Received: 4 January 2023 | Revised: 15 March 2023 | Accepted: 4 May 2023

DOI: 10.1111/tgis.13071

Transactions in GIS WILEY

RESEARCH ARTICLE

Framework for constructing multimodal transport networks and routing using a graph database: A case study in London

Seula Park[1,2] | Tao Cheng[2]

[1]Department of Civil and Environmental Engineering, Seoul National University, Seoul, Republic of Korea

[2]SpaceTimeLab, Department of Civil, Environmental & Geomatic Engineering, University College London, London, UK

Correspondence
Tao Cheng, SpaceTimeLab, Department of Civil, Environmental & Geomatic Engineering, University College London, London WC1E 6BT, UK.
Email: tao.cheng@ucl.ac.uk

Funding information
National Research Foundation of Korea, Grant/Award Number: NRF-2021R1A6A3A01086427; UK Economic and Social Research Council, Grant/Award Number: ES/L011840/1

Abstract

Most prior multimodal transport networks have been organized as relational databases with multilayer structures to support transport management and routing; however, database expandability and update efficiency in new networks and timetables are low due to the strict database schemas. This study aimed to develop multimodal transport networks using a graph database that can accommodate efficient updates and extensions, high relation-based query performance, and flexible integration in multimodal routing. As a case study, a database was constructed for London transport networks, and routing tests were performed under various conditions. The constructed multimodal graph database showed stable performance in processing iterative queries, and efficient multi-stop routing was particularly enhanced. By applying the proposed framework, databases for multimodal routing can be readily constructed for other regions, while enabling responses to diversified routings, such as personalized routing through integration with various unstructured information, due to the flexible schema of the graph database.

1 | INTRODUCTION

As global transport infrastructure develops, people travel freely via various means, with the majority using public transport combined with walking, cycling, or other modes (Idri et al., 2017). Since the complexity of personal travel is increasing, how to effectively represent the transport system is an emerging issue. In particular, the importance of support planning that utilizes various possible modes while reflecting personalized user needs is increasing (Hrncir & Jakob, 2013). Conventionally, multimodal transport systems have been represented as node–edge network models with weights, where nodes represent intersections, and edges depict road segments between two adjacent intersections (Wang et al., 2019).

Storing network models using a database system remains a prerequisite for the utilization of various attributes required for efficient routing and data management. Specifically, as various database-based systems for transport routing have been developed (Miler et al., 2014), the demand for databases has increased. Accordingly, transport network data has been stored and utilized with a relational database (Miler et al., 2014). Unfortunately, most navigation services do not disclose their database technology and data model; nonetheless, GIS data with attributes for applications provided by governments or commercial operations are generally managed through a relational database (Ali, 2020).

User habits, such as route selection style, and friendly environments are critical factors that can be used to resolve personalized multimodal problems (Faye et al., 2017). To this end, as interest in utilizing not only geographic information, but routing contexts (e.g., personalized preferences), place semantics, and time dependency emerges, thus increasing the demand for non-spatial and irregular data sources. For example, information on the temporary restriction of specific roads can be extracted from several documents (e.g., articles or posts), while semantic contexts of places or paths can be obtained from reviews. Moreover, transport networks used in multimodal routing must describe travel through various modes and rapidly reflect spatiotemporal changes; as, for example, public transport schedules may change at any time, and restrictions of roads or public transport can occur frequently occur due to construction or strikes. Also, traffic modes can be added according to infrastructural development or line expansions. Therefore, transport networks for routing require efficient update and expansion.

When using relational databases, however, complex data processing is required according to predefined schema to integrate differently structured data. Some critical information, such as travel motives and personal data, remain entirely unstructured (Zhuravleva & Poliak, 2020), thus creating difficulties with integration into existing relational database models, so as for the updating and expanding of the network and timetables. Also, a relational database is characterized by inefficient data management, such as adding new items, due to its strict schema (Batra & Tyagi, 2012; Medhi & Baruah, 2017).

In this context, previous networks with fixed schema are inefficient; with representative non-relational graph databases with flexible schema as potential alternatives. Graph databases employ graphs to model or generalize schema and instances (Angles & Gutierrez, 2008), and are optimized for storing, querying, and updating data with graph structure (Robinson et al., 2015). Such databases have been introduced to address limitations of relational databases (Jaiswal & Agrawal, 2013). Intuitive data models enable the effective storage of data describing complex real-world transportation systems; whereas the flexible data schema allows for the model to be readily extended to accommodate updates of transport networks and timetables. Moreover, graph databases can search and load only relevant data according to its built-in index structure (Liu et al., 2021); thus, the query performance is excellent for large datasets (De Virgilio et al., 2014). Furthermore, due to its strong scalability, graph traverse can be executed across all data sizes with consistent performance (Angles & Gutierrez, 2018). Accordingly, the high performance of graph databases can be expected in routing, a particularly spatial relation-based task, using large datasets.

Although the need for implementing graph databases in multimodal routing is increasing, the construction of such comprehensive databases remains rare. Diversified multimodal routing can be realized by integrating the semantic context of travel environments with multimodal transport networks using appropriate database systems. However, constructing a graph database that well-describes multimodal systems should precede integration of route information with semantic contexts. Accordingly, to effectively utilize a graph database for multimodal routing, the appropriate data model must complement existing network designs, and describe various modes of transport. Since integrating the road network with public transit networks is challenging (Berger et al., 2009), an optimal mode-coupling approach must be developed.

In previous multilayer networks for multimodal routing, layers were built by each mode independently, and interconnections between layers were created by designating identical stations (stops, places, etc.) as common nodes; however, the sole use of common nodes for interconnections cannot provide a complete transfer description between modes (Orozco et al., 2021). Furthermore, connecting nodes are unnecessarily duplicated since they must persist across all layers. Therefore, it is essential that networks are efficiently configured while fully describing the multimodal transport system. Furthermore, the data model should be applied to construct database for any regions in terms of expandability by providing a detailed structure as a standard; however, several database design issues must be addressed, including how to interconnect layers, label entities considering query performance, and define layer separation according to data update efficiency.

Consequently, it is necessary to design a comprehensive data model that considers these issues, and provides a framework for efficiently constructing a database based on the model. To this end, this study aimed to develop a framework for creating a multimodal transport model using a graph database to efficiently resolve multimodal problems. A conceptual data model design is proposed to be adapted for any area, and expanded to any other modes. Also, an approach for efficiently building the graph database from prebuilt spatial transport network data is proposed, and a graph database for London transport is built as a case study to verify the proposed methodology. An evaluation framework for the utility of graph database on multimodal routing problems is defined, and two tests are performed: graph constructing/projecting, and multimodal routing with several conditions.

2 | LITERATURE REVIEW

Previous multimodal transport networks have been modeled with weighted nodes and edges in a multilayer structure. For example, Ma and Lebacque (2013) provided a multimodal transport network with a multilayer structure comprised of a combination of unimodal subnetworks of bus and metro. However, it was tested with a simple network unable to fully reflect the complexity of true transport structures. Talasila et al. (2018) presented a three-layer network with train stations, stops, and changes based on timetable information and configured with weights. The model represented physical links between adjacent stations, links between consecutive train stops, and train availability. Although they combined temporal availability based on timetable with the spatial relationships of train stations, their model was unimodal. Previously, Jamal et al. (2017) modeled a multilayer temporal network of public transport and carpooling services, and timetable information was used to represent the temporal network; however, a prohibitively large graph with duplicated nodes was created using the time-expanded model, and the entire graph must be modified for schedule changes in this approach.

Relational databases have been employed for the continuous utilization and management of transport network data. For example, Chondrogiannis et al. (2016) presented a road and bus network designed with a timetable-based time-dependent model. Furthermore, they built a transport database using a spatial-relational database, PostGIS; however, it is necessary to expand the database to include multimodal systems. Similarly, Hrncir and Jakob (2013) used PostGIS to store network data composed of a time-dependent graph for public transit, and a network graph for individual transport. Although their model could integrate both public and private modes, the transfer route among modes within each graph should be described in detail. Notably, both Chondrogiannis et al. (2016) and Hrncir and Jakob (2013) considered the temporal context of public transit; however, such contexts within the transport system are highly volatile. Therefore, a new approach is required for immediate response to schedule changes, and efficient database updating. Both Gil's (2014) and Smarzaro et al.'s (2021) networks were managed using a relational database with strict schema. Especially, Smarzaro et al.'s (2021) framework included a complex schema matching process for integrating independently built datasets. A schema-less database that is advantageous for data integration can be adopted for building a rich database by combining independently constructed source data or different data types. Additionally, Ibrahim et al. (2021) and Ibrahim et al. (2022) have employed CSV and JSON network data, which can be managed using both relational and non-relational databases. Although their data had fewer schema restrictions than other networks with relational databases since they were stored in a document-based format, it remains difficult to modify the data model for a large dataset, similar to the relational database.

With respect to model configuration, previous multimodal transport network data models adopted a multilayer structure. Generally, multilayered transport networks are constructed by combining individual layers for each transport mode (Ibrahim et al., 2021; Idri et al., 2017; Natera et al., 2020; Tischner, 2018), with each sub-layer connected spatially using common stops in other networks (Huang et al., 2018). Idri et al. (2017) introduced the concept of transfer nodes for interconnecting different transport modes according to two temporal factor-based approaches: a time-dependent model with changeable cost function, and a time-expanded model with duplicated nodes for every time interval. Transfer actions were reflected with time costs in their model; however, they did not provide a detailed description of transfer routes. Bellocchi et al. (2021) reflected the temporal effects of configured transport networks by setting variable weights with time. In their approach, a transportation system was represented with a multilayered network including car, walking, bus, and metro modes. Although multiple transport modes were incorporated, their model was configured under the assumption of fixed topology and same node interconnections.

In the existing approach where independently constructed layers for each mode are mutually combined using common nodes, unnecessary nodes are created by the duplication of counterpart nodes in each layer. Further, transfer between layers through common nodes has less flexibility in describing various transfer actions. In summary, a database with good scalability that enables efficient data updates should be adopted for managing large datasets expressing complex and realistic networks spanning various modes. Furthermore, the multimodal transport database should be designed in detail to be adopted for any region, considering data redundancy, expandability, and efficient query performance.

To improve transport networks using non-relational databases, there have been attempts to utilize the graph database to represent transportation systems. The graph database has steadily attracted attention, and developed rapidly over the past decade (Wang et al., 2019). Fortin et al. (2016) analyzed transportation systems using graph databases by applying network metrics, but only covered the bus system. Maduako et al. (2018) proposed a time-varying graph to model the dynamic relationship between a transit network's topological structure, and the mobility patterns of vehicles. They implemented their model in the *Neo4j* (https://neo4j.com/) graph database using transit feeds, but only considered public bus transit. Elsewhere, Shibanova et al. (2021) proposed an approach for modeling road traffic flow using a graph database to describe movement patterns; however, their model focused on vehicle movement rather than dealing with transport system structure, reducing its suitability for deriving travel routes.

Maduako et al. (2019) summarized the advantages of graph databases via GIS applications, such as path-finding. Based on the benefits of the graph database, the authors developed a new graph data model accounting for the mobility and geographical contexts of transit networks comprised of two subgraphs: sequence of places, and sequence of moves and halts. Although temporal factors were considered as well, only bus routes were covered. Wirawan et al. (2019) also developed a graph database schema for multimodal transportation, where nodes of place, shelters, and Angkot stopper were included in their model. However, their data model described a single mode, and its simplistic design cannot readily be applied to true multimodal networks. Czerepicki (2016) constructed a graph database for the public transport system based on timetable information; however, the private transport mode data should be added to this database for use in multimodal routing, while transfer graphs must be designed in greater detail. Huang et al. (2018) proposed merged networks of public transport and carpooling for multimodal route planning via a time-expanded model implemented with the Neo4j graph database. The database design can be improved to enable graph filtering for memory-efficient routing that more precisely specifies relationship types. Also, it remains necessary to design a graph model for reflecting the temporal factor, while effectively reducing graph volume.

TABLE 1 Previous transport databases.

Research	Model composition	Interconnection between different modes	Purpose of model	Key feature of model	Graph database use
Hrncir and Jakob (2013)	Combination of time-dependent graph for public transit, and network graph for individual transport	Connecting stops to entrance or nearest pavement nodes	Journey planning	Adopting time-dependent model	N
Ma and Lebacque (2013)	Multilayered model for bus and metro	Using walking/transfer links between stations	Route planning	Using origin–destination (OD) network as reference layer	N
Gil (2014)	Model representing public transport, private transport, and building land use	Using transit interface links between systems	Mobility assessment	Including building land use system	N
Chondrogiannis et al. (2016)	Multilayered model for road and bus	Using connecting links between stops, and both end nodes of associated roads	Route planning	Adopting time-dependent model	N
Czerepicki (2016)	Graphs of public transport system	–	Route searching	Adopting timetable graph	Y
Fortin et al. (2016)	Graph of bus system	–	Analyzing transportation system	Adopting timetable graph	Y
Idri et al. (2017)	Multilayered model for transport modes (not fixed)	Using transfer nodes	Time-dependent routing	Adoting of time-dependent and time-expanded models	N
Jamal et al. (2017)	Multilayer model for public transportation and carpooling services	Using interlinks with walking time between car and bus layers	Journey planning	Adopting time-expanded model	N
Ding et al. (2018)	Multilayered model for street and rail	Creating interlayer links between rail stations with road nodes	Network structure analysis	Expandability for other modes, such as walkway	N
Maduako et al. (2018)	Graph of bus system	–	Network structure analysis	Adopting time-varying graph	Y
Talasila et al. (2018)	Multilayered model for train	–	Journey planning	Using timetable of trains	N
Bonnetain et al. (2019)	Multilayered model for road, bus, tramway, and subway	Using transfer stops	Map matching	Using parking information for transferring among modes	N

(Continues)

TABLE 1 (Continued)

Research	Model composition	Interconnection between different modes	Purpose of model	Key feature of model	Graph database use
Huang et al. (2018)	Multilayered model for public transit and carpooling	Using transfer edges between carpooling stops to nearest public transport stop	Route planning	Adopting time-expanded model	Y
Maduako et al. (2019)	Combination of two subgraphs for street and bus	–	Utilized GIS applications, such as shortest path finding	Considering temporal context	Y
Wirawan et al. (2019)	Graph of place, BRT shelters, and Angkot stopper	–	Route planning	Using three labeled nodes. and two typed relationships	Y
Natera et al. (2020)	Multilayered model for bicycle path, street, and rail	Using common nodes as transport hub	Identifying the potential for multimodal transport	Using the overlap census concept	N
Bellocchi et al. (2021)	Multilayered model for road, walk, bus, and metro	Creating interlayer links between same nodes in different layers	Traffic analysis—congestion and bottleneck detection	Using flexible weights by times	N
Hu et al. (2021)	Double-layered model for bus, railway, and air network	One layer is part of another	Route planning	Considering differences in transport speeds	N
Shibanova et al. (2021)	Model for road traffic flow	–	Movement pattern analysis	Focusing on vehicle movement	Y
Smarzaro et al. (2021)	Model for individual and collective transport modes	Using transfer segments and connection nodes	Not specified, but routing was performed as a case study	Including schema matching process	N
Ibrahim et al. (2022)	Two cases of two-layered models for road–rail and bus–rail	Using station nodes in both layers	Routing considering carbon efficiency	Expandability for a higher number of layers	N
Jing et al. (2022)	Double-layered model	One layer is part of another	Route planning	Considering both traffic velocity and network topology	N

Graph databases can be an optimal alternative for organizing transport data due to fewer constraints on stored information, higher efficiency of data management, and improved performance; however, existing graph database models should be expanded to multimodal, as their current target modes are limited. When increasing modes, interconnection issues on how to minimize duplicates should be addressed. Moreover, detailed data model descriptions, such as label and property specifications, are required to adapt databases for other areas. Expressly, optimal layer configurations should be provided for easier updates and responses to changes in the transportation system, such as the timetable of public transport. In addition, it is essential to provide an approach for efficiently building a database using existing transport data, so that a multimodal transport graph database can be easily constructed for any region. A summary of transport networks explored in previous studies can be found in Table 1.

3 | MULTIMODAL TRANSPORT GRAPH DATABASE

3.1 | Conceptual model for multimodal transport in graphs

Interconnected metropolitan transport systems include private travel (e.g., cars, bikes and walking), along with public transit modes, such as subways and buses (Bellocchi et al., 2021). The most basic mode of travel is by foot, which can cover short trips (Orozco et al., 2021). Bicycles or buses can cover medium-distance travel; whereas long-distance travel is most reasonably accomplished via rail systems or cars (Varga et al., 2016). Accordingly, cities above a certain size require a balanced mix of different modes to reflect an even distribution of travel distances (Orozco et al., 2021). Walking, driving, and cycling are typical private transit modes whereas buses, metros, tubes, and trains are regular public transit modes, with a combination representing people's inner-city travel. In this context, the proposed conceptual model of multimodal transport here included the following modes, considering both private and public transit of people: Walking, driving, cycling, bus, and rail (under- and overground).

Public transit modes largely operate over established routes and directions, moving between specific stops and stations in a predetermined order. Accordingly, graphs for public transit modes can be represented as a sequence of bus and rail stops and stations that are represented as nodes (Figure 1, orange and yellow nodes), with the connections between nodes expressed as relationships according to their predetermined sequence for each route/line. Rails operate through independent facilities separate from the road, allowing movement between stations at relatively fixed time intervals. Alternatively, buses move along roads, and the corresponding travel time between stops varies with traffic conditions. Therefore, a rail graph designates travel time between stations as a relationship property; whereas the bus graph stores the network distance between stops as a fundamental property to be used as costs when routing.

Conversely, private transit modes do not move according to a fixed node order. When modeling roads as graphs, an intersection can be represented as a vertex, and a segment between two adjacent intersections as an edge (Wang et al., 2019). Referring to this general structure, intersections were created as nodes, and road/path segments connecting associated intersections for movement were built as relationships in the present model. Relationships can be used to identify lowest cost routes; thus, travel costs, such as distance and travel time, are set as relationship properties.

Each mode's layers can be identified through labels, and via multi-labelling, it is possible to distinguish modes, routes, and lines for public transit. A label can also identify the type of transport for private transit modes, such as driving, walking, and cycling. Although it is possible to distinguish modes through labels, each layer is not independently constructed. Unlike previous multilayer networks, the present model is horizontally expanded, and does not include duplicate nodes (i.e., 'common nodes' or 'transfer nodes') between modes. Instead, modes are connected via graphs for walking. Notably, almost all transfers between different transport modes include minimal walking. For example, people have to walk to enter a subway station to transfer

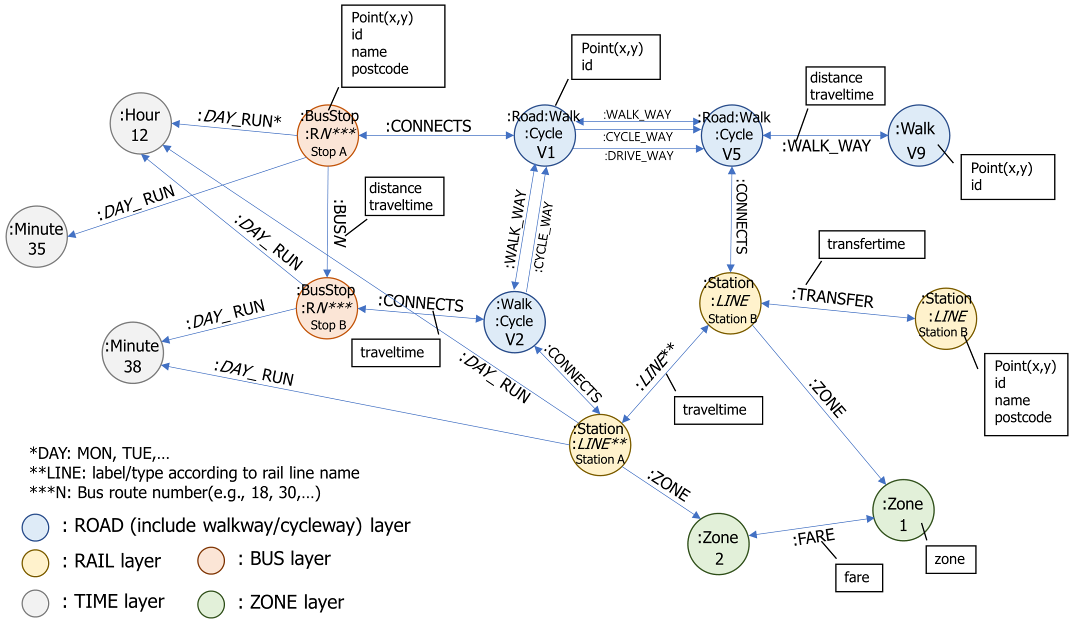


FIGURE 1 Conceptual model of multimodal transport graph database.

after getting off a bus, or people must exit public transport and walk to their destination as last mile travel. In this context, the proposed model included a basic walkway layer for interconnecting different transport modes.

Although most nodes in roads for cars, walkways for pedestrians, and cycleways are duplicated, each layer can be configured without creating additional unnecessary nodes by incorporating multiple labels, such as *n:Road:Walk:Cycle* (Figure 1, blue nodes). Walkways are defined as bidirectional relationships, as they are used to travel in either direction, regardless of road orientation. Further, bus stops and rail stations are connected to the start nodes of the nearest walkway link with specified relationships, where the transfer cost can be stored as a property in those relationships connecting graphs of public transit to walkways.

In the present model, an independent time layer was configured so that timetable information could be reflected by linking entities of public transit modes to time. Previously, transit graphs were created according to operating time, or available time information was set as individual entity attributes. However, the time-dependent model of buses, tubes, and trains is integrated with the time-independent model of walking, driving, and cycling in the present conceptual design. Therefore, connections are created only for modes that require temporal contexts by configuring the time layer independently. Accordingly, the present model can improve query efficiency by forming a light subgraph through graph projection, further allowing for flexible configuration by modifying only relevant relationships when timetables are changed.

Here, the time layer is modeled by referring to the time-tree concept of Maduako et al. (2018), consisting of a hierarchical temporal index structure supporting time indexing, from years to milliseconds. By adopting a time-tree from hours to minutes, 24-h-labeled nodes and 60min-labeled nodes were created and connected with public transit modes (Figure 1, gray nodes). Concurrently, differences in operating times according to the day of the week were considered through changeable relationships (Figure 1, *DAY_RUN* relationship).

Depending on the transport system of each country, zone division may be used to apply different fares for public transport. Therefore, a zone layer was also defined to be connected with public transit modes, and store flexible fare information as a relationship property (Figure 1, green nodes), allowing also for multiple relationships to be created if there are predefined fare types by time (e.g., peak fare/off-peak fare).

3.2 | Structure of multimodal graphs

The conceptual graph model of multimodal transport can be configured by a graph structure denoted by $G_{\text{multimodal}} = (E, R, f_t, PE, PR)$, where E is the set of entities with specific labels; R is the set of multi-typed relationships; the function f_t represents the topological connection between entities; and PE and PR denote properties of entities and relationships, respectively. When considering routing, entities and relationships are defined while preserving network-based topological relationships for each mode.

3.2.1 | Entity

Seven labeled entities were defined in the graph models here: *Road* (E^R), *Walk* (E^W), *Cycle* (E^C), *Stations* (E^S), *Bus* (E^B), *Time* (E^T), and *Zone* (E^Z), where transit by railways (e.g., tube and train) can be represented by *Stations* (E^S), and *Time* (E^T) configures the independent time layer regardless of transit. Each entity configures a subgraph for transit by each mode, where a time-dependent subgraph can be filtered through the connection between public transit nodes to *Time* entity nodes. A detailed description of the graph configuration is as follows:

$$E = < E^R \cup E^W \cup E^C \cup E^S \cup E^B \cup E^T \cup E^Z > \tag{1}$$

Road (E^R) represents junctions and end points of road networks. Multiple labels are set to this entity when *Road* nodes are included in *Walkways* and *Cycleways*.

Walk (E^W) represents junctions and end points of pedestrian networks. Most *Road* nodes can also be labeled as *Walk* nodes.

Cycle (E^C) represents junctions and end points of cycle ways.

Station (E^S) represents each tube and train station. All nodes of this entity share a common *Station* label, and are distinguished with a *Line* label. For stations where ≥2 lines pass through, nodes are duplicated by each line, and have the line name as a label (e.g. (*n1:Station:Northern* {name: 'Warren Street'}), (*n2:Station:Victoria* {name: 'Warren Street'})).

BusStop (E^B) represents each bus stop along the road with a common *BusStop* label. Even for the same bus stop, nodes are duplicated by each bus route, and a separate label is assigned using the combination of '*route number*', while the identical bus stop has the same code property (e.g. (*n:BusStop:R18* {code: '36542'})).

Time (E^T) is composed of *Hour* and *Minute* entities, with corresponding labels. All such nodes have the *Time* label in common, while 24 nodes were created for the hour entity (0–23) and 60 nodes were created for the minute entity (0–59).

Zone (E^Z) is an optional entity composed of six nodes with *Zone* labels (1–6).

3.2.2 | Relationship

$G_{\text{multimodal}}$ is divided into bi- and unidirectional components, including a total of ten typed relationships (five each). Buses travel in a predefined direction by routes, and vehicle travel in a regulated direction for each road segment, while cycles run in the same direction as other vehicles. Therefore, routes for these three modes are expressed through unidirectional relationships; whereas the remaining modes maintain bidirectional relationships. Additional information related to transport systems, such as timetable and zone (fare) connections, are non-directional components, and are thus modeled with bidirectional relationships.

$$R = < R^B \cup R^U > \tag{2}$$

$$R^B = E_i \leftrightarrow E_j, \text{where } E_i, E_j \in E \tag{3}$$

$$R^U = E_i \rightarrow E_j, \text{where } E_i, E_j \in E$$

$$f\left(R^B\right) = \begin{cases} \mathit{WALK\_WAY} & \mathit{if}\ E_i \in E^W \wedge E_j \in f_t^a\left(E^W\right) \\ \mathit{LINE} & \mathit{if}\ E_i \in E^S \wedge E_j \in f_t^c\left(E^S\right) \\ \mathit{CONNECTS} & \mathit{if}\ E_i \in \left(E^B \cup E^S\right) \wedge E_j \in f_t^s\left(E^W\right) \\ \mathit{TRANSFER} & \mathit{if}\ E_i \in E^S \wedge E_j \in f_t^m\left(E^S\right) \\ \mathit{FARE} & \mathit{if}\ E_i \in E^Z \wedge E_j \in E^Z \end{cases} \tag{4}$$

$$f\left(R^U\right) = \begin{cases} \mathit{DRIVE\_WAY} & \mathit{if}\ E_i \in E^R \wedge E_j \in f_t^a\left(E^R\right) \\ \mathit{CYCLE\_WAY} & \mathit{if}\ E_i \in E^C \wedge E_j \in f_t^a\left(E^C\right) \\ \mathit{BUSN} & \mathit{if}\ E_i \in E^B \wedge E_j \in f_t^c\left(E^B\right) \\ \mathit{DAY\_RUN} & \mathit{if}\ E_i \in \left(E^B \cup E^S\right) \wedge E_j \in f_t^m\left(E^T\right) \\ \mathit{ZONE} & \mathit{if}\ E_i \in E^S \wedge E_j \in f_t^m\left(E^Z\right) \end{cases} \tag{5}$$

According to the following rules, appropriate relationships for each type can be created.

Rule 1. The topological relationship of f_t is classified into four cases: $f_t = \{f_t^a, f_t^s, f_t^c, f_t^m\}$, where f_t^a assigns the associated entity to each entity, f_t^s assigns an entity to another with the shortest distance, f_t^c assigns a consecutive entity to each entity, and f_t^m assigns a matched entity under specific property conditions.

Rule 2. *Road*, *Walk*, *Cycle* entities are connected to associated *Road*, *Walk*, *Cycle* entities along the road and path (with walkway and cycleway) networks based on the topological connections of network nodes.

Rule 3. *DRIVE_WAY-CYCLE_WAY*-relationships are created unidirectionally according to the direction of travel; however, roads represented by single lines for travel in both directions, are modeled in bidirectional relationships.

Rule 4. Unidirectional relationships are created for bus routes, where the type of relationship between consecutive *BusStops* is defined by the route number (e.g.(*n1:BusStop*)-[:BUS30]->(*n2:BusStop*)).

Rule 5. Type of relationship between consecutive *Stations* is defined by the line name (e.g. (*n1:Northern* {name: 'Warren Street'})-[:Northern]->(*n2:Northern* {name: 'Euston'})).

Rule 6. Each *BusStop* and *Station* are connected to a *Walk* entity within the shortest distance via a *CONNECTS*-relationship.

Rule 7. Identical *Stations* with different *Line*-labels are connected with *TRANSFER* type relationships.

Rule 8. Each *BusStop* and *Station* are connected to a *Time* entity based on a timetable of operating buses and tube/trains by day using *DAY_RUN* type relationships.

Rule 9. Different fares based on connections between zones are stored as a property of *FARE*-type relationships. If fares are diversified by timeframe, multiple relationships can be created (e.g., *PEAKFARE-/OFFPEAKFARE-* relationships).

3.3 | Implementation

Multimodal transport networks were constructed using a Neo4j graph database, currently the most popular native graph database available (Liu et al., 2021; Maduako et al., 2018). The location and properties of nodes and relationships representing node-to-node connections were derived from the prebuilt dataset for transport, such as road networks. Multi- or single-labels were assigned to each node, and relationships were built with a specified single type. Layers were identified with node labels and relationship types, while light subgraphs could be configured through filtering by label and type before querying optimal routes. Both nodes and relationships can have properties in the Neo4j graph database, so detailed properties (e.g., road name, stop/station name, and routing costs) are stored as corresponding properties.

Bus and rail graphs where nodes and relationships should be created for each route or line were built through iterative execution of Cipher commands for relationship creation; whereas roads and paths have a large number of nodes and links, as well as single labels and types for driving, walking, and cycling. Accordingly, bulk processing is required to create graphs for private transit modes. To efficiently create vast nodes and relationships, commands are required to be sent in bulk by a single transaction. Here, the Neo4j CSV importer was used for bulk creation of private transit graphs. A map for each relationship containing both end nodes and values of properties was prepared in CSV documents from prebuilt road and path networks. Subsequently, private transit graphs were created from the CSV files based on the proposed model.

Figure 2 emphasizes how to construct driveway, walkway, and cycleway relationships from a prebuilt transport dataset, such as road networks. In the prebuilt transport network, the road is modeled in two ways: dual lines and single lines. Since dual-line roads specify the direction of vehicle travel, the direction of *DRIVE_WAY*- and *CYCLE_WAY*- relationships are also defined according to this direction. Alternatively, bidirectional relationships are created because single-line roads include cases where two-way travel is available.

3.4 | Evaluation criteria

While traditional relational databases have a standard benchmark for evaluating performance, no such equivalent exists for graph databases (Chu et al., 2020). Chen et al. (2020) presented a benchmark to measure the performance of graph databases compared with the relational databases, measuring the time for loading a graph and executing shortest path algorithms, such as K-shortest path and A*; however, this benchmark cannot be directly used to evaluate the database in the present study for several reasons: the created graphs are structurally different from those used here, and the present multimodal path-finding results are substantially more complex.

Elsewhere, the performance of multimodal routing has been confirmed by measuring the runtimes of various graph queries using the self-constructed database (Debrouvier et al., 2021; Giannakopoulou et al., 2019; Hrncir & Jakob, 2013; Idri et al., 2017; Potthoff & Sauer, 2022; Tischner, 2018). For example, Giannakopoulou et al. (2019) estimated graph querying and preprocessing times; whereas Tischner (2018), Idri et al. (2017), and Debrouvier et al. (2021) measured the change in query time according to shifts in graph size, number of nodes, and the number of OD pairs.

Referring to previous research, the runtimes for three processes were checked here to see if the graph database performed well for multimodal routing problems: graph creation, graph loading, and graph querying. Two tests were then performed: First, the graph creation time was measured while building a graph database according to the proposed model structure. Further, graph projection time was measured while projecting graphs for each mode. Second, the routing query time was measured to check the performance of the proposed database in multimodal routing. Additionally, routing was performed under multiple condition combinations, and the

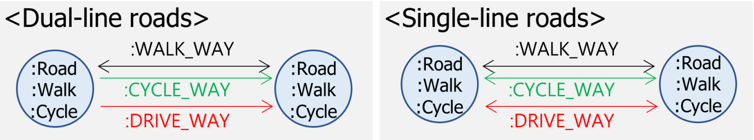


FIGURE 2 Different modeling in dual-line and single-line roads.

results were discussed qualitatively. Execution times can be affected by the graph size, i.e., the counts of nodes, relationships, and properties in a graph. The numbers of nodes and relationships for each transport mode were examined to find the influence of the graph size while measuring graph creation and projection times. Since the number of properties is not significantly different (most relationships of the proposed model have two common properties, distance and travel time), we focused on the time difference according to the number of nodes and relationships. Additionally, the numbers of nodes and relationships for the projected graphs used in multimodal routing were also discussed. A more detailed description of each measure is given below:

3.4.1 | Graph creation time

After inputting the prebuilt spatial network, labeled nodes and typed relationships were automatically created in the Neo4j graph database based on a previously defined conceptual model through the proposed framework. During this process, the time for creating the whole graph was checked.

3.4.2 | Graph projection time

Fast query is possible by filtering only necessary subgraphs that meet the routing conditions, and then projecting them for graph querying. After configuring a subgraph for each mode, the time for completing projections can be measured.

3.4.3 | Graph querying time

After setting the origin and destination in the projected graph, the time to explore optimal routes by applying the Dijkstra algorithm was checked. OD pairs were randomly selected in this process, and the time index was measured while adjusting two parameters. First, the number of nodes was varied (or the number of OD pairs in the case of graph querying time) to verify whether they were suitable for large amounts of iterative query processing. Second, the number of stops were adjusted to assess the proposed graph database's flexibility to respond to multi-stop routing.

4 | CASE STUDY

4.1 | Data and test area

Given its comprehensive and developed multimodal transport system, London was selected as the case study area. As explained in Section 3.1, both private and public modes of walking, cycling, driving, bus, and rail were considered here. London's rail system involves an underground tube and overground train; thus, both tube and train were included in the database. Referring the relevant public transit information, 796 bus routes, and 28 tube and train lines were included in the database.

A test block in London was set for the routing test to verify the efficiency of the proposed database in a multimodal routing problem (Figure 3). The area is suitable for multimodal routing experiments, as it encompasses various rail lines, including Circle, District, Hammersmith & City, Northern, Bakerloo, Central, and Victoria, as well as the 36 bus routes that pass through this area. The test block also includes Regents Park, which has paths for pedestrians only.

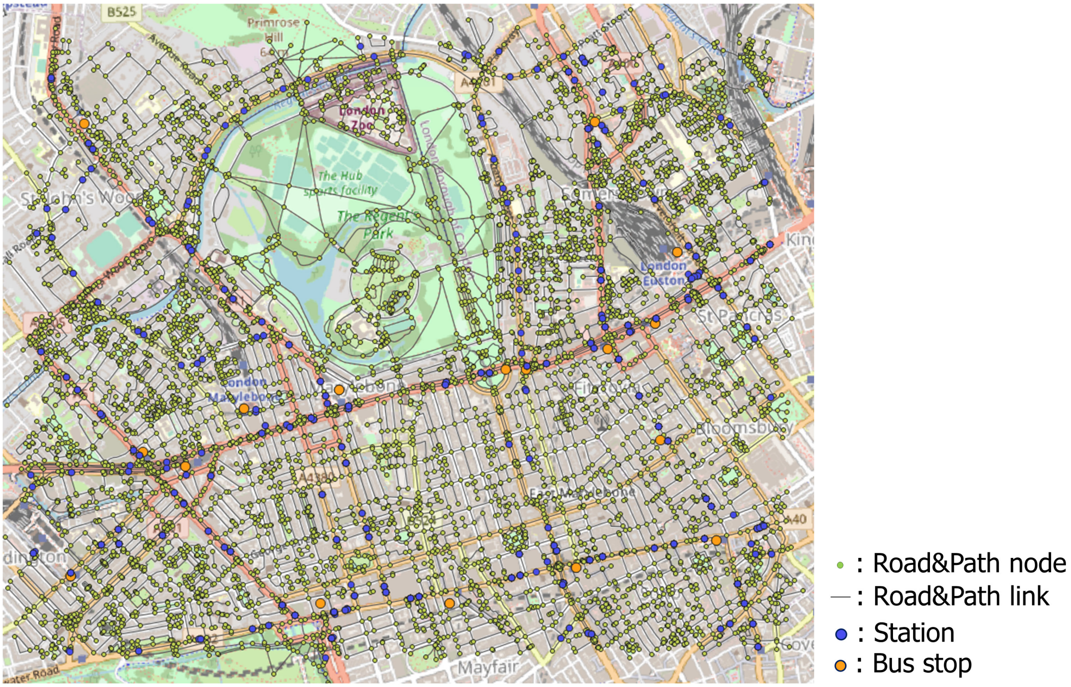


FIGURE 3 Test block for multimodal routing analysis.

TABLE 2 Description of data sources.

Mode	Description of source data
Rail (tube/train)	• Individual station data (including longitude, latitude) • Sequence of stations by each line
Bus	• Individual bus stop data (including longitude, latitude) • Sequence of bus stops by routes provided from National Public Transport Access Nodes (NaPTAN) database • Bus route network to extract network distance between consecutive stops as 'distance' property of relationships
Road Walkway Cycleway	• OS highway road/path network • OS highway dedication data for classification of modes (driving, walking, cycling)

Prebuilt transport data (e.g., road networks) were used to construct the graph-based multimode database, consisting of data presented in Table 2, which included information from Transport for London (TfL) and Ordnance survey (OS). Those independently constructed networks were integrated into a single graph database based on the aforementioned approach.

4.2 | Graph database construction using prebuilt dataset

The multimodal transport graph database was constructed using the Neo4j (*v*.4.2.3) graph database. The graph creating, projecting, and querying were conducted on an Intel Core i7-1165G7 processor with 16 GB of RAM.

Bus and rail layers were commonly built using the sequence of stations and stops. First, nodes of *Station* and *BusStop* were created from station/stop point data. Specifically, nodes through which different lines and routes pass were duplicated with different labels; for example, there are two nodes for Warren Street station: (*n1:Northern* {name: 'Warren Street'}), (*n2: Victoria* {name: 'Warren Street'}). According to the sequence, relationships were only created after searching two nodes of consecutive stations/stops. For the rail, a *LINE*-typed bidirectional relationship was created, where same station nodes with different line labels (e.g., 'Warren Street') were connected with *TRANSFER*-type relationships, taking transfer time cost as a property. Six *Zone* nodes (1–6) were created and connected to *Station* nodes belonging to each zone. In London, there are two types of fares according to the time of day: peak and off-peak; therefore, *Zone* nodes were connected to each other with *PEAKFARE/OFFPEAKFARE* relationships (e.g. (*n:Zone* {zone:1})-[:*PEAKFARE* {fare: 3.2}]-(*m:Zone* {zone:2})). For the bus, a more complex procedure was required to calculate the property of network distance between each connected stop pair, as shown in Table 3.

Road and path networks were composed of node and link layers, where links can be mapped with both start and end nodes. Multiple node labels and relationship types were specified according to the availability by modes, as stored in OS highway dedication data. OS highway road networks do not have directional information; thus, bidirectional *DRIVE_WAY*-relationships were created in the case study here. The closest nodes among two end nodes of the nearest link for each station and stop nodes were selected to form the *CONNECTS*-relationships to be used for identifying routes with mode transfers.

For multimodal routing, a virtual bus timetable was designated, and *BusStop* nodes were connected to *HourMinute* nodes with *DAY_RUN*-type relationships based on the operating schedule. Through this connection, the timetable of public transit modes included in the optimal route can be provided, or efficient routing can be performed by forming a filtered graph fit for specified departure and arrival times.

For the bus, driveway, walkway, and cycleway, after calculating the network distance between nodes using the spatial network, these distances were stored as a 'distance' property of relationships. Travel time can thus be estimated by applying speed to the network distance for each mode, and these properties of 'distance' and 'travel time' can be used as routing costs. Notably, it is possible to extract the network distance rather than the Euclidean distance between the connected nodes using the spatial network so that precise pathfinding based on actual travel routes are obtained.

4.3 | Results

4.3.1 | Multimodal graph database for London

Across all transport nodes in London, 1,079,962 nodes were generated, including: 981 *Station*; 57,413 *BusStop*; 215,941 *Road*; 590,915 *Walk*, and 215,693 *Cycle* (214,899 were created with the: *Road:Walk:Cycle*; 794 with:

TABLE 3 Process of constructing bus graphs.

All sub-procedures should be conducted by each route separately

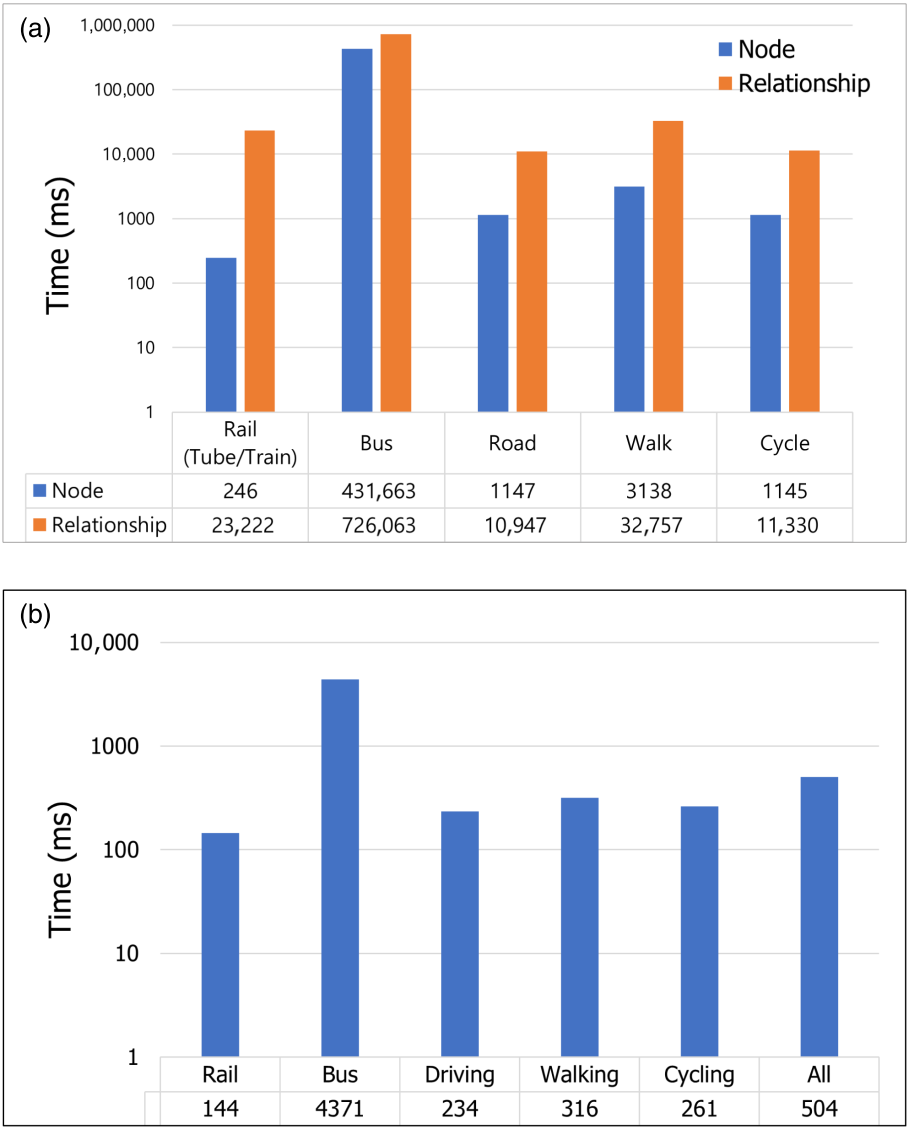
1. **Snap** bus stop points to bus route network, then **split** bus network as the segment with snapped bus stop points
2. **Find two intersecting stops** for each segment through **spatial join**
3. **Calculate** segment length as a **distance** attribute

Performed using a single process, and iterate until all segments are processed

1. **Create nodes** of *BusStop* entity with location information
2. **Put** *Route* **label** (e.g., *R30*)
3. For each segment, **find matched stops** and **connect** each other **with unidirectional relationships**; relationship types can be assigned with *Route* information (e.g., *BUS30*)
4. **Set distance** and **travel time** property using distance attribute of segments

Walk:Cycle; 375,222 with only: *Walk*-; and 1042 with only: Road-labels). The graph database included 6208 *LINE*-; 113,258 *BUSN*-; 867,702 *DRIVE_WAY*-; 2,279,126 *WALK_WAY*-; and 864,990 *CYCLE_WAY*-relationships. 6 *Zone* nodes were created, and 1085 *PEAK-/OFFPEAK-/ZONE*-relationships were connected.

Figure 4a is the result of measuring the graph creation time for each mode. The whole multimodal graph database for the test block was built within 21 min (1242 s), with the majority of this time (19 min) spent on creating bus nodes and relationships. Since bus nodes have the most diverse types of route number labels, creation time is relatively long compared to other modes. Similarly, creating the 796 types of relationships with route numbers for buses was found to be the most various types. Road, Walk, and Cycle nodes were created simultaneously over 5.43 s, and the time divided by the ratio of the number of nodes is plotted in Figure 4a. Notably, these three private modes were created in bulk due to a single node label and relationship type. In contrast, public modes were created iteratively by line/route due to differentiated labels and types for each line/route. Therefore, although the absolute number of nodes and relationships was highest for the private transit modes of driving, walking, and



	Rail (Tube/Train)	Bus	Road	Walk	Cycle
🟦 Node	246	431,663	1147	3138	1145
🟧 Relationship	23,222	726,063	10,947	32,757	11,330

Rail	Bus	Driving	Walking	Cycling	All
144	4371	234	316	261	504

FIGURE 4 Execution times for: (a) graph creation; and (b) graph projection with log scale.

cycling graphs, the graph creation time for these private modes was shorter than that for the public transit modes of rails and buses.

Subgraphs only with analytically relevant topological and property information can be configured via graph projection (Neo4j Graph Data Science Library Manual v.1.8 (https://neo4j.com/docs/graph-datascience/current/). Further, this subgraph with compressed data structures can be entirely stored in-memory. Filtered graphs can be projected by specifying relevant nodes and relationships in a fine-grained manner; whereas graphs configured with only necessary parts can be projected to be efficiently queried while routing. For example, suppose a user prefers to select only driving routes without transferring to public transit. In such a case, it is possible to reduce the number of nodes touched when executing routing queries by creating subgraphs consisting of only road nodes and *DRIVE_WAY* relationships. Also, in the case of routing under specific time conditions, queries can be performed after filtering for public transit connected to time nodes that satisfy a given condition. Figure 4b presents the results of measuring the projection time by each mode, where times were measured as the average execution time of three consecutive projections after restarting the database. Here, projections were performed very quickly, with each mode except for buses completed within 1 s.

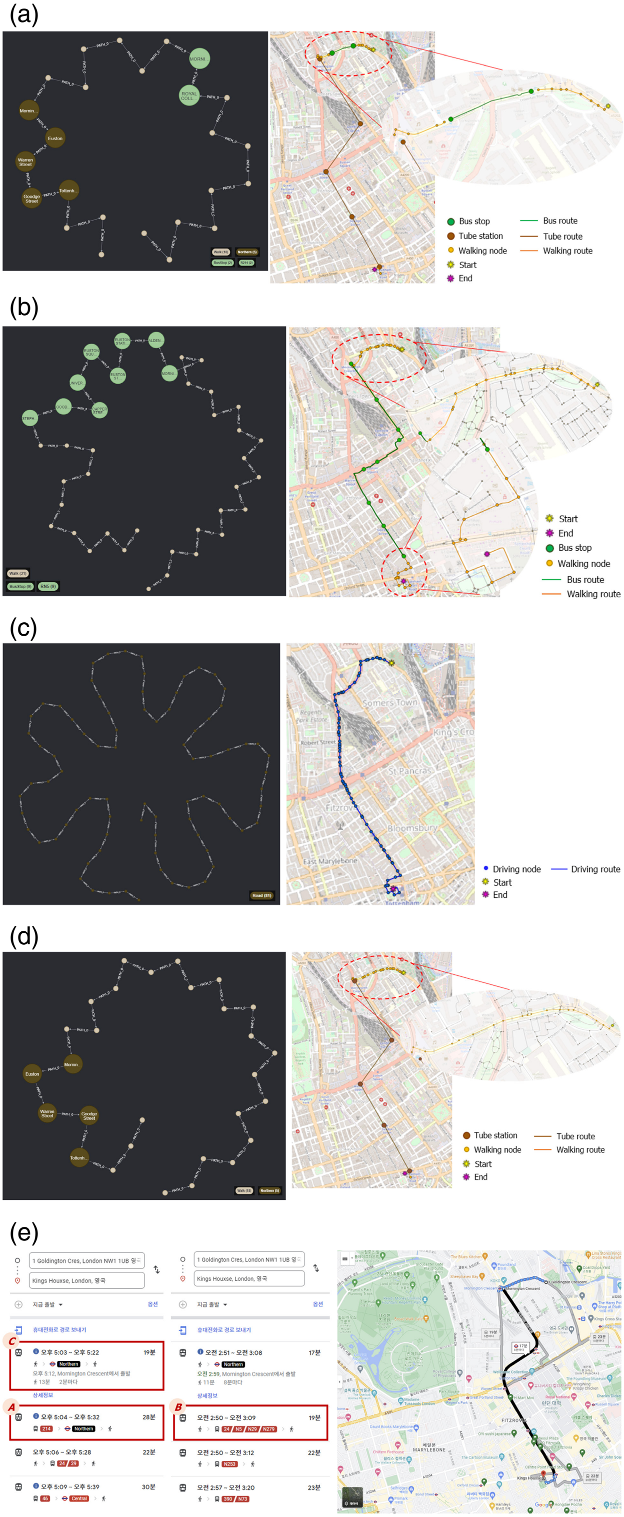
As the number of generated nodes and relationships increased, the time required for both creation and projection also increased. In particular, the driving graph and cycling graph, which had similar numbers of nodes and relationships, exhibited comparable creation and projection times. However, buses were projected with various typed relationships for each route number, while private modes were connected through single typed relationships. Therefore, the projection time for the bus is relatively longer than those of private modes, although the graph size of the bus is smaller than modes of driving, walking, and cycling.

4.3.2 | Multimodal routing results

To verify the utility of the proposed model, multimodal routing was performed using the constructed graph database. The graph size of the test block is: the number of nodes and relationships are 8808 and 52,203, respectively. Here, two types of routing were conducted: multimodal routing with three different combinations of [Start, End, Mode, Preference], and multi-stop routing with multimode. For multimodal routing, the fastest route with all modes was derived first after setting two random points, A and B, as an OD pair. Secondly, routing by limiting to two transport modes (walk and bus) were performed for the same OD pair. Lastly, the routing with all modes, but under minimum-transfer between the same OD pair, was constructed. For the multi-stop routing, after five random stops and the order of visits were established, the fastest route through three intermediate stops was derived for all modes.

Neo4j Graph Data Science (v.1.8.2) was used to execute routing queries based on scenarios after selecting a random OD pair. Since the speed varies by transport mode, the average speed for each mode was applied to calculate the travel time costs. The test's purpose was not to extract accurate routes or verify the routing algorithm, but rather to check the applicability of the constructed graph database based on the proposed model in the multimodal problem. Accordingly, routing was performed by simplifying route costs, and applying the Dijkstra algorithm under the following time assumptions: the average adult walking speed was 1.3 m s^{-1} (Mohamed & Appling, 2020; Waters et al., 1988), the average driving speed was 9.5 mph (https://www.london.gov.uk/who-are-we/what-london-assembly-does/questions-mayor/find-an-answer/average-traffic-speed-london-0), bus speed was 9.3 mph (https://www.london.gov.uk/questions/2020/1423), and travel time between two stations was 2 min. Transfer costs between different rail lines was set as 3 min, and the time cost of the *CONNECTS*-relationships connecting *BusStop* and *Station* nodes with the nearest walkway was set to 1 min.

The routing results for four different combinations of [Start, End, Mode, Preference] are shown in Figure 5: [*A*, *B*, all, fastest], [*A*, *B*, walk-bus, fastest], [*A*, *B*, drive only, fastest], and [*A*, *B*, all, minimum-transfer]. The left sides of the



1 Goldington Cres, London NW1 1UB 영국

Kings Houxse, London, 영국

지금 출발 옵션

휴대전화로 경로 보내기

오후 5:03 ~ 오후 5:22 19분
Northern
오후 5:12, Mornington Crescent에서 출발
13분 2분마다
상세정보

오후 5:04 ~ 오후 5:32 28분
214 Northern

오후 5:06 ~ 오후 5:28 22분
24 / 29

오후 5:09 ~ 오후 5:39 30분
46 Central

1 Goldington Cres, London NW1 1UB 영국

Kings Houxse, London, 영국

지금 출발 옵션

휴대전화로 경로 보내기

오전 2:51 ~ 오전 3:08 17분
Northern
오전 2:59, Mornington Crescent에서 출발
11분 8분마다
상세정보

오전 2:50 ~ 오전 3:09 19분
24 / N5 / N29 / N279

오전 2:50 ~ 오전 3:12 22분
N253

오전 2:57 ~ 오전 3:20 23분
390 / N73

FIGURE 5 Routing results: (a) fastest routes using all modes from *A* to *B*; (b) fastest routes using bus and walking from *A* to *B*; (c) fastest routes by driving from *A* to *B*; (d) routes with minimum-transfer using all modes from *A* to *B*; and (e) routes from *A* to *B* in Google Maps.

figure show the graph results in Neo4j, and the right sides display the results visualized with the spatial transport networks. The result of the fastest path finding considering all modes from *A* to *B* is shown in Figure 5a, with the recommended route to walk from the departure point to the nearest bus stop, board bus 214 for one stop, walk from the bus stop to Mornington Crescent Station to transfer to the Northern line tube, travel to the Tottenham court road station via the Northern Line, and reach the destination on foot. It was confirmed that the route returned from the constructed graph database was included in the optimal route list from Origin *A* to destination *B* in Google Maps, a representative online navigation service (marked with *A* in Figure 5e). However, despite being included in the optimal route list, this route was not the fastest in the list from Google Maps; because the travel time cost of the returned route differed between our framework and Google Maps, being 22 and 28 min, respectively. This discrepancy was caused due to experimental assumptions, such as transfer time between different transport modes and average walking or bus speed.

For the same OD pair, the routing results with only walk and bus modes are shown in Figure 5b. A filtered graph with nodes and relationships relevant to buses and walkways was projected before routing to query the graph for specified modes. If traveling via bus alone, the N5 bus was identified as the fastest route, different from the result in Figure 5a. Since no specific time conditions were specified, the night bus was chosen, with the remaining optimal route comprised of boarding the N5 at the nearest stop from the departure point, traveling eight stops, getting off the bus, and walking to the destination. As a result of routing for the identical OD after setting the departure time to late at night, it was found that travel through the N5 bus, which was derived in this study, was one of the optimal routes in Google Maps (marked with *B* in Figure 5e).

Meanwhile, the driving routes from origin *A* to destination *B* are illustrated in Figure 5c. By visualizing the returned route graph onto the transport network, it was confirmed that the resulting graph appropriately followed the driveways. However, as previously mentioned, the returned graphs may have some discrepancies with the actual road conditions due to the assumption of bidirectional traffic on the driveway.

For three scenarios of exploring the fastest routes using different modes (i.e., all modes, walk-bus, driving), the querying times were examined in terms of the projected graph's size. The size of the projected graphs decreased in the following order: all modes, walking and bus, and driving. The routing query for each scenario was executed after restarting the database and projecting the graph for specific modes. Queries were consecutively performed three times; then, the average execution time was calculated. The execution time for the routing query from origin *A* to destination *B* using projected graphs for all modes, walk-bus, and driving modes were 261, 224, and 146 ms, respectively. That is, it was found that the execution time of the routing query was affected as the projected graphs' size increased.

Routing was performed using the same OD pair under the minimum-transfer condition, with the results shown in Figure 5d. A route that moves through only the tube without any transfer was queried, while minimizing the number of transfers by increasing the transfer cost. Here the optimal route was identified as walking to Mornington Crescent Station on the Northern Line closest to the starting point, traveling to 4 stations, and reaching the destination on foot. This route from Mornington Crescent Station to Tottenham Court Road Station by Northern Line was also derived in Google Maps as one of the optimal routes (marked with *C* in Figure 5e).

Accordingly, different paths were appropriately derived according to input conditions in the multimodal routing test. Moreover, it was possible to explore routes that freely reflected transfers between different modes. The routing results from Google Maps did not precisely match our findings; nonetheless, derived routes from the graph database are also compatible with those from Google Maps. Although the applicability of the constructed graph database in multimodal problems could be confirmed through the test, it remains necessary to set elaborate cost values. Since the identified route in the first scenario was a result of considering the predefined travel time according to the assumptions made, the optimal route may change depending on how additional time costs, such as the time to get on and off the bus, were reflected. Also, fixed transfer times between different rail lines were used in this test; however, the actual times required for each transfer case varies. Accordingly, the route that minimizes total travel time may vary depending on transfer time settings and calculations; thus, more accurate routing can be attained by considering actual travel times between stations and transfer costs between different lines.

Figure 6 displays the result of the multi-stop routing test using the constructed graph database. Here, travel from the starting point to Stop 1 was achieved through a Circle Line of the tube (Figure 6a, Section A). After walking to the nearest station, mixed travel using the tube and walking was identified. Next, travel between Stop 1 and Stop 2 included tube travel using the Victoria Line from Warren Street to Euston. Similarly, the journey from Stop 2 to Stop3 also included tube transit, and a route back to Euston Station on foot (Figure 6a, Section B, where this sub-route is further visualized using a transport network in the lower right corner of Figure 6b). Specifically, routes moving from Euston via Victoria Line, and transferring to Central Line at Oxford Circus to Marble arch were queried, and Stop 3 can be reached on foot after getting off at Marble arch station (Figure 6b, lower left). Again, walking to the adjacent bus stop is required to get to the bus from Stop 3 to the final destination. The final route requires taking bus 98 to the vicinity of the destination, then walking the remainder of the distance. In summary, multimodal routes, including various cases, such as travel through tubes and buses, transferring between different tube lines, and walking to stations and stops, were effectively extracted using the built-in graph database.

4.3.3 | Graph querying time in multimodal routing

The graph querying times in multimodal routing were measured in two ways: times dependent upon increasing the number of OD pairs, and those dependent upon increasing the number of intermediate stops. *N* OD pairs were selected randomly, and the fastest paths were identified by applying the Dijkstra algorithm. Route queries were continuously executed by increasing the OD pairs from 10 to 2000, and each query time was measured (Figure 7a). As a result, it was confirmed that the query time increased linearly to the number of OD pairs. That is, stable query performance was guaranteed regardless, of the total number of iterative computations. Furthermore, the findings of the multi-stop routing test with increasing intermediate stops showed that the increasing rate of query time was lower than that for the increasing rate of the number of visits (Figure 7b). Consequently, the efficiency of using a graph database in multi-stop routing was confirmed. In the proposed framework, multi-stop routes can be derived without limiting the number of stops entered using the graph database, while existing navigation services tend to limit the number of stops for multi-stop routing (e.g., a maximum of 10 stops can be set in Google Maps).

Furthermore, tests were performed in two environments: cold and hot, where the cold run is the query execution performed immediately after rebooting the database or flushing caches; whereas the hot run is the query execution conducted without flushing any cache after the cold run (Rishe et al., 2000). Not only was the graph database's performance checked, but the more rapid hot effect was also confirmed.

4.3.4 | Integrated timetable and fare information

Here, the graph database effectively integrated different types of information due to its flexible schema. Considering this advantage, additional information, such as timetable and fares, were modeled as separate layers, along mode-based routes in this study. First, *DAY_RUN*-relationships were added by setting virtual timetable information for some bus routes. Figure 8a shows the results of querying travel from the ‘Old Marylebone Town Hall’ bus stop, to the ‘Euston Square Station’ bus stop, with all available bus routes searched. As a result of querying travel after assuming departure before 12:30 pm on Monday, only bus 30 was identified for the same OD pair (Figure 8b), where boarding is available at 12:25. In contrast, only bus 18 was identified when assuming departure after 12:30 on Monday (Figure 8c), with a 12:36 departure. Thus, for public transit with a designated timetable of operations, the temporal factors can be easily combined with the travel route through the proposed model.

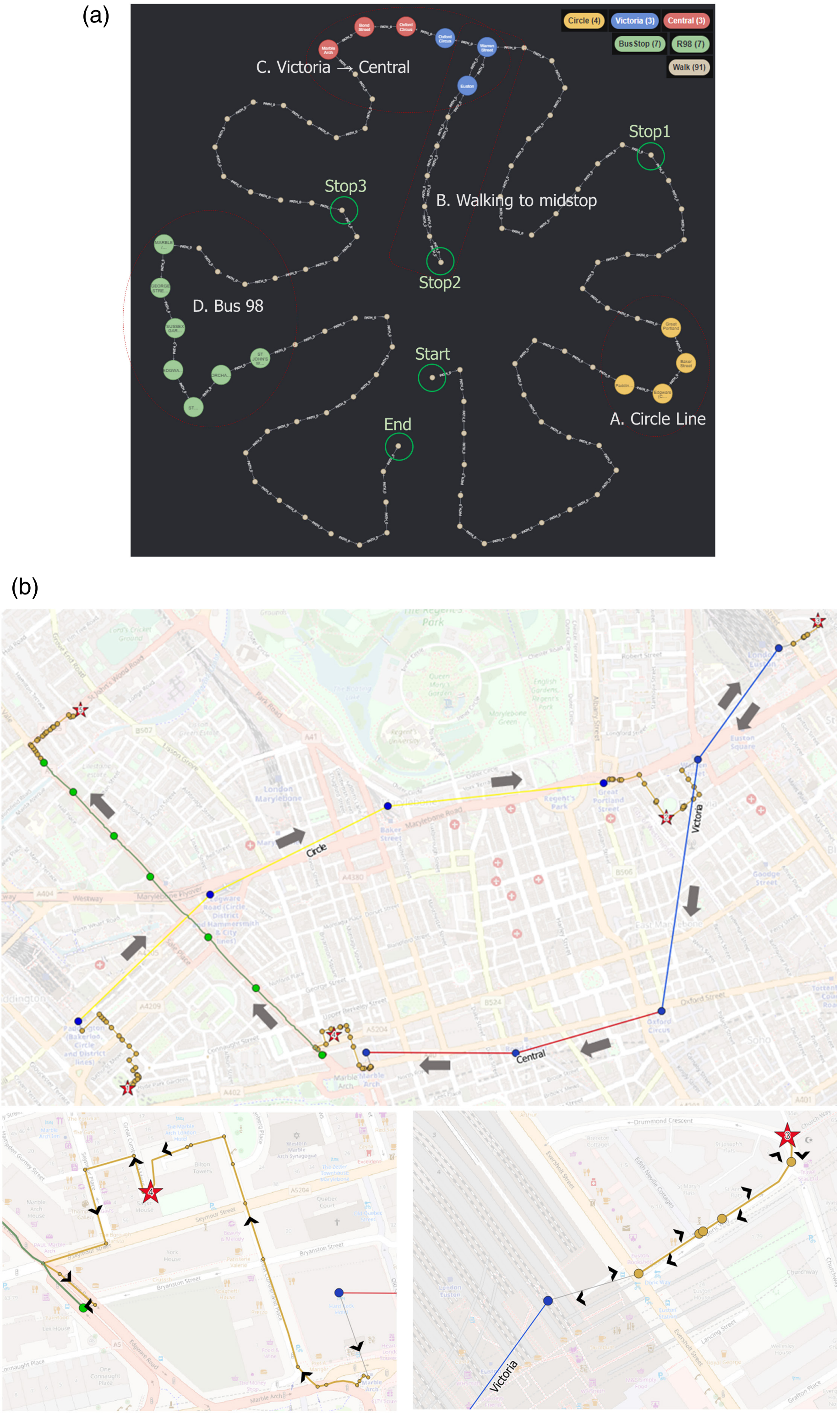


FIGURE 6 Results of multi-stop routing: (a) visualized using graph database; and (b) visualized using transport networks.

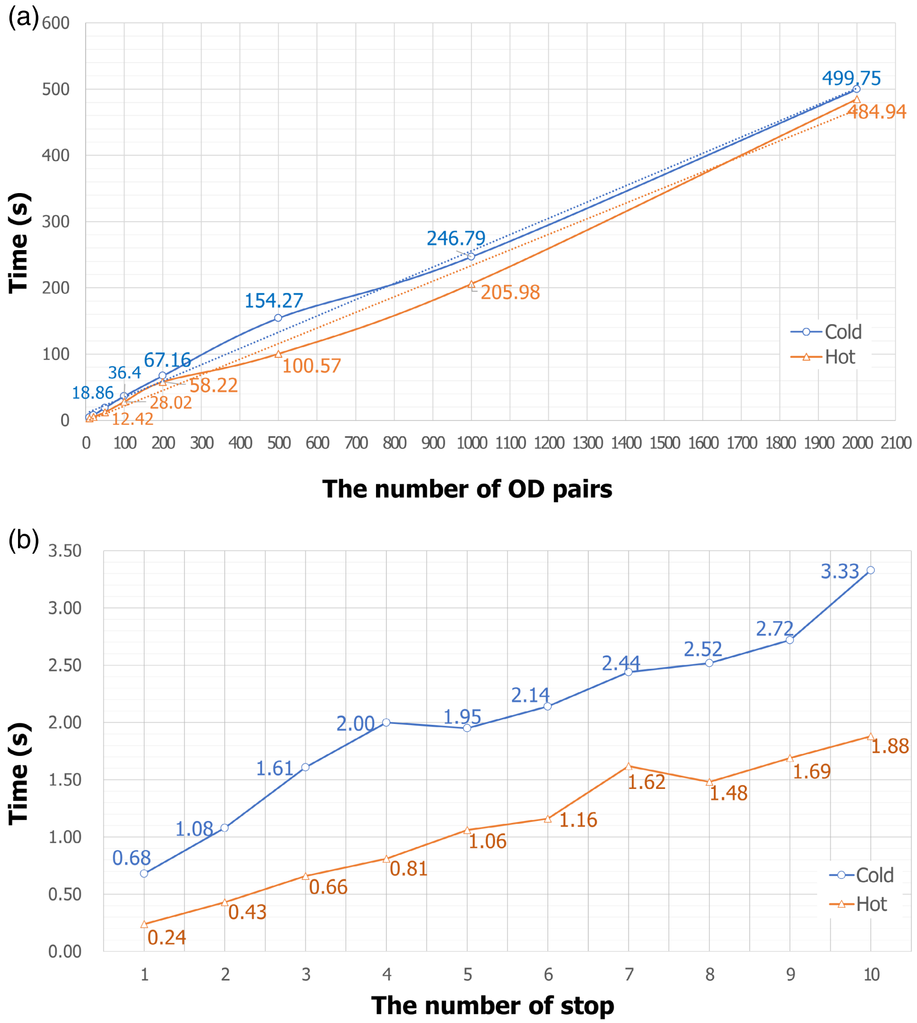


FIGURE 7 Graph query times according to: (a) the number of OD pairs (R^2 of the linear trendline for cold and hot are 0.9964 and 0.9923, respectively); and (b) the number of stops.

Furthermore, as an example of additional information integration, fare information was integrated into proposed multimodal graphs to present fares for the returned route. This example pertains to data integration for retrieving additional information on the returned graphs. Fare information for travel via rail can be provided along with the route through the PEAKFARE- and OFFPEAKFARE-relationships between *Station* and *Zone* nodes (Figure 9). Figures 9a,b show peak and off-peak fares for travel via tube from “Finchley Central” station in zone 4, to “Marble arch” in zone 1, respectively.

Based on this data integration, more complex routing (e.g., retrieving the least expensive route) can be achieved through multi-criteria decision-making: users can determine the optimal route by combining the time cost for each transport mode and retrieved fare for queried route; also, time-dependent routing can be efficiently performed by projecting only subgraphs that satisfy temporal conditions, such as indicated departure/arrival times using a combined time layer. Consequently, by utilizing a graph database, information on contextual information related to travel, as well as optimal routes can be easily combined, and they can be efficiently updated by adding, changing, and removing only those relevant relationships.

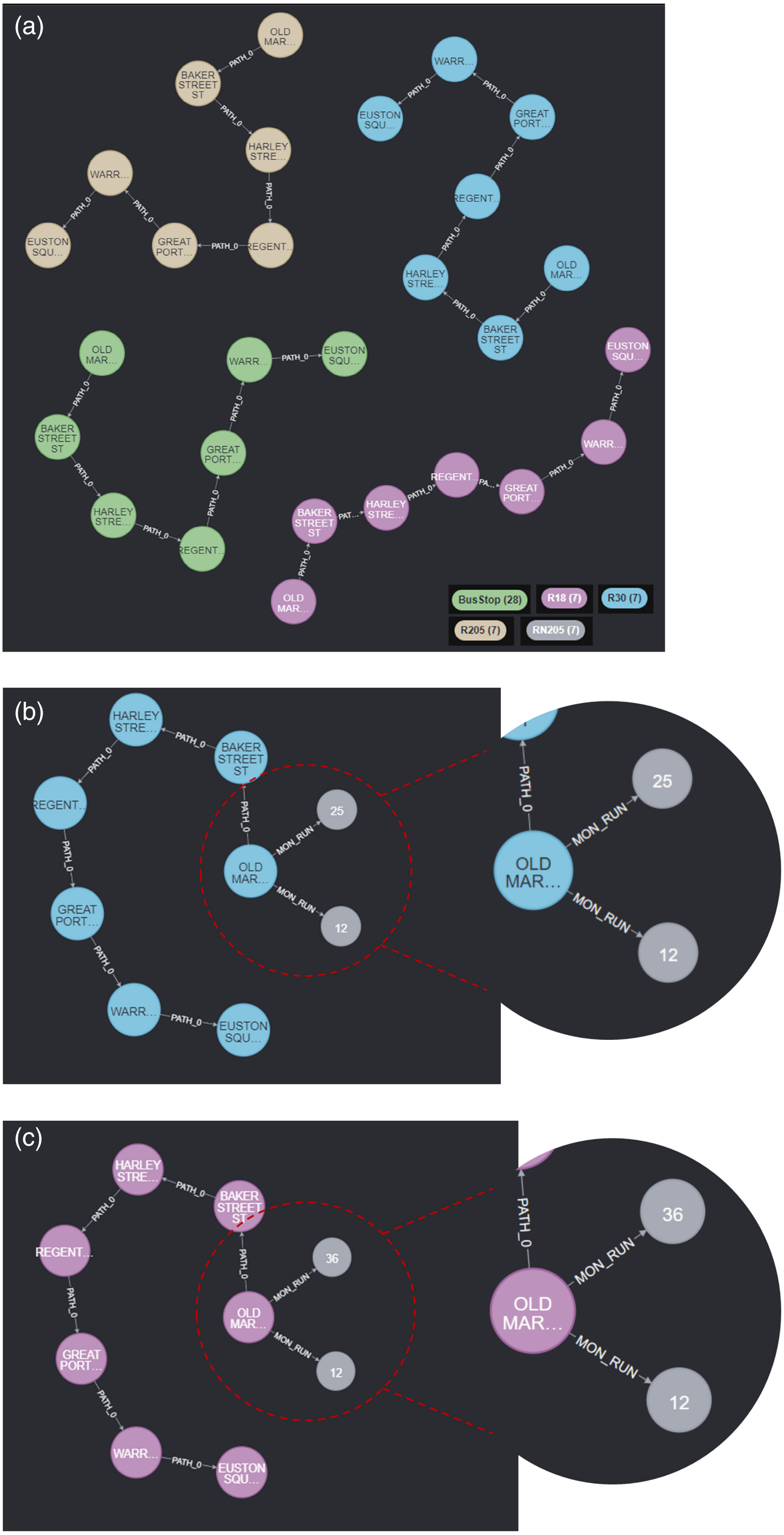


FIGURE 8 Query results considering bus timetable: (a) all available bus routes; (b) for the departure before 12:30 pm on Monday; and (c) for the departure after 12:30 pm on Monday.

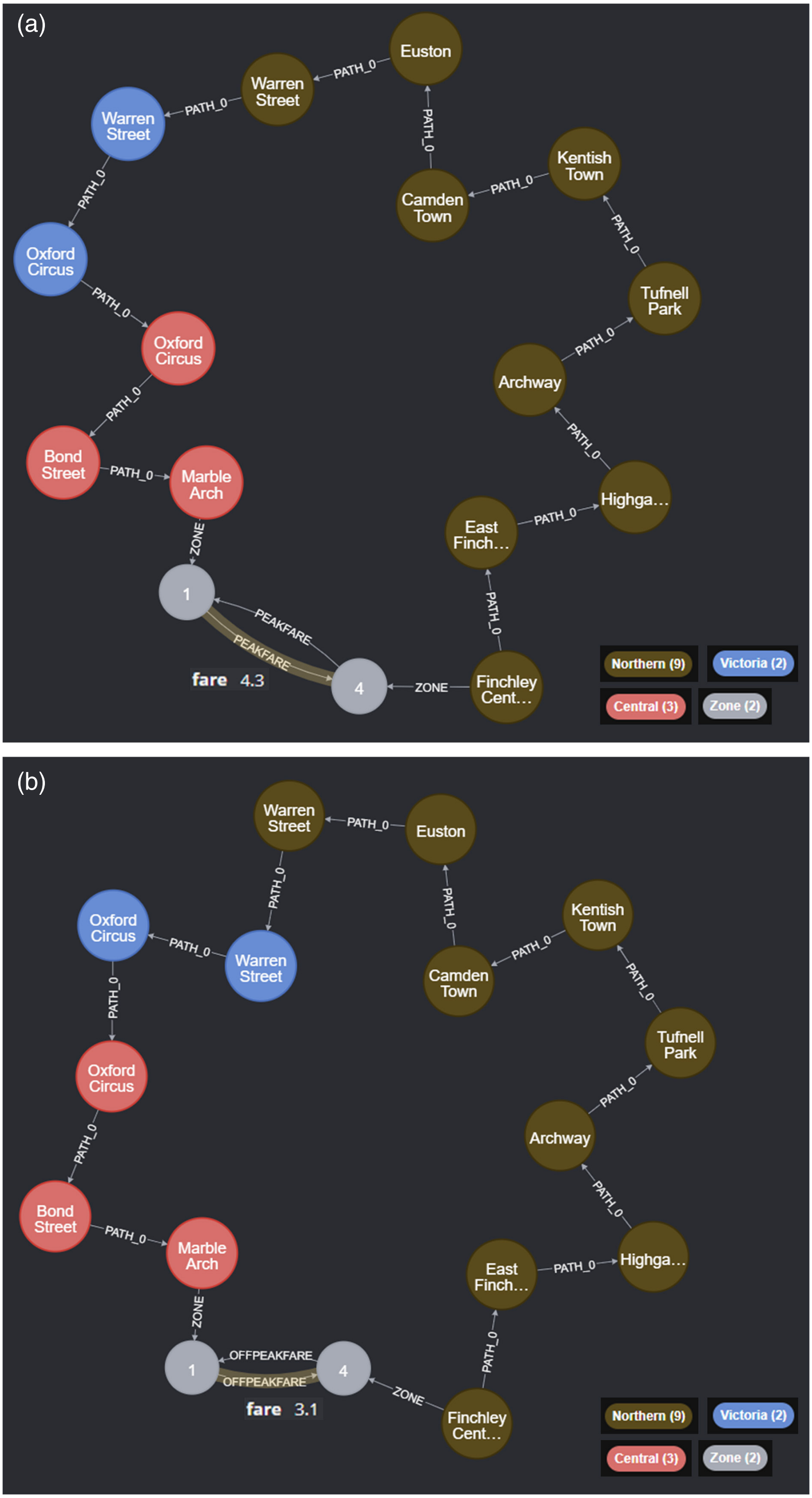


FIGURE 9 Query results with fare information in (a) *PEAKFARE*- and (b) *OFFPEAKFARE*- relationships.

5 | CONCLUSIONS AND FUTURE WORK

This study proposed a framework for a multimodal graph database considering public and private transit, including driving, walking, cycling, bus, and rail. Based on the proposed conceptual model, the graph database can be efficiently constructed using prebuilt transport data. To illustrate the efficiency and usefulness of the proposed model, a case study in London was conducted to test performance of the multimodal routing under various conditions. The results showed the optimal routes with mixed modes were appropriately identified with stable performance for iterative query processing. Its efficient use in multi-stop routing was confirmed as well. Compared to previous transport networks with a traditional relational database for multimodal routing applications, a graph database is suitable for managing multimodal networks more effectively regarding data integration, expansion, and updates, based on a flexible schema. To evolve existing routing services to personalized and context-aware routing, it is necessary to establish a database combining different typed contextual data with spatial data. In response to this demand, the proposed framework can be applied to construct fundamental multimodal networks suitable for integrating and expanding data using a graph database.

Regarding the future research directions, three aspects were considered in response to high-level routing queries, such as personalized routing.

5.1 | Integration with semantic information from other data sources

Here, a multimodal graph database was proposed focusing on transport routes by multiple modes, including some additional information, such as timetables and fare information. Based on the multilayered structure, and the advantages of the graph database, it is expected that the proposed model enables routing that reflects preferences by integrating various unstructured information into an additional layer. For example, information on temporarily blocked roads or non-operational bus and rail routes can be extracted from text data, such as SNS or articles, and subsequently added to the database. Also, different traffic flows by transport links can be added as a property of the relationship, which can be used as weights for routing.

5.2 | Setting of precise values as cost properties

Several assumptions were made in the case study with regards to costs, and the results of routing queries can be improved by setting precise cost values. Each road has a different speed limit, and the average moving speed varies with traffic. A more accurate travel time can thus be calculated by applying this speed variation. In addition, if different line transfer times in each tube station were input, more practical multimodal routes could be derived.

5.3 | Extension of various routing

The proposed multimodal graph database can achieve various routing applications such as seamless routing from outdoor to indoor, time-dependent routing, and personalized routing with semantic contexts. A seamless database connecting indoors to outdoors could be constructed by linking the proposed multimodal graph database to indoor graphs with several relationships between anchor nodes. Also, this multimodal graph database can be combined with layers with semantic contexts of travel environments to build a spatial-semantic integrated graph database in routing problems that enables high-level routing queries.

ACKNOWLEDGMENTS

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2021R1A6A3A01086427), and UK Economic and Social Research Council (ES/L011840/1).

CONFLICT OF INTEREST STATEMENT

The authors declare that there is no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Seula Park https://orcid.org/0000-0002-6059-4731

REFERENCES

Ali, E. (2020). *Geographic information system (GIS): Definition, development, applications & components*. Department of Geography, Ananda Chandra College.

Angles, R., & Gutierrez, C. (2008). Survey of graph database models. *ACM Computing Surveys*, *40*(1), 1–39. https://doi.org/10.1145/1322432.1322433

Angles, R., & Gutierrez, C. (2018). An introduction to graph data management. In G. Fletcher, J. Hidders, & J. Larriba-Pey (Eds.), *Graph data management: Data-centric systems and applications* (pp. 1–32). Springer. https://doi.org/10.1007/978-3-319-96193-4_1

Batra, S., & Tyagi, C. (2012). Comparative analysis of relational and graph databases. *International Journal of Soft Computing and Engineering*, *2*(2), 509–512.

Bellocchi, L., Latora, V., & Geroliminis, N. (2021). Dynamical efficiency for multimodal time-varying transportation networks. *Scientific Reports*, *11*(1), 1–14. https://doi.org/10.1038/s41598-021-02418-5

Berger, A., Delling, D., Gebhardt, A., & Müller-Hannemann, M. (2009). Accelerating time-dependent multi-criteria timetable information is harder than expected. *9th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems*, Copenhagen, Denmark (pp. 1–21). https://doi.org/10.4230/OASIcs.ATMOS.2009.2148

Bonnetain, L., Furno, A., Krug, J., & Faouzi, N. E. E. (2019). Can we map-match individual cellular network signaling trajectories in urban environments? Data-driven study. *Transportation Research Record*, *2673*(7), 74–88. https://doi.org/10.1177/0361198119847472

Chen, J., Song, Q., Zhao, C., & Li, Z. (2020). Graph database and relational database performance comparison on a transportation network. *Communications in Computer and Information Science*, *1244*, 407–418. https://doi.org/10.1007/978-981-15-6634-9_37

Chondrogiannis, T., Gamper, J., Cavaliere, R., & Ohnewein, P. (2016). MoTrIS: A framework for route planning on multimodal transportation networks. *24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Burlingame, CA (pp. 1–4). https://doi.org/10.1145/2996913.2997007

Chu, Z., Yu, J., & Hamdulla, A. (2020). A novel deep learning method for query task execution time prediction in graph database. *Future Generation Computer Systems*, *112*, 534–548. https://doi.org/10.1016/j.future.2020.06.006

Czerepicki, A. (2016). Application of graph databases for transport purposes. *Bulletin of the Polish Academy of Sciences. Technical Sciences*, *64*(3), 1–10. https://doi.org/10.1515/bpasts-2016-0051

De Virgilio, R., Maccioni, A., & Torlone, R. (2014). Model-driven design of graph databases. In E. Yu, G. Dobbie, M. Jarke, & S. Purao (Eds.), *Conceptual modeling: ER 2014* (pp. 172–185). Springer. https://doi.org/10.1007/978-3-319-12206-9_14

Debrouvier, A., Parodi, E., Perazzo, M., Soliani, V., & Vaisman, A. (2021). A model and query language for temporal graph databases. *The VLDB Journal*, *30*(5), 825–858. https://doi.org/10.1007/s00778-021-00675-4

Ding, R., Ujang, N., Bin Hamid, H., Abd Manan, M. S., He, Y., Li, R., & Wu, J. (2018). Detecting the urban traffic network structure dynamics through the growth and analysis of multi-layer networks. *Physica A: Statistical Mechanics and its Applications*, *503*, 800–817. https://doi.org/10.1016/j.physa.2018.02.059

Faye, S., Cantelmo, G., Tahirou, I., Derrmann, T., Viti, F., & Engel, T. (2017). Demo: MAMBA: A platform for personalised multimodal trip planning. *IEEE Vehicular Networking Conference*, Torino, Italy (pp. 117–118). https://doi.org/10.1109/vnc.2017.8275611

Fortin, P., Morency, C., & Trépanier, M. (2016). Innovative GTFS data application for transit network analysis using a graph-oriented method. *Journal of Public Transportation*, *19*(4), 18–37. https://doi.org/10.5038/2375-0901.19.4.2

Giannakopoulou, K., Paraskevopoulos, A., & Zaroliagis, C. (2019). Multimodal dynamic journey-planning. *Algorithms*, *12*(10), 213. https://doi.org/10.3390/a12100213

Gil, J. (2014). Analyzing the configuration of multimodal urban networks. *Geographical Analysis*, *46*(4), 368–391. https://doi.org/10.1111/gean.12062

Hrncir, J., & Jakob, M. (2013). Generalised time-dependent graphs for fully multimodal journey planning. *16th International IEEE Conference on Intelligent Transportation Systems*, The Hague, Netherlands (pp. 2138–2145). https://doi.org/10.1109/itsc.2013.6728545

Hu, Y., Xu, M., Tang, M., Han, D., & Liu, Y. (2021). Efficient traffic-aware routing strategy on multilayer networks. *Communications in Nonlinear Science and Numerical Simulation*, *98*, 105758. https://doi.org/10.1016/j.cnsns.2021.105758

Huang, H., Bucher, D., Kissling, J., Weibel, R., & Raubal, M. (2018). Multimodal route planning with public transport and carpooling. *IEEE Transactions on Intelligent Transportation Systems*, *20*(9), 3513–3525. https://doi.org/10.1109/TITS.2018.2876570

Ibrahim, A. A., Leite, D., & De Bacco, C. (2022). Sustainable optimal transport in multilayer networks. *Physical Review E*, *105*(6), 064302. https://doi.org/10.1103/PhysRevE.105.064302

Ibrahim, A. A., Lonardi, A., & Bacco, C. D. (2021). Optimal transport in multilayer networks for traffic flow optimization. *Algorithms*, *14*(7), 189. https://doi.org/10.3390/a14070189

Idri, A., Oukarfi, M., Boulmakoul, A., Zeitouni, K., & Masri, A. (2017). A new time-dependent shortest path algorithm for multimodal transportation network. *Procedia Computer Science*, *109*, 692–697. https://doi.org/10.1016/j.procs.2017.05.379

Jaiswal, G., & Agrawal, A. P. (2013). Comparative analysis of relational and graph databases. *IOSR Journal of Engineering*, *3*(8), 25–27. https://doi.org/10.9790/3021-03822527

Jamal, J., Montemanni, R., Huber, D., Derboni, M., & Rizzoli, A. E. (2017). A multi-modal and multi-objective journey planner for integrating carpooling and public transport. *Journal of Traffic and Logistics Engineering*, *5*(2), 68–72. https://doi.org/10.18178/jtle.5.2.68-72

Jing, X. L., Hu, M. B., Shi, C. L., & Ling, X. (2022). An efficient routing strategy for coupled spatial networks. *Modern Physics Letters B*, *36*(5), 2150584. https://doi.org/10.1142/S0217984921505849

Liu, Y., Qu, S., & Fan, B. (2021). Current status and application analysis of graph database technology. *3rd International Conference on Machine Learning, Big Data and Business Intelligence*, Taiyuan, China (pp. 735–744).

Ma, T. Y., & Lebacque, J. P. (2013). Dynamic system optimal routing in multimodal transit network. *Transportation Research Record*, *2351*(1), 76–84. https://doi.org/10.3141/2351-09

Maduako, I., Cavalheri, E., & Wachowicz, M. (2018). Exploring the use of time-varying graphs for modelling transit networks. https://arxiv.org/abs/1803.07610

Maduako, I. D., Wachowicz, M., & Hanson, T. (2019). Transit performance assessment based on graph analytics. *Transportmetrica A: Transport Science*, *15*(2), 1382–1401. https://doi.org/10.1080/23249935.2019.1596991

Medhi, S., & Baruah, H. K. (2017). Relational database and graph database: A comparative analysis. *Journal of Process Management and New Technologies*, *5*(2), 1–9. https://doi.org/10.5937/jouproman5-13553

Miler, M., Medak, D., & Odobašić, D. (2014). The shortest path algorithm performance comparison in graph and relational database on a transportation network. *Promet-Traffic & Transportation*, *26*(1), 75–82. https://doi.org/10.7307/ptt.v26i1.1268

Mohamed, O., & Appling, H. (2020). *Clinical assessment of gait*. *Orthotics and prosthetics in rehabilitation* (4th ed.). Elsevier. https://doi.org/10.1016/B978-0-323-60913-5.00005-2

Natera, L., Battiston, F., Iñiguez, G., & Szell, M. (2020). Extracting the multimodal fingerprint of urban transportation networks. *arXiv:2006.03435*. https://doi.org/10.32866/001c.13171

Neo4j. (n.d.). Retrieved September 10, 2022, from https://neo4j.com/

Neo4j Graph Data Science Library Manual v1.8. (n.d.) Retrieved September 10, 2022, from https://neo4j.com/docs/graph-datascience/current/

Orozco, L. G. N., Alessandretti, L., Saberi, M., Szell, M., & Battiston, F. (2021). Multimodal urban mobility and multilayer transport networks. *arXiv:2111.02152*. https://doi.org/10.48550/arXiv.2111.02152

Potthoff, M., & Sauer, J. (2022). Fast multimodal journey planning for three criteria. *SIAM Symposium on Algorithm Engineering and Experiments* (pp. 1–15). https://doi.org/10.1137/1.9781611977042.12

Rishe, N., Vaschillo, A., Vasilevsky, D., Shaposhnikov, A., & Chen, S. C. (2000). A benchmarking technique for DBMS's with advanced data models. *ACM SIGMOD ADBIS-DASFAA Symposium on Advances in Databases and Information Systems*, Prague, Czech Republic (pp. 138–149).

Robinson, I., Webber, J., & Eifrem, E. (2015). *Graph databases: New opportunities for connected data* (2nd ed.). O'Reilly Media, Inc.

Shibanova, D. A., Stroganov, I. V., & Rudakov, I. V. (2021). Data formalization in transport system modeling using a graph database. *IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering*, St. Petersburg and Moscow, Russia (pp. 2245–2251). https://doi.org/10.1109/elconrus51938.2021.9396137

Smarzaro, R., Davis, C. A., Jr., & Quintanilha, J. A. (2021). Creation of a multimodal urban transportation network through spatial data integration from authoritative and crowdsourced data. *ISPRS International Journal of Geo-Information*, *10*(7), 470. https://doi.org/10.3390/ijgi10070470

Talasila, P., Asifullah, S., Goveas, N., & Deshpande, B. (2018). Transit timetables as multi-layer networks. *10th International Conference on Communication Systems & Networks*, Bengaluru, India (pp. 625–630). https://doi.org/10.1109/comsnets.2018.8328285

Tischner, D. (2018). Multi-modal route planning in road and transit networks. *arXiv:1809.05481*. https://doi.org/10.48550/arXiv.1809.05481

Varga, L., Kovács, A., Tóth, G., Papp, I., & Néda, Z. (2016). Further we travel the faster we go. *PLoS ONE*, *11*(2), e0148913. https://doi.org/10.1371/journal.pone.0148913

Wang, X., Zou, L., Wang, C., Peng, P., & Feng, Z. (2019). A review of knowledge graph data management research. *Journal of Software*, *30*(7), 2139–2174.

Waters, R. L., Lunsford, B. R., Perry, J., & Byrd, R. (1988). Energy-speed relationship of walking: Standard tables. *Journal of Orthopaedic Research*, *6*(2), 215–222. https://doi.org/10.1002/jor.1100060208

Wirawan, P. W., Riyanto, D. E., Nugraheni, D. M. K., & Yasmin, Y. (2019). Graph database schema for multimodal transportation in Semarang. *Journal of Information Systems Engineering and Business Intelligence*, *5*(2), 163–170. https://doi.org/10.20473/jisebi.5.2.163-170

Zhuravleva, N. A., & Poliak, M. (2020). Architecture of managing big data of mixed transportation of passengers in aglomerations. *IOP Conference Series: Materials Science and Engineering*, *918*, 012055. https://doi.org/10.1088/1757-899X/918/1/012055

How to cite this article: Park, S., & Cheng, T. (2023). Framework for constructing multimodal transport networks and routing using a graph database: A case study in London. *Transactions in GIS*, *00*, 1–27. https://doi.org/10.1111/tgis.13071