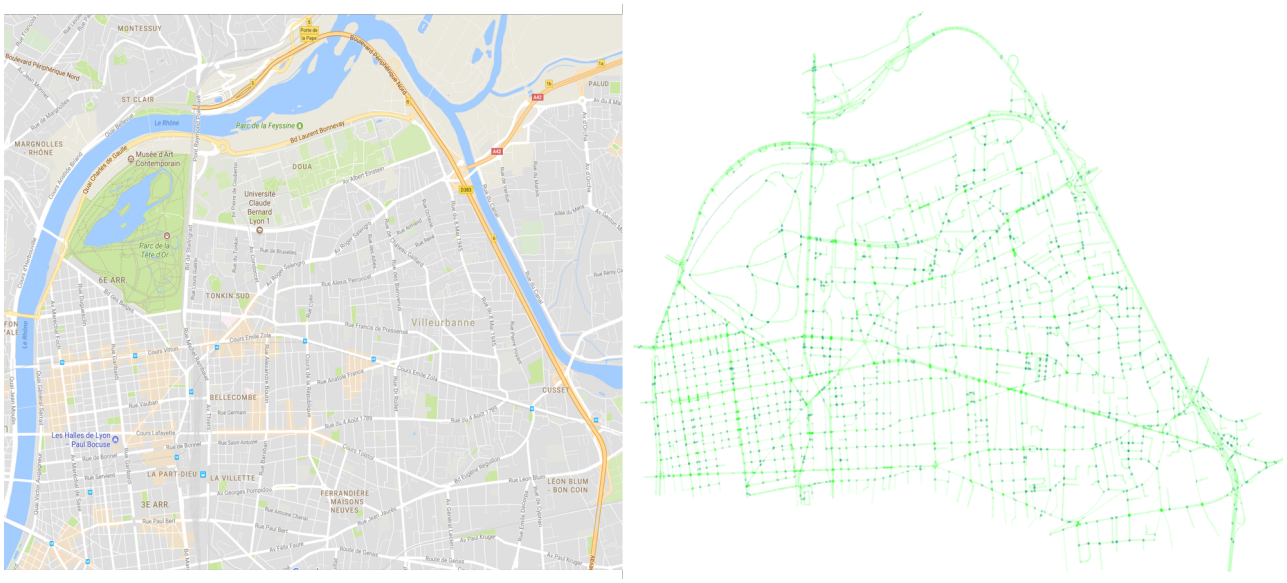
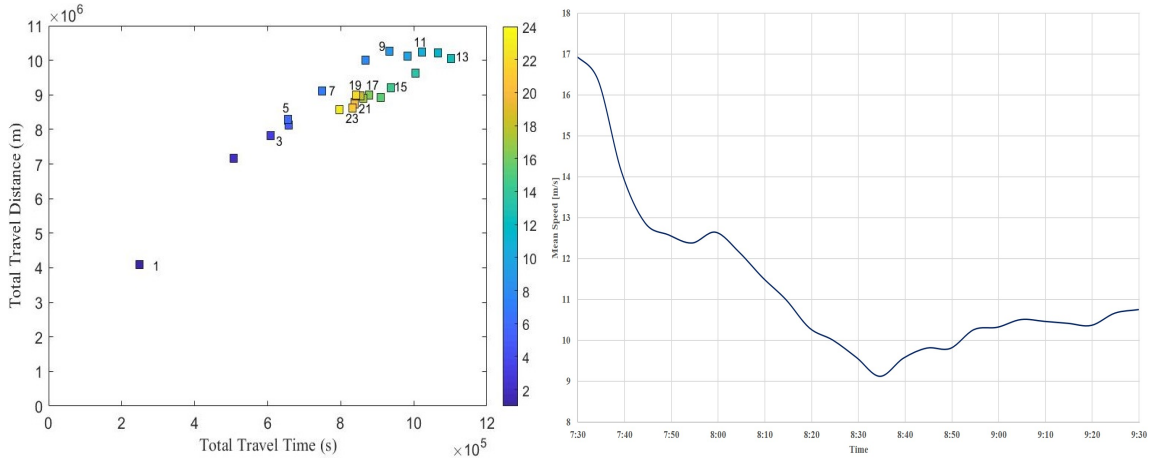


FIGURE 3.5 – Lyon 6e + Villeurbanne: Mapping data ©Google 2019 and the traffic network used by Symuvia

The MFD shows the state of the network for the period of 5 minutes during the full simulation. For instance, point number 15 of the MFD (Figure 3.6(a)) shows the total travel distance and total travel time of all the travelers in the network between 8:40 to 8:45. First, the MFD curve increases from (0, 0) and the traffic states remain under-saturated (point numbers 1 to 8) when demand is light, in this case from 7:30 to 8:10. Afterwards, travel production, which is equivalent to the total travel distance for a given period of time, stabilizes while the accumulation (or total travel time) continues to increase (point numbers 9 to 13). This corresponds to the saturation level occurring from 8:10 to 8:35. The decrease in travel production and accumulation (point numbers 14 to 24) shows that the network starts



(a) The macroscopic fundamental diagram (b) The evolution of the mean speed

FIGURE 3.6 – The demand scenario of Lyon 6e + Villeurbanne.

to exit the saturation level at the end of the simulation period and slowly returns to the under-saturation level from 8:35 to 9:30.

3.6 Results

In this section, the results of the implementation and application of SA and GA to the simulation-based DTA problem for the large-scale test case are presented in view to comparing them with the MSA, MSAR, and Gap-based algorithms. All the experiments are first initiated by the first outer loop with the All-or-nothing assignment algorithm (see Step 1 in Figure 3.1). In order to compare the performance of the algorithms, we use the same outer loop component with different methods in the inner loop.

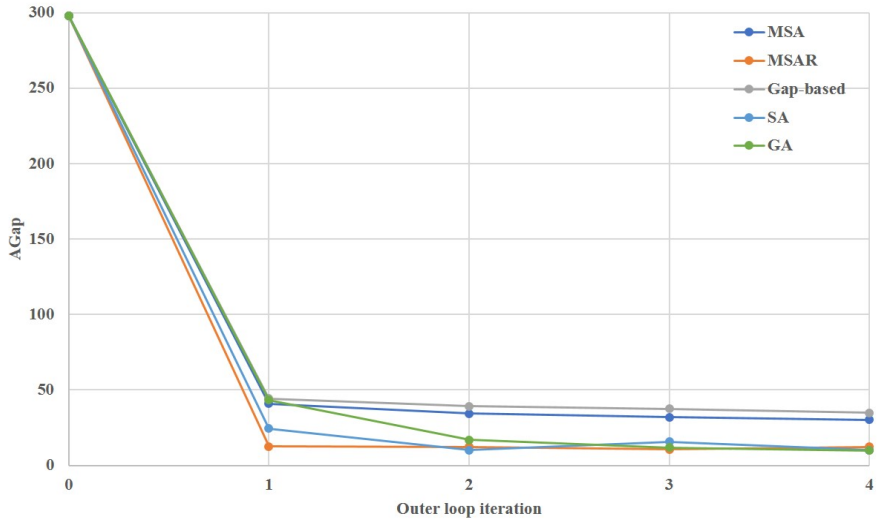
In this chapter, we impose a limit on the maximum number of iterations and compare the final solutions obtained by the different algorithms. This is to restrict the computational cost. Each classical algorithm is run for ten outer loops. This means that users have to finally choose from a minimum of eleven paths (the first path comes from Step 1) for each OD pair. For SA and GA, in order to avoid blocking the random methods with a small number of paths per OD, we start the first outer loop with the K-shortest path algorithm ($K = 3$) and continue in the same way as the other methods with one shortest path per outer loop iteration. This means the maximum number of outer loops for meta-heuristic algorithms is 8 in order to keep the total number of minimum paths at 11. In addition, the computation time of the time-dependant k-shortest path algorithm at each assignment period is much higher than a single shortest path discovery process in the considered large-scale network. The extra computation time for meta-heuristic algorithms approximately equivalent to having two less outer loop iterations than classical algorithms in our test case. The inner loop runs for a maximum of twenty iterations ($i_{max} = 20$) for all the algorithms. With a 15 min assignment interval, we will then have a maximum of 1600 simulations for the classical methods. With the meta-heuristic algorithms, the maximum can be exceeded as some simulations will run in parallel.

The aim of the experiment performed on the Lyon 6e + Villeurbanne network is to examine the convergence pattern and validate the solutions for a large-scale network case. Thus, the UE is calculated for the considered network five times with the following algorithms: MSA, MSAR, Gap-based, SA, and GA. The $AGap$ indicator is used to evaluate the quality of the solution. Figure 3.7 presents the convergence pattern for the five algorithms. Figure 3.7(a) presents the convergence pattern of the first four outer loop iterations. Figure 3.7(b) shows the convergence patterns of all the algorithm from outer loop 4 to the end of the optimization process, and magnifies the difference between the convergence patterns of the algorithms. Note that each algorithm is terminated once the maximum number of outer loops is reached. Another possibility, which does not occur here, is that convergence is reached when the solution is not changed in two consecutive outer loops or no new shortest path is found for all the ODs.

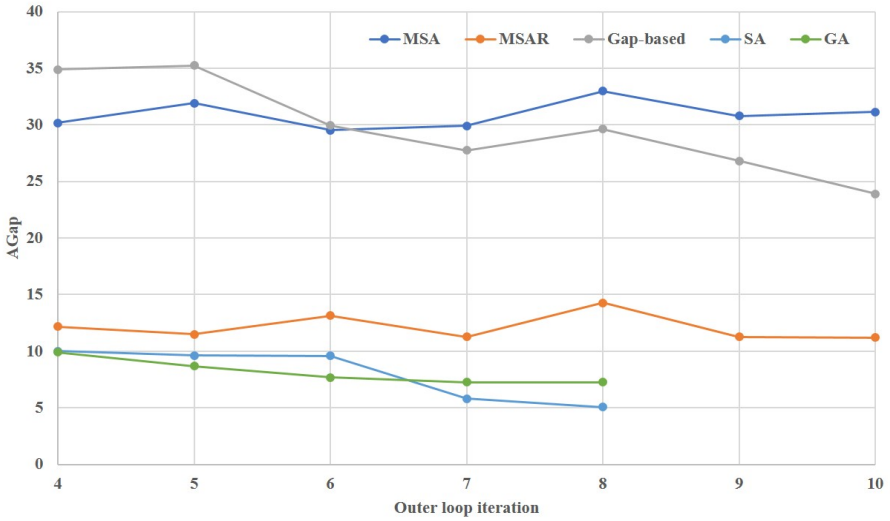
In Figure 3.7, the convergence pattern and the final result of MSA and Gap-based algorithms are close, but MSAR dominates both algorithms easily from the first outer loop. Note that the better performances of the MSAR have a cost. Each inner loop iteration takes longer because of the sorting of users by the experienced travel time for each OD. MSAR algorithm converges faster than SA and GA at the first outer loop, but it is dominated by both meta-heuristic algorithms after the fourth outer loop. Both meta-heuristic algorithms produce better convergence patterns than the classical algorithms. Note that in GA, the best DNA of the population is sent to the outer loop. Therefore, the convergence pattern of GA is always decreasing compared to the other methods.

The results for the performance indicators of all the algorithms are presented in Table 3.2. As expected, the numbers of total simulations for SA and GA are larger than those of the classical methods around 320%, despite the fact that the computation times are significantly lower than those of the classical methods because of the parallel simulation framework. Moreover, the solutions obtained by the meta-heuristic algorithms are significantly closer to the optimal UE than those of the classical methods. The $AGap$ of the GA solution is better than that of the classical method, indeed it is 76% better than the MSA algorithm. The SA algorithm manages to reduce the UE $AGap$ of the MSAR method by more than 54%, the MSAR method being the best classical method compared to the MSA and Gap-based methods. The Violation indicator also shows that GA and SA work much better than the MSA and Gap-based methods (reduction of 82%) and even better than the MSAR method as they reduce the Violation by one third of its value (-6%). In addition, the meta-heuristic algorithms dominate the others regarding the percentage of incomplete trips. The incomplete trips denote the share of travelers who could not finish their trip by the end of the simulation in the final path flow distribution of each algorithm. A lower number of incomplete trips means a lower total travel time spent in the system over the simulation period. The SA algorithm finds the closest solution to UE (minimum $AGap$) in this study. Moreover, the final solution of SA has the best value for other quality indicators that are highlighted in green in the SA row in Table 3.2.

In order to evaluate the performance of the algorithms, the computation time (CT) should also be considered. The CTs of both meta-heuristic algorithms are better than those of the classical methods. In particular, it is significantly better than the MSA method (green values in GA row inside Table 3.2). Note that each iteration for the MSAR method takes longer than for the MSA method, but the MSAR method dominates the MSA method at the end as it requires fewer iterations than MSA. Note also that because of the network size, the DTA



(a)



(b)

FIGURE 3.7 – Convergence patterns for the swapping algorithms.

process over the full simulation period requires considerable computational resources, i.e. about a week for the classical methods (MSA, MSAR, Gap-based). Therefore, the computational improvement obtained by switching to the meta-heuristic methods is huge; 36 hours (a day and a half) for the SA algorithm and 67.5 hours (two and a half days) for the GA when compared to the MSA algorithm.

The GA can take the most advantage of parallel computing as each DNA can be run as a separate thread. From the application standpoint, every simulation usually needs one central processing unit (core) of the computer. The classical algorithms are run in series, so they use one core per iteration. The performance of SA is very good in terms of solution quality, but the potential for parallelization is limited. Only the difference exploration methods running in parallel can be assigned to different threads. The SA generates a maximum of three new path flow patterns (*liquid method*) per iteration and which must be simulated at once.

TABLE 3.2 – Solution quality and performance indicators

Indicator/ Method	Number of simulations	Incomplete travels (%)	Violation	Final *AGap*	CT [Hours]	Improvement to CT compared to MSA	MAX number of cores used
MSA	1485	6,04%	35.34%	31.12	183.74	-	1
MSAR	1145	5.48%	9.01%	11.19	175.41	4.54%	1
Gap-based	1512	6.71%	22.70%	23.90	186.16	-6.13%	1
SA	2317	5.07%	2.61%	5.05	147.86	14.85%	3
GA	9242	5.26%	3.09%	7.27	116.10	21.48%	12

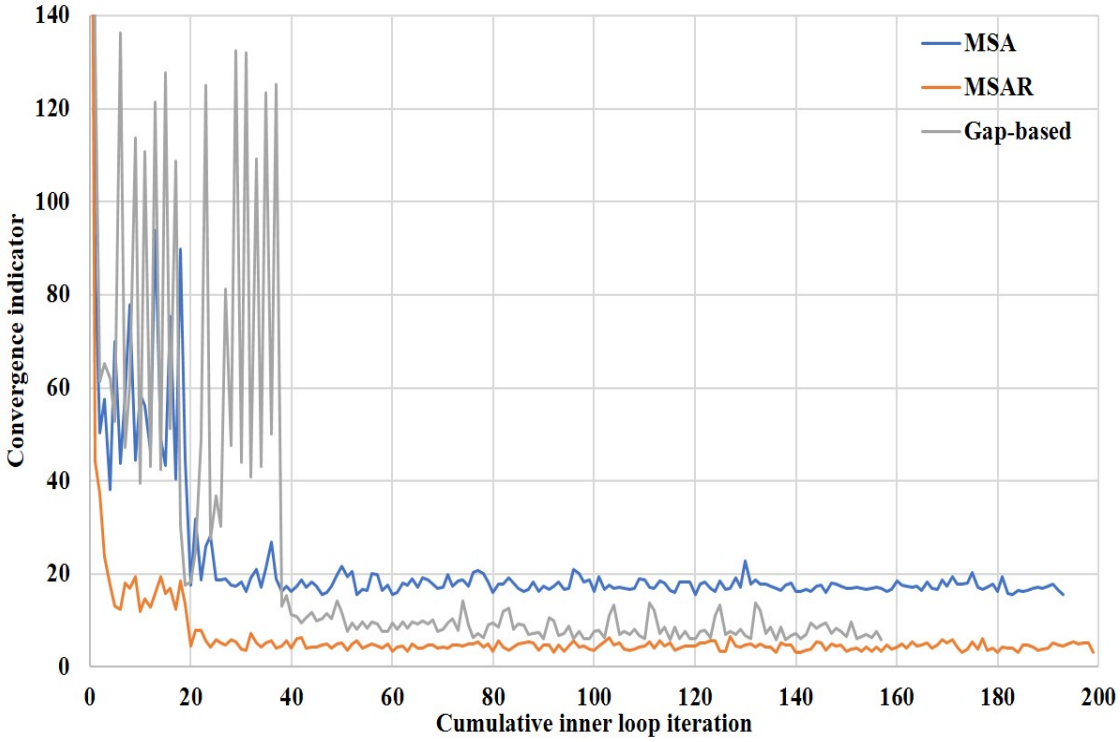
Therefore, we use a maximum of 3 sub-methods and then three threads, and finally three cores for the simulation process. For the GA algorithm, according to PS, Cr, and the mutation rate, we will have a maximum of 20 new individuals (children) from the GA-operators and the GA Inner. We limit the number of cores to 12 because all the experiments are conducted on a 64-bit personal computer with 12 cores. If the number of new individuals is bigger than 12, the algorithm executes in two successive phases, the first 12 simulations in the first phase and the remaining individuals in the second phase.

To analyze the behavior of the algorithms, the inner loop convergence patterns of the algorithms are presented in Figure 3.8. As we are looking for the closest solution to perfect UE ($AGap = 0$), the value of $AGap$ was not used as a stopping criterion. Before analyzing the convergence pattern, we explain why fluctuations exist in the convergence figures. First, when the outer loop index is changed, a new shortest path is added to the system, and the inner loop $AGap$ is calculated by considering the new shortest path, so $AGap$ increases for the first inner loop and then decreases after executing the inner loop iterations. The second reason is that the swapping algorithm does not necessarily improve the $AGap$ at every iteration. The outcome will be even worse when using trip-based approaches because their discrete nature makes them less stable. Thus, we expect more variations, especially when the step size is fixed as in classical methods.

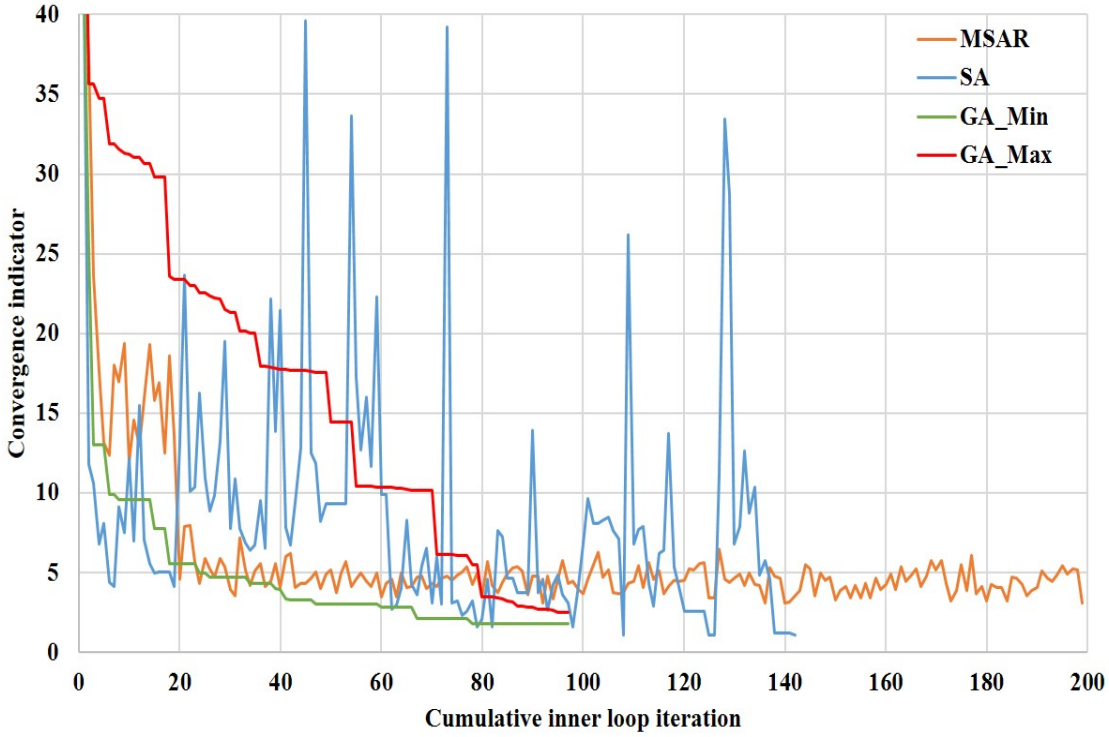
Figure 3.8(a) shows that MSA and Gap-based algorithms are dominated by MSAR. The convergence pattern of MSA shows that by increasing the inner loop index, the flexibility of the method for exploring the solution space is decreased. The same scenario occurs for the Gap-based method with one major difference, which is the high searching flexibility at the beginning of the process. This stems from the gap criterion in the swapping formula. However, decreasing the step size prevents the Gap-based method from finding a better solution. The MSAR algorithm also suffers from the step size, but with ranking technique it is able to find a better solution than the other classical algorithms. We keep the MSAR in Figure 3.8(b) in order to compare it with meta-heuristic algorithms.

The SA algorithm has a regular variation behavior in each outer loop, which corresponds to the three optimization phases. The gas phase has no limitation for exploring the solution space; hence, we have high variations at the beginning of the outer loop. Then, the variation is decreased in the liquid phase and afterwards the SA algorithm looks for the local optimum in the solid phase. This approach finds the best UE solution in comparison with the other algorithms.

Two values represent the convergence pattern of the GA algorithm in Figure 3.8(b). The lower value (GA_Min) is the best solution in the solution population of GA, and the other value (GA_Max) represents the worst individual of the current population. The interval



(a)



(b)

FIGURE 3.8 – Convergence patterns for the inner loop iterations of the swapping algorithms.

between two $AGap$ values shows the range of the other individuals in the population. The initial population at the beginning of each outer loop is generated randomly, so the new shortest path attracts users before starting the inner loop iterations. Note that the outer loop finds the new shortest path and it has zero flow in the final solution of the previous outer loop. Thus, the newly generated solutions for the initial population decrease the variation of $AGap$ compared to the other algorithms when starting a new outer loop. During the inner loop, the quality of the best solution never increases because the top ten solutions are always kept. The range of population $AGap$ is decreased until the process converges. In GA, with two layers of optimization, we consider a wide range of solutions at each iteration while the algorithm has less focus on finding the local optimum. The parallel simulation framework helps GA to converge faster than the other methods with minimum CT.

3.7 Discussion

This chapter has introduced a new branch of optimization algorithms for simulation-based DTA models. Two meta-heuristic algorithms were applied to solve the network equilibrium problem. The SA algorithm has three layers of optimization for searching the solution space. The algorithm starts in the gas phase and searched the feasible solution space without limitations. Then, the temperature of the algorithm is decreased and the search space is narrowed by shifting the optimization process from the gas phase to the liquid phase and then to the solid phase (Figure 3.2). In the GA framework, a new layer in the optimization process has been added to account for correlations between OD assignments (Figure 3.4). Moreover, GA considers a set of solutions instead of a single solution at every iteration. Both algorithms were implemented using parallel computing for calculating the UE in the DTA model. The new algorithms were applied to the real large-scale network of Lyon6e + Villeurbanne in a simulation-based DTA model for a time period of two hours. To compare the SA and GA with existing methods, we have considered three algorithms from the classical approach in the literature: the MSA algorithm, which is one of the commonest solution algorithms implemented in the field of DTA, and two recent extensions of the MSA algorithm for simulation-based DTA, namely MSAR and Gap-based algorithms.

The results show that meta-heuristic algorithms dominate classical methods. They provide wide coverage when exploring the solution space. Hence, they find a better solution in terms of closeness to the optimal UE solution. The SA has provided the best solution, which was significantly better than the best solution of obtained by the classical methods (MSAR). The parallel simulation framework helped the meta-heuristic algorithms to run more simulations compared to serial algorithms and speeded up the exploration process by more than 37%, meaning that we could obtain the solutions about three days in advance.

Conclusion of part I

In this first part, we have investigated solution algorithms for finding user equilibria by considering trip-based dynamic network loading. This problem is computationally challenging for large-scale networks. A large body of literature exists and proposes solution algorithms. Most of the works done are based on fixed-point algorithms and iterate in series to improve the current solution.

We, first, performed a full benchmarking of existing algorithms. Second, we highlighted the current drawbacks of the swapping process and proposed several solutions to overcome them and speed up convergence. Numerical experiments were performed for different network sizes and levels of saturation in order to compare different algorithms. The results showed that network size and saturation level have an impact on the performance of solution algorithms to solve the DTA problem. For instance, the Method of Successive Average algorithm, the most common algorithm in the literature, exhibited good performance in the small-scale network, but it did not provide a good solution in terms of closeness to optimal at large-scale.

The investigation in Chapter 2 showed that the computational costs of the classic algorithms were prohibitive for the large-scale network. We noticed that the process of simulation-based optimization was performed serially. This fact motivated us to design a new framework in Chapter 3 to take advantages of parallel computation. The new framework allows not only executing several simulations at the same time but also using the larger class of evolutionary algorithms to solve the DTA problem. Therefore, two new meta-heuristic algorithms were used to overcome the disadvantages of the serial algorithms: the first one derived from the Simulated Annealing (SA) framework and the second one from that of Genetic Algorithm (GA). The new design appears promising to solve the DTA problem in the large-scale network.

The contributions of this part can be summarized as follows:

- ✓ Benchmarks of quality and speed for solution algorithms in the literature for different network sizes and levels of saturation. The best configuration of the classical solution algorithm that achieved the best compromises between quality and computation time for all network sizes and saturation levels were presented.
- ✓ Identification of the solution algorithms' drawbacks in the domain of simulation-based DTA. The study of classical optimization framework showed that the exploration of the solution space in order to find the equilibrium is very limited because of the serial process.
- ✓ Application of the parallelization approach to the simulation-based equilibrium calculation. The parallel simulation framework helps the solution algorithms to run more simulations compared to serial algorithms and amplifies the exploration process.
- ✓ Proposition of meta-heuristic algorithms (SA and GA), for the first time, to solve simulation-based DTA. The results show that meta-heuristic algorithms dominate classical methods. They provide wide coverage when exploring the solution space. Hence, they find a better solution in terms of closeness to the optimal UE solution.

The investigations conducted in this part have been focused on the solution algorithm to find network equilibrium in the large-scale network. However, as we have multi-modality

in real test cases, there exist multiple equilibria in the system. The tools provided in this part can find one of the equilibria. Therefore, one question is: since there are possibly several equilibria, does the output of DTA models represent the effective equilibrium of a real network? Note that we consider within-day DTA problems in this part, while the day-to-day approach may help identify the set of equilibria. In addition, the network design can be changed in the long term (e.g., several new transportation facilities are added to the system). The way this long term process is carried out can have an impact on which equilibrium solution is reached at the end by the long-term equilibration. Thus, further research directions should now investigate (i) the existence of multiple equilibria and (ii) the impact of the history of the network on the end-result that represent the real state of the network.

Moreover, most studies in the literature stop after calculating the equilibrium path flow distribution. However, analyzing the output of DTA models may derive practical insights. Thus, designing a framework to analyze the equilibrium solution and propose improvements to the system should also be included in our scope. All these research directions are then detailed below:

- Investigate the existence of multiple equilibria in practice.
- Evaluate the impact of the network design history on the final equilibrium in the multimodal context.
- Study the equilibrium path flow distribution in order to design a framework to analyze the equilibrium.
- Investigate the possibility of a control strategy application based on equilibrium analysis in order to improve system performance.

Part II

Applying DTA method to new network problems

Introduction

This part aims to use the optimization tool that was previously developed to investigate new network problems. Analyzing the properties of the network equilibria is the main focus. The purpose is to identify the opportunities wherein the system performance can be improved. In this perspective, close attention is paid to the non-unicity of network equilibria and the differences between the path flow distributions of the various equilibria.

This part is split into two different studies. The first one deals with the unicity of the user equilibrium in a real multimodal large-scale network. In order to investigate the existence of multiple equilibria, the day-to-day approach is applied to address the equilibration process. In particular, we look for multiple equilibria when we have multiple network design history. The second study investigates the impact of different demand levels on distinct equilibria (e.g., User Equilibrium and System Optimum). The idea is to analyze path flow distributions pertaining to different equilibria and find situations wherein control strategies can be effective. To this end, the road banning strategy is applied to a real test case.

In both studies of this part, the ideas are first introduced in a static context and evaluated analytically. Then the study is extended to a dynamic settings in order to address the real test case.

Contents

4. Impact of network design history on multimodal UE

Dynamic loading of the multimodal transportation network leads to multiple solutions for user equilibrium because the link travel time function is no longer strictly monotone. The immediate consequence is that the final solution not only depends on the initial state but also on the convergence process. This chapter focuses on the second aspect when considering a long-term, day-to-day learning process. In particular, it investigates the network design history, i.e. the long-term evolution of the network, including opening new multimodal options and its impacts on the final network equilibrium. First, the analysis focuses on static network loading with different successive configurations. This structure makes analytical derivations possible, highlighting the problem. Then, a more realistic setting is studied in simulation. A large-scale multimodal network with the flexible opening over time of three possible transport facilities is investigated to evaluate the uniqueness of the final equilibrium.

This chapter is an updated version of the paper:

> Ameli, M., Lebacque, J. P. & Leclercq, L. (2019). Non-unicity of day-to-day multimodal user equilibrium: the network design history effect. *Transportation Research Part B: Methodological*, (under first round of review).

4.1 Motivations

The issue of unicity for UE has long been a subject of concern in the literature on traffic assignment problems (Beckmann *et al.* (1956); Daganzo (1985); Mounce & Smith (2007); Iryo & Smith (2018)). Iryo (2015) defines the unicity of the UE solution as one of two situations: only a unique link flow value vector meets UE requirements, or the solution set for the network equilibrium model is convex. A key argument for unicity is a strictly monotone travel time function with respect to the number of travelers that use a path (Smith (1979); Aashtiani & Magnanti (1981); Florian & Hearn (1995)). Traffic assignment models address the network equilibrium problem, including the travel time calculation, mathematically.

The Static Traffic Assignment (STA) problem is defined when the system properties, e.g., Origin-Destination (OD) matrix and the link flows, are assumed to be time-independent. On the other hand, if time dependence is considered, the problem becomes a Dynamic Traffic Assignment (DTA). DTA is much more complicated than STA, both computationally and conceptually (Peeta & Ziliaskopoulos, 2001). Here, we first investigate the problem analytically in STA by taking a flow-based approach and then address a real case using a trip-based DTA model.

Much research has been performed on the unicity of STA solutions with several assumptions and limitations on the traffic network model (Netter (1972); Dafermos (1982); Nagurney (1984); Wynter (2001); Wie *et al.* (2002); Florian & Morosan (2014); Sun *et al.* (2014)). For DTA models, the conditions of unicity have been appropriately reviewed by Iryo (2013). He explained that in the large-scale DTA problem, evaluating the solution set is not feasible, and almost all the studies address unicity in small and medium traffic networks by applying analytical approaches. This study takes a different angle, as we are interested in situations where unicity conditions do not hold. The following factors may drive a dynamic traffic network to multiple equilibria:

- The link cost function, i.e., the bottleneck model vs. whole link model (Friesz *et al.* (2001); Lindsey (2004); Silva *et al.* (2016); Osawa *et al.* (2018))
- The transportation mode cost function.
- The transportation mode interaction model (Jiang *et al.*, 2016).
- The symmetry of the network, i.e., loopy network (Iryo, 2011).
- Multi-class users (Marcotte & Wynter (2004); Konishi (2004); Nilsson *et al.* (2018)).

One very classical setting in which multiple equilibria can be reached is the multimodal traffic assignment problem. In a multimodal urban transportation network, users have access to different modes, e.g., car, bus, and metro, which changes their characteristics when they swap between modes (Corman *et al.*, 2017). Therefore, according to the transportation mode cost function and the mode interaction model, the monotonicity condition simply does not hold (Mounce & Smith, 2007). In such cases, the final equilibrium not only depends on the initial state but also on the convergence process. This study focuses on the second aspect.

To investigate the multiple equilibria problem with multimodal settings, we consider a day-to-day convergence process. This considers that a fraction of users may update their

path choice for the next day in the light of the traffic conditions that they experience during the current day. Users swap to a path with minimum travel time as they try to optimize their own gain (Mounce, 2007). When unicity holds, this process converges to the single equilibrium loading (Zhao *et al.*, 2018). Here, we focus on what happens when multi solutions can be reached and on what drives the system to one equilibrium rather than another. More specifically, when considering a long-term day-to-day process, the final network may not be built at once but corresponds to the successive opening of additional facilities. In this study, we consider that the road network is stable over the entire time horizon and that the public transport network is subject to the regular opening of new metro lines. We define this opening process as the network history and investigate how it would affect the final equilibrium state of the network.

First, we perform an analytical investigation and demonstrate the existence of multiple equilibria with respect to the network history of day-to-day STA. Second, we address the same question through simulation for a more complex test case (large-scale network, dynamic traffic assignment, multiple configurations for the network history). The analytical study highlights the causes for the existence of multiple solutions and the influence of network history. The numerical study is performed to investigate in greater depth and show how different the final solution can be. A specific finding is that several network history configurations lead to shorter total travel times for the system than others, and to different mode ratios in the system. This may be of interest when considering public transport planning.

The Static test case and methodology of the STA problem is presented in Section 4.2. The non unicity of equilibrium for bi-modal test case (car-bus) is discussed in Section 4.3. The multiple equilibria in multimodal STA problem is discussed in Section 4.4. Section 4.5 presents the simulation-based DTA model, the multimodal large-scale network and the learning process of day-to-day equilibration. The numerical results are presented and analyzed in Section 4.6 and finally we conclude this chapter in Section 4.7.

4.2 Multimodal STA test case

In this section, we perform an analytical exploration of the non-uniqueness of the network UE solution in a day-to-day multimodal framework. Netter (1972) was the first to study multiple equilibria for a bi-modal (car and bus) network loading for a two-arc network with linear cost functions. Wynter (2001) extended the study by designing a numerical example with polynomial cost functions on the same graph. Here, we focus on the impact of network design history on the final network state. The general idea is that the day-to-day process represents the way in which the users learn about the network state. Each day, a fraction of users adjust their paths to account for the travel times experienced the previous day. The system is known to converge the solution to a stable solution under this process, but the solution may differ depending on the initial state and the learning process. Introducing the possible options successively and in different orders results in the same final network design but through different learning paths.

Let us consider a network with two nodes ($N = \{O, D\}$) and three directed paths from O to D ($A = \{1, 2, 3\}$). There are three modes of transportation ($M = \{C, B, T\}$) which are referred to as car, bus and train (metro) (Figure 4.1). There are two bus lines between origin

and destination. Paths 1 and 2 are shared between car and bus and path 3 is the metro line.

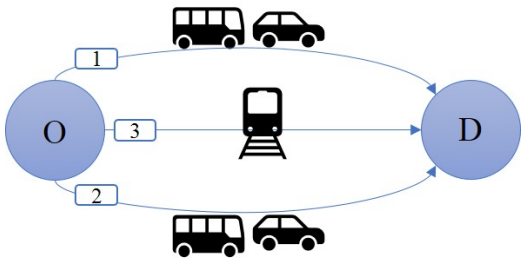


FIGURE 4.1 – A network with a single OD pair, three paths and three transportation modes.

The notations are:

a: Index of path, $a \in A$.
m: Index of mode, $m \in M$.
M_{MV}: The set of motor vehicles (car and bus), $M_{MV} = \{C, B\}$
D^m: Demand of mode m.
c^m: Minimum cost for mode m.
c_a^m: Cost of path a for mode m.
f_a^m: Flow of mode m on path a.

According to the definition, the total flow of each mode is:

$$D^m = \sum_a f_a^m; \quad \forall f_a^m \geq 0, m \in M_{MV} \tag{4.1}$$

$$D^T = f_3^T \tag{4.2}$$

The demand for public transportation, D_{PT} is computed as below:

$$D_{PT} = \sum_m D^m, \quad \forall m = B, T \tag{4.3}$$

The cost for path a and mode m not only depends on the path flow for this mode but also the path flow of other modes as all vehicles interact.

$$c_a^m = g_a^m + \sum_{\mu \in M_{MV}} h_a^{m,\mu} f_a^\mu, \quad \forall m \in M_{MV}, a \in A \tag{4.4}$$

$$c_3^T = c_3 + \frac{\alpha \quad Cap_T}{D^T + Cap_T} \tag{4.5}$$

The c_a^m is defined as a linear function where g_a^m is the free flow cost of mode m on path a and $h_a^{m,\mu}$ is the impact factor of f_a^μ on the cost of mode m. c_3^T denotes the cost of the metro (train) which is independent of the motor vehicles and depends on the capacity of a train (Cap_T). Let Cap_{total} denotes the total capacity of the train line. We assume that $Cap_{\text{total}} \geq D_{PT}$. In general, for long term equilibrium, the capacity of the train line would be adapted to D^T and D_{PT}: with a higher demand the frequency would be increased and the travel cost diminished. Note that c_3^T is a decreasing function of the frequency of trains. In static case, C_3 and α are constant values.

According to the Wardrop equilibrium definition, the network is at the UE flow distribution if and only if for every path a:

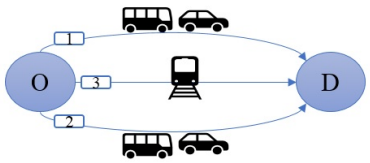
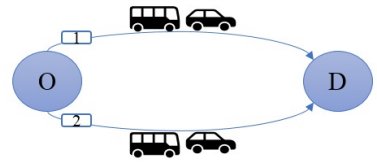
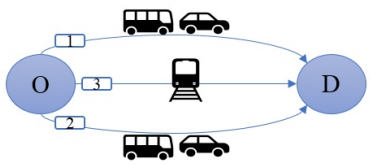
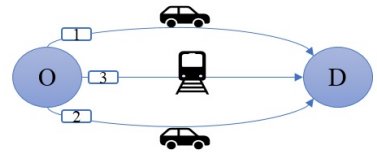
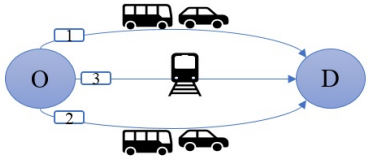
$$f_a^m(c_a^m - c^m) = 0, \quad \forall m \in M, a \in A \tag{4.6}$$

Therefore, when the equilibrium is reached, the cost of all used paths of mode m is equal to c^m. The costs are assumed to be asymmetric: the effect of cars on buses is not the same as the effect of buses on cars. In other words, in the network (Figure 4.1) for one or more modes $m \neq \mu$:

$$\frac{\partial c_a^m}{\partial f_a^\mu} \neq \frac{\partial c_a^\mu}{\partial f_a^m} \quad \forall a = 1, 2, \ \forall m, \mu = B, C \tag{4.7}$$

Here, we are interested in investigating the equilibrium situation where intermediate changes in the network design occur. In other words, the final network is always the same, but this may result from different intermediate steps, see Table 4.1. Scenario 1 is when all modes were active from the beginning. Scenario 2 has no metro line at first place. Thus, an intermediate equilibrium state (UE) is first achieved through the day-to-day learning process before the metro line is added. Then, the metro line is added, and the second convergence process proceeds. Scenario 3 assumes that there are only cars and trains, and no buses during the first convergence period. Then buses are added and a second convergence process is initiated starting from the equilibrium obtained by the first process. We calculate the final network equilibrium for all the scenarios in Table 4.1.

TABLE 4.1 – The scenarios of network design for the mono-OD test case

Scenario	Initial equilibrium state	Final state
1		
2		
3		

4.3 Bi-modal equilibrium analysis: the car-bus case

In order to simplify the presentation of equilibrium analysis, we first explore the non-uniqueness of the network without train ($D^T = 0$), i.e., we analyze the equilibrium solution(s) for the

initial equilibrium state of scenario 2 in Table 4.1. The demand data are D_{PT} and D^C. Let us express costs on paths 1 and 2 as functions of the flows f_1^m, for $m \in M_{MV}$). Therefore, the demand and cost functions are as follows:

$$D^B = D_{PT} \tag{4.8}$$

$$f_2^m = D^m - f_1^m, \quad \forall m \in M_{MV} \tag{4.9}$$

$$c_1^m = g_1^m + \sum_{\mu \in M_{MV}} h_1^{m,\mu} f_1^\mu, \quad \forall m \in M_{MV} \tag{4.10}$$

$$c_2^m = g_2^m + \sum_{\mu \in M_{MV}} h_2^{m,\mu} D^\mu - \sum_{\mu \in M_{MV}} h_2^{m,\mu} f_1^\mu, \quad \forall m \in M_{MV} \tag{4.11}$$

To avoid the repetition, we define $\bar{g}_2^m$:

$$\bar{g}_2^m = g_2^m + \sum_{\mu \in M_{MV}} h_2^{m,\mu} D^\mu, \quad \forall m \in M_{MV} \tag{4.12}$$

The first assumption is that the Jacobian matrix ($\nabla c(f)$) of the car and the bus cost functions is not symmetric positive definite ($c(f)$ is not the gradient of a convex function). Thus, the cost functions are not monotonic. The data for the bi-modal equilibrium are D^C and D^B. The equilibrium condition for the network with two modes has two possibilities for each path: the path is used in the equilibrium path flow distribution or not. Here, we discuss the possibilities for path 1. Figure 4.2 presents the cost-flow diagram of the network based on path 1 and on the independent flows f_1^m (using Equations 4.9 and 4.11).

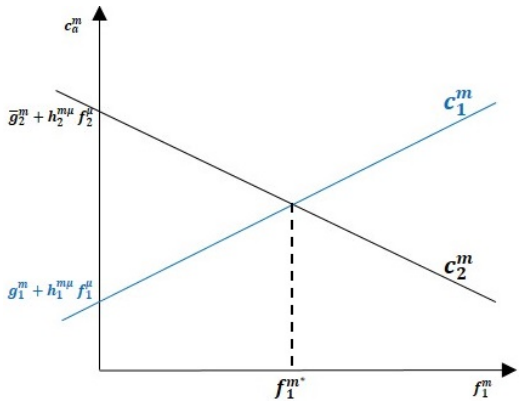


FIGURE 4.2 – Cost-flow diagram based on path 1

If $c_1^m < c_2^m$, we need to increase f_1^m, thus we can show in the plane (f_1^C,f_1^B) the natural direction of variation for flows is $\Delta c^m = (c_2^m - c_1^m)_{m=1,2}$. According to Figure 4.2 and Equation 4.9 (when there is no train), we have to draw $\Delta c^m = c_2^m - c_1^m = 0$ based on the flow value of f_a^m to show the equilibrium point. The flows f_2^m can be considered dependent variables and by Equation 4.11 we can draw the curves $\Delta c^m = 0$ of the two vehicular modes on the flow diagram of path 1 in the (f_1^C,f_1^B) plane. The configuration of the diagram depends on the value of $h_a^{m,\mu}$. In order to draw the two linear flow diagrams, we need at least three

points if they have an intersection. The conditions are extracted from the possible common point:

$$\Delta c^m = c_2^m - c_1^m = \bar{g}_2^m - g_1^m - \sum_{\mu=1,2} (h_1^{m,\mu} + h_2^{m,\mu}) f_1^\mu = 0 \tag{4.13}$$

According to the Equation 4.13, we can extract the slope of Δc^m diagram. The intersect with $f_1^B = 0$ is:

$$F_1^C(m) = (\bar{g}_2^m - g_1^m)/(h_1^{m,C} + h_2^{m,C}) \tag{4.14}$$

and the intersect with $f_1^C = 0$:

$$F_1^B(m) = (\bar{g}_2^m - g_1^m)/(h_1^{m,B} + h_2^{m,B}) \tag{4.15}$$

Therefore, the slope of $\Delta c^m = 0$ line for mode m on path 1 is as follows:

$$\frac{1}{(h_1^{m,B} + h_2^{m,B})} / \frac{1}{(h_1^{m,C} + h_2^{m,C})} = \rho_m = \frac{(h_1^{m,C} + h_2^{m,C})}{(h_1^{m,B} + h_2^{m,B})} \tag{4.16}$$

Thus, with ρ_m we can conditionally draw the flow diagram of $\Delta c^m = 0$ for two modes on path 1. There are three possibilities: both modes are used, only cars are used, only buses are used. In the case where the two modes are used, i.e., $f_1^m \in]0, D^m[$, we obtain:

$$-g_1^m + \bar{g}_2^m = \sum_{\mu=1,2} (h_1^{m,\mu} + h_2^{m,\mu}) f_1^\mu \quad \forall m \in M_{MV} \tag{4.17}$$

The diagram configuration depends on the values of ρ_m. We need to know the relation of the flow diagram's slope for mode B (ρ_B) in relation to the slope of the line Δc^C (ρ_C). In order to consider a day-to-day process in static case, we define the network equilibrium problem as a projected dynamical system. Nagurney & Zhang (2012) prove that projected dynamical systems find the equilibrium point(s) by producing the solution trajectory (mapping function) based on a fixed point theory.

The projected dynamical system of traffic assignment

Let K be the rectangle $[0, D^C] \times [0, D^B]$ of admissible flows based on the constraints 4.1. Let $\Delta c : \mathbb{R}^2 \mapsto \mathbb{R}^2$ be the vector field given by $(\Delta c^C, \Delta c^B)$ and let $f = (f_1^C, f_1^B)$ be the vector of independent path flows. Let $\Pi_K(f, .)$ denote the projector on the tangent cone of K at f. The projected dynamical system PDS$(f, \Delta c(f))$ is defined by:

$$\dot{f} = \Pi_K(f, \Delta c(f)) \tag{4.18}$$

Let $\Gamma^K(f) \overset{def}{=} \Pi_K(f, \Delta c(f))$ be the projection of the field $\Delta c(f))$ on K. The point f^* is an equilibrium point (i.e) satisfies Equations 4.1, 4.4, and 4.6, if and only if $\Gamma^K(f^*)$ is equal to zero, which means that f^* is a fixed point of the projected dynamical system PDS$(f, \Delta c(f))$. Thus the field lines (trajectories) of Γ^K in the solution space can be used to describe the day-to-day STA learning process of travellers.

Figure 4.3 presents the flow diagram of path 1 depending on the respective values of ρ_B and ρ_C. The arrows are cardinal field directions in Figure 4.3 and indicate how the solution moves toward the equilibrium.

In each region of the diagram, the network state is not at equilibrium, so there are two possibilities for Δc^m. If $\Delta c^C > 0$, which means that the cost of the car on path 1 is lower than that on path 2, so a part of the flow of path 2 will be shifted to path 1, thus on the flow diagram of path 1, f_1^C increases and the network state moves to the right. In a similar manner, when $\Delta c^C < 0$, f_1^C decreases and the solution moves to the left. For Δc^B, if the flow of bus lines shifts from one path to another, according to Equations 4.8 and 4.12, lines $f_1^B = D^B$ and $\Delta c^B = 0$ are moved. Therefore, if $\Delta c^B > 0$, as with the car, a part of f_2^B is swapped to path 1, which means f_1^B increases and the network state moves up. Finally, when $\Delta c^B < 0$, f_1^B decreases and the solution moves down. Consequently, the day-to-day STA (PDS$(f, \Delta c(f))$) changes the state of the network until the UE is reached.

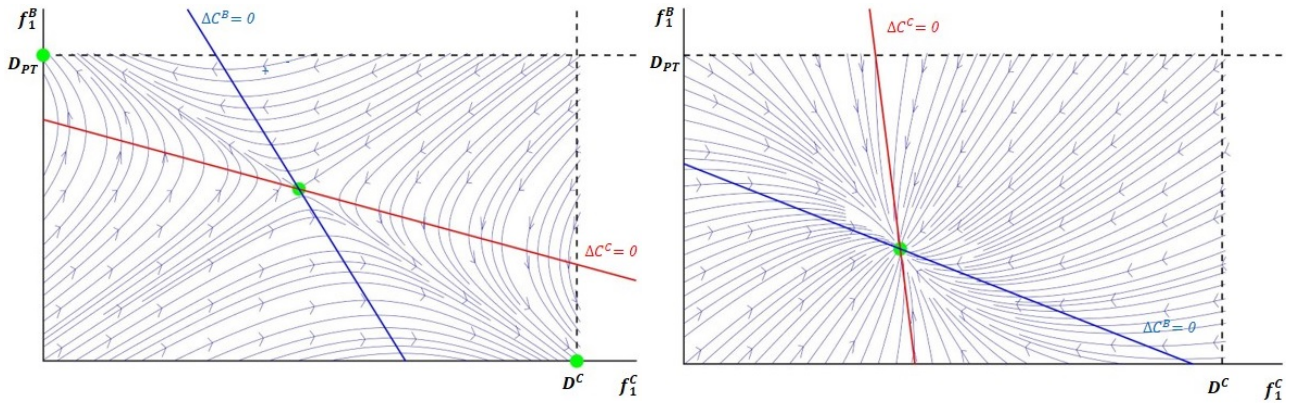
By definition, the network is at UE state where the result of the arrows is zero at the current point, which corresponds to a fixed point of the PDS$(f, \Delta c(f))$. This refers to the following conditions for the equilibrium solution based on Δc^m:

- Either If $\Delta c^m = 0; \forall m$

- Or If the result of arrows is zero and the other arrows direct the solution to the non-feasible region (violate Equation 4.1) $\Rightarrow \Delta c^m$ is zero for mode m and $f_1^\mu = f_2^\mu = 0$ and $c_1^\mu, c_2^\mu > c_1^m = c_2^m$ for mode μ where $m \neq \mu$ and we reach the equilibrium. This equilibrium is located on the axes or demand borders ($f_1^B = D^B$ and $f_1^C = D^C$)

If $\rho_B > \rho_C$ the flow diagram for this configuration is presented in Figure 4.3(a). Based on the initial point, we have three possible equilibria.

Note that the definition of the equilibrium point is different from the stationary point in this context. The stationary point is the stable equilibrium in our test case. The stability of the solution can be analyzed based on the PDS$(f, \Delta c(f))$. The unstable equilibrium may exist in the solution space, where the $\Delta c^m = 0$, but the PDS$(f, \Delta c(f))$ of neighbor points lead the system to another equilibrium. In other words, the UE solution E is stable only if, when it moves by step size ϵ; $\epsilon \to 0$ in any direction in the feasible region, the day-to-day process converges the solution to E. By this definition, the intersection of the two diagrams is an unstable equilibrium, and the two other equilibria are stable.



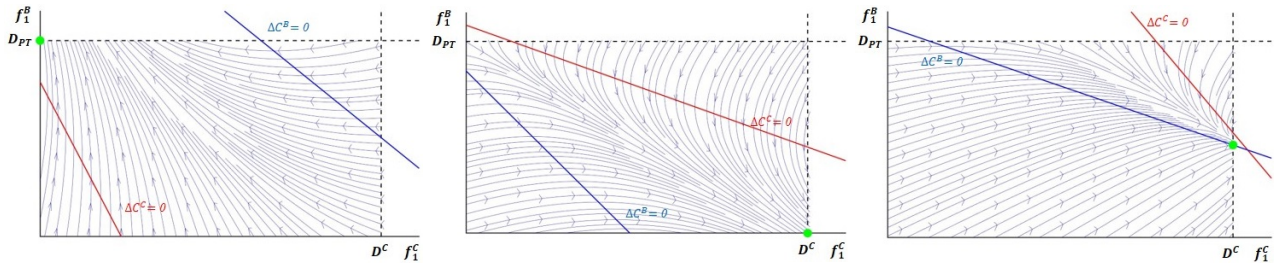
(a) $\rho_B > \rho_C$ (b) $\rho_B < \rho_C$

FIGURE 4.3 – The equilibrium solution(s) on the flow diagram of path 1

Figure 4.3(b) presents the flow diagram where $\rho_B < \rho_C$. This configuration leads to a unique stable solution at the intersection of two flow diagrams. Consequently, in the first configuration, we will have a conditional equilibrium based on the values of ρ_C, ρ_B.

$$\text{Solution set} = \begin{cases} \rho_B > \rho_C & \text{3 equilibria, 1 unstable and 2 stable,} \\ \rho_C > \rho_B & \text{1 equilibrium, stable.} \end{cases} \tag{4.19}$$

If we consider other configurations for $\rho_B > \rho_C$ where the two diagrams $\Delta c^C = 0$ and $\Delta c^B = 0$ do not intersect inside K, we will have a unique equilibrium. For instance, if the flow line for car ($m = C$) lies below the bus line ($m = B$), then there is an equilibrium which is stable with $f_1^B = D^B$ and $f_1^B = 0$. This configuration is shown in Figure 4.4(a). If the relation between the flow of two modes on Path 1 is inverse ($f_1^B = 0$), we will have a stable equilibrium according to Figure 4.4(b). If we look at the configuration where two diagrams intersect (as in Figures 4.3(a)), we do not necessarily obtain multiple solutions. If the intersection is outside the feasible solution, we also obtain a unique solution. Figure 4.4(c) presents an example when we obtain a unique solution, and two diagrams intersect outside the feasible region (K).



(a) $F_1^B = 0$ (b) $F_1^C = 0$ (c) $F_1^C = D^C$ and $F_2^C = 0$

FIGURE 4.4 – The flow diagram of special cases for $\rho_B > \rho_C$ (The intersection is outside K)

Thus, in order to obtain multiple solutions, it is required that the lines $\Delta c^m = 0$ intersect inside K and that their slopes are such that the intersection point is unstable for the PDS$(f, \Delta c(f))$. The resulting conditions are:

$$\begin{cases} \rho_B > \rho_C \\ (\Delta c^C = 0) \cap (\Delta c^B = 0) \in K \end{cases} \tag{4.20}$$

where $(\Delta c^C = 0) \cap (\Delta c^B = 0)$ is obtained by solving $\Delta c^B = \Delta c^C = 0$ and depends on the value of D^C and D^B.

We proved that even with two modes car and bus, we have multiple equilibria. The final equilibrium is defined by the initial network state before the learning process starts. For example, if at the initial state, the demand for cars is split into both paths whereas Public transportation (PT) users only take the metro, then equilibrium will be reached at the bottom right corner in Figure 4.3(a). If we have the same assignment for cars whereas PT users take buses, then the equilibrium will be converged to the point at the top left corner in Figure 4.3(a). Now, we can calculate the final network equilibria for all the scenarios in Table 4.1.

4.4 Non unicity of equilibrium states in multimodal STA

4.4.1 Equilibrium analysis for Scenario 1: car-bus-train case

In the first scenario, all modes are active, and the initial state of the network can be any path flow distribution with respect to constraints 4.1- 4.3, and 4.21.

$$D_{PT} = D^B + D^T \tag{4.21}$$

where D^B depends on D^T, which is determined by solving $c_3^T = c^B$. It should be remembered that we consider the train as an independent travel mode, which has an impact on the demand for the bus. In this case, the flow vector of independent flows ($\dot{f}$), the feasible domain K for the independent flow variables, and the field variables (Δc) are as follows:

$$\dot{f} = (f_1^C, f_1^B, D^T) \tag{4.22}$$

$$\begin{cases} 0 \leq f_1^C \leq D^C \\ 0 \leq f_1^B + D^T \leq D_{PT} \end{cases} \tag{4.23}$$

$$\Delta c = \begin{cases} \Delta c^C = c_2^C - c_1^C \\ \Delta c^B = c_2^B - c_1^B \\ \Delta c^T = c^B - c_3^T \end{cases} \tag{4.24}$$

Therefore, $K = [0, D^C] \times [0, D_{PT}] \times [0, D_{PT}]$ is shown in Figure 4.5. The PDS$(f, \Delta c(f))$ is defined by Equation 4.18 with $\Delta c(f)$ is given by Equation 4.24.

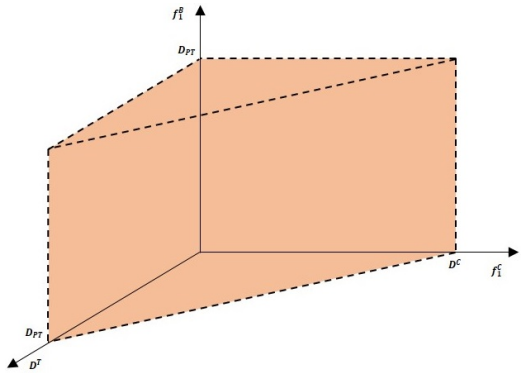


FIGURE 4.5 – The feasible solution space (K) when all modes are active

Same as the previous section, the network equilibria are the fixed points of PDS$(f, \Delta c(f))$ in K. In the sequel, we consider c_3^T a constant. The initial condition to have multiple equilibria is $c_3^T \not< c_a^B; \forall f_a^m$, which means that the cost of the train is not always less than the cost of the bus lines. Otherwise, only cars and trains will be used, where both modes are independent, and consequently the equilibrium is unique. Moreover, if we have $c_a^B \not< c_3^T; \forall f_a^m$, which means that C_3^T is comparable with the cost of the bus, so at the equilibrium state a part of D_{PT} takes the train. Otherwise, the train will not be used ($D^T = 0$), and we could

have multiple equilibria based on the analysis in Section 4.3. Consequently, for scenario 1, we will have conditional equilibrium based on the values ρ_C, ρ_B, and c_3^T.

$$\text{Solution set} = \begin{cases} c_3^T < c_a^B; \forall a, f_a^m \rightarrow 1 \text{ equilibrium with } D^T = D_{PT}, \\ o.w. \rightarrow \begin{cases} \rho_B > \rho_C \rightarrow \begin{cases} (\Delta c^C = 0) \cap (\Delta c^B = 0) \in K \rightarrow 3 \text{ equilibria}, \\ o.w. \rightarrow 1 \text{ equilibrium}, \end{cases} \\ \rho_B < \rho_C \rightarrow 1 \text{ equilibrium}. \end{cases} \end{cases} \tag{4.25}$$

In the case of multiple equilibria, based on the initial path flow distribution, we can converge to three possible equilibria based on the day-to-day process. Consequently, similar to bi-modal equilibria, the initial state of the network, determines the final equilibrium in Scenario 1.

4.4.2 Equilibrium analysis for Scenario 2: (car-bus)-train case

In Scenario 2, We have $D^T = 0$ at the initial equilibrium state when only car and bus lines are active. Thus $D_{PT} = D^B$. The equilibrium is reached in the network, excluding the metro line. This problem is already discussed in Section 4.3. Here, we want to investigate the situation where we have multiple equilibria, so let us assume that we have multiple solutions at the equilibrium state by holding conditions 4.20 and $c_3^T \not< c_a^B; \forall f_a^m$ where all modes are active, and Equation 4.21 holds. Figure 4.6 presents the nine possibilities for multiple equilibria according to the slope of two figures. These possibilities are the starting point for scenario 2, where the metro line is added, and the convergence process starts. In Figure 4.6, the unstable equilibrium is the intersection of two lines and presented by a point (I) and the two stable equilibria are points (P) and (Q).

The general analysis for all cases in Figure 4.6 is complicated. There are few studies that provide only numerical examples to show multiple equilibria in a multimodal context. Netter (1972), Wynter (2001) and Marcotte & Wynter (2004) presented numerical examples which correspond to the case in Figure 4.6(a). Here, we consider Figure 4.6(b) to analyze the equilibria solutions.

4.4.2.1 car-bus equilibria in case of Figure 4.6(b)

Before adding the train to the system, we need to analyze two stable equilibria. For point (P), we have the following conditions:

$$\begin{cases} D^T = 0 \\ f_1^C = 0 \\ f_2^C = D^C \\ f_1^B = D_{PT} \\ f_2^B = 0 \end{cases} \tag{4.26}$$

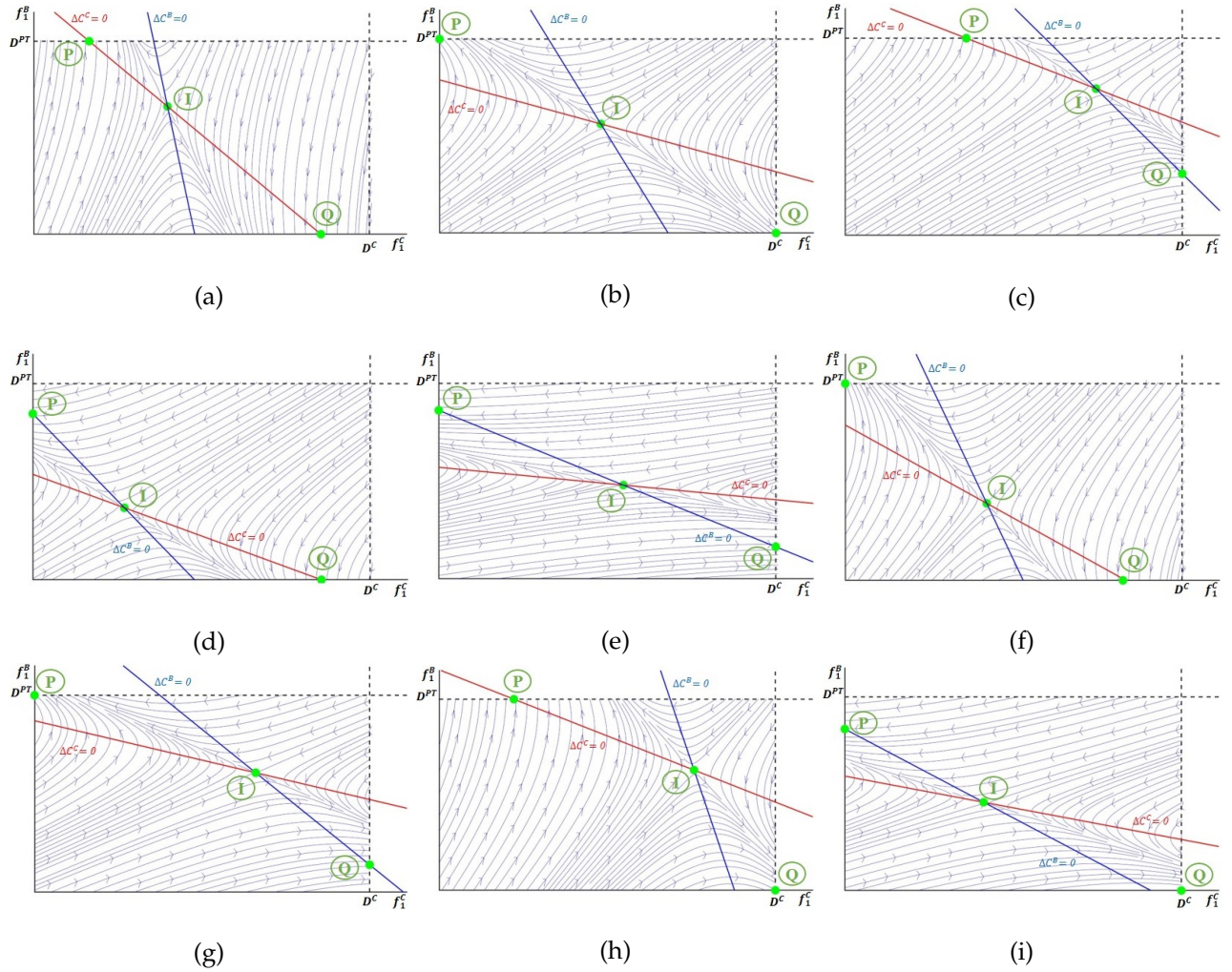


FIGURE 4.6 – Initial equilibrium state for Scenario 2: The flow diagram of path 1 when $\rho_B > \rho_C$

Therefore, the OD cost at point (P) is as follows:

$$\begin{cases} S^C(P) = c_2^C = g_2^C + h_2^{C,C} D^C \leq g_1^C + h_1^{C,B} D_{PT} \\ S^B(P) = c_1^B = g_1^B + h_1^{B,B} D_{PT} \leq g_2^B + h_2^{B,C} D^C \end{cases} \tag{4.27}$$

where, $S^m(x)$ is the OD cost for mode m at point x. Consequently, the feasibility conditions for the parameters are:

$$\begin{cases} g_1^C - g_2^C \geq h_2^{C,C} D^C - h_1^{C,B} D_{PT} \\ g_1^B - g_2^B \leq h_2^{B,C} D^C - h_1^{B,B} D_{PT} \end{cases} \tag{4.28}$$

In the same manner, we will have the following equations for point (Q):

$$\begin{cases} D^T = 0 \\ f_1^C = D^C \\ f_2^C = 0 \\ f_1^B = 0 \\ f_2^B = D_{PT} \end{cases} \tag{4.29}$$

$$\begin{cases} S^C(Q) = c_1^C = g_1^C + h_1^{C,C} D^C \leq g_2^C + h_2^{C,B} D_{PT} \\ S^B(Q) = c_2^B = g_2^B + h_2^{B,B} D_{PT} \leq g_1^B + h_1^{B,C} D^C \end{cases} \tag{4.30}$$

$$\begin{cases} g_1^C - g_2^C \leq h_2^{C,B} D_{PT} - h_1^{C,C} D^C \\ g_1^B - g_2^B \geq h_2^{B,B} D_{PT} - h_1^{B,C} D^C \end{cases} \tag{4.31}$$

Note that Equations 4.28 and 4.31 are equivalent to the fact that (i) the intersects of $\Delta c^m = 0$ with $f_1^B = 0$ are not less than D^c if m is car and not bus; (ii) the intersects of $\Delta c^m = 0$ with $f_1^C = 0$ are not less than D^B if m is bus and not car. By considering $H^{m,\mu} = h_1^{m,\mu} + h_2^{m,\mu}$, we can summarize the conditions for the parameters in Equations 4.28 and 4.31:

$$\begin{cases} H^{C,C} D^C < H^{C,B} D_{PT} \\ H^{B,B} D_{PT} < H^{B,C} D^C \end{cases} \tag{4.32}$$

From Equations 4.27 and 4.30, we deduce a conditions for the data parameters (D^C and $D_{PT} = D^B$) to yield the situation of Figure 4.6(b). For instance, if $g_1^m = g_2^m$, then D^C and D^B should satisfy the following condition:

$$max \left\{ \frac{h_1^{B,B}}{h_2^{B,C}}, \frac{h_2^{B,B}}{h_1^{B,C}} \right\} \leq \frac{D^C}{D^B} \leq min \left\{ \frac{h_2^{C,B}}{h_1^{C,C}}, \frac{h_2^{C,B}}{h_2^{C,C}} \right\} \tag{4.33}$$

The Equations 4.27, 4.30 and 4.32 provide the conditions for the parameters to generate a Figure 4.6(b) type of situation. In the same way, we can calculate the parameter domains and equilibria set for other cases in Figure 4.6.

4.4.2.2 Adding train to the system

Here, we consider the three equilibria ($\textcircled{P}$, $\textcircled{Q}$, and $\textcircled{I}$) as the possible intermediate states, i.e., the starting point for the convergence process of scenario 2. Now, we add the train to the system. If $c_3^T > c_a^B; \forall a \in A$, then the train is not used and does not change the system, also, if $c_3^T < c_a^B \forall a \in A$, then the users stop using the bus and we will have two independent modes (car and train) and the given demand which converge the system to a unique equilibrium. The challenge is when the cost of the train is comparable to the cost of the bus, and the system moves to the new equilibrium.

For Δc^B, if the flow of bus lines shifts to the metro line or the reverse, according to Equations 4.12 and 4.21, lines $f_1^B = D^B$ and $\Delta c^B = 0$ are moved. Therefore, if $\Delta c^B > 0$ there is a possibility that a part of f_2^B is swapped with the metro line and the network state point on the flow diagram does not move. Otherwise, as with the car, f_1^B increases and the network state moves up. Finally, when $\Delta c^B < 0$, f_1^B decreases and the solution moves down.

Indeed, when the train enters the system, two lines move in all cases in Figure 4.6:

- Line $f_1^B = D_{PT}$: It changes to $f_1^B = D^B = D_{PT} - D^T$ and starts decreasing. If one user swaps from this pattern to the train, then the border is shifted down by 1 unit.
- Line $\Delta C_1^B = 0$: It moves to left and downwards because the flow of the bus shifts to the train. The translation of this line if one user swaps to the train is $\frac{h^{m,B}}{H^{m,B}}$ according to the Equation 4.20.

Note that line $\Delta C_1^C = 0$ can move in some cases but the line translation directly depends on the translation of $\Delta C_1^B = 0$. The translation size of line $\Delta C_1^B = 0$ is smaller than line $f_1^B = D_{PT}$ ($\frac{h^{m,B}}{H^{m,B}} \leq 1$). Therefore, the line $f_1^B = D_{PT}$ can pass $\Delta C_1^B = 0$ or $\Delta C_1^C = 0$ and point (I) in case of Figures 4.6(b), 4.6(d) and 4.6(e). The analysis of the final equilibrium based on the intermediate state is as follows:

1. The intermediate state is point (I): According to the movement of the line, $m = B$ to the left and downwards, the system will converge to point (Q) in all cases in Equation 4.20.
2. The intermediate state is point (Q): Any movement of the equilibrium will place the starting point (intermediate state) in the region where the PDS($\Delta c(f), f$) pushes the system to the updated point (Q). Thus, the system converges to the unique equilibrium.
3. The intermediate state is point (P): As with the previous case, any movement of the equilibrium places the intermediate state in the region that converges to the updated point (P).

We can calculate the new equilibrium P or Q by considering the initial equilibrium. For instance, at (P) if the initial state is $\bar{P}$, then we have:

$$S^B(P) = g_1^B + h_1^{B,B}(D_{PT} - D^T) = C_3^T = S^B(\bar{P}) - \epsilon = S^B(\bar{P}) - h_1^{B,B} D^T \tag{4.34}$$

where ϵ is the difference between C_3^T and the cost of bus at $\bar{P}$. Therefore, $D^T = \frac{\epsilon}{h_1^{B,B}}$ and based on Equation 4.21, we have:

$$D^B(\epsilon) = D_{PT} - \frac{\epsilon}{h_1^{B,B}} \tag{4.35}$$

To complete this example, if Equation 4.33 is satisfied, we obtain a stable equilibrium at point (P):

$$\begin{cases} f_1^C = 0 \\ f_2^C = D^C \\ f_1^B = D^B(\epsilon) \\ f_2^B = 0 \end{cases} \tag{4.36}$$

Consequently, for the second scenario, we could have two different stable equilibria by adding a new mode even if the infrastructure is independent. It should be noted that not all situations are reachable at the end; secondly, that the final equilibrium state of Scenario 2 depends on the initial state which is determined by bi-modal equilibrium (Section 4.3), i.e., the network design history defines which states are the real possible final states.

4.4.3 Equilibrium analysis for Scenario 3: (car-train)-bus case

4.4.3.1 Car-train equilibrium analysis

For the third scenario, the intermediate state is the equilibrium state of the system when just car and train are active ($f_m^B = 0 \; ; \forall m$). Therefore, We have two independent modes with

fixed demand. In this case, the equilibrium is unique because $D^T = D_{PT}$, i.e., the totality of public transportation demand is assigned to train, and there are two independent linear strictly monotone functions for cars.

$$\begin{cases} c_1^C = g_1^C + h_1^{C,C} f_1^C \\ c_2^C = g_2^C + h_2^{C,C} f_2^C \end{cases} \tag{4.37}$$

The demand D^C is spread between paths 1 and 2. Finally, we have one of the following possibilities as a function of $g_a^C, h_a^{C,C}; \forall a$:

- Only path 1 is used: $f_1^C = D^C \quad ; \quad f_2^C = 0 \quad ; \quad C_1^C \leq C_2^C$
- Only path 2 is used: $f_1^C = 0 \quad ; \quad f_2^C = D^C \quad ; \quad C_1^C \geq C_2^C$
- Both path are used: $f_1^C, f_2^C \geq 0 \quad | \quad C_1^C = C_2^C \quad ; \quad f_1^C + f_2^C = D^C$

A unique starting point for scenario 3 is located on line $f_1^B = 0$ (f_1^c axis) in the flow diagram of path 1 (Figure 4.6) at the intersect of $\Delta c^C = 0$ with $f_1^B = 0$. Finally, the unique equilibrium for car and train network on the path flow diagram of path 1 is as follows:

$$\begin{cases} D^T = D_{PT} \\ f_1^C = \frac{g_2^C - g_1^C + h_2^{C,C} D^C}{H^{C,C}} \\ f_1^B = 0 \end{cases} \tag{4.38}$$

4.4.3.2 Adding bus to the system

When the bus lines are added to the system, the initial situation is the car-train equilibrium 4.38 and the cost for bus lines are:

$$\begin{cases} c_1^B = g_1^B + h_1^{B,C} f_1^C \\ c_2^B = g_2^B + h_2^{B,C} (D^C - f_1^C) \end{cases} \tag{4.39}$$

where f_1^C is given by Equation 4.38. The minimum OD cost for bus at the car-train equilibrium point is:

$$S^B(\mathrm{C+T}) = min \left\{ c_1^B, c_2^B \right\} \tag{4.40}$$

If the $c_3^T \leq S^B(\mathrm{C+T})$, public transportation travellers have no incentive to switch mode to bus. From a mathematical point of view, Δc^m remains at zero and the PDS$(f, \Delta c(f))$ cannot move the point. If only path 1 or both paths are used, the convergence process starts in the region where the intermediate solution is located pushes the system to point (Q). If only path 2 is used ($f_1^C = 0$), the final equilibrium depends on the mode cost functions. In this case, one of two stable equilibria (points (P) and (Q)) can be reached in Figure 4.6. Therefore, we will have a unique equilibrium. If multiple equilibria are possible, the initial car-train equilibrium determines which equilibrium will be reached at the end of the third scenario.

To conclude this section, Figure 4.7 presents the relation between the intermediate state of the network and the final equilibrium with a different order of mode activation. The analysis shows that even in the linear STA if we have a different order for opening the new network facility (Table 4.1) and even with a rational learning process, we converge to different equilibrium.

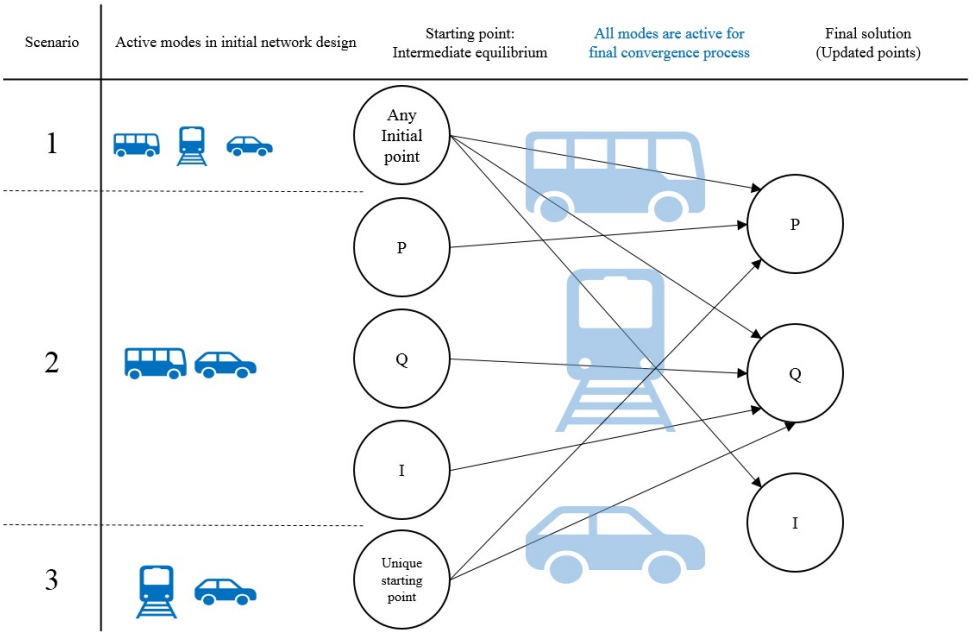


FIGURE 4.7 – The convergence of scenarios for the mono-OD test case

4.5 Multi-modal simulation-based day-to-day DTA

In this section, we address the question of network history and multiple multimodal user equilibria in a more realistic framework. Now, we resort to a dynamic traffic simulator for the network loading and focus on a real network. Although the setting is more complex, mechanisms similar to those described in the previous section apply and induce non unicity of equilibria, and dependence of equilibria on the order of activation of facilities.

4.5.1 Day-to-day network equilibrium model

Consider a network $G(N, A)$ with a finite set of nodes N and a finite set of directed links A. The period of interest (planning horizon) of duration D is D_{max} days indexed by d ($d \in D = \{0, 1, 2, ..., D_{max}\}$). In a day d, travel time and traffic conditions are calculated by simulation and the users choose the path for the next day based on the travel time experienced during the current day. The important notations for introducing the dynamic equilibrium model are as follows:

W: set of OD pairs.
P_w^d: set of paths for w in day d.
w: index of OD pair, $w \in W$.
p: index of path, $p \in P_{w,d}$.

$\pi^d_{w,p}$: number of users from an OD pair w that are assigned to path p in day d.
$C^d_p(\alpha_i, \beta_i)$: Cost of path p in day d.
$C^{d\,*}_w$: minimum Cost of OD pair w in day d.

Note that P^d_w is not necessarily equal to P^{d+1}_w because the network design can be changed, i.e., the new transportation facility is added to the network from the next day $(d+1)$.

As mentioned in previous chapters, in the trip-based DTA problem, the solution space is discrete, so the goal according to UE discipline is to minimize the gap between path travel time and the shortest path travel time of the related OD pair for all OD pairs, which is equivalent to Equation 4.6. In other words, finding the UE situation is equivalent to minimizing the delay of each user compared to the optimal option of the associated OD pair (shortest path) in the network. Using this definition, for each OD pair $w \in W$ and for all paths $p \in P_{w,d}$, the dynamic traffic network equilibrium with a given travel demand and user departure time for the trip-based DTA model is reached on day $d \in D$ if the following conditions are satisfied (Smith, 1993):

$$\begin{cases} C^d_p - C^{d\,*}_w \geq 0 \\ \pi^d_{w,p}(C^d_p - C^{d\,*}_w) = 0 \\ \pi^d_{w,p} \geq 0 \end{cases} \tag{4.41}$$

Based on Equation 4.41, we can define a quality indicator for the solutions which is calculated as the average delay of the network (Janson, 1991) for day d:

$$AG^d = \frac{\sum_{w \in W} \sum_{p \in P^d_w} (C^d_p - C^{d\,*}_w)}{\sum_{w \in W} \sum_{p \in P(w,d)} \pi^d_{w,p}} \tag{4.42}$$

Note that $AG^d = 0$ when the perfect UE path flow distribution is achieved. The aim of the day-to-day process is to minimize the Average Gap of the network by considering the learning curve of the users.

At the end of each day, the users are ranked based on the TC they experience and 50% of users with the highest TC are allowed to swap to the time-dependent shortest path(s). Note that changing the swapping rate would only change the convergence speed, not the final state. Thus, the swap decision is taken by each user based on the Bernoulli trial:

$$P(S^d = 1) = \frac{EC - EC^*_w}{EC} \tag{4.43}$$

where S^d denotes the binary swap decision variable for day d, EC denotes the TC experienced by the user on the current day and EC^*_w denotes the minimum TC experienced of the OD pair w on the current day which is related to the user. Following the result of the trial, the user decides to swap or not. The day-to-day DTA framework is presented in Figure 4.8, and detailed in the following:

1. Read network and demand: The network and demand of the scenario designed are configured.

2. Initialization: The initial paths are assigned to the users based on the scenario of the experiment.
3. Traffic simulation: The simulation is executed for the inputs, and the simulator calculates all variables.
4. All cost functions are calculated by the updated variables from the simulation.
5. Identify the shortest path based on network facilities and access permission of users.
6. Calculate the indicators and check the End condition based on Average Gap (Equation 4.42):
   - **IF** Average Gap does not change, End the process
   - **ELSE** Update the path flow distribution for the next day by day-to-day process (Equation 4.43).

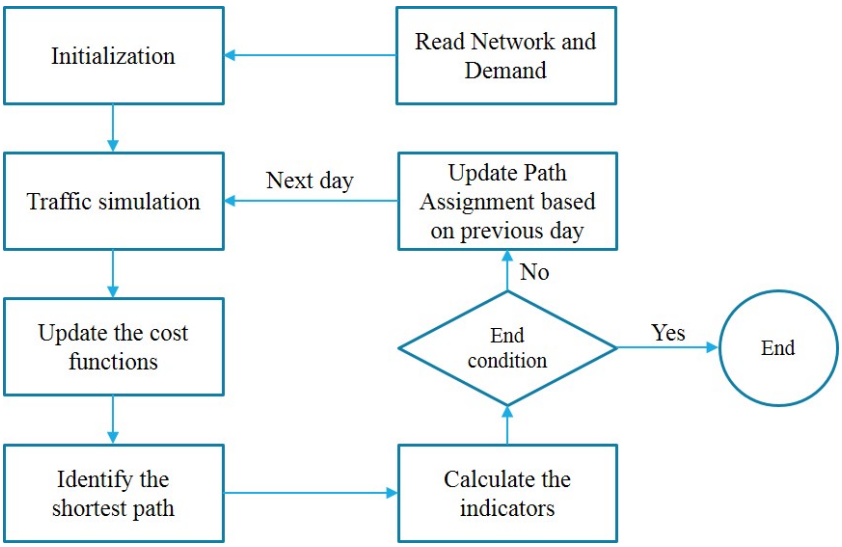
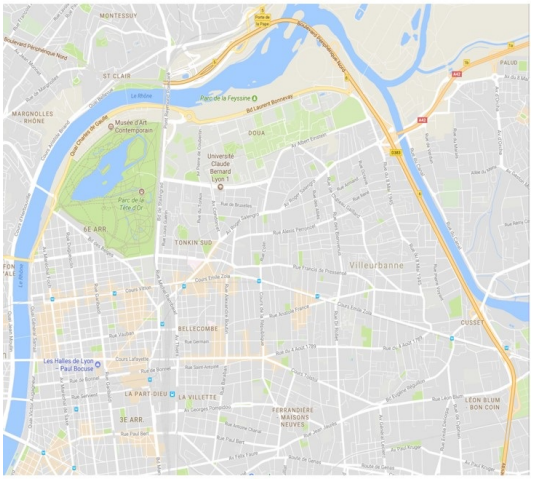


FIGURE 4.8 – The day-to-day framework

4.5.2 Dynamic test case

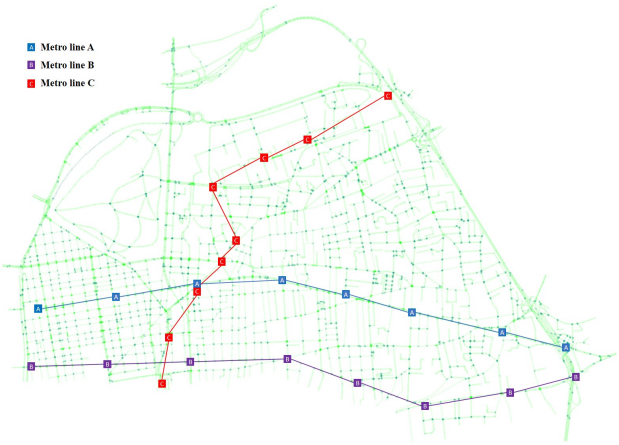
In this work, The Symuvia platform, including the trip-based simulator (Section 1.2.2) and the command module: SymuMaster (Section 1.2.3) is used in order to compare the solution of different scenarios. Using the dynamic simulator permits us to consider the large-scale network of Lyon 6e + Villeurbanne (Figure 4.9(a)) with 1,883 Nodes, 3,383 Links, 94 Origins, 227 Destinations and 54,190 trips. Walking, buses and private cars are initially available transportation modes in the network. Figure 4.9(b) presents 31 bus lines in the Lyon 6e + Villeurbanne network includes 176 bus station (Figure 4.9(c)). There are three metro lines (A, B and C) and 25 metro stations in the network (Figure 4.9(d)). Each metro station has parking facilities. Carparks are the connectors between the metro grid and the traffic network. Therefore, the traveler can start their trip with a private car then use the carpark to



(a) Mapping data ©Google 2019

(b) 31 bus lines

(c) 176 bus station



(d) 3 metro lines

FIGURE 4.9 – Multimodal traffic network of Lyon 6e + Villeurbanne

take the train. All mode changes during a trip introduce walking time for connection and possibly a waiting time for the next bus or metro to arrive at the station.

The network is loaded with travelers of all ODs with a given departure time in order to represent 1.5 hours of the network with the demand level based on the study of (Krug *et al.*, 2019). The goal is to analyze the final equilibrium solution obtained by a day-to-day DTA model with different settings corresponding to different successive introductions of the metro lines.

4.5.3 Experiments scenarios

For each scenario of opening metro lines, we run the day-to-day DTA for 300 days. A quarter of users only have access to the public transportation system (bus and metro) and the other three quarters have access to all transportation facilities (private car, bus, and metro) in the network. Note that bus lines are active for all scenarios. We can open three metro lines at the

same time and calculate the equilibrium or successively open one metro line every 100 days and look for the final network state after 300 days. From the viewpoint of optimization, it means we change the intermediate assignment pattern to find the equilibrium. There are seven possible orders to activate the metro lines (Figure 4.10). All the scenarios are started by the final equilibrium solution of the network without metro lines. The initial assignment pattern of each step is the final equilibrium flow distribution of the previous step (for more details see Appendix A.1). For Scenario 1, all three metro lines are activated at the same time and once the day-to-day process is executed in order to equilibrate the system.

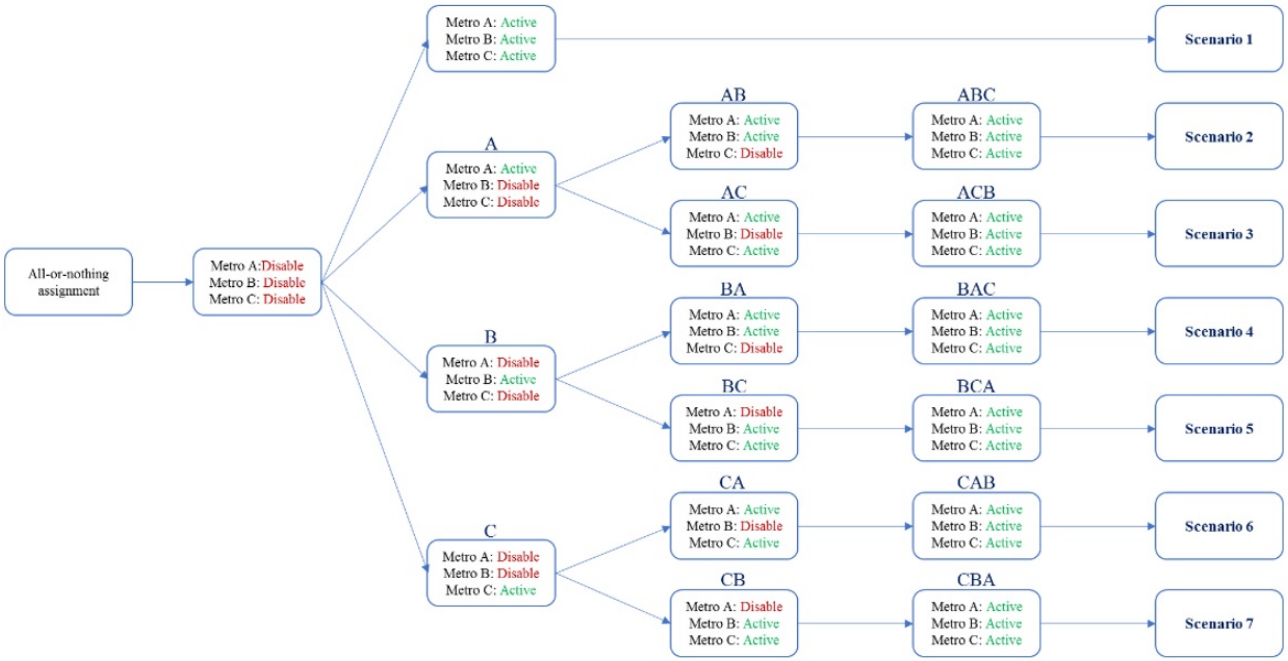


FIGURE 4.10 – Chart of experiments

4.6 Numerical results

The full day-to-day process is conducted for all the scenarios, and we verify that all the simulations converge to a satisfactory UE solution, i.e., the Average Gap of all scenarios is less than 12 seconds, which shows good quality for the equilibrium in the large-scale network and given the demand level. Figure 4.11 (excluding 4.11(h)) presents the convergence pattern of the last step of all the scenarios. For instance, Figure 4.11(b) presents the convergence pattern of the last step of the ABC scenario, which means the simulation starts in the network with all the metro lines, with the equilibrium flow distribution of the AB simulation. The final solution of the AB simulation is obtained by the day-to-day process in the network wherein only metro lines, A and B, are active, and metro line C is deactivated (see Figure 4.10). Mathematically speaking, all the convergence patterns correspond to the same final network design, including three metro lines with different intermediate steps.

Scenario 1 (A&B&C) starts with the equilibrium solution of the network without metro lines, and three metro lines become active at the same time for the last convergence process. This explains why the initial Average Gap of this scenario is much larger than others and

the variation of the Average Gap is larger than in the other figures because the three new public transportation modes become active at the same time. It should be remembered that the day-to-day process stops when the Average Gap does not change for two consecutive days, which occurs for all the scenarios before day 46. Next, we evaluate the final solution in order to investigate the unicity of the solution.

We calculate the Violation indicator for the final path flow distribution of each scenario. the Violation indicator was introduced in Section 2.3.2. Figure 4.11(g) presents the network Violation of the final solution of each scenario. The results show that the quality of each solution is slightly different.

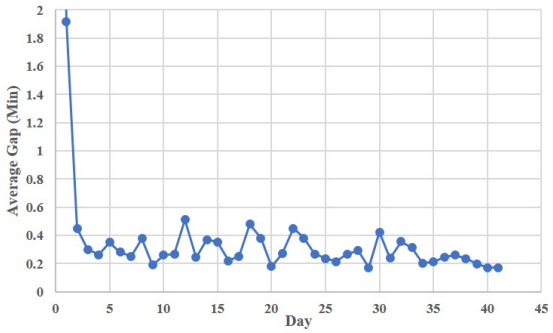
To evaluate the travel time distribution of OD pairs, we calculate the mean travel time and the percentage of failed trips for five most crowded OD pairs (highest demand level). A user fails the trip if they cannot arrive at their destination before the end of the simulation period. The results are shown in Table 4.2. The variation of Mean travel time and % trip failed shows that the path flow distribution of the scenario equilibria are different. For instance, the values of both indicators for scenario BAC are completely different from the other scenarios, and no two scenarios obtain similar equilibria.

Scenario CBA has minimum mean OD travel time for ODs 1 and 2 but the % trip failed is not minimum for OD 1. This explains that the impact of the other ODs path flow distributions prevent several users from finishing their trip even when the corresponding OD mean travel time is minimum. The system can choose a specific order for opening metro lines to minimize the mean OD travel time or % trip failed of targeted OD(s). For ODs 3 and 4, scenario CAB provides a minimum mean OD travel time and % trip failed, and finally scenario BCA has minimum indicators for OD 5.

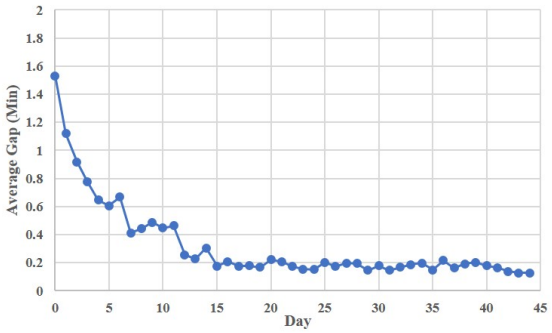
TABLE 4.2 – Mean travel time (Mean OD TT) [min] and percentage of failed trips (% trip failed) for top five most crowded OD pairs

Scenario \OD		1	2	3	4	5
A&B&C	% trip failed	5.1%	6.3%	2.0%	2.1%	6.5%
	Mean OD TT	28.4	47.3	16.8	24.2	38.6
ABC	% trip failed	4.2%	6.3%	5.2%	4.1%	7.0%
	Mean OD TT	27.1	47.4	16.1	17.4	37.8
ACB	% trip failed	4.2%	6.3%	2.0%	2.1%	6.5%
	Mean OD TT	26.6	47.3	18.1	23.7	37.7
BAC	% trip failed	4.9%	6.2%	2.1%	2.3%	6.9%
	Mean OD TT	26.9	54	20.9	26.3	39.6
BCA	% trip failed	4.4%	6.2%	4.0%	4.3%	6.5%
	Mean OD TT	26.6	47	29	34	37.1
CAB	% trip failed	4.5%	6.2%	2.0%	2.1%	7.0%
	Mean OD TT	26.6	47.9	15.7	15.6	37.7
CBA	% trip failed	5.2%	6.2%	3.3%	3.3%	6.9%
	Mean OD TT	26.6	46.7	28.2	29.8	37.7

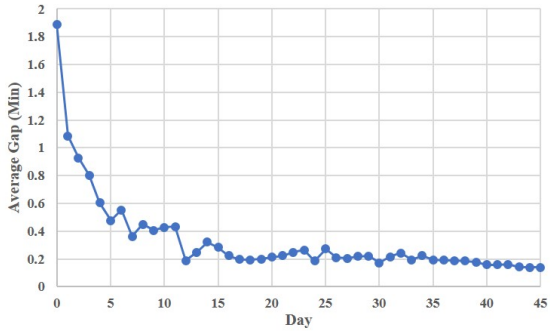
In order to complete the investigation and show that we have multiple equilibria, we need to prove that the solutions do not have similar link flow distributions. To do so, we evaluate the performance of the PT system and the mode choice of users at the equilibrium state. Table 4.3 presents the usage of PT at the equilibrium state for all scenarios. The number



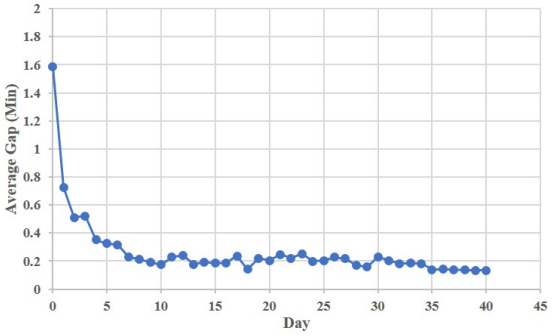
(a) Scenario 1: A&B&C



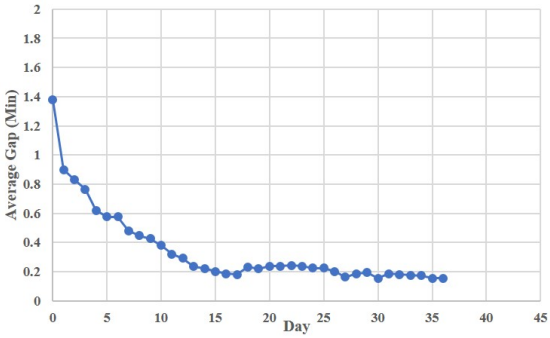
(b) Scenario 2: ABC



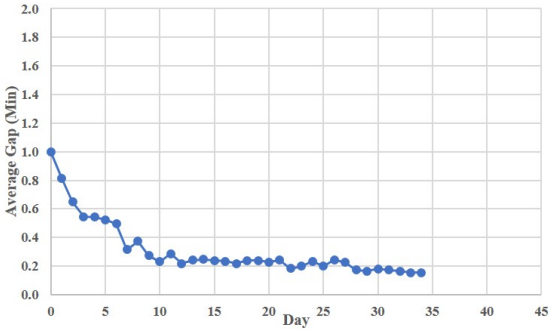
(c) Scenario 3: ACB



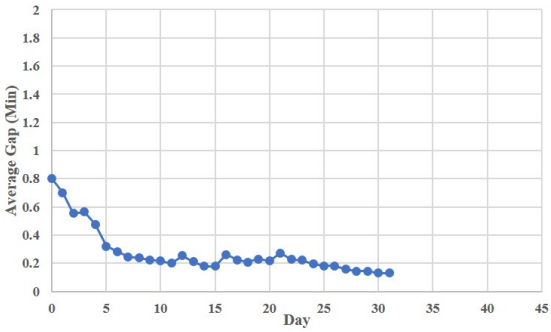
(d) Scenario 4: BAC



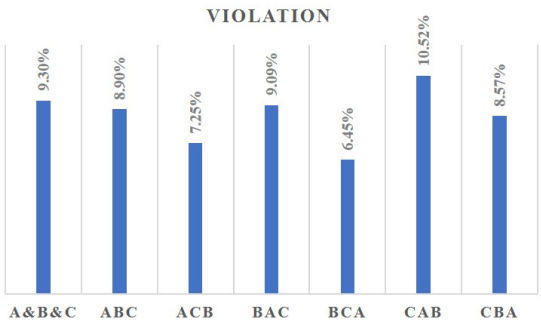
(e) Scenario 5: BCA



(f) Scenario 6: CAB



(g) Scenario 7: CBA



(g) Violation of final solution for all scenarios

FIGURE 4.11 – The average gap and violation in the day-to-day process for the final phase of all the scenarios.

of users who take metro line A is between 2089 and 2983. This means if we open the metro line in the order ACB, we will have 42% more users that take metro line A than in the CAB scenario. This width of interval for metro lines B and C is 1250 and 640. According to the demand scenario, 13530 users have to use the PT system, and other users have the choice between the PT system, driving only or both (combined mode). The other criteria in Table 4.3 show that the different orders of opening the metro lines have an impact on attracting users to use the PT system. In scenario CAB, opening metro line C at the beginning attracts more users to takes this metro line, and then metro line A and after B provides the intermediate solutions which finally converge to an equilibrium with the highest number of users using PT. Opening metro line A before B when metro line C is activated before the final step always provides a larger number of users using the PT system (e.g., when comparing scenarios CAB and CBA or ACB and BCA). Moreover, the system can manage the use of bus lines and metro lines, e.g., the equilibrium for scenario CBA balances the number of users that use both systems while in other scenarios the share of users who take the bus is larger.

TABLE 4.3 – Public transportation criteria; (#: number of)

Sequence	# of users used Metro			# of users	# of times	# of times	# of times
	A	B	C	used only PT	PT used	metro used	bus used
A&B&C	2236	3057	3708	15421	24685	9001	15684
ABC	2826	2771	3316	14751	24093	8913	15180
ACB	2983	2636	3678	15336	22780	9297	13483
BAC	2419	3077	3138	13674	25626	8634	16992
BCA	2608	3117	3481	14559	22502	9206	13296
CAB	2089	3886	3564	17284	25983	9539	16444
CBA	2313	2977	3778	15298	20888	9068	11820

The criteria for the users' choice of mode are presented in Table 4.4 for the equilibrium state of all the scenarios. The results show that we have a different number of users that decide to only drive between their OD pairs, and also a different number of vehicles in the system. Consequently, there are multiple link flow distributions for the end solution of different scenarios, as we expected from the STA case. Choosing the order of opening new metro lines can reduce the number of cars in the system by a maximum of 3%. The combined mode corresponds to the users that start their trip by car and then enter the PT system via a car-park. Scenario CAB, as explained before, motivates more users to use the PT system and also choose a combined mode more than scenario A&B&C in which metro lines are active and empty at the beginning of the experiment. This shows that the intermediate state of the network has a significant impact on the final UE.

The share of users that use the PT system and total travel time are standard criteria for evaluating traffic network performance. According to the results in Table 4.4, by opening the metro lines in the order ABC, we can save 600 hours (3%) on average compared to the other scenarios. The total travel time values of the scenarios are in the range of $[18559.33, 19701.43]$, which is the range of the potential equilibrium space.

Scenario CAB provides a lower number of cars and the highest percentage of PT system use in comparison to the other scenarios. Consequently, the system can choose the specific order for adding new transportation facilities so as to reach equilibrium with appropriate performance with respect to the indicators targeted.

TABLE 4.4 – Mode choice criteria; (#: number of)

Scenario	Sequence	# of users just drive	# of vehicles	Users used Combined mode	Users use PT	Total travel time (hours)
1	A&B&C	34142	39459	9.81%	28.46%	19454.22
2	ABC	36348	40267	7.23%	27.22%	18559.33
3	ACB	36351	40325	7.33%	28.30%	19070.50
4	BAC	37488	40516	5.59%	25.23%	18967.44
5	BCA	37014	40389	6.23%	26.87%	19199.75
6	CAB	33351	39325	11.02%	31.90%	19701.43
7	CBA	36334	40265	7.25%	28.23%	18644.19

4.7 Conclusion

In this chapter we investigated the impacts of network design history on day-to-day multimodal UE. Generally, in multimodal urban transportation networks, the monotonicity condition simply does not hold. We highlighted the source of multiple equilibria in traffic network systems (Section 4.1). Then, we studied a particular reason for multiple equilibria: network design history. When multiple facilities are progressively introduced in the system at different times, the learning process is subject to multiple steps. When users have time to adjust to these different steps, it changes the global convergence process and may lead the system towards multiple different situations while the final network setting remains the same. Based on the static and the dynamic context we demonstrated that the order of the successive introduction of such facilities matters when determining the final equilibrium. This is a crucial finding as this means that the study of the current network situation may not be sufficient to grasp the real user distribution inside the network and that it is necessary to consider the history of the network. In other words, a unique UE calculation with the current network setting may lead to an equilibrium other than that resulting from the different steps corresponding to the network history.

Another key result we obtained from the dynamic simulations is that certain final equilibria were more efficient from a systems viewpoint than others. The self-organization of the system led to different network performances depending on the history of the network. The results showed that not only do we have non-unicity, but that total travel time can be saved and other network performance indicators optimized by opening public transportation facilities in a specific order.

5. Equilibria analysis: improving traffic network performance

The purpose of this chapter is to compare network loadings related to different network equilibria by a simulation-based framework. The direct comparison of path flows or trajectory patterns is hard to achieve so here we propose a more aggregate approach based on the comparison of demand level breakpoints. A breakpoint is a demand threshold value that leads to significant changes in path flow loading. More specifically, we set in this chapter a demand breakpoint when the list of effective route alternatives differs by at least one path. This is for example the case when one route is no longer considered for one equilibrium while being used by some vehicles in the second one. We are going to investigate both static and dynamic network loading while scanning all demand levels to identify the breakpoints. We focus on discrete demand formulation and choices and use a trip-based traffic simulator.

First, we investigate breakpoints on a well-known network (Braess) in the static case in order to better define this concept. Second, breakpoints are investigated on a real network (Lyon, France) where dynamic travel times are provided by a microscopic traffic simulator. When the breakpoints are obtained for a given scenario, we focus on identifying demand level ranges where some paths are not used in SO while being travelled in UE or BRUE. Following the concept of Braess paradox, this permits to design banning strategies at some key locations in the network to prevent some alternatives from being used and thus to improve the system performance. We show by simulation that such a strategy is effective, which demonstrates the importance of breakpoint identification.

This chapter is an updated version of the paper:

Ameli, M., Lebacque, J. P. & Leclercq, L. (2020). Improving traffic network performance with road banning strategy: a simulation approach comparing user equilibrium and system optimum. *Simulation Modelling Practice and Theory*, 99, 101995, doi:10.1016/j.simpat.2019.101995.

5.1 Notations for this chapter

TABLE 5.1 – Specific notations in this chapter

Notation	Definition [units]
W	Origin-Destination (OD) pairs, subset of origin $\times$ destination nodes, $W \subset N \times N$.
a	index of link, $a \in A$.
w	index of origin and destination pair, $w \in W$.
$P_{w,\tau}$	set of paths for w in departure time τ.
$P^*_{w,\tau}$	set of shortest paths for w in departure time interval τ.
p	index of path, $p \in P_{w,\tau}$.
p^*	index of shortest path, $p^* \in P^*_{w,\tau}$.
D_w	total demand for w pair.
$C_{p,\tau}$	travel cost of path p in departure time τ.
$C^*_{w,\tau}$	minimum travel cost of OD pair w in departure time τ.
$\hat{C}_{p,\tau}$	marginal travel cost of path p in departure time τ.
$\hat{C}^*_{w,\tau}$	minimum marginal travel cost of OD pair w in departure time τ.

5.2 Motivations

The difference of the total travel cost between User Equilibrium (UE) and System Optimum (SO) is called Price of Anarchy (PoA) (Roughgarden, 2005). There are several researches about how to reduce PoA in the literature (Youn *et al.* (2008), Colini-Baldeschi *et al.* (2017)). The goal of this chapter is to investigate the evolution of PoA depending on the level of demand by introducing a new concept named demand level breakpoints. This concept permits to identify demand level ranges where PoA is high because of some paths which are used in UE whereas they are not used in SO. Such situations are known in the literature as the Braess paradox (Frank, 1981). While being highly documented this paradox remains very hard to detect in the real field as the level of demand plays a crucial role (Askoura *et al.*, 2011). This study aims to go one step further in a better identification of these situations by introducing systematic methods to determine the demand level breakpoints. At the end, we will show how the identification of breakpoints enables to design efficient control strategies at the network level that consist in banning some routes when critical demand level values are experienced.

Path costs can be estimated based on models or simulators. As discussed in Section 1.1.3, traffic assignment problems are classified into three main groups: Static Traffic Assignment (STA), semi-dynamic and Dynamic Traffic Assignment (DTA) (Bliemer *et al.*, 2017). Since the 1950s there is much research about finding the assignment solutions for UE and SO. As discussed before and based on reviews of Szeto & Wong (2012), Wang *et al.* (2018), Mahmassani (2001), a simulation based approach is preferred because the results are easy to interpret and relatively close to reality (Sundaram *et al.*, 2011).

Identifying situations in which the Braess paradox holds and induces the breakpoints means in practice investigating the qualitative differences between UE and SO equilibrium. The UE acts as a proxy for the current network situation whereas SO reproduces the optimal situations with smart traffic control (Ehrgott *et al.*, 2015). Moreover, it is well-known that users are not always taking the shortest paths (Szeto & Lo, 2006a) because they lack a perfect knowledge of the traffic conditions (Mahmassani & Chang, 1987, Delle Site, 2018) or because they also favor other criteria when choosing their travel path (Abdel-Aty *et al.*, 1997, Zhou *et al.*, 2017). To investigate the robustness of breakpoint definition while relaxing perfect UE assumptions, we are going to calculate also Boundedly Rational User Equilibrium (BRUE). BRUE can be considered as a relaxation of UE where users try to optimize their own benefit up to a point but stop the process where they are satisfied with the current solution (indifference bound). Equilibrium is achieved when all user costs are within the boundary around the UE solution (Di & Liu, 2016, Han *et al.*, 2015b).

In this chapter, we consider UE, SO and BRUE independently and try to investigate the relation between these three equilibria for different demand levels. To the best of our knowledge the studies in the literature focus mainly on finding the path flow distributions over the global network related to each equilibrium. Here, we would like to go further and cross-compare trip patterns. We attempt to design new traffic management methods based on the idea of incenting users to change paths, so that the network moves closer to SO. The questions are which users should be switched from one route to another and which routes should be promoted by the route guidance. We are going to directly answer these questions by studying demand level breakpoints and apply some banning and rerouting strategy.

Generally, the travel demand is not fixed even in the short term. There are few studies

that focus on the impact of different levels of demand on the UE (see e.g. Wie *et al.* (2002), Szeto & Lo (2004), Han *et al.* (2011), D'Ambrogio *et al.* (2009), Han *et al.* (2015a)). First we define breakpoints and analyze them. Then, for each level of demand, and given a fixed active paths set, we try to analyze how we can determine the critical users for rerouting. Critical users are the users who, when they change their path, have more impact than others on moving the equilibrium towards SO conditions. There are four main points that we aim at investigating:

- The impact of one origin-destination demand level on the network equilibria.
- Critical users who can have maximum impact on moving UE and BRUE toward SO by change their path.
- Demand level ranges that we can apply control strategies to follow the SO assignment by respecting the network UE or BRUE.
- Does the banning strategy work in order to shift the network UE/BRUE towards an equilibrium with lower total travel costs?

The layout of this chapter is as follows: Section 5.3 presents the definition of the breakpoint and solves a simple static traffic model in order to define the breakpoints in the Braess network. In Section 5.4, we explain the dynamic equilibrium model and how we calculate the breakpoints in DTA context. The network for numerical experiments and the process of breakpoint detection are presented in the Section 5.5. We consider two test cases on the dynamic network. The control strategy and the obtained results for the first test case are discussed in Section 5.6. The second test case and the application of breakpoint detection and ban strategy to the second test case are presented in section 5.7. Finally, we state concluding remarks of this chapter in the Section 5.8 section.

5.3 Breakpoint definition

Traffic assignment provides the path flow distribution for all OD pairs, i.e. how many users take each possible path from each origin to each destination. There is a path set which contains all the active paths from each origin to each destination. A path chosen by at least one user is an active path. The active path set is a component of the optimal path flow distribution defined for each equilibrium: UE, SO and BRUE. For a given origin and destination pair w, the active path set for a given equilibrium can be the same at different levels of demand. We define the breakpoints as demand levels where we observe a change in the active path set (e.g. one new path in or/and one current path out). We will first study the breakpoints in the static case on the classical Braess network (Braess *et al.*, 2005). As explained in the introduction, the Braess paradox arises when UE and SO have difference path set. Determining the breakpoint for both equilibria will permit to easily identify such situations.

The classic Braess network with linear cost functions in the static case is shown is Figure 5.1. There are five links with cost functions t_{ij} for the link ij connecting node i to node j. The flow of the link ij is f_{ij} and there are three alternative paths from origin node 1 to destination node 4: path 1 (1-3-4), path 2 (1-2-4) and path 3 (1-3-2-4). Therefore the cost functions (C_k) of paths k are as follow:

$$C_1(\pi) = t_{13} + t_{34} \tag{5.1}$$

$$C_2(\pi) = t_{12} + t_{24} \tag{5.2}$$

$$C_3(\pi) = t_{13} + t_{32} + t_{24} \tag{5.3}$$

Where π_k denotes the flow of path k and $\pi = (\pi_k)_{k=1,2,3}$.

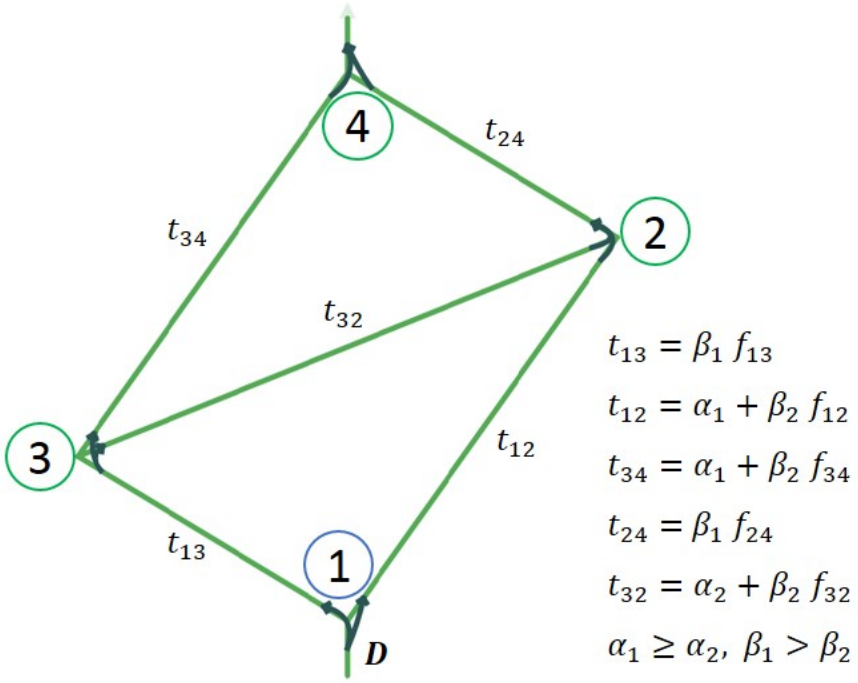


FIGURE 5.1 – Classic Braess network with the link cost functions and assumptions

The two assumptions in Figure 5.1 ($\alpha_1 \geq \alpha_2$, $\beta_1 > \beta_2$) guarantee that by increasing the demand, there exist three cases of UE (see e.g. Pas & Principio (1997)). They also ensure that the path 3 is the cheapest free flow path, due to lower coefficients (α_2, β_2) in the path cost function. This is a critical condition for the Braess paradox to appear. The different demand levels will be defined in relation with the structure of the equilibrium (active paths). Consequently, for UE we will have three scenarios. According to the definition of the UE, the conditions of static UE can be stated mathematically as follows:

$$C_p - {C_w}^* \geq 0 \quad ; \forall p \in P_w \tag{5.4}$$

$$\pi_p(C_p - {C_w}^*) = 0 \quad ; \forall p \in P_w \tag{5.5}$$

$$\pi_p \geq 0 \quad ; \forall p \in P_w \tag{5.6}$$

Where ${C_w}^*$ is the minimum travel cost for origin-destination pair w. P_w is the set of possible paths for w and π_p denotes the flow on path p.

The path flow distribution is a $[1 \times 3]$ vector π that contains the flow value for three paths. Note that the feasible path flow vector is defined as: $\Pi \triangleq \{\pi : \pi \geq 0, \sum_{p \in P_w} \pi_p = D_w\}$. So the active path set is the set of path $p \in P_w$ such that $\pi > 0$. The path flow

distribution for UE is a function $UE(D)$ of demand level D and can be shown to be as follows:

$$UE(D) = \begin{cases} [0,\, 0,\, D] & 0 \leq D < \frac{\alpha_1-\alpha_2}{\beta_1+\beta_2}, \\ [\frac{-\alpha_1+\alpha_2+(\beta_1+\beta_2)D}{\beta_1+3\beta_2},\, \frac{-\alpha_1+\alpha_2+(\beta_1+\beta_2)D}{\beta_1+3\beta_2},\, \frac{2(\alpha_1-\alpha_2)-(\beta_1-\beta_2)D}{\beta_1+3\beta_2}] & \frac{\alpha_1-\alpha_2}{\beta_1+\beta_2} \leq D < \frac{2(\alpha_1-\alpha_2)}{\beta_1-\beta_2}, \\ [\frac{D}{2},\, \frac{D}{2},\, 0] & D \geq \frac{2(\alpha_1-\alpha_2)}{\beta_1-\beta_2} \end{cases} \tag{5.7}$$

Consequently, there are two breakpoints for UE: $D = \frac{\alpha_1-\alpha_2}{\beta_1+\beta_2}$ and $D = \frac{2(\alpha_1-\alpha_2)}{\beta_1-\beta_2}$.
Let us now investigate the network under BRUE. The boundary rational model is one behavioral model used to relax the perfect rational hypothesis in the definition of UE by considering an indifference band (ϵ). There are two main differences in the definition of breakpoints between the BRUE and UE:

- The path flow distribution for BRUE not only depends on the demand level, it also depends on the ϵ value for the indifference band for this origin and destination pair.
- The path flow distribution is not unique in some scenarios, so for the Braess network there are two convex sets (simplices) of flow vectors ϕ_1 and ϕ_2 which can satisfy the conditions of BRUE:

$$\phi_1 = \{\pi \geq 0 : C_i(\pi) - C_j(\pi) \leq \epsilon;\, i,\, j \in \{1,\, 2\},\, i \neq j\} \tag{5.8}$$

$$\phi_2 = \{\pi \geq 0 : C_i(\pi) - C_j(\pi) \leq \epsilon;\, i,\, j \in \{1,\, 2,\, 3\},\, i \neq j\} \tag{5.9}$$

Note that Equation 5.8 is obtained by assuming that both paths are used. In Equation 5.9, ϕ_2 is a set in which all three paths can be active. If the indifference band is equal to zero ($\epsilon = 0$) the BRUE and UE assignment are the same. Due to the non-uniqueness of the BRUE, for a given ϵ, an active path has a cost lower or equal to the minimum travel cost path plus ϵ. The ϵ-BRUE path flow pattern can be shown to be given by:

$$\pi_p > 0 \rightarrow C_p - {C_w}^* \leq \epsilon \quad ;\forall p \in P_w \tag{5.10}$$

The path flow distribution for BRUE is a function of the demand level and the band value $BRUE(D,\, \epsilon)$ is given by:

$$BRUE(D,\, \epsilon) = \begin{cases} [0,\, 0,\, D] & 0 \leq D < \frac{\alpha_1-\alpha_2}{\beta_1+\beta_2} - \frac{\epsilon}{\beta_1+\beta_2}, \\ \phi_1 \cup \phi_2 & D > \frac{2(\alpha_1-\alpha_2)}{\beta_1-\beta_2} + \frac{\epsilon}{\beta_1-\beta_2}, \\ \phi_2 & \text{o.w.} \end{cases} \tag{5.11}$$

For a detailed calculation of Equation 5.11 readers can refer to Di *et al.* (2014). Therefore, there are two breakpoints but they depend on the ϵ value: $D = \frac{\alpha_1-\alpha_2}{\beta_1+\beta_2} - \frac{\epsilon}{\beta_1+\beta_2}$ and $D = \frac{2(\alpha_1-\alpha_2)}{\beta_1-\beta_2} + \frac{\epsilon}{\beta_1-\beta_2}$.

The third considered equilibrium is SO. The SO path flow distribution is based on the second principle of Wardrop (1952), i.e. minimizing the total travel cost. Therefore, mathematically the goal is to minimize the total travel cost as a function of demand and flow vector:

$$\min \quad T(D,\, \pi) = \sum_{p \in P_w} \pi_p C_p \tag{5.12}$$

Given our assumptions on coefficient values for the Braess network (Figure 5.1), three scenarios that can occur as in the UE case. The objective function in each scenario is given by:

$$T(D) = \begin{cases} (2\beta_1 + \beta_2)D^2 + \alpha_2 D & \text{if only path 3 used,} \\ (\beta_1 + \beta_2)({\pi_1}^2 + {\pi_2}^2) + \alpha_1 D & \text{if path 1 and 2 used,} \\ (\beta_1 + 2\beta_2) + & \\ [\alpha_1 - \alpha_2 - 2(\beta_1 + \beta_2)D](\pi_1 + \pi_2) + & \text{if all paths used and } \pi_3 = D - (\pi_1 + \pi_2) \\ 2\beta_2\pi_1\pi_2 + (2\beta_1 + \beta_2)D^2 + \alpha_2 D & \end{cases} \tag{5.13}$$

Therefore and mathematically, the path flow distribution for SO is defined as follow:

$$SO(D) = \begin{cases} [0,\ 0,\ D] & 0 \leq D < \frac{\alpha_1 - \alpha_2}{2(\beta_1 + \beta_2)}, \\ [\frac{\frac{-\alpha_1+\alpha_2}{2} + (\beta_1+\beta_2)D}{\beta_1+3\beta_2},\ \frac{\frac{-\alpha_1+\alpha_2}{2} + (\beta_1+\beta_2)D}{\beta_1+3\beta_2},\ \frac{(\alpha_1-\alpha_2)-(\beta_1-\beta_2)D}{\beta_1+3\beta_2}] & \frac{\alpha_1-\alpha_2}{2(\beta_1+\beta_2)} \leq D < \frac{\alpha_1-\alpha_2}{\beta_1-\beta_2}, \\ [\frac{D}{2},\ \frac{D}{2},\ 0] & D \geq \frac{\alpha_1-\alpha_2}{\beta_1-\beta_2} \end{cases} \tag{5.14}$$

There are two breakpoints for SO: $D = \frac{\alpha_1-\alpha_2}{2(\beta_1+\beta_2)}$ and $D = \frac{\alpha_1-\alpha_2}{\beta_1-\beta_2}$. There is a fixed relationship between breakpoints of SO and UE which is shown in Equation 5.15. The relationship between BRUE's breakpoints and the breakpoints of SO is depending on the value of ϵ.

$$BP_{i,SO} = \frac{1}{2} BP_{i,UE} \ ; \ \forall i \in \{1, 2\} \tag{5.15}$$

$$BP_{1,SO} = \frac{1}{2}[BP_{1,BRUE} + \frac{\epsilon}{(\beta_1 + \beta_2)}] \ ; \ BP_{2,SO} = \frac{1}{2}[BP_{2,BRUE} - \frac{\epsilon}{(\beta_1 - \beta_2)}] \tag{5.16}$$

Where $BP_{i,j}$ is the ith breakpoint of the equilibrium $j \in \{UE, SO, BRUE\}$.
As aforementioned, existing studies usually finish the breakpoints analysis here (finding the path flow distribution for equilibria). We are now going to analyze the equilibria based on breakpoints.

The path flow distribution for UE, SO and BRUE is shown in Figure 5.2 to present the optimal path set of two different types of paths in the Braess network. It is not easy to present the BRUE in the path flow distribution diagram because the ϵ allowance for cost implies that for each path there is a set of possible flow values in the BRUE path flow distribution around the UE flow value. Thus in Figures 5.2(a) and 5.2(b), we represent the BRUE solution in terms of the maximum range of flows at equilibrium. Figure 5.2(a) presents the demand-flow diagram for path 1 and 2 in UE, SO and BRUE situations. Breakpoints are shown on the demand axis. (a↔b) presents the active path set for UE and SO and also possible active path set for BRUE depending on the value of ϵ. The same demand-flow diagram for path 3 is shown in Figure 5.2(b). Note that the breakpoints of BRUE, $\frac{\alpha_1-\alpha_2}{\beta_1+\beta_2} - \frac{\epsilon}{\beta_1+\beta_2}$ and $\frac{2(\alpha_1-\alpha_2)}{\beta_1-\beta_2} + \frac{\epsilon}{\beta_1-\beta_2}$, are equal to UE breakpoints when $\epsilon = 0$.

The first breakpoint occurs when $D = \frac{\alpha_1-\alpha_2}{2(\beta_1+\beta_2)}$ for analyzing the UE solution and SO. Determining this level of demand as a breakpoint means that the active path set of one of the equilibria is changed. Before this breakpoint, the active path set of both equilibria has the

same path: $P_{UE} = P_{SO} = \{3\}$. In other words, from $D = 0$ to $D = \frac{\alpha_1 - \alpha_2}{2(\beta_1 + \beta_2)}$, the equilibrium under both disciplines refers to the same active path set. Then after this breakpoint, only paths 1 and 2 should be used under SO discipline while path 3 is the only active path for UE. From the first breakpoint of the SO ($BP_{1,SO}$) to the first breakpoint of the UE ($BP_{1,UE}$), the flow is distributed differently for UE and SO. For UE all users are still taking the third path but for SO some of them start to take the paths 1 and 2 to minimize the total travel time. It means that for demand level higher than the $BP_{1,SO}$, paths 1 and 2 will be added to the active path set in SO solution. The active path sets for UE and SO are fixed after $BP_{1,SO}$ until $BP_{1,UE}$ is detected at $D = \frac{\alpha_1 - \alpha_2}{\beta_1 + \beta_2}$. Where paths 1 and 2 are used under UE discipline. The active path set is the same for UE and SO when the demand level is $\frac{\alpha_1 - \alpha_2}{\beta_1 + \beta_2} \leq D < \frac{\alpha_1 - \alpha_2}{\beta_1 - \beta_2}$. Path 3 is not in the active path set for SO but used in UE solution when we pass the $BP_{2,SO}$ and the demand is in range $\frac{\alpha_1 - \alpha_2}{\beta_1 - \beta_2} \leq D < \frac{2(\alpha_1 - \alpha_2)}{\beta_1 - \beta_2}$. Therefore the $BP_{2,UE}$ is $D = \frac{2(\alpha_1 - \alpha_2)}{\beta_1 - \beta_2}$ where the active path set of UE is changed. In Braess network, after the last breakpoint, $D \geq \frac{2(\alpha_1 - \alpha_2)}{\beta_1 - \beta_2}$, the UE and SO path flow distribution will be the same.

In general, the BRUE path flow distribution is close to UE but the solution is not symmetric with respect to paths 1 and 2 as in SO and UE. In the BRUE-SO analysis, $D = \frac{\alpha_1 - \alpha_2}{2(\beta_1 + \beta_2)}$ is the first breakpoint as in the UE-SO only if $\epsilon \leq \frac{\alpha_1 - \alpha_2}{2}$. If $\epsilon > \frac{\alpha_1 - \alpha_2}{2}$, the first breakpoint is located in the lower demand level with respect to the first SO breakpoint. Path 1 and(or) path 2 are added to the path set after the first breakpoint. For higher demand levels, by taking into account ϵ, the BRUE-SO breakpoints are same as the UE-SO breakpoints until the last breakpoint, when path 3 exits the active path set. If $D \geq \frac{2(\alpha_1 - \alpha_2)}{\beta_1 - \beta_2}$, the BRUE and SO path flow distribution will be the same only if $\epsilon = 0$. It means, for $\epsilon > 0$, the last breakpoint will be $\frac{2(\alpha_1 - \alpha_2)}{\beta_1 - \beta_2} + \frac{\epsilon}{\beta_1 - \beta_2}$. All paths are active for BRUE and path 3 is not active for SO while $\frac{2(\alpha_1 - \alpha_2)}{\beta_1 - \beta_2} \leq D < \frac{2(\alpha_1 - \alpha_2)}{\beta_1 - \beta_2} + \frac{\epsilon}{\beta_1 - \beta_2}$. Consequently, the path flow distribution is the same for BRUE and SO when $D \geq \frac{2(\alpha_1 - \alpha_2)}{\beta_1 - \beta_2} + \frac{\epsilon}{\beta_1 - \beta_2}$. Note that BRUE accepts 3 non zero path-flows for values of $D < \frac{\alpha_1 - \alpha_2}{\beta_1 + \beta_2} - \frac{\epsilon}{\beta_1 + \beta_2}$ when $\epsilon > \alpha_1 - \alpha_2$. Moreover, if $\epsilon \leq \frac{\alpha_1 - \alpha_2}{2}$ and $D < \frac{\alpha_1 - \alpha_2}{2(\beta_1 + \beta_2)}$, the solution is the same for all 3 equilibria. Finally, the mathematical formulas for path flow distributions for the 3 equilibria (UE, SO and BRUE) are shown in Table 5.2. This table is obtained by merging the equations 5.7, 5.11 and 5.14 in order to jointly investigate the breakpoints when demand is increasing.

This study by breakpoint analysis attempts to identify ranges of demand where we have a qualitative difference between UE/BRUE and SO. We are looking in particular for situations where some paths are used in UE/BRUE while not in SO. This corresponds to demand ranges with breakpoints as boundaries, because breakpoints are identifying changes in the active paths for each equilibrium. Then, we can apply a control strategy in order to improve the performance of the network. At each range of demand between two breakpoints, we know the active path set of SO and the network state (UE or BRUE). So at the range of demand where $P_{UE/BRUE} \neq P_{SO}$, changing the path set may improve the system. For instance, when $\frac{2(\alpha_1 - \alpha_2)}{\beta_1 - \beta_2} \leq D < \frac{2(\alpha_1 - \alpha_2)}{\beta_1 - \beta_2} + \frac{\epsilon}{\beta_1 - \beta_2}$ path 3 is not used under SO discipline. So in this demand interval the system controller can ban path 3 or design other control strategies in order to prevent users from travelling on path 3, and the system will move towards the SO state. In other ranges of demand the network controller can induce a certain number of users to travel by specific paths, in order to get closer to SO. Such users are critical for an efficient

control strategy.

The breakpoint analysis helps the system controller to evaluate the possibility of applying control strategies in order to improve the system performance. In the sequel, we focus on the banning strategy in which, by banning one link or by preventing users from turning at one intersection, the controller can ban one path which is in the active path set of the UE solution but does not belong to the SO active path set. An alternate possibility, which will be explored in future work, consists in using guidance. In urban networks, transportation system controllers can guide (e.g. by Advanced Traveler Information System (ATIS)) the limited number of users that are equipped (Klein *et al.*, 2018). Generally, it will be useful to know how many users need to change their path in order to move the network from one equilibrium on the user side (UE or BRUE) to one equilibrium on the system side (SO).

This part of the chapter has attempted to explain the path flow distribution breakpoints concept and to define them in a simple manner using the static assignment on the Braess network. In the following parts of this chapter, we present the dynamic trip-based framework to find the breakpoints and identify the critical user(s) for rerouting and critical path(s) for banning.

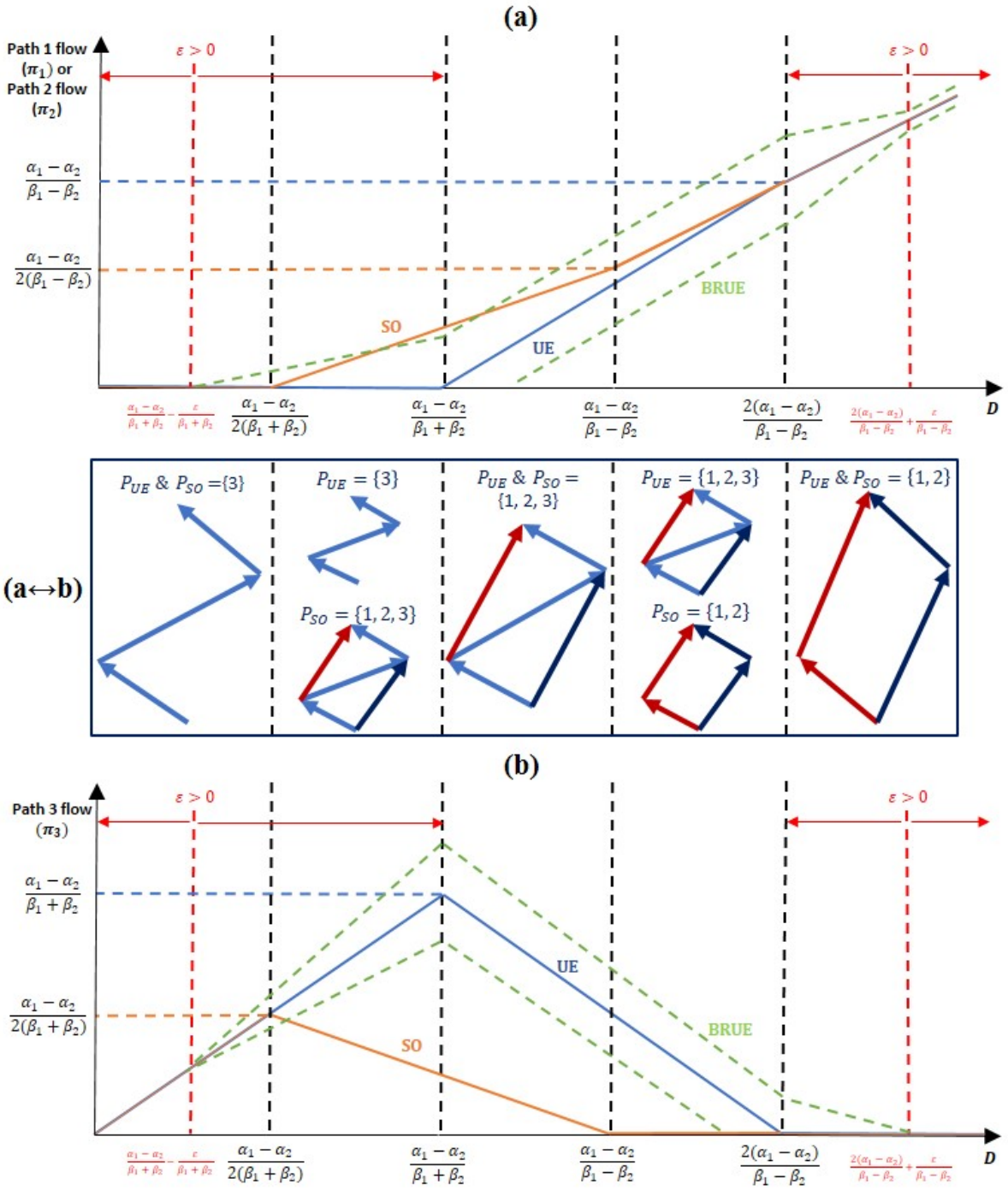


FIGURE 5.2 – (a): Path 1 or Path 2 flow-demand diagram for UE, SO and BRUE. (a↔b): Possible path set of optimal solution [P_x = Possible path set of optimal solution for equilibrium x, where x stands for UE or SO]. Note that for BRUE, it depends on the ϵ. (b): Path 3 flow-demand diagram for UE, SO and BRUE.
Red dash lines in figures (a) and (b) presents the value of breakpoints in BRUE path flow distribution based on the given ϵ. It can change in ranges that are specified by the red arrows.

TABLE 5.2 – Path flow distribution in Braess network for UE, SO and BRUE

Demand level (D)	User Equilibrium (UE)	System optimum (SO)	Band Value (ε)	Boundary rational UE
$0 \le D \le \frac{\alpha_1 - \alpha_2}{2(\beta_1 + \beta_2)}$	$[0,0,D]$	$[0,0,D]$	$\varepsilon < \alpha_1 - \alpha_2 - (\beta_1 + \beta_2)D$	$[0,0,D]$
			$\varepsilon > \alpha_1 - \alpha_2 - (\beta_1 + \beta_2)D$	φ_2
$\frac{\alpha_1 - \alpha_2}{2(\beta_1 + \beta_2)} \le D \le \frac{\alpha_1 - \alpha_2}{\beta_1 + \beta_2}$	$[0,0,D]$	$[\frac{\frac{-\alpha_1+\alpha_2}{2} + (\beta_1+\beta_2)D}{\beta_1+3\beta_2}, \frac{\frac{-\alpha_1+\alpha_2}{2} + (\beta_1+\beta_2)D}{\beta_1+3\beta_2}, \frac{(\alpha_1-\alpha_2)-(\beta_1-\beta_2)D}{\beta_1+3\beta_2}]$	$\varepsilon < \alpha_1 - \alpha_2 - (\beta_1 + \beta_2)D$	$[0,0,D]$
			$\varepsilon > \alpha_1 - \alpha_2 - (\beta_1 + \beta_2)D$	φ_2
$\frac{\alpha_1 - \alpha_2}{\beta_1 + \beta_2} \le D \le \frac{\alpha_1 - \alpha_2}{\beta_1 - \beta_2}$	$[\frac{-\alpha_1+\alpha_2+(\beta_1+\beta_2)D}{\beta_1+3\beta_2}, \frac{-\alpha_1+\alpha_2+(\beta_1+\beta_2)D}{\beta_1+3\beta_2}, \frac{2(\alpha_1-\alpha_2)-(\beta_1-\beta_2)D}{\beta_1+3\beta_2}]$	$[\frac{\frac{-\alpha_1+\alpha_2}{2} + (\beta_1+\beta_2)D}{\beta_1+3\beta_2}, \frac{\frac{-\alpha_1+\alpha_2}{2} + (\beta_1+\beta_2)D}{\beta_1+3\beta_2}, \frac{(\alpha_1-\alpha_2)-(\beta_1-\beta_2)D}{\beta_1+3\beta_2}]$	All values	φ_2
$\frac{\alpha_1 - \alpha_2}{\beta_1 - \beta_2} \le D \le \frac{2(\alpha_1 - \alpha_2)}{\beta_1 - \beta_2}$	$[\frac{-\alpha_1+\alpha_2+(\beta_1+\beta_2)D}{\beta_1+3\beta_2}, \frac{-\alpha_1+\alpha_2+(\beta_1+\beta_2)D}{\beta_1+3\beta_2}, \frac{2(\alpha_1-\alpha_2)-(\beta_1-\beta_2)D}{\beta_1+3\beta_2}]$	$[\frac{D}{2}, \frac{D}{2}, 0]$	All values	φ_2
$D \ge \frac{2(\alpha_1 - \alpha_2)}{\beta_1 - \beta_2}$	$[\frac{D}{2}, \frac{D}{2}, 0]$	$[\frac{D}{2}, \frac{D}{2}, 0]$	$\varepsilon < -2(\alpha_1 - \alpha_2) + (\beta_1 - \beta_2)D$	$\varphi_1 \cup \varphi_2$
			$\varepsilon > -2(\alpha_1 - \alpha_2) + (\beta_1 - \beta_2)D$	φ_2

5.4 Simulation-based dynamic network equilibrium

The concept and the application of breakpoint detection are presented in the static case. This section presents the network equilibrium model and tools to calculate the equilibria in the dynamic case. In this work, The Symuvia platform, including the trip-based simulator (Section 1.2.2) and the command module: SymuMaster (Section 1.2.3) is used in order to calculate travel times in the network for any given path flow distributions for all OD pairs. The travel demand is given (dynamic OD pair demand) and users' routes are determined by a dynamic traffic assignment model, which guides each vehicle in the network on the route that optimizes its travel time to its initially assigned destination based on some specific equilibria discipline (UE, SO and BRUE are considered here).

5.4.1 Network equilibrium model

Here, we recall the UE conditions in order to extend the conditions to SO and BRUE disciplines. Let us consider a network $G(N, A)$ with a finite set of nodes N and a finite set of directed links A. The demand is time-dependent. The period of interest (planning horizon) of duration H is discretized into a set of small time intervals indexed by τ ($\tau \in T = \{\tau_0, \tau_0 + \sigma, \tau_0 + 2\sigma, ..., \tau_0 + M\sigma\}$ and $\tau_0 + M\sigma = H$). σ is the duration of the time intervals. In an interval τ, travel times and traffic conditions do not change. The main notations to introduce in the dynamic equilibrium model are presented in Table 5.1. According to the definition of the variables and time interval in DTA and based on the study of (Sbayti *et al.*, 2007), the conditions of dynamic UE can be mathematically restated from equations 5.4, 5.5 and 5.6:

$$C_{p,\tau} - C^*_{w,\tau} \geq 0 \quad ;\forall w \in W,\ p \in P_{w,\tau},\ \tau \in T \tag{5.17}$$

$$\pi_{p,\tau}(C_{p,\tau} - C^*_{w,\tau}) = 0 \quad ;\forall w \in W,\ p \in P_{w,\tau},\ \tau \in T \tag{5.18}$$

$$\pi_{p,\tau} \geq 0 \quad ;\forall p \in P_{w,\tau},\ \tau \in T \tag{5.19}$$

The equilibrium condition for BRUE is:

$$\pi_{p,\tau} > 0 \rightarrow C_{p,\tau} - {C_{w,\tau}}^* \leq \epsilon \quad ;\forall p \in P_{w,\tau} \tag{5.20}$$

5.4.2 Equilibration process

In order to find the equilibrium, we are using an iterative algorithm. As the sub-area network Lyon 6 is not a large-scale network, we use MSA ranking method, which is introduced in Section 2.5.2.1. The algorithm is executed until the final solution does not change, and finally, we converge by the best solution in terms of closeness to equilibrium (see Section 2.5.1).

5.4.3 Definition of SO for dynamic case

In the static case, we focused on specific OD pair in mono-OD network but here we are going to focus on specific OD pairs in the network with multiple OD pairs. Therefore we consider two different definitions for SO: Global SO and Local SO.

5.4.3.1 Global SO

The SO conditions in the dynamic case are mathematically stated in equations 5.21, 5.22 and 5.23 based on marginal travel time. The path marginal travel time is the extra travel time that will be added to the path travel time if an extra user is assigned to the path at the current time interval. The SO conditions state that if the path flow is positive, then the experienced path marginal travel time should be equal to the minimum path marginal travel times (Sbayti *et al.*, 2007).

$$\hat{C}_{p,\tau} - \hat{C}^*_{w,\tau} \geq 0 \quad ;\forall w \in W,\ p \in P_{w,\tau},\ \tau \in T \tag{5.21}$$

$$\pi_{p,\tau}(\hat{C}_{p,\tau} - \hat{C}^*_{w,\tau}) = 0 \quad ;\forall w \in W,\ p \in P_{w,\tau},\ \tau \in T \tag{5.22}$$

$$\pi_{p,\tau} \geq 0 \quad ;\forall p \in P_{w,\tau},\ \tau \in T \tag{5.23}$$

Computing the path marginal travel time analytically is very costly. In Leclercq *et al.* (2016) it has been shown that using the simulation-based approach to compute the path marginals is also very costly, even in a simple grid network. Therefore, we use the simulator to compute a surrogate model for the marginal travel time. We used three methods for calculating the SO and consider the minimum total travel time as the SO path flow distribution. The first method updates the link marginal every time a vehicle is exiting the link by calculating the marginal variation since the last exit the link. Finally, link marginal are averaged using the 1 minute window. Consequently, the path marginal travel time can obtained by:

$$\hat{C}_{p,\tau} = \sum_{a \in A_p} [\frac{\sum_{t=\tau}^{\tau+\sigma} \hat{C}_{a,t}}{\sigma}] \tag{5.24}$$

Where $\hat{C}_{a,\tau}$ is the marginal travel time of link a at second t, σ is the length of each time interval and A_p is the subset of link set A which defines path p. In fact, the surrogate function by Peeta & Mahmassani (1995) considers the sum of link marginal travel times as the actual function. In the SO problem, we aim to minimize the cost in Equation 5.24 for all users of each OD pair. We also consider two methods based on the observation of trajectories at the link level. The observation process of link marginal travel time for both methods (second and third) is the same but in the second method we use the average operator for link marginal travel time and for the third one, median operator is used in order to calculate the link marginal travel time. Finally, the path marginal travel time will be the sum of link marginal travel times of that path like the first method. The last two methods are well defined in Yildirimoglu & Kahraman (2018). For each scenario of SO, the simulation is executed three times with these methods and the path flow distribution by minimum total travel time is considered as the SO solution.

5.4.3.2 Local SO

The idea of the local SO is to look for a SO to be achieved only by the users of specific OD pair(s), while all other users choose the path based on the UE framework. This permits to study control strategies that are focusing on a single OD pair in the network while other

users for other OD pairs are not targeted. Therefore, Conditions 5.21, 5.22 and 5.23 for SO are applied to the users of specific OD pair(s) who are aiming to achieve SO and Conditions 5.17, 5.18 and 5.19 for UE are applied to the other users in the network (who choose their paths according to UE). Finally we will have a mixed equilibrium in the network which is called "ME" solution in this chapter.

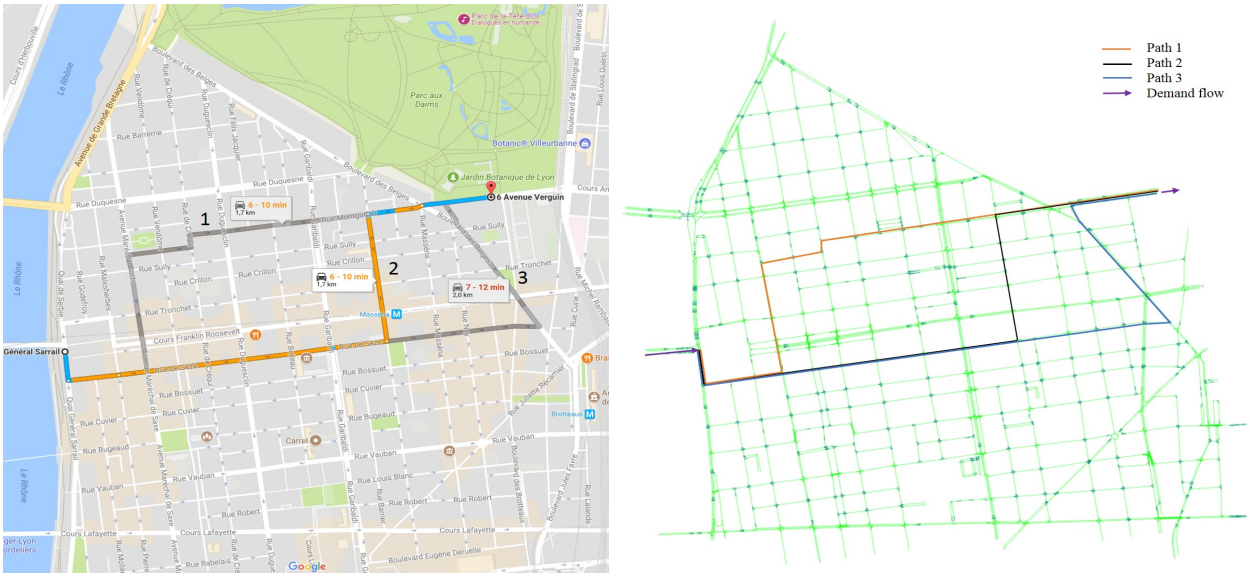
5.5 Dynamic test case

As we are now using a simulator to derive dynamic travel time, we can investigate a more complex network configuration than the Braess network. We are then now considering a sub-area of the Lyon full regional network. It is a network of the Lyon 6 district (Figures 5.3(a)). It is a Braess-like network when considering a specific OD pair that corresponds to travel from the west ("Quai de Serbie") to the east of the network ("6 Avenue Verguin"). In this network we are looking for the breakpoints at various demand levels. Note that the travelers of other OD pairs load the complete network, in order to represent the peak half an hour of the network based on the study of Krug *et al.* (2019). The simulation and optimization are carried out for each level of demand with a 30 minutes horizon. The demand pattern has been set up to adjust the regular level observed in this area during the peak hour. We select the 3 most likely routes from each origin to destination to define the set of path candidates. They are shown in figures Figures 5.3(b). The network of Lyon 6 has: 430 nodes, 786 links, 26 origins, 24 destinations and 3732 trips for OD pairs excluding the demand of "Quai de Serbie" to "6 Avenue Verguin".

We search for the three equilibria in the complete network. It means that the optimization process is executed for all OD pairs and users in the network and we try to analyze the breakpoints by increasing the demand level on the specific OD pair. A simulation-based DTA is used to find the UE, SO and BRUE at every level of the demand on the three predefined paths. The departure time for the test users is spread uniformly in the 30 minutes duration of the horizon. Moreover, the users with other OD pairs travel with fixed departure time. According to the scale of the network, for calculating the BRUE assignment, we consider $\epsilon = 0.1\, C_{p,\tau}$. It means users will be satisfied if they perceive a maximum ten percent more than the shortest path cost.

5.5.1 Breakpoint detection in the dynamic case

As mentioned earlier, we solve the problem with different demand levels. Demand of the specific "Quai de Serbie to Verguin" OD pair is increased from one user to 1439 users (maximum demand based on the study of Krug *et al.* (2019)) over the simulation time horizon = 30 min. We focus on the travels from the specific "Quai de Serbie to Verguin" OD pair with 3 paths like the Braess network in order to detect the breakpoints which are expected to exist in the real network. The departure time is fixed for all users who do not travel between the specific OD pair. The distribution of departure time for this OD pair is uniform. The optimization process has been carried out for all equilibria and the full network. All experiments for UE and SO are reproducible and lead to a unique solution but for BRUE the solution depends on the first network initialization (starting point in optimization process) and the value of ϵ. Note that we will present one instance for the BRUE path flow



(a) Origin, destination and paths (Mapping data ©Google 2019)

(b) The test case of Lyon 6 network with one OD pair

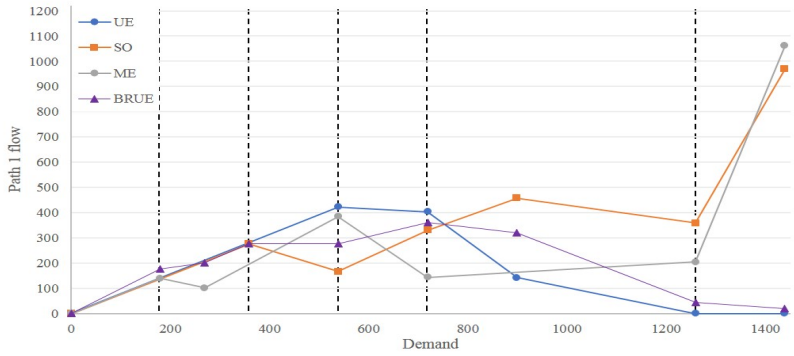
FIGURE 5.3 – Network of Lyon 6

distribution for breakpoint detection with all-or-nothing initialization and $\epsilon = 0.1C_{p,\tau}$.

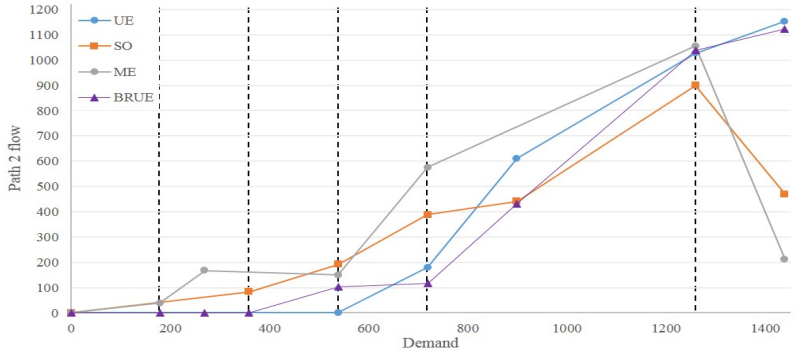
The solution space is not continuous because of the trip-based simulation (flows have integer values). Therefore, for each experiment, we have three integer numbers as the flow on each path in the vector of assignment. In order to represent and analyze the breakpoints in a continuous space, we draw the flow distribution diagram by making a (piece-wise) linear regression with $R^2 < 0.9$ on the integer data. The demand-flow diagram for each path is presented in Figure 5.4(a)-(c). Moreover, the results of breakpoint detection are presented in Figure 5.4(d). In each experiment, each equilibrium is calculated and the breakpoint occurs when at least one path enters or exits the active path set of one equilibrium. The breakpoint analysis will be carried out by comparing one equilibrium from travellers' point of view (UE or BRUE) and one equilibrium from system point of view (SO or ME).

5.5.2 Breakpoint analysis

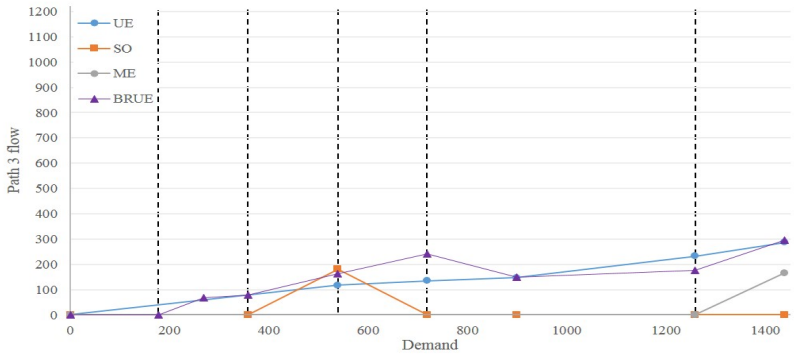
According to the Figure 5.4, the SO active path set contains paths 1 and 2 at low demand level until the first SO breakpoint is reached at the demand level of 359 users where path 3 enters the active path set. Then, path 3 exits the active path set at a demand level of 719 users where the second breakpoint is detected. The active path set remains the same for all demand levels bigger than 719 users. Therefore, SO path flow distribution has two breakpoints. On the other hand, UE path flow distribution has two breakpoints at a demand level of 539 travellers where path 2 enters to the active path set and at a demand level of 1259 travellers where path 1 exits the active set; path 1 is not used beyond this level. Consequently, the breakpoint analysis for UE-SO contains 4 breakpoints. It is remarkable that path 3 is not being used at most demand levels in the SO path flow distribution but it always belongs the active path set of UE for all demand levels.



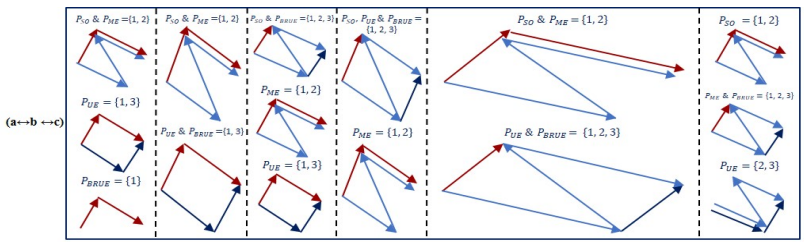
(a) Path 1 flow-demand diagram.



(b) Path 2 flow-demand diagram.



(c) Path 3 flow-demand diagram.



(d) Active path sets between breakpoints for four equilibria.

FIGURE 5.4 – Paths flow-demand diagram for UE, BRUE, SO and ME. Breakpoints are presented by black vertical dash lines on total demand axis.

The ME path flow distribution is obtained by applying UE discipline to users of all OD pairs except the users of the specific OD pair (Figure 5.3(b)) who travel under SO discipline. ME has paths 1 and 2 in the active path set until the only breakpoint of ME where path 3 enters the path set. According to the UE breakpoint detection, the analysis of UE-ME breakpoint involves two breakpoints. Path 3 is not used at most demand levels as in the SO path flow distribution.

As mentioned previously the BRUE scenario is one instance from BRUE solution set. There are two breakpoints in this scenario. The first one is where path 3 starts to be used and the second one happens where path 2 enters to the active path set. All three paths are used beyond the second breakpoint. There are 3 breakpoints with respect to the SO breakpoints for BRUE-SO breakpoint analysis. Note that in BRUE, path 3 is used at all demand levels beyond the first breakpoint. In the BRUE-ME breakpoint analysis, there are three breakpoints. The active path set for both equilibria is same when demand is below the first breakpoint where path 3 enters the BRUE active path set. Path 2 enters the active path set of BRUE at the second breakpoint and the last breakpoint is the BRUE breakpoint.

Here, we focus on one specific OD pair to detect the breakpoints in the real network. We compare the breakpoints of different equilibria. We identify the range where the active path sets of two equilibria are different. This range is used for applying control strategies in order to push the system from the current state to SO.

5.6 Control strategy

The idea of shifting the network from one equilibrium to another requires applying a control strategy in the demand range between two breakpoints, where the active path set remains fixed. When both equilibria have the same path set but different flows, we need to design a strategy to reroute the critical users. The number of critical users, in this case, is the difference between the two flow values of the paths. For instance, in this network, the pattern of each equilibrium (set of active paths) is constant between two breakpoints. The traffic management system can induce a change of path for the critical users. Also, when the difference in flow is high for two equilibria, then promoting users to use one specific path may also be efficient. On the other hand, if the active path set is not the same for the two equilibria (e.g. paths used in UE which are not used in SO), the system can ban some paths (unused in SO) through routing advises. In this chapter, we highlight the situation when one path is not used in SO or ME while it is being used in UE or BRUE. Note that it is important to consider which discipline is used with respect to a control strategy. If we consider the ME discipline as a reference, the range of banning or the path involved in banning may change. Here, we want to investigate which discipline would be a reference for control strategy? And does banning improve the network performance? The answer to the latter question is connected to the Braess paradox.

In these experiments, according to the breakpoint detection (Figure 5.4(c)) when the demand level is below 359 and when it exceeds 719, the SO flow for the third path is zero. For ME, when the demand is higher than 1259 the flow of path 3 is not zero. Therefore, if we ban path 3 for UE and BRUE scenarios, users will use the two other paths and the performance of the network is changed. The goal of the control methods is to improve the total travel time of the network. We will check that banning is efficient in those ranges of demand

where path 3 is unused in SO or ME. Here we present the ban strategy that we apply to the test case.

5.6.1 Ban Strategy (BS)

Banning some links or some turning movements at intersections and optimizing traffic lights' settings are major tools that traffic managers use in order to improve the performance of a traffic network. Here based on the breakpoints analysis we apply the BS to prevent users of specific OD pair(s) from taking a path which has zero flow in the SO/ME framework and non-zero flow in the UE/BRUE solution. Such a strategy may also stop users from other OD pairs to use the banned link or turning at the specific intersection in the network. Therefore, we can have side effects because the route choice of other users is also affected.

In order to choose the best location banning point, we first list all possibilities and then count how many users from other OD pair use that intersection or link. Then we choose the banning point that affects fewer users from other OD pairs. Note that all the simulations with BS are executed in the UE/BRUE framework for all users in the network. We consider one banning point which is presented in Figure 5.5 which prevents the users of the "Quai de Serbie to Verguin" OD pair from using path 3. The banning point prevent all users from going straight forward at the intersection.

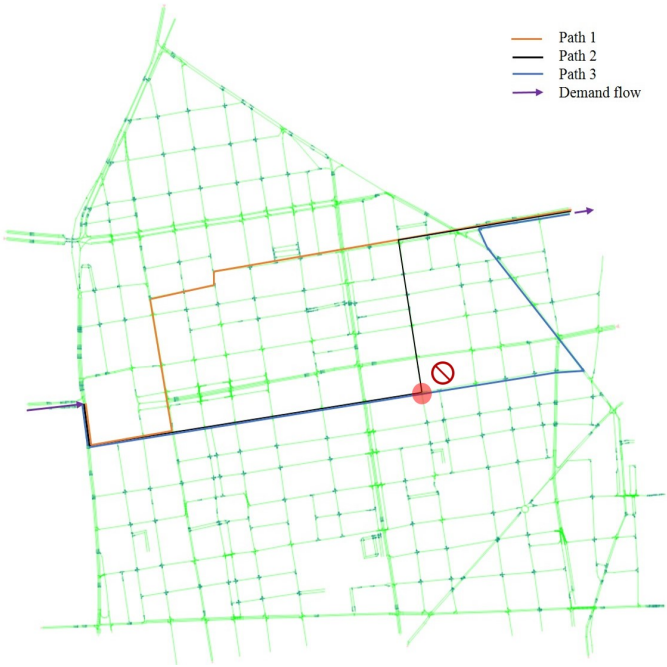


FIGURE 5.5 – Network of Lyon 6 with banning point

5.6.2 Applying BS to one OD pair test case

For the "Quai de Serbie to Verguin" OD pair, we consider one BS based on SO and ME because with both disciplines there are approximately same zone between breakpoints that the level of demand in ME and SO is zero for path 3 and not zero in UE/BRUE framework. In

other words, the BS optimal ranges of demand (identifying by breakpoints) for both references (SO and ME) are similar. We apply the banning strategy for all demand levels to see the impact of the strategy on the different demand levels. Therefore, path 3 is banned for all levels of demand. We remark that the BS scenarios are run in UE framework. The BRUE is also considered for all ban scenarios (BRUE+BS) in order to compare with the BRUE scenario without ban strategy.

The total travel time of the breakpoints for all equilibria are presented in Figure 5.6. We present it here in order to first show that optimizing the surrogate model instead of the actual value of the path marginal still yields a better total travel time than UE and BRUE. Second, in the mixed equilibrium the rerouting of a specific OD pair improves the performance of the network but it also shows that when users of one OD pair switch to the SO solution, other users take advantage and use the capacity of those links which are less used. Third, the total travel time of BS is below the total travel time of UE. Even the BRUE+BS obtains better performance than BRUE. For ban strategies, we also close a link of path 3 between two breakpoints where path 3 is used in SO, and the result shows that the total travel time of this level is higher than UE (dash lines in Figure 5.6).

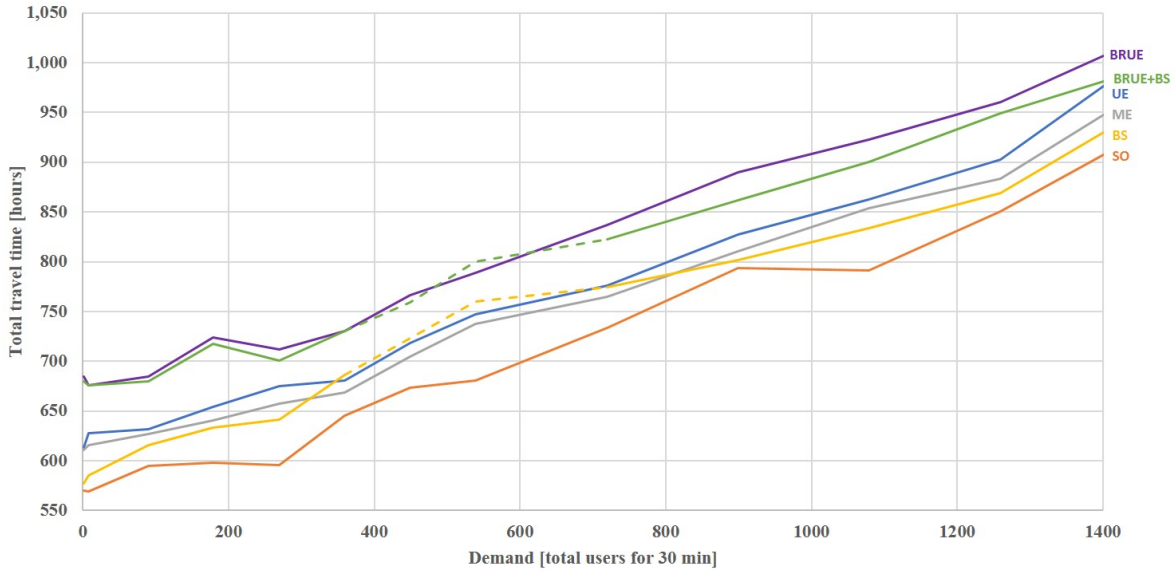
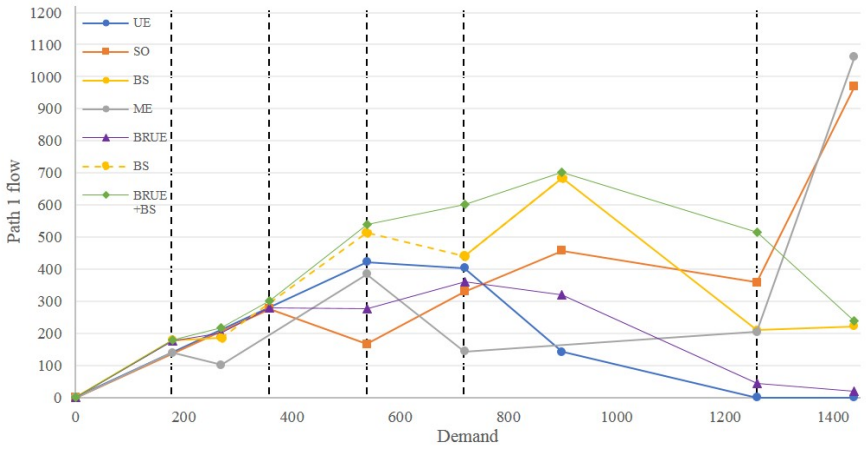


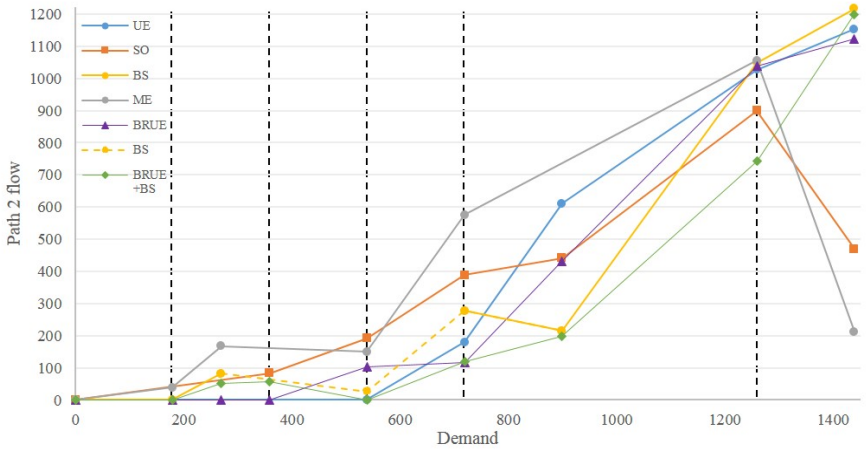
FIGURE 5.6 – Total travel time for each level of demand in one OD pair test case for UE, SO, BRUE, ME, BS and BRUE+BS.

The results of all experiments for one OD pair test case are presented in Figure 5.7. Note that the ban strategy is applied for all demand level so the path 3 flow of BS and BRUE+BS in Figure 5.7(c) is zero for all demand levels. At the demand level at which we should not apply BS, flows and travel time are presented by dash lines for BS. The dash lines in Figures 5.7(a) and 5.7(b) show that between demand levels 359 and 719 the flow assigned to path one and the flow level is higher than SO solution. The BS improves the total travel time also after the last breakpoint where the SO solution guides us to ban path 3 and ME contains this path in the active path set. The result for one OD pair test case shows that considering the SO solution as the reference can provide a better control strategy than ME solution in order to improve the total travel time of the system. Also, because all users in BS/BRUE+BS are looking for UE/BRUE, we do not expect the total travel time for the whole network to be equal to SO travel time. But the result shows that the performance of

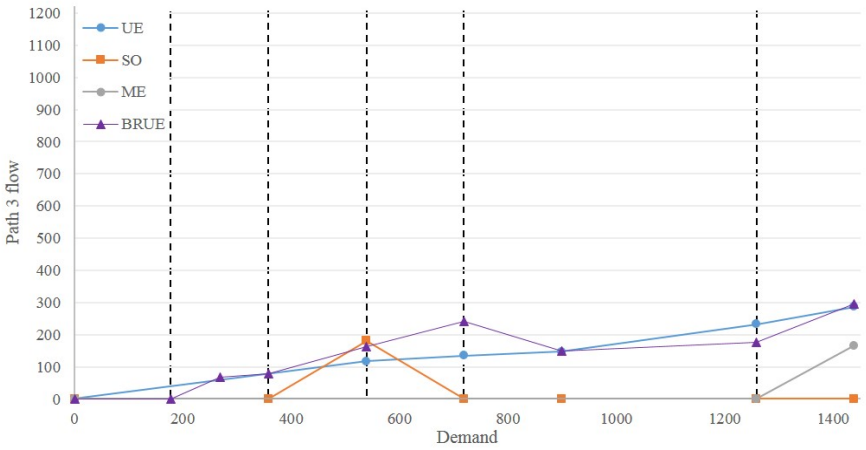
the network is improved with BS in UE and BRUE frameworks. If we would be able also to apply BS to more than one OD pair flow based on the breakpoint analysis we would expect better performance in term of total travel time from the traffic network.



(a) Path 1 flow-demand diagram.



(b) Path 2 flow-demand diagram.



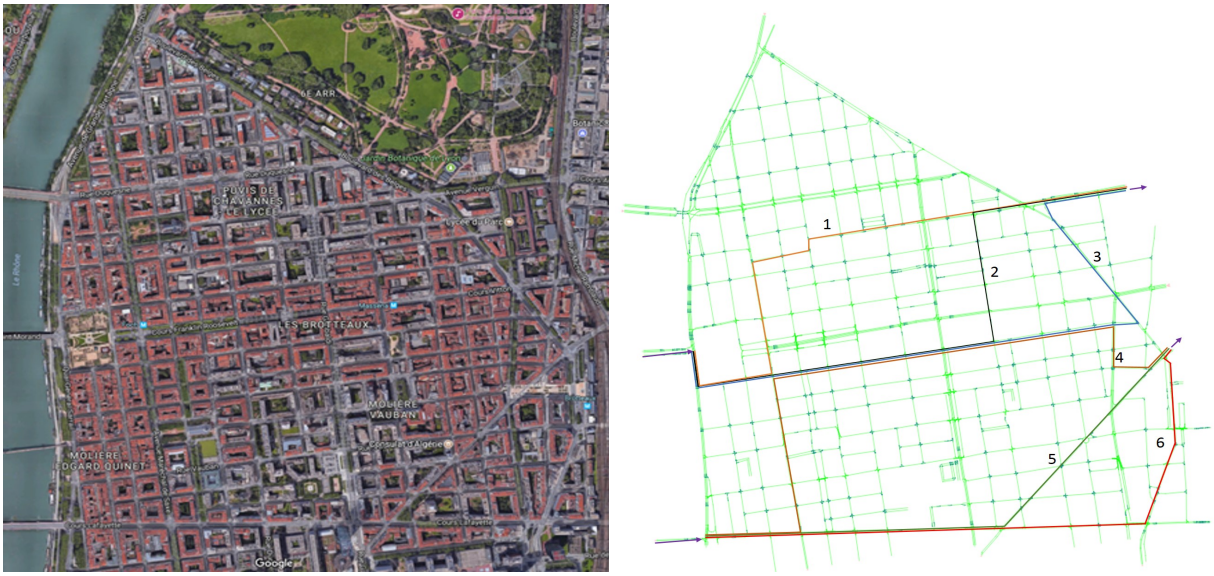
(c) Path 3 flow-demand diagram.

FIGURE 5.7 – flow diagrams in one OD pair test case for UE, SO, BRUE, Mixed Equilibrium (ME), Ban strategy (BS) and the combination of BRUE and Ban Strategy (BRUE+BS). Breakpoints are presented by black vertical dash lines on total demand axis.

5.7 Two OD pairs numerical experiments

In this section, we apply the breakpoint analysis and the ban strategy based on the breakpoint analysis to the second test case considering jointly two OD pairs in order to show that our framework can be applied to more general configurations. We first identify what are the two most problematic OD pairs in terms of congestion (carry the most demand). Recall that the main idea is through the banning we try to improve the traffic condition of the full network by focusing on these OD pairs. We trigger our banning based on the evolution of demand on these OD pairs.

We consider an experiment which contains two OD pairs in order to analyze the breakpoints by considering the interaction of the two OD pairs. The Lyon 6 network, including the two OD pairs test case, is presented in figure 5.8. There are 3139 trips for all OD pairs excluding the two targeted OD pairs (for more details see B). In order to find the breakpoints, the demand levels of both OD pairs are increased at the same time because both demands come from the same direction and ME discipline is also applied for the users of these OD pairs while the other users follow the UE/BRUE framework. Figure 5.8 shows that the two OD pairs have many links in common. Taking this fact into consideration, we try to choose the banning intersection in a smart way based on the breakpoint analysis.



(a) Satellite view of Lyon 6, France (Mapping data ©Google 2019)

(b) The paths on the network for two OD pairs test case with banning points

FIGURE 5.8 – Network of Lyon 6

5.7.1 Breakpoint analysis

The flow-demand diagrams of all six paths are presented in Figure 5.9. The demand level of both OD pairs is increased at the same time and the breakpoints are detected from the path flows in order to identify the range of demand for applying the ban strategy. The demand axis shows the inflow of both targeted OD pairs which are equal and increase together for

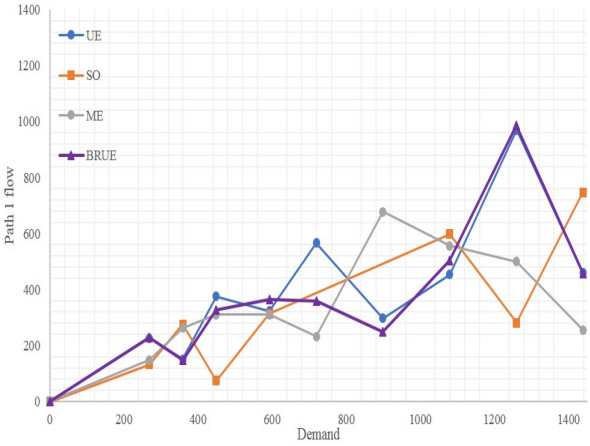
each simulation. The 3 figures on the right (Figures 5.9(a),(c) and (e)) are related to the first OD pair. They show the breakpoints which are changed because of the impact of the demand level of the second OD pair. The third path is used in the SO solution when the demand level of both OD pairs exceeds than 362. It shows the impact of OD pairs on each other when the demand level is changed. For the second OD pair the results in Figures 5.9(b),(d) and (f) show the opportunity of banning strategy for path 4 which is almost not used for SO solution. Path 3 and path 4 have many common links and we observe that these links are more used for the UE discipline of the first OD pair which shows that these links are critical for breakpoint detection and ban strategies.

The breakpoint detection is carried out for each equilibrium by considering the active path set and the ban strategy is designed based on the breakpoint analysis. Here, we do not represent the process which is same as in subsections 5.5.2 and 5.6.1 and just demonstrate the result of breakpoint analysis and BS designing. The ban strategies based on SO and ME for two OD pairs test case are presented in Table 5.3. For instance, when the path flow is zero for ME and not zero for UE/BRUE, we ban that flow until the next breakpoint that this path is used.

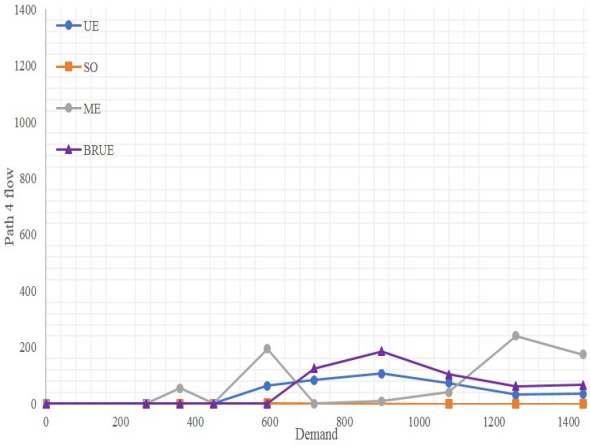
TABLE 5.3 – The ban strategies for two OD pairs test case

Ban Strategy based on system optimum (BS1)			Ban Strategy based on mixed equilibrium (BS2)		
starting breakpoints	ending breakpoints	Ban	starting breakpoints	ending breakpoints	Ban
0	359	3, 4 and 6	0	269	3, 4 and 6
359	1439	4 and 6	269	539	3 and 6
-	-	-	539	1439	4

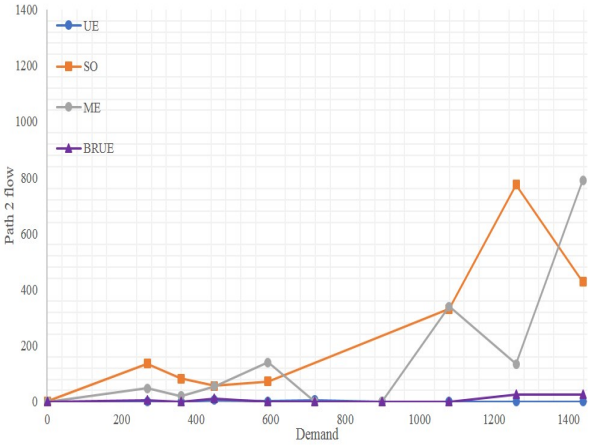
For this case, we consider two ban strategies. The first BS (BS1) is applied based on the SO and the second one (BS2) is designed based on ME path flow distribution (Table 5.3). Four banning points are chosen in order to implement the BS which are presented in Figure 5.10. Point α is same as in the first scenario and we ban path 3 for the first OD pair and path 4 for the second OD pair. The second point is point β which prevents users from turning left at the intersection and bans path 3 only for the first OD pair. Point γ prevents users from turning left and ban path 4 and the last banning point δ is forbidding users to go straight forward and use path 6. These four banning point(s) are activated based on the breakpoint analysis. When the flows on both paths 3 and 4 are zero in the SO solution, point α is activated and when the flow of path 3 is zero, point β becomes activated. The point γ allows us to ban path 4 and prevents users from turning left at the intersection. Finally point δ becomes active when we need to ban path 6 in order to prevent users from moving straight at the intersection. Considering these banning points we are able to apply the designed ban strategies in each range of demand.



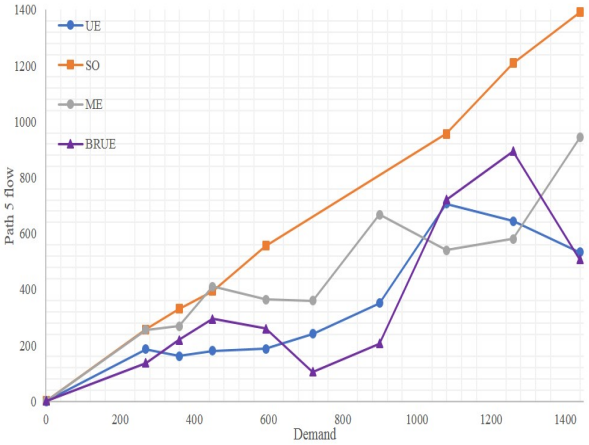
(a) Path 1 flow-demand diagram.



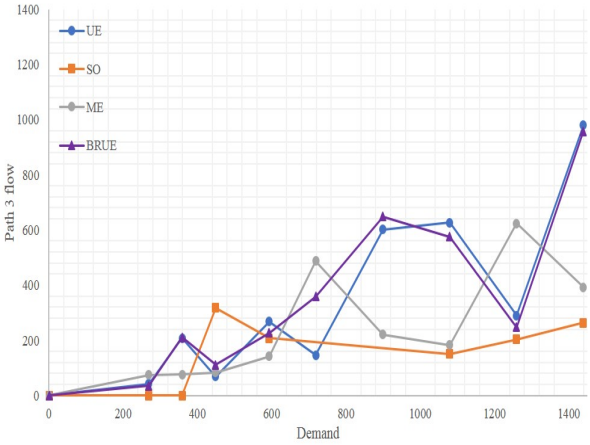
(b) Path 4 flow-demand diagram.



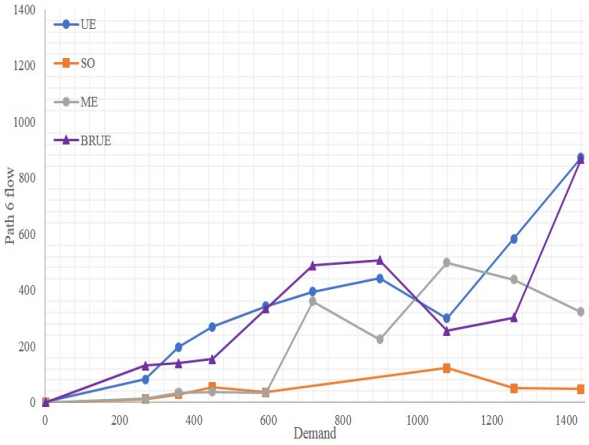
(c) Path 2 flow-demand diagram.



(d) Path 5 flow-demand diagram.



(e) Path 3 flow-demand diagram.



(f) Path 6 flow-demand diagram.

FIGURE 5.9 – flow diagrams in two OD pairs test case for UE, SO, BRUE, Mixed Equilibrium (ME).

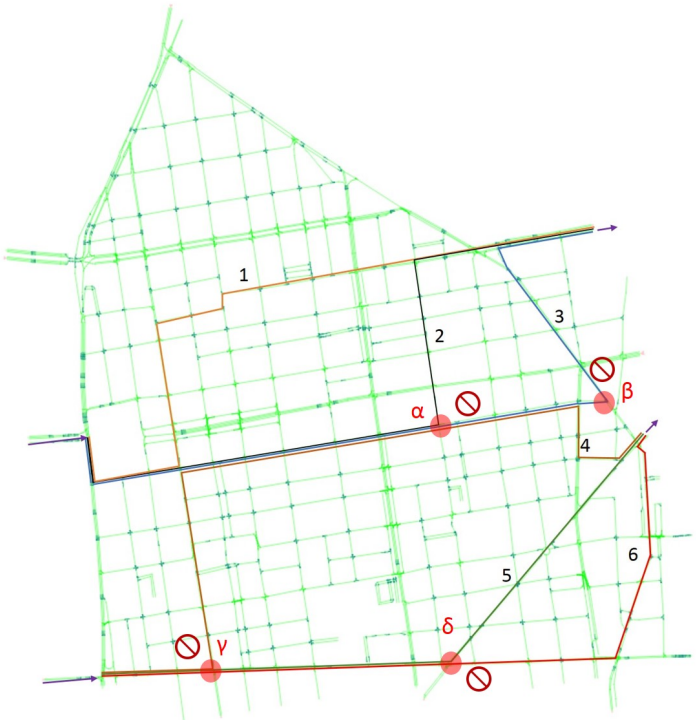


FIGURE 5.10 – Network of Lyon 6 with banning points for two OD pairs scenario

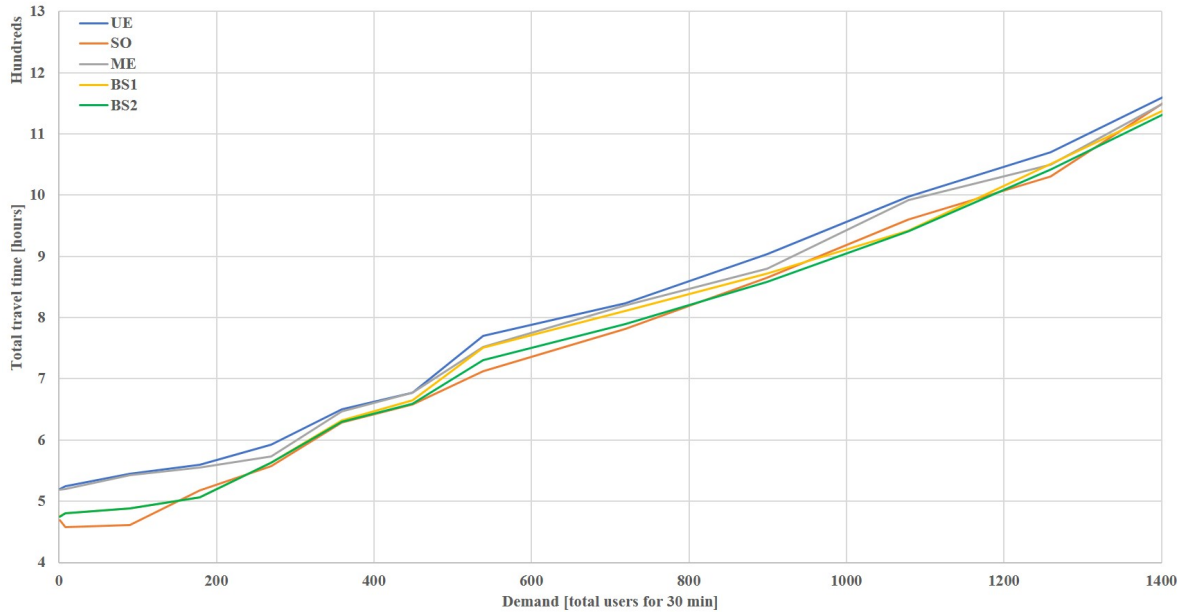
5.7.2 Applying BS to two OD pairs test case

The results of the Ban strategies are provided for the two OD pairs scenario. Figure 5.11 presents the evolution of the total travel time at different demand levels of the two OD pairs for different disciplines. We present the application of ban strategy in the UE and BRUE frameworks. "BS1" and "BS2" correspond to the UE framework in Figure 5.11(a) and "BRUE+BS1" and "BRUE+BS2" in Figure 5.11(b) present the total travel time of ban strategies in BRUE framework.

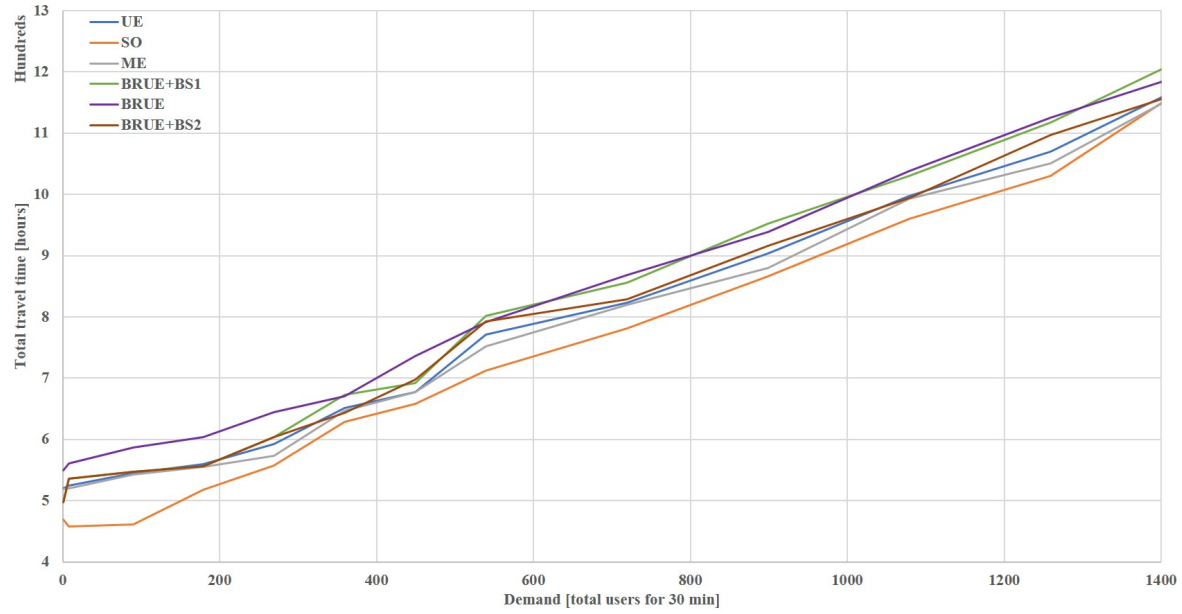
The total travel times for UE, SO, ME, BS1 and BS2 are shown in Figure 5.11(a). The figure shows that BS2 works better than BS1 in the two ODs test case, but we should consider that both strategies improve the network performance. BS1 and BS2 also change the network design. Therefore, there is a possibility, while applying BS, to obtain a total travel time lower than the SO total travel time in some ranges of demand. This has happened in two ranges of demand: $[813 - 1187]$ and $[155 - 241]$, where the total travel time of BS is lower than the total travel time SO. The second reason for getting lower total travel time with BS than with SO could be using the surrogate model instead of the analytical path marginal cost in SO calculation. The surrogate model only yields an approximate SO. But the results shows that considering the approximation of the SO solution as a reference is enough to design a control strategy based on breakpoint detection in order to improve the performance of the dynamic traffic network.

The second figure (Figure 5.11(b)) presents the evolution of the total travel time for reference disciplines (SO and ME) and BRUE discipline combined with two ban strategies. BRUE+BS1 is the BS based on SO and the second one (BRUE+BS2) is the BS based on ME. ME obtains a better solution for designing the ban strategy in the context of BRUE discipline.

The BRUE+BS2 solution is close to UE and better than BRUE in term of total travel time. The application of ban strategies to the second test case shows that this strategy can improve the network performance also when we consider more than one OD pair with common links.



(a)



(b)

FIGURE 5.11 – Total travel time for each level of demand in two OD pairs test case for UE, SO, BRUE, ME, BS1, BS2, BRUE+BS1 and BRUE+BS2.

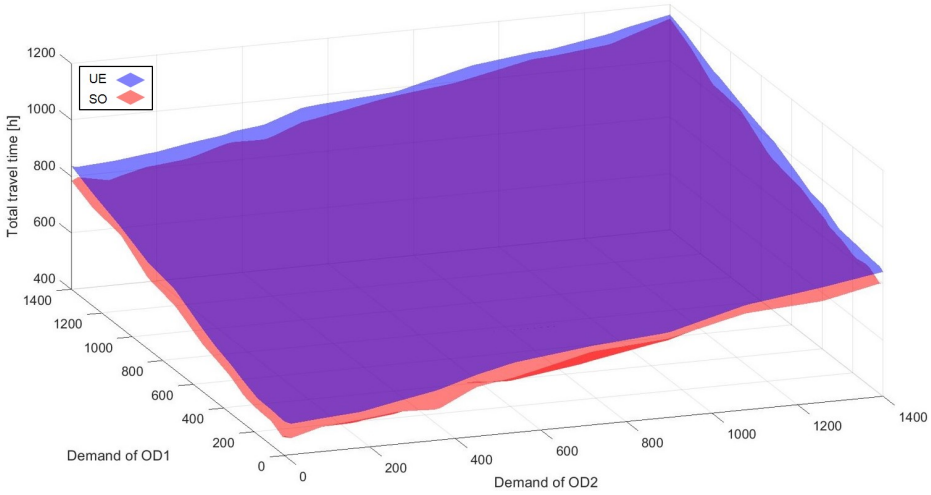
5.7.3 Sensitivity analysis

For now, breakpoints for two OD pairs have been derived when the demand level is similar on both side. This situation is quite realistic because, during network loading, we expect that the demand grows in a coordinated way. However, we want to investigate in this subsection what happens if we relax this hypothesis and have different demand levels for each OD pair.

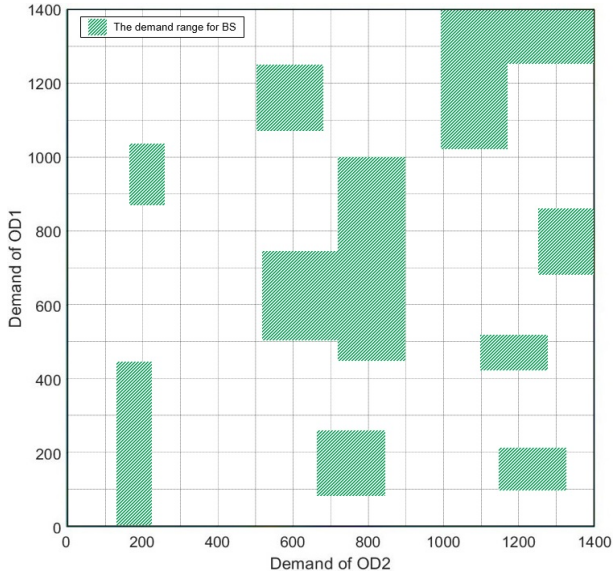
First, we run simulations to derive UE and SO equilibrium on the network for any demand levels on both OD pairs. Figure 5.12(a) presents the total TT for UE and SO conditions and all demand levels. We draw the planes by using the linear interpolation method (Blu *et al.*, 2004). The total TT in SO is obviously less than UE for all demand levels. Second, we look for breakpoints and path 4 as previous results in section 5.7.1 show that banning this path at point γ (figure 5.10) significantly improves the system performance when the demands very coordinated on both OD pairs. In other words, path 4 is the most promising path for the application of banning strategy. The breakpoint detection is carried out by comparing UE and SO solutions and looking for situations when path 4 should not be used in SO while being used in UE. Figure 5.12(b) presents the demand ranges when the banishment should be applied to path 4 based on such a breakpoint analysis. Note that we look for rectangle areas when all conditions are met and we did not do fine tuning of the area shapes.

To assess how BS is effective, we first ban path 4 for all demand levels and run again the simulations considering UE equilibrium. We expect that the BS improves the network situation compared to UE without banning within the ranges of demand we have previously identified based on the breakpoint analysis. Figure 5.13 presents the total TT of UE and UE + banning path 4 (BP4). Two deformed planes are overlapping each other in different demand levels. When the UE plane is placed at the top of BP4 plane, it means by banning path 4 we reduce the total TT.

Figure 5.14(a) presents the comparison between BS plan (from Figure 5.12(b)) and the results of banning path 4 for all demand levels of two targeted OD pairs. The green regions are where the system has better performance (less total TT) than UE without banishment, i.e., the banishment is effective. The red regions are the demand ranges where the BS cannot improve (decrease) the total TT of the network. Besides, the heat map in figure 5.14(b) shows how many hours can be saved or lost by banning path 4 at each level of demand. The results in figure 5.14 show that breakpoint analysis properly cover the ranges of demand when the BS is effective. Futhermore, it provides a full coverage of the region when the BS is the most effective, i.e. save the more hours, see red regions in figure 5.14(b).



(a) The total TT of UE and SO for different demand levels of joint OD pairs



(b) The output of breakpoint analysis

FIGURE 5.12 – Breakpoint analysis on path 4 for two targeted OD pairs

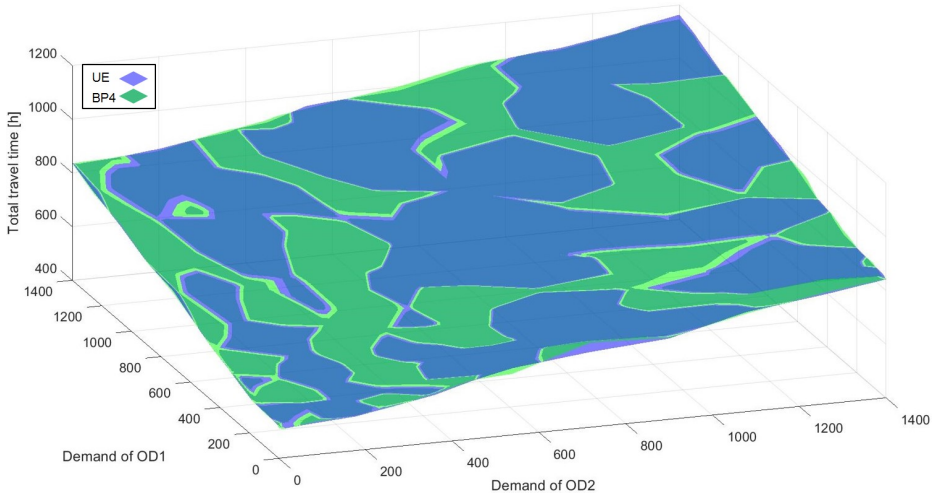
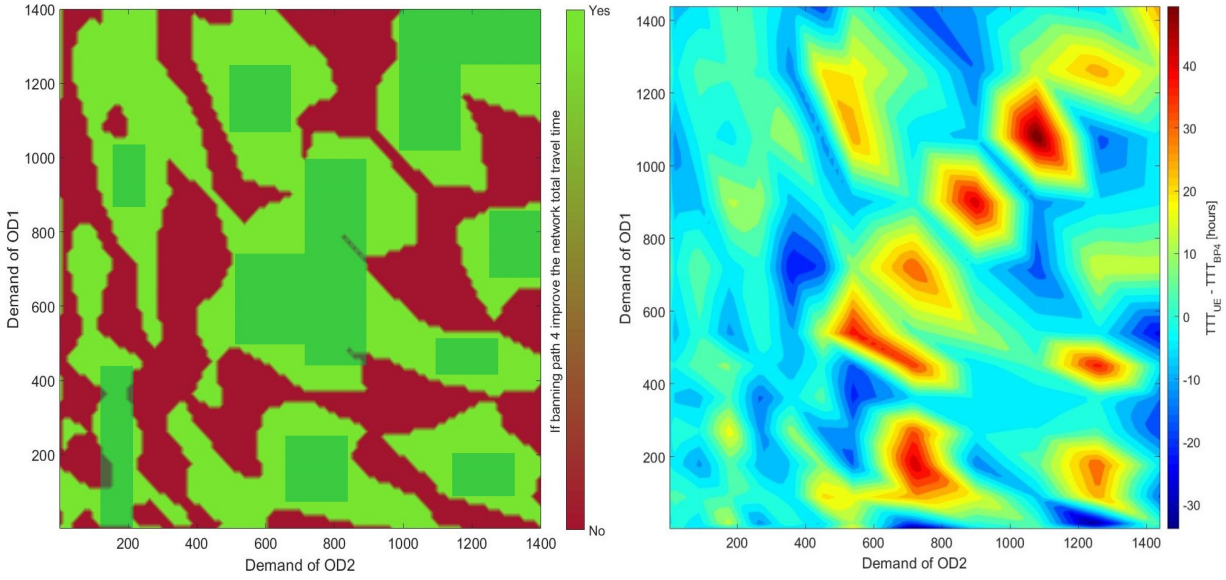


FIGURE 5.13 – The total TT of UE and BP4 for different demand levels of targeted OD pairs



(a) (b)

FIGURE 5.14 – Comparison between BS by breakpoint analysis and the result of banning path 4 for all demand levels of two targeted OD pairs: (a) The comparison between BS plan by breakpoint analysis and the results of banning path 4 for all demand levels of two targeted OD pairs; (b) The difference between UE Total Travel Time (TTT) and the TTT when path 4 is banned for all demand levels of two targeted OD pairs.

5.8 Conclusion

This chapter investigated the impact of different demand levels on three equilibria (User Equilibrium, System Optimum, and Boundary Rational User Equilibrium). It defined a breakpoint as a demand level where we observe a change in the active path set of one equilibrium. This study attempted to find the breakpoints and to investigate the possibility to use breakpoint information in order to move from one equilibrium (UE/BRUE) to another (SO). In the static case, we first introduced the process of breakpoint detection and then demonstrated the identification of the situation in which we can improve the network performance by using the ban strategy in the Braess paradox context. For the dynamic case, this study proposed a novel approach to analyze network DTA equilibrium as a function of demand level. The potential implications of this approach for network suppliers concern the analysis of the network status and the design of ban strategies to move from an initial UE or BRUE situation towards SO.

The numerical experiments were conducted on a dynamic real sub-area network Lyon 6 in order to examine the equilibrium patterns at different demand levels. We considered the full network equilibrium and analyzed the pattern when the demand of either one, or two, origin-destination pairs is changed. Two control strategies are applied based on the breakpoint detection in order to evaluate their impact. The results showed that in the dynamic case, the banning strategy is efficient, and we should apply it in the right range of demand. The mixed equilibrium strategy is designed in order to consider a second reference for designing the ban strategy. The results showed that both references could help the design process, and the ban strategy improves network performance. Finally, we have done a sensitivity analysis on the effectiveness of banning a potential path for different demand levels of two OD pairs. The results show that the breakpoint analysis is a powerful tool in order to detect the demand ranges wherein the banning strategy is effective. Note that the experiments of this chapter have been carried out with homogeneous users. The approach in this chapter can be carried over to heterogeneous users and can consider the profile of each user, which allows to consider the BRUE in the stable form.

The numerical experiments showed that approximating the marginal cost and estimating SO solution yields a breakpoint-based control strategy which improves network equilibrium, so there is a possibility to get a better total travel time by improving the approximation of SO. Thus it will also be possible to investigate if what we observe here in Section 5.7.2 (BS total travel time lower than SO total travel time) is caused by a non-optimal solution or related to network effects (e.g., correlations of the effects between multiple OD pairs). Finally, the results of this chapter show the existence of the breakpoints, which is to be expected from the static case analysis but has not been analyzed in the dynamic case before. They also show the efficiency of the banning strategy, particularly based on mixed equilibrium discipline at the proper level of demand.

GENERAL CONCLUSION

Summary and global overview

This PhD thesis purposed dynamic network loading process, and in particular, path flows calculations for different network equilibrium conditions. Our research has been entirely focused on the simulation-based DTA model, for which a rich and prolific literature has been produced over the past decade, as shown in the introduction chapter and the beginning of each following chapter of this manuscript. Simulation-based DTA models can be applied to large-scale urban networks. However, despite the growing attention paid to simulation-based DTA modeling, the literature review of our introduction chapter unveiled several theoretical and methodological issues in the solution algorithm for trip-based dynamic network equilibrium. The reason lies in the fact that there is no comprehensive comparison of the performance of the existing solution algorithms, and also, there are few studies on applying solutions methods from other fields of optimization. Most existing solution methods are often simple to implement to solve the DTA problem, but they are not fast enough and do not guarantee the quality of the solution, particularly for large-scale networks. Not only the calculation of the network equilibrium is crucial, but analyzing the equilibrium path flow distribution is also important. Studying path flow distributions for UE solution has not received lots of attention because lots of works are dedicated to finding the optimal solution.

In this context, the objectives of our work were two-fold: (i) first, improving the DTA calculation algorithms, and (ii) second, using the algorithm to solve new DTA problems and innovate in analyzing the DTA outputs to improve the transportation system performance. Parallel to these two objectives, the storyline of this manuscript can be seen as the introduction of optimization framework from calculating the equilibrium as fast as possible (part I) to optimize the system based on equilibrium analysis (part II) for DTA models. For the first objective, our main contributions were cross-comparison of the existing algorithms and developing an efficient framework based on meta-heuristic algorithms to find the traffic network equilibrium. These questions were mainly in the topics of chapters 2 and 3 (part I). In chapter 2, first, the existing solution algorithms were investigated, and several improvements proposed to improve the performance of solution algorithms compared to the recent methodology in the literature. This investigation gave a good background for designing a new framework for large-scale traffic network. As described in chapter 3, the fundamentals of the most existing algorithms are similar. Therefore, another significant contribution of present work was introducing a new approach based on parallel computation for solving

the DTA problem. Two meta-heuristic algorithms (Genetic and Simulated annealing) were designed and adapted to the problem. Both of them reduced the computation time and also provided better solutions in terms of closeness to the optimum than existing algorithms. The parallel computation framework empowers the solution algorithm to explore more points in the solution space.

All the efforts in part I aimed to provide an efficient tool to calculate the equilibrium. For the second objective, our main contribution was innovating in analyzing the equilibrium solution inspired by real urban network settings. In part II, two methodologies are developed in order to improve the performance of the transportation system. The first one corresponds to the features of the equilibrium solution. One of the key features is unicity of the equilibrium solution, which is simply violated from the theoretical point of view in multimodal and urban traffic networks. As shown by simulation in chapter 4, multiple equilibrium solutions exist in large-scale networks. Moreover, the impact of network design history on day-to-day DTA problem was investigated, where we have multiple solutions for equilibrium. This question is addressed and discussed in chapter 4. Another key result that was obtained from the dynamic simulations is that certain final equilibria were more efficient from the systems point of view than other equilibria.

The second methodology was an innovation in the equilibrium analysis to design control strategies. This question was the subject of chapter 5. The breakpoint concept was introduced in order to design a novel approach to analyze the path flow distribution of different network equilibrium rules (e.g., user equilibrium and system optimum) as a function of demand level. This analysis is able to find the demand levels, where the control strategies can be effective for system improvement. Using dynamic simulator showed the existence of the breakpoints, which has not been analyzed in the dynamic case before. Application of one control strategy (banning) at the proper level of demand also proved the advantage of the breakpoint analysis.

Finally, the practical output of our work was the development of the SymuMaster as an optimization platform presented in chapter 1.2.3. It contains all of the implementations that have been done in this thesis, particularly one of our notable contributions to the field of DTA simulation, namely the implementation of the meta-heuristic algorithms with parallel computation for simulation-based DTA, operating in both flow-based and trip-based settings. From a broader research perspective, this platform is portable and has been connected to the different simulators and also used with other members involved in the **MAGnUM** project. SymuMaster is going to be available open-source as part of the publication of the Symuvia platform.

Contributions to our initial research questions

According to the list of the research questions that was presented in section 1.5.1, our main contributions are listed below in the same order:

- The full benchmark of all algorithms has been done for different network size and level of saturation in chapter 2. The performance of the algorithms is evaluated based on the quality of solutions and computation times. Several improvements are also proposed to speed up the convergence. All the extensions significantly reduce the number of iterations to get a good convergence rate and drastically speed up the overall simulations.

The results also show that the performance of different components of the solution algorithm is sensitive to the network size and saturation. Finally, the best configurations of the solution algorithms have been identified for all network sizes, with a particular focus on the largest scale.

- About designing a new generation of solution methods, chapter 3, first, highlighted the drawbacks of existing algorithms for DTA models. Second, for the first time, parallelized meta-heuristic approaches were applied to solve the network equilibrium problem. Two new solution methods were proposed: an extension of the simulated annealing and an adapted genetic algorithm. With parallel simulation, the algorithm runs more simulations in comparison with existing methods, but the algorithm explores the solution space better and therefore obtains better solutions in terms of closeness to the optimal solution and computation time compared to classical methods. The results showed that meta-heuristic algorithms dominate classical methods.

- In chapter 4, the assumption of unicity for DTA models was first discussed. Second, the relaxation of this hypothesis with the multi-modality on traffic dynamics were studied. Multiple solutions are first investigated analytically in the static test case and then examined on a real test case by long-term, day-to-day learning process. The results showed that not only non-unicity exists, but that total travel time can be saved and other network performance indicators optimized by switching from one solution to another.

- Chapter 4 studied the network design history in the day-to-day multimodal context where there are multiple equilibria. When multiple facilities are progressively introduced in the system at different times, the learning process is subject to multiple steps. When users have time to adjust to these different steps, it changes the global convergence process and may lead the system towards multiple different situations while the final network setting remains the same. Based on static and the dynamic settings, it was demonstrated that the order of the successive introduction of such facilities matters when determining the final equilibrium. This is a crucial finding, which means that the study of the current network situation may not be sufficient to grasp the real user distribution inside the network and that it is necessary to consider the history of the network. In other words, a unique UE calculation with the current network setting may lead to an equilibrium other than the one resulting from the different steps corresponding to the network history.

- The analysis on the solution of three popular equilibrium conditions: User equilibrium (UE), System optimum (SO) and Boundary Rational User Equilibrium (BRUE) has been done in chapter 5 considering static and dynamic traffic assignment. Here the novelty of our approach for analyzing the equilibrium solutions would lie in introducing the new concept: demand level breakpoint. A breakpoint is the demand level where we observe a change in the active path set of one equilibrium. It was used in this study as identification of the situation in which there is an opportunity to improve the network performance. Following the concept of Braess paradox, breakpoint detection permitted to design banning strategies at some key locations in the network to prevent some alternatives from being used and thus to improve the system performance. Dynamic simulation on a real test case showed that such a strategy is effective, which

demonstrated the importance of breakpoint identification. The potential implications of this approach for network suppliers concern the analysis of the network status and the design of ban strategies in order to move from an initial UE or BRUE situation towards SO.

Research perspectives

In the following part, we propose recommendations for future research:

- First of all, improving the existing solution algorithms in the literature for equilibrium calculation is not completed by this study, and there is still room for further improvements. For instance, designing the framework to predetermine the computation budget, based on network size, topology, and saturation level can be an interesting topic.

- In addition to improving the path flow calculation, identifying the component of the problem which has more impact on the computation time and the quality of end-result for different algorithms is also crucial. Therefore, testing the performance of algorithms on more detailed models, e. g., activity-based models can be worthy.

- Other comparison studies should definitively be carried on to further validate and investigate the meta-heuristic approaches we introduced. Indeed, although we tried to keep them quite general, one major issue with the tests we conducted is that we applied the algorithm to specific congestion patterns on the large-scale network. Therefore, it is recommended to test the algorithm on other networks and loading profile.

- This study has introduced a new branch of optimization algorithms for simulation-based DTA models. Therefore, designing other types of meta-heuristic algorithms is undoubtedly a very promising direction of research when attempting to overcome the curse of dimensionality related to large-scale DTA problems.

- Regarding the wide literature on solving fixed-point problems and variational inequality, there are still many solution methods in the field of game theory and operation research that can be applied to DTA problems.

- Here, we introduced the first parallel computation framework to solve DTA problems. It is also interesting to investigate other effective parallel and distributed algorithms or computing platforms in order to improve the computation time of large-scale problems. Integrating this approach with the path discovery and simulation steps can be an interesting research direction.

- Investigating other approaches to shorten the running time is also worthy. For instance, developing effective surrogate models for large-scale DTA simulation models.

- The application of hybrid or multi-objective variants of such meta-heuristics to solve simulation-based DTA problems with environmental considerations can be interesting also for future research.

- As it is proved mathematically in the literature and shown in the large-scale by this study, non-unicity and existing of multiple equilibria with different features are in the nature of urban traffic models. One of the research directions for future work can be building an equilibria prediction model to estimate the multiple possible equilibria of the system and control strategies to shift from one equilibrium to others.
- Another direction of research is considering a multi-objective equilibrium or a mixed equilibrium DTA, which is a challenging problem in day-to-day multimodal DTA models. The model is closer to the real world, but it is also more complex, and new sources of non-unicity can be correlated with other components of the system.
- As mentioned before, most of the research in the DTA field are stopped after calculating the DTA solution. Here, we analyzed the path flow distribution of equilibrium and found situations wherein we can improve the system performance by control strategies. We think that there is still lots of space to go further in better identification of these situations by introducing systematic methods.
- In this thesis, we tried to investigate how we can analyze and use the output of DTA models. Looking for the impact of initialization on the calculation of equilibrium for different demand levels, applying the breakpoint analysis process to more than two Origin-Destination pairs while considering correlations between paths and also testing other control strategies with breakpoint constitute some interesting topics for extending our study.

References

Aashtiani, H. Z. & Magnanti, T. L. (1981). Equilibria on a congested transportation network. *SIAM Journal on Algebraic Discrete Methods*, 2(3):213–226.

Abdel-Aty, M. A., Kitamura, R. & Jovanis, P. P. (1997). Using stated preference data for studying the effect of advanced traffic information on drivers' route choice. *Transportation Research Part C: Emerging Technologies*, 5(1):39–50.

Akamatsu, T. (2001). An efficient algorithm for dynamic traffic equilibrium assignment with queues. *Transportation Science*, 35(4):389–404.

Ameli, M., Lebacque, J.-P. & Leclercq, L. (2017). Multi-attribute, multi-class, trip-based, multi-modal traffic network equilibrium model. *The 12th Conference on Traffic and Granular Flow (TGF)*.

Ameli, M., Lebacque, J.-P. & Leclercq, L. (2018). Day-to-day multimodal dynamic traffic assignment: Impacts of the learning process in case of non-unique solutions. *7th International Symposium on Dynamic Traffic Assignment (DTA)*.

Askoura, Y., Lebacque, J.-P. & Haj-Salem, H. (2011). Optimal sub-networks in traffic assignment problem and the braess paradox. *Computers & Industrial Engineering*, 61(2):382–390.

Bar-Gera, H. (2002). Origin-based algorithm for the traffic assignment problem. *Transportation Science*, 36(4):398–417.

Beckmann, M., McGuire, C. B. & Winsten, C. B. (1956). Studies in the economics of transportation. Tech. rep.

Bekhor, S., Toledo, T. & Reznikova, L. (2009). A path-based algorithm for the cross-nested logit stochastic user equilibrium traffic assignment. *Computer-Aided Civil and Infrastructure Engineering*, 24(1):15–25.

Ben-Akiva, M., Bierlaire, M., Koutsopoulos, H. & Mishalani, R. (1998). Dynamit: a simulation-based system for traffic prediction. In *DACCORD Short Term Forecasting Workshop*, pages 1–12. Delft The Netherlands.

Ben-Akiva, M., Gao, W. Z., S. & Wen, Y. (2012). A dynamic traffic assignment model for highly congested urban networks. *Transportation Research Part C: Emerging Technologies*, 24:62–82.

Ben-Akiva, M. E., Gao, S., Wei, Z. & Wen, Y. (2012). A dynamic traffic assignment model for highly congested urban networks. *Transportation Research Part C: Emerging Technologies*, 24:62–82.

Bertsekas, D. & Gafni, E. (1983). Projected newton methods and optimization of multicommodity flows. *IEEE Transactions on Automatic Control*, 28(12):1090–1096.

Bliemer, M., Brederode, L., Wismans, L. J. J. & Smits, E. (2012). Quasi-dynamic traffic assignment: static traffic assignment with queueing and spillback. In *91st Transportation Research Board (TRB) Annual Meeting 2012*, pages 1–24. Transportation Research Board (TRB).

Bliemer, M. C., Raadsen, M. P., Brederode, L. J., Bell, M. G., Wismans, L. J. & Smith, M. J. (2017). Genetics of traffic assignment models for strategic transport planning. *Transport reviews*, 37(1):56–78.

Bliemer, M. C., Raadsen, M. P., Smits, E.-S., Zhou, B. & Bell, M. G. (2014). Quasi-dynamic traffic assignment with residual point queues incorporating a first order node model. *Transportation Research Part B: Methodological*, 68:363–384.

Blu, T., Thévenaz, P. & Unser, M. (2004). Linear interpolation revitalized. *IEEE Transactions on Image Processing*, 13(5):710–719.

Boyce, D., Lee, D.-H. & Ran, B. (2001). Analytical models of the dynamic traffic assignment problem. *Networks and Spatial Economics*, 1(3-4):377–390.

Boyles, S., Duthie, J., Melson, C. & Rambha, T. (2013). Diverge models and dynamic traffic equilibria. In *INFORMS Annual Meeting*.

Braess, D., Nagurney, A. & Wakolbinger, T. (2005). On a paradox of traffic planning. *Transportation science*, 39(4):446–450.

Cameron, G. D. & Duncan, G. I. (1996). Paramics—parallel microscopic simulation of road traffic. *The Journal of Supercomputing*, 10(1):25–53.

Cascetta, E. & Cantarella, G. E. (1993). Modelling dynamics in transportation networks: State of the art and future developments. *Simulation practice and theory*, 1(2):65–91.

CEBR (2014). The future economic and environmental costs of gridlock in 2030. Tech. rep., Centre for Economics and Business Research (CEBR) and INRIX.

Cetin, N. (2005). *Large-scale parallel graph-based simulations*. Ph.D. thesis, ETH Zurich.

Chevallier, E. & Leclercq, L. (2007). A macroscopic theory for unsignalized intersections. *Transportation Research Part B: Methodological*, 41(10):1139–1150.

Chevallier, E. & Leclercq, L. (2009a). Do microscopic merging models reproduce the observed priority sharing ratio in congestion? *Transportation Research Part C: Emerging Technologies*, 17(3):328–336.

Chevallier, E. & Leclercq, L. (2009b). Microscopic dual-regime model for single-lane roundabouts. *Journal of Transportation Engineering*, 135(6):386–394.

Chong, L. & Osorio, C. (2017). A simulation-based optimization algorithm for dynamic large-scale urban transportation problems. *Transportation Science*, 52(3):637–656.

Christidis, P., Rivas, J. N. I. *et al.* (2012). Measuring road congestion. *Institute for Prospective Technological Studies (IPTS), European Commission Joint Research Centre. Retrieved from http://ipts. jrc. ec. euro pa. eu/publications/pub. cfm*.

Colini-Baldeschi, R., Cominetti, R., Mertikopoulos, P. & Scarsini, M. (2017). The price of anarchy in light and heavy traffic: When is selfish routing bad? In *Book of Abstracts*, page 39.

Corman, F., Viti, F. & Negenborn, R. R. (2017). Equilibrium models in multimodal container transport systems. *Flexible Services and Manufacturing Journal*, 29(1):125–153.

Dafermos, S. (1982). The general multimodal network equilibrium problem with elastic demand. *Networks*, 12(1):57–72.

Dafermos, S. C. (1969). Traffic assignment and resource allocation in transportation networks.

Daganzo, C. F. (1985). The uniqueness of a time-dependent equilibrium distribution of arrivals at a single bottleneck. *Transportation science*, 19(1):29–37.

Delle Site, P. (2018). A mixed-behaviour equilibrium model under predictive and static advanced traveller information systems (atis) and state-dependent route choice. *Transportation Research Part C: Emerging Technologies*, 86:549–562.

Di, X., He, X., Guo, X. & Liu, H. X. (2014). Braess paradox under the boundedly rational user equilibria. *Transportation Research Part B: Methodological*, 67:86–108.

Di, X. & Liu, H. X. (2016). Boundedly rational route choice behavior: A review of models and methodologies. *Transportation Research Part B: Methodological*, 85:142–179.

Dial, R. B. (2006). A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration. *Transportation Research Part B: Methodological*, 40(10):917–936.

Ding, B., Yu, J. X. & Qin, L. (2008). Finding time-dependent shortest paths over large graphs. In *Proceedings of the 11th international conference on Extending database technology: Advances in database technology*, pages 205–216. ACM.

Drissi-Kaïtouni, O. & Hameda-Benchekroun, A. (1992). A dynamic traffic assignment model and a solution algorithm. *Transportation Science*, 26(2):119–128.

D'Ambrogio, A., Iazeolla, G., Pasini, L. & Pieroni, A. (2009). Simulation model building of traffic intersections. *Simulation Modelling Practice and Theory*, 17(4):625–640.

Ehrgott, M., Wang, J. Y. & Watling, D. P. (2015). On multi-objective stochastic user equilibrium. *Transportation Research Part B: Methodological*, 81:704–717.

Fellendorf, M. (1994). Vissim: A microscopic simulation tool to evaluate actuated signal control including bus priority. In *64th Institute of Transportation Engineers Annual Meeting*, vol. 32, pages 1–9. Springer.

Florian, M. & Hearn, D. (1995). Network equilibrium models and algorithms. *Handbooks in Operations Research and Management Science*, 8:485–550.

Florian, M., Mahut, M. & Tremblay, N. (2008). Application of a simulation-based dynamic traffic assignment model. *European journal of operational research*, 189(3):1381–1392.

Florian, M. & Morosan, C. D. (2014). On uniqueness and proportionality in multi-class equilibrium assignment. *Transportation Research Part B: Methodological*, 70:173–185.

Flötteröd, G. (2018). Dta simulation with reduced number of iterations. In *Swedish national transport conference, Göteborg, Sweden, October 16, 2018.*

Fonseca, C. M. & Fleming, P. J. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16.

Foytik, P., Jordan, C. & Robinson, R. M. (2017). Exploring simulation based dynamic traffic assignment with a large-scale microscopic traffic simulation model. In *Proceedings of the 50th Annual Simulation Symposium*, page 11. Society for Computer Simulation International.

Frank, M. (1981). The braess paradox. *Mathematical Programming*, 20(1):283–302.

Frank, M. & Wolfe, P. (1956). An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110.

Friesz, T. L. (2010). *Dynamic optimization and differential games*, vol. 135. Springer Science & Business Media.

Friesz, T. L., Bernstein, D., Suo, Z. & Tobin, R. L. (2001). Dynamic network user equilibrium with state-dependent time lags. *Networks and Spatial Economics*, 1(3-4):319–347.

Friesz, T. L. & Han, K. (2018). The mathematical foundations of dynamic user equilibrium.

Transportation Research Part B: Methodological.

Friesz, T. L., Kim, T., Kwon, C. & Rigdon, M. A. (2011). Approximate network loading and dual-time-scale dynamic user equilibrium. *Transportation Research Part B: Methodological*, 45(1):176–207.

Galligari, A. & Sciandrone, M. (2017). A convergent and fast path equilibration algorithm for the traffic assignment problem. *Optimization Methods and Software*, 33(2):354–371.

Gendron-Carrier, N., Gonzalez-Navarro, M., Polloni, S. & Turner, M. A. (2018). Subways and urban air pollution. Tech. rep., National Bureau of Economic Research.

Gentile, G. (2012). Local user cost equilibrium: a bush-based algorithm for traffic assignment. *Transportmetrica A: Transport Science*, 10(1):15–54.

Gentile, G. (2016). Solving a dynamic user equilibrium model based on splitting rates with gradient projection algorithms. *Transportation Research Part B: Methodological*, 92:120–147.

Guevara, C., Valdés, E. & Dávila, J. (2011). Urban transportation system optimum: The effect of accounting for the interaction of buses and cars. *Ingeniería y Desarrollo*, 29:101–126.

Halat, H., Zockaie, A., Mahmassani, H. S., Xu, X. & Verbas, O. (2016). Dynamic network equilibrium for daily activity-trip chains of heterogeneous travelers: application to large-scale networks. *Transportation*, 43(6):1041–1059.

Halpern, B. (1967). Fixed points of nonexpanding maps. *Bulletin of the American Mathematical Society*, 73(6):957–962.

Han, K., Eve, G. & Friesz, T. L. (2019). Computing dynamic user equilibria on large-scale networks with software implementation. *Networks and Spatial Economics*.

Han, K., Friesz, T. L., Szeto, W. & Liu, H. (2015a). Elastic demand dynamic network user equilibrium: Formulation, existence and computation. *Transportation Research Part B: Methodological*, 81:183–209.

Han, K., Szeto, W. & Friesz, T. L. (2015b). Formulation, existence, and computation of boundedly rational dynamic user equilibrium with fixed or endogenous user tolerance. *Transportation Research Part B: Methodological*, 79:16–49.

Han, L., Ukkusuri, S. & Doan, K. (2011). Complementarity formulations for the cell transmission model based dynamic user equilibrium with departure time choice, elastic demand and user heterogeneity. *Transportation Research Part B: Methodological*, 45(10):1749–1767.

Hawas, Y., Mahmassani, H., Taylor, R., Ziliaskopoulos, A., Chang, G.-L. & Peeta, S. (1998). Development of dynasmart-x software for real-time dynamic traffic assignment. Unknown Publisher. Reportnumber: RFPST067-85 -Task E.

Hilber, C. A. & Palmer, C. (2014). Urban development and air pollution: Evidence from a global panel of cities. *Grantham Research Institute on Climate Change and the Environment Working Paper series*.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

Hoogendoorn, S. P. & Bovy, P. H. (2001). State-of-the-art of vehicular traffic flow modelling. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 215(4):283–303.

Huang, H.-J. & Lam, W. H. (2002). Modeling and solving the dynamic user equilibrium route and departure time choice problem in network with queues. *Transportation Research Part B: Methodological*, 36(3):253–273.

Huapu, L., Qixin, S. & Yafeng, Y. (1996). Dynamic traffic assignment model–a review and future. *Journal of Highway and Transportation Research and Development*, 13(2).

Iryo, T. (2011). Multiple equilibria in a dynamic traffic network. *Transportation Research Part B: Methodological*, 45(6):867–879.

Iryo, T. (2013). Properties of dynamic user equilibrium solution: existence, uniqueness, stability, and robust solution methodology. *Transportmetrica B: Transport Dynamics*, 1(1):52–67.

Iryo, T. (2015). Investigating factors for existence of multiple equilibria in dynamic traffic network. *Networks and Spatial Economics*, 15(3):599–616.

Iryo, T. & Smith, M. J. (2017). On the uniqueness of equilibrated dynamic traffic flow patterns in unidirectional networks. *Transportation Research Procedia*, 23:283–302.

Iryo, T. & Smith, M. J. (2018). On the uniqueness of equilibrated dynamic traffic flow patterns in unidirectional networks. *Transportation Research Part B: Methodological*, 117:757–773.

Janson, B. N. (1991). Dynamic traffic assignment for urban road networks. *Transportation Research Part B: Methodological*, 25(2-3):143–161.

Jayakrishnan, R. & Rindt, C. R. (1999). Distributed computing and simulation in a traffic research test bed. *Computer-Aided Civil and Infrastructure Engineering*, 14(6):429–443.

Jayakrishnan, R., Tsai, W. T., Prashker, J. N. & Rajadhyaksha, S. (1994). A faster path-based algorithm for traffic assignment. *Transportation Research Record*, (1443).

Jeihani, M. (2007). A review of dynamic traffic assignment computer packages. In *Journal of the Transportation Research Forum*, vol. 46, pages 34–46.

Jiang, Y., Szeto, W., Long, J. & Han, K. (2016). Multi-class dynamic traffic assignment with physical queues: intersection-movement-based formulation and paradox. *Transportmetrica A: Transport Science*, 12(10):878–908.

Jordan, C., Foytik, P., Collins, A. & Robinson, R. M. (2017). Development of a future year large-scale microscopic traffic simulation model. In *TRB 2017, Transportation Research Board 96th Annual Meeting*.

Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Klein, I., Levy, N. & Ben-Elia, E. (2018). An agent-based model of the emergence of cooperation and a fair and stable system optimum using atis on a simple road network. *Transportation research part C: emerging technologies*, 86:183–201.

Konishi, H. (2004). Uniqueness of user equilibrium in transportation networks with heterogeneous commuters. *Transportation science*, 38(3):315–330.

Krug, J., Burianne, A. & Leclercq, L. (2019). Reconstituting demand patterns of the city of lyon by using multiple gis data sources. Tech. rep., University of Lyon, ENTPE, LICIT.

Larsson, T. & Patriksson, M. (1992). Simplicial decomposition with disaggregated representation for the traffic assignment problem. *Transportation Science*, 26(1):4–17.

Laval, J. A. & Leclercq, L. (2008). Microscopic modeling of the relaxation phenomenon using a macroscopic lane-changing model. *Transportation Research Part B: Methodological*, 42(6):511–522.

LeBlanc, L. J., Morlok, E. K. & Pierskalla, W. P. (1975). An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation research*, 9(5):309–318.

Leclercq, L. (2007a). Bounded acceleration close to fixed and moving bottlenecks. *Trans-*

portation Research Part B: Methodological, 41(3):309–319.

Leclercq, L. (2007b). Hybrid approaches to the solutions of the "lighthill–whitham–richards" model. *Transportation Research Part B: Methodological*, 41(7):701–709.

Leclercq, L., Chevallier, E. & Laval, J. (2008). The lagrangian coordinates applied to the LWR model. In *Hyperbolic Problems: Theory, Numerics, Applications*, pages 671–678. Springer Berlin Heidelberg.

Leclercq, L. & Laval, J. A. (2009). A multiclass car-following rule based on the lwr model. In *Traffic and Granular Flow'07*, pages 151–160. Springer.

Leclercq, L., Verchier, A., Krug, J. & Menendez, M. (2016). Investigating the performances of the method of successive averages for determining dynamic user equilibrium and system optimum in manhattan networks. In *DTA2016, 6th International Symposium on Dynamic Traffic Assignment*, pages 1–p.

Levin, M. W., Boyles, S. D. & Nezamuddin (2014a). Warm-starting dynamic traffic assignment with static solutions. *Transportmetrica B: Transport Dynamics*, 3(2):99–113.

Levin, M. W., Pool, M., Owens, T., Juri, N. R. & Waller, S. T. (2014b). Improving the convergence of simulation-based dynamic traffic assignment methodologies. *Networks and Spatial Economics*, 15(3):655–676.

Lin, D.-Y., Valsaraj, V. & Waller, S. T. (2011). A dantzig-wolfe decomposition-based heuristic for off-line capacity calibration of dynamic traffic assignment. *Computer-Aided Civil and Infrastructure Engineering*, 26(1):1–15.

Lindsey, R. (2004). Existence, uniqueness, and trip cost function properties of user equilibrium in the bottleneck model with multiple user classes. *Transportation science*, 38(3):293–314.

Liu, H. X., He, X. & He, B. (2009). Method of successive weighted averages (mswa) and self-regulated averaging schemes for solving stochastic user equilibrium problem. *Networks and Spatial Economics*, 9(4):485.

Lu, C.-C., Mahmassani, H. S. & Zhou, X. (2009). Equivalent gap function-based reformulation and solution algorithm for the dynamic user equilibrium problem. *Transportation Research Part B: Methodological*, 43(3):345 – 364.

Mahmassani, H. S. (1998). Dynamic traffic simulation and assignment: Models, algorithms and application to ATIS / ATMS evaluation and operation. In *Operations Research and Decision Aid Methodologies in Traffic and Transportation Management*, pages 104–135. Springer Berlin Heidelberg.

Mahmassani, H. S. (2001). Dynamic network traffic assignment and simulation methodology for advanced system management applications. *Networks and spatial economics*, 1(3):267–292.

Mahmassani, H. S. & Abdelghany, K. F. (2002). Dynasmart-ip: Dynamic traffic assignment meso-simulator for intermodal networks. In *Advanced Modeling for Transit Operations and Service Planning*, pages 200–229. Emerald Group Publishing Limited.

Mahmassani, H. S. & Chang, G.-L. (1987). On boundedly rational user equilibrium in transportation systems. *Transportation science*, 21(2):89–99.

Mahmassani, H. S., Peeta, S., Chang, G.-L. & Junchaya, T. (1991). A review of dynamic assignment and traffic simulation models for adis/atms applications. In *Technical Report DTFH61-90-R-0074-1*. Citeseer.

Mahut, M., Florian, M. & Tremblay, N. (2008). Comparison of assignment methods for simulation-based dynamic-equilibrium traffic assignment. In *Proceeding of the Transporta-*

tion Research Board 87th Annual Meeting (DVD), Washington, DC.

Maini, H., Mehrotra, K., Mohan, C. & Ranka, S. (1994). Knowledge-based nonuniform crossover. *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*.

Marcotte, P. & Nguyen, S., eds. (1998). *Equilibrium and Advanced Transportation Modelling*. Springer US.

Marcotte, P. & Wynter, L. (2004). A new look at the multiclass network equilibrium problem. *Transportation Science*, 38(3):282–292.

Mehrabipour, M., Hajibabai, L. & Hajbabaie, A. (2019). A decomposition scheme for parallelization of system optimal dynamic traffic assignment on urban networks with multiple origins and destinations. *Computer-Aided Civil and Infrastructure Engineering*.

Miaou, S.-P., Summers, M. S. & Lieu, H. C. (1999). Laboratory evaluation of real-time dynamic traffic assignment systems. *Computer-Aided Civil and Infrastructure Engineering*, 14(4):281–298.

Mitradjieva, M. & Lindberg, P. O. (2013). The stiff is moving—conjugate direction frank-wolfe methods with applications to traffic assignment. *Transportation Science*, 47(2):280–293.

Mounce, R. (2007). Existence of equilibrium in a continuous dynamic queueing model for traffic networks. In *4th IMA International Conference on Mathematics in TransportInstitute of Mathematics and its Applications*.

Mounce, R. & Carey, M. (2011). Route swapping in dynamic traffic networks. *Transportation Research Part B: Methodological*, 45(1):102–111.

Mounce, R. & Carey, M. (2015). On the convergence of the method of successive averages for calculating equilibrium in traffic networks. *Transportation Science*, 49(3):535–542.

Mounce, R. & Smith, M. (2007). Uniqueness of equilibrium in steady state and dynamic traffic networks. In *Transportation and Traffic Theory 2007 (ISTTT17).*

Mun, J.-S. (2007). Traffic performance models for dynamic traffic assignment: an assessment of existing models. *Transport reviews*, 27(2):231–249.

Myerson, R. B. (1999). Nash equilibrium and the history of economic theory. *Journal of Economic Literature*, 37(3):1067–1082.

Nagel, K. & Flötteröd, G. (2016). Agent-based traffic assignment. In *The Multi-Agent Transport Simulation MATSim*, pages 315–326. Ubiquity Press.

Nagurney, A. & Zhang, D. (2012). *Projected dynamical systems and variational inequalities with applications*, vol. 2. Springer Science & Business Media.

Nagurney, A. B. (1984). Comparative tests of multimodal traffic equilibrium methods. *Transportation Research Part B: Methodological*, 18(6):469–485.

Narayanaswami, S. (2016). Urban transportation: trends, challenges and opportunities.

Netter, M. (1972). Affectation de trafic et tarification au cout marginal social: Critique de quelques idees admises. *Transp Res*, 6(4).

Ng, M. & Waller, S. T. (2012). A dynamic route choice model considering uncertain capacities. *Computer-Aided Civil and Infrastructure Engineering*, 27(4):231–243.

Nguyen, S. & Dupuis, C. (1984). An efficient method for computing traffic equilibria in networks with asymmetric transportation costs. *Transportation Science*, 18(2):185–202.

Nie, Y. M. (2010). A class of bush-based algorithms for the traffic assignment problem. *Transportation Research Part B: Methodological*, 44(1):73–89.

Nilsson, G., Grover, P. & Kalabić, U. (2018). Assignment and control of two-tiered vehicle

traffic. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 1023–1028. IEEE.

Osawa, M., Fu, H. & Akamatsu, T. (2018). First-best dynamic assignment of commuters with endogenous heterogeneities in a corridor network. *Transportation Research Part B: Methodological*, 117:811–831.

Osorio, C. & Bierlaire, M. (2013). A simulation-based optimization framework for urban transportation problems. *Operations Research*, 61(6):1333–1345.

Pas, E. I. & Principio, S. L. (1997). Braess' paradox: Some new insights. *Transportation Research Part B: Methodological*, 31(3):265–276.

Patriksson, M. (2015). *The traffic assignment problem: models and methods*. Courier Dover Publications.

Peeta, S. & Mahmassani, H. S. (1995). System optimal and user equilibrium time-dependent traffic assignment in congested networks. *Annals of Operations Research*, 60(1):81–113.

Peeta, S. & Ziliaskopoulos, A. K. (2001). Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and spatial economics*, 1(3-4):233–265.

Peeta, S. & Ziliaskopoulos, A. K. (2001). Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and Spatial Economics*, 1:233–265.

Pel, A. J., Bliemer, M. C. & Hoogendoorn, S. P. (2012). A review on travel behaviour modelling in dynamic traffic simulation models for evacuations. *Transportation*, 39(1):97–123.

Peque, G., Miyagi, T. & Kurauchi, F. (2018). Adaptive learning algorithms for simulation-based dynamic traffic user equilibrium. *International Journal of Intelligent Transportation Systems Research*, 16(3):215–226.

Perederieieva, O., Ehrgott, M., Raith, A. & Wang, J. Y. (2015). Numerical stability of path-based algorithms for traffic assignment. *Optimization Methods and Software*, 31(1):53–67.

Qian, Z. & Zhang, H. M. (2011). Computing individual path marginal cost in networks with queue spillbacks. *Transportation Research Record*, 2263(1):9–18.

Raadsen, M. P., Bliemer, M. C. & Bell, M. G. (2019). A review of (dis)aggregation and decomposition methods in traffic assignment.

Rakha, H. & Tawfik, A. (2009). Traffic networks: dynamic traffic routing, assignment, and assessment. *Encyclopedia of Complexity and Systems Science*, pages 9429–9470.

Ramadurai, G. & Ukkusuri, S. (2011). B-dynamic: An efficient algorithm for dynamic user equilibrium assignment in activity-travel networks1. *Computer-Aided Civil and Infrastructure Engineering*, 26(4):254–269.

Robbins, H. & Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Roughgarden, T. (2005). *Selfish routing and the price of anarchy*, vol. 174. MIT press Cambridge.

Samaranayake, S., Krichene, W., Reilly, J., Monache, M. L. D., Goatin, P. & Bayen, A. (2018). Discrete-time system optimal dynamic traffic assignment (so-dta) with partial control for physical queuing networks. *Transportation Science*, 52(4):982–1001.

Sancho, E. C., Ibáñez Marí, G. & Bugeda, J. B. (2015). Applying projection-based methods to the asymmetric traffic assignment problem. *Computer-Aided Civil and Infrastructure Engineering*, 30(2):103–119.

Sbayti, H., Lu, C.-C. & Mahmassani, H. S. (2007). Efficient implementation of method of successive averages in simulation-based dynamic traffic assignment models for large-scale network applications. *Transportation Research Record: Journal of the Transportation Research Board*, 2029:22–30.

Schreiter, T., Wageningen-Kessels, V., Yuan, Y., Van Lint, J., Hoogendoorn, S. *et al.* (2012). Fastlane: Traffic flow modeling and multi-class dynamic traffic management. In *Trail-Beta Congress 2012: Mobility and Logistics-Science Meets Practice, Rotterdam, The Netherlands, 30-31 October 2012*. Citeseer.

Seshadri, R. & Srinivasan, K. K. (2017). Robust traffic assignment model: Formulation, solution algorithms and empirical application. *Journal of Intelligent Transportation Systems*, 21(6):507–524.

Sheffi, Y. (1985). *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*, pages 322–331. Prentice-Hall Inc, Englewood Cliffs, New Jersey, USA.

Silva, H. E., Lindsey, R., De Palma, A. & Van den Berg, V. A. (2016). On the existence and uniqueness of equilibrium in the bottleneck model with atomic users. *Transportation Science*, 51(3):863–881.

Smith, L., Beckman, R. & Baggerly, K. (1995). Transims: Transportation analysis and simulation system. Tech. rep., Los Alamos National Lab., NM (United States).

Smith, M. (1993). A new dynamic traffic model and the existence and calculation of dynamic user equilibria on congested capacity-constrained road networks. *Transportation Research Part B: Methodological*, 27(1):49–63.

Smith, M. J. (1979). The existence, uniqueness and stability of traffic equilibria. *Transportation Research Part B: Methodological*, 13(4):295–304.

Song, W., Han, K., Wang, Y., Friesz, T. & Del Castillo, E. (2017). Statistical metamodeling of dynamic network loading. *Transportation research procedia*, 23:263–282.

Souche, S., Mercier, A. & Ovtracht, N. (2016). The impacts of urban pricing on social and spatial inequalities: The case study of lyon (france). *Urban Studies*, 53(2):373–399.

Srinivas, M. & Patnaik, L. (1994). Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(4):656–667.

Srinivasan, S., Riazi, S., Norris, B., Das, S. K. & Bhowmick, S. (2018). A shared-memory parallel algorithm for updating single-source shortest paths in large dynamic networks. In *2018 IEEE 25th International Conference on High Performance Computing (HiPC)*. IEEE.

Sun, C., Cheng, L. & Xu, T. (2014). Range of user-equilibrium route flow with applications. *Procedia-Social and Behavioral Sciences*, 138:86–96.

Sun, Y., Yu, X., Bie, R. & Song, H. (2017). Discovering time-dependent shortest path on traffic graph for drivers towards green driving. *Journal of Network and Computer Applications*, 83:204–212.

Sundaram, S., Koutsopoulos, H. N., Ben-Akiva, M., Antoniou, C. & Balakrishna, R. (2011). Simulation-based dynamic traffic assignment for short-term planning applications. *Simulation Modelling Practice and Theory*, 19(1):450–462.

Szeto, W. & Wong, S. (2012). Dynamic traffic assignment: model classifications and recent advances in travel choice principles. *Open Engineering*, 2(1):1–18.

Szeto, W. Y. & Lo, H. K. (2004). A cell-based simultaneous route and departure time choice model with elastic demand. *Transportation Research Part B: Methodological*, 38(7):593–612.

Szeto, W. Y. & Lo, H. K. (2005a). Dynamic traffic 'assignment; review and future research directions. *Communication and Transportati0n Systems Engineering and Information*, (5):19.

Szeto, W. Y. & Lo, H. K. (2005b). Non-equilibrium dynamic traffic assignment. In *Transportation and Traffic Theory*, pages 427–445. Elsevier.

Szeto, W. Y. & Lo, H. K. (2005c). Properties of dynamic traffic assignment with physical queues. *Journal of the Eastern Asia Society for Transportation Studies*, 6:2108–2123.

Szeto, W. Y. & Lo, H. K. (2006a). Dynamic traffic assignment: properties and extensions. *Transportmetrica*, 2(1):31–52.

Szeto, W. Y. & Lo, H. K. (2006b). Dynamic traffic assignment: properties and extensions. *Transportmetrica*, 2(1):31–52.

Taale, H. & Pel, A. (2015). Better convergence for dynamic traffic assignment methods. *Transportation Research Procedia*, 10:197–206.

Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*, vol. 74. John Wiley & Sons.

Taylor, N. B. (2003). The contram dynamic traffic assignment model. *Networks and Spatial Economics*, 3(3):297–322.

Tong, C. & Wong, S. (2010). Heuristic algorithms for simulation-based dynamic traffic assignment. *Transportmetrica*, 6(2):97–120.

Van Essen, M., Thomas, T., van Berkum, E. & Chorus, C. (2016). From user equilibrium to system optimum: a literature review on the role of travel information, bounded rationality and non-selfish behaviour at the network and individual levels. *Transport reviews*, 36(4):527–548.

Verbas, İ. Ö., Mahmassani, H. S. & Hyland, M. F. (2015). Dynamic assignment-simulation methodology for multimodal urban transit networks. *Transportation Research Record: Journal of the Transportation Research Board*, 2498(1):64–74.

Verbas, İ. Ö., Mahmassani, H. S. & Hyland, M. F. (2016a). Gap-based transit assignment algorithm with vehicle capacity constraints: Simulation-based implementation and large-scale application. *Transportation Research Part B: Methodological*, 93:1–16.

Verbas, İ. Ö., Mahmassani, H. S., Hyland, M. F. & Halat, H. (2016b). Integrated mode choice and dynamic traveler assignment in multimodal transit networks: Mathematical formulation, solution procedure, and large-scale application. *Transportation Research Record*, 2564(1):78–88.

Wang, Y., Szeto, W., Han, K. & Friesz, T. L. (2018). Dynamic traffic assignment: A review of the methodological advances for environmentally sustainable road transportation applications. *Transportation Research Part B: Methodological*, 111:370–394.

Wardrop, J. G. (1952). Road paper. some theoretical aspects of road traffic research. *Proceedings of the institution of civil engineers*, 1(3):325–362.

Wardrop, J. G. (1952). Some theoretical aspects of road traffic research. *Institution of Civil Engineering*, 1:325–362.

Wie, B.-W., Tobin, R. L. & Carey, M. (2002). The existence, uniqueness and computation of an arc-based dynamic network user equilibrium formulation. *Transportation Research Part B: Methodological*, 36(10):897–918.

Wynter, L. (2001). A convergent algorithm for the multimodal traffic equilibrium problem. Tech. Rep. 4125, Institut National De Recherche en Informatique et en Automatique, Le Chesnay Cedex, France. Available from Internet: https://hal.inria.fr/file/index/docid/72503/filename/RR-4125.pdf.

Xie, J., Nie, Y. M. & Liu, X. (2018). A greedy path-based algorithm for traffic assignment. *Transportation Research Record: Journal of the Transportation Research Board*, 2672(48):36–44.

Xu, H.-K. (2002). Iterative algorithms for nonlinear operators. *Journal of the London Mathematical Society*, 66(1):240–256.

Xu, S., Jiang, W., Deng, X. & Shou, Y. (2018). A modified physarum-inspired model for the user equilibrium traffic assignment problem. *Applied Mathematical Modelling*, 55:340–353.

Yang, Q., Balakrishna, R., Morgan, D. & Slavin, H. (2017). Large-scale, high-fidelity dy-

namic traffic assignment: framework and real-world case studies. *Transportation Research Procedia*, 25:1290–1299.

Yildirimoglu, M. & Kahraman, O. (2018). Investigating the empirical existence of equilibrium conditions. Tech. rep.

Yildirimoglu, M. & Ramezani, M. (2019). Demand management with limited cooperation among travellers: a doubly dynamic approach. *Transportation Research Part B: Methodological*.

Youn, H., Gastner, M. T. & Jeong, H. (2008). Price of anarchy in transportation networks: efficiency and optimality control. *Physical review letters*, 101(12):128701.

Ypma, T. J. (1995). Historical development of the newton–raphson method. *SIAM Review*, 37(4):531–551.

Yu, N., Ma, J. & Zhang, H. M. (2008). A polymorphic dynamic network loading model. *Computer-Aided Civil and Infrastructure Engineering*, 23(2):86–103.

Yun, I. & Park, B. (2006). Application of stochastic optimization method for an urban corridor. In *Proceedings of the 2006 Winter Simulation Conference*. IEEE.

Zhao, X., Wan, C. & Bi, J. (2018). Day-to-day assignment models and traffic dynamics under information provision. *Networks and Spatial Economics*, pages 1–30.

Zhou, B., Xu, M., Meng, Q. & Huang, Z. (2017). A day-to-day route flow evolution process towards the mixed equilibria. *Transportation Research Part C: Emerging Technologies*, 82:210–228.

Zhou, X., Mahmassani, H. S. & Zhang, K. (2008). Dynamic micro-assignment modeling approach for integrated multimodal urban corridor management. *Transportation Research Part C: Emerging Technologies*, 16(2):167–186.

Ziliaskopoulos, A. K. & Mahmassani, H. S. (1993). Time-dependent, shortest-path algorithm for real-time intelligent vehicle highway system applications. *Transportation Research Record*.

Ziliaskopoulos, A. K. & Mahmassani, H. S. (1996). A note on least time path computation considering delays and prohibitions for intersection movements. *Transportation Research Part B: Methodological*, 30(5):359–367.

A. APPENDIX FOR CHAPTER 4

TABLE A.1 – The scenarios of network design for the dynamic test case.

	Simulation code	Num. of active metro	Sequence	Initialization with results of simulation code	Scenario	Description
Phase 1 (day 0)	P1.1	0	-	all-or-nothing	-	There is no metro line in the network
	P1.2	3	A&B&C	P1.1	1st Scenario results	all metro lines are available for the users
	P1.3	1	A	P1.1	-	Just metro A is available
	P1.4	1	B	P1.1	-	Just metro B is available
	P1.5	1	C	P1.1	-	Just metro C is available
Phase 2 (day 100)	P2.1	2	AB	P1.3	-	Activation: metro B - initial assignment pattern is obtained by phase 1 simulation code P1.3
	P2.2	2	AC	P1.3	-	Activation: metro C - initial assignment pattern is obtained by phase 1 simulation code P1.3
	P2.3	2	BA	P1.4	-	Activation: metro A - initial assignment pattern is obtained by phase 1 simulation code P1.4
	P2.4	2	BC	P1.4	-	Activation: metro C - initial assignment pattern is obtained by phase 1 simulation code P1.4
	P2.5	2	CA	P1.5	-	Activation: metro A - initial assignment pattern is obtained by phase 1 simulation code P1.5
	P2.6	2	CB	P1.5	-	Activation: metro B - initial assignment pattern is obtained by phase 1 simulation code P1.5
Phase 3 (day 200)	P3.1	3	ABC	P2.1	2nd Scenario results	Activation: metro C - initial assignment pattern is obtained by phase 2 simulation code P2.1
	P3.2	3	ACB	P2.2	3rd Scenario results	Activation: metro B - initial assignment pattern is obtained by phase 2 simulation code P2.2
	P3.3	3	BAC	P2.3	4th Scenario results	Activation: metro C - initial assignment pattern is obtained by phase 2 simulation code P2.3
	P3.4	3	BCA	P2.4	5th Scenario results	Activation: metro A - initial assignment pattern is obtained by phase 2 simulation code P2.4
	P3.5	3	CAB	P2.5	6th Scenario results	Activation: metro B - initial assignment pattern is obtained by phase 2 simulation code P2.5
	P3.6	3	CBA	P2.6	7th Scenario results	Activation: metro A - initial assignment pattern is obtained by phase 2 simulation code P2.6

B. Appendix for chapter 5

Lyon 6 demand description

The total travel demand for Lyon metropolis (Figure B.1) is about one million trips per day (Souche *et al.*, 2016). During peak hour, we should then observe about 100,000 trips. The size of the Lyon 6 network (1.72 km^2) is about 7% of the full network of Lyon. Therefore, approximately, we should have about 7,000 trips per hour. In this study, based on the real data (Krug *et al.*, 2019), we have, in total, 4361 trips for half an hour in Lyon 6 network, which shows that our test case is fully consistence with real life pattern.

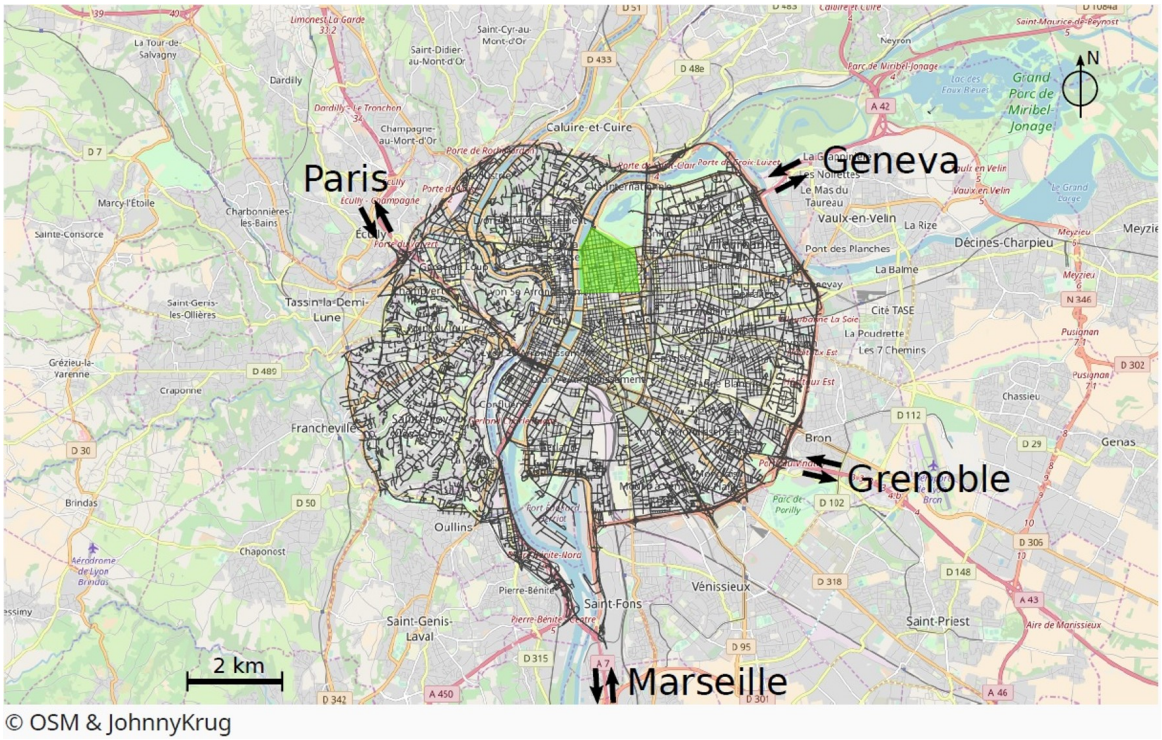


FIGURE B.1 – Full network of Lyon: Lyon 6 is highlighted by green color