

Hermes: Identificação de Menores Rotas em Dispositivos Móveis*

Eduardo Augusto Sobral Junior¹, Jonice Oliveira¹

¹ Universidade Federal do Rio de Janeiro

Caixa Postal: 68.530, Cep: 21.941-909 – Rio de Janeiro – RJ – Brasil

easobral@gmail.com, jonice@dcc.ufrj.br

Abstract. *Traffic jam is a frequent problem in Brazilian metropolises. Part of this problem could be solved if drivers knew alternative routes. In addition, we note the increase of computational power and versatility of phones and mobile devices. This project, entitled Hermes, aims to recommend new routes from the original-planned route and data from what has been traveled. All data collection, processing and recommendation are made locally. In this article, we also show the results obtained by one simulation, where we analyzed the solution's effectiveness and efficiency.*

Resumo. *Trânsito lento é um problema recorrente nas grandes cidades brasileiras. Parte do problema poderia ser resolvido se os motoristas conhecessem alternativas de rotas. Junto a este fato, notamos o aumento do poder computacional e versatilidade dos celulares e dispositivos móveis. Este projeto, intitulado Hermes, visa recomendar novas rotas a partir da rota planejada e dados do percurso realizado. Toda a coleta de dados, processamento e recomendação são feitos localmente. Neste artigo, também são mostrados os resultados obtidos através de uma simulação, onde foram analisados a eficácia e eficiência da solução.*

1. Introdução

Nos grandes centros urbanos, problemas como congestionamentos se tornam cada vez maiores. Além da perda de tempo e impacto financeiro, este problema impacta diretamente a qualidade de vida das pessoas. Congestionamentos também atrapalham serviços de resgate e prestação de socorro de pessoas em situações de emergência.

O problema de encontrar rotas de trânsito na malha rodoviária é análogo ao de encontrar caminhos em grafos, o que exige um bom poder computacional. No caso de dispositivos móveis, ocorre a dependência de um servidor (que realizará todo o processamento e devolverá o resultado), bem como uma conexão à Internet. Uma abordagem cliente-servidor para dispositivos móveis possui algumas vantagens como: i) Coleta e agregação de uma maior quantidade de informações sobre a cidade e ii) Custo menor de processamento e menor gasto da bateria nos dispositivos móveis. Porém, em situações de emergência, onde os desastres podem afetar a infraestrutura local de comunicação ou em áreas onde não há conexão à Internet, o modelo cliente-servidor não se aplica, exigindo o processamento local das rotas.

* Um vídeo de demonstração da ferramenta está disponível em <http://sbbd2013.cin.ufpe.br/screenscasts>.

Hermes é um aplicativo desenvolvido para dispositivos móveis Android que visa recomendar o caminho de menor tempo até um determinado local. Neste trabalho iremos abordar o aspecto de encontrar o menor caminho (considerando o custo em tempo) entre dois pontos da malha rodoviária de uma cidade.

2. Trabalhos Correlatos

Ding et al. [2007] descrevem uma solução que sugere rotas, a qual se baseia em VANET¹ para conseguir informação do estado local das vias e deduzir um bom caminho.

Thiagarajan et al. [2009] exploram um mecanismo de coletar dados do GPS durante um percurso e mapeia os pontos em ruas, estimando o tempo gasto por trechos. Os autores apenas demonstram o mapeamento, mas não oferecem soluções para a recomendação de rotas.

Schultes [2008] faz uma revisão dos algoritmos mais comumente utilizados no roteamento em estradas. Diferentemente do Hermes, as malhas rodoviárias são extensas e o autor não considera a execução dos algoritmos em dispositivos móveis, que possuem processamento e memória limitados.

O Waze² é um aplicativo cliente-servidor que permite que os usuários compartilhem colaborativamente as informações sobre as condições de trânsito: localização de paradas policiais (*blitz*), congestionamentos, acidentes e obras. Ao informar o ponto inicial e final, o aplicativo dimensiona a melhor rota.

O sistema Hermes, apresentado neste artigo, atua no principal desafio da área: capturar, processar e recomendar rotas utilizando unicamente os recursos computacionais de um dispositivo móvel.

3. Arquitetura

Como mostrado na Figura 1, o sistema trabalha com fontes externas e possui 3 módulos, os quais são explicados a seguir.

3.1. Fontes Externas

GPS³ - Os trechos já percorridos são coletados através do GPS do próprio dispositivo. Para isto, utilizamos a API do Android de acesso a esse dispositivo.

Mapas – Para obtermos as informações sobre as vias e suas direções, utilizamos o OpenStreetMaps⁴. O banco de dados da aplicação se limitou à região contida entre as latitudes -23.099 e -22.743 e entre as longitudes -43.798 e -43.137, o que corresponde à cidade do Rio de Janeiro. Foi obtido um arquivo .osm e importado em um banco de dados postgresql, utilizando a ferramenta Osmosis para isso. Como esse banco de dados possui muito mais informação que a necessária para este trabalho (há informação de rota de ônibus, localização de determinados edifícios, alguns nomes de ruas, traçado de lagos e rios) foi retirada dele apenas o traçado da malha rodoviária (devido à tentativa

¹ VANET - Vehicular Ad-Hoc Network

² <http://www.waze.com/>

³ GPS - Global Positioning System

⁴ <http://www.openstreetmap.org/>

de ocupar o menor espaço possível nos dispositivos). Essa informação se limita aos nós (id, latitude e longitude) e arestas (quais nós são ligados e uma estimativa do tempo de viagem baseada na distância).

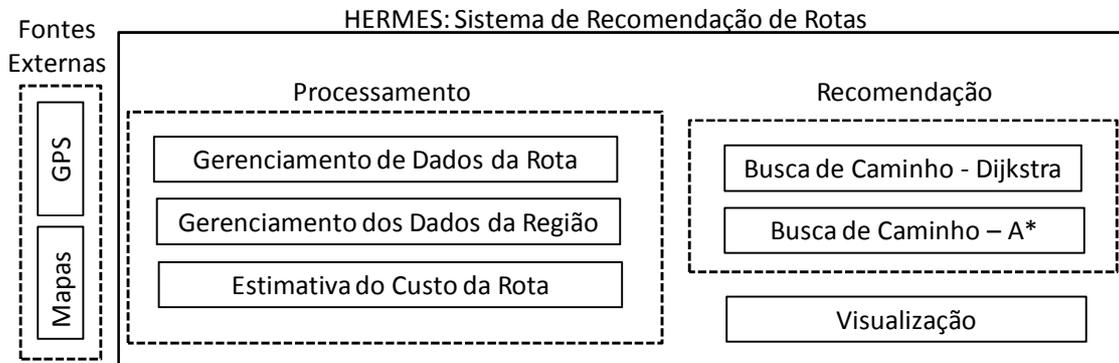


Figura 1: Arquitetura Conceitual

3.2. Processamento

Gerenciamento dos Dados da Rota – este é o módulo responsável por gerenciar os dados de uma viagem em uma malha/localidade. Os dados utilizados são representados através de um grafo direcionado com custos nas arestas, onde cada vértice é um ponto representativo da rota, como os cruzamentos. Podemos mencionar como responsabilidades deste módulo:

- coletar as informações da viagem - são obtidas iniciando-se um serviço que fica em espera e periodicamente captura a posição indicada pelo GPS e o horário.
- operações básicas do grafo – atualizar e prover informações em resposta a solicitações dos demais módulos, como detalhes dos vértices (identificação, nome e posição), listas de adjacências, distância entre nós, dentre outras.
- manter as informações sobre os custos – os custos, representados como pesos nas arestas, estão relacionados ao tempo gasto em se deslocar em um ponto a outro.

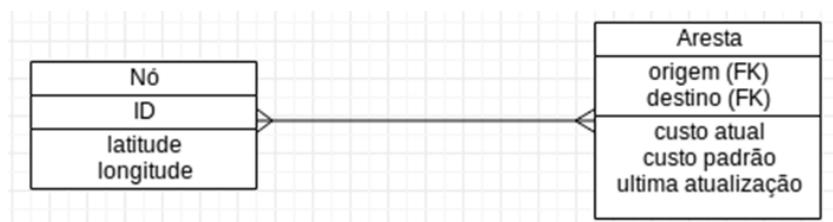


Figura 2: Modelagem do Banco de Dados

Nosso grafo é mantido em um banco de dados relacional SQLite (o único disponível para a plataforma no início do desenvolvimento), como mostrado na Figura 2. Cada aresta representa um segmento de rua e alguns nós representam as esquinas (outros nós existem para fins de visualização, com o intuito de traçar caminhos sinuosos de forma correta). O custo de uma aresta representa o tempo necessário para atravessar um segmento. Por falta de dados precisos sobre o comportamento das vias, foi estimado o tempo necessário para atravessar a via da seguinte forma: tamanho do segmento / velocidade de 39,6 km/h. Essa velocidade é utilizada por ser uma estimativa da

velocidade média na cidade do Rio de Janeiro, baseado em notícias divulgadas em jornais.

As principais classes utilizadas, mostradas na Figura 3, são:

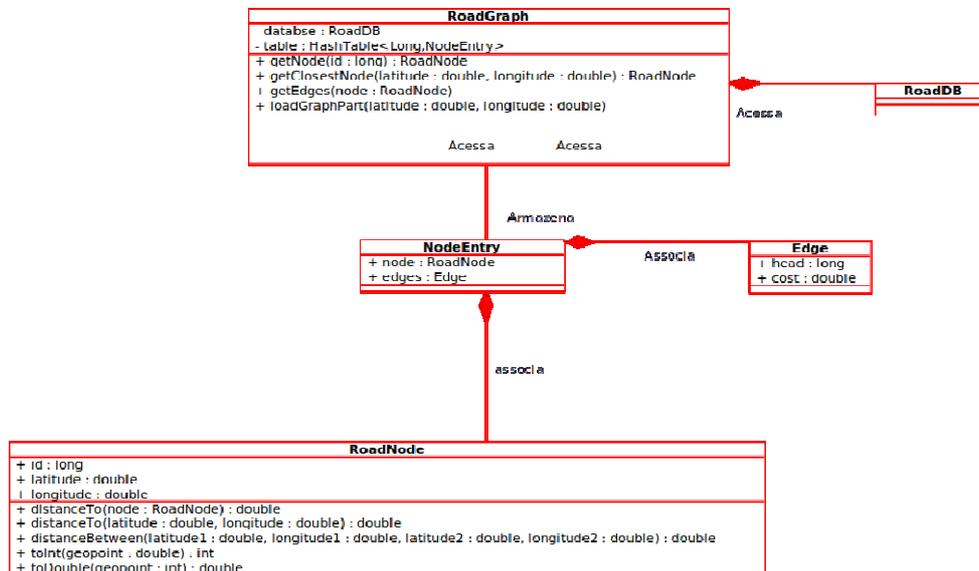


Figura 3: Diagrama de classes

RoadGraph – ‘singleton’ responsável pelas abstrações para converter as informações gravadas no banco de dados em um grafo em memória.

RoadNode - responsável pelas informações que envolvem apenas nós (por exemplo, distância baseada na latitude e longitude).

Edge - Classe que guarda o custo e o destino de uma aresta.

Gerenciamento dos Dados da Região – importação e gerencia os dados relativos à informação da malha rodoviária da região.

Estimativa do Custo da Rota – As informações são guardadas do início até o fim da viagem. A sequência de nós visitados e o horário em que foram visitados são armazenados. O tempo de viagem entre nós é calculado como a diferença entre os instantes de visita. O custo (em tempo) pode ser então atualizado no banco de dados, assim como o tempo da última atualização.

3.3. Cálculo dos menores caminhos

Foram utilizados dois algoritmos de busca em grafos: O algoritmo Dijkstra e o algoritmo A* simples, ambos explicados detalhadamente em (Cormen, 2011). Ambos foram desenvolvidos como ‘threads’ e após o término de suas execuções, devolvem o resultado ao processo original para serem exibidos.

3.4. Visualização

Com o resultado obtido, a rota alternativa é mostrada, como exemplificado pela Figura 4. A visualização é feita utilizando a API do Google Maps para Android.

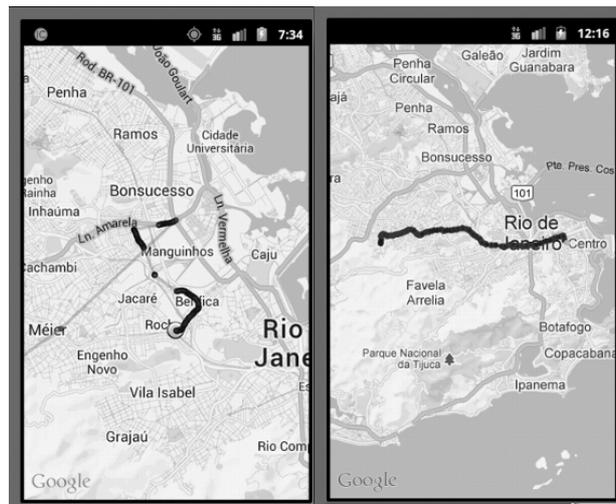


Figura 4: Esquerda: a coleta de dados da viagem. Direita: a resposta de recomendação

4. Avaliação

Para esta avaliação utilizamos um telefone celular Sony Ericsson Xperia X8, processador ARM6 com 600Mhz e 48Mb de memória virtual para a máquina Java e rotas entre diversos pontos da cidade do Rio de Janeiro.

Nos gráficos abaixo é mostrada a relação da distância entre nós em quilômetros e tempo de execução em segundos para os algoritmos A* (Figura 5) e Dijkstra (Figura 6). As rotas foram geradas aleatoriamente.

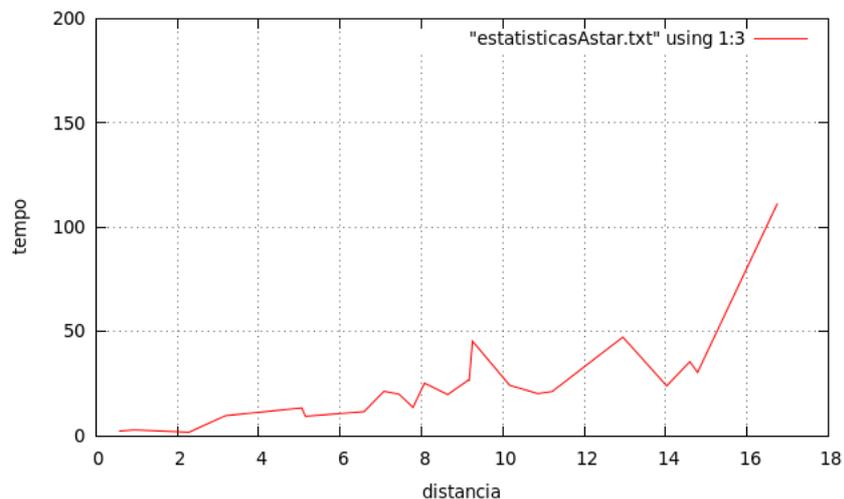


Figura 5: Desempenho A*

Para efeito de comparação, o software Waze, que utiliza conexão com servidor para obter sua recomendação de rota possui tempo independente da distância. O gargalo da resposta do tempo, neste caso, se encontra na rede. Mantendo origem e destinos iguais, a demora na resposta variou de 1 a 5 segundos, em redes Wifi e 3G. Por ser um produto fechado, utilizou-se um cronômetro para realizar a medição, podendo trazer imprecisão na medida de tempo.

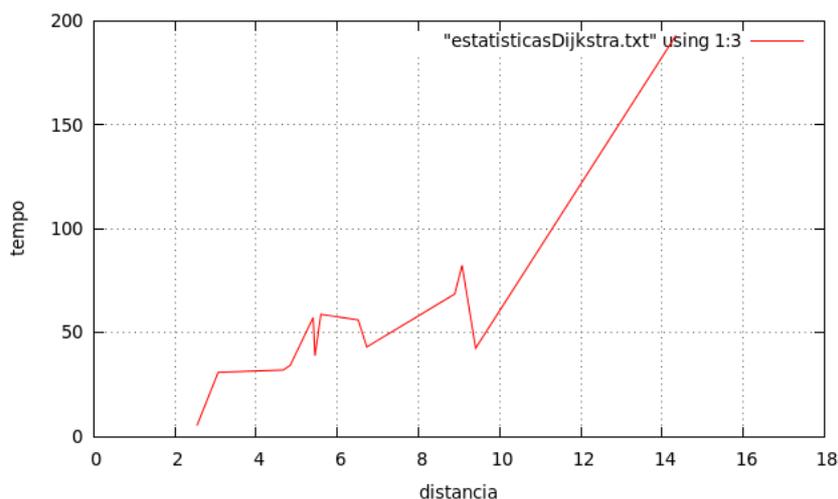


Figura 6: Desempenho Dijkstra

5. Conclusão

Este artigo descreveu o Hermes, um sistema para recomendação de rotas em dispositivos móveis. A execução totalmente local é possível, mas ainda há alguns desafios a serem superados como o consumo de memória e tempo de execução. Além disso, utilizaremos outros tipos de custos para as arestas, como a periculosidade da área, custo monetário (pedágios, por exemplo), acidentes e probabilidade de congestionamento, integrando o Hermes com outros ambientes que possam prover tais informações, como (Lauand, 2013). Outro objetivo é obter heurísticas mais precisas.

Agradecimentos. Ao CNPq e à FAPERJ.

Referências

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). Introduction to Algorithms. MIT Press and McGraw-Hill, 2nd edition.
- Ding, Y., Wang, C., Xiao, L. (2007). A static-node assisted adaptive routing protocol in vehicular networks. In *ACM VANET*.
- Jerbi, M., Senouci, S., Doudane, Y., and Beylot, A. L. (2008). Geo-localized virtual infrastructure for urban vehicular networks. In *8th International Conference on ITS Telecommunications*, pages 305–310.
- Lauand, B., Oliveira, J. (2013). TweepTraffic: ferramenta de análise das condições de trânsito baseado nas informações do Twitter. In *BraSNAM*.
- Schultes, D. (2008). Route planning in road networks. In *Ausgezeichnete Informatikdissertationen*, pages 271–280.
- Thiagarajan, A., Ravindranath, L., LaCurts, K., Madden, S., Balakrishnan, H., Toledo, S., and Eriksson, J. (2009). Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, SenSys '09*, pages 85–98, New York, NY, USA. ACM.