Proposal of a Decision Support System for Planning Bicycle Path Networks – An Approach Based on Graph Theory

Lucas José Machado[1], Diogo Bortolini[2], and Fabiano Baldo[3]

Departament of Computer Science
Santa Catarina State University – UDESC
Joinville, Santa Catarina, Brazil
E-mail: [1]lucasmachado.jose@gmail.com, [2]diogo.bortolini@udesc.br, and [3]fabiano.baldo@udesc.br

***Abstract*—Bicycle paths are increasingly becoming part of road infrastructure of cities. However, the expansion of bicycle paths network demands planning to prioritize investment in order to attend the population needs properly. The connectivity between bicycle paths is considered the most important issue pointed out by cyclists. In this work, it is assumed that the problem of planning bicycle path networks can be solved by using graph theory. Therefore, this work proposes a decision support to help planning bicycle path network, prioritizing the connectivity between the existing bicycle paths by using concepts of clustering, centrality and shortest path. In experiments performed in Joinville – Brazil, it could be seen that the proposed method can help specialists by promoting discussions about the bicycle path network connectivity. Also, it was observed that the usage of graph database increased considerably the solution performance.**

***Index Terms*—Graphs Theory; Centrality; Connectivity; Bicycle Paths; Decision Support System.**

I. INTRODUCTION

A proper bicycle path infrastructure allows cyclists to move safely due to the limitation of a specific space for this mean of transportation. However, such as road networks, the expansion of bicycle paths network also demands planning to prioritize investment in order to attend the population needs properly.

The bicycle path infrastructure needs to fulfill at least the following features: provide feeling of safe, present network connectivity and be pleasant and well kept [1]. Each one of the aforementioned characteristics causes direct impact on the bicycle path network quality and costs. Therefore, planning is mandatory in order to harmonise population needs and public budget.

According to Cardoso and Campos [2], the intersection between bicycle paths inside the network is considered the most important issue pointed out by cyclists in the interviews. Therefore, the planning of new bicycle paths should maintain or, preferably, increase the connectivity between the already existing bicycle paths on the network.

In this work, it is assumed that the problem of planning bicycle path networks can be fitted as a graph problem. Therefore, the objective of this work is to propose a decision support system to help planning the expansion of bicycle path network coverage, prioritizing the connectivity between existing bicycle paths by using concepts of clustering, centrality and shortest path from graph theory. The proposed solution was assessed on the bicycle path network from city of Joinville, Brazil.

The next sections are organized as follows. Section II presents a literature review of bicycle path planning, graphs and graph datadases. Section III makes an comprehensible review of related works. Section IV details the proposed solution. Section V presents the system developed. Section VI shows the results assessment. Section VII presents the conclusion and future works.

II. LITERATURE REVIEW

The present work addresses the following concepts: bicycle paths planning, graph theory and graph databases.

A. Bicycle Path Planning

Urban transportation comprises the set of means and services used to allow movements inside the cities. Within the existing means of transportation, bicycle has a special attention due to some features that it has compared to others [3]. It is cheaper and less pollutant, as well as it uses more rationally the road infrastructure when compared to other means. For these reasons, more investments are being made in bicycle transportation means in order to decrease the streets overcrowding caused by the massive use of cars.

Even being considered a safe transportation mean, many people are afraid to use bicycle, especially in crowded areas. However, studies show that many people accept to use bicycle when bicycle paths are available [4]. However, the existence of bicycle paths are not enough, they must be well kept and connected to each other as a integrated network. Isolated bicycle paths force the cyclist to share the space with motorised vehicles, which decreases considerably the safety and hence demotivates the usage of bicycles.

B. Graphs

A graph can be defined as a set of entities related to each other via relationships. Its visual representation describes entities as circles and the relationships as lines that connect the related entities. Nevertheless, formally a graph is defined as $G = (V, E)$, where V represents the set of vertex and E the set of edges [5]. In the scope of transportation networks, one of the

 ISBN: 978-1-61208-539-5

possible approaches is to represent the intersections between the streets as entities (vertices) and the streets themselves as relationships (edges) between the intersections.

A graph presents a vast range of features and properties. The most important ones for this work are presented below.

1) *Vertex Degree*: the amount of edges incident to a vertex is called the vertex degree. To be considered incident on a vertex x the edge needs to connect x to a vertex y, where x and $y \in V$ [6].
2) *Shortest Path*: The shortest path in a graph is formed by the minimal set of non repeated vertices and a set of edges that connect these vertices in sequence. The size of this path is defined by the amount of edges that form it [7].
3) *Graph with Properties*: The properties are key-value pairs that may belong to vertices or edges, where each vertex and/or edge can have zero or more properties [5]. In a graph that represents a road network, the properties may identify the type of paving, unevenness and other street features.
4) *Graph Connectivity*: A graph is considered connected only if for every pair of vertices there is a path connecting both vertex, otherwise it is not connected. Although a graph may not be connected as a whole, it is possible to find pairs of connected vertices. This happens when there is at least one path that connects the pair [7].
5) *Centrality*: The centrality metrics are used to evaluate the importance of vertices and/or edges of a graph [8]. Some of them are: a) The degree centrality refers to the vertex degree; b) The proximity centrality refers to the distance of a vertex to all other vertices; c) The betweenness centrality classifies the vertices by the amount of times that each vertex participates on the shortest paths [8].
6) *Cluster*: Cluster means a set of elements that are grouped by an common feature [9]. The process used to identify clusters is called clustering [10]. The elements belonged to the same cluster are very similar, while the elements in different clusters have lower similarity [11]. The similarity can be verified by checking distances or comparing centrality metrics [11].
7) *Prediction of connections*: The prediction of connections aims to estimate the probability to have relationship between two elements on a network that are not directly connected yet [12]. This prediction is calculated analysing properties such as centralities, distances and so forth.

C. Databases

During the last decades the most used model for data persistence is the relational. However, it has not been conceived to storage graph. So, it performance considerably decreases as the graph stored increases [5]. This problem is faced especially in queries that perform recursive self-joins in order to find data indirectly related inside the graph.

An alternative to tackle the problems presented by the relational model is to use the graph model. The databases that implement this data model tend to remain the performance constant, even when the dataset grows, because the queries are localized to a portion of the graph. So, the execution time for each query is proportional to the size of the part of the graph traversed, rather than the size of the overall graph [5]. Another benefit is the flexibility of the graph model. It allows adding new types of vertices and new types of edges without compromising the existing network.

To select the graph database used in this work, it has been performed a evaluation of the three most popular graph databases from DBEngines [13]. The databases evaluated are Neo4J, OrientDB and Titan.

Having into account the analysis presented in [14] - [16], it has been selected the Neo4J. It presents the best performance in queries, as well as in shortest path and betweenness centrality computations. This can be explained because it was designed to optimize the execution of graph exploring algorithms, rather than insert and delete operations.

III. Related Works

This section presents a brief overview about each related work found in the literature.

Sevtsuk and Mekonnen [8] presented an approach to include builds, as schools, hospitals, houses and building, in the analysis of transportation networks modeled as a graph. These builds are represented as vertices connected to the streets through specific edges. This approach uses centrality metrics adapted specifically for analysing trasnportation networks with buildings.

Jun and Yikui [17] proposed a framework for constructing decision support systems for planning transportation networks. This framework had as objective to contribute for solving the limitations of traditional methods to fulfill the requirements of modern transportation systems. This framework describes the components and decision indicators needed to plan and assess any kind of transportation network.

Mishra, Welch and Jha [18] proposed the adaptation of graph connectivity metrics used in social networks to the scope of transportation networks. Most of this metrics are specializations of classical centrality metrics presented in graph theory.

Porta, Crucitti and Latora [19] presented a study demonstrating that the primal modeling is more appropriate to represent the streets in a transportation network for applying algorithms of vertex-centered analysis. Moreover, they emphasize the diversity of existing centrality metrics and explain how each one of them can be applied in the primal model.

Based on the review presented above, it can be noted that works like [8], [18] and [19] highlight that to represent a transportation network as a graph is considered a suitable approach. Besides that, all reviewed works state that centrality metrics can be applied in the connectivity analysis of transportation networks. At last, is can be seen that none of the reviewed works is devoted to plan or analyse the bicycle path network, which represents a lack of solutions concerning this important means of transport.

 ISBN: 978-1-61208-539-5

IV. Proposed Solution

Considering that the aim of this work is to aid planning bicycle path networks, the proposed method should be able to suggest bicycle paths that leverage the connectivity among the existing bicycle paths. In order to do that, it is assumed that it is necessary to interconnect the clusters of bicycle paths that are not connected yet. Therefore, to fulfill this objective, the proposed method is composed of the following steps:

a) Identification of bicycle path clusters by finding groups of interconnected vertices. Each cluster should be formed by vertices that have at least one edge connecting themselves and none connecting them with other clusters;
b) Selection of pairs of vertices that will be used as source and destination in suggested new bicycle paths in step (c). Each vertex from the pair must belong to a different cluster;
c) Suggestion of new bicycle paths that increase the connectivity between the clusters identified in step (a), use the vertices from each pair selected in step (b). These suggested paths must be the shortest ones that pass through existing streets that do not have bicycle paths.

The following subsections present the main features of the proposed solution.

A. Data Schema

The data schema designed models streets as edges and the intersection among them as vertices. Besides that, each street can also be composed of several parts (edges) connected to each other by intersections as presented in Figure 1. This means that a single street can be formed by the combination of several edges interconnected by intersections. Therefore, the proposed schema is defined as a graph $G = (V, E)$ where V is the set of intersections and E is the set of street parts that connect intersections from V.

Each intersection in V contains the following attributes: geometry (point), cluster identification and betweenness classification. Meanwhile, each street (part) in E has the following attributes: geometry (line), length and the definition whether it has or not bicycle path.

B. Clustering Algorithm

The clustering algorithm designed in this work has been adapted from the one proposed by Girvan and Newman [20], which uses the betweenness centrality metric to identify the edges that represent bridges that connect clusters.

As this metric emphasizes the vertices and edges used in most of the shortest paths, the edges responsible to connect different clusters are identified because they participate in many paths that connect these two clusters.

The clustering algorithm presented in Figure 2 works as follows. First, it is calculated the betweenness centrality of each edge (lines 4-11). After that, the algorithm discards the edge with the highest centrality value (lines 12-13). Then, it is performed a breadth-first search to reorganize the clusters (line 14), attributing for each vertex the cluster id of its neighbourhood. If the neighbourhood do not have cluster id

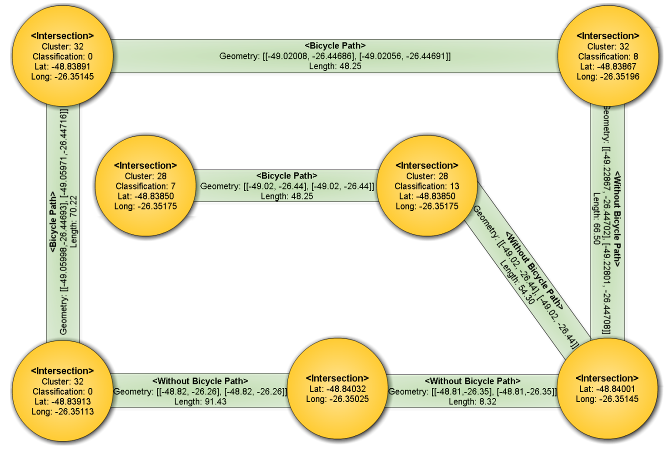

Figure 1. Graph data schema

then a new id is defined. This process is repeated until the stop criterium is fulfilled (line 3). In this work, the stop criterium is the number of clusters formed, defined by the user, previously. Only vertices that have edges with bicycle path are considered in this process.

The algorithm allows to discard only one edge for each iteration. This restriction is imposed due to the fact that it can not be ensure that after removing the edge with highest betweenness score the others will stay at the same position. So, after each discard it is necessary to perform the clustering process again.

The clustering algorithm has been implemented in Gremlin language, using the graph processing library called JUNG.

C. Pair Selection

The selection of vertices to compose the pairs is based on the value of their betweenness centrality metric. Roughly, there will be selected the ones that have the higher betweenness centrality scores. Moreover, both vertices of the pair need to belong to different clusters. Using this approach, it is expected to select the vertices that participate in most of the shortest paths inside the road network and, hence, which allow the highest bicycle network connectivity.

The algorithm starts with the selection of pairs (a, b) by selecting a vertex a from the bicycle path network that belongs to a cluster and that has the betweenness score greater than zero, and that has at least one edge connecting it to any other vertex c, where c must belong either to another cluster or not be part of any cluster. This is performed to ensure that the selected vertex is near to the cluster border. Then, the same process is performed to select the vertex b. After identifying the sets A and B, their elements are combined to form the pair where the betweenness score of $(a_i + b_i)$ is greater than $(a_{i+1}+b_{i+1})$. So, the result is a list of pairs sorted by the value of betweenneess from the highest to the lowest. This algorithm has been implemented using Cypher language in Neo4J.

 ISBN: 978-1-61208-539-5

```
Data:
    G = graph of bicycle path network (only the
  edges with bicycle paths);
    min_clusters = minimal number of clusters;
Result:
    clusters = set of vertex grouped by cluster;
begin
  | clusters =
  |  travel_edges_defining_groups(edges(G));
  | while quantity(clusters) < min_clusters do
  |   | for v ∈ vertex(G) do
  |   |   | for v' ∈ vertex(G) where v' ≠ v do
  |   |   |   | path = shortest_path(v, v');
  |   |   |   | for a ∈ edges(path) do
  |   |   |   |   | increase betweenness
  |   |   |   |   |   classification of a by 1;
  |   |   |   | end
  |   |   | end
  |   | end
  |   | a' = high-
  |   |  est_betweenness_classification(edges(G));
  |   |
  |   | remove a' from edges(G);
  |   | clusters =
  |   |  travel_edges_defining_groups(edges(G));
  | end
  | return clusters;
end
```

Figure 2. Algorithm of Clustering

```
   Data:
       G = graph of road network;
       v' = source vertex of path;
       v'' = destination vertex of path;
   Result:
       path = set of edges between source and
   destination;
1  begin
2    | penalty = 0.2;
3    | while penalty <= 1.0 do
4    |   | path = a_star(
5    |   |     source: v',
6    |   |     destination: v'',
7    |   |     cost_function: if edge has bicycle path
     |   |  then
8    |   |     | cost(edge)*penalty
9    |   |     else
10   |   |     | cost(edge)
11   |   | );
12   |   | if ∃ path then
13   |   |   | if ∃ edge ∈edges(path) | edge without
     |   |   |   bicycle path then
14   |   |   |   | return path;
15   |   |   | end
16   |   | end
17   |   | penalty = penalty + 0.1;
18   | end
19   | return ∅;
20 end
```

Figure 3. Algorithm to Calculate Shortest Path

D. New Bicycle Paths Calculation

The last step proposes new paths to connect the existing bicycle paths belonged to the network. The suggested paths should contain at least one street without existing bicycle path. The result should be sorted prioritizing the suggestions with shortest length. This restriction contributes to save costs in building new bicycle paths, as well as helps the cyclists to move around the city faster.

To calculate the suggested paths it has been modified the original A* algorithm to enable it to prioritize the suggestions that present streets without bicycle paths, as presented in Figure 3. Initially, this algorithm applies a penalisation of +20% (line 2) on the length of the streets with bicycle paths. For each interaction the algorithm tries to find a path between the pair of vertices received as input (lines 7-12). If it has succeeded, then it is verified if the path has at least one edge without bicycle path (line 13). If so, then the path is returned as a suggestion (line 14). If not, then the penalty is increased in 10% (line 17) and the process of finding a path is repeated again. This loop is repeated until the length of the streets with bicycle paths has been doubled (line 3). This algorithm has been implemented in Java as an extension inside Neo4J.

The algorithm that suggests new bicycle paths, as seen in Figure 4, uses the other algorithms specified in this work, as follows. First, it executes the clustering (line 2) and the betweenness classification (line 3). Afterwards, it executes the pair selection (line 4) and for every pair it finds the shortest path (line 7) and select the paths with at least on edge without bicycle path (lines 10-11). Finally, it filters the suggestions considered equivalents (lines 14-15). Equivalent suggestions are those that contain a subset of edges without bicycle paths present in another suggestion already calculated.

V. System Development

This section presents the prototype architecture and the data used to build the assessment scenario.

A. System Architecture

The developed prototype has two layers, named front-end and back-end. The front-end encompasses the user interface made with AngularJS framework [21] and Leaflet library [22]. The back-end contains the web service and the extension for Neo4J. The web service processes the requests from front-end, makes requests for either Neo4J or Neo4J extension, and sends back answers to front-end. It has been implemented

Data:
G = graph of road network;
Result:
suggestions = set of suggested bicycle paths between pairs of vertex;
begin
  apply clustering algorithm;
  apply betweenness classification algorithm;
  $pairs$ = select_pairs(vertex(G));
  $suggestions = \emptyset$;
  for $pair \in pairs$ **do**
    $path$ = calculate_bicycle_path($pair$);
    $new\_suggestion = \emptyset$;
    for $edge \in edges(path)$ **do**
      if *edge does not have bicycle path* **then**
        add $edge$ to $new\_suggestion$;
      end
    end
    if $|new\_suggestion| > 0 \wedge new\_suggestion \notin suggestions$ **then**
      add $new\_suggestion$ in $suggestions$;
    end
  end
  return $suggestion$;
end

Figure 4. Algorithm to Suggest Bicycle Paths

using Ruby on Rails framework. The extension for Neo4J was implemented using Java. Figure 5 shows a prototype screenshot identifying the clusters in colours and the suggested bicycle path in black inside the red rectangle.

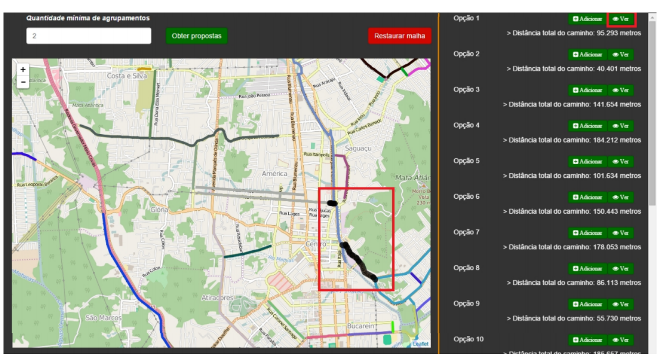

Figure 5. Prototype suggestions of bicycle paths

B. Assessment Scenario

To perform the experiments it has been chosen the city of Joinville as scenario. This scenario comprises the road network provided by Open Street Map (OSM) database [23] and the bicycle path network provided by the BikeTrails project database [24]. The OSM is a website for collaborative road mapping of any part of the world. The BikeTrails is a project developed by Santa Catarina State University (UDESC) that collects and provides the map of bicycle paths from Joinville city [25].

To extract the data from OSM it was necessary to download a XML with the road network of Joinville. Then, this file was imported to PostGIS using the $osm2po$ tool. The bicycle path from BikeTrails was already available in a PostGIS database. The geometries of the OSM database were converted into a graph topology by using the $ST\_InitTopoGeo$ function from PostGIS [26]. All streets from OSM were included in the topology as ”Without Bicycle Path” edges. After that, the bicycle path geometries from BikeTrails were included in this topology by a implemented function called $TopoGeo\_AddLineString$. The edges from BikeTrails are labeled as ”Bicycle Path”. At the end, the topology database has two tables, one with the intersections and another with the streets with and without bicycle paths.

To import this topology from PostGIS to the graph database Neo4J it has been created two JSON files, one containing the intersections (the vertices) and another containing the parts of the streets (the edges). To import these files to the Neo4J it has been implemented a program in Ruby language that makes a REST call to Neo4J containing a statement written in Cypher to insert either a vertex or a edge in the topology.

VI. Result Analysis

This work was evaluated in November 2015 in meetings of the infrastructure working group promoted by IPPUJ (Institute of Research and Planning for Sustainable Development in Joinville). IPPUJ is the public organization in charge of planning Joinville urbanization. This meetings had the participation of cycling groups, universities and the IPPUJ staff, as well.

The assessments made during these meetings comprehend interviews with specialist and environmental observations of the participants. As result, it could be observed that the proposed method, as well as the implemented prototype, may help to improve the discussions about the bicycle path network planning. As main contributions it has been pointed out the possibility to visualize the existing bicycle paths clustered in different colors in a digital map. According to the participants, to distinguish the clusters of connected bicycle paths with colours facilitate considerably the identification of the disrupted points on the network.

Another positive aspect mentioned is related to the classification of the suggested bicycle paths. They are sorted by length prioritising the bicycle paths with shortest length. However, it has been observed as shortcoming that the suggested paths sometimes did not privilegiate straight lines, which is an important aspect mentioned by the cyclists. Also, it has been observed that some suggestions are very similar to each other which restricts the set of applicable suggestions.

Some advises collected during the meetings are regarded to the suggestion criteria. Even considering that length is an important criterium, some participants propose to include as criteria also the direction, inclination and width of the street. Streets less inclined, wider and single directed are preferred.

 ISBN: 978-1-61208-539-5

Regarding the performance, it was observed that the system properly respond to the users with the suggested bicycle paths, even when it is necessary to calculate thousands of path in order to discard paths that are a sub-path of another suggested one. During the experiments, it was observed times between 30 seconds and 1 minute in most of the executions.

VII. CONCLUSION

The bicycle path planning is a complex process that embraces many different geographical, social and economical factors. To help the specialists, this work provides a solution conceived as a decision support system that analyses the road network and offers suggestions for new bicycle paths to improve the connectivity of the existing bicycle path network with the lowest cost possible.

In this work, the problem of suggesting bicycle paths was fitted as a graph analysis. Then, a graph data schema was created to represent the road network topology. This data schema aided in the clusters identification and the betweenness centrality calculation, as well as the implementation of the shortest path algorithm and, hence, the suggestion for new bicycle paths.

Based on the results assessment, it could be seen that the proposed solution helped the specialists on road network planning. Moreover, the prototype showed that the solution is able to promote discussions about the importance of connectivity on bicycle path network planning considering the existence of bicycle path clusters.

As future works, it is possible to consider other road network properties as criteria for suggesting bicycle paths, such as direction, inclination and width. Besides that, it is also possible to consider to change the undirected graph to the directed one in order to evaluate its impacts and benefits.

REFERENCES

[1] Transport Scotland, “Cycling by design 2010,” 2010, [Retrieved: Feb., 2017]. [Online]. Available: http://www.transport.gov.scot/system/files/uploaded_content/documents/tsc_basic_pages/Environment/Cycling_by_Design_2010__Rev_1__June_2011_.pdf

[2] P. de B. Cardoso and V. B. G. Campos, “Methodology for defining a bicycle network system,” 2013, [Retrieved: Feb., 2017]. [Online]. Available: http://www.anpet.org.br/ssat/interface/content/autor/trabalhos/publicacao/2013/188_RT.pdf

[3] Brazil, “Federal law n.º 12.587, from jan. 3rd of 2012,” 2012, urban Mobility Law. [Online]. Available: http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2012/lei/l12587.htm

[4] K. J. Krizek and R. W. Roland, “What is at the end of the road? understanding discontinuities of on-street bicycle lanes in urban settings,” *Transportation Research Part D: Transport and Environment*, vol. 10, no. 1, pp. 55–68, 2005.

[5] I. Robinson, J. Webber, and E. Eifrem, *Graph databases*, 1st ed. Beijing, China: ”O’Reilly Media, Inc.”, 2013.

[6] R. Diestel, “The degree of a vertex,” in *Graph Theory*, 3rd ed., ser. Electronic library of mathematics. Hamburg, Germany: Springer, 2006, pp. 3–13.

[7] R. Balakrishnan and K. Ranganathan, “Basic results,” in *A Textbook of Graph Theory*, 2nd ed., ser. Universitext (Berlin. Print). New York: Springer New York, 2012, pp. 10–39.

[8] A. Sevtsuk and M. Mekonnen, “Urban network analysis: A new toolbox for measuring city form in arcgis,” in *Proceedings of the 2012 Symposium on Simulation for Architecture and Urban Design*, ser. SimAUD ’12, San Diego, CA, USA, 2012, pp. 18:1–18:10.

[9] E. Hartuv and R. Shamir, “A clustering algorithm based on graph connectivity,” *Information Processing Letters*, vol. 76, no. 4, pp. 175–181, 2000.

[10] S. E. Schaeffer, “Graph clustering,” *Computer Science Review*, vol. 1, no. 1, pp. 27–64, 2007.

[11] R. Xu and D. Wunsch, “Survey of clustering algorithms,” *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, May 2005.

[12] J. C. Valverde-Rebaza and A. de Andrade Lopes, *Link Prediction in Complex Networks Based on Cluster Information*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 92–101. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-34459-6_10

[13] DBEngines, “Db-engines ranking - popularity ranking of graph dbms,” 2017, [Retrieved: Feb., 2017]. [Online]. Available: http://db-engines.com/en/ranking/graph+dbms

[14] S. Jouili and V. Vansteenberghe, “An empirical comparison of graph databases,” in *Social Computing (SocialCom), 2013 International Conference on*, Sept 2013, pp. 708–715.

[15] A. Vaikuntam and V. K. Perumal, “Evaluation of contemporary graph databases,” in *Proceedings of the 7th ACM India Computing Conference*, ser. COMPUTE ’14. New York, NY, USA: ACM, 2014, pp. 6:1–6:10.

[16] Y. Liu, “A survey of persistent graph databases,” Master’s thesis, Dept. of Comput. Sci., Kent State University, Columbus, Ohio, 2014.

[17] D. Jun and M. Yikui, “Intelligent decision support system for road network planning,” in *ISECS Int. Colloquium on Computing, Communication, Control, and Management*, vol. 4, Aug 2009, pp. 139–142.

[18] S. Mishra, T. F. Welch, and M. K. Jha, “Performance indicators for public transit connectivity in multi-modal transportation networks,” *Transportation Research Part A: Policy and Practice*, vol. 46, no. 7, pp. 1066 – 1085, 2012.

[19] S. Porta, P. Crucitti, and v. Latora, “The network analysis of urban streets: A primal approach,” *Environment and Planning B: Planning and Design*, vol. 33, no. 5, pp. 705–725, 2006.

[20] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[21] Google, “Angularjs,” 2017, [Retrieved: Jan., 2017]. [Online]. Available: https://angularjs.org/

[22] V. Agafonkin, “Leaflet,” 2017, [Retrieved: Feb., 2017]. [Online]. Available: http://leafletjs.com

[23] Open Street Map, “Open street map,” 2017, [Retrieved: Feb., 2017]. [Online]. Available: https://www.openstreetmap.org

[24] UDESC, “Biketrails,” 2017, [Retrieved: Feb., 2017]. [Online]. Available: http://bdes.dcc.joinville.udesc.br:100/ciclo

[25] G. H. R. Costa, F. Baldo, and L. L. Taveira, “A service-oriented platform to map cycling routes,” in *Proceedings of the Fifth International Conference on Advanced Geographic Information Systems, Applications, and Services*, ser. GEOProcessing 2013. IARIA, 2013, pp. 1–6.

[26] The PostGIS Development Group, “Chapter 11. topology,” 2017, [Retrieved: Jan., 2017]. [Online]. Available: http://postgis.net/docs/Topology.html

[27] D. Abramson *et al.*, “A clustering-based link prediction method in social networks,” *Procedia Computer Science*, vol. 29, pp. 432–442, 2014.