

A geospatial analysis framework for fine scale urban infrastructure networks

Q. Ji^{1,*}, S. Barr¹, P. James¹, D. Fairbairn¹

¹School of Engineering, Newcastle University, Newcastle upon Tyne, UK – (q.ji2, stuart.barr, phil.james, david.fairbairn}@ncl.ac.uk

Commission IV, WG IV/7

KEY WORDS: Infrastructure Network, Fine Spatial Scale, Spatial Heuristics, Database System

ABSTRACT:

Understanding the spatial connectivity of urban infrastructure networks that connect assets to buildings is important for the fine-scale spatial analysis and modelling of the resource flows within cities. However, rarely are spatially explicit representations of infrastructure networks available for such analysis. Further, an appropriate database system is the core of development of an infrastructure assets information and management platform, capable of handling the wide range of data for infrastructure system modelling and analysis. In this paper, we develop a geospatial simulation and analysis framework, which is capable of generating fine-scale urban infrastructure networks and storing the network instances in a hybrid database system for further modelling and analysis needs. We demonstrate the use of this platform by simulating the entire-city electricity distribution network for the city of Newcastle upon Tyne. Validation of the resulting network is performed using the network layout diagram from the local power company. The heuristically derived network was found to have a 91% spatial accuracy.

1. INTRODUCTION

Modern cities consist of spatially and temporally complex networks which connect infrastructure assets to the buildings they service (Moss and Marvin, 2016). Critical infrastructure networks include transport, electricity, water supply, waste water and gas, all of which play a key role in the functionality of the modern cities (Murray and Grubestic, 2007). Having access to good quality data on these infrastructure networks is crucial, with their spatial connectivity being the most important information. At the intra-city scale, such information allows for an understanding of how each individual building is connected and served by infrastructure assets, and allow network vulnerability, demand, capacity and interdependency to be modelled and analysed at a fine spatial scale. However, it is rare for information on spatial layout and configuration of fine scale infrastructure networks to be available (Fu et al, 2005). Therefore, there is urgent need for approaches that can generate, at very fine spatial scales, plausible infrastructure networks connecting assets to the buildings they service.

Once good quality data (spatial connectivity) on infrastructure networks is available, it is also imperative to store and manage it properly. In many countries, individual operators in specific infrastructure sectors have realised the importance of developing their data and information management platform for better infrastructure planning and decision support (Woodhouse, 2014). Additionally, there are also several large research initiatives, such as the US National Research Council report on Sustainable Critical Infrastructure Systems, the Dutch programmes on Next Generation Infrastructure and Knowledge for Climate, the Australian Critical Infrastructure Protection Modelling and Analysis (CIPMA) Programme and the UK Infrastructure Transitions Research Consortium (ITRC), which are developing new infrastructure modelling and analysis tools for understanding critical infrastructure (Barr et al, 2016).

Generally, infrastructure asset management systems require appropriate database management systems that can handle the wide range of disparate data and relationships required for infrastructure systems modelling and analysis (Barr et al, 2016). Traditionally the database solutions rely on spatial relational databases, such as PostGIS (Barr et al, 2013) or Oracle Spatial (Fikjez and Rezanina, 2016), which are naturally strong in relational and spatial queries, but have performance issues in network queries, when modelling large network instances (Ji et al, 2018). Therefore, an appropriate database architecture should be carefully designed to efficiently model and analyse urban infrastructure networks in a finer spatial scale.

In this paper, we develop a geospatial analysis framework of fine scale urban infrastructure networks. It is an open-source framework consisting of two major parts, the data generation package and the data modelling package. The data generation package is built on a generic heuristic spatial algorithm that is responsible for generating plausible infrastructure distribution networks at a fine scale. The data modelling package is based on a hybrid database systems developed using PostGIS and Neo4j. The framework is also equipped with different APIs for network data I/O, and general network analysis tasks. We demonstrate the usage of our framework to generate and model the electricity distribution network for the entire city of Newcastle upon Tyne, UK.

2. FRAMEWORK DETAILS

The overall architecture of the framework is shown in figure 1. It is an open source framework built on Python scripts, PostGIS and Neo4j database softwares and several data APIs for input and output work. Its two sub components, the data generation package and data modelling package are introduced into details below.

* Corresponding author

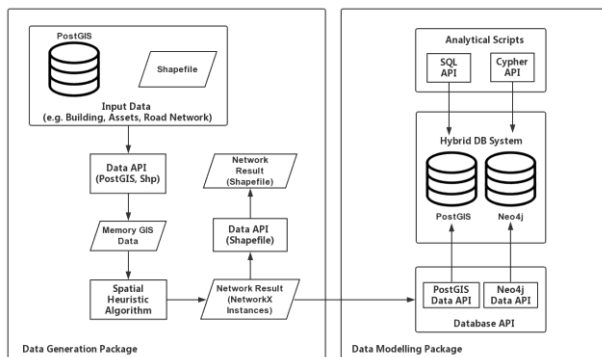


Figure 1. Overall architecture of the infrastructure network analysis framework.

2.1 Data Generation Package

The data generation package is developed based on NetworkX library (NetworkX, 2014) and spatial heuristic algorithm (Ji et al., 2017). It reads necessary input data including spatial location of infrastructure assets, buildings, and the road network. Spatial heuristics is then employed to generate infrastructure networks predominately follow the road network layout and whose total length is kept as short as possible (Larkevi and Holmes, 1997). The detailed algorithm implementation is shown in figure 2.

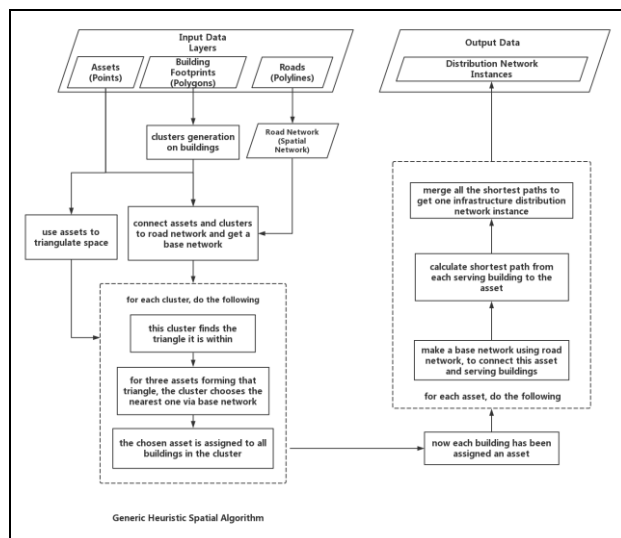


Figure 2. Implementation of generic heuristic spatial algorithm.

An example area containing necessary input data set (figure 3) is used to illustrate the steps of the algorithm. The entire algorithm can be divided into main sequential steps: the topology generation step and the geometry generation step. The topology generation step assigns each building a servicing asset. The geometry generation step generates the geometry layout for each asset and its servicing buildings.

The topology generation step is conducted as follows. First, geospatially buildings are grouped into different clusters distance; the clustering result is shown in figure 4 (A) where each cluster is depicted in one colour. Then the road network is expanded into a “base network” to connect all the clusters (represented by centroids) and asset points (figure 4 (B)). Thereafter, the asset points are used to triangulate the entire space (figure 5 (A)), and for each cluster, from the three asset points forming the triangle containing that cluster, it will choose

the nearest one asset via the base network. Finally, the chosen asset will be assigned to all the buildings which belongs to that cluster. The result for topology generation step is shown in figure (5 (B)), where buildings in the same group are served by the same asset.

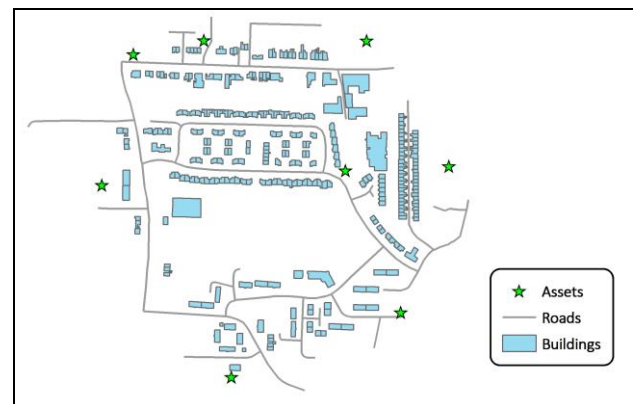


Figure 3. Input data set (Contains OS data © Crown copyright and database right 2018).

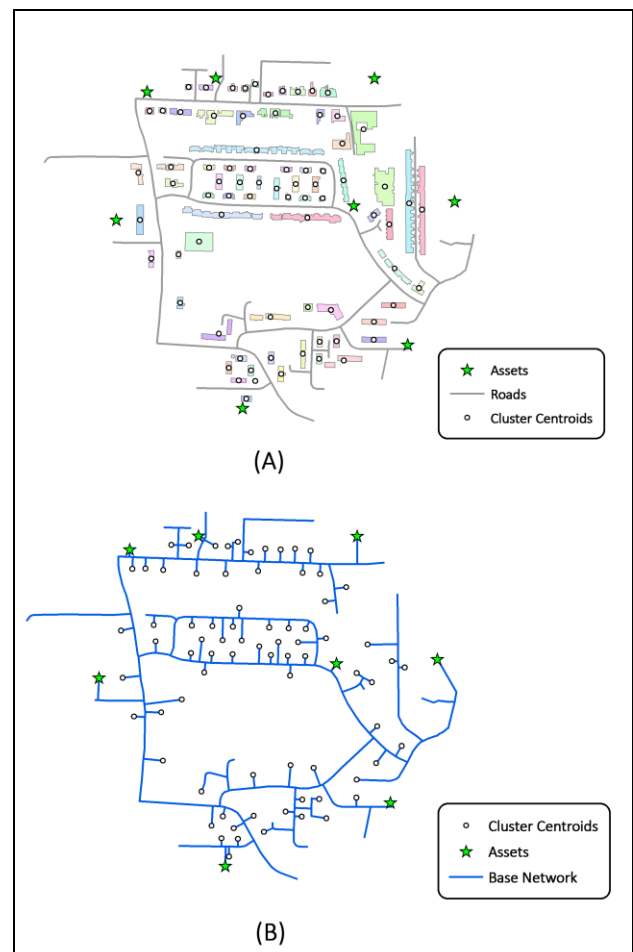


Figure 4. (A) Cluster generation on buildings. (B) Base network generation to connect assets and clusters (Contains OS data © Crown copyright and database right 2018).

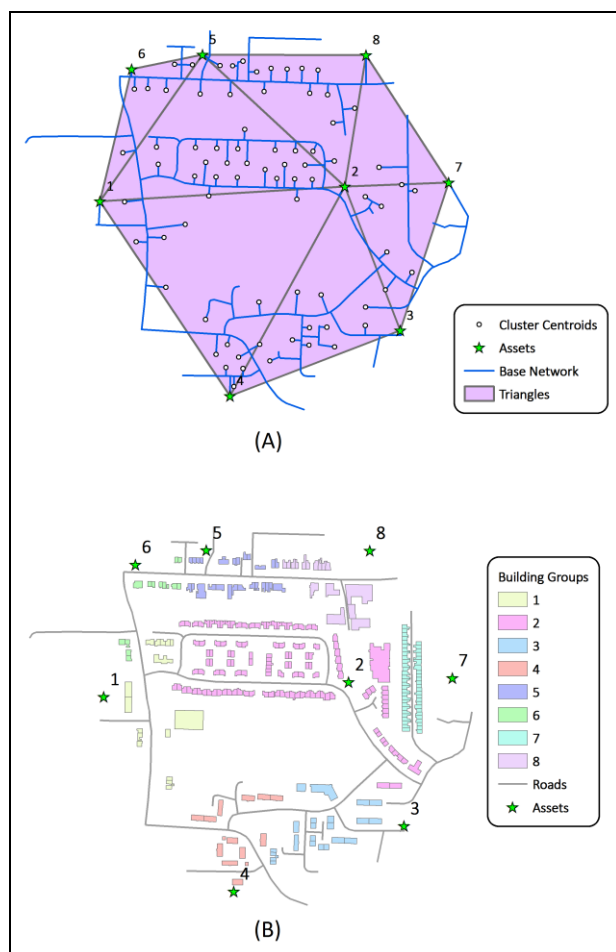


Figure 5. (A) Triangulation using asset points. (B) Result for topology generation step (Contains OS data © Crown copyright and database right 2018).

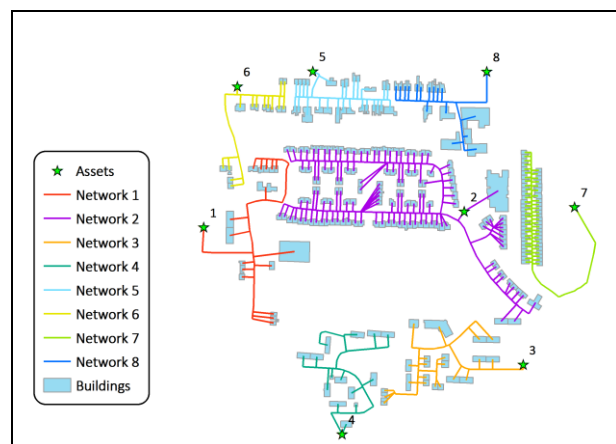


Figure 6. Generated infrastructure network data (Contains OS data © Crown copyright and database right 2018).

Within an infrastructure network, the resultant types of network nodes and edges are shown in figure 7: a node can be an “asset node”, “building node”, “assetAccess node”, “buildingAccess node”, or “distribution node”; an edge can be “assetAccess edge”, “buildingAccess edge”, or “distribution edge”. Figure 7 shows the resulting spatial network for sub-network 1 in figure 6.

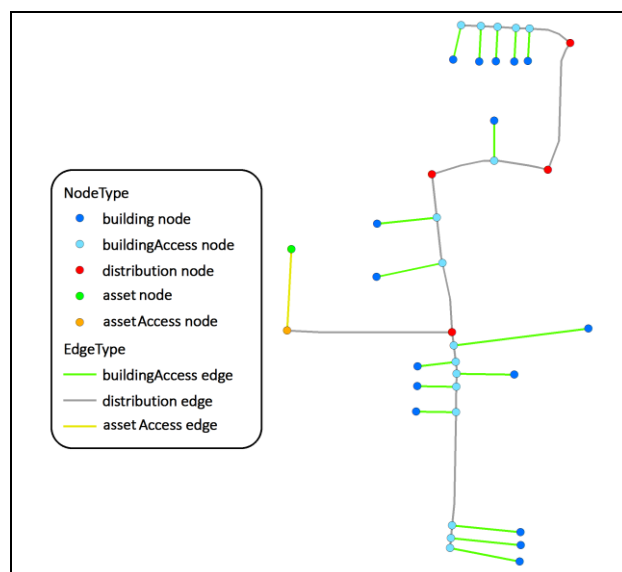


Figure 7. Different types of nodes and edges in an infrastructure network (Contains OS data © Crown copyright and database right 2018).

The “buildingAccess” and “assetAccess” nodes refer to the junction nodes, that connect “building” and “asset” nodes to the main network via “buildingAccess” and “assetAccess” edges.

2.2 Data Modelling Package

The core of the data modelling package is a hybrid database system, which contains a PostGIS and a Neo4j database. The infrastructure networks are spatial networks, where geometry, topology and attributes are required to be stored, retrieved and analysed efficiently. The PostGIS database is a natural solution for spatial data storage (namely, the geometry of the network edge and node). The Neo4j Graph database, on the other hand is a native and efficient solution for storing and querying network topology and attributes. Neo4j has its own data model “property graph” which consists of nodes and relationships connecting these nodes. Property, which is a collection of (key:value) pairs can exist on either a node or a relationship. Neo4j also has its own query language Cypher, which is naturally designed for network related query tasks.

Although network topology can be stored in relational spatial databases using a carefully designed database schema (Barr et al., 2013), querying the network instance can meet performance bottle-necks when the network is large and has a complex topology (Ji et al., 2018). On the other hand, Neo4j’s support on spatial data is limited. Currently, geometry data can only be added on Neo4j nodes (but not relationships), and its supported spatial query is simple (fetching all the nodes within a Euclidean distance to a given point). Due to this, it is still preferred to store all the geometry (of nodes and edges) in PostGIS, where more complex spatial query is supported. Below is the general implementation of the hybrid database system using PostGIS and Neo4j.

In PostGIS, only two tables are used: **nodes** and **edges** table (no matter how many network instances there are). These two tables are used to store geometry of all the network nodes and edges, respectively.

The columns for the **nodes** table are **node_id**, **net_id**, and **geom**. The **net_id** is the id to indicate which network instance it belongs to. The **node_id** is the id for a node within a network instance. Finally, the **geom** is the actually geometry of the node (a point). In other words, (**node_id**, **net_id**) is a primary key in the **nodes** table. Similarly, the columns for the **edges** table are **edge_id**, **net_id** and **geom**. Again, the **net_id** indicates which network instance it is, and the **edge_id** is used to distinguish edges within a network instance. The **geom** is the actually geometry of the edge (a linestring). Likewise, (**edge_id**, **net_id**) is a primary key in the **edges** table.

In Neo4j, infrastructure network nodes and edges are stored as nodes and relationships of property graph, and by doing this, network topology is stored. All the attributes associated with infrastructure network nodes and edges are stored as properties of the corresponding nodes and relationships in Neo4j. Moreover, to link PostGIS and Neo4j, the attributes **net_id**, **node_id** or **edge_id** are also stored in Neo4j.

For example, in order to store one electricity distribution network, its topology and attributes are stored Neo4j as a property graph, shown in figure 8. The geometries of this electricity distribution network are stored into PostGIS using the **nodes** and **edges** table, shown in figure 9. Moreover, the highlighted record in the **nodes** table corresponds to the node in red rectangle in Neo4j property graph.

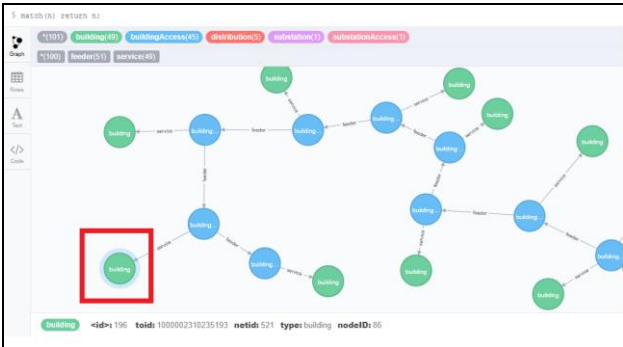


Figure 8. An electricity distribution network, used as an example to show how infrastructure networks are stored as a property graph in Neo4j.

node_id	net_id	geom
integer	integer	geometry(Point,27700)
70	69	S21 0101000020346C0000083C0A21E9E
71	70	S21 0101000020346C00000E17A14AE1AC
72	71	S21 0101000020346C00000D974E92EE
73	72	S21 0101000020346C00000E17A14AE1AC
74	73	S21 0101000020346C0000000000000000
75	74	S21 0101000020346C000008E9F1A2F2E
76	75	S21 0101000020346C000008E1E5A89E
77	76	S21 0101000020346C00000EC918B1EC
78	77	S21 0101000020346C000006B274313C
79	78	S21 0101000020346C000008E1E5A89E
80	79	S21 0101000020346C000008E1E5A89E
81	80	S21 0101000020346C00000B0726B1F6E
82	81	S21 0101000020346C0000025068115AC
83	82	S21 0101000020346C000004E77A74F4E
84	83	S21 0101000020346C00000EC51B89EC
85	84	S21 0101000020346C000002F0246C
86	85	S21 0101000020346C000004B6F3FDF0E
87	86	S21 0101000020346C0000000000000000
88	87	S21 0101000020346C000004E77A74F4E
89	88	S21 0101000020346C000008E1E5A89E

edge_id	net_id	geom
integer	integer	geometry(LineString,27700)
1	0	S21 0102000020346C0000020000000000
2	1	S21 0102000020346C0000020000000000
3	2	S21 0102000020346C0000020000000000
4	3	S21 0102000020346C0000020000000000
5	4	S21 0102000020346C0000020000000000
6	5	S21 0102000020346C0000020000000000
7	6	S21 0102000020346C0000020000000000
8	7	S21 0102000020346C0000020000000000
9	8	S21 0102000020346C0000020000000000
10	9	S21 0102000020346C0000020000000000
11	10	S21 0102000020346C0000020000000000
12	11	S21 0102000020346C0000020000000000
13	12	S21 0102000020346C0000020000000000
14	13	S21 0102000020346C0000020000000000
15	14	S21 0102000020346C0000020000000000
16	15	S21 0102000020346C0000020000000000
17	16	S21 0102000020346C0000020000000000
18	17	S21 0102000020346C0000020000000000
19	18	S21 0102000020346C0000020000000000
20	19	S21 0102000020346C0000020000000000

Table nodes

Table edges

Figure 9. The table **nodes** and **edges** in PostGIS used to store node and edge geometries for the network shown in figure 8. Note the highlighted record in the **nodes** table corresponds to the Neo4j node in red rectangle in figure 8 (**node_id** is 86).

Using this approach, it is possible to link data in PostGIS and Neo4j, which allows complex queries on network topology, attributes and geometry at the same time. For example, besides the database APIs, the hybrid database system is also allowed to be visited by analytical scripts that are built based on SQL APIs and Cypher APIs. A geometry- or spatial-related script is normally executed using SQL to query the spatial data stored in PostGIS. A network topology- or attributes-related query is normally executed using Cypher to efficiently query large and complex network instances stored in Neo4j.

3. PILOT STUDY

To demonstrate the utility and scalability of our framework in a full scale city area, it was applied to generate the local electricity distribution networks for Newcastle upon Tyne. The input infrastructure assets comprised all 11kv electricity substations, identified from Ordnance Survey Point of Interest Layer. Building footprints were obtained by filtering the Ordnance Survey MasterMap Topography Layer. The road network was obtained using the Ordnance Survey ITN Layer (centerlines of roads). In total, 657 distribution networks were generated serving 105,583 buildings, with each network serving 160 buildings on average. The total number of edges and nodes generated are 190,989 and 191,595, respectively. Figure 10 shows the synthetic distribution networks generated for the entire Newcastle upon Tyne city area, separately coloured for each single distribution network.

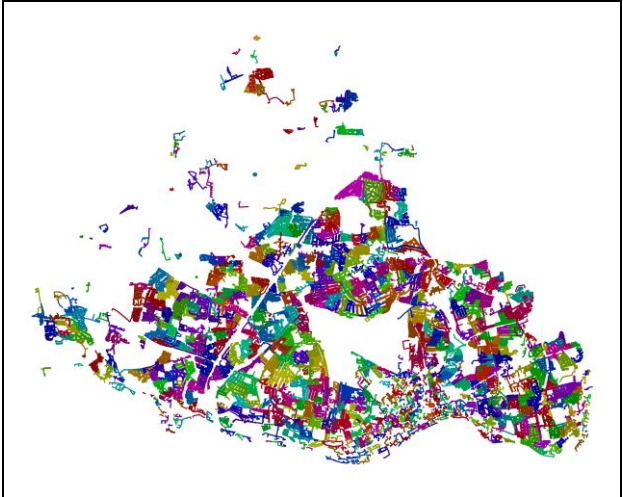


Figure 10. Synthetic electricity distribution networks generated for Newcastle upon Tyne. Each colour refers to a network instance (Contains OS data © Crown copyright and database right 2018).

When generating synthetic networks heuristically, the biggest concern is the data quality, or how well our synthetic networks represent the real networks. With the assistance of Northern Power Grid (NPG) the utility company who supply electricity to Newcastle upon Tyne a validation of the heuristically derived network was undertaken.

The NPG diagram in shown in figure 11. NPG labels cables to be either “service line” or “feeder”. The “service line” refers to the cable directly connected with a building, and corresponds to the “buildingAccess” edge in the synthetic network (Figure 7); and the “feeder” refers to any other cables and corresponds to all the other types of edge in the synthetic network. To avoid confusion during validation, “synthetic feeder” and “synthetic

service line” are used to term the edges in our synthetic network model, which correspond to the feeders and service lines in the NPG diagram. Validation will be done on the feeders and service lines separately.

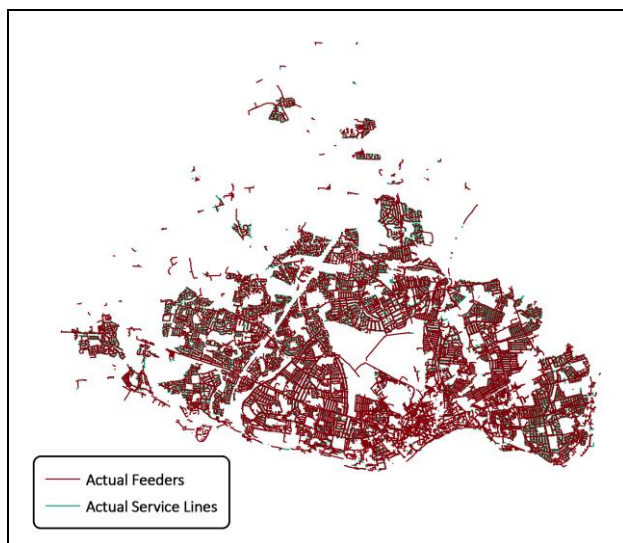


Figure 11. Distribution networks diagram from Northern Power Grid.

To validate the feeders, percentage of presence is used to measure how close the layout of synthetic network matches the actual ones. Generally, for each actual feeder, there should be a synthetic feeder nearby so that it is generated necessarily. Likewise, for each synthetic feeder, there should be an actual feeder nearby. Given a Euclidean distance, it is possible to calculate this two percentages. If both percentages are high, it means layouts of these two networks match well with each other. It is found that **72.7%** of the synthetic feeders are within 5 meters buffer of the actual feeders. On the other hand, **74.2%** of the actual feeders are within 5 meters buffer of the synthetic feeders. These two percentages are medium and not too high, and the main cause is explained in figure 12.

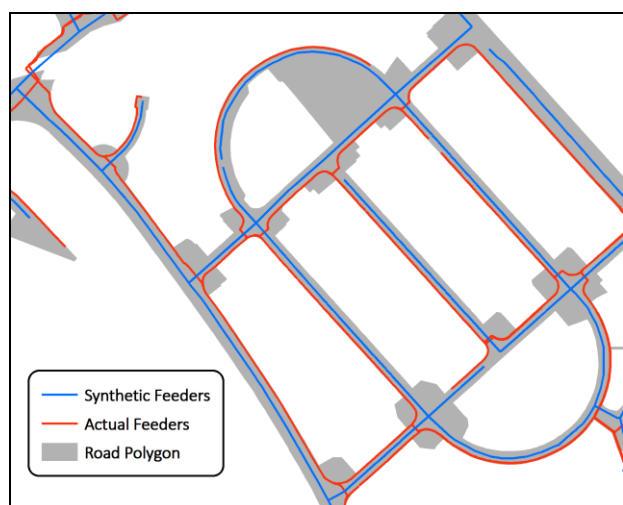


Figure 12. Use of road polygon layer for better validation on the feeders (Contains OS data © Crown copyright and database right 2018).

The synthetic feeders generated by heuristic algorithm are based on ITN road network, which are actual road centrelines. Since the synthetic feeders always follow the ITN road network, they

will just follow the road centrelines. However, in reality roads are represented as polygons. If the actual feeder follows one side of the road (rather than the centreline), it is still *following* the road network, but there can be considerable mismatch between the layout of the synthetic feeders and actual ones (depending on the road width). To address this issue, another layer from the MasterMap, the road polygon layer, is used here to represent the real space occupied by the road (Figure 12).

This road polygon layer is considered to demonstrate equivalence to the layout of synthetic feeders (which are actually fixed to centrelines) and will thus be used when validating by the actual feeders. The “spatial accuracy” is defined as the percentage (of the total length) of the actual feeders which fall within the road polygon layer. This spatial accuracy is **91%** in the entire city, which means **91%** of the actual feeders fall within the road polygons. The high percentage here indicates that the actual feeders do follow the road network, which proves the basic assumption of the algorithm.

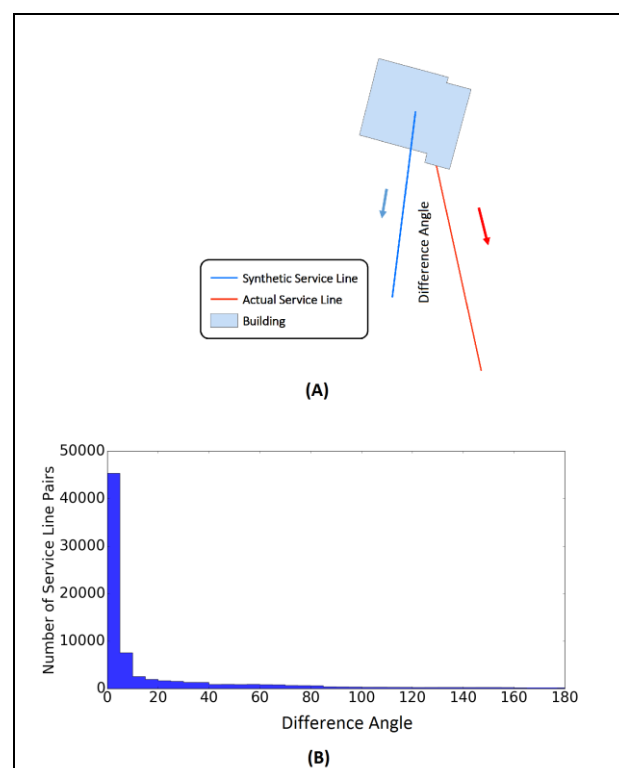


Figure 13. Validating service lines using difference angle. (A) Definition of difference angle. (B) Difference angles across the entire Newcastle.

On the other hand, to validate the service lines, the road polygon layer is unnecessary, since a service line is directly connected to a building, and thus will not necessarily follow the road network. Instead the validation approach is called “difference angle”, which is defined in Figure 13 (A) to show the angle in a service line pair (actual service line, synthetic service line), 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° and 180° . For each building in the city, the difference angle is calculated (where data on actual service line exist), and a histogram is generated (figure 13 (B)). In total, 75548 service lines pairs in Newcastle upon Tyne were found and used for

validation. The difference angles of over 70% service line pairs (52902 pairs) are less than 10°, and the average value in the entire city is 19°. Based on this value, the directions of the synthetic service lines match well with those of the actual ones.

4. CONCLUSION

In this paper, we develop a geospatial analytics framework for finer scale urban infrastructure networks. The entire framework consists of two major components, the data generation package and the data modelling package. The data generation package is designed for generating plausible spatial layout of various infrastructure networks, often lacking in the fine spatial scale for many utilities and services. Then the generated network data can be stored using our data modelling package, which is built based on a federated database system to store the geometry, attributes and topology of the network instances. Additional APIs are provided to retrieve and query network data. We also demonstrate the application of this framework to generate entire-city level electricity distribution networks in Newcastle upon Tyne, and store it in the database schema for further analysis needs. Further work can be done to integrate different infrastructure networks (e.g. water supply, waste water and gas, etc) for Newcastle upon Tyne in our framework to support network interdependency modelling and analysis.

ACKNOWLEDGEMENTS

The authors thank OS (Ordnance Survey) for providing the ITN and MasterMap data.

REFERENCES

- Barr, S.L., 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', In: *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.
- 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*, Springer, Cham, p.935-945.
- Fu, G., Jones, C.B. and Abdelmoty, A.I., 2005. Ontology-Based Spatial Query Expansion in Information Retrieval, In: *ODBASE, OTM Confederated International Conferences*. Agia Napa, Cyprus.
- Gilbert, T., Barr, S., James, P., Morley, J., 2018. A prototype system for integrating geospatial and BIM electricity network models and simulating the flow of electrical power. In: *26th GISRUUK 2018*, Leicester, UK.
- Ji, Q., Barr, S., James, P., and Fairbairn, D., 2017. A heuristic spatial algorithm for generating fine-scale infrastructure distribution networks. In: *25th GISRUUK 2017*, Manchester, UK.
- Ji, Q., Barr, S., James, P., and Fairbairn, D., 2018. Graph database implementation of fine spatial scale urban infrastructure networks. In: *26th GISRUUK 2018*, Leicester, UK.
- Lakervi, E. and Holmes, E.J. (1995). *Electricity distribution network design*, Peter Peregrines Ltd, London, UK.
- Moss, T. and Marvin, S., 2016. *Urban infrastructure in transition: networks, buildings and plans*, Routledge, London, UK.
- Murray, A.T. and Grubestic, T.H., 2007. Overview of reliability and vulnerability in critical infrastructure. *Critical Infrastructure: reliability and vulnerability*, Springer, Berlin, Germany.
- NetworkX., 2014. <https://networkx.github.io/>
- Woodhouse, J. (2014). Standards in asset management: PAS55 to ISO55000. *Infrastructure Asset Management*, 1(3), p.57-59.