University of Cyprus

DEPARTMENT OF CIVIL AND ENVIRONMENTAL ENGINEERING

Synthesis of disaggregate mobility information from aggregate Origin-Destination matrices: A graph-theoretical and combinatorial optimisation approach

Theocharis Ballis

A dissertation submitted to the University of Cyprus in partial fulfilment of the requirements for the degree of Doctor of Philosophy

December, 2020

VALIDATION PAGE

Doctoral Candidate: Theocharis Ballis

Doctoral Thesis Title: Synthesis of disaggregate mobility information from aggregate Origin-Destination matrices: A graph-theoretical, and combinatorial optimisation approach

The present Doctoral Dissertation was submitted in partial fulfilment of the requirements for the Degree of Doctor of Philosophy at the Department of Civil and Environmental Engineering and was approved on the 22nd of July 2020. by the members of the Examination Committee:

Research Supervisor: Dr. Loukas Dimitriou, Assistant Professor, UCY_____

Committee Member: Dr. Symeon Christodoulou, Professor, UCY_____

Committee Member: Dr. Dimos Charmpis, Associate Professor, UCY_____

Committee Member: Dr. Georgios Ellinas, Professor, UCY_____

Committee Member: Dr. Constantinos Antoniou, Professor, TUM_____

DECLARATION OF DOCTORAL CANDIDATE

The present doctoral dissertation was submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy of the University of Cyprus. It is a product of original work of my own, unless otherwise mentioned through references, notes, or any other statements.

_____ (Full name of Doctoral Candidate)

_____ (Signature)

ΠΕΡΙΛΗΨΗ

Τα μητρώα Προέλευσης-Προορισμού (ΠΠ) αποτελούν ένα αναντικατάστατο μέσο αναπαράστασης της ζήτησης για μετακινήσεις, ικανό να αποτυπώσει με συνεκτικό και αποτελεσματικό τρόπο τον συνολικό όγκο μετακινήσεων τόσο στον χώρο όσο και στον χρόνο. Πλήθος οργανισμών, διαχειριστών, μελετητών, και ερευνητών έχουν παραδοσιακά αφιερώσει σημαντικούς πόρους για την ανάπτυξη και την συντήρηση μητρώων ΠΠ για την τεκμηρίωση αποφάσεων σχετικών με τον αστικό σχεδιασμό, την διαχείριση της ζήτησης για μετακινήσεις, την αξιολόγηση πολιτικών, και τελικά τον προγραμματισμό επενδύσεων στην συγκοινωνιακή υποδομή. Πιο πρόσφατα, τα μητρώα ΠΠ έχουν χρησιμοποιηθεί ως ένα ασφαλές από πλευράς ιδιωτικότητας μέσο αναπαράστασης της μετακινησιακής συμπεριφοράς (travel behaviour) χρηστών που φέρουν συσκευές εντοπισμού θέσης (π.χ. κινητά τηλέφωνα, GPS, κτλ.). Εντούτοις, ο αθροιστικός χαρακτήρας των μητρώων ΠΠ τους στερεί την ικανότητα να αποτυπώσουν σημαντικές διαστάσεις της μετακίνησης όπως η αλληλουχία και η αλληλεξάρτηση μεταξύ ταξιδιών. Απότοκο αυτής της αδυναμίας είναι η περιορισμένη αξία της απευθείας χρήσης τους για μελέτες συμπεριφοράς μετακίνησης, ειδικότερα όταν αυτές αφορούν στο ατομικό επίπεδο.

Η παρούσα Διδακτορική Διατριβή προτείνει ένα νέο μεθοδολογικό πλαίσιο για την εξαγωγή εξατομικευμένης μετακινησιακής πληροφορίας (disaggregate mobility information) από μητρώα ΠΠ. Ειδικότερα, μητρώα ΠΠ διατεταγμένα κατά χρονική περίοδο (time-period) και κατηγοριοποιημένα ανά σκοπό μετακίνησης (trip-purpose) μετατρέπονται σε αλυσίδες μετακινήσεων (trip-chains) ή προγράμματα δραστηριοτήτων (activity schedules) που ανασυνθέτουν τα μοτίβα ζήτησης όπως περιγράφονται στα αρχικά μητρώα ΠΠ. Το προτεινόμενο μεθοδολογικό πλαίσιο αναπτύσσεται σε τέσσερα σκέλη (modules) συνδυάζοντας στοιχεία της θεωρίας γράφων καθώς και της μαθηματικής βελτιστοποίησης συνδυαστικού μαθηματικού προγραμματισμού σε μεγάλη κλίμακα.

Το πρώτο σκέλος μετατρέπει την χωρο-χρονική πληροφορία των εισαχθέντων μητρών ΠΠ σε έναν υβριδικό δυναμικό γράφο (hTVG) ώστε να επιτευχθεί η αποδοτική εφαρμογή αλγορίθμων θεωρίας γράφων σε δυναμικά συστήματα. Ειδικότερα, η μετατροπή των μητρώων ΠΠ σε hTVG επιτρέπει την αναπαράσταση των Αλυσίδων με Βάση την Οικία (ΑΒΟ) ως διαδρομές (paths) μέσα στον γράφο. Το δεύτερο σκέλος αναλαμβάνει την απαρίθμηση όλων των πιθανών ΑΒΟ (αναφέρονται ως *υποψήφιες ΑΒΟ*) που μπορούν να δημιουργηθούν στον υπό μελέτη γράφο. Το τρίτο σκέλος, αξιοποιεί την πληροφορία του σκοπού μετακίνησης ώστε να μετατρέψει τις ΑΒΟ σε *Προγράμματα Δραστηριοτήτων* (ΠΔ) και τελικώς να παράξει το σύνολο των *υποψήφιων ΠΔ*. Το τέταρτο και τελευταίο σκέλος αναζητεί τον βέλτιστο συνδυασμό μεταξύ των υποψήφιων ΠΔ ο οποίος αναπαράγει την

ζήτηση για μετακινήσεις όπως αυτή αποτυπώνεται στα εισαχθέντα μητρώα ΠΠ. Ο στόχος της βελτιστοποίησης είναι η μεγιστοποίηση της χρήσης των μετακινήσεων που εμπεριέχονται στα μητρώα ΠΠ τηρώντας παράλληλα περιορισμούς που προέρχονται από διαθέσιμα δεδομένα βαθμονόμησης (calibration data) όπως το προφίλ αναχωρήσεων των ΑΒΟ, η ημερήσια κατανομή των δραστηριοτήτων στα ΠΔ, ή άλλη διαθέσιμη πληροφορία.

Η εφαρμογή της παραπάνω μεθοδολογίας σε ρεαλιστικές περιπτώσεις απαιτεί την ανάπτυξη κατάλληλων τεχνικών που να επιτρέπουν την επεκτασιμότητα (scalability) της. Στα πλαίσια αυτής της Διατριβής, η επεκτασιμότητα επετεύχθη μέσω της απλοποίησης του hTVG γράφου καθώς και μέσω της απαλοιφής υποψήφιων ΑΒΟ και ΠΔ με χαμηλή πιθανότητα παρατήρησης (π.χ. ΑΒΟ με δυσανάλογο χρόνο μετακίνησης).

Η εγκυρότητα της μεθοδολογίας δοκιμάστηκε σε ένα πλήρως ελεγχόμενο δειγματικό χώρο. Συγκεκριμένα, 25,000 *παρατηρημένα ΠΔ* συναθροίστηκαν για την δημιουργία 28 *παρατηρημένων πινάκων ΠΠ* που περιέχουν 53,104 μετακινήσεις, κατηγοριοποιημένες με βάση τον σκοπό και την χρονική περίοδο έναρξης της μετακίνησης. Η εφαρμογή της μεθοδολογίας μετέτρεψε τους παρατηρημένους πίνακες ΠΠ σε ένα σύνολο 24,818 *μοντελοποιημένων ΠΔ* τα οποία αναπαράστησαν τα αντίστοιχα παρατηρημένα ΠΔ με ακρίβεια άνω του 90% και τα οποία απαιτούν για την ολοκλήρωσή τους το 99% της ζήτησης για μετακίνηση όπως αυτή αποτυπώνεται στους παρατηρημένους πίνακες ΠΠ. Η επεκτασιμότητα της μεθοδολογίας επικυρώθηκε μέσω ενός παρόμοιου με το προηγούμενο αλλά σημαντικώς μεγαλύτερης έκτασης πειράματος. Συγκεκριμένα, 28 *παρατηρημένοι πίνακες ΠΠ* αποτελούμενοι από 268,315 μετακινήσεις/ταξίδια που απαιτούνται για την ολοκλήρωση 125,000 *παρατηρημένων ΑΒΟ* καθώς και δεδομένα βαθμονόμησης (κατανομή συνολικού χρόνου μετακίνησης και προφίλ αναχωρήσεων για τις παρατηρημένες ΑΒΟ) αποτέλεσαν τα δεδομένα εισόδου για την εφαρμογή της μεθοδολογίας σε ευρεία κλίμακα. Οι παραχθείσες *μοντελοποιημένες ΑΔΟ* αναγνώρισαν το 90% των παρατηρούμενων πινάκων ΠΠ χωρίς να διαφέρουν σημαντικά (±2.0% σφάλμα) από τα δεδομένα βαθμονόμησης.

Σημειώνεται ότι ειδικά για την επίλυση του δειγματικού χώρου μεγάλη κλίμακας, αναπτύχθηκε νέος αλγόριθμος στοχαστικής βελτιστοποίησης, ο οποίος επεκτείνει τον ευρέως διαδεδομένο αλγόριθμο Προσομοιωμένης Ανόπτησης (Simulated Annealing) εισάγοντας επιπλέον μηχανισμό που εξασφαλίζει την τήρηση στοχαστικών περιορισμών πολλαπλών διαστάσεων. Ο προτεινόμενος αλγόριθμος, απεδείχθη ικανός να αντιμετωπίσει προβλήματα βελτιστοποίησης εξαιρετικά μεγάλων διαστάσεων καθώς και προβλήματα στοχαστικής σύνθεσης που δεν είναι δυνατόν να αντιμετωπισθούν από αναλυτικές ρουτίνες εμπορικών επιλυτών (commercial solvers).

Εν κατακλείδι, το προτεινόμενο μεθοδολογικό πλαίσιο συνεισφέρει ένα αποτελεσματικό αναλυτικό πλαίσιο για την μετατροπή αθροιστικών πινάκων ΠΠ σε εξατομικευμένη πληροφορία κινητικότητας υπό την μορφή αλληλουχιών ταξιδιών ή προγραμμάτων δραστηριοτήτων. Η συνεισφορά της Διατριβής είναι σημαντική τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο καθώς επιτρέπει την χρήση των ευρέως διαδεδομένων πινάκων ΠΠ για την μελέτη της μεταφορικής συμπεριφοράς σε ατομικό επίπεδο κάτι το οποίο μέχρι πρότινος δεν είχε παρουσιαστεί ούτε στη σχετική βιβλιογραφία ούτε στην πρακτική. Επιπροσθέτως, η προτεινόμενη μεθοδολογία διευκολύνει την δημιουργία δεδομένων εισόδου κατάλληλα για χρησιμοποίηση σε εξατομικευμένα και οδηγούμενα από δεδομένα μοντέλα (disaggregate and data-driven) ικανά να αντιμετωπίσουν μελλοντικά προβλήματα στο πεδίο της ανάλυσης της ζήτησης για κινητικότητα.

Λέξεις κλειδιά: Πίνακες Προέλευσης-Προορισμού, εξατομίκευση πληροφορίας, θεωρία γράφων, συνδυαστική βελτιστοποίηση μεγάλης κλίμακας, κινητικότητα

ABSTRACT

Origin-Destination (OD) matrices constitute an irreplaceable component of transport planning and modelling to represent effectively and concisely the volume of movements both in space and time. Countless Transport Authorities, operators, practitioners, and researchers have over the years allocated significant resources for the development and maintenance of ODs to support a plethora of decisions regarding urban planning, transport policing, and infrastructure investments. More recently, ODs have been also utilised by urban sensing data providers as an effective means for the representation of peoples' mobility traces while ensuring the intractability of the tracked users. Nonetheless, the aggregate nature of ODs deprives them from the ability to express significant dimensions of travel behaviour such as trip-chaining and trip-interdependency. Consequently, ODs do not prove particularly suitable for the analysis of mobility and travel behaviour, especially at the person-level.

The currently presented Ph.D. Thesis proposes a novel methodological framework for the preparation of disaggregate mobility information from aggregate ODs. In particular, the methodology allows the conversion of multi-period, trip-purpose segmented ODs into sets of travel demand equivalent, home-based trip-chains (i.e. tours) and the corresponding activity schedules. The framework combines advanced graph-theory with large scale integer/combinatorial optimisation concepts and is executed in a modular fashion including four steps.

At first, the spatiotemporal information present in the input ODs is used to create a hybrid Time Varying Graph (hTVG) supporting the application of graph-theory-based methodologies for the study of dynamic systems. Converting ODs to a graph enables the expression of tours as paths originating and ending at the same vertex (home location). The second step entails the application of a suitably modified algorithm for the enumeration of all the plausible tours within the hTVG. The third step exploits the trip-purpose information in the ODs to convert the identified tours into activity schedules whose complete set is referred to as the *candidate activity schedules*. In the final step, an advanced combinatorial optimization procedure attempts to identify compositions of the candidate activity schedules which replicate the travel demand patterns expressed in the input ODs. The objective of the optimization process is the minimisation of unused trips from the input ODs while respecting constraints imposed by any available calibration information (e.g. departure time profile of tours, diurnal distribution of activities in activity schedules, etc.).

Despite the sound theoretical foundation, the combinatorial nature of the formulation and the potentially excessive number of plausible tours in large-scale graphs can jeopardise the applicability of the framework on realistic, large-scale cases. However, scalability was ensured through the development of suitable methodologies aiming at the simplification of the graph's structure and as well as at the elimination of unrealistic candidate tours/activity schedules (e.g. activity schedules with disproportional total travel time).

The ability of the proposed methodology to reveal hidden disaggregate travel demand patterns within aggregate ODs was established based on a fully controlled experiment. In particular, a set of 25,000 *observed activity schedules* was used to form 28 multi-period and purpose-dependent *observed ODs* which included 53,104 trips in total. Trips within the observed ODs were utilised by 99% to form *modelled activity schedules* which replicated their observed counterparts with 90% accuracy. Furthermore, the scalability of the framework was verified by a similar to the previous but considerably larger experiment. In detail, 28 *observed ODs* (268,315 trips) deriving from the aggregation of 125,000 *observed tours* and relevant calibration data (i.e. distribution of total travel time and departure profile of the observed tours) were provided as input to the methodology. The resulting *modelled tours* incorporated 90% of the observed travel demand without deviating considerably (±2.0 error) from the calibration information.

It should be also stated that the solution of the large-scale experiment required the development of a new optimisation algorithm which extends the widely used Simulated Annealing algorithm with a mechanism ensuring the adherence to multi-dimensional stochastic constraints. The suggested Adaptive Sampling Simulated Annealing (ASSA) achieved the efficient addressing of excessively large combinatorial optimisation problem, not easily solvable by state-of-the-art commercial optimisation solvers.

In conclusion, the suggested framework provides an effective approach for the conversion of aggregate OD matrices into disaggregate mobility traces (i.e. trip-chains, tours, and activity schedules). The contribution in the field of transport modelling and travel behaviour analysis can prove substantial because the widely available ODs can be now utilised for the studying of mobility in a disaggregate and considerably more informative manner. Finally, the proposed methodological framework can support the transition to the new disaggregate and data-driven modelling era by allowing the exploitation of ODs to produce input suitable for the emerging disaggregate modelling paradigms.

Keywords: Origin-Destination matrices, data disaggregation, graph-theory, large-scale combinatorial optimisation, mobility

ACKNOWLEDGMENTS

The conception as well as the conduction of this Ph. D. Thesis would have never been achieved without the continuous guidance and support from my supervisor Prof. Loukas Dimitriou. His ideas and recommendations proved invaluable for the shaping and the completion of the Thesis.

I would also like to sincerely thank the colleagues I had the chance to work with during the duration of this study. Their views and suggestions helped me greatly to re-evaluate many of my decisions. In addition, it would be unfair to forket expressing my gratitude to all those friends who supported this pursuit.

Finally, a special mention is devoted to my Family for their unwavering and constant support. This Ph. D. Thesis is dedicated to them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACO	Ant Colony Optimisation
ASA	Adaptive Simulated Annealing
ASSA	Adaptive Sampling Simulated Annealing
CDR	Call Detail Record
CPU	Central Processing Unit
DNA	Deoxyribonucleic acid
EPR	Exploration and Preferential Return
EV	Eigenvector centrality
FH	From Home
GB	Great Britain
GIS	Geographical Information Systems
GPS	Global Positioning System
HB	Home-Based
HBO	Home-Based-Other
HBW	Home-Based-Work
HMM	Hidden Markov Model
hTVG	hybrid Time Varying Graph
ID	Identity
IO-HMM	Input-Output Hidden Markov Model
IP	Inter Peak
IPF	Iterative Proportional Fitting
IPU	Iterative Proportional Update
LSOA	Lower Super Output Areas
LSTM	Long Short-Term Memory
ML	Machine Learning
MND	Mobile Network Data
MSOA	Middle Super Output Areas
NHB	Non-Home-Based
NHBO	Non-Home-Based Other
NHBW	Non-Home-Based Work
NMS	New Mobility Services
NTS	National Travel Survey
OD	Origin-Destination (matrix)

OP	Off Peak
PISAA	Parallel and Interacting Stochastic Approximation Annealing
PR	PageRank centrality
RAM	Random Access Memory
RWB	Random Walk Betweenness centrality
SA	Simulated Annealing
SAA	Stochastic Approximation Monte Carlo
SAM	Sequential Alignment Method
SC	Subgraph centrality
TH	To Home
TSP	Travelling Salesman Problem
TVG	Time Varying Graph
UK	United Kingdom
USA	United States of America

Chapter 1

Introduction

The Chapter introduces the reader into the study by presenting the motivation behind its conduction. Additionally, the Chapter defines the objectives of the Thesis and clearly outlines the followed approach to accomplish them.

1.1 Motivation

The instrumental role of transportation planning for the ensuring of progress and economic growth has been well understood in modern societies. Until the past four decades, the primary focus of transportation planning was the provision of the required infrastructure to meet long-term transport demand. However, due to increased capital costs and environmental concerns, the focus of transport planning is gradually shifting towards the optimisation of transport and travel demand management with the aim to influence the travel behaviour of individuals for the effective control of aggregate travel demand. Nonetheless, efficient transport management policies require the in-depth understanding of travel behaviour at person-level because the reaction of different persons to the same travel demand management policy can vary considerably. As a result, the core of travel demand research is increasingly moving from aggregate, long-term predictions to the estimation of immediate travel behaviour reactions at the disaggregate-level. This trend has been also bolstered by recent technological advances (e.g. wireless connectivity, smartphones, autonomous vehicles, etc.) which have enabled the effective coordination between travel demand and transportation supply, unlocking a plethora of new mobility solutions. These emerging solutions are expected to drastically affect the way people perceive mobility, something that can result in further pressures on the already stressed transport system. The mitigation against future mobility challenges requires that transport authorities and operators will have at their disposal appropriate modelling tools to accurately assess the effects of travel behaviour change on the transport landscape.

Transport modelling has traditionally aided the understanding and the estimation of travel behaviour for individuals, groups, and the masses. Nonetheless, the rapidly changing transport environment requires for even more sophisticated transport modelling approaches. Disaggregate modelling paradigms such as agent- and activity-based microsimulation are promoted as the most suitable methods to address the future challenges of transport. These approaches can explicitly model the complex interactions between agents in dynamic environments (e.g. transport system) and allow for the emergence of complicated behaviours which could have not manifested in aggregate modelling paradigms. Despite their advantages, the wider adoption of disaggregate models has been considerably hindered by the scarcity of detailed mobility data at the disaggregate-level. However, the emergence of various urban sensing data sources such as GPS traces and Call Detail Records (CDRs) has started countering data scarcity issues related to the modelling of personal mobility. Nonetheless, well-justified privacy concerns have raised the need for cautious treatment of

such sensitive information. One of the most common approaches to ensure the intractability of the tracked users is the aggregation of data. In the context of mobility, the movements of individuals with similar origin and destination are often aggregated and presented in the so-called Origin-Destination (OD) matrices.

The use of ODs as an anonymization means represents only a fraction of their actual use. OD matrices have traditionally constituted a fundamental element/'devise' of transport modelling, allowing the concise and accurate representation of various dimensions of travel demand (e.g. trip purpose, time period of departure, etc.) other than the origin and destination of trips. Their concise and straightforward form has facilitated the transferability of information and has established them as the main data exchange format within the transport community. On the other hand, ODs have not been widely utilised for the study of personal mobility because their aggregate nature deprives them from the ability to directly represent complex travel behaviours such as trip-chaining and group travel. From that aspect, the development of a methodology able to enhance ODs by disaggregating them into personal mobility traces is significant.

1.2 Objectives

OD matrices constitute an irreplaceable component of transport planning and modelling. Countless transport authorities, operators, practitioners, and researchers have over the years allocated significant resources for the development and maintenance of ODs to support a plethora of decisions related to urban planning, policy evaluation and transport infrastructure investment. More recently, ODs have been also extensively utilised by urban sensing data providers as an efficient and privacy safe means for the presentation of peoples' traces.

Despite the wide range of applications concerning ODs, little effort has been devoted to their disaggregation and the exploitation of the rich information laying within in OD records. The present Ph.D. research aims to enhance ODs by suggesting a methodological framework that enables the disaggregation of ODs to individual mobility traces and unveil a significant amount of additional travel-related information not directly provided in typical ODs. In particular, the Ph. D. Thesis emphasises on the evaluation of the potential to:

- **Use of ODs for the studying of travel behaviour and mobility at the person-level.** Reasons of anonymity dictate that the individual traces of users must be aggregated before being presented to ensure privacy and intractability. Given the increasing concerns regarding data privacy as well as the rapid introduction of relevant legislations (GDPR, APPI, OAIC etc.) the chances of obtaining, even for pure

academic purposes, disaggregate mobility information are constantly thinning. Therefore, a methodology allowing the privacy-safe study of mobility using aggregate ODs can prove of significant importance.

- **Use of ODs to produce highly detailed input for disaggregate transport models.** Methodologies utilising ODs for the synthesis of disaggregate mobility traces can significantly increase their value and further justify past and future investments.

1.3 Approach

The current Ph.D. Thesis proposes a novel methodological framework for the synthesis of disaggregate mobility traces from aggregate ODs. More precisely, the methodology delves in the synthesis of personal activity schedules from the individual trips contained within multi-period and trip-purpose segmented ODs.

The framework is founded upon the observation that the vast majority of people begin and end their daily activity schedules at their home (or residing location), after the completion of a series of home-based trip-chains (i.e. tours). Assuming that accurate ODs containing all the individual trips required for the completion of the previously mentioned tours suggests that there must exist at least one combination of the captured trips which recreates the tours. Additionally, in the case where the inputted ODs contain trip-purpose information, it is reasonable to claim that such information can be utilised for the inference of the executed activities and consequently enable the conversion of tours to the more informative form of activity schedules.

The methodology is developed upon advanced graph-theoretical and integer/combinatorial mathematical optimisation concepts and is completed in a modular fashion:

1. The *Graph-generation module* utilises the spatiotemporal information present in the input ODs to create a suitable graph supporting the application of graph-theory-based methodologies. In particular, the methodology introduces the hybrid Time Varying Graph (hTVG) which enables the presentation of mobility patterns captured by multiple time-dependent ODs in a concise and integrated manner. This is achieved by firstly expressing the individual multi-period ODs as discrete graphs and subsequently layering and connecting them in a chronologically ordered fashion.
2. The *Identification module* capitalises on the conversion of the inputted ODs to a hybrid Time Varying graph (hTVG) to express tours as paths originating and ending at the same vertex (home location). A suitably modified algorithm is assigned with the enumeration of all the plausible tours within the hTVG. In detail, the algorithm

is sequentially applied on each vertex with the aim to enumerate all the paths originating from and ending at it. The completion of the enumeration results in the creation of the *candidate tours set*.

3. The *Activity-scheduling module* exploits the trip-purpose information in the ODs to convert the candidate tours into *candidate activity schedules*. The conversion can take place because trip-purpose segmented ODs contain information regarding the type of activity executed at the ends of each trip. Although, typical ODs do not contain information regarding the sequence under which activities take place, a suitable algorithm exploits the interdependency between the individual trips within a tour to convert the latter in the more informative form of *activity schedules*.
4. Finally, the *Optimisation module*, deploys an advanced integer/combinatorial optimization procedure for the identification of compositions of candidate activity schedules and their volume which replicate the travel demand expressed in the input ODs. The objective of the optimization is to minimise the number of unused trips from the input while any available calibration data (e.g. departure time profile, diurnal distribution of activities, other survey-based data) can be enforced as constraints for the enhancement of the realness of the resulting activity schedules.

Despite the sound theoretical foundation, the integer/combinatorial nature of the formulation and in particular the potentially excessive number of plausible tours in large-scale graphs, require appropriate techniques to ensure the scalability of the methodology. Two additional simplification modules were developed for the reduction of the problem's dimensionality and the solution of the problem in reasonable time.

i. The first simplification module (*Graph-filtering module*) achieves the reduction of the number of plausible tours in a hTVG via the simplification of the graph's structure. A suitable simplification process removes vertices which present limited effect to the traversability of the graph as well as limited travel demand volume.
ii. The second simplification module (*Candidates-filtering module*) evaluates all the candidate activity-schedules in terms of their likelihood of being observed in real-world contexts and subsequently discards rare ones (e.g. activity schedules with excessive number of activities or with inordinate total travel time).

Both simplification modules, although not mandatory to the methodology, can be applied individually or complimentary to allow the addressing of scenarios requiring the extraction of individual mobility traces from very large-scale ODs. The methodological framework developed for the purposes of this Ph. D. Thesis is illustrated in Figure 1.1.

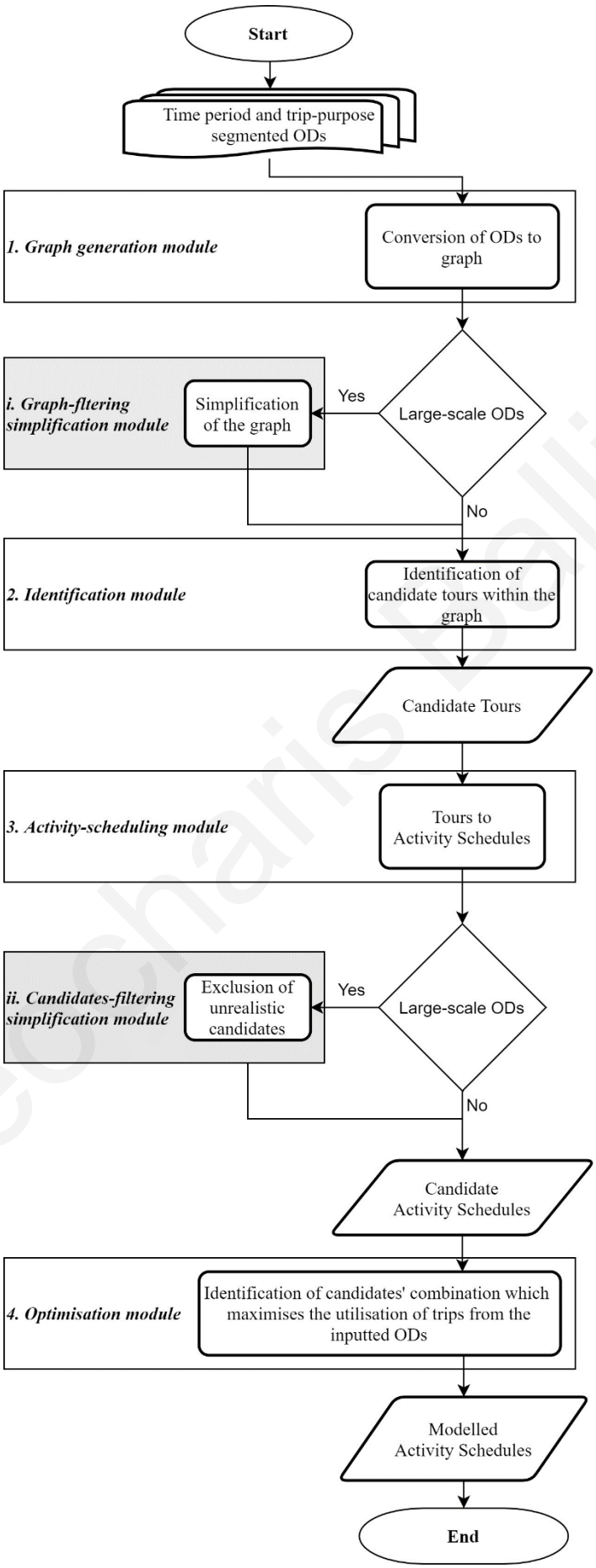

Figure 1.1 Flowchart of the suggested methodological framework.

1.4 Outline

The multidisciplinary methodological approach developed for the purposes of this Ph. D. Thesis is thoroughly presented in the following Chapters, structured as follows:

Chapter 2 is devoted to the presentation of the relevant literature regarding the understanding of travel behaviour for the synthesis of disaggregate mobility information. In addition, Chapter 2 includes an elaborate presentation of the characteristics of the Origin-Destination matrices which constitute the primary input of the proposed methodology.

Chapter 3 hosts a detailed presentation of the four modules constituting the core of the methodology, namely the *Graph-generation module*, *the Identification module*, the *Activity-scheduling module*, and the *Optimisation module*.

Chapter 4 presents the suggested approach to address complexity issues involved in the problem and render it applicable for real-world cases. In particular, it presents two additional simplification modules which ensure the scalability of the methodology without jeopardising the quality of the solution.

Chapter 5 provides details regarding the followed approach to enable the large-scale optimisation, required for the completion of the proposed methodological framework. In addition, the Chapter presents the Adaptive Sampling Simulated Annealing (ASSA) optimisation algorithm developed for the purposes of the study.

Chapter 6 presents the proof of concept of the proposed methodology performed over a set of ODs deriving from a large number of observed activity schedules. Furthermore, in Chapter 6 the analysis goes into greater depth to evaluate the methodology from multiple perspectives as well as to showcase the additional travel behaviour insights which can be drawn when aggregate ODs are converted into individual activity schedules.

Chapter 7 evaluates the scalability of the methodology through its application on a set of large-scale ODs deriving from hundreds of thousands of individual activity schedules. This Chapter also verifies the effectiveness of the ASSA algorithm for the addressing of integer/combinatorial problems of excessively large size.

Chapter 8 summarises the findings of the thesis, highlights its contribution and suggests the areas for future research.

Chapter 2

Literature Review

Chapter 2 is devoted to the presentation of the relevant literature regarding the understanding of travel behaviour for the synthesis of disaggregate mobility information. In addition, the Chapter includes an elaborate presentation of the characteristics of the Origin-Destination matrices which constitute the primary input of the proposed methodology.

2.1 Modelling Mobility

2.1.1 Changing Travel Behaviour

Transport modelling has traditionally faced significant challenges to accurately estimate and predict mobility. This can be attributed to a plethora of reasons, such as the questionable modelling paradigms, reliance on cross-sectional data, inadequate sampling, and coarse zoning systems among many others (Hartgen, 2013). The modelling of mobility is founded upon the understanding of the driving forces behind travel behaviour. For decades, a plethora of travel behaviour studies have aimed at the provision of both long- and short-term guidance for urban planning and transportation development (Wang et al., 2018). According to recent studies, travel behaviour is getting more and more difficult to predict (Pinho and Silva, 2015; Pawlak et al., 2019) with various reasons contributing to this, including the modern lifestyle (Ferreira et al., 2007; García-Jiménez et al., 2014), the increased demand for autonomy (Gardner and Abraham, 2007), flexible work arrangements (Brewer, 1998), the diverse travel patterns as well as the increasing availability of transportation alternatives for transport-deprived population groups (Steg, 2003; Hopkins et al., 2019). Another factor affecting travel behaviour stems from the widespread introduction of Intelligent Transportation Systems (ITS) which allow travellers to make better informed decisions regarding their journeys (Emmerink et al., 1994; Srinivasan and Mahmassani, 2000; Guilherme and Soares, 2013). Due to the availability and accuracy of real-time information, travellers now have the ability to optimise their journeys and optimally adjust their travel patterns, often leading to unexpected travel behaviours (Pel et al., 2012; Antoniou and Spyropoulou, 2014). Our inability to understand the motives behind the change of travel behaviour will most likely further complicate the efforts around mobility modelling (Wegener, 2013).

2.1.2 Evolving Transport Landscape

The current transportation trends indicate that mobility in the future will be considerably different. Advances in technology such as wireless connectivity and improvements in the sector of telematics, have enabled the better coordination between demand and supply provision. This fact has been translated into the introduction of flexible and on-demand mobility solutions, also known as New-Mobility-Services (NMS). NMS can be configured in multiple ways and allow people to access transport modes (e.g. car, bicycles, vans, etc.) for as long as they need them, reducing the nuisances attached to vehicle ownership (Franco et al., 2019). These new services blur the lines between private and public transport by

combining the flexibility of being able to access your own vehicle with the lower cost associated with mass transit options (Hall et al., 2018; Smith et al., 2018). Their most widespread examples are those of ride-hailing, ride/car/bike-sharing, and shuttle bus services. However, NMS are just part of the greater change that Mobility-as-a-Service (MaaS) is expected to bring. MaaS has the potential to merge multiple NMS in a complimentary fashion which can address transport needs more efficiently compared to the provision of transport services in isolation. As the name suggests, the MaaS vision expects mobility to be provided as service and that people will solely rely on mobility providers for their transport needs (Giesecke et al., 2016). As with most services, consumers will be presented with bundles of different options and will be able to choose the one which satisfies them the most. The customers will be primarily interested in fulfilling their need of reaching the destination under a set of personal criteria (e.g. time, cost, comfort, etc.) and will be probably indifferent to the transport modes involved to complete the journey as long as those criteria are met (Dacko and Spalteholz, 2014). Travellers in the MaaS era are likely to perceive travelling substantially differently than they do today (Durand and Harms, 2018). Contrary to the current situation where mobility options are generally limited, travellers of the future will be exposed to a larger variety of transportation alternatives (Kamargianni and Matyas, 2017). However, the radical change of the transport landscape is expected with the wide introduction of Autonomous and Connected Vehicles (CAVs) vehicles. CAVs will most likely redefine the rules of mobility and as a consequence tremendously influence the way people perceive travelling (Fagnant and Kockelman, 2014; Holmberg et al., 2016; Pavone, 2016). For instance, the flexibility and ease of use of autonomous vehicles could allow mobility deprived sections of the population (e.g. the elderly, children, etc.) to reconsider their mobility needs and increase their travel footprint (Harper et al., 2016). The constantly evolving transportation system in conjunction with the changing travel behaviour can put transport networks over their available capacity and render them unable to provide quality services.

Based on the above, it becomes apparent that modelling paradigms aiming at the preparation against the forthcoming challenges and the assessment of possible mitigation strategies are of paramount importance (Goulias and Barbara, 2009). The next section delves in the presentation of the current state of the mobility modelling landscape as well as its future direction towards disaggregate approaches.

2.2 Mobility Modelling Approaches

2.2.1 Aggregate Approaches

The typical approach to model mobility encompasses the use of aggregate transport models. Traditional aggregate transport models (such as the widely used 4-step model) have proven invaluable tools towards the understanding of transportation trends and have tremendously supported crucial strategic decisions. However, the future mobility challenges are calling for the development of disaggregate modelling paradigms, better suited to capture individuals' travel behaviour and complex multiagent systems (Goulias, 2009). This statement has been supported by many researchers who have emphasised the need of incorporating user-centricity in the core of transport modelling for mitigation against future mobility challenges (Ben-Akiva et al., 2007; Hilgert et al., 2016). However, the introduction of user-centricity in typical aggregate (i.e. 4-step) transport models can prove troublesome since their aggregate nature hinders the representation of travel behaviour at person-level (Mladenovic and Trifunovic, 2014). Traditional aggregate transport models were not designed with the aim to model individual responses of travellers at the disaggregate level, therefore they are less capable of evaluating the travellers' reaction to personalised transport services such as NMS (Pinjari and Bhat, 2011).

Aggregate transport models typically apply analytical relationships, based on closed form mathematical equations to estimate total travel demand between pairs of zones (Ben-Akiva et al., 2007). Attempting to include the individual characteristics of each user to the modelling framework of 4-step models would require the impractical introduction of a unique user class for each traveller. On the other hand, disaggregate paradigms are considerably more capable of incorporating user-centricity into transport and travel behaviour modelling since different agents can be natively modelled.

Apart from the negative implications of using typical aggregate models for the prediction of travel behaviour under the emerging flexible and on-demand transport services, difficulties also arise for the modelling of the operational side of transport (Horn, 2002; Segui-Gasco et al., 2019). The future of transport promises seamless journeys which can possibly require multiple complementing modes and cooperating transport operators (Kamargianni et al., 2015; Smith et al., 2018). Such a complex system will not operate efficiently support from sophisticated modelling tools able to incorporate highly dynamic networks, inhomogeneous fleets, and continuously changing travel behaviours (Cich et al., 2017). In addition, these modelling tools should be also able to predict future conditions and suggest appropriate

mitigation strategies in case of emergencies or unforeseen events. From the transport supply perspective aggregate transport models are in a weak position (compared to disaggregate models) to effectively model the dynamic and multipart operations manifesting in the emerging transportation landscape (Cich et al., 2016).

2.2.2 Disaggregate Approaches

Disaggregate transport modelling approaches such as agent-based modelling, activity-based modelling, microsimulation, etc., have the potential to effectively and accurately simulate complex systems (Ben-Akiva et al., 2007) but only recently have started to gain momentum (Zhang and Levinson, 2004; Ronald et al., 2015; Djavadian and Chow, 2017). Under the disaggregate modelling context, interactions and interrelations between agents are modelled explicitly and dynamics emerge as the aggregation of all the decisions taken by autonomous agents (Bonabeau, 2002; Azevedo et al., 2016). This disaggregate representation allows the emergence of complex behaviours between agents which would have been difficult to be expressed in an aggregate modelling environment (Borshchev and Filippov, 2004).

Despite the considerable potential of disaggregate models for the accurate simulation of complex dynamic environments, their requirement for fine-grained information has hindered their wider adoption (Molla et al., 2017; Bassolas et al., 2019). However, the introduction of a reasonable decision-making logic to the agents of a simulation requires a wide breadth of information (Angria S et al., 2018; Aziz et al., 2018). Since decisions depend on the characteristics of the agents as well as on their perception about the surrounding environment (Zhu et al., 2007), the representativeness and the accuracy of the decision-making process is dictated to a large extent by the quality and the volume of the available input data (Klügl, 2010). The more detailed the characteristics of the simulated agents and environment, the more likely the disaggregate model to accurately replicate reality. Despite the recent increase in data availability, the acquisition of precise data describing all the above-mentioned aspects, especially at person-level, can still prove challenging (Wise et al., 2017). However, the transition to the Big-Data age promises the reduction of data scarcity issues (Katrakazas et al., 2019) mainly due the penetration of technology in people's everyday life (Calabrese et al., 2013; Anda et al., 2016). In addition, since urban planners and policy makers have already highlighted the importance of information flow for future Smart Cities (Batty, 2013; Kitchin, 2014; Beckwith et al., 2019; Bouzidi et al., 2020), the amount of available mobility related information is expected to rise. The acquisition of rich data at large scale can aid the wider adoption of disaggregate transport models able to allow the transition to the new mobility era.

2.3 Mobility Modelling Data Requirements

2.3.1 Population Synthesis

The application of disaggregate transport models for the analysis of travel behaviour relies heavily on a wide range of data including the socio-demographic and the economic attributes of the simulated population as well as their detailed travel diaries. Since the acquisition of the complete socio-demographic profile for every person in the studied area can prove prohibitively expensive, if not impossible (E. Ramadan and P. Sisiopiku, 2019), the field of population synthesis provides an alternative for the creation of fully detailed disaggregate populations based on openly available sources (e.g. census, land-use data, anonymised travel diaries, etc.). The field has drawn considerable attention over the last 30 years and numerous methodologies have been already presented (Muller, 2010). The most frequently used approaches are based on the Iterative Proportional Fitting (IPF) method (Beckman et al., 1996; Choupani and Mamdoohi, 2016) although other alternatives have been also evaluated. For example, Ye et al. (2009) proposed the Iterative Proportional Updating (IPU) method in order to counter the inefficiency of IPF's algorithms to control for person-level attributes and joint distributions of personal characteristics. Additionally, combinatorial optimisation (Abraham et al., 2012) and Markov process-based approaches (Farooq et al., 2013; Saadi et al., 2016b) have also presented significant advantages. Population synthesisers are focusing on the assignment of socio-demographic attributes to each of the simulated persons and usually disregard the mobility behaviour of the population. A notable exception stands for the study of Saadi et al. (2016a) who combined a simulation-based synthesiser with a Hidden Markov model to assign activity schedules to the synthetic persons. Apart from the required population, access to mobility data (e.g. travel diaries, traces) is also essential for the development of advanced travel behavioural modelling frameworks (Rashidi et al., 2017) and as a consequence significant research has been devoted on the issue.

2.3.2 Activity Scheduling

Understanding mobility is a particularly active research topic which stretches among multiple disciplines (Bhat and Koppelman, 1999; Bowman and Ben-Akiva, 2000; Axhausen, 2007; Gonzalez et al., 2008). Travel behaviour theory accepts that travelling is a derived need required for the completion of activities (Mokhtarian and Salomon, 2001; Donnelly, 2010), therefore, mobility can be interpreted through the study of activity scheduling (Jovicic, 2001). The term 'activity scheduling' refers to the act of designing the schedule which will allow a person to complete all her/his required activities, usually within the course

of a day. This design corresponds to all the associated aspects such as the number of executed activities, their locations, durations, sequence, type, etc.

With regards to disaggregate transport modelling, the main question to be answered is how can detailed activity schedules be accurately estimated for every person in the population under study. Before delving into the suggested approaches for the understanding of activity scheduling, it should be stated that although activity patterns may seem random and unpredictable, research suggests that activity scheduling can be predicted with accuracy (Gonzalez et al., 2008; Song et al., 2010; Schneider et al., 2013). Multiple approaches have been suggested, ranging from statistical methods to Markovian Chains, Principal Component Analysis, Machine Learning, Network Analysis and Sequence Alignment Methods. The selection between the available options is strongly influenced by the type and the availability of the input data. While the first suggested approaches relied on traditional travel surveys, more recent studies have taken advantage of new urban sensing technologies like Mobile Network Data (MND), GPS traces and smart-card data to name a few. The common factor between all the above-mentioned methodologies is their dependence on disaggregate data sources except few exceptions (Ballis et al., 2018; Huber and Lißner, 2019; Ballis and Dimitriou, 2020a, 2020b). The next section describes the so far suggested approaches for the production of disaggregate mobility data (e.g. trip-chains, activity schedules, etc.) depending on the type of the data input.

2.3.2.1 Survey Data

Prior to the emergence and the wide availability of urban sensing information, researchers had traditionally relied on travel surveys to conduct analysis regarding personal mobility (Calabrese et al., 2013; Yue et al., 2014). A non-exhaustive but representative collection of such examples is offered here. Bowman and Ben-Akiva (2000) utilised the Boston 1991 survey to develop an econometric model to impute personal day activity schedules. Schoenfelder and Axhausen (2001) proposed the use of survival analysis theory for the identification of rhythmic patterns based on a long-term survey of 316 participants over a course of six years. In 2007, Lee et al. (2007) developed simultaneous, doubly-censored Tobit models to estimate the relationships between household type and structure, time allocation strategies, and trip-chaining patterns, using data from the 2000 Tucson Household Travel Survey. Nurul Habib (2011) developed a random utility maximisation framework for the modelling of dynamic weekend activity scheduling based on information available in the CHASE survey collected for the Toronto area in 2002. More recently, large scale travel surveys, requiring sophisticated analysis approaches, have also become available. For

example, Jiang et. al. (2012) applied Principal Component Analysis on a large travel survey including more than 30 thousand individuals to explore daily activity structures and cluster them based on socio-demographic information. According to their results, seven to eight groups are adequate for the representative classification of individual activity patterns. Similarly, a network-based approach to identify and categorise activity patterns of individuals was presented by Zhang and Thill (2017). In their research, Zhang and Thill provide a methodology for the clustering of travellers in 'community structures' where individuals in the same community tend to interact more intensively compared to agents belonging to different communities. The potential of their methodology to classify large datasets of space-time trajectories was evaluated using 9000 individual travel spanning across Carolina, USA.

A widely examined stream of research aiming at the studying of activity scheduling is developed based on the Sequence Alignment Method (SAM). Although SAMs were originally developed to study DNA sequences, they have been extensively utilised for the study of the sequential dependencies between daily activities (Wilson, 1998; Joh et al., 2002). These approaches attempt to classify activity-chains (usually obtained from travel surveys) into clusters based on their sequencing characteristics and composition. Despite their wide spread, SAMs have been criticised for their inability to capture infrequent activity patterns (Liu et al., 2015; Saadi et al., 2016a). Nonetheless, improvements based on Markovian approaches have been suggested. For instance, Liu et al. (2015) used a profiling method called profile Hidden Markov Models (pHMM) to enable the capturing of the irregular activity patterns. Likewise, Saadi et al. (2016a) combined the pHMM method with a population synthesiser to develop a framework capable of assigning activity sequences to all the agents of a population.

Activity scheduling has been also studied under the prism of hazard-based methodologies (Ettema et al., 1995; Bhat, 1996; Schoenfelder and Axhausen, 2001). Hazard-based methodologies appreciate that the duration between the duration of participation of an individual at the same activity (e.g. work, shopping, leisure, etc.) depends on the elapsed time since the last participation. As an example, in the work of Bhat et al. (2005), a sophisticated multivariate hazard model was developed and applied on a multi-week survey for the cities of Halle and Karlsruhe, Germany. The results indicated distinct weekly rhythms for individuals participating in social, recreational, and personal business activities. Manual surveys are still the most effective mean to acquire precise information regarding the travel behaviour of individuals. Nonetheless, extensive manual surveys are costly, require

significant time for preparation and are not easily updated. For that reason, researchers have started exploring the use of passively collected data for the studying of activity-scheduling.

2.3.2.2 Urban Sensing Data

Non-invasive, automated, continuous data collection technologies are increasingly used to complement manual survey techniques as well as to improve the statistical representativeness of traditional surveys (Cottrill et al., 2013). The significant role of these modern urban sensing data sources (e.g. Mobile Phone Data, GPS traces, transit smart-cards, etc.) in the study of travel behaviour has been explored by numerous researchers (Caceres et al., 2013; Calabrese et al., 2013; Yue et al., 2014; Çolak et al., 2015; Vlahogianni et al., 2015; Bassolas et al., 2019). Based on the obtained literature, the synthesis of mobility information from urban sensing sources for the fuelling of advanced transport modelling, has been attempted by two main approaches. The first approach (analytical methods) suggests the analysis of disaggregate mobility data (e.g. travel diaries, GPS traces, etc.) for the creation of travel behaviour models, able to produce the required input for disaggregate transport models. The second approach (obfuscation methods) suggests the obfuscation of the raw personal mobility data so that the anonymity of the tracked user can be guaranteed and therefore the data can be directly used for modelling purposes. The next section delves in the presentation of the so far suggested methodologies in both fields.

Analytical methods

The availability of vast quantities of data obtained by urban sensing sources has ignited a significant amount of research with regards to travel behaviour. Amongst urban sensing data sources, the most widely used for the analysis, clustering and estimation of activity schedules are Mobile Network Data (MND), GPS traces, and smart-card transit data (Toole et al., 2015; Anda et al., 2016; Antoniou et al., 2019) with their potential having been evaluated in many studies. For example, Ebadi et al. (2017) constructed spatiotemporal 'activity-mobility trajectories' based on a small (37 smart-cards) but detailed smart-card dataset, obtained from students at the University of Buffalo. Their results presented a prediction accuracy between 75 to 88%, showcasing that smart-card data can be utilised for the accurate estimation of activity recognition. a large sample of smart-card data obtained from the London's public transport network was utilised by Goulet-Langois et al. (2016) for the identification of travel behaviour heterogeneity between public transport users. The researchers firstly inferred a 4-week continuous activity sequence for each of the smart card holders and then clustered them into 11 distinct sequence structures. Sociodemographic information of a small sub-sample allowed them to identify significant connections between the activity sequence structures

and the characteristics of individuals. Smart-card data have been also used as input for hidden Markov Chain models. For instance, Han and Sohn (2016), relied on smart-card data and land-use information for the transit network of Seoul to impute activity chains using a continuous hidden Markov model. The modelled results yielded plausible and intuitive activity patterns which were also consistent with observed activity patterns.

Between the available urban sensing data, MND are gradually becoming the main source of travel behaviour information, mainly due to their relative low-cost, large sample size and extended spatial coverage (Pan et al., 2006; Chen et al., 2016; Ni et al., 2017). In particular, large volumes of Call Detail Records (CDRs) obtained from mobile phones are often used to construct individual daily itineraries and train travel activity models using weeks and months of data rather than several days' worth (Widhalm et al., 2015). Despite their advantages, CDRs are not explicitly designed to fuel travel behaviour analyses, therefore they do not include significant travel behaviour dimensions such as the type of activity executed by the mobile phone users. For that reason, analysts have attempted to infer the type of the executed activity mostly through rule-based approaches (Chen et al., 2014). An attempt to improve the activity type estimation by combining Points of Interest (POIs) datasets with CDRs, is presented by Phithakkitnukoon et. al. (2010). In that research the probability to execute a certain type of activity was calculated based on the number and the type of POIs laying inside each of the modelled areas. A mechanistic approach to synthesise urban mobility profiles through the exploitation of data generated by communication technologies (i.e. MND) is presented by Jiang et al. (2016). Jiang et al. utilised MND to model the location and the duration of primary (Home and Work) as well as secondary (e.g. Other) activities using a rank-based Exploration and Preferential Return (r-EPR) mechanism. Furthermore, the use of MND derived trip-chains as input for a microsimulation agent/activity-based model has been explored in Zilske and Nagel (2015). The researchers' results indicate that MND combined with other sources of information such as traffic counts can provide valuable input to simulation models. On a similar stream, Liu et al (2014) explored and verified the potential of utilising MND to validate activity-based models. As a last example of MND-based studies, Eagle and Pentland (2009) relied on Principal Component Analysis to identify the behavioural structure of 100 users, carrying their mobile phones for 9 months. The main aim of their study was to identify a set of characteristic vectors (i.e. patterns), termed as 'eigenbehaviors', which can approximate the individual's actual behaviour. According to their research, utilising just six eigenbehaviors can

approximate individual's travel behaviour with 90% accuracy. For the interested reader, an exhaustive survey on the MND applications can be found in the study of Blondel (2015).

The increasing availability of mobility related data has led researchers to the development of models able to identify patterns and connections between the system state variables (i.e. inputs and outputs) without explicit knowledge of the analysed system. These so-called Data-driven models promise to minimise uncertainty and improve accuracy by fusing and integrating multiple sources of (dynamic) data into the core of (transport) modelling (Jha, 2015; Angria S et al., 2018; Antoniou et al., 2019). As an example, Liu et al. (2013) employed multiple Machine Learning (ML) algorithms on a dataset covering a year of MND for 80 users. The supervised ML algorithm was trained with the 2.3% of the locations in the dataset where the respective activities performed at these places were known. According to the researchers, the prediction accuracy of the model reached a remarkable 70% which was further increased to 77% after the application of a post processing algorithm. The use of ML algorithms for the identification of a Markov model's parameters is presented by Allahviranloo and Recker (2013). They employed their methodology and showcased the supremacy of the ML-based methodologies against a standard multinomial logit model. A data-driven modelling framework for the estimation of human mobility trajectories has been presented by Pappalardo and Simini (2018) where observed MND data were utilised to construct individual diaries based on an Exploration and Preferential Return methodology. The comparison of their results against observed data showcased the capability of the methodology to accurately reproduce the statistical properties of the observed trajectories. Finally, a prominent methodology providing anonymised and fully detailed activity schedules from MND is presented by Lin et al. (2017). The authors first utilise an Input-Output Hidden Markov Model (IO-HMM) to infer activity sequences and subsequently apply a Long Short Term Memory (LSTM) deep neural network for the assignment of exact locations to the previously identified activities. The framework presented reasonable performance when 465 thousand synthetic activity schedules were assigned in a multi-modal, micro simulator model and the observed traffic and transit counts were compared against the corresponding modelled figures. A more direct approach for the obtainment of mobility data at large quantities compared to their extrapolation from samples is the anonymisation of mobility traces.

Obfuscation methods

The abundance of information in the Big-Data era, has the potential to alleviate data scarcity issues and to provide researchers with substantial quantities of information. Nonetheless,

privacy concerns will most likely still be present (Batty et al., 2012), therefore methodologies ensuring the anonymity and the quality of data are very important. Nowadays, multiple entities (e.g. Mobile Phone Carriers, transport providers, smartphone app developers, etc.) record large volumes of mobility traces at high resolution. Nonetheless such data can be very rarely provided as is without some form of anonymization process, since the mobility footprint of people can be particularly distinct (De Montjoye et al., 2013). Various methodologies have been suggested for the achievement of what is often referred as differential privacy. In a nutshell, differential privacy requires that the probability distribution on the published results of an analysis is "essentially the same," independent of whether any individual opts in to, or opts out of, the data set (Dwork et al., 2010). Despite, the widespread research in relation to ensure differential privacy as well as the obfuscation of mobility traces (You et al., 2007; Krumm, 2009; Suzuki et al., 2010; Kato et al., 2012; Shokri et al., 2012; Bindschaedler and Shokri, 2016), no standard procedure has been established so far. One of the most common approaches to guarantee intractability is the aggregation of mobility traces with similar characteristics (e.g. similar origin). For example, the study of Balzotti et al. (2018) conducted a travel behavioural analysis using only aggregated cellular network data (in the form of hourly counts of mobile phones in predefined zones) without subjecting the tracked users at risk. Another frequently deployed methodology for the construction of privacy-safe traces is based on generative models (Chow and Golle, 2009; Krumm, 2009; Kato et al., 2012; Shokri et al., 2012; Bindschaedler and Shokri, 2016). These models utilise observed traces to create realistic trajectories with similar semantics while at the same time ensure intractability through Location Privacy Protection Mechanisms (LPPMs). LPPMs rely on a wide range of techniques including data perturbation (Andrés et al., 2013), data encryption (Mascetti et al., 2011) and fake data generation (Pelekis et al., 2011). For example, Isaacman et al. (2012) introduced a probabilistic modelling framework (coined as WHERE) to produce synthetic Call Detail Records (CDRs) while Mir et al. (2013) enhanced the framework by adding a differential privacy mechanism (DP-WHERE) to guarantee privacy-preservation. The interested reader can find an extensive review of relevant data anonymisation techniques in (Primault et al., 2019).

Finally, it should be emphasised that despite the wide range of data anonymisation techniques which have been so far suggested, the standard approach for the presentation of MPD Data is through aggregate Origin-Destination (ODs) matrices (Caceres et al., 2007; Bonnel et al., 2015; Tolouei and Alvarez, 2015). ODs ensure anonymity through the

segmentation of the mobility traces into individual trips and the aggregation of these trips into groups with similar characteristics (e.g. trip-purpose, time period of departure, mode of transport, etc.). The previous sections presented various approaches enabling the production of disaggregate mobility input for advanced transport modelling. Contrary to the previously presented studies, the novel approach presented in this thesis depends solely on aggregated travel demand data (i.e. OD matrices and high-level distributions) instead of disaggregate information. The next section concludes the literature review by providing the necessary background regarding Origin-Destination matrices (ODs) which constitute the basic input for the methodological framework presented in this Thesis.

2.4 Origin-Destination Matrices (ODs)

2.4.1 Development

Despite the advances in mobility tracking technology and the availability of relevant information in a plethora of data sources (e.g. Call Detail Records, GPS Traces, etc.), the most widely used mean to represent travel demand is still the standard form of Origin-Destination matrices (ODs). ODs have traditionally constituted a fundamental element of transport modelling and it may not be an overstatement to claim that the majority of transportation related projects involves at some point their use (Montero et al., 2019). In their simplest form, ODs represent mobility as the total volume of movements between pairs of locations. In practise, the studied area is divided into multiple smaller areas which are usually referred as zones. The purpose of these zones is to aggregate areas with similar characteristics into larger spatial units and therefore divide the continuous space in discrete segments. Once the zoning system has been defined, the flows between zones can be expressed via a square matrix where rows and columns correspond to the available zones in the area. This straightforward structure has facilitated the transferability of results and has established ODs as the main travel demand data exchange format in the transport community. However, information regarding the volume of demand between locations is not adequate to allow for in-depth analysis and additional dimensions are required to achieve so. For that reason, ODs are often segregated by dimensions such as the purpose of the executed trips, the used transport mode, and the time period of departure, etc. in order to enable the obtainment of a more complete picture regarding the mobility motif within the studied area. The aggregate nature of ODs dictates that all flows within an OD are homogeneous, therefore each of the different dimensions must be expressed via a different OD (Donnelly, 2010). For

example, the morning-peak flows cannot be separated from the corresponding flows for the evening-peak if not presented in two separate ODs.

Over the years, transport authorities and operators have allocated significant resources to the development and maintenance of OD matrices to support a plethora of decisions related to urban planning, policy evaluation and transport infrastructure investments (Peterson, 2007; Ickowicz and Sparks, 2015). However, the estimation of accurate OD matrices is an extremely challenging task since very often the data used for ODs estimation is limited. Since the recording of all the movements taking place in the studied area is infeasible, various methods have been suggested from the accurate estimation of ODs based on partial observations. The most common approach to derive ODs is through the combination of roadside interviews (RSIs) and the application of trip-end and gravity models (to extrapolate and infill unobserved movements), followed by matrix estimation methods for the incorporation of supplementary traffic counts (Iqbal et al., 2014). Various methodologies have been suggested including Bayesian methods (Maher, 1983; Li, 2005), Generalised Least Squares (Cascetta, 1984; Bell, 1991; Nie et al., 2005; Y. Wang et al., 2016), Maximum Likelihood (Spiess, 1987; Ickowicz and Sparks, 2015) and Entropy Maximisation (Van Zuylen and Willumsen, 1980). More recent approaches have relied on urban sensing data sources (Zhao et al., 2007) such as MND (Alexander et al., 2015; Bonnel et al., 2015; Horn et al., 2017; Tolouei et al., 2017), GPS traces (Parry and Hazelton, 2012; Ge and Fukuda, 2016), smart-card data (Jun and Dongyuan, 2013) as well as combinations between those (Toole et al., 2015). The study of Antoniou et. al. (2016) has proposed a common evaluation framework to enable the standardised comparison between different OD estimation methodologies.

Despite the long history of OD matrix development, the research on the field is still particularly active and no sign indicates that it will be ceased in the (near) future. Therefore, methodologies aiming at the exploitation and the enhancement of ODs, such as the one presented in this Thesis, can be considered of significant value.

2.4.2 Usage

The previous section was devoted to the presentation of the importance of ODs and the various methodologies which have been suggested for their creation. Once an OD matrix has been built it can be utilised to inform a wide range of transportation modelling related tasks. From a transport planning perspective, an OD can provide useful insight regarding the attractiveness of certain areas or pinpoint pairs of location with significant demand for travel,

now or in the future. Additionally, ODs can help with the short-term management of the network (Zhou et al., 2003; Sundaram et al., 2011) or with long-term strategic decisions such as the planning of the public transport network (Borndörfer et al., 2005). One of the most common usages of ODs is traffic assignment (Antoniou et al., 1997; Peeta and Ziliaskopoulos, 2001; Maerivoet, Sven; De Moor, 2006; Balakrishna et al., 2007; Nagel and Flötteröd, 2009; Bekhor et al., 2011). Evidently, traffic assignment models (dynamic or static) cannot be executed if the origin and the destination of the inputted trips is not known, therefore OD matrices constitute an essential input for such purposes. Even though OD matrices are perfectly suitable for the aggregated representation of travel demand, their aggregate nature forbids them from representing the interdependency between trips often manifesting as trip-chains or tours (Pendyala and Goulias, 2002; Mcnally and Rindt, 2008).

2.4.3 Trip-chaining

The significance of trip-chaining for travel behaviour analysis has drawn considerable attention over the years (Thill and Thomas, 1987; Goulias and Kitamura, 1991; McGuckin and Murakami, 1999; Yue et al., 2014). Despite the incapability of OD matrices to represent trip-chaining and trip-interdependency phenomena, many researchers have suggested approaches to incorporate such elements into the OD estimation process. Some of these studies have focused on the exploitation of trip-chaining information obtained from automated data collection sources (e.g. smart-cards) in order to enhance the accuracy of the transit ODs estimation (Wang et al., 2011; Jun and Dongyuan, 2013). A different stream of methodologies has expressed trip-chains as Markov chains with the purpose to convert data obtained from traffic flows to ODs (Morimura et al., 2013; Tesselkin and Khabarov, 2017). Additionally, efforts to incorporate trip-chaining information in an dynamic OD estimation framework have been also presented (Lindveld, 2003; Flötteröd et al., 2011). More recently, Cantelmo et al. (2019), suggested the use of an online dynamic OD estimation framework which combines a departure time choice model with a Kalman Filter to identify correlation between different OD pairs in space and time.

Even if trip-chaining has been considered during the synthesis of ODs, their aggregate format deprives them from the ability to represent such information. The study of Abdelghany et al. (2007) exemplifies this limitation by executing a traffic assignment exercise using trip-chains and an equivalent scenario where trips are assigned individually. According to their results, the total travel time for the case where trip-chaining was ignored increased by 20% compared to the opposite scenario. Transport modelling paradigms such as activity-based modelling attempt to counter this limitation by expressing travelling

behaviour as the series of interrelated trips (i.e. trip-chains) required to complete an activity schedule (Bhat et al., 2004; Pinjari and Bhat, 2011; Chu et al., 2012). Representing travel demand through trip-chains is more flexible and better suited for the purposes of disaggregate modelling but the expression of mobility through trip-chains can prove an expensive, tedious and complex task (Gu, 2004; Ben-Akiva et al., 2007). Therefore, the aggregation of trip-chains to form ODs which are more easily handleable and managed is a typical approach.

The previous section showcased the importance of trip-chaining information as well as various methods for its incorporation in the OD estimation process. Nonetheless, to the best knowledge of the author, no study has attempted to convert aggregate ODs into trip-chains.

2.4.4 Disaggregation

Based on the previously presented literature review, it becomes evident that the study of activity scheduling and the preparation of disaggregate mobility data has been primarily based on the analysis and the exploitation of disaggregate inputs, however some exceptions do exist. As an example Balmer et al. (2006) suggested a framework capable of combining multiple sources of information, including OD matrices, to generate disaggregate travel demand data (in the form of trip-chains), for the purposes of a large scale microsimulation. That study was based on a mechanistic approach which iteratively subtracted trips from an OD to recreate activity schedules retrieved form a relevant survey. The main drawbacks of that methodology are the reliance on exogenous data for the formation of activity schedules and the rather simplistic mechanism for the utilisation of trips from the input ODs.

The increasing requirements for high precision, disaggregate mobility information, in conjunction with the data-privacy regulations (e.g. General Data Protection Regulation, Japan's Act on Protection of Personal Information, etc.) which promote the aggregated publishing of information (e.g. ODs) has led researchers to experiment with data disaggregation methodologies. Recently, Huber and Lißner (2019) utilised aggregate cycling data obtained from the Strava app to synthesise disaggregate mobility data. Their approach applies a double-constrained routing algorithm on aggregate OD cycling demand to derive single bicycle routes. However, their model does not aim at the reproduction of the cycling travel demand through individual cycling traces but rather on the development of a bicycle route choice model based on the OD information. The possibility of synthesising travel demand based on aggregated data from TSPs has been recently evaluated by Anda et al. (2020). Their Markovian-based approach allows the synthesis of realistic daily tours using

aggregate joint distributions (histograms) which can be provided by TSPs since they are considerably less likely to raise data-privacy concerns. All the different model architectures were evaluated over a large dataset of 1 million synthetic travellers and resulted in remarkably high accuracy (≥ 95%) in terms of replicating the observed travel patterns. A potential drawback of the methodology is the reliance on multiple and very detailed hourly distributions at zonal level (e.g. duration of stay time in a zone by hour, number of people transitioning to a previously unvisited zone by zone and departure hour, etc.).

To the best knowledge of the author, except from the above-mentioned studies and the relevant work supporting this Ph. D. Thesis (Ballis et al., 2018; Ballis and Dimitriou, 2020c, 2020a, 2020b), no other study has attempted the disaggregation of ODs for the synthesis of mobility data at person-level. However, the currently suggested methodological framework attempts to fill this gap and provide a comprehensive framework for the synthesis of highly detailed, disaggregate mobility information based on the widely available data source of OD matrices.

2.5 Research Needs

The previously presented literature review emphasised the need for disaggregate modelling approaches to efficiently address forthcoming mobility challenges. The so far suggested modelling paradigms rely on a wide range of very detailed mobility information, usually at person-level but such data are difficult to be acquired mainly due to reasons of anonymity and cost. On the other hand, aggregate data sources are usually more easily available but lack in terms of representativeness and detail, therefore they are less useful for the study of travel behaviour at person-level. According to the previously presented literature review, the potential of utilising aggregate mobility data sources, in particular ODs, has not been examined thoroughly enough up to now. Based on this observation, the following research needs have been identified:

- The exploitation of the ubiquitous, aggregate OD matrices for the study of complex travel behaviour phenomena (e.g. trip-chaining, activity-scheduling), especially at person-level, has not been thoroughly investigated.
- The potential of synthesising disaggregate mobility data from aggregate data sources should be further explored.
- Not adequate research has been devoted on the development of a flexible methodological framework, able to disaggregate commercially available, aggregated mobility data in a privacy-safe fashion.

The currently presented Thesis attempts to fill the above-mentioned gaps by evaluating the potential of utilising aggregate data sources for the synthesis of representative mobility information at the person-level. Amongst the available aggregated data sources describing mobility, OD matrices are the most widely used, therefore the most promising candidate for this purpose. Their continuous development indicates that their usability and value will not diminish in the foreseeable future, hence the investment towards their enhancement is well justified. To achieve so, a novel methodology is proposed for the exploitation of the spatiotemporal as well as the trip-purpose information in typical ODs to synthesise highly detailed disaggregate mobility data. The detailed methodology to enable this conversion is meticulously presented in the next Chapter.

Chapter 3

Methodology

Chapter 3 hosts a detailed presentation of the four modules constituting the core of the methodology, namely the Graph-generation module, the Identification module, the Activity-scheduling module, and finally the Optimisation module.

3.1 Overview

In the current Thesis a novel methodological framework is proposed for the synthesis of disaggregate activity schedules based on multi-period, purpose-dependent OD matrices. The main principle supporting the proposed framework is the observation that the majority of the population in a region begins and ends their daily activity schedules at home or residing location (Bowman, 1998; Schneider et al., 2020). The methodology is based upon the assumption that if all the trips captured in multi-period ODs belong to tours, then there must exist a combination between the captured trips that recreates the ODs. This assumption holds particularly true in cases where the OD matrices have derived from observational data sources (e.g. mobile phone data, GPS, etc.). These ODs are usually built by tracking the movements of individual people for consecutive days or even months. Therefore, such ODs are indeed formed as the aggregation of the trips belonging to tours. However, even in cases where ODs have stemmed from modelling processes (e.g. typical 4-step models), and therefore flows are not entirely based on consistent observations, the fact that most trips within ODs should belong to tours, still holds true. The aim of the methodology is to reconstruct input ODs into the travel demand equivalent tours. Although, not a prerequisite for the application of the methodology, information regarding the purpose of each trip (i.e. trip-purpose) can be utilised to transform the tours into the more meaningful and contextual form of activity schedules. For the brevity of the presentation, the onwards sections assume the presence of trip-purpose information within the utilised ODs.

The identification of activity schedules within multi-period, purpose-segmented ODs is accomplished in a modular fashion. Firstly, the *graph generation module* handles the conversion of the inputted ODs into a suitable graph. Secondly, the graph-theory-based *identification module* completes the identification of all the plausible tours within the graph. Thirdly, the *activity scheduling module* exploits the available trip-purpose information to convert tours into activity schedules. Finally, the *optimisation module* identifies the combination of tours whose enclosed trips recreate the inputted travel demand (i.e. input ODs). In case that calibration information regarding the expected schedules are known, the optimisation module attempts the identification of a solution adhering to the calibration data.

In short, the methodology is accomplished as follows (also depicted in Figure 3.1):

1) Conversion of multi-period, purpose segmented ODs into a graph (*Graph-generation module*)
2) Identification of all the plausible tours in the graph (*Identification module*)

3) Exploitation of trip purpose information to convert tours into activity schedules (*Activity-Scheduling module*)
4) Identification of the activity schedules' combination which maximises the utilisation of the inputted OD trips while respecting any available calibration information (*Optimisation module*)

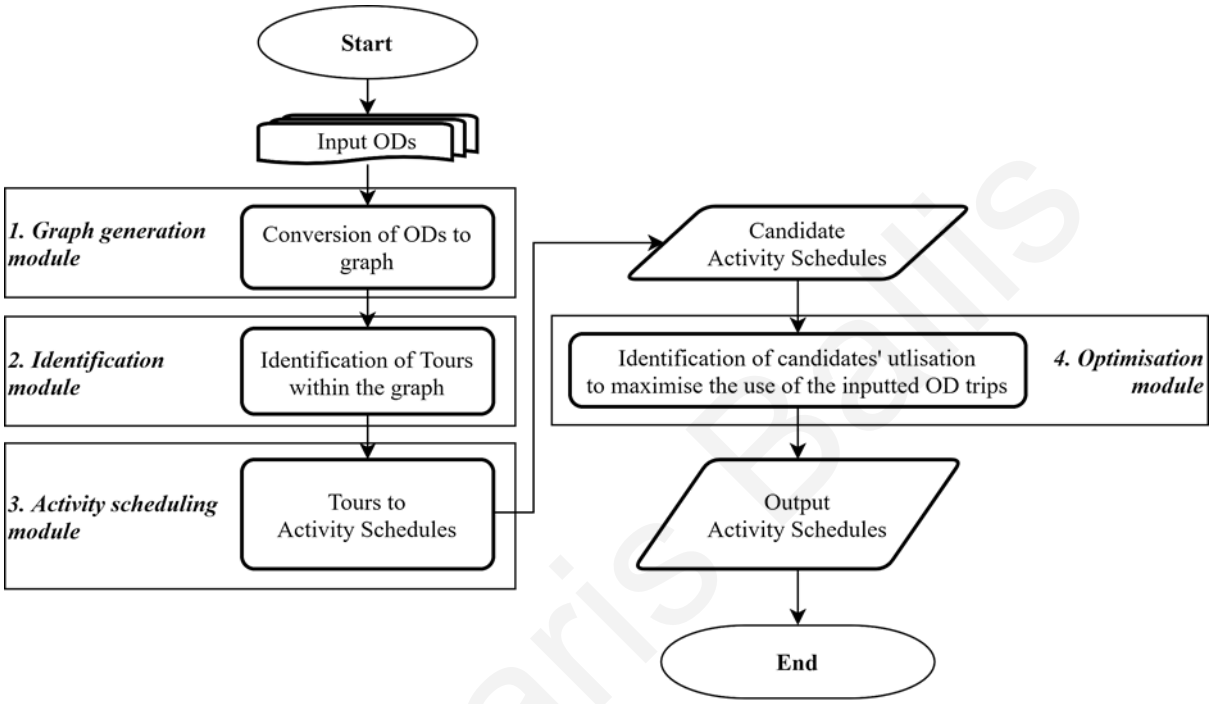

Figure 3.1 Flowchart depicting the suggested methodology.

For the ease of understanding an example case presenting the expected input OD matrices along with the corresponding output can be found in Table A.1 and Table A.2 of the Appendix. The next sections thoroughly describe the modules required for the completion of the proposed methodology.

3.2 Graph-generation Module

3.2.1 Multi-period OD Matrices to hybrid Time Varying Graph (hTVG)

The first step of the proposed methodology entails the conversion of multi-period, purpose segmented ODs into a suitable graph for the application of advanced graph theoretical algorithms. A typical representation of a graph can be accomplished by a tuple $G = (V, E)$ where V is the set of vertices (nodes) and E the set of edges (links). This type of representation is very suitable to model situations where relationships between nodes are static. Representing travel demand using graphs is a well-documented approach (Wood et al., 2010). However, most studies either neglect the temporal dimension of demand (Phan et al., 2005) or utilise multiple but isolated networks for the representation of different demand

states (Von Landesberger et al., 2016). Nevertheless, travel demand unravels as a highly dynamic phenomenon and therefore it could be more appropriate to model and analyse it as such. In the past few years, intensive research has been allocated on methodologies capable of handling dynamic networks, also known as Time Varying Graphs (Wang et al., 2019). Following the definition given by Casteigts (2018) a Time Varying Graph (TVG) can be defined as a tuple $\mathrm{G} = (\mathrm{V}, \mathrm{E}, \mathrm{T}, \rho, \zeta)$ where V, E stand respectively for the nodes and the edges of the network and T represents its lifetime (Time Domain). The dynamic nature of the network is handled by parameter ρ which denotes the presence of each edge $\mathrm{e} \in \mathrm{E}$ at a given time $\mathrm{t} \in \mathrm{T}$. Finally, ζ constitutes the latency (i.e. cost) to traverse each edge in E. Based on this, it becomes apparent that multi-period OD matrices can be expressed as TVGs where the zones, trips, time periods and travel-costs constitute the corresponding nodes, links, time domain and latency of the TVG. TVGs have already exhibited their useful properties in numerous studies (Cheng et al., 2003; Ferreira, 2004; Kostakos, 2009) but they can still can prove cumbersome to model and manipulate (Casteigts, 2018). On the other hand, standard static networks have been thoroughly studied for many decades and therefore very robust and efficient methodologies have been developed for their analysis. The suggested framework counters the complexity of TVG's by adopting a hybrid solution referred as the hybrid TVG (hTVG).

A hTVG combines the dynamic properties of TVGs with the simplicity of static graphs by expressing the temporal changes as a series of interconnected and chronologically arranged static graphs, also known as snapshots (Wehmuth et al., 2015) . The following section describes the proposed methodology to convert multi-period ODs into a hTVG. In this type of graph, each of the available multi-period ODs is expressed as a separate layer allowing the distinction of trip departures taking place at different time periods. Following this multilayer network format, the spatial characteristics across layers (i.e. the location of nodes on the XY plane) remain stable but the connections between nodes can vary, allowing the emergence of variant connectivity patterns across time. Nonetheless, without further modification, nodes on different layers are isolated and therefore no paths traversing across different time periods would be able to be formed. To address this issue, nodes representing the same spatial location in consecutive time periods are connected by a special type of links referred as *temporal link* (Lin et al., 2016). Temporal links do not represent a movement in space nor time but are solely used to enable the forward in time transition between consecutive layers. The above-mentioned conversion process is clearly illustrated in Figure 3.2. In the single layer graph (Figure 3.2a), the spatiotemporal information is expressed on

one level. According to this layout, two nodes become connected once a trip takes place between them, regardless of its departure time. Nonetheless, the graph generation module disentangles the temporal information into multiple layers (Figure 3.2b) and allows for a more detailed representation of the system. The process is completed by the insertion of the temporal links which can be distinguished by the gold cones notating their direction. As it can be observed, the initially fully connected graph is converted to a more informative equivalent which better represents the temporal dimension of travel demand.

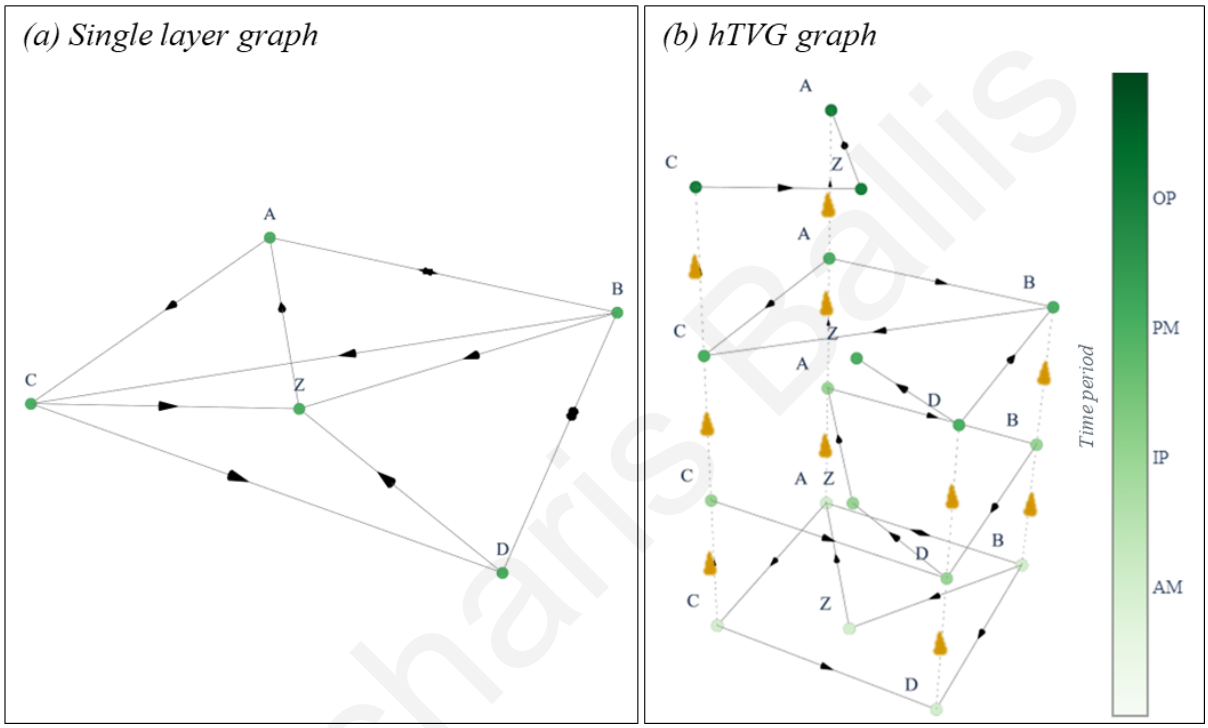

Figure 3.2 Conversion of (a) a single layer graph (b) to a hTVG.

The following section summarises the advantages accompanying hTVGs with regards to transport modelling as well as to the here presented methodological framework.

3.2.2 Advantages of hybrid Time Varying Graphs (hTVGs)

The previously presented graph formation exhibits some significant advantages. Firstly, hTVGs achieve the encoding of temporal elements directly into a static-like graph, suitable for the application of efficient graph-theory-based methodologies. The most beneficial effect is that in contrast to single layer graphs, hTVGs do not allow the formation of chronologically inconsistent paths. The reason is that the presence of the chronologically directed temporal links forbids the creation of unchronological paths. Secondly, the identification of closed paths connecting the same spatial location is more straightforward in the hTVG case. In graph-theory a cycle represents a closed path with the same origin and destination where no nodes other than the origin can be repeated more than once. Although

the cycles identification problem has been thoroughly studied for static graphs, this is not the case for TVGs (Kumar and Calders, 2018). The hTVG format, alleviates this issue by substituting cycles with simple paths connecting the same location in different time-periods. The former two advantages are illustrated in Figure 3.3 where a tour from begins from zone Z in the morning (AM) and finishes at night (OP). As it can be observed, according to the hTVG format, a tour can be straightforwardly expressed as a simple path. The figure also clearly illustrates the achieved chronological consistency due to directionality of temporal links (depicted with gold cones). Thirdly, hTVGs can represent more eloquently the temporal variability of networks in terms of travel times, cost, dynamic tolls, etc. Consequently, any analysis affected by the dynamic nature of the networks (e.g. shortest path identification) is considerably more accurate when executed in a hTVGs.

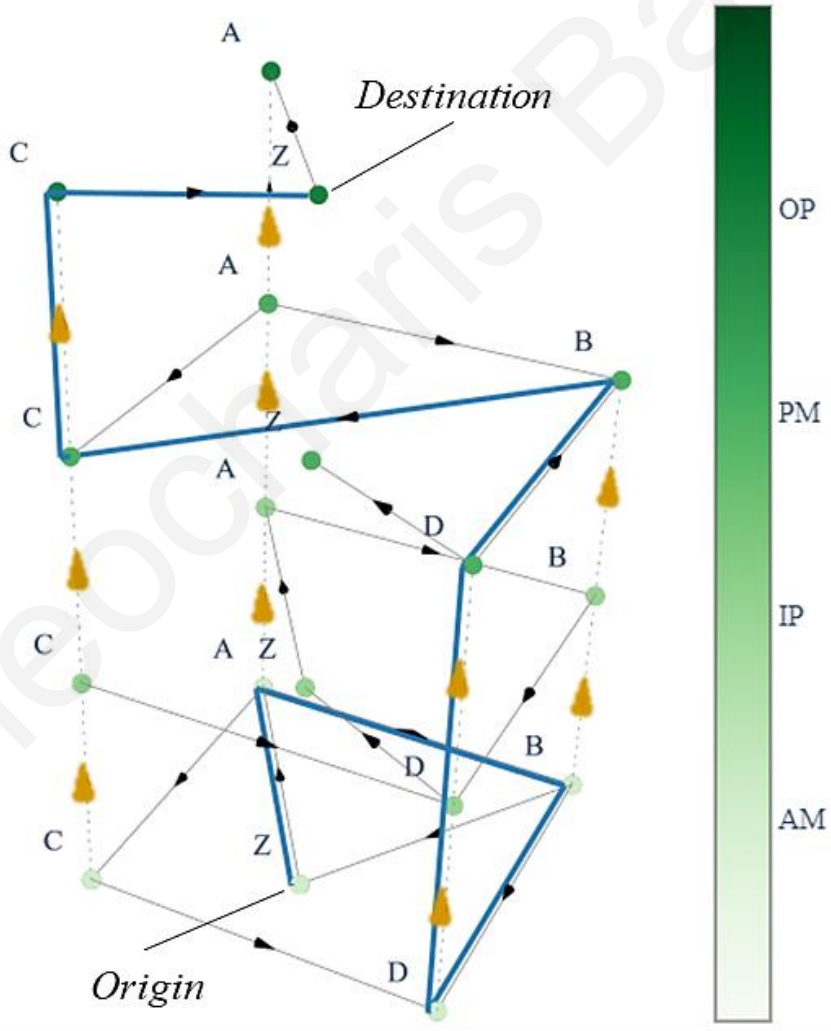

Figure 3.3 Formation of a chronologically consistent tour in a hTVG.

The transformation of the single layer network to a hTVG comes with an additional benefit specific to the suggested methodology. As it has been already discussed in one of the authors' previous studies (Ballis and Dimitriou, 2019), one of the key factors affecting the

performance of the identification module is the spatiotemporal resolution of the provided network. Based on that study, high-resolution networks result in precise trajectories which can be more easily traced and identified. The quantification of the network resolution and the associated analytical complexity is expressed through the proxy of *network density*. The higher the density of a network, the higher the number of potential paths and consequently the higher the required computation time to identify the full set of candidate tours. Nonetheless, hTVGs present lower density compared to their single-layer equivalents something that leads to reductions of the required processing time. In particular, the network density (d_s) for a directed, single layer network is calculated as the fraction between the actual number of edges E in a graph of V vertices and the number of its plausible edges:

$$d_s = \frac{E}{V(V-1)} \qquad \text{(Eq. 3.1)}$$

For the equivalent hTVG, the network density (d_h) is calculated as:

$$d_h = \frac{\lambda E + k(V-1)}{kV(k(V-1))} \qquad \text{(Eq. 3.2)}$$

where k stands for the number of time periods (layers) of the hTVG and λ ($1 \leq \lambda \leq k$) denotes the increase in the number of links due to the replication of links across multiple layers. Also, the $k(V-1)$ factor represents the maximum number of temporal links which are required to complete the conversion of a single layer graph to a hTVG. The density of an hTVG is maximised when all nodes are connected with their counterparts in the next layer ($\lambda = k$).

Proposition 1. The density of realistic, single layer transport networks is greater than the density of the equivalent hTVG (i.e. $d_h \leq d_s$) when $k \geq 1 + \frac{V}{E}$.

Proof. Substituting Eq. (1) and Eq. (2) in $d_m \leq d_s$ results to $\frac{kE+k(V-1)}{kV(k(V-1))} \leq \frac{E}{V(V-1)}$. For large networks it can be assumed that $V-1 \approx V$, therefore $\frac{kE+kV}{k^2V^2} \leq \frac{E}{V^2} \Rightarrow \frac{E+V}{k} \leq \frac{E}{1}$. Finally, solving with respect to k leads to $k \geq 1 + \frac{V}{E}$, which for realistic transport networks holds true since edges are usually at least one order of magnitude more than the nodes (Barabási, 2016). The reduction of density for TVGs has been also experimentally verified by Santoro et al. (2011)

3.3 Identification Module

3.3.1 Expressing tours as graph paths

The conversion of the input ODs to a hTVG allows the identification of all the plausible tours within the graph in an efficient and eloquent manner. As it has been already discussed, a tour is defined as a sequence of trips originating and ending at the same (home) location. According to the hTVG format, zones in different time periods are represented as separate entities (nodes), therefore tours can be expressed as simple graph paths (i.e. sequences of nodes with no repeats) connecting the same zone across different time periods (Figure 3.4). As a result, their identification can be achieved through standard and very efficient path identification algorithms (Sedgewick, 2001). An exception to this procedure is the case of tours starting and ending within the same time period. For these instances, the standard operation of *cycles identification* is employed (Johnson, 1975).

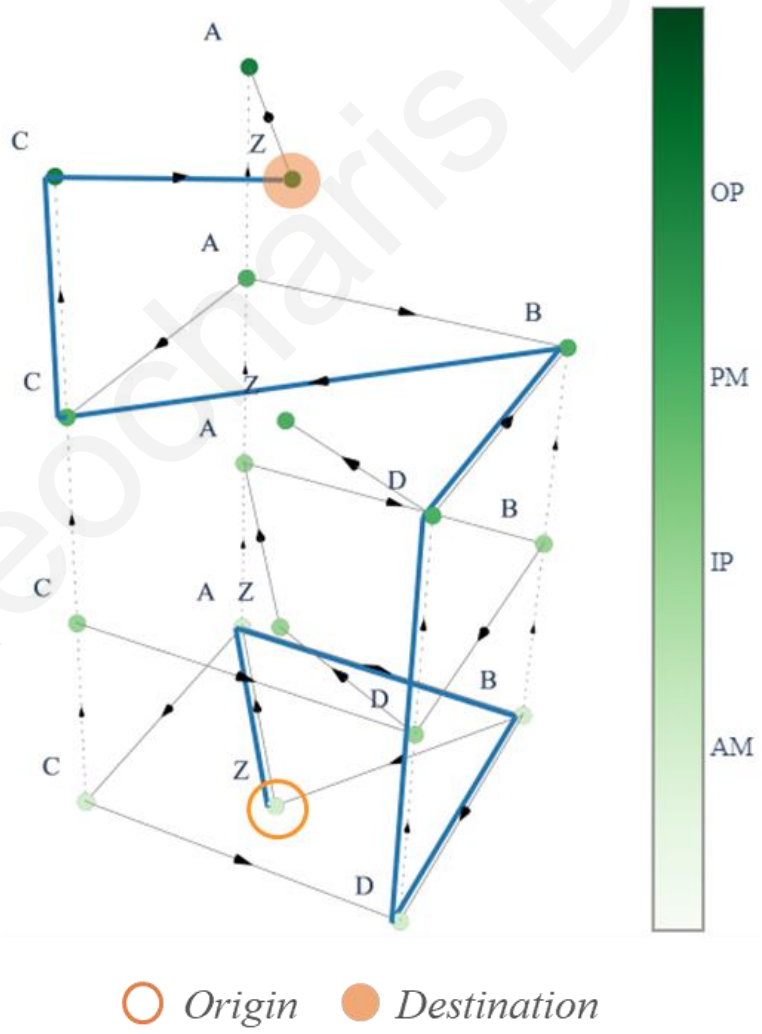

Figure 3.4 Identification of a tour in a hTVG network.

3.3.2 Identification of all possible paths

The retrieval of all the possible tours within a hTVG requires the execution of the path identification algorithm multiple times. In detail, for each zone in the ODs, the process is applied $\sum_T(t-1)$ times, where T is the number of time periods in the input ODs. The origin-destination pairs required for the application of the path identification algorithm are formed by connecting each zone in the current time period with the corresponding ones in all the consecutive time periods. As stated above, an exception to this procedure is the case of tours starting and ending within the same time period where the standard operation of cycles identification is employed. A visual example is presented in Figure 3.5 where zone Z appears in four time periods. As it can be observed, the identification process must be executed ten times (six times as a simple path and four times as a cycle identification procedure) to identify all the chronologically ordered tours originating from zone Z. The full set of plausible tours is obtained by repeating the path identification process for all the zones in the input ODs across all the available time periods.

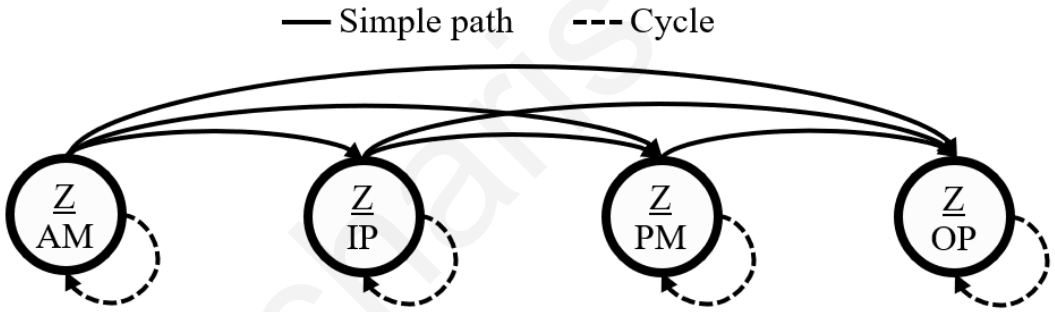

Figure 3.5 Presentation of the valid time-period combinations for the identification of all the chronologically ordered tours.

Once the process has been completed and all the tours within the hTVG have been identified, the methodology can continue with the conversion of tours to the more informative structure of activity schedules. However, this conversion is optional and can be omitted in the case where no trip-purpose information is available within the input ODs. In that scenario the inputted ODs are converted to tours instead of activity schedules.

3.4 Activity-scheduling Module

3.4.1 Trip-purpose Information in ODs

The previously presented section described the required procedure to identify tours within multi-period OD matrices expressed as a hTVG. However, ODs often contain additional information regarding dimensions of travel such as trip-purpose, transport mode, user-group,

etc. For the purposes of the presented study, the focus has been placed on cases where trip-purpose information in addition to departure time-period is also provided in the input ODs. These two travel behaviour dimensions are particularly important since they can be utilised to convert tours into detailed activity schedules.

The segmentation of OD trips based on their trip-purpose is common practice since trip-purpose is a primary driver of travel decision influencing multiple travel behaviour aspects such as destination choice, mode choice, the value of time, etc. The typical categorisation of trips with respect to trip-purpose usually refers to two discrete levels. The first level is with regards to the inclusion (or not) of the traveller's home at either ends of the trip (Home-Based/Non-Home-Based trips). The second level is related to the main purpose each trip is taking place for (work, education, shopping, employer's business, etc.). Nonetheless, a serious limitation for most ODs is that they do not include information regarding the sequencing of activities at the ends of each trip (Ortúzar and Willumsen, 2011). For example, a Home-Based-Work trip can be used to either express the transition from home to work or vice versa. The methodology presented in the following paragraphs presents an approach to exploit trip-chaining behaviour for the inference of the type of activity at the ends of trips.

3.4.2 Tours to Activity Sequences

A basic assumption in travel behaviour theory reads that trips are regarded as the necessary mean to enable the transition from one activity to the next. Therefore, trip-purpose information can be utilised to infer the executed activities at the ends of trips and subsequently enable the conversion of tours into sequences of activities. Tours resulting from the so far presented process contain the required information to enable this conversion. The methodology exploits the fact that activities within a tour take place in a sequential and closed loop fashion, therefore the ambiguity regarding the sequence of the activities can be eliminated (Figure 3.6). This is further elaborated through the following example.

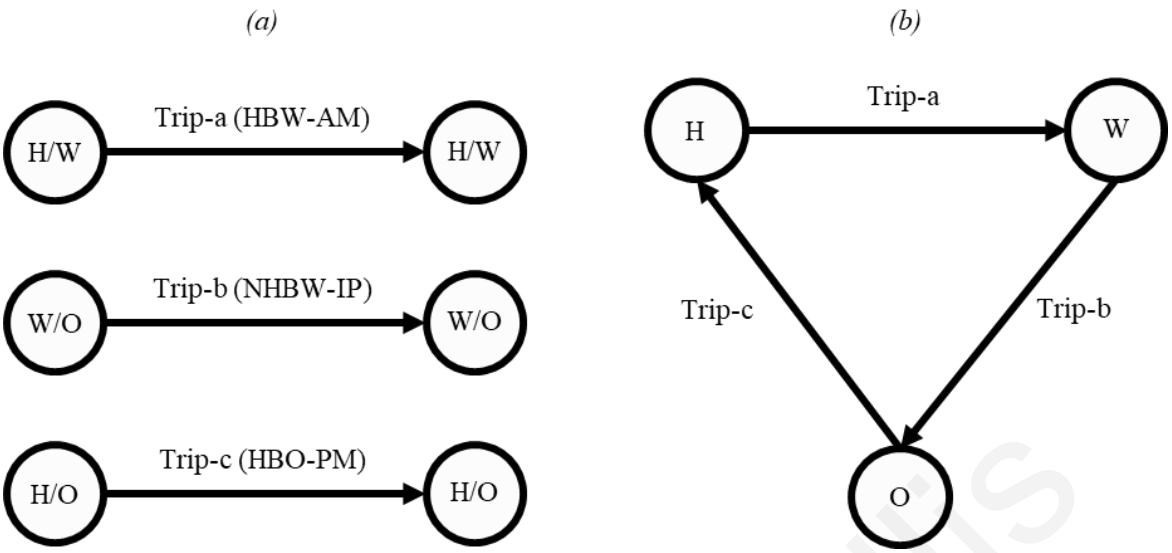

Figure 3.6 Chaining of individual OD trips eliminates the ambiguity regarding activity sequencing. (a) Unchained trips (b) Chained trips.

Assume a set of four different ODs used to segregate trips according to their trip-purpose, namely:

- Home-Based (HB): The activity at one end of the journey is staying at home (Home) while at the other end is either:
    - Work (HBW)
    - any Other (HBO)
- Non-Home-Based (NHB): None of the activities at either ends of the trip is Home.
    - if the activity at one end is Work then the trip is classified as (NHBW) while
    - for all Other cases as (NHBO).

Consider also a 4-leg tour which contains two HB and two NHB trips. As described above, traditional ODs do not provide information regarding the sequence of activities at the ends of each trip. Nonetheless, combining the individual trips into tours, allows for the elimination of activity sequencing ambiguity. As noted in the first row of

Table 3.1, a tour cannot consist of a HBW trip followed by a NHBO, then a NHBW and a final returning to home HBO trip because no valid combination of the enclosed activities can be formed. The initial HBW trip can only signify a transition from Home to Work and consequently the next trip should connect Work with the subsequent activity. However, a NHBO trip does not include the Work activity in its definition, hence this trip-purpose sequence is not valid. On the other hand, the rest of the presented trip-purpose sequences are valid and should not be eliminated. The application of this methodology to all the candidate tours results in a set of candidate activity sequences.

Table 3.1 Identification of activity sequences from trip-purpose sequences.

Trip-purpose sequence	**Activity sequence * ****	**Valid**
[HBW, NHBO, NHBW, HBO]	[(H | W), (O | O), (O | W), (O | H)]	No
[HBW, NHBW, NHBO, HBO]	[(H | W), (W | O), (O | O), (O | H)]	Yes
[HBO, NHBO, NHBW, HBW]	[(H | O), (O | O), (O | W), (W | H)]	Yes

* The pipe symbol ('|') denotes the 'OR' operator
** H=Home, W=Work, O=Other

The implications of this observation are significant since it enables the enrichment of typical ODs with information regarding the sequencing of activities for the captured trips. Once the methodology is completed, the individual trips utilised to synthesise tours can be enriched with information regarding the sequence of the activities they connect.

3.4.3 Activity Sequences to Activity Schedules

The activity sequences obtained from the previously presented methodology can be further enriched with information regarding the time of departure from each activity. Since the time period of departure for all the inputted trips is known, an estimation regarding the exact time of departure can be attempted. Evidently, the error of this estimation diminishes with the increase of the temporal resolution in the original ODs (i.e. the number of available time periods). The estimation of the exact departure time from each activity can be completed with simple (e.g. uniform distributions) or more refined approaches (e.g. travel survey based), depending on the required level of detail or information availability. This estimation can be further improved by considering the travel time required to reach the locations at the ends of each trip. Ultimately, the assignment of the exact time of departure for each trip, allows the conversion of results in fully detailed activity schedules. The implications are significant since the initially aggregate OD trips can be now expressed in a much more detailed, decomposed, informative and contextual manner.

After the assignment of the exact departure time from each activity within the activity schedules, the latter can be fully defined as sequences of (a) visited locations, (b) departure times (or time periods) and (c) activities. An example of a typical activity sequence is depicted in Figure 3.7 where a traveller departs from Home in zone-Z at 07:45 (AM), executes consecutively two short activities of type Other in zone-A until 08:05 (AM) and in zone-B until 08:25 (AM) respectively and finally leaves Work from zone-C at 17:30 (PM) to return back Home in zone-Z. The traveller can be tracked both in space and time since the

vertical dimension (z-axis) is used to represent time duration. The example activity schedule is also presented in Table 3.2.

Table 3.2 Definition of an example activity schedule as sequences of various types.

Activity sequence*	**Locations**	**Departure time-periods**	**Departure times**
[H, O, O, W, H]	[Z, A, B, C, Z]	[AM, AM, AM, PM]	[07:45, 08:05, 08:24, 17:32]

* H=Home, W=Work, O=Other

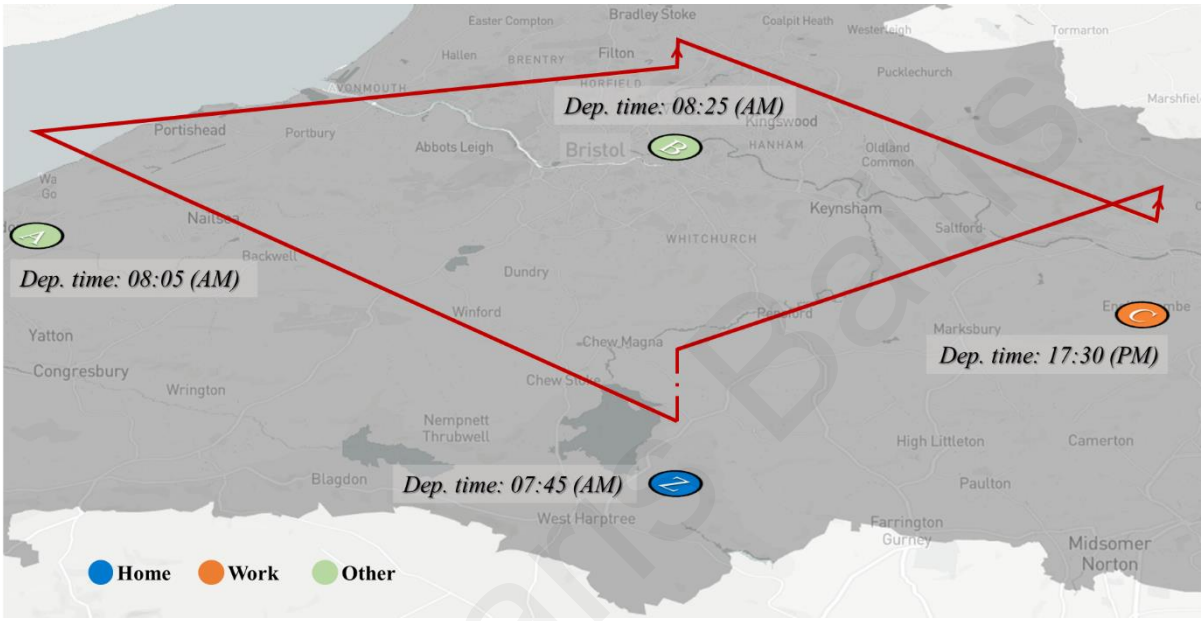

Figure 3.7 Visual representation of a typical activity schedule.

3.5 Optimisation Module

The identification of all the possible activity schedules within a hTVG deriving from a set of multi-period, purpose segmented ODs allows to proceed with the optimisation part of the methodology. The aim of the identification module is to identify the combination of activity schedules whose included trips recreate the input ODs as closely as possible. The following section presents two mathematical formulations resolving the combinatorial problem in hand. The first formulation (*Exact mathematical programming formulation*) describes the problem in mathematical programming terms, calling for the suitable analytical optimisation routine (e.g. branch-and-bound, cutting-plane, branch-and-cut, etc.) while the second formulation (*Metaheuristics formulation*)calls for metaheuristic optimisation approaches (e.g. Simulated Annealing, Genetic algorithms, etc.). Despite addressing the same problem, the two suggested formulations differ considerably. Firstly, the two approaches are supposed to address different in terms of scale instances of the problem. The first programming formulation, although more accurate, cannot deal with particularly large instances of the

problem where the metaheuristics formulation can prove as an effective approach for realistically sized cases. Secondly, the approaches also differ from a technical perspective. The exact mathematical programming alternative attempts the identification of the optimum frequency of use (utilisation) of each candidate schedule which recreates the input ODs, while the metaheuristics formulation iteratively builds a population of schedules to achieve the optimisation objective. Finally, an important differentiation concerns the way the two methods deal with the calibration information (if any is provided). The first formulation imposes the calibration as a hard constraint to the optimisation of the problem, while the second incorporates the calibration information during the space sampling procedure. The following section delves in the presentation of the two alternative formulations.

3.5.1 Exact Mathematical Programming Formulation

Table 3.3 Nomenclature for exact mathematical optimisation methods

Variable	Description
$\mathbf{C}$	Candidate activity schedules ($c \in C$)
$\mathbf{K}$	Available time periods ($k \in K$)
$\mathbf{P_k}$	Zone-pairs in each k ($p_k \in P_k \; \forall \; k \in K$)
$\mathbf{T_{p_k}}$	The number of trips between each p_k, as recorded in the input ODs
$\mathbf{I}$	Distribution groups ($i \in I$)
$\mathbf{D_c^{p_k}}$	Binary variable indicating whether p_k is part of c
$\mathbf{G_c^i}$	Binary variable indicating whether c belongs to i
$\boldsymbol{b_c^i}$	The probability of c to belong in i
$\boldsymbol{\delta_i}$	Maximum percentage error between the input and the modelled probability for each i
$\boldsymbol{N_c}$	The frequency of usage for each c

3.5.1.1 Formulation

Let C be the set of unique candidate activity schedules ($c \in C$). The aim of the optimisation problem is the identification of the frequency of use (N_c) for each c which optimally reproduces the inputted ODs. In detail, the objective function described in Eq. 3.3 aims to minimize the absolute error between the total number of trips produced by the utilised number of schedules and the trips present in the input ODs by controlling the utilisation of each unique candidate activity schedule $N_c \forall \; c \in C$. Hard constraint Eq. 3.4 guarantees that the required trips to form the activity schedule s will not exceed the available trips in the inputted OD matrices. Additionally, constraint Eq. 3.5 assures that N_c does not become negative. The objective function takes its minimum value of zero when the observed and the modelled ODs are identical.

Due to the combinatorial nature of the problem, it is possible that multiple global optima may exist (Redondo et al., 2011; Petit and Trapp, 2019) and consequently more than one combinations of activity schedules can lead to the same optimal objective function value. For this reason, a mechanism to calibrate the optimisation routine towards the identification of a closer to reality solution is required. If a (joint) distribution describing the characteristics of the expected activity schedules (e.g. total travel time, number of activities, modes of transport used, etc.) is available, then this calibrating distribution can be used to shape the output accordingly. To achieve so, each activity schedule within C is assigned with the distribution group i which belongs to. For instance, in the case where the distribution regarding the count of activities within schedules is known (i.e. share of activity schedules including two activities, three activities, etc.), schedules are assigned the appropriate distribution group i based on the number of the included activities. Such high-level information regarding the characteristics of the expected activity schedules can be retrieved from widely available sources such as travel surveys. The final constraint (Eq. 3.6) guarantees that the resulting combination of activity schedules will follow the calibrating distribution. This constraint can be relaxed by the introduction of the term δ_i which allows for tolerance between the observed and the modelled distribution shares. The optimisation problem is mathematically formulated as:

$$min\, Z = \sum_{k \in K} \left(\sum_{p_k \in P_k} \left(\left| \sum_{c \in C} \left(N_c D_c^{p_k} \right) - T_{p_k} \right| \right) \right) \qquad \text{(Eq. 3.3)}$$

subject to:

$$\sum_{c \in C} \left(N_c D_c^{p_k} \right) - T_{p_k} \leq 0 \;\; \forall\, t \in T, p_k \in P_k \qquad \text{(Eq 3.4)}$$

$$N_c \geq 0 \;\; \forall\, s \in S \qquad \text{(Eq. 3.5)}$$

$$\left| b_c^i - \frac{N_c G_c^i}{\sum_{c \in C} N_c} \right| \leq \delta_i \;\; \forall\, c \in C, i \in I \qquad \text{(Eq. 3.6)}$$

3.5.2 Metaheuristics Formulation

Table 3.4 Nomenclature for metaheuristic-based optimisation methods

Variable	Description
$\boldsymbol{M^O}$	Observed OD matrix
$\boldsymbol{ij}$	Pair of locations i and j
$\boldsymbol{m_{ij}^O}$	Number of trips between i, j in M^O
$\boldsymbol{L}$	Connectivity matrix of M^O
$\boldsymbol{ij}$	Pair of locations i and j
$\boldsymbol{l_{ij}}$	Binary variable indicating the presence of a connection (i.e. trip) between i and j in M^O
$\boldsymbol{C}$	All possible tours in M^O
$\boldsymbol{c}$	A tour ($c = \{l_{ij}, \dots, l_{ji}\}$) ($c \in C$)
$\boldsymbol{S}$	All possible combinations of tours (Domain of discourse)
$\boldsymbol{s}$	One combination of tours (i.e. solution) ($s \in S$)
$\boldsymbol{M^s}$	The modelled OD matrix resulting from the aggregation of trips in s
$\boldsymbol{s^*}$	An optimum solution resulting in $M^s \equiv M^O$
$\boldsymbol{D}$	The calibration distribution
$\boldsymbol{H}$	The modelled distribution (i.e. distribution of solution s_b)
$\boldsymbol{d}$	A distribution group ($d \in D, H$)
$\boldsymbol{D_d}$	The share of d in D
$\boldsymbol{H_d}$	The share of d in H
$\boldsymbol{\delta}$	Accepted tolerance between the shares of $D_d$and H_d

3.5.2.1 Formulation

Consider a square OD matrix M^0 containing trips (m_{ij}^0) between the available pairs of locations ij. Additionally, consider the binary connectivity matrix L of M^0 which includes information regarding the presence or not of a trip between the available pairs of locations ij ($l_{ij} \in [0,1]$). The connectivity matrix L allows the synthesis of trip sequences $c = \{l_{ij}, \dots, l_{ji}\}$ with the same origin and destination which are also known as tours. The application of the methodology requires firstly the identification of all the possible tours ($c \in C$) and subsequently the identification of tours' combinations ($s = \{c_1, \dots, c_n\}$) which reproduce M^0. The objective of the presented optimisation problem is the identification of a solution s^* which minimises the difference between the number of trips present in M^0 and the respective number in M^s which is obtained from the aggregation of trips in s. Objective function (Eq. 3.7) must be minimised in accordance to an extensive array of k constraints ensuring the formation of tours based on the availability of trips as described in M^0. This is achieved by the iterative investigation of all tours in s for the presence of $l_{ij} \forall\ ij$ in their

definition and the verification that the total number of occurrences of $l_{ij} \forall\ ij$ does not exceed the corresponding value m_{ij}^0 (Eq. 3.8). It should be also noted that the number k is equal to the number of non-empty cells in M^0, therefore large-scale ODs can result in a substantial number of constraints.

The combinatorial nature of the problem allows for multiple combinations of tours with the same, optimal objective value (Redondo et al., 2011; Petit and Trapp, 2019). The identification of solutions with realistic characteristics can be achieved with the insertion of multiple inequality constraints based on a (joint) distribution, referred as the calibration distribution and denoted by D. The calibration distribution describes the expected characteristics of the tours in the optimal solution and can refer to various dimensions such the tours' length (e.g. frequency of 2-leg, 3-leg, …, n-leg tours) or the sequence of transport modes used for their completion (e.g. car-car, car-bus-car, etc.). To enable so, a pre-processing step completes the classification of all the possible tours (C) to their respective distribution group $d \in D$. If the accepted level of diversion between the calibration distribution and the distribution of s is denoted as δ then the set of constraints in (3) ensure the adherence of the solution to the calibration distribution.

$$\min_{s \in S} \sum_{ij \in M^0} \left(m_{ij}^0 - \sum_{c \in s} \sum_{l_{uv} \in c} l_{uv}[ij = uv] \right) \qquad \text{(Eq. 3.7)}$$

subject to:

$$\sum_{c \in C} \sum_{l_{uv} \in c} l_{uv}[ij = uv] \leq m_{ij}\ \forall\ ij \in M^0 \qquad \text{(Eq. 3.8)}$$

$$|D_d - H_d| \ \leq \delta_{\mathrm{d}}\ \forall\ \mathrm{d} \in \mathrm{D} \qquad \text{(Eq. 3.9)}$$

The previous section showcased two alternatives methods for the addressing of the studied combinatorial problem. Despite the straightforward and concise formulations, the combinatorial nature of the problem can raise the combinatorial explosion issue, preventing the application of the methodology on large-scale instances of the problem. The next Chapter presents a suitable methodology for the confinement of the problem's domain and the achievement of the scalability of the methodology.

Chapter 4
Scalability

Chapter 4 presents the suggested approach to reduce the complexity of the problem and render it applicable for real-world cases. In particular, it presents two additional to the main methodology modules (simplification modules) which ensure the scalability of the methodology without jeopardising the solution's quality.

4.1 Introduction

4.1.1 Combinatorial Explosion

The previous methodological sections presented the required steps to convert a set of multi-period, purpose segmented ODs to the corresponding set of activity schedules which results in travel demand patterns equivalent to the ones observed in the inputted ODs. Despite the strong theoretical foundations, the methodology can face scalability issues due to the combinatorial nature of the formulation.

The synthesis of complete structures from the combination of various building blocks constitutes a typical combinatorial problem. Problems of such nature typically suffer from what is referred as the *combinatorial explosion* issue (Schuster, 2000). Combinatorial explosion occurs due to the exponential increase of combinations resulting from the number of the available building blocks and/or the number of the ways they can be rearranged. The combinatorial explosion phenomenon is encountered in various fields and very often manifests in problems of graph-theoretical context (Michele Conforti, Gérard Cornuéjols, 2014). An example drawn from material science is presented by Treacy et al. (2004). In that study, the researchers developed a methodology for the enumeration of all possible 4-connected graphs within each space group type given the number of unique tetrahedral vertices. Not surprisingly, the authors state the combinatorial explosion issue and the limitations it posed to their study. Similarly, Edwards and Glass (2000) modelled a gene network with the aim to enumerate the distinct logical structures which exist in n-dimensional gene networks. Due to the combinatorial explosion, they were only able to study 4-dimensional networks but still managed to identify patterns of periodic behaviour.

Based on the above, it becomes apparent that the enumeration of all the possibilities manifesting in combinatorial optimisation problems may prove particularly troublesome when the dimensions of the problem exceed some (limited) boundaries. In the context of the presented Thesis, the combinatorial explosion manifests in the number of possible tours/activity schedules deriving from ODs which can grow exponentially with the increase of the latter's size. However, the appropriate simplification strategies (referred as the *simplification modules*) can considerably reduce complexity and render the problem solvable within reasonable time.

4.1.2 Effect of the OD's Resolution

The combinatorial explosion issue becomes more intense with the increase of the available combinations under which individual pieces can be rearranged to form a solution. In the context of the here presented Thesis, complexity is mainly affected by the resolution of the utilised ODs. In order to quantify the resolution of the ODs, the concept of network density (g) is used as a proxy. In graph theory, network density, also known as gamma index (Rodrigue et al., 2017), is defined as the fraction between the actual connections in the network and the possible ones. The arithmetic value of network density typically ranges from 0 to 1 but it can exceed unity for multigraphs. The formula to calculate network density for directed graphs is presented in Eq. 4.1 where e is the number of edges and v the number of vertices present in the network.

$$g = \frac{e}{v\,(\mathrm{v}-1)} \qquad \text{(Eq. 4.1)}$$

A dense network allows the connection between multiple pairs of nodes, leading potentially to a significant increase in the number of plausible tours. As a result, this has a negative effect on the performance of the tours' identification process. To illustrate the effect of spatial resolution to the methodology, two simplified networks, characterised by different resolutions (densities) along with their attributes are depicted in Figure 4.1 and Table 4.1.

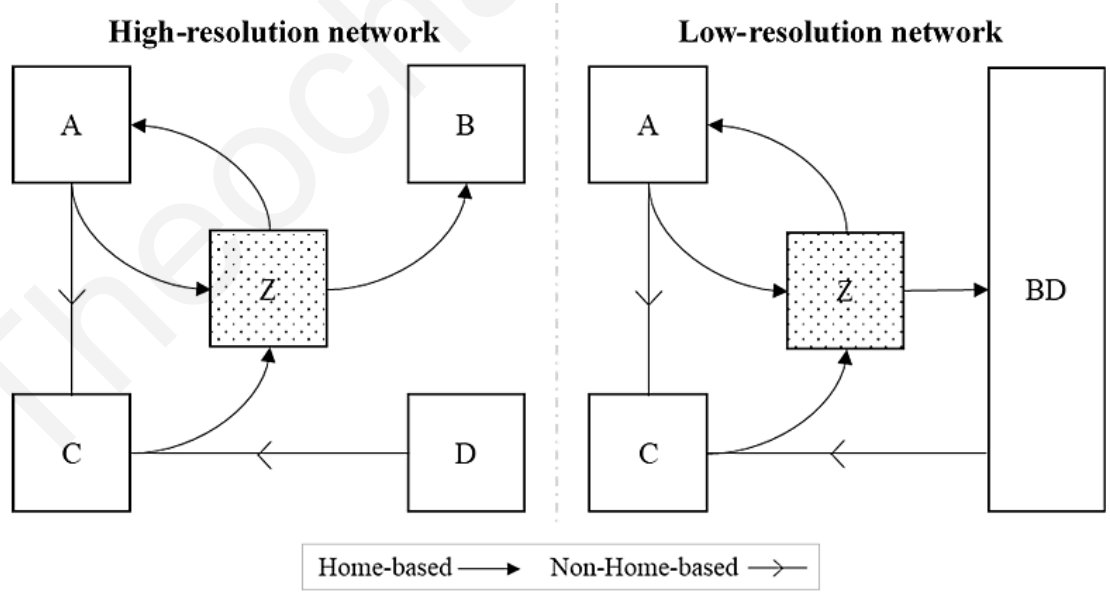

Figure 4.1 Representation of the same OD matrix using a high-resolution (left) and low-resolution (right) network.

For this example, a low-resolution network is created by aggregating the zones of a high-resolution one. Although, the number of captured trips between these two cases remains the

same, the effect of the spatial tessellation in the low-resolution scenario is significant. The network densities for the high- and low-resolution networks are 30% and 50% respectively. The most notable effect is that the reduction of spatial resolution (i.e. increase in density) enables the formation of tours which were impossible in the high-resolution case. For instance, in Table 4.1 it can be observed that the number of tours originating from zone Z increases from two to three. For large-scale networks, the implications of using a low-resolution zoning system can be even more significant, leading to a many-fold increase of the number of plausible tours. The effect of network density will be evaluated at a greater extent at the case study section (Section 6.4).

Table 4.1 Effect of network density on tours' identification process

Network resolution	**Zones**	**Links**	**Density**	**Tours**
High	5	6	30%	[Z, A, Z], [Z, A, C, Z]
Low	4	6	50%	[Z, A, Z], [Z, A, C, Z], [Z, BD, C, Z]

The next section presents the followed approach to enable the simplification and the reduction of the search space concerning the enumeration of tours/activity schedules within an OD derived graph. Reducing the available search space for any combinatorial optimisation problem through the elimination of infeasible solutions, corresponds to a reasonable first step (Hoffman, 2000).

4.2 Simplification Modules

4.2.1 Search Space Reduction

The simplification approach aiming at the reduction of the problem's complexity through the confinement of the available search space is accomplished via two additional simplification modules. The first module (*Graph-filtering module*) attempts the simplification of the graph's structure for the reduction of the possible tours that can be formed within it. The second module (*Candidates-filtering module*) reduces the number of candidate activity schedules by excluding the unrealistic ones. The evaluation of the 'realness' of activity schedules is achieved by comparing their characteristics (e.g. travel time distribution, departure times, combinations of used modes, etc.) against observed travel behaviour patterns (e.g. travel surveys).

The conversion of an OD to a Time Varying Graph (G) allows the identification of the candidate activity schedules set ($C = \{c_1, c_2, \dots, c_n\}$), where vertices c_n correspond to locations of non-empty OD elements $T_{ij} \neq 0$ and as so each tour can be described as a

combination of edges. Once the *candidates set* has been established, an optimisation algorithm can be assigned with the identification of a combination amongst them which reproduces the travel demand as observed in the input ODs. The aim of the currently presented section is to examine the possibility of efficiently reducing the candidates set (C) without excluding candidates required for the synthesis of the optimum solution. For the purposes of the currently presented research, the reduction of the available candidates in C (i.e. search space) takes place in multiple steps, depicted in Figure 4.2 and thoroughly described in the next paragraphs.

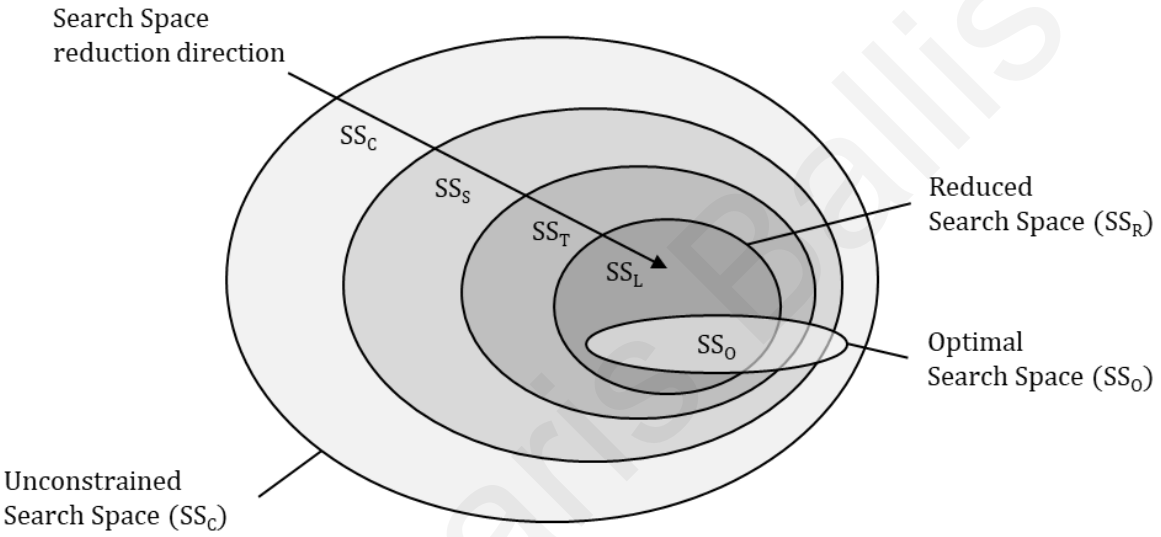

Figure 4.2 Progressive reduction of the initial search space SS_C to the reduced SS_R.

The initially unconstrained search space (SS_C) includes all the activity schedules able to be formed in the OD derived graph (G). An initial step aiming at the reduction of the number of candidates in C, is the selective removal of vertices via a network simplification procedure. This simplification process results in the search space referred to as SS_S. Furthermore, SS_S can be reduced to the SS_T search space by the introduction of cost thresholds related to the impendence of completing each activity schedule. This step imposes an upper bound to the search space by allowing only the identification of activity schedules completed under a predefined cost threshold. The third and final filtering mechanism exploits the likelihood of observing certain travel behaviour patterns in the real world and reduces the previous search space SS_T to the final and considerably more compact SS_L by eliminating unlikely activity schedules. The likelihood of observing an activity schedule is estimated based on available observational data (e.g. travel surveys). Since the above-mentioned simplification steps can be executed independently and in isolation, it is useful to

distinguish SS_L from the search space resulting from the application of all the simplification processes. We refer to the final reduced search space as the r*educed search space* (SS_R).

The outlined simplification methodology can substantially reduce the available search space. Apart from the decrease in the processing time requirements, the reduction of the number of candidates can also simplify the optimisation process required by the OD reverse-engineer problem. However, the reduction of the search space should be executed with caution since it can result to oversimplification and to the exclusion of great shares of candidates which are included in the optimal search space (SS_O). The term *optimal search space* refers to the search space containing only the absolutely required activity schedules to perfectly recreate the initial travel demand patterns (i.e. inputted ODs). Finally, it should be noted that the currently presented study assumes the completion of the reduction steps in the presented order and the direct correspondence between SS_L and SS_R. The following sections elaborate on the suggested search space simplification mechanisms.

4.2.2 Graph-filtering Module

The first simplification mechanism, referred as the *Graph-filtering Module,* developed for the reduction of the number of candidates (C), entails the simplification of the graph on which the identification algorithm will be applied. This simplification step leads to a confined search space denoted as SS_S. Network simplification is a very active research field and numerous methodologies have been proposed (Zhou et al., 2010; Willenborg, 2019). A common approach stands for the 'pruning' of the least 'important' vertices or edges with the ranking of importance (Oldham et al., 2019) being achieved through various measures of centrality (Gómez et al., 2013; Marsden, 2015). Depending on the centrality measure, the type of the network (e.g. directed/undirected, weighted/unweighted) and its characteristics (size, density, scale, etc.), the ranking between nodes may differ considerably. The selection of the most appropriate simplification method depends on the specifics of each application with some centrality measures proving more suitable than others (Gómez et al., 2013). For the purposes of the presented study, the simplification process aims at retaining the variability of paths within the graph and at the minimisation of travel demand exclusion due to the elimination of vertices. For that reason, centrality measures which consider the directionality of edges are considered more suitable for the purposes of the methodology. The evaluation of the most suitable centrality measure for network simplification in this context included four different centrality measures, namely the:

a) EV: The Eigenvector centrality (Bonacich, 1972) for a vertex v is based on the centrality of its neighbours and is calculated as the v^{th} element of the vector x defined by the equation $Ax = \lambda x$ where A is the adjacency matrix of the graph G with eigenvalue λ. Eigenvector centrality can be calculated for directed and weighted graphs. This centrality measure was selected for evaluation due to its simplicity and the fact that the weights and the directions of edges influence the centrality ranking of each node.

b) PR: PageRank (Page et al., 1998; Langville and Meyer, 2005) is a centrality measure developed by Google to rank the importance of websites (i.e. vertices) in the web. The importance of vertices is measured based on the number of links (i.e. edges) which point to websites while considering the importance of the websites themselves. PageRank centrality can be calculated for directed and weighted graphs constituting it very suitable for the purposes of the suggested methodology.

c) RWB: Random-Walk Betweenness centrality (Newman, 2005) estimates the number of random walks which traverse through each vertex and eliminates the assumption of other centrality measures (e.g. shortest-path betweenness) that information is solely spread among shortest paths. Betweenness centrality is calculated for weighted but not directed graphs. This centrality measure was selected for evaluation due to the expected preservation of variability of paths after the simplification.

d) SC: Subgraph centrality (Estrada and Rodríguez-Velázquez, 2005) of a vertex v is defined as the sum of all length weighted cycles originating at v. Weights decrease with the increase of cycle length and each cycle is associated with a subgraph. Subgraph centrality is calculated for undirected and non-weighted graphs. This centrality measure was selected for evaluation due to the inclusion of closed walks (cycles) in its definition, closely resembling the given problem of tours' enumeration within graphs.

The previous section focused on the simplification of the graph's structure for the reduction of the possible number of plausible tours/activity schedules within a graph. Nonetheless, this reduction can be also bolstered by the exploitation of observational data for the exclusion of rare tours/activity schedules.

4.2.3 Candidates-filtering Module

Travel behaviour theory accepts that people make rational travel behaviour choices, usually constrained by some form of budget (Goodwin, 1981; Nurul Habib and Miller, 2008). In addition, recent studies have showcased high degree of spatial and temporal regularity of travel behaviour (Gonzalez et al., 2008; Schneider et al., 2013), therefore the likelihood of observing certain mobility patterns can be estimated with high certainty. The *Candidates-filtering module* is capitalising on these two observations for the reduction of the problem's search space through the exclusion of particularly costly or irregular travel patterns.

4.2.3.1 Cost thresholds

Travel behaviour is strongly affected by the cost of travelling, where the notion of 'cost' can refer to all dimensions impeding travel such as monetary cost, travel time, number of interchanges, mode availability, etc. As a consequence, the introduction of cost thresholds (e.g. the total number of legs in each path) during the enumeration of activity schedules within ODs (i.e. Identification module) can considerably reduce the size of the problem. For example, the number of possible paths between a pair of nodes in a fully connected graph is calculated by the factorial $(V-2)!$. However, requesting the identification of paths consisting of up to three legs results in considerably less alternatives ($\prod_{V-3}^{V-2} V$), leading to a significant reduction of the required processing time and the total number of resulting paths. In the context of the tours' identification problem, cost thresholds can be retrieved from relevant travel behaviour analysis sources (e.g. travel surveys). Depending on the available information regarding the network (e.g. link travel times, link travel costs, public transport coverage, etc.), the path finding algorithm can be tailored to eliminate paths based on predefined upper cost limits (i.e. travel time, interchanges, etc.). As an example, activity schedules with excessively long travel times (e.g. exceeding 5 hours of commuting time) can be characterised as unrealistic and get excluded from the search space. The implications regarding the network simplification are significant since the initially unconstrained search space can be now bounded. The effects of imposing cost cut-offs to the candidates tours/activity schedules identification process are further elaborated and exemplified in Section 7.2.1.1.

Despite the high-performance of algorithms suitable for the identification of paths within graphs, the required time to identify all possible paths between two nodes can grow prohibitively long (Sedgewick, 2001). More precisely, although a single path can be found in $O(V + E)$ time, where V, E stand for the number of vertices and edges of the graph

respectively, the total number of paths may require significantly more time ($O(V!)$). The observation that the cost (disutility) of travel has a great impact on the shaping of the daily travel schedule of individuals (Goodwin, 1981; Recker, 2001) can be usefully exploited. For instance, the total travel time of a tour is usually subjected to time or budget constraints, therefore information regarding the users' time-budget can be exploited to discard excessively 'expensive' tours. Moreover, results from travel behaviour analysis have verified that travellers tend to limit the number of trips they execute during a day (Han and Sohn, 2016; Department for Transport, 2017) and only a small percentage of people (around 2.5%) completes more than five trips a day. The application of sensible travel behaviour-based thresholds can reduce the processing time required to identify all tours within a graph without though discarding frequent travel behaviour patterns. Practically, the reduction of the search space is achieved by the introduction of maximum cost thresholds for various dimensions such as the number of legs in tours, the total travel time, the geodesic distance, the monetary budget, etc. The following figure (Figure 4.3) depicts the application of the tour's identification module on the hybrid network which derived from the ODs presented in Table A.3 of the Appendix. The identification algorithm is executed with different thresholds regarding the maximum number of legs in the tours. In the first case (a) tours can reach lengths of up to eight legs (excluding the temporal links), while in the second case (b) the threshold is reduced to three legs. The reduction in the search space is significant since the initial 64 tours are reduced to just four. As it will be verified in a later section (Section 7.2.1.1), the benefits of imposing such constraints on realistic transport networks can prove even more substantial. The programming implementation of the tours identification process with thresholds can be found in the relevant section of the Appendix (Section B.1.1).

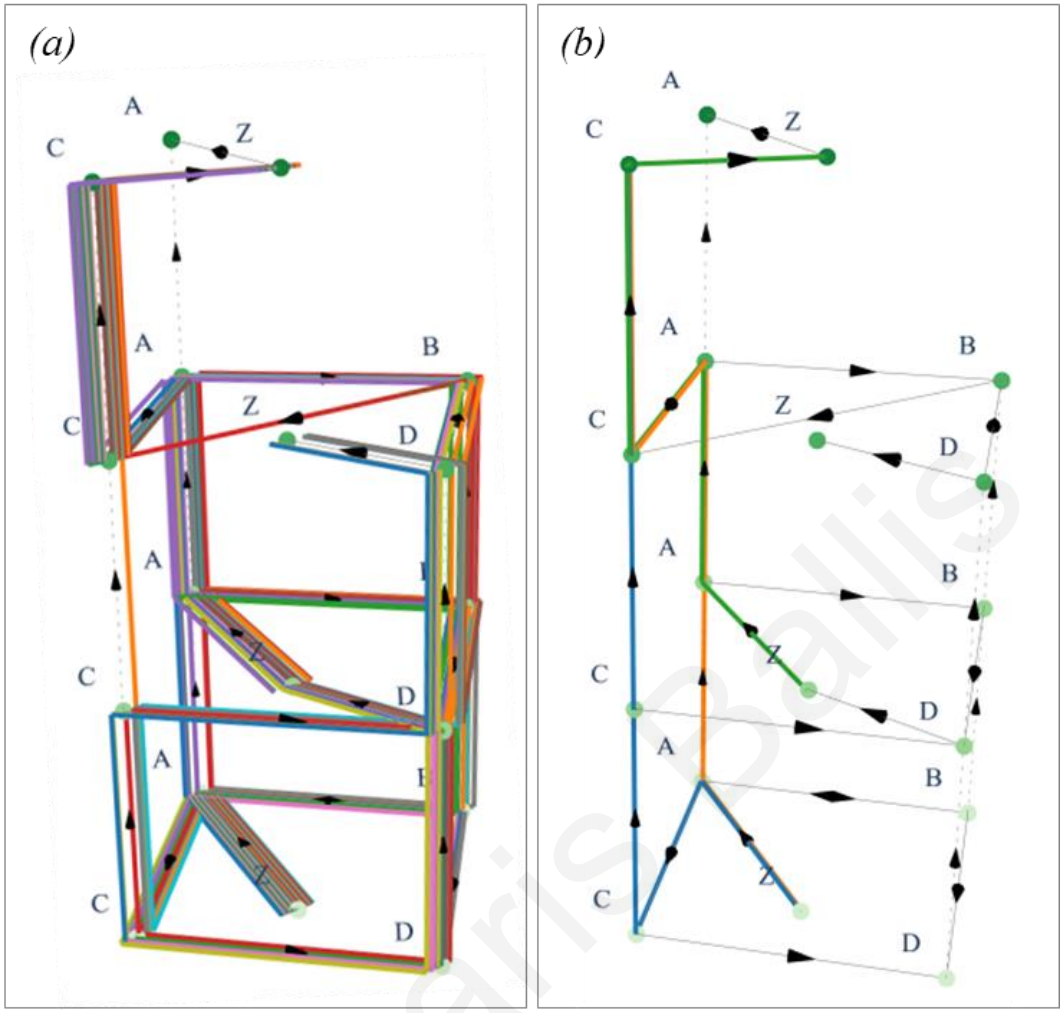

Figure 4.3 Identification of all tours originating from zone Z which a maximum number of allowed legs set to (a) eight and (b) three.

In order to highlight the positive effect of imposing cost thresholds on the required processing time, the computational burden for a hypothetical network of 3,000 vertices and 35,000 edges is demonstrated in Table 4.2. As it can be observed, halving the total travel time threshold from four hours to two, reduces the required processing time by at least 96% regardless of the maximum number of allowed legs in tours. The implications are very important because the required processing time to identify all the possible paths between two nodes increases factorially with the maximum length of the path ($O(V!)$). Nonetheless, imposing thresholds can counter this increase and significantly confine the search space.

Table 4.2 Required processing time for two total travel time thresholds.

Max tour length (legs)	**Processing time (4 hours travel time threshold)**	**Processing time (2 hours travel time threshold)**	**Decrease %**
4	34.0s	1.1s	96
5	67.7s	1.2s	98

4.2.3.2 Likelihood

Apart from the reduction of the available search space through network simplification and the introduction of cost thresholds, the candidate tours set (C) can be further reduced by the exclusion of unlikely tours or groups of those referred as *tour-types*. A tour-type is identified as a group of tours with similar characteristics (e.g. number of legs). Depending on the available dimensions in the input ODs, multiple tour-types can be created. For instance, if the ODs are segmented by transport mode, then various combinations of transport mode sequences can be formed (e.g. [car, car], [car, bus, bus, car], etc.). Based on the principle that some tour-types are significantly more frequent than others, the rare ones can be eliminated from the search space without impacting considerably the quality of the solution. However, the exclusion of even a small fraction of rare tour-types can drastically reduce the available search space due to the disproportionality between the share of unlikely tour-types in the optimum search space (SS_O) and their corresponding share in the unconstrained search space (SS_C).

As it will be demonstrated in the application of the methodology in a large-scale implementation (Chapter 7), the application of the two previously presented simplification modules can render the problem of tours enumeration problem within large scale ODs solvable in reasonable time.

Chapter 5
Large-Scale Optimisation

Chapter 5 provides details regarding the followed approach to enable the large-scale optimisation required for the completion of the proposed methodological framework. In addition, the Chapter presents the Adaptive Sampling Simulated Annealing (ASSA) optimisation algorithm developed for the purposes of the Thesis.

5.1 Nomenclature

Variable	Description
M^O	Observed OD matrix
ij	Pair of locations i and j
m_{ij}^O	Number of trips between i, j in M^O
L	Connectivity matrix of M^O
ij	Pair of locations i and j
l_{ij}	Binary variable indicating the presence of a connection (i.e. trip) between i and j in M^O
C	All possible tours in M^O
c	A tour ($c = \{l_{ij}, \dots, l_{ji}\}$) ($c \in C$)
S	All possible combinations of tours (Domain of discourse)
s	One combination of tours (i.e. solution) ($s \in S$)
M^s	The modelled OD matrix resulting from the aggregation of trips in s
s^*	An optimum solution resulting in $M^s \equiv M^O$
s_b	The modelled solution
s_n	A neighbour to s_b solution
D	The calibration distribution
H	The modelled distribution (i.e. distribution of solution s_b)
d	A distribution group ($d \in D, H$)
D_d	The share of d in D
H_d	The share of d in H
δ	Accepted tolerance between the shares of D_d and H_d
C_d	Tours of C belonging to distribution group d ($d \in D$)
V	Vector of probabilities to draw each tour in C ($v_c \in V$)
R	Tours to be removed from s_b to produce s_n
A	A sample of tours from C based on V

5.2 Introduction

The appearance of performant and efficient computational means has enabled the optimisation of a plethora of complicated and often dynamic real-world tasks such as the finding of shortest closed paths, optimal scheduling, resource allocation, timetabling, genome sequencing, etc. In many cases the optimal addressing of the problem in hand

requires the identification of groups, orders, or assignments of a discrete finite set of objects which comply to some certain conditions or constraints. Combinations of these solution components form the potential solutions of a combinatorial problem. According to complexity theory, it is very hard to design exact algorithms to solve large scale NP-hard combinatorial optimisation problems in moderate computational effort (Tian et al., 1999). NP-hard problems include all the problems where although a solution can be verified in polynomial time, the searching through the solutions is much more complicated. Drawing from the transportation field, many combinatorial optimisation problems (e.g. vehicle routing, fleet management, toll pricing, dynamic traffic assignment, etc.) cannot be optimally solved in polynomial time with 'naïve' brute-force approaches (Vogiatzis and Pardalos, 2013). One of the most prominent examples showcasing the issue is the benchmark Travelling Salesman Problem-TSP (MacGregor and Chu, 2011; Hoffman et al., 2013). The brute force evaluation of all the $(n-1)!$ paths becomes computationally expensive for networks containing more than a dozen of nodes. As a consequence, various optimisation approaches ranging from heuristics to integer linear programming have been suggested for the tackling of the TSP as well as similar combinatorial problems (Yanasse, 2013).

As it has been discussed earlier (Section 4.1.1), the combinatorial optimisation problem of converting multi-period and purpose segmented ODs to activity schedules can quickly grow intractable and from that perspective it can be considered as NP-hard. In particular, the problem expands with the increase of the unique location pairs in the inputted ODs which results in the subsequent and very rapid increase of the number of candidate tours. In the extreme case of a fully connected OD, the number of possible tours can be calculated as $\sum_{p=3}^{n} \binom{n}{p} \frac{(p-1)!}{2}$, where n is the total number of zones (i.e. nodes) in the OD. Although, realistic ODs are not usually fully connected, they nonetheless include a high number of zones (e.g. more than 100) something that can result in a particularly large number of possible tours. Consequently, this large number of possible tours can significantly increase the time required for the identification of the optimal solution. As it becomes apparent, the solution of the problem demands for the deployment of a very efficient optimisation algorithm.

For that reason, an appropriate optimisation methodology able to deal with particularly large search spaces is required. The following sections present some of the so far suggested approaches for large-scale optimisation as well as a novel combinatorial optimisation algorithm. The proposed Adaptive Sampling Simulated Annealing (ASSA) algorithm is able

to provide accurate estimates for remarkably large combinatorial problems by exploiting available information regarding the characteristics of the expected output.

5.3 Large-scale optimisation approaches

Despite the straightforward mathematical formulation of the conversion of ODs to activity schedules problem (Section 3.5), the size of the presented combinatorial problem can increase rapidly due to the combinatorial explosion issue. In more detail, the problem expands with the increase of the unique location pairs in the inputs ODs leading to a subsequent increase of the number of candidate tours. In particular, the number of candidates within OD matrices of realistic size can exceed millions. Despite the previously presented (Chapter 4) approaches for the confinement of the problem's domain, the extensive bounds of the problem may still pose a significant burden on the optimisation module. The next section presents optimisation approaches which can be utilised to overcome this obstacle.

5.3.1 Exact algorithms

Combinatorial problems are very often tackled by integer programming methodologies. In particular, branch-and-bound, and cutting plane methods have dominated the field since their first appearance (Land and Doig, 1960; Johnson et al., 1984; Korte, 2001). These methodologies guarantee the retrieval of an optimum (*exact*) solution and have proven particularly efficient at addressing problems of considerable size in what is usually referred as *polynomial time*. Polynomial time algorithms are those algorithms whose computing time is bounded by a polynomial function of the problem's instance size. However, polynomial time does not necessary resolve to practical time. Despite the arguably impressive capabilities of exact mathematical programming methodologies (e.g. branch-and-bound, branch-and-cut, etc.) and their accompanied implementations (Saltzman, 2002; Makhorin, 2012; Gurobi, 2020; IBM, 2020), their efficiency can be drastically hindered by the size of the problem's domain (Urbanucci, 2018). Nonetheless, exact mathematical programming is not the only available option for the solution of combinatorial problems neither the most effective approach for real-world problems. For these reasons, a variety of alternatives such as metaheuristics has been suggested.

5.3.2 Metaheuristics

Metaheuristics can prove particularly suitable for problems with excessively large universe of discourses as well as for cases where an approximate solution is more desirable global optimum. Notable examples of metaheuristics include Genetic Algorithms, Tabu Search,

Simulated Annealing, Variable Neighbourhood search, (adaptive) Large Neighbourhood search, and Ant Colony optimization (Antosiewicz et al., 2013; Sörensen, 2015). Although the algorithmic mechanics differ significantly between the above-mentioned variants, the majority replies on the gradual improvement of an initial approximation of the optimal solution. The here presented Thesis, focused on the Simulated Annealing (SA) metaheuristic for the addressing of the large-scale combinatorial problem of converting ODs to activity schedules.

The next section presents the standard SA as well as a novel version of SA referred to as Adaptive Sampling Simulated Annealing (ASSA). ASSA improves the standard SA through an adaptive sampling mechanism exploiting high-level calibration information regarding the characteristics of the expected optimal solution. In addition, ASSA constitutes a novel optimisation algorithm in the sense that it suggests a new, generic methodology for the utilisation of calibration information for the increase of the accuracy as well as the performance of the optimisation method.

5.4 Adaptive Sampling Simulated Annealing (ASSA)

5.4.1 Background on Simulated Annealing

Simulated Annealing (SA) is a stochastic approximation type of metaheuristic, able to cope with problems involving large, continuous or discrete, search spaces (Kirkpatrick et al., 1983; Qin et al., 2012). The algorithm draws its analogy from the thermodynamic physical process of annealing where a solid is carefully heated and cooled until the desired (optimum) molecular structure is achieved. Utilizing this analogy to the virtual numerical world, the aim of the SA algorithm is the identification of the s^* solution which minimises the value of objective function ($s^* = \arg\min_{\forall \omega \in \Omega} E(s)$), often termed as 'Energy'. The algorithm avoids getting trapped in local minima by accepting solutions ($s \in S$) worse than the current with a decreasing over time probability. The progressive decrease of this probability is often referred as the 'cooling schedule' and efforts to identify its optimum rate have drawn considerable attention (Nourani and Andresen, 1998; Karagiannis et al., 2017). The decision of accepting the transition from the currently best solution s_b to a new solution s_n is based on the difference of their energies $\Delta = E(s_b) - E(s_n)$ and a decreasing parameter called temperature (T). The transitioning probability is usually calculated following the Metropolis et. al (1953) criterion (Eq. 5.1), although alternatives do exist (Glauber, 1963).

$$P(\Delta) = \begin{cases} 1, & \Delta > 0 \\ e^{-\Delta/T}, & \Delta \leq 0 \end{cases} \qquad (Eq.\,5.1)$$

Under the SA context, the identification of the next solution s_n is expected to be a neighbour of the current s_b, meaning that these two solutions should be similar. Typical methods to produce a neighbour solution include the swapping, replacement, deletion, addition or other similar operations on a relatively low number of elements in s_b however, the 'move' from one solution to the next depends on each application. Only for some extensively studied problems such as the Travelling Salesman Problem (TSP), standard move functions have been established (e.g. 2-opt, 3-opt, etc.). The outline of SA algorithm is presented in Algorithm 5.1.

Algorithm 5.1 Simulated Annealing with random sampling

```
1   T ← T_max # initialize temperature
2   While T ≥ T_min
3     s_n ← DrawSample() # create a random solution
4     Δ ← E(s_b) − E(s_n)
5     If Δ ≤ 0
6       P(Δ) ← 1
7     Else
8       P(Δ) ← e^(−Δ/T)
9     End if
10    r ← random(0, 1) # draw random
11    If r > P(Δ)
12      s_b ← s_n
13    End if
14    T ← T − f(T) # decrease the temperature
15  Repeat
```

Despite their simplicity and wide use, SA-based optimisation methodologies have been characterised of slow convergence (Sadati et al., 2009). Although, the convergence of SA algorithms to a global minimum can be guaranteed with the adoption of a logarithmic cooling schedule $O(\frac{1}{\log(t)})$ (Geman and Geman, 1984; Haario and Saksman, 1991), this rate proves impractical for real applications. Consequently, a considerable part of the relevant literature has concentrated on the identification of more efficient cooling schedules as well

as more efficient sampling methodologies (Fox, 1993; Nourani and Andresen, 1998). One of the first improvements over the standard SA methodology was suggested by Ingber (1996). In that study the suggested Adaptive Simulated Annealing (ASA) method incorporated the process of 're-annealing' to accelerate the annealing schedule and to permit the adaptation to changing sensitivities in the multi-dimensional parameter-space. Other researchers have studied the potential of combining SA with other simulation approaches to improve its performance. For instance a particle swarm algorithm is utilised to improve the generation of candidate solutions (Sadati et al., 2009). Similarly, Wang et al. (2016) employ an Ant Colony Optimisation (ACO) algorithm to guide the generation of candidate solutions towards neighbourhoods with lower energy and present promising performance in terms of convergence speed and solution accuracy. In a similar stream of research, Liang et al. (2014) suggested the combination of an SA algorithm with the Stochastic Approximation Monte Carlo algorithm (SAA) to accelerate the cooling schedule and subsequently showcase its superiority over the standard approach through a set of benchmark optimisation problems. The study of Karagiannis et al. (2017) further improved SAA by simulating a population of interacting Monte Carlo chains, enabling the better exploration of the sampling space. The results obtained from the Parallel and Interacting Stochastic Approximation Annealing (PISAA) method indicated improved performance of PISAA over SAA especially in high dimensional scenarios. The currently presented modification (ASSA) differs by suggesting the acceleration of convergence through the exploitation of calibration information for the efficient sampling of the search space.

5.4.2 The Adaptive Sampling Mechanism

In extension to most of the up to date presented SA methodologies, the current Thesis presents a novel approach for the exploitation of high-level calibrating information regarding the characteristics of the expected optimal solution (s^*) or near optimal solutions. Since optimum solutions are expected to present the characteristics described by the calibration information, sampling from neighbourhoods with such characteristics can improve the efficiency of the algorithm. The calibration information is expected to be expressed in the form of a marginal distribution D which is referred as the 'calibration distribution' since it is utilised to calibrate the characteristics of the identified solutions. An example of such a calibration distribution can be the frequency of tours in the optimum solution s^* by length (e.g. frequency of 2-leg, 3-leg, …n-leg tours). The calibrating distribution H is also used to classify all the available tours in C in distinct distribution groups $D = \{d_1, d_2, \dots d_n\}$.

Finally, it should be noted that the share of each distribution group d is denoted by $H^d \; \forall \; d \in D$.

According to typical SA-class algorithms, solutions $s \in S$ are continuously identified and evaluated until a predefined condition is met (e.g. the drop of the temperature parameter below a certain level). The identification of the next solution s_n is usually expected to be a neighbour of the currently best s_b, meaning that these two solutions should be similar. Typical methods to produce a neighbour solution include the swapping, replacement, deletion, addition or other similar operations on a relatively low number of elements in s_n however, the 'move' from one solution to the next depends on each application. Only for some extensively studied problems such as the Travelling Salesman Problem (TSP), standard move functions have been established (e.g. 2-opt, 3-opt, etc.).

The here presented algorithm (ASSA) adopts an iterative approach for the identification of neighbour solutions. The algorithm begins with an empty solution ($s_0 = \emptyset$) and gradually builds the next solutions through the iterative sampling and appending of tours. The move from solution s_c to a new solution s_n is completed as follows. Firstly, a relatively small set of tours (R) is discarded from s_c. Then a sample A of tours in C is drawn and appended to s_c with the aim to create a neighbour solution. Instead of relying on a random sampling process, the probability of drawing each tour $c \in C$ is adjusted so that the identification of solutions adhering to D is favoured. The iterative adjustment of the probability vector P is required due to trip-availability constraints since not all the sampled tours can be included in the new solution s_n. In particular, each of the sampled tours is added to the new solution only if the availability of trips in M^0 allows so (Eq. 5.2).

$$(\sum_{a \in A} \sum_{l_{ij} \in a} l_{ij}) + 1 \leq m_{ij}^0 \qquad (\text{Eq. } 5.2)$$

It can be proven that the retainment of the distribution of a population deriving from consecutive samples requires the appropriate adjustment of the sampling weights between consecutive draws.

Proposition 1. Retaining the share D_d in solution s^n requires that the probability $p_{c_d}^n$ to draw a tour of type d should be calculated as

$$p_{c_d}^n = \frac{D_d + \frac{|R_d| - |s_d^c|}{\sum_{d \in D} |s_d^n|}}{|\mathrm{C_d}|} \qquad (Eq. 5.3)$$

where s_d^c, s_d^n, A_d and R_d denote the subsets of s^c, s^n, A and R whose tours belong in the distribution group d.

Proof. The preservation of distribution D on solution s^{n+1} requires that the symmetric difference between the excluded tours in R_d, the union of the newly sampled tours in A_d and the existing ones in s_d^n , retains the share D_d $((s^{n+1} \Delta R_d) \cup A_d)$.

$$\frac{|s_d^n| - |R_d| + |A_d|}{\sum_{d \in D}|s_d^{n+1}|} = D_d \Rightarrow \frac{|\mathrm{A_d}|}{\sum_{d \in D}|s_d^{n+1}|} = D_d + \frac{|\mathrm{R_d}| - |s_d^n|}{\sum_{d \in D}|s_d^{n+1}|} \Rightarrow D_d^{n+1} = D_d + \frac{|\mathrm{R_d}| - |s_d^n|}{\sum_{d \in D}|s_d^{n+1}|} \quad \text{(Eq. 5.4)}$$

The probability D_d^{n+1} must be distributed across all the tours in C_d, therefore the vector of probabilities P^{n+1} for each tour in C for iteration $n+1$ is calculated as:

$$p_{c_d}^{n+1} = \frac{D_d^{n+1}}{|\mathrm{C_d}|} \forall\, c \in \mathrm{C_d}, \forall\, d \in D \Rightarrow P^{(n+1)} = \{p_1, p_2, .. p_c\}\ \forall\, \mathrm{c} \in \mathrm{C} \qquad \text{(Eq. 5.4)}$$

The pseudocode of the process is summarised in Algorithm 5.2.

Algorithm 5.2 Adaptive Sampling Simulated Annealing (ASSA)

1	$\boldsymbol{T} \leftarrow \boldsymbol{T_{max}}$ # initialise temperature
2	$\boldsymbol{D^n} \leftarrow \boldsymbol{D}$ # initialise the sampling distribution
3	While $\boldsymbol{T} \geq \boldsymbol{T_{min}}$
4	$\boldsymbol{R} \leftarrow$ DiscardElements$(\boldsymbol{s_b})$
5	$\boldsymbol{P} \leftarrow$UpdateProbabilityVector$(\boldsymbol{R}, \boldsymbol{s_b})$
6	$s_n \leftarrow$ DrawSample$(\boldsymbol{P})$ # create a new solution based on $\boldsymbol{P}$
7	$\boldsymbol{\Delta} \leftarrow \boldsymbol{E}(\boldsymbol{s_c}) - \boldsymbol{E}(\boldsymbol{s_{n+1}})$
8	If $\boldsymbol{\Delta} \leq \boldsymbol{0}$
9	$\boldsymbol{P}(\boldsymbol{\Delta}) \leftarrow \boldsymbol{1}$
10	Else
11	$\boldsymbol{P}(\boldsymbol{\Delta}) \leftarrow \boldsymbol{e}^{-\boldsymbol{\Delta}/\boldsymbol{T}}$
12	End if
13	$\boldsymbol{r} \leftarrow \boldsymbol{random}(\boldsymbol{0}, \boldsymbol{1})$ # draw random
14	If $\boldsymbol{r} > \boldsymbol{P}(\boldsymbol{\Delta})$
15	$\boldsymbol{s_b} \leftarrow \boldsymbol{s_n}$
16	End if
17	$\boldsymbol{T} \leftarrow \boldsymbol{T} - \boldsymbol{f}(\boldsymbol{T})$ # decrease the temperature
18	Repeat

As it will be presented in the following validating scenario, adjusting the vector of probabilities during consecutive iterations as described in (Eq. 5.4) significantly improves the performance of the optimisation process. In addition, ASSA suggests an effective approach for the introduction of constraints to metaheuristic-based methods through the projection of the calibrating distribution D on the output.

The previous section described in detail simplification steps to allow the efficient application of the methodology on ODs of realistic size while the next section proves the potential of the methodology through a proof of concept application.

Chapter 6
Proof of Concept

Chapter 6 presents the proof of concept of the proposed methodology performed over a set of ODs deriving from a large number of observed activity schedules. Furthermore, the Chapter goes into great depth to evaluate the methodology from multiple perspectives as well as to showcase the additional travel behaviour insights which can be drawn when aggregate ODs are converted to individual activity schedules.

6.1 Model Execution and Experimental Setup

The following section presents the required details for the evaluation of the proposed methodological framework and its constituents, for proof of concept purposes. The section begins with the description of the input, continues with the configuration of the models' run, and concludes with the presentation of the results.

6.1.1 Input Dataset

The currently presented methodological framework is evaluated based on set of multi-period and purpose segmented ODs deriving from the aggregation of synthetic activity schedules which are referred to as the *observed activity schedules*. The observed activity schedules form the ground-truth based on which the while evaluation process was executed upon. The decision to synthesise the required ODs by aggregating activity schedules rather than utilising a pre-existing set of ODs aims at the enablement of the meticulous, one-to-one comparison between the observed and the resulting (*modelled*) activity schedules. In the opposite case where, pre-existing ODs had been used, the true potential of the methodology could have been underestimated due to inconsistencies of the input rather than inefficiencies of the methodology itself.

6.1.1.1 The Zoning System

The synthesis of the observed activity schedules entails the definition of a zoning system which will be used to express the sequence of zones visited by each of the schedules. As it has already been pointed out (Section 4.1.2), the utilised zoning system can significantly affect the complexity of the problem. To simplify the process, the required zoning system was developed based on UK standard census geographic boundaries. In particular, the locations of the activities taking place within the observed activity schedules were expressed in the standard 'Lower Layer Super Output Areas' (LSOAs) zoning system. As of 2011 UK and Wales are divided in 34,753 LSOAs with a minimum population of 1,000 and an average of 1,500. For the purposes of the current analysis, a zoning system consisting of 470 LSOAs covering the area of Bristol, UK was employed (Figure 6.1).

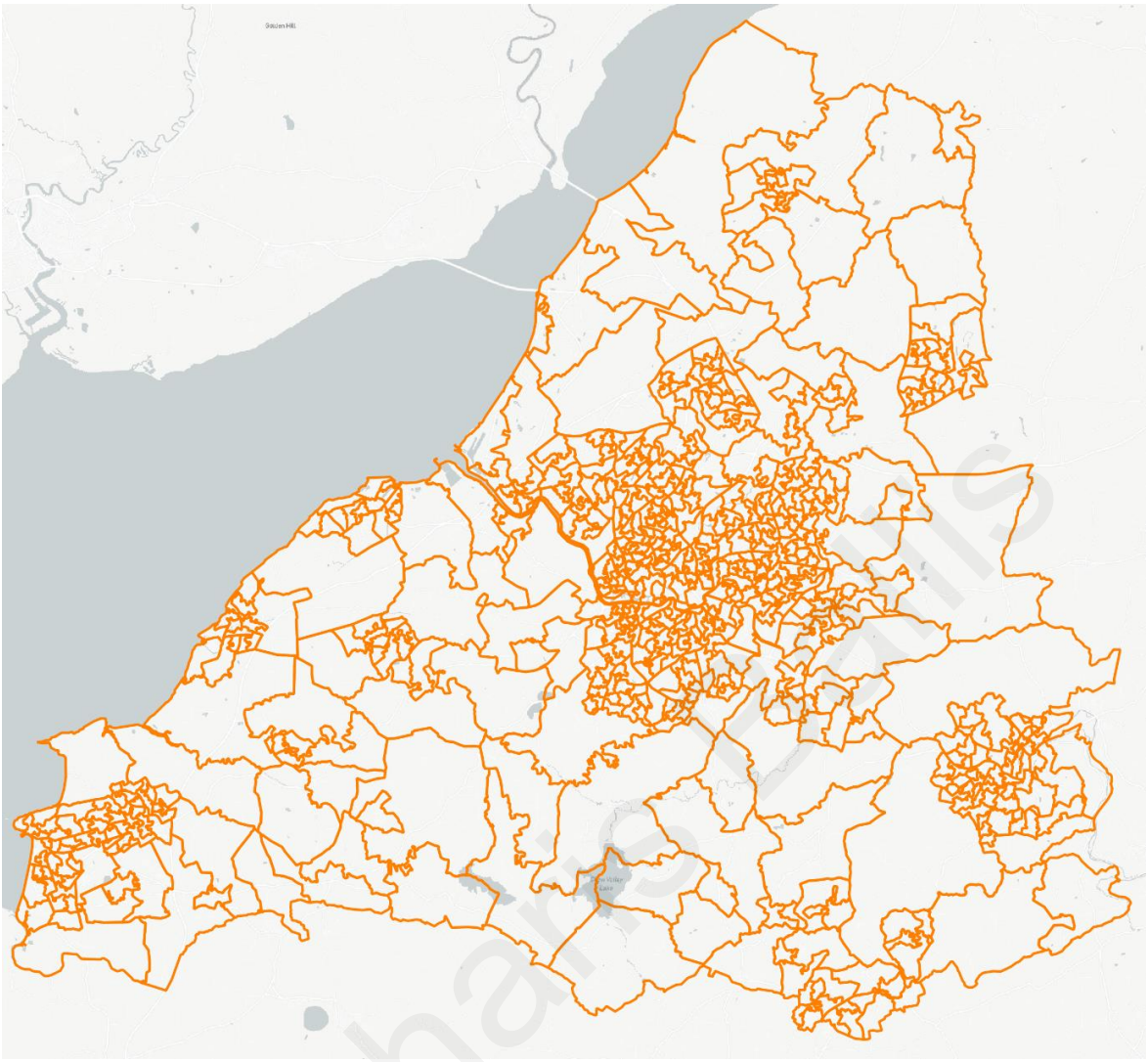

Figure 6.1 The modelled area of Bristol, UK and the corresponding LSOA-based zoning system consisting of 470 zones.

6.1.1.2 Observed Activity Schedules

The reflection of realistic travel behaviour patterns on the observed activity schedules was ensured by the synthesis of the input from information available in the National Travel Survey (NTS) of UK concerning the wider area of Bristol (Department for Transport, 2017). In particular, information regarding the location, the duration and the type of activities included in the activity schedules of the surveyed participants were used for the synthesis of 25,000 unique schedules. The synthesised activity schedules vary in all the available dimensions except from the total number of visited locations which were constrained up to a maximum of five. According to NTS, activity schedules visiting more than five locations are rare (0.5%), therefore such complex schedules were excluded from the analysis.

The result of this synthesis is an extensive list of individual schedules described by three tuples containing the locations, the departure times, and the activities executed at each of the visited location. The locations of the executed activities within a schedule were expressed as a sequence of LSOAs where the origin and the destination of the activity schedule correspond to the same zone. The exact departure time from each activity within a schedule were assigned and aggregated as presented in Table 6.1. Finally, the activity taking place prior to each departure was classified either as Home, Work or Other. Finally, examples of indicative activity schedules are presented in Table 6.2.

Table 6.1 Definition of available time periods for trips' departures.

Time period	Covered period
OP1	00:00 – 07:00
AM	07:00 – 10:00
IP1	10:00 – 13:00
IP2	13:00 – 16:00
PM	16:00 – 19:00
OP2	19:00 – 22:00
OP3	22:00 – 23:59

Table 6.2 Sample from the observed activity schedules

Activity schedule	Locations	Departure time periods	Departure from activity
1	(E01014530, E01033079)	(AM, PM)	(Home, Work)
2	(E01014797, E01014621, E01014403)	(IP1, PM, OP2)	(Home, Work, Other)

The 'to Home' activity is omitted for brevity

6.1.1.3 The Calibration Distribution

The marginal distribution relating the total travel time of the observed activity schedules and the sequence of their departure time periods is presented in Figure 6.2. The distribution includes six travel time bins of 900 second durations (15 minutes) and combinations of seven time periods covering a whole day. The joint marginal distribution includes 386 unique distribution-groups. As an example, a "*AM;PM & (900, 1800]*" distribution-group includes all tours with their beginning leg taking place during the AM period and their second during the PM, while their total travel time ranges between 900 and 1800 seconds. This distribution is referred as the '*calibration distribution*' since it was also used for calibration purposes at later stages of the application.

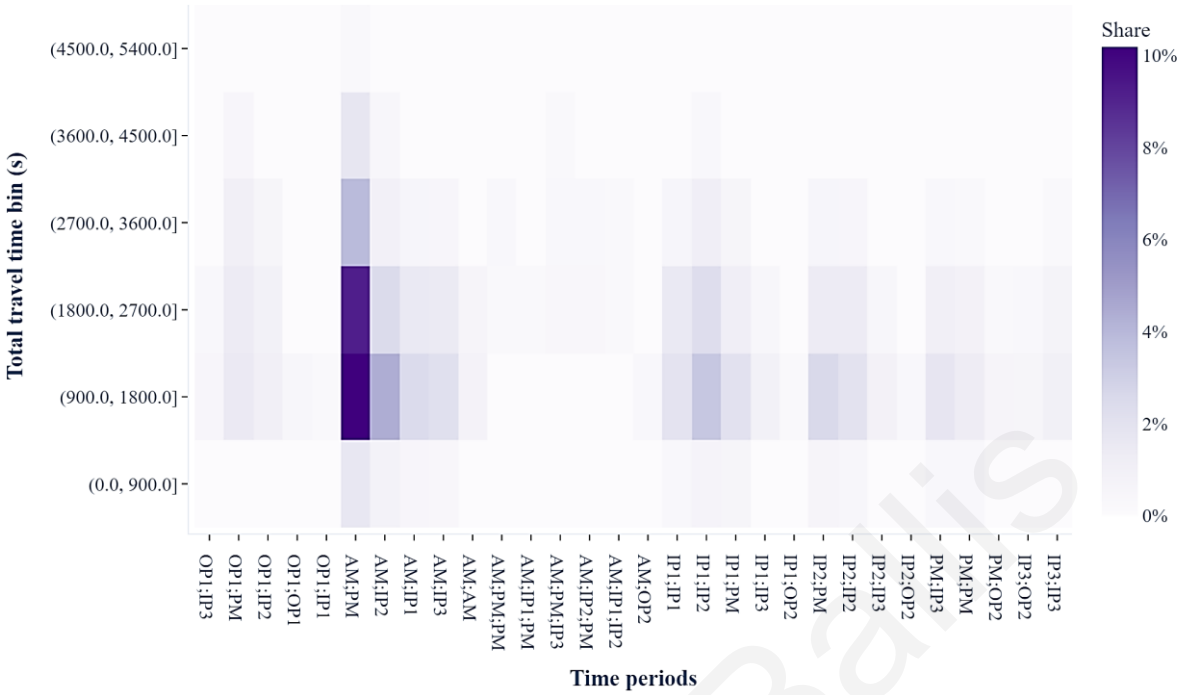

Figure 6.2 The distribution of the observed tours in terms of total travel time and time periods of departure (150 largest out of 386 groups).

6.1.1.4 Observed ODs

The previously described observed activity schedules are completed via a series of interdependent trips. Since the origin, the destination, the time-period of departure and the activity to be executed at the destination are known, the schedules can be converted to time-period, purpose segmented OD matrices through the aggregation of the required trips to complete the schedules. The 28 resulting ODs cover an area of 470 LSOAs and segment 53,104 trips across four trip purposes and seven time periods (Table 6.3). The application of the Graph generation module, converted the observed ODs into the presented hTVG (Figure 6.3) consisting of 3,245 nodes and 42,171 links. For reasons of clarity the 2,818 temporal connectors, enabling the traversal between time periods are not depicted. As it can be noticed, the number of edges between the different layers of the multilayer network varies significantly. This is expected since travel demand in urban areas is not usually uniformly distributed across the day. On the other hand, the number of visited nodes remains generally stable, indicating that the spatial dimension of travel, at least for this scenario, does not vary significantly during the day.

Table 6.3 Summary of observed ODs (proof of concept scenario).

Trip-purpose	OP1	AM	IP1	IP2	PM	OP2	OP3	Total
HBW	631	2,690	1,521	1,798	2,656	1,064	256	10,616
HBO	2,329	9,895	5,367	6,665	9,703	3,760	963	38,682
NHBW	7	149	265	282	284	59	4	1,050
NHBO	6	359	657	830	774	122	8	2,756
Total	2,973	13,093	7,810	9,575	13,417	5,005	1,231	53,104

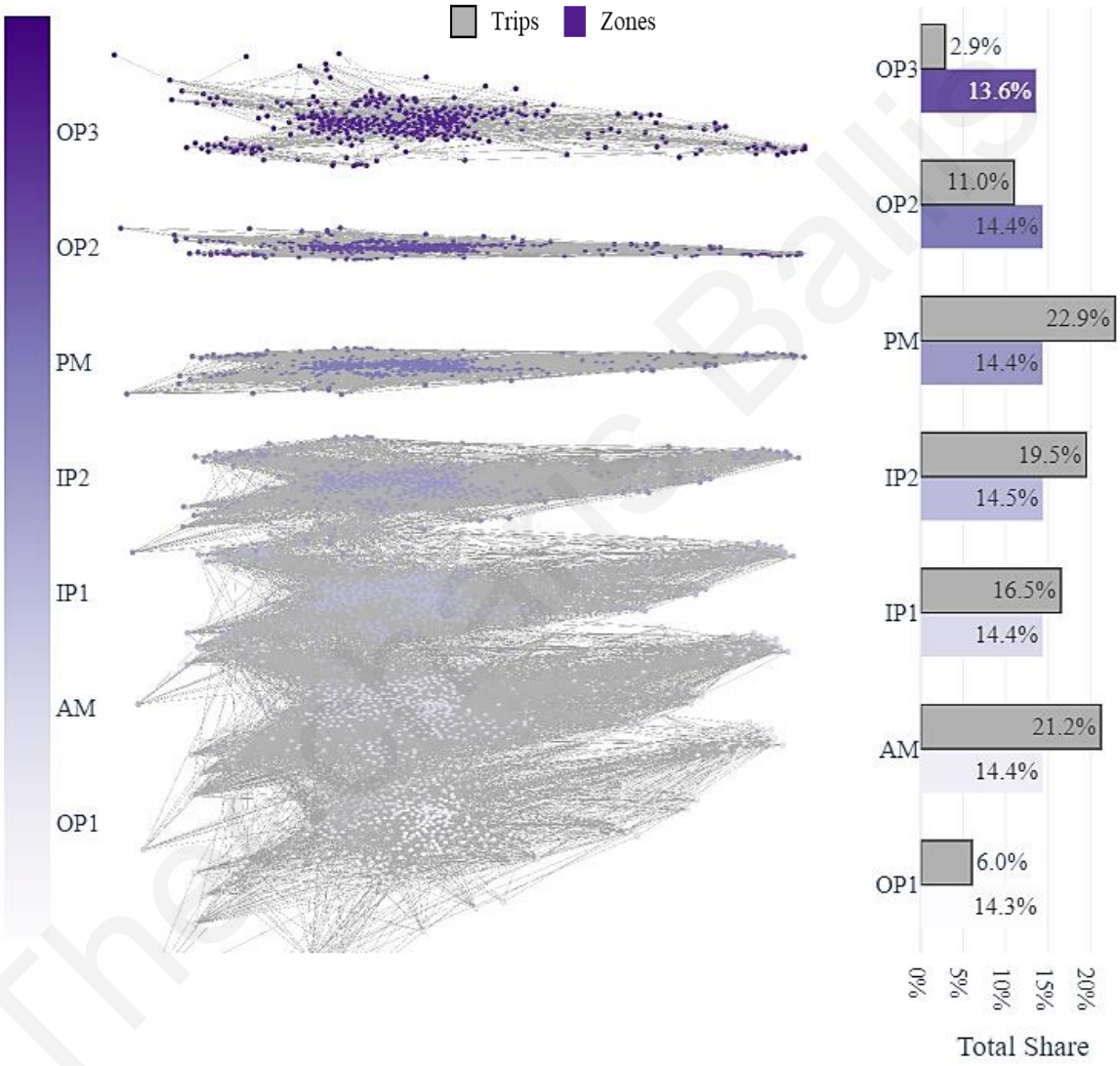

Figure 6.3 The hybrid Time Varying Graph (hTVG) resulting from the aggregation of the observed tours into OD matrices. The right-hand side presents the distribution of nodes and edges across the available time periods.

The presentation of the observed travel demand patterns is also depicted in Figure 6.4 where the outline of the studied area (Bristol, UK) as well as the number of originating trips from each zone is presented. As it can be noted, the observed demand is spread throughout the

urban space (and the corresponding network) but some zones, particularly within the centre of Bristol, tend to produce more trips.

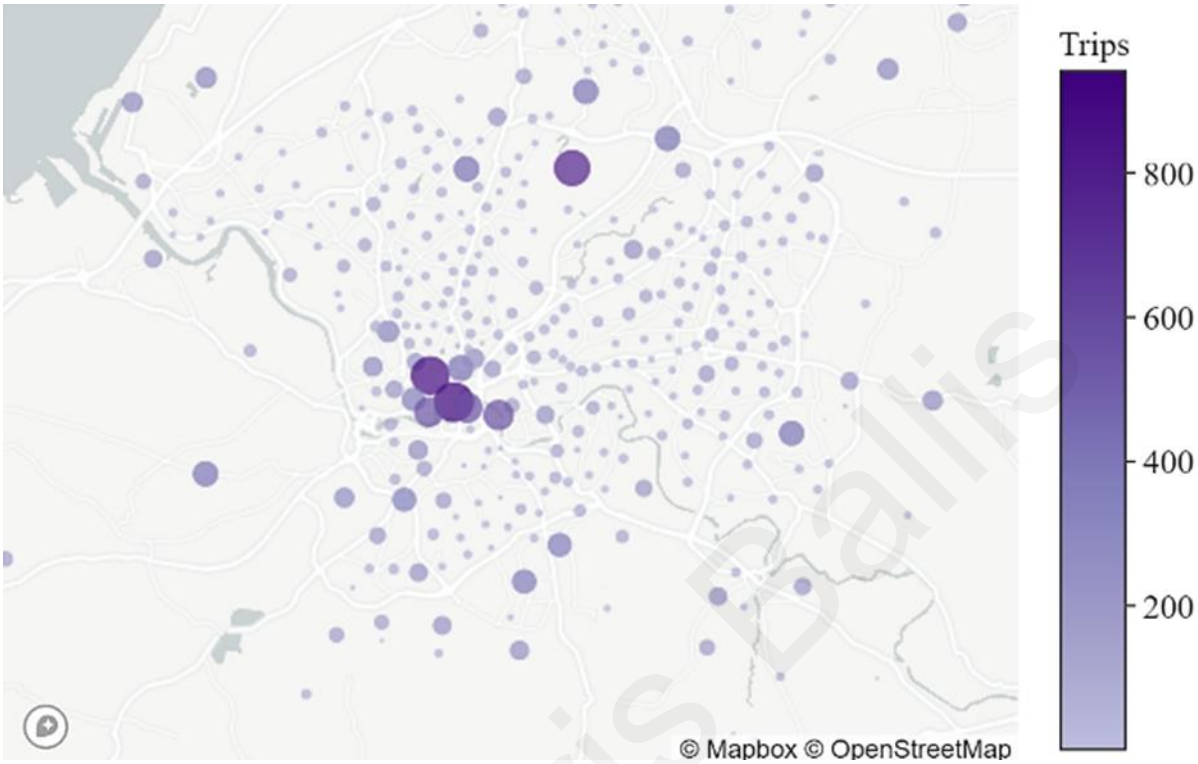

Figure 6.4 Number of originating trips by zone for the proof of concept scenario.

6.1.2 Configuration

The previously presented ODs were inputted to the methodology with the aim to produce the travel demand equivalent set of activity schedules whose characteristics adhere to the calibration distribution. The following section provides the details relevant to the technical configuration required for the execution of the methodology.

The size and the complexity of the proof of concept scenario did not demand for significant reduction of the available search space except for the restriction of identifying all the possible activity schedules with a total travel time less than 5,400 seconds. This threshold derived from the maximum duration of travelling time recorded in the observed activity schedules. No other simplification measure was applied to the identification module.

The combinatorial optimisation process was expressed using the optimisation modelling framework Pyomo (Hart et al., 2017) and solved by the CPLEX optimiser (IBM, 2020). The relevant code is presented in the Appendix (Appendix B.1.5) The maximum processing time devoted to the optimisation module was set at five hours. Finally, the tolerance of the

distribution group shares between the observed and the modelled activity schedules was set at ± 1%.

6.1.3 Results

The application of the suggested methodology on the multi-period and purpose dependent observed ODs resulted in 460,831 candidate activity schedules, out of which 24,818 were used in the final solution. As it will be showcased in the following section, the modelled activity schedules represent the observed patterns vey accurately. In terms of performance, the whole process was executed in approximately 8 hours (27,482 sec) on an Intel® Xeon CPU powered computer with 32GB of available RAM. The identification module accrued roughly for 45% (≈3.5 hours) of the total processing time while the rest 55% (≈4.5 hours) was devoted to the optimisation module. The Graph-generation and the Activity-scheduling modules have an almost insignificant effect on the performance of the methodology since they require less than 1% of the total processing time. It should not be disregarded that problems of combinatorial nature like the one presented above can often prove particularly cumbersome even with state-of-the art methodologies and high-end industrial computing resources (Klotz and Newman, 2013). Therefore, the previously mentioned solving times can be considered satisfactory. Since the process is highly parallelisable, additional processing time reductions can occur if computational systems with more cores are to be utilised.

A visual representation of the spatial density of the utilised activity schedules in presented in Figure 6.5. In particular, the figure presents the cumulative number of activity schedules traversing through each zone. As it can be observed, some zones (mainly around the centre of Bristol) attract significantly more visitors than the rest. Nonetheless, it can be observed that the whole area is traversed by a considerable number of tours.

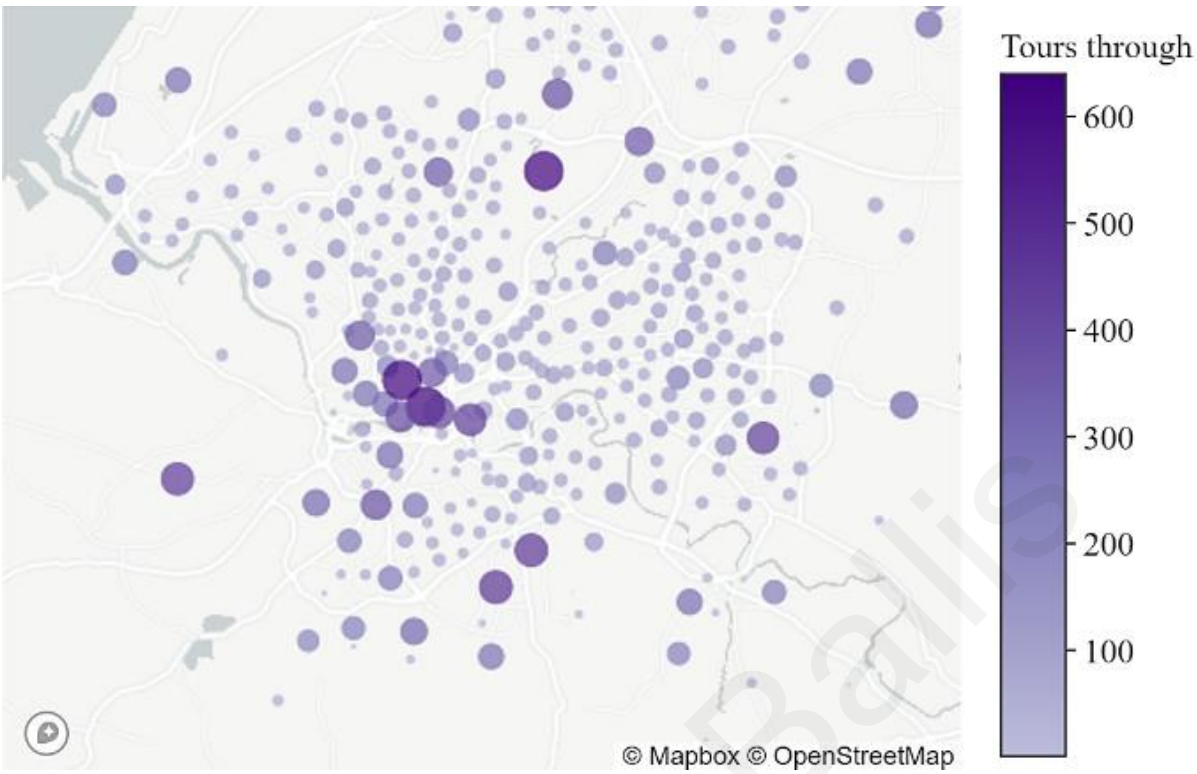

Figure 6.5 Number of tours crossing through each zone for the validation scenario.

6.2 Evaluation

This section aims at the meticulous validation of the methodology both at the aggregate- as well as at the disaggregate-level. The aggregate-level assesses the ability of the methodology to identify a combination of activity schedules able to represent travel demand as captured in the inputted ODs while adhering to the available calibration information. The disaggregate-level of validation focuses on the assessment of the outputs' realism and entails the one-to-one comparison between the observed activity schedules and the modelled ones.

6.2.1 Aggregate-level

6.2.1.1 Comparison of ODs

The first level of the methodology's assessment includes the comparison between the observed (input) ODs against the modelled. In brief, the 24,818 modelled activity schedules utilise more than 99.2% of the total trips from the observed ODs. To validate this, the comparison between the observed and the modelled ODs by time period of departure and trip-purpose is presented in Table 6.4. As it can be noticed, the differences between the compared ODs are minimal. Some few exceptions presenting large percentage differences are observed only for ODs with very low demand (e.g. NHBO-OP2).

Table 6.4 Absolute and percentage difference between the observed and modelled ODs. Values in parentheses represent the percentage difference.

Purpose	OP1	AM	IP1	IP2	IP3	PM	OP2	Total
HBW	16 (0.7)	55 (0.6)	32 (0.6)	33 (0.5)	43 (0.4)	21 (0.6)	7 (0.7)	207 (0.5)
HBO	2 (0.3)	13 (0.5)	12 (0.8)	12 (0.7)	11 (0.4)	3 (0.3)	2 (0.8)	55 (0.5)
NHBW	0 (0.0)	5 (1.4)	14 (2.1)	15 (1.8)	18 (2.3)	1 (0.8)	0 (0.0)	53 (1.9)
NHBO	1 (14.3)	2 (1.3)	9 (3.4)	9 (3.2)	7 (2.5)	2 (3.4)	1 (25.0)	31 (3.0)
Total	19 (0.6)	75 (0.6)	67 (0.9)	69 (0.7)	79 (0.6)	27 (0.5)	10 (0.8)	346 (0.7)

The accuracy of the methodology is also examined in the scatter diagram of Figure 6.6. The size of each point represents the number of trips between the available pairs of locations in the ODs. As it can be noticed, the number of missing trips (orange points) is significantly lower compared to the number of the observed trips. This can be further verified by the minor error terms visualised in the accompanying histograms presenting (in logarithmic scale) the total origins and destinations from and to zones.

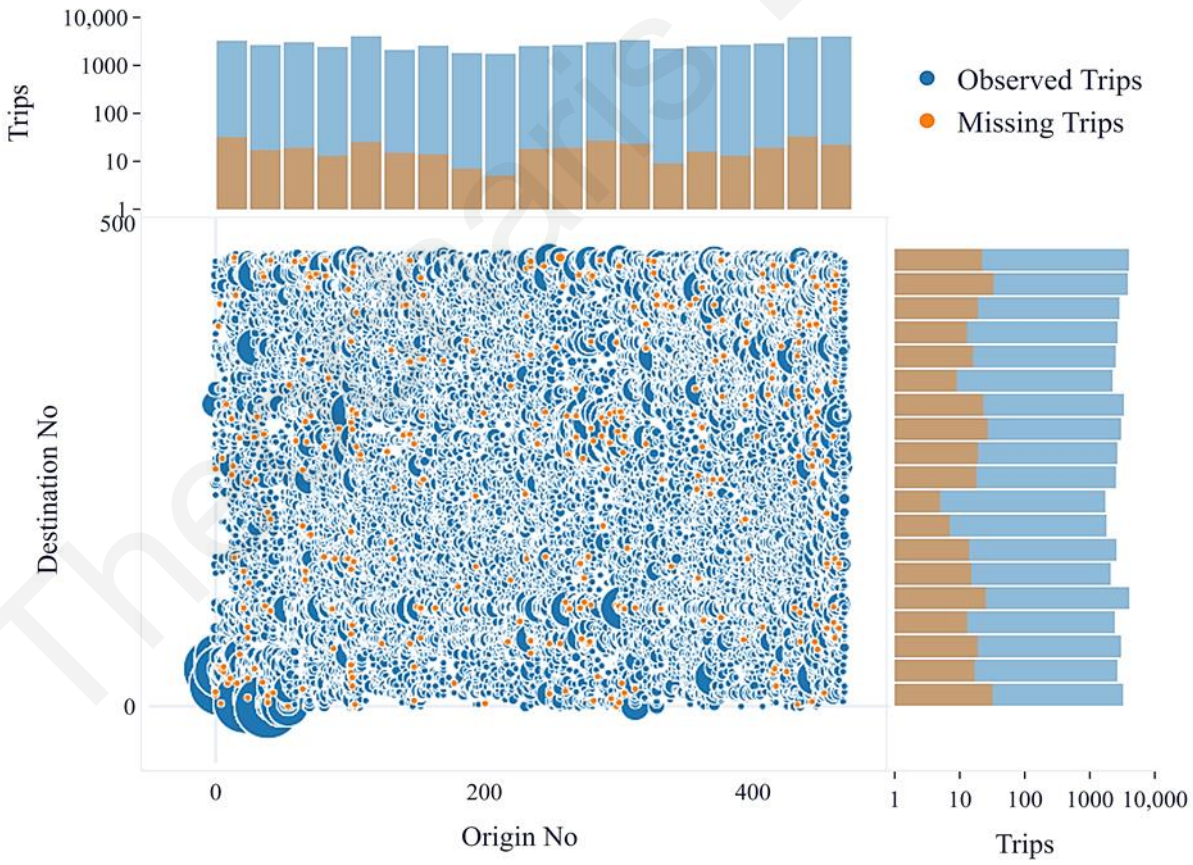

Figure 6.6 Comparison between the number of person trips in the observed and the modelled ODs.

The accuracy of the methodology has been further assessed through a regression analysis comparing the values of the respective cells (29,476 in total) between the observed and

modelled ODs. As it can be noticed in Figure 6.7, the discrepancies between the OD cells of the target and the modelled ODs are minimal, the R^2 value significantly high and the slope of the regression line very close to 1.

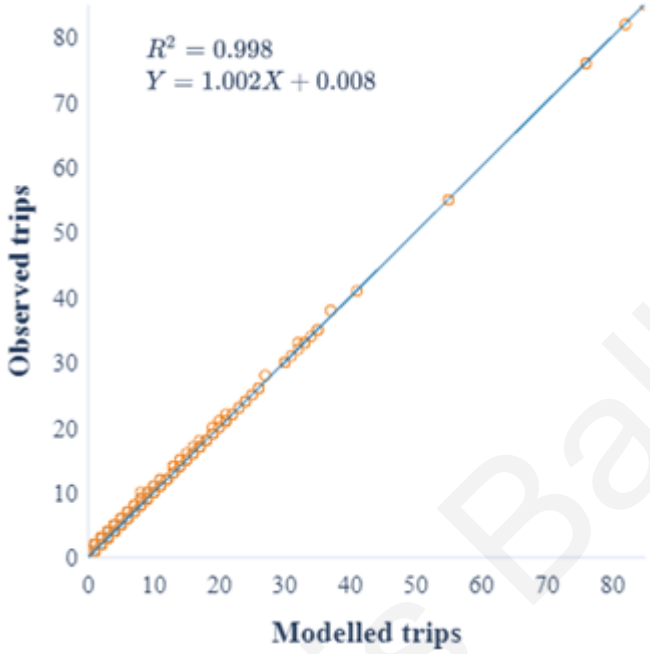

Figure 6.7 Comparison between the cells of the Observed and the Modelled ODs

Finally, the conversion of a multi-period ODs to individual activity schedules is visualised in the two following figures (Figure 6.8 and Figure 6.9). In particular, Figure 6.8 depicts the observed ODs where colour shading and vertical positioning are used to differentiate trips departing at different time periods. Darker tones and higher elevated trips indicate departures later in the day. On the contrary, Figure 6.9 presents the completion of the identified modelled activity schedules with each schedule being represented by a different colour. As it can be observed, the majority of trips has been utilised to form activity schedules. More importantly, the initially unrelated trips are linked in continuous sequences able to present mobility in a considerably more contextual manner.

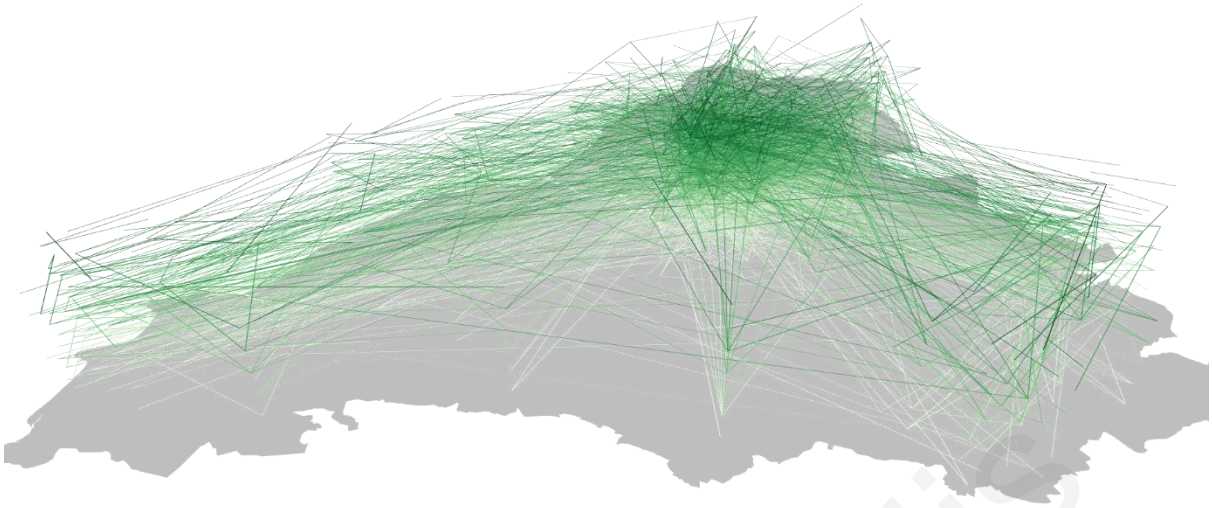

Figure 6.8 The multi-period ODs; darker tones indicate trips departing later in the day.

Figure 6.9 Individual colour-coded activity schedules.

6.2.1.2 Comparison of high-level distributions

As it has been already discussed, the activity schedules' combinations which optimally recreate the inputted travel demand may be of excessive number. In order to guide the optimiser towards a solution closer to reality, the distribution of activity schedules' high-level characteristics (i.e. calibration distribution) was enforced as a constraint. The bar chart depicted in Figure 6.10 validates the accuracy of the enforcement of the above-mentioned constraint. It can be noted that the discrepancies between the share of the observed and the modelled distribution groups are minimal. Moreover, since the absolute number of the observed activity schedules (25,000) is very close to the corresponding number of the modelled schedules (24,818), it can be deducted that the absolute number of activity schedules within each of the compared distribution groups is very similar.

The close resemblance between the two distributions was expected due to the formulation presented in Section 3.5.1. However, the visualisation of the comparison was executed solely to verify the proper implementation of the previously presented formulation.

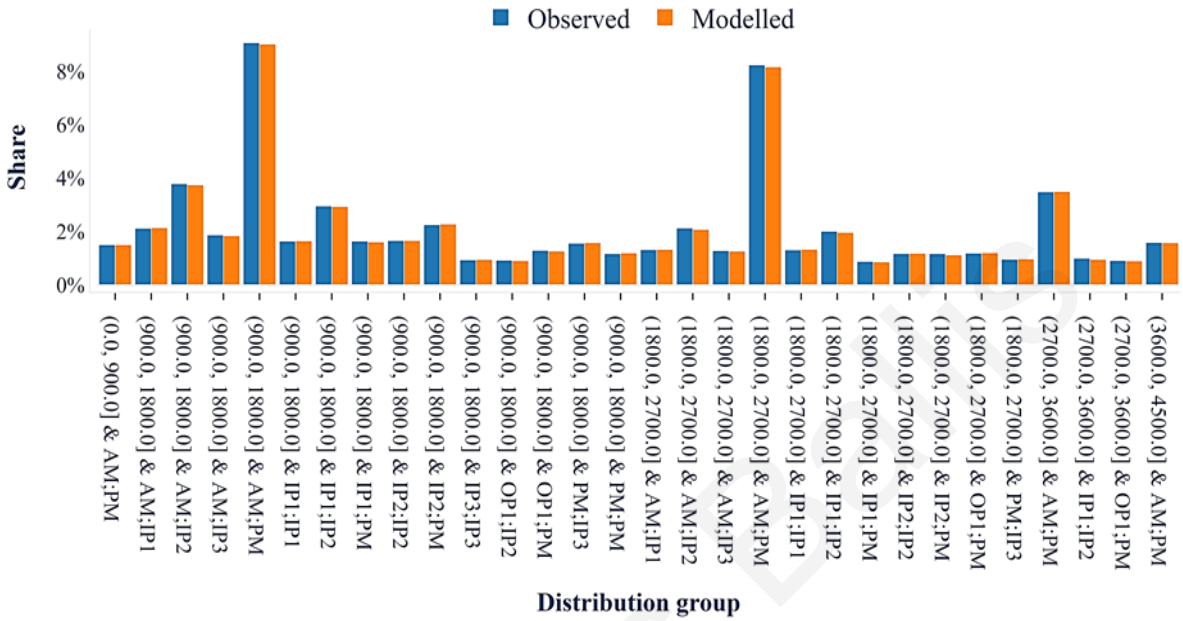

Figure 6.10 Comparison of the 30 distribution groups with the largest share between the observed and the modelled activity schedules.

Based on the previous, it can be argued that the presented methodology can indeed create a set of activity schedules able to accurately represent the total travel demand as described in the observed ODs. Hence, the high-level accuracy of the methodology has been successfully evaluated. Nonetheless, the characteristics of the modelled activity schedules may differ significantly in the microscopic level. For that purpose, the next section focuses on the one-to-one (i.e. disaggregate) comparison between the observed and the modelled activity schedules.

6.2.2 Disaggregate-level

The results presented in Section 6.1, validate the capability of the presented methodology to produce activity schedules able to retain the aggregate-level characteristics of the observed activity schedules and at the same time represent the total travel demand as described in the observed ODs. Nonetheless, the complete validation requires the comparison of the schedules at the disaggregate-level.

6.2.2.1 Comparative dimensions

As a first comment, the difference between the number of modelled activity schedules (24,818) and the observed ones (25,000) is particularly low (1.0%). Despite this low

difference, the output was also validated against the individual characteristics (*comparative dimensions*) of each schedule. The comparative dimensions which were evaluated correspond to the sequences of (a) the visited zones, (b) the departure time periods and (c) the activity type of each schedule.

For reasons of visual clarity, the onwards analysis focuses on the four most common (out of 24) activity type sequences and the ten most common (out of 256) departure time sequences while the rest of the sequences being classified as '*Rest*'. More information regarding this classification can be found in Table A.4 and Table A.5 of the Appendix. Additionally, the relevant *Home*, *Work* and *Other* activity types have been shortened to their initial (H, W, O). Finally, due to the high number of observed location sequences, the corresponding results are assigned a sequential numeric ID and are grouped in bins containing 2,000 location-sequences each.

6.2.2.2 Daily activity schedules

The individualistic nature of the resulting activity schedules allowed for a very thorough comparative analysis between the observed and the modelled activity schedules. The scatter matrix presented in Figure 6.11, depicts the pairwise scatter plots for the main dimensions describing each schedule. The dimensions are namely, the number of intermediate trips (legs), the total travel time, the code of each activity sequence type (e.g. [*Home, Work, Home*], [*Home, Work, Other, Home*], etc.) and finally their frequency. Finally, each circle on the plot represents an individual activity schedule. As it can be observed the proposed methodology has managed to accurately replicate activity schedules in all the above-mentioned dimensions. According to the scatter matrix, the modelled distributions are very similar to the observed ones and the variation of the observed patterns has been preserved to a great extent.

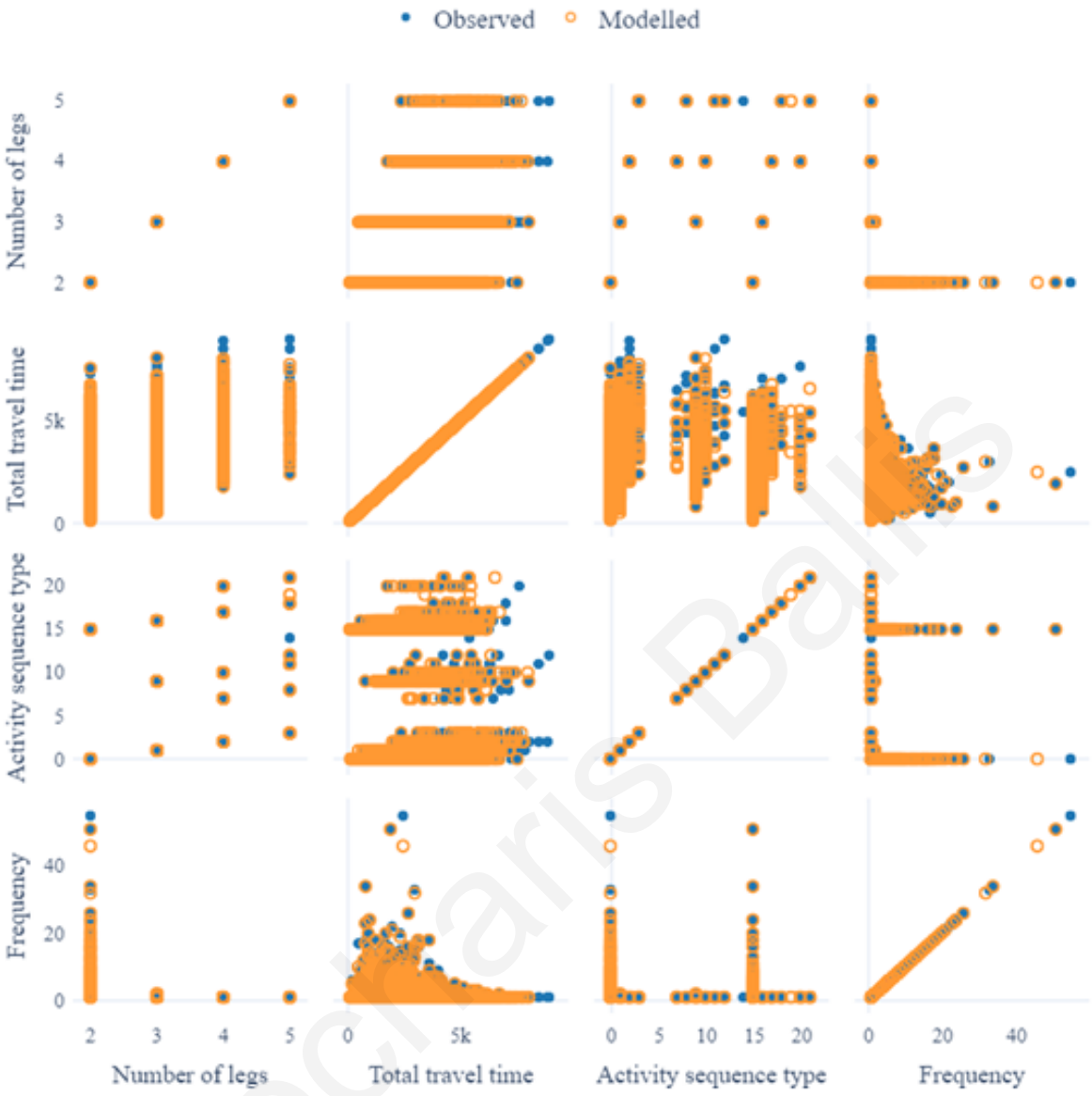

Figure 6.11 Scatter matrix analysis for the observed and the modelled activity schedules.

To further support the validation process, the percentage of the unmatched observed activity schedules is presented in Figure 6.12. An observed activity schedule is considered as unmatched when it cannot be paired with an equivalent modelled schedule. Since, the matching process takes place without replacement, the same modelled schedule cannot be assigned to more than one from the respective observed schedules. Depending on the number of the simultaneously considered comparative dimensions, the percentage of the unmatched schedules varies significantly. As it can be noted, the main dimension contributing to the misalignment between the observed and the modelled schedules is the combination of the location and the time period sequences. When examined in isolation, these two sequences attribute for 2.1-2.4% of the discrepancy but their simultaneous examination results in a misalignment of 9.51%. Finally, the percentage of unmatched observed schedules when the comparative dimensions are altogether considered does not exceed 9.55%. This encouraging

result provides strong evidence regarding the capability of the methodology to reproduce realistic multidimensional activity schedules based on limited aggregate input (marginal distribution).

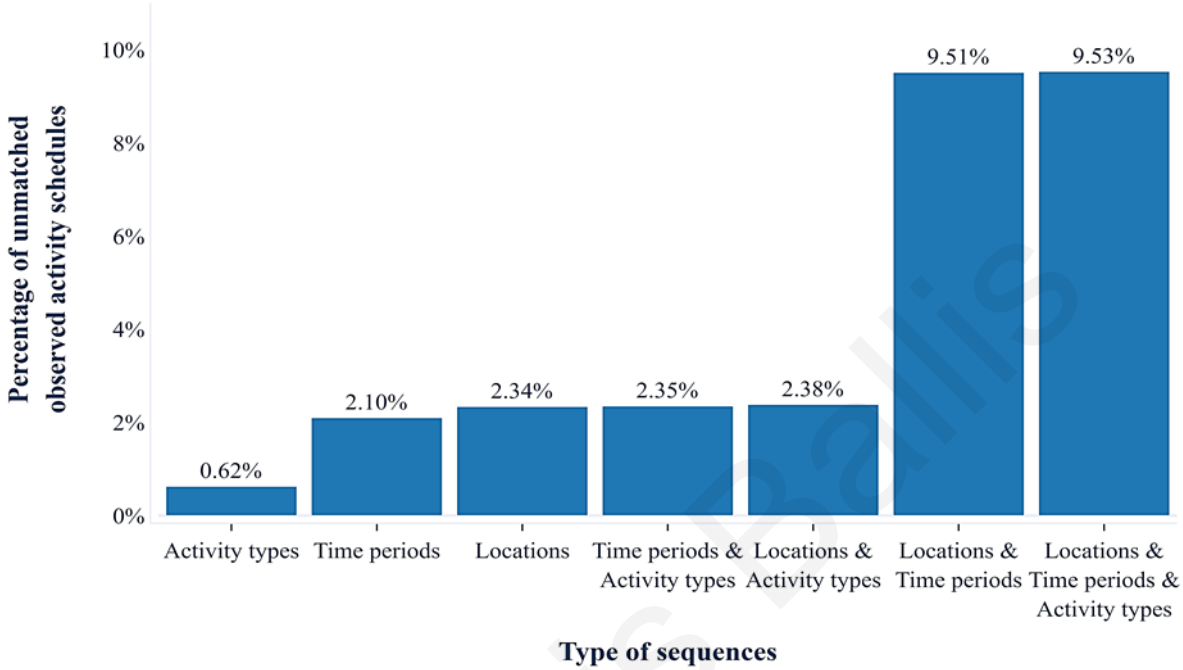

Figure 6.12 Examination of the comparative dimensions on the accuracy of the suggested methodology

The next section delves into the distribution of error (i.e. mismatch) among the comparative dimensions. Firstly, the distribution of error for each comparative dimension separately is depicted in Figure 6.13 to Figure 6.15. Blue bars represent the share of observed distribution groups while orange bars the percentage for unmatched schedules. As it can be noticed, the error term is distributed almost proportionally between the different groups across all the comparative dimensions. The low percentage error presented in Figure 6.13 can be attributed to the calibration distribution which controlled the number of schedules within each time-period sequence group. Up to some extent, this also holds true for the low error term presented in the distribution of error within the location sequence groups (Figure 6.14) due to the constraints indirectly imposed by the availability of trips within the input ODs. Although, the exact number of schedules following each location sequence is not known, the provided information regarding the total travel time between locations improves the quality of the output. Interestingly though, the methodology has accomplished a particularly accurate solution even for the unconstrained dimension regarding the sequence of activities (Figure 6.15). This is particularly important because the location and the activity type sequences have been endogenously estimated without relying on any calibration data.

Focusing on Figure 6.15 reveals that the error term is more notable for complex activity schedules (i.e. schedules including more than three activities) which are nonetheless infrequent and do not significantly affect the accuracy of the output. In cases where the activity type sequencing is an important factor of the analysis, the researcher should attempt to incorporate such information in the calibration distribution.

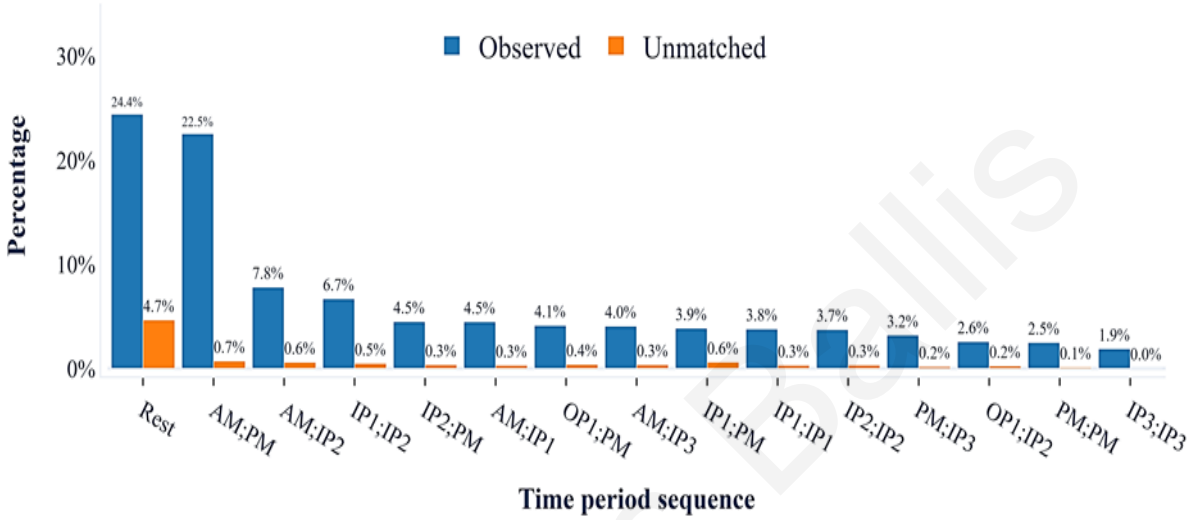

Figure 6.13 Percentage of unmatched activity schedules for the departure time periods sequence comparative dimension.

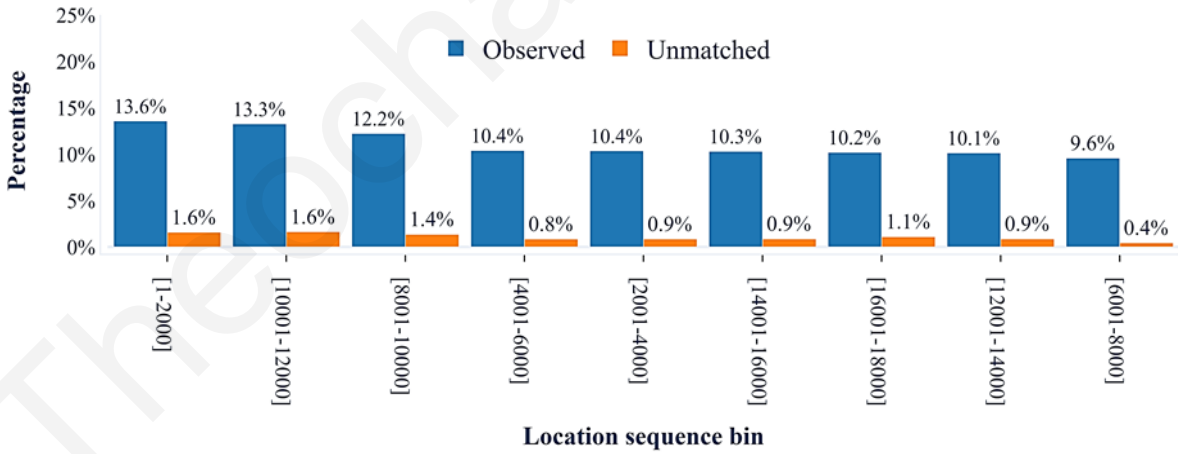

Figure 6.14 Percentage of unmatched activity schedules for the location sequence comparative dimension.

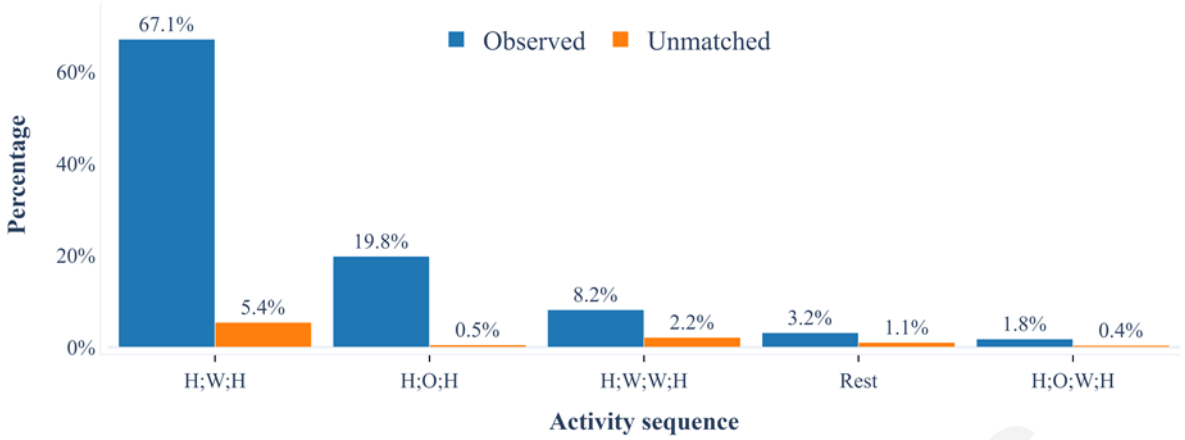

Figure 6.15 Percentage of unmatched activity schedules for the activity type sequence comparative dimension.

Finally, the presentation of the unmatched activity schedules is visualised through the parallel categories diagram of Figure 6.16. In this diagram each individual activity schedule is presented as a string crossing through its defining characteristics. The activity schedules which were not perfectly matched are clearly depicted with orange colour. This visual representation emphatically showcases the high accuracy of the suggested methodology since only a relatively small percentage (9.53%) of the output does not fully comply with the considerably complex input.

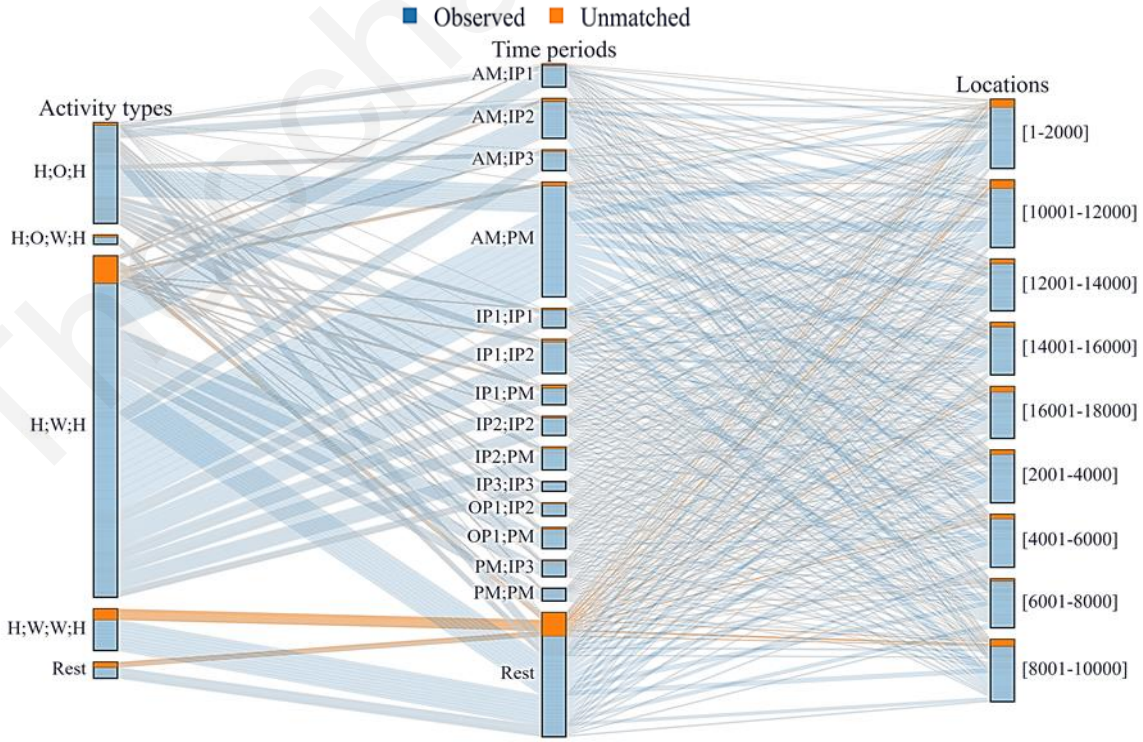

Figure 6.16 Presentation of the unmatched activity schedules between the observed and the modelled ones.

6.2.2.3 Activity participation profiles

The combination of individual trips within ODs into tours and subsequently in activity schedules produces additional insight, useful for further descriptive behavioural analysis such as the participation of the population in different activities. Figure 6.17 depicts the distribution of activities taking place in the studied area over the course of a day while presents the percentage differences between the observed and the modelled activity schedules in terms of their activity participation profiles. As it can be noticed, the comparison between the corresponding distributions assures that the methodology can replicate the observed patterns with great fidelity since only low percentage errors (less than 4%) can be noted between the observed and the modelled figures (Figure 6.18).

Figure 6.17 Distribution of participation in different activities throughout the day

Figure 6.18 Percentage difference between the observed and the modelled participation for different activities throughout the day.

6.2.2.4 Departure time profiles

Another set of comparisons between the observed and the modelled activity schedules revolved around their departure time profiles. As it can be noticed in Figure 6.19, the proposed estimation framework has managed to replicate the trend of departures between the observed and the modelled schedules without significant discrepancies. Peaks and troughs arise almost at the same time, while the rates of departures are generally similar. Considering that the assignment of the exact departure time for each trip was based on a uniform distribution, it becomes apparent that the methodology can accurately estimate realistic activity schedules even with limited input. In the case where the duration of activities plays a crucial factor to the analysis, the researcher can incorporate relevant information as optimisation constraints.

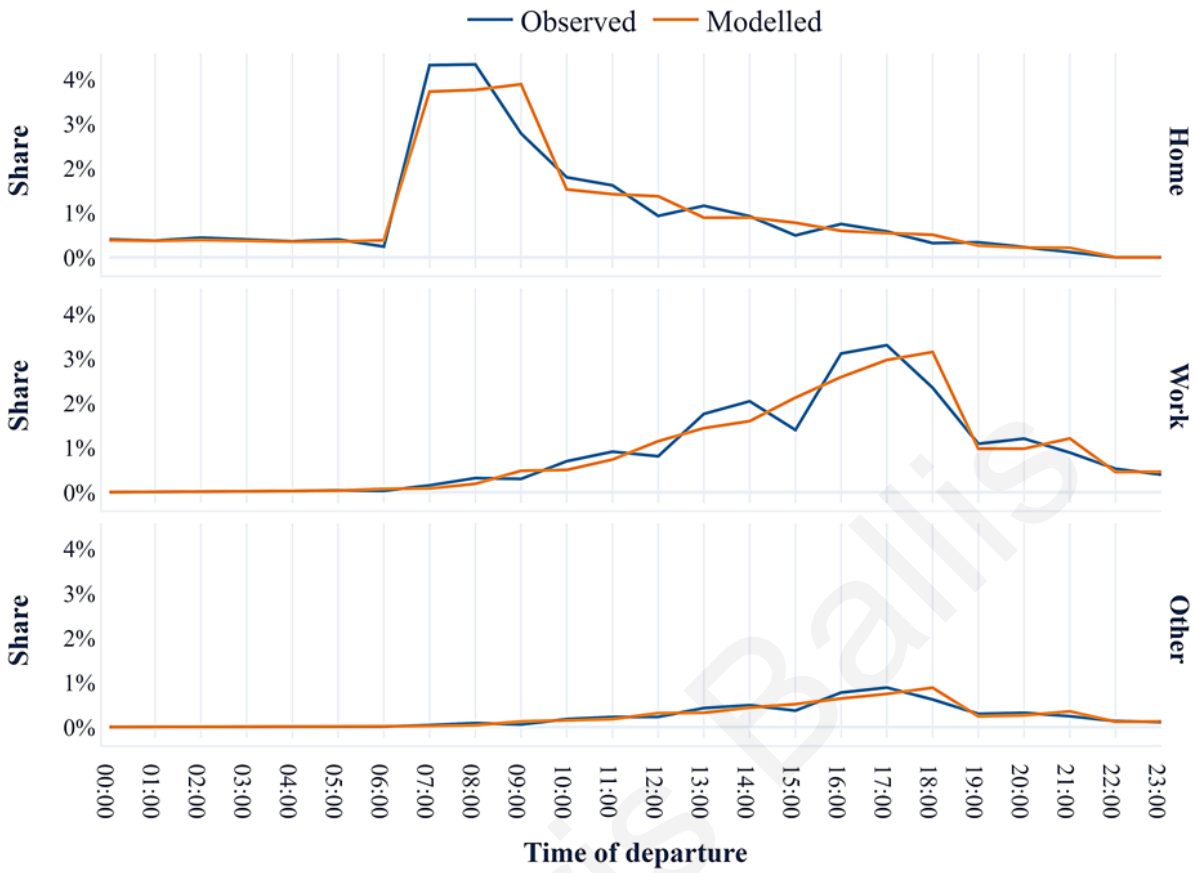

Figure 6.19 Departure profiles for the available activity types.

6.2.2.5 Duration of activities

The final stream of comparative analysis emphasised on the duration of activities within the observed and the modelled schedules. Figure 6.20 depicts the distribution of the duration for the available activity types classified into bins of 1-hour duration. As it can be noticed, the percentage difference for most of the cases is below 0.5% with only a couple of exceptions related to short duration activities. As stated earlier, these discrepancies can be potentially eliminated by the application of a more refined departure time profiles.

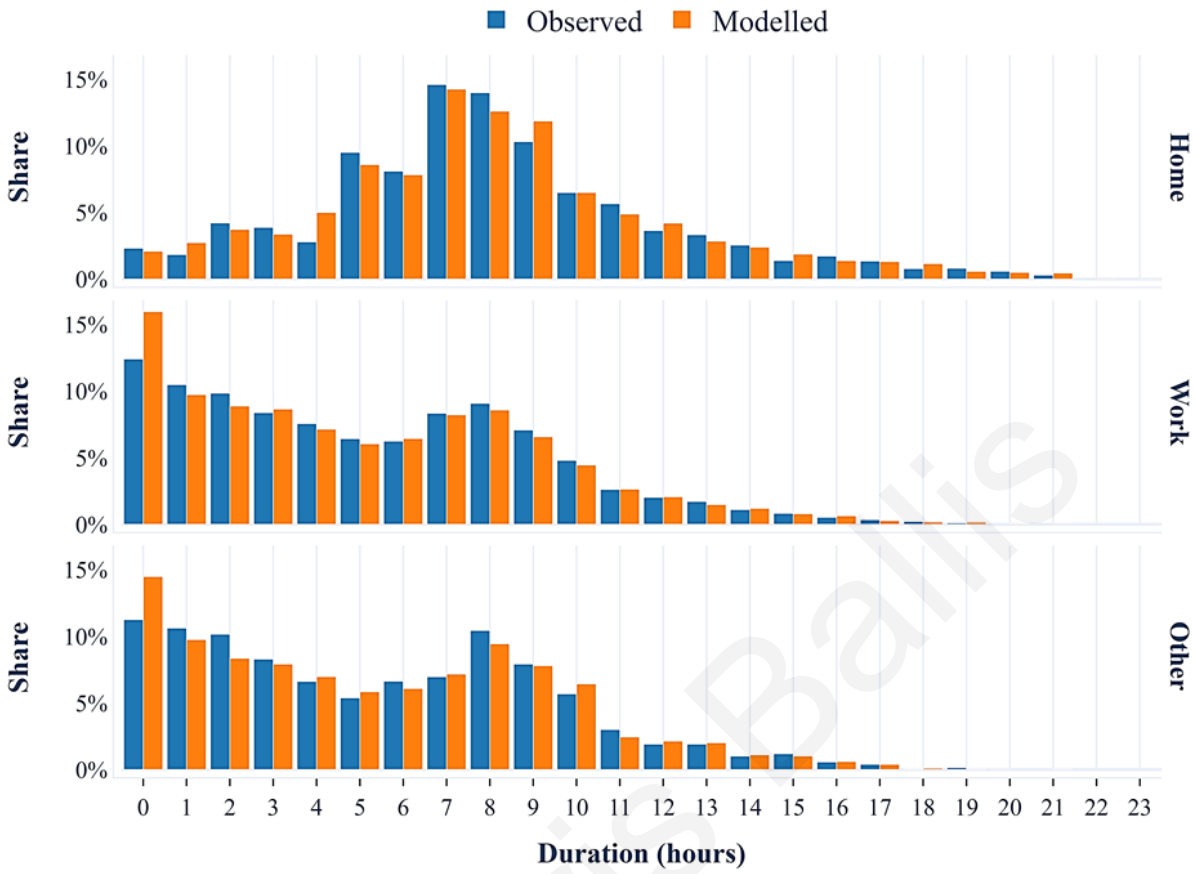

Figure 6.20 Duration profiles for the available activity types.

The previous section meticulously validated the ability of the methodology to produce activity schedules which closely resemble the observed travel behaviour patterns. The implications are considerable since it becomes evident that particularly detailed travel behaviour patterns can accurately emerge when aggregated data sources such as OD matrices and a high-level calibration distribution are smartly utilised. The next section elaborates on the travel behavioural information which can derive from the application of the methodology.

6.3 Travel behaviour analysis

The following section highlights the additional insight that can be drawn when aggregate ODs are converted to fully tractable activity schedules. Traditional ODs assume independency between trips therefore no assumption regarding the duration of stay between consecutive activities can be made. Nonetheless, the application of the suggested methodology allows the inference of travel patterns in great depth, something which is not possible solely on the direct analysis of aggregate OD matrices.

6.3.1 Activity Participation

With respect to the profile of activity participation, the distribution of activities taking place during the day in the studied urban area as well as a sample of zones is presented in Figure 6.21 and Figure 6.22. As it can be observed, the patterns and the mixture of activities vary considerably across zones. Some areas present a balanced composition of activities while others present a skewed profile towards working or recreational activities. On the other hand, the aggregate diagram for the whole of the studied area presents the arguably expected pattern, with most of the out-of-home activities taking place between 08:00 to 17:00. The comparison between the aggregate and the per-zone analysis, highlights the multiple activity profiles which can arise depending on the characteristics of each zone.

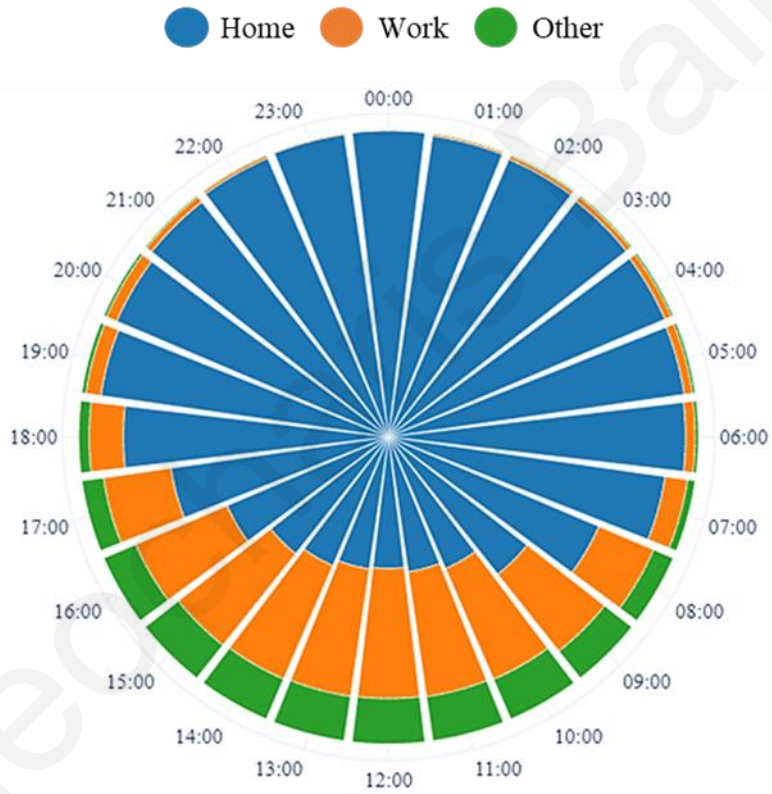

Figure 6.21 Profile of activity participation for the studied urban area.

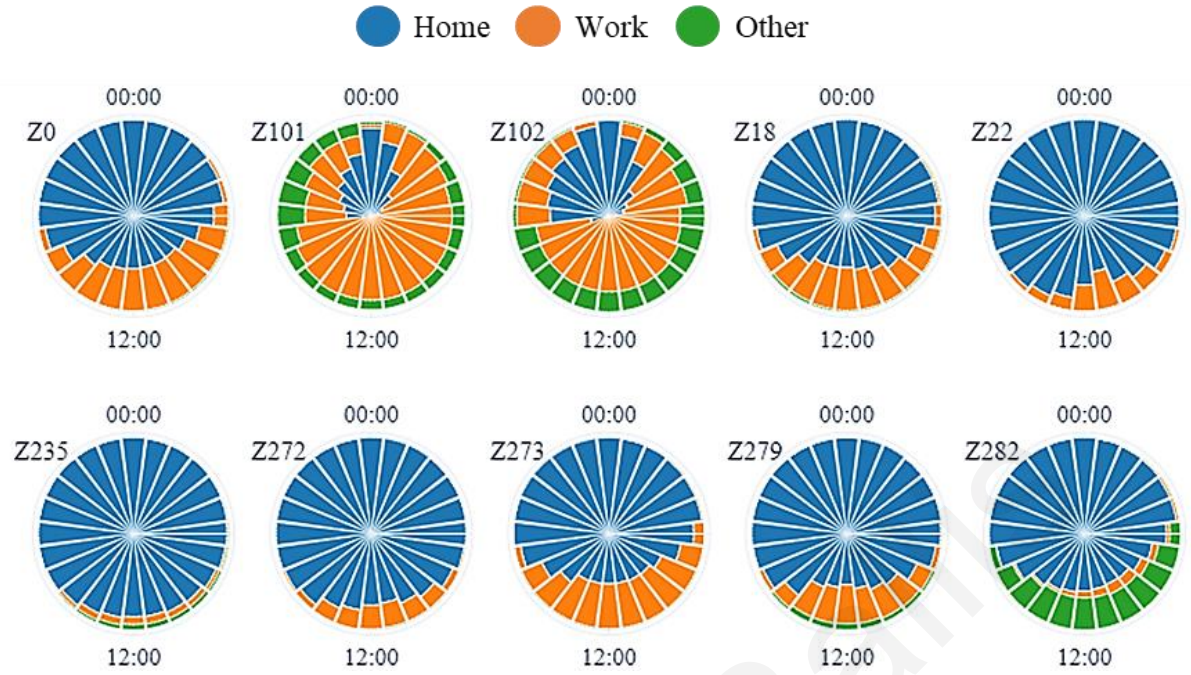

Figure 6.22 Profile of activity participation for a set of sampled zones.

On a similar stream, Figure 6.23 presents the daily distribution of activities taking place at twenty-five, randomly sampled zones with an interval of 30 minutes. As it can be observed the activities' patterns between zones are dynamic and can vary significantly during the day. As an example, some zones (e.g. Z0, Z235, Z372) can be classified as purely residential since they are mostly occupied by their residents regardless the time of day. On the contrary zones Z101, Z448 and Z456 are mainly visited for work purposes. Nonetheless, more diverse patterns can also arise like in zone Z4 which is primarily visited for recreational and secondarily for work related activities. Another observation is that the primary activity executed in a zone can change numerous times during the day. For instance, zone Z102, is mostly occupied by its residents in the early morning, then flooded with workers until the evening when it is visited for recreational activities before its residents return home later in the night. Such detailed information regarding the activity profile of different areas can prove particularly useful for policy-making and urban planning purposes.

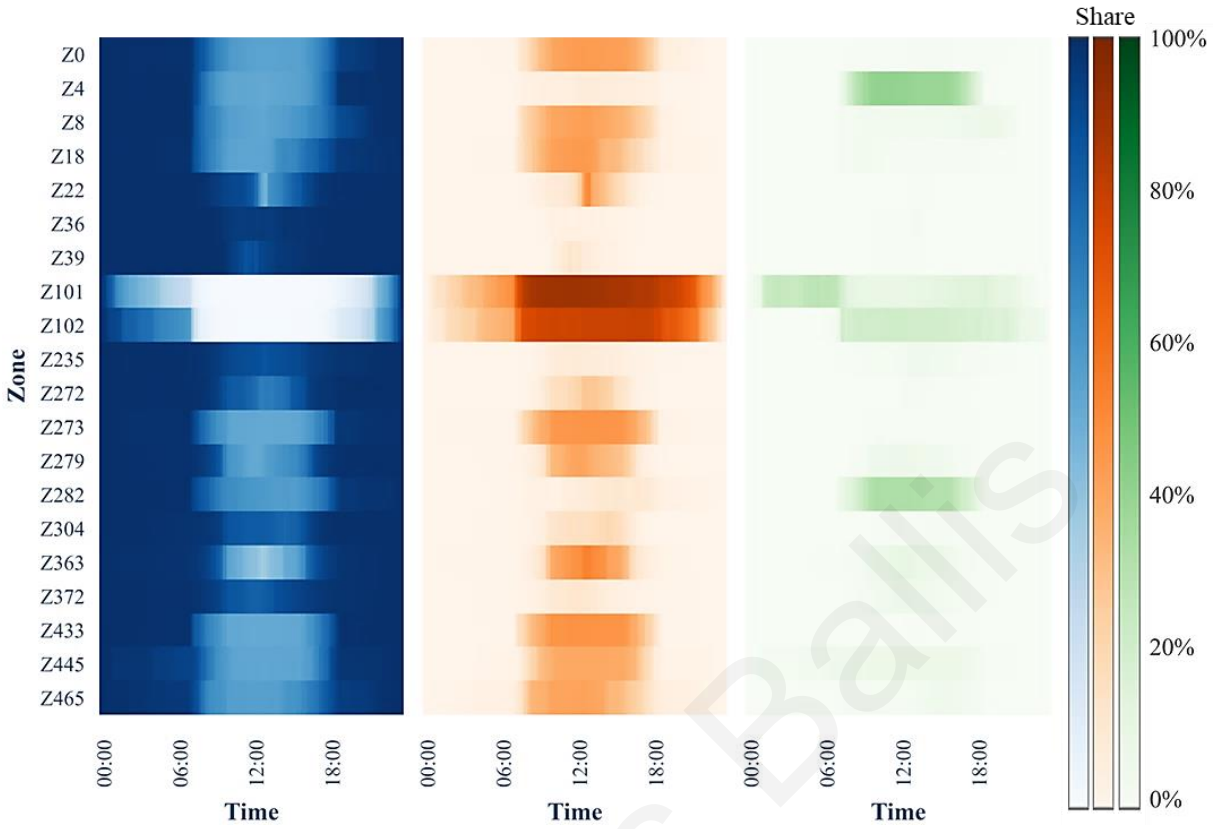

Figure 6.23 Daily distribution of activity-participation for 25 randomly selected zones.

6.3.2 Activity Duration

Apart from the concentration of people participating in different activities, useful information can be deducted by the inclusion of the duration of the executed activities into the analysis. Prior to the interpretation of the relevant results, it must be reminded that all the studied activity schedules were completed within a single day, therefore a gradual decrease of the total and the average duration is expected. Figure 6.24 depicts the average remaining time for participation in the available activities. Studying each activity type separately leads to useful insights. For instance, the average duration of stay for activity type Home is long in the early morning hours as well as in the late evening when people are indeed likely to remain at their residencies for longer durations. Likewise, the methodology accurately captures the short-duration trips in the morning period (08:00 to 11:00) which can be attributed to short duration errands (e.g. taking children to school). On the other hand, the profile for the duration of stays for Work activities differs significantly. As expected, for average duration for most of the zones is close to 8 hours for arrivals to workplace between 8:00 and 10:00. Moreover, early morning workers seem to spend considerably more time at their workplace compared to those who arrive later. Finally, with regards to the Other activity types, considerable spatiotemporal variation is observed. With the exception that

most zones become attractive for the participation in Other activity later in the day (gaps in the early hours), no other patter seems to emerge, since the average duration of stay fluctuates both across zones and time. This element emphasises the inhomogeneous patterns observed for the activities classified as Other.

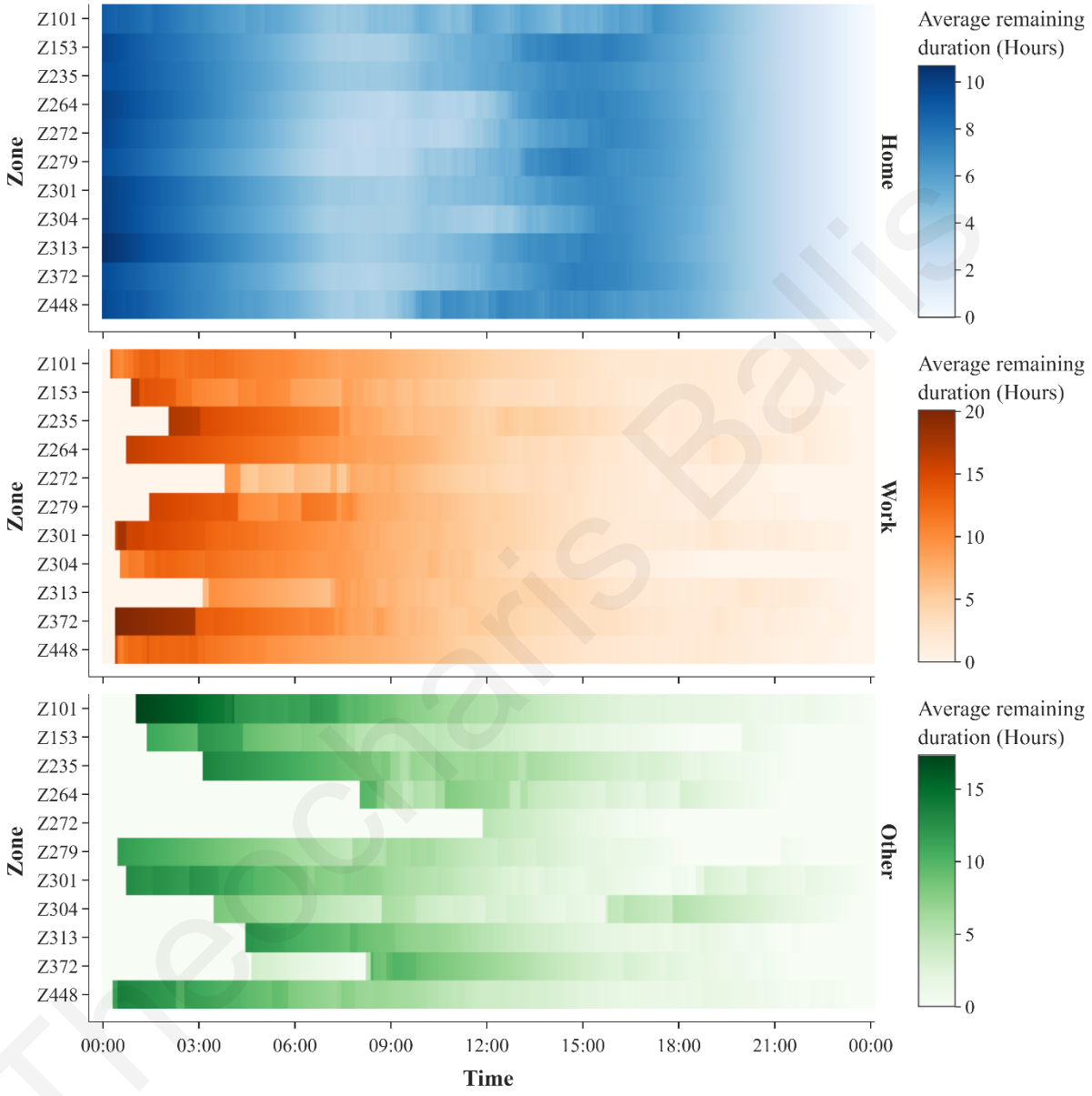

Figure 6.24 Presentation of the average remaining duration of participation in activities by time of arrival and activity type.

Apart from the concentration of people participating in different activities, useful information can be deducted by the inclusion of the duration of the executed activities into the analysis. Figure 6.25 presents the average as well as the total remaining duration of participation depending on the time of arrival and the type of activity. Prior to the interpretation of the relevant results, it must be reminded that all the studied activity schedules were completed within a single day, therefore a gradual decrease of the average

duration is expected. The left-hand side of the figure (Figure 6.25a) depicts the average remaining time to participate in the available activities. Studying each activity type separately leads to additional useful information. For instance, the average duration of stay for activity of type Home is long in the early morning hours and in the late evening when people are indeed not likely to leave their residencies in short time. Likewise, the methodology is able to capture the short-term returns of people to their homes in the morning period (08:00 to 11:00) which can be attributed to purposes such as taking kids to school or taking care of errands prior to leaving for work. On the other hand, the profile for the duration of stays for Work activities differs significantly. As expected, for average duration for most of the zones is close to 8 hours for arrivals between 8:00 and 10:00. Nonetheless, for specific zones, people arriving to their workplaces in the early morning hours tend to spend considerably more time that what it would be expected (e.g. 8-10 hours). The disaggregate analysis allows for the identification and a potential closer investigation for such cases. Finally, with regards to the rest of the activity types, considerable spatial and temporal variation is observed. With the exception that most zones become attractive for the participation in Other activity later in the day (gaps in the early hours), no other apparent pattern seems to emerge, since the average duration of stay fluctuates both across zones as well as through time. This element emphasises the inhomogeneous patterns of the activities classified as Other.

The right-hand side of Figure 6.25 focuses on the total remaining duration and allows the comparative analysis between the sampled zones. As is becomes evident, the attractiveness of certain zones for different activities is considerably greater compared to the rest. As an example, zones Z101 and Z102 concentrate a significantly larger portion of the Work and Other activities while Z4 is an attractive destination only for activities classified as Other.

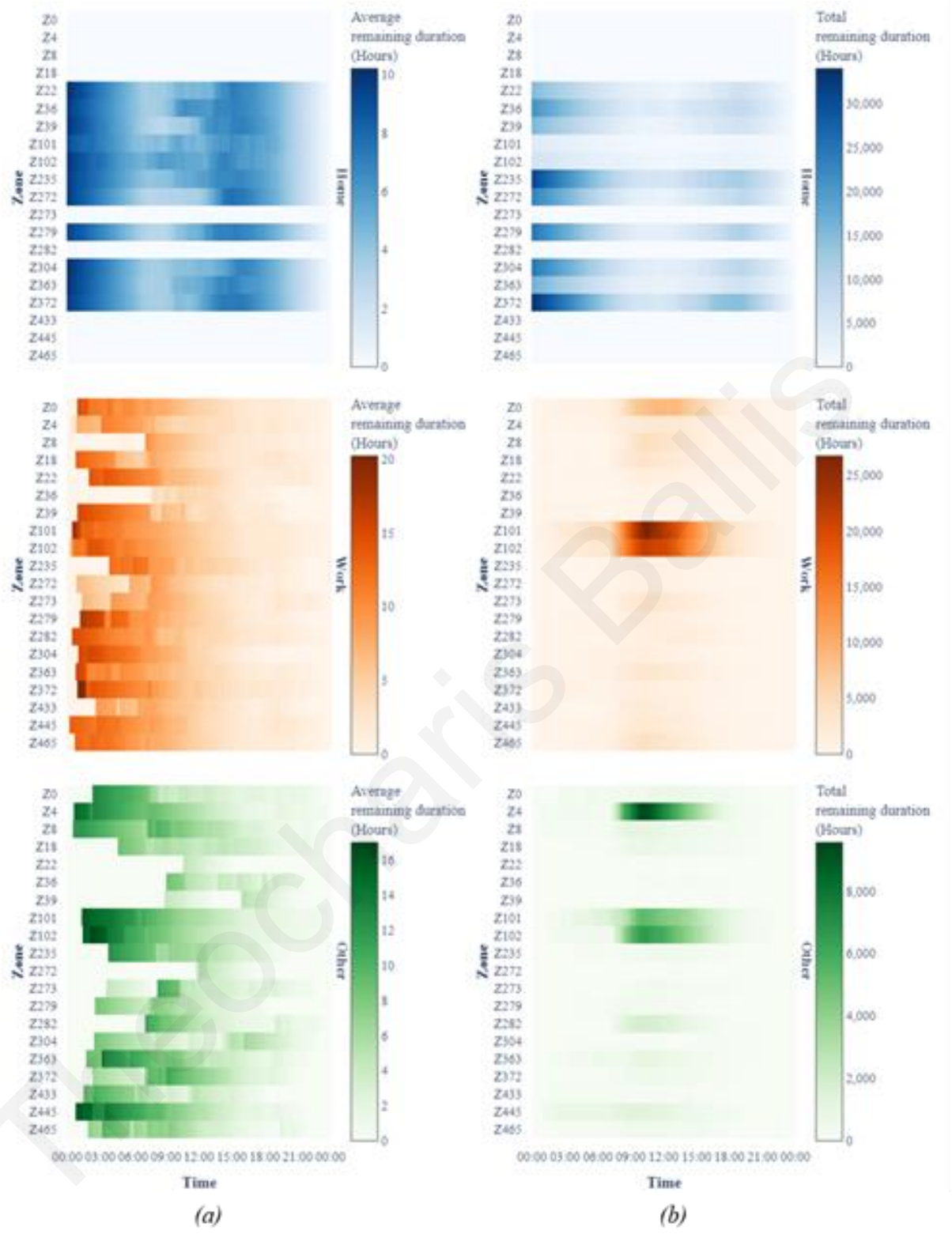

Figure 6.25 Comparison between the average and the total remaining duration of participation in activities by time of arrival and activity type.

6.3.3 Geospatial Analysis

The last set of travel behaviour analysis attempts to include the geospatial factor in the process. Figure 6.26 depicts the spatial distribution of people being at their workplace in the

wider area of Bristol, UK, at different times of the day. As a first comment, it is obvious that the centre of Bristol is more attractive as a workplace compared to the rest of the area, regardless of the time of the day. Nonetheless, other areas in the outskirts seem to also attract a considerable share of the working population, although this share fluctuates within the day. As the day progresses, the participation in work related activities diminishes but not at the same rate for all zones, since some retain high numbers of workers even in the late evening. Including the geospatial dimension enables the identification of hotspots for different activities both in space and time and therefore allows for the evaluation of sophisticated policy scenarios.

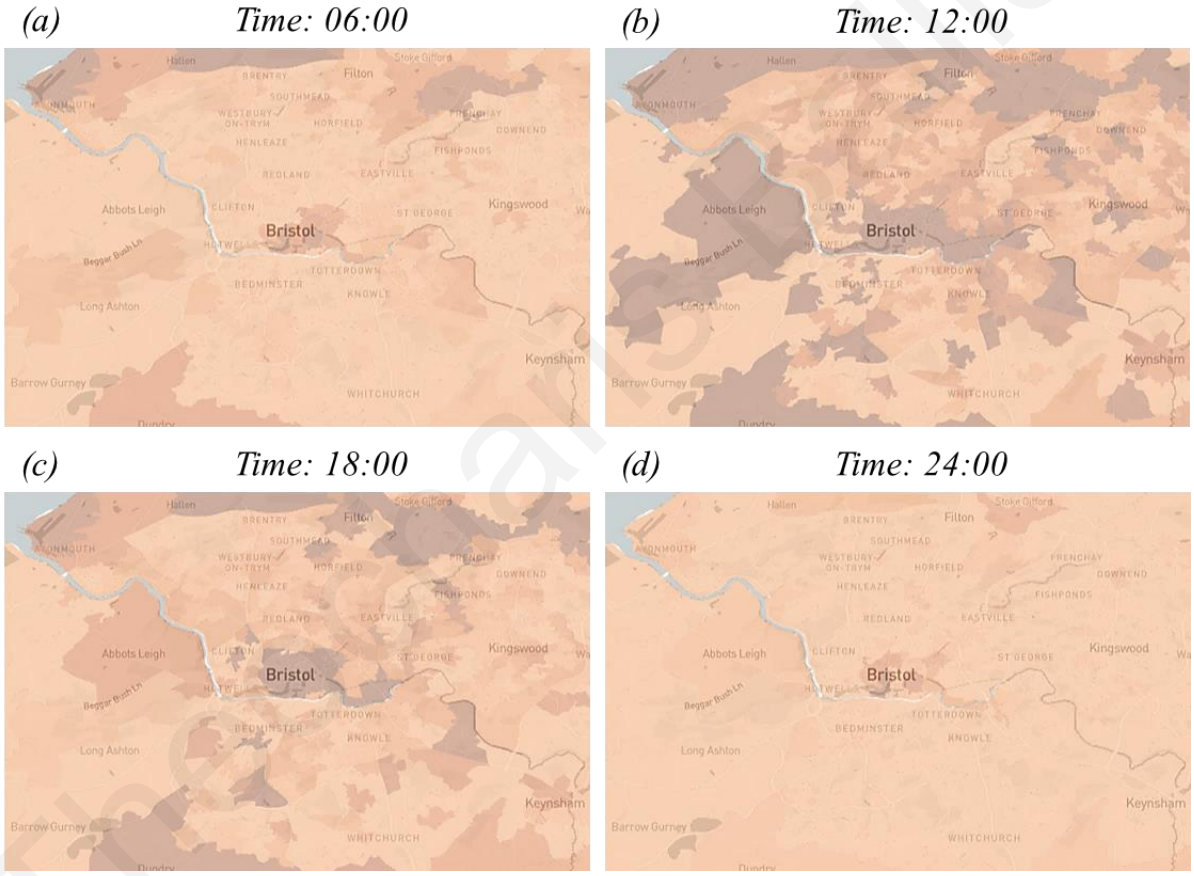

Figure 6.26 Progression of participation in 'Work' type activities during a day; Darker tones indicate higher participation.

The previously presented analysis does not aim to perform exhaustive explanatory analysis on the observed travel behaviour patterns for the studied area but rather to highlight the degree and the detail of the information that can be extracted from aggregate OD data. The direct analysis of ODs could have by no means provided enough information to complete a meticulous study of travel behaviour and urban dynamics. In contrary, the application of the

suggested methodology allowed for an in-depth analysis able to unveil and properly capture the dynamic nature of urban environments.

6.4 Effect of the Zoning System's Resolution

The previously presented methodological Section 4.1.2 highlighted the effect that the zoning system of the input ODs, can have on the accuracy and the efficiency of the suggested methodology. The next section completes the quantification of this effect by re-running the process on the same set of observed activity schedules but now expressed in a coarser zoning system. The aggregation of the previously utilised zoning system of LSOAs to larger groups results in a coarser census geographic boundary referred as 'Middle Layer Super Output Areas' (MSOAs). Each MSOA contains a mean population of around 7,200 people. Since MSOAs emerge as pure aggregation of LSOAs therefore a direct mapping between them does exist. Aggregating the initial 470 LSOA zones to the corresponding MSOAs, resulted in a low-resolution zoning system of 140 zones (Figure 6.27). The conversion from the high- to the low-resolution zoning system led to a 70% reduction in the number of zones with a subsequent eightfold increase of the network density (Table 6.5).

Table 6.5 Summary of zoning-systems used for the synthesis of the observed tours.

Spatial Resolution	**Based on**	**Zones**	**Network density (%)**
High	LSOAs	470	0.44
Low	MSOAs	140	3.52

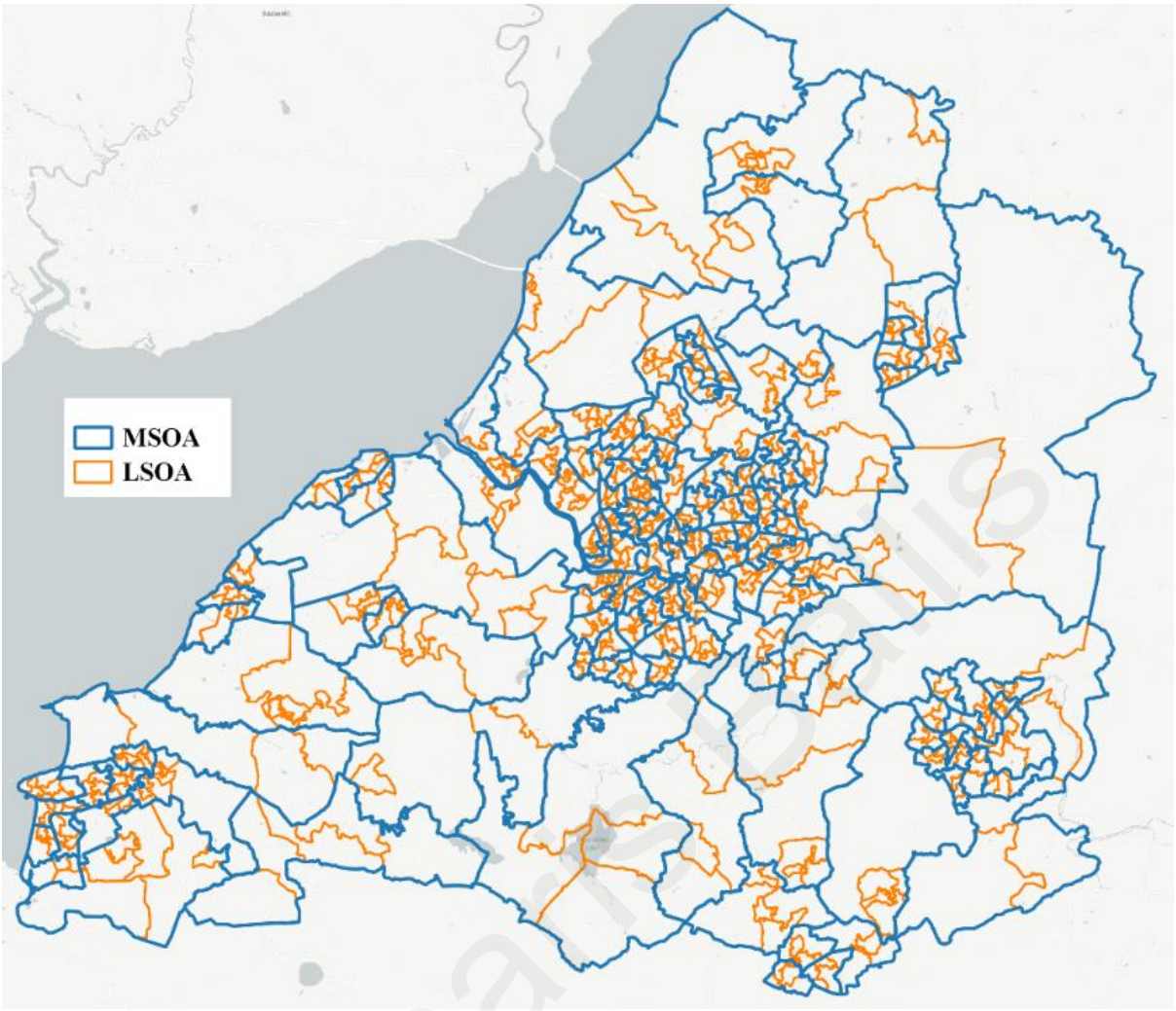

Figure 6.27 Presentation of the modelled area (Bristol, UK) and the high-resolution (LSOA) and the low-resolution (MSOA) zoning systems.

6.4.1 Processing Time

Table 6.6 presents the processing time requirements of the methodology by zoning system and processing module. All scenarios were completed on Intel® Xeon CPU powered computer with 32GB of available RAM. As it can be noticed, the use of low spatial resolution leads to considerably higher identification processing time requirements (nine-fold increase). On the other hand, the increase of the required optimisation processing time does not exceed 60%. This comparison highlights the influence of the zoning-system's resolution on the processing time requirements of the suggested methodology. Fine-grained zoning systems prove more suitable for the purposes of the suggested methodology and should be preferred if given the option. However, the methodology proves applicable even for coarse zoning systems since the overall processing time for the low-resolution scenario is still reasonable (less than 2 days) and comparable to other transport modelling tasks of similar scale (e.g. microsimulation traffic assignment). It should not be disregarded that problems

of combinatorial nature like the one presented above can often prove particularly cumbersome to solve even with state-of-the art methodologies and high-end computing resources (Klotz and Newman, 2013). Therefore, the previously mentioned solving times can be considered satisfactory while additional processing time reduction can occur if computational systems with multiple threads are utilised.

Table 6.6 Processing time requirements per scenario

Spatial Resolution	Identification module (sec)	Optimisation module (sec)	Total processing time (sec)
High	12,028	15,454	27,482
Low	105,530	44,000	149,530

6.4.2 Comparison of ODs

The increased processing time for the low-resolution scenario does not affect the accuracy of the output in terms of the resemblance between the modelled and the observed ODs. For the low-resolution scenario, the total trip difference was calculated at 0.9%. The breakdown of the missing trips is presented in Table 6.7. Results showcase the capability of the methodology to identify accurate solutions even with coarser zoning systems.

Table 6.7 Absolute trips difference between the observed and modelled ODs (low-resolution scenario).

Purpose	OP1	AM	IP1	IP2	IP3	PM	OP2	Total
HBW	3	13	12	11	3	21	0	**63**
HBO	25	104	54	70	38	100	3	**394**
NHBW	0	1	0	0	0	0	0	**1**
NHBO	0	0	0	0	0	0	0	**0**
Total	**28**	**118**	**66**	**81**	**41**	**121**	**3**	**458**

The minor effect of the zoning system on the OD matrix estimation accuracy of the methodology is also presented in Figure 6.28. The similar characteristics (particularly high R^2 values and slope very close to 1) between the scatter graphs of the low- and the high-resolution scenarios indicate the minor effect that the zoning system imposes on the quality of the solution, in terms of OD replication.

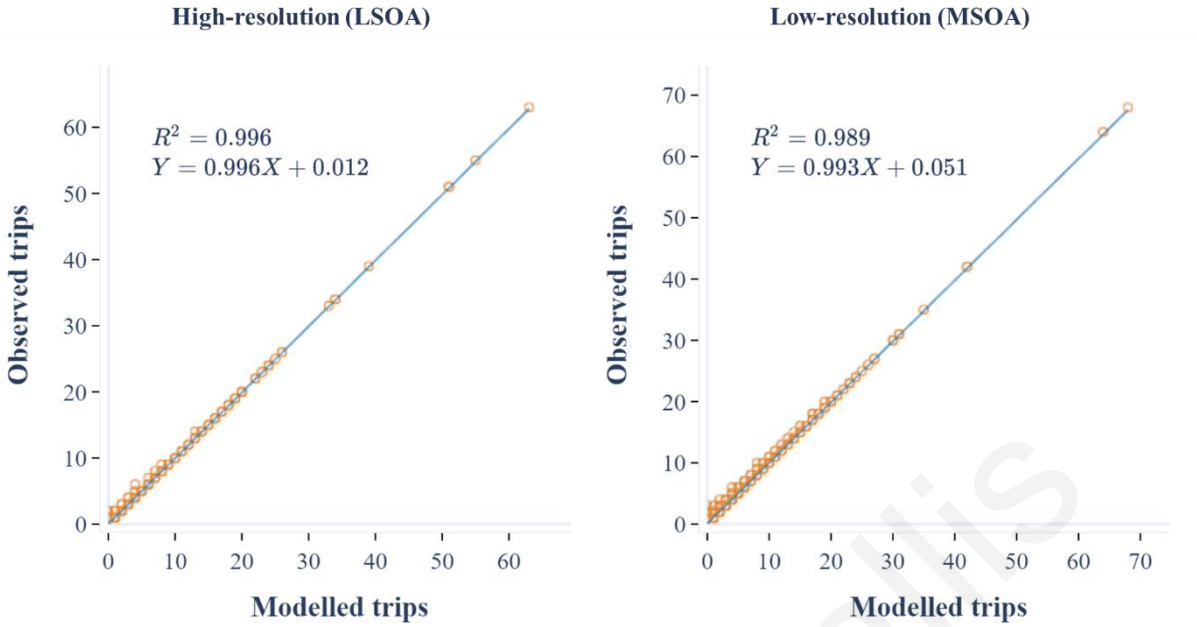

Figure 6.28 Comparison of OD matrix resemblance (Observed vs Modelled) for the high- and the low-resolution zoning systems.

6.4.3 Fidelity of the Modelled Activity Schedules

The previous section verified the minor effect of the spatial resolution on the replication of the modelled and the observed ODs. However, the decrease of the zoning system's resolution is considerably more significant with regards to the resemblance between the observed and the modelled activity schedules. Despite the similarity of their high-level characteristics (Figure 6.29), the one-to-one comparison between the activity schedules unveils a less accurate matching.

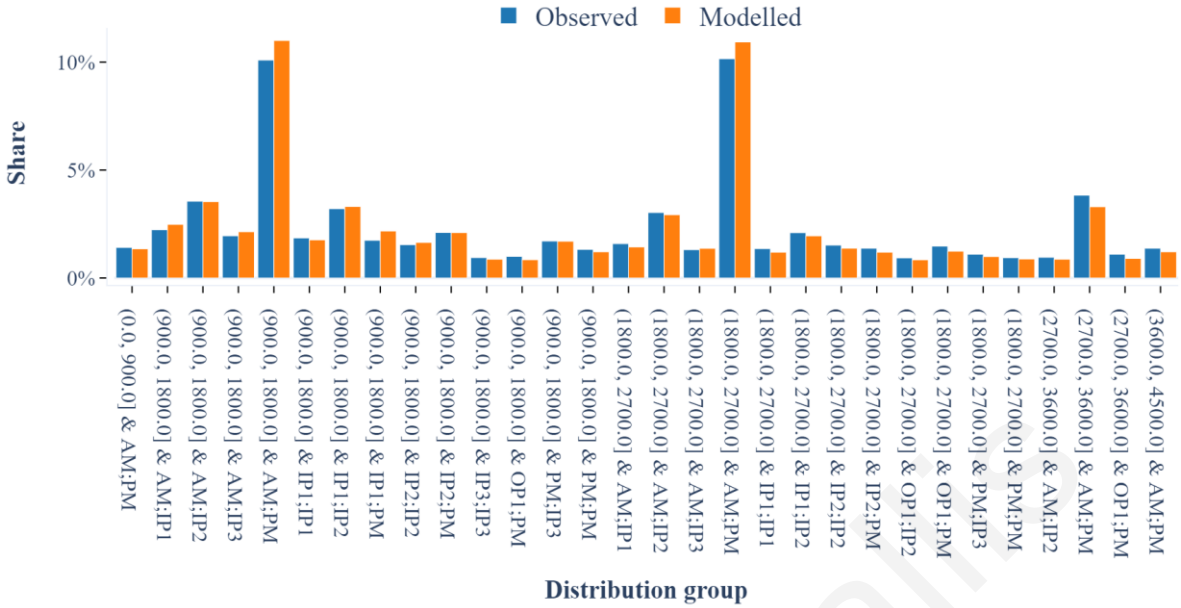

Figure 6.29 Comparison of the 30 distribution groups with the largest share between the observed and the modelled activity schedules (low-resolution scenario).

In detail, Figure 6.30 depicts the comparison of the modelled and the observed activity schedules for the high- and the low-resolution scenarios. The scatter diagrams compare the schedules in terms of (a) the zone sequences, (b) the departure profiles, (c) the activities sequence and (d) the travel time group each schedule belongs into. For example, a point on the scatter plot can represent the schedule visiting zones E01014370 and E01014377, departing in the AM and the PM time periods, going from Home to Work and with a total travel time between 1800 and 2700 seconds. As it can be observed, the correlation between the modelled and the observed schedules is particularly high in the case of the high-resolution scenario. On the other hand, the correlation is significantly lower for the low-resolution case. This is due to the higher number of candidate activity schedules and the considerably increased number of solutions recreating the observed ODs. However, the results can be still considered encouraging since, even for the low-resolution scenario, the methodology has managed to accurately reproduce the calibration distribution, as well as to perfectly replicate a large proportion of the observed activity schedules.

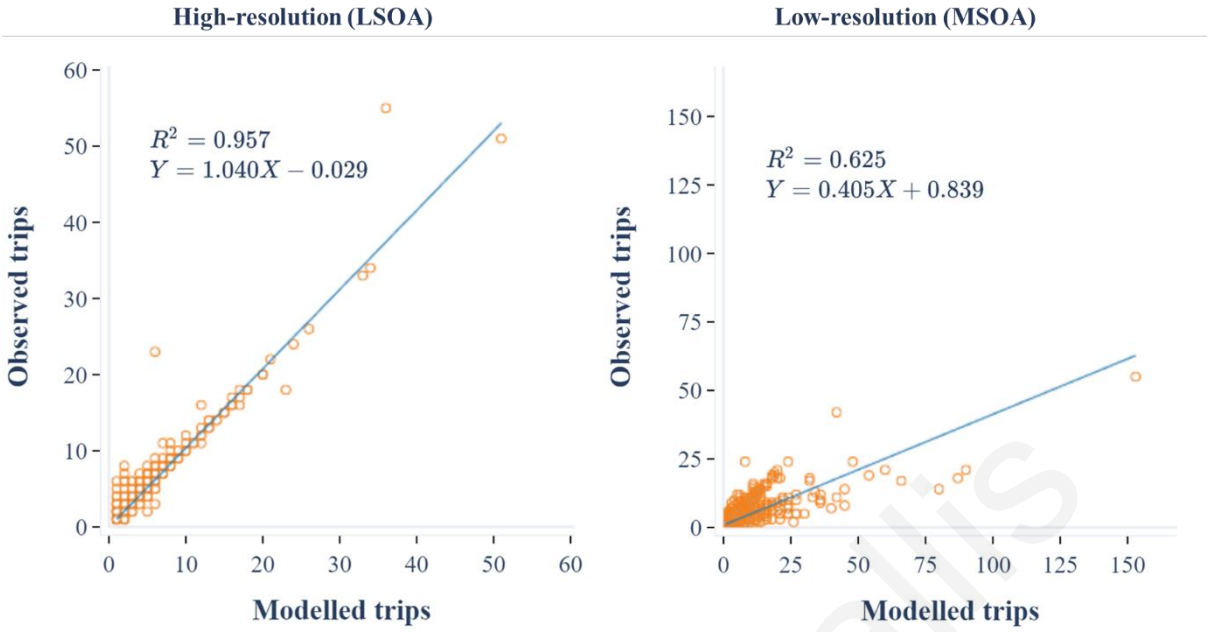

Figure 6.30 Comparison between the identified tours in terms of zone sequencing, profile of departures and activity sequencing.

6.5 Discussion of the Results

The previous sections evaluated the potential of the suggested methodological framework through a suitable proof of concept application. The evaluation began with the description of the utilised dataset and continued with the meticulous validation of the methodology on multiple levels. In particular, the evaluation took place both at the aggregate as well as at the disaggregate level. The aggregate-level assessed the ability of the methodology to identify a combination of activity schedules able to represent travel demand as captured in the inputted ODs. The disaggregate-level focused on the assessment of the outputs' realism and entailed the one-to-one comparison between the observed activity schedules and the modelled ones. Finally, the effect of the resolution of the utilised zoning system was also quantified.

The previously presented results verify the suggestion that multi-period, purpose-dependent ODs can be indeed transformed into individual activity schedules, suitable for in-depth travel behaviour analysis. A set of 25,000 thousand observed activity schedules were aggregated into the corresponding set of ODs describing the travel demand required for the completion of the schedules. Inputted with these ODs and a calibration distribution regarding the total duration and the departure profile of the observed schedules, the methodology managed to reverse-engineer the ODs into a set of modelled activity schedules, very similar to the observed ones. In particular, the methodology utilised 99% of the inputted travel demand for the synthesis of a set of modelled activity schedules where almost 9 out of 10 modelled

activity schedules were identical to the observed ones. It should be also stated that the methodology proved particularly efficient since it was completed in short time (approximately three hours) using a standard computing system. The evaluation process also stressed the effect of the utilised zoning system's resolution on the quality of the output. Results obtained from a case where a coarse zoning system was used resulted in highly accurate travel demand patterns (1% difference between observed and modelled ODs) but less accurate replication of the individual observed schedules ($R^2 = 0.62$). In addition, the utilisation of a coarse zoning system resulted in significantly higher processing time requirements (ten-fold increase). Finally, the travel behaviour analysis conducted for the purposes of the evaluation process, highlighted the additional value obtained from the disaggregation of ODs to individual activity schedules. As presented in the previous section, the disaggregation of ODs can lead to significant insight regarding travel behaviour. The rich travel behaviour information contained within ODs would have remained unobserved without the application of the suggested methodology.

Chapter 7
Large-scale Implementation

Chapter 7 evaluates the scalability of the methodology through its application on a set of large-scale ODs deriving from hundreds of thousands of individual activity schedules. This Chapter also verifies the effectiveness of the ASSA algorithm for addressing combinatorial problems of excessively large size.

7.1 Introduction

The earlier presented analysis for the proof of concept application verified the potential of the methodology for the conversion of multi-period, purpose segmented ODs to individual activity schedules. Although, the size of the proof of concept case was considerable, many realistic ODs are of larger size. The current Chapter evaluates the scalability of the methodology over significantly larger input. To distinguish the results obtained from the proof of concept scenario and the currently presented one, the latter is referred as the *large-scale scenario*. For reasons of brevity, the large-scale scenario focuses on the identification of tours within OD matrices instead of activity schedules since the scalability of the methodology is hindered by the identification and the optimisation modules and not by the activity-scheduling module which enables the conversion of tours to activity schedules. Finally, the large combinatorial optimisation problem used to assess the scalability of the methodology was also used for the evaluation of the ASSA algorithm.

7.2 Preliminary Analysis

Prior to the application of the methodology on the full population of tours (large-scale scenario), the input dataset used for the evaluation of the proof of concept application (Section 6.1.1) was also utilised for the conduction of a preliminary analysis. This preliminary analysis allowed for better-informed decisions regarding the selection of the required parameters for the execution of the methodology on a large scale.

In particular, the preliminary analysis allowed for the identification of the:

- most suitable parameters regarding the simplification process (i.e. search space reduction), as presented in Section 4.2.
- most suitable values of the optimisation-related parameters.

7.2.1 Parametrisation for Search Space Reduction

The following section presents the benefits as well as the implications arising from the search space reduction process which was described in the methodological Section 4.2. In more detail, the parameters to be defined after this sensitivity analysis are the:

- maximum total travel time of the modelled tours
- level of network simplification
- centrality measure to be used for network simplification
- level of the observed tour-types exclusion

The first parameter is expressed in total seconds of travel time and is based on the expected share of excluded observed tours due to the introduction of the corresponding cost threshold. This share can be estimated based on the available calibration distribution. In detail, two thresholds set at 5,400 and 4,500 seconds ensured that the percentage of excluded tours will not exceed 1% and 5% respectively. Additionally, the upper total travel time threshold was set at 9,900 seconds (2.75 hours) which is the maximum duration of all tours observed in the optimum search space (SS_O). The above-mentioned are summarised in Figure 7.1.

Figure 7.1 Distribution of the frequency of observed tours by total travel time.

The second parameter required by the search space reduction process is expressed based on the expected volume of excluded trips due to this filtering step. Since each node is connected to edges, weighted by the number of traversing trips, the exclusion of trips can be directly translated to the corresponding reduction of travel demand in the network. Based on this, nodes are iteratively excluded until a predefined travel demand threshold (i.e. number of trips originating from/ending to them) is met. For the purposes of the presented analysis, the four available levels of network simplification were defined at 0%, 5%, 10% and 20% of the excluded travel demand.

The third evaluation parameter corresponds to the identification of the most suitable centrality measure to be utilised for the simplification of the network. The evaluation is taking place among the centrality measures presented in Section 4.2.2 (i.e. Eigenvector, PageRank, Random-Walk-Betweenness and Subgraph-Centrality). The final parameter refers to the level of exclusion for tours belonging in unlikely tour-types. Since the optimal

search space (SS_O) is not a priori known, the likelihood of tour-types can be instead estimated based on observational data sources (e.g. travel surveys). For example, data obtained from the National Travel Survey (NTS) of UK (Department for Transport, 2017) indicate that tours beginning and ending during 00:00 and 07:00 constitute only the 0.13% between all the observed tours. Therefore, tours belonging in that tour-type could be potentially excluded from the solution without significant impact. Similarly, to the rest of the parameters, the tour-type likelihood one is also expressed via the expected losses in travel demand.

As it can be observed in Figure 7.2, the contribution of different observed tour-types to the total travel demand varies significantly. In this example, a tour-type is defined by its duration and the sequence of departure time periods. Despite the variability of the tour-type combinations, some prove to be significantly more frequent. In detail, the most frequent 10% of tour-types present in the observed dataset includes roughly 80% of the travel demand while the 25% and the 40% of tour-types describe the 90% and the 95% of the demand respectively. The remaining 60% explains only 5% of the demand, signifying that a large number of tour-types are considerably less likely to belong in the optimal search space (SS_O). Based on the previous, the levels of the excluded tour-types were defined at 0%, 5%, 10% and 20% of the excluded travel demand.

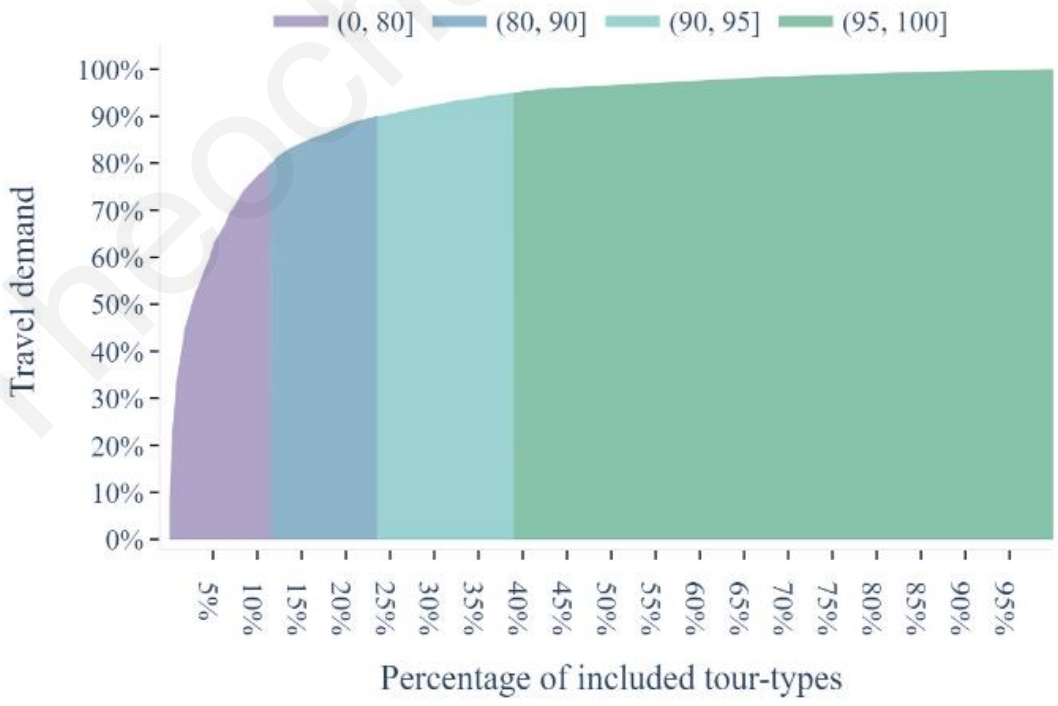

Figure 7.2 Percentage of included travel demand in relation to the percentage of the included tour-types.

The reduction of the search space based on the exclusion of unlikely tour-types can be considerable. This is due the disproportionality between the observability of a tour-type in real world and its identifiability in a graph. Figure 7.3 presents the share of the 25 most frequent departure time period sequences according to the calibration distribution. Each departure time period sequence is named after the time periods under which trips take place during the tour. As an example, "AM;PM", represents all the two-leg tour-types with the first departure in AM peak and second in the PM peak. As it can be observed, the observability (SS_O) and the identifiability (SS_C) of tour-types can differ significantly. As tours become more complex (e.g. include more departures) the available trips' combinations of that type increase exponentially while many of them become less likely to be observed in the optimal search space. Therefore, eliminating such disproportionate tour-types can lead to a drastic reduction of the search space without affecting considerably the quality of the final output. The evaluated levels of the excluded tour-types were defined at 0%, 5%, 10% and 20% of the excluded demand.

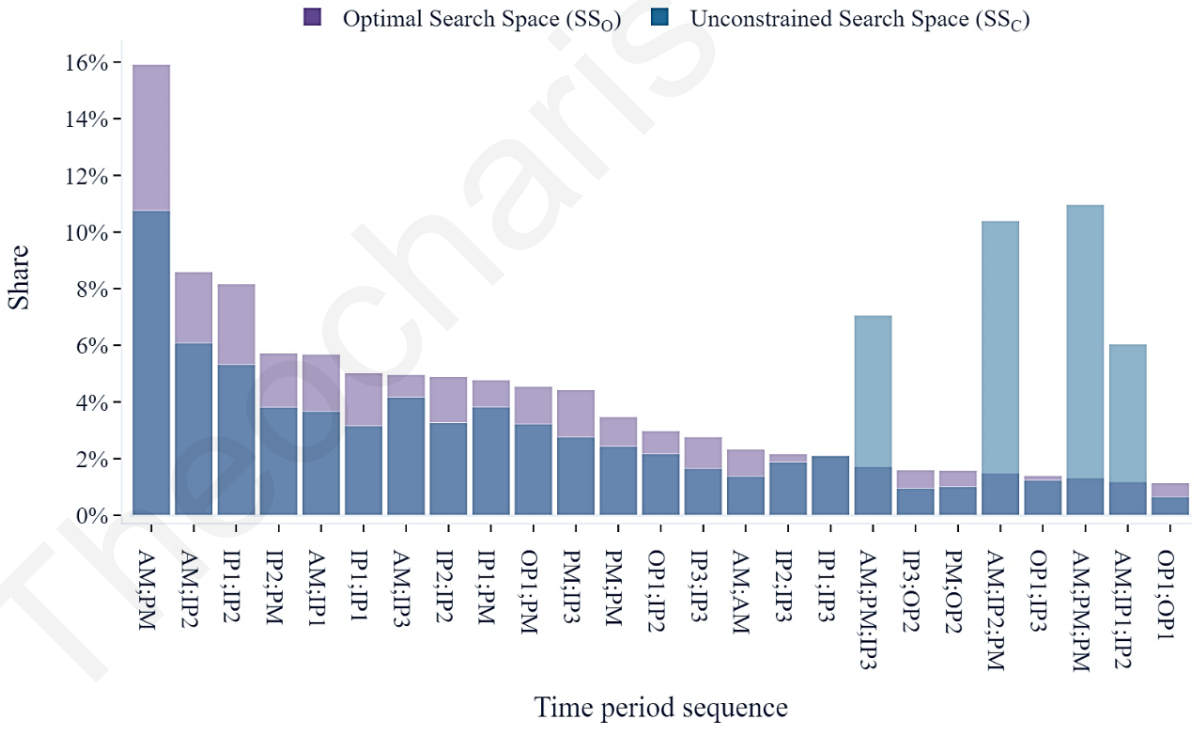

Figure 7.3 Number of identified tours in the optimal (SS_O) and the unconstrained (SS_C) search spaces.

To sum up, the parameters defining the exploration space are concisely presented in Table 7.1. Competing all the parameter combinations, resulted in an extensive array of 192 different configuration scenarios, allowing the meticulous sensitivity analysis of the parameters' effect on the solution.

Table 7.1 Summary of sensitivity analysis scenarios

Parameter	Search Space	Unit
Max total travel time	[0%, 1%, 5%]	% of excluded observed tours
Level of network simplification	[0%, 5%, 10%, 20%]	% of excluded travel demand
Level of excluded tour-groups	[0%, 5%, 10%, 20%]	% of excluded travel demand
Method of network simplification	[Eigenvector (EV), PageRank (PR), Random walk betweenness (RWB), Subgraph Centrality (SC)]	Centrality measure type

As it has been already mentioned, the sensitivity analysis allowed the evaluation of multiple sets of configuration parameters and facilitated the selection of the most appropriate values for the application of the methodology on the large-scale scenario. The following section presents the results of this sensitivity analysis.

7.2.1.1 Cost thresholds

A key decision affecting the bounds of the studied problem's search space is the maximum cost of tours to be identified. The selection of an appropriate threshold excluding high cost tours can drastically reduce the search space as well as the processing time requirements. Figure 7.4 presents the effect of three different travel time (i.e. cost) thresholds on the quality of the solution as well as the corresponding processing times. As it can be observed, the benefits of increasing the maximum allowed tour travel time are disproportionate to the requirements in processing time. Improving the percentage of unmatched tours from 4.5% to 1.5% requires 23% more time (4348s vs. 5366s) while a further reduction to 0% requires 47% more time compared to the 4500s threshold. Although, the here presented low running times (≈4,000s-6,000s), allow the completion of all cost related scenarios within reasonable time, this may not hold true for larger cases. Larger scenarios may require the implementation of a cost threshold lower than the maximum observed cost to be solved in reasonable time. However, the selection of the most appropriate cost threshold may vary across different applications, thus the experimentation with smaller instances of the problem are recommended prior to the application on the full problem. Based on the results obtained from this preliminary analysis, the maximum travel time for the large-scale scenario was set at 5400s (1% of excluded demand) in order to reduce the required processing time without considerably damaging the quality of the output.

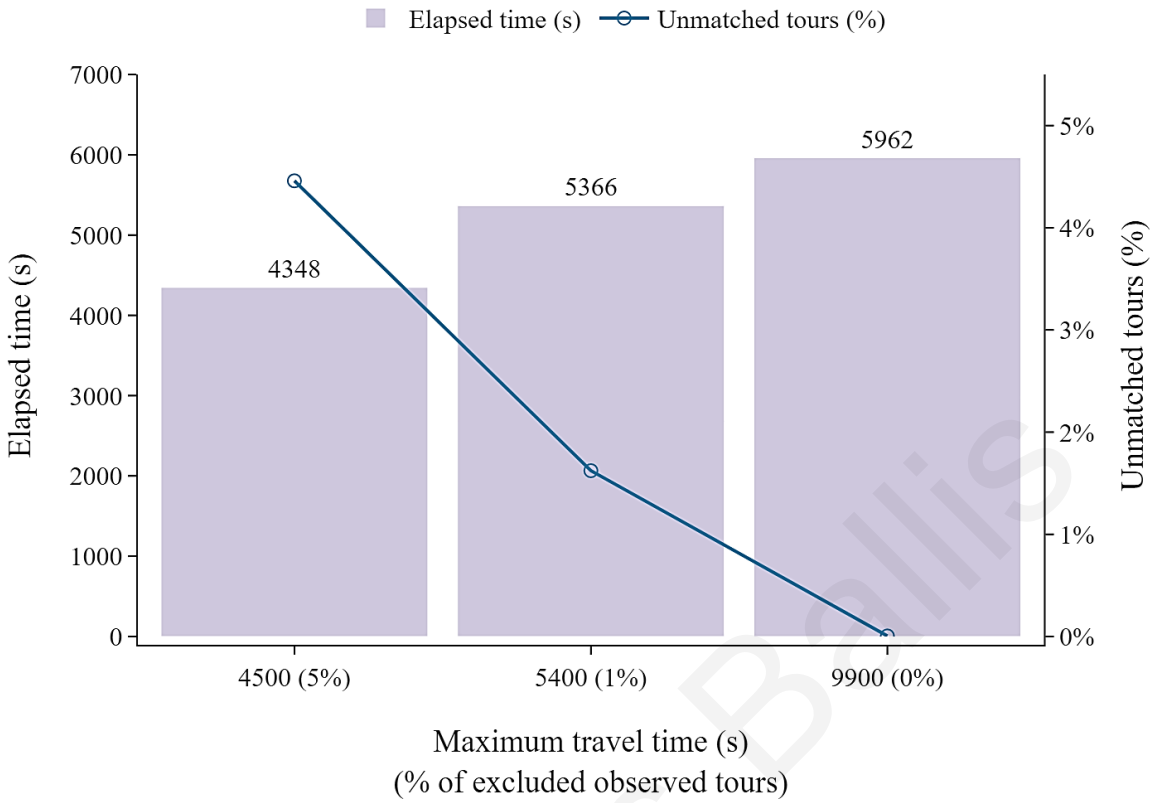

Figure 7.4 Processing time requirements and accuracy by maximum cost thresholds.

7.2.1.2 Effects of network simplification

As it has been already discussed, the reduction of the search space can be further bolstered by the process of network simplification. The following section delves in the presentation of the network simplification process as well as the potential caveats that may arise. The first presented metric concerns the number of eliminated nodes by simplification method and by level of simplification. As it can be seen in Figure 7.7, the number of eliminated nodes varies significantly across the evaluated methods, with Eigenvector (EV) and Subgraph-Centrality (SC) methods removing between 5% and 15% more nodes compared to the alternatives. On the contrary, the Random-Walk-Betweenness (RWB) method has systematically removed considerably less nodes (≈5%-15%) than the rest of the approaches. It should be also reminded, that since the simplification level is based on a maximum percentage of discarded demand, the comparison of the alternative methods across the same simplification level results to the same level of demand exclusion.

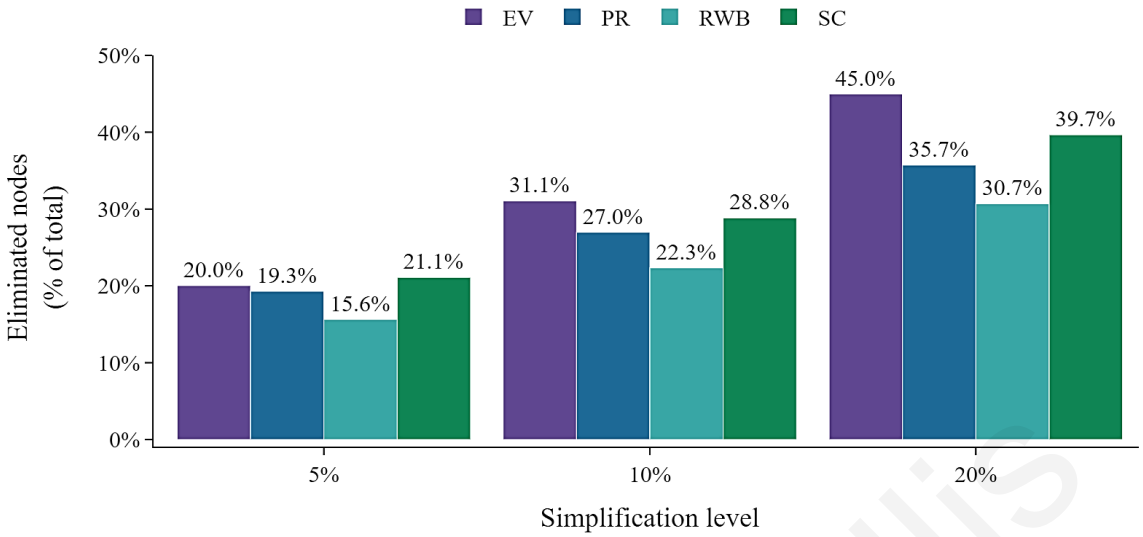

Figure 7.5 Number of eliminated nodes by method and level of simplification.

In addition to the previous results, Figure 7.6 presents the temporal distribution of (a) the eliminated nodes and (b) the corresponding reduction in travel demand. As it can be seen, nodes at the edges of the studied horizon (i.e. OP1 and OP3), are considerably more likely for elimination regardless of the simplification method and despite a few exceptions the trend is generally uniform. This can be explained by the low travel demand during the early morning (OP1) and the late night (OP3) periods as well as by the attribute of the evaluated centrality measures to prioritise the elimination of nodes with low demand.

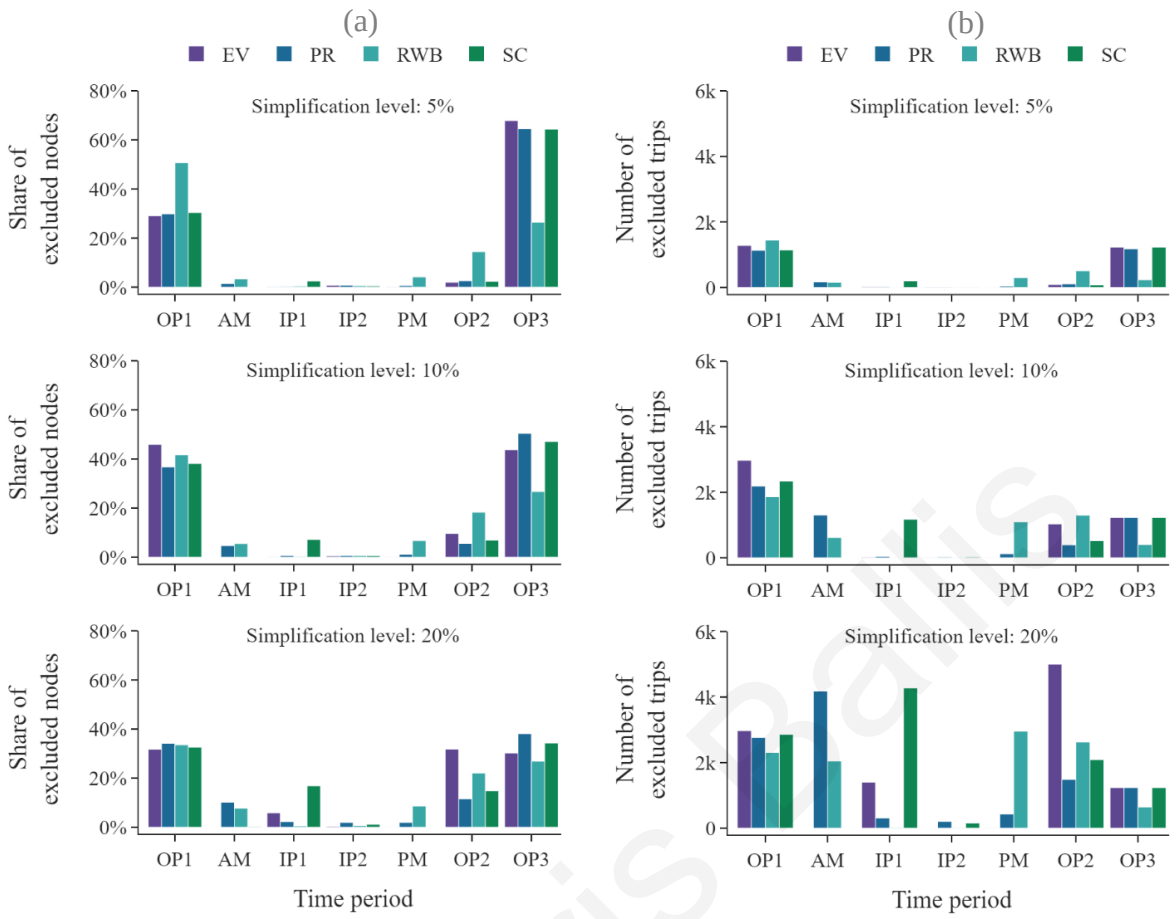

Figure 7.6 Distribution of eliminated nodes by level of simplification and time period.

The information presented in Figure 7.7, completes the analysis of the network simplification process by depicting its spatiotemporal dimension. At this stage it should be reminded that according to the hTVG representation, each zone is represented by t individual nodes, where t stands for the number of available time periods (Figure 7.7). Therefore, each zone can be eliminated up to t times depending on the level and the method of simplification. The location of the nodes in the figure corresponds to the X and Y coordinates of the zone's centroid while the colour tone represents the frequency of elimination across the time periods. As it can be noted, the patterns of simplification vary considerably between the available simplification methodologies. In accordance to the results presented in Figure 7.5, the EV method results in a more intense but uniformly distributed simplification across zones. On the other hand, PR, RWB and SC methods tend to eliminate less nodes while concentrating the simplification among fewer.

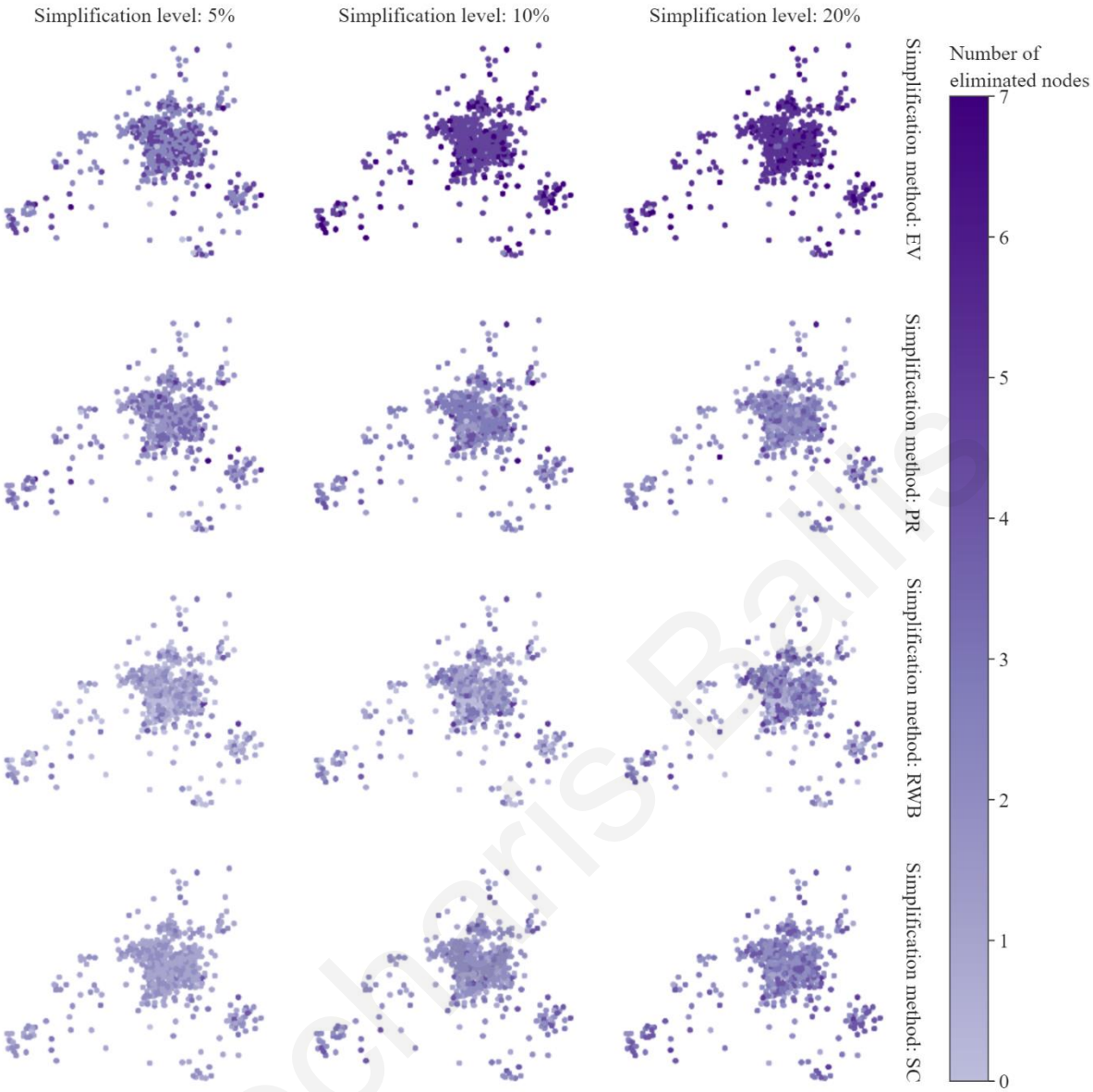

Figure 7.7 Density of simplification across the seven available time periods by level and method of simplification.

The previously presented results, although informative, they are not adequate to justify the most appropriate centrality measure for network simplification. The complete assessment requires the comparison of the simplification methods in relation to their effect on the expected accuracy of the methodology as well as their respective processing time and the memory size requirements. The relevant analysis begins with the reporting of the required processing time and the number of identified tours as a proxy for the memory requirements of each method. Although, the following results are presented in absolute figures, it is most useful to interpret them in a relative manner since both the processing time and the memory requirements depend on the utilised computing system. Figure 7.8 showcases the reduction of the required processing time incurred by the introduction of network simplification. For example, the EV method can progressively drop the processing time requirements from

roughly 600s to 400s when the level of simplification increases from 0% to 20% (tour-type exclusion at 0%). However, the process of calculating the centrality for each node requires time and that may counterbalance any subsequent benefits (e.g. RWB). Finally, the resemblance between the graphs highlights the negligible processing time impact of the search space reduction due to the likelihood filtering mechanism (Section 4.2.3.2).

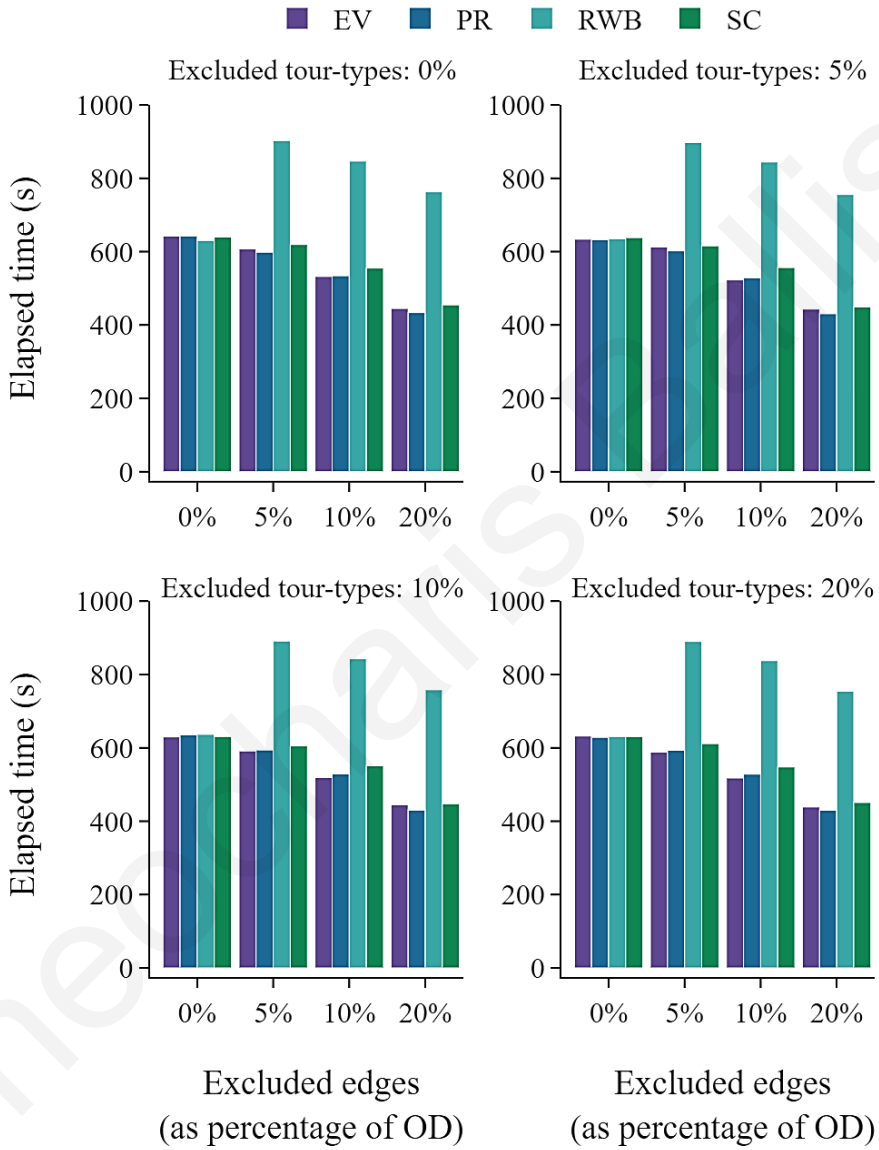

Figure 7.8 Processing time requirements by level and method of network simplification.

Despite the insignificant processing time of the tour-types exclusion mechanism, the corresponding implications regarding the memory requirements are considerable. As depicted in Figure 7.9 the exclusion of tours based on their likelihood can drastically constrain the number of identified tours. For example, while the initial figure of identified tours for the unconstrained scenario exceeds 250 thousand, the corresponding figure for the 20% level of exclusion barely reaches 50 thousand (80% reduction). Furthermore, Figure

7.8 supports the argument that the size of the search space is affected significantly more from the included tour-types rather than the level of network simplification. For instance, the increase of the network simplification level from 0% to 20% when no likelihood filter is applied (tour-types exclusion at 0%) drops the number of identified tours from 250 thousand to roughly 170 thousand (≈32% reduction), a figure significantly lower than the corresponding one for the reverse scenario (i.e. tour-types exclusion at 20% and network simplification level at 0%).

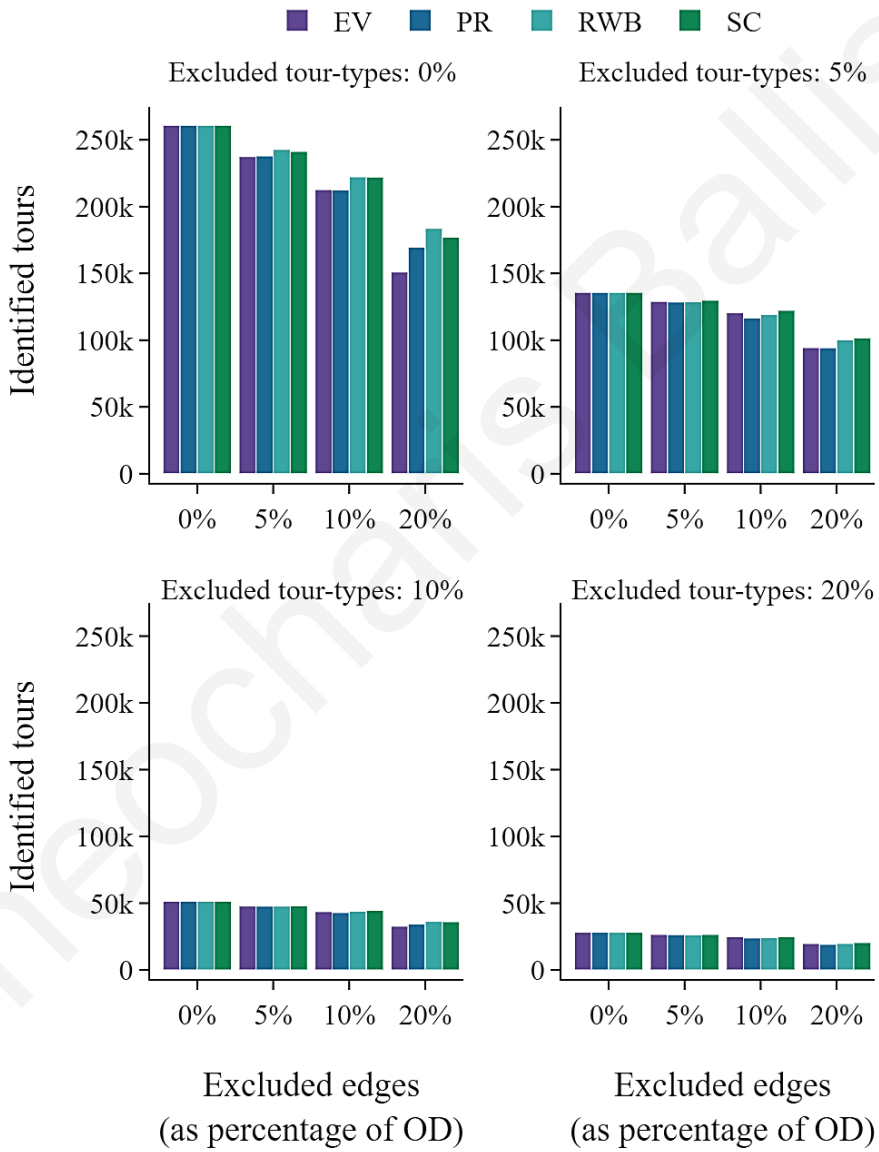

Figure 7.9 Total number of identified tours by level and method of network simplification.

The final and arguably most crucial metric related to the evaluation of the configuration parameters is the accuracy of the solution. More precisely, the accuracy is measured by the percentage of tours in SS_O (optimal search space) which were matched with a tour in the SS_R (reduced search space). The achieved level of accuracy for the combinations between the

levels of tour-type exclusion as well as the level and the method of simplification is presented in Figure 7.10. As it can be observed, the differences between the suggested simplification methods are subtle, with no method exhibiting considerable advantages in terms of accuracy. The key outcome of from this figure is the adverse effect of the increase of network's simplification level on the quality of the solution. For instance, retaining all tour-types (i.e. excluded tour-types: 0%) while applying the simplification of the network at 20% level, results in 40% of unmatched tours in SS_O. On the contrary, excluding 20% of the least frequent tour-types while retaining the integrity of the network (excluded edges: 0%) results in 19% of unmatched tours. This observation emphasises the abundance of non-required tours in the unconstrained search space (SS_C). However, network simplification can result in significant reductions of the required processing time, therefore the acceptance of a simplification level other than 0% may become unavoidable for larger scenarios. Even in cases where tours can be identified in reasonable time for low levels of network simplification, the possibly immense number of resulting tours may require the application of tour-type likelihood-based filters. Based on the results obtained for this preliminary analysis, a selection of 5% for both levels of network and tour-type exclusion can yield adequately accurate results (≈83% accuracy) while retaining the processing time and the memory requirements within reasonable levels.

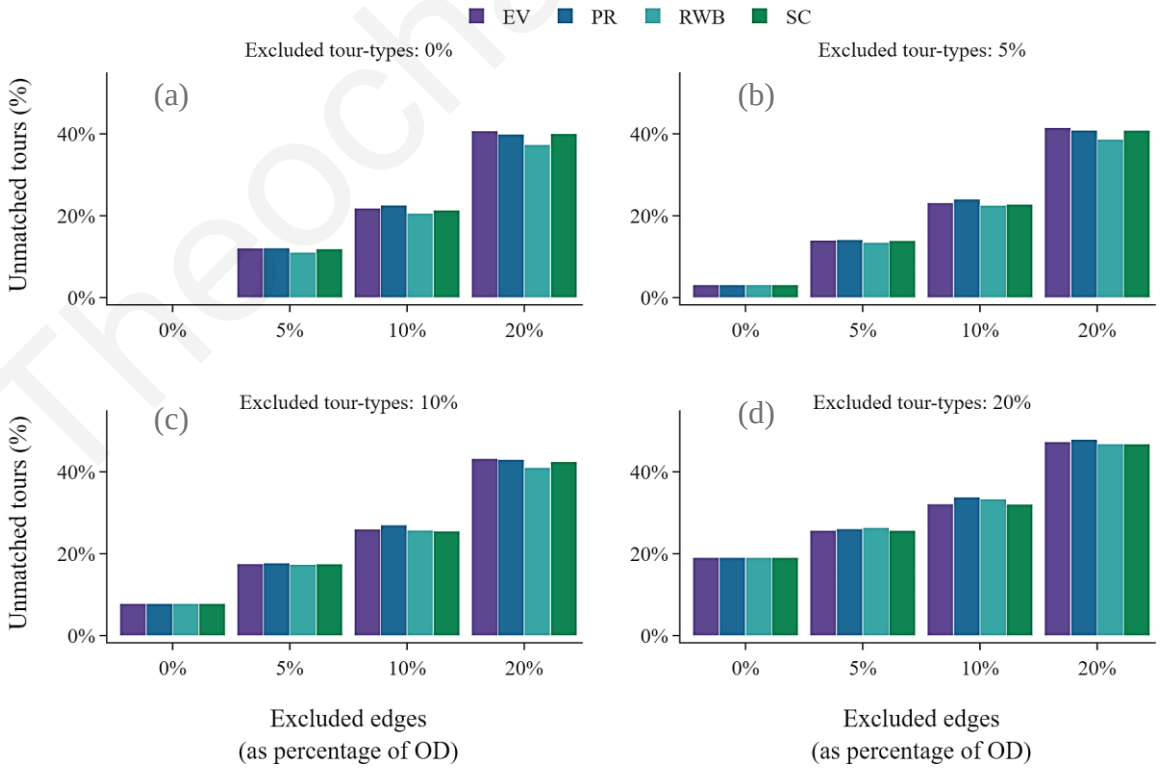

Figure 7.10 Accuracy by level and method of network simplification.

The next section delves with the identification of the most suitable parameters concerning the optimisation part of the methodology.

7.2.2 Parametrisation for Large-scale Optimisation

The following analysis attempts the identification of the most suitable parameters for the application of the Adaptive Sampling Simulated Annealing (ASSA) optimisation algorithm.

The two parameters which were explored were (a) the optimum replacement factor to enable the transition from a solution to the next as well as (b) the most suitable number of iterations to achieve convergence. In detail, the ASSA algorithm was evaluated for three different replacement factors (i.e. 1%, 2% and 5%) as well as for 100 steps and 500 steps, respectively. Based on the information retrieved from Figure 7.11, two hundred fifty steps proved adequate to allow the convergence of the ASSA algorithm regardless of the replacement factor. Moreover, smaller replacement factors (e.g. 1%) seem to be more appropriate for the purposes of the methodology. As an example, for the 500 steps case, the 1% replacement factor achieved more accurate results in less time compared to both the 2.5% and 5% cases (Figure 7.12). Therefore, the large-scale scenario was decided to be executed for 250 steps and with the replacement factor set at 1%.

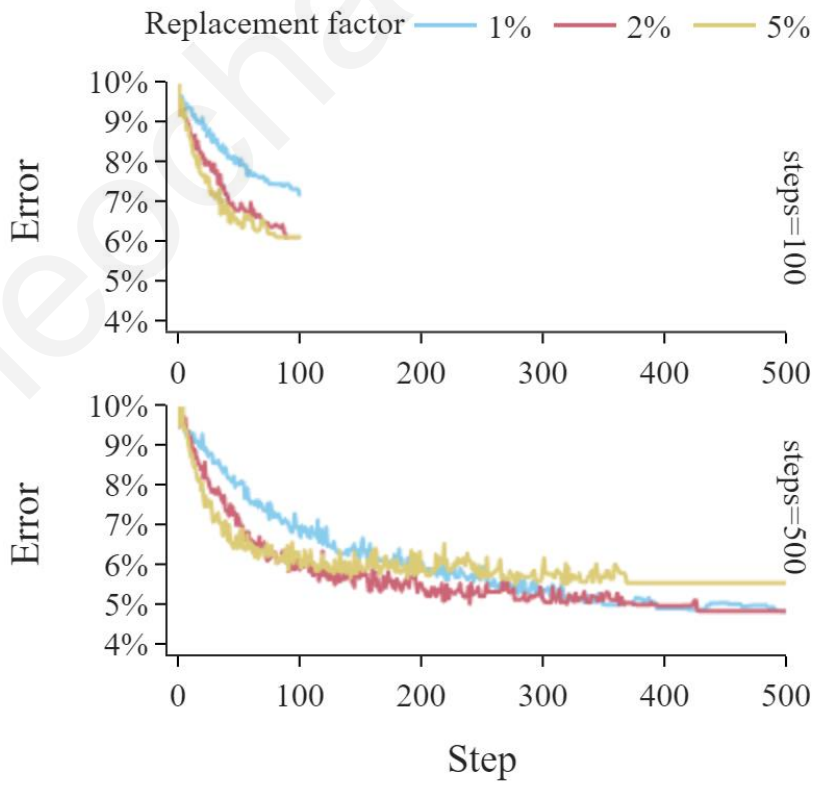

Figure 7.11 Convergence of ASSA algorithm by number of simulation steps, replacement factor and elapsed steps.

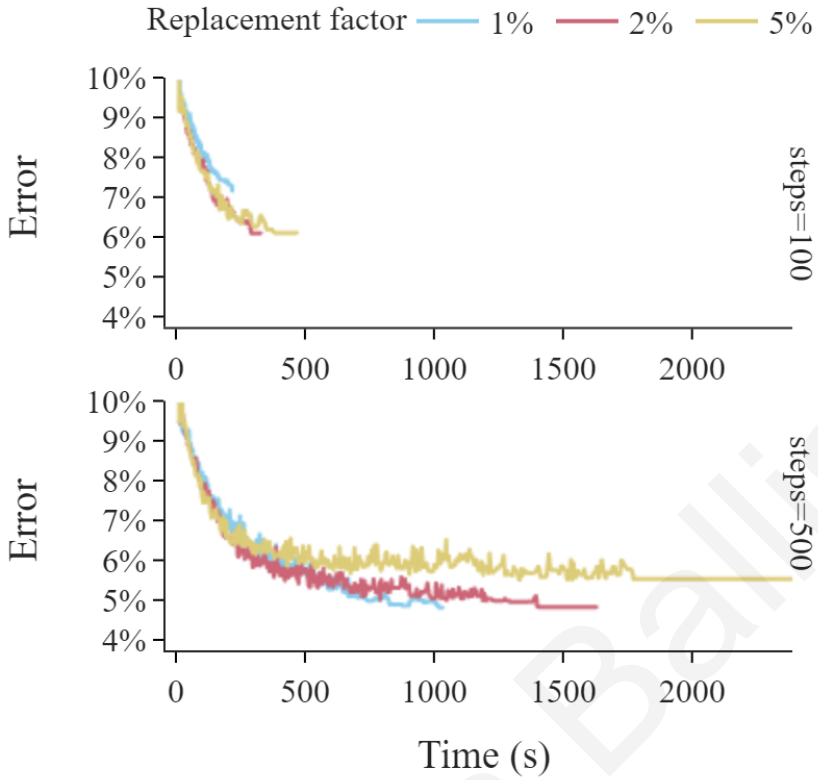

Figure 7.12 Convergence of ASSA algorithm by number of simulation steps, replacement factor and elapsed processing time.

The two previous sections delved in the identification of the most suitable parameters which allow the completion of the enumeration of tours within an hTVG in reasonable time, without significantly diminishing the accuracy of the output. The analysis showed that indeed the suggested simplification methodology can drastically reduce the problem domain without seriously damaging the quality of the output. However, the selection of the most appropriate simplification parameters depends highly on the characteristics of the input, therefore a preliminary analysis similar to the one presented above is strongly suggested for cases of large-scale input. Finally, it should be stated that a summary of the parameters deriving from the preliminary analysis can be found in Section 7.3.2 where the complete configuration of the large-scale scenario is presented.

7.3 Model Testing

The following section presents the required details for the application of the methodology on the large-scale case. The section begins with the presentation of the input, continues with the parameters' configuration, and concludes with the presentation of the results.

7.3.1 Input Data

The evaluation of the methodology's scalability was conducted based on input similar to the one described in Section 6.1.1 but of considerably larger size.

7.3.1.1 Observed tours

The same methodology used to build the input for the proof of concept case was used to synthesise 125,000 activity schedules (instead of 25,000) which once aggregated resulted in a set of fully realistic, multi period and purpose segmented ODs, suitable for the evaluation of the methodology at large scale.

7.3.1.2 Calibration distribution

The observed tours for the full scale-scenario were synthesised based on the distribution presented in Section 6.1.1.3. This distribution was also used to calibrate the solution deriving from the optimisation module; therefore, it is referred as the '*calibration distribution'*.

7.3.1.3 Observed OD matrices

The aggregation of the enclosed trips within the 125,000 observed tours resulted in 28 ODs, segmenting 268,315 trips across four trip purposes and seven time periods (Table 7.2). The resulting observed OD were subsequently inputted to the methodology for the identification of the individual tours recreating the observed travel demand patterns.

Table 7.2 Summary of observed ODs (large-scale scenario).

Trip-purpose	**OP1**	**AM**	**IP1**	**IP2**	**PM**	**OP2**	**OP3**	**Total**
HBW	3,065	13,850	7,924	9,130	13,184	5,189	1,133	53,475
HBO	11,323	50,132	27,016	33,718	49,215	19,165	4,726	195,295
NHBW	30	713	1,392	1,567	1,504	302	9	5,517
NHBO	76	1,764	3,311	4,002	3,986	675	34	13,848
Total	66,459	39,643	48,417	25,331	14,494	5,902	67,889	268,135

7.3.2 Configuration

The above-described observed ODs and calibration distribution were provided as input for the synthesis of the travel demand equivalent tours whose characteristics adhere as much as possible to the provided distribution. In contrast to the proof of concept case, the completion of the methodology required the introduction of constraints enabling the reduction of the problem's size to handleable limits. The selection of the appropriate constraining parameters

(Table 7.3) achieved the completion of the application in reasonable time without inflicting significant discount on the accuracy of the output.

Table 7.3 Parametrisation of the large-scale scenario.

Identification module

Parameter	Value	Description
Max tour length	4	Maximum length of tours in the observed dataset
Max total travel time	5400s	Maximum travel time for the identified tours
Level of network simplification	5%	Percentage of excluded travel demand
Level of excluded tour-types	5%	Percentage of excluded travel demand
Method of network simplification	PageRank (PR)	Centrality measure

Optimisation module

Parameter	Value	Description
Steps	250	Number of iterations
Max solution size	125,000	The maximum number of tours in the solution
Replication factor	1%	The number of replaced tours per iteration as percentage of the maximum solution size
Max temperature	5,000	Maximum temperature (calculated as: Max solution size * Replication factor * Max tour length)
Min temperature	10	Minimum temperature

Finally, the enforcement of the observed tours high-level characteristics to the output were achieved through the provision of the calibration distribution presented earlier (Section 7.3.1.2) as input to the ASSA optimisation algorithm.

7.3.3 Results

The application of the suggested methodology to the previously presented input produced a particularly large number of candidate tours (22 million) able to be formed in the hTVG deriving from the observed ODs. Among these 22 million candidates, the optimisation algorithm managed to identify a combination of realistic candidate tours which utilises almost 90% of the observed travel demand. In terms of performance, the overall processing time required for the conversion of 28 multi-period, purpose segmented ODs reached 43 hours with the majority of the processing time being allocated to the identification module (40 hours).

In the examined large-scale scenario, the modelled area consists of 470 different zones with an average number of 46,800 originating tours per zone, therefore it becomes apparent that

the total number of candidate tours can grow very rapidly. As previously mentioned, the total number of returned tours exceeds 22 million with the majority of zones producing up to half a million candidate tours, although some few exceptions produce significantly more (e.g. 3-4 million). The zones producing significantly more tours than the rest were identified as the centre of the city as well as areas of particular interest such as university campuses. The above mentioned are depicted in Figure 7.13 where the spatial distribution of candidate tours as well as the corresponding histogram are presented. In addition, Figure 7.14 depicts the observed tours over all the possible ones which could have originated from a single zone, based on the available trips in the observed ODs. The figure emphasises the complexity of the studied combinatorial problem by depicting the disproportionality between the number of candidate tours and the number of the observed ones.

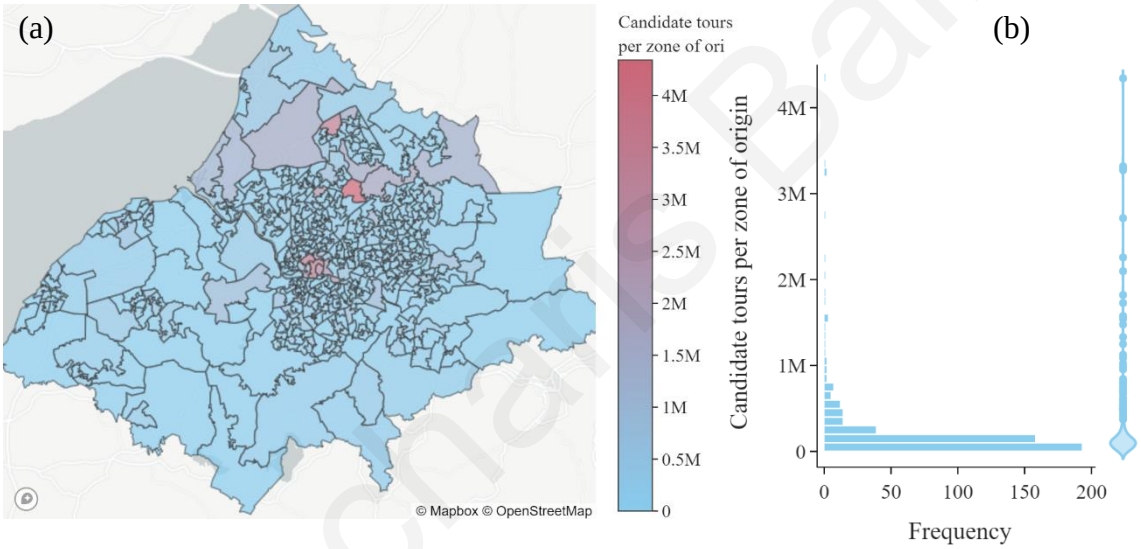

Figure 7.13 (a) Spatial distribution of candidate tours per zone of origin and (b) the respective histogram.

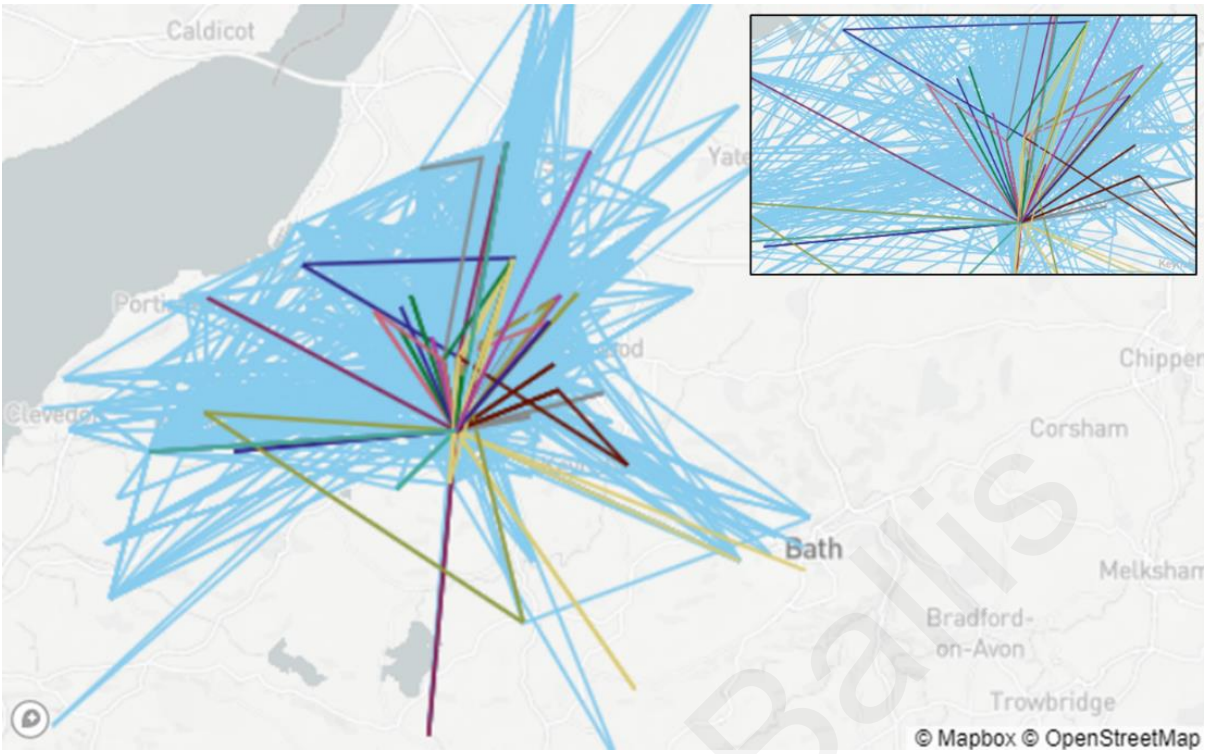

Figure 7.14 Observed tours over the candidate ones (Observed tours coloured in blue).

The next section delves in the evaluation of the suggested methodology for the conversion of ODs to individual tours. Additionally, the application of the ASSA algorithm to the large-scale scenario allowed the meticulous evaluation of the newly suggested approach on a particularly complex and large combinatorial problem. The superiority of ASSA is verified through the comparison of the former with the standard Simulated Annealing (SA) approach.

7.4 Evaluation

The next section has a twofold role. Firstly, it is used to verify the scalability of the proposed methodological framework. Secondly, the large-scale combinatorial problem of converting ODs to tours is also utilised to assess the performance and the efficiency of the ASSA algorithm. The evaluation of the ASSA algorithm has been strengthened by comparing it against the widely used standard SA algorithm.

7.4.1 Comparison of ODs

The results regarding accuracy and convergence of the large-scale scenario agree with the corresponding ones for the proof of concept case. In detail, the methodology has managed to identify a reasonably accurate solution (14.1% error), closely replicating the observed ODs (Table 7.4 and Table 7.5). The most notable discrepancies between the observed and the modelled ODs are presented for the low-demand ODs. It should be also reminded that according to the previously presented preliminary analysis, the maximum accuracy of the

methodology under the current levels of simplification (Section 7.3.2) is estimated at around 15%. Therefore, the methodology has produced a solution very close to the optimum.

Table 7.4 Absolute difference between the observed and modelled ODs for the large-scale scenario.

Purpose	OP1	AM	IP1	IP2	IP3	PM	OP2	Total
HBW	1,034	1,738	708	888	703	1,672	454	7,197
HBO	3,661	5,435	2,370	3,312	2,549	5,986	1,996	25,309
NHBW	30	219	403	402	146	488	9	1,697
NHBO	76	485	815	878	309	1,052	34	3,649
Total	4,801	7,877	4,296	5,480	3,707	9,198	2,493	37,852

Table 7.5 Percentage difference between the observed and modelled ODs for the large-scale scenario.

Purpose	OP1	AM	IP1	IP2	IP3	PM	OP2	Total
HBW	33.7%	12.5%	8.9%	9.7%	13.5%	12.7%	40.1%	13.5%
HBO	32.3%	10.8%	8.8%	9.8%	13.3%	12.2%	42.2%	13.0%
NHBW	100.0%	30.7%	29.0%	25.7%	48.3%	32.4%	100.0%	30.8%
NHBO	100.0%	27.5%	24.6%	21.9%	45.8%	26.4%	100.0%	26.4%
Total	33.1%	11.9%	10.8%	11.3%	14.6%	13.5%	42.2%	14.1%

7.4.2 Adherence to the Calibration Information

The first part of the assessment concluded that the majority of trips in the large-scale input ODs were successfully incorporated into tours in reasonable time. Nonetheless, the quality of these schedules from a travel behaviour point of view must be also verified. In contrast to the proof of concept application, the large-scale implementation required the application of a heuristic optimisation algorithm for the optimisation part of the methodology. The metaheuristics formulation (Section 3.5.2) does not allow the enforcement of the available calibration information as strict constraints. However, the developed ASSA algorithm, if provided with a calibration distribution, it favours the identification of solutions with distributions similar to the calibration one. The efficiency of ASSA on projecting the calibration data on the modelled solution is evaluated in the following section.

The scatter plot analysis presented in Figure 7.15 verifies that the application of the ASSA algorithm produced a solution aligning closely to the available calibration data. Each point in the figure represents the frequency of tours belonging to the 386 available tour-types. As it can be observed the R^2 value is very close to 1 (0.997) while the same applies for slope of

the regression fit (1.08), therefore it can be evidently claimed that the frequencies of tour-types between the observed and the modelled tours are very similar.

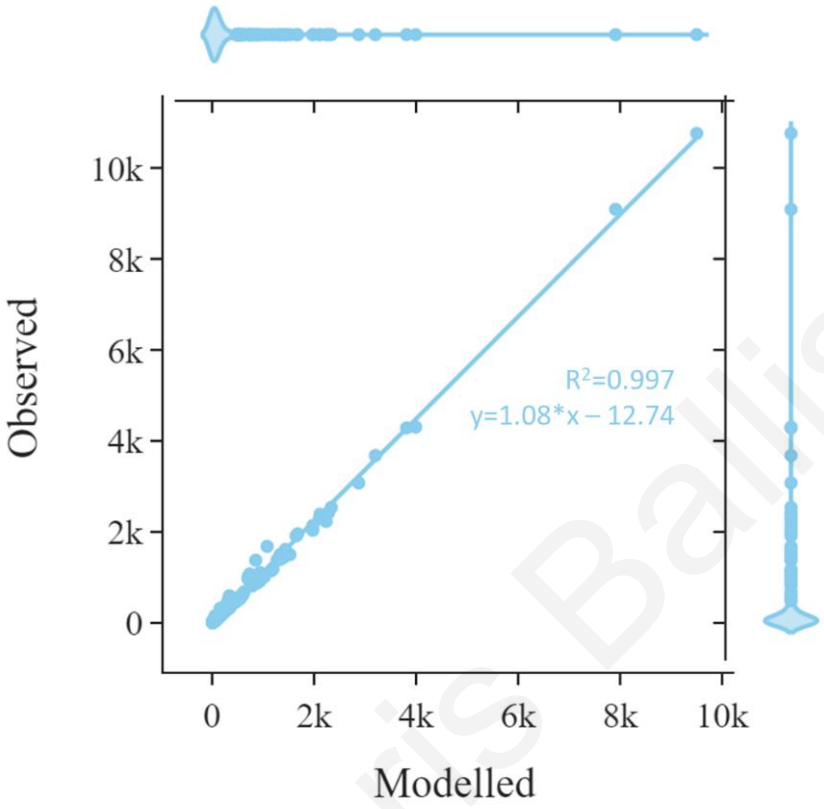

Figure 7.15 Scatter plot analysis between the observed and the modelled tour-types.

The evaluation continues with the comparison between the observed and the modelled distributions for the 50 most frequent tour-types which accrue for 75% of the total travel demand (Figure 7.16). As it can be seen, the coincidence between the observed and the modelled solutions is generally well respected, but the ASSA algorithm results in a significantly closer resemblance compared to the standard SA algorithm (especially for the two most frequent tour-types). The ability of the methodology to project the calibration distribution on the result is further evaluated in a following section (Section 7.5.3).

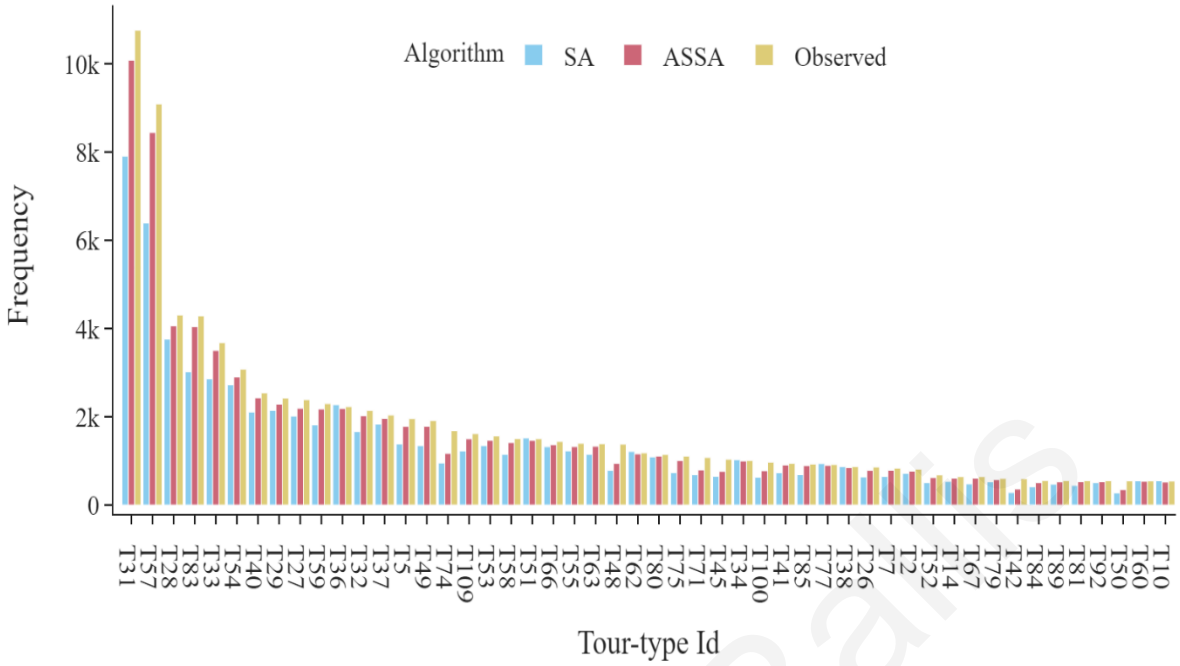

Figure 7.16 Comparison between the SA, ASSA, and the calibration distribution for the 50 most frequent tour-types.

The evaluation of the quality of the modelled solution is finally verified from a spatial perspective. Figure 7.17 depicts the number of modelled (ASSA solution) and observed tours traversing through each of the zones in the covered area. As it can be noted, the discrepancy is distributed in accordance to the volume of tours through each zone and no systematic bias is evident.

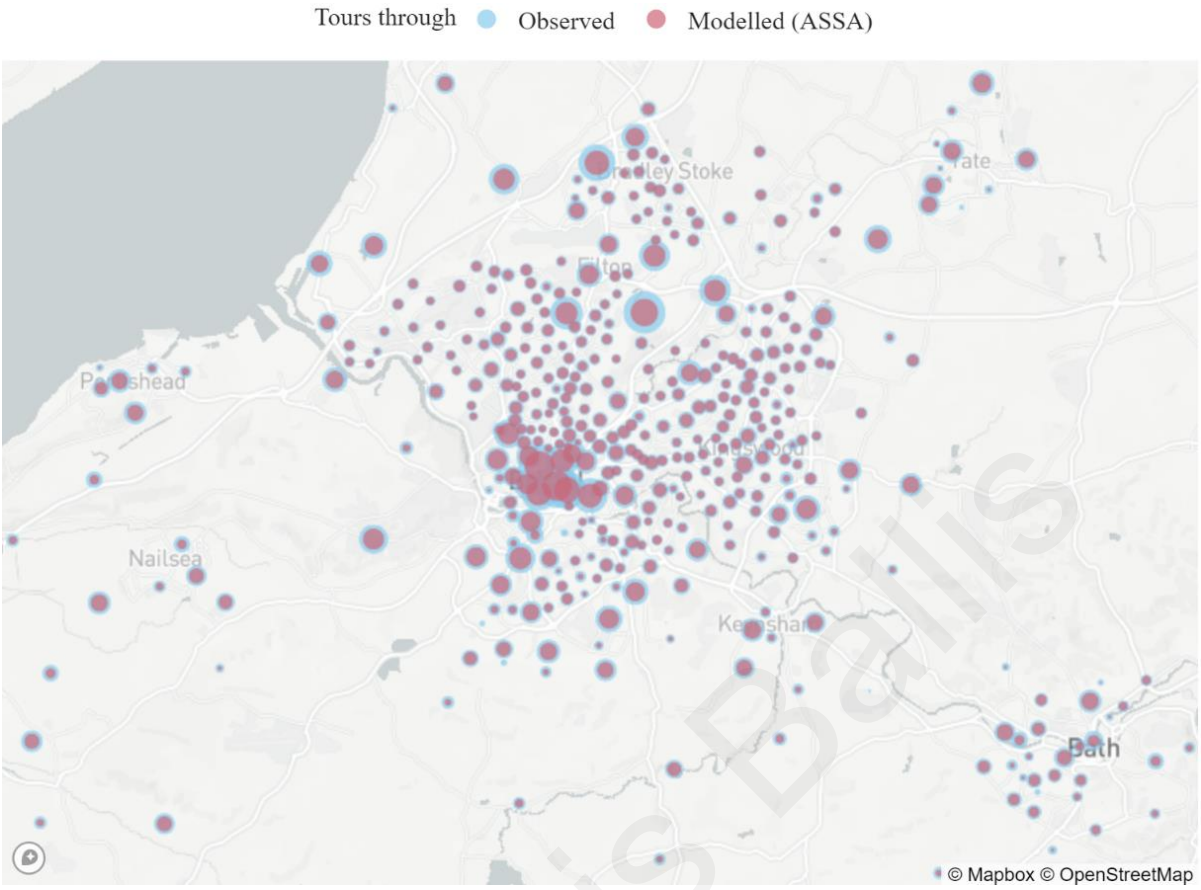

Figure 7.17 Number of tours traversing through each zone (observed vs ASSA derived solution)

Based on the previous results it can be claimed that the methodology can indeed produce accurate results, even for an excessively large-scale scenario as the evaluated one. However, the methodology should be also assessed in terms of the efficiency and the processing time requirements for its completion.

7.4.3 Efficiency and Processing Time Requirements

Due to the combinatorial nature of the studied problem, the processing time required for the application of the suggested methodology increases with the increase of the problem's size. Although, the effect is insignificant for the Graph-generation and the Activity-scheduling modules (additional processing time in the order of tens of seconds), the Identification module as well as the Optimisation module require considerably more time in comparison to the proof of concept application.

Five-folding the size of the problem (from 25,000 initial tours to 125,000) increased the required processing time for the identification of candidate tours from 3 to 40 hours using an Intel® Xeon CPU powered computer with 32GB of available RAM. Although not prohibiting for real-world applications, stricter simplification parameters can reduce the

above-mentioned processing time, at the expense though of accuracy (Section 7.2.1.2). In addition, it should be also reminded that the identification module has been designed in a fully parallelisable manner, therefore computing systems with multiple processing cores can significantly boost performance.

The considerable increase of the candidate tours (compared to the proof of concept case) inevitably complicated the optimisation process. Attempts to apply exact mathematical programming approaches (e.g. branch-and-bound) were proven unsuccessful since the problem was overly large to be instantiated using the available computational system, despite its relatively high-end specifications. However, the previously presented ASSA algorithm (Section 5.4.2) proved very efficient at addressing the problem using only a fraction of the available computing resources. In detail, the optimisation routine converged to an adequately accurate result in around three hours, setting the total required time for the completion of the overall methodology at approximately 43 hours.

In summary, the overall processing time required for the conversion of 28 multi-period, purpose segmented ODs of 260 thousand trips reached 43 hours with most of the processing time being allocated to the identification module (40 hours).

7.5 Assessment of the ASSA Algorithm

As it has been already mentioned, the large-scale scenario was also used to verify the superiority of ASSA over the standard SA approach. The next section completes this evaluation by comparing the results of the Identification module when the ASSA and the standards SA algorithms are implemented.

7.5.1 Preliminary Evaluation

The developed for the purposes of this Ph.D. Thesis ASSA optimisation algorithm is mainly directed for addressing large-scale combinatorial optimisation problems. Therefore, the evaluation of ASSA was assessed over a large-scale scenario. For the completeness of the presentation, the evaluation of ASSA on a small-scale scenario, able to be also addressed by standard analytical optimisation approaches (e.g. branch-and-bound) is also included. In particular, ASSA was firstly evaluated for the proof of concept application presented in Chapter 6. The novel algorithm was evaluated against the standard SA as well as against the widely applied branch-and-bound implementation of CPLEX by IBM (IBM, 2020). As it can be observed in Figure 7.18, branch-and-bound outperforms the SA-based alternatives, especially in terms of the accuracy of the produced solution. The branch-and-bound (B&B)

alternative achieves a near optimum solution (error < 1%) in practically the same time required for SA and ASSA to converge. However, it should be reminded that despite the accuracy of exact optimisation methods, such as B&B, their applicability and efficiency can be significantly deteriorated in cases of particularly large combinatorial problems. Therefore, metaheuristics can provide an alternative for large scale cases. With regards to the SA-based algorithms, ASSA proves superior to the standard version of SA since it has managed to achieve a significantly more accurate solution (4.8% error vs. 9.2%). Not surprisingly, the ASSA algorithm requires more processing time (≈ +25%) to complete the same number of iterations (i.e. steps) compared to the standard SA, due to the extra calculations executed by the adaptive sampling mechanism. However, the increased processing time is counterbalanced by faster convergence, resulting to more accurate solutions in fewer iterations (Figure 7.19)

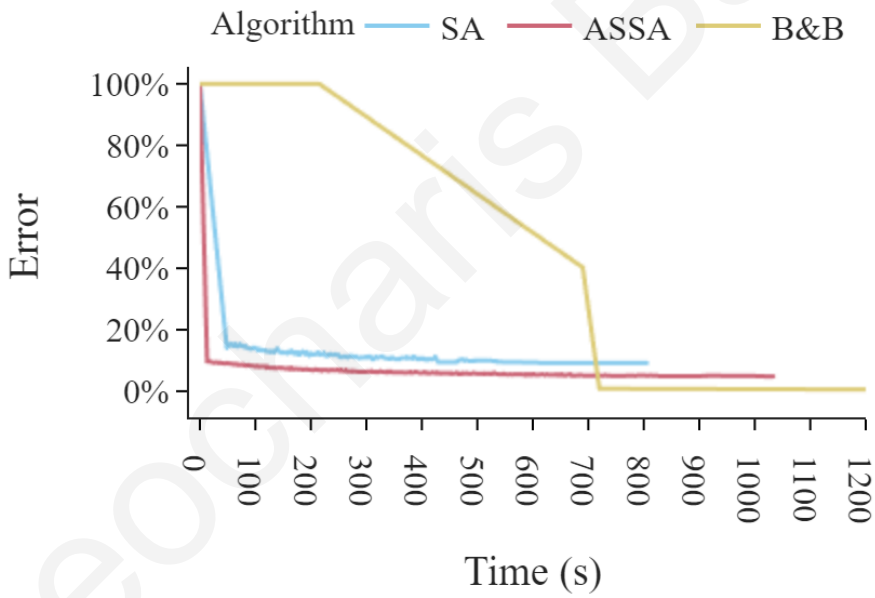

Figure 7.18 Evaluation between the branch-and-bound (B&B), SA and ASSA optimisation algorithms in terms of processing time and accuracy.

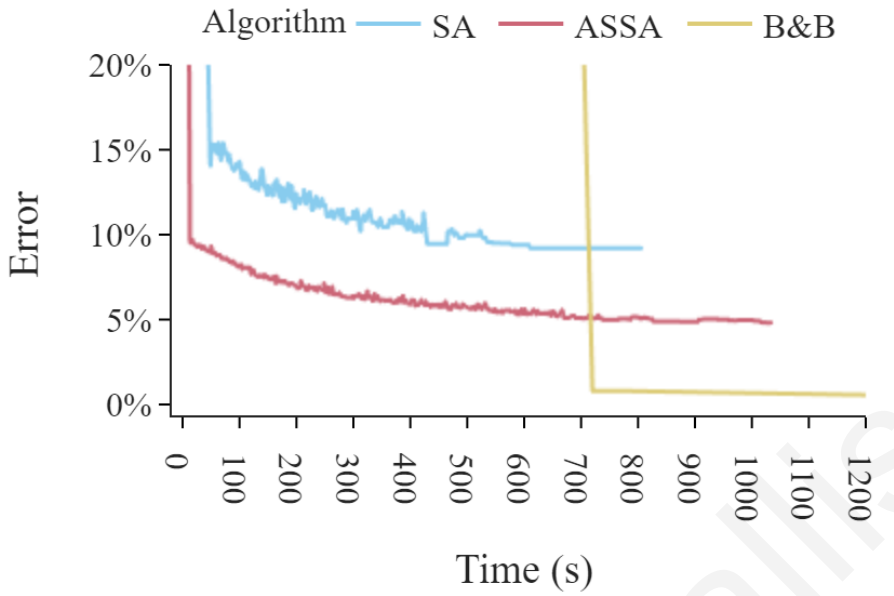

Figure 7.19 Magnification of the convergence area between the B&B, SA, and ASSA optimisation algorithms.

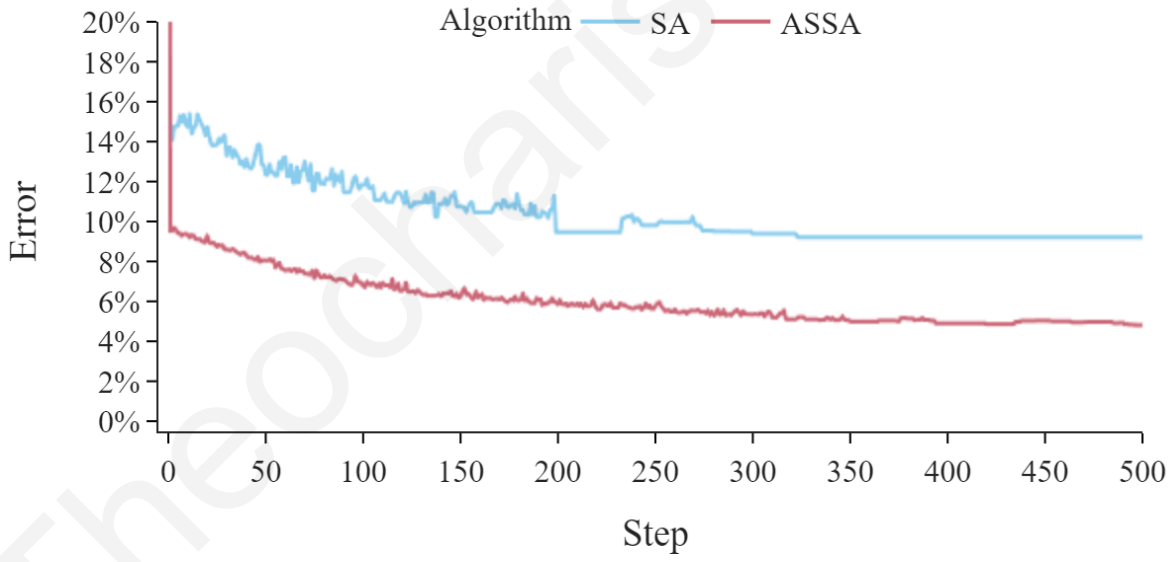

Figure 7.20 Evaluation of the SA and ASSA algorithms for the preliminary analysis (iterations vs accuracy).

7.5.2 Convergence and Efficiency

The following section provides evidence regarding the efficiency of the suggested ASSA algorithm for the addressing of excessively large combinatorial problems. In detail, ASSA produced a solution with 14.1% error in 40 thousand seconds (≈ 11 hours) but similarly accurate solutions (e.g. 15% error) can be obtained much faster (e.g. under 4 hours). This indicates that ASSA (as many other metaheuristics) can quickly reach a good approximate

solution but faces difficulties in fine-tuning the quality of the solution even with increasing processing time. As it can be observed in Figure 7.21 and Figure 7.22, ASSA clearly outperforms standard SA both in terms of accuracy and efficiency. The ASSA algorithm converges significantly faster than the alternative SA algorithm while at the same time produces a considerably more accurate solution (≈14% accuracy vs ≈21%).

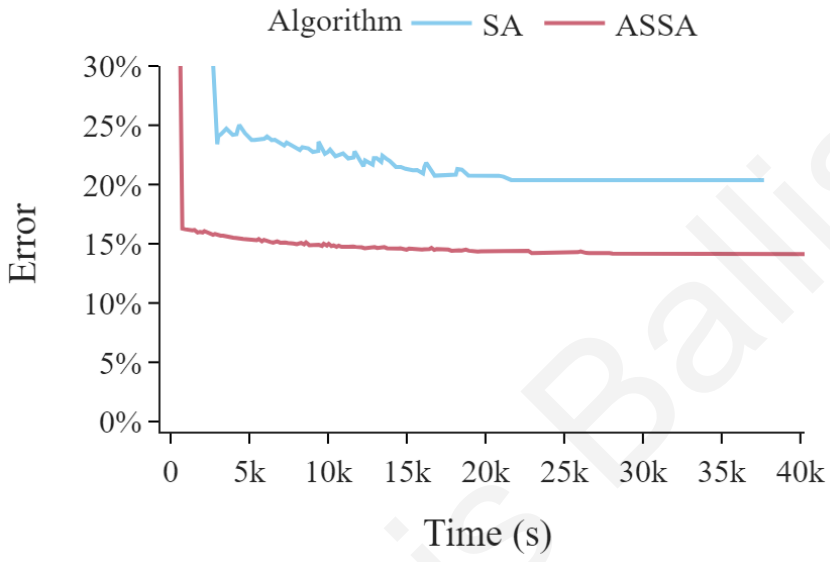

Figure 7.21 Evaluation of the SA and ASSA algorithms for the large-scale scenario (processing time vs accuracy).

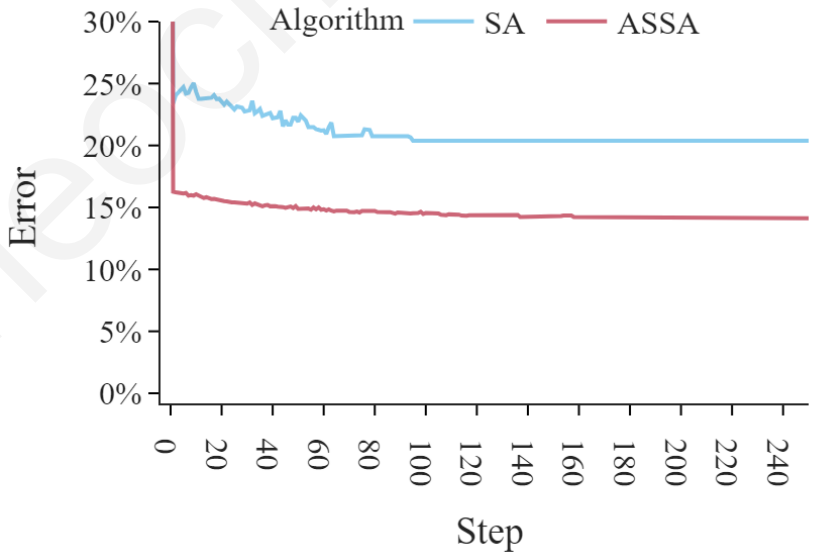

Figure 7.22 Evaluation of the SA and ASSA algorithms for the large-scale scenario (iterations vs accuracy).

The previous section verified the superiority of ASSA over standard SA in terms of accuracy and efficiency. The next section emphasises on the additional accuracy benefits obtained by the application of ASSA instead of SA under the presence of calibration information.

7.5.3 Adherence to the Calibration Information

As it has been already mentioned in Section 3.5, the utilised metaheuristic approach for the solution of the combinatorial problem in hand, does not strictly enforce the calibration distribution on the output but instead utilises the distribution to appropriately guide the optimisation process. Therefore, the accuracy of enforcing the calibrating distribution onto the identified solution using the ASSA algorithm must be thoroughly evaluated. In order to emphasise the benefits obtained from the application of the Adaptive Sampling mechanism described in Section 5.4.2, the optimisation results obtained from the standard Simulated Annealing (SA) method are also presented. In particular, Figure 7.23 presents the scatter plot comparison between the frequency of the observed and the modelled tour-types for the SA and ASSA algorithms, respectively. The depicted scatter plots (Figure 7.23) as well as the accompanying linear regression summaries (Table 7.6) showcase the ability of the methodology to effectively project the calibration distribution onto the output. As it can be noted, the ASSA algorithm accurately achieves the alignment between the resulting distribution of tour-types and the calibrating one. In contrast to the standard SA, the ASSA algorithm enhances significantly the matching for the low-frequency tour-types (Figure 7.24) which are underrepresented in the SA solution.

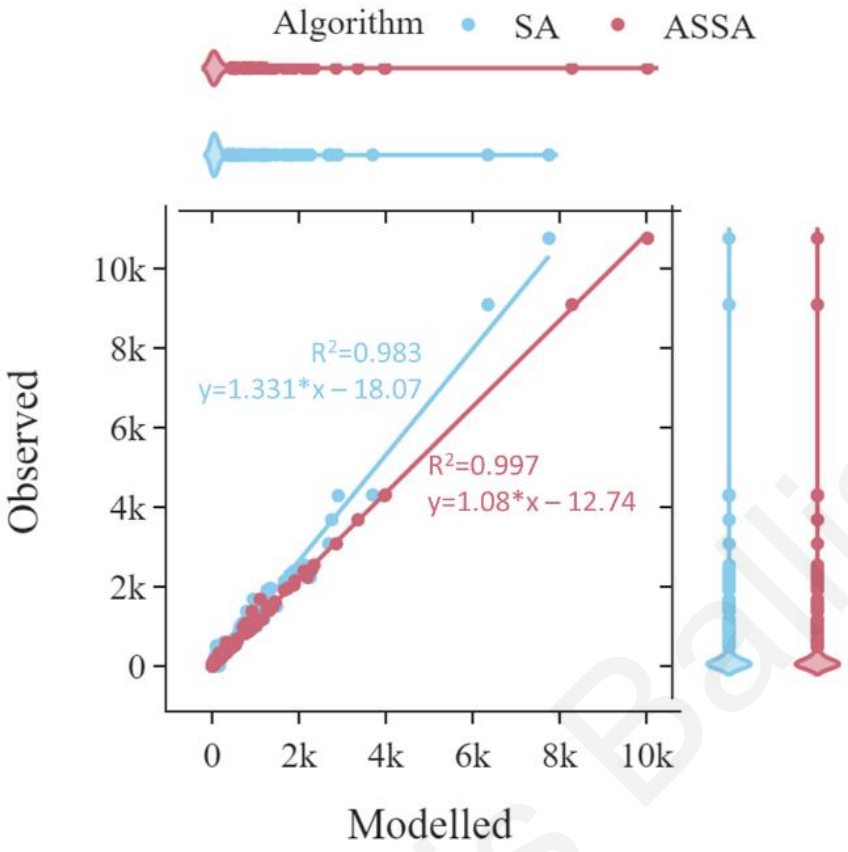

Figure 7.23 Comparison between the resulting and the calibration distribution.

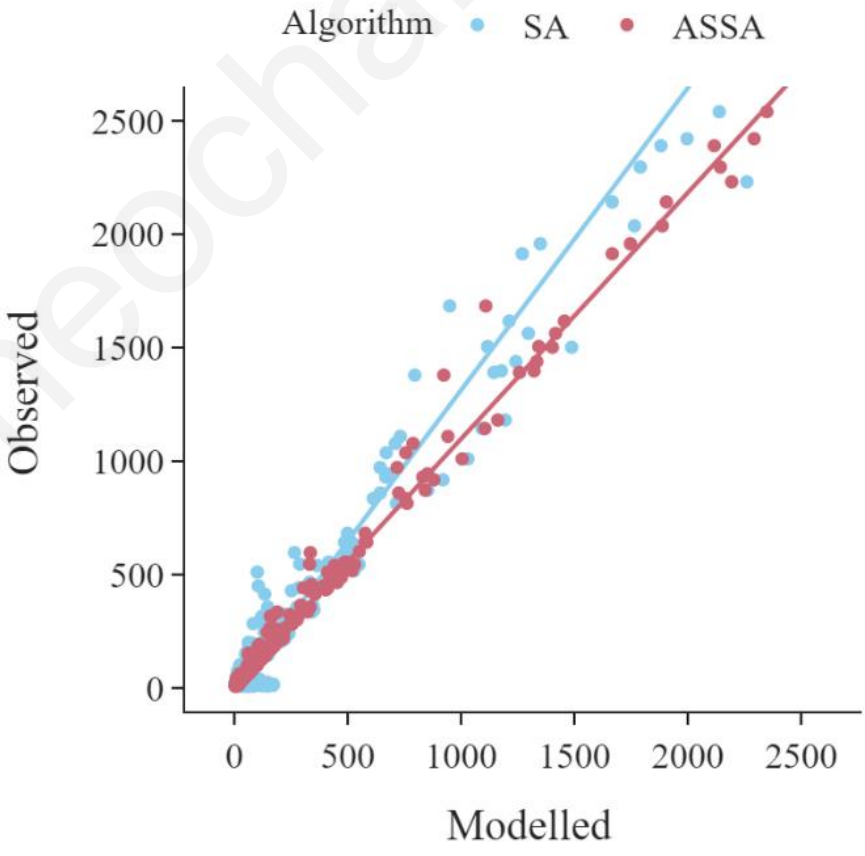

Figure 7.24 Magnification of the high-density area (0-2500 trips).

Table 7.6 Results of linear regression for the SA and ASSA algorithms.

Algorithm	**Equation**	**R^2**
SA	y = 1.33x - 18.07	0.983
ASSA	y = 1.08x - 12.74	0.997

The comparison between the expected and the resulting distributions for the SA and ASSA algorithms is also verified in Figure 7.25. As it can be observed, ASSA has considerably outperformed SA, especially at the edges of the distribution (presented in more detail in Figure 7.26 and Figure 7.27). Contrary to ASSA, the standard SA algorithm has produced a solution which underrepresents frequent tour-types and overrepresents rare ones (SA frequencies are larger than the observed at the tail of the distribution).

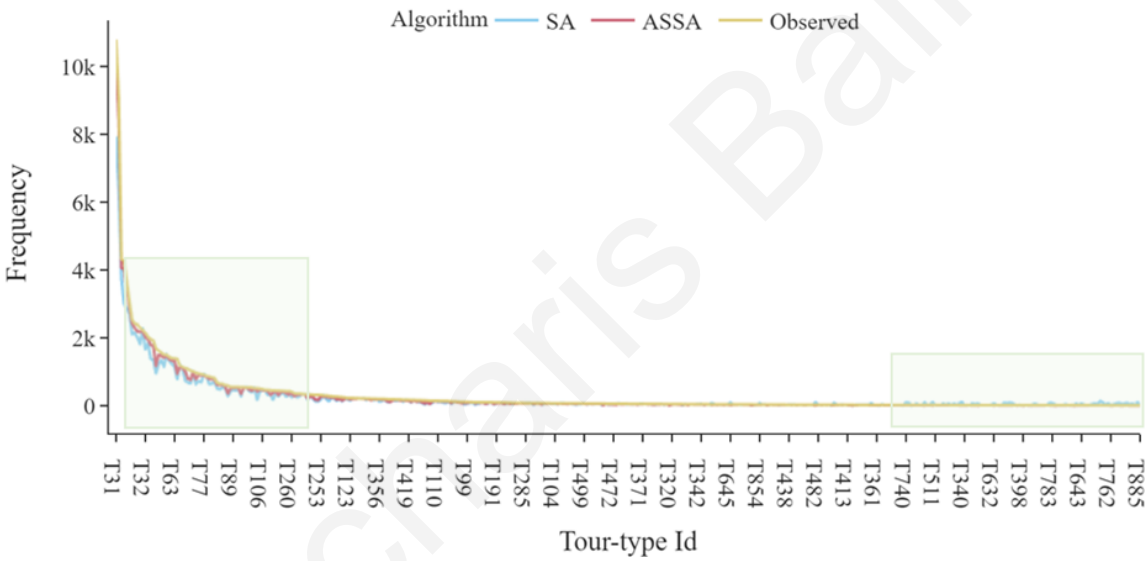

Figure 7.25 Comparison between the SA and ASSA resulting distributions and the calibrating one.

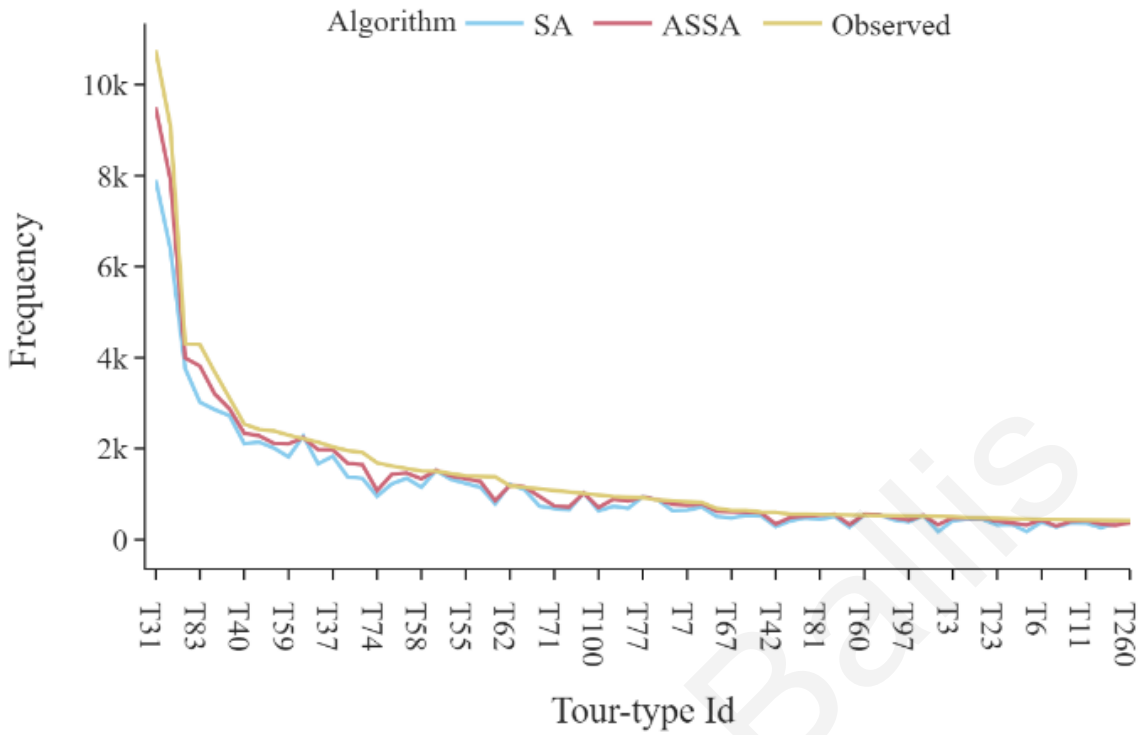

Figure 7.26 Comparison between the SA and ASSA resulting distributions and the calibrating one for the 20 most frequent tour-types.

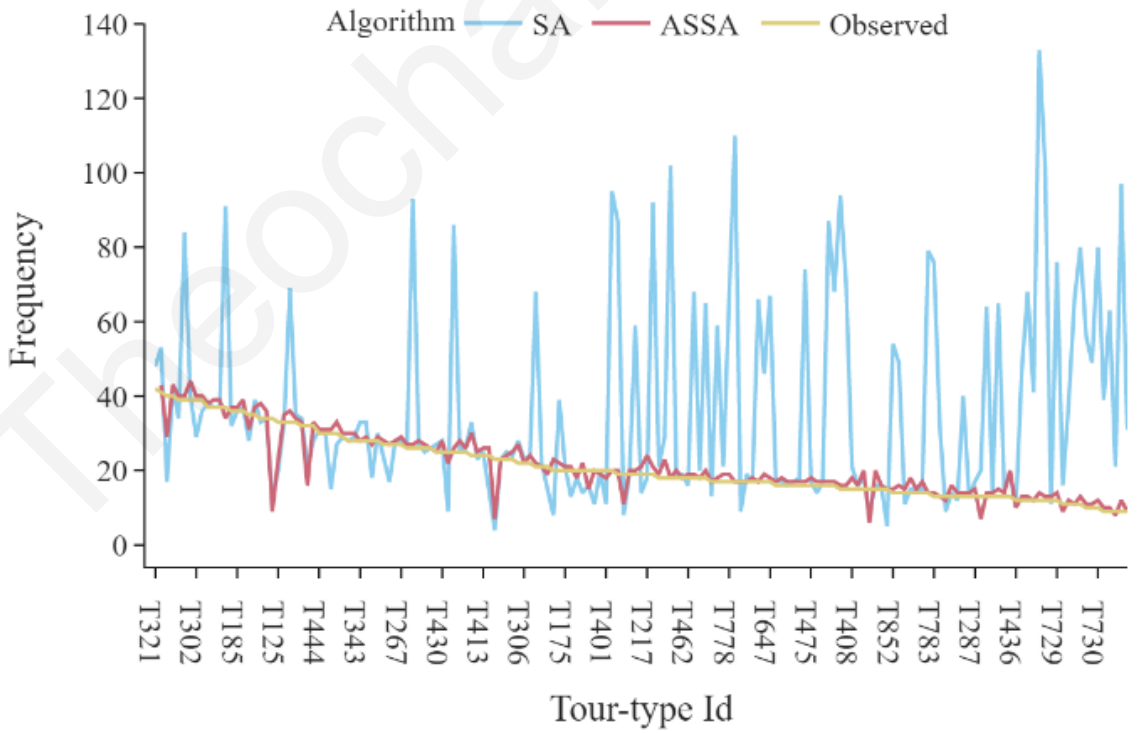

Figure 7.27 Comparison between the SA and ASSA resulting distributions and the calibrating one for the 20 least frequent tour-types.

7.5.4 Adaptive Sampling

As it has been already mentioned, the ASSA algorithm builds progressively a solution based on the difference between the current distribution of the solution's characteristics and the targeted one (i.e. calibration distribution). The continuous improvement of the solution is supported by the adaptive sampling mechanism which favours the introduction to the current solution of tours which will reduce the difference between the desired and the modelled output. The next section presents the benefits obtained from the adaptive sampling mechanism in contrast to the random sampling process taking place during the standard SA.

The first metric utilised to evaluate the adaptive sampling mechanism is the number of tours present in the solution during the optimisation process. The benchmark for this comparison is set at 25 thousand tours which coincides with the number of the observed tours. Figure 7.28 presents the progression of solution's size for the SA and ASSA algorithms, respectively. It can be noted that although neither algorithm manages to include the exact number of observed tours, however the ASSA alternative produces a significantly more populated and constantly improving solution. More precisely, ASSA results in a solution including almost 110 thousand tours compared to the 125 thousand required for the optimal solution (88% accuracy). It should be also noted that for the case of ASSA, the passage of iterations results in improved solutions and avoids getting trapped in local minima as in the case of SA.

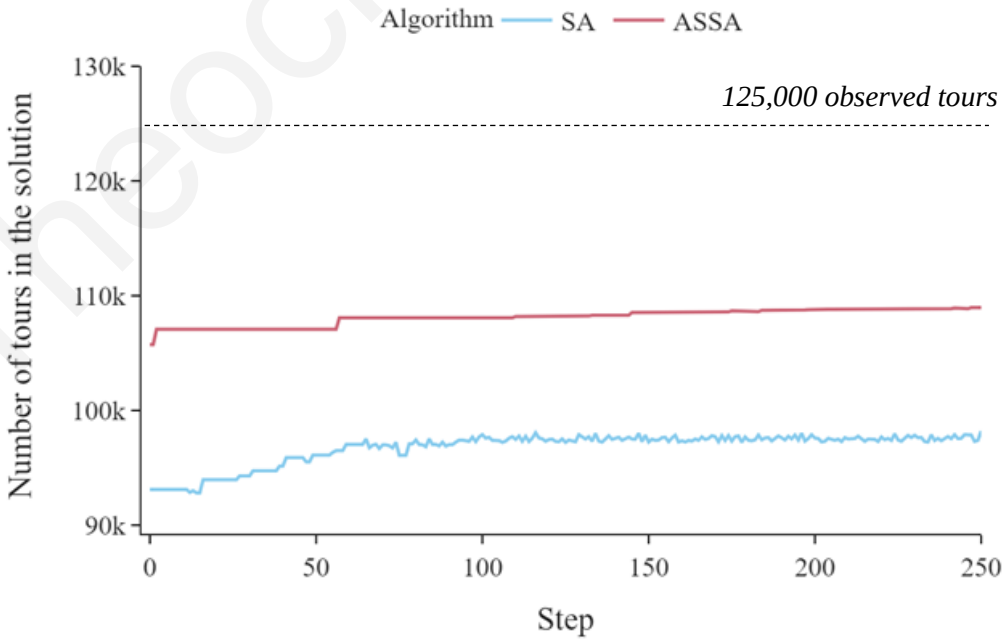

Figure 7.28 Number of tours in the solution during the optimisation process

Additionally, the evolution of the progression of the calibration fitting during the simulated annealing process (SA and ASSA) is presented in Figure 7.29. As it can be noted, the passage of iterations improves the fitting for both the SA and the ASSA algorithms, however ASSA achieves significantly greater accuracy in fewer steps. The bottom row presents in more detail the shaded area of the top row.

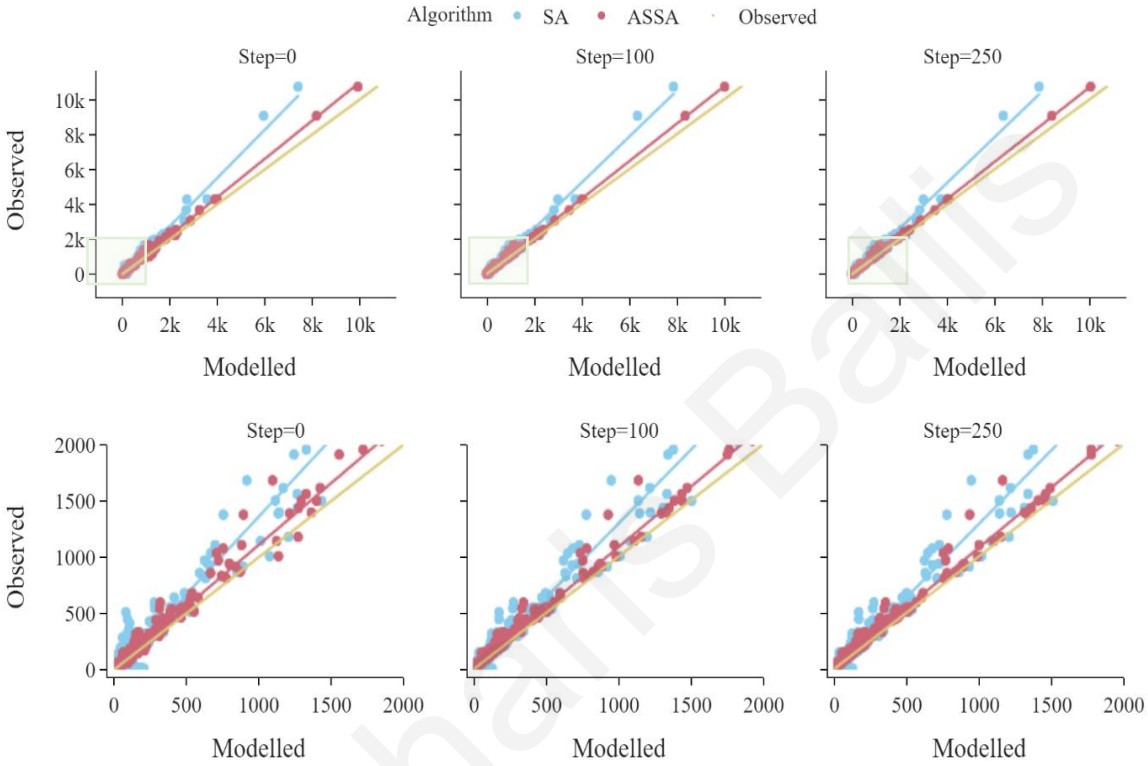

Figure 7.29 Evolution of calibration fitting during the simulated annealing process. The bottom row presents in more detail the shaded area of the top row.

Finally, Figure 7.30 presents in more detail the evolution of the calibration fitting for the 20 most frequent tour-types. As it can be observed, the ASSA algorithm builds solutions adhering to the calibration since a very early step. Although subtle, the improvement of the calibration fitting between the initial iteration (step=0) and the final one (step=250) is still notable. For example, the difference between the observed and the modelled tour-types T33 and T57 diminishes considerably with the passage of optimisation steps.

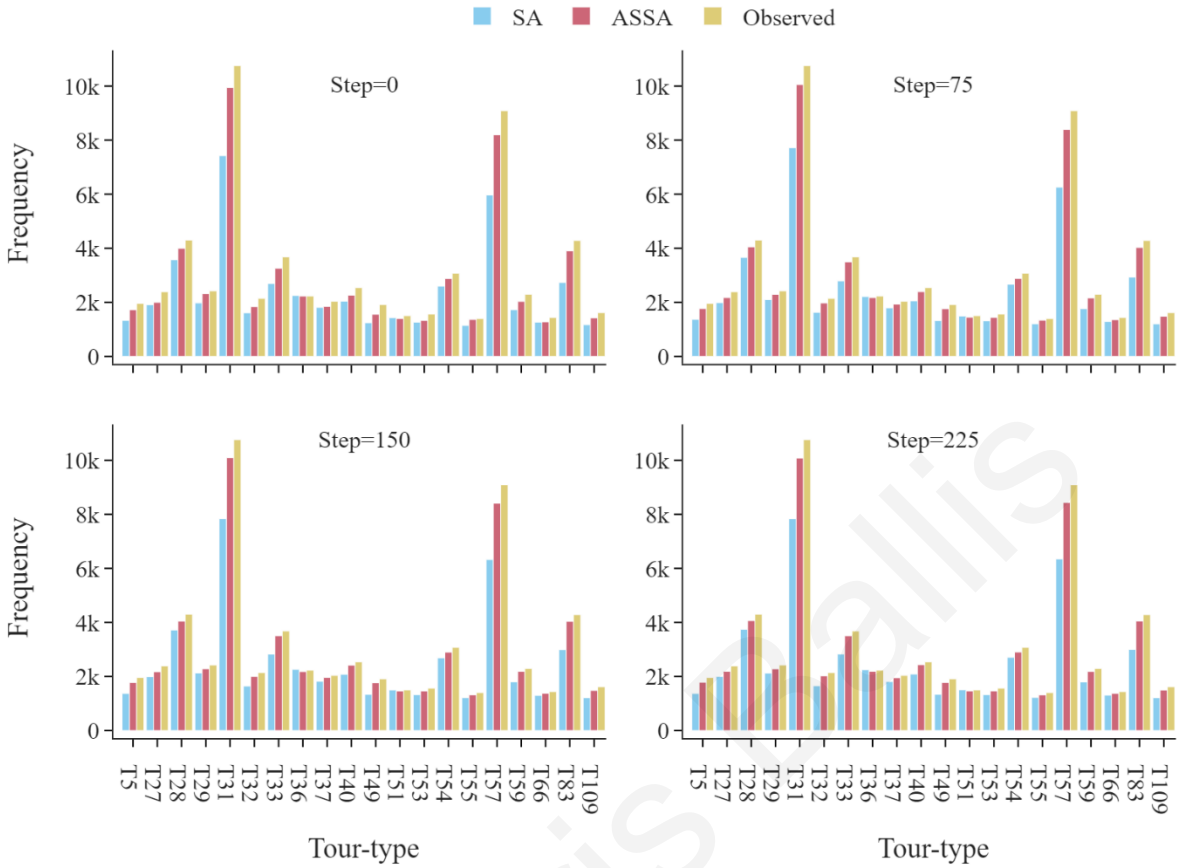

Figure 7.30 Progressive comparison of the modelled and the observed distributions during the optimisation process (20 most frequent tour-types).

The previous section verified the potential of the ASSA algorithm as a very efficient metaheuristic able to address problems of excessively large scale such as the one presented for the purposes of this thesis. In addition, it emphasized the benefits of utilizing an adaptive sampling mechanism during the optimization process.

7.6 Discussion of the Results

The previously presented evaluation process verified the scalability of the methodology as well as the applicability of the suggested ASSA algorithm for the tackling of large-scale combinatorial problems such as the conversion of ODs into individual tours. ASSA enabled the production of an accurate solution (85.9% accuracy), closely adhering to the provided calibration distribution without requiring prohibitively long processing time for its completion. In detail the overall processing time required for the conversion of 28 multi-period, purpose segmented ODs of 260 thousand trips reached 43 hours with most of the processing time being allocated to the identification module (40 hours). Therefore, it can be strongly suggested that large-scale OD matrices can efficiently and accurately be converted

to person-tours that can enable the in-depth study of travel behaviour at person-level. The implications are significant since this is one of the few examples allowing the preparation of disaggregate mobility information from widely available, aggregate data sources in the form of ODs.

Chapter 8
Conclusions and Future Research

Chapter 8 concludes the thesis, highlights its contribution, and suggests directions for future research.

8.1 Conclusions

The future of mobility promises environmentally friendly, integrated, and personalised travel on demand. The realisation of this prospect requires the development of advanced transport models and simulation tools able to accurately estimate travel behaviour at the person-level. However, the development of such models requires rich mobility information at person-level which is not always easily available. The transition to the Big-Data age accompanied by a plethora of technological advances in the field of urban sensing has started alleviating the scarcity of information regarding travel behaviour. For example, Mobile Phones and GPS tracking devices can provide vast quantities of very precise information regarding the travel behaviour of their holders. However, justified concerns regarding the privacy of users, demand for the anonymization of such data. A typical form of anonymization refers to the representation of travel behaviour information via aggregated Origin-Destination (OD) matrices. Apart from their use as data anonymisation device, ODs have traditionally constituted the most widespread mean of presenting travel demand patterns. ODs have over the years proven as a very efficient format to represent various dimensions of mobility (e.g. trip-purpose, time of departure, transport mode, etc.) but their aggregate nature prevents them from the study of travel behaviour at person-level. Although, ODs are perfectly suitable for the representation of total travel demand flows between pairs of locations, they are incapable of retaining significant travel behaviour information at the disaggregate-level such as trip-chaining and trip interdependency.

The previously presented Ph.D. Thesis proposed a novel modular methodological framework able to fully exploit the aggregated mobility information within ODs for the synthesis of completely tractable travel behaviour information at the disaggregate-level. In particular, the Thesis presented an efficient methodology for the conversion of multi-period and purpose dependent ODs firstly into sequences of trips originating and ending at a home location (i.e. tours) and subsequently into activity schedules. The completion of the proposed methodological framework required for interdisciplinary treatment. Elements from travel demand modelling, travel behaviour theory, graph-theory, combinatorial optimisation, and Big-Data analytics were forged into a cohesive framework able to address the problem of large-scale ODs' disaggregation to tours and activity schedules. During the development of the framework, various hindrances required attention while a plethora of interesting observations were also recorded. The conclusions drawn and the main contributions of the present Ph.D. research are summarised as follows:

1. **Multi-period and purpose segmented ODs can be indeed used for the study of travel behaviour at the person-level.** Despite their aggregate nature, ODs contain all the necessary information to reconstruct disaggregate travel demand patterns such as trip-chains, tours, and activity schedules. Restructuring them appropriately enables the unveiling of hidden information which can be exploited for the analysis of mobility at the disaggregate-level.
2. **ODs can be utilised as input data sources for disaggregate transport modelling.** The application of the suggested methodology allows for the synthesis of disaggregate information from aggregate ODs which are considerably easier to acquire compared to individual mobility traces.
3. **Conversion of multi-period ODs to a single Time Varying Graph (TVG) proves as a very eloquent form for the representation of mobility**, particularly suitable for the identification of continuous and interdependent travel behaviour manifestations such as trip-chains and tours.
4. **The spatiotemporal resolution of the inputted ODs is the most crucial factor regarding the performance of the suggested methodology**. In particular, the higher the resolution the more efficient the methodology.
5. **The developed integer/combinatorial optimisation routines require significant attention for efficient implementation, especially for large-scale applications.** In particular, the number of all the plausible sequences of trips originating and ending at home (i.e. tours) that can be formed from the individual trips contained in ODs can quickly grow to intractable levels. Nonetheless, observations regarding travel behaviour can drastically reduce the total number of the identifiable tours/activity schedules to handleable levels.
6. **The complexity of enumerating all the plausible tours within a hTVG can be significantly reduced by the application of network simplification techniques.** In detail, this research verified that various centrality measures can be effectively utilised for the simplification of a graph without inflicting considerable damage on traversability. A meticulous validation on the aspect concluded that PageRank centrality measure as the most suitable approach to simplify the structure of a graph while minimising the impact on traversability.
7. **Integer programming is the most accurate method for the identification of optimum combinations of tours/activity schedules which reproduce the travel demand patterns as captured in the inputted ODs.** However, expressing the optimisation problem as an integer exact mathematical programming problem can

result in excessive computational requirements. On the other hand, metaheuristic methods can prove as effective alternatives.

8. **Metaheuristic methods can be greatly benefited by the incorporation of problem specific information.** The study showcased the benefits in accuracy and processing time arising from exploitation of information regarding the expected output (i.e. calibration data).
9. **Anonymization of mobility related information through their representation in aggregate ODs is not guaranteeing the privacy of the tracked users.** The results obtained in the proof of concept application indicate that ODs can be very accurately reversed-engineered under the condition that the spatiotemporal resolution of the ODs is relatively (but not unrealistically) high.

8.2 Contribution

The completion of the previously presented Ph.D. Thesis resulted in a plethora of notable contributions summarised as follows:

1. **The Ph. D. Thesis proposes a novel framework for the exploitation of aggregate ODs to synthesise disaggregate mobility and travel behaviour information.** The verification of the methodology's potential opens new paths with regards to the analysis of personal travel behaviour based solely on aggregate data. As identified from the relevant literature review, apart from very few exceptions, the relevant methodologies have been entirely based on disaggregate input (e.g. travel diaries, GPS traces, personal smart-card data, etc.). Therefore, the presented proof regarding the potential of utilising aggregate data for the studying of personal travel behaviour can spark attention to a relatively unexplored research field.
2. **Preparation of relevant input for advanced disaggregate transport models based on widely available aggregate OD matrices.** The limited input requirements as well as the generality and the flexibility of the framework allow the synthesis of disaggregate mobility information based on a very wide range of ODs. Utilising ODs for the synthesis of suitable input for disaggregate transport models (e.g. agent-based, activity-based, microsimulation, etc.) can drastically increase their wider adoption.
3. **The introduction and the evaluation of the hybrid Time Varying Graph (hTVG) for transport related problems.** The completion of the presented framework required the expression of travel demand in a consistent and chronologically ordered fashion allowing the identification of continuous travel behaviour manifestations

(e.g. trip-chains, tours, etc.). Expressing travel demand through hTVG proved as the most suitable approach for the purposes of the framework. The here proposed hTVG format combines the advantages of static graphs with the dynamic nature of TVGs and allows the representation of dynamic phenomena using the extensively studied static graph format. This enables the application of standard, efficient and scalable graph-theory-based methodologies (originally developed for static graphs) for the analysis of highly dynamic systems. The use of a hTVG for the completion of this Thesis validated and supported the use of TVGs for transportation related problems, a possibility that has not been extensively evaluated.

4. **Proposition of a methodological framework achieving the reduction of the tours' enumeration problem within an OD matrix.** The requirement of scalability led to the development of sophisticated procedures able to simplify the studied, excessively large combinatorial problem. In particular, a suitable process for the simplification of the tours enumeration problem within a graph was developed and evaluated (Ballis and Dimitriou, 2020d). Although, the simplification procedure was specifically developed for the purposes of identifying all the possible tours within a graph, the process can be straightforwardly applied for the confinement of any enumeration process concerning a graph.
5. **Development of the Adaptive Sampling Simulated Annealing (ASSA) algorithm.** ASSA (Ballis and Dimitriou, 2020e) suggests a novel method for the solution of large-scale combinatorial problems when the solution is expected to adhere to certain characteristics expressed through a calibration distribution. Finally, the excessive scale of the studied combinatorial problem demanded for efficient combinatorial optimisation techniques. ASSA exploits any available calibration information in the form of a (joint) distribution for the guidance of the optimisation routine to neighbourhoods with solutions which adhere to the calibration data. Despite, being developed for the purposes of the niche studied problem, ASSA proves as a highly generalisable technique since it requires only a marginal distribution for its deployment.
6. **Proposition of a methodology to validate the consistency of OD matrices**. Enhancing the quality of existing ODs is particularly important to transport authorities and planners which have traditionally relied on ODs for the evaluation of a variety of policy scenarios (e.g. urban traffic optimisation, pedestrianisation schemes, regeneration of areas, etc.). Poor performance at decomposing existing

ODs to individual tours/activity schedules can signify inconsistencies related to travel behaviour (i.e. incomplete trip-chains, tours, etc.).

7. **Suggestion of a methodology enabling the anonymisation of trip-chains/activity schedules deriving from urban sensing data sources (mobile phones, GPS traces, etc.).** Despite the expected abundance of data in the Big-Data era, it is very likely that reasons of anonymity as well as practicality may require the aggregation of such data for their efficient and privacy-safe handling. Towards this direction, methodologies able to exploit vast quantities of aggregated data for the understanding of disaggregate behaviours can prove of considerable contribution. In addition, aggregating mobility traces to create ODs and subsequently desegregating them following the presented methodological framework, could introduce randomness in the dataset without affecting the observed travel demand patterns.

8.3 Future Research

The methodological framework developed for the purposes of this Ph. D. Thesis proposed a novel methodology aiming at the unveiling of disaggregate mobility patterns within aggregate ODs. Although, the Thesis went into great depth with regards to the evaluation of the methodology from multiple perspectives, many aspects remain untouched or require further study.

The suggestions regarding the direction of the future study on the topic are summarised as follows:

1. **Further experimentation with large-scale ODs to identify scalability inefficiencies and plausible approaches to counter them.** One of the key areas requiring further study is the application of the methodology on cases where the ODs are not as consistent as the ones utilised for evaluation. The application of the methodology on existing ODs can pinpoint possible inefficiencies that require attention. Similarly, further evaluation based on more ODs, varying in size and resolution, can provide further insight regarding the applicability of the methodology on a broader range of cases.
2. **Combination of the methodology with a population synthesis component.** The possibility to assign sociodemographic characteristics to the produced tour and activity schedules can significantly increase their explanatory value.
3. **Incorporation of multimodality in the methodological framework.** Although, the methodology is already capable of synthesising tours and activity schedules

completed with multiple transport modes, the graph component of the analysis (hTVG) could be enhanced in order to prevent the formation of transport-mode inconsistent tours.

4. **Incorporation of the network loading step into the methodological framework.** The combinatorial nature of the problem allows for the appearance of multiple optimum solutions. Although, the presence of calibration information can significantly aid the identification of a tours/activity schedules which resembles reality, further effort should be devoted on the assurance of the solution's representativeness. A possible improvement entails the incorporation of the network loading element into the procedure. More specifically, the framework can be extended by utilising the traffic conditions arising from the identified solutions to evaluate the realness of the output.
5. **Exploration of the possibility to express tours within ODs through a Markovian process.** The main element limiting the scalability of the presented framework is the time-consuming process of enumerating the possible tours in the OD derived graph. An alternative approach suggests the identification of tours through a Markovian process where the hTVG representation of travel demand can be utilised for the estimation of the Markovian transition probabilities required to form trip-chains and tours.
6. **Evaluation of alternative optimisation techniques other than the suggested ASSA algorithm.** The optimisation module could be replaced by appropriate alternatives (e.g. Genetic algorithms, Tabu Search, etc.) which could potentially prove more efficient at addressing the excessively large combinatorial presented problem.
7. **Exploration of the possibility to utilise the framework as a privacy guaranteeing mechanism for ODs deriving from urban sensing data sources.** As stated earlier, mobility data providers are usually unable to provide the individual traces of their users, however, they often accept to present them in the aggregate form of ODs. Incorporating additional privacy checks could convert the methodology to a mechanism guaranteeing the intractability of personal mobility information within ODs.

Bibliography

Abdelghany, A.F., Mahmassani, H.S., Chiu, Y.-C., 2007. Spatial Microassignment of Travel Demand with Activity Trip Chains. Transp. Res. Rec. J. Transp. Res. Board 1777, 36–46. https://doi.org/10.3141/1777-04

Abraham, J.E., Stefan, K.J., Hunt, J.D., 2012. Population synthesis using combinatorial optimization at multiple levels, in: Transportation Research Record. p. 17.

Alexander, L., Jiang, S., Murga, M., González, M.C., Gonzalez, M.C., 2015. Origin-destination trips by purpose and time of day inferred from mobile phone data. Transp. Res. Part C Emerg. Technol. 58, 240–250. https://doi.org/10.1016/j.trc.2015.02.018

Allahviranloo, M., Recker, W., 2013. Daily activity pattern recognition by using support vector machines with multiple classes. Transp. Res. Part B Methodol. 58, 16–43. https://doi.org/10.1016/j.trb.2013.09.008

Anda, C., Fourie, P., Erath, A., 2016. Transport Modelling in the Age of Big Data. Singapore - ETH Cent. Futur. Cities Lab. Work Repor.

Anda, C., Ordonez Medina, S., Arturo, S., Axhausen, K.W., Medina, S.A.O., 2020. Synthesising digital twin travellers Individual travel demand from aggregated mobile phone data Synthesising Digital Twin Travellers: Individual travel demand from aggregated mobile phone data. https://doi.org/10.3929/ethz-b-000442517

Andrés, M.E., Bordenabe, N.E., Chatzikokolakis, K., Palamidessi, C., 2013. Geo-indistinguishability: Differential privacy for location-based systems, in: Proceedings of the ACM Conference on Computer and Communications Security. pp. 901–914. https://doi.org/10.1145/2508859.2516735

Angria S, L., Dwi Sari, Y., Zarlis, M., Tulus, 2018. Data-driven Modelling for decision making under uncertainty, in: IOP Conference Series: Materials Science and Engineering. https://doi.org/10.1088/1757-899X/300/1/012013

Antoniou, C., Barcel??, J., Breen, M., Bullejos, M., Casas, J., Cipriani, E., Ciuffo, B., Djukic, T., Hoogendoorn, S., Marzano, V., Montero, L., Nigro, M., Perarnau, J., Punzo, V., Toledo, T., van Lint, H., 2016. Towards a generic benchmarking platform for origin-destination flows estimation/updating algorithms: Design, demonstration and validation. Transp. Res. Part C Emerg. Technol. 66, 79–98. https://doi.org/10.1016/j.trc.2015.08.009

Antoniou, C., Ben-Akiva, M., Bierlaire, M., Mishalani, R., 1997. Demand Simulation for Dynamic Traffic Assignment. IFAC Proc. Vol. 30, 633–637. https://doi.org/10.1016/s1474-6670(17)43892-4

Antoniou, C., Dimitriou, L., Pereira, F., 2019. Mobility patterns, big data and transport analytics : tools and applications for modeling. Elsevier.

Antoniou, C., Spyropoulou, I., 2014. Determinants of driver response to variable message sign information in Athens. IET Intell. Transp. Syst. 9, 453–466. https://doi.org/10.1049/iet-its.2014.0053

Antosiewicz, M., Koloch, G., Kaminski, B., 2013. Choice of best possible metaheuristic algorithm for the travelling salesman problem with limited computational time : quality

, uncertainty and speed. J. Theor. Appl. Comput. Sci. 7, 46–55.

Axhausen, K.W., 2007. Concepts of Travel Behaviour Research, in: Threats from Car Traffic to the Quality of Urban Life. Emerald Group Publishing Limited, pp. 165–185. https://doi.org/10.1108/9780080481449-009

Azevedo, C.L., Marczuk, K., Raveau, S., Soh, H., Adnan, M., Basak, K., Loganathan, H., Deshmunkh, N., Lee, D.-H., Frazzoli, E., Ben-Akiva, M., 2016. Microsimulation of Demand and Supply of Autonomous Mobility On Demand. Transp. Res. Rec. J. Transp. Res. Board 2564, 21–30. https://doi.org/10.3141/2564-03

Aziz, H.M.A., Park, B.H., Morton, A., Stewart, R.N., Hilliard, M., Maness, M., 2018. A high resolution agent-based model to support walk-bicycle infrastructure investment decisions: A case study with New York City. Transp. Res. Part C Emerg. Technol. 86, 280–299. https://doi.org/10.1016/j.trc.2017.11.008

Balakrishna, R., Antoniou, C., Ben-Akiva, M.E., Koutsopoulos, H.N., Wen, Y., 2007. Calibration of Microscopic Traffic Simulation Models: Methods and Application. Transp. Res. Rec. J. Transp. Res. Board 1999, 198–207. https://doi.org/10.3141/1999-21

Ballis, H., Dimitriou, L., 2020a. Optimal synthesis of tours from multi-period origin-destination matrices using elements from graph theory and integer programming. Eur. J. Transp. Infrastruct. Res. 20, 1–21. https://doi.org/10.18757/ejtir.2020.20.4.5303

Ballis, H., Dimitriou, L., 2020b. Revealing personal activities schedules from synthesizing multi-period origin-destination matrices. Transp. Res. Part B Methodol. 139, 224–258. https://doi.org/10.1016/j.trb.2020.06.007

Ballis, H., Dimitriou, L., 2020c. Deriving Daily Activity Schedules from Dynamic, Purpose Dependent Origin-Destination Matrices. Transp. Res. Rec.

Ballis, H., Dimitriou, L., 2020d. Enumeration of tours in large-scale graphs derived from OD matrices' connectivity: Overcoming the combinatorial explosion, Working Paper, Submitted for publication in Networks., March, 2020.

Ballis, H., Dimitriou, L., 2020e. Adaptive Sampling Simulated Annealing for the conversion of Origin-Destination matrices to home-based trip-chains, Working Paper, Submitted for publication in Transportation Research Part C: Emerging Technologies., May, 2020.

Ballis, H., Dimitriou, L., 2019. Optimal population of trip chains synthesis from multi-period origin-destination matrices, in: Proceedings of Transportation Research Board 98th Annual Meeting, Washington D.C.

Ballis, H., Dimitriou, L., Ballis, A., 2018. A preliminary preparation of trip chains from origin-destination matrices for supporting activity-based models, in: Proceedings of Transportation Research Board 97th Annual Meeting, Washington D.C.

Balmer, M., Axhausen, K.W., Nagel, K., 2006. A Demand Generation Framework for Large Scale Micro Simulations, in: Transportation Research Board (TRB) Annual Meeting.

Balzotti, C., Bragagnini, A., Briani, M., Cristiani, E., 2018. Understanding Human Mobility Flows from Aggregated Mobile Phone Data. IFAC-PapersOnLine 51, 25–30. https://doi.org/10.1016/j.ifacol.2018.07.005

Barabási, A.-L., 2016. Network science. Cambridge university press.

Bassolas, A., Ramasco, J.J., Herranz, R., Cantú-Ros, O.G., 2019. Mobile phone records to feed activity-based travel demand models: MATSim for studying a cordon toll policy in Barcelona. Transp. Res. Part A Policy Pract. 121, 56–74.

https://doi.org/10.1016/j.tra.2018.12.024

Batty, M., 2013. Big data, smart cities and city planning. Dialogues Hum. Geogr. 3, 274–279. https://doi.org/10.1177/2043820613513390

Batty, M., Axhausen, K.W., Giannotti, F., Pozdnoukhov, A., Bazzani, A., Wachowicz, M., Ouzounis, G., Portugali, Y., 2012. Smart cities of the future. Eur. Phys. J. Spec. Top. 214, 481–518. https://doi.org/10.1140/epjst/e2012-01703-3

Beckman, R.J., Baggerly, K.A., McKay, M.D., 1996. Creating synthetic baseline populations. Transp. Res. Part A Policy Pract. 30, 415–429. https://doi.org/10.1016/0965-8564(96)00004-3

Beckwith, R., Sherry, J., Prendergast, D., 2019. Data Flow in the Smart City: Open Data Versus the Commons, in: The Hackable City. Springer Singapore, pp. 205–221. https://doi.org/10.1007/978-981-13-2694-3_11

Bekhor, S., Dobler, C., Axhausen, K., 2011. Integration of Activity-Based and Agent-Based Models. Transp. Res. Rec. J. Transp. Res. Board 2255, 38–47. https://doi.org/10.3141/2255-05

Bell, M.G.H.H., 1991. The estimation of origin-destination matrices by constrained generalised least squares. Transp. Res. Part B 25, 13–22. https://doi.org/10.1016/0191-2615(91)90010-G

Ben-Akiva, M., Bottom, J., Gao, S., Koutsopoulos, H.N., Wen, Y., 2007. Towards Disaggregate Dynamic Travel Forecasting Models. Tsinghua Sci. Technol. 12, 115–130. https://doi.org/10.1016/S1007-0214(07)70019-6

Bhat, C.R., 1996. A hazard-based duration model of shopping activity with nonparametric baseline specification and nonparametric control for unobserved heterogeneity. Transp. Res. Part B Methodol. 30, 189–207. https://doi.org/10.1016/0191-2615(95)00029-1

Bhat, C.R., Guo, J., Srinivasan, S., Sivakumar, A., 2004. Comprehensive Econometric Microsimulator for Daily Activity-Travel Patterns. Transp. Res. Rec. J. Transp. Res. Board 1894, 57–66. https://doi.org/10.3141/1894-07

Bhat, C.R., Koppelman, F.S., 1999. Activity-Based Modeling of Travel Demand 35–61. https://doi.org/10.1007/978-1-4615-5203-1_3

Bhat, C.R., Srinivasan, S., Axhausen, K.W., 2005. An analysis of multiple interepisode durations using a unifying multivariate hazard model. Transp. Res. Part B Methodol. 39, 797–823. https://doi.org/10.1016/j.trb.2004.11.002

Bindschaedler, V., Shokri, R., 2016. Synthesizing Plausible Privacy-Preserving Location Traces. Proc. - 2016 IEEE Symp. Secur. Privacy, SP 2016 546–563. https://doi.org/10.1109/SP.2016.39

Blondel, V.D., Decuyper, A., Krings, G., 2015. A survey of results on mobile phone datasets analysis. EPJ Data Sci. https://doi.org/10.1140/epjds/s13688-015-0046-0

Bonabeau, E., 2002. Agent-based modeling: Methods and techniques for simulating human systems. Proc. Natl. Acad. Sci. U. S. A. 99, 7280–7287. https://doi.org/10.1073/pnas.082080899

Bonacich, P., 1972. Factoring and weighting approaches to status scores and clique identification. J. Math. Sociol. 2, 113–120. https://doi.org/10.1080/0022250X.1972.9989806

Bonnel, P., Hombourger, E., Olteanu-Raimond, A.M., Smoreda, Z., 2015. Passive mobile phone dataset to construct origin-destination matrix: Potentials and limitations, in:

Transportation Research Procedia. Elsevier, pp. 381–398. https://doi.org/10.1016/j.trpro.2015.12.032

Borndörfer, R., Grötschel, M., Pfetsch, M.E., 2005. A path-based model for line planning in public transport. Informationstechnik 1–17.

Borshchev, A., Filippov, A., 2004. From System Dynamics and Discrete Event to Practical Agent Based Modeling : Reasons , Techniques , Tools 1 . Simulation Modeling : Abstraction Levels , Major Paradigms. 22nd Int. Conf. Syst. Dyn. Soc. 25-29 July 2004 45.

Bouzidi, M., Dalveren, Y., Cheikh, F.A., Derawi, M., 2020. Use of the IQRF Technology in Internet-of-Things-Based Smart Cities. IEEE Access 8, 56615–56629. https://doi.org/10.1109/ACCESS.2020.2982558

Bowman, J.L., 1998. The day activity schedule approach to travel demand analysis. Metro 185.

Bowman, J.L., Ben-Akiva, M., 2000. Activity-based disaggregate travel demand model system with activity schedules. Transp. Res. Part A Policy Pract. 35, 1–28. https://doi.org/10.1016/S0965-8564(99)00043-9

Brewer, A.M., 1998. Work design, felxible work arrangements and travel behaviour: policy implications. Transp. Policy 5, 93–101. https://doi.org/10.1016/S0967-070X(98)00003-1

Caceres, N., Romero, L.M., Benitez, F.G., 2013. Inferring origin-destination trip matrices from aggregate volumes on groups of links: A case study using volumes inferred from mobile phone data. J. Adv. Transp. 47, 650–666. https://doi.org/10.1002/atr.187

Caceres, N., Wideberg, J.P., Benitez, F.G., 2007. Deriving origin-destination data from a mobile phone network. IET Intell. Transp. Syst. 1, 15–26. https://doi.org/10.1049/iet-its:20060020

Calabrese, F., Diao, M., Di Lorenzo, G., Ferreira, J., Ratti, C., Lorenzo, G. Di, Ferreira, J., Ratti, C., 2013. Understanding individual mobility patterns from urban sensing data - A mobile phone trace example. Transp. Res. Part C Emerg. Technol. 26, 301–313. https://doi.org/10.1016/j.trc.2012.09.009

Cantelmo, G., Qurashi, M., Prakash, A.A., Antoniou, C., Viti, F., 2019. Incorporating trip chaining within online demand estimation. Transp. Res. Part B Methodol. https://doi.org/10.1016/j.trb.2019.05.010

Cascetta, E., 1984. Estimation of trip matrices from traffic counts and survey data: A generalized least squares estimator. Transp. Res. Part B 18, 289–299. https://doi.org/10.1016/0191-2615(84)90012-2

Casteigts, A., 2018. Finding Structure in Dynamic Networks.

Chen, C., Bian, L., Ma, J., 2014. From traces to trajectories: How well can we guess activity locations from mobile phone traces? Transp. Res. Part C Emerg. Technol. 46, 326–337. https://doi.org/10.1016/j.trc.2014.07.001

Chen, C., Ma, J., Susilo, Y., Liu, Y., Wang, M., 2016. The promises of big data and small data for travel behavior (aka human mobility) analysis. Transp. Res. Part C Emerg. Technol. https://doi.org/10.1016/j.trc.2016.04.005

Cheng, E., Grossman, J.W., Lipman, M.J., 2003. Time-stamped graphs and their associated influence digraphs. Discret. Appl. Math. 128, 317–335. https://doi.org/10.1016/S0166-218X(02)00497-3

Choupani, A.-A., Mamdoohi, A.R., 2016. Population Synthesis Using Iterative Proportional Fitting (IPF): A Review and Future Research. Transp. Res. Procedia 17, 223–233. https://doi.org/10.1016/j.trpro.2016.11.078

Chow, R., Golle, P., 2009. Faking contextual data for fun, profit, and privacy, in: Proceedings of the ACM Conference on Computer and Communications Security. ACM Press, New York, New York, USA, pp. 105–108. https://doi.org/10.1145/1655188.1655204

Chu, Z., Cheng, L., Chen, H., 2012. A Review of Activity-Based Travel Demand Modeling, in: CICTP 2012. pp. 48–59. https://doi.org/10.1061/9780784412442.006

Cich, G., Knapen, L., Galland, S., Vuurstaek, J., Neven, A., Bellemans, T., 2016. Towards an Agent-based Model for Demand-Responsive Transport Serving Thin Flows, in: Procedia Computer Science. pp. 952–957. https://doi.org/10.1016/j.procs.2016.04.191

Cich, G., Knapen, L., Maciejewski, M., Yasar, A.U.H., Bellemans, T., Janssens, D., 2017. Modeling Demand Responsive Transport using SARL and MATSim, in: Procedia Computer Science. pp. 1074–1079. https://doi.org/10.1016/j.procs.2017.05.387

Çolak, S., Alexander, L.P., Alvim, B.G., Mehndiratta, S.R., Gonzalez, M.C., 2015. Analyzing Cell Phone Location Data for Urban Travel. Transp. Res. Rec. J. Transp. Res. Board 2526, 126–135. https://doi.org/10.3141/2526-14

Cottrill, C.D., Pereira, F.C., Zhao, F., Dias, I.F., Lim, H.B., Ben-Akiva, M.E., Zegras, P.C., 2013. Future mobility survey. Transp. Res. Rec. https://doi.org/10.3141/2354-07

Dacko, S.G., Spalteholz, C., 2014. Upgrading the city: Enabling intermodal travel behaviour. Technol. Forecast. Soc. Change 89, 222–235. https://doi.org/10.1016/j.techfore.2013.08.039

De Montjoye, Y.A., Hidalgo, C.A., Verleysen, M., Blondel, V.D., 2013. Unique in the Crowd: The privacy bounds of human mobility. Sci. Rep. 3, 1–5. https://doi.org/10.1038/srep01376

Department for Transport, 2017. National Travel Survey: England 2016, National Travel Survey.

Djavadian, S., Chow, J.Y.J., 2017. An agent-based day-to-day adjustment process for modeling 'Mobility as a Service' with a two-sided flexible transport market. Transp. Res. Part B Methodol. 104, 36–57. https://doi.org/10.1016/j.trb.2017.06.015

Donnelly, R. (Rick), 2010. Advanced Practices in Travel Forecasting, Advanced Practices in Travel Forecasting. https://doi.org/10.17226/22950

Durand, A., Harms, L., 2018. Mobility-as-a-Service and changes in travel preferences and travel behaviour : a systematic literature review. Bijdr. aan het Colloq. Vervoer. Speurw. 1–15.

Dwork, C., Naor, M., Pitassi, T., Rothblum, G.N., 2010. Differential privacy under continual observation, in: Proceedings of the Annual ACM Symposium on Theory of Computing. pp. 715–724. https://doi.org/10.1145/1806689.1806787

E. Ramadan, O., P. Sisiopiku, V., 2019. A Critical Review on Population Synthesis for Activity- and Agent-Based Transportation Models, in: Transportation [Working Title]. IntechOpen. https://doi.org/10.5772/intechopen.86307

Eagle, N., Pentland, A.S., 2009. Eigenbehaviors: Identifying structure in routine. Behav. Ecol. Sociobiol. 63, 1057–1066. https://doi.org/10.1007/s00265-009-0739-0

Ebadi, N., Kang, J.E., Hasan, S., 2017. Constructing activity–mobility trajectories of college

students based on smart card transaction data. Int. J. Transp. Sci. Technol. 6, 316–329. https://doi.org/10.1016/j.ijtst.2017.08.003

Edwards, R., Glass, L., 2000. Combinatorial explosion in model gene networks. Chaos 10, 691–704. https://doi.org/10.1063/1.1286997

Emmerink, R.H.., Axhausen, K.W., Nijkamp, P., Rietveld, P., 1994. Effects of information in road transport networks with non-recurrent congestion 21–53.

Estrada, E., Rodríguez-Velázquez, J.A., 2005. Subgraph centrality in complex networks. Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys. 71. https://doi.org/10.1103/PhysRevE.71.056103

Ettema, D., Borgers, A., Timmermans, H., 1995. Competing risk hazard model of activity choice, timing, sequencing, and duration. Transp. Res. Rec. 1493, 101–109.

Fagnant, D.J., Kockelman, K.M., 2014. The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. Transp. Res. Part C Emerg. Technol. 40, 1–13. https://doi.org/10.1016/j.trc.2013.12.001

Farooq, B., Bierlaire, M., Hurtubia, R., Flötteröd, G., 2013. Simulation based population synthesis. Transp. Res. Part B Methodol. 58, 243–263. https://doi.org/10.1016/j.trb.2013.09.012

Ferreira, A., 2004. Building a reference combinatorial model for MANETs. IEEE Netw. 18, 24–29. https://doi.org/10.1109/MNET.2004.1337732

Ferreira, L., Charles, P., Tether, C., 2007. Evaluating flexible transport solutions. Transp. Plan. Technol. 30, 249–269. https://doi.org/10.1080/03081060701395501

Flötteröd, G., Bierlaire, M., Nagel, K., 2011. Bayesian demand calibration for dynamic traffic simulations. Transp. Sci. 45, 541–561. https://doi.org/10.1287/trsc.1100.0367

Fox, B.L., 1993. Integrating and accelerating tabu search, simulated annealing, and genetic algorithms. Ann. Oper. Res. 41, 47–67. https://doi.org/10.1007/BF02022562

Franco, P., Ballis, H., Stefanescu, C., Sari, N., 2019. Business Models for New Mobility Service : Demand Modelling tools for a successful implementation of MaaS, in: 13th ITS European Congress. pp. 3–6.

García-Jiménez, M.E., Ruíz, T., Mars, L., García-Garcés, P., 2014. Changes in the Scheduling Process According to Observed Activity-travel Flexibility. Procedia - Soc. Behav. Sci. 160, 484–493. https://doi.org/10.1016/j.sbspro.2014.12.161

Gardner, B., Abraham, C., 2007. What drives car use? A grounded theory analysis of commuters' reasons for driving. Transp. Res. Part F Traffic Psychol. Behav. 10, 187–200. https://doi.org/10.1016/j.trf.2006.09.004

Ge, Q., Fukuda, D., 2016. Updating origin-destination matrices with aggregated data of GPS traces. Transp. Res. Part C Emerg. Technol. 69, 291–312. https://doi.org/10.1016/j.trc.2016.06.002

Geman, S., Geman, D., 1984. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-6, 721–741. https://doi.org/10.1109/TPAMI.1984.4767596

Giesecke, R., Surakka, T., Hakonen, M., 2016. Conceptualising Mobility as a Service. 2016 11th Int. Conf. Ecol. Veh. Renew. Energies, EVER 2016. https://doi.org/10.1109/EVER.2016.7476443

Glauber, R.J., 1963. Time-dependent statistics of the Ising model. J. Math. Phys. 4, 294–

307. https://doi.org/10.1063/1.1703954

Gómez, D., Figueira, J.R., Eusébio, A., 2013. Modeling centrality measures in social network analysis using bi-criteria network flow optimization problems. Eur. J. Oper. Res. 226, 354–365. https://doi.org/10.1016/j.ejor.2012.11.027

Gonzalez, M.C., Hidalgo, C.A., Barabasi, A.-L., 2008. Understanding individual human mobility patterns 453, 779–782. https://doi.org/10.1038/nature06958

Goodwin, P.B., 1981. The usefulness of travel budgets. Transp. Res. Part A Gen. 15, 97–106. https://doi.org/10.1016/0191-2607(83)90019-5

Goulet Langlois, G., Koutsopoulos, H.N., Zhao, J., 2016. Inferring patterns in the multi-week activity sequences of public transport users. Transp. Res. Part C Emerg. Technol. 64, 1–16. https://doi.org/10.1016/j.trc.2015.12.012

Goulias, K., Barbara, S., 2009. Encyclopedia of Complexity and Systems Science, Encyclopedia of Complexity and Systems Science. https://doi.org/10.1007/978-0-387-30440-3

Goulias, K.G., 2009. Travel Behavior and Demand Analysis and Prediction, in: Encyclopedia of Complexity and Systems Science. Springer New York, pp. 9536–9565. https://doi.org/10.1007/978-0-387-30440-3_565

Goulias, K.G., Kitamura, R., 1991. Recursive Model System for Trip Generation and Trip Chaining. Transp. Res. Rec. 59–66.

Gu, Y., 2004. Integrating a Regional Planning Model (TRANSIMS) With an Operational Model (CORSIM).

Guilherme, M., Soares, F., 2013. Simulating Drivers ' Decision-making Under Information Dissemination. Simulating Drivers' Decis. Under Inf. Dissem. 68.

Gurobi, 2020. Gurobi - The fastest solver [WWW Document]. URL https://www.gurobi.com/ (accessed 4.25.20).

Haario, H., Saksman, E., 1991. Simulated annealing process in general state space. Adv. Appl. Probab. 23, 866–893. https://doi.org/10.2307/1427681

Hagberg, A.A., Schult, D.A., Swart, P.J., 2008. Exploring network structure, dynamics, and function using NetworkX, in: Proc. 7th SciPy Conf., Varoquaux G, Vaught T, and Millman J (Eds). pp. 11–15.

Hall, J.D., Palsson, C., Price, J., 2018. Is Uber a substitute or complement for public transit? J. Urban Econ. 108, 36–50. https://doi.org/10.1016/j.jue.2018.09.003

Han, G., Sohn, K., 2016. Activity imputation for trip-chains elicited from smart-card data using a continuous hidden Markov model. Transp. Res. Part B Methodol. 83, 121–135. https://doi.org/10.1016/j.trb.2015.11.015

Harper, C.D., Hendrickson, C.T., Mangones, S., Samaras, C., 2016. Estimating potential increases in travel with autonomous vehicles for the non-driving, elderly and people with travel-restrictive medical conditions. Transp. Res. Part C Emerg. Technol. 72, 1–9. https://doi.org/10.1016/j.trc.2016.09.003

Hart, W.E., Laird, C.D., Watson, J.-P., Woodruff, D.L., Hackebeil, G.A., Nicholson, B.L., Siirola, J.D., 2017. Pyomo — Optimization Modeling in Python. https://doi.org/10.1007/978-3-319-58821-6

Hartgen, D.T., 2013. Hubris or humility? Accuracy issues for the next 50 years of travel demand modeling. Transportation (Amst). 40, 1133–1157.

https://doi.org/10.1007/s11116-013-9497-y

Hilgert, T., Kagerbauer, M., Schuster, T., Becker, C., 2016. Optimization of Individual Travel Behavior through Customized Mobility Services and their Effects on Travel Demand and Transportation Systems. Transp. Res. Procedia 19, 58–69. https://doi.org/10.1016/j.trpro.2016.12.068

Hoffman, K.L., 2000. Combinatorial optimization: Current successes and directions for the future. J. Comput. Appl. Math. 124, 341–360. https://doi.org/10.1016/S0377-0427(00)00430-1

Hoffman, K.L., Padberg, M., Rinaldi, G., 2013. Traveling Salesman Problem, in: Gass, S.I., Fu, M.C. (Eds.), Encyclopedia of Operations Research and Management Science. Springer US, Boston, MA, pp. 1573–1578. https://doi.org/10.1007/978-1-4419-1153-7_1068

Holmberg, P.-E., Collado, M., Sarasini, S., Williander, M., 2016. Mobility as a Service: Describing the Framework.

Hopkins, D., García Bengoechea, E., Mandic, S., 2019. Adolescents and their aspirations for private car-based transport. Transportation (Amst). 1–27. https://doi.org/10.1007/s11116-019-10044-4

Horn, C., Gursch, H., Kern, R., Cik, M., 2017. QZtool—Automatically generated origin-destination matrices from cell phone trajectories, in: Advances in Intelligent Systems and Computing. Springer Verlag, pp. 823–833. https://doi.org/10.1007/978-3-319-41682-3_68

Horn, M.E.., 2002. Multi-modal and demand-responsive passenger transport systems: A modelling framework with embedded control systems. Transp. Res. Part A Policy Pract. 36, 167–188. https://doi.org/10.1016/S0965-8564(00)00043-4

Huber, S., Lißner, S., 2019. Disaggregation of aggregate GPS-based cycling data – How to enrich commercial cycling data sets for detailed cycling behaviour analysis. Transp. Res. Interdiscip. Perspect. 2, 100041. https://doi.org/10.1016/j.trip.2019.100041

IBM, 2020. CPLEX Optimizer | IBM [WWW Document]. URL https://www.ibm.com/analytics/cplex-optimizer (accessed 7.28.18).

Ickowicz, A., Sparks, R., 2015. Estimation of an origin/destination matrix: application to a ferry transport data. Public Transp. 7, 235–258. https://doi.org/10.1007/s12469-015-0102-y

Ingber, L., 1996. Adaptive simulated annealing (ASA): Lessons learned. Control Cybern. 25, 32–54.

Iqbal, M.S., Choudhury, C.F., Wang, P., Gonzalez, M.C., 2014. Development of origin-destination matrices using mobile phone call data. Transp. Res. Part C Emerg. Technol. 40, 63–74. https://doi.org/10.1016/j.trc.2014.01.002

Isaacman, S., Becker, R., Cáceres, R., Martonosi, M., Rowland, J., Varshavsky, A., Willinger, W., 2012. Human mobility modeling at metropolitan scales, in: MobiSys'12 - Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services. pp. 239–251. https://doi.org/10.1145/2307636.2307659

Jha, V., 2015. Study of Machine Learning Methods in Intelligent Transportation Systems.

Jiang, S., Ferreira, J., González, M.C., 2012. Clustering daily patterns of human activities in the city, in: Data Mining and Knowledge Discovery. pp. 478–510. https://doi.org/10.1007/s10618-012-0264-z

Jiang, S., Yang, Y., Gupta, S., Veneziano, D., Athavale, S., González, M.C., 2016. The TimeGeo modeling framework for urban motility without travel surveys, in: Proceedings of the National Academy of Sciences of the United States of America. pp. E5370–E5378. https://doi.org/10.1073/pnas.1524261113

Joh, C.H., Arentze, T., Hofman, F., Timmermans, H., 2002. Activity pattern similarity: A multidimensional sequence alignment method. Transp. Res. Part B Methodol. 36, 385–403. https://doi.org/10.1016/S0191-2615(01)00009-1

Johnson, D.B., 1975. Finding All the Elementary Circuits of a Directed Graph. SIAM J. Comput. 4, 77–84. https://doi.org/10.1137/0204007

Johnson, D.S., Papadimitriou, C.H., Steiglitz, K., 1984. Combinatorial Optimization: Algorithms and Complexity. Am. Math. Mon. 91, 209. https://doi.org/10.1007/PL00009491

Jovicic, G., 2001. Activity Based Travel Demand Modelling: A Literature Study, Danmarks TransportForskning.

Jun, C., Dongyuan, Y., 2013. Estimating smart card commuters origin-destination distribution based on APTS data. J. Transp. Syst. Eng. Inf. Technol. 13, 47–53. https://doi.org/10.1016/S1570-6672(13)60116-6

Kamargianni, M., Matyas, M., 2017. The Business Ecosystem of Mobility-as-a-Service. Transp. Res. Board Annu. Meet. 14.

Kamargianni, M., Matyas, M., Li, W., Schäfer, A., 2015. Feasibility Study for " Mobility as a Service " concept in London 84. https://doi.org/10.13140/RG.2.1.3808.1124

Karagiannis, G., Konomi, B.A., Lin, G., Liang, F., 2017. Parallel and interacting stochastic approximation annealing algorithms for global optimisation. Stat. Comput. 27, 927–945. https://doi.org/10.1007/s11222-016-9663-0

Kato, R., Iwata, M., Hara, T., Suzuki, A., Xie, X., Arase, Y., Nishio, S., 2012. A dummy-based anonymization method based on user trajectory with pauses, in: GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems. ACM Press, New York, New York, USA, pp. 249–258. https://doi.org/10.1145/2424321.2424354

Katrakazas, C., Antoniou, C., Vazquez, N.S., Trochidis, I., Arampatzis, S., 2019. Big Data and Emerging Transportation Challenges: Findings from the NOESIS project, in: 2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS). IEEE, pp. 1–9. https://doi.org/10.1109/MTITS.2019.8883308

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science (80-.). 220, 671–680. https://doi.org/10.1126/science.220.4598.671

Kitchin, R., 2014. The real-time city? Big data and smart urbanism. GeoJournal 79, 1–14. https://doi.org/10.1007/s10708-013-9516-8

Klotz, E., Newman, A.M., 2013. Practical guidelines for solving difficult mixed integer linear programs. Surv. Oper. Res. Manag. Sci. https://doi.org/10.1016/j.sorms.2012.12.001

Klügl, F., 2010. Agent-Based Simulation Engineering. Habilitation 217. https://doi.org/10.1007/978-1-4471-5634-5_12

Korte, B., 2001. Combinatorial optimization: Theory and algorithms, Computers & Mathematics with Applications. https://doi.org/10.1016/s0898-1221(01)90023-9

Kostakos, V., 2009. Temporal graphs. Phys. A Stat. Mech. its Appl. 388, 1007–1023. https://doi.org/10.1016/j.physa.2008.11.021

Krumm, J., 2009. Realistic driving trips for location privacy, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer, Berlin, Heidelberg, pp. 25–41. https://doi.org/10.1007/978-3-642-01516-8_4

Kumar, R., Calders, T., 2018. 2SCENT: An Efficient algorithm for enumerating all simple temporal cycles. Proc. VLDB Endow. 11, 1441–1453. https://doi.org/10.14778/3236187.3236197

Land, A.H., Doig, A.G., 1960. An Automatic Method of Solving Discrete Programming Problems. Econometrica 28, 497. https://doi.org/10.2307/1910129

Langville, A.N., Meyer, C.D., 2005. A survey of eigenvector methods for web information retrieval. SIAM Rev. 47, 135–161. https://doi.org/10.1137/S0036144503424786

Lee, Y., Hickman, M., Washington, S., 2007. Household type and structure, time-use pattern, and trip-chaining behavior. Transp. Res. Part A Policy Pract. 41, 1004–1020. https://doi.org/10.1016/j.tra.2007.06.007

Li, B., 2005. Bayesian inference for origin-destination matrices of transport networks using the em algorithm. Technometrics 47, 399–408. https://doi.org/10.1198/004017005000000283

Liang, F., Cheng, Y., Lin, G., 2014. Simulated stochastic approximation annealing for global optimization with a square-root cooling schedule. J. Am. Stat. Assoc. 109, 847–863. https://doi.org/10.1080/01621459.2013.872993

Lin, X., Sun, W., Veeraraghavan, M., Hu, W., 2016. Time-shifted multilayer graph: A routing framework for bulk data transfer in optical circuit-switched networks with assistive storage. J. Opt. Commun. Netw. 8, 162–174. https://doi.org/10.1364/JOCN.8.000162

Lin, Z., Yin, M., Feygin, S., Sheehan, M., Paiement, J.-F., Pozdnoukhov, A., 2017. Deep Generative Models of Urban Mobility. ACM SIGKDD Conf. 9. https://doi.org/10.475/123

Lindveld, C.D.R., 2003. Dynamic O-D matrix estimation: A behavioural approach.

Liu, F., Janssens, D., Cui, J., Wang, Y., Wets, G., Cools, M., 2014. Building a validation measure for activity-based transportation models based on mobile phone data. Expert Syst. Appl. 41, 6174–6189. https://doi.org/10.1016/j.eswa.2014.03.054

Liu, F., Janssens, D., Cui, J., Wets, G., Cools, M., 2015. Characterizing activity sequences using profile Hidden Markov Models. Expert Syst. Appl. 42, 5705–5722. https://doi.org/10.1016/j.eswa.2015.02.057

Liu, F., Janssens, D., Wets, G., Cools, M., 2013. Annotating mobile phone location data with activity purposes using machine learning algorithms. Expert Syst. Appl. 40, 3299–3311. https://doi.org/10.1016/j.eswa.2012.12.100

MacGregor, J.N., Chu, Y., 2011. Human Performance on the Traveling Salesman and Related Problems: A Review. J. Probl. Solving 3, 1. https://doi.org/10.7771/1932-6246.1090

Maerivoet, Sven; De Moor, B.L.R., 2006. Dynamic Traffic Assignment based on Cellular Automata. Month 22, 2002–2006.

Maher, M.J., 1983. Inferences on trip matrices from observations on link volumes: A

Bayesian statistical approach. Transp. Res. Part B 17, 435–447. https://doi.org/10.1016/0191-2615(83)90030-9

Makhorin, A., 2012. Glpk (gnu linear programming kit) [WWW Document]. Free Softw. Found. URL https://www.gnu.org/software/glpk/ (accessed 5.8.20).

Marsden, P. V., 2015. Network Centrality, Measures of, in: International Encyclopedia of the Social & Behavioral Sciences: Second Edition. Elsevier Inc., pp. 532–539. https://doi.org/10.1016/B978-0-08-097086-8.43115-6

Mascetti, S., Freni, D., Bettini, C., Wang, X.S., Jajodia, S., 2011. Privacy in geo-social networks: Proximity notification with untrusted service providers and curious buddies. VLDB J. 20, 541–566. https://doi.org/10.1007/s00778-010-0213-7

McGuckin, N., Murakami, E., 1999. Examining Trip-Chaining Behavior: Comparison of Travel by Men and Women. Transp. Res. Rec. J. Transp. Res. Board 1693, 79–85. https://doi.org/10.3141/1693-12

Mcnally, M.G., Rindt, C., 2008. The Activity-Based Approach, in: Hensher, D.A., Button, K. (Eds.), Handbook of Transport Modelling. Emerald Group Publishing Limited.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equation of state calculations by fast computing machines. J. Chem. Phys. 21, 1087–1092. https://doi.org/10.1063/1.1699114

Michele Conforti, Gérard Cornuéjols, G.Z., 2014. Graduate Texts in Mathematics Integer Programming.

Mir, D.J., Isaacman, S., Caceres, R., Martonosi, M., Wright, R.N., 2013. DP-WHERE: Differentially private modeling of human mobility, in: Proceedings - 2013 IEEE International Conference on Big Data, Big Data 2013. pp. 580–588. https://doi.org/10.1109/BigData.2013.6691626

Mladenovic, M.N., Trifunovic, A., 2014. The Shortcomings of the Conventional Four Step Travel Demand Forecasting Process. J. Road Traffic Eng. 60, 5–12.

Mokhtarian, P.L., Salomon, I., 2001. How derived is the demand for travel? Some conceptual and measurement considerations. Transp. Res. Part A Policy Pract. 35, 695–719. https://doi.org/10.1016/S0965-8564(00)00013-6

Molla, M.M., Stone, M.L., Motuba, D., 2017. Developing an activity-based trip generation model for small/medium size planning agencies. Transp. Plan. Technol. 40, 540–555. https://doi.org/10.1080/03081060.2017.1314505

Montero, L., Ros-Roca, X., Herranz, R., Barceló, J., 2019. Fusing mobile phone data with other data sources to generate input OD matrices for transport models, in: Transportation Research Procedia. Elsevier B.V., pp. 417–424. https://doi.org/10.1016/j.trpro.2018.12.211

Morimura, T., Osogami, T., Idé, T., 2013. Solving inverse problem of Markov chain with partial observations, in: Advances in Neural Information Processing Systems. pp. 1655–1663.

Muller, K., 2010. Population synthesis: State of the art.

Nagel, K., Flötteröd, G., 2009. Agent-Based Traffic Assignment: Going From Trips to Behavioral Travelers. Int. Conf. Travel Behav. Res. 1–26.

Newman, M.E.J., 2005. A measure of betweenness centrality based on random walks. Soc. Networks 27, 39–54. https://doi.org/10.1016/j.socnet.2004.11.009

Ni, B., Shen, Q., Xu, J., Qu, H., 2017. Spatio-temporal flow maps for visualizing movement and contact patterns. Vis. Informatics 1, 57–64. https://doi.org/10.1016/j.visinf.2017.01.007

Nie, Y., Zhang, H.M., Recker, W.W., 2005. Inferring origin-destination trip matrices with a decoupled GLS path flow estimator. Transp. Res. Part B Methodol. 39, 497–518. https://doi.org/10.1016/j.trb.2004.07.002

Nourani, Y., Andresen, B., 1998. A comparison of simulated annealing cooling strategies. J. Phys. A. Math. Gen. 31, 8373–8385. https://doi.org/10.1088/0305-4470/31/41/011

Nurul Habib, K.M., 2011. A random utility maximization (RUM) based dynamic activity scheduling model: Application in weekend activity scheduling. Transportation (Amst). 38, 123–151. https://doi.org/10.1007/s11116-010-9294-9

Nurul Habib, K.M., Miller, E.J., 2008. Modelling daily activity program generation considering within-day and day-to-day dynamics in activity-travel behaviour. Transportation (Amst). 35, 467–484. https://doi.org/10.1007/s11116-008-9166-8

Oldham, S., Fulcher, B., Parkes, L., Arnatkevičiūtė, A., Suo, C., Fornito, A., 2019. Consistency and differences between centrality measures across distinct classes of networks. PLoS One 14, e0220061. https://doi.org/10.1371/journal.pone.0220061

Ortúzar, J. de D., Willumsen, L.G., 2011. Modelling Transport, 4th ed, Modelling Transport. Wiley-Blackwell. https://doi.org/10.1002/9781119993308

Page, L., Brin, S., Motwani, R., Winograd, T., 1998. The PageRank Citation Ranking: Bringing Order to the Web. World Wide Web Internet Web Inf. Syst. 54, 1–17. https://doi.org/10.1.1.31.1768

Pan, C., Lu, J., Di, S., Ran, B., 2006. Cellular-based data-extracting method for trip distribution, in: Transportation Research Record. pp. 33–39. https://doi.org/10.3141/1945-04

Pappalardo, L., Simini, F., 2018. Data-driven generation of spatio-temporal routines in human mobility. Data Min. Knowl. Discov. 32, 787–829. https://doi.org/10.1007/s10618-017-0548-4

Parry, K., Hazelton, M.L., 2012. Estimation of origin-destination matrices from link counts and sporadic routing data. Transp. Res. Part B Methodol. 46, 175–188. https://doi.org/10.1016/j.trb.2011.09.009

Pavone, M., 2016. Autonomous mobility-on-demand systems for future urban mobility, in: Autonomous Driving: Technical, Legal and Social Aspects. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 387–404. https://doi.org/10.1007/978-3-662-48847-8_19

Pawlak, J., Circella, G., Mahmassani, H.S., Mokhtarian, P.L., 2019. ICT, Activity Decisions and Travel Choices : 20 years into the Second Millennium and where do we go next? Transp. Res. Board 1–12.

Peeta, S., Ziliaskopoulos, A., 2001. Foundations of Dynamic Traffic Assignment: The Past, the Present and the Future. Networks Spat. Econ. 1, 233–265. https://doi.org/10.1023/A:1012827724856

Pel, A.J., Bliemer, M.C.J., Hoogendoorn, S.P., 2012. A review on travel behaviour modelling in dynamic traffic simulation models for evacuations. Transportation (Amst). https://doi.org/10.1007/s11116-011-9320-6

Pelekis, N., Gkoulalas-Divanis, A., Vodas, M., Kopanaki, D., Theodoridis, Y., 2011.

Privacy-aware querying over sensitive trajectory data, in: International Conference on Information and Knowledge Management, Proceedings. ACM Press, New York, New York, USA, pp. 895–904. https://doi.org/10.1145/2063576.2063706

Pendyala, R.M., Goulias, K.G., 2002. Time use and activity perspectives in travel behavior research. Transportation (Amst). 29, 1–4.

Peterson, A., 2007. The Origin-Destination Matrix Estimation Problem : Analysis and Computations, Linköping Studies in Science and Technology. Dissertations.

Petit, T., Trapp, A.C., 2019. Enriching solutions to combinatorial problems via solution engineering. INFORMS J. Comput. 31, 429–444. https://doi.org/10.1287/ijoc.2018.0855

Phan, D., Xiao, L., Yeh, R., Hanrahan, P., Winograd, T., 2005. Flow map layout. Proc. - IEEE Symp. Inf. Vis. INFO VIS 219–224. https://doi.org/10.1109/INFVIS.2005.1532150

Phithakkitnukoon, S., Horanont, T., Di Lorenzo, G., Shibasaki, R., Ratti, C., 2010. Activity-aware map: Identifying human daily activity pattern using mobile phone data, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 14–25. https://doi.org/10.1007/978-3-642-14715-9_3

Pinho, P., Silva, C., 2015. Mobility patterns and urban structure. Ashgate Publishing, Ltd.

Pinjari, A.R., Bhat, C.R., 2011. Activity-based Travel Demand Analysis, in: de Palma, A., Lindsey, R., Quinet, E. (Eds.), A Handbook of Transport Economics. Edward Elgar Publishing Ltd., pp. 213–248. https://doi.org/https://doi.org/10.4337/9780857930873.00017

Primault, V., Boutet, A., Mokhtar, S. Ben, Brunie, L., 2019. The Long Road to Computational Location Privacy: A Survey. IEEE Commun. Surv. Tutorials 21, 2772–2793. https://doi.org/10.1109/COMST.2018.2873950

Qin, J., Ni, L.L., Shi, F., 2012. Combined simulated annealing algorithm for the discrete facility location problem. Sci. World J. 2012. https://doi.org/10.1100/2012/576392

Rashidi, T.H., Abbasi, A., Maghrebi, M., Hasan, S., Waller, T.S., 2017. Exploring the capacity of social media data for modelling travel behaviour: Opportunities and challenges. Transp. Res. Part C Emerg. Technol. 75, 197–211. https://doi.org/10.1016/j.trc.2016.12.008

Recker, W.W., 2001. A bridge between travel demand modeling and activity-based travel analysis. Transp. Res. Part B Methodol. 35, 481–506. https://doi.org/10.1016/S0191-2615(00)00006-0

Redondo, J.L., Pelegrin, B., Fernandez, P., Garcia, I., Ortigosa, P.M., 2011. Finding multiple global optima for unconstrained discrete location problems. Optim. Methods Softw. 26, 207–224. https://doi.org/10.1080/10556780903567760

Rodrigue, J.-P., Comtois, C., Slack, B., 2017. The geography of transport systems. Routledge.

Ronald, N., Thompson, R., Winter, S., 2015. Simulating Demand-responsive Transportation: A Review of Agent-based Approaches. Transp. Rev. 35, 404–421. https://doi.org/10.1080/01441647.2015.1017749

Saadi, I., Mustafa, A., Teller, J., Cools, M., 2016a. Forecasting travel behavior using Markov Chains-based approaches. Transp. Res. Part C Emerg. Technol. 69, 402–417.

https://doi.org/10.1016/j.trc.2016.06.020

Saadi, I., Mustafa, A., Teller, J., Farooq, B., Cools, M., 2016b. Hidden Markov Model-based population synthesis. Transp. Res. Part B Methodol. 90, 1–21. https://doi.org/10.1016/j.trb.2016.04.007

Sadati, N., Amraee, T., Ranjbar, A.M., 2009. A global Particle Swarm-Based-Simulated Annealing Optimization technique for under-voltage load shedding problem. Appl. Soft Comput. J. 9, 652–657. https://doi.org/10.1016/j.asoc.2008.09.005

Saltzman, M.J., 2002. Coin-Or: An Open-Source Library for Optimization. Springer, Boston, MA, pp. 3–32. https://doi.org/10.1007/978-1-4615-1049-9_1

Santoro, N., Quattrociocchi, W., Flocchini, P., Casteigts, A., Amblard, F., 2011. Time-varying graphs and social network analysis: Temporal indicators and metrics, in: AISB 2011: Social Networks and Multiagent Systems. pp. 33–38.

Schneider, C.M., Belik, V., Couronné, T., Smoreda, Z., González, M.C., 2013. Unravelling daily human mobility motifs. J. R. Soc. Interface 10, 20130246. https://doi.org/10.1098/rsif.2013.0246

Schneider, F., Ton, D., Zomer, L.-B., Daamen, W., Duives, D., Hoogendoorn-Lanser, S., Hoogendoorn, S., 2020. Trip chain complexity: a comparison among latent classes of daily mobility patterns. Transportation (Amst). 1–23. https://doi.org/10.1007/s11116-020-10084-1

Schoenfelder, S., Axhausen, K.W., 2001. Analysing the rhythms of travel using survival analysis, in: Transportation Research Board (TRB) Annual Meeting. https://doi.org/10.3929/ETHZ-A-004241369

Schuster, P., 2000. Taming combinatorial explosion. Proc. Natl. Acad. Sci. U. S. A. https://doi.org/10.1073/pnas.150237097

Sedgewick, R., 2001. Algorithms in C, Addison-Wesley Professional. https://doi.org/10.1016/0965-9978(92)90046-i

Segui-Gasco, P., Ballis, H., Parisi, V., Kelsall, D.G.D.G., North, R.J.R.J., Busquets, D., 2019. Simulating a rich ride-share mobility service using agent-based models. Transportation (Amst). 46, 1–22. https://doi.org/10.1007/s11116-019-10012-y

Shokri, R., Theodorakopoulos, G., Troncoso, C., Hubaux, J.P., Le Boudec, J.Y., 2012. Protecting location privacy: Optimal strategy against localization attacks, in: Proceedings of the ACM Conference on Computer and Communications Security. pp. 617–627. https://doi.org/10.1145/2382196.2382261

Smith, G., Sochor, J., Karlsson, I.C.M.A., 2018. Mobility as a Service: Development scenarios and implications for public transport. Res. Transp. Econ. 1–8. https://doi.org/10.1016/j.retrec.2018.04.001

Song, C., Qu, Z., Blumm, N., Barabási, A.L., 2010. Limits of predictability in human mobility. Science (80-.). 327, 1018–1021. https://doi.org/10.1126/science.1177170

Sörensen, K., 2015. Metaheuristics-the metaphor exposed. Int. Trans. Oper. Res. 22, 3–18. https://doi.org/10.1111/itor.12001

Spiess, H., 1987. A maximum likelihood model for estimating origin-destination matrices. Transp. Res. Part B 21, 395–412. https://doi.org/10.1016/0191-2615(87)90037-3

Srinivasan, K.K., Mahmassani, H.S., 2000. Modeling Inertia and Compliance Mechanisms in Route Choice Behavior Under Real-Time Information. Transp. Res. Rec. J. Transp. Res. Board 1725, 45–53. https://doi.org/10.3141/1725-07

Steg, L., 2003. Can public transport compete with private car? IATSS Res. 27, 27–35. https://doi.org/10.1016/s0386-1112(14)60141-2

Sundaram, S., Koutsopoulos, H.N., Ben-Akiva, M., Antoniou, C., Balakrishna, R., 2011. Simulation-based dynamic traffic assignment for short-term planning applications. Simul. Model. Pract. Theory 19, 450–462. https://doi.org/10.1016/j.simpat.2010.08.004

Suzuki, A., Iwata, M., Arase, Y., Hara, T., Xie, X., Nishio, S., 2010. A user location anonymization method for location based services in a real environment, in: GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems. pp. 398–401. https://doi.org/10.1145/1869790.1869846

Tesselkin, A., Khabarov, V., 2017. Estimation of Origin-Destination Matrices Based on Markov Chains. Procedia Eng. 178, 107–116. https://doi.org/10.1016/j.proeng.2017.01.071

Thill, J. -C, Thomas, I., 1987. Toward Conceptualizing Trip-Chaining Behavior: A Review. Geogr. Anal. 19, 1–17. https://doi.org/10.1111/j.1538-4632.1987.tb00110.x

Tian, P., Ma, J., Zhang, D.M., 1999. Application of the simulated annealing algorithm to the combinatorial optimization problem with permutation property: An investigation of generation mechanism. Eur. J. Oper. Res. 118, 81–94. https://doi.org/10.1016/S0377-2217(98)00308-7

Tolouei, R., Alvarez, P., 2015. Developing and verifying Origin-Destination matrices using mobile phone data : the LLITM case, in: European Transport Conference.

Tolouei, R., Psarras, S., Prince, R., 2017. Origin-Destination Trip Matrix Development: Conventional Methods versus Mobile Phone Data, in: Transportation Research Procedia. Elsevier, pp. 39–52. https://doi.org/10.1016/j.trpro.2017.07.007

Toole, J.L., Colak, S., Sturt, B., Alexander, L.P., Evsukoff, A., Gonzalez, M.C., 2015. The path most traveled: Travel demand estimation using big data resources. Transp. Res. Part C Emerg. Technol. 58, 162–177. https://doi.org/10.1016/j.trc.2015.04.022

Treacy, M.M.J., Rivin, I., Balkovsky, E., Randall, K.H., Foster, M.D., 2004. Enumeration of periodic tetrahedral frameworks. II. Polynodal graphs. Microporous Mesoporous Mater. 74, 121–132. https://doi.org/10.1016/j.micromeso.2004.06.013

Urbanucci, L., 2018. Limits and potentials of Mixed Integer Linear Programming methods for optimization of polygeneration energy systems, in: Energy Procedia. Elsevier Ltd, pp. 1199–1205. https://doi.org/10.1016/j.egypro.2018.08.021

Van Zuylen, H.J., Willumsen, L.G., 1980. The most likely trip matrix estimated from traffic counts. Transp. Res. Part B Methodol. 14, 281–293. https://doi.org/10.1016/0191-2615(80)90008-9

Vlahogianni, E.I., Park, B.B., van Lint, J.W.C.W.C., 2015. Big data in transportation and traffic engineering. Transp. Res. Part C Emerg. Technol. 58, 161. https://doi.org/10.1016/j.trc.2015.08.006

Vogiatzis, C., Pardalos, P.M., 2013. Combinatorial Optimization in Transportation and Logistics Networks, in: Pardalos, P.M., Du, D.-Z., Graham, R.L. (Eds.), Handbook of Combinatorial Optimization. Springer New York, New York, NY, pp. 673–722. https://doi.org/10.1007/978-1-4419-7997-1_63

Von Landesberger, T., Brodkorb, F., Roskosch, P., Andrienko, N., Andrienko, G., Kerren, A., 2016. MobilityGraphs: Visual Analysis of Mass Mobility Dynamics via Spatio-Temporal Graphs and Clustering. IEEE Trans. Vis. Comput. Graph. 22, 11–20.

https://doi.org/10.1109/TVCG.2015.2468111

Wang, C., Lin, M., Zhong, Y., Zhang, H., 2016. Swarm simulated annealing algorithm with knowledge-based sampling for travelling salesman problem. Int. J. Intell. Syst. Technol. Appl. 15, 74–94. https://doi.org/10.1504/IJISTA.2016.076100

Wang, W., Attanucci, J.P., Wilson, N.H.M.M., 2011. Bus Passenger Origin-Destination Estimation and Related Analyses Using Automated Data Collection Systems. J. Public Transp. 14, 131–150. https://doi.org/10.5038/2375-0901.14.4.7

Wang, Y., Yuan, Y., Ma, Y., Wang, G., 2019. Time-Dependent Graphs: Definitions, Applications, and Algorithms. Data Sci. Eng. https://doi.org/10.1007/s41019-019-00105-0

Wang, Y., Ma, X., Liu, Y., Gong, K., Henricakson, K.C., Xu, M., Wang, Yinhai, 2016. A two-stage algorithm for origin-destination matrices estimation considering dynamic dispersion parameter for route choice. PLoS One 11. https://doi.org/10.1371/journal.pone.0146850

Wang, Z., He, S.Y., Leung, Y., 2018. Applying mobile phone data to travel behaviour research: A literature review. Travel Behav. Soc. 11, 141–155. https://doi.org/10.1016/j.tbs.2017.02.005

Wegener, M., 2013. The future of mobility in cities: Challenges for urban modelling. Transp. Policy 29, 275–282. https://doi.org/10.1016/j.tranpol.2012.07.004

Wehmuth, K., Ziviani, A., Fleury, E., 2015. A unifying model for representing time-varying graphs, in: Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/DSAA.2015.7344810

Widhalm, P., Yang, Y., Ulm, M., Athavale, S., González, M.C., 2015. Discovering urban activity patterns in cell phone data. Transportation (Amst). 42, 597–623. https://doi.org/10.1007/s11116-015-9598-x

Willenborg, L., 2019. Complexity and simplification of networks.

Wilson, W.C., 1998. Activity pattern analysis by means of sequence-alignment methods. Environ. Plan. A 30, 1017–1038. https://doi.org/10.1068/a301017

Wise, S., Crooks, A., Batty, M., 2017. Agent Based Modelling of Urban Systems. Int. Work. Agent Based Model. Urban Syst. 10051, 129–148. https://doi.org/10.1007/978-3-319-51957-9

Wood, J., Dykes, J., Slingsby, A., 2010. Visualisation of Origins, Destinations and Flows with OD Maps. Cartogr. J. 47, 117–129. https://doi.org/10.1179/000870410x12658023467367

Yanasse, H.H., 2013. A review of three decades of research on some combinatorial optimization problems. Pesqui. Operacional 33, 11–36. https://doi.org/10.1590/S0101-74382013000100002

Ye, X., Konduri, K.C., Pendyala, R.M., Sana, B., Waddell, P., 2009. Methodology to Match Distributions of Both Household and Person Attributes in Generation of Synthetic Populations. Transp. Res. Board Annu. Meet. 2009 9601, 1–24.

You, T.H., Peng, W.C., Lee, W.C., 2007. Protecting moving trajectories with dummies, in: Proceedings - IEEE International Conference on Mobile Data Management. pp. 278–282. https://doi.org/10.1109/MDM.2007.58

Yue, Y., Lan, T., Yeh, A.G.O., Li, Q.Q., 2014. Zooming into individuals to understand the

collective: A review of trajectory-based travel behaviour studies. Travel Behav. Soc. 1, 69–78. https://doi.org/10.1016/j.tbs.2013.12.002

Zhang, L., Levinson, D., 2004. Agent-based approach to travel demand modeling exploratory analysis, in: Transportation Research Record. pp. 28–36. https://doi.org/10.3141/1898-04

Zhang, W., Thill, J.C., 2017. Detecting and visualizing cohesive activity-travel patterns: A network analysis approach. Comput. Environ. Urban Syst. 66, 117–129. https://doi.org/10.1016/j.compenvurbsys.2017.08.004

Zhao, J., Rahbee, A., Wilson, N.H.M., 2007. Estimating a rail passenger trip origin-destination matrix using automatic data collection systems. Comput. Civ. Infrastruct. Eng. 22, 376–387. https://doi.org/10.1111/j.1467-8667.2007.00494.x

Zhou, F., Mahler, S., Toivonen, H., 2010. Network simplification with minimal loss of connectivity, in: Proceedings - IEEE International Conference on Data Mining, ICDM. pp. 659–668. https://doi.org/10.1109/ICDM.2010.133

Zhou, X., Qin, X., Mahmassani, H.S., 2003. Dynamic Origin-Destination Demand Estimation with Multiday Link Traffic Counts for Planning Applications, in: Transportation Research Record. pp. 30–38. https://doi.org/10.3141/1831-04

Zhu, S., Levinson, D., Zhang, L., 2007. An Agent-based Route Choice Model. Environ. Eng. 1–31.

Zilske, M., Nagel, K., 2015. A simulation-based approach for constructing all-day travel chains from mobile phone data, in: Procedia Computer Science. pp. 468–475. https://doi.org/10.1016/j.procs.2015.05.017

Appendix A

References by Chapter

A.1 References from Chapter 3

The next section contains the references of the Thesis' manuscript for Chapter 3.

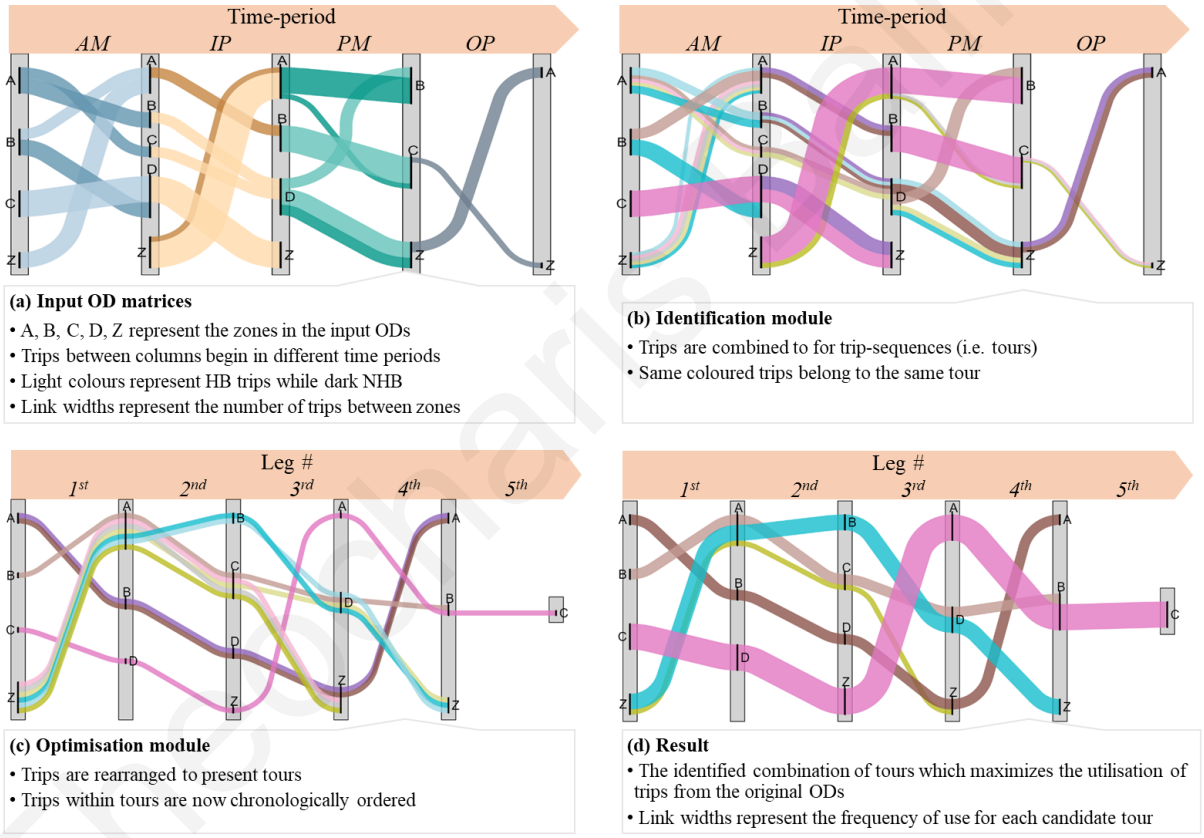

Figure A.1 Visual representation of the suggested methodology.

Table A.1 Example of input Origin-Destination matrix

Origin Zone	Destination Zone	Purpose	Time period	Trips
Z	A	HB	AM	3
Z	A	HB	IP	1
Z	A	HB	OP	2
Z	A	NHB	IP	5
A	B	HB	IP	2
A	B	NHB	AM	3
A	B	NHB	PM	5
A	C	NHB	AM	2
A	C	NHB	PM	1
B	A	HB	AM	2
B	C	HB	PM	5
B	D	NHB	AM	3
B	D	NHB	IP	2
C	D	HB	AM	5
C	D	NHB	IP	2
C	Z	HB	OP	1
D	B	HB	PM	2
D	Z	HB	PM	3
D	Z	NHB	IP	5
D	Z	NHB	PM	2

Purpose: Home-Based (HB), Non-Home-Based (NHB)
Direction: From Home (FH), To Home (TH), Not Applicable (N/A)
Time period: Morning Peak (AM), Inter-Peak (IP), Evening Peak (PM), Off-Peak (OP)
Purpose: Home-Based (HB), Non-Home-Based (NHB)

Table A.2 Example of methodology's output

Schedule id	Activities count	Zones sequence	Trip-purposes	Activities	Time periods	Departure times
1	4	A;B;D;Z	HB;NHB;NHB;HB	H;O;O;H	IP;IP;PM;OP	11:54;12:10;16:09;22:07
2	4	A;B;D;Z	HB;NHB;NHB;HB	H;O;O;H	IP;IP;PM;OP	11:51;12:35;16:57;22:37
3	4	B;A;C;D	HB;NHB;NHB;HB	H;O;O;H	AM;AM;IP;PM	08:20;08:42;10:18;17:14
4	4	B;A;C;D	HB;NHB;NHB;HB	H;O;O;H	AM;AM;IP;PM	08:46;08:22;10:38;17:22
5	5	C;D;Z;A;B	HB;NHB;NHB;NHB;HB	H;O;O;O;H	AM;IP;IP;PM;PM	09:58;10:09;11:04;18:23;8:56
6	5	C;D;Z;A;B	HB;NHB;NHB;NHB;HB	H;O;O;O;H	AM;IP;IP;PM;PM	08:32;11:34;12:08;19:13;20:15
7	5	C;D;Z;A;B	HB;NHB;NHB;NHB;HB	H;O;O;O;H	AM;IP;IP;PM;PM	07:58;10:09;11:09;18:29;18:16
8	5	C;D;Z;A;B	HB;NHB;NHB;NHB;HB	H;O;O;O;H	AM;IP;IP;PM;PM	08:22;10:19;11:41;19:45;19:51
9	5	C;D;Z;A;B	HB;NHB;NHB;NHB;HB	H;O;O;O;H	AM;IP;IP;PM;PM	09:58;10:09;11:45;18:53;19:56
10	4	Z;A;B;D	HB;NHB;NHB;HB	H;O;O;H	AM;AM;AM;PM	08:19;09:04;09:29;18:31
11	4	Z;A;B;D	HB;NHB;NHB;HB	H;O;O;H	AM;AM;AM;PM	08:56;10:07;11:11;18:51
12	4	Z;A;B;D	HB;NHB;NHB;HB	H;O;O;H	AM;AM;AM;PM	08:29;09:54;10:19;19:31
13	3	Z;A;C	HB;NHB;HB	H;O;H	IP;PM;OP	08:55;12:45;18:42;23:21

Purpose: Home-Based (HB), Non-Home-Based (NHB)
Activities: Home (H), Other (O)
Time period: Morning Peak (AM), Inter-Peak (IP), Evening Peak (PM), Off-Peak (OP)

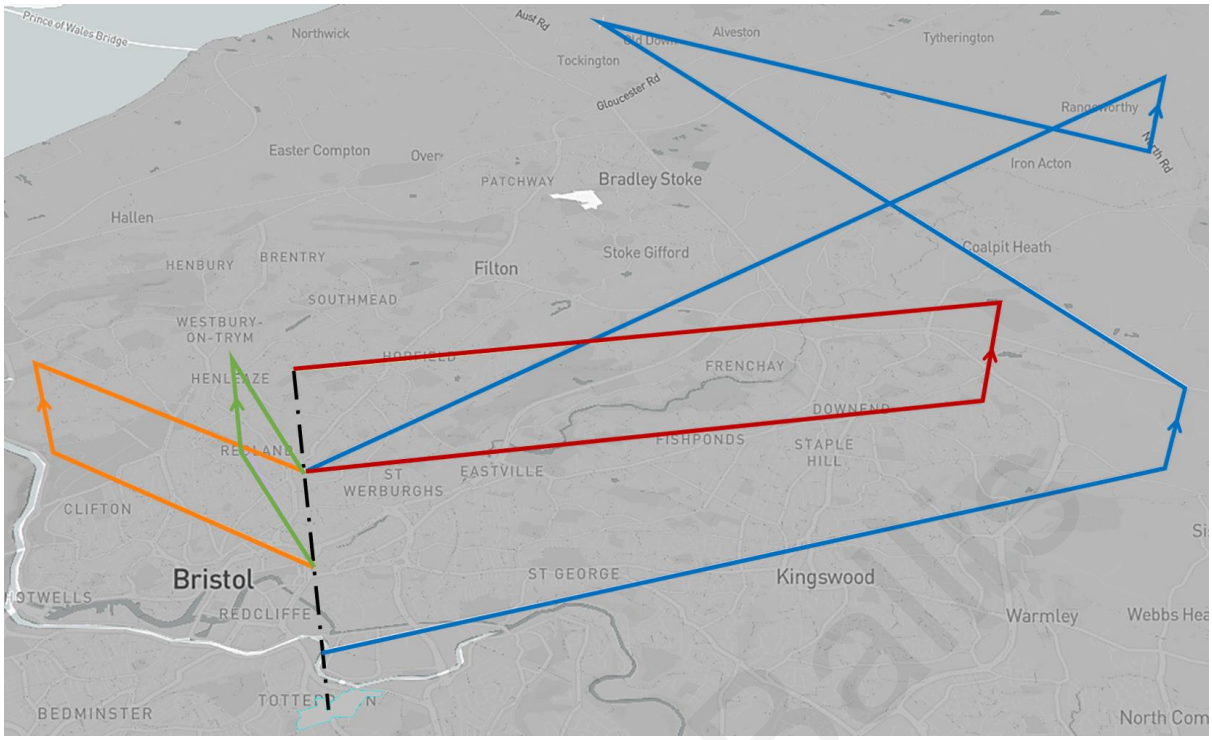

Figure A.2 Visual examples of tours originating from a single zone

A.2 References from Chapter 4

The next section contains the references of the Thesis' manuscript from Chapter 4.

Table A.3 Example of input Origin-Destination matrix

Origin Zone	Destination Zone	Purpose	Time period	Trips
Z	A	HB	AM	3
Z	A	HB	IP	1
Z	A	HB	OP	2
Z	A	NHB	IP	5
A	B	HB	IP	2
A	B	NHB	AM	3
A	B	NHB	PM	5
A	C	NHB	AM	2
A	C	NHB	PM	1
B	A	HB	AM	2
B	C	HB	PM	5
B	D	NHB	AM	3
B	D	NHB	IP	2
C	D	HB	AM	5
C	D	NHB	IP	2
C	Z	HB	OP	1
D	B	HB	PM	2
D	Z	HB	PM	3
D	Z	NHB	IP	5
D	Z	NHB	PM	2

Purpose: Home-Based (HB), Non-Home-Based (NHB)
Direction: From Home (FH), To Home (TH), Not Applicable (N/A)
Time period: Morning Peak (AM), Inter-Peak (IP), Evening Peak (PM), Off-Peak (OP)
Purpose: Home-Based (HB), Non-Home-Based (NHB)

A.3 References from Chapter 7

The next section contains the references of the Thesis' manuscript for Chapter 7

Table A.4 Presentation of activity sequence classification.

Original activity sequence	**Analysis group**	**Frequency**	**Percentage**
H;W;H	H;W;H	16534	67.078%
H;O;H	H;O;H	4883	19.810%
H;W;W;H	H;W;W;H	2012	8.163%
H;O;W;H	H;O;W;H	441	1.789%
H;W;W;W;H	Rest	282	1.144%
H;W;O;H	Rest	271	1.099%
H;W;O;W;H	Rest	75	0.304%
H;W;W;W;W;H	Rest	33	0.134%
H;O;W;W;H	Rest	28	0.114%
H;O;W;O;H	Rest	23	0.093%
H;O;O;H	Rest	19	0.077%
H;W;O;W;W;H	Rest	8	0.032%
H;O;W;W;W;H	Rest	7	0.028%
H;W;O;W;O;H	Rest	7	0.028%
H;W;W;O;H	Rest	7	0.028%
H;W;W;O;W;H	Rest	5	0.020%
H;O;W;O;W;H	Rest	4	0.016%
H;W;O;W;O;W;H	Rest	3	0.012%
H;W;W;W;W;W;H	Rest	3	0.012%
H;O;W;O;W;W;H	Rest	1	0.004%
H;W;O;O;W;H	Rest	1	0.004%
H;W;W;W;W;W;O;H	Rest	1	0.004%
H;W;W;W;W;W;W;H	Rest	1	0.004%

Table A.5 Presentation of time period sequence classification.

Original time period sequence	Analysis group	Frequency
AM;PM	AM;PM	5551
AM;IP2	AM;IP2	1921
IP1;IP2	IP1;IP2	1647
IP2;PM	IP2;PM	1105
AM;IP1	AM;IP1	1102
OP1;PM	OP1;PM	1016
AM;IP3	AM;IP3	998
IP1;PM	IP1;PM	951
IP1;IP1	IP1;IP1	931
IP2;IP2	IP2;IP2	917
PM;IP3	PM;IP3	787
OP1;IP2	OP1;IP2	632
PM;PM	PM;PM	613
IP3;IP3	IP3;IP3	464
AM;AM	Rest	414
IP1;IP3	Rest	385
IP2;IP3	Rest	378
AM;PM;IP3	Rest	272
PM;OP2	Rest	268
IP3;OP2	Rest	266
OP1;IP3	Rest	248
AM;IP2;PM	Rest	234
AM;PM;PM	Rest	207
OP1;OP1	Rest	189
AM;IP1;IP2	Rest	186
AM;IP1;PM	Rest	175
IP2;OP2	Rest	162
AM;OP2	Rest	156
AM;AM;PM	Rest	134
OP1;IP1	Rest	128
IP1;IP2;PM	Rest	123
IP1;OP2	Rest	108
OP1;AM	Rest	80
IP1;IP1;IP2	Rest	77
AM;IP2;IP2	Rest	69
AM;PM;OP2	Rest	63
OP1;IP2;PM	Rest	62
AM;IP2;IP3	Rest	61
OP1;AM;PM	Rest	61
IP1;IP2;IP2	Rest	61
IP1;PM;IP3	Rest	57
AM;IP1;IP1	Rest	54
OP1;PM;IP3	Rest	47
AM;AM;IP2	Rest	47
IP1;IP2;IP3	Rest	44
OP1;PM;PM	Rest	40
IP2;PM;IP3	Rest	37
OP1;AM;IP2	Rest	35
IP1;IP1;PM	Rest	35
AM;AM;IP1	Rest	33
AM;IP1;IP2;PM	Rest	32
IP2;IP2;PM	Rest	31
IP2;PM;PM	Rest	29
AM;IP1;IP3	Rest	28
IP1;PM;PM	Rest	25
PM;IP3;OP2	Rest	23
OP1;IP1;IP2	Rest	23
AM;IP1;IP2;IP2	Rest	23
PM;IP3;IP3	Rest	22
OP1;IP1;PM	Rest	19
PM;PM;IP3	Rest	19
AM;AM;IP3	Rest	19
AM;IP3;OP2	Rest	18
OP1;AM;IP1	Rest	18
AM;IP3;IP3	Rest	17
AM;IP2;IP2;PM	Rest	17
AM;AM;IP1;IP2	Rest	15
OP1;IP2;IP3	Rest	15
AM;IP1;IP1;IP2	Rest	15
OP1;IP2;IP2	Rest	14
AM;IP2;PM;IP3	Rest	14
AM;PM;PM;IP3	Rest	14
IP1;IP2;OP2	Rest	14
AM;AM;IP2;PM	Rest	14
IP1;IP1;IP1	Rest	13
AM;IP2;PM;PM	Rest	13
AM;PM;IP3;OP2	Rest	12
IP1;IP3;OP2	Rest	12
IP2;IP2;IP3	Rest	12
IP1;IP2;IP2;PM	Rest	12
IP2;IP2;IP2	Rest	11
IP1;PM;OP2	Rest	11
IP1;IP1;IP3	Rest	11
AM;AM;PM;IP3	Rest	10
OP1;PM;OP2	Rest	10
IP1;IP1;IP2;PM	Rest	10
IP2;IP3;IP3	Rest	10
AM;IP2;OP2	Rest	10
OP1;IP3;IP3	Rest	9
AM;AM;IP1;PM	Rest	8
OP1;IP1;IP1	Rest	8
AM;IP1;IP1;PM	Rest	8
AM;AM;PM;PM	Rest	8
OP1;AM;IP3	Rest	8
PM;PM;PM	Rest	7
IP2;IP3;OP2	Rest	7
IP1;IP2;PM;PM	Rest	7
AM;IP1;PM;IP3	Rest	7
OP1;AM;IP2;PM	Rest	7
IP1;IP1;IP2;IP2	Rest	7
AM;IP1;PM;PM	Rest	6
AM;PM;IP3;IP3	Rest	6
IP1;IP2;PM;IP3	Rest	6
OP1;AM;IP1;IP2	Rest	6
IP1;IP3;IP3	Rest	6
AM;IP1;IP1;IP2;IP2	Rest	5
AM;AM;OP2	Rest	5
OP1;OP1;IP2	Rest	5
AM;AM;IP2;IP2	Rest	4
IP1;IP1;PM;IP3	Rest	4
AM;IP1;IP2;IP3	Rest	4
AM;AM;IP1;IP2;PM	Rest	4
OP1;PM;PM;IP3	Rest	4
IP3;OP2;OP2	Rest	4
OP1;AM;AM	Rest	4
OP1;IP1;IP2;IP2	Rest	4
OP1;IP1;IP2;IP3	Rest	4
AM;IP2;IP2;IP3	Rest	4
OP1;IP2;IP2;PM	Rest	4
AM;IP2;PM;OP2	Rest	3
AM;PM;PM;PM	Rest	3
AM;IP3;IP3;OP2	Rest	3
AM;IP1;IP2;IP2;PM	Rest	3
AM;IP1;PM;PM;IP3	Rest	3
AM;IP1;IP1;IP1	Rest	3
OP1;AM;PM;PM	Rest	3
OP1;AM;AM;PM	Rest	3
OP1;AM;IP1;IP2;PM	Rest	3
OP1;AM;IP1;PM	Rest	3

OP1;AM;PM;IP3	Rest	3
IP1;IP1;OP2	Rest	3
OP1;AM;AM;IP1	Rest	3
OP1;IP1;OP2	Rest	3
OP1;IP2;OP2	Rest	3
OP1;IP3;OP2	Rest	3
IP1;IP2;IP2;IP3	Rest	3
IP2;IP2;OP2	Rest	3
IP2;IP2;PM;IP3	Rest	3
AM;AM;IP2;PM;PM	Rest	2
AM;AM;IP1;IP1;IP2	Rest	2
IP1;IP2;PM;OP2	Rest	2
IP2;PM;IP3;IP3	Rest	2
IP2;IP2;IP3;OP2	Rest	2
IP2;IP2;IP2;PM;IP3	Rest	2
IP1;PM;PM;IP3	Rest	2
IP1;IP2;IP3;IP3	Rest	2
IP1;IP1;IP1;IP2	Rest	2
IP2;OP2;OP2	Rest	2
IP2;PM;IP3;OP2	Rest	2
AM;OP2;OP2	Rest	2
IP3;IP3;OP2	Rest	2
OP1;OP1;PM	Rest	2
PM;OP2;OP2	Rest	2
OP1;IP2;PM;OP2	Rest	2
OP1;IP1;IP2;PM	Rest	2
OP1;AM;OP2	Rest	2
OP1;AM;IP2;IP2	Rest	2
AM;PM;PM;OP2	Rest	2
OP1;PM;IP3;IP3	Rest	2
AM;IP1;IP1;IP2;PM	Rest	2
AM;IP2;IP2;IP2	Rest	2
AM;IP1;IP2;OP2	Rest	2
AM;IP1;IP1;IP3	Rest	2
AM;IP1;PM;OP2	Rest	2
AM;IP1;IP2;PM;IP3	Rest	2
AM;IP1;IP2;PM;PM;IP3	Rest	2
PM;PM;IP3;OP2	Rest	1
PM;PM;OP2	Rest	1
OP1;PM;PM;OP2	Rest	1
AM;IP1;IP1;IP1;IP2	Rest	1
OP1;IP2;IP3;OP2	Rest	1
OP1;PM;PM;IP3;IP3	Rest	1
AM;IP3;IP3;IP3	Rest	1
AM;PM;PM;PM;IP3	Rest	1
IP1;IP1;IP1;OP2	Rest	1
IP1;IP1;IP1;PM	Rest	1
IP1;IP1;IP2;IP2;PM	Rest	1
IP1;IP1;IP2;IP2;PM;IP3	Rest	1
IP1;IP1;IP2;IP2;PM;PM	Rest	1
IP1;IP1;IP2;IP3	Rest	1
IP1;IP1;IP2;PM;PM	Rest	1
IP1;IP1;PM;OP2	Rest	1
IP1;IP1;PM;PM	Rest	1
IP1;IP1;PM;PM;OP2	Rest	1
IP1;IP2;IP2;IP2	Rest	1
IP1;IP2;IP2;OP2	Rest	1
IP1;IP2;IP2;PM;IP3	Rest	1
IP1;IP2;IP2;PM;PM	Rest	1
IP1;IP2;IP2;PM;PM;PM	Rest	1
IP1;IP2;IP3;OP2	Rest	1
AM;IP2;PM;PM;IP3	Rest	1
OP1;OP1;OP1;AM;IP1	Rest	1
AM;IP2;IP3;IP3	Rest	1
AM;AM;PM;OP2	Rest	1
AM;AM;PM;IP3;IP3	Rest	1
AM;IP1;IP1;IP2;IP3	Rest	1
AM;IP1;IP1;PM;PM	Rest	1
AM;IP1;IP2;IP3;IP3	Rest	1
AM;AM;IP2;IP3	Rest	1
AM;AM;IP2;IP2;PM	Rest	1
AM;IP1;IP2;PM;OP2	Rest	1
AM;IP1;IP2;PM;PM	Rest	1
AM;AM;IP1;IP3	Rest	1
AM;AM;IP1;IP2;IP2	Rest	1
AM;IP1;IP3;OP2	Rest	1
AM;AM;IP1;IP1;OP2	Rest	1
AM;AM;IP1;IP1;IP2;IP2;IP2	Rest	1
AM;IP1;OP2;OP2	Rest	1
AM;AM;AM;PM	Rest	1
AM;AM;AM;IP2	Rest	1
AM;AM;AM;IP1	Rest	1
AM;IP1;PM;PM;IP3;OP2	Rest	1
AM;IP2;IP2;IP2;PM	Rest	1
AM;IP2;IP2;IP3;IP3	Rest	1
AM;IP2;IP2;OP2	Rest	1
AM;IP2;IP2;PM;OP2	Rest	1
IP1;IP2;PM;IP3;OP2	Rest	1
IP1;IP2;PM;PM;IP3	Rest	1
IP1;PM;IP3;IP3	Rest	1
OP1;AM;IP2;IP3	Rest	1
OP1;AM;IP2;PM;IP3	Rest	1
OP1;AM;IP2;PM;OP2	Rest	1
OP1;AM;IP3;IP3	Rest	1
OP1;AM;PM;OP2	Rest	1
OP1;IP1;IP1;IP2	Rest	1
OP1;IP1;IP1;IP2;PM	Rest	1
OP1;IP1;IP1;PM	Rest	1
OP1;IP1;IP2;OP2	Rest	1
OP1;IP1;IP2;PM;PM	Rest	1
OP1;IP1;IP3	Rest	1
OP1;IP1;PM;IP3	Rest	1
OP1;IP2;IP2;IP3	Rest	1
AM;IP1;IP1;IP1;IP2;PM;PM	Rest	1
OP1;IP2;PM;IP3	Rest	1
OP1;IP2;PM;PM	Rest	1
OP1;OP1;AM	Rest	1
OP1;OP1;AM;AM;OP2;OP2	Rest	1
OP1;OP1;IP2;PM	Rest	1
OP1;OP1;OP1	Rest	1
OP1;AM;IP1;IP1;IP3	Rest	1
OP1;AM;IP1;IP1;IP2	Rest	1
IP2;PM;PM;IP3	Rest	1
IP1;PM;PM;OP2	Rest	1
IP2;IP2;IP2;PM	Rest	1
IP2;IP2;PM;OP2	Rest	1
IP2;IP2;PM;PM	Rest	1
IP2;IP3;IP3;OP2	Rest	1
OP1;AM;AM;IP2;PM	Rest	1
IP2;PM;PM;PM;IP3	Rest	1
IP3;IP3;IP3	Rest	1
OP1;AM;AM;IP1;IP1	Rest	1
OP1;AM;AM;IP1;IP2	Rest	1
OP1;AM;AM;IP1;IP3	Rest	1
OP1;AM;AM;IP1;PM	Rest	1
OP1;AM;AM;IP2	Rest	1
IP1;IP1;IP2;IP2;IP2	Rest	1

Appendix B

Developed code

The completion of the presented Ph. D. required the development of a significant amount of code implemented in the Python programming language. The methodological framework is fully automated and parametrizable from a simple configuration file. The codebase of the framework exceeds 2,000 lines (including comments). The following section presents the most crucial segments of the code enabling the execution of the previously presented methodology.

B.1.1 Identification of all paths under threshold constraints

The efficient search of the all the available paths between two nodes in a graph was completed with a modified version of the algorithm presented by Johnson (1975). In particular, the Johnson's algorithm was modified to limit the search space by excluding areas of the graph were user-defined maximum costs have been exceeded. The below presented modified code, extends the widely used python programming library *networkX* (Hagberg et al., 2008). In addition, the modification can be retrieved from the GitHub repository at the web address:

https://github.com/harisbal/networkx/blob/cutoffs/networkx/algorithms/simple_paths.py

```
def _is_path_under_cutoff(G, path, cutoff):

    cutoff_cp = cutoff.copy()
    cost = dict.fromkeys(cutoff.keys(), 0)

    if None in cutoff_cp:
        cost[None] = len(path)
        cutoff_cp.pop(None)

    for u, v in pairwise(path):
        if G.is_multigraph():
            for w in cutoff_cp:
                cost[w] += min([k.get(w, 1) for k in G[u][v].values()])
        else:
            for w in cutoff_cp:
                cost[w] += G[u][v].get(w, 1)

    for w, c in cutoff.items():
        if cost[w] > c:
            return False

    return True

def _all_simple_paths_under_cutoff(G, source, targets, cutoff):

    is_multigraph = G.is_multigraph()
    visited = collections.OrderedDict.fromkeys([source])
    stack = [iter(G[source])]

    while stack:
        children = stack[-1]
        child = next(children, None)
        if child is None:
            stack.pop()
            visited.popitem()
        elif _is_path_under_cutoff(G, list(visited), cutoff):
            if child in visited:
                continue
            if child in targets:
                if _is_path_under_cutoff(G, list(visited) + [child], cutoff):
                    yield list(visited) + [child]
            visited[child] = None
            if targets - set(visited.keys()):  # expand stack until find all
targets
                if is_multigraph:
                    stack.append((v for u, v in G.edges(child)))
                else:
                    stack.append(iter(G[child]))
            else:
                visited.popitem()  # maybe other ways to child
        else:
            stack.pop()
            visited.popitem()
```

B.1.2 Conversion of ODs to a hybrid Time Varying Graph (hTVG)

```
import networkx as nx

def od_to_graph(od, src_col, tgt_col, attr_cols,
                ordered_tps=None, costs=None,
                zones_info=None, verbose=True):

    def _add_chron_connectors(_g, _ordered_tps):

        nx.set_edge_attributes(_g, False, 'Connector')

        # dict with all the timeperiods a link starts/ends from/to each node
        zone_dts = dict()
        for n, z in _g.nodes(data='Zone'):
            dt = _g.nodes[n]['Timeperiod']
            zone_dts.setdefault(z, set()).update([dt])

        for z, zdt in zone_dts.items():
            zdt = list(zdt)
            zdt.sort(key=lambda tp: _ordered_tps.index(tp))

            for src_dt, tgt_dt in utils.pairwise(zdt):
                # create the connections for all the subsequent timeperiods
                src_name = '{}-{}'.format(z, src_dt)
                tgt_name = '{}-{}'.format(z, tgt_dt)

                _d = dict()
                if costs is not None:
                    _d = dict(zip(costs.columns,
                                  len(costs.columns) * [0]))

                # Time-periods between nodes
                height_diff = ordered_tps.index(tgt_dt) - ordered_tps.index(
src_dt)
                _g.add_edge(src_name, tgt_name,
                            Connector=True,
                            Direction='Connector',
                            HeightDiff=height_diff,
                            Length=0, **_d)

        _g.remove_nodes_from(list(nx.isolates(_g)))
        return _g

    od = od.reset_index()
    od = od[[src_col, tgt_col] + attr_cols]

    # Split the nodes of the networks in levels according to the time period
of departure
    if ordered_tps is not None:
        od[src_col] = od[src_col].astype(str) + '-' + od['Timeperiod'].astype
(str)
        od[tgt_col] = od[tgt_col].astype(str) + '-' + od['Timeperiod'].astype
(str)

    g = nx.from_pandas_edgelist(od.reset_index(),
                                src_col, tgt_col,
                                attr_cols,
                                nx.MultiDiGraph())

    # add an artificial length for edges (connectors will have 0)
    nx.set_edge_attributes(g, 1, 'Length')
```

```
57      # Number of tps (height) between connected nodes
58      nx.set_edge_attributes(g, 0, 'HeightDiff')
59
60      if ordered_tps:
61          # Set the node attributes
62          d = {n: {'Zone': n.split('-', 1)[0],
63                   'Timeperiod': n.split('-', 1)[1]} for n in g.nodes}
64          nx.set_node_attributes(g, d)
65          g = _add_chron_connectors(g, ordered_tps)
66      else:
67          # Set the node attributes
68          d = {n: {'Zone': n} for n in g.nodes}
69          nx.set_node_attributes(g, d)
70
71      if zones_info is not None:
72          # add the extra zones info
73          d_attrs = dict()
74          d_info = zones_info.to_dict(orient='index')
75          for n in g.nodes:
76              zone = g.nodes[n]['Zone']
77              d_attrs[n] = d_info[zone]
78          nx.set_node_attributes(g, d_attrs)
79
80      # assure that nodes are unique
81      names = g.nodes.keys()
82      if len(names) != len(set(names)):
83          raise ValueError('Duplicate node names in graph')
84
85      if verbose:
86          print('Network density: {:.2%}'.format(nx.density(g)))
87
88      return g
```

B.1.3 Identification of candidate tours within hTVGs

```
def identify_tours(g, src, direction_names, ordered_tps,
                   max_legs, max_costs,
                   simplify_net_method, simplify_net_place,
                   excl_edges_od_perc):

    srcs = [n for n, z in g.nodes(data='Zone') if z == src]
    srcs.sort(key=lambda z: ordered_tps.index(z.split('-', 1)[1]))
    tgts = srcs.copy()

    tours = []
    g_hb_nhb = filters.prepare_graph(g, srcs, tgts, direction_names)
    g_hb_nhb.remove_nodes_from(list(nx.isolates(g_hb_nhb)))
    #  nx.write_gpickle(g_hb_nhb, './nets/gf-net-{}.pkl'.format(src))

    # Some of the srcs (or tgts) may became isolates due to filtering
    srcs = [src for src in srcs if g_hb_nhb.has_node(src)]
    tgts = srcs.copy()

    # keep only the nhb trips
    g_nhb = filters.filter_edges_by_direction(g_hb_nhb, direction_names.
non_home, keep_connectors=True)

    if (simplify_net_place == 'iterative') and (excl_edges_od_perc > 0):
        simplify_net_weight = 'Dailytrips'
        cm = calculate_centralities(g_nhb, simplify_net_method,
simplify_net_weight)
        nx.set_node_attributes(g_nhb, cm, 'Centrality')
        keep_nodes = srcs
        g_nhb = simplify_net(g_nhb, keep_nodes=keep_nodes, weight=
simplify_net_weight,
                             excl_weight_perc=excl_edges_od_perc, verbose=
False)

    # TODO removing isolates from g_nhb fails the tests
    # g_nhb.remove_nodes_from(list(nx.isolates(g_nhb)))

    # The resulting filtered graphs may still be multigraph when multiple
types of NHB trips exist
    # we only need to ensure that the multigraph edges will collapse to edges
 with the maximum distance
    # (the rest of the attributes are irrelevant)
    if max_costs:
        attrs = max_costs.keys()
        aggr_funcs = dict(zip(attrs, len(attrs) * [max]))
        gb = multigraph_to_graph(g_hb_nhb, aggr_funcs)
        # gf = multigraph_to_graph(g_nhb, aggr_funcs)
    else:
        gb = nx.DiGraph(g_hb_nhb)

    gf = nx.DiGraph(g_nhb)
    for src in srcs:
        successors = set(gb.successors(src))
        for succ in successors:
            predecessors = set()
            for tgt in tgts:
                preds = list(gb.predecessors(tgt))
                if succ in preds:
                    tours.append([src, succ, tgt])
                    preds.remove(succ)
                # If succ in predecessors all_simple_paths returns []
```

```
55                  predecessors.update(preds)
56
57              weights = []
58              cutoffs = []
59              if max_costs:
60                  for w, mc in list(max_costs.items()):
61                      weights.append(w)
62                      cost_src_to_succ = gb[src][succ][w]
63                      costs_pred_to_tgt = []
64
65                      # I'm searching for multiple tgts so I am not sure about
    the distance of the
66                      # predecessor to the tgt. Nonetheless, I can reduce it
   by the minimum
67                      for tgt in tgts:
68                          for pred in gb.predecessors(tgt):
69                              cost_pred_to_tgt = gb[pred][tgt][w]
70                              if cost_src_to_succ + cost_pred_to_tgt > mc:
71                                  # no need to search for this predecessor
72                                  predecessors.discard(pred)
73                              else:
74                                  costs_pred_to_tgt.append(cost_pred_to_tgt)
75
76                      if costs_pred_to_tgt:
77                          min_cost_pred_to_tgt = min(costs_pred_to_tgt)
78                      else:
79                          min_cost_pred_to_tgt = 0
80
81                      min_cost_firstlast_legs = cost_src_to_succ +
   min_cost_pred_to_tgt
82                      cutoffs.append(max(mc - min_cost_firstlast_legs, 0))
83
84              # I have noticed that the distance related cutoffs have stronger
    influence than the rest
85              # therefore I prioritise them in the list
86              tp_min = min([ordered_tps.index(g.nodes[src]['Timeperiod']) for
   src in srcs])
87              tp_max = max([ordered_tps.index(g.nodes[tgt]['Timeperiod']) for
   tgt in tgts])
88
89              weights.extend(['Length', 'HeightDiff'])  # Number of
   intermediate time-periods, Length excludes connectors
90              cutoffs.extend([max_legs - 2, tp_max - tp_min])  # Two legs are
   devoted to successor and predecessor
91
92              # Convert to dict
93              cutoffs = dict(zip(weights, cutoffs))
94              if succ in gf:
95                  zss = list(nx.all_simple_paths(gf, succ, predecessors,
   cutoff=cutoffs))
96              else:
97                  zss = []
98
99              if zss:
100                 # remove the src suffix
101                 for zs in zss:
102                     # prepend the src
103                     zs.insert(0, src)
104                     # the tgt depends on the predecessor last zone of zs (zs
    [-1])

105                     for tgt in tgts:
106                         if zs[-1] in list(gb.predecessors(tgt)):
107                             zs.append(tgt)
108                             break
109
110                 tours.extend(zss)
111
112             # TODO I still need to include loops
113             # if loops:
114             #    zss = _include_loops(zss)
115
116         tgts.remove(src)
117
118     # print('len zones_seqs', len(zones_seqs))
119     return tours
```

B.1.4 Simplification of hTVG based on centrality measures

```
def simplify_net(g, keep_nodes=None,
                 weight=None, excl_weight_perc=0.01,
                 direction_names=None,
                 verbose=True):

    def _remove_edges_by_excl_weight_perc(_g, centralities, weight, perc):
        edges_to_rmv = []
        centralities = centralities.sort_values()
        weight_init = sum(nx.get_edge_attributes(_g, weight).values())

        gc = _g.copy()

        # nx.set_node_attributes(g, False, name='Simplified')
        nx.set_edge_attributes(g, False, name='Simplified')

        excl_weight = 0
        for k, _ in centralities.iteritems():
            rmvs = []
            outs = set(gc.out_edges(k, data='Connector'))
            # Ensure that connectors will not be removed
            ins = set(gc.in_edges(k, data='Connector'))
            for u, v, connector in (ins | outs):
                if not connector:
                    if direction_names is not None:
                        if gc.is_multigraph():
                            for key in gc[u][v]:
                                if gc[u][v][key]['Direction'] ==
direction_names.non_home:
                                    excl_weight += gc[u][v][key][weight]
                                    rmvs.append((u, v, key))
                        else:
                            excl_weight += gc[u][v][weight]
                            rmvs.append((u, v))
                    else:
                        excl_weight += gc[u][v][weight]
                        rmvs.append((u, v))

            if (excl_weight / weight_init) > perc:
                if _g.is_multigraph():
                    for u, v, key in edges_to_rmv:
                        nx.set_node_attributes(_g, {u: {'Simplified': True},
                                                    v: {'Simplified': True}})
                        _g[u][v][key]['Simplified'] = True
                else:
                    for u, v in edges_to_rmv:
                        nx.set_node_attributes(_g, {u: {'Simplified': True},
                                                    v: {'Simplified': True}})
                        _g[u][v]['Simplified'] = True

                _g.remove_edges_from(edges_to_rmv)
                return _g
            else:
                edges_to_rmv.extend(rmvs)

        return _g

    if keep_nodes is None:
        keep_nodes = []

    if excl_weight_perc == 0:
```

```
60          return g
61
62      if verbose:
63          print('nodes before simplification', g.number_of_nodes())
64          print('dailytrips before simplification', sum(nx.get_edge_attributes
    (g, weight).values()))
65
66      if 0 < excl_weight_perc <= 1:
67          centralities = (pd.Series(nx.get_node_attributes(g, 'Centrality')).
    sort_values()
68                          .drop(set(keep_nodes), errors='ignore'))
69          g = _remove_edges_by_excl_weight_perc(g, centralities, weight,
    excl_weight_perc)
70      else:
71          raise ValueError('excl_perc should be between 0 and 1')
72
73      # Remove the isolates
74      g.remove_nodes_from(list(nx.isolates(g)))
75
76      if verbose:
77          print('nodes after simplification', g.number_of_nodes())
78          print('Updated Network density: {:.2%}'.format(nx.density(g)))
79          print('{} after simplification'.format(weight), sum(nx.
    get_edge_attributes(g, weight).values()))
80
81      return g
```

B.1.5 Optimisation module

```
def prepare_model(info, od, ntrs, tgt_distr,
                  distr_constr_enabled, distr_perc_tolerance,
                  mip_tolerances_lowercutoff, init_sol=None):

    # TODO for some reason it passes 2 args
    def _init_descr_by_odpair(m, _):
        # Break the tuples to columns
        df = pd.DataFrame(m.info['Odpairids'].values.tolist(),
                          index=m.info.index).reset_index()

        cols = list(df.columns)
        cols.remove('Tourid')
        dfs = []
        for c in cols:
            s = df[['Tourid', c]]
            s.columns = ['Tourid', 'Odpairid']
            s = s.dropna()
            s = s.astype(int)
            dfs.append(s)

        df = pd.concat(dfs)
        df = df.groupby('Odpairid')['Tourid'].apply(set).apply(list)
        return df.to_dict()

    def _init_tr_by_distr_grps(m):
        _info = m.info.reset_index()
        d_distr_grps = (utils.aggr_tolist(_info, 'Distrgroupid', 'Tourid')
                             .set_index('Distrgroupid').squeeze()
                             .to_dict())
        return d_distr_grps

    def _init_solution(m, i):
        if m.init_sol is None:
            return 0
        else:
            return m.init_sol.get(i, 0)

    def _find_ubnds(m):
        odvals = m.od.set_index('Odpairid')['Dailytrips'].values

        s = m.info['Odpairids']
        idxs = s.index.get_level_values(0).values
        vals = s.values
        mins = [odvals[list(sv)].min() for sv in vals]

        ubnds = dict(zip(idxs, mins))
        return ubnds

    def _get_bnds(m, i):
        return (0, m.tr_ubnds[i])

    model = ConcreteModel()
    model.od = od.copy()
    d_leg_ntrips = model.od.reset_index().set_index('Odpairid')['Dailytrips'
].to_dict()

    model.info = info
    model.init_sol = init_sol
    model.distr_perc_tolerance = distr_perc_tolerance

```

```
    # get the required precision
    if tgt_distr is not None:
        prec = len(str(distr_perc_tolerance).split('.')[1])
        model.tgt_distr = tgt_distr.round(prec)

    model.odpair_ids = Set(initialize=model.od['Odpairid'].unique().tolist
())
    model.tr_ids = Set(initialize=model.info.index.get_level_values('Tourid'
).values.tolist())

    model.trips = Param(model.odpair_ids, initialize=d_leg_ntrips, within=
NonNegativeIntegers)
    model.descr_by_odpair = Param(model.odpair_ids, initialize=
_init_descr_by_odpair, default=None,
                                  within=AnyWithNone)

    model.tr_ubnds = _find_ubnds(model)
    model.ntrs = Param(default=ntrs, mutable=True)

    model.trs_util = Var(model.tr_ids, within=NonNegativeIntegers, bounds=
_get_bnds,
                         initialize=_init_solution)

    # The cutoff should take into account the excluded trips (in low
frequency groups)
    if mip_tolerances_lowercutoff is not None:
        model.cutoff = -mip_tolerances_lowercutoff * model.od['Dailytrips'].
sum()

    if distr_constr_enabled:
        model.distr_grp_ids = Set(initialize=model.tgt_distr.reset_index()['
Distrgroupid'].unique().tolist())
        model.tgt_distr = model.tgt_distr.reset_index().set_index('
Distrgroupid')

        d_init = model.tgt_distr['Percentage'].to_dict()
        model.distr_grp_percs = Param(model.distr_grp_ids, initialize=d_init
, default=0)

        d_init = _init_tr_by_distr_grps(model)
        model.tr_by_distr_grps = Param(model.distr_grp_ids, initialize=
d_init, default=None)

    # TempfileManager.tempdir = r'./tmp'

    def maximise_used_trips(m):
        odsum = m.od['Dailytrips'].sum()
        nlegs = m.info['Nolegs'].to_dict()
        return quicksum(m.trs_util[tr_id] * nlegs[tr_id] for tr_id in m.
tr_ids) - odsum
    model.objective = Objective(rule=maximise_used_trips, sense=maximize)

    def trips_rule(m, odpair_id):
        if m.descr_by_odpair[odpair_id]:
            e = quicksum(m.trs_util[tr_id] for tr_id in m.descr_by_odpair[
odpair_id])
            return e <= model.trips[odpair_id]
        else:
            return Constraint.Skip
    model.od_trips_rule = Constraint(model.odpair_ids, rule=trips_rule)
```

```
107
108     def distr_rule(m, grp_id):
109         if m.tr_by_distr_grps[grp_id]:
110             lower_lim = math.floor((m.distr_grp_percs[grp_id] - m.
    distr_perc_tolerance) * m.ntrs.value)
111             upper_lim = math.ceil((m.distr_grp_percs[grp_id] + m.
    distr_perc_tolerance) * m.ntrs.value)
112             e = quicksum(m.trs_util[tr_id] for tr_id in m.tr_by_distr_grps[
    grp_id])
113             return (lower_lim, e, upper_lim)
114         else:
115             return Constraint.Skip
116
117     if distr_constr_enabled:
118         model.distr_rule = Constraint(model.distr_grp_ids, rule=distr_rule)
119
120     return model
121
122
123 def run_model(model, submit_to_neos=False, workmem=None,
124               mip_strategy_file=2, mip_strategy_probe=3,
    mip_tolerances_lowercutoff=None,
125               timelimit=None, solver_factory='cplex', simplex_display=2):
126
127     opt = SolverFactory(solver_factory)
128
129     if workmem:
130         opt.options['workmem'] = workmem
131
132     opt.options['mip strategy file'] = mip_strategy_file
133     opt.options['mip strategy probe'] = mip_strategy_probe
134     opt.options['simplex display'] = simplex_display
135
136     if mip_tolerances_lowercutoff is not None:
137         opt.options['mip tolerances lowercutoff'] = model.cutoff
138         opt.options['mip limits solutions'] = 1  # Will stop once the first
    feasible solution is achieved
139
140     if timelimit:
141         opt.options['timelimit'] = timelimit
142
143     if submit_to_neos:
144         solver_manager = SolverManagerFactory('neos')
145         results = solver_manager.solve(model, opt=opt)
146         model.solutions.store_to(results)
147     else:
148         try:
149             results = opt.solve(model, tee=True, keepfiles=True)
150         except ApplicationError:
151             opt.set_executable(r'/opt/ibm/ILOG/CPLEX_Studio128/cplex/bin/x86
    -64_linux/cplex')
152             results = opt.solve(model, tee=True, keepfiles=True)
153
154     model.solutions.store_to(results)
155     return results
```

B.1.6 Main program (od2trs)

```
def od2trs(od, filepaths, max_legs=None, direction_names=None, ordered_tps=None,
           tgt_distr=None, distr_dims=None, cost_bins=None,
           costs=None, max_costs=None, purp_to_acts=None,
           excl_edges_od_perc=0,
           simplify_net_method='pagerank', simplify_net_place='begin',
           zones_info=None,
           ncores=None, low_memory=False):

    if costs is not None:
        od = od.join(costs)
        # TODO Drop missing values (could cause problems)
        od = od.dropna(axis=1)

    attr_cols = [c for c in od.reset_index().columns.tolist() if c not in ['Source', 'Target']]
    G = od_to_graph(od, src_col='Source', tgt_col='Target',
                    ordered_tps=ordered_tps, costs=costs,
                    attr_cols=attr_cols, zones_info=zones_info)

    if (simplify_net_place == 'begin') and (excl_edges_od_perc > 0):
        simplify_net_weight = 'Dailytrips'
        cm = nettools.calculate_centralities(G, simplify_net_method, simplify_net_weight)
        nx.set_node_attributes(G, cm, 'Centrality')
        G = nettools.simplify_net(G, weight=simplify_net_weight, excl_weight_perc=excl_edges_od_perc,
                                  direction_names=direction_names)

    # Export the filtered graph
    nx.write_gpickle(G, filepaths['out']['graph'])

    infos = []
    tr_id = 1

    zones = list(set([z for k, z in G.nodes.data('Zone')]))
    zones.sort()

    if isinstance(ncores, int):
        if ncores < 0:
            ncores = cpu_count() + ncores
    elif ncores.f_is_empty():
        ncores = cpu_count()
    else:
        raise ValueError('ncores should be either an integer or null')
    if isinstance(ncores, int):
        if ncores == 0:
            ncores = cpu_count()
        elif ncores < 0:
            ncores = cpu_count() + ncores
        else:
            pass
    elif ncores.f_is_empty():
        ncores = cpu_count() - 1

    print('Running on {} cores'.format(ncores))
    # clean the processing folder
    now = datetime.now().strftime('%Y%m%d%H%M%S')
    filepaths['processfolder'] = os.path.join(r'./tmp/identification', now)
    myio.create_folders([filepaths['processfolder']], overwrite=True)
```

```
57      with Pool(ncores, initializer=init_multiproc_wrapper, initargs=[G]) as
    pool:
58          for trs in tqdm(pool.imap_unordered(
59                          partial(multiproc_wrapper,
60                                  direction_names=direction_names,
61                                  ordered_tps=ordered_tps,
62                                  max_legs=max_legs,
63                                  max_costs=max_costs,
64                                  cost_bins=cost_bins,
65                                  tgt_distr=tgt_distr,
66                                  distr_dims=distr_dims,
67                                  purp_to_acts=purp_to_acts,
68                                  simplify_net_method=simplify_net_method,
69                                  simplify_net_place=simplify_net_place,
70                                  excl_edges_od_perc=excl_edges_od_perc),
    zones), total=len(zones)):
71
72              if trs:
73                  for tr in trs:
74                      tr.id = tr_id
75                      tr_id += 1
76
77                  info_df = trs_to_df(trs, tgt_distr, max_costs)
78
79                  if low_memory:
80                      myio.export_low_memory(info_df, filepaths['processfolder
    '])
81                  else:
82                      infos.append(info_df)
83
84      if low_memory:
85          info_df = myio.combine_interm_results(filepaths['processfolder'],
    purge=True)
86      else:
87          if infos:
88              info_df = pd.concat(infos)
89              if not info_df.index.is_unique:
90                  raise ValueError('Returned index is duplicated')
91          else:
92              print('No trs identified')
93              return None, None
94
95      print('Total trs after filtering: {:,}'.format(info_df.index.unique().
    size))
96
97      return info_df, G
98
99  def multiproc_wrapper(src_zone, direction_names, ordered_tps,
100                        max_legs, max_costs, cost_bins,
101                        tgt_distr, distr_dims,
102                        purp_to_acts,
103                        simplify_net_method, simplify_net_place,
104                        excl_edges_od_perc):
105
106     zones_seqs = nettools.identify_zones_sequences(G, src_zone,
    direction_names, ordered_tps,
107                                                    max_legs, max_costs,
108                                                    simplify_net_method,
    simplify_net_place,
109                                                    excl_edges_od_perc)
```

```
110
111     if not zones_seqs:
112         return None
113
114     if max_costs is not None:
115         cost_types = max_costs.keys()
116     else:
117         cost_types = None
118
119     filt_trs = list()
120     for zs in zones_seqs:
121         trs = zones_seq_to_trs(zs, G, direction_names, cost_types=cost_types
    )
122
123         trs = filters.filter_trs(trs, max_costs, purp_to_acts,
124                                  tgt_distr, distr_dims, cost_bins)
125
126         if trs:
127             # Assign the activities to the filtered trs
128             filt_trs.extend(trs)
129
130     if filt_trs:
131         return filt_trs
132     else:
133         return None
```