

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/317413603>

A teoria dos grafos e o algoritmo Dijkstra aplicados à análise de redes de transporte e à construção de matrizes de deslocamento em rede

Conference Paper · June 2017

CITATIONS

0

READS

729

3 authors, including:



Joao Bonett

University of São Paulo

1 PUBLICATION 0 CITATIONS

[SEE PROFILE](#)



Thilo Koch

University of São Paulo

5 PUBLICATIONS 5 CITATIONS

[SEE PROFILE](#)

A teoria dos grafos e o algoritmo Dijkstra aplicados à análise de redes de transporte e à construção de matrizes de deslocamento em rede.

Autores: João Bonett (doutorando)¹; Thilo Koch (doutorando)²; Dr. Nuno de Azevedo Fonseca³.

1. Universidade de São Paulo – USP, Faculdade de Arquitetura e Urbanismo – FAU. Rua do Lago, 876, Butantã, São Paulo – SP. Tel.: (11) 30914548. E-mail: jn.bonett@usp.br.
2. Universidade de São Paulo – USP, Instituto de Matemática e Estatística – IME. Rua do Matão, 1010, Butantã, São Paulo – SP. Tel.: (11) 3091-6134. E-mail: tiko@ime.usp.br.
3. Universidade de São Paulo – USP, Faculdade de Arquitetura e Urbanismo – FAU. Rua do Lago, 876, Butantã, São Paulo – SP. Tel.: (11) 30914548. E-mail: nuno16@uol.com.br.

Sinopse

Este trabalho reflete a elaboração e a experimentação de um programa computacional que faz uso da chamada teoria dos grafos e do algoritmo de Dijkstra para análise de redes de transporte. Especificamente, trata-se de script desenvolvido para o ambiente de geoprocessamento Qgis, com base em ferramentas e bibliotecas de funções livres. O que se procura é calcular uma matriz de distâncias e de tempos de percurso entre nós de uma rede de transporte – no caso, de metrô. Os resultados, os *potenciais de mobilidade*, são mapeados de modo que possam ser úteis a análises espaciais e a comparações entre diferentes redes, assim como, enquanto elemento para construção de cenários e para simulações de determinados aspectos da dinâmica urbana. Em essência, o artigo constitui um esforço de discutir a importância do software livre no desenvolvimento e na divulgação de instrumentos de análise como o aqui abordado.

Palavras-chaves: Matriz de tempo e custo; Algoritmo de Dijkstra; Análise de redes.

Introdução

Métodos de análise de redes de transporte estão inscritos em uma extensa e certamente não esgotada discussão. São várias as técnicas adotadas no estudo dessas redes, que recorrem a desde comparações históricas até procedimentos analíticos ou estimativos de cálculo. Não raro, esses métodos envolvem softwares e modelos “proprietários” de operação extremamente complexa. Além disso, sem contar os custos de processamento, implicam pesquisas muito caras, a ponto de sua efetivação ser viabilizada somente por meio de grandes orçamentos estatais e dispendiosas infraestruturas computacionais.

Em face disso, a conjuntura dos softwares livres e dos dados abertos – acrescentada do chamada computação paralela em nuvem, ou “cloud computing” – apresenta-se como um novo ponto de apoio a essa realidade. Tais instrumentos, devido à essência de sua proposta, contribuem para a democratização de técnicas e de informações, corroborando os tão evocados conceitos de participação e de governo aberto. Com efeito, este trabalho procura se enquadrar no âmbito do Encontro com uma proposta de discutir ferramentas para análise de redes de transporte, usando soluções livres.

Redes de transporte são elementos de relevância na definição de diretrizes de planejamento urbano e, nesse sentido, métodos que possibilitam ampliar o restrito domínio técnico em sua concepção respondem à dimensão participativa do planejamento contemporâneo. Por meio de um experimento com dados e softwares livres, este artigo busca lançar ao debate as potencialidades dessas ferramentas. Tal experimento consiste na aplicação de rudimentos de teoria dos grafos e do algoritmo de Edsger Dijkstra para análise de propostas de rede de metrô concebidas para São Paulo. Para isso foi codificado um script internamente ao software Qgis, usando noções elementares de programação e bibliotecas de funções elaboradas especificamente para análise de grafos e redes. Os resultados apresentam

mapas que servem a comparações gráficas entre 10 desenhos propostos para a rede de metrô da cidade de São Paulo, projetados entre as décadas de 1960 e 2010.

Em síntese, o que se tentará mostrar é como, além da extensão das redes sobre o território, sua configuração corresponde a uma específica forma de estruturação do espaço urbano. A depender do desenho da rede, um determinado ponto da cidade pode ter associado a si um custo relativamente maior ou menor para acessar os outros pontos dessa mesma cidade. Isso tem reflexos sobre distribuição efetiva de usos e densidades, assim como, sobre a própria diferenciação, ou grau de homogeneização, do espaço urbano. O exercício aqui proposto procura calcular e mapear esses custos, com base em que é possível comparar não apenas os desempenhos das redes entre si, mas também sua correspondência a outras diretrizes urbanísticas.

Diagnóstico

Há séculos a chamada teoria dos grafos tem sido usada para exploração de problemas matemáticos. Nas últimas décadas, no entanto, a computação e novas concepções impulsionaram sua investigação por meio de algoritmos. Suas aplicações são diversas e incluem usos nos mais distintos campos de pesquisa e desenvolvimento (Sedgewick, et al., 2012). Um grafo pode ser considerado um modelo de dados que representa a relação, ou conectividade, entre itens de um determinado conjunto. Em uma definição formal, um grafo nada mais é do que uma composição de “vértices” e “arestas”. Um vértice é o ponto que conecta duas ou mais arestas, que, por sua vez, podem ou não ter uma direção especificada e também podem ou não ter pesos a elas associados, de modo a representar custos, como tempos de travessia. Assim, caminhos são constituídos pela sequência de vértices conectados por arestas, sendo o tamanho desse caminho a soma das arestas ou dos pesos a ele associados. Adicionalmente, caminhos que têm ao menos uma aresta e cujo primeiro vértice é o mesmo que o último definem grafos cíclicos. Grafos que não contêm ciclos são chamados de árvores. Dessa maneira, é possível modelar a rede de transporte de uma cidade por meio de um grafo. Nesse caso, os vértices representam as estações ou nós da rede, e as arestas representam as linhas ou vias do sistema, que normalmente são bidirecionais. Isto é, a rede de transportes pode ser representada como um grafo não direcionado com pesos positivos contendo ciclos não negativos.

Esse modelo pode ser elaborado diretamente por um usuário ou, o que seria mais plausível, com base em informações pré-existentes, como arquivos vetoriais armazenados em Sistema de Informações Geográficas (SIG). Em um SIG, os vértices são coordenadas que conectam polilinhas, ou seja, vias ou linhas de transporte a cujos comprimentos podem ser associados pesos que representam a velocidade dos veículos. Trata-se de um modelo de dados vetorial simples que é convertido em um grafo por meio de um “padrão”, programa que, no caso, constrói, automaticamente, um modelo de dados com base em outro. Este programa, mais precisamente, constrói um grafo a partir de uma imagem vetorial. Uma vez que ambos os modelos representam o mesmo objeto, com o mesmo nível de detalhe, a vantagem desse método é a não duplicação de tarefas na produção de dados, além da permanência de um raciocínio espacial – de desenho, pode-se dizer – em todas as etapas de concepção e teste da rede, o que facilita inclusive, o cotejamento dos resultados a outros dados georreferenciados. Em análise comparada de redes de transporte, a teoria dos grafos muitas vezes é aplicada por meio do cálculo de índices, como *complexidade*, *conectividade* e *número de loops* (Isoda, 2013). Tais índices tratam, basicamente, das relações entre o número de vértices e de arestas, entre o número de arestas e de “arestas possíveis”, ou simplesmente do número de ciclos – loops. Embora esses índices sejam uma maneira de sintetizar e de quantificar as características das redes, não são concebidos para uma análise espacial.

A maior parte das pesquisas que se valem de grafos para análise espacial de redes de transporte buscam, fundamentalmente, resolver o chamado problema do caminho mais

curto. Em uma rede de vértices relativamente numerosos, como é uma rede real de transportes, o caminho entre um ponto e outro pode assumir múltiplas possibilidades. Assim, para que os custos desse caminho, em tempo ou distância, sejam razoáveis, é preciso que esse represente, entre todas as opções possíveis, o caminho mais curto. A depender da extensão e da complexidade da rede, assim como do número de análises a proceder, essa tarefa torna-se inexecutável sem o auxílio de algoritmos. Um dos algoritmos mais eficazes para resolver o problema do caminho mais curto entre dois vértices é o algoritmo de Dijkstra. Esse algoritmo foi concebido pelo matemático holandês Edsger Dijkstra na década de 1970 e aplica-se, especificamente, a um grafo que não contém ciclos negativos. Outros algoritmos para problemas parecidos existem e podem ser explorados em futuros trabalhos, como, por exemplo, o algoritmo de Bellman-Ford ou as próprias modificações do algoritmo de Dijkstra, que usam o peso das arestas relacionado a localizações geográficas dos vértices. No entanto, por resolver também o problema de encontrar o caminho mais curto para todos os vértices partindo de um, isto é, o problema da construção de uma “árvore de caminhos mais curtos”, o algoritmo de Dijkstra será adotado neste experimento.

O uso de Dijkstra e a construção de árvores de caminhos mais curtos é um método recorrente na elaboração dos chamados mapas de área de alcance, ou “areas of availability” (Palmier, 2015). Tais mapas especializam os custos de uma árvore de caminhos mais curtos, de modo que regiões próximas ou mais acessíveis à raiz da árvore, ao vértice de origem, apresentam custos menores, representados, normalmente, por cores mais claras – o que seria inverso para as regiões mais inacessíveis. Em geral, os mapas de área de alcance são usados para representar ou dimensionar a área de influência de equipamentos ou serviços públicos, mas, apesar de refletirem características da rede de transporte e terem um resultado gráfico, não são a ferramenta mais adequada para comparação ou estudo de redes. Para tanto, é necessário que os custos de acesso de cada nó da rede a todos os outros sejam computados em uma matriz de distâncias ou de tempos. Com base nessa matriz, é possível calcular e especializar os custos relativos associados à mobilidade e à acessibilidade de cada nó, ou vértice, da rede. Configura-se, assim, um mapa do que pode ser chamado de “potencial de mobilidade”. Com a preparação de um script simples, é possível aplicar o algoritmo de Dijkstra para a construção de tal matriz, o que, para a análise de uma rede metrô, como propõe este trabalho, é algo que pode ser feito a um relativamente baixo custo de processamento. Isso se deve à pequena quantidade de vértices a serem calculados – cerca de 1000 nós, considerando os vértices associados às “curvas” da rede.

O algoritmo de Dijkstra, no entanto, pertence a uma classe de algoritmos extremamente custosa em termos de processamento. Sendo V o conjunto dos vértices do grafo, E o conjunto das arestas e $| \cdot |$ o número de elementos de um conjunto, a execução do algoritmo de Dijkstra apresenta custos máximos proporcionais a $|E|$ vezes $\log |V|$. Como para elaborar a matriz de distâncias, ou de tempos, é preciso rodar o algoritmo para todos os vértices da rede, isso leva um custo proporcional a $|V||E| \log |V|$. Na prática, esses custos podem levar a tempos de execução longos dependendo do desempenho da plataforma computacional e, principalmente, da grandeza do problema – número de vértices e de arestas. No caso da rede viária completa da Região Metropolitana de São Paulo, a quantidade de vértices ultrapassa o número de 1.14 bilhões, o que torna uma análise completa de todo o sistema de transporte da Região Metropolitana algo absolutamente inviável para ser realizada em um computador convencional. Mesmo redes mais recortadas, como somente trilhos e ônibus, continuariam expressivamente dispendiosas para serem analisadas.

Tal dificuldade pode ser superada por meio da chamada computação paralela. A velocidade de computadores é restrita à frequência máxima das CPUs, da memória e do tempo de transferência dos dados entre a CPU e a memória. Assim, a aceleração do processamento em máquinas convencionais, a níveis muito significativos, só pode ser alcançada com a paralelização de algoritmos, ou seja, com a execução simultânea de tarefas de um

programa em CPUs diferentes. O limite a paralelização se encontra na dependência das instruções de uma ordem sequencial, isto é, certas instruções precisam do resultado de outras, de modo que necessitam ser executadas sempre em sequência (Ben-Ari, 2006). Para o problema em questão, lembra-se que o algoritmo Dijkstra é rodado $|V|$ vezes e que cada uma de suas execuções é independente da outra, o que traduz um grande potencial para sua paralelização. Além disso, a técnica de computação em “nuvem” possibilita o aluguel, a preços muito acessíveis, de grandes capacidades de processamento com o uso da paralelização. Dessa maneira, a análise de grandes redes poderia se tornar algo acessível para pesquisas e propostas de orçamentos convencionais. Neste experimento, uma tentativa de paralelização foi empreendida usando um computador com 4 núcleos de processamento e também a técnica da computação em nuvem. Os resultados serão brevemente discutidos adiante.

Proposições e resultados

Código e ambiente de processamento

Todo este trabalho está baseado na operação do SIG desktop Qgis, software livre distribuído de acordo com a licença GPL 2. Seu uso possibilita não apenas a manipulação de dados georreferenciados e representação cartográfica, mas também a programação de scripts, em linguagem Python. Além disso, os conteúdos elaborados pela comunidade de usuários e desenvolvedores incluem bibliotecas de funções específicas para análise de redes e grafos, o que traduz a possibilidade de recurso a uma extensa quantidade de trabalho pré-existente e coletivamente desenvolvido. O script para a realização do experimento foi programado na linguagem python por meio da plataforma interna ao programa, “terminal python”. Lançou-se mão, para tanto, de uma biblioteca para análise de redes (network analysis library), distribuída juntamente com o programa e cuja documentação encontra-se disponível no site oficial – qgis.org. Nessa biblioteca, o algoritmo de Dijkstra já está codificado. Além disso, outra importante biblioteca programada em Python, NetworkX, disponibiliza uma série de outros algoritmos para análise de grafos, representando um poderoso instrumento para propostas como a aqui desenvolvida.

A estrutura do script segue recortes de código disseminados na documentação oficial e na rede em geral, dividindo-se, basicamente, em 4 partes. Inicialmente, são preparados e declarados os dados e variáveis usadas para a construção do grafo, em essência: um desenho vetorial da rede. Seguem as instruções para a construção do grafo, a partir desse desenho. Em terceiro lugar, o algoritmo Dijkstra é rodado para cada vértice da rede e os custos são computados em uma tabela que, finalmente, é exportada em um arquivo no formato csv (valores separados por vírgulas). Essa tabela consiste em uma matriz de distâncias, em que nas colunas e linhas estão representados os nós da rede e, nas células intermediárias, as distâncias correspondentes entre cada um desses nós. Assim, o resultado é uma matriz simétrica com diagonal nula, isto é, preenchida com zeros. Com auxílio de funções da biblioteca Numpy, para cálculo matricial, os custos, então, são agregados para cada nó da rede e posteriormente mapeados com base nas coordenadas dos mesmos nós.

Este método assume, a princípio, 2 simplificações: 1) a velocidade das linhas de metrô é invariável; 2) não foram consideradas estações, mas somente os nós da rede. Uma reformulação do código concebida recentemente resolve algumas dessas simplificações. O processo de construção do grafo no novo código transforma as poli-linhas, que representam as vias da rede, em arcos do grafo e a cujos pesos estão associadas as velocidades das vias, extraídas de um campo da tabela de atributos do arquivo vetorial. De tal modo, não apenas vias de velocidades diferentes podem ser comparadas como também modos distintos de transportes podem ser incorporados à análise. Isso, entretanto, e como já apontado, a depender do tamanho da rede e do número de nós, pode levar a consideráveis custos de processamento. Em vista disso, uma parte do código foi concebida para a

paralelização do algoritmo e um experimento com computação distribuída – em nuvem – foi realizado para testá-lo. Esse experimento será brevemente descrito abaixo.

Dados

Os esquemas das redes foram vetorizados com base em esboços e mapas. Apesar de simplificados, procuram retratar as nuances dos desenhos, respeitando curvaturas e distâncias. Essas imagens vetoriais, de extensão .shp, são convertidas em grafos automaticamente pelo script, sobre os quais, então, são feitos os cálculos de custo. Posteriormente, os dados são reinseridos no SIG por meio de georreferenciamento da camada CSV. As redes mapeadas dividem-se em três categorias: 1) as redes aqui chamadas **históricas**, concebidas em conhecidos estudos e planos, a saber: a rede do consórcio HDM, de 1967 e a rede do Plano Urbanístico Básico de São Paulo (PUB-1968). 2) redes **experimentais**, elaboradas como parte de trabalhos acadêmicos: rede de 1990, desenhada por Csaba Deák; rede Kyoto, integrante da dissertação de Marcos Kiyoto de Tange e Isoda, arguida em 2012; e a rede Zaidan, parte da dissertação de Moreno Zaidan, de 2014. 3) Por fim, as redes concebidas como alternativas de estudo para o **Plano Integrado de Transportes Urbanos da Região Metropolitana de São Paulo (PITU)**, elaboradas durante a década de 2000: rede aberta; rede central; rede densa; reformulação para 2025 e rede para 2030. Ainda, a rede atual de metrô, juntamente com a malha viária da Região Metropolitana de São Paulo, foram usadas no teste de paralelização. Adicionalmente, dados de extensão da mancha urbana foram incorporados com objetivos de cotejamento. O polígono da mancha urbana contínua foi vetorizado com base na classificação automática de uma cena registrada em junho de 2016 pelo satélite landsat 8, referente à banda 10 de seus sensores embarcados e cuja resolução espacial é 100 metros. Ainda, para estimação e generalização de valores, as zonas da pesquisa Origem-Destino (OD-2007) foram também consideradas no estudo.

Experimentos iniciais

É possível dizer que existem, basicamente, três arquétipos de rede: malha ortogonal, rede rádio-concêntrica e rede em arcos (Zaidan, 2014). Com o objetivo de uma demonstração preliminar, desenhos abstratos de redes foram testados. A figura 1, a seguir, sintetiza esses resultados. Todas as redes foram concebidas com 6 linhas e se estendem sobre um polígono medindo 50 km x 50 km de arestas. As cores próximas do azul representam custos menores de deslocamento a toda a extensão da rede, as próximas do vermelho, o contrário.

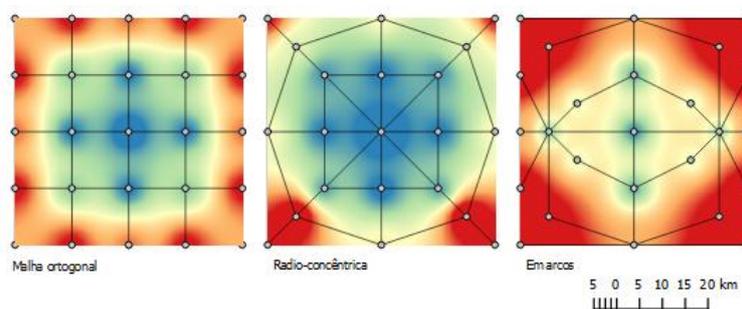


Fig. 1. Custos de deslocamento conforme modelo de rede. Fonte: autores.

As figuras das páginas abaixo ilustram os resultados do experimento feito com as dez redes de metrô descritas acima. Sucintamente, observa-se que, para além da extensão das redes, sua configuração representa distintas formas de estruturação do espaço urbano. Considerando que o centro de uma estrutura urbana é definido pela sua qualidade de maior mobilidade potencial, então, redes com nós mais uniformemente distribuídos, a exemplo da rede PUB, Aberta e Deák-1990, tendem a uma maior homogeneidade espacial em relação às oportunidades e aos custos de deslocamento. Ao contrário, redes muito densas quanto à

distribuição de suas ligações traduzem também uma concentração de investimentos e oportunidades. Em uma análise pormenorizada, seria possível perceber que em muitas dessas redes, o centro por elas estabelecido corresponde ao setor da metrópole que historicamente concentrou a maior quantidade de infraestruturas e, paradoxalmente, a menor demanda por transporte coletivo: os chamados quadrantes centro-oeste e sudoeste. É importante destacar, entretanto, que o esquema de cores abaixo representa apenas a proporção dos custos internamente a cada rede. Isto é, as mesmas cores de um diagrama não representam, necessariamente, os mesmos tempos de deslocamento em outro diagrama. Além disso, cabe ressaltar que os deslocamentos são calculados considerando apenas o modo metroviário. Uma análise do sistema de mobilidade da metrópole dependeria da incorporação de outros modos de transporte, o que, no entanto, só seria possível por meio do uso de uma significativamente maior capacidade de processamento, alcançável, por exemplo, mediante a paralelização do script.

Tendo isso em vista, um teste de paralelização do script foi realizado com a rede atual de metrô e a rede viária completa da Região Metropolitana de São Paulo. Esse conjunto de dados constitui um grafo com mais de 7.5 milhões de nós, analisado de acordo com dois métodos. No primeiro teste, o script foi rodado em uma mesma máquina com dois CPU. Em segundo teste, recorreu-se a um serviço web de computação paralela, ou computação em nuvem: google computing, com 24 CPU. Ressalta-se que parte do script, referente à construção do grafo, não é passível de paralelização devido à dependência das tarefas. O quadro abaixo resume os resultados do teste em relação aos tempos de execução das tarefas. Em sequência, a fig. 2 espacializa os custos de deslocamento entre os nós da rede, considerando para esse mapeando sintético, os centroides das zonas OD-2007.

Tabela comparativa de custos de processamento

Método	Num. cpu	RAM [GB]	Num. thread	Não paralelizável [s]	Paralelizável [s]	Total [s]
Serviço Web	1	3.6	1 xeon	170	2241	2435
	4	16	4 xeon	180	741	947
	8	30	8 xeon	169	467	659
	16	60	16 xeon	186	312	521
	24	32	24 xeon	171	179	362
Máquina física	2+ HT	16	1 core i5	373	2535	2928
	2+ HT	16	4 core i5	373	1100	1466

Fonte: autores.

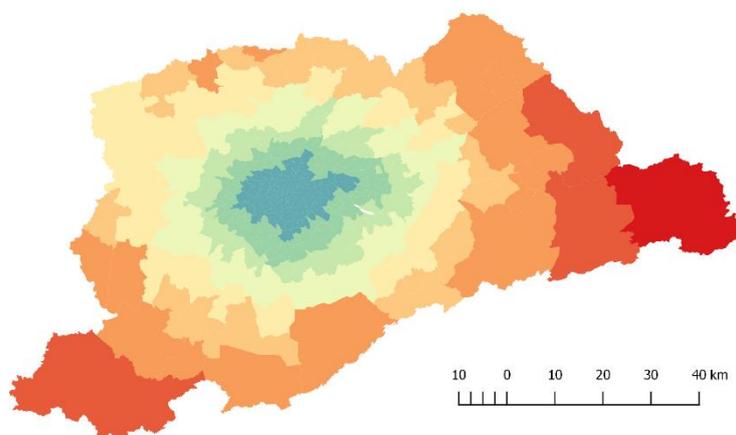
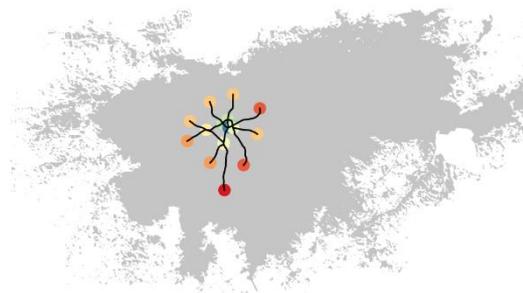
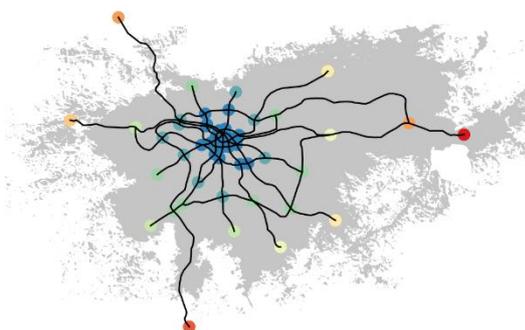


Fig. 2. Custos de deslocamento para rede atual de metrô e conjugada à rede viária existente.

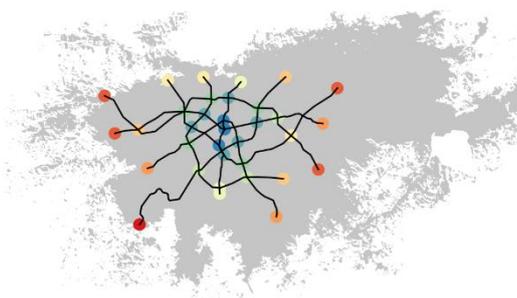
Redes históricas e experimentais.



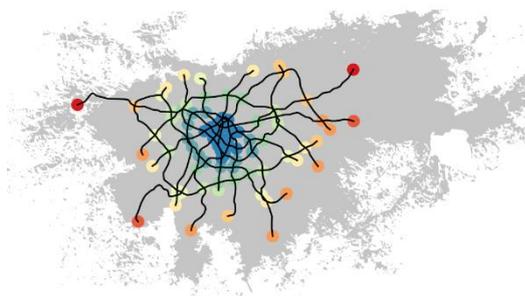
Rede HDM 1967
Extensão: 62 km.



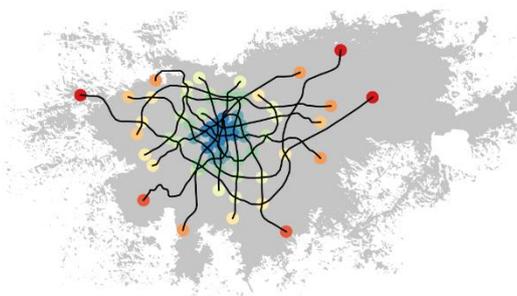
Rede PUB 1968
Extensão: 450 km.



Rede Deák 1990
Extensão: 260 km.



Rede Kioto 2012
Extensão: 450 km.



Rede Zaidan 2014
Extensão: 230 km.

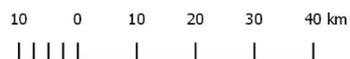
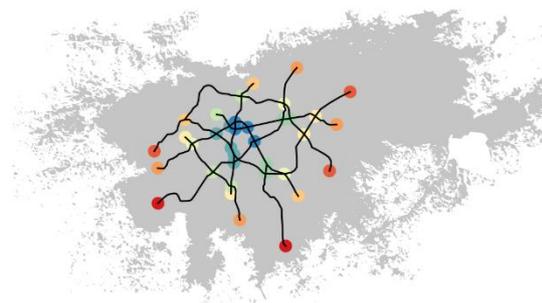
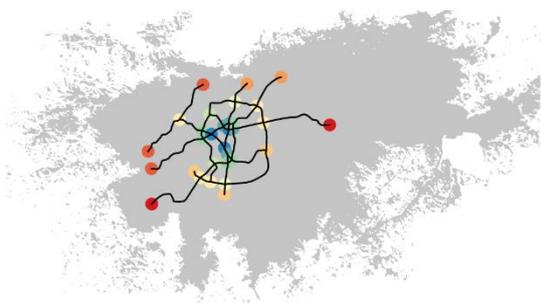


Fig. 2. Redes históricas e experimentais: Custos de deslocamento. Fonte: autores.

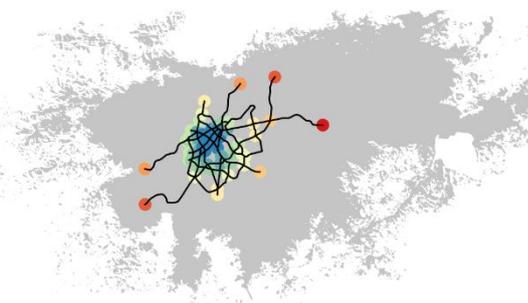
Rede do Plano Integrado de Transportes Urbanos.



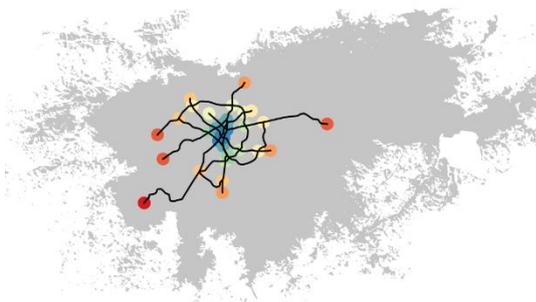
Rede aberta PITU 2020
Extensão: 230 km



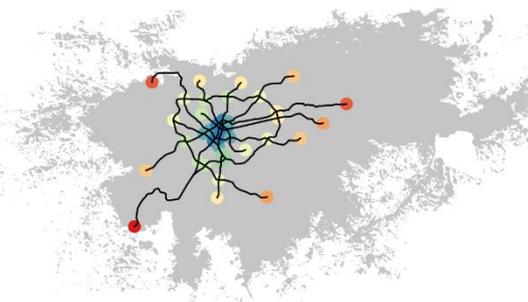
Rede central PITU 2020
Extensão: 180 km



Rede densa PITU 2020
Extensão: 210 km



Rede PITU 2025
Extensão: 165 km



Rede 2030
Extensão: 250 km

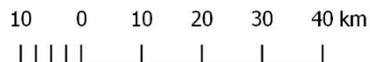


Fig. 3. Redes do Plano Integrado de Transportes Urbanos: Custos de deslocamento. Fonte: autores.

Conclusões

Apesar de constituir uma investigação ainda muito elementar, o experimento desenvolvido e apresentado neste artigo aponta resultados encorajadores para o estudo e a proposição de redes de transporte por meio da abordagem do software livre e da computação paralela. Não apenas os custos com equipamentos e desenvolvimento poderiam deixar de ser um obstáculo a muitas iniciativas, como também o trabalho colaborativo e já acumulado em códigos abertos poderiam acelerar enormemente a concepção de novas ferramentas.

Com isso, o profundo conhecimento empírico de técnicos que operam modelos proprietários, fechados e, não raro, genéricos, poderia ser agregado a análises e simulações das especificidades de cada estrutura urbana, isto é, na operação de seus próprios modelos. Além disso, a maior velocidade na obtenção de resultados possibilita também maior número de alternativas testadas e analisadas em um processo decisório tão importante e de longo prazo, como a definição de um sistema de mobilidade. Os dados georreferenciados e abertos, por sua vez, facilmente convertidos em diferentes modelos de dados, ensejam cotejamentos e cruzamentos virtualmente infinitos entre as mais distintas informações e proposições urbanísticas.

Ainda, a maneira como tais resultados podem ser representados e disseminados os qualifica como um importante e democrático meio de argumentação em uma discussão necessariamente política. Diante disso, o debate a que se propõe este encontro não poderia ser mais oportuno para temática deste artigo. Espera-se, assim, que estas páginas contribuam de algum modo ao amadurecimento de seus questionamentos e encaminhem sólidas e conjuntas iniciativas.

Referências

BEN-ARI, Mordechai. **Principles of concurrent and distributed programming**. Boston: Addison-Wesley, 2006.

GARCIA, Moreno Zaidan. Redes de transportes e espaço urbano: um ensaio de traçado para São Paulo, à luz das experiências de Londres e Paris. Dissertação (Mestrado). Faculdade de Arquitetura e Urbanismo – Universidade de São Paulo. São Paulo, 2013.

ISODA, Marcos Kiyoto de Tani. Transportes sobre trilhos na Região Metropolitana de São Paulo: estudo sobre a concepção e inserção das redes de transporte de alta capacidade. Dissertação (Mestrado). Faculdade de Arquitetura e Urbanismo – Universidade de São Paulo. São Paulo, 2013.

SEDGEWICK, Robert; WAYNE, Kevin. **Algorithms**. Boston: Addison-Wesley, 2012.