

Plataforma Computacional para Construção de um Banco de Dados de Grafo do Sistema de Transporte de Curitiba

Altieris M. Peixoto¹, Marcelo O. Rosa¹, Ricardo Lüders¹, Keiko V. O. Fonseca¹

¹CPGEI / DAELT – Universidade Tecnológica Federal do Paraná (UTFPR)
Av. Sete de Setembro 3165 – 80.230-901 – Curitiba – PR – Brazil

Abstract. *The study of dynamic relationships between topological structures of a transport network and mobility patterns in this network is important for building smart solutions to problems of reliability, optimization, vulnerability and traffic forecast. However, one of the biggest challenge in transport planning and operation management is to deal with large volume of data often provided by geolocation sensors installed in vehicles. This work aims at modeling Curitiba transport system using a graph database built with information of daily transport operation available from an open data repository. The main contribution is the development of a computational framework for automatic building of the graph database from raw data of the repository. Illustrative results of several metrics for Curitiba transport are presented by showing the potential analysis provided by the proposed tool.*

Resumo. *O estudo das relações dinâmicas entre as estruturas topológicas de uma rede de transporte e os padrões de mobilidade nesta rede se faz importante para a criação de soluções inovadoras para problemas de confiabilidade, otimização, vulnerabilidade e previsão de tráfego. Todavia, um dos maiores desafios da área de gerência, planejamento e operação do transporte é a manipulação de um grande volume de dados, geralmente provenientes de sensores de localização instalados nos veículos. Este trabalho tem por objetivo modelar o sistema de transporte coletivo da cidade de Curitiba usando uma base de dados de grafos gerada a partir de um repositório de dados abertos com informações da operação diária do transporte. A principal contribuição é o desenvolvimento de uma plataforma computacional para a geração automática da base de grafo a partir do repositório de dados abertos. Resultados ilustrativos de diversas métricas do transporte de Curitiba são apresentados, mostrando o potencial de análise da ferramenta.*

1. Introdução

Segundo [Ferraz and Torres 2004], uma maior utilização do transporte público favorece a mitigação dos problemas de congestionamento, poluição, acidentes, desumanização e outros males que afligem as cidades modernas. Conhecida pela constante inovação em mobilidade urbana desde os anos 70, Curitiba investe na priorização do transporte público, sendo citada como uma cidade referência na área [Reed 2015]. Desde 1974, a cidade de Curitiba conta com sua Rede Integrada de Transporte (RIT), baseada em ônibus urbanos e que permite que o passageiro troque de linhas em determinados pontos com o custo de uma única tarifa. Atualmente, Curitiba é atendida por uma frota de 1410 ônibus operantes

(mais reserva) e atende cerca de 1.389.731 passageiros por dia com 251 linhas de ônibus, 329 estações e 21 terminais [URBS 2019].

Todavia, uma das dificuldades no estudo da mobilidade e transporte público urbanos é a coleta e manipulação das mais variadas fontes e volumes de dados disponíveis [Wessel et al. 2017]. O sistema de transporte coletivo de ônibus de Curitiba possui atualmente um repositório com dados abertos sobre sua operação¹. Este repositório é continuamente alimentado com dados brutos da operação provenientes de sensores e dispositivos móveis que capturam a movimentação dos ônibus ao longo do dia. Entretanto, a utilização destes dados requer um modelo de dados adequado para análises mais complexas, que envolvem não apenas dados estáticos de configuração da rede de transporte, mas também dinâmicos de sua operação.

Em geral, modelos estáticos de redes complexas são utilizados para caracterizar estatisticamente as redes de transporte a partir de métricas como centralidades de grau, proximidade e intermediação, distribuição dos comprimentos dos caminhos mínimos, coeficiente de agrupamento, dentre outras. Métricas de redes complexas foram utilizadas para caracterizar a conexão entre estações do sistema de transporte na Polônia [Sienkiewicz and Hołyst 2005], China [Xu et al. 2013] e Brasil [Izawa et al. 2017]. A caracterização temporal de redes de transporte de ônibus de Curitiba aparece em [Curzel et al. 2019] usando *link streams* e de diversos modais de transporte na Grã-Bretanha em [Gallotti and Barthelemy 2015] usando redes *multilayer*.

Para análise de desempenho da operação do transporte, é necessário incluir informação de tempo no modelo de rede utilizado. Em [Holme and Saramäki 2012], uma variedade de termos é apresentada para designar redes cuja estrutura é dependente do tempo: *temporal graphs*, *evolving graphs*, *time-varying graphs*, *time-aggregated graphs*, *time-stamped graphs*, *dynamic networks*, *dynamic graphs*, *dynamical graphs*, entre outros. O objetivo aqui não é discutir estes termos em profundidade, mas apenas destacar a ampla gama de opções para modelagem de redes dinâmicas. Particularmente, o modelo *Time Varying Graph* (TVG) é adotado neste trabalho, tendo sido usado em [Flocchini et al. 2009], além de [Tang et al. 2010a] e [Tang et al. 2010b] como uma sequência discreta e ordenada de grafos (a forma mais intuitiva de representação de um grafo variante no tempo). Um extensão dos conceitos clássicos de grafos para TVGs é apresentada em [Nicosia et al. 2012]. Em [Maduako et al. 2019], é apresentado um modelo TVG do transporte de ônibus de Greater Moncton no Canadá e sua implementação no banco de dados de grafos Neo4j². Conforme apontado em [Vicknair et al. 2010], há um interesse crescente em bancos de dados noSQL (*not only SQL*), como o Neo4j, para armazenamento e recuperação de dados com informação dinâmica.

Este trabalho tem por objetivo modelar o sistema de transporte coletivo da cidade de Curitiba usando uma base de dados de grafos gerada a partir de um repositório de dados abertos com informações da operação diária do transporte. O modelo proposto é baseado em [Maduako et al. 2019], sendo o foco do trabalho a implementação deste modelo para Curitiba. A principal contribuição é o desenvolvimento de uma plataforma computacional para a construção automática da base de grafo a partir do repositório de dados abertos. A infraestrutura computacional e a transformação dos dados do repositório para a base de

¹<http://dadosabertos.c3sl.ufpr.br/>

²<http://neo4j.org>

dados de grafo são detalhadas. Resultados ilustrativos de diversas métricas do transporte de Curitiba são apresentados, mostrando o potencial de análise da ferramenta.

Na organização do artigo, a Seção 2 contém conceitos básicos e trabalhos relacionados. A Seção 3 apresenta o modelo proposto. A Seção 4 define as métricas de redes complexas. A Seção 5 descreve a plataforma computacional desenvolvida. A Seção 6 apresenta os resultados obtidos das métricas computadas para o transporte de Curitiba a partir do banco de dados de grafos. Conclusões e trabalhos futuros estão na Seção 7.

2. Time Varying Graph (TVG)

O termo *Time Varying Graph* (grafo variante no tempo ou TVG) pode ser encontrado em [Flocchini et al. 2009], [Tang et al. 2010a], [Tang et al. 2010b] e [Nicosia et al. 2012]. O formalismo adotado no presente trabalho é devido a [Casteigts et al. 2012].

Um TVG é uma quintupla $\mathcal{G} = (V, E, \mathcal{T}, \rho, \zeta)$, tal que V e E são os conjuntos de vértices e arestas do grafo rotuladas pelo conjunto L de rótulos ($E \subseteq V \times V \times L$), respectivamente, $\mathcal{T} \subseteq \mathbb{T}$ é o tempo de vida do sistema, $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ é a função de presença e $\zeta : E \times \mathcal{T} \rightarrow \mathbb{T}$ é a função de latência. O domínio temporal \mathbb{T} é assumido igual a \mathbb{N} para sistemas a tempo discreto e \mathbb{R}^+ para sistemas a tempo contínuo.

A função ρ indica a presença (1) ou não (0) de uma aresta em um determinado instante de tempo e a função ζ representa o tempo necessário para cruzar uma aresta a partir de um determinado instante de tempo (a latência pode variar com o tempo). Este modelo é geral o suficiente para representar vários cenários, desde redes de comunicação e transporte até sistemas complexos ou redes sociais [Casteigts et al. 2012].

Por exemplo, um TVG pode ser usado para modelar viagens de uma cidade A para uma cidade B . As cidades A e B são os vértices do grafo conectados por duas arestas paralelas rotuladas ‘ônibus’ e ‘carro’ para indicar viagens feitas de ônibus e de carro, respectivamente. Neste caso, a viagem de ônibus inicia em instantes de tempo determinados e, portanto, a função ρ da aresta ‘ônibus’ retorna valor 1 apenas nas datas e horários para os quais a viagem de ônibus pode ser iniciada. A função de latência ζ de cada aresta pode retornar diferentes valores, dependendo dos tempos de viagem de ônibus e de carro (ou mesmo diferentes valores para uma mesma aresta, dependendo do tempo de viagem em diferentes horários).

A partir deste modelo geral de [Casteigts et al. 2012], um modelo de dados é proposto em [Cattuto et al. 2013] para capturar o comportamento temporal de redes sociais e implementado no banco de dados de grafos Neo4j. O banco Neo4j implementa um tipo de grafo denominado *property graph model*, capaz de representar multigrafos direcionados, rotulados e com atributos. Estes grafos permitem a representação de vértices e arestas rotulados, além de metadados (propriedades) associados aos vértices e arestas [Rodriguez and Neubauer 2010], particularmente importantes na implementação de TVGs. Além disso, o banco de dados Neo4j tem outras características que favorecem a implementação: (i) armazenamento persistente e transacional de grafos de elevada dimensão; suporte para análise em profundidade via buscas eficientes de múltiplos saltos; (iii) suporte para linguagem declarativa de *queries* de grafos denominada Cypher³.

³<https://neo4j.com/docs/cypher-manual/current/>

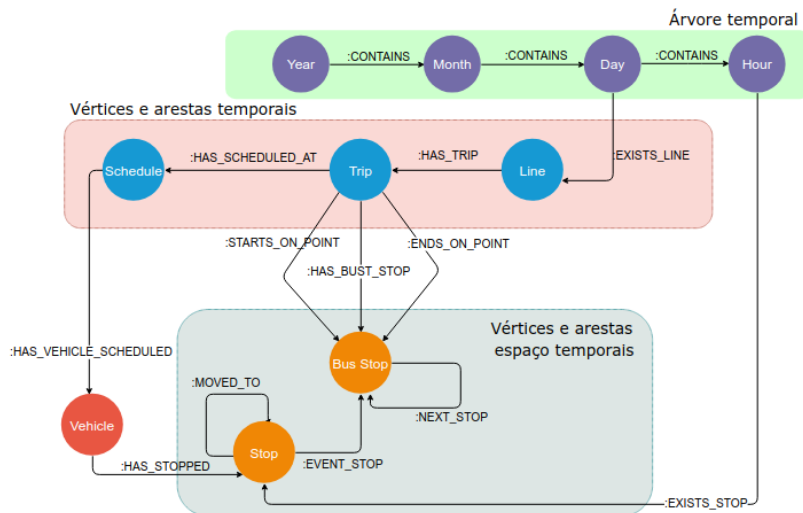


Figura 1. Modelo TVG da movimentação dos ônibus do transporte.

O modelo proposto para o transporte público de Curitiba tem por base o modelo de grafo utilizado em [Maduako et al. 2019] para o transporte de ônibus da região de Greater Moncton no Canadá. Este modelo deriva de [Cattuto et al. 2013], tendo sido também implementado no banco Neo4j.

3. Modelo Proposto

O modelo proposto para o transporte público de Curitiba é mostrado na Figura 1. Este modelo tem por base o modelo de [Maduako et al. 2019], com a inclusão do vértice *Vehicle* para reduzir parte das informações sobre os veículos (ônibus) utilizados nas viagens que eram incluídas como atributos em vértices e arestas. A Figura 1 mostra um grafo com vértices temporais (*Schedule*, *Trip* e *Line*) e espaço-temporais (*BusStop* e *Stop*). Os vértices temporais carregam informações que variam com o tempo, enquanto os vértices espaço-temporais possuem informações de tempo associadas a dados de posicionamento georreferenciado. Vértices adicionais definem agrupamentos temporais do modelo que permitem recuperar informações para diferentes escalas de tempo (*Year*, *Month*, *Day* e *Hour*). Em resumo, o modelo da Figura 1 representa dependências temporais e espaço-temporais entre os vértices do grafo que descreve a operação de um sistema de transporte.

Os atributos dos vértices do grafo da Figura 1 são mostrados nas Tabelas 1, 2, 3, 4, 5 e 6 do Apêndice. Cada linha de ônibus (vértices *Line* - Tabela 5) dá origem a viagens (vértices *Trip* - Tabela 1). As viagens são programadas (vértices *Schedule* - Tabela 3) para serem executadas por veículos (vértices *Vehicle* - Tabela 2), que geram sequências de movimentação e paradas (vértices *Stop* - Tabela 6). Estas paradas incluem paradas nos pontos de ônibus (vértices *Bus Stop* - Tabela 4) para embarque e desembarque de passageiros. Além disso, as viagens possuem pontos de ônibus inicial, intermediário(s) e final (vértices *Bus Stop* - Tabela 4). Além disso, linhas de ônibus podem ser agrupadas e recuperadas por dia (aresta *EXISTS_LINE*), assim como paradas podem ser agrupadas e recuperadas por hora (aresta *EXISTS_STOP*). Os atributos das arestas da Figura 1 são mostrados na Tabela 7 do Apêndice. As arestas estabelecem relacionamentos entre os vértices do grafo, além de carregarem atributos espaciais, temporais ou espaço-temporais

para orientar consultas futuras. Conforme será discutido na Seção 5, a construção do grafo correspondente ao modelo da Figura 1 no Neo4j é realizada a partir dos dados brutos da operação do transporte.

4. Métricas de Redes Complexas

Uma vez que o grafo do sistema de transporte é modelado e armazenado no Neo4j, métricas de redes complexas podem ser computadas para avaliar o comportamento do sistema. Esta seção descreve as métricas utilizadas neste trabalho: i) centralidade de grau; ii) *page rank*; iii) caminho mínimo e diâmetro; iv) centralidade de intermediação. Estas métricas foram escolhidas por permitirem uma avaliação da importância dos vértices na rede (pontos de ônibus), tanto do ponto de vista estático da topologia da rede quanto dinâmico da movimentação dos ônibus, e sua relação com os outros pontos da rede.

Proposto por [Freeman 1979], a **centralidade de grau** é proporcional ao número de arestas (de entrada, ou de saída em grafos direcionados) que se conectam a um determinado vértice. Quanto maior a centralidade de grau de um vértice, maior é o número de conexões com outros vértices. No caso da rede de transporte, a centralidade de grau de um vértice *Bus Stop* é proporcional ao número de paradas de ônibus no respectivo ponto para o caso dinâmico. Isso significa que pontos com elevada centralidade de grau podem corresponder a locais de potencial congestionamento ou de formação de comboios decorrente de uma estratégia de atendimento de demanda elevada. Para o caso estático, a centralidade de grau de um vértice *Bus Stop* é proporcional ao número de linhas de ônibus que passam pelo vértice, já que apenas arestas para outros vértices *Bus Stop* são consideradas.

O *page rank* foi concebido com o intuito de ranquear páginas (*web sites*) relevantes da *World Wide Web*. A ideia é que o fluxo de navegação dos usuários defina a relevância das páginas ao reforçar suas interconexões (*hyperlinks*). O algoritmo PageRank [Brin and Page 1998] traduz a ideia intuitiva de que usuários em geral tendem a acessar páginas e navegar pela WWW usando *hiperlinks* até uma certa profundidade. Assim, páginas bem posicionadas no fluxo de navegação são relevantes, pois tendem a ser muito visitadas pelos usuários. No contexto das redes de transporte, um ponto de ônibus com elevado *page rank* pode indicar conexões com outros pontos importantes. Por exemplo, ligações diretas entre terminais de ônibus no caso estático.

O **caminho mínimo** entre dois vértices é o caminho com o menor número de arestas, no caso de arestas não ponderadas. Caso as arestas tenham peso, deve-se considerar a soma dos pesos das arestas do caminho de um vértice a outro. A determinação de caminhos mínimos (considerando todos os pares de vértices de uma rede) é informação significativa para o planejamento urbano e a operação de um sistema de transporte, pois pode induzir no fluxo de pessoas na cidade. Algoritmos de caminho mínimo identificam rotas de conexão entre dois pontos da cidade de forma a minimizar o tempo de percurso necessário no transporte público [Martinek and Zemlicka 2009, Larson and Odoni 1981]. Uma vez conhecidos os caminhos mínimos de todos os vértices para todos os outros da rede, define-se o **diâmetro da rede** como o caminho mínimo mais longo da rede.

Introduzido por [Freeman 1977] como uma medida para quantificar o controle de um ser humano sobre a comunicação entre outros seres humanos em uma rede social, a **centralidade de intermediação** (*betweenness centrality*) quantifica o número de

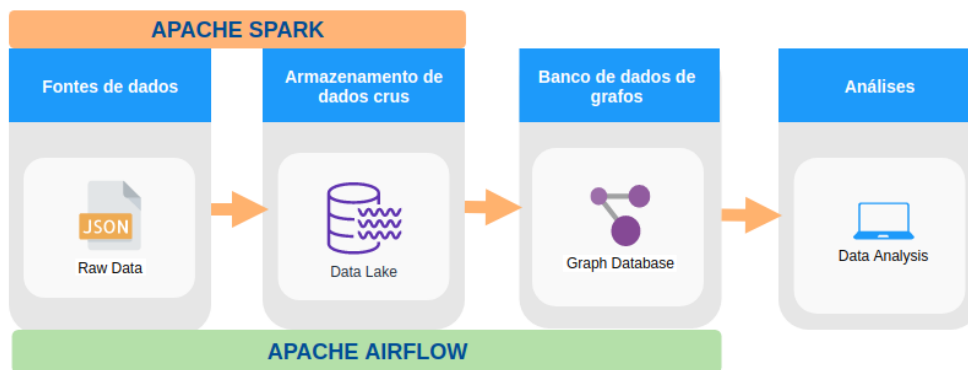


Figura 2. Fluxos de transformação da fonte de dados para o banco de grafo.

caminhos que passam por um determinado vértice. Em outras palavras, a remoção de vértices com elevada centralidade de intermediação podem degradar ou mesmo interromper o fluxo de informação em uma rede. No contexto de sistemas de transporte, estes vértices são potenciais pontos de estrangulamento do sistema. Essa dependência pode ser temporal (ocorre em um intervalo de tempo específico) ou espacial (depende da posição espacial do ponto de ônibus), de acordo com o contexto da análise.

5. Plataforma Computacional

A plataforma computacional desenvolvida gera um banco de dados de grafo no Neo4j a partir do *dataset* disponibilizado pela operação do transporte coletivo de Curitiba.

5.1. Dataset

O portal de dados abertos de Curitiba fornece diariamente um *dataset* em formato JSON (JavaScript Object Notation) com informações sobre o transporte coletivo da cidade. Conforme mencionado anteriormente, o transporte opera em média com uma frota de 1.410 ônibus que atende cerca de 1.389.731 passageiros por dia com 251 linhas de ônibus, 329 estações e 21 terminais.

A empresa URBS (gestora da Rede Integrada de Transporte Coletivo de Curitiba) define um conjunto de tabelas que contêm informações referentes às linhas de ônibus existentes, seus pontos de parada (que podem atender múltiplas linhas), os itinerários das linhas, os veículos que percorrem essas linhas, o percurso georeferenciado das linhas e o relacionamento entre essas informações. A descrição destas tabelas pode ser encontrada no dicionário de dados disponível no repositório de dados abertos.

5.2. Transformação do *dataset* para a base de dados de grafo do Neo4j

A Figura 2 mostra o fluxo de transformação dos dados para a construção do modelo de grafo no Neo4j a partir do *dataset* disponibilizado. Diversos fluxos foram implementados na linguagem Python usando as ferramentas Apache Airflow⁴ e Apache Spark⁵.

⁴<https://airflow.apache.org/>

⁵<https://spark.apache.org/>

A primeira etapa consiste no carregamento dos dados brutos (em formato JSON) para uma área temporária de armazenamento (ou *data lake*). Os arquivos JSON são transformados em arquivos parquet⁶ [Boufeia et al. 2017] utilizando o Apache Spark. Arquivos parquet são formatos de armazenamento em colunas, otimizados para compressão e rápida recuperação de dados. Na sequência, o conteúdo do *data lake* é processado - também usando o Apache Spark - para gerar arquivos no formato CSV com os vértices e arestas do modelo de grafo apresentado na Seção 3.

Embora vários fluxos de transformação de dados tenham sido implementados, detalha-se a seguir as transformações dos dados que criam o vértice *Stop* e a aresta *EVENT_STOP* do grafo. Estas estruturas modelam a dinâmica de movimentação dos ônibus e justificam o uso de um grafo variante no tempo.

O vértice *Stop* é criado a partir dos dados de geolocalização dos ônibus nos respectivos instantes de tempo (ver descrição da tabela *VEICULOS* no dicionário de dados do repositório). Com essa informação, calcula-se a distância percorrida, o tempo decorrido entre posicionamentos consecutivos e a velocidade média em km/h. Se a velocidade for menor do que 15 km/h, assume-se que houve uma parada do ônibus neste intervalo de tempo e um vértice *Stop* é gerado no grafo com os respectivos atributos.

A aresta *EVENT_STOP* é criada a partir dos vértices *Stop* previamente obtidos das linhas dos ônibus, dos pontos de ônibus e da tabela horária dos ônibus nas linhas (ver descrição das tabelas *LINHAS*, *PONTOS_LINHA* e *TABELA_VEICULO*, respectivamente, no dicionário de dados do repositório). Se uma parada (vértice *Stop*) ocorrer a menos de 20 m de um ponto de ônibus (vértice *Bus Stop*), considera-se que tal parada ocorreu em um ponto de ônibus e, portanto, cria-se uma aresta *EVENT_STOP* para conectar os vértices *Stop* e *Bus Stop* no grafo. Uma vez construído o banco de dados de grafo para o transporte, consultas e análises de redes complexas podem ser realizadas a partir do Neo4j (*Data Analysis* na Figura 2).

6. Resultados

Os resultados foram obtidos para o período de 01/01/2019 a 30/06/2019, tendo sido coletados 98,6 Gigabytes de dados, com 753.977.108 registros de geolocalização dos 1.835 ônibus em operação. Este volume de dados gerou, após a transformação para o banco de dados de grafo do Neo4j, 88.860.135 vértices e 200.075.643 arestas.

A Figura 3 mostra a centralidade de grau considerando apenas os pontos de ônibus (rede estática, ou seja, sem movimentação de ônibus). Os pontos 150331, 110026 e 150332 são os pontos de maior centralidade de grau, ou seja, conectam um número maior de pontos de ônibus devido a um maior número de linhas que passam por estes pontos. Em Curitiba, enquanto que o segundo ponto localiza-se na região central da cidade, os dois outros pontos (150331 e 150332) localizam-se na região sul da cidade. Em todos os casos, são pontos para onde convergem várias linhas de ônibus antes de chegarem a terminais importantes da cidade (terminal Pinheirinho ao sul, nos caso dos pontos 150331 e 150332, e praça Rui Barbosa no centro, no caso do ponto 110026).

A Figura 4 mostra a centralidade de grau considerando as paradas de ônibus nos pontos de ônibus ao longo do dia (rede dinâmica, ou seja, com movimentação de ônibus).

⁶<https://parquet.apache.org/>

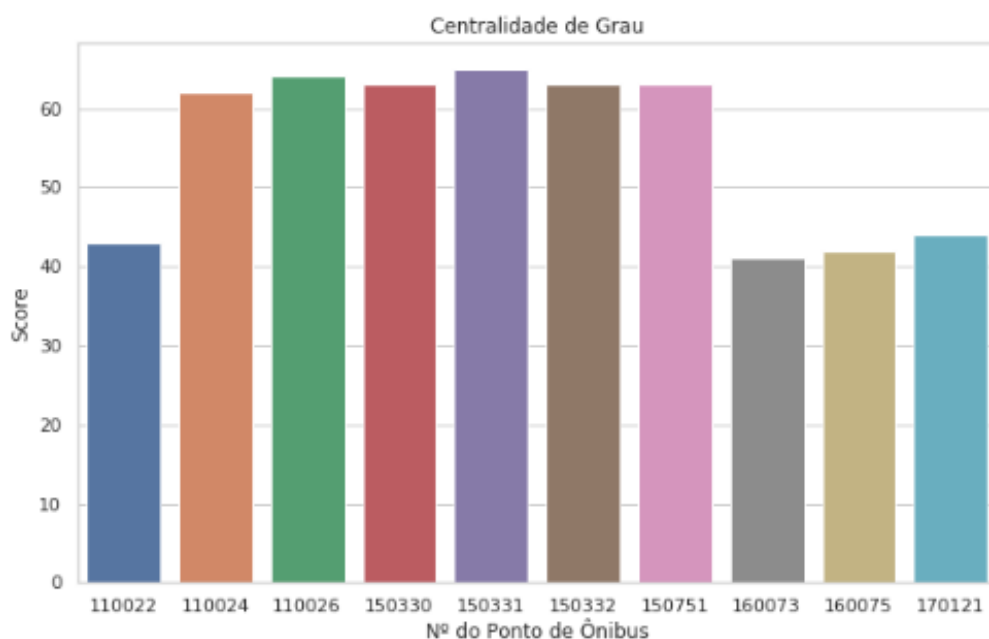


Figura 3. Centralidade de grau para pontos de ônibus (dez maiores) sem movimentação de ônibus (rede estática).

São mostrados os resultados para os três pontos de maior centralidade de grau da rede estática (150331, 110026 e 150332). Nota-se uma maior concentração no período da manhã e meio do dia, com diminuição gradual até o final da noite. Ao mesmo tempo, o ponto 150331 atende um elevado número de ônibus no período da manhã e no final da tarde. Essa última característica ocorre em Curitiba por conta do encerramento de atividades em empresas (trabalhadores retornando a suas casas) e estudantes de ensino noturno se deslocando para instituições de ensino.

A Figura 5 mostra o *page rank* considerando apenas a rede estática (pontos de ônibus). Os pontos 150331, 110026 e 110016 são os pontos de maior *page rank*. Estes pontos se localizam, respectivamente, na região sul da cidade (na saída do terminal Pinheirinho), na região que conecta o centro a região nordeste da cidade (na direção de terminais Cabral, Boa Vista e Santa Cândida), nas proximidades do centro da cidade. Todos esses pontos conectam várias linhas que se afastam (pontos 150331 e 110126) ou se aproximam (ponto 11016) de regiões de grande concentração de pontos finais de linhas de ônibus. Ou seja, a importância destes pontos se deve a ligações com outros pontos importantes (pontos ou terminais com grande concentração de linhas).

As medidas de comprimento das linhas da rede estática de transporte resultaram em 40 m, 38,14 km e 9,43 km para os comprimentos mínimo, máximo e médio (com desvio padrão de 6,5 km) das linhas de ônibus. Já as medidas baseadas em caminhos mínimos entre pontos da rede resultaram em um valor mínimo de 3,6 m, diâmetro da rede (caminho mínimo mais longo) de 55 km e um percurso médio de 14,04 km. Estes caminhos mínimos foram obtidos através dos atributos de distância (*DISTANCE*) das arestas *NEXT_STOP*, gerando uma rede ponderada do sistema de transporte de Curitiba. Note que esse percurso médio pode ou não ser factível com troca de ônibus (baldeação), dependendo da origem e destino do passageiro.

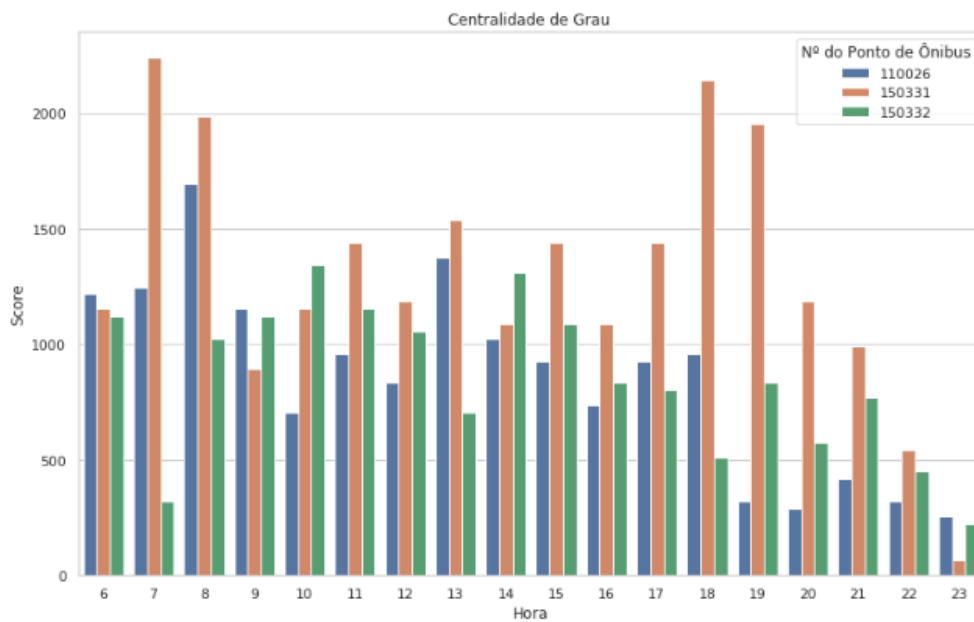


Figura 4. Centralidade de grau para pontos de ônibus (três maiores) com movimentação de ônibus ao longo do dia (rede dinâmica)

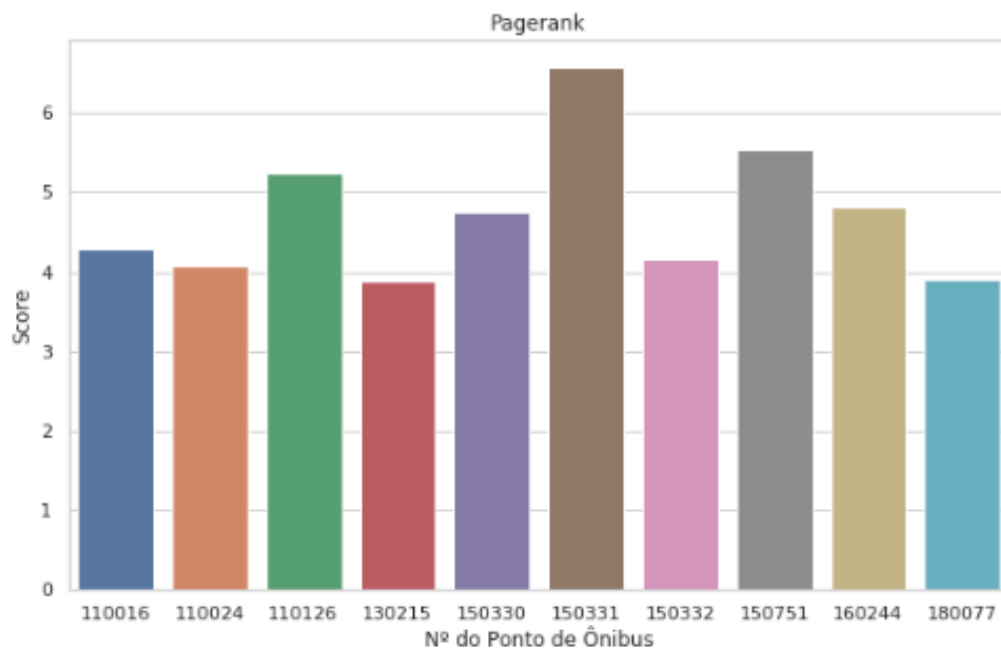


Figura 5. Page rank para pontos de ônibus (dez maiores) sem movimentação de ônibus (rede estática).

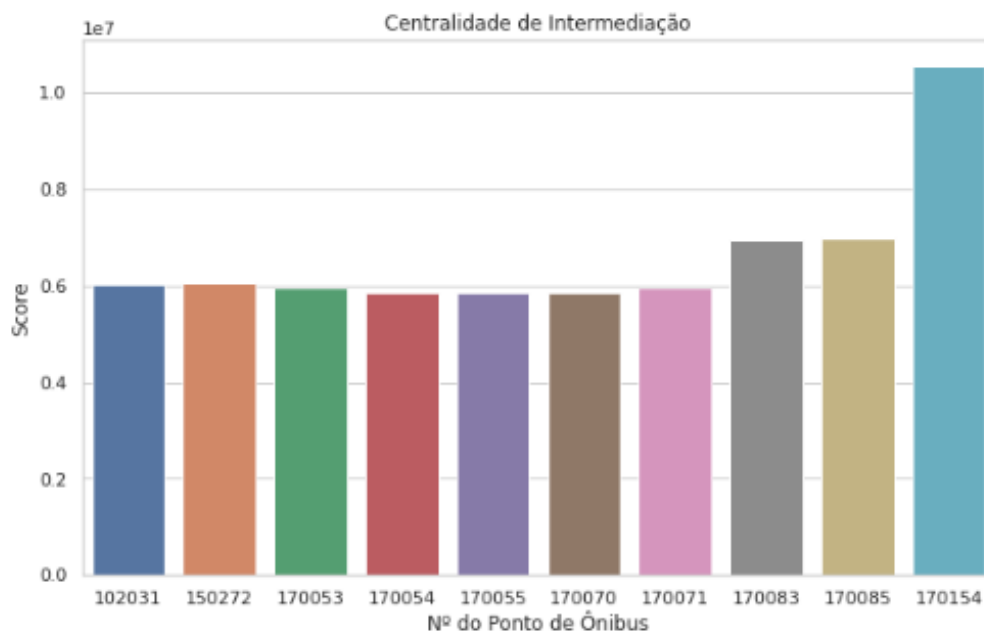


Figura 6. Centralidade de intermediação para pontos de ônibus (dez maiores) sem movimentação de ônibus (rede estática).

A Figura 6 mostra a centralidade de intermediação considerando apenas a rede estática (pontos de ônibus). Os pontos 170154, 170085 e 170083 são os pontos de maior centralidade de intermediação. Estes pontos localizam-se na vizinhança do terminal Pinheirinho, localizado na parte sul de Curitiba, comportando-se como um *hub* por onde passam percursos mais curtos entre pontos de ônibus da rede. Note também que esses percursos podem ou não ser factíveis com trocas de ônibus.

7. Conclusão

Este trabalho propôs uma plataforma computacional para construção automática de um banco de dados de grafo a partir de um repositório de dados abertos com informações da operação diária do transporte de Curitiba. O modelo utilizado corresponde a um grafo variante no tempo (TVG) disponível na literatura com poucas modificações. A plataforma computacional transforma dados brutos do repositório para a base de dados de grafos do Neo4j. Diversos algoritmos de análise de redes complexas podem então ser aplicados. Resultados de centralidade de grau, *page rank* e centralidade de intermediação permitiram identificar pontos de ônibus relevantes na rede, seja pela concentração de linhas (rede estática, sem movimentação de ônibus) ou paradas de ônibus (rede dinâmica, considerando a movimentação de ônibus), além de características dos caminhos da rede. Estas medidas são úteis para o planejamento do transporte, pois permitem avaliar a área coberta e a frequência e regularidade dos serviços. Esta última é particularmente importante no estabelecimento da integração temporal das linhas, quando uma tarifa é válida por um determinado período de tempo em toda ou parte da rede de transporte. Embora os resultados de análise tenham sido apresentados para o sistema de transporte de Curitiba, o modelo pode ser adaptado para o transporte de ônibus de outras cidades. Como trabalhos futuros pode-se incluir o tratamento de regiões geográficas para análises de bairros ou outros agrupamentos georreferenciados. Da mesma forma, a inclusão de informações de

arruamento baseado no *OpenStreetMap* permitirá relacionar a operação do transporte às vias de circulação da cidade.

Referências

- Boufeia, K., Finkers, R., Kaauwen, M., Kramer, M., and Athanasiadis, I. (2017). Managing variant calling files the big data way: Using HDFS and Apache parquet. In *Proc. of the Fourth IEEE/ACM Int. Conf. on Big Data Computing, Applications and Technologies*, pages 219–226.
- Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117.
- Casteigts, A., Flocchini, P., Quattrociocchi, W., and Santoro, N. (2012). Time-varying graphs and dynamic networks. *Int. Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408.
- Cattuto, C., Quaggiotto, M., Panisson, A., and Averbuch, A. (2013). Time-varying social networks in a graph database: A Neo4j use case. In *First Int. Workshop on Graph Data Management Experiences and Systems, GRADES '13*, NY, USA. ACM.
- Curzel, J. L., Lüders, R., Fonseca, K. V. O., and Rosa, M. O. (2019). Temporal performance analysis of bus transportation using link streams. *Math. Problems in Eng.*, 2019:6139379.
- Ferraz, A. C. P. and Torres, I. G. E. (2004). *Transporte público urbano*. Rima Editora.
- Flocchini, P., Mans, B., and Santoro, N. (2009). Exploration of periodically varying graphs. In *Proc. of 20th Int. Symp. on Algo. and Comp. (ISAAC)*, pages 534–543.
- Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41.
- Freeman, L. C. (1979). Centrality in social networks: I. conceptual clarification. *Social Networks 1*, page 215–239.
- Gallotti, R. and Barthelemy, M. (2015). The multilayer temporal network of public transport in Great Britain. *Sci. Data*, 2:1–8.
- Holme, P. and Saramäki, J. (2012). Temporal networks. *Physics Reports*, 519(3):97 – 125. Temporal Networks.
- Izawa, M., Oliveira, F., Cajueiro, D., and Mello, B. (2017). Pendular behavior of public transportation networks. *Phys. Rev. E*, 96:012309.
- Larson, R. C. and Odoni, A. R. (1981). *Urban Operations Research*. Prentice-Hall.
- Maduako, I., Wachowicz, M., and Hanson, T. (2019). STVG: an evolutionary graph framework for analyzing fast-evolving networks. *J Big Data*, 6(55):1–24.
- Martinek, V. and Zemlicka, M. (2009). Speeding up shortest path search in public transport networks. In *CEUR Workshop Proceedings*, volume 471.
- Nicosia, V., Tang, J., Musolesi, M., Russo, G., Mascolo, C., and Latora, V. (2012). Components in time-varying graphs. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(2):023101.

- Reed, D. (2015). How Curitiba's BRT stations sparked a transport revolution – a history of cities in 50 buildings, day 43. *The Guardian*.
- Rodriguez, M. A. and Neubauer, P. (2010). Constructions from dots and lines. *Bulletin of the American Society for Information Science and Technology*, 36:35–41.
- Sienkiewicz, J. and Hołyst, J. (2005). Statistical analysis of 22 public transport networks in Poland. *Phys. Rev. E*, 72:046127.
- Tang, J., Musolesi, M., Mascolo, C., and Latora, V. (2010a). Characterising temporal distance and reachability in mobile and online social networks. *ACM Computer Communication Review*, 40:118–124.
- Tang, J., Scellato, S., Musolesi, M., Mascolo, C., and Latora, V. (2010b). Small-world behavior in time-varying graphs. *Phys. Rev. E*, 81:055101.
- URBS (2019). URBS em números. <https://www.urbs.curitiba.pr.gov.br/transporte/rede-integrada-de-transporte> acesso em 09/03/2019.
- Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y., and Wilkins, D. (2010). A comparison of a graph database and a relational database: A data provenance perspective. In *Proc. of the 48th Annual Southeast Regional Conf.*, ACM SE '10, NY, USA. ACM.
- Wessel, N., Allen, J., and Farber, S. (2017). Constructing a routable retrospective transit timetable from a real-time vehicle location feed and GTFS. *J. Transp. Geo.*, 62:92–97.
- Xu, Q., Zu, Z., Xu, Z., Zhang, W., and Zheng, T. (2013). Space p-based empirical research on public transport complex networks in 330 cities of China. *J. Transp. Sys Eng. & IT*, 13:193–198.

Apêndice - Tabelas com informações do modelo de grafo e dos dados brutos

Tabela 1. Atributos dos vértices *Trip*: identificam os sentidos das linhas.

| Atributo | Descrição |
|----------|------------------|
| line_way | sentido da linha |

Tabela 2. Atributos dos vértices *Vehicle*: identificam os veículos da frota.

| Atributo | Descrição |
|----------|-------------------|
| vehicle | código do veículo |

Tabela 3. Atributos dos vértices *Schedule*: identificam os itinerários das linhas.

| Atributo | Descrição |
|-------------|---|
| start_time | horário programado de início da viagem |
| line_code | código da linha |
| time_table | tabela horária |
| start_point | código do ponto de partida da viagem |
| end_time | horário programado de término da viagem |
| line_way | sentido da linha |
| vehicle | veículo programado para a viagem |

Tabela 4. Atributos dos vértices *Bus Stop*: identificam os pontos de ônibus.

| Atributo | Descrição |
|-----------|--------------------------------|
| name | nome do ponto de ônibus |
| number | número do ponto de ônibus |
| geometry | ponto de ônibus em formato WKT |
| type | tipo do ponto de ônibus |
| latitude | latitude do ponto |
| longitude | longitude do ponto |

Tabela 5. Atributos dos vértices *Line*: identificam a linha de ônibus.

| Atributo | Descrição |
|-----------|--|
| name | nome da linha |
| card_only | indicador se a linha aceita somente cartão |
| line_code | código da linha |
| color | cor da linha |
| category | categoria da linha |

Tabela 6. Atributos dos vértices *Stop*: identificam as paradas dos ônibus.

| Atributo | Descrição |
|-----------------|--|
| line_code | código da linha |
| latitude | latitude da parada |
| longitude | longitude da parada |
| event_timestamp | <i>timestamp</i> do evento de parada do ônibus |
| event_time | horário do evento de parada do ônibus |
| vehicle | veículo que parou |

Tabela 7. Descrição das arestas: significado das conexões e atributos.

| Rótulo | Descrição |
|-----------------------|--|
| EXISTS_LINE | agrupa as informações da linha de ônibus por dia |
| EXISTS_STOP | agrupa as informações de paradas do veículo por hora |
| HAS_TRIP | conecta a linha a várias viagens de ônibus |
| HAS_SCHEDULED_AT | conecta a viagem a um horário programado |
| HAS_VEHICLE_SCHEDULED | conecta o horário programado ao veículo da viagem |
| HAS_STOPPED | conecta o ônibus a paradas (veloc. < 15 km/h) |
| STARTS_ON_POINT | conecta a viagem ao ponto de ônibus inicial |
| HAS_BUST_STOP | conecta a viagem aos pontos de ônibus intermediários |
| ENDS_ON_POINT | conecta a viagem ao ponto de ônibus final |
| NEXT_STOP | conecta pontos de ônibus consecutivos da viagem: distance: distância (m) entre pontos de ônibus line_name: nome da linha line_way: sentido da linha |
| MOVED_TO | conecta a sequência de paradas do veículo: delta_time: tempo (s) entre paradas delta_distance: distância (m) entre paradas delta_velocity: veloc. média (m/s) entre paradas |
| EVENT_STOP | conecta parada do ônibus a um ponto (a cerca de 20 m): line_way: sentido da linha antes da parada |