

Um Método Heurístico para Cálculo de Rotas com Desvios de Obstáculos em Serviços Web de Mapas Digitais

Victor E. D. Basso, Vasco Furtado

Programa de Pós-graduação em Informática Aplicada (DIA/MIA)

Laboratório Engenharia do Conhecimento, NATI – Universidade de Fortaleza (UNIFOR) – Fortaleza – CE – Brasil

`vitobasso@gmail.com`, `vasco@unifor.br`

***Abstract.** Computing a route between two points that deviate from an obstacle is a typical activity of geoprocessing systems, but that is not supported by services of digital map in the web due to the high computational cost. We propose here an approach to the problem that uses conjointly the shortest route provided by these services and heuristic methods based on visibility algorithms. This method is inspired in route planning for robots what can suggest alternative waypoints to force the contour of the obstacles identified. A comparative assessment of different scenarios indicates the feasibility of the method and its potential for improvement.*

***Resumo.** O cálculo de rotas entre dois pontos que desviem de obstáculos é uma atividade típica de sistemas de geoprocessamento, mas que não é provida por serviços web de mapas digitais dado o seu alto custo computacional. Propomos aqui uma abordagem para o problema que consiste em usar conjuntamente o cálculo de rota mais curta provido por esses serviços e métodos heurísticos baseados em algoritmos de visibilidade. Esses métodos são inspirados em planejamento de rotas para robôs, levando a sugerir pontos intermediários alternativos às rotas originalmente traçadas pelos SWMD que forcem o desvio dos obstáculos identificados. Uma avaliação comparativa em diversos cenários indica a viabilidade do método e seu potencial de melhoria.*

1. Introdução

Serviços Web de Mapas Digitais (SWMD) como os proporcionados por Google (www.maps.google.com), Yahoo (www.maps.yahoo.com) e Bing (www.bing.com/maps) são extremamente populares hoje em dia. Tornaram-se essenciais para um grande número de aplicações, em particular, aquelas conhecidas como *mashup* [Duane, 2006], onde informações são automaticamente agregadas aos mapas e funcionalidades típicas de um Sistema de Informações Geográficas são providas. Recuperar informações dentro de um determinado raio, consultar informações para determinada região são exemplos destas funcionalidades.

O cálculo de rotas entre dois pontos é outra funcionalidade típica deste contexto, no entanto, o provimento desse tipo de serviço não se faz sem desafios. O cálculo de uma rota mais curta entre dois pontos é um problema de complexidade $O(v^2)$ onde v é o número de possíveis pontos intermediários entre o início e destino. Praticamente, essa complexidade é ainda muito custosa para os SWMD que de uma forma geral implementam métodos heurísticos inclusive com a escolha de rotas mais promissoras (avenidas ao invés de ruas pequenas, por exemplo) [Ishikawa, 1991].

Esse problema é mais sério quando se trata do cálculo de rotas em que obstáculos precisam ser contornados. Via de regra, os obstáculos são formados pela aglomeração de eventos vindos tanto da participação voluntária das pessoas, dito VGI [Rouse, 2007] ou vindos de importação de outros sistemas de informações. O dinamismo dessa captura de informações bem como a complexidade de tempo que é ainda maior do que um simples cálculo de rota tornam difícil o provimento deste tipo de serviço pelos provedores de SWMD.

A solução que propomos para esse problema é realizar o cálculo de uma *rota desobstruída* (que contorne os obstáculos) no cliente fazendo uso de serviços dos SWMD. Tal solução impõe novos desafios. O primeiro deles advém do fato que os SWMD não oferecem acesso direto ao mapa com todos os dados geográficos, por exemplo, o eixo de ruas, que são acessíveis apenas visualmente. Conseqüência disto é que o desenvolvimento de métodos alternativos de cálculo de rotas requer o uso do serviço de cálculo de rotas provido pelos SWMD e assim as características das rotas fornecidas pelos SWMD (comprimento, formato, locais por onde passa) só são conhecidas após uma requisição ser feita a este serviço. O segundo problema é que essas próprias requisições ao serviço de cálculo de rotas possuem um custo e devem ser minimizadas. Dadas essas restrições o problema que buscamos resolver é desenvolver um algoritmo de cálculo de rotas que contornem obstáculos, minimizem a distância e o número de chamadas aos serviços de cálculo de rotas dos SWMD.

Propomos uma abordagem inspirada em planejamento de rotas para robôs [Sánchez, 2004], em particular, em algoritmos de visibilidade [Senjuti, 2007] que consiste em definir heurísticas para gerar pontos intermediários alternativos às rotas originalmente traçadas pelas rotas mais curtas sugeridas pelos SWMD.

A aplicação de nosso método foi feita no sistema WikiCrimes [Furtado et al. 2010] onde informações voluntárias sobre ocorrências criminais são informadas no Google Maps. A ideia é desenvolver rotas ditas seguras que contornam áreas de alta densidade de ocorrências criminais calculadas por métodos de Kernel [McLafferty, 2000].

2. Delineamento Básico da Solução Proposta

Em linhas gerais, o método que propomos segue o seguinte procedimento iterativo. A cada iteração, o serviço de rotas do SWMD é acessado uma vez e uma nova rota real é descoberta e agregada a um grafo que liga a nova rota com rotas descobertas em iterações anteriores. Em cada iteração, novas rotas são estimadas entre alguns pares de vértices do grafo. Estas estimativas são armazenadas em uma fila de prioridades baseada em uma função de peso da rota e acumulam com as estimativas feitas em outras

iterações. A melhor estimativa da fila em determinada iteração é escolhida para ser parâmetro do cálculo da próxima rota real. Ao ser escolhida uma rota estimada, os pontos que a compõem são inseridos no grafo, abrindo possibilidades para novas estimativas. O processo se repete até que uma rota desobstruída ligando a origem ao destino seja reconhecida por uma função de tolerância.

Nos tópicos a seguir descrevemos em mais detalhe cada aspecto da técnica desenvolvida. Iniciamos descrevendo a forma como as ruas e a obstrução são representadas. Depois apresentamos as funções de peso de rotas e de tolerância. Seguimos apresentando a estrutura geral do algoritmo e por fim as estratégias de estimativa de rotas que experimentamos.

2.1. Representação da obstrução

A *obstrução* é representada por um *mapa de calor* gerado a partir dos eventos pontuais.

Um algoritmo de *mapa de calor*, como o estudado em [McLafferty, 2000], divide o mapa em uma grade fina. A cada célula da grade, é associado um valor de densidade representativo para os eventos em sua proximidade. Ao final, cores são atribuídas a cada célula, dependendo do valor, resultando em um *mapa de calor* visual.

Utilizamos a grade de densidades do *mapa de calor* para representar a obstrução.

2.2. O grafo de rotas

No mapa original, ou seja, o conhecido pelo SWMD, existem informações de ruas, seu nome, sua localização física, sentidos e conexões. No mapa conhecido pelo nosso algoritmo, apenas partes dessas informações estão disponíveis.

Modelamos este mapa em um grafo de rotas como visto na figura 1. Neste grafo, as arestas são rotas unidirecionais e os vértices são origens e destinos das rotas. Cada vértice é definido por um ponto no plano. Cada aresta liga dois vértices e contém uma sequência de pontos cujo traçado define uma rota gerada pelo SWMD.

Para o problema aqui discutido, as arestas do grafo possuem um peso, que representa tanto o comprimento da rota como a obstrução associada à região pela qual ela passa.

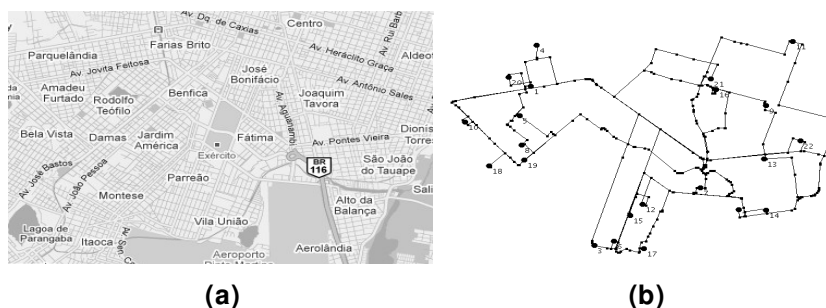


Figura 1. (a) O mapa disponível na interface gráfica do SWMD com a malha viária completa da região. (b) O grafo de rotas que contém parte da malha viária revelada a partir de 20 requisições.

Sejam:

r uma rota;

$o(i,j)$ a densidade da célula do mapa de calor que fica na i -ésima linha e j -ésima coluna;

n e m o número de linhas e colunas do mapa de calor respectivamente;

$c(r,i,j)$ o comprimento da parte da rota que está contida na célula (i,j) do mapa de calor;

O peso da rota r é definido por:
$$peso(r) = \sum_{i=1}^n \sum_{j=1}^m [c(r,i,j) \cdot o(i,j)]$$

O peso é usado, a cada iteração, na escolha da rota estimada pela fila de prioridades, para ser enviada ao SWMD, e também na identificação da melhor rota do grafo de rotas, a fim de testar sua tolerância.

2.3. A Função Tolerância

Cada vez que uma nova rota é revelada, é possível que um caminho melhor seja encontrado entre a origem e o destino. A função peso quantifica o nível de obstrução e o comprimento da rota. Usamos o peso para decidir se uma rota é melhor que outra, porém, decidir se uma rota já é boa o suficiente, ou seja, se a busca terminou, é outro problema.

Sejam:

c um cenário composto por origem, destino, área e mapa de calor;

$l_{min}(c)$ a distância entre a origem e o destino de c ;

$o_{med}(c)$ a obstrução média em c ;

$o_{max}(c)$ a obstrução máxima em c ;

$o_n(c) = \frac{o_{med}(c)}{o_{max}(c)}$ a obstrução normalizada;

K uma constante que representa a influência da distância no resultado;

F um fator de ajuste;

A tolerância no cenário c é definida como: $t(c) = l_{min}(c) * [K + o_n(c)] * F$, onde K e F são parâmetros arbitrários de calibração.

3. Heurísticas de geração de rotas estimadas

O ponto principal do algoritmo proposto refere-se às heurísticas para gerar as rotas estimadas. Usamos três heurísticas que têm como função principal gerar o que chamamos de pontos promissores. Trata-se de uma sequência de pontos que, ao ser usada como intermediária em uma chamada ao SWMD, há boas chances de que a rota resultante seja uma rota desobstruída. Um conjunto de pontos é promissor para um mapa de calor, origem e destino específicos. Ou seja, um ponto que é promissor para um subproblema de origem O_1 e destino D_1 pode não ser promissor para outro com O_2 e D_2 distintos.

Sejam (P_1, P_2, \dots, P_n) pontos promissores para P_o e P_d , então a sequência de pontos $(P_o, P_1, P_2, \dots, P_n, P_d)$ é uma *rota estimada*. Ou seja, uma rota estimada pode ser

usada como parâmetro de uma chamada ao SWMD, tendo boas chances de gerar um bom resultado.

3.1 Mapa de visibilidade

Nesta estratégia, consideramos que um ponto é promissor quando o ponto tem visibilidade em relação à origem e ao destino da rota.

Usamos o termo visibilidade em analogia com o seguinte cenário: imagine que a origem e destino sejam fontes de luz e a obstrução do mapa diminui a passagem da luz. Dizemos que um ponto P tem boa visibilidade quando a região do mapa entre ele e a origem O (e o destino D) não está obstruída, ou seja, P recebe bastante luz de O e D .

Se os obstáculos fossem representados por polígonos opacos, a visibilidade deveria ser representada por uma função binária: “visível”/“não visível”. Porém, na estratégia mapa de visibilidade, os obstáculos são representados por manchas de densidade variável, ou seja, o mapa de calor. Neste caso, a visibilidade é representada por um número decimal.

Definimos a função visibilidade para um ponto P , origem O e destino D como segue: $visibilidade(P) = \frac{1}{peso(OPD)}$, onde OPD é a concatenação do segmento de reta que vai de O até P com o que vai de P até D .

A visibilidade é calculada para pontos no mapa inteiro, então os de melhor visibilidade são organizados numa fila de pontos promissores.

O primeiro passo do processo consiste dividir o mapa em uma *grid* com granularidade arbitrária. Depois, calcular a visibilidade para um ponto em cada célula da *grid*. É obtida uma matriz de valores de visibilidade. O próximo passo é encontrar mínimos locais e armazená-los ordenados por valor numa fila, com referências aos respectivos pontos. Esta fila contém os pontos promissores em ordem. Ver exemplo na Figura 2.

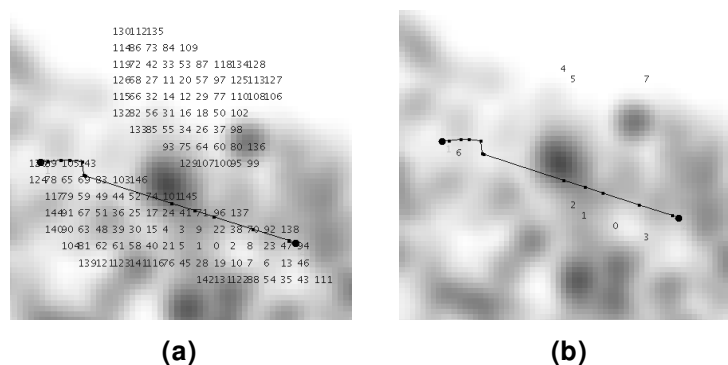


Figura 2. As imagens (a) e (b) representam pontos promissores para um cenário. Em (a) é exibido um número para cada nó da *grid*. O número representa a posição do nó na fila ordenada por valor de visibilidade. Os números de nós com valor de visibilidade muito ruim, comparados com a obstrução da rota original, em preto, estão omitidos. Em (b) estão visíveis apenas os mínimos locais da função de visibilidade.

As rotas geradas pelo Mapa de Visibilidade podem ter apenas um ponto intermediário. Uma vantagem deste método é que ele avalia possibilidades no mapa inteiro e gera muitas rotas estimadas para um cenário.

3.2. Grafo de Visibilidade

Uma variação da estratégia anterior é calcular o Grafo de Visibilidade [O'Rourke, 2003] usado para encontrar menores caminhos desviando de obstáculos poligonais em um contexto mais genérico, ou seja, independente de uma malha viária. Num Grafo de Visibilidade, os vértices são todos os vértices dos obstáculos mais a origem e o destino e as arestas são todas as linhas que ligam dois vértices do grafo exceto as que cortam obstáculos. Sabemos que o menor caminho do grafo é também o menor caminho entre a origem e o destino que desvia de obstáculos.

Neste trabalho, experimentamos usar a mesma estratégia em nosso contexto particular para encontrar pontos promissores. Para obter obstáculos poligonais a partir da obstrução do mapa, definimos um valor de obstrução limite que separa as células do mapa que farão parte de obstáculos das outras. Tendo estas células identificadas, usamos um algoritmo que caminha pela borda de agrupamentos de células e define polígonos. A partir dos polígonos, é construído o Grafo de Visibilidade, e seu menor caminho é calculado. Os vértices contidos no menor caminho compõem uma rota estimada.

Note que os pontos promissores gerados por esta técnica tangem os obstáculos. É necessário definir uma densidade limite “frouxa” o bastante para que os pontos não fiquem em áreas obstruídas.

Uma vantagem desta técnica é que pode gerar uma rota estimada com múltiplos pontos intermediários, o que não é possível com o Mapa de Visibilidade.

3.3. Grafo de Caminhos Livres

Em [Lima, 2008], um Grafo de Caminhos Livres é usado num passo intermediário de um algoritmo de cálculo de menor caminho desviando de obstáculos como meio de demonstrar a existência de, pelo menos, um caminho entre a origem e o destino.

Os vértices de um Grafo de Caminhos Livres são equidistantes dos obstáculos diretamente acima e abaixo. Os caminhos neste grafo, portanto, tendem a passar longe dos obstáculos. Por esse motivo, experimentamos este método em nosso trabalho, como uma alternativa ao Grafo de Visibilidade, para encontrar pontos promissores.

4. Um algoritmo heurístico para rotas desobstruídas em SWMD

Feitas as definições dos principais conceitos e das heurísticas de geração de rotas estimadas, podemos descrever a idéia geral do algoritmo. Seus passos estão descritos no Quadro 1. Cada um dos métodos de cálculo de pontos promissores para rotas estimadas apresentados é usado. O resultado deles é inserido na fila de prioridades. Já que o gargalo do algoritmo está na requisição remota ao SWMD, vale à pena pagar este custo adicional em processamento local para obter mais rotas estimadas alternativas dentre as quais a melhor será escolhida.

No início do algoritmo, o grafo de rotas contém apenas dois vértices: origem P_o e destino P_d . Na primeira iteração, é calculada a rota real $\text{rota}(P_o, P_d)$, se a rota não for tolerável, são calculadas rotas estimadas ligando a P_o e P_d . Destas, uma é escolhida para gerar a rota real da segunda iteração. Para cada ponto da rota estimada escolhida um vértice é adicionado ao grafo .

Nas iterações seguintes, cada par de pontos adjacentes da rota estimada escolhida na iteração anterior é considerado origem e destino de um problema de cálculo de rota com desvio de obstáculos menor. Novas rotas estimadas são calculadas para o par de pontos e assim sucessivamente. Se, por exemplo, uma rota estimada com três pontos P_1 , P_2 e P_3 gerar uma rota real que é desobstruída apenas entre P_1 e P_2 , este resultado parcial é aproveitado e o algoritmo passará a se preocupar com a rota entre P_2 e P_3 .

O algoritmo A^* [Fu, 2006] atribui a cada nó i uma função de avaliação heurística $F_{(i)} = L_{(i)} + e_{(i,d)}$, onde $L_{(i)}$ é o peso do caminho até então avaliado da origem até o nó i e $e_{(i,d)}$ é uma estimativa do peso do caminho não avaliado ainda entre o nó i e o destino. De maneira análoga, o algoritmo para rotas desobstruídas em SWMD, ao calcular rotas estimadas em sub-problemas, ou seja, com origens e destinos P_o' e P_d' , leva em conta no cálculo do peso de cada rota estimada os trechos de rotas reais de menor peso que ligam P_o a P_o' e P_d' a P_d .

Sejam:

- G, o grafo de rotas descrito em 2.2;
- Q, a fila de rotas estimadas, uma fila de prioridade ordenada pelo peso das rotas definido em 2.2.;
- i_{\max} , o limite máximo de iterações definido arbitrariamente;
- o e d, os pontos de origem e destino respectivamente; e
- $\text{rota}(P)$, a função de rotas oferecida pelo SWMD, onde P é uma sequência de pontos.

O algoritmo inicia:

- R_1 é obtida de $\text{rota}(o,d)$.
- Para cada iteração i :
 - R_i é inserida em G;
 - S, a rota de peso mínimo em G, é calculada, como em 2.2;
 - Se S for tolerável, conforme definido em 2.3, então:
 - Retorna S;
 - Fim.
 - Senão:
 - E, um conjunto de rotas estimadas, é calculado a partir de R_i usando uma das estratégias apresentadas em 3;
 - Os elementos de E são inseridos em Q;
 - e_{\min} , a rota estimada de menor peso em Q, é retirada;
 - R_{i+1} , é obtida de $\text{rota}(e_{\min})$;
 - Se $i = i_{\max}$:
 - Não foi encontrada rota desobstruída;
 - Fim.
 - Senão:
 - Continua na iteração $i+1$

Quadro 1. Pseudo-código do algoritmo de cálculo de rotas desobstruídas

5. Avaliação

A técnica apresentada foi utilizada em Wikicrimes, um sistema web baseado em GoogleMaps que mapeia eventos de criminalidade tanto a partir de dados providos por entidades governamentais como da colaboração de usuários comuns, fornecendo uma visão geográfica da distribuição de criminalidade em qualquer região do globo. Uma das funcionalidades do Wikicrimes é a apresentação da concentração criminal em um mapa de calor, que é representado internamente por uma matriz de valores reais.

Para implementar uma funcionalidade de rotas seguras no Wikicrimes, foi aplicada a técnica apresentada neste trabalho. Uma rota é considerada segura se evitar regiões de alta criminalidade. Fazendo correspondência com o problema descrito na seção 2, a função de obstrução é representada pela matriz do mapa de calor no Wikicrimes. O *GoogleMaps*, disponibiliza em sua API, uma função de cálculo de rotas que recebe como parâmetro uma lista de pontos, assim como $r(x)$ descrita na mesma sessão.

Fizemos testes no Wikicrimes com 50 cenários em 13 cidades diferentes. Cada cenário tem uma origem e um destino para os quais a o SWMD retorna uma rota obstruída. O critério de avaliação adotado é o número de requisições feitas ao SWMD até uma rota satisfatória ser obtida. O julgamento do quão satisfatória é uma rota é feito manualmente. Consideramos que o algoritmo obteve sucesso em um cenário se fizer no máximo 20 requisições.

Os gráficos nas Figuras 3 e 4 relacionam cada cenário ao número de requisições feitas. As barras marcam a qualidade das rotas durante o progresso do algoritmo. O tom mais escuro indica que as rotas conhecidas durante a iteração correspondente são todas obstruídas. O intermediário indica que houve evolução, ou seja, foi encontrada alguma rota melhor do que a original, mas ainda não satisfatória. O tom mais claro indica a descoberta de uma rota boa. As marcas pretas indicam evoluções reconhecidas pelo algoritmo com a função peso. O gráfico está ordenado pelos resultados mais satisfatórios, da esquerda para a direita.

Por exemplo, a Figura 3a mostra que, segundo a avaliação visual feita, o cenário “1. Treze de Maio” (próximo à metade do eixo de cenários) tinha uma rota obstruída antes da execução do algoritmo. Na segunda iteração começaram a aparecer rotas que desviam de alguns obstáculos, mas não desviam de todos ou são muito longas. E na décima iteração foi descoberta uma rota satisfatória. Segundo a avaliação do algoritmo (marcas pretas) houve evoluções nas iterações 1, 4, 7 e 10, ou seja, o menor caminho no grafo de rotas ponderadas pela função peso mudou nestas iterações. O resultado visto nesta figura refere-se ao algoritmo que usa as três estratégias descritas em 3: mapa de visibilidade, grafo de visibilidade e caminhos livres.

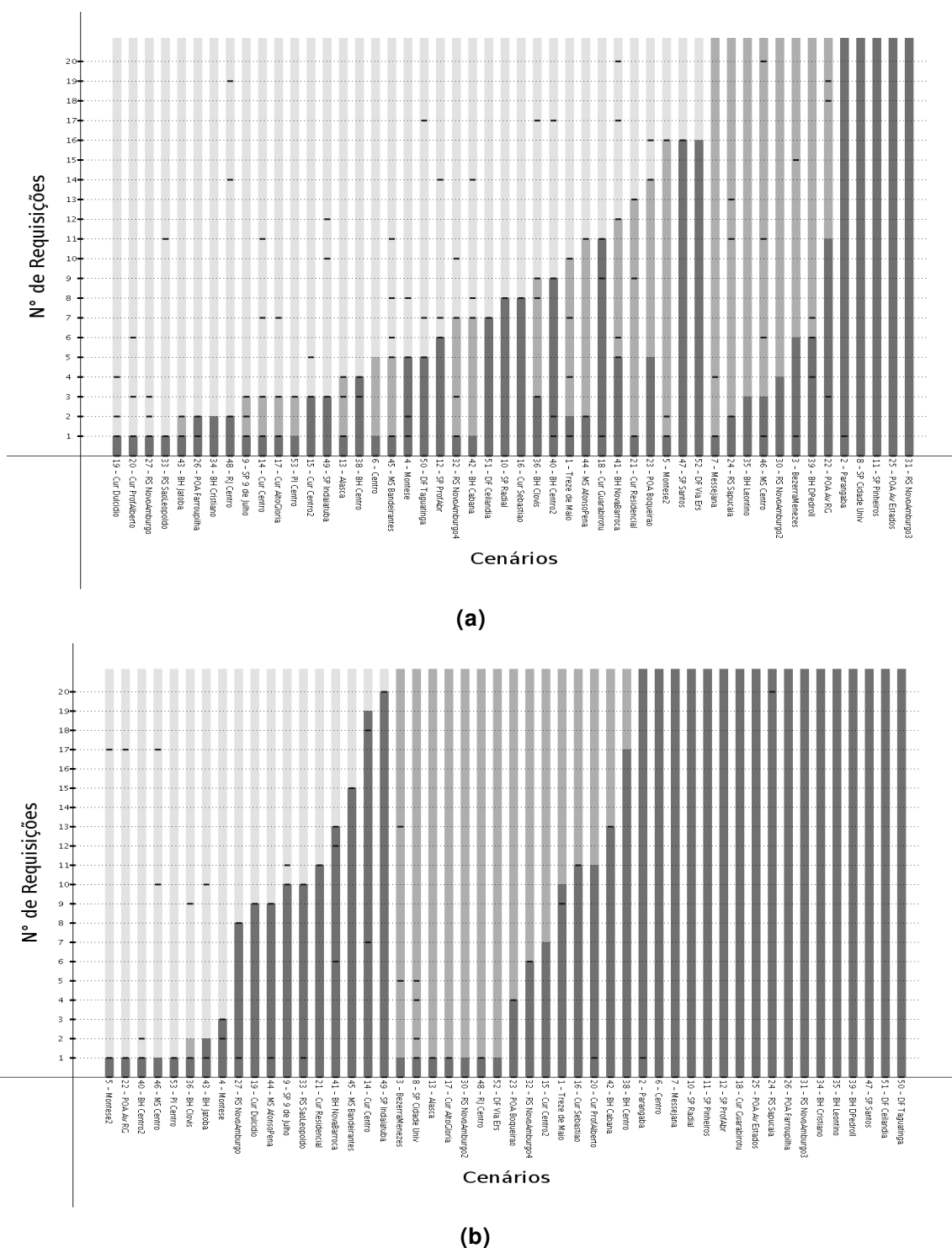


Figura 3. (a) resultado obtido com uso das três heurísticas propostas. (b) resultado obtido com pontos promissores aleatórios

A Figura 3b mostra o resultado de quando os pontos promissores são gerados aleatoriamente. Serve como parâmetro de comparação mínimo (*baseline*).

A primeira versão, que usa a melhor estratégia disponível, obteve sucesso em 74% dos cenários. O método proposto melhorou a rota em todos os cenários. A segunda obteve em apenas 36%, um resultado bem menos satisfatório, conforme esperado.

É importante observar que pode não ser possível encontrar uma rota satisfatória em alguns cenários. Considere por exemplo um cenário em que a própria malha viária oferece poucas alternativas de caminhos e todas são obstruídas entre a origem e o destino. Neste caso, não seria possível encontrar rota desobstruída.

Decidimos fazer uma variação do método aleatório. Continuamos gerando pontos aleatórios mas com uma fila comum no lugar da fila de prioridades, ou seja, não ordena as rotas estimadas antes de escolher uma para mandar ao SWMD. Os resultados foram ainda inferiores aos obtidos na primeira estratégia, mas aumentos consideráveis foram conseguidos. Obteve-se cerca de 56% de sucesso. Atribuímos este resultado ao fato de que a fila de prioridades frequentemente escolhe rotas estimadas que tenham trechos reais, induzindo o algoritmo a insistir em aproveitar um trecho real encontrado anteriormente e buscar alternativas para outra parte da rota que esteja em uma região difícil, ao invés de descartar o trecho real e tentar encontrar rotas totalmente diferentes.

No futuro, pretendemos experimentar variações da fila de prioridades que priorizem a escolher de rotas inteiramente alternativas.

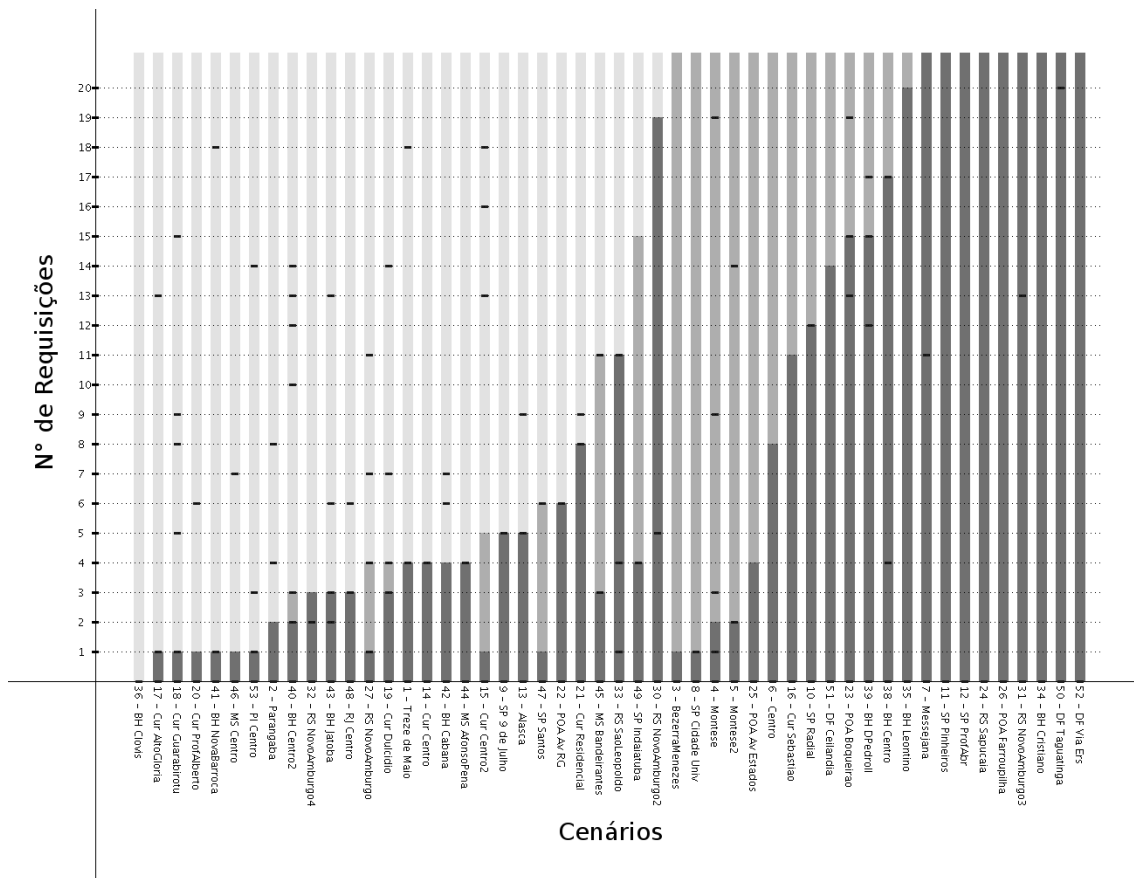


Figura 4. Estimando rotas com pontos aleatórios e sem usar fila de prioridade para ordenar as estimativas.

6. Trabalhos relacionados

Buscas heurísticas para cálculo de caminhos mais curtos é um tema extensivamente estudado em IA e pesquisa operacional. Uma revisão e classificação destes trabalhos pode ser encontrada em [Fu, 2006].

Pesquisas que tratam do desenvolvimento de planejamento de rotas para robôs compartilham igualmente muita similaridade com o problema que estamos atacando. Um exemplo relevante pode ser visto em [Sánchez Miralles, 2004], onde obstáculos são modelados por mapas de distribuição probabilística contínua. O problema e, em especial, a forma como os obstáculos são modelados têm muito em comum com a forma como modelamos a criminalidade no trabalho aqui exposto. No entanto, há uma diferença fundamental no que diz respeito à liberdade de movimento, já que somos limitados a traçar as rotas pelas ruas que só são conhecidas gradualmente a partir das chamadas ao SWMD.

Outros trabalhos relacionados neste contexto buscam usar o conceito de visibilidade como se o ambiente fosse um campo 2d vigiado por sensores e com a presença de obstáculos. [Senjuti, 2007], por exemplo, utiliza formas especializadas dos diagramas de Voronoi e Delaunay para a geração de potenciais campos visíveis.

7. Conclusão

Apresentamos o progresso alcançado por nossa pesquisa em estratégias para encontrar rotas que desviam de obstáculos usando Serviços Web de Mapas Digitais. Foram descritas três heurísticas para estimativa de rotas alternativas visando minimizar as requisições remotas e também a estrutura de algoritmo que as põe em prática. Os testes realizados apontam um sucesso significativo e o potencial de melhoria. A aplicação do método está sendo feita no sistema WikiCrimes e está permitindo calcular uma rota segura a partir do aparelho celular.

Diversas linhas de trabalho futuro serão seguidas. A principal delas é inserir um fator estocástico no método, de forma que problemas de máxima local sejam minimizados. O *ranking* da fila de prioridade é um desses alvos para se introduzir a aleatoriedade.

Agradecimentos

Este trabalho foi parcialmente financiado pelo CNPq (559977/2010 e 306266/2008) e FUNCAP.

Referencias Bibliográficas

- Cormen, Thomas H. et al. (2001). “Section 24.3: Dijkstra's algorithm”. In: “Introduction to Algorithms”, 2nd ed. MIT Press and McGraw Hill. pp. 595-601.
- Dobrin, Adam (2005). “A Review of Properties and Variations of Voronoi Diagrams”, Whitman College.
- Fu, L., Sun D. and Rilett L. R. (2006) “Heuristic shortest path algorithms for transportation applications: state of the art”, In: Computers and Operations Research, Volume 33, Issue 11, Elsevier Science Ltd., Oxford, UK.

- Furtado, Vasco, Ayres, Leonardo, de Oliveira, Marcos, Vasconcelos, Eurico, Caminha, Carlos, D'Orleans, Jonathan, Belchior, Mairon. (2010) "Collective Intelligence: The WikiCrimes System. Information Science", v(180), 1.
- Ishikawa et al (1991): "Map Navigation Software of the Electro Multivision of the '91 Toyota Soarer". In: IEEE Int. Conf. Vehicle Navig.Inform. Syst. pp. 463–473.
- Joseph O'Rourke (1993) "Computational geometry column 18", In: ACM SIGACT News, Volume 24, Issue 1, New York NY, USA.
- Lima, Natan Costa and Ferreira, Carlos Eduardo (2008). "Algoritmos e estrutura de dados para problemas de deslocamento no plano", In: IV Simpósio de Iniciação Científica e Pós-Graduação, Instituto de Matemática e Estatística, USP, São Paulo, SP, Brasil.
- McLafferty (2000). "Identification, development and implementation of innovative crime mapping techniques and spatial analysis". Washington, D.C: U.S. Department of Justice, p.27.
- Merril Duane. "Mashups: The new breed of Web app" at <http://www.ibm.com/developerworks/xml/library/x-mashups/index.html>
- Rosen, Kenneth H. "Handbook of Discrete and Computational Geometry and its Applications".
- Rouse, J., Bergeron, S.J., Harris, T.M., "Participating in the geospatial web: collaborative mapping, social networks and participatory GIS," in Scharl, A., Tochtermann, K. (Eds.), The Geospatial Web: How Geobrowsers, Social Software and the Web 2.0 are Shaping the Network Society, Springer, New York, NY, pp. 153-8, 2007
- Sánchez Miralles, A. and Sanz Bobi (2004), M. A.: "Global Path Planning in Gaussian Probabilistic Maps", Journal of Intelligent and Robotic Systems, Volume 40, Issue 1, Kluwer Academic Publishers, Hingham, MA, USA.
- Senjuti, B. R., Gautam, D. and Sajal D. (2007) "Computing Best Coverage Path in the Presence of Obstacles in a Sensor Field", In: WADS.