

Aggregation and decomposition methods in traffic assignment:

Towards consistent and efficient planning models in a multi-scale environment

Thesis submitted to fulfil requirements for the degree of Doctor of Philosophy

By *Mark Pieter Hendrik Raadsen*

Institute:

Institute of Transport and Logistics Studies (ITLS)

Faculty:

University of Sydney Business School

University:

University of Sydney

Supervision:

Research supervisor: *Michiel C.J. Bliemer*

Co-supervisor: *Michael G.H. Bell*

This is to certify that to the best of my knowledge, the content of this thesis is my own work.
This thesis has not been submitted for any degree or other purposes.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

I also certify that I acknowledge that when my candidature is successful, this thesis will be lodged with the University Librarian and made available for immediate public use.

Mark P.H. Raadsen

I certify that the form of presentation of this thesis is satisfactory, is sufficiently well presented to be examined and does not exceed the prescribed word limit or any extended word limit for which priori approval has been granted.

Professor Michiel C.J. Bliemer

Abstract

Transport models adopt a simplified version of reality to model the movement of people within a transport system. This simplification limits the accuracy of any model. This research focuses on developing novel techniques that, depending on the application context, try to maximise the level of simplification given the minimum result accuracy that is required. To do so, we explore both aggregation and decomposition methods. Besides maximising simplification, we also investigate the requirements to ensure consistency between models that operate in the same spatial domain. In this, so called, multi-scale setting, it is paramount that differences in results between models can be attributed to a particular set of simplifying assumptions. To date, hardly any efforts have been made to formalise, or assess the conditions that need to be satisfied in order to achieve this much desired consistency. The focus of this work is therefore twofold; (i) exploit the combination of both model and application characteristics to achieve the best possible result with the least amount of computational burden, (ii) develop methodology to construct transport model representations in a multi-scale environment following the identified conditions that guarantee consistency between various model granularities.

Contents

Preface.....vi

Notation..... viii

Part I

| | | |
|----------|--|-----------|
| 1 | Introduction..... | 2 |
| 1.1 | Context and Background | 3 |
| 1.2 | Traffic assignment models..... | 5 |
| 1.3 | Multi-scale model environment..... | 8 |
| 1.4 | Representation altering methods | 10 |
| 1.5 | Part I: Altering traffic assignment model representations | 12 |
| 1.6 | Part II: Path induced traffic assignment model decomposition..... | 12 |
| 1.6.1 | Applications of path induced traffic assignment model simplifications..... | 13 |
| 1.7 | Part III: Multi-scale traffic assignment model representation | 13 |
| 1.8 | Case studies | 14 |
| 1.9 | Traffic assignment scope | 16 |
| 1.9.1 | Equilibrium | 16 |
| 1.9.2 | Paths and path choice..... | 16 |
| 1.9.3 | Demand inelasticity and single user class..... | 17 |
| 1.9.4 | Scope Overview | 17 |
| 1.10 | Thesis contributions..... | 18 |
| 1.11 | Thesis outline..... | 19 |
| 1.12 | Remarks on notation..... | 20 |
| 2 | Traffic assignment representation framework | 21 |
| 2.1 | Demand-supply interaction..... | 21 |
| 2.1.1 | Explicit demand-supply interface and path set considerations | 23 |
| 2.2 | General framework | 24 |
| 3 | Traffic assignment | 26 |
| 3.1 | Representations of traffic flow | 26 |
| 3.2 | Fundamental diagram | 28 |
| 3.3 | Traffic assignment procedure classification | 30 |
| 3.4 | Traffic assignment from a temporal perspective | 30 |
| 3.5 | Traffic assignment from a spatial interaction perspective..... | 32 |
| 3.5.1 | Microscopic and mesoscopic traffic flow propagation..... | 32 |
| 3.5.2 | Macroscopic traffic flow propagation..... | 34 |
| 3.6 | Traffic assignment from a behavioural perspective | 36 |

Part II

| | | |
|----------|---|-----------|
| 4 | Link travel time decomposition | 40 |
| 4.1 | The fundamental diagram and delay | 40 |

| | | |
|----------|---|-----------|
| 4.2 | Link travel time in strategic planning models | 41 |
| 4.2.1 | Physical link travel time decomposition | 41 |
| 4.2.2 | Functional link travel time decomposition | 42 |
| 4.3 | Delay at signalised intersections..... | 44 |
| 4.3.1 | Overflow delay..... | 44 |
| 4.3.2 | Uniform delay | 45 |
| 5 | Extracting delay subnetworks under varying demand and fixed supply | 47 |
| 5.1 | Travel time decomposition | 48 |
| 5.1.1 | Path travel time decomposition..... | 49 |
| 5.2 | Network infrastructure..... | 51 |
| 5.3 | Path choice and SUE | 52 |
| 5.3.1 | Accepted turn and link flow..... | 53 |
| 5.4 | Traditional capacity restrained static assignment | 54 |
| 5.5 | Capacity constrained model with point queues | 55 |
| 5.5.1 | Node model inputs and outputs..... | 57 |
| 5.5.2 | Path travel time formulation | 60 |
| 5.5.3 | Path travel time under varying demand | 61 |
| 5.5.4 | The impact of the location of congestion..... | 62 |
| 5.6 | Minimum travel time and delay subnetworks | 64 |
| 5.6.1 | Network loading and travel time: minimum travel time subnetwork | 64 |
| 5.6.2 | Lossless versus lossy decomposition | 64 |
| 5.6.3 | Delay subnetwork: capacity constrained versus capacity restrained | 66 |
| 5.6.4 | Network loading and travel time: the delay subnetwork | 67 |
| 5.7 | Critical-delay path consolidation..... | 72 |
| 5.8 | Varying demand scenarios and the super scenario | 74 |
| 5.8.1 | Alternative super-scenario approaches | 74 |
| 5.8.2 | Incorporating flow margins in the super-scenario | 76 |
| 5.9 | Synthesis and discussion | 77 |
| 6 | Case studies in delay subnetwork construction and calibration | 78 |
| 6.1 | Super scenario calibration: Amsterdam case study | 78 |
| 6.1.1 | Super scenario results without flow margin..... | 80 |
| 6.1.2 | Super scenario results with bandwidth margin | 82 |
| 6.2 | Potential computational gains: Gold Coast case study | 85 |
| 6.3 | Synthesis and discussion | 88 |

Part III

| | | |
|----------|---|-----------|
| 7 | Aggregation methods | 90 |
| 7.1 | Types of (dis)aggregation | 90 |
| 7.2 | Aggregation methods in traffic assignment..... | 92 |
| 7.3 | Zonal (dis)aggregation..... | 92 |
| 7.3.1 | Modifiable area unit problem and zoning effects | 93 |
| 7.3.2 | Zonal scaling..... | 94 |
| 7.4 | Connector representation and centroid placement..... | 95 |
| 7.5 | Network aggregation | 97 |

| | | |
|-----------|---|------------|
| 7.5.1 | Macroscopic fundamental diagrams | 98 |
| 7.6 | Clustering..... | 98 |
| 7.6.1 | Hierarchical and partitional clustering..... | 99 |
| 7.6.2 | Unsupervised Clustering techniques overview | 100 |
| 7.6.3 | Semi-supervised clustering techniques | 101 |
| 7.6.4 | Clustering and optimization..... | 102 |
| 7.7 | Traffic assignment in a multi-scale environment | 103 |
| 7.8 | Summary and Discussion | 103 |
| 8 | On traffic assignment consistency in a multi-scale environment | 105 |
| 8.1 | Traffic assignment conversions in a multi-scale environment | 106 |
| 8.2 | Traffic assignment assumptions revisited..... | 107 |
| 8.2.1 | Spatial assumption subcategories and capabilities | 107 |
| 8.2.2 | Temporal assumption subcategories and capabilities | 108 |
| 8.2.3 | Behavioural assumption subcategories and capabilities | 109 |
| 8.3 | Traffic assignment consistency in a multi-scale environment..... | 109 |
| 8.4 | Summary..... | 112 |
| 9 | Methodology for consistent traffic assignment model inputs | 113 |
| 9.1 | Traffic assignment model input representation | 113 |
| 9.2 | Framework for consistent traffic assignment model input design..... | 114 |
| 9.2.1 | Travel time variance through expected road usage..... | 115 |
| 9.2.2 | Integrated approach..... | 115 |
| 9.2.3 | Conceptual Framework | 116 |
| 9.2.4 | Consistency revisited | 117 |
| 9.2.5 | Notational conventions | 118 |
| 9.3 | Step 1: Constructing the source model | 118 |
| 9.3.1 | Original zoning system and its infrastructure | 118 |
| 9.3.2 | Constructing the source model demand..... | 120 |
| 9.3.3 | Constructing node weights..... | 121 |
| 9.4 | Step 2: Source model assignment..... | 121 |
| 9.5 | Step 3: Supply input representation..... | 123 |
| 9.5.1 | Network and zonal connectivity of paths..... | 124 |
| 9.5.2 | Impact of road usage on disaggregate demand | 126 |
| 9.6 | Step 4: Demand-supply interface representation..... | 127 |
| 9.6.1 | Zone components | 127 |
| 9.6.2 | Traditional connector and centroid design..... | 130 |
| 9.6.3 | Connectoids..... | 130 |
| 9.6.4 | Final network representation..... | 133 |
| 9.6.5 | Connectoid cost..... | 134 |
| 9.6.6 | Connectoid cost: base method | 135 |
| 9.6.7 | Connectoid cost: service area scaling method | 137 |
| 9.6.8 | Connectoid cost: centrality scaling method | 140 |
| 9.7 | Synthesis and discussion | 145 |
| 9.7.1 | Model limitations | 146 |
| 10 | Zonal representation: problem formulation and solution scheme | 147 |

| | | |
|-----------|--|------------|
| 10.1 | Problem formulation..... | 147 |
| 10.2 | Clustering zone components..... | 148 |
| 10.3 | Zonal clustering objective function | 149 |
| 10.3.1 | Cluster based connectoid cost..... | 149 |
| 10.3.2 | Measuring connectoid cost distortion | 151 |
| 10.3.3 | Expected connectoid usage | 152 |
| 10.3.4 | Missing demand penalty | 154 |
| 10.4 | Zoning system criteria and constraints | 155 |
| 10.5 | Instance-level constraints | 156 |
| 10.5.1 | Between component travel time constraint | 156 |
| 10.5.2 | Between component similarity constraint..... | 157 |
| 10.6 | Cluster-level constraints | 158 |
| 10.6.1 | Neighbours..... | 159 |
| 10.6.2 | Laplacian matrix and the number of connected components..... | 161 |
| 10.7 | Solving combinatorial problems..... | 163 |
| 10.7.1 | On clustering complexity..... | 163 |
| 10.8 | Branch-and-bound solution scheme | 164 |
| 10.9 | Spatial cluster bounds..... | 167 |
| 10.9.1 | Single zone component bound..... | 169 |
| 10.9.2 | Zone component reachability bound..... | 170 |
| 10.9.3 | Hybrid bounds..... | 171 |
| 10.10 | Soft constraint multiplier estimation | 171 |
| 10.11 | Network partitioning..... | 172 |
| 10.12 | Constructing the final representation..... | 174 |
| 10.12.1 | Retained boundary nodes..... | 175 |
| 10.12.2 | Final zoning system and demand..... | 175 |
| 10.13 | Synthesis and discussion | 175 |
| 11 | Case studies in consistent traffic assignment representation..... | 177 |
| 11.1 | Amsterdam case study preliminaries..... | 178 |
| 11.1.1 | Comparing results between scenarios..... | 179 |
| 11.1.2 | Conducted case studies | 180 |
| 11.2 | Amsterdam case study I: Basic approach..... | 180 |
| 11.2.1 | Step 1 and 2: creating the Amsterdam source model..... | 180 |
| 11.2.2 | Step 3: creating the Amsterdam supply input representation | 182 |
| 11.2.3 | Step 4: creating the Amsterdam demand-supply interface | 183 |
| 11.2.4 | Step 5: creating the Amsterdam demand input | 184 |
| 11.2.5 | Amsterdam case study I results..... | 187 |
| 11.3 | Amsterdam case study II: post-processing connectoid costs..... | 188 |
| 11.3.1 | Connectoid cost choice based on relative zone component size..... | 188 |
| 11.3.2 | Service area scaling factor | 191 |
| 11.3.3 | Centrality scaling factor | 193 |
| 11.3.4 | Verifying parameter estimates under a different zoning system | 197 |
| 11.4 | Performance..... | 199 |
| 11.5 | Synthesis and discussion | 199 |
| 11.5.1 | Model limitations and extensions | 201 |

Part IV

| | | |
|-----------|---|------------|
| 12 | Conclusions..... | 204 |
| 12.1 | Overview | 204 |
| 12.2 | Conclusions Part II | 205 |
| 12.3 | Possible extensions Part II..... | 205 |
| 12.4 | Conclusions Part III..... | 206 |
| 12.5 | Possible extensions Part III..... | 208 |
| | Bibliography | 210 |
| | Appendix A..... | 225 |
| A.1 | Cluster based service area scaling factor | 225 |
| A.2 | Cluster based centrality scaling factor | 225 |

Preface

The last three years have been very intense, but in a good way. Little did I know what I had gotten myself into when I said yes to enter ITLS's PhD programme. Now that it is complete, I do know and I do not regret any second of it. The only disappointment I do want to mention is that I am now even more aware of just how little I actually know about anything.

Before my PhD candidacy I thought I could write, but really, I couldn't. I also thought my research was the most important research in the world. Surprisingly, it turns out, everyone who researches anything thinks that, and another surprise, most of us are wrong, if not very wrong. Still, I think these past three years have been the best years of my working life. I learnt more than I thought was possible. I visited new places, met new people, worked (and work) with great colleagues, and went to work with a smile on my face every single day. I admit there was the occasional exception, but this was mainly related to a broken Italian coffee machine and had little to do with the actual work.

I will be forever indebted to so many people who supported me and who I want to thank.

Michiel, you used to be my colleague, sometimes client, and – since I moved to Sydney - my main supervisor, as well as my boss. I find it hard to put into words just how much I appreciate all that you have done for me and the things that you taught me. I have never met anyone who is so smart, yet does not judge, and is so helpful and willing to share his knowledge. You can explain difficult concepts so, so, well, and made me realise that, as long as you are willing to invest time and effort, you can master almost anything. Your perspective on things, on work, as well as life in general, have made an everlasting impact on my life. Thank you!

Mike, as my co-supervisor you always had time to provide me with some advice or insight whenever Michiel was unavailable. I profited from your wide network of connections in the transport world, not in the least being able to attend INSTR2018, which I enjoyed a lot. I hope we can keep working together in the future and want to thank you for being part of this journey. By the way, if it wasn't for you, this thesis would not be in matrix notation...

I want to thank all my colleagues at ITLS who I so happily worked with in the last five years. I want to thank David Hensher for letting Michiel hire me in the first place, allowing me to move to the great city of Sydney, and join ITLS in 2012. I want to thank Andrew, Geoffrey, Matt, Rico, and Xiaowen, for putting up with me during lunch and coffee breaks all this time. Thanks guys for all these "important" conversations on the next superhero movie, kitchen hygiene, or how to be compliant with the Australian tax system.

When I just started my PhD, I met Camila, Wen, Collins, Mahbub, and Ines, and through Ines' football (soccer, not rugby) obsession, we quickly bonded in acknowledging how good she was and how poor we were. I want to thank you for the fun times, playing soccer in the Domain, but also the other things we did outside work. It was really great and hope to keep seeing you all, wherever we end up.

I want to thank the guys at Aimsun for supporting my PhD and facilitating the visits to Barcelona. Jordi, Martijn, it was great to get insights in how people actually use the models we develop and the problems you are facing in making them work in practice. I hope we can keep working together because you have an amazing group of talented people.

I also want to thank the people at DAT.Mobility and VLC (Tim, Jamie). They provided me with networks and a platform to program all my algorithms in. Also, they are just great people. Luuk, Erik-Sander, it was great having you here to visit, and Jeroen, I always enjoy working with you on StreamLine. Thanks for everything. I also want to thank Edwin, who I wrote my first (white) papers with, together with Feike, Maarten, and Erik. I might not have ended up where I am now without those first baby-steps!

I would also like to thank the three anonymous examiners for effort in reviewing this dissertation. Their comments helped to improve the final document and I appreciate the time they spent on reading the work.

Pap, mam, zonder jullie was niets van dit alles terecht gekomen. Dankzij jullie kon ik studeren in Enschede, dankzij jullie ben ik naar Brisbane geweest als “arme” student, dankzij jullie ben ik de persoon die ik nu ben, met de fijnste ouders die ik me kan wensen! Koen, Wout, en natuurlijk Mariska en Michelle we zien elkaar te weinig, maar ik denk vaak aan jullie.

Diana, jij bent mijn rots in de branding. Ik zou niet weten wat ik zonder je zou moeten. Het was niet altijd een feest met mij tijdens mijn promotie en toch was je er altijd, ook als ik het even niet zag zitten. De laatste twee jaar waren helemaal speciaal met Jesse erbij. Onze lieve grote vent, die mij altijd weer doet realiseren dat het allemaal eigenlijk niet zo heel belangrijk is allemaal, zolang we maar met zijn drietjes zijn. Wat een geluk!

Notation

- Indices, variables, and constants are denoted in small caps italic font, e.g. a ,
- Maximum values of ranges are denoted in capitals and non-italic, e.g. A ,
- Vectors are denoted in non-italic, small caps, and bold, e.g. \mathbf{a} ,
- Matrices are denoted in non-italic, capitals, and bold, e.g. \mathbf{A} ,
 - Isolating row/column via subscript, e.g. first row via $\mathbf{A}_{1\bullet}$,
- Functions/operations take parameter lists denoted by (\cdot) , e.g. $g(\cdot)$,
- Subscripts denote an entry in a range, e.g. α_a is the a^{th} entry in vector $\boldsymbol{\alpha}$,
- Superscripts denote a flavour, e.g. t^{end} denotes simulation end time
 - when superscript is text; non-italic and small caps

General

| | |
|----------------|---|
| \mathbb{F}_2 | Galois field containing only zero and one |
| \mathbb{R} | Set of real numbers |
| \mathbb{R}_+ | Set of non-negative real numbers |
| \mathbb{Z} | Set of natural numbers |
| \mathbb{Z}_+ | Set of non-negative natural numbers |

Indices

| | |
|------------|------------------------------|
| a | Link |
| n | Node |
| p | Path |
| s | Demand scenario |
| z | Zone |
| \bar{z} | Zone component |
| \hat{z} | Zone component cluster |
| ρ | Equidelay path |
| υ | Zone component partition |
| γ | Representation altering rule |

Ranges and sets

| | |
|-----------------|-----------------------------------|
| A | Number of links |
| N | Number of nodes |
| P | Number of paths |
| \bar{P} | Number of disaggregate paths |
| \mathcal{P} | Number of equidelay paths |
| S | Number of demand scenarios |
| Z | Number of zones |
| \bar{Z} | Number of disaggregate zones |
| $\bar{\bar{Z}}$ | Number of zone components |
| \hat{Z} | Number of zone component clusters |
| Ω | Set of feasible path flow vectors |

Constants

| | | |
|--------------------------|---|---------|
| d^{\min} | Desired minimum demand per zone component cluster | [veh] |
| e^{\max} | Zone component can-link maximum allowed dissimilarity | [-] |
| $q^\Delta, q^{ \Delta }$ | Relative/absolute flow rate margin threshold | [veh/h] |
| q_a^{\max} | Maximum flow rate on link a | [veh/h] |
| t^{end} | Simulation end time | [h] |

| | | |
|---------------------------------|---|--------|
| β^{\min} | Volume-capacity ratio threshold for delay links | [-] |
| χ^{\min} | Lower bound on centrality factor | [-] |
| h^{\max} | Zone component missing trip travel time penalty multiplier | [h] |
| l | Portion of trips accessing zone centre point via ideal quadrant | [-] |
| κ^{\min} | Volume-capacity ratio threshold for keep links | [-] |
| ℓ_a | Length of link a | [km] |
| $g_a^{\max}, g_a^{\text{crit}}$ | Maximum/critical speed on link a | [km/h] |
| τ^{\max} | Maximum travel time allowed between zone components | [h] |
| (ξ_a, ζ_a) | Positive calibration parameter pair for BPR link performance function | [-] |

Variables

| | | |
|--|---|---------|
| l_z^G | Zone component \bar{z} relative size factor conditional on clustering \mathbf{G} | [-] |
| r | Circle radius | [-] |
| δ^-, δ^+ | Network wide connectoid access/egress cost distortion | [veh.h] |
| $\bar{\delta}_z^-, \bar{\delta}_z^+$ | Connectoid access/egress cost distortion for zone component \bar{z} | [veh.h] |
| $\delta^{d^{\min}}$ | Network wide missing trip penalty | [veh.h] |
| $\bar{\delta}_z^{d^{\min}}, \hat{\delta}_z^{d^{\min}}$ | Missing trip penalty for zone component \bar{z} , cluster \hat{z} , respectively. | [veh.h] |
| $\lambda_z^I, \lambda_z^{II}$ | Lower bounds I and II for zone component \bar{z} | [veh.h] |
| ω | Angle in radians | [-] |

Non-indicator vectors

| | | | |
|--|--|---------|-----------------------------------|
| \mathbf{c} | Total link travel times | [h] | $\mathbb{R}_+^{A \times 1}$ |
| $\mathbf{c}^{\min}, \mathbf{c}^{\text{delay}}$ | Minimum and delay component of link travel time | [h] | $\mathbb{R}_+^{A \times 1}$ |
| $\mathbf{c}^{\text{hypo}}, \mathbf{c}^{\text{hyper}}$ | Hypocritical/hypercritical delay component of link travel time | [h] | $\mathbb{R}_+^{A \times 1}$ |
| $\bar{\mathbf{d}}^+, \bar{\mathbf{d}}^-$ | Zone component production/attractions | [veh] | $\mathbb{R}_+^{\bar{Z} \times 1}$ |
| $\hat{\mathbf{d}}^+, \hat{\mathbf{d}}^-$ | Cluster based production/attractions per zone component in cluster | [veh] | $\mathbb{R}_+^{\hat{Z} \times 1}$ |
| $\hat{\mathbf{d}}^{\text{total}}, \hat{\mathbf{d}}^{\text{missing}}$ | Total/missing number of trips per zone component cluster | [veh] | $\mathbb{R}_+^{\hat{Z} \times 1}$ |
| \mathbf{f}^s | Path flows for demand scenario s | [veh/h] | $\mathbb{R}_+^{P \times 1}$ |
| $\bar{\mathbf{f}}$ | Disaggregate path flows | [veh/h] | $\mathbb{R}_+^{P \times 1}$ |
| \mathbf{h} | Path travel times | [h] | $\mathbb{R}_+^{P \times 1}$ |
| $\mathbf{h}^{\min}, \mathbf{h}^{\text{delay}}$ | Minimum and delay component of path travel time | [h] | $\mathbb{R}_+^{P \times 1}$ |
| $\mathbf{h}^{\text{hypo}}, \mathbf{h}^{\text{hyper}}$ | Hypocritical/hypercritical delay component of path travel time | [h] | $\mathbb{R}_+^{P \times 1}$ |
| \mathbf{q} | Link flows | [veh/h] | $\mathbb{R}_+^{A \times 1}$ |
| $\bar{\mathbf{q}}$ | Disaggregate link flows | [veh/h] | $\mathbb{R}_+^{A \times 1}$ |
| $\mathbf{w}^z, \bar{\mathbf{w}}^z, \hat{\mathbf{w}}^z$ | Node weight conditional on membership zone/zone component/cluster | [-] | $\mathbb{R}_+^{N \times 1}$ |
| \mathbf{x}, \mathbf{y} | Node based x and y coordinates | [-] | $\mathbb{R}^{N \times 1}$ |
| $\bar{\mathbf{x}}^{\max}, \hat{\mathbf{x}}^{\max}$ | Zone component/zone component cluster based maximum x coordinates | [-] | $\mathbb{R}^{\bar{Z} \times 1}$ |
| $\bar{\mathbf{x}}^{\min}, \hat{\mathbf{x}}^{\min}$ | Zone component/zone component cluster based minimum x coordinates | [-] | $\mathbb{R}^{\bar{Z} \times 1}$ |
| $\bar{\mathbf{y}}^{\max}, \hat{\mathbf{y}}^{\max}$ | Zone component/zone component cluster based maximum y coordinates | [-] | $\mathbb{R}^{\bar{Z} \times 1}$ |
| $\bar{\mathbf{y}}^{\min}, \hat{\mathbf{y}}^{\min}$ | Zone component/zone component cluster based minimum y coordinates | [-] | $\mathbb{R}^{\bar{Z} \times 1}$ |
| $\bar{\mathbf{x}}, \hat{\mathbf{x}}$ | Zone component/zone component cluster based average x coordinate | [-] | $\mathbb{R}^{\bar{Z} \times 1}$ |

| | | | |
|--------------------------|---|-----|---------------------------------|
| \bar{y}, \hat{y} | Zone component/zone component cluster based average y coordinate | [-] | $\mathbb{R}^{\bar{Z} \times 1}$ |
| \mathbf{a} | Link flow rate reduction factors | [-] | $[0, 1]^{A \times 1}$ |
| \mathbf{a}^n | Link flow rate reduction factors only related to node n | [-] | $[0, 1]^{A \times 1}$ |
| $\boldsymbol{\eta}^{ap}$ | Indicators for links on path p up to but not including link a | [-] | $\mathbb{F}_2^{A \times 1}$ |

Indicator vectors

| | | | |
|--|---|-----|-----------------------------|
| $\mathbf{0}, \mathbf{1}$ | All-zeros/all-ones vector | [-] | context |
| $\boldsymbol{\beta}$ | Delay link indicator vector | [-] | $\mathbb{F}_2^{A \times 1}$ |
| $\boldsymbol{\beta}^\Delta, \boldsymbol{\beta}^{ \Delta }$ | Delay link indicator vector including relative/absolute flow margin | [-] | $\mathbb{F}_2^{A \times 1}$ |

Non-indicator matrices

| | | | |
|--|--|---------|--|
| \mathbf{D}^s | Original trips between zones for a demand scenario s | [veh] | $\mathbb{R}_+^{Z \times Z}$ |
| $\mathbf{D}^{\max}, \mathbf{D}^{\min}, \bar{\mathbf{D}}$ | Maximum/minimum/average trips between zones across demand scenarios | [veh] | $\mathbb{R}_+^{Z \times Z}$ |
| $\bar{\mathbf{D}}^{zz'}$ | Disaggregate node demand departing/arriving from zone z, z' , respectively | [veh] | $\mathbb{R}_+^{\bar{Z} \times \bar{Z}}$ |
| $\bar{\mathbf{E}}$ | Zone component dissimilarity | [-] | $[0, 1]^{\bar{Z} \times \bar{Z}}$ |
| $\bar{\mathbf{F}}^+, \bar{\mathbf{F}}^-$ | Expected trips departing/arriving from zone component connectoids | [veh] | $\mathbb{R}_+^{\bar{Z} \times N}$ |
| $\hat{\mathbf{F}}^+, \hat{\mathbf{F}}^-$ | Expected trips departing/arriving from cluster connectoids | [veh] | $\mathbb{R}_+^{\bar{Z} \times N}$ |
| $\mathbf{H}, \bar{\mathbf{H}}$ | Shortest path cost from zone-to-zone/node-to-node | [h] | $\mathbb{R}_+^{Z \times Z}$ |
| $\bar{\mathbf{H}}^+, \bar{\mathbf{H}}^-$ | Shortest path departure/arrival cost; node-to-zone component connectoid | [h] | $\mathbb{R}_+^{\bar{Z} \times N}$ |
| $\hat{\mathbf{H}}^+, \hat{\mathbf{H}}^-$ | Shortest path departure/arrival cost; node-to-cluster connectoid | [h] | $\mathbb{R}_+^{\bar{Z} \times N}$ |
| $\hat{\mathcal{L}}$ | Laplacian matrix for clustered zone component adjacency | [-] | $\mathbb{Z}^{\bar{Z} \times \bar{Z}}$ |
| \mathbf{Q}^p | Accepted link-to-link turn flows, link to link conditional on path p | [veh/h] | $\mathbb{R}_+^{A \times A}$ |
| \mathcal{R} | Turn receiving flows conditional | [veh/h] | $\mathbb{R}_+^{A \times 1}$ |
| \mathcal{S} | Sending turn flows | [veh/h] | $\mathbb{R}_+^{A \times A}$ |
| $\bar{\mathcal{T}}^{\min}$ | Travel time, zone component to zone component | [h] | $\mathbb{R}_+^{\bar{Z} \times \bar{Z}}$ |
| $\bar{\mathcal{X}}^I, \hat{\mathcal{X}}^I$ | Service area scaling factors for zone component/cluster connectoid costs | [-] | $[0, 1]^{\bar{Z} \times N}$ |
| $\bar{\mathcal{X}}^{II}, \hat{\mathcal{X}}^{II}$ | Centrality scaling factors for zone component/cluster connectoid costs | [-] | $[\lambda^{\min}, 1]^{\bar{Z} \times N}$ |

Indicator matrices

| | | | |
|---|---|-----|---|
| \mathbf{A} | Directed links from-node-to-node | [-] | $\mathbb{F}_2^{N \times N}$ |
| $\mathbf{A}^z, \bar{\mathbf{A}}^{\bar{z}}$ | Directed links from-node-to-node internal to zone z /zone component \bar{z} | [-] | $\mathbb{F}_2^{N \times N}$ |
| $\mathbf{A}^+, \mathbf{A}^-$ | Outgoing/incoming links per node | [-] | $\mathbb{F}_2^{N \times A}$ |
| $\mathbf{A}^{z+}, \mathbf{A}^{z-}$ | Outgoing/incoming links per node related to zone z | [-] | $\mathbb{F}_2^{N \times N}$ |
| $\bar{\mathbf{A}}, \hat{\mathbf{A}}$ | Adjacency indicators for zone components/within cluster zone components | [-] | $\mathbb{F}_2^{\bar{Z} \times \bar{Z}}$ |
| \mathbf{G} | Zone component clustering indicators | [-] | $\mathbb{F}_2^{\bar{Z} \times \bar{Z}}$ |
| \mathbf{I} | Identity matrix | [-] | $\mathbb{F}_2^{? \times ?}$ |
| \mathbf{J} | All-ones matrix | [-] | $\mathbb{F}_2^{? \times ?}$ |
| \mathbf{K} | Zone component can-link constraints | [-] | $\mathbb{F}_2^{\bar{Z} \times \bar{Z}}$ |
| $\mathbf{K}^{\mathcal{V}_v}$ | Zone component can-link constraints per for natural partition \mathcal{V} | [-] | $\mathbb{F}_2^{\bar{Z} \times \bar{Z}}$ |
| $\mathbf{K}^{\lambda_z^I}, \mathbf{K}^{\lambda_z^{II}}$ | Zone component \bar{z} can-link constraints for bounds $\lambda_z^I / \lambda_z^{II}$ | [-] | $\mathbb{F}_2^{\bar{Z} \times \bar{Z}}$ |

| | | | |
|---|--|-----|---|
| \mathbf{K}^G | Cluster-level zone component can-link constraints (non-derived) | [-] | $\mathbb{F}_2^{\tilde{Z} \times \tilde{Z}}$ |
| $\mathcal{K}, \bar{\mathcal{K}}$ | Keep network without/with zone component connectoid infrastructure | [-] | $\mathbb{F}_2^{N \times A}$ |
| $\mathbf{M}^p, \mathbf{M}^{p,\text{delay}}$ | Turns for original/delay subnetwork conditional on path p | [-] | $\mathbb{F}_2^{A \times A}$ |
| \mathbf{N} | Zone to internal node indicators | [-] | $\mathbb{F}_2^{Z \times N}$ |
| $\bar{\mathbf{N}}, \hat{\mathbf{N}}$ | Zone component/zone component cluster to internal node indicators | [-] | $\mathbb{F}_2^{\tilde{Z} \times N}$ |
| $\bar{\mathbf{N}}^+, \bar{\mathbf{N}}^-$ | Zone component egress/access connectoid indicators | [-] | $\mathbb{F}_2^{\tilde{Z} \times N}$ |
| $\hat{\mathbf{N}}^+, \hat{\mathbf{N}}^-$ | Cluster based zone component egress/access connectoid indicators | [-] | $\mathbb{F}_2^{\tilde{Z} \times N}$ |
| \mathbf{N}° | Zone boundary node indicators | [-] | $\mathbb{F}_2^{Z \times N}$ |
| \mathcal{O} | Equidelay paths to original path mapping based on complete overlap | [-] | $\mathbb{F}_2^{P \times P}$ |
| $\mathbf{P}, \mathbf{P}^{\text{delay}}$ | Original/delay path to link indicators | [-] | $\mathbb{F}_2^{P \times A}$ |
| $\mathbf{P}^{\text{equidelay}}$ | Equidelay path to link indicators | [-] | $\mathbb{F}_2^{P \times A}$ |
| $\mathbf{P}^+, \mathbf{P}^-$ | Path to departing/arriving zone mapping | [-] | $\mathbb{F}_2^{Z \times P}$ |
| $\bar{\mathbf{P}}^+, \bar{\mathbf{P}}^-$ | Path to departing/arriving disaggregate zone indicators | [-] | $\mathbb{F}_2^{Z \times P}$ |
| \mathbf{Z} | Zone to centroid node mapping | [-] | $\mathbb{F}_2^{Z \times N}$ |
| Υ | Transitive closure of zone component reachability | [-] | $\mathbb{F}_2^{\tilde{Z} \times \tilde{Z}}$ |

Matrix/vector operations

| | | |
|------------------------|--|---|
| $\text{dim}(\cdot)$ | Vector space dimension | $\text{dim} : \mathbb{Z}_+ \times \mathbb{R}^{\mathbb{Z}_+ \times 1} \rightarrow \mathbb{Z}_+$ |
| $\text{ker}(\cdot)$ | Matrix kernel | $\text{ker} : \mathbb{R}^{\mathbb{Z}_+ \times \mathbb{Z}_+} \rightarrow \mathbb{Z}_+ \times \mathbb{R}^{\mathbb{Z}_+ \times 1}$ |
| $\text{merge}(\cdot)$ | Recursive merge function on two turn indicator vectors | $\text{merge} : \mathbb{F}_2^{A \times 1} \times \mathbb{F}_2^{A \times 1} \rightarrow \mathbb{F}_2^{A \times 1}$ |
| $\text{rref}(\cdot)$ | Reduced row echelon form of matrix | $\text{rref} : \mathbb{R}^{\mathbb{Z} \times \mathbb{Z}} \rightarrow \mathbb{R}^{\mathbb{Z} \times \mathbb{Z}}$ |
| $\text{rk}(\cdot)$ | Matrix rank | $\text{rk} : \mathbb{R}^{\mathbb{Z}_+ \times \mathbb{Z}_+} \rightarrow \mathbb{Z}_+$ |
| \mathbf{T} | Transpose of a matrix | $\mathbf{T} : \mathbb{R}^{\mathbb{Z}_+ \times \mathbb{Z}'_+} \rightarrow \mathbb{R}^{\mathbb{Z}'_+ \times \mathbb{Z}_+}$ |
| $\text{trans}(\cdot)$ | Transitive closure of a matrix | $\text{trans} : \mathbb{R}^{\mathbb{Z}_+ \times \mathbb{Z}_+} \rightarrow \mathbb{R}^{\mathbb{Z}_+ \times \mathbb{Z}_+}$ |
| $\text{unique}(\cdot)$ | Retain unique rows in matrix | $\text{unique} : \mathbb{R}^{\mathbb{Z}_+ \times \mathbb{Z}_+} \rightarrow \mathbb{R}^{\mathbb{Z}_+ \times \mathbb{Z}_+}$ |

Functions

| | | |
|----------------------|--|--|
| $\Phi(\cdot)$ | Network loading function | $\Phi : \mathbf{f} \rightarrow \mathbf{h}$ |
| $\Gamma^n(\cdot)$ | Implicit general first order node model formulation for node n | $\Gamma^n : \mathcal{S}^n \times \mathcal{R}^n \rightarrow \mathbf{a}^n$ |
| $\Xi(\cdot)$ | Abstract general representation method function, context dependent | - |
| $\Psi(\cdot)$ | Path choice representation function | $\Psi : \mathbf{h} \rightarrow \mathbf{f}$ |
| $\varepsilon(\cdot)$ | Abstract error analysis function, context dependent | - |
| $g(\cdot)$ | Zone component clustering constrained optimisation problem | $g : \mathbf{G} \times \mathbf{K}^G \times \mathbf{K} \rightarrow \mathbb{R}_+$ |
| $\zeta(\cdot)$ | Inverse aggregation scaling magnitude function, context dependent | - |
| $\Theta(\cdot)$ | Lens area of two circles | $\Theta : \mathbb{R}_+ \times \mathbb{R}_+ \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$ |

Part I

1 Introduction

Today's transport policy makers frequently rely on forecasts of transport models to make their decisions, propose new legislation, or approve new infrastructure projects. This however, is still a fairly recent trend given that the widespread adoption of transport models only started in the middle of the twentieth century. This adoption of transport models went hand in hand with rapid technological advances making it possible to design, operate and utilise these complicated transport models in a practical setting.

This symbiosis between transport models and available computing power has been a balancing act ever since. Often, the adopted level of complexity in a transport model is based on the time it takes for the model to run. The maximum amount of time deemed acceptable varies, but typically ranges from a few hours up to a number of days. Many practitioners only accept models that can be solved within a single overnight run, i.e. the time between leaving work and turning up the next day. Even so, contemporary transport models for middle-to-large cities can still easily take multiple days to run and this is not likely to change in the near future. To reduce these long run times, aggregation and decomposition methods are a popular approach to, at least in part, address the ever increasing complexity of our transport models.

A more recent development is the adoption of multiple transport models alongside each other. For example, large scale static macroscopic models for transport planning purposes and small scale microscopic simulation models for traffic management. Each of these models has its own specific application area, with the main differentiator being their level of detail in which travel demand, infrastructure supply, and traffic flows are represented. In this specific context, aggregation and disaggregation methods are often adopted to facilitate conversions between various levels of detail. In such a *multi-scale* approach, practitioners and researchers are faced with new challenges, especially in terms of ensuring consistency between each of the adopted modelling scales.

In this thesis we propose novel methodology to alter existing representations of transport models used for planning purposes. This includes, but is not limited to, aggregation, disaggregation and decomposition methods. The results improve upon current methodology by either reducing the computational cost, while having little to no negative effects compared to the original model results, and/or are aimed at maintaining consistency in a multi-scale environment. The thesis is split in four distinct parts. Part I is a general introduction to the subject of transport model representation and proposes a general framework to capture methods that fall within this scope. In Part II of this thesis, we solely focus on methodology aimed at reducing the computational burden of a particular class of transport models. Part III is concerned with the construction of transport model representations at various levels of detail. In this part, our aim is not so much on computational efficiency, but instead develop an integrated framework and methodology suitable for the construction of transport models in a multi-scale environment with a specific focus on consistency. The developed methods allow both practitioners and policy makers to make better informed decisions for a range of compatible applications. Finally, Part IV constitutes the conclusions chapter.

This chapter is outlined as follows; in Section 1.1, a general introduction to transport models, and their representation, is discussed. Section 1.2 discusses the main components of a traffic assignment model which is a particular type of transport model. Section 1.3 introduces the reader to the challenges of traffic assignment representation in a multi-scale environment. Then, we discuss the two main methods used to alter model representations in Section 1.4, i.e. (dis)aggregation and decomposition. This is followed by Sections 1.5, 1.6 and 1.7 that conceptually summarise the ideas and justification for the approaches that are presented Parts I, II and III of this thesis. The case studies used throughout this work are briefly introduced in Section 1.8. The scope of the considered traffic assignment models is outlined in Section 0, followed by a summary of thesis contributions in Section 1.10. The outline of the thesis itself is discussed in Section 1.11 followed by some remarks on the adopted notation in this work in Section 1.12.

1.1 Context and Background

Our focus is on the construction of appropriate transport model *representations*. We choose to define “appropriate” by assessing two objectives. The transport model representation should be both (i) *capable*, and (ii) *minimal*. We deliberately choose the term representation at this stage to prevent any distinction between model procedures, inputs, or methodologies to achieve our objectives.

To quantify how capable a transport model representation is, we measure the amount of *information loss*. Similarly, we quantify how minimal a transport model representation is by measuring the magnitude of simplification, which is also referred to as *scaling*. The term information loss refers to how capable a simplified (scaled) model representation is, when it comes to approximating or reconstructing the results of its original counterpart and it should, for example, not be confused with information loss in physical processes, such as loss of data on a computer hard disk. The term scaling is used in various disciplines as well, to prevent any confusion, we only refer to scaling in the context of aggregation methods, where it signifies, the extent to which model input data (and possibly model procedures) are aggregated, it should not be confused with the concept of scale-free methods used in, for example graph theory.

The magnitude of information loss and scaling – in aggregation - are relative measures and require a reference point to be meaningful. The original transport model representation serves as this reference point in this work. Figure 1.1 shows the schematic relation of a transport model’s original representation, its alternative representations, and how to conceptually assess the impact of any differences between them. Following this line of reasoning, the magnitude scaling is then determined based on the differences between the original model and the alternative model representation. When one constructs multiple alternative model representation consistency between alternatives is an additional consideration that one needs to take into account. One can for example construct two alternative transport model representations that are both regarded capable and minimal given their respective applications, but this does not necessarily qualify them to be considered alongside each other within the same multi-scale environment. In a multi-scale environment, the consistency between the two models is an additional constraint that needs to be satisfied, something that is often overlooked in practice and can lead to serious complications. An example of such a complication is the

situation that different models with different levels of detail adopt identical inputs, but yield very different results. When these differences are the result of inconsistencies between the two models rather than stemming from the difference in granularity it is no longer possible to determine which of the two results can be regarded as more accurate, rendering both model outputs effectively useless.

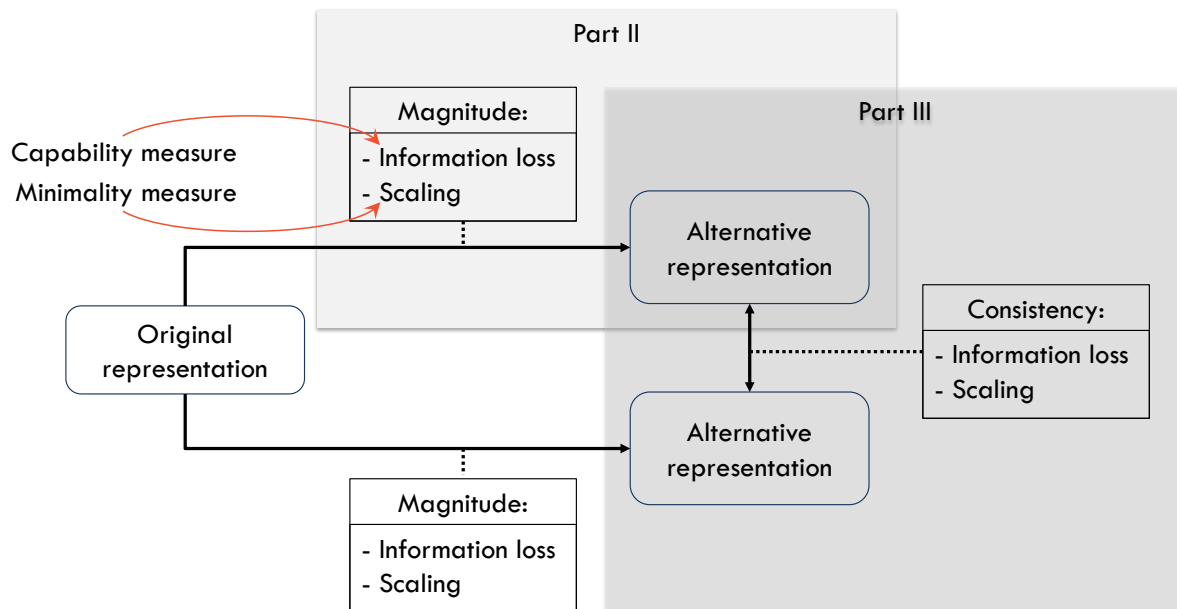


Figure 1.1: Measuring impact of alternative (transport model) representations.

In addition, we argue that, to successfully quantify how capable and minimal an alternative model representation is, the application context needs to be taken into account. To illustrate this consider the following example, suppose that we would like to develop a transport model to determine accurate suburb-to-suburb travel times in Sydney (Australia) for a number of different travel demand scenarios. Let our original transport model be the detailed large-scale model such as the one depicted in Figure 1.2(a). A possible alternative representation is given in Figure 1.2(b).

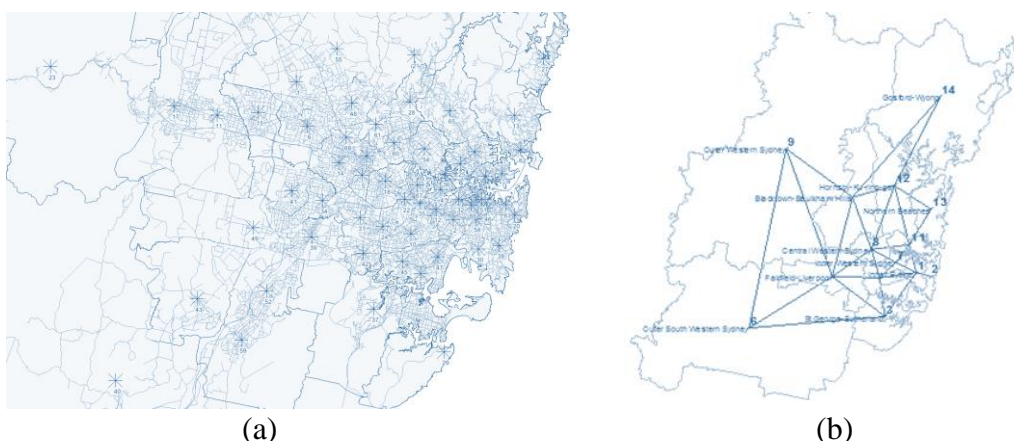


Figure 1.2: Example of traffic model aggregation applied to the Sydney metropolitan area, (a) original large-scale traffic model and (b) simplified version.

Since, our application requires accurate travel times, the measure of information loss should account for this. One could for example assess the differences in travel time between the

original detailed model and the alternative coarser representation. The alternative representation in Figure 1.2(b) is greatly simplified and is therefore likely to suffer significant information loss compared to the original model (since it is likely to yield very different travel times due to oversimplification given its application context). The original model is therefore preferred in terms of capability. On the other hand, capability might not be our only objective, we also might like a result that we can compute quickly. Of course, computation is quick when the model is more simplified, i.e. minimal. Therefore, from a computational perspective, the alternative representation in Figure 1.2(b) is therefore very attractive. In general, we are looking for the best possible trade-off between minimising information loss and maximising simplification, conditional on the application at hand.

That application context matters becomes even more apparent when we change the underlying objective. For example, when our application would measure toll road revenues instead of point-to-point travel times, the accuracy of such travel times become of lesser importance. Then, the model should instead focus on an accurate representation of the amount of traffic around, and on, the toll roads under investigation. This likely would lead to a very different optimal representation. Acknowledging the relation between application context and transport model representation, by embedding application based assumptions in the proposed methodology, is paramount in constructing a capable, yet minimal model. Part II of this thesis demonstrates this by showing that an application centric solution can be optimised to an extent that is simply not possible by using existing, more general purpose, methods.

The other premise of this thesis is that when designing alternative representations of transport models, all relevant aspects should be considered. Traditional methods often focus on a particular component of a transport model and optimise its representation in isolation. A classic example of this is found in network aggregation, where one aims to simplify the road network in a standalone fashion while leaving the rest of the model untouched. This is known to have unwanted side effects (see Chapter 5). We therefore propose a more holistic approach. In Part III, we do not consider just a single model component, but all major transport model inputs, as well as internal components, interactions and underlying (simplifying) assumptions.

In this work we only consider transport models within the well-known *traffic assignment* paradigm. Traffic assignment models are a particular type of transport model and have the benefit of being clearly defined and being widely used in practice.

1.2 Traffic assignment models

Traffic assignment models consist of two main components: A *demand model* and a *supply model*. The demand model is responsible for estimating traffic demand based on some zoning system. This results in trips between zones. Trips can be classified in many ways, for example by time-of-day, means of transport (mode), reason for travelling (purpose), path followed, or a combination of the above. Regardless of the actual classification, all trips need access to the physical road infrastructure. The supply model is responsible for providing the level of service of this infrastructure. The interaction between supply and demand, i.e. network and trips, results in the demand being distributed across eligible paths conditional on the level of service of the network. This process is known as *traffic assignment* or alternatively, demand-supply

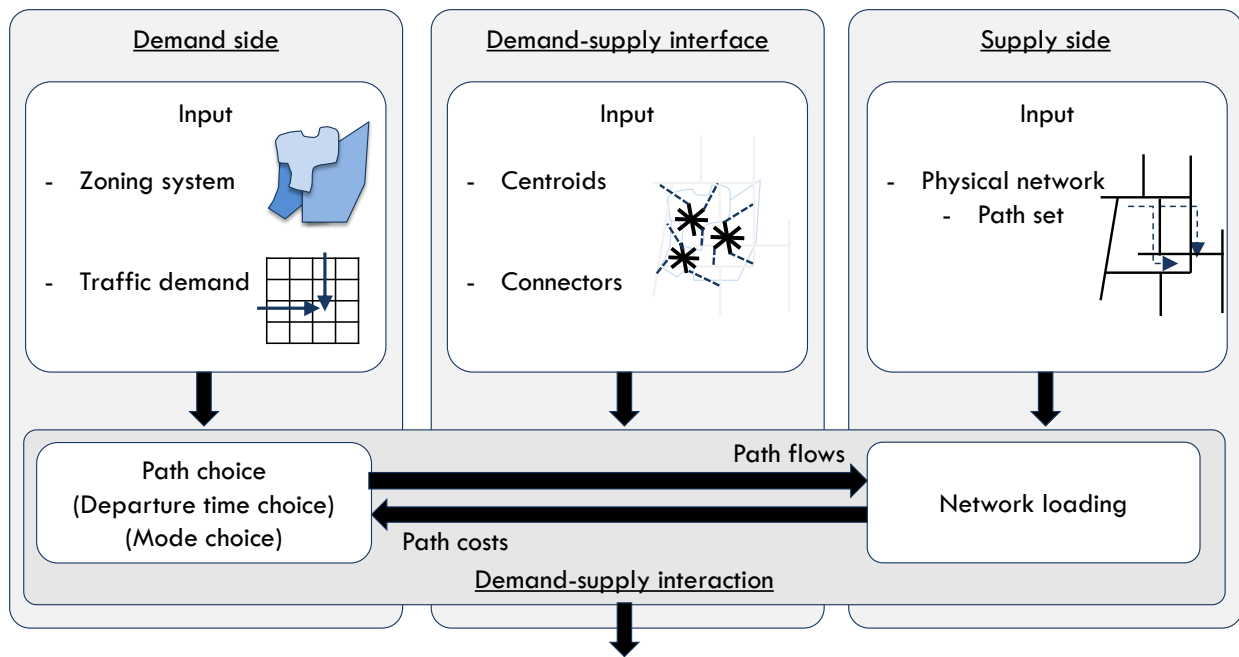
interaction. The procedure that governs the traffic assignment process typically consists of two components; *network loading* and *path choice*.

Network loading is part of the supply side, it loads the given path flows onto the transport network resulting in *generalised costs*. Generalised costs describe the impediment or disutility a traveller attaches to a path and can contain various types of cost such as travel time, tolls, number of traffic lights, or distance. These costs are then used in the path choice model to update path flows. The path choice model is considered part of the demand side and determines the number of trips assigned to each available path conditional on the generalised costs. Apart from path choice, there are many other choices individuals make when it comes to travel demand. These choices can be long term, for example where one lives and works (location choice, destination choice), but also shorter term, for example, when to leave for work (departure time choice) or how to get there (mode choice). In traffic assignment we typically limit ourselves to path choice, possibly extended with *departure time choice* and/or *mode choice*.

The traffic assignment model is considered solved when demand and supply are in *equilibrium*. Conceptually, equilibrium involves finding an/the optimal, preferably unique, point where demand and supply are in perfect balance. While one can argue if a state of equilibrium has any meaning in real life, or even exists in real transport networks, it is deemed essential for transport planning purposes in order to objectively compare current scenarios with future scenarios, due to the lack of a better alternative. The results of a traffic assignment model can take on many forms, common outputs are for example indicators for travel times, congestion levels, toll revenue estimates, accessibility indicators, emission predictions etc.

Figure 1.3 depicts the traffic assignment model, its inputs and its outputs. The demand side inputs comprise the *traffic demand* and the *zoning system*. The zoning system contains the Travel Analysis Zones (TAZs), also referred to as *zones*. Zones serve both as an *origin*, i.e. departure zone, and *destination*, i.e. arrival zone, of trips. A zone constitutes a geographical area that is constructed to be compliant with a number of characteristics, which often include homogeneous land use, homogeneous population characteristics, boundaries based on topographic, political, or census based features, and a relatively constant number of trips (Martinez et al., 2009; Baass, 1981).

The design of the zoning system is a complex process and it is normally assumed to be exogenous and given. The adopted zoning system configuration heavily influences the results of traffic assignment, because trip departure location, trip arrival location, as well as the zone-to-zone demand are all conditional on it. In case the zoning system would be completely disaggregate, then it would contain a separate zone for each household, or person even. In a practical setting though, a zoning system more often adopts a zone per block, suburb, or even village, depending on the granularity or geographical coverage of the model. Trips in a traffic assignment model always travel between zones, they depart from an origin zone and arrive at a destination zone. In general, adopting a more detailed zoning system has the potential of less information loss (compared to reality), albeit at the cost of a higher computational burden.



Travel times, congestion levels, toll revenue, emissions
 Figure 1.3: Schematic interaction of traffic assignment components.

On the supply side, the *physical road network* is considered to be a fixed input. A typical network consists of *nodes* and *links*. Links represent a stretch of road where homogeneous characteristics are assumed. These characteristics can include, but are not limited to, the maximum speed, number of lanes, length, and maximum throughput, i.e. link capacity. Nodes represent locations where links intersect, such as junctions, roundabouts, or on-ramps. Paths are, in this context, typically defined as a sequence of consecutive links, hence they are considered to be supply side entities. Since paths can - potentially - be constructed beforehand, they can also be considered as supply side inputs.

Centroids and *connectors* are part of the interface that allow the demand and supply to interact. A centroid represents a zone via a single point which can be considered a special type of node. Connectors, also known as connector links, are virtual, non-physical, links between a centroid and a regular node. Centroid and connectors allow the traffic demand to enter, or leave, the physical road network during the network loading procedure in order to depart from, or arrive at, a zone. Unlike other traffic assignment components there is no general consensus on whether centroids and connectors are demand or supply side entities. They do however play an important, but often underestimated, role in traffic assignment models. We consider them to be neither demand nor supply side and instead classify them separately. This separate traffic assignment model component is referred to as the *demand-supply interface*.

In reality, trips take time to complete and interactions with other vehicles that occur during each trip are both space and time dependent. Consequently, traffic assignment procedures, ideally, are dynamic in nature because dynamic network loading procedures explicitly consider the time-varying aspect of traffic during the network loading process. Static network loading procedures on the other hand do not and instead aim to describe some average steady-state situation. As a result, dynamic network loading models have a higher potential to accurately reflect traffic conditions than their static counterparts. While dynamic network loading

procedures can be more realistic, they are generally much more time consuming to solve and lack the attractive mathematical properties of their static siblings. There also exist network loading procedures that include both static and dynamic characteristics with the objective to reduce computation times (compared to dynamic models), while improving the model accuracy (compared to static models). Examples of such models are quasi-dynamic and semi-dynamic models. Quasi-dynamic models are in fact static models that aim to produce outcomes that are somewhat comparable to dynamic models. Semi-dynamic models do explicitly consider the time dimension, but do this by modelling a limited number of (larger) time periods where each time period is modelled with a separate static network loading procedure. The model then applies some rules to regulate the interaction between periods in order to improve model results. We discuss the characteristics of various existing traffic assignment models in more detail in Chapters 3 and 6.

1.3 Multi-scale model environment

As mentioned in the introduction, it becomes increasingly common to operate multiple traffic assignment models alongside each other, all modelling the same, or parts of, some spatial domain. In such a multi-scale environment, the main differentiator between the deployed traffic assignment models is found in their level of simplification. In practice, this distinction is often tied to the model's planning horizon. Broadly speaking, three categories of models are distinguished, each with different planning horizons. They are known as *strategic*, *tactical* and *operational* planning models. *Strategic planning models* are the coarsest model type and are used for long term planning purposes. These models aim to forecast traffic conditions from roughly two, up to ten, or more, years into the future. *Tactical transport planning models* have a shorter forecasting time span of a few months up to two years. Such models can be used to test the effects of Intelligent Transport System (ITS) measures such as ramp metering effects, event management systems, or the impacts of local land use changes and developments. They are equally applicable for testing the influence of new residential or commercial developments on the local road infrastructure. Finally, there are *operational transport models*, these models support short term decision making. They provide forecasts on the effects of, for example, diverting traffic given some scheduled road works, possible incidents, or traffic signal optimisation. An increasingly popular application of operational models is to adopt them in an online setting where the model predicts the state of the network in the next 10-15 minutes. The latter however, is not considered a planning model and therefore falls outside the scope of this work.

The owners and operators of each of these three model types have traditionally been located in different branches of government. Strategic models were restricted to the planning department domain, tactical models were operated by city councils, or the transport department, and operational models resided in the domain of the transport management centres and road services. As a consequence, there has been little interaction between these models. Over time, each model type evolved in a way that suited their respective application niche.

Multi-scale traffic models are the result of the blurring boundaries that traditionally separated these three model categories (and levels of government). Ideally, a multi-scale approach is able to bring together strategic, tactical and operational models in a way that they operate alongside

each other and draw from the same resources, in the hope to reduce cost, increase efficiency and improve model outputs. This has revealed practical and theoretical complications. Because the original models have been developed separately, their inputs are often inconsistent; they use different zoning systems, networks, and their travel demand is estimated and calibrated separately as well. To complicate matters, the traffic assignment procedures themselves differ as well. Consequently, model results differ significantly even when applied to the exact same spatial domain. The obvious question then rises, what model is more accurate, and why?

To answer this question, first, a more precise way to assess the capabilities of each traffic assignment model is required. Qualifications based on a single model aspect, such as the planning horizon, do not suffice. To be able to objectively assess model capabilities its underlying assumptions should be known and taken into account. Bliemer et al. (2017) propose a traffic assignment classification framework based on three main assumption dimensions that can be adapted to do exactly this. They propose a *spatial*, *temporal*, and *behavioural* dimension with further subcategories to classify existing traffic assignment models in an objective manner.

Depending on these spatial, temporal and behavioural assumptions different model representations arise. For example, long term strategic models typically adopt more aggressive simplifications resulting in a coarser model representation than, let's say, a typical operational model. As shown in Figure 1.4, the amount of information loss suffered (compared to some "perfect" base model) in such long term models is therefore often much larger than in short term models.

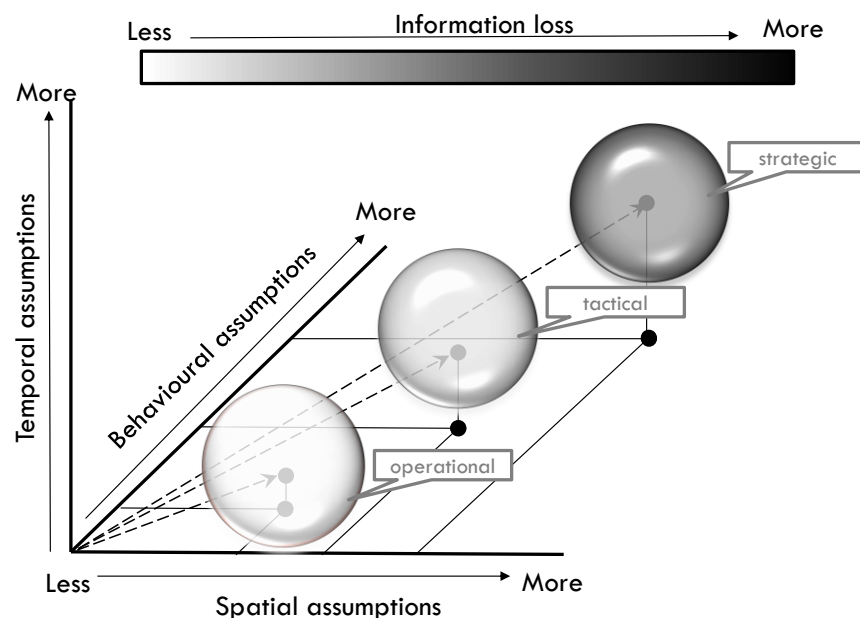


Figure 1.4: Impact of assumptions on information loss by planning horizon paradigm.

It should be noted that the labels less and more in Figure 1.4 do not refer to the number of assumptions that are being made, but rather pertain to the crudeness of the assumption regarding one and the same modelling aspect across the different models. For example, models that simulate each vehicle separately are less crude than models that represent vehicles as an

aggregate stream (in terms of average flow rates per hour), which is considered a more crude assumption.

Although there seems to be a clear separation between the three types of models, this is in fact a too simplified view. While in general it holds that strategic models are coarser than tactical models, there is no rule against creating a long term planning model that has the same, or even a less compromising, set of assumptions than a tactical model. Knowing what the underlying set of assumptions for each model is will result in a better insight in the model's capabilities than relying on ambiguous terminology such as the strategic/tactical/operational paradigm. We use this classification framework as a starting point for discussing existing traffic assignment models and their capabilities (Chapter 3) and show that it can also be used to verify consistency between different traffic assignment procedures in a multi-scale environment (Chapters 7 and 8).

1.4 Representation altering methods

Changing the representation of an original model and construct an alternative representation requires methodology. *Aggregation* and *disaggregation* methods are the two most common types of representation altering approaches. Aggregation methods simplify the original representation while disaggregation methods enhance the level of detail in the model. Both methods can be applied to the traffic assignment model inputs, the traffic assignment procedure, or both. Rogers et al. (1991) discuss a general framework for disaggregation-aggregation analysis in the context of optimisation problems, of which an adapted version is shown in Figure 1.5. While most contemporary traffic assignment models are no longer formulated as optimisation problems, the components in this framework remain relevant. The error analysis component for example, relates to the amount of information loss suffered, while the scaling magnitude can be regarded as the result of adopting a particular (dis)aggregation procedure.

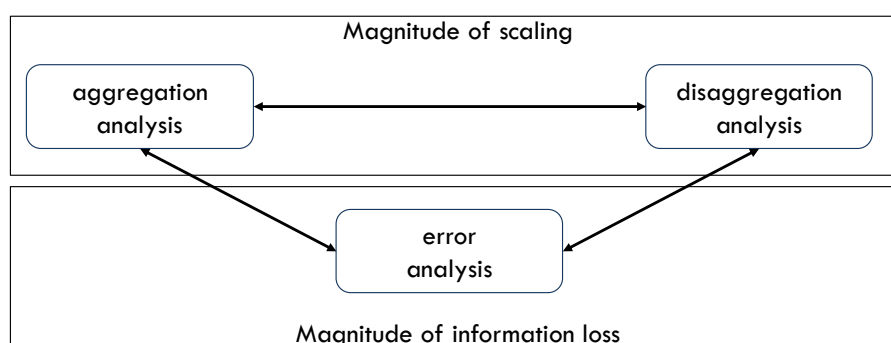


Figure 1.5: Traditional (dis)aggregation approach in optimisation (adapted from Rogers et al. (1991)).

In aggregation, individual data points are grouped together and the aggregate result, i.e. group, is then considered to be a single data point. The procedure then uses the aggregate data points to solve the problem. The reason to aggregate is to either reduce the computational complexity and therefore the computational burden, or to improve the reliability of the results when data is scarce. The latter however, is not very common in the context of traffic assignment procedures. Conversely, disaggregation methods disentangle data points, in the hope to reduce

the error. Disaggregation comes at the cost of a higher model complexity and therefore increased simulation times. Aggregation and disaggregation are not limited to data, but can also occur on a procedural level. Again, instead of modelling individual cars in a microscopic model, one can take on a more aggregate view in a macroscopic model, where traffic is modelled on the basis of average flow rates. This means that at the procedural level, a simplifying assumption is applied, resulting in individual data points being aggregated, inevitably causing some information loss.

Besides aggregation and disaggregation there also exist *decomposition methods*. In decomposition, an original model procedure is broken up in smaller parts. The idea is that a successful decomposition method allows one to solve each part separately where the combined cost of the individual parts is less than solving the original problem as a whole, i.e. the dimensionality of the problem is reduced. This in contrast to aggregation where the problem is still considered as a whole, albeit in aggregate form. Another difference with (dis)aggregation methods is that in decomposition, the original input is preserved and not replaced. Finally, in decomposition methods, the identified sub-components may partially overlap to exchange information, see for example Flötteröd and Osorio (2017), something uncommon in (dis)aggregation. Figure 1.6 provides another way of looking at the differences between decomposition and (dis)aggregation methods¹: decomposition methods do not change the underlying assumptions that are made, but merely focus on changing the procedure to find the same, or similar solutions, at a lesser cost. Aggregation methods on the other hand do change the underlying assumptions to achieve their objectives and typically suffer from information loss as a result. When the decomposition method is perfect there occurs no information loss, but this is might not always be possible, hence the additional dashed line in Figure 1.6(b).

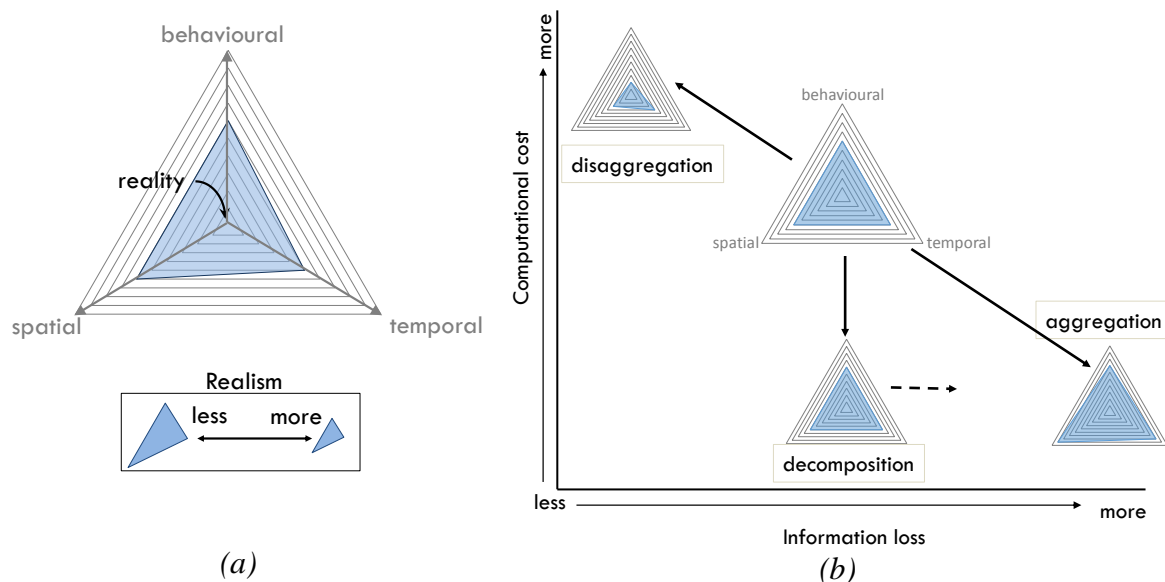


Figure 1.6: (a) Schematic spider chart of assumption impact on realism, and (b) relation between information loss and computational cost when adopting different methodologies.

¹ Figure 1.6 (a) adopts a spider chart to depict the assumptions in each of the three assumption categories (the centre represents reality). Figure 1.6(b) depicts the expected effect of decomposition versus (dis)aggregation in terms of computational cost, information loss, and changes in underlying assumptions.

1.5 Part I: Altering traffic assignment model representations

Part I of this thesis expands on the concepts discussed in the previous sections and utilises them to do two things. First the components concerning traffic assignment are formalised. We also introduce a high level framework that allows us to formulate any traffic assignment model representation altering method on a conceptual level. This framework comprises all elements that are (potentially) affected when one alters a traffic assignment model's original representation. Both in Part II and Part III we adopt this framework to introduce the reader to the main concepts of the proposed methodologies. Further, the existing traffic assignment literature is discussed in order to provide a broad overview of existing methods, how they relate, and what potential they have for applying either aggregation or decomposition oriented techniques to yield alternative representations that are both capable and minimal in their respective application context.

1.6 Part II: Path induced traffic assignment model decomposition

In Part II of this thesis we are not yet concerned with a multi-scale environment. Instead, we propose methodology to minimise the computational cost for a particular group of applications. We assume the supply, i.e. road network, and zoning system to be given and fixed, while the traffic demand may vary across scenarios. By explicitly targeting this application area we are able to construct a minimal, yet equally capable model compared to its original representation.

To achieve this, we propose a novel decomposition method. The original network is decomposed in a free flow and a delay subnetwork where the latter only includes the topology of identified potential bottlenecks. In the free flow network, path travel times are assumed to be flow invariant and are based on free flow conditions. This allows us to reuse them across demand scenarios. Path travel times in the delay subnetwork do vary with flow and require equilibration for each considered demand scenario. However, we only construct a single delay subnetwork across all demand scenarios. The constructed delay subnetwork is likely much smaller than the original network and therefore significantly easier to solve. On top of the delay subnetwork methodology, we also propose a path consolidation method. This method constructs the minimal path set while still being able to reconstruct the travel times found in the original model representation. This second stage of the method does not yield any information loss. The two approaches can be combined to maximise the computational gains. However, it cannot be guaranteed that the overall procedures is lossless. That said, any information loss that does occur can be detected a-posteriori and compensated for. Finally, we also introduce an adaptation of the original approach allowing the user to increase the likelihood of obtaining a lossless result directly.

The more demand scenarios are considered in the application, the greater the computational gain will be. This is due to the fact that the relative overhead involved in constructing the delay subnetwork and consolidated path set reduces which each additional demand scenario. This implies that applications that can benefit most from this approach are applications that evaluate many different demand scenarios.

1.6.1 Applications of path induced traffic assignment model simplifications

As long as an application has a fixed supply, but considers multiple demand scenarios, it is compatible with the proposed methods. However, we wish to highlight three especially suitable applications, namely:

- Quick-scan tools,
- Travel demand matrix calibration procedures, and
- Travel time unreliability studies

Quick-scan tools are applications that serve as a first step in identifying viable scenarios worthy of further, more detailed, investigation. They are often employed to aid for example mobility management policies. They typically investigate many demand scenarios based on different underlying assumptions about, for example, future demand trends. By reducing the computational cost per explored demand scenario, policy makers will be able to either investigate more scenarios, or adopt more sophisticated methods, in order to improve the quality of their recommendations.

Travel demand matrix calibration procedures are known for their computationally demanding nature, they explore hundreds, if not thousands, slightly different demand scenarios in order to calibrate a prior demand matrix. By adopting our decomposition method each individual demand scenario can be solved quicker, which either improves the final result, or at the very least reduces run times.

Travel time unreliability is increasingly considered as one of the main components of the generalist cost function. Travel time unreliability can be the result of both demand and supply factors, but it is clear that daily varying travel demand is an important factor in travel times being unreliable. Probability distributions of travel times can be obtained through Monte Carlo simulation in which travel demand matrices are repeatedly varied and assigned to the model while its infrastructure remains fixed.

These three types of applications most commonly apply static assignment procedures due to their computationally costly nature. We therefore optimised our method to be compatible with a class of static traffic assignment procedures.

1.7 Part III: Multi-scale traffic assignment model representation

Part III of this thesis is dedicated to addressing consistency issues between various levels of detail in transport models. This makes it especially useful in the context of a multi-scale environment. The majority of Part III is dedicated to a novel disaggregation-aggregation based framework to construct consistent traffic assignment model inputs at various levels of detail. The zoning system, traffic demand, network, as well as the demand-supply interface (centroids, connectors) are all constructed automatically, consistently, and in an integrated fashion. To make the method generally applicable, no further assumptions are made on the adopted traffic assignment procedure itself. Instead, we qualitatively discuss the requirements for traffic assignment procedures to qualify for utilising the constructed inputs. This qualitative

assessment is based on the assumption driven classification framework of Bliemer et al. (2017) which we adopt and extend.

To ensure consistency across demand and supply side inputs we consider them jointly. To avoid a complete overhaul of current practice, we propose to apply our method as an intermediate step; we take the original demand and supply side inputs, apply our method, and then pass them on to the traffic assignment procedure. The objective of the procedure is to ensure consistency between demand and supply assumptions, while at the same time construct a representation with the desired level of detail. One of the main weaknesses, in our opinion, of current zoning system design is their extensive use of, relatively crude, rule based proxies to mimic supply side information. For example, to ensure homogeneity across zones, zones are sometimes delineated by certain road types, either for historical reasons or based on the idea that roads such as freeways yield different travel times than local uncongested roads. In reality though, some smaller roads can have serious congestion issues while some larger roads experience little congestion, even during peak hours. Instead of using this simplified proxy, an estimate of the actual road usage is used. This is especially relevant in traffic assignment models where travel times are one of the most important model outputs.

Based on the expected road usage, the representation of the zoning system and travel demand distribution is refined. Interestingly, we found that the same metric can be used to identify the appropriate level of detail in the network representation, as well as to estimate connector costs. Based on our findings, we developed a framework around this concept and provide a reference implementation and methods for each of the steps involved. By situating this procedure between the original construction of demand and the actual traffic assignment procedure itself, it can be readily adopted in any existing traffic assignment model with relatively little effort. We would like to stress that even though the proposed method is developed to be used in a multi-scale environment, it can also be used to simply create a more justifiable demand-supply interface representation. Given that connector costs, in practice, still mostly rely on ad-hoc methods or subjective expert opinions, we feel our method, in this respect, can be of direct value to standalone traffic assignment models as well.










1.8 Case studies

To demonstrate the results of our findings and justify choices that are made regarding parameter calibration, we include a number of case studies. In addition, small hypothetical example networks are used throughout this thesis to illustrate basic concepts. The two real-world networks we use throughout this work are the Amsterdam (the Netherlands) network and the Gold Coast (Australia) network.

An impression of the networks is provided in Table 1.1. The Amsterdam network takes on of two possible forms, its original strategic transport model incarnation, kindly provided by Amsterdam city council, and referred to as Amsterdam I. We also use a variant containing the complete network, which not only contains the main roads, but all roads accessible to private vehicles. This network is referred to as Amsterdam II. The Gold Coast case study is kindly provided by Veitch-Lister consultancy, it

covers a larger area than the Amsterdam networks and serves as a demonstration of the possible computational gains when adopting the methodology proposed in Part II.

Table 1.1: Networks for real world case studies.

| Network | Characteristics | Granularity impression | |
|---|--|--|---|
|  | <p>Amsterdam I (Netherlands)</p> <ul style="list-style-type: none"> - 279 centroids - 3,039 nodes - 7,736 links - 174,297 routes |  |  |
|  | <p>Amsterdam II (Netherlands)</p> <ul style="list-style-type: none"> - Centroids not considered - 8,098 nodes - 22,738 links - No routes |  |  |
|  | <p>Gold Coast (Australia)</p> <ul style="list-style-type: none"> - 1,067 zones - 2,987 nodes - 10,016 links - 1,221,524 routes |  |  |

1.9 Traffic assignment scope

In this work we are predominantly interested in developing novel methodology to improve the current practice of (long term) planning model applications, both in terms of computation times (Part II) and consistency between planning models in a multi-scale setting (Part III). Because of our focus on strategic and possibly tactical planning models, a number of scope reducing choices are made with respect to the considered traffic assignment models. These choices are in line with how these models are used in practice and the applications that we target (see Section 1.6.1 and 1.7).

1.9.1 Equilibrium

First, we subscribe to the equilibrium paradigm. The models aiming to find equilibrium are sometimes alternatively referred to as *within day models* because their solution reflects a result for, at most, a single day. In planning studies this is still the dominant way of modelling, because it allows us to compare different future scenarios in an objective fashion. Also, planning models often still adhere to static traffic assignment where the equilibrium solution is comparatively easy to find. For short term models on the other hand, one is less interested in the rather artificial construct of equilibrium because there is less need for scenario comparisons nor describing an equilibrium state. Also, short term models often adopt dynamic traffic assignment procedures for which it is notoriously difficult to find equilibrium based solutions in the first place. When these short term models describe a trend of changing travel behaviour over a period of days they are referred to as *day-to-day models*. Finally, all non-equilibrium models that just performs a single simulation, for let's say a single morning peak, are termed *one-shot models*.

Second, there exist different types of equilibrium approaches to choose from when adopting the within day modelling approach. Each equilibrium type results in a different result depending on the definition of what finding the equilibrium solution actually means. In planning models, the *User Equilibrium* (UE) approach is still the most commonly adopted approach. In UE, drivers are assumed to behave selfishly by switching paths as long as they can improve their absolute, or perceived, generalised costs. This approach, where individuals independently try to minimise their costs, under complete or incomplete information, is commonly used when trying to model (human) behaviour. Conversely, when drivers behave cooperatively or altruistically we refer to it as a *System Optimum* (SO) approach. In an SO approach travellers work together to achieve some network wide objective such as minimising the total travel time of the system, minimising emissions, or minimising noise. Such SO approaches are sometimes used to obtain a benchmark solution, but are generally not considered to describe actual behaviour. In this thesis we only consider UE assignment methods as the dominant approach for describing travel behaviour.

1.9.2 Paths and path choice

Path choices can be modelled on a *pre-trip* bases and/or when a trip has already commenced. The latter is known as *en-route* path choice. In real life, drivers likely have an initial route in mind (pre-trip) but may deviate when special traffic conditions are encountered (en-route). Pre-trip path choice is consistent with UE in the sense that each individual is assumed to follow their own best route. The potential deviations from this route are generally not considered in

transport planning studies because strategic applications generally only aim to model habitual (commuter) behaviour whereas short term studies more often focus on non-recurrent situations that require en-route path choice, e.g. accidents, road closures etc.. Hence, we do not consider en-route path choices in this work.

As mentioned, the generalised path cost that drives the path choice can consist of many different cost components. Due to our focus on aggregation and decomposition methods for traffic assignment rather than path choice itself, we therefore, for simplicity, consider generalised cost to consist solely out of travel time. We do emphasize however that our proposed methodology can easily be extended to support any other, more sophisticated, generalised cost function as well.

Knowing the path choices based on the generalised cost, there exist different ways to track the travel demand being loaded onto the network. A simple approach, often adopted in real-time models, entails constructing a (fixed) turn fraction for each turn at a node. This fraction indicates the percentage of traffic making this specific turn at the intersection. These fractions can for example be based on empirical data. While computationally convenient, this approach can lead to significant inaccuracies because there is no direct relation between the origins of traffic and where it is directed to. An alternative to this approach is to track paths explicitly such that the full information of the trip is maintained. This is a costlier approach in terms of computation time and memory use. The paths themselves can either be constructed beforehand, generated while the model is being solved, or a combination of the two. We choose to adopt a path based approach, because it is both a more flexible and more realistic approach compared to the fixed turn fraction method.

1.9.3 Demand inelasticity and single user class

Traffic assignment models that incorporate *departure time choice* allow travellers to shift their departure time to another (modelled) time period. This is also referred to as demand elasticity. *Mode choice* is another possible extension sometimes considered within the traffic assignment paradigm. When including mode choice, trips can choose between modes substituting a car trip with a multi-modal public transport trip for example. Again, for simplicity, we do not consider departure time choice nor mode choice in our assignment procedures. Traffic demand may still be provided across multiple departure times (and/or modes), but it is assumed its proportions are fixed, i.e. it is demand and mode inelastic. Also, we only assign trips that are made by private car, and we only consider homogeneous driver behaviour. This is known as a single user class approach. It allows us to discuss the relevant traffic assignment concepts while remaining focussed on our objective of constructing appropriate representations. Again, the proposed methods in this thesis are easily extendable to a multi-modal and/or multi-class context and where relevant this is pointed out to the reader.

1.9.4 Scope Overview

Based on the scope decisions discussed in the previous sections, Table 1.2 reiterates the choices made and outlines them specifically for each Part of the thesis where relevant.

Table 1.2: High level scope.

| Feature | General | |
|-------------------------|--------------------------------|----------------------------|
| Transport model type | Path based traffic assignment | |
| Traffic assignment type | Within day, user equilibrium | |
| Driver behaviour | Single user-class | |
| Travel choices | Pre-trip path choice | |
| Generalised cost | Travel time only | |
| Modes | Private car | |
| | Part II specific | Part III specific |
| Applications | Quick-scan, matrix calibration | Traffic assignment models |
| Planning horizon | Strategic, tactical | No restriction |
| Applied methodology | Decomposition | Disaggregation-aggregation |
| Network loading | static assignment | No restriction |

1.10 Thesis contributions

Here, we summarise the contributions of this thesis. Note that some of the contributions are formulated using concepts that have not yet been discussed and as such this listing is merely to be considered as a reference at this stage.

In Part II:

- 1) Methodology to extract a path induced delay subnetwork;
- 2) Methodology to identify a minimal consolidated path set for delay subnetworks
- 3) Super-scenario construction and analysis to identify bottleneck infrastructure across demand scenarios.
- 4) Extensive analysis of the proposed two methods on two real-world case studies, in the context of static capacity constrained assignment with residual point queues.

In Part III:

- 5) A qualitative assessment of traffic assignment model requirements in a multi-scale environment.
- 6) A general disaggregation-aggregation framework to construct consistent traffic assignment model inputs in an integrated fashion, suitable to be used in a multi-scale environment. Within aforementioned framework, we propose;
- 7) Methodology to construct the supply representation based on expected road usage.

- 8) Methodology to construct the demand-supply interface based on the concept of internal travel time stability, following from the found supply representation.
- 9) Methodology to refine the demand representation by formulating a constrained optimisation problem for the construction of the zoning system, following from the found demand-supply interface representation.
- 10) A custom optimal branch-and-bound solution scheme to solve the constrained optimisation problem mentioned in 9).
- 11) Extensive analysis of results as well as parameter calibration based on a real-world case study. We compare results to a (subjective) ad-hoc approach as well as a fully disaggregate representation.

1.11 Thesis outline

This thesis is organised as follows: It contains four parts, Part I-IV, where Part I and Part IV serve as introduction and conclusion, respectively. Parts II and III comprise the core of the thesis.

Part I, apart from this Chapter 1 containing the introduction, consists of Chapters 2 and 3. Chapter 2 introduces the reader to a high level framework for altering traffic assignment model representations. We utilise this framework both in Part II and Part III to discuss the affected model components and the general principles underpinning both approaches. In Chapter 3 we discuss the current state of traffic assignment models in the literature, we do so based on the modified and extended classification framework originally proposed in Bliemer et al. (2017).

Part II consists of three chapters, i.e. Chapters 4 to 6. In Chapter 4 we discuss the various ways that one can capture travel time delay and how this leads to different implications with respect to decomposing link level travel times. Chapter 5 adopts a particular travel time decomposition approach termed functional travel time decomposition, where we demonstrate we can decompose the transport network utilising a path centric travel time decomposition method that is compatible with a class of static traffic assignment procedures. Chapter 6 demonstrates the effectiveness of this method and calibrates the parameters involved through several case studies.

Part III consists of five chapters, i.e. Chapters 7 to 11. In Chapter 7 the current state of aggregation methods in traffic assignment is discussed. In Chapter 8 an assessment on the requirements for traffic assignment procedures in a multi-scale environment is given, resulting in three consistency criteria. Chapter 9 introduces the disaggregation-aggregation framework for the construction of consistent traffic assignment model inputs in a multi-scale environment. It also formulates methods for each of the steps involved, except for the construction of the zoning system. Instead, the construction of the zoning system is captured by formulating a constrained optimisation problem in Chapter 10. In addition, Chapter 10 also proposes a cluster oriented branch-and-bound solution scheme to solve aforementioned constrained optimisation problem. Chapter 11 discusses a number of case studies that are conducted to demonstrate the

suitability of the framework as well as the proposed methods within the framework, including any parameter calibration where needed.

Conclusions and possible extensions are found in Chapter 12 which constitutes Part IV of this work. An alternative outline is shown in Table 1.3, where we outline chapters by content type for the reader's convenience.

Table 1.3: Chapters by content type.

| Type of content | Part I/IV | Part II | Part II |
|--|------------|------------|-----------------------|
| Background | Chapter 1 | | |
| Literature review and qualitative assessment | Chapter 3 | Chapter 4, | Chapter 7, Chapter 8 |
| Methodology | Chapter 2 | Chapter 5 | Chapter 9, Chapter 10 |
| Case studies | | Chapter 6 | Chapter 11 |
| Solution schemes | | Chapter 5 | Chapter 10 |
| Conclusions | Chapter 12 | | |

1.12 Remarks on notation

Throughout this thesis matrix notation is adopted whenever possible. Especially the optimisation problem formulations, their constraints, and clustering procedures benefit from this choice, mainly because they can be concisely formulated. Also, the network infrastructure as well as travel demand can be formulated elegantly via this approach. To be consistent and avoid mixing different types of notation, we choose to adopt this notation throughout this work. The trade-off here is that some formulations, for example regarding traffic assignment, might seem more complex than usual compared to non-matrix based approaches. This is a side effect of preferring consistency over familiarity, something for which we apologise in advance. Whenever, we feel this situation occurs, an effort has been made to add examples in text and figures, to illustrate the effects of the formulation at hand.

2 Traffic assignment representation framework

Chapter 1 provided an introduction to traffic assignment models and the trade-off between information loss and computational cost that occurs when deciding on appropriate model representation. Given the many components that are involved in constructing a traffic assignment model, e.g. the physical road network, centroids, connectors, travel demand, network loading, path choice, as well as the fact that there exist different methods to alter the representation of each of these components, we propose to capture these aspects in a general framework. We then utilise this framework to introduce the reader to the proposed methods and their impacts.

The interaction between supply and demand is at the core of any traffic assignment model regardless of its representation. We therefore formalise the components involved in this interaction first in Section 2.1. Then, in Section 2.2, the traffic assignment representation framework is presented. Within this framework the objective of constructing both capable and minimal model representations are formulated. To quantify results, the framework caters for comparisons between the newly created model representation and the original model. Both Part II and Part III utilise this framework to illustrate the impact of the proposed methodology on the relevant traffic assignment components.

2.1 Demand-supply interaction

Traffic assignment consists of a demand model and a supply model (Section 1.2), where we assume that the number of trips between each origin and destination zone is known and given. The responsibility of the model is to distribute this known demand over the available supply, i.e. the physical road network, using eligible paths between origins and destinations.

The demand model is defined as function $\Psi(\cdot)$ that maps path costs to path flows. Path costs are denoted by generalised path cost vector $\mathbf{h} \in \mathbb{R}_+^{P \times 1}$, where h_p denotes the cost of path p , with $p \in \{1, \dots, P\}$, where the total number of paths is given by P . Generalised path costs are assumed to only consist of travel time (h), including delay. The output of the demand model is given by path flow vector $\mathbf{f} \in \mathbb{R}_+^{P \times 1}$, where f_p denotes the desired flow rate (veh/h) on path p . The mapping from cost to flow in the demand model is conditional on travel demand matrix $\mathbf{D} \in \mathbb{R}_+^{Z \times Z}$. Travel demand is defined between zones, where each zone is uniquely identified through $z \in \{1, \dots, Z\}$. Therefore, the demand model captures the path choice behaviour of travellers making trips based on the given demand via:

$$\mathbf{f} = \Psi(\mathbf{h} | \mathbf{D}). \quad (2.1)$$

Equation (2.1) can easily be extended to support multiples modes and/or time periods by making the demand, costs, and flows, mode and/or time (period) dependent. However, as discussed in Chapter 1, we refrain from doing so to prevent unnecessary notational complexity given our focus on representation altering methodology and not traffic assignment as such.

The supply model provides the level of service on the transport network by loading path flows onto the infrastructure. This network loading procedure is formalised via supply model function $\Phi(\cdot)$, mapping path flows \mathbf{f} to path costs \mathbf{h} , such that:

$$\mathbf{h} = \Phi(\mathbf{f} | \mathbf{A}), \quad (2.2)$$

where $\mathbf{A} \in \mathbb{F}_2^{N \times N}$ represents the transport network through nodes and links. This network is modelled via an indicator matrix, also known as incidence matrix, binary matrix, or Boolean matrix. An indicator matrix only allows two values, zeros and ones. We adopt the Galois field modulo two (\mathbb{F}_2) notation to model this binary space. Each non-zero entry in \mathbf{A} indicates that a directional link exists from one node, i.e. row, to another node, i.e. column, with each node $n \in \{1, \dots, N\}$, where the total number of nodes in the network is denoted by N . Directional links $a \in \{1, \dots, A\}$ are constructed implicitly by incrementally numbering ordered node pairs, as illustrated in Figure 2.1.

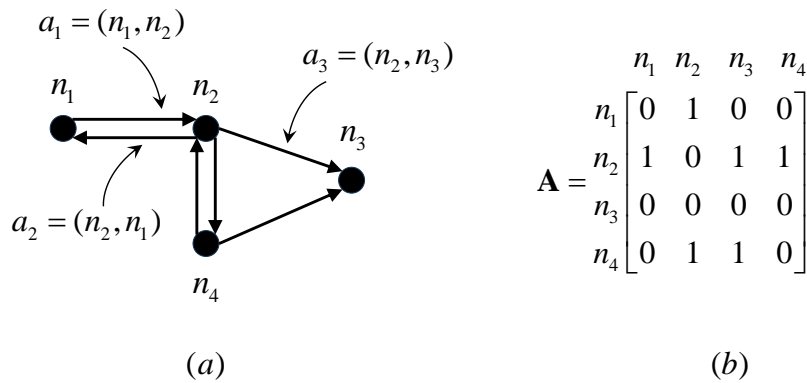


Figure 2.1: (a) Road network of nodes and links as a directional graph, (b) the same network in matrix notation².

The interaction between the supply and demand model, conditional on the demand and transport network, is depicted in Figure 2.2. The purpose of this interaction is to find an equilibrium solution. When this solution is found, the model yields equilibrium path flows and path costs that can be considered the final result. We denote these resulting vectors via $\mathbf{f}^*, \mathbf{h}^*$, respectively.

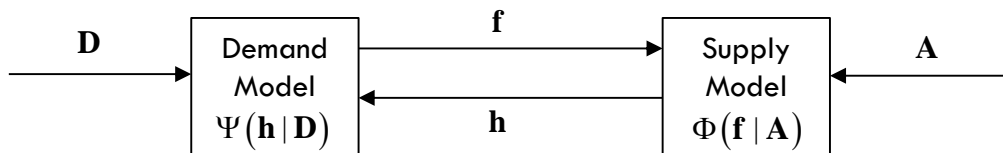


Figure 2.2: Schematic representation of traditional demand-supply interaction model.

Different solution approaches exist to solve this problem. For example, on the demand side path choice can be implemented by adopting a logit model or some other method of distributing

² Throughout this thesis, examples in text and figures sometimes adopt a subscript notation on variables that technically have no subscript and are a value in a range, i.e. a_1 means simply a link a where $a = 1$. This, we feel, is an intuitive way to refer to specific variable instances in the context of these illustrative examples.

flow across the available paths. On the supply side, the network loading (for the utilised paths) can be formulated as a single equation in case of a traditional static assignment procedure (Beckmann et al., 1956), a fixed point problem formulation in case of a quasi-dynamic procedure (Bliemer et al., 2014; Bifulco and Crisalli, 1998), or a more complex dynamic system of equations if one chooses to adopt a dynamic network loading procedure. Then, to model the interaction between the demand and supply, we can also employ different model formulations. These include optimisation problem formulations, fixed point problem formulations, or the popular Variational Inequality (VI) approach (Chen, 1999; Friesz et al., 1993; Dafermos, 1980; Smith, 1979). In Part II we subscribe to a logit based path choice model, a fixed point formulation for the network loading, and a VI formulation to solve the demand-supply interaction. In Part III we prefer to accommodate any such formulation because the actual problem formulation has little to do with the underlying assumptions that lead to specific model representations. Therefore, for the sake of general applicability, we do not force the adoption of a particular model formulation in the proposed framework, but instead focus on the formulation of the individual traffic assignment building blocks which together determine the traffic assignment representation.

2.1.1 Explicit demand-supply interface and path set considerations

The traffic assignment model depicted in Figure 2.2 is not considered complete because it does not explicitly account for paths. Yet, it implicitly assumes paths are constructed on-the-fly, as part of the demand-supply interaction procedure itself. While this is common in a deterministic setting, it is less common in models where uncertainty or imperfect information play a role (see also Chapter 3). It is therefore very well possible paths are provided a-priori, possibly as a fixed input, to the supply model. Alternatively, this pre-constructed path set might be supplemented with additional paths during the assignment procedure. To accommodate these alternative approaches, the more general procedure depicted in Figure 2.3 is adopted, where the dashed line indicates an optional (predefined) path matrix $\mathbf{P} \in \mathbb{F}_2^{P \times A}$. Like transport network \mathbf{A} , paths are denoted via an indicator matrix, where each row represents a path $p \in \{1, \dots, P\}$, while each non-zero column entry denotes that link a is present on the path. In the absence of cycles, the path link order is implicitly determined as are the links.

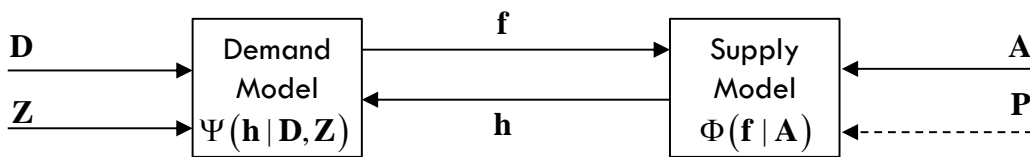


Figure 2.3: Representation of proposed demand-supply interaction model.

Another aspect that needs addressing is the demand-supply interface. Traditionally, in traffic assignment, trips access the transport network via centroids, yet demand \mathbf{D} is provided on a zone-to-zone basis. One could replace the zone-to-zone demand with a centroid-to-centroid matrix. This however, would unnecessarily restrict the flexibility of our formulation, because any relation between the geographical area and the centroid is lost. We therefore choose to do the following instead: (i) we retain our zone based formulation and add a dedicated mapping from zones, denoted $z \in \{1, \dots, Z\}$, to centroids via indicator matrix $\mathbf{Z} \in \mathbb{F}_2^{Z \times N}$, (ii) our transport network \mathbf{A} explicitly incorporates these centroids as a special type of node, (iii) our transport

network explicitly incorporates connectors as a special type of link. Hence, the demand model function is then altered accordingly:

$$\mathbf{f} = \Psi(\mathbf{h} | \mathbf{D}, \mathbf{Z}). \quad (2.3)$$

Figure 2.4 shows an example of the how the discussed traffic assignment model inputs relate to their matrix based formulation in terms of the zone-to-centroid mapping and path-to-link mappings.

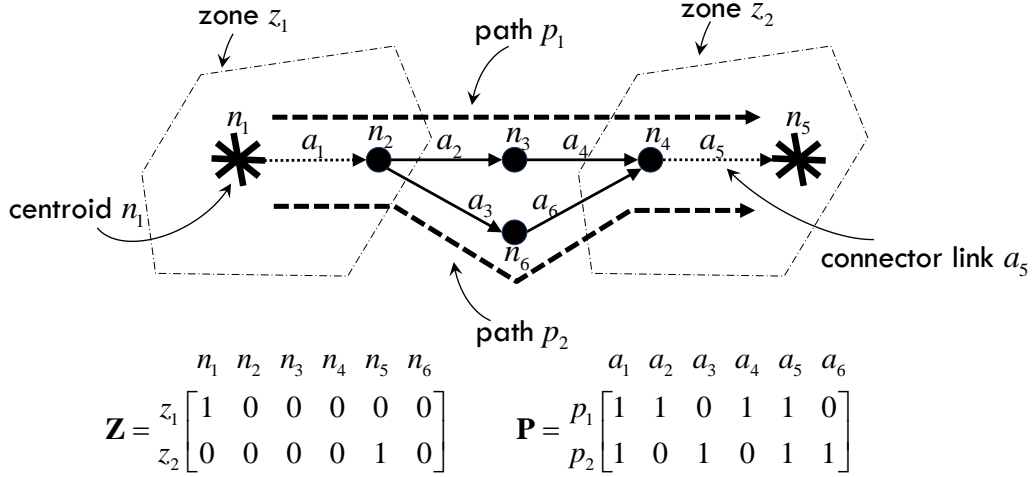


Figure 2.4: Example network with geographical zones, centroids, connector links and paths.

2.2 General framework

The model formulation and its inputs discussed in the previous section are, from here on forward, considered as the original, or reference, representation of the traffic assignment model. Clearly, when one or more of these original components are altered, an alternative representation emerges. Let us first introduce a shorthand notation \mathcal{M} representing all relevant model components via:

$$\mathcal{M} = (\mathbf{A}, \mathbf{D}, \mathbf{Z}, \mathbf{P}, \Psi(\cdot), \Phi(\cdot)). \quad (2.4)$$

An alternative representation is then denoted $\mathcal{M}^* = (\mathbf{A}^*, \mathbf{D}^*, \mathbf{Z}^*, \mathbf{P}^*, \Psi(\cdot)^*, \Phi(\cdot)^*)$. Transforming \mathcal{M} to \mathcal{M}^* is the result of applying implicit representation function $\Xi_\gamma(\mathcal{M})$, where the chosen “rules” governing this transformation are denoted by γ such that each choice of γ yields a particular alternative representation. Hence, we find that:

$$\mathcal{M}^* = \Xi_\gamma(\mathcal{M}). \quad (2.5)$$

We emphasize that not all components need to be altered necessarily. The formulation merely caters for any number of components being changed. When adopting an alternative representation instead of the original model the traffic assignment results between the two

models will likely differ. These differences can either be regarded as an improvement or not. To assess any such differences objectively and quantitatively, metrics are required. In Chapter 1, two overarching objectives were discussed informally: the first objective concerned constructing capable model representations, while the second objective was found to be constructing minimal model representations. The former is linked to the magnitude of information loss, while the latter relates to the magnitude of scaling. The information loss objective is formalised via implicit function $\varepsilon(\mathcal{M}, \Xi_\gamma(\mathcal{M}))$, which can alternatively be interpreted as the error analysis function. We measure the capability objective via the inverse magnitude of scaling, via implicit function $\zeta(\mathcal{M}, \Xi_\gamma(\mathcal{M}))$. We adopt an inverse formulation to be able to formulate both objectives as part of the same *minimisation* based problem formulation:

$$\min_{\gamma} \begin{pmatrix} \varepsilon(\mathcal{M}, \Xi_\gamma(\mathcal{M})) \\ \zeta(\mathcal{M}, \Xi_\gamma(\mathcal{M})) \end{pmatrix} \quad (2.6)$$

Alternatively, a graphical interpretation of this framework is provided in Figure 2.5. It shows how each of the model components interacts when comparing the quality of an alternative model representation to its original counterpart.

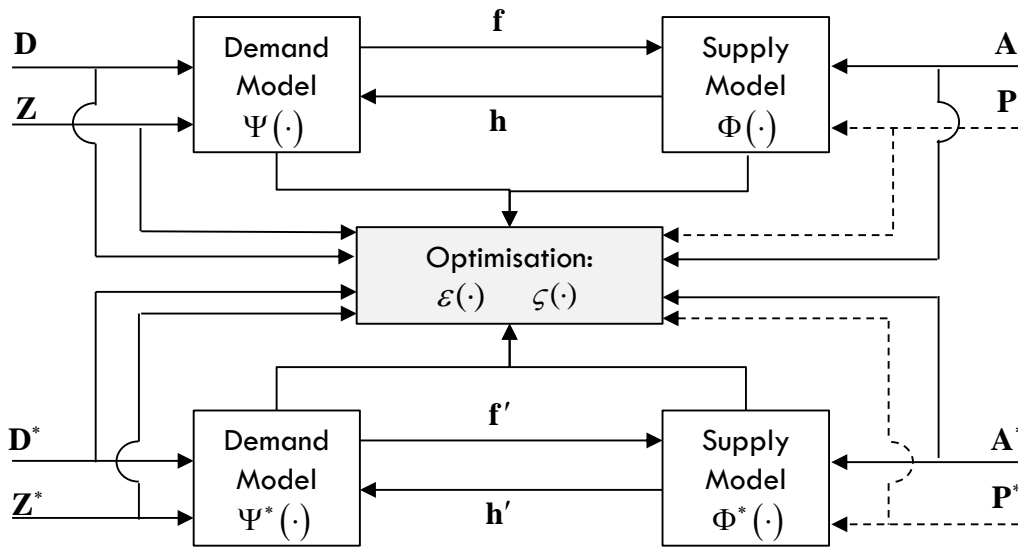


Figure 2.5: Schematic impression of representation optimisation framework.

This framework imposes no restrictions on the adopted formulation, methodology, or underlying model assumptions that lead to a change in model representation. It therefore supports a broad range of methods and serves as the starting point for the methods proposed in both Part II and Part III of this thesis.

3 Traffic assignment

In Chapter 2 we introduced a general framework for altering the representation of traffic assignment models. To present novel methodology based on this framework, first, an understanding of existing traffic assignment procedures is required. In this chapter we provide an overview of the literature on traffic assignment. The concepts and discussed model types are relevant to both Part II and Part III of this thesis.

In Section 3.1, we briefly introduce the reader to the three different representations of traffic flow that lead into the discussion of the microscopic, mesoscopic, and macroscopic traffic assignment paradigms. The fundamental diagram, that describes the relationship between essential traffic flow variables, plays a pivotal role in virtually all traffic assignment models and is discussed in Section 3.2. In Section 3.3, we look in more detail at the classification of traffic assignment models based on spatial, temporal, and behavioural assumptions. We utilise this framework to discuss the existing traffic assignment literature in more detail. Section 3.4 considers the literature from a temporal perspective, while Sections 3.5 and 3.6 discuss the spatial and behavioural perspectives, respectively.

3.1 Representations of traffic flow

The network loading procedure responsible for modelling the traffic flow dynamics is the most visible component of traffic assignment models. This is perhaps why, in practice, traffic assignment models are predominantly categorised by their representation of traffic flow. There exist three main categories which are known as; *microscopic*, *mesoscopic*, and *macroscopic* traffic assignment. Microscopic models represent the most detailed type of models. In this model type, traffic flow is disaggregated, meaning that each individual vehicle is modelled separately. On the other end of the spectrum, macroscopic models reside. In macroscopic models trips are represented in a more aggregate form, namely, via average flow rates instead of individual vehicles. This aggregated approach is less realistic than microscopic approaches, but is generally more attractive computationally. Mesoscopic models reside somewhere in between macroscopic and microscopic models, they are simplified compared to microscopic models, but are more detailed than macroscopic models. Often, but not always, they still simulate vehicles individually, but incorporate aggregate modelling features of macroscopic approaches to reduce their computational complexity.

Laval and Leclercq (2013) formally discuss the three representations of traffic flow for the dynamic context. They demonstrated that each of these representations can be rewritten into the other. The distinction between the three model types can informally be stated like the following:

- Microscopic: track location x of each vehicle at time t ,
- Mesoscopic: track time t each vehicle crosses a location x , and
- Macroscopic: track the cumulative number of vehicles crossing a location x at time t .

The implication of these different representations are further illustrated in Figure 3.1. The microscopic model explicitly tracks each vehicle in time and space, as per Figure 3.1(a). The mesoscopic approach, in Figure 3.1(b), illustrates a simplification compared to the microscopic model, because it allows one to relax the constraint of tracking each vehicle over the entire space. Instead, we can choose a limited number of discrete locations, e.g. x_1, x_2 . In a macroscopic approach only cumulative vehicle numbers are tracked at discrete locations as depicted in Figure 3.1(c).

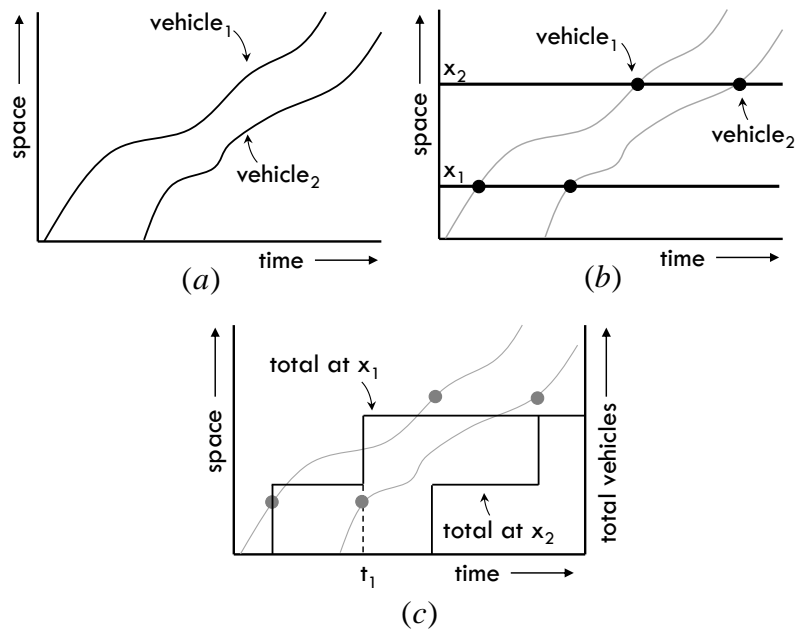


Figure 3.1: Three representations of traffic flow, (a) microscopic, (b) mesoscopic, (c) macroscopic dynamic without averaging.

The macroscopic representation of flow in Figure 3.1(c) allows for further simplification compared to a mesoscopic approach by replacing the individual vehicles altogether with average flow rates (which can be obtained from the difference in total vehicles counted over a period of time). This is exemplified in Figure 3.2(a), where we depict two flow rates, i.e. up to t_1 and after t_1 , both relating to location x_1 in Figure 3.1(c).

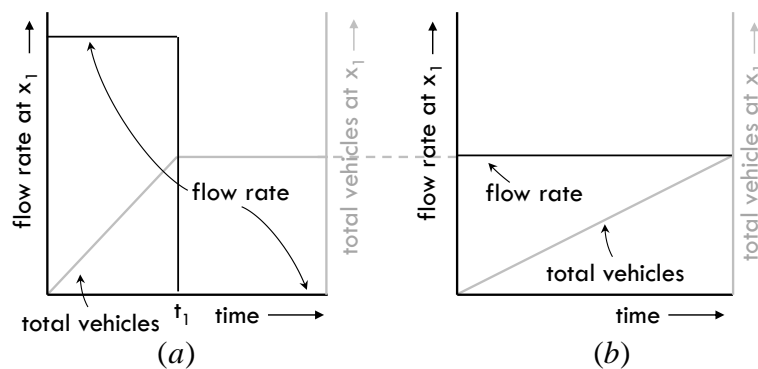


Figure 3.2: (a) Macroscopic dynamic approach with time dependent average flow rates (black) vs total vehicles (grey), (b) macroscopic static with average flow rates vs total vehicles over the entire period.

Moving from a dynamic to a static (macroscopic) context, there is one more additional simplifying assumption imposed. The simplification concerns the aggregation of time. The simulation time period is no longer considered to be continuous, nor is it modelled via multiple (discrete) steps, but is condensed into a single result, i.e. only a single average flow rate over the entire simulation period is constructed. This leads to the situation depicted in Figure 3.2(b), where the piece-wise curve for the flow rate is replaced with a single stationary average flow rate during the entire simulation period.

Because microscopic and mesoscopic models offer a relatively high level of detail, they are most commonly used for (short term) operational model applications in which multiple lanes, multiple vehicle types, random variations, and signalised interactions can be considered with relative ease. Macroscopic approaches on the other hand are more suited to strategic models, because their level of detail is more in line with the uncertainties encountered when performing long term forecasts for planning purposes. Typically, they also exclude any random variations making them especially suitable for scenario comparisons.

Table 3.1 roughly illustrates how, in practice, the planning horizon correlates with the adopted traffic model paradigms. While these best practices are mostly intuitive, we do point out that macroscopic approaches for operational models are somewhat of a mixed bag. On the one hand they are rarely adopted in urban applications that include signals. Conversely, macroscopic operational models pertaining to motorway related applications are common practice indeed, hence our classification of reasonably common for this type of model.

Table 3.1: Best practices in combining planning horizons with traffic flow representation.

| | | Traffic flow representation | | |
|------------------|-------------|-----------------------------|-------------------|-------------------|
| | | Microscopic | Mesoscopic | Macroscopic |
| Planning horizon | Operational | Very common | Reasonably common | Reasonably common |
| | Tactical | Uncommon | Common | Reasonably common |
| | Strategic | Very uncommon | Uncommon | Very common |

3.2 Fundamental diagram

Before discussing microscopic, mesoscopic, and macroscopic models in more detail, we look at the representations of traffic flow and their relation to the *fundamental diagram*. The fundamental diagram describes the relation between density, speed, and flow, for a stationary observer somewhere along the road. Note that the fundamental diagram is sometimes alternatively referred to as the fundamental relation, Hamiltonian, or flux function. The fundamental diagram was first discussed by Greenshields (1935), who claimed a linear relation between speed and density existed, resulting in a quadratic fundamental diagram in the flow-density, and speed-density plane. Figure 3.3 shows a schematic example of the Greenshields fundamental diagram and although this particular shape of the diagram is no longer considered to be very realistic, the importance of the fundamental diagram as a concept cannot be overstated.

The maximum flow that can pass a stationary observer along the road per chosen unit of time is referred to as the *capacity* of that road. The *density* describes the number of vehicles present on a stretch of road (veh/km). The density at capacity is termed *critical density*, while *jam density* occurs when vehicles come to a complete standstill as a result of congestion. Furthermore, as long as the link is below critical density, it is uncongested and considered to be in a *hypocritical* traffic state (Cascetta, 2009). On the other hand, when the link is in a *hypercritical* traffic state, the critical density is exceeded and queues start to form.

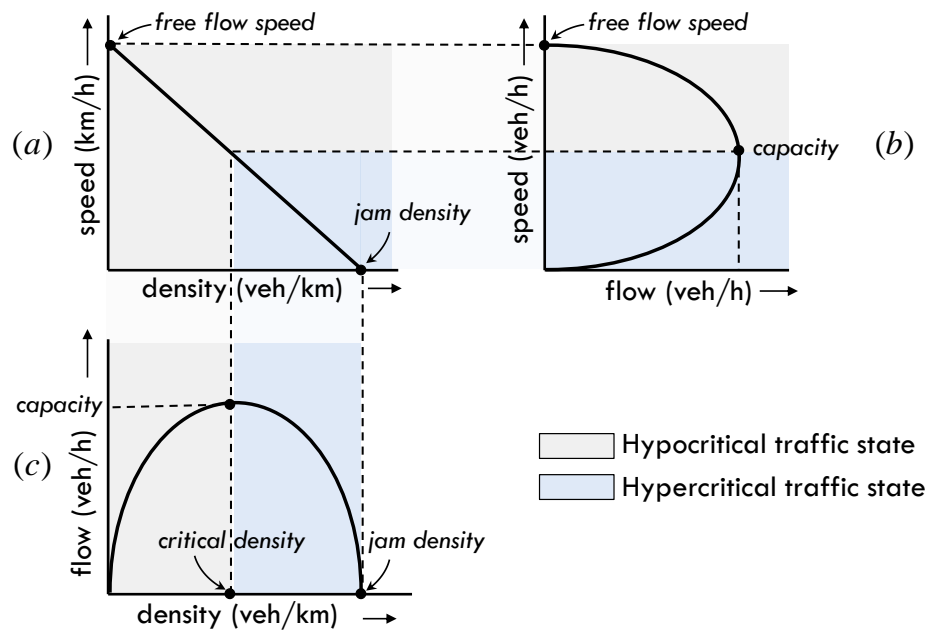


Figure 3.3: Greenshields fundamental diagram, (a) speed-density, (b) speed-flow, and (c) flow-density.

The fundamental diagram is closely related to the representation of traffic flow, and is the foundation upon which each of the three model types is built. This becomes immediately apparent when we look at the units of density (veh/km), flow (veh/h), and speed (km/h), which comprise the same units underpinning the three representations of traffic flow. It can be shown that each macroscopic, microscopic, and mesoscopic model is consistent with a particular fundamental diagram. For example, microscopic car following models often adopt time headway (h/veh), speed, or acceleration/deceleration (derivative of speed) as their main variables. Since time headway is the inverse of flow, and acceleration (where a negative acceleration is a deceleration) represents a change in speed, a corresponding speed-flow relationship as in Figure 3.2(b) can be determined under the assumption of steady-state conditions.

The close relationship between microscopic, mesoscopic and macroscopic models based on the fundamental relation has been known for decades (Pipes, 1967). Microscopic model formulations for example, have been rewritten into macroscopic model formulations (Rakha and Crowther, 2002). Similarly, empirical fundamental diagram data has been used to validate and/or justify microscopic and mesoscopic models (Zheng, 2017; Treiber et al., 2000). These similarities between traffic flow representations are especially relevant in the context of a multi-scale environment that values consistency and is revisited in Part III of this thesis.

3.3 Traffic assignment procedure classification

The microscopic/mesoscopic/macrosopic paradigm distinguishes between models based on the difference in their representation of traffic flow. However, there are many other model features to consider. To discuss the existing literature on traffic assignment procedures in a structured manner we look at the capabilities of these models based on their spatial, temporal, and behavioural capabilities, respectively. We adopt and extend the classification approach introduced in Bliemer et al. (2017) to do so. An important difference compared with the original classification is found in the inclusion of microscopic and mesoscopic traffic flow representations, which are considered as main model types in the spatial dimension, see Figure 3.4.

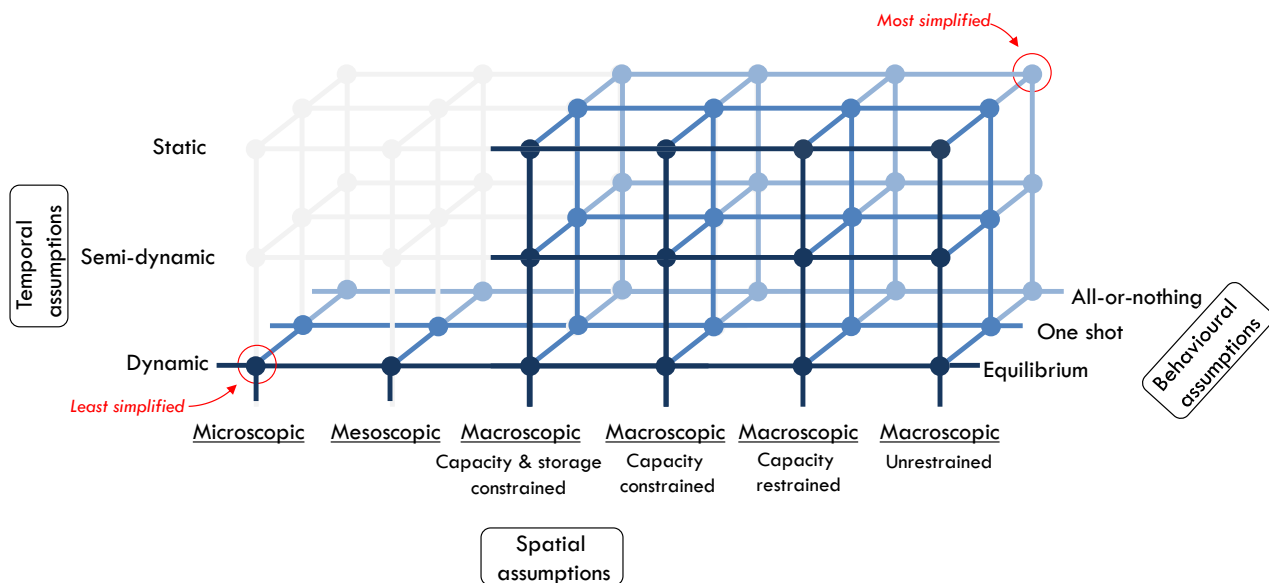


Figure 3.4: Model types by assumption dimension, extended based on Bliemer et al. (2017).

Observe that microscopic and mesoscopic models are assumed to be dynamic in nature, since there is little point in vehicle based propagation in a non-dynamic context, hence the greyed out areas in the framework regarding the temporal dimension.

Macroscopic models are simplified in terms of their spatial interaction compared to microscopic and mesoscopic models. The main distinction being their different representation and propagation of traffic flow. The propagation of individual vehicles in microscopic (and most mesoscopic) models is replaced by propagation methods governed by the fundamental diagram, typically resulting in flow rate based or cumulative vehicle based methods. While microscopic and mesoscopic models are by definition dynamic in nature, macroscopic models comprise a range of models with different temporal assumptions.

3.4 Traffic assignment from a temporal perspective

In the temporal dimensions, three main model types are distinguished: *dynamic models*, *semi-dynamic models*, and *static models*. The most capable models from a temporal perspective are *dynamic models*. Dynamic models can explicitly account for the time dimension in both the

propagation of traffic flow as well in the representation of demand and route choice. In contrast to dynamic models, static models do not consider the time dimension explicitly in the propagation of traffic flow. As a result, only a single route choice period is considered and only a single average flow rate per link results for the period under consideration. One can look at static models in different ways. Traditionally, static models are often thought of as a particular period that experiences steady-state flows, where the construction of this steady-state is considered exogenous to the considered modelling period itself and does not require “worrying about”, it simply exists and remains in place for the period at hand (Payne and Thompson, 1975; Smith, 1987). While this is one way of looking at it, this perspective poses problems when attempting to demonstrate that a static model is consistent with some dynamic counterpart, where flows are a direct result of the explicitly modelled period. Therefore, in recent years, a philosophical shift in perspective arose, where one might alternatively think of static models as a dynamic model where the time period is abstracted out and the path flows are propagated instantaneously. This has the benefit that the static flows are the result of the modelled period, identical to dynamic models, only the time it takes for them to propagate has been abstracted out, which is consistent with the adopted modelling paradigm. From this point of view, it has been shown that existing static model formulations are in fact fully consistent with their dynamic counterparts by directly deriving them, see for example Bliemer and Raadsen (2017).

Early macroscopic dynamic traffic assignment models were mainly extensions to static models. They often utilised a capacity restrained approaches and attempted to find analytical solutions (Janson, 1991; Carey 1987), but due to their high computational costs, and the fact that increased realism often precluded the guarantee of attractive mathematical properties, quickly resulted in a shift towards simulation based approaches. Most contemporary macroscopic approaches that are used in practice are no longer considered to be extensions of static models. Most of these approaches represent different solution schemes (Bliemer and Raadsen, 2018; Raadsen and Bliemer, 2018; van der Gun et al., 2017; Raadsen et al., 2016; Himpe et al., 2016; Gentile, 2010; Yperman, 2007, Daganzo, 1994) to the same underlying model, namely the LWR model (Lighthill and Witham, 1995; Richards, 1956). Similarly, microscopic and mesoscopic models are also dominated by simulation based approaches, see also Section 3.5.1.

A small but growing literature exists on bridging the gap between static and dynamic models. *Semi-dynamic* models are more capable than static models because they break up the time period under consideration into a small number of intervals. Each interval is then modelled statically, but by shortening the period an attempt is made to mitigate the effects of ignoring the time dimension within the propagation procedure. An additional mechanism is then used to model the interaction between the various smaller periods (Nakayama et al., 2012; Akamatsu et al. (in Japanese), 1998), namely by transferring residual traffic on links from one period to the other.

Finally, *quasi-dynamic* models are in fact a type of static model, but one that is considered more capable. The distinction is based on the fact that a quasi-dynamic model is often assumed to be capacity constrained (and possibly also storage constrained by accounting for spillback) and considers queues in some way or form, i.e. some of the instantaneous traffic demand can “get stuck” in the network due to capacity restrictions. These queues can originate either inside

the bottleneck (Bundschuh et al., 2006; Bifulco and Crisalli, 1998) or in front of the bottleneck (Bliemer et al., 2014). In Smith (2013) an explicit location of the queue is lacking because the used toy network is agnostic to the actual location of the queue. Either way, the travel time function needs to be reconsidered to incorporate the effect of these queues. The term quasi-dynamic is sometimes confusing and to avoid such confusion these models are classified as static models with additional capabilities in the spatial dimension.

3.5 Traffic assignment from a spatial interaction perspective

Our focus within the spatial dimension is predominantly on macroscopic approaches. The reason for this is the fact that it is difficult to design effective aggregation methods for microscopic and mesoscopic modelling approaches because the applications they are used for do not allow much compromise in modelling results. Hence, hardly any aggregation methods exist in this context. One could therefore argue that there is little point in including them in this discussion. There is however an important argument to do so anyway, namely, the issues around consistency in a multi-scale setting. In a multi-scale environment microscopic, mesoscopic, and macroscopic models exist alongside each other. To provide insight in the requirements of constructing a successful multi-scale environment, where each of these models should ideally be consistent in relation to one another, a basic understanding of all three modelling paradigms is required. Based on the introductory literature review discussed in this chapter, we discuss such requirements (qualitatively) in more detail in Chapter 8.

As can be seen in Figure 3.4, within the macroscopic modelling paradigm, differences in spatial assumptions are found in the way road capacity and spillback are modelled. We find that models considering both capacity and storage constraints to be among the most capable (i.e., can be applied to both uncongested and congested conditions), while capacity unrestrained models are among the least capable and are best suited for uncongested conditions only, see Section 3.5.2 for more details. However, we start our discussion with the most capable models from a spatial perspective; microscopic and mesoscopic models.

3.5.1 Microscopic and mesoscopic traffic flow propagation

Microscopic network loading models propagate individual vehicles through a *car following* model that is responsible for the longitudinal movement of each vehicle. In addition, a *lane changing* model guides the latitudinal movements. Together, they determine the location of each vehicle in space and time. The term car following stems from the behaviour of the considered vehicle, which is influenced by a lead vehicle.

3.5.1.1 Microscopic car following and lane changing models

The earliest car following models were of the *safety-distance* type (Pipes, 1953), although more recent examples of this model type also exist (Newell, 2002; Gipps, 1981). The idea behind these models is that the space headway to the lead vehicle determines its behaviour. The model proposed by Pipes (1953) is essentially identical to the model proposed by Forbes et al. (1958) in which space headways are replaced by time headways. To avoid collisions, the model

ensures that the time headway between vehicles exceeds the reaction time needed to match the speed of the lead vehicle.

Soon, it was found that only using headways to govern car following behaviour is rather restrictive. This led to the first *stimulus-response* models. Here, not only the headway is considered, but in addition, the vehicle's speed as well as the (absolute) difference in speed with the lead vehicle is taken into account. Stimulus-response models originated from research undertaken at General Motors (GM) and are therefore sometimes referred to as the GM model (Chandler et al., 1958). The GHR model later generalised the GM model (Gazis et al., 1961). The name stimulus-response originated from the fact that the observed speed difference is considered a stimulus, while the vehicle's acceleration is the response that follows from it.

Over time, more behaviour centric approaches emerged, among which the well-known *action-point* models, which are sometimes also classified as psychophysical models (Wiedemann, 1974; Brackstone and McDonald, 1999), these models recognise that there are certain triggers, or action points, that instigate different types of driver behaviour depending on, for example, how close a driver is to a lead vehicle. While computationally more costly, they can more closely resemble actual driver behaviour (Aghabayk et al., 2015; Brackstone and McDonald, 1999). There exist more model types than the three broad categories mentioned here, among them we find *optimal-velocity* models, *cellular automata models*, as well as the popular *Intelligent Driver Model* (IDM) by Treiber et al. (2000). We refer the reader to Wageningen-Kessels et al. (2014), for a more in depth discussion on this topic.

Car following models are supplemented with lane changing models to allow for mandatory and discretionary lateral movements that occur when overtaking or merging. Like car following models, early lane changing models were mostly deterministic rule based models (Gipps, 1986), but since lane changing is inherently linked to making choices, driver behaviour plays an important role. Discrete choice model based lane changing models try to take this into account. There are also increasingly more machine learning based and probabilistic approaches trying to capture the heterogeneity among driver populations, see for example Rahman et al. (2013), or Moridpour et al. (2010).

3.5.1.2 Mesoscopic traffic flow propagation

There is no single definition of what exactly a mesoscopic model is, except that they reside in between microscopic and macroscopic models. As a result, some conceptually very different approaches exist. The two dominant types of mesoscopic models involve *cluster based* - also known as *platooning* - approaches, and approaches where *individual vehicle arrival times* are predicted for link boundaries. The platooning approach is for example adopted in DYNASMART (Jayakrishnan et al., 1994). In DYNASMART packets of vehicles are propagated according to macroscopic principles to reduce computation times. Alternatively, one can propagate vehicles individually, but with greatly simplified behaviour, for example, by disallowing lane changes while traversing a road section. Examples of this particular approach can be found in DTALite (Zhou and Taylor, 2014), MATSIM (Strippgen and Nagel, 2009), and Mahut (2001). Analogies with gas-kinetic models also exist in the literature, but are less common in practice (Hoogendoorn and Bovy, 2001).

3.5.1.3 Heterogeneity in microscopic and mesoscopic models

Although we do not consider multi-modal nor multi-class approaches, it should be noted that when modelling individual vehicles, it is relatively easy to support multiple driver classes, vehicle types, as well as driving styles and attitudes, for example through modelling differences in anticipatory behaviour (Ossen and Hoogendoorn, 2007; Treiber et al., 2006; Lenz et al; 1999). One can also sample from a pool of driver characteristics and attach them to individual vehicles in the simulation, to achieve some desired distribution. Something which is far more difficult, if not practically impossible, to achieve in a macroscopic setting. At the same time, such potentially realistic approaches contribute to instability in modelling results, because the generation of vehicles, as well as the sampling of driver behaviour, is almost necessarily based on random processes. The absence of randomness, reduced computation times, and reduced calibration effort, are among the main attractors to adopt macroscopic network loading models instead of (heterogeneous) microscopic and mesoscopic approaches.

3.5.2 Macroscopic traffic flow propagation

Macroscopic dynamic models are either *first order* or *higher order*. A first order macroscopic dynamic model only considers instantaneous changes in speed (speed is the first derivative of location over time, hence the name). The LWR model is the most well-known example in this branch of models. Second order models on the other hand, do consider the effects of acceleration and deceleration. The idea behind higher order models is that they are able to capture more intricate behavioural aspects of drivers while still modelling average flow rates. This allows these models to include empirically observed effects such as capacity drops and stop-and-go-waves. Examples can be found in Aw and Rascle (2000), Messmer and Papageorgiou (1990), or Payne (1973). We however argue that such detailed behaviour is better suited to be replicated by mesoscopic and/or microscopic models and therefore limit ourselves to first order (dynamic) macroscopic models in the remainder of this work.

Within the macroscopic traffic flow propagation paradigm we identify the following four spatial model types in increasing order of simplification: *capacity and storage constrained*, *capacity constrained*, *capacity restrained*, and *capacity unrestrained*.

The simplest model type, yielding travel times invariant to link capacities, is termed *capacity unrestrained*. This typically only occurs when one only considers free flow travel times irrespective of the link volume. In practice, this approach is mainly used to initialise more capable assignment types, find potential paths, or for initial analysis purposes.

Capacity restrained models do consider link capacities, but only to a certain extent. These models still allow the demand, i.e. link volume, to exceed link capacity. As a result, in case of oversaturated link, no flow is held back even though it physically does not fit on the link. Instead of withholding flow from propagating, these models adopt link performance functions, such as the widely used BPR function (Bureau of Public Roads, 1964), or Akçelik function (Akçelik, 1991) to deter traffic from using oversaturated links. Hence, the name capacity restrained. While results of these approaches are physically infeasible in congested conditions, it does cater for strictly convex link cost functions. Also, the path travel times are link additive. This results in attractive mathematical properties that can guarantee the existence of a unique optimal solution. Because of this, there is still an active literature (Bar-Gera, 2010; Dial, 2006)

on solution schemes for capacity restrained approaches and these models are still widely used in planning applications, despite their well-known shortcomings.

To avoid the limitations of physically infeasible link volumes, *capacity constrained* models are a more capable alternative. Early capacity constrained approaches, in a static context, introduced side constraints where link flow was no longer allowed to exceed link capacity. A penalty function or Lagrange multiplier automatically reallocates flows from congested links to uncongested links, see (Shahpar et al., 2008; Larsson and Patriksson, 1995; Bell, 1995; Yang and Yagar, 1994). In most of these models, the penalties imposed are interpreted as queuing delay. However, the queue imposing this delay is not constructed from the demand, it is a mathematical construct and might therefore not be necessarily realistic nor feasible. In more recent residual queueing models, the queuing delay is derived from an actual (residual) queue, which is, arguably, a more natural way to construct the travel time. (Smith, 2013; Smith et al., 2013; Bifulco and Crisalli, 1998; Smith, 1987; Vickrey, 1969). These queues then drive the experienced delay instead of the more traditional link performance functions. Queues can either be modelled by letting them occupy physical space, or, alternatively, can be modelled as vertical (point) queues. The former is more realistic in general, but if the queues do not grow beyond the link length, vertical queues give exactly the same result as queues that are modelled physically. Notable is the work of Bliemer et al. (2014), who, for the first time, place the queue in front of the bottleneck analogous to dynamic models, whereas all aforementioned models place the queue inside the bottleneck. Besides the static vertical point queue models mentioned, there also exist dynamic macroscopic vertical point queue models, although these are hardly used in practice, for some example see Pang et al. (2012), Huang and Lam (2002), or Smith (1993).

In contrast to aforementioned point queue models, in *capacity and storage constrained* models, the impact of a (physical) queue exceeding the length, i.e. storage space, of a link is explicitly considered. In such cases, queues spill back upstream into preceding links, as would happen in reality. This type of model is therefore considered the most capable spatial model type. In practice, most approaches that consider spillback are dynamic (van der Gun et al. 2017; Raadsen et al, 2016; Han et al, 2012; Gentile, 2010; Yperman, 2007; Daganzo, 1994), but there do exist static and semi-dynamic models that takes storage constraints into account (Bliemer and Raadsen, 2017; Smith et al., 2013; Davidson et al., 2011).

3.5.2.1 Link models and Node models

Capacity restrained models are predominantly formulated on the link level and little attention is paid to the interactions that occur at intersections, i.e. nodes. These interactions became a topic of interest when capacity constrained models emerged. In capacity constrained models the available capacity on links exiting a node needs to be distributed across the competing incoming flows. This allocation of available capacity is the responsibility of the *node model*. The first macroscopic node models are applied on highly simplified networks and only dealt with either merging flows, i.e. on-ramps, or diverging flows, i.e. off ramps (Daganzo, 1994, 1995). Later, extensions to general intersections emerged (Smits et al., 2015; Tampère et al., 2011; Bliemer, 2007). The work by Tampère et al. (2011) is especially notable since it formulates a number of conditions to construct a so called general first order node model.

Today, in our view, any “proper” macroscopic traffic assignment model should include a node model so it can justify its distribution of flows across nodes.

The aforementioned node models are all designed for macroscopic assignment approaches. They assume average flow rates, average turn capacities, and consider the intersection to be a point, rather than an area. Conversely, in microscopic and mesoscopic models intersections are modelled spatially and consider vehicle interactions via, for example, gap acceptance factors (unsignalised intersections), or explicit signal phases (signalised intersections), resulting in a more capable model representation. However, we limit ourselves to considering only macroscopic node models in this work.

3.6 Traffic assignment from a behavioural perspective

On a macroscopic level, the behavioural aspect to traffic assignment is most prominent in how path choice is established. Here, we distinguish between *All-Or-Nothing* (AON) models, *one-shot* models and the well-known (user) *equilibrium* model types.

Within the considered UE paradigm, the static *Deterministic User Equilibrium* (DUE) approach (Wardrop, 1952) is still the most widely used. It assumes perfect knowledge on the part of the decision maker, which means we assume the driver knows the exact travel time for each path. A solution is found when “no traveller can unilaterally switch routes to improve its travel time”. Different approaches exist in the literature to determine the travel time conditional on the level of service of the network. Traditionally, link performance functions are used in combination with static capacity restrained models. More capable dynamic assignment methods are able to compute the delays more accurately by taking queue formation into account. This can for example be done by deriving the travel times from cumulative flow curves (Szeto and Lo, 2005).

Finding an equilibrium through simulation (when no closed form analytical solution is available) typically involves a solution scheme that relies on an iterative procedure. One of the reasons DUE is such a popular approach is the simplicity of its solution scheme. In a static context one finds the cheapest path - for example using Dijkstra’s algorithm - at the beginning of each iteration and shifts a portion of the demand onto the (newly) found path. The optimal portion of demand to shift can be determined analytically when the cost function is link additive and strictly concave, for example by using the Frank-Wolfe algorithm (Frank and Wolfe, 1956). In practice through, this shift in demand is often based on heuristics such as the Method of Successive Averages (MSA), or one of its many alternatives. The reason to resort to heuristics is based on the fact that the cost of finding the optimal step size can become larger than the benefits of a, per iteration, improved convergence rate. Downsides of traditional DUE solutions is the increasingly slow convergence around equilibrium, although there are contemporary methods that show significant improvements on this part (Bar-Gera, 2010, Gentile and Noekel, 2009). That said, the rather unrealistic assumption of perfect knowledge on the decision maker’s part remains.

Stochastic User Equilibrium (SUE) attempts to replace the behavioural assumption of perfect knowledge in DUE by proposing to act on imperfect knowledge instead (Fisk, 1980; Daganzo and Sheffi, 1977). Imperfect knowledge is modelled through the concept of perceived travel times. An error term is added to the path choice model to accommodate different perceptions across the decision makers. This results in a probability of choosing a potential path given its perceived cost. On an aggregate level this results in a distribution of the demand across the available alternative paths, where low probability paths acquire less trips than high probability paths. SUE is often paired with logit based path choice models to determine the path probabilities (McFadden, 1974; Sheffi, 1984; Train, 2003). The simplest and most widely used logit model is the Multinomial Logit Model (MNL), its popularity stems from its low computational cost and simple, yet flexible formulation. Its biggest limitation in the context of path choice is that it assumes that each alternative path is regarded as an independent choice, i.e. no path overlap is assumed. Other (closed form) path choice models such as a C-logit (Cascetta et al., 1996), path size logit (Bekhor et al., 2002), or paired combinatorial logit (Koppelman and Wen, 2000) aim to account for path overlap, but this comes at the cost of compromising the computationally friendly nature compared to MNL.

One difficulty with SUE approaches, irrespective of the chosen logit model, lies in the question what paths to consider. When using a logit model, every conceivable path will be allocated a non-zero probability of being used. Therefore, a selection has to be made on what paths to consider within the model; either a “representative” subset of considered paths can be constructed a-priori (Fiorenzo-Catalano, 2004), or alternatively, when updating the path set while iterating, some threshold needs to be considered where paths are no longer considered eligible (Watling and Rasmussen, 2015). Both approaches result in a *conditional equilibrium*, because they require an additional restricting condition on top of the stochastic equilibrium assumption. Some examples of various existing (equilibrium) models in the literature and how they classify against the spatial and temporal model types can be found in Table 3.2. There also exist many natural extensions of DUE and SUE to a dynamic context. However, we point out that the proposed methodology in Part II revolves around static traffic assignment procedures while Part III is mainly concerned with traffic assignment inputs rather than the actual procedures. Hence, while dynamic DUE and SUE traffic assignment models can be used in conjunction with the results of Part II of this thesis, the actual (dynamic) traffic assignment model specifications or properties are of less concern in this work and need not be discussed further.

A *one-shot* approach is a simplified version of the equilibrium approach. Its name correctly suggests that it only conducts a single iteration. Path choice can be either deterministic, stochastic, or some other approach. It is most common in operational (microscopic) traffic assignment models. It can for example be used to investigate the impact of non-recurrent traffic conditions such as accidents, where learning and route updating effects in the short run are assumed to be absent. Often an equilibrium result is used as the starting point. Note that even though one-shot approaches constitute only a single iteration, they can have multiple pre-trip path choice moments, in case the model is dynamic, and have multiple departure time periods. Given our focus on planning applications, one-shot approaches are not further considered.

Even simpler than a one-shot model, is the *AON* approach. Based on – often free flow – traffic conditions, the single cheapest path between each origin-destination pair is found and all traffic demand is loaded onto this single path. It is a fully deterministic approach and is often used as a starting point for DUE, bottleneck identification, or gain preliminary insight in the problem at hand.

Table 3.2: Some existing models per spatial-temporal model type assumption combination.

| | | | Temporal model types | | |
|---------------------------|-----------------------------------|-------------|------------------------------|---------------------------------------|-------------------------------|
| | | | Dynamic | Semi-dynamic | Static |
| Spatial model types | Capacity & storage constrained | Microscopic | Gipps, 1981 | - | - |
| | Capacity & storage constrained | Mesoscopic | Strippgen and Nagel, 2009 | - | - |
| | Capacity & storage constrained | Macroscopic | Daganzo, 1994 | Davidson et al., 2011 | Bliemer and Raadsen, 2017 |
| | Capacity constrained | Macroscopic | Friesz et al., 2013 | Akamatsu et al., 1998 ³ | Bifulco and Crisalli, 1998 |
| | Capacity restrained | Macroscopic | Janson, 1991 | Nakayama et al., 2012 | Beckmann, 1956 |
| | Capacity unrestrained | Macroscopic | - | - | AON |

³ In Japanese, classification based on abstract and discussion in other peer-reviewed Japanese journal articles.

Part II

4 Link travel time decomposition

Travel time delays and their formulation are key in the decomposition method proposed in Part II of this thesis. Different types of delay exist, each of them with their unique formulation and properties that might or might not be useful to adopt, depending on the chosen application context. In this chapter we discuss various types of delay in the context of strategic planning models from the perspective of a single link. Because we solely focus on strategic traffic assignment models for planning purposes here, we only consider uniform vehicle arrival rates and assume that traffic is modelled via average flow rates. The concepts and definitions of delay introduced in this chapter are then adopted in Chapter 5 to design and formulate our travel time decomposition method which is subsequently extended to the path and network level.

In Section 4.1 we relate the fundamental diagram to the concept of delay, followed by a discussion of two different type of link delay decomposition methods in Section 4.2. Then, in Section 4.3, we briefly discuss some of the additional complexities involved in modelling delay in the presence of signalised intersections.

4.1 The fundamental diagram and delay

To get a better understanding of delay, we first revisit the general concave fundamental diagram introduced in Chapter 3. The fundamental diagram typically has two branches, a *hypocritical branch*, i.e. uncongested branch, and a *hypercritical branch*, i.e. congested branch (Cascetta, 2009). The hypercritical branch represents congested link states where queues start to form or are already present. Furthermore, it embodies the part of the fundamental diagram where flow decreases with increasing density, see Figure 4.1(a).

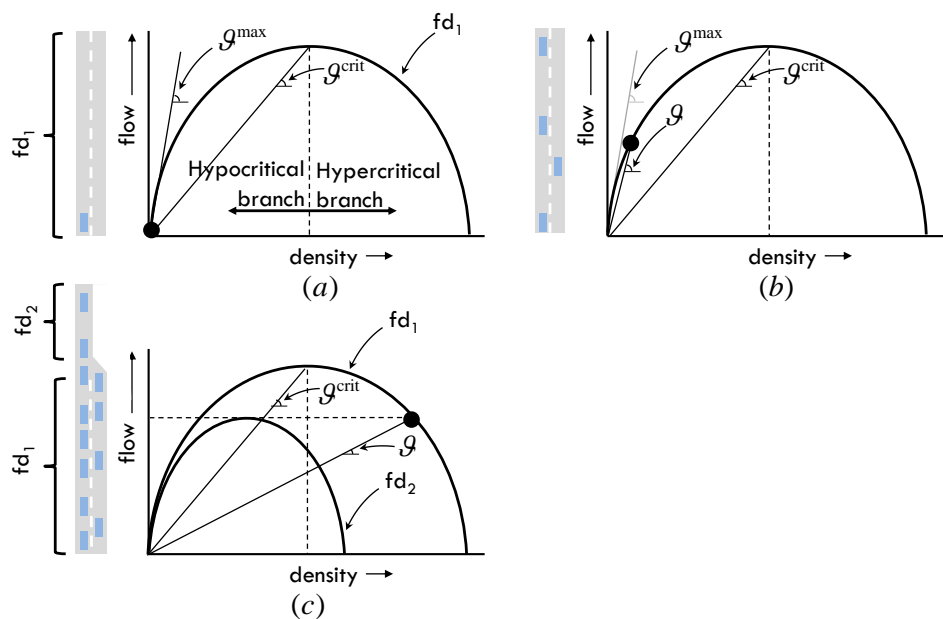


Figure 4.1: Schematic impression of the different components of travel time on the link level (a) free flow travel time, (b) non-zero hypocritical delay, (c) non-zero hypercritical delay.

The hypocritical branch on the other hand, represents all uncongested link states. In uncongested link states, flow increases with increasing density. In the absence of any delay, the link is by definition in an uncongested (hypocritical) state and vehicles travel at the absolute maximum allowed speed on the link, denoted by \mathcal{G}^{\max} . This then results in an absolute minimum link travel time. In the situation of a link being in a hypocritical uncongested state, the accompanying vehicle speed \mathcal{G} (km/h) must reside between $\mathcal{G}^{\text{crit}} \leq \mathcal{G} \leq \mathcal{G}^{\max}$, where critical speed $\mathcal{G}^{\text{crit}}$ is the steady state speed experienced when flow is at capacity, see Figure 4.1(b). When a link is in a congested state vehicle speeds are found to be $0 \leq \mathcal{G} \leq \mathcal{G}^{\text{crit}}$, as depicted in Figure 4.1(c). Clearly, both in uncongested and congested link states, the vehicle speed is most likely to be smaller than \mathcal{G}^{\max} and hence some form of delay is experienced by the traveller.

It is important to note that the shape of fundamental diagram has an impact on the magnitude of delay and/or the presence of certain types of delay. Newell (1993) for example, proposed a triangular shaped fundamental diagram, in such a diagram $\mathcal{G}^{\max} = \mathcal{G}^{\text{crit}}$, resulting in an uncongested branch where the vehicle speed is invariant to the flow rate as long as the link is in a hypocritical state. In other words, there exists no delay on the hypocritical branch of this fundamental diagram, see Figure 4.2(a). Daganzo (1994) proposed a trapezoidal fundamental diagram, where vehicle speed is also largely invariant to flow up to a point where it slowly deteriorates before moving to a congested state. In this case there does exist delay on the hypocritical branch, but only when the flow equates to capacity, see Figure 4.2(b).

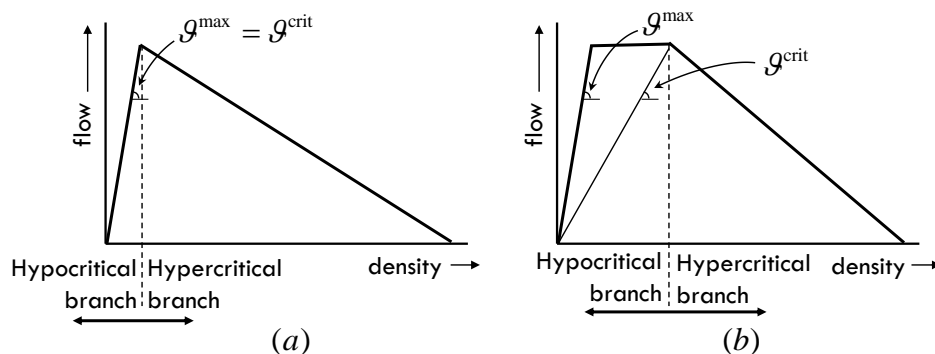


Figure 4.2: (a) Triangular fundamental diagram, (b) trapezoidal fundamental diagram.

4.2 Link travel time in strategic planning models

We discuss two different methods to classify and extract delay from the results of a traffic assignment model. Both approaches decompose the link travel time based on certain delay characteristics. The first is termed *physical link delay decomposition* and aims to replicate delay as it is experienced in reality by the traveller traversing the link, the second is termed *functional link travel time decomposition* and revolves around a more abstract approach. Both methods however yield the exact same total travel time.

4.2.1 Physical link travel time decomposition

In physical travel time decomposition we explicitly separate travel time based on how it is experienced by the traveller. This is useful when applying different behavioural weights to these components, similar to generalised cost functions for public transport in which waiting

time during transfers leads to more disutility per minute than in-vehicle travel time. Consider the example in Figure 4.3, here the available downstream supply is not sufficient based on the incoming traffic flow. As a result the cumulative inflow curve, i.e. total number of vehicles that passed the upstream link border, and outflow curve, i.e. total number of vehicles that passed the downstream link border, are no longer parallel. Based on these two cumulative curves, information can be extracted on link travel times, link densities, and flows, for example to perform post-simulation analysis (Szeto and Lo, 2005).

For an individual traveller, its total link travel time is determined by the time difference between entering and leaving the link, i.e. the horizontal (red) line separating the two cumulative curves. In this situation, the traveller experiences uncongested conditions, i.e. free flow travel time in a hypocritical state, for part of the link, while at some point it joins the backward propagating queue. From this point onward, until departing the link, the traveller experiences a congested hypercritical flow state with a lower speed.

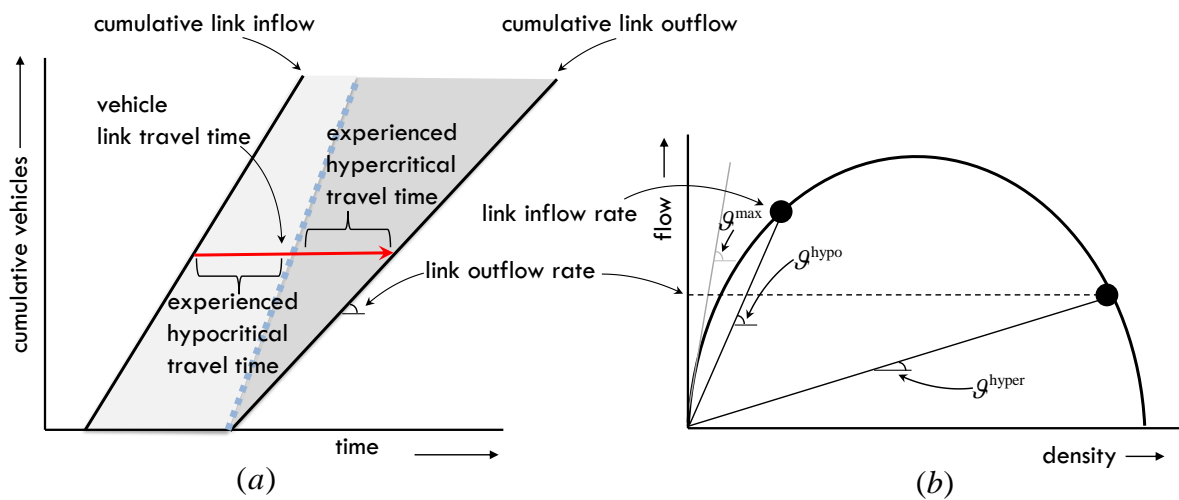


Figure 4.3: (a) Physical delay link travel time decomposition, (b) example concave underlying fundamental diagram

In this approach both link travel time components are flow dependent and the explicit tracking of the tail of the queue is required, making it a computationally challenging approach. Once the queue reaches the upstream border of the link, the entire link is in a hypercritical state and the queue spills back into the preceding link(s).

4.2.2 Functional link travel time decomposition

In a functional link travel time decomposition there is less of a direct relationship between what a traveller experiences and how the various components of the travel time are built up. Instead, the approach puts computational convenience on the forefront, resulting in a somewhat more abstract decomposition. This approach does not need to explicitly track the tail of the queue and is best suited for situations when one is interested in the total travel time only and there is no difference in weighting between the experienced hypocritical and hypercritical travel times involved.

When adopting a general concave fundamental diagram, a functional decomposition of travel time comprises three components: (i) minimum link travel time, (ii) *functional hypocritical delay*, and (iii) *functional hypercritical delay*. An illustrative example is provided in Figure 4.4(a). The minimum link travel time is constant and based on g^{\max} as mentioned before, while the delay components are flow dependent. The functional hypocritical delay should not be confused with experienced hypocritical delay since it has little to do with how a traveller experiences the link state spatially. Functional hypocritical delay merely reflects the additional delay one would have experienced on top of the minimum travel time if the entire link were to be in an uncongested state (which might or might not be the case). Note that we can always add this delay even if part of the link is in a congested state. For example, in Figure 4.4(a), the functional hypocritical delay is based on speed g^{hypo} even though the vehicle speed when exiting the link is in fact g^{hyper} see Figure 4.4(b). The functional hypocritical delay remains constant as long as the flow is constant, something which is not the case for experienced hypocritical delay in a congested situation. However, once the inflow rate does change, the functional hypocritical delay is also affected, hence it is not invariant to flow.

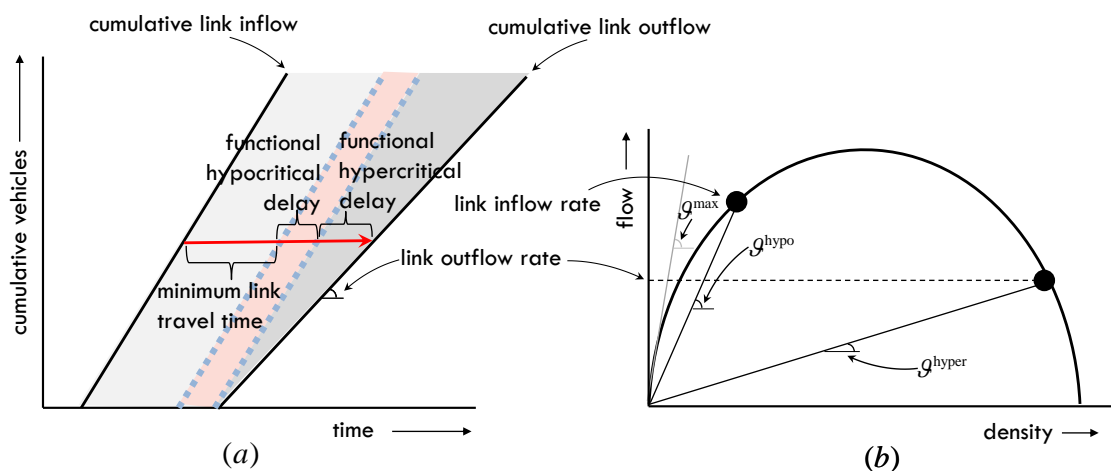


Figure 4.4: (a) Functional link travel time decomposition, (b) example concave underlying fundamental diagram.

The last component, the functional hypercritical delay, is the additional travel time it takes to depart from the link beyond the minimum link travel time supplemented with the additional hypocritical delay. Observe that this is indeed different to the experienced hypercritical delay in Figure 4.3 even though the total travel time in both cases is exactly the same.

In a functional travel time decomposition, when adopting a general concave fundamental diagram, the first component is flow invariant, while the two delay components are not. In case one adopts a simplified triangular fundamental diagram, then the functional hypocritical delay vanishes, leaving only the functional hypercritical delay as a flow dependent delay component, see Figure 4.5. Generally, the functional hypocritical delay is dominated by both the minimum travel time as well as the hypercritical delay (if any) and therefore adopting a simplified triangular fundamental diagram can be an attractive proposition if it allows one to simplify the underlying methodology, as we will see in Chapter 5.

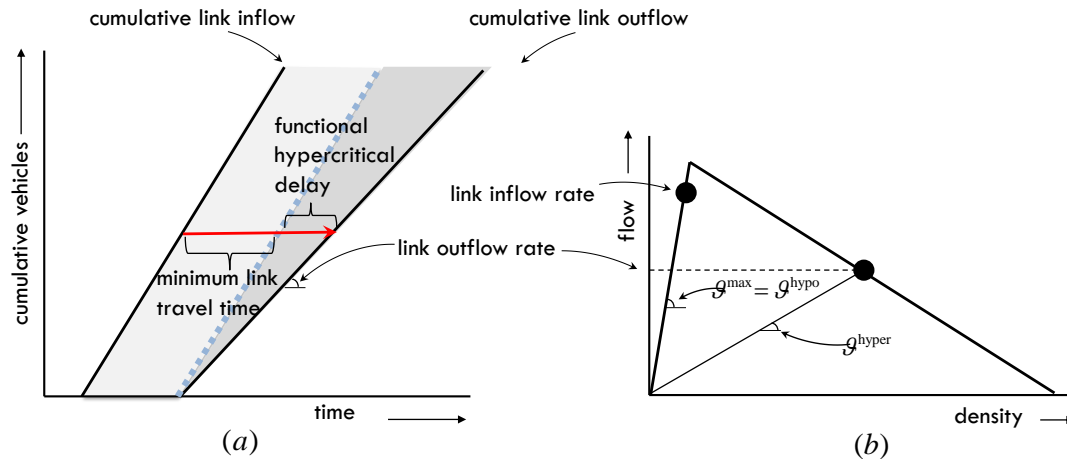


Figure 4.5: (a) Functional link travel time decomposition, (b) example triangular underlying fundamental diagram.

4.3 Delay at signalised intersections

The hypercritical delay discussed so far is due to a lack of capacity at the downstream end of the link, this can be caused by, for example, a reduction in number of lanes downstream. However, in practice, most hypercritical delay is the result of encountering (signalised) intersections that require a vehicle to slow down or even stop. At the disaggregate level, i.e. on a per vehicle basis, this intersection based delay is decomposed into a number of different types such as stopped time delay, approach delay, time-in queue delay, or control delay, see Matthew (2014). However, we choose to only discuss the effects of aggregate intersection delays due to our focus on planning models.

4.3.1 Overflow delay

The most common way to take the effects of signalised intersections into account in an aggregate setting is to impose some kind of reduction in (turn) capacity based on the available average green time, possibly specific to a particular period of time within the simulation. Hence, only when the arrival rate of vehicles at the signal exceeds this reduced capacity, a queue starts to form. The delay resulting from this queue is termed *overflow delay* (Webster, 1958), although we prefer the term *persistent delay*, because it reflects delay that does not dissolve under the given travel demand pattern, but persists instead, see Figure 4.6(a) for an example. From an intersection perspective one is only interested in the discrepancy between arriving at the intersection, i.e. arriving at the downstream link border, and traversing the intersection, i.e. departing from the downstream link border. We therefore use a specific cumulative plot, named Input/Output (I/O) diagram. In an I/O diagram both curves reflect the downstream end of the link, one denoting the Input (I), i.e. arrivals, while the other depicts the Output (O), i.e. departures. As can be seen in Figure 4.6(b), in the presence of persistent delay the surface area resulting from a lack of (average) green time only increases. In general, to obtain the average delay per vehicle, one needs to compute the total surface area of persistent delay and divide it by the number of vehicles that traversed the link. Finally, we also see that the persistent delay as it is used here corresponds 1:1 with the functional hypercritical delay as discussed in the previous section.

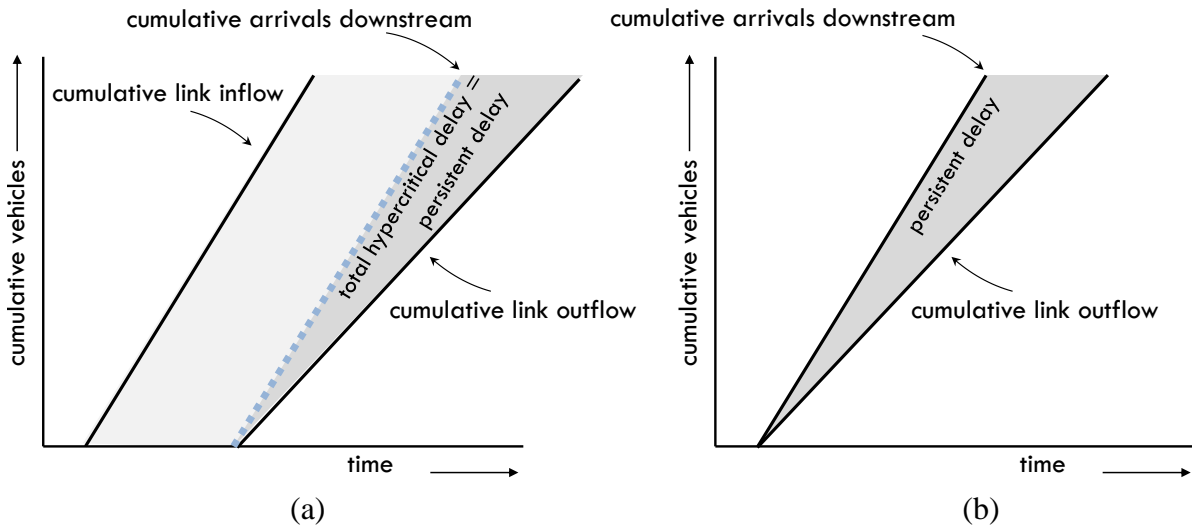


Figure 4.6: (a) persistent delay in cumulative plot, (b) alternative I/O plot of the same delay.

If an intersection is unsignalised, but adheres to a set of rules such as give way or all-stop, one can also approximate average capacity reductions based on some rules (Bovy 1991; Akçelik and Troutbeck, 1991), many such rules have been standardised through the Highway Capacity Manual (HCM, 2000).

4.3.2 Uniform delay

In addition to persistent delay, one might be interested to consider *uniform delay* as well for signalised intersections. Uniform delay is delay that occurs when one has to wait for a traffic signal in an undersaturated situation. In planning models where signal phases are not modelled explicitly, uniform delay cannot be captured within the assignment process itself, simply because in undersaturated conditions, travel demand never experiences a queue since the average capacity that is available suffices. However, if one were to model the phases explicitly, inevitably some portion of the arriving downstream vehicles experiences a red signal and accrues some uniform delay, as is depicted in Figure 4.7(a).

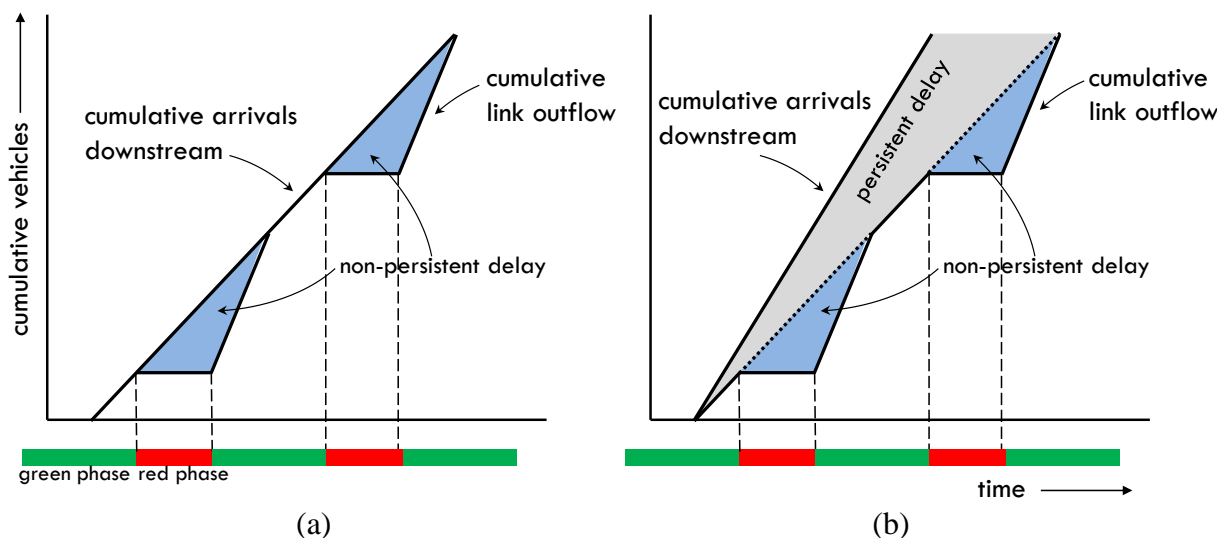


Figure 4.7: Uniform, i.e. non-persistent delay, under explicit signal phases for (a) undersaturated condition, (b) oversaturated condition.

The reason this delay is termed uniform is because it assumes a uniform arrival of vehicles. Clearly, this uniformity of arrival has nothing to do with the actual type of delay, we therefore prefer to use the term *non-persistent delay* instead, to reflect the fact that this delay is dissolved within a single cycle of the signal. When the arrival rate increases to the point that it exceeds the available (exit)capacity, non-persistent delay is supplemented with persistent delay, an example of which is provided in Figure 4.7(b). If one wishes to incorporate non-persistent delay in planning models that do not explicitly model signal phases (which is almost always the case), one typically supplements the travel times with an estimate of the non-persistent delay in a post-processing step. This delay can be calculated based on the arrival rate, effective green times, cycle time of the signal, and capacity or saturation flow of the link through the well-known formula of Webster (1958). As mentioned, we only consider persistent delay in this work, due to exit-link capacity restrictions rather than restrictions caused by traffic signals. Yet, it is clear that including traffic signal based capacity restrictions, would be a trivial extension to make.

5 Extracting delay subnetworks under varying demand and fixed supply

In Chapter 1 it was argued that any methodology aimed at altering the representation of traffic assignment should consider the application context it is applied for. Only then, its representation can be optimised to yield both a capable and minimal outcome. Here, we choose our application area of interest to be applications with a fixed supply side, i.e. physical road network, while demand may vary. As mentioned, the best known applications that comply with this restriction are quick-scan methods, matrix calibration methods, analysis of travel time variability due to varying demand, but also bi-level optimisation methods fall under this category, where the lower level consists of a traffic assignment model and the upper level, for example, constitutes the testing of different control strategies. A common characteristic of all these applications is their computationally costly nature. As a result, this application domain is still dominated by static traffic assignment methods because more sophisticated approaches are considered to be computationally infeasible. However, even when using static traffic assignment, not all the desired scenarios can be investigated due to time constraints. Our aim therefore is to develop methodology that results in an equally capable static traffic assignment representation, but one that is truly minimal, i.e. we focus on minimizing the computational burden with the least possible sacrifice in information loss.

Quick-scan applications are not concerned with highly detailed outputs, but instead use more aggregate indicators such as general accessibility measures that are predominantly obtained through (total) travel times. Matrix calibration procedures use a combination of data sources to adjust the demand, among which are measured link volumes and trajectory travel times. Analysing travel time variability typically means simulating many different travel demand scenarios and computing resulting travel times. Similarly, in applications where assignment is used as a lower level, often only costs, i.e. total travel times, are fed back to the upper level. Given the central role that travel time plays in these applications we focus on minimising the information loss with respect to the original travel times.

We propose a decomposition based approach. One of the main distinctions between aggregation and decomposition, as argued in Chapter 1, is found in the fact that decomposition methods do not necessarily require additional simplifying assumptions to reduce the computational burden. Instead, they rely on procedural/functional changes to achieve their goal. This in contrast to aggregation methods which group individual data points, i.e. links, nodes, zones, paths, to reduce the computational effort, which inevitably lead to a degradation in result accuracy. We therefore focus on a decomposition based approach because they, potentially, allow for a lossless representation altering method while still being able to reduce the computational cost.

The remainder of this chapter proposes a novel decomposition procedure based on the concept of a *delay subnetwork*. We construct this delay subnetwork by decomposing path travel times into a minimum travel time part and a delay part and use this decomposition to extract the related subnetworks. In addition, we propose a procedure to consolidate paths in the delay

subnetwork to further reduce the computational cost. Applicability of this procedure is demonstrated via hypothetical examples for both traditional capacity restrained static assignment and static capacity constrained assignment with point queues based on the model by Bliemer et al. (2014). Discussion of real world case is found in Chapter 6.

For completeness, we explicitly state the assumptions relevant to this chapter: (i) The transport network is assumed given and fixed across all demand scenarios, (ii) traffic assignment is path based, (iii) the path set is generated a-priori and fixed across demand scenarios, (iv) the conditional⁴ stochastic user equilibrium (SUE) approach is adopted. (v) Path choice is based on deploying the multinomial logit model (MNL), (vi) generalised costs are based on total path travel time only, (vii) spillback effects, i.e. physical queues are not considered, and (viii) only a single user class in the form of private vehicles is considered. The last assumption is not strictly necessary as our method can also be applied to multi-class assignment, but it simplifies exposition of our method. The same holds for the adoption of MNL as our path choice model. The usefulness of adopting the SUE approach in combination with the a-priori path set is discussed in further detail in Section 5.3.

We first discuss our decomposition method in relation to the general framework (Chapter 2) and utilise the concepts discussed in Chapter 4 to introduce the reader to, and formalise the concept of, a travel time decomposition method impacting on the traffic assignment representation as a whole, in Section 5.1. In Section 5.2 we introduce notation pertaining to the network infrastructure followed by the demand side formulation of the model in Section 5.3. The computational gain that can be achieved, as well as the magnitude of information loss suffered when applying our method is highly dependent on the adopted traffic assignment procedure, to highlight this, we briefly reiterate the traditional static capacity restrained model in Section 5.4. Then, in Section 5.5, we discuss the capacity constrained residual point queue model by Bliemer et al. (2014). It is shown that the underlying assumptions of the two models – which are substantially different - dictate how delay is measured and how it drives their suitability for the travel time decomposition method. The model by Bliemer et al. is a demonstrably better fit and is therefore used in the subsequent case studies. The decomposition method itself is then further formalised in Sections 5.6 and 5.7. In Section 5.8 we discuss the extension to multiple demand scenarios and how to construct a single representative delay subnetwork based on the available demand scenarios. This is followed by some final remarks on the subject in Section 5.9.

5.1 Travel time decomposition

To extend and formalise the concept of travel time decomposition following Chapter 4 we first consider only a single demand scenario⁵. Further, the proposed travel time decomposition is functional in nature based on the discussion in Chapter 4). There are a number of compelling reasons to prefer this type of decomposition over a physical travel time decomposition. First, our objective is to maximise the reduction in computational cost to solve the model. As discussed, a physical travel time decomposition is computationally more complex as none of

⁴ The SUE is *conditional* since it adopts a fixed path set to which the SUE is limited to.

⁵ The extension to multiple demand scenarios is postponed until Section 5.8.

the decomposed components are flow invariant. Second, the adopted application context only considers total travel time to drive the path choice, hence weighting of the experienced travel time components is absent and we can adopt the more abstract functional decomposition without any “behavioural” penalty. Third, we do not consider spillback, so there is little benefit in the explicit tracking of the tail of the queue as one does in a physical decomposition.

5.1.1 Path travel time decomposition

Let us now extend and formalise the decomposition of a functional link travel time decomposition to the path (and network level). To do so we follow the general structure of our traffic assignment representation framework proposed in Chapter 2. On the link level, functional travel time decomposition distinguishes between the minimum travel time and additional delay, where the latter consists of functional hypocritical delay and functional hypercritical delay. We do the same on the path level. For each path p , its path cost h_p is a combination of minimum travel time h_p^{\min} and (functional) delay h_p^{delay} , where the functional delay consists of both hypocritical delay h_p^{hypo} and hypercritical delay h_p^{hyper} . In general, we adopt a vector based notation where:

$$\mathbf{h} = \mathbf{h}^{\min} + \mathbf{h}^{\text{delay}}, \quad (5.1)$$

with:

$$\mathbf{h}^{\text{delay}} = \mathbf{h}^{\text{hypo}} + \mathbf{h}^{\text{hyper}}, \quad (5.2)$$

where $\mathbf{h}, \mathbf{h}^{\min}, \mathbf{h}^{\text{delay}}, \mathbf{h}^{\text{hypo}}, \mathbf{h}^{\text{hyper}} \in \mathbb{R}_+^{P \times 1}$, respectively. We can then reformulate network loading function $\Phi(\cdot)$ like the following:

$$\mathbf{h}^{\min} + \mathbf{h}^{\text{delay}} = \Phi(\mathbf{f} \mid \mathbf{A}, \mathbf{P}), \quad (5.3)$$

where network \mathbf{A} is an input, as well as paths \mathbf{P} . The latter are considered input because of our assumption regarding the construction of an a-priori path set. Because we can decompose the resulting path travel times, we now attempt to decompose the network loading procedure as well, based on the same premise, where we know that the minimum path travel time is by definition invariant to flow rate \mathbf{f} , yielding:

$$\Phi^{\min}(\mathbf{A}, \mathbf{P}) + \Phi^{\text{delay}}(\mathbf{f} \mid \mathbf{A}, \mathbf{P}) = \Phi(\mathbf{f} \mid \mathbf{A}, \mathbf{P}), \quad (5.4)$$

such that $\mathbf{h}^{\min} = \Phi^{\min}(\mathbf{A}, \mathbf{P})$ and $\mathbf{h}^{\text{delay}} = \Phi^{\text{delay}}(\mathbf{f} \mid \mathbf{A}, \mathbf{P})$. Substituting Equation (5.4) into path choice function $\Psi(\cdot)$ yields:

$$\mathbf{f} = \Psi\left(\left(\Phi^{\min}(\mathbf{A}, \mathbf{P}) + \Phi^{\text{delay}}(\mathbf{f} \mid \mathbf{A}, \mathbf{P})\right) \mid \mathbf{D}\right) = \Psi\left(\Phi^{\min}(\mathbf{A}, \mathbf{P}) + \Phi^{\text{delay}}(\mathbf{f} \mid \mathbf{A}, \mathbf{P}, \mathbf{D})\right). \quad (5.5)$$

Because the minimum path travel time is invariant to demand \mathbf{D} , it can be excluded from the interdependency between path choice and network loading. The delay of course does remain

conditional on demand, i.e. flow rates, and still needs equilibrating. Hence, this formulation, while correct, does not yet yield any computational gain; the same network, path set, and interaction between demand and supply is considered to find a solution. That said, there is potential to significantly reduce computation times based on the following two observations. First, in general networks, only a minority of the links is typically saturated while causing the majority of the delay. If we somehow can identify this subset of locations and use this, much smaller, delay subnetwork to solve $\Phi^{\text{delay}}(\cdot)$, the equilibration process would be simplified considerably. Second, when not considering just a single demand scenario, but a great many demand scenarios, the majority of the bottlenecks causing path delays can be assumed to remain the same. This is especially true given our application context of matrix calibration, travel time variability, and quick-scan applications that consider many, but only slightly varying demands. Since each demand scenario can be assumed to only differ marginally, the identification of a relatively modest superset of delay infrastructure comprising all potential bottlenecks across demand scenarios is deemed feasible. Therefore, we aim to construct a single, comparatively small, delay subnetwork $\mathbf{A}^{\text{delay}} \in \mathbb{F}_2^{N \times N}$, that is able to capture the original path travel time delays across all demand scenarios considered. This reduced delay subnetwork representation utilises a simplified path representation $\mathbf{P}^{\text{delay}} \in \mathbb{F}_2^{P \times A}$. Simplified paths result in computational gains since the computational cost of *path based* network loading (in static traffic assignment) is in fact driven by its path complexity⁶. We rewrite Equation (5.5) to reflect this approach:

$$\mathbf{f}^s = \Psi\left(\Phi^{\min}(\mathbf{A}, \mathbf{P}) + \Phi^{\text{delay}}(\mathbf{f}^s \mid \mathbf{A}^{\text{delay}}, \mathbf{P}^{\text{delay}}, \mathbf{D}^s)\right), \quad s \in \{1, \dots, S\}, \quad (5.6)$$

where the number of demand scenarios is denoted by S . We emphasize that we formulated Equation (5.6) such that we reuse the same delay subnetwork across all demand scenarios s . Clearly, the challenge here lies in the identification of the minimal delay subnetwork that is capable of reproducing the original delay that would have resulted when adopting the original network and original path set for each demand scenario.

The number of paths in $\mathbf{P}^{\text{delay}}$ is equal to the number of paths in \mathbf{P} , only the number of links in a path might have decreased as a result of the reduced delay subnetwork $\mathbf{A}^{\text{delay}}$. As we will see in Section 5.7, due to this reduced representation, the overlap between paths increases significantly. A further reduction in computational cost can therefore be achieved by exploiting this increased path overlap. Consequently, we propose a novel path consolidation procedure, resulting in a minimal path set representation. The paths in this minimal path set are termed *equidelay paths*, denoted by $\mathbf{P}^{\text{equidelay}} \in \mathbb{F}_2^{P \times A}$, again referring to Section 5.7 for details. Adopting this alternative path set does introduce a complication within the traffic assignment procedure itself. On the one hand, path choice operates on the original path set, but is now provided with path travel times for the consolidated path set. On the other hand, the network loading procedure receives path flows for the original path set, while it adopts the consolidated path set to perform the network loading. To correctly map original paths to consolidated paths and vice versa, the path choice procedure and network loading procedure for the delay subnetwork need

⁶ When network loading is path based, flows are loaded on link on a per path basis, see for example Bliemer et al. (2014), hence the number of paths as well as the number of links per path determine the computational effort required to carry out the network loading, see also the results presented in Chapter 6.

to be altered accordingly. For now we assume this is possible by adopting an altered path choice function $\Psi^{\text{equidelay}}(\cdot)$ and network loading function $\Phi^{\text{equidelay}}(\cdot)$, respectively, yielding:

$$\mathbf{f}^s = \Psi^{\text{equidelay}}\left(\Phi^{\min}(\mathbf{A}, \mathbf{P}) + \Phi^{\text{equidelay}}(\mathbf{f}^s \mid \mathbf{A}^{\text{delay}}, \mathbf{P}^{\text{equidelay}}, \mathbf{D}^s)\right), \quad s \in \{1, \dots, S\}. \quad (5.7)$$

Equation (5.7) can be considered as a special case of our general framework, discussed in Chapter 2. To do so we slightly modify our shorthand notation for the traffic assignment components $\mathcal{M}(\cdot)$ to make it suitable for a multiple demand scenario setting such that:

$$\mathcal{M} = \left(\mathbf{A}, \mathbf{Z}, \mathbf{P}, \Psi(\cdot), \Phi(\cdot) \mid \{\mathbf{D}^s \mid s \in \{1, \dots, S\}\}\right). \quad (5.8)$$

Further, in the original framework, the representation optimisation problem considers all possible “rules” governing a change in representation (Equation 2.6). In this case however, we formulate a single fixed procedure yielding just one “rule” denoted $\gamma^{\text{equidelay}}$. Furthermore, because the procedure is fixed and assumed to be lossless, there is no error function $\varepsilon(\cdot)$ to consider. This then, leaves only a single instance of the inverse scaling function to consider, i.e. $\zeta(\mathcal{M}, \Xi_{\gamma^{\text{equidelay}}}(\mathcal{M}))$. The rule specific representation function $\Xi_{\gamma^{\text{equidelay}}}(\mathcal{M}) = (\mathbf{A}^{\text{delay}}, \mathbf{P}^{\text{equidelay}}, \Psi^{\text{equidelay}}(\cdot), \Phi^{\min}(\cdot), \Phi^{\text{equidelay}}(\cdot))$ and given that we are only interested in the resulting traffic assignment components we can reduce $\zeta(\mathcal{M}, \Xi_{\gamma^{\text{equidelay}}}(\mathcal{M}))$ even further, such that our entire decomposition method can simply be denoted by:

$$\Xi_{\gamma^{\text{equidelay}}}(\mathcal{M}). \quad (5.9)$$

Before we proceed with formalising the decomposition method itself, we introduce the reader to the network infrastructure, the two considered traffic assignment procedures considered, and the impact of each procedure on the effectiveness of the proposed decomposition method.

5.2 Network infrastructure

Recall transport network matrix $\mathbf{A} \in \mathbb{F}_2^{N \times N}$, where each non-zero cell denotes a link between two nodes. We construct this matrix via:

$$\mathbf{A} = \mathbf{A}^+(\mathbf{A}^-)^T, \quad (5.10)$$

with the mapping from nodes to their incoming links via indicator matrix $\mathbf{A}^- \in \mathbb{F}_2^{N \times A}$, while outgoing links are mapped similarly via $\mathbf{A}^+ \in \mathbb{F}_2^{N \times A}$. The transpose of a matrix is denoted by T. Each link a in the network has a length ℓ_a (km), maximum speed $\mathcal{G}_a^{\text{max}}$ (km/h) and capacity q_a^{max} (veh/h). Recall that each path $p \in \{1, \dots, P\}$ is mapped to the network via $\mathbf{P} \in \mathbb{F}_2^{P \times A}$. Mappings between a path and its departing, arriving zone $z \in \{1, \dots, Z\}$ are given by indicator matrices $\mathbf{P}^+, \mathbf{P}^- \in \mathbb{F}_2^{Z \times P}$, respectively. A turn between link a and a' is denoted by link pair (a, a') , which, on the network level, is formalised by link-to-link turn indicator matrix $\mathbf{M} \in \mathbb{F}_2^{A \times A}$, where we can construct \mathbf{M} via:

$$\mathbf{M} = (\mathbf{A}^-)^T \mathbf{A}^+ \quad (5.11)$$

The path specific turn mapping is denoted by $\mathbf{M}^p \in \mathbb{F}_2^{A \times A}$, this mapping only includes non-zero indicators if part of path p , where necessarily:

$$\mathbf{M} = \mathbf{M}^1 \parallel \mathbf{M}^2 \parallel \dots \parallel \mathbf{M}^{p-1} \parallel \mathbf{M}^p, \quad (5.12)$$

where operator \parallel denotes the *logical or* operator. Logical operators can be employed on indicator matrices because they are Boolean matrices at the same time. An example on the turn based formulation is provided in Figure 5.1.

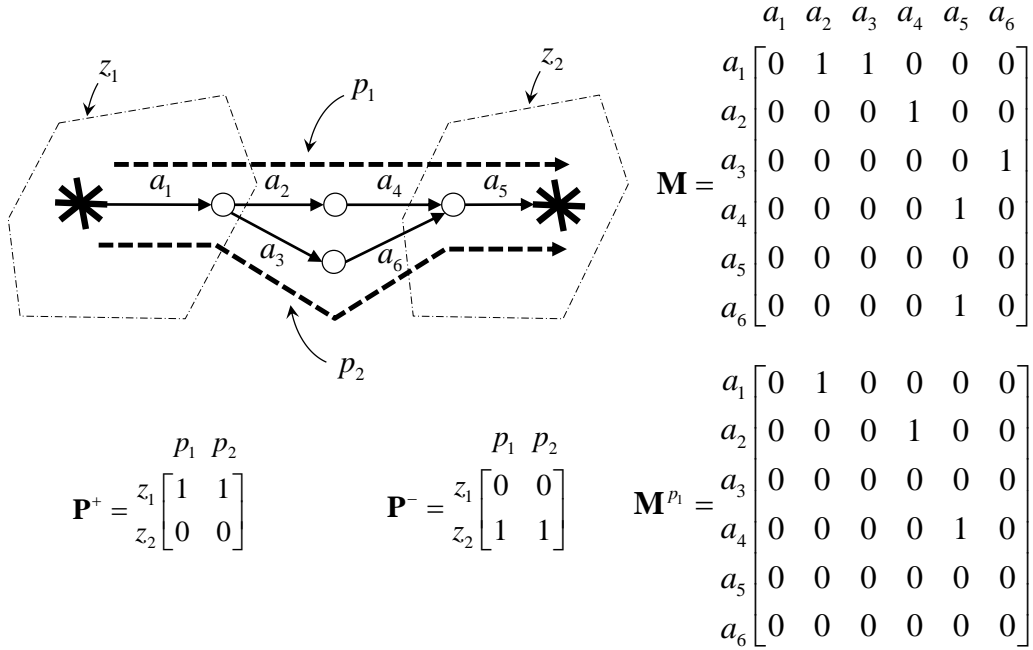


Figure 5.1: Example of turn mapping and path departure/arrival zone mapping.

5.3 Path choice and SUE

We adopt an SUE approach where the path set is generated a-priori. The combination of SUE and the adoption of a fixed path set is quite common in a planning context, because SUE yields non-zero flows on all paths considered. Enumerating all possible paths in assignment is infeasible for any but the smallest networks. It is argued one should therefore only consider *relevant* paths (Bliemer and Bovy, 2008). A large body of literature exists on path set generation, see for example Watling et al. (2015) or Fiorenzo-Catalano (2004). For now, we assume relevant paths have been identified adopting some path choice set generation procedure. We return to this topic in Section 5.8 because the construction of the relevant path set is important in identifying our, single, delay subnetwork. To determine path choice proportions the MNL model is adopted. Other logit models are equally compatible with our methodology, but since our focus is not on path choice as such, we choose to adopt a basic yet widely used approach. In MNL, path flows f_p are determined via:

$$f_p = \left(\frac{\exp(-\theta h_p)}{\sum_{p'=1}^P P_{zp'}^+ P_{z'p}^- \exp(-\theta h_{p'})} \right) D_{zz'}, \quad \forall p : P_{zp}^+ P_{z'p}^- = 1 \text{ with } z, z' \in \{1, \dots, Z\}, \quad (5.13)$$

where $P_{zp}^+ P_{z'p}^- = 1$ merely selects the departing and arriving zone of path p such that it is consistent with the alternative paths considered in the denominator. Scale parameter θ denotes the magnitude of the perception error for each traveller, with 0 resulting in a random choice and ∞ yielding a fully deterministic choice. The equilibrium path flow demand vector \mathbf{f}^* can be found by solving the following variational inequality problem (Chen, 1999):

$$\begin{aligned} \sum_{p=1}^P (h_p(\mathbf{f}^*) + \theta^{-1} \ln f_p^*) (f_p - f_p^*) &\geq 0, \quad \mathbf{f} \in \Omega, \\ \text{s.t.} \\ f_p &\geq 0, \quad p \in \{1, \dots, P\}, \\ D_{zz'} &= \sum_{p=1}^P P_{zp}^+ P_{z'p}^- f_p, \quad z, z' \in \{1, \dots, Z\}. \end{aligned} \quad (5.14)$$

with $h_p(\mathbf{f}^*)$ denoting the absolute equilibrium path cost, while $h_p(\mathbf{f}^*) + \theta^{-1} \ln f_p^*$ denotes the perceived equilibrium path cost consistent with MNL. The set of feasible non-negative path flow vectors \mathbf{f} is given by Ω , where a path flow vector is considered feasible when it satisfies the demand given by \mathbf{D} .

5.3.1 Accepted turn and link flow

Path flows \mathbf{f} are loaded onto the network, which, depending on the adopted method, results in accepted link and/or turn flows. It is important to realise that accepted flows can differ from desired flows. This difference can for example be caused by insufficient available supply in the form of capacity constraints, reducing the desired flows and resulting in smaller accepted flows. In case the desired flows are reduced this is reflected in the formation of queues, see Section 5.5, ensuring that the conservation of vehicles is not violated. The Accepted turn flows, denoted $\mathbf{Q} \in \mathbb{R}_+^{A \times A}$, are based on accepted path turn flows, $\mathbf{Q}^p \in \mathbb{R}_+^{A \times A}$ such that:

$$\mathbf{Q} = \sum_{p=1}^P \mathbf{Q}^p. \quad (5.15)$$

The accepted turn path flows only contain non-zero entries when it concerns turns that are traversed by path p . the Link level flow rates, denoted $\mathbf{q} \in \mathbb{R}_+^{A \times 1}$, can then also be obtained via:

$$q_a = \begin{cases} \sum_{p=1}^P f_p, & \text{if } a \text{ is the initial link on path } p, \\ \mathbf{1}^T \mathbf{Q}_{\bullet a}, & \text{otherwise.} \end{cases} \quad (5.16)$$

Observe that q_a is simply the sum of accepted turn flow rates towards link a , found by multiplying column vector $\mathbf{Q}_{\cdot a}$ (consisting of values in row a in matrix \mathbf{Q}) with transposed all-ones vector $\mathbf{1}^T$. The dimensionality of $\mathbf{1}$ is assumed to be implicitly determined by its context. Initial path links receive the sum of the desired path flows since there are no inflow restrictions and spillback is not considered. A simple illustrative example for a merge node with some reduction of accepted flow on turn (a_3, a_4) , of this path and turn based formulation, is provided in Figure 5.2. Let us now discuss how to construct \mathbf{Q}^p , depending on which assignment method is chosen.

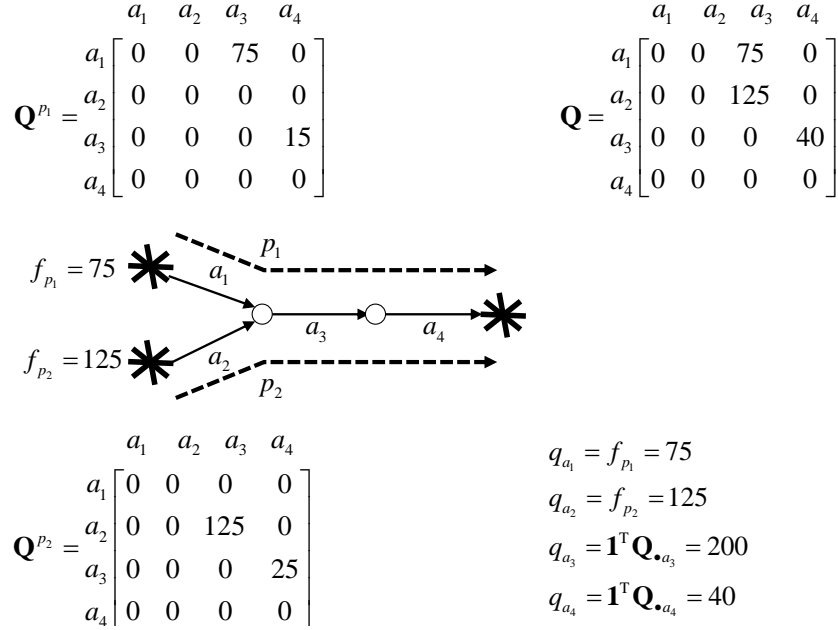


Figure 5.2: Example of accepted turn and link flows based on some given \mathbf{Q}^p .

5.4 Traditional capacity restrained static assignment

In traditional capacity restrained static assignment, there exists no distinction between desired path flows and accepted path flows, because flow is not restricted to the link capacity. As a result it generally holds that:

$$\mathbf{Q}^p = f_p \mathbf{M}^p. \quad (5.17)$$

We assume that in this assignment model path costs are link additive. In addition, a strictly concave link performance function is chosen. This is in line with how this model is typically used in practice. There exist many link performance functions. Here, we choose the BPR function (Bureau of Public Roads, 1964) given by:

$$c_a = \frac{\ell_a}{g_a^{\max}} \left(1 + \xi_a \left(\frac{q_a}{q_a^{\max}} \right)^{\zeta_a} \right), \quad (5.18)$$

resulting in a vector of link costs $\mathbf{c} \in \mathbb{R}_+^{A \times 1}$, where ξ_a, ζ_a , are positive parameters that require estimation, preferably based on empirical data. Path costs are then found via:

$$\mathbf{h} = \mathbf{P}\mathbf{c}. \quad (5.19)$$

Following the functional travel time decomposition as discussed in the previous chapter, we can decompose Equation (5.19) into a minimum link travel time component and a delay component via:

$$c_a^{\min} = \frac{\ell_a}{g_a^{\max}}, \quad (5.20)$$

$$c_a^{\text{delay}} = \frac{\xi_a \ell_a}{g_a^{\max}} \left(\frac{q_a}{q_a^{\max}} \right)^{\zeta_a}, \quad (5.21)$$

such that $c_a = c_a^{\min} + c_a^{\text{delay}}$, which translates to vectors $\mathbf{c}^{\min}, \mathbf{c}^{\text{delay}} \in \mathbb{R}^{A \times 1}$, respectively. Because path costs are link additive in this model, we can construct the minimum path travel times and the additional path delay travel times simply via:

$$\mathbf{h}^{\min} = \mathbf{P}\mathbf{c}^{\min}, \quad (5.22)$$

$$\mathbf{h}^{\text{delay}} = \mathbf{P}\mathbf{c}^{\text{delay}}, \quad (5.23)$$

respectively, where we recall that the total path travel times are the sum of the two components, i.e. $\mathbf{h} = \mathbf{h}^{\min} + \mathbf{h}^{\text{delay}}$. Since a link performance function only captures situations where the density increases with increasing flow, it necessarily only captures the hypocritical branch of the fundamental diagram, see also Figure 5.3 in the next section. Hence, Equation (5.21) cannot be decomposed any further into a hypocritical and hypercritical component, since the latter is non-existent (or zero, depending on one's perspective).

5.5 Capacity constrained model with point queues

As discussed in Chapter 3, traditional capacity restrained models make strong assumptions in favour of attractive mathematical properties resulting in: (i) congestion being modelled inside bottlenecks rather than in front of them, (ii) flows can exceed capacities resulting in implicit queues, rather than actual physical queues, (iii) path travel times are obtained through link performance functions which are known to be inconsistent with congested traffic conditions and require (additional) calibration. With this in mind, and the fact that the application context considered here requires the model to produce as accurate travel times as possible within the static modelling paradigm, more than warrants the consideration of an alternative assignment model. We therefore consider the recent residual point queue model of Bliemer et al. (2014) for the following reasons. First, unlike most static models, it includes a first order node model compliant with the requirements formulated in Tampère et al. (2011). This allows the model to formulate a path travel time function that incorporates hypercritical delays resulting from explicit residual queues imposed by the node model rather than implicit queuing delays through link performance functions. Second, all residual point queues are placed in front of the bottleneck instead of in the bottleneck. This is important, because if delay is modelled inside the bottleneck, delay is imposed in the wrong location and only paths traversing the bottleneck

experience the delay. However, in reality, other paths also might experience delay imposed by this bottleneck without actually traversing it. For example, paths that do not enter the bottleneck itself, but enter an alternative exit link of the node which is not the bottleneck link. By placing the queue in front of the bottleneck, delay is also affecting these paths, as one would expect. Observe that this is accomplished without considering spillback effects. Lastly, as shown in Bliemer and Raadsen (2017), this model can be directly derived from the dynamic first order network loading model originally proposed by Gentile (2010), which makes it, arguably, more realistic than its traditional static counterparts. Comparing this model with traditional static assignment shows that the obtained path travel times under a congested regime are found to be closer to what one would expect to see in reality, especially under varying demand scenarios, as is demonstrated in Section 5.5.3.

This capacity constrained model is compatible with any fundamental diagram with a concave uncongested branch. Since we aim to demonstrate that this model is preferable over the traditional capacity restrained approach using BPR, we, for now, adopt a fundamental diagram with an uncongested branch identical to the BPR function, such that the hypocritical delay of the uncongested branch, becomes a capacity constrained version of Equation (5.23):

$$c_a^{\text{hypo}} = \frac{\xi_a \ell_a}{g_a^{\text{max}}} \left(\frac{q_a}{q_a^{\text{max}}} \right)^{\zeta_a}, \quad q_a \in [0, q_a^{\text{max}}], \quad (5.24)$$

where $\mathbf{h}^{\text{hypo}} = \mathbf{P}\mathbf{c}^{\text{hypo}}$. The difference between the two models, on the link level, can be illustrated graphically through the fundamental diagram, Figure 5.3(a) depicts a BPR function, converted to its fundamental diagram form in the flow-density plane, while Figure 5.3(b) shows the fundamental diagram consistent with the model in Bliemer et al. (2014).

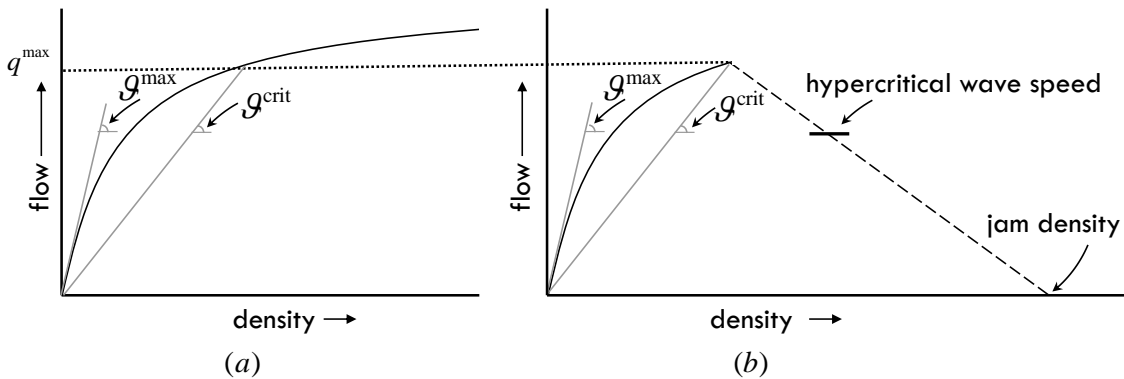


Figure 5.3: Schematic fundamental diagram comparison between (a) BPR function and Bliemer et al. (2014).

Observe that the BPR function is not consistent with the available link capacity whereas the capacity constrained model does limit the flow to capacity. Although the model in Bliemer et al. (2014) is capacity constrained, it is not storage constrained. If it would be, queues would propagate backwards with a non-zero hypercritical wave speed equal to the tangent of the hypercritical fundamental diagram branch. In Bliemer et al. (2014) the hypercritical wave speed is instead assumed to be zero across all hypercritical flow states, making sure queues do not spillback, hereby not considering storage constraints. Hence, our depiction of the dashed hypercritical branch with an assumed horizontal tangent. To construct the hypercritical delay

$\mathbf{h}^{\text{hyper}}$ - which in a capacity constrained context can be non-zero - we first discuss how queues form in this model, namely, via its node model.

5.5.1 Node model inputs and outputs

The accepted turn flows \mathbf{Q} , in Bliemer et al. (2014), are the results of desired path flows \mathbf{f} and the restrictions imposed by the node model. These node model supply restrictions are not only based on the given, and fixed, available exit link capacities governed by the fundamental diagram. They also depend on the offered incoming flow rates, also known as *sending flows* or desired outflows. In general networks, sending flows on the downstream end of a link are in fact the accepted flows entering the link at the upstream end. Hence, node model outcomes depend on sending flow rates, while sending flow rates depend on the (preceding) node model outcomes. To be able to find a solution, Bliemer et al. (2014) cast the network loading problem as a fixed point problem where the node model is assumed to be any first order node model compliant with the conditions outlined in Tampère et al. (2011). We denote this general node model, for each node n , via implicit function $\Gamma^n(\cdot)$. This model has two inputs; *sending flow rates* and *receiving flow rates*. In both cases we construct these flow rates on the network level and leave it to the node model function $\Gamma^n(\cdot)$ to extract the relevant information pertaining to node n . The sending turn flow rates are denoted $\mathcal{S} \in \mathbb{R}_+^{A \times A}$, which we obtain through:

$$\mathcal{S}_{aa'} = \begin{cases} \sum_{p=1}^P f_p M_{aa'}^p, & \text{if } a \text{ is the first link on path } p, \\ \sum_{a''=1}^A \sum_{p=1}^P M_{a''a}^p M_{aa'}^p Q_{a''a}^p, & \text{otherwise,} \end{cases} \quad (5.25)$$

where, in the initial case, the initial turn flow rate from connector link a to link a' comprises the sum of all path travel demands utilising that particular turn. The second case simply collects all accepted path turn flows $Q_{a''a}^p$ into link a and following path p , conditional on that path utilising the turn (a, a') under consideration. In other words, the sending flow on turn (a, a') is all the flow that is/can be offered to this turn after taking the upstream encountered capacity restrictions (if any), affecting the desired path flow f_p into account via \mathbf{Q}^p .

To clearly illustrate the distinction between *accepted* and *sending* flow rates, we revisit the example of Figure 5.2, only now explicitly showing the sending flow rates in addition to the assumed accepted path flow rates \mathbf{Q}^p . Clearly, the sending flow rates (200) offered to turn (a_3, a_4) , were somehow reduced to the accepted flow rates (40), as the result of some node model restriction. Also observe that the sending flow is simply the sum of the accepted preceding turn flows into link a_3 , see Figure 5.4.

The second input of the node model consists of the node's available supply, also known as *receiving flow*. Because spillback is not considered and neither are traffic signals (which could potentially impose turn dependent capacities), the available supply always equates to the outgoing link's capacity. As such it is denoted by $\mathcal{R} \in \mathbb{R}_+^{A \times 1}$ and defined via:

$$\mathcal{R} = \mathbf{q}^{\max} \quad (5.26)$$

Let us illustrate how this is applied on a corridor network as depicted in Figure 5.5(a). The network has a single path with a travel demand of $f_1 = 3000$ (veh/h), for a period of one hour. The sending flows, accepted flows, and flow acceptance factors are shown in Figure 5.5(b). Applying the node model on the turn from link 3 to link 4 yields a link flow acceptance factor of $\alpha_3 = 2000/3000 = 2/3$. This is because the sending flow from link 3 to link 4 exceeds the available outgoing supply and is scaled back accordingly. The residual point queue on link 3 can be made explicit via $q_3(1 - \alpha_3) = 1000$. The flow on subsequent link 4 is therefore reduced to 2000, in line with the available capacity. Note that the supply on link 5 also does not suffice given the incoming flow, leading to $\alpha_4 = 1000/2000 = 1/2$, reducing the flow even further. The remaining links have enough supply to accommodate the remaining flow. For these links, the acceptance factors are equal to 1.

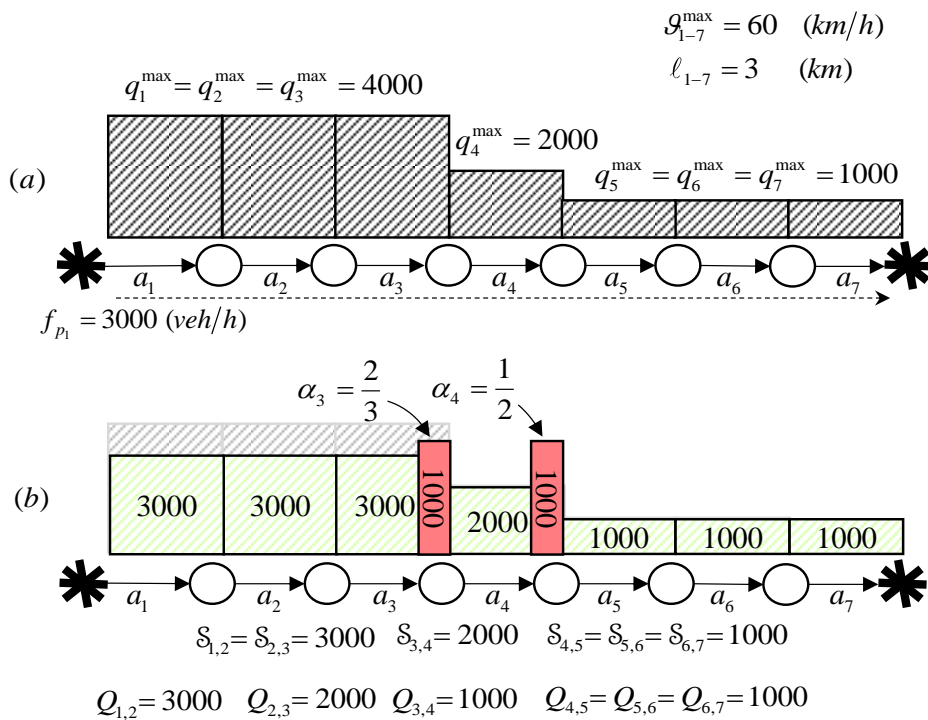


Figure 5.5: Example corridor network denoting (a) the infrastructure and (b) results for the residual queuing model.

For examples on how to determine the accepted flow rates for more complex nodes, we refer the reader to Tampère et al. (2011). As Figure 5.5 highlights, the accepted turn path flows in this model are no longer by definition equal to the desired path flows, as is the case in capacity restrained models. Intuitively, we can see that the accepted flow on a turn is given by the sending flow multiplied by the acceptance factor such that $Q_{aa'} = \alpha_a S_{aa'}$ for all turns (a, a') . Therefore, the capacity constrained formulation of \mathbf{Q}^p cannot be obtained as easily as defined in Equation (5.17), which does not consider any capacity restrictions. Instead, the path based accepted turn flows are the result of all the upstream encountered acceptance factors from the moment the path departed from its origin (with its original travel demand f_p). Hence, for the residual queuing model, accepted flows are found via:

$$Q_{aa'}^p = \begin{cases} f_p \prod_{a''=1}^A (\alpha_{a''})^{\eta_{a''}^{a'p}}, & \text{if } M_{aa'}^p = 1, \\ 0, & \text{otherwise,} \end{cases} \quad (5.29)$$

where indicator vector $\boldsymbol{\eta}^{ap} \in \mathbb{F}_2^{A \times 1}$ identifies the links that path p encountered up to, but not including link a . In the first case, we utilise $\boldsymbol{\eta}^{ap}$ to selectively apply the acceptance factors such that we only utilise them when it relates to a link already traversed by path p . If this is not the case, the acceptance factor is ignored because it reverts to one. As a result the original travel demand f_p is multiplied with all relevant acceptance factor up to the turn under consideration. Further, if turn (a, a') is not part of path p in the first place, it, by definition, carries no demand and we revert to zero flow via the second case.

5.5.2 Path travel time formulation

In the residual point queue model, the cost function as a whole and the hypercritical path delay in particular are not link additive. It is dependent on earlier encountered residual queues on the path. The delay corresponding with these queues represent the model's hypercritical delay and is obtained via:

$$h_p^{\text{hyper}} = \frac{t^{\text{end}}}{2} \left(\left(\prod_{a''=1}^A (\alpha_{a''})^{\eta_{a''}^{a'p}} \right)^{-1} - 1 \right), \quad (5.30)$$

where t^{end} is the simulation end time (h) and a' is the last link on path p . A derivation of this function, albeit in a set based formulation, can be found in Bliemer et al. (2014) or Bliemer and Raadsen (2017). Intuitively, this function can be interpreted as follows. The reciprocal of the multiplication of encountered acceptance factors can be considered as a special path demand to path capacity ratio which reverts to exactly 1 in case no excess demand exists on the path. In the corridor example discussed earlier (which has excess demand) this delay yields $(\frac{2}{3} \cdot \frac{1}{2})^{-1} = 3$. Since the available capacity is of course utilised as well, the path's residual queue to capacity ratio is obtained by decrementing the found value by one, i.e $3-1=2$. In the example this amounts to a 1000 (veh/h) supply versus 3000 (veh/h) demand, hence a 2000 (veh/h) queue, leading to a queue to capacity ratio of 2. Observe that this ratio equates to the delay experienced by the "final" vehicle on the path assuming the simulation runs exactly one hour. However, we are only interested in the average path delay over the simulation period considered. To obtain this average delay, we take the average of the delay of the first vehicle – no delay- and the last vehicle – final delay. The last vehicle's delay is obtained by taking the actual simulation duration into account, which leads to the hourly delay being multiplied by a factor $(t^{\text{end}} + 0)/2$. Figure 5.6 gives an impression of how various path queue to path capacity ratios across different simulation end times result in changing hypercritical delays in this particular model. Note that the resulting surface is linear in both directions.

Recall that unlike the hypercritical delay, hypocritical delay and minimum travel time remain link additive and identical to the capacity restrained model as long as the flow does not reach capacity (Equations (5.22) and (5.24)). When the flow exceeds capacity, the capacity constrained model yields a non-zero hypercritical delay as defined above, while the capacity restrained model allows an ever increasing (hypocritical) delay when flows exceed capacity.

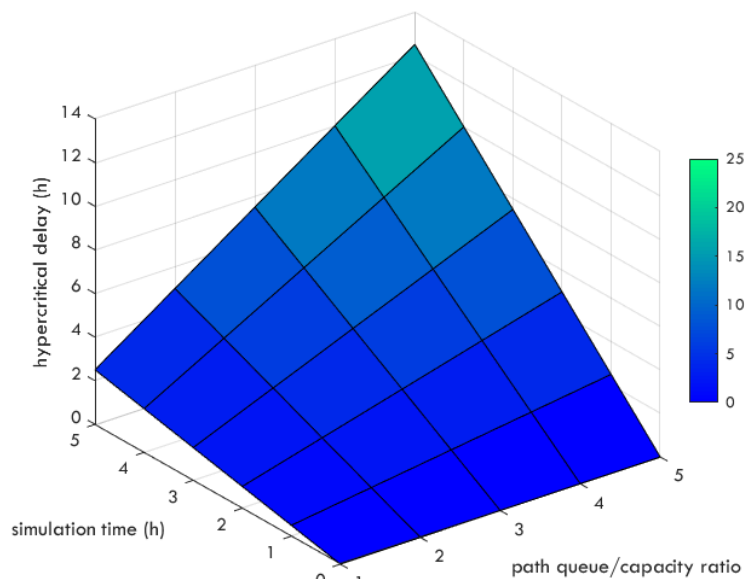


Figure 5.6: Hypercritical average path delay as a function of residual queues and simulation time.

5.5.3 Path travel time under varying demand

The two discussed assignment methods are both compatible with the proposed decomposition method; they are compliant with Equation (5.1) and do not consider spillback. However, we argue that the capacity constrained residual point queue model is the better choice. We illustrate this by returning to the simple corridor example introduced in the previous section and investigate results for both models under varying demand scenarios. We consider two demand scenarios, $D^1 = 1500$, $D^2 = 3000$. Let us consider the situation on link 4 in the first scenario. In reality, the demand of 1500 vehicles results in the last vehicle spending an additional half hour in the queue, before it can exit the link ($(1500 - 1000)/1000 = \frac{1}{2}$), while the first vehicle experiences no queuing delay at all, so the average delay, under uniform arrival rates, is expected to be $\frac{1}{4}$ (h). Similarly, we expect an average queueing delay of 1 hour for D^2 . Let us now consider the capacity restrained model where we must first calibrate the BPR function parameters (ξ, ζ) which we choose to do based on our most congested link. We find values of 0.9 and 4.1 respectively⁷. These values are then applied to all links in the network given their similar characteristics. Results of both traditional static assignment, with the calibrated BPR, and the residual queuing model are provided in Table 5.1. We would like to mention that this particular example is an extreme case, constructed to highlight issues when adopting the BPR function, so while representative, in general networks, these differences are typically somewhat less pronounced.

We find that due to the lack of hard capacity constraints, the traditional static capacity restrained model overestimates the true travel times on the path level, despite our link level calibration efforts. This is mainly because links 6 and 7 do not experience any congestion in reality, something that cannot be reflected in a capacity restrained setting. The residual queuing model on the other hand computes the hypercritical delay correctly based on the expected

⁷ We calibrated the parameters to link 5 in order to match the expected real delay in the first demand scenario. Clearly, there are other ways to do this, but regardless of how one calibrates, the results would reflect the same issues when adopting traditional static assignment under varying demand scenarios.

average delay due to capacity constraints. Since its hypercritical delay is not link additive, we only show the final delay on the path for each scenario.

Table 5.1: Path and link travel time comparison under varying demand.

| | | Travel time [h] per link | | | | | | | :Path |
|--------------|---|--------------------------|-------------------|-------------------|-------|------|------|------|-------|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| c^{\min} | Eqn. (5.20) | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.35 |
| $D^1 = 1500$ | BPR c^{delay} Eqn.(5.21) | $9 \cdot 10^{-4}$ | $9 \cdot 10^{-4}$ | $9 \cdot 10^{-4}$ | 0.015 | 0.25 | 0.25 | 0.25 | 0.77 |
| | Res. queue c^{hypo} Eqn.(5.24) | $9 \cdot 10^{-4}$ | $9 \cdot 10^{-4}$ | $9 \cdot 10^{-4}$ | 0.015 | 0.05 | 0.05 | 0.05 | 0.16 |
| | Res. queue h_1^{hyper} Eqn. (5.30) | - | - | - | 0.25 | - | - | - | 0.25 |
| | Flow acceptance factor α_a | 1 | 1 | 1 | 2/3 | 1 | 1 | 1 | |
| $D^2 = 3000$ | BPR c^{delay} Eqn. (5.21) | 0.015 | 0.015 | 0.015 | 0.25 | 4.29 | 4.29 | 4.29 | 13.17 |
| | Res. queue c^{hypo} Eqn. (5.24) | 0.015 | 0.015 | 0.015 | 0.05 | 0.05 | 0.05 | 0.05 | 0.24 |
| | Res. queue h_1^{hyper} Eqn. (5.30) | - | - | 1.00 | - | - | - | - | 1.00 |
| | Flow acceptance factor α_a | 1 | 1 | 2/3 | 1/2 | 1 | 1 | 1 | |

One could of course argue that we could have calibrated the BPR function against path travel times instead of link travel times, but in such a case the link travel times would have been underestimated, also this is problematic if links do not have the same characteristics. Further, once demand starts to vary, also a path based calibration would yield poor estimates across demand scenarios due to the model simply being inconsistent with how traffic behaves in an oversaturated situation. Either way, problems arise and since the calibration parameters in the BPR function only can approximate hypercritical delay and incorporate this result in the hypocritical uncongested branch rather than properly model the fundamental diagram's hypercritical branch, we find that the travel time is compromised in any case.

5.5.4 The impact of the location of congestion

Let us now consider the impact of congestion, through another example, depicted in Figure 5.7(a), it is identical to the earlier example in Figure 5.5(a), except for additional path p_2 that diverges upstream of link 4. A (virtual) traveller on this additional path would, in reality, also experience queuing delay in case the demand on the original path f_{p_1} exceeds 2000 (veh/h). This is due to the fact that they would be hindered by vehicles following path p_1 trying to enter link 4. Only models that take capacity constraints into account by modelling queuing delays in front of the bottleneck, are able to capture this effect. Here, the residual queuing model, see Figure 5.7(b), would yield an average queuing delay of 0.25 (h) for path p_2 , while traditional static assignment would not impose any hypercritical delay on this path.

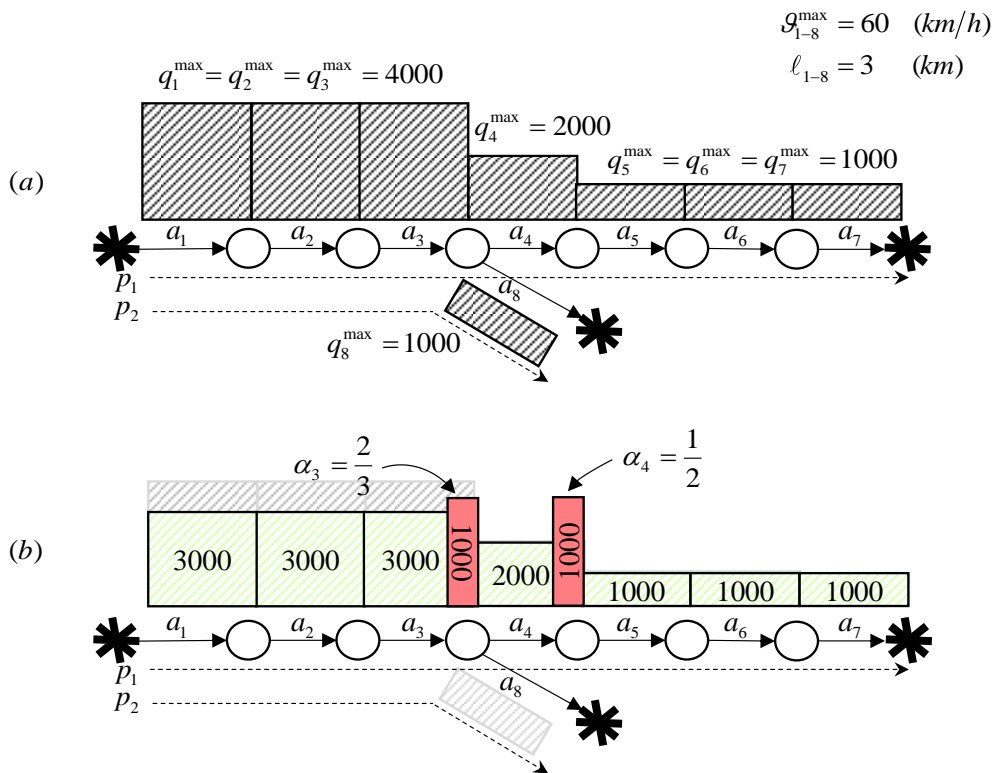
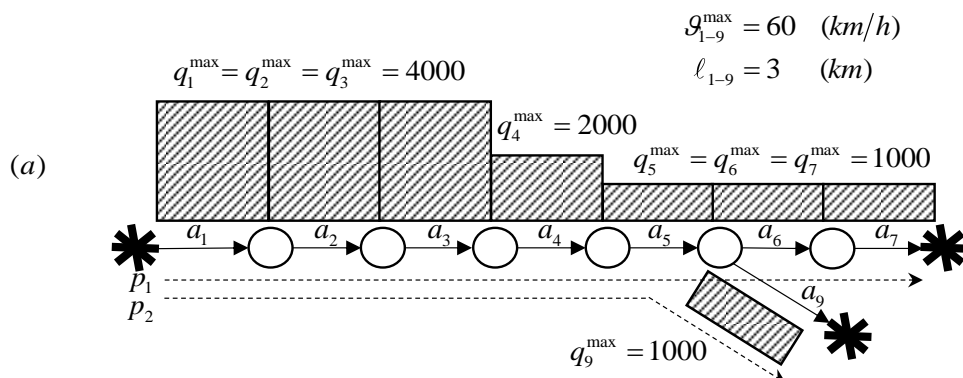


Figure 5.7: Corridor example with one additional path diverging (a) before initial bottleneck and its (b) residual queue result.

The inverse situation occurs when we assume $f_{p_1} = f_{p_2} = 1500$, but move the bottleneck link in front of the diverge node as depicted in Figure 5.8(a). Then, both paths traverse the exact same bottleneck links (4 and 5), which should result in equal hypercritical delay. Yet, in traditional static assignment, recall Table 5.1, the accrued delay on path 1 yields $h_1^{\text{delay}} = 3 \cdot 0.015 + 0.25 + 4.29 + 2 \cdot 0.25 = 5.09$ (h) while for the second path we find $h_2^{\text{delay}} = h_1^{\text{delay}} - c_6^{\text{delay}} - c_7^{\text{delay}} + c_9^{\text{delay}} = 4.84$ (h). The reason for this difference is; path 1 includes one more link that is falsely assessed as being oversaturated. The residual queuing model on the other hand computes the delay correctly and finds $h_1^{\text{delay}} = h_2^{\text{delay}} = 1$ (h), see Figure 5.8(c).



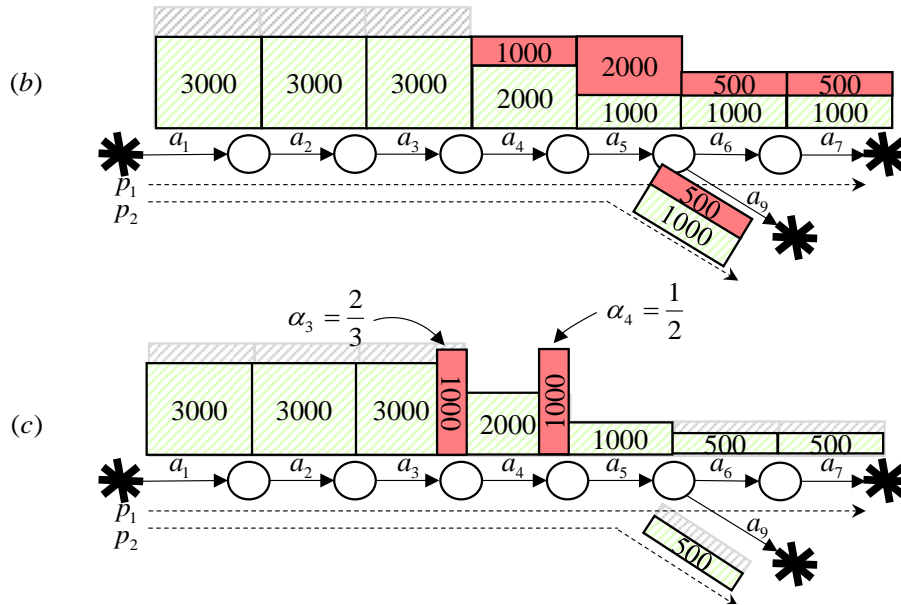


Figure 5.8: (a) Corridor example with one additional path diverging after last bottleneck, (b) traditional static assignment link level flows, (c) residual queuing turn level flows and residual queues.

5.6 Minimum travel time and delay subnetworks

As discussed in Section 5.1 the original network loading procedure $\Phi(\cdot)$ can be functionally decomposed into, $\Phi^{\min}(\mathbf{A}, \mathbf{P})$, and $\Phi^{\text{delay}}(\mathbf{f} | \mathbf{A}^{\text{delay}}, \mathbf{P}^{\text{delay}}, \mathbf{D})$, respectively. Here, we now formalise the methodology to construct each of the *altered* model components involved; $\Phi^{\min}(\cdot)$, $\Phi^{\text{delay}}(\cdot)$, $\mathbf{A}^{\text{delay}}$, and most importantly, $\mathbf{P}^{\text{delay}}$. In addition, we argue why the proposed method is most effective when hypocritical delay is not considered in the underlying assignment model such that the path delay equates to the hypercritical delay.

5.6.1 Network loading and travel time: minimum travel time subnetwork

The minimum travel time subnetwork loading procedure $\Phi^{\min}(\cdot)$ is identical to the original network loading procedure $\Phi(\cdot)$, albeit that it is flow invariant. The only real difference is to be found in the construction of the minimum path travel times \mathbf{h}^{\min} that only require the basic link characteristics of the traversed paths, see Equation (5.22). Since the paths remain the same, the original network and path set are adopted as well. It is however important to note that this minimum travel time vector only needs to be computed once, can be stored, and is reusable across any demand scenario to supplement the scenario specific delay. In case of our corridor example in Figure 5.5(a), we would compute and retain the total minimum travel time of $h_1^{\min} = 0.35$, see Table 5.1. This of course only remains a valid approach as long as both the network and path set remain fixed.

5.6.2 Lossless versus lossy decomposition

To reconstruct the total path travel times, any additional path delay that is incurred needs to be identified and added to the minimum path travel times. Since delay varies with flow it requires equilibration for each demand scenario considered. To maximise computational gains, we aim to construct a single delay subnetwork to capture the delay component of the path travel times.

As discussed, a functional decomposition of delay includes both hypocritical and hypercritical delay. We first discuss the impact of including hypocritical delay in this decomposition method.

So far, the capacity restrained static traffic assignment procedure as well as the residual queuing model adopted an uncongested branch of the fundamental diagram that yields positive hypocritical delay whenever a link has non-zero flow. Hence, all links in the original network that carry positive demand in any of the demand scenarios contribute to one or more path's delay. Therefore, the original path travel time delays can only be reconstructed in a delay subnetwork when the entire (used) original network is retained. Clearly, this can never amount to any computational gain, because the network cannot be simplified without losing information in the form of missed delay.

There are a number of possible approaches to get around this issue. First, one can accept the decomposition method becomes *lossy* rather than *lossless*. If a lossy approach is deemed acceptable, the next choice pertains to at what point a link can be removed and when to retain it. For traditional static assignment for example, we could impose a minimum volume-capacity (v/c) ratio threshold and only when this threshold is exceeded we retain the link, where retained links are referred to as *critical-delay links*, while all other links are classified as *non-critical-delay links*. As an example consider the corridor network in Figure 5.9, where we assumed a volume-capacity threshold of 1 as an example, leading to a delay subnetwork smaller than the original network, containing only links 4-7.

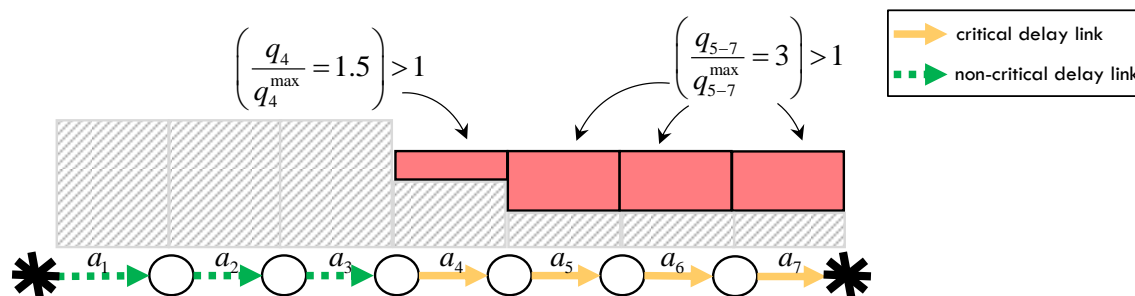


Figure 5.9: Lossy delay subnetwork under BPR function with minimum v/c ratio threshold of one.

Do note that the hypocritical delay of links 1-3 can no longer be captured when they are removed, because the delay is flow dependent. Hence, adopting traditional static assignment with a BPR like travel time function, by definition leads to information loss under this decomposition method. Similar information loss would occur when we adopt the residual queuing model with the fundamental diagram as originally proposed in Figure 5.3.

As an alternative to imposing a v/c ratio threshold, we could also change the assumptions on the underlying assignment model instead. When we for example adopt Newell's popular triangular fundamental diagram (Newell, 1993), the hypocritical delay vanishes because $g^{\max} = g^{\text{crit}}$, see Figure 5.10. This means the decomposition method would not suffer from any hypocritical delay related information loss with respect to the original travel times because there simply isn't any. It does mean the underlying assignment model is somewhat simplified. Given that it is well known that the hypocritical delay in reality is generally relatively small compared to both the minimum travel time and especially any hypercritical delay, such a

simplification is deemed preferable over the aforementioned threshold based approach. This method can potentially also be adopted for a traditional static assignment method, for example by linearizing the uncongested branch by fixing the speed to g^{crit} .

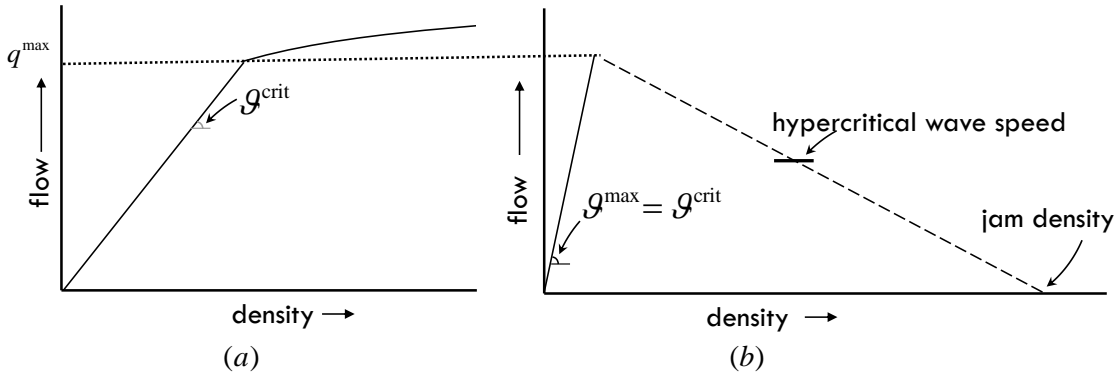


Figure 5.10: Fundamental diagram comparison with linear uncongested branches for (a) BPR like link performance function and (b) Newell's triangular fundamental diagram for Bliemer et al. (2014).

From here on we choose to adopt a linear uncongested branch to allow for our decomposition method to remain lossless. Hence, hypocritical delay is absent in the underlying assignment, i.e. $\mathbf{h}^{\text{hypo}} = 0$, and therefore $\mathbf{h}^{\text{delay}} = \mathbf{h}^{\text{hyper}}$.

5.6.3 Delay subnetwork: capacity constrained versus capacity restrained

Based on our findings in the previous section we know that our delay subnetwork only needs to reflect the functional hypercritical delay component of path travel time in order to be able to replicate the original path travel times. Recall that hypercritical delay follows from supply restricted infrastructure, i.e. bottlenecks. Since the number of bottlenecks in a general network only constitutes a relatively small part of the total infrastructure, this delay subnetwork is also small compared to the original network⁸. The delays obtained from the delay subnetwork only match the original delay perfectly when it contains the original bottlenecks without exception. If this is not the case, the method again becomes lossy rather than lossless. The delay subnetwork should therefore be constructed as the union over all identified bottlenecks across all considered demand scenarios. We first discuss how to obtain the delay subnetwork under a single demand scenario. This does not yet yield any computational gains, but demonstrates the concept. We postpone the discussion of identifying the delay subnetwork under varying demand scenarios to Section 5.8.

The contents of the delay subnetwork also depend on the adopted assignment method. We first consider the case of constructing the delay subnetwork in combination with the residual queuing model. In this model, turns affected by links with residual point queues need to be retained in order to reconstruct the original path delay. Under a single demand scenario, one can find these turns by equilibrating the original model and retaining the turns on links with a residual queue. Observe that, again for a single demand scenario, this by definition allows for a lossless, yet not computationally more attractive, approach. To illustrate this, we, again, consider the corridor network example of Figure 5.5 under $D = 3000$. The critical-delay links

⁸ We base this statement on our observations testing this method on real-world large scale applications, see the results presented in Chapter 6.

to retain are 3, 4 and 5, as depicted in Figure 5.11. Note that while link 5 does not have a residual queue on the link itself, it is still needed to reconstruct the delay implicitly defined through α_4 . This is because, in the residual queuing model, delay follows from the node model. The node model requires the inflow rates of link 4 as well as the available outgoing supply, i.e. receiving flow, of link 5 to determine the flow acceptance factor. Therefore, the node's topology needs to be retained in full whenever any of the incoming links hold a residual queue.

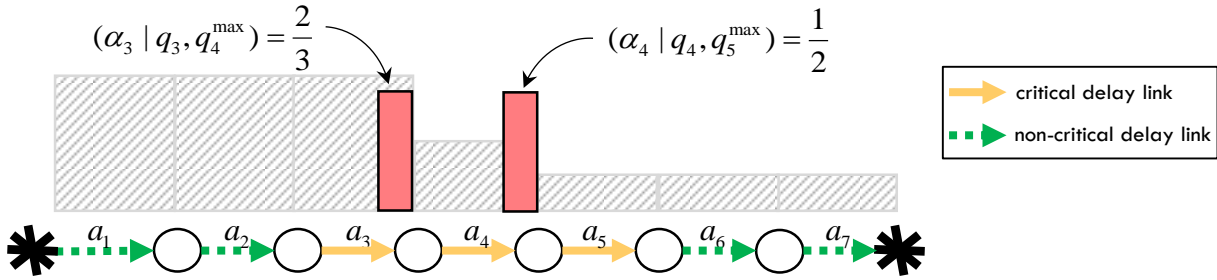


Figure 5.11: Delay subnetwork for residual queuing model on corridor network.

From Figure 5.11 we can also observe that bottlenecks serve as a filter for the propagation of excess flows in case of the residual queuing model, limiting the number of downstream links that are considered oversaturated. This in contrast to capacity restrained models. In capacity restrained models flows are not withheld and continue to (potentially impose hypercritical delays on downstream links. In our decomposition method this is especially undesirable since it results in more links to be included in the delay subnetwork reducing the potential for computational gains. In this particular example, we would find hypercritical delay on links 4-7 (not on link 3 since delay is wrongly imposed within the bottleneck instead of in front of it).

Finally, since spillback is not considered in either model, queues/delays remains local and do not physically propagate upstream. The number of links exhibiting hypercritical delay, compared to models that do consider spillback, is therefore relatively small, reducing the size of the delay subnetwork that needs to be considered and increasing the potential computational gain.

5.6.4 Network loading and travel time: the delay subnetwork

We now formalise the, so far informally discussed, construction of the delay subnetwork for general networks. All links to be retained in the delay subnetwork are marked as such in link indicator vector $\beta \in \mathbb{F}_2^{A \times 1}$. The construction of this vector is conditional on the assignment method, but once this classification is in place, the remainder of the procedure is agnostic to the underlying model.

5.6.4.1 Critical-delay link classification

For traditional static capacity restrained assignment adopting a BPR like function with a linear uncongested branch we obtain β via:

$$\beta_a = \begin{cases} 1, & \text{if } \frac{q_a}{q_a^{\max}} \geq \beta^{\min}, \\ 0, & \text{otherwise,} \end{cases} \quad (5.31)$$

Where β^{\min} denotes the minimum v/c ratio threshold for a link to be considered delay critical. In the absence of modelling hypocritical delay, $\beta^{\min} > 1$ by definition. Clearly, one could also adopt the original BPR function and choose $\beta^{\min} < 1$ to retain more links, albeit at the cost of missing some of the original hypocritical delay. However, as mentioned before we focus on a lossless decomposition and as such do not explore this latter scenario any further.

For the residual queuing model (with a triangular fundamental diagram), the situation is slightly more complicated. Clearly, links with a flow acceptance factor $\alpha_a < 1$ should be included in the delay subnetwork. However, when this occurs, the node's supply restrictions, i.e. outgoing links, also need to be retained as discussed in the previous section. Hence, the critical-delay link classification for this particular assignment model is given by:

$$\beta_a = \begin{cases} A_{na}^- \parallel A_{n'a}^+, & \text{if } \exists(a', n): \alpha_{a'} A_{na'}^- < 1 \text{ or } \exists(a', n'): \alpha_{a'} A_{n'a'}^+ < 1, \\ 0, & \text{otherwise,} \end{cases} \quad (5.32)$$

where link a is retained whenever any of the links a' of its upstream or downstream node n, n' , respectively, holds a residual queue and exhibits hypercritical delay. So, either link a itself holds a residual queue, i.e. $a = a'$, or the link is needed to reconstruct the queue by providing the sending or receiving flows as input to the node model $\Gamma^n(\cdot)$.

5.6.4.2 Dual representation of networks

The formulation of our decomposition method benefits from adopting a dual representation of the network. Such a representation is not new, it has been used before in graph based networks, see for example Eichler et al. (2013), or Añez et al. (1996). In a dual representation, each link is modelled as a vertex, while the edges represent turns between links. An example of an original network representation and its dual representation is provided in Figure 5.12.

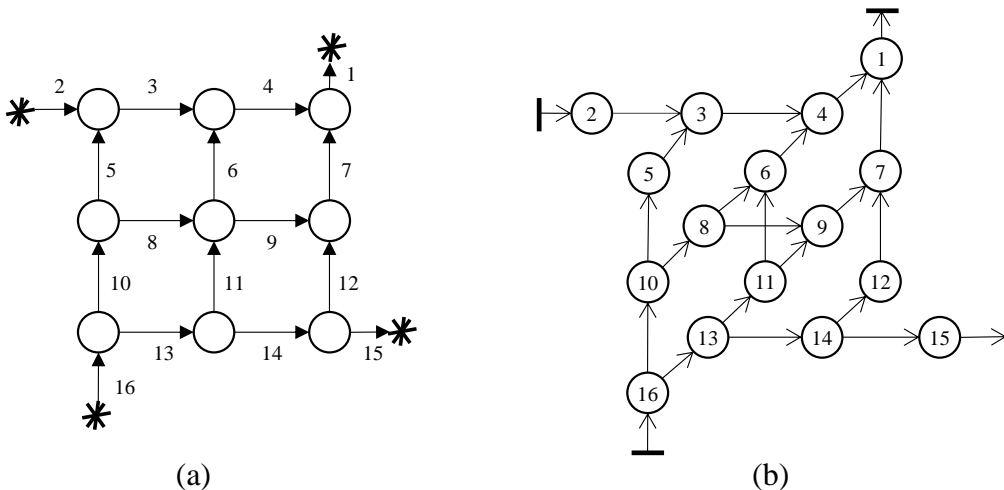


Figure 5.12: (a) Example Manhattan type network with four centroids. (b) Dual representation of the network shown in (a).

This dual representation should not be confused with a dual graph, which is something else entirely, this dual representation of the transport network instead equates to the medial graph in a graph theoretical sense. The extraction of the delay subnetwork - conditional on the link classification in the previous section - revolves around contracting relevant turns. In the

original network, a turn comprises two links, while in dual representation a turn is depicted by a single edge making it more suitable to illustrate the contraction process.

5.6.4.3 Path based turn contraction

Constructing the delay subnetwork based on link classification vector β is a two-step process. In the first step, path specific delay subnetworks, one per path $p \in \{1, \dots, P\}$, are constructed. Each of these (partial) delay subnetworks represents the infrastructure required to reproduce the hypercritical delay for a single path, termed the *critical-delay path*. A critical-delay path only contains critical-delay links, hence, constructing these paths requires merging subsequent path links on the path that are classified as non-critical. Since the minimum travel time of this link is already captured in the minimum travel time subnetwork, it is safe to do so, without incurring any information loss. By merging path links instead of removing them, path connectivity is maintained while still reducing the number of links per path. The second step involves the construction of the actual delay subnetwork based on the critical-delay paths.

Merging subsequent path links can, arguably, be formulated more elegantly from a turn based perspective. Merging occurs in the upstream direction; whenever a non-critical-delay link is encountered it is merged with its directly adjacent upstream link on the path's turn, where we recall that path turns are denoted via matrix \mathbf{M}^p originally defined in Equation (5.12). An example for a single merge is illustrated in Figure 5.13, utilising our dual representation introduced in the previous section.

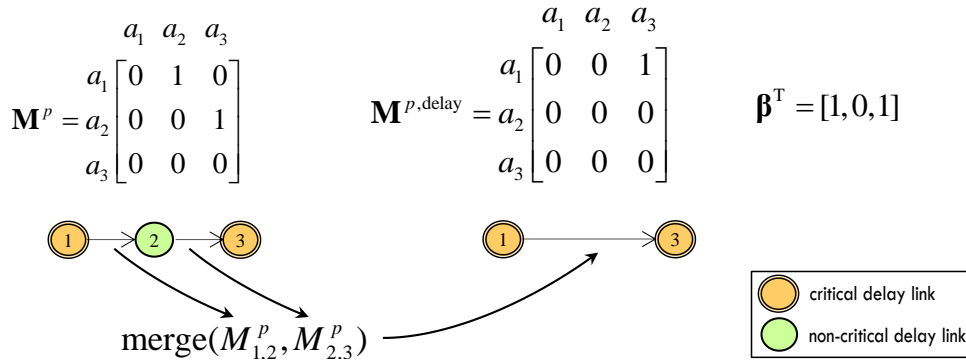


Figure 5.13: Schematic example of (upstream) merging of a single non-critical link in dual network representation.

The function $\text{merge}(\cdot)$ takes two turns, comprising three consecutive links, and connects the initial and final link while eliminating the intermediate (non-critical-delay) link. In case multiple consecutive non-critical-delay links are present, the procedure should be repeated recursively up to the point when either the next critical-delay link is encountered or the path terminates. While this is conceptually straightforward, formalising such recursiveness is often somewhat less intuitive. Yet, we still do so in order to provide the required rigour to our method. Also, it obviates the need to provide a separate algorithm since the formulation in Equation (5.33) already suffices.

$$\text{merge}(\mathbf{M}_{a^*}^p, \mathbf{M}_{a'^*}^p) = \begin{cases} \text{merge}(\mathbf{M}_{a^*}^p, \mathbf{M}_{a'^*}^p), & \text{if } \exists a'' : \beta_{a''} = 0 \text{ and } M_{a^* a''}^p = 1, \\ \mathbf{M}_{a'^*}^p, & \text{else if } \exists a'' : \beta_{a''} = 1 \text{ and } M_{a^* a''}^p = 1, \\ \mathbf{0}, & \text{otherwise,} \end{cases} \quad (5.33)$$

with $a'' \in \{1, \dots, A\}$. In this function, turns are merged recursively – via the first case – as long as the next turn’s downstream link remains non-critical. If this no longer holds, the function terminates in one of two possible ways: if it terminates via the second case, it means the path has not ended yet and the upstream link of the initial recursion is connected with the downstream critical-delay link encountered in the last recursion (by taking on its row values via $\mathbf{M}_{a''}^p$). Otherwise, there no longer exists a next turn, i.e. the path ends, and the initial upstream link becomes the last link on the path, i.e. setting the row to all zero vector $\mathbf{0}$.

To obtain the entire critical-delay path subnetwork $\mathbf{M}^{p,\text{delay}} \in \mathbb{F}_2^{A \times A}$, the merge procedure is selectively applied to eligible turns (a, a') of the original path \mathbf{M}^p via:

$$\mathbf{M}_{a\bullet}^{p,\text{delay}} = \begin{cases} \text{merge}(\mathbf{M}_{a\bullet}^p, \mathbf{M}_{a'\bullet}^p), & \text{if } \exists a' : \beta_a M_{aa'}^p = 1 \text{ and } \beta_{a'} = 0, \\ \mathbf{M}_{a\bullet}^p, & \text{else if } \exists a' : \beta_a M_{aa'}^p = 1 \text{ and } \beta_{a'} = 1, \\ \mathbf{0}, & \text{otherwise,} \end{cases} \quad (5.34)$$

with $a' \in \{1, \dots, A\}$. As can be seen a recursive merge is only invoked when the path transitions from a critical-delay link to a non-critical-delay link ($\exists a' : \beta_a M_{aa'}^p = 1$ and $\beta_{a'} = 0$). Alternatively, when a turn is found between two critical delay links ($\exists a' : \beta_a M_{aa'}^p = 1$ and $\beta_{a'} = 1$), the original turn is retained. Whenever a turn’s incoming link a is classified non-critical the turn can be removed, because it will be replaced by the result of the recursive merge. An example of constructing a critical-delay path, is depicted in Figure 5.14.

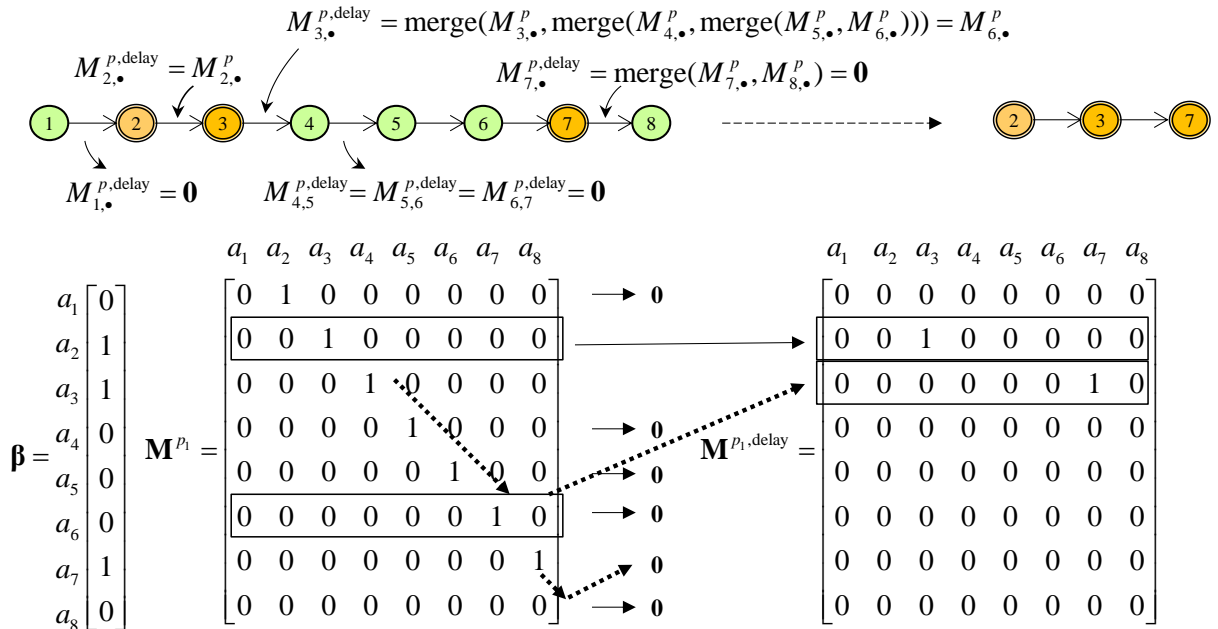


Figure 5.14: Example extraction of path delay subnetwork using dual representation (dashed lines between matrices indicate merge function is invoked).

5.6.4.4 Constructing the network wide delay subnetwork

The extension to the network level to obtain the overall delay subnetwork, denoted $\mathbf{M}^{\text{delay}} \in \mathbb{F}_2^{A \times A}$, is straightforward: a turn is included in the delay subnetwork if there exists a critical-delay path where the turn is included as well such that:

$$M_{aa'}^{\text{delay}} = M_{aa'}^{1,\text{delay}} \parallel M_{aa'}^{2,\text{delay}} \parallel \dots \parallel M_{aa'}^{P-1,\text{delay}} \parallel M_{aa'}^{P,\text{delay}}. \quad (5.35)$$

It is left to the reader to observe that the provided turn based formulation can be converted to the more common node and link based form yielding $\mathbf{A}^{\text{delay}}, \mathbf{P}^{\text{delay}}$, respectively. To illustrate the effect of the procedure on the network level, consider the network, in dual representation, as depicted in Figure 5.15(a). A hypothetical classification of the original links is readily provided, allowing the construction of the delay subnetwork, by applying Equations (5.35), as shown in Figure 5.15(b).

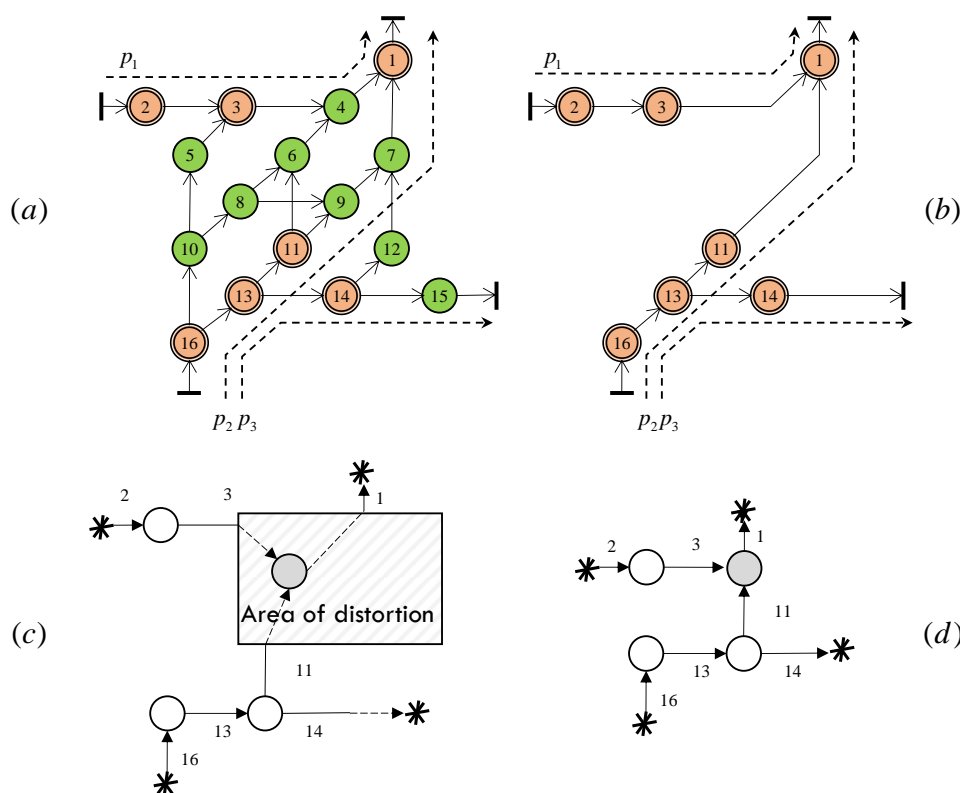


Figure 5.15: Dual representation; (a) link classification of original network, (b) extracted delay subnetwork. Original representation; (c) extracted delay subnetwork with area of distortion, (d) alternative representation.

We included Figure 5.15(c) to illustrate what the delay subnetwork in non-dual representation looks like. Observe how the node shown in grey is the result of the merging of turns (3,4), (4,1) and (11,9), (9,7), (7,1), respectively. Furthermore, by letting the retained links and nodes of the original network reside in their original locations, one or more areas of spatial distortion are created in the delay network. The dashed part of each link in this spatially distorted area has no length, nor travel time cost, because it solely exists to maintain connectivity. Alternatively, if one wishes to have a network with spatially correct link lengths, one could consider altering the positions of the originally retained network nodes as shown in Figure 5.15(d), although in general networks, this is usually not possible. This

approach is discouraged because it requires displacing nodes and links. From our experience, the link length distorted network is preferred, because it allows for a meaningful interpretation of bottleneck locations.

5.7 Critical-delay path consolidation

The proposed delay subnetwork method discussed in the previous section reduces the computational burden by limiting the number of links on each path. That said, the total number of paths remains unaffected. We propose to extend the method presented in the previous section by consolidating critical-delay path set $\mathbf{P}^{\text{delay}}$. This consolidation procedure exploits the fact that the likelihood of complete path overlap in the delay subnetwork has significantly increased compared to the original network. Replacing completely overlapping critical-delay paths with a single representative path allows for a significant reduction in the number of paths that need to be considered in network loading. It is demonstrated that the majority of the computational gain is the result of applying this consolidation method on top of the delay subnetwork extraction.

Let us illustrate the concept of path consolidation with the example presented in Figure 5.16. Initially, none of the four original paths overlap completely, as can be seen in Figure 5.16(a). Figure 5.16 (b) shows the assumed link classification such that we obtain the delay subnetwork, in Figure 5.16(c).

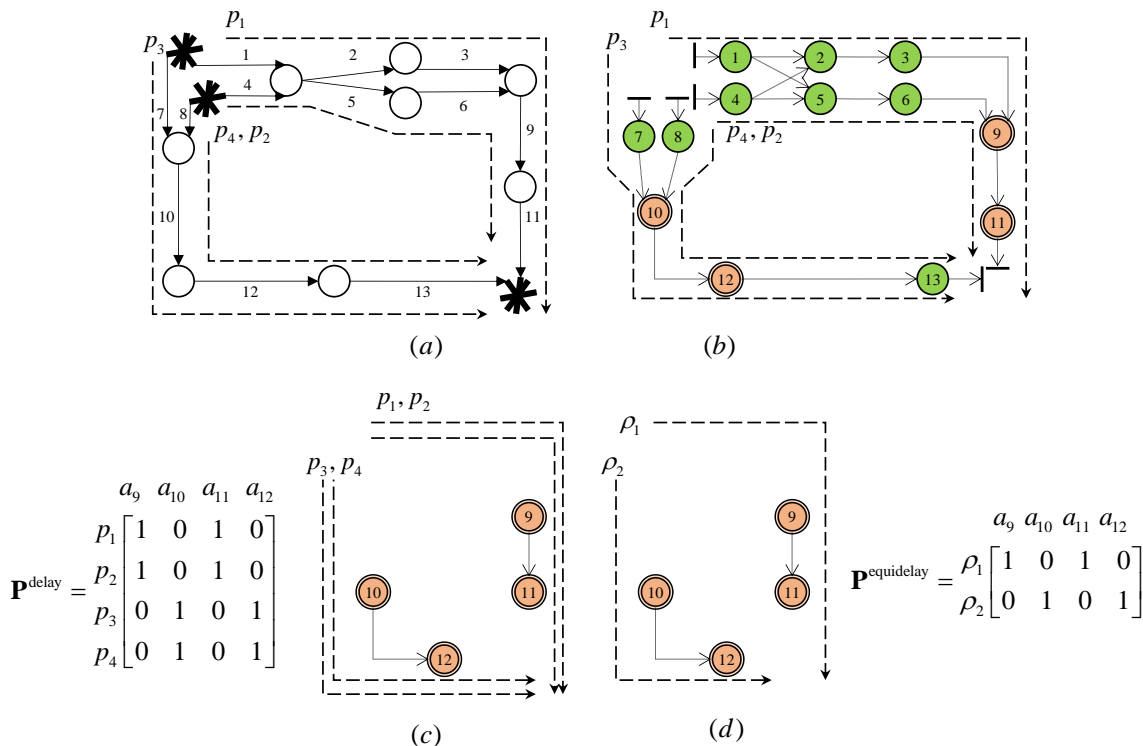


Figure 5.16: Original network, with four paths, (b) dual representation of the same network, including link classification, (c) delay subnetwork induced from critical-delay paths, (d) path consolidation based on equidelay paths.

In the delay subnetwork, the first two paths and last two paths now overlap fully and can therefore be consolidated, reducing the number of paths. The two resulting paths are termed

equidelay paths because they represent one or more critical-delay paths - traversing the same turns in the delay subnetwork - having equal delay. Equidelay paths are denoted by $\rho \in \{1, \dots, \mathcal{P}\}$, where by definition it holds that $\mathcal{P} \leq \mathcal{P}$.

The equidelay path indicator matrix $\mathbf{P}^{\text{equidelay}} \in \mathbb{F}_2^{\mathcal{P} \times \mathcal{A}}$ can be constructed in different ways. We choose to define implicit function $\text{unique} : \mathbb{R}^{\mathbb{Z}_+ \times \mathbb{Z}_+} \rightarrow \mathbb{R}^{\mathbb{Z}_+ \times \mathbb{Z}_+}$ that removes duplicate rows in $\mathbf{P}^{\text{delay}}$ such that:

$$\mathbf{P}^{\text{equidelay}} = \text{unique}(\mathbf{P}^{\text{delay}}), \quad (5.36)$$

where the number of unique rows in $\mathbf{P}^{\text{delay}}$ becomes the number of equidelay paths \mathcal{P} . In $\mathbf{P}^{\text{equidelay}}$, the number of remaining paths that needs to be considered during network loading is reduced significantly. However, we can only use $\mathbf{P}^{\text{equidelay}}$ when we know the demand on each equidelay path ρ . Yet, this demand is conditional on the path choice that is being applied to the original paths. We also emphasize that fully overlapping critical-delay paths do not necessarily belong to the same origin-destination pair. Since only bottlenecks are retained, paths having different origins and/or destinations can just as well be represented by the same equidelay path, such as ρ_1 in our example. We therefore maintain a dedicated mapping between original (critical-delay) paths and their equidelay counterpart denoted by $\mathcal{O} \in \mathbb{F}_2^{\mathcal{P} \times \mathcal{P}}$. In this mapping $\mathcal{O}_{\rho p}$ takes on a value of 1 when *critical-delay* path p and *equidelay* path ρ overlap completely and otherwise it is 0. More formally:

$$\mathcal{O}_{\rho p} = \begin{cases} 1, & \text{if } \mathbf{P}_{\rho \bullet}^{\text{equidelay}} = \mathbf{P}_{p \bullet}^{\text{delay}}, \\ 0, & \text{otherwise.} \end{cases} \quad (5.37)$$

The equidelay path flows used in network loading can then be constructed via:

$$f_\rho = \sum_{p=1}^{\mathcal{P}} \mathcal{O}_{\rho p} f_p, \quad (5.38)$$

where we note that the original path flow f_p is the result of the path choice model discussed in Section 5.3, which in turn is conditional on all alternative original paths and their path costs for the relevant od-pair. The original path cost formulation however, no longer suffices either because h_p^{delay} is no longer available. Instead, we only have delay obtained from equidelay paths, i.e. h_ρ^{delay} . We therefore replace Equation (5.1) – on a per path basis - by the following:

$$h_p = h_p^{\min} + \underbrace{\sum_{\rho=1}^{\mathcal{P}} \mathcal{O}_{\rho p} h_\rho^{\text{delay}}}_{\Phi^{\text{equidelay}}}, \quad (5.39)$$

Φ^{\min}

where the minimum path travel time is obtained via the minimum travel time subnetwork, while the delay is collected from the delay subnetwork by mapping the equidelay path delay to its original path counterpart(s). The path choice procedure $\Psi^{\text{equidelay}}$ then, reflects the adoption of

Equation (5.39) to construct the path flows conditional on the costs obtained from Φ^{\min} , $\Phi^{\text{equidelay}}$, respectively, resulting in the decomposition as originally formulated in Equation (5.7), albeit for a single demand scenario only.

5.8 Varying demand scenarios and the super scenario

So far, we only considered a single demand scenario s . Let us now extend this to multiple demand scenarios. First, recall that the proposed decomposition method only remains lossless as long as all critical-delay links across all considered demand scenarios are included in the delay subnetwork. Which links are critical-delay links for a given demand scenario depends on that demand scenario's equilibrium solution. This poses a challenge, because the purpose of this method is to speed up the traffic assignment procedure and finding each scenario's equilibrium solution does not contribute in achieving this goal. To circumvent this issue, we aim to construct delay subnetwork $\mathbf{A}^{\text{delay}}$ by equilibrating a single demand *super-scenario* s^* . We then, in addition, apply our path consolidation method to identify equidelay paths $\mathbf{P}^{\text{equidelay}}$. The delay subnetwork $\mathbf{A}^{\text{delay}}$ and consolidated paths $\mathbf{P}^{\text{equidelay}}$ are then used to equilibrate each actual demand scenario s at a fraction of the time it would cost if we would have used the original network and paths.

For this approach to work, the initial paths \mathbf{P} - which are fixed across all demand scenarios - need to contain the relevant paths for all demand scenarios. We point out that this requires generating a somewhat broader path choice set compared to set that one would use for a single demand scenario. However, we expect it does not differ significantly given our considered application context; for both quick-scan applications and matrix calibration procedures many, but relatively small, variations in demand are explored. Also, stochastic path choice set generators typically already perturb expected link costs with a user-specified amount of variance to (partially) account for this type of uncertainty, see for example Fiorenzo-Catalano et al. (2004).

5.8.1 Alternative super-scenario approaches

The second challenge is creating the single demand matrix \mathbf{D}^{s^*} for super-scenario s^* such that it contains all bottlenecks across all scenarios s . While it is always possible to verify if any information loss occurred a-posteriori, we cannot guarantee a lossless result a-priori without performing costly assignment. Intuitively, one might assume that, in attempting to avoid additional assignments, we can capture all critical-delay links across demand scenarios by taking the maximum demand for each od-pair across the various demand scenarios via:

$$D_{zz'}^{\max} = \max_{s \in \{1, \dots, S\}} \{D_{zz'}^s\}, \quad (5.40)$$

where we then subsequently set $\mathbf{D}^{s^*} = \mathbf{D}^{\max}$. Even though this will likely be a good approach in many cases, it cannot guarantee that all critical-delay links are being captured. To demonstrate this, we provide an example of such a situation in the hypothetical network depicted in Figure 5.17(a). It contains 4 paths, p_1 to p_4 , of which p_1 and p_2 are alternatives for the same origin-destination-pair.

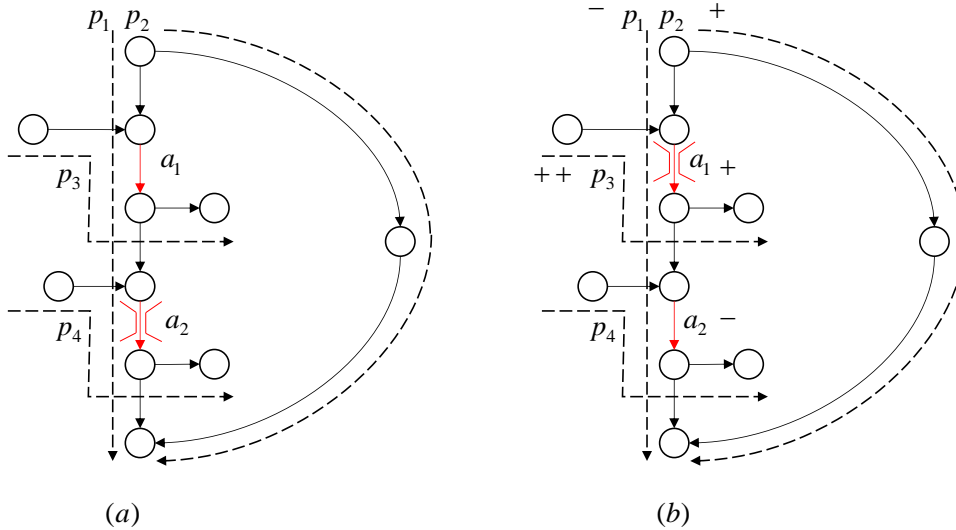


Figure 5.17: (a) Base network with a single bottleneck at link a_2 (b) same network under increased demand on path p_3 with a new bottleneck appearing on link a_1 and one bottleneck disappearing at link a_2 .

Let us consider two different demand scenarios; s_1 and s_2 . For initial demand scenario s_1 we assume that $h_1^s \ll h_2^s$ such that the majority of the demand is assigned to path p_1 under SUE conditions. Link a_1 is close to, but not exceeding, capacity while link a_2 only just exceeds capacity under s_1 , due to the combined path flows of p_1 and p_4 . Let us now consider demand scenario s_2 , which is identical to s_1 except for an increase in demand on the origin-destination pair represented by path p_3 . This leads to link a_1 becoming a bottleneck. Some of the flow on path p_1 will be diverted to p_2 because of its increased cost. As a result, the flow on link a_2 decreases, which no longer makes it a bottleneck, see Figure 5.17(b). This demonstrates that even with just two different demand scenarios applied on the same network, taking the maximum demand might not only yield additional bottlenecks, but can also remove bottlenecks. Hence, \mathbf{D}^{\max} does not guarantee that all critical-delay links are captured. The example in Figure 5.17 is specifically tailored to highlight this issue, it therefore remains to be seen how often bottlenecks actually disappear with increasing demand in a more practical setting. This is investigated further in Chapter 6.

Perhaps, adopting an average demand across all available demand scenarios s might be able to capture more of the bottlenecks than a maximum demand approach, especially if the number of bottlenecks that disappears with increasing demand exceeds the number of additional bottlenecks. Conversely, if more bottlenecks appear than disappear with increasing demand, adopting the maximum demand approach will yield the better result. By also exploring this alternative we ensure that either possibility is captured. The average demand approach is formulated as follows:

$$\bar{D}_{zz'} = \frac{1}{S} \sum_{s=1}^S D_{zz'}^s, \quad (5.41)$$

when adopting this approach we assume $\mathbf{D}^{s^*} = \bar{\mathbf{D}}$.

5.8.2 Incorporating flow margins in the super-scenario

Given that we cannot guarantee a lossless a-priori construction of the super-scenario, we also investigate alternative ways to reduce the likelihood of missing critical-delay links. One way to do this is to include more links in the delay subnetwork, by introducing a flow rate based margin as part of the original critical-delay links classifications in Equations (5.31) and (5.32), respectively. This margin can be defined either in absolute or relative terms. The link classification vector $\boldsymbol{\beta}^{|\Delta|} \in \mathbb{F}_2^{A \times 1}$ under an absolute flow margin, denoted $q^{|\Delta|}$, is then given by:

$$\beta_a^{|\Delta|} = \begin{cases} 1, & \text{if } \frac{q_a + q^{|\Delta|}}{q_a^{\max}} \geq \beta^{\min}, \\ 0, & \text{otherwise,} \end{cases} \quad (5.42)$$

when adopting the traditional static capacity restrained model, or:

$$\beta_a^{|\Delta|} = \begin{cases} A_{na}^- \parallel A_{n'a}^+, & \text{if } \exists(a', n): \alpha_{a'} A_{na'}^- < 1 \text{ or } \exists(a', n'): A_{n'a'}^+(q_{a'} + q^{|\Delta|}) > q_{a'}^{\max}, \\ 0, & \text{otherwise,} \end{cases} \quad (5.43)$$

in case of the residual queueing model. Observe in the latter case that we now also retain the entire node topology in case any outgoing link exceeds capacity under the assumed margin via $A_{na'}^+(q_{a'} + q^{|\Delta|}) > q_{a'}^{\max}$. Similarly, when adopting a relative flow margin, denoted q^Δ , we find link classification vector $\boldsymbol{\beta}^\Delta \in \mathbb{F}_2^{A \times 1}$ through:

$$\beta_a^\Delta = \begin{cases} 1, & \text{if } \frac{(1 + q^\Delta)q_a}{q_a^{\max}} \geq \beta^{\min}, \\ 0, & \text{otherwise,} \end{cases} \quad (5.44)$$

when adopting the traditional static capacity restrained model, or:

$$\beta_a^\Delta = \begin{cases} A_{na}^- \parallel A_{n'a}^+, & \text{if } \exists(a', n): \alpha_{a'} A_{na'}^- < 1 \text{ or } \exists(a', n'): A_{n'a'}^+(1 + q^\Delta)q_{a'} > q_{a'}^{\max}, \\ 0, & \text{otherwise,} \end{cases} \quad (5.45)$$

in case of the residual queueing model. When revisiting the example in Figure 5.17, adopting a sufficiently large flow margin would have caused the super-scenario to also include ‘‘almost’’ bottleneck link a_2 and subsequently would yield a lossless result when choosing $\mathbf{D}^{s,*} = \mathbf{D}^{\max}$.

Adopting a ‘‘sufficient’’ flow margin ensures we obtain a lossless aggregation result across our demand scenarios, but we still do not know (a-priori) what this ‘‘sufficient’’ margin is to achieve this, unless we again equilibrate all demand scenarios. To avoid this, we aim to investigate if we can quantify the value of a ‘‘sufficient’’ margin in a number of case studies presented in the next chapter.

5.9 Synthesis and discussion

In this chapter we proposed a novel decomposition method for static traffic assignment procedures and showed its potential in the case of an advanced capacity constrained model. This method is tailored towards applications that have a fixed supply while allowing the demand to vary. The proposed method consists of two stages. The first stage constructs a minimum travel time and delay subnetwork, while the second stage consolidates the path set. The delay subnetwork is typically smaller than the original network and can be constructed by equilibrating a single super-scenario demand matrix. The resulting delay subnetwork yields potentially lossless results when adopting the residual queuing model of Bliemer et al. (2014). The decomposition method is equally compatible with other traffic assignment models, such as traditional capacity restrained models, albeit that they result in information loss when they incorporate hypocritical delay in their (path) travel time function.

To reduce the likelihood of failing to capture some of the critical-delay links in the final delay subnetwork, we proposed various approaches among which two different ways of constructing the super-scenario: via a maximum demand and average demand approach. In addition, we proposed to consider either an absolute or relative flow margin in order to include more links in the delay subnetwork, even when these links do not exhibit any hypercritical delay in the super-scenario (but are comparatively close to doing so).

The next chapter aims to investigate the effects of these different approaches in order to provide recommendations on the parameter settings such that a-priori lossless results become more likely.

6 Case studies in delay subnetwork construction and calibration

In this chapter we present a number of case studies based on the decomposition method introduced in Chapter 5. We mainly focus on the calibration and choice of parameters involved in constructing a delay subnetwork that exhibits no information loss under the demand scenarios considered. The results presented in this chapter suggest that it is possible to adopt parameter settings such that we can ensure a lossless result at a relatively small performance penalty compared to not taking any precautionary measures.

The purpose of the case studies is twofold. First, in Section 6.1, we investigate how to best construct super-scenario s^* and absolute/relative flow margins $q^{|\Delta|}, q^\Delta$, respectively. This in order to construct the minimal delay subnetwork containing all bottlenecks across all considered demand scenarios. We use a strategic planning network of the (inner) city of Amsterdam (Amsterdam I) for this purpose. Our second case study, in Section 6.2 demonstrates the capability of the proposed method to reconstruct the original travel times when all bottlenecks are present in the delay subnetwork. In this case study we apply the model to the large scale network of Gold Coast (Australia). Here, we also assess the potential computational gains by comparing the cost of reaching equilibrium on the original network, to the cost of reaching equilibrium on the decomposed free-flow and delay subnetwork, with and without the path consolidation method. A summary of the results is provided in Section 6.3.

All case studies adopt the residual queueing model of Bliemer et al. (2014), introduced in Section 5.5, combined with the node model originally proposed in Tampère et al. (2011). We do so for the following reasons: (i) this model is more realistic than a capacity restrained assignment model, due to its capacity constrained nature, (ii) it places queues in its correct location, i.e. in front of bottlenecks rather than in the bottlenecks, (iii) it filters flow beyond a bottleneck such that the delay subnetwork only captures true bottlenecks and, more importantly, there are less of such bottlenecks compared to a capacity restrained approach leading to higher computational gains. This model is therefore both a better fit with respect to the decomposition method, as well as being more capable in general.

The decomposition and path consolidation methods proposed in Chapter 5 have been implemented in C++, as prototypes, within the StreamLine traffic assignment framework. This framework is part of the OmniTRANS transport planning software suite, kindly provided by DAT.Mobility (The Netherlands).

6.1 Super-scenario calibration: Amsterdam case study

This first case study is performed on the Amsterdam I network, see Figure 6.1. It contains major roads, but leaves out most local streets. We first investigate the effect of the two different approaches - the maximum demand and average demand approach - in constructing the trip matrices for super-scenario s^* .

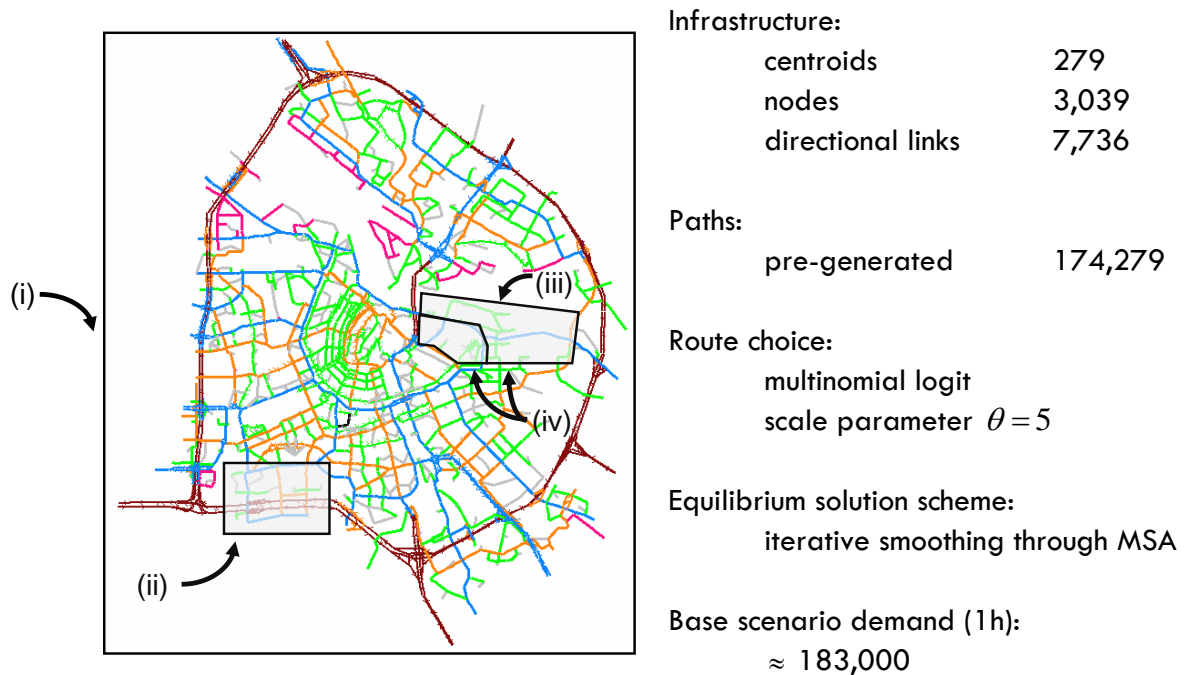


Figure 6.1: Amsterdam I network and its characteristics. Also depicting the four areas where demand variations are carried out according to different scenario types.

We explore a total of four different ways to vary demand across scenarios to cover a wide range of possible applications and interactions: (i) Uniform demand variations across all zones, (ii) varying trip attractions, i.e. trips towards zones, for a subset of zones in a specific area, (iii) varying trip productions, i.e. trips departing from zones, for a subset of zones in a specific area, and (iv) varying both productions and attractions for a subset of zones in a specific area. The considered areas relating to these scenarios are shown in Figure 6.1 and are selected because they represent areas in Amsterdam that have seen rapid development. The number of zones and demand per considered region varies: Scenario (ii) holds 15 zones with a base attraction of $\approx 6,861$ trips, Scenario (iii) holds 11 zones with a base production of $\approx 2,065$ trips, while the combined production/attraction scenario (iv) holds 31 zones with identical base attraction to scenario (ii) and a base production of $\approx 2,800$ trips. The base scenario, i.e. 0% demand increase, holds a synthetic peak demand matrix for a hypothetical base year with according (relatively heavy) congestion. Since all scenarios vary demand by upscaling demand compared to the base case \mathbf{D}^{base} , it therefore holds that $\mathbf{D}^{\text{base}} \leq \mathbf{D}^s, \forall s \in \{1, \dots, S\}$.

The variation in demand across the considered scenarios is chosen to be in the range of 0-20% for the uniform approach of (i), versus 0-40% in the local scenarios, i.e. approaches (ii), (iii) and (iv), respectively. The idea behind having larger variations for the smaller regions is to compensate for their smaller absolute number of base trips. Results are obtained for each of the four different approaches for both types of super-scenario construction; the maximum demand approach via Equation (5.40) and the average demand approach of Equation (5.41). All simulation runs comprised 30 iterations⁹ in order to approximate a stochastic user equilibrium, where the MSA procedure is adopted to smooth iteration results, adopting a smoothing factor $\sigma = 2/3$, in line with Polyak (1990). To illustrate the level of convergence under this

⁹ Given the large amount of different scenarios we tested, we chose a fixed number of iterations that yielded results relatively close to what can be considered equilibrium flows while at the same time minimising the computation times across all test scenarios.

configuration we provide the gap value for each iteration in Figure 6.2, demonstrating we achieve a decent gap of ± 0.0001 for such a limited number of iterations. This gap is computed via the following well-known gap function:

$$\text{GAP} = \frac{\sum_{z=1}^Z \sum_{z'=1}^Z \sum_{p=1}^P P_{zp}^+ P_{z'p}^- f_p (h_p - \tilde{h}_p)}{\sum_{z=1}^Z \sum_{z'=1}^Z \sum_{p=1}^P P_{zp}^+ P_{z'p}^- f_p \tilde{h}_p}, \quad \text{with } \tilde{h}_p = \min \{h_p \mid P_{zp}^+ P_{z'p}^- h_p = 1; z, z' \in \{1, \dots, Z\}, p \in \{1, \dots, P\}\}, \quad (6.1)$$

where one compares the difference in cost of each path to the minimum cost path of the od pair weighed by each path's assigned path flow rate.

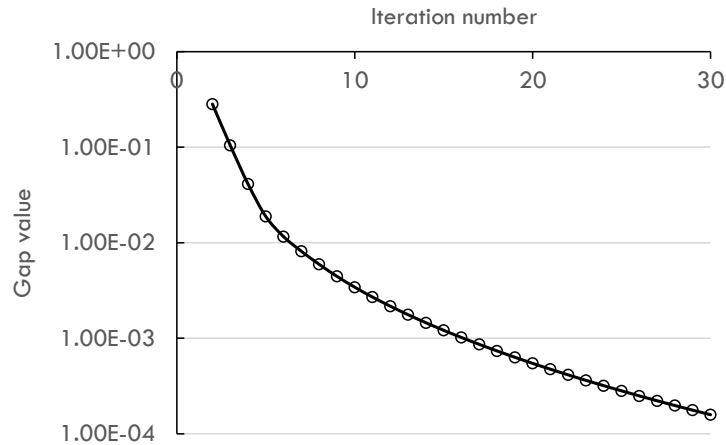


Figure 6.2: Gap for base scenario convergence over first 30 iterations. All other scenarios demonstrated to show similar gaps.

To determine how well the super-scenario performed, we determine the number of missing critical-delay links compared to a lossless result including all required infrastructure. The lossless result is found by equilibrating each demand scenario separately and taking the union of the critical-delay links across all demand scenarios.

6.1.1 Super-scenario results without flow margin

Let us first discuss to what extent the two super-scenarios miss any critical-delay infrastructure without imposing any flow bandwidth margins, i.e. $q^\Delta = q^{|\Delta|} = 0$. Results under uniform demand scaling, referred to as scenario (i) in the previous section are depicted in Figure 6.3. The horizontal axis displays the various levels of demand increase, while the two vertical axis denote the absolute number of critical-delay links (left) and percentage of missing links found (right). Results are presented incrementally, meaning that in, for example, the +8% demand entry in Figure 6.3 we counted all unique critical-delay links found in demand scenarios +1%, +2%, up to and including +8%. A critical-delay-link is considered missing when it is present in any of the equilibrated stand-alone demand scenarios (up to the current demand increase percentage), but is absent in the delay subnetwork identified by the super-scenario.

When adopting the maximum demand based super-scenario, we see that it captures virtually all bottlenecks and misses $<1\%$ of the critical-delay links across all considered demand scenarios. This $<1\%$ represents a total of 4 links across 2 nodes. On the other hand, adopting an average demand super-scenario performs poorly, with an increasingly large number of

critical-delay links missing as is clear from Figure 6.3(b). This stems from the fact that the average yields too low a demand to capture the newly emerging bottlenecks.

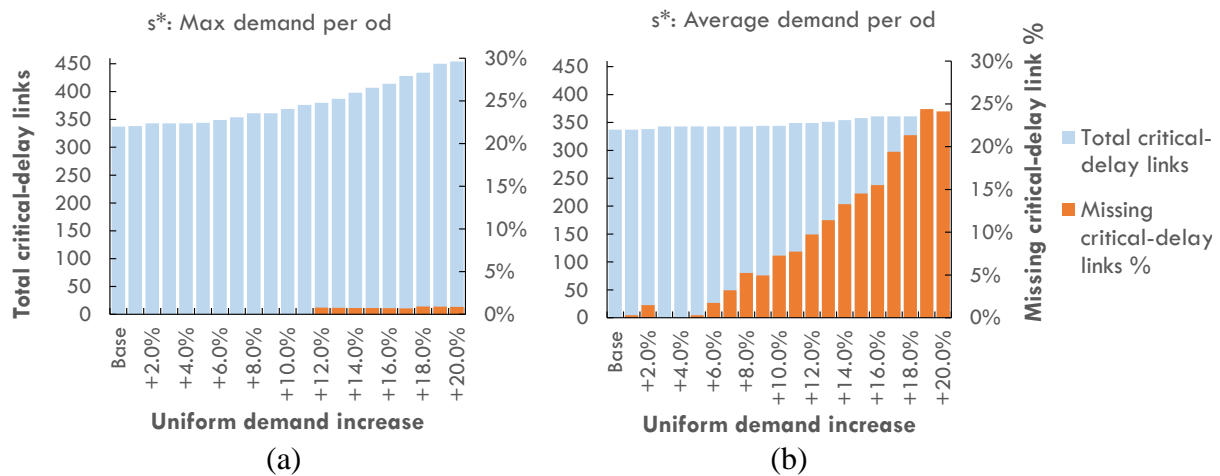
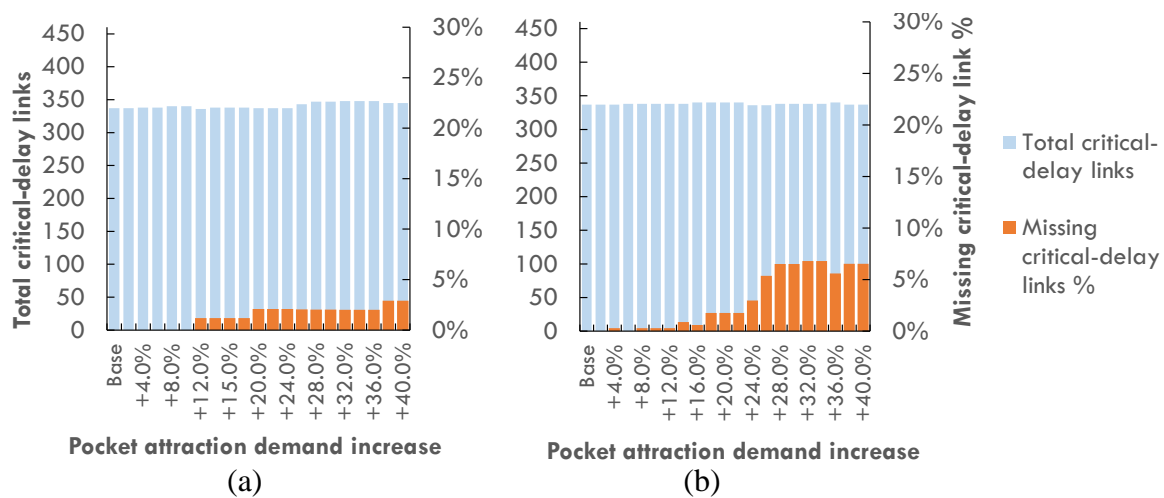


Figure 6.3: Total critical-delay links versus missing critical-delay links, as a percentage of total critical-delay links, under uniform demand scaling (i) for super-scenarios under (a) maximum demand, (b) average demand.

Let us now consider the other three application scenarios, where demand is being varied more locally. Results are depicted in Figure 6.4. Under local variations of demand, the maximum demand based super-scenario is again outperforming the average demand approach. Compared to uniform variations, local demand variations do increase the number of missing critical-delay links. In the super-scenario under maximum demand, we find a maximum of 2.9% missing bottlenecks, equating to 10 critical-delay links. This occurs in both the attraction and combined production/attraction scenarios, see Figure 6.4(a), and (e). The only scenario type which would so far yield a lossless result is the local production scenario under the maximum demand approach, Figure 6.4(c). This is likely due to the fact that this scenario has the fewest zones and smallest base demand, so more significant changes in demand are needed to trigger missing bottlenecks.



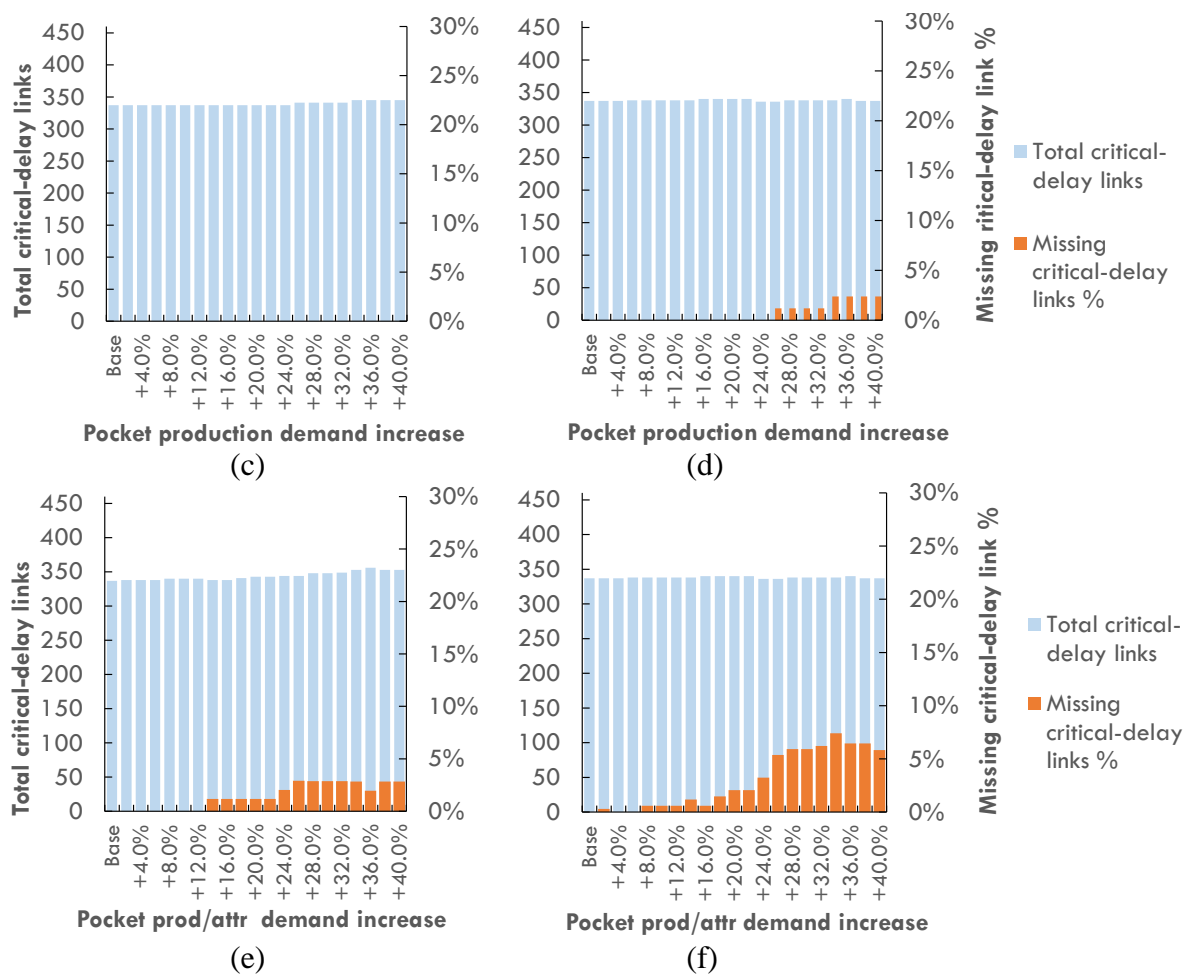


Figure 6.4: Local attraction scenario (ii) compared to super-scenarios with, (a) max demand, (b) average demand. Local production scenario (iii) compared to super-scenarios, with (c) max demand, (d) average demand. Local production/attraction scenario (iv) compared to super-scenario with (e) max demand, (f) average demand.

6.1.2 Super-scenario results with bandwidth margin

Let us now investigate what values of q^Δ , or $q^{|\Delta|}$ would lead to a lossless result across the entire range of scenarios discussed in the previous section. Further, we discard the average demand approach for constructing the super-scenario, due to its significantly poorer performance across all considered application scenarios.

To determine the absolute and relative flow margins, we assessed, for each of the identified missing critical-delay links, the difference between the measured flow rates and the link's capacity. This was done on a per demand scenario basis. The result for uniform demand scenario (i) is provided in Figure 6.5. Note that on the x-axis, we only plot the demand scenarios where missing critical-delay links are detected and therefore omit demand increase percentages below 12%.

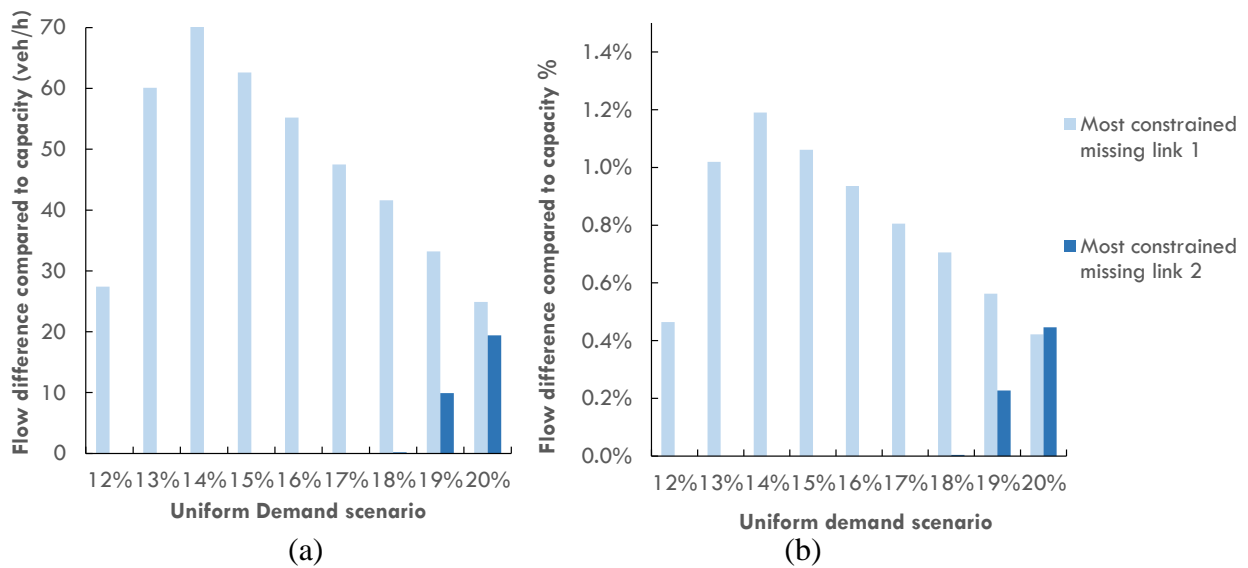


Figure 6.5: Uniform demand scenario (i) results for; (a) absolute, and (b) relative difference between measured flow and capacity.

Both the absolute and relative measured flow margins are depicted in Figure 6.5, on a per *most constrained missing link* basis. The most constrained missing link is defined as the outgoing link that is the most restricting bottleneck of its node at the considered demand level. For example, in the uniform scenario, four links in total were missing, but they relate to only two nodes. So, there exist exactly two most constrained missing links; one per node.

Observe that for the demand levels with a scaling between 12 and 18%, only the first of the two identified most constrained missing links was absent, while the other most constrained missing link only surfaced for the 19%-20% demand levels. The results for the demand scenario types (ii) and (iv) are provided in Figure 6.6. No missing links were detected for demand scenario type (iii), when using the maximum demand approach, and hence no further results are shown for this scenario.

The most extreme flow margins for the most constrained missing links are found in the uniform scenario setting. Overall, the smallest and largest maximum absolute flow margins are 26.1, 70.2 (veh/h), respectively. Similarly, the smallest and largest maximum relative margins were found to be 0.985%, 1.19%, respectively. Imposing these margins results in including all originally missing critical-delay links, in turn yielding a lossless result. Observe from Figure 6.6(a) and (b) that the maximum relative margin does not relate to the link with the maximum absolute margin, indicating this missing link has a relatively low capacity. A similar situation occurs in Figure 6.6(c) and (d).

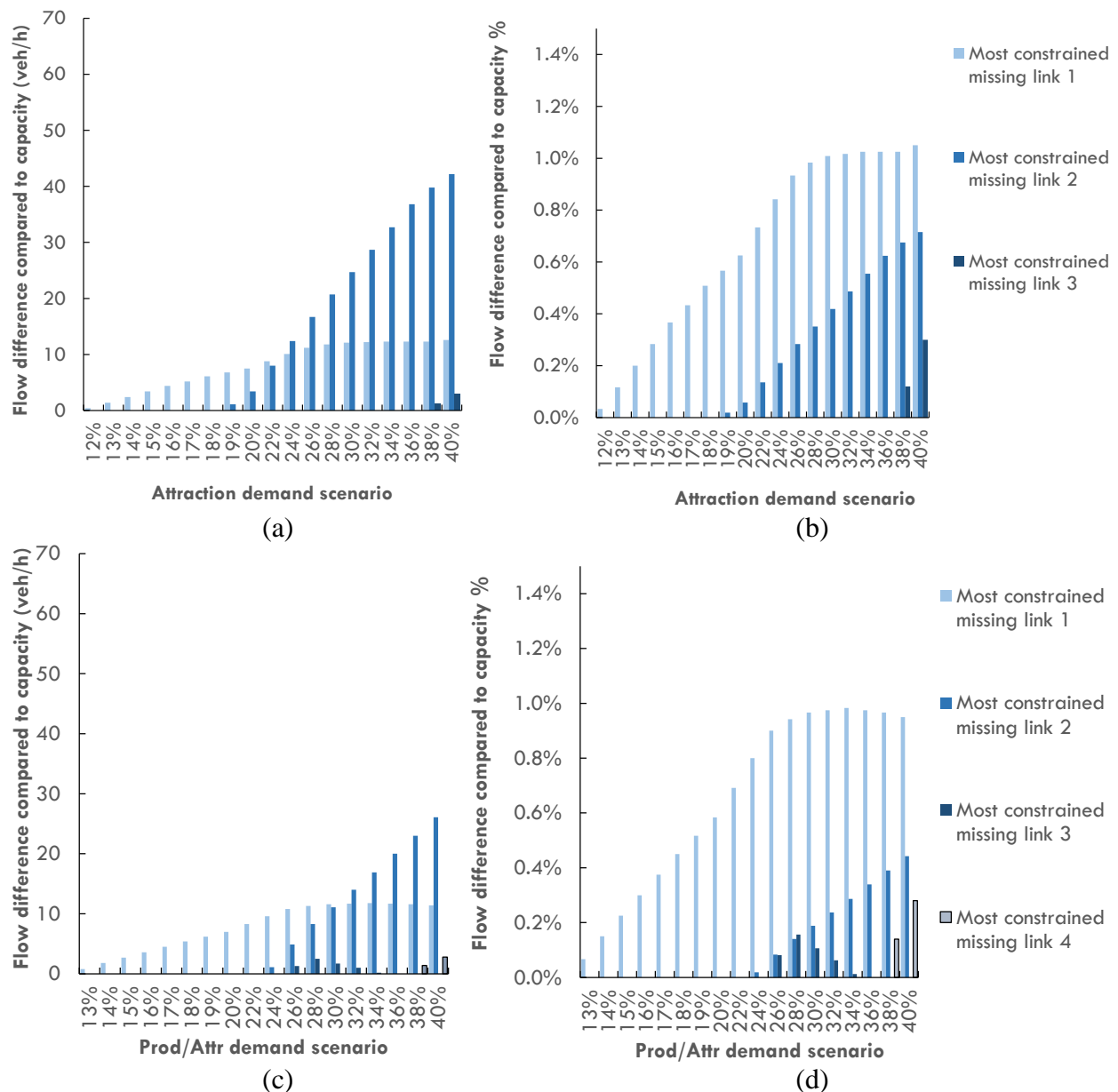


Figure 6.6: Attraction demand scenario (ii), with (a) absolute and (b) relative difference between measured flow and capacity. Production/Attraction scenario (iv), with (c) absolute and (d) relative difference between measured flow and capacity.

Imposing any type of flow margin in the procedure inevitably results in false positives, yielding a delay subnetwork that includes more links than strictly needed. Obviously, we seek to minimise the number of false positives because they contribute negatively to the potential computational gain. Let us therefore compare the absolute margin and relative margin approaches by investigating how many false positives they yield and therefore inadvertently increase the size of the retained delay subnetwork. Results are shown in Figure 6.7.

As can be seen, in all cases, adopting a relative margin – instead of an absolute margin - results in a smaller delay subnetwork. This can be attributed to the fact that, among the dominant missing links found at each demand level, there is always at least one higher capacity (arterial) link. These arterials exhibit relatively high absolute flow margins, capturing a larger share of false positives among local streets. This effect is mitigated when one adopts a relative margin

instead. We therefore propose to adopt a relative flow margin approach. Recall from the flow margin based results that with $q^\Delta = 1.19\%$ a lossless result is achieved. Hence, we propose to use this relative flow margin as a general lower bound such that we choose $q^\Delta \geq 1.19\%$. The penalty for this approach appears to be a significant amount of false positives which, in case of the uniform scenario, leads to a near 50% increase in infrastructure retained in the delay subnetwork. This number, however, needs to be put in perspective because the computational gain that can be achieved is relative to the size of the original network and not the delay subnetwork under $q^\Delta = 0$. When we compare to the original number of links in the network (3,868), to the delay subnetworks without and with the relative flow margin, we find the more representative absolute percentage increase of 4.4% (from 11.6%, to 16% of the original network).

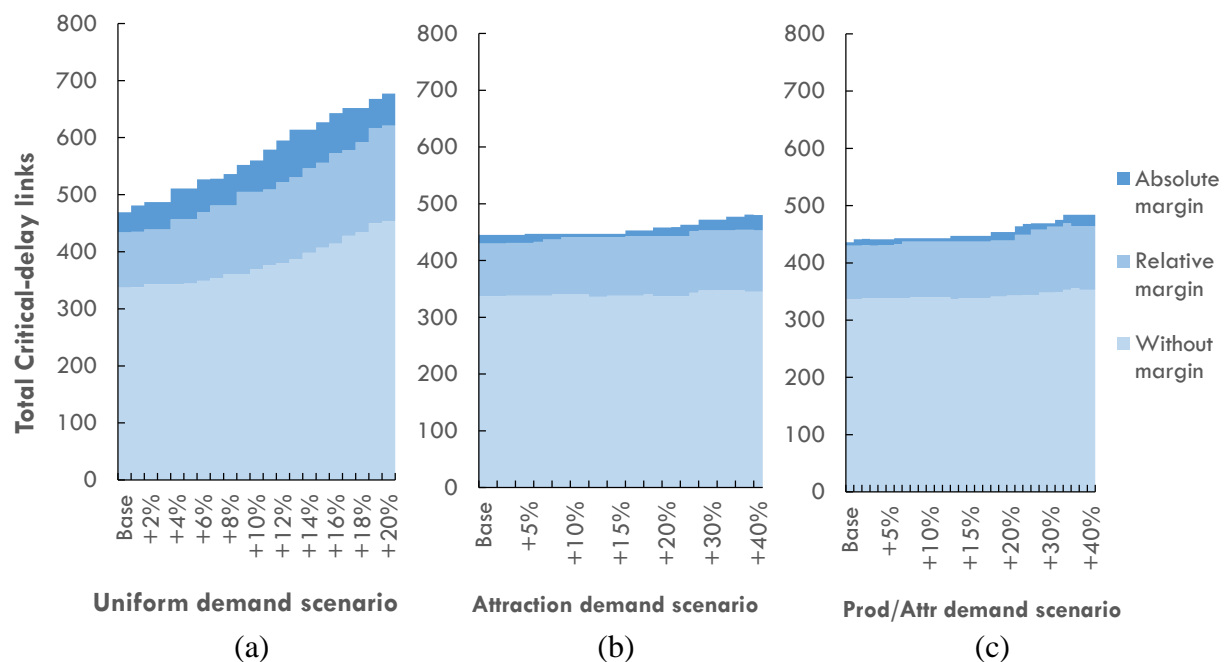


Figure 6.7: Total critical-delay links including false positives under the minimum absolute and relative margins found that yield a lossless result for (a) uniform demand scenario type (i), and (b) attraction demand scenario type (ii), and (c) combined production/attraction demand scenario type (iv).

6.2 Potential computational gains: Gold Coast case study

In this section, the delay network decomposition and path consolidation method are applied to the large-scale network of Gold Coast (Queensland, Australia) depicted in Figure 6.8(a). The purpose of this case study is to demonstrate the potential computational gains that are possible when adopting our proposed method. In addition, we verify that the obtained hypercritical delay is indeed identical to the original network's delay. To be able to assess the maximum computational gain possible for this case study, we consider only a single demand scenario. In such a case there is no need for applying a flow margin either, hence false positives are absent. This allows for a direct comparison of the delay subnetwork assignment to the original network assignment. Note that we do not consider the cost of creating the delay subnetwork here, because it is difficult to objectively incorporate such a fixed cost in the context of an unknown number of varying demand scenarios.

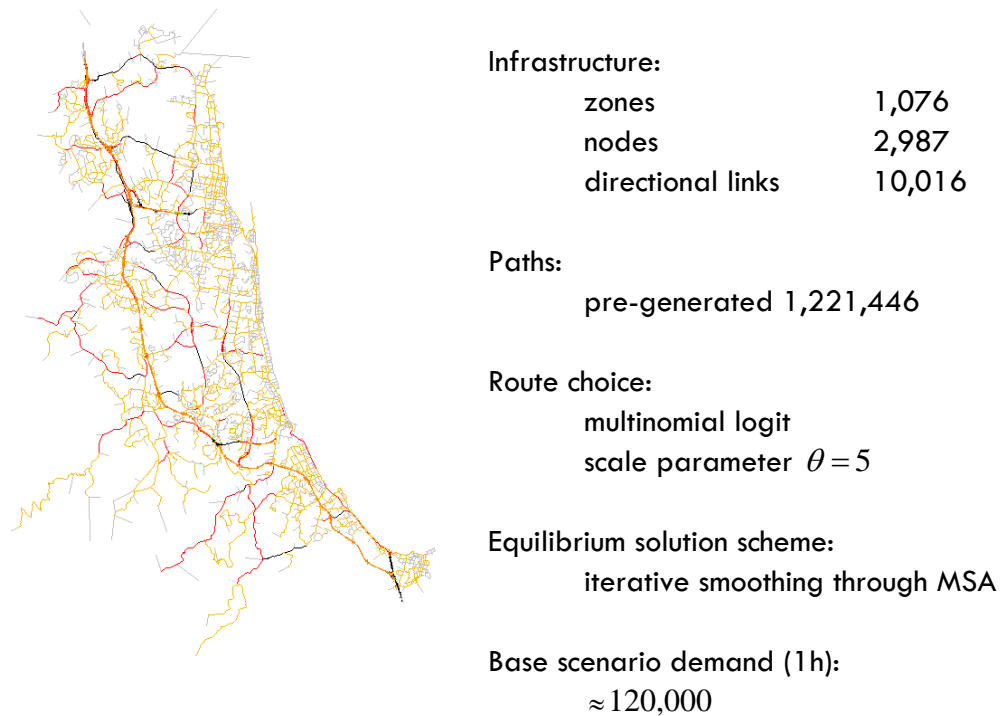


Figure 6.8: Gold Coast network and its main characteristics.

After equilibrating the single demand (base) scenario, the network delay decomposition method is applied, yielding the delay subnetwork as shown in Figure 6.9(a). We do not depict the links within the areas of distortion so we can focus on the portion of the original network that is retained and exhibits delays.

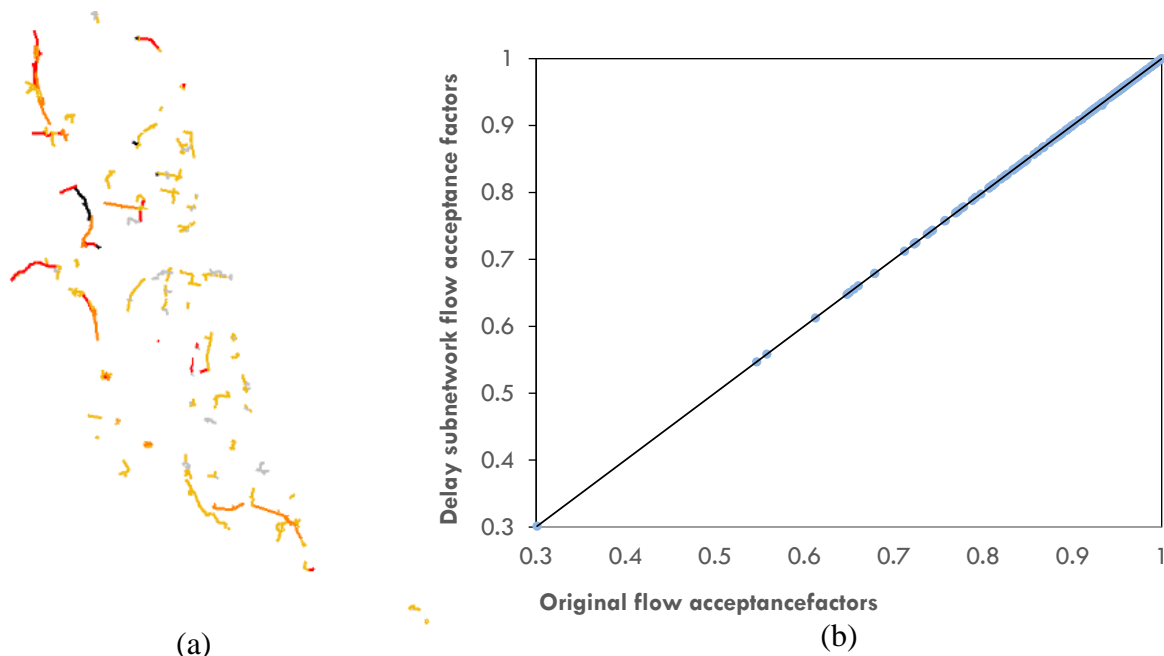


Figure 6.9: (a) Gold Coast delay subnetwork, (b) comparing equilibrium link flow acceptance factors between original network and delay subnetwork.

To illustrate that the equilibrated aggregate network yields the same result as the original network, the assignment results of the original network and the delay subnetwork are compared. Recall that path delay is conditional on the encountered flow acceptance factors α .

Therefore, as long as the flow acceptance factors in the original network and delay subnetwork are identical, the path delay is also identical. A graphical comparison of the found flow acceptance factors is provided in Figure 6.9(b) through a 45-degree plot. We additionally capture the compliance of the delay subnetwork's flow acceptance factors with the original results via the following Root Mean Squared Error (RMSE) measure:

$$\text{RMSE} = \sqrt{\frac{\sum_{a=1}^A (\alpha_a - \alpha_a^{\text{delay}})^2}{A}}, \quad (6.2)$$

where we found the RMSE to be $3.66 \cdot 10^{-5}$. The reason the result is not exactly zero is expected to be due to computational ordering effects, resulting from the two different ways the travel time computation is executed, which over many iterations inevitably causes very small, but measurable, differences on the link level flow acceptance factors. However, these differences are sufficiently small their impact on the hypercritical path delay was found to be negligible.

The results regarding the computational gains are outlined in Table 6.1. Note that we fixed the number of iterations to 80 to be able to provide an exact comparison between simulation runs. For each of the 80 iteration runs the resulting gap (see Equation (6.1)), was found to be $< 10^{-4}$, which is generally considered sufficient to approximate an equilibrium solution. The computational cost was found to be reduced by 96% to 4% of the original network loading times. If the path consolidation method is not applied and we only consider the delay subnetwork decomposition, network loading times are halved instead. This can be attributed to the reduced number of links per path (10.6 vs 45.7 links on average). Clearly, the combined procedure yields the best results and the path consolidation procedure contributes most in reducing computational cost. This is because the consolidated equidelay path set contains less than 10% of the original paths ($< 100,000$ vs $1,221,446$). With 88.62 seconds over 80 iterations, it now only takes 1.1 seconds to perform network loading per iteration compared to 27.2 seconds in the original network.

Table 6.1: Effects of delay decomposition and path consolidation on network loading and topology (80 iterations).

| Network | Links | Paths | Total path links | Total network loading time (s) |
|--|-------|-----------|------------------|--------------------------------|
| Original | 5,076 | 1,221,446 | 55,852,786 | 2,178.48 |
| Delay-decomposition + Original path set | 836 | 1,221,446 | 12,919,862 | 902.57 (-59%) |
| Delay-decomposition + Path set aggregation | 836 | 99,528 | 1,142,338 | 88.62 (-96%) |

Recalling the applications this method is targeting, i.e. quick-scan methods and matrix calibration procedures (among others), we observe that this reduction in network loading can be used to either explore more demand scenarios, run a more elaborate matrix calibration procedure, or allow one to switch from traditional assignment methods, such as the capacity restraint static assignment procedure, to a more capable one, without the penalty of increased

computation times. There are however still three limiting factors that affect the computational reductions outlined in Table 6.1. First, there is a fixed computational overhead to pay for the initial construction of the delay subnetwork and consolidated path set. This cost also necessarily includes the equilibration of the super-scenario, plus a small penalty for storing the minimum path travel time costs, persisting the delay decomposed subnetwork, and extracting the consolidated path set. In our experience this fixed cost roughly equates to the –by far - most costly component in this process, the equilibration of the super-scenario.

Also note that the cost for performing path choice remains roughly the same because path choice occurs on the level of original paths, regardless whether or not the consolidated path set is used during network loading. Lastly, if the modeller desires a lossless result, a non-zero flow margin must likely be adopted, which in turn increases the size of the delay subnetwork somewhat. Therefore, the 96% computational reduction is an upper bound and the actual computational gain will be somewhat less depending on the application context.

However, it is clear from Table 6.1 that the fixed overhead costs can quickly be “earned back”. For this case study, two demand scenarios suffice to compensate for the cost of constructing and equilibrating the delay subnetwork and equidelay path set¹⁰. Clearly, when hundreds or even thousands of demand scenarios need to be explored, this fixed overhead becomes increasingly less significant.

6.3 Synthesis and discussion

In this chapter we explored a number of case studies to investigate and calibrate the parameters involved in aiding the user to construct a lossless delay subnetwork using a single super-scenario demand matrix. The case studies demonstrate that, while the proposed method is potentially lossless, such a result cannot be guaranteed a-priori. To minimise information loss, choosing an appropriate relative flow margin threshold q^Δ is investigated. This margin includes additional links in the delay subnetwork and increases the likelihood of capturing all bottleneck infrastructure in the super-scenario. In the Amsterdam case study we investigated the minimum required values for q^Δ . It was found that adopting a relative flow margin of $q^\Delta \geq 1.19\%$ resulted a lossless results for this particular study, which means including the node infrastructure of outgoing links with flows at 98.81% of capacity.

We also explored the potential computational gain of adopting this method on the large scale network of Gold Coast, Australia. It was found that by applying delay subnetwork decomposition, network loading times were reduced by 59% compared to the original network loading cost. When supplementing this procedure with the path consolidation procedure, a 96% computation time reduction is achievable.

¹⁰ This is under the assumption that the path choice procedure takes less time than the original network loading procedure, something which is typically the case when adopting a fixed path set approach.

Part III

7 Aggregation methods

Part II of this thesis illustrated how to reduce the computational burden by exploiting specific characteristics of a traffic assignment procedure in the context of its application domain. Now, in Part III, instead of investigating the optimisation of a single traffic assignment model, we turn our attention to a *multi-scale environment*, where multiple traffic assignment models are operated alongside each other within the same spatial domain, but at different levels of granularity. We propose novel methodology to automatically generate traffic assignment model inputs at the desired level of detail. We employ both disaggregation and aggregation methods to do so. In this chapter we discuss the (dis)aggregation literature in the context of traffic assignment models. The concepts introduced in this chapter are then fused with the traffic assignment literature discussed in Chapter 3 leading to the newly proposed (dis)aggregation methods discussed in Chapter 9 and Chapter 10.

The existing (dis)aggregation literature - relevant to traffic assignment models - is both limited and broad at the same time. It is broad because the used techniques are not necessarily specific to transport, they are applied across a multitude of research areas. Yet, it is limited because the majority of research on traffic assignment methodology revolves around extending and proposing novel assignment models, often resulting in increased procedural complexity. Methods concerning the simplification of procedures or its inputs is far less common. A brief exception to this trend occurred in the 1980s, when computers became more widely available, yet their computation power was still limited. Today, the research that does focus on the representation of traffic assignment models is mostly driven by the objective of computational gains, rather than achieving inter-model consistency.

We first discuss the existing types of (dis)aggregation in Section 7.1. Existing (dis)aggregation methods in the context of traffic assignment are the focus of Section 7.2. This is followed by aggregation and representation methods for each of the traffic assignment model inputs, namely, zonal aggregation (Section 7.3), connector and centroid representation (Section 7.4) and network aggregation (Section 7.5). Techniques used to implement aggregation methods, such as clustering are discussed in Section 7.6, followed by how this relates to a multi-scale environment in Section 7.7. A brief summary and discussion is provided in Section 7.8.

7.1 Types of (dis)aggregation

The earliest aggregation methods in transport mainly focussed on removing links in a network. Goldman (1966), for example, removed links from a network based on the criteria of unchanged shortest paths. This process of object removal is referred to as *extraction*. In contrast to *extraction* methods, *abstraction* methods not only remove parts of an original representation, but replace them with a proxy, mimicking the original behaviour as much as possible. Abstraction based methods are generally thought of as the more desirable approach, see for example Chan (1976), and are therefore considered to be more capable. A well-known abstraction method in traffic assignment models is the replacement of local roads in a zone with connector links.

Chan (1976) also differentiated between *uniform* aggregation methods and *non-uniform* aggregation methods (or ‘focal’ aggregation as the author refers to it). Uniform aggregation methods apply to the entire network while non-uniform methods only impact on part(s) of the network. In this thesis we do not consider non-uniform methods because we adhere to an integrated solution method. This, in our view, means considering the network as a whole, i.e. uniformly, resulting in the desired outcome without having to focus on sub-areas as a post-processing step. In other words, it might be possible that some parts of the network become coarser than others, but this is due to the method being capable of incorporating spatial differences or constraints, instead of having to apply different methods in succession on different parts of the network.

While it is common for aggregation methods to aggregate *data* (model inputs), aggregation can also be applied to *procedures*. This does not mean it groups procedures, but rather changes its underlying assumptions. For example, when replacing a microscopic model with a macroscopic approach, the underlying assumption that vehicles require individual modelling is replaced by assuming that traffic can be modelled on a more aggregate level through average flow rates. Figure 0.1 depicts the taxonomy of the aforementioned (dis)aggregation methods. This does not reflect an ordering, it merely demonstrates a way to categorise existing methods by traversing the tree from the root to a leaf.

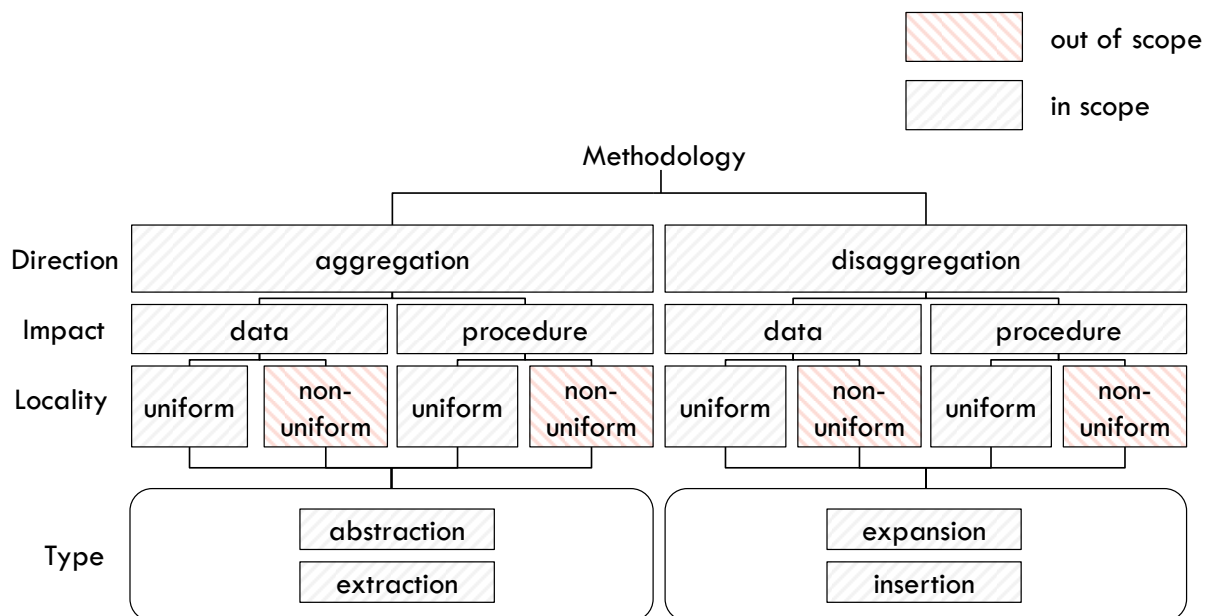


Figure 0.1: General taxonomy for traffic assignment (dis)aggregation methods.

The opposite of abstraction is termed *expansion*, while we propose the term *insertion* to reflect the opposite of an extraction process, both of which are types of disaggregation methods. Based on the above classification, the following general definitions are proposed to describe aggregation, disaggregation, respectively in the context of traffic assignment models:

Definition 7.1: Traffic assignment model aggregation

The process of decreasing the complexity of a traffic assignment model by aggregating its data, procedures, or both, either uniformly or not, by: (i) extracting original components, (ii) abstracting original components, or (iii) a combination of (i) and (ii).

Definition 7.2: Traffic assignment model disaggregation

The process of increasing the complexity of a traffic assignment model by disaggregating its data, procedures, or both, either uniformly or not, by: (i) expanding original components, (ii) inserting new components, or (iii) a combination of (i) and (ii).

7.2 Aggregation methods in traffic assignment

Most existing representation altering methods in the traffic assignment literature are aggregation methods pertaining to model inputs. A clear distinction is made between *zonal aggregation* and *network aggregation*.

Zonal aggregation scales the zoning system, i.e. the geographical areas. This means that given some original representation existing zones are grouped (or split up in case of disaggregation). In Openshaw and Taylor (1979) this is referred to as the *scaling effect* of zoning. Traditionally, zonal aggregation methods in traffic assignment only consider scaling effects based on a single original zoning system. Besides scaling effects there also exist *zoning effects*. Zoning effects refer to changes in model results when one adopts different representations at the same level of scaling. Zoning effects provide insight in inconsistencies that can arise between different model representations. We find that the term zonal aggregation is used rather loosely in the traffic assignment context because it can refer to the aggregation of centroids as well as zones. This is unfortunate since centroids do not equate to zones. However, to be consistent with the existing literature we concede to discuss both zones and centroids in the context of zonal aggregation.

Network aggregation is concerned with the grouping links, nodes, and sometimes connectors in the physical network. *Connector representation* is discussed separately because it has a number of distinct features relating to cost and placement that differ significantly from ordinary links. Therefore, we also refrain from using the term aggregation here, because connectors are not the result of a simple grouping of data points; they represent a portion of the network that is abstracted into a virtual link.

7.3 Zonal (dis)aggregation

Zonal (dis)aggregation methods originally emerged as a natural extension to statistical methods with a spatial component. It involves spatial data that needs to be pooled for a variety of possible reasons, for example to ensure the privacy of individuals, guarantee statistical significance, or a lack of resources to deal with the data in a disaggregate manner. Aggregating such data points can have unexpected and undesirable side effects. The first person to notice that the choice of geographic area used to aggregate data points had an impact on the aggregate results was Dr. Henry Sheldon in 1931 (cited in Gehlke and Biehl, 1934). Gehlke and Biehl (1934) confirmed this by showing that the correlation coefficients between geographic areas changed depending on the chosen size. Over the years this effect has been reconfirmed in many different areas and approaches (Kwigizile and Teng, 2009; Openshaw and Rao, 1995; Openshaw, 1977), and has since been referred to as the *Modifiable Area Unit Problem* (MAUP).

7.3.1 Modifiable area unit problem and zoning effects

In general, there are an infinite number of possibilities to construct the shapes of geographical zones. Typically, each geographical area is constructed based on the location of individual data points within this area. The number of combinations of grouping these data points is of course not infinite, but while this number is finite, it is typically still extremely large. Therefore, it is notoriously difficult to find an acceptable (base) zoning structure without spending excessive amounts of time. It is also important to realise that, unless each household is modelled separately, any traffic assignment model remains conditional on the chosen zoning system and is therefore affected by the MAUP (Weeks, 2004). Knowing that the zoning system is always just one possible representation out of the many possible options, traffic assignment results will also just reflect one of many possible outcomes, as also pointed out by Páez and Scot (2004).

Addressing the MAUP has traditionally been a statistical exercise more than anything else (Wong, 2001). However, in the last three decades Geographical Information Systems (GIS) have become increasingly popular. Today, they supplement, or even replace, traditional methods when analysing and constructing zoning systems. This seems especially the case in transport related applications. So far, this has resulted in a list of somewhat pragmatic criteria that a zoning system should comply with. These criteria are based on qualitative and quantitative analyses (Ding, 1998; O'Neill, 1991; Baass, 1981) and have also found their way to practice, where they are frequently mentioned as part of, for example, industry guidelines, government recommendations and textbooks (Aecom, 2007; Ortuzar and Willumsun, 2002). Some argue that GIS based approaches are not only a step forward, but can also be regarded as a step backwards, mainly due to a lack of knowledge on the part of the practitioners using the, sometimes, complex software tools required to implement GIS solutions. An interesting discussion on this topic can be found in Fotheringham (2000). Nevertheless, the impact of GIS inspired methods cannot be ignored. An overview of the most commonly referred to criteria when constructing zones is given in Table 0.1.

Table 0.1: Commonly adopted zoning design criteria.

| Criteria | Type | Rationale |
|---|--------------|---|
| Within zone data homogeneity | Quantitative | Low data variance means little error in aggregate form |
| Between zone data homogeneity | Quantitative | Comparable errors between zones |
| Minimise intrazonal trips | Quantitative | Maximise modelled travel demand |
| Adopt census boundaries | Quantitative | Data availability is mostly tied to this spatial unit |
| Adopt physical, political and historical boundaries when sensible | Qualitative | Implicit information is assumed to be contained within these boundaries |
| Convex area shape, i.e. no "holes" | Quantitative | Convex shape attributes to spatial homogeneity |
| Within zone connectivity | Quantitative | If disconnected travel time homogeneity suffers |

There exist more criteria than listed here, and some of the criteria conflict with each other as well, indicating the difficulty to verify the correctness, or adoption, of such rules. In Martinèz

et al. (2009) the authors also highlight this issue and instead propose to construct a zoning system that is not based on the available census boundaries, but instead utilise a collected dataset of smoothed geocoded travel demand (converted into a density surface) and base their zoning system on this data source. This seems a natural approach, because when the underlying assignment method relies on travel demand, why not construct the zones based on this same metric? Similarly, in Openshaw and Rao (1995) it was argued that census boundaries are not ideal. Instead, data should be released at a highly detailed level (or should be disaggregated as such) and modellers should then construct an appropriate zoning system specific to their application, instead of relying on rigid census boundaries provided by governments.

We agree with the line of reasoning in Openshaw and Rao (1995) and therefore either assume to have access to a very fine zoning system at a high level of detail, or construct such a zoning system ourselves by disaggregating large zones into smaller zones, to subsequently provide methodology to scale it to the appropriate (lower) level of detail.

7.3.2 Zonal scaling

Typically, zonal aggregation is used to reduce computational cost. An exception to this is proposed by Daganzo (1980a), he proposes a disaggregation method, increasing the number of zones in order to improve the accuracy of results. Daganzo argued that the edges of the original zones are misrepresented by a coarse zoning structure, leading to a “spatial aggregation problem”. The method is shown to work in conjunction with the Frank and Wolfe algorithm (1956) solving a traditional static assignment. Daganzo also points out that, in general, there is a lack of well understood rules on how to construct zoning systems and more specifically, where to locate, and how to connect centroids to the physical road network. He also points out there is a lack of understanding in what elements of the road network should be included or excluded. So, even though Daganzo only focussed on the representation of centroids, he was already aware of the fact that all spatial components play a role in choosing an appropriate granularity for the traffic assignment model representation.

Instead of only disaggregation, one can also opt for a combined aggregation-disaggregation approach. Ruddel and Raith (2013) for example propose a method to reduce computational costs by obtaining a better initial solution. This initial solution is based on an aggregated zoning system and then feeds the result back into the original (disaggregate) problem.

Zonal aggregation methods only consider scaling a reference zoning system downward. Bovy and Jansen (1983) provide an early example of this approach where the authors first employ a network aggregation scheme to simplify the road network and based on these results aggregate the zoning system which, in their case, is represented by centroids, see also Jeon et al. (2012) for a similar approach. Based on their results, Bovy and Jansen concluded that: (i) Outcomes of traffic assignment are significantly influenced by the level of detail available in the network, (ii) increasing the level of detail in the zoning system and network representation improves traffic assignment results, and (iii) this effect becomes marginal beyond a certain level. These findings are in line with Daganzo’s earlier observations.

A recent, more sophisticated approach is proposed by Hagen-Zanker and Jin (2015) who propose an adaptive zoning scheme where, dependent on the interaction between two zones, a

choice is made on the granularity of the departure and arrival zone being modelled within the assignment, where there exist multiple levels of detail for each zone (see also van Steijn, 2016). Results show a marked improvement in accuracy for a given computational budget. Observe that this implicitly also affects the original network representation since coarser zones bypass some of the road network.

7.4 Connector representation and centroid placement

Once the zoning system is in place, the interface to the underlying road network needs to be established. This involves choosing the location of the centroids, choosing the number and/or location of connectors, and estimating the cost and/or demand distribution across the connectors. The representation of this interface can have significant consequences for the modelling results. Also note that this is a special type of aggregation because they do not necessarily scale the original input in the traditional sense.

There are various approaches as to where “best” place centroids, these approaches are listed in in Table 0.2, in increasing order given their reliance on data availability. Note that there has been very little attention in the literature on this topic and even the two sources cited here only mention these approaches informally and as side notes to the actual research topic. Also, often, the centroid location is simply assumed given.

Table 0.2: Centroid placement approaches.

| Centroid placement | Required data | Assumption | Mentioned in |
|--|---|--|--|
| Geometric centre | Zone shape | Uniform distribution of trips | Friedrich and Galster (2009), Khatib et al. (2001) |
| Centre of gravity of weighted network nodes | Internal zone topology | Distribution of trips unknown | Friedrich and Galster (2009) |
| City location | City locations | Population density is representative for trips | Khatib et al. (2001) |
| Node with highest accessibility becomes centroid | Internal zone topology and node accessibility indices | Accessibility is representative for trips | Friedrich and Galster (2009) |
| Population-weighted centre of gravity | Spatial population density | Population density is representative for trips | Khatib et al. (2001) |
| Household-weighted density | households locations | household density is representative for trips | Khatib et al. (2001) |
| Centre of gravity of disaggregate trips | Trip origins and destinations | Actual trips representative for trips | Friedrich and Galster (2009) |

The placement and construction of connector costs have received slightly more attention. A good introduction to issues regarding connector costs and their respective placement is given in Friedrich and Galster (2009), they investigated five different schemes for the cost and

placement of connector links. They compare results from a (more) detailed model to some of their simplified approaches, where local roads are replaced by connectors. However, as the authors point out, they only consider travel times and not link volumes. Also, their zoning system is assumed fixed and given. They found that within the tested approaches there was no definitive best solution, suggesting they did not consider enough methods, metrics, or there are simply multiple ways of designing connectors in an acceptable fashion. We also would like to refer the reader to much earlier, but conceptually still interesting, work by Daganzo (1980b) where a method for estimating connector costs (termed access costs) is embedded in a traditional static traffic assignment procedure. It is based on an approach where demand in zones is modelled as a continuum, inspired by earlier work in Newell (1980).

A more pragmatic approach to connector placement is proposed by Mann (2002). In this work, existing connector end nodes are upgraded to a sub-centroid, termed B-node. Each B-node is then assigned a fixed portion of the total demand in order to improve accuracy. This approach is used in practice as well, for example in the Aimsun traffic simulation software. Downsides to this approach are the increased computational cost, reliance on pre-existing connector placement decisions, and the difficulty of constructing a justifiable distribution of demand across the sub-centroids.

Both Jafari et al. (2015) and Qian et al. (2012) confirm that not only the zoning system, but also the choice of connectors can severely impact traffic assignment results. In Qian et al. (2012) the authors add/remove connectors and identify how this impacts results based on reference link volumes. Jafari et al. (2015) not only investigate the effect of connectors, but also propose a method they call bi-level selection. They argue that when network detail is added to an original network, the original connectors might bypass this added detail which is deemed undesirable. They argue network detail should match the zoning as well as the supply-demand interface and propose a bi-level connector design with additional connectors close to the centroid. They then distribute demand across the two levels. Although the number of added connectors, as well as the choice for two levels is rather arbitrary, the observation of matching the level of detail between supply, demand and supply-demand interface is interesting and intuitive.

All these approaches determine connector costs and/or demand distribution deterministically. Benezech (2011) approaches the problem from a more probabilistic perspective. His approach can be considered as a more general method compared to Jafari et al. (2015). Instead of two levels, four anchor points (connectors) are defined around each zone and a logit model is adopted to estimate the distribution of demand across these points. The utilities in the logit model are obtained from an underlying - more detailed - network that is used to determine the travel times to the various anchor points. As the author also points out, the number of four is arbitrary. Similar to the work of Mann (2002) and Jafari et al. (2015), the results do not lead to a connector cost, but effectively split each centroid/zone in four parts.

All in all, existing literature on connector cost and placement is limited. While there exists agreement on the fact that connector placement and costs are important and significantly impact results, existing methods often lack justification for the choices that are being made. As we can observe from Table 0.3, connector placement and the number of connectors is virtually always

chosen arbitrarily or based on an assumption that is not verified. Also, centroid design is hardly ever considered in conjunction with connector design even though connector costs rely on the location of the centroid. Yet, the validity of the adopted centroid placement is never questioned.

Table 0.3: Overview of approaches regarding centroid and connector placement. Colours: orange - arbitrary, light orange- assumed given, green – justified.

| Approach | Connector placement | Connector cost | Number of connectors | Centroid design | Authors |
|---|---------------------------------|--|----------------------|----------------------------------|------------------------------|
| <i>Deterministic disaggregation</i> | Assumed given | Assumed given | Assumed given | Disaggregate to connector nodes | Mann (2002) |
| <i>Single closest node</i> | closest node to centroid | Zero cost | 1 | Assumed given | Friedrich and Galster (2009) |
| <i>Single concentric sectors</i> | closest node to sector centroid | Zero cost | 1 (per sector) | Assumed given | Jafari et al. (2015) |
| <i>Advanced single concentric sectors</i> | closest node to sector centroid | regression analysis data (land use/traffic states) | 1 (per sector) | Assumed given | Friedrich and Galster (2009) |
| <i>Double concentric sectors</i> | closest node to sector centroid | Zero cost | 1 (per sector) | Bi-level centroid disaggregation | Jafari et al. (2015) |
| <i>Equal travel time isochrones</i> | Equal travel time to centroid | Average travel time, potentially based on simulation | Topology dependent | Assumed given | Friedrich and Galster (2009) |
| <i>Stochastic disaggregation</i> | “variety of directions” | Stochastic, distribution based estimation | 4 | Disaggregate to “anchor” points | Benezech (2011) |

7.5 Network aggregation

On the supply-side of traffic assignment, network aggregation emerged as a popular method to reduce computational complexity of a model. Early approaches (Bovy and Jansen, 1983; Long and Stover, 1967; Goldman, 1966) mainly considered network extraction methods. There are some more recent examples of network extraction as well, see for example Jeon et al. (2012) or Cui (2016). Early on, it was quickly established that extraction methods are undesirable (Chan et al., 1968) for a number of reasons. It causes (i) reduced capacity on the network, (ii) unrealistic diversion of traffic to remaining arcs, and (iii) reduction, or absence, of network connectivity. While this has been known for decades, practitioners still use ad-hoc extraction/insertion methods to match network detail to their choice of traffic assignment model, highlighting one of the key problems in the current state of practice. This observation is not at all new, Friesz (1985) already pointed out the need for novel methodology to change the perspective on the use of network aggregation, an observation that arguably still rings true today.

Early network abstraction methods are found in for example Chan (1976) and Zipkin (1980), where original links, or nodes, are replaced by a simplified proxy. In Chan (1976) this results in three types of proxies: bypass links, interzonal links and intrazonal links. Error analyses demonstrated that total travel time remained unaffected under this scheme. Other more recent network abstraction methods are found in for example Connors and Watling (2015, 2008), or Jafari and Boyles (2016). A downside of these more recent approaches, apart from solely considering network representation, is that they are tied to a particular traditional static traffic assignment formulation, or that they adopt highly simplified cost functions that limit their use in a practical context.

Aforementioned methods are uniform, i.e. applied to the entire network. Some aggregation methods however are non-uniform, sub-area aggregation methods for example only simplify parts of the network. Examples of such local aggregation methods can be found in Jafari et al. (2016), Boyles (2012), or Zhou et al. (2006). As mentioned earlier, we do not consider non-uniform methods in this thesis.

7.5.1 Macroscopic fundamental diagrams

The *Macroscopic Fundamental Diagram* (MFD), also known as the *Network Fundamental Diagram* (NFD) is an attempt to uniquely capture the relationship between the number of vehicles in an area, referred to as accumulation, and the trip completion rate, i.e. vehicles leaving the network, within this area (Daganzo, 2007). Originally discussed in Godfrey (1969), this concept gained renewed attention in recent years, mainly in the context of control and Intelligent Transport Systems (ITS). The first control oriented applications aimed to regulate the number of vehicles entering an area via perimeter control (Geroliminis and Sun, 2011; Daganzo and Geroliminis, 2008; Geroliminis and Daganzo, 2008). The last few years have seen alternative uses, from taxi dispatch control strategies (Ramezani and Nourinejad, 2017) to MFD based assignment procedures (Knoop et al., 2016; Zhang et al., 2015; Knoop and Hoogendoorn, 2013). In the latter case, MFDs are used to abstract out – part of - the road network and replace it with a single functional relationship that treats the MFD area as a “reservoir” like entity. In this sense, the MFD acts as a network aggregation method.

Challenges with this approach are the fact that the functional relationship is very much topology dependent. Thus, it cannot be extracted without reverting to simulation or empirical data analysis (if at all). Knoop et al. (2015) also noticed that a homogenous layout reduces the scatter on the MFD, this means it only works reliably when the infrastructure characteristics, as well as the traffic states within the area, are relatively homogeneous. Also, the level of abstraction is such that it can only approximate the true interactions of travel flows to a limited extent making it, arguably, unattractive for planning purposes. Given our focus on methods that exhibit minimal information loss and guarantee consistency across different levels of detail, MFDs are not (yet) considered suitable.

7.6 Clustering

Aggregation methods can in many ways be thought of as a specific type of *clustering*. Clustering methods, like aggregation methods, group data points. Typically, a clustering procedure is either *unsupervised*, *semi-supervised*, or *supervised*. Supervised methods are sometimes, for example in machine learning, also referred to as *classification* based clustering.

Most of the discussed aggregation methods in transport are in fact classification based clustering methods.

Classification based approaches rely on prior knowledge to assign components into known classes. For example, using road types to dictate which links are removed, or abstracted out (Jeon et al., 2012; Chan, 1976; Goldman, 1966). Similar classification based methods are used to place or (dis)aggregate centroids (Chang et al., 2002; Mann, 2002; Khatib et al., 2001; Bovy and Jansen, 1983), or where and how many connectors should be included (Jafari et al., 2015; Benezech, 2011; Friedrich and Galster, 2009). These methods are considered to be supervised because the criteria for clustering are known a-priori, so the clustering procedure operates under supervision of these criteria. This in contrast to *semi-supervised* and *unsupervised learning* (Manning and Schütze, 1999) where only some, or even no, prior knowledge is used.

While the distinction between the types of clustering is agreed upon, no single definition of what clustering exactly stands for exists. Here, we adopt the formulation in Pfitzner et al. (2009), who state that clustering is: “*simply a process in which the members of a data set are divided into groups such that the members of each cluster, i.e. group, are sufficiently similar to infer they are of the same type and the members of the separate clusters are sufficiently different to infer they are of different types*”. The similarity between cluster members can refer to anything that relates to the application context. Figure 0.2 shows an example where two different similarity measures result in two different clusterings on the same dataset.

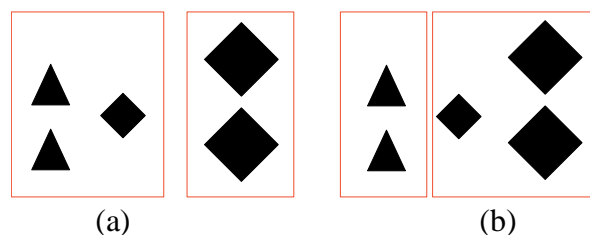


Figure 0.2: Example clustering based on: (a) shape size classification, (b) shape type classification.

The chosen similarity measure drives the clustering result. When using a categorical scale, as shown in Figure 0.2(b), solutions are generally easier to obtain and can be considered classification based. Finding appropriate clusterings when the measure of similarity is continuous, possibly dependent on multiple variables, and/or multiple data points, significantly complicates matters. In those circumstances clustering techniques beyond simple classification come into play.

7.6.1 Hierarchical and partitional clustering

Given that even comprehensive reviews on clustering techniques only succeed in discussing a small subset of the vast literature on the subject (Jain et al., 1999), we only introduce the reader to some of the core concepts of (non-classification based) clustering and, whenever possible, directly relate them to the transport domain.

A clustering technique is either hierarchical or it is not. *Hierarchical clustering* methods have a nested structure where each level closer to the root, i.e. cluster with all data points, merges

the clusters of the previous level until only a single partition containing all data points remains (Ward, 1963). In non-hierarchical methods such a tree like structure is absent and only a single level of partitioning results, hence the name *partitional clustering*. Only a few examples of (partial) hierarchical clustering in a traffic assignment context exist, such as the adaptive zoning approach in Hagen-Zanker and Jin (2015), or van Steijn (2016). This is likely due to the fact that in traffic assignment, traditionally, only a single zoning system or network is considered. One could argue that a hierarchical approach in constructing consistent representations in a multi-scale environment would be a natural fit because the relation between disaggregate zones and more aggregate zones is for example clearly defined. On the other hand, it imposes a rigid structure where original zone boundaries always remain. Based on our earlier discussion on MAUP we know that it is unlikely that, at a different level of detail, these boundaries remain the best choice. So, while a hierarchical zoning systems might have some benefits in a multi-scale context, we choose to focus on partitional clustering techniques to retain maximum flexibility. To maintain consistency between different granularities we utilise hard and soft constraints instead of a hierarchical zoning system.

A special type of partitional clustering worth mentioning is *spectral partitioning* (Alpert and Yao, 1995). Spectral partitioning is a graph based clustering technique that uses eigenvalues and eigenvectors to partition data points. It is considered attractive because it allows for a concise mathematical formulation. Also, it seems to be reasonably efficient in terms of computational cost which has led to a small literature on the subject in the transport domain, see for example, Ruddell and Raith (2013), or Bell et al. (2017). We do not consider this type of clustering in this work, although there are, in places, some similarities, which are pointed out when relevant.

7.6.2 Unsupervised Clustering techniques overview

Most traditional unsupervised partitional clustering techniques rely on *heuristics*. A heuristic implies that there exists an underlying objective that one tries to satisfy, which indeed is the case in these approaches, although it is not always explicitly formulated. Also, heuristic based procedures, by definition, cannot guarantee an optimal solution to this underlying objective. The best known example of a heuristic clustering technique (without an explicitly formulated objective) is the k -means algorithm (MacQueen, 1967). In k -means, a dataset is partitioned in k sets of arbitrary size. It requires an initial location for each of the k clusters, the algorithm then assigns each data point to the closest cluster. After each iteration, the average location of each cluster is updated based on the locations of its assigned data points until convergence is reached. It is an elegant and simple approach, but has two major drawbacks. First, results are strongly dependent on the initial locations. Second, the number of clusters needs to be specified beforehand and is fixed. The k -means algorithm is an example of *hard clustering*, where each data point is assigned to exactly one cluster. There also exist clustering algorithms that do not necessarily assign data points fully to one cluster, but utilise a membership probability instead. This partial membership approach is known as *fuzzy clustering*. A well-known example is found in the Expectation Maximisation (EM) algorithm which can be considered as the fuzzy equivalent of k -means (Kearns et al., 1997). We restrict ourselves to hard clustering techniques only, because fuzzy partitioning is generally incompatible with how traffic assignment methods operate. For a comprehensive overview of unsupervised hard partitioning clustering algorithms – among other things – we refer the reader to Gan et al. (2007).

7.6.3 Semi-supervised clustering techniques

Semi-supervised clustering techniques are a hybrid form of their unsupervised and supervised counterparts. This branch of methods differs from traditional unsupervised clustering techniques by incorporating some background knowledge into the method. This background knowledge can be included via different types of constraints, or additional distance based metrics (Basu et al., 2004). This is particularly useful in case contextual information is available, but this information is not sufficient to revert to a classification based approach. In these situations, one can construct an objective function subject to these constraints, in turn providing the opportunity to formulate the problem as a constrained optimisation problem. This is attractive because there exist many tools and methods to solve problems that are formulated in this particular way.

Wagstaff et al. (2001) were the first to introduce background knowledge into the k -means algorithm by proposing two types of constraints, a *must-link* constraint and a *cannot-link* constraint. Both constraints impose a limitation on the relation between two data points and are therefore known as *pair-wise* constraints, or *instance-level* constraints. Observe that, because of their pair-wise nature, these constraints can be constructed a-priori and serve as additional input to the clustering algorithm. Wagstaff et al. (2001) use their method for lane identification based on GPS data. Some adaptations of clustering methods using instance-level constraints can be found in for example Ruiz et al. (2009), or Klein et al. (2002).

In addition to instance-level constraints there also exist constraints that act upon a cluster rather than on two individual data points. In Joshi et al. (2012) this type of constraint is termed *cluster-level* constraint. The first to incorporate this type of constraint, as far as the author is aware, are Davidson and Ravi (2005), who impose a so called minimum-separation-constraint where they ensure that all data points in the cluster can no further be apart from any other data point in the cluster than a predefined value. Clearly, the restrictions on cluster-level constraints can be constructed a-priori. Instead, compliance with these constraints requires the construction of a cluster, which makes this type of constraint more complex (and computationally costly) compared to instance-level constraints. Figure 0.3 depicts the main categories of clustering techniques, constraint types that are associated with them, and if we consider them or not in this work.

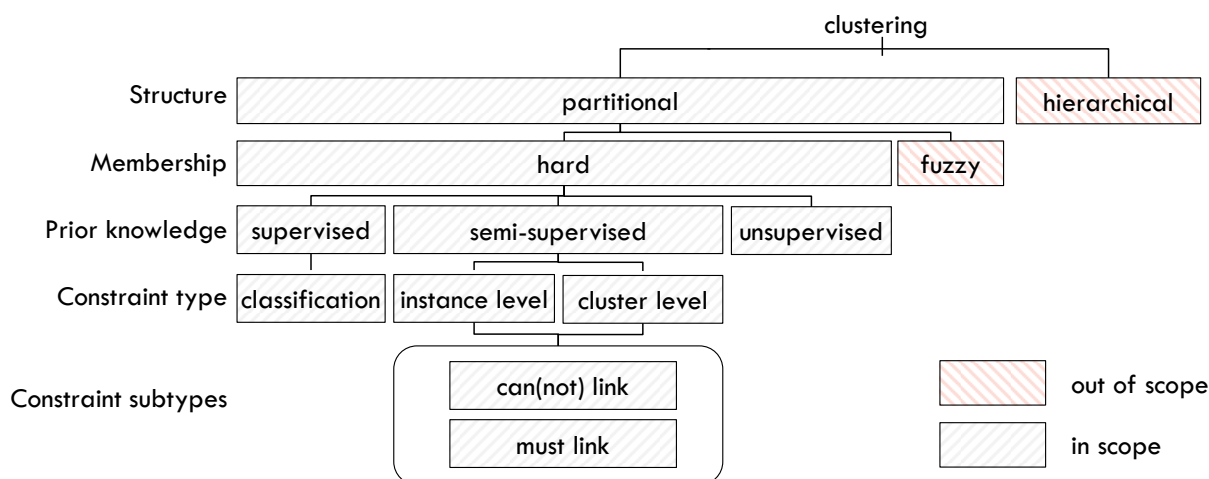


Figure 0.3: Clustering techniques and constraint type overview.

Spatial (semi-supervised) clustering is a specific type of clustering where data points are clustered based on geographical characteristics. One of the most well-known algorithms in this area is DBSCAN (Ester et al, 1996). It proposes to cluster points based on density and requires just one parameter which acts as the constraint to delineate clusters. A benefit of this approach is that it does not require a predetermined number of clusters, like k -means. Extensions to DBSCAN also exist, including versions that incorporate must-link and can(not)link constraints (Lelis and Sander, 2009; Ruiz et al., 2009). From a transport perspective however, considering only the density of (spatial) data points is somewhat limited. When clustering zones, we would like to consider multiple characteristics, for example, achieving both high intrazonal connectivity and a more or less equal number of trips across zones. Such a combination of objectives restricted by a number of constraints is therefore likely the best way to formulate our zoning problem.

7.6.4 Clustering and optimization

Solving constrained optimization problems either results in an optimal solution or not. The most basic optimal solution algorithms adopt brute-force and search the entire solution space to guarantee the best solution is found. A commonly adopted, and more sophisticated, method is found in *branch-and-bound* (Morrison et al., 2016), where the solution space is dynamically reduced by bounding it. Branch-and-bound approaches can be optimal or non-optimal depending on their design. Branch-and-bound algorithms are also suitable for cluster based problems, see Koontz et al. (1975) for an early example.

Most real world clustering applications deal with a solution space that is simply too large to solve optimally. This is where heuristic approaches are used, to provide a solution that is of acceptable quality, i.e. close to optimality. Numerous heuristic approaches exist and new approaches still emerge frequently. Besides simple greedy heuristics, the, arguably, most popular type of heuristic algorithms are *metaheuristics*. Metaheuristics propose a high-level solution approach that can be applied to a wide variety of optimisation problems. This is possible because they are so generally defined that they require little to no knowledge on the underlying problem that is being solved. For an introduction on metaheuristics see for example Voß (2000). To increase the effectiveness of metaheuristics in a particular application context, they are often adapted or combined (Puchinger and Raidl, 2005). Some well-known examples of metaheuristics are Genetic and evolutionary algorithms (Hruschka et al., 2009), simulated annealing (Aarts et al., 2005), tabu search (Gendreau and Potvin, 2005), or swarm intelligence approaches such as ant-colony optimisation (Dorigo et al., 2006).

In traffic assignment metaheuristics are mostly used to construct zoning systems, since these are highly non-linear complex combinatorial problems in their own right. Constructing zoning systems (instead of aggregating existing ones) typically adopt a land use perspective. Zoning systems in the context of land use sometimes drop the term TAZ or zone and refer to *regions* instead. For example, in Li et al. (2014) the so called p -compact regions problem is specified where the authors aim to create p regions out of a given number of smaller polygonal units. The resulting regions are then used in urban economic models which in turn incorporate a traffic assignment model. The authors define an objective function with multiple criteria and constraints. They use metaheuristics to find a near optimal solution. Analogous to other demand driven models in the literature, they consider socio-economic indicators in addition to shape

constraints (compactness measures) to determine their regions. Some other examples of (meta)heuristics employed to construct zoning systems can be found in Kim et al. (2016), Schockaert et al. (2011), Wei and Chai (2004), or Taillard (2003). We do emphasize however that, as long as it is possible to solve a problem optimally, one should try and avoid using heuristics because it is difficult to assess their solution quality.

7.7 Traffic assignment in a multi-scale environment

We make a distinction between *multi-scale traffic assignment procedures* and *multi-scale traffic assignment models*. The former allows for different granularities within a single assignment, i.e. network loading procedure. This means that some parts, i.e. spatial areas, of the simulation run at a different levels of detail than others. This type of simulation is sometimes also referred to as hybrid traffic assignment (Burghout et al., 2005; Casas et al., 2011). In such approaches it is possible to have “pockets” of microscopic simulation that are supplemented with coarser mesoscopic and/or macroscopic modelling techniques. This can alternatively be classified as a non-uniform aggregation approach.

Multi-scale traffic assignment models on the other hand, are models that operate separately from one another, having different networks, zoning systems, network loading and path choice components. The reference to multi-scale in this context mainly alludes to the fact that these models operate on overlapping spatial areas, are supposed to be able to exchange information, and somehow should yield comparable results. Our interest is focussed on this latter category, but literature on this topic is scarce. The few references that mention multi-scale modelling, either only consider multi-scale visualisation, where the underlying model is the same (Cheng et al., 2010), or provide a software design capable of hosting multiple models alongside each other (Chaker et al., 2010), but are lacking any consideration on how to ensure consistency between the different levels of granularity. Hence, as far as we are aware there exists no literature that either qualitatively or quantitatively investigates requirements for the successful design and/or operation of multiple traffic assignment models in a multi-scale setting.

7.8 Summary and Discussion

We can draw a number of conclusions from the existing literature on methods impacting on the representation of traffic assignment models and their inputs. First, the research is scattered across different fields (land use modelling, traffic assignment, operations research, GIS, machine learning) resulting in different terminology and more importantly, often only considering a subset of the data, components, or information that eventually is relevant to the final application. At the same time, research suggests that altering the representation of traffic assignment components should in fact be considered in an integrated fashion, mainly because choices regarding the granularity of one component impact on the remaining components.

Second, most existing (dis)aggregation methods in traffic assignment focus on a single component such as zoning, or network, or connectors. There exists research on connector costs without considering the zoning system, research on centroid placement without considering network design, or research on network design without considering the zoning, connectors, nor

centroids. These isolated approaches could well be the result from the traditional split between demand and supply side, both in research as well as in practice.

Third, there is a gap in the literature regarding the justification on the placement and number of connectors for a given zoning system, while at the same time it has been recognised that connectors (and their costs) significantly impact traffic assignment results.

Fourth, the literature on zoning systems and trip matrices considers socio-economic data, statistical information such as census boundaries and household travel survey results. This is demand side information. The literature does not seem to consider supply side information (travel times, congestion levels, path flows) in this context. This seems odd given the fact that the level of service of a network impacts travel choices and could - and arguably should - be considered when constructing these long term zoning systems and their related trip matrices.

Fifth, the last two decades gave rise to an increased interest in clustering methods. The similarities between aggregation and clustering provide opportunities to exploit these developments and bring them into the transport domain. Currently, in the context of traffic assignment, only a relatively small literature touches on this topic, mainly related to constructing zoning systems. In most cases, these approaches aim to cater for high level economic models rather than traffic assignment specifically. Therefore, there is room for improvement with respect to designing integrated approaches that considers supply side characteristics.

Based on these observations, we fill the existing gap in the literature by proposing novel methodology to address the identified issues of consistency and lack of integrated approaches when constructing traffic assignment model representations. As briefly mentioned in Chapter 1, we propose to incorporate supply side information, in the form of expected road usage, to not only improve the zoning system (and its demand), but at the same time use it to construct the network, the connector representation, estimate connector cost, and decide on centroid placement at the same time. We think that such a holistic approach is more natural than following the existing isolated methods. We claim our method is consistent based on the fact that all components are considered in unison and are constructed based on the same set of metrics.

In a multi-scale context, it is not only consistency on the *model inputs* that matters. The adopted traffic assignment *procedures* applied to the different granularities of model inputs need to be consistent as well. We therefore first assess the requirements for consistent traffic assignment procedures (Chapter 8) before constructing their inputs (Chapter 9, 10).

8 On traffic assignment consistency in a multi-scale environment

A traffic assignment model is always constructed in a way that is deemed “correct” by its operators. For example, the methodology proposed in Part II of this thesis can be considered correct in the context of quick-scan methods. However, when placing this same model in a multi-scale environment without any further consideration and together with other traffic assignment models, problems may arise. These problems mostly stem from inconsistencies in the underlying assumptions across the different models. By itself, adopting different assumptions need not be problematic, but in a multi-scale environment, where outputs of these different models can and will be compared, such inconsistencies are no longer acceptable because it compromises the credibility of the model results because it is no longer clear which model results can be trusted and which cannot.

Another practical issue that arises when setting up a multi-scale environment are conversions between different model types. Any new traffic assignment model within this environment is bound to adopt a different level of detail than already available models. Given that they operate on the same, or at least a similar, spatial area, the following situation frequently occurs; components, or even an entire model, that already exists is taken as a starting point. This model, or some of its components, are then copied and somehow modified to the required level of detail of the new model. More often than not, this process is driven by ad-hoc decisions, remains undocumented, and hardly ever involves standardised (automated) procedures. Even if a certain protocol is followed, this approach is often abandoned when a subsequent new model granularity needs to be constructed. Therefore, in addition to inconsistencies in the assumptions, all kinds of additional inconsistencies, in the process leading up to the final model representation, are introduced. Hence, the result accuracy in the newly constructed models can be compromised severely. Especially problematic is that the modeller, due to lack of information, is unable to pinpoint the cause of any discrepancies that he/she finds when comparing model results. In general, we should therefore be careful to adopt results obtained from such “derived” models.

In this chapter, we discuss how to choose traffic assignment procedure such that it is consistent with other traffic assignment procedures and formulate a number of (qualitative) criteria to ensure this. Methodology for the construction of appropriate traffic assignment model inputs is postponed to Chapters 9 and 10.

We first discuss some aspects of traffic assignment model conversions in Section 8.1. In Section 8.2 we classify traffic assignment procedures, by exposing all of their underlying simplifying assumptions and rank them accordingly based on their relative capability. We argue that only when all assumptions are made explicit, we can verify the consistency between traffic assignment procedures. In Section 8.3, we then propose three consistency criteria to perform this verification process. This is followed by a brief summary of the chapter in Section 8.4

8.1 Traffic assignment conversions in a multi-scale environment

Traditionally, strategic planning models cover large surface areas while adopting a relatively coarse level of detail. In practice they are therefore often used as a feeder model for more detailed, shorter term, and more localised planning models. This conversion from a coarse model to a more detailed model is a significant challenge. It is not uncommon that the majority of project budgets for these tactical and operational models is spent on resolving issues that stem from this conversion. Among the most notable challenges are the conversion of aggregate, and often static, demand matrices to more disaggregate and time varying versions, as well as constructing disaggregated zoning systems and their demand-supply interface. Similarly, there needs to be consensus on what type of network detail is to be added, where it is added, and why. One of the reasons this is difficult to execute is because the original simplifications that led to the original strategic planning model are no longer known and the original disaggregate data is no longer available either. This then, results in the situation depicted in Figure 8.1(a): two models both with different assumptions, different inputs, and different procedures, yet applied in the same multi-scale environment. As a result there is little that can be said about the comparative modelling power of the “derived” model and sometimes even on the original model, let alone any insight on the level of consistency between the models.

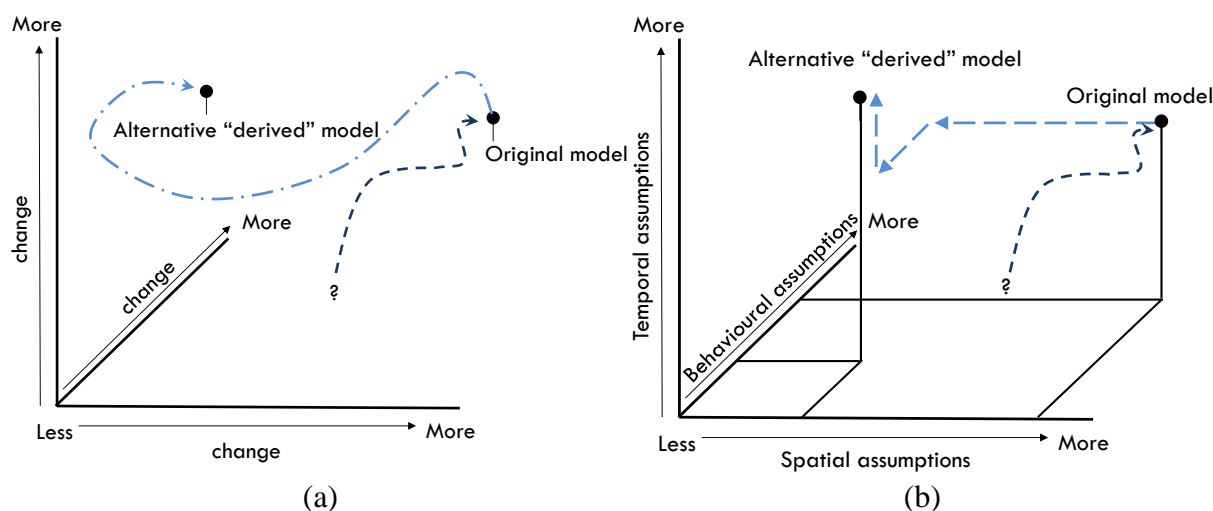


Figure 8.1: Model conversion under (a) implicit assumptions, (b) explicit assumptions.

Our first step in avoiding this situation relies on making all relevant modelling assumptions explicit. We do so based on the spatial, temporal, and behavioural assumption dimensions discussed in Chapter 3. When we know what changes in assumptions led to the derived modelling procedure, we can at least assess the relative modelling power of the “derived” model in Figure 8.1(b). In this example, we would find out that the alternative model is more capable in two dimensions (less simplifying assumptions made), while it is considered less capable in one dimension (more simplifying assumptions made). Once we list the possible model assumptions explicitly, we use them to objectively assess the procedural consistency between models in Section 8.3.

8.2 Traffic assignment assumptions revisited

In Chapter 3 we discussed the traffic assignment literature by assessing the different model types per assumption dimension. We now refine each dimension by splitting it in subcategories, to the point that all assumptions become explicit. To prevent discussing a complete taxonomy of traffic assignment models, we only describe the subcategories and model features in terms of their relative capabilities. More general and in-depth discussions can be found in the excellent review by Wageningen-Kessels et al.(2014), regarding microscopic and mesoscopic model characteristics, or Bliemer et al. (2017), regarding macroscopic characteristics.

8.2.1 Spatial assumption subcategories and capabilities

On the link level, vehicle based traffic flow interactions are considered the most capable, see Figure 8.2. Microscopic and mesoscopic models fall in this category. Fundamental diagram based approaches, i.e. macroscopic models based on average flow rates, are less capable due to their more aggregate nature.

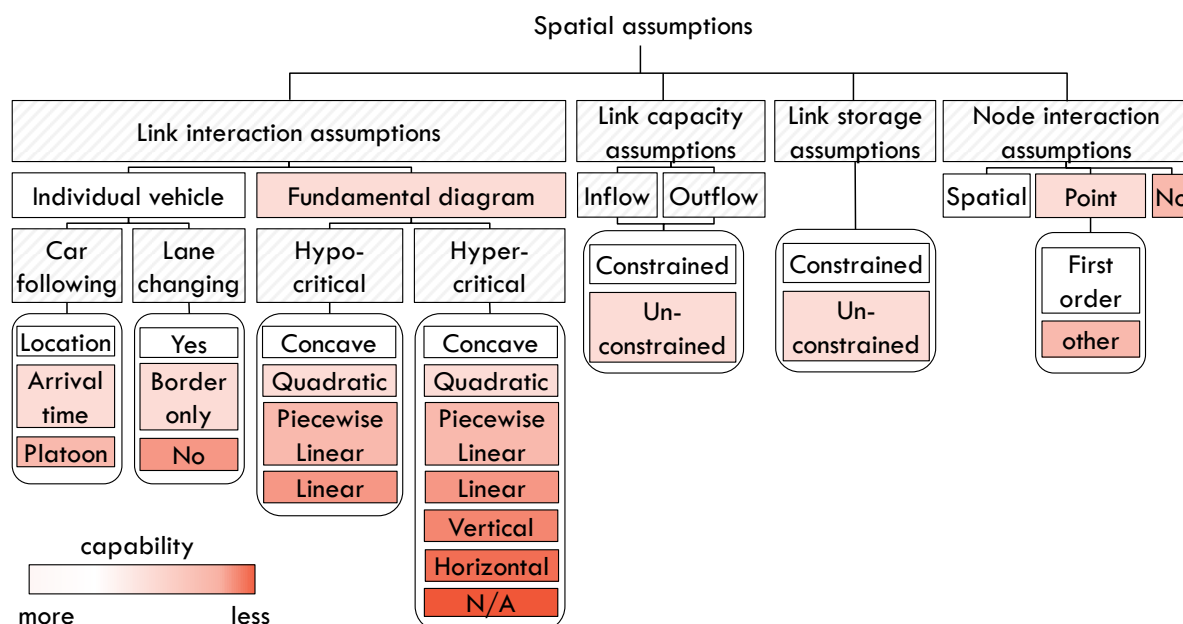


Figure 8.2: Spatial assumptions in traffic assignment by subcategory, in decreasing order of capability.

Within the vehicle based approaches, the assumptions regarding longitudinal interaction (car following) and lateral interaction (lane changing) determine the relative model capability. In the case of flow based interactions, the shape of the adopted fundamental diagram drives the model capability. We distinguish between the shape for the hypocritical and hypercritical regime. The most capable approach here is when both the hypo and hypercritical regimes can be modelled through any concave function. On the other hand, when adopting for example a BPR link performance function, one makes the simplifying assumptions that the link is nor capacity constrained, nor storage constrained, and there exists no hypercritical regime on the fundamental diagram. This then leads to a significantly less capable model from a spatial perspective. Link capacity constraints in reality are enforced across the entire road section, there do however exist models that only impose a capacity constraint at the upstream, or

downstream border, or neither of the two in case of the BPR function. We therefore explicitly allow for this by considering the location where capacity constraints are enforced (if any).

Finally, spatial interactions on intersections lead to the potential inclusion of a node model. This node model can for example be point based such as the class of first order node models described in Tampère et al. (2011), or it relates to some other model type. Micro and meso approaches typically model node interactions spatially, instead of abstracting it out to an analytical point model and are therefore considered more capable.

8.2.2 Temporal assumption subcategories and capabilities

Temporal assumptions leading to the static, semi-dynamic, and dynamic models discussed in Chapter 3, are in fact the result of a combinations of assumptions. The assumptions relate to the *propagation speeds of traffic states, travel demand propagation, and residual traffic transfer*¹¹. An overview of each temporal subcategory is provided in Figure 8.3, ordered by their relative capability.

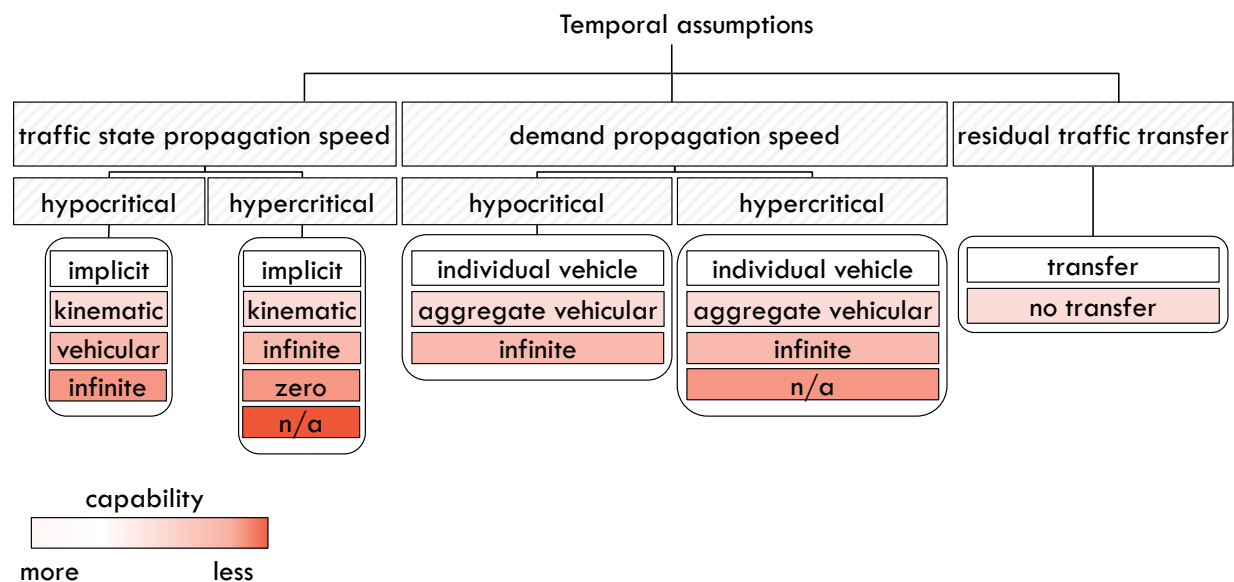


Figure 8.3: Temporal assumptions in traffic assignment by subcategory, in decreasing order of capability.

In saturated conditions, traffic states determine the speed with which queues build up (if any), while in unsaturated conditions they dictate how upstream conditions propagate downstream. Travel demand propagation determines the speed of the vehicles themselves, which is in general different to the speed with which traffic states propagate. Finally, assumptions regarding residual traffic transfer determine if traffic, still in the network at the end of the simulation, is either assumed to disappear, or is retained in case of subsequent modelling periods. Again, vehicle based approaches are considered most capable because traffic state and demand propagation is by definition consistent since the former implicitly follows from the latter. Macroscopic dynamic first order approaches that are fully compliant with kinematic wave theory can be considered most capable within the limitations of aggregate traffic flow

¹¹ In Bliemer et al. (2017) only macroscopic approaches were considered. We therefore extended and altered original definitions. Traffic state propagation speed was originally termed wave speed, while travel demand propagation speed originally was referred to as vehicle propagation speed.

modelling. Some macroscopic approaches however are less capable than others, especially so when they do not comply with kinematic wave theory. This occurs when they propagate queues upstream too fast (infinite speed), keep them in one place (zero speed, resulting in vertical queues), or do not model them at all (not available), leading to a much simplified hypercritical traffic state propagation. Similarly, from a hypocritical perspective, some models base their traffic state propagation on vehicular speeds that are inconsistent with the underlying fundamental diagram and therefore their capability is compromised as well (e.g., such models cannot describe fanning effects that occur when flow increases).

8.2.3 Behavioural assumption subcategories and capabilities

The different behavioural model types mainly emerge from assumptions that relate to how travel time is constructed and interpreted by travellers. The decision making process can be broken down in full rational approaches, or boundedly rational (Di and Liu, 2016). The latter is considered more capable yet harder to model. SUE and DUE are examples of fully rational approaches, where SUE is considered more capable because there is support for imperfect information on the side of the decision maker, while DUE approaches only consider perfect information. The actual travel times can be constructed based on experience (most capable), to predictive, or simply instantaneous. The latter is least capable because it the decision maker simplifies its behaviour by assuming the current travel time is in fact the true travel time. The classification for the behavioural assumption dimension from a capability perspective is provided in Figure 8.4 and follows the original classification as described in Bliemer et al. (2017).

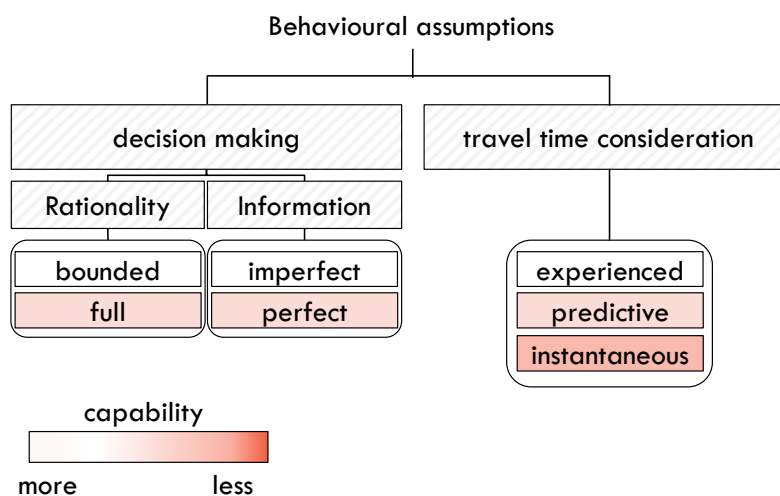


Figure 8.4: Behavioural assumptions in traffic assignment by subcategory, in decreasing order of capability.

8.3 Traffic assignment consistency in a multi-scale environment

We utilise the explicit and capability based classification of the previous sections to objectively assess differences between (existing) traffic assignment model procedures, from these differences we draw conclusions on how consistent these models are and if they are suitable to be adopted within the same multi-scale environment.

Recall that we postpone the construction of consistent model inputs to Chapters 9 and 10. However, we do have to discuss the properties that can ensure this consistency in the same way as we do for model procedures. As shown in Figure 8.5, all traffic assignment model inputs can be considered spatial in nature. Then, naturally, aggregating model inputs leads to less capable models, while disaggregating model inputs leads to more capable models. The question that needs to be answered is, what underlying assumptions leading to this (dis)aggregation are acceptable such that we maintain consistency between these different model granularities.

| component | | subcomponent | Spatial | Temporal | Behavioural |
|-----------------------------------|---------------------------|--------------------------------|---------|----------|-------------|
| Traffic assignment representation | supply side | path representation | ✓ | | |
| | | network representation | ✓ | | |
| | demand-supply interface | connector representation | ✓ | | |
| | | zonal representation | ✓ | | |
| | demand side | trip demand representation | | ✓ | |
| | | Path choice representation | | ✓ | ✓ |
| | demand-supply interaction | network loading representation | ✓ | ✓ | ✓ |

Figure 8.5: Categorisation tree for representation methods and their assumption dimension.

We argue that the following requirements need to be satisfied in order to guarantee inter-model consistency: (i) *directional consistency*, (ii) *source consistency*, and (iii) *abstraction consistency*.

First, let us discuss *directional consistency*. Consider the earlier example of Figure 8.1(b) again, a less capable model was found with respect to one assumption dimension, while the other assumptions led to increased model capability. We argue that consistency between models can only be achieved when all changes in assumptions that are imposed affect the model capability in the same direction. The underlying rationale here is that, at the very least, it must be clear if differences in model results are in fact an improvement over the original results or not. We can only do this when each of the altered assumptions results in a more capable model (or all yield a less capable model depending on the perspective). If not, then it is no longer certain to what extent the changes are in fact compromising or enhancing results compared to the original model. Following this line of reasoning, the example in Figure 8.1(b) is in fact not an acceptable model conversion because it is not directionally consistent.

Second, we impose a condition of *source consistency*. We argue that a single point of reference should be adopted. Also, this reference source model, in our view, should be the most capable traffic assignment model that one would consider within the multi-scale setting. From a procedural point of view, this means that we relate everything to the single most capable model under consideration. For traffic assignment model inputs this is slightly more challenging. In practice, the data collection and spatial granularity of a network are often already tailored to the desired level of detail of the new traffic assignment model. In case of a strategic planning model, this means that most data is collected at a coarse spatial aggregation level. This becomes a problem when trying to construct a more detailed model based on this aggregate model; additional detail is “invented” from the aggregate sources. This problem is amplified when

multiple disaggregate models are derived from the aggregate base model, due to lack of methodology, resulting in aforementioned inconsistencies. To avoid this, all model inputs should be created from the same data source(s) as much as possible. In other words, a strategic model remains coarser than a tactical model, but its original data sources should be retained and collected at a disaggregate level as much as possible. This data can then be reused for any other model that is constructed within the same multi-scale environment. That way, the same (automated) methodology driven aggregation procedure, one of which is discussed in Chapters 9 and 10, can be applied on the inputs such that it is always possible to construct consistent traffic assignment model representations at any level of detail. Conceptually, this results in the situation depicted in Figure 8.6. This approach does require a fundamental change in how traffic assignment models are created. The conversion oriented paradigm is abandoned in favour of a source model design principle.

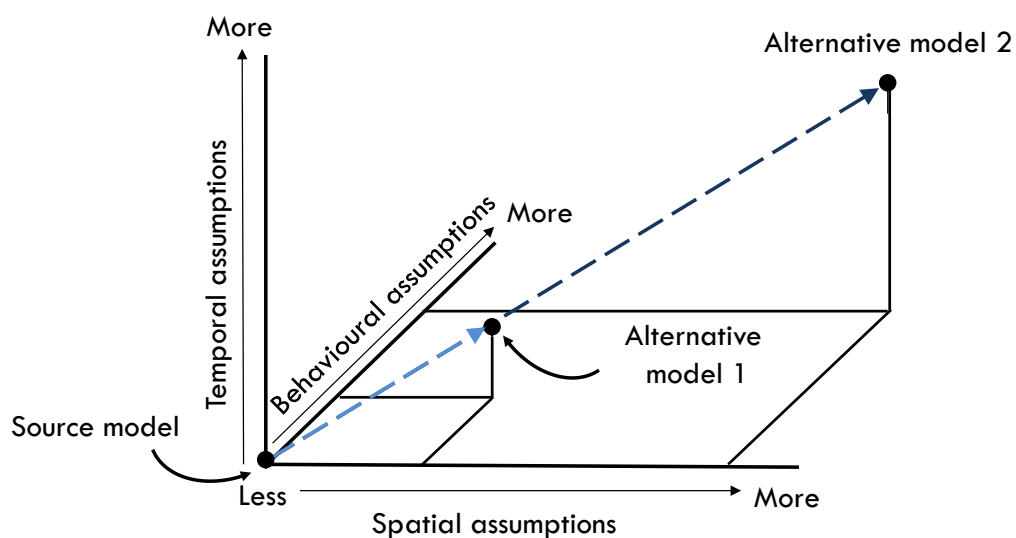


Figure 8.6: Model construction in a multiscale environment under explicit, consistent simplifying assumptions utilising a single point of origin.

Finally, the third condition that needs to be considered is termed *abstraction consistency*. Whenever simplifying assumptions in model procedures lead to abandoning particular model features altogether, there is no issue with consistency because the capability of the model has changed in a single direction (less capable). This can be regarded as the procedural equivalent of extraction (see Section 7.1). Yet, some simplifying assumptions result in a more complex change in model capability. For example, let's assume two identical models except for their spatial interaction assumptions; one is a macroscopic model and the other a microscopic model. However, as discussed in Section 3.2, car following models can be converted into a macroscopic representation by adopting the fundamental diagram consistent with the car following behaviour, under steady state conditions. As a result, it is possible to convert a model from a microscopic to a macroscopic representation in a consistent fashion. Such a change in assumptions, from a procedural perspective, should be regarded as an abstraction method rather than an extraction method because the macroscopic behaviour replaces the individual car following behaviour. In this case, abstraction consistency is satisfied as long as the fundamental diagram corresponds to the steady state car following model of the microscopic model (or a less capable version of it). Similar considerations apply, for example, when replacing spatial node models with point based alternatives.

8.4 Summary

In this chapter we classified traffic assignment models based on the capabilities of their underlying assumptions. We adapted the framework of Bliemer et al. (2017) to do so. The spatial categories of the original framework, extended with the microscopic and mesoscopic modelling paradigm, led to a capability based classification of the subsequent subcategorization of modelling features and assumptions. This assessment led to three novel qualitative conditions that, we argue, need to be satisfied in order to guarantee consistency between traffic assignment models residing in the same multi-scale environment. These conditions are referred to as: (i) *source model consistency*, (ii) *directional consistency*, and (iii) *abstraction consistency*. When any of the conditions are violated, the differences between model outputs are compromised in the sense that it becomes difficult to justify which result is better.

As long as the underlying assumptions of traffic assignment procedures are known, verification of the above conditions for existing traffic assignment models is relatively straightforward. The difficulty lies in obtaining or constructing appropriately disaggregate model inputs that can be used as the source model in a multi-scale setting. Moreover, methodology is required to scale these source model inputs holistically, i.e. in an integrated manner, and consistently, following the reasoning in Chapter 7 and complying with the conditions laid out in this chapter. These two observations are the focal point for Chapters 9 and 10.

9 Methodology for consistent traffic assignment model inputs

In Chapter 8 the procedural side of achieving consistency between traffic assignment models in a multi-scale environment is discussed. In this chapter we explore the same topic, only from the perspective of the traffic assignment model inputs. When combining the methodology presented in this chapter with the conditions and classification regarding traffic assignment procedures, a fully integrated approach results, allowing practitioners to achieve consistency across traffic assignment model representations.

The proposed methodology is compliant with the conditions specified in the previous chapter, meaning that we construct our model inputs such that we adhere to: directional consistency, source consistency, and abstraction consistency. To do so, we first discuss how our methodology fits the general representation framework in Section 9.1. Then, a general disaggregation-aggregation framework, specific to constructing consistent traffic assignment model inputs in a multi-scale setting is proposed in Section 9.2. This is followed by methodology for four of the five steps proposed within this framework. These steps include: (i) construction of the source model inputs in Section 9.3, (ii) source model assignment procedure in Section 9.4, (iii) supply input representation in Section 9.5, (iv) and demand-supply interface representation in Section 9.6. The last step in the framework, the demand input representation, is discussed in a separate chapter, namely Chapter 10. We conclude with a summary of this chapter in Section 9.7.

9.1 Traffic assignment model input representation

The focus of this chapter is solely on traffic assignment model inputs. We make no assumptions on the adopted traffic assignment procedure once the inputs are made available. We do however encourage that this procedure is selected based on the reasoning and conditions discussed in Chapter 8.

Recall that in Chapter 2 we introduced implicit (rule based) representation function $\Xi_\gamma(\cdot)$. Since we only consider traffic assignment model inputs here, Equations (2.4) and (2.5) simplify to:

$$\mathcal{M}^* = \Xi_\gamma(\mathcal{M}), \quad \text{with } \mathcal{M} = (\mathbf{A}, \mathbf{D}, \mathbf{Z}) \text{ and } \mathcal{M}^* = (\mathbf{A}^*, \mathbf{D}^*, \mathbf{Z}^*) \quad (9.1)$$

Path choice $\Psi(\cdot)$ and network loading $\Phi(\cdot)$ are not considered, neither is a predefined path set \mathbf{P} . We are only concerned with altering the physical road network, centroids, and connectors via \mathbf{A} , the trip demand through \mathbf{D} , and the zoning structure \mathbf{Z} . The overall problem of Equation (2.6) can therefore be written as:

$$\min_{\gamma} \left(\begin{array}{l} \varepsilon(\mathcal{M}, \Xi_{\gamma}(\mathcal{M})) \\ \zeta(\mathcal{M}, \Xi_{\gamma}(\mathcal{M})) \end{array} \right), \quad \text{with } \mathcal{M} = (\mathbf{A}, \mathbf{D}, \mathbf{Z}) \quad (9.2)$$

We explicitly relate the information loss function $\varepsilon(\cdot)$ and magnitude of scaling function $\zeta(\cdot)$ to our proposed objectives when discussing the zonal representation in Chapter 10.

9.2 Framework for consistent traffic assignment model input design

In Chapter 7 we concluded that constructing a zoning system, regardless of its level of sophistication, will always remain just one of an infinite number of possible representations. We also found that existing methodology in this context is mainly driven by demand side considerations; meaning that the supply side, as well as demand-supply interactions, in the construction of the zoning system are largely ignored. This omission does not by definition mean supply side information should be considered. It is of course entirely possible that it is unnecessary to do so. However, we do not subscribe to this view. Moreover, existing methods regarding the construction of zoning systems implicitly acknowledge the importance of supply side characteristics by incorporating - rather crude - measures trying to capture the effects of this interaction. They do so, for example, by delineating zones based on road types, or, even more indirectly; zone delineation by bodies of water, or other natural structures. These proxies, among other things, aim to minimise the variance in travel times within a zone without estimating its actual effect. Given that travel time is one of the most important outputs of a (strategic) traffic assignment model, ideally, the zoning structure should not compromise its design by settling for these proxy based solutions.

Of course, when supply side information would be embedded in the construction of the zoning system, the zoning system becomes conditional on the results of traffic assignment. Yet, the results of traffic assignment are also conditional on the zoning system. This would therefore introduce a mutual dependency and requires some kind of iterative procedure that terminates upon convergence, see Figure 9.1.

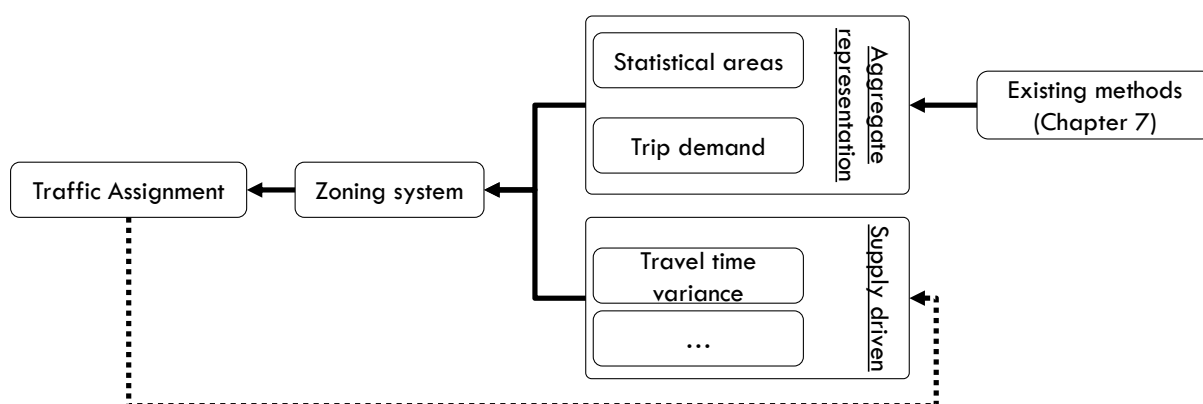


Figure 9.1: Zoning system and traffic assignments' mutual dependency.

Given the existing complexity of the demand driven zonal design methods, the additional complexity of embedding supply side information in any such procedures, and the fact that our

objective is to provide methodology that is of demonstrable practical use, we explicitly choose to only explore a non-iterative approach instead. This has three benefits compensating for the lack of a “convergence based” representation procedure. First, any of the methods proposed can be directly embedded in existing modelling processes as a supplementary procedure. Second, it obviates the need to make a choice on what existing zoning method we choose to embed our method in. Therefore, differences compared to the original zoning can be objectively assessed and are not compromised by the effect of other, existing – and possibly conflicting or unknown – metrics stemming from the combination of the various methods. Third, it always remains a possibility to introduce a feedback loop afterwards if one so desires.

9.2.1 Travel time variance through expected road usage

To incorporate supply side information, let us consider a zone that, in terms of its demand side data, has been constructed in a perfectly homogeneous way. Traditionally, the following happens; the zone is accepted and replaced with a centroid, the local roads (based on road type) are replaced with “representative” connectors and assignment is performed. Now consider the situation that we have additional information that shows that some paths from within the zone to a particular edge of the zone exhibit travel times that are severely affected by congested local roads, while the other local roads experience no congestion. Also note that all these local roads are no longer present in the accepted zone due to being abstracted out for connectors with a fixed – low – cost. In this situation, travel time accuracy is compromised by aggregating out the high level of travel time variance in reaching the physical road network. We therefore argue that in addition to demand side considerations that lead to a “homogenous” zone, also the *internal travel time stability*, i.e. low intrazonal travel time variance, of a zone should explicitly be taken into account. Low variance in disaggregate travel times within zones can only be achieved when we can guarantee that roads being replaced by connectors are in fact uncongested roads. Consequently, this information, in absence of empirical data, can only be obtained by performing some kind of traffic assignment procedure, yielding information on the level of *expected road usage*. This metric allows us to distinguish between uncongested and potentially congested roads, subsequently leading to the identification of areas with low variance in travel times. We refer to these areas as *zone components*, we discuss zone components in more detail in Section 9.6.1. Zone components serve as the foundation for constructing the final zoning system.

9.2.2 Integrated approach

Besides the zoning system, we also consider the other traffic assignment model inputs. Earlier, we found that while there exists literature on the placement of centroids (Friedrich and Galster, 2009; Chang et al, 2002; Bovy and Jansen, 1983), the construction of connectors (Jafari et al., 2015; Qian and Zhang, 2012; Benezech, 2011) and the granularity of the transport network (Jafari and Boyles, 2016; Bovy and Jansen, 1983), they often consider these components in isolation. Following the conclusions in Chapter 7, we propose to consider the creation of the zoning system, centroids, connectors and the (aggregate) network in an integrated fashion. Interestingly, the notion of expected road usage turns out to be equally suitable in driving the desired granularity of the aggregate network, which in turn allows for a consistent, accurate, and justifiable estimation of connector costs and their placement. We also argue that there is no need for - a spatial location of - centroids anymore because we remove the dependency between the cost of a connector and the location of this manufactured construct that is a

centroid. An important practical benefit of considering all traffic assignment model inputs in unison is that the construction of the interface between demand and supply becomes more natural, is consistent, seamless and reproducible. This in contrast to current practice where, all too often, one still relies on ad-hoc decisions by the modeller at hand.

9.2.3 Conceptual Framework

We assume that the following inputs are available and given: the original travel demand and zoning, i.e. statistical areas and trip matrices, as well as the complete disaggregate road infrastructure supply (e.g. extracted from OpenStreetMaps, networks underpinning route navigation systems, or obtained from governmental bodies). The methods acting on these inputs are implemented as part of a general framework for the construction of traffic assignment model inputs under supply side constraints. This framework is depicted in Figure 9.2.

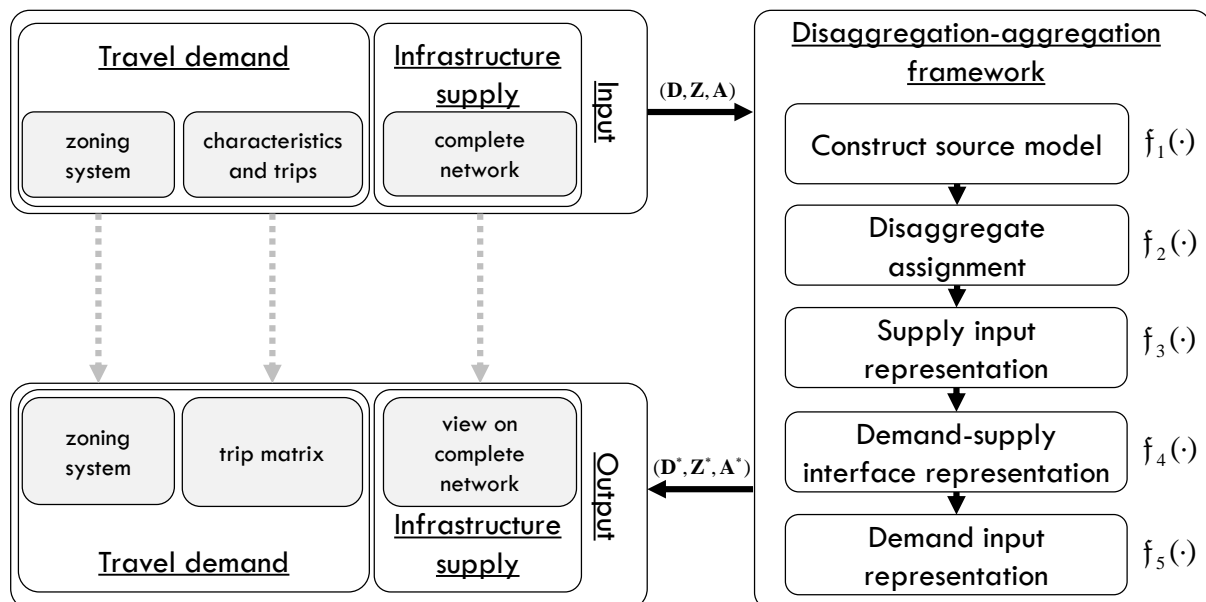


Figure 9.2: Disaggregation-aggregation framework for holistic approach to supply side zone, centroid, connector and network representation.

The framework comprises five components described by five implicit functions $f_1(\cdot), f_2(\cdot), \dots, f_5(\cdot)$. First, the original zoning and its travel demand is disaggregated as performed by function $f_1(\cdot)$. The result can be considered the starting point for constructing the appropriate granularity of the traffic assignment model inputs. Note that this disaggregation step is only necessary as long as complete individual or household trip data is unavailable. In present day, this is nearly always the case, but maybe in the future this might be different. Clearly, once this data becomes available this first step can and should be skipped. The inputs available after completing $f_1(\cdot)$ represent the source model inputs as discussed in Chapter 8. We explicitly account for this step within the framework itself because, while it is typically possible to obtain a network at its finest level of detail, this is much harder when it comes to demand side data. The reasons this data is so hard to obtain can be explained by the sheer volume of data that is involved, as well as other factors such as privacy related issues. Next, function $f_2(\cdot)$ applies - some form of - a disaggregate assignment resulting in information regarding the level of expected road usage. This result is then used in $f_3(\cdot)$ to decide which

network elements can be abstracted out and which ones should be kept. Special care should be taken to ensure network connectivity is guaranteed. In $f_4(\cdot)$ the links that are marked to be abstracted out serve as the foundation for the new zoning structure, as well as the starting point for an accurate estimation for connector costs and their placement. In $f_5(\cdot)$ the final zoning system is constructed utilising clustering methodology that groups the basic zoning components to the extent deemed desirable by the modeller utilising a constrained optimisation problem formulation.

To demonstrate the suitability of this framework, we illustrate each of the steps with a particular implementation. We acknowledge that each of the steps' proposed implementations is just one of many possible approaches. For each of the methods adopted in the various steps, we opted for approaches that require little calibration in order to make them generally applicable. Also, while the methodology itself is novel, we, in places, adopt well known modelling paradigms to highlight that good results can be achieved not only in an academic setting, but that our methods can directly be adopted in practice if one so desires.

Similar to Part II of this thesis, the steps $f_{1-4}(\cdot)$ are discussed and formulated such that there is no need to specify a separate algorithm, one can simply follow the formalised versions of each step, discussed in the remainder of this chapter, to achieve the desired results. The zonal representation in $f_5(\cdot)$ however, is formulated as an optimisation problem and does require an explicit solution scheme. As mentioned, we postpone both the formulation of this step, as well as the accompanying solution scheme, to Chapter 10.

9.2.4 Consistency revisited

In Chapter 8 we argued that constructing traffic assignment models in a multi-scale environment requires (i) directional consistency, (ii) source consistency, and (iii) abstraction consistency. We argue that the framework in the previous section is compliant with these requirements.

- (i) **Source consistency:** After the completion of $f_1(\cdot)$, we consider the disaggregate model inputs as the source model inputs. The subsequent aggregation procedures therefore rely on the exact same source data guaranteeing the source consistency condition.
- (ii) **Directional consistency:** All model inputs, from the perspective of the source model, are subject only to aggregation. Hence, all modifications are in the same direction, namely spatially aggregated, satisfying the directional consistency condition.
- (iii) **Abstraction consistency:** The assumption that drives the aggregation procedure(s) is that we can identify areas of stable internal travel times and utilise them as the building blocks for zones in the aggregate representation. Abstraction only takes place when replacing the infrastructure, within the area with stable internal travel times, with connectors with a fixed cost. The construction of these costs, like the zones, and like the network, all rely on the exact same metric of expected road

usage. Therefore the process of abstraction is argued to be consistent across the considered components.

9.2.5 Notational conventions

Formalising the steps in the disaggregation-aggregation framework requires denoting the same types of data at varying levels of detail. We start with the original model inputs, which are disaggregated to source model inputs, then we construct the data that serves as the starting point for the zonal clustering (via so called zone components), followed by aggregated clustering results. To prevent an excessive amount of notation, we “flavour” existing notation by their granularity, an example – with respect to zones $z \in \{1, \dots, Z\}$ - is outlined in Table 9.1. This table only serves as an illustration of the notation, the related concepts are discussed at a later stage in this chapter.

Table 9.1: Different notational flavours depending on the granularity of the data.

| Variable | Meaning | Properties |
|-----------|--|-------------------------------------|
| Z | Original number of zones $z \in \{1, \dots, Z\}$. | |
| \bar{Z} | Disaggregate number of zones, with $\bar{z} \in \{1, \dots, \bar{Z}\}$. | $\bar{Z} \geq Z$ |
| \vec{Z} | Number of zone components with $\vec{z} \in \{1, \dots, \vec{Z}\}$. | $\vec{Z} \geq \bar{Z} \geq Z$ |
| \hat{Z} | Number of zone component clusters with $\hat{z} \in \{1, \dots, \hat{Z}\}$. | $\vec{Z} \geq \bar{Z} \geq \hat{Z}$ |
| Z^* | Number of final zones with $z^* \in \{1, \dots, Z^*\}$. | $\vec{Z} \geq \bar{Z} \geq Z^*$ |

9.3 Step 1: Constructing the source model

To construct the source model we adopt a link centric approach where we assume that all trip demand originates from the physical links in the network. To formalise the underlying disaggregation method, we first discuss the relation between the original zoning system and its underlying infrastructure, followed by the construction of the disaggregate demand based on this infrastructure.

9.3.1 Original zoning system and its infrastructure

Each physical link in the network is attributed to exactly one zone. Initially, we are provided with the original zones $z \in \{1, \dots, Z\}$, all of which represent a geographical area. We denote the relation between physical links and their respective zones via membership indicator matrix $\mathbf{A}^z \in \mathbb{F}_2^{N \times N}$. Note that we exclude connector links in this respect because they are not physical links. An example is provided in Figure 9.3. Observe that both link a_3 , i.e. node pair (n_2, n_3) , and link a_4 , i.e. node pair (n_3, n_2) , cross a zone boundary, in such cases the link is attributed to the zone containing the majority of the link infrastructure, here, this means both links belong to zone z_2 .

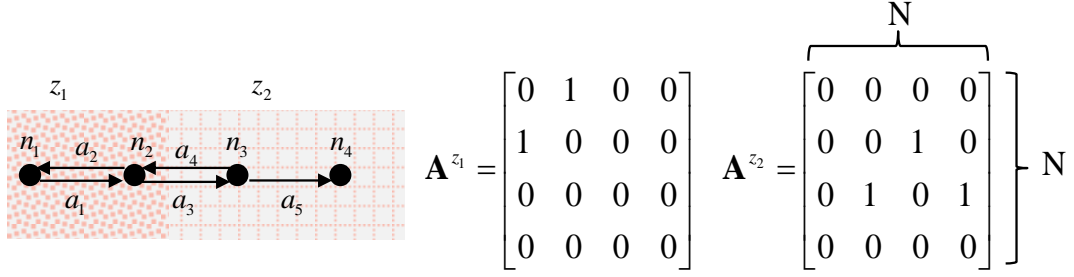


Figure 9.3: Links attributed to zones based on location.

Outgoing and incoming links of nodes can be partitioned similarly through $\mathbf{A}^{z+} \in \mathbb{F}_2^{N \times A}$, $\mathbf{A}^{z-} \in \mathbb{F}_2^{N \times A}$, respectively, with:

$$\mathbf{A}^z = \mathbf{A}^{z+} (\mathbf{A}^{z-})^T. \quad (9.3)$$

From this link-to-zone mapping, we also extract the relation between nodes and the original zones in a similar fashion. Typically, a node n is either *internal* or *external* to an original zone z via:

$$N_{zn} = \begin{cases} 1, & \text{if } A_n^z \cdot \mathbf{1} \geq 1, \\ 0, & \text{otherwise,} \end{cases} \quad (9.4)$$

with $\mathbf{N} \in \mathbb{F}_2^{Z \times N}$ where $N_{zn} = 1$ means that node n has one or more incoming or outgoing links attributed to zone z , hence it is considered internal to z . Otherwise, it is considered external, see Figure 9.4 for an example

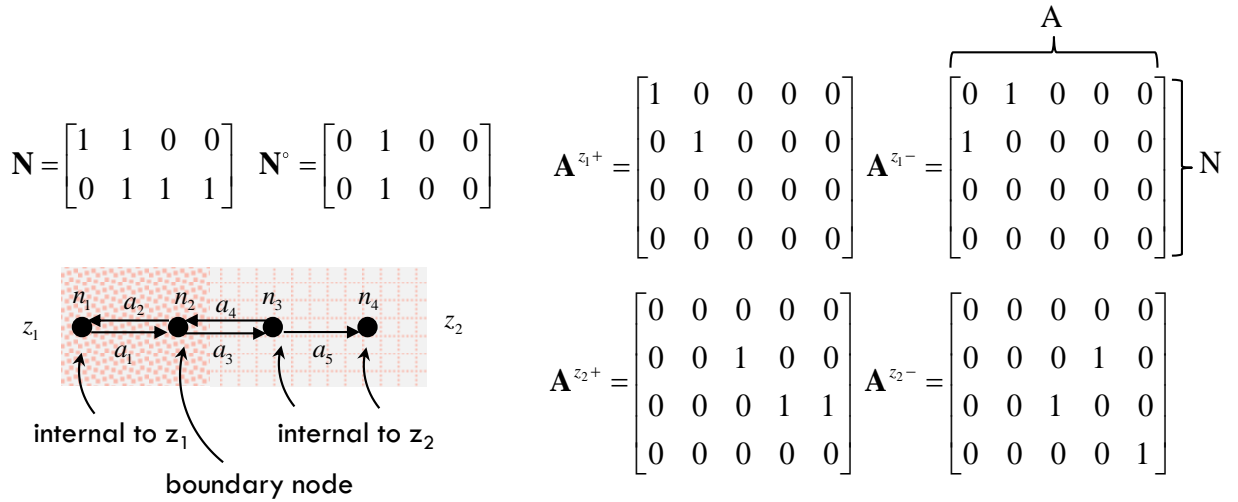


Figure 9.4: Node and link membership of original zones.

Observe that node n_2 is in fact internal to both z_1 and z_2 . Such nodes are referred to as *boundary nodes* because they allow for a transition from one original zone to the other via its connected links. We formalise boundary nodes through $\mathbf{N}^\circ \in \mathbb{F}_2^{Z \times N}$, where:

$$N_{zn}^\circ = \begin{cases} N_{zn}, & \text{if } \mathbf{N}_{\bullet n} \cdot \mathbf{1} > 1, \\ 0, & \text{otherwise.} \end{cases} \quad (9.5)$$

Because boundary nodes represent the physical transition point from one (original) zone to the other, they play an important role in the construction of the – to be created - refined zoning system, hence our explicit identification of these locations, see also Section 9.6.1.

9.3.2 Constructing the source model demand

Observe that by placing the demand on the links, we redistribute existing zonal demand to their internal links, therefore, based on the terminology discussed in Chapter 7, this disaggregation step is classified as an expansion method - the inverse of abstraction - rather than an insertion method.

Because travellers live on streets represented by links in the model, the origins and destinations of trips reside on links as well. To simplify, we assume that trips go from the upstream node of their origin link to the upstream node of their destination link. Hence, the more outgoing links a node has (i.e. the more streets it represents), the higher the demand is likely to be. To accommodate any type of such node weighting based on this “bundling” of link demand, we propose a generic node weight, or priority, based on said node characteristics, which we capture via weight vector $\mathbf{w}^z \in \mathbb{R}_+^{N \times 1}$. This weight vector is zone specific, such that only nodes residing within zone z can have non-zero weights.

Consider the example in Figure 9.3 again, where, if we base our weight on the number of outgoing links on each node within the respective zone, we would find that $\mathbf{w}^1 = [1, 1, 0, 0]$ and $\mathbf{w}^2 = [0, 1, 2, 0]$. Observe that for boundary node n_2 we should only consider the outgoing links attributed to the zone in question, so both zones receive a weight of 1 for this node. Because boundary nodes have non-zero weights in multiple zones, we must take measures to prevent losing any information when constructing the disaggregate node-to-node demand matrix $\bar{\mathbf{D}} \in \mathbb{R}_+^{N \times N}$. We do so by constructing *od-pair specific* disaggregate node-to-node demand matrices first, denoted $\bar{\mathbf{D}}^{zz'} \in \mathbb{R}_+^{N \times N}$, via:

$$\bar{D}_{mn'}^{zz'} = \left(\frac{w_n^z}{\mathbf{1}^T \mathbf{w}^z} \cdot \frac{w_{n'}^{z'}}{\mathbf{1}^T \mathbf{w}^{z'}} \right) D_{zn'}^{zz'}, \quad n, n' \in \{1, \dots, N\}, \quad (9.6)$$

where the appropriate node weight is applied for the point of departure (from zone z) as well as the point of arrival (from zone z'), relative to the total weight of the departing, arriving zone, respectively. The combined weight ratio is then multiplied with the original zone demand to yield the disaggregate node based trip demand. The final disaggregate *interzonal* demand matrix is then simply obtained by summing over all od-pair based disaggregate node-to-node matrices:

$$\bar{D}_{mn'} = \sum_{z=1}^Z \sum_{z'=1}^Z \bar{D}_{mn'}^{zz'} \quad (9.7)$$

In case there exists *intrazonal* demand in the original zoning, this is automatically distributed across the nodes internal to the zone via their weights. We have not taken any precaution

against having non-zero intrazonal demand for disaggregate node pair (n, n) , but one of course can do so if desired.

As a numerical example, consider the network in Figure 9.3 again and let us assume an original demand between z_1 and z_2 of $D_{1,2} = 100$. Zone z_1 has a total weight of 2 since there are two nodes with a single outgoing link within z_1 . For node 1, we find that $\bar{D}_{1,2}^{1,2} = \frac{1}{2}(\frac{1}{3} \cdot 100) = 16\frac{2}{3}$, $\bar{D}_{1,3}^{1,2} = \frac{1}{2}(\frac{2}{3} \cdot 100) = 33\frac{1}{3}$, and $\bar{D}_{1,4}^{1,2} = \frac{1}{2}(\frac{0}{3} \cdot 100) = 0$. Since, the boundary node portion residing in z_1 has the same weight as node 1, it also gains the same number of trips as node 1, with sums up to the total demand of 100. The number of disaggregate zones is denoted by \bar{Z} , which equates to the number of nodes in the original network i.e. $\bar{Z} = N$.

9.3.3 Constructing node weights

Constructing node weights w^z can be done in many ways. In the previous section we provided an example that assumed a weight based on the number of outgoing links of each node. Clearly, many other possibilities exist. The simplest being applying a uniform weight across all nodes. Ideally however, one would like to consider more sophisticated approaches considering the variation in population density in zones, specific land use characteristics etc. Any such approaches are compatible with this proposed generic weighting and it depends on the application or case study what weighting is considered appropriate.

9.4 Step 2: Source model assignment

In our pursuit of refining the original zones such that they exhibit low travel time variance as well as determining which links to retain based on the desired granularity, information on the network's expected road usage is needed. The only way to get this insight on a network wide scale (in the absence of empirical data) is to perform some form of traffic assignment.

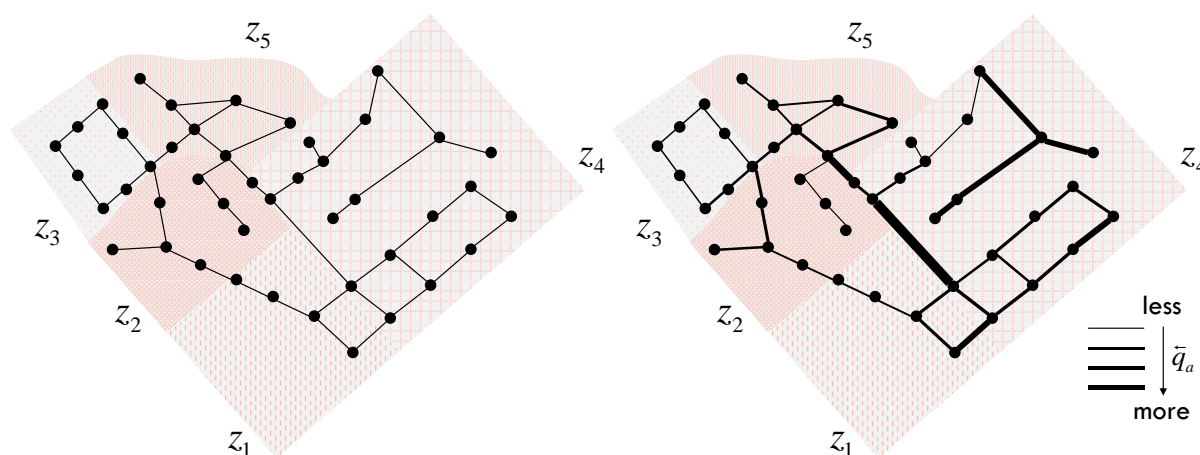


Figure 9.5: (a) Hypothetical 5 zone network, (b) example link flows after disaggregate node-to-node AON assignment.

While we could try to obtain very accurate estimates of densities, flows, and speeds, we argue this is not necessary. Instead we only make a distinction between roads without (virtually) any delay and roads with some or significant delay, based on our chosen metric of expected road usage. We argue this suffices because once we establish a link exhibits delay, it can no longer

be trusted to not exhibit significant travel time variance, especially if demand starts to vary. In that case it should not be considered as a candidate to abstract out. Further, it is far easier to identify links that have potential for exhibiting delay than determine their actual delay, since the former allows one to adopt an (expected) lower bound rather than trying to get an exact number. Following this line of reasoning we choose to apply All-Or-Nothing (AON) assignment, where we utilise Dijkstra's shortest path algorithm to find the single shortest paths between each disaggregate node pair. We denote the cost of the node-to-node shortest paths via $\bar{\mathbf{H}} \in \mathbb{R}_+^{N \times N}$. The related disaggregate paths are denoted by $\bar{\mathbf{P}} \in \mathbb{F}_2^{\bar{P} \times A}$, with $\bar{p} \in \{1, \dots, \bar{P}\}$. Analogous to outgoing and incoming links of a node, we define disaggregate paths departing/arriving from a disaggregate zone via $\bar{\mathbf{P}}^+ \in \mathbb{F}_2^{\bar{Z} \times \bar{P}}$, $\bar{\mathbf{P}}^- \in \mathbb{F}_2^{\bar{Z} \times \bar{P}}$, respectively. The disaggregate path flow vector $\bar{\mathbf{f}} \in \mathbb{R}_+^{\bar{P} \times 1}$ is then found via:

$$\bar{\mathbf{f}} = ((\bar{\mathbf{P}}^+)^T \bar{\mathbf{D}} \bar{\mathbf{P}}^-) \mathbf{1}, \quad (9.8)$$

or, if a non-matrix based notation is preferred, can alternatively be acquired via:

$$\bar{f}_{\bar{p}} = \sum_{\bar{z}=1}^{\bar{Z}} \sum_{\bar{z}'=1}^{\bar{Z}} \bar{P}_{\bar{p}\bar{z}}^+ \bar{P}_{\bar{p}\bar{z}'}^- \bar{D}_{\bar{z}\bar{z}'}, \quad \bar{p} \in \{1, \dots, \bar{P}\}. \quad (9.9)$$

Assigning the disaggregate path flows $\bar{\mathbf{f}}$ to the network yields disaggregate link flow vector $\bar{\mathbf{q}} \in \mathbb{R}_+^{A \times 1}$ via:

$$\bar{\mathbf{q}} = \bar{\mathbf{P}}^T \bar{\mathbf{f}}. \quad (9.10)$$

Note that we do not concern ourselves with the interpretation of congested path or link costs, because we do not equilibrate this assignment. The link flow rates obtained via Equation (9.10) are used as an indicator for the link classification that drives the supply input representation, see also Figure 9.5(b).

Of course, adopting AON puts a limitation on the accuracy of the result and, if one would consider equilibrating the disaggregate assignment, a more accurate estimate of the actual road usage can be obtained. In its current form two types of error might occur that are not accounted for: (i) links that labelled as ‘‘high variance’’ in AON are in fact not high variance in an equilibrium result, (ii) links that are not labelled as high variance, are in fact congested in an equilibrium result. Both errors are the result of the lack of redistribution effects based on the traveller's experience on its path. That said, even under this crudest of approaches, results indicate a notable improvement over the original result (see Chapter 10), suggesting that when one would have the computational power to adopt a more sophisticated approach (and above mentioned errors are likely to be reduced if they existed in the first place) an even better result is possible. It is left for future research to compare our current AON approach with an alternative equilibrium assignment. The reason for this is mainly a pragmatic one; when the number of disaggregate zones \bar{Z} becomes high, the number of paths that one needs to consider grows exponentially, even when one considers only a single path per zone pair, i.e. $\bar{P} = \bar{Z} \cdot (\bar{Z} - 1)$. For example, the extremely small hypothetical network in Figure 9.5(a), already has 1980 disaggregate single paths resulting from 45 disaggregate zones. So, for most real world

networks, an AON approach is likely the only realistic approach to obtain this estimate considering the computational constraints involved.

It should be noted that although we create a single shortest path between all nodes, this is not related nor directly compatible with the well-known concept of *betweenness centrality* often used in graph theory, which counts the number of shortest paths passing through a node. In our case, each path still carries a weight, based on its flow, so counting the number of paths is not that meaningful when identifying high variance links/nodes. If such a measure were to be adopted, one should take this into account when disaggregating the demand into, let's say, single trips, only then such a measure would be representative. It could be of interest to explore such an alternative scheme, but again, this is left for future research.

9.5 Step 3: Supply input representation

In Part II we found that the computational cost of assignment is mostly determined by the number of paths, which in turn is dictated by the number of zones in the network. We also found that having an aggregated, or simplified, network also reduces computational cost, but this has not nearly as much impact as aggregating the zonal structure. Hence, in determining the granularity of the network it is always, in our view, more important to be able to capture spatial traffic flow interactions than to maximise network aggregation, simply because there is less to gain in the latter than there is to lose in the former. We apply this same principle when deciding on what links to retain and what links to abstract out. We only abstract out links that we consider to be truly local roads that are virtually guaranteed to exhibit no delay. These roads therefore exhibit stable travel times under all circumstances. All other links are considered as *keep links*. To identify the keep links, we specify keep link v/c ratio threshold κ^{\min} , reflecting the minimum road usage required for a link to be retained. Conversely, links that drop below this threshold are expected to remain undelayed. Any node with at least a single keep link adjacent to it, regardless of its direction is also retained and is termed *keep node*. The retained keep infrastructure is denoted by node-to-link indicator matrix $\mathcal{K} \in \mathbb{F}_2^{N \times A}$ such that:

$$\mathcal{K}_{na} = \begin{cases} \mathbf{A}_{na}^+ \parallel \mathbf{A}_{na}^-, & \text{if } \frac{\bar{q}_a}{q_a^{\max}} \geq \kappa^{\min}, \\ 0, & \text{otherwise.} \end{cases} \quad (9.11)$$

The observant reader might have noticed that Equation (9.11) is similar to the delay link classification in Part II, but in contrast to Part I, $\kappa^{\min} \ll \beta^{\min}$ because we require a high level of certainty regarding a link being a truly uncongested road. Figure 9.6 illustrates the concept of constructing a keep network following our binary link/node classification given our earlier hypothetical network and some given κ^{\min} .

In the extreme case all infrastructure in an original zone is marked to be kept, i.e. all links are above the threshold, an infeasible situation occurs since there is no mechanism to redistribute this demand to other original zones. However, this, in our experience, only occurs due to errors in the modelling of the base model's infrastructure. Typically, this only happens in CBD/inner city areas, and the modeller has not included all exits and entrances of (underground) parking.

By including these dominant sources of trips in such areas, the above described situation is not only avoided, but the method automatically places these trips in the correct locations. Hence, we emphasize that it is critical to include all infrastructure, including (mass) parking facilities, in the construction of the base model.

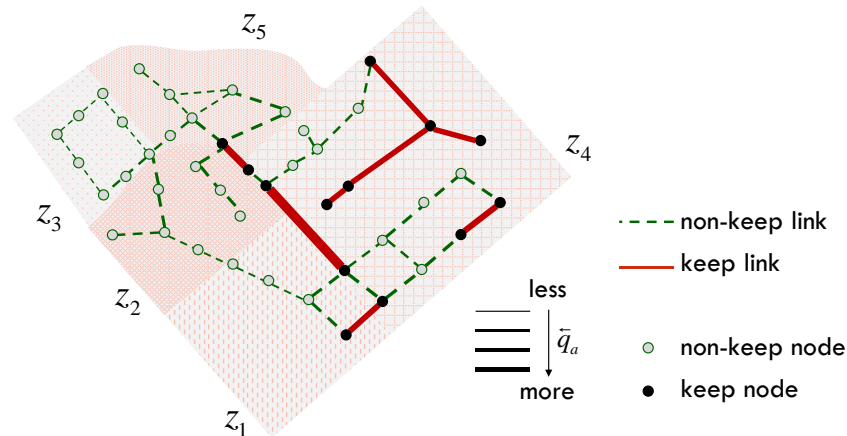


Figure 9.6: Binary link classification based on some expected road usage threshold κ^{\min} .

9.5.1 Network and zonal connectivity of paths

If we were to apply a traditional extraction method and remove all non-keep links from the original network, the situation depicted in Figure 9.7 results. It highlights why Chan (1976) classified extraction approaches as undesirable; in this particular case it leads to loss of network and path connectivity.

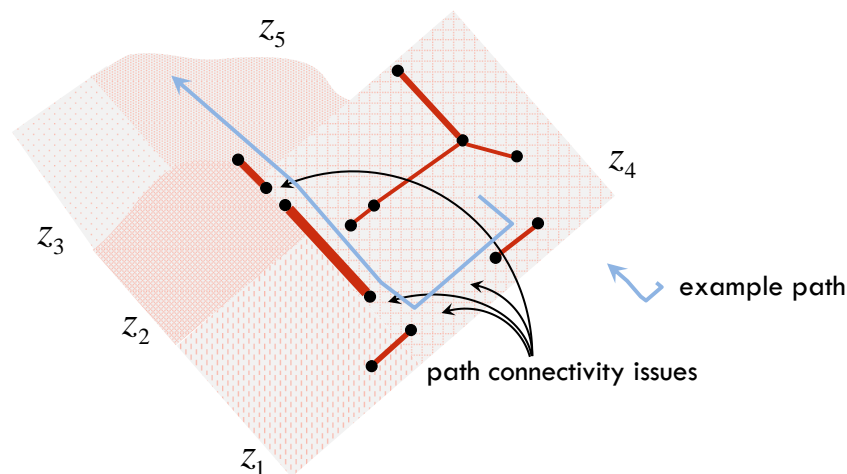


Figure 9.7: Connectivity violations under network extraction.

To avoid loss of connectivity we propose an approach where we rely on paths being decomposed into three separate sections: a *departure section*, an *arrival section*, and a *main section*. This split is similar to the three-way path split proposed in Benezech (2011), but improves upon this method in a number of ways. Benezech predefined a number of so called anchor points (Leurent et al., 2011). These anchor points are specific to each original zone and are used to mark a transition between path sections. In our approach, we do not force path sections to comply with the underlying original zoning system, which makes it more flexible. Second, we do not require a predefined number nor any predefined locations of anchor points.

Instead of relying on such rather arbitrary choices, our demarcation automatically follows from the expected road usage obtained earlier. Third, Benezech used these anchor points mainly to search for paths - in almost identical fashion as originally proposed by Raadsen et al. (2009). These paths were then used to estimate connector costs, while we use path sections to guarantee network connectivity instead.

Consider the path traversing the links depicted in Figure 9.8(a). The original link classification would break path connectivity if we were to remove the non-keep links. We avoid this by first identifying the three aforementioned path sections. The departure section starts at the path's origin and ends at, but does not include, the first keep link that is encountered. Similarly, the arrival section starts at, but does not include, the last keep link that is encountered on the path and ends at the path's destination. The main section comprises all links in between the departure and arrival section. Any non-keep links on a path's main section cannot be considered as truly local roads, because they are now used for "through" traffic. Roads are argued to be only truly local when they are solely utilised to access the physical road network for the "main" part of the trip (or egress after the main section). Hence, the underutilised links on main sections of trips should therefore be retained as well, see Figure 9.8(b). These additionally retained links are referred to as *connectivity-keep links* because they guarantee path connectivity.

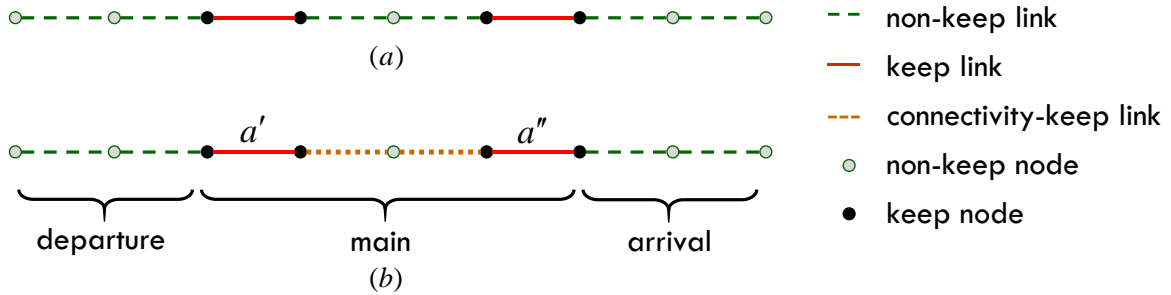


Figure 9.8: Example path under link extraction, (b) the same path utilising connectivity keep approach.

It is of course possible that the first/last link of a path is already marked as a keep link, then the departure/arrival path section is simply skipped. The inclusion of connectivity-keep links in our keep network formulation requires us to modify Equation (9.11) and replace it with:

$$\mathcal{K}_{na} = \begin{cases} A_{na}^+ \parallel A_{na}^-, & \text{if } \frac{\bar{q}_a}{q_a^{\max}} \geq \kappa^{\min}, \\ A_{na}^+ \parallel A_{na}^-, & \text{else if } \exists (\bar{p}, a', a'') : \eta_{a'}^{a\bar{p}} \eta_{a''}^{a'\bar{p}} = 1 ; \frac{\bar{q}_{a'}}{q_{a'}^{\max}} \geq \kappa^{\min} \text{ and } \frac{\bar{q}_{a''}}{q_{a''}^{\max}} \geq \kappa^{\min}, \\ 0 & \text{otherwise,} \end{cases} \quad (9.12)$$

with $a, a', a'' \in \{1, \dots, A\}, n \in \{1, \dots, N\}$, and $\bar{p} \in \{1, \dots, \bar{P}\}$. This can be interpreted as follows; link a is a connectivity keep link as long as there exists a disaggregate path \bar{p} passing through a , with an upstream and downstream link a', a'' , respectively, and both these links are marked as keep link (via $\eta_{a'}^{a\bar{p}} = 1$, and $\eta_{a''}^{a'\bar{p}} = 1$). Applying this approach to the earlier example in Figure 9.6, we obtain the supply representation as depicted in Figure 9.9.

The connectivity keep links ensure path connectivity on their main sections, i.e. on the physical network, but they do not solve the issue of path connectivity on the departure and arrival sections. This problem, of guaranteeing a path's access to its departing and arriving zone, is part of the demand-supply interface representation discussed in Section 9.6, because it involves the way paths, and therefore trip demand, enters or leaves the network.

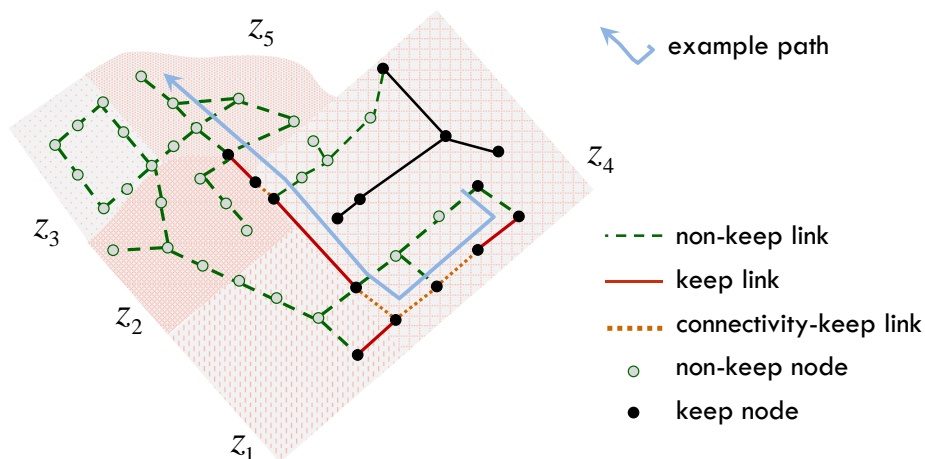


Figure 9.9: Additional connectivity-keep links for example path under given link classification.

9.5.2 Impact of road usage on disaggregate demand

Depending on how the initial node demand weights \mathbf{w}^z were established, there might be some additional insight available based on the link classification that allows us to improve upon our initial weights. Clearly, if the node weights are already based on detailed land use information or even disaggregate household trip data (unlikely today, but maybe in the future) we retain these weights. However, in case the original weighting adopted a relatively basic approach, such as for example, a uniform distribution, or the number of outgoing links of the node there might be some benefit in considering the following adaptation.

The link classification identifies what links are more prone to being congested. Often – but not always - such links include main arterials, motorways and other “road only” infrastructure that neither produces nor attracts travel demand and only serves as a thoroughfare. Clearly, it would be better to exclude these roads from carrying any demand. Also, smaller roads that are classified as keep links might require some reconsideration regarding their original demand weight. We argue that such potentially congested roads are more likely to not allow on-street parking and at the same time are less favourable as a residential location. The exception to this would be big apartment buildings or businesses with dedicated parking. Hence, the original basic weighting could benefit by setting the node weights of keep nodes to zero. Note that in order to gain the most from this adaptation, it is important to explicitly account for the inclusion of public parking garages, and possibly the location of underground parking facilities in major residential apartment buildings and/or businesses/shopping malls. This leads to the altered, and final, node weights $\mathbf{w}^{*z} \in \mathbb{R}_+^{N \times 1}$ via:

$$w_n^{*z} = \begin{cases} 0, & \text{if } \mathcal{K}_n \cdot \mathbf{1} \geq 1, \\ w_n^z, & \text{otherwise.} \end{cases} \quad (9.13)$$

These weights then also lead to an updated formulation for the disaggregate demand estimates as well with:

$$\bar{D}_{nn'}^{*zz'} = \left(\frac{w_n^{*z}}{\mathbf{1}^T \mathbf{w}^{*z}} \cdot \frac{w_{n'}^{*z'}}{\mathbf{1}^T \mathbf{w}^{*z'}} \right) D_{zz'}, \quad n, n' \in \{1, \dots, N\}, \quad (9.14)$$

and:

$$\bar{D}_{nn'}^* = \sum_{z=1}^Z \sum_{z'=1}^Z \bar{D}_{nn'}^{*zz'} \quad (9.15)$$

with $\bar{\mathbf{D}}^* \in \mathbb{R}_+^{N \times N}$. Of course, updated disaggregate demands would lead to changes in the assignment results in Step 2, which in turn would lead to a (slightly) different link classification and therefore, again, different weights $\bar{\mathbf{w}}^{*z}$. Therefore, when we would choose to set $\bar{\mathbf{D}} = \bar{\mathbf{D}}^*$ and revert back to Step 2 an iterative procedure follows that continues as long as $\bar{\mathbf{D}} \neq \bar{\mathbf{D}}^*$. This however, becomes a very time consuming procedure given the current disaggregate state of the model. Also, a preliminary investigation into such an iterative scheme suggested that the changes in keep network \mathcal{K} caused by performing additional iterations are small, at least beyond the initial construction of $\bar{\mathbf{D}}^*$. We therefore, update the weights (and disaggregate demand) only once and do not pursue further updates via subsequent iterations.

9.6 Step 4: Demand-supply interface representation

The construction of the demand-supply interface, i.e. connectors and centroids is closely related to the zoning structure. Both the zoning system, as well as the demand-supply interface draw from the same basic building blocks to determine their final representation. These building blocks are referred to as *zone components*. They are the smallest non-divisible geographic entity that is used to construct the new zoning system.

9.6.1 Zone components

We construct the zone components based on both the link classification obtained in the previous steps as well as the boundary nodes that delineated the original zones. Together they determine the infrastructure that is attributed to each zone component. Informally, a zone component is the largest subnetwork that can be constructed without including any keep links, nor crossing a boundary node. Figure 9.10(b) shows the zone components that are present in the example network given our earlier hypothetical link classification of Figure 9.10(a). The purpose of boundary nodes is to delineate zone components at the original zone boundaries. These boundaries represent differences in land use, socio-demographics, and other useful demand side information that should not be ignored. Note that boundary nodes that are also keep nodes already delineate zone components based on their keep node status. Therefore, only the boundary nodes that are classified as non-keep nodes actively contribute to incorporating the original demand side delineation result. This then results in subdividing the original zoning

structure in smaller areas that are delineated by keep links or infrastructure based disconnectivity within the original zone. In this particular example, it has for example led to the subdivision of original zone z_2 in two zone components \bar{z}_2 and \bar{z}_3 , respectively.

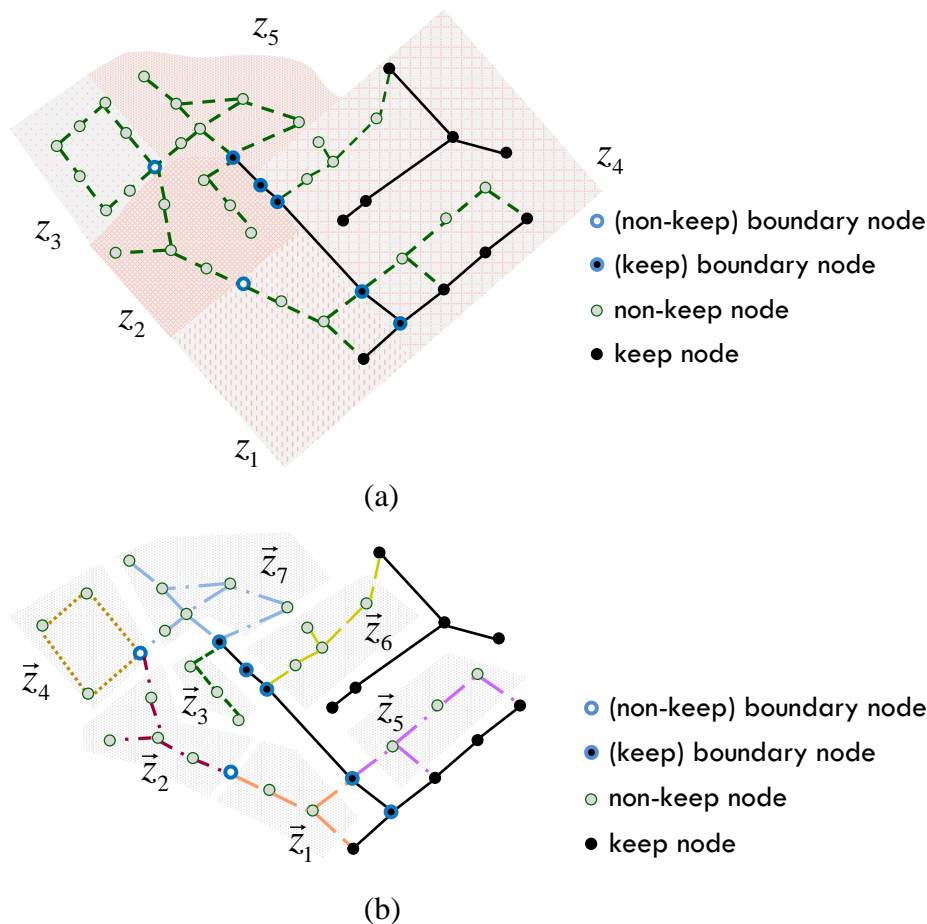


Figure 9.10: (a) link classification and (non-keep) boundary nodes, (b) zone components delineated by original zoning system and keep network.

We choose zone components as our basic building block for two reasons. The first reason is because it guarantees internal travel time stability for each zone component, i.e. a high level of certainty that all links internal to the zone component are uncongested under the adopted demand scheme. The second reason is because it takes the original zoning structure into account, hereby implicitly accounting for any possible – demand side – dissimilarities between original zones that might be overlooked when only considering travel times. A typical example of demand side dissimilarities that need to be taken into account would be the situation when a residential zone borders on a commercial zone. In the morning peak commuters would travel from the residential zone to the commercial zone. It is undesirable to merge the two zones because it results in a significant shift from interzonal to intrazonal trips, compromising the modelling power of the final outcome.

Identifying all zone components is straightforward and closely relates to the identification of connected components in a graph. A connected component, in graph theory, is a subgraph where any two vertices within the connected component can be connected via a path, while at the same time, replacing any of these vertices with a vertex not part of the subgraph does not

allow for a connection via a path. As discussed in Hopcroft and Tarjan (1973), connected components can be found through a simple depth-first search where one takes any edge as a starting point. Then adjacent edges are recursively added until no more adjacent edges can be found. The resulting subgraph is a connected component. The subgraph's edges are then removed from the original graph and the procedure is repeated, starting with any of the remaining edges in the reduced original graph.

We employ a slightly adapted version of this algorithm to make it suitable for identifying zone components. Two modifications are required: (i) the only adjacent edges accessible from boundary nodes are edges that are attributed to the same original zone as the current edge. (ii) Edges that are marked as keep link are not considered. Figure 9.11 shows the view on the example network on which the proposed algorithm operates, we leave it to the reader to observe that this leads to the seven zone components depicted in Figure 9.10(b).

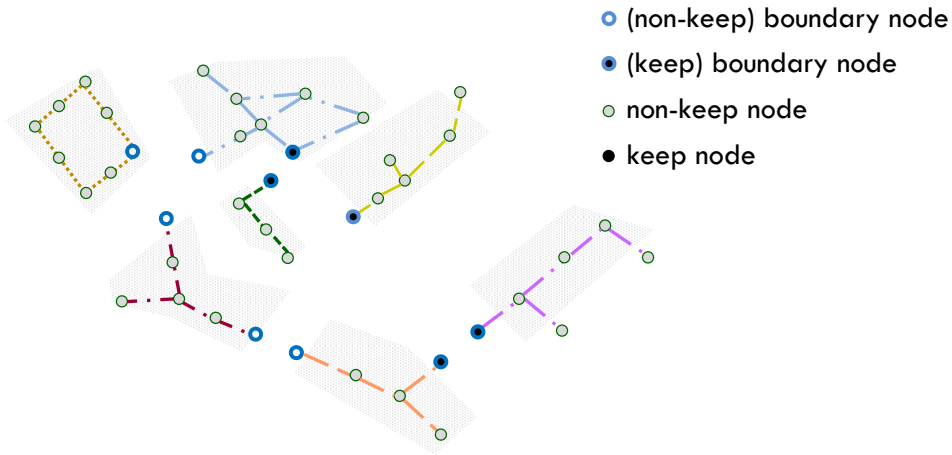


Figure 9.11: Identifying connected components in modified view on example network.

The number of zone components found by utilising this algorithm is denoted by \vec{Z} , with each zone component denoted $\vec{z} \in \{1, \dots, \vec{Z}\}$. We point out that $\vec{Z} \geq Z$ due to the inclusion of boundary node based delineation. Analogous to the original zoning system, we denote links internal to a zone component by $\vec{\mathbf{A}}^{\vec{z}} \in \mathbb{F}_2^{N \times N}$, and nodes internal to zone components by $\vec{\mathbf{N}} \in \mathbb{F}_2^{\vec{Z} \times N}$, which we obtain via:

$$\vec{N}_{\vec{z}n} = \begin{cases} 1, & \text{if } \vec{\mathbf{A}}_{n\bullet}^{\vec{z}} \geq 1, \\ 0, & \text{otherwise.} \end{cases} \quad (9.16)$$

with $\vec{z} \in \{1, \dots, \vec{Z}\}, n \in \{1, \dots, N\}$. The found zone components represent the most disaggregate zoning system our method is capable of constructing. It depends on the coarseness of the original zoning system, as well as on the original travel demand what this zonal representation looks like. Because the (private) travel demand is ever changing, due to new residential developments, changing infrastructure, or public transport services, it also highlights an important benefit of automating the process of constructing zoning systems and the incorporation of supply side information within this process. By adopting an automated method it becomes much easier to update the zoning system once an update on the travel demand

becomes available. All too often, in current practice, one only changes the trip matrices, but retains the underlying zoning system, even though it no longer might be appropriate.

We discuss the construction of our zoning system based on the identified zone components in Chapter 10, but first discuss how zone components are used to construct the demand-supply interface representation.

9.6.2 Traditional connector and centroid design

The following aspects are typically considered when constructing a traditional demand-supply interface, see also Chapter 5:

- Centroid placement
- Number of connectors
- Connector placement
- Connector cost

The assumptions leading to the placement of centroids is mainly driven by the desire to represent the “average” trip departure/arrival location. This location is then utilised to achieve the actual underlying objective of finding representative - average - costs for the zone’s trips to access or egress from the physical road network, modelled via connectors. Therefore, it is not the centroid nor the connector that matters. What matters is to have a mechanism that can reflect this representative access/egress cost. Yet, existing approaches all too often cling to the centroid/connector paradigm without being fully aware of this underlying objective. We choose to first focus on estimating representative access/egress costs and only then determine how this relates to the existing paradigm of centroid and connectors.

9.6.3 Connectoids

Our first premise in the construction of the costs to access/egress from a zone (component) is that, because centroids are a virtual construct, this cost should ideally not be related to a centroid’s location. Instead, it should follow from the zone’s internal trip travel times to and from its boundaries. A zone boundary is a point where its internal *non-keep* infrastructure interacts with the *physical* road network. Our second premise in constructing connector costs is that it is difficult to assess which points of interaction with the physical road network are “better” than others. Hence, the choice for a limited number of modelled connections, as far as we can see, remains rather arbitrary. The only way to avoid this, is to accept all points of interaction as valid. It might well be that some of these points are used more than others, but given that the computational cost of a traffic assignment model is influenced more by the number of paths and zones than the complexity of the network, this is not considered to be a problem. In fact, as long as the estimated access/egress costs remain representative, this approach more closely resembles reality than an arbitrary limited number of points. We refer to these points of interaction as *connectoids*; they are a node based interface to the zone’s demand, without the need for an explicit centroid location nor connector link.

Note that in case of traditional virtual connectors it is often discouraged to directly link them to intersections because they do not represent physical roads. With connectoids this problem no longer exists because connectoids reside one link away from the keep network, so that only

true physical roads are available to get to a connectoid. Figure 9.12(a) depicts connectoid locations based on this rationale for our example network. Since connectoids should interface with the keep network, the link separating the connectoid from the keep network must be added to the keep network as well, as is shown in Figure 9.12(b). We do emphasize that this added keep link is a physical link, with its own travel time cost, and is therefore not part of the, to be estimated, connectoid cost.

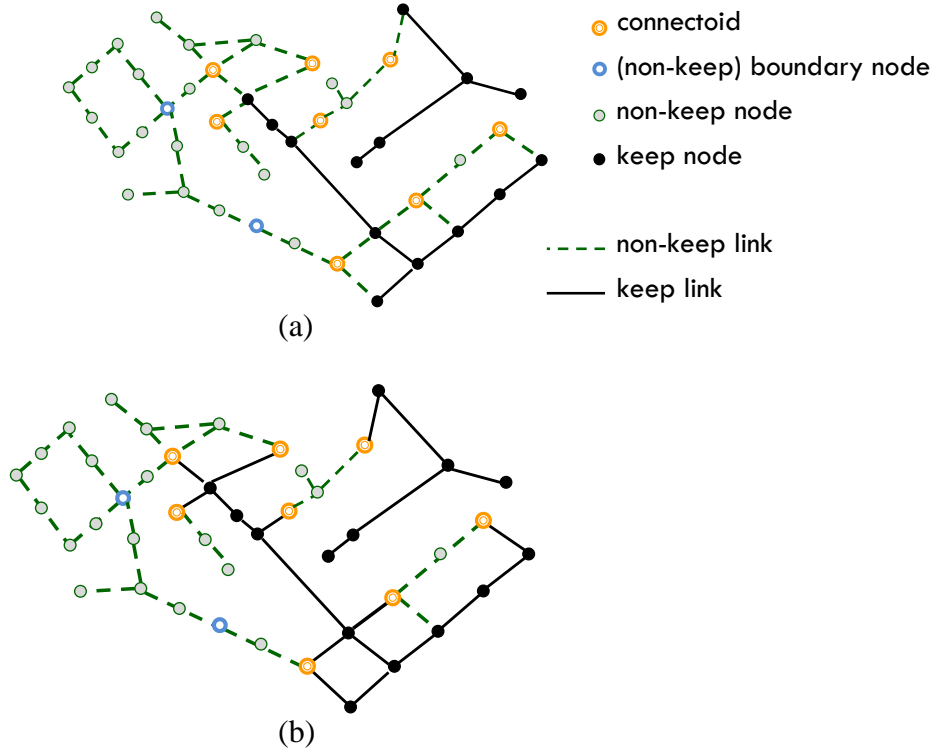


Figure 9.12: (a) Basic connectoid placement, one link away from keep network, (b) extending keep link network to guarantee connectoid connectivity.

The identified connectoids are formalised on a per zone component basis. We differentiate between departure connectoids, denoted by indicator matrix $\vec{\mathbf{N}}^+ \in \mathbb{F}_2^{\bar{Z} \times \mathbf{N}}$, and arrival connectoids, denoted $\vec{\mathbf{N}}^- \in \mathbb{F}_2^{\bar{Z} \times \mathbf{N}}$. This does not mean they are physically different nodes, but each connectoid can take on different roles. This allows for the separate estimation of departure and arrival connector costs which is particularly useful in the presence of one-way streets. The aforementioned additional keep links that let connectoids interface with the physical road network are formalised through $\vec{\mathcal{K}}$, which extends the original formulation of \mathcal{K} through:

$$\vec{\mathcal{K}}_{na} = \begin{cases} 1, & \text{if } \exists(\vec{z}, a, n'): \mathcal{K}_{n' \bullet} \mathbf{1} \geq 1 \text{ and } (\vec{N}_{\vec{z}n}^- A_{n'a}^+ A_{na}^- \parallel \vec{N}_{\vec{z}n}^+ A_{n'a}^- A_{na}^+) = 1, \\ \mathcal{K}_{na}, & \text{otherwise,} \end{cases} \quad (9.17)$$

with $\vec{z} \in \{1, \dots, \bar{Z}\}$, $a \in \{1, \dots, A\}$, and $n' \in \{1, \dots, \mathbf{N}\}$. The first case in Equation (9.17) marks link a as an additional keep link, when link a is an incoming or outgoing link of connectoid n , and its respective upstream/downstream node n' is marked as a keep node already.

9.6.3.1 Disconnected zone components

In most cases zone components will have direct access to the keep network and therefore have connectoids available to them. It is however possible that a zone component ends up without connectoids given the formulation presented in the previous section. In our example network this is the case for zone components \bar{z}_4 and \bar{z}_2 , see Figure 9.13.

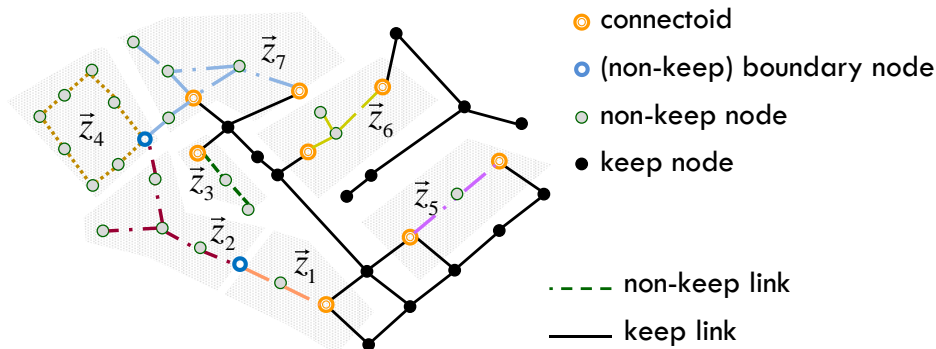


Figure 9.13: Zone components with and without connectoids.

Since a zone component might become an actual zone, access to the retained physical road network needs to be provided for. One could allow trips from \bar{z}_4 and \bar{z}_2 to access through other zone's their existing connectoids, but this would result in a rather convoluted demand-supply interface and additional complexity in estimating the connectoid costs. Instead, we prefer to create a situation where each zone gains its own dedicated connectoids that are also guaranteed to be directly connected to the final keep network. Furthermore, disconnected zone components \bar{z}_4 and \bar{z}_2 might or might not be grouped with other zone components when constructing the final zoning system. Zone component \bar{z}_4 could for example be merged with \bar{z}_7 . However, if this is not the case, any trips between \bar{z}_4 and \bar{z}_7 need to be modelled explicitly and physical infrastructure between the two zone components is required. In reality, these trips likely traverse the boundary node separating the two zone components. The boundary node is therefore chosen to become a physical node in this situation. When this happens, by definition, connectoids are then constructed around the, now physical and retained boundary node, as shown in Figure 9.14.

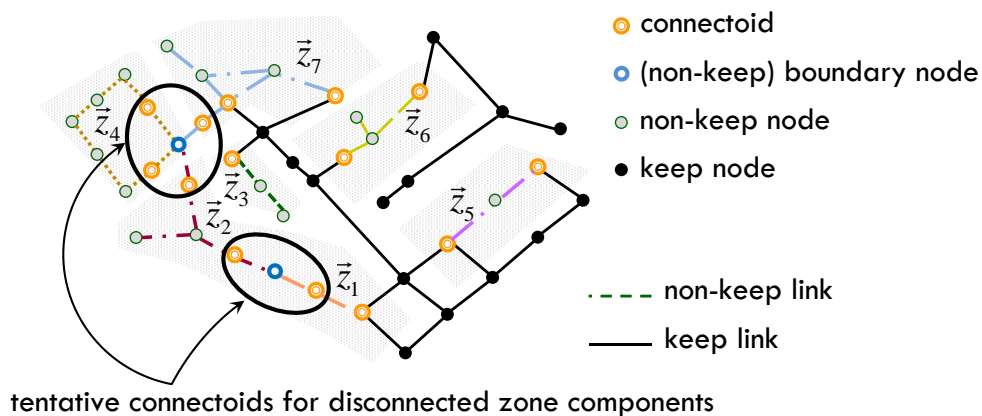


Figure 9.14: Boundary nodes and their tentative connectoids.

This ensures the connectivity of disconnected zone components to the physical road network as well. We only apply this procedure once the final zoning system is in place and only in case a zone (component) is still disconnected from the physical road network. Hence, the connectoids around the non-keep boundary nodes are – at this stage - tentative. Depending on the outcome of the zonal clustering in Step 5, tentative connectoids become permanent or not. When a final zone is disconnected its tentative connectoids and boundary node(s) are retained, otherwise they are disposed of. Given that at this stage we do not know which of the two occurs, we treat tentative connectoids as regular connectoids and include them in $\bar{\mathbf{N}}^+$, $\bar{\mathbf{N}}^-$, respectively.

9.6.4 Final network representation

When we choose to retain the boundary nodes of disconnected zone components \bar{z}_4 and \bar{z}_2 (in Figure 9.14) in the final network representation, connectivity to the keep network must be satisfied. We therefore identify the shortest path from each non-keep boundary node to the keep network. Whenever it is decided to retain the boundary node due to zonal disconnectivity, the links on this shortest path are included in the keep network. Let us consider this for our example network under the assumption that \bar{z}_1 and \bar{z}_2 are grouped, but all other zone components remained unclustered. We then find that zone component \bar{z}_4 remains disconnected and requires its boundary node to be retained, its tentative connectoids to be made final, and connectivity to the keep network must be ensured by including all links on the shortest path to the keep network to be part of the final keep network as well. This is illustrated in Figure 9.15.

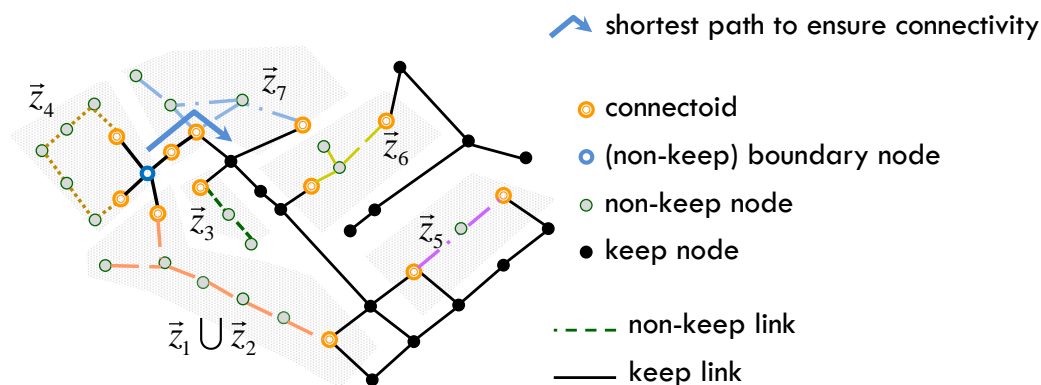


Figure 9.15: Example of retained and discarded boundary nodes based on assumed final zoning where only \bar{z}_1 and \bar{z}_2 are merged.

To avoid postponing the formulation of constructing the final network representation $\mathcal{K}^* \in \mathbb{F}_2^{N \times A}$ until we discussed the zonal clustering method, let us, for now, assume we know which boundary nodes are to be retained given some final clustering (and resulting disconnectivity of the final zones). These retained boundary nodes are denoted via $\mathbf{N}^{o*} \in \mathbb{F}_2^{Z^* \times N}$, with final zones $z^* \in \{1, \dots, Z^*\}$. We refer the reader to Chapter 10 regarding details on how to construct \mathbf{N}^{o*} as well as on how to construct the final zoning system. Given the retained boundary nodes, the final keep link network \mathcal{K}^* is then constructed by supplementing keep link network $\vec{\mathcal{K}}$ with the links surrounding retained boundary nodes, as well as the links on the shortest path from a retained boundary node to the keep network via:

$$\mathcal{K}_{na}^* = \begin{cases} A_{na}^+ \parallel A_{na}^-, & \text{if } \mathbf{1}^T N_{\bullet n}^{\circ*} \geq 1, \\ (\mathbf{1}^T N_{\bullet n}^{\circ*}) \bar{P}_{pa}, & \text{if } \exists(p, n') \text{ with } \forall n'' : \bar{P}_{np}^+ \bar{P}_{n'p}^- \bar{H}_{mn'} \leq \bar{H}_{mn''} \text{ given } (\bar{\mathcal{K}}_{n'} \cdot \mathbf{1})(\bar{\mathcal{K}}_{n''} \cdot \mathbf{1}) \geq 1, \\ \bar{\mathcal{K}}_{na}, & \text{otherwise,} \end{cases} \quad (9.18)$$

with $p \in \{1, \dots, P\}$, and $n', n'' \in \{1, \dots, N\}$. The first case adds link a , adjacent to node n , conditional on node n being a retained boundary node, to the keep network. The second case marks link a on path p as a keep node, conditional on path p departing from retained boundary node n and terminating at a keep node n' such that no cheaper path exists departing from the same boundary node and terminating at any (other) keep node n'' . The final network representation \mathbf{A}^* is subsequently be obtained via:

$$\mathbf{A}^* = (\mathcal{K}^* \circ \mathbf{A}^+) (\mathcal{K}^* \circ \mathbf{A}^-)^T. \quad (9.19)$$

Effectively, \mathbf{A}^* is the original network where non-keep topology is filtered out based on \mathcal{K}^* . Filtering is applied via element-wise multiplication. Figure 9.16 depicts the final network representation \mathbf{A}^* for our example network based on the assumed retained boundary node in Figure 9.15. This concludes the supply side component of our method. In the remainder of this chapter we focus on the estimation of the connectoid cost and construction of the final zoning system.

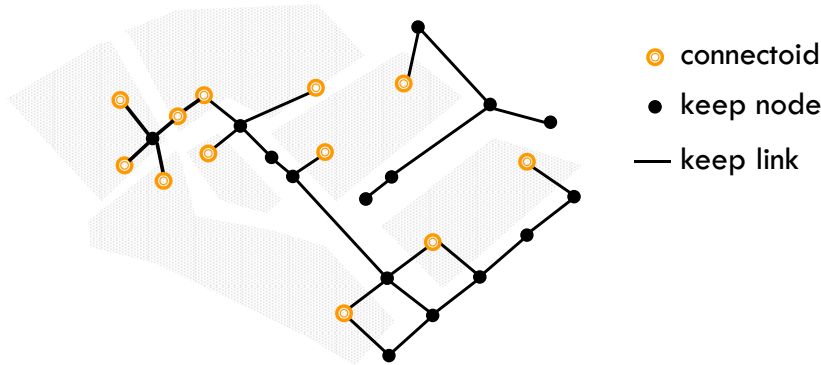


Figure 9.16: Example final network representation conditional on assumed zonal system.

9.6.5 Connectoid cost

Unlike existing methods, we do not determine the cost to access or egress a zone (component) based on a centroid location. In fact, there is no need for centroids to have a physical location at all. Instead, connectoid costs are estimated based on the abstracted out underlying non-keep network. Recall from Section 9.6.1 that the internal infrastructure of each zone component is expected to have stable internal travel times, therefore we can estimate disaggregate travel times from nodes internal to the zone component, to the zone component's connectoids with a comparatively high level of accuracy. Then, these travel times need to be aggregated to obtain a representative average connectoid cost. This cost is then attached to the connectoid node obviating the need for a separate *virtual* connector link, with a cost based on some *virtual* link length, *virtual* link speed, and *virtual* location of a *virtual* centroid. Instead, connectoids can be thought of as a boom gate to access any location in the zone directly, while the access/egress cost to utilise the connectoid resembles a (travel time) toll that must be paid to do so. This

concept is illustrated in Figure 9.17. Of course, for assignment purposes, it might still be convenient to have centroids and connectors. If so, we suggest to attach the connectoid cost onto a (zero length) connector link that does not contain any physical link characteristics. Hence, a connector should not have a length, nor a speed, nor a capacity, reflecting that it truly is a virtual connection. Similarly, the accompanying centroid has no meaningful coordinates and because it is no longer representative as a zone's centre, the term *zonoid* would, for example, be more appropriate.

In the remainder of Step 4, we propose three different methods to estimate connectoid cost: a base method and two additional methods that propose an additional scaling factor on top of the base method to improve the initial estimate. All three methods require the following data:

- Zone component topology (see Section 9.6.1)
- Disaggregate node-to-node demand (see Section 9.5.2)
- Connectoids (see Section 9.6.3)

We formulate the estimation of connectoid costs on the level of zone components. Of course, when clustering zone components in Step 5, connectoid costs are affected because they suddenly reside in a cluster rather than a zone component. However, this discussion is postponed until we discuss the clustering process itself (Chapter 10).

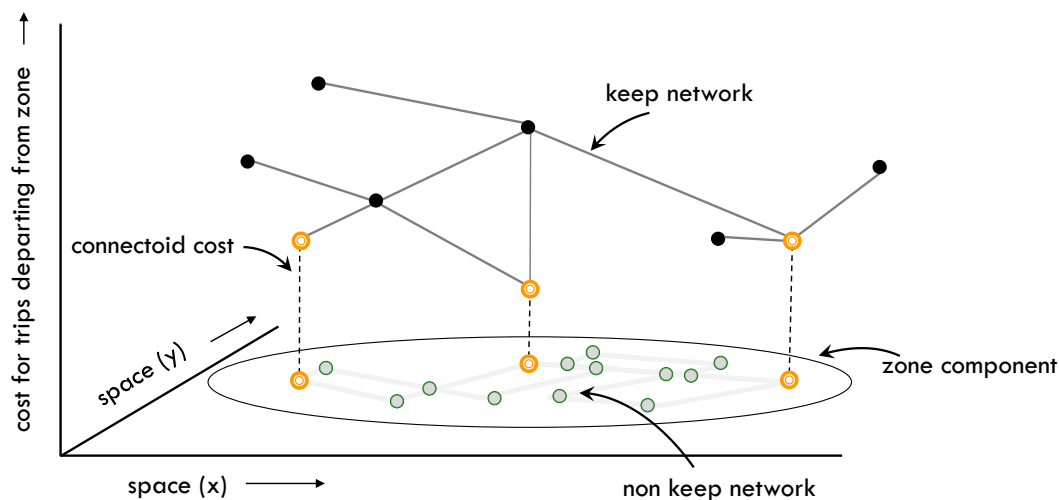


Figure 9.17: Conceptual illustration of zones as centroids and connectoid (egress) cost as independent fixed node cost.

9.6.6 Connectoid cost: base method

In this base method, we simply determine a connectoids' access/egress cost by taking the average travel time from each node internal to the zone component of the connectoid to the respective connectoid itself. This travel time is node weighted based on the - original zone based - node weight vector \mathbf{w}^{*z} . We formulate these - zone component - weights, denoted $\bar{\mathbf{w}}^z \in \mathbb{R}_+^{N \times 1}$, via:

$$\bar{w}_n^z = \begin{cases} w_n^{*z}, & \text{if } N_{zn} N_{\bar{z}n} = 1, \\ 0, & \text{otherwise.} \end{cases} \quad (9.20)$$

In other words, each node retains its original weight, but the relation is based on the node and its zone component rather than the original zone. As an example, let us estimate the egress cost of connectoid node n_1 in Figure 9.18.

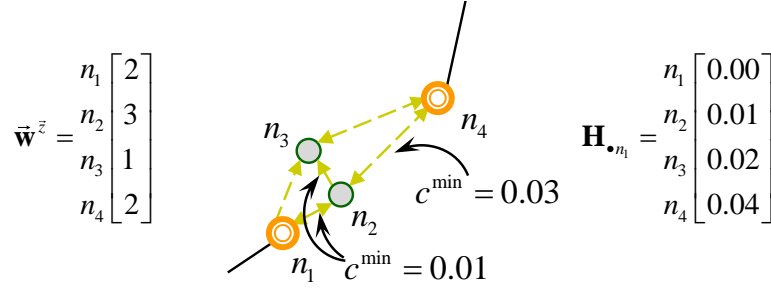


Figure 9.18: Close-up of zone component \bar{z}_6 and its egress cost estimation for connectoid n_1 , under uniform connectoid usage, hypothetical node weighting $\bar{\mathbf{w}}^{*z}$, and hypothetical node-to-node shortest path cost \mathbf{H} . Some links are one-way only.

Observe that the cost to travel from n_1 to itself is zero by definition. Also, since connectoids are internal to the zone, they do carry demand and therefore both n_1 and n_4 contribute to the average travel time to utilise n_1 , as do regular internal nodes. Hence, the egress cost to use connectoid n_1 is then simply found to be $(2 \cdot 0 + 3 \cdot 0.01 + 1 \cdot 0.02 + 2 \cdot 0.04) / 8 = 0.01625$ (h). We denote the connectoid egress costs via $\bar{\mathbf{H}}^+ \in \mathbb{R}^{\bar{Z} \times N}$, which in turn is obtained via:

$$\bar{H}_{\bar{z}n}^+ = \bar{N}_{\bar{z}n}^+ \left(\frac{1}{\mathbf{1}^T \bar{\mathbf{w}}^{\bar{z}}} \sum_{n'=1}^N \bar{w}_{n'}^{\bar{z}} \bar{H}_{n'n} \right), \quad (9.21)$$

with $\bar{z} \in \{1, \dots, \bar{Z}\}$, $n \in \{1, \dots, N\}$. Observe that non-connectoid entries n for zone component \bar{z} are filtered out and default to a zero result due to $\bar{N}_{\bar{z}n}^+$. We define the connectoid access costs in a similar fashion:

$$\bar{H}_{\bar{z}n}^- = \bar{N}_{\bar{z}n}^- \left(\frac{1}{\mathbf{1}^T \bar{\mathbf{w}}^{\bar{z}}} \sum_{n'=1}^N \bar{w}_{n'}^{\bar{z}} \bar{H}_{nn'} \right). \quad (9.22)$$

This approach is straightforward to use and, at first glance, makes intuitive sense. However, it may lead to inaccuracies in certain situations. For example, trips originating far from one connectoid and close to another connectoid, are not likely to utilise both connectoids equally. It is expected that there will be a bias towards using the connectoid that is closest, because in general, keep network infrastructure consists of main roads with higher speeds and therefore comparatively lower travel times. Also, in some situations, double counting can occur between the connectoid cost and the path cost leading up to the connectoid. Based on these two observations, we propose two extensions to the original base estimates that are able to scale the originally estimated costs to mitigate these potential issues. These scaling methods are complementary and yield a, per connectoid, multiplication factor denoted via $\bar{\mathcal{X}}^I, \bar{\mathcal{X}}^{II} \in \mathbb{R}^{\bar{Z} \times N}$, respectively. They are formulated on a zone component basis so the scaling is easily applied by element-wise matrix multiplication, for example via $\bar{\mathbf{H}}^- \circ \bar{\mathcal{X}}^I \circ \bar{\mathcal{X}}^{II}$, and $\bar{\mathbf{H}}^+ \circ \bar{\mathcal{X}}^I \circ \bar{\mathcal{X}}^{II}$, respectively. Alternatively, they can be applied only as a post-processing step after the clustering procedure in Step 5 completes.

9.6.7 Connectoid cost: service area scaling method

The first scaling method is based on the premise that the originally estimated cost is, in most cases, an overestimation of the true connectoid cost. We argue this is in fact the case whenever there resides more than a single connectoid on the zone component, since having more connectoid options, in reality, only reduces free flow path travel times. Hence, the more connectoids the more local the geographical area that the connectoid services.

This method aims to compensate for the expected reduction of the connectoid's service area. This is a deliberately basic method to find out if we can achieve improvements in our estimates with a minimal amount of computational effort. Hence, we first make a number of simplifying assumptions: (i) each zone component is represented by a unit circle, i.e. radius $r = 1$, (ii) connectoids are assumed to reside on the edges of a zone component, (iii) connectoids are assumed to be uniformly distributed across the zone component edge, (iv) origins and destinations of trips are uniformly and continuously distributed in space, (v) trips are modelled as-the-crow-flies between points (origin-connectoid, connectoid-destination, etc.),

As an example, consider the zone component with four connectoids in Figure 9.19(a). We can then draw the *average* cost contour around the connectoid. This contour should be thought of like the following: if we were to draw an infinite number of lines between the connectoid and all locations on the zone's edge, and then for each line determine its middle point; this middle point is the average cost of all origins/destinations on this line based on our assumptions. Then, if we would only draw these middle points, we obtain the shown average cost contour. We then find four overlapping areas that reflect locations where the average cost of trips utilising the connectoid is larger than the travel time from the contour to the closest connectoid. We now make two additional assumptions; (vi) all origins and destinations in the zone are projected onto this average cost contour, (vii) trips desire to use the connectoid that is closest to them. In this situation, the trips on the overlapping contour areas are in fact attributed to a suboptimal connectoid.

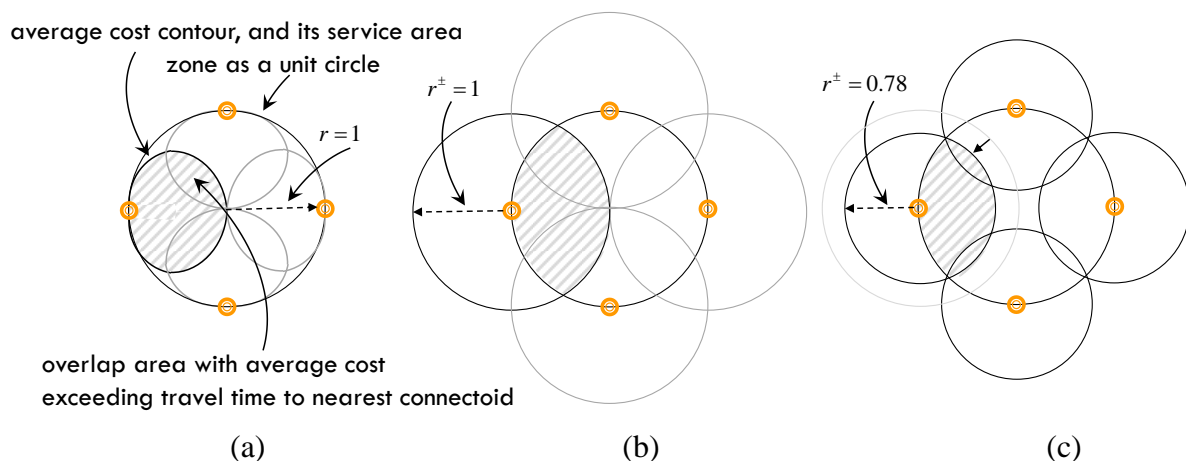


Figure 9.19: (a) Stylised zone and its average cost service area per connectoid, (b) same service area approximated by unit circles, (c) reduced service area based on estimated service area scaling factor.

We use this observation to determine our *service area scaling factor*, where the area of the contour is considered to be the *service area*. For simplicity, we first replace the “true” contour

by unit circle proxies as depicted in Figure 9.19(b). The radius of this proxy unit circle, denoted by r^\pm and initially $r^\pm = 1$, can be thought of as the original unscaled connectoid cost estimate. We then argue that we should minimise the number of trips (projected onto the contour) that exhibit an average cost higher than the cost to access the closest connectoid. In other words, overlap between contours should be minimised while still servicing as much of the zone as possible. We do so by scaling back the connectoid's service area to the point that the combined service areas of all connectoids match the total surface area of the zone component, see Figure 9.19(c). This desired situation is formulated as follows:

$$\pi r^2 = \Theta(r, r^\pm, \Delta_{r^\pm}^r) \sum_{n=1}^N (\vec{N}_{\vec{z}_n}^+ \parallel \vec{N}_{\vec{z}_n}^-), \quad \vec{z} \in \{1, \dots, \vec{Z}\}, \quad (9.23)$$

with on the left hand side the surface area of stylised zone component \vec{z} with radius $r = 1$, and on the right hand side function $\Theta(r, r^\pm, \Delta_{r^\pm}^r)$, which yields the surface covered by each connectoid service area (due to symmetry they are all the same). This area is then multiplied by the number of connectoids available in \vec{z} . The connectoid service area, obtained via $\Theta(\cdot)$, is a *lens*; a shared area between two intersecting circles with radii r, r^\pm , respectively, where the latter represents the radius of the circle around the connectoid. The area of this lens, besides the two radii, is also conditional on the distance between the two circle centres, denoted $\Delta_{r^\pm}^r$. An impression is provided in Figure 9.20(a).

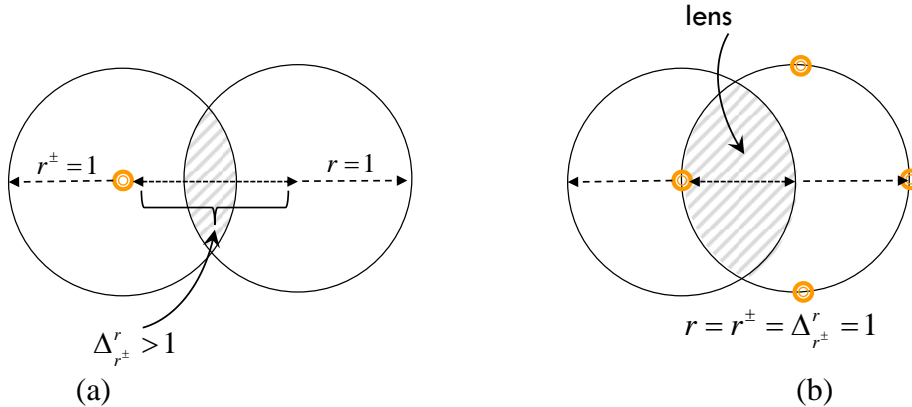


Figure 9.20: (a) Lens area of two circles in general, (b) assuming second circle resides on initial circles edge.

We compute lens area $\Theta(\cdot)$ like the following (Weisstein, n.d.):

$$\Theta(r, r^\pm, \Delta_{r^\pm}^r) = (r^\pm)^2 \arccos\left(\frac{(\Delta_{r^\pm}^r)^2 + (r^\pm)^2 - r^2}{2\Delta_{r^\pm}^r r^\pm}\right) + r^2 \arccos\left(\frac{(\Delta_{r^\pm}^r)^2 + r^2 - (r^\pm)^2}{2\Delta_{r^\pm}^r r^\pm}\right) - \frac{1}{2} \sqrt{(-\Delta_{r^\pm}^r + r^\pm + r)(\Delta_{r^\pm}^r + r^\pm - r)(\Delta_{r^\pm}^r - r^\pm + r)(\Delta_{r^\pm}^r + r^\pm + r)}, \quad (9.24)$$

In this particular case, Equation (9.24) can be simplified because connectoids reside on the edge of the zone, i.e. $\Delta_{r^\pm}^r = 1$, yielding:

$$\Theta(r, r^\pm) = r_2^2 \arccos\left(\frac{(r^\pm)^2}{2rr^\pm}\right) + r^2 \arccos\left(\frac{1+r^2-(r^\pm)^2}{2rr^\pm}\right) - \frac{1}{2}\sqrt{(r^\pm)^2(2r-r^\pm)(2r+r^\pm)}. \quad (9.25)$$

Furthermore, the zone itself is also a unit circle, so $r = 1$. This then yields:

$$\Theta(r^\pm) = r_2^2 \arccos\left(\frac{(r^\pm)^2}{2r^\pm}\right) + \arccos\left(\frac{2-(r^\pm)^2}{2r^\pm}\right) - \frac{1}{2}\sqrt{4(r^\pm)^2 - (r^\pm)^4}. \quad (9.26)$$

Knowing that the original connectoid cost estimate is assumed to be $r^\pm = 1$, then, choosing r^\pm such that Equation (9.23) is satisfied means that r^\pm also reflects our desired service area scaling factor, which we generally denote via $\vec{\mathcal{X}}^I \in \mathbb{R}^{\vec{Z} \times N}$. We therefore rewrite Equation (9.23) to solve for r^\pm directly, as well as setting non-connectoid nodes to zero, yielding the, zone component \vec{z} , connectoid n specific, service area scaling factor $\vec{\mathcal{X}}_{\vec{z}n}^I$ via:

$$\vec{\mathcal{X}}_{\vec{z}n}^I = \begin{cases} \min\left\{1, \Theta^{-1}\left(\frac{\pi}{\sum_{n=1}^N (\vec{N}_{\vec{z}n}^+ \parallel \vec{N}_{\vec{z}n}^-)}\right)\right\}, & \text{if } \vec{N}_{\vec{z}n}^+ \parallel \vec{N}_{\vec{z}n}^- = 1, \\ 0, & \text{otherwise,} \end{cases} \quad (9.27)$$

with $\vec{z} \in \{1, \dots, \vec{Z}\}, n \in \{1, \dots, N\}$, and the inverse of the original lens area function, denoted $\Theta^{-1}(\cdot)$. Note that if the number of connectoids is less than two, the method actually yields a scaling factor >1 , therefore the minimum clause in Equation (9.27) caps the factor at the original cost estimate. Table 9.1 shows some of the different scaling factors obtained, conditional on the number of connectoids in the zone component.

Table 9.2: Service area scaling factors depending on number of zone component connectoids.

| Number of connectoids | Service area scaling factor |
|-----------------------|-----------------------------|
| 2 | 1.00 (capped) |
| 3 | 0.91 |
| 4 | 0.78 |
| 5 | 0.69 |
| 6 | 0.62 |
| 7 | 0.57 |
| 8 | 0.53 |

This approach is attractive because it is simple, justifiable, does not require calibration, and can improve modelling results, as our case study in Chapter 11 reveals. It does suffer from the drawback that it at least requires 3 connectoids before an actual reduction in cost can be modelled. We also found that because the method assumes all connectoids reside on a zone's edge, it cannot account for any travel time double counting effects that occur when a connectoid is more centrally located. Therefore, a second complementary scaling method is proposed, in addition to the service area scaling factor.

9.6.8 Connectoid cost: centrality scaling method

The original connectoid cost estimates only consider the internal topology of the zone component. No information on path travel times leading up to the use of a connectoid are accounted for. This can lead to unwanted side effects. This is best illustrated with an example. Consider the situation that all inbound paths into a zone component arrive from a single direction, as per Figure 9.21(a).

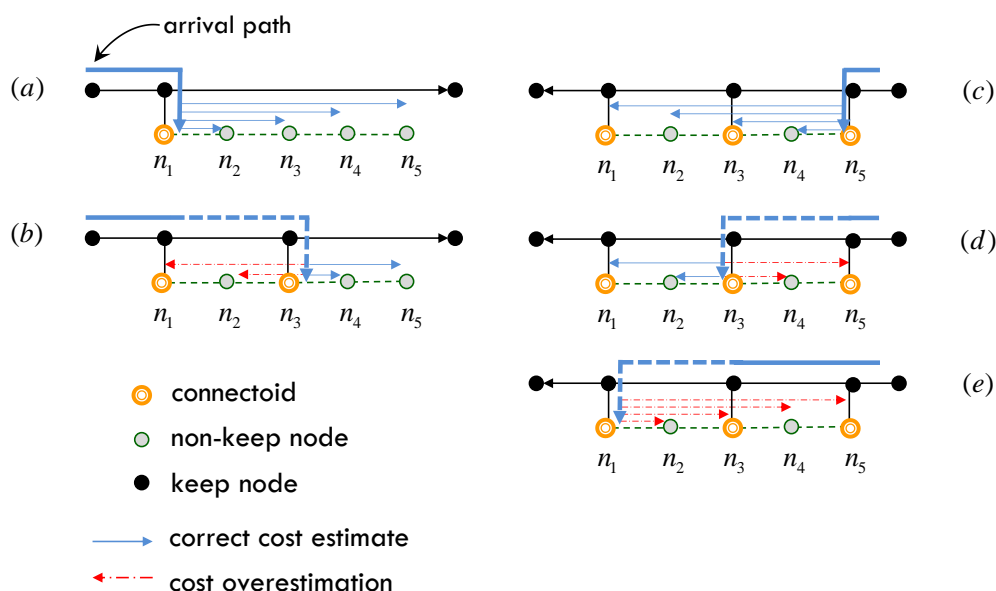


Figure 9.21: Path travel time overestimation through connectoid cost double counting, under demand arriving from East: (a) correct average cost estimate, (b) incorrect overestimated connectoid cost. Demand arriving from West: (c) correct cost estimate, (d) incorrect overestimated connectoid cost, (e) correct overestimated connectoid cost.

Initially, a single connectoid n_1 exists and the base method yields an accurate average cost estimate because the combination of external path travel time plus the travel time from the connectoid to each destination yield the actual total shortest path travel time. Therefore, the base connectoid cost estimate is correct. However, when another connectoid is introduced by the likes of n_3 , we observe that the base estimate connectoid cost of n_3 is compromised, because some of the destinations reached via n_3 require to first travel into the zone and then partly backtrack again, i.e. implicit double counting of travel time occurs. In that case, the overall path travel time to reach destinations n_1 and n_2 is overestimated, see Figure 9.21(b). As a result, the average connectoid cost for n_3 is overestimated, which leads to an unused connectoid in the actual assignment, even though destinations n_4 and n_5 are reachable at minimum cost via n_3 . Conversely, when assuming all demand enters from the opposite direction and assuming another connectoid in n_5 , only the connectoid cost of n_5 is correct, see Figure 9.21(c), while connectoids n_1 and n_3 become overestimated. This overestimation in itself need not to be problematic given that none of the destinations reachable from n_1 are viable shortest paths (Figure 9.21(e)), then the double counting effect in n_1 is, arguably, an effective way of modelling its unattractiveness. However, again, connectoid n_3 has viable destinations in Figure 9.21(d), is assigned an overestimated average cost, and will – incorrectly so – not be utilised in assignment.

We argue that the more central a connectoid is located within a zone, the more likely it is that it suffers from travel time double counting effects, where some of its destinations are unrealistically unattractive, where others are not, causing the average connectoid cost to be compromised making it unrealistically unattractive compared to its edge located peers. Edge located connectoids do not suffer from this effect because their cost estimate results in the desired self-selection of trip demand that one would expect conditional on the arrival direction of the paths.

One might reason that this is not a problem, because paths will simply avoid internal connectoids. This however also means that paths bypass the maximum amount of physical infrastructure possible, leading to reduced accuracy of link loads in assignment. It would be better if we can compensate for this effect, yield better overall travel time estimates for internal connectoids, and get better utilisation of the internal zone infrastructure. To do so, we propose a method that estimates a *centrality scaling factor* for connectoid costs based on the expected magnitude of travel time double counting. Since edges of a zone component are not negatively affected by travel time double counting, their scaling factor should revert to one, i.e. the original cost remains. Conversely, the scaling factor in the centre of the zone, denoted by χ^{\min} , experiences the full magnitude of travel time double counting and serves as a lower bound on the connectoids centrality scaling factor. This results in a centrality scaling factor matrix $\vec{\mathcal{X}}'' \in [\chi^{\min}, 1]^{\vec{Z} \times N}$. We point out that we adopted the term *centrality* to indicate that the scaling relates to the location relative to the centre, it has no relation with, for example, the concept of centrality in graphs, which has a completely different meaning altogether.

We estimate χ^{\min} in a general fashion in Section 9.6.8.2, independent of zone specific characteristics. Then, χ^{\min} is partially applied through linear interpolation, depending on the location of connectoid n , within zone \vec{z} , captured via $\vec{\chi}_{\vec{z}n}^{\Delta} \in [0, 1]$. This results in the connectoid specific centrality factor $\vec{\chi}_{\vec{z}n}''$ via:

$$\vec{\chi}_{\vec{z}n}'' = \begin{cases} \chi^{\min} + \vec{\chi}_{\vec{z}n}^{\Delta}(1 - \chi^{\min}), & \text{if } \vec{N}_{\vec{z}n}^+ \parallel \vec{N}_{\vec{z}n}^- = 1, \\ 0, & \text{otherwise,} \end{cases} \quad (9.28)$$

The remainder of this section is dedicated to formalising $\vec{\chi}_{\vec{z}n}^{\Delta}$ and estimating χ^{\min} , respectively.

9.6.8.1 Location based scaling

The portion $\vec{\chi}_{\vec{z}n}^{\Delta}$ of the centre point centrality factor χ^{\min} attributed to a connectoid n within a zone component \vec{z} , is determined based on its location relative to the zone centre and zone extremities, as depicted in Figure 9.22.

A Cartesian coordinate system is assumed and the four extreme points of each zone component, are captured by vectors; $\vec{\mathbf{x}}^{\max}, \vec{\mathbf{y}}^{\max}, \vec{\mathbf{x}}^{\min}, \vec{\mathbf{y}}^{\min} \in \mathbb{R}_+^{\vec{Z} \times 1}$, respectively. Similarly, we define the virtual centre point vectors through $\vec{\mathbf{x}}, \vec{\mathbf{y}} \in \mathbb{R}_+^{\vec{Z} \times 1}$, respectively. The centre point location $(\vec{x}_{\vec{z}}, \vec{y}_{\vec{z}})$ for each zone component \vec{z} is simply determined via:

$$\vec{x}_{\vec{z}} = \frac{1}{2}(\vec{x}_{\vec{z}}^{\max} + \vec{x}_{\vec{z}}^{\min}), \quad \vec{z} \in \{1, \dots, \vec{Z}\}, \quad (9.29)$$

$$\vec{y}_{\vec{z}} = \frac{1}{2}(\vec{y}_{\vec{z}}^{\max} + \vec{y}_{\vec{z}}^{\min}), \quad \vec{z} \in \{1, \dots, \vec{Z}\}. \quad (9.30)$$

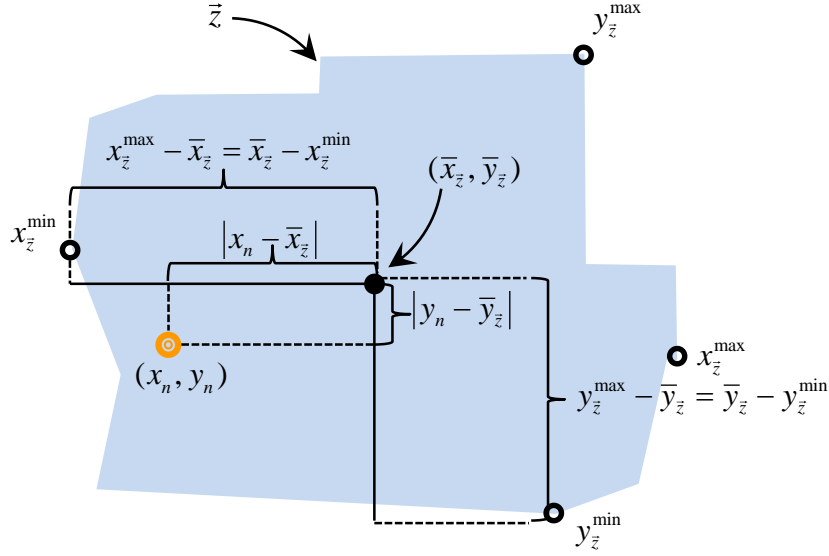


Figure 9.22: Schematic impression of location dependent factors influencing the scaling of a connectoid centrality factor.

We obtain $\vec{\chi}^\Delta \in [0,1]^{\vec{Z} \times N}$ by taking a distance weighted ratio per coordinate dimension on how close connectoid (x_n, y_n) is to the centre, compared to the zone's extremity such that:

$$\begin{aligned} \vec{\chi}_{zn}^\Delta &= \frac{\frac{|x_n - \bar{x}_z|}{(\bar{x}_z^{\max} - \bar{x}_z)} \cdot (\bar{x}_z^{\max} - \bar{x}_z) + \frac{|y_n - \bar{y}_z|}{(\bar{y}_z^{\max} - \bar{y}_z)} \cdot (\bar{y}_z^{\max} - \bar{y}_z)}{(\bar{x}_z^{\max} - \bar{x}_z) + (\bar{y}_z^{\max} - \bar{y}_z)} \\ &= \frac{|x_n - \bar{x}_z| + |y_n - \bar{y}_z|}{(\bar{x}_z^{\max} - \bar{x}_z) + (\bar{y}_z^{\max} - \bar{y}_z)}. \end{aligned} \quad (9.31)$$

The weighting that we include is needed to, at least partially, accommodate for the zone shape. Observe that when $(x_n, y_n) \equiv (\bar{x}_z, \bar{y}_z)$, then $\vec{\chi}_{zn}^\Delta = 0$ and we apply the full centrality factor χ^{\min} , as per Equation (9.28). Conversely, when $(x_n, y_n) \equiv (x_z^{\max}, y_z^{\max})$, then $\vec{\chi}_{zn}^\Delta = 1$, and hence the centrality scaling factor $\vec{\chi}_{zn}^{\Delta}$ is also 1. Finally, note that $\vec{\chi}_{zn}^\Delta \leq 1$, for all relevant connectoids because $(\bar{x}_z^{\max}, \bar{y}_z^{\max}) \geq (x_n, y_n)$ for each connectoid n in zone component \vec{z} .

9.6.8.2 Centre point centrality scaling factor

Let us now estimate the double counting of path costs that occurs for the virtual centre point. Our method is symmetric in the sense that it makes no distinction between departing or arriving trips when estimating χ^{\min} . For simplicity, we therefore only discuss the method from the perspective of arriving trips.

First, we make the following simplifying assumptions: (i) each centre point's original connectoid cost is represented by a unit circle with radius $r = 1$, (ii) all destinations are projected uniformly on the contour of this circle, (iii) paths with a destination in the zone component can utilise two virtual roads to enter the zone; a North-South road and an East-West road, (iv) destinations on the contour can be reached by traversing the contour, which can be accessed from the intersection points with the two roads, or, alternatively, by travelling to the connectoid and "paying" the original connectoid cost.

The underlying idea of this approach is the following: by letting paths reach a destination by using the original path plus connectoid cost, but also consider a potentially – but not necessarily – shorter path using the contour, we can compare the difference between the two approaches. Whenever a shorter path is found by using the contour, the original cost has likely been overestimated (in the actual zone and infrastructure) and double counting occurred. Based on these differences we obtain a rough, but justifiable estimate of the amount of double counting in our original estimate. Note, that by uniformly distributing destinations across the contour, only some of the destinations will be more attractive to reach via this contour.

Let us consider the extreme example depicted in Figure 9.23(a), where the shortest path cost h_p to an example destination at the intersection of the contour with the road is shown. This reflects the “true” path cost. However, when we utilise the centre connectoid to reach this destination, first additional travel time is required to reach the connectoid itself, and on top of that, the connectoid cost – in the reverse direction – must be “paid” as well, see Figure 9.24(b). To avoid this double counting effect and obtain the true path travel time while still using the centre connectoid, we must solve $h_p + r + \chi^{\min} r = h_p$, see Figure 9.25(c), where the centrality scaling factor χ^{\min} is utilised to alter the cost such that it reflects the true path cost to the destination. Note that χ^{\min} can only be applied to the connectoid cost estimate, so the first r to reach the connectoid cannot be scaled. Solving this results in $\chi^{\min} = \frac{-r}{r} = -1$, which negates the unjust double counting that resulted from the original cost estimate. Also note that it is therefore possible that a connectoid cost might indeed become negative in order to reflect the correct path travel time.

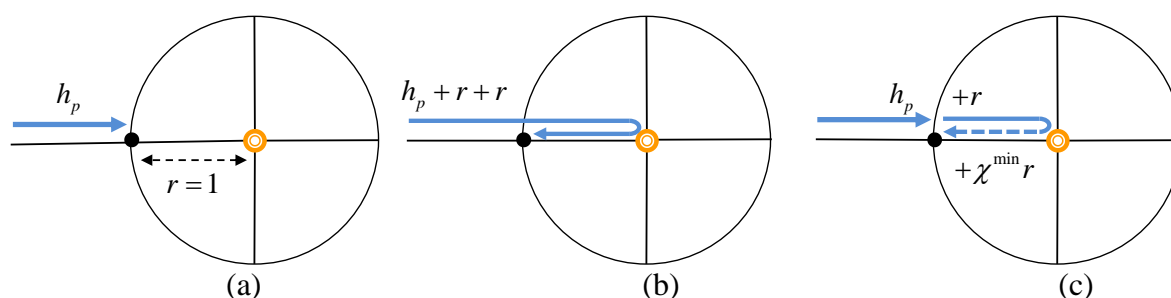


Figure 9.25: Example of travel time double counting under simplifying assumptions.

Let us now formalise this approach for all destinations. In practice, a path’s point of access into a zone depends on its destination but also on its origin. In general there will be a bias towards a convenient point of access, but due to external effects less optimal access points will also be utilised. In our stylised approach, we consider four principal points of access. For now, let us assume that all paths arrive *only* via the optimal point of access given their destination on the contour. This means that each main direction covers exactly $2\pi/4$ (radians) of the contour, with each “slice” intersected by one of the two roads, in turn splitting the slice in two identical sub-slices of $\pi/4$ each, see Figure 9.26(a). Observe that all destinations in a slice are cheapest to reach via the contour. When we know the angle (in radians) of a destination, denoted ω , we can determine the cost of traversing the contour from the point of intersection with the regular road section via:

$$\frac{\omega r}{2\pi r} \cdot 2\pi r = \omega r. \quad (9.32)$$

Observe that the further the destination resides from the intersection point, the costlier the contour based alternative path becomes and at some point, the original cost estimate becomes favourable.

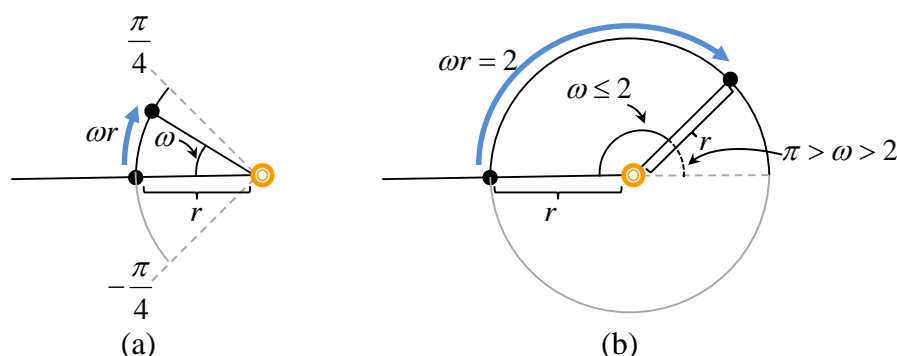


Figure 9.26: (a) Impression of destination “slice” serviced by ideal point of access, (b) all destinations serviced by each point of access.

In general, the “true” cost consists of the contour cost ωr to reach the destination, but requires subtracting the double counting for first reaching the connectoid, i.e. $-r$. The (single destination) scaling factor is then given by the ratio between the “true” cost and current connectoid cost via:

$$\frac{-r + \omega r}{r} = \omega - 1. \quad (9.33)$$

At the intersection point of the road with the contour, we find $\omega r = 0$, resulting in the aforementioned connectoid cost of $-r$. Given the assumption of a uniform distribution of destinations across the contour, we can compute the average scaling factor by integrating over the interval $[0, \pi/4]$ via:

$$\frac{4}{\pi} \int_0^{\pi/4} (\omega - 1) d\omega = \frac{4}{\pi} \left[\frac{\omega^2}{2} - \omega \right]_0^{\pi/4} = \frac{4}{\pi} \cdot \frac{\pi^2}{32} - \frac{4}{\pi} \cdot \frac{\pi}{4} = \frac{\pi}{8} - 1 \approx -61\%. \quad (9.34)$$

Hence, under the given assumptions, the centrality scaling factor accounting for path travel time double counting, becomes negative. Of course, not all paths do arrive from their ideal access point. To be able to account for this, we also estimate a second scaling factor, under the assumption that paths arrive uniformly across access points. We can estimate this cost by assuming all destinations are serviced (from a single point of access), leading to two symmetric slices of exactly π , as depicted in Figure 9.26(b). For each slice, it is cheapest to traverse the contour as long as $\omega r < 2$. After that, accessing the destination via the connectoid is preferred (it is cheaper) at the originally estimated and fixed cost of $r + r = 2$. The average cost across destinations with $\omega r < 2$, given $r = 1$, is again found via integration through:

$$\frac{1}{2} \int_0^2 \omega - 1 d\omega = \frac{1}{2} \left[\frac{\omega^2}{2} - \omega \right]_0^2 = \frac{1}{2} \cdot \frac{4}{2} - 2 = 0. \quad (9.35)$$

The destination weighted average scaling factor for the full slice, assuming all destinations are serviced, is then given by:

$$\frac{2 \cdot 0 + (\pi - 2) \cdot 1}{\pi} = \frac{\pi - 2}{\pi} \approx 36\%. \quad (9.36)$$

We can therefore expect the final centrality factor to be bounded by these two cases such that $(\pi/8) - 1 \leq \chi^{\min} \leq (\pi - 2)/\pi$, where the actual centrality scaling factor for the centre point is defined by assigning a portion ι to each of the two approaches via:

$$\chi^{\min} = \iota \left(\frac{\pi}{8} - 1 \right) + (1 - \iota) \left(\frac{\pi - 2}{\pi} \right), \quad 0 \leq \iota \leq 1, \quad (9.37)$$

where we postpone the discussion on estimating ι to Section 8.5.3, where we utilise a real world case study to do so. This concludes Step 4 of our multi-scale representation framework. The final step of the disaggregation-aggregation framework is discussed in a separate chapter due to its slightly different nature in being formulated as a (separate) constrained optimisation problem.

9.7 Synthesis and discussion

In this chapter we proposed an integrated approach to the design and representation of traffic assignment model inputs suitable for a multi-scale environment. A generic five step disaggregation-aggregation framework is introduced to accommodate the construction of the relevant supply side and demand side inputs to traffic assignment. In this chapter, the first four steps of this framework are formalised, leading to a concrete step-by-step procedure to construct a representative supply input representation, as well as a justifiable demand-supply interface via the concept of connectoids and their estimated costs. We fully acknowledge that our interpretation of each of the framework's steps represents just one of many possible approaches.

First and foremost, we attempted to achieve consistency in our proposed methodology. We argue that we succeeded by explicitly considering the notion of expected road usage to construct the supply side representation as well as adopting this metric to construct zone components with stable internal travel times which, in turn, serve as the foundation for constructing the desired granularity of the final zoning system. The main novelty however, lies in our representation (and estimation) of the demand-supply interface where the traditional centroid/connector paradigm is refined and replaced by the concept of connectoids. The cost of entering, or departing, the physical road network from, or to, a zone is no longer arbitrary in our approach. It does not rely on unverified assumptions, virtual link lengths, virtual speeds, and virtual centroid locations. Instead, costs are estimated based on the underlying complete network and the identified zone components with expected stable internal travel times. Additional scaling methods have been developed to further enhance the connectoid cost estimates.

9.7.1 Model limitations

The first steps in the framework that disaggregate the demand and perform subsequent disaggregate assignment are implemented relatively crudely, by adopting an AON based assignment approach. This approach does have a number of notable drawbacks, most of which we already discussed in Part I of this thesis. In the context of estimating expected road usage, the most notable drawbacks are found in that flows are not restricted to the physical road capacity leading to an overestimation of congested links. The fact that spillback is not catered for, so physical queues do not materialise, leads to an underestimation of congested links. Finally, due to the single iteration approach, congestion is not taken into account which can lead to inaccurate path choices. We partially account for these issues by imposing a conservative flow threshold κ^{\min} , but one should be aware of these limitations when analysing results under this modelling scheme.

10 Zonal representation: problem formulation and solution scheme

In this chapter we solely focus on step five of the proposed disaggregation-aggregation framework where we construct the new zoning system. There are two aspects to this step. First, the formulation of a cluster based constrained optimisation problem is discussed. This formulation determines the objectives and constraints involved in constructing the final zoning system, but does not specify how this is achieved. Second, based on the posed problem formulation, we are able to choose from a wide range of methods to design and implement solution schemes. We are primarily interested in demonstrating the potential of our method and therefore prefer a solution scheme that guarantees an optimal solution. We propose a customised branch-and-bound algorithm to achieve this. We specifically designed this algorithm to be suitable for spatial clustering procedures. In addition, we propose and incorporate multiple bounds in order to reduce the search space and increase the effectiveness of this solution scheme.

In Sections 10.1 and 10.2 we discuss the overall problem formulation and the underlying concepts. In Section 10.3, the objective function is discussed in more detail. The relationship between the criteria of constructing a “valid” zoning system and our imposed constraints is the topic of interest in Section 10.4. This is followed by Sections 10.5 and 10.6, where we discuss the two types of constraints adopted in our problem formulation. Issues around solving combinatorial problems - such as ours - are discussed in Section 10.7. We then propose our branch-and-bound solution scheme (Section 10.8), including the constraints (Sections 10.9 and 10.10) and partitioning method (Section 10.11). The final traffic assignment representation based on the final zoning system is discussed in Section 10.12. Lastly, we conclude with a summary and discussion in Section 10.13.

10.1 Problem formulation

To construct the final zoning system, we aggregate the zone components $\vec{z} \in \{1, \dots, \vec{Z}\}$ identified in Section 9.6.1. This aggregation process considers two (conflicting) objectives and a number of constraints, all of which are discussed and formalised in the following sections. The result of this procedure yields the final zones and their demand via $\mathbf{D}^* \in \mathbb{R}_+^{Z^* \times Z^*}$, $\mathbf{Z}^* \in \mathbb{F}_2^{Z^* \times N}$, respectively. Each final zone $z^* \in \{1, \dots, Z^*\}$ consists of one or more grouped zone components, with the number of final zones denoted by Z^* , where $Z^* \leq \vec{Z} \leq \bar{Z}$.

There are two explicitly considered objectives when constructing the zoning system: (i) minimise the total network distortion in terms of connectoid access cost δ^- , and connectoid egress cost δ^+ (veh.h), and (ii) minimise the penalty $\delta^{d^{\min}}$ (veh.h) imposed by the number of trips missing in the final zones. Missing trips should be interpreted as follows: to approximate the desired zonal granularity, we set a user-defined target number of trips denoted by d^{\min} (veh). When constructing the zones we try to let each final zone contain exactly this target number of trips. Whenever a zone does not meet this number of trips, the resulting difference is considered to be “missing” and for each missing trip a penalty is imposed.

The first objective measures the information loss, i.e. it represents $\varepsilon(\cdot)$ in the context of the representation framework of Chapter 2. The second objective serves as a way to achieve a particular granularity. It therefore controls the magnitude of scaling, i.e. it represents $\zeta(\cdot)$ in the context of the representation framework in Chapter 2. We propose to achieve these two objectives by formulating the zone design as a cluster based constrained optimisation problem.

Each zone component is considered a data point in the clustering procedure that follows from our problem formulation. The zone component characteristics are used as background information to support the clustering process. Following the terminology discussed in Chapter 7, we then develop a *semi-supervised* clustering approach, by introducing both *instance-level* and *cluster-level* constraints. The clusters that result from solving this problem describe the zone component partitioning that is the final zoning system.

Let us now discuss the final problem formulation and its constraints. Each component involved in this formulation is discussed in one of the subsequent sections. Note that all inequality constraints are defined to be element-wise inequalities on matrices, for example $\mathbf{G} \geq \mathbf{I}$, is identical to $G_{\bar{z}\bar{z}'} \geq I_{\bar{z}\bar{z}'}, \forall (\bar{z}, \bar{z}') \in \{1, \dots, \bar{Z}\}$. The constrained optimisation problem itself is provided, as a reference, in Equation (10.1).

$$\begin{aligned}
 & \text{minimise}(\delta^+ + \delta^- + \delta^{d^{\min}}) && \text{(Section 10.3)} \\
 & \text{s.t.} \\
 & \mathbf{G} \geq \mathbf{I}, && \text{(Section 10.2)} \\
 & \mathbf{G} = \mathbf{G}^T, && \text{(Section 10.2)} \\
 & \mathbf{K} \geq \mathbf{G}, && \text{(can-link constraints - Section 10.5)} \\
 & \mathbf{K}^G \geq \mathbf{G}. && \text{(cluster-level constraints - Section 10.6)}
 \end{aligned} \tag{10.1}$$

10.2 Clustering zone components

Before discussing the objective function or instance-level and cluster-level constraints, we introduce some prerequisites in order to formalise our clustering based procedures. *Clustering*, interchangeably referred to as *grouping* in this context, is applied to zone components. We denote a clustering of zone components via grouping matrix $\mathbf{G} \in \mathbb{F}_2^{\bar{Z} \times \bar{Z}}$. This grouping matrix is symmetric because whenever \bar{z}_1 is grouped with \bar{z}_2 , then \bar{z}_2 must also be grouped with \bar{z}_1 , i.e. $\mathbf{G} = \mathbf{G}^T$. A zone component is, also by definition, always grouped with itself, i.e. $G_{\bar{z}\bar{z}} = 1$. Finally, a hard partitioning scheme is adopted, meaning that each zone component can only be a member of a single cluster. Before the clustering procedure starts and no zone components have been grouped, it holds that $\mathbf{G} = \mathbf{I}$, with \mathbf{I} being the identify matrix, with dimensions implicitly determined by its context. Therefore, in general, we can state that, regardless of the chosen grouping, it holds that $\mathbf{G} \geq \mathbf{I}$. This constraint ensures that each zone component belongs to at least one group.

Other useful properties of \mathbf{G} are the fact that the *reduced row echelon form* of \mathbf{G} , denoted $\text{rref}(\mathbf{G})$, has a meaningful interpretation. Given \mathbf{G} is an indicator matrix, the fact that $\mathbf{G} = \mathbf{G}^T$, and all zone components are attributed to a single cluster; $\text{rref}(\mathbf{G})$ can be used to elegantly extract and identify the clustered zones. In the example of Figure 10.1, $\text{rref}(\mathbf{G})$ is

obtained by subtracting the first row from the second and the third row from the fourth, and then swapping the second and third row. By definition, reduced row echelon form results in a matrix where zero rows are placed below non-zero rows, each non-zero row leads with a 1, and the leading entry of a row is in a column that is more to the right than in the preceding row. In this case, this results in two non-zero rows that by definition are linearly independent and denote the two resulting clusters of zone components. The number of clusters conveniently corresponds with the *rank*, denoted $\text{rk}(\cdot)$, of the original grouping matrix, i.e. $\text{rk}(\mathbf{G}) = 2$. In other words, the number of clusters of zone components \hat{Z} conditional on grouping \mathbf{G} is obtained via $\hat{Z} = \text{rk}(\mathbf{G})$.

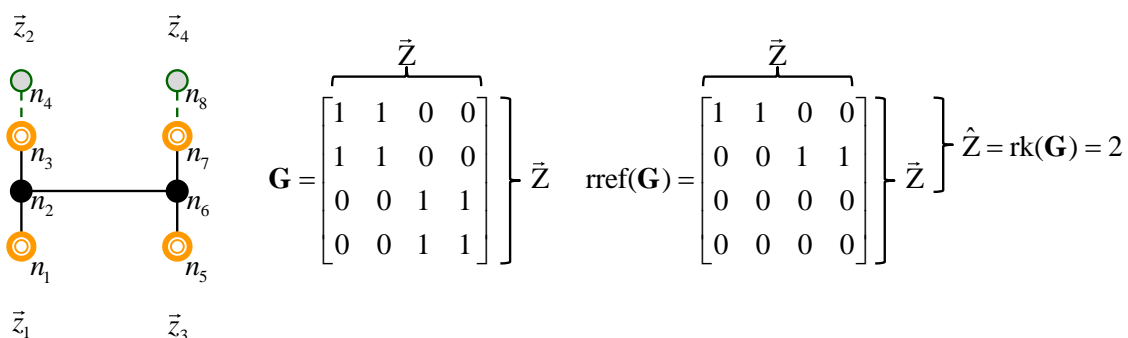


Figure 10.1: Identifying nodes internal to a cluster based on zone components and grouping.

10.3 Zonal clustering objective function

As mentioned, we consider two explicit objectives. The first minimises the total network distortion in connectoid access/egress costs via δ^- , δ^+ , respectively. The second objective aims to satisfy a minimum number of trips in order to achieve a particular zonal granularity. These two objectives are contradictory since the former is optimal when not clustering at all, while the latter likely must cluster zone components to meet the minimum trip requirements.

The minimum demand objective could alternatively have been modelled as a hard constraint. However, this is not possible for the following reasons: the desired zone demand cannot be expected to be met exactly for every zone. As a result the constraint has to be formulated as an inequality. In a setting where a zone's travel demand is forced to be less or equal than the desired demand there is no more incentive to create clusters, due to the connectoid cost distortion objective. Conversely, adopting a greater than relation, causes situations where no solution can be found, because, as we will see, some zone components are not allowed to cluster. In such cases, the entire problem becomes infeasible. Hence, the desired demand is modelled as a soft constraint by including it in the objective function.

10.3.1 Cluster based connectoid cost

Using the changes in connectoid costs as a measure for information loss is not perfect. It does for example not allow to account for any interaction effects between zones. Yet, computationally, it is simply infeasible to consider such interactions comprehensively without consuming an exorbitant amount of computation time, because it would involve performing assignments for each of the tested cluster combinations. We therefore rely on a simplified, but

in our view representative, proxy by considering the connectoid cost distortion (or lack thereof) that arises from adopting a particular clustering.

The original connectoid costs are computed based on the premise that only nodes internal to its zone component are considered. However, when the zone expands due to clustering, the connectoid's catchment increases; all nodes internal to the cluster become eligible to utilise the connectoid. This distorts its original cost estimate. This distorted cost is argued to be less accurate due to the larger area and the lesser likelihood of stable internal travel times. While this is unfortunate, it also provides the opportunity to quantify the magnitude of information loss suffered. Consider the same example as before (Figure 10.1) where four zone components are clustered. In this clustering, nodes n_{1-4} are internal to cluster \hat{z}_1 , where node n_2 is a boundary node between \bar{z}_1 and \bar{z}_2 . Let us assume all links have a travel time cost of 1 and we have weights of $\vec{w}^{\bar{z}_1} = [1, \frac{1}{2}, 0, 0, 0, 0, 0, 0]$, $\vec{w}^{\bar{z}_2} = [0, \frac{1}{2}, 1, 1, 0, 0, 0, 0]$. In that case connectoid n_1 , in the absence of any clustering, holds a connectoid egress cost of $\vec{H}_{\bar{z}_1 n_1}^+ = \frac{2}{3}(1 \cdot 0 + \frac{1}{2} \cdot 1) = \frac{1}{3}$. However, under clustering \mathbf{G} , nodes n_3, n_4 become eligible internal nodes as well. Adopting the same node to node demands, but considering all node weights of zone components within the same cluster via $\hat{w}^{\hat{z}_1} = [1, 1, 1, 1, 0, 0, 0, 0]$, where we observe that the previously separate boundary node weights have been merged because they now reside in the same cluster. The cluster based connectoid egress cost for n_1 is then found in an identical fashion as before, only now based on cluster weights such that $\hat{H}_{\hat{z}_1 n_1}^+ = \frac{1}{4}(0 \cdot 1 + 1 \cdot 1 + 2 \cdot 1 + 3 \cdot 1) = 1\frac{1}{2}$. Since the total weight of the cluster increases compared to the original zone component, clustering zone components impacts on the resulting connectoid cost, as is to be expected.

Let us now formalise the construction of the cluster based connectoid costs $\hat{\mathbf{H}}^+, \hat{\mathbf{H}}^- \in \mathbb{R}^{\bar{Z} \times N}$, respectively, following the exact same reasoning as discussed when constructing their zone component counterparts in Equations (9.21) and (9.22), resulting in:

$$\hat{H}_{\bar{z}n}^+ = \bar{N}_{\bar{z}n}^+ \left(\frac{1}{\mathbf{1}^T \hat{\mathbf{w}}^{\bar{z}}} \sum_{n'=1}^N \hat{w}_{n'}^{\bar{z}} H_{n'n} \right), \quad (10.2)$$

$$\hat{H}_{\bar{z}n}^- = \bar{N}_{\bar{z}n}^- \left(\frac{1}{\mathbf{1}^T \hat{\mathbf{w}}^{\bar{z}}} \sum_{n'=1}^N \hat{w}_{n'}^{\bar{z}} H_{nn'} \right), \quad (10.3)$$

where we replace the zone component based weights $\vec{w}^{\bar{z}}$ with the cluster based counterpart denoted $\hat{\mathbf{w}}^{\bar{z}} \in \mathbb{R}_+^{N \times 1}$ and which are constructed like the following:

$$\hat{w}_n^{\bar{z}} = \sum_{\bar{z}'=1}^{\bar{Z}} G_{\bar{z}\bar{z}'} \vec{w}_n^{\bar{z}'}. \quad (10.4)$$

Note that deliberately choose to place the connectoid costs on their original zone component in $\hat{\mathbf{H}}^+, \hat{\mathbf{H}}^-$, so that we can make quick comparisons between the original zone component results and the newly obtained cluster results, see Figure 10.2 for an illustrative example.

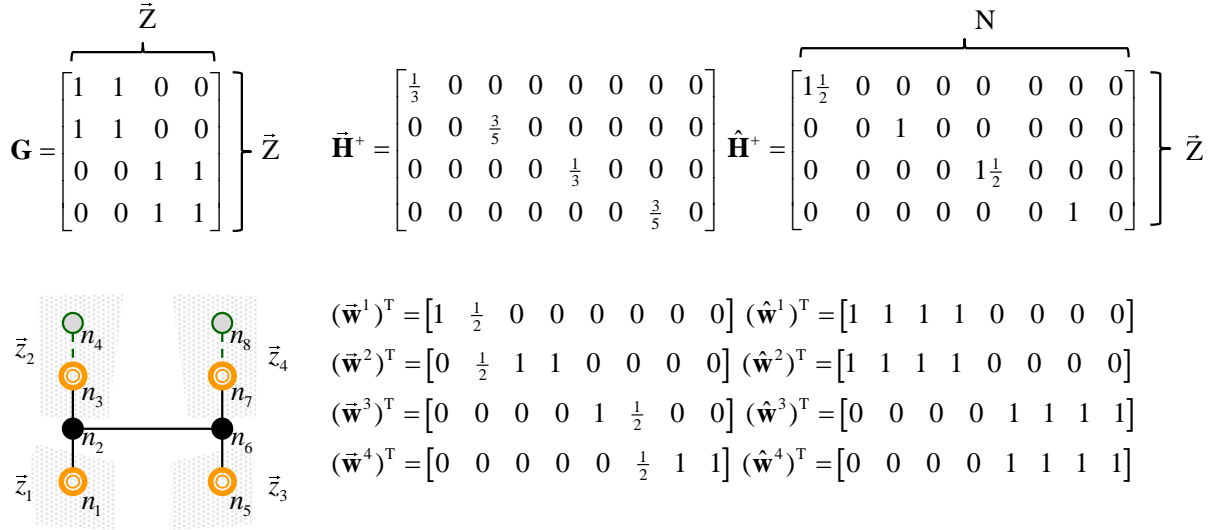


Figure 10.2: Example of connectoid cost affected by clustering

10.3.2 Measuring connectoid cost distortion

The distortion of the connectoid cost is defined by a measure of difference between the original zone component connectoid costs and their cluster based counterparts. In its simplest form this could be an unweighted absolute difference measure, simply comparing $|\bar{H}_{zn}^+ - \hat{H}_{zn}^+|, \forall(\bar{z}, n)$. However, we choose to avoid comparing absolute connectoid cost differences because the number of trips using a connectoid greatly impacts on the significance of the found (absolute) cost difference. A small distortion experienced by many trips might be more problematic than a high distortion experienced by few trips. Also, we cannot assume that the usage of a connectoid remains identical once it has been clustered. When empirical or other data sources are available they can also be used to estimate this usage directly.

Let us denote the expected egress/access connectoid usage for zone components – in terms of demand (veh) – via $\bar{\mathbf{F}}^+, \bar{\mathbf{F}}^- \in \mathbb{R}_+^{\bar{Z} \times N}$, respectively. If we then also construct this same expected usage conditional on some clustering \mathbf{G} , and denoted via $\hat{\mathbf{F}}^+, \hat{\mathbf{F}}^- \in \mathbb{R}_+^{\bar{Z} \times N}$, respectively, the weighted connectoid egress cost distortion $\bar{\delta}_{\bar{z}}^+$ (veh.h) for zone component \bar{z} can be formulated as follows:

$$\bar{\delta}_{\bar{z}}^+ = \sum_{n=1}^N |(\bar{H}_{zn}^+ \circ \bar{F}_{zn}^+) - (\hat{H}_{zn}^+ \circ \hat{F}_{zn}^+)|, \quad (10.5)$$

where $\bar{H}_{zn}^+ \circ \bar{F}_{zn}^+$ is only non-zero when n is a connectoid in zone component \bar{z} and its value represents the total weighted contribution of this zone component's egress cost. This is then compared, by taking the absolute difference, to this same connectoids weighted contribution under a given clustering \mathbf{G} . The summation across all connectoids in the zone component then yields the zone component's total egress cost distortion. We obtain the access cost distortion $\bar{\delta}_{\bar{z}}^-$ in a similar fashion via:

$$\bar{\delta}_{\bar{z}}^- = \sum_{n=1}^N |(\bar{H}_{zn}^- \circ \bar{F}_{zn}^-) - (\hat{H}_{zn}^- \circ \hat{F}_{zn}^-)|, \quad (10.6)$$

which then translates to the network wide respective connectoid distortions δ^+, δ^- , (veh.h) via:

$$\delta^+ = \sum_{\bar{z}=1}^{\bar{Z}} \bar{\delta}_{\bar{z}}^+, \quad (10.7)$$

$$\delta^- = \sum_{\bar{z}=1}^{\bar{Z}} \bar{\delta}_{\bar{z}}^-. \quad (10.8)$$

10.3.3 Expected connectoid usage

So far, we have not yet formalised how to obtain the expected connectoid usage for the zone components nor for the clusters. To do so, we settle for the assumption of a uniform distribution of trips across the available connectoids - in either the zone component or the cluster. We adopt this fairly basic approach because we mainly want to demonstrate suitability of the method itself.

Regarding the zone component based construction of $\bar{\mathbf{F}}^+, \bar{\mathbf{F}}^-$, we simply take a zone component's production/attraction and divide it by the number of connectoids in the zone component. The result is then attributed to each connectoid in this zone component via:

$$\bar{F}_{\bar{z}n}^+ = \bar{N}_{\bar{z}n}^+ \left(\frac{\bar{d}_{\bar{z}}^+}{\bar{\mathbf{N}}_{\bar{z}}^+ \mathbf{1}} \right), \quad (10.9)$$

$$\bar{F}_{\bar{z}n}^- = \bar{N}_{\bar{z}n}^- \left(\frac{\bar{d}_{\bar{z}}^-}{\bar{\mathbf{N}}_{\bar{z}}^- \mathbf{1}} \right), \quad (10.10)$$

with $\bar{\mathbf{d}}^+, \bar{\mathbf{d}}^- \in \mathbb{R}_+^{\bar{Z} \times 1}$ holding the total production/attraction of each zone component \bar{z} , such that:

$$\bar{d}_{\bar{z}}^+ = \sum_{\bar{z}'=1}^{\bar{Z}} \mathbf{1}^T \bar{\mathbf{D}}^{\bar{z}\bar{z}'} \mathbf{1}, \quad (10.11)$$

$$\bar{d}_{\bar{z}}^- = \sum_{\bar{z}'=1}^{\bar{Z}} \mathbf{1}^T \bar{\mathbf{D}}^{\bar{z}'\bar{z}} \mathbf{1}, \quad (10.12)$$

where the zone component specific node demand matrix $\bar{\mathbf{D}}^{\bar{z}\bar{z}'}$ is the result of a simple transferral of the disaggregate node demand matrix, given that the departure/arrival node under consideration resides in both the original origin/destination zone as well as in the zone component origin/destination, respectively, or, more formally:

$$\bar{D}_{mn'}^{\bar{z}\bar{z}'} = \bar{N}_{\bar{z}n} \bar{N}_{\bar{z}'n'} \sum_{z=1}^Z N_{zn} \sum_{z'=1}^Z N_{z'n'} \bar{\mathbf{D}}_{mn'}^{*zz'}, \quad (10.13)$$

We now continue with formulating the cluster based expected connectoid usage $\hat{\mathbf{F}}^+, \hat{\mathbf{F}}^-$. This requires some additional bookkeeping, because one first needs to identify which nodes and connectoids are internal to each cluster. The nodes internal to each cluster are denoted by $\hat{\mathbf{N}} \in \mathbb{F}_2^{\bar{Z} \times N}$ and formalised via:

$$\hat{N}_{\bar{z}n} = \begin{cases} 1, & \text{if } \exists \bar{z}' : G_{\bar{z}\bar{z}'} \bar{N}_{\bar{z}n}^-, \\ 0, & \text{otherwise,} \end{cases} \quad (10.14)$$

with $\bar{z}' \in \{1, \dots, \bar{Z}\}$. This approach is graphically illustrated in Figure 10.3. Observe that, again, internal cluster nodes remain defined on a per zone component basis, so we can compare the original pre-clustering situation, with the impact of the chosen clustering later on.

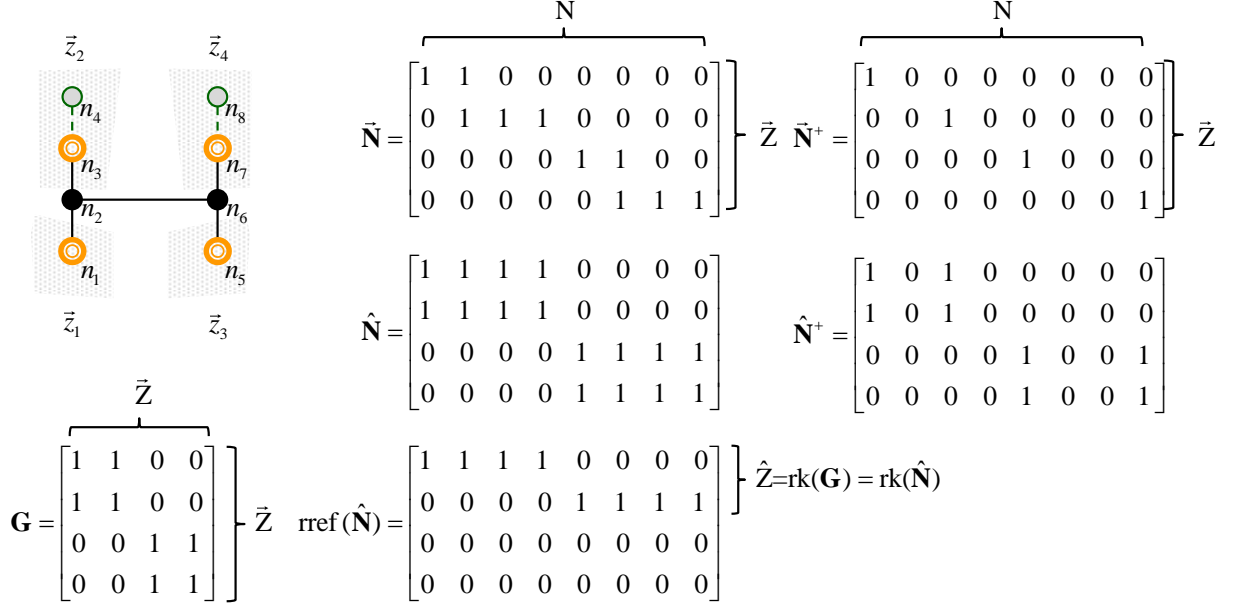


Figure 10.3: Identifying internal nodes to clusters based on original zone components.

The egress / access connectoid membership matrices, denoted $\hat{\mathbf{N}}^+, \hat{\mathbf{N}}^- \in \mathbb{F}_2^{\bar{Z} \times N}$, respectively, are obtained similarly, like the following:

$$\hat{N}_{\bar{z}n}^+ = \begin{cases} 1, & \text{if } \exists \bar{z}' : G_{\bar{z}\bar{z}'} \bar{N}_{\bar{z}n}^+, \\ 0, & \text{otherwise,} \end{cases} \quad (10.15)$$

$$\hat{N}_{\bar{z}n}^- = \begin{cases} 1, & \text{if } \exists \bar{z}' : G_{\bar{z}\bar{z}'} \bar{N}_{\bar{z}n}^-, \\ 0, & \text{otherwise,} \end{cases} \quad (10.16)$$

Although it is not needed when comparing costs, we do point out that if one takes the reduced row echelon form of each of these matrices, such as for example $\text{rref}(\hat{\mathbf{N}})$, the node (and connectoid) memberships of each cluster $\hat{z} \in \{1, \dots, \hat{Z}\}$ are obtained.

Knowing the nodes and connectoids internal to each cluster, we can formalise the assumed *cluster based* connectoid usage $\hat{\mathbf{F}}^+, \hat{\mathbf{F}}^-$, respectively. This is done in an identical albeit cluster based fashion compared to Equations (10.5) and (10.6), yielding:

$$\hat{F}_{\bar{z}n}^+ = \hat{N}_{\bar{z}n}^+ \begin{pmatrix} \hat{d}_{\bar{z}}^+ \\ \hat{\mathbf{N}}_{\bar{z}}^+ \cdot \mathbf{1} \end{pmatrix}, \quad (10.17)$$

$$\hat{F}_{\hat{z}n}^- = \hat{N}_{\hat{z}n}^- \left(\frac{\hat{d}_{\hat{z}}^-}{\hat{N}_{\hat{z}}^+ \mathbf{1}} \right), \quad (10.18)$$

with $\hat{\mathbf{d}}^+, \hat{\mathbf{d}}^- \in \mathbb{R}_+^{\hat{Z} \times 1}$ holding the total production/attraction of the cluster that each zone component \hat{z} resides in, such that:

$$\hat{d}_{\hat{z}}^+ = \sum_{\hat{z}'=1}^{\hat{Z}} G_{\hat{z}\hat{z}'} (\mathbf{1}^T \bar{\mathbf{D}}^{\hat{z}\hat{z}'} \mathbf{1}), \quad (10.19)$$

$$\bar{d}_{\hat{z}}^- = \sum_{\hat{z}'=1}^{\hat{Z}} G_{\hat{z}\hat{z}'} (\mathbf{1}^T \bar{\mathbf{D}}^{\hat{z}'\hat{z}} \mathbf{1}). \quad (10.20)$$

This concludes the construction of the connectoid cost distortion portion of the objective function and the formulation of the components involved. We now proceed with the second part of the objective function, the penalty imposed for clustered zone components that do not meet the minimum number of trips.

10.3.4 Missing demand penalty

The second component of the objective function drives the granularity of the resulting zoning system. Recall that the desired number of trips for each to be constructed zone is assumed to be user defined and given by d^{\min} . Whenever a zone, due to some constraints, cannot meet this desired number of trips, i.e. the final zone remains too disaggregate compared to the expected level of detail, a penalty is imposed. The severity of this penalty depends on: (i) the choice of d^{\min} , (ii) the found number of missing trips per cluster, denoted $\hat{\mathbf{d}}^{\text{missing}} \in \mathbb{R}_+^{\hat{Z} \times 1}$ (veh), (iii) the, to be estimated, per trip penalty \hat{h}^{\max} (h) for not meeting the desired zonal demand under the given clustering. Once this information is available, the penalty $\hat{\delta}_{\hat{z}}^{d^{\min}}$ for cluster \hat{z} is determined as follows:

$$\hat{\delta}_{\hat{z}}^{d^{\min}} = \hat{h}^{\max} \hat{d}_{\hat{z}}^{\text{missing}}, \quad \forall \hat{z} \in \{1, \dots, \hat{Z}\}, \quad (10.21)$$

where one can think of \hat{h}^{\max} as the “worst case” zonal access/egress travel time attributed to each trip less than d^{\min} in the cluster under consideration. Equation (10.21) therefore serves as a surrogate connectoid cost to “complete” the connectoid cost distortion estimate for zones with too little trips. Let us also define a per zone component version of $\hat{\delta}_{\hat{z}}^{d^{\min}}$, denoted $\bar{\delta}_{\hat{z}}^{d^{\min}}$. This is only done to aid the formulation of the solution algorithm and bounds discussed in Section 10.8 and is given by:

$$\bar{\delta}_{\hat{z}}^{d^{\min}} = \sum_{\hat{z}=1}^{\hat{Z}} \frac{\text{rref}(\mathbf{G})_{\hat{z}\hat{z}} \hat{\delta}_{\hat{z}}^{d^{\min}}}{\text{rref}(\mathbf{G})_{\hat{z}\cdot} \mathbf{1}}, \quad (10.22)$$

where each zone component in \hat{z} is assigned an equal share of the $\hat{\delta}_{\hat{z}}^{d^{\min}}$. The network wide penalty is then obtained by the sum over all zone components:

$$\delta^{d^{\min}} = \sum_{\bar{z}} \bar{\delta}_{\bar{z}}^{d^{\min}}. \quad (10.23)$$

Let us now construct the number of missing trips $\hat{\mathbf{d}}^{\text{missing}}$, which depends on threshold d^{\min} . We deliberately only penalise missing trips because whenever a zone meets d^{\min} , the penalty of adding more zone components is already captured by the increased distortion of connectoid costs. When we would, in addition, also penalise the excess demand, it becomes relatively too attractive to not meet d^{\min} , defying the purpose of our exercise. To obtain $\hat{\mathbf{d}}^{\text{missing}}$, we first collect the total number of zonal trips $\hat{\mathbf{d}}^{\text{total}} \in \mathbb{R}_+^{\bar{Z} \times 1}$, conditional on clustering \mathbf{G} via:

$$\hat{d}_{\bar{z}}^{\text{total}} = \hat{d}_{\bar{z}}^+ + \hat{d}_{\bar{z}}^-, \quad \forall \bar{z} : \text{rref}(\mathbf{G})_{\bar{z}\bar{z}} = 1. \quad (10.24)$$

where we take the sum of the cluster's production and attraction. We accept a slight inconsistency in notation here, but because *cluster* based productions/attractions are duplicated across its *zone components*, they are indeed compatible with the cluster based total on the left hand side of this formulation. The number of missing trips for each cluster \hat{z} is then simply obtained via:

$$\hat{d}_{\hat{z}}^{\text{missing}} = \max\{0, d^{\min} - \hat{d}_{\hat{z}}^{\text{total}}\}, \quad \hat{z} \in \{1, \dots, \hat{Z}\}. \quad (10.25)$$

Finally, the value of per trip penalty h^{\max} in Equation (10.21) requires estimation. We postpone the discussion on how one can estimate h^{\max} to Section 10.10 (it relies on the not yet discussed branch-and-bound solution scheme). At this stage it suffices to say that h^{\max} needs to be high enough such that it is attractive to form clusters that meet the minimum demand d^{\min} , but at the same time h^{\max} must be low enough to avoid accepting unreasonable zone clusterings just to meet the minimum demand objective.

10.4 Zoning system criteria and constraints

The objective function of our optimisation problem considers the supply side cost aspect by minimising its distortion while aggregating zone components. It also matches the zoning system to the desired level of detail. What it does not do, is guarantee that these objectives are in line with the traditional demand side criteria for zonal design as discussed in Chapter 7. Because we disaggregated the original zoning system we must be careful to not ignore the original effort and information captured in this original zoning structure. Table 10.1 outlines the demand side criteria as we originally discussed them. Here, we relate them to the constraints imposed in our optimisation problem formulation in order to be able to satisfy these criteria as much as deemed necessary when refining the original zoning.

As can be observed from Table 10.1, some demand side criteria have already been addressed, either when constructing the zone components, when we delineated them through boundary nodes, by considering connectoid cost in our objective function, or by imposing a minimum demand. The proposed constraints discussed in the upcoming sections serve to address the following remaining criteria: within zone homogeneity, minimising intrazonal trips, and yielding sensible zonal area (shapes). They also serve the practical purpose of reducing the

solution space and in turn increasing the likelihood of finding solutions in a reasonable amount of time.

Table 10.1: Zoning system design criteria in relation to proposed methodology

| Criteria | Conceptually | Practically | Constraint | Section |
|---|--|---|-----------------|-------------------------------|
| Within zone data homogeneity | Land use and socio-economic similarity | Similarity measure | Hard constraint | Section 10.5.2 |
| Between zone data homogeneity | Create zones with similar trip numbers | Minimum demand | Soft constraint | Section 10.3.4 |
| Minimise intrazonal trips | Maximise land use differences between clusters | Similarity measure | Hard constraint | Section 10.5.2 |
| Adopt census boundaries | Captured in original zone areas | Zone component delineation | - | Section 9.6.1 |
| Adopt physical, political and historical boundaries when sensible | Captured in original zone areas | Zone component delineation | - | Section 9.6.1 |
| Convex area shape, i.e. no “holes” | Verify cluster shape | Contiguity check, Connectoid cost | Hard constraint | Section 10.6, Section 10.3.1 |
| Within zone connectivity | Internal travel time stability constraint | Zone component delineation, Connectoid cost | - | Section 9.6.1, Section 10.3.1 |

10.5 Instance-level constraints

Instance-level constraints are defined between two zone components. These constraints are easy to verify and can be constructed beforehand, independent of the chosen clustering. Recall from Chapter 7 that the most common instance-level constraints are must-link and can-link constraints. We only consider can-link constraints because the must-link condition has been dealt with already; each zone component holds the nodes and links that were considered must-link. We denote *pair-wise can-link options* via indicator matrix $\mathbf{K} \in \mathbb{F}_2^{\bar{Z} \times \bar{Z}}$. Can-link options embed two constraints; the first concerns a between component *closeness* constraint, while the second one imposes a *similarity* measure.

10.5.1 Between component travel time constraint

While the objective function makes it unattractive to cluster zone components that reside far apart via the distortion of connectoid costs, it does not forbid it. As a result the solution space, without additional constraints, becomes too large to explore efficiently. By imposing a hard constraint on the maximum - between zone component - travel time, denoted by $\tau^{\max}(\mathbf{h})$, unlikely clustering candidates can be removed from the search space altogether.

The travel time between two components is captured by $\vec{\mathcal{T}}^{\min} \in \mathbb{R}_+^{\bar{Z} \times \bar{Z}}$. this travel time is found by taking the minimum over all shortest paths between any two connectoids of the zone component pair. We indicate two zone components to be can-link options of each other as long as $\vec{\mathcal{T}}_{\vec{z}\vec{z}}^{\min} \leq \tau^{\max}$, $\vec{z} \in \{1, \dots, \bar{Z}\}$, otherwise they are marked as cannot-link. Note that this constraint

has no impact on the final result as long as τ^{\max} is chosen such that it does not preclude sensible clusterings.

10.5.2 Between component similarity constraint

The second constraint incorporated in the can-link matrix relates to the aforementioned demand-side criteria, namely the criterion of *homogeneous land use* and the *minimisation of intrazonal trips*. To minimise intrazonal trips, as many trips between zone components should remain interzonal after clustering. Similarly, homogeneous land use implies a bias towards particular types of trips in that area, for example residential areas mainly have trip productions in morning peak, while commercial areas predominantly contain trip attractions in this same time period. By not clustering them, we indirectly minimise intrazonal trips since one can expect most trips to occur between areas with different production/attraction patterns. Socio-economic data is a driver for the estimation for the number of trips and where they go. This is therefore already mostly captured in the original trip matrix.

We aim to comply with these demand side criteria by formulating a similarity constraint, where similarity is defined in terms of *trip similarity* (rather than demand side data similarity). We argue this is a reasonable proxy since the original demand side criteria, whatever they may have been, culminated in the creation of the original trip demand matrix. Zones with similar trip patterns are considered more attractive, or at least less unattractive, to be clustered because they are more likely to minimise intrazonal trips, exhibit homogeneous land use, as well as socio-economic characteristics, which all come down to the same objective of minimising the variance in the underlying data, making the aggregate results relatively more representative.

Trip productions and attractions across an entire day are virtually always the same on a per zone basis, i.e. people who leave for work, return to home as well. That said, for a more limited time period, such as the morning or evening peak period, trip productions and attractions are asymmetric and are a decent proxy for the similarity of travel patterns across (neighbouring) zones. We therefore choose to measure trip similarity by identifying differences across the zone component's productions and attractions (assuming our demand is peak period based). The result of this metric is captured by a ratio and is stored in similarity matrix $\bar{\mathbf{S}} \in [0,1]^{\bar{Z} \times \bar{Z}}$. This allows for quick verification, while still picking up (aggregate) differences in trip patterns. More sophisticated similarity measures are of course possible, but are left for future research.

The (dis)similarity between two zone component's productions and attractions has nothing to do with absolute differences in trips, only the pattern matters, not the magnitude. Hence, similarity is captured by comparing relative differences in productions and attractions. Note that we cannot take a simple ratio of the two values because this does not yield a value that changes linearly. Instead we define the ratio vector $\bar{\mathbf{d}}^{\pm} \in [0,1]^{\bar{Z} \times 1}$ via:

$$\bar{d}_{\bar{z}}^{\pm} = \frac{1}{2} \left(\frac{\bar{d}_{\bar{z}}^{+} - \bar{d}_{\bar{z}}^{-}}{\bar{d}_{\bar{z}}^{+} + \bar{d}_{\bar{z}}^{-}} + 1 \right), \quad (10.26)$$

Where the inner part ranges from $[-1,1]$, with -1 representing only attractions and $+1$ only productions. We translate to non-negative values, i.e. $[0,2]$, and then normalise to yield values

in the range $[0,1]$. We denote the (dis)similarity between two zone components via $\vec{E} \in \mathbb{R}_+^{\vec{Z} \times \vec{Z}}$, where two zone components \vec{z}, \vec{z}' are considered equivalent when $\vec{E}_{\vec{z}\vec{z}'} = 0$. In general we define $\vec{E}_{\vec{z}\vec{z}'}$ via:

$$\vec{E}_{\vec{z}\vec{z}'} = \left| \vec{d}_{\vec{z}}^{\pm} - \vec{d}_{\vec{z}'}^{\pm} \right|, \quad \vec{z}, \vec{z}' \in \{1, \dots, \vec{Z}\}. \quad (10.27)$$

A threshold on the maximum acceptable dissimilarity is then required to determine if zone components can-link or not, which we denote by e^{\max} . Assuming we somehow estimated e^{\max} , we can construct the can-link matrix (including the maximum travel time constraint as well), via:

$$K_{\vec{z}\vec{z}'} = \begin{cases} 1, & \text{if } \vec{\mathcal{T}}_{\vec{z}\vec{z}'}^{\min} \leq \tau^{\max} \text{ and } \vec{E}_{\vec{z}\vec{z}'} \leq e^{\max}, \\ 0, & \text{otherwise,} \end{cases} \quad (10.28)$$

with $\vec{z}, \vec{z}' \in \{1, \dots, \vec{Z}\}$. A schematic example of zone components, hypothetical between-zone component travel times, and productions/attractions is given in Figure 10.4. Depending on the thresholds we obtain different can-link matrices. Observe that if we would choose $\tau^{\max} = 3$, then \vec{z}_6 has no can-link options anymore. Similarly, when we choose $e^{\max} = 0.2$ zone component \vec{z}_5 is too dissimilar to be allowed to cluster with any other zone component.

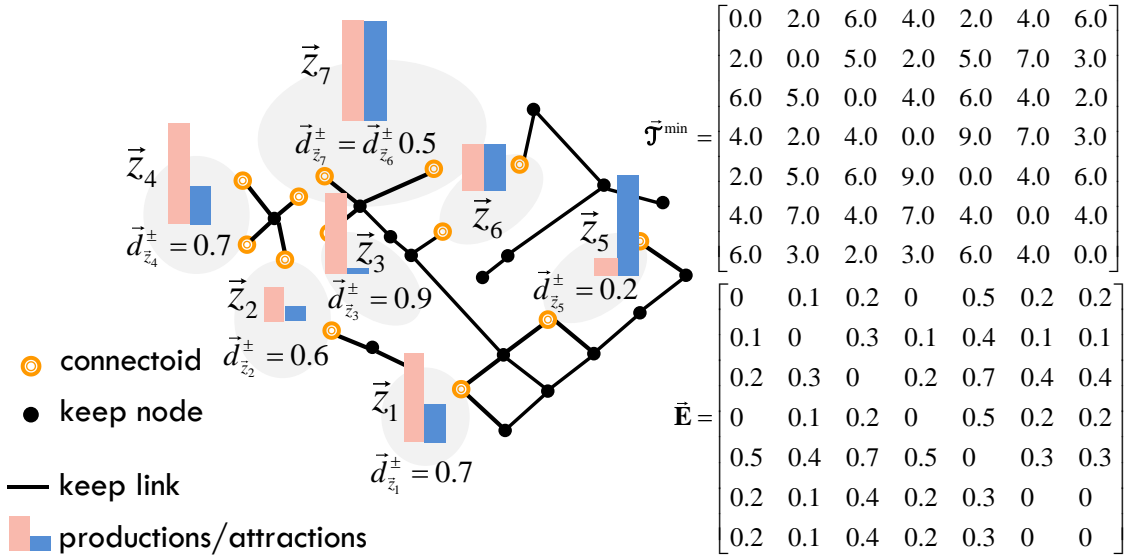


Figure 10.4: Example of how instance-level zone component characteristics influence can-link options.

10.6 Cluster-level constraints

Cluster-level can-link constraints are costlier to verify than instance-level constraints because verification requires information on the cluster, information that is only available during the clustering procedure itself. We found that we can differentiate between two types of cluster-level constraints, which we term, for the lack of an established terminology as: *derived cluster-level constraints* and *non-derived cluster-level constraints*.

Derived cluster-level constraints follow directly from instance-level constraints. They are based on the premise that every pair-wise combination of zone components within the cluster must be marked can-link for the cluster to be regarded as valid. The constraint is termed “derived” because it can be constructed based on instance-level constraints. We formulate two derived cluster-level constraints by promoting both the similarity and closeness measure, captured in can-link matrix \mathbf{K} , to the cluster-level. This means that all zone components within the cluster must be able to reach each other within τ^{\max} , as well as being sufficiently similar to all other zone components in the cluster. The derived cluster-level constraints are elegantly enforced via $\mathbf{K} \geq \mathbf{G}$, i.e. every zone component that is in the same cluster with another zone component must be marked as can-link.

Non-derived cluster-level constraints are slightly more complicated, because they cannot be constructed from pair-wise constraints. In our case, there is only one such constraint, namely a *contiguity constraint*. It relates to the demand side criteria to construct “logical” shapes and ensure connectivity within zones. A cluster that satisfies the contiguity constraint is said to be *contiguous*. In a contiguous cluster, any zone component within the cluster can reach any other zone component in the same cluster by only moving from one neighbouring zone component (in the cluster) to the next, assuming the point of departure is itself. Observe that we cannot capture this constraint in a pair-wise can-link constraint, because it depends on the cluster what other zone components must be reachable. To formalise the contiguity constraint further, we first need a definition of what it means to be a *neighbour*.

10.6.1 Neighbours

Being a *neighbour* is captured by zone component adjacency matrix $\vec{\mathcal{A}} \in \mathbb{F}_2^{\vec{z} \times \vec{z}}$, which is defined to be symmetric such that $\vec{\mathcal{A}} = \vec{\mathcal{A}}^T$. Given our supply side perspective, we utilise the keep link network to establish what constitutes a neighbour, which we define in two steps:

Definition 10.1: Direct adjacency

Let \vec{z} be a zone component. Then, zone component \vec{z}' is *directly adjacent* to \vec{z} when \vec{z}' has a connectoid adjacent to a keep node n , such that a path through the keep network can be constructed between a connectoid of \vec{z} and keep node n , without this path traversing any other keep nodes being adjacent to a zone component other than \vec{z} ¹².

The definition of *direct adjacency* does not yet ensure *symmetry*. Consider Figure 10.5, zone component \vec{z}_6 is directly adjacent to \vec{z}_1 and \vec{z}_5 , but at the same time \vec{z}_1 and \vec{z}_5 are only directly adjacent to each other. By making all adjacency relationships bi-directional (Definition 7.2), we guarantee the symmetry one expects when being classified as a neighbour.

Definition 10.2: Zone component neighbour

Let \vec{z}, \vec{z}' be zone components. When \vec{z} is directly adjacent to \vec{z}' , and/or when \vec{z}' is directly adjacent to \vec{z} , then \vec{z}, \vec{z}' are considered *neighbours*.

¹² A node based algorithm is employed to identify direct adjacencies, where each connectoid’s closest keep node acts as a starting point, i.e. marked current. If other zone components are adjacent to the current node, they are marked as directly adjacent and the search stops, otherwise the current node’s adjacent keep nodes are marked as current and the search continues.

So far, adjacency matrix $\bar{\mathcal{A}}$ is defined without considering clustering. At the same time, only neighbours internal to a cluster are of interest when verifying contiguity (within the cluster). Therefore, we make $\bar{\mathcal{A}}$ conditional on clustering \mathbf{G} by constructing $\hat{\mathcal{A}} \in \mathbb{F}_2^{\mathcal{Z} \times \mathcal{Z}}$ via:

$$\hat{\mathcal{A}} = \bar{\mathcal{A}} \circ \mathbf{G}, \tag{10.29}$$

where $\hat{\mathcal{A}}$ only retains neighbours within the same cluster.

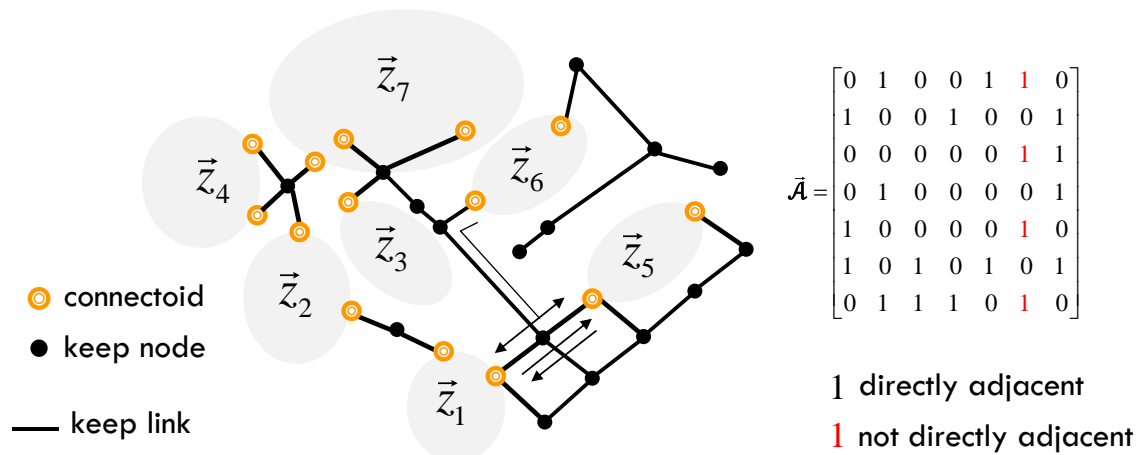


Figure 10.5: Zone component neighbours and how to ensure symmetry.

These cluster internal neighbours are referred to as *cluster-neighbours*. See Figure 10.6, for an impression of the different graphs that arise when identifying neighbours, and cluster neighbours, respectively.

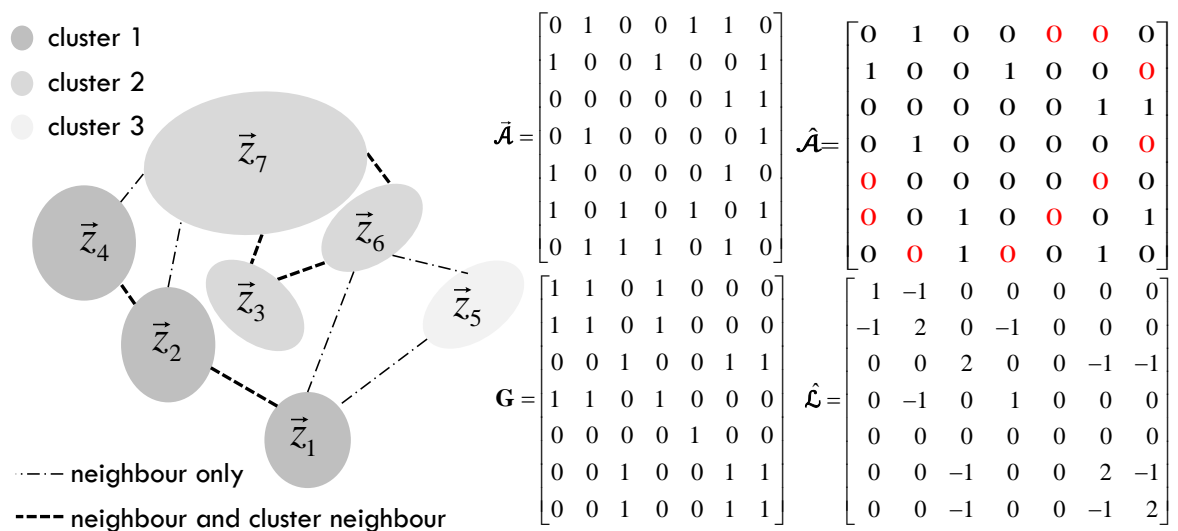


Figure 10.6: Example of difference between neighbours and cluster neighbours conditional on clustering \mathbf{G} .

10.6.2 Laplacian matrix and the number of connected components

To verify if a cluster is contiguous we make use of the Laplacian matrix, denoted $\hat{\mathcal{L}} \in \mathbb{Z}^{\bar{z} \times \bar{z}}$. The Laplacian matrix holds an alternative representation of $\hat{\mathcal{A}}$, by negating its values and, on the diagonal, place the *degree* (sum) of the *cluster-neighbours* of each zone component such that the sums of each row amount to exactly zero:

$$\hat{\mathcal{L}}_{\bar{z}\bar{z}'} = \begin{cases} \hat{\mathcal{A}}_{\bar{z}} \cdot \mathbf{1}, & \text{if } \bar{z} = \bar{z}', \\ -\hat{\mathcal{A}}_{\bar{z}\bar{z}'}, & \text{otherwise.} \end{cases} \quad (10.30)$$

The Laplacian of our example network is included in Figure 10.6. The Laplacian has many uses and is often adopted in conjunction with clustering related procedures. For example, in *spectral analysis* the Laplacian can be used to partition networks, see for example Bell et al. (2017), where the eigenvector related to the second smallest eigenvalue of $\hat{\mathcal{L}}$ is used to split a network in two disjoint clusters. Our objective however is not to partition a network, but instead verify if the existing clusters are contiguous. For this, the Laplacian also offers another useful and well known property; the *nullity*. A Laplacian's nullity reveals the number of connected components in its matrix.

The *nullity* is obtained by establishing the *dimensionality* of the *kernel* of the Laplacian, denoted as $\dim(\ker(\hat{\mathcal{L}}))$, where $\dim(\cdot)$ denotes the dimensionality function and $\ker(\cdot)$ denotes the function obtaining a basis of the kernel. When we know the number of connected components, we can demonstrate that this information can be used to verify if a cluster is contiguous.

To determine $\dim(\ker(\hat{\mathcal{L}}))$, we first “solve” the Laplacian as if it were a system of equations such that through $\hat{\mathcal{L}}\mathbf{v} = \mathbf{0}$, where $\mathbf{v} \in \mathbb{Z}^{\bar{z} \times 1}$ is any vector yielding the zero vector when multiplied with $\hat{\mathcal{L}}$. For our example network, we solve $\hat{\mathcal{L}}\mathbf{v} = \mathbf{0}$, by first rewriting the Laplacian to its reduced row echelon form, i.e. $\text{rref}(\hat{\mathcal{L}})\mathbf{v} = \mathbf{0}$, see Figure 10.7.

If we then write out the system of equations we find that cluster-neighbours are represented by dependencies that are transitively interlocked in the resulting equations. For example, \bar{z}_1 is a cluster-neighbour of \bar{z}_2 and \bar{z}_2 is a cluster-neighbour of \bar{z}_4 , when solving $\text{rref}(\hat{\mathcal{L}})\mathbf{v} = \mathbf{0}$, we therefore find; $\bar{z}_1 = \bar{z}_2, \bar{z}_2 = \bar{z}_4$, hence $\bar{z}_1 = \bar{z}_2 = \bar{z}_4$, asserting interdependency which we can interpret as being neighbour reachable and therefore forming a contiguous cluster. Alternatively, if a zone component has no cluster-neighbours, for example \bar{z}_5 , its column is a zero vector. Then a stand-alone “contiguous” cluster of just the one zone component results. As can be seen in Figure 10.7, each contiguous group of zone components results in a unique (non-zero) solution vector that solves $\hat{\mathcal{L}}\mathbf{v} = \mathbf{0}$. These vectors are known as a *basis of the kernel* of $\hat{\mathcal{L}}$, denoted $\ker(\hat{\mathcal{L}})$. Of course $\mathbf{v} = \mathbf{0}$ is by definition a solution, but it holds no information regarding contiguity and is therefore not considered. In the example, following this method, we find three contiguous groups of zone components because the cardinality of the kernel basis, i.e. the number of non-zero solution vectors, is three. This cardinality equates to the dimension of the kernel basis, denoted $\dim(\ker(\hat{\mathcal{L}}))$, and is referred to as the nullity.

$$\begin{array}{c}
\vec{z}_1 \quad \vec{z}_2 \quad \vec{z}_3 \quad \vec{z}_4 \quad \vec{z}_5 \quad \vec{z}_6 \quad \vec{z}_7 \\
\text{rref}(\hat{\mathcal{L}})\mathbf{v} = 0 \rightarrow \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{v} = 0
\end{array}$$

interdependencies enforced across rows

no cluster-neighbours for \vec{z}_5

$$\begin{array}{l}
\vec{z}_1 - \vec{z}_2 = 0, \\
\vec{z}_2 - \vec{z}_4 = 0, \\
\rightarrow \vec{z}_3 - 2\vec{z}_6 + \vec{z}_7 = 0, \\
\vec{z}_6 - \vec{z}_7 = 0, \\
\vec{z}_5 = 0,
\end{array}
\quad
\begin{array}{l}
\vec{z}_1 = \vec{z}_2 = \vec{z}_4, \\
\vec{z}_3 = \vec{z}_6 = \vec{z}_7, \rightarrow \vec{z}_4 \\
\vec{z}_5 = 0,
\end{array}$$

$$\begin{array}{cccc}
\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} & , \vec{z}_3 & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} & , \vec{z}_5 & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} & , & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
\mathbf{v}^1 & & \mathbf{v}^2 & & \mathbf{v}^3 & & \mathbf{v}^0
\end{array}$$

Figure 10.7: Basis of the kernel of the neighbour Laplacian conditional on clustering \mathbf{G} .

Knowing the nullity, we now formulate our contiguity constraint: given that each cluster in \mathbf{G} must be contiguous, each such cluster should yield exactly one contiguous connected component. Hence, the number of connected components in $\hat{\mathcal{L}}$ should exactly match the number of clusters in \mathbf{G} . Therefore our contiguity constraint is formulated as follows:

$$\dim(\ker(\hat{\mathcal{L}})) = \text{rk}(\mathbf{G}). \quad (10.31)$$

In case not all zone components in a cluster are contiguous, more than a single connected component is found for at least one cluster. In that case $\dim(\ker(\hat{\mathcal{L}})) > \text{rk}(\mathbf{G})$, violating the constraint. For example, if we would exclude the direct neighbour relation between \vec{z}_1 and \vec{z}_2 , while maintaining the same clustering, we would find $\dim(\ker(\hat{\mathcal{L}})) = 4$, while $\text{rk}(\mathbf{G}) = 3$. Hence, the contiguity constraint would be violated and \mathbf{G} is no longer a valid clustering.

In our optimisation problem we formulate non-derived cluster-level constraints in a more general fashion via $\mathbf{K}^{\mathbf{G}} \geq \mathbf{G}$. We therefore slightly rewrite Equation (10.31) to be compatible, although this change is only cosmetic:

$$\mathbf{K}^{\mathbf{G}} = \begin{cases} \mathbf{J}, & \text{if } \dim(\ker(\hat{\mathcal{L}})) = \text{rk}(\mathbf{G}), \\ \mathbf{I}, & \text{otherwise,} \end{cases} \quad (10.32)$$

where \mathbf{J} represents an all-ones matrix with context dependent dimensions. Observe that Equation (10.32) guarantees that $\mathbf{K}^{\mathbf{G}} \geq \mathbf{G}$ holds as long as $\dim(\ker(\hat{\mathcal{L}})) = \text{rk}(\mathbf{G})$, while otherwise it is violated. This concludes the constrained optimisation problem formulation.

10.7 Solving combinatorial problems

We proceed with solving our optimisation problem formulation by proposing a particular solution scheme. Clustering procedures effectively solve an underlying combinatorial problem and to get some insight in the issues around combinatorial problems, and clustering in particular, we first provide some relevant background information on this topic before proceeding to discuss our branch-and-bound solution scheme.

10.7.1 On clustering complexity

Due to the combinatorial nature of clustering problems, they are generally computationally costly problems to solve. For example, the underlying problem of the well-known k -means algorithm; finding a partitioning such that the sum of distances to the virtual k cluster centres is minimised, is proven to be NP-hard (Garey et al., 1982). Our underlying objective when constructing the zoning system is even more complex to solve. First, similar to k -means, our objective is measured in relation to individual data points, i.e. zone components, as well. Recall that we measure the connectoid cost distortion, which represents at least one, but often multiple values per zone component. Second, unlike k -means, we do not know the number of final clusters k beforehand, making our problem harder to solve from a combinatorial point of view.

In general, we can construct the number of possible combinations of partitioning b numbers in j partitions via a binomial tree \mathcal{B}_b , as depicted in Figure 10.8 in a recursive manner. The *depth* j of this tree determines how many edges one has traversed, while b is referred to as the *order* of the tree. A single possible partition, or cluster, is formed by grouping all vertex numbers encountered while traversing a path from the root of the tree to a particular depth. The number of unique paths, given some order b and depth j , reveals all possible clusters, ignoring permutations. Observe that with $j = 2$ and $b = 3$ we find $\{2,1\}, \{3,1\}, \{3,2\}$, respectively.

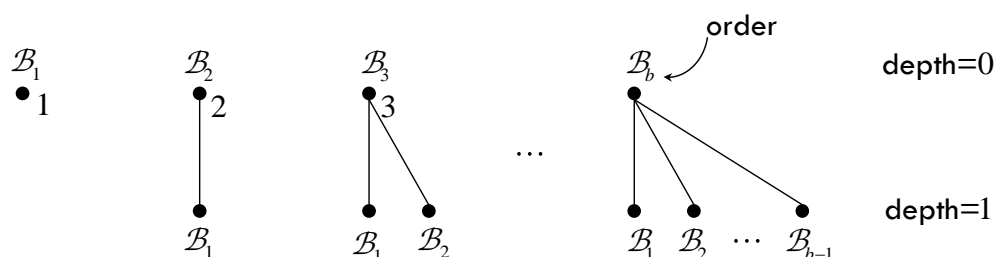


Figure 10.8: Binomial tree and unique single cluster solution space.

Binomial trees can be made compatible with our binary clustering matrix $\mathbf{G} \in \mathbb{F}_2^{\bar{z} \times \bar{z}}$ by replacing vertex numbers, i.e. the order, with an indicator vector, where the only non-zero entry is at the index equal to the order of the vertex. The cluster is found by summing the individual vertex vectors up to the desired depth, see Figure 10.9 for an example.

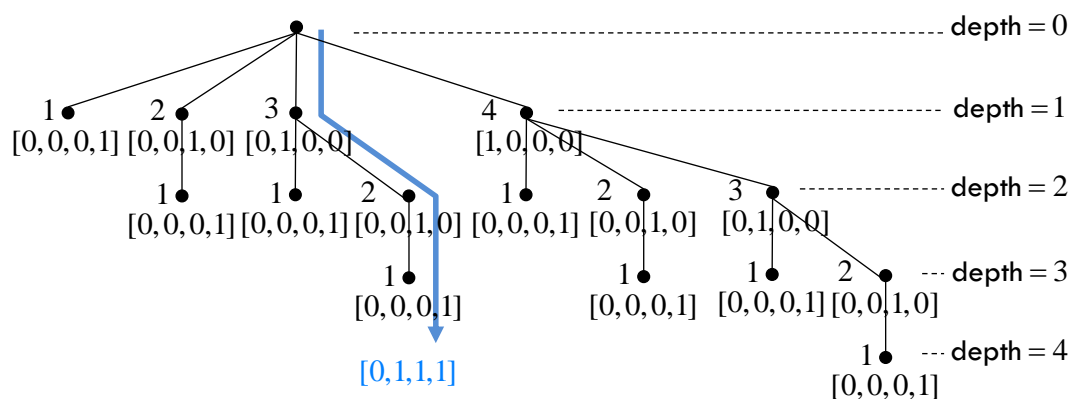


Figure 10.9: Binary version of binomial tree, compatible with clustering **G**.

As can be seen, the number of possible paths, for even a single cluster, quickly expands when the order of the tree grows. In general, the solution space, i.e. number of possible combinations, to partition \vec{Z} zone components in j (non-empty) subsets is given by the Stirling number of the second kind (Graham et al, 1994) via:

$$\left\{ \begin{matrix} \vec{Z} \\ j \end{matrix} \right\} = \frac{1}{j!} \sum_{i=0}^j (-1)^i \binom{j}{i} (j-i)^{\vec{Z}}. \quad (10.33)$$

However, in our case, j is not predetermined. In that case, the possible number of combinations is the summation of Stirling numbers across all values of j , better known as the Bell number, denoted $B(\vec{Z})$ and obtained via:

$$B(\vec{Z}) = \sum_{j=0}^{\vec{Z}} \left\{ \begin{matrix} \vec{Z} \\ j \end{matrix} \right\} = \sum_{j=0}^{\vec{Z}} \left(\frac{1}{j!} \sum_{i=0}^j (-1)^i \binom{j}{i} (j-i)^{\vec{Z}} \right). \quad (10.34)$$

To give an impression on how quickly this number grows, some of the first few Bell numbers are given by $B(1)=1; B(4)=15; B(8)=4,140; B(15)=1,382,958,545$. Clearly, for general networks with hundreds of zone components, it is computationally infeasible to explore the entire solution space, even if we can remove all redundancies in the search tree (Fränti et al., 2002).

10.8 Branch-and-bound solution scheme

The most common way to deal with combinatorial problems that are too large to solve with *brute-force*, is to revert to a *heuristic*. As discussed in Chapter 7, metaheuristics, and evolutionary algorithms in particular, are popular approaches in this situation because they yield reasonably good solutions in an acceptable amount of time while still providing some level of sophistication in their search procedure. The downside of heuristics is their lack of guaranteeing optimality, making it hard to assess the true potential of what it means to solve the underlying problem. We therefore opt not to adopt a heuristic as our main solution algorithm, but instead propose a custom branch-and-bound algorithm. Branch-and-bound algorithms exist both in heuristic and non-heuristic form. Here, a non-heuristic cluster based branch-and-bound algorithm is formulated.

A typical branch-and-bound algorithm aims to minimise (or maximise) some given objective function. The possible solutions are explored by traversing a tree-like structure, for example as depicted earlier in Figure 10.9. The algorithm then explores *branches* of the tree, starting at the root. Each branch partitions the search space by making a particular choice, which also constitutes a partial potential solution. Exploring all branches in the tree reverts to an exhaustive search of the entire solution space. To improve on this, bounds are used to remove branches, but only do so when the branch is guaranteed to not contain the optimal solution. For example, let us assume we have access to a partial solution constructed by reaching some vertex of the branching tree, the best overall solution found so far, and some candidate branches to continue our search. We then only explore branches with the potential to yield a better solution than the current best solution. This potential is determined by the theoretically best possible solution that can be found by exploring all candidates in the branch. This theoretical best solution across the branch is the *lower bound* of the branch. The tighter the lower bound, the more branches can be removed and the faster the optimal solution is found.

To formulate a branch-and-bound algorithm tailored to our optimisation problem, we first cast our optimisation problem of Equation (10.1), in an alternative form by casting the result of the objective function for a single clustering in functional form, denoted by $g(\cdot)$ via:

$$g(\mathbf{G}, \mathbf{K}, \mathbf{K}^G) = \begin{cases} \delta^+ + \delta^- + \delta^{d^{\min}}, & \text{if } \mathbf{G} \geq \mathbf{I}, \mathbf{G} = \mathbf{G}^T, \mathbf{K} \geq \mathbf{G}, \mathbf{K}^G \geq \mathbf{G}, \\ \infty, & \text{otherwise,} \end{cases} \quad (10.35)$$

The optimisation problem as a whole can then alternatively be formulated as:

$$\min_{\mathbf{G}} (g(\mathbf{G}, \mathbf{K}, \mathbf{K}^G)). \quad (10.36)$$

We utilise the aforementioned alternative form of our problem when constructing the bounds in the next section. The branch-and-bound solution scheme utilising these bounds is provided in Algorithm 1. It described a depth-first - cluster enabled - branch-and-bound algorithm under can-link constraints adopting a binary branching approach.

During the search for a solution we must maintain track of three vectors: $\hat{\mathbf{z}}^{\text{current}} \in \mathbb{F}_2^{\bar{Z} \times 1}$ to track the current cluster we are constructing incrementally out of eligible zone components, $\hat{\mathbf{z}}^{\text{excluded}} \in \mathbb{F}_2^{\bar{Z} \times 1}$ to track the zone components that have been denied inclusion in the current cluster based on the partitioning of the solution space, and $\bar{\mathbf{z}}^{\text{processed}} \in \mathbb{F}_2^{\bar{Z} \times 1}$ which tracks the zone components that have been attributed to clusters already part of the partial solution leading up to the current cluster under consideration.

The main difference in Algorithm 1 compared to a more conventional branch-and-bound approach is that the verification of the lower bound cannot occur at every vertex. We only verify if a (partial) solution is still valid whenever each cluster is completed. This is needed because our objective function is dependent on the cluster as a whole (recall Section 10.3). Further, the algorithm adopts a binary branching approach by splitting the solution space in two at every vertex. We do so by either including the zone component candidate in the cluster,

or by excluding the zone component from the cluster. Lastly, the functions referred to within Algorithm 1 are directly discussed after the algorithm formulation below.

Algorithm 1: Depth-first branch-and-bound semi-supervised clustering algorithm.

```

Start:
 $g^{\text{best}} = \text{findInitialSolution}()$  // reference solution to compare against
 $\hat{\mathbf{z}}^{\text{current}} = \mathbf{0}$  // construct new empty cluster  $\hat{\mathbf{z}}^{\text{current}} \in \mathbb{F}_2^{\bar{z} \times 1}$ 
 $\bar{\mathbf{z}}^{\text{processed}} = \mathbf{0}$  // processed zone components  $\bar{\mathbf{z}}^{\text{processed}} \in \mathbb{F}_2^{\bar{z} \times 1}$ 
 $\hat{\mathbf{z}}^{\text{excluded}} = \mathbf{0}$  // temporary excluded zone components  $\hat{\mathbf{z}}^{\text{excluded}} \in \mathbb{F}_2^{\bar{z} \times 1}$ 

branchFromVertex ( $\hat{\mathbf{z}}^{\text{current}}, \hat{\mathbf{z}}^{\text{excluded}}, \bar{\mathbf{z}}^{\text{processed}}$ )

Recursive Function: branchFromVertex ( $\hat{\mathbf{z}}^{\text{current}}, \hat{\mathbf{z}}^{\text{excluded}}, \bar{\mathbf{z}}^{\text{processed}}$ )
// find candidate compliant with constraints
 $\bar{z} = \text{findCandidateFor}(\hat{\mathbf{z}}^{\text{current}}, \hat{\mathbf{z}}^{\text{excluded}}, \bar{\mathbf{z}}^{\text{processed}})$ 

if ( $\bar{z} > 0$ ) // continue with current cluster
     $\hat{z}_{\bar{z}}^{\text{current}} = 1$  // branch (i): include candidate in cluster
    branchFromVertex( $\hat{\mathbf{z}}^{\text{current}}, \hat{\mathbf{z}}^{\text{excluded}}, \bar{\mathbf{z}}^{\text{processed}}$ )
     $\hat{z}_{\bar{z}}^{\text{current}} = 0$  // branch (ii): exclude candidate from cluster...
     $\hat{z}_{\bar{z}}^{\text{excluded}} = 1$  // ... (temporary) excluded until cluster is final
    branchFromVertex( $\hat{\mathbf{z}}^{\text{current}}, \hat{\mathbf{z}}^{\text{excluded}}, \bar{\mathbf{z}}^{\text{processed}}$ )
else // finalise current cluster
     $\mathbf{G}_{\bar{z}} = \hat{\mathbf{z}}^{\text{current}}$  // update total clustering
     $\bar{\mathbf{z}}^{\text{processed}} = \bar{\mathbf{z}}^{\text{processed}} + \hat{\mathbf{z}}^{\text{current}}$  // update processed zone components
    ( $g^{\text{partial}}, g^{\text{bounds}}$ ) = computePartialSolution( $\bar{\mathbf{z}}^{\text{processed}}$ )
    if ( $g^{\text{partial}} + g^{\text{bounds}} < g^{\text{best}}$ ) // only proceed if potential for better solution
        if ( $\exists \bar{z} : \bar{z}_{\bar{z}}^{\text{processed}} = 0, \bar{z}_{\bar{z}}^{\text{excluded}} = 0$ ) // eligible unprocessed zone components?
             $\hat{\mathbf{z}}^{\text{next}} = \mathbf{0}$  // start new cluster
             $\hat{\mathbf{z}}^{\text{excluded}} = \mathbf{0}$  // reset excluded elements on current cluster
            branchFromVertex ( $\hat{\mathbf{z}}^{\text{current}}, \hat{\mathbf{z}}^{\text{excluded}}, \bar{\mathbf{z}}^{\text{processed}}$ )
        else
             $g^{\text{best}} = g^{\text{partial}} + g^{\text{bounds}}$ 
        end
    end
end

```

findInitialSolution: To obtain a reference best solution g^{best} , we take the very first feasible solution found by *branchFromVertex*(.).

findCandidateFor: Eligible candidates are identified via the imposed constraints. An eligible candidate must be able to: (i) can-link based on contiguity (Section 10.6), (ii) can-link based on reachability (Section 10.5.1), (iii) can-link based on similarity (Section 10.5.2). While we formulated the constraints based on the full clustering \mathbf{G} , we are able to verify eligibility on partially completed clusterings as well. One way of implementing these criteria is to create a

partial clustering, denoted $\mathbf{G}^{\text{current}} \in \mathbb{F}_2^{\bar{Z} \times \bar{Z}}$, and verify the constraints in Equation (10.35) based on $\mathbf{G}^{\text{current}}$, instead of \mathbf{G} . We can construct this partial clustering like the following:

$$\mathbf{G}_{\bar{z}\bar{z}'}^{\text{current}} = \begin{cases} 1, & \text{if } \bar{z} = \bar{z}', \\ \hat{z}_{\bar{z}'}^{\text{current}}, & \text{else if } \hat{z}_{\bar{z}}^{\text{current}} = 1, \\ 0, & \text{otherwise.} \end{cases} \quad (10.37)$$

where we include the potential candidate in the cluster under investigation by temporarily including it in $\hat{\mathbf{z}}^{\text{current}}$. When multiple eligible candidates are found, the one with the smallest demand is chosen. The rationale being that low demand components are more attractive to be clustered because they likely contribute the least to the objective function in terms of connectoid cost distortion, while not clustering might cause a high penalty for not meeting the desired minimum demand when they end up as stand-alone clusters.

computePartialSolution: Determines the contribution of the (partial) clustering \mathbf{G} to the objective function, denoted g^{partial} via

$$g^{\text{partial}} = \sum_{\bar{z}=1}^{\bar{Z}} \bar{z}_{\bar{z}}^{\text{processed}} (\bar{\delta}_{\bar{z}}^+ + \bar{\delta}_{\bar{z}}^- + \bar{\delta}_{\bar{z}}^{d^{\text{min}}}), \quad (10.38)$$

where $\bar{\delta}_{\bar{z}}^+, \bar{\delta}_{\bar{z}}^-, \bar{\delta}_{\bar{z}}^{d^{\text{min}}}$, respectively, are readily available per zone component via Equations (10.5), (10.6), and (10.22). Then, the bounds g^{bounds} of the remaining solution space are also determined; it constitutes summing the individual lower bounds of the unprocessed zone components $g_{\bar{z}}^{\text{bound}}$, $\bar{z} \in \{1, \dots, \bar{Z}\}$, via

$$g^{\text{bounds}} = \sum_{\bar{z}=1}^{\bar{Z}} (1 - \bar{z}_{\bar{z}}^{\text{processed}}) g_{\bar{z}}^{\text{bound}}, \quad (10.39)$$

where $g_{\bar{z}}^{\text{bound}}$ needs to be estimated. How to do so is discussed in detail in the next section. Observe that once all zone components have been processed $g^{\text{partial}} + g^{\text{bounds}} = \delta^+ + \delta^- + \delta^{d^{\text{min}}}$, matching original function $g(\cdot)$, as one would expect.

Finally, the recursion continues as long as there are unprocessed zone components and the lower bound of a branch potentially holds a solution better than the current best solution. However, to prevent infinite recursion, the condition to start a new cluster must take into account the fact that the remaining unclustered zone components have not been excluded from the current cluster (otherwise we simply keep excluding zone components infinitely), hence the condition $\exists \bar{z} : \bar{z}_{\bar{z}}^{\text{processed}} = 0, \bar{z}_{\bar{z}}^{\text{excluded}} = 0$.

10.9 Spatial cluster bounds

The effectiveness of branch-and-bound-algorithms is predominantly determined by the tightness of the bounds. In our case we have information both in terms of constraints, as well as insight in how the connectoid cost distortion is likely to change, depending on how zone

components are clustered spatially. We utilise these insights to construct bounds that are as tight as possible.

We propose to apply a spatially aware partitioned *Russian doll* approach. In traditional Russian doll, the constrained optimisation problem is solved multiple times where in each successive round an increasing subset of constraints is imposed on the data points. Once a subset is solved, the results of the previous Russian doll are incorporated as a bound (Verfaillie et al., 1996). This means that while the (unbounded) solution space increases with each larger subset, the bounds available to solve the problem also become tighter, see Figure 10.10(a) for a schematic impression. An alternative to Russian doll, as originally presented by Koontz et al. (1975), can be found in partitioning data points into a number of smaller disjoint subsets. Each subset is solved via a branch-and-bound algorithm and the results are used as bounds when solving the problem as a whole, see Figure 10.10(b). However, in general, the larger the number of data points and the more partitions one creates, the less tight these bounds become.

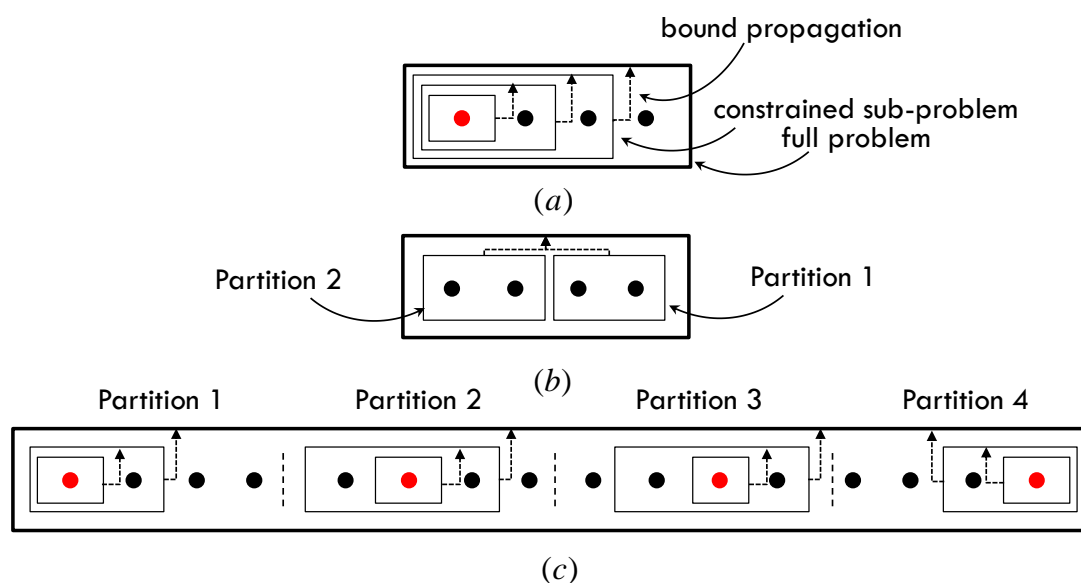


Figure 10.10: (a) Traditional Russian doll, (b) Koontz et al (1975), (c) spatial two-step Russian doll (initial data point for Russian doll in red).

Our method is a combination of the two aforementioned approaches. We argue that in general, zone components on one extreme of the network are unlikely to influence the zoning at the other extreme. The reason for this is that, due to the underlying land use patterns and network topology, natural barriers occur throughout the network that, for the most part, limit interdependencies of spatially distant zone components. Hence, we can obtain relatively tight bounds by exploring the direct neighbourhood of zone components. Therefore, a traditional Russian doll approach is only effective in the first few steps, assuming that we consider spatially close variables at each step. Following the same reasoning we argue that the partitioning approach of Figure 10.10(b) is most effective for spatially (compact) partitions. We therefore propose to first partition and solve the problem per individual zone component, to obtain initial bounds. Then subsequently extend the area around each zone component to tighten the bounds further by imposing constraints on the zone's reachable zone components,

i.e. its can-link options. This leads to solving multiple overlapping problems, similar to Russian doll, but also adheres to an initial partitioning of zone components, similar to Koontz et al.

10.9.1 Single zone component bound

This initial step performs \vec{Z} branch-and-bound runs based on Algorithm 1, where we have no bounds yet, i.e. $g^{\text{bounds}} = 0$. The objective is to obtain the optimal cluster for each zone component without considering any other constraints on any other zone component. The result is used to construct a zone component's initial bound, denoted via $\lambda_{\vec{z}}^I$. Conveniently, we can use Algorithm 1 to quickly find this bound by removing all can-link options of zone components lacking a can-link option to the zone component under consideration (while retaining the can-link option to themselves). This reduces the solution space to a fraction of the original solution space. We denote this reduced can-link matrix via $\mathbf{K}^{\lambda_{\vec{z}}^I} \in \mathbb{F}_2^{\vec{Z} \times \vec{Z}}$, $\vec{z} \in \{1, \dots, \vec{Z}\}$, which is defined via:

$$K_{\vec{z}'\vec{z}''}^{\lambda_{\vec{z}}^I} = \begin{cases} K_{\vec{z}'\vec{z}''}, & \text{if } \vec{z}' = \vec{z} \text{ or } \vec{z}'' = \vec{z} \text{ or } \vec{z}' = \vec{z}'', \\ 0, & \text{otherwise.} \end{cases} \quad (10.40)$$

Note that the optimal cluster under the original objective function $g(\mathbf{G}, \mathbf{K}^{\lambda_{\vec{z}}^I}, \mathbf{K}^G)$ could be a cluster where the contribution of \vec{z} is higher than in some other less optimal cluster. This is because in the original objective function only the total connectoid distortion matters, an example of which is depicted in Figure 10.11.

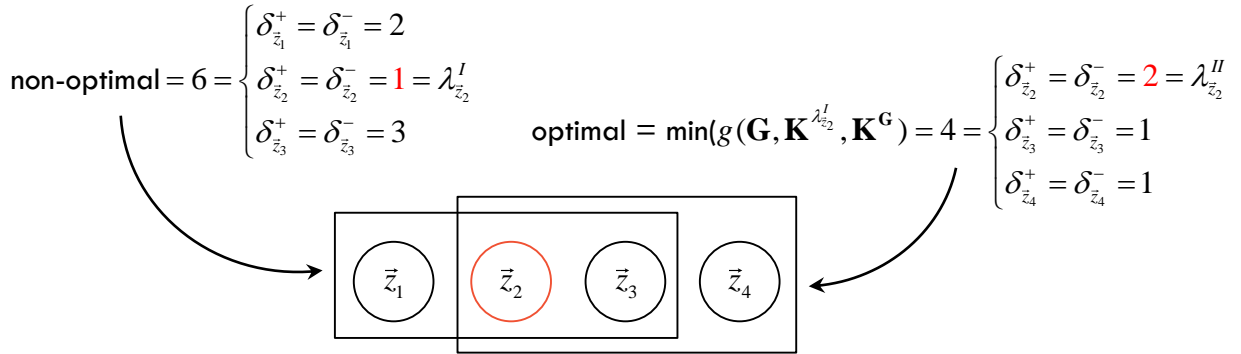


Figure 10.11: Optimal cluster based on original objective function with incorrect lower bound for zone component \vec{z}_2 (assuming d^{\min} is satisfied).

Therefore, the original objective function is not valid to find the true lower bound. To make sure we obtain the actual component based lower bound, we must only take into account the current zone component's contribution to the objective function. Hence, we introduce an alternative to $g(\cdot)$ via:

$$g^{\lambda_{\vec{z}}^I}(\mathbf{G}, \mathbf{K}^{\lambda_{\vec{z}}^I}, \mathbf{K}^G) = \begin{cases} \bar{\delta}_{\vec{z}}^+ + \bar{\delta}_{\vec{z}}^- + \bar{\delta}_{\vec{z}}^{d^{\min}}, & \text{if } \mathbf{G} \geq \mathbf{I}, \mathbf{G} = \mathbf{G}^T, \mathbf{K}^{\lambda_{\vec{z}}^I} \geq \mathbf{G}, \mathbf{K}^G \geq \mathbf{G}, \\ \infty, & \text{otherwise.} \end{cases} \quad (10.41)$$

We then obtain the, per zone component, bound $\lambda_{\vec{z}}^I$ by solving:

$$\lambda_{\vec{z}}^I = \min_{\mathbf{G}}(g^{\lambda_{\vec{z}}^I}(\mathbf{G}, \mathbf{K}^{\lambda_{\vec{z}}^I}, \mathbf{K}^G)), \quad \forall \vec{z} \in \{1, \dots, \vec{Z}\}. \quad (10.42)$$

We can now, for the first time, utilise a non-zero bound in our algorithm and replace $\mathbf{g}^{\text{bound}} = \mathbf{0}$ by the following:

$$g_{\vec{z}}^{\text{bound}} = \lambda_{\vec{z}}^I, \quad \forall \vec{z} \in \{1, \dots, \vec{Z}\}. \quad (10.43)$$

This reduces the search space already. However, we aim to improve on this bound, by finding an even tighter bound. Much in the spirit of a Russian doll based approach, we utilise our new bound to find our second bound. Since it is more costly to find this second bound, we utilise our first bound to make this search more efficient.

10.9.2 Zone component reachability bound

For this bound we again, perform \vec{Z} branch-and-bound runs via Algorithm 1, only now adopting the bound in Equation (10.43). The objective of each run is still to obtain the optimal cluster for each zone component $\vec{z} \in \{1, \dots, \vec{Z}\}$, but we now expand the imposed constraints from only \vec{z} , to all the can-link options of \vec{z} as well. This bound, as argued before, and denoted $\lambda_{\vec{z}}^{\text{II}}$, is much tighter. To enforce exploring the search space related to this bound we construct $\mathbf{K}^{\lambda_{\vec{z}}^{\text{II}}} \in \mathbb{F}_2^{\vec{Z} \times \vec{Z}}$, $\vec{z} \in \{1, \dots, \vec{Z}\}$. Now, only zone components unrelated to *any* can-link option of \vec{z} , are marked cannot-link (except to themselves). Since these zone components are marked cannot-link they are not considered in the clustering, and as we will see, nor in the objective function. This reduces the search to exactly \vec{z} and its can-link options:

$$K_{\vec{z}\vec{z}'}^{\lambda_{\vec{z}}^{\text{II}}} = \begin{cases} K_{\vec{z}\vec{z}'} & \text{if } (K_{\vec{z}\vec{z}'} \parallel K_{\vec{z}\vec{z}''}) = 1 \text{ or } \vec{z}' = \vec{z}'', \\ 0, & \text{otherwise.} \end{cases} \quad (10.44)$$

We utilise $\mathbf{K}^{\lambda_{\vec{z}}^{\text{II}}}$ to replace the original can-link matrix and solve the problem under these relaxed conditions. We, again, cannot utilise the original objective function for the same reason when obtaining the initial bound. On the other hand, we cannot use $g^{\lambda_{\vec{z}}^{\text{II}}}(\cdot)$ either because it will yield the same solution as before, i.e. it only considers \vec{z} instead of including its can-link options. Instead, we include the contribution of all zone components, conditional on being part of the cluster \vec{z} resides in, via:

$$g^{\lambda_{\vec{z}}^{\text{II}}}(\mathbf{G}, \mathbf{K}^{\lambda_{\vec{z}}^{\text{II}}}, \mathbf{K}^{\mathbf{G}}) = \begin{cases} \sum_{\vec{z}'}^{\vec{Z}} \mathbf{G}'_{\vec{z}\vec{z}'} (\vec{\delta}_{\vec{z}'}^+ + \vec{\delta}_{\vec{z}'}^- + \vec{\delta}_{\vec{z}'}^{d^{\text{min}}}), & \text{if } \mathbf{G} \geq \mathbf{I}, \mathbf{G} = \mathbf{G}^T, \mathbf{K}^{\lambda_{\vec{z}}^{\text{II}}} \geq \mathbf{G}, \mathbf{K}^{\mathbf{G}} \geq \mathbf{G}, \\ \infty, & \text{otherwise.} \end{cases} \quad (10.45)$$

This then, yields the optimal *cluster* wide result for zone component \vec{z} when considering all valid cluster combinations this zone component can take part in. Based on this we formulate $\lambda_{\vec{z}}^{\text{II}}$ by extracting the zone component contribution to the optimal cluster value via:

$$\lambda_{\vec{z}}^{\text{II}} = \vec{\delta}_{\vec{z}}^+ + \vec{\delta}_{\vec{z}}^- + \vec{\delta}_{\vec{z}}^{d^{\text{min}}}, \quad \text{conditional on } \min_{\mathbf{G}} \left(g^{\lambda_{\vec{z}}^{\text{II}}}(\mathbf{G}, \mathbf{K}^{\lambda_{\vec{z}}^{\text{II}}}, \mathbf{K}^{\mathbf{G}}) \right), \quad \forall \vec{z} \in \{1, \dots, \vec{Z}\}. \quad (10.46)$$

When we now consider the example in Figure 10.11 again, we would find the optimal cluster because $4 < 6$, hereby setting $\lambda_{\bar{z}_2}'' = 2$, which is indeed a tighter bound ($2 > 1$). In general, it holds that $\sum_{\bar{z}=1}^{\bar{Z}} \lambda_{\bar{z}}' \leq \sum_{\bar{z}=1}^{\bar{Z}} \lambda_{\bar{z}}'' \leq \min_{\mathbf{G}}(g(\mathbf{G}, \mathbf{K}, \mathbf{K}^{\mathbf{G}}))$. There is however a catch to this bound.

10.9.3 Hybrid bounds

Based on the results of the previous section, we prefer to only utilise $\lambda_{\bar{z}}''$. However, this tighter bound is a bound we cannot always use; $\lambda_{\bar{z}}''$ is only valid as long as we can still construct the cluster that the bound is based on, something which might not necessarily be the case. For example, if \bar{z}_4 in Figure 10.11 has already been clustered, then it is no longer available to form the cluster yielding $\lambda_{\bar{z}_2}'' = 2$. In such a situation, a suboptimal result, from the perspective of \bar{z}_2 , follows. For this suboptimal result, the contribution of \bar{z}_2 (by itself) might in fact be less than $\lambda_{\bar{z}_2}''$ (see Figure 10.11), which means that $\lambda_{\bar{z}_2}''$ is no longer the lower bound. To still be able to use our tighter bound we propose a hybrid bound that determines the tightest bound available given the current (partial) clustering via:

$$g_{\bar{z}}^{\text{bound}} = \begin{cases} \lambda_{\bar{z}}', & \text{if } \exists \bar{z}': \bar{z}' \neq \bar{z} \text{ and } \bar{z}'^{\text{processed}} \mathbf{G}'_{\bar{z}'} = 1, \text{ with } \mathbf{G}' = \underset{\mathbf{G}'}{\text{argmin}}(g^{\lambda_{\bar{z}}''}(\mathbf{G}', \mathbf{K}^{\lambda_{\bar{z}}''}, \mathbf{K}^{\mathbf{G}'})), \\ \lambda_{\bar{z}}'', & \text{otherwise,} \end{cases} \quad (10.47)$$

with $\bar{z}' \in \{1, \dots, \bar{Z}\}$. In this hybrid bound, we adopt tight bound $\lambda_{\bar{z}}''$ only when the cluster it is sourced from can still be constructed. This is verified via the first case in Equation (10.47); the optimal cluster that yielded $\lambda_{\bar{z}}''$ is obtained via $\mathbf{G}'_{\bar{z}'}$, if any of the zone components in that cluster are already clustered, i.e. $\exists \bar{z}': \bar{z}'^{\text{processed}} \mathbf{G}'_{\bar{z}'} = 1$, we can no longer construct $\mathbf{G}'_{\bar{z}'}$, so we revert to less tight bound $\lambda_{\bar{z}}'$. When solving the overall problem via Algorithm 1, we adopt Equation (10.47) for determining the bounds.

Observe that this process of partitioning and Russian doll steps can be repeated as many times as desired by iteratively extending the hybrid approach by expanding the considered zone components exposed to the can-link constraints. However, verifying the availability of the multi-cluster optimal solutions becomes more and more costly while the gains in bound tightness are expected to become less significant. We therefore opt to only consider these two bounds and a single hybrid step.

10.10 Soft constraint multiplier estimation

The reader might have noticed that in estimating our bounds we included the demand violation penalty component $\delta^{d^{\min}}$. As discussed in Section 10.3.4 however, we can only do so when we know the per missing trip penalty \hbar^{\max} , which requires estimation. Therefore, in order to estimate the bounds, this estimation must be conducted as a preliminary step to the bound estimation. We now utilise Algorithm 1 to formulate this preliminary step. Instead of relying on empirical data, we integrate the estimation of \hbar^{\max} as part of our general solution procedure. As a result there is no need for the user to perform any additional tasks and \hbar^{\max} simply follows automatically from the earlier constructed zone components.

We argued that \hbar^{\max} should be high enough to avoid the objective function not to cluster zone components whenever d^{\min} has not been met. We also found that \hbar^{\max} can be thought of as a

connectoid cost for a missing trip. This implies that it should at least match or exceed the highest found per trip connectoid cost across all viable clusterings that match or exceed d^{\min} . We therefore estimate \bar{h}^{\max} by imposing d^{\min} as a hard constraint. We also necessarily remove soft constraint $\delta^{d^{\min}}$ from the objective function as well because we do not know \bar{h}^{\max} yet. Imposing d^{\min} as a hard constraint does make the problem as a whole infeasible to solve (in most cases), however by considering each zone component separately, we can expect to be able to solve a large number of runs that allow us to construct a representative value for \bar{h}^{\max} , where the runs that we cannot solve are simply ignored. In total, we conduct $2 \cdot \bar{Z}$ runs of Algorithm 1, each run resulting in either an *average* per trip egress cost or *average* per trip access cost estimate per zone component \bar{z} . We adopt the can-link matrix $\mathbf{K}^{\lambda_{\bar{z}}^l}$ of our initial bound estimation to reduce the search space. This leads to the following two functions, denoted $\bar{g}^{+\bar{z}}(\cdot)$, $\bar{g}^{-\bar{z}}(\cdot)$, respectively, which are defined as follows:

$$\bar{g}^{+\bar{z}}(\mathbf{G}, \mathbf{K}^{\lambda_{\bar{z}}^l}, \mathbf{K}^{\mathbf{G}}) = \begin{cases} -\bar{\delta}_{\bar{z}}^+, & \text{if } \exists \hat{z} : \hat{d}_{\hat{z}}^{\text{total}} > d^{\min}, \text{ rref}(\mathbf{G})_{\bar{z}\bar{z}} = 1, \\ \hat{d}_{\bar{z}}^{\text{total}}, & \text{s.t. } \mathbf{G} \geq \mathbf{I}, \mathbf{G} = \mathbf{G}^T, \mathbf{K}^{\lambda_{\bar{z}}^l} \geq \mathbf{G}, \mathbf{K}^{\mathbf{G}} \geq \mathbf{G}, \\ \infty, & \text{otherwise,} \end{cases} \quad (10.48)$$

and:

$$\bar{g}^{-\bar{z}}(\mathbf{G}, \mathbf{K}^{\lambda_{\bar{z}}^l}, \mathbf{K}^{\mathbf{G}}) = \begin{cases} -\bar{\delta}_{\bar{z}}^-, & \exists \hat{z} : \hat{d}_{\hat{z}} > d^{\min}, \text{ rref}(\mathbf{G})_{\bar{z}\bar{z}} = 1, \\ \hat{d}_{\bar{z}}, & \text{s.t. } \mathbf{G} \geq \mathbf{I}, \mathbf{G} = \mathbf{G}^T, \mathbf{K}^{\lambda_{\bar{z}}^l} \geq \mathbf{G}, \mathbf{K}^{\mathbf{G}} \geq \mathbf{G}, \\ \infty, & \text{otherwise,} \end{cases} \quad (10.49)$$

with $\bar{z} \in \{1, \dots, \bar{Z}\}$, $\hat{z} \in \{1, \dots, \hat{Z}\}$. We negate the average per trip egress/access connectoid cost to retain our minimisation based formulation. The network wide penalty \bar{h}^{\max} is then obtained by taking the negated minimum value, i.e. maximum per trip connectoid cost, across all individual zone component solutions via:

$$\bar{h}^{\max} = - \min_{\bar{z} \in \{1, \dots, \bar{Z}\}} \left\{ \min_{\mathbf{G}} \left(\bar{g}^{+\bar{z}}(\mathbf{G}, \mathbf{K}^{\lambda_{\bar{z}}^l}, \mathbf{K}^{\mathbf{G}}) \right), \min_{\mathbf{G}} \left(\bar{g}^{-\bar{z}}(\mathbf{G}, \mathbf{K}^{\lambda_{\bar{z}}^l}, \mathbf{K}^{\mathbf{G}}) \right) \right\}. \quad (10.50)$$

If one desires, \bar{h}^{\max} can be re-estimated/updated after constructing the bounds in the previous sections. In those cases there is no more need for a hard constraint on d^{\min} because there already exists an earlier estimated \bar{h}^{\max} to construct $\bar{\delta}_{\bar{z}}^{d^{\min}}$.

10.11 Network partitioning

The soft constraint multiplier allows us to compute values for the (original) objective function, while the bounds, constructed in the previous sections, reduce the search space to explore when constructing our zoning system via Algorithm 1. Yet, even with tight bounds, a branch-and-bound approach can quickly become infeasible to solve, especially when the number of zone components becomes large. To mitigate this problem, we decompose the problem as much as possible. If we can split the original problem in multiple sub-problems that have no interdependencies, then, we can solve each component separately and combine the results

without loss of generality. This however, is only possible when these partitions exist, something which heavily depends on the application context.

One of the great benefits of both transport networks and land use planning - which influences zone component shapes - is that often such independent partitions do exist and can be identified. For example, rivers limit can-link options of zones due to reduced accessibility, dissimilarities in land use forbid many zone components to cluster, and motorways (with limited on/off ramps) act as barriers reducing connectivity. This leads to areas with a limited number of internal zone components while lacking options to cluster beyond aforementioned barriers. Hence, we can solve these areas locally, optimally, and separately from other sub-areas, leading to a computationally much more attractive proposition than to try and solve the problem as a whole.

We identify the existing *natural partitions* of zone components by investigating the reachability of our can-link matrix \mathbf{K} . First, we exclude all one-directional can-link options, because if $K_{\bar{z}\bar{z}'} \neq K_{\bar{z}'\bar{z}}$, there is no possibility to cluster these zone components (a cluster relation must be symmetric). We do so via $\mathbf{K} \circ \mathbf{K}^T$, leading to the desired result as depicted in Figure 10.12.

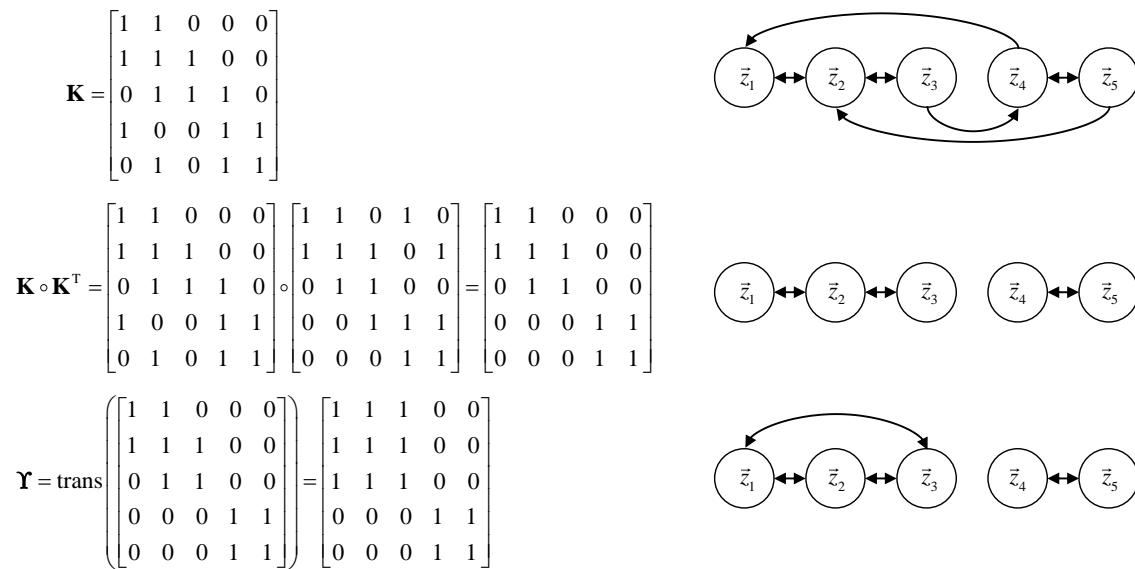


Figure 10.12: Identifying connected components based on bi-directional can-link options \mathbf{K} , (cycles not depicted).

Because $\mathbf{K} \circ \mathbf{K}^T$ is symmetric, its connected components can be identified by taking its *transitive closure*, denoted $\text{trans}(\cdot)$, such that:

$$\mathbf{Y} = \text{trans}(\mathbf{K} \circ \mathbf{K}^T), \quad (10.51)$$

with $\mathbf{Y} \in \mathbb{F}_2^{\bar{Z} \times \bar{Z}}$. As the name implies this function creates a direct link (closure) between two zone components whenever a path between the two zone components can be constructed based on transitivity, i.e. when \bar{z}_1 is adjacent to \bar{z}_2 , and \bar{z}_2 is adjacent to \bar{z}_3 , then \bar{z}_1 becomes also adjacent to \bar{z}_3 as well. The resulting matrix \mathbf{Y} its rows can be rearranged so that it becomes a block diagonal matrix. In block diagonal form it is easy to see that the number of (disjoint)

connected components, i.e. the number of blocks, is simply given by $\text{rk}(\Upsilon)$, while the zone components present within each connected component are found in the non-zero rows of $\text{rref}(\Upsilon)$. A commonly adopted solution scheme to construct Υ is, for example, the Floyd-Warshall algorithm. Observe that the number of connected components is also the number of natural partitions of our problem. We can therefore decompose the original constrained optimisation problem in exactly $\text{rk}(\Upsilon)$ sub-problems, where it holds that:

$$\begin{aligned} \underset{\mathbf{G}}{\text{argmin}} \left(g(\mathbf{G}, \mathbf{K}, \mathbf{K}^{\mathbf{G}}) \right) \equiv \\ \underset{\mathbf{G}}{\text{argmin}} \left(g(\mathbf{G}, \mathbf{K}^{\Upsilon_1}, \mathbf{K}^{\mathbf{G}}) \right) \parallel \underset{\mathbf{G}}{\text{argmin}} \left(g(\mathbf{G}, \mathbf{K}^{\Upsilon_2}, \mathbf{K}^{\mathbf{G}}) \right) \parallel \dots \parallel \underset{\mathbf{G}}{\text{argmin}} \left(g(\mathbf{G}, \mathbf{K}^{\Upsilon_{\text{rk}(\Upsilon)}}, \mathbf{K}^{\mathbf{G}}) \right), \end{aligned} \quad (10.52)$$

with:

$$\mathbf{K}^{\Upsilon_\nu} = (\mathbf{K} \circ (\text{rref}(\Upsilon)_{\nu, \cdot} \mathbf{I})), \quad (10.53)$$

Where $\mathbf{K}^{\Upsilon_\nu} \in \mathbb{F}_2^{\bar{Z} \times \bar{Z}}$ represents the can-link options of each natural partition $\nu \in \{1, \dots, \text{rk}(\Upsilon)\}$. In Figure 10.13 we see how this approach, based on the previous example, identifies the first partition ν_1 , comprising \bar{z}_1, \bar{z}_2 , and \bar{z}_3 , and limits the clustering procedure accordingly via its can-link options.

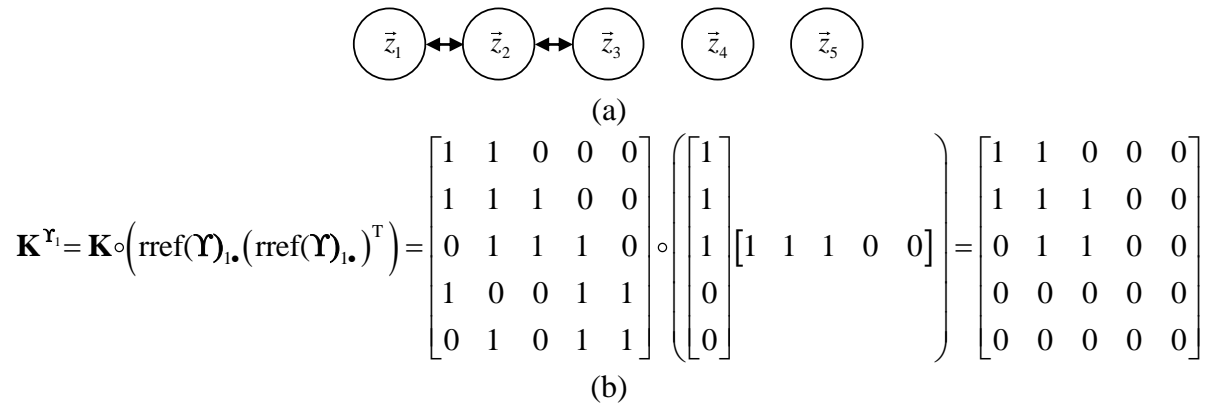


Figure 10.13: (a) Partition ν_1 as a graph (b) obtaining the same partition in matrix notation.

On a final note, we discussed two possible alternative methods to identify connected components in a graph before; via a simple recursive depth-first search algorithm (Section 9.6.1), and via a linear algebra inspired approach (Section 10.6.2). Effectively, any of the methods can be used here as well, yet the transitive closure approach, in our view, allows for the most natural formulation in this particular context. On this note, we conclude the solution scheme for construction of the zonal system.

10.12 Constructing the final representation

With both the formulation and solution scheme discussed, all steps required to construct a consistent traffic assignment representation are in place. To finalise the procedure, we construct supply side representation \mathbf{A}^* , as well as our demand side demand, zoning (and centroid) representations $\mathbf{D}^*, \mathbf{Z}^*$, respectively.

10.12.1 Retained boundary nodes

First, recall that we already discussed how to obtain the final network representation \mathbf{A}^* in Section 9.6.4. However, this representation is conditional on the retained boundary nodes $\mathbf{N}^* \in \mathbb{F}_2^{Z^* \times N}$. These retained boundary nodes, could not yet be formulated because they are conditional on final clustering $\mathbf{G}^* \in \mathbb{F}_2^{\bar{Z} \times \bar{Z}}$. Since we finalised our clustering, we are now finally able to identify \mathbf{N}^* . As we discussed earlier, retained boundary nodes serve as a way to add additional infrastructure to the keep network in case a final zone remains disconnected from this keep network. To verify the need for retaining the boundary node we therefore first establish which final zones are disconnected from the physical road network, denoted $\hat{\mathbf{z}}^{\text{disconnected}} \in \mathbb{F}_2^{Z^* \times 1}$, via:

$$\hat{\mathbf{z}}_{z^*}^{\text{disconnected}} = \begin{cases} 0, & \text{if } \exists n': \mathcal{K}_{n'}^* \cdot \mathbf{1} \geq 1 \text{ and } (\text{rref}(\hat{\mathbf{N}}^+)_{z^* n'} \parallel \text{rref}(\hat{\mathbf{N}}^-)_{z^* n'}) = 1, \\ 1, & \text{otherwise,} \end{cases} \quad (10.54)$$

where $n' \in \{1, \dots, N\}$. The first case verifies if there exists a connectoid in this zone that is connected to the keep network, if so, then z^* is also classified as connected to the physical road network, i.e. $\hat{\mathbf{z}}_{z^*}^{\text{connected}} = 1$, otherwise it reverts to 0. Knowing which final zones are disconnected allows us to construct the retained boundary nodes through:

$$N_{z^* n}^{\circ*} = \begin{cases} \hat{\mathbf{z}}_{z^*}^{\text{disconnected}}, & \text{if } \mathbf{1}^T \mathbf{N}_{\bullet n}^{\circ} = 1 \text{ and } \text{rref}(\hat{\mathbf{N}})_{z^* n} = 1 \text{ and } \mathcal{K}_{n'}^* \cdot \mathbf{1} = 0, \\ 0, & \text{otherwise,} \end{cases} \quad (10.55)$$

where the first case verifies that n is a boundary node in final zone z^* that is not part of the keep network. If this is the case, then the boundary node is retained only when z^* is classified as disconnected, in all other cases we discard it (or it wasn't a boundary node in the first place).

10.12.2 Final zoning system and demand

Knowing the final clustering also allows us to construct the representation of the final zoning system and its demand. We emphasize that the mapping from zone to centroid, is in fact unimportant. One can choose to place the centroid anywhere in the zone since the cost is attached to the connectoid. We therefore leave it to the reader to formalise $\mathbf{Z}^* \in \mathbb{F}_2^{Z^* \times N}$, where we recall that the number of final zones Z^* is given by $\text{rk}(\mathbf{G}^*)$. The final trip matrix $\mathbf{D}^* \in \mathbb{R}_+^{Z^* \times Z^*}$, aggregates the disaggregate node demands of zone component origin-destination pairs (\bar{z}, \bar{z}') based on which respective clusters they reside in via:

$$D_{z^* z'}^* = \sum_{\bar{z}=1}^{\bar{Z}} \text{rref}(\mathbf{G})_{z^* \bar{z}} \sum_{\bar{z}'=1}^{\bar{Z}} \text{rref}(\mathbf{G})_{z' \bar{z}'} (\mathbf{1} \bar{\mathbf{D}}^{\bar{z} \bar{z}'} \mathbf{1}^T) \quad (10.56)$$

10.13 Synthesis and discussion

In order to construct a refined zoning system based on the original zoning system and supply side information based on expected road usage, a general optimisation problem formulation to cluster zone components is proposed. The problem formulation supports both instance-level and cluster-level constraints. Solving this problem yields the optimal zoning system under the

given constraints and objective function considered. Two explicit objectives are considered: (i) minimise the distortion of the originally estimated connectoid costs, which acts as a measure of information loss, (ii) construct clusters with a suitable number of trips, which is the main driver for the desired granularity of the model inputs in a multi-scale environment. This latter objective can also be regarded as a measure of scaling, i.e. the magnitude of the aggregation.

To solve our problem formulation, an optimal branch-and-bound algorithm is proposed. This algorithm is tailored towards our particular problem by: (i) proposing two increasingly tighter bounds following a Russian doll type approach, (ii) a cluster based branching mechanism, (iii) a can-link based natural partitioning method.

We do want to point out that the optimisation problem formulation is mainly suited for when the inputs are provided at a relatively coarse granularity. When inputs, especially the zoning system are already very detailed, the number of zone components will increase. As a result constructing a coarse final zoning system based on the optimisation problem formulation will significantly increase the number of possible combinations and is likely to become much harder to solve optimally. Fortunately, in practice, most problems arise when one attempts to convert a coarse (strategic planning) model into a less coarse (operational) model. It is in those situations that finding an optimal solution is much easier; the number of zone components is then minimal while the desired output is at a higher level of detail such that there is less need for clustering zone components in the first place.

11 Case studies in consistent traffic assignment representation

To demonstrate the suitability of the disaggregation-aggregation framework and our, per step, proposed methodology and solution methods, we conduct two case studies on the Amsterdam II model network and its trip demands. This network contains every single road in the inner city area, the main motorways leading into the city, and a zoning scheme in line with the granularity of a strategic planning model. It contains a wide variety of land uses such as ports, industrial areas, residential areas, and commercial areas. The city itself exhibits a rich diversity in infrastructure as the result of the different time periods in which it flourished: 15th-18th century; the golden age of Amsterdam yielding the canals and old-city, then the 19th century with the 2nd golden age and industrialisation, followed by the 20th century; with contemporary “Vinex” residential areas and wide spread car use. This diversity is beneficial for testing out the effect of our zoning system method as it will be applied to each of these topologies/urban planning philosophies, within a single case study. The detailed network and original zoning system is depicted in Figure 11.1, a hypothetical morning peak demand matrix is also available. This model is in fact identical to Amsterdam I, only with a lower demand and containing the full supply representation.

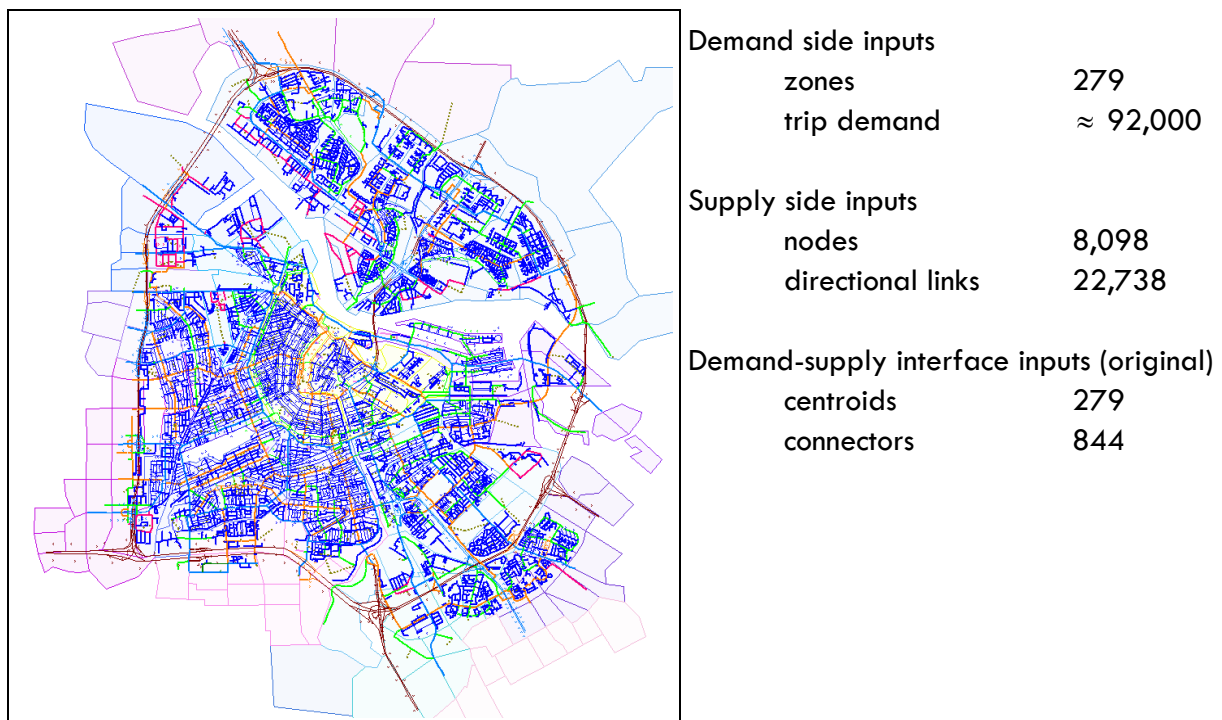


Figure 11.1: Amsterdam II network and related traffic assignment model input characteristics, colours indicate differences in road type and/or zone type.

In Section 11.1 we propose metrics for result comparison and discuss our reference scenarios. Then, the first case study is discussed in Section 11.2, demonstrating the effects of adopting the base connectoid cost estimates as well as all other steps involved in the framework. In Section 11.3, we calibrate the parameters involved in connectoid cost scaling and provide some insight in the effect of altering the constraints on the zone granularity through our second case

study. In 11.4 we briefly discuss the performance, i.e. computational cost of our methods and in Section 11.5 we conclude with a summary and a discussion on model limitations.

11.1 Amsterdam case study preliminaries

Let us first provide some insight in the effects of choosing certain parameter settings on the final representation. The granularity of the representation resulting from our method is mainly determined by our choice for the minimum number of desired trips per zone d^{\min} . When we choose $d^{\min}=0$, there is no incentive to cluster and the final zones are identified by the found zone components. Conversely, if we set $d^{\min}=\infty$, clustering is maximised. In that case it depends on the configuration of the other (hard) constraints to what extent clustering occurs. For example, when relaxing pair-wise can-link constraints fully, i.e. 100% dissimilarity is allowed as well as any travel time distance between zone components, i.e. $e^{\max}=1, \tau^{\max}=\infty$, respectively. Then, a single zone per natural partition results. Alternatively, we can also attempt to recreate the original zoning system by only allowing zone components to merge that are exactly similar in terms of productions/attractions, irrespective of their distance, via $e^{\max}=0, \tau^{\max}=\infty$, i.e. only if zone components originated from the same original zone they are considered similar enough. We can use this latter configuration to compare the original network's zoning system to our method. In this situation, the only difference should be found in the demand-supply interface representation, see also our second case study.

It is notoriously difficult to objectively assess the results of design problems such as this, because it is hard to establish meaningful reference points to compare against. To still provide insights in the effectiveness of the method we propose to construct the following two reference scenarios. A *best-case* reference scenario and a *status-quo-case* reference scenario.

The *best-case reference scenario* is our source model resulting from the disaggregation step in the framework. In this scenario each node becomes a zone, it represents the situation without any aggregation and, under the given assumptions, does not suffer any information loss. Hence, any aggregation based on our methodology, in the best case, matches the results of this disaggregate representation.

The *status-quo-case reference scenario* is, unavoidably, subjective. In our case, it is the Amsterdam I model (used in the Chapter 6 case studies). Recall that this model is identical to Amsterdam II, albeit for the reduced network detail and its centroids and connectors are readily provided by Amsterdam city council. These centroids and connectors have been constructed based on legacy projects, expert opinions, and ad-hoc modeller decisions. Our approach should be able to match the results of this alternative representation and hopefully surpass them, when adopting the same zoning system. Hence, to demonstrate suitability of our method we aim to verify that our methodological approach to placing and estimating connectoid costs can outperform the original model's centroid/connector based results.

Note that this status-quo reference model only provides a subjective comparison and results cannot be generalised, but at least we can provide some results that we feel are indicative of the potential of our method. Also, we can only compare against this second scenario when the

zoning system resulting from our method is virtually identical (e.g. in terms of the number of zones, their shapes and locations) to the original zoning system.

11.1.1 Comparing results between scenarios

Traffic assignment model input representations are eventually used in traffic assignment. Comparing the quality of these representations is therefore best verified based on traffic assignment results. We use a traditional static capacity restrained DUE assignment model to generate traffic assignment results. A volume-averaging method is used to smooth path flows across the conducted 50 iterations (to approximate equilibrium conditions). Given that we are only interested in the relative accuracy of the model representations compared to the disaggregate representation, the actual assignment model is of lesser importance and the aforementioned traditional capacity restrained assignment model, arguably, suffices.

Two different traffic assignment model outputs are used as metrics to compare results: zone-to-zone travel times, and average link flow rates. Both metrics adopt a Root Mean Squared Error (RMSE) approach, where the disaggregate results serve as the predicted value. To account for differences in link characteristics, we utilise the v/c ratio differences of links rather than absolute flow differences. Also, we only include *comparable links*, denoted by indicator matrix $\mathbf{A}^{\text{comparable}} \in \mathbb{F}_2^{N \times N}$. A link is deemed comparable when: (i) the link is present in all network representations being compared, (ii) links are not directly adjacent to an original connector nor to a connectoid end node. The latter is due to the fact that link flows around connectors/connectoids are severely distorted because they are the main points of access to the physical network for trips. Therefore, we do not want to introduce an arguably unfair bias towards favouring our results because, in our method, we generally use more connectoids that allow for a bigger spread of access/egress flows than is the case when one only adopts a limited number of connectors (as in the status quo model). Note that connectoids can reside in different places than the original network's connectors, hence these distortions are representation dependent. The link flow based RMSE for comparing the status-quo scenario to the source model is then given by:

$$\text{RMSE} = \sqrt{\frac{\sum_{n=1}^N \sum_{n'=1}^N A_{nn'}^{\text{comparable}} \left(\frac{\bar{q}_a^*}{q_a^{\max}} - \frac{q_a^*}{q_a^{\max}} \right)^2}{\mathbf{1}^T \mathbf{A}^{\text{comparable}} \mathbf{1}}}, \quad (11.1)$$

where \bar{q}_a^* is the flow following from equilibrating the disaggregate source model, while q_a^* is the equilibrated link flow for the status-quo scenario. The RMSE for the zone-to-zone travel times are obtained in a similar fashion. Comparing the status-quo scenario to the disaggregate source model results in RMSE as given in Equation (11.2):

$$\text{RMSE} = \sqrt{\frac{\sum_{z=1}^Z \sum_{z'=1}^Z \left(\frac{H_{zz'} - \bar{\mathcal{H}}_{zz'}}{\bar{\mathcal{H}}_{zz'}} \right)^2}{Z^2}}, \quad \text{with } \bar{\mathcal{H}}_{zz'} = \frac{\sum_{n=1}^N \sum_{n'=1}^N N_{zn'} N_{zn'} \bar{\mathbf{D}}_{nn'} \bar{H}_{nn'}}{(\mathbf{N}_{z \bullet} \bar{\mathbf{D}}_{\bullet n})(\mathbf{N}_{z' \bullet} \bar{\mathbf{D}}_{\bullet n'})}, \quad (11.2)$$

where $H_{zz'}$ denotes the original status-quo zone-to-zone travel time. The disaggregate node-to-node travel times $\tilde{\mathbf{H}}$ are aggregated up (taking the demand weighted average) to match the granularity of original zoning system resulting in $\tilde{\mathcal{H}}_{zz'}$. We leave it to the reader to observe that RMSE formulations for alternative representations based on our disaggregation-aggregation method scenario can be formulated following the same approach. Finally, in our case studies we adjusted Equation (11.2) such that we only considered zone-to-zone pairs with non-negative demand to ensure we capture the differences that make an actual impact on the results.

11.1.2 Conducted case studies

We conduct two comprehensive case studies; the first case study demonstrates the results of applying the proposed methods for each of the steps in the disaggregation-aggregation framework while adopting the base estimation method for constructing connectoid costs. The second case study focusses on parameter estimation, especially with regards to the scaling of the connectoid cost base estimates. Both case studies aim to construct a zoning system identical to the original zones, but do so by employing branch-and-bound Algorithm 1. The purpose of choosing this particular granularity is to compare results with the status-quo scenario. Lastly, the second case study also explores a more fine-grained zoning system to verify if the found parameter settings are transferrable across granularities.

11.2 Amsterdam case study I: Basic approach

As mentioned, this initial case study serves as a demonstration of applying our disaggregation-aggregation framework on a real-world network. This allows us to demonstrate all the steps involved and compare the differences in our demand-supply interface to the centroid/connectors in the status-quo case study. Table 11.1 provides an overview of the steps and the adopted configuration settings.

Table 11.1: Case study I design decision overview.

| Framework step | Implementation method | Parameters |
|------------------------------------|---|---|
| Step 1-2 Source model construction | AON node-to-node assignment | - |
| Step 3 Supply representation | v/c ratio based link classification | $\kappa^{\min} = 0.1$ |
| Step 4 Demand-supply interface | connectoid placement, base connectoid cost estimation | - |
| Step 5 Demand representation | Branch-and-bound Algorithm 1 | $e^{\max} = 0, \tau^{\max} = \infty, d^{\min} = \infty$ |

11.2.1 Step 1 and 2: creating the Amsterdam source model

We first create the disaggregate source model inputs as discussed in Section 9.3. This involves creating the disaggregate node-to-node trip demand matrix first. Recall that the portion of the original zone demand assigned to each node is determined by the node's weight. We choose the node weight to be determined by the number of outgoing links of each node, while setting the weight of boundary nodes to zero:

$$w_n^z = \begin{cases} 0, & \text{if } N_{zn}^\circ = 1, \\ \mathbf{A}_{n\bullet}^{z+} \mathbf{1}, & \text{otherwise,} \end{cases} \quad (11.3)$$

We acknowledge that this is a rather basic approach. However, this is not a problem when – as we do here – only construct this base model as a reference to compare against a more aggregate alternative model representation, based on this very source model. Clearly, if this model were to be used for an actual application, one would for example include land use information, or other available data sources to provide a more realistic estimate.

Once, the disaggregate demand matrix is constructed, it is assigned to the network using AON assignment (Section 9.4). The v/c ratio based results are depicted in Figure 11.2. This step does not involve the use of centroids and connectors, instead, all nodes acts as zones. This means that virtually all links experience some demand, yet truly local roads exhibit very low flow rates, which is utilised in subsequent steps to identify the zone components.

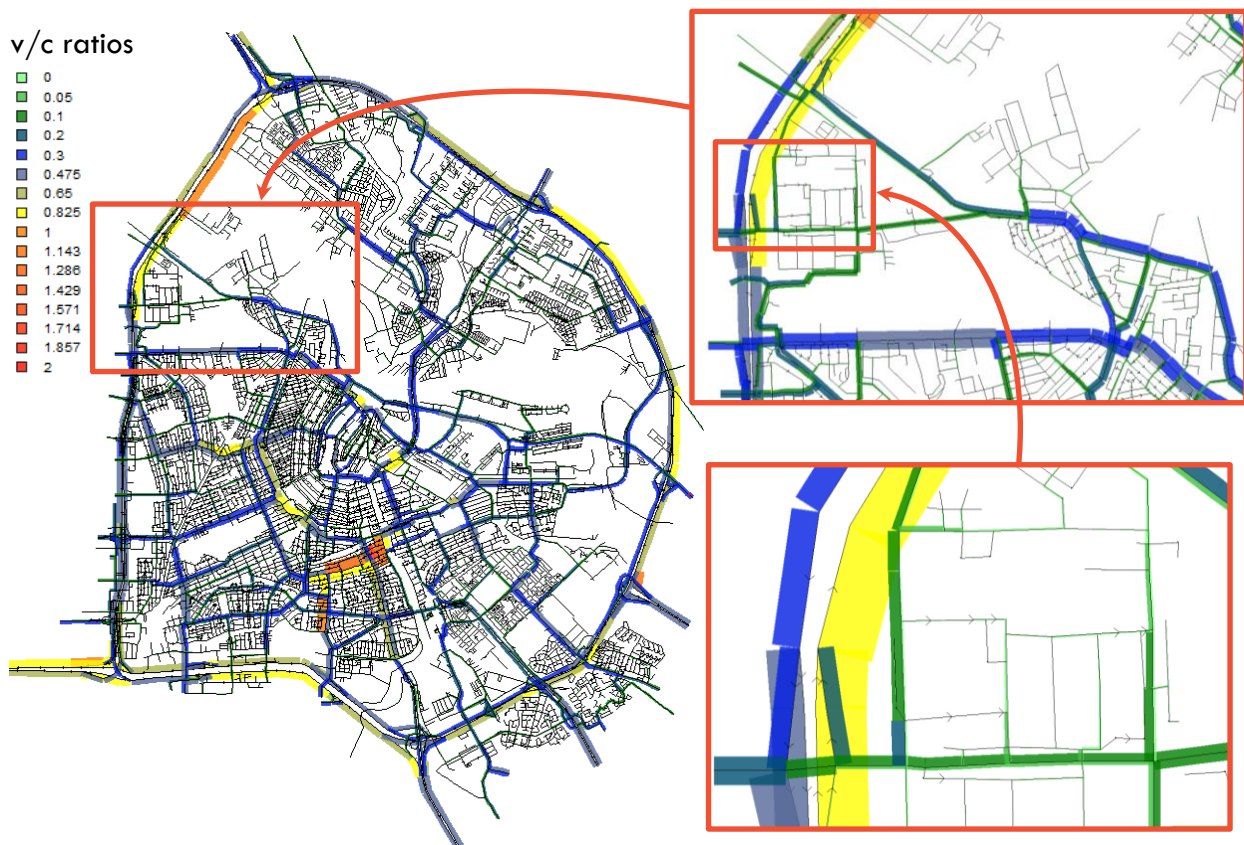


Figure 11.2: Disaggregate AON assignment results in terms of v/c ratios, from various perspectives.

Note that we deliberately scaled back the demand matrix (compared to the demand used in Part II), so that it leads to a comparatively higher impact of connectoid costs on travel times. In doing so, the model becomes more sensitive to connectoid cost estimates allowing for more accurate parameter calibration (see Section 11.3). Secondly, we find that in the inner city v/c ratios are comparatively low compared to motorways. This is partly due to the fact that we do not take (the available) traffic signals into account.

A more sophisticated model would be able to incorporate the turn flow restrictions based on the available green times. However, we argue that while these effects on v/c ratios can be significant, they have far less influence on our link classification method and the final supply model representation. For example, let us assume that queues resulting from reduced capacity at traffic lights do not spillback beyond the link preceding the signal and that this preceding link is marked as a keep-link under AON. If this is true, then the supply representation based on AON is identical to a situation where traffic signals would be considered, simply because all the links experiencing increased v/c ratios would already be marked as keep-links. While we cannot guarantee that queues do not spillback beyond the signal's preceding link, we can choose to calibrate our threshold value κ^{\min} such that we capture all links preceding any signalised intersection. Therefore AON, while crude, is considered capable enough of establishing our link classification conditional on the fact that we choose κ^{\min} appropriately.

11.2.2 Step 3: creating the Amsterdam supply input representation

The granularity of the final supply input representation depends on how we choose the link classification v/c ratio threshold κ^{\min} . In this classification, we differentiate between links with low expected road usage, leading to stable internal travel times and links that might experience delays. As argued previously in Section 9.5, it is more important to retain the infrastructure that allows for accurate traffic flow interactions than to succumb to the temptation to remove or abstract out infrastructure to reduce computation times. We already saw in Part II that computational effort is mainly driven by the number of zones, rather than network granularity. Therefore, only when links have very low expected levels of usage we can justify abstracting them out without risking much information loss.

We therefore argue that κ^{\min} must be chosen conservatively. Following the discussion in the previous section on the impact of signals and the fact that we have access to where the signals reside in the network, we choose to calibrate κ^{\min} with the objective to obtain a final supply representation that, at the very least, captures the infrastructure around existing signals. In doing so, we also comply with the assumption we made in the previous section that, if we were to include signals, the final supply representation under a v/c ratio based AON assignment does not change.

After some experimentation, we found that choosing a threshold of $\kappa^{\min} = 0.1$, satisfied these conditions. The fact that this is a low value, we argue, is a good result because there is little room to reduce κ^{\min} further without risking to include the full network, while increasing κ^{\min} would lead to missing important parts of the infrastructure, such as signalised intersections. This seems also to suggest that, regardless of the granularity of the model, the underlying physical road network that should be considered, to obtain accurate path travel times, is more or less the same. Something which might come as a surprise, given the wildly varying levels of network detail found across the current modelling paradigms. Figure 11.3 depicts the link classification resulting from applying $\kappa^{\min} = 0.1$ to the Amsterdam case study, the inset shows intersection status as well. We specifically depicted the connectivity-keep links that ensure the network connectivity between the identified keep links based on the main section of the shortest paths as discussed in Section 9.5.1.

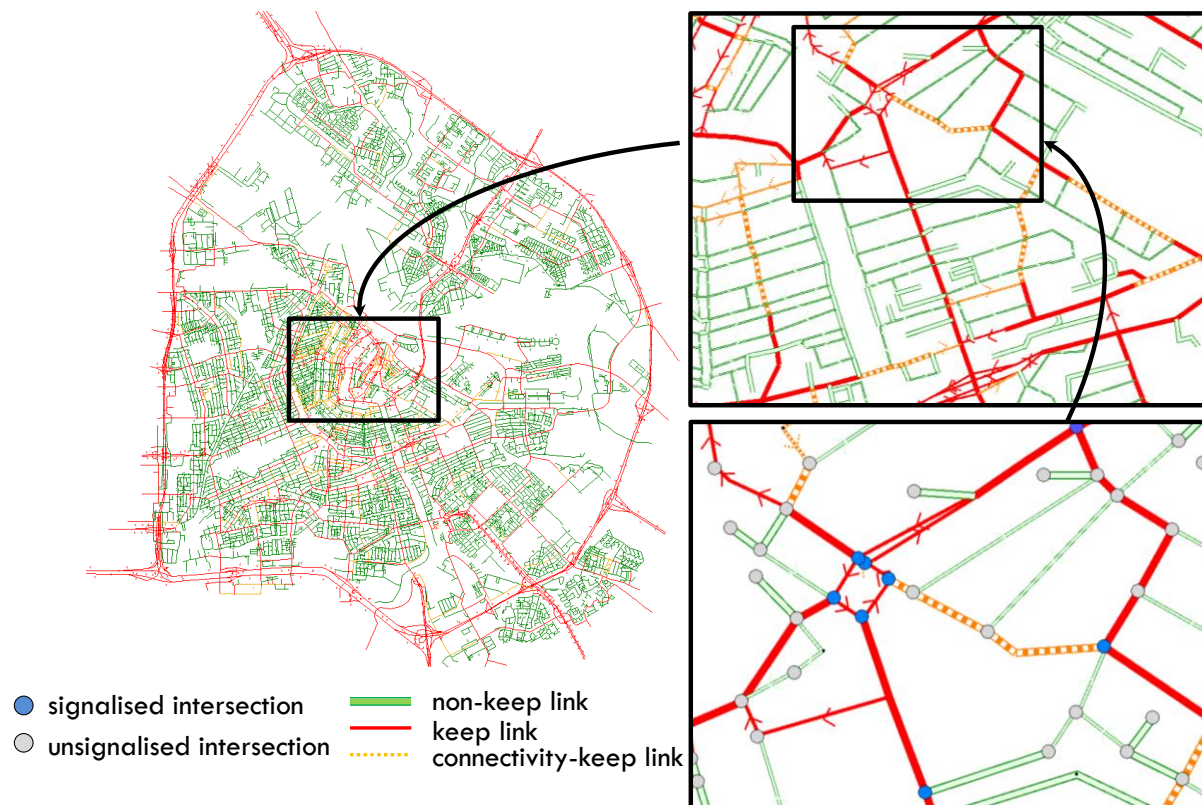


Figure 11.3: Link classification on Amsterdam case study, $\kappa^{\min} = 0.1$, at various zoom levels.

We find that, at this stage, there are 4,774 keep links and 1,051 connectivity keep links. Hence, the keep network accounts for roughly 25% of the original number of links in the network.

11.2.3 Step 4: creating the Amsterdam demand-supply interface

In this step we start by identifying the zone components as originally formulated in Section 9.6.1. They represent topologically connected areas with expected internal travel time stability. They are obtained based on the link classification of the previous step.

The importance of delineating zone components not only based on expected road usage, but also by the original zone boundaries is illustrated by depicting what zone components look like without boundary nodes, as is depicted in Figure 11.4(a) and (b). As can be seen, in the absence of boundary nodes, many of the zone components span multiple (partial) original zones. In case of the highlighted zone component, there are significant differences in travel patterns between the original zones the zone component covers, exemplified by the differences in productions/attraction patterns, see Figure 11.4(c). Boundary nodes prevent such undesired initial amalgamations. Figure 11.4(d) shows the actual obtained zone components when considering boundary nodes, i.e. additional delineation by original zone boundaries, resulting in a more appropriate starting point for our zonal aggregation in Step 5.

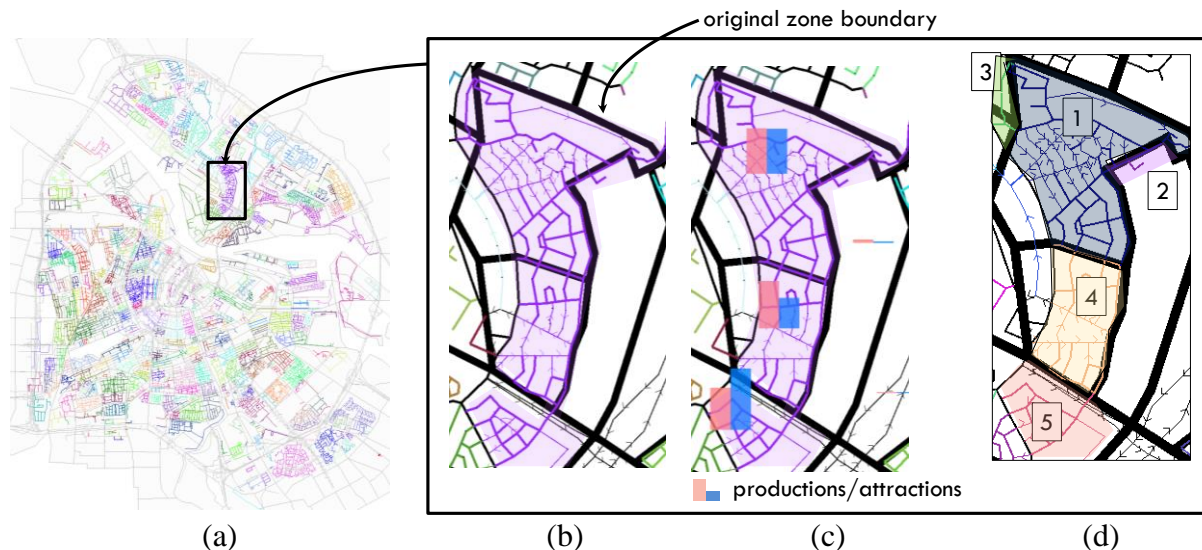


Figure 11.4: (a) Zone components in Amsterdam network without boundary nodes, (b) close up example zone component (purple) without considering boundary nodes, (c) original zone productions attractions, (d) zone component delineation with boundary nodes.

11.2.3.1 Connectoid placement and cost estimation

Connectoids are automatically constructed on all physical links that interface between the keep link network and the non-keep link network, following the procedure proposed in Section 9.6.3. Some examples are provided in Figure 11.5. As can be seen, some zone components are quite large with only a few connectoids, while other zone components are relatively small, but have many connectoids.



Figure 11.5: Impression of differences in connectoid densities across Amsterdam network.

Based on the supply side representation, zone components, and connectoid placement, we estimate the zone component connectoid costs \bar{H}^+ , \bar{H}^- , respectively. We do so by adopting the base method without any additional scaling, as discussed in Section 9.6.6.

11.2.4 Step 5: creating the Amsterdam demand input

We intend to use the representation resulting from this first case study to compare against the status-quo reference scenario. Therefore, we aim to reconstruct the original zoning system

through our branch-and-bound procedure. We do so by setting $e^{\max}=0$, $\tau^{\max}=\infty$, respectively. Interestingly, we only succeed in replicating the original zoning system when the original zoning system complies with our contiguity constraint. Given that we enforce a topological contiguity rather than a shape based contiguity, violations of this constraint were found in the original zoning system and they occur more frequently than one might initially expect, as is illustrated in Figure 11.6.

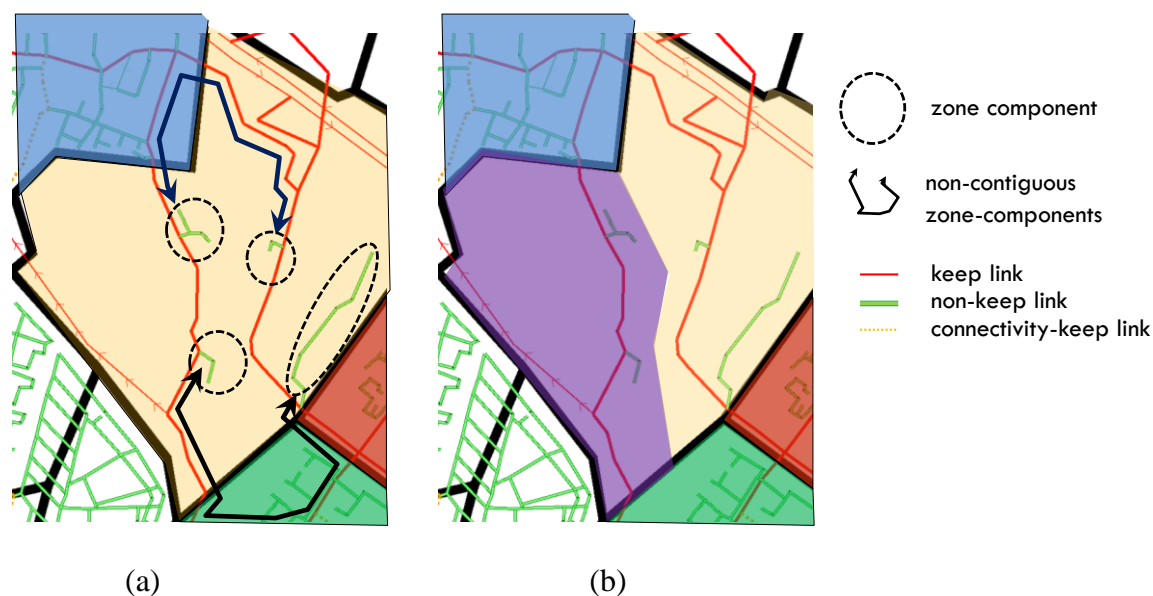


Figure 11.6: (a) Contiguity violation in original network zone, (b) final zone representation based on Algorithm 1.

Here we find that the zone components (dashed circles) in the original zone (marked yellow, Figure 11.6(a)) cannot get to one another without going through other original zone's infrastructure. Hence, the original zone (provided as a given and fixed input) is not contiguous from a supply side perspective, so the original zoning system cannot be reconstructed, unless we forcibly violate topological contiguity. In our comparison, we choose to maintain our contiguity constraint. Yet, whenever an "illegal" original zone is identified, the results of the contiguous subdivided zones are grouped, based on a demand weighted average, to yield values that can be compared to the original zone, consistent with the construction of the zone-to-zone travel time RMSE in Equation (11.2). This way, we construct a viable zoning structure while still being able to compare results to the original zoning system.

Figure 11.7 gives an impression of the final zoning system constructed by Algorithm 1, the original shape based zoning system is added as an overlay for reference. Note that links, retained in the supply representation, are depicted in black and are not attributed to a final zone, the colour coded links are the links that are being abstracted out when performing assignment and represent the areas of internal travel time stability of each final zone. Finalising the zoning system also allows us to construct the final boundary nodes. This results in additional keep infrastructure because we classify all links on the shortest paths from these nodes to the keep network. Further, we add all non-keep links, connecting the connectoids to the keep network, to the keep network as well. This results in an additional 3,208 keep links. Hence, we retain 41% of the original complete network in our new supply side representation.

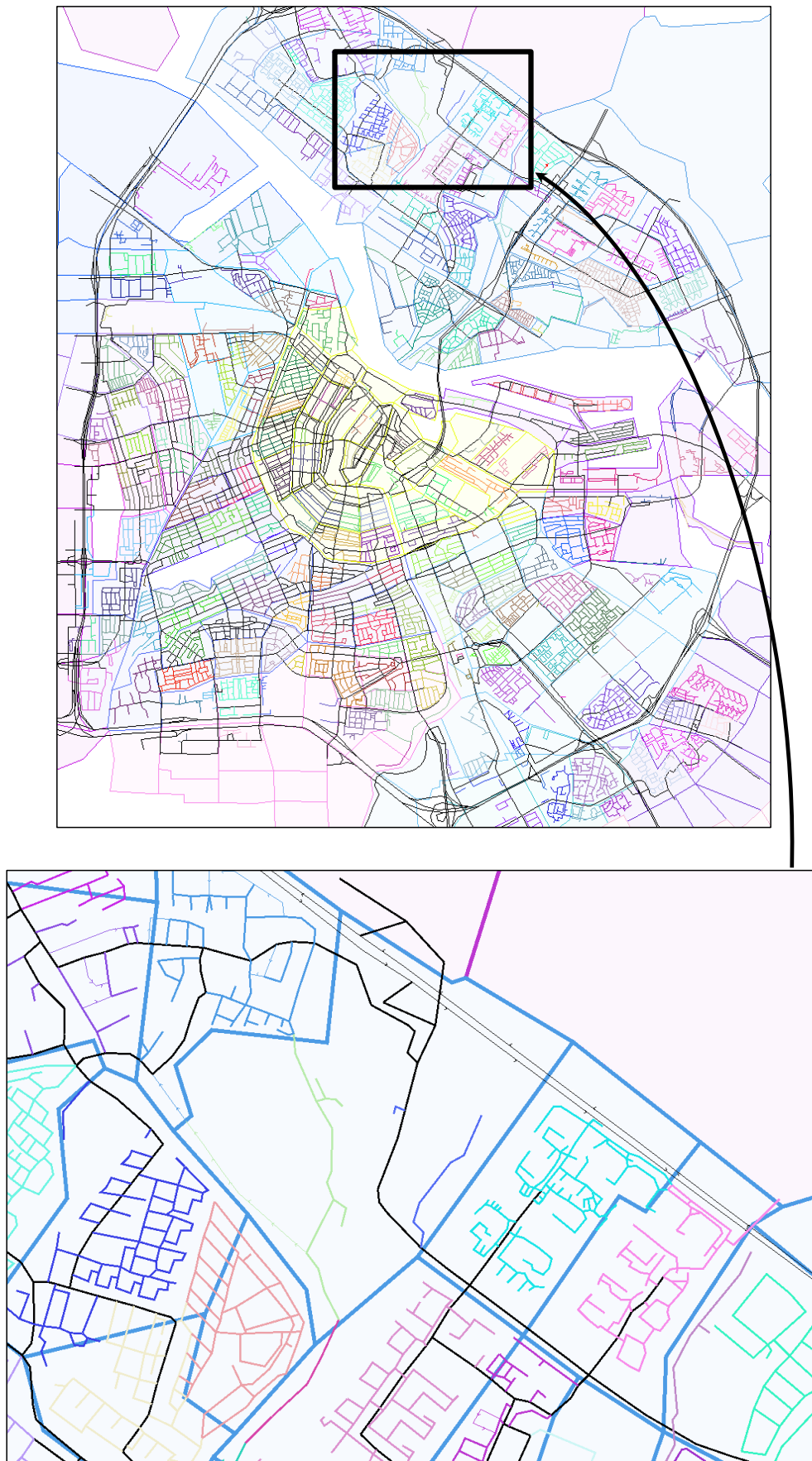


Figure 11.7: Constructed zoning system as colour coded links, aligned with original zoning system.

11.2.5 Amsterdam case study I results

For each of the three representations; the best-case scenario, the status-quo scenario, and our disaggregation-aggregation based representation, we obtained results via DUE volume-averaging traffic assignment equilibration, outlined in Section 11.1.1. The best-case scenario is used as a reference allowing us to compare the results of the other two scenarios in Figure 11.8.

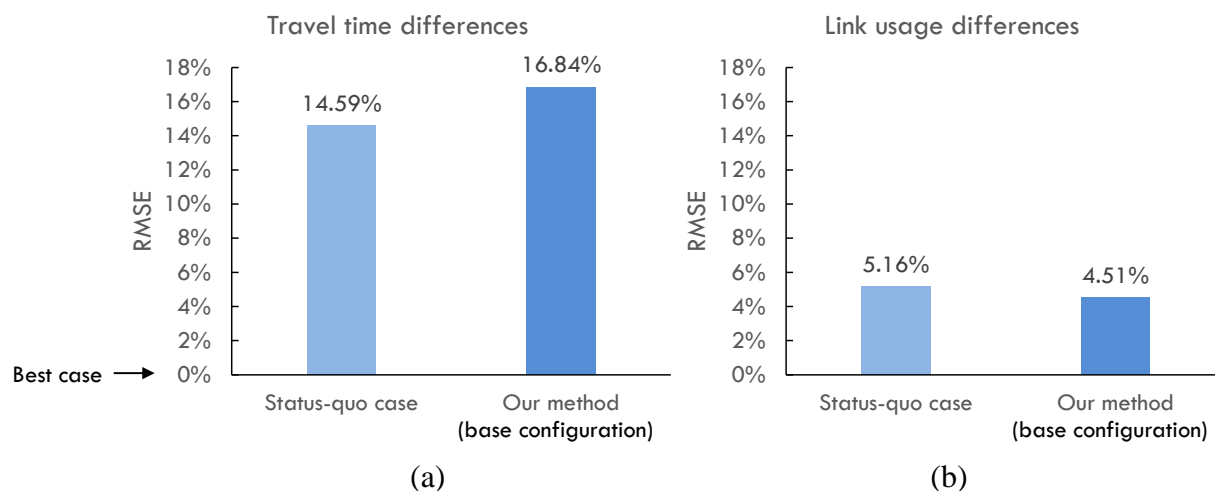


Figure 11.8: Results of status-quo and base configuration of disaggregation-aggregation framework in comparison to disaggregate best-case scenario for (a) zone-to-zone travel times, (b) link usage.

We find that both the coarser model representations suffer significant information loss. This loss amounts to almost 17% RMSE of travel times compared to the disaggregate case. The RMSE on the link level (for comparable links) is surprisingly low. Interestingly, we also find that, in this initial configuration, our results are in fact somewhat worse than the status-quo model. This either indicates the practitioner's model is actually pretty good, our current configuration is too simplistic, or both. Analysing the results more closely we find that both the status-quo scenario, as well as our method, overestimate zone-to-zone travel times significantly, where our method's overestimation exceeds that of the status-quo representation, see Figure 11.9.

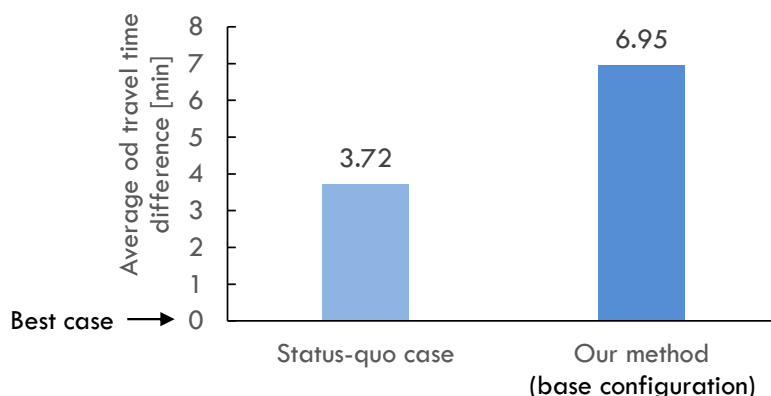


Figure 11.9: Average origin-destination travel time difference in minutes, via summation of absolute origin-destination travel time differences divided by number of origin-destination pairs.

The status-quo overestimation is likely the result of the strategic network missing local infrastructure to accommodate short trips, requiring larger detours to the sparsely available connectors. For our method, this overestimation is caused by carrying over the distorted cluster based connectoid costs into the assignment procedure. Figure 11.10, shows an example where two zone components are clustered, however some of the shortest paths between internal cluster nodes and cluster connectoids - used in estimating the cluster connectoid costs - are costly and are unrepresentative of the true first/last mile cost.

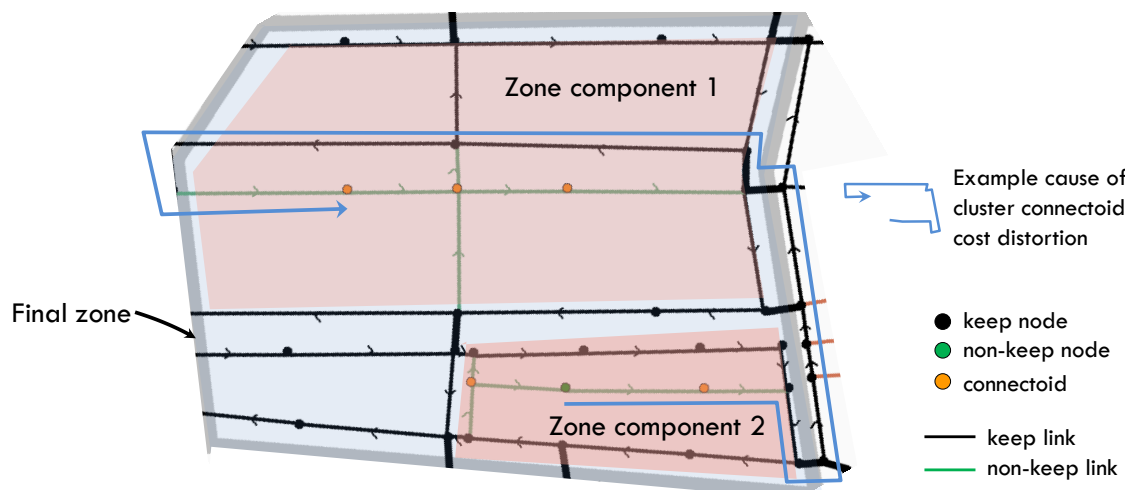


Figure 11.10: Clustering and the effects of connectoid cost distortion.

This, in our view, does not discredit using connectoid cost distortion as a measure to quantify the (un)attractiveness of clustering zone components, because clearly the two zone components in the example have poor intrazonal connectivity, mainly due to the one-way streets in this area, which should deter them from clustering. However, when the clustering procedure does decide to cluster such zone components, for example to meet a minimum demand, or to force a particular zone cluster to match some reference zoning; the cluster connectoid cost might cause an overestimation in the zone-to-zone path travel times. In such cases we should not carry it over directly for assignment purposes. Instead, we aim to calibrate, change, and/or scale the available cluster and/or component based connectoid costs in a post-processing step to better match the disaggregate travel times and link flows as much as possible.

11.3 Amsterdam case study II: post-processing connectoid costs

This second case study is dedicated to exploring different ways to calibrate the connectoid costs discussed in the previous section. We do so in four distinct ways: (i) vary connectoid cost between cluster connectoid cost and the original – lower - zone component based cost, (ii) we employ the service area scaling factor method, proposed in Section 9.6.7, (iii) we employ the centrality scaling factor, proposed in Section 9.6.8, (iv) we explore the effect of reducing the number of connectoids.

11.3.1 Connectoid cost choice based on relative zone component size

To provide insight in to what extent the cluster distortion influences the overestimation of travel times, we first explore reverting the cluster based connectoid cost back to the zone component connectoid cost. In other words, regardless of what cluster a zone component belongs to, the

employed connectoid cost used in assignment is estimated based only on the nodes internal to the zone component the connectoid resides in. This should lead to an underestimation of travel times because it disregards the access/egress cost of trips in other zone components in the same cluster that can use the connectoid. Therefore, the “true” connectoid cost is likely to be found somewhere in between those two extremes. In addition, we also explore taking the average of the zone component connectoid cost and cluster based connectoid cost. The results are plotted in Figure 11.11. We find that the zone component based cost performs better on path travel times, but worse on the link usage, compared to the original cluster based costs. Taking the average connectoid cost yields results in between the two approaches, but does outperform the status-quo scenario on both metrics used.

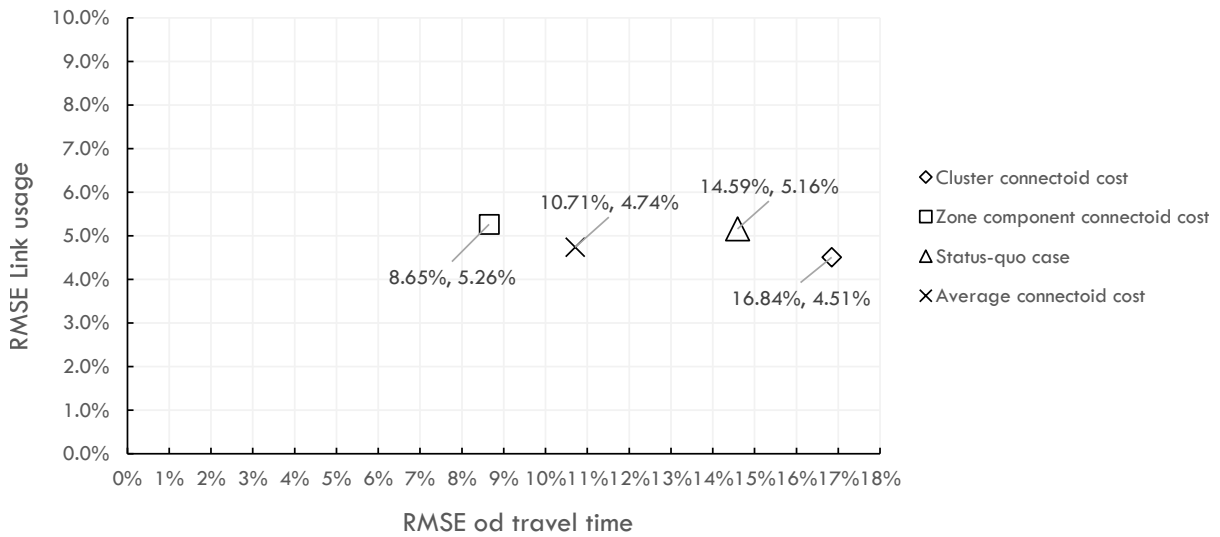


Figure 11.11: Comparing different connectoid cost estimates against the status-quo scenario.

The alternative connectoid costs as outlined above are crude because they are applied to all connectoids in the same fashion. We therefore propose a slightly more sophisticated way to attribute either zone component based or cluster based connectoid costs to connectoids. We let each clustered zone component choose between applying the original zone component connectoid cost or the cluster based connectoid cost depending on the relative size of the zone component within the cluster.

Let us measure the (topological) size of a zone component by its internal infrastructure, i.e. total internal link length. The relative size factor $l_{\vec{z}}^G \in [0, 1]$, of a zone component $\vec{z} \in \{1, \dots, \vec{Z}\}$ is then expressed by the ratio between the zone component’s size and the largest zone component in its cluster via

$$l_{\vec{z}}^G = \frac{\sum_{a=1}^A \sum_{n=1}^N \ell_a \vec{N}_{\vec{z}n} A_{na}^+}{\max \left\{ \sum_{a=1}^A \sum_{n=1}^N \ell_a \vec{N}_{\vec{z}'n} A_{na}^+ \mid G_{\vec{z}\vec{z}'}^* = 1; (\vec{z}, \vec{z}') \in \{1, \dots, \vec{Z}\} \right\}}. \quad (11.4)$$

We use this metric to determine which of the two connectoid costs is likely to be most appropriate. When a zone component \vec{z} is relatively small because there exist other larger zone components, then the other larger zone component(s) are more representative for the zone as a

whole and we should deter using the small zone component's connectoid cost estimate. Hence, the cluster based connectoid cost is most appropriate. On the other hand, when the zone component is small when all other zone components in the cluster are also small, then all zone component are roughly equally representative as access/egress points for the zone and we are probably better off using the zone component based cost (because on average the originally adopted cluster based cost yielded an overestimation). When a zone component is relatively large, its zone component based costs is most likely already representative for the zone cluster as well. We can therefore use it instead of the cluster cost, also reducing our initial cost overestimation. We could alternatively adopt the cluster based cost here as well, but as shown before, we rather avoid including potential outliers that are contributed to our initial overestimation, so we refrain from doing so.

Given that we do not know what relative size provides the best choice to switch from cluster costs to zone component costs, we estimate a threshold $l^{\min} \in [0,1]$, where, once $l_z^G \geq l^{\min}$ zone component \vec{z} adopts zone component based connectoid costs via $\vec{H}_{\vec{z}}^+$, $\vec{H}_{\vec{z}}^-$, respectively. Otherwise, the cluster based costs, via $\hat{H}_{\vec{z}}^+$, $\hat{H}_{\vec{z}}^-$, respectively, are retained as shown in Equations (11.5) and (11.6):

$$\hat{H}_{\vec{z}n}^{+*} = \begin{cases} \vec{H}_{\vec{z}n}^+, & \text{if } \exists \vec{z} : \text{rref}(\mathbf{G})_{\vec{z}\vec{z}} l_z^G > l^{\min}, \\ \hat{H}_{\vec{z}n}^+, & \text{otherwise,} \end{cases} \quad (11.5)$$

and:

$$\hat{H}_{\vec{z}n}^{-*} = \begin{cases} \vec{H}_{\vec{z}n}^-, & \text{if } \exists \vec{z} : \text{rref}(\mathbf{G})_{\vec{z}\vec{z}} l_z^G > l^{\min}, \\ \hat{H}_{\vec{z}n}^-, & \text{otherwise,} \end{cases} \quad (11.6)$$

with $\vec{z} \in \{1, \dots, \vec{Z}\}$ and where $H_{\vec{z}n}^{+*}, H_{\vec{z}n}^{-*}$, denote the connectoid costs applied in assignment. Results are provided in Figure 11.12. Note that $l^{\min} = 1$ signifies that only the largest zone component adopts its original zone component cost while all other zone components in the cluster adopt the cluster based cost. It therefore does not provide the same result as the original cluster based connectoid cost.

We find that link results are fairly robust across the different values of l^{\min} with reduced RMSE for link usage ($\approx 0.5\%$) when adopting more cluster oriented connectoid costs. Conversely, travel time results improve for zone component oriented connectoid costs with low thresholds. We also see that some threshold based results ($l^{\min} < 0.3$) outperform the fixed zone component connectoid cost approach on travel times, some threshold based results outperform the average approach on link results ($l^{\min} > 0.4$), but not both at the same time. Comparing results against the status-quo case, we find link RMSE roughly on par or slightly better, while travel times are more accurate across all explored threshold settings

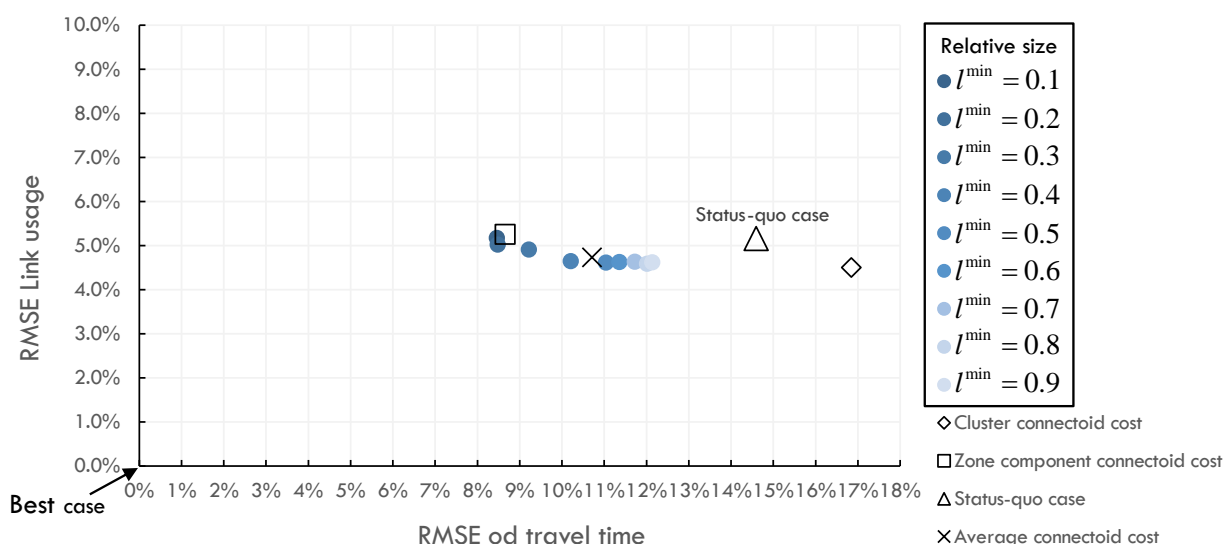


Figure 11.12: Pareto front of imposing various relative size thresholds in adopting cluster or zone component connectoid costs on RMSE link usage and RMSE travel times.

The (absolute) overestimation of travel times has reduced significantly under the modified connectoid costs as can be seen in Figure 11.13, confirming that connectoid costs play an important role in the total path travel times (in this case study), and more importantly, we can use path travel time information as an indicator to improve connectoid cost travel time estimates.

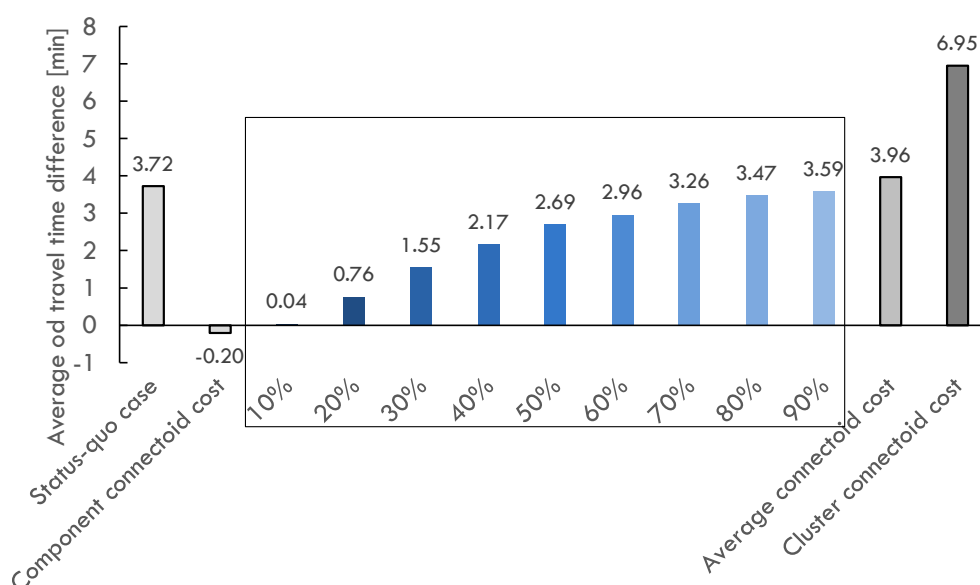


Figure 11.13: Average origin-destination travel time differences compared to best-case for additional approaches under different relative size thresholds l^{\min} .

11.3.2 Service area scaling factor

The choice between zone component and cluster level connectoid costs, as discussed in the previous section, is calibrated on the network level and considers differences between zone component and cluster. It does however not consider connectoid specific characteristics such as to what extent we expect a connectoid to be used in the presence, or absence, of other connectoids within the cluster. The service area scaling factor, see Section 9.6.7, attempts to

scale back a connectoid's cost estimate whenever we expect that trips originating farther away from the connectoid will underutilise the connectoid compared to the base estimate. The service area scaling factor is conditional on the number of connectoids in the cluster that we consider eligible of capturing trips originally attributed to the connectoid under consideration and therefore reducing its service area. The more such connectoids we find, the more the connectoid cost is scaled back. Given it is difficult to justify this choice without any calibration, we explore a number of scenarios, as outlined in Table 11.2. In each scenario we created one or more bins, where each bin got assigned the service area scaling factor of the minimum value in the bin, i.e. in scenario B, if a cluster has three connectoids a scaling factor of 0.91 is assigned, while if the cluster has five connectoids, it gets assigned the service area scaling factor of 0.78, which is in fact the value based on having four connectoids etc.

Table 11.2: Explored scenarios¹³ for service area scaling factors.

| Scenario | Number of cluster connectoids | Service area scaling factor | RMSE Link best-worst (1-4) | RMSE travel time best-worst (1-4) |
|----------|-------------------------------|-----------------------------|----------------------------|-----------------------------------|
| A | ≥ 4 | 0.78 | No improvement | 4 |
| B | 3 | 0.91 | 1 | 3 |
| | ≥ 4 | 0.78 | | |
| C | 3 | 0.91 | No improvement | 1 |
| | 4 | 0.78 | | |
| | 5 | 0.69 | | |
| | ≥ 6 | 0.62 | | |
| D | 3 | 0.91 | No improvement | 2 |
| | 4 | 0.78 | | |
| | 5 | 0.69 | | |
| | 6 | 0.62 | | |
| | ≥ 8 | 0.53 | | |

Without discussing each individual result, we found that travel times do improve for all scenarios considered, but that link usage estimates improved only for scenario B while compromising all other scenarios. The cause of deteriorating link estimates may be found in that large zone components (with many connectoids) are scaled back more, making fringe connectoids very attractive allowing trips to bypass significant portions of the physical road network. While travel times may improve in such a scheme, physical link flows within the larger zones are reduced causing additional error. To ensure our method contributes only positively to the reduction in error, we recommend a conservative approach and only apply scenario B.

The service area scaling factor is originally formulated on the level of zone components, yielding $\tilde{\mathcal{X}}^I \in \mathbb{R}^{\tilde{Z} \times N}$. Given that we apply the connectoid cost calibration as a post-processing step, we construct the service area scaling factors on the cluster level instead, denoted $\hat{\mathcal{X}}^I \in \mathbb{R}^{\hat{Z} \times N}$. This is a straightforward conversion and is provided in Appendix A. The connectoid costs used in assignment are then simply constructed via $\hat{\mathbf{H}}^{+*} \circ \hat{\mathcal{X}}^I, \hat{\mathbf{H}}^{-*} \circ \hat{\mathcal{X}}^I$, respectively.

¹³ Each of the scenarios was investigated with $l^{\min} = 0.5$.

Results, in conjunction with the various configurations of l^{\min} , are provided in Figure 11.14. As can be seen RMSE of travel times improve significantly, especially for values of l^{\min} that yield good link estimates. Link use estimates remain roughly the same, with slight improvements for values of l^{\min} yielding the best travel time estimates. We still find that there is a trade-off between obtaining slightly better link results (cluster oriented connectoid costs) versus better travel times (zone component oriented results), but applying service area scaling factors demonstrably improve overall model performance.

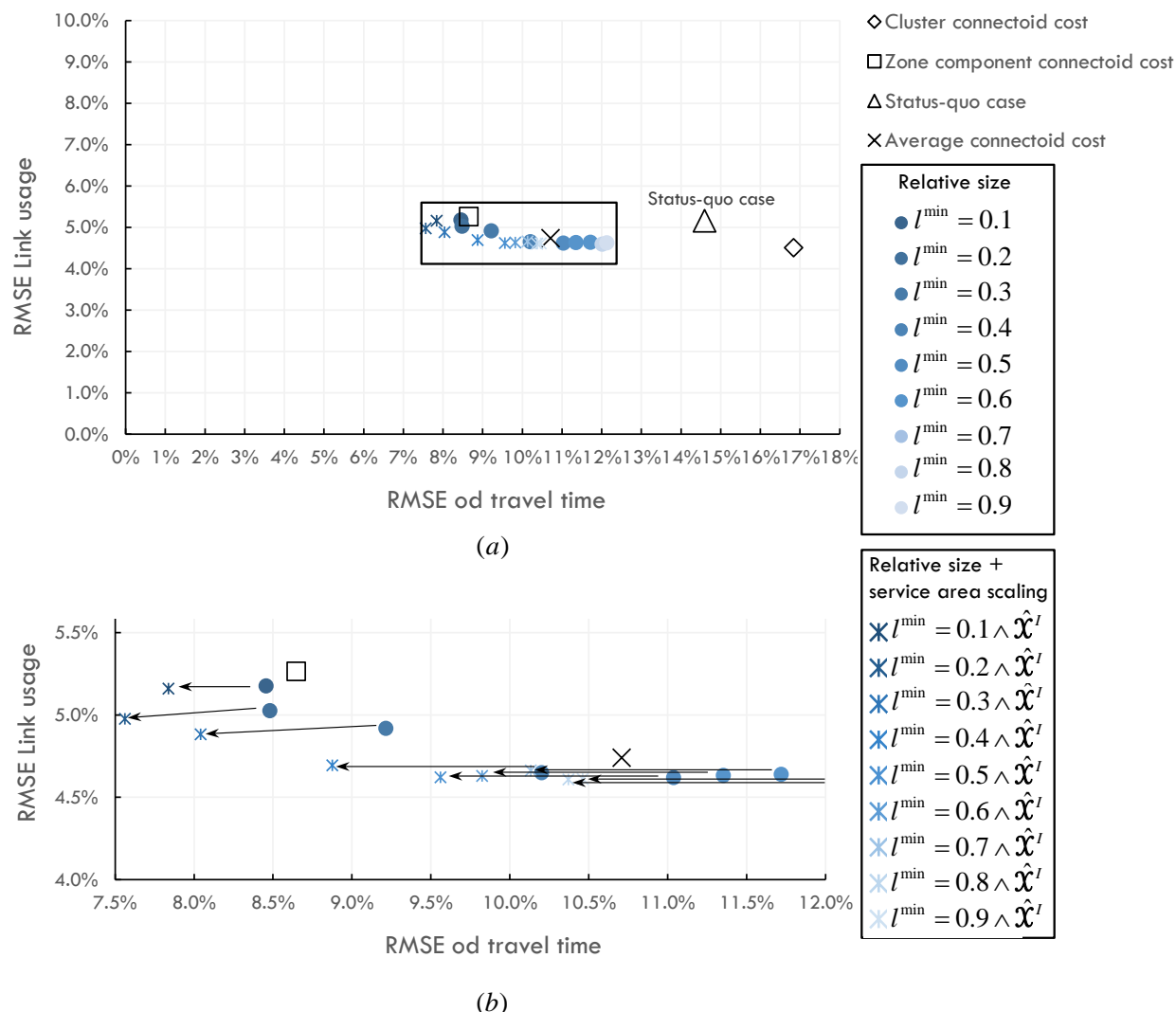


Figure 11.14: (a) Improvements in RMSE by including service area scaling factors $\hat{\chi}^l$, based on scenario B (b) close-up of per estimate improvement.

11.3.3 Centrality scaling factor

The third adjustment to the connectoid cost pertains to the centrality scaling factor, as originally formulated in Section 9.6.8. This measure scales back the connectoid cost, based on the expected double counting that might occur when combining connectoid costs with path costs. We argued that this double counting effect becomes more pronounced when connectoids reside closer to the zone centre, hence we apply this scaling progressively conditional on the connectoid location. Recall that the centrality scaling factor χ^{\min} , for the zone centre (Equation 7.35) relies on the choice of $\iota \in [0, 1]$. This factor ι dictates the portion of trips that approaches

their cluster destination from their ideal point of access, e.g. when the destination is on the Eastern side of the zone, they are expected to arrive via a road entering from the East. Conversely, the $1-\iota$ portion of trips is assumed to arrive uniformly from all directions. The combination of the two portions determines χ^{\min} and therefore the centrality scaling factors $\hat{\mathcal{X}}^{\prime\prime}$. Given we do not know ι , we must estimate it. For each estimate we then construct final connectoid cost estimates $\hat{\mathbf{H}}^{+*} \circ \hat{\mathcal{X}}^{\prime} \circ \hat{\mathcal{X}}^{\prime\prime}$, $\hat{\mathbf{H}}^{-*} \circ \hat{\mathcal{X}}^{\prime} \circ \hat{\mathcal{X}}^{\prime\prime}$, respectively. Note that the original centrality scaling factors, like the service area scaling factor, are formulated on a zone component basis, i.e. $\vec{\mathcal{X}}^{\prime\prime} \in \mathbb{R}^{\bar{Z} \times N}$. The conversion from zone component based formulation to the cluster based formulation $\hat{\mathcal{X}}^{\prime\prime}$ is also provided in Appendix A for the reader's convenience. We also point out that for each value of $\iota \in [0.0, 0.1, 0.2, 0.3, 0.4, 0.6, 0.8]$, we explored $l^{\min} \in [0.1, 0.2, \dots, 0.9]$ resulting in a total of $7 \cdot 9 = 63$ data points, where we left out some higher values of ι since it is highly unlikely that more than 50% of all trips approach a zone from the ideal entrance. Results are depicted in Figure 11.15.

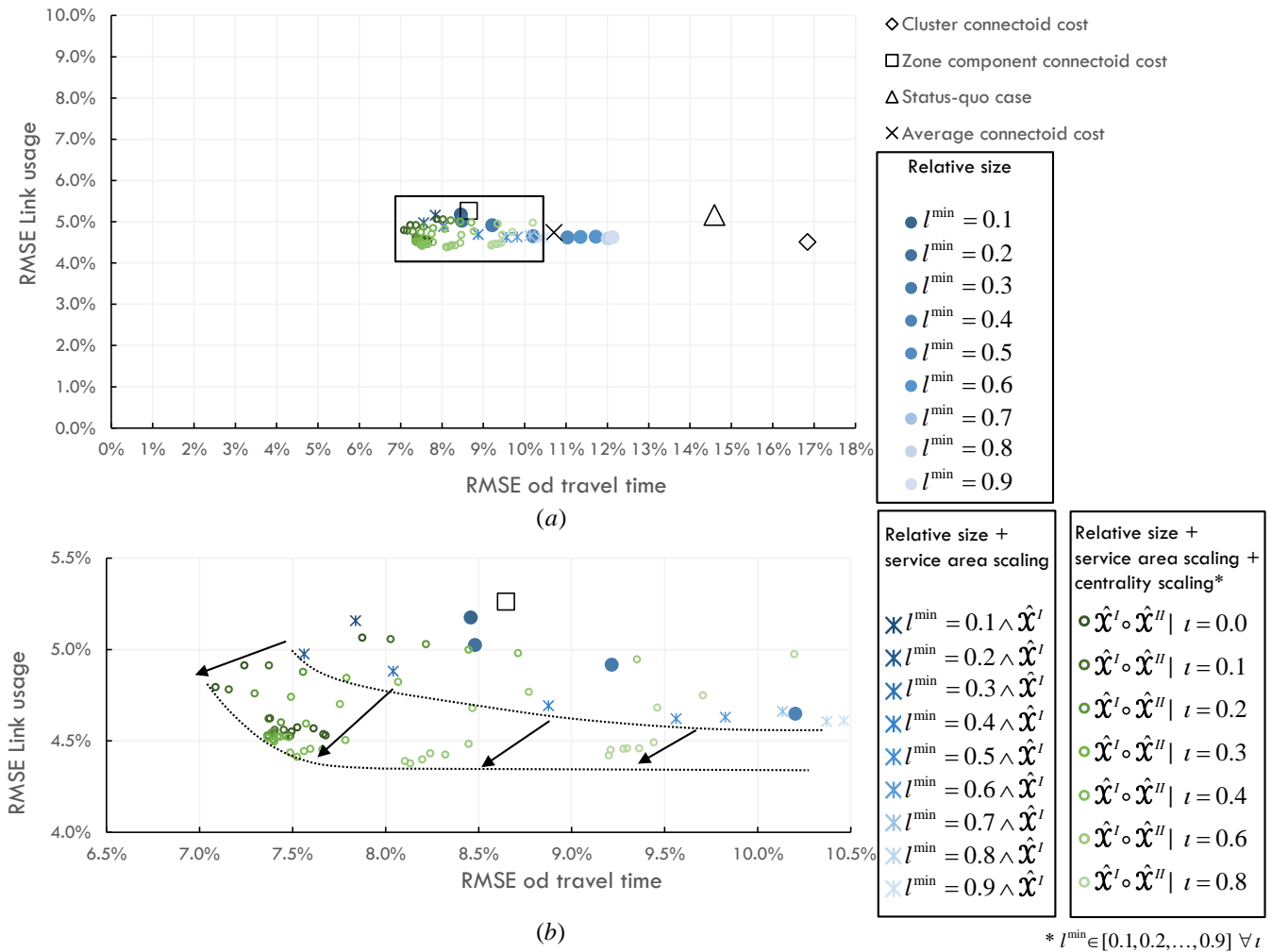


Figure 11.15: (a) Results including centrality scaling factors, (b) close-up of Pareto front improvements.

We find that the best travel time and link RMSE estimates improve noticeably, although not all combinations of parameters exhibit improvements. Upon closer inspection we find that $\iota \leq 0.2$, with $0.3 \leq l^{\min} \leq 0.4$, yields the best travel time RMSE of $\approx 7.1\%$. These settings also provide decent link usage RMSE estimates of $\approx 4.6\%$. However, the best link use RMSE

estimates are found to be around $\approx 4.4\%$ requiring $\iota > 0.3$, with $l^{\min} > 0.5$. Yet, these latter settings result in a noticeable deterioration of the travel time RMSE estimates ($> 7.5\%$).

11.3.3.1 Reducing accessible connectoids

While the results of using the centrality scaling factors are demonstrably good, Figure 11.15 does reveal there remains some trade-off between acquiring the best possible travel times versus acquiring the best possible link estimates. Also, ideally, we would prefer to find less scatter in our results to make our choice of parameters more robust. Generally though, we find that when ι is high, then χ^{\min} becomes smaller, and link load estimates improve. Meaning that, by making connectoids close to the centre more attractive (by scaling back their connectoid costs more aggressively), we increase usage of links within the zone, due to more paths preferring centrally located connectoids over fringe connectoids, leading to less bypassing of infrastructure and better link load estimates.

To obtain our best path travel time estimates, we do not scale back centrally located costs as aggressively as we do for our best link estimates. In these cases, fringe connectoids are still relatively attractive and a significant portion of the internal zone structure can be bypassed. At the same time, fringe connectoid costs are generally higher than internally located connectoids, but not high enough to deter travellers from using them. Disallowing the usage of the most costly connectoids, i.e. the connectoids at the fringes, should therefore improve link estimates while, due to our earlier calibration efforts, path travel times should remain mostly unaffected. To test this hypothesis, we perform one additional study where we only mark the four cheapest connectoids as accessible while imposing an infinite connectoid cost on the remaining connectoids (if any). The number of four is somewhat arbitrary, but the idea is that, for travel times, it is largely irrelevant how many connectoids exist as long as there is roughly one available per main wind direction, hence our choice of four. The results are depicted in Figure 11.16.

As can be seen the scatter is reduced drastically compared to Figure 11.15, with all data points within 0.45% RMSE regarding link usage (4.3%-4.75%) and within 1.8% travel time RMSE (7.2%-9.0%). So, even a suboptimal choice of parameter settings is less likely to drastically affect the result in a negative way. We also find that there is less of a trade-off as before. The Pareto front is now close to invariant to the link usage estimate, due to forcing paths to choose centrally located connectoids.

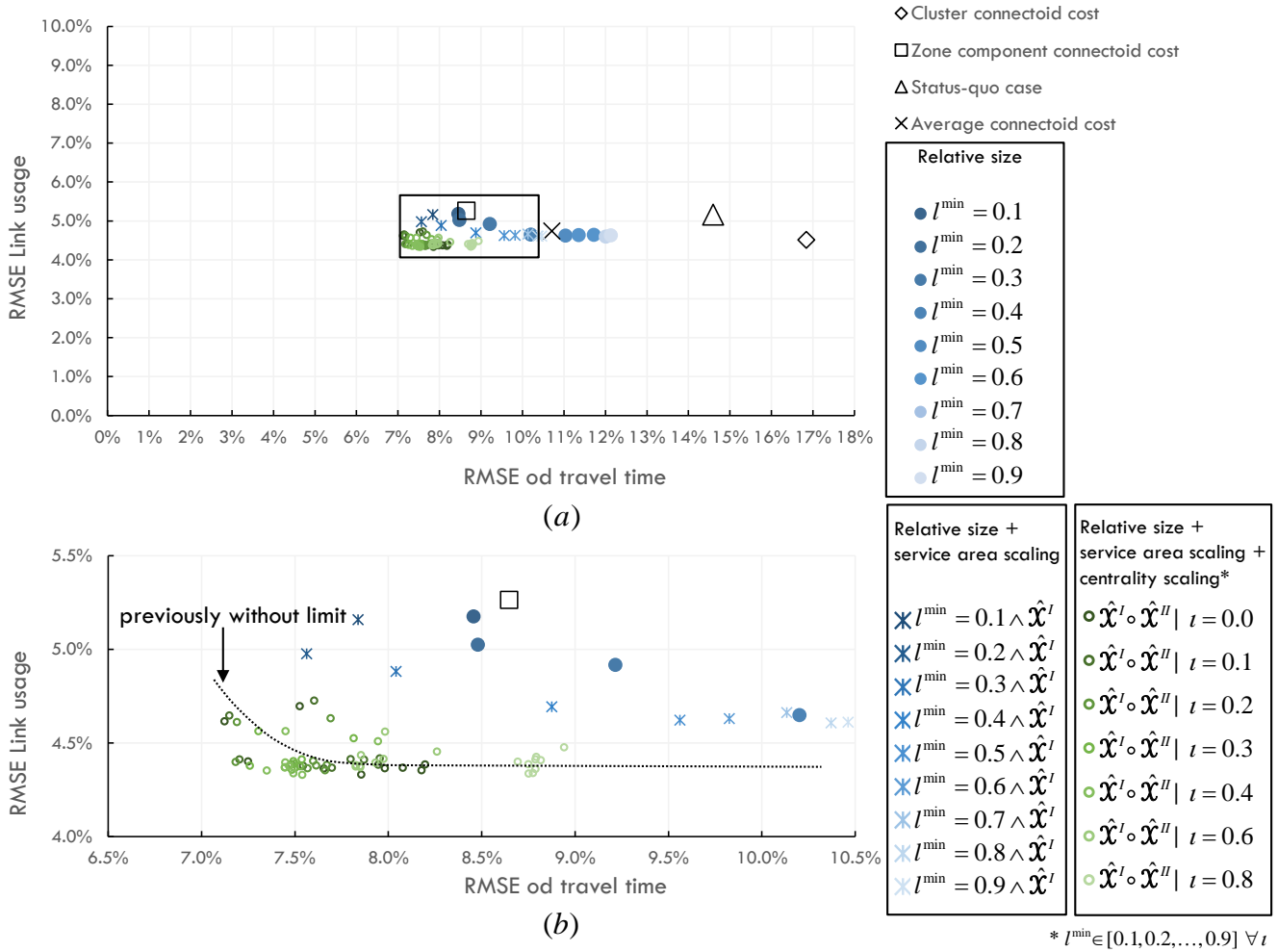


Figure 11.16: (a) Link load estimate improvements by excluding high cost connectoids (b) close-up showing Pareto front without imposing limit on connectoids.

Table 11.3 outlines the top 5 results based on uniform weighting of the two metrics. The best link usage RMSE remain roughly at $\approx 4.4\%$, but this estimate can now be obtained in combination with a travel time RMSE of 7.18%, only slightly above our overall best travel time RMSE estimate of 7.12%.

Table 11.3: Top 5 results when imposing limit on connectoids, including service area scaling (Scenario B), relative size factor and centrality scaling factors.

| Result | Travel time RMSE | Link usage RMSE | Centrality factor | Relative size threshold |
|--------|------------------|-----------------|-------------------|-------------------------|
| 1 | 7.18% | 4.40% | $t = 0.2$ | $l^{\min} = 0.3$ |
| 2 | 7.20% | 4.41% | $t = 0.1$ | $l^{\min} = 0.3$ |
| 3 | 7.26% | 4.38% | $t = 0.3$ | $l^{\min} = 0.3$ |
| 4 | 7.25% | 4.4% | $t = 0.0$ | $l^{\min} = 0.3$ |
| 5 | 7.35% | 4.35% | $t = 0.4$ | $l^{\min} = 0.3$ |

Intuitively, the settings for our best result seem reasonable as well, since $t = 0.2$ represents a maximum of 40% of trips approach destinations from the ideal direction; 20% directly via $t = 0.2$ and another 20% via the uniformly assumed arrival, i.e. $(\frac{1}{4}(1 - 0.2))$. The $l^{\min} = 0.3$ reflects that zone components that make up less than 30% of largest zone component should

revert to using the cluster connectoid cost estimate to deter their overly attractive connectoids to be utilised. At the same time, zone components larger than 30% of the largest zone component in the cluster benefit from not being influenced by outliers in cluster estimates (causing overestimation) and have sufficient size for their zone component based connectoid cost to obtain a decent estimate. We do note that the choice of $l^{\min} = 0.3$ seems more important than the choice of centrality factor. Figure 11.17 depicts our best calibrated result against the status-quo case and our earlier attempts. While the comparison with the status-quo scenario is subjective, it confirms a fully automated method is capable of outperforming this scenario to the point that travel time RMSE decreases with 50.8%, while link RMSE reduces by 14.8%.

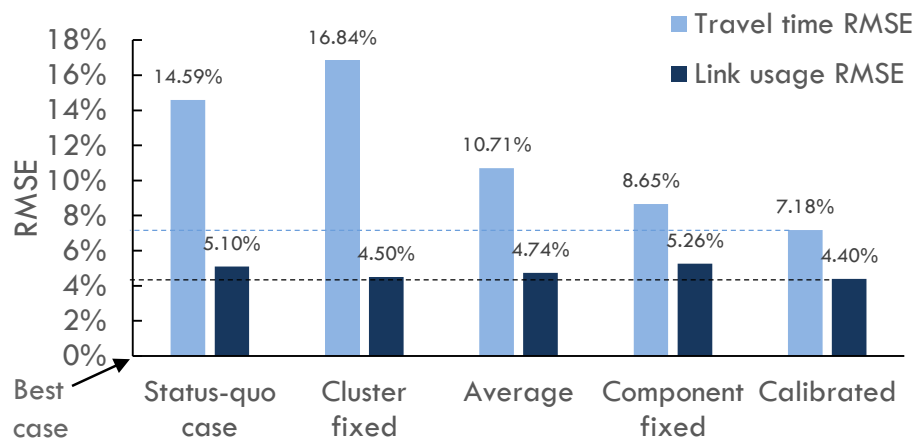


Figure 11.17: Overview of results compared to calibrated parameter estimate.

11.3.4 Verifying parameter estimates under a different zoning system

To verify if the found parameter estimates transfer to other zone granularities we now set $d^{\min} = 100$ (veh/h), ceteris paribus. In this context, this means more and smaller zones at roughly half the size compared to the original zoning, since the original zoning system averaged ≈ 235 produced trips and ≈ 221 attracted vehicle trips per zone per hour. One would also expect link load estimates, as well as travel time estimates, to improve due to the increased detail in the underlying zoning system.

Note that, to be able to compare these results with the status-quo case, we aggregated and averaged the travel times of our, now more disaggregate, zoning system to the granularity of the original zoning system. We adopted the same service area scaling approach as before as well as reducing the number of eligible connectoids as discussed in the previous section. Results for the combinations of ι , i.e. centrality scaling, and l^{\min} , i.e. relative size based connectoid cost configuration, are depicted in Figure 11.18. We find that the best link usage RMSE improve from $\approx 4.4\%$ to a RMSE of $\approx 3.4\%$. We also find improved RMSE for the best travel time estimates from $\approx 7.2\%$ to $\approx 6.8\%$.

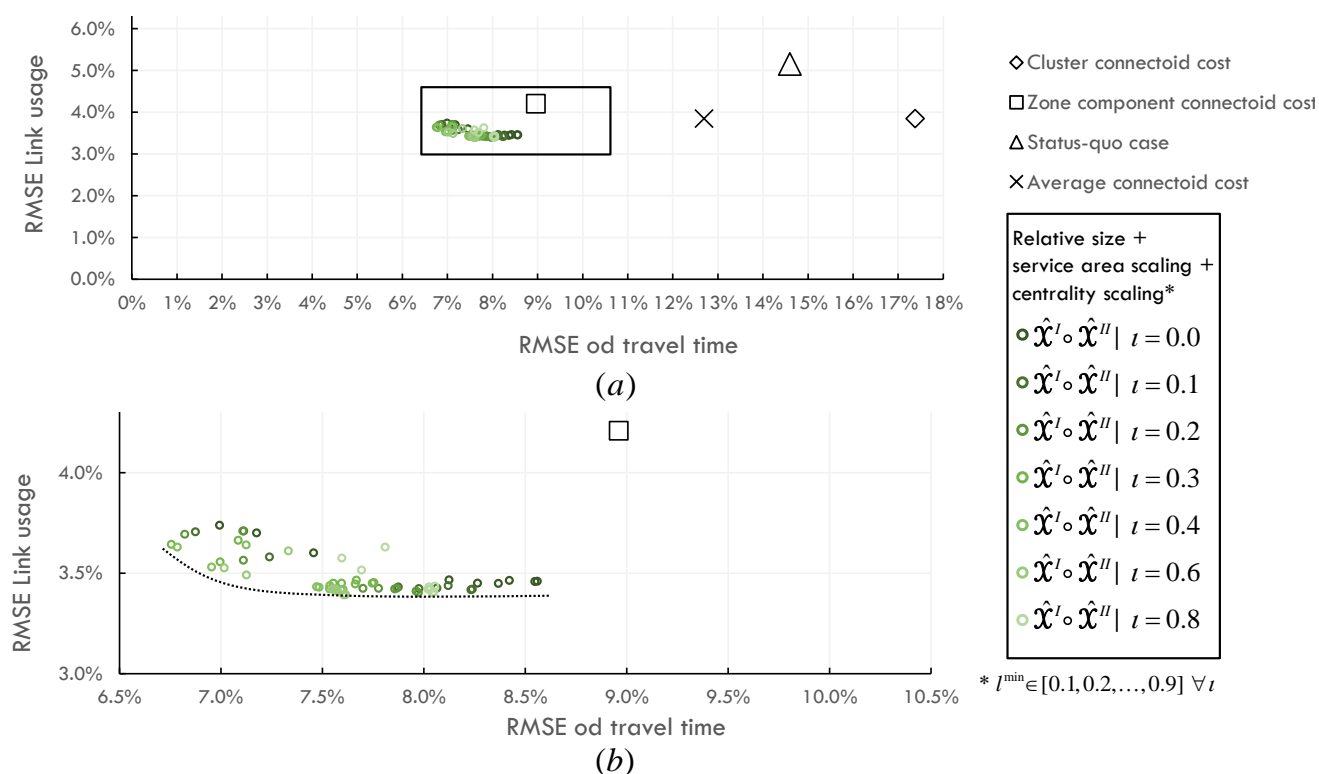


Figure 11.18: (a) Increased zonal detail leading to improved link and travel time estimates, (b) close-up of Pareto front.

Note that the RMSE for the status-quo case differ from the earlier values. This is not because results have changed, but due to a change in the set of links that is considered comparable (Section 11.1.1); the different zoning system results in additional connectoids emerging around newly retained boundary nodes leading to different comparable links and resulting in this change. Table 11.5 outlines the top 5 results, based on uniform weighting of the two metrics.

Table 11.4: Top 5 results conditional on $d^{\min} = 100$, when imposing limit on connectoids, including service area scaling (Scenario B), relative size factor and centrality scaling factors.

| Result | Travel time RMSE | Link usage RMSE | Centrality factor | Relative size threshold |
|--------|------------------|-----------------|-------------------|-------------------------|
| 1 | 6.76% | 3.64% | $\iota = 0.3$ | $\iota^{\min} = 0.2$ |
| 2 | 6.78% | 3.63% | $\iota = 0.4$ | $\iota^{\min} = 0.2$ |
| 3 | 6.95% | 3.53% | $\iota = 0.4$ | $\iota^{\min} = 0.3$ |
| 4 | 6.82% | 3.69% | $\iota = 0.2$ | $\iota^{\min} = 0.2$ |
| 5 | 6.87% | 3.71% | $\iota = 0.1$ | $\iota^{\min} = 0.2$ |

Compared to the results in the previous section we see some subtle differences. There is a shift from consistently choosing $\iota^{\min} = 0.3$, to now sometimes preferring $\iota^{\min} = 0.2$. This might be explained by the fact that when zones get smaller the connectoid cost based on the original zone component becomes relatively more representative, compared to when the zones are larger, hence accommodating this shift. The choice of centrality factor, again, plays a less significant role, although in general we obtain better results with slightly higher values of ι than before, with the best result shifting from $\iota = 0.2$ to $\iota = 0.3$. This is perhaps due to the fact that with smaller zones, the reduction of the number of connectoids to four has less effect because there are less connectoids per zone. In that case fringe connectoids remain active more

often and a more aggressive scaling is required to attract paths to the more centrally located connectoids.

Overall, we see results as expected here, with improved link usage and travel time estimates. Also parameter settings seem to fall within similar ranges and exhibit only moderate scatter, suggesting there is potential for these settings to be transferable to other networks and levels of detail.

11.4 Performance

While the main objective of the representations constructed through our disaggregation-aggregation framework has been to guarantee consistency, rather than optimising performance, we do want to at least provide some insight as in how the various steps compared in terms of computational costs as well. All methods are implemented in C++ as part of the StreamLine assignment framework within the OmniTRANS transport planning suite. Table 11.5 provides indicative, per step, computation times based on our prototype implementation¹⁴ that we found when constructing the representation used in our two case studies.

The disaggregate assignment is currently the mostly costly component which, even for such a moderately small network, already takes 3 minutes for just a single AON assignment. The link classification is computationally on par with the AON assignment since each path is revisited to identify the connectivity-keep links. The branch-and-bound solution scheme is, under the current settings, easy to solve. It should be noted however that relaxing the can-link constraints, is exponentially increases the solution space making it much more difficult to solve optimally. This situation can for example arise when one desires a coarse zoning system while the input zoning system is more detailed. Fortunately, in a practical setting, the opposite is currently more common.

Table 11.5: Indicative computation times for Amsterdam II case study.

| Framework step | Implementation method | Computation time |
|--|--|------------------|
| Step 1-2: Source model construction | AON node-to-node assignment | ≈ 3 min |
| Step 3: Supply representation | v/c ratio based link classification | ≈ 3 min |
| Step 4: Demand-supply interface | connectoid placement, base connectoid cost estimation | ≈ 1 min |
| Step 5 : Demand representation | Branch-and-bound Algorithm 1 | ≈ 2 min |

11.5 Synthesis and discussion

The Amsterdam II network (and demand) served as input for our case studies. While network design problems inherently lead to subjective results as to what can be considered a “good”

¹⁴ Computation times obtained based on following setup: 32bit C++ MS Visual Studio 2008 SP1 compiler, 16 GB RAM, Intel Xeon 3.1 GHz, Windows7 SP1.

outcome, we argue that by utilising two reference scenarios as a lower and upper bound, sufficient insight has been provided to draw some conclusions on the effectiveness of our method. Further, we calibrated the parameters proposed in Chapter 9 based on two RMSE based metrics, one metric related to path travel times, while the other focussed on link usage.

As concluded in Part II already, it is typically better to retain a significant amount of detail in the supply representation because it has relatively little impact on computation times, while missing critical infrastructure can quickly compromise assignment results. Following this line of reasoning, and based on the findings resulting from the disaggregate assignment, we advise a keep-link v/c ratio threshold of $\kappa^{\min} \leq 0.1$, irrespective of the desired level of detail in zoning. In this case it leads to retaining 25% of the original network, which expands to 41% once we also include the additional non-keep links needed to allow the connectoids to access the physical road network.

The zone components resulting from the link classification are used to obtain the base estimated for connectoid costs. When comparing our final representation with a best-case scenario (fully disaggregate assignment) and a status-quo scenario (original strategic planning network Amsterdam I), this initial case study outperformed the status-quo scenario with identical zoning based on link load accuracy, but lagged behind in travel times.

The second case study was used to calibrate the base estimated for connectoid costs to improve upon the link level usage and zone-to-zone travel time accuracy. Three different methods have been explored, each of which required some form of parameter calibration. The optimal parameter settings were found to be:

- Connectoids residing in an original zone components that comprises more than 30% of the infrastructure of the largest zone component in the cluster, i.e. $l^{\min} = 0.3$, should adopt the original zone component based connectoid cost because it is best representing the cost to access/egress the zone. Otherwise, the cluster based connectoid cost is to be adopted, which is higher, and deters paths from using the connectoid in order to avoid underestimation of the travel time.
- Service area scaling factors are applied to adjust the original connectoid cost estimate to the portion of the zone it is actually servicing. We found that the settings based on Scenario B yielded the best results, meaning that whenever a cluster has three connectoids, its cost is scaled back to 91% of its original cost, when it has four or more connectoids the connectoid cost is scaled back to 78% of its original cost, while otherwise its cost remains unchanged.
- To account for double counting of path travel times leading up to a connectoid and the cost imposed by the connectoid, a centrality scaling factor is applied to adjust the original connectoid cost estimate, leading to improved results in conjunction with service area scaling factors. where we found the top three results for $\iota \in [0.1, 0.2, 0.3]$.

- Reducing the number of connectoids to the four cheapest connectoids can lead to a further improvement of link load estimates without compromising travel time estimates.

Applying the optimal parameter settings found, our proposed methodology yields a 50.8% improvement regarding the travel time RMSE and a 14.8% improvement in link RMSE in comparison to the status-quo scenario. Note that the main difference between these scenarios is found solely in the construction of the demand-supply interface where the ad-hoc centroid/connectors have been replaced with a connectoid based approach which is fully automated.

Lastly, the second case study explored setting the desired zonal demand to a smaller value of $d^{\min} = 100$ to increase the number of zones found by the branch-and-bound algorithm. It was found that indeed link and travel time estimates improved, as you would expect. Also, the optimal parameter settings remained largely the same, with $l^{\min} \in [0.2, 0.3]$ and the top three results adopting $\iota \in [0.3, 0.4]$. The slight decrease in l^{\min} and slight increase in ι is likely due to the differently chosen granularity, suggesting that the found settings might be somewhat dependent on the desired level of detail. Yet, we also found that by simply applying the proposed scaling methods already results in marked improvements over the base estimates as well as the status-quo scenario, regardless of the chosen parameter settings that accommodate further fine-tuning.

11.5.1 Model limitations and extensions

The results presented in this chapter demonstrate the suitability of both the disaggregation-aggregation framework itself as the methods proposed for each step. However, due to the complexity of combining these different methods, in a practical setting, as well as calibrating the parameters involved, there are inevitably some limitations that require addressing. Most of all, there is a need for exploring additional case studies to be able to draw stronger conclusions.

The presented case studies took a considerable amount of time and effort to put together. Yet we would like to conduct additional tests to further strengthen our findings. These additional tests would mainly involve:

- Explore different original travel demand scenarios to investigate the effect of demand impacting the supply representation when fixing κ^{\min} .
- Explore the impact of the granularity of original zoning systems on the final zoning system while keeping parameter settings fixed.
- Explore the impact on computation times when varying model inputs, i.e. does a larger model scale linearly, exponentially, and how feasible is it to solve the problem optimally under other circumstances?
- Explore the impact on computation times on varying desired zone granularity, especially when original zoning is more detailed while the desired output is more aggregated.

- Explore heuristics that can guarantee competitive computation times of the constrained optimisation problem on large networks and compare how well they perform against the optimal branch-and-bound algorithm.

Part IV

12 Conclusions

In this chapter we draw conclusions based on the combined findings of Part II and Part III. Each part focussed on a specific topic regarding the representation of traffic assignment models. In Part II computational optimisation was the objective and we used the construction of an alternative (lossless) representation as the way to achieve this goal. In Part III we focussed on consistency in the representation of traffic assignment models in a multi-scale environment and proposed an integrated disaggregation-aggregation framework to achieve this.

In Section 12.1, we briefly reiterate the background and motivation for each of the two parts, followed by the conclusions of Part II and possible extensions in Section 12.2 and Section 12.3, respectively. Then, we do the same for Part III, drawing conclusions in Section 12.4 and summarise possible extensions in Section 12.5.

12.1 Overview

Reducing the computational cost of traffic assignment methods has long resulted in either choosing a simpler model, for example replacing a dynamic model with a static model, or apply some kind of simplification on one of the model inputs. At the same time, there exist very few methods that explicitly consider the application context of the model and remove redundant detail based on how the model is used. We argue that the latter approach can be much more effective in achieving a reduction in computational cost. The main reason for this is found in the fact that the context information of the targeted application can be used to reduce information loss while it can also be used to maximise the simplification. In Part II we proposed such a tailored method, targeted at applications where the supply is fixed while demand may vary. Typical applications that would benefit from such an optimisation procedure are quick-scan methods, matrix calibration procedures, or applications that consider demand variability. All of these applications explore a large number of different demand scenarios under a fixed network and can benefit from our findings.

Besides optimisation, consistency can be of similar importance, if not more important, when constructing traffic assignment representations. With the increasing popularity of multi-scale environments, where multiple models operate alongside each other, it has become paramount that model results are consistent. This is needed to be able to attribute differences in the obtained results between models to certain simplifying assumptions made. Without fully consistent models, this is simply not possible. To date, as far as the author is aware, no methods existed to guarantee this consistency. In Part III we provided both methodology as well as a number of conditions regarding the construction of consistent traffic assignment inputs in a multi-scale context.

We hope that providing novel methodology to reduce the computational cost of traffic assignment models as well as designing consistent model representations across different levels of detail contributes to a more effective as well as a more responsible use of traffic assignment models both in theory and in practice.

12.2 Conclusions Part II

Based on the findings in Part II we draw the following conclusions:

- 1) Existing aggregation and decomposition methods in traffic assignment are mostly focussed on properties of the traffic assignment procedure, rather than the application they are used for.
- 2) Explicitly considering the application context opens up new possibilities for traffic assignment optimisation that otherwise would not be possible.

We choose a particular context, where the supply side is fixed while demand may vary. Further, when adopting a static capacity constrained traffic assignment model utilising a triangular fundamental diagram and a fixed a-priori path set, we conclude that:

- 3) One can decompose the original network in a free flow subnetwork and a delay subnetwork such that the combined path travel times from the two networks equate to the path travel time in the original network. The free flow network path travel times are invariant to flow and therefore do not require equilibration.
- 4) Equilibrating a delay subnetwork is less costly than equilibrating the original network since it only contains a subset of the original infrastructure. This also causes increased path overlap. When paths overlap completely, they can be aggregated into equidelay paths without any loss of information. The resulting consolidated path set can be used to replace the original path set in network loading significantly reducing computation times.
- 5) We demonstrated that replacing the original traffic assignment procedure with our decomposition based approach can reduce network loading computation times by 59% in our Gold Coast case study. When also using the consolidated path set this further improved to a reduction of 96%, on this same case study.
- 6) The proposed decomposition method is lossless when all bottleneck infrastructure across demand scenarios is contained in the delay subnetwork. To date, there exists no method able to construct a delay subnetwork while guaranteeing this condition is met. Yet, as our case study demonstrates, it is relatively straightforward to find delay subnetworks that contain virtually all bottleneck infrastructure. We also found that any missing infrastructure can be easily identified a-posteriori. Hence, in a practical setting a lossless result can always be achieved when needed.

12.3 Possible extensions Part II

The case studies that we considered in Part II adopted the static residual queuing model of Bliemer et al. (2014) in a single user class setting. However, this assignment model can easily be made suitable for multi-modal assignment and the proposed decomposition and path consolidation methodology are equally capable to handle multiple modes as well. Extending

our method to a multi-modal approach increases the applicability of the method. If the additional mode is a freight based mode, it is not expected that the results will be very different compared to the current results. The difference in vehicle speed will have little impact on bottleneck locations because in static assignment the propagation of flow is instantaneous. In case the additional mode(s) include public transport the impact might be more significant. Then, bus stops (train stations, tram stops) need to be retained, even if these locations reside on links that do not exhibit any delay. Therefore, the delay subnetwork might grow substantially. On the other hand, most bus lines reside on main roads and main roads are more likely to be already included in the delay subnetwork, so maybe there is not that much impact after all. We simply do not know at this stage and this would be worth pursuing.

In the context of matrix calibration, there are also some additional measures that one needs to consider. Calibration procedures rely on data sources linked to particular spatial locations such as loop detector data, probe vehicle path travel times, and/or average travel times for monitored corridors. This means that the delay subnetwork needs to include this additional infrastructure, not for delay purposes but to support the calibration procedure. This can be achieved by extending the link classification method with additional constraints, but is something we have not included so far.

It would also be interesting to see if the method can be extended, so it can deal with assignment models that are either dynamic, consider spillback, or both. Clearly, this would mean a larger delay subnetwork because of the (temporal) spillback that can occur, but it is our expectation that even under spillback the delay subnetwork would still be significantly smaller than the original network. Another potentially interesting area of research can be found in improving the methodology for constructing the super-scenario. Currently, it is based on a heuristic (taking the maximum origin-destination demand across demand scenarios) that in turn is fine-tuned by adopting another heuristic (relative flow margin) to minimise information loss (missed critical delay links). There is potential for improved methodology on this end, to obtain a more minimal delay subnetwork containing less false positive links.

12.4 Conclusions Part III

Based on the finding in Part III we draw the following conclusions:

- 1) There exist no methods in the literature regarding traffic assignment representation from an integrated demand and supply perspective. So far, most methods only consider a single, or at most a subset, of the model components.
- 2) There is hardly any literature on how to construct traffic assignment model representation in a multi-scale environment despite the fact that in practice it increasingly occurs that traffic assignment models operate alongside each other in a similar spatial domain.
- 3) We argue that, in our assessment on the procedural consistency of traffic assignment models, consistency can only be achieved when the following three conditions are

satisfied: (i) directional consistency, (ii) source consistency, and (iii) abstraction consistency.

A disaggregation-aggregation framework for constructing consistent traffic assignment model inputs is proposed. The original model inputs together with a procedural consistent model, satisfying aforementioned three conditions, can then be used to create traffic assignment representations in a multi-scale environment.

- 4) It is possible to construct an integrated aggregation method that considers all traffic assignment model inputs in unison: zoning system, trip demand, demand-supply interface, and the transport network. We demonstrated this by adopting a supply side perspective and adopting expected road usage as our base metric within the proposed disaggregation-aggregation framework. This supply perspective then supplements the original demand based methods that led to the original zoning system and demand-supply interface.
- 5) Original zoning systems hardly take supply side characteristics into account. By explicitly considering expected road usage as an additional metric we can alter the original zoning system. This can either lead to a more disaggregate representation, or a more aggregate one, depending on the desired granularity.
- 6) The traditional centroid/connector paradigm is based on a number of, arguably, poorly justified simplifications which are further compromised by lack of methodology. This results in issues regarding the placement of centroids, placement of connectors, connector cost estimation, as well as the number of connectors to consider. We replace the centroid/connector paradigm with an automated method to estimate the costs of the interaction between zone and physical network. This new method replaces connectors with connectoids and no longer requires the placement of centroids. It remains possible to implement our approach within the centroid/connector paradigm so it is compatible with existing implementations as well.
- 7) Estimating connectoid access/egress costs of a zone by computing the average cost from all disaggregate departure/arrival nodes within a zone (considering the complete underlying transport network) reveals an overestimation of the “true” path travel times compared to a full disaggregate assignment.
- 8) Employing two additional scaling methods estimating the double counting of path travel times as well as acknowledging that not all nodes internal to a zone are likely to use every connectoid can significantly mitigate the effects of aforementioned overestimation.
- 9) The Amsterdam case study demonstrated that our method, with additional scaling methods in place, compared to a strategic planning network of the city, resulted in markedly reduced zone-to-zone travel time and link usage RMSE estimates (50.8% and 14.8%, respectively) for a matching zoning system. In addition, compared to a fully disaggregate result, our coarser model exhibited 7.2% RMSE on travel time estimates

and 4.4% RMSE on link usage. This was achieved, while the number of links in the coarser model comprised only 41% of the disaggregate model links. The actual keep link network (excluding converted non-keep links retained to allow connectoids to access the keep network) comprises only 25% of the original network.

- 10) The RMSE on link usage improves further when restricting the use of connectoids with a high cost (typically close to the zone fringes). However, estimating connectoid costs for connectoids close to zone centres is more difficult due to double counting of travel times. It was found that additional measures are needed to calibrate this cost on top of the simple base cost estimate adopted originally.
- 11) When adopting our approach, i.e. complying with procedural consistency as well as consistent model inputs through our disaggregation-aggregation framework, results between different models at different granularities can be compared objectively. Differences can be attributed to (simplifying) assumptions made, due to the model assumptions being explicit, known and consistent across the models. This, in our view, is much more important than obtaining perfectly calibrated results and is in fact the most important contribution of this work.

12.5 Possible extensions Part III

There are numerous ways to extend the current methodology. Most notably, we only provided a single method for each of the steps in the disaggregation-aggregation framework. As stated before, we deliberately choose relatively basic methods for each of steps to demonstrate that even with simple methods good results can be achieved. Yet, more sophisticated methods are likely to yield even better results. Also, the premise so far has been that we have no access to empirical data to enhance some of the steps in the modelling procedure. It would be of interest to verify to what extent empirical data can improve (the predictive power of) the final model representations, for example by more sophisticated approaches to: disaggregating demand (Section 9.3.2), distribution of demand across connectoids (Section 9.3.3), or calibration of connectoid costs (Sections 9.6 to 9.8).

With respect to the link classification in the first three steps of the disaggregation-aggregation framework, it would be interesting to improve upon the AON assignment method to assess the impact of a more capable model, for example by adopting the capacity restrained model by Bliemer et al. (2014), see also Part II, or adopt an iterative procedure with a simplified disaggregate demand matrix. This prevents overestimation of flows downstream due to the inclusion of capacity constraints and would improve upon the path choice. Also, the found vertical queues can be transformed (post simulation) into physical (horizontal) queues to obtain a better estimate for upstream congestion caused by bottlenecks. Alternatively, one might be able to adopt a (one shot) macroscopic dynamic assignment model such as computationally efficient event based link transmission models (Raadsen et al., 2016), or iterative solutions (Himpe et al., 2016). Do note however that all these alternative approaches come at a significantly higher computational cost. A cost that, today, might still be unacceptable for medium to large transport model projects.

Regarding the demand-supply interface, we found that applying (methodologically justified) scaling factors based on specific properties of connectoids is quite successful, as the results in Chapter 11 confirm. However, the service area scaling factor is formulated on such an abstract level that it might be possible to improve on it. For example by employing some kind of deterrence function instead. Perhaps something similar can be done for the estimation of path cost double counting that we capture via the centrality scaling factor. We also would like to test if incorporating these two concepts in the optimisation problem formulation would yield any (dis)benefits.

The zoning system problem formulation is sufficiently general to accommodate any constraint formulation. However, we constructed some of our constraints rather crudely and in a way that can possibly be improved upon. For example, the similarity constraint in the clustering procedure is based on productions/attractions. This does capture some of the underlying land use homogeneity, but it fails to account for the possible heterogeneity in the distribution of trips across destinations. It can be expected that including such additional characteristics will only improve results further.

The branch-and-bound solution scheme to solve the constrained optimisation problem for the zoning system adopted a (search space) partitioning approach that is fairly basic. We are confident better partitioning algorithms (for example using spectral partitioning) can be designed, leading to much improved computation times, hopefully allowing the algorithm to scale up to much larger networks.

Finally, the branch-and-bound algorithm provides optimal solutions which, due to the strict constraints involved in the supply side zone refinement, allows us to solve optimally in the case studies explored. Unfortunately, it remains likely that, for (very) large models, solving the optimisation problem optimally becomes too costly at some point. The design of heuristic alternatives is therefore of interest. We can use our current case studies as benchmarks to verify how well these heuristic alternatives perform against the optimal solution.

Bibliography

- Aarts, E., Korst, J., Michiels, W., 2005. Simulated annealing. In *Search Methodologies* (pp. 187–210). Springer.
- AECOM Consult, Cambridge Systematics., 2007. A Recommended Approach to Delineating Traffic Analysis Zones in Florida.
http://www.fsutmsonline.net/images/uploads/reports/fr1_fdot_taz_white_paper_final.pdf
- Aghabayk, K., Sarvi, M., Young, W., 2015. A State-of-the-Art Review of Car-Following Models with Particular Considerations of Heavy Vehicles. *Transp. Rev.* 35, 82–105. doi:10.1080/01441647.2014.997323
- Akamatsu T, Makino Y, Takahashi E., 1998. Semi-dynamic traffic assignment models with queue evolution and elastic OD demand. *Infrastructure Planning Review*; 15: 535–545 (Japanese).
- Akçelik, R., 1991. Travel time functions for transport planning purposes: Davidson's function, its time dependent form and alternative travel time function. *Aust. Road Res.* 21, 49–59.
- Akçelik, R., Troutbeck, R., 1991. Implementation of the Australian roundabout analysis method in SIDRA. *Proc. Int. Symp. Highw. Capacit.* 17–34.
- Alpert, C. J., Yao, S., 1995. Spectral Partitioning: The More Eigen Vectors, The Better. In *Proceedings of the 32nd ACM/IEEE conference on design automation* (pp. 195–200). San Francisco.
- Añez, J., De La Barra, T., Pérez, B., 1996. Dual graph representation of transport networks. *Transp. Res. Part B Methodol.* 30, 209–216. doi:10.1016/0191-2615(95)00024-0
- Aw, A., Rasclé, M., Mathematics, A., 2000. Resurrection of “Second Order” Models of Traffic Flow. *SIAM J. Appl. Math.* 60, 916–938. doi:10.1137/S0036139997332099
- Baass K.G., 1981. Design of Zonal systems for aggregation transportation planning models, in: *Transportation Research Record*. p. 807 1-6.
- Bar-Gera, H., 2010. Traffic assignment by paired alternative segments. *Transp. Res. Part B Methodol.* 44, 1022–1046. doi:10.1016/j.trb.2009.11.004
- Basu, S., Bilenko, M., Mooney, R. J., 2004. A probabilistic framework for semi-supervised clustering. *Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '04*, 59–68.
<https://doi.org/10.1145/1014052.1014062>
- Beckmann, M., McGuire, C.B., Winsten, B.W., 1956. *Studies in the economics of transportation*. Yale University Press, New Haven CT, USA.

- Bekhor, S., Ben-Akiva, M.E., Ramming, S.M., 2002. Adaptation of Logit Kernel to Route Choice Situation. *Transp. Res. Rec. J. Transp. Res. Board* 1805, 78–85. doi:10.3141/1805-10.
- Bell, M.G.H., 1995. Stochastic user equilibrium assignment in networks with queues. *Transp. Res. Part B Methodol.* doi:10.1016/0191-2615(94)00030-4.
- Bell, M. G. H., Kurauchi, F., Perera, S., Wong, W., 2017. Investigating transport network vulnerability by capacity weighted spectral analysis. *Transportation Research Part B: Methodological*, 99, 251–266. <https://doi.org/10.1016/j.trb.2017.03.002>
- Benezech, V., 2011. A new model for disaggregate traffic assignment making explicit the spatial distribution of trip extremities. *European Transport Conference, Glasgow, October 2011.*
- Bifulco, G., Crisalli, U., 1998. Stochastic user equilibrium and link capacity constraints: formulation and theoretical evidencies, in: *Proceedings of the European Transport Conference.* pp. 85–96.
- Bliemer, M.C.J., 2007. Dynamic Queuing and Spillback in Analytical Multiclass Dynamic Network Loading Model. *Transp. Res. Rec.* 2029, 14–21. doi:10.3141/2029-02.
- Bliemer, M. C. J., Bovy, P. H. L., 2008. Impact of Route Choice Set on Route Choice Probabilities. *Transportation Research Record*, (2076), 10–19. <https://doi.org/Doi10.3141/2076-02>.
- Bliemer, M.C.J., Raadsen, M.P.H., 2017. Static traffic assignment with residual queues and spillback, 17th Swiss Transport Research Conference (STRC2017). Monte Verita.
- Bliemer, M.C.J., Raadsen, M.P.H., 2018. Continuous-time general link transmission model with simplified fanning, part I: Theory and link model formulation. *Transp. Res. Part B Methodol.* 0, 1–29. doi:10.1016/j.trb.2018.01.001
- Bliemer, M.C.J., Raadsen, M.P.H., 2018. Continuous-time general link transmission model with simplified fanning, part II: Event-based algorithm for networks. *Transp. Res. Part B Methodol.*
- Bliemer, M.C.J., Raadsen, M.P.H., Brederode, L.J.N., Bell, M.G.H., Wismans, L.J.J., Smith, M.J., 2017. Genetics of traffic assignment models for strategic transport planning. *Transp. Rev.* 37, 56–78. doi:10.1080/01441647.2016.1207211
- Bliemer, M.C.J., Raadsen, M.P.H., Smits, E.-S., Zhou, B., Bell, M.G.H., 2014. Quasi-dynamic traffic assignment with residual point queues incorporating a first order node model. *Transp. Res. Part B Methodol.* 68, 363–384. doi:10.1016/j.trb.2014.07.001
- Bovy, P., Jansen, G., 1983. Network aggregation effects upon equilibrium assignment outcomes: An empirical investigation. *Transportation Science*, 17(3), 240–261.
- Bovy P.H., 1991. Zusammenfassung des schweizerischen Kreisellhandbuchs, *Strabe und Verkehr*, nr. 3.

- Boyles, S. D., 2012. Bush-based sensitivity analysis for approximating subnetwork diversion. *Transportation Research Part B: Methodological*, 46(1), 139–155. <https://doi.org/10.1016/j.trb.2011.09.004>
- Brackstone, M., McDonald, M., 1999. Car-following : a historical review. *Transp. Res. Part F Traffic Psychol. Behav.* 2, 181–196.
- Bundschuh, M., Vortisch, P., van Vuuren, T., Mott McDonald, 2006. Modelling queues in static traffic assignment. *Eur. Transp. Conf. Proc.*
- Bureau of Public Roads (United States), Traffic assignment manual, 1964. Washington.
- Burghout, W., Koutsopoulos, H., & Andréasson, I., 2005. Hybrid Mesoscopic-Microscopic Traffic Simulation. *Transportation Research Record: Journal of the Transportation Research Board*, 1934, 218–255. <https://doi.org/10.3141/1934-23>
- Carey, M., 1987. Optimal time varying flows on congested networks. *Oper. Res.* 35, 58–69.
- Casas, J., Perarnau, J., Torday, A., 2011. The need to combine different traffic modelling levels for effectively tackling large-scale projects adding a hybrid meso/micro approach. *Procedia - Social and Behavioral Sciences*, 20, 251–262. <https://doi.org/10.1016/j.sbspro.2011.08.031>
- Cascetta, E., Nuzzolo, A., Russo, F., Vitetta, A., 1996. A modified logit route choice model overcoming path overlapping problems: Specification and some calibration results for interurban networks. *Transp. Traffic Theory*.
- Cascetta, E., 2009. *Transportation Systems analysis* (book).
- Chaker, W., Moulin, B., & Thériault, M., 2010. Multiscale Modeling of Virtual Urban Environments and Associated Populations. In *The Cartographic Journal* (Vol. 47, pp. 139–162). https://doi.org/10.1007/978-90-481-8572-6_8
- Chan, Y., Follensbee, K. G., Manheim, M. L., Mumford, J. R., 1968. Aggregation in transport networks: An Application of Hierarchical Structure. M.I.T. dept. Civil Engng Res. Report No R68-47.
- Chan, Y., 1976. A method to simplify network representation in transportation planning. *Transportation Research*, 10(3), 179–191.
- Chandler, R.E., Herman, R., Montroll, E.W., 1958. Traffic Dynamics: Studies in Car Following. *Oper. Res.* 6, 165–184. doi:10.1287/opre.6.2.165
- Chen, H.-K., 1999. *Dynamic travel choice models a variational inequality approach*. Springer-Verlag, Berlin.
- Cheng, T., Tanaksaranond, G., Emmonds, A., & Sonoiki, D., 2010. Multi-Scale Visualisation of Inbound and Outbound Traffic Delays in London. *The Cartographic Journal*, 47(4), 323–329. <https://doi.org/10.1179/000870410X12911311788152>

- Connors, R. D., Watling, D. P., 2008. Aggregation of Transport Networks Using Sensitivity Analysis. In Proceedings of the European Transport Conference. Association for European Transport.
- Connors, R. D., Watling, D. P., 2015. Assessing the Demand Vulnerability of Equilibrium Traffic Networks via Network Aggregation. *Networks and Spatial Economics*, 15(2), 367–395. <https://doi.org/10.1007/s11067-014-9251-9>
- Cui, Y., 2016. Defining the resolution of a network for transportation analyses: a new method to improve transportation planning decisions. phd dissertation. University of Maryland.
- Dial, R.B., 2006. A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration. *Transp. Res. Part B Methodol.* 40, 917–936. doi:10.1016/j.trb.2006.02.008
- Dafermos, S. 1980. Traffic Equilibrium and Variational Inequalities. *Transportation Science.* 14, 42-54.
- Daganzo, C. F., 1980. Network representation, continuum approximations and a solution to the spatial aggregation problem of traffic assignment. *Transportation Research Part B: Methodological*, 14(3), 229–239. [https://doi.org/10.1016/0191-2615\(80\)90002-8](https://doi.org/10.1016/0191-2615(80)90002-8)
- Daganzo, C., 1980a. An equilibrium algorithm for the spatial aggregation problem of traffic assignment. *Transportation Research Part B: Methodological*, 14(3), 221–228.
- Daganzo, C.F., 1994. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transp. Res. Part B* 28, 269–287. doi:10.1016/0191-2615(94)90002-7
- Daganzo, C.F., 1995. The cell transmission model, part II: Network traffic. *Transp. Res. Part B Methodol.* 29, 79–93. doi:10.1016/0191-2615(94)00022-R
- Daganzo, C. F., 2007. Urban gridlock: Macroscopic modeling and mitigation approaches. *Transportation Research Part B: Methodological*, 41(1), 49–62. <https://doi.org/10.1016/j.trb.2006.03.001>
- Daganzo, C., Geroliminis, N., 2008. An analytical approximation for the macroscopic fundamental diagram of urban traffic. *Transportation Research Part B*: 42(9), 771–781.
- Daganzo, C.F., Sheffi, Y., 1977. On Stochastic Models of Traffic Assignment. *Transp. Sci.* 11, 253–274. doi:10.1287/trsc.11.3.253
- Davidson, I., Ravi, S. S., 2005. Clustering with constraints: Feasibility issues and the K-Means algorithm. Of the Fifth SIAM International Conference, 138–149. <https://doi.org/10.1137/1.9781611972757.13>
- Davidson, P., Thomas, A., Teye-Ali, C., 2011. Clocktime assignment: a new mesoscopic junction delay highway assignment approach to continuously assign traffic over the whole day, in: European Transport Conference Proceedings.

- Ding, C., 1998. The GIS-based human-interactive TAZ design algorithm: examining the impacts of data aggregation on transportation-planning analysis. *Environment and Planning B: Planning and Design*, 25(4), 601–616. <https://doi.org/10.1068/b250601>
- Dorigo M., Birattari, M. T. S., 2006. Ant Colony Optimization. *A Computational Intelligence Technique*. *IEEE Computational Intelligence Magazine*, 1(4), 28–39. <https://doi.org/http://dx.doi.org/10.1109/CI-M.2006.248054>
- Eichler, D., Bar-Gera, H., Blachman, M., 2013. Vortex-Based Zero-Conflict Design of Urban Road Networks. *Networks and Spatial Economics*, 13(3), 229–254. <https://doi.org/10.1007/s11067-012-9179-x>
- Ester, M., Kriegel, H. P., Sander, J., Xu, X., 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Second International Conference on Knowledge Discovery and Data Mining*, 226–231. <https://doi.org/10.1.1.71.1980>
- Fiorenzo-Catalano, S., van Nes, R., Bovy, P.H.L., 2004. Choice Set Generation for Multi-modal Travel Analysis. *Eur. J. Transp. Infrastruct. Res.* 4, 195–209.
- Fisk, C., 1980. Some developments in equilibrium traffic assignment. *Transp. Res. Part B Methodol.* 14, 243–255. doi:10.1016/0191-2615(80)90004-1
- Flötteröd, G., Osorio, C., 2017. Stochastic network link transmission model. *Transp. Res. Part B Methodol.* 102, 180–209. doi:10.1016/j.trb.2017.04.009
- Forbes, T., H. Zagorski, E. Holshouser, Deterline W., 1958. Measurement of driver reactions to tunnel conditions. *Highway Research Board Proceedings*, Vol. 37, pp. 345-357.
- Fotheringham, A., 2000. GIS-based spatial modelling: a step forward or a step backwards. In *Spatial Models and GIS: New Potential and New Models* (pp. 21–30). Taylor and Francis.
- Frank, M., Wolfe, P., 1956. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3, 95–109. [https://doi.org/10.1016/S0377-2217\(87\)80152-2](https://doi.org/10.1016/S0377-2217(87)80152-2)
- Fränti, P., Virtajoki, O., Kaukoranta, T., 2002. Branch-and-bound technique for solving optimal clustering, in: *Object Recognition Supported by User Interaction for Service Robots*. *IEEE Comput. Soc*, pp. 232–235. doi:10.1109/ICPR.2002.1048281
- Friedrich, M., Galster, M., 2009. Methods for Generating Connectors in Transport Planning Models. *Transportation Research Record: Journal of the Transportation Research Board*, 2132(1), 133–142. <https://doi.org/10.3141/2132-15>
- Friesz, T.L., 1985. Transportation network equilibrium, design and aggregation: Key developments and research opportunities. *Transp. Res. Part A Gen.* 19, 413–427. doi:10.1016/0191-2607(85)90041-X

- Friesz, T.L., Bernstein, D., Smith, T.E., Tobin, R.L., Wie, B.W., 1993. A Variational Inequality Formulation of the Dynamic Network User Equilibrium Problem. *Oper. Res.* 41, 179–191. doi:10.1287/opre.41.1.179
- Gan, G., Ma, C., Wu, J., 2007. *Data Clustering: Theory, Algorithms, and Applications*. ASASIAM Series on Statistics and Applied Probability (Volume 20, Vol. 20). SIAM, Society for Industrial and Applied Mathematics.
- Garey, M.R., Johnson, D.S., Witsenhausen, H.S., 1982. The Complexity of the Generalized Lloyd-Max Problem. *IEEE Trans. Inf. Theory* 28, 255–256. doi:10.1109/TIT.1982.1056488
- Gazis, D.C., Herman, R., Rothery, R.W., 1961. Nonlinear Follow-The-Leader Models of Traffic Flow. *Oper. Res.* 9, 545–567
- Gehlke, C. E., and Biehl, K., 1934. Certain Effects of Grouping Upon the Size of the Correlation Coefficient in Census Tract Material. *Journal of the American Statistical Association*, Vol . 29 , No . 185 , 169–170.
- Gendreau, M., Potvin, J.-Y., 2005. Tabu search. In *Search Methodologies* (pp. 165–186). Springer.
- Gentile, G., 2010. The General Link Transmission Model for Dynamic Network Loading and a comparison with the DUE algorithm, in: *New Developments in Transport Planning: Advances in Dynamic Traffic Assignment* (Chapter 8). pp. 1615–1620.
- Gentile, G., Noekel, K., 2009. Linear user cost equilibrium: the new traffic assignment model in Visum, in: *European Transport Conference*. The Netherlands.
- Geroliminis, N., Daganzo, C. F., 2008. Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. *Transportation Research Part B: Methodological*, 42(9), 759–770. <https://doi.org/10.1016/j.trb.2008.02.002>
- Geroliminis, N., Sun, J., 2011. Properties of a well-defined macroscopic fundamental diagram for urban traffic. *Transportation Research Part B: Methodological*, 45(3), 605–617.
- Gipps, P.G., 1981. A behavioural car-following model for computer simulation. *Transp. Res. Part B Methodol.* 15, 105–111. doi:10.1016/0191-2615(81)90037-0
- Gipps, P.G., 1986. A model for the structure of lane-changing decisions. *Transp. Res. Part B Methodol.* 20, 403–414. doi:10.1016/0191-2615(86)90012-3
- Godfrey, J. W., 1969. The mechanism of a road network. *Traffic Engineering and Control*, 11(7), 323–327.
- Goldman, A. J., 1966. Realizing the distance matrix of a graph. *Journal of research of the National Bureau of Standards*, 70B(2), 29–31. <https://doi.org/10.6028/jres.070B.013>

- Graham, R.L., Knuth, D.E., Patashnik, O., 1994. Concrete Mathematics: A Foundation for Computer Science, Book. Addison-Wesley. doi:10.2307/3619021
- Greenshields, B.D., 1935. A study of traffic capacity. 14 Annu. Meet. Highw. Res. Board Proc. 448–477.
- van der Gun, J.P.T., Pel, A.J., van Arem, B., 2017. Extending the Link Transmission Model with non-triangular fundamental diagrams and capacity drops. *Transp. Res. Part B Methodol.* 98, 154–178. doi:10.1016/j.trb.2016.12.011
- Hagen-Zanker, A., Jin, Y., 2015. Adaptive Zoning for Efficient Transport Modelling in Urban Models (pp. 673–687). https://doi.org/10.1007/978-3-319-21470-2_49
- Han, K., Piccoli, B., Friesz, T.L., Yao, T., 2012. A Continuous-time Link-based Kinematic Wave Model for Dynamic Traffic Networks 23.
- Highway Capacity Manual, 2000. Chapter 17: Unsignalized Intersections.
- Himpe, W.W.E., Corthout, R., Tampere, C.M.J., 2013. An implicit solution scheme for the Link Transmission Model. *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC* 572–577. doi:10.1109/ITSC.2013.6728292
- Hoogendoorn, S.P., Bovy, P.H.L., 2001. Generic gas-kinetic traffic systems modeling with applications to vehicular traffic flow. *Transp. Res. Part B Methodol.* 35, 317–336. doi:10.1016/S0191-2615(99)00053-3
- Hopcroft, J., Tarjan, R., 1973. Algorithm 447: Efficient Algorithms for Graph Manipulation. *Commun. ACM* 16, 372–378. doi:10.1145/362248.362272
- Hruschka, E. R., Campello, R. J. G. B., Freitas, A. a., de Carvalho, A. C. P. L. F., 2009. A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 39(2), 133–155. <https://doi.org/10.1109/TSMCC.2008.2007252>
- Huang, H.J., Lam, W.H.K., 2002. Modeling and solving the dynamic user equilibrium route and departure time choice problem in network with queues. *Transp. Res. Part B Methodol.* 36, 253–273. doi:10.1016/S0191-2615(00)00049-7
- Jafari, E., Pandey, V., Boyles, S. D., 2016. Static traffic assignment : a decentralized approach. In TRB 95th annual meeting.
- Jafari, E., Gemar, M. D., Juri, N. R., Duthie, J., 2015. An Investigation of Centroid Connector Placement for Advanced Traffic Assignment Models With Added Network. *Transportation Research Record Journal of the Transportation Research Board*, (June 2015). <https://doi.org/10.3141/2498-03>
- Jain, A. K., Murty, M. N., Flynn, P. J., 1999. Data clustering: a review. *ACM Computing Surveys*, 31(3), 264–323. <https://doi.org/10.1145/331499.331504>

- Janson, B.N., 1991. Dynamic traffic assignment for urban road networks. *Transp. Res. Part B Methodol.* 25, 143–161. doi:10.1016/0191-2615(91)90020-J
- Jayakrishnan, R., Mahmassani, H.S., Hu, T.-Y., 1994. An Evaluation Tool for Advanced Traffic Information and Management Systems in Urban Networks. *Transp. Res. Part C Emerg. Technol.* 2, 129–147.
- Jeon, J.-H., Kho, S.-Y., Park, J. J., Kim, D.-K., 2012. Effects of spatial aggregation level on an urban transportation planning model. *KSCE Journal of Civil Engineering*. <https://doi.org/10.1007/s12205-012-1400-4>
- Joshi, D., Soh, L. K., Samal, A., 2012. Redistricting using constrained polygonal clustering. *IEEE Transactions on Knowledge and Data Engineering*, 24(11), 2065–2079. <https://doi.org/10.1109/TKDE.2011.140>
- Kearns, M., Mansour, Y., Ng, A. Y., 1997. An Information-Theoretic Analysis of Hard and Soft Assignment Methods for Clustering. *Proc. of Conference on Uncertainty in Artificial Intelligence*, 282–293.
- Khatib, Z., Chang, K., Ou, Y., 2001. Impacts of Analysis Zone Structures on Modeled Statewide Traffic. *Journal of Transportation Engineering*, 127(1), 31–38. [https://doi.org/10.1061/\(ASCE\)0733-947X\(2001\)127:1\(31\)](https://doi.org/10.1061/(ASCE)0733-947X(2001)127:1(31))
- Kim, K., Dean, D. J., Kim, H., Chun, Y., 2016. Spatial optimization for regionalization problems with spatial interaction: a heuristic approach. *International Journal of Geographical Information Science*, 30(3), 451–473. <https://doi.org/10.1080/13658816.2015.1031671>
- Klein, D., Kamvar, S. D., Manning, C. D. (2002). From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. *Int. Conf. Mach. Learn.*, 307–314. <https://doi.org/citeulike-article-id:112870>
- Knoop, V. L., Hoogendoorn, S. P., 2013. Network Transmission Model: a dynamic traffic model at network level. In *TRB 93rd Annual Meeting Compendium of Papers* (pp. 1–16).
- Knoop, V. L., van Lint, H., Hoogendoorn, S. P., 2015. Traffic dynamics: Its impact on the Macroscopic Fundamental Diagram. *Physica A: Statistical Mechanics and Its Applications*, 438, 236–250. <https://doi.org/10.1016/j.physa.2015.06.016>
- Knoop, V. L., Tamminga, G. F., Leclercq, L., 2016. Network Transmission Model: Application to a Real World City. *TRB 95th Annual Meeting Compendium of Papers*.
- Koontz, W. L. G., Narendra, P. M., Fukunaga, K., 1975. A Branch and Bound Clustering Algorithm. *IEEE Transactions on Computers*, C-24(9). <https://doi.org/10.1109/T-C.1975.224336>
- Koppelman, F.S., Wen, C.H., 2000. The paired combinatorial logit model: Properties, estimation and application. *Transp. Res. Part B Methodol.* 34, 75–89. doi:10.1016/S0191-2615(99)00012-0

- Kwigizile, V., Teng, H., 2009. Comparison of methods for defining geographical connectivity for variables of trip generation models. *Journal of Transportation Engineering*, 135(7), 454–466. [https://doi.org/10.1061/\(ASCE\)0733-947X\(2009\)135:7\(454\)](https://doi.org/10.1061/(ASCE)0733-947X(2009)135:7(454))
- Larsson, T., Patriksson, M., 1995. An augmented lagrangean dual algorithm for link capacity side constrained traffic assignment problems. *Transp. Res. Part B Methodol.* doi:10.1016/0191-2615(95)00016-7
- Laval, J.A., Leclercq, L., 2013. The Hamilton-Jacobi partial differential equation and the three representations of traffic flow. *Transp. Res. Part B Methodol.* 52, 17–30. doi:10.1016/j.trb.2013.02.008
- Lelis, L., Sander, J. (2009). Semi-supervised Density-Based Clustering. 2009 Ninth IEEE International Conference on Data Mining, 842–847. <https://doi.org/10.1109/ICDM.2009.143>
- Lenz, H., Wagner, C.K., Sollacher, R., 1999. Multi-anticipative car-following model 335, 331–335.
- Leurent, F., Benezech, V. and Samadzad, M., 2011. A Stochastic Model of Trip End Disaggregation in Traffic Assignment to a Transportation Network. *Procedia – Social and Behavioral Sciences*, Elsevier
- Li, W., Church, R. L., Goodchild, M. F., 2014. An extendable heuristic framework to solve the p-compact-regions problem for urban economic modeling. *Computers, Environment and Urban Systems*, 43, 1–13. <https://doi.org/10.1016/j.compenvurbsys.2013.10.002>
- Di, X., Liu, H.X., 2016. Boundedly rational route choice behavior: A review of models and methodologies. *Transp. Res. Part B Methodol.* 85, 142–179. doi:10.1016/j.trb.2016.01.002
- Lighthill, M.J., Whitham, G.B., 1955. On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads. *Proc. R. Soc. A Math. Phys. Eng. Sci.* 229, 317–345. doi:10.1098/rspa.1955.0089
- Long, G. D., Stover, V. G., 1967. The effect of network detail on traffic assignment results. College Station, Texas.
- MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations. In *Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 1, pp. 281–297). <https://doi.org/citeulike-article-id:6083430>
- Mahut, M., 2001. A Discrete Flow Model For Dynamic Network Loading.
- Mann, W. W., 2002. B-node model: new subarea traffic assignment model & application. in *transportation research record* (pp. 273–281).
- Manning, C. D., Schutze, H., 1999. *Foundations of statistical natural language processing*. Cambridge Mass.: MIT press.

- Martínez, L.M., Viegas, J.M., Silva, E. a., 2009. A traffic analysis zone definition: A new methodology and algorithm. *Transportation (Amst)*. 36, 581–599. doi:10.1007/s11116-009-9214-z
- Mathew, T., 2014. Signalized Intersection Delay Models. *Transportation systems Engineering. Syst. Eng.* Chapter 35.
- McFadden, D., 1974. Conditional logit analysis of qualitative choice behavior, in: *Frontiers in Econometrics*. pp. 105–142. doi:10.1108/eb028592
- Messmer, A., Papageorgiou, M., 1990. METANET: a macroscopic simulation program for motorway networks. *Traffic Eng. Control* 31, 466–470.
- Moridpour, S., Sarvi, M., Rose, G., 2010. Lane changing models: a critical review. *Transp. Lett.* 2, 157–173. doi:10.3328/TL.2010.02.03.157-173
- Morrison, D. R., Jacobson, S. H., Sauppe, J. J., Sewell, E. C., 2016. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization*, 19, 79–102. <https://doi.org/10.1016/j.disopt.2016.01.005>
- Nakayama, S., Takayama, J., Nakai, J., Nagao, K., 2012. Semi-dynamic traffic assignment model with mode and route choices under stochastic travel times. *J. Adv. Transp.* 46, 269–281. doi:10.1002/atr.208
- Newell, G. F., 1980. *Traffic flow on transportation networks*. Cambridge Mass.: MIT Press.
- Newell, G. F., 1993. A simplified theory of kinematic waves in highway traffic, part II: Queueing at freeway bottlenecks. *Transportation Research Part B: Methodological*, 27(4), 289–303. [https://doi.org/10.1016/0191-2615\(93\)90039-D](https://doi.org/10.1016/0191-2615(93)90039-D).
- Newell, G.F., 2002. A simplified car-following theory: A lower order model. *Transp. Res. Part B Methodol.* 36, 195–205. doi:10.1016/S0191-2615(00)00044-8
- O'Neill, W., 1991. Developing Optimal Transportation Analysis Zones Using GIS. *ITE Journal*, (December), 33–36.
- Ossen, S., Hoogendoorn, S.P., 2007. Driver Heterogeneity in Car-Following and Its Impact on Modeling Traffic Dynamics. *Transp. Res. Board Annu. Meet.* 95–103.
- Openshaw, S., 1977. Optimal zoning systems for spatial interaction models. *Environment and Planning A*, 9(2), 169–184. <https://doi.org/10.1068/a090169>
- Openshaw, S., Rao, L., 1995. Algorithms for reengineering 1991 Census geography. *Environment and Planning A*, 27(April 1994), 425–446. <https://doi.org/10.1068/a270425>
- Openshaw, S. and Taylor, P. J., 1979. A million or so correlation coefficients: three experiments on the modifiable areal unit problem. *Statistical Applications in Spatial Sciences*, 44–127.

- Ortuzar, J., Willumsen, L. G., 2002. *Modelling Transport*. John Wiley & Sons Inc.
- Páez, A., Scott, M., 2004. Spatial statistics for urban analysis: A review of techniques with examples. *GeoJournal*, 61, 53–67. <https://doi.org/10.1007/sGEJO-004-0877-x>
- Pang, J.S., Han, L., Ramadurai, G., Ukkusuri, S., 2012. A continuous-time linear complementarity system for dynamic user equilibria in single bottleneck traffic flows. *Math. Program.* 133, 437–460. doi:10.1007/s10107-010-0433-z
- Payne, H.J., Thompson, W. a., Isaksen, L., 1973. Design of a Traffic-Responsive Control System for a Los Angeles Freeway. *IEEE Trans. Syst. Man. Cybern.* 3, 213–224. doi:10.1109/TSMC.1973.4309209
- Payne, H.J., Thompson, W.A., 1975. Traffic assignment on transportation networks with capacity constraints and queueing. In: Paper Presented at the 47th National ORSA Meeting/TIMS 1975 North-American Meeting, Chicago, IL
- Pfitzner, D., Leibbrandt, R., Powers, D., 2009. Characterization and evaluation of similarity measures for pairs of clusterings. *Knowledge and Information Systems*, 19(3), 361–394. <https://doi.org/10.1007/s10115-008-0150-6>
- Pipes, L.A., 1953. An Operational Analysis of Traffic Dynamics. *J. Appl. Phys.* 24, 274–281. doi:10.1063/1.1721265
- Pipes, L.A., 1967. Car following models and the fundamental diagram of road traffic. *Transp. Res.* 1, 21–29. doi:10.1016/0041-1647(67)90092-5
- Polyak, B. K., 1990. New method of stochastic approximation type. *Automated Remote Control*, 51(7), 937–946.
- Puchinger, J., Raidl, G., 2005. Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In *Artificial intelligence and knowledge engineering applications: A bioinspired approach* (pp. 41–54). Las Palmas.
- Qian, Z., Zhang, H. M., 2012. On centroid connectors in static traffic assignment: Their effects on flow patterns and how to optimize their selections. *Transportation Research Part B: Methodological*, 46(10), 1489–1503. <https://doi.org/10.1016/j.trb.2012.07.006>
- Raadsen, M.P.H., Bliemer, M.C.J., Bell, M.G.H., 2016. An efficient and exact event-based algorithm for solving simplified first order dynamic network loading problems in continuous time. *Transp. Res. Part B Methodol.* 92, 191–210. doi:10.1016/j.trb.2015.08.004
- Raadsen, M.P.H., Schilpzand, M.P., Mein, H.E., 2009. Applying inter regional shared routes in detailed multiregional dynamic traffic models, in: *European Transport Conference*.
- Ramezani, M., Nourinejad, M., 2017. Dynamic modeling and control of taxi services in large-scale urban networks: A macroscopic approach. *Transportation Research Procedia*, 23, 41–60. <https://doi.org/10.1016/j.trpro.2017.05.004>

- Ramezani, M., Nourinejad, M., 2017. Dynamic modeling and control of taxi services in large-scale urban networks: A macroscopic approach. *Transportation Research Procedia*, 23, 41–60. <https://doi.org/10.1016/j.trpro.2017.05.004>
- Rogers, D.F., Plante, R.D., Wong, R.T., Evans, J.R., 1991. Aggregation and disaggregation techniques and methodology in optimization. *Oper. Res.* 39, 553–582.
- Rahman, M., Chowdhury, M., Xie, Y., He, Y., 2013. Review of microscopic lane-changing models and future research opportunities. *IEEE Trans. Intell. Transp. Syst.* 14, 1942–1956. doi:10.1109/TITS.2013.2272074
- Rakha, H., Crowther, B., 2002. Comparison of Greenshields, Pipes, and Van Aerde car-following and traffic stream models. *Transp. Res. Rec.* 248–262.
- Richards, P.I., 1956. Shock Waves on the Highway. *Oper. Res.* 4, 42–51. doi:10.1287/opre.4.1.42
- Ruddell, K., Raith, A., 2013. Initializing the Traffic Assignment Problem by Zone Aggregation and Disaggregation. In *TRB 93rd Annual Meeting Compendium of Papers*.
- Ruiz, C., Spiliopoulou, M., Menasalvas, E., 2009. Density-based semi-supervised clustering. *Data Mining and Knowledge Discovery*, 21(3), 345–370. <https://doi.org/10.1007/s10618-009-0157-y>
- Schockaert, S., Smart, P. D., Twaroch, F. A., 2011. Generating approximate region boundaries from heterogeneous spatial information: An evolutionary approach. *Information Sciences*, 181(2), 257–283. <https://doi.org/10.1016/j.ins.2010.09.021>
- Shahpar, A.H., Aashtiani, H.Z., Babazadeh, A., 2008. Dynamic penalty function method for the side constrained traffic assignment problem. *Appl. Math. Comput.* 206, 332–345. doi:10.1016/j.amc.2008.09.014
- Sheffi, Y., 1984. Aggregation and equilibrium with multinomial logit models. *Appl. Math. Model.* 8, 121–127. doi:10.1016/0307-904X(84)90064-7
- Smith, M.J., 1979. The existence, uniqueness and stability of traffic equilibria. *Transp. Res. Part B Methodol.* 13, 295–304. doi:10.1016/0191-2615(79)90022-5
- Smith, M.J., 1987. Traffic control and traffic assignment in a signal-controlled network with queueing, in: Gartner, N.H., Wilson M. (Eds.), *Proceedings of the Tenth International Symposium on Transportation and Traffic Theory*. Elsevier, pp. 61–68.
- Smith, M.J., 1993. A new dynamic traffic model and the existence and calculation of dynamic user equilibria on congested capacity-constrained road networks. *Transp. Res. Part B Methodol.* 27B, 49–63.
- Smith, M.J., 2013. A link-based elastic demand equilibrium model with capacity constraints and queueing delays. *Transp. Res. Part C Emerg. Technol.* 29, 131–147. doi:10.1016/j.trc.2012.04.011

- Smith, M., Huang, W., Viti, F., 2013. Equilibrium in Capacitated Network Models with Queueing Delays, Queue-storage, Blocking Back and Control. *Procedia - Soc. Behav. Sci.* 80, 860–879. doi:10.1016/j.sbspro.2013.05.047
- Smits, E.-S., Bliemer, M.C.J., Pel, A.J., van Arem, B., 2015. A family of macroscopic node models. *Transp. Res. Part B Methodol.* 74, 20–39. doi:10.1016/j.trb.2015.01.002
- Steijn, J. Van., 2016. Aggregation in transport networks for a flexible assignment model. Master thesis Civil Engineering, Twente University.
- Strippgen, D., Nagel, K., 2009. Multi-agent traffic simulation with CUDA. *Proc. 2009 Int. Conf. High Perform. Comput. Simulation, HPCS 2009* 106–114. doi:10.1109/HPCSIM.2009.5192895
- Szeto, W.Y., Lo, H.K., 2005. Properties of dynamic traffic assignment with physical queues. *J. East. Asia Soc. Transp. Stud.* 6, 2108–2123.
- Taillard, É. D., 2003. Heuristic methods for large centroid clustering problems. *Journal of Heuristics*, 9(1), 51–73. <https://doi.org/10.1023/A:1021841728075>
- Tampère, C.M.J., Corthout, R., Cattrysse, D., Immers, L.H., 2011. A generic class of first order node models for dynamic macroscopic simulation of traffic flows. *Transp. Res. Part B Methodol.* 45, 289–309. doi:10.1016/j.trb.2010.06.004
- Train, K., 2003. *Discrete Choice Methods with Simulation*, Cambridge University Press. doi:10.1017/CBO9780511753930
- Treiber, M., Hennecke, A., Helbing, D., 2000. Congested Traffic States in Empirical Observations and Microscopic Simulations 62, 1805–1824. doi:10.1103/PhysRevE.62.1805
- Treiber, M., Kesting, A., Helbing, D., 2006. Delays, inaccuracies and anticipation in microscopic traffic models. *Phys. A Stat. Mech. its Appl.* 360, 71–88. doi:10.1016/j.physa.2005.05.001
- Verfaillie, G., Lemaître, M., Schiex, T., 1996. Russian Doll Search for Solving Constraint Optimization Problems. *Proc. Thirteen. Natl. Conf. Artif. Intell.* 181–187.
- Vickrey, W.S., 1969. Congestion Theory and Transport Investment. *Am. Econ. Rev.* 59, 251–260.
- Voß, S., 2000. Meta-heuristics: The State of the Art. In *Local search for planning and scheduling*, ECAI 2000 workshop. Berlin: Springer. <https://doi.org/10.1007/978-3-642-15314-3>
- van Wageningen-Kessels, F., van Lint, H., Vuik, K., Hoogendoorn, S., 2014. Genealogy of traffic flow models. *EURO J. Transp. Logist.* 4, 445–473. doi:10.1007/s13676-014-0045-5

- Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S., 2001. Constrained K-means Clustering with Background Knowledge. *International Conference on Machine Learning*, 577–584. <https://doi.org/10.1109/TPAMI.2002.1017616>
- Ward, J. H., 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*. <https://doi.org/10.1080/01621459.1963.10500845>
- Wardrop, J.G., 1952. Some theoretical aspects of road traffic research. *ICE Proc. Eng. Div.* 1, 325–362. doi:10.1680/ipeds.1952.11259
- Watling, D.P., Rasmussen, T.K., Prato, C.G., Nielsen, O.A., 2015. Stochastic user equilibrium with equilibrated choice sets: Part I - Model formulations under alternative distributions and restrictions. *Transp. Res. Part B Methodol.* 77, 166–181. doi:10.1016/j.trb.2015.03.008.
- Webster, F.V., 1958. Traffic signal settings. *Road Res. Lab. Tech. Pap.* No 39.
- Weeks, J. R., 2004. The Role of Spatial Analysis in Demographic Research. In M. Goodchild D. Janelle (Eds.), *Spatially Integrated Social Science: Examples in Best Practice* (pp. 381–399). New York: Oxford University Press.
- Wei, B. C., Chai, W. Y., 2004. A multiobjective hybrid metaheuristic approach for GIS-based spatial zoning model. *Journal of Mathematical Modelling and Algorithms*, 3(3), 245–261. <https://doi.org/10.1023/B:JMMA.0000038615.32559.af>
- Weisstein, Eric W. "Circle-Circle Intersection." From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/Circle-CircleIntersection.html>
- Wiedemann R., 1974. Simulation des Strassenverkehrsflusses. Institute for Traffic Engineering, University of Karlsruhe, Tech. Rep.
- Wong, D., 2001. Location-specific cumulative distribution function (LSCDF): An alternative to spatial correlation analysis. *Geographical Analysis*, 33(1), 76–93. <https://doi.org/10.1111/j.1538-4632.2001.tb00438.x>
- Yang, H., Yagar, S., 1994. Traffic assignment and traffic control in general freeway-arterial corridor systems. *Transp. Res. Part B Methodol.* 28, 463–486. doi:10.1016/0191-2615(94)90015-9
- Yperman, I., 2007. The Link Transmission Model for Dynamic Network Loading. Katholieke Universiteit Leuven.
- Zhang, Z., Wolshon, B., Dixit, V. V., 2015. Integration of a cell transmission model and macroscopic fundamental diagram: Network aggregation for dynamic traffic models. *Transportation Research Part C: Emerging Technologies*, 55, 298–309. <https://doi.org/10.1016/j.trc.2015.03.040>
- Zheng, L., He, Z., He, T., 2017. A flexible traffic stream model and its three representations of traffic flow. *Transp. Res. Part C Emerg. Technol.* 75, 136–167. doi:10.1016/j.trc.2016.12.006

- Zhou, X., Erdogan, S., Mahmassani, H. S., 2006. Dynamic Origin—Destination Trip Demand Estimation for Subarea Analysis. *Transportation Research Record: Journal of the Transportation Research Board*, 1964, 176–184.
- Zhou, X., Taylor, J., 2014. DTALite: A queue-based mesoscopic traffic simulator for fast model evaluation and calibration. *Cogent Eng.* 1, 1–19.
doi:10.1080/23311916.2014.961345
- Zipkin, P., 1980. Bounds for aggregating nodes in network problems. *Mathematical Programming*, 19, 155–177.

Appendix A

Both scaling factors introduced in Chapter 9 are formulated on the zone component level. However in the case studies they are utilised on a cluster level. This requires an alternative formulation, although the underlying principle remains identical. In this appendix, the cluster based formulation is provided.

A.1 Cluster based service area scaling factor

The original service area scaling factors are made cluster aware, denoted $\hat{\mathcal{X}}^I \in [0,1]^{\hat{Z} \times N}$, via

$$\hat{x}_{\hat{z}n}^I = \begin{cases} \min \left\{ 1, \Theta^{-1} \left(\frac{\pi}{\sum_{n=1}^N (\hat{N}_{\hat{z}n}^+ \parallel \hat{N}_{\hat{z}n}^-)} \right) \right\}, & \text{if } \hat{N}_{\hat{z}n}^+ \parallel \hat{N}_{\hat{z}n}^- = 1, \\ 0, & \text{otherwise,} \end{cases} \quad (\text{A.1})$$

with $\hat{z} \in \{1, \dots, \hat{Z}\}, n \in \{1, \dots, N\}$. The only difference with Equation (9.27) being that the number of connectoids is conditional on the cluster rather than the zone component.

A.2 Cluster based centrality scaling factor

Like the service scaling factor, the centrality scaling factors also require a cluster based counterpart. First, the coordinates for the cluster extreme points $\hat{\mathbf{x}}^{\max}, \hat{\mathbf{y}}^{\max}, \hat{\mathbf{x}}^{\min}, \hat{\mathbf{y}}^{\min} \in \mathbb{R}_+^{\hat{Z} \times 1}$ are defined:

$$\begin{aligned} \hat{x}_{\hat{z}}^{\max} &= \max \{ \vec{x}_{\vec{z}}^{\max} \mid \text{rref}(\mathbf{G})_{\vec{z}\vec{z}} = 1 \}, \\ \hat{y}_{\hat{z}}^{\max} &= \max \{ \vec{y}_{\vec{z}}^{\max} \mid \text{rref}(\mathbf{G})_{\vec{z}\vec{z}} = 1 \}, \\ \hat{x}_{\hat{z}}^{\min} &= \max \{ \vec{x}_{\vec{z}}^{\min} \mid \text{rref}(\mathbf{G})_{\vec{z}\vec{z}} = 1 \}, \\ \hat{y}_{\hat{z}}^{\min} &= \max \{ \vec{y}_{\vec{z}}^{\min} \mid \text{rref}(\mathbf{G})_{\vec{z}\vec{z}} = 1 \}, \end{aligned} \quad (\text{A.2})$$

with $\hat{z} \in \{1, \dots, \hat{Z}\}, \vec{z} \in \{1, \dots, \vec{Z}\}$. Each cluster \hat{z} its centre points is obtained in a similar fashion as Equation (9.29) and (9.30), only now conditional on cluster extremities:

$$\hat{x}_{\hat{z}} = \frac{1}{2} (\hat{x}_{\hat{z}}^{\max} + \hat{x}_{\hat{z}}^{\min}), \quad \hat{z} \in \{1, \dots, \hat{Z}\}, \quad (\text{A.3})$$

$$\hat{y}_{\hat{z}} = \frac{1}{2} (\hat{y}_{\hat{z}}^{\max} + \hat{y}_{\hat{z}}^{\min}), \quad \hat{z} \in \{1, \dots, \hat{Z}\}, \quad (\text{A.4})$$

which then leads to the portion $\hat{\mathcal{X}}^\Lambda \in [0,1]^{\hat{Z} \times N}$ of the centrality factor χ^{\min} to be applied to the connectoid, analogous to Equation (9.31), but conditional on the cluster \hat{z} . This is achieved via

$$\hat{\chi}_{\hat{z}n}^{\Delta} = 1 - \frac{|x_n - \hat{x}_{\hat{z}}| + |y_n - \hat{y}_{\hat{z}}|}{(\hat{x}_{\hat{z}}^{\max} - \bar{x}_{\hat{z}}) + (\hat{y}_{\hat{z}}^{\max} - \hat{y}_{\hat{z}})}, \quad (\text{A.5})$$

with $\hat{z} \in \{1, \dots, \hat{Z}\}, n \in \{1, \dots, N\}$. The cluster based centrality scaling factors $\hat{\chi}^H \in [\chi^{\min}, 1]^{\hat{Z} \times N}$ are then found, based on the same concept originally defined in Equation 9.28, via

$$\hat{\chi}_{\hat{z}n}^H = \begin{cases} \chi^{\min} + \hat{\chi}_{\hat{z}n}^{\Delta}(1 - \chi^{\min}), & \text{if } \hat{N}_{\hat{z}n}^+ \parallel \bar{N}_{\hat{z}n}^- = 1, \\ 0, & \text{otherwise,} \end{cases} \quad (\text{A.6})$$

with $\hat{z} \in \{1, \dots, \hat{Z}\}, n \in \{1, \dots, N\}$.