A Graph theoretic approach for spatial connectivity analysis of Building Information Models

Department of Architecture | RWTH Aachen University

Design Computation | CAAD

by:
Silvia Vangelova

Supervisor:
Univ.-Prof. Dr. Jakob Beetz

Presentation Date: 01.03.2022

Contents:

1. Summary

The aim of this research project is to develop an efficient framework for analysing spatial configuration topology from Building Information Models (BIMs) by representing relations between primitive building elements as graphs. The main objective is finding and analysing room adjacency in floor plans. To achieve this, a methodological approach for extracting wall connectivity data from IFC files, which are an industry standard for storing BIMs, is proposed. The obtained information, such as wall axis geometry, is usually present in all IFC building models and thus no custom Model View Definition (MVD) is required. Because of this, spatial relations can also be retrieved from building projects that do not meet specific Exchange Requirements for entities and properties from the IFC Schema such as spaces and spatial boundaries. This functionality can be utilized for various design goals. For example, inferring relevant information for energy analysis, e.g., specifying which walls are external and which internal, emergency path planning, automatic room type categorisation and many others.

Keywords: Graph Representation, Graph Theory, Building Information Modelling (BIM), Industry Foundation Classes (IFC)

2. Abstract

The representation of room connectivity in building floor plans is the main goal of this project. By reading IFC files, information about wall-to-wall connectivity and wall axis geometry on each building story can be extracted. This information is used to represent wall adjacency as a graph, where each node represents a wall segment and each edge represent a connection between two wall segments. By finding the geometric dual of this graph the rooms on each floor and the connection type between them can be identified. As a result, spatial relations in a floor plan are transformed into a machine-readable format. This capability may be used to "fill the gaps" in the case of insufficient amount of data about rooms and spaces in a building model. By further processing this information, walls in a building model can be categorized as either external or internal. Furthermore, room types can be classified based on centrality measures on the graph, such as degree centrality, e.g., the number of walls containing a door that surround a room gives the number of connections a room has. The precise knowledge of how rooms are connected can be used to reason about emergency path planning and mobility access.

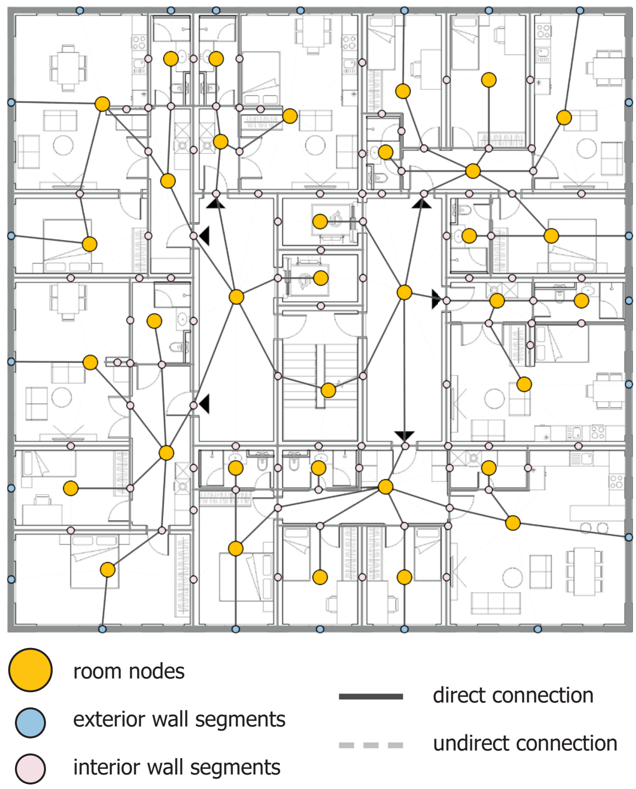


Fig. 1. Illustration of room connectivity in a floor plan by representing access relationships between rooms and the external environment

3. Research context and related work

3.1. Applications of graph-based representation in the AEC industry

Graph-based representations of data contained in building projects are used for various purposes. In general, graph models in the AEC industry can be divided into four categories: Space connectivity, Navigation, IFC model and Knowledge representation graphs [1]. The proposed framework in this research project implements a type of a navigation graph model (Fig. 1), in which nodes represent spaces or wall segments in the building, and links between the nodes represent the different relationships between them , for instance adjacency between a wall segment and a room. By applying this method, the topology of only a specific part of a building project can be extracted and analysed. This dimensionality reduction greatly facilitates the integration and adjustment of the obtained information [2].

One of the most widely researched problems is the analysis of room connectivity in buildings and finding escape routes in emergency situations [3]. In order to find topological connectivity of spaces, relations between IFC entities can be queried. Thus, a spatial configuration can be extracted and stored as a "semantic fingerprint" in a graph structure [4].

3.2. Methods for abstracting a floor plan into a graph

It has been observed that the class of graphs that correspond to realizable floor plans consists of planar graphs [5]. In this project the assumption is made that wall connectivity in a floor plan can be embedded in a planar graph by segmenting walls (Fig. 2 and Fig. 3).

Rather than constructing the graph corresponding to the functional zoning of a floor plan, another approach is to compute the geometric dual of the floor-plan graph (Fig. 4a). In addition to describing wall junctions, the dual directly defines spaces and their adjacency [6]. A more detailed definition of a geometric dual of a graph is given in section 4.3.1 Definition 2.

There are multiple ways to compute the dual of a graph. The approach used in this project is finding the minimum weight cycle basis of the wall segment connectivity graph. A more detailed explanation of what a minimum cycle basis is can be found in 4.3.1. Definition 7. This basis essentially gives information about the nodes surrounding each face, or equivalently, the nodes contained in each cell of the graph. The algorithm used to find the minimum weight cycle basis [7] returns a list of indices which correspond to the segments surrounding each face of the graph. The lists containing exactly 3 indices correspond to a "T" wall junction, and those containing more than 3 correspond to rooms or spaces in the floor plan.

By adding a node in each cell that corresponds to a room and connecting it to its neighboring wall segment nodes room adjacency can be obtained (Fig. 4b). Furthermore, by considering only walls that are connected at both ends, or equivalently, removing all wall segment nodes that have a degree (number of connections) less than 2, it can be inferred which of the remaining walls are external or internal. If a wall segment is connected to only one room node, then the wall is external, otherwise it is internal. An access relationship between two rooms exists if the wall segment connected to both contains an opening or a door instance.

A great advantage of this method is that it does not rely on whether there are spatial elements defined in the IFC file. Room connectivity is found only based on wall intersections. By using the element hierarchy of the IFC Schema the type of connections between rooms can be specified further.

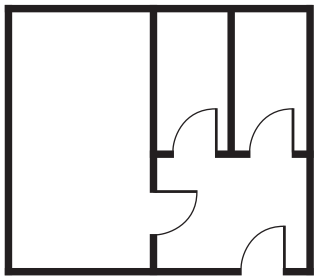


Fig. 2. Simplified representation of a floor plan

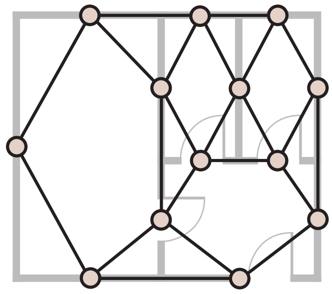


Fig. 3. Wall segment connectivity represented as a graph

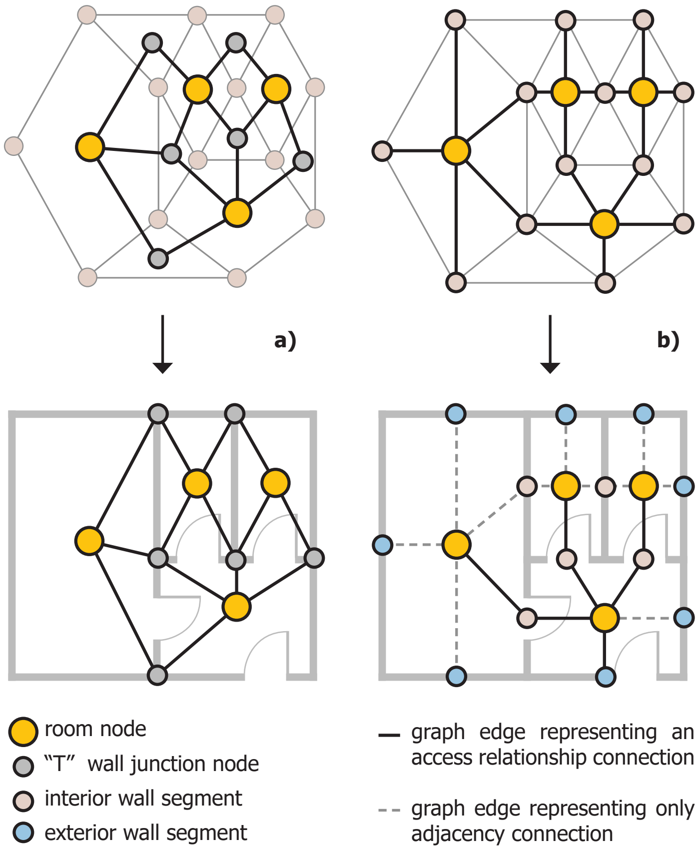


Fig. 4. Graph representation of room and wall adjacency: **a)** the dual also describes wall junctions; **b)** connectivity types between rooms;

4. Project concept and methodology

The main aspect of this research project is to explore an alternative way for extracting spatial topology from BIMs. Instead of using the most common method for finding space adjacency (described and illustrated in section 4.1), a different aproach is proposed.

The goal is to simplify and remove some of the limitations of the current state of room connectivity extraction methods, which rely primarily on the existence of space boundaries in the building project.

4.1. Extracting information from IFC files

Provided that space boundaries are specified in the Model View Definition of the IFC building model, information about rooms and spaces can be acquired. According to the buildingSMART International Standards [9], the entity definition of a space represents an area or a volume that is bounded either physically or virtually. The IfcSpace class describes the spatial structure of a building and also serves as the spatial container for space-related elements. Each IfcSpace has a space boundary that can be either physical or virtual and it is obtained by the relationship IfcRelSpaceBoundary (Fig. 5). The IfcRelSpaceBoundary describes the one-to-one relationship between an element and the space it bounds.

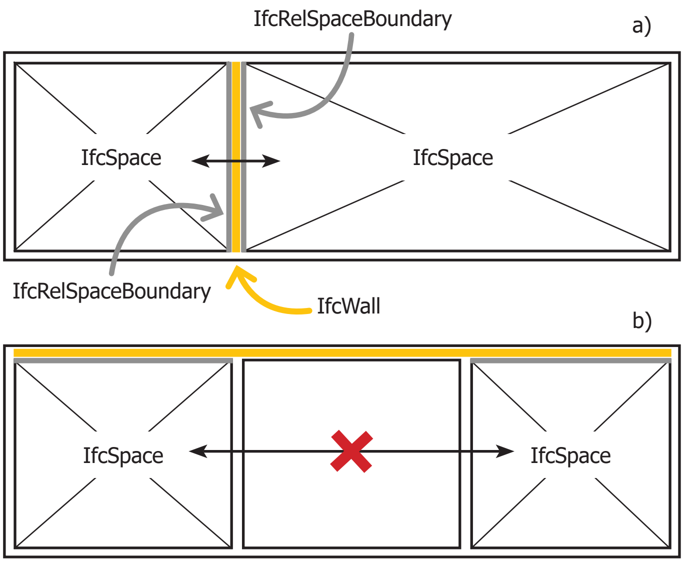


Fig. 6. Schematic represenation IfcSpace adjacency cases: **a)** two spaces are adjacent therefore they have a wall relation in common; **b)** two spaces have a wall relation in common but they are not adjacent.

It is evident that if two rooms on a given building storey are adjacent, meaning they are both separated by the same wall, then, by using the IfcRelSpaceBoundary relationship, their corresponding IfcSpace entities refer to the same IfcWall which is a sub-class of the IfcElement class (Fig. 6a). However, it is not true that two IfcSpace entities are adjacent if they refer to the same IfcWall (Fig. 6b). For this reason, instead of analysing the relationships of IfcRelSpaceBoundary entities the proposed method explores connectivity of IfcWall entities by using the IfcRelConnectsPathElements relationship (Fig. 7) Furthermore, analysing geometric wall connectivity by extracting wall axis geometry (Fig. 8) and transforming it according to its real project location (Fig. 9) gives additional information about wall intersections. To ensure that all wall axis geometry is correct, curves are processed by extracting IfcWall entity thickness (Fig. 10).

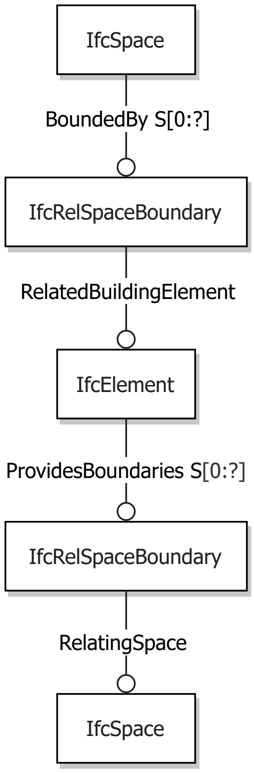


Fig. 5. Finding a relation between IfcSpace entities

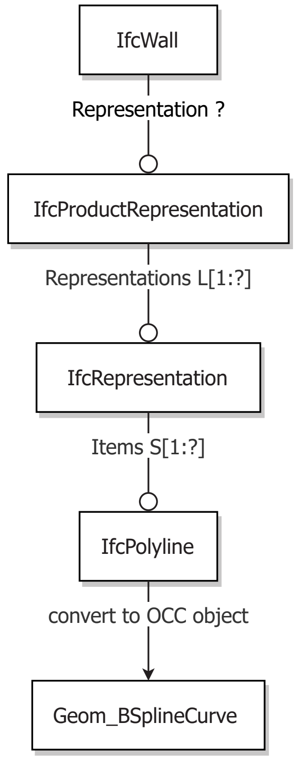


Fig. 8. Extracting and converting wall axis geometry

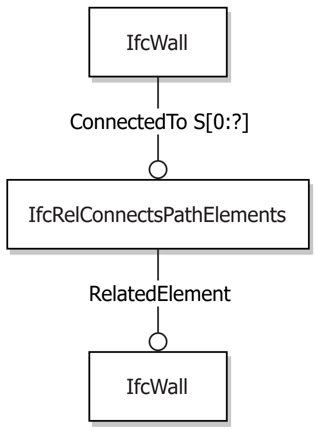


Fig. 7. Finding a relation between two IfcWall entities

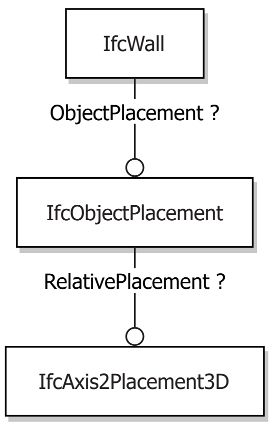


Fig. 9. Finding the transformation matrix of an IfcWall entity

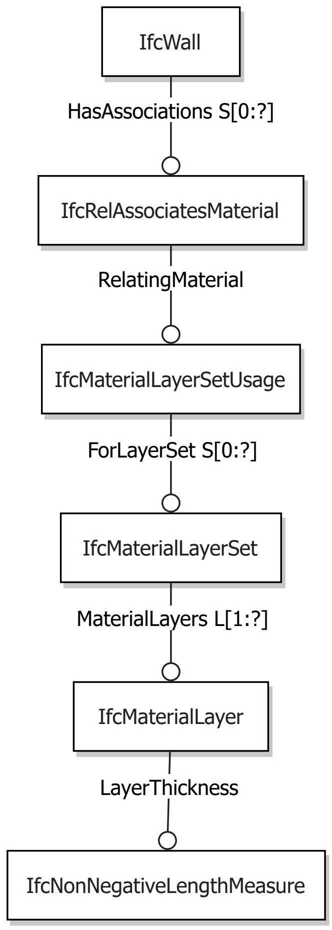


Fig. 10. Extracting wall thickness by finding material layer thickness

The IfcRelConnectsPathElements relationship also provides the connectivity information between two IfcWall elements (Fig. 11). However, this relationship is also reflected in the 2D shape representation of the wall axis (Fig. 12 a and b). By knowing the type of connection between each two wall instances and knowing their respective thickness, the axis curve geometry is "extended" so that all axes are connected (Fig. 12c).

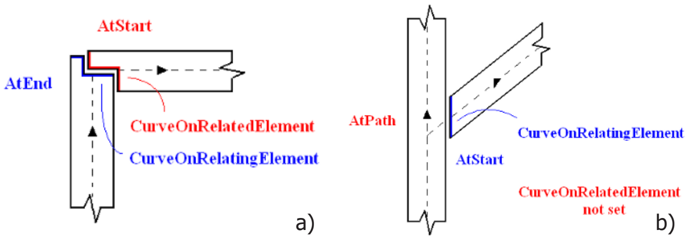


Fig. 11. Visualisation of wall connection geometry: **a)** IfcRelConnectsPathElements for a "T" type connection between two instances of IfcWallStandardCase; **b)** IfcRelConnectsPathElements for a "L" type connection between two instances of IfcWallStandardCase

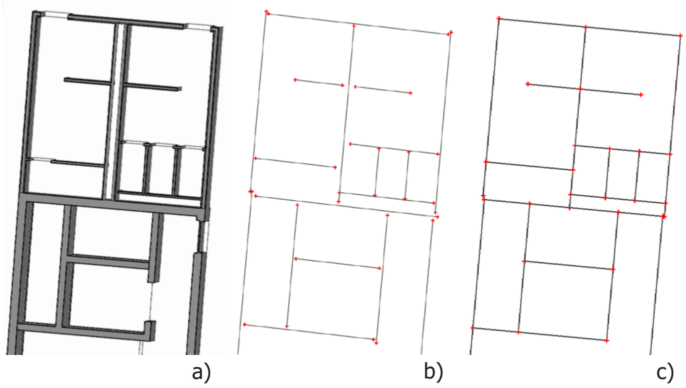


Fig. 12. Visualisation of wall axis geometry: **a)** 3D representation of the IfcWall entities; **b)** IfcWall axis geometry as defined in its respective Representation attribute; **c)** corrected wall axis geometry based on wall thickness and connection type

4.2. Object oriented approach - main classes and their relations

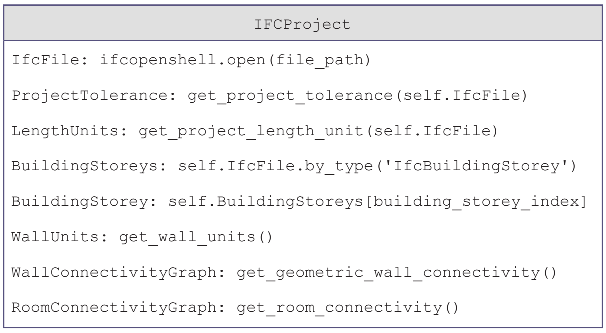


Fig. 13. Diagram of the IFCProject class which stores and processes relevant project information when initialized

All relevant project information is stored in an IFCProject class. (Fig. 13). When initialized, functions are called in a specific order and the graphs of wall and room connectivity are computed. After computations are complete the respective graph information is saved and can be compared with graphs extracted from other projects.

To easily store and modify information for each building element a class hierarchy is implemented (Fig. 14). Classes are named using CamelCase (capital letters at the start of each word). Functions, methods and variable names are lower_case_underscore (lowercase with an underscore representing a space between words).

Each wall and its relationships are represented in a WallUnit class. If a WallUnit is connected to another WallUnit its axis geometry gets segmented at the point of intersection. These segments are defined as a WallSegment class. A list of identifiers for each segment in a wall are stored in each WallUnit. WallSegments which belong to a room, or equivalently form a simple cycle with a length greater than 3 (a definition for a simple cycle is given in 4.3.1. Definition 5.), define a face in which a node corresponding to a room is inserted.

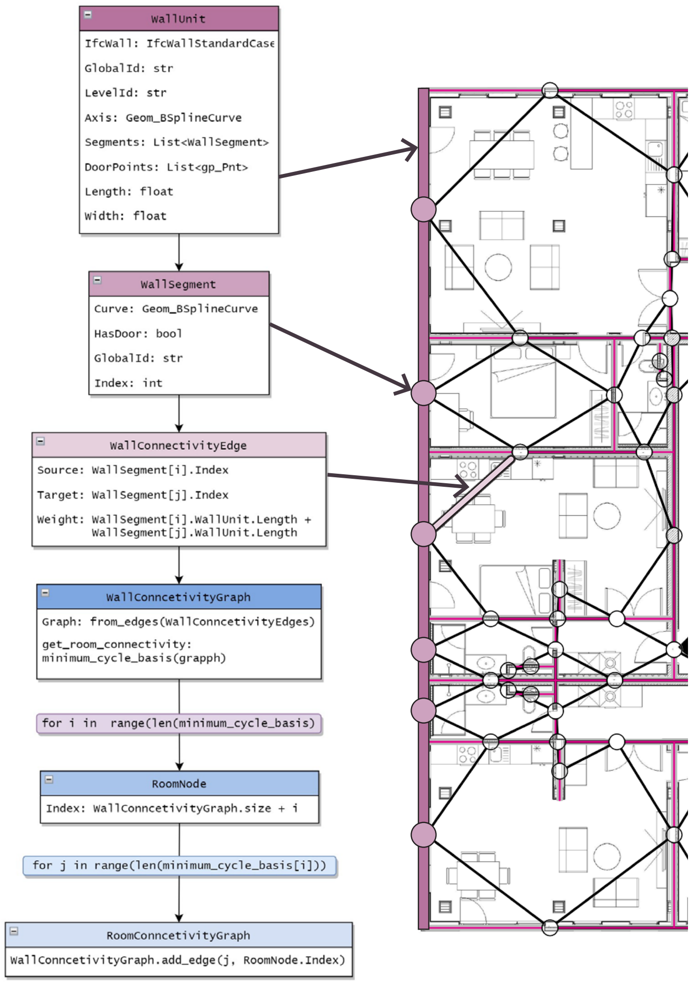


Fig. 14. Representation of the wall connectivity graph and the relations between the different classes

4.3. Graph representation of connectivity

4.3.1 Notations and definitions

This section provides a short clarification about fundamental definitions used in the algorithm for finding rooms.

Definition 1: A labelled weighted graph of order n is a collection of nodes together with a collection of edges that are pairs of nodes (Fig. 15). It is represented as $G = (V, E, L_v, L_e, W_e, l_v, l_e, w_e)$ where:

V is a set with n elements called *nodes*,
E is a set of ordered pairs of nodes called *edges*,
L_v is a set of labels for the nodes,
L_e is a set of labels for the edges,
W_e is a set of weights for the edges,
l_v is a function which assigns a label to the vertices,
l_e is a function which assigns a label to the edges,
w_e is a function which assigns a weight to the edges,

Definition 2: Given a planar graph G, its geometric dual G^* is constructed by placing a node in each region of G and, if two regions have an edge x in common, joining the corresponding vertices by an edge X^* crossing only x (Fig. 16).

Definition 3: A path on a graph is an ordered sequence of nodes such that any pair of consecutive nodes in the sequence is an edge of the graph. A path is simple if no node appears more than once in it, except for the case in which the initial node is the same as the final node (Fig. 17).

Definition 4: Connectivity and connected components. A graph is connected if there exists a path between any two nodes. If a graph is not connected, then it is composed of multiple connected components, that is, multiple connected subgraphs.

Definition 5: A cycle is a simple path that starts and ends at the same node and has at least three distinct nodes (Fig. 18).

Definition 6: A cycle space basis of a network is a minimal collection of cycles such that any cycle in the network can be written as a sum of cycles in the basis. (Fig. 19).

Definition 7: Minimal cycle basis means a cycle basis for which the total length of all the cycles is minimum. A cycle basis where the sum of the weights of the cycles is minimum is called a minimum weight cycle basis (Fig. 20).

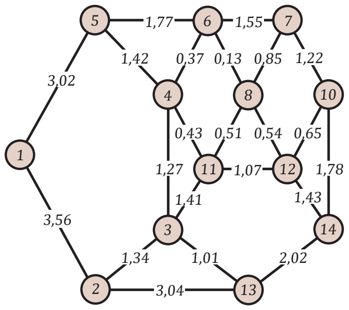


Fig. 15. A laballed edge-weighted planar graph of order $n = 14$

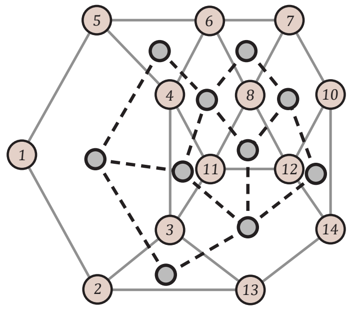


Fig. 16 The dual of the graph illustrated in Fig. 10

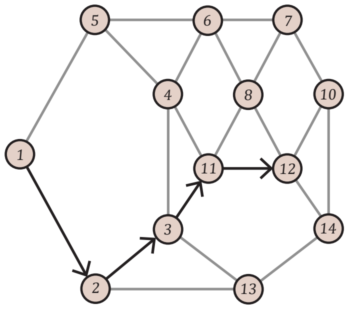


Fig. 17. Example of a simple path through nodes 1, 2, 3, 11, 12

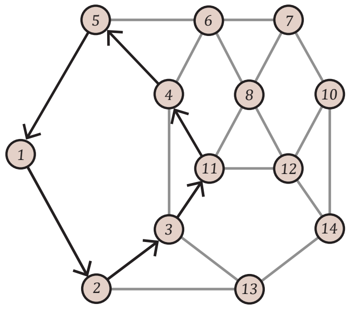


Fig. 18 Example of a simple cycle through nodes 1, 2, 3, 11, 4, 5

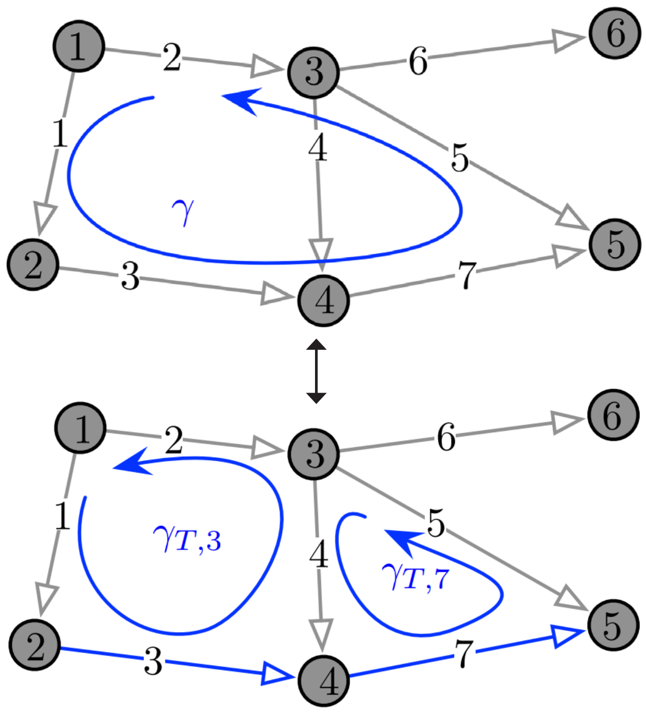


Fig. 19. Cycle Basis decomposition principle, Images: Francesco Bullo , Lectures on Network Systems [8]

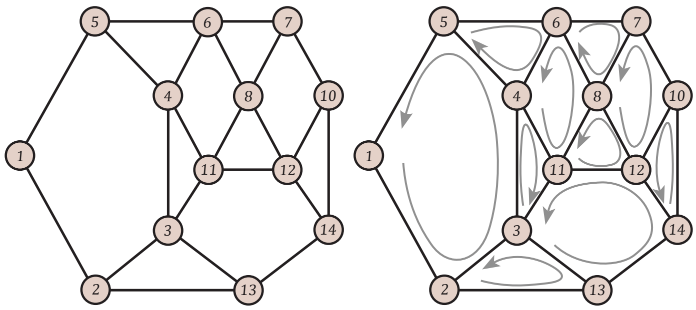


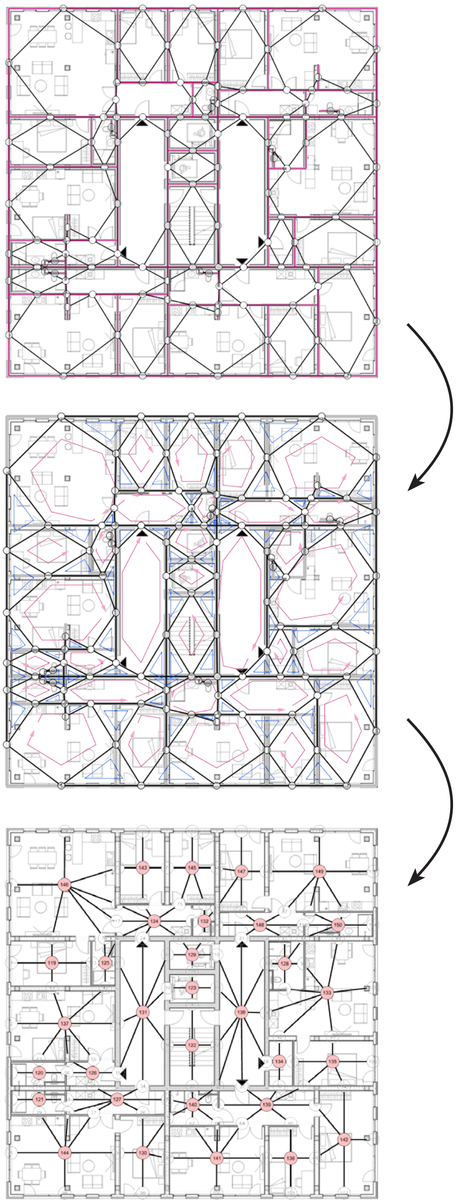
Fig. 20. Minimal Cycle Basis decomposition of the graph in Fig. 15

4.3.2. Graph construction

By intersecting all wall axes and segmenting them at their corresponding intersection points, wall segment connectivity is represented as an edge-weighted plane graph - WallConnectivityGraph. In this graph, each node corresponds to a WallSegment class. Each edge represents two adjacent wall segments and its weight equals the maximum of two values - either the distance between the middle points of those two segments or the average of the sum of their host wall lengths. Every node of the WallConnectivityGraph is mapped to a point corresponding to the middle point of its corresponding wall segment (Fig. 21a).

In order to find the rooms in the floor plan, the node indices forming each cell of the WallConnectivityGraph are found (Fig. 21b). This is achieved by computing the graphs minimal cycle basis where edge weights are rounded to integers to reduce computation time. In each of these cells who have more than 3 nodes a new node is placed. Each of these nodes corresponds to a room and is mapped to the average location of its surrounding nodes. This method essentially gives the RoomConnectivityGraph (Fig. 21c).

An access relationship exists if a WallSegment has an opening or a door to go from one room to the other. According to the IFC specification, IfcRelFillsElement and IfcRelVoidsElement are used in conjunction to recognize the relationship of an opening or a door to a wall, and then again to check the relationship of the wall to the adjacent rooms. However, in this project door containment in wall segments is considered rather than in wall instances.



a)

b)

c)

Fig. 21. Step-by-step construction of the room connectivity graph

5. Results and evaluation

The source code of the project is published on GitHub [10]. The framework is implemented on two types of projects. The first type consists of projects designed for the purpose of validating the project's algorithms and specifying requirements such as drafting quality and IFC Export Options. Such projects include both simple example cases for wall connections and entire residential buildings. The second type are building models from open-source data bases such as "Open IFC Model Repository" [11] and other exemplary IFC files [12]. While the framework is compatible with all current official versions of the IFC Schema [13], there are yet requirements for the model quality. There are also "corner cases" such as walls represented as B-spline curves with degree greater than 1. All projects considered contain only walls with straight line geometry.

5.1. Requirements

There are several base requirements for the IFC files that must be met:

1. All walls must be connected.
2. All walls must have axes that are either represented as straight lines or polylines.
3. All walls must have a valid 2D geometric representation.
4. The location line of each wall must correspond to its host building storey.

Core programming libraries implemented in the scripts include:

1. IfcOpenShell - for reading IFC files [14]
2. pythonOCC - for geometric operations [15]
3. NetworkX for graph creation and analysis [16]

5.2. Use cases

5.2.1. Projects designed for the purpose of algorithm validation

These are sample projects whose model quality is further edited so that the aforementioned requirements are met. The BIM authoring software used for the creation of the models and their export in the IFC file format is Autodesk Revit 2022.

5.2.1.1. A single apartment

This is a simple example (Fig. 22a) which validates wall axes geometry extraction (Fig. 22b) and correction (Fig. 22d). It also illustrates room connectivity (Fig. 22a).

5.2.1.2. A residential building

In this case the connectivity of rooms (Fig. 23a) and walls (Fig. 23b) in a residential building is extracted. The graphs are overlayed for visualisation purposes.

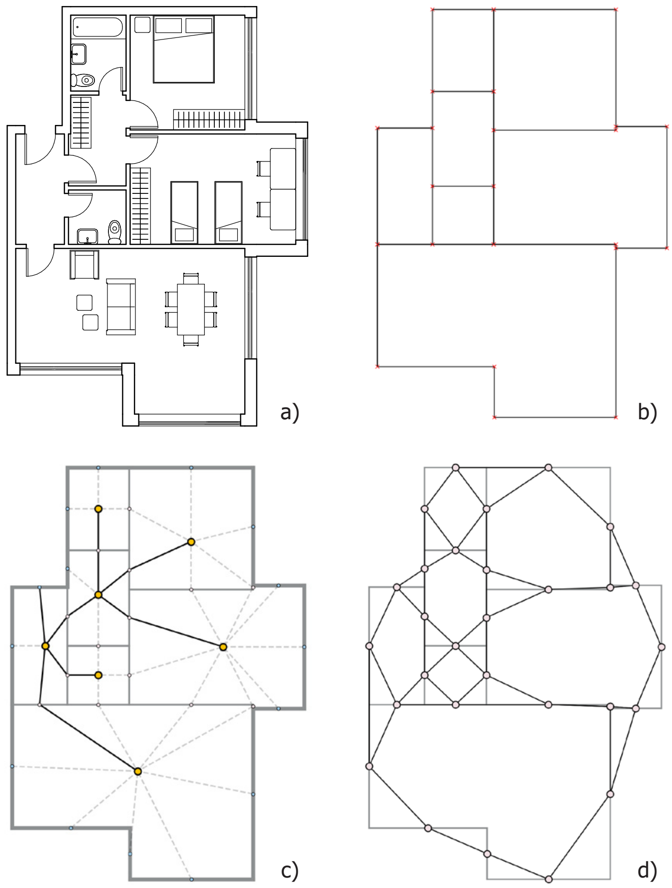


Fig. 22. Room and wall connectivity extraction from a simple floor plan

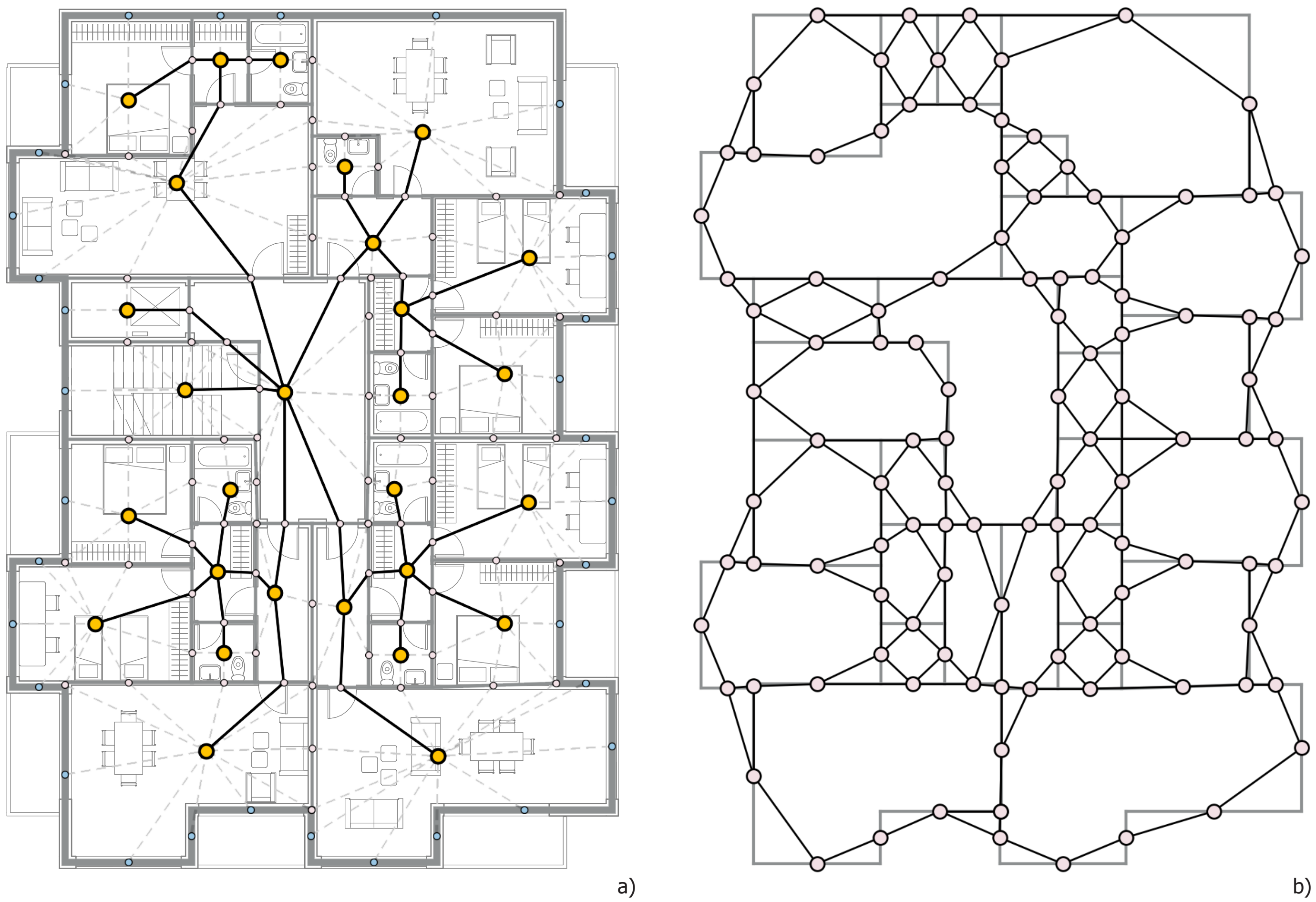
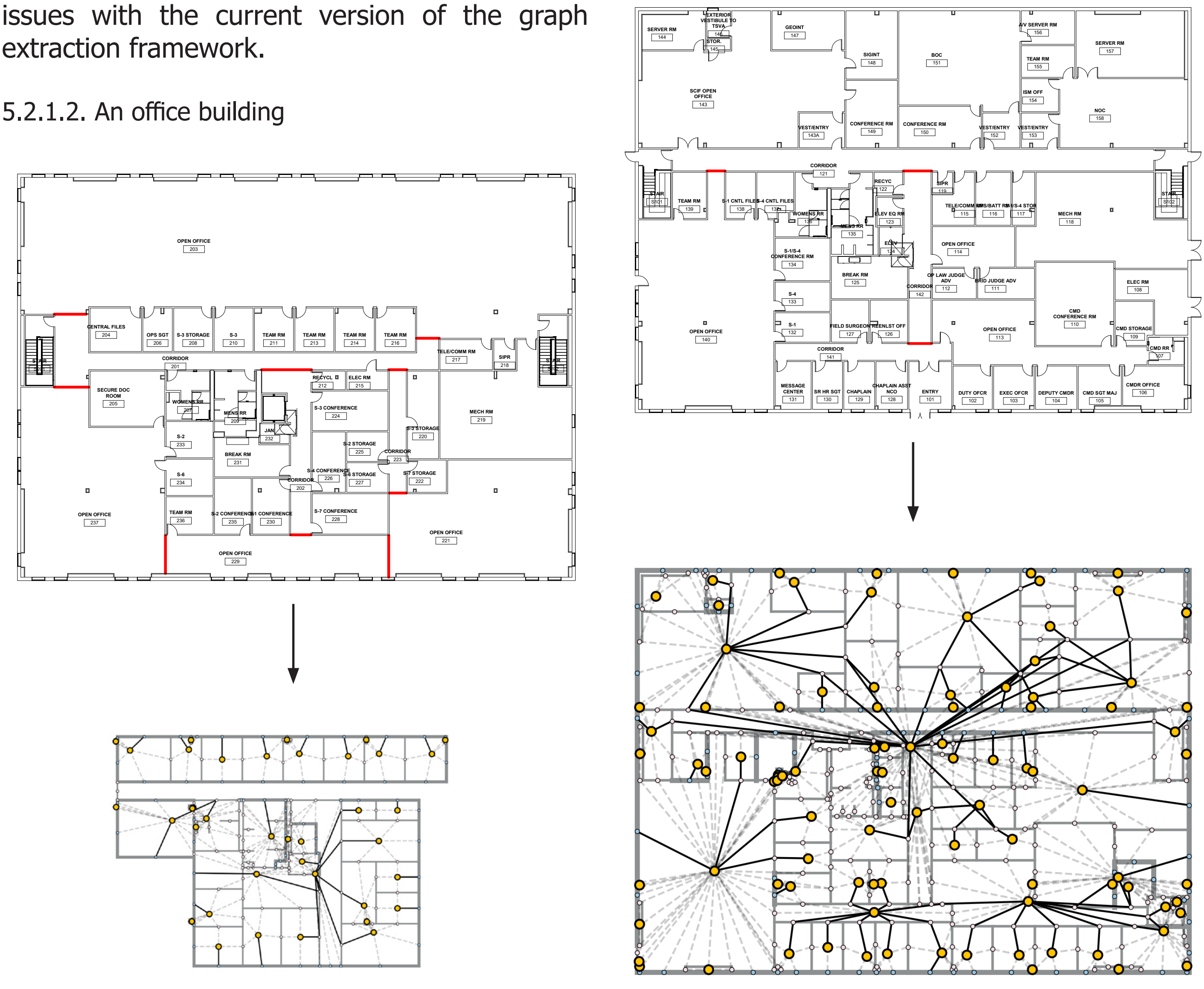


Fig. 23. Room and wall connectivity extraction from a floor plan of a residential building: a) room connectivity graph b) wall connectivity graph

5.2.2. Projects from open model repositories

This section serves to illustrate some notable issues with the current version of the graph extraction framework.

5.2.1.2. An office building



a) b)

Fig. 24. Room connectivity extraction from a floor plan of an office building:
a) the case of multiple connected components on a floor plan b) the case of long corridors

The analysed IFC file of an office building can be obtained from the common files provided by the National Institute of Building Sciences (NIBS) [17]. As previously stated, the model has not been additionally modified and thus, there are some cases in which the requirements are not met.

It can be observed that not all walls in the floor plans are connected. This results in a graph that has multiple connected components. (a definition for a connected component is given in 4.3.1. Definition 4.). The room connectivity graph is computed only for the graph with the most elements (Fig. 24a). A solution would be to identify space separation lines as wall axes and add them as nodes which may make the graph connected.

While most of the rooms that are surrounded by four walls are identified correctly, more complex spaces, such as long corridors that have multiple adjacent wall segments cannot be easily inferred (Fig. 24b). Therefore, for the computation of the minimal weight cycle basis, a different type of edge-weighting principle must be derived when such type of rooms is present in the project. A direct consequence of this is the incorrect categorization of inner and outer wall segments.

Another notable issue is that ventilation shafts are also identified as rooms since information about room types is not being extracted. A threshold for room area may be implemented for room classification.

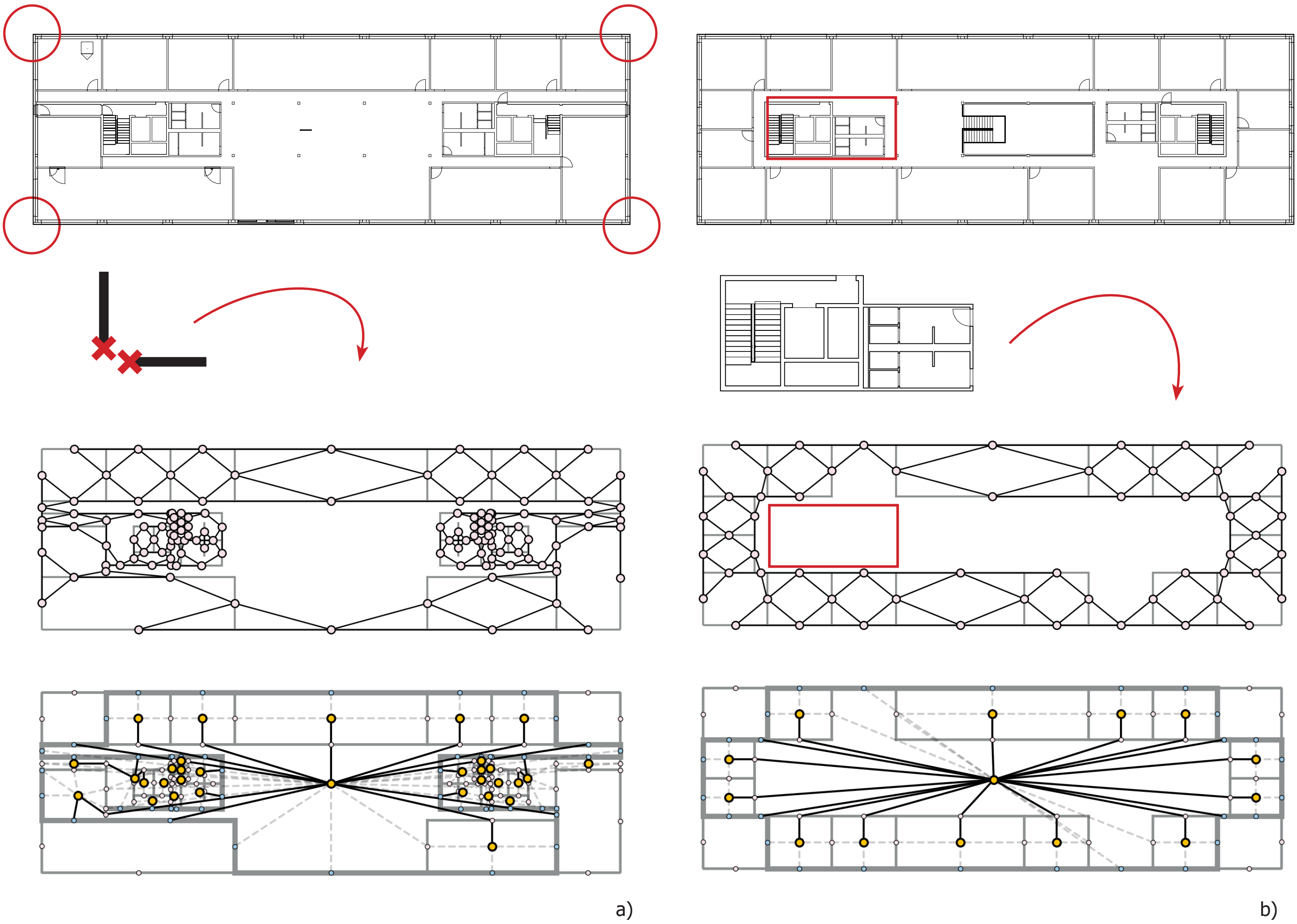


Fig. 25. Room connectivity extraction from a floor plan of an office building:
a) the case of disjoint wall axes b) the case of multiple connected components

5.2.1.2. DigitalHub

The analysed file can be obtained from the repository of the Institute of Energy Efficiency and Sustainable Building, RWTH Aachen [18].

In this example the limitation of wall axis geometry extraction is illustrated. Disjoint segments may lead to incorrect representation of the wall and room connectivity (Fig. 25a). The geometric manipulation of wall axes is used to retrieve wall relationships that are not specified in the entities themselves. However, because of the specific way wall axis geometry is represented in IFC files, not all cases are covered.

As in the previous sample file there are multiple connected components in the floor plan and only the one with the most nodes is being extracted (Fig. 25b).

It can be seen that, again, simple rooms are correctly identified and that there is still an issue with the identification of corridor spaces.

6. Conclusions and future work

It is unknown what the best method to compute a dual of a floor plan is. Even if one computes the minimal-weight cycle basis of a weighted floor plan graph, it is not clear how the edges of the graph must be weighted. One solution might be to generate a whole database of such graphs and weight them iteratively with a cost function which evaluates how “good” the resulting room connectivity graph is. Based on empirical analysis, the developed method is more suitable for the extraction spatial topology from residential buildings because they usually contain a smaller number of long corridors.

Finally, because this framework can be applied for 2D drawings, making use only of simple wall axis geometry, it can easily be made compatible with any kind project formats. This also includes vectorized PDF files and images, a format used to preserve floor plans of historical architecture. Thus, in the future this work may also be used to store and analyse existing buildings.

7. References

1. J Abualdenien, A Borrmann (2021). PBG: A parametric building graph capturing and transferring detailing patterns of building models. Proceedings of the 38th International Conference of CIB W78, Luxembourg, 13-15 October, pp 1-10 (ISSN: 2706-6568). http://itc.scix.net/paper/w78-2021-paper-001
2. Isaac, S., Sadeghpour, F., & Navon, R. (2013). Analyzing Building Information Using Graph Theory. https://doi.org/10.22260/ISARC2013/0111
3. Ismail, Ali & Strug, Barbara & Ślusarczyk, Grażyna. (2018). Building Knowledge Extraction from BIM/IFC Data for Analysis in Graph Databases. https://doi.org/10.1007/978-3-319-91262-2_57
4. Christoph Langenhan, Markus Weber, Marcus Liwicki, Frank Petzold, and Andreas Dengel. (2013). Graph-based retrieval of building information models for supporting the early design stages. Adv. Eng. Inform. 27, 4 (October 2013) https://doi.org/10.1016/j.aei.2013.04.005
5. Lynes, J. A. (1977). Windows and Floor Plans. Environment and Planning B: Planning and De-sign, 4(1), 51–55. https://doi.org/10.1068/b040051
6. Baybars, I., & Eastman, C. M. (1980). Enumerating Architectural Arrangements by Generating Their Underlying Graphs. Environment and Planning B: Planning and Design, 7(3), 289–310. https://doi.org/10.1068/b070289
7. Kavitha, T., Mehlhorn, K., Michail, D. et al. (2008). An O~(m2n) Algorithm for Minimum Cycle Basis of Graphs. Algorithmica 52, 333–349 https://doi.org/10.1007/s00453-007-9064-z
8. F. Bullo (2022). Lectures on Network Systems. 1.6. (ISBN: 978-1986425643). http://motion.me.ucsb.edu/book-lns
9. BuildingSmart, "BuildingSmart.org," (Online). Available: https://technical.buildingsmart.org/standards/ifc/ (Accessed 26 February 2022).
10. Link to Project repository (Online). Available: https://github.com/vangelova-silvia/GraphAnalysisIFC (Created 03 March 2022).
11. Open IFC Model Repository (Online). Available: http://openifcmodel.cs.auckland.ac.nz/ (Accessed 26 February 2022).
12. Example IFC files (Online). Available: https://www.ifcwiki.org/index.php/Examples (Accessed 26 February 2022).
13. BuildingSmart IFC Schema, "BuildingSmart.org," (Online). Available: https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/ (Accessed 26 February 2022).
14. IfcOpenShell (Online). Available: http://ifcopenshell.org/ (Accessed 26 February 2022).
15. pythonOCC (Online). Available: https://dev.opencascade.org/project/pythonocc (Accessed 26 February 2022).
16. NetworkX (Online). Available: https://networkx.org/ (Accessed 26 February 2022).
17. Project 2. Office Building, Common Building Information Model Files and Tools (Online). Available: https://www.wbdg.org/bim/cobie/common-bim-files (Accessed 26 February 2022).
18. DigitalHub IFC model, Institute of Energy Efficiency and Sustainable Building, RWTH Aachen (Online). Available: https://github.com/RWTH-E3D/DigitalHub (Accessed 26 February 2022).