

Visualização de Dados de Roteamento para Cidades Inteligentes

Heder Dorneles Soares¹, Edcarllos Santos¹, José Viterbo¹, Hugo Fernandes²,
and Igor Machado Coelho²

¹ Universidade Federal Fluminense, Instituto de Computação
{hdorneles, esantos, viterbo}@ic.uff.br

² Universidade do Estado do Rio de Janeiro, Instituto de Matemática e Estatística
{hugo.fernandes, igor.machado}@ime.uerj.br

Resumo Com o aumento do número de dispositivos conectados à Internet, temos diversos objetos do cotidiano fazendo parte da rede (Internet das Coisas), como carros, drones, dispositivos móveis em geral, casas e até mesmo cidades inteiras. Neste contexto de Cidades Inteligentes, novos problemas surgem com a convergência de diversas áreas relacionadas à computação, visualização e tratamento de grandes volumes de dados de interesse público. O transporte urbano é uma atividade fundamental para as relações entre as diversas entidades em uma Cidade Inteligente. O processo de roteamento um conjunto de decisões multidisciplinares, em que diversos algoritmos eficientes (de otimização) tem sido propostos na literatura, bem como diversos arcabouços focados em comunicação entre os nós. Neste contexto, é necessário que os dados coletados de sistemas de localização geográfica sejam obtidos (através de sensores ou bases de dados), processados e entregues de forma compreensível ao usuário ou tomador de decisão. Porém, poucos trabalhos na literatura tratam da integração das diversas ferramentas existentes, o que se torna um gargalo para o desenvolvimento de novas tecnologias acessíveis ao usuário comum. Neste trabalho, propomos a integração de diversos arcabouços especializados em roteamento e Cidades Inteligentes, com o objetivo de desenvolver um artefato de código-aberto que possa contribuir com o desenvolvimento de novas pesquisas na área.

Keywords: Visualização de Dados, Problemas de Roteamento, Internet das Coisas, Cidades Inteligentes

1 Introdução

Cidades Inteligentes (CI) têm sido o foco de diversos trabalhos recentes na literatura, apresentando desafios e novas possibilidades em CI [15], interfaces com tecnologias de otimização [11], tecnologias assistivas [2], e novas plataformas de visualização [3]. Do ponto de vista de Internet das Coisas [20], do inglês *Internet of Things* (IoT), existe uma convergência tecnológica causada pelo aumento do número de dispositivos conectados à Internet, em que os aparelhos operam

ao mesmo tempo como sensores e dispositivos de processamento e troca de informações.

Neste ecossistema digital, novas ferramentas surgem como pequenos blocos que propiciam a disseminação e integração das diversas tecnologias existentes. Entretanto, o desenvolvimento de sistemas para ambientes inteligentes sempre conta com desafios multidisciplinares. Listamos três grandes desafios relacionados ao tema deste trabalho:

- Redes e Comunicação: coleta de dados em diferentes tipos de tecnologia como 3G, 4G, WiFi; suporte a desconexão e troca de redes (*handover*); e desenvolvimento de Middleware para IoT [10]
- Inteligência Computacional: inclui a busca de caminhos mínimos e planejamento de viagem (*route planning*), utilizando algoritmos clássicos como o de Dijkstra e variantes com A* [14], tendo como outra vertente a resolução de alguma variante NP-Difícil do Problema de Roteamento e Veículos [8], usualmente por meio de estratégias heurísticas e metaheurísticas [13]
- Visualização de dados: interface com o usuário, integrando arcabouços visuais para visualização de mapas por meio linguagens de programação web (usualmente via integração com navegadores)

Propomos um sistema de visualização de rotas em mapas integrado com a base de mapas aberta do OpenStreetMap, para visualização de rotas e projetado para integração futura com um *middleware* de IoT para monitoramento e distribuição de dados em cidades inteligentes. De forma a fornecer através de serviços na nuvem, apresentamos a ideia de um sistema completo de monitoramento de nós móveis, com heurísticas para otimização e tomada de decisão, utilizando uma interface de visualização de informações.

O restante do artigo é organizado cinco seções (incluindo esta introdução). A Seção 2 apresenta uma revisão da literatura sobre trabalhos em cidades inteligentes, otimização para roteamento e visualização para sistemas de geolocalização. A Seção 3 a arquitetura proposta integrando comunicação, visualização e otimização em cidades inteligentes. Na Seção 4, o sistema proposto é apresentado, bem como uma integração com um resolvidor de otimização para roteamento. Finalmente, a Seção 5 apresenta as conclusões do trabalho e perspectivas para trabalhos futuros.

2 Revisão de Literatura

Pesquisas no tópico de Cidades Inteligentes tem buscado estratégias tecnológicas para melhorar a qualidade de vida, e resolver problemas causados pelo crescimento da população e das áreas urbanas. Diversos trabalhos exploram tecnologias de comunicação para melhorar os serviços aos cidadãos e agregar valor à administração pública [15].

Diversas aplicações em monitoramento de veículos (caminhões, carros, bicicletas, drones) e logística/despacho aplicam conceitos de computação móvel (ou redes móveis), o que implica em comunicação, compartilhamento de dados e

coordenação entre um número muito grande de nós móveis [10]. Em especial, diversos problemas de otimização e roteamento de veículos podem ser encontrados na literatura [19], considerando restrições como capacidade do veículo, frota homogênea/heterogênea, limitada/ilimitada, demandas de entrega/coleta, janelas de tempo e fracionamento de entregas. Tais problemas usualmente consideram dados estáticos, considerando apenas a distância ou tempo de deslocamento entre os elementos.

Na literatura, o custo (distância ou tempo) entre um dado par de nós é modelada por uma simplificação da realidade, geralmente considerando apenas a distância Euclidiana entre dois pontos ou o tempo para percorrer a distância Euclidiana com velocidade constante. Assim, o planejamento da rota (mesmo com apenas dois pontos) pode se tornar uma tarefa complexa quando são consideradas vias de acesso e sentido das ruas [9]. O problema se torna ainda mais complexo quando ciclovias, vias exclusivas para pedestres e diferentes modais de transporte são combinados em uma única rota [18]. Neste caso, os autores utilizam uma generalização do algoritmo clássico de Ford-Moore-Bellman [4] através de grafos coloridos que representam redes de transporte multimodal.

A Figura 1 apresenta uma comparação entre os dados de otimização com distância Euclidiana (considerados na vasta maioria dos trabalhos da literatura) e uma abordagem que inclui o planejamento da rota. Embora possa existir uma grande diferença entre os valores de tempo e distância encontradas, tais distâncias e tempos após o planejamento da rota podem ser pré-calculados para cada par de nós e armazenados em uma matriz de ordem quadrática.



Figura 1: Comparação entre dados da literatura que consideram apenas a distância euclidiana ou o planejamento da rota [Fonte: autoria própria]

Esta abordagem é capaz de compatibilizar o uso distâncias Euclidianas através de matrizes para diversos casos, mas vale ressaltar que existem algoritmos na literatura que exploram informações Euclidianas durante o processo de roteamento. Rios et al. [17] apresenta um algoritmo de roteamento que é acelerado quando são utilizadas informações Euclidianas juntamente com arquiteturas de computação de alta performance, não aceitando como entrada uma matriz quadrática de custos. Para a resolução do clássico Problema do Caixeiro Viajante, que consiste em visitar um conjunto de cidades (sem repetição) e retornar à origem com um custo mínimo, existem algoritmos de otimização especializados no uso de distâncias Euclidianas, como o solver Concorde [1].

Uma aplicação interessante no contexto de turismo em Cidades Inteligentes é baseada no *Team Orienteering Problem* [5], um problema NP-Difícil. Nesta aplicação, notas são dadas a diversos Pontos de Interesse da cidade e o processo de otimização visa encontrar rotas que maximizem o interesse dos turistas dentro das janelas de tempo disponíveis [12]. De acordo com os autores, os resultados do trabalho confirmam a importância da integração de sistemas de transporte público com o planejamento de viagens. Testes efetuados com dados da cidade de San Sebastián, localizada no País Basco, indicam que rotas de 4 dias são suficientes para visitar a maioria dos pontos de interesse da cidade.

Diversas ferramentas existem para resolver problemas distintos relacionados à visualização de rotas e aquisição de informações geográficas. Algumas ferramentas são proprietárias e outras são de código-aberto, dentre as quais citamos os seguintes serviços:

Geocoding é o processo de converter endereços como “Rua São Francisco Xavier, 524” para coordenadas geográficas, que seria $-22.9113381, -43.2384333$ neste caso. Google Geocoding API e Nominatim são exemplos do serviço.

Navegação identifica a estrada que o veículo está trafegando e fornece informações adicionais sobre a mesma, como por exemplo o limite de velocidade. Google Roads API e o Skobbler são exemplos do serviço.

Planejamento da Rota (*Route Planning*) calcula rotas entre localidades. Pode-se pesquisar por diversos modos de transporte, como veículo próprio, transportes públicos e até mesmo andando. Google Maps Directions e o Open Source Routing Machine são exemplos do serviço

Matriz de Distâncias/Tempos O Google Maps Distance Matrix API fornece uma matriz com tempo de viagem e distância entre múltiplos endereços. Porém há limitações de requisições e inerentes da latência de rede, o que pode inviabilizar o uso do mesmo para grandes quantidades de localidades. Neste sentido o OSRM também fornece tal funcionalidade com a vantagem de ser instalado localmente sem necessidade de uso da rede, sendo capaz processar grandes volumes de informação em um baixo tempo computacional.

Dentre os serviços apresentados, temos interesse do ponto de vista de levantamento de dados e apresentação de rotas para usuários os serviços de Matriz de Distâncias e Planejamento da Rota.

3 Arcabouço Proposto

A proposta de integração entre os diversos serviços é ilustrada pela Figura 2. O arcabouço considera a interação entre a ferramenta de visualização e dois serviços principais: coleta/distribuição de dados e o serviço de rotas. Por sua vez, o serviço de rotas opera com a integração entre um serviço de planejamento de rota (considerando ciclovias, vias de pedestres, etc) e um serviço de otimização que tem enfoque na resolução de Problemas de Roteamento de Veículos.

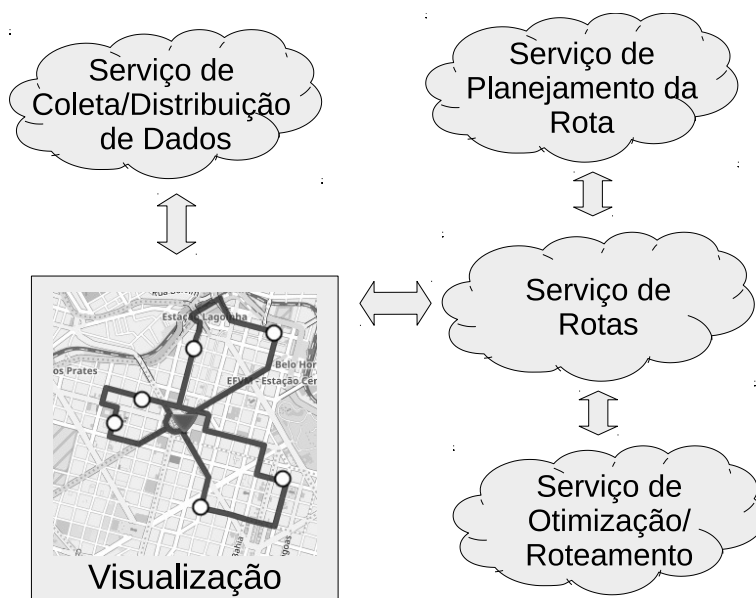


Figura 2: Proposta de integração entre diferentes serviços [Fonte: autoria própria]

A Figura 3 apresenta uma visão geral do arcabouço, sendo organizado em camadas. Neste modelo, a camada de visualização se conecta a um serviço de mapas, sendo na presente versão deste trabalho considerada a base de dados do OpenStreetMap³. Os dados podem ser obtidos de forma gratuita, no qual usuários voluntariamente editam locais, endereços e ruas. A proposta contempla uma integração futura com um serviço de aquisição de informações em tempo real para IoT, como o ContextNet [10], que fornece uma camada de transparente para

³ <http://openstreetmap.org>

acesso a sensores e agentes inteligentes que operam com diferentes protocolos de redes.

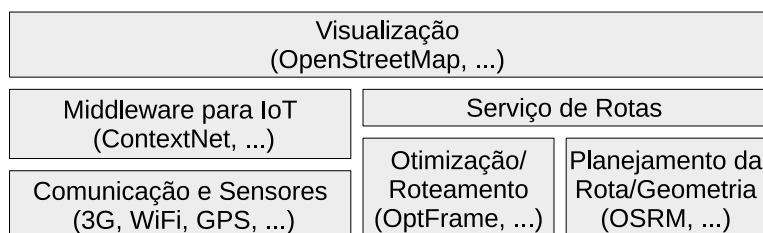


Figura 3: Proposta das camadas do sistema proposto [Fonte: autoria própria]

O serviço de rotas é implementado via a integração entre o Open Source Routing Machine (OSRM) e o arcabouço de otimização OptFrame [6], sendo este um projeto de código-aberto⁴ com implementação C++ de diversas metaheurísticas e algoritmos para resolução de problemas de roteamento de veículos, de classe NP-Difícil.

O OSRM efetua o planejamento da rota, sendo um projeto de código-aberto que implementa em C++ uma engine de alta performance para menores caminhos em redes de rotas. Utilizando algoritmos sofisticados de roteamento com os dados do projeto OpenStreetMap. Neste trabalho, utilizamos um fork temporário⁵ do OSRM gentilmente cedido para os autores do presente trabalho. Esse fork é desenvolvido pelo usuário `niemeier-PSI` e espera-se que seja incorporado ao projeto principal nos próximos meses⁶. Neste fork, foi incluída a possibilidade de construir uma matriz de distancias, dadas as coordenadas dos pontos. O OSRM atua no planejamento de rota (*Route Planning*) e é baseado no algoritmo *Contraction hierarchies* [16], usando técnicas para aumentar a velocidade do roteamento do *caminho mais curto* criando primeiro versões pré-computadas do grafo.

A integração entre os serviços é feita através da ferramenta Docker⁷, de forma a isolar os componentes de forma interoperável. Docker é uma ferramenta recentemente proposta para plataformas baseadas em container, sendo usado principalmente para eliminar possíveis diferenças de um mesmo software executado em máquinas diferentes. O Docker empacota o software de uma maneira que ele possa ser executado isolado do restante do Sistema Operacional, carregando somente configurações e bibliotecas necessárias para a execução do mesmo.

⁴ <https://sourceforge.net/projects/optframe/>

⁵ <https://github.com/niemeier-PSI/osrm-backend/tree/v5.2.7-psi>

⁶ <https://github.com/Project-OSRM/osrm-backend/pull/2764>

⁷ <https://www.docker.com/>

4 Simulação Utilizando o Sistema Proposto

Um protótipo do arcabouço proposto foi implementado em um ambiente Linux Ubuntu 16.04, utilizando as tecnologias apresentadas anteriormente. A ferramenta desenvolvida pode ser encontrada no repositório de código-aberto GitHub sob licença MIT: <https://github.com/Hugbro/cvrp-draw>.

Para ilustrar a proposta e testar o protótipo, sete pontos foram escolhidos na cidade de Belo Horizonte/MG, sendo o objetivo do roteamento criar três rotas limitadas a dois pontos em cada (o primeiro é considerado o ponto de partida, ou depósito, em um contexto de logística de entregas). A proposta de trabalhar com um exemplo de pequeno porte visa unicamente apresentar a integração entre as diferentes ferramentas, mas ressaltamos que problemas de grande porte tem sido resolvidos pelos autores em diversos trabalhos anteriores na literatura [7].

As Tabelas 1-2 apresentam as matrizes de distâncias e tempos relativas às coordenadas consideradas no experimento. Vale ressaltar que, mesmo após um extenso levantamento das tecnologias existentes, os autores não encontraram essa funcionalidade na versão padrão de nenhuma ferramenta de forma a atender demandas de grandes quantidades de nós (centenas e até milhares), o que ocorre frequentemente na prática. A solução a este problema se deve à integração da ferramenta com um fork específico do OSRM, desenvolvido pelo usuário GitHub `niemeier-PSI`.

Tabela 1: Matriz de distâncias obtida do OSRM (via fork)

Distâncias entre pontos (em metros)						
0	659.30	845.23	1526.19	1409.07	1309.22	1136.02
1149.15	0	776.26	2405.92	2055.26	2314.10	905.37
923.99	1554.49	0	1767.20	1645.08	2204.41	1542.01
1843.16	2208.42	2597.90	0	1169.12	1932.94	2983.70
849.19	1441.42	1694.43	899.14	0	1861.63	1985.22
2163.45	2054.13	2541.60	1969.86	2576.91	0	1294.22
1142.14	900.31	1382.76	2398.92	2048.25	2113.21	0

Tabela 2: Matriz de tempos obtida do OSRM (via fork)

Tempos entre pontos (em segundos)						
0	67.8	90.6	128.5	129.6	99.4	106.3
113.6	0	81.9	211.7	188.7	170	86.4
83.6	142.5	0	162.1	148.2	174.1	157.4
156.5	190	229.7	0	106.6	152.3	235.9
70.7	135.8	161.3	79.6	0	159.1	177
180.3	170.3	191.3	164.3	220.4	0	106.2
101.5	78.2	131.8	199.6	176.6	152.7	0

Um segundo experimento visava integrar o serviço de roteamento e otimização (via arcabouço OptFrame) com a ferramenta de visualização de rotas. A etapa de otimização foi integrada via linguagem C++ (nativa do arcabouço OptFrame), porém no futuro pretendemos efetuar a integração através da linguagem de modelagem CombView⁸, atualmente em desenvolvimento.

A Figura 4 apresenta a rota desenhada e exportada em formato gráfico utilizando informações do serviço de rotas desenvolvido. Diversos desafios foram encontrados para a obtenção do mapa final, pois as ferramentas existentes são desenvolvidas para navegadores, e recursos de extração de imagem (*static map*) deixam de funcionar quando elementos externos (como rotas e marcadores) são inseridos no mapa. Para contornar esse problema, a ferramenta de visualização foi integrada ao PhantomJS⁹. PhantomJS é um WebKit scriptável com uma API JavaScript. Tem suporte rápido e nativo para vários padrões web: DOM, CSS, JSON, Canvas e SVG. Em outras palavras: PhantomJS é um navegador web sem interface gráfica.

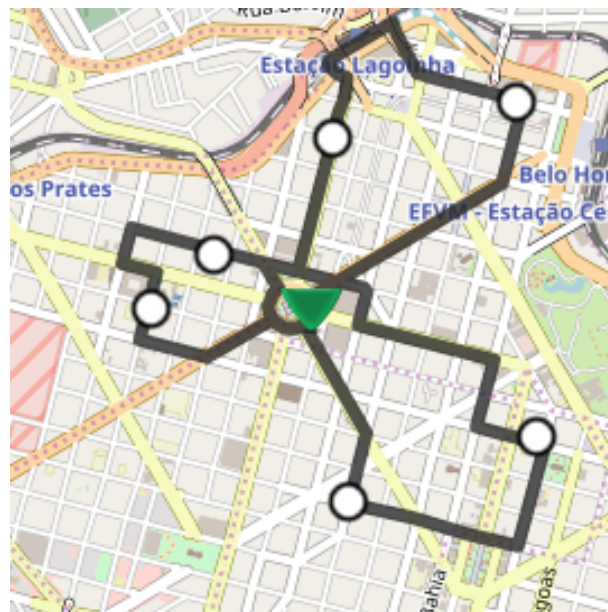


Figura 4: Exemplo de rota obtida pelo arcabouço proposto [Fonte: autoria própria]

⁸ <https://github.com/igormcoelho/language-combview>

⁹ <http://phantomjs.org/>

5 Conclusões e Trabalhos Futuros

Este trabalho apresenta uma proposta de arquitetura para aquisição de dados, roteamento e visualização de rotas em uma Cidade Inteligente. Embora diversas ferramentas de alta qualidade existam para cada uma destas vertentes, a integração entre elas é capaz de trazer grandes desafios.

Dentre os desafios elencados, apresentamos soluções para: aquisição e processamento de grandes porções de informação através da base do OpenStreetMaps e do OSRM (um fork específico fora do ramo principal do projeto OSRM foi considerado); integração da ferramenta de visualização com um arcabouço de otimização, independentemente da necessidade de um navegador.

Em uma Cidade Inteligente, problemas de roteamento estão ligados ao sistema de transporte público e particular, entrega/coleta de produtos, transporte multimodal, carros elétricos com pontos de recarga e roteamento de drones. Já existem diversos trabalhos na literatura com foco na resolução de problemas de otimização em cada uma dessas linhas, porém a maioria considera apenas dados estáticos (muitas vezes subestimados e fora da realidade). Neste sentido, propomos a integração futura da ferramenta com arcabouços de IoT como o ContextNet, abrindo a possibilidade de gerenciar a tomada de decisão com dados em tempo real, considerando dados dinâmicos e conectando as ferramentas de otimização existentes a agentes inteligentes de forma transparente.

Agradecimentos

Os autores agradecem às agências de fomento FAPERJ, CAPES e CNPq pelas bolsas concedidas, e ao usuário do GitHub `niemeier-PSI` por gentilmente ceder uma versão estável do código de cálculo de matrizes de distância para uso no OSRM.

Referências

1. Applegate, D.L., Bixby, R.E., Chvatal, V., Cook, W.J.: The traveling salesman problem: a computational study. Princeton university press (2011)
2. Barbosa, J., dos Santos, J.C., Mengue, L., Telles, M., Barth, M., Righi, R.: Uma simulação de uma cidade inteligente assistiva. In: Proceedings of the XLIX SBPO (2017)
3. Barbosa, J.L.V., dos Santos, J.C., Mengue, L., Telles, M.J., Barth, M.J.: Otimização multi-nível para projeto de redes de acesso para implementação de cidades inteligentes. In: Proceedings of the XLIX SBPO (2017)
4. Bellman, R.: On a Routing Problem. Quarterly of Applied Mathematics 16, 87–90 (1958)
5. Chao, I.M., Golden, B.L., Wasil, E.A.: The team orienteering problem. European Journal of Operational Research 88(3), 464 – 474 (1996), <http://www.sciencedirect.com/science/article/pii/0377221794002894>
6. Coelho, I.M., Ribas, S., Perché, M.H.d.P., Munhoz, P.L.A., Souza, M.J.F., Ochi, L.S.: Optframe: a computational framework for combinatorial optimization problems. In: Proceedings of the XLII SBPO (2010)

7. Coelho, V., Grasas, A., Ramalhinho, H., Coelho, I., Souza, M., Cruz, R.: An ils-based algorithm to solve a large-scale real heterogeneous fleet vrp with multi-trips and docking constraints. *European Journal of Operational Research* 250(2), 367 – 376 (2016), <http://www.sciencedirect.com/science/article/pii/S0377221715008899>
8. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. *Management Science* 6(1), 80–91 (1959), <https://doi.org/10.1287/mnsc.6.1.80>
9. Delling, D., Sanders, P., Schultes, D., Wagner, D.: Engineering route planning algorithms. *Algorithmics of large and complex networks* 5515, 117–139 (2009)
10. Endler, M., Baptista, G., Silva, L., Vasconcelos, R., Malcher, M., Pantoja, V., Pinheiro, V., Viterbo, J.: Contextnet: context reasoning and sharing middleware for large-scale pervasive collaboration and social networking. In: *Proceedings of the Workshop on Posters and Demos Track*. p. 2. ACM (2011)
11. Figueiredo, A., Veloso, I., Veloso, I., Souza, M., Oliveira, T., Veloso, V.M., Coelho, V., Coelho, V.: Uma visão multicritério para julgamentos em cidades e sociedades inteligentes. In: *Proceedings of the XLIX SBPO* (2017)
12. Garcia, A., Vansteenwegen, P., Arbelaitz, O., Souffriau, W., Linaza, M.T.: Integrating public transportation in personalised electronic tourist guides. *Computers & Operations Research* 40(3), 758 – 774 (2013), <http://www.sciencedirect.com/science/article/pii/S0305054811000967>, *transport Scheduling*
13. Gendreau, M., Potvin, J.Y.: *Handbook of Metaheuristics*, vol. 2. Springer (2010)
14. Goldberg, A.V., Harrelson, C.: Computing the shortest path: A search meets graph theory. In: *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. pp. 156–165. SODA '05, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2005), <http://dl.acm.org/citation.cfm?id=1070432.1070455>
15. Lourenço, H.R., Oliver, M., Oliveira, T., Coelho, V., Tavares, W.: Desafios operacionais e digitais para conectar cidadãos em cidades inteligentes. In: *Proceedings of the XLIX SBPO* (2017)
16. Luxen, D., Vetter, C.: Real-time routing with openstreetmap data. In: *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. pp. 513–516. GIS '11, ACM, New York, NY, USA (2011), <http://doi.acm.org/10.1145/2093973.2094062>
17. Rios, E., Ochi, L.S., Boeres, C., Coelho, V.N., Coelho, I.M., Farias, R.: Exploring parallel multi-gpu local search strategies in a metaheuristic framework. *Journal of Parallel and Distributed Computing* 111(Supplement C), 39 – 55 (2018), <http://www.sciencedirect.com/science/article/pii/S0743731517302009>
18. Shirabayashi, J.V., Yamakami, A., Silva, R.C., Shirabayashi, W.V.: Metodologias de soluções para o problema de redes multimodais aplicadas ao tráfego urbano. In: *Proceedings of the XLIX SBPO* (2017)
19. Vidal, T., Crainic, T., Gendreau, M., Prins, C.: Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research* 231(1), 1–21 (2013), <https://www.cirrelet.ca/DocumentsTravail/CIRRELT-2012-05.pdf>
20. Weiser, M.: The computer for the 21st century. *Mobile Computing and Communications Review* 3(3), 3–11 (1999)