

# **Geospatial Inference and Management of Utility Infrastructure Networks**

Qingyuan Ji

Thesis submitted for the Degree of  
Doctor of Philosophy



School of Engineering  
Newcastle University

January 2020



## **Abstract**

Modern cities consist of spatially and temporally complex networks that connect urban infrastructure assets to the buildings they service. Critical infrastructure networks include transport, electricity, water supply, waste water and gas, all of which play a key role in the functioning of modern cities. Understanding network spatial connectivity, resource flow, dependencies and interdependencies is essential for infrastructure planning, management, and assessment of system robustness and resilience. However, there is a sparsity of fine spatial scale data from which such understanding can be derived or inferred. Often data is held within commercially sensitive organisations and may be incomplete topologically and/or spatially. Thus, there is an urgent need to develop new approaches to the integrated inference, management and analysis of the complex utility infrastructure networks. Such approaches should allow the highly granular representation of utility network connectivity to be represented in a spatially explicit manner, employing methods of data and information management to ensure they are scalable and generic.

This thesis presents the development of such an approach, one that employs a geospatial ontology to formally define the key entities, attributes and relationships of fine spatial scale utility infrastructure networks. This ontology is used as the conceptual framework for the development of a suite of algorithms that allow the heuristic inference of the spatial layout of utility infrastructure networks for any urban conurbation within the UK. This is demonstrated via several case studies where the electricity feeder network between substations and buildings is generated for several different cities within the UK. Validation against the known network for the city of Newcastle upon Tyne indicates that the network can be inferred to high levels of accuracy (about 90%). Moreover, the algorithm is shown to be transferable to the inference and integration of other utility infrastructure networks (gas, water supply, waste water, and new road layouts).

The representation, management and analysis of such spatially complex and large utility networks is, however, a major challenge. The efficient storage, management and analysis of such spatial networks is explored via a comparison of a traditional RDMS approach (PgRouting within Postgres), spatial database (PostGIS) and a NoSQL graph-database (Neo4j), as well as a bespoke hybrid spatial-graph framework (combination of PostGIS and Neo4j). A suite of comparison tests of data writing, data reading and complex network analysis demonstrated that significant performance benefits in the use of the NoSQL graph database approach for data read (around 210% faster) and network analysis (between 420 and 1170 % faster). However, this was at the expenses of data writing which was found to be between 135 and 150% slower.

## Declaration

I declare that the thesis has been composed by myself and that the work has not been submitted for any other degree or professional qualification. I confirm that the work submitted is my own. I confirm that appropriate credit has been given within this thesis where reference has been made to the work of others.

Part of the work presented in Chapter 4 (spatial heuristic algorithm) and Chapter 7 (database performance benchmarking tests) was previously published in the following papers:

**Ji, Q.**, Barr, S., James, P., and Fairbairn, D., 2017. A heuristic spatial algorithm for generating fine-scale infrastructure distribution networks. In: 25<sup>th</sup> GISRUK 2017, Manchester, UK.

**Ji, Q.**, Barr, S., James, P., and Fairbairn, D., 2018. Graph database implementation of fine spatial scale urban infrastructure networks. In: 26<sup>th</sup> GISRUK 2018, Leicester, UK.

**Ji, Q.**, Barr, S., James, P., and Fairbairn, D., 2018. A geospatial analysis framework for fine scale urban infrastructure networks. In: ISPRS Technical Commission IV Symposium 2018, Delft, the Netherlands.

In all of these publications, the dissertation author is the primary researcher and conducted all the experiments and analyses, whilst the co-authors listed are the main author's supervisors, who directed research, and fine-tuned the English.

Qingyuan Ji

June 2019, Newcastle upon Tyne



## **Acknowledgement**

I would like to express my gratitude to everyone that has helped and supported me while I was undertaking this research. First and foremost, thanks to my supervisors Stuart Barr, Phillip James, and David Fairbairn, for their years of support and guidance. This research would not have been possible without them. Thanks also to my Geomatics colleagues that have provided useful support or suggestion to my work.

I have received a great deal of technical assistances from Craig Robson, who is always patient to get me familiar with necessary software stacks used in the research group. I would like to thank Northern Power Grid, Northumbria Water Group, and Northern Gas Networks, who are generous to provide me necessary infrastructure network data. In particular, I would like to thank Daniel Evans and Peter Thomson from NGN, for their knowledge and patience that let me understands how gas network works. I would like to thank Jialiang Yi from CESI, who also provided great help for me to access and understand electricity distribution network data.

My family have been extremely kind and understanding throughout the course of this research. Thank you for always being there.

This research was partially funded by the MISTRAL project and the School of Engineering at Newcastle University.





# Table of Contents

<b>Abstract .....</b>	<b>i</b>
<b>Declaration .....</b>	<b>iii</b>
<b>Acknowledgement.....</b>	<b>v</b>
<b>Table of Contents .....</b>	<b>vii</b>
<b>List of Figures .....</b>	<b>xii</b>
<b>List of Tables .....</b>	<b>xxi</b>
<b>List of Listings.....</b>	<b>xxiv</b>
<b>Glossary .....</b>	<b>xxv</b>
<b>Chapter 1. Introduction .....</b>	<b>1</b>
1.1 Urban Infrastructure Challenges.....	1
1.2 Aims and Objectives .....	4
1.3 Thesis Structure .....	5
<b>Chapter 2. Literature Review .....</b>	<b>6</b>
2.1 The Changing Cities .....	6
2.2 Critical Urban Infrastructures .....	10
2.3 Approaches of Modelling Urban Infrastructures .....	11
2.4 Research Focus on Urban Infrastructure Networks.....	16
2.4.1 Resilience of Individual Infrastructure Sectors .....	17
2.4.2 Dependencies and Interdependencies.....	19
2.5 Geospatial Urban Infrastructure Models .....	21
2.5.1 Geospatial Infrastructure Ontology Development.....	22
2.5.2 Geospatial Infrastructure Data Inference.....	26
2.5.3 Database System Implementation .....	28
2.6 Summary.....	29
<b>Chapter 3. An Ontology for Modelling Urban Infrastructure Networks.....</b>	<b>31</b>
3.1 Introduction .....	31
3.2 Ontology Construction .....	33

3.3 Building.....	37
3.4 Utility Network .....	39
3.5 Transport Networks.....	44
3.6 Dependencies .....	49
3.7 Formal Representation of Ontology.....	51
3.8 Conclusion .....	56
<b>Chapter 4. A heuristic spatial algorithm for generating fine-scale infrastructure</b>	
<b>distribution networks.....</b>	<b>57</b>
4.1 Introduction.....	57
4.2 Algorithm Overview .....	60
4.3 Algorithm Description.....	62
4.3.1 Topology Generation.....	63
4.3.2 Geometry Generation.....	67
4.4 Algorithm Implementation.....	75
4.5 Pilot Study (Newcastle upon Tyne).....	76
4.6 Synthetic Network Validation .....	81
4.6.1 Validations on Feeders .....	83
4.6.2 Validation on Service Lines .....	85
4.7 Algorithm Transferability Test .....	89
4.8 Conclusion .....	97
<b>Chapter 5. Utility Network Integration .....</b>	<b>99</b>
5.1 Introduction.....	99
5.2 Gas Network Integration.....	100
5.2.1 Gas Network Data.....	100
5.2.2 NGN Network Completion .....	103
5.2.3 Gas Distribution Network Generation .....	110
5.2.4 Parameter Sensitivity .....	112
5.2.5 Gas Network Validation .....	119
5.3 Water Supply Network Integration .....	121
5.3.1 Water Supply Network Data .....	122

5.3.2 Water Flow Infer .....	124
5.3.3 Water Distribution Network Generation.....	129
5.4 Sewer Network Integration.....	131
5.4.1 Sewer Network Data.....	132
5.4.2 Fine Scale Sewer Network Generation.....	134
5.4.3 Sewer Network Flow Infer .....	137
5.5 Utility Network Dependency Integration .....	143
5.6 Conclusion.....	149
<b>Chapter 6 – Road Network Generation Algorithm .....</b>	<b>151</b>
6.1 Introduction .....	151
6.2 Automatic Network Generation.....	152
6.3 Data Sets .....	153
6.4 Road Network Generation Algorithm.....	155
6.4.1 Building cluster generation using MST.....	156
6.4.2 Road geometry generation.....	159
6.4.3 Road geometry optimization .....	163
6.4.4 Road Network Validation .....	164
6.5 Electricity Distribution Network Generation .....	168
6.5.1 Synthetic Electricity Network generation.....	168
6.6 Parameter Sensitivity Test .....	174
6.7 Transferability Test .....	176
6.7.1 Data Sets .....	176
6.7.2 Results and Validation .....	179
6.8 Conclusion.....	192
<b>Chapter 7 – Database Performance Benchmarking Tests .....</b>	<b>194</b>
7.1 Introduction .....	194
7.2 Database Approaches for Tests.....	196
7.2.1 ITRC Interdependency Network Schema.....	197
7.2.2 PgRouting .....	199
7.2.3 Hybrid Database .....	202

7.3 Performance Benchmarking Tests Overview .....	206
7.4 Performance Test on Different Sized Network Data.....	207
7.4.1 Network Data .....	208
7.4.2 Writing Test.....	211
7.4.3 Reading Test.....	212
7.4.4 Shortest path test .....	214
7.5 Performance Test on City Scale Network Data from Newcastle .....	217
7.5.1 Test Data.....	218
7.5.2 Performance Test on querying Newcastle Integrated Road Network .....	222
7.5.3 Performance Test on querying Newcastle Electricity Distribution Network .....	227
7.6 Performance Test on Mega City Scale Network Data from London .....	230
7.6.1 Test Data.....	231
7.6.2 Writing, Reading, and Shortest Path Test.....	232
7.6.3 Complex Query Test .....	233
7.7 Conclusion .....	238
<b>Chapter 8. Discussion .....</b>	<b>242</b>
8.1 Introduction.....	242
8.2 Geospatial Infrastructure Network Ontology.....	242
8.3 Inference of Spatial Infrastructure Network .....	247
8.3.1 Generic Spatial Heuristic Algorithm.....	247
8.3.2 Algorithm Transferability in Different Utility Networks .....	252
8.3.3 Road Network Generation Algorithm .....	253
8.4 Database Approach for Management of Spatial Network Data.....	255
8.5 Application of infrastructure data inference and management .....	257
8.6 Summary .....	260
<b>Chapter 9. Conclusion .....</b>	<b>262</b>
9.1 Introduction.....	262
9.2 Research Summary .....	262
9.3 Future Work .....	265
9.3.1 Critical Infrastructure Decision Support .....	265

9.3.2 Understanding Dynamics of Infrastructure Networks.....	267
9.3.3 Big Data Processing Capability.....	268
9.4 Key Findings and Implications.....	269
<b>Appendix A – Basic Software Stacks used in the Thesis .....</b>	<b>271</b>
<b>Appendix B – Installation of the ITRC schema .....</b>	<b>272</b>
<b>Appendix C – Spatial heuristic algorithm.....</b>	<b>273</b>
<b>Appendix D – Gas Network Integration.....</b>	<b>281</b>
<b>Appendix E – Water Supply Network Integration .....</b>	<b>284</b>
<b>Appendix F – Sewer Network Integration.....</b>	<b>286</b>
<b>Appendix G – Road Network Generation Algorithm.....</b>	<b>288</b>
<b>Appendix H – Database Performance Benchmarking Test Data .....</b>	<b>290</b>
<b>Appendix I – Database Performance Benchmarking Test Result.....</b>	<b>296</b>
<b>Appendix J – Scripts for database performance benchmarking tests.....</b>	<b>299</b>
<b>References.....</b>	<b>303</b>
<b>Bibliography.....</b>	<b>326</b>

## List of Figures

<b>Figure 2.1.</b> Example of converting of vector road data to raster cells (Delameter, et al., 2012). .....	12
<b>Figure 2.2.</b> Example of converting an urban street network to an axial map.....	13
<b>Figure 3.1.</b> Top-level entities and relationships in the ontology. ....	33
<b>Figure 3.2.</b> Use linear reference to represent spatially transient speed limit on a road... 36	
<b>Figure 3.3.</b> Use time reference to represent temporal transient attribute. ....	37
<b>Figure 3.4.</b> Entities and relationships for Building. ....	38
<b>Figure 3.5.</b> Entities of Utility Network. ....	40
<b>Figure 3.6.</b> Example of an Electricity Network and Buildings. ....	41
<b>Figure 3.7.</b> Ontologies specific for each type of Utility Network.....	42
<b>Figure 3.8.</b> Attributes related to Utility Network. ....	43
<b>Figure 3.9.</b> Entities and relationships for Transport Network. ....	45
<b>Figure 3.10.</b> An example of transfer between Road Transfer and Rail Station.....	46
<b>Figure 3.11.</b> Use sequence of Network Node to represent Number of Lanes. ....	47
<b>Figure 3.12.</b> The connection between Building and Road. ....	49
<b>Figure 3.13.</b> Relationships to represent dependency. ....	50
<b>Figure 3.14.</b> Visual representation of basic notations (sets).....	52
<b>Figure 3.15.</b> Visual representation of different infrastructure networks. ....	52
<b>Figure 3.16.</b> Visual representation on the constitution of an edge $e$ and a node $v$ , in the electricity network $G_e$ . ....	53
<b>Figure 4.1.</b> Flow of spatial heuristic algorithm. ....	61
<b>Figure 4.2.</b> Example of algorithm input data (Contains OS data © 2018).....	61
<b>Figure 4.3.</b> Example of algorithm output result (Contains OS data © 2018). ....	62
<b>Figure 4.4.</b> 77 clusters generated based on the buildings from figure 4.2 (Contains OS data © 2018).....	64

<b>Figure 4.5.</b> Base network generated by connecting clusters and assets (Contains OS data © 2018).....	65
<b>Figure 4.6.</b> Delaunay triangulation and assigning an asset to each cluster (Contains OS data © 2018). .....	66
<b>Figure 4.7.</b> Result of topology generation process (Contains OS data © 2018).....	67
<b>Figure 4.8.</b> 107 terraces generated based on buildings shown in figure 4.2 (Contains OS data © 2018). .....	69
<b>Figure 4.9.</b> Access point calculation for the asset (Contains OS data © 2018).....	70
<b>Figure 4.10.</b> A single-building terrace can directly access an asset if close enough (Contains OS data © 2018). .....	71
<b>Figure 4.11.</b> Explanation of the access angle for a terrace (Contains OS data © 2018)..	72
<b>Figure 4.12.</b> Pick up the access point for each building within the terrace (Contains OS data © 2018). .....	73
<b>Figure 4.13.</b> For asset 4, calculate the shortest path from the asset to a building (Contains OS data © 2018). .....	73
<b>Figure 4.14.</b> Different types of nodes and edges in a distribution network.....	74
<b>Figure 4.15.</b> Computational implementation of the spatial heuristic algorithm. ....	76
<b>Figure 4.16.</b> Electricity transmission networks connecting 11kv and 33kv substations in Newcastle upon Tyne (Robson, 2017).....	77
<b>Figure 4.17.</b> Generated synthetic electricity distribution networks in Newcastle upon Tyne. ....	78
<b>Figure 4.18.</b> Synthetic network result may change depending on different $d_{thresh}$ value. 80	
<b>Figure 4.19.</b> NPG data of electricity distribution networks in Newcastle upon Tyne (Contains NPG data © 2018).....	82
<b>Figure 4.20.</b> (A). Difficulty in retrieving topology from NPG data. (B). Difficulty in retrieving expected network instance boundary (orange circle) (Contains NPG data © 2018).....	82
<b>Figure 4.21.</b> Location of actual feeders and synthetic feeders, with regards to roads (Contains OS and NPG data © 2018).....	83
<b>Figure 4.22.</b> Errors of commissions (grey circles) (Contains OS and NPG data © 2018).	

.....	84
<b>Figure 4.23.</b> Error of omissions (grey circles) (Contains OS and NPG data © 2018)....	85
<b>Figure 4.24.</b> Definition of difference angle (Contains OS and NPG data © 2018). .....	86
<b>Figure 4.25.</b> Distribution of difference angles. ....	87
<b>Figure 4.26.</b> Large difference angles caused by different feeder layout (Contains OS and NPG data © 2018).....	87
<b>Figure 4.27.</b> Large difference angles within yellow circle (Contains OS and NPG data © 2018). ....	88
<b>Figure 4.28.</b> Different ways to define distance from a road to a terrace of building (Contains OS and NPG data © 2018). ....	88
<b>Figure 4.29.</b> Location of chosen cities or regions for algorithm transferability test. ....	90
<b>Figure 4.30.</b> Synthetic electricity distribution networks for Greater London, where each colour represents a single network instance.....	92
<b>Figure 4.31.</b> Algorithm running time for different test areas. ....	92
<b>Figure 5.1.</b> NGN network for Newcastle upon Tyne (Contains NGN Data © 2018). ..	101
<b>Figure 5.2.</b> Different sub-systems within NGN network data, each in different colours (Contains NGN data © 2018). ....	102
<b>Figure 5.3.</b> General work flow for gas network integration.....	103
<b>Figure 5.4.</b> Absence of actual data in some area of the city (Contains NGN data © 2018). .....	104
<b>Figure 5.5.</b> All the buildings that are too far (distance > 50 meters) from NGN network. They are indicated within the red circles (Contains NGN data © 2018).....	105
<b>Figure 5.6.</b> Road segments fetched, which are nearest to the fetched buildings (Contains NGN data © 2018). ....	106
<b>Figure 5.7.</b> CSEP nodes in the NGN network data (Contains NGN Data © 2018). ....	107
<b>Figure 5.8.</b> Before connecting a sub network instance to NGN network (Contains NGN Data © 2018).....	108
<b>Figure 5.9.</b> After connecting a sub network instance to NGN network (Contains NGN Data © 2018).....	108



<b>Figure 5.10.</b> Infer flow direction on the sub network instance (Contains NGN data © 2018).....	109
<b>Figure 5.11.</b> Completed gas main pipe network in Newcastle upon Tyne (Contains NGN Data © 2018). .....	110
<b>Figure 5.12.</b> (A) Completed gas main pipe network, with flow direction, (B) Gas distribution network to the buildings, with flow direction encoded (Contains NGN data © 2018). .....	111
<b>Figure 5.13.</b> Gas distribution network (including service pipes) generated for Newcastle upon Tyne (Contains NGN Data © 2018). .....	112
<b>Figure 5.14.</b> Area for explaining parameter sensitivity of $d$ (Contains NGN Data © 2018).....	113
<b>Figure 5.15.</b> Buildings fetched ( $d = 100$ meters) (Contains NGN Data © 2018).....	113
<b>Figure 5.16.</b> Buildings fetched ( $d = 50$ meters) (Contains NGN Data © 2018).....	114
<b>Figure 5.17.</b> Buildings fetched ( $d = 25$ meters) (Contains NGN Data © 2018).....	114
<b>Figure 5.18.</b> Synthetic main pipes ( $d = 100$ meters) (Contains NGN Data © 2018).....	115
<b>Figure 5.19.</b> Synthetic main pipes ( $d = 50$ meters) (Contains NGN Data © 2018).....	116
<b>Figure 5.20.</b> Synthetic main pipes ( $d = 25$ meters) (Contains NGN Data © 2018).....	116
<b>Figure 5.21.</b> Synthetic service pipes ( $d = 100$ meters) (Contains NGN Data © 2018). ..	117
<b>Figure 5.22.</b> Synthetic service pipes ( $d = 50$ meters) (Contains NGN Data © 2018). ..	117
<b>Figure 5.23.</b> Synthetic service pipes ( $d = 25$ meters) (Contains NGN Data © 2018). ..	118
<b>Figure 5.24.</b> Validation area 1 (Contains NGN Data © 2018).....	119
<b>Figure 5.25.</b> Validation area 2 (Contains NGN Data © 2018).....	120
<b>Figure 5.26.</b> Validation area 3 (Contains NGN Data © 2018).....	120
<b>Figure 5.27.</b> Water supply network data for Newcastle upon Tyne (Contains NWG Data © 2018).....	122
<b>Figure 5.28.</b> General work flow of water supply network integration. ....	123
<b>Figure 5.29.</b> Simple example of the water flow infer algorithm. ....	126
<b>Figure 5.30.</b> WDA representation for Newcastle upon Tyne. (Contains NWG Data © 2018).....	129
<b>Figure 5.31.</b> (A) Water main pipe network, with flow directions. (B) Water distribution	

network to the buildings, with flow direction calculated (Contains NWG Data © 2018). .....	130
<b>Figure 5.32.</b> Fine scale water distribution networks (including service pipes) in Newcastle upon Tyne (Contains NWG Data © 2018). .....	131
<b>Figure 5.33.</b> Available sewer network data (CityCAT Model) for Newcastle upon Tyne. ....	133
<b>Figure 5.34.</b> Location of the pumps and outflow nodes in CityCAT sewer network. ....	133
<b>Figure 5.35.</b> General work flow for sewer network integration work. ....	135
<b>Figure 5.36.</b> (A) Sewer main network, with flow directions. (B) Fine scale sewer network with buildings integrated. ....	136
<b>Figure 5.37.</b> Fine scale sewer network generated, which includes sewer service pipes. ....	137
<b>Figure 5.38.</b> DTM layer used in the algorithm (Contains OS data © 2018). ....	139
<b>Figure 5.39.</b> A simple example to illustrate generic sewer flow infer algorithm. ....	140
<b>Figure 5.40.</b> Validation of flow direction, <i>inferred by the algorithm</i> . ....	141
<b>Figure 5.41.</b> Validation of flow direction, <i>inferred by only using the DTM layer</i> . ....	142
<b>Figure 5.42.</b> Algorithm flow of integrating utility assets to electricity distribution networks. ....	145
<b>Figure 5.43.</b> Location of utility assets and vital substations in Newcastle upon Tyne (Contains OS data © 2018). ....	146
<b>Figure 5.44.</b> Utility asset (regulate site in this case) integrated into electricity distribution networks (Contains OS data © 2018). ....	146
<b>Figure 5.45.</b> PostGIS ITRC database schema. ....	147
<b>Figure 5.46.</b> An example of using ITRC schema to store network dependency. ....	148
<b>Figure 5.47.</b> Location of utility assets and vital substations in London. ....	148
<b>Figure 6.1.</b> Case study area to develop road network generation algorithm (from Google Maps 2018). ....	154
<b>Figure 6.2.</b> Location of case study area in Newcastle upon Tyne (Contains OS data © 2018). ....	154

<b>Figure 6.3.</b> Input data sets for the case study area (Contains OS data © 2018). .....	155
<b>Figure 6.4.</b> Flow of road network generation algorithm.....	156
<b>Figure 6.5.</b> MST generation (Contains OS data © 2018). .....	157
<b>Figure 6.6.</b> MST partitioned into 29 clusters (Contains OS data © 2018). .....	159
<b>Figure 6.7.</b> Constrained Delaunay triangulation result (Contains OS data © 2018). ....	160
<b>Figure 6.8.</b> Simple example about road segments. ....	161
<b>Figure 6.9.</b> Generated road segments (Contains OS data © 2018).....	162
<b>Figure 6.10.</b> Chiakin algorithm example. ....	163
<b>Figure 6.11.</b> Smoothed road segments (Contains OS data © 2018). .....	163
<b>Figure 6.12.</b> Final result of synthetic road network (Contains OS data © 2018). .....	164
<b>Figure 6.13.</b> Synthetic and ITN road network (Contains OS data © 2018). .....	165
<b>Figure 6.14.</b> Topology comparison of synthetic and ITN road network.....	167
<b>Figure 6.15.</b> Residential buildings (area > 30m <sup>2</sup> ) reserved for the case study area (Contains OS data © 2018). .....	169
<b>Figure 6.16.</b> Synthetic electricity network generated ( <b>based on synthetic road network</b> ) (Contains OS data © 2018). .....	170
<b>Figure 6.17.</b> Reference electricity network generated ( <b>based on ITN network</b> ) (Contains OS data © 2018). .....	170
<b>Figure 6.18.</b> Synthetic feeders and reference feeders (Contains OS data © 2018). ....	172
<b>Figure 6.19.</b> Topology comparison of the synthetic and reference feeder networks. ....	172
<b>Figure 6.20.</b> Visual result of building-substation dependency (Contains OS data © 2018).....	173
<b>Figure 6.21.</b> Parameter sensitivity of $\epsilon$ (Contains OS data © 2018).....	175
<b>Figure 6.22.</b> Location of the three test areas in Newcastle. ....	177
<b>Figure 6.23.</b> Input data for test area 1 (Contains OS data © 2018). .....	178
<b>Figure 6.24.</b> Input data for test area 2 (Contains OS data © 2018). .....	178
<b>Figure 6.25.</b> Input data for test area 3 (Contains OS data © 2018). .....	179
<b>Figure 6.26.</b> Synthetic and ITN road network in test area 1 (Contains OS data © 2018). .....	179
<b>Figure 6.27.</b> Synthetic and ITN road network in test area 2 (Contains OS data © 2018).	

.....	180
<b>Figure 6.28.</b> Synthetic and ITN road network in test area 3 (Contains OS data © 2018).	
.....	180
<b>Figure 6.29.</b> Degree distributions of synthetic and ITN road networks in 3 test areas.	182
<b>Figure 6.30.</b> Closeness centrality distribution of synthetic and ITN networks in 3 test areas.	183
<b>Figure 6.31.</b> Generated electricity network in test area 1, based on <b>synthetic road network</b> (Contains OS data © 2018).	185
<b>Figure 6.32.</b> Generated electricity network in test area 1, based on <b>ITN road network</b> (Contains OS data © 2018).	185
<b>Figure 6.33.</b> Generated electricity network in test area 2, based on <b>synthetic road network</b> (Contains OS data © 2018).	186
<b>Figure 6.34.</b> Generated electricity network in test area 2, based on <b>ITN road network</b> (Contains OS data © 2018).	186
<b>Figure 6.35.</b> Generated electricity network in test area 3, based on <b>synthetic road network</b> (Contains OS data © 2018).	187
<b>Figure 6.36.</b> Generated electricity network in test area 3, based on <b>ITN road network</b> (Contains OS data © 2018).	187
<b>Figure 6.37.</b> Synthetic and reference feeder networks for test area 1.	188
<b>Figure 6.38.</b> Synthetic and reference feeder networks for test area 2.	188
<b>Figure 6.39.</b> Synthetic and reference feeder networks for test area 3.	189
<b>Figure 6.40.</b> Degree distributions of the reference and synthetic feeder networks for three areas.	190
<b>Figure 6.41.</b> Closeness centrality distribution of the reference and synthetic feeder networks for three areas.	191
<b>Figure 7.1.</b> ITRC schema.	198
<b>Figure 7.2.</b> General pipe line for ITRC schema.	198
<b>Figure 7.3.</b> General pipe line for PgRouting approach.	200
<b>Figure 7.4.</b> The actual detailed flow to write network into PgRouting, supposing writing	

electricity distribution network.....	201
<b>Figure 7.5.</b> An example of Neo4j property graph.....	203
<b>Figure 7.6.</b> General pipe line for the hybrid database approach.....	204
<b>Figure 7.7.</b> Linking PostGIS and Neo4j using <b>node_id</b> and <b>edge_id</b> .....	205
<b>Figure 7.8.</b> Four common and simple scenarios of retrieving data using both databases. .....	206
<b>Figure 7.9.</b> Three types of data used in this test.....	208
<b>Figure 7.10.</b> The network instance of size 800, in the type 1 network data.....	209
<b>Figure 7.11.</b> The type 2 network data, with size being ‘Newcastle’. Each colour in the figure refers to a single network instance.....	210
<b>Figure 7.12.</b> The type 3 network data, with size being ‘UK’.....	210
<b>Figure 7.13.</b> Performance comparison of writing different sized network data.....	211
<b>Figure 7.14.</b> Performance comparison of reading different size network.....	213
<b>Figure 7.15.</b> Pipe lines for shortest path query on type 1 and type 2 network data.....	215
<b>Figure 7.16.</b> Pipe lines for shortest path query on type 3 network data.....	215
<b>Figure 7.17.</b> Performance comparison of performing shortest path query on different sized network data.....	216
<b>Figure 7.18.</b> The ITN network (Contains OS data © 2018).....	219
<b>Figure 7.19.</b> The IRN network (Contains OS data © 2018).....	220
<b>Figure 7.20.</b> A closer view of the IRN, with regards to the building layout (Contains OS data © 2018).....	220
<b>Figure 7.21.</b> Entire city scale electricity distribution network data in Newcastle upon Tyne. Each colour refers to a single network instance.....	221
<b>Figure 7.22.</b> The CityCAT flooding footprint.....	222
<b>Figure 7.23.</b> The IRN and city centre node.....	223
<b>Figure 7.24.</b> Pipe lines to resolve the IRN complex query.....	224
<b>Figure 7.25.</b> 2397 disrupted edges (in Cyan) in the IRN.....	225
<b>Figure 7.26.</b> 5279 building nodes that cannot reach city centre due to flood.....	226
<b>Figure 7.27.</b> Performance comparison of executing IRN complex query.....	226
<b>Figure 7.28.</b> Pipe lines for <b>ITRC schema</b> and <b>PgRouting</b> , to resolve complex query on	

Newcastle Electricity Network. ....	228
<b>Figure 7.29.</b> Pipe lines for <b>hybrid database</b> , to resolve complex query on Newcastle Electricity Network. ....	229
<b>Figure 7.30.</b> Performance comparison on complex query on Newcastle Electricity Network.....	230
<b>Figure 7.31.</b> Electricity distribution networks of London. Each colour refers to a network instance.....	231
<b>Figure 7.32.</b> Performance comparison on performing writing, reading, and shortest path queries on London electricity network data.....	232
<b>Figure 7.33.</b> Synthetic random hazards used for complex queries. ....	234
<b>Figure 7.34.</b> Performance comparison on performing <b>complex query 1</b> on London electricity network data.....	235
<b>Figure 7.35.</b> Pipe lines to resolve <b>complex query 2</b> on London electricity network data. ....	237
<b>Figure 7.36.</b> Performance comparison on performing <b>complex query 2</b> on London electricity network data.....	238
<b>Figure 7.37.</b> A prototype platform for geospatial infrastructure network inference and management. ....	241
<b>Figure 9.1.</b> Architecture of three layer decision support system developed by Sabeur et al (2016). ....	266

## List of Tables

<b>Table 2.1.</b> Problems and challenges cities are facing due to artificial factors. ....	7
<b>Table 2.2.</b> Natural hazards which can threaten cities. ....	8
<b>Table 2.3.</b> Some common types of models applied to city. ....	10
<b>Table 2.4.</b> Common properties of a network model. ....	15
<b>Table 2.5.</b> A selection of related research of infrastructure resilience models. ....	18
<b>Table 2.6.</b> Major types of approaches of modelling interdependent infrastructures. ....	21
<b>Table 2.7.</b> Comparison of related ontologies and models with regards to urban infrastructures. ....	25
<b>Table 3.1.</b> Examples of “Is-A” and “Part-of” relationships. ....	34
<b>Table 3.2.</b> Definition of geometry entities. ....	34
<b>Table 3.3.</b> Common spatial relationships and semantic examples. ....	35
<b>Table 3.4.</b> Description of common attributes. ....	36
<b>Table 3.5.</b> Attributes that can be inherited by subclass of Building. ....	39
<b>Table 3.6.</b> Attributes related to the Utility Networks. ....	44
<b>Table 3.7.</b> Attributes associated with Road. ....	47
<b>Table 3.8.</b> Attributes associated with Rail Way or Metro Way. ....	48
<b>Table 3.9.</b> Explanations for basic notations. ....	51
<b>Table 3.10.</b> Description on notations with regards to <b>R</b> . ....	54
<b>Table 3.11.</b> Scopes for mappings $f_{w_e}$ , $f_{g_e}$ , $f_{s_e}$ , $f_{r_e}$ , $f_{t_e}$ , $f_{m_e}$ . ....	55
<b>Table 3.12.</b> Scopes of mappings $f_{b_r}$ , $f_{b_e}$ , $f_{b_w}$ , $f_{b_s}$ , $f_{b_g}$ . ....	56
<b>Table 4.1.</b> Related studies in generating geospatial infrastructure network. ....	59
<b>Table 4.2.</b> Description of different types and edges and nodes in a distribution network. .....	75
<b>Table 4.3.</b> Sensitivity of parameter $d_{thresh}$ . ....	79
<b>Table 4.4.</b> Effect of applying Delaunay triangulation process. ....	80

<b>Table 4.5.</b> Chosen cities or regions for test algorithm transferability.....	90
<b>Table 4.6.</b> Characteristics of synthetic networks generated for test cities or regions.....	91
<b>Table 4.7.</b> Percentage of cluster-asset dependency calculation time.....	93
<b>Table 4.8.</b> Notations used to assess time complexity of CADC process.....	94
<b>Table 4.9.</b> Values of notations for the test area.....	94
<b>Table 4.10.</b> Change of value $r * (1 + \log_2 (N_c * r))$ , when area size is doubled.....	96
<b>Table 5.1.</b> Change of pipe total length as $d$ changes.....	118
<b>Table 5.2.</b> Error of omissions and commissions for validate gas main pipes.....	121
<b>Table 5.3.</b> Validation result for the above three areas.....	121
<b>Table 5.4.</b> Utility network dependencies.....	143
<b>Table 6.1.</b> Related approaches for automatic road network generation.....	153
<b>Table 6.2.</b> Spatial comparison of synthetic and ITN road network.....	166
<b>Table 6.3.</b> Spatial comparison on the reference and synthetic feeder networks.....	171
<b>Table 6.4.</b> Building-substation dependency comparison result.....	174
<b>Table 6.5.</b> Evaluation of synthetic road network based on different $\varepsilon$ values.....	174
<b>Table 6.6.</b> Basic information of test area.....	177
<b>Table 6.7.</b> Validation of synthetic road network in testing areas.....	181
<b>Table 6.8.</b> Network size of ITN and synthetic road networks in three areas.....	181
<b>Table 6.9.</b> Spatial comparison on the reference and synthetic feeder networks for three areas.....	189
<b>Table 6.10.</b> Network size of reference and synthetic feeder network in three areas.....	189
<b>Table 6.11.</b> Comparison result on building-substation dependency.....	192
<b>Table 7.1.</b> Shortest path query to be executed.....	214
<b>Table 7.2.</b> Breakdown of IRN complex query.....	224
<b>Table 7.3.</b> Four tasks for complex query on Newcastle Electricity Network.....	228
<b>Table 7.4.</b> Result of complex query on Newcastle Electricity Network.....	229
<b>Table 7.5.</b> Four tasks in <b>complex query 1</b> on London electricity network data.....	235



<b>Table 7.6.</b> Result of <b>complex query 1</b> on London electricity network data.....	235
<b>Table 7.7.</b> Two tasks in <b>complex query 2</b> on London electricity network data. ....	236
<b>Table 7.8.</b> Result of <b>complex query 2</b> on London electricity network data.....	237

## List of Listings

<b>Listing 4.1.</b> The pseudo code for topology generation process.....	63
<b>Listing 4.2.</b> The pseudo code for topology generation process.....	68
<b>Listing 5.1.</b> Pseudo code for the gas network infer algorithm. ....	105
<b>Listing 5.2.</b> Pseudo code for building service infer algorithm (gas).....	111
<b>Listing 5.3.</b> Pseudo code for water flow infer algorithm.....	127
<b>Listing 5.4.</b> Building service infer algorithm (water).....	130
<b>Listing 5.5.</b> Pseudo code for the building service infer algorithm (sewer). ....	135
<b>Listing 5.6.</b> Pseudo code for the generic sewer flow infer algorithm. ....	138
<b>Listing 6.1.</b> MST partitioning operation (Zhou et al., 2009).....	158
<b>Listing 6.2.</b> Pseudo code of road segments generation. ....	162
<b>Listing 7.1.</b> The building-ITN integration algorithm. ....	219

## Glossary

**CADC process:** Cluster-Asset Dependency Calculation process

**CAS:** Complex Adaptive System

**FIS:** Fuzzy Inference System

**HSPN:** Hybrid Social Physical Network

**IFI:** Infrastructure Failure Interdependency

**ITN:** Integrated Transport Network, which is a data from MasterMap

**IRN:** Integrated Road Network, which is a synthetic road network in Chapter 7

**INSPIRE:** Infrastructure for Spatial Information in the European Community

**MST:** Minimum Spanning Tree

**NDP:** Network Design Problem

**NGN:** Northern Gas Networks

**NPG:** Northern Powergrid

**NWG:** Northumbria Water Group

**OS:** Ordnance Survey

**OTN:** Ontology for Transport Network

**PTN:** Public Transport Network

**RDMS:** Relational Database Management System

**SCADA:** Supervisory Control and Data Acquisition

**WDA:** Water Distribution Area



# Chapter 1. Introduction

## 1.1 Urban Infrastructure Challenges

Rapid uncontrolled urbanization has become a significant global problem which needs to be addressed in order to develop a sustainable biosphere (EU, 2010). Currently more than 52% of humans worldwide are living within urban areas (UN, 2013), and by 2050 this value is expected to reach 64% (UN, 2014). This irreversible process of urbanization is leading to an emergence of mega-cities (>10 million inhabitants) (Kourtiti, et al., 2013). Such urbanization has in general raised living standards, with improved water supplies and sewage systems, residential and official buildings, education and health service, as well as public transport (D'Agostino, 2014; Yin, et al., 2015), but also brings issues such as pollution, crime and poverty (Hu, et al., 2013; Mohit, et al., 2017).

Modern cities are comprised of spatially and temporally complex relationships between urban infrastructure systems and the buildings and residents they service (Guy et al, 2001). These urban infrastructure systems, including energy, water supply, waste, power and transport, provide the resources required to support the day-to-day functioning of cities (Murray and Grubestic, 2007). The integrity and reliability of these urban infrastructure assets, and the resources and services they provide are crucial for assuring public health, environmental sustainability, national security, social and economic productivity (HM Treasury and Infrastructure UK, 2014).

Managing spatial data of fine spatial scale critical infrastructure networks is essential in many modern urban applications, such as smart city sensing (Gabrys, 2014; Hancke, et al., 2013; Perera, et al., 2014), smart neighbourhood (Lara, et al., 2016; Piotrowski, et al., 2014), digital twin (Mohammadi, et al., 2017; Shelton, et al., 2015), metering studies of local energy distributions (Albaugh, et al., 2004; Kleissel, et al., 2010; Karnouskos, et al., 2007),

infrastructure failure positioning and repair (Fang, et al., 2016; Hu, et al., 2016; Soltani-Sobh, et al., 2016), infrastructure planning and decision support (Gurung, et al., 2015; Malekpour, et al., 2016; Narayanaswami, 2007), and evaluating impact of spatial event on infrastructure networks (Borden, et al., 2007; Sokolov, et al., 2013).

As city becomes more complex, the networked infrastructure systems become more vulnerable, as disruption can potentially cascade through individual and interdependent networks leading to impacts far beyond the original spatial footprint of the disturbance (Royal Academy of Engineering, 2011), potentially causing great disruption and loss for the society. For example, a power outage stroke entire country of Italy in September, 2003 which lasted for 19 hours (Rosato et al, 2008). The event was reported to cause an economic loss of € 1,182 million (Schmidthaler, et al., 2016), with more than 100 trains stranded and all flights (from or to Italy) cancelled (Rosato, et al., 2008). The initial cause was just storm damage on few electricity cables serving electricity from Switzerland to Italy (Rosato, et al., 2008). Likewise, the North America blackout in 2003 was reported to cause \$ 6 billion loss in the US and 18.9 million lost work hours in Canada (Bennet, et al., 2005). The blackout ended up shutting down oil refineries and pipes, transport systems and manufacturing industries for more than 24 hours, while this event was initially triggered by failure of few power transmission lines in Ohio (St-Pierre, et al., 2000).

Therefore, it is crucial to characterise the interdependency of critical infrastructure networks (Holmgren, et al., 2006; Lhomme, et al., 2013; Ouyang, 2014; Rinaldi, 2001) and understand how these failures occur and cascade to the buildings, which require infrastructure services. However, at fine geospatial scale, little attention has been made on the application of infrastructure network data and infrastructure interdependency models. This is due to the absence of generic information management tool on such data. There are three major reasons.

First, it is very rare that fine scale spatial data on critical infrastructure networks are easily available. Often data is held within commercially sensitive organisations (utility companies) and may be incomplete topologically and/or spatially (Bon, 2017; Fu, et al., 2008; Jaw, et al.,

2013). If such data is not available, it is imperative to have approach that can infer plausible layout of infrastructure network for understanding the spatial connectivity between infrastructure assets and buildings (Bon, 2017; Cavallaro, et al., 2014).

Secondly, geospatial infrastructure network data come from different sources, and therefore data can be different in terms of what information is encoded and how the information is encoded (Almeida, et al., 2009; Fu, et al., 2008; Hepp, 2007), and integrating data from different sources can be a challenge (Popovich, et al., 2014). Therefore, an ontology is needed to explicitly define what entities, attributes, and relationships are required to represent heterogeneous infrastructure networks (Fu, et al., 2008). Although there are currently some observations on infrastructure network ontologies, such as iCity (Katsumi, et al., 2017), Townology (Berdier, 2007) and Utility Knowledge Ontology (Xu, et al., 2018), none of them is defined in an explicitly spatial manner, or considers the connections between critical infrastructure and buildings.

Finally, to efficiently manage and analyse (query) such complex geospatial infrastructure network data, a database system is essential. Spatial relational databases such as PostGIS (Nguyen, 2009; Zheng, et al., 2017), and Oracle Spatial Extension (British Telecom, 2012; Fikjez and Řezanina, 2016) are the traditional solutions for handling coarse spatial scale infrastructure network such as electricity transmission grid in the UK (Barr, et al., 2016). However, fine scale geospatial infrastructure network is more complicated in terms of more nodes/edges. It is not clear whether or not traditional database approaches would be efficient in querying such complex networks. Recently, NoSQL database is proposed for more efficient management of network data, such as social network (Cattuto, et al., 2013), biology network (Yoon, et al., 2017), and knowledge graph (Lin, et al., 2017). However, there is no relevant study in applying NoSQL databases to manage geospatial infrastructure network data.

## 1.2 Aims and Objectives

Have access to good quality geospatial data on infrastructure networks is a challenge but can open up opportunities in different digital urban models and applications. The research aim of this thesis is to develop approaches for the inference and management of fine scale geospatial urban utility infrastructure networks. To address this aim, four objectives have been identified:

1. Review the research field pertaining geospatial urban infrastructure network models and identify the research gaps in the inference and management of geospatial infrastructure network data.
2. Develop a geospatial ontology, to conceptually model the knowledge of the entities, attributes and relationships that are indispensable to represent fine scale urban infrastructure networks. The focus is to understand the spatial connectivity between infrastructure assets and buildings.
3. Develop a generic approach to infer geospatial layout of the utility infrastructure network if actual data does not exist or only partially exists. The approach should be transferable so that it can be applied in different major utility sectors (electricity, gas, water supply and waste water).
4. Develop a database approach that is able to encode, manage, and query the complex geospatial infrastructure network data in an efficient manner. In particular, several potential database approaches will be investigated, and performance benchmarking tests will be carried out to decide the most appropriate one.

The research will investigate new approaches to the integrated inference, management and analysis of the complex utility infrastructure networks. Such approaches should allow the highly granular representation of utility network connectivity to be represented in a spatially explicit manner, employing methods of data and information management to ensure they are scalable and generic.



### **1.3 Thesis Structure**

The remainder of the thesis addresses the aims and objectives as set out above in Section 1.2 and is split into eight chapters. Chapter 2 reviews the previous research which has been undertaken, in terms of critical infrastructure networks, the geospatial infrastructure network models, and identifies current challenges and objectives (the Objective 2, 3, and 4) in inference and management of the geospatial data on the fine scale infrastructure networks.

Objective 2 is addressed in Chapter 3, where a geospatial ontology on fine scale infrastructure network is proposed.

Objective 3 is addressed in Chapter 4, 5, and 6. In Chapter 4, a generic spatial heuristic algorithm is proposed, which can infer layout of infrastructure network, based on the layout of infrastructure assets, buildings, and a road network. This algorithm is applied and validated in generating city scale electricity distribution networks. Then Chapter 5 discusses transferability of the algorithm, where the algorithm is applied to infer layout of gas, water supply and sewer networks. Chapter 6 further proposes a road network generation algorithm, when it is even not possible to access road network layout.

Objective 4 is addressed in Chapter 7, where database performance benchmarking tests are done to decide an appropriate database approach to handle such complex geospatial infrastructure network data.

Chapter 8 discusses the results and major findings from this research and critiques the employed methods. Chapter 9 finally presents the conclusions of this thesis with potential future outlook in this research field.

## **Chapter 2. Literature Review**

### **2.1 The Changing Cities**

During the last decade, rapid urbanisation has triggered a series of global processes which are reshaping the world. One of them is the long-term trend of population movement to the cities (Castles, et al., 2013). Only a few centuries ago, the urban population was about 20%, while by 2007, more than 50% of the world's population had settled in urban areas, resulting in cities gradually taking over 'power' from their hinterlands (Kourtit, et al., 2013). According to the United Nations, this value will continue to rise, and global percentage is expected to reach 64% by 2050 (United Nations, 2014). In Europe, the urbanisation rate will be even higher, reaching 83% by 2050 (European Union, 2010). This long-term trend is primarily driven by two forces, which are the exponential growth of world population (annual growth rate at 1.2%) and rural-urban shift (when the urban area is generally more attractive than rural settlement in terms of favourable opportunities and services) (Tacoli, et al., 2015).

The population movement further means increasing requirement of living standards in cities (Nijkam, et al., 2013). These living standards can be tangible or intangible. These include residential and office buildings, water supply and drainage systems, public transports, energy supply, ICT (information and communication technology), education and health services (Yin, et al., 2015). Rural population migrated to the urban areas for more favourable access to living quantities, and this in turn also improved the regional social and economic prosperity of the city and created job opportunities (LeGates, et al., 2015). Due to the increasing population, cities are expected to evolve into urban agglomerations or megacities (inhabitants of more than 10 million) (Nijkamp, et al., 2013). It is believed that in this way, modern city is becoming a complex system, comprised of many units (physical and geographical structures, citizen, and ubiquitous social, economic and environmental aspects) which actively interact with each other (Lombardi, et al., 2012).

This complex system, the modern city, which might look promising in some ways (higher

living quality, better education and job opportunities, etc.), is also facing new problems and challenges that emerged recently (Jenks, et al., 2009). Some problems, in the form of ecological, environmental and social issues, are caused by artificial factors, and examples are shown in table 2.1.

<b>Problem &amp; Challenge</b>	<b>Description and Example</b>
Air Pollution	Currently cities account for 70% of global greenhouse gas emissions. This uncontrolled process leads to serious air quality deterioration in many cities. A research found that for all 18 megacities in the world, only 5 of them have ‘fair’ air quality while the other 13 have ‘poor’ air quality (Gurjar, et al., 2008).
Water Pollution	Water contamination can cause degradation of aquatic ecosystems or public health problem. Sometimes, poor decisions in selecting construction and industry sites can lead to water reservoir pollution. In Istanbul, Turkey, all of the 6 water reservoirs faced eutrophic issue due to this and clean water supply to all 10 million inhabitants in the city was seriously disrupted (Baykal, et al., 2000).
Traffic Congestion	A very common problem in both developing and developed countries. In 2011, traffic congestion in USA was so severe that urban Americans had to spend 5.5 billion more hours and purchase 2.9 billion extra fuel for the total congestion cost of \$ 121 billion (Schrank, et al., 2012).
Crime	When urbanisation rate increases, so does crime (Krivo, et al., 1996). Japan is always viewed as a country of low crime rate. However, an increasing trend has been observed recently. Mean annual increase of assault and robbery rate between 1996 and 2006 were 10.7% and 7.4%, much higher than other developed countries. The urbanisation process is considered as the leading factor (Halicioglu, et al., 2012).

**Table 2.1.** Problems and challenges cities are facing due to artificial factors.

These issues occur due to artificial factors and therefore can be relieved or tackled from government policies and sustainable development decisions (Jenks, et al., 2009). However there exist other problems which are more difficult to foresee, the climate and environment induced problems. As city becomes more complex (with regards to its spatial extent and physical configuration), it is more sensitive and vulnerable to natural hazards (Klein, et al., 2003). Some common natural hazards threatening cities are shown in table 2.2.

<b>Natural Hazards</b>	<b>Description and Example</b>
Flood	Excessive rain fall and inefficient urban drainage system can cause this issue. In the UK, it is the most serious natural hazard, which has threatened 1/6 (about 5 million) properties. The flood also gave rise to severe economic losses, at the rate of £ 3.2 billion in the year 2007 (Thorne, 2014).

Drought	The contrary of flood, can directly affect freshwater resources, and further cause water shortage in cities. Drought is affecting cities globally, including some of the tropical countries such as Singapore. Between January and March, 2014, some part of Singapore city received less than 1 mm rain, resulting drinking water shortage for 5.6 million residents in the city (Buurman, et al., 2016).
Earthquake	Earthquake can be already destructive by itself, and sometimes it can trigger other hazards such as Tsunami. In 2011, an earthquake stroke Tohoku region of Japan, followed by a tsunami which eventually submerged millions of properties in Fukushima Ken (Fujii, et al., 2011). Due to damage of Fukushima nuclear station, horrible nuclear pollution still exists today.
Volcano Eruption	This event is less frequent than the previous three, but can be equally devastating to modern cities. In 2010, the Eyjafjallajökul volcano in Iceland erupted, which was considered as a small eruption event. But the spreading volcanic ashes interrupted the major airline network in Europe. Thousands of flights were cancelled from, to, or within Europe, creating the highest level of air travel disruption since World War II (Gudmundsson, et al., 2010).

**Table 2.2.** Natural hazards which can threaten cities.

Being able to tackle these problems is essential for any modern city. In order to understand how these social, economic, or environment issues occur and affect city, different city models are developed and employed. They are mostly based on mathematical and computational approaches, and aim to analyse and simulate the dynamic evolution of modern cities (Egger, et al., 2006). There are various types of city models, with each focusing on a specific aspect. Some common types of city models are described and explained in table 2.3.

Different Models	Description and Explanation
Population	This type of model studies city population dynamics and aims to predict population change. A basic population growth equation, the current census data, and city growth scenario (how the city itself expands, does the city develop in a sustainable way or not, etc.) are essential for a good population model (Arnell, et al., 2011; Bettencourt, et al., 2007; Kc, et al., 2011; Lutz, et al., 2011). Population model can be applied at any city as long as necessary input data are available (such as census). But accurate prediction on population is difficult, since it relies on a good assumption of city growth scenario and model developed for one city might be not applicable for another one (Hoornweg, et al., 2016).
Economy	A typical urban economic model involves some input variables (for example, population, spatial structure of city, location of firms or household, etc.), a set of logic or relationships between them, and some output variables (Ueda, et al, 2013). Due to different application purposes, there can be different output variables that reflect urban economics, such as unemployment (Liu, et al., 2013), housing price (Guerrieri,

	<p>et al., 2013), government finance (Mesquita, et al., 2010), etc. Urban economic model is useful for suggesting fairer economic policies to local government (Deng, et al., 2010). However, this model is a branch of microeconomic model, and that means its result will not be very trustful when evaluating economy at macro-scope (Quigley, 2008).</p>
Pollution	<p>Urban pollution models are the numeric models which focus on the mathematic simulation of how a specific urban pollution material spreads (Gobiet, et al., 2000). They can be applied in air pollution (Berkowicz, 2000), noise pollution (Holt, et al., 2007), and water pollution (Volk, et al., 2008), which have high impact of environmental degradation on public health and urban liveability (Wei, et al., 2014). Urban pollution model is useful in assessing long term urban environment change, but it must rely on accurate outputs from other models which describe physical structure of the city (3D building layout, topography, etc.) (Fiedler, et al., 2015).</p>
Natural Hazard	<p>The hazard models tackle the environment threats to the overall functionality and sustainability of urban areas, from almost unpredictable extreme climate and natural events. The typical hazards for example, are flooding (Prodanović, et al., 2009), extreme drought (Gober, et al., 2011), and earthquake (Carreño, et al., 2007). Hazard models focus on the simulation of possible damage to the urban system (buildings, street, etc.) and support decision making in urban pre-hazard fortification and post-hazard reconstruction (Godshalk, 2003). However, like the pollution model, the physical structure of the city must be also given accurately in advance, in order to run any hazard model.</p>
Planning (Land Use)	<p>Urban land has different uses (residential, commercial, infrastructure, etc.) and being able to model urban land use change is essential to city planners, and resource managers (Rahimi, 2016). There are various methods available for modelling land use change. Common methods include machine learning (Samardžić, et al., 2016), deep learning (Varney, 2018), etc. They aim to understand the relationship between the input variables (land use driving forces, such as population density, slope, etc.) and output variables (land use change). Normally two different types of maps are needed: 1) land use maps at different time, and 2) maps of input variables at different time. Both machine learning and deep learning approaches are useful at predicting future land use change, but tuning parameters is difficult, and there can be a risk of over-fitting (Samardžić, et al., 2016; Varney, 2018).</p>
Behaviour	<p>This type of model aims to understand why and how, a specific behaviour such as crime (Malleon, et al., 2009), insurgency (Fonoberova, et al., 2018), or residential choice (Beneson, et al., 2004) occurs in a city. Many urban behaviour models use agent-based model as its backbone. The agent-based model is based on independent interactive units called agent (like resident in the city). Each agent can make decisions according to their own characteristics and can also interact with (and be affected by) other agents (Castle, et al., 2006). Agent-based models help to understand how individual behaviours create aggregating pattern in city, but models can be very sensitive to initial conditions and or small variations in interaction rules (Couclelis, 2002).</p>

Sector	This type of model focuses on functioning of a specific sector of city, for example, congestion in transport system (Jacyna, et al., 2014), failure of power grid (Wang, et al., 2011), etc. It helps us to understand why a system fails or how to improve its efficiency. But due to heterogeneous characteristics of different sectors within city, sector-based models are normally very specific. That means they do not have good transferability (model developed for one sector is almost not usable for another one).
--------	--

**Table 2.3.** Some common types of models applied to city.

It can be seen here, due to the increasing complexity, modern cities are vulnerable to many problems and are facing different challenges. Different urban models are being developed and applied to understand city from different angles. As one essential part of the modern city, critical urban infrastructure plays a vital role, but it did not gain enough public awareness and attention until recently (Steele, et al., 2017).

## 2.2 Critical Urban Infrastructures

The term ‘critical urban infrastructures’ was first introduced by the US government in 1991, to refer to the infrastructures that are indispensable to the functioning of modern city (Murrari, et al., 2007). The reason for identifying of critical urban infrastructures is that they are so vital that their incapability, malfunction or destruction will have devastating impact on the sustainability, social and economic security of a country (D’Agostino, et al., 2014). For example, in the midnight of September 28, 2003, a country scale power outage stroke Italy. The entire country was left black in 12 hours, affecting 56 million people. More than 100 trains were stranded and all flights from, to or within Italy were cancelled (Rosato, et al., 2008). Damages to the societies are believed to be at least € 1.15 billion, which is about 0.1 percent of Italy annual GDP (Schmidthaler, et al., 2016). Likewise, in the afternoon of August 14, 2003, a serious blackout occurred in Northeastern and Midwestern of United States and Ontario province of Canada. More than 55 million people in the US and Canada were affected, with some areas spending two weeks to restore power supply (Bennet, et al., 2005). Moreover, this blackout also created significant chaos in other infrastructures. For example, in water supply system, pressure loss occurred due to pumps lacking power. Loss of pressure will further cause potential contamination in water supply (St-Pierre, et al., 2000). Four

million customers in eight counties of the Detroit water system received 'boil-water order' for four days, because of this issue (Water World, 2003). The total economic loss of this blackout is estimated between 6 and 10 \$ million (ELCON, 2004).

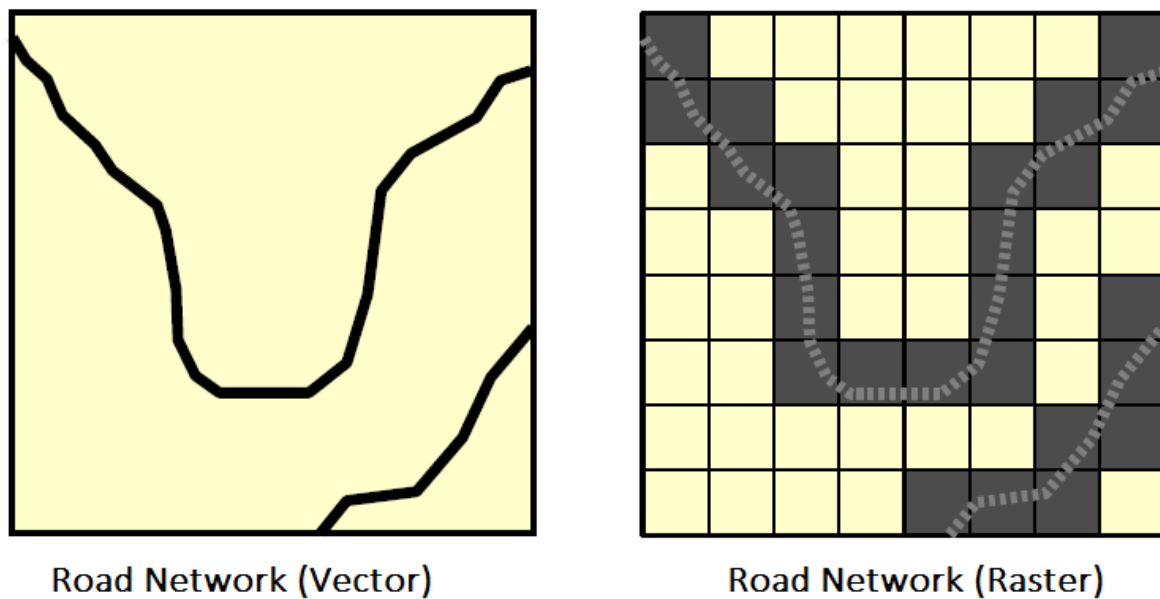
These two examples are related to power systems, but should provide enough insight on how significant a critical urban infrastructure is. Despite heterogeneous configuration and functionalities of urban infrastructure systems in different countries, basic inventory of critical urban infrastructures is generally identified and agreed. They include telecommunications, electricity power systems, transportation, gas systems, water supply and waste water systems (Murray, et al., 2007). Critical urban infrastructures are complex and heterogeneous. The urban infrastructure models, just like other city models have introduced earlier, are developed to explain how the infrastructure systems work or how they fail on a simplified view. These modelling approaches will be discussed in the next section.

### **2.3 Approaches of Modelling Urban Infrastructures**

Critical urban infrastructure encompasses a wide range of engineered systems (transport systems, cable-based electricity power systems, pipe-based water supply systems, etc.) and assets (electricity substations, water pumping stations, etc.). In order to understand the structure and functionality of such complex systems, different modelling approaches have been applied. Two common ones are raster-based model and space syntax.

The raster-based approach is often used in modelling transport system, especially in travel-cost or accessibility related analysis. Georeferenced vector data of the transport system (e.g. road network) is usually needed to generate the raster representation of the system. The connectivity and travel cost of the original system can be represented by the connectivity and attribute (e.g. travel cost) of grid cells in the raster layer (figure 2.1). For example, Delamater et al (2012) employed this approach to analyse the travel cost for residents to access to different health care centres in Michigan. The raster-based approach is relatively simple to

implement, and is efficient when coupling with other raster-based land simulation (Fuglsang, et al., 2011). However, due to its simplicity, this approach itself is not capable of performing more complex analysis on the infrastructure system (resilience, interdependency, e.g.) without coupling with other modelling approaches (such as a graph-based approach) (Schintler, et al., 2007). Another issue with this approach is that, it is difficult to decide the optimal spatial resolution for the raster layer. A coarse spatial resolution is likely to cause unwanted and incorrect connectivity on the transport system, while computational burden of the model can increase tremendously if finer spatial resolution is employed (Delamater, et al., 2012).

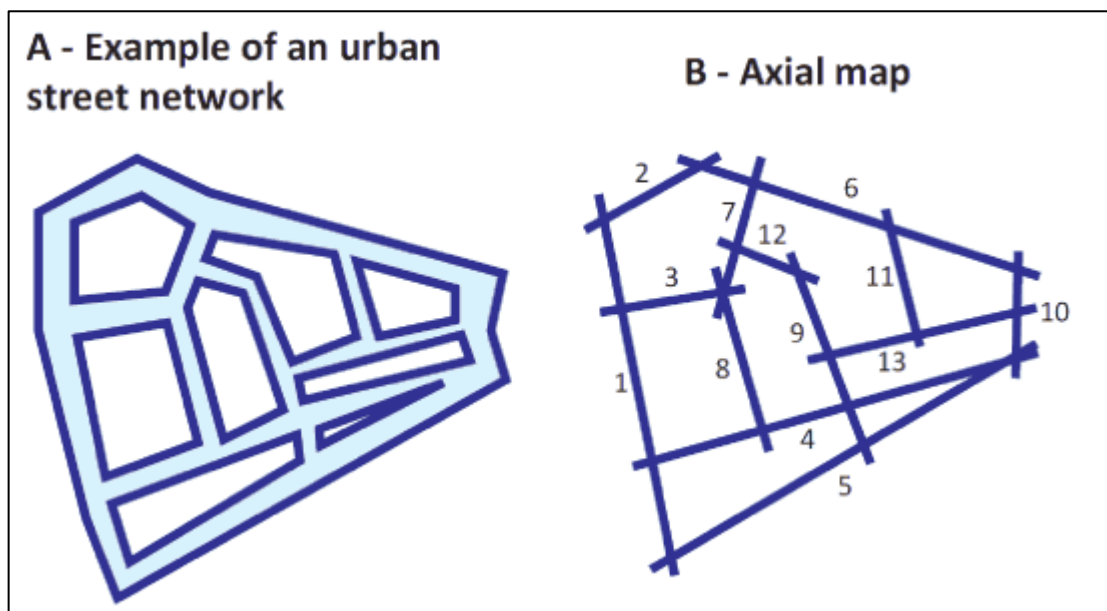


**Figure 2.1.** Example of converting of vector road data to raster cells (Delamater, et al., 2012).

Space syntax (Hillier, et al., 1989) is a set of theories for quantitative analysis on spatial network instances. It uses an *axial map* (figure 2.2) to represent the structure and connectivity of the spatial network (Patterson, 2016). For a network instance, one or more network segments are converted to an *axis* (long-straight line), based on continuity (e.g. based on names of avenues and boulevards, or another qualitative criterion in the street network). The connectivity between different axes are measured based their topological connectivity in space. Space syntax also provides other measures, such as *integration value* (average shortest distance of an axis to all the other axes in the axial map based on connectivity, similar to the *betweenness centrality* in graph model), to predict traffic or resource flows on the spatial network (McCahill, et al., 2008). Space syntax focuses on the representation of connectivity



of space, and has been extensively applied in predicting traffic flow (e.g. pedestrians, vehicles, and bicycles) and planning new streets to accommodate increasing traffic, and it is an efficient modelling approach to be coupled with other traffic models (Duan, et al., 2008; Zheng et al., 2009). However, a major issue is that it over-simplifies the structure of the network (combine multiple segments into one axis) (Patterson, 2016) and thus can lose some connectivity representations in the axial map (e.g. in a street network, a large avenue made of several street segments becomes a single axis, and that means the axial map is unable to represent the connectivity among these street segments inside the axis).



**Figure 2.2.** Example of converting an urban street network to an axial map

(Source: [https://transportgeography.org/?page\\_id=6038](https://transportgeography.org/?page_id=6038)).

Considering the limitation of raster data model and space syntax (in the context of this PhD research), they are not the optimal modelling approaches. As such, it is argued that the classic network/graph model is the most appropriate modelling approach. In spite of the heterogeneity with regards to the physical and engineering configurations and functionalities of different critical urban infrastructures, most of them exhibit a network structure, which allows for the transmission or distribution of material or services (Dunn, et al., 2013). The network theory, a rigorous mathematical tool, is applied to analyse urban infrastructures (Holmgren, et al., 2006; Lhomme, et al., 2013) and support infrastructure design and

management (Wilkinson, et al., 2012).

In network theory, a network concerns itself as the representation of discrete objects (nodes) and relationships (edges) connecting these objects. Mathematically, a network  $G$  can be represented as follows:

$$G = (V, E, f)$$

The network  $G$  is an ordered triplet of  $V, E, f$ , where  $V$  is the set of nodes,  $E$  is the set of edges, and  $f$  is a function that maps each element in  $E$  to an unordered pair of two nodes in  $V$ .

Converting a real-world urban infrastructure to a network model is normally straightforward (Dunn, et al., 2013). The components of an infrastructure system, which generate, consume material or resources (electricity, water, telecom signals, etc.) are represented as nodes (electricity substations, water reservoirs, telecom base station, infrastructure service consumers, etc.). Components that simply allows material or resources to pass through are also represented as nodes (water pumping stations, etc.). Then network edges are generated if there exist flows which allow exchange of material or resources on the corresponding nodes. Depending on the type of infrastructure systems, network edges correspond to actual physical components (in electricity power systems, the cables for example).

In addition to the basic definition, a network model also contains useful properties which allow for quantified analysis on the infrastructure system. The most common properties are explained in table 2.4.

Property	Definition and Application in Infrastructure Research
Direction (edge)	A network can be undirected or directed, depending on whether there should be orientation (direction) on edges. It means whether flow is allowed in both (or only one) directions for an edge. In a network, it is also possible that some edges are directed and others are not (for example, there are one-way and two-way roads in a road network). Any network flow-based analysis, such as traffic flow optimization in transport network (Chiu, et al., 2007), will be based on this important property.

Degree (node)	A network is represented as discrete objects with connection among them. Degree quantifies the connection level of each node. For each node, it measures number of connections to other nodes. This property is used to identify important nodes (those with high degrees), and is useful in infrastructure vulnerability analysis (Apostolakis, et al., 2005), and infrastructure fortification (Matisziw, et al., 2009), for example.
Capacity (edge/node)	Capacity means the maximum amount of material or resources that is allowed to flow through a node or an edge. Some nodes in the infrastructure network model are called demand nodes. They represent consumers which have specific demand of infrastructure material or resources. The capacity and demand properties are important in infrastructure planning and supply / demand analysis (Lucas, et al., 2010).
Weight (edge)	Cost can occur when material or resources travels through infrastructure network. The cost can be for example, travel time, electricity voltage drop or supply water loss (Brandes, et al., 2005). A weight (a numeric value) is associated with each edge to indicate this cost. Weight is an important property in optimizing infrastructure flow to minimize infrastructure loss (Chiu, et al., 2007), or in transport route planning (Delling, et al., 2009), etc.
Path (nodal pair)	Path means a legal travel route from one node to another, based on the topological connectivity of the network. When weight is introduced, shortest path can be calculated, which indicates the path that corresponds to the least total weight. Path and shortest path are important properties in transport route planning (Delling, et al., 2009) and infrastructure planning (Ji, et al., 2007), etc.

**Table 2.4.** Common properties of a network model.

Network model introduced above is simple and straightforward, but it is an essential tool which allows to computationally represent complex and large-scale infrastructure systems. For example, it makes it possible to computationally represent national scale electricity transmission networks of the whole United Kingdom, which allows for further analysis such as identifying vulnerable nodes (transmission substations) and simulating electricity cascading failure at national scale (Barr, et al., 2013). Yazdani et al (2011) modelled the water distribution network (WDN) in the four cities of the US (East-Mersea, Colorado Springs, Kumasi, and Richmond), to study the network vulnerability, the efficiency of demand-supply structure, as well as topology optimization (e.g. where should we remove or add a pipe in the water distribution network). Jacyna et al (2014) developed a computational network model to represent public transport network in the entire of Poland in which transport infrastructures (road, rail, etc.), demand of public transports is characterized. This allows for modelling emission of exhaust gas due to travel demand of within Poland.

The network/graph model is efficient in representing not only the structure and functionality of infrastructure systems, but also the relationships between urban space (such as building) and infrastructure (such as streets). For example, Domingo et al (2019) proposed a graph approach for structural layout analysis on buildings, parcels (neighbourhoods), and roads. They defined roads as the *nodes* in a graph, and a parcel (modelled as a *node*) is connected to a road *via a graph edge*, if the parcel is externally connected to a road. A building (also modelled as a *node*) can connect to a road *via a graph edge*, if the parcel which contains the building, connects to a road. Similarly, Cavallaro et al (2014) employed graph to understand building-street relationship, where streets are modelled as *graph edge* instead of *node*, as it helps to retain street geometry to evaluate efficiency of goods and serviced delivered to the buildings via the street network.

To summarize, these computational network/graph approaches allow representing large and complex infrastructure network to characterize its topological connectivity, network metric as well as its own dynamics. However, it is also essential to study the urban infrastructures with hazards and the interactions between them (Murray, et al., 2007). This will help us better understand why infrastructures are vulnerable to hazards and how to facilitate more stable infrastructures for our cities.

## **2.4 Research Focus on Urban Infrastructure Networks**

The network theory provides us a convenient tool to convert the complex infrastructure systems to network models consisting of nodes and edges. This further allows us to study infrastructure resilience and dependencies / interdependencies (Bozza, et al., 2017), which are identified as the recent research focus on critical urban infrastructures (Ouyang, et al., 2012; Mensah, et al., 2015; Hokstad, et al., 2012; Ouyang, et al., 2014).

### 2.4.1 Resilience of Individual Infrastructure Sectors

Resilience is a historical term and dates back to the 19<sup>th</sup> century. It has been used in many research domains, such as medicine, psychology, and ecology with different definitions (Bozza, et al., 2017). For urban infrastructures, a common definition of resilience is “the joint ability to resist (prevent and withstand) any possible hazards and absorb initial damage, and then to recover to normal operation” (Ouyang, et al., 2012). In other words, resilience is related to two aspects: 1) when hazards occur, how robust the infrastructure system is to still maintain its operation, and 2) after hazards have inflicted damage, the ability for infrastructure system to “bounce back” to its normal operation state.

As introduced earlier, disruption of critical urban infrastructure can be devastating to the modern cities. Modelling resilience of urban infrastructures facilitates a better understanding in how infrastructures interact with hazards, and that is essential for infrastructure planning, management and fortification (Franchin, et al., 2015). Recently, a growing interest has been triggered with regards to modelling resilience in individual infrastructure sectors, each with different ways of quantifying resilience. A selection of related studies is introduced in table 2.5.

<b>Authors</b>	<b>Resilience Model of Urban Infrastructures</b>
Murray, 2006	Murray focused on transportation networks and proposed using four metrics to collaborative evaluate the resilience of transport networks. These four metrics are: Adaptability (e.g. vehicle switching to lanes not generally used for traffic), Safety (e.g. number of traffic incidents occur along a given road), Mobility (e.g. traffic capacity of a given road), and Recovery (e.g. amount of time required to alleviate congestion). This approach is related to integration, interpretation and comparison between heterogeneous indicators. Thus, it is considered to be methodological, and not effective to be implemented in real practice.
Berche, et al., 2009	Resilience of public transport networks (PTN) was analysed under different attack scenarios. PTNs were mapped as network model, and network connectivity was used to define random attack scenarios (e.g. remove specific nodes). Resilience is evaluated as a proxy of the network characteristics (e.g. mean shortest path length). It is an easy approach to be implemented, but authors only consider connectivity of network, but not the vulnerability of the actual physical component, which is essential in evaluating the performance of transport network under catastrophic events.

Freckleton, et al., 2012	A method was developed to assess and quantify resilience using fuzzy inference system (FIS). The authors developed a framework which introduced two concepts: the resilience cycle (Normalcy-Breakdown-Annealing-Recovery) and system performance (resilience). Resilience is collaboratively assessed by multiple fuzzy indicators, such as mobility index, personal transport cost index, goods and material access index, etc. They proposed a methodology which enables integration of heterogeneous components contributing to resilience. However, its implementation in real practice is computationally expensive due to great number of variables to be calculated.
Dorbritz, 2011	Dorbritz focused on modelling the resilience of large-scale rail transport networks, in different disaster scenarios. Nodes in transport network are removed from topological and operational perspective to simulate disasters. Resilience is measured by four dimensions: robustness (disaster withstand ability), resourcefulness (capacity to mobilize resource), redundancy (ability of alternative resource), and rapidity (capacity to contain loss in a timely manner). The approach is easily implemented in R packages. The main weakness of this approach is that resilience is only assessed based on topological characteristics, and there is no consideration of network dynamics.
Leu, et al., 2010	The authors proposed an approach for quantifying resilience of transport networks using network theory. Using GPS data, they modelled a network consisting of three interacting layers: the physical structure, the service function, and the cognitive properties (citizen's cognition). This approach can be further generalized to any ground transport system as long as GPS data is available. However, this approach does not apply agent-based models, and that means human behaviours are not realistically represented.
Davis, 2014	Davis understood the resilience of a water distribution system as its ability to provide post-earthquake services to emergency operations such as hospitals, emergency operation centres, and evacuation centres, so that no critical disruption of these emergency operations will occur. It is a novel approach which considers infrastructure resilience together with other critical components of the city. However, only service time lost is used as a metric to assess resilience of water distribution system, without the consideration of the damage on the actual physical system.
Mensah, et al., 2015	Authors proposed a framework for quantifying resilience of electric power grids. Electricity power grids are modelled as minimum spanning trees (MST). Resilience is assessed by the fraction of customers served or not served by electricity power after hurricane occurs. This approach is computationally cheap due to its simplicity. However, the topology of network (modelled as MST) might be oversimplified, without considering the redundancy design in power grids.
Cavallaro, et al., 2014	A hybrid social-physical network (HSPN) is proposed to assess infrastructure service resilience to seismic catastrophe within urban space. The network consists of two types of nodes: service nodes (schools, shops, energy distribution station, hospitals, etc.) and social nodes (residential buildings). Nodes are connected using the urban street network. When assessing resilience with HSPN, the probability of the HSPN being disrupted is acquired by assessing the fragility of service nodes representing infrastructure. This approach is easy to implement, although it over-simplifies how infrastructure service is connected from infrastructure asset to buildings.

**Table 2.5.** A selection of related research of infrastructure resilience models.

From table 2.5, customized approaches are applied in local infrastructure sectors to quantify infrastructure resilience. While it is important to understand how resilient an infrastructure sector is, it is equally important to understand how resilient multiple urban infrastructures are, when seen as an integrated system within city. This is where infrastructure dependencies and interdependencies play a key role.

#### ***2.4.2 Dependencies and Interdependencies***

Critical urban infrastructure sectors are not isolated, but instead highly connected (Rinaldi, et al., 2001). The connections between different urban infrastructures sectors, are termed “dependencies” and “interdependencies”. According to Rinaldi, et al (2001):

**A dependency** refers to “a linkage or connection between two infrastructure assets, by which the state of one infrastructure asset influences or is reliant on the state of the other”.

**An interdependency** refers to “bi-directional relationship infrastructure assets, in which the state of each asset influences or is reliant on the state of the other”.

As an example of dependency, water pumping station relies on electricity and thus is dependent on electricity substation (from electricity power network). As an example of interdependency, water treatment plant requires communication of its SCADA system (supervisory control and data acquisition) and in turn, it provides water for SCADA system to cool down.

It is considered that dependencies and interdependencies make critical urban infrastructures more vulnerable, as disruption can easily cascade from one infrastructure sectors to another (Ouyang, et al., 2014). For example, in August, 2005, the hurricane Katrina stroke southern Louisiana, USA. The supply of crude oil and refine petroleum products was interrupted due to loss of electricity power at three pumping stations at three major oil transmission lines. As a

result, 160 million litres per day of the gasoline production was lost, accounting for 10 percent of the US supply (O'Rourke, 2007).

Thus, modelling dependencies and interdependences between critical urban infrastructures has become a key research field (Min, et al., 2007). For the existing infrastructure dependencies and interdependencies models, it is considered that they can be broadly categorized into three major groups (Ouyang, et al., 2014), and they are summarized and explained in table 2.6.

Type	Description
Empirical	<p>Empirical approaches analyse dependencies and interdependencies according to historical accident, disaster data and expert experience. Study with this type of approach aims to identify frequent and significant failure patterns, to inform decision making and empirically based analysis (Laefer, et al., 2016). For example, McDaniel et al (2007) proposed a framework for characterizing infrastructure failure interdependencies (IFI). Data of three kinds of events were used (2003 North America Blackout, 1998 Quebec Ice Storm, and 2004 Florida Hurricanes). IFIs are characterized by the sectors affected, and consequences for society. IFIs in different events were compared, which in the end serves as a basis for considering priorities of risk mitigation. Clearly the empirical approach is very subject to the data availability. That means this approach is not feasible if no hazard or infrastructure failure data is available for an area. Also, this approach is more at the system-level, without understanding interdependencies at component-level (Guikema, 2009).</p>
Agent Based	<p>Agent based approaches aims to understand interdependent infrastructures as CAS - complex adaptive system (in which a perfect understanding of individual parts does not covert to perfect understanding of the system behaviour) (Amin, 2000). This approach assumes that complex system behaviours emerge from many individual relatively simple interactions of autonomous agents. Most components of critical infrastructures can be viewed as agents (Ouyang, 2014). Using this approach, Idaho laboratory (Dudenhoeffer, et al., 2006) developed the agent-based CIMS (critical infrastructure model system), to simulate and visualize cascading effects within different infrastructure sectors (energy, telecom, transport, water). Agent-based approach is flexible with other modelling techniques to provide more comprehensive analysis. However, its main drawback is that quality of simulation highly depends on modeller's assumption of agent behaviour and it is difficult to justify theoretically (Ouyang, 2014).</p>
Network Based	<p>This approach applies network theory to model the interdependent infrastructures as <i>Networks of Networks</i> (D'Agostino, et al., 2014). That is to say, any single infrastructure sector is modelled as network model, with nodes and edges. An interdependency is modelled as inter-edge connecting two nodes from two network models. Network based</p>



---

approach aims to understand performance response of interdependent infrastructures under different hazards. Many metrics are used to assess the performance of each network, such as number of failed components, connectivity loss, and cluster related metrics, (Osorio, et al., 2007). Flow can also be introduced in this approach to account for service and flow delivered by critical infrastructures, such as the model developed by Wallace et al (2001). Their model enables mathematical representation of interdependencies and allows users to assess post-disruption impact and restoration process. Generally, the network-based approach can identify critical infrastructure components, providing more realistic description on operation mechanism of infrastructure. However, it can be very computationally expensive, if operation mechanisms are modelled in detail (Ibanez, et al., 2011).

---

**Table 2.6.** Major types of approaches of modelling interdependent infrastructures.

From table 2.6, it is found that different approaches have their own advantages and drawbacks. Depending on the actual modelling requirement, data availability, and computation capability, an appropriate one can be chosen for specific problems.

## **2.5 Geospatial Urban Infrastructure Models**

Critical urban infrastructures, as seen earlier, are of grave importance to modern cities and are attracting increasing attention with regards to its resilience and interdependencies. However, it is not enough to regard critical urban infrastructure as “self-contained” systems, without considering its spatial relationships with city (Shepard, 2011). They are “embedded into” the spatial domain of the city and therefore spatially interact with the city. For example, at infrastructure planning or fortification stage, decision must be made to use urban space efficiently while causing minimum disruption on the existing urban facilities (Short, et al., 2005). Another example is related to hazards and infrastructure failure. When natural hazards (such as floods) occur, they can cause failure on certain infrastructure system (such as electricity power supply). As a result, a number of consumers (such as individual buildings) will be disrupted, and spatially an infrastructure disruption area is generated (Deshmukh, et al., 2011).

Therefore, it is imperative to develop geospatial modelling platform, by which crucial

information of urban infrastructure systems can be collectively gathered, analysed, and published to those, who need such data for their specific applications (Coutinho-Rodrigues, et al., 2011; Su, et al., 2011; Zygiaris, et al., 2013). As such, geospatial urban models started to emerge in the recent years (Hall, et al., 2016). The geospatial approach allows for spatial data exchange and interoperability, further analytical, simulation and visualization purposes (Rautenbach, et al., 2013). A number of large research initiatives have looked to develop a suite of infrastructure analysis and modelling tools where geospatial data and location of infrastructure assets and network play a key role (Barr, et al., 2013), such as the US National Council on sustainable critical infrastructure systems (National Research Council, 2014), the Dutch programmes on next generation infrastructure and knowledge for climate (Dutch Ministry of Infrastructure and Environment, 2014), Australia critical infrastructure protection and modelling analysis programme (National Security and Resilience Policy Division, 2009) and the UK Infrastructure Transitions Research Consortium (Barr, et al., 2013). Within such initiatives, it was recognised that it is a key requirement for a geospatial urban infrastructure modelling platform to have the ability to collect, integrate and manage a wide range of different infrastructure data at geospatial perspective. However, three major challenges exist in this field, and are discussed in the following sub-sections.

### ***2.5.1 Geospatial Infrastructure Ontology Development***

Geospatial infrastructure data can come from diverse sources, because different infrastructure systems are generally owned and managed by different departments, such as utility companies, and governments. This means data (from multiple sources) can have poor interoperability, because infrastructure data of one company can differ from that of another company, not in what is encoded but also how it is encoded in their data platforms (Fu, et al., 2008). For example, with regards to road engineering, some governments use the term ‘median strip’ and others use the term ‘central reservation’. They are talking about the same thing, which means ‘the reserved area on the road used to separate opposing traffic’. When there are more and more data sources, the problem of data interoperability can become more

apparent. That makes it very difficult to exchange and integrate information from different data platforms.

A typical solution is to have a ‘common language’ which has a carefully designed vocabulary and detailed meaning of each word is given. This is very much like the situation when people speaking different native languages can still communicate with each other, because everybody also knows a common language, such as English.

This ‘common language’ in this sense is the ontology. According to Gruber, an ontology is “an explicit specification of conceptualization” (Gruber, 1993). It is a knowledge of a specific domain, about what entities exist in that domain and their relationships with each other. It is a data model, but in a high level and in a more generalized way, which aims to capture the most important information within a domain. A well-designed ontology should explicitly define semantics on the entities and their relationships. In this way, the ontology serves as a common language to both relate and distinguish entities between different data platforms, and thus supports knowledge and information exchange (Katsumi, et al., 2018).

Ontology is domain-specific, that is very much related to what we want to do with the data, and what information we need. When developing a geospatial urban infrastructure modelling platform, it is considered the topological connectivity, spatial information and attributes are the most vital information that must be included (Barr, et al., 2016). The topological connectivity allows us to model the complex urban infrastructure system using network model. The spatial information allows us to perform necessary spatial query on the urban infrastructure. The attributes (for example, the capacity of an electricity substation, or the resistance of an electricity cable, etc.) allows us to run basic simulation on the urban infrastructures.

When dealing with urban infrastructure data in the geospatial perspective, the spatial scale or the ‘granularity’ of the data is something that must be considered, and this is essential in developing an ontology. For example, at electric engineer’s perspective, an electricity

substation (in the distribution level) generally consists of switches, protection and control equipment and transformers (Larkevi, et al., 1995). However, when developing ontology for modelling geospatial infrastructure networks, it is unnecessary to further break an electricity substation into these four parts. Instead, it is more appropriate to treat an electricity substation simply as an electricity asset, which is enough to model the electricity infrastructure as geospatial network instances (Barr, et al., 2013).

Research on developing ontologies of urban infrastructure has attracted increasing attention in the recent years (Howell, et al., 2018). Industrial and academic experts have proposed many common infrastructure / urban data models, such as Utility Content Data Standards (Facilities Working Group, 2000), Utility and Pipeline Data Model (ESRI, 2015), IFC Utility Model (Liebich, et al., 2012), Utility Network ADE (Becker, et al., 2012), INSPIRE data specification on utility and transport network (INSPIRE, 2013), Towntology (Berdier, 2007), KM4City Model (Bellini, et al., 2014), Utility Knowledge Ontology (Xu, et al., 2018), OTN (Lorenz, et al., 2005), and iCity Ontology (Katsumi, et al., 2017). A comparison of these data models is shown in table 2.7.

<b>Name</b>	<b>Description</b>
UCDS	A utility data standard proposed by the US government, to support large-scale, intra-city applications such as engineering and life cycle maintenance of utility systems. It covers major utility infrastructure such as electricity, water supply, waste water, gas. However, it is rather shallow in representing semantic relationships, also it does not mention topology. It only focuses on utility and no transport infrastructure.
UPDM	A geodatabase data model template developed by ESRI, for operators of pipe networks in the gas and hazardous liquids industries. It represents spatial information and topological connectivity. However, it is shallow in representing attributes. It is rather a specific data schema for geodatabase, rather than a generalized data model. Also, it does not include transport infrastructure.
IFC Utility	A data model compatible to the IFC building model. It focuses on representing utility networks within buildings, which means the ‘granularity’ is too fine for us. That also means this model does not care about transport infrastructure.
Utility Network ADE	A network extension for the CityGML, which is a 3D city data model. The utility network ADE model allows to represent 3D utility components and their topological connectivity. However, it is shallow in representing attributes. It is more like a specific data format, rather than a more generalized knowledge.

INSPIRE	Proposed by European Commission, in the INSPIRE Knowledge Base project. It is a high level and generalized model which covers both utility and transport infrastructures. Topological connectivity and spatial information are represented, although the model is not rich in representing semantic relationships and attributes.
Towntology	An ontology developed by two French laboratories to clarify and organise terminology used by French urban planners. It focuses on urban road network and urban mobility. The ontology is rich in representing semantics about component of road network, but it is only at a geospatial perspective. There is no inclusion of topology or attributes, and the Towntology does not deal with utility infrastructure.
KM4City	“Knowledge Model for city” is an ontology developed for smart city, which covers domains of weather, sensors, services, transport, event, locations, etc. It does mention the transport, but it focuses on the mobility rather than the transport infrastructure. There is also no inclusion of utility infrastructure.
Utility Knowledge Ontology	It is an ontology approach for utility knowledge exchange representation. A high-level data model for utility networks. It is rich in defining utility entities, their semantic relationships. Spatial information is also included. Although it is rather shallow in representing the topological connectivity and attributes. Transport infrastructures are not considered.
OTN	Ontology for transport network, as part of the Reasoning on the Web with Rules and Semantics (REWERSE) project. A high-level data model for transport network, with rich representation of connectivity and semantic relationships. However, it is not rich in attributes and spatial information, and there is no inclusion of utility infrastructure.
iCity	iCity is an ontology under development as a part of urban system. It focuses on the transport system, and is rich in representing its entities, semantic relationships and topological connectivity. It is also rich in representing dynamic transport flow. Topological connectivity is also included. However, iCity is not rich spatial information (relationships) and attributes, and does not model the utility infrastructure.

**Table 2.7.** Comparison of related ontologies and models with regards to urban infrastructures.

It can be seen that, it is difficult for a data model/ontology to both include utility and transport infrastructures (except for INSPIRE), and it is also difficult for a data model/ontology to be rich in topological connectivity, attributes, and spatial information. Moreover, at city scale, the buildings are regarded as consumers of infrastructure services and material, and it is crucial to know how buildings are connected to infrastructure networks. Therefore, building is an indispensable part of an urban infrastructure ontology (at this scale). However, there is no such ontology developed in this context until now. Finally, as introduced earlier, different infrastructure systems have dependencies and interdependencies, and this is something that must be taken into account. There do exist ontologies that represent infrastructure dependencies and interdependencies (McNally, et al., 2007; Sicilia, et al., 2009), but they only

focus on dependencies / interdependencies themselves, without integrating the actual representations of urban infrastructure systems. All of these call for the development of an integrated urban infrastructure ontology, that is rich in topological connectivity, spatial information, attributes, and it should present dependencies, interdependencies and the relationships between infrastructures and buildings.

### ***2.5.2 Geospatial Infrastructure Data Inference***

A well-designed ontology is essential when collecting and integrating infrastructure data from multiple data sources. However, it is under the assumption that data is present and collectable, which in many cases is not true. Companies and governments which own and manage the geospatial infrastructure data, often forbid public uses of their data due to confidential issues (Bon, 2017). It is also possible that some of them even do not have their data in the geospatial format (Fu, et al., 2008). Thus, there is an urgent need for approaches that can infer, at very fine spatial scales, plausible infrastructure networks from infrastructure assets to the buildings they service.

Heuristically generating spatial network data is a complex problem, as spatial constraint is normally needed to indicate at which location spatial network should be (or should not be) generated (Heijnen, et al., 2014). A common spatial constraint is the space syntax (as introduced in section 2.3). For example, using measures (e.g. integration value, accessibility) from space syntax, it is possible to predict traffic flows and possible congestions on the urban road network (Duan, et al., 2008), and identify possible locations for constructing new roads on the existing road network to accommodate increasing traffic demand (Zheng, et al., 2009). However, the space-syntax based approach is more like a *network expanding* approach, rather than a *network generation* approach, which is not useful if network layout is completed unknown (this is the worst case in real scenarios, but it is possible). Therefore, observations are made on the generative methods for network data inference, and two most common methods are agent-based models and fractal geometry models.

Agent-based model (ABM) has been introduced in section 2.1, table 2.3 (in the *Behaviour Model*), but it is also an efficient tool to generate (design) infrastructure system layout. An agent is mobile and can interact with the external environment, it makes its own decision to achieve a required aim. For example, when designing sewer network layout, the agents defined by Ulrich et al (2010) operate on different landscape maps (e.g. digital terrain model, land use map), and agents prefer to move to lower positions or places close to rivers. The trajectories of the agents (act as sewer planners) can suggest plausible layout of sewer pipes. Likewise, Adamatzky et al (2016) employed ABM to simulate the evolution of French motorway network, and this is done by defining agents as small bugs for transporting food among different French major cities. ABM method simulates the way the human beings design network, which is its main strength. However, a major issue is that generation result highly depends on modeller's assumption on agent's behaviour, and such assumption is difficult to be justified theoretically (Ouyang, 2014) (i.e. setting up and tuning model parameters can be difficult, and there can be a risk of over-fitting).

Fractal geometry methods employ the concept of fractals, which are geometric shapes that are self-similar over a wide range of scales (Ghosh, et al., 2006). Fractal tree is a class of fractal that can be used to represent the dendritic geometry structure of urban infrastructure networks (Möderl, et al., 2009). Fractal tree-based method has been applied in designing or generating infrastructure network layout for different sectors such as sewer (Jeffers, 2017), water supply (Möderl, et al., 2011), and electricity distribution (Barakou, et al., 2015). However, this approach suffers from the similar issue as the ABM. Spatial resolution (more precisely speaking, the Strahler degree) must be manually tuned to control how many branches should exist in the synthetic network (Jeffers, 2017). Another problem is that, fractal tree methods lead to non-loop network structure, and thus cannot generate redundant network structure (Mensah, et al., 2015).

As is seen here, space syntax, ABM, and fractal geometry methods still have their limitations for generating infrastructure network layout. Besides they often ignore the spatial urban configuration (e.g. land use, building location, road layout, etc.). As pointed by Bon (2017)

and Cavarallo et al (2014), the layout of infrastructure network should be related to the building and streets. Moreover, infrastructure network layout should be related to building layout types (detached buildings, terraces, etc.), as this is supposed to affect how the infrastructure network should be constructed (Larkevi, 1995). Another thing unclear is the scalability and generalization of the network generation algorithm. A good algorithm should have a high level of generalization and does not over-fit to the area or city where it is developed and applied (Mao, et al., 2013). However, for all the algorithms introduced so far, each of them only focuses on a specific area, without considerations of scalability or generalization.

### ***2.5.3 Database System Implementation***

Once good quality geospatial data are collected or generated, the next step is to find a appropriate database system to accommodate them. Urban infrastructure network data have complex topology, attributes and geometry (Barr, et al., 2016). An efficient data platform is essential for managing such complex network data. In many countries, individual operators in specific infrastructure sectors (Woodhouse, 2014), as well as several large research initiatives (Barr, et al., 2016), have realised the importance of developing their data and information management platforms for better infrastructure planning and decision support.

At its core, such platforms require appropriate database systems that can handle the wide range of disparate data and relationships required for infrastructure systems modelling and analysis (Barr et al, 2016). Traditionally a spatial relational approach is used, such as the Oracle Spatial Network Extension (British Telecom, 2012; Fikjez and Řezanina, 2016) or specifically developed schema for representing dependence/interdependence between infrastructure networks (e.g., the NISMOD-DB approach developed by the Infrastructure Transitions Research Consortium (Barr et al, 2013)).

The spatial relational approach is naturally strong in dealing with queries involving the



attributes matching (such as finding all the assets with specific attribute values), and spatial calculations (such as finding all assets within a certain distance). However, it is somewhat limited in analysing large complex network topologies, such as intra-city scale electricity distribution networks (Ji, et al., 2018).

Recently, NoSQL graph database have been proposed as a general approach for the more efficient storage and retrieval of network data (Have, et al., 2013). The most popular graph database, Neo4j has been proven for more efficient management of network data, including social network (Cattuto, et al., 2013), biology network (Yoon, et al., 2017), and knowledge graph (Lin, et al., 2017). However, there is no related research in applying NoSQL graph database (such as Neo4j) in the management geospatial urban infrastructure networks, and evaluating the performance, which is considered as a research gap, and an interesting topic to explore.

## **2.6 Summary**

In this chapter, a review was done on the recent fast urbanizations of the modern cities and the different challenges they are facing. The city is a complex system, which has different components interacting with each other. Critical urban infrastructure is an indispensable component of the city and has great impact on the functioning of the city. Little malfunction and disruption on the urban infrastructures can end up into severe urban disaster, which is why increasing attention is being attract to understand the resilience and interdependencies of critical infrastructures. Accessing and managing highly granularity geospatial data on infrastructure network, is essential for different urban applications as well as infrastructure planning, modelling, simulation and fortification. Accordingly, there are still several research gaps that need to be filled in:

A geospatial ontology must be developed with regards to the geospatial urban infrastructure data. The ontology must be rich in topological connectivity, spatial

information, and attributes. It should also include dependencies and interdependencies and the relationships between infrastructure and buildings.

A generic heuristic algorithm must be developed in order to infer spatial layout of the urban infrastructure network, when accessing actual data is not possible. The algorithm should be responsible for generating plausible synthetic network layout spatially, and be scalable (regardless of city size).

An appropriate database system must be developed to accommodate the complex geospatial urban infrastructure network data. Application of NoSQL graph database (compared with traditional RDMS) must be further explored. In this context, database benchmarking tests should be designed test database performance in different scenarios (different network data, different queries, etc.). Finally, based on benchmarking test, a most appropriate database architecture will be chosen as the generic database solution.

## **Chapter 3. An Ontology for Modelling Urban Infrastructure Networks**

### **3.1 Introduction**

In the last chapter, current challenges with regards to accessing and managing fine granularity geospatial infrastructure network data are identified, and one of them is the data heterogeneity issue. To address this issue, an ontology must be developed, which serves as a “common language” to allow data integration from different data sources, platforms or databases (Gruber, et al., 1993). This chapter aims to develop an infrastructure network ontology which fits this purpose.

To develop an ontology, as suggested by Uschold and King (1995), and Uschold and Gruninger (1996), the key step is to identify the purpose and scope of this ontology (For what kind of applications it is used? What information to include and what not?). After that, it will be much clearer about what knowledge (entities, relationships) should be represented and how to represent them.

Modern city consists of heterogeneous critical infrastructure networks (utility, transport) and the buildings they serve. Understanding the spatial connectivity between the infrastructure assets and buildings is the key to analyse and model flows within city (Ji, 2019). This is identified as the purpose of this ontology, which basically outlines the necessary knowledge to be represented. First, “connectivity” must be represented, and preferably in the perspective of network theory. That means entities like “edges” and “nodes”, as well as the relationships like “connect” need to be defined. Secondly, spatial information and relationships are indispensable, and relationships such as “this cable is above that pipe” or “the substation is 10 meters away from that building” should be represented. Finally, necessary attributes must be defined to model and characterize flows within infrastructure networks, such as “diameter of a water supply pipe” and “number of lanes of a road”.

The other thing to consider is the scope of this ontology. First, it is clear this should be an

integrated ontology to include all critical infrastructures, namely the utility networks (electricity, gas, water, and waste water) as well as the transport networks (road, rail, and metro). Secondly, buildings are considered as consumers of infrastructure services within cities (Cavallaro, et al., 2014), so they should be also properly represented. Finally, since the ontology aims to represent different infrastructure sectors, it is natural and vital to introduce knowledge of dependencies and interdependencies.

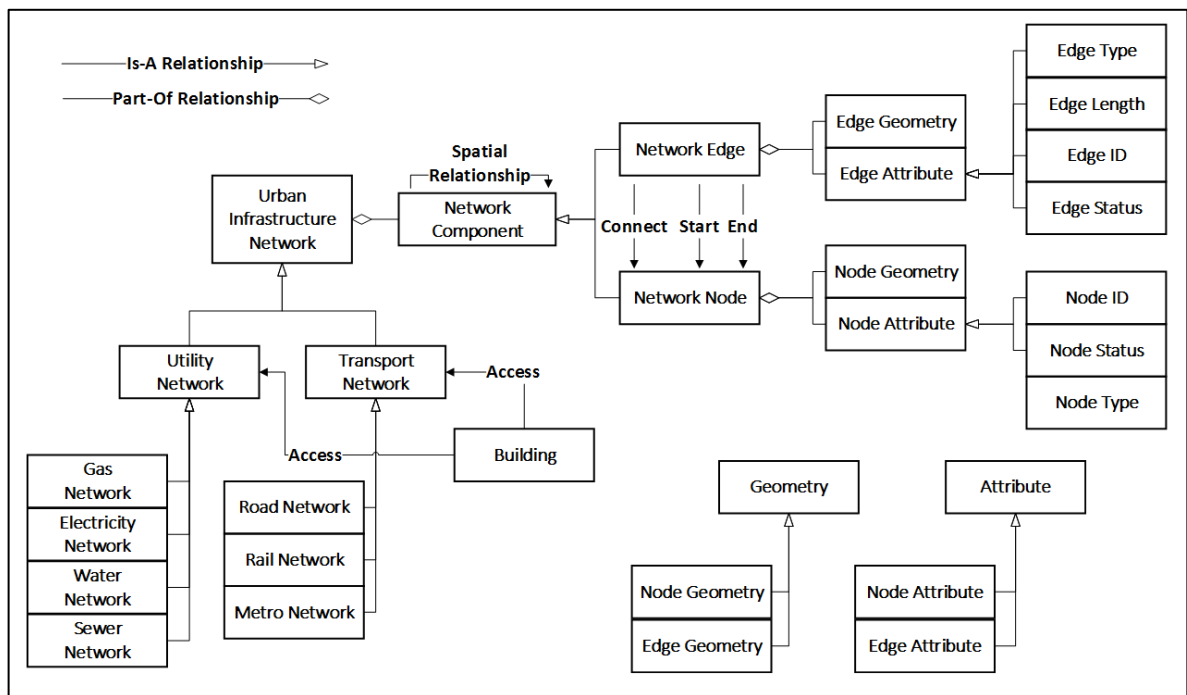
While it is possible to develop an ontology from scratch, it is advised that ontology developers should try to re-use common knowledge (entities, relationships) from existing ontologies, or common data models if possible (Hendler, et al., 2001; Leung, et al., 2013; Lau, et al., 2016). This helps to develop the ontology in a more generalized way and allows easier data integration. In Chapter 2, 10 related ontologies/data models with regards to the urban infrastructure networks (Ji, 2019) were reviewed, and a comparison was made on them. Based on the comparison, it is argued that INSPIRE data model (INSPIRE, 2013), Ontology of Transport Network (Lorenz, et al., 2005), and the Utility Knowledge Ontology (Xu, et al., 2018) are the three most relevant contributions.

The INSPIRE data model is the only one model that covers knowledge of both utility and transport networks. Any other model focuses on either one of them. The INSPIRE data model is also rich in representing topological connectivity (using nodes and edges) and in representing the geometry of the objects. It also contains some attributes, but since it is not an ontology, it is very shallow in semantics. The OTN (Ontology of Transport Network) is rich in topological connectivity, and semantic relationships, but lacks enough support for attributes (for example, number of lanes on road, etc.) and spatial knowledge. The Utility Knowledge Ontology is rich in defining components within utility infrastructures, the semantic relationships (including spatial relationships). But it does not mention topological connectivity, and lacks enough attributes support. Despite the relevance of these three models, none of them represents the relationships between buildings and infrastructure networks (INSPIRE does mention building, but only focuses on geometry) and dependencies or interdependencies. Therefore, this is considered to be the biggest research gap currently and it

could be a potential contribution from the development of this ontology.

For the layout of the remaining part of this chapter: Section 3.2 introduces top-level ontology; Section 3.3, 3.4, 3.5 introduces ontology of utility network, transport network and building; Section 3.6 discusses about dependency; Section 3.7 formally represents the ontology; Section 3.8 concludes the chapter.

### 3.2 Ontology Construction



**Figure 3.1.** Top-level entities and relationships in the ontology.

The top-level entities and relationships of this ontology are shown in figure 3.1. As in any other ontology (Xu, et al., 2018), there are two most important semantic relationships, the “Is-A Relationship” and “Part-of Relationship”. These two relationships help to allow class inheritance and represent a real-world knowledge if an object consists of several parts. Examples of these two relationships are given in table 3.1.

Relationship	Examples
Is-A	1. A Utility Network is an Urban Infrastructure Network.

	2. An Electricity Network is a Utility Network.
	3. Network Edge or Network Node is a Network Component.
Part-Of	1. Network Component is part of an Urban Infrastructure Network.
	2. Edge Geometry and Edge Attribute is part of a Network Edge.

**Table 3.1.** Examples of “Is-A” and “Part-of” relationships.

Entities “Network Edge” and “Network Node”, and the “Connect” relationship between them are introduced. By doing this, each type of infrastructure network (road, electricity, etc.) can be easily represented by a network model mathematically. This is a common approach to represent knowledge of topological connectivity between infrastructure components (INSPIRE, 2013). It also makes it easy and straightforward to formally represent the urban infrastructure networks using mathematical notations (section 3.6).

It is argued that geometry should be associated with Network Edge or Network Node. These geometry entities are called Edge Geometry and Node Geometry, respectively. They are subclass of Geometry (a generic geometry object). The ontology is developed at a high-level generalisation and therefore the Edge Geometry and Node Geometry will be defined as simple as possible (INSPIRE, 2013; Lorenz, 2005). The definition of these two entities are given in table 3.2.

Entity	Definition
Node Geometry	A point in the 3-dimensional space, represented by its x, y, z coordinates to indicate its location. A coordinate system must be given such as the British National Grid for the UK.
Edge Geometry	A polyline in the 3-dimensional space, which is represented by a sequence of points.

**Table 3.2.** Definition of geometry entities.

Depending on the actual application, Node Geometry and Edge Geometry can be simplified into 2-dimensional point and polyline (without z coordinate), if it is difficult to access 3-dimensional data. But then some semantics in spatial relationships will be lost (such “Above” relationship), please see below (table 3.3).

It is argued that any Network Component (whether it is Network Edge or Network Node) should have “Spatial Relationship” with one or more Network Components, since any Network Component has geometry in a coordinate system. “Spatial Relationship” is considered to be an abstract relationship, and can be specified depending on the actual application where the ontology is used. For example, according to Borrmann et al (2009), common spatial relationships can be divided into three categories: topological, metric, and directional. The semantic examples are given in table 3.3, and definitions of these relationships are according to W3C Geospatial Ontologies (W3C, 2007).

Category	Spatial Relationship	Semantic Examples
Topological	Touch	A touches B.
	Disjoint	A and B are disjoint.
Metric	Distance	Distance between A and B is 100 meters.
	Closer / Nearer	C is closer to B than to A.
Directional	Above / Below	A is above B.
	North/South/East/West Of	A is north of B.

**Table 3.3.** Common spatial relationships and semantic examples.

Note that “Spatial Relationship” applies to any Network Components, whether or not they belong to a same type of infrastructure network. For example, a Network Edge from Electricity Network can have “Spatial Relationship” with a Network Edge from a Road Network. This is how to ensure the ontology is rich in representing spatial knowledge.

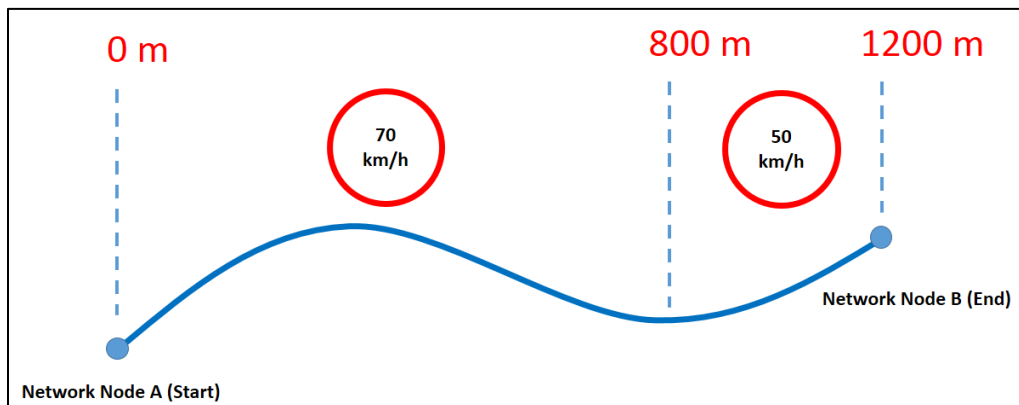
In the ontology, attributes are associated with Network Edge and Network Node separately, they are Edge Attribute and Node Attribute. They are the subclass of Attribute. It is considered the most common Attributes (that are sharable among different types of infrastructure networks) are below (INSPIRE, 2013; Xu et al., 2018), given in table 3.4.

Attribute	Description
Edge Length	Numeric value to represent length of the Network Edge.
Flow Direction	Ordered pair of Nodes to indicate the flow direction on a Network Edge.
Edge Status / Node Status	A text to show whether the Network Edge / Network Node is “In Use”, “Out of Service” or “In Maintenance”.
Edge ID / Node ID	A unique number or text, which serves as an identifier of a Network Edge or Network Node.
Edge Type/ Node Type	A text showing the type of a Network Edge or Network Node. For example, for a Network Node in Electricity Network, its Node Type can be ‘Substation’.

**Table 3.4.** Description of common attributes.

By default, an Attribute is represented by a static value, and that means its value is fixed for a given Network Component it is associated with. However, attributes can be spatially and temporally transient on the infrastructure networks (Min, et al., 2011). In this situation, an Attribute can be represented by an abstract function.

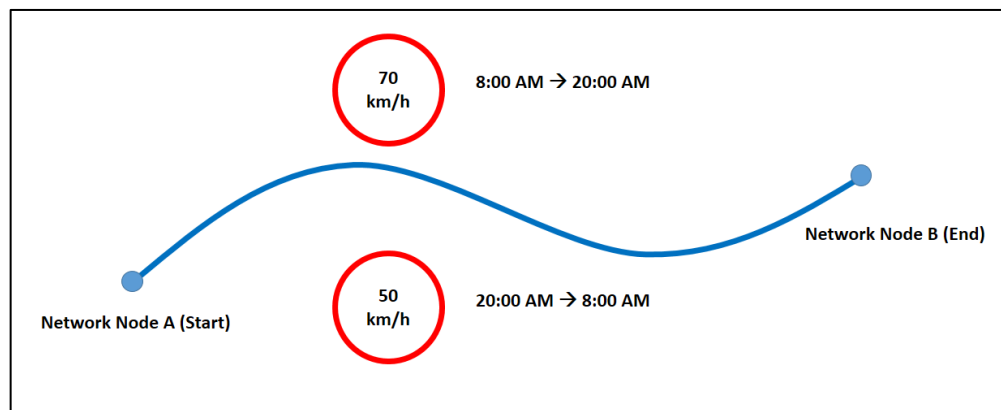
To represent a spatial transient attribute, a typical solution is to define a linear reference on the Network Edge (INSPIRE, 2013). This is represented by a sequence of Network Node pair. That is why two more relationships “Start” and “End” are added in figure 3.1. An example is given in figure 3.2, to show how to represent spatially transient speed limit on a road. For instance, it is plausible to say speed limit on this road (Network Edge) is 70 km/h from 0 m to 800 m and it is 50 km/h from 800 m to 1200 m, and it is based on the sequence of Network Node pair (A, B). At a more generalised level, it can be represented as a function  $v = f(x)$ , to map the location  $x$  (along the Network Edge) to a value  $v$  of that attribute.



**Figure 3.2.** Use linear reference to represent spatially transient speed limit on a road.



For a temporal transient attribute, time reference is needed. The example of speed limit is still used here. In this situation (figure 3.3), the speed limit does not change spatially, but it depends on the time. It is plausible to say, the speed limit is 70 km/h from 8:00 to 20:00, and is 50 km/h from 20:00 to 8:00. At a more generalized level, it can be represented as a function  $v = f(t)$ , to map a time  $t$  to a value  $v$  of that attribute. Figure 3.2 and figure 3.3 together show the flexibility of our ontology to represent any attribute whenever necessary, if it is not static.



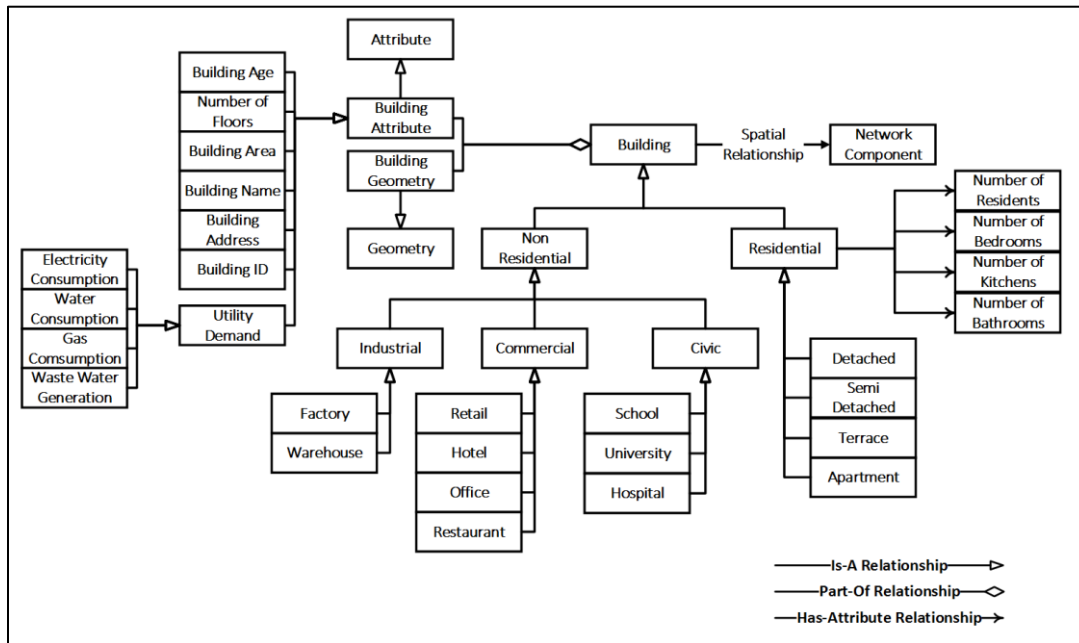
**Figure 3.3.** Use time reference to represent temporal transient attribute.

Finally, entity Building is briefly introduced here, with its relationship “Access” with Utility Network and Transport Network. This is a still high-level relationship to indicate Building needs infrastructure service. Specifically, that means *connection* between Building and these types of networks. This will be covered into details in section 3.4 and 3.5. Before that, a good definition of Building is needed.

### 3.3 Building

The entities and relationships with regards to Building are defined in figure 3.4. The ontology mostly reused knowledge from Urban Building Ontology (Zhu, et al., 2015). First of all, a Building has its own geometry called Building Geometry, which is represented by a 3D body object (Zlatanova, 2000), a very simple and common 3D GIS data model, to indicate the space a building actually occupies, as suggested by Zhu et al (2005) and Katsumi (2017). However, if accessing 3D data is not possible, or if application only cares about 2 dimensions, then Building Geometry can be simplified and represented by a 2D polygon (footprint). Note Buildings Geometry also allows the Building to have a spatial relationship with any Network

Component. For example, we can say “this Building is 5 meters away from that Road”.



**Figure 3.4.** Entities and relationships for Building.

Building have some attributes that can be inherited by any subclass of Building. Note that Utility Demand is explicitly defined for any Building, in order to quantify utility service demand at building level. Other attributes either provide basic information of the building (Zhu, et al., 2015), such as Building Address, Building Name, or allow us to model utility service demand (Swan, et al., 2009), such as Number of Floors or Building Area. These attributes are explained in table 3.5.

Attribute	Description
Building ID	Unique number as an identifier of the Building.
Building Name	Text as the name of the Building.
Building Address	Text as the address of the Building.
Building Age	Age of the Building. Older building can be less energy conservative and thus demand more utility service.
Building Area	Number as the area of footprint of the Building. Larger building can demand more utility service.
Number of Floors	Number to show how many floors in the Building. Building with more floors requires higher utility demand.

Electricity Consumption	Average daily electricity consumption, number in J.
Water Consumption	Average daily water consumption, number in m <sup>3</sup> .
Gas Consumption	Average daily gas consumption, number in m <sup>3</sup> .
Waste Water Generation	Average daily waste water generation, number in m <sup>3</sup> .

**Table 3.5.** Attributes that can be inherited by subclass of Building.

Buildings are first classified into Residential and Non Residential, and this is meaningful. Because residential buildings account for a large proportion of urban buildings (above 90% in the UK) and therefore a large proportion of utility service consumption (Pregolato, et al., 2018). They are more important with regards to localized utility demand model (Kavigic, et al., 2010).

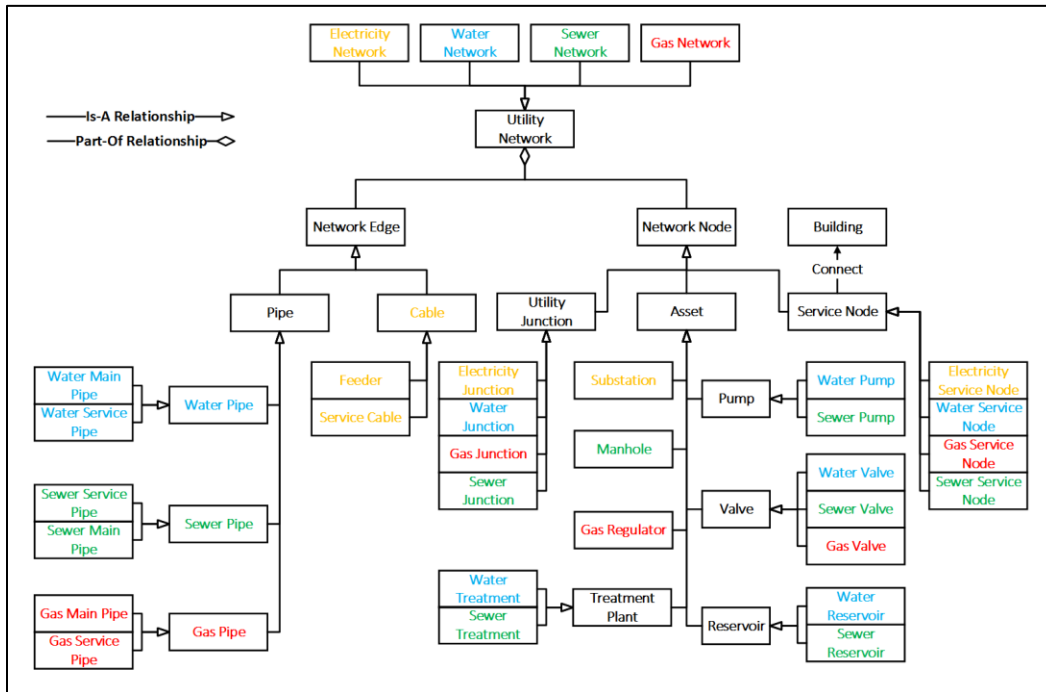
Residential can be further classified into Detached, Semi Detached, Terrace, and Apartment. This also helps to model utility demand. For example, a Semi Detached shares a wall with another building, so it can be more energy conservative than a Detached and therefore has lower utility demand (Nouvel, et al., 2015). Further, Residential has additional attributes, namely Number of Residents, Number of Bedrooms, Number of Kitchens, and Number of Bathrooms. These also help to model utility demand. For instance, more residents in a Building corresponds to higher utility demand (Blokker, et al., 2009).

Non Residential is further classified based on functionalities, because different types of buildings have utility demand at different level (Nouvel, et al., 2015). For instance, a factory normally has a higher electricity consumption than a residential building (Yu, et al., 2010). Another example is that, the electricity supply disruption to a hospital is considered to be more fatal than to a residential building (Murray, et al., 2007).

### 3.4 Utility Network

Entities with regards to the Utility Network are identified and defined in figure 3.5. By convention a utility network consists of transmission and distribution level. But the purpose of

the ontology is to understand connectivity between infrastructure networks and buildings. Therefore, the focus here is the *distribution* level. Entities like “Electricity Generator” in the Electricity Network is not included, because that belongs to the electricity transmission network.



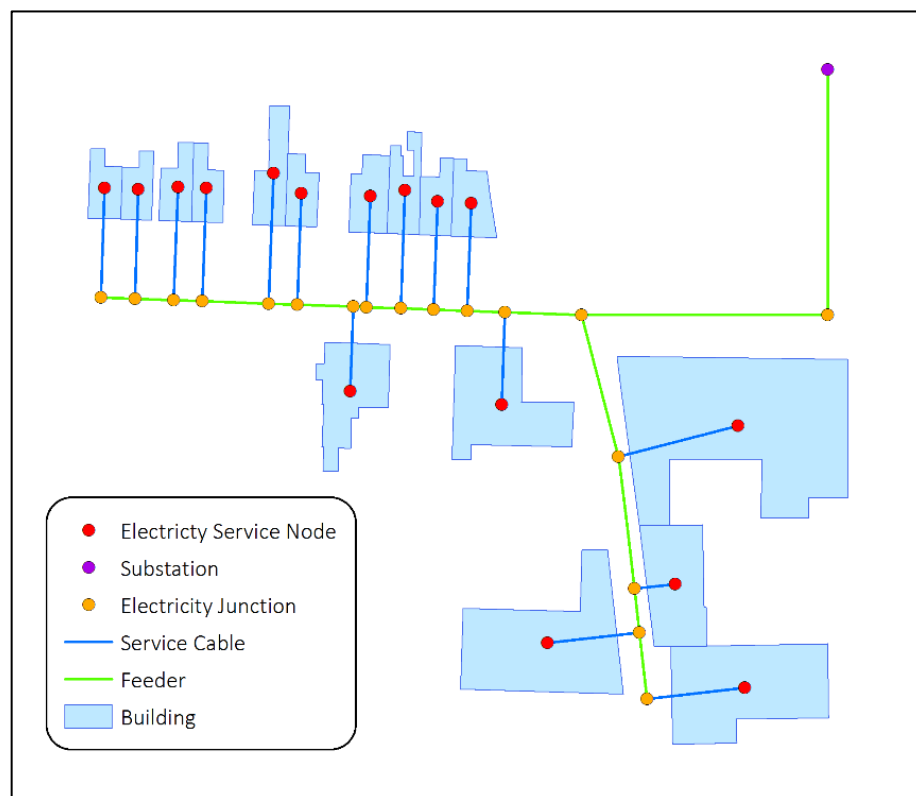
**Figure 3.5.** Entities of Utility Network.

Most of entities are Utility Assets, which are reused from INSPIRE data model (INSPIRE, 2013), Utility Knowledge Ontology (Xu, et al., 2018) and common utility distribution network models for electricity (Tanyimboh, et al., 2011), water (Avi, 2014), sewer (Vickridge, 2004), and gas (Osiadacz, 1987). Some of them are the sources where utility service enters the utility network at distribution level, such as Substation in the Electricity Network, Gas Regulator in the Gas Network or Water Treatment in the Water Network. Some of them are control elements in the network, such as Pump or Valve.

In ontology, an Asset can be represented as a Network Node, and Pipe or Cable connecting Assets is a Network Edge, as suggested in common geospatial utility network models (Tanyimboh, et al., 2011). Pipe or Cable can have subclasses. For example, a Cable can be

‘Feeder’ or ‘Service Cable’, and that is because different types of Cables have different connectivity relationships in the Electricity Network (details in figure 3.7).

Note that subclass of Network Node called Utility Junction is introduced, which refers to the location where Cables or Pipes connect with each other. This is used to ensure valid topology (every edge is connected to two nodes) when using a network model to represent a Utility Network. As a major innovation of my work, another subclass of Network Node called Utility Service Node is introduced, and it is connected to a Building. This is exactly how a Building “access” a Utility Network and is “connected” to it.

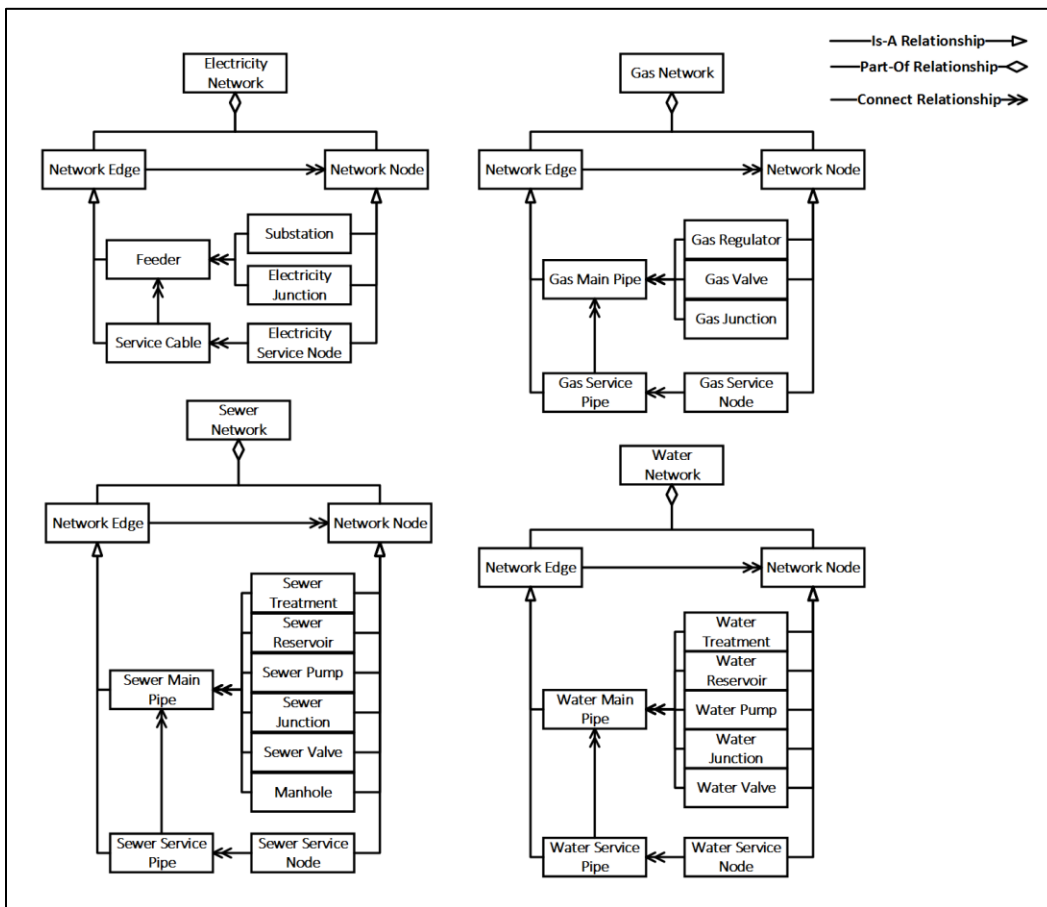


**Figure 3.6.** Example of an Electricity Network and Buildings.

In reality utility service is delivered to individual building via an “entry point”, which is commonly a meter (Osiaacz, et al., 1987; Avi, et al., 2014). That meter depends on the actually type of utility network, for example in electricity network it is an electricity meter. Therefore, in our ontology, Utility Service Node is used to denote that “entry point”. To be clear, the connection between a Utility Service Node and a Building is actually a *mapping*

(each Utility Service Node corresponds to a Building), and will be formally represented in section 3.7. An example is given in figure 3.6, showing the Network Edges and Network Nodes exist in an Electricity Network in a two dimensional space. Buildings are displayed via their footprints. Note in actual applications (chapter 4, 5, 6, and 7), Utility Service Node is simply called *Building Node* in a specific type of Utility Network, if no confusion is caused.

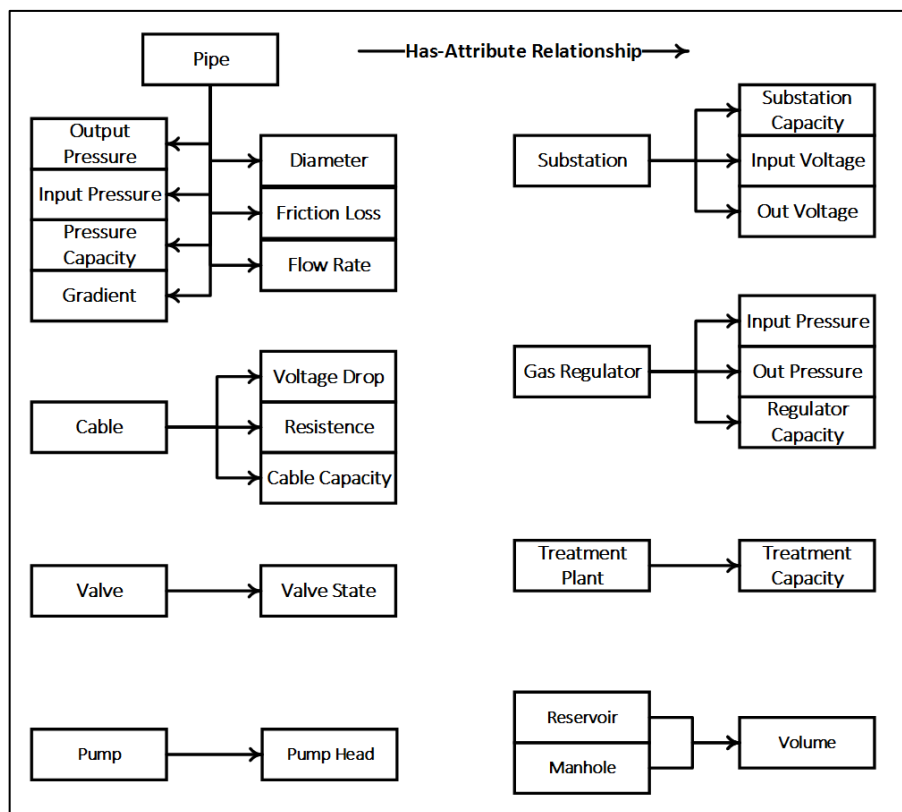
It is considered necessary to display entities from all different Utility Networks in figure 3.5. Because many entities have rich semantic relationships with each other. For example, a Water Valve and a Gas Valve are both Valves. A Valve and a Substation are both Assets. That helps to represent the rich semantics in my ontology. If an entity only belongs to a specific type of Utility Network, the entity is named using a specific colour (for example, yellow for Electricity Network). That allows us to construct ontologies specifically for Electricity Network, Water Network, Sewer Network, and Gas Network (figure 3.7).



**Figure 3.7.** Ontologies specific for each type of Utility Network.

Figure 3.7 also helps us to better understand the most vital topological connectivity within a specific type of Utility Network. For example, in Electricity Network, there is a “Connect” relationship from a Substation to a Feeder, and that means “A Substation connects to a Feeder”. The “Connect” relationships are identified from previous literatures in modelling utility distribution networks (Tanyimboh, et al., 2011; Avi, 2014; Vickridge, 2004; Osiadacz, 1987).

Finally, the attributes associated with entities of Utility Networks are displayed in figure 3.8. They are considered to be the most important attributes to characterize and model flow (electricity, supply water, waste water and gas) in the network. These attributes are described in table 3.6, with sources of choice given.



**Figure 3.8.** Attributes related to Utility Network.

Attributes	Value	Description	Source
Substation Capacity	Number in Watt	Maximum power a substation can supply.	NPG, 2013
Input Voltage /	Number in	Input and output voltage of a	NPG, 2013

Output Voltage	Voltage	substation.	
Resistance	Number in Ohm/m	Resistance of cable per meter, used to calculate voltage drop.	NPG, 2013
Voltage Drop	Percentage	Percentage of voltage drop along the cable.	NPG, 2013
Cable Capacity	Number in Watt	Maximum power a cable can supply.	NPG, 2013
Input Pressure / Output Pressure	Number in Bar	Input and output pressure of gas, or water in a pipe or gas regulator.	Rahal, et al., 1980; Osidascz, 1987
Diameter	Number in m	Diameter of the pipe, used to compute pipe friction loss.	Rahal, et al., 1980; Osidascz, 1987
Friction Loss	Number in Bar	Pressure loss (gas, water) due to inner friction within pipe.	Rahal, et al., 1980; Osidascz, 1987
Gradient	Number	Gradient of pipe, affects the flow rate.	Rahal, et al., 1980; Osidascz, 1987
Flow Rate	Number in m <sup>3</sup> /s	Amount of gas or water which flows through a pipe in a given time.	Rahal, et al., 1980; Osidascz, 1987
Pressure Capacity	Number in Bar	Maximum pressure a pipe can stand.	Osidascz, 1987
Regulator Capacity	Number in m <sup>3</sup> /s	Maximum amount of gas a regulator can process in a given time.	Osidascz, 1987
Treatment Capacity	Number in m <sup>3</sup> /s	Maximum amount of supply water or waste water a treatment can process in a given time.	Hammed, et al., 2004; Avi, 2014
Volume	Number in m <sup>3</sup>	Maximum amount of water a reservoir or manhole can hold.	Hammer, 1986; Avi, 2014
Pump Head	Number in m	How high the water will be pumped. The can cause additional pressure gain when water flow through a pump.	Hammer, 1986; Avi, 2014
Valve State	Boolean	Indicate whether the valve is open or close.	Osidascz, 1987

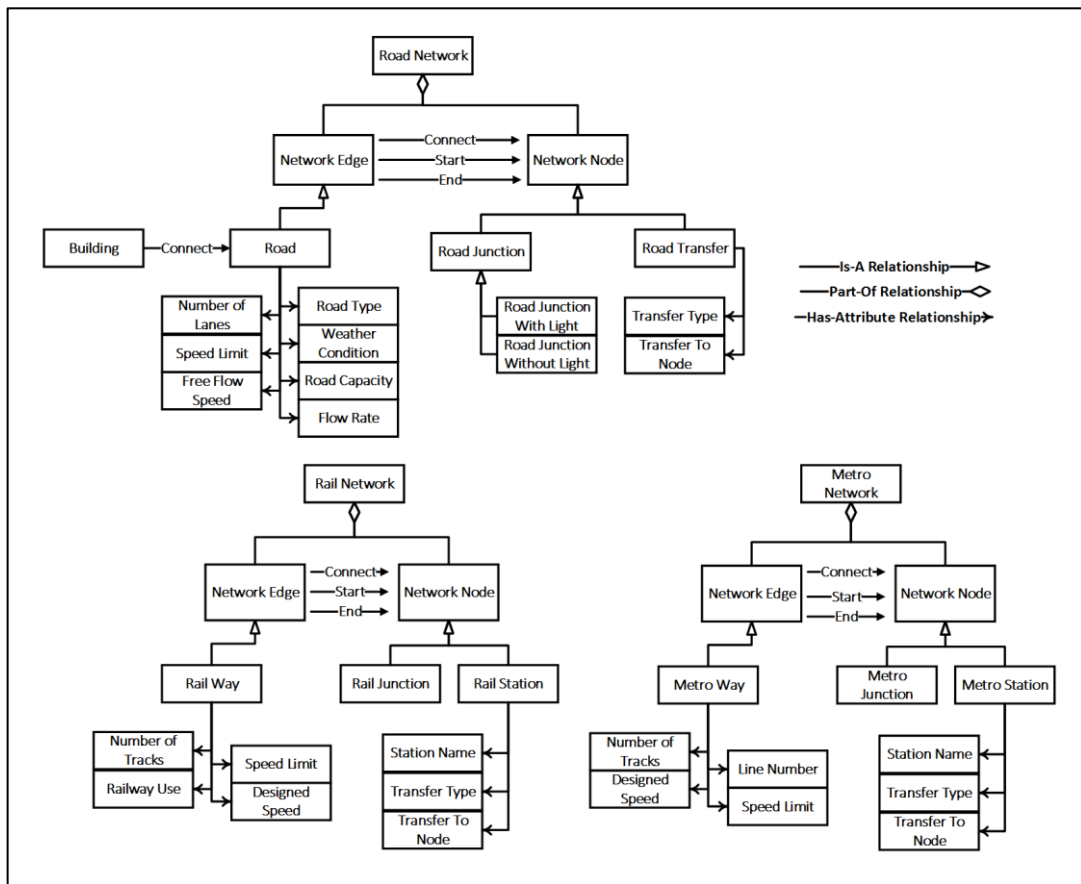
**Table 3.6.** Attributes related to the Utility Networks.

### 3.5 Transport Networks

Transport Networks are either Road Network, Rail Network or Metro Network. They are defined in figure 3.9. All three types of networks are defined based on INSPIRE data model



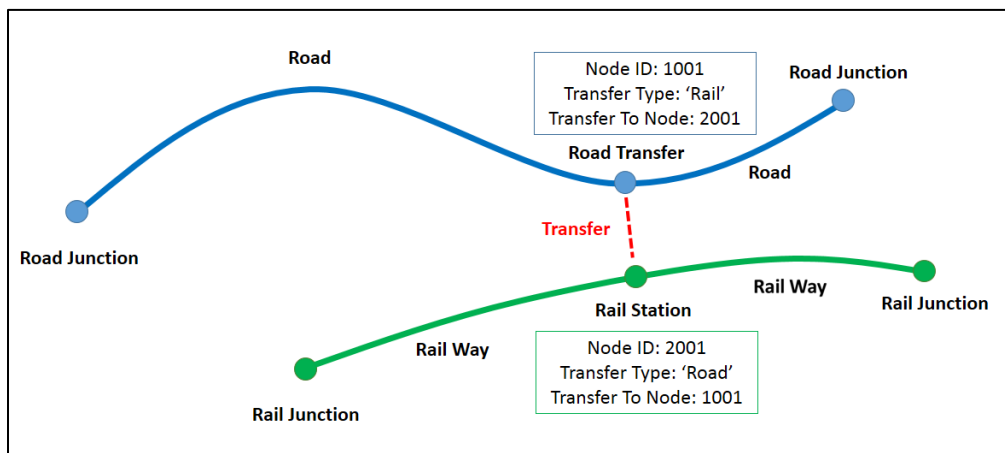
(INSPIRE, 2013) and OTN (Lorenz, 2005). Each of them is a subclass of Urban Infrastructure Network, consisting of Network Edge and Network Node.



**Figure 3.9.** Entities and relationships for Transport Network.

Typical road network models simply use edge (some call it arc or link) and node to represent the network itself (Katsumi, 2018). But that is not informative enough, when it is necessary to represent transfer service between different transport approaches (for example, from road to rail). In fact, considering the purpose of our ontology, understanding connectivity is essential. Both INSPIRE data model (INSPIRE, 2013) and OTN (Lorenz, 2005) deal with this issue by introducing an additional type of node (called “connection node” in INSPIRE and “transfer node” in OTN). That is why in this Road Network ontology, it is advisable to further break Network Node into subclass “Road Junction” and “Road Transfer”. Transfer is allowed to exist between a Road Transfer to a Rail Station or Metro Station.

An example is given in figure 3.10 to show the transfer service between Road Network and Rail Network. It is already known a Road Transfer and Rail Station are both Network Nodes, therefore they have a unique Node ID as an attribute. They also have attributes of Transfer Type (a text showing what service a passenger transfer to) and Transfer To Node (Node ID corresponding to a Road Transfer, Rail Station or Metro Station). This is how to represent the knowledge of transfer between different Transport Networks.



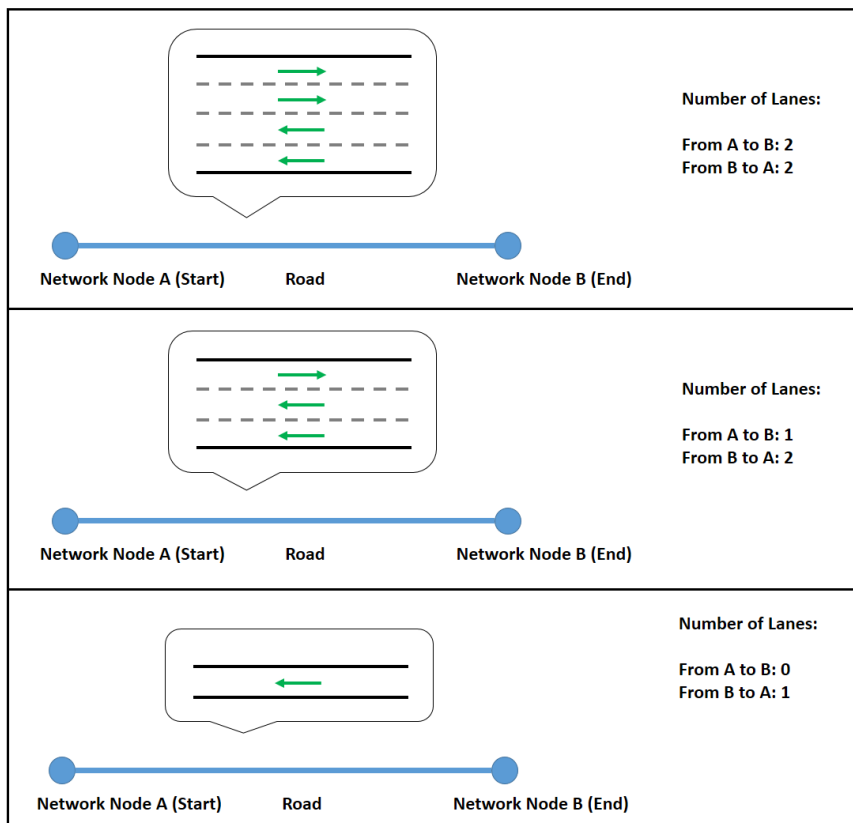
**Figure 3.10.** An example of transfer between Road Transfer and Rail Station.

For a Road Network, a Road Junction has subclass Road Junction with Light and Road Junction Without Light. This classification helps model dependency between Road Network and Electricity Network (Ouyang, 2012). A Road has its own attributes. They are chosen in order to characterize and model traffic flow. They are described in table 3.7, with source given to the choice of the attributes.

Road Attributes	Value	Description	Source
Number of Lanes	Integers	Number of lanes in both directions, a sequence of nodal pairs must be given.	INSPIRE, 2013
Speed Limit	Number in km/h	Highest allowed vehicle speed regardless of weather.	INSPIRE, 2013
Road Type	Text	Type of road, such as "A Road", "B Road", "Minor Road", or "Motor Way" in UK.	Department for Transport, 2005
Weather Condition	Text	"Sunny", "Rainy", or "Snowy"	Kyte, et al., 2001

		to show condition on the road. Bad weather will lower free flow speed.	
Free Flow Speed	Number in km/h	Highest speed a motorist is willing to travel on the road. Free flow speed can change based on weather condition and congestion (flow rate).	Banks, 1989
Flow Rate	Number in km/h	The rate of how many vehicles travel through a road in a given time. Flow rate depends on number of lanes, and has negative relationship with free flow speed.	Smith, et al., 2001
Road Capacity	Number in vehicle/h	The maximum flow rate obtainable on a given road using all available lanes.	Mogridge, 1997

**Table 3.7.** Attributes associated with Road.



**Figure 3.11.** Use sequence of Network Node to represent Number of Lanes.

Number of Lanes is the most important attribute, and must be defined in a careful way. In

here, the linear reference, i.e. sequence of Network Node pair, is still used to define it. An example is given in figure 3.11. In most cases, a Road can be travelled from both directions, and lanes number is equal in both directions. But in some situations, lane number can be not equal or the road is simply a one-way road. Using the approach shown in figure 3.11, then this ontology can handle this situation too.

The Rail Network and Metro Network are defined in the similar way, because they are both track based systems, and considered to be less complex than Road Network (INSPIRE, 2013; Lorenz, 2005). Important attributes are summarized in table 3.8.

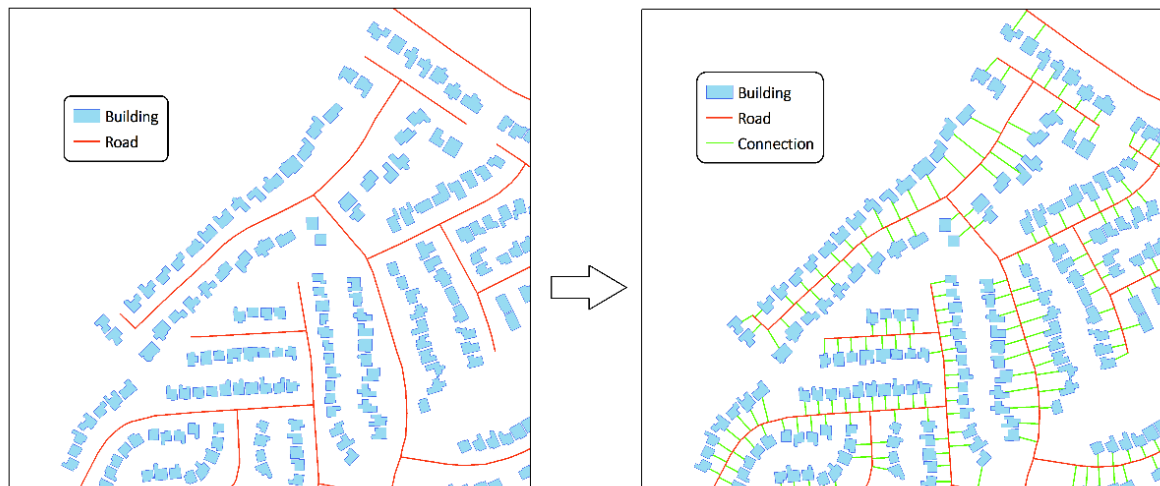
<b>Attribute</b>	<b>Value</b>	<b>Description</b>	<b>Source</b>
Number of Tracks	Integers	Number of tracks in both directions.	Lorenz, 2005
Designed Speed	Number in km/h	How fast the train should run on the track.	INSPIRE, 2013
Speed Limit	Number in km/h	Highest allowed speed of train.	Tutcher, 2016
Railway Use	Text	Only applies to Railway. What types of train can run, such as “Cargo”, “Passenger” or “Mixed”.	Tutcher, 2016
Line Number	Number or Numbers	Only applies to Metro Way. Shows what metro lines it belongs to.	INSPIRE, 2013

**Table 3.8.** Attributes associated with Rail Way or Metro Way.

Finally, Building is mentioned here, and it is argued that a Building should be connected to a Road. This is one major contribution of this ontology, because no other similar work has done that. Considering the fact that goods, service can be delivered to buildings via road network, and the fact people can enter road network from buildings, it is feasible to represent the connection between them. The interesting part is how to exactly represent it.

The decision was made to follow the approach developed by Cavallaro et al (2014), in which they use a straight line to connect the centroid of footprint of each building to its nearest road (figure 3.12). This is simple, straightforward, and sensible. For example, when a passenger from a building needs to travel to other places, he always needs to move to the nearest road

first and then start moving in the transport network. This is exactly the “Connect” relationship between Building and Road in this ontology. Semantically, that means “every Building connects to the nearest Road”, which is virtually a mapping from a Building to a Road. This connection will be formally represented in section 3.7.



**Figure 3.12.** The connection between Building and Road.

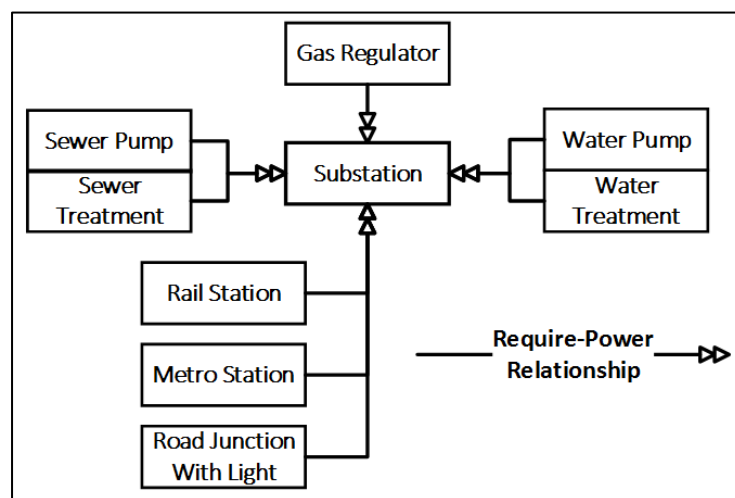
Note that Building is allowed to connect Road, but not to Rail Way or Metro Way, even if residents from the Building do need to access the Rail Network and Metro Network. This is because Rail Network and Metro Network can only be accessed at a Rail Station or Metro Station. Even if there is a Railway very close a resident’s house, he cannot directly “jump” to it to access the Rail Network. Instead, he needs to go to the nearest Road, and travel along the Road Network until he can reach a Rail Station (via a Road Transfer).

### 3.6 Dependencies

In infrastructure dependency related ontologies (McNally, et al., 2007; Sicilia, et al., 2009), a dependency is represented as unidirectional relationship from entity A to B, which reads as “A depends on B”. While in a broader sense, interdependency can be seen as special case of dependency (Ouyang, 2014), where there are two dependencies in the opposite directions between A and B, which reads as “A depends on B, and B depends on A”. Therefore, in this ontology, only dependencies will be explicitly represented.

There are two major types of infrastructure dependencies (Zimmerman, et al., 2001), which are functional dependency and spatial dependency. Functional dependency refers to the situation, where operation of A depends on the material, resource, or signal from B (for example, water pump requires electricity from a substation). Spatial dependency refers to the spatial proximity (for example, a water pipe and a gas pipe are very close with each other spatially, when earthquake breaks one of them, the other one can be affected as well).

In this ontology, it is considered that only functional dependencies will be represented. Because spatial dependency can be implicitly inferred from the Node Geometry and Edge Geometry that are defined earlier. The dependencies that exist in my ontology are shown in figure 3.13. These are all the power requirements identified from literatures (Tanyimboh, et al., 2011; Avi, 2014; Vickridge, 2004; Osiadacz, 1987). Note here only the “electricity power requirement” relationship is represented in my ontology. Broadly speaking, there exist requirements of other resources between utility networks. For example, an electricity generator requires water from pumping station to cool down (Ouyang, 2014). But the ontology only focuses on fine spatial scale (infrastructure distribution level), so that electricity generator is not represented in our Electricity Network, and therefore the water requirement is not represented. The dependency in my ontology can be formally represented as a mapping relationship, and is explained in section 3.7.



**Figure 3.13.** Relationships to represent dependency.

### 3.7 Formal Representation of Ontology

With regards to the urban infrastructure networks and buildings, a city  $C$  can be described with the help of network theory:

$$C = \{N, B, R\}$$

$$N = \{U, T\}$$

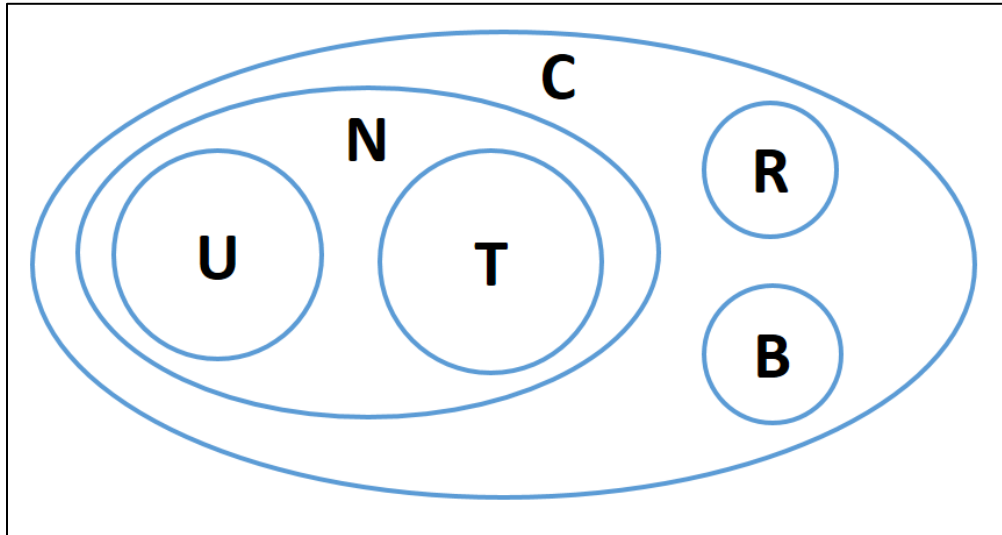
$$U = \{G_e, G_g, G_w, G_s\}$$

$$T = \{G_r, G_t, G_m\}$$

These notations are explained in table 3.9. Figure 3.14 helps to understand relationships of these notations (sets) visually.

Notation	Description
$C$	Set to denote the city.
$N$	Set to denote the urban infrastructure networks within the city.
$B$	Set to denote the buildings within the city.
$R$	Set to denote the relationships within city.
$U$	Set to denote the utility networks within city.
$T$	Set to denote the transport networks within city.
$G_e$	A network instance to denote electricity network. $G_e = \{V_e, E_e, f_e\}$
$G_g$	A network instance to denote gas network. $G_g = \{V_g, E_g, f_g\}$
$G_w$	A network instance to denote the water network. $G_w = \{V_w, E_w, f_w\}$
$G_s$	A network instance to denote the sewer network. $G_s = \{V_s, E_s, f_s\}$
$G_r$	A network instance to denote the road network. $G_r = \{V_r, E_r, f_r\}$
$G_t$	A network instance to denote the rail network. $G_t = \{V_t, E_t, f_t\}$
$G_m$	A network instance to denote the metro network. $G_m = \{V_m, E_m, f_m\}$

**Table 3.9.** Explanations for basic notations.

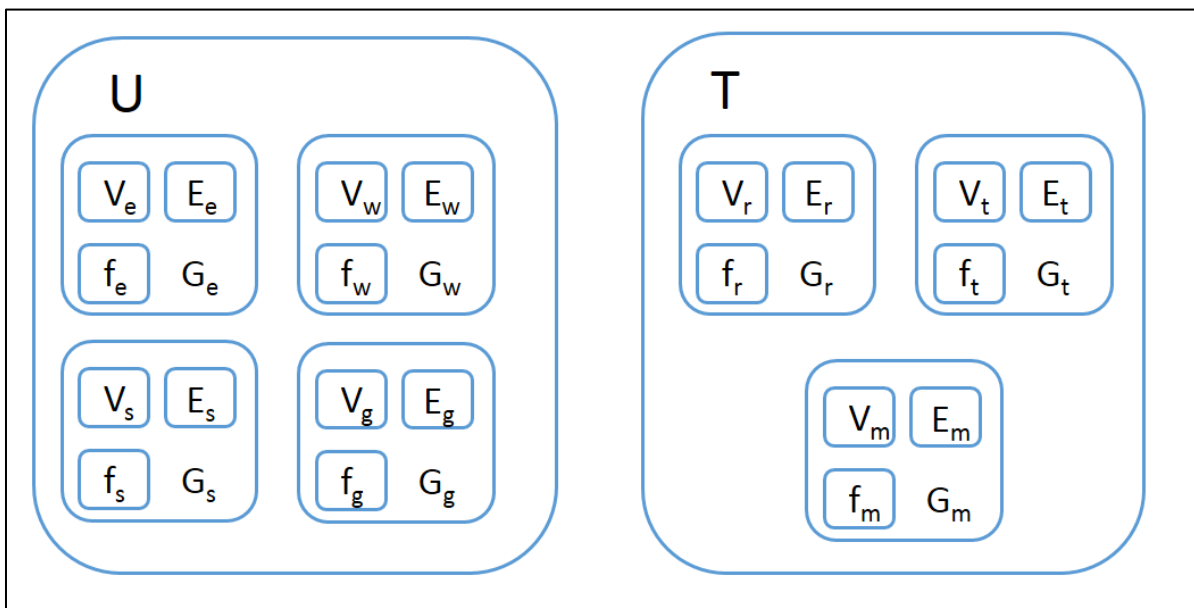


**Figure 3.14.** Visual representation of basic notations (sets).

Note each of the  $G_e, G_g, G_w, G_s, G_r, G_t, G_m$ , is considered to be a directed network model mathematically. That means any of them (such as  $G_e$ ) can be further defined as follows:

$$G_e = \{V_e, E_e, f_e\}$$

In here  $V_e$  refers the set of nodes in electricity network, where  $E_e$  refers to the set of edges in the electricity network. The  $f_e$  is a function that maps each element in  $E_e$  to an ordered pair of two nodes in  $V_e$ . This is represented by figure 15 visually.



**Figure 3.15.** Visual representation of different infrastructure networks.



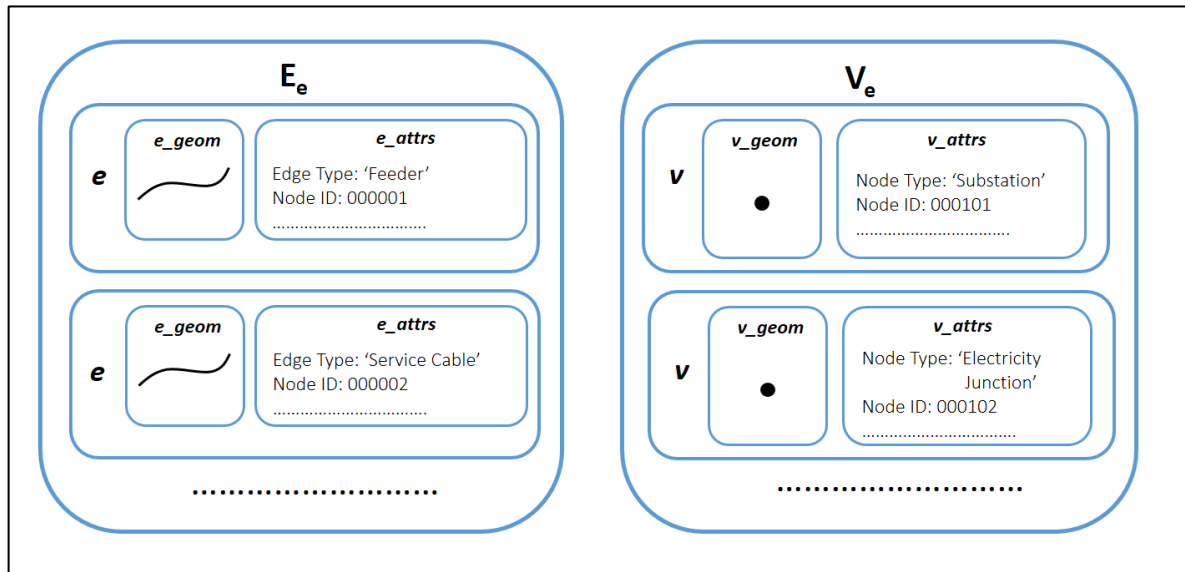
Any network edge defined here, can be further broken into two parts, to indicate the geometry and attributes of that edge. For example, let us use  $v$  to denote the element in  $V_e$ , then  $v$  can be represented as:

$$v = \{v\_geom, v\_attrs\}$$

In here,  $v\_geom$  is the geometry of that edge, which is a point, and  $v\_attrs$  is (key:value) pairs to indicate the attributes of that node. Similarly, let us use  $e$  to denote the element in  $E_e$ , then  $e$  can be represented as:

$$e = \{e\_geom, e\_attrs\}$$

In here,  $e\_geom$  is the geometry of that edge, which is a polyline, and  $e\_attrs$  is (key:value) pairs to indicate the attributes of that edge. This can be better understood in figure 3.16.



**Figure 3.16.** Visual representation on the constitution of an edge  $e$  and a node  $v$ , in the electricity network  $G_e$ .

Figure 3.16 shows how an edge  $e$  and a node  $v$  can be further broken down in the electricity network  $G_e$ . Such rules also apply to other networks, namely  $G_g$ ,  $G_w$ ,  $G_s$ ,  $G_r$ ,  $G_t$ ,  $G_m$ .

For the set  $B$ , it is set of buildings. Let us use  $b$  to denote the element in  $B$ , and the  $b$  is

virtually an individual building. It can be further broken down into two parts:

$$b = \{b\_geom, b\_attrs\}$$

The  $b\_geom$  is the geometry of the building. It can be represented by either a 3D body (if data is available) or simply a 2D polygon as the footprint of that building. The  $b\_attrs$  is the (key:value) pairs indicate the attributes of that building.

Finally, for the set  $\mathbf{R}$ , it can be described as follows:

$$\mathbf{R} = \{\mathbf{R}_{dep}, \mathbf{R}_{con}\}$$

$$\mathbf{R}_{dep} = \{f_{w\_e}, f_{g\_e}, f_{s\_e}, f_{r\_e}, f_{t\_e}, f_{m\_e}\}$$

$$\mathbf{R}_{con} = \{fb_r, fb_e, fb_w, fb_s, fb_g\}$$

The notations are explained in table 3.10.

<b>Notation</b>	<b>Description</b>
$\mathbf{R}_{dep}$	Set to denote the infrastructure dependencies within the city.
$\mathbf{R}_{con}$	Set to denote the connections between buildings and infrastructure networks within the city.
$f_{w\_e}$	A mapping to represent the dependencies from water network to electricity network.
$f_{g\_e}$	A mapping to represent the dependencies from gas network to electricity network.
$f_{s\_e}$	A mapping to represent the dependencies from sewer network to electricity network.
$f_{r\_e}$	A mapping to represent the dependencies from road network to electricity network.
$f_{t\_e}$	A mapping to represent the dependencies from rail network to electricity network.
$f_{m\_e}$	A mapping to represent the dependencies from metro network to electricity network.
$fb_r$	A mapping to represent the connection between building and road network.
$fb_e$	A mapping to represent the connection between building and electricity network.
$fb_w$	A mapping to represent the connection between building and water network.
$fb_s$	A mapping to represent the connection between building and sewer network.
$fb_g$	A mapping to represent the connection between building and gas network.

**Table 3.10.** Description on notations with regards to  $\mathbf{R}$ .

$\mathbf{R}_{\text{dep}}$  is used to represent the infrastructure dependencies via all seven mappings defined within it. Until now it is still not explained, from which set to which set, each of the mappings applies. To explain it in an easier way, several subsets will be defined first. Let  $v$  denote an element (a node) from  $\mathbf{V}_e$ , and  $v.\text{node\_type}$ , refers to its Node Type, then a subset of  $\mathbf{V}_e$  can be defined as follows:

$$\mathbf{V}_{\text{substation}} = \{v \in \mathbf{V}_e \mid v.\text{node\_type} = \text{'Substation'}\}$$

In here, all the nodes  $v$  from  $\mathbf{V}_e$  are selected, which are the substation nodes and then put them to a set called  $\mathbf{V}_{\text{substation}}$ . Being able to represent subsets allows us to explain the scope of the mappings, see table 3.11.

Mapping	Mapping Scope
$\mathbf{f}_{w\_e}$	$\mathbf{V}_{w\_subset} = \{v \in \mathbf{V}_w \mid v.\text{node\_type} = \text{'Water Pump'} \vee v.\text{node\_type} = \text{'Water Treatment'}\}$ $\mathbf{f}_{w\_e}: \mathbf{V}_{w\_subset} \rightarrow \mathbf{V}_{\text{substation}}$
$\mathbf{f}_{g\_e}$	$\mathbf{V}_{g\_subset} = \{v \in \mathbf{V}_g \mid v.\text{node\_type} = \text{'Gas Regulator'}\}$ $\mathbf{f}_{g\_e}: \mathbf{V}_{g\_subset} \rightarrow \mathbf{V}_{\text{substation}}$
$\mathbf{f}_{s\_e}$	$\mathbf{V}_{s\_subset} = \{v \in \mathbf{V}_s \mid v.\text{node\_type} = \text{'Sewer Pump'} \vee v.\text{node\_type} = \text{'Sewer Treatment'}\}$ $\mathbf{f}_{s\_e}: \mathbf{V}_{s\_subset} \rightarrow \mathbf{V}_{\text{substation}}$
$\mathbf{f}_{r\_e}$	$\mathbf{V}_{r\_subset} = \{v \in \mathbf{V}_r \mid v.\text{node\_type} = \text{'Road Junction With Light'}\}$ $\mathbf{f}_{r\_e}: \mathbf{V}_{r\_subset} \rightarrow \mathbf{V}_{\text{substation}}$
$\mathbf{f}_{t\_e}$	$\mathbf{V}_{t\_subset} = \{v \in \mathbf{V}_t \mid v.\text{node\_type} = \text{'Rail Station'}\}$ $\mathbf{f}_{t\_e}: \mathbf{V}_{t\_subset} \rightarrow \mathbf{V}_{\text{substation}}$
$\mathbf{f}_{m\_e}$	$\mathbf{V}_{m\_subset} = \{v \in \mathbf{V}_m \mid v.\text{node\_type} = \text{'Metro Station'}\}$ $\mathbf{f}_{m\_e}: \mathbf{V}_{m\_subset} \rightarrow \mathbf{V}_{\text{substation}}$

**Table 3.11.** Scopes for mappings  $\mathbf{f}_{w\_e}$ ,  $\mathbf{f}_{g\_e}$ ,  $\mathbf{f}_{s\_e}$ ,  $\mathbf{f}_{r\_e}$ ,  $\mathbf{f}_{t\_e}$ ,  $\mathbf{f}_{m\_e}$ .

$\mathbf{R}_{\text{con}}$  is used to represent connections between buildings and infrastructure networks, such as connection from a building to a road (section 3.6), or the connection between an electricity service node to a building (section 3.5). Five mappings are defined within  $\mathbf{R}_{\text{con}}$ . Subsets are also used here, to clarify the scopes for these mappings, which is shown in table 3.12.

Mapping	Mapping Scope
$\mathbf{f}_{b\_r}$	$\mathbf{f}_{b\_r}: \mathbf{B} \rightarrow \mathbf{E}_r$
$\mathbf{f}_{b\_e}$	$\mathbf{V}_{e\_subset} = \{v \in \mathbf{V}_e \mid v.\text{node\_type} = \text{'Electricity Service Node'}\}$ $\mathbf{f}_{b\_e}: \mathbf{V}_{e\_subset} \rightarrow \mathbf{B}$

$\mathbf{fb}_w$	$V_{w\_subset} = \{v \in V_w \mid v.node\_type = \text{'Water Service Node'}\}$ $\mathbf{fb}_w: V_{w\_subset} \rightarrow \mathbf{B}$
$\mathbf{fb}_s$	$V_{s\_subset} = \{v \in V_s \mid v.node\_type = \text{'Sewer Service Node'}\}$ $\mathbf{fb}_s: V_{s\_subset} \rightarrow \mathbf{B}$
$\mathbf{fb}_g$	$V_{g\_subset} = \{v \in V_s \mid v.node\_type = \text{'Gas Service Node'}\}$ $\mathbf{fb}_g: V_{g\_subset} \rightarrow \mathbf{B}$

**Table 3.12.** Scopes of mappings  $\mathbf{fb}_r$ ,  $\mathbf{fb}_e$ ,  $\mathbf{fb}_w$ ,  $\mathbf{fb}_s$ ,  $\mathbf{fb}_g$ .

### 3.8 Conclusion

In this chapter an ontology was developed to represent the urban infrastructure networks and buildings within the city. The major contribution of this work is, at individual building level, to identify the connections within infrastructure networks and the connections between buildings and infrastructures. Basic attributes that are associated with urban infrastructure networks, which allows us to model and characterize flows within infrastructure networks.

Moreover, this ontology is defined in a spatially explicit manner, in which geometry and spatial relationships can be represented. The ontology also includes all the major utility and transport infrastructure networks, which are considered as major added value compared with existing research. The ontology will be used as a conceptual model and implemented in Chapter 4, 5, 6, and 7 to model different high granularity geospatial infrastructure networks.

## **Chapter 4. A heuristic spatial algorithm for generating fine-scale infrastructure distribution networks**

### **4.1 Introduction**

In Chapter 3, a formal ontology was developed for modelling fine scale geospatial urban infrastructure networks at building level. Spatial network model with attributes (edges and nodes associated with geometry and attributes) is used to represent an infrastructure network. The network topology helps to understand the connectivity between infrastructure assets and buildings they serve at fine spatial scale. The network attributes allow for running generic network simulation on infrastructure networks (for example, voltage drop simulation on the electricity network, and traffic flow simulation on the road network, etc.). The network geometry helps to understand how spatially infrastructure networks interact with urban environment (for example, if flood occurs, some electricity substations can malfunction due to falling within footprint of flood, etc.).

As such, acquiring the spatial layout (geometry) of urban infrastructure networks which connect assets and buildings is considered as an essential step for modelling them. However, in Chapter 2, challenge of acquiring good quality geospatial data has been discussed. The major reason is that private utility companies restrict public use of their data (Bon, 2017) or that they simply do not have the data in the geospatial format (Fu, et al., 2008). That means, in the worst case, except the location of infrastructure assets (such as electricity substations) and buildings, nothing is known about geometry of the infrastructure networks (such as the layout of electricity cables). Therefore, when actual data is not available, it is essential to have approaches that can automatically infer fine scale geospatial layout of the infrastructure networks.

At the time of conducting this PhD project, there is no existing approach or algorithm for this specific problem (generating spatial network connecting assets and buildings). Geospatial

network generation algorithms in the infrastructure domain are sparse, although there are still some related examples observed, which are shown in table 4.1.

Author	Description of network generation algorithm
Gastner, et al., 2006	The algorithm focuses on designing large scale spatial distribution network, and in particular the location of facilities such as hospitals and airports. Facility locations are designed with a non-uniform population density, so that average distance from a person's home to the nearest facilities is minimized. The algorithm is suitable for large scale facility planning (such as for entire US), but not useful in city (where population density changes slightly). Moreover, it focuses more on "assets location" planning, rather than "network layout" planning.
Trifunovic', et al., 2013	The algorithm is used for planning layout of water distribution network for properties. To make the algorithm work, "seed nodes" must be defined already. The seed nodes refer to the pipe junctions that are allowed to exist in the synthetic network. No other additional nodes can be created. The author also makes extra constraints, such as "each seed node can connect no more than 3 pipes". Additional parameters must be given (diameter of pipe, demand for each property node, etc.) to decide the optimal layout of water distribution network. The algorithm can generate more plausible network (since it considers hydrology condition) but seed nodes (junctions) must be explicitly given, which are normally missing in our case.
Hadas, et al., 2013	The algorithm focuses on designing an optimal spatial network in terms of minimizing construction cost and evacuation time (under terrorism activities). User needs to first explicitly define the nodes that exist in the network. Moreover, there will be some nodes called "origin nodes", and some called "destination nodes". Residents must move from origin nodes to destination nodes for terrorism evacuation. Pre-defined node location is the major disadvantage of this approach, like Trifunovic' et al (2013).
Cavallaro, et al., 2014	Strictly speaking, this approach is not a network generation or design algorithm. Instead, the author suggests that individual buildings and infrastructure assets should be connected to the nearest road network to construct a hybrid network to assess resilience within the city. Although it is not an actual algorithm, it suggests that layout of infrastructure networks is related to the road network.
Heijnen, et al., 2014	The author developed an approach to design geometric infrastructure network connecting a source node and demand nodes with least construction cost. This problem is actually "finding an edge-weighted Steiner minimal tree that connects all the demand nodes to the one source node within a bounded region". This is a generic algorithm can be applied for any type of infrastructure. But there are two drawbacks: (1) The algorithm only focuses on one source (asset) node but not multiple; (2) The algorithm does not consider the fact the infrastructure network is related to the road network.
Dunn, et al., 2016	An algorithm was developed to generate the dynamic spatial nodal layout of large-scale infrastructure networks (such as UK rail network, US airport network). The

	author focuses more on the evolution of the networks rather than its current static layout. Several assumptions must be made to make the algorithm work (such as nodal cluster size will change over time, etc.). The major weakness of this algorithm is that it does not consider any consumer nodes, and that it does not generate the layout of the network edges.
Bon, 2017	The author developed an approach for generating layout of underground utility networks at individual building level for Amsterdam, the Netherlands. To make the algorithm work, layout of main lines (cables or pipes) must be known first, then connections can be made from individual buildings (for example, an access point within the building) to the main lines. Clearly, this algorithm makes the assumption that layout of main lines is available, which is unfortunately not our case. But the author pointed out in general, main lines should follow the road network, which is a good suggestion to us.

**Table 4.1.** Related studies in generating geospatial infrastructure network.

In a more general way, automatic infrastructure network layout generation is a network design problem (NDP) (Magnanti, et al., 1984), which aims to construct a network with different constraints or objective functions. From table 4.1, although none of the approaches is directly related to our problem, each of them applies some constraints (for example, network construction cost is minimized, etc.). This is essential in generating a spatial network instance in a deterministic way. However, it is clear that most of these approaches need to know location of all the nodes within the network, which is not feasible in our case. To be clear, location of building nodes and asset nodes are known in our case, but not the location of all the network junctions. That is why, the approaches developed by Cavallaro et al (2014) and Bon (2017) are considered most useful to our situation, because they argue the layout of infrastructure network is associated with road network. Moreover, road network data is made public in many countries and it is easy to access them.

Following this rationale, a spatial heuristic algorithm is developed based on the location of infrastructure assets, buildings, and road network. The output is the spatial layout of infrastructure networks connecting these assets and buildings. Details will be discussed in the following sections. Section 4.2 gives an overview of the algorithm. Section 4.3 describes the algorithm in a formal way. Section 4.4 introduces the computational implementation of this algorithm (software stack, libraries, etc.). Section 4.5 shows the pilot study, which is

generating city scale electricity distribution networks for the city of Newcastle upon Tyne. Section 4.6 is the validation on the synthetic network result. Section 4.7 is about transferability test of the algorithm. Section 4.8 concludes this chapter.

## 4.2 Algorithm Overview

The algorithm aims to generate fine spatial scale plausible distribution networks that connect assets to the dependent buildings. It might not produce 100% exact layout of the actual network, but it aims to be as close as possible. The algorithm is considered to be a generic solution to any type of infrastructure network, as long as layouts of the assets, buildings, and road network are known.

The algorithm is built on several basic assumptions:

**Basic Assumption 1** – Each individual building depends on one and only one asset.

**Basic Assumption 2** – The cables and pipes should be paved along road network. Buildings are connected to assets via network cables or pipes as short as possible.

**Basic Assumption 3** – Spatially, individual buildings can form clusters, and buildings within a cluster must depend on the same asset.

A diagram is shown in figure 4.1 to explain how the algorithm works in a general way. In here, algorithm reads three input datasets,  $A$ ,  $B$  and  $R$ , which stand for the sets of assets, buildings and roads. After reading initial input, the algorithm will go through two major processes (topology generation and geometry generation) to generate spatial distribution networks as the result.

Figure 4.2 shows an example of input data for this algorithm, and figure 4.3 shows the output based on figure 4.2.



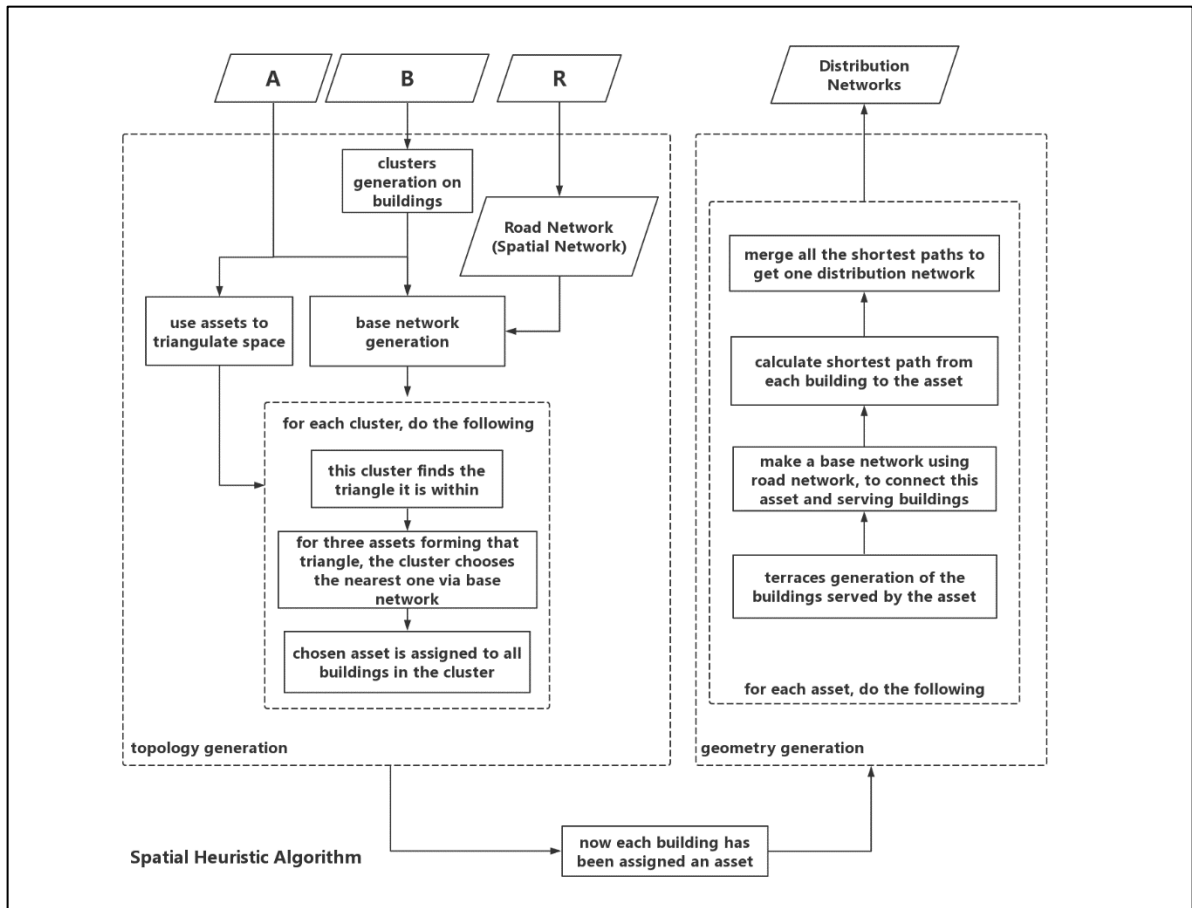


Figure 4.1. Flow of spatial heuristic algorithm.

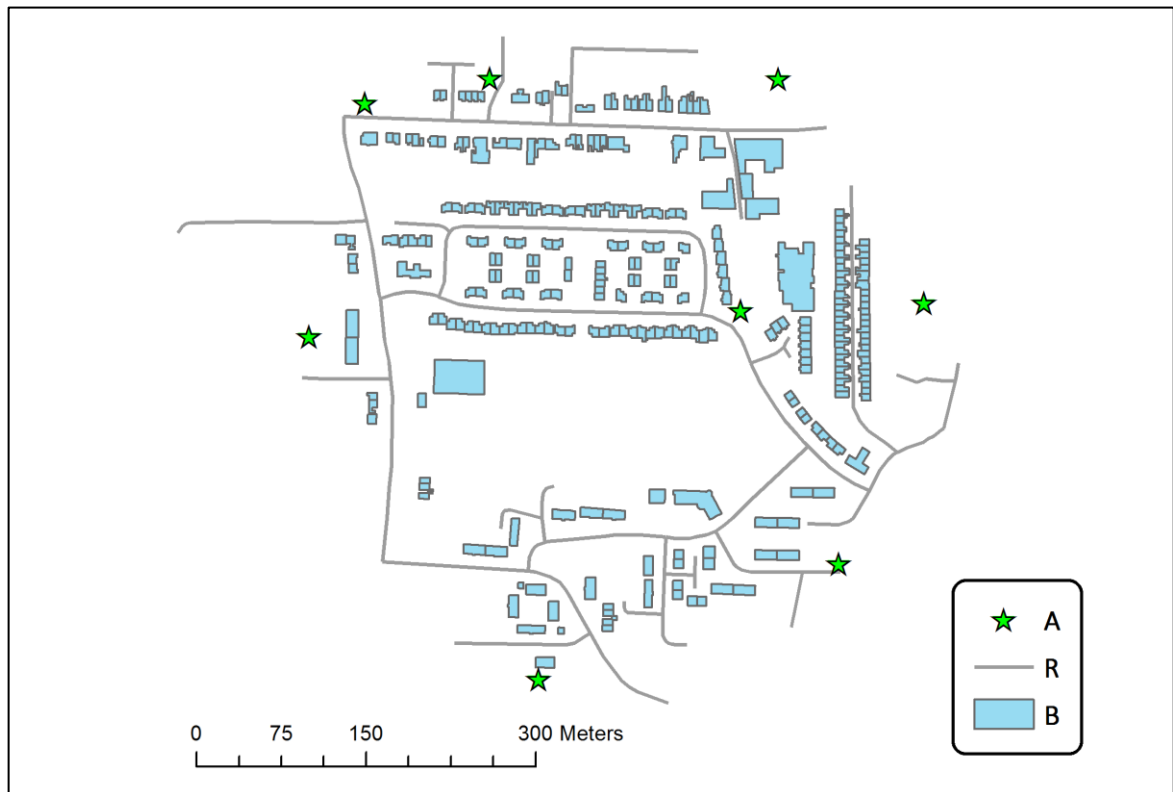
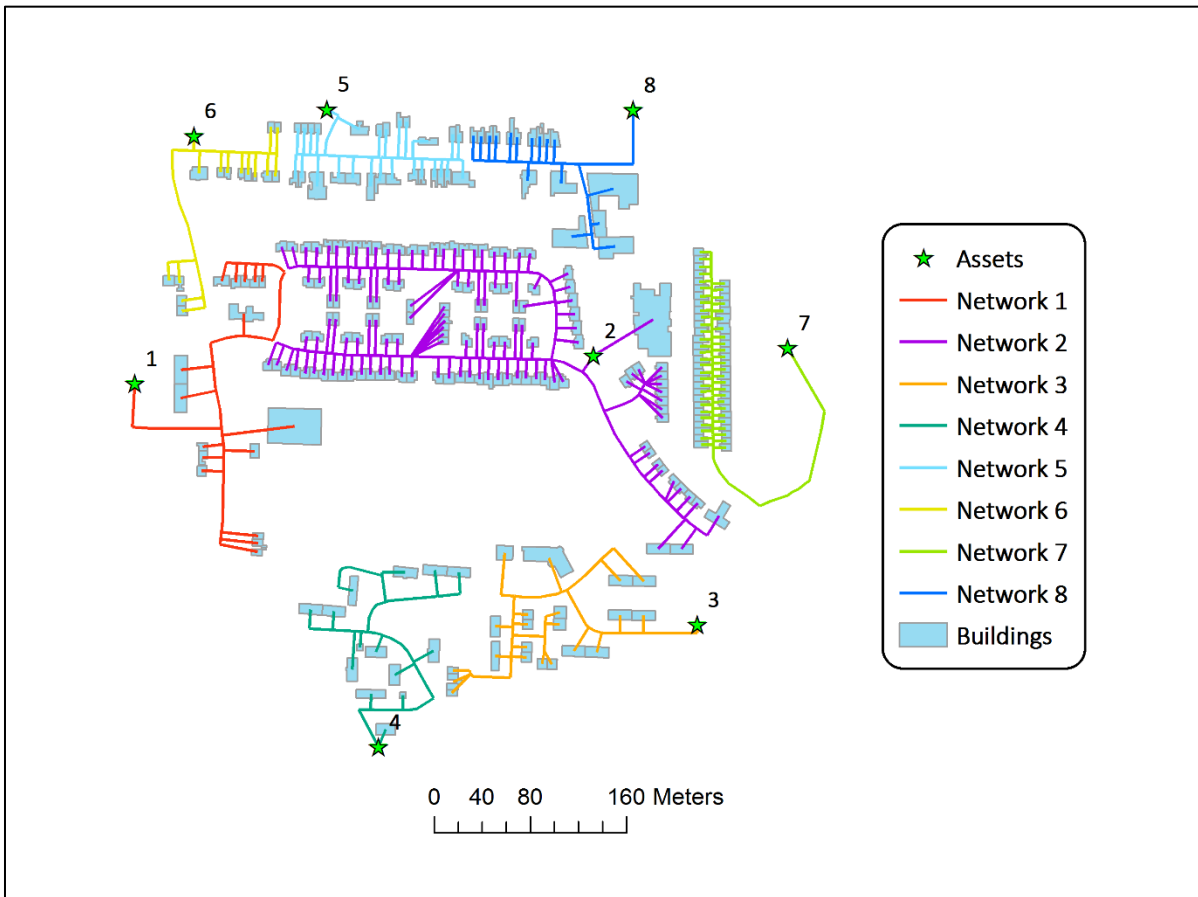


Figure 4.2. Example of algorithm input data (Contains OS data © 2018).



**Figure 4.3.** Example of algorithm output result (Contains OS data © 2018).

### 4.3 Algorithm Description

The sets  $R = \{r_1, r_2, \dots, r_i\}$ ,  $B = \{b_1, b_2, \dots, b_j\}$ , and  $A = \{a_1, a_2, \dots, a_k\}$  are used to denote the spatial objects representing roads, buildings and assets within the spatial domain under consideration. In particular, a road, a building and an asset should be represented by polyline, polygon, and point respectively. These three sets  $R$ ,  $B$ , and  $A$  are necessary input for the algorithm. For example, in figure 4.2, there are 8 assets, 288 buildings and 61 roads.

The algorithm can be divided into two sequential major steps: topology generation and geometry generation. Step one (topology generation) will assign an asset to each building, and step two (geometry generation) will generate the spatial network instance connecting each asset and all its dependent buildings.

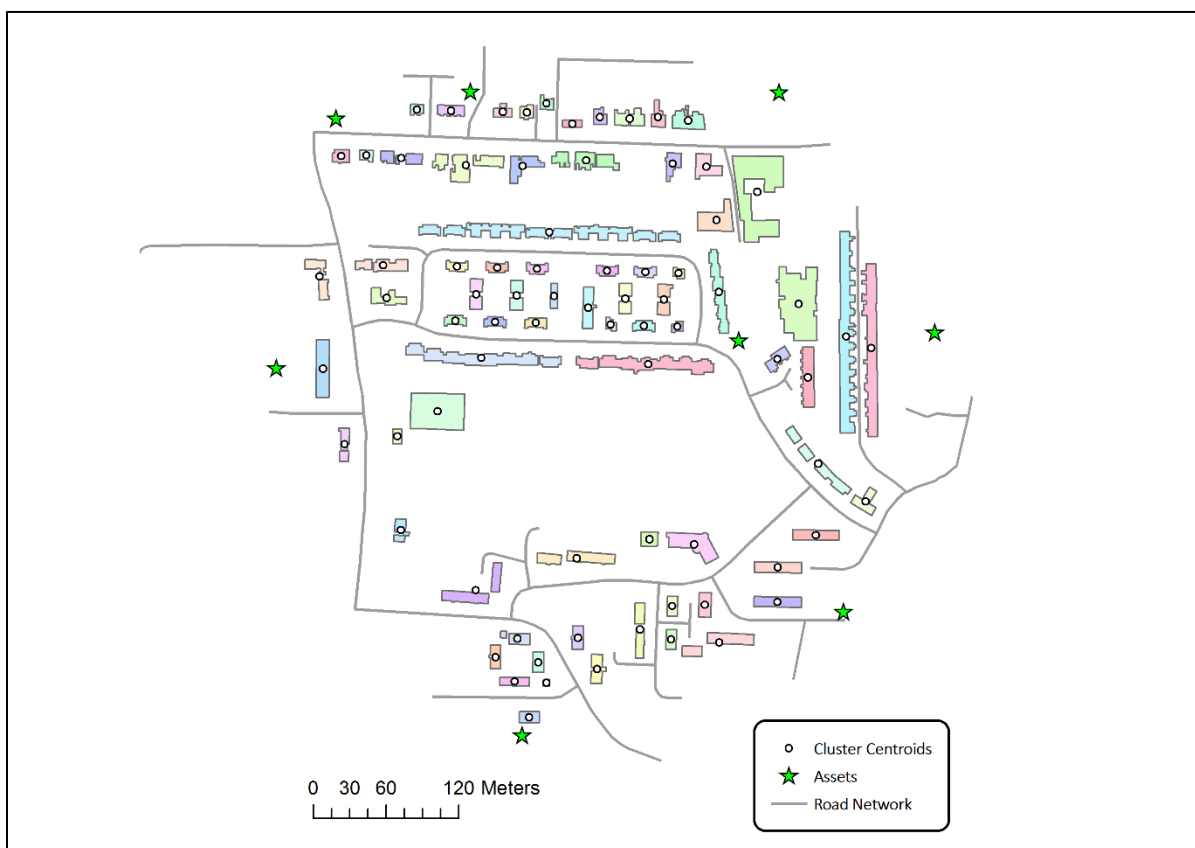
### 4.3.1 Topology Generation

<b>Algorithm 1:</b> Generation of Network Topology	
	<b>Input:</b> $A, B, R, G_{road}$
	<b>Output:</b> $f_{assign} : B \rightarrow A$
	1: initialize function $f_{assign}$ ;
Clusters Generation	{ 2: decide the threshold distance $d_{thresh}$ ; 3: use $B$ and $d_{thresh}$ to perform buffered cascaded union operation, generate the set of clusters $C$ ;
Calculate access points for each cluster and asset	{ 4: <b>for</b> each $c_i \in C$ <b>do</b> 5:   calculate cluster centroid $c_i.cen$ ; 6:   find the nearest road segment $r \in R$ to $c_i.cen$ , denoted as $r_{nearest}$ ; 7:   project $c_i.cen$ to $r_{nearest}$ to get the access point $c_i.acc$ ; 8: <b>end for</b> 9: <b>for</b> each $a_i \in A$ <b>do</b> 10:    find the nearest road segment $r \in R$ to $a_i$ , denoted as $r_{nearest}$ ; 11:    project $a_i$ to $r_{nearest}$ to get the access point $a_i.acc$ ; 12: <b>end for</b> 13: copy the road network $G_{road}$ to $G_{base}$ ;
Connect clusters and assets to the road network to generate base network	{ 14: <b>for</b> $a_i \in A$ <b>do</b> 15:   add a new node in $G_{base}$ at the position $a_i.acc$ ; 16:   add an edge in $G_{base}$ to connect $a_i.acc$ and $a_i$ ; 17: <b>end for</b> 18: <b>for</b> $c_i \in C$ <b>do</b> 19:   add a new node in $G_{base}$ at the position $c_i.cen$ ; 20:   add an edge in $G_{base}$ to connect $c_i.acc$ and $c_i.cen$ ; 21: <b>end for</b> 22: use $A$ to triangulate the space, return the triangles in the set $TR$ ;
Assign an asset to each cluster	{ 23: <b>for</b> $c_i \in C$ <b>do</b> 24:   find the triangle $tr \in TR$ containing $c_i.cen$ denoted as $tr_{contain}$ ; 25:   find the three asset points $a_k, a_l, a_m$ that form $tr_{contain}$ ; 26:   from $a_k, a_l, a_m$ , find the point that has shortest path distance to $c_i.cen$ via $G_{base}$ , denote the point as $a_n$ ; 27: <b>for</b> $b \in B$ contained by $c_i$ <b>do</b> 28:     update function $f_{assign}(b) = a_n$ ; 29: <b>end for</b> 30: <b>end for</b>

**Listing 4.1.** The pseudo code for topology generation process.

Before this process starts, a spatial network instance  $G_{road}$  (based on  $R$ ) needs to be generated to represent the road network. The topology generation process can be represented by the pseudo code shown in Listing 4.1. Necessary input includes  $A$ ,  $B$ , and  $R$ , and  $G_{road}$ . The expected output is a mapping relationship:  $f_{assign}: B \rightarrow A$ , so that for every individual building  $b$  there is one and only one corresponding asset  $a$  to it. Sequentially, this entire process can be divided into small steps.

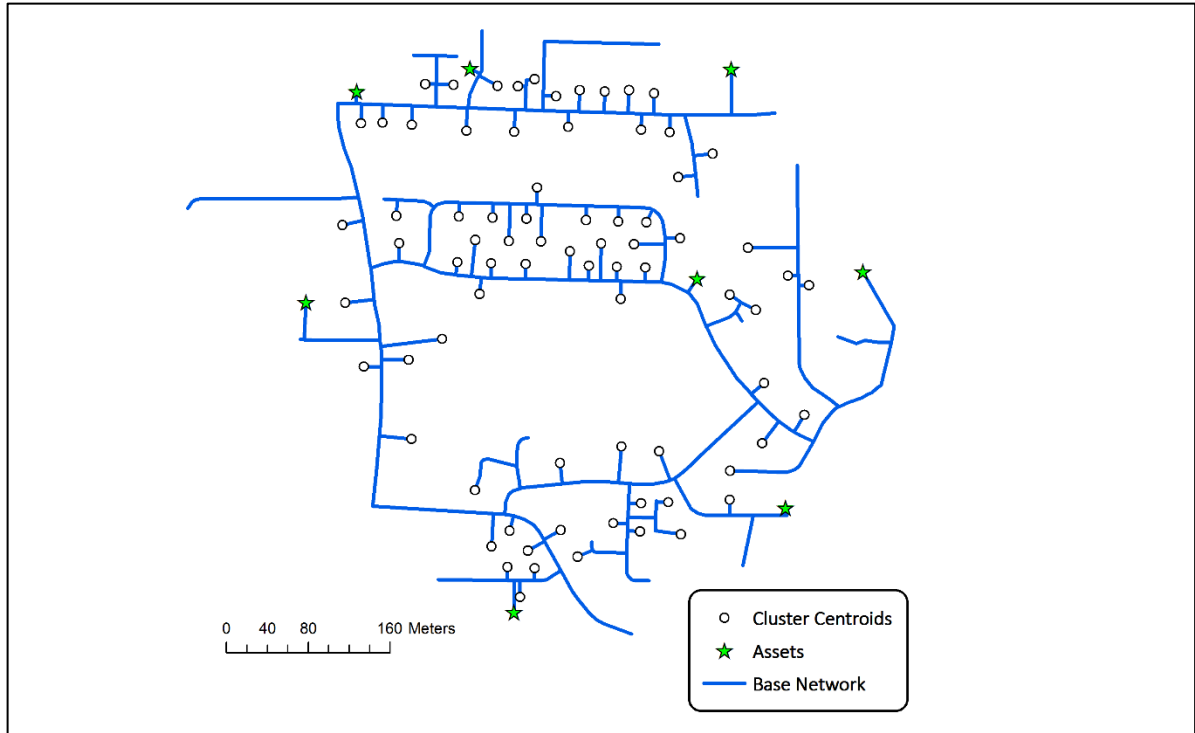
The first step (line 2-12 in listing 4.1) is to find clusters of the buildings, using an approach called “buffered cascading union” (Shapely, 2018). There is a parameter (called threshold distance  $d_{thresh}$ ) to control the clustering process. If distance of any two buildings is less than  $d_{thresh}$ , then they belong to the same cluster. As a result, the “buffered cascading union” operation will generate several clusters based on the building footprints, and for any two clusters, the Euclidean distance between any building in the first cluster and any building in the second cluster, is always greater than the  $d_{thresh}$ . A multi-polygon object is created to represent each cluster.



**Figure 4.4.** 77 clusters generated based on the buildings from figure 4.2 (Contains OS data © 2018).

Using the building data shown in figure 4.2, and a pre-set  $d_{thresh}$  value (10 meters in this case), 77 clusters are generated, shown in figure 4.4, where each colour refers to a cluster of buildings. The “buffered cascading union” operation actually corresponds to our third assumption in section 4.2. Note that clustering result can change according to  $d_{thresh}$  (a smaller

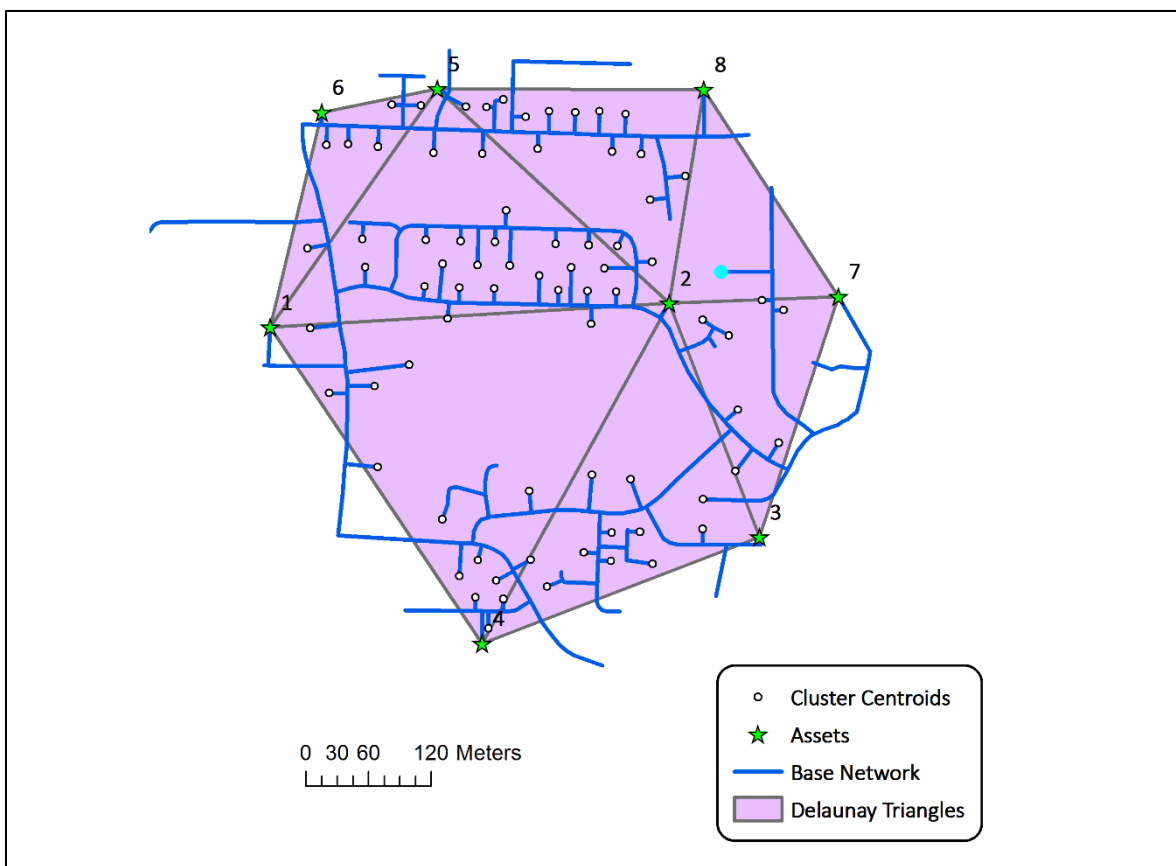
$d_{thesh}$  value result in more clusters being generated). More about parameter sensitivity will be discussed in section 4.6. Afterwards, for each cluster, its centroid (geometric centroid of the multi-polygon) is calculated and extracted for later use.



**Figure 4.5.** Base network generated by connecting clusters and assets (Contains OS data © 2018).

The next step (line 13-21 in listing 4.1) connects each asset and cluster to the road network to generate a network called “base network” (figure 4.5). To achieve this, each asset and cluster (represented by centroid) will find the access point to its nearest roads. In particular, from the asset or cluster centroid, the project point is calculated on the nearest road and that projection point is the access point. The road network is then copied to another network instance called “base network”, to avoid being directly changed. Additional edges are created in the base network to connect each asset/cluster to its access point. Creation of this base network which connects all asset and cluster of buildings, will help fulfil our *second* assumption in section 4.2. The based network is built on the road network and helps to estimate the distance from clusters to each asset.

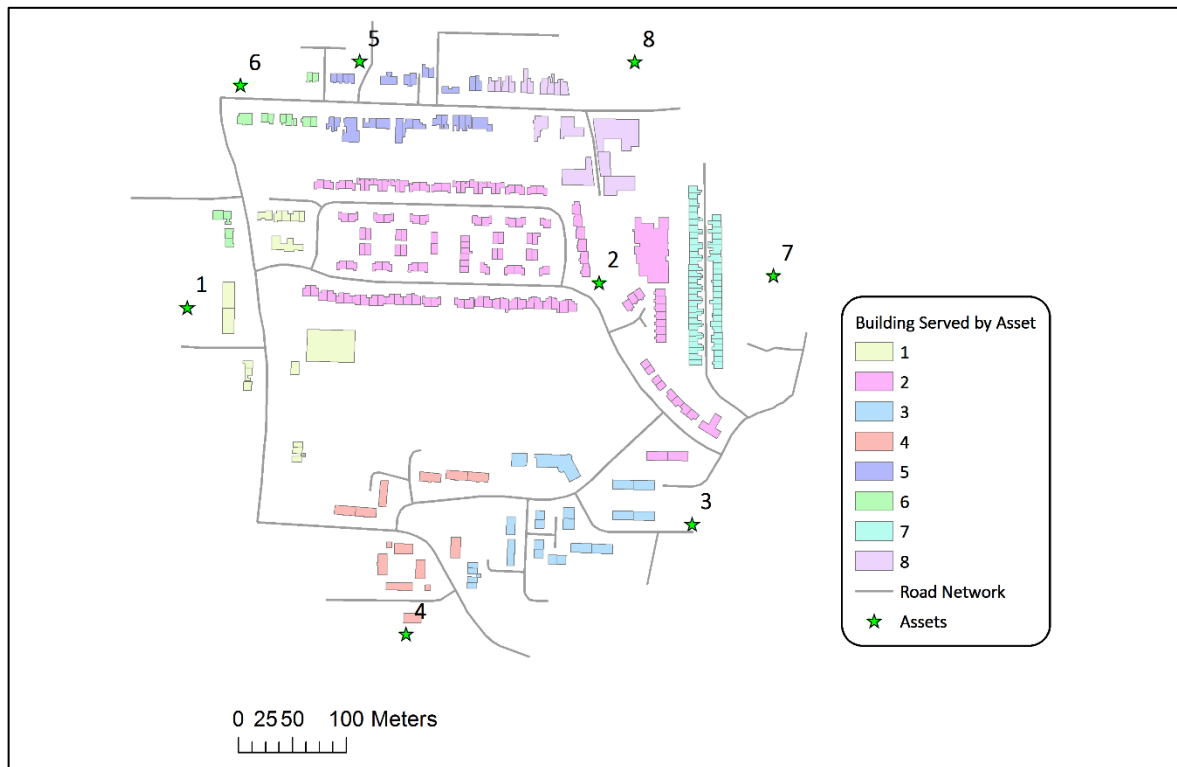
Finally (line 22-30 in listing 4.1), the asset points are used to triangulate (by Delaunay) the entire space, so that each cluster centroid is within one containing triangle (figure 4.6). For each cluster, the three assets forming the vertices of its containing triangle are identified and the asset with the shortest network path distance to the cluster centroid, via the base network, is allocated to that cluster. For example, in figure 4.6, the highlighted cluster centroid will choose from the assets at points 2, 7, 8 to find the nearest one via the base network. For each building belonging to the corresponding cluster, the chosen asset is assigned to it, and this is the function  $f_{assign}: B \rightarrow A$  is generated.



**Figure 4.6.** Delaunay triangulation and assigning an asset to each cluster (Contains OS data © 2018).

The reason to apply Delaunay triangulation is to speed up the algorithm. By default, no triangulation is done, so that each cluster will be assigned a nearest asset (from all the assets in the area) via the base network. This calculation can be very expensive using real city data. Therefore, an assumption is made that “if an asset is close to a cluster via base network, it

must be close in the Euclidean space”. Therefore, for each cluster, triangulation process helps to reduce the amount of shortest path calculations on the base network, and helps to speed up the algorithm. On the other hand, the reason to select “nearest” asset (via base network) to the cluster, is to ensure that if cables are used to connect the asset to the cluster, total length of cable can be kept as short as possible (basic assumption 2 of the algorithm).



**Figure 4.7.** Result of topology generation process (Contains OS data © 2018).

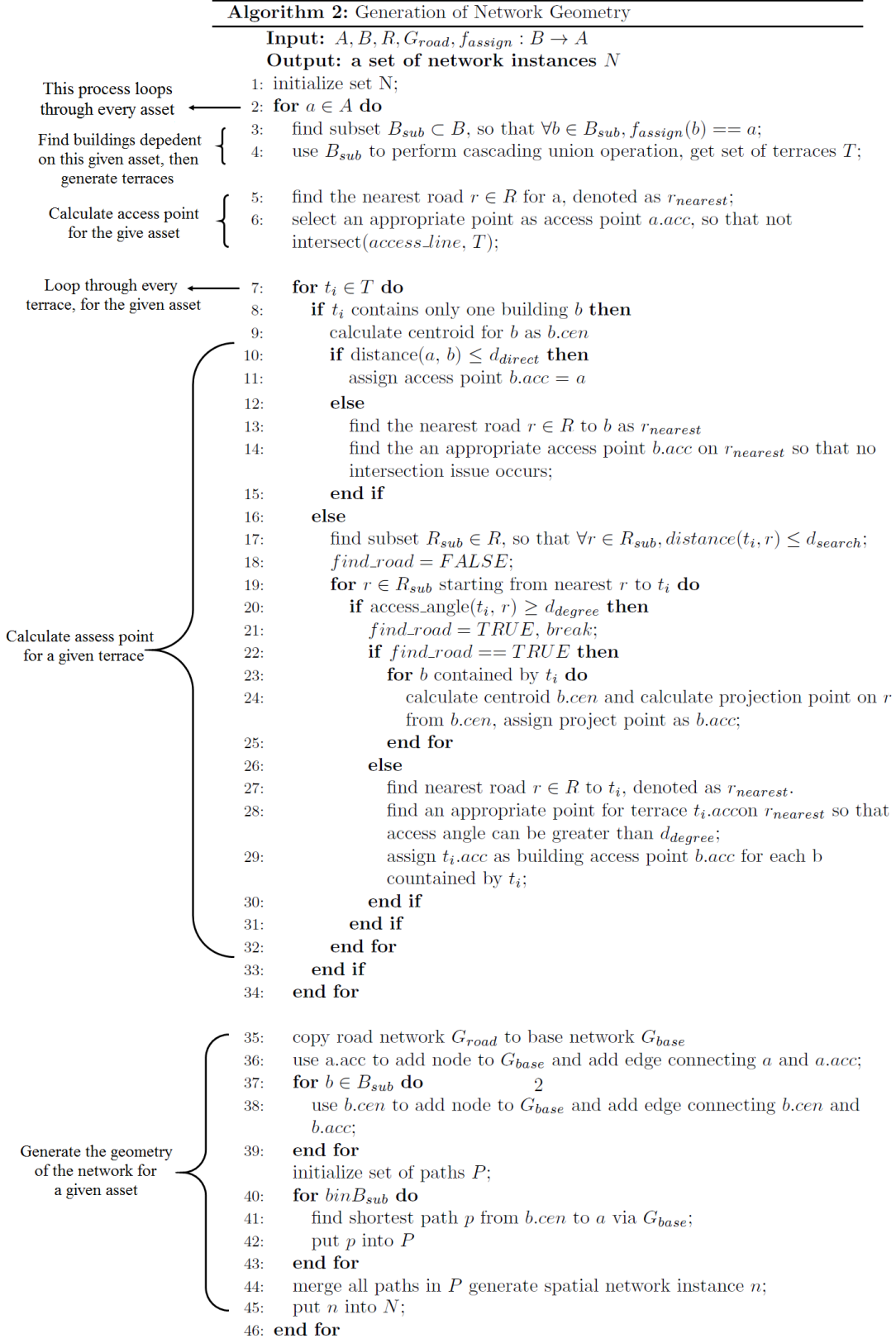
Figure 4.7 shows the final result in this topology generation process, where those buildings assigned the same infrastructure asset are shown in a same colour.

### 4.3.2 Geometry Generation

At this stage, the relationships between the infrastructure assets and individual buildings have been resolved, by the mapping  $f_{assign}: B \rightarrow A$ . The remaining work involves generating the actual spatial network instances that connect each asset and its dependent buildings.

Necessary input includes  $A$  (assets),  $B$  (buildings),  $R$  (roads),  $G_{road}$  (road network), and  $f_{assign}$ .

The expected output is a set  $N$ , which consists of spatial network instances. Each network instance connects an asset and its dependent buildings. Listing 4.2 helps to explain the geometry generation process.

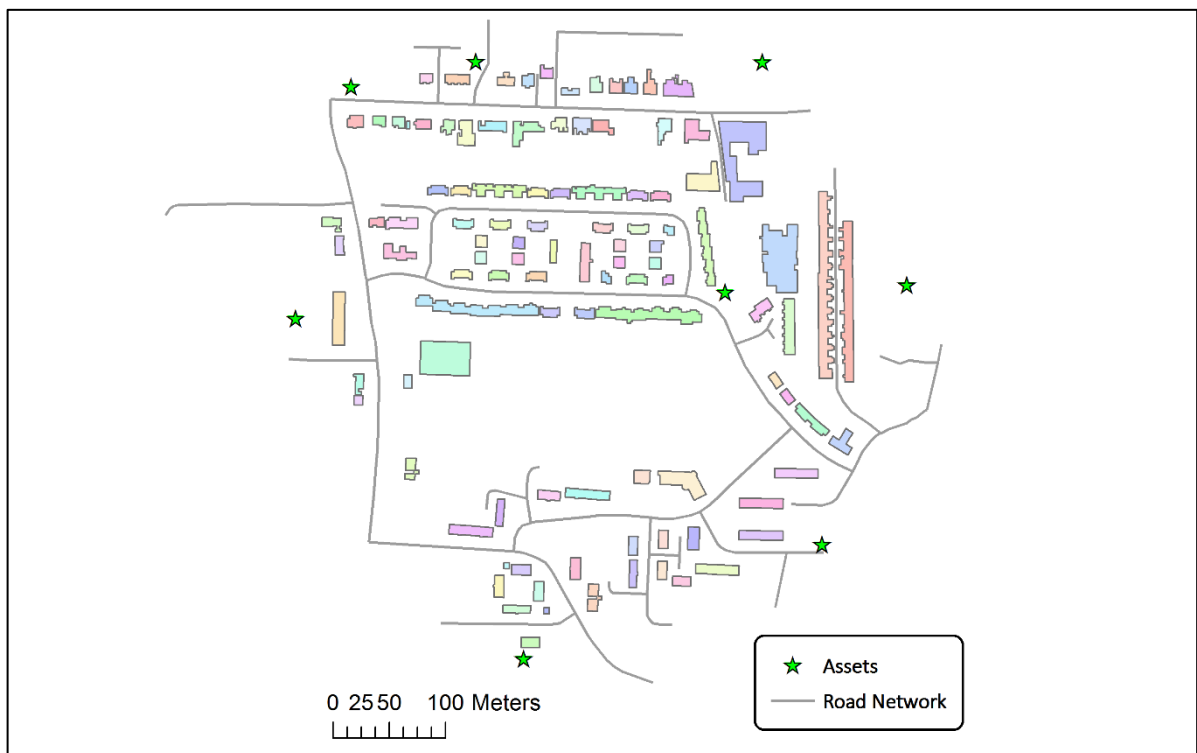


**Listing 4.2.** The pseudo code for topology generation process.



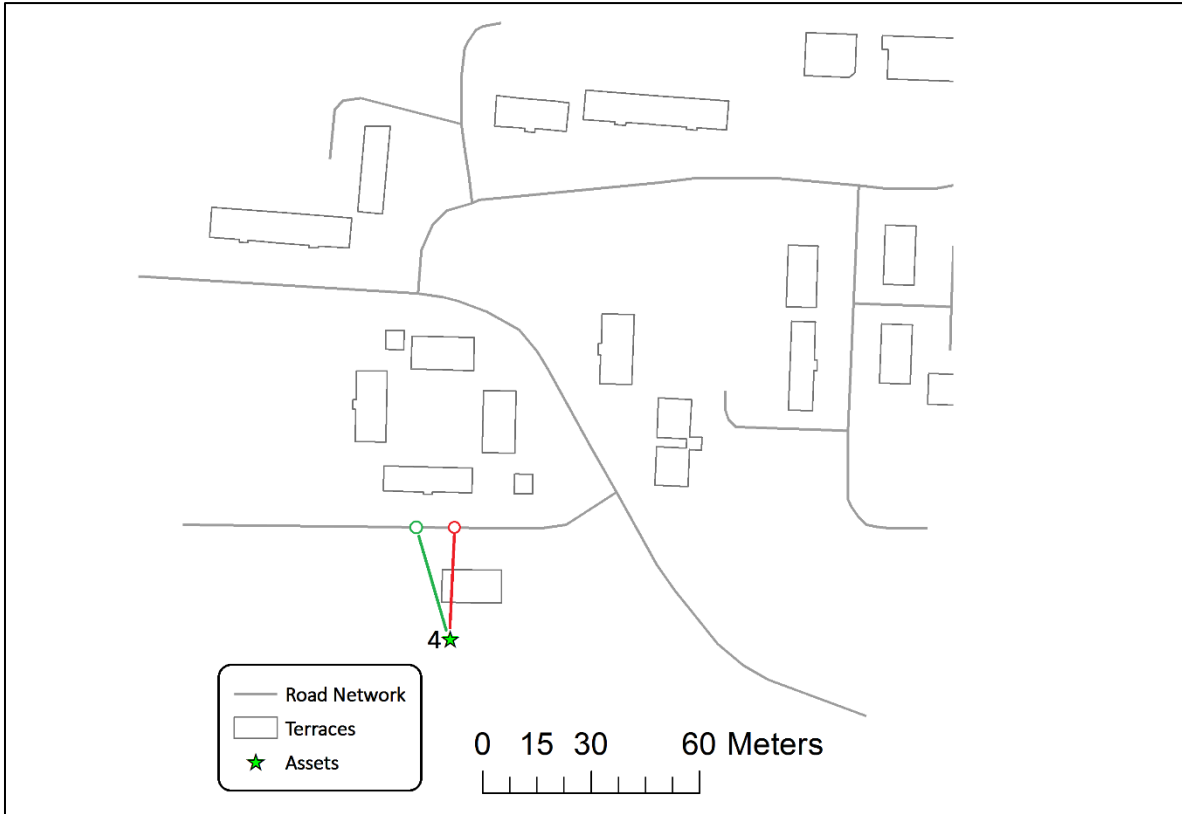
The entire geometry generation process will loop through each asset  $a$ , and generate its distribution network instance. To make things clear, below only the network generation process for a given asset  $a$ , is discussed.

Firstly (line 3-4 in listing 4.2), using  $f_{assign}:B \rightarrow A$ , a subset  $B_{sub}$  of  $B$  is fetched, so that every building  $b$  in  $B_{sub}$  has been assigned to the given asset  $a$ . Then using buildings in  $B_{sub}$ , an operation called “cascaded union” (Shapely, 2018) is performed to group buildings into terraces (buildings topologically connecting one another in a row). The reason for having this process is to make sure better spatial layout of network can be generated (for details, please see figure 4.11 and 4.12). Note in this algorithm, an individual building is allowed to form a terrace itself, if it is not topologically connected to any other building. The set  $T$  is used to denote all the terraces generated, where each  $t$  is a terrace, represented by a polygon. For example, figure 4.8 shows the 107 terraces generated based on the layout of buildings in figure 4.1.



**Figure 4.8.** 107 terraces generated based on buildings shown in figure 4.2 (Contains OS data © 2018).

Then (line 5-6 in listing 4.2), the nearest road  $r_{nearest}$ , will be chosen for the asset  $a$  to generate an access point  $a.acc$  on the  $r_{nearest}$ . The access point will be chosen in a way that the access line does not intersect with any  $t$  in  $T$ . For example, in figure 4.9, the green point can be chosen as the access point while the original project point (red one) cannot.

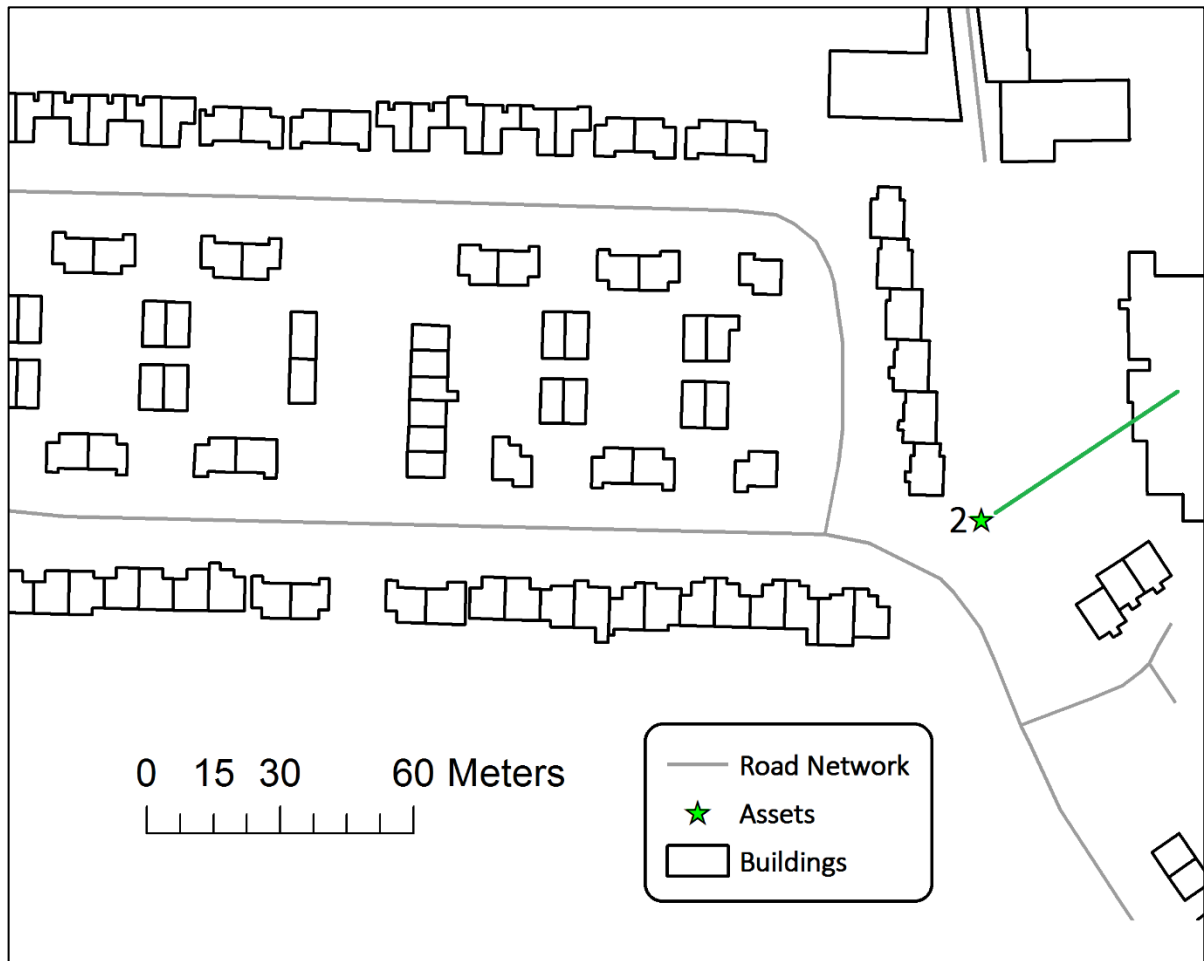


**Figure 4.9.** Access point calculation for the asset (Contains OS data © 2018).

Similarly (line 7-34 in listing 4.2), for each terrace  $t$  in  $T$ , the algorithm calculates the access point for each building within that terrace. The actual workload in this step depends on the number of buildings in that terrace. There are two situations.

**Situation 1** (line 8-15 in listing 4.2) is that  $t$  is a “one building terrace”. First, a check will be done to see if the building is close to the asset (using a threshold distance called  $d_{direct}$ , for example 100 meters), such that the asset can be connected directly to the building, without using a road access point (green line figure 4.10). Otherwise, the building will choose the nearest road to generate an appropriate access point called  $b.acc$ , which causes no intersection issue (using the same approach as generating an access point for an asset). The reason to allow direct access from a building to an asset is that this helps to further shorten the total

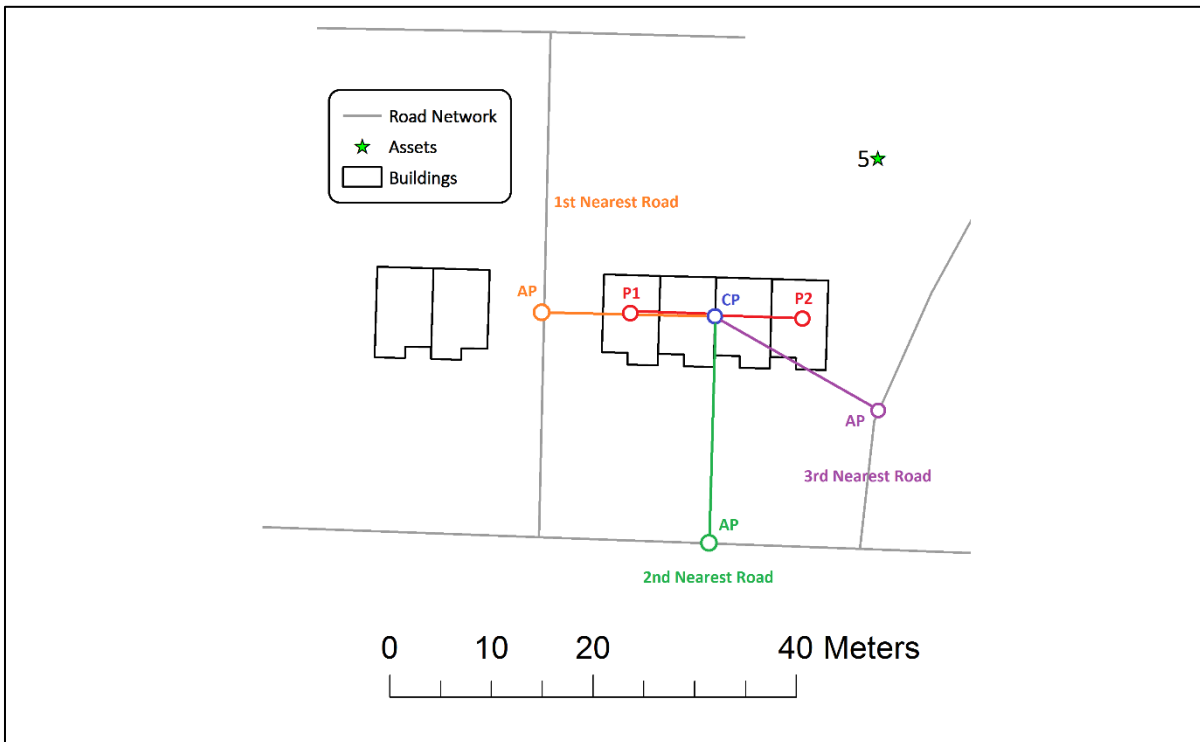
length of distribution network generated.



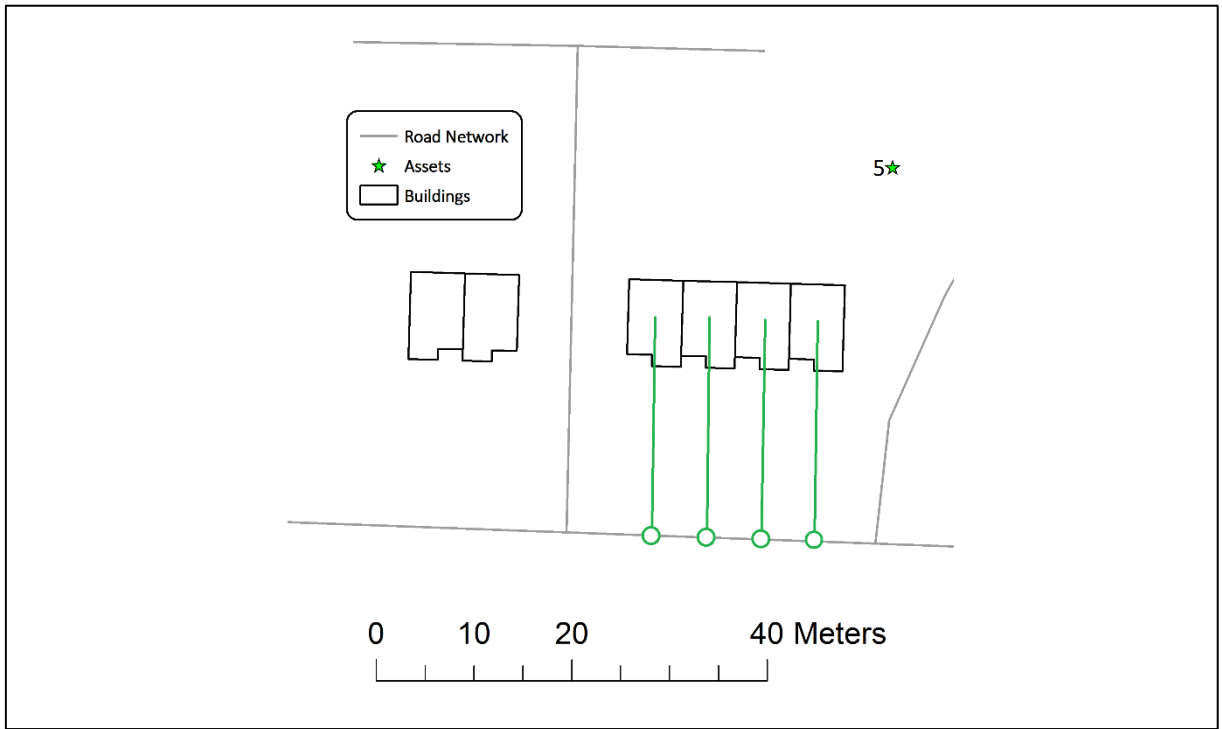
**Figure 4.10.** A single-building terrace can directly access an asset if close enough (Contains OS data © 2018).

**Situation 2** (line 16-34 in listing 4.2) is that the  $t$  is a normal terrace (contains at least two buildings). Using a parameter search distance  $d_{search}$  (for example, 100 meters), all the roads within the search distance to the  $t$  will be checked (starting from the nearest road), if the access angle is large enough (using a pre-set parameter called  $\beta$ , for example  $45^\circ$ ) for the terrace  $t$  to access that road. The access angle is defined as the acute intersection angle between the access line (perpendicular access) and feature line of terrace  $t$ . The feature line is defined as the line connecting centroids of the two buildings in  $t$  which are most distant from each other. In figure 4.11, access line is the line CP-AP and the feature line is line P1-P2. The access angle for the 2<sup>nd</sup> nearest road is the largest, compared with the 1<sup>st</sup> and 3<sup>rd</sup> nearest road. If there is such a road segment within  $d_{search}$ , which corresponds to an access angle large

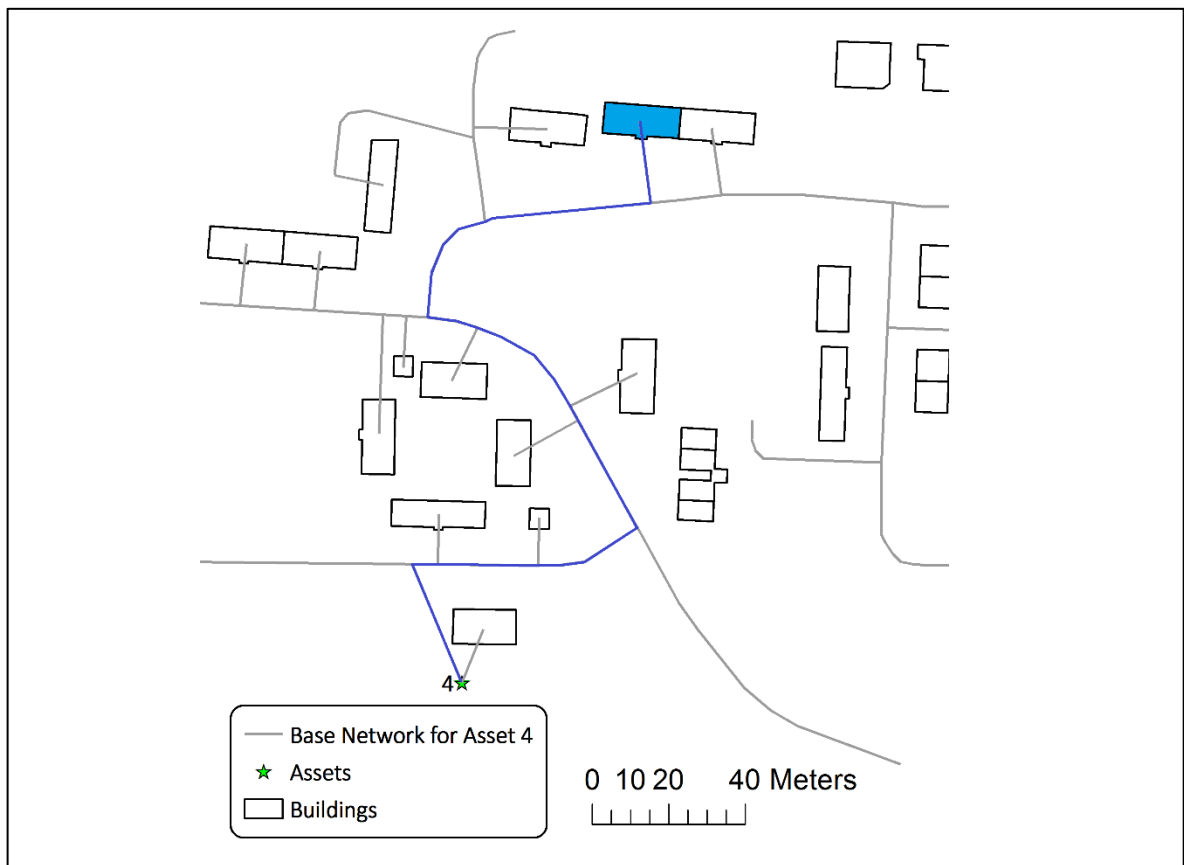
enough, then all the buildings with the terrace  $t$  will access that road in a perpendicular approach, showed as green points in figure 4.12. If not, the nearest road to  $t$  is still chosen, and the access point is chosen to ensure now the access angle is at least as large as  $\beta$ . The reason to set up  $d_{search}$  and  $\beta$  is to make sure that all buildings within the terrace can access a nearby road as perpendicularly as possible.



**Figure 4.11.** Explanation of the access angle for a terrace (Contains OS data © 2018).



**Figure 4.12.** Pick up the access point for each building within the terrace (Contains OS data © 2018).

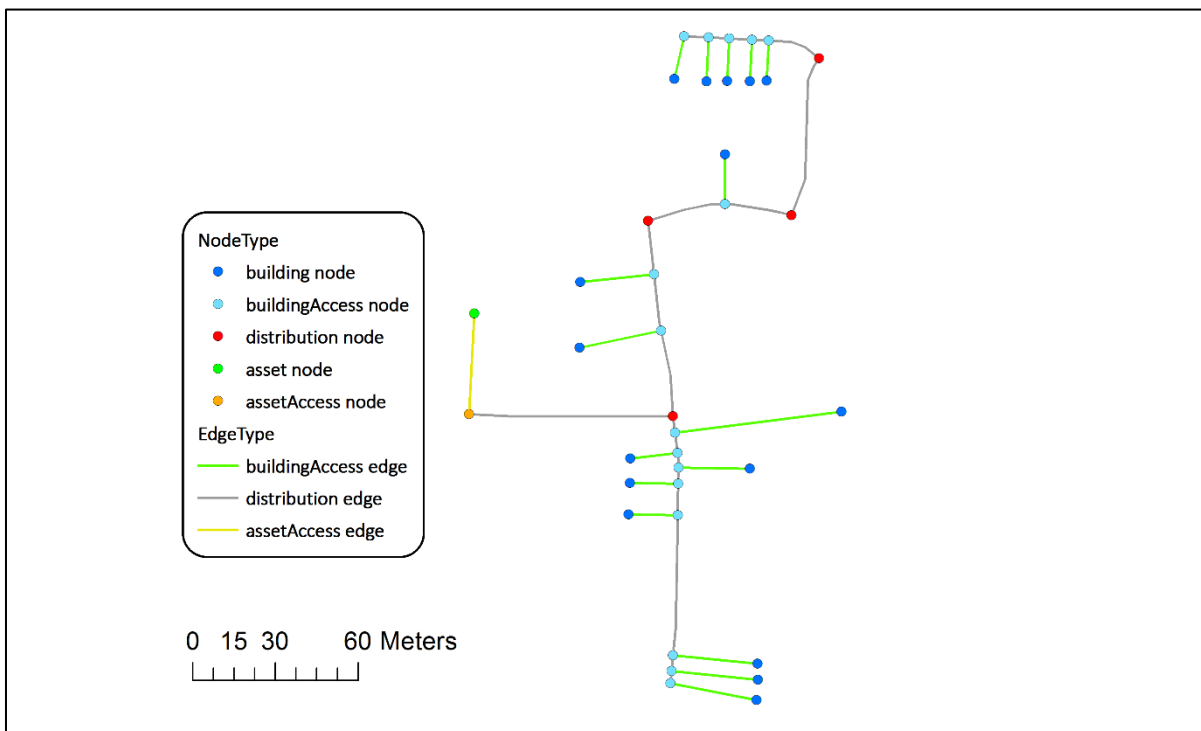


**Figure 4.13.** For asset 4, calculate the shortest path from the asset to a building (Contains OS data © 2018).

Finally, using all the buildings, the assets, and all their access points, a base network is generated to them (using road network). Note this base network is not the same as the one in the topology generation process (where **all assets** and **all clusters** are connected). In here, in the geometry generation process, for each asset, the base network only connects **this asset** and **all its dependent buildings**.

For each building, a shortest path is calculated via the base network to the asset (blue path shown in figure 4.13). A spatial network instance can be generated by merging all these paths, and this network is actually the specific infrastructure network connecting this asset *a* and all its dependent buildings. Moreover, each network instance is actually an *acyclic graph* (no loop inside), the flow direction can be easily resolved from the asset node to each building node.

Within an infrastructure network, there are different types of nodes and edges, shown in figure 4.14. The naming of different types of nodes and edges is based on topological connectivity and is explain in table 4.2.



**Figure 4.14.** Different types of nodes and edges in a distribution network.

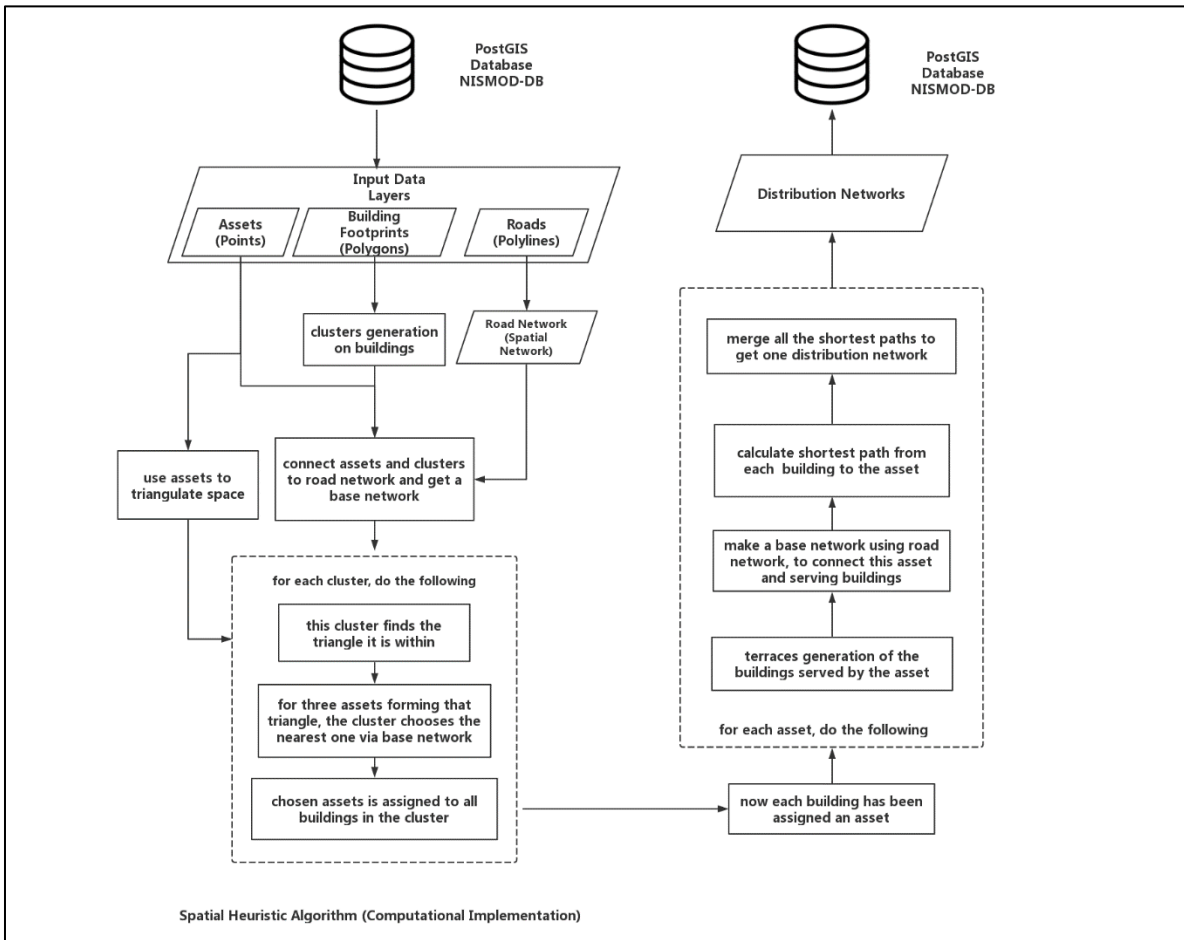
<b>Type of node or edge</b>	<b>Description</b>
building node	A node that represents an individual building.
asset node	A node that represents an asset.
buildingAccess node	A node that directly connects a building node.
assetAccess node	A node that directly connects an asset node.
distribution node	A node that is not a building node, asset node, buildingAccess node, or assetAccess node.
buildingAccess edge	An edge connecting a building node and a buildingAccess node.
assetAccess edge	An edge connecting an asset node and an assetAccess node.
distribution edge	An edge that is not a buildingAccess edge or an accessAccess edge.

**Table 4.2.** Description of different types and edges and nodes in a distribution network.

As a result, the geometry generation process generates multiple distribution networks (one for each asset), and figure 4.3 shows the 8 distribution networks generated for the example area.

#### **4.4 Algorithm Implementation**

The spatial heuristic algorithm is developed and implemented in Python using the NetworkX package (NetworkX, 2014) for manipulation and analysis of complex networks, and the Shapely package for performing geometry calculation on spatial objects. The algorithm employs the Infrastructure Transitions Research Consortium (ITRC) PostgreSQL/PostGIS National Infrastructure Systems Modelling Database (NISMOD-DB), for primary data extraction via a Python binding, and generates network models that are written back to NISMOD-DB in the form of an instance of the ITRC interdependent network database schema reported by Barr et al.(2013). The implementation of the heuristic algorithm is shown in figure 4.15.

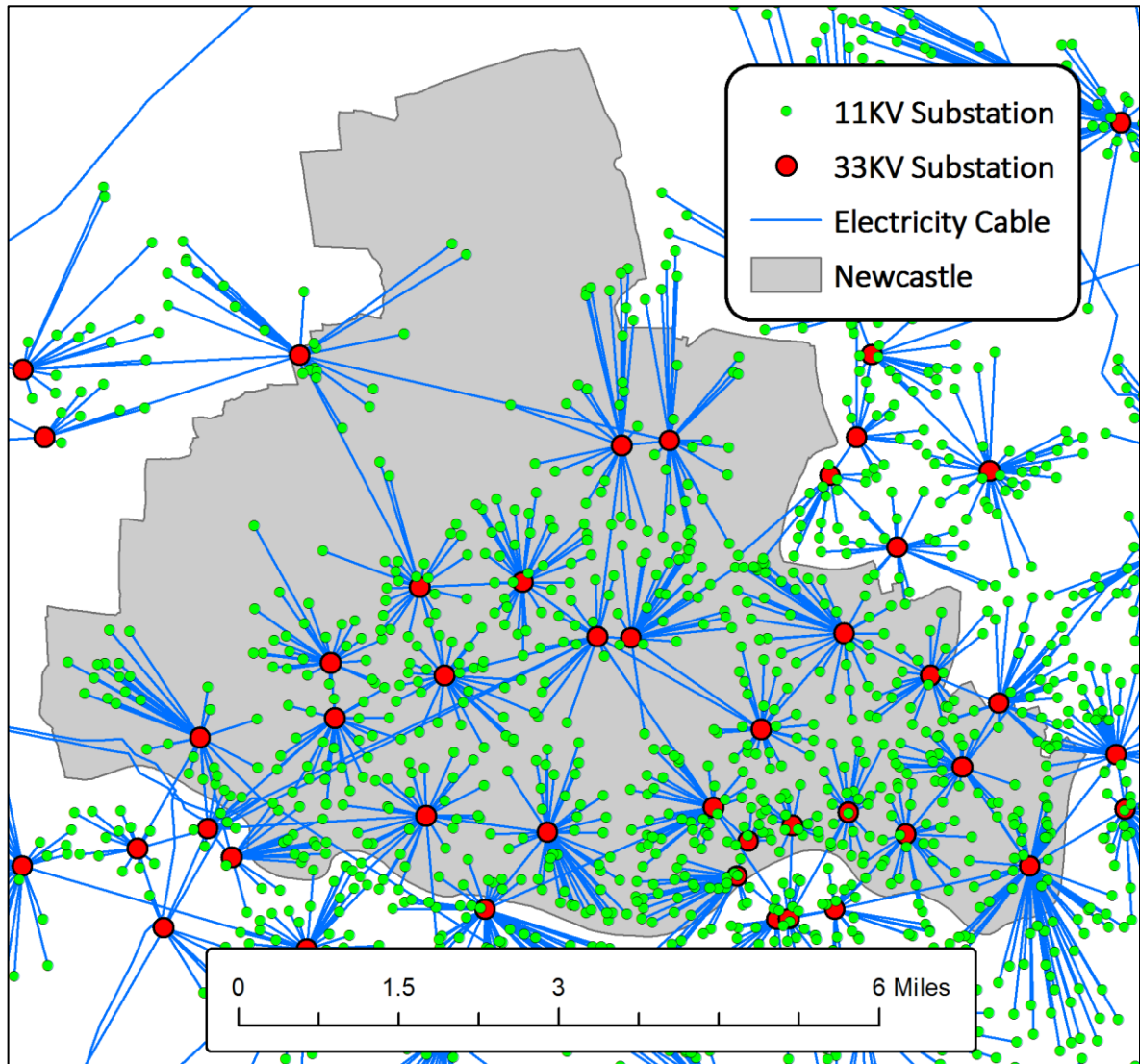


**Figure 4.15.** Computational implementation of the spatial heuristic algorithm.

#### 4.5 Pilot Study (Newcastle upon Tyne)

To demonstrate the utility and applicability of the spatial heuristic algorithm at the city scale, it was applied to generate the local electricity distribution networks for Newcastle upon Tyne (a city of approximately 282,300 people covering an area of 112 km<sup>2</sup>, the most populous city in North East England). The infrastructure networks to be generated in this pilot study are electricity distribution networks. The assets in this case are the electricity substations, which send electricity to each individual building via a cable-based network. Electricity is normally generated from generation plants and pressurized to 400kv via the electricity transmission network. When electricity is transmitted to the urban areas, electricity voltage will be decreased by substations of different levels (132kv, 66kv, 33kv, and 11kv). The lowest level substations are 11kv ones, and they are connected with the 33kv substations (figure 4.16).

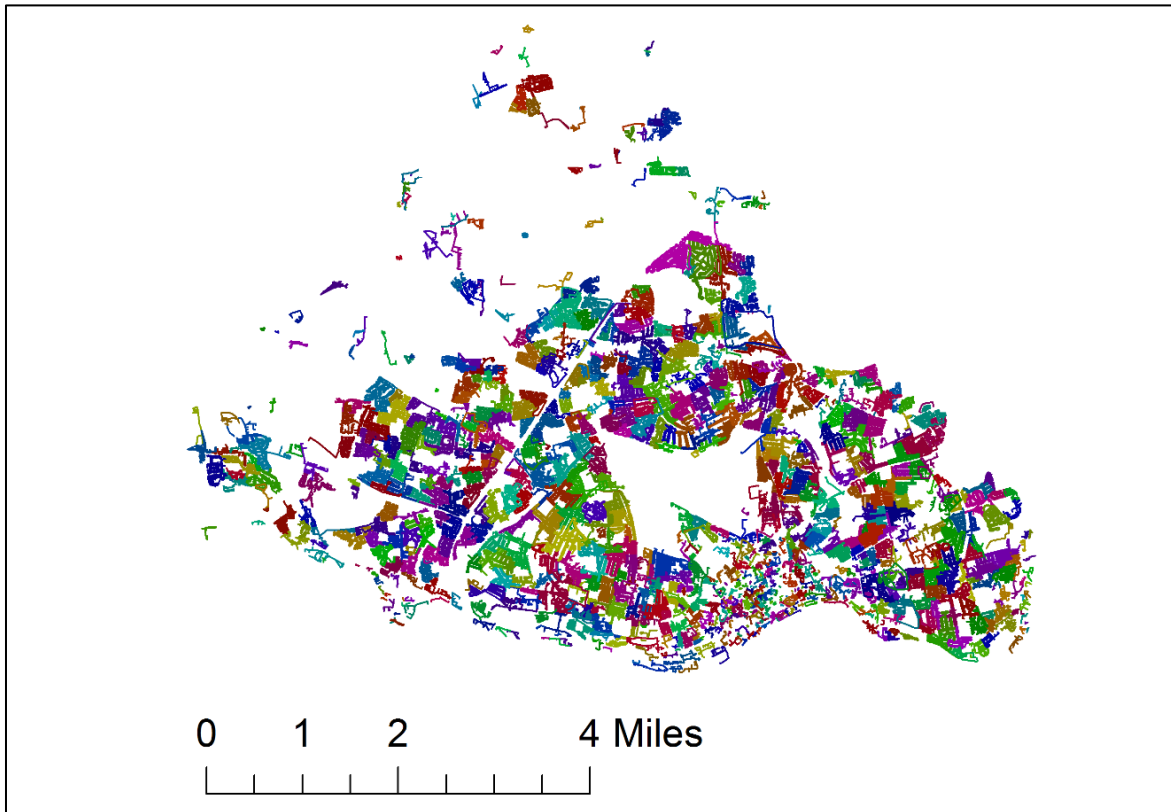




**Figure 4.16.** Electricity transmission networks connecting 11kv and 33kv substations in Newcastle upon Tyne (Robson, 2017).

While in this pilot study, the focus is electricity distribution networks, and the input infrastructure assets comprised all 636 11kv electricity substations (lowest levels), identified from the Ordnance Survey Point of Interest Layer (Ordnance Survey, 2018). The road network was obtained using Ordnance Survey Integrated Transport Network (ITN) Layer (Ordnance Survey, 2018). Building footprints were obtained by filtering (select only building feature) the Ordnance Survey MasterMap topography Layer (Ordnance Survey, 2018). Initially, 142,763 buildings were extracted. Then only buildings with at least 30 m<sup>2</sup> area were kept, since smaller buildings are considered to be buildings that do not require infrastructure services (Barr, et al., 2017). In the end, 104,855 buildings were left and used for generating

electricity distribution networks.



**Figure 4.17.** Generated synthetic electricity distribution networks in Newcastle upon Tyne.

Figure 4.17 shows the synthetic distribution networks that were generated for the entire Newcastle upon Tyne city area, separately coloured for each single distribution network. For the complete area, a total of 104,855 buildings were processed, creating 636 new local electricity distribution networks, each serving, on average 164 buildings. The total number of edges and nodes (of any type) generated are 209,892 and 209,886, respectively. Each distribution network has on average 330 edges and 330 nodes. The total length of network edges are 2,807,478 meters.

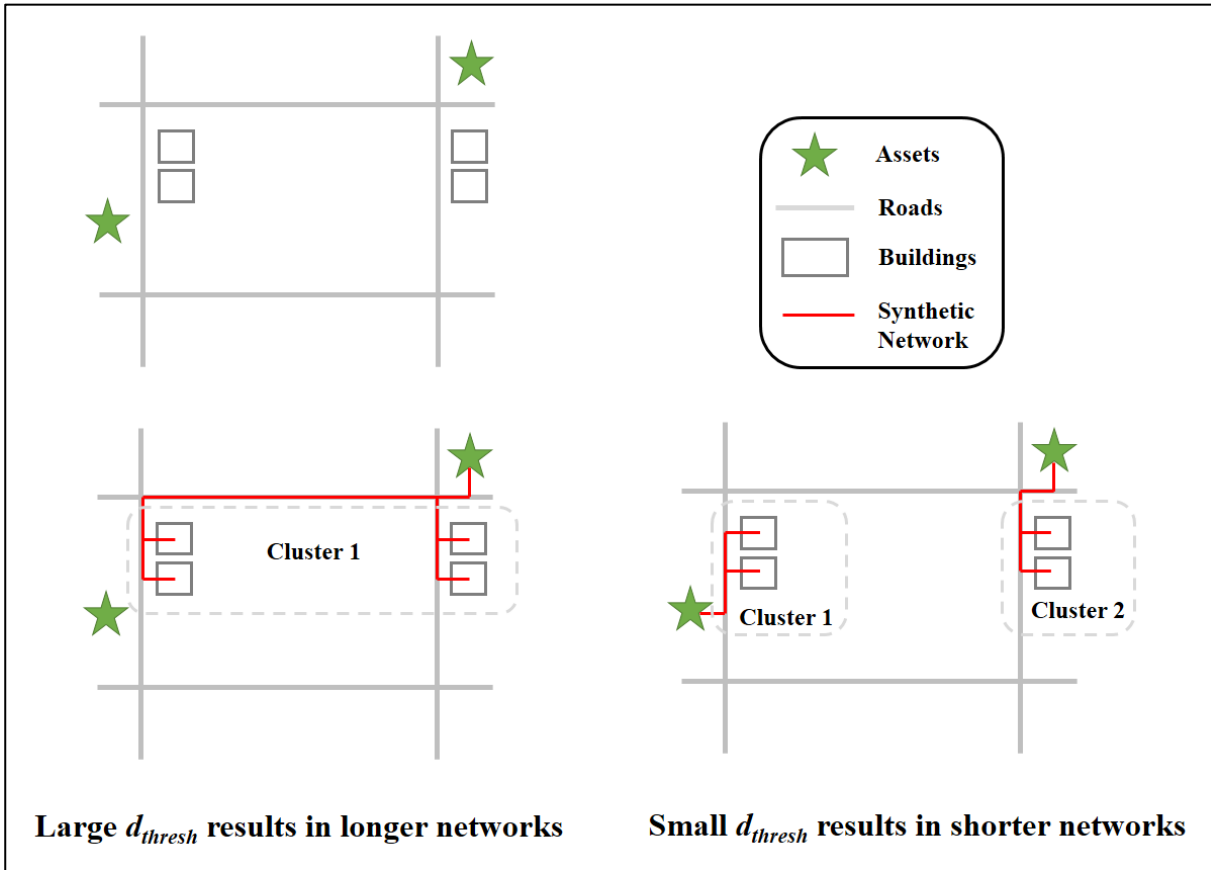
It is important to point out, that threshold distance  $d_{thresh}$  is an essential parameter in the algorithm (as mentioned in section 4.3). Different  $d_{thresh}$  values will result in different number of clusters being generated, and thus can affect the characteristics of the distribution networks to be generated. In this pilot study,  $d_{thresh}$  was set to be 10 meters. To justify the choice of this value, a parameter sensitivity test was also done, using different  $d_{thresh}$  values (5, 10, 15, 20, 25 meters), and result is shown in table 4.3.

$d_{thresh}$ (meters)	Number of Cluster Generated	Number of Networks Generated	Algorithm Running Time (hours)	Network Total Length (meters)
5	15164 (= 196 %)	636 (= 100%)	4.04 (= 178%)	2796248 (= 99.6%)
<b>10</b>	<b>7719 (= 100%)</b>	<b>636 (= 100%)</b>	<b>2.27 (= 100%)</b>	<b>2807478 (= 100%)</b>
15	4300 (= 56 %)	631 (= 99.3%)	1.77 (= 78%)	2902932 (= 103 %)
20	1905 (= 25%)	607 (= 96.0%)	1.45 (= 63%)	3054536 (= 108%)
25	895 (= 11%)	514 (= 80.8%)	1.12 (= 48%)	3214562 (= 115%)

**Table 4.3.** Sensitivity of parameter  $d_{thresh}$ .

In table 4.3, it is clear that as  $d_{thresh}$  increases gradually, number of clusters generated will drop significantly. That will further result in fewer distribution networks generated. For example, when setting  $d_{thresh}$  to be 25 meters, only 80.8% of the input substation points were used to generate the distribution networks. That low ratio is considered to indicate potentially inaccurate result, because all the 636 substation points are 11kv substations so each of them should connect buildings. Therefore, a good  $d_{thresh}$  value should result in a high ratio.

On the other hand, the total length of synthetic networks should be kept as small as possible (since that is the algorithm assumption). As  $d_{thresh}$  increases, network total length will increase, because with fewer clusters being generated, each cluster contains more buildings. When assigning a substation to each building (in the topology generation process), each building is represented by the geometric centroid of the cluster (could be very far away from buildings within the cluster), and therefore the cable connecting a building to its dependent substation can be longer in this case (simple example in figure 4.18).



**Figure 4.18.** Synthetic network result may change depending on different  $d_{thresh}$  value.

Following this logic, it is considered that 5 meters and 10 meters are the good values for  $d_{thresh}$ . When using 5 meters, 78% more processing time is needed. That is because with more clusters generated, it is computationally more expensive to connect each cluster centroid to the road network to generate the base network (in the topology generation process).

Newcastle is a small city, and algorithm running time difference will be more relevant when processing data from much larger city (such as London). That is why 10 meters is finally chosen as a good value for  $d_{thresh}$ .

In the pilot study, the Delaunay triangulation process was also applied, as mentioned in section 4.3. Without that process, the algorithm will be much slower, as shown in table 4.4.

Triangulation Process	Number of Networks Generated	Algorithm Running Time (hours)	Network Total Length (meters)
Not Applied	636 (= 100%)	30.86 (= 1360%)	2802985 (= 99.84%)
Applied	636 (= 100%)	2.27 (= 100%)	2807478 (= 100 %)

**Table 4.4.** Effect of applying Delaunay triangulation process.

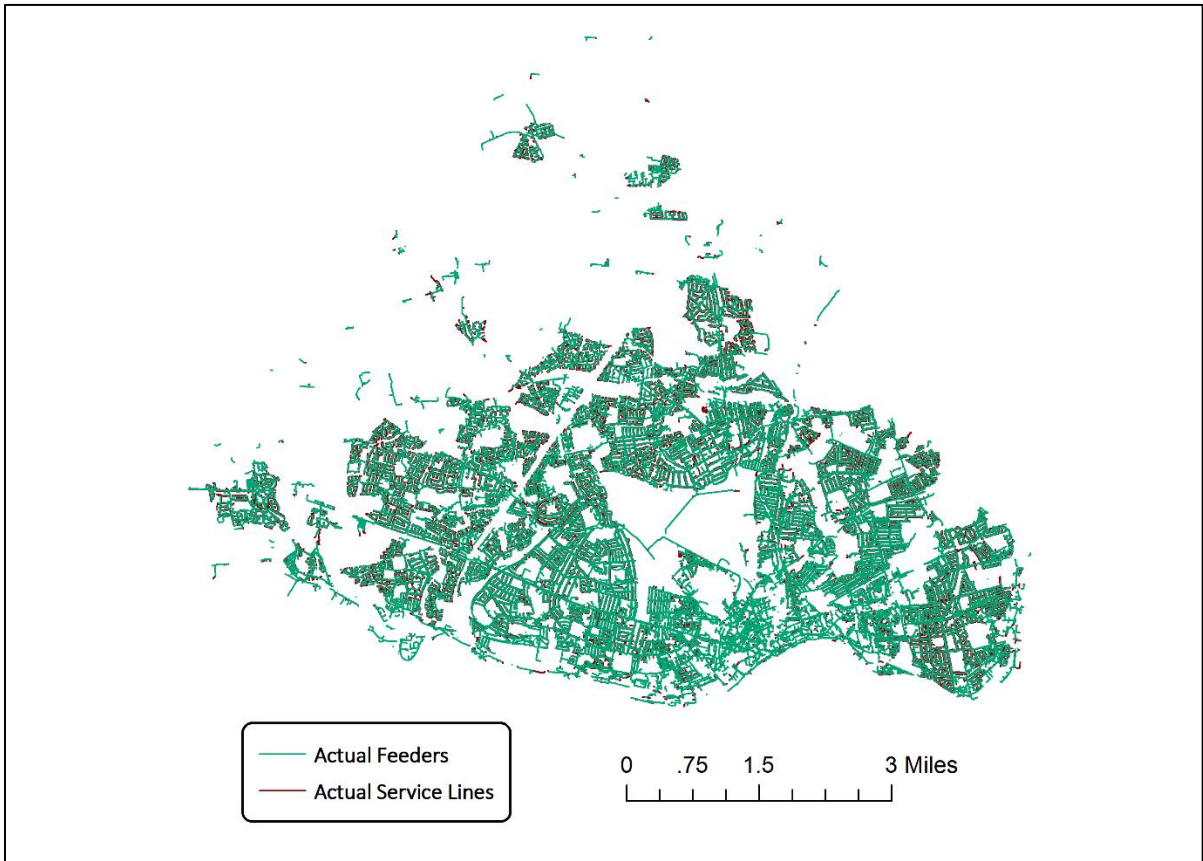
In table 4.4, it is clear that if not applying the triangulation process, algorithm will be 13.6 times slower. That is because the triangulation process greatly reduces the amount of shortest path calculations that need to be resolved. That can be a big problem when processing large city data. The triangulation process helps to save a great amount of time but does not cause big difference in the synthetic network result, which is why this process is applied.

#### 4.6 Synthetic Network Validation

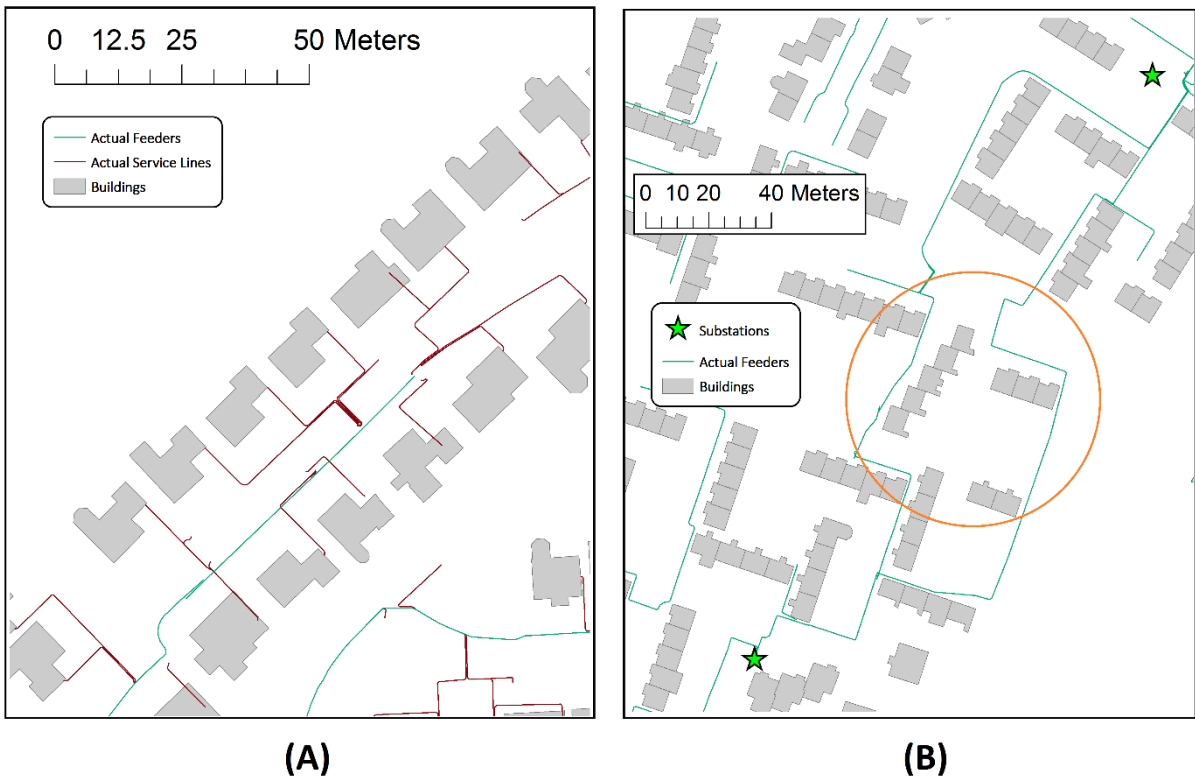
When generating synthetic networks heuristically, the biggest concern is the data quality, or how well the synthetic network represents the real network. Northern Power Grid (NPG) utility company provided the actual layout of the distribution networks in Newcastle upon Tyne, the *best real data* for validation. The actual data is a polyline shapefile file, in which the spatial layout of the electricity cables is stored.

NPG has labelled all the cables to be either of type “service line” or “feeder” (figure 4.19). The “service line” refers to the cable directly connected with a building, and corresponds to the “buildingAccess edge” in the synthetic network model (figure 4.14). The “feeder” refers to any other cables, corresponding to all the other types of edge in the synthetic network. To avoid confusion during validation, the edges in the synthetic networks of type “assetAccess edge” and “distribution edge” are termed “Synthetic Feeder”, and the edges of type “buildingAccess edge” are termed “Synthetic Service Line”.

Close observation on the NPG data reveals one issue. Network topology cannot be derived from spatial connectivity in the data, because cables (polylines) often disjoint or intersect when they should have touched each other (figure 4.20 (A)). More importantly, to date no information is available on its partitioning *in terms of different distribution network instances*. As NPG data is one large network data with inaccurate connectivity, it is not possible to *infer the partition* due to the lack of the *expected boundary (gap)* between two distribution network instances (figure 4.20 (B)), which can be obvious in synthetic networks (figure 4.3). There is also no information on the *dependent substation for each actual feeder and/or service line*.



**Figure 4.19.** NPG data of electricity distribution networks in Newcastle upon Tyne (Contains NPG data © 2018).



**Figure 4.20.** (A). Difficulty in retrieving topology from NPG data. (B). Difficulty in retrieving expected network instance boundary (orange circle) (Contains NPG data © 2018).

Considering these limitations, it is not feasible to understand (or infer) the building-substation dependency from the actual data (and thus validate). However, spatial validation is still possible on the synthetic networks with regards to their spatial proximity to actual data. There will be two validations in this section: validation on the feeders, and validation on the service lines. The reason to do separate validations on feeders and service lines is because they have different topological connectivity in electricity distribution networks (a service line connects a building and a feeder does not). Each of the validations will be explained into details below.

#### 4.6.1 Validations on Feeders

Level of spatial proximity should be measured when validating the synthetic feeders against the actual feeders. But before even defining spatial proximity in this situation, one measurement needs to be done first, to validate one of our basic assumptions of the algorithm.

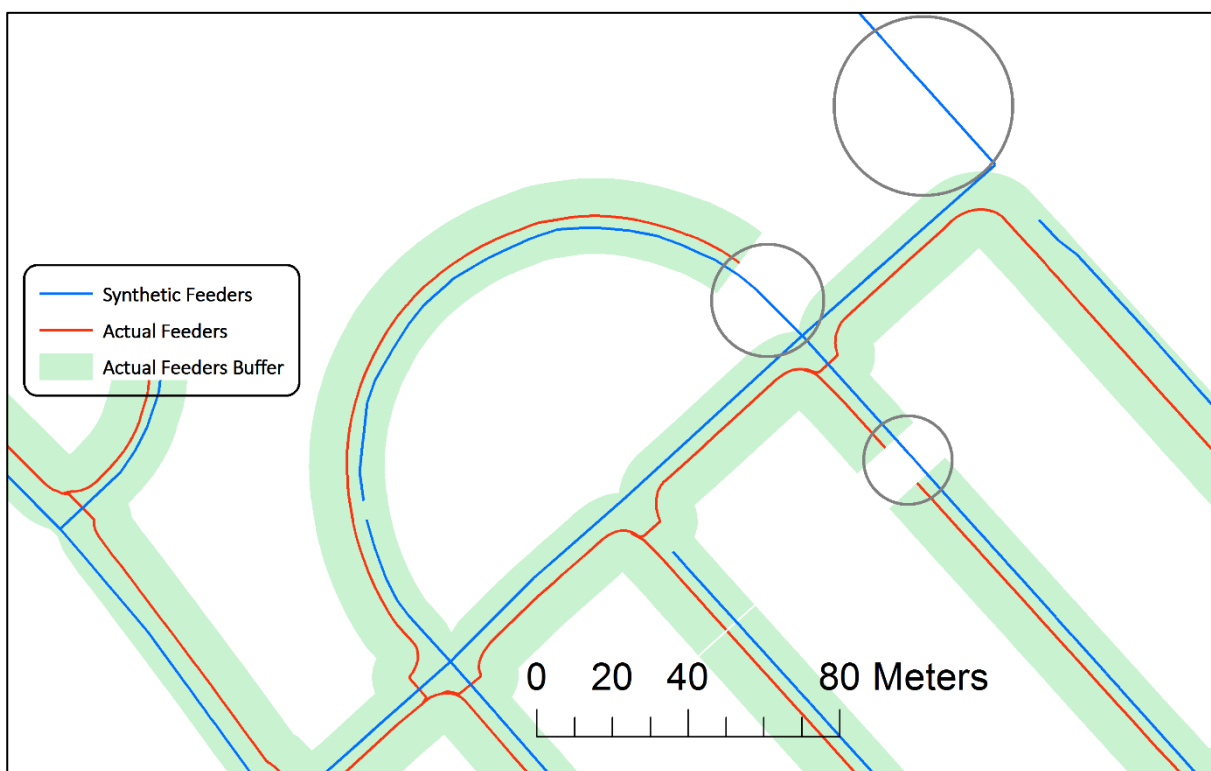


**Figure 4.21.** Location of actual feeders and synthetic feeders, with regards to roads (Contains OS and NPG data © 2018).

The algorithm assumes that infrastructure networks should be paved along the road network.

In the pilot study (electricity distribution networks), that means the feeders should follow the road network. To be precise, the road network (ITN) as the input of the algorithm, is based on the road centrelines, and therefore the synthetic feeders always follows the road centrelines. But the actual data show the actual feeders normally follow one side of the road, and this situation is shown in figure 4.21. In figure 4.21, the road polygon layer represents the actual space occupied by roads. By measurement, it is found that **92%** of the total length of the actual feeders fall within the road polygon, which means our basic assumption of the algorithm is generally correct.

Meanwhile, figure 4.21 indicates how to validate the synthetic feeders against the actual feeders. In GIS data validation, errors of omission and errors of commission are the two most common measurements (Weng, 2010). In this validation, errors of omission refer to the error of this algorithm to not generate feeders where it should have, while errors of commission refer to the error of our algorithm to generate feeders where it should not have. These two errors are used to measure the spatial proximity between the synthetic and actual feeders.

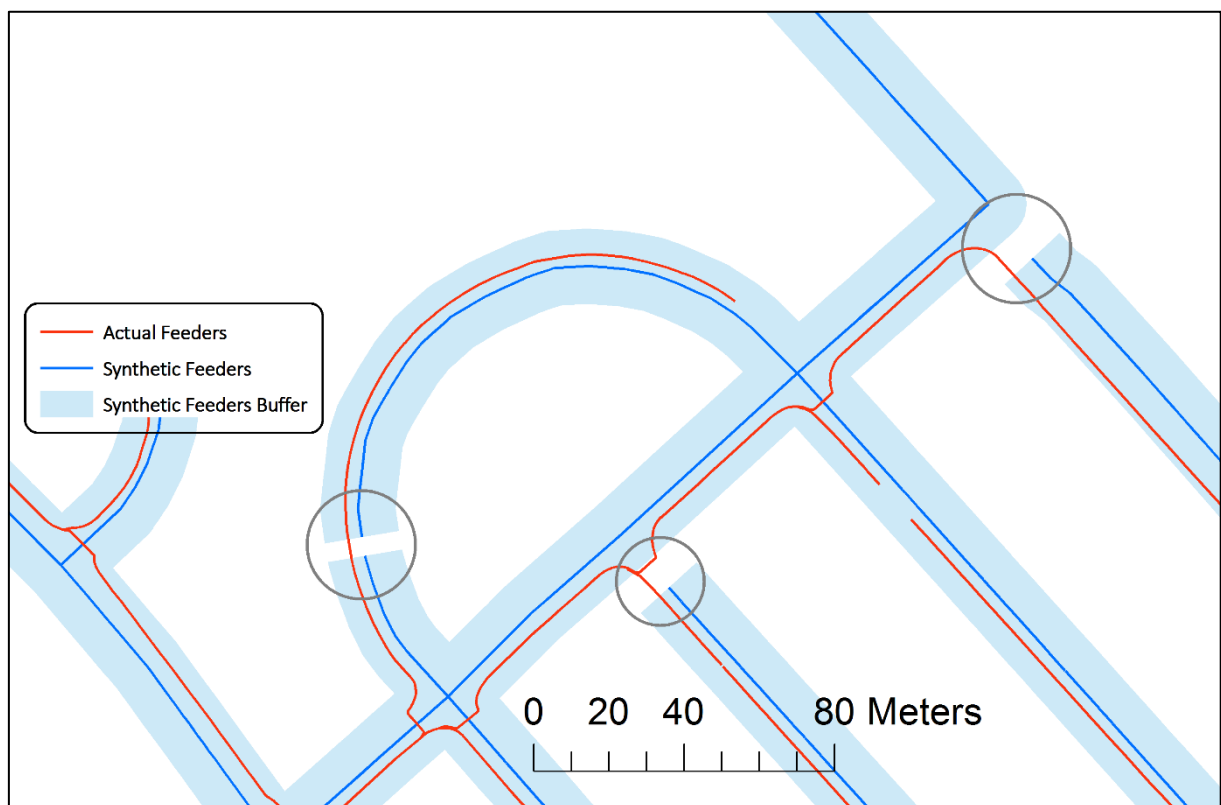


**Figure 4.22.** Errors of commissions (grey circles) (Contains OS and NPG data © 2018).

For calculating the error of commissions, a buffer (buffer on both sides) is be generated for



the actual feeders (figure 4.22). The buffer distance is 10 meters. The reason to use this value is that, in the UK, a single lane width should be 3.65 meters (Newcastle City Council, 2011). For a dual carriage way with two lanes in each direction, the distance from centrelines to the side of the road should be 7.3 meters. A slightly larger value (10 meters) is used for that as it considers possible presence of bicycle lane and median strip. The buffer type is flat-end (note the cut-off at the end of the actual feeders). Similarly, to calculate the error of commissions, a buffer (distance is 10 meters) is generated on both sides of the synthetic feeders (figure 4.23).



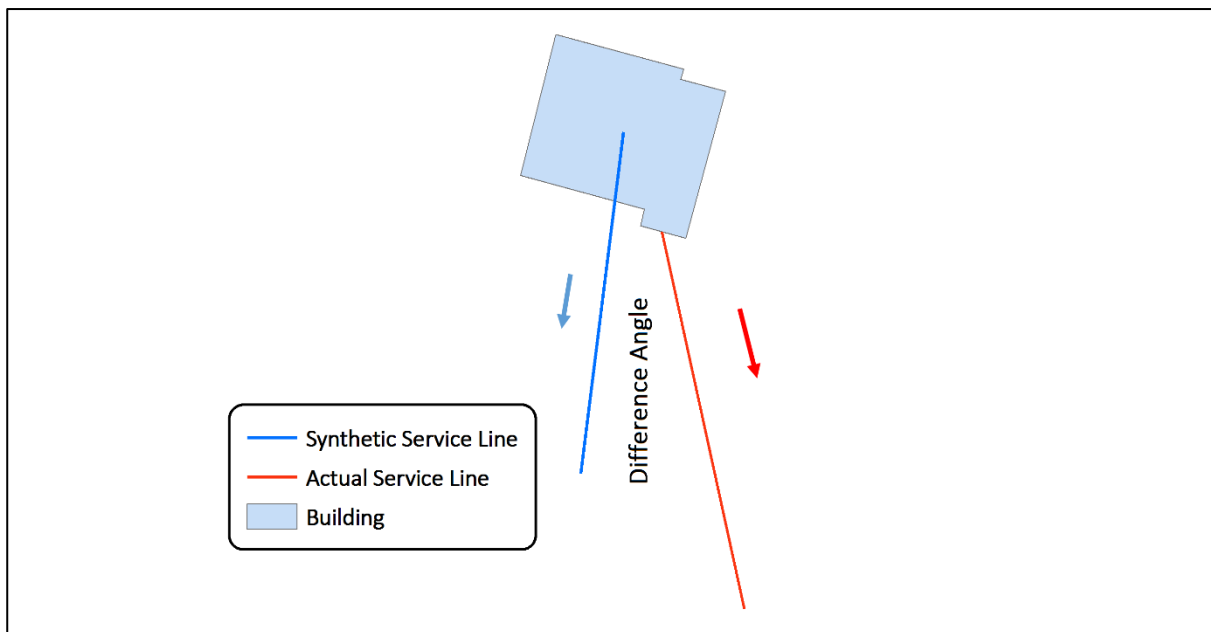
**Figure 4.23.** Error of omissions (grey circles) (Contains OS and NPG data © 2018).

By measurement, it is found that, in the entire city, **86%** of the actual feeders (total length) are within the buffer of synthetic feeders and **89%** of the synthetic feeders (total length) are within the buffer of actual feeders. Based on these two values, it is argued consider the level of spatial proximity between the actual and synthetic feeders is high.

#### **4.6.2 Validation on Service Lines**

To validate the service lines, the errors of commission and omission defined above are not

used here. That is because the service lines do not follow the road network. Instead, the validation relies on a difference angle, which is defined in figure 4.24 to show the intersection angle in a (actual service line, synthetic service line) pair, where both lines serve the same building. Note that a service line is considered to be directional (direction from building), so that the difference angle can be between  $0^\circ$  and  $180^\circ$ . For each building in the city, the difference angle is calculated (where data on the actual service line exists), and a histogram is generated (figure 4.25).

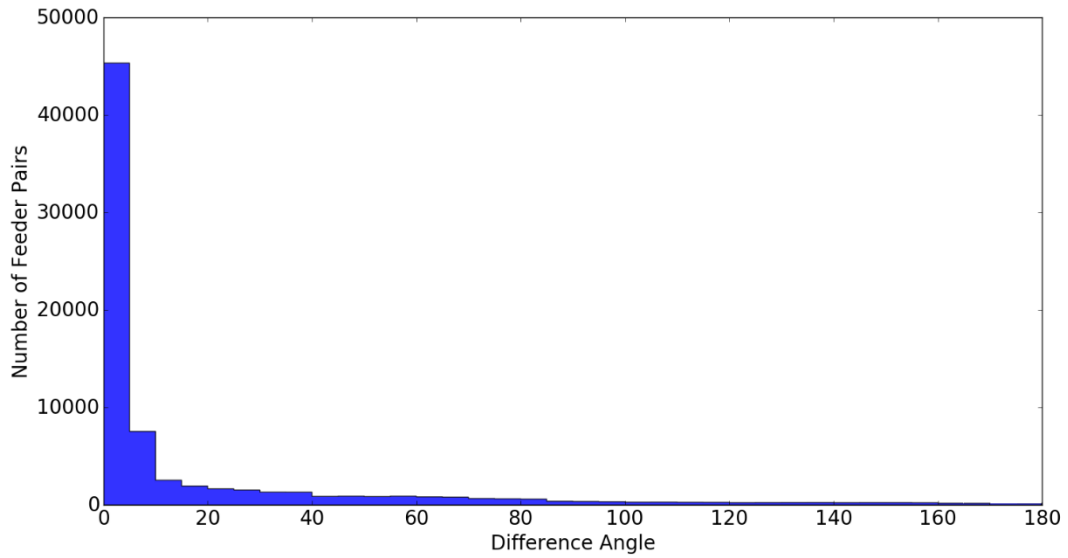


**Figure 4.24.** Definition of difference angle (Contains OS and NPG data © 2018).

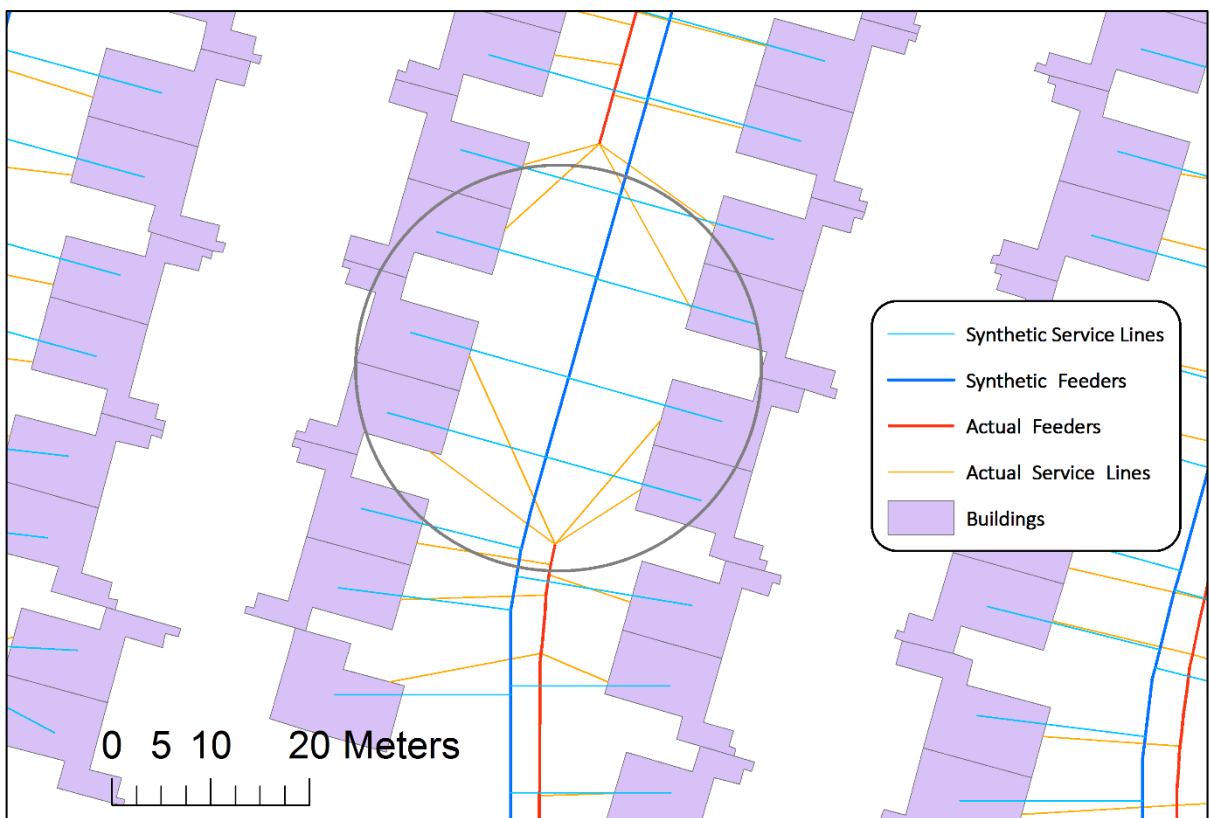
In total, 75,430 service lines pairs in Newcastle upon Tyne were found and used for validation. A histogram was drawn to show the distribution of the difference angles in the whole city. It is found that difference angle of over 70% service line pairs (52,872 pairs) is less than  $10^\circ$ , and that of over 76% service line pairs (57,409 pairs) is less than  $20^\circ$ . It is considered that the direction of synthetic service lines generally matches the actual ones. The average difference angle in the entire city is  $17.3^\circ$ . This value is considered relatively small, but still there is discrepancy. It is caused by two major reasons.

The first reason is the discrepancy of layout between synthetic and actual feeders (figure 4.26). Within grey circle of figure 4.26, actual feeders do not connect with each other, while

the synthetic feeders do, which cause large difference angles. Despite this issue, it is found that actual service lines do connect actual feeders as perpendicularly as possible, which is exactly the way to generate the layout of service lines in our algorithm.

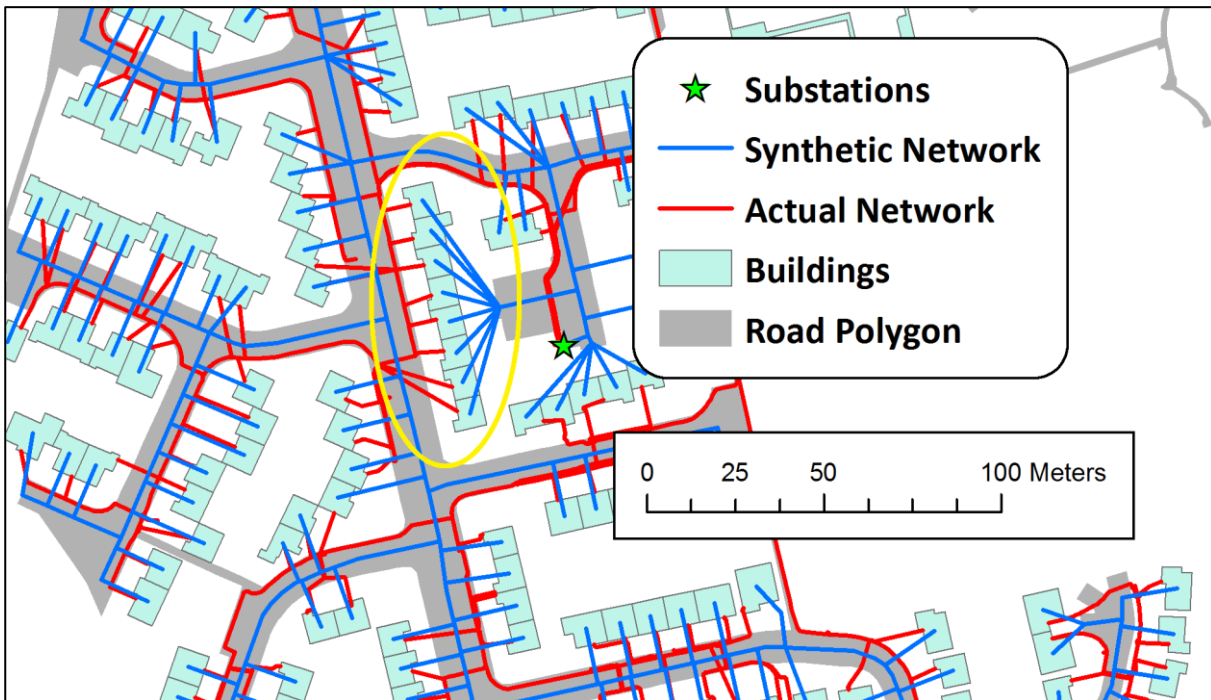


**Figure 4.25.** Distribution of difference angles.

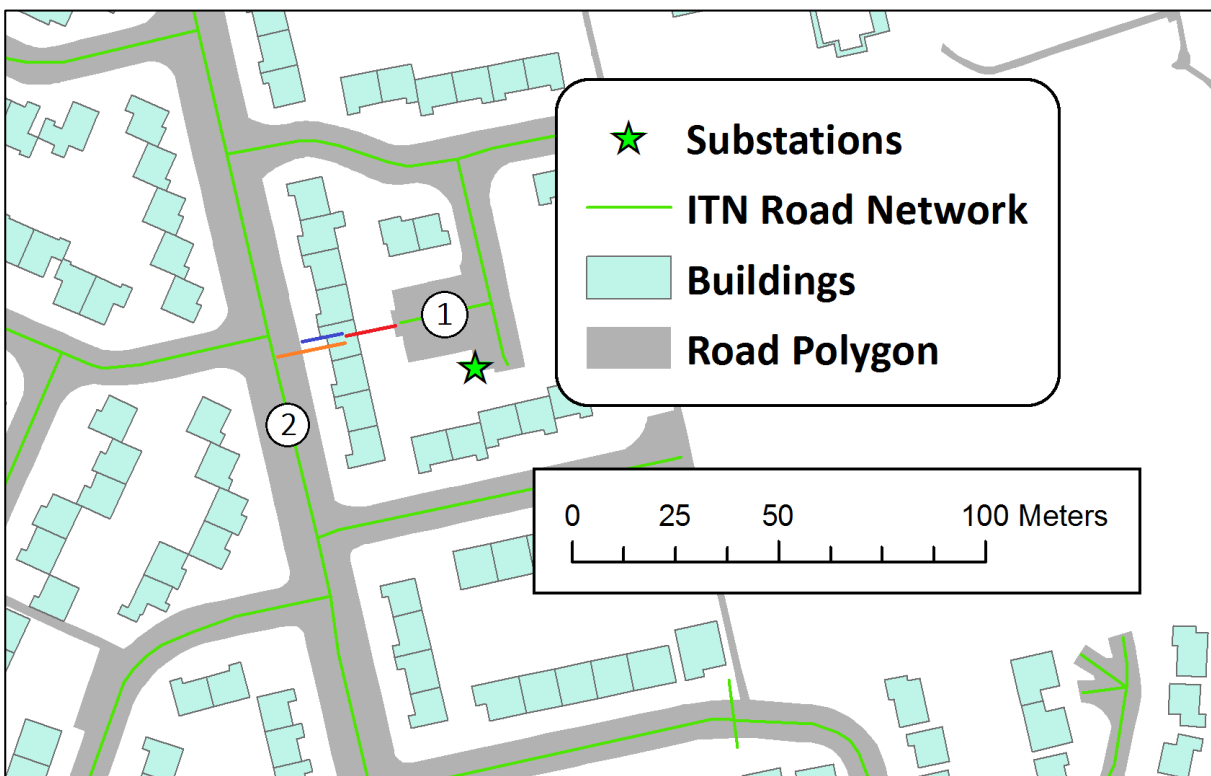


**Figure 4.26.** Large difference angles caused by different feeder layout (Contains OS and NPG data © 2018).

The second reason is related to how feeders are paved along the road. Figure 4.27 shows this issue, where the terrace within yellow circle generates very large difference angles.



**Figure 4.27.** Large difference angles within yellow circle (Contains OS and NPG data © 2018).



**Figure 4.28.** Different ways to define distance from a road to a terrace of building (Contains OS and NPG data © 2018).

In the algorithm, centrelines of roads are used to represent the road network. To generate service lines for a terrace or a building, the nearest road to centroid of it is chosen. According to the figure 4.21, that “nearest rule” also applies to actual data, but in a slightly different way. The actual data indicates that terrace or building should choose the nearest road (distance from **nearer side** of the road to it) to generate service lines (as actual feeders paved on road side instead of centrelines). Figure 4.28 gives a clearer explanation. If the distance is defined from road centrelines, then the terrace is closer to road No.2 than to road No.1 (i.e., red line is shorter than orange line). But if the distance is defined from the nearer road side, then the terrace is closer to road No.1 than to road No.2 (i.e., blue line is shorter than red line).

Both figure 4.27 and figure 4.28 indicate potential optimization of the algorithm, which is to use road centreline network together with road polygon. This should help to generate more plausible synthetic network layout compared with the actual data.

#### 4.7 Algorithm Transferability Test

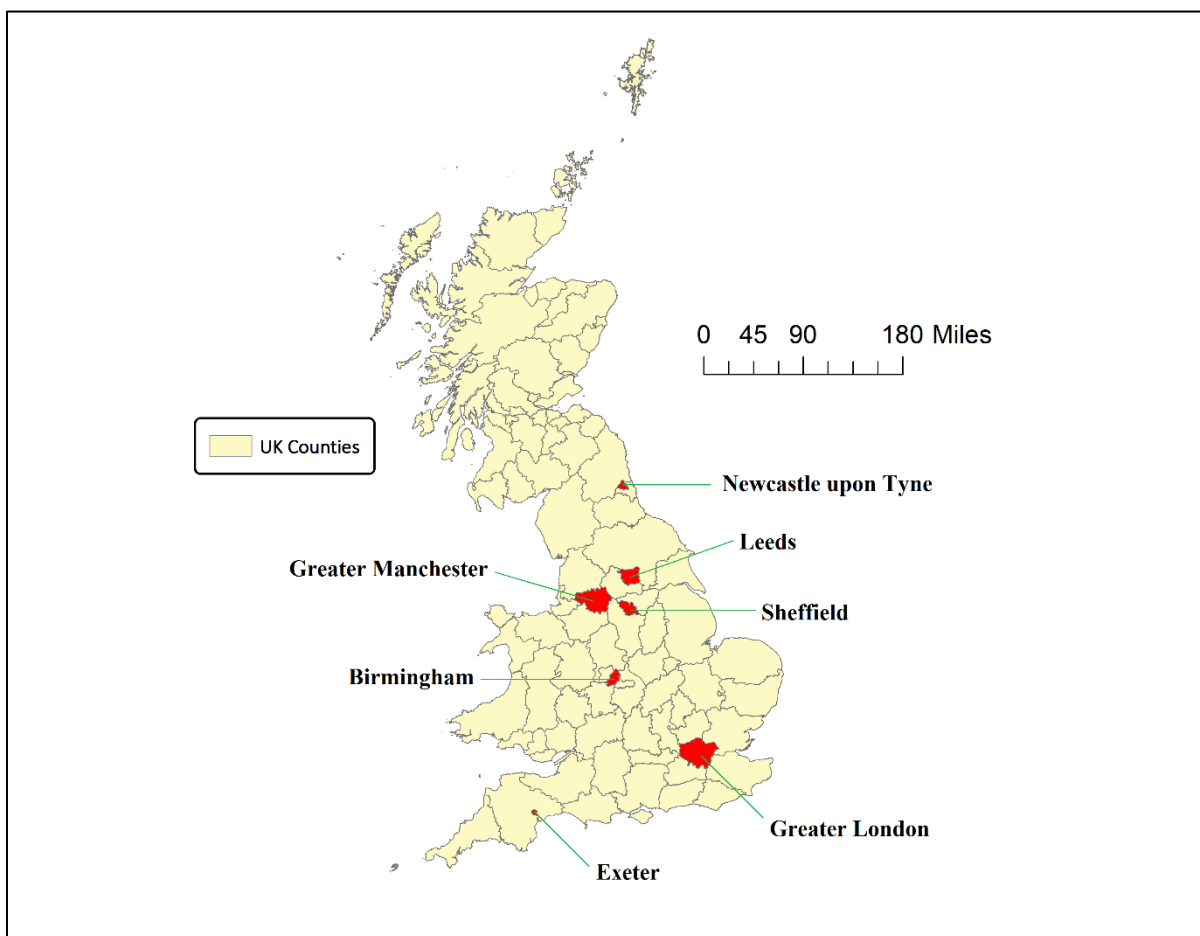
Until now it has been explained how the spatial heuristic algorithm works and how to apply it to generate electricity distribution networks in Newcastle upon Tyne. This city is relatively a small city. If ranked by population, Newcastle upon Tyne is the 30<sup>th</sup> largest city in the UK (City Mayor, 2018). This algorithm is developed as a generic algorithm for potentially any city in the world. Therefore, it is essential to further test the algorithm transferability for cities of different sizes. In this section, seven different cities (or regions) are chosen in the UK, to further test this algorithm (from small city like Exeter to mega city like London). For each city, the algorithm is executed to generate the electricity distribution networks (based on road network, buildings and substations). Table 4.5 shows the basic information of the chosen cities or regions. Figure 4.29 shows the location of these cities or regions within the UK.

City / Region	No. Residents	Area (km <sup>2</sup> )	No. Buildings	No. Substations
Exeter	107,700	47	48,821	475
Newcastle	282,300	112	104,855	636

Sheffield	530,300	368	223,159	1,512
Leeds	726,900	552	310,546	2,461
Birmingham	1,020,500	598	395,509	2,252
Greater Manchester	2,798,800	1276	1,131,645	6,913
Greater London	8,546,700	1572	2,239,213	16,839

**Table 4.5.** Chosen cities or regions for test algorithm transferability.

As mentioned in the first section of this chapter, accessing good quality spatial data for fine scale infrastructure network can be extremely difficult. Until the completion of this PhD, only Northern Powergrid data is available as actual data for electricity distribution networks. Therefore, validation for other cities or regions (other than Newcastle upon Tyne) is not possible. Even so, it is still possible to estimate the time complexity of the algorithm, by running the algorithm for these areas.

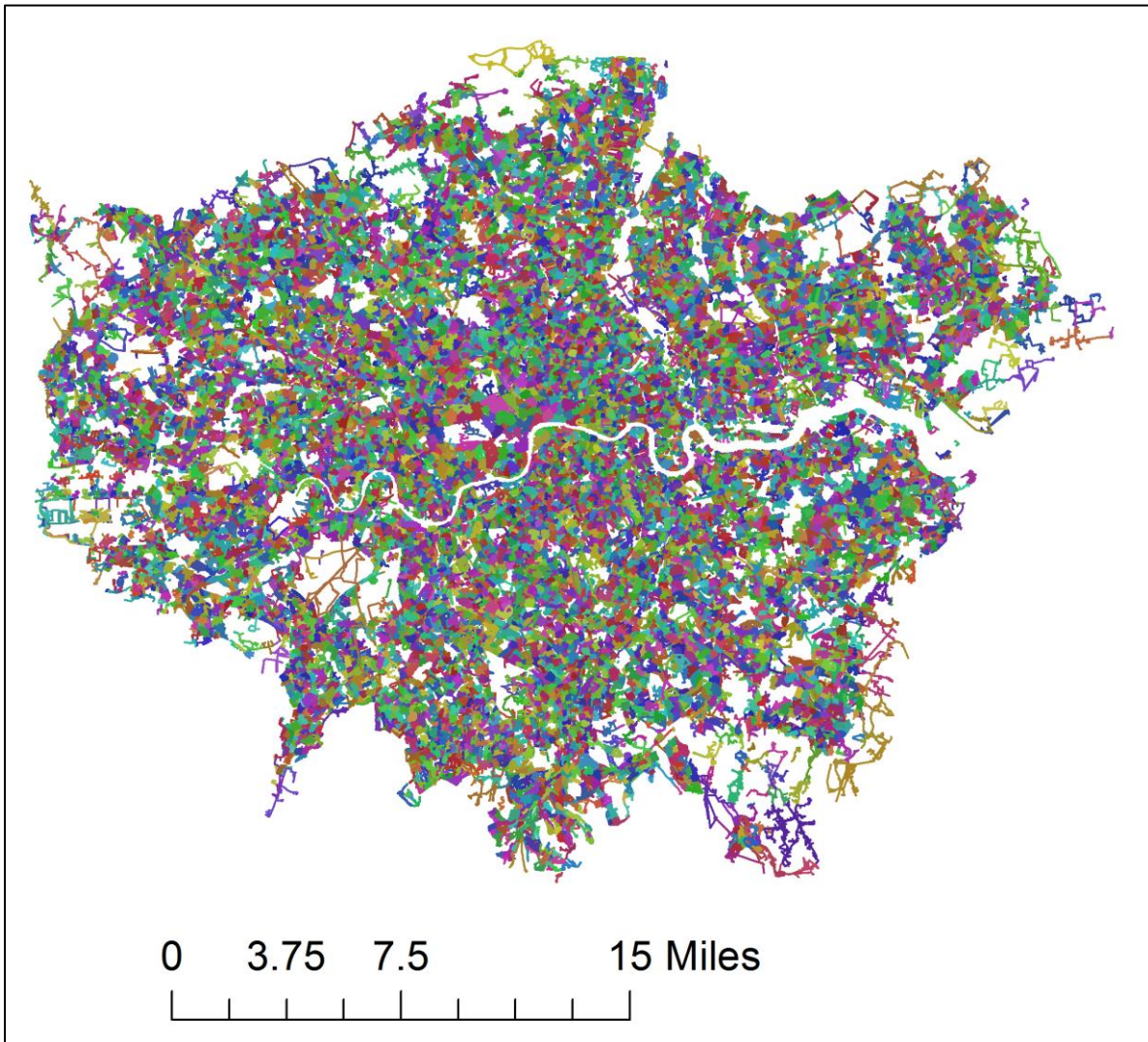


**Figure 4.29.** Location of chosen cities or regions for algorithm transferability test.

For each city or region, all the input data were downloaded from OS MasterMap ITN layer (roads), OS MasterMap PoI layer (substations) and OS MasterMap Topography layer (buildings) (Ordnance Survey, 2018). The algorithm was run as a Python script on a desktop workstation, with 2 core CPUs (Intel(R) Xeon(R) Gold 6134 CPU @ 3.20 GHz), and 512 GB memory. 10 meters was used as  $d_{thresh}$  and the Delaunay triangulation was applied. The characteristics of the algorithm result were shown in table 4.6. Table 4.6 shows that, for any city, the size of the synthetic networks (total number of the nodes) is almost twice the number of buildings (actually more than twice). That is because for each building, a “building node” and a “buildingAccess node” are generated in the result networks. Moreover, road network is used as a “back bone” to generate synthetic networks, therefore some nodes from the road network will also be kept in the synthetic networks. Therefore, the more buildings there are in the input data, the larger synthetic networks will be generated. The largest network result is the electricity distribution networks for Greater London, where 16,839 substations serve 2,239,213 individual buildings (figure 4.30). The synthetic network results for Exeter, Sheffield, Leeds, Birmingham, and Greater Manchester are shown in Appendix C, figure C7 to C11.

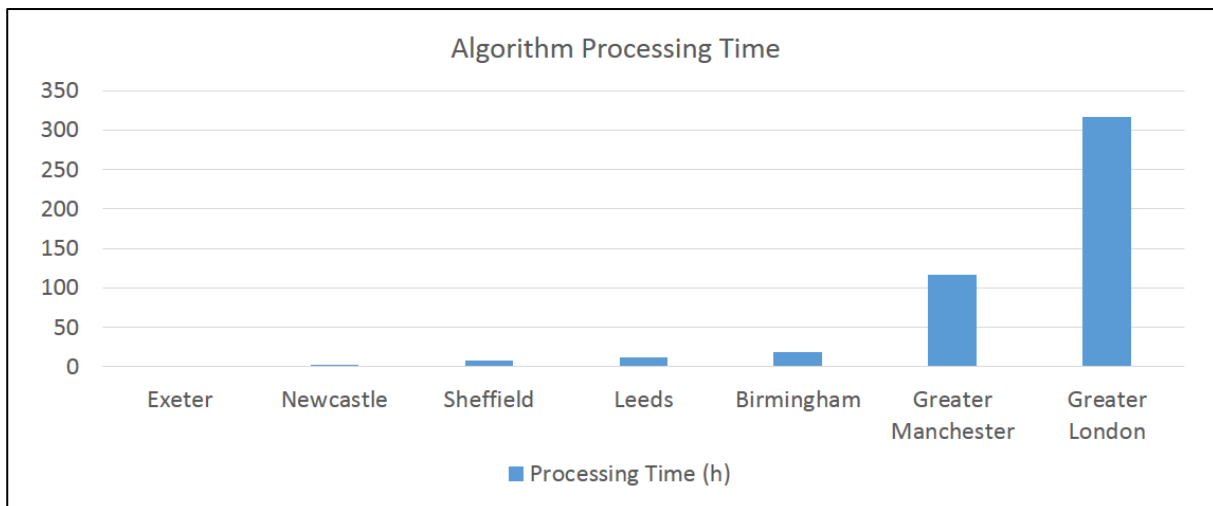
Area/Region	No. Buildings	No. Substations	No. Nodes	No. Edges	Processing Time (h)
Exeter	48,821	475	99,338	99,242	1.41
Newcastle	104,855	636	209,886	209,892	2.27
Sheffield	223,159	1,512	446,697	446,742	7.85
Leeds	310,546	2,461	628,234	628,079	12.7
Birmingham	395,509	2,252	797,741	798,137	14.2
Greater Manchester	1,131,645	6,913	2,288,186	2,289,317	87
Greater London	2,239,213	16,839	4,528,952	4,512,779	316

**Table 4.6.** Characteristics of synthetic networks generated for test cities or regions.



**Figure 4.30.** Synthetic electricity distribution networks for Greater London, where each colour represents a single network instance.

The algorithm processing time was shown in a histogram (figure 4.31).



**Figure 4.31.** Algorithm running time for different test areas.



It is argued that in this algorithm, the most computationally expensive part is the “cluster-asset dependency calculation”. This is the process in the topology generation part (section 4.3.1, see figure 4.6) of the algorithm, which is, for each cluster, assigning a nearest asset to it (via the base network). The percentage of processing time “cluster-asset dependency calculation” was measured and shown in table 4.7.

Area/Region	Algorithm running time (h)	Cluster-asset dependency calculation time (h)
Exeter	1.41	0.80 (57%)
Newcastle	2.27	1.33 (59%)
Sheffield	7.85	4.95 (63%)
Leeds	11.7	7.9 (68%)
Birmingham	14.2	9.9 (70%)
Greater Manchester	87	64 (74%)
Greater London	316	278 (83%)

**Table 4.7.** Percentage of cluster-asset dependency calculation time.

From table 4.7, it is found the cluster-asset dependency calculation accounts for a large percentage of algorithm total running time. This becomes more apparent when city is large (for example, Birmingham, Greater Manchester, and Greater London).

Therefore, it is argued that the time complexity of the cluster-asset dependency calculation will be a good proxy for the overall algorithm (especially for large data set). From now, the cluster-asset dependency calculation will be termed “CADC process” until the end of this chapter, for easy reference.

Dijkstra shortest path calculation is the essential part in the CADC process, because for each cluster, Dijkstra shortest path algorithm will be called to find the nearest asset (via the base network). The time complexity of Dijkstra path algorithm is  $O(E + V \log_2 V)$  (Barbeheen, 1998). In here  $E$  and  $V$  refer to the number of edges and nodes in the graph. To further

understand the complexity of CADC process, the notations shown in table 4.8 are used. For each different test area, the values for these notations are shown in table 4.9.

Notation	Description
$N_b$	Number of buildings
$N_a$	Number of assets
$N_c$	Number of clusters
$E_r$	Number of edges in the road network
$V_r$	Number of nodes in the road network
$E_b$	Number of edges in the base network (in the topology generation process)
$V_b$	Number of nodes in the base network (in the topology generation process)

**Table 4.8.** Notations used to assess time complexity of CADC process.

City/Area	$N_b$	$N_a$	$N_c$	$E_r$	$V_r$	$E_b$	$V_b$
Exeter	48,821	475	4,739	7,987	7,963	17,431	17,424
Newcastle	104,855	636	7,719	16,963	16,776	32,370	32,347
Sheffield	223,159	1,512	16,778	21,490	21,447	55,039	54,986
Leeds	310,546	2,461	25,044	39,203	38,262	89,240	88,793
Birmingham	395,509	2,252	33,236	33,495	33,294	99,656	98,858
Greater Manchester	1,131,645	6,913	89,105	143,976	142,123	320,279	318,357
Greater London	2,239,213	16,839	162,251	275,191	273,264	597,717	592,989

**Table 4.9.** Values of notations for the test area.

First, the time complexity to resolve Dijkstra path algorithm one time, is transformed to  $O(E_b + V_b \log_2 V_b)$  in our case. Since Delaunay triangulation is applied, each cluster will only need to find the nearest asset from *three assets* (a constant value) via the base network. Therefore, time complexity of the entire CADC process becomes:

$$O(N_c (E_b + V_b \log_2 V_b))$$

From table 4.9,  $E_b$  is almost always equal to  $V_b$ , therefore:

$$O(N_c(E_b + V_b \log_2 V_b)) \approx O(N_c(V_b + V_b \log_2 V_b))$$

This is equal to:

$$O(N_c(V_b(1 + \log_2 V_b)))$$

Note in table 4.9,  $V_b$  is approximately proportional to  $N_c$ , and the ratio of  $V_b/N_c$  is between 3 and 4 regardless of city size. That is because, to construct a base network, for each cluster, its centroid and the project point (on the road network) will be added to the road network, that means:

$$V_b \approx N_c * 2 + V_r$$

It is also found for any city, the size of the road network (number of nodes) is approximately proportional to  $N_c$ , and the ratio of  $V_r/N_c$  is between 1 and 2. This is exactly the reason that  $N_c$  is almost proportional to  $V_b$ .

Knowing this, the CADC complexity can be simplified as follows, where  $r$  is the ratio of  $V_b/N_c$ , which is a number between 3 and 4:

$$O(N_c(V_b(1 + \log_2 V_b))) = O(N_c(N_c * r * (1 + \log_2(N_c * r))))$$

Now it is argued that, due to the *log* function, the value of  $r * (1 + \log_2(N_c * r))$  will be approximately fixed, especially for large city or area, for example for Greater Manchester and Greater London. Please see table 4.10 for details.

City/Area	r	N <sub>c</sub>	r * (1 + log <sub>2</sub> (N <sub>c</sub> * r))
Exeter	3.67	4,739	51.673
Greater Manchester	3.57	89,105	68.901
Greater London	3.65	162,251	73.365

**Table 4.10.** Change of value  $r * (1 + \log_2 (N_c * r))$ , when area size is doubled.

Table 4.10 shows that even city size (number of clusters) increases 35 times (from Exeter to Greater London), the value  $r * (1 + \log_2 (N_c * r))$  only increases by 33%. When city size is almost doubled (from Greater Manchester to Greater London), the value  $r * (1 + \log_2 (N_c * r))$  only increased by 1.5%. This increase will become less apparent when processing even larger city data.

Due to this, for large city, the CADC complexity can be further simplified as follows:

$$O(N_c(N_c * r * (1 + \log_2 (N_c * r)))) \approx O(N_c^2), \text{ (especially for large } N_c \text{ value)}$$

Finally, note in table 4.9,  $N_b$  is proportional to  $N_c$  (for a fixed  $d_{thresh}$  such as 10 meters), and the ratio of  $N_b / N_c$  is between 10 and 14, that means the CADC complexity can be roughly transformed to:

$$O(N_c^2) \approx O(N_b^2)$$

Therefore, it is concluded that, by approximation, the CADC time is proportional to square of the number of input buildings, and would be a proxy to evaluate the running time of the entire algorithm. This approximation will be more accurate when city size is very large (such as Greater Manchester or Greater London). For example, Greater London contains twice the number of buildings compared with Greater Manchester, and therefore it is reasonable to expect the algorithm to be roughly four times slower.

## 4.8 Conclusion

In this chapter, a generic applicable spatial heuristic algorithm was presented and explained for generating plausible fine-scale infrastructure networks which connect assets (of any type) and their dependent buildings. A pilot study was undertaken to generate all the low voltage electricity distribution networks in Newcastle upon Tyne. A validation was done using the mapped distribution network from the local power company. Validation was done to measure the spatial proximity between the synthetic and actual network. In the end, a transferability test was run to test the processing time of algorithm using different sized data. There are several interesting findings in this chapters which might point our potential future work.

First of all, when doing the validation, it is found that at least for the electricity distribution networks, the feeder cables should be paved only along one side of the road, instead of the centrelines. That created some discrepancy between our synthetic feeders and actual ones. This discrepancy apparently depends on the width of the road. For a more accurate version of the algorithm, using the ITN road network together with the road polygon layer would be essential. This is considered to be an important optimization in the future.

Secondly, the algorithm is a generic spatial algorithm for any type of infrastructure network. Therefore, no other non-spatial attributes are considered, such as capacities (the maximum number of buildings each asset can serve). Accuracy of the synthetic networks can be improved by taking this into consideration.

Finally, running this algorithm can be expensive (backed by transferability test) especially for large city. Doubling the city size means spending four more times to complete the algorithm. This can be very long if we process even larger city than London (for example, Tokyo, New York, and Beijing, etc.). There are several potential improvements that can be made.

First optimization is to have more transferability tests. Now only UK cities were chosen for the transferability test. Therefore, the rules found here (such as number of clusters is

proportional to the number of buildings, or number of roads is proportional to the number of buildings, etc.) might not apply to cities in other countries. Algorithm time complexity might not be able to be simplified as  $O(N_b^2)$  in general. Therefore, more transferability tests (using data from other countries) will be beneficial.

Second optimization is the improvement of the graph engine. Currently, the graph engine to implement Dijkstra path algorithm is NetworkX library in our implementation. Therefore, if a faster graph engine is available, it is possible to save more time, otherwise, running the algorithm on even more powerful computers (such as on the cloud) would be a good idea to complete the algorithm within reasonable time.

The third optimization is to possibly partition input data. If processing large data (all the buildings, all the assets, and all the roads) in one-go is expensive, then it would be a good idea to segment the original area into several parts. Then algorithm can run on each segment with reduced amount of input data, which can be computationally cheaper. However, how to segment the original area can be another problem, to not cause significant difference in the synthetic networks generated.

## Chapter 5. Utility Network Integration

### 5.1 Introduction

In the last chapter, a geospatial heuristic algorithm that infers the spatial layout of fine scale urban infrastructure networks, based on the location of buildings, infrastructure assets and the local road network was developed. The algorithm was applied to generate the electricity distribution networks for Newcastle upon Tyne, and has achieved high spatial accuracy when validated using network data from local utility company Northern Power Grid. The algorithm is aimed to solve the problem, in which layout of cables or pipes of infrastructure networks is completely missing.

In this chapter, the work of inferring fine scale infrastructure networks will continue, for other utility sectors for the city of Newcastle upon Tyne. The targeted utility networks are gas supply network (section 5.2), water supply network (section 5.3), and the sewer network (section 5.4). For these utility networks, layout of *main pipes* (those follow the layout of road network) is known from local utility companies (Northern Gas Networks and Northumbria Water Group). Therefore, it means there is no need to repeat work in the last chapter (such as using road network to generate geometry of main pipes / cables of the network).

However, it is necessary to carry out additional data correction work, such as data completion, or inferring flow direction if it is missing in order to generate a *complete* fine spatial scale infrastructure network (from asset to building). Furthermore, this chapter will explore how dependency between different utility networks can be represented. Case studies will be conducted in Newcastle upon Tyne and London (section 5.6), to represent the dependency from gas supply, water supply and sewer networks to the electricity distribution networks.

## 5.2 Gas Network Integration

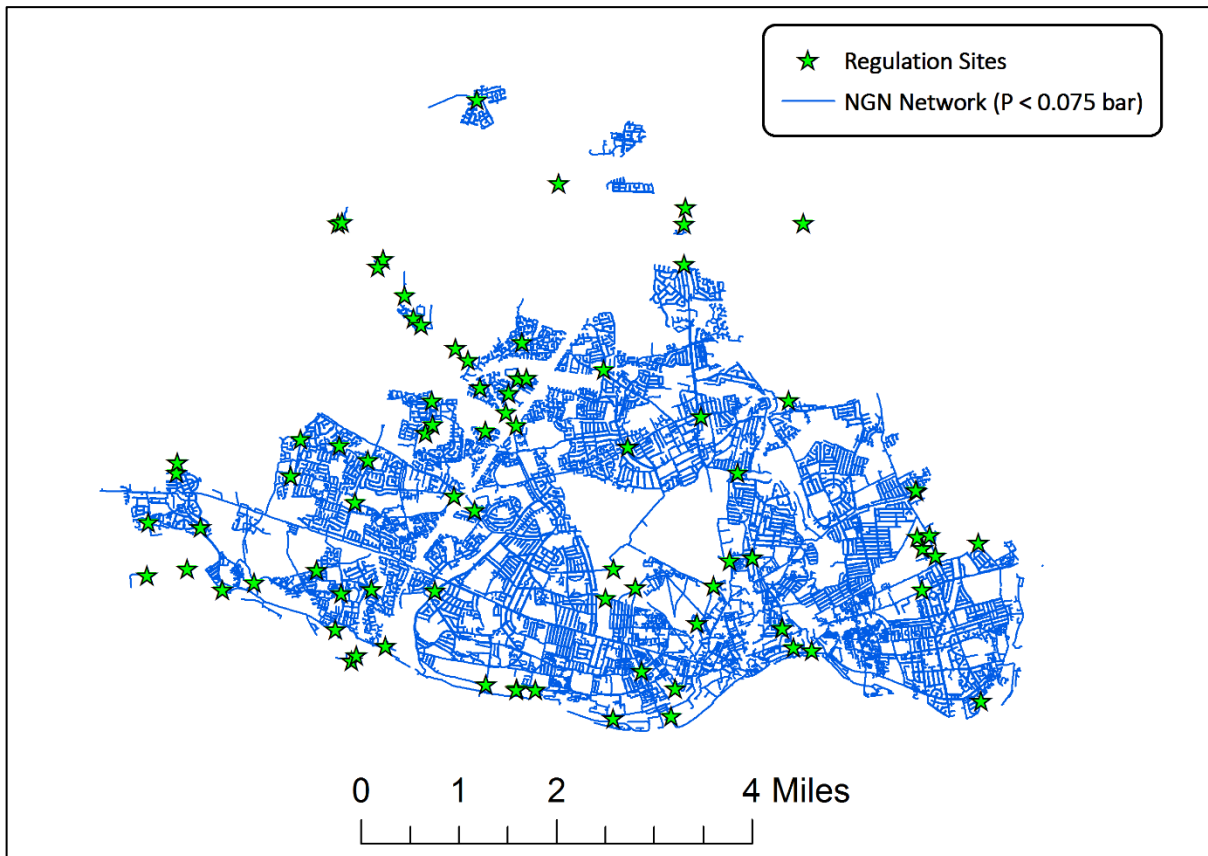
Natural gas is exploited from gas wells or imported from external countries, and then pressurized and transported through regional gas transmission networks (Vianello and Maschio, 2014). Compressor stations are set up along the transmission networks to compensate the gas pressure loss due to friction occurring within the pipes. When gas is approaching urban areas, it is sent to the pressure regulation sites to reduce the pressure of the gas feed to customers (Fügenschuh et al, 2015). The gas pressure in the transmission networks can be between 40 and 90 bar, and the gas pressure that is suitable for customer use is around 0.075 bar. Generally, it is not possible to use only one gas regulation site to reduce the pressure from transmission level to the domestic level. Instead, in the gas industry, multiple gas regulation sites are necessary to gradually reduce gas pressure. This situation is like the electricity network, where there are 132kv, 66kv, 33kv, and 11kv substations are used to gradually reduce the voltage of electricity from transmission level to the domestic level. For a gas company, only the spatial layout of gas main pipes is available. Therefore, in order to construct a fine scale gas distribution network to individual buildings, it is necessary to generate the service pipes and connect them to the gas main pipes.

### 5.2.1 Gas Network Data

With the help of the local gas provider, Northern Gas Networks (NGN), it is possible to access the layout of the low-pressure gas distribution networks for Newcastle upon Tyne. The low-pressure gas distribution networks are the lowest level of gas distribution networks within cities, where the gas pressure is around 0.075 bar. NGN provides data in the shapefile format as two files: a polyline shapefile containing the geometry the gas main pipes, and a point file shapefile containing the nodes which are junctions of gas main pipes and gas sources (gas regulation sites). Figure 5.1 shows the layout of NGN pipes and the gas regulation sites. Please note that these regulation sites are fed by high-pressure gas distribution pipes (between 0.075 and 40 bar). National Grid does provide the layout of gas *transmission* network (<https://www.nationalgridgas.com/land-and-assets/network-route-maps>) (between 40 and 90 bar), but currently data of *high-pressure gas distribution* pipes (between 0.075 and 40 bar) is



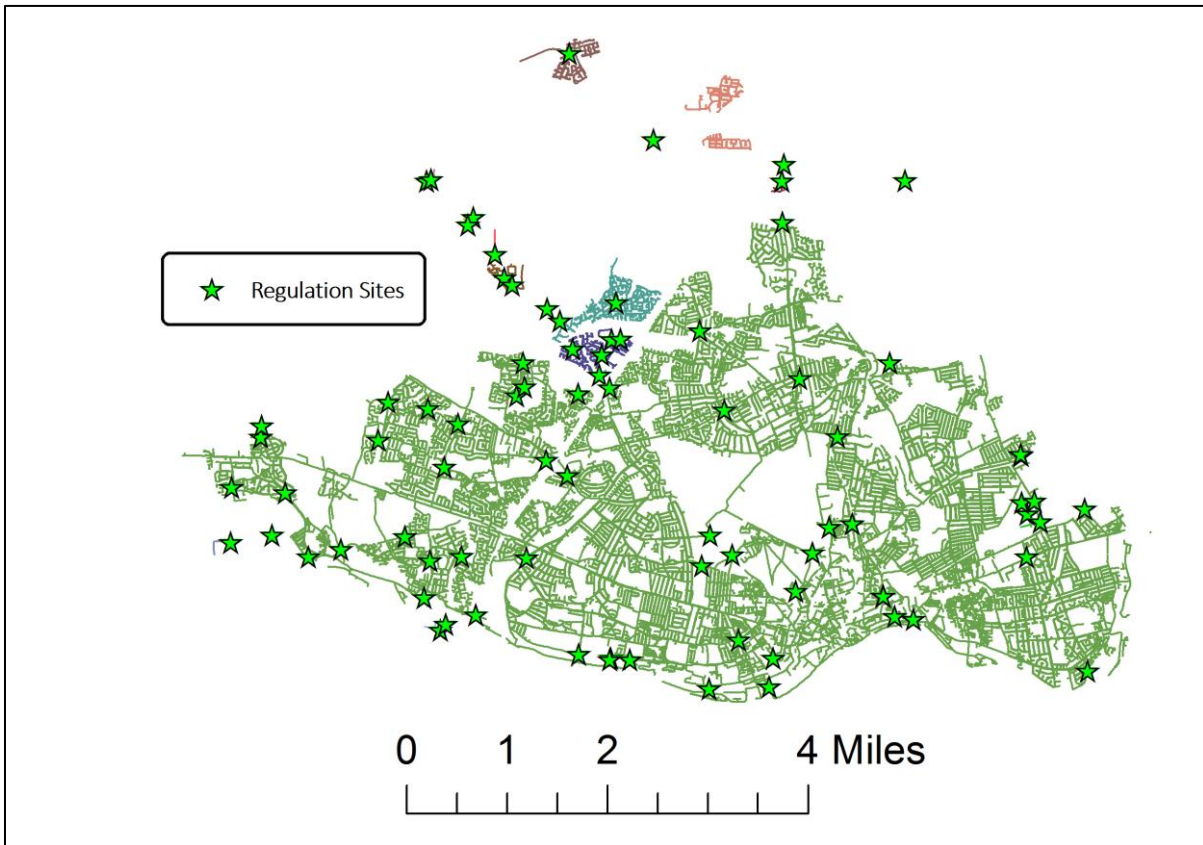
unavailable, which is why it is not visually displayed in figure 5.1.



**Figure 5.1.** NGN network for Newcastle upon Tyne (Contains NGN Data © 2018).

In total, the NGN network data totally contains 34,644 nodes and 37,655 edges. 105 of the nodes correspond to gas regulation sites. The NGN network data contains 43 sub-network instances (technically speaking, the sub-systems in the gas industry). Each sub-system is a connected network instance with one or more sources (regulation sites). Figure 5.2 shows the different sub-systems in the NGN network data, where each colour indicates one sub-system.

NGN has labelled each node and each edge with a unique Node\_ID and an Edge\_ID, respectively. Each edge (pipe) has numeric or text attributes, such as pipe diameter and pipe material (steel pipe, PVC pipe, etc.). Moreover, gas flow direction is recorded across the entire NGN network. The flow direction is encoded on each edge, by specifying the flow from-to topologically connected nodes that connect an edge (using Node\_ID). This makes it possible to integrate the buildings in order to construct a fine scale gas distribution network (from regulation sites to buildings) with flow direction encoded.

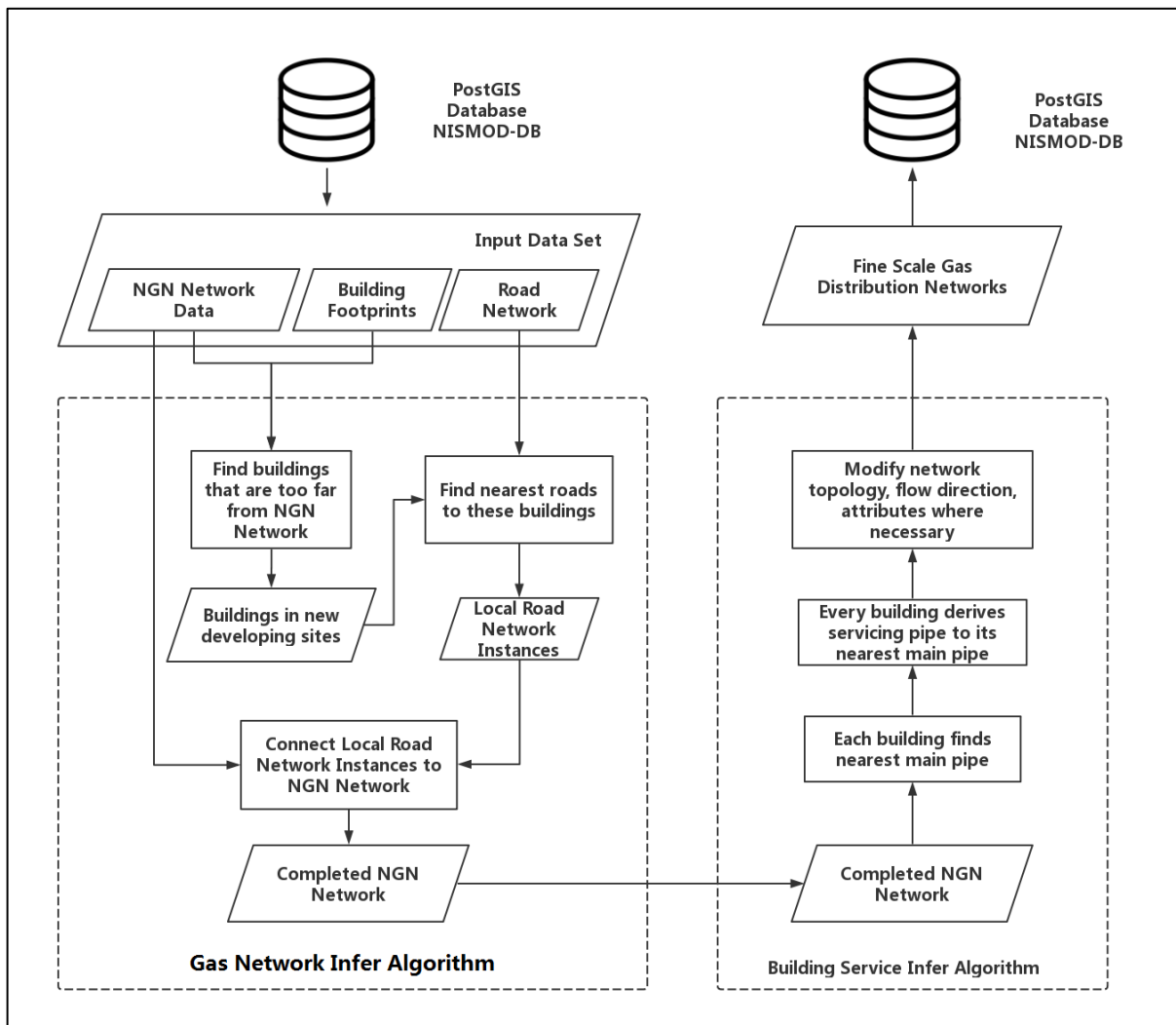


**Figure 5.2.** Different sub-systems within NGN network data, each in different colours  
(Contains NGN data © 2018).

However, a close inspection of the supplied data revealed data incompleteness. NGN network data does not exist for recently new development within Newcastle upon Tyne. Thus, before a full directed gas distribution network could be generated, the algorithm developed in Chapter 4 was (slightly modified and) employed to generate *main pipe* gas network for the areas, where NGN network data are absent.

Figure 5.3 shows the overall work flow of gas network integration. The data is first stored in a PostGIS database, then the input data sets (NGN network data, buildings, and road network) are retrieved from the database, and processed through a gas network infer algorithm to produce the completed layout of the gas main pipes (*completed NGN network*). After that, the *completed NGN network* is then processed via a building service infer algorithm where service pipes are generated and connected to the *completed NGN network*. Finally, the fine scale gas distribution network is written back to the PostGIS database. Details of these

algorithms are presented in the next two sections.

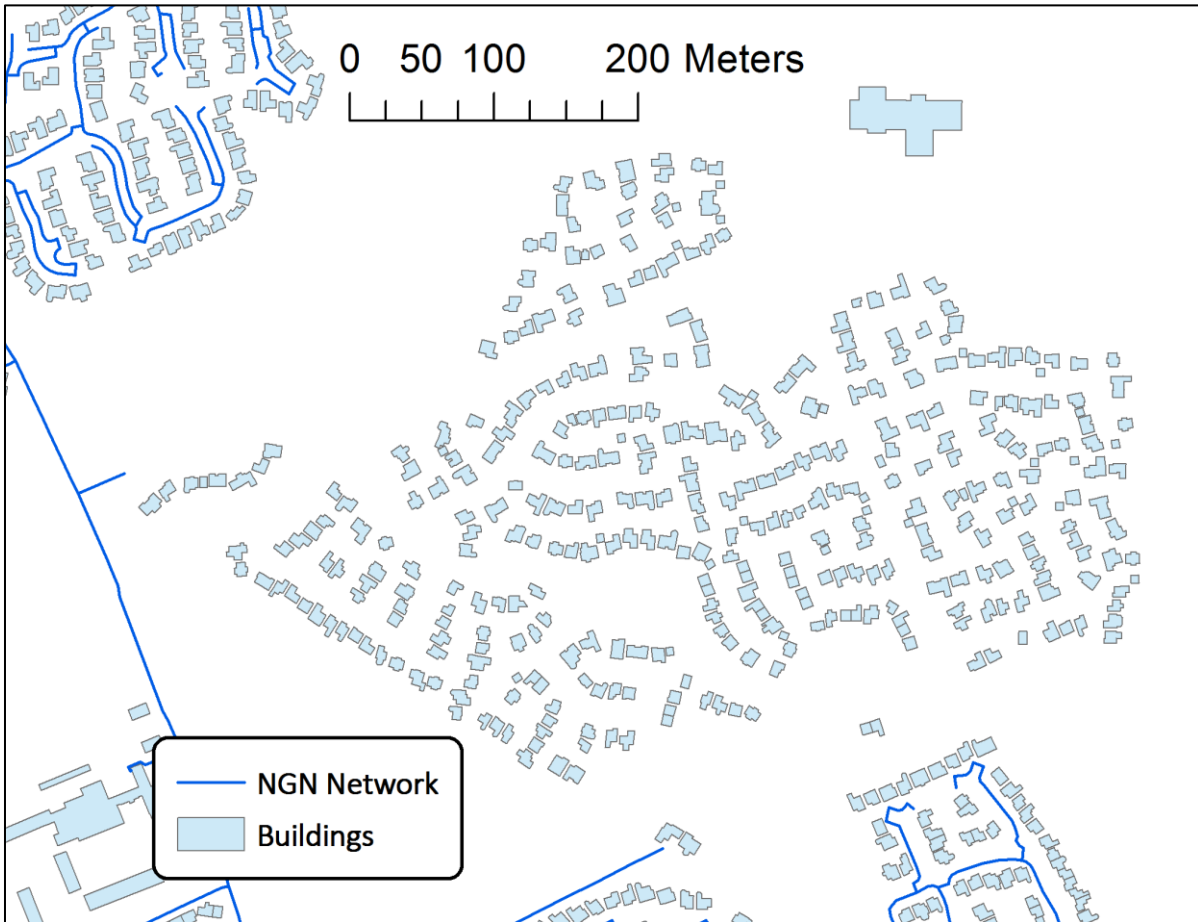


**Figure 5.3.** General work flow for gas network integration.

### 5.2.2 NGN Network Completion

For the city of Newcastle upon Tyne, it is noticed that there are some areas which are clearly not covered by NGN network (figure 5.4). As noted earlier, these problematic areas are the new development areas in the city, which will hinder generation of gas distribution networks that connect *every* building in the city.

Despite the lack of the actual data, NGN informed that it is quite reasonable to generate the synthetic layout of gas main pipe network using a local road network. Therefore, based on this information, an algorithm called gas network infer algorithm was developed to tackle this issue. Details of this algorithm are shown in listing 5.1.



**Figure 5.4.** Absence of actual data in some area of the city (Contains NGN data © 2018).

First (line 1-2 in listing 5.1), it is necessary to identify buildings where there are not existing NGN network data nearby (buildings that are too far from NGN network). The interesting part will be to quantify how far is “too far”. In here a parameter  $d$  (50 meters in this case) is defined. By setting  $d$ , it is possible to find all the buildings ( $B_{fetched}$ ) that have larger distance (than  $d$ ) to the existing NGN network. For all 104,855 buildings in Newcastle upon Tyne, 4,287 of them are identified to be at least 50 meters away from NGN network, which is shown in Figure 5.5.

After that, for each building fetched, the nearest road to this building will be selected, which will be stored in a set called  $R_{fetched}$ . These road segments will be used to infer the “missing” parts of the NGN network. For all 16,963 road segments in Newcastle upon Tyne, 711 of them are selected as  $R_{fetched}$ , which is shown in figure 5.6.

---

**Algorithm : Gas Network Infer Algorithm**

---

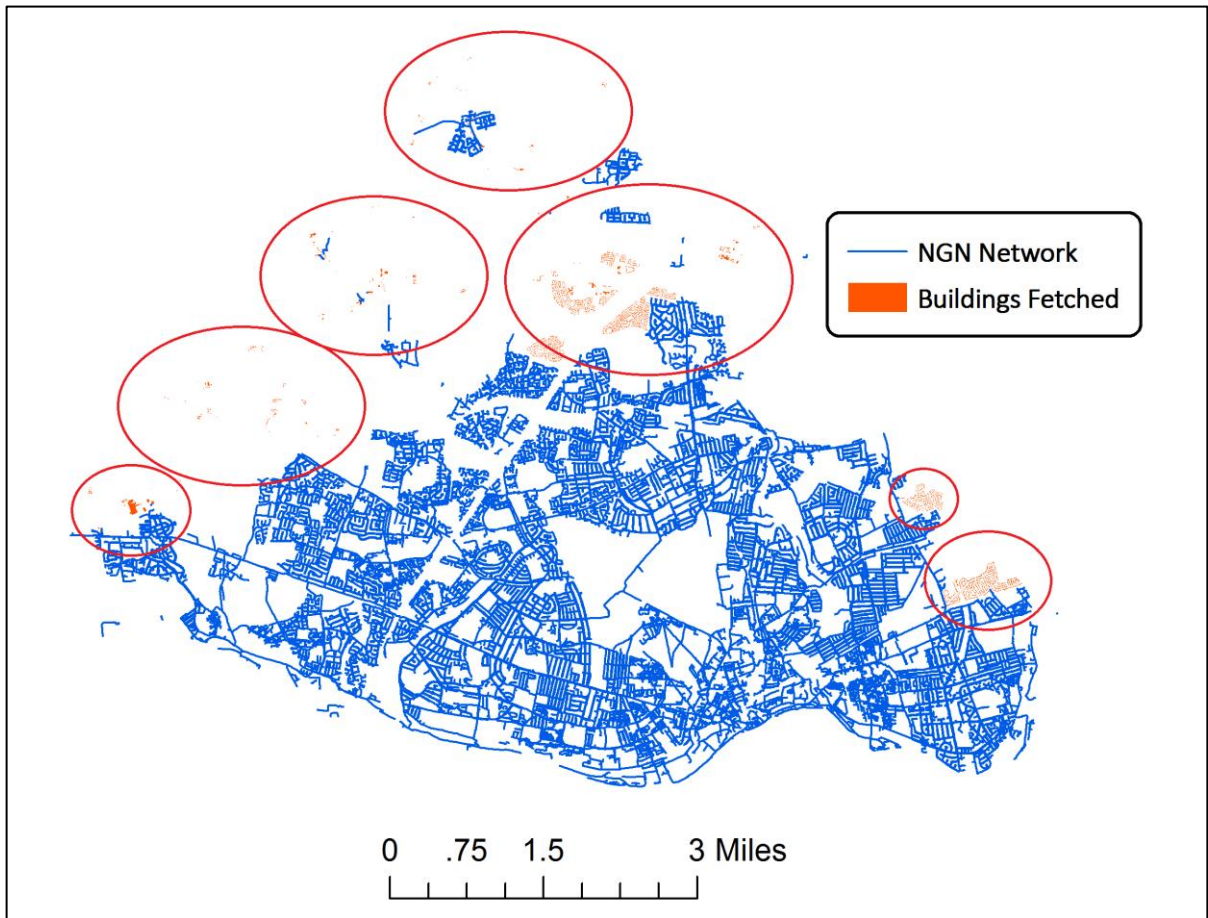
**Input:** NGN Gas Network  $G_{NGN}$ , set of Buildings  $B$ , set of Roads  $R$ , Road Network  $G_{road}$

**Output:** the completed NGN Network  $G_{completed}$

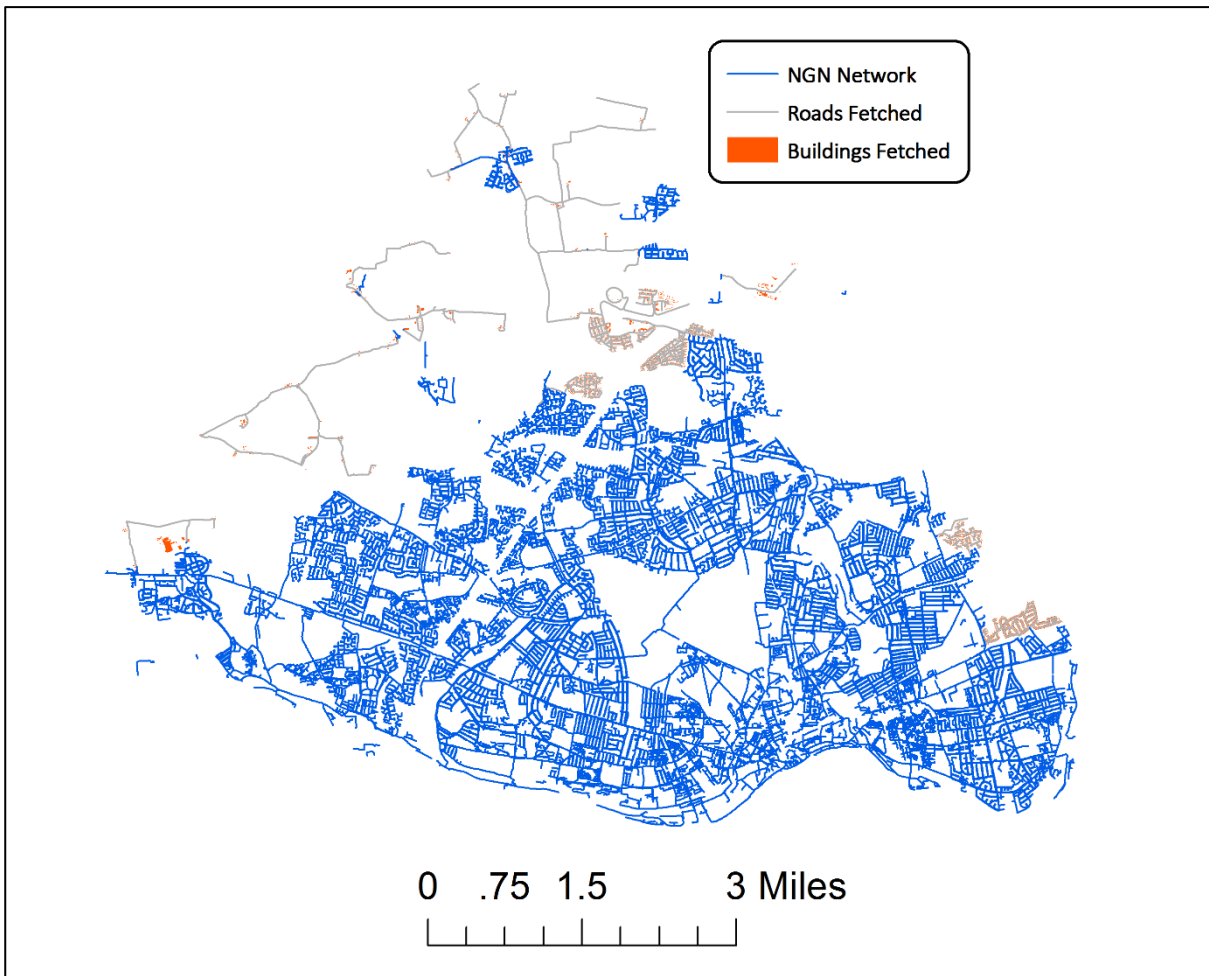
---

- 1: for  $b \in B$ , find  $b$  if  $distance(b, G_{NGN}) > d$ , put all the fetched  $b$  in a new set called  $B_{fetched}$
  - 2: for  $b \in B_{fetched}$ , find nearest  $r \in R$  to it. Then put fetched  $r$  in a new set  $R_{fetched}$
  - 3: use  $R_{fetched}$  to construct sub network instances, each instance is  $g_{instance}$ , and put them in a set  $G_{instance}$
  - 4: extend each  $g_{instance} \in G_{instance}$  by adding edge to connect the nearest CSEP node in  $G_{NGN}$
  - 5: for  $g_{instance} \in G_{instance}$ , infer flow direction on each edge via dijkstra shortest path algorithm
  - 6: now connect each  $g_{instance}$  to the  $G_{NGN}$  and generate the  $G_{completed}$
- 

**Listing 5.1.** Pseudo code for the gas network infer algorithm.



**Figure 5.5.** All the buildings that are too far (distance > 50 meters) from NGN network. They are indicated within the red circles (Contains NGN data © 2018).

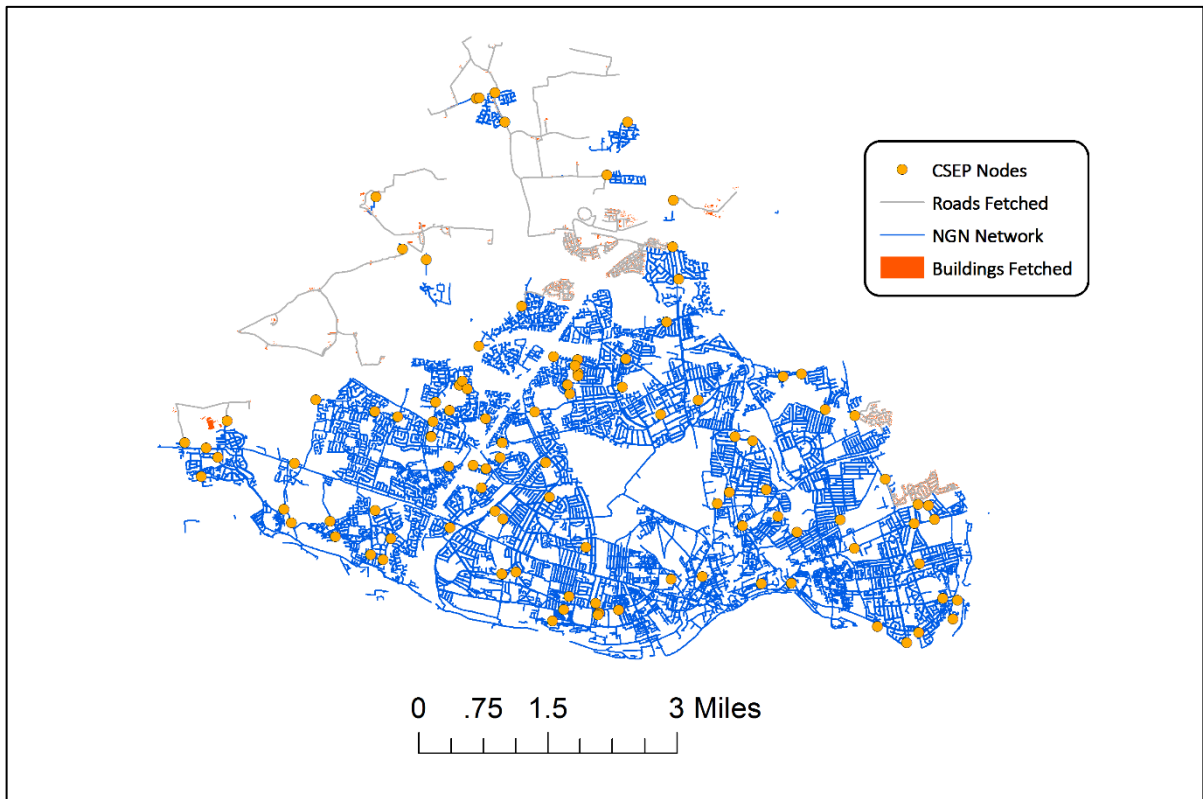


**Figure 5.6.** Road segments fetched, which are nearest to the fetched buildings (Contains NGN data © 2018).

The next step (line 3 – 4 in listing 5.1) identifies how many connected sub network instances can all the fetched road segments form. This is done using NetworkX library (NetworkX, 2018). In the end, 9 connected sub network instances were found. Each sub network instance can be regarded as the synthetic part of the gas main pipes where NGN network data are missing.

Then each sub network instance will find the correct “off take” location to be able to connect to the existing NGN network. NGN explained that, despite lacking the gas main pipe network data in these recently developed areas, in the existing data, there are some nodes with a specific type called “CSEP”. CSEP nodes are those reserved future development areas, as these CSEP nodes are connected with large diameter pipes.

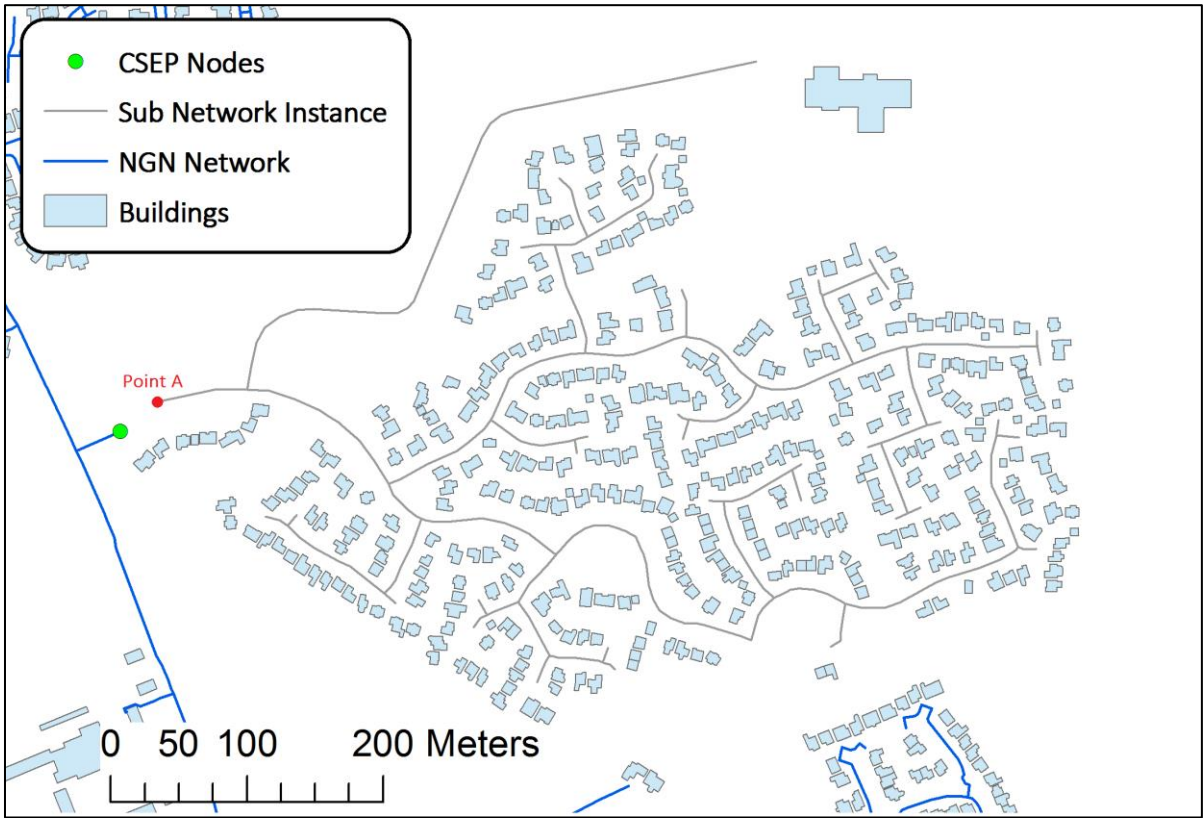
Therefore, it is plausible to consider these CSEP nodes are the “off take” locations, where synthetic main pipes can be connected to the existing ones (NGN network). There are totally 97 CSEP nodes in the Newcastle upon Tyne (figure 5.7).



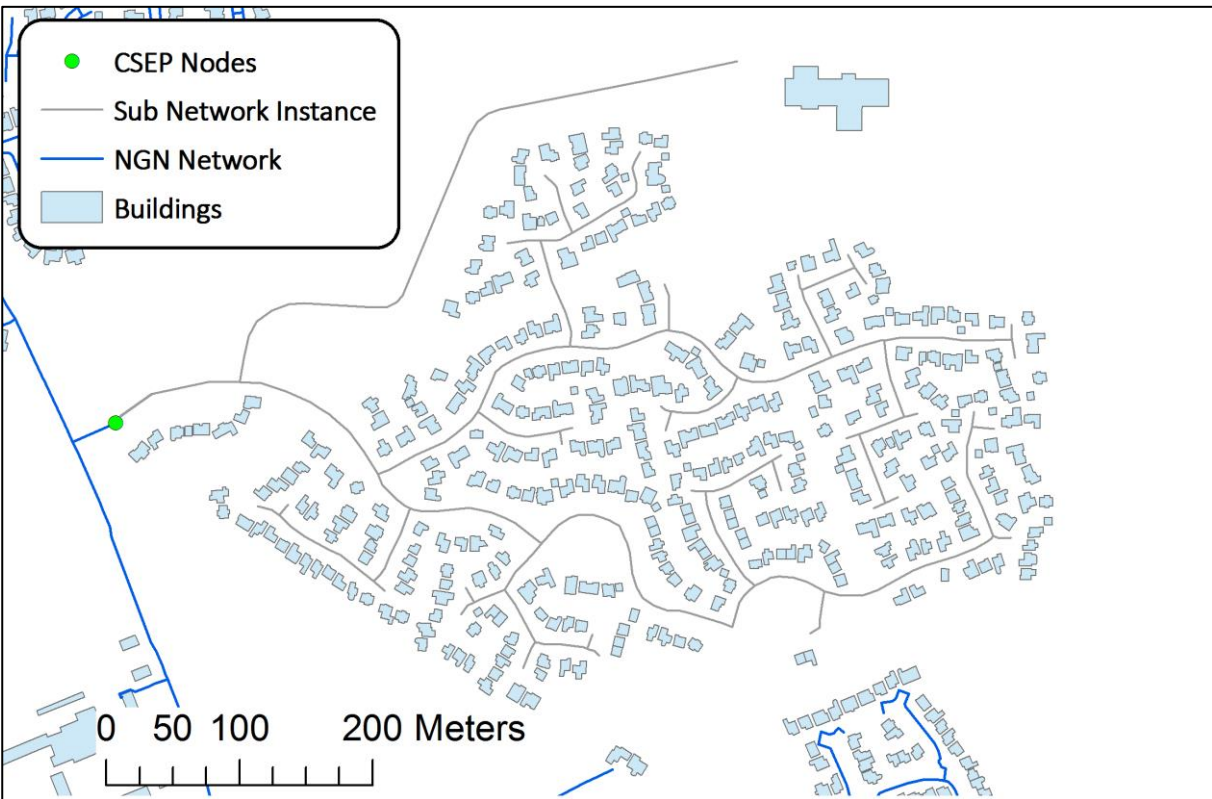
**Figure 5.7.** CSEP nodes in the NGN network data (Contains NGN Data © 2018).

Figure 5.8 and 5.9 show how to exactly make the connection between a sub network instance and a CSEP node. Figure 5.8 shows the area that is the same as figure 5.4, where there is one sub network instance (synthetic layout of gas main pipes in this area). To make things clear, it is necessary to define *the distance from a CSEP node to a sub network instance*. The distance is defined as the Euclidean distance from the CSEP node to the nearest location (point) within the sub network instance.

Then using this distance definition, for the sub network instance in figure 5.8, the nearest CSEP node will be selected, and the point A is used to calculate the aforementioned distance. After that, a straight line is used to connect the point A and the CSEP node (figure 5.9). By doing this, a sub network instance is connected to a CSEP node.



**Figure 5.8.** Before connecting a sub network instance to NGN network (Contains NGN Data © 2018).

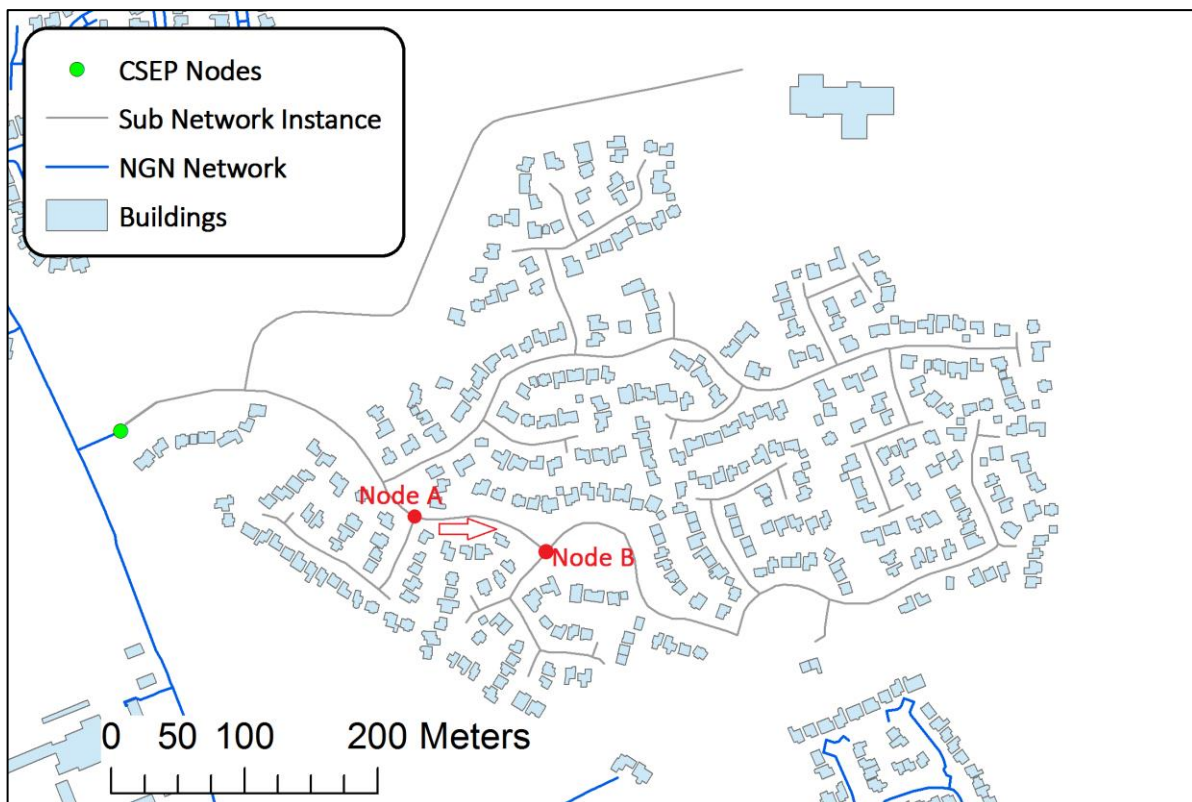


**Figure 5.9.** After connecting a sub network instance to NGN network (Contains NGN Data © 2018).



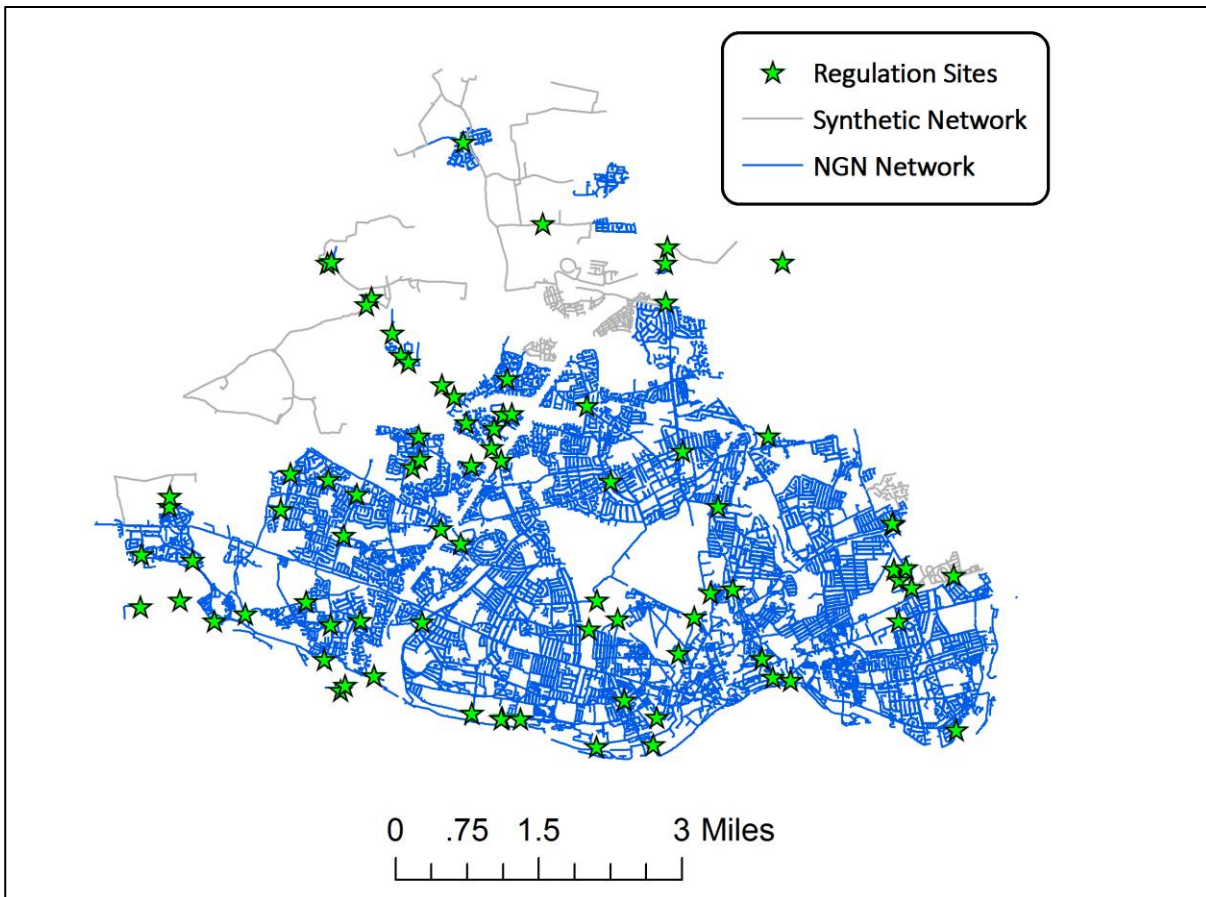
Once each sub network instance has been extended to the CSEP node, there will be two final steps: flow direction calculation and network data merging. First, flow direction will be inferred on each sub network instance (line 5 in listing 5.1). To do that (figure 5.10), it is natural to assume the gas is first fed into the CSEP node, and from there gas will flow into the entire sub network instance. Therefore, for any edge in the sub network, and the two nodes connecting this edge (node A, and node B), the Dijkstra shortest path is calculated from the CSEP node to node A and to node B respectively.

Flow direction on the edge A-B is inferred from the node corresponding to a shorter path to the node corresponding to a longer path. In this case, flow direction is from node A to node B. The calculation is done using NetworkX library (NetworkX, 2018) and is repeated for every edge on the sub network instance.



**Figure 5.10.** Infer flow direction on the sub network instance (Contains NGN data © 2018).

The completed gas main pipe network contains 32,884 network edges, of which 32,177 (97.8%) are from the existing data and 707 (2.2%) are synthetically generated. Total length for the completed gas main pipes is 1,332,971 meters, where 94.6% of them (1,261,376 meters) is from existing data, and 5.4% of them (71,595 meters) are from synthetic pipes. The result is shown in figure 5.11.



**Figure 5.11.** Completed gas main pipe network in Newcastle upon Tyne (Contains NGN Data © 2018).

### 5.2.3 Gas Distribution Network Generation

When gas main network for the whole city is available, it is possible to generate the service pipes that connect buildings and main pipes. By doing this, fine scale gas distribution networks can be generated. This process is achieved via the building service infer algorithm shown in listing 5.2.

---

**Algorithm** : Building Service Infer Algorithm - Gas

---

**Input:** a set of Buildings  $B$ , completed Gas Main Network

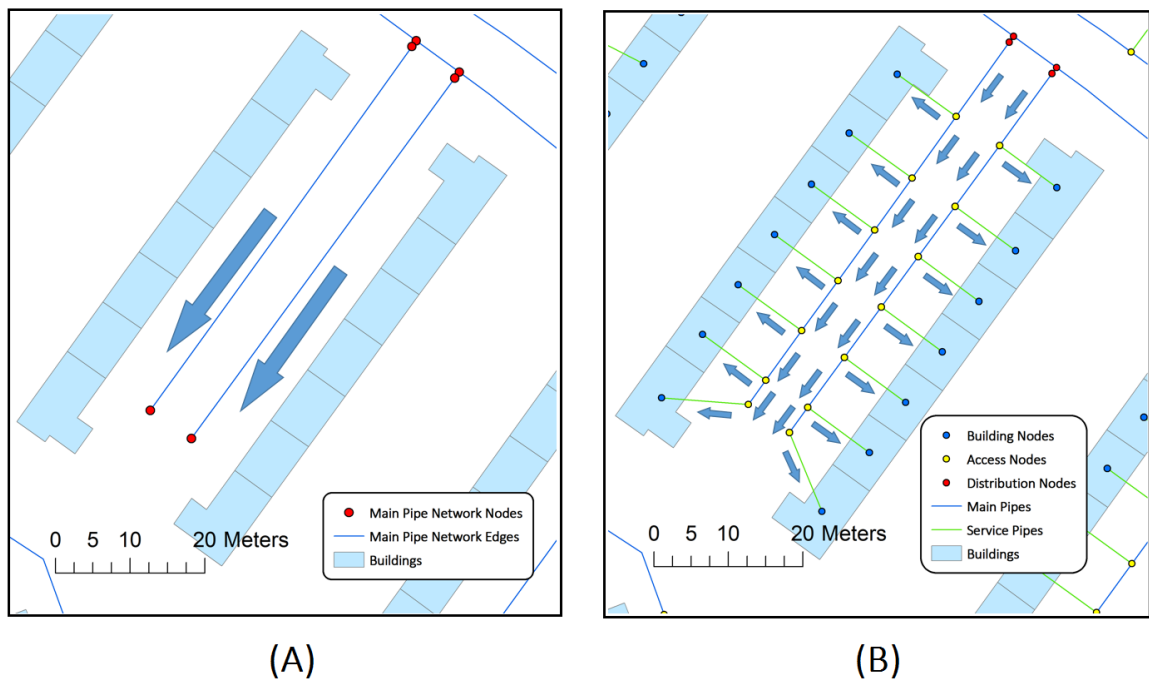
$G_{completed}$

**Output:** Gas Distribution Network  $G_{dis}$

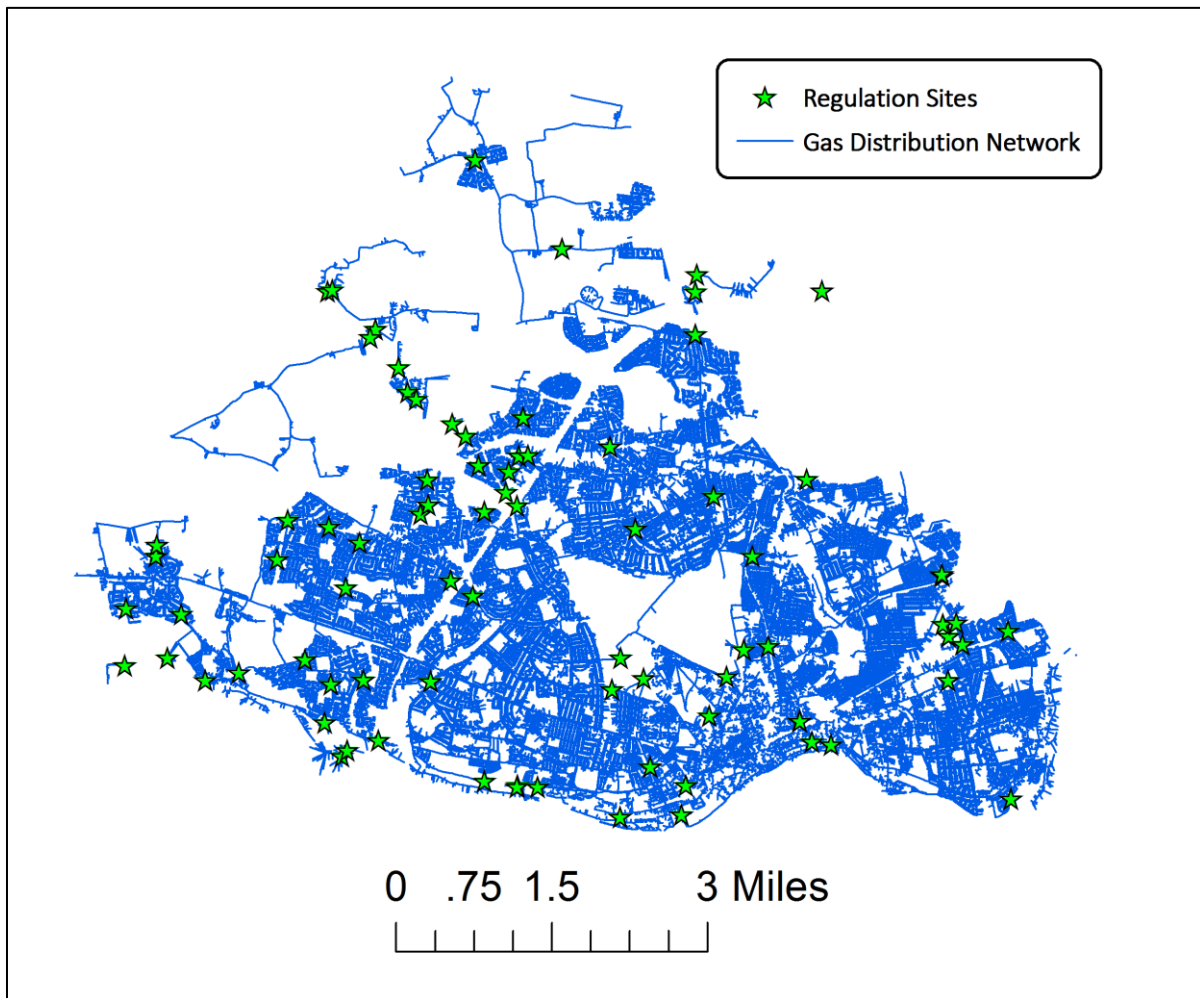
- 1: for  $b \in B$ , find nearest edge from  $G_{completed}$
  - 2: for  $b \in B$ , extract its centroid  $b.cen$  and then derive an edge to its nearest edge from  $G_{completed}$
  - 3: merge all the building nodes and derived edges (servicing pipes) to  $G_{completed}$  to save to  $G_{dis}$
  - 4: on  $G_{dis}$ , modify topology and attributes where necessary, and record flow direction on servicing pipes.
- 

**Listing 5.2.** Pseudo code for building service infer algorithm (gas).

This algorithm applies the similar approach as Chapter 4. For each building, the nearest gas main pipe will be selected, and a service pipe will be used to connect the centroid of the building and the main pipe (in a perpendicular way). Flow direction on the service pipes will be calculated, which is always to the building node. Figure 5.12 shows the example of integrating buildings to the completed gas main pipe network. For the entire city of Newcastle upon Tyne, the fine scale gas distribution network contains 236,307 nodes and 239,484 edges, which are shown in figure 5.13.



**Figure 5.12.** (A) Completed gas main pipe network, with flow direction, (B) Gas distribution network to the buildings, with flow direction encoded (Contains NGN data © 2018).

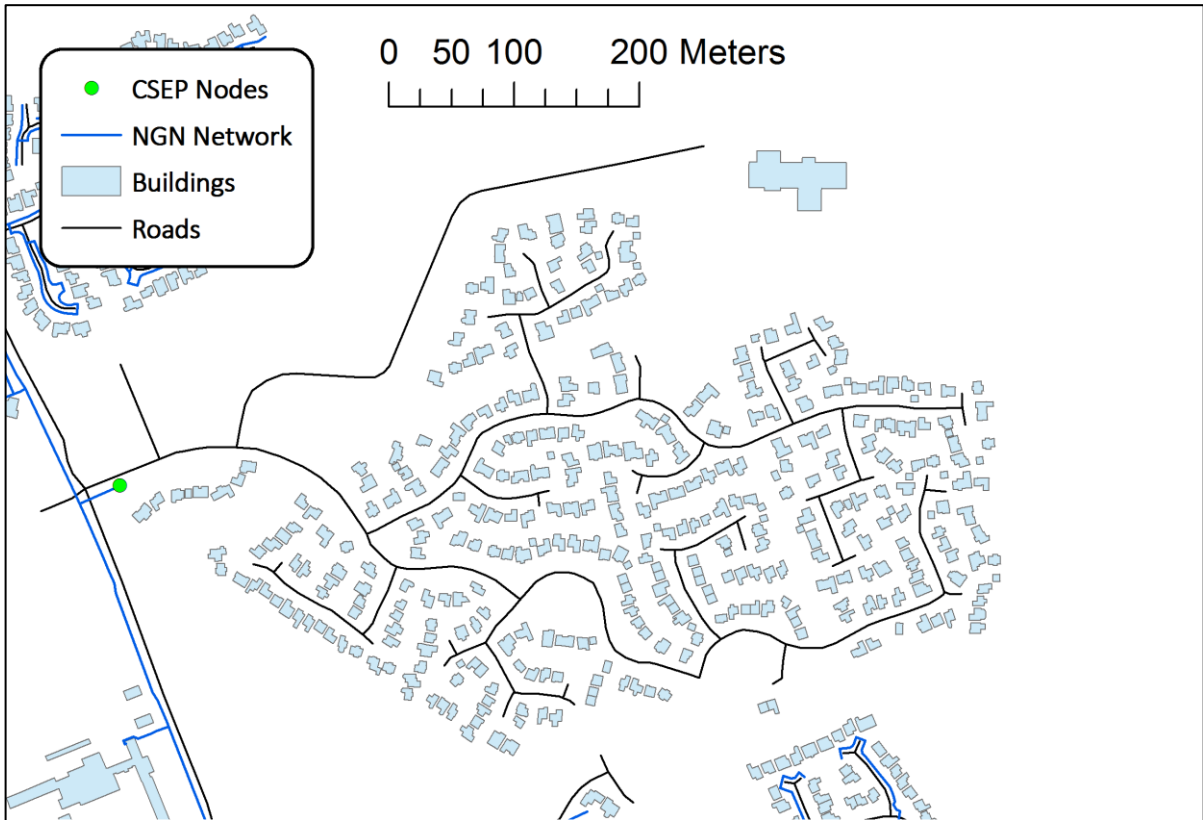


**Figure 5.13.** Gas distribution network (including service pipes) generated for Newcastle upon Tyne (Contains NGN Data © 2018).

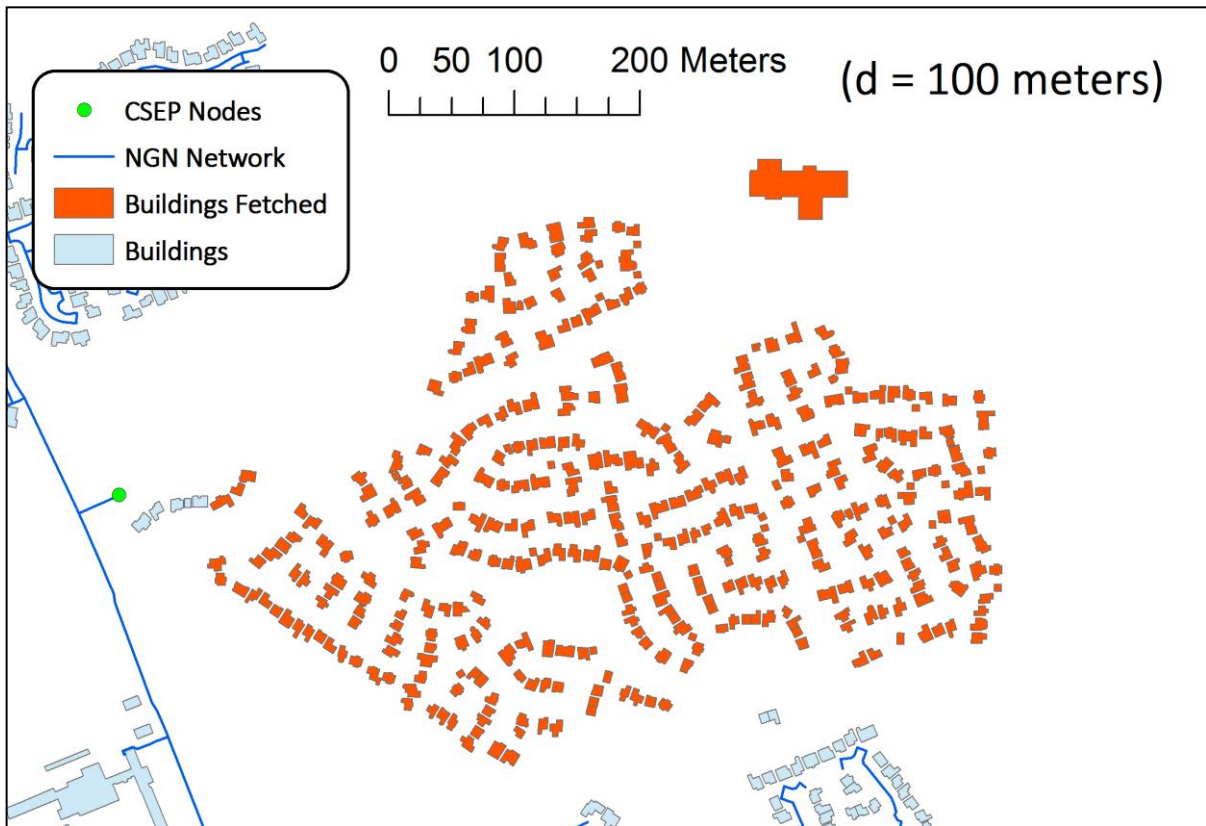
#### 5.2.4 Parameter Sensitivity

In section 5.2.2, an important parameter  $d$  is defined for the process of generating synthetic gas main pipes in the areas where NGN data is not available. This section will explore the parameter sensitivity of  $d$ . Three values are used to set up the parameter  $d$ , 25 meters, 50 meters, and 100 meters. To explain how the gas distribution network can change according to different  $d$  values, the small area shown in figure 5.4 is used here again. This is shown in figure 5.14, where road network is also displayed.

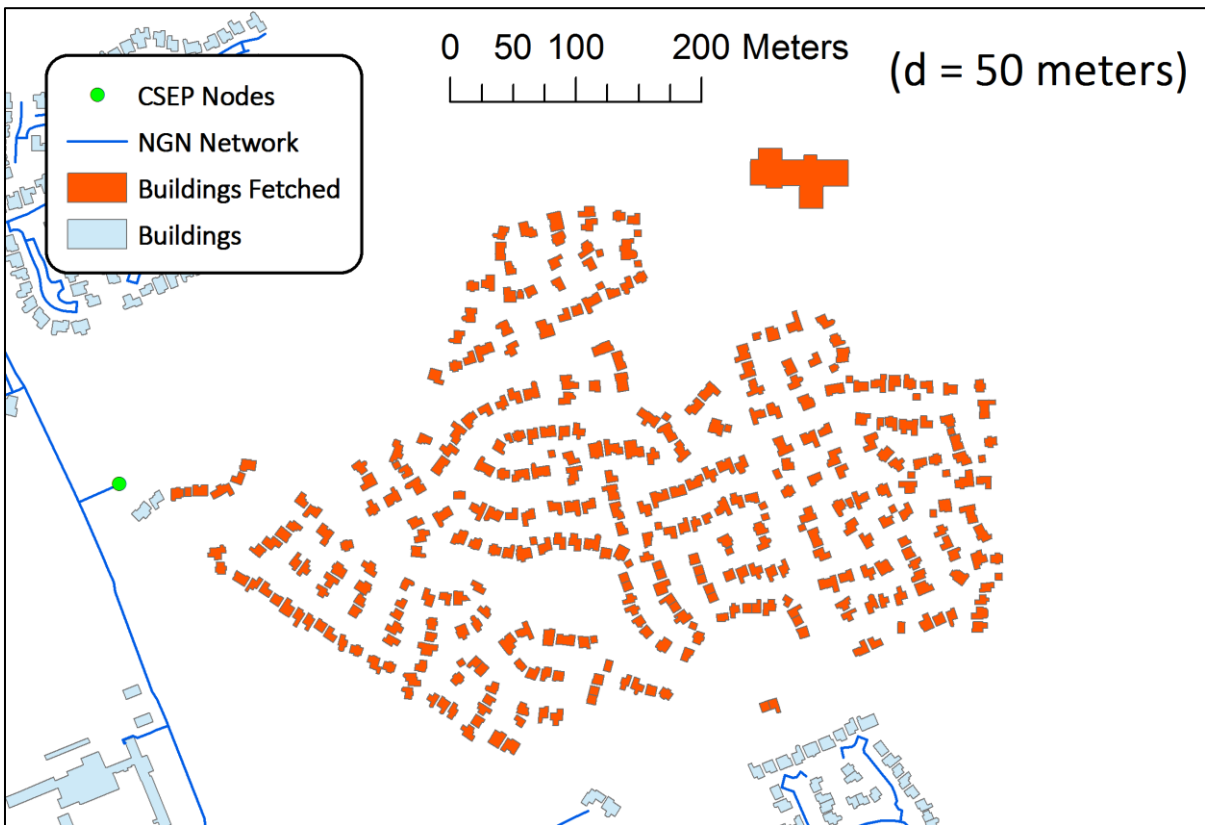
First, using these three values, buildings which have larger distance (than  $d$ ) to the NGN network are selected and shown in figure 5.15, 5.16 and 5.17, respectively.



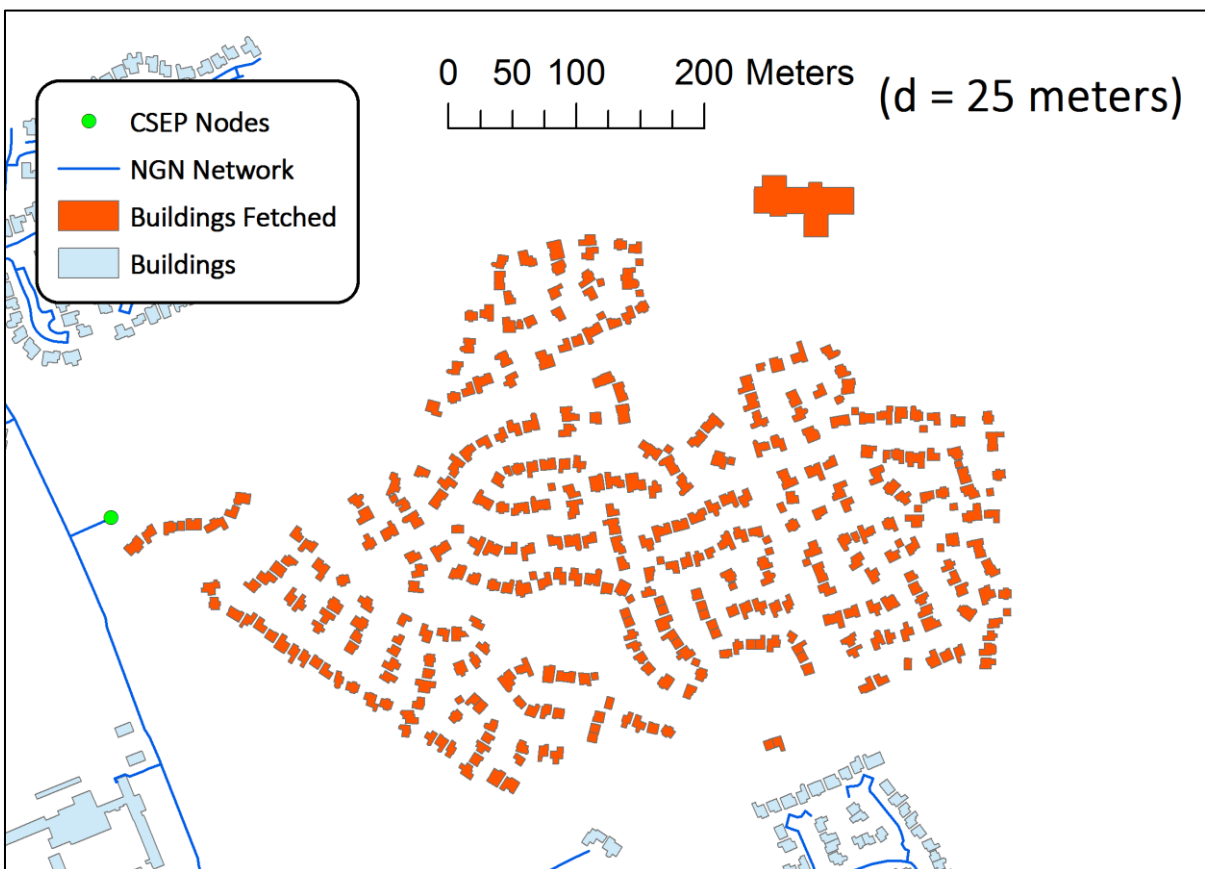
**Figure 5.14.** Area for explaining parameter sensitivity of  $d$  (Contains NGN Data © 2018).



**Figure 5.15.** Buildings fetched ( $d = 100$  meters) (Contains NGN Data © 2018).

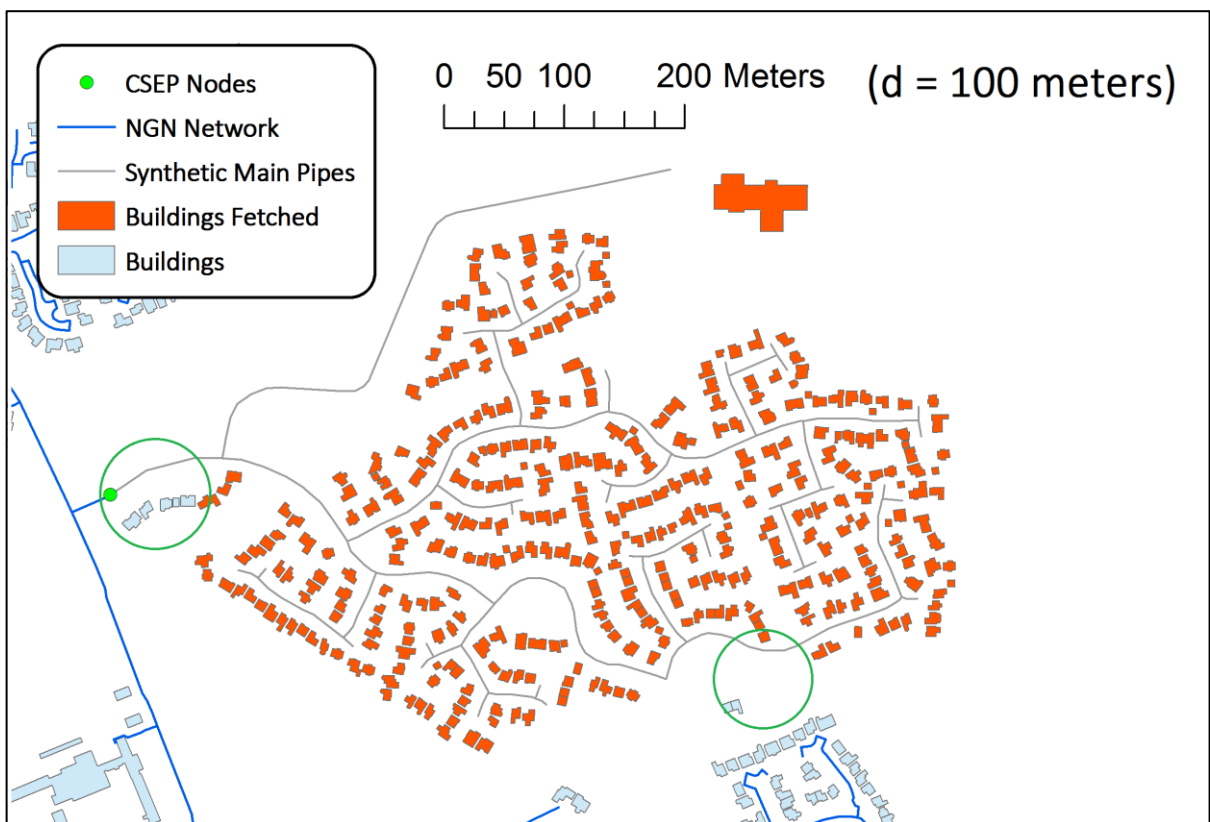


**Figure 5.16.** Buildings fetched ( $d = 50$  meters) (Contains NGN Data © 2018).

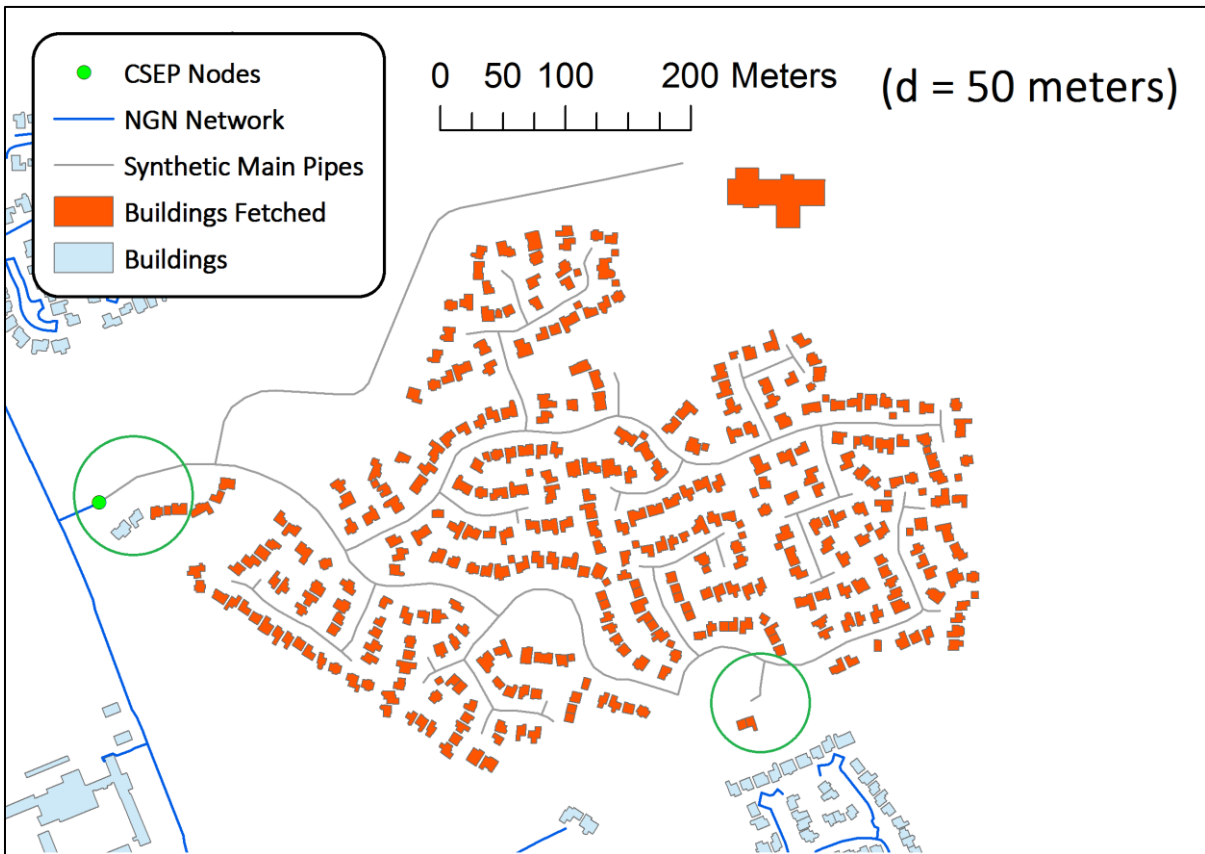


**Figure 5.17.** Buildings fetched ( $d = 25$  meters) (Contains NGN Data © 2018).

From figure 5.15, 5.16 and 5.17, it is easy to understand the number of buildings fetched will increase as  $d$  decreases. But such difference is very subtle. For example, when  $d$  is 100 meters, 373 buildings are fetched, and this number is 377 if  $d$  is 25 meters. When different number of buildings are fetched, number of road segments nearest to these buildings will also change. This will affect the sub network instance (serves as the synthetic main pipes) generated. Figure 5.18, 5.19, and 5.20 shows the different sub network instances generated with different  $d$  values. Differences in synthetic main pipes are minor, which are highlighted in green circles. The length of synthetic main pipes also increases as  $d$  decreases.



**Figure 5.18.** Synthetic main pipes ( $d = 100$  meters) (Contains NGN Data © 2018).



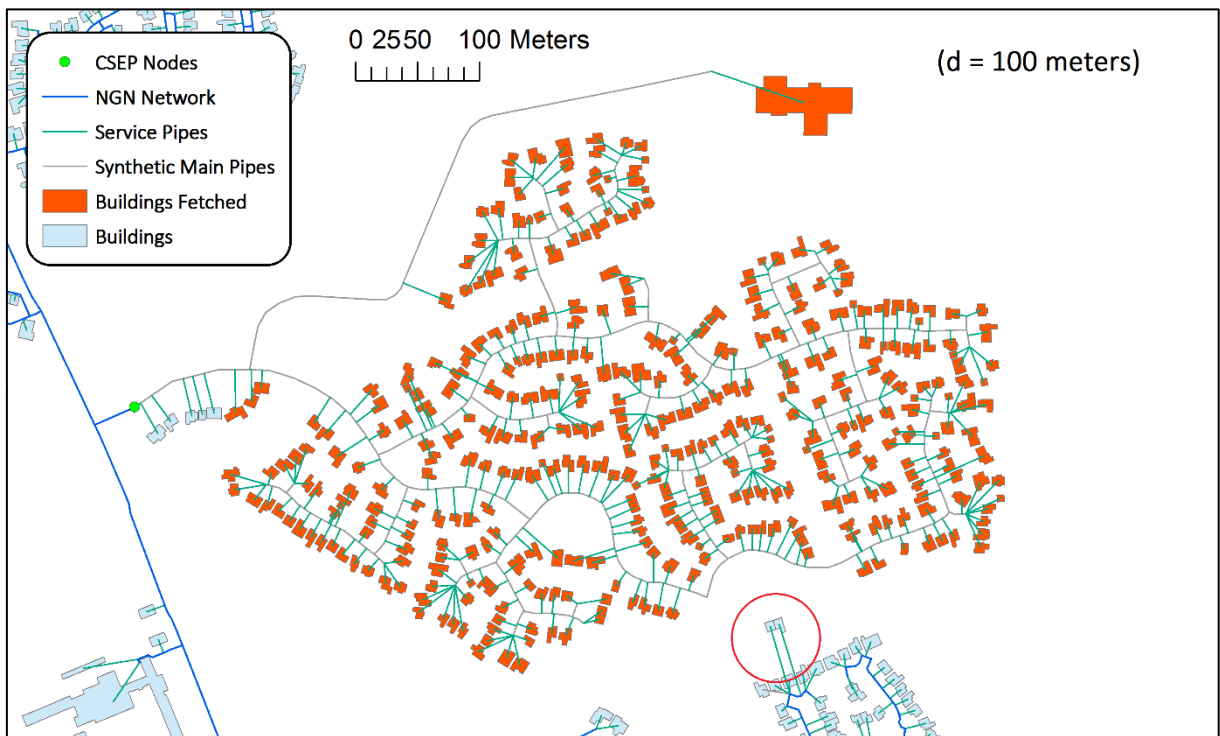
**Figure 5.19.** Synthetic main pipes ( $d = 50$  meters) (Contains NGN Data © 2018).



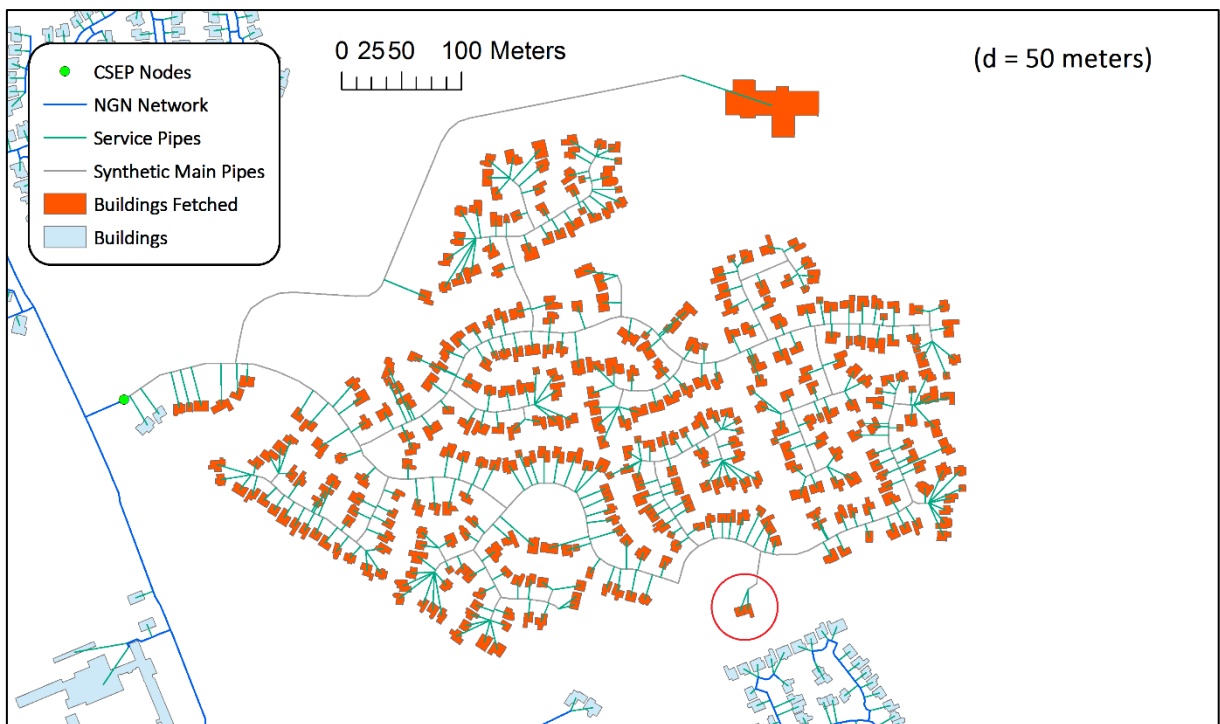
**Figure 5.20.** Synthetic main pipes ( $d = 25$  meters) (Contains NGN Data © 2018).



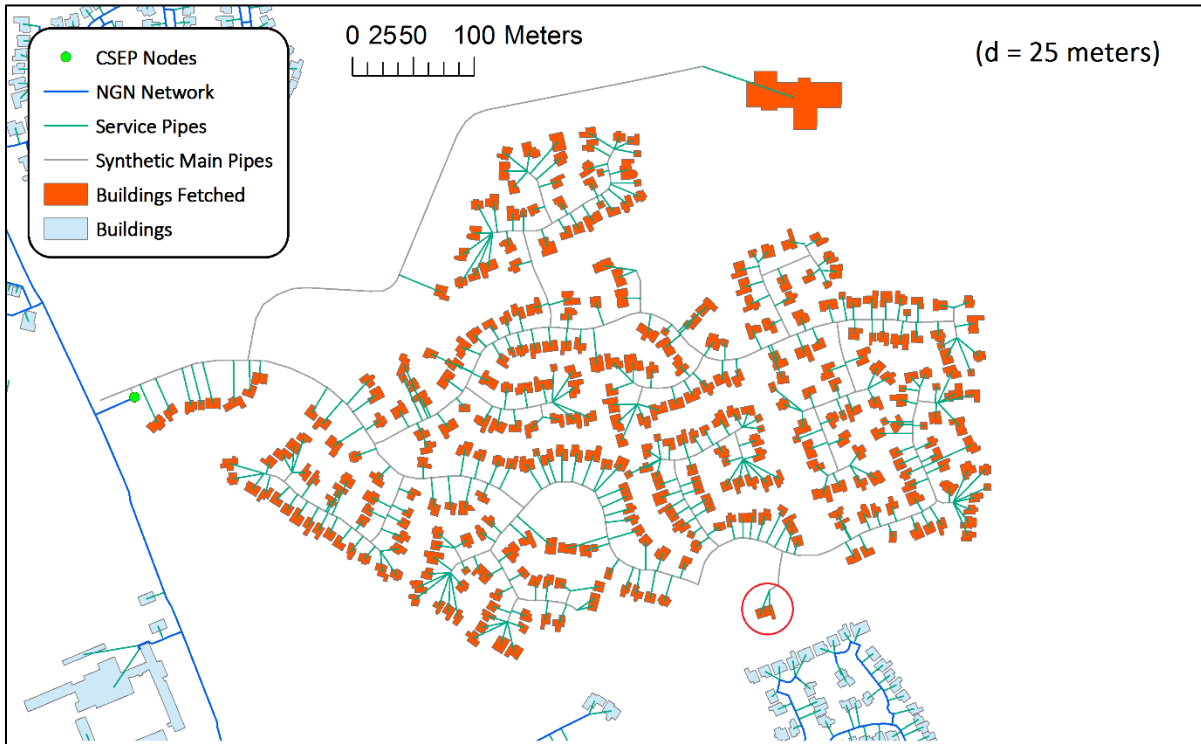
Finally, gas service pipes can be generated when gas main pipes are available (figure 5.21, 5.22, and 5.23). Note that when  $d$  increases, service pipe length can increase (red circle in figure 5.21, compared with 5.22 and 5.23).



**Figure 5.21.** Synthetic service pipes ( $d = 100$  meters) (Contains NGN Data © 2018).



**Figure 5.22.** Synthetic service pipes ( $d = 50$  meters) (Contains NGN Data © 2018).



**Figure 5.23.** Synthetic service pipes ( $d = 25$  meters) (Contains NGN Data © 2018).

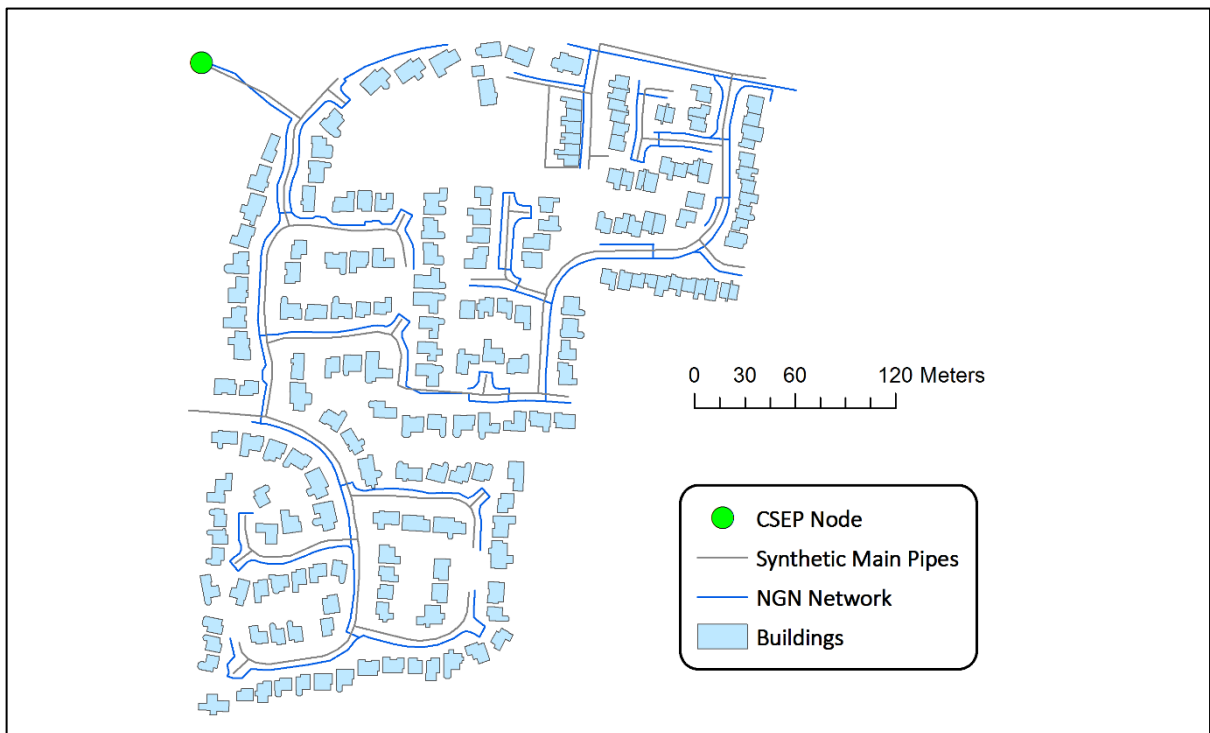
The six above figures (from figure 5.18 to figure 5.23) suggest that synthetic main pipes length will decrease as  $d$  increases, while the service pipes length will increase as  $d$  increases. As discussed in Chapter 4, when designing spatial layout of infrastructure networks, total length needs to be kept as small as possible, which corresponds to the length of gas main pipes and service pipes in this case. Therefore, these three  $d$  values are used to generate fine scale gas distribution networks in the entire city and measurement of pipe length is shown in table 5.1. To get shortest gas distribution networks in Newcastle upon Tyne, then 50 meters is a plausible value for  $d$ , which is the reason to use this value in this case study.

$d$	Length (main pipes)	Length (service pipes)	Length (total)
100 m	1,331,734 m	1,593,969 m	2,925,703 m
50 m	1,332,971 m	1,591,146 m	2,924,117 m
25 m	1,334,572 m	1,590,742 m	2,925,314 m

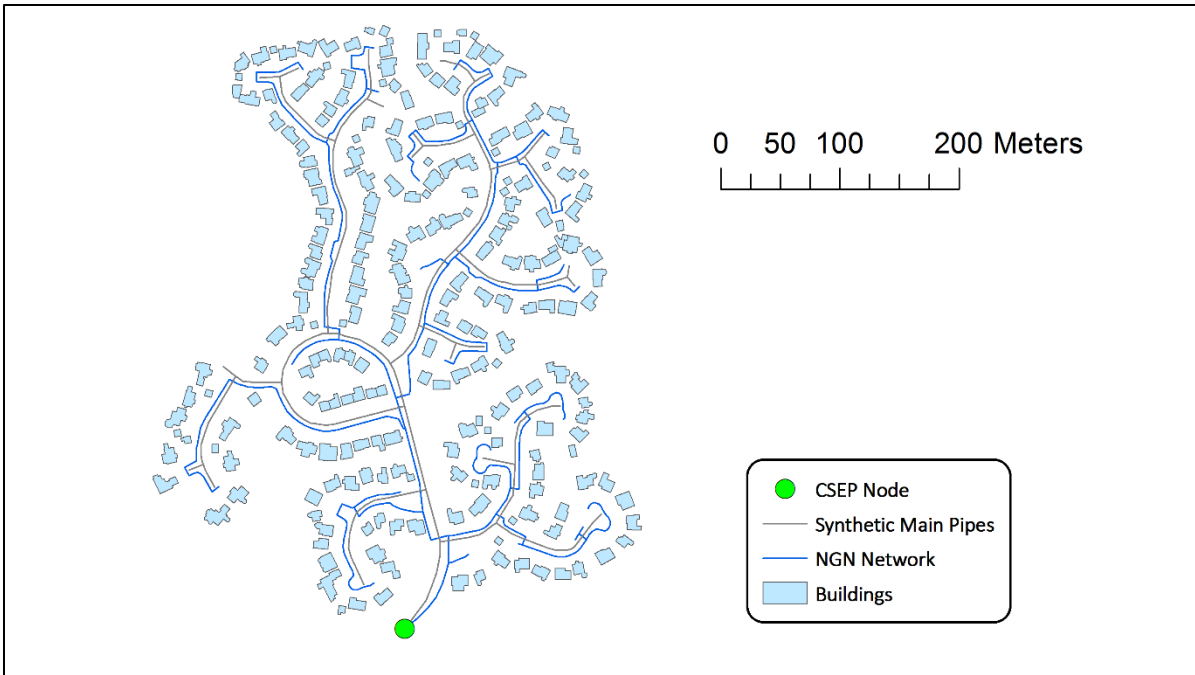
**Table 5.1.** Change of pipe total length as  $d$  changes.

### 5.2.5 Gas Network Validation

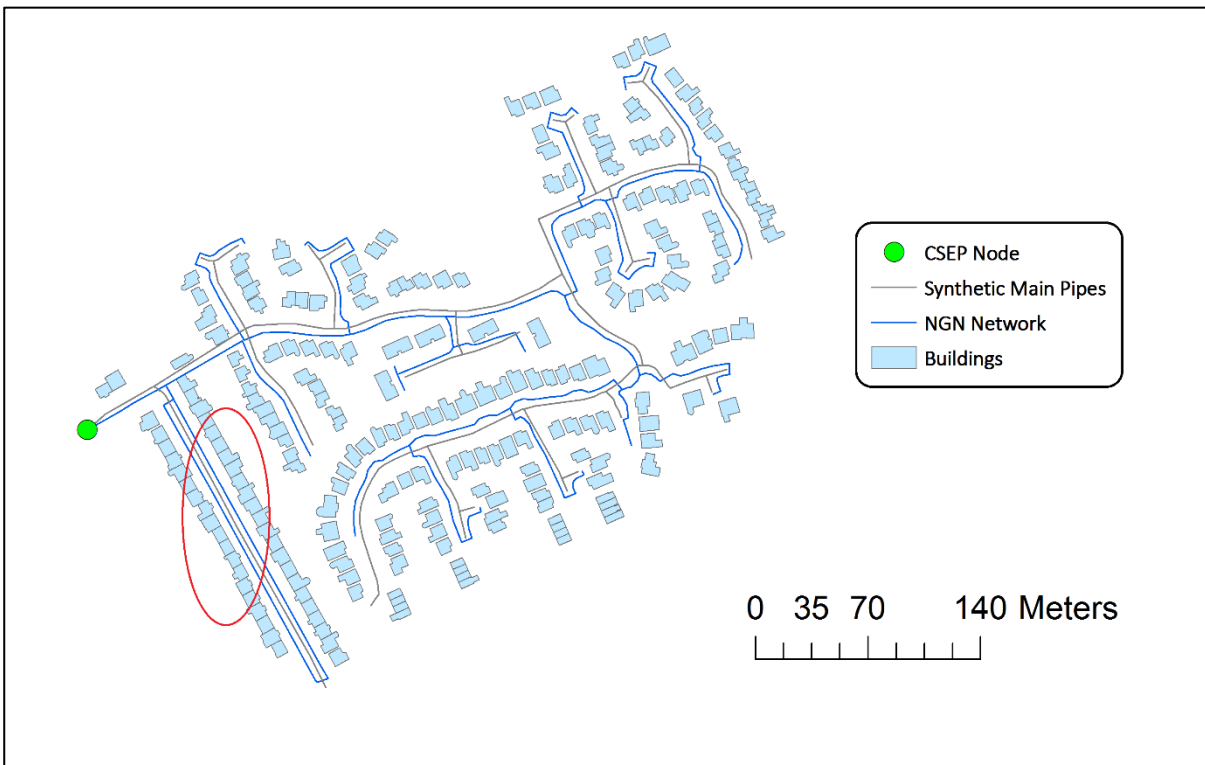
Section 5.2.2 discussed the approach for generating synthetic gas main pipes in areas without NGN network data. It is important to assess how accurate this approach is. Validation is difficult as main pipe network data are unavailable in these areas (which is why gas network infer algorithm is developed). However, there is still one way for validation based on available data: from the existing NGN network data, remove a small part from a CSEP node, and then generate the synthetic main pipe network, and validate it against the actual one. Three small areas are chosen and shown in figure 5.24, 5.25, and 5.26.



**Figure 5.24.** Validation area 1 (Contains NGN Data © 2018).



**Figure 5.25.** Validation area 2 (Contains NGN Data © 2018).



**Figure 5.26.** Validation area 3 (Contains NGN Data © 2018).

To validate the synthetic main pipes against the NGN network data, the error of commissions and error of omissions are used here (which were introduced in Chapter 4, for validating electricity feeders). The buffer distance is still 10 meters (same as chapter 4). The definition of

these two types of errors are shown in table 5.2. Validation result is shown in table 5.3.

<b>Error</b>	<b>Description</b>
Error of omissions	Buffer the NGN network data. The percentage of total length of synthetic main pipes that do not fall within the buffer.
Error of commissions	Buffer the synthetic main pipes. The percentage of total length of NGN network that does not fall within the buffer.

**Table 5.2.** Error of omissions and commissions for validate gas main pipes.

<b>Validation Area</b>	<b>Error of omissions</b>	<b>Error of commissions</b>
Area 1	2.7 %	8.9 %
Area 2	4.8 %	3.9 %
Area 3	3.4 %	7.6 %

**Table 5.3.** Validation result for the above three areas.

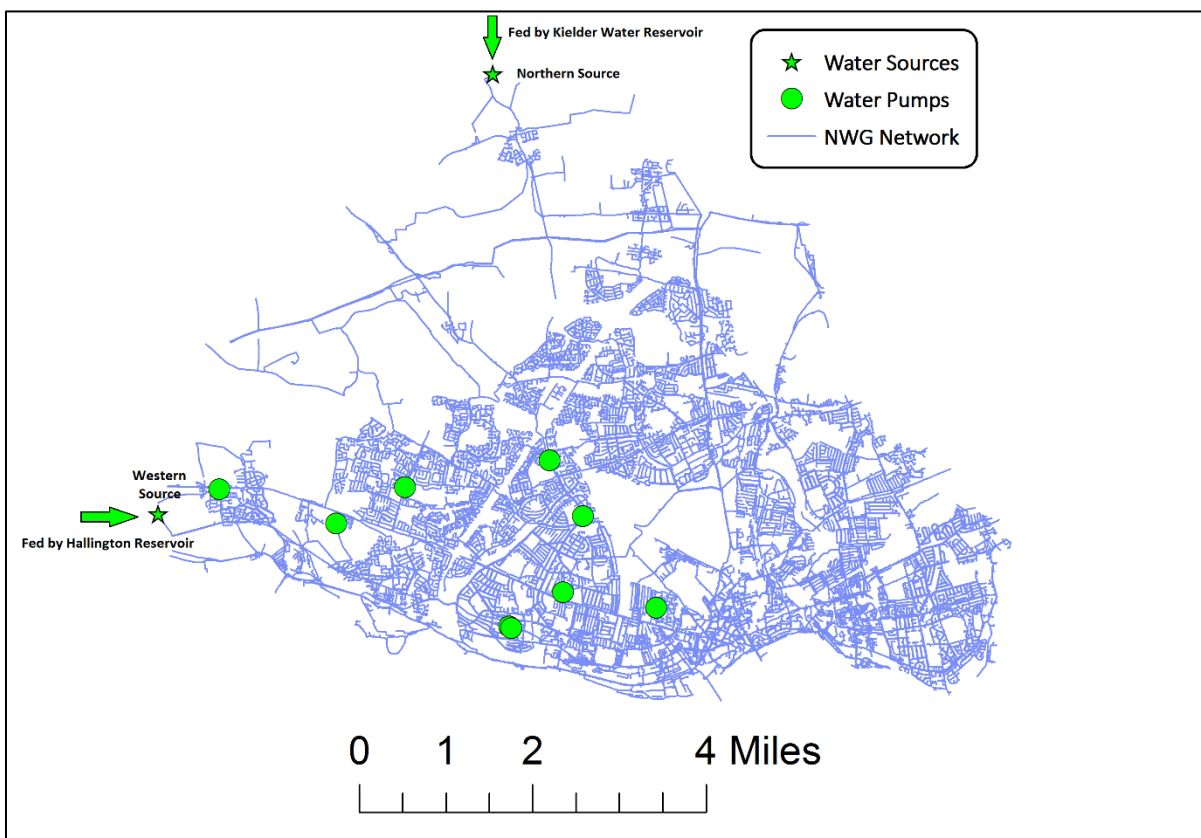
From table 5.3 it can be concluded that both types of errors are small within different validation areas, which suggests that the gas network infer algorithm (in section 5.2.2) is a good way to generate gas main pipes if actual data is unavailable. There is one interesting thing to note in the validation, especially in validation area 3 (the area within red circle). It is mentioned already that synthetic main pipes are generated using ITN network (more precisely, road centrelines). While in the red circle area in figure 5.26, the actual NGN network are paved along *both sides* of the road. This is considered to be major limitation of the current gas network infer algorithm, and future optimization can be done that (e.g. if there are two very lone terraces along both sides of a road, then two main pipes are generated along the road, instead of one).

### **5.3 Water Supply Network Integration**

Water supply network is a pipe-based network to deliver clean water from water source (such as treatment plant or reservoir) to individual buildings at desired pressure and quantity (Mays, 2000). It is generally a pressurized system. Water from the water source are normally first

pumped to a high location, such as water tower. Then due to pump head (Ostfeld, 2014), water is pressurized, and can be transported through the water supply network. In some areas, where water must be transported against pipe gradient, additional water pumping stations might be necessary to help re-pressurize water locally (Walski, et al., 2001). Like many utility companies, the water supply company (such as Northumbria Water Group, NWG, for the city of Newcastle upon Tyne) normally only keeps records for their water main pipes, and that additional servicing pipes are necessary to be generated, in order to construct a geospatial water distribution network, from source to buildings.

### 5.3.1 Water Supply Network Data

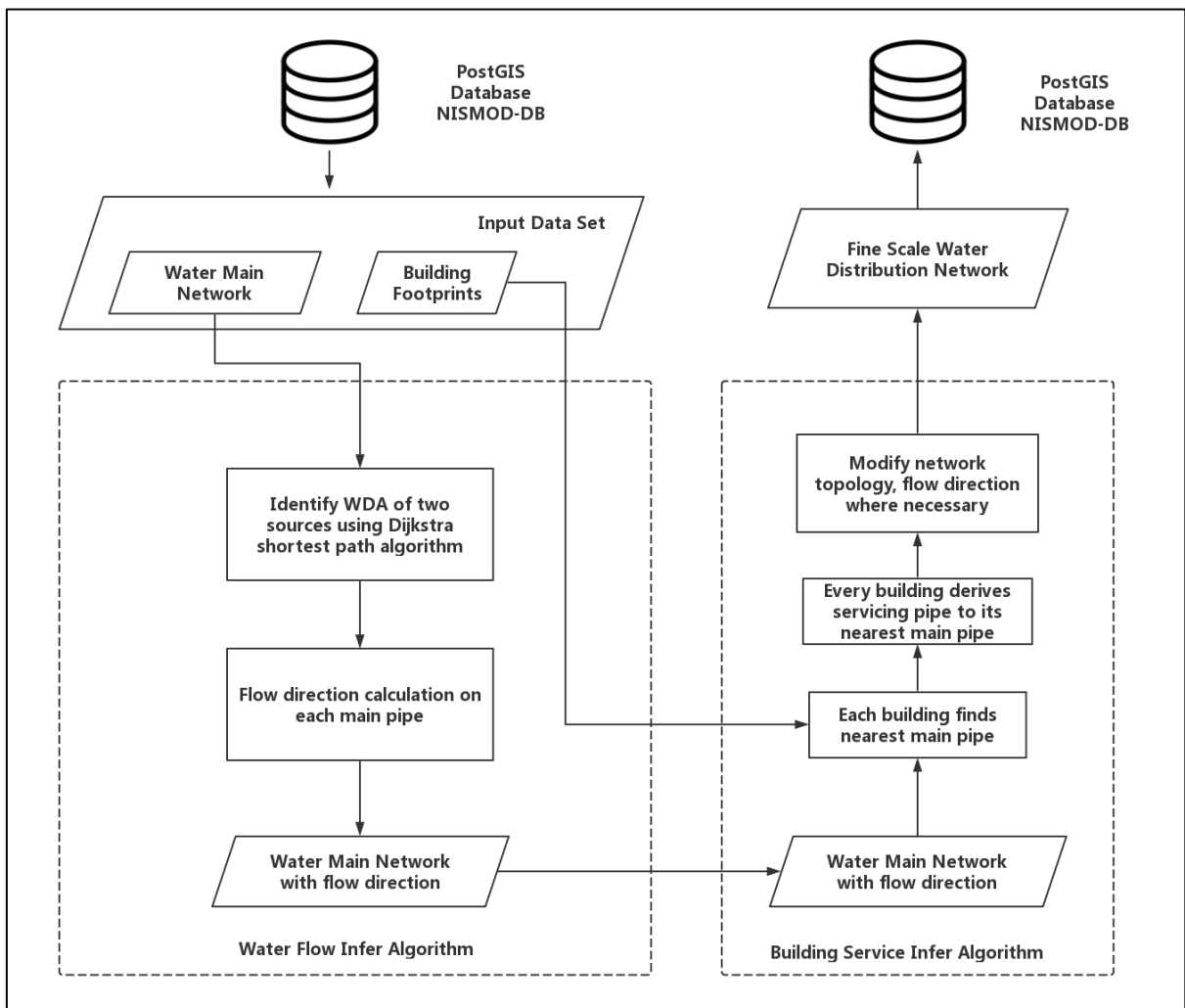


**Figure 5.27.** Water supply network data for Newcastle upon Tyne (Contains NWG Data © 2018).

With the assistance of the local water supply company, Northumbria Water Group (NWG), it is possible to access water supply network data in Newcastle upon Tyne. The data comprises of two shapefiles: a polyline data-set which records the layout of water supply pipes, and a

point data-set which records the layout of water sources and local water pumping stations. The water supply network from NWG is shown in figure 5.27. There are two water sources (*water service reservoirs*) and nine water pumping stations in the NWG network. The northern water source is fed by Kielder natural water reservoir from the north, and the western water source is fed by the Hallington natural water reservoir from the west. It is worth noting the water sources are fed by their own water reservoirs via large diameter water transmission pipes, but due to data unavailability, these pipes are not visually displayed.

The NWG network data does not contain nodes (junctions of pipes), only the layout of pipes. But it is still possible to infer the location nodes based on spatial connectivity of pipes. The work is done via NetworkX library (NetworkX, 2018). After the node generation process, it is found that NWG network data contain 36,806 nodes and 39,282 edges.



**Figure 5.28.** General work flow of water supply network integration.

Before the full water supply network can be generated that includes service pipes connecting main pipes and individual buildings, the flow direction of the main network needs to be inferred, because such information is not available from the actual data.

Figure 5.28 shows the general work flow of the water supply network integration. It is based on two major algorithms: water flow infer algorithm and building service infer algorithm. First input data (NWG network and building footprints) are read from PostGIS database, the NWG network data is processed via the water flow infer algorithm and encoded with flow direction. After that, water service pipes will be generated to connect buildings and NWG network (the main pipes). The generated fine scale water distribution network will be finally written back to the PostGIS database. Details of these two algorithms will be discussed in the next two sections.

### ***5.3.2 Water Flow Infer***

NWG network data is a single connected network instance with two water sources (service reservoirs). However, to make the water supply work function properly, a special type of valve called gate valve is used to shut off some pipes in order to partition the water supply network into several water distribution areas (WDAs) (Mays, et al., 2000). The number of WDA is equal to the number of water sources. For each WDA, it is served by one water source. Since the goal of this major section (5.3) is to generate fine scale water distribution networks, understanding dependency from building to infrastructure assets (water sources) is important. Therefore, WDAs must be identified for Newcastle upon Tyne.

However, NWG does not record the spatial location of any value in their water supply network data (including the gate valves), which means it is not possible to deterministically derive the WDAs for Newcastle upon Tyne. Therefore, a heuristic approach (contained in the flow infer algorithm) was developed to infer WDAs in Newcastle upon Tyne. After that, flow direction needs to be inferred for each WDA.

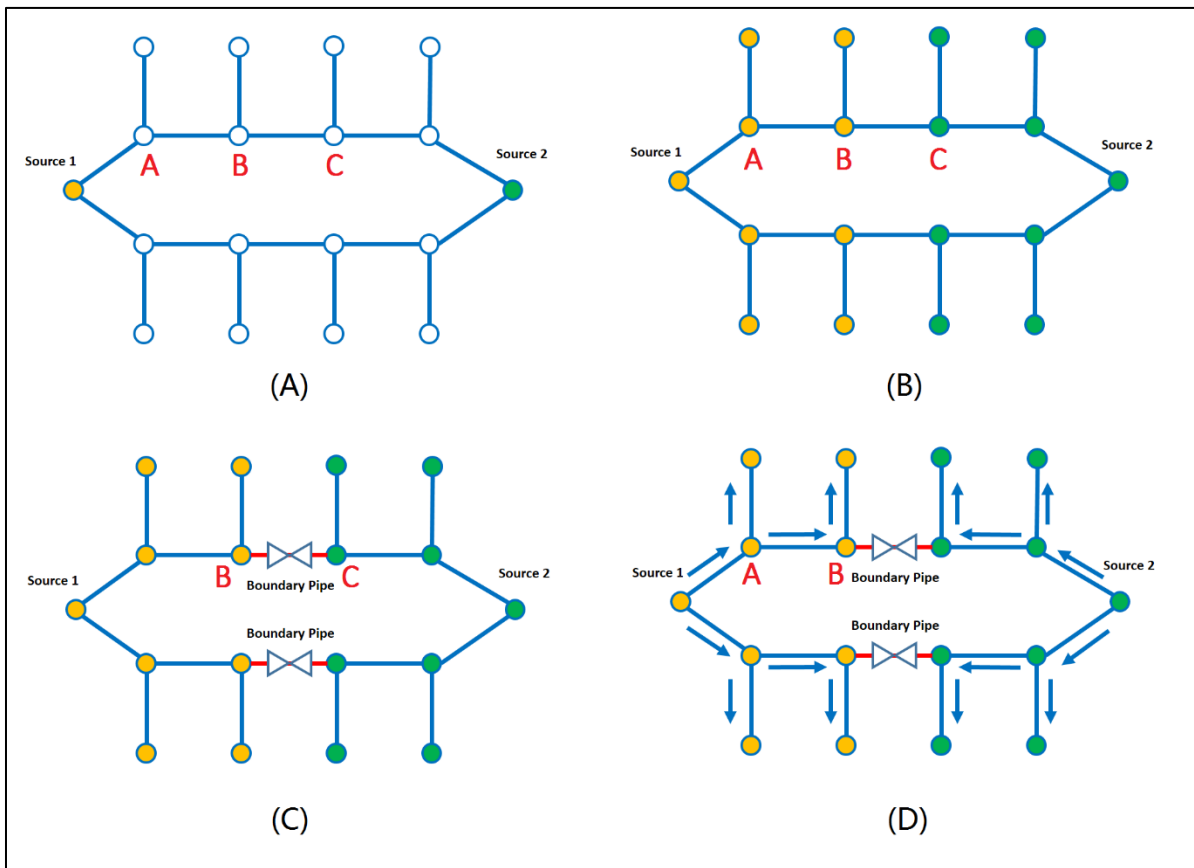


Inferring WDAs from NWG network data is actually a graph partition problem, which aims to partition a single connected network instance into several sub-connected components.

Traditional graph partition algorithms include the Label Propagation algorithm (Zhu et al, 2002), Kernighan–Lin algorithm (Lin et al, 1973) and Fiduccia-Mattheyses algorithm (Fiduccia et al, 1982), which can all solve the bipartition problem where one graph is divided into two sub-graph components, based on the assumption that size (node number) in each sub-graph is almost equal and the total weight of edges connecting the two sub-graphs are kept as small as possible. However, these classic algorithms are not suitable for solving this specific WDA problem in Newcastle upon Tyne, since they are designed for graphs without special nodes. In our case, it is natural to consider that water source nodes must belong to different partitions (which means it is a constraint to partition the graph).

To address this specific WDA problem for the water supply network, Ferrari et al (2011) put forward an optimization algorithm for automatically partitioning water supply network into multiple distribution areas (can be more than 2). The optimization algorithm requires running a hydraulic model on the water supply network, requiring additional attributes of water source volumes, source pump head, pressure within the water network and water pipe diameter (Ferrari, et al, 2011).

Currently, the NWG network data only provides geometry layout of water pipes (as polylines), without additional pipe information (e.g. pipe diameter). That means resolving hydraulic equation is not possible. However, even in such case, Ferrari et al (2011) suggested that it is still possible to partition a graph using Dijkstra shortest path algorithm, where the weight is the length of each pipe. Therefore, water flow inference algorithm is developed based on this idea, figure 5.29 is an example to show how the algorithm works and listing 5.3 is the algorithm pseudo code.



**Figure 5.29.** Simple example of the water flow infer algorithm.

---

**Algorithm: Water Flow Infer Algorithm**

---

**Input: Raw Water Main Network  $G_{raw}$  with two source  $s_1, s_2$** **Output: the Water Main Network  $G_{flow}$  with flow direction**

```
1:  $G_{raw} = G(E, V)$  where  $E$  is the set of edges, and  $V$  is the set of nodes
2: for  $node \in V$ , calculate whether it is closer to  $s_1$  or  $s_2$  via Dijkstra path
   distance, assign the node to that source
3: create two new sets  $V_1$  and  $V_2$ , to store nodes assigned to  $s_1$  and  $s_2$ ,
   respectively
4: create three new sets  $E_1, E_2, E_{bound}$ 
5: for  $e \in E$  do
6:   if both bounding  $nodes \in E_1$  then
7:     put  $e$  into  $E_1$ 
8:   else if both bounding  $nodes \in E_2$  then
9:     put  $e$  into  $E_2$ 
10:  else
11:    put  $e$  into  $E_{bound}$ 
12:  end if
13: end for

14: for  $e \in E_1$  do
15:   let  $n_1, n_2$  be its bounding nodes
16:   let  $dist_1$  and  $dist_2$  be the distances from these nodes to  $s_1$ 
17:   if  $dist_1 < dist_2$  then
18:     direction on  $e$  is from  $n_1$  to  $n_2$ 
19:   else
20:     direction on  $e$  is from  $n_2$  to  $n_1$ 
21:   end if
22: end for

23: for  $e \in E_2$  do
24:   let  $n_1, n_2$  be its bounding nodes
25:   let  $dist_1$  and  $dist_2$  be the distances from these nodes to  $s_2$ 
26:   if  $dist_1 < dist_2$  then
27:     direction on  $e$  is from  $n_1$  to  $n_2$ 
28:   else
29:     direction on  $e$  is from  $n_2$  to  $n_1$ 
30:   end if
31: end for
32: use set  $V_1, V_2, E_1, E_2, E_{bound}$  to construct a new graph instance  $G_{flow}$ 
```

---

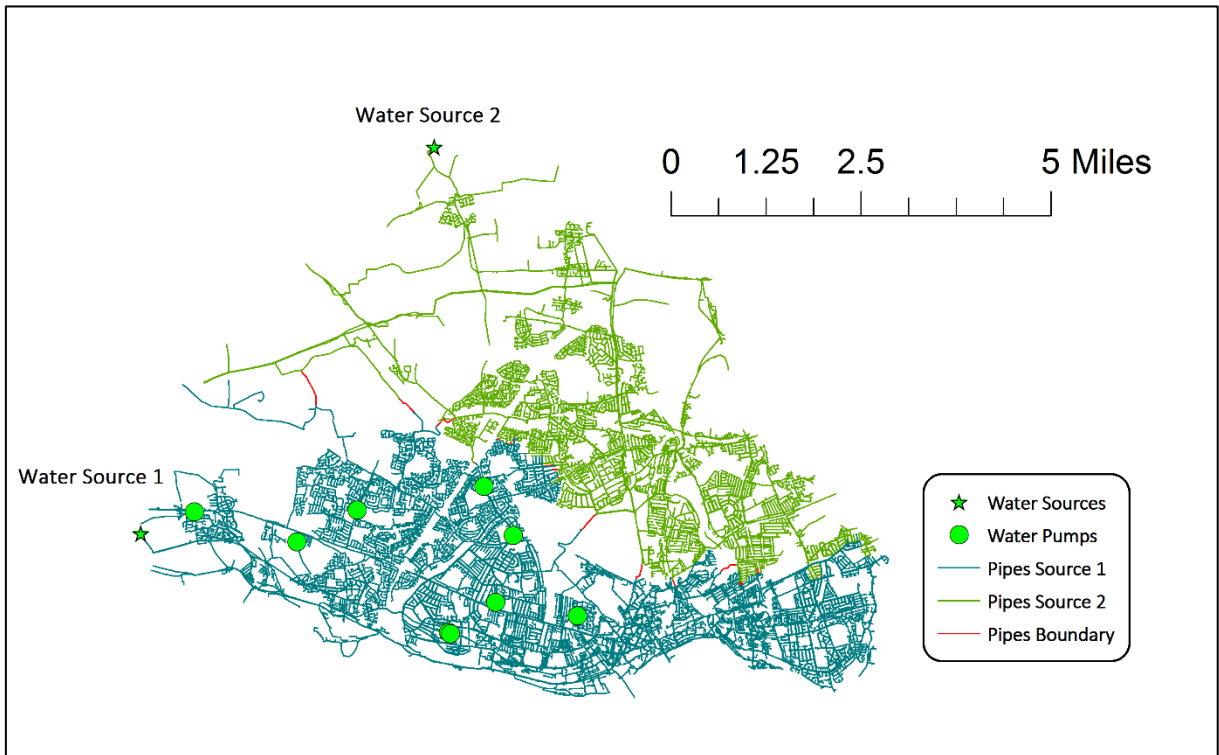
**Listing 5.3.** Pseudo code for water flow infer algorithm.

Figure 5.29 shows a simple area with two water sources and 18 pipes. Length of each pipe is the same (100 meters). Water flow infer algorithm basically can be divided into two big steps: (1) WDA identification, and (2) flow direction infer.

The first step (line 1-13 in listing 5.3): for the source 1 and 2, for each node, a calculation will be done to see if it is closer to source 1 and source 2 via Dijkstra path distance. The node is assigned a source closer to it. For example, in sub plot (A), the node A is closer to source 1 (distance is 100 meters) than to source 2 (distance is 200 meters), and therefore node A is assigned to source 1. This allows for assigning a source to each node (sub plot (B)). After that, for an edge, if its two connecting nodes are assigned different sources, this edge (pipe) is considered to be a boundary pipe, in which there is gate valve to shut it off. For example, the edge B-C is a boundary pipe in sub plot (C), and therefore there is no water within edge B-C. When boundary pipes are identified, the two WDAs are naturally generated.

Then algorithm will move to the second step (line 14 – 32 in listing 5.3), which is inferring flow direction for each pipe which is not a boundary pipe. In particular, based on the specific water source that pipe belongs to, Dijkstra path distance will be calculated from that water source to both bounding nodes of that pipe. The flow direction is defined from the node having a shorter distance to the node having a longer distance. For example, in figure 5.29, in sub plot (D), the water flow direction on the edge A-B is inferred to be from A to B, because A (distance is 100 meters) is closer than B (distance is 200 meters) to source 1.

Following this strategy, it is possible to infer the flow direction on the entire NWG network data, the result is shown in figure 5.30. Of all the 39,282 pipes in the water main network, 27,800 of them are served by water source 1 and 11,443 of them are served by water source 2. The other 39 pipes are considered as the boundary pipes, which are served as the boundary between WDA 1 and WDA 2.



**Figure 5.30.** WDA representation for Newcastle upon Tyne. (Contains NWG Data © 2018)

### ***5.3.3 Water Distribution Network Generation***

Once the flow direction on the NWG network is inferred, fine scale water distribution network can be generated. The building service infer algorithm (water) is developed to generate the service pipes connecting buildings and water main pipes (listing 5.4). This algorithm is similar to the building service infer algorithm (gas) in section 5.2.3. However, there is a small difference. A constraint is made that service pipes cannot connect boundary pipes, since there is no flow in them. Figure 5.31 shows an example of the final water supply network for a small part of Newcastle upon Tyne.

---

**Algorithm : Building Service Infer Algorithm - Water**

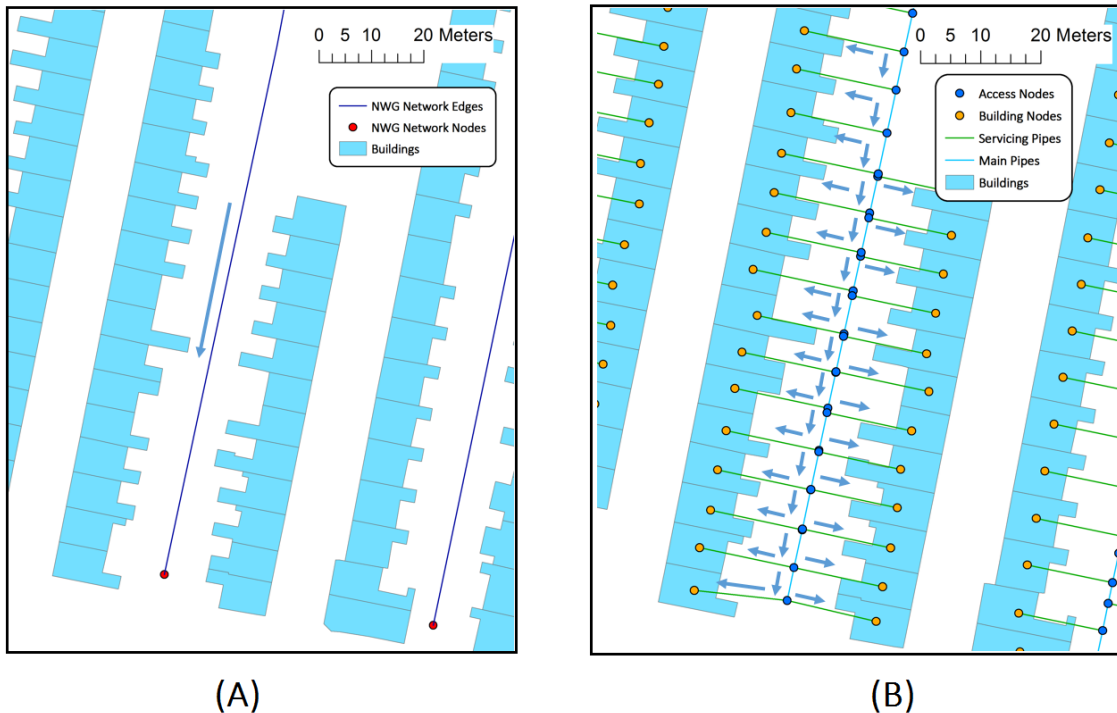
---

**Input:** a set of Buildings  $B$ , Water Main Network with flow direction  $G_{flow}$

**Output:** Water Distribution Network  $G_{dis}$

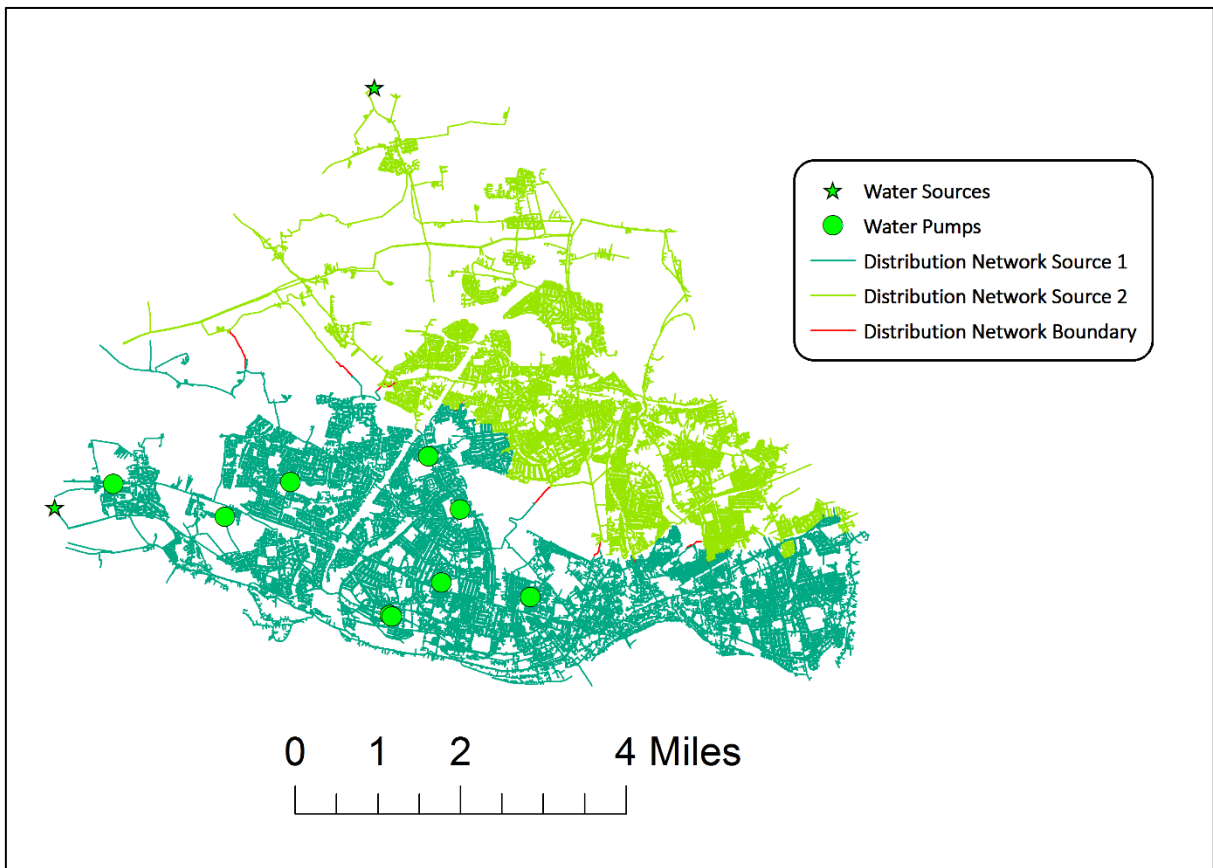
- 1: for  $b \in B$ , find nearest pipe from  $G_{flow}$  that is not a boundary pipe as its access pipe
  - 2: for  $b \in B$ , extract its centroid  $b.cen$  and then derive an edge to its access pipe
  - 3: merge all the building nodes and derived edges (servicing pipes) to  $G_{flow}$  to save to  $G_{dis}$
  - 4: on  $G_{dis}$ , modify topology where necessary, and record flow direction on the service pipes
- 

**Listing 5.4.** Building service infer algorithm (water).



**Figure 5.31.** (A) Water main pipe network, with flow directions. (B) Water distribution network to the buildings, with flow direction calculated (Contains NWG Data © 2018).

Figure 5.32 shows the water distribution network generated for the entire city of Newcastle upon Tyne. The whole distribution network contains 238,951 nodes and 241,436 edges, servicing 104,855 buildings in the city. Among all the edges, 156,762 of them are served by water source 1, and 84,635 are served by water source 2, and 39 edges are the boundary pipes.



**Figure 5.32.** Fine scale water distribution networks (including service pipes) in Newcastle upon Tyne (Contains NWG Data © 2018).

Generally speaking, validation is needed to assess the quality of the data generated (fine scale water distribution networks), especially the WDAs generated and the flow direction inferred on the NWG network. However, until the completion of this PhD, such information is still not publicly accessible from NWG data portal (the only data available are the layout of NWG network, without additional information on the pipes). Therefore, validation is not carried out within the water supply network. If possible, future work will focus on trying to accessing actual data for water supply network validation.

#### 5.4 Sewer Network Integration

The sewer network is a pipe-based network system to collect and transport domestic waste water from each individual building to the specific facilities that can treat the waste water, such as waste water treatment plants (Hammer, 1986). Pipes are connected with either

manhole (inspection chamber) or simply with a pipe junction. The entire sewer network is generally a gravity-based system (Halfawy, et al., 2008), and therefore without external pressure, the waste water can flow from upstream location to the downstream location. In some mountain areas, where waste water must be transported to higher places (against gradient), sewer pumping stations are set up to pressurize the waste water (Guisasola, et al., 2008).

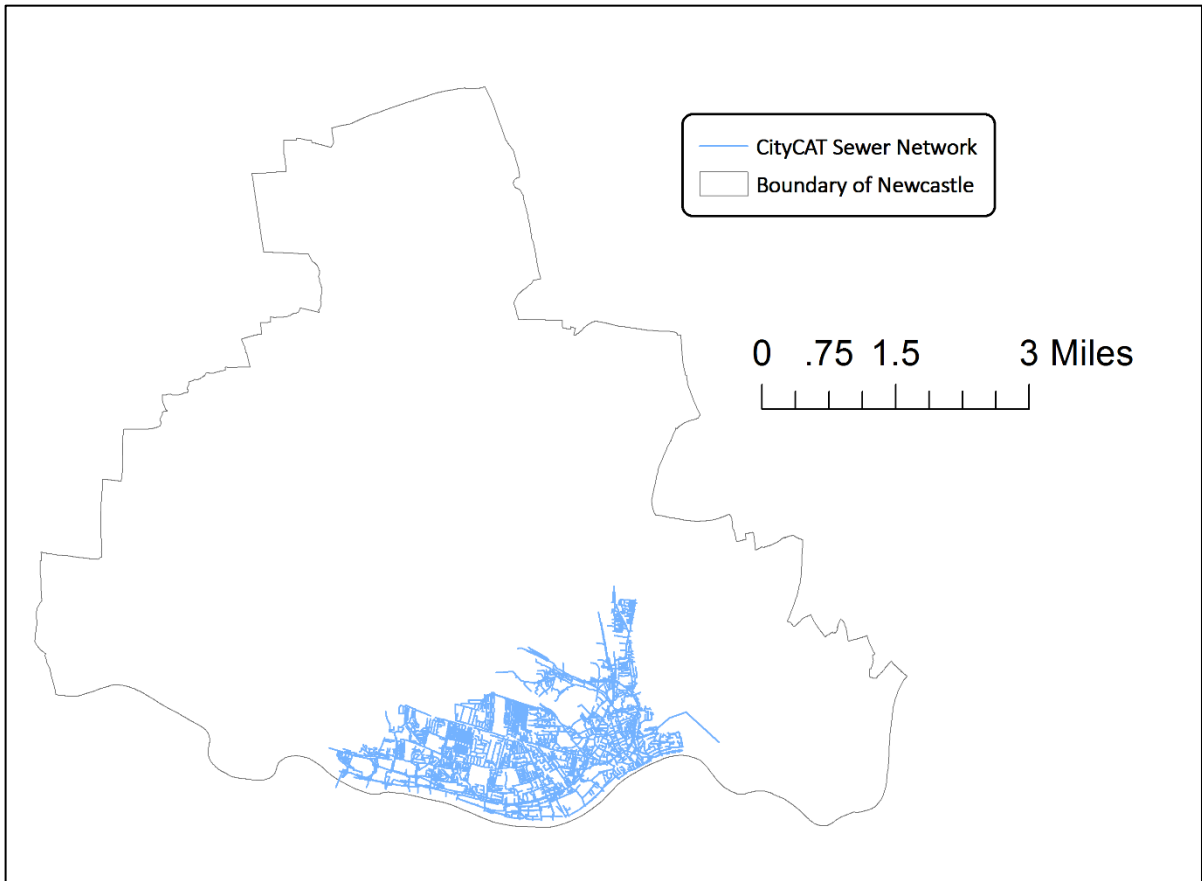
The biggest difference between the sewer and other utility network (electricity, gas, and water supply), is the flow direction within the network. For the other three types of the network, buildings are the sink nodes where infrastructure service is provided to. In the sewer network, buildings are actually the source nodes, where flows are generated.

#### ***5.4.1 Sewer Network Data***

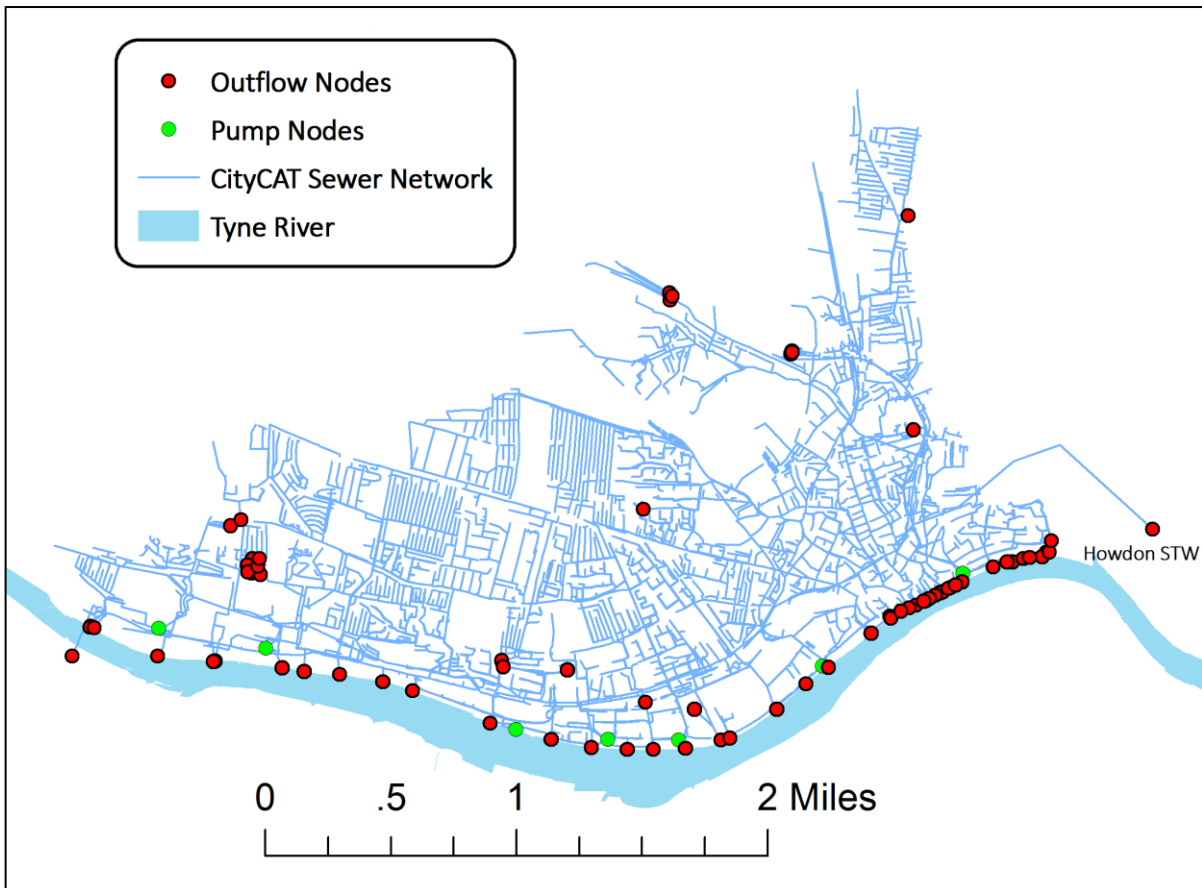
NWG (Northumbria Water Group), the same company for managing water supply network, also manages sewer network for the city of Newcastle upon Tyne. Requests have been made to the NWG to access spatial layout of sewer network data. However, until the completion of this PhD, such data is not available from their spatial data portal. As an alternative solution, the sewer network data used in this section is sewer network model, generated from the CityCAT project (Bertsch, et al., 2017).

The data comprise of two shapefiles, where one is the layout of the sewer main pipes, and the other one contains nodes connecting the main pipes. The sewer network model to date, covers not the entire city but only its central part. Figure 5.33 shows the covered area of sewer network compared with the boundary of the city Newcastle upon Tyne.





**Figure 5.33.** Available sewer network data (CityCAT Model) for Newcastle upon Tyne.



**Figure 5.34.** Location of the pumps and outflow nodes in CityCAT sewer network.

The CityCAT sewer network contains 8132 nodes and 8306 edges. 8048 of the nodes are manholes, 7 are sewer pumping stations and 77 are outflow nodes (where waste water exits the network). The location of these special nodes is shown in figure 5.34. Each network node or edge has a specific ID, and flow direction on the sewer network has been given across the entire network, by specifying the upstream and downstream node for each edge. It is worth noting that in Newcastle, storm runoff and domestic waste water are both transported using the same sewer network system (Bertsch, et al., 2017). When the waste water exits the network, normally it arrives at the Tyne river, or at a major sewer treatment plant (Howdon STW) in the east of the city.

#### ***5.4.2 Fine Scale Sewer Network Generation***

Since CityCAT sewer network only covers one part of Newcastle upon Tyne, not the entire city, a key question is whether it is possible to infer layout of sewer main pipes where there is no existing data. In section 5.2.2, work has been done to complete the NGN gas network, based on road network.

However, such approach is not plausible in this situation. The major reason is that, layout of key infrastructure assets (manholes, outflow nodes) is not available across the entire city. The algorithm used in 5.2.2 (as well as the one discussed in Chapter 4) assumed that knowledge on infrastructure asset is complete, and aims to generate the layout pipes or cables connecting these assets. These algorithms are not able to *guess* the location of infrastructure assets.

Therefore, with regards to CityCAT sewer network, no data generation work will be done on it. Only the current CityCAT sewer network is used to generate fine scale sewer network (connecting buildings with sewer service pipes). Figure 5.35 shows the general workflow of integrating buildings to the sewer network model. Data (sewer network and buildings) is read from a PostGIS database and processed via a building service infer (sewer) algorithm (listing 5.5). This algorithm generates layout of sewer service pipes connecting a building and a sewer

main pipe. Fine scale sewer network is then generated and written back to the PostGIS database.

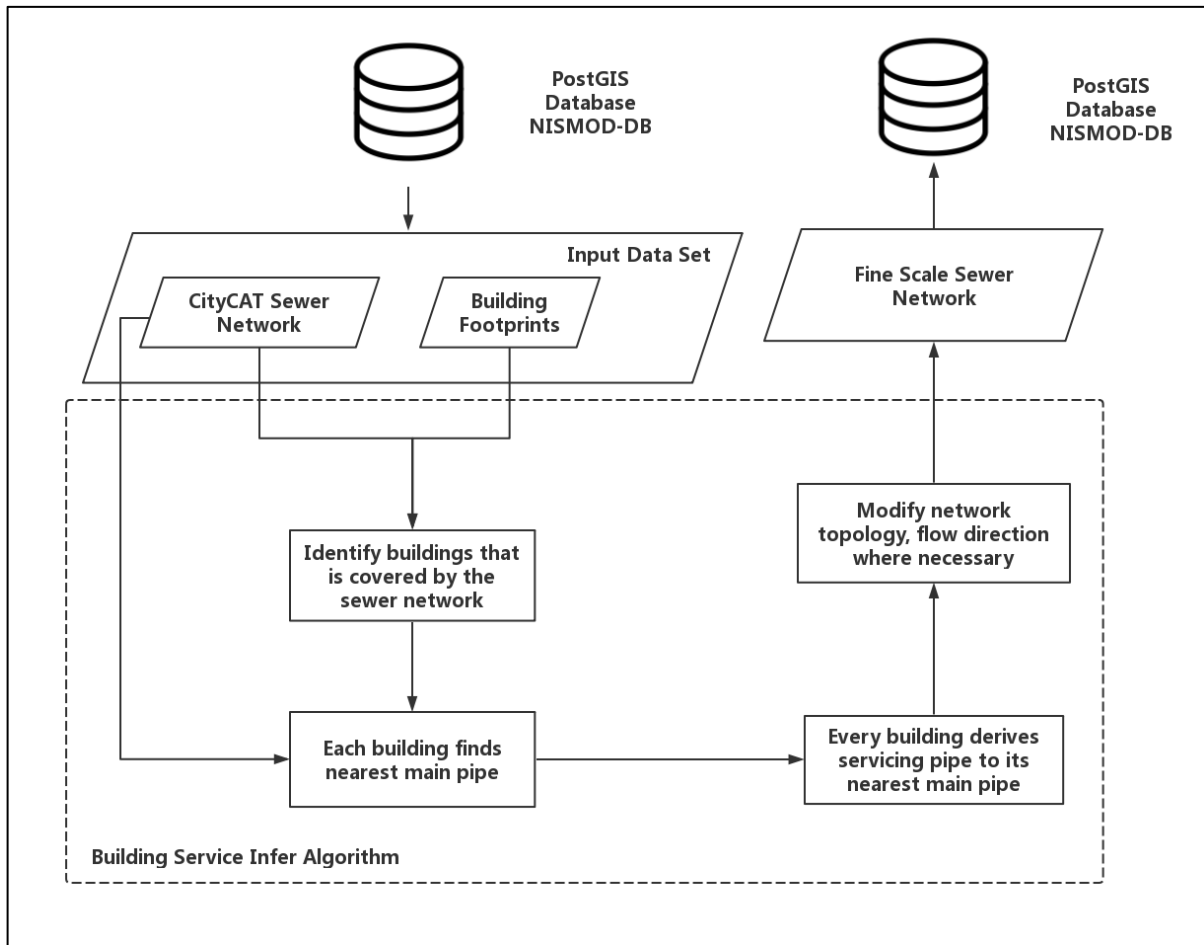


Figure 5.35. General work flow for sewer network integration work.

---

**Algorithm :** Building Integration Algorithm - Sewer

---

**Input:** a set of buildings  $B$ , sewer network model with flow

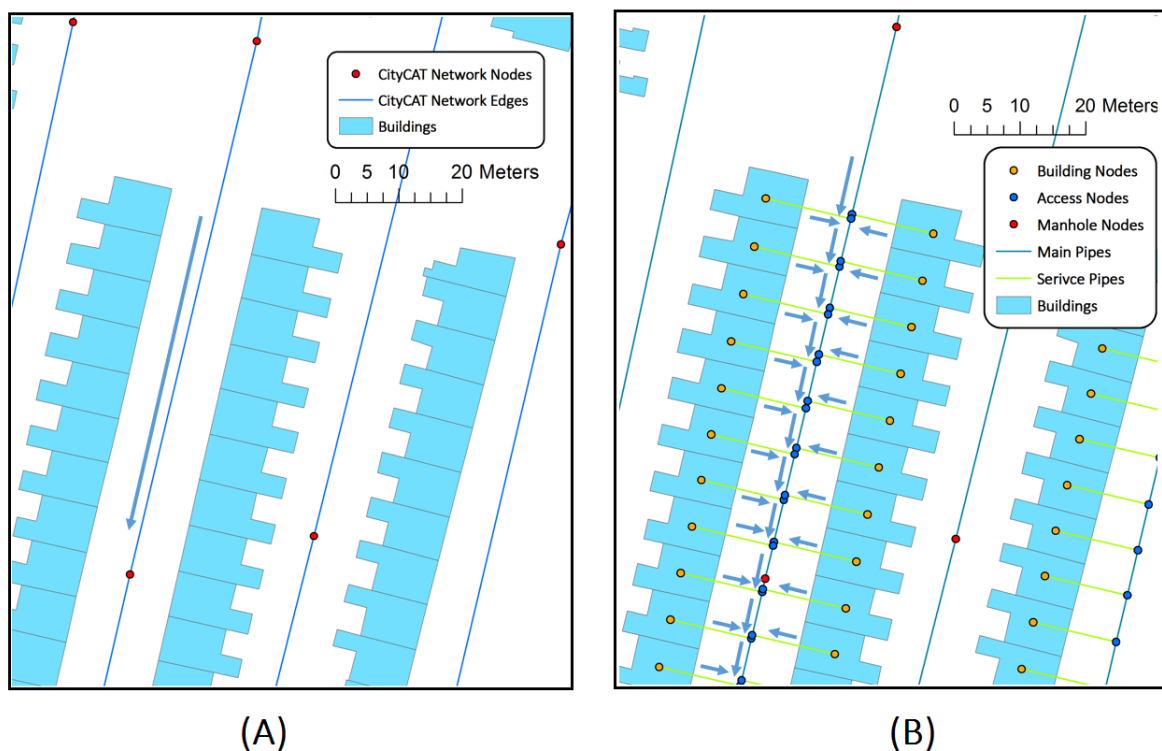
$G_{flow}$

**Output:** fine scale sewer network model  $G_{fine}$

- 1: set up a threshold distance  $d$
  - 2: find subset  $B_{subset}$  of  $B$ , so that for  $b \in B_{subset}$ ,  $\text{distance}(b, G_{flow}) < d$
  - 3: for  $b \in B_{subset}$ , find nearest pipe from  $G_{flow}$
  - 4: for  $b \in B_{subset}$ , extract its centroid  $b.cen$  and then derive an edge to its access pipe
  - 5: merge all the building nodes and derived edges (servicing pipes) to  $G_{flow}$  to save to  $G_{fine}$
  - 6: on  $G_{fine}$ , modify topology where necessary, and record flow direction on the service pipes
- 

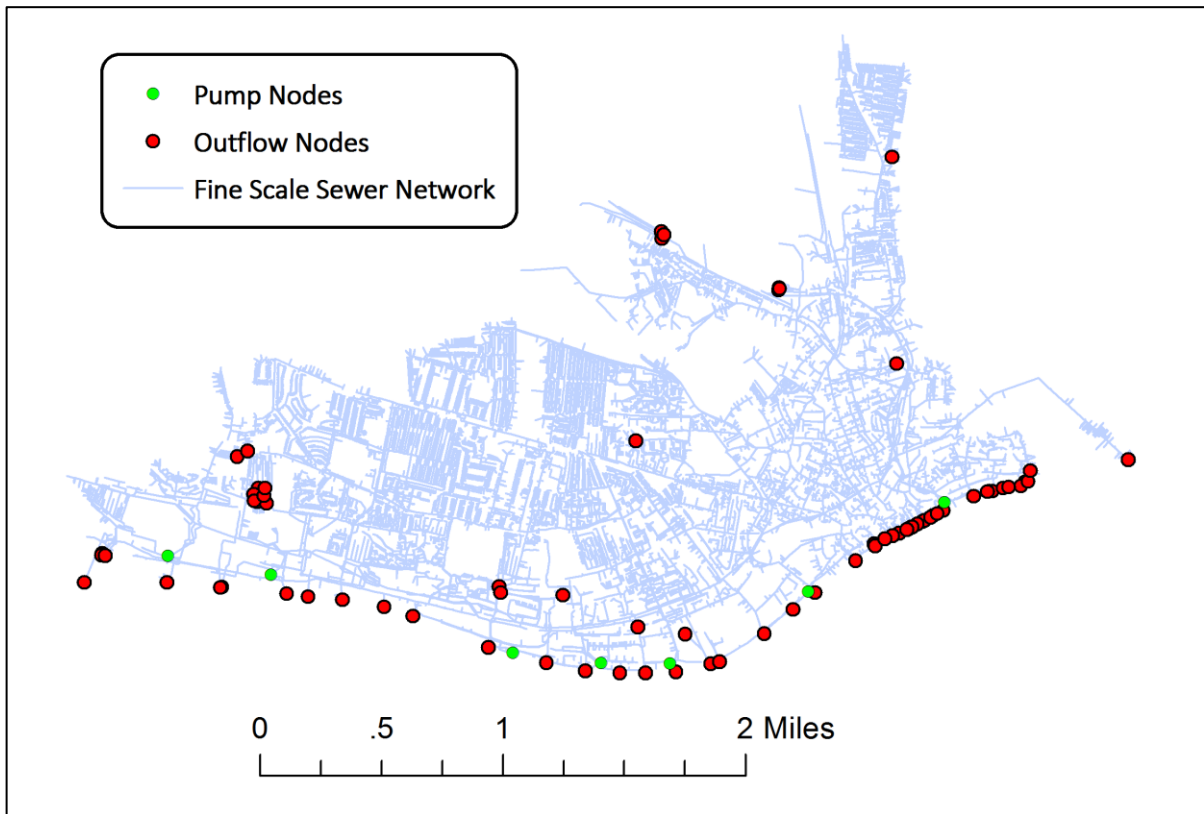
Listing 5.5. Pseudo code for the building service infer algorithm (sewer).

The building service infer algorithm (sewer) starts from applying the local search strategy, to find only the nearby buildings of the sewer network model. This is achieved by setting up a threshold distance  $d$  of 50 meters (same reason for the parameter  $d$  in section 5.2.2). In this case, of all the 104,855 buildings in Newcastle upon Tyne, 13,882 of them are fetched to be served by the sewer network. After that, remaining work is to derive service pipes which connect the chosen buildings to the sewer main pipes. Finally, all these building nodes, additional service pipes are merged to the sewer network to generate fine scale sewer network. Figure 5.36 shows the service pipe infer process, and note that flow direction on the service pipes is opposite (compared with gas and water supply servicing pipes).



**Figure 5.36.** (A) Sewer main network, with flow directions. (B) Fine scale sewer network with buildings integrated.

Figure 5.37 shows the overview of the fine scale sewer network generated (that contains sewer service pipes), which contains 34,225 nodes and 34,375 edges, serving 13,882 buildings.



**Figure 5.37.** Fine scale sewer network generated, which includes sewer service pipes.

#### 5.4.3 Sewer Network Flow Infer

The CityCAT sewer network model contains an essential attribute, which is waste water flow direction across the entire network. Once fine scale sewer network model is developed (figure 5.37), this information allows for understanding how waste water flows from an individual building to an outflow node. However, being able to access layout of sewer network *together with* flow information is not always the case. Therefore, an interesting question is, if flow on the sewer network is missing, is it possible to infer such information?

In this section, a sewer flow infer algorithm is developed to infer plausible flow direction on the network. This algorithm, like many other algorithms that have been discussed, is a generic spatial heuristic algorithm, which is built on as least amount of input data as possible. This algorithm requires layout of sewer network, location of outflow nodes, and a DTM layer used to estimate the height of each node. Listing 5.6 shows the pseudo code for the sewer flow infer algorithm.

---

**Algorithm : Generic Sewer Flow Infer**

---

**Input:** Sewer Network  $G$ , knowledge of outflow nodes location, **DTM Layer**

**Output:** Sewer Network  $G$  with flow inferred

```
1: use outflow nodes to initialize current_sinks
2: visited_edges = []
3: visited_nodes = []
4: edges_connecting_sinks = []
5: nodes_connecting_these_edges = []

6: while current_sinks! = [] do

7:   find unvisited edges connecting current_sinks, assign them to
   edges_connecting_sinks
8:   find unvisited nodes connecting these edges, assign them to
   nodes_connecting_these_edges

9:   for  $e$  in edges_connecting_sinks do
10:    if  $e$  is connecting to only one current sink node then
11:      assign direction on  $e$ , which is to that current sink node
12:    else
13:      assign direction on  $e$ , based on heights of two nodes it connects
14:    end if
15:  end for

16:  update visited_edges
17:  update visited_nodes by adding every current sink node to it
18:  current_sinks = []

19:  for  $n$  in nodes_connecting_these_edges do
20:    if  $n$  still connects any edge that is unvisited then
21:      add  $n$  to current_sinks
22:    else
23:      add  $n$  to visited_nodes
24:    end if
25:  end for

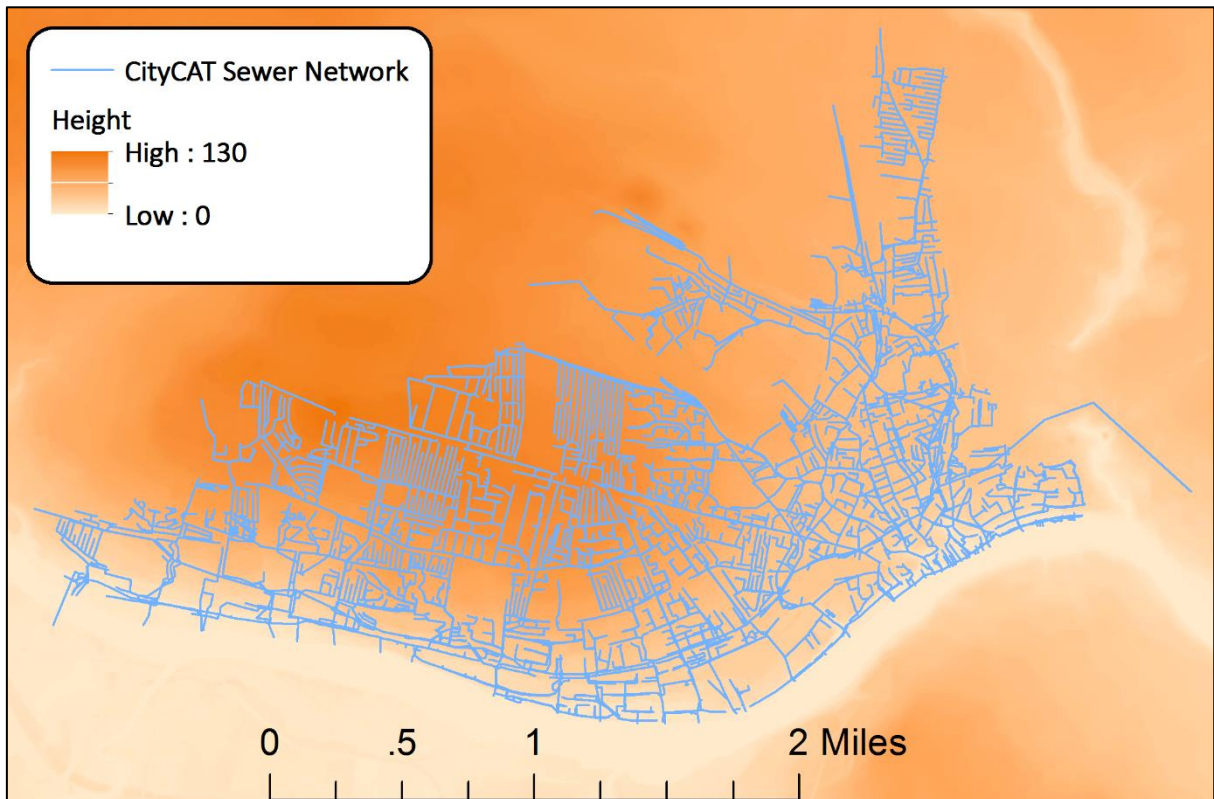
26: end while
```

---

**Listing 5.6.** Pseudo code for the generic sewer flow infer algorithm.

This algorithm is developed using NetworkX library (NetworkX, 2018), and is based on the assumption that waste water should only exit the network at outflow nodes. Moreover, it takes gradient into consideration (since sewer is generally a gravity-based system). However, acquiring height of each node is almost impossible because sewer systems are buried

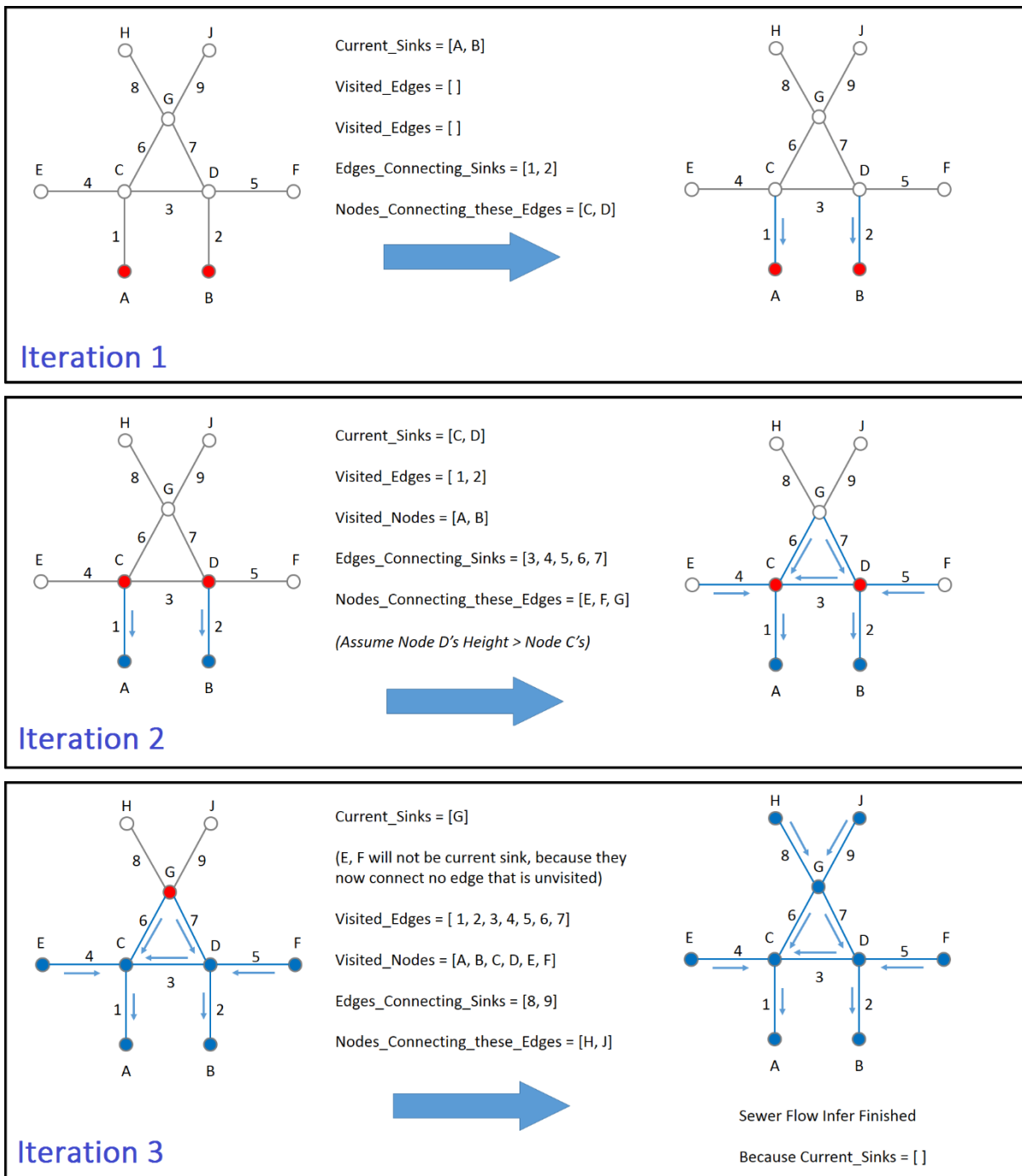
underground. But if assuming each sewer node is buried at relatively same depth underground, then it is possible to use a digital terrain model to estimate height of every node (Obermayer, et al., 2010). The DTM layer (figure 5.38) used in this research is the OS Terrain 5 model (Ordnance Survey, 2018), which has a high spatial resolution (grid size is 5m).



**Figure 5.38.** DTM layer used in the algorithm (Contains OS data © 2018).

The main point of the algorithm is that it tries to infer waste water flow based on spatial connectivity, and this process starts from the outflow nodes. This algorithm is an iterative process and, in each iteration, some number of edges will be assigned directions. The key in this algorithm is a list of nodes called *current\_sinks* to help identify what edges should be assigned what directions in each iteration. The *current\_sinks* can change at each iteration, and algorithm finishes when *current\_sinks* is empty.

To explain the algorithm more clearly, a small simple example is used (figure 5.39), which contains a sewer network of 9 nodes and 9 edges. There are two outflow nodes.



**Figure 5.39.** A simple example to illustrate generic sewer flow infer algorithm.

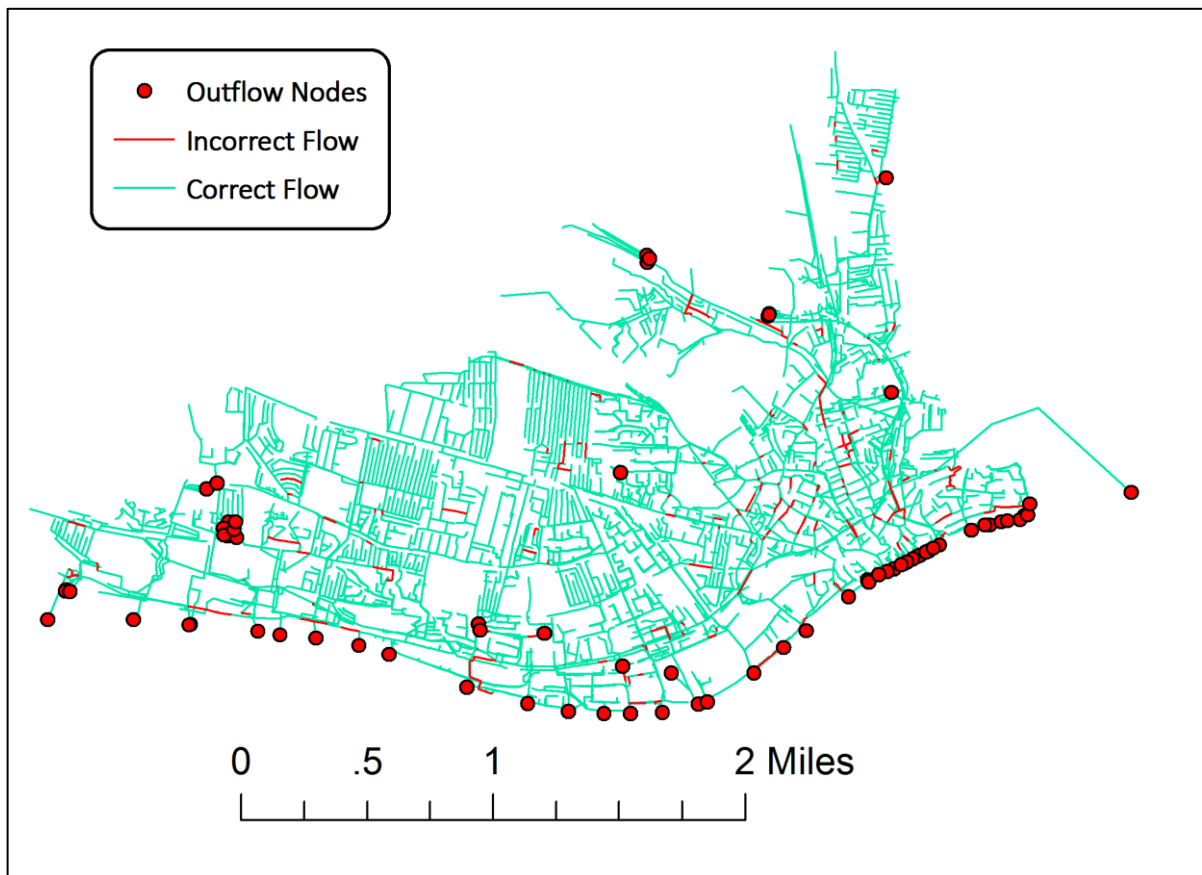
Two important lists *visited\_nodes* and *visited\_edges* are defined and to indicate which nodes and edges have been visited at each iteration. In each iteration, *current\_sinks* can change and is initialized to be the outflow nodes [A, B] when algorithm begins. The list *visited\_nodes* and *visited\_edges* are initialized to be empty.

In iteration 1, *current\_sinks* are node A, and B. The all the unvisited edges connecting current



sink nodes are edge 1 and 2. These edges will be assigned direction (to the corresponding current sink node). Then edge 1 and 2 are visited. Node 1 and 2 are visited. The list *current\_sinks* is emptied. Then node C and D (on the other side of edge 1 and 2), will be put into *current\_sinks* if these nodes still connect any unvisited edges (true in this case), otherwise they will be marked visited as well.

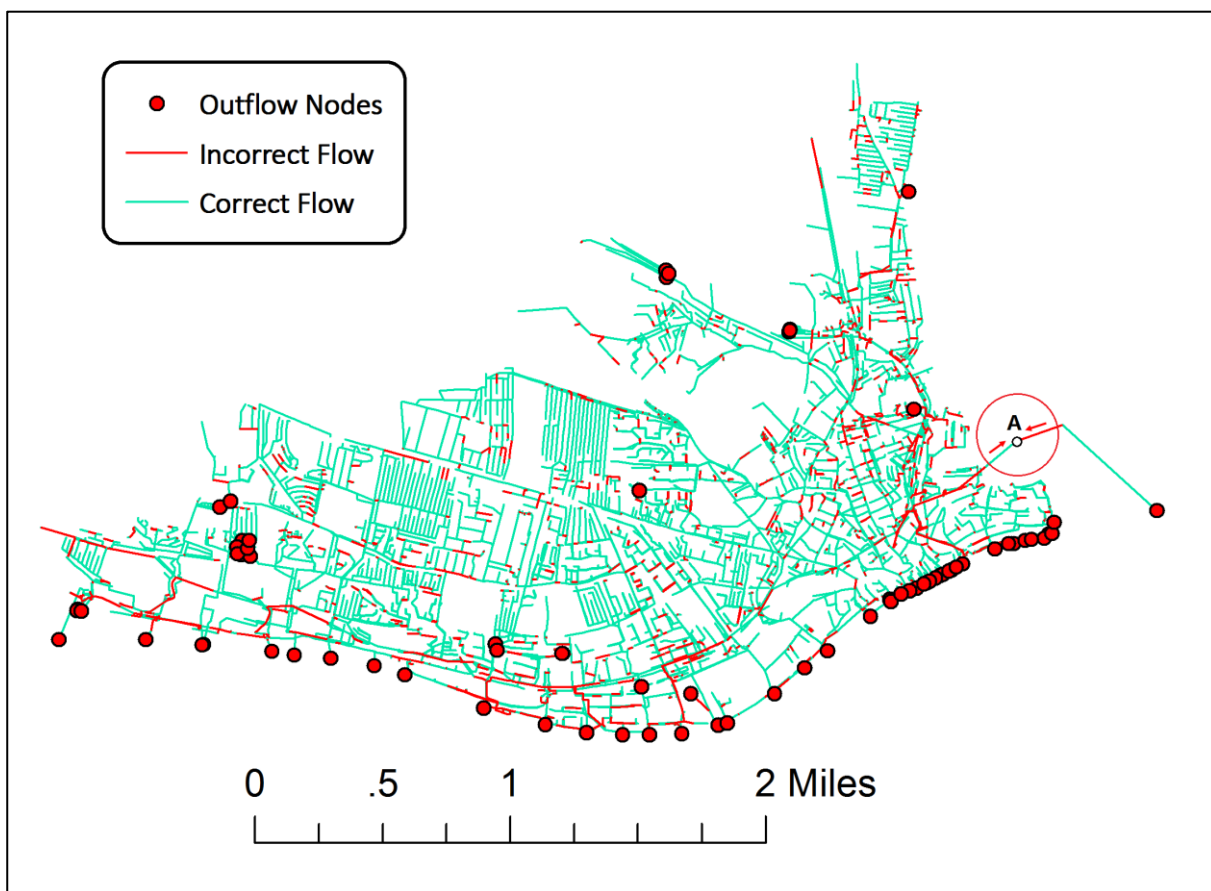
The iteration 2 starts with C and D being the current sink nodes. The edges 3, 4, 5, 6, 7 will be then assigned direction. Note edge 3 is special here, because two nodes connecting edge 3 are both current sink nodes (C and D). Therefore, height information (DTM layer) is used here, and if assuming node D is higher than node C, then it is considered more plausible to say flow direction is from D to C on edge 3. Finally, iteration 3 starts with G being the only current sink node. After assigning direction on edge 8 and 9, the algorithm finished, as there is no more node that can be put into *current\_sinks*.



**Figure 5.40.** Validation of flow direction, *inferred by the algorithm.*

Using this algorithm, flow is inferred on the CityCAT network, and is validated against the actual flow direction, which is shown in figure 5.40. Of all the 8306 edges in the CityCAT network, flow on 7959 edges are inferred correctly, which means an accuracy of **96%**. This accuracy is considered high, as this result is generated without resolving hydrologic models.

Now one interesting question is that, is it possible to infer flow only using height information (DTM layer)? That means every edge is assigned a flow direction, from a higher node to a lower node it connects. A test has been done for that, and result (validation) is shown in figure 5.41.



**Figure 5.41.** Validation of flow direction, *inferred by only using the DTM layer.*

If only using DTM layer, then flow directions on 1556 edges are inferred incorrectly, that means the accuracy in this situation is only **81.2%**, much lower than the accuracy achieved via the algorithm. True flow directions on about 20% of the edges are actually against slope calculated from the DTM layer. The major cause is that the DTM might not represent the *exact height* of each node. When using the DTM layer, it is assumed that each node is buried

for *same depth* underground. For any edge (pipe), if depths of two nodes it connects are different, then it can be no longer accurate to infer node height via the DTM layer.

There is one bigger problem when inferring flow only using the DTM layer. That is generating *false* sink nodes in the network when it should not have. In figure 5.41, within red circle, if using the inferred flow, the node A is a sink node (mathematically a node whose out degree is 0, in a directed graph). This is invalid, because the waste water is only allowed to exit the sewer network at *one of the outflow nodes*. That means when inferring the flow, sink nodes except for the outflow nodes, should never exist.

That is why the algorithm is developed this way (infer flow from outflows nodes first). Flow direction is inferred using spatial connectivity first, and when it is no longer possible, height information is then used. Since the algorithm is easy to be implemented and requires only sewer network layout, outflow nodes and DTM layer, it is considered to be a generic solution when there is not enough data to generate a more accurate flow model (via hydrologic approaches).

## 5.5 Utility Network Dependency Integration

In Chapter 3, it has been identified that dependencies exist among different types of utility networks (Ji, 2019). In the formal ontology (Chapter 3), dependency is represented via a mapping from a utility asset in gas, water supply or sewer network to a substation in the electricity distribution networks (Ji, 2019). Let  $\mathbf{S}$  be the set of substations, then utility dependency can be represented in table 5.4.

Utility Network	Utility Asset	Dependency
Gas	Regulation Sites ( $R_s$ )	$f: R_s \rightarrow S$
Water Supply	Water Pumps ( $W_p$ )	$f: W_p \rightarrow S$
	Water Treatments ( $W_t$ )	$f: W_t \rightarrow S$
Sewer	Sewer Pumps ( $S_p$ )	$f: S_p \rightarrow S$
	Sewer Treatments ( $S_t$ )	$f: S_t \rightarrow S$

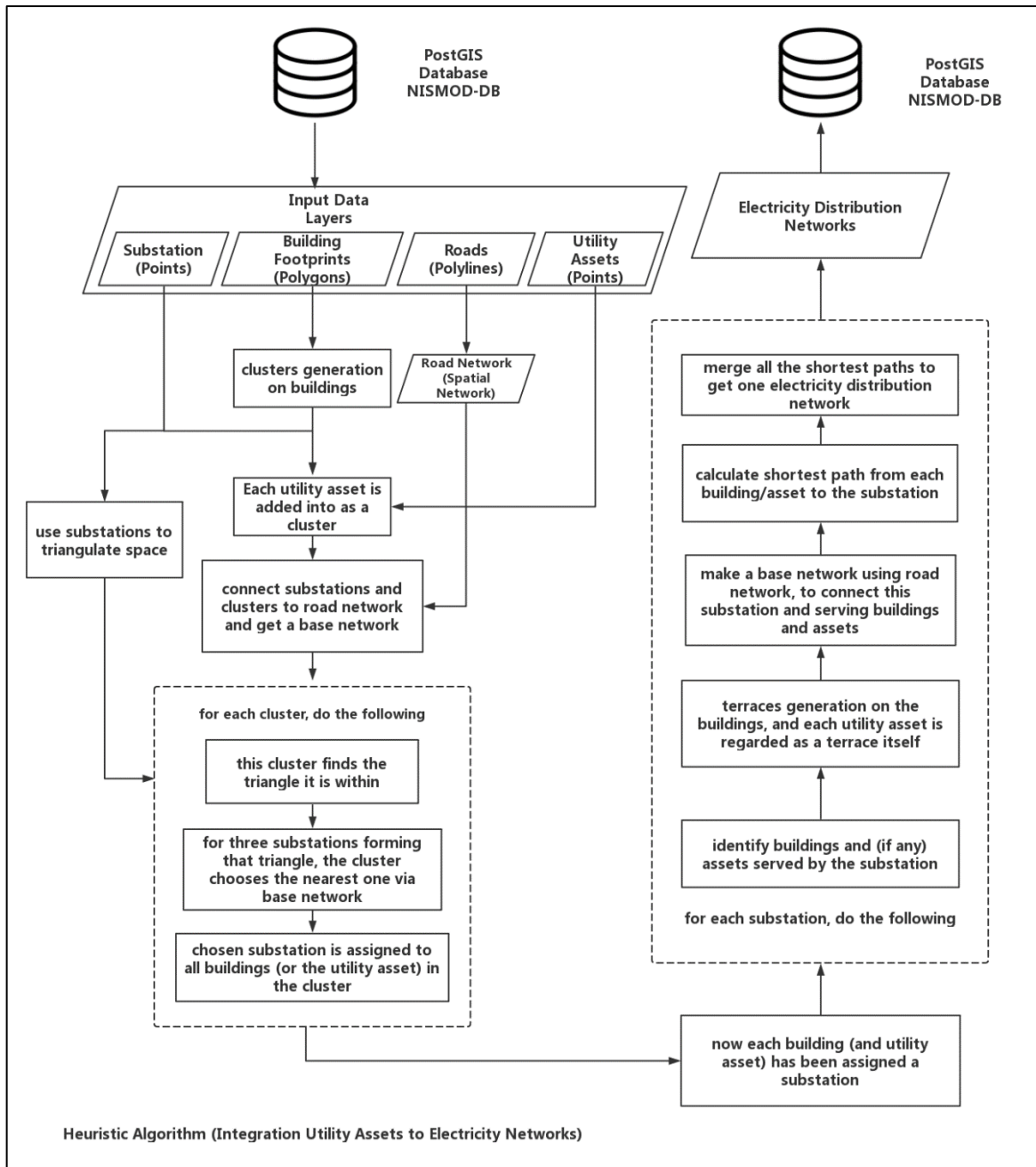
**Table 5.4.** Utility network dependencies.

The dependency is a one-to-one mapping (for example, a gas regulation site depends on electricity power from a substation). The knowledge of dependencies allows for representing utility networks as *Networks of Networks* (D'Agostino, et al., 2014) and it is essential in understanding cascading failures between different utility networks (Johnson, et al., 2007).

In Chapter 4, spatial heuristic algorithm is used to generate electricity distribution networks in Newcastle upon Tyne, which connect substations (of 11 kv) to the buildings. According to the local electricity supplier NPG, utility assets are also served by substations of this level (Northern Power Grid, 2017).

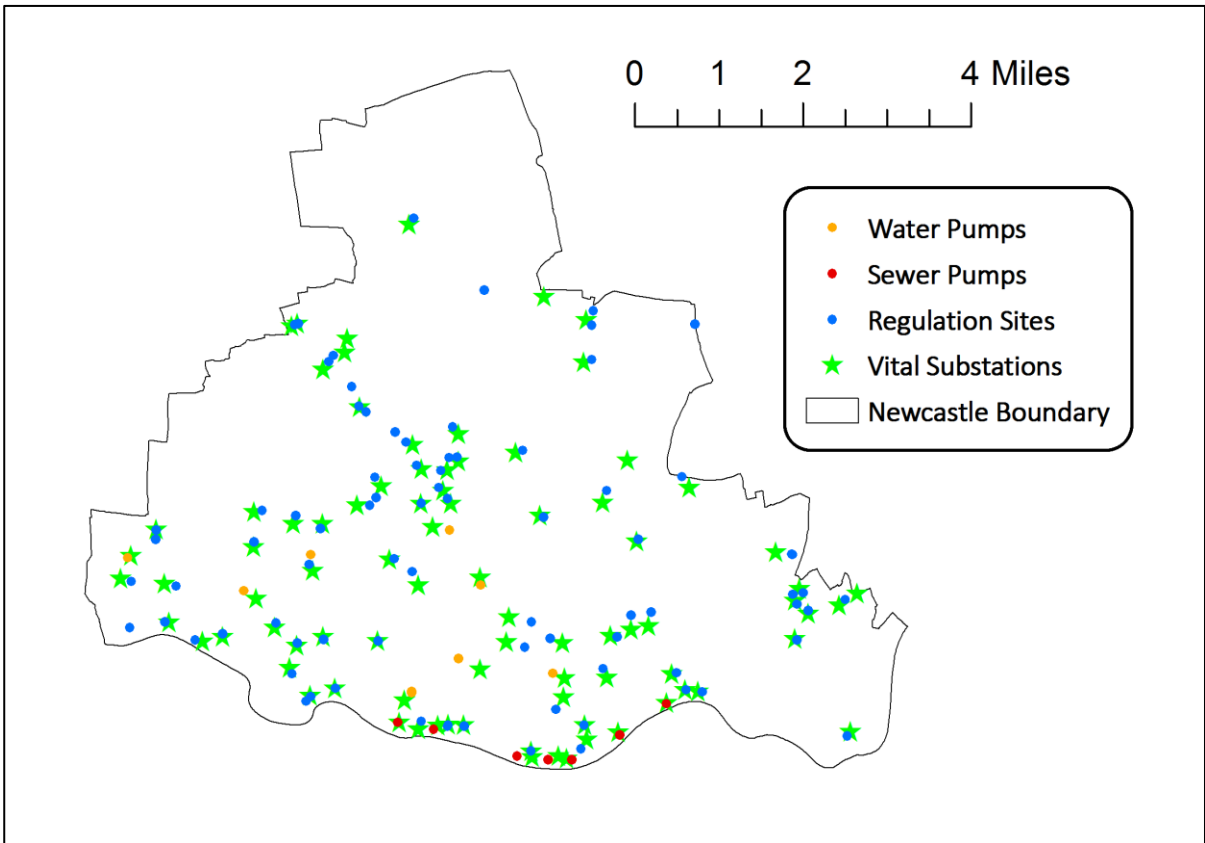
Therefore, an algorithm is developed in this section, to connect utility assets to the electricity distribution network, following a similar approach discussed in Chapter 4. Figure 5.42 shows the keys stages involved in integrating utility assets to the electricity networks. The rationale behind this approach is that, cables used to connect a utility asset and its dependent substation should be as short as possible.

The algorithm starts from reading initial input (utility asset points, building footprints, roads, substation point) from PostGIS database. Then clusters are generated using building footprints and asset points. Later a base network will be generated by connecting every cluster into the road network. For each cluster, a substation (nearest one via path distance on the base network) will be assigned to each cluster. Then spatial layout of each electricity distribution network can be generated to connect the substation to the buildings and utility assets (if there is any). Finally, synthetic electricity distribution networks will be written back to the PostGIS database.

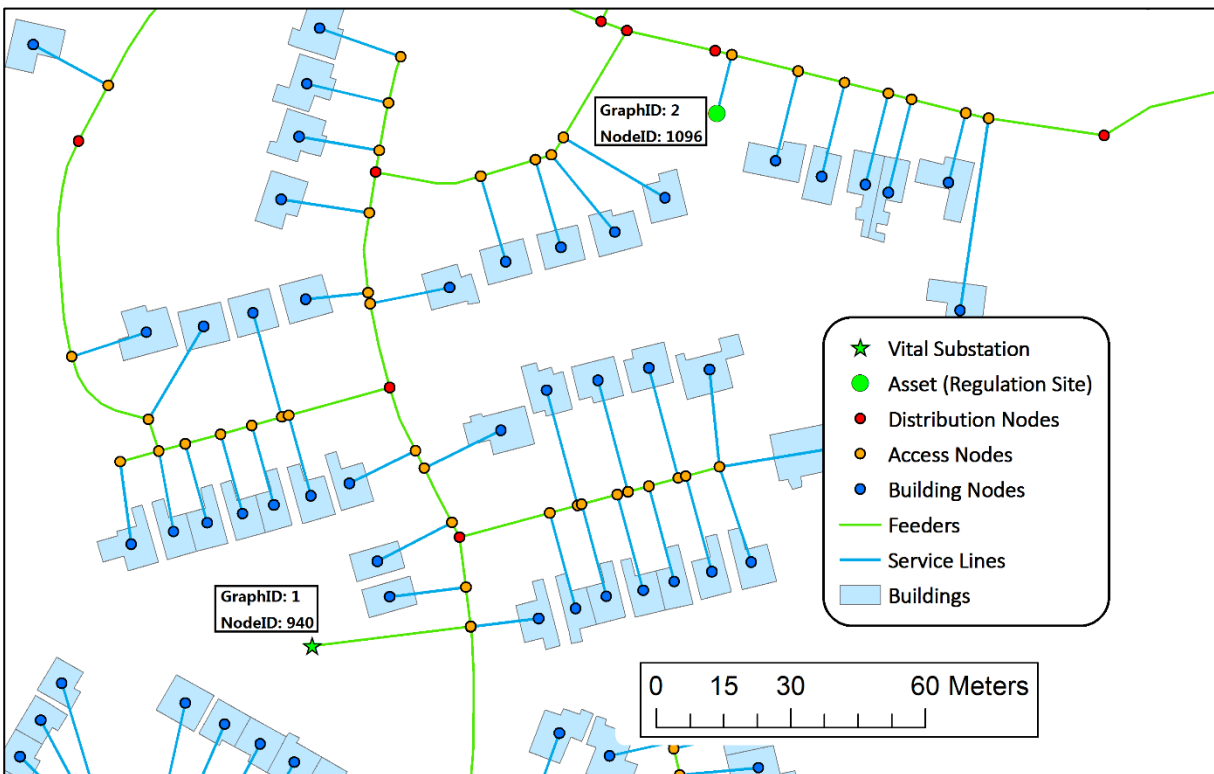


**Figure 5.42.** Algorithm flow of integrating utility assets to electricity distribution networks.

The algorithm was applied to integrate utility assets to electricity distribution networks in Newcastle upon Tyne. The utility assets are 105 gas regulation sites, 9 water pumping stations and 7 sewer pumping stations based on available data. There are 636 substations in the entire city, and according to the algorithm, 551 of them serve electricity *only* to the buildings and 85 of them serve electricity to both buildings and utility asset(s). These 85 substations are termed *vital substations*, and they are shown in figure 5.43. Figure 5.44 shows how an asset (gas regulation site in this case) is exactly integrated to an electricity distribution network.

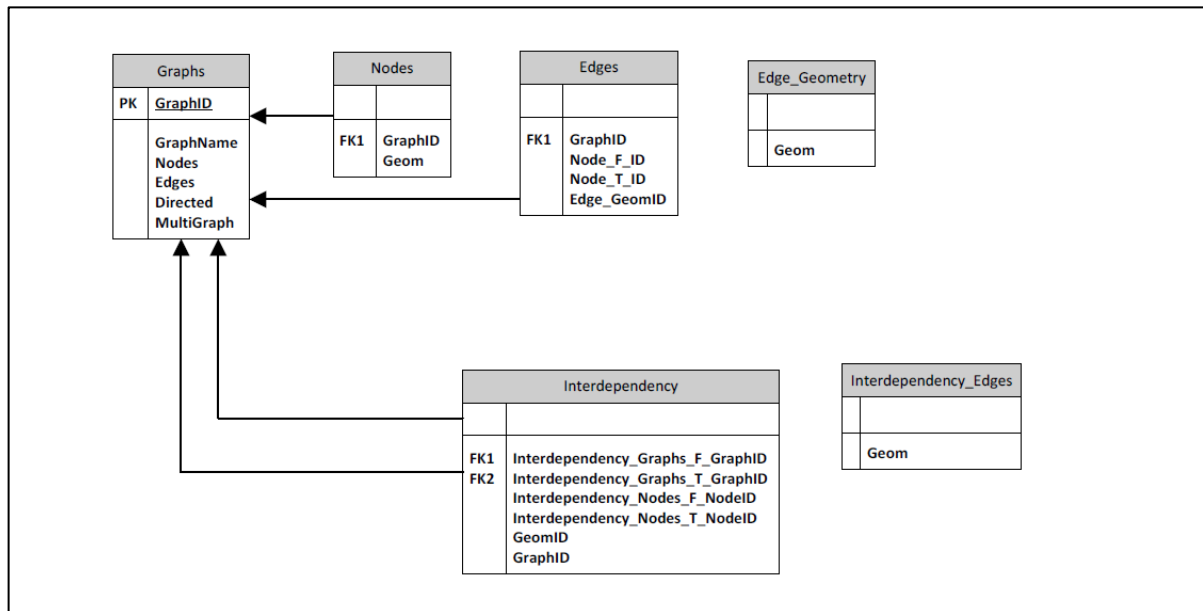


**Figure 5.43.** Location of utility assets and vital substations in Newcastle upon Tyne (Contains OS data © 2018).



**Figure 5.44.** Utility asset (regulate site in this case) integrated into electricity distribution networks (Contains OS data © 2018).

Representation of dependency (from utility asset to an electricity substation) makes it possible to store utility networks as *Networks of Networks* in a database system. For example, a common approach to store interdependent geospatial network instances is to use ITRC database schema (figure 5.45), which is developed for modelling national scale geospatial infrastructure networks in the United Kingdom (Barr, et al., 2013).



**Figure 5.45.** PostGIS ITRC database schema.

With the ITRC schema, for each type of infrastructure network, a table is used to store network nodes. For example, a table **electricity\_net\_Nodes** and **gas\_net\_Nodes** are the tables to store nodes for electricity and gas networks (figure 5.46). Within each table **NodeID** is the primary key. To distinguish nodes from different types of infrastructure networks, a specific **GraphID** is given for one network (in this case, 1 for electricity and 2 for gas). To store network dependency, ITRC schema uses an **Interdependency** table, which stores the **GraphID** and **NodeID** for the node where the dependency is from and for the node where the dependency is to. For example, in figure 5.45, (**GraphID**, **NodeID**) is (2, 1096) for the gas regulation site in the gas network and (1, 940) for its dependent substation in the electricity network. Then dependency can will be stored in the **Interdependency** table in figure 5.46. This is exactly how to store infrastructure networks as *NetworksOfNetworks* in ITRC schema and makes it very easy to write simple SQL queries to select dependent node(s) for any node.

GraphID integer	geom geometry	NodeID bigint	nodename character varying	nodetype character varying	nodesubsys integer	toid character varying
1091	2 0101000020346C000014AE47E1F9AB1941CDCCCC127E2141	1091	new 0000118143	building	30	0001000038591402
1092	2 0101000020346C00009A999999C9B319410AD7A370BB7F2141	1092	new 0000118565	building	30	0001000038622132
1093	2 0101000020346C000048E17A14B3B319419A999999B87F2141	1093	new 0000086134	access	30	0
1094	2 0101000020346C00003D0AD7A39A9F19413D0AD7239C7F2141	1094	new 0000004514	access	30	0
1095	2 0101000020346C0000EC51B81ED7AF1941A4703D0AA87F2141	1095	new 0000118419	building	30	0001000038591752
1096	2 0101000020346C000085E51B855AE194114AE4761217F2141	1096	661095002950001	source	30	0
1097	2 0101000020346C000014AE47E1A5B51941F6285CF047802141	1097	new 0000118515	building	30	0001000038622076
1098	2 0101000020346C00003D0AD7A3C4B319410000008035802141	1098	new 0000118542	building	30	0001000038622107
1099	2 0101000020346C00008FC2F5280FB41941F6285C8F2E802141	1099	new 0000118544	building	30	0001000038622109

gas regulation site

### gas\_net\_Nodes

GraphID integer	geom geometry	NodeID bigint	type character varying	toid character varying	netid integer
936	1 0101000020346C00007B14AE47D7AC1941D7A370BD9A7E2141	936	buildingAccess	0	61
937	1 0101000020346C0000713D0AD721AB1941C3F5285C217D2141	937	building	0001000038591381	61
938	1 0101000020346C000014AE47E1B3B5194100000002E802141	938	building	0001000038622077	61
939	1 0101000020346C00005C8FC2F512AC194152B81E058D7C2141	939	building	0001000038591372	61
940	1 0101000020346C000066666666E6CAC1941CDCCCC337E2141	940	substation	0	61
941	1 0101000020346C0000713D0AD777AC194166666666287D2141	941	buildingAccess	0	61
942	1 0101000020346C0000713D0AD757AC1941F6285C8F0A7D2141	942	building	0001000038591385	61
943	1 0101000020346C00007B14AE4758AC194152B81E05917D2141	943	buildingAccess	0	61

electricity substation

### electricity\_net\_Nodes

Interdependency_Graphs_F_GraphID integer	Interdependency_Graphs_T_GraphID integer	Interdependency_Nodes_F_NodeID bigint	Interdependency_Nodes_T_NodeID bigint
1	2	1	1096
2	2	1	136
3	2	1	1175
4	2	1	3606

dependency

### Interdependency

Figure 5.46. An example of using ITRC schema to store network dependency.

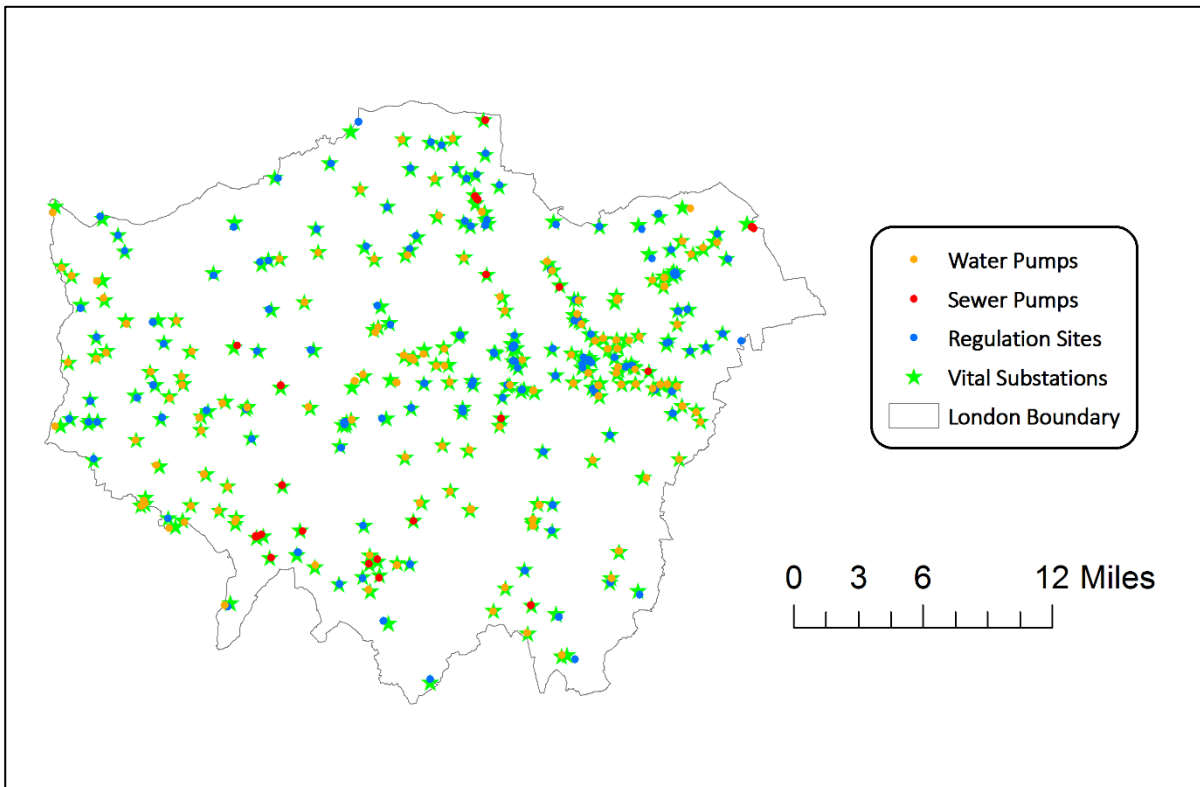


Figure 5.47. Location of utility assets and vital substations in London.



The algorithm shown in figure 5.42 requires utility asset location to be the only necessary information from utility networks. That means this algorithm will work without knowing utility network layout (location of pipes, or cables). For example, for the city of London, from MasterMap PoI layer (Ordnance Survey, 2018), 174 gas regulation sites, 135 water supply pumping stations, and 27 sewer pumping stations. This information is enough to infer the dependency from these utility assets to 335 electricity substations (figure 5.47).

## **5.6 Conclusion**

In the last chapter, a generic spatial heuristic algorithm was presented generate fine scale layout of infrastructure networks based on location of infrastructure assets, roads and buildings. Based on this algorithm, in this chapter, modified approaches were discussed to generate layout of fine scale utility networks (gas, water supply, and sewer) for the city of Newcastle upon Tyne. For the utility networks, part of network layout (of the main pipes) is known, but data incompleteness can exist in each of them.

For gas network, layout of main pipes can be missing in new developing areas. A gas network infer algorithm was developed to infer the layout of main pipes in these areas, and has achieved high accuracy via validation. For water supply network, flow direction is not included in original data. A water flow infer algorithm was developed to first identify WDAs (water distribution areas) in the water supply network and then infer water flow on each pipe. However, validation was not able to be done, because actual water flow direction is to date still not available. Therefore, trying to access actual data and validating flow accuracy will be one of the future objectives. For the sewer network, the data covers only central part of the city. Currently, layout of sewer network in the entire city cannot be inferred from my algorithm, because the assets location (manholes, outflow nodes) is unavailable in the entire city. This is one of the major limitations of my algorithm (can infer layout of pipes or cables, but not location of assets). Sewer flow has been encoded into the sewer network data. However, it is considered necessary to have an approach to infer sewer flow as if it does not

exist. Therefore, a generic sewer flow infer algorithm was developed based on network spatial connectivity and DTM model, and has achieved high accuracy.

Finally, work was represented to infer dependencies from utility networks to the electricity distribution networks. This was achieved by applying a slightly modified version of algorithm discussed in Chapter 4. Utility assets are inferred to be dependent on the nearest substation via the road network. A major achievement of this algorithm is that, it can infer utility dependencies without knowing utility network layout (only asset location is necessary). However, my approach is still a pure spatial algorithm, and that means capacity (of the substation) is not considered here. As is discussed at the end of Chapter 4, the number of buildings (and asset if any) is limited for each substation, and this needs to be taken into account in the future work.

## Chapter 6 – Road Network Generation Algorithm

### 6.1 Introduction

In Chapter 4, the case study section (section 4.4) showed the automatic generation of plausible synthetic electricity distribution networks in the entire city of Newcastle upon Tyne, based on a generic spatial heuristic algorithm. The algorithm relies on a local road network, which serves as the *backbone* to help generate both the topology and geometry of the distribution network. The case study showed the capability of infrastructure network planning in the urban area, as long as the layouts of buildings, road network and infrastructure assets are known.

However, road network layout is not always available, especially in the early urban planning stages (McGill University, 2008). For new developing sites, the urban planners will first decide use of land (decide layout of residential buildings, water bodies, factories, park, etc.), based on the considerations including environment conservation, prevention of land use conflict, minimizing residents transport cost, and reduction in exposure to pollutants (Kaiser, et al., 1995). After that, infrastructure networks such road, communication and distribution networks can be planned according to the given land use layout (Moss, et al., 2016).

Therefore, for the new developing sites, generating layout of infrastructure distribution networks is more difficult, as layout of road network is not always present. That leads to an interesting question: is it possible to automatically generate layout of road network in the new developing urban areas, if layout of land use (at least buildings) is given?

This chapter aims to develop a spatial heuristic algorithm, which allows automatic generation of road network layout. It can be applied together with algorithm developed in Chapter 4, to show much stronger capability in infrastructure network planning.

## 6.2 Automatic Network Generation

Automatic generation of road network is a typical network design problem (NDP) (Magnanti, et al., 1984). It is a challenging problem, as it requires to decide the optimal configuration of road network elements with regards to a set of criteria (Yang et al., 1998). The road network elements generally refer to the network topology, geometry, capacity, and traffic signal configuration, etc (Cantarella, et al., 2006). Generation of road network, according to different requirements, if done manually by road design specialists, can be a very time-consuming task (Campos, et al., 2015). Several related studies have been done, in automatic planning and designing of road network layout, which are explained in table 6.1.

Author	Explanation on the approach
Parish, et al., 2001	The author developed a procedural modelling platform for cities, to generate the layout of buildings and road networks in the urban areas. The platform requires geographical maps (DTM, land/water/vegetation maps), and social maps (population density, zone maps, etc.). The approach is based on L-system and will generate layout road network first, and then allocate space for buildings. The approach can be easily implemented computationally, but requires information such as population density, and does not consider buildings as the input (rather it is algorithm output).
Cantarella, et al., 2006	A heuristic multi-criteria algorithm was developed to automatically design urban transport network. Both the network layout and capacity (such as traffic lights configuration) can be optimised. However, this is algorithm that can be computationally expensive (solving NDP problem under multi-criteria) and it is only at the theoretical stage, without any application or validation using real city data.
Teoh, 2007	A platform was developed for generating realistic cities in the game industry. The user needs to give terrain information and some preference (such as desired city size) as the input, then urban centres (such as commercial and industrial centres, residential, and airports) can be generated. Roads can be then generated to connect these centres. This requires even less input than Parish's approach, however, it still does not consider existing layout of buildings.
Nie, et al., 2010	An algorithm of generating rectilinear Steiner tree was developed to generate rural network layout. Initial input is only the nodes known in the rural network (layout of the counties). Then a rectilinear Steiner tree will be built to connect these nodes and it will be further optimized as final output. The algorithm is computationally cheap, but it focuses on the road

	network at rural level, not urban level. Moreover, it requires nodes (road junction) to be known already, which is not available in our problem.
Rui, 2013	The author developed a platform for dynamic modelling urban growth and city road network evolution, where population is considered as the major driving force. In the author's model, road network layout will be extended to accommodate increasing travel demand due to increasing population. The author's model is more like a dynamic model, rather than a generative algorithm. Therefore, this approach must know existing layout of road network, which is its major drawback.
Zhang et al., 2017	An approach was developed to acquire real-time mapping information and automatically produce layout of road networks. The approach relies on volunteered geographic information, and in particular, the GPS trajectories from vehicles. The main idea is that, where there is a road, it is always reachable for any vehicle. Therefore, the approach collects large amount of GPS trajectories from taxis and merges them into a directed graph, as digital map of road network. This approach is a generative algorithm, but needs to acquire a large amount of additional data (GPS trajectory) to be efficient, and still there is not consideration on the layout of buildings.

**Table 6.1.** Related approaches for automatic road network generation.

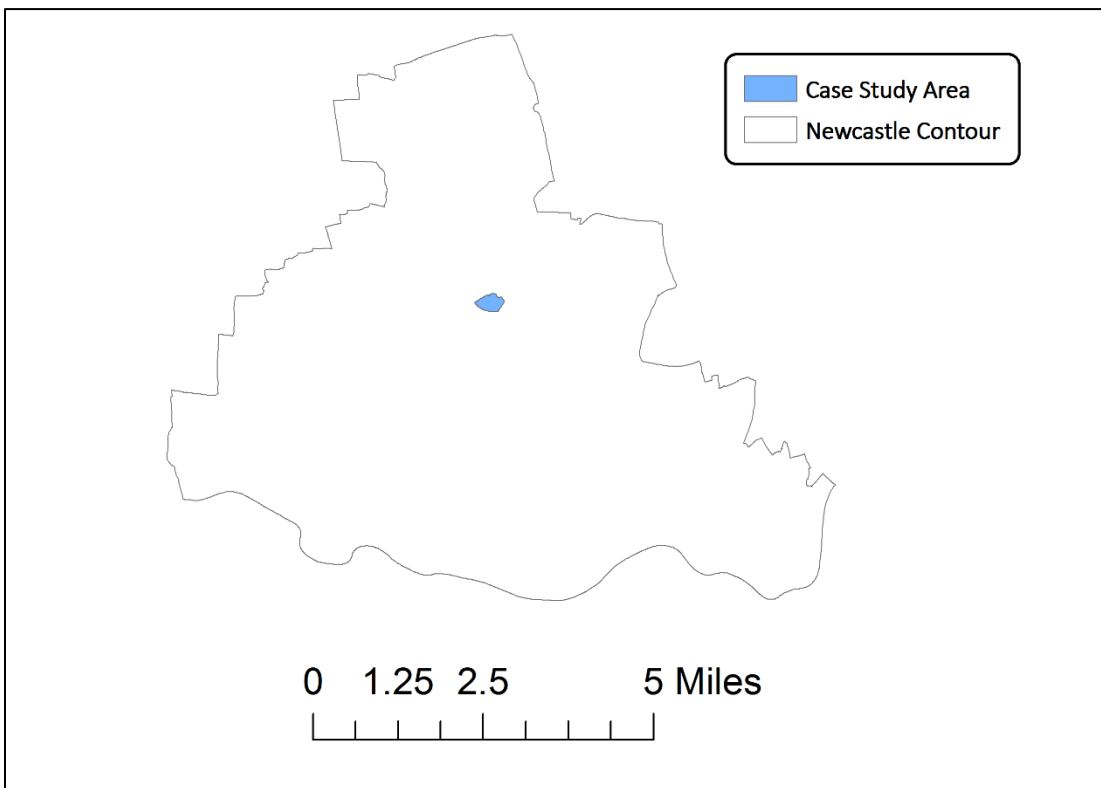
From table 6.1, it is found that the existing approaches can generate layout of road networks based on different constraints and requirements, but they do not fit this particular problem. In fact, none of them considers building layout as the input data (actually Rui's approach does, but it is an evolution model of existing road network, not a generative algorithm). In this chapter, a new and automatic road network generation algorithm will be discussed to tackle our specific problem. The input data sets are introduced in section 6.2, and section 6.3 describes the algorithm and explains the rationale behind it.

### 6.3 Data Sets

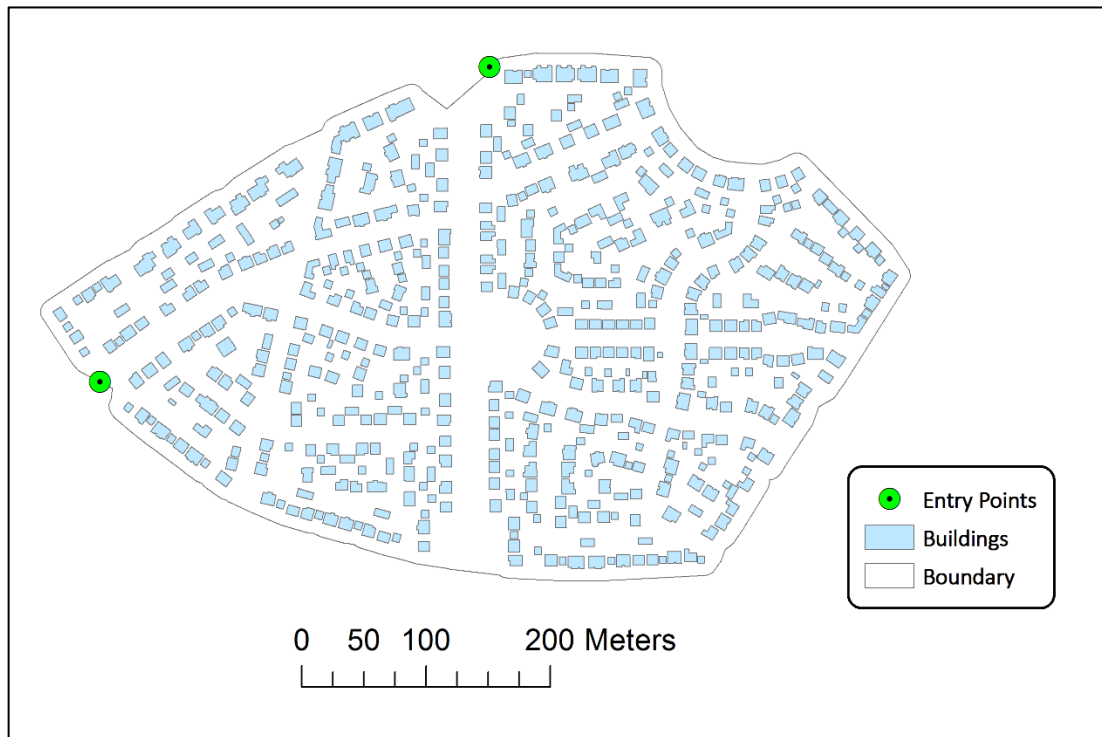
By consulting Arup Group ([www.arup.com](http://www.arup.com)), which is a civil engineering and design company in Newcastle upon Tyne, an appropriate case study area (to develop this algorithm) was chosen. It is relatively a new and small development area at the north of the city (figure 6.1, and figure 6.2), covering an area of 197,326 m<sup>2</sup>.



**Figure 6.1.** Case study area to develop road network generation algorithm (from Google Maps 2018).



**Figure 6.2.** Location of case study area in Newcastle upon Tyne (Contains OS data © 2018).

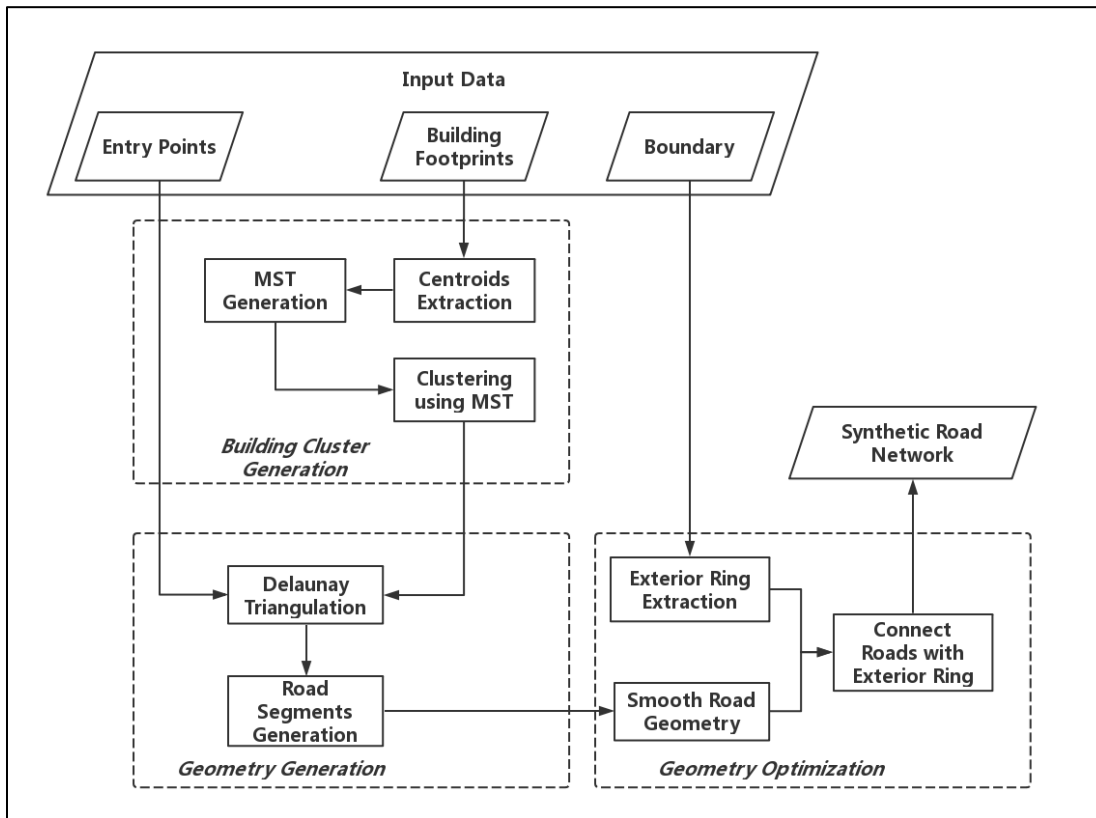


**Figure 6.3.** Input data sets for the case study area (Contains OS data © 2018).

For the case study area, the input data sets (figure 6.3) contain layout of buildings, a boundary, and the entry points. The boundary is a manually digitized polygon which covers entire case study area, and it is assumed the exterior ring (the polyline) of the boundary should represent the external road network surrounding the area. The entry points refer to the points where the road network inside study area should be connected with the road network outside (on the boundary). Totally there are 536 buildings and 2 entry points in this area.

#### **6.4 Road Network Generation Algorithm**

The basic flow of road network generation algorithm is shown in figure 6.4. The algorithm reads entry points (points), building footprints (polygons) and the boundary (polygon). The algorithm consists of three major steps. Step 1 generates building clusters based on a minimum spanning tree (MST) partitioning algorithm. Step 2 generates geometry of road segments based on Delaunay Triangles. Step 3 optimizes the geometry of the roads. Details of these three major steps are discussed in sub section 6.4.1, 6.4.2, and 6.4.3.



**Figure 6.4.** Flow of road network generation algorithm.

#### 6.4.1 Building cluster generation using MST

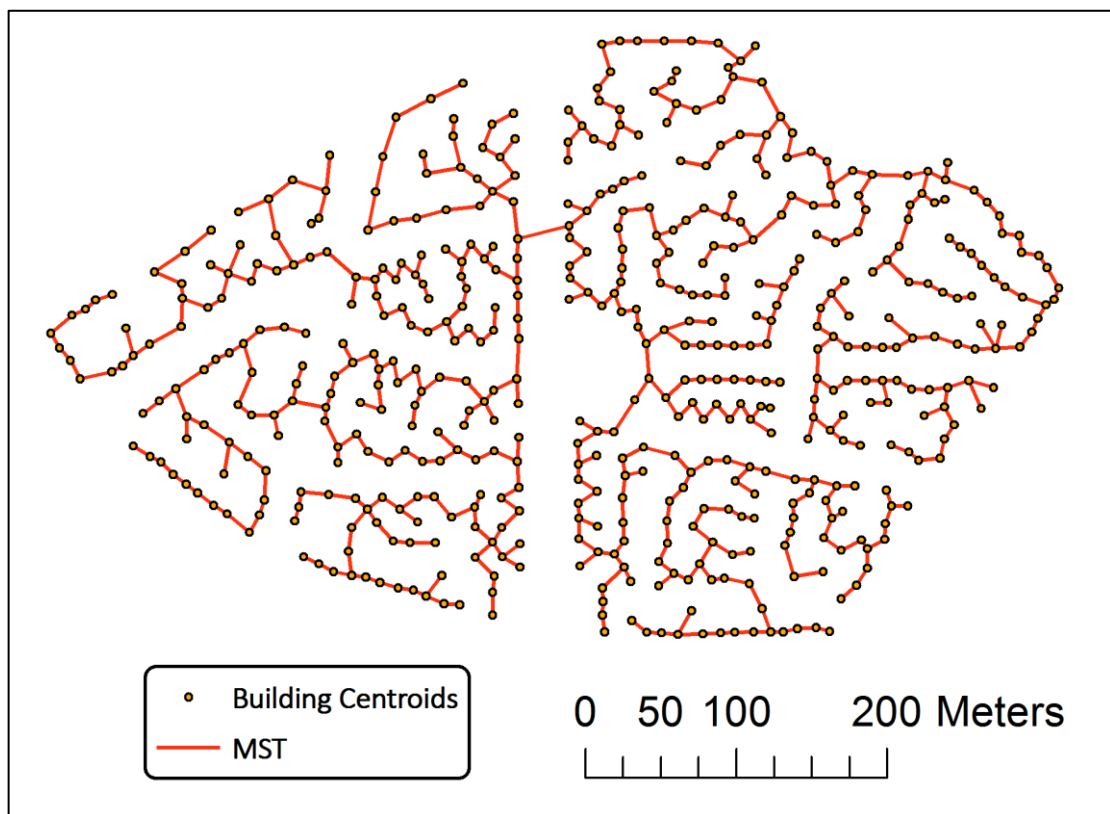
A close observation of the case study data, as well as the data in the entire city of Newcastle upon Tyne, indicates presence of roads is related to the layout of buildings. Geospatially, close buildings can form clusters and for any cluster of buildings, it is surrounded by road segments. Therefore, the key is to find building clusters from input data.

Geospatially, each building can be represented by its centroid, and therefore, the problem can be generalized to a clustering problem on points in the 2D space. The most common clustering algorithm is the k-means algorithm (Krishna, et al., 1999). However, k-means algorithm requires to set up a hyper-parameter  $k$  (the number of clusters to be generated). That is a big problem in our situation, because it is impossible to know the correct number of clusters beforehand.

Therefore, a clustering algorithm that does not require prior knowledge of number of clusters



will be more appropriate to solve this specific problem. The chosen algorithm is the clustering algorithm based on minimum spanning tree (Zhou, et al., 2009). A minimum spanning tree (will be termed MST later) is a spanning tree (a graph connecting all the nodes) whose sum of edge weights is as small as possible. In this clustering algorithm, an MST is first generated to connect all the points (edge weight is the geometry length of the edge). Then the MST will be partitioned to generate clusters.



**Figure 6.5.** MST generation (Contains OS data © 2018).

Generation of MST is achieved via the NetworkX library (NetworkX, 2018) and the result is shown in figure 6.5. Then this MST will be partitioned, and that means *some* edges will be removed from the MST. If one edge is removed from MST, the MST becomes two connected components (each is a cluster). If one more edge is removed, the MST becomes totally three connected components. This is the main rationale of generating clusters using MST. The most important part, is to decide *which edges should be removed* from MST. This is explained in listing 6.1, which is the MST partitioning operation, suggested by Zhou et al. (2009).

---

**Algorithm :** Partition MST to generate clusters

---

**Input:** The minimum spanning tree  $MST$ **Output:** The trimmed minimum spanning tree  $MST$ 

---

```
1: while true do
2:    $Edge = MST.find\_best\_edge()$ 
3:    $MST.remove\_edge(Edge)$ 
4:   if  $|\sigma(MST^n) - \sigma(MST^{n-1})| < \epsilon(|\sigma(MST^0) - \sigma(MST^n)| + 1)$  then
5:     break
6:   end if
7: end while
8: return MST
```

---

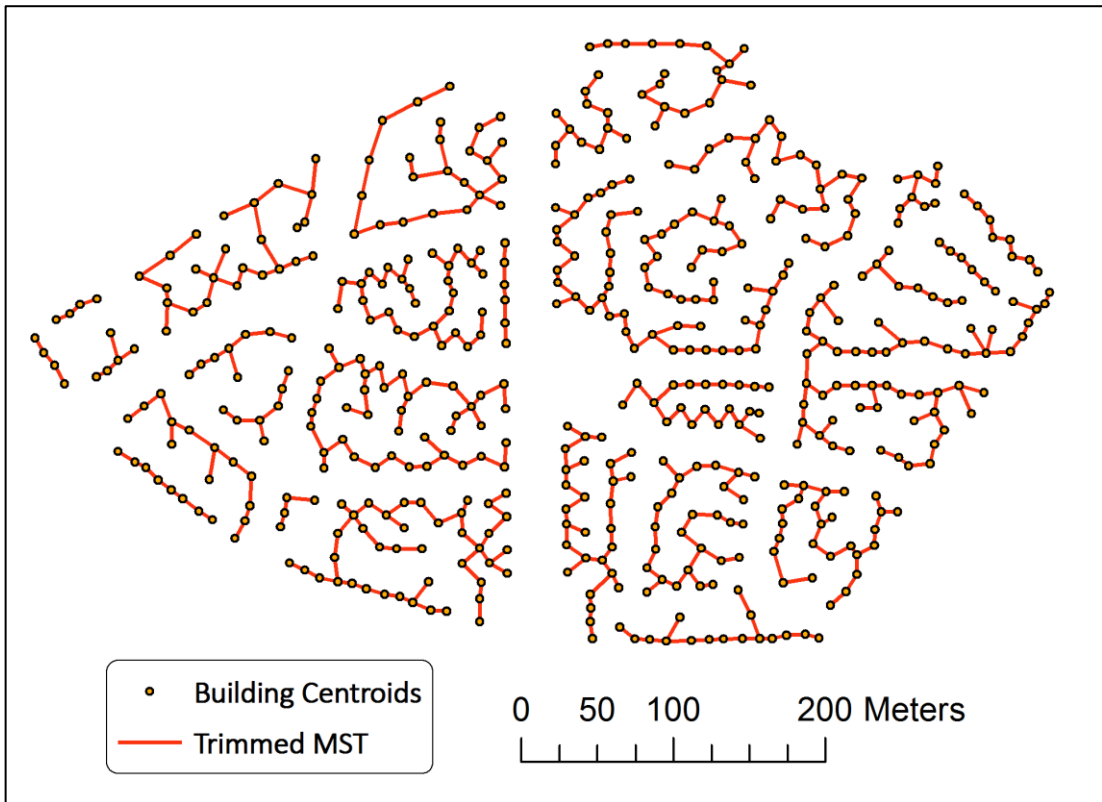
**Listing 6.1.** MST partitioning operation (Zhou et al., 2009).

The MST partitioning operation is an iterative process, and in each iteration, one edge from MST is removed. This iteration will stop when a certain condition is satisfied.

In listing 6.1,  $\sigma$  is the global standard deviation of edge lengths on the given network.  $MST^0$  is the initial MST, while  $MST^n$  is the MST after  $n$  iterations. The *find\_best\_edge()* is a function to check MST (in the current iteration), in order to find an edge that causes largest change in global standard deviation ( $\Delta\sigma$ ), if this edge is removed.

In this iterative process, the more edges that are removed from the initial MST, the less difference there will be between global standard deviations, of the MST in the current iteration and MST in the previous iteration ( $|\sigma(MST^n) - \sigma(MST^{n-1})|$ ). Therefore, if this difference becomes too small, it is considered to be time to stop the iteration. The  $\epsilon$  is the parameter to control when to jump out of the iteration. After iteration finishes, the MST will be returned, which has been modified and partitioned into several clusters.

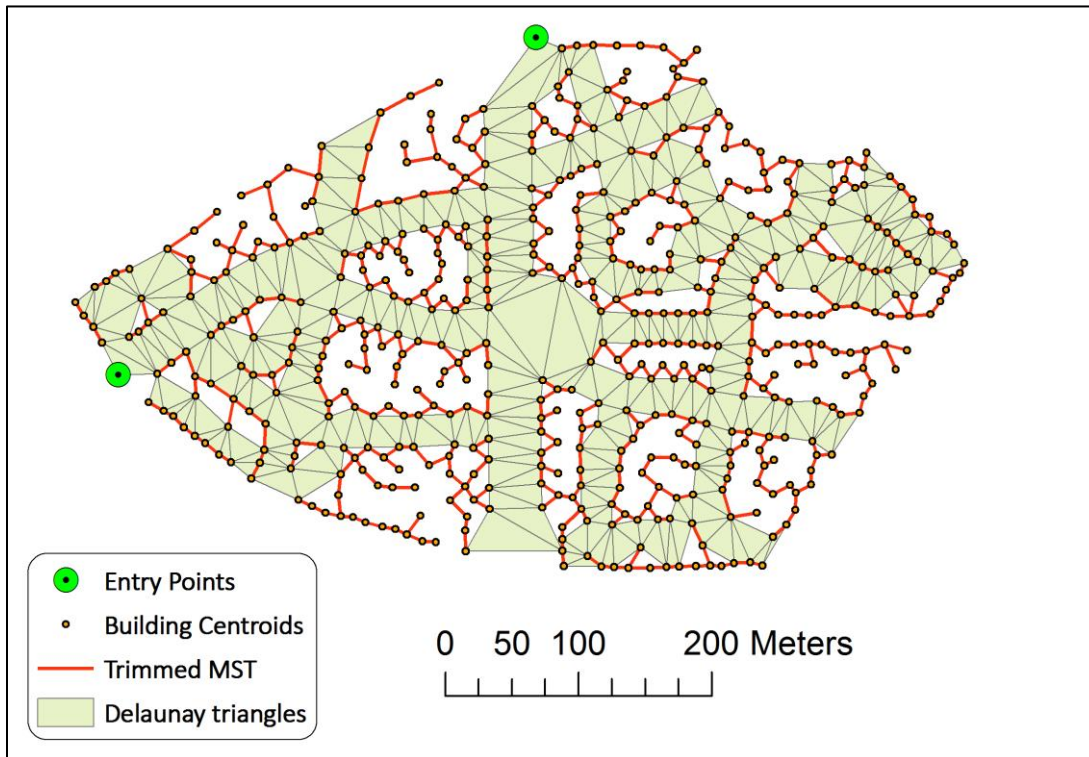
Still, there is one parameter  $\epsilon$  that needs to be tuned. Number of clusters generated is sensitive to the  $\epsilon$  value. The larger  $\epsilon$  value results in later stop of the iteration, and that means more edges will be removed from MST, and consequently more clusters will be generated. The  $\epsilon$  used in this case study is **0.0075**. In the end, the MST is partitioned into 29 components (clusters). More about parameter sensitivity will be discussed in section 6.6.



**Figure 6.6.** MST partitioned into 29 clusters (Contains OS data © 2018).

#### **6.4.2 Road geometry generation**

The basic assumption road network generation on the basis of the partitioning of MST performed in section 6.4.1, is that each cluster of buildings should be fully surrounded by road segments. In the road network generation algorithm, the space that road segments may occupy can be derived by constrained Delaunay triangulations (Chew, 1989). It is a *constrained* process, as the triangle is only allowed to be generated, if all its three vertices do not belong to the same cluster. That means it is a Delaunay triangulation *between different clusters*. Entry points are also used in triangulation process. The result of constrained Delaunay triangulation is shown in figure 6.7.

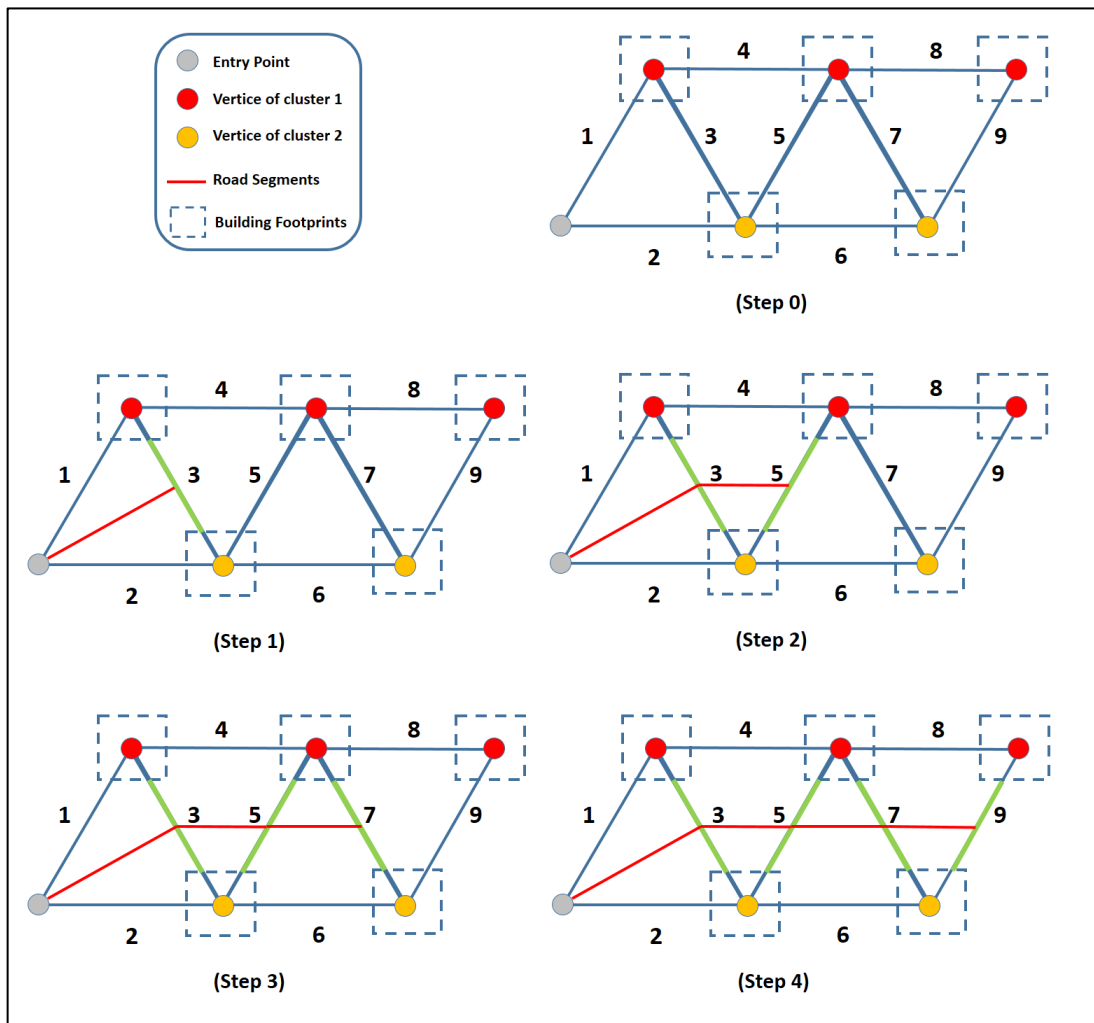


**Figure 6.7.** Constrained Delaunay triangulation result (Contains OS data © 2018).

The generation of road segments is done by traversing topologically touching facets within the Delaunay triangles, starting from any entry point. A simple example (figure 6.8) shows how to exactly generate road segments using triangles. In figure 6.8, there are four triangles, one entry point (step 0). All other points (vertices) are from two clusters. Building footprints will be considered during road network generation. To generate the first road segment (step 1), part of edge No.3 that is not within building footprint is extracted (the green line in step 1), and a line is drawn to connect the entry point and the *midpoint* of green line (on edge 3). Then sequentially midpoint of part of edge No.5, No.7, and No.9 (depicted as green lines) will be used to generate road segments.

The main rationale behind this process is that, only *inter-cluster edge* (if two vertices connecting this edge are from different clusters) will be used to generate road segment, because algorithm assumes road segments *should only* bypass space between different clusters. That is why edge No.4, No.6, and No.8 are not used, since they are all *inner-cluster* edges (if two vertices connecting this edge are from same cluster). The algorithm also assumes that, the road segment (paved between two clusters) should be equally distant to the

two vertices (buildings actually) from two clusters. To avoid collision of road segment with building footprints, for each inter-cluster edge, only part (green line) that is *not within building footprint* is extracted, and the midpoint of *that green line* is used for road generation. The road segment generation will finish, when all triangles are visited. A more detailed version of pseudo code of this process is shown in listing 6.2. Figure 6.9 shows the result of road segments generation in the case study area.



**Figure 6.8.** Simple example about road segments.

---

**Algorithm : Road Generation Process**

---

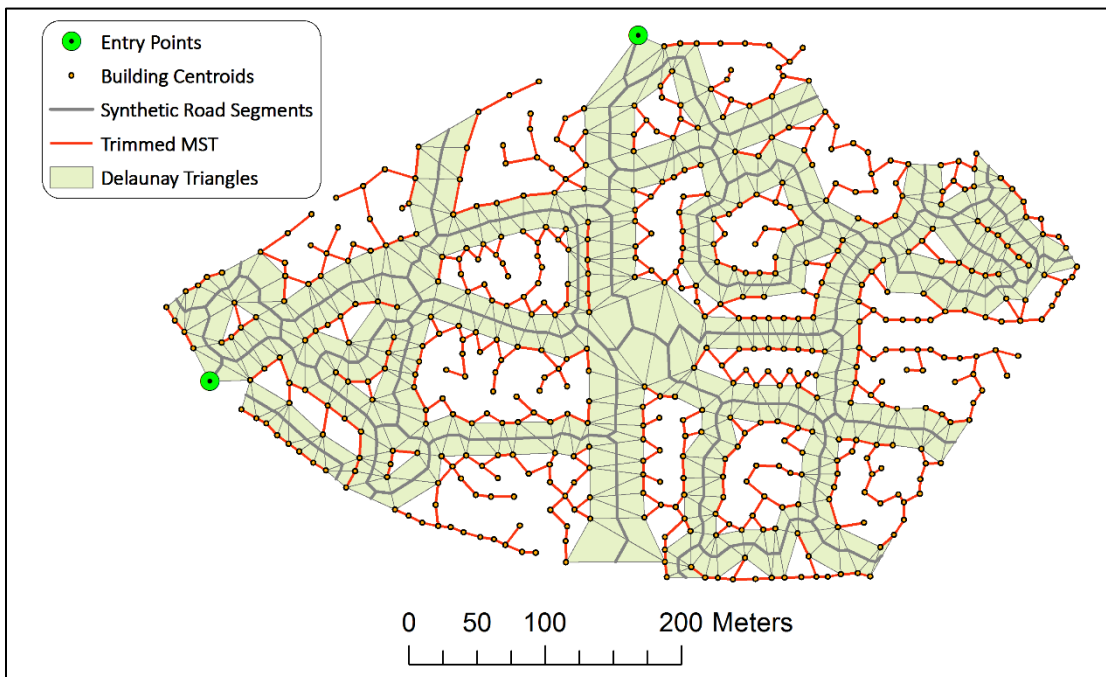
**Input:** The set of delaunay Triangles  $T$ , the set of entry points  $P$

**Output:** The set of road segments generated  $R$

```
1: initialize set  $R$ 
2:  $visited\_all\_triangles = FALSE$ 
3: while  $visited\_all\_triangles == FALSE$  do
4:   for  $p$  in  $P$  do
5:     find the  $triangle\_A$  containing  $p$ 
6:      $triangle\_A\_visited = TRUE$ 
7:      $previous\_mid\_point = p$ 
8:     while there exists a  $triangle\_B$  touching  $triangle\_A$  and  $triangle\_B$ 
hasn't been visited and sharing edge doesn't belong to a same
building cluster do
9:        $triangle\_B\_visited = TRUE$ 
10:       $next\_mid\_point = sharing\_edge.mid\_point$ 
11:      generate road segment  $r$  connecting  $previous\_mid\_point$  and
 $next\_mid\_point$ 
12:      add  $r$  to  $R$ 
13:       $triangle\_A = triangle\_B$ 
14:       $previous\_mid\_point = next\_mid\_point$ 
15:    end while
16:  end for
17: end while
18: return  $R$ 
```

---

**Listing 6.2.** Pseudo code of road segments generation.



**Figure 6.9.** Generated road segments (Contains OS data © 2018).

### 6.4.3 Road geometry optimization

One thing to note from figure 6.9 is that, the geometry of the synthetic road segments is not optimized. In fact, sharp corners can be observed when one road segment connects another. Therefore, it is considered a necessary step to smooth the road segments to be more like real ones. A common algorithm to remove sharp corners is the Chiarkin algorithm (Chiarkin, 1974), which actually *cuts off* 1/4 of each line segment at both ends. An example (figure 6.10) shows how Chiarkin algorithm works, and figure 6.11 shows the smoothed road segments.

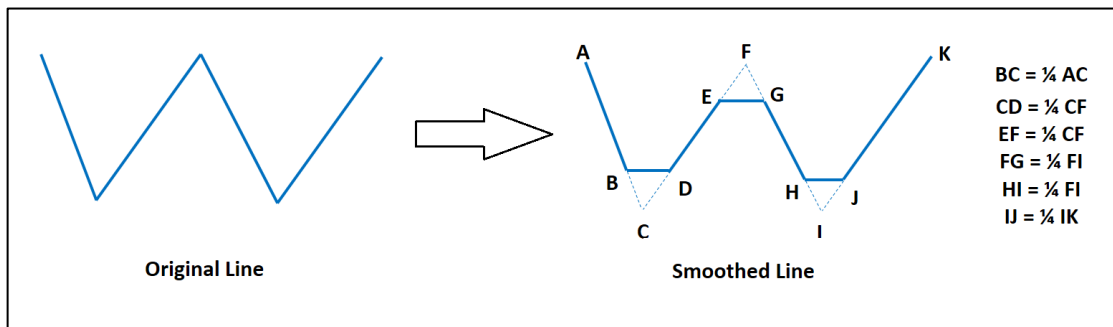
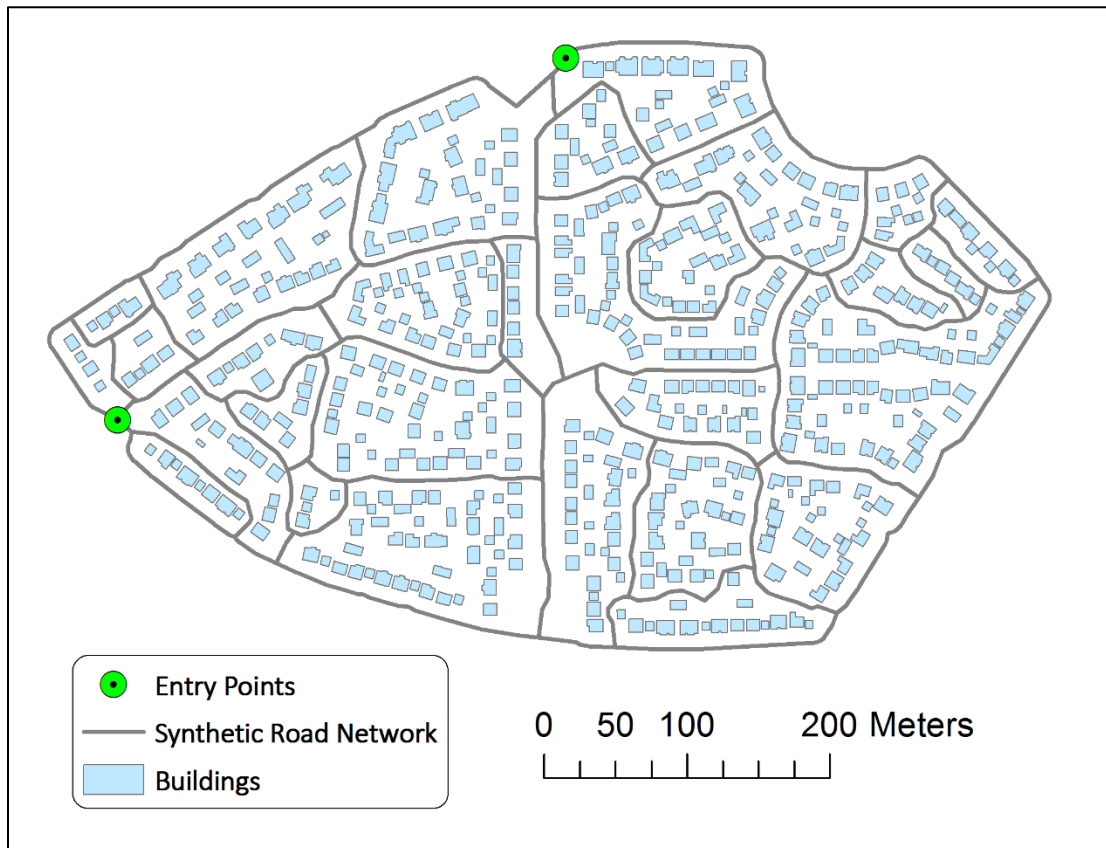


Figure 6.10. Chiarkin algorithm example.



Figure 6.11. Smoothed road segments (Contains OS data © 2018).

After smoothing road segments, the final step is to add an *exterior ring* on the synthetic road network. That is the actually the *exterior ring* of the case study area boundary (figure 6.3). The road network generation algorithm assumes, synthetic road network needs an exterior ring to encapsulate all the buildings. Figure 6.12 shows the final result of synthetic road network.

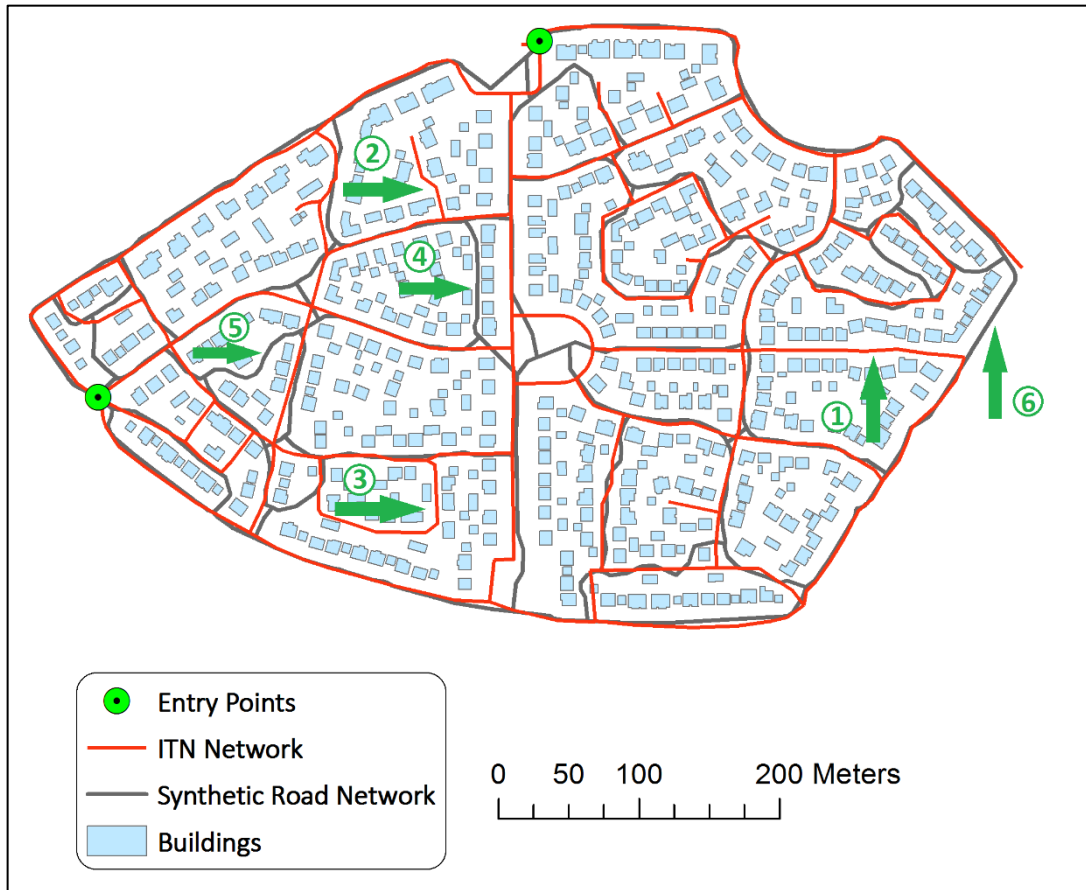


**Figure 6.12.** Final result of synthetic road network (Contains OS data © 2018).

#### **6.4.4 Road Network Validation**

The actual road network is available from the Ordnance Survey Integrated Transport Network (ITN) layer (Ordnance Survey, 2018), which is displayed in figure 6.13. To assess the accuracy of the synthetic road network, spatial comparison and topology comparison are both considered.





**Figure 6.13.** Synthetic and ITN road network (Contains OS data © 2018).

In spatial comparison, error of commission, error of omission, network length difference, and IoU (Intersection over Union) are measured. The error of commission and omission are still based on *buffer approach* defined in section 4.6, and the buffer distance is 10 meters. The IoU is a single metric to assess the fitness of synthetic and actual data (Bates, et al., 2005), and it is calculated as follows, where  $A_{syn}$  is the buffer of the synthetic road network,  $A_{real}$  is the buffer of the ITN network, the  $\cap$  is the intersection operation and the  $\cup$  is the union operation.

$$IoU = \frac{A_{syn} \cap A_{real}}{A_{syn} \cup A_{real}}$$

Table 6.2 shows the spatial comparison between synthetic and ITN road network, indicating a relatively good fitness on geometry of the two network instances.

Commission Error	Omission Error	IoU	Length Difference
5.7 %	4.6 %	92.7 %	0.6 %

**Table 6.2.** Spatial comparison of synthetic and ITN road network.

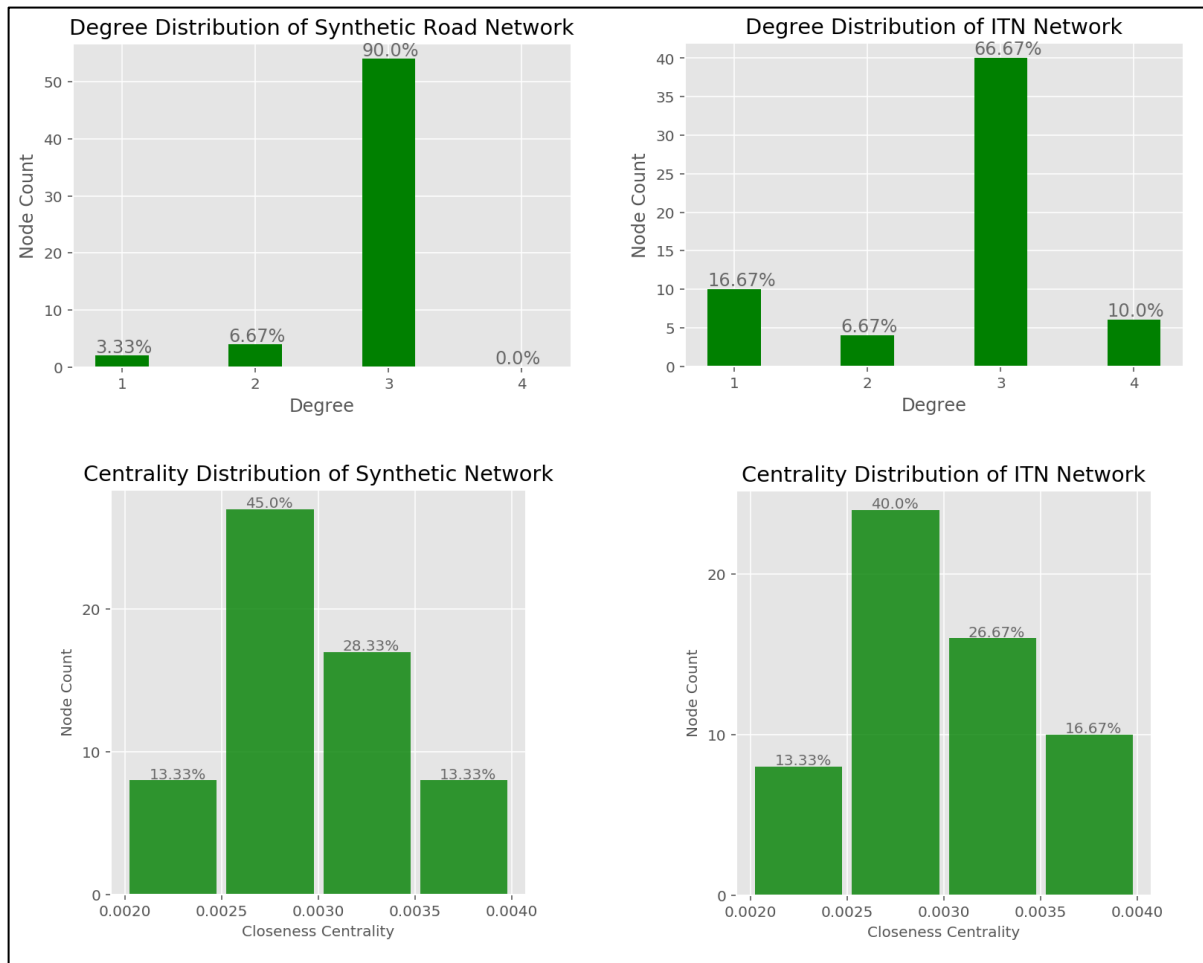
In topology comparison, network size (total number of nodes) are calculated. Moreover, the degree distribution and closeness centrality distribution are the measured as they are the most important indicators of network connectivity and resilience (Porta, et al., 2008). Degree measures how many nodes each node connects. Closeness centrality  $C(u)$  of a node  $u$  is the reciprocal of the average shortest path distance to  $u$  over all  $n-1$  reachable nodes. If  $d(u, v)$  is denoted as the weighted path distance (weight is geometric length of each edge) from node  $u$  to node  $v$ , then  $C(u)$  is defined as follows:

$$C(u) = \frac{n - 1}{\sum_{v=1}^{n-1} d(u, v)}$$

Topology comparison result is displayed in figure 6.14. Both synthetic and ITN road networks have the *same* network size (which is **60**). A majority of nodes in both networks are degree-three nodes (more than 66%). An interesting finding is that the synthetic road network does not have degree 4 nodes, while the ratio of degree-four nodes in ITN network is 10%. This is because the approach for generating road geometry using Delaunay triangles (figure 6.8 and listing 6.2), can only generate node whose degree is 1, 2, or 3. On the other hand, ITN network has more degree-one nodes (16%) than the synthetic network, this is because ITN network (in this area) allows road segments (e.g. arrow No.2 in figure 6.13) that are inside a *building cluster*, but it is not allowed by algorithm (and thus not allowed in the synthetic road network).

Despite the slightly different degree patterns, both networks have similar patterns in closeness distribution. A majority of nodes (40 – 45 %) have closeness value between 0.0025 and 0.0030. Then fewer nodes (26 - 28%) have closeness value between 0.0030 and 0.0035. The remaining nodes are split in two groups (each accounting for around 15%), with closeness

values of 0.0020 – 0.0025 and 0.0035 – 0.0040. Such similar closeness distribution pattern indicates both networks are similar in terms of resilience.



**Figure 6.14.** Topology comparison of synthetic and ITN road network.

From table 6.2 and figure 6.13, despite the high spatial accuracy, there are still errors of commission and omission. For example, in figure 6.13, the green arrows No.1, No.2, and No.3 indicate the areas where the algorithm fails to generate a road (where there should have been). For location No.1 and No.3, the algorithm fails to recognize there is more than one cluster of buildings in these locations. For location No.2, the algorithm cannot generate synthetic roads that *insert into* a cluster of buildings, as synthetic roads must fully encapsulate a cluster of buildings. On the contrary, the green arrows No.4 and No.5 show the algorithm mistakenly generates a road (when it should not). Still the clustering process is the reason, where the algorithm recognizes more than two clusters in these locations, but in fact should be only one (according to the ITN data).

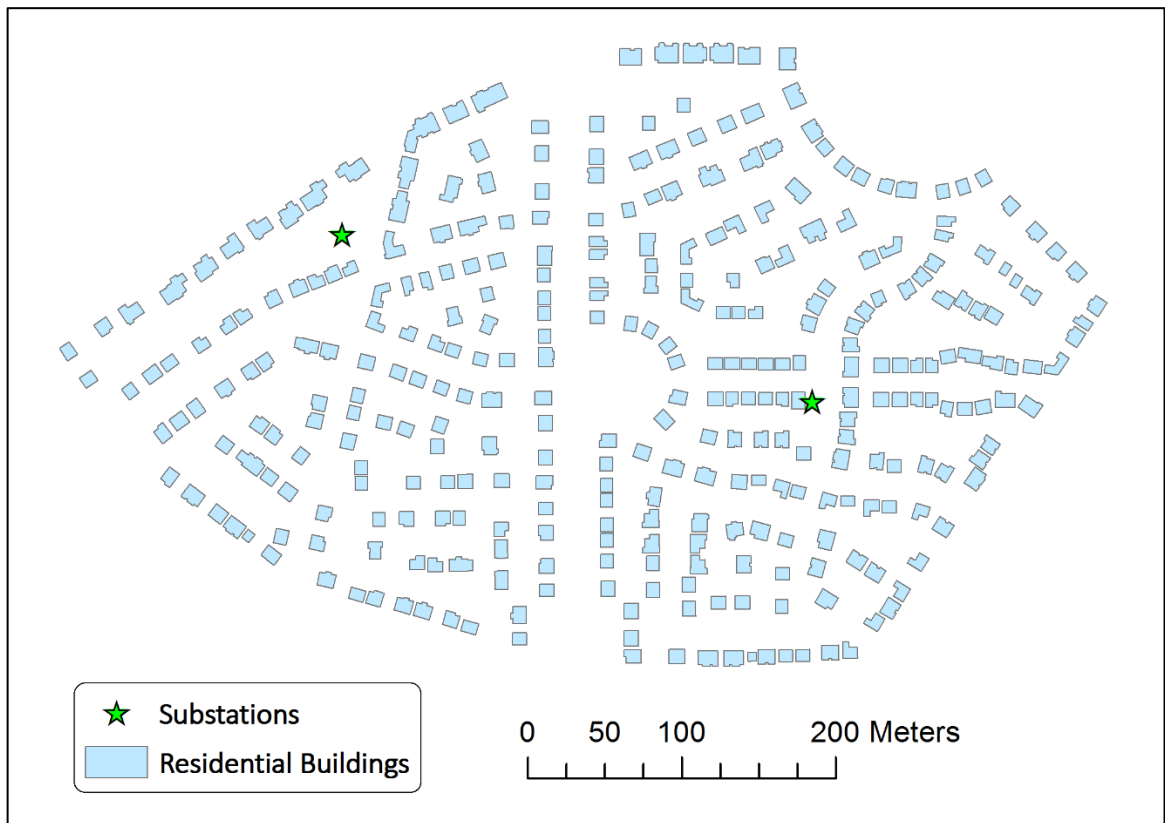
Interestingly, it is observed that over-commission of synthetic road network also occurs at the location indicated by the green arrow No.6. It seems the actual road network is not completely closed and there is a gap. An interesting question is *should the algorithm produce a closed synthetic road network (based on the input boundary)?* The answer is yes in the author's opinion. When observing the Google Map (figure 6.1), it is clear that there does exist a road in this location (northeast corner). But possibly this is only a small road (or this road is relatively new), so that ITN network data does not include it. Therefore, an *external closed boundary* of the synthetic road network is considered necessary for the algorithm.

## 6.5 Electricity Distribution Network Generation

### 6.5.1 Synthetic Electricity Network generation

As synthetic road network is available, it is possible to generate synthetic electricity distribution network. Two substation points in this area are identified and downloaded from Ordnance Survey Point of Interest layer (Ordnance Survey, 2018). The generic spatial heuristic algorithm (Chapter 4), will be used to generate electricity distribution network in the case study area. Before generating electricity network, buildings will be filtered first, and only those whose areas are larger than 30 m<sup>2</sup>, are kept.

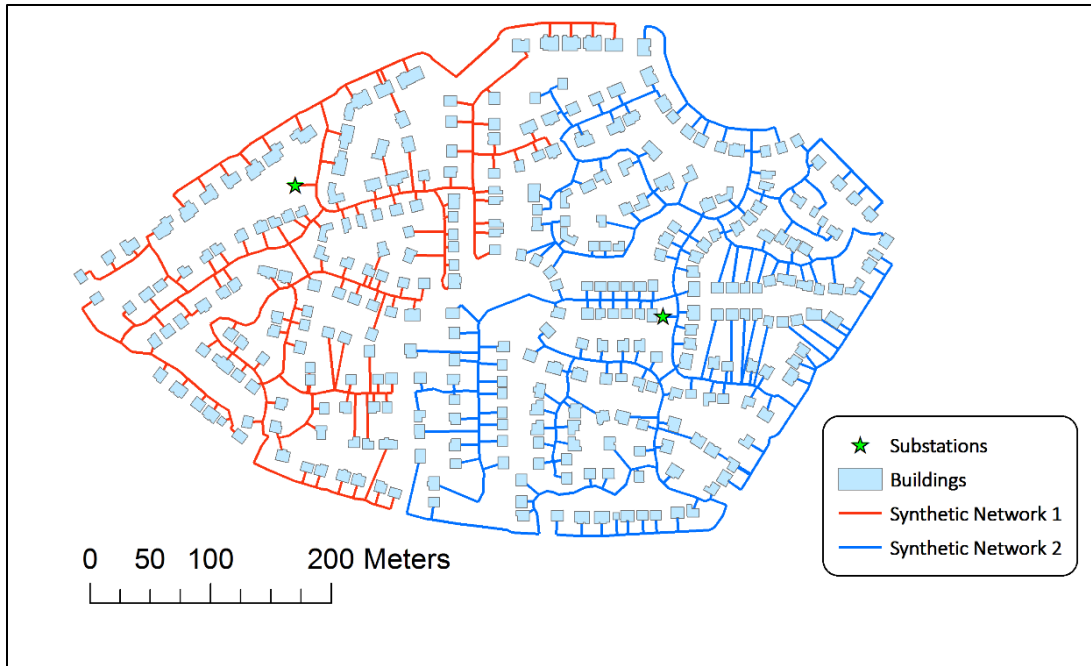
That is because smaller buildings are considered to be the buildings that do not require infrastructure services (Barr et al., 2017). This operation is also done here, which leaves 332 buildings (now termed *residential buildings*) (figure 6.15). The synthetic electricity distribution networks (termed *synthetic network 1 and 2*) generated are shown in figure 6.16.



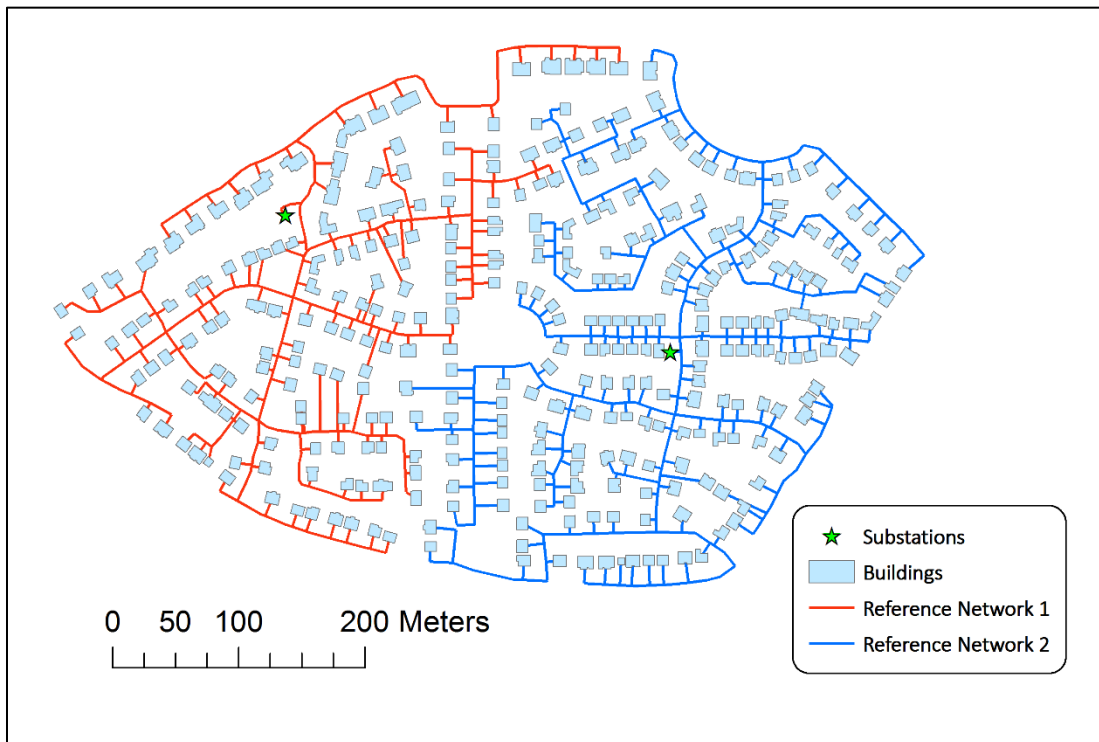
**Figure 6.15.** Residential buildings (area > 30m<sup>2</sup>) reserved for the case study area (Contains OS data © 2018).

To validate the synthetic network 1 and 2, the best option is to use actual electricity network data. But unfortunately, this case area is a relatively new developing site, and Northern Power Grid (local electricity supplier) does not have record on the spatial layout of electricity distribution network.

However, it is considered feasible to use the electricity network generated based on the actual road network (ITN network) as the *reference* data for validation. This way, it is still possible to evaluate how *the difference between synthetic and actual road layout* affects generation of electricity network layout. The reference data are termed reference *network 1 and 2*.



**Figure 6.16.** Synthetic electricity network generated (**based on synthetic road network**)  
(Contains OS data © 2018).



**Figure 6.17.** Reference electricity network generated (**based on ITN network**) (Contains OS  
data © 2018).

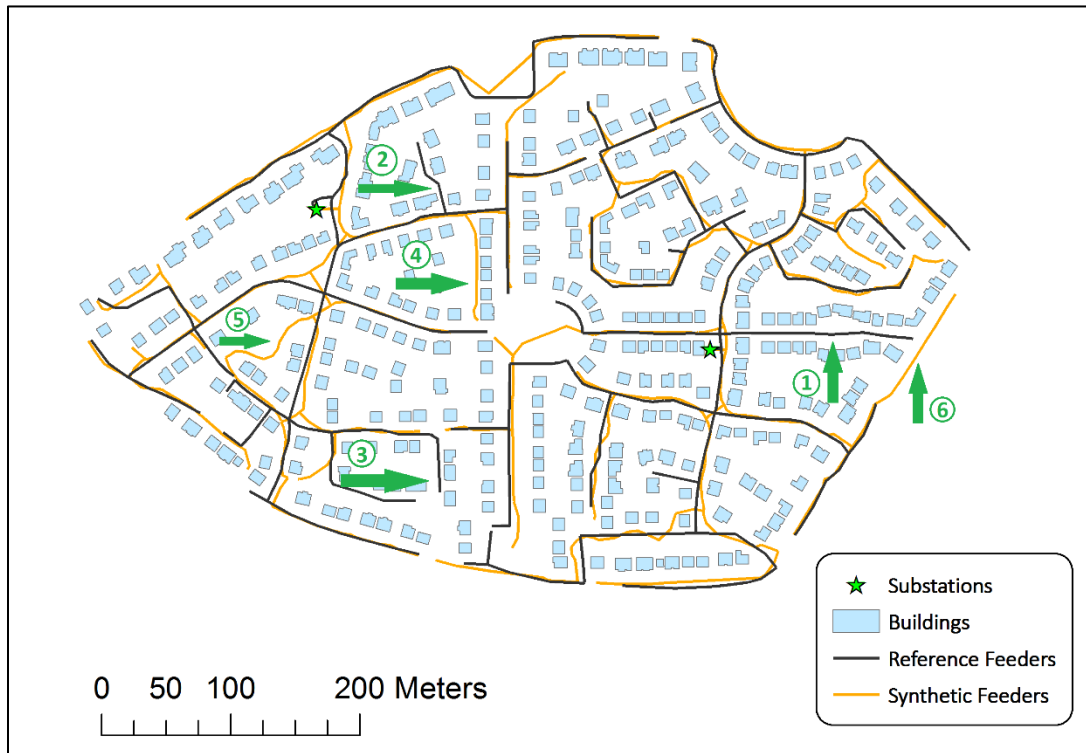
Validation (or more strictly speaking, comparison) will be done on the feeder network as well as building-substation dependency. For comparing feeder network, spatial and topology comparisons will be done (same as validating synthetic road network). The building-substation dependency comparison will be explained later.

First of all, it is worth pointing out that only feeders (the back-bone cables of the electricity distribution network) will be compared, and there will be no consideration on the service lines (cables directly connect to buildings). *The inclusion of service lines, will introduce many degree-one and degree-three nodes in the networks, resulting a skewed degree and centrality distribution (i.e. makes the topology comparison not indicative).*

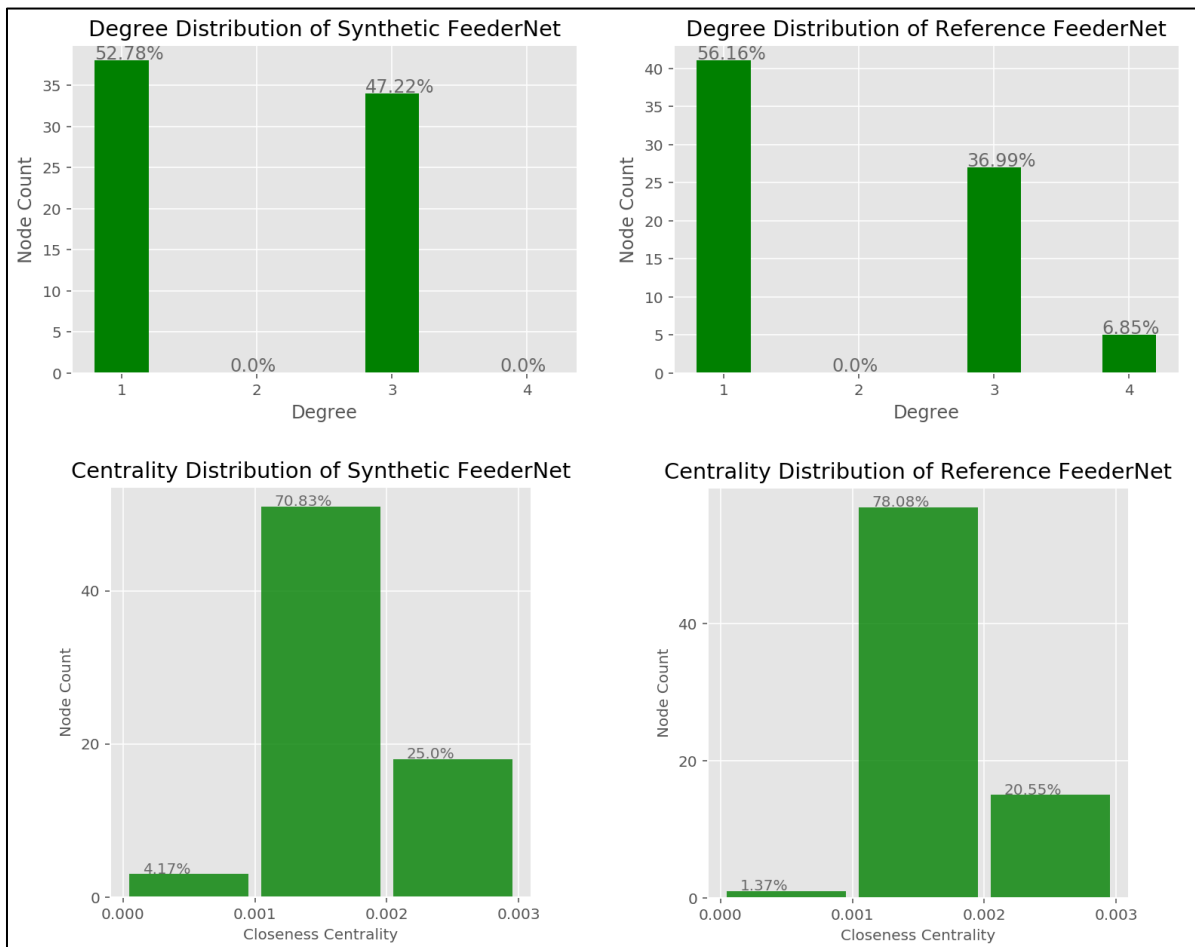
Therefore, only feeder networks of the electricity distribution networks are extracted for comparison (figure 6.18). Spatial and topology comparison results are shown in table 6.3, and figure 6.19. The sizes of reference and synthetic feeder networks are 73, and 72 respectively, almost the same.

<b>Commission Error</b>	<b>Omission Error</b>	<b>IoU</b>	<b>Length Difference</b>
6.3 %	5.4 %	91.2 %	0.5 %

**Table 6.3.** Spatial comparison on the reference and synthetic feeder networks.



**Figure 6.18.** Synthetic feeders and reference feeders (Contains OS data © 2018).

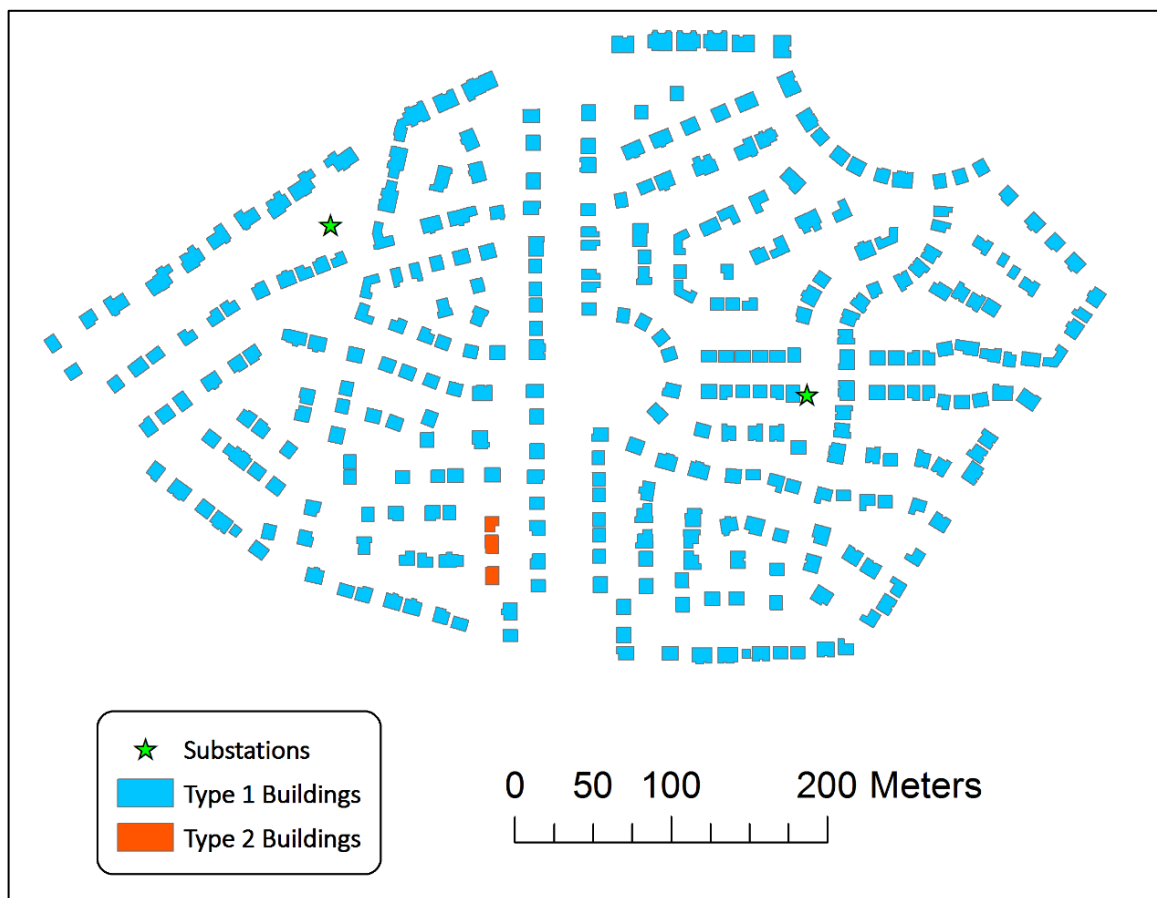


**Figure 6.19.** Topology comparison of the synthetic and reference feeder networks.



From table 6.3, and figure 6.18, it is found that commission and omissions errors on synthetic feeders are still relatively small. The error patterns in synthetic feeders (green arrow No.1, 2 and 3 for omission errors, No.4, 5, and 6 for commission errors) are similar as that in synthetic roads (figure 6.13), since the spatial heuristic algorithm (to generate electricity network) highly depends on roads. Moreover, from figure 6.19, despite the spatial discrepancy, two networks show similar topological features, as demonstrated by the degree and closeness centrality distributions.

Although there exist some discrepancies between the synthetic and reference feeders, a more important thing is to compare the building-substation dependency. That is to say, does every building depend on the same substation, from the synthetic networks and from reference networks? Table 6.4 shows the validation result for building-substation dependency, and figure 6.20 shows the visual result.



**Figure 6.20.** Visual result of building-substation dependency (Contains OS data © 2018).

Building type	Quantity
<b>Type 1:</b> The building depends on the same substation according to synthetic and reference networks.	329 (99%)
<b>Type 2:</b> The building depends on the different substations according to synthetic and reference networks.	3 (1%)

**Table 6.4.** Building-substation dependency comparison result.

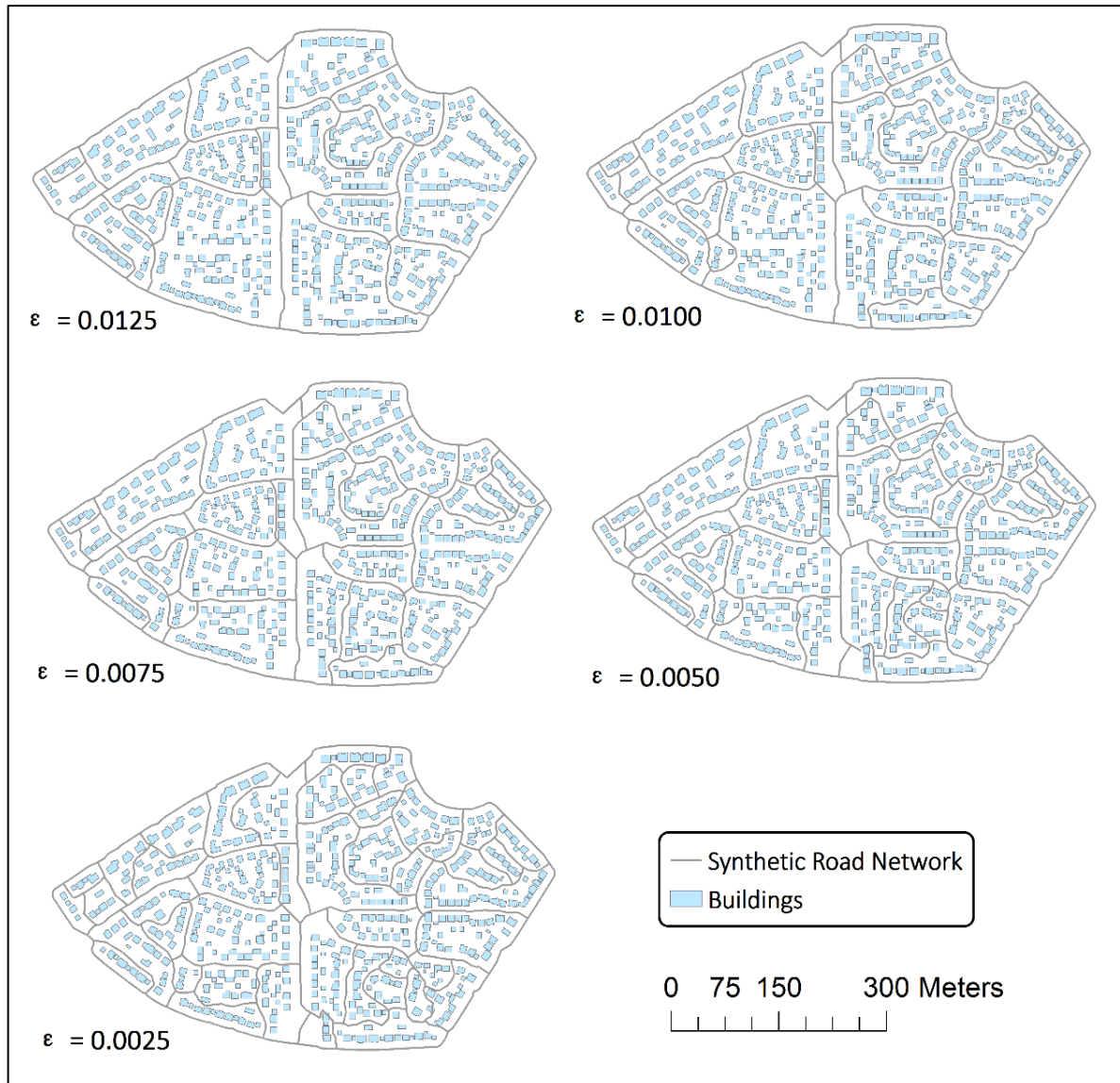
It is found that, 99% of buildings depend on the *correct* substation, according to the reference network data, which shows high accuracy. The error occurs on only 3 buildings. The cause is the omission error (green arrow No.3) on the synthetic road network (figure 6.18), which affects building-substation assignment in the spatial heuristic algorithm to generate electricity distribution networks.

## 6.6 Parameter Sensitivity Test

Until now, there is still one important thing that has not been covered in the road network generation algorithm. That is the choice of  $\epsilon$  value in the MST partitioning algorithm in section 6.4.1. The author of this algorithm, Zhu et al. (2009) mentioned when using this algorithm, the  $\epsilon$  value needs be carefully chosen depending on the actual application. Therefore, this chapter will explore parameter sensitivity of  $\epsilon$  in generating synthetic road network, and justify the choice of value **0.0075**, used previously. Synthetic road networks generated based on five different  $\epsilon$  values are shown in figure 6.21, and are evaluated in table 6.5.

Value of $\epsilon$	Total network length (m)	Error of omission	Error of commission
0.0125	4906	8.2 %	5.9 %
0.0100	5729	6.9 %	6.3 %
0.0075	5992	5.4 %	6.3 %
0.0050	6516	5.2 %	7.6 %
0.0025	7452	5.2 %	8.5 %

**Table 6.5.** Evaluation of synthetic road network based on different  $\epsilon$  values.



**Figure 6.21.** Parameter sensitivity of  $\epsilon$  (Contains OS data © 2018).

Figure 6.21 and table 6.5 indicated that, as  $\epsilon$  value decreases, the MST partitioning algorithm will stop later. That means more clusters will be generated, and as a result, more road segments will be generated, so that total length of synthetic road network also increases. With more synthetic road segments being generated, it is easy to prove the error of omission always decreases. On the other hand, error of commission always increases.

That is interesting because, it is preferred that synthetic road network should have *both* low errors of commission and omission. From table 6.5, when  $\epsilon$  decreases from 0.0075 to 0.0050 or 0.0025, the error of omission drops from 5.4 % (already a small value) to 5.2 %, which is good, but improvement is not obvious. However, the error of commission has a significant

increase from 6.3 % to 7.6% and to 8.5%. Therefore, choosing 0.0050 and 0.0025 as  $\epsilon$  value is not a good idea. On the other hand, if using the value 0.0125, error of omission is too large (8.2 %) to be acceptable.

Therefore, it is considered that the value 0.0100 and 0.0075 are appropriate to use. In fact, from figure 6.21, the two synthetic road networks generated from these two  $\epsilon$  values are almost identical. The particular reason to choose 0.0075 in our case study, is that error of omission and commission from this  $\epsilon$  value are both smaller.

## **6.7 Transferability Test**

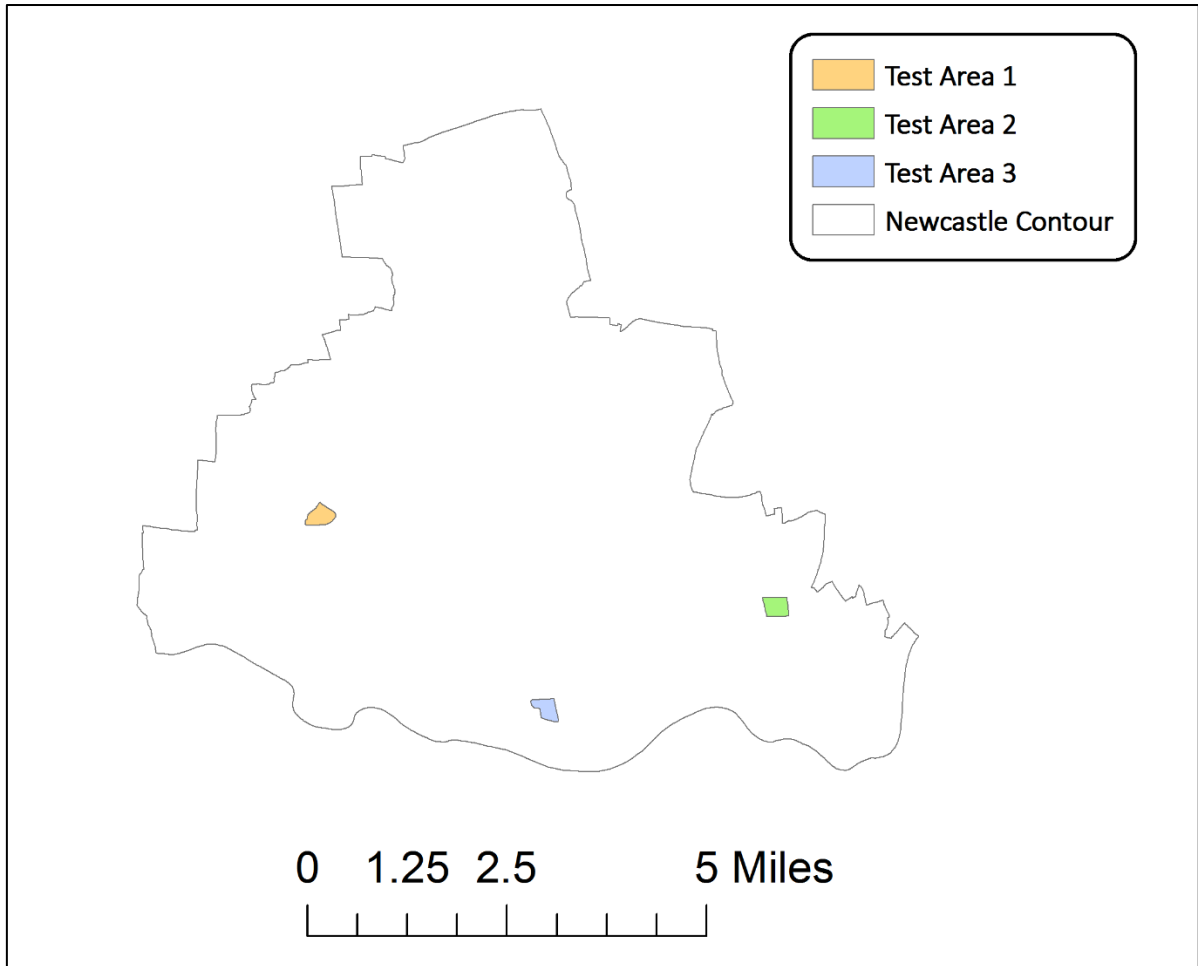
The road network generation algorithm was developed using the data from the small case study area (figure 6.1, figure 6.2), and has achieved relatively good performance. However, this algorithm is developed as a *generic* solution for generating road network in urban areas as long as necessary input data (buildings, entry points, and boundary) are available. The algorithm (and more importantly, the  $\epsilon$  value) should not over-fit to the case study area. That is to say, the algorithm should be generalized well and also has still good performance on input data from other areas.

In this section, a test was done to generate road network (and electricity network later) in three more areas in Newcastle upon Tyne, to explore the transferability of the road network generation algorithm.

### **6.7.1 Data Sets**

The basic information for these tests area is shown in table 6.6. The location of these three areas in Newcastle are shown in figure 6.22. Figure 6.23, 6.24, and 6.25 show the input data of these three areas. The test areas are carefully chosen in three aspects: (1) The test area size is close to the case study area (about 197,000 m<sup>2</sup>); (2) Major building layout in the test areas

are different; (3) Number of buildings in each test area is different. The choice of test areas helps better test the transferability of algorithm, when building layout and building density (number of buildings over area size) is different from the case study area.



**Figure 6.22.** Location of the three test areas in Newcastle.

Area	No. Buildings	Major Building Layout	Size (m <sup>2</sup> )
No.1	250	Detached	188,200
No.2	703	Terrace	223,400
No.3	553	Semi-Detached	207,900

**Table 6.6.** Basic information of test area.

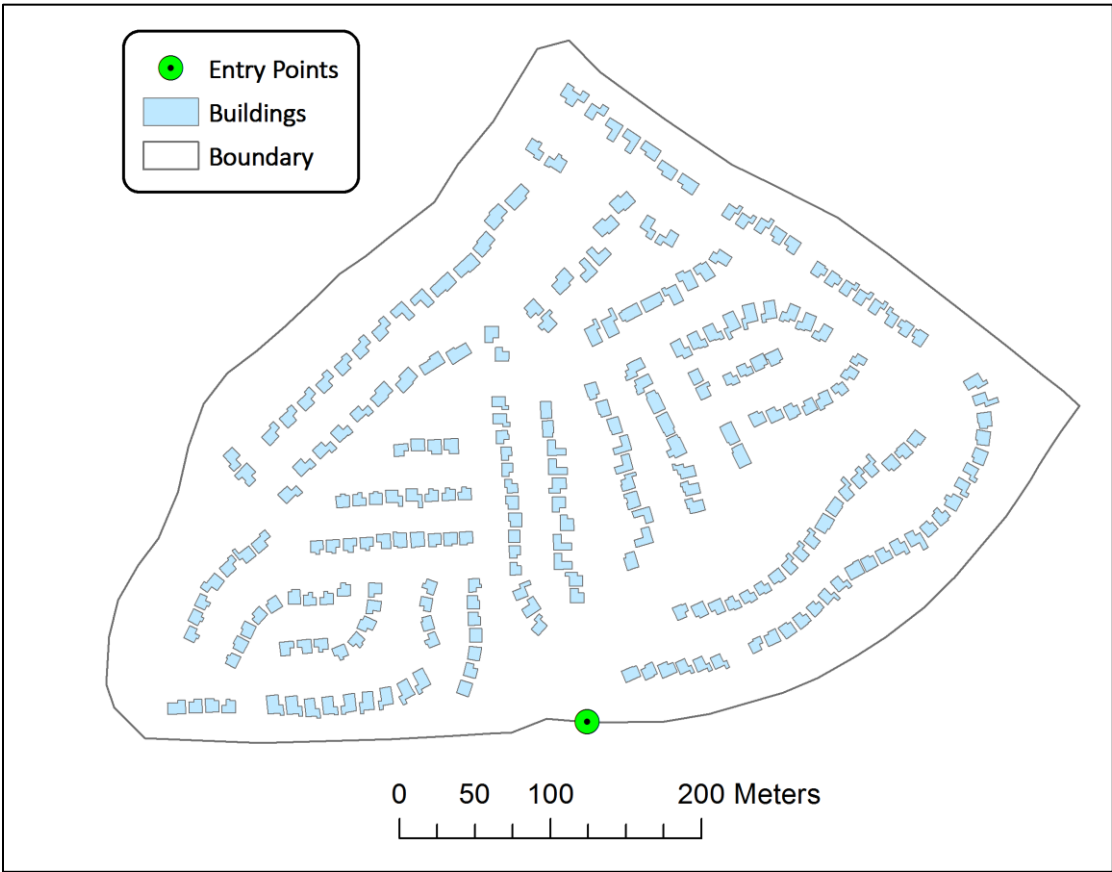


Figure 6.23. Input data for test area 1 (Contains OS data © 2018).

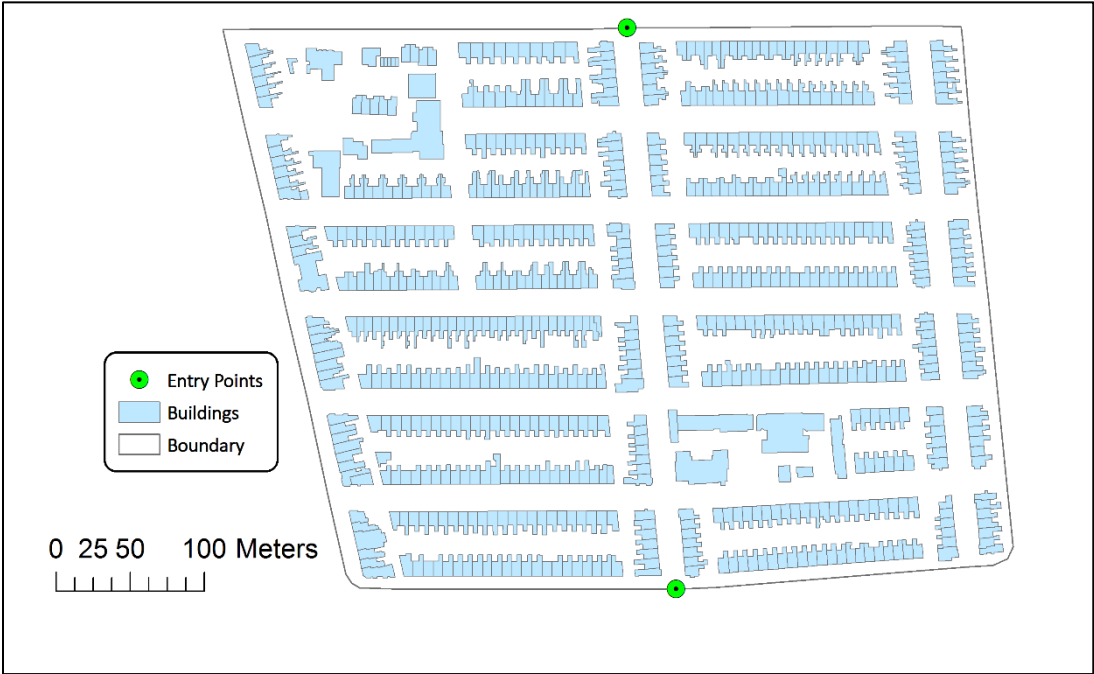
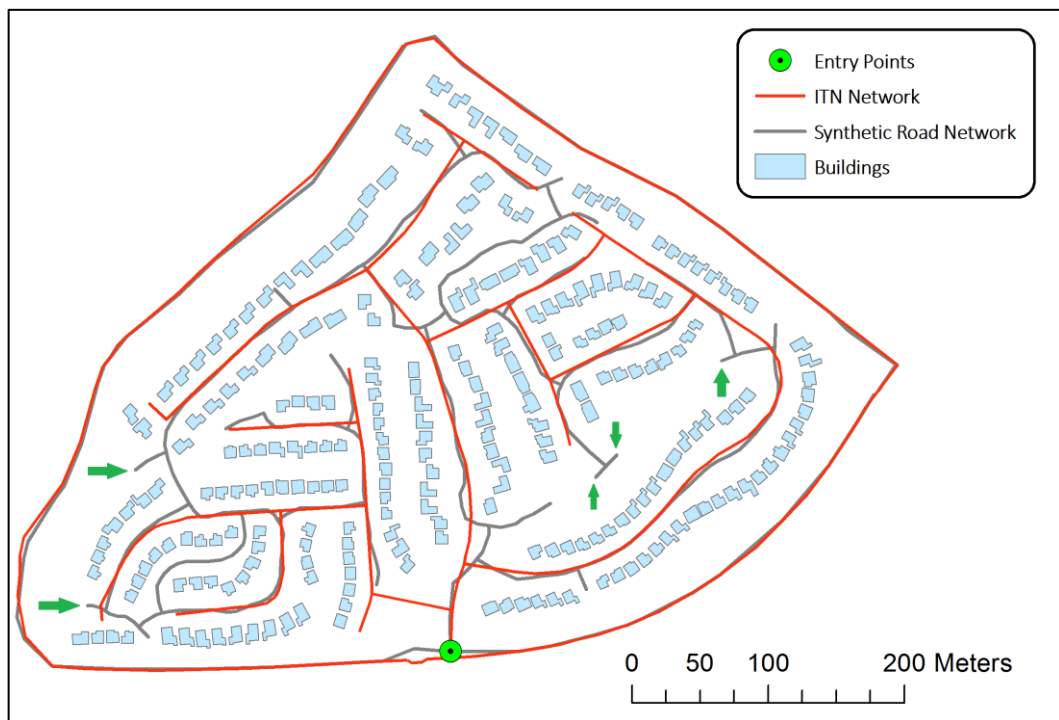


Figure 6.24. Input data for test area 2 (Contains OS data © 2018).



**Figure 6.25.** Input data for test area 3 (Contains OS data © 2018).

### 6.7.2 Results and Validation



**Figure 6.26.** Synthetic and ITN road network in test area 1 (Contains OS data © 2018).



**Figure 6.27.** Synthetic and ITN road network in test area 2 (Contains OS data © 2018).



**Figure 6.28.** Synthetic and ITN road network in test area 3 (Contains OS data © 2018).

First synthetic road networks are generated in the three test areas (figure 6.26, 6.27, and 6.28).

Note for each area, the  $\epsilon$  value is still 0.0075. Spatial and topology comparisons (table 6.7,



table 6.8, figure 6.29, figure 6.30) are made to assess the accuracy of synthetic road networks, compared with ITN networks.

In all of the three areas, the spatial discrepancy on the road networks are small, and the difference on the network sizes is more obvious (in all three areas). In area 1, synthetic road network has many small segments (indicated by green arrows in figure 6.26) which contributes to more nodes being generated. While in area 3, it is the opposite case, as the real ITN network has more small segments (indicated by green arrows in figure 6.28) and thus has a larger size.

Figure 6.29 suggests that synthetic and ITN networks in *all three areas* have similar degree distributions. Figure 6.30 shows that there is discrepancy in centrality distribution, especially in area 1 and area 3, which is mainly caused by network-size difference. In area 1, synthetic road network has more *small segments*, and nodes on these small segments have relatively low centrality values (0.002 - 0.003). While in area 3, ITN road network has more *small segments* and nodes on these small segments have relatively low centrality values (0.002 – 0.003).

Area	Commission Error	Omission Error	IoU	Length Difference
No.1	5.1 %	5.5 %	91.7 %	7.9 %
No.2	5.7 %	3.2 %	93.2 %	1.3 %
No.3	3.6 %	6.5 %	91.6 %	5.5 %

**Table 6.7.** Validation of synthetic road network in testing areas.

Network Size (Node Count)	ITN Network	Synthetic Road Network
Area 1	24	44
Area 2	84	116
Area 3	28	16

**Table 6.8.** Network size of ITN and synthetic road networks in three areas.

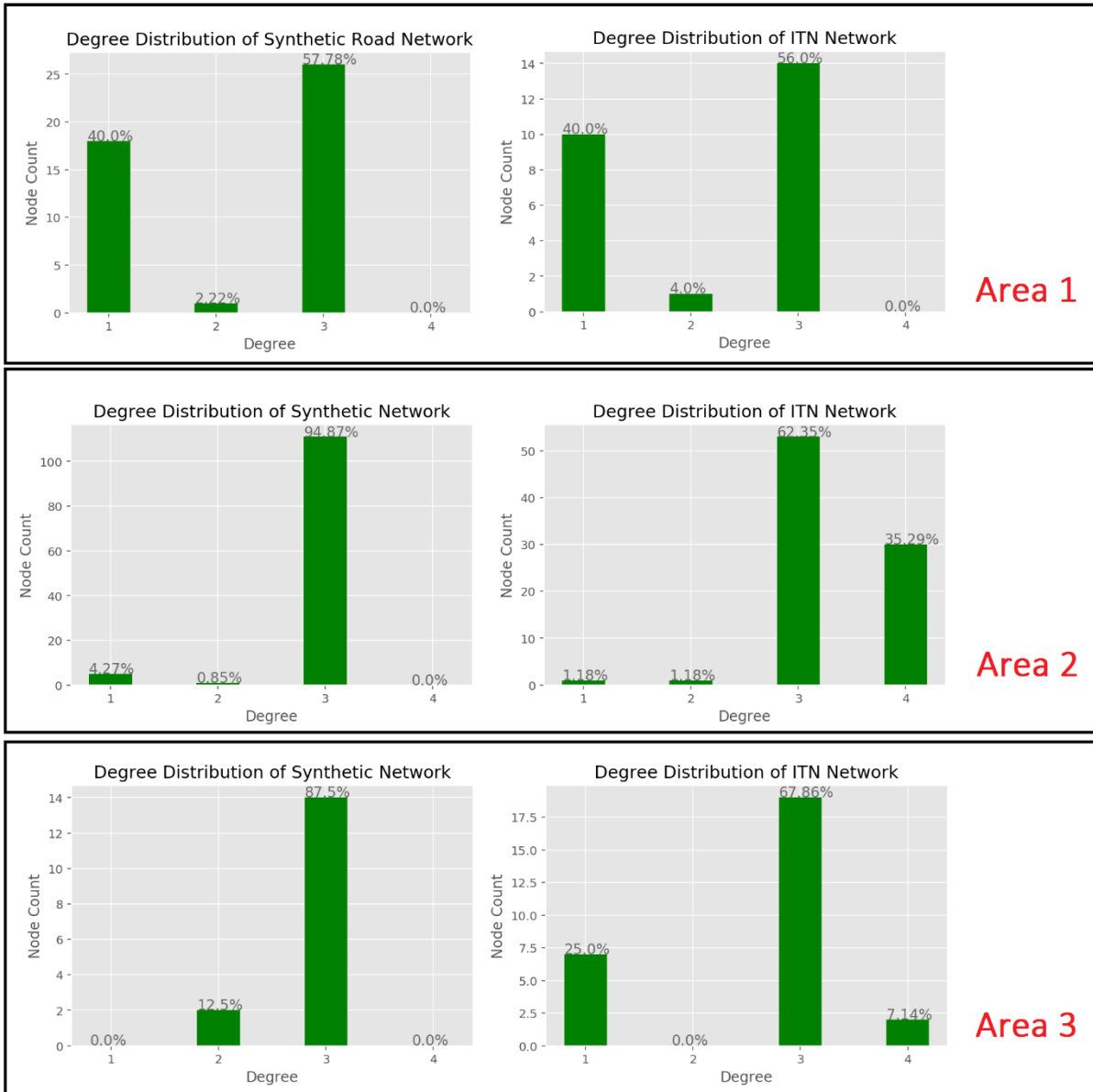
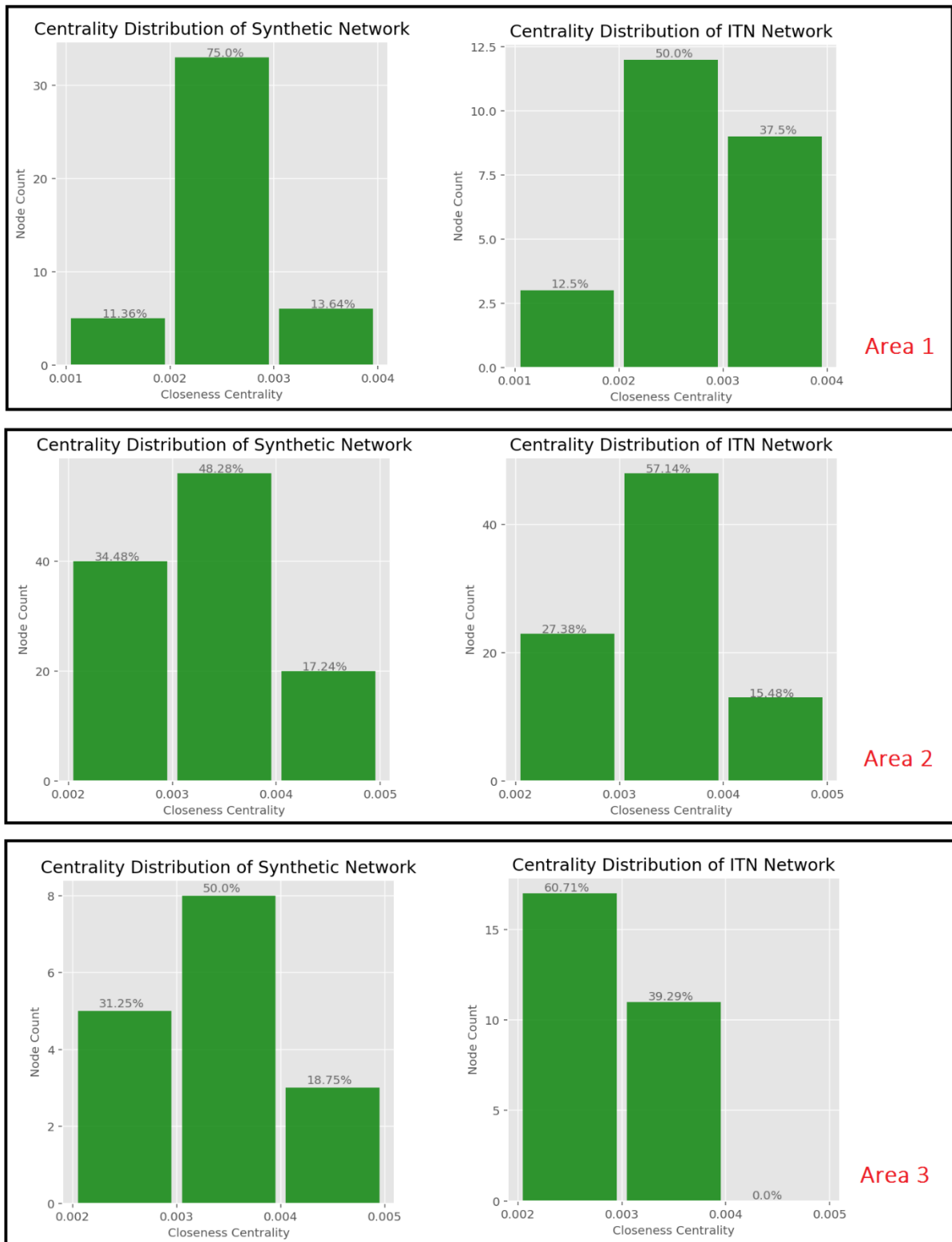


Figure 6.29. Degree distributions of synthetic and ITN road networks in 3 test areas.



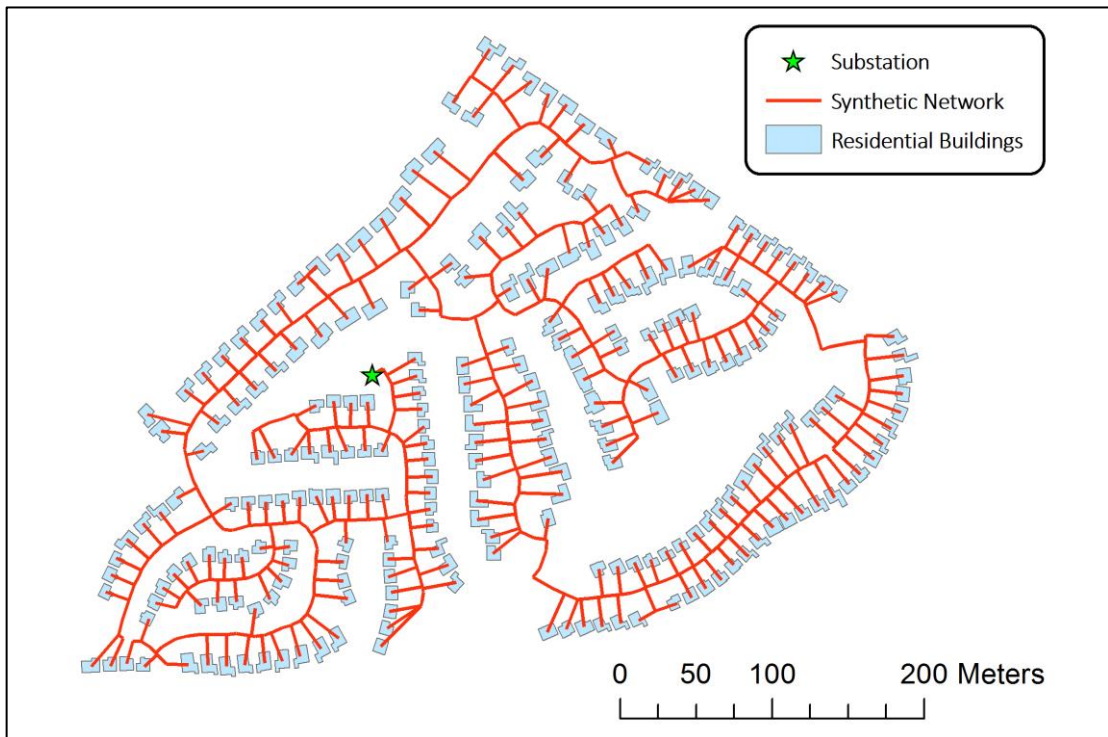
**Figure 6.30.** Closeness centrality distribution of synthetic and ITN networks in 3 test areas.

Another interesting finding during the spatial comparison is that, there is no over-commission at the boundary, as real roads at boundary of the three areas are all recoded in the ITN network data. This is good (in terms of accuracy), but it also reveals a potential limitation of

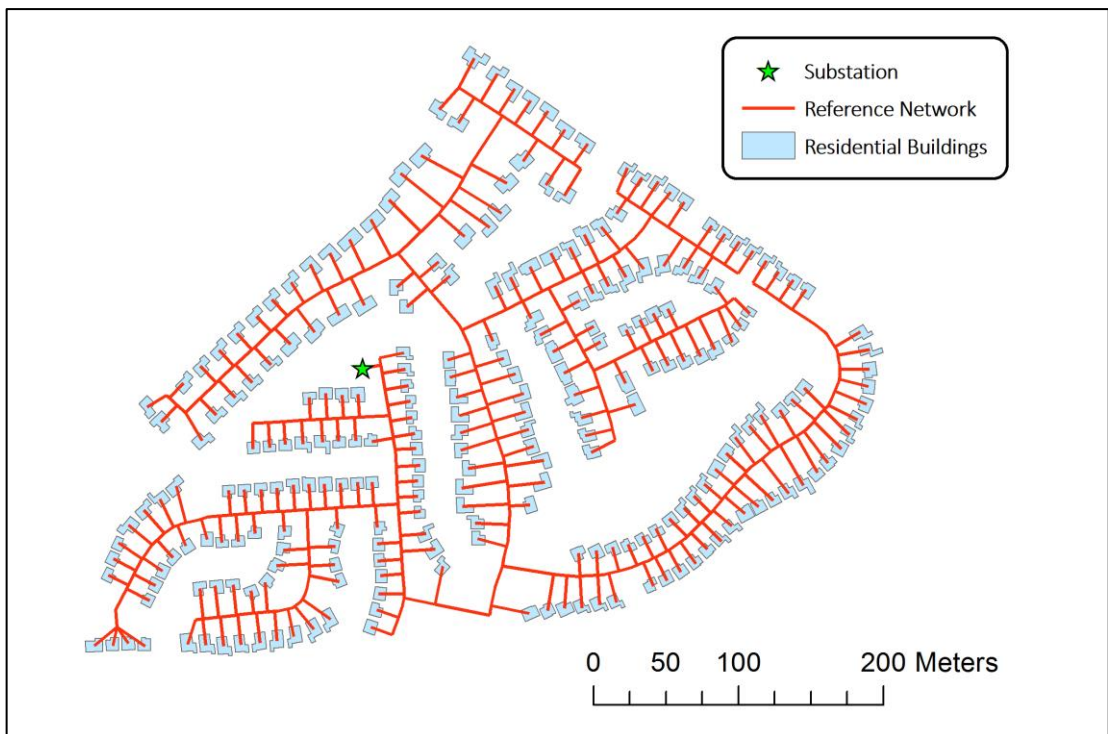
the algorithm. The algorithm assumes there should exist an *exterior ring* on the road, but this *exterior ring* cannot be automatically generated by the algorithm, but instead it must be given as one input (the boundary). Constrained Delaunay triangulation process (section 6.4.2, figure 6.7) causes this limitation. For any point A (the centroid of a building) that is already on the boundary of the area, there is *no outside point* that can make triangulation with point A, which means on the outside of point A, it is *impossible* to generate the geometry of a road segment. That is why a boundary must be given as an input, and if possible, this had better be the exterior ring that can represents the actual road network at the boundary.

Regardless of this limitation, the validation result indicates that in spite of different building layout and different building density, the algorithm (and more importantly, the  $\epsilon$  value 0.0075) can generate plausible layout of road network in all these three areas. This is essential, because it shows this algorithm has been generalized and can be applied as a generic approach that is scalable (regardless of input area size) and transferable (regardless of the building layout in the area).

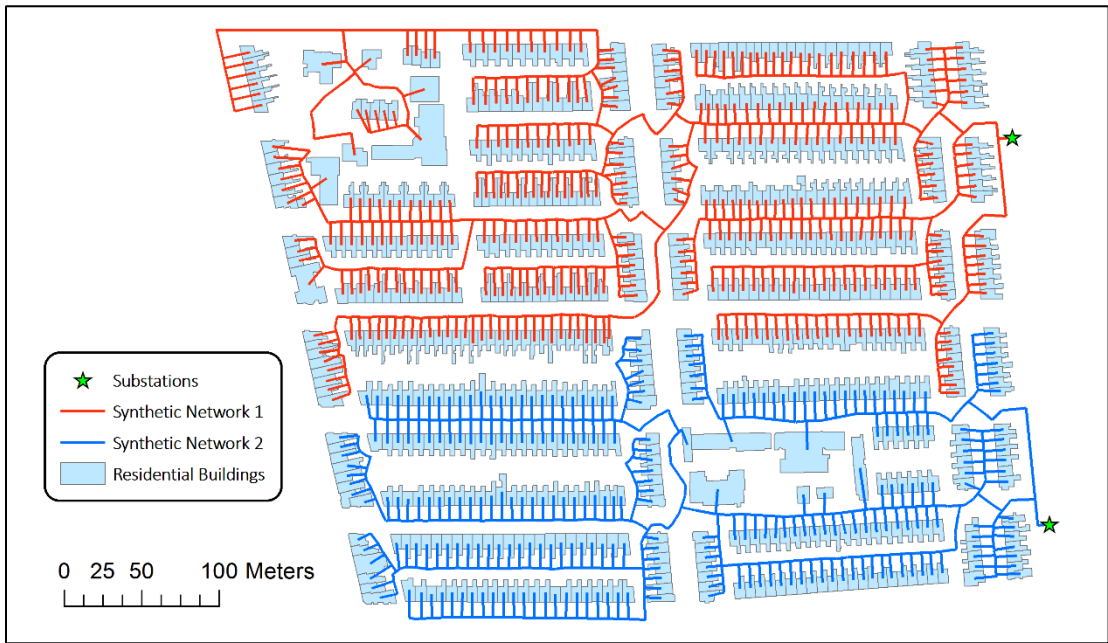
After synthetic road networks are generated, it is possible to generate layout of electricity distribution networks, using layout of residential buildings (area  $> 30\text{m}^2$ ), and substations as additional input data. Figure 6.31, 6.33 and 6.35 show the synthetic electricity distribution networks (termed *synthetic networks*) generated in these areas. For validation, electricity distribution networks generated based on ITN network are shown in figure 6.32, 6.34, and 6.36, and they are termed *reference networks*.



**Figure 6.31.** Generated electricity network in test area 1, based on **synthetic road network**  
 (Contains OS data © 2018).



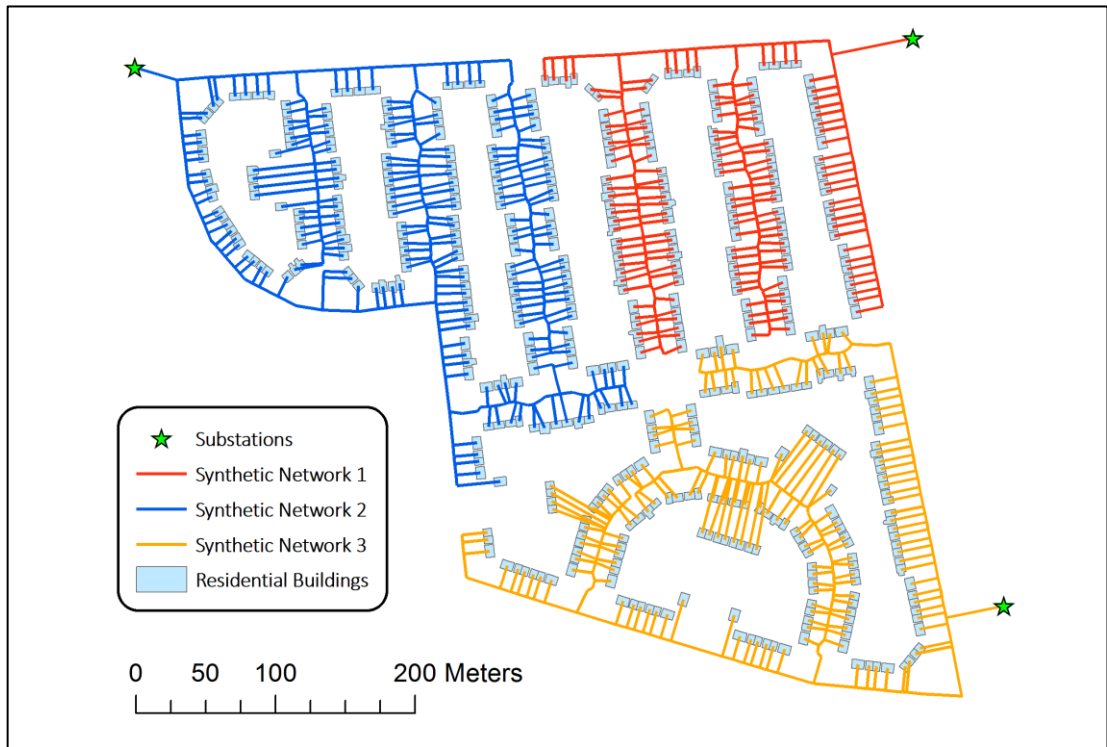
**Figure 6.32.** Generated electricity network in test area 1, based on **ITN road network**  
 (Contains OS data © 2018).



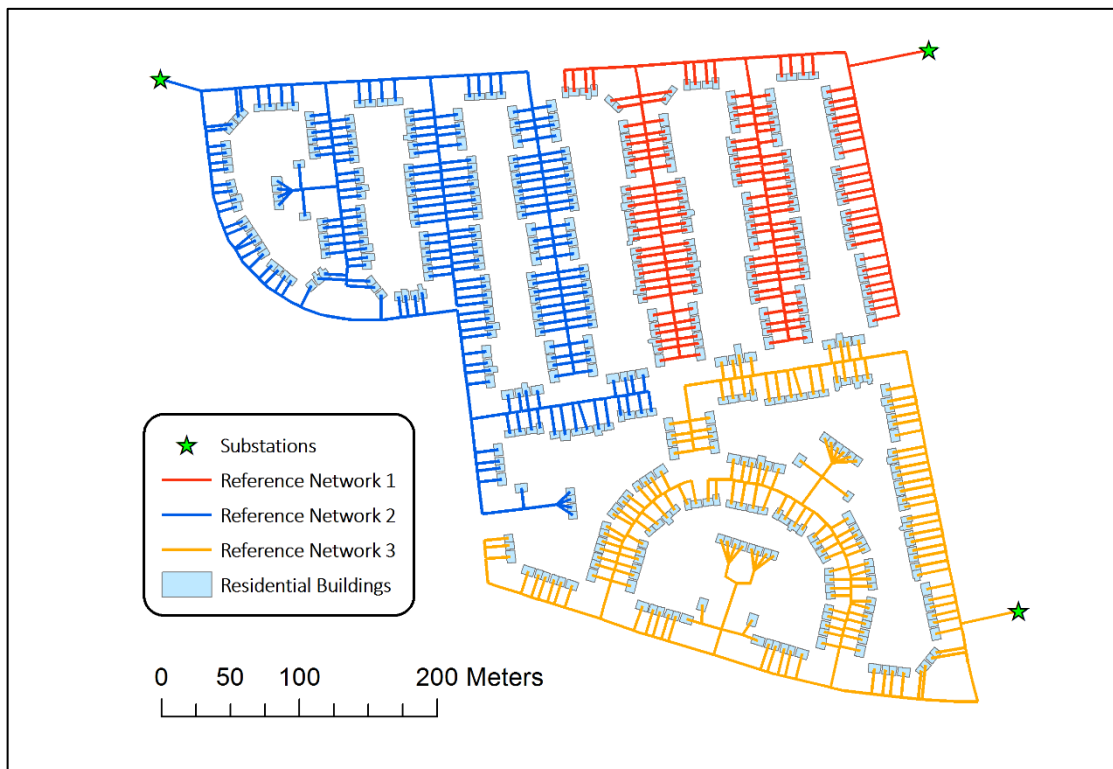
**Figure 6.33.** Generated electricity network in test area 2, based on **synthetic road network**  
 (Contains OS data © 2018).



**Figure 6.34.** Generated electricity network in test area 2, based on **ITN road network**  
 (Contains OS data © 2018).

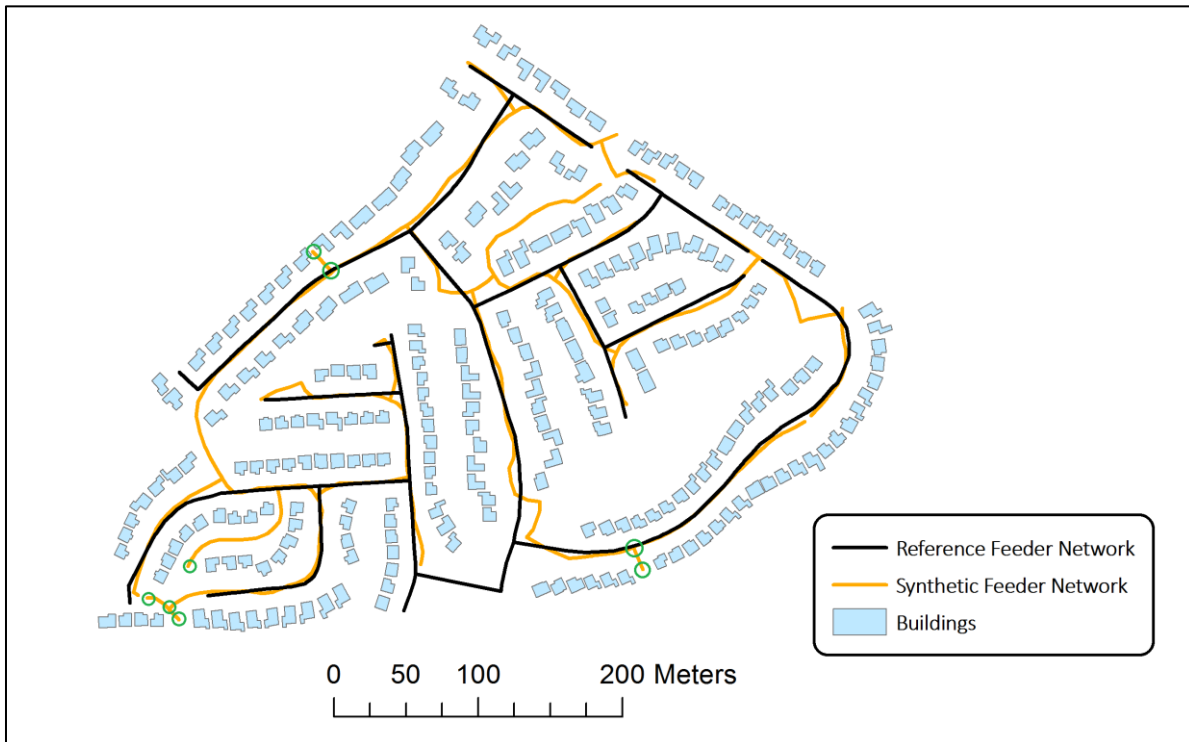


**Figure 6.35.** Generated electricity network in test area 3, based on **synthetic road network** (Contains OS data © 2018).



**Figure 6.36.** Generated electricity network in test area 3, based on **ITN road network** (Contains OS data © 2018).

Then (synthetic and reference) feeder networks for the three test areas are extracted (figure 6.37, 6.38, and 6.39) for spatial and topology comparisons. The comparison results are shown in table 6.9, table 6.10, figure 6.40 and figure 6.41 respectively.

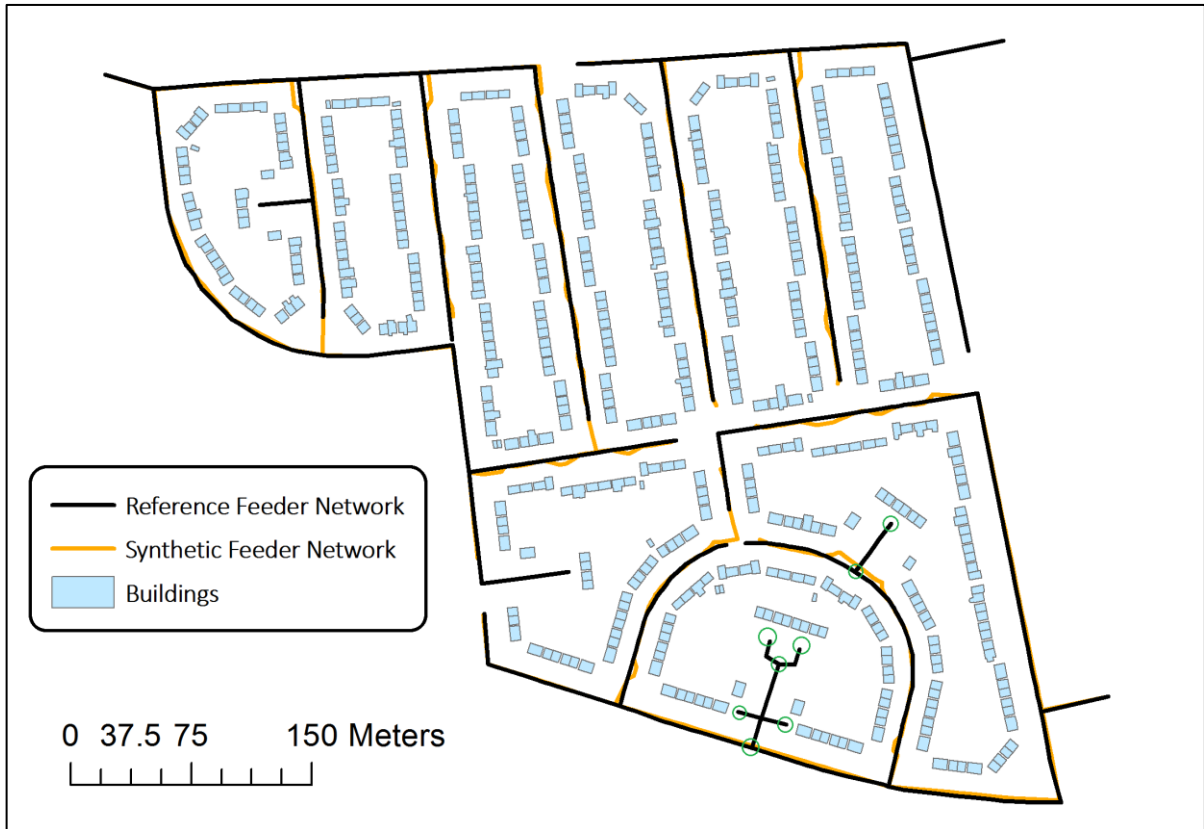


**Figure 6.37.** Synthetic and reference feeder networks for test area 1.



**Figure 6.38.** Synthetic and reference feeder networks for test area 2.





**Figure 6.39.** Synthetic and reference feeder networks for test area 3.

Area	Commission Error	Omission Error	IoU	Length Difference
No.1	4.3 %	6.0 %	91.7 %	8.9 %
No.2	5.2 %	4.6 %	93.6 %	4.3 %
No.2	2.7 %	3.3 %	96.4 %	4.7 %

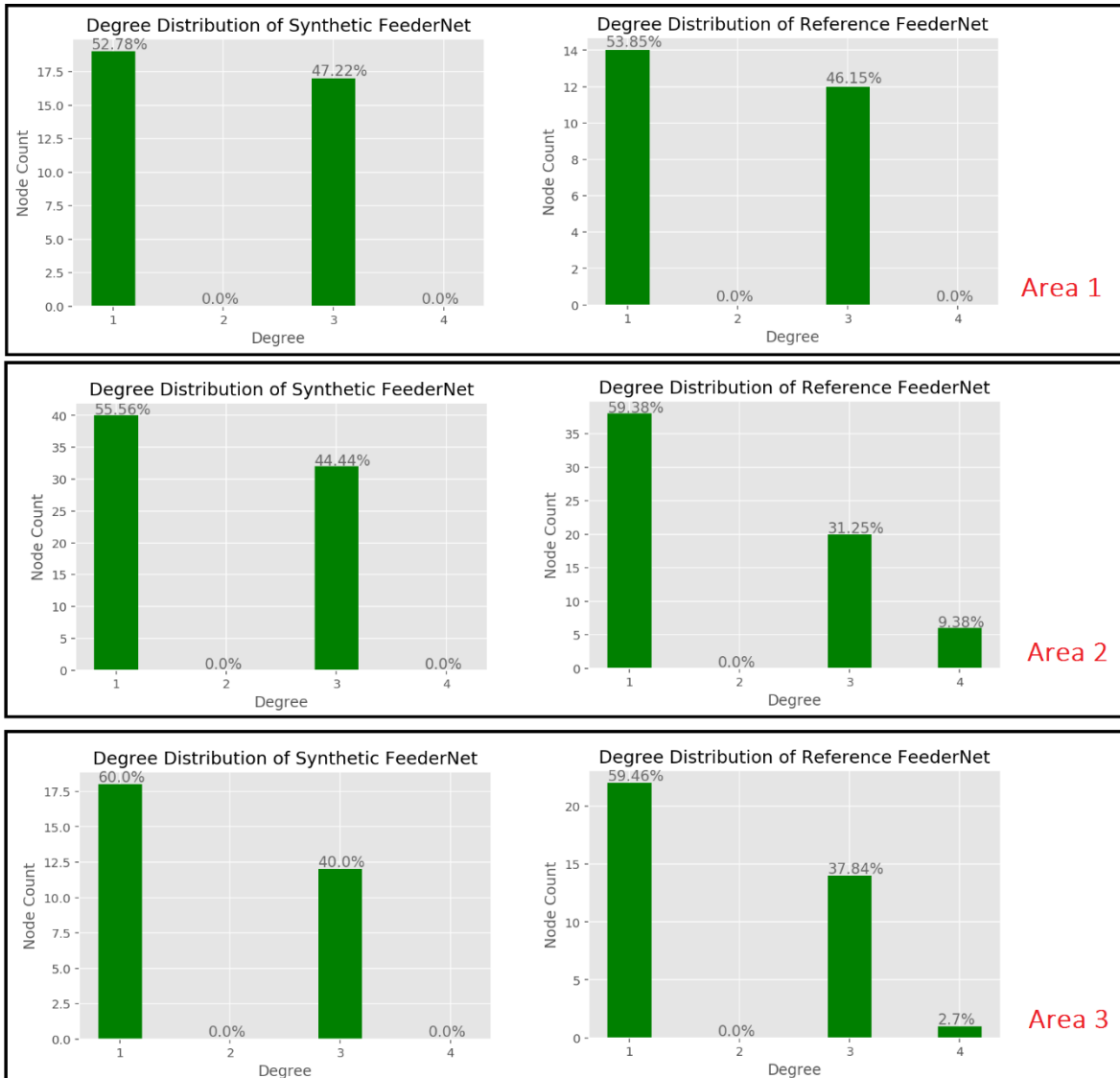
**Table 6.9.** Spatial comparison on the reference and synthetic feeder networks for three areas.

Network Size (Node Count)	Reference Feeder Network	Synthetic Feeder Network
Area 1	26	36
Area 2	64	72
Area 3	37	30

**Table 6.10.** Network size of reference and synthetic feeder network in three areas.

First of all, spatial accuracy maintains high for feeder networks in all of the three areas, due to high spatial accuracy of road networks. More interestingly, feeder-network size differences in

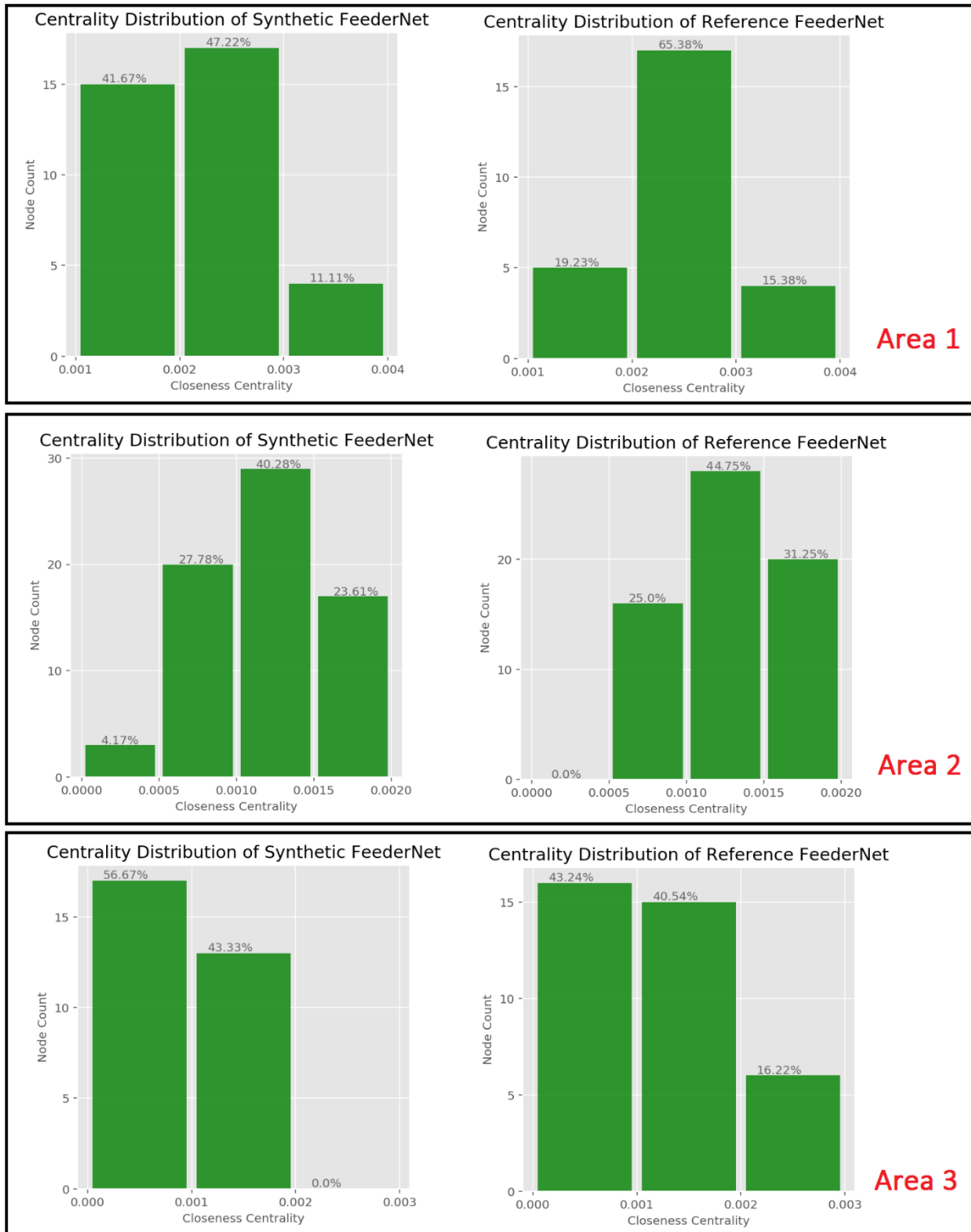
all the three areas, are *smaller* than road network size differences (table 6.8). This is because the feeders are actually only *a part* of the road network (i.e. if there is something wrong with the input road network layer, it does not necessarily affect the infrastructure network layout generated based on it). That is why degree distribution also has a high level of fitness.



**Figure 6.40.** Degree distributions of the reference and synthetic feeder networks for three areas.

Figure 6.41 indicates that there is small discrepancy in centrality distribution especially in area 1, and area 3. The reason is the same here (i.e. road-network size difference), as discussed around figure 6.30. In area 1 (figure 6.37), synthetic feeder network follows the layout of synthetic road network, and thus has more such degree-one and degree-three nodes

(indicated by green circles), which has small centrality values (0.001 – 0.002). In area 3 (figure 6.39), the reference feeder network has more such degree-one and degree-three nodes (indicated by green circles), and they contribute to a slightly different centrality distribution.



**Figure 6.41.** Closeness centrality distribution of the reference and synthetic feeder networks for three areas.

Finally, building-substation dependency is compared between reference and synthetic electricity distribution networks, and result is shown in table 6.11.

Area	No. Type 1 Buildings	No. Type 2 Buildings
No.1	N/A	N/A
No.2	680 (98%)	16 (2%)
No.2	537 (99%)	4 (1%)

**Table 6.11.** Comparison result on building-substation dependency.

In table 6.11, the type 1 and type 2 buildings, are defined in table 6.4. For area 1, there is one substation, and therefore it does not make much sense to measure numbers of type 1 and type 2 buildings. From table 6.9 and 6.10, it is found that the layout of feeders between synthetic and reference networks highly match with each other. More importantly, even relying on the synthetic road network data, the generic heuristic algorithm (developed in Chapter 4) still achieved high accuracy in connecting the buildings to the *correct* substation (compared with the electricity networks generated based on ITN network).

The comparison results on both synthetic road networks and electricity distribution networks, indicates that the road network generation algorithm is generalized well, and has good performance in other areas (other than the area where it is developed and tuned). Using such synthetic road network layout, it is possible to generate plausible infrastructure network layout, that has relatively high spatial and topology accuracy.

## 6.8 Conclusion

Road network layout is a necessary input for algorithm developed in Chapter 4, to generate layout of infrastructure networks (such as electricity distribution networks). However, in new developing sites, road network layout is not always available, and the only information can be the layout of buildings.

Traditional approaches for automatic generation of road network layout does not consider building layout as the input. Therefore, in this chapter, a novel road network generation algorithm developed to solve this problem. It relies on building layout, entry points and a pre-given boundary as the input data. The algorithm is based on an MST partitioning algorithm, which first generates building clusters, and then generates road segments that surround each building cluster.

This algorithm is developed and tuned using data from a small case study area in Newcastle upon Tyne, but it is generalized well when generating road network for other testing areas. That shows the algorithm can be used as general solution for generating geospatial layout of road network. One limitation of the algorithm though, is that it assumes an exterior ring should exist on the synthetic road network, and it cannot be generated by the algorithm (instead it should be explicitly given as a boundary). Despite this limitation, this algorithm is considered as a generic, scalable, and transferable approach to generate layout for road network, and can be applied together the algorithm discussed in Chapter 4 and 5 for infrastructure network inference.

## Chapter 7 – Database Performance Benchmarking Tests

### 7.1 Introduction

Urban infrastructure network data often have complex topology, attribute and geometry (Barr et al., 2016). An efficient data platform is essential for managing such complex network data (Wang, et al., 2015). In many countries, individual operators in specific infrastructure sectors (Woodhouse, 2014) and several large research initiatives (Barr et al., 2016), have realised the importance of developing data and information management platforms for better infrastructure network planning and decision support.

At its core, such platforms require appropriate database systems that can handle the wide range of disparate data and relationships required for infrastructure network modelling and analysis (Barr, et al., 2013; Haider, 2013). Traditionally a spatial relational approach is used, such as the Oracle Spatial Network Extension (British Telecom, 2012; Fikejz et al., 2016) or PostGIS database (Barr, et al., 2013; Zhang, et al., 2012).

The spatial relational approach relies on relational models and applies tables of predefined schema to store large amount of data (Tang, 2016), and it is naturally strong in resolving relational query (e.g. return all the nodes with type ‘building’) or spatial query (e.g. return all the nodes that are spatially within a given footprint) (Agarwal, et al., 2017). However, when storing the large and complex network data (e.g. fine scale urban infrastructure network discussed in this PhD), this approach shows potential performance bottleneck in analysing network topology (Robson, et al., 2018), as this task often transforms to an expensive join operation among multiple tables (Vicknair, et al., 2010).

Recently, NoSQL graph database, based on graph data model, has been proposed as a generic approach for more efficient storage and retrieval of complex network data, and it has been applied in different fields, such as bioinformatics (Have, et al, 2013), social network (Fan, 2012), and recommendation system (Bagci, et al., 2016). However, very little attention has

been made in applying graph database in the management of geospatial infrastructure network data. The major reason is that, no database performance benchmarking tests have been done, to justify performance boost in applying graph database over the traditional approach, when dealing with geospatial infrastructure network data. The purpose of this chapter is to fill in this research gap.

The objective of a performance benchmarking test, is to evaluate performance of a database system against a reference one (TPC-C benchmark, 1992). The performance, normally refers to the execution time of a database to resolve a given query (Tang, 2016; Ferro, 2018; Ray, et al., 2011). Database performance is often evaluated on tests of different complexities (Ray, et al., 2011), which is related to size of data (e.g. number of nodes for network data) to be processed, and the difficulty of the query (e.g. *return all nodes* compared with *return all nodes with specific attribute value*) that needs to be resolved (Vicknair, et al., 2010). Tests of different complexities help understand the strength and weakness of each database, and to evaluate what database to use in which situation (Jung, et al, 2015).

Writing and reading data are the most basic queries that are used in almost any database benchmarking test for any database (McColl, et al., 2014). For spatial database, the additional test queries can be spatial operations (e.g. intersection calculation, within calculation, distance calculation) (Paton, et al., 2000). For graph database, additional test queries can be network search queries (e.g. neighbour search, shortest path search) (ArangoDB, 2018). These common queries are used for general performance evaluation for spatial and graph databases (Vicknair, et al., 2010; Mpinda, et al., 2015). However, as pointed out by Papadias et al (2003), if the database is used for a specific application or is using specific data (e.g. in our case, fine scale geospatial infrastructure network data), then test query must be carefully designed to simulate the operations that can actually occur in real applications.

The aim of this chapter is to develop performance benchmarking tests, to evaluate the performance of graph database against the traditional approach (spatial relational database), in processing geospatial infrastructure network data. Section 7.2 discusses the database

approaches used in the tests. Section 7.3 gives an overview of the tests to be done. Section 7.4, 7.5, and 7.6 are the tests and result interpretations. Section 7.7 concludes this chapter.

## 7.2 Database Approaches for Tests

Three database approaches are chosen for the performance benchmarking tests. They are the ITRC interdependency network schema, PgRouting, and a hybrid database based on a PostGIS and Neo4j database.

The ITRC interdependency network schema is a database schema based on PostGIS, developed for the NISMOD-DB project (Barr, et al., 2013). It is proved to be an efficient and reliable approach in the management of national scale interdependent infrastructure networks in the United Kingdom. Therefore, this approach is considered to be a good benchmark, when processing fine scale urban infrastructure network data.

The PgRouting approach is actually a PostGIS database with PgRouting extension (PgRouting, 2018). This extension gives PostGIS database a routing functionality (e.g. resolving shortest path algorithm) when storing network data. Due to routing functionality and PostGIS's original strength in querying spatial data, the PgRouting has been widely considered as an economic (free) and efficient solution for spatial network routing applications, such as road network routing (Zhang, et al., 2012). Therefore, this approach is considered related to the management of geospatial infrastructure network data, and also chosen here.

The final approach is a hybrid database, which is based on two databases (PostGIS and Neo4j) linked with each. Neo4j is the most popular graph database (DB-Engines Ranking, 2018), which is based on a new data model called *property graph*, and it is suitable for storing and querying large and complex network data (Neo4j, 2018). Therefore, it is considered to be a good solution when performing network search queries on complex infrastructure network



data. However, currently Neo4j does not have good support for on spatial data (more details to be discussed in section 7.2.3), and therefore a hybrid database is used here.

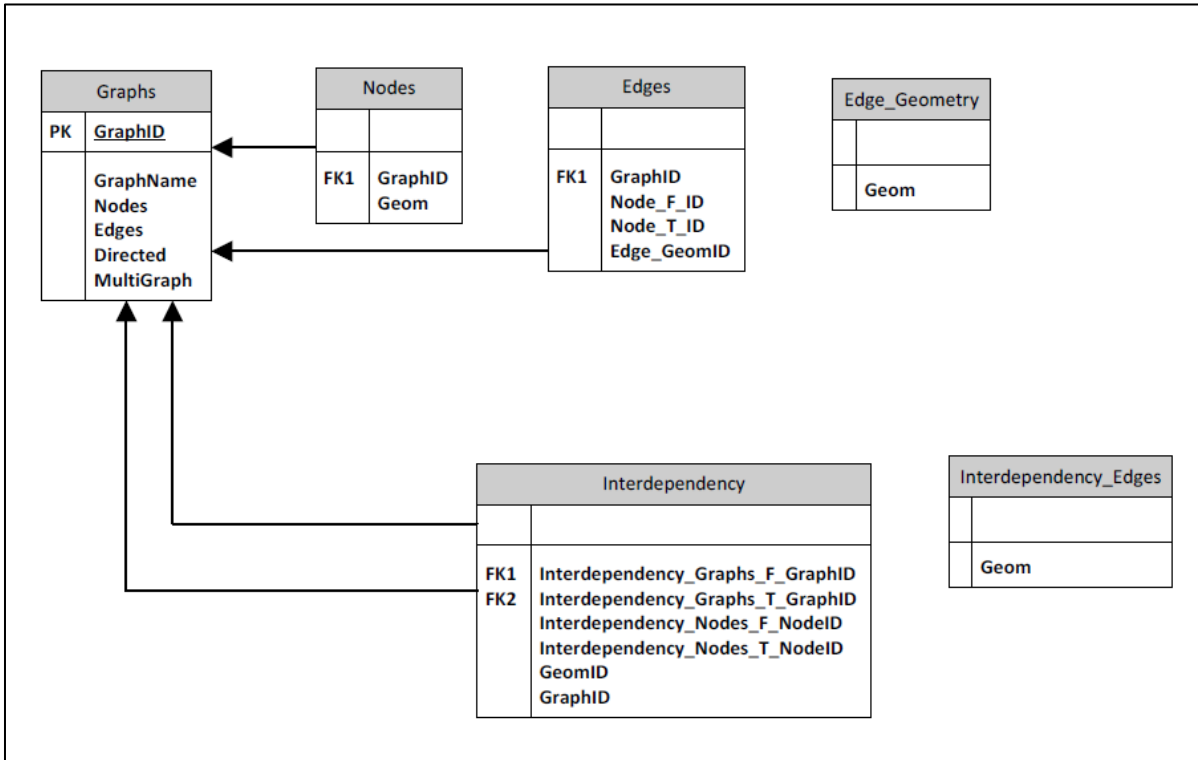
Section 7.2.1, 7.2.2, and 7.2.3 gives more explanation about how each database approach stores data and how to perform general queries (such as writing, reading, or network search).

### ***7.2.1 ITRC Interdependency Network Schema***

The ITRC Interdependency Network Schema (from now will be termed ITRC schema) is shown in figure 7.1. In this schema, the **Graphs** table (the meta-data table) is used to store the name of each individual network instance.

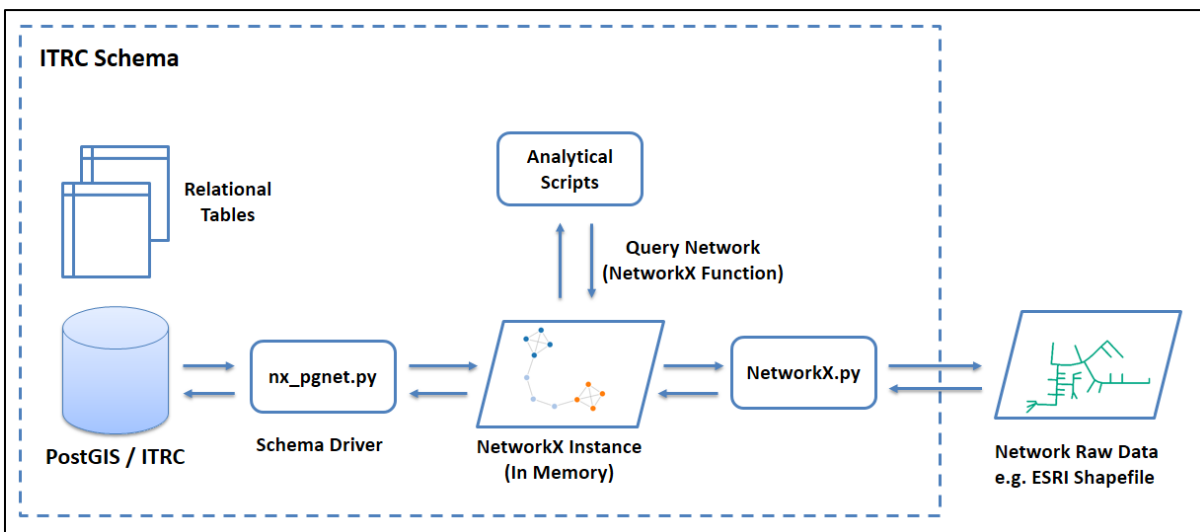
For each individual network instance: a **Nodes** table is used to store the geometries and attributes of nodes, an **Edges** table is used to store the edges (topological connectivity of nodes) and their attributes, and an **Edge\_Geometry** table is used to store the geometries of edges. Within a network instance, each node or edge is indexed using a **Node\_ID** or **Edge\_ID**. The topology (which edge is connected to which two nodes) is exactly represented by storing the **Node\_ID** into the **Edges** table.

There is also an **Interdependencies** table, which is used to store the interdependency from a node in a network instance to a node in another network instance. An **interdependency\_Edges** is the table to store the geometry for such interdependency.



**Figure 7.1.** ITRC schema.

With such schema, it is possible to store geometries and attributes (e.g. node type, edge type) and topology in a single PostGIS database. Moreover, additional database APIs and libraries exist for writing, reading data to/from the database, as well as querying the data. A generic pipe line for ITRC schema is shown in figure 7.2.



**Figure 7.2.** General pipe line for ITRC schema.

In figure 7.2, the Python library NetworkX is the most vital part in the entire pipe line. This library is used for creating and manipulating complex network data in memory (NetworkX, 2018). With NetworkX, network raw data (e.g. one ESRI shapefile file for network edges, and another ESRI shapefile file for network nodes) are converted to a NetworkX instance first, and then written to an instance of the ITRC schema, via a schema driver called nx\_pgnet. Likewise, reading data from the database must be done also via nx\_pgnet and NetworkX. To query the data, as long as network topology is involved (e.g. return the neighbours of a given node, or return a shortest path between two nodes), network data must be read into memory as NetworkX instance, and queried via the NetworkX function.

The ITRC schema is proved to be effective when modelling national scale geospatial infrastructure networks (Barr, et al., 2013). However, there is no evidence to show it is still efficient in processing fine scale infrastructure network data that has more complex topology. This database approach will be used as the benchmark to evaluate performance of the other two database approaches.

### **7.2.2 PgRouting**

The PgRouting approach is a PostGIS database with PgRouting extension (PgRouting, 2018). The way it stores network data is almost the same as the ITRC schema. The PgRouting uses one table to store network nodes and another table to store the network edges. With the PgRouting extension, additional routing functions (e.g. resolving shortest path between two nodes) are introduced and can be called as SQL queries. The general pipe line for the PgRouting approach is shown in figure 7.3, with necessary data APIs provided.

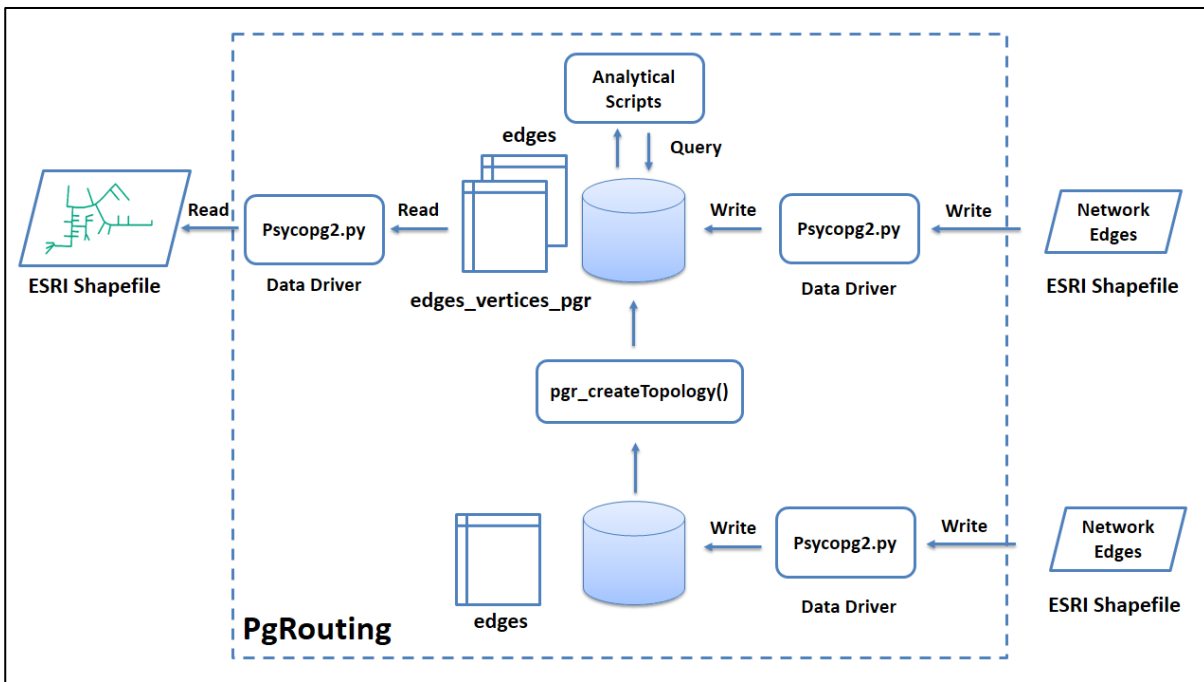
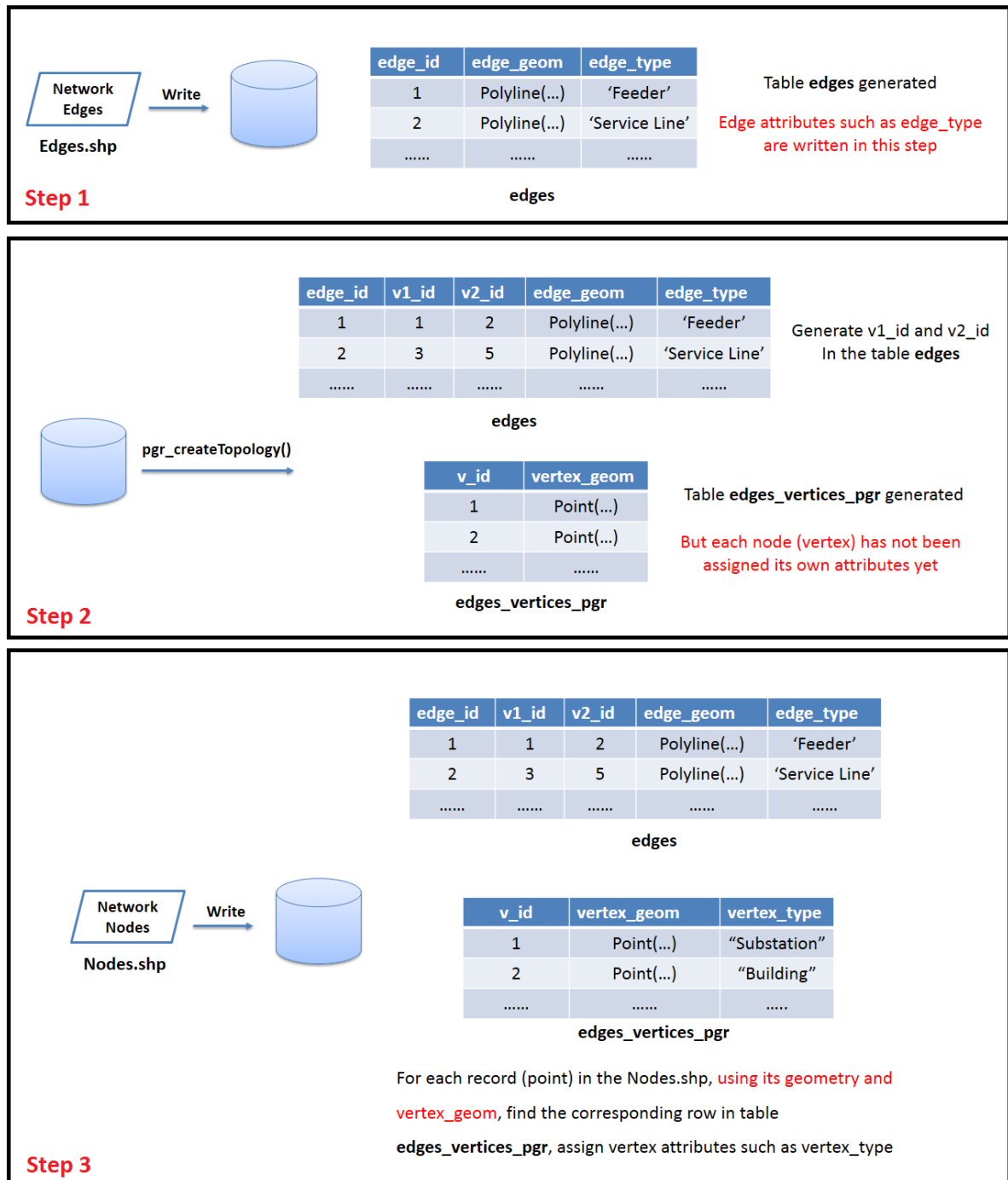


Figure 7.3. General pipe line for PgRouting approach.

Figure 7.3 shows that when using PgRouting, the PostGIS database no longer relies on any external library (e.g. NetworkX), except for the data driver Psychopg2 (Psychopg, 2018). However, the writing process is more complicated now, and it is important to know the network edges and nodes are written separately into database.

The major reason is that PgRouting must use a very special function (called **pgr\_createTopology**) to construct network topology (PgRouting, 2018), which only accepts network edges as input. That actually creates barrier in writing nodes (especially the node attributes) into PgRouting. Figure 7.4 illustrates what exactly happens when writing network data into PgRouting.



**Figure 7.4.** The actual detailed flow to write network into PgRouting, supposing writing electricity distribution network.

First (Step 1), reading **Edges.shp** as input, a table called **edges** is generated. The **edges** table contains geometry of each edge, and contain edge attributes as well. PgRouting automatically assign an **edge\_id** for each edge. Then (Step 2), **pgr\_createTopology** function needs to be called, so that PgRouting can infer network topology based on spatial connectivity of edges. The result is the generation of a table called **edges\_vertices\_pgr**, which stores the geometry

of each vertex (node), and each vertex (node) is assigned a **v\_id** automatically. Note in table **edges**, two new columns **v1\_id** and **v2\_id** are generated, to indicate the topological connectivity between edges and vertices (nodes). However, vertex (node) attributes have not been assigned yet. That is why, finally (Step 3) **Nodes.shp** is read as input. To assign node attributes, spatial matching must be done (for each record in **Nodes.shp**, find the record in **edges\_vertices\_pgr** that has same geometry).

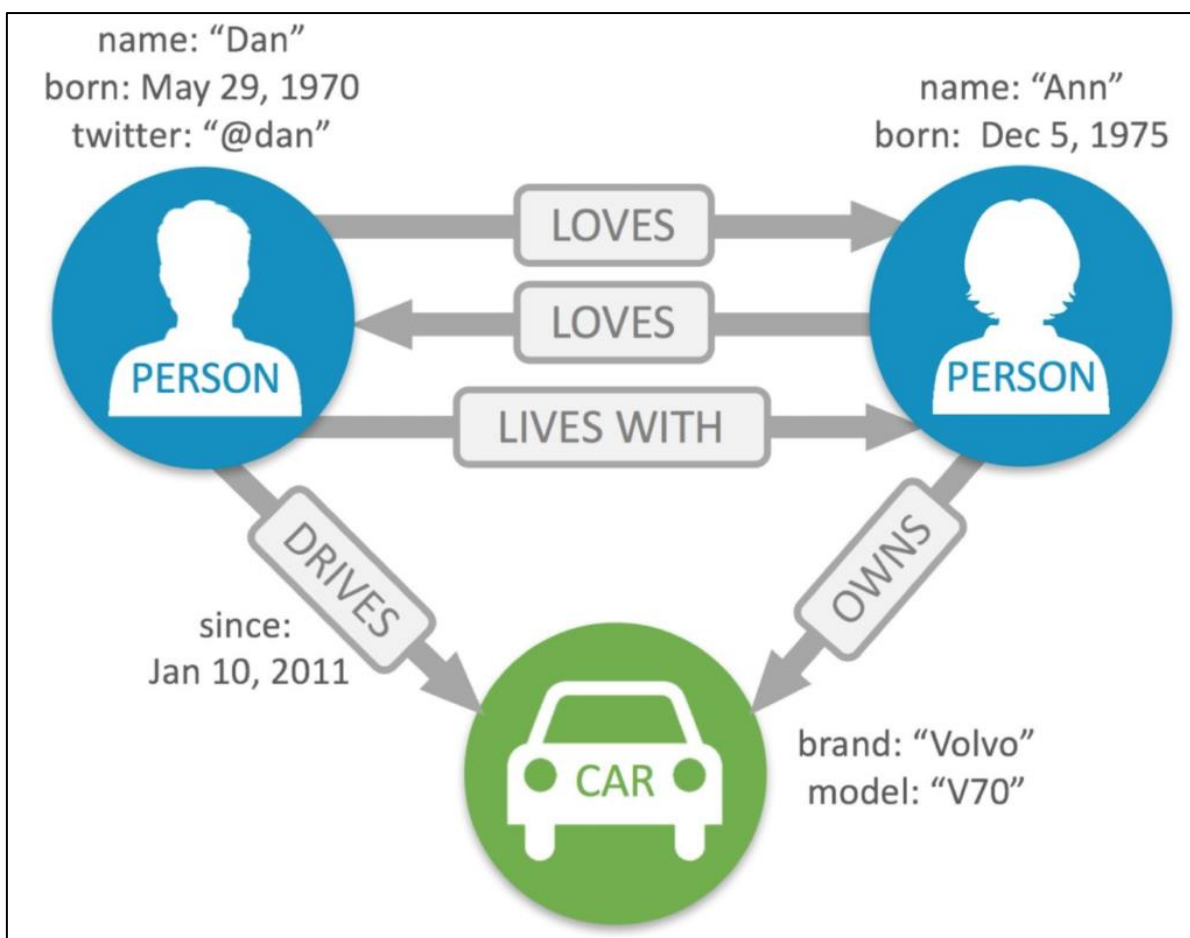
Despite the long pipe line of writing, reading is more efficient for PgRouting, as only Psycopg2 is called to directly retrieve data from PostGIS. More importantly, querying data is easier, compared with ITRC schema. No matter what query needs to be executed (whether it is spatial, attribute, or network query), the query can be directly made to the PostGIS database via SQL (no need to read data into memory and ask NetworkX to perform the query).

### **7.2.3 Hybrid Database**

A hybrid database normally refers to a system consisting of multiple databases, which acts as one single system (Maislos, 2017). A simple hybrid database can be a combination of two databases, for example, a relational database and a NoSQL database (Thant, et al., 2014). The reason to use a hybrid database is often to gain performance improvement, compared with a system of a single database. For example, Robson et al (2018) presented work in developing the NISMOD-DB ++ database, which combines the graph database Neo4j and relational database PostGIS to process geospatial network data. Using this approach, query can be executed via SQL or Cypher (Neo4j's query language) to the PostGIS or Neo4j, in order to achieve better system performance.

The hybrid database approach to be introduced in this sub-section is similar to NISMOD-DB++. It is a combination of a PostGIS and Neo4j database. The reason to choose Neo4j is that it is currently *the most* popular graph database (DB-Engines Ranking, 2018). Neo4j has its own data model called property graph (Neo4j, 2018). A property graph consists of nodes

and relationships that connect nodes. Each node and relationship can have its own property, where each property is a (key: value) pair. An example is shown in figure 7.5.



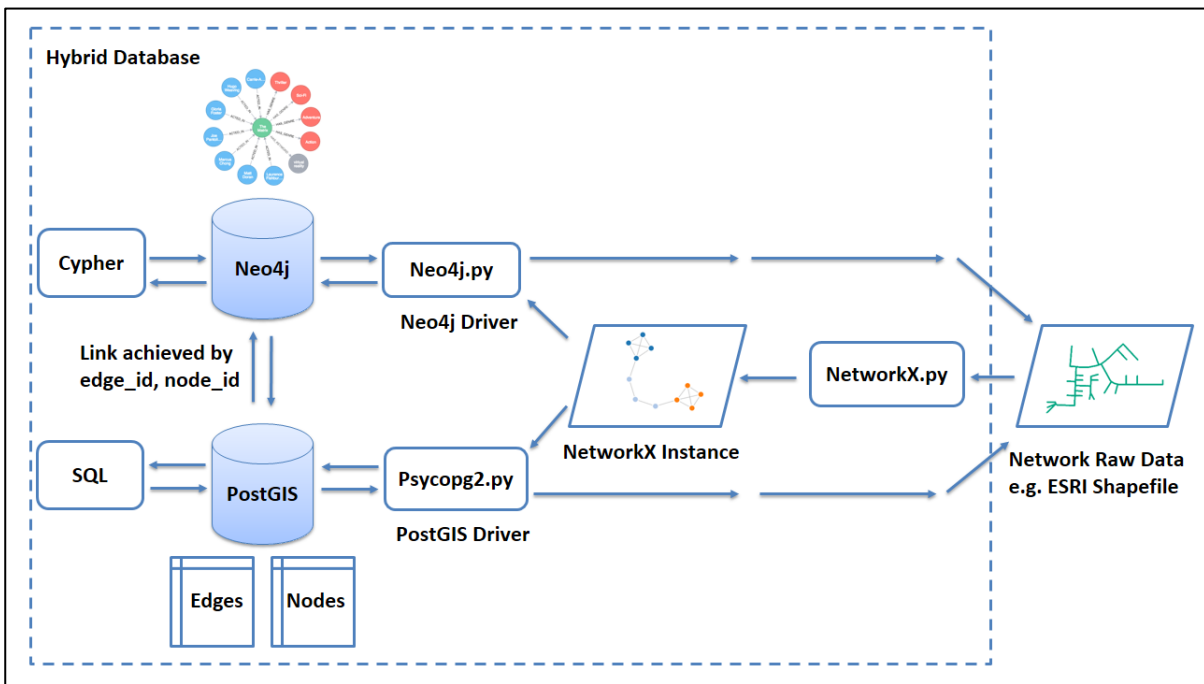
**Figure 7.5.** An example of Neo4j property graph.

Source: <https://www.sitepoint.com/introducing-the-neo4j-symfony-bundle/>

The Neo4j property graph model makes it easy to store network data with attributes in Neo4j. Moreover, Neo4j has its special query language Cypher which allows for attribute or network query. However, its capability is somewhat limited in spatial operation. Neo4j does provide an extension *neo4j-spatial* for extra functionality in storing and querying spatial data. However, currently (late 2018), there are still two major disadvantages of *neo4j-spatial* in modelling geospatial infrastructure network data (Neo4j-Spatial, 2018). First, while geometry of the network nodes can be stored and indexed, the geometry of network edges cannot. That means it is impossible to perform spatial query on network edges. Secondly, supported spatial query on network node is too simple. For example, if performing a *within* operation using a given

footprint as input (e.g. return all the nodes within the given spatial footprint), that footprint (polygon) must be a circle, it cannot be a more complex irregular polygon.

Given the above, *neo4j-spatial* is considered to be inappropriate for modelling fine scale geospatial infrastructure networks. Thus, a hybrid database is developed that employs a combination of Neo4j and PostGIS (figure 7.6).



**Figure 7.6.** General pipe line for the hybrid database approach.

In the hybrid database, network data are *separately stored* in PostGIS and Neo4j. PostGIS only stores the *geometry* of nodes (e.g. point) and edges (e.g. polygon), using two tables (**nodes** and **edges**). Neo4j only stores network *topology* and *attribute* of nodes and edges (as properties in the property graph model). NetworkX is still a necessary external library in the writing process, but no longer needed in reading. Moreover, PostGIS and Neo4j can be queried via SQL and Cypher, depending on the actual workload. For example, if a spatial calculation needs to be resolved (e.g. return all the edges within a given spatial footprint), SQL is called on PostGIS; if a network search needs to be resolved (e.g. return all the nodes connecting a given node), Cypher is called on Neo4j.



Due to the separate data storage, a link must be made between PostGIS and Neo4j, so that we know the corresponding geometry for the node and edge (relationship) in the Neo4j property graph. The link is achieved is via assigning each node a unique **node\_id** and each edge a unique **edge\_id**. The **node\_id** and **edge\_id** are stored both in Neo4j and PostGIS.

For example, figure 7.7 shows how to use **node\_id** and **edge\_id** to link data (in this example, electricity network data) stored in the hybrid database. The node in red rectangle (in property graph) has its corresponding geometry in red rectangle (in the table **nodes**). The edge in orange rectangle (in property graph) has its corresponding geometry in orange rectangle (in the table **edges**).

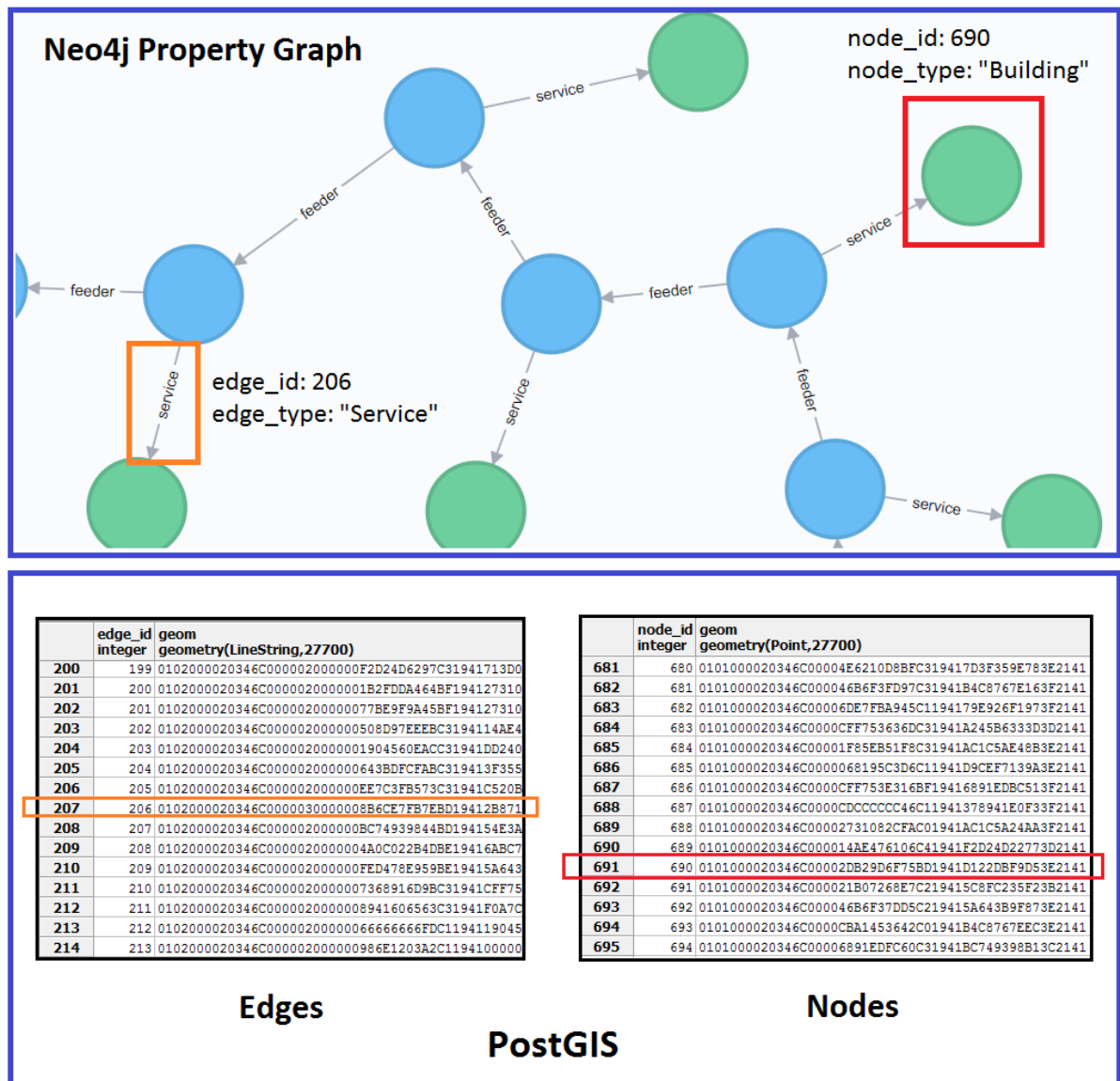
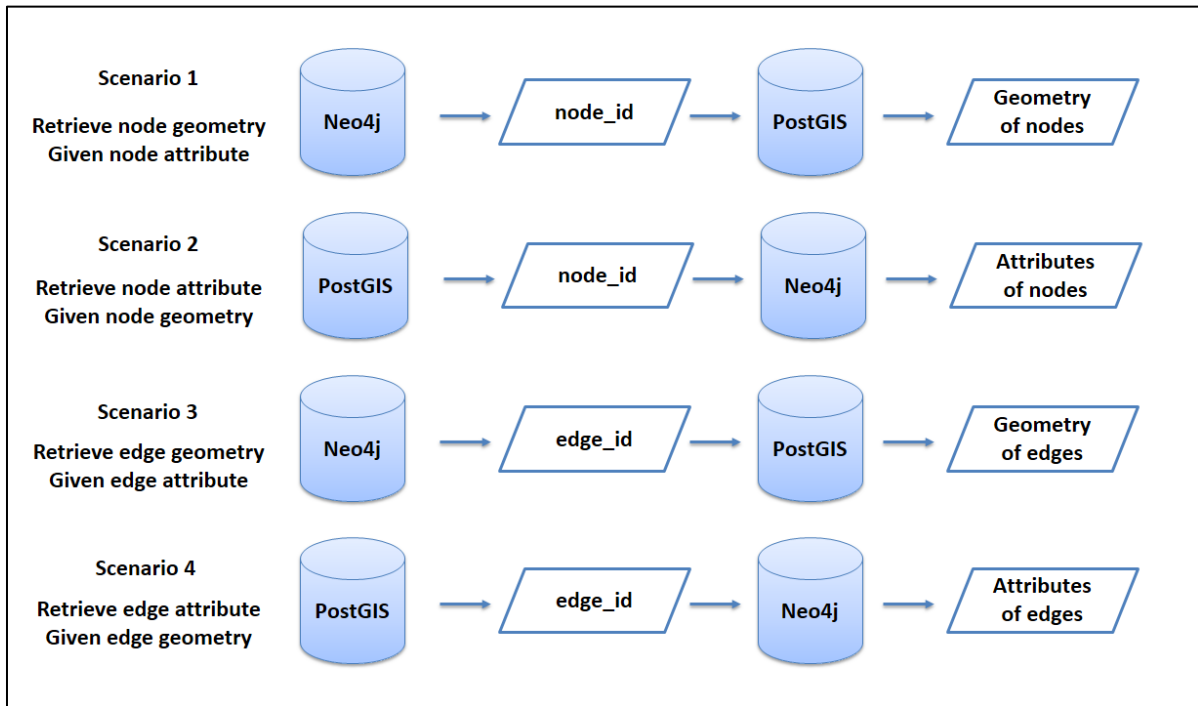


Figure 7.7. Linking PostGIS and Neo4j using **node\_id** and **edge\_id**.

This data reference approach is important, when the hybrid database needs to execute a query that needs to visit both Neo4j and PostGIS to retrieve the final result. Four common scenarios are shown in figure 7.8, depending on which database is visited first and whether the query is related to node or edge.



**Figure 7.8.** Four common and simple scenarios of retrieving data using both databases.

### 7.3 Performance Benchmarking Tests Overview

The performance benchmarking tests developed in this chapter measure the query execution time (Tang, 2016; Ferro, 2018; Ray, et al., 2011), to evaluate performance of the three database approaches mentioned earlier. Three major tests (with increasing complexity) are designed, and performance of ITRC schema is regarded as the benchmark (i.e. 100%).

The first test has the lowest complexity, which relies on infrastructure network data of different sizes (i.e. number of nodes). The actual workloads are simple database operations, including writing, reading, and network shortest path query. This test aims to generally evaluate database performance when processing different sized infrastructure network data.

The second test is more difficult than the first test, because it relies on two large infrastructure network data (the road network and electricity distribution network in Newcastle upon Tyne). The actual workloads are more difficult, which will be complex queries where there is spatial calculation. This aims to evaluate database performance when spatial query is involved in analysing infrastructure network at city scale.

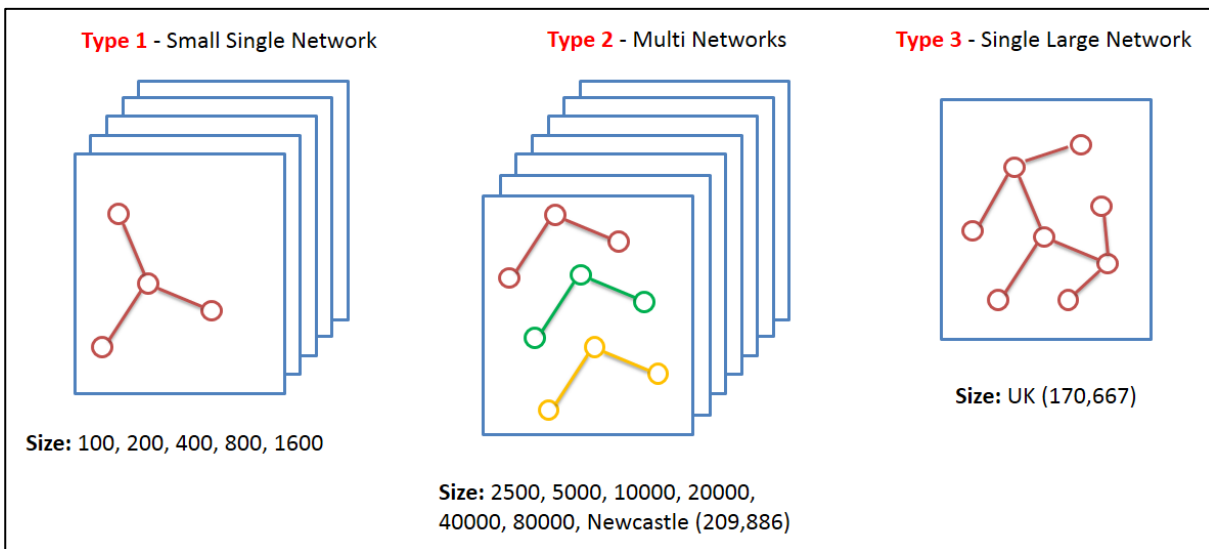
The third test is the hardest one, as it uses a massive network data set (the electricity distribution network in London). Workloads include simple operations (writing and reading) as well as complex ones (spatial calculation and network search). This test aims to comprehensively evaluate database performance when performing complex queries on massive network data.

The benchmarking tests were run on a desktop workstation, with 2 core CPUs (Intel(R) Xeon(R) Gold 6134 CPU @ 3.20 GHz), and 512 GB memory. The versions of database software are: PostgreSQL 10.3 / PostGIS 9.4, PgRouting 2.2, and Neo4j 3.1.3. The versions for the external libraries and data drives are: NetworkX 1.11, nx\_pgnnet 0.9, Psycopg2 2.7.7, Neo4j Python Driver 1.5.1.

#### **7.4 Performance Test on Different Sized Network Data**

This test evaluates databases performance when processing different sized network data (from network of about 100 nodes to the one of about 200,000 nodes). It is designed to be a test of lowest complexity, and therefore only simple operations are considered (writing data, reading data, and shortest path test). Details of the data, and test results are introduced below.

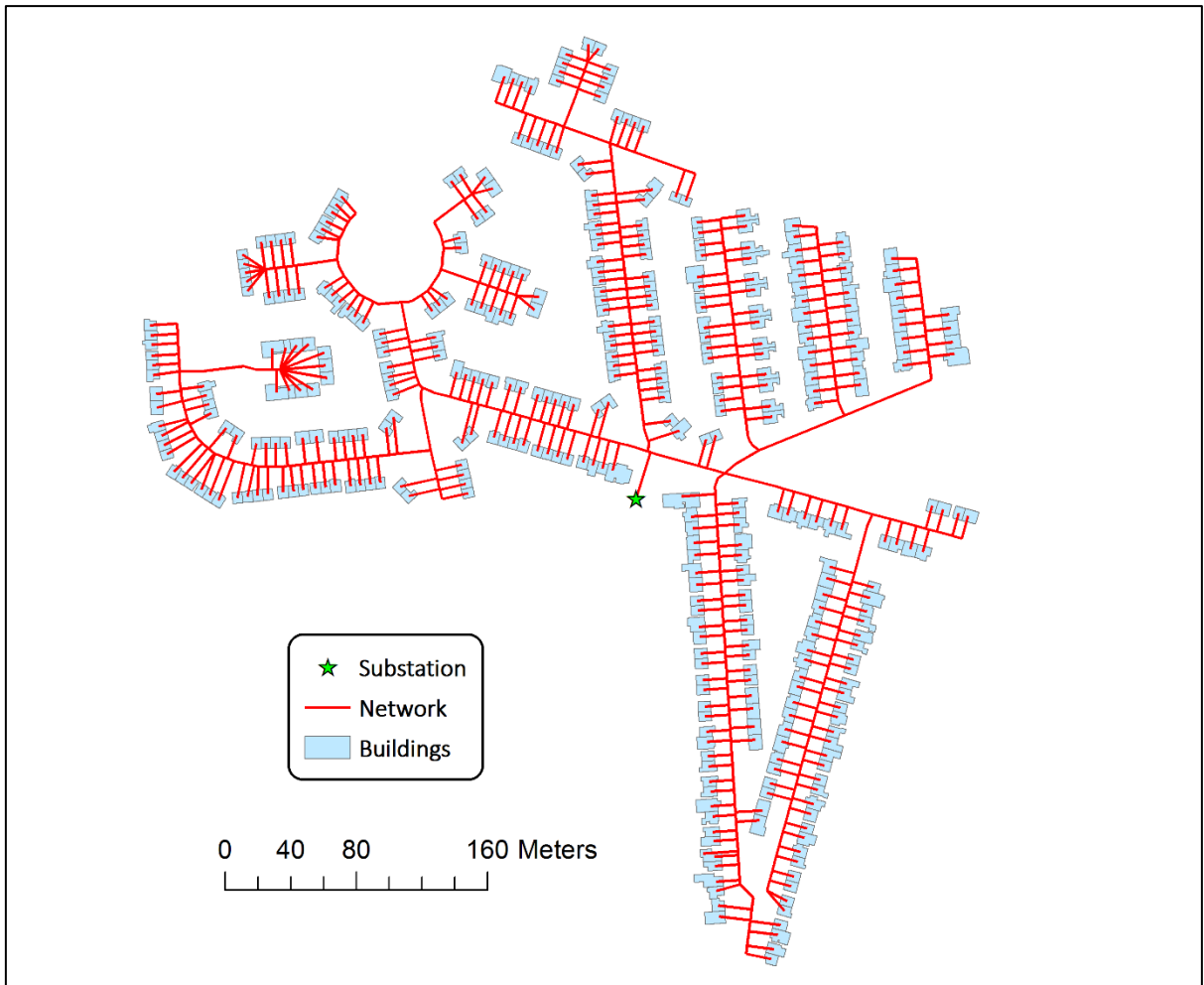
### 7.4.1 Network Data



**Figure 7.9.** Three types of data used in this test.

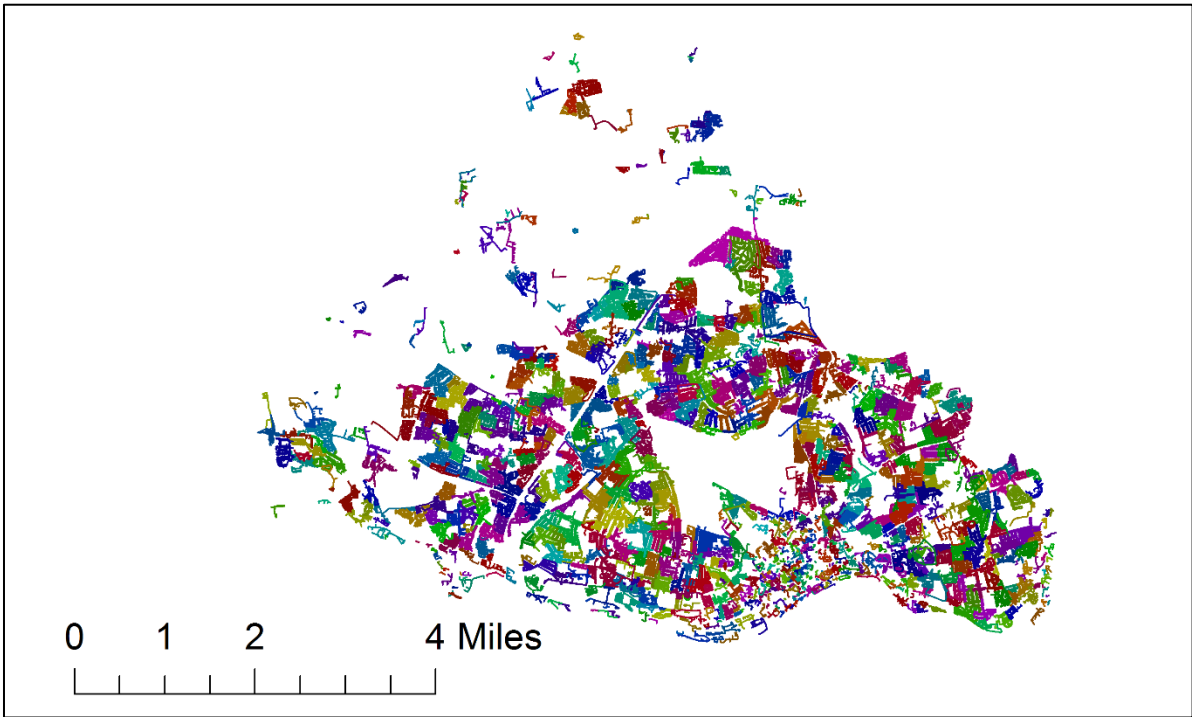
There are 13 different network data sets used in the test, and they can be classified into three types (figure 7.9). Complete datasets are explained in Appendix H.

For the type 1, there are 5 data sets. Each is a single instance of electricity distribution network in Newcastle upon Tyne. Their sizes (number of nodes) are about 100, 200, 400, 800, and 1600 respectively. Each network instance has one asset (electricity substation) and several building nodes. For example, figure 7.10 shows the network instance of size 800.



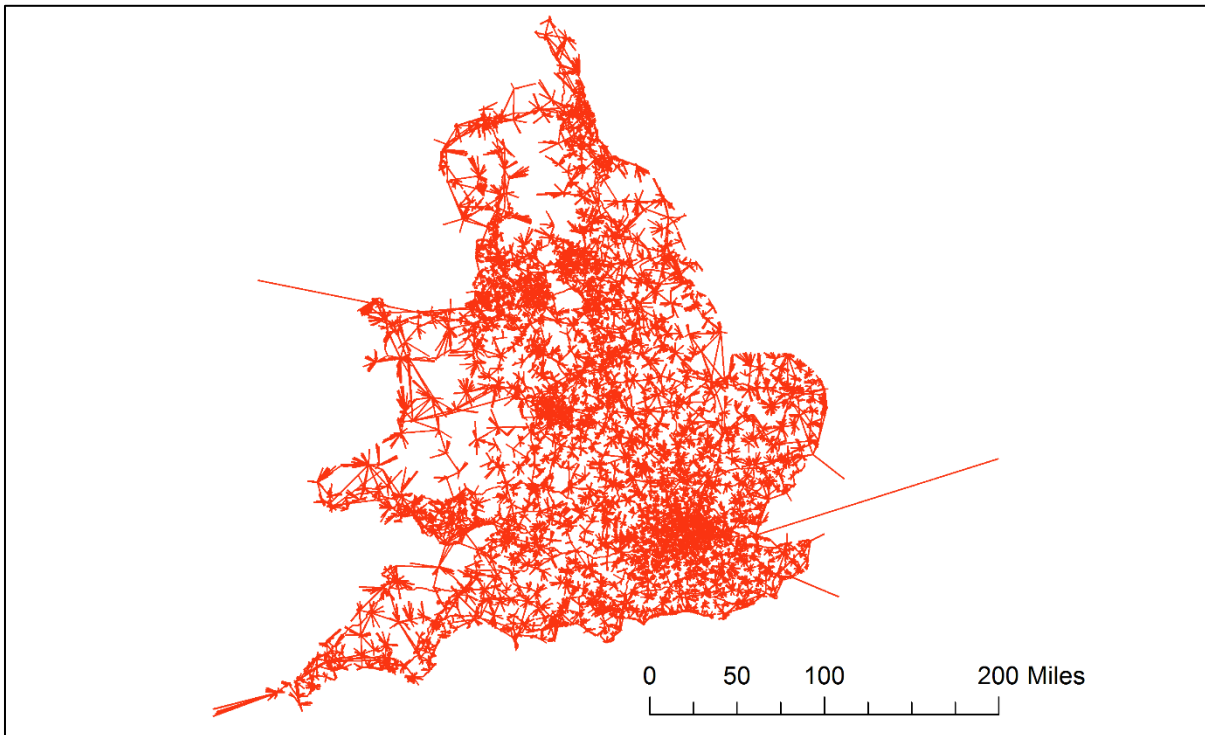
**Figure 7.10.** The network instance of size 800, in the **type 1** network data.

For the type 2, there are 7 data sets. Each one contains multiple instances of electricity distribution networks in Newcastle upon Tyne, with their sizes being 2500, 5000, 10000, 20000, 40000, 80000, and ‘Newcastle’. The size ‘Newcastle’ corresponds to the entire city scale electricity distribution network in Newcastle upon Tyne (generated in Chapter 4), which contains totally 209,886 nodes (figure 7.11).



**Figure 7.11.** The **type 2** network data, with size being ‘Newcastle’. Each colour in the figure refers to a single network instance.

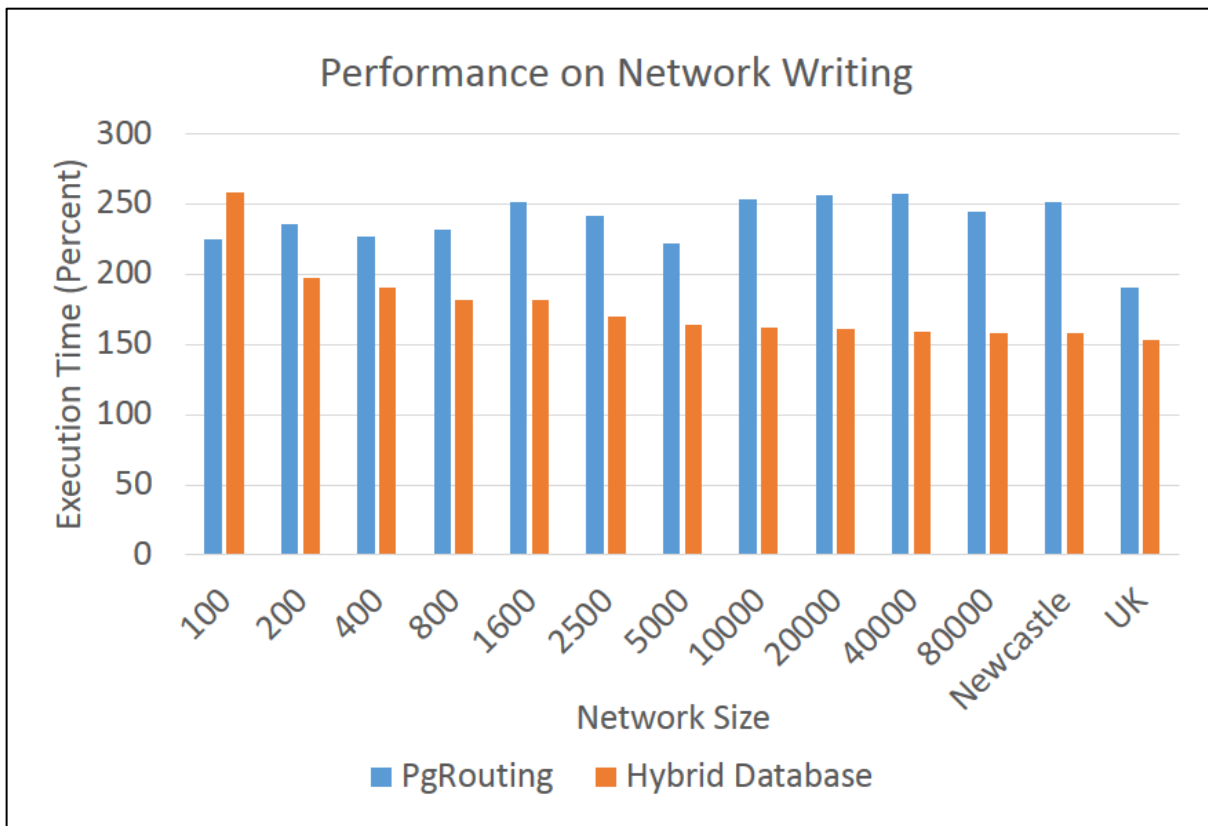
For the type 3, there is only one network data set (size ‘UK’), which is a single large spatial network instance comprising of the England and Wales national electricity transmission-distribution network (figure 7.12), containing 170,667 nodes and 173,039 edges.



**Figure 7.12.** The **type 3** network data, with size being ‘UK’.

### 7.4.2 Writing Test

The writing test evaluates the performance of writing network data (from ESRI shapefile format) into the database. The actual execution time for the writing test is shown in table I1, Appendix I. It is found that the execution time of any database approach is almost proportional to the network size. ITRC schema is the fastest one, regardless of network size, from 1.2 seconds (to write network of size 100) to 1936 seconds (to write network of size ‘Newcastle’). The PgRouting approach is always slower (than the ITRC schema), which costs 3.1 seconds (to write network of size 100) and 4859 seconds (to write network of size ‘Newcastle’). The hybrid database is also slower than the ITRC approach, but is faster than PgRouting, especially in writing large network data (costs 2884 seconds to write network of size ‘Newcastle’). When comparing the performance of PgRouting and hybrid database against ITRC schema (benchmark), figure 7.13 shows the relative difference.



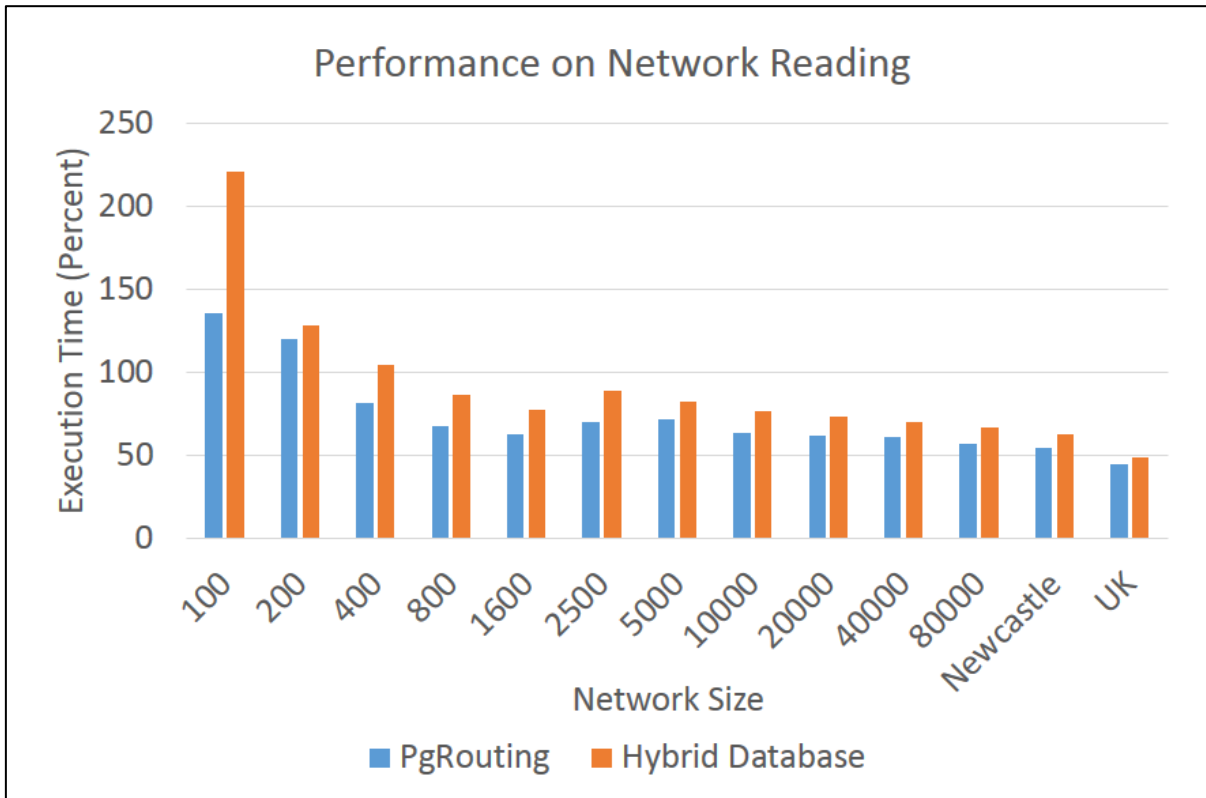
**Figure 7.13.** Performance comparison of writing different sized network data.

Figure 7.13 shows the percentage difference of execution time of PgRouting and hybrid database against the ITRC schema (100%). It is interesting to see that PgRouting is always about 2 – 2.5 times slower than ITRC schema, even if the PgRouting approach does not rely on NetworkX in writing data. The major reason is that, the pipe line to write data into PgRouting is very expensive (explained in section 7.2.2, figure 7.4). PgRouting needs to write edges and nodes separately into PostGIS, and assigning node attributes can be time consuming. For hybrid database, it is about 2.5 times slower than benchmark, when writing extremely small data (e.g. size is 100), but that ratio decreases as network size increases, and finally stays around 150%. The reason is that, when writing very small network, the time for database driver to connect hybrid database can be longer than the time to do the actual writing. Still, hybrid database is 1.5 times slower when writing large network. The major reason is that, writing needs to be done to two databases, and Neo4j driver can be slower than the database driver of ITRC schema.

### **7.4.3 Reading Test**

The reading test evaluates performance of database to read network data (from the database) to a GIS file (e.g. ESRI shapefile format). The actual execution time for the writing test is shown in table I2, Appendix I. It is found that, like writing network data, when reading network data, execution time of all databases are still almost proportional to the network size. ITRC schema needs to cost 1.4 seconds (to read network of size 100) and 3012 seconds (to read network of size ‘Newcastle’). The corresponding execution time of PgRouting and hybrid database, are 1.9 seconds, 1772 seconds, and 3.1 seconds, 1012 seconds, respectively. A percentage performance comparison is shown in figure 7.14.





**Figure 7.14.** Performance comparison of reading different size network.

Figure 7.14 clearly shows that the benchmark (ITRC schema) is no longer the most effective approach when reading data. In fact, it is always slower than the other two. When processing large network (size > 10000), PgRouting can be almost 2 times faster than the benchmark. The hybrid database is slightly slower than PgRouting.

For ITRC schema, the major reason for its poor reading performance, is that it must read data into NetworkX first, then output the GIS files. However, NetworkX is not needed for PgRouting and hybrid database, and they can directly read data from the database. Hybrid database is comparatively slightly slower, and that is still because it needs to read data from two databases, instead of one.

#### 7.4.4 Shortest path test

When evaluating database performance in handling network data, the weighted shortest path query is always considered to be the most important one (ArangoDB, 2018, Tang, 2016) and therefore it is also undertaken here. To be clear, the shortest path query on the 13 data sets are not exactly the same, and it is shown in table 7.1.

Data Set	Shortest Path Query
Type 1 and 2	Resolve Dijkstra shortest path for each substation node to each building node it serves.
Type 3	Given 50 nodal pairs (node_id, node_id), resolve Dijkstra shortest path between nodes in each nodal pair.

**Table 7.1.** Shortest path query to be executed.

For type 1 and 2 data set, the query is resolving Dijkstra shortest path from each substation node to each building node it serves, where the weight is the edge length. The reason to consider such query is that, as mentioned in this thesis, the connection between infrastructure asset and buildings it services is essential. However, for type 3 data (UK transmission network data), same shortest path query cannot be done, because there are no building nodes in it, but only substations of different levels. Therefore, a *special* shortest path query is designed for type 3 data, which is given several nodal pairs, resolving Dijkstra shortest path between nodes in each pair. The nodal pairs are manually picked up, and topological distance (how many nodes between them) in a nodal pair is at least 20, and this helps us evaluate how efficiently database can resolve shortest path between *relatively distant nodes* in a single large network instance.

For type 1 and 2 data set, one thing that must be made clear is that, each electricity distribution network instance has a unique **net\_id** (from input ESRI shapefile), and it is encoded as an attribute on each node and edge, and is stored in all of the three databases approaches (ITRC schema, PgRouting, and hybrid database). Since the shortest path query is more difficult than reading and writing data, the actual pipe lines for different database

approaches are shown in figure 7.15 and 7.16.

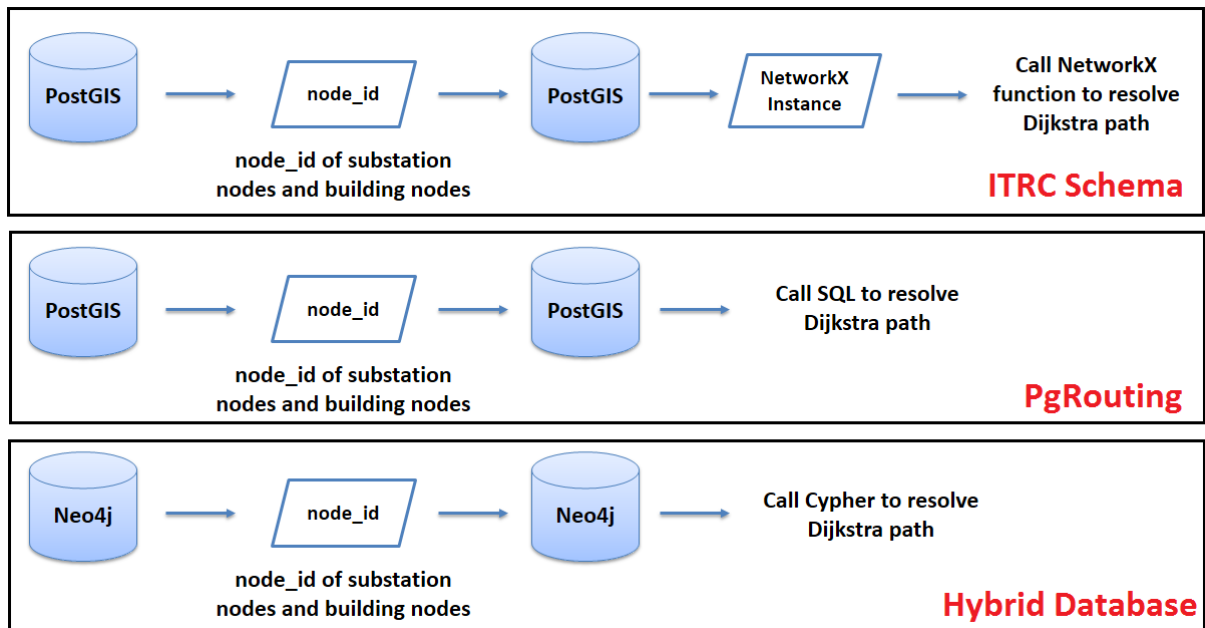


Figure 7.15. Pipe lines for shortest path query on **type 1** and **type 2** network data.

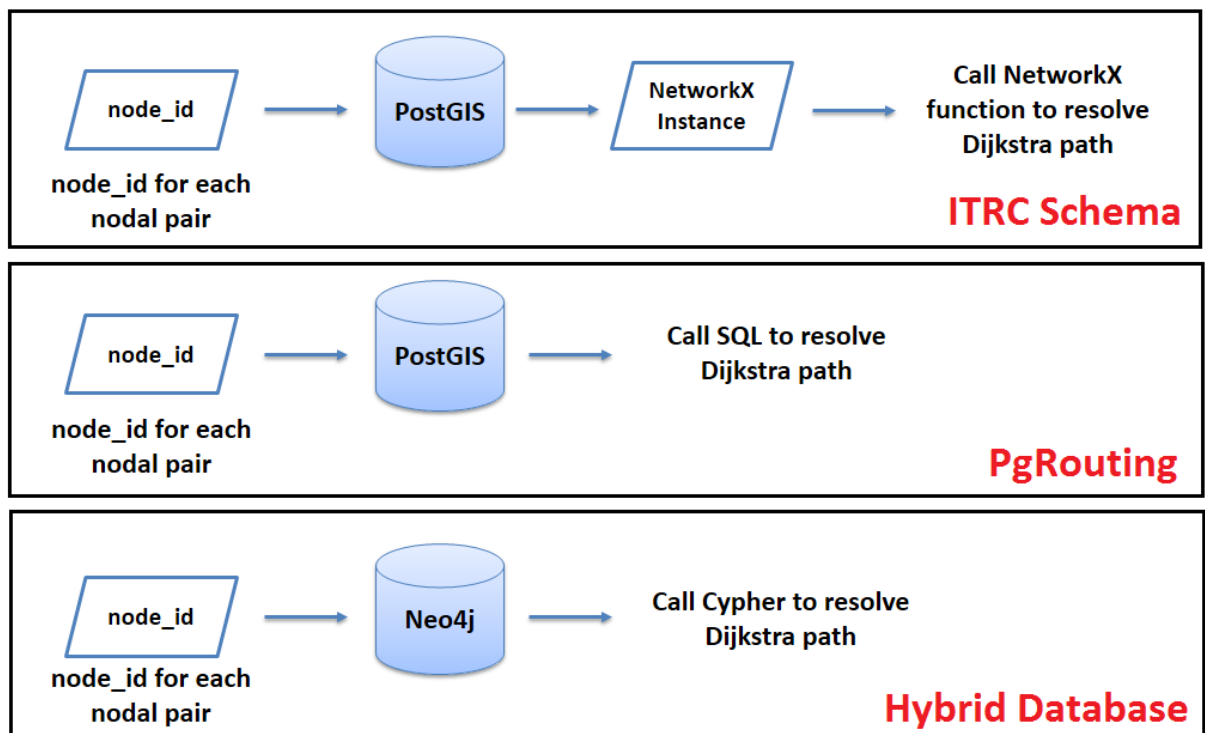
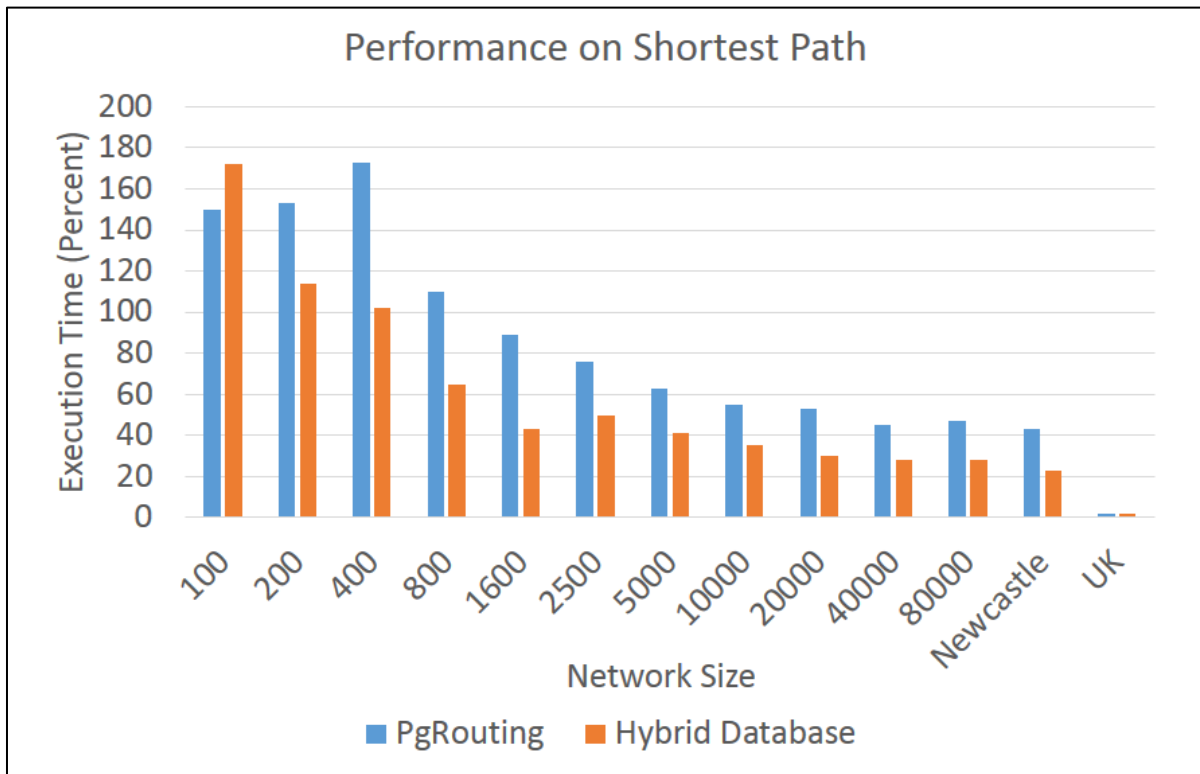


Figure 7.16. Pipe lines for shortest path query on **type 3** network data.

Figure 7.15 shows that to perform shortest path query on type 1 and 2 network data, the database needs to figure out which nodes are substation nodes first (via the attribute

**node\_type**), retrieve the **net\_id** for each substation node, and then find all the building nodes that have same **net\_id**. The all the substation nodes and their dependent building nodes are retrieved (essentially retrieving their **node\_id**), then Dijkstra shortest path calculation can be resolved on these nodes. However, when performing shortest path query on type 3 network data (figure 7.16), the **node\_id** (of the nodes to be resolved Dijkstra shortest path) is given, and therefore pipe lines are shorter.

The actual execution time of shortest path query is shown in figure I3, Appendix I. For ITRC schema, execution time is still proportional to the network size, from 1.8 seconds (query network of size 100) to 2502 seconds (query network of size ‘Newcastle’). For PgRouting and hybrid database, they spend 2.7 seconds and 3.1 seconds (query network of size 100) and 945 seconds and 595 seconds (query network of size ‘Newcastle’) respectively. The percentage performance comparison is shown in figure 7.17.



**Figure 7.17.** Performance comparison of performing shortest path query on different sized network data.

From table 7.17, it is found that PgRouting is slower than benchmark when network size is

smaller than 1600, but faster when network is larger. The hybrid database follows the same pattern, but that threshold value (network size) is 400. Again, this is because when querying small network, the actual time for the database driver to connect the database, is not neglectable. The benchmark becomes much slower than the other two approaches when network is large. For example, PgRouting and hybrid database is about 2.5 times and 5 times faster than ITRC schema, when querying network of size ‘Newcastle’. That is because ITRC schema needs to *read the entire network data into memory* to be able to query it. If a network stored in ITRC schema contains 1 million nodes, and even if the network query is very easy (e.g. find the neighbour for only one given node), still all these 1 million nodes needs to be read into memory.

This is *the biggest problem* for ITRC schema. It cannot directly perform network query on the database, but PgRouting and hybrid database can. That is why these two databases are about 40 times faster when query network data of size ‘UK’, which is not an expensive operation for PgRouting and hybrid database (both finish within one minute). Moreover, hybrid database is even faster compared with PgRouting (almost regardless of network size). That shows the graph engine of hybrid database (Neo4j) is more efficient in resolve network query, compared with the routing functionality provided by PgRouting.

## **7.5 Performance Test on City Scale Network Data from Newcastle**

In the last section, performance of three database approaches was evaluated in three sub-tests: writing, reading and query. The query (shortest path query) is a simple one, which is based on network attributes (on the nodes) and topology, and there is no geometry involved. However, as mentioned in Chapter 2 (literature review), geometry is an important part of geospatial infrastructure networks, therefore spatial query can be relevant or even frequent in the analysis of infrastructure networks. For example, the 2003 Italy blackout that affected the entire country, was only triggered by few cables that were damaged due to storm (Rosato, et al., 2008). Therefore, when analysing infrastructure network cascading failure triggered by

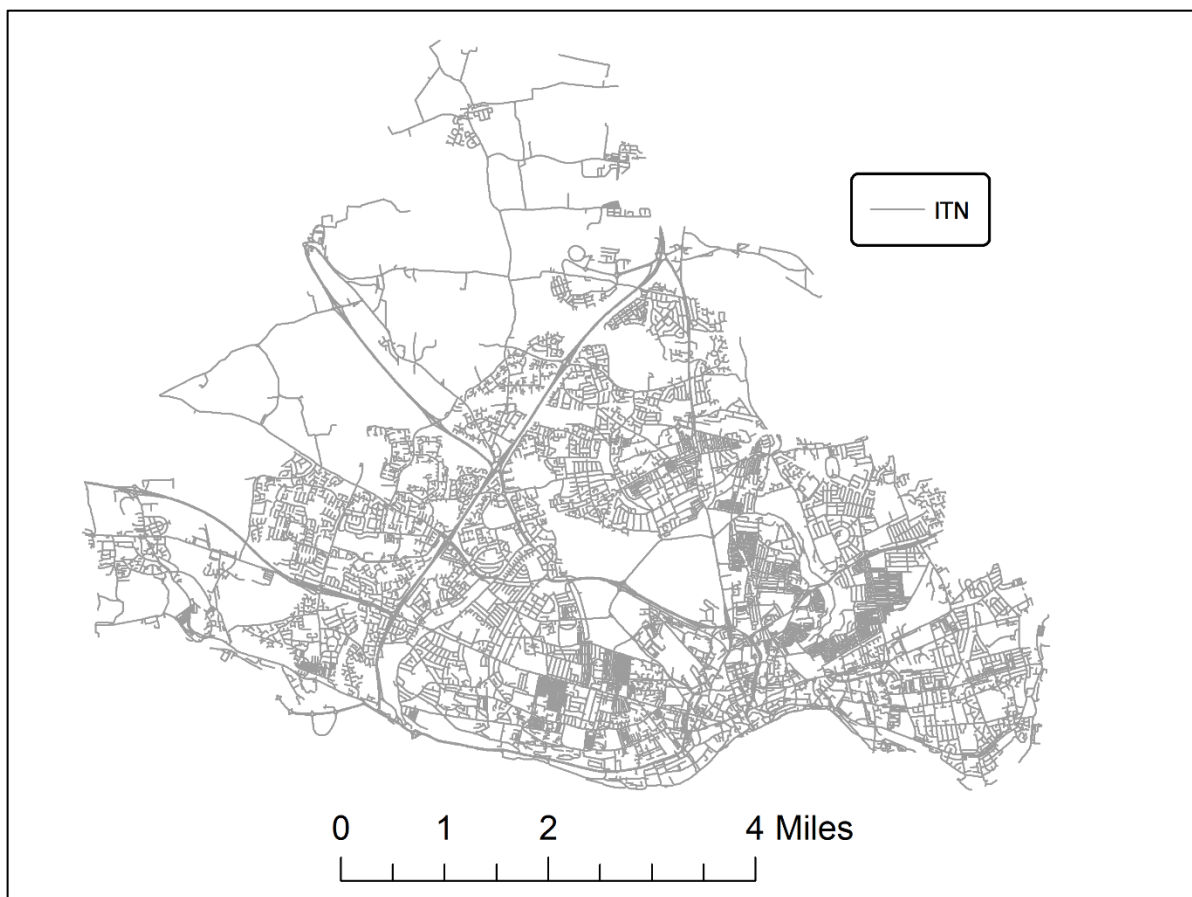
spatial hazards, a common operation is to first perform a spatial query to find affected network nodes or edges, and then perform a network query to find the affected nodes in the network.

Therefore, being able to efficiently resolve complex query (involving spatial, attribute and network query) is considered to be an essential capability of the database. In this section, performance tests will focus on this aspect and the tests can be much harder than the ones done in the last section. The tests are harder here because: 1) the networks are the entire city scale infrastructure network in Newcastle (a network can contain more than 200,000 nodes, and 2) the queries to be performed are more complex (than for example, a Dijkstra shortest path query). The details of test data and the performance tests are introduced below.

### ***7.5.1 Test Data***

There are two network data sets to be used in this test, the integrated road network (IRN) and the electricity distribution network in Newcastle upon Tyne.

The IRN (figure 7.18) is a synthetic network by integrating buildings into the existing road network (ITN) of Newcastle upon Tyne. It contains 13,698 nodes and 16,960 edges. The IRN is generated via the building-ITN integration algorithm (Listing 7.1), and the IRN contains 213,897 nodes and 217,166 edges. Of all the nodes in IRN, there are 104,855 building nodes.



**Figure 7.18.** The ITN network (Contains OS data © 2018).

---

**Algorithm :** Building-ITN Integration Algorithm

---

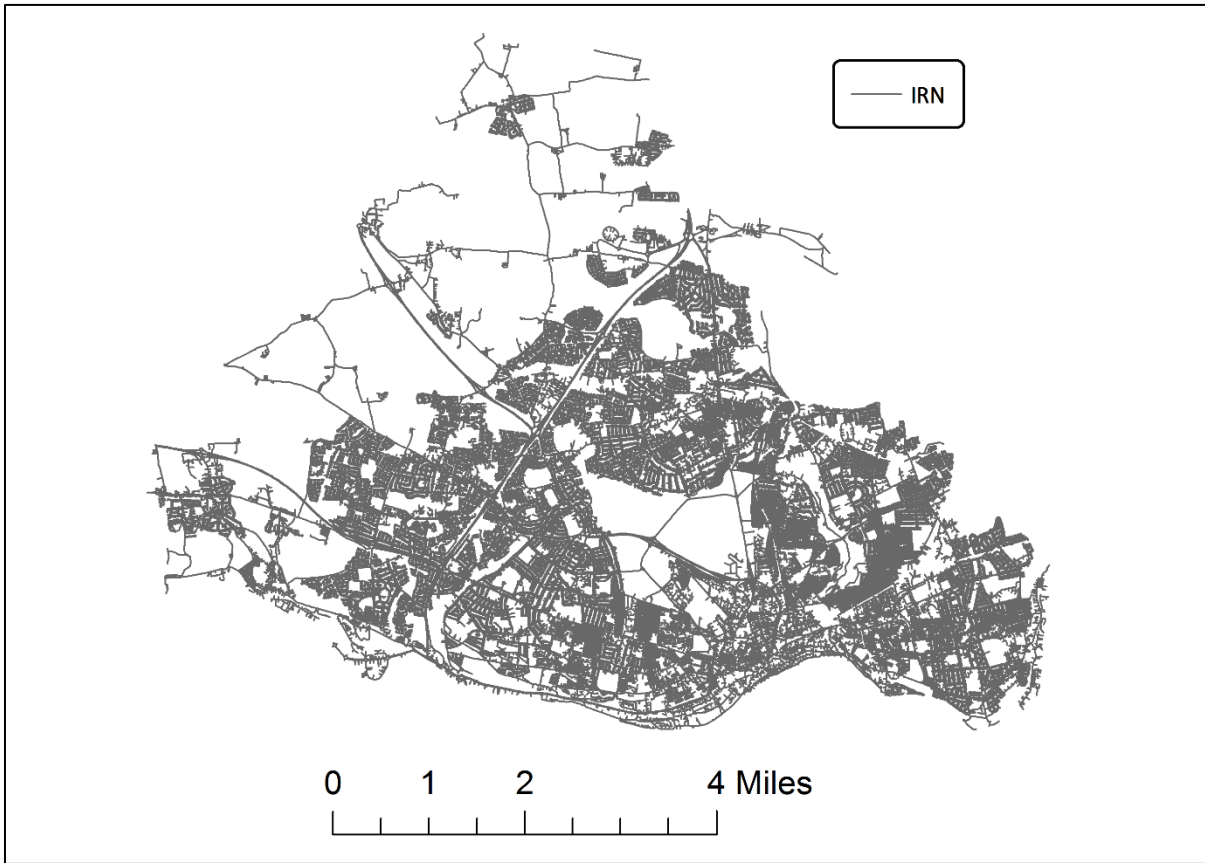
**Input:** a set of Buildings  $B$ , ITN network  $G_{initial}$

**Output:** IRN network  $G_{new}$

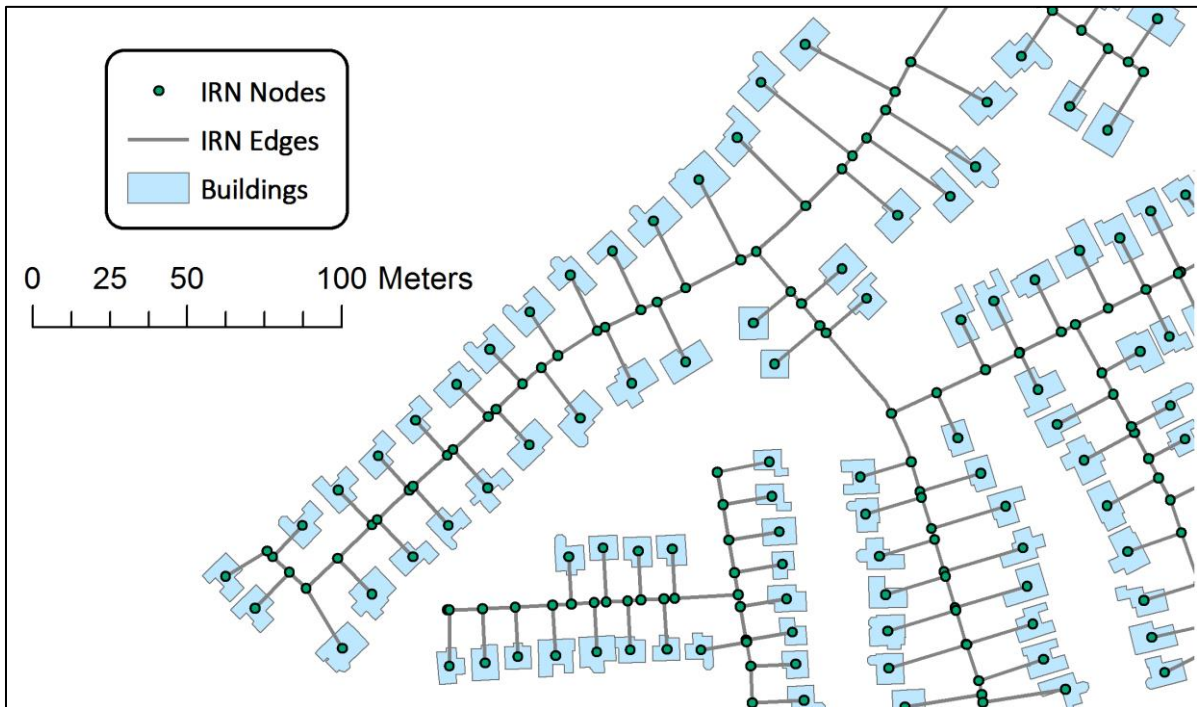
- 1: for  $b \in B$ , find nearest edge from  $G_{initial}$
  - 2: for  $b \in B$ , extract its centroid  $b.cen$  and then derive an edge to its nearest edge from  $G_{initial}$
  - 3: merge all the building nodes and derived edges (servicing pipes) to  $G_{initial}$  to save to  $G_{new}$
  - 4: on  $G_{new}$ , modify topology where necessary
- 

**Listing 7.1.** The building-ITN integration algorithm.

Figure 7.20 shows the IRN layout in a very fine spatial scale. The reason to generate the IRN (instead of using the original ITN) is that, it represents the spatial connectivity between buildings and roads, which is essential in modelling fine scale infrastructure networks.



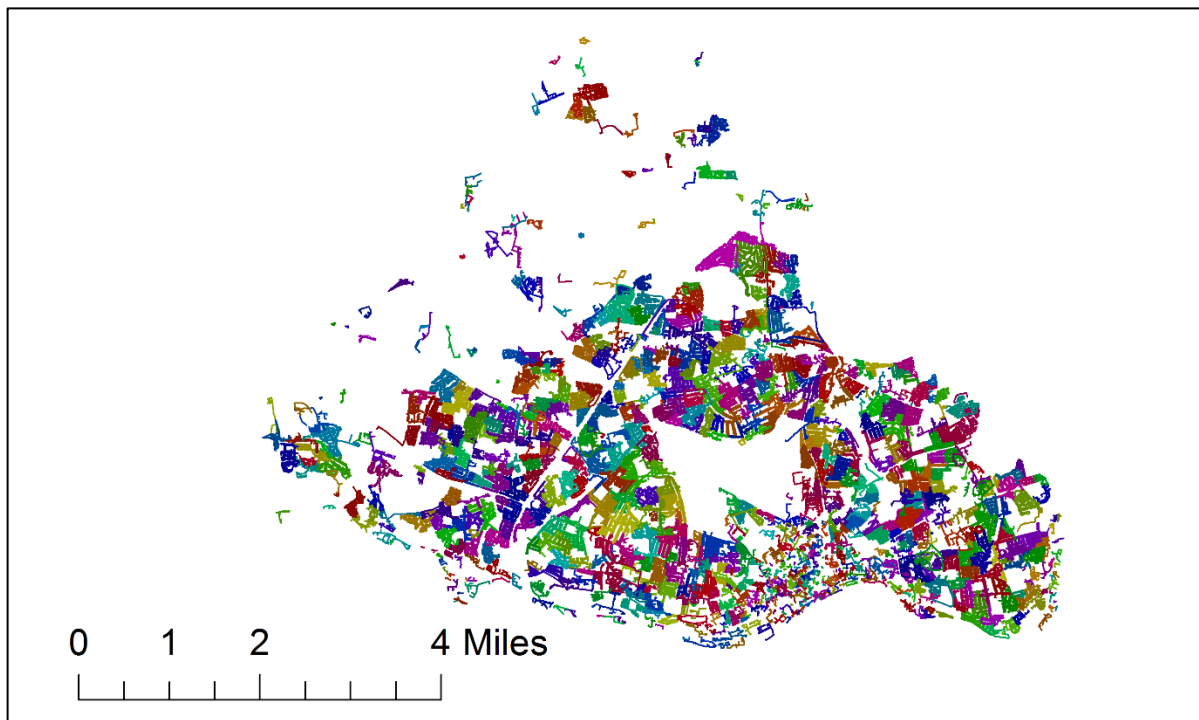
**Figure 7.19.** The IRN network (Contains OS data © 2018).



**Figure 7.20.** A closer view of the IRN, with regards to the building layout (Contains OS data © 2018).



The other network data used in the test, is the one we have seen in section 7.4, the type 2 network data of size ‘Newcastle’. This is the entire city scale electricity distribution network data in Newcastle upon Tyne (figure 7.21), containing 209,886 nodes and 209,892 edges.

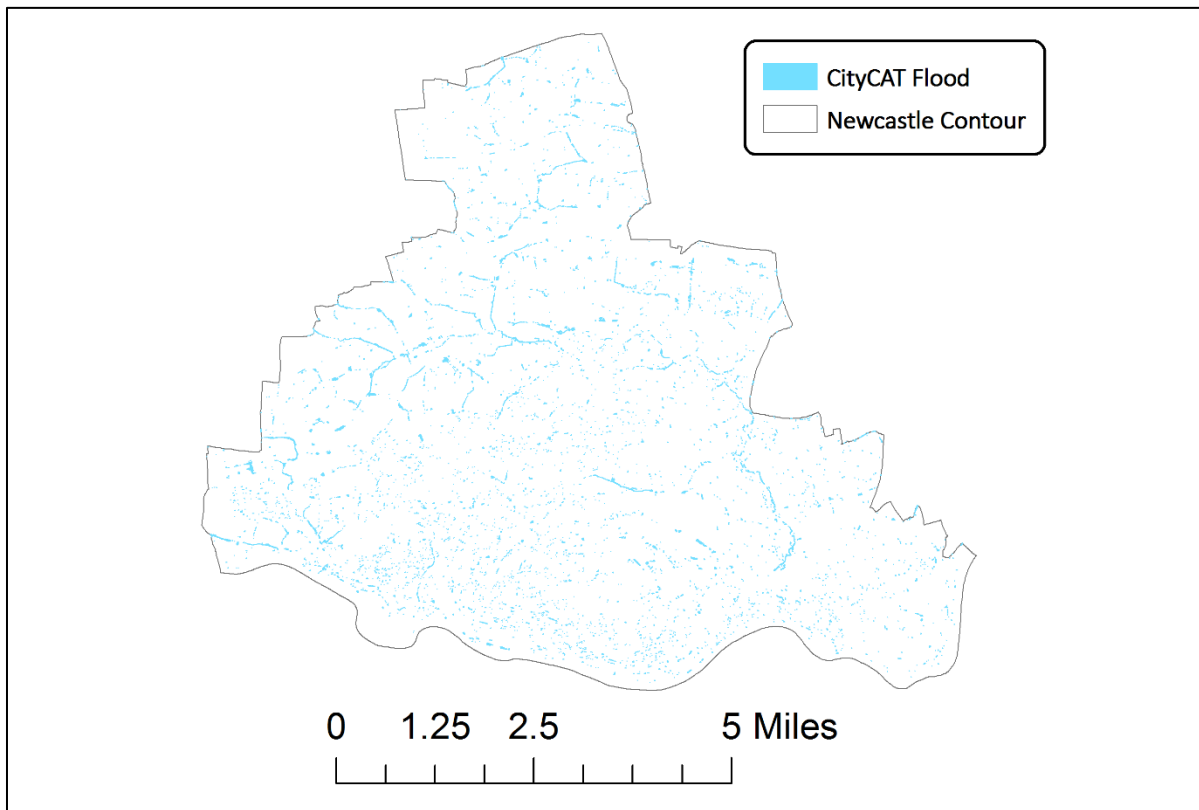


**Figure 7.21.** Entire city scale electricity distribution network data in Newcastle upon Tyne. Each colour refers to a single network instance.

The reason to choose these two different data sets here, is that they have relatively different topologies. The IRN network is a single large network instance that contains about 200,000 nodes. While the Newcastle electricity distribution networks data is about the same size, it consists of 636 single network instances. These are the two common types of urban infrastructure networks (one of a single large connected network instance and one of multiple connected network instances). It is considered a good database approach should be able to handle both network data efficiently, and that is why both of them are used in the test.

One thing to mention in this section is that tests will not be done on *writing* and *reading* these two network data sets, because the operations on similar sized data have been evaluated in the last section (7.4). Instead, the focus of this section is to *query* the network data, and that relies on another data. It is an ESRI shapefile Polygon layer (figure 7.21), which contains the spatial

footprint of floods in Newcastle upon Tyne, and it is generated by an urban flood model CityCAT (Glenis, et al., 2013).



**Figure 7.22.** The CityCAT flooding footprint.

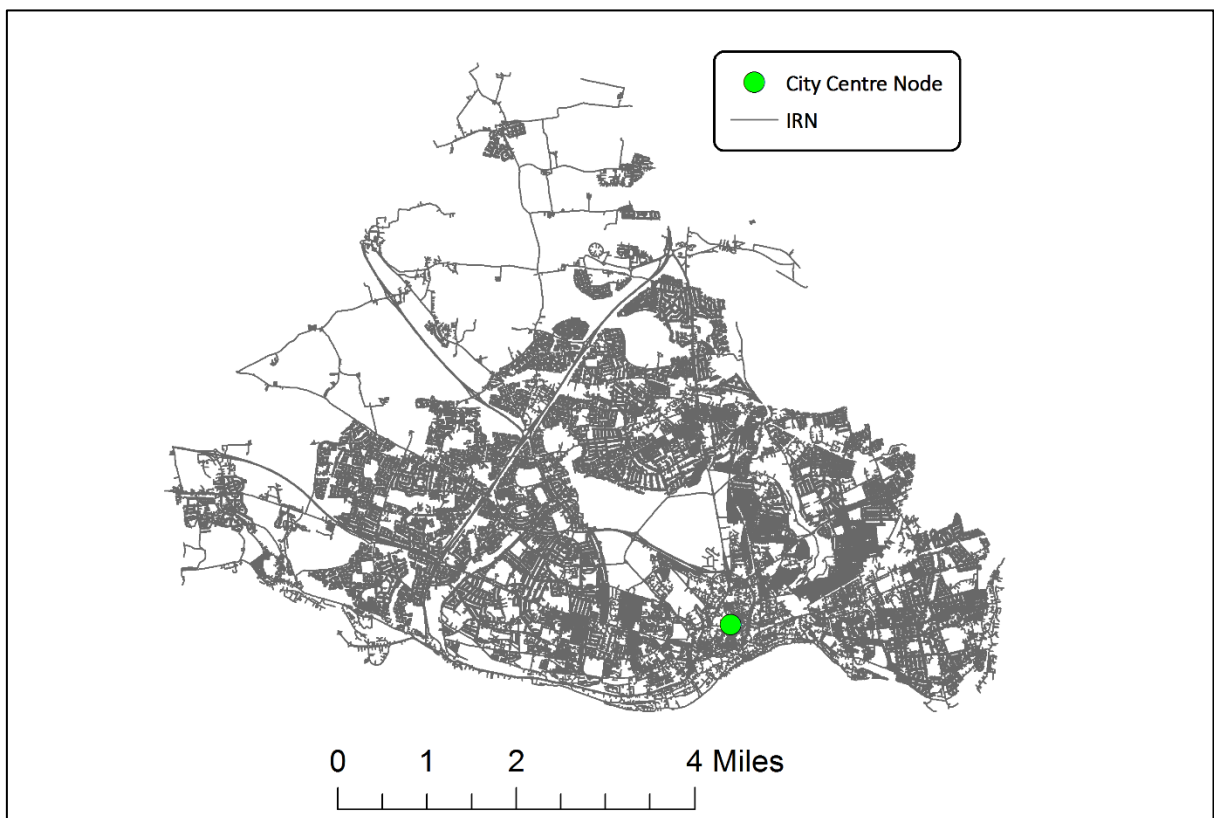
The reason to use a flooding footprint is that, Newcastle upon Tyne (and in fact UK in general) suffered severely from flooding (Glenius, et al., 2013). The most recent flooding hazard in Newcastle upon Tyne occurred on June 28, 2012, when 50 mm rain fall in two hours (which basically should have been a month's amount) caused £ 8m of damage to homes, roads and businesses, and 3000 residents were affected (BBC News, 2012). Given the above, flooding footprint is used as it is a spatial hazard that can *actually occur* in the city. The actual performance tests (on the IRN and electricity distribution network) are discussed as follows.

### **7.5.2 Performance Test on querying Newcastle Integrated Road Network**

Road network routing applications often need to solve *conditional shortest path problems* for their customers (Medhi, 2017). The *conditional* means some edges in the road network are not

used when resolving shortest path (e.g. some roads are blocked due to construction work, or damage). This test simulates a scenario, in which roads (in the IRN) within flooding footprint are submerged and cannot be used. The test evaluates how the failure on IRN affects the travel ability of residents from their houses to a pre-defined city centre node (figure 7.23).

The location of the city centre node is the Newcastle Monument Plaza (Wikipedia, 2018), which is considered to be the most crowded area and centre for the city. The query on the IRN (called IRN complex query) is shown in table 7.2. It is a long query that consists of four small steps. The pipe lines for each database to resolve IRN complex query is shown in figure 7.24.



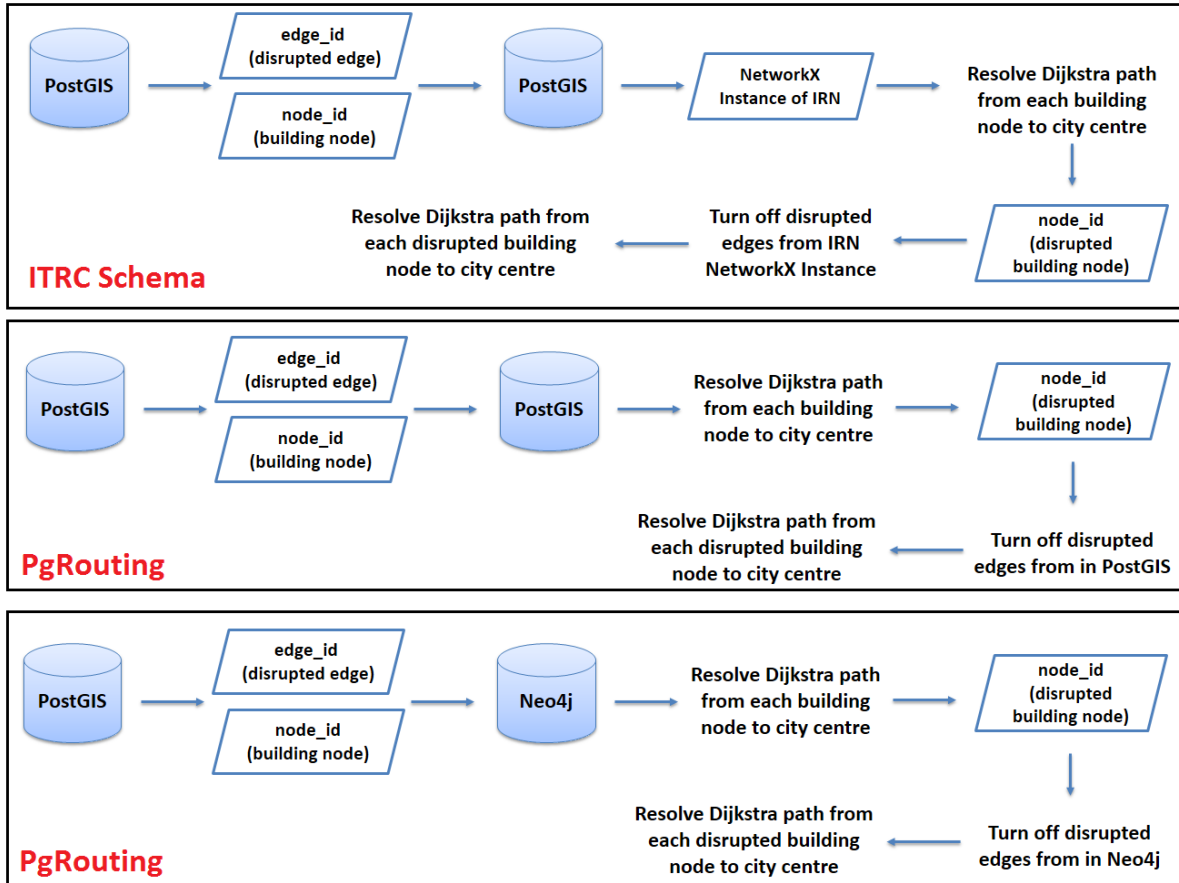
**Figure 7.23.** The IRN and city centre node.

<b>IRN complex query</b>	<b>Operation</b>
Step 1	Find IRN's edges that are intersecting with CityCAT flooding footprint, mark them as <i>disrupted edges</i>
Step 2	Resolve Dijkstra path from each building node to the city centre
Step 3	If a building's shortest path consists of at least one disrupted edge, mark the building as a <i>disrupted building</i>

Step 4

Turn off the disrupted road segments, re-calculate Dijkstra shortest path, for each *disrupted building* to the city centre node, if there is still a path

**Table 7.2.** Breakdown of IRN complex query.

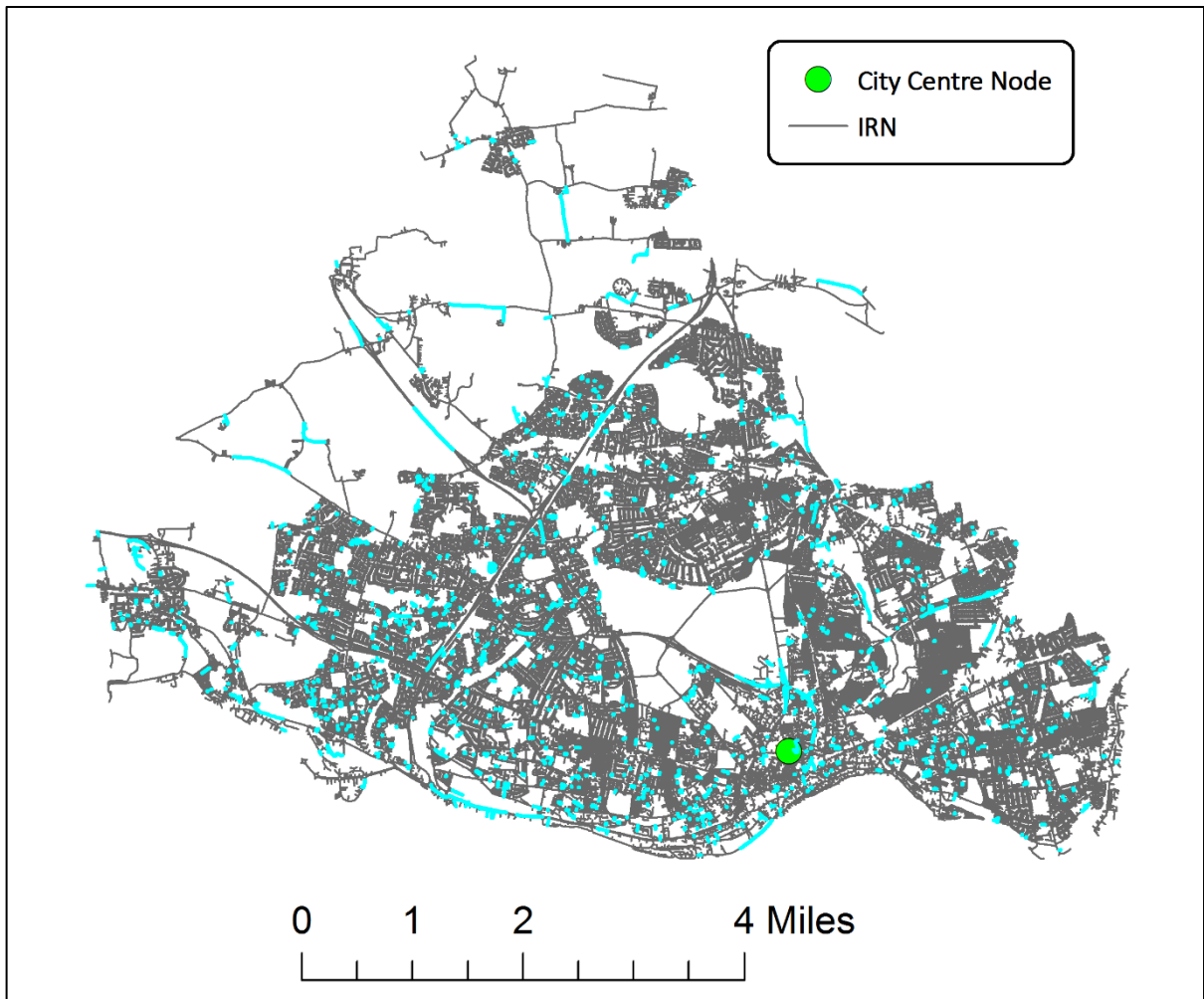


**Figure 7.24.** Pipe lines to resolve the IRN complex query.

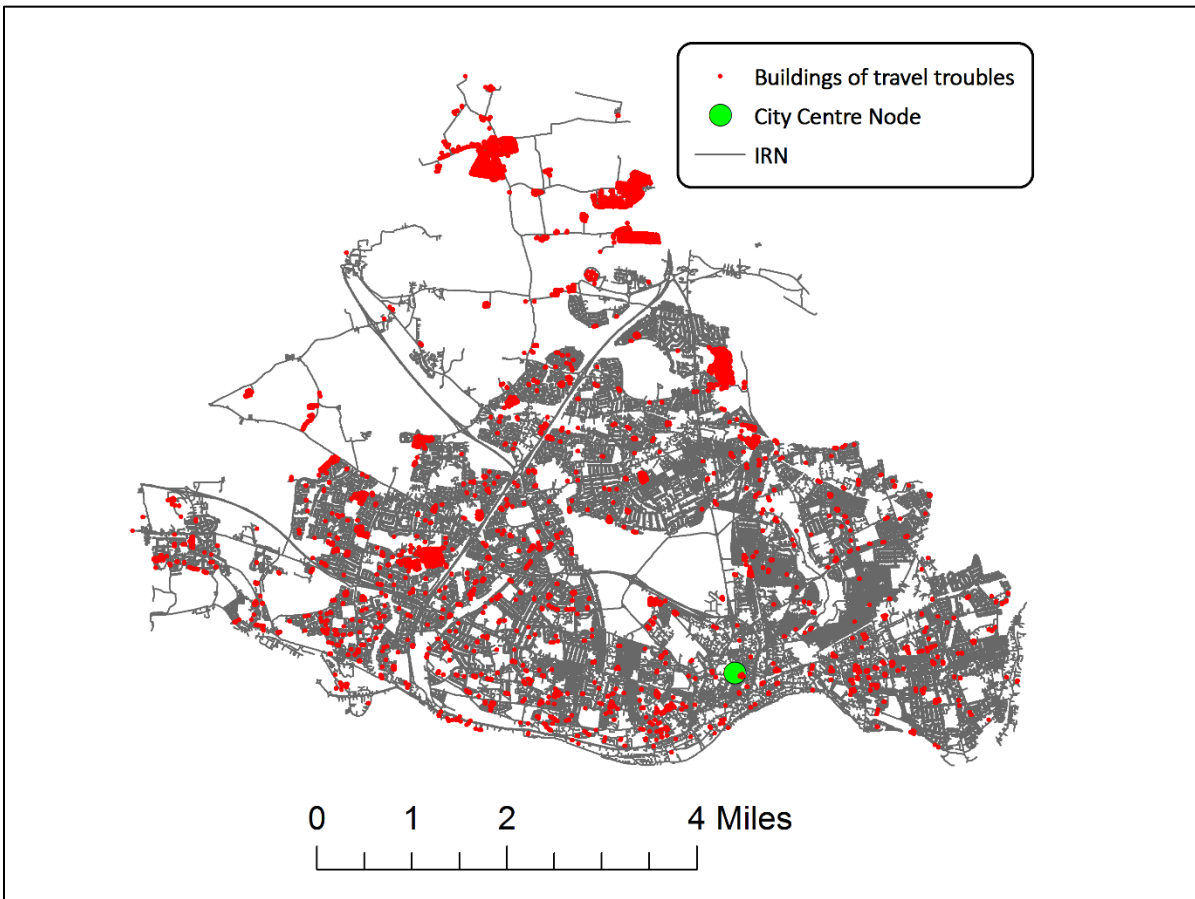
For any database approach, first a spatial query is resolved to find disrupted edges within flooding footprint. An attribute query is resolved to find the building nodes (node\_type = ‘building’). After that, there will be differences for each database approach. ITRC schema needs to read IRN instance into NetworkX instance and then perform shortest path query. While PgRouting and hybrid database can query IRN directly. Note that IRN needs to be queried twice (1<sup>st</sup> time is to find disrupted buildings, and 2<sup>nd</sup> time is to resolve shortest path from disrupted buildings to city centre).

As a result, the IRN complex query found that 2397 edges from IRN are disrupted (figure

7.25). For all the 104,855 building nodes, 67% of them (70,120) buildings are disrupted. After turning off the disrupted edges, for all the disrupted building nodes, 64,841 of them still have new shortest paths, but the remaining 5279 are could no longer reach the city centre node (figure 7.26).

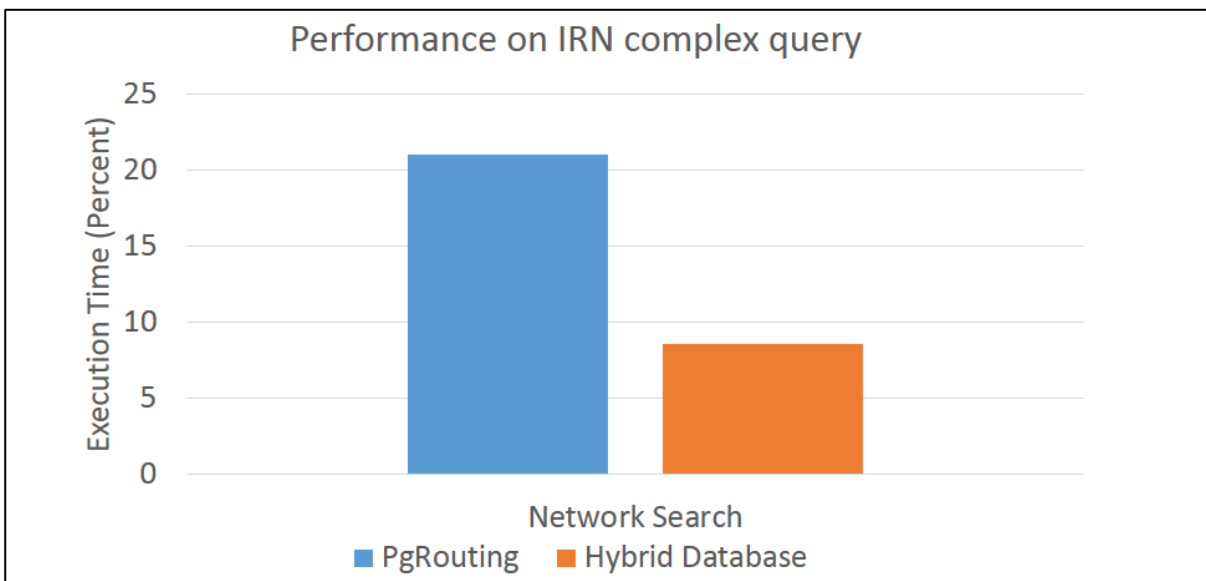


**Figure 7.25.** 2397 disrupted edges (in Cyan) in the IRN.



**Figure 7.26.** 5279 building nodes that cannot reach city centre due to flood.

The IRN complex query execution time is shown in figure I4, Appendix I. The execution time for ITRC schema, PgRouting and hybrid database is 24,602, 5183, and 2139 seconds respectively. The relative performance comparison is shown in figure 7.27.



**Figure 7.27.** Performance comparison of executing IRN complex query.

It is found that the IRN complex query is so difficult for the ITRC schema that it costs more than 6 hours. While for PgRouting and hybrid database, they are about 5 times and 12 times faster. All the three approaches use PostGIS to resolved spatial query (find disrupted edges within flood), and therefore the performance difference is related to how the database resolves network query. The ITRC schema has poor performance since still it needs to read the IRN network into memory to be able to query it.

Moreover, ITRC schema has another disadvantage, which is it *only* supports shortest path query that *has a single start node and a single end node*. That means when resolving Dijkstra path from each building node to the city centre node, it needs to iterate on every building node, and resolve shortest path from that building node (to city centre node). This is less flexible, as PgRouting and hybrid database (actually Neo4j inside) allows for shortest path query that has multiple start nodes or multiple end nodes. For this example, hybrid database is about 2.3 times faster than PgRouting, shows Neo4j's property graph model and Cypher is more efficient than PgRouting's relational tables and SQL, for querying a large network instance at city scale.

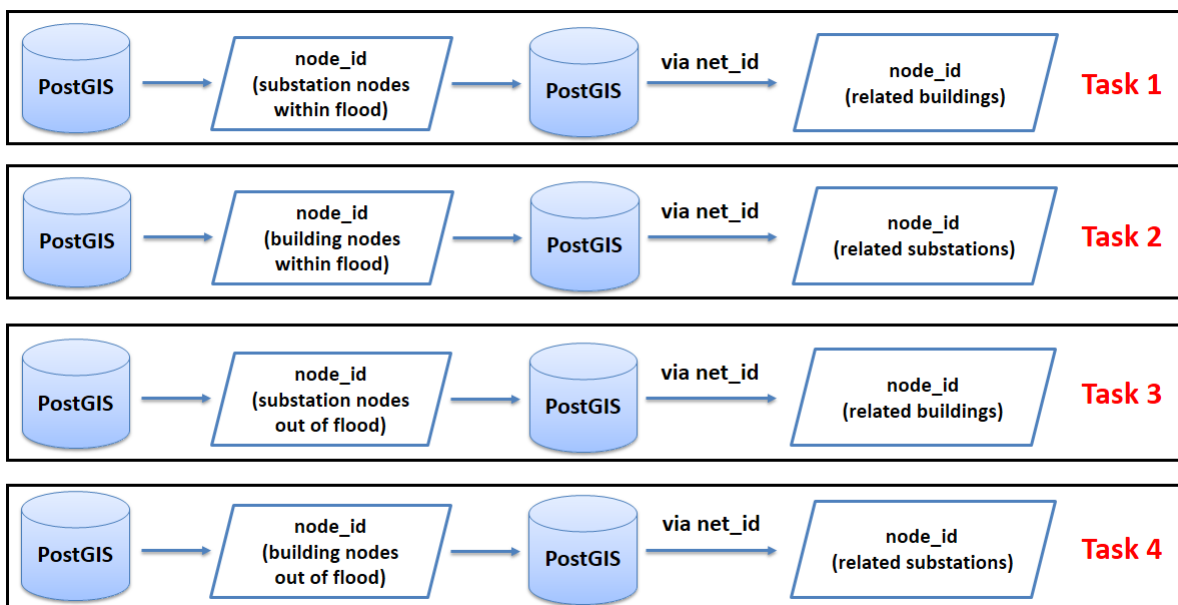
### ***7.5.3 Performance Test on querying Newcastle Electricity Distribution Network***

The IRN complex query in section 7.5.2 showed a typical scenario when geometry on network edges are queried. The IRN complex query is virtually a long query consisting of a spatial query and two shortest path queries. The hybrid database outperformed the other two due to its graph engine (Neo4j). However, it is still not clear how efficient the hybrid database is, *suppose it is only used to perform spatial and attribute queries, but no network query*. Therefore, this section is developed by such intention. The network data used here is entire city scale electricity distribution network in Newcastle upon Tyne (figure 7.21). There are 636 network instances, and each network instance has a unique **net\_id**, which is assigned as an attribute to every node and edge in this network instance. The test here is called a complex query on Newcastle Electricity Network, and it consists of four distinctive tasks (table 7.3).

Task	Operation
1	Find substation nodes within flood, and then find all the buildings served by these substations (has same <b>net_id</b> )
2	Find building nodes within flood, and then find all the substations serving them (has same <b>net_id</b> )
3	Find substation nodes <b>NOT</b> within flood, and then find building nodes served by these substations (has same <b>net_id</b> )
4	Find network instances which contain <b>NO</b> flooded buildings, then find substations from these network instances

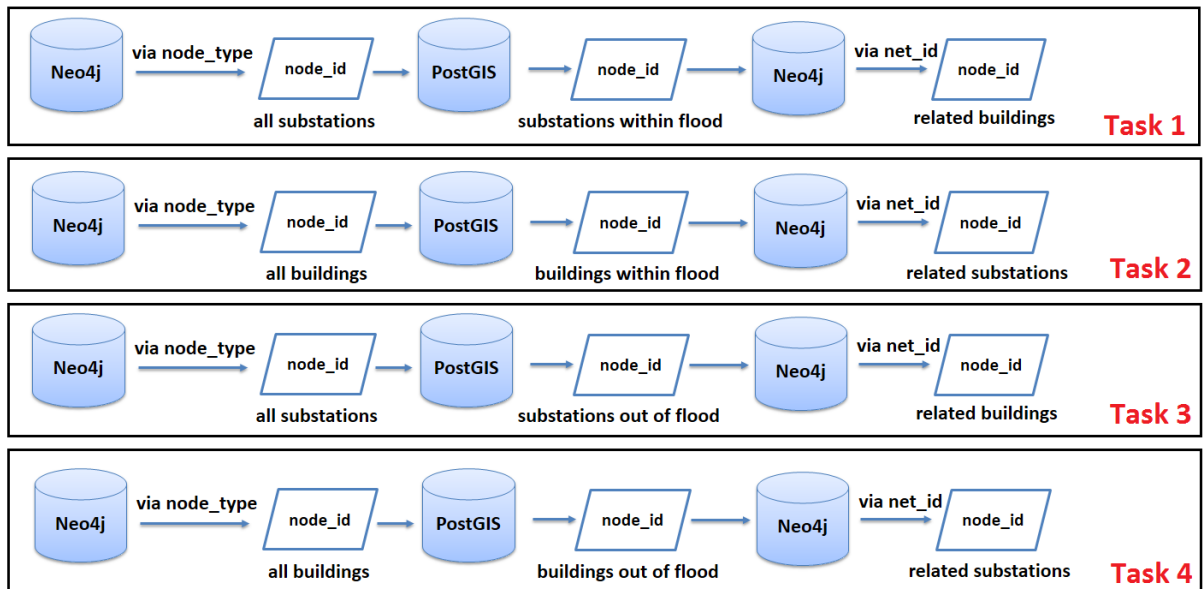
**Table 7.3.** Four tasks for complex query on Newcastle Electricity Network.

Each task in table 7.3, is designed to evaluate database performance when handling a *spatial query plus attribute queries*. For each task, a spatial query is done to find substation nodes or building nodes (within or out of) the flooding footprint, then attribute queries are done to find the dependent building nodes or substation nodes. Note task 1 and 3 are negation operations, so are task 2 and task 4. The reason to design the four tasks this way, is that asset nodes and building nodes are considered to be of top priorities when assessing impact of spatial hazard to infrastructure network. To resolve this complex query, the pipe lines of the ITRC schema and PgRouting are exactly the same, and shown in figure 7.28. The pipe lines of hybrid database is shown in figure 7.29.



**Figure 7.28.** Pipe lines for **ITRC schema** and **PgRouting**, to resolve complex query on Newcastle Electricity Network.



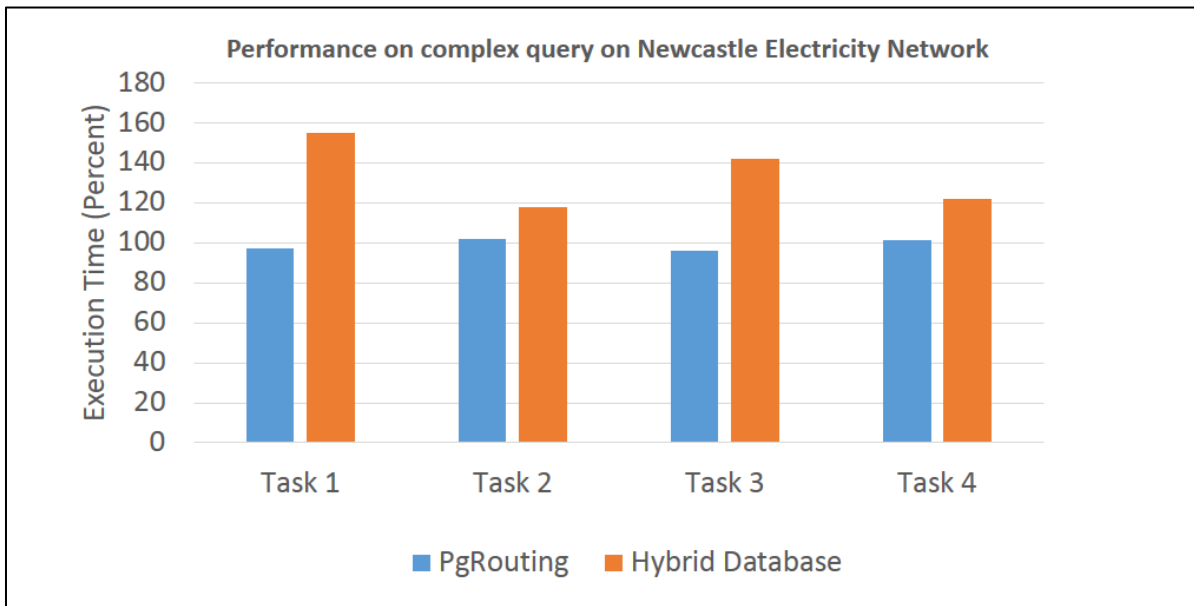


**Figure 7.29.** Pipe lines for **hybrid database**, to resolve complex query on Newcastle Electricity Network.

The reason for ITRC schema and PgRouting to have same pipe lines is that, they both use PostGIS relational table to store attributes. While for hybrid database, attributes are stored in property graph in Neo4j, so that pipe lines are longer. The actual retrieved number of buildings or substations are shown in table 7.4. The execution time of complex query on Newcastle Electricity Network is shown in figure 15, Appendix I. The execution time of ITRC schema and PgRouting are almost the same, which are about 3.7, 204, 26, and 257 seconds, for task 1, 2, 3 and 4 respectively. While the execution time hybrid database is longer, which are 5.6, 241, 37, and 314 seconds respectively. The performance comparison is shown in figure 7.30.

Task	Task Result
1	retrieved substations: 2, retrieved buildings: 372.
2	retrieved buildings: 586, retrieved substations: 15.
3	retrieved substations: 634, retrieved buildings: 104,483.
4	retrieved network instances: 621, retrieved substations: 621.

**Table 7.4.** Result of complex query on Newcastle Electricity Network.



**Figure 7.30.** Performance comparison on complex query on Newcastle Electricity Network.

It is not surprising to see PgRouting is almost exactly as fast as ITRC schema, since there is no network topology query, but only relational queries using PostGIS. Hybrid database is about 1.2 – 1.4 times slower. The major reason is that attributes are only stored in Neo4j, so the hybrid database needs to switch between Neo4j and PostGIS multiple times to get final result.

### 7.6 Performance Test on Mega City Scale Network Data from London

Section 7.5 evaluated database performance to process entire city scale infrastructure network for Newcastle upon Tyne. However, it is actually a small city, and if ranked by population, it is the 30<sup>th</sup> largest city in the UK (City Mayors, 2018). There are many mega cities in world, much larger than Newcastle, such as London, New York, Tokyo, and Shanghai. The purpose of data performance benchmarking test is to choose a database that is a generic data management solution for city of any size. That is the reason to develop this section 7.6, in which network data from a mega city London is used to test database performance.

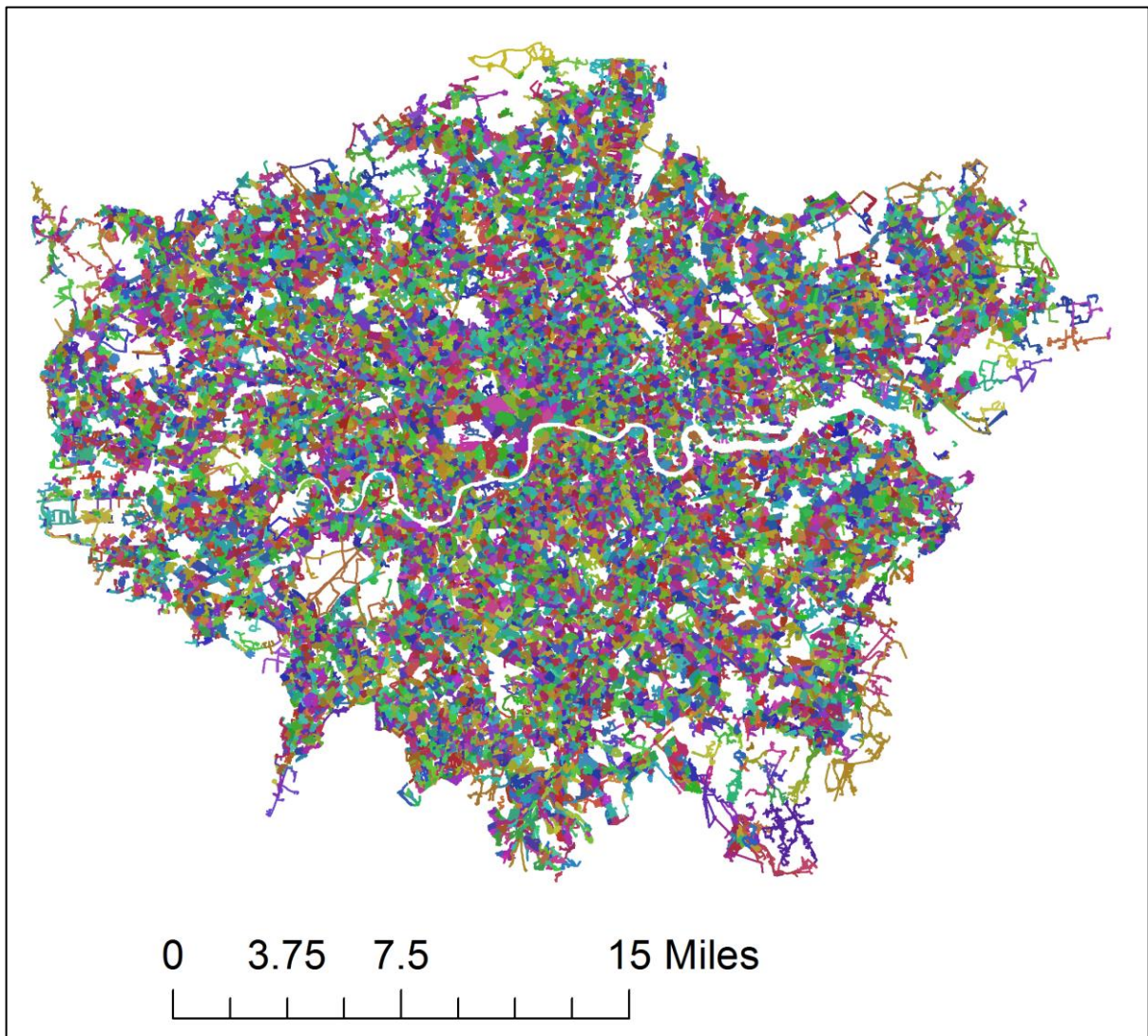
The tests to be performed in this section are considered of highest complexity (compared with tests in section 7.4 and 7.5) due to the data volume. Simple tests (writing, reading, and

shortest path query) are performed as well as complex tests (e.g. combining spatial query with attribute or network queries). Details of network data and tests are introduced below.

### 7.6.1 Test Data

The entire city scale electricity distribution network data of London (generated in chapter 4) is used here (figure 7.31) which comprises of totally 4,528,952 nodes and 4,512,779 edges.

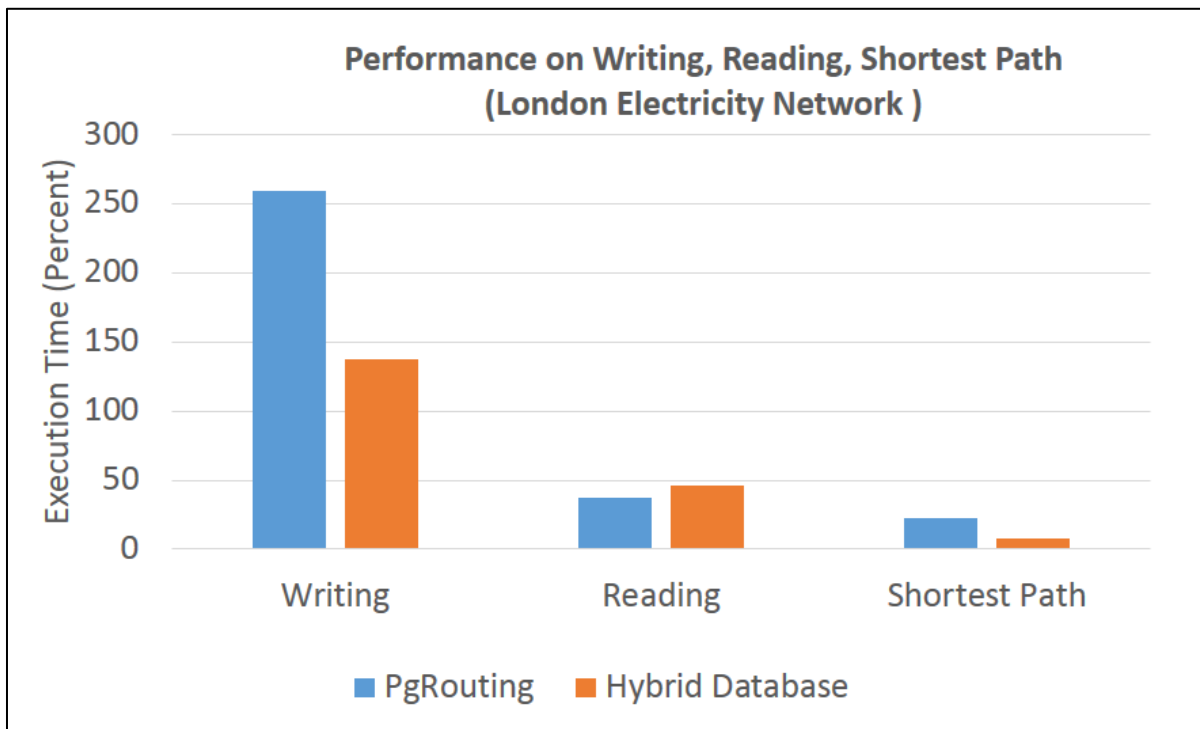
There are 16,839 network instances (substations) which serve electricity to 2,239,213 buildings.



**Figure 7.31.** Electricity distribution networks of London. Each colour refers to a network instance.

### 7.6.2 Writing, Reading, and Shortest Path Test

Database performance on writing, reading and shortest path queries are evaluated using London electricity network data. Shortest path query is the same as the one in section 7.4.3, which is “resolve Dijkstra shortest path from each substation to all its dependent buildings”. The execution time for these tests are shown in table I6, Appendix I. To write the network, the ITRC schema, PgRouting and hybrid database spent 47688, 123961, and 65322 seconds. To read the network, these three approaches spent 64785, 23728, and 29897 seconds, respectively. To perform shortest path query, these three approaches spent 58980, 13716, and 5034 seconds, respectively. Based on these values, the percentage performance is shown in figure 7.32.



**Figure 7.32.** Performance comparison on performing writing, reading, and shortest path queries on London electricity network data.

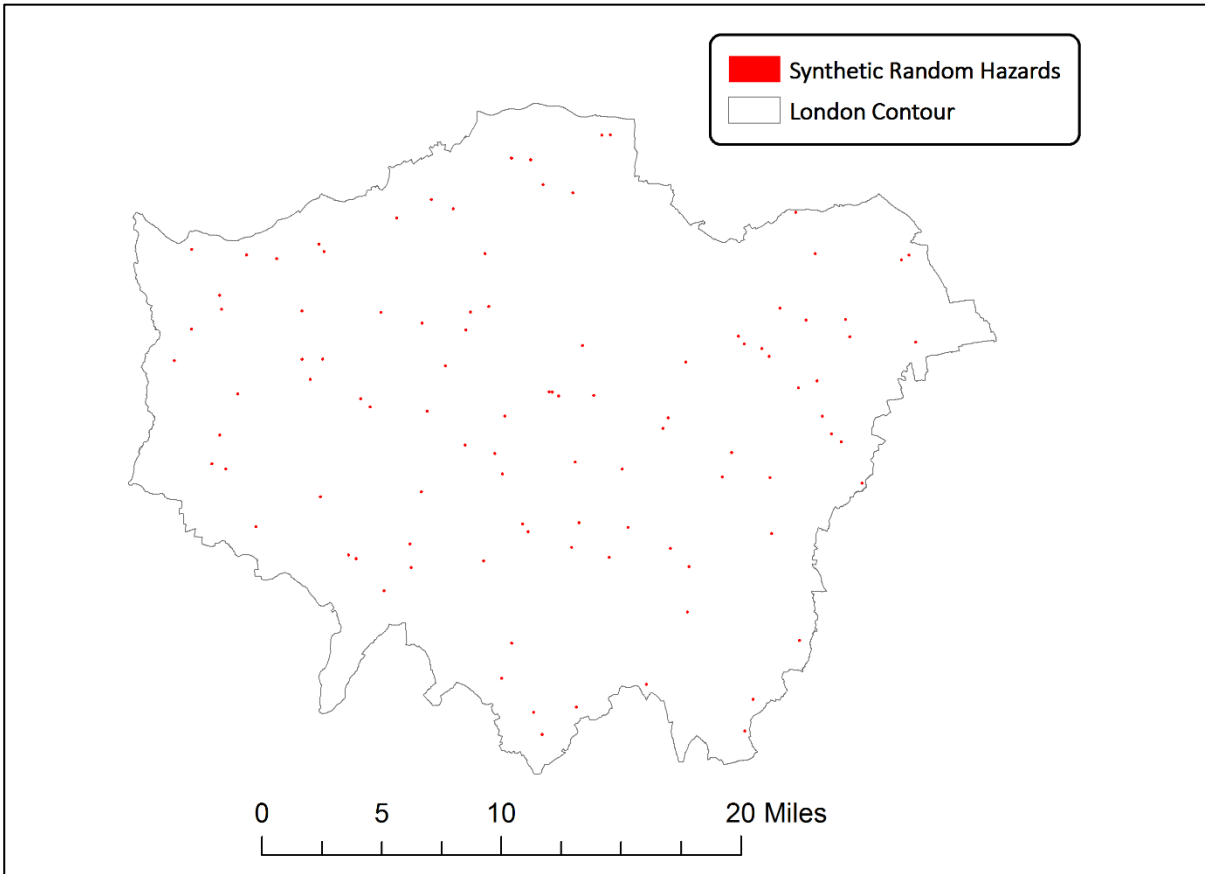
With regards to network size (number of nodes), the London electricity network data set is about 21 times larger than the Newcastle electricity network data set. It is interesting to see, that for any of the three database approaches, writing and reading execution time also increase almost 21 times. ITRC schema is still the fastest at writing (cost almost 13 hours), while

hybrid database and PgRouting are 1.35 and 2.59 times slower. For reading data, PgRouting is still the fastest one (around 6.6 hours), followed by hybrid database (around 8.3 hours), and they are 2.7 and 2.1 times faster than the benchmark. For shortest path query, ITRC schema is still slowest one (cost 16 hours, very unacceptable), while PgRouting and hybrid database are 4.3 and 12 times faster. When performing shortest path query on Newcastle electricity network data in section 7.4.3, the PgRouting and hybrid database were 2.5 times and 5 times faster than ITRC schema. That means, as network data size increases, PgRouting and hybrid databases have better scalability (on network query such as shortest path) compared with the ITRC schema. The major reason is that, the graph engine ITRC schema uses (the NetworkX library) is less efficient compared with PgRouting and Neo4j when querying extremely large network data (e.g. at mega city size).

### ***7.6.3 Complex Query Test***

Two complex queries are designed (called complex query 1 and 2 as below), when database needs to perform a spatial query plus attribute or network topology queries. A spatial footprint (figure 7.33) is used for both complex queries. It is a generated synthetic data, which consists of 100 polygons, and each polygon is a circle of 100 meters radius.

The circles are generated randomly within contour of London, and each circle simulates a spatial hazard that can occur in London. The reason to use only 100 circles (instead of 10,000 for example) is to make sure every database approach can finish complex query still in almost acceptable time. This is the same reason to use 100 meters as circle radius, instead of 10,000 meters for example.



**Figure 7.33.** Synthetic random hazards used for complex queries.

**Complex query 1** is almost same as the one discussed in section 7.5.3. The only difference is that in complex query 1, there is an iteration over every single random hazard. In each iteration, substation nodes (or building nodes) within (or out of random hazard) are retrieved, and then related building nodes or substation nodes are retrieved. The complex query 1 consists of four tasks (table 7.5). Similarly, a unique **net\_id** is given to every network instance and assigned to every node and edge. Therefore, every task in table 7.5 is actually a *spatial query* plus *attribute queries*. The pipe lines of database to resolve complex query 1 are the same as the one shown in figure 7.28 and 7.29.

<b>Task</b>	<b>Operation</b>
1	<u>For each random hazard:</u> find the substations within, and then all the buildings served by these substations
2	<u>For each random hazard:</u> find the buildings within, and then all the substations serving them
3	<u>For each random hazard:</u> find the substations <b>NOT</b> within, and then all the buildings served by these substations

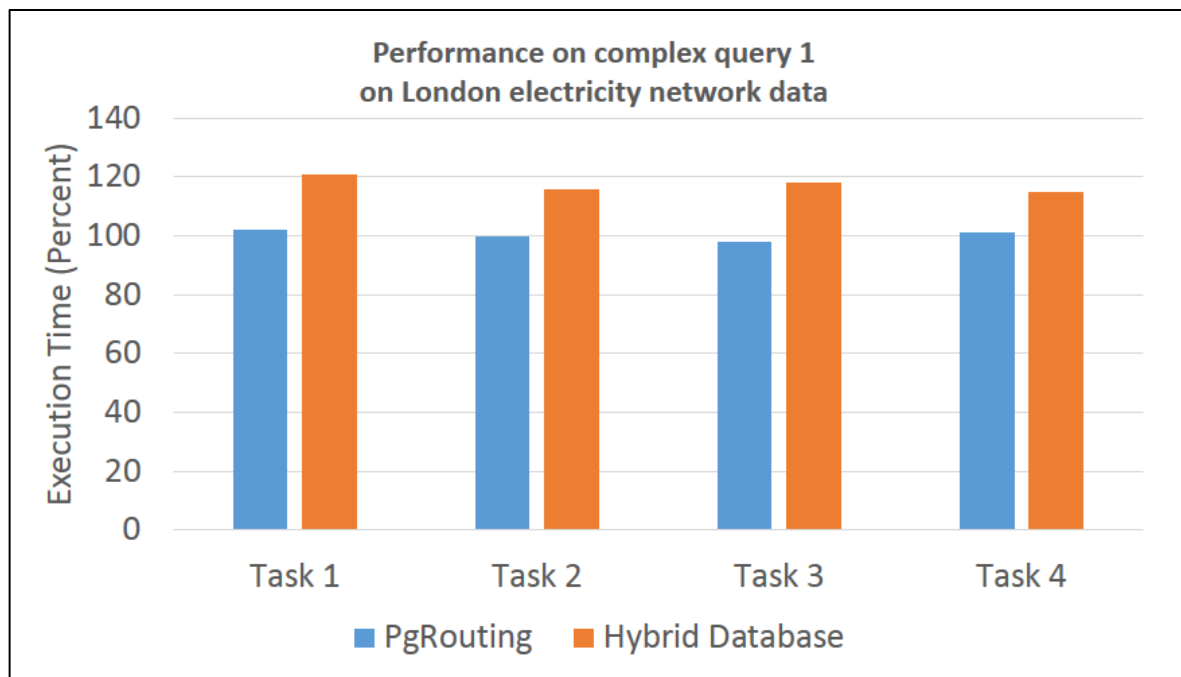
- 
- 4 For each random hazard: find network instances (where no building nodes are within hazard), then find all the substations from these network instances
- 

**Table 7.5.** Four tasks in **complex query 1** on London electricity network data.

The result of complex query 1 is shown in table 7.6, in which the average numbers of building nodes (or substation nodes) retrieved are displayed. The query execution time is shown in table I7, Appendix I. The execution time of ITRC schema and PgRouting are almost same, and for the four tasks, it is about 2168, 2205, 2140, and 2590 seconds respectively. The execution time of hybrid database is slightly longer, which is 2620, 2561, 2396, and 2990, respectively. The percentage performance comparison on complex query 1 is shown in figure 7.34.

Task	Result (Avg No. Data Retrieved on each Random Hazard)
1	Substations: 0.3, Buildings: 376
2	Buildings: 245, Substations: 1.9
3	Substations: 16838.7, Buildings: 2238837
4	Network Instances: 16837.1, Substations: 16837.1

**Table 7.6.** Result of **complex query 1** on London electricity network data.



**Figure 7.34.** Performance comparison on performing **complex query 1** on London electricity network data.

From figure 7.34, PgRouting is still as fast as ITRC schema, since they use both PostGIS to resolve spatial and attribute queries. The hybrid database is about 1.15 – 1.2 times slower than them, due to split storage of data. However, if compared with figure 7.30 (in which hybrid database is about 1.2 – 1.4 times slower), it is found the hybrid database (more precisely the Neo4j inside) shows good scalability in performing attribute query when network size increases.

**Complex query 2** is based on performing spatial query on (disrupted) network edges first, and then network topology query to find related building or substation nodes. The complex query 2 consists of 2 different tasks in which one negates the other (table 7.7). Note each electricity network instance is a network with direction (electricity flows from the substation node to building nodes), that is why complex query 2 considers flow direction. Like complex query 1, each task in complex query 2 is resolved on *each* random hazard separately. Pipe lines for complex query 2 are displayed in figure 7.35. Still ITRC schema has the longest pipe line compared with the other two, as it requires reading network data into NetworkX instance in the process. The result for complex query 2 is shown in table 7.8.

Task	Operation
1	<u>For each random hazard:</u> find network edges within that hazard, and then find all the downstream buildings (disrupted buildings) served by these edges and all the upstream substations serving these edges.
2	<u>For each random hazard:</u> find network edges within that hazard, and then find all (downstream) buildings served (disrupted buildings) by these edges. Then do the negation to find the undisrupted buildings. Finally find all the substations serving these undisrupted buildings.

**Table 7.7.** Two tasks in **complex query 2** on London electricity network data.



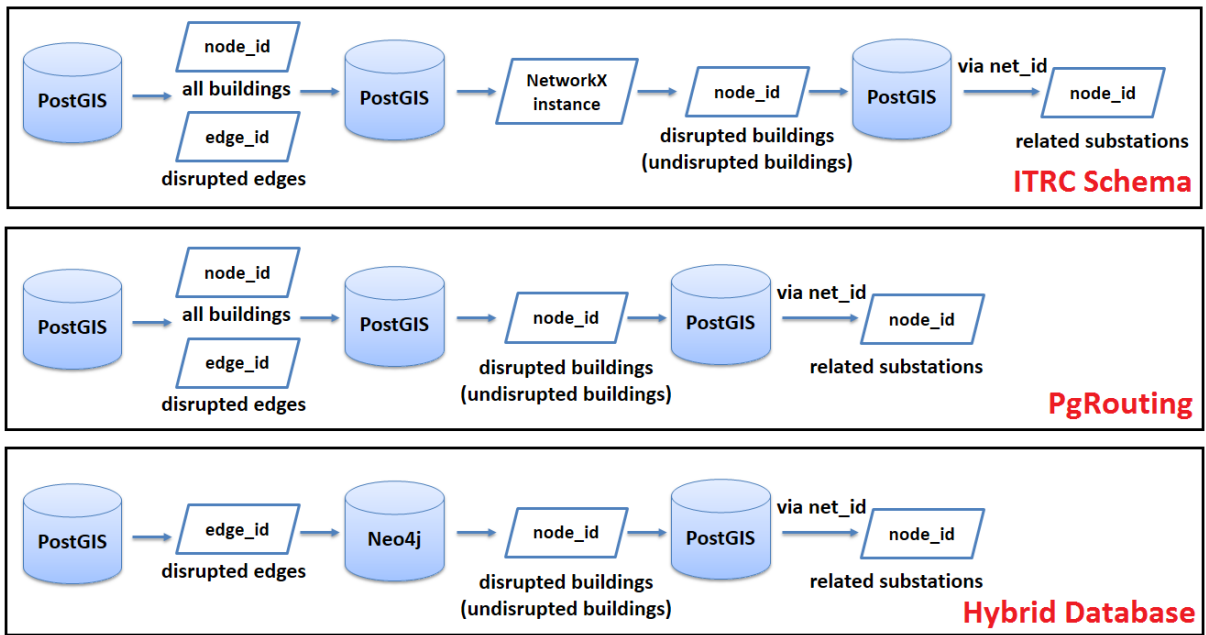
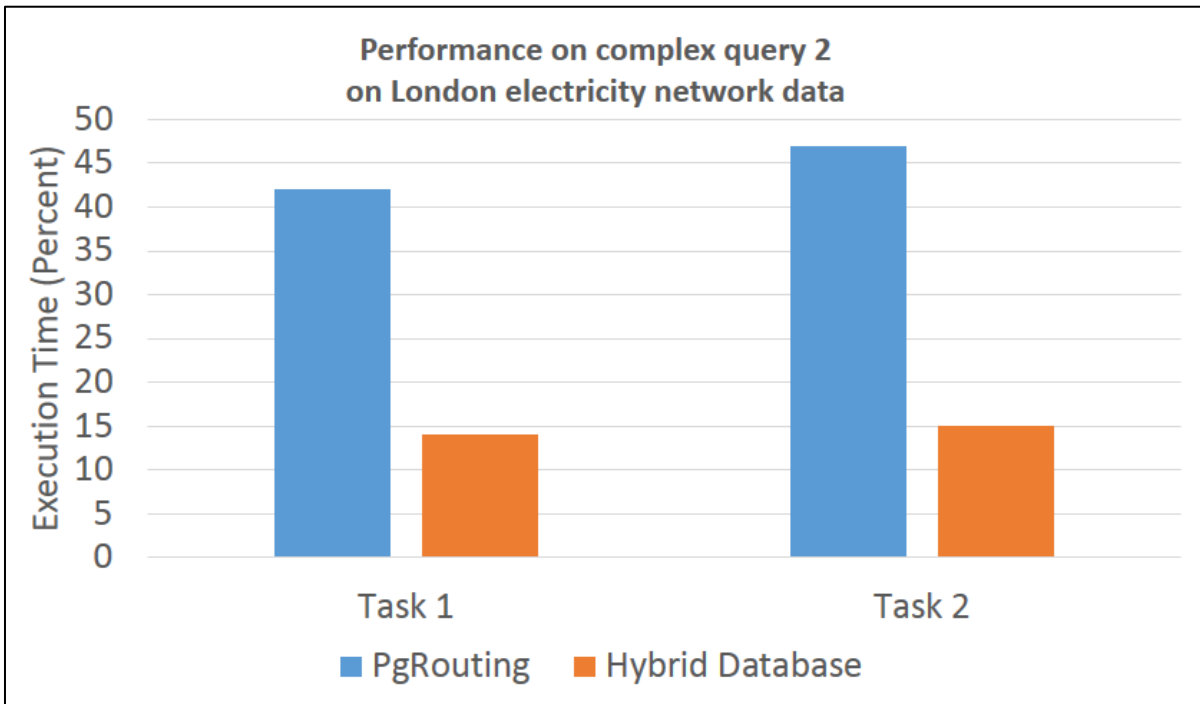


Figure 7.35. Pipe lines to resolve **complex query 2** on London electricity network data.

Task	Result (Avg No. Data Retrieved on Each Random Hazard)
1	Disrupted Buildings: 164, Substations: 2.6
2	Undisrupted Buildings: 2239049, Substations: 16836.4

Table 7.8. Result of **complex query 2** on London electricity network data.

Execution time to resolve complex query 2 is shown in table I7, Appendix I. The execution time for finishing task 1 for ITRC schema, PgRouting, and hybrid database are 21649, 9061, and 3125 seconds respectively. The execution time for finishing task 2 are 23155, 10793, and 3507 seconds respectively. A percentage performance comparison is shown in figure 7.36.



**Figure 7.36.** Performance comparison on performing **complex query 2** on London electricity network data.

The complex query 2 (either task 1 or 2) is virtually a spatial query followed by a network topology query and attribute query. It is designed by intention to see how efficient each database approach is, when it needs to handle three completely different types of sub-queries. Figure 7.36 indicates that the ITRC schema is the slowest one, and the biggest reason is that it is very poor at performing network topology query (reading data into NetworkX is time consuming, and NetworkX functionality is less effective than PgRouting and Neo4j). Due to this, PgRouting and hybrid database are about 2 times and 7 times faster than ITRC schema. Hybrid database is even about 3.5 times faster than PgRouting, which is because the efficiency of its graph engine (Neo4j) in resolving the network topology sub-query for such massive network data.

## 7.7 Conclusion

An efficient database approach is essential in managing and analysing complex geospatial infrastructure network data. This chapter focused on database performance benchmarking

tests on three candidate database approaches: ITRC schema (the benchmark), PgRouting, and hybrid database (combination of a PostGIS and Neo4j). Tests of different complexities are designed to evaluate performance of each approach, when processing different network data, or performing different operations on the data.

With regards to writing data, ITRC schema is always the most efficient one (regardless of network size). The hybrid database is about 1.5 – 2 times slower, due to its separate data storage system and it needs to interact with both PostGIS and Neo4j databases. The PgRouting is even slower, due to its long pipe line for writing and its difficulty in writing node attributes, and that makes PgRouting 2 – 2.5 times slower.

However, with regards to reading data, ITRC schema is the slowest one, because it must read data into NetworkX instance first. Hybrid database is about 2 times faster, since it does not rely on NetworkX library. PgRouting is the even slightly faster than hybrid database, since it only reads data from one database, instead of two.

Considering the fact that reading is a more frequent operation than writing in actual applications, the writing inefficiencies of PgRouting and hybrid database are totally acceptable as long as they read data much faster.

Except for reading and writing, another simple query that can occur frequently on infrastructure network data, is network query, such as shortest path query. In the tests, ITRC schema is the most inefficient approach for that, and major reason is that it must data into NetworkX instance before network query. PgRouting and hybrid database can both perform network query directly, which is why they are about 2 times and 5 times faster. PgRouting allows using SQL to directly query network data, but it is virtually still a join on relational tables, which is why it is still less efficient than hybrid database, which relies on Neo4j and its own property data model.

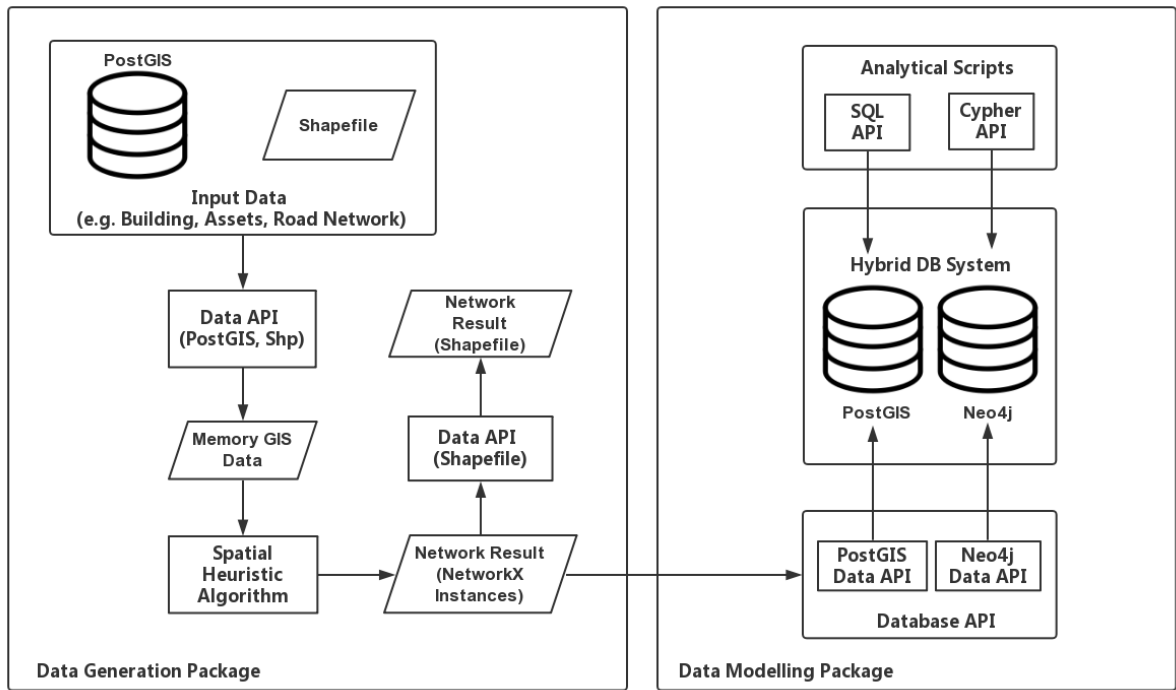
One of the potential problems of using hybrid database is its separate data storage, so that it

can be less efficient when only spatial or attribute query is performed. This is actually verified in section 7.5.2. Hybrid database can be about 1.2-1.4 times slower (than ITRC schema or PgRouting) when on performing spatial and attribute query on Newcastle integrated road network (IRN) data. However, as network size increases, hybrid database performs less poorly, and is about 1.15 – 1.2 times slower on London electricity network data. This is considered to be the better scalability of hybrid database (actually the Neo4j inside) at performing attribute queries.

Finally, section 7.6.3 shows that when performing all of the spatial, attribute and network query on network data of mega city size, the hybrid database is the most efficient one, which is about 7 times faster than the ITRC schema and 3.5 times faster than PgRouting. That indicates that as long as a network topology query is involved, the hybrid database is an efficient approach for handling large and complex network data.

Given the above, it is considered that hybrid database is the most efficient approach of the three, when managing and analysing geospatial infrastructure network data. Combining with other work from this PhD, a prototype platform (figure 7.37) is proposed for geospatial infrastructure network data inference and management. It consists of two major packages. One is the data generation package, to infer layout of infrastructure networks. The other is the data modelling package (based on a hybrid database) for managing and analysing infrastructure network data.

A potential drawback for using a hybrid database though, is that there are two databases that can be queried instead of one. When performing a complex query (e.g. which comprises of a spatial query and network topology query), it is currently the user (the human) that decides whether to first visit PostGIS (on the spatial query) or to first visit Neo4j (on the network or attribute query). The system is still not automatic enough, which is why figure 7.37 is only called a *prototype* platform. This is an interesting topic to explore and should be further studied as the future work on top of this PhD.



**Figure 7.37.** A prototype platform for geospatial infrastructure network inference and management.

## **Chapter 8. Discussion**

### **8.1 Introduction**

Having fine granularity geospatial data on critical infrastructure networks is essential in different digital urban models (Albaugh, et al., 2004; Fang, et al., 2016; Gabrys, 2014; Lara, et al., 2016; Malekpour, et al., 2016), for example, in order to understand infrastructure interdependency and cascading failure from infrastructure assets to the buildings (Ouyant, 2014; Rinaldi, 2001). However, until now relatively little attention has been paid to the representation and use of fine spatial scale infrastructure network data in infrastructure analysis, simulations and models. Chapter 2 (Literature Review) discussed the three major issues: the lack of an ontology when integrating data from different sources (section 2.5.1, page 21), the lack of a data inference approach to generate plausible spatial network layouts (section 2.5.2, page 26), and the lack of an efficient database to manage such complex geospatial network data (section 2.5.3, page 28).

This thesis addressed these issues by developing generic geospatial data management tools for fine spatial granularity spatial infrastructure network data.

### **8.2 Geospatial Infrastructure Network Ontology**

Data from different sources can be encoded in different ways, and to integrate such data in information system, a standard (ontology) is needed (Gruber, 1993). The ontology (in the context of this research) should conceptually define entities, attributes and relationships that to represent fine scale geospatial infrastructure networks. Section 2.5.1 (table 2.7, page 24) discussed about the related ontologies with regards to critical infrastructures. However, none of them meets the requirement for the management of fine scale geospatial infrastructure network data.

First, many existing ontologies only represent the infrastructure network at a topological level, ignoring the spatial level, such as KM4City (Bellini, et al, 2014), and Utility Knowledge Ontology (Xu, et al., 2018). Secondly, almost all the ontologies (except for INSPIRE data model) focus on a single sector of infrastructure network, such as transport (Lorenz, et al., 2005) or utility (Becker, et al., 2012), without considering all of them. Thirdly, all of the ontologies only represent infrastructures themselves, without considering the buildings and relationship between buildings and infrastructure; thus, that these ontologies cannot represent flows from assets to the buildings (D'Agostino, 2014). Finally, there is no ontology that considers dependencies and interdependencies within infrastructure networks. Some studies have developed an infrastructure interdependency ontology (McNally, et al., 2007; Sicilia, et al., 2009), but they only focus on the dependencies / interdependencies themselves, without integrating it to the infrastructure networks.

Identifying these major research gaps, the objective of developing a geospatial infrastructure network ontology is addressed in Chapter 3. The ontology (section 3.2, figure 3.1, page 33) is designed to cover all major types of infrastructure networks, including utility network (electricity, gas, water supply and waste water) as well as transport network (road, metro, rail). Each infrastructure network is defined as a spatial network instance, where each node or edge has its own geometry (section 3.2, table 3.2, page 34). By doing so, spatial relationships (for example, distance, north of, south of) can be represented on the infrastructure networks (section 3.2, table 3.3, page 35). The ontology also defines that attributes associated with edges and nodes. Table 3.4 (section 3.2, page 36) describes the inheritable node or edge attributes (such as Edge Type and Edge Length) that enable basic network analytical functionalities as suggested by Xu et al (2018).

Except for these inheritable attributes, each type of infrastructure network also has its own attributes (figure 3.8 on page 43, table 3.6 on page 44, table 3.8 on page 48). By referring to related infrastructure literatures, these sector-specific attributes are defined, which ensures stronger analytical and simulation capabilities on different types of infrastructure networks (for example, simulating electricity voltage drops on electricity network, or simulating water

pressure in the water supply network) (Northern Powergrid, 2017). The completeness of sector-specific attributes is something that is not covered in existing infrastructure network ontology, and is considered as one of the added-values of this ontology.

The inclusion of buildings (section 3.3, figure 3.4, page 38) and infrastructure-building connection (section 3.4, figure 3.5, page 40; section 3.5, figure 3.9, page 45) is considered as the major contribution in Chapter 3. The building ontology is mainly based on the work of Zhu et al (2015) and Swan et al (2009), but also proposes additional attributes for the building such as Utility Demand (Water Consumption, Electricity Consumption, etc.) that can characterise and represent the supply/demand relationship from infrastructure assets to the buildings. The ontology defines each Building has one and only one connection to one Utility Network (section 3.4, figure 3.6, page 41). While the connection between road network and building is represented as a many-to-one mapping relationship from a Building to a Road (section 3.5, figure 3.12, page 49), based on an approach proposed by Cavallaro et al (2014). With such representation, it is possible to represent and potentially understand how the resource, energy or service flows from infrastructure assets to the individual buildings occur, which is currently absent in existing research.

The ontology also proposed the representations of infrastructure dependencies and interdependencies (section 3.6, figure 3.13, page 50), which is essential in understanding and modelling infrastructure cascading failures (Ouyang 2012; Rinaldi, et al., 2001). While the focus of this PhD research is at a fine spatial scale (distribution level), *only* the dependencies or interdependencies *in the distribution level* are covered in the ontology. Therefore, there is no dependency from electricity generator from pumping station (generator needs water to cool down) (Ouyang, 2014) for example, since the generator belongs to electricity transmission network.

However, there is still considerable scope to further extend the ontological framework presented in Chapter 3. First and foremost, within this PhD research, this ontology is not fully implemented at a practical level. Due to the requirement of the work in Chapter 4, 5, 6, and 7,



infrastructure networks are only modelled as geospatial network instances with flow directions. Attributes discussed in section 3.4 (table 3.6, page 44) are not implemented (such as Resistance on Cable in Electricity Network). To implement this ontology at full practical level, additional data is needed to enable the full modelling capability of this ontology, which is beyond the scope of this PhD research. However, it is a very interesting topic to explore in the future. Cooperation with related infrastructure industry is needed to acquire these additional data, to extend the modelling capability of this ontology, and optimize it.

Secondly, spatial and temporal dynamics exist on infrastructure networks, and are discussed in section 3.2, in figure 3.2 (page 36) and figure 3.3 (page 37), where temporally and spatially transient attributes are defined based on a temporal and spatial reference system proposed by the INSPIRE data model (INSPIRE, 2013). However, the representation of spatial and temporal dynamics still remains at the conceptual level, without being practically implemented. From Chapter 4 to Chapter 7, every attribute on the infrastructure network data is treated as static attribute, and it is still unclear how to represent temporal dynamics in a database system. Understanding the dynamics of resource flow is essential in infrastructure analysis and simulation (Li, et al., 2013; Puig, et al., 2017), therefore future work would focus on the storage and management of spatial temporal dynamics of infrastructure network in a data information system.

Thirdly, representations with regards to building-infrastructure connection in this ontology is simplified. The ontology assumes a Building has one and only one connection to one Utility Network (for example, electricity) (section 3.4, figure 3.6, page 41), which can be inaccurate if the Building (such as a hospital) cannot afford to lose infrastructure service (Cimellaro, et al., 2010). In that case, the Building can have multiple connections to a Utility Network. Likewise, the ontology defines that each Building connects one and only one Road (section 3.5, figure 3.12, page 49), and this is spatially the nearest Road to the Building. The assumption that *residents of this Building will only choose to access the nearest Road* can be inaccurate, as there might be multiple Roads that can be accessed by the residents from the Building (for example, one Road at the front door of the Building and the other at the back

door). In both cases, more detailed information on the type and geospatial layout of the Building (where the doors/entrances are) is needed in order to deliver more plausible building-infrastructure connection.

Finally, the ontology focuses on representation of *tangible* infrastructure sectors (e.g. road network, electricity network), but there also exist infrastructure sectors that are *less tangible*, such as the wireless cellular telecommunication network, and pedestrian/cyclist traffic network. Due to the emerging 5G techniques, the wireless cellular network is playing a more crucial role in the smart city applications (Kamilaris, et al., 2018). Cellular network is special as the *last edge* in the network is wireless: a resident's mobile phone or a wireless sensor on a taxi directly sends or receives data to or from a transceiver (a cellular asset). As each transceiver only covers a specific service area (cell), if a resident or a taxi moves from one cell to another, it will loss connection with the old transceiver and establish connection with a new one. In other words, the cellular network topology *changes temporarily*. In order to represent such network, a possible adaption of the ontology could be to add the *time dimensionality* (Whiteback, et al., 2010) to the basic graph model (just like taking snap-shots on cellular network topology continuously, so that its topology at different time can be represented and recorded). Another classic *intangible* network is the pedestrian or cyclist traffic network. Pedestrians or cyclists still use the same infrastructure (road network) as vehicles, but they *cannot* use any road segment, if there is not sidewalk or bicycle lane on that road. A common way to model such networks is to represent them as part of road network (Lorenz, et al., 2005), by adding attributes like *pedestrian-access* or *cyclist-access* to the road segments. Moreover, when characterizing pedestrian or cyclist flow, flow density needs to be clear represented, as travellers always prefer to use street that has small density or less congested (Bezbradica, et al., 2019). Besides, concepts like *pedestrian speed* and *route* (the most time-saving path from a location A to B) are also important features in many urban mobility models (Das, et al., 2015), and should be also included in the ontology. These attributes also change over time (e.g. pedestrian route could change depending on current pedestrian flow in the city), and should be represented as temporal transient attributes (section 3.2, figure 3.3, page 37) in the ontology.

### **8.3 Inference of Spatial Infrastructure Network**

Accessing good quality geospatial infrastructure network data is the biggest challenge in order to model fine scale geospatial infrastructure networks (Bon, 2007; Fu, et al., 2008). Chapter 2 (section 2.5.2, page 26) reviewed this issue and identified the research objective to develop a generic approach for inferring layout of geospatial infrastructure network. This objective is a complicated one, and is addressed via three pieces of work: a generic spatial heuristic algorithm, algorithm transferability to different utility networks, and a road network generation algorithm. The methodologies, results and findings are discussed in section 8.3.1, 8.3.2, and 8.3.3, respectively.

#### ***8.3.1 Generic Spatial Heuristic Algorithm***

A review was undertaken on related approaches on automatic generation of geospatial network layout (section 4.1, table 4.1, page 57). It is found that no approach exists in generating spatial layout of infrastructure network that connects assets and buildings. Most related approaches (Hadas, et al., 2013; Heijnen, et al., 2014) require location seed/origin nodes (intersection of cables/pipes) to be known. But such information is not available in our case. However, the studies from Bon (2017) and Cavallaro et al (2014) revealed that the layout of an infrastructure network is related to the layout of road network. Moreover, since it is an NDP (Network Design Problem), a constraint or an objective function is normally needed (Magnanti, et al., 1984). As suggested by Larkevi (1985), the constraint in this context, should be *keeping the network as short as possible*.

Following this rationale, Chapter 4 proposed a new and a generic spatial algorithm, that can infer geospatial layout of infrastructure network, based on layout of assets, buildings, and road network (section 4.1, figure 4.1, page 61). As the major innovation of this algorithm, it can generate geospatial network that contains topology, geometry (of the nodes and edges), the node and edge type (section 4.3.2, figure 4.14, page 74), and network flow direction from infrastructure assets to buildings. The algorithm is also scalable (regardless of input data size).

The algorithm is tested and validated for generating a plausible spatial layout of electricity distribution networks for the city of Newcastle upon Tyne (section 4.5, page 76), a network consisting of more than 200,000 nodes and edges. Validation (based on Northern Powergrid data) revealed the high spatial accuracy on feeders (around 89%) (section 4.6.1, page 83), and low average difference angles on service lines (around  $17.3^\circ$ ) (section 4.6.2, page 85). This indicates the algorithm can generate a feasible layout of infrastructure network that is relatively close to the actual data.

However, currently there are several limitations in the algorithms. First, the validation on feeders (section 4.6.1, figure 4.21, page 83) reveals that the actual feeders follow only on *one side* of a road. But as the algorithm uses ITN network (a polyline file), the algorithm will produce the synthetic feeders that follow the centreline of a road. This discrepancy (between synthetic and actual feeders) does not affect network topology, but will become a problem if high geospatial accuracy on the feeders is needed for specific applications (for example, when electricity failure occurs, electricity provider needs to locate the problematic feeder on map, and send technical teams to repair it). Future work would focus on this issue, and algorithm would need modification so that it reads ITN network together with the *road polygon layer* as input. A possible solution would be to link each road segment in the ITN network with a road polygon in the road polygon layer (using spatial relationship *contain*), so the geometry of the synthetic feeder can be modified using a road polygon layer. However, there is still an interesting question: as a road has *two sides*, how to decide which side of the road should a feeder follow? Potentially, cooperation with electrical engineering research teams is needed, with regards to the *feeder layout planning* in the perspective of electrical engineering.

Secondly, the algorithm requires the location of infrastructure node assets to work. If such information is not given, then it is impossible to infer the geospatial layout of infrastructure networks. This could be a potential problem, as it is not always possible to access layout of assets (at least good quality assets layout). Future work should investigate the possibility of inferring layout of infrastructure assets (based on layout of buildings, and road network for example). One possible starting point, would be using cluster approach to identify the

appropriate locations to plan infrastructure assets (Rui, 2013).

Finally, the algorithm is proved to be scalable in section 4.7. But algorithm can be computationally very expensive as input data volume increases. The algorithm time complexity is  $O(N_b^2)$ , where  $N_b$  is the number of buildings in the input data (page 96). Even for a good desktop workstation, it needs to spend more than 12 days to generate the Greater London electricity distribution network (figure 4.30, page 92). This spatial heuristic algorithm is designed as a generic data inference approach for any city (regardless of size), therefore the processing time on the Greater London data is considered too much. Future work should focus on improve processing speed (such as partitioning the input data and using parallel or distributed computing techniques).

Following the current implementation of the algorithm, there are still plenty rooms of optimizations for the future work. These are mainly related to three aspects and are discussed as follows: (a) Extendibility of generic algorithm for other countries/areas outside of the UK. (b) Adaption of generic algorithm based on input data it can receive. (c) The general philosophy behind the building – infrastructure planning process.

Firstly, despite the validation (section 4.6), and transferability test (section 4.7) of the generic algorithm, it is not clear whether such algorithm can still perform relatively well for *other* countries or areas, outside of the UK. For example, Hong Kong is a city that has 4<sup>th</sup> largest population density in the world (density of 6777/km<sup>2</sup>) (World Bank, 2019), even larger than UK's most populated city Greater London (density of 5590/km<sup>2</sup>) (Trust for London, 2019). The high population density in Hong Kong, and small area, lead to a very special *stacked* building architecture, the skyscrapers. In Hong Kong, there are 341 buildings taller than 150 meters, and an apartment building (which may have *thousands* of residents) can be as tall as 2500 feet (Skyscraper Centre, 2019). That can cause issues for the generic algorithm, as it is currently a capacity-free algorithm. But when inferring network layout in city such as Hong Kong, capacity and demand is not neglectable (as a skyscraper of thousand residents obviously requires *much more* infrastructure resource than a normal residential building that

has two or three floors) and further develop the algorithm based on that. Another problem related to extendibility, is that it may not have good performance in less developed areas, such as the slums of Nairobi, Kenya, and Rio de Janeiro, Brazil. The algorithm requires that buildings have *different* topological features (detached, terraces, etc), and uses road geometry and building topology to assign to assets to buildings. But in these less developed areas, buildings are less formally structured (*shanty town*), and that may cause *all the buildings* to topologically connect with each other in an entire slum (BBC News, 2019). The current algorithm would not be able to assign an asset to each building in this case, and a possible future adaption, is that asset assignment is done directly using Euclidean distance (each building is assigned an asset, that is nearest in Euclidean space), without using road layout at all. Lastly, the algorithm can only deal with 2D network layout, and this can be inaccurate if network needs to couple with a 3D city model for a city with steep terrain (Becker, et al., 2011), such as Lucerne, Switzerland, where there is around *400 meters* height difference in the city terrain (Wikipedia, 2019). In order to generate plausible network layout in such city, digital terrain model needs to be considered to generate a plausible 3D network layout.

Secondly, the algorithm is developed in a *generic* way, that only layout of buildings, assets, and road network are required as input. The algorithm still has potential to be further optimized, given more relevant input data (in other words, more constraints). For example, building age is an important feature to consider, when the local utility company (such as Norther Power Grid) plans for infrastructure layout. In fact, old buildings are more likely to be served by old infrastructure assets, than by the young ones (Schiller, 2007). If age data (of buildings and assets) becomes available, then topology generation process (that assigns each building an asset) (section 4.3.1, page 63) can adapt and possibly produce more accurate layout. To add the *age* constraint, a possible modification of the algorithm could be that, first dividing the buildings and assets to a number of age groups (For example, there are 3 groups. The 1<sup>st</sup> group represents buildings or assets which are younger than 20 years, the 2<sup>nd</sup> group represents those whose ages are between 20 and 50 years, and the 3<sup>rd</sup> group represents those that are older than 50 years), and apply the current algorithm to each age group. Except for age, adding demand / capacity constraint on the buildings and assets (as mentioned in the last

paragraph) is also a viable future option (Baskan, et al., 2014). These are the optimizations related to the topology generation process (i.e. assign the asset to buildings). There are also optimizations related to the geometry generation process (i.e. generate geometry of cables or pipes) (section 4.3.2, page 67). For example, having more geometry information on the buildings would be beneficial, as the front-door, back-door or the utility meter location can affect how the infrastructure pipe or cable connects each building (Avi, et al., 2014). Another example would be the adding spatial constraint (Heijnen, et al., 2014) about where the cables or pipes cannot cross (e.g. river, greenspace, restricted area). Additional geometry checks can be added in the algorithm to enforce such spatial constraint, so that cable or pipe geometry cannot intersect any constrained area, and if intersection occurs, cable or pipe needs to re-route (maybe *around* that constrained area).

Finally, the generic algorithm relies on a *complete* layout of buildings to produce infrastructure layout. It assumes that buildings *exist before* the infrastructure networks. Such assumption could be somewhat arbitrary, and it slightly simplifies the true urban planning process. In fact, as suggested by Parish et al (2001), Teoh (2007), and Rui (2013), planning the layout of buildings and infrastructure networks can be a complex and *iterative* process. When given a blank urban area, a common and ideal planning strategy would be first deciding the city centre areas (residential, industrial, etc) and generating major transport infrastructure network connecting them, then using major transport infrastructure network to constrain the space to generate buildings, and then using the building layout to generate finer transport infrastructure network, and so on. Therefore, a possible adaption of the algorithm would be to see the problem in the other way, that is to say, given a predefined network layout, is it possible to infer (or plan) a plausible building layout? It is also worth exploring the possibility to adapt this algorithm as an iterative process, so that it can *truly* reflect process for planning infrastructures together with the buildings.

### **8.3.2 Algorithm Transferability in Different Utility Networks**

Chapter 4 demonstrated the application of generic spatial heuristic algorithm in electricity utility networks. However, algorithm is designed to solve the worse scenario (completely missing layout of any pipes/cables). It is not clear, how the algorithm should deal with situation, when the network layout is already *partially available*. Moreover, the algorithm only focuses on the inference of geometry and connectivity of the network, but sometimes, additional attribute (especially resource flow direction) would need to be inferred as well. Chapter 5 investigated these two issues, and the major contribution is to improve the capability/transferability of the generic spatial heuristic algorithm. The improvement of capability/transferability is demonstrated in two aspects.

First, the algorithm is now capable of generating network layout based on partially existing layout of infrastructure network (section 5.2, page 100). It is demonstrated by completing the gas main pipe network based on data provided by NGN. NGN data contains layout of gas main pipes, except for the new development sites (section 5.2.2, figure 5.5, page 105). By consulting the Northern Gas Network, a Gas Network Infer Algorithm was developed (based on CSEP nodes and road network) (section 5.2, listing 5.1, page 105) so that layout of infrastructure main pipes/cables can be inferred in new developing areas, and are integrated to the existing network layout. The high spatial accuracy (around 92%) from validation (section 5.2.5, page 119) indicated the inference is plausible. This demonstrated the algorithm capability to infer network layout based on existing network (instead of almost nothing, in Chapter 4).

Secondly, the algorithm now has the capability of inferring network flow and this is demonstrated in water-related infrastructure network (water supply network, and sewer network). Technically, inferring water flow on these networks requires resolving full hydraulic equations (Preis, et al., 2010). But this can be computationally very expensive, and would not work if necessary attributes are missing (for example, water pipe diameter, location and state of the valves, network topology, etc.) (Giustolisi, et al., 2011). To address the lack of



water flow in water supply and sewer network, two spatial heuristic algorithms were proposed. Water Flow Infer Algorithm (section 5.3.2, listing 5.3, page 127) can infer the WDAs (water distribution areas) and water flow on the water supply network, based on layout of water supply network and water sources (service reservoirs). Sewer Flow Infer Algorithm (section 5.4, listing 5.6, page 138) can infer waste water flow, based on layout of sewer network, outflow nodes, and a DTM layer, which achieved high accuracy (96%). These two flow-infer algorithms extend the functionality of the generic spatial heuristic algorithm, so that it is able to not only infer the spatial layout of utility the infrastructure network, but *also additional plausible attribute on the network*, without needing to run computationally expensive mathematical models.

There are also potential room for optimizations and future work. First, the transferability of the algorithm is explored based on data from the city of Newcastle upon Tyne. To evaluate the algorithm transferability in terms of *different cities*, more case studies (using data from other cities) are necessary. Secondly, the algorithm can now infer additional attribute on the utility network (resource flow), and this capability needs to be further explored, if some other important attributes are missing and are required for specific applications (for example, pressure (of gas and water) is an important attribute to assess potential pipe failure in the network). In order to do so, more actual data from utility companies are required, in order to develop approach that can plausibly infer these attributes.

### ***8.3.3 Road Network Generation Algorithm***

One potential limitation of the algorithm developed in Chapter 4 and 5, is that it must rely on a road network to work. If road network layout is missing, then it is not possible to infer layout of infrastructure networks. This is actually an issue, in new development areas, road network layout is not always present during the master planning phase, where land use (layout of residential buildings, water bodies, factories, park, etc.) is decided (Moss, et al., 2016). Therefore, Chapter 6 extended the work in Chapter 4, and 5 by exploring the approach to infer

road network based on building layout. The major contribution of Chapter 6, is that it proposed a new road network generation algorithm, that can be either applied as a plausible tool for road network planning (for example, in new development areas), or can be applied together with generic spatial heuristic algorithm, in order to infer plausible utility infrastructure network layout, if road network layout is missing.

Section 6.2 reviewed existing approaches for automatic road network generation (section 6.1, table 6.1, page 153). However, none of them considers the existing layout of buildings. For the most related approaches, the L-system based algorithms do not consider building layout (Parish, et al., 2001; Teoh, et al., 2007), while the rectilinear Steiner tree-based algorithm, proposed by Nie et al (2010) requires *seed nodes* (road junctions) to be known already.

Chapter 6 addressed this research gap, by proposing a new generic road network generation algorithm, based on layout of buildings, boundary, and entry points as input (section 6.3, figure 6.3, page 155). By observing real data (road network and buildings), it is found that buildings form clusters spatially and each cluster of buildings is surrounded by roads. Following this rationale, the road network generation algorithm first finds clusters on the buildings (based on MST partitioning algorithm proposed by Zhou et al (2009)), then performs constrained Delaunay triangulation to indicate the space where road segments can be generated.

This algorithm relies on an important parameter  $\epsilon$ , which controls when to stop the MST partitioning process (section 6.4.1, page 158). This parameter  $\epsilon$  is tuned based on the small case study area (contains about 550 buildings) in Newcastle upon Tyne. Through parameter sensitivity test (section 6.6, table 6.5, page 174), it is found 0.0075 is an appropriate value for  $\epsilon$  to generate the most plausible road network in the case study area (accuracy is around 94% compared with ITN data). The transferability test (section 6.7, page 176) shows the algorithm and parameter  $\epsilon$  (value 0.0075) *do not over-fit* to small case study area, and generate plausible road network (accuracy around 95%) in other areas in Newcastle with different building layout and building density. This justifies the application of MST

partitioning algorithm (and more importantly, choice of  $\epsilon$  value) in generic road network generation problem.

It is essential that the road network generation algorithm can infer *plausible* road network (as proved in table 6.7, page 181). An interesting question is that, will the *usage of synthetic road network* cause any major difference in generating utility network layout? Section 6.7 proved that, even using the synthetic road, the synthetic electricity distribution networks still have very high accuracy (95% - 99%) compared with reference networks (table 6.9, table 6.10, page 189). Therefore, it is considered this algorithm is not only able to infer/plan plausible road network layout, but also able to work together with generic spatial heuristic algorithm to *still generate plausible utility network layout*.

However, two limitations are observed in the road network generation algorithm. First, despite the high accuracy, discrepancy still exists between the synthetic and real road network (section 6.4.4, figure 6.13, page 165). The main reason is that, actual road segments can partially surround a building cluster, but the algorithm must assume the building cluster is *entirely* surrounded by roads. Secondly, the algorithm cannot generate road segment at the boundary areas. Constrained Delaunay triangulation is the reason. For any point A (centroid of a building) that is already on the boundary area, there is *no outside point* that can make triangulation with point A, which means on the outside of point A, it is impossible to generate the geometry of a road segment. Therefore, the boundary (which is considered as the exterior ring of the road network) must be given.

#### **8.4 Database Approach for Management of Spatial Network Data**

In an infrastructure information management system, an efficient database is an essential part in handling wide range of disparate data and relationships required for infrastructure systems modelling and analysis (Robson, et al., 2018). However, little attention has been made on the database system for handling fine spatial scale infrastructure network data. Chapter 7

investigated this problem and proposed employment of a Hybrid Database approach (a combination of PostGIS and Neo4j), based on the result of database performance benchmarking tests.

The tests compared three database approaches: PostGIS/ITRC, PgRouting, and Hybrid Database. The former two are the traditional solutions, which are spatial relational databases. The last one is based on Neo4j, a popular NoSQL database that has been applied in more efficient management of network data (Cattuto, et al., 2013; Lin, et al., 2017; Yoon, et al., 2017). However, as discussed in section 7.2.3 (page 202), Neo4j itself does not have enough capability of encoding or querying geospatial data. This is why Hybrid Database architecture is proposed (Neo4j for encoding non-spatial attributes and network topology, and PostGIS for encoding geometry).

The benchmarking tests indicate the Hybrid Database is less efficient than PostGIS/ITRC at writing data (could be 140% to 150% slower) (figure 7.13, page 211; figure 7.32, page 232). This is due to different data model (property graph) and data driver used in Neo4j. However, the Hybrid Database is more efficient at reading network data, which is about 220% faster than PostGIS/ITRC (figure 7.13, figure 7.32). Considering the fact that reading data is a more frequent operation than writing in real life applications, the underperformance of writing for Hybrid Database is acceptable.

As a major contribution, the benchmarking tests indicate that as long as network topology is involved in a query, Hybrid Database is always at least 2.4 times faster than PostGIS/ITRC and PgRouting (figure 7.27, page 226; figure 7.36, page 238). When performing a network topology query only, such performance difference will become greater: when resolving shortest path query, the Hybrid Database is between 5 to 12 times faster than PostGIS/ITRC and 2.1 to 4.3 times faster than PgRouting (figure 7.17, page 216; figure 7.32, page 232). This is due to Neo4j's natural strength at querying large and complex network on topology. For PostGIS/ITRC, it needs to read network data into NetworkX instance to be able to perform topology query, which can be very time consuming when network is large (figure 7.2, page

198). For PgRouting, it can query network data without external library, but this operation is still virtually relational-joins on tables, which is still less efficient than Neo4j's property graph model (figure 7.6, page 204).

Due to its efficiency, especially at performing network topology query, the Hybrid Database is proposed as an appropriate database approach for the management of fine spatial scale infrastructure network data. However, there is a major limitation of applying it in actual applications. The split storage of data is the problem. Although figure 7.7 (section 7.2.3, page 205) shows that it is possible to link (data) between PostGIS and Neo4j using `node_id`, and `edge_id`, it still becomes an issue *when both databases need to be visited in a long and complex query* (such as finding nodes within a given spatial footprint, and then resolve topology queries from these nodes).

Recent research has started to tackle such “split storage” issue and a feasible solution would be to use a federated database architecture (Robson, et al., 2018), in which there is a master database (instead of the user) that decides how to decompose a complex query into small sub-queries, and visit the databases accordingly. For example, a federated database framework NISMOD-DB++ was developed based a similar idea (split storage of geometry and topology data using PostGIS and Neo4j), to manage and analyse geospatial infrastructure network data, and it employs a PostgreSQL database as a master database (Robson, et al., 2018). Therefore, a possible future work of this PhD would be to continue exploring the federated database architecture to manage geospatial infrastructure data.

## **8.5 Application of infrastructure data inference and management**

A prototype platform for infrastructure network data inference and management, was put forward at the end of Chapter 7 (page 241), as the final output of this research. While the previous four sections (8.1, 8.2, 8.3, and 8.4) focus on result assessments and key research findings, this section (8.5) discusses the possible applications of such platform in

infrastructure data inference and management. It is considered that this piece of work could be beneficial to different user groups who are concerned with geospatial infrastructure network data. They are (a) utility companies, (b) urban planners, (c) normal residents, and (d) scientific researchers and infrastructure committees, and major potential applications are discussed as follows.

For the utility owners, the platform helps to locate their infrastructure and to understand infrastructure demand / supply. Firstly, the platform can help them map their infrastructure cables or pipes, as in many cases, they themselves do not always have good quality geospatial data of their infrastructures (Jaw, et al., 2013). The network maps help these companies avoid digging unnecessary holes on the ground to repair buried infrastructures (Fu, et al., 2008). The generic algorithm has achieved relatively high spatial accuracy (89%) (section 4.6.1, page 84). But its accuracy still needs to be improved to better serve such utility companies. For example, optimization needs to be done, so that electricity feeders no longer follow the road centre line, but instead follow only one side of the road (figure 4.21, page 83). Furthermore, the generic algorithm should adapt from 2D to 3D space, so that it can provide additional information (such as how deep the infrastructure is buried). Secondly, understanding demand and supply between infrastructure assets and residential buildings is essential in many smart city applications such as smart neighbourhood (Lara, et al., 2016; Piotrowski, et al., 2014), and metering studies of local energy distributions (Albaugh, et al., 2004; Kleissel, et al., 2010), and this platform helps utility company model and understand such demand / supply relationships. By generating the geospatial layout of infrastructure network, demand / supply can be characterised by network topological connectivity (e.g. which asset connects which buildings). However, there are still some limitations to accurately represent demand / supply and they should be addressed in future work. The algorithm is a capacity-free algorithm (as explained in section 8.3.1) and introduction of capacity (of infrastructure assets) would add more constraint to the algorithm to improve accuracy. Similarly, different types of buildings have different demand level (related to number of floors of the building, the age of the building, etc) (Blokker, et al., 2009; Nouvel, et al., 2015), and it should be addressed in future work if accessing such data is possible.

For the urban planners, they would benefit from the capability to design automatic infrastructure network layout (especially the road network generation algorithm discussed in Chapter 6). Such road planning algorithm has achieved high accuracy (94%) in different validation areas in Newcastle, and it is considered such accuracy is already high enough for the urban planners. However, as mentioned in section 8.3.1, urban planning is a complex and iterative process, so that the capability (instead of accuracy) of the algorithm needs to be further improved (e.g. plan building layout given road layout) to better serve the urban planners in different application scenarios.

For the urban residents, they are mostly concerned about whether or not infrastructure service to their houses are disrupted (Glenis, et al., 2017). The prototype platform already has the capability to characterize flow from infrastructure assets to individual buildings, and can locate (query) the affected buildings, if disruptions occur on the infrastructure networks. However, to better serve the urban residents, web-based visualization (or any reporting) tools (Sabeur, et al., 2016) would need to be developed on top of the platform, so that infrastructure disruption events (in terms of buildings) can be easily reported and understood by the urban residents.

Finally, for the scientific researchers and infrastructure committees, they are mostly concerned about the urban infrastructures at systematic level, and there are three major types of applications. Firstly, such platform is a generic framework to characterize infrastructure resilience (Cavallaro, et al., 2014; Leu, et al., 2007) in the perspective of graph models. Generic graph operations (e.g. degree calculation, clustering coefficient calculation) are directly supported, which can be used as basic metrics to evaluate network resilience (Berche, et al., 2009; Murray, 2006). But these operations need to be done via Neo4j Cypher queries, and possible future work would be to develop APIs or UIs for the users to retrieve such resilience metrics more easily. Secondly, as urban infrastructure becomes more complex and vulnerable, having a systematic data and information approach to represent the multi-sector urban infrastructures is essential in understanding the infrastructure dependencies / interdependencies (Pant, et al., 2016; Zimmerman, et al., 2017). Such representation of

interdependent networks is also supported by the platform. By employing the graph database architectures, interdependency queries can be simply resolved as Neo4j Cypher queries. Moreover, the platform has the capability to infer the dependency / interdependency, as long as assets layout is present (figure 5.47, page 148). However, the prototype platform currently only focuses on *distribution* networks (thus only dependency / interdependency at distribution level), but there also exist dependencies and interdependencies at *transmission* level (Avi, 2014; Vickridge, 2004). Therefore, future work would expand the platform functionality to transmission level and represent other dependencies and interdependencies accordingly. Thirdly, infrastructure network models are normally coupled with spatial hazard models to assess how extreme spatial events (such as flooding and earthquake) affect infrastructure networks and trigger failures (Glenis, et al., 2017; Pant, et al., 2014). Such capability is demonstrated by section 7.5 (page 217), in which flooding impact on road and electricity networks are analysed and evaluated. However, the efficiency (instead of capability) needs to be improved to reduce query time (e.g. in section 7.6.3, spending hours to evaluate flooding impact on electricity networks in Greater London is still too slow, even if it is a massive city). For example, if the same network needs to be queried in many different parallel scenarios (e.g. the test discussed in section 7.6.3, page 233), it would be a better approach to parallelize the platform so that it can be deployed on different computers or clusters to significantly speed up computation (Abuzalaf, et al., 2016).

## 8.6 Summary

This PhD contributes to the development of generic approaches for the inference and management of high granularity geospatial infrastructure network data. Plausible geospatial layout of fine spatial scale infrastructure network can be now generated via the generic spatial heuristic algorithm (Chapter 4, 5, and 6). Spatial connectivity between infrastructure assets and buildings is represented and resource flow is characterized. The Hybrid Database (Chapter 7) is proposed as an efficient data management tool on such complex network data. All of these open up opportunities in applying the fine granularity infrastructure network data



in different digital urban models and applications, such as smart sensing (Gabrys, 2014), metering studies of local energy distributions (Kleissel, et al., 2010), digital twins (Mohammadi, et al., 2017), and infrastructure interdependency and failure model at fine spatial scale (Ouyang, 2014).

## **Chapter 9. Conclusion**

### **9.1 Introduction**

The aim of this research is to develop generic approaches for inference and management of fine spatial scale geospatial infrastructure networks, which opens up opportunities for different digital urban models and applications at fine granularity. The following objectives were set out to address the aim:

1. Review the research field pertaining geospatial urban infrastructure network models and identify the research gaps in the inference and management of complex infrastructure network data.
2. Develop a geospatial ontology, to conceptually model the knowledge of the entities, attributes and relationships that are indispensable to represent fine scale urban infrastructure networks. The focus is to understand the spatial connectivity between infrastructure assets and buildings.
3. Develop an approach, to infer geospatial layout of the utility infrastructure network if actual data does not exist or only partially exists. The approach should be transferable so that it can be applied in different major utility sectors (electricity, gas, water supply and waste water).
4. Develop a database approach that is able to encode, manage, and query the complex geospatial infrastructure network data in an efficient manner. Several potential database approaches will be investigated, and performance benchmarking tests will be carried out to decide the most appropriate one.

### **9.2 Research Summary**

Objective 1 was achieved by performing an extensive review of related literature (Chapter 2) in the field of geospatial infrastructure network models. Chapter 2 highlighted the importance

of the geospatial data in fine scale infrastructure network models, such as in smart city sensing (Gabrys, 2014; Hancke, et al., 2013; Perera, et al., 2014), smart metering and neighbourhood (Lara, et al., 2016; Piotrowski, et al., 2014), assessing impact of geospatial event on critical infrastructure network (Cabinet Office, 2008; Leavitt and Kiefer, 2006), and infrastructure planning and decision support (Gurung, et al., 2015; Malekpour, et al., 2016). Chapter 2 also identified the key challenges in this field: the lack of a geospatial ontology, the lack of generic data inference approach (when accessing real data not possible), and the lack of an efficient database approach for the management of complex geospatial infrastructure network data. These challenges lead to the development of Objectives 2, 3, and 4, which are addressed in Chapters 3, 4, 5, 6, and 7 respectively.

Chapter 3 addressed Objective 2, by proposing a geospatial ontology that represents fine scale urban infrastructure networks. This ontology covers major critical infrastructure networks (utility and transport), and it defines an infrastructure network as spatial network instance where attributes are associated with nodes and edges. This ontology employed knowledge from INSPIRE data specification of utility and transport network (INSPIRE, 2013), OTN (Lorenz, et al., 2005), and Utility Knowledge Ontology (Xu, et al., 2018). However, as a major innovation, this ontology represents the building-infrastructure connections and infrastructure network dependencies and interdependencies, which are missing in any existing infrastructure ontology. By reviewing related literature, this ontology identifies the key attributes for different types of infrastructure networks. This ontology is aimed as a generic data modelling approach to represent, analyse and simulate the spatial connectivity and resource and service flow from infrastructure assets to the buildings they service.

Objective 3 is addressed in Chapter 4, 5, and 6. First a generic spatial heuristic algorithm is proposed in Chapter 4, which infers the geospatial layout of infrastructure networks, based on the layout of infrastructure assets, buildings, and the road network. The algorithm is developed mainly based on the assumption that *infrastructure network follows along or very close to the road network*, as suggested by Bon (2017), Cavallaro et al (2014), and Larkevi (2005). This algorithm was demonstrated via generating the electricity distribution network

for the city of Newcastle upon Tyne, and validation indicated synthetic network layout is plausible (accuracy around 89%). The algorithm's scalability was also investigated by generating electricity networks for different cities (of different sizes) in the UK, and the largest synthetic network is generated for Greater London, which contains more than 4 million nodes.

Chapter 5 extended the work in Chapter 4, by investigating the algorithm transferability, when applied in other utility sectors (gas, water supply, and waste water). For these three types of networks, network layout is partially available, so the algorithm is extended in a way that it can integrate existing network layout and infer network flow if it is not available. Both algorithms developed in Chapter 4 and 5 depend on a road network to function properly. Therefore, Chapter 6 further explored the data inference problem, by proposing a road network generation algorithm, if the layout of buildings is available. This algorithm is developed based on reviewing related literatures and observations of real road network. The algorithm first employs an MST partitioning approach (Zhu et al., 2009) to generate building clusters spatially, and algorithm generates roads that surround each cluster. It was tested and validated to generate plausible road network layout (commission and omission error around 3% to 8%) in different areas for the city of Newcastle upon Tyne. As the major contribution, Chapter 4, 5, and 6 collectively proposed a generic data inference approach to infer fine scale infrastructure network layout. It is scalable (regardless of city size) and transferable (regardless of utility type), and can generate synthetic network that contains geometry, connectivity, type (on the nodes and edges), and flow direction, which delivers basic spatial network analytical capabilities.

Objective 4 is addressed in Chapter 7, and this Chapter investigated whether or not a traditional database approach (spatial relational database such as PostGIS) is still efficient in managing and querying the complex fine scale geospatial infrastructure network, compared with NoSQL database (Neo4j). Database performance benchmarking tests were designed, and database performances were compared and evaluated. Three database approaches were involved, which are PostGIS/ITRC, PgRouting, and Hybrid Database (PostGIS + Neo4j). It is

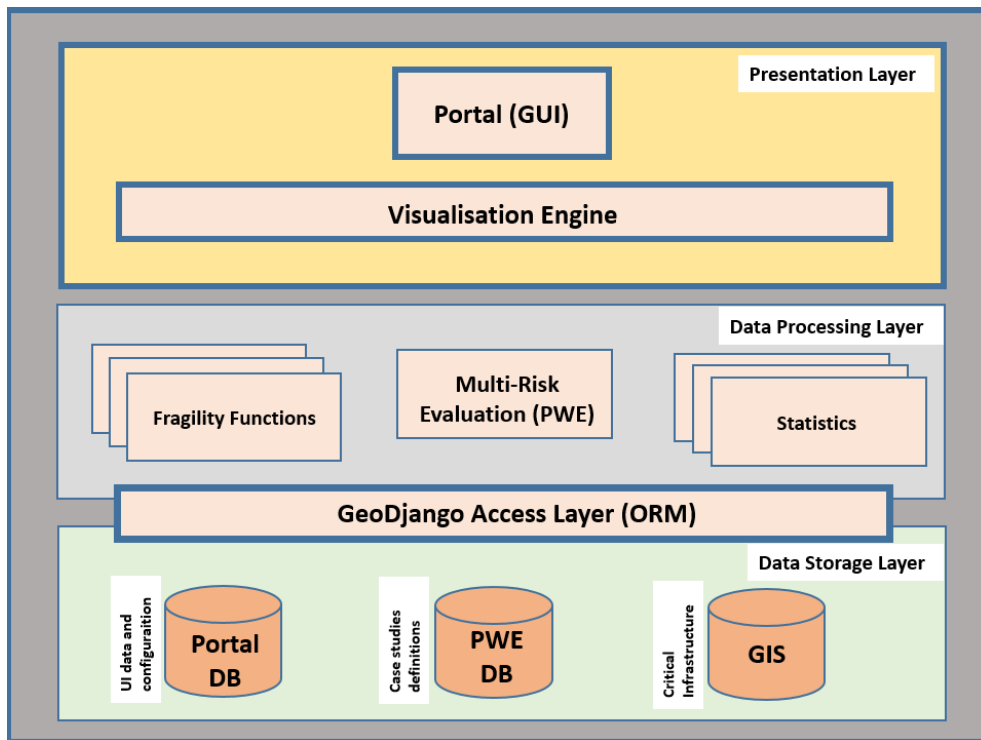
found that, despite being relatively inefficient at writing data (about 1.5 times slower), the Hybrid Database is much more efficient than the other two approaches (between 4.2 to 11.7 times faster) when performing network topology queries. The efficiency of Hybrid Database is due to the property graph data model employed in Neo4j, which is shown to be more efficient than relational tables, when encoding and querying large and complex spatial network data. As a result, Chapter 7 proposed the Hybrid Database for the management of fine spatial scale infrastructure network data, over the traditional database approaches.

### **9.3 Future Work**

The thesis proposed a generic approach for the inference and management of fine scale geospatial infrastructure network data. However, limitations exist and need to be addressed in future work, mainly discussed in section 8.2, 8.3, and 8.4. In this section, a number of future research directions are also discussed and evaluated.

#### ***9.3.1 Critical Infrastructure Decision Support***

A decision support framework or system is essential to stakeholders and decision makers, to better assess infrastructure vulnerability, analyse infrastructure failures and disruptions and provide suggestions for infrastructure planning and fortification in the long run (Kiel, et al., 2016; Rosato, 2015; Wang, 2013). In general, such system is based on three-layer architectures, the data storage layer, the data processing (simulation) layer, and the presentation layer (Mascucci, 2016; Sauber, et al, 2017; Wang, 2013). For example, Figure 9.1 shows the architecture of the integrated decision support information system developed by Sabeur et al (2016) for assessing impact of extreme hazards on the critical infrastructures.



**Figure 9.1.** Architecture of three layer decision support system developed by Sabeur et al (2016).

The data storage layer consists of database systems to efficiently encode infrastructure network data. The data processing layer normally consists of analytical and simulation programmes depending on specific needs (for example, models that can evaluate impact to critical infrastructure from extreme natural hazard). The data presentation layer is designed to render and visually report model result (for example, vulnerable or disrupted infrastructure assets) to the users.

In this PhD research, the focus is only on the data storage layer, where a Hybrid Database is proposed as a data management system. Moreover, as mentioned in section 8.4, issues exist for the two databases (PostGIS and Neo4j) to *automatically talk* with each other, when a query needs to visit both databases. Potentially a federated database architecture would overcome this limitation (Robson, et al., 2018), where the user only needs to visit a master database to perform any query.

In the future, the work can be extended to develop a data processing layer and presentation

layer. Then the next question will be what are the necessary analytical and simulation programmes/scripts/APIs that needs to be developed to provide better decision support on the fine scale infrastructure networks. Also, it is interesting to explore how to develop the presentation layer. What visualisation engine should be used, how to render the complex infrastructure network, and how to design the user interface are considered to be the focus in the future (Leskens, et al., 2017).

### ***9.3.2 Understanding Dynamics of Infrastructure Networks***

There has been a growing trend of using digital city models and sensor network data to understand in real time (if possible) supply and demand between utility assets and buildings they service (Metke, et al., 2010; Rosen, et al., 2016; Tao, et al., 2018). Achieving that requires the representation of the spatial and temporal dynamics of the resource flows (Li, et al., 2013; Puig, et al., 2017).

In this PhD research, in Chapter 3, representation of spatial and temporal transient attributes is discussed (figure 3.2, figure 3.3). However, it still remains at an abstract and theoretical level without be implemented in a practical manner. In fact, all the network data discussed in Chapter 4, 5, 6, and 7 are *static*, where no temporal dynamics are considered. It is not clear how to implement such spatial and temporal dynamics in real applications, or how to encode, and manage such spatial and temporal dynamic data in an information system (Sun, et al., 2016). This is a major issue that should be addressed in the future.

Related research (Gilbert, et al., 2018) has been undertaken using open source streaming software (such as Apache Kafka) together with NoSQL database (Neo4j) to represent and monitor real time dynamic resource flows within utility infrastructure networks. However, this research focused on an individual building, and simplified utility network. This can be a good starting point, and future challenge will be to represent dynamic flows across multiple potentially hundreds of thousands of assets simultaneously. Meanwhile, this work needs to be

integrated with the development of decision support system (section 9.3.1), where the key question is how to render and visualise the spatial and temporal dynamics in the presentation layer.

### ***9.3.3 Big Data Processing Capability***

Efficient processing and analysis on geospatial big data is always considered as a major challenge for any geospatial information system (Amirian, et al., 2014). This is also true for developing a decision support system described in section 9.3.1. As discussed in Chapter 4 (section 4.7), computation time for generating electricity distribution networks for Great London would typically take a single desktop workstation 12 days. This is considered inefficient in real applications. The configuration of generic spatial heuristic algorithm is difficult and not-straightforward enough for the user. The user now needs to manually download input data from the data source (MasterMap, for example) and then run the algorithm to generate the result. This can be a tedious task if the user needs to generate electricity for every city in the UK.

Likewise, the Hybrid Database approach proposed in Chapter 7, can still suffer from the same issue. In section 7.6.3, the complex query 2, accessing the impact from each of the 100 spatial hazards to the electricity distribution networks in Great London, requires almost one hour for Hybrid Database to return the result. That is because the Hybrid Database needs (for each spatial hazard) perform spatial/attribute/topology queries to assess its impact on the electricity distribution networks, which can be time consuming (suppose there are 1000 spatial hazards instead of 100, then this complex query 2 will take 10 more time).

To address the current disadvantages in handling geospatial big data, future work can focus on the follow aspects.

First, in section 4.7, time complexity of the generic spatial heuristic algorithm is  $O(N_b^2)$ ,



where  $N_b$  is number of buildings in input area. This can be computationally expensive (as  $N_b$  doubles, processing time increases four times). Currently, the algorithm reads input data in one-go, that is, it reads all the assets, all the buildings, and all the roads to generate result. Parallel computing can be a potential solution to accelerate the algorithm (for example, create an instance for each asset to generate network and later merge these networks). This can be done via GPU or cloud computing techniques (Xia, et al., 2011). The key challenge here is how to modify the algorithm so that it can be parallelized.

Secondly, APIs on top of the spatial heuristic algorithm, can be developed so that the algorithm can retrieve input data from data sources automatically, and inference of network data is easier to the user side.

Finally, to improve efficiency of querying data in databases, distributed computing or cloud computing can be a possible solution (Abuzalaf, et al., 2016). When setting up multiple instances of workstations, operations, such as complex query 2 in section 7.6.3, can be executed more efficiently in a parallel way, by running each hazard footprint separately.

#### **9.4 Key Findings and Implications**

High granularity geospatial data on infrastructure network is crucial in many digital urban models and applications. However, accessing such good quality data is difficult or almost impossible. It is also not clear, what database approach is efficient in handling such complex network data. The thesis aims to tackle these challenges by proposing generic approaches for the inference and management of fine scale geospatial infrastructure network data.

A geospatial ontology is proposed which contains key entities, attributes and relationships to represent fine scale geospatial infrastructure network and the resource flows. The major contribution is the inclusion of building-infrastructure connections and infrastructure dependency/interdependency. This ontology serves as a general data model which facilitates

better information and knowledge shares in geospatial infrastructure network data. A spatial heuristic algorithm is developed as a scalable and transferable approach to infer layout of utility or road network if accessing real network layout. This algorithm is tested and validated to ensure the synthetic network layout is plausible, and is considered as a new and generic data inference approach. Finally, a Hybrid Database (PostGIS + Neo4j) is proposed for the efficient management and query of fine scale geospatial infrastructure network. Through performance benchmarking test, the Hybrid Database outperformed the traditional spatial and relational database, especially at resolving network topology queries.

To conclude, this PhD contributes to inference of quality geospatial infrastructure network data, and a database system to efficiently manage such data. All of these opens up opportunities of the development of digital city models and applications, as well as management, fortification and planning of critical infrastructure at fine geospatial scale.

## Appendix A – Basic Software Stacks used in the Thesis

1. Python 3.5 Development Environment
2. PostgreSQL 9.4.8  
<https://www.postgresql.org/>
3. PostGIS 2.2  
<https://postgis.net/>
4. Neo4j 3.1.3  
<https://neo4j.com/>
5. Psycopg2 2.7.7 (python driver for PostGIS)  
<http://initd.org/psycopg/>
6. Neo4j-driver 1.7.2 (python driver for Neo4j)
7. NetworkX 1.1.1 (python library for manipulating network data)  
<https://networkx.github.io/>
8. Shapely 1.6.4 (python library for complex geometric operation)  
<https://shapely.readthedocs.io/en/stable/manual.html>
9. Fiona 1.7.12 (python driver for reading and writing shapefile document)  
<https://fiona.readthedocs.io/en/latest/manual.html>
10. PostGIS ITRC database schema  
<https://github.com/BurningWish/ITRC>

## Appendix B – Installation of the ITRC schema

In appendix A, a URL is given to download the ITRC schema

1. Please first install the follow library shown in the URL:

[https://github.com/BurningWish/ITRC/blob/master/nx\\_pgnet-0.9.post0.dev70%2Bngdb91640.dirty-py2.py3-none-any.whl](https://github.com/BurningWish/ITRC/blob/master/nx_pgnet-0.9.post0.dev70%2Bngdb91640.dirty-py2.py3-none-any.whl)

This **nx\_pgnet** is the python driver for reading/writing data into PostGIS in the ITRC schema.

2. Now start the PostGIS on your computer, and create a new database. Then turn on the PostGIS extension for the database.

3. Now restore this database, using the backup file, from this URL:

[https://github.com/BurningWish/ITRC/blob/master/nx\\_pgnet-master/pg\\_schema/backup/network\\_schema\\_empty.backup](https://github.com/BurningWish/ITRC/blob/master/nx_pgnet-master/pg_schema/backup/network_schema_empty.backup)

4. A database in the ITRC schema has been created, and the document with regards to the **nx\_pgnet** library is in this URL:

[https://github.com/BurningWish/ITRC/blob/master/nx\\_pgnet-master/doc/api.pdf](https://github.com/BurningWish/ITRC/blob/master/nx_pgnet-master/doc/api.pdf)

## Appendix C – Spatial heuristic algorithm

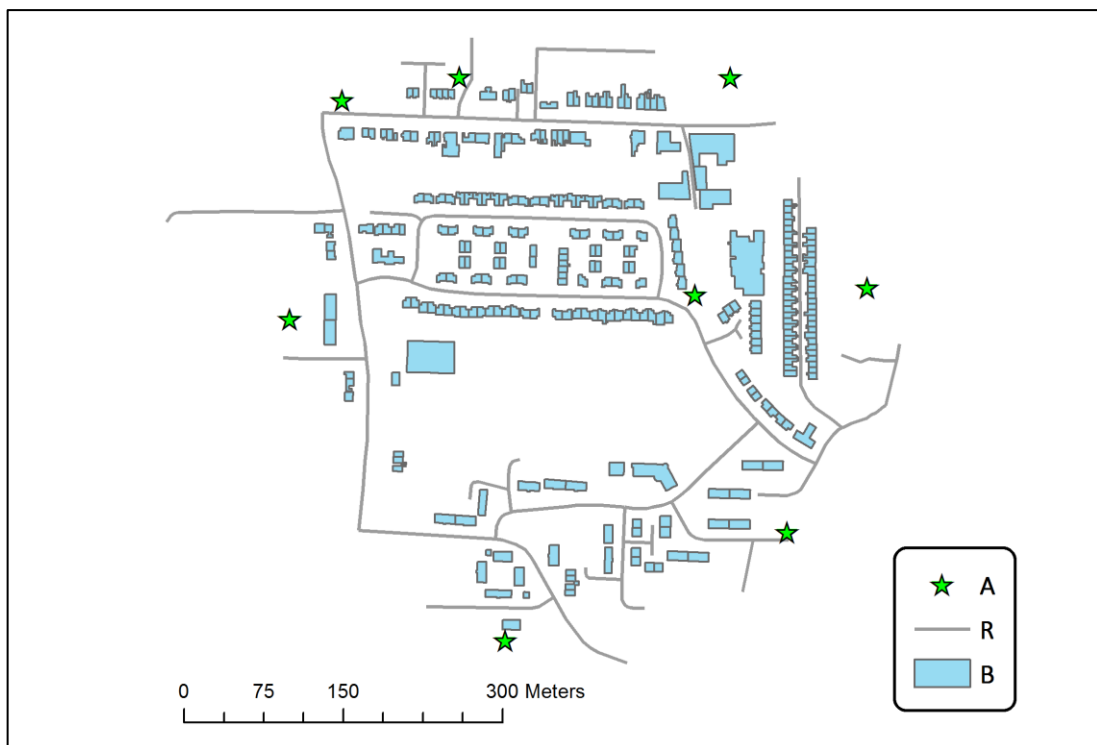
This appendix shows how to generate fine scale electricity distribution networks from input data (substations, road network, buildings) for an example area of Newcastle upon Tyne. This algorithm is discussed in chapter 4. The appendix also shows city scale electricity networks generated for major cities in the UK.

The code can be found in this URL:

<https://github.com/BurningWish/Heuristic-Algorithm>

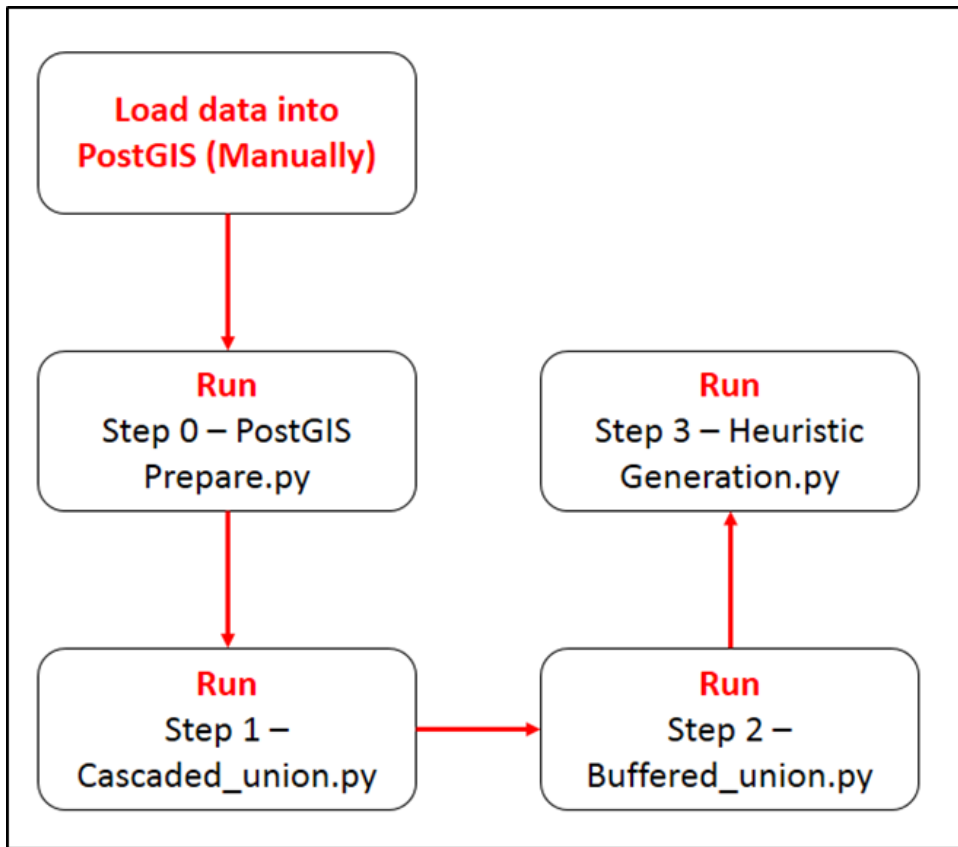
The input data are downloadable from digimap: <https://digimap.edina.ac.uk/>.

We need three layers from MasterMap: ITN – Integrated Transport Network, Topography (building), Point of Interest (substation). An example is given in figure C1.



**Figure C1.** An example of input for the spatial heuristic algorithm. (A = substations, R = roads, and B = buildings).

The way the algorithm works is shown in figure C2.

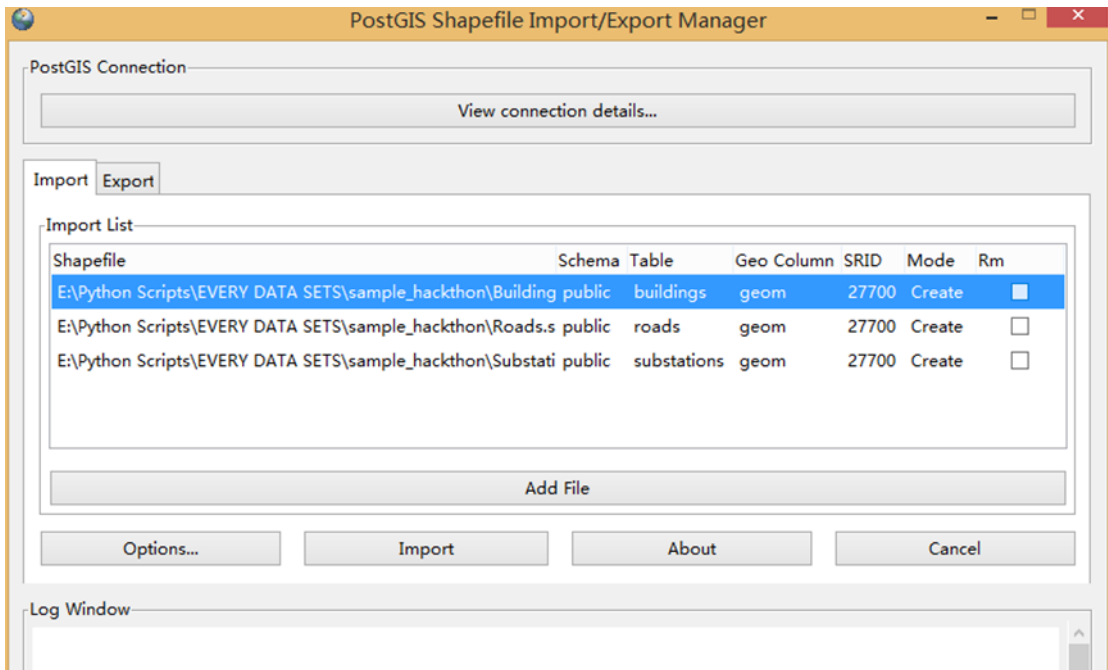


**Figure C2.** The way in which the algorithm works.

Basically, we will first load data into PostGIS (manually) and then sequentially execute 4 python scripts.

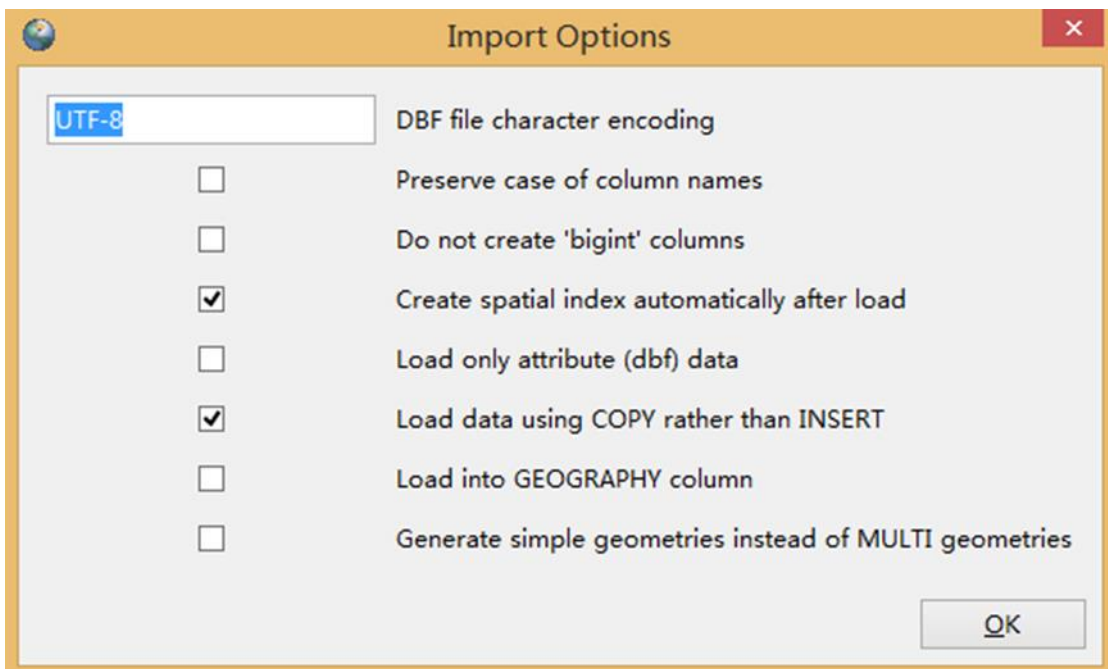
### 1. Loading data into PostGIS

Please open your PostGIS and create a new database (which I called “sample\_hackthon” in this example). Please set SRID to be 27700 when loading the shapefiles. In the end, you will create 3 tables, which are “buildings”, “substations”, and “roads”. Please note the table names are all in lowercase. Please see figure C3 for details.



**Figure C3.** Loading data into PostGIS.

Please note that when loading data, there is an import option. In here, please make sure that the box for “Generate simple geometries instead of MULTI geometries” is not ticked. Otherwise, algorithm will fail. See figure C4 for details.

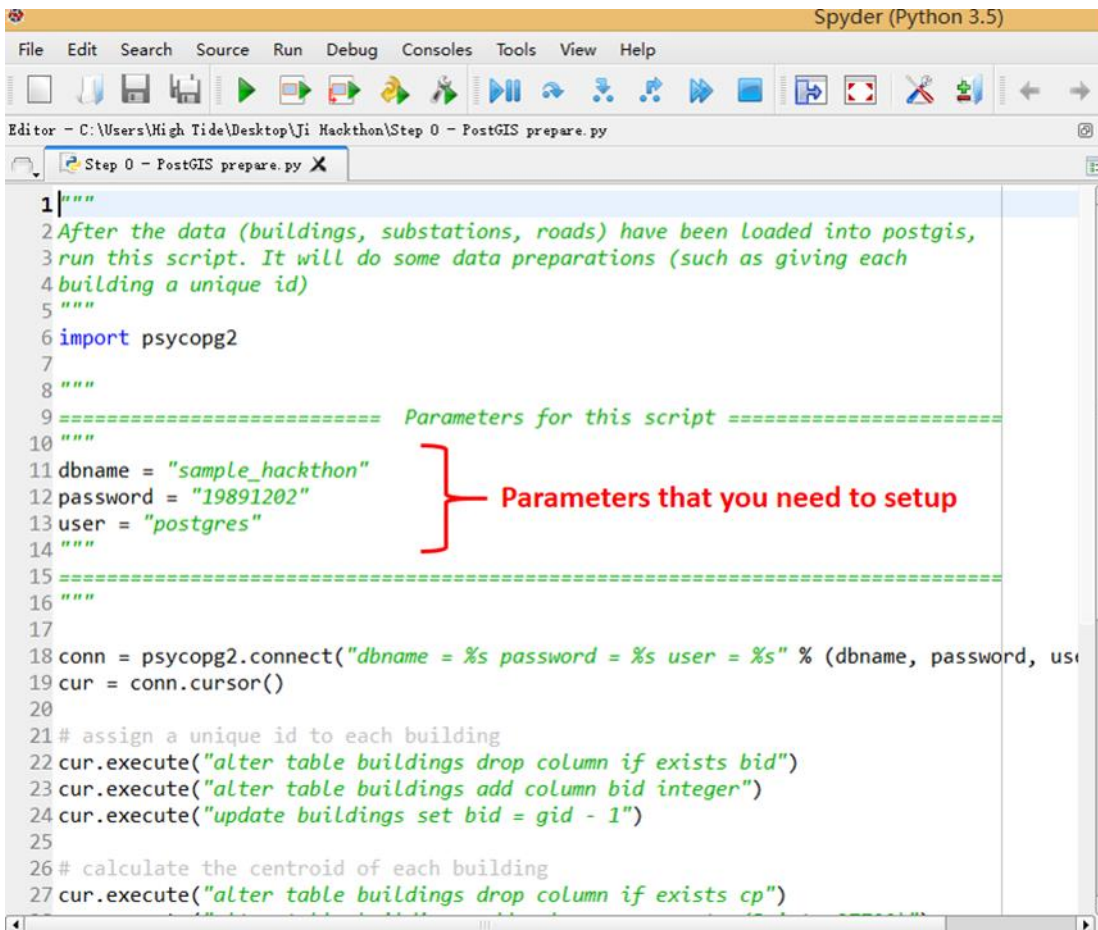


**Figure C4.** No need to tick box “Generate simple geometries instead of MULTI geometries”.

## 2. Running Scripts (Step 0 → Step 3)

Now the data have been loaded, then we can just run the scripts to execute the algorithm. You can use any IDE (such as Pycharm or Spyder) to open the scripts and simply run them.

Before running each script, there might be some parameters that you need to change. In general, these are parameters used to connect to your PostGIS database. For example, for the script **Step 0 – PostGIS prepare.py**, there are some parameters that you might need to change, see figure C5. I always put the parameters section near the top within each script, so they are easy to find.



```
1 """
2 After the data (buildings, substations, roads) have been loaded into postgis,
3 run this script. It will do some data preparations (such as giving each
4 building a unique id)
5 """
6 import psycopg2
7
8 """
9 ===== Parameters for this script =====
10 """
11 dbname = "sample_hackthon"
12 password = "19891202"
13 user = "postgres"
14 """
15 =====
16 """
17
18 conn = psycopg2.connect("dbname = %s password = %s user = %s" % (dbname, password, user))
19 cur = conn.cursor()
20
21 # assign a unique id to each building
22 cur.execute("alter table buildings drop column if exists bid")
23 cur.execute("alter table buildings add column bid integer")
24 cur.execute("update buildings set bid = gid - 1")
25
26 # calculate the centroid of each building
27 cur.execute("alter table buildings drop column if exists cp")
```

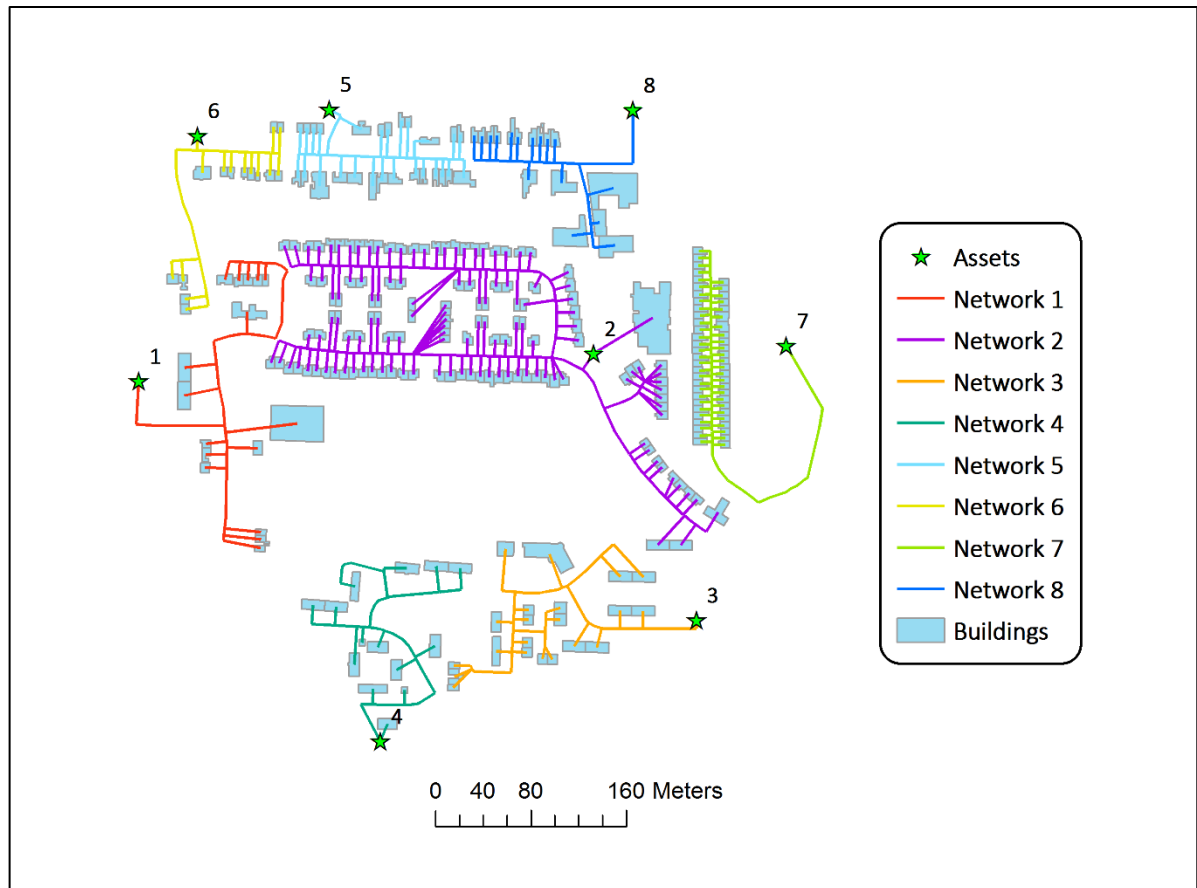
Figure C5. Parameters for Step 0 – PostGIS prepare.py.

If there is nothing wrong, after running **Step 3 – Heuristic Generation.py**, in your current working directory there will be a folder called result, and within it there are two folders called



**Edges** and **Nodes**. I store single network instance separately. For example, the file **Edges0.shp** and **Nodes0.shp** are the edges and nodes for the electricity distribution network, where the id for the substation is 0.

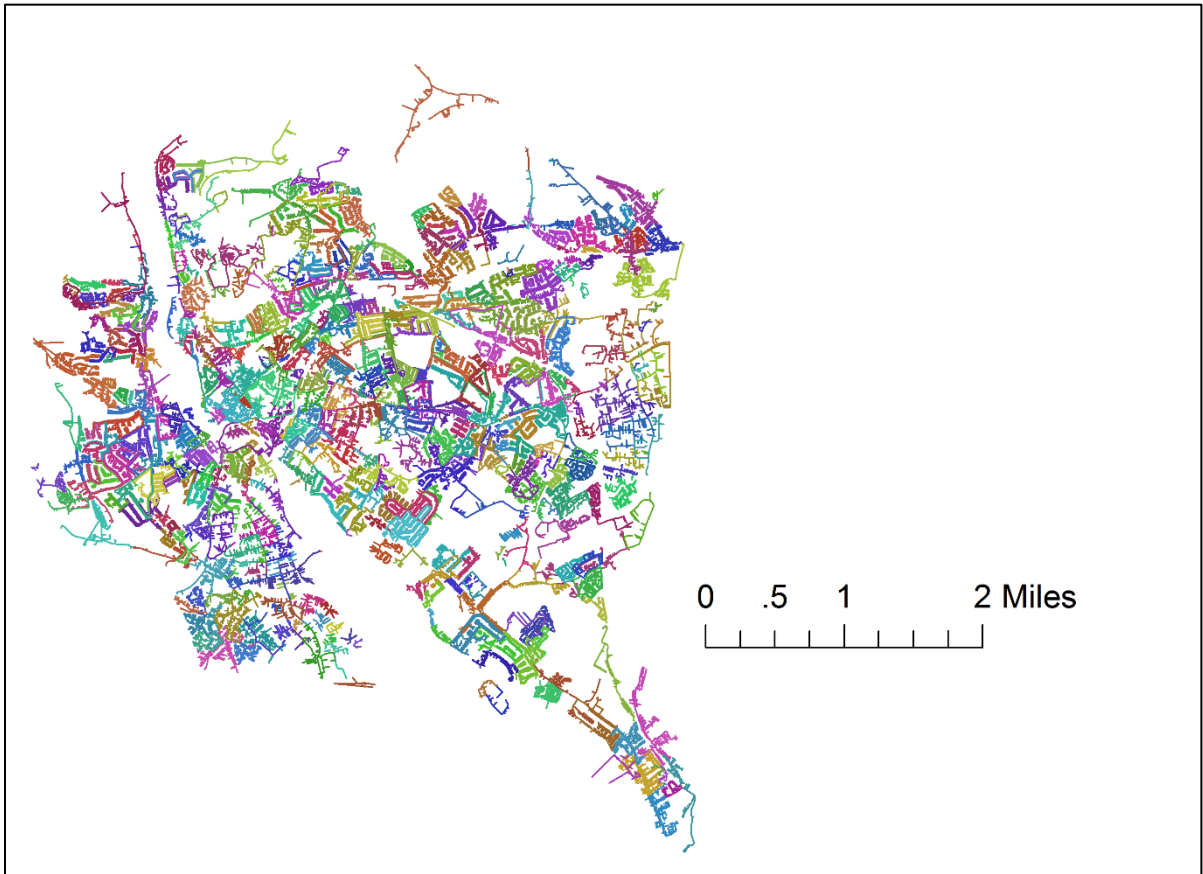
Figure C6 shows the result of synthetic networks, based on figure C1 as input data.



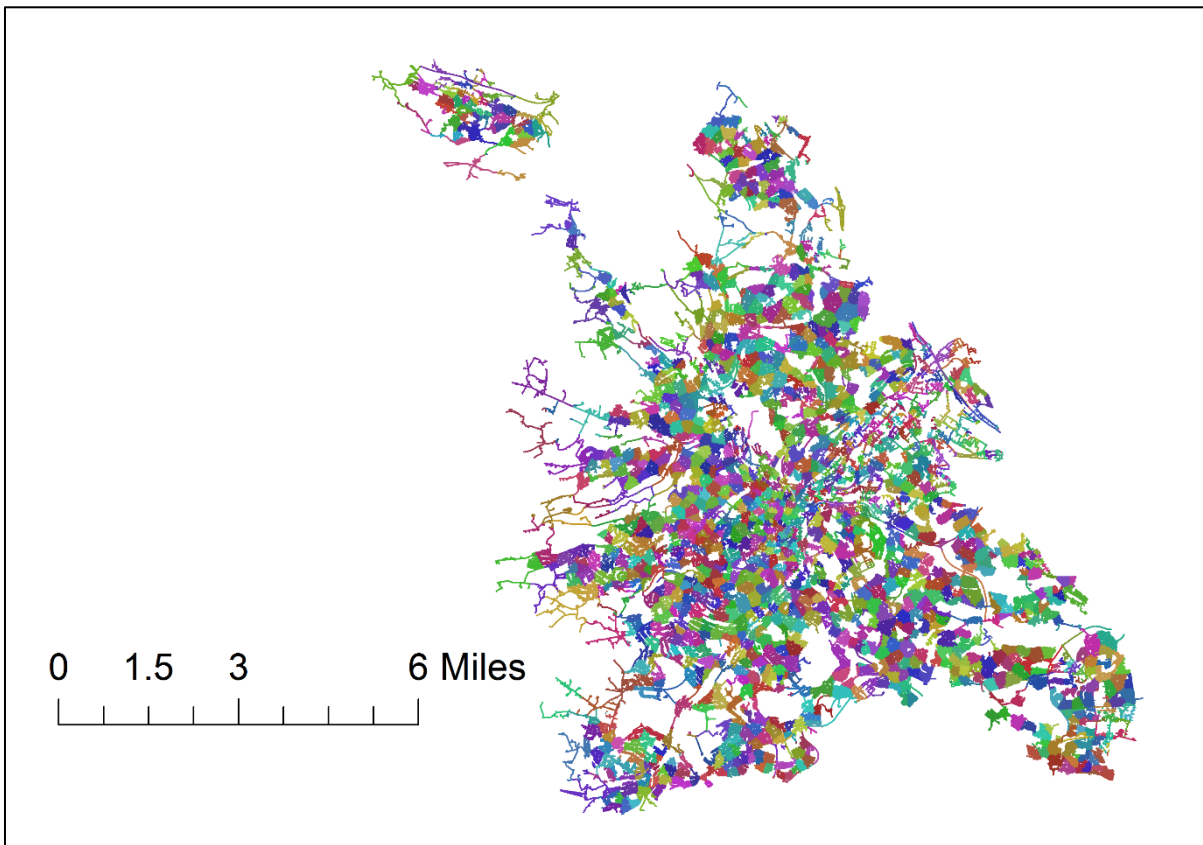
**Figure C6.** The synthetic networks generated, based on figure C1.

### 3. City scale electricity distribution networks generated for UK major cities

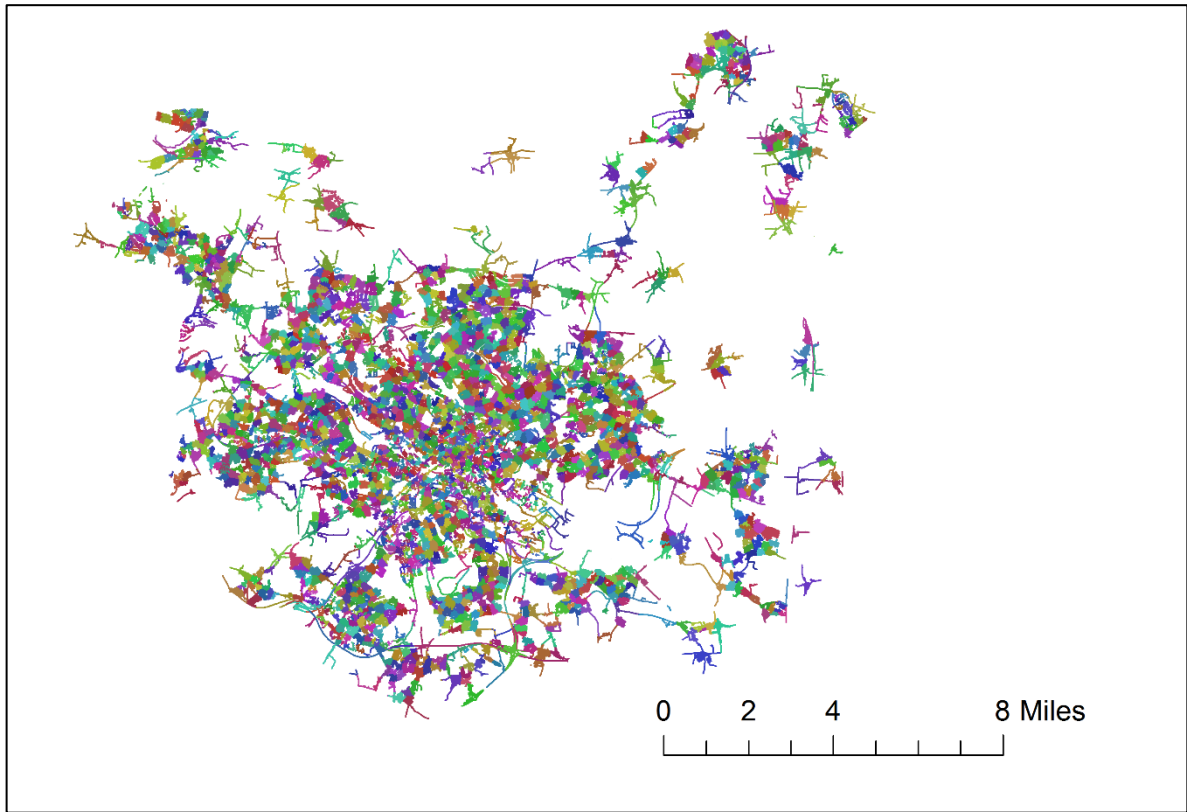
Below are the electricity distribution networks generated for Exeter, Sheffield, Leeds, Birmingham, and Greater Manchester. Note each colour represents a single network instance.



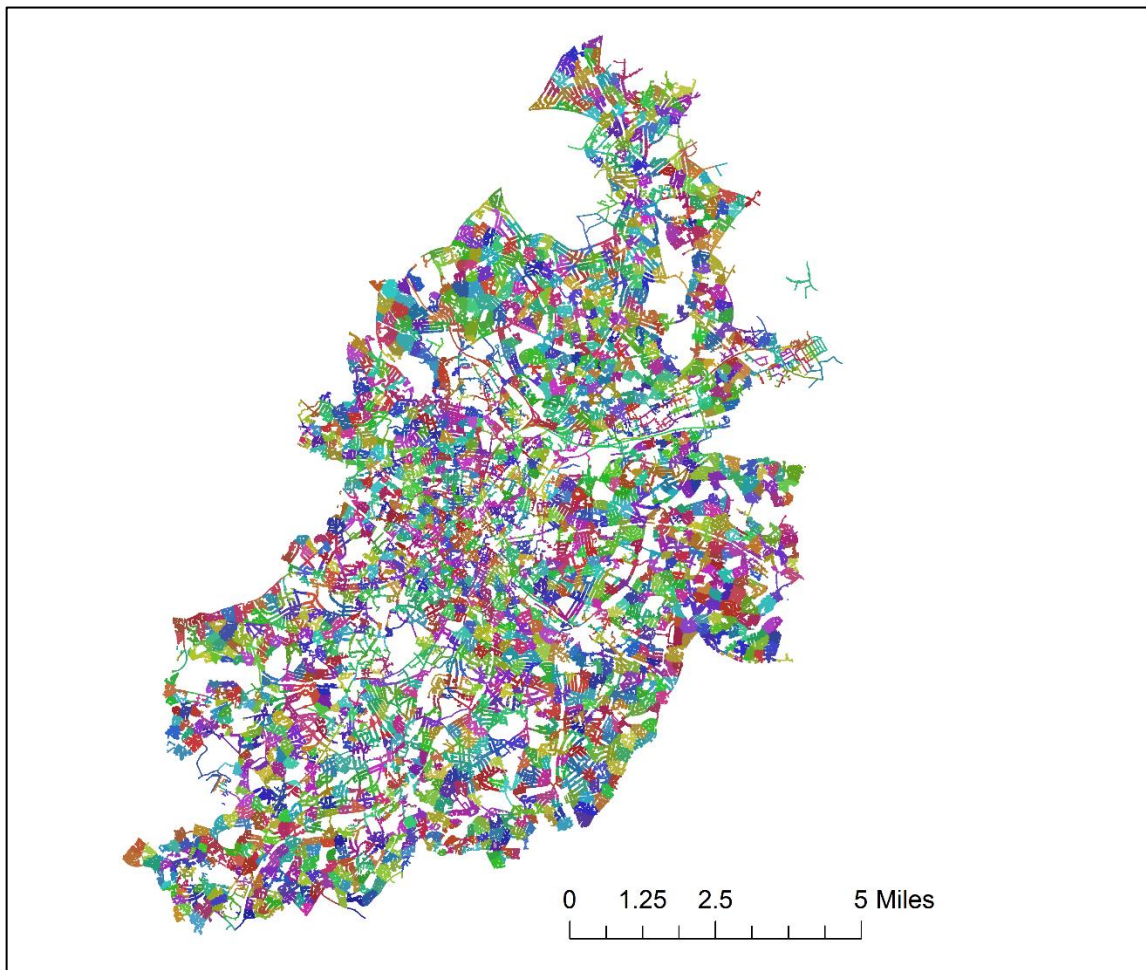
**Figure C7.** Synthetic electricity distribution networks for Exeter.



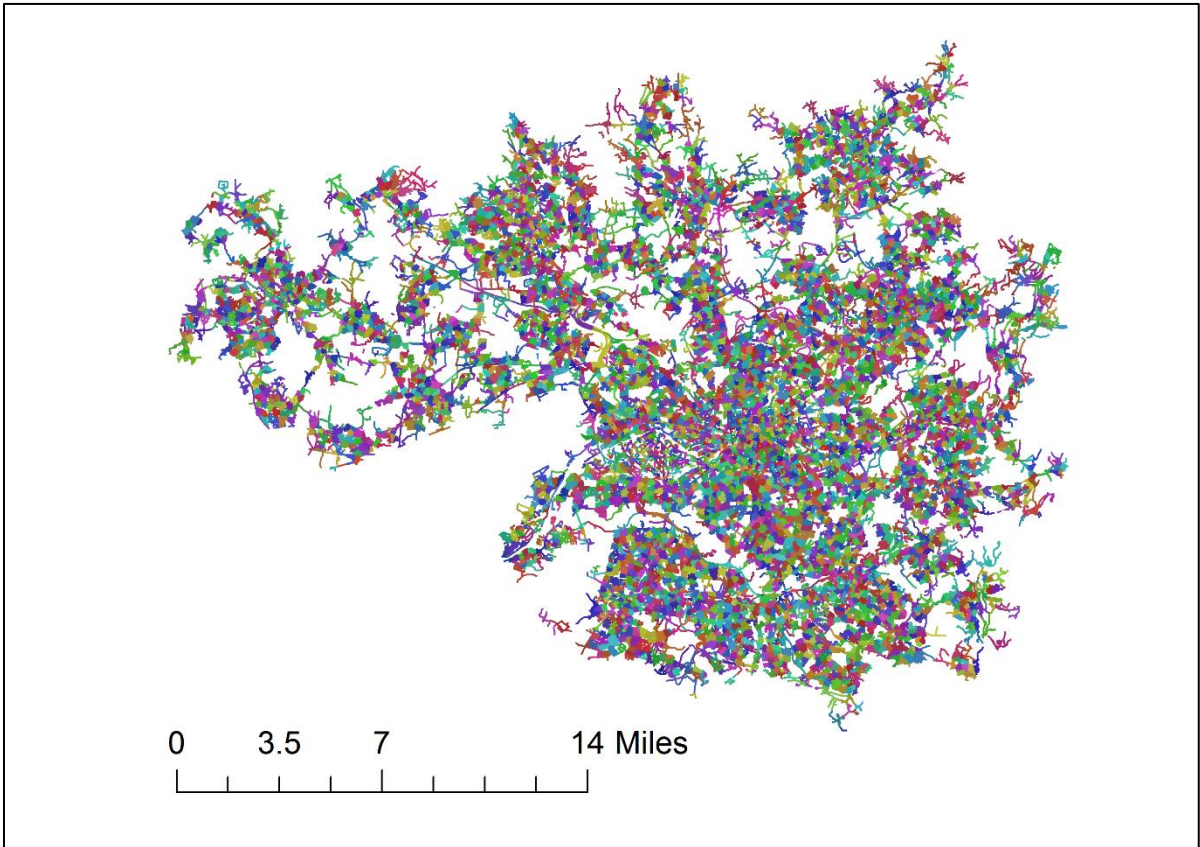
**Figure C8.** Synthetic electricity distribution networks for Sheffield.



**Figure C9.** Synthetic electricity distribution networks for Leeds.



**Figure C10.** Synthetic electricity distribution networks for Birmingham.



**Figure C11.** Synthetic electricity distribution networks for Greater Manchester.

## Appendix D – Gas Network Integration

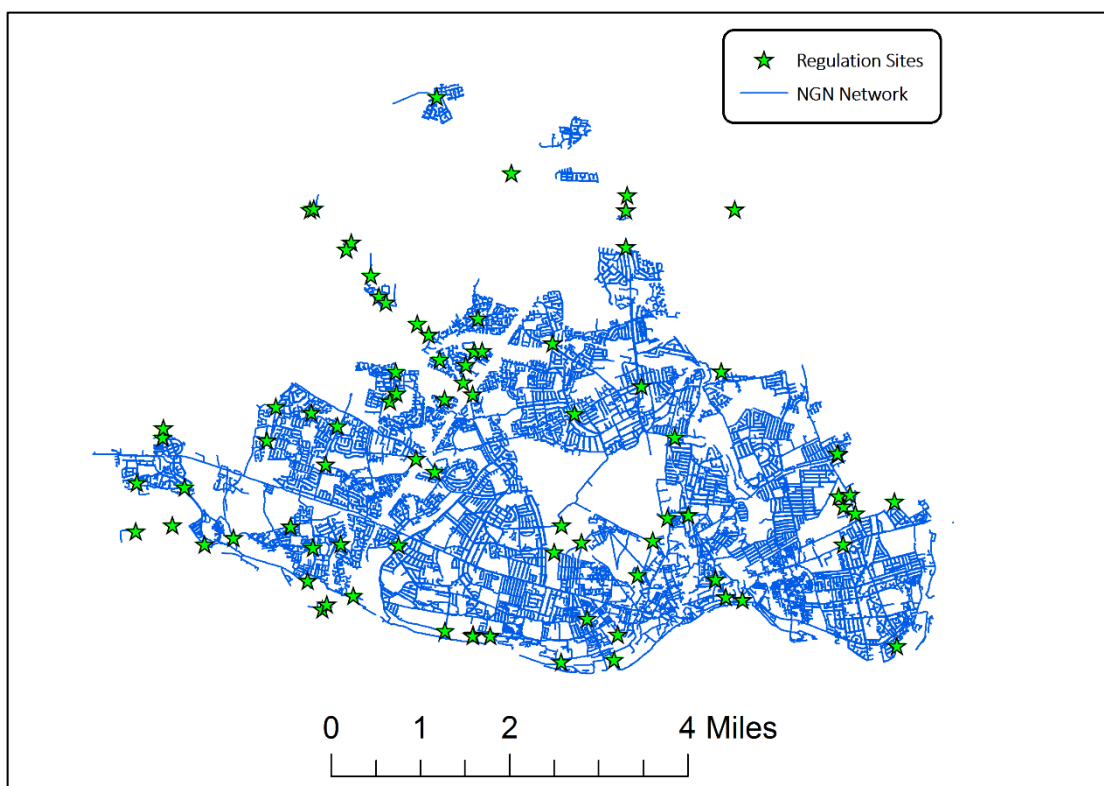
The appendix shows the code to generate fine scale gas distribution networks (connecting buildings) in Newcastle upon Tyne. The work is discussed in section 5.2

The code can be found at this URL:

<https://github.com/BurningWish/Gas-Network-Integration>

**Necessary input layers:** buildings, ITN, Northern Gas Network (NGN) network.

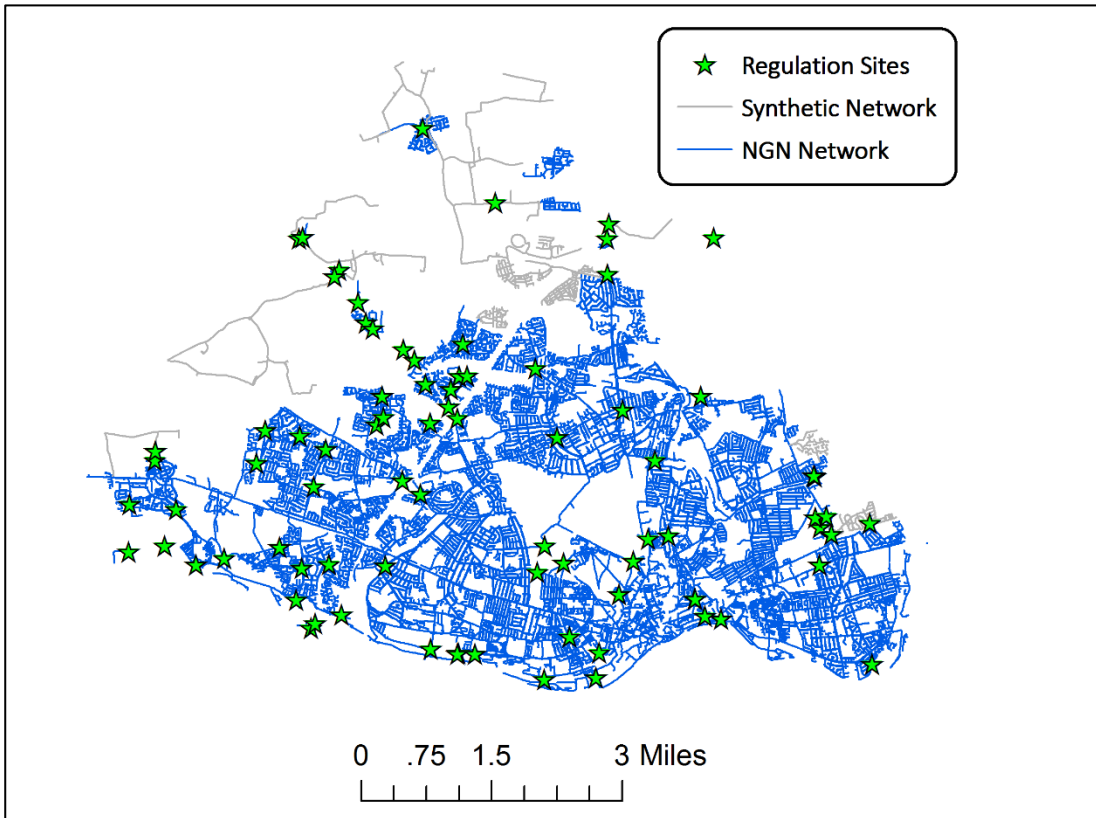
The buildings and ITN layer are available from OS MasterMap. The NGN layer cannot be made public due to data sensitivity. Figure D1 shows the NGN network.



**Figure D1.** The NGN network data (layout of main pipes).

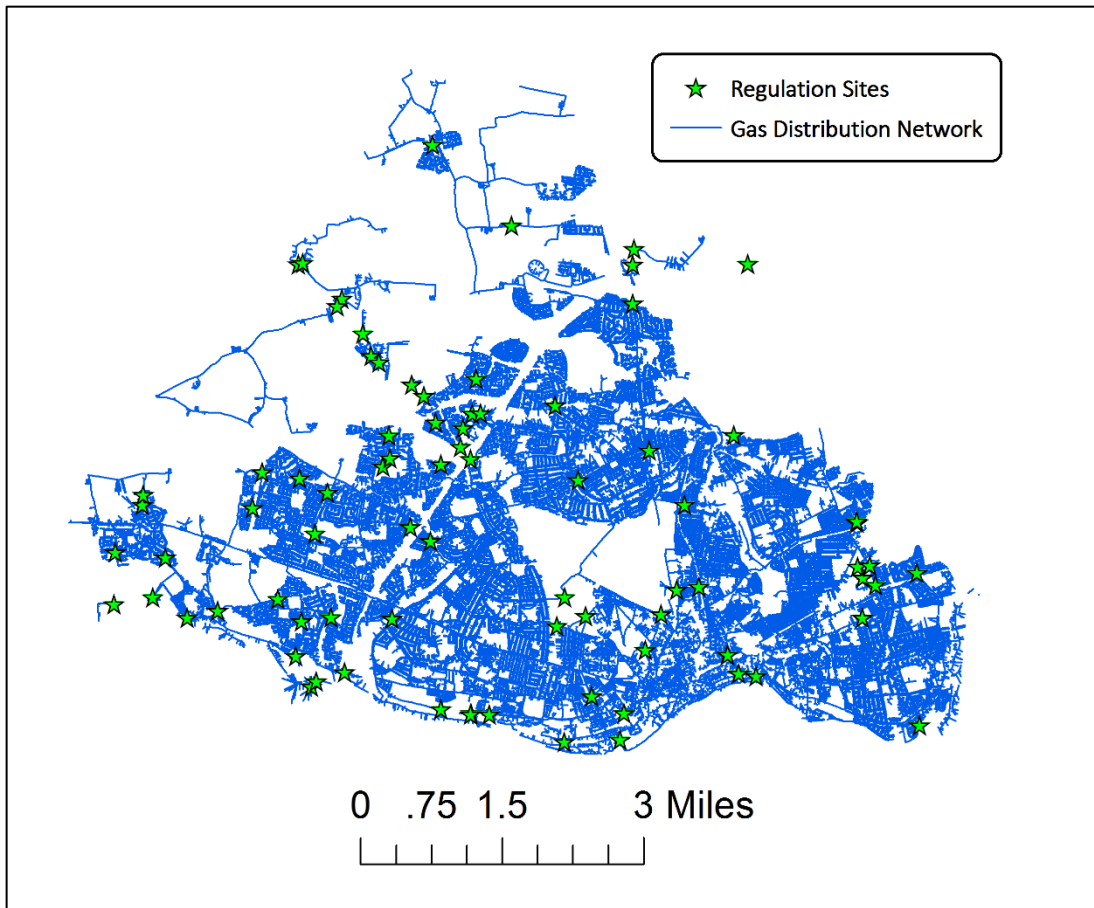
First run the script **Step 0 - Gas Main Infer.py**, which will infer gas main pipes in new

developing areas in Newcastle where there is none. The result is shown in figure D2.



**Figure D2.** Inferred layout of gas main pipes, where there is none.

Now combine synthetic Network and NGN Network into one layer, run script **Step 1 - Preprocessing Data.py**, **Step 2 - Terrace Generation.py**, and **Step 3 - Main Script.py** sequentially. Then fine scale gas distribution networks (connecting buildings) will be generated for Newcastle upon Tyne (figure D3).



**Figure D3.** Gas distribution network for Newcastle upon Tyne.

## Appendix E – Water Supply Network Integration

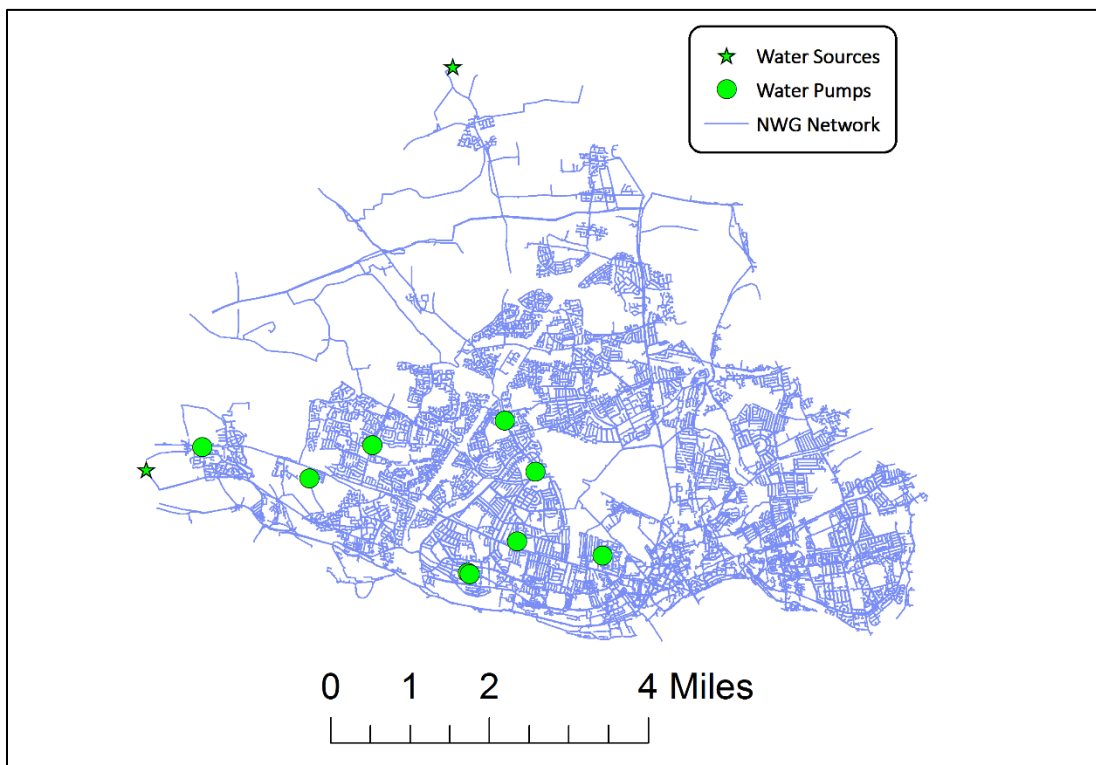
The appendix shows the code to generate fine scale water distribution networks (connecting buildings) in Newcastle upon Tyne. The work is discussed in section 5.3.

The code can be found at this URL:

<https://github.com/BurningWish/Water-Network-Integration>

**Necessary input layers:** buildings, Northumbria Water Group (NWG) network.

The buildings layer is available from OS MasterMap. The NWG layer cannot be made public due to data sensitivity. Figure E1 shows the NWG network.

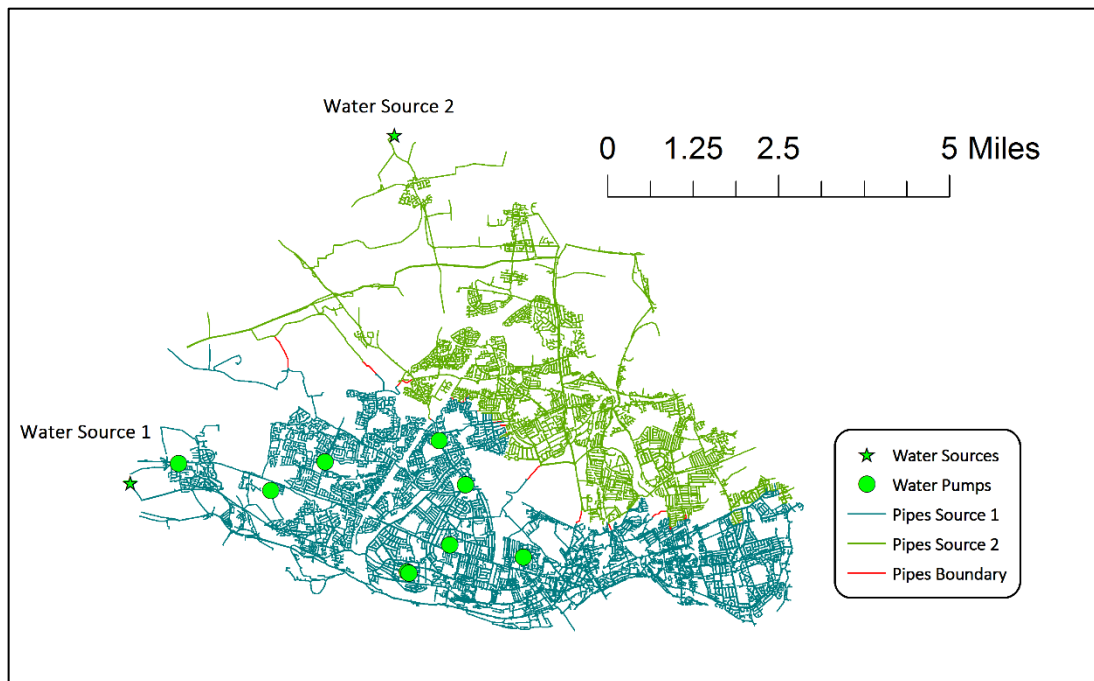


**Figure E1.** The NWG network data.

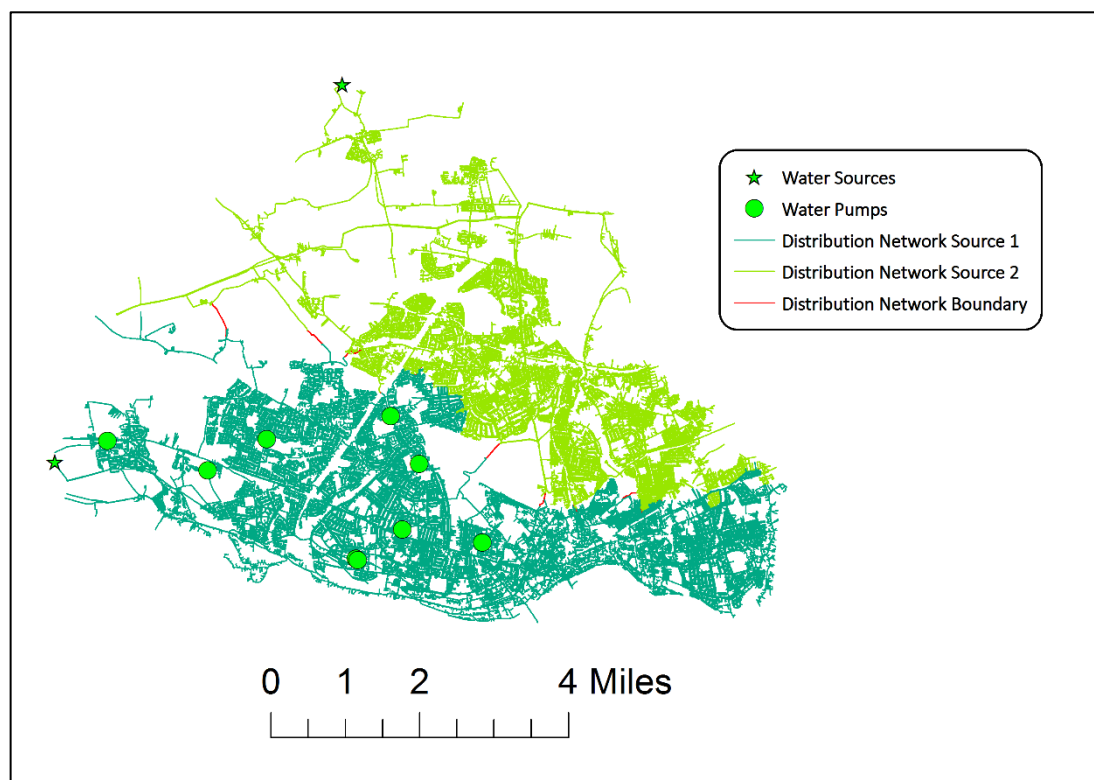
First run the scripts **Step 0 - WDA Calculation.py**, and **Step 1 - NWG Flow Infer.py** to infer



flow direction on the NWG network (figure E2). Then run the scripts **Step 2 - Preprocessing Data.py** and **Step 3 - Main Script.py** to generate fine scale water distribution network (connecting buildings) in Newcastle upon Tyne (figure E3).



**Figure E2.** Inferred water distribution area (WDA) based on NWG data.



**Figure E3.** Fine scale water distribution networks in Newcastle upon Tyne.

## Appendix F – Sewer Network Integration

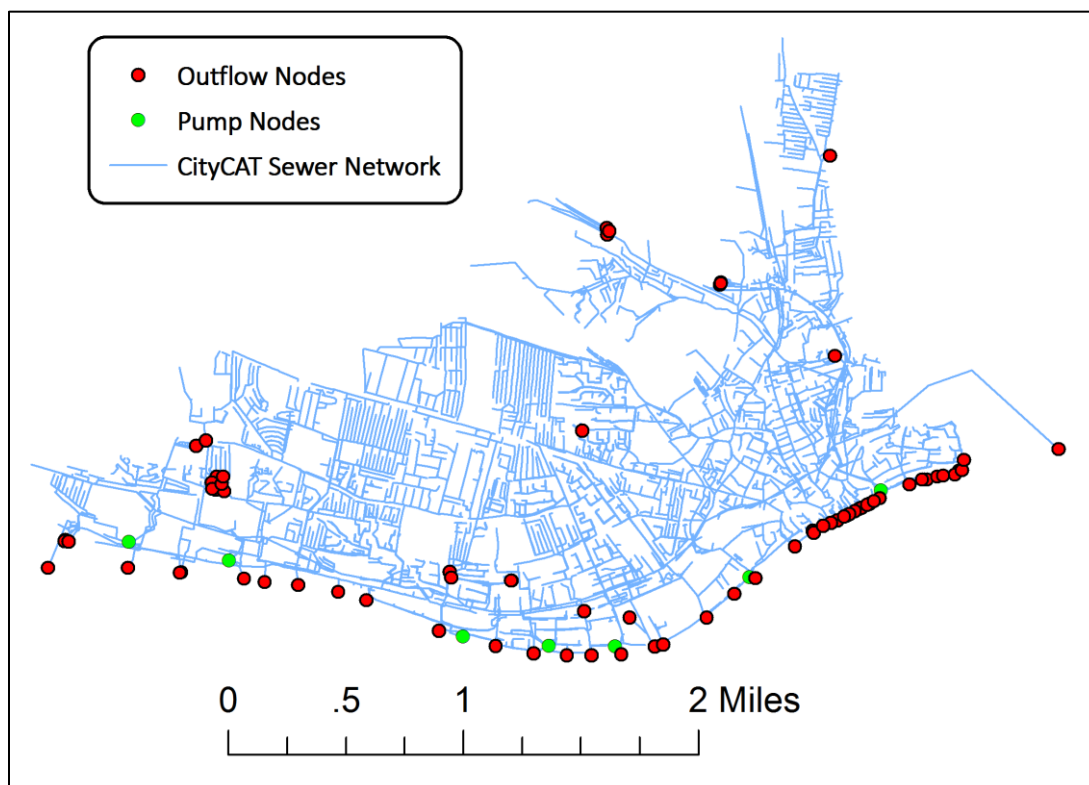
The appendix shows the code to generate fine scale sewer networks (connecting buildings) in Newcastle upon Tyne. The work is discussed in section 5.4.

The code can be found at this URL:

<https://github.com/BurningWish/Sewer-Network-Integration>

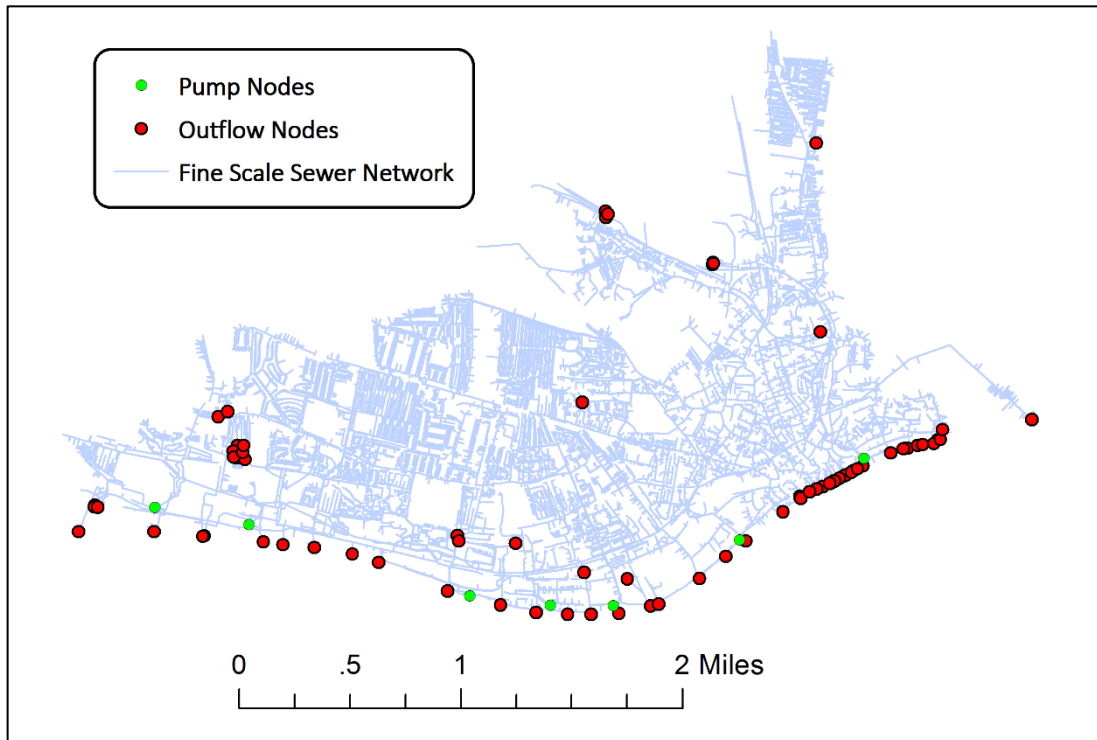
**Necessary input layers:** buildings, CityCAT sewer network, Newcastle DTM.

The buildings layer is available from OS MasterMap. The sewer network layer cannot be made public due to data sensitivity. Figure F1 shows the sewer network.



**Figure F1.** CityCAT sewer network.

Run the scripts **0 - Preprocessing Data.py** and **1 - Main Script.py** to generate fine scale sewer network in Newcastle upon Tyne (figure F2).



**Figure F2.** Fine scale sewer network in Newcastle (connecting buildings).

Note in section 5.4.3, we discussed an algorithm to infer sewer flow direction as if there is no flow information. The algorithm relies on the DTM layer, and is implemented via the script **Extra - Sewer Flow Infer.py**.

## Appendix G – Road Network Generation Algorithm

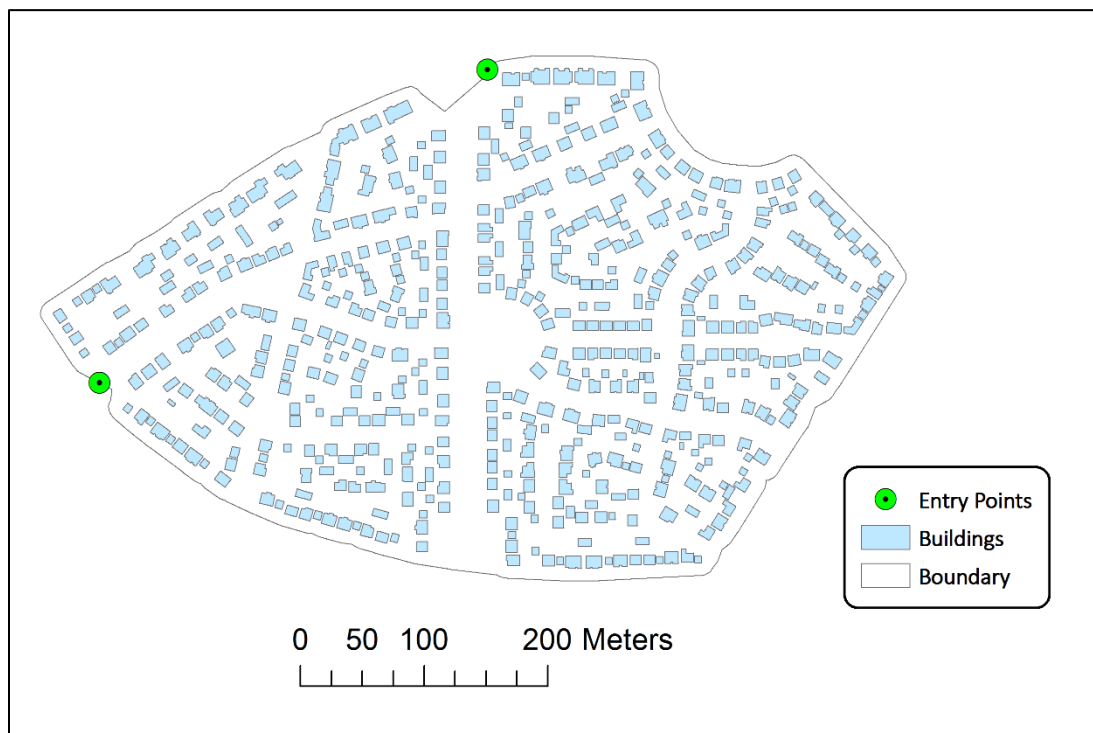
This appendix shows the code to generate road network, an approach that is discussed in chapter 6.

The code is available from the follow URL:

<https://github.com/BurningWish/Road-Network-Generation>

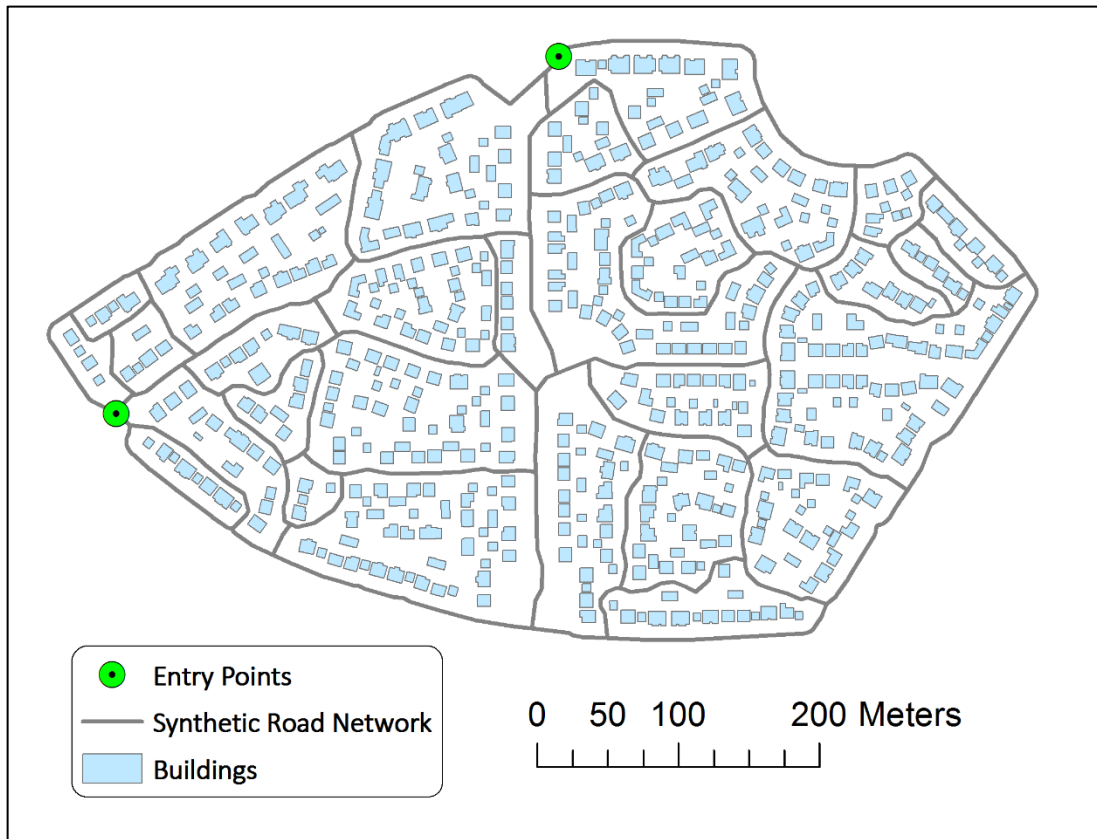
**Necessary input layers:** buildings, boundary, and entry points. The buildings layer is available from OS MasterMap. The other two layers needs to be given manually.

For example, figure G1 shows an example of input data layers (for the Arup project).



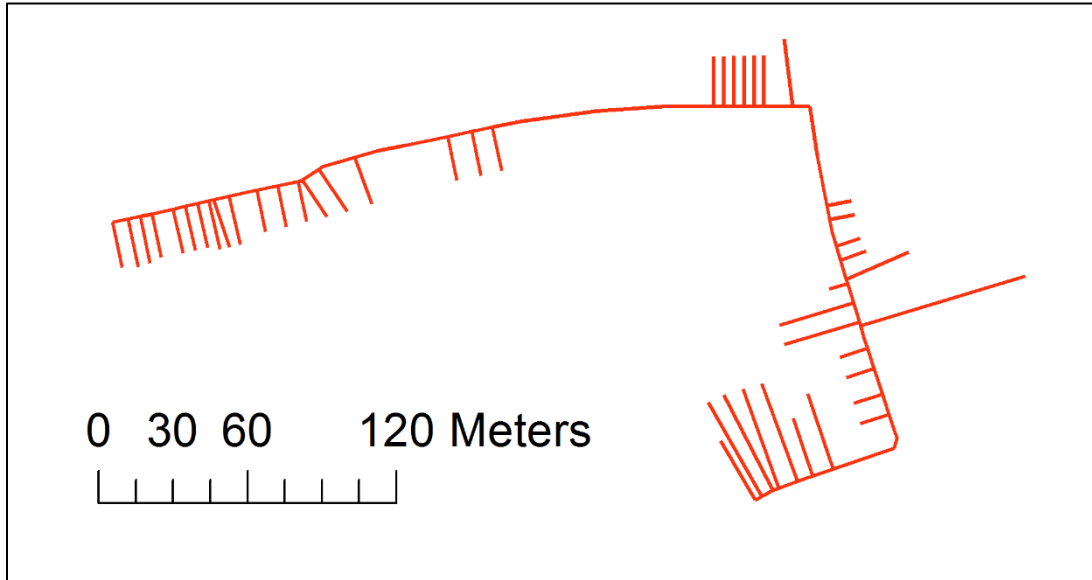
**Figure G1.** An example of input layers for the road network generation algorithm.

Then run the scripts from **0. Alter table attributes.py** to **7. Smooth Road Network Geometry.py** sequentially to generate synthetic road network (figure G2).

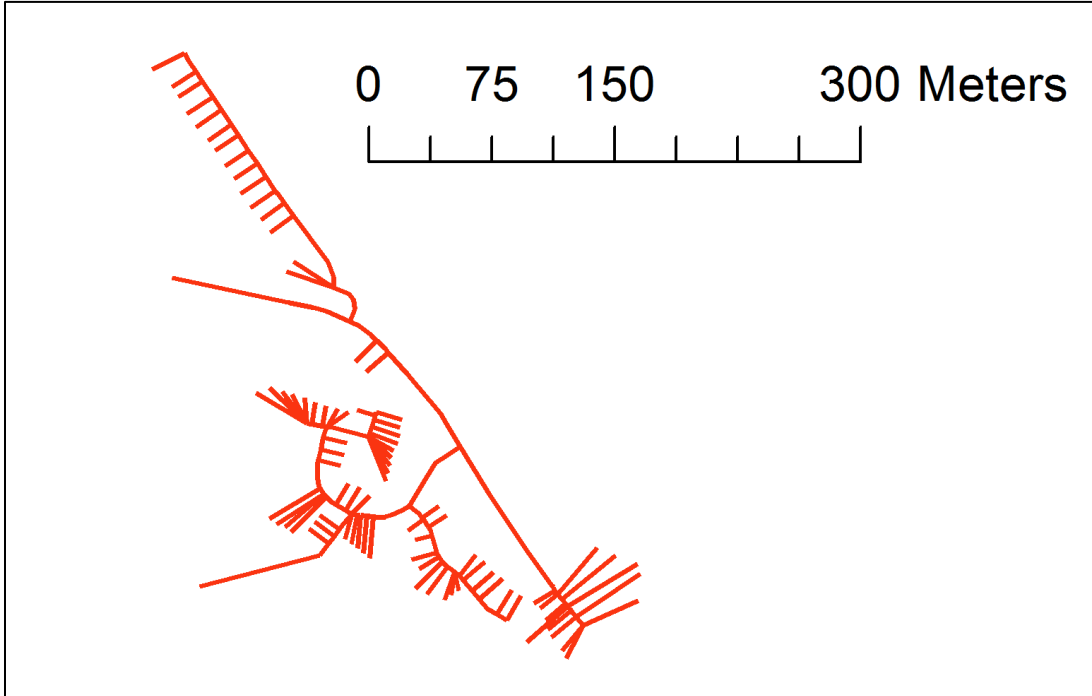


**Figure G2.** Synthetic road network, based on input from figure G1.

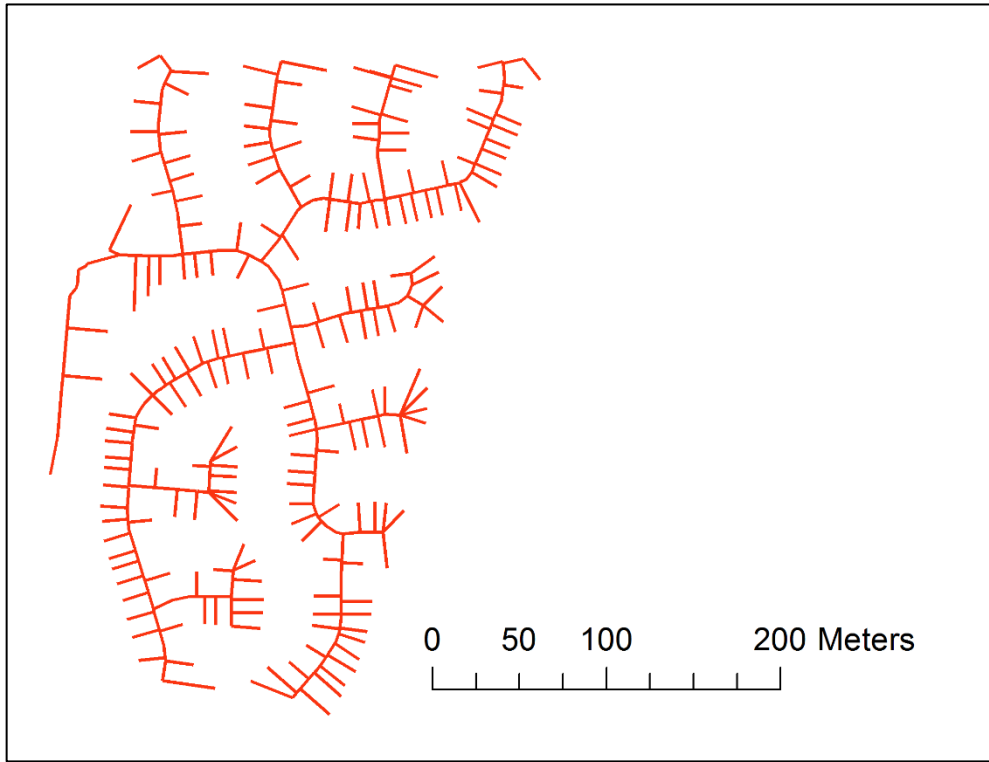
## Appendix H – Database Performance Benchmarking Test Data



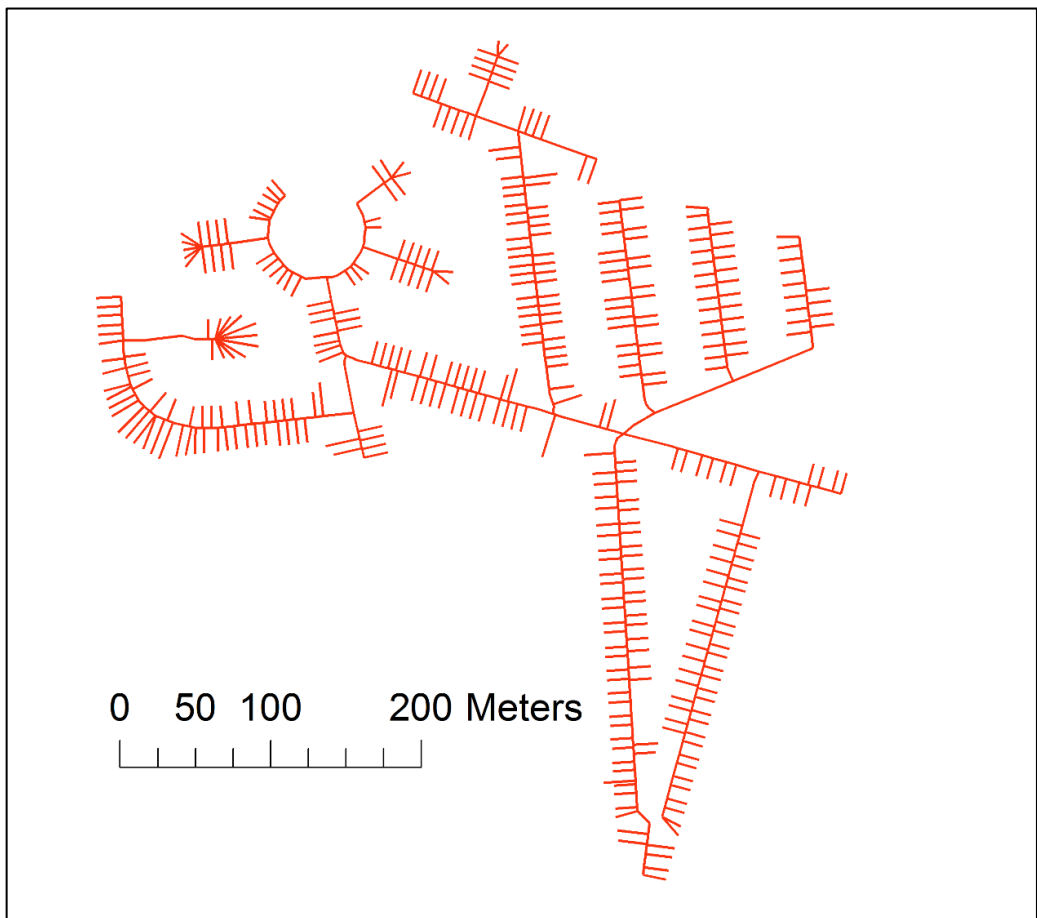
**Figure H1.** Type 1 data set, size 100.



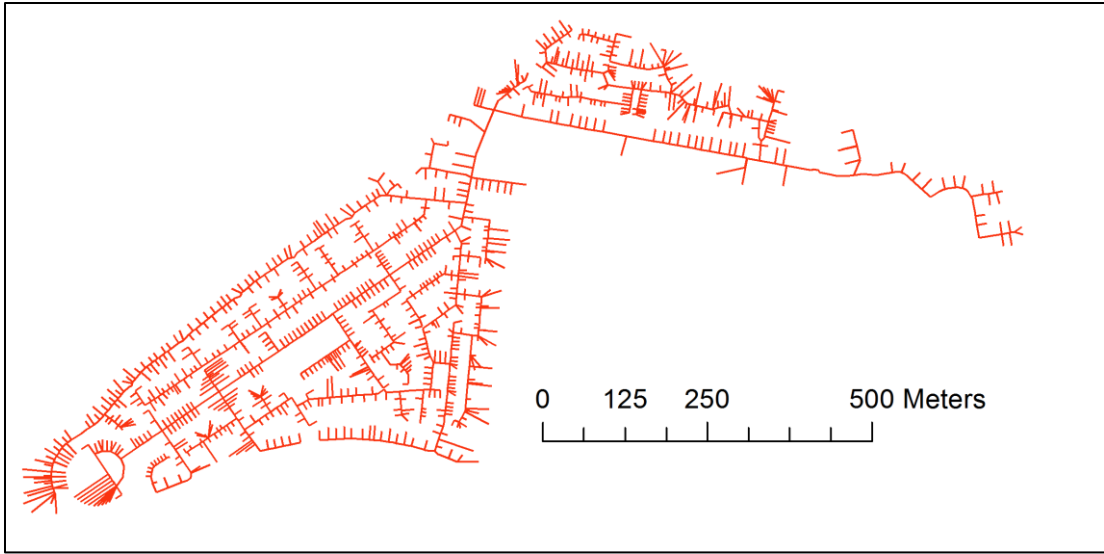
**Figure H2.** Type 2 data set, size 200.



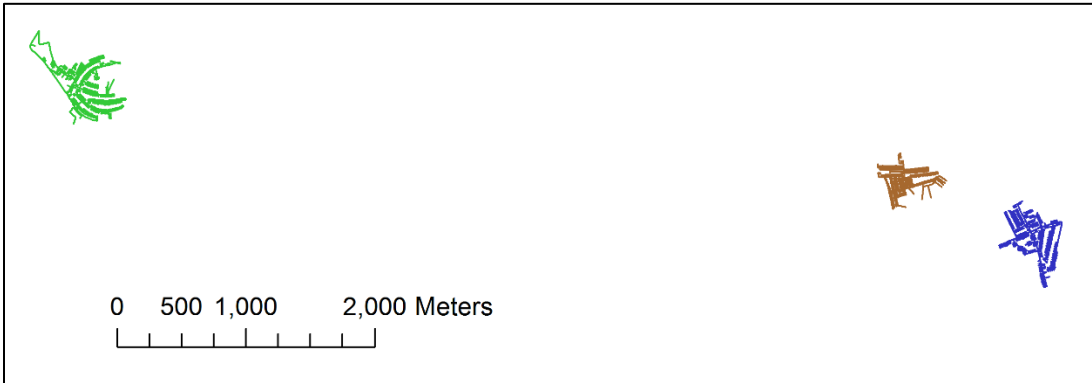
**Figure H3.** Type 1 data set, size 400.



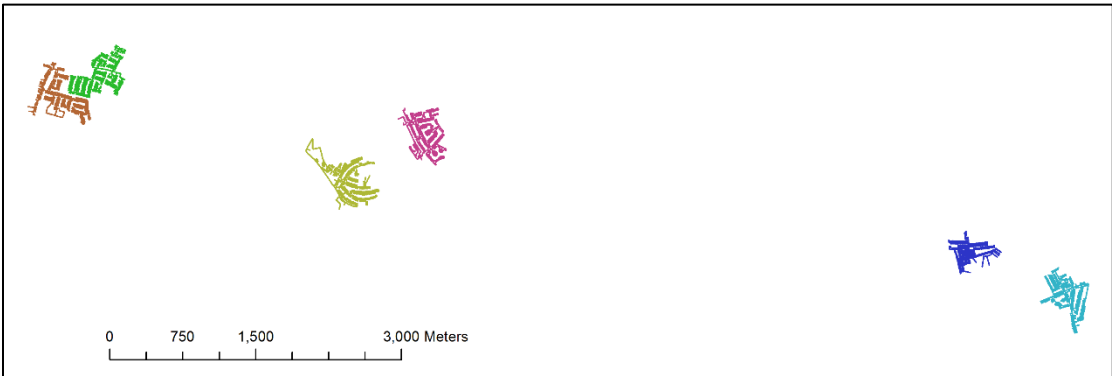
**Figure H4.** Type 1 data set, size 800.



**Figure H5.** Type 1 data set, size 1600.

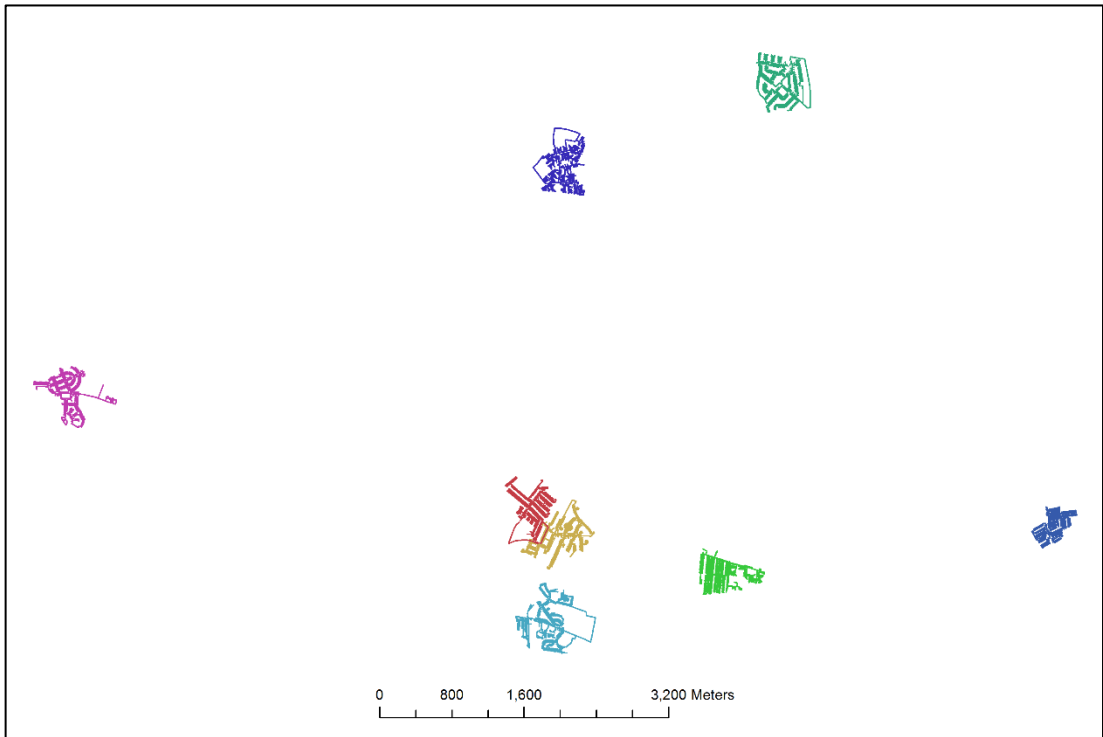


**Figure H6.** Type 2 data set, size 2500. Each colour refers to a single network instance.

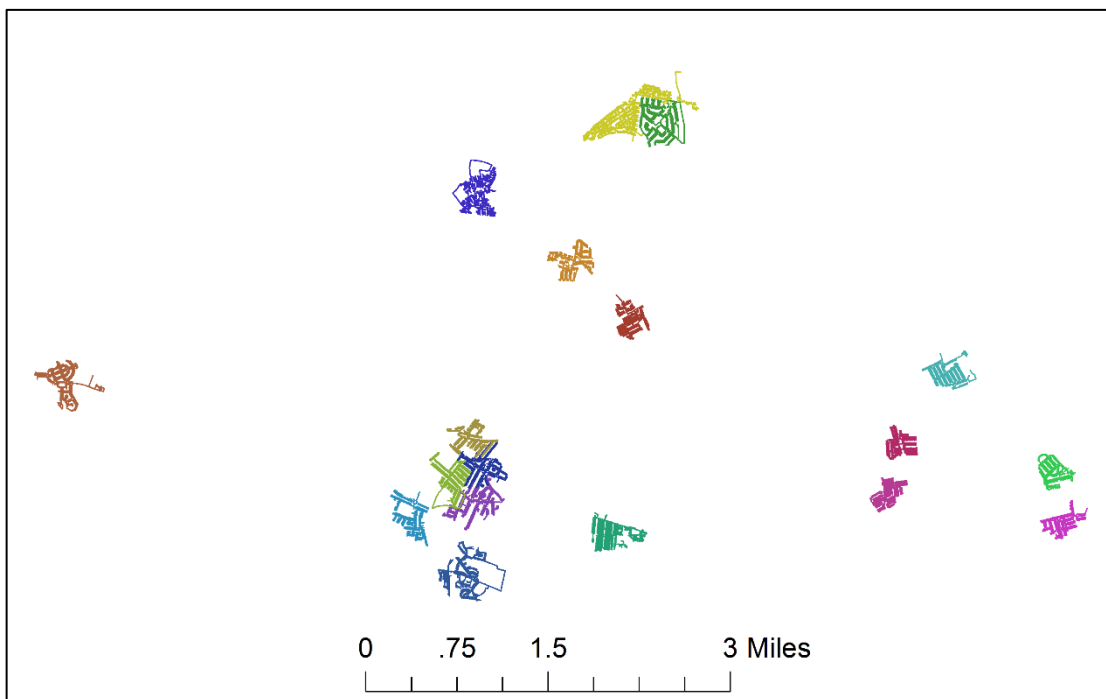


**Figure H7.** Type 2 data set, size 5000. Each colour refers to a single network instance.

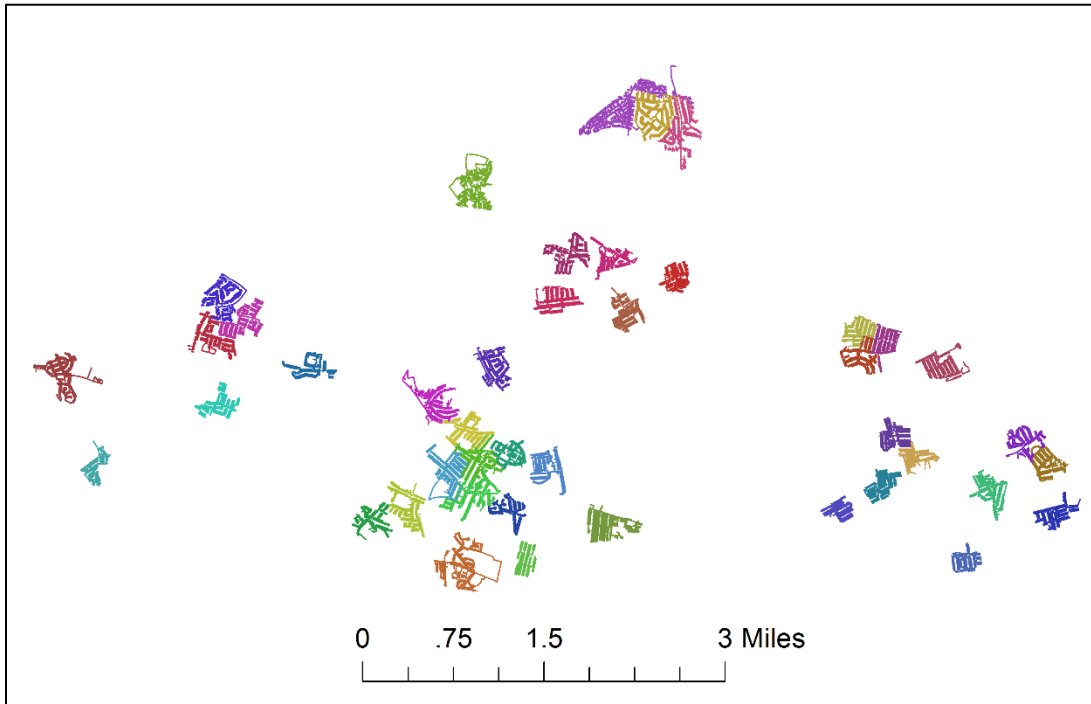




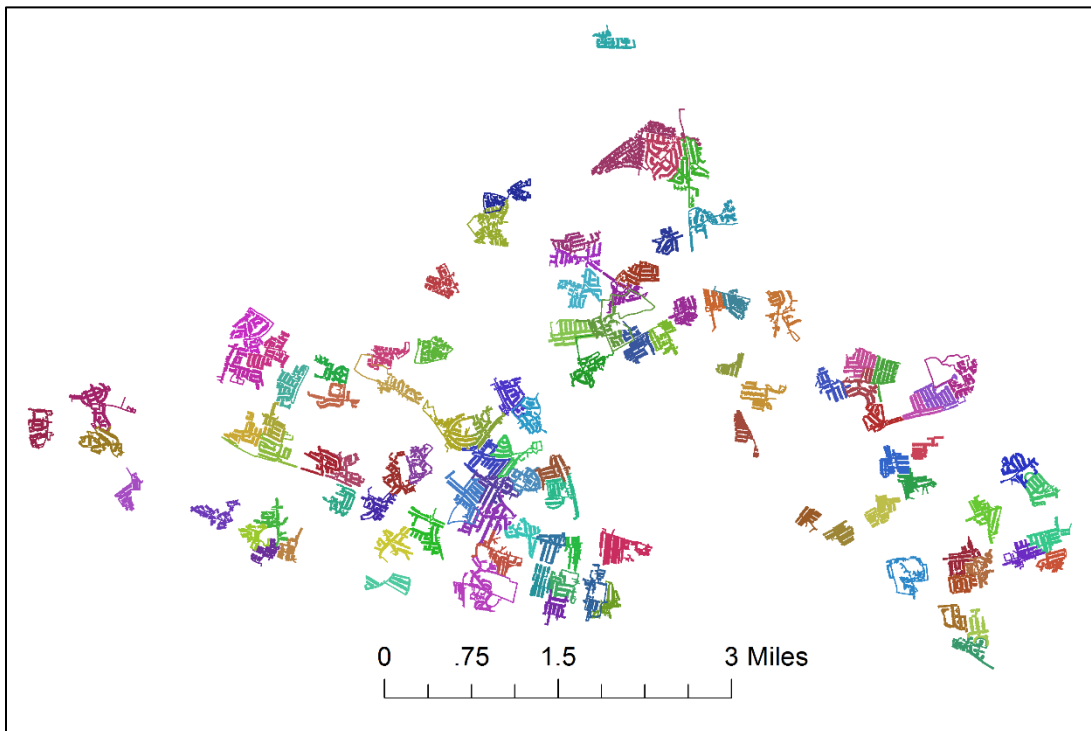
**Figure H8.** Type 2 data set, size 10000. Each colour refers to a single network instance.



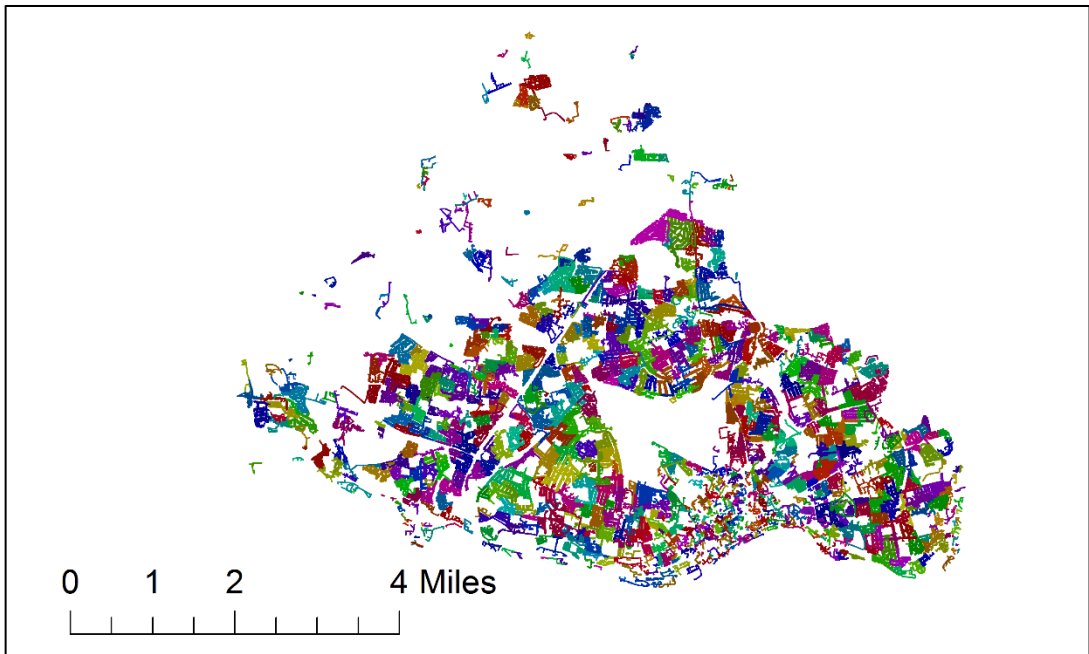
**Figure H9.** Type 2 data set, size 20000. Each colour refers to a single network instance.



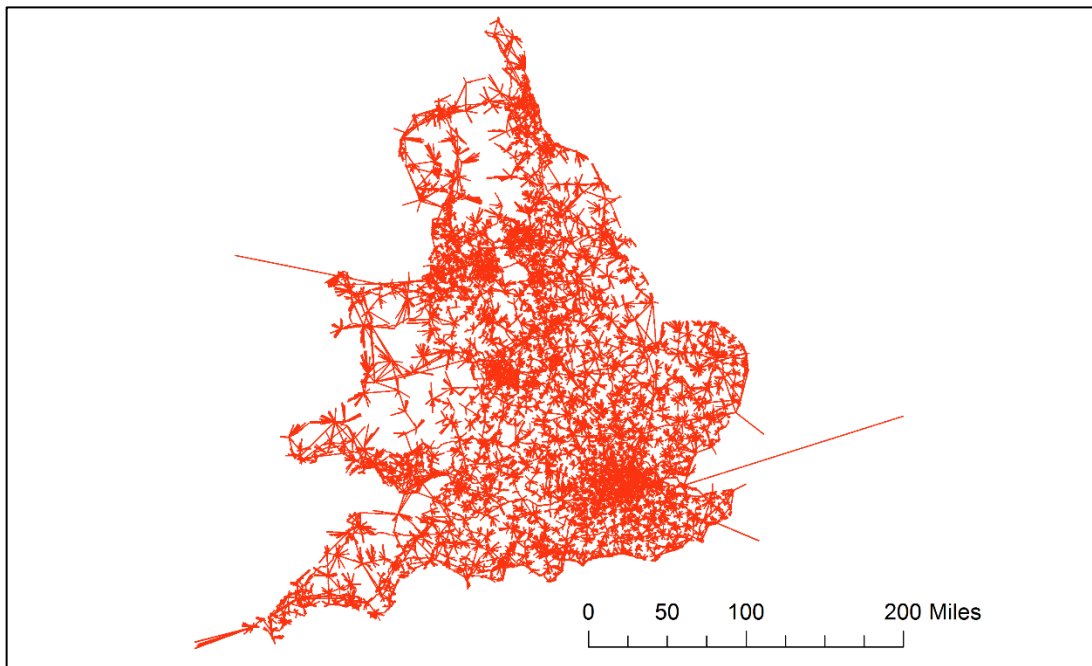
**Figure H10.** Type 2 data set, size 40000. Each colour refers to a single network instance.



**Figure H11.** Type 2 data set, size 80000. Each colour refers to a single network instance.



**Figure H12.** Type 2 data set, size 'Newcastle'. Each colour refers to a single network instance.



**Figure H13.** Type 3 data set, size 'UK'.

## Appendix I – Database Performance Benchmarking Test Result

Network data Size	ITRC Schema	PgRouting	Hybrid Database
100 (Type 1)	1.2	2.7	3.1
200 (Type 1)	2.2	5.2	4.3
400 (Type 1)	4.3	9.6	8.2
800 (Type 1)	8.8	21	15.3
1600 (Type 1)	16	40	29.2
2500 (Type 2)	24	58	41
5000 (Type 2)	51	127	84
10000 (Type 2)	105	266	171
20000 (Type 2)	210	538	338
40000 (Type 2)	430	1107	675
80000 (Type 2)	853	2081	1297
Newcastle (Type 2)	1936	4859	2884
UK (Type 3)	1534	2920	2347

**Table I1.** Execution time (in seconds) of writing different sized network data.

Size	ITRC Schema	PgRouting	Hybrid Database
100 (Type 1)	1.4	1.9	3.1
200 (Type 1)	2.5	3.0	3.2
400 (Type 1)	5.2	4.3	4.5
800 (Type 1)	12.3	8.4	10.8
1600 (Type 1)	27	17	21
2500 (Type 2)	37	26	33
5000 (Type 2)	73	53	61
10000 (Type 2)	159	102	123
20000 (Type 2)	323	199	239

<b>40000 (Type 2)</b>	657	401	472
<b>80000 (Type 2)</b>	1412	808	953
<b>Newcastle (Type 2)</b>	3012	1772	1891
<b>UK (Type 3)</b>	2451	1123	1202

**Table I2.** Execution time (in seconds) of reading different sized network data.

<b>Size</b>	<b>IIRC Schema</b>	<b>PgRouting</b>	<b>Hybrid Database</b>
<b>100 (Type 1)</b>	1.8	2.7	3.1
<b>200 (Type 1)</b>	2.8	4.3	3.2
<b>400 (Type 1)</b>	3.4	5.9	3.5
<b>800 (Type 1)</b>	7.4	8.2	4.8
<b>1600 (Type 1)</b>	19	17	8.1
<b>2500 (Type 2)</b>	30	23	15
<b>5000 (Type 2)</b>	67	42	28
<b>10000 (Type 2)</b>	142	78	49
<b>20000 (Type 2)</b>	288	151	87
<b>40000 (Type 2)</b>	585	287	165
<b>80000 (Type 2)</b>	1142	537	322
<b>Newcastle (Type 2)</b>	2502	945	595
<b>UK (Type 3)</b>	1966	54	38

**Table I3.** Execution time (in seconds) of performing shortest path query on different sized data.

	<b>IIRC schema</b>	<b>PgRouting</b>	<b>Hybrid Database</b>
<b>IRN Complex Query</b>	24,602	5183	2139

**Table I4.** Execution time (in seconds) of performing IRN complex query.

	<b>ITRC schema</b>	<b>PgRouting</b>	<b>Hybrid Database</b>
<b>Task 1</b>	3.7	3.6	5.6
<b>Task 2</b>	204	210	241
<b>Task 3</b>	26	25	37
<b>Task 4</b>	257	261	314

**Table I5.** Execution time (in seconds) of performing complex query on Newcastle Electricity Network.

	<b>ITRC Schema</b>	<b>PgRouting</b>	<b>Hybrid Database</b>
Writing	47688	123961	65322
Reading	64785	23728	29897
Shortest path query	58980	13716	5034

**Table I6.** Execution time (in seconds) of performing writing, reading and shortest path query on London electricity network data.

	<b>ITRC Schema</b>	<b>PgRouting</b>	<b>Hybrid Database</b>
<b>Task 1</b>	2168	2221	2620
<b>Task 2</b>	2205	2227	2561
<b>Task 3</b>	2140	2105	2524
<b>Task 3</b>	2590	2623	2990

**Table I6.** Execution time (in seconds) of performing complex query 1 on London electricity network data.

	<b>ITRC Schema</b>	<b>PgRouting</b>	<b>Hybrid Database</b>
<b>Task 1</b>	21649	9061	3125
<b>Task 2</b>	23155	11793	3507

**Table I7.** Execution time (in seconds) of performing complex query 2 on London electricity network data.

## Appendix J – Scripts for database performance benchmarking tests

This appendix includes the scripts for the database benchmarking tests discussed in chapter 7.

### 1 – Performance test on different size network data (section 7.4)

Writing data with **ITRC schema**

[https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/ITRC/ITRC\\_write.py](https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/ITRC/ITRC_write.py)

Reading data with **ITRC schema**

[https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/ITRC/ITRC\\_read.py](https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/ITRC/ITRC_read.py)

Performing network query with **ITRC schema** (type 1 and type 2 data)

[https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/ITRC/ITRC\\_query\\_1\\_2.py](https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/ITRC/ITRC_query_1_2.py)

Performing network query with **ITRC schema** (type 3 data)

[https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/ITRC/ITRC\\_query\\_3.py](https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/ITRC/ITRC_query_3.py)

Writing data with **PgRouting**

[https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/PgRouting/PgRouting\\_write.py](https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/PgRouting/PgRouting_write.py)

Reading data with **PgRouting**

[https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/PgRouting/PgRouting\\_read.py](https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/PgRouting/PgRouting_read.py)

Performing network query with **PgRouting** (type 1 and type 2 data)

[https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/PgRouting/PgRouting\\_query\\_1\\_2.py](https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/PgRouting/PgRouting_query_1_2.py)

Performing network query with **PgRouting** (type 3 data)

[https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/PgRouting/PgRouting\\_query\\_3.py](https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/PgRouting/PgRouting_query_3.py)

Writing data with **hybrid database**

[https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/Hybrid/Hybrid\\_write.py](https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/Hybrid/Hybrid_write.py)

Reading data with **hybrid database**

[https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/Hybrid/Hybrid\\_read.py](https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/Hybrid/Hybrid_read.py)

Performing network query with **hybrid database** (type 1 and type 2 data)

[https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/Hybrid/Hybrid\\_query\\_1\\_2.py](https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/Hybrid/Hybrid_query_1_2.py)

Performing network query with **hybrid database** (type 3 data)

[https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/Hybrid/Hybrid\\_query\\_3.py](https://github.com/BurningWish/Benchmarking-Scripts/blob/master/1/Hybrid/Hybrid_query_3.py)

## **2 – Performance test on city scale network from Newcastle (section 7.5)**

Complex query on IRN with **ITRC schema**

<https://github.com/BurningWish/Benchmarking-Scripts/tree/master/2/IRN/ITRC>

Complex query on IRN with **PgRouting**

<https://github.com/BurningWish/Benchmarking-Scripts/tree/master/2/IRN/PgRouting>



Complex query on IRN with **hybrid database**

<https://github.com/BurningWish/Benchmarking-Scripts/tree/master/2/IRN/Hybrid>

Complex query on Newcastle electricity network with **ITRC schema**

<https://github.com/BurningWish/Benchmarking-Scripts/tree/master/2/Electricity/ITRC>

Complex query on Newcastle electricity network with **PgRouting**

<https://github.com/BurningWish/Benchmarking-Scripts/tree/master/2/Electricity/PgRouting>

Complex query on Newcastle electricity network with **hybrid database**

<https://github.com/BurningWish/Benchmarking-Scripts/tree/master/2/Electricity/Hybrid>

### **3 – Performance test on mega city scale network from London (section 7.6)**

Write/Read/Network Query on London network with **ITRC schema**

<https://github.com/BurningWish/Benchmarking-Scripts/tree/master/3/simple/ITRC>

Write/Read/Network Query on London network with **PgRouting**

<https://github.com/BurningWish/Benchmarking-Scripts/tree/master/3/simple/PgRouting>

Write/Read/Network Query on London network with **hybrid database**

<https://github.com/BurningWish/Benchmarking-Scripts/tree/master/3/simple/Hybrid>

Complex query 1 on London network with **ITRC schema**

<https://github.com/BurningWish/Benchmarking-Scripts/tree/master/3/complex%20query%201/ITRC>

Complex query 1 on London network with **PgRouting**

<https://github.com/BurningWish/Benchmarking-Scripts/tree/master/3/complex%20query%201/PgRouting>

Complex query 1 on London network with **hybrid database**

<https://github.com/BurningWish/Benchmarking-Scripts/tree/master/3/complex%20query%201/Hybrid>

Complex query 2 on London network with **ITRC schema**

<https://github.com/BurningWish/Benchmarking-Scripts/tree/master/3/complex%20query%202/ITRC>

Complex query 2 on London network with **PgRouting**

<https://github.com/BurningWish/Benchmarking-Scripts/tree/master/3/complex%20query%202/PgRouting>

Complex query 2 on London network with **hybrid database**

<https://github.com/BurningWish/Benchmarking-Scripts/tree/master/3/complex%20query%202/Hybrid>

## References

- Abuzalaf, S., Suttle, C. and Reinbachs, N., Leidos. (2016). Methods and systems for facilitating online collaboration and distribution of geospatial data. U.S. Patent 9,344,466. Available at: <https://patents.google.com/patent/US9344466B1/en> [Accessed 27 May, 2019]
- Adamatzky, A., Allard, O., Jones, J. and Armstrong, R. (2017). Evaluation of French motorway network in relation to slime mould transport networks. *Environment and Planning B: Urban Analytics and City Science*, 44(2), pp.364-383.
- Agarwal, S. and Rajan, K.S. (2017). Analyzing the performance of NoSQL vs. SQL databases for Spatial and Aggregate queries. In 2015 Free and Open Source Software for Geospatial Conference Proceedings, 17(1), p.4. Seoul, South Korea.
- Albaugh, V. and Madduri, H. (2004). The utility metering service of the Universal Management Infrastructure. *IBM Systems Journal*, 43(1), pp.179-189.
- Almeida, M.B. and Barbosa, R.R. (2009). Ontologies in knowledge management support: A case study. *Journal of the American Society for Information Science and Technology*, 60(10), pp.2032-2047.
- Amin, M. (2000). Toward self-healing infrastructure systems. *Computer*, 33(8), pp.44-53.
- Amirian, P., Basiri, A. and Winstanley, A. (2014). Evaluation of data management systems for geospatial big data. In 14<sup>th</sup> International Conference on Computational Science and Its Applications, pp. 678-690. Banff, Canada.
- Apostolakis, G.E. and Lemon, D.M. (2005). A screening methodology for the identification and ranking of infrastructure vulnerabilities due to terrorism. *Risk Analysis: An International Journal*, 25(2), pp.361-376.
- ArangoDB. (2018). NoSQL Performance Benchmark 2018 – MongoDB, PostgreSQL, OrientDB, Neo4j and ArangoDB. Available at: <https://www.arangodb.com/2018/02/nosql-performance-benchmark-2018-mongodb-postgresql-orientdb-neo4j-arangodb/> [Accessed 12 December, 2018]
- Arnell, N., Kram, T., Carter, T., Ebi, K., Edmonds, J., Hallegatte, S., Kriegler, E., Mathur, R., O'Neill, B., Riahi, K. and Winkler, H. (2014). A framework for a new generation of socioeconomic scenarios for climate change impact, adaptation, vulnerability and mitigation research. Working Paper. Available at: <http://hal.cirad.fr/hal-00991870/> [Accessed 12 December, 2018]
- Avi, O. (2014). *Water Distribution Networks*. Springer, Berlin, Germany.
- Bagci, H. and Karagoz, P. (2016). Context-aware location recommendation by using a random walk-based approach. *Knowledge and Information Systems*, 47(2), pp.241-260.
- Banks, J.H. (1989). Freeway speed-flow-concentration relationships: more evidence and interpretations. *Transportation research record*, (1225), pp.53-60.
- Barakou, F., Koukoula, D., Hatziaargyriou, N. and Dimeas, A. (2015). Fractal geometry for

- distribution grid topologies. In 2015 IEEE Eindhoven PowerTech, pp. 1-6. The Netherlands.
- Barbehenn, M. (1998). A note on the complexity of Dijkstra's algorithm for graphs with weighted vertices. *IEEE transactions on computers*, 47(2), p.263.
- Barr, S., Alderson, D., Robson, C., Otto, A., Hall, J., Thacker, S. and Pant, R. (2013). A national scale infrastructure database and modelling environment for the UK, International Symposium for Next Generation Infrastructure. Wollongong, New South Wales, Australia.
- Barr, S., Alderson, D., Ives, M.C. and Robson, C. (2016). Database, simulation modelling and visualisation for national infrastructure assessment. *The Future of National Infrastructure: A System-of-Systems Approach*, Cambridge University Press, p.268.
- Barr, S., Robson, C., Pregolato, M., Ji, Q. (2017). A next generation spatiotemporal database framework for infrastructure systems analytics and modelling. International Symposium for Next Generation Infrastructure. London, UK.
- Baskan, O. (2014). Harmony search algorithm for continuous network design problem with link capacity expansions. *KSCE Journal of Civil Engineering*, 18(1), pp.273-283.
- Bates, P.D., Dawson, R.J., Hall, J.W., Horritt, M.S., Nicholls, R.J., Wicks, J. and Hassan, M.A.A.M. (2005). Simplified two-dimensional numerical modelling of coastal flooding and example applications. *Coastal Engineering*, 52(9), pp.793-810.
- Baykal, B., Tanik, A., and Gonce, I. (2000). Water quality in drinking water reservoirs of a megacity, Istanbul. *Environment Management* 26(6), pp.607-614.
- BBC News. (2012). 'Thunder Thursday' floods cost Newcastle Council £8m. Available at: <https://www.bbc.co.uk/news/uk-england-tyne-23447838> [Accessed 12 December, 2018]
- BBC News. (2019). Urban Environment. Available at: <https://www.bbc.co.uk/bitesize/guides/-zk32pv4/revision/11> [Accessed 12 Nov, 2019]
- Becker, T., Nagel, C. and Kolbe, T.H. (2011). Integrated 3D modelling of multi-utility networks and their interdependencies for critical infrastructure analysis. *Advances in 3D Geo-Information Sciences*, pp.1-20. Springer, Berlin, Heidelberg.
- Becker, T., Nagel, C. and Kolbe, T.H. (2013). Semantic 3D modeling of multi-utility networks in cities for analysis and 3D visualization. *Progress and New Trends in 3D Geoinformation Sciences*, pp. 41-62. Springer, Berlin, Heidelberg.
- Bell, M.G. and Iida, Y. (1997). *Transportation network analysis*. John Wiley & Sons, Ltd. Chichester, UK.
- Benenson, I. (2004). Agent-based modeling: From individual residential choice to urban residential dynamics. *Spatially integrated social science: Examples in best practice*, 42(6-7), pp.67-95.
- Bennet, J. (2005). The agency of assemblages and the North American blackout. Available at: [https://jscholarship.library.jhu.edu/bitstream/handle/1774.2/-32808/bennet\\_public\\_culture.pdf](https://jscholarship.library.jhu.edu/bitstream/handle/1774.2/-32808/bennet_public_culture.pdf) [Accessed 12 December, 2018]
- Berkowicz, R. (2000). OSPM-A parameterised street pollution model. *Environmental monitoring and*

- assessment, 65(1-2), pp.323-331.
- Bettencourt, L.M., Lobo, J., Helbing, D., Kühnert, C. and West, G.B. (2007). Growth, innovation, scaling, and the pace of life in cities. *Proceedings of the national academy of sciences*, 104(17), pp.7301-7306.
- Berche, B., von Ferber, C., Holovatch, T. and Holovatch, Y. (2009). Resilience of public transport networks against attacks. *The European Physical Journal B*, 71(1), pp.125-137.
- Berdier, C. and Roussey, C. (2007). Urban ontologies: The towntology prototype towards case studies. *Ontologies for Urban Development*, pp. 143-155. Springer, Berlin, Heidelberg.
- Bertsch, R., Glenis, V. and Kilsby, C. (2017). Urban flood simulation using synthetic storm drain networks. *Water*, 9(12), p.925.
- Bezbradica, M. and Ruskin, H.J. (2019). Understanding Urban Mobility and Pedestrian Movement. Available at: <https://www.intechopen.com/online-first/understanding-urban-mobility-and-pedestrian-movement> [Accessed 13 Nov, 2019]
- Blokker, E.J.M., Vreeburg, J.H.G. and Van Dijk, J.C. (2009). Simulating residential water demand with a stochastic end-use model. *Journal of Water Resources Planning and Management*, 136(1), pp.19-26.
- Bon, M. (2017). An advanced prospecting method for assessing the quantity of underground metal cables in urban mines. Master Dissertation. Delft University, The Netherlands. Available at: <https://repository.tudelft.nl/islandora/object/uuid%3A2f2c6f35-ad29-4d7d-b382-232b7bbb4cbe> [Accessed 12 December, 2018]
- Borden, K.A., Schmidlein, M.C., Emrich, C.T., Piegorsch, W.W. and Cutter, S.L. (2007). Vulnerability of US cities to environmental hazards. *Journal of Homeland Security and Emergency Management*, 4(2), pp.1-21.
- Borrmann, A., Schraufstetter, S. and Rank, E. (2009). Implementing metric operators of a spatial query language for 3D building models: octree and B-Rep approaches. *Journal of Computing in Civil Engineering*, 23(1), pp.34-46.
- Bozza, A., Asprone, D., and Manfredi, G. (2017). Physical Resilience in Cities. *Oxford Research Encyclopaedia of Natural Hazard Science*. Available at: <http://oxfordre.com/naturalhazard-science/view/10.1093/acrefore/9780199389407.001.0001/acrefore-9780199389407-e-83>[Accessed 12 December, 2018]
- Brandes U., Erlebach T. (2005). Fundamentals. Network Analysis. *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg.
- British Telecom. (2012). Use of Oracle Spatial Network Data Model at British Telecom. Available at: [http://download.oracle.com/otndocs/products/spatial/pdf/british\\_telecom\\_ug.pdf](http://download.oracle.com/otndocs/products/spatial/pdf/british_telecom_ug.pdf) [Accessed 12 December, 2018]
- Buurman, J., Mens, M.J. and Dahm, R.J. (2017). Strategies for urban drought risk management: a comparison of 10 large cities. *International journal of water resources development*, 33(1), pp.31-50.

- Cabinet Office. (2008). Learning lessons from the 2007 flood. Available at: [https://webarchive.-nationalarchives.gov.uk/20100702215619/http://archive.cabinetoffice.gov.uk/pittreview/the-pitt-review/final\\_report.html](https://webarchive.-nationalarchives.gov.uk/20100702215619/http://archive.cabinetoffice.gov.uk/pittreview/the-pitt-review/final_report.html) [Accessed 22 May, 2019]
- Campos, C., Leitão, J.M., Pereira, J.P., Ribas, A. and Coelho, A.F. (2015). Procedural generation of topologic road networks for driving simulation. In 2015 10th Iberian Conference on Information Systems and Technologies (CISTI), pp.1-6. Aveiro, Portugal.
- Cantarella, G.E. and Vitetta, A. (2006). The multi-criteria road network design problem in an urban area. *Transportation*, 33(6), pp.567-588.
- Carreño, M., Cardona, O. and Barbat, A. (2007). Neuro-fuzzy assessment of building damage and safety after an earthquake. *Intelligent computational paradigms in earthquake engineering*, pp.123-157. IDEA group publishing. Hershey, London, UK.
- Castle, C.J. and Crooks, A.T. (2006). Principles and concepts of agent-based modelling for developing geospatial simulations. Available at: <http://discovery.ucl.ac.uk/3342/> [Accessed 12 December, 2018]
- Castles, S., De Haas, H, and Miller, M.J. (2013). *The age of migration: International population movements in the modern world*. Macmillan International Higher Education. Basingstoke, UK.
- Cattuto, C., Quaghiotto, M., Panisson, A. and Averbuch, A. (2013). Time-varying social networks in a graph database: a Neo4j use case. In 1st international workshop on graph data management experiences and systems, p. 11. New York, NY, USA.
- Cavallaro, M., Asprone, D., Latora, V., Manfredi, G. and Nicosia, V. (2014). Assessment of urban ecosystem resilience through hybrid social – physical complex networks. *Computer - Aided Civil and Infrastructure Engineering*, 29(8), pp.608-625.
- Chaikin, G.M. (1974). An algorithm for high-speed curve generation. *Computer graphics and image processing*, 3(4), pp.346-349.
- Chang, S.E., McDaniels, T.L., Mikawoz, J. and Peterson, K. (2007). Infrastructure failure interdependencies in extreme events: power outage consequences in the 1998 Ice Storm. *Natural Hazards*, 41(2), pp.337-358.
- Chew, L.P. (1989). Constrained delaunay triangulations. *Algorithmica*, 4(1-4), pp.97-108.
- Chiu, Y.C., Zheng, H., Villalobos, J. and Gautam, B. (2007). Modeling no-notice mass evacuation using a dynamic traffic flow optimization model. *IIE Transactions*, 39(1), pp.83-94.
- Cimellaro, G.P., Reinhorn, A.M. and Bruneau, M. (2010). Seismic resilience of a hospital system. *Structure and Infrastructure Engineering*, 6(1-2), pp.127-144.
- City Mayors. (2018). The UK's 200 largest towns, cities, and districts. Available at: [http://www.citymayors.com/gratis/uk\\_topcities.html](http://www.citymayors.com/gratis/uk_topcities.html) [Accessed December, 2018]
- Coutinho-Rodrigues, J., Simão, A. and Antunes, C.H. (2011). A GIS-based multicriteria spatial decision support system for planning urban infrastructures. *Decision Support Systems*, 51(3), pp.720-726.

- Couclelis, H. (2002). Modeling frameworks, paradigms, and approaches. *Geographic information systems and environmental modelling*, Prentice Hall, London.
- D'Agostino, G. and Scala, A. (2014). *Networks of networks: the last frontier of complexity*. Springer, Berlin.
- Das, P., Parida, M. and Katiyar, V.K. (2015). Analysis of interrelationship between pedestrian flow parameters using artificial neural network. *Journal of Modern Transportation*, 23(4), pp.298-309.
- Davis, C.A. (2014). Water system service categories, post-earthquake interaction, and restoration strategies. *Earthquake Spectra*, 30(4), pp.1487-1509.
- Delamater, P.L., Messina, J.P., Shortridge, A.M. and Grady, S.C. (2012). Measuring geographic access to health care: raster and network-based methods. *International journal of health geographics*, 11(1), pp.15.
- Delling, D., Sanders, P., Schultes, D. and Wagner, D. (2009). Engineering route planning algorithms. *Algorithmics of large and complex networks*. Springer, Berlin, Heidelberg.
- Deng, X., Huang, J., Rozelle, S. and Uchida, E. (2010). Economic growth and the expansion of urban land in China. *Urban studies*, 47(4), pp.813-843.
- Department for Transport. (2005). FOI Request - Road numbering. Available at: <https://webarchive.nationalarchives.gov.uk/20070403181235/http://www.dft.gov.uk/foi/responses/2005/aug/roadnumbering/letteraboutroadclassification> [Accessed 12 December, 2018]
- Deshmukh, A., Ho Oh, E. and Hastak, M. (2011). Impact of flood damaged critical infrastructure on communities and industries. *Built Environment Project and Asset Management*, 1(2), pp.156-175.
- Domingo, M., Thibaud, R. and Claramunt, C. (2019). A graph-based approach for the structural analysis of road and building layouts. *Geo-spatial Information Science*, 22(1), pp.59-72.
- Dorbritz, R. (2011). Assessing the resilience of transportation systems in case of large-scale disastrous events. In *2011 Proceedings of the International Conference on Environmental Engineering*, pp.1070-1076. Vilnius, Lithuania.
- Duan, Z.Y. and Wang, Q. (2009). Road network analysis and evaluation of Huizhou city based on space syntax. In *2009 International conference on measuring technology and mechatronics automation*, pp. 579-582, Zhangjiajie, Hunan, China.
- Dudenhoeffer, D.D., Permann, M.R. and Manic, M. (2006). CIMS: A framework for infrastructure interdependency modelling and analysis. In *2006 Proceedings of the 38th conference on Winter Simulation*, pp. 478-485. Monterey, CA, USA.
- Dunn, S., Fu, G., Wilkinson, S., and Dawson, R. (2013). Network theory for infrastructure systems modelling. *Proceedings of the ICE-Engineering Sustainability*, 166(5), pp.281-292.
- Dunn, S., Wilkinson, S. and Ford, A. (2016). Spatial structure and evolution of infrastructure networks. *Sustainable Cities and Society*, 27, pp.23-31.

- Dutch Ministry of Infrastructure and Environment. (2014). Dutch programmes on next generation infrastructure and knowledge for climate. Available at: <http://www.knowledgeforclimate.nl/-/infrastructure> [Accessed 12 December, 2018]
- Dueñas-Osorio, L., Craig, J.I., Goodno, B.J. and Bostrom, A. (2007). Interdependent response of networked systems. *Journal of Infrastructure Systems*, 13(3), pp.185-194.
- Dunn, S., Fu, G., Wilkinson, S. and Dawson, R. (2013). Network theory for infrastructure systems modelling. *Proceedings of the Institution of Civil Engineers-Engineering Sustainability*, 166(5), pp. 281-292.
- Egger, S. (2006). Determining a sustainable city model. *Environmental Modelling & Software*, 21(9), pp.1235-1246.
- ELCON. (2004). The Economic Impacts of the August 2003 Blackout. Available at: <https://elcon.-org/wpcontent/uploads/Economic20Impacts20of20August20200320Blackout1.pdf> [Accessed 12 December, 2018]
- Electricity Consumer Resource Council. (2004). The Economic Impacts of the August 2003 Blackout. Available at: <https://www.nrc.gov/docs/ML1113/ML111300584.pdf> [Accessed 5 July, 2017]
- ESRI. (2015). Utility & Pipeline Data Model. Available at: [http://proceedings.esri.com/library/-userconf/eguc15/papers/eguc\\_61.pdf](http://proceedings.esri.com/library/-userconf/eguc15/papers/eguc_61.pdf) [Accessed 12 December, 2018]
- European Union. (2010). Demography Report 2010. Accessible at: <http://ec.europa.eu/social/-BlobServlet?docId=6824> [Accessed 5 July, 2017]
- Facilities Working Group. (2000). Utility Data Content Standard. Available at: <https://www.fgdc.-gov/standards/projects/utilities/utilities.pdf> [Accessed 12 December, 2018]
- Fan, W. (2012). Graph pattern matching revised for social network analysis. In 2012 Proceedings of the 15<sup>th</sup> International Conference on Database Theory, pp.8-21. Berlin, Germany.
- Fang, Y.P., Pedroni, N. and Zio, E. (2016). Resilience-based component importance measures for critical infrastructure network systems. *IEEE Transactions on Reliability*, 65(2), pp.502-512.
- Feng, Y. Y., Chen, S. Q., and Zhang, L. X. (2013). System dynamics modelling for urban energy consumption and CO 2 emissions: a case study of Beijing, China. *Ecological Modelling*, 252, pp.44-52.
- Ferrari, G., and Becciu, G. (2011). Re-design of water distribution networks using hybrid optimization. Available at: <https://core.ac.uk/download/pdf/55216596.pdf> [Accessed 12 December, 2018]
- Ferro, N. and Sinico, L. (2018). Graph Databases Benchmarking on the Italian Business Register. In SEBD 2018, Castellana Marina, Italy. Available at: <http://ceur-ws.org/Vol-2161/paper43.pdf> [Accessed 12 December, 2018]
- Fiedler, P.E.K. and Zannin, P.H.T. (2015). Evaluation of noise pollution in urban traffic hubs—Noise maps and measurements. *Environmental Impact Assessment Review*, 51, pp.1-9.



- Fiduccia, C.M. and Mattheyses, R.M. (1982). A linear-time heuristic for improving network partitions. In Proceedings of the 19th design automation conference, pp. 175-181. Piscataway, NJ, USA.
- Fikejz, J. and Řezanina, E. (2016). The Design of Railway Network Infrastructure Model for Localization of Rolling Stock with Utilization Technology Oracle Spatial and Dynamic Database Views. *Advanced Computer and Communication Engineering Technology*, p.935, Springer, Berlin.
- Fonoberova, M., Mezić, I., Mezić, J. and Mohr, R. (2018). An agent-based model of urban insurgence: Effect of gathering sites and Koopman mode analysis. *PloS one*, 13(10), p.e0205259.
- Franchin, P. and Cavalieri, F. (2015). Probabilistic assessment of civil infrastructure resilience to earthquakes. *Computer - Aided Civil and Infrastructure Engineering*, 30(7), pp.583-600.
- Freckleton, D., Heaslip, K., Louisell, W. and Collura, J. (2012). Evaluation of resiliency of transportation networks after disasters. *Transportation research record*, 2284(1), pp.109-116.
- Fu, G. and Cohn, A.G. (2008). Utility Ontology Development with Formal Concept Analysis. In 2008 Proceedings of 5th International Conference on Formal Ontology in Information System, pp.297-310. Amsterdam, The Netherlands.
- Fuglsang, M., Hansen, H.S. and Münier, B. (2011). Accessibility analysis and modelling in public transport networks – a raster based approach. In 2011 International Conference on Computational Science and Its Applications, pp. 207-224. Nanyang Technological University, Singapore.
- Fujii, Y., Satake, K., Sakai, S.I., Shinohara, M. and Kanazawa, T. (2011). Tsunami source of the 2011 off the Pacific coast of Tohoku Earthquake. *Earth, planets and space*, 63(7), p.55.
- Fügenschuh, A., Geißler, B., Gollmer, R., Morsi, A., Rövekamp, J., Schmidt, M., Spreckelsen, K. and Steinbach, M.C. (2015). Chapter 2: Physical and technical fundamentals of gas networks. *Evaluating Gas Network Capacities*, pp.17-43. Society for Industrial and Applied Mathematics. Philadelphia, USA.
- Gabrys, J. (2014). Programming environments: environmentality and citizen sensing in the smart city. *Environment and Planning D: Society and Space*, 32(1), pp.30-48.
- Gastner, M.T. and Newman, M.E.J. (2006). Optimal design of spatial distribution networks. *Physical Review E*, 74(1), p.016117.
- Ghosh, I., Hellweger, F.L. and Fritch, T.G. (2006). Fractal generation of artificial sewer networks for hydrologic simulations, pp.7-11. In 2006 Proceedings of the ESRI International User Conference, San Diego, California, USA.
- Giustolisi, O., Laucelli, D., Berardi, L. and Savić, D.A. (2011). Computationally efficient modeling method for large water network analysis. *Journal of Hydraulic Engineering*, 138(4), pp.313-326.
- Glenis, V., McGough, A.S., Kutija, V., Kilsby, C. and Woodman, S. (2013). Flood modelling for cities using Cloud computing. *Journal of Cloud Computing: Advances, Systems and*

Applications, 2(1), p.7.

- Gober, P., Wentz, E.A., Lant, T., Tschudi, M.K. and Kirkwood, C.W. (2011). WaterSim: a simulation model for urban water planning in Phoenix, Arizona, USA. *Environment and Planning B: Planning and Design*, 38(2), pp.197-215.
- Gobiet, A., Baumgartner, D., Krobath, T., Maderbacher, R. and Putz, E. (2000). Urban air pollution monitoring with DOAS considering the local meteorological situation. *Urban Air Quality: Measurement, Modelling and Management*, pp. 119-127. Springer, Dordrecht.
- Godschalk, D.R. (2003). Urban hazard mitigation: creating resilient cities. *Natural hazards review*, 4(3), pp.136-143.
- Gudmundsson, M.T., Pedersen, R., Vogfjörð, K., Thorbjarnardóttir, B., Jakobsdóttir, S. and Roberts, M.J. (2010). Eruptions of Eyjafjallajökull Volcano, Iceland. *Eos, Transactions American Geophysical Union*, 91(21), pp.190-191.
- Guerrieri, V., Hartley, D. and Hurst, E. (2013). Endogenous gentrification and housing price dynamics. *Journal of Public Economics*, 100, pp.45-60.
- Guikema, S.D. (2009). Natural disaster risk analysis for critical infrastructure systems: An approach based on statistical learning theory. *Reliability Engineering & System Safety*, 94(4), pp.855-860.
- Guisasola, A., de Haas, D., Keller, J. and Yuan, Z. (2008). Methane formation in sewer systems. *Water Research*, 42(6-7), pp.1421-1430.
- Gurjar, B.R., Butler, T.M., Lawrence, M.G. and Lelieveld, J. (2008). Evaluation of emissions and air quality in megacities. *Atmospheric Environment*, 42(7), pp.1593-1606.
- Gurung, T.R., Stewart, R.A., Beal, C.D. and Sharma, A.K. (2015). Smart meter enabled water end-use demand data: platform for the enhanced infrastructure planning of contemporary urban water supply networks. *Journal of Cleaner Production*, 87, pp.642-654.
- Gruber T.R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* 5(2), 199-220.
- Hadas, Y. and Laor, A. (2013). Network design model with evacuation constraints. *Transportation research part A: policy and practice*, 47, pp.1-9.
- Haider, A. (2013). *Information Systems for Engineering and Infrastructure Asset Management*. Springer, Berlin.
- Halfawy, M.R., Dridi, L. and Baker, S. (2008). Integrated decision support system for optimal renewal planning of sewer networks. *Journal of Computing in Civil Engineering*, 22(6), pp.360-372.
- Halicioglu, F., Andrés, A.R. and Yamamura, E. (2012). Modelling crime in Japan. *Economic Modelling*, 29(5), pp.1640-1645.
- Hall, J.W., Tran, M., Hickford, A.J. and Nicholls, R.J. (2016). *The future of national infrastructure: A system-of-systems approach*. Cambridge University Press, UK.

- Hamed, M.M., Khalafallah, M.G. and Hassanien, E.A. (2004). Prediction of wastewater treatment plant performance using artificial neural networks. *Environmental Modelling & Software*, 19(10), pp.919-928.
- Hammer, M.J. (1986). *Water and wastewater technology*. Upper Saddle River, New Jersey, USA.
- Hancke, G., Silva, B. and Hancke Jr, G. (2013). The role of advanced sensing in smart cities. *Sensors*, 13(1), pp.393-425.
- Have, C.T. and Jensen, L.J. (2013). Are graph databases ready for bioinformatics? *Bioinformatics*, 29(24), p.3107.
- Heijnen, P., Chappin, E. and Nikolic, I. (2014). Infrastructure Network Design with a Multi-Model Approach: Comparing Geometric Graph Theory with an Agent-Based Implementation of an Ant Colony Optimization. *Journal of Artificial Societies and Social Simulation*, 17(4), p.1.
- Hendler, J. (2001). Agents and the semantic web. *IEEE Intelligent systems*, 16(2), pp.30-37.
- Hepp, M. (2007). Possible ontologies: How reality constrains the development of relevant ontologies. *IEEE Internet Computing*, 11(1), pp.90-96.
- Hillier, B. and Hanson, J. (1989). *The social logic of space*. Cambridge university press. UK.
- Hillier, B. and Iida, S. (2005). Network and psychological effects in urban movement. *International Conference on Spatial Information Theory*, pp. 475-490.
- HM Treasury. (2014). Government report, HM Treasury. Available at: <https://www.gov.uk/government/publications/national-infrastructure-plan-2014> [Accessed 12 December, 2012]
- Hokstad, P., Utne, I.B. and Vatn, J. (2012). *Risk and interdependencies in critical infrastructures*. Springer, London.
- Holmgren, Å.J. (2006). Using graph models to analyze the vulnerability of electric power networks. *Risk analysis*, 26(4), pp.955-969.
- Holt, T. and Pullen, J. (2007). Urban canopy modelling of the New York City metropolitan area: A comparison and validation of single-and multilayer parameterizations. *Monthly Weather Review*, 135(5), pp.1906-1930.
- Hoornweg, D. and Pope, K. (2017). Population predictions for the world's largest cities in the 21<sup>st</sup> century. *Environment and Urbanization*, 29(1), pp.195-216.
- Howard, B., Parshall, L., Thompson, J., Hammer, S., Dickinson, J., and Modi, V. (2012). Spatial distribution of urban building energy consumption by end use. *Energy and Buildings*, 45, pp.141-151.
- Howell, S., Rezgui, Y. and Beach, T. (2018). Water utility decision support through the semantic web of things. *Environmental Modelling & Software*, 102(C), pp.94-114.
- Hu, F., Yeung, C.H., Yang, S., Wang, W. and Zeng, A. (2016). Recovery of infrastructure networks after localised attacks. *Scientific reports*, 6, p.24522.
- Hu, Y., Liu, X., Bai, J., Shih, K., Zeng, E.Y. and Cheng, H. (2013). Assessing heavy metal pollution

- in the surface soils of a region that had undergone three decades of intense industrialization and urbanization. *Environmental Science and Pollution Research*, 20(9), pp.6150-6159.
- Ibanez, E. and McCalley, J.D. (2011). Multiobjective evolutionary algorithm for long-term planning of the national energy and transportation systems. *Energy Systems*, 2(2), pp.151-169.
- INSPIRE. (2013). INSPIRE data specification on Utility and Government Services. Available at: <https://inspire.ec.europa.eu/id/document/tg/us> [Accessed 12 December, 2018]
- INSPIRE. (2013). INSPIRE data specification on Transport Networks. Available at: <https://inspire.ec.europa.eu/Themes/115/2892> [Accessed 12 December, 2018]
- Jacyna, M., Wasiak, M., Lewczuk, K. and Kłodawski, M. (2014). Simulation model of transport system of Poland as a tool for developing sustainable transport. *Archives of Transport*, 31(3), pp.23-35.
- Jaw, S.W. and Hashim, M. (2013). Locational accuracy of underground utility mapping using ground penetrating radar. *Tunnelling and Underground Space Technology*, 35, pp.20-29.
- Jeffers, S. (2017). Using Fractal Geometries to Understand Urban Drainage Networks and Green Stormwater Infrastructure Development. Drexel University. PhD Thesis. Available at: <https://idea.library.drexel.edu/islandora/object/idea%3A7555> [Accessed 10 November, 2019]
- Jenks, M. and Jones, C. (2009). *Dimensions of the sustainable city*. Springer Science & Business Media.
- Jha, A. K., Bloch, R., & Lamond, J. (2012). *Cities and flooding: a guide to integrated urban flood risk management for the 21st century*. World Bank Publications. Washington, D.C. USA.
- Ji, Q. (2019). Geospatial inference and management of utility infrastructure networks. Doctoral Dissertation. Newcastle University, UK.
- Ji, Q., Barr, S., James, P., and Fairbairn, D. (2017). A heuristic spatial algorithm for generating fine-scale infrastructure distribution networks. In 25<sup>th</sup> GISRUK 2017, Manchester, UK.
- Ji, Q., Barr, S., James, P., and Fairbarin, D. (2018). Graph database implementation of fine spatial scale urban infrastructure networks. In 26<sup>th</sup> GISRUK 2018, Leicester, UK.
- Johnson, C.W. (2007). Analysing the causes of the Italian and Swiss blackout, 28<sup>th</sup> September 2003. In 2007 Proceedings of the 12<sup>th</sup> Australian workshop on Safety critical systems and software and safety-related programmable systems, 86, pp. 21-30. Darlinghurst, Australia.
- Jung, M.G., Youn, S.A., Bae, J. and Choi, Y.L. (2015). A study on data input and output performance comparison of MongoDB and PostgreSQL in the big data environment. In 2015 8<sup>th</sup> International Conference on Database Theory and Application (DTA), pp.14-17. Jeju Island, South Korea.
- Kaiser, E.J., Godschalk, D.R. and Chapin, F.S. (1995). *Urban land use planning*. Urbana, IL: University of Illinois Press. Available at: <https://journals.sagepub.com/doi/abs/10.1177/-0739456X9501500107?journalCode=jpea> [Accessed 12 December, 2018]
- Karnouskos, S., Terzidis, O. and Karnouskos, P. (2007). An advanced metering infrastructure for future energy networks. *New Technologies, Mobility and Security*. Springer, Dordrecht, The

Netherlands.

- Kamilaris, A. and Ostermann, F. (2018). Geospatial analysis and the Internet of Things. *ISPRS international journal of geo-information*, 7(7), p.269.
- Katsumi, M., and Fox, X. (2017). Using the iCity Ontology. Available at: [https://uttri.utoronto.ca/files/2017/06/Katsumi\\_Using-the-iCity-Ontology.pdf](https://uttri.utoronto.ca/files/2017/06/Katsumi_Using-the-iCity-Ontology.pdf) [Accessed 12 December, 2018]
- Katsumi, M. and Fox, M. (2018). Ontologies for transportation research: A survey. *Transportation Research Part C: Emerging Technologies*, 89, pp.53-82.
- Kavgic, M., Mavrogianni, A., Mumovic, D., Summerfield, A., Stevanovic, Z. and Djurovic-Petrovic, M. (2010). A review of bottom-up building stock models for energy consumption in the residential sector. *Building and environment*, 45(7), pp.1683-1697.
- Kc, S., Barakat, B., Goujon, A., Skirbekk, V., Sanderson, W. and Lutz, W. (2010). Projection of populations by level of educational attainment, age, and sex for 120 countries for 2005-2050. *Demographic research*, 22, pp.383-472.
- Kiel, J., Petiet, P., Nieuwenhuis, A., Peters, T. and van Ruiten, K. (2016). A decision support system for the resilience of critical transport infrastructure to extreme weather events. *Transportation research procedia*, 14, pp.68-77.
- Klein, R.J., Nicholls, R.J. and Thomalla, F. (2003). Resilience to natural hazards: How useful is this concept? *Global Environmental Change Part B: Environmental Hazards*, 5(1), pp.35-45.
- Kleissl, J. and Agarwal, Y. (2010). Cyber-physical energy systems: Focus on smart buildings. In 2010 Design Automation Conference, pp. 749-754. Anaheim, CA, USA.
- Kourtit, K. and Nijkamp, P. (2013). In praise of megacities in a global world. *Regional Science Policy & Practice* 5(2), pp.167-182.
- Krivo, L.J. and Peterson, R.D. (1996). Extremely disadvantaged neighbourhoods and urban crime. *Social forces*, 75(2), pp.619-648.
- Krishna, K. and Murty, N.M. (1999). Genetic K-means algorithm. *IEEE Transactions on Systems Man and Cybernetics-Part B: Cybernetics*, 29(3), pp.433-439.
- Kyte, M., Khatib, Z., Shannon, P. and Kitchener, F. (2001). Effect of weather on free-flow speed. *Transportation research record: Journal of the transportation research board*, (1776), pp.60-68.
- Laefer, D.F., Koss, A. and Pradhan, A. (2006). The need for baseline data characteristics for GIS-based disaster management systems. *Journal of urban planning and development*, 132(3), pp.115-119.
- Lakervi, E. and Holmes, E.J. (1995). *Electricity distribution network design*. Peter Peregrines Ltd, London, UK.
- Lara, A.P., Da Costa, E.M., Furlani, T.Z. and Yigitcanla, T. (2016). Smartness that matters: towards a comprehensive and human-centred characterisation of smart cities. *Journal of Open Innovation: Technology, Market, and Complexity*, 2(2), p.8.

- Lau, S.K., Zamani, R. and Susilo, W. (2016). A semantic web vision for an intelligent community transport service brokering system. In 2016 IEEE International Conference on Intelligent Transportation Engineering, pp. 172-175. Singapore.
- LeGates, R.T. and Stout, F. (2015). *The city reader*. Routledge, UK.
- Leu, G., Abbass, H. and Curtis, N. (2010). Resilience of ground transportation networks: a case study on Melbourne. Available at: [https://www.researchgate.net/publication/228926300\\_Resilience\\_of\\_ground\\_transportation\\_networks\\_A\\_case\\_study\\_on\\_Melbourne](https://www.researchgate.net/publication/228926300_Resilience_of_ground_transportation_networks_A_case_study_on_Melbourne) [Accessed 12 December, 2018]
- Leung, N.K., Lau, S.K. and Tsang, N. (2013). An ontology-based collaborative inter-organisational knowledge management network (CIK-NET). *Journal of Information & Knowledge Management*, 12(01), p.1350005.
- Leavitt, W.M. and Kiefer, J.J. (2006). Infrastructure interdependency and the creation of a normal disaster: The case of Hurricane Katrina and the city of New Orleans. *Public works management & policy*, 10(4), pp.306-314
- Leskens, J.G., Kehl, C., Tutenel, T., Kol, T., De Haan, G., Stelling, G. and Eisemann, E. (2017). An interactive simulation and visualization tool for flood analysis usable for practitioners. *Mitigation and adaptation strategies for global change*, 22(2), pp.307-324.
- Lhomme, S., Serre, D., Diab, Y. and Laganier, R. (2013). Analyzing resilience of urban networks: a preliminary step towards more flood resilient cities. *Natural hazards and earth system sciences*, 13(2), pp.221-230.
- Li, B., Springer, J., Bebis, G. and Gunes, M.H. (2013). A survey of network flow applications. *Journal of Network and Computer Applications*, 36(2), pp.567-581.
- Liebich, T. (2009). IFC 2x edition 3. Available at: <http://www.buildingsmarttech.org/downloads/-accompanyingdocuments/guidelines/IFC2x%20Model%20Implementation%20Guide%20V2-0b.pdf/view> [Accessed 12 December, 2018]
- Lin, S., De Schutter, B., Hegyi, A., Xi, Y., and Hellendoorn, H. (2014). On a spatiotemporally discrete urban traffic model. *Intelligent Transport Systems*, 8(3), 219-231.
- Lin, S. and Kernighan, B.W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2), pp.498-516.
- Liu, Y. (2013). Labour market matching and unemployment in urban China. *China Economic Review*, 24, pp.108-128.
- Lin, Z.Q., Xie, B., Zou, Y.Z., Zhao, J.F., Li, X.D., Wei, J., Sun, H.L. and Yin, G. (2017). Intelligent development environment and software knowledge graph. *Journal of Computer Science and Technology*, 32(2), pp.242-249.
- Lombardi, P., Giordano, S., Farouh, H. and Yousef, W. (2012). Modelling the smart city performance. *Innovation: The European Journal of Social Science Research*, 25(2), pp.137-149.
- Lorenz, B., Ohlbach, H.J. and Yang, L. (2005). Ontology of transportation networks. Available at:

- <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.133.2953> [Accessed 12 December, 2018]
- Lucas, S.A., Coombes, P.J. and Sharma, A.K. (2010). The impact of diurnal water use patterns, demand management and rainwater tanks on water supply network design. *Water Science and Technology: Water Supply*, 10(1), pp.69-80.
- Lutz, W. and KC, S. (2011). SSP Population Projections—Assumptions and Methods. Supplementary Note for the SSP Data Sets. Available at: [https://tntcat.iiasa.ac.at/SspDb/static/download/-ssp\\_supplementary%20text.pdf](https://tntcat.iiasa.ac.at/SspDb/static/download/-ssp_supplementary%20text.pdf) [Accessed 12 December, 2018]
- Masucci, D., Palazzo, C., Foglietta, C. and Panzieri, S. (2016). Enhancing decision support with interdependency modelling. In 10<sup>th</sup> International Conference on Critical Infrastructure Protection, pp. 169-183. Virginia, USA.
- Magnanti, T.L. and Wong, R.T. (1984). Network design and transportation planning: Models and algorithms. *Transportation science*, 18(1), pp.1-55.
- Maislos, A. (2017). Hybrid Databases: Combining Relational and NoSQL. Available at: <https://www.stratoscale.com/blog/dbaas/hybrid-databases-combining-relational-nosql/> [Accessed 12 December, 2018]
- Malekpour, S., Brown, R.R. and de Haan, F.J. (2015). Strategic planning of urban infrastructure for environmental sustainability: Understanding the past to intervene for the future. *Cities*, 46, pp.67-75.
- Malleson, N., Evans, A. and Jenkins, T. (2009). An agent-based model of burglary. *Environment and Planning B: Planning and Design*, 36(6), pp.1103-1123.
- Mao, R. and Mao, G. (2013). Road traffic density estimation in vehicular networks. In 2013 IEEE Wireless Communications and Networking Conference (WCNC), pp.4653-4658. Shanghai, China.
- Matisziw, T.C. and Murray, A.T. (2009). Modelling s–t path availability to support disaster vulnerability assessment of network infrastructure. *Computers & Operations Research*, 36(1), pp.16-26.
- Mays, L.W. (2000). *Water distribution systems handbook*. McGraw-Hill, New York, USA.
- McCahill, C. and Garrick, N.W. (2008). The applicability of space syntax to bicycle facility planning. *Transportation research record*, 2074(1), pp.46-51.
- McColl, R.C., Ediger, D., Poovey, J., Campbell, D. and Bader, D.A. (2014). A performance evaluation of open source graph databases. In 2014 Proceedings of the 1<sup>st</sup> workshop on Parallel programming for analytics applications, pp. 11-18. Orlando, FL, USA.
- McDaniels, T., Chang, S., Peterson, K., Mikawoz, J. and Reed, D. (2007). Empirical framework for characterizing infrastructure failure interdependencies. *Journal of Infrastructure Systems*, 13(3), pp.175-184.
- McGill University, School of Planning. (2008). What is urban planning? Available at: <https://->

- McNally, R.K., Lee, S.W., Yavagal, D. and Xiang, W.N. (2007). Learning the critical infrastructure interdependencies through an ontology-based information system. *Environment and Planning B: Planning and Design*, 34(6), pp.1103-1124.
- Medhi, D. and Ramasamy, K. (2017). *Network routing: algorithms, protocols, and architectures*. Morgan Kaufmann, Elsevier, USA.
- Mensah, A.F. and Dueñas-Osorio, L. (2015). Efficient resilience assessment framework for electric power systems affected by hurricane events. *Journal of Structural Engineering*, 142(8), p.C4015013.
- Mesquita, B.B. and Smith, A. (2010). Leader survival, revolutions, and the nature of government finance. *American Journal of Political Science*, 54(4), pp.936-950.
- Metke, A.R. and Ekl, R.L. (2010). Security technology for smart grid networks. *IEEE Transactions on Smart Grid*, 1(1), pp.99-107.
- Min, H.S.J., Beyeler, W., Brown, T., Son, Y.J. and Jones, A.T. (2007). Toward modelling and simulation of critical national infrastructure interdependencies. *IIE Transactions*, 39(1), pp.57-71.
- Min, W. and Wynter, L. (2011). Real-time road traffic prediction with spatio-temporal correlations. *Transportation Research Part C: Emerging Technologies*, 19(4), pp.606-616.
- Mogridge, M.J. (1997). The self-defeating nature of urban road capacity policy: A review of theories, disputes and available evidence. *Transport Policy*, 4(1), pp.5-23.
- Mohit, M.A. and Elsawahli, H.M.H. (2017). Crime and Housing in Kuala Lumpur: Taman Melati terrace housing. *Asian Journal of Environment-Behaviour Studies*, 2(2), pp.53-63.
- Moss, T. and Marvin, S. (2016). *Urban infrastructure in transition: networks, buildings and plans*. Routledge, UK.
- Möderl, M., Butler, D. and Rauch, W. (2009). A stochastic approach for automatic generation of urban drainage systems. *Water Science and Technology*, 59(6), pp.1137-1143.
- Möderl, M., Sitzenfrei, R., Fetz, T., Fleischhacker, E. and Rauch, W. (2011). Systematic generation of virtual networks for water supply. *Water Resources Research*, 47(2), pp.1-10.
- Mpinda, S.A.T., Ferreira, L.C., Ribeiro, M.X. and Santos, M.T.P. (2015). Evaluation of graph databases performance through indexing techniques. *International Journal of Artificial Intelligence & Applications (IJAIA)*, 6, pp.87-98.
- Murray, A.T. and Grubestic, T. (2007). *Critical infrastructure: reliability and vulnerability*. Springer Science & Business Media.
- Murray-Tuite, P.M. (2006). A comparison of transportation network resilience under simulated system optimum and user equilibrium conditions. In *2006 Proceedings of the 38<sup>th</sup> Winter Simulation Conference*, pp.1398-1405. Monterey, CA, USA.



- Narayanaswami, S. (2017). Urban transportation: innovations in infrastructure planning and development. *The International Journal of Logistics Management*, 28(1), pp.150-171.
- National Research Council. (2014). *Advancing land change modelling: opportunities and research requirements*. National Academies Press, Washington, USA.
- National Security and Resilience Policy Division. (2009). Critical infrastructure protection and modelling analysis programme. Available at: <https://www.tisn.gov.au/Documents/CIPMA+-tasking+and+dissemination+protocols.pdf> [Accessed 12 December, 2018].
- Neo4j. (2018). <https://neo4j.com/>
- Neo4j. (2018). Temporal Indexing. <https://neo4j.com/docs/cyphermanual/current/syntax/temporal-/#cypher-temporal-specify-date> [Accessed 24 May 2019]
- Neo4j-Spatial. (2018). <https://neo4j-contrib.github.io/spatial/0.24-neo4j-3.1/index.html>
- NetworkX. (2014). NetworkX: Overview. Available at: <https://networkx.github.io/>. [Accessed 12 December, 2018]
- Newcastle City Council. (2011). Design and Construction of Roads and Accesses to Adoptable standards – Developer Guide. Available at: <https://www.newcastle.gov.uk/sites/default/files/-wwwfileroot/legacy/regen/plantrans/DesignAndConstructionOfRoadsAndAccessesToAdoptableStandardsMarch2011.pdf> [Accessed 12 December, 2018]
- Nguyen, T.T. (2009). Indexing PostGIS databases and spatial query performance evaluations. *International Journal of Geoinformatics*, 5(3), p.1.
- Nie, X., Zhu, S., Wang, H. and Huang, F. (2010). A new method of rural road network layout designing in the county: Generating algorithm of rectilinear steiner tree. In 2010 International Conference on Intelligent Computation Technology and Automation, 1, pp. 37-41.
- Nijkamp, P. and Kourtit, K. (2013). The “new urban Europe”: Global challenges and local responses in the urban century. *European Planning Studies*, 21(3), pp.291-315.
- Northern Powergrid. (2017). IMP/001/911 Code of Practice for the Economic Development of the LV System. Yorkshire, UK. Available at: <https://www.northernpowergrid.com/asset/0/-document/109.pdf> [Accessed 12 December, 2018]
- Nouvel, R., Mastrucci, A., Leopold, U., Baume, O., Coors, V. and Eicker, U. (2015). Combining GIS-based statistical and engineering urban heat consumption models: Towards a new framework for multi-scale policy support. *Energy and Buildings*, 107, pp.204-212.
- Obermayer, A., Guentert, F.W., Angermair, G., Tandler, R., Braunschmidt, S. and Milojevic, N. (2010). Different approaches for modelling of sewer caused urban flooding. *Water Science and Technology*, 62(9), pp.2175-2182.
- Ogie, R., Holderness, T., Dunbar, M. and Turpin, E. (2017). Spatio-topological network analysis of hydrological infrastructure as a decision support tool for flood mitigation in coastal mega-cities. *Environment and Planning B: Urban Analytics and City Science*, 44(4), pp.718-739.
- Ordnance Survey. (2018). OS MasterMap Point of Interest Layer. Information available from:

<https://www.ordnancesurvey.co.uk/business-and-government/products/points-of-interest.html>  
[Accessed 12 December, 2018]

Ordnance Survey. (2018). OS MasterMap Topography Layer. Information available from:  
<https://www.ordnancesurvey.co.uk/business-and-government/products/topography-layer.html>  
[Accessed 12 December, 2018]

Ordnance Survey. (2018). OS MasterMap ITN Layer. Information available from: <https://data.gov.uk/dataset/6459b2ce-87ba-4735-b0c0-a35282cd6311/os-mastermap-integrated-transport-network-layer> [Accessed 12 December, 2018]

O'Rourke, T, D. (2007). Critical Infrastructure, Interdependencies, and Resilience. Available at:  
<https://pdfs.semanticscholar.org/6c17/b35ec7555a9f27d5ccb6ca1d357a20b5ce0a.pdf> [Accessed 12 December, 2018]

Osiadacz, A. (1987). *Simulation and analysis of gas networks*. J.W. Arrowsmith Ltd, Bristol, UK.

Ostfeld, A., Oliker, N. and Salomons, E. (2013). Multiobjective optimization for least cost design and resiliency of water distribution systems. *Journal of Water Resources Planning and Management*, 140(12), p.04014037.

Ouyang, M., Dueñas-Osorio, L. and Min, X. (2012). A three-stage resilience analysis framework for urban infrastructure systems. *Structural safety*, 36, pp.23-31.

Ouyang, M. (2014). Review on modelling and simulation of interdependent critical infrastructure systems. *Reliability engineering & System safety*, 121, pp.43-60.

Parish, Y.I. and Müller, P. (2001). Procedural modelling of cities. In 2001 Proceedings of the 28<sup>th</sup> annual conference on Computer graphics and interactive techniques, pp.301-308. Los Angeles, CA, USA.

Pant, R., Hall, J.W. and Blainey, S.P. (2016). Vulnerability assessment framework for interdependent critical infrastructures: case-study for Great Britain's rail network. *European Journal of Transport and Infrastructure Research*, 16(1).

Pant, R., Hall, J.W., Barr, S. and Alderson, D. (2014). Spatial risk analysis of interdependent infrastructures subjected to extreme hazards. *Vulnerability, Uncertainty, and Risk: Quantification, Mitigation, and Management*, pp.677-686.

Papadias, D., Zhang, J., Mamoulis, N. and Tao, Y. (2003). Query processing in spatial network databases. In 2003 Proceedings of the 29<sup>th</sup> international conference on very large data bases, 29, pp. 802-813. Berlin, Germany.

Patterson, J.L. (2016). Traffic modelling in cities—validation of space syntax at an urban scale. *Indoor and built environment*, 25(7), pp.1163-1178.

Paton, N.W., Williams, M.H., Dietrich, K., Liew, O., Dinn, A. and Patrick, A. (2000). VESPA: A benchmark for vector spatial databases. In 2000 17<sup>th</sup> British National Conference on Databases, pp. 81-101. London, UK

Perera, C., Zaslavsky, A., Christen, P. and Georgakopoulos, D. (2014). Sensing as a service model for

smart cities supported by internet of things. *Transactions on Emerging Telecommunications Technologies*, 25(1), pp.81-93.

PgRouting. (2018). <http://docs.pgrouting.org/>

Piotrowski, K., Peralta, J.J., Jimenez-Redondo, N., Matusiak, B.E., Zieliński, J.S., Casaca, A., Ciemniowski, W., Krejtz, K. and Kowalski, J. (2014). How to balance the energy production and consumption in energy efficient smart neighbourhood. In *MedPower 2014 Conference*, Athens, Greece.

Pinto, H.S. and Martins, J.P. (2004). Ontologies: How can they be built? *Knowledge and information systems*, 6(4), pp.441-464.

Popovich, V.V. (2014). Intelligent GIS conceptualization. *Information Fusion and Geographic Information Systems*. Springer, Berlin, Heidelberg.

Porta, S., Crucitti, P. and Latora, V. (2008). Multiple centrality assessment in Parma: a network analysis of paths and open spaces. *urban design International*, 13(1), pp.41-50.

Pregolato, M., Robson, C., Smith, A., Peterson, J., Lomax, N., Barr, S. (2018). A building stock and composition model for the UK. In *26<sup>th</sup> GISRUUK 2018 conference*. Leicester, UK.

Preis, A., Whittle, A.J., Ostfeld, A. and Perelman, L. (2010). Efficient hydraulic state estimation technique using reduced models of urban water networks. *Journal of Water Resources Planning and Management*, 137(4), pp.343-351.

Prodanović, D., Stanić, M., Milivojević, V., Simić, Z. and Arsić, M. (2009). DEM-based GIS algorithms for automatic creation of hydrological models data. *J Serbian Soc Computation Mech*, 3(1), pp.64-85.

Psycopg (2018). <http://initd.org/psycopg/>

Puig, V., Ocampo-Martínez, C., Pérez, R., Cembrano, G., Quevedo, J. and Escobet, T. (2017). *Real-time Monitoring and Operational Control of Drinking-Water Systems*. Springer International Publishing.

Quigley, J. M. (2008). *Urban Economics. The New Palgrave Dictionary of Economics*. Palgrave Macmillan, UK.

Rahal, C.M., Sterling, M.J.H. and Coulbeck, B. (1980). Parameter tuning for simulation models of water distribution networks. *Proceedings of the Institution of Civil Engineers*, 69(3), pp.751-762.

Rahimi, A. (2016). A methodological approach to urban land-use change modelling using infill development pattern—a case study in Tabriz, Iran. *Ecological Processes*, 5(1), p.1-15.

Rautenbach, V., Coetzee, S. and Iwaniak, A. (2013). Orchestrating OGC web services to produce thematic maps in a spatial information infrastructure. *Computers, Environment and Urban Systems*, 37, pp.107-120.

Rinaldi, S.M., Peerenboom, J.P. and Kelly, T.K. (2001). Identifying, understanding, and analyzing critical infrastructure interdependencies. *IEEE Control Systems*, 21(6), pp.11-25.

- Robson, C. (2017). Robustness of hierarchical spatial critical infrastructure networks. PhD thesis. Newcastle University, UK.
- Robson, C., Barr, S., Prenolato, M., Ji, Q. (2018). A spatiotemporal database framework for infrastructure systems analytics and modelling. In 26<sup>th</sup> GISRUUK 2018, Leicester, UK.
- Rosato, V., Issacharoff, L., Tiriticco, F., Meloni, S., Porcellinis, S. and Setola, R. (2008). Modelling interdependent infrastructures using interacting dynamical models. *International Journal of Critical Infrastructures*, 4(1-2), pp.63-79.
- Rosato, V. (2015). Decision support system for critical infrastructure protection. ERNCIP Pre-Conference Special Session 2nd, Brussels, Belgium. Available at: <https://erncipproject.jrc.ec.europa.eu/sites/default/files/Decision%20Support%20System%20for%20Critical%20Infrastructure%20Protection%20-%20Vittorio%20Rosato.pdf> [Accessed 27 May, 2019]
- Rosen, R., Von Wichert, G., Lo, G. and Bettenhausen, K.D. (2015). About the importance of autonomy and digital twins for the future of manufacturing. *IFAC-PapersOnLine*, 48(3), pp.567-572.
- Royal Academy of Engineering. (2011). Infrastructure, Engineering and Climate Change Adaptation – Ensuring Services in an Uncertain Future. RAE, London, UK. Available at: <http://www.raeng.org.uk/publications/reports/engineering-the-future> [Accessed 12 December, 2018]
- Rui, Y.K. (2013). Urban Growth Modelling Based on Land-use Changes and Road Network Expansion. Doctoral Dissertation. Royal Institute of Technology Stockholm, Sweden. Available at: <http://www.diva-portal.org/smash/get/diva2:621238/FULLTEXT01.pdf> [Accessed 12 December, 2018]
- Sabeur, Z.A., Melas, P., Meacham, K., Corbally, R., D' Ayala, D. and Adey, B. (2017). An Integrated Decision-Support Information System on the Impact of Extreme Natural Hazards on Critical Infrastructure. In 12<sup>th</sup> Environmental Software Systems. Computer Science for Environmental Protection, ISESS 2017, pp. 302-314. Zadar, Croatia
- Samardžić - Petrović, M., Dragičević, S., Kovačević, M. and Bajat, B. (2016). Modelling urban land use changes using support vector machines. *Transactions in GIS*, 20(5), pp.718-734.
- Schintler, L.A., Kulkarni, R., Gorman, S. and Stough, R. (2007). Using raster-based GIS and graph theory to analyze complex networks. *Networks and Spatial Economics*, 7(4), pp.301-313.
- Schiller, G. (2007). Urban infrastructure: challenges for resource efficiency in the building stock. *Building Research & Information*, 35(4), pp.399-411.
- Schmidthaler, M. and Reichl, J. (2016). Assessing the socio-economic effects of power outages ad hoc. *Computer Science-Research and Development*, 31(3), pp.157-161.
- Schrank, D., Eisele, B. and Lomax, T. (2012). TTI's 2012 urban mobility report. Texas A&M Transportation Institute. Available at: <https://www.pagregion.com/Portals/0/documents/-HumanServices/2012MobilityReport.pdf> [Accessed 12 December, 2018]
- Shapely. (2018). The Shapely User Manual: Available at <http://shapely.readthedocs.io/en/stable/manual.html> [Accessed 12 December, 2018]

- Shelton, T., Zook, M. and Wiig, A. (2015). The ‘actually existing smart city’. *Cambridge Journal of Regions, Economy and Society*, 8(1), pp.13-25.
- Shepard, M. (2011). *Sentient city: Ubiquitous computing, architecture, and the future of urban space*. The MIT press, Cambridge, Massachusetts, USA.
- Short, J. and Kopp, A. (2005). Transport infrastructure: Investment and planning. Policy and research aspects. *Transport policy*, 12(4), pp.360-367.
- Sicilia, M.Á. and Santos, L. (2009). Main elements of a basic ontology of infrastructure interdependency for the assessment of incidents. *World Summit on Knowledge Society*, pp.533-542. Springer, Berlin, Heidelberg.
- Skyscraper Centre. (2019). Hong Kong Facts. <https://www.skyscrapercenter.com/city/hong-kong> [Accessed Nov 2019]
- Smith, B.L., Williams, B.M. and Oswald, R.K. (2002). Comparison of parametric and nonparametric models for traffic flow forecasting. *Transportation Research Part C: Emerging Technologies*, 10(4), pp.303-321.
- Sokolov, V. and Wenzel, F. (2013). Spatial correlation of ground motions in estimating seismic hazards to civil infrastructure. *Handbook of seismic risk analysis and management of civil infrastructure systems*. Woodhead Publishing. Sawston, Cambridge, UK.
- Soltani-Sobh, A., Heaslip, K., Scarlatos, P. and Kaiser, E. (2016). Reliability based pre-positioning of recovery centers for resilient transportation infrastructure. *International Journal of Disaster Risk Reduction*, 19, pp.324-333.
- St-Pierre, J., Wilkinsor, D.P., Knights, S. and Bos, M.L. (2000). Relationships between water management, contamination and lifetime degradation in PEFC. *Journal of New Materials for Electrochemical Systems*, 3(2), pp.99-106.
- Steele, W., and Legacy, C. (2017). Critical Urban Infrastructure. *Urban Policy and Research*. 35(1), pp.1-6.
- Su, K., Li, J. and Fu, H. (2011). Smart city and the applications. In 2011 International Conference on Electronics, Communications and Control (ICECC), pp. 1028-1031. Ningbo, China.
- Sun, Y. and Li, S. (2016). Real-time collaborative GIS: A technological review. *ISPRS Journal of Photogrammetry and remote sensing*, 115, pp.143-152.
- Swan, L.G. and Ugursal, V.I. (2009). Modelling of end-use energy consumption in the residential sector: A review of modelling techniques. *Renewable and sustainable energy reviews*, 13(8), pp.1819-1835.
- Tacoli, C., McGranahan, G. and Satterthwaite, D. (2015). Urbanisation, rural-urban migration and urban poverty. Human Settlements Group, International Institute for Environment and Development. Available at: <https://pubs.iied.org/pdfs/10725IIED.pdf> [Accessed 12 December, 2018]
- Tang, Y. (2016). Benchmarking graph databases with cyclone benchmark. Master Dissertation. Iowa

State University, USA. Available at: <https://lib.dr.iastate.edu/cgi/viewcontent.cgi?article=6827-&context=etd> [Accessed 12 December, 2018]

Tanyimboh, T. and Key, M. (2011). *Distribution network elements*. ICE Publishing, Exeter, UK.

Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H. and Sui, F. (2018). Digital twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology*, 94(9-12), pp.3563-3576.

Teoh, S.T. (2007). Autopolis: Allowing user influence in the automatic creation of realistic cities. In 2007 3<sup>rd</sup> International Symposium on Visual Computing, pp. 118-129. Lake Tahoe, NV, USA

Thant, P.T. and Naing, T.T. (2014). Hybrid Query Processing System (HQPS) for Heterogeneous Database (Relational and NoSQL). In 2014 Proceeding of the International Conference on Computer Networks and Information Technology, pp.53-58. Bali, Indonesia.

Thorne, C. (2014). Geographies of UK flooding in 2013/4. *The Geographical Journal*, 180(4), pp.297-309.

TPC-C Benchmark. (1992). <http://www.tpc.org/tpcc/>

Trifunović, N., Maharjan, B. and Vairavamoorthy, K. (2012). Spatial network generation tool for water distribution network design and performance analysis. *Water Science and Technology: Water Supply*, 13(1), pp.1-19.

Trust for London. (2019). London's geography and population. Available at: <https://www.trustforlondon.org.uk/data/londons-geography/> [Accessed Nov 2019]

Tutcher, J. (2016). Development of semantic data models to support data interoperability in the rail industry. Doctoral dissertation, University of Birmingham, UK. Available at: <https://etheses.bham.ac.uk/id/eprint/6774/> [Accessed 12 December, 2018]

Ueda, T., Tsutsumi, M., Muto, S. and Yamasaki, K. (2013). Unified computable urban economic model. *The annals of regional science*, 50(1), pp.341-362.

Urich, C., Sitzenfren, R., Möderl, M. and Rauch, W. (2010). An agent-based approach for generating virtual sewer systems. *Water Science and Technology*, 62(5), pp.1090-1097.

United Nations. (2013). Population, Development and the Environment 2013. Technical report, UN Population Division. Available at: [http://www.un.org/en/development/desa/population/publications/pdf/development/pde\\_wallchart\\_2013.pdf](http://www.un.org/en/development/desa/population/publications/pdf/development/pde_wallchart_2013.pdf) [Accessed 12 December, 2018]

United Nations. (2014). World Urbanization Prospects 2014: Highlights, United Nations Publications. Available at: <https://esa.un.org/unpd/wup/publications/files/wup2014-report.pdf> [Accessed 12 December, 2018]

Uschold, M., and Gruninger, M. (1996). Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*, 11(2), 93-155.

Uschold, M. and King, M. (1995). Towards a Methodology for Building Ontologies. In IJCAI'95

- Workshop on Basic Ontological Issues in Knowledge Sharing. Available at: <http://www.aiai.ed.ac.uk/publications/documents/1995/95-ont-ijcai95-ont-method.pdf> [Accessed 12 December, 2018]
- Varney, V. (2018). What we can learn from AI when planning urban spaces. Available at: <https://360.here.com/what-we-can-learn-from-ai-when-planning-urban-spaces> [Accessed 12 December, 2018]
- Vianello, C. and Maschio, G. (2014). Quantitative risk assessment of the Italian gas distribution network. *Journal of Loss Prevention in the Process Industries*, 32, pp.5-17.
- Vickridge, I. (2004). Aspects of sewer design. Sewers. *Replacement and new construction*. Butterworth-Heinemann, Oxford, UK.
- Volk, M., Hirschfeld, J., Dehnhardt, A., Schmidt, G., Bohn, C., Liersch, S. and Gassman, P.W. (2008). Integrated ecological-economic modelling of water pollution abatement management options in the Upper Ems River Basin. *Ecological Economics*, 66(1), pp.66-76.
- W3C. (2007). W3C Geospatial Ontologies. Available at: <https://www.w3.org/2005/Incubator/geo/-XGR-geo-ont-20071023/> [Accessed 5 July, 2017]
- Wallace, W.A., Mendonça, D., Lee, E., Mitchell, J. and Chow, J. (2001). Managing disruptions to critical interdependent infrastructures in the context of the 2001 World Trade Center attack. Available at: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.116.3591> [Accessed 12 December, 2018]
- Walski, Thomas M., Donald V. Chase, and Dragan Savic. (2001). *Water Distribution modelling*. Waterbury, CT, USA.
- Wang, C.H., Gong, H., George, B. and Freiwald, C. (2015). Spatial Network Database and Routing in Oracle Spatial. *Encyclopedia of GIS*, pp.1-11.
- Wang, H. (2013). A rule-based decision support system for critical infrastructure management. *Human and Ecological Risk Assessment: An International Journal*, 19(2), pp.566-576.
- Wang, J.W. and Rong, L.L. (2011). Robustness of the western United States power grid under edge attack strategies due to cascading failures. *Safety science*, 49(6), pp.807-812.
- Water World. (2003). Boil-Water Advisory. Available at: <https://www.waterworld.com/articles/2003/08/city-of-pontiac-stays-boil-water-advisory-for-one-more-day.html> [Accessed 5 July, 2017]
- Wei, Y.D. and Ye, X. (2014). Urbanization, urban land expansion and environmental change in China. *Stochastic environmental research and risk assessment*, 28(4), pp.757-765.
- Weng, Q. (2010). *Remote sensing and GIS integration: theories, methods, and applications*. New York: McGraw-Hill, USA.
- Whitbeck, J. and Conan, V. (2010). HYMAD: Hybrid DTN-MANET routing for dense and highly dynamic wireless networks. *Computer communications*, 33(13), pp.1483-1492.
- Wikipedia. (2018). Grey's Monument. Available at: [https://en.wikipedia.org/wiki/Grey%27s\\_Monument](https://en.wikipedia.org/wiki/Grey%27s_Monument) [Accessed 12 December, 2018]

- Wikipedia. (2019). Lucerne. Available at: <https://en.wikipedia.org/wiki/Lucerne#Topography> [Accessed 12 Nov, 2019]
- Wilkinson, S.M., Dunn, S. and Ma, S. (2012). The vulnerability of the European air traffic network to spatial hazards. *Natural hazards*, 60(3), pp.1027-1036.
- Woodhouse, J. (2014). Standards in asset management: PAS55 to ISO55000. *Infrastructure Asset Management*, 1(3), 57-59.
- World Bank. (2019). Hong Kong SAR, population density. <https://data.worldbank.org/country/hong-kong-sar-china> [accessed 13 Nov, 2019]
- Xia, Y.J., Kuang, L. and Li, X.M. (2011). Accelerating geospatial analysis on GPUs using CUDA. *Journal of Zhejiang University SCIENCE C*, 12(12), pp.990-999.
- Xu, X., Cai, H. and Chen, K. (2012). An Ontology Approach to Utility Knowledge Representation. In *Construction Research Congress 2018*, pp. 311-321.
- Yang, H. and H. Bell, M.G. (1998). Models and algorithms for road network design: a review and some new developments. *Transport Reviews*, 18(3), pp.257-278.
- Yazdani, A. and Jeffrey, P. (2011). Complex network analysis of water distribution systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 21(1), p.016111.
- Yin, C., Xiong, Z., Chen, H., Wang, J., Cooper, D. and David, B. (2015). A literature survey on smart cities. *Science China Information Sciences*, 58(10), pp.1-18.
- Yoon, B.H., Kim, S.K. and Kim, S.Y. (2017). Use of graph database for the integration of heterogeneous biological data. *Genomics & informatics*, 15(1), p.19.
- Yu, Z., Haghighat, F., Fung, B.C. and Yoshino, H. (2010). A decision tree method for building energy demand modeling. *Energy and Buildings*, 42(10), pp.1637-1646.
- Zhang, L. and He, X. (2012). Route Search Base on pgRouting. *Software Engineering and Knowledge Engineering: Theory and Practice*, pp.1003-1007. Springer, Berlin, Heidelberg.
- Zhang, Y., Liu, J., Qian, X., Qiu, A. and Zhang, F. (2017). An Automatic Road Network Construction Method Using Massive GPS Trajectory Data. *ISPRS International Journal of Geo-Information*, 6(12), p.400.
- Zhao, F., Wu, J., Sun, H., Gao, Z. and Liu, R. (2016). Population-driven urban road evolution dynamic model. *Networks and Spatial Economics*, 16(4), pp.997-1018.
- Zheng, J., Zhang, Z., Ciepluch, B., Winstanley, A.C., Mooney, P. and Jacob, R. (2013). A PostGIS-based pedestrian way finding module using OpenStreetMap data. In *2013 21<sup>st</sup> International Conference on Geoinformatics*, pp. 1-5. Henan, China.
- Zheng, X., Zhao, L., Fu, M. and Wang, S. (2008). Extension and application of space syntax a case study of urban traffic network optimizing in Beijing. In *2008 Workshop on Power Electronics and Intelligent Transportation System*, pp. 291-295. Washington, D.C, USA.
- Zhou, Y., Grygorash, O. and Hain, T.F. (2011). Clustering with minimum spanning trees.



International Journal on Artificial Intelligence Tools, 20(01), pp.139-177.

Zhu, X. and Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation. Available at: <http://mlg.eng.cam.ac.uk/zoubin/papers/CMU-CALD-02-107.pdf> [Accessed 12 December, 2018]

Zhu, Y. and Ferreira, J. (2015). Data integration to create large-scale spatially detailed synthetic populations. *Planning Support Systems and Smart Cities*, pp. 121-141. Springer, Berlin.

Zimmerman, R. (2001). Social implications of infrastructure network interactions. *Journal of Urban Technology*, 8(3), pp.97-119.

Zimmerman, R., Zhu, Q., De Leon, F. and Guo, Z. (2017). Conceptual modeling framework to integrate resilient and interdependent infrastructure in extreme weather. *Journal of Infrastructure Systems*, 23(4), p.04017034.

Zlatanova, S. (2000). 3D GIS for Urban Development. Doctoral Dissertation. ITC, The Netherlands. Available at: <http://zlatanova.xyz/PhDthesis/pdf/content.html> [Accessed 12 December, 2018]

Zygiaris, S. (2013). Smart city reference model: Assisting planners to conceptualize the building of smart city innovation ecosystems. *Journal of the Knowledge Economy*, 4(2), pp.217-231.

## Bibliography

The following publications have been produced from the Research presented in this thesis.

**Ji, Q.**, Barr, S., James, P., and Fairbairn, D., 2017. A heuristic spatial algorithm for generating fine-scale infrastructure distribution networks. In: 25<sup>th</sup> GISRUK 2017, Manchester, UK.

**Ji, Q.**, Barr, S., James, P., and Fairbairn, D., 2018. Graph database implementation of fine spatial scale urban infrastructure networks. In: 26<sup>th</sup> GISRUK 2018, Leicester, UK.

Robson, C., Barr, S., Pregolato, M., and **Ji, Q.**, 2018. A spatiotemporal database framework for infrastructure systems analytics and modelling. In: 26<sup>th</sup> GISRUK 2018, Leicester, UK.

**Ji, Q.**, Barr, S., James, P., and Fairbairn, D., 2018. A geospatial analysis framework for fine scale urban infrastructure networks. In: ISPRS Technical Commission IV Symposium 2018, Delft, the Netherlands.

Gilbert, T., Barr, S., James, P., Morley, J. and **Ji, Q.**, 2018. Software Systems Approach to Multi-Scale GIS-BIM Utility Infrastructure Network Integration and Resource Flow Simulation. ISPRS International Journal of Geo-Information, 7(8), p.310.

Fairbairn, D., and **Ji, Q.**, 2019. Using schematic mapping for synthetic networks. In: 2<sup>nd</sup> Schematic Mapping Workshop, Vienna, Austria.