

TerraNetwork: Sistema de Análise de Redes Urbanas

Geraldo Ribeiro Filho¹
Reinaldo Gen Ichiro Arakaki²
Marcio Paim de Aquino²
Luiz Antonio Nogueira Lorena²

¹Instituto de Pesquisas em Tecnologia da Informação
Avenida São Luís, 86 – cj. 192 - 01046-000 - São Paulo – SP – Brasil
gerald@ipti.org.br

² Instituto Nacional de Pesquisas Espaciais - INPE
Caixa Postal 515 - 12227-010 - São José dos Campos - SP, Brasil
{reinaldo.arakaki, marcio.paim,lorena}@lac.inpe.br

Abstract. This paper presents the current state of the TerraNetwork project, its structure, characteristics, functionalities and the implemented combinatorial optimization algorithms, used to solve urban street networks related to location and transportation problems. The street clusters functionality and the Chinese Postman are particularly detailed. Ongoing development and some future work possibilities and objectives are also described.

Palavras-chave: Geographic Information Systems, combinatorial optimization, sistema de informações geográficas

1. Introdução

Existem muitos problemas práticos de tomada decisão envolvendo a instalação de facilidades como hospitais, fábricas, centros de distribuição de produtos, depósitos, antenas de telefonia móvel, etc. Esses problemas de localização são ligados a problemas de transporte cujo objetivo é otimizar o uso dessas facilidades. Problemas desse tipo podem ser muito difíceis de resolver e atraem especial interesse quando ocorrem em cidades com grandes redes viárias, exigindo o uso de técnicas e algoritmos da área de otimização combinatória.

O desenvolvimento de sistemas computacionais com interfaces gráficas para processamento de dados geograficamente referenciados vem influenciando de maneira crescente diversas áreas de atividades como cartografia, mapeamento, planejamento urbano e regional, e transportes, entre outras. Esta tecnologia torna possível a automatização de tarefas realizadas manualmente e facilita a realização de análises complexas, através da possibilidade de integração de dados de diversas fontes em bancos de dados geocodificados. Sistemas com tais características são denominados Sistemas de Informações Geográficas (SIG).

Redes viárias são representadas nos SIG através de linhas, normalmente formadas por segmentos de retas, e intersecções. Dados e características dos objetos reais (em geral, ruas e avenidas) são associados a esses elementos geométricos como atributos, e também armazenados em bases de dados. Pontos de interesse como escolas, hospitais, ou quaisquer objetos localizados sobre a malha viária também são representados em SIG através de pontos e atributos a eles associados.

Sendo assim, a análise de redes viárias usando as potencialidades dos SIGs tornou-se uma tecnologia eficiente no tratamento de informações com características espaciais. A vantagem em usar um SIG está na habilidade de associar a cada arco e nó da rede um conjunto de atributos que podem ser manipulados e visualizados graficamente. Além disto, alguns algoritmos bem conhecidos, como por exemplo: para minimizar a distância e o tempo

percorridos em uma rota ou determinar a rota de menor custo entre dois pontos de interesse, estão imediatamente disponíveis para integração a esses sistemas.

TerraNetwork é um projeto com objetivo de criar um protótipo de sistema de informação para aplicar técnicas e algoritmos de otimização combinatória na resolução de problemas de localização e transportes, num ambiente de SIG. Usando a linguagem C++, um conjunto de classes foi criado com base no TerraLib, um modelo de banco de dados espaciais com uma biblioteca classes e funções para acesso e manipulação, gratuitamente disponível, desenvolvido no Instituto Nacional de Pesquisas Espaciais (INPE). As classes do TerraNetwork usam também uma biblioteca de nome Boost Graph Library (BGL), também gratuitamente disponível, que dispõe de eficientes implementações de vários algoritmos para processamento de problemas relacionados a grafos.

As seções seguintes apresentam o software de suporte utilizado, as classes do projeto TerraNetwork, sua estrutura e funcionalidades implementadas, particularmente o algoritmo para formação de agrupamentos de ruas, e o Problema do Carteiro Chinês para formação de rotas que percorrem totalmente esses agrupamentos. Finalmente, o estágio atual de desenvolvimento e os esforços futuros são comentados.

2. Software de suporte

O projeto TerraNetwork faz uso de duas bibliotecas de classes e rotinas como software de suporte, TerraLib e Boost Graph Library, brevemente descritas abaixo.

2.1 TerraLib

Terralib (<http://www.terralib.org>) é um projeto desenvolvido no âmbito do INPE e propõe um esquema de banco de dados para armazenar dados de entidades com características espaciais e atributos a elas associados. Também é disponibilizada de forma gratuita uma biblioteca de classes C++ para manutenção dos dados no banco

Tipicamente, nos SIG os dados são organizados em camadas, e no modelo do Terralib as camadas têm a elas associadas “vistas” e “temas”, usados na visualização dos dados. As rotinas do Terralib permitem a conexão a diversos SGBD (Sistemas Gerenciadores de Bancos de Dados) para criar e manter o banco de dados.

Para visualização dos resultados obtidos nos testes foi utilizada uma ferramenta desenvolvida também no INPE, chamada TerraView, que pode ser gratuitamente obtida em <http://www.dpi.inpe.br/terraview/index.php>.

2.2 Boost Graph Library e Grafos de Ruas

Os algoritmos de otimização combinatória usados para resolver problemas associados a redes agem sobre estruturas na forma de grafos. As classes TerraNetwork lêem a camada de ruas do banco Terralib e criam um grafo direcionado $G=(V, E)$, em que cada vértice $v_i \in V$ corresponde uma intersecção de ruas, e cada aresta $e_i \in E$ corresponde a um segmento de rua entre duas intersecções. O sentido das arestas é obtido a partir de um atributo específico das ruas que armazena um código indicando a direção do tráfego e cujo nome é especificado como parâmetro dos métodos.

Uma biblioteca chamada Boost Graph Library (BGL), gratuitamente disponível em <http://www.boost.org>, possui uma série de algoritmos para tratamento de grafos implementados de modo eficiente. Foram utilizadas alguns métodos desta biblioteca.

3. Modelo de classes e suas funcionalidades

Como o objetivo é implementar funcionalidades para tratamento de diversos problemas sobre redes, foi desenvolvido um modelo simples de classes que permite a representação de um grafo criado a partir de uma malha viária e de pontos de interesse, e a implementação de funcionalidades através de métodos dessas classes. A linguagem escolhida, para maior adequação ao software de suporte foi C++ e o ambiente de desenvolvimento utilizado foi Microsoft Visual C++.

O modelo de classes do projeto TerraNetwork possui oito classes relacionadas entre si como mostrado na Figura 1. A classe principal do modelo é chamada *Net*, e possui vários métodos que implementam algoritmos para resolver problemas sobre redes.

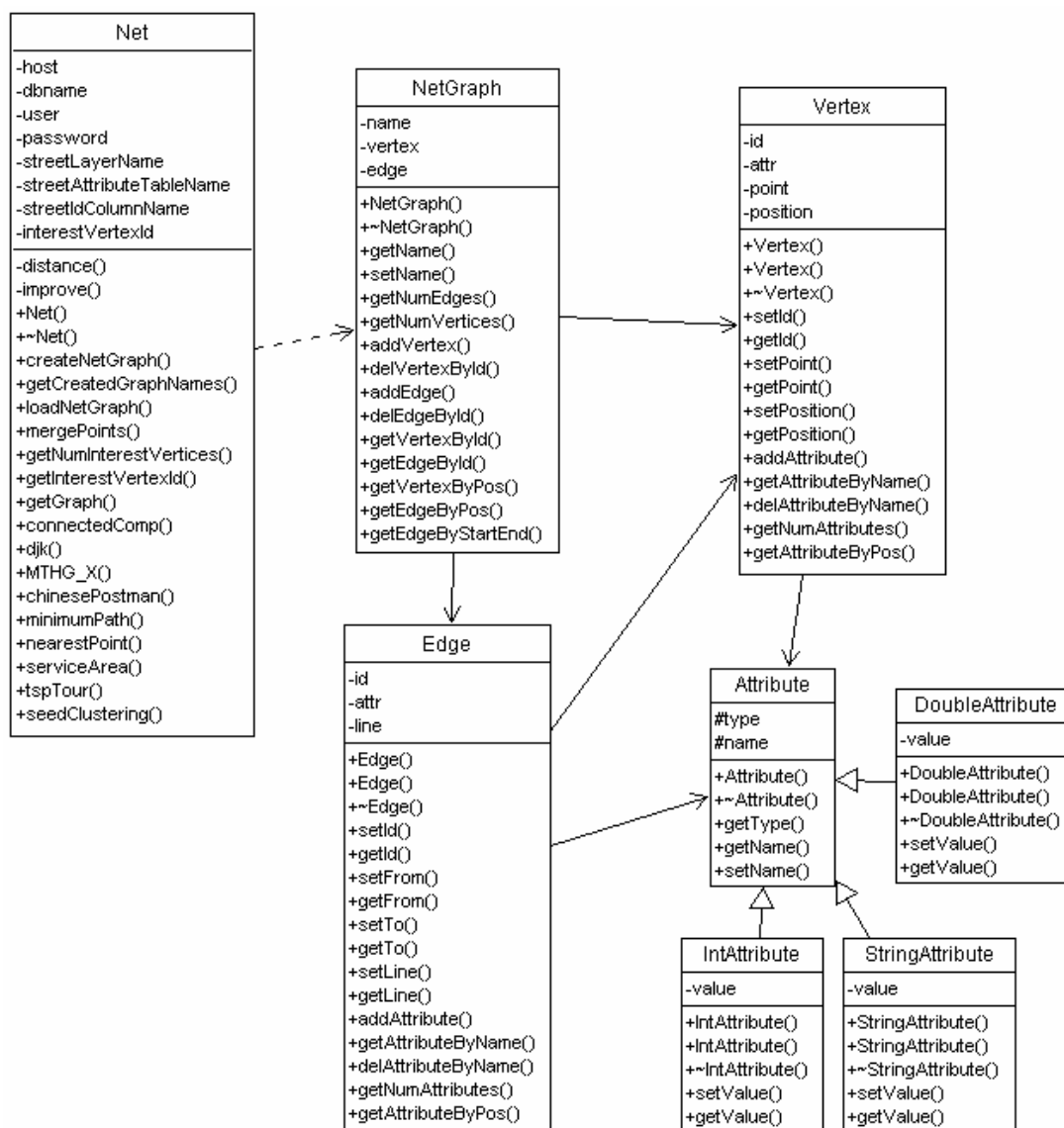


Figura 1: Modelo de Classes TerraNetwork.

A classe *Net* possui métodos que permitem a criação de um grafo a partir de uma camada de dados geográficos com ruas e suas intersecções. As ruas se tornam arestas e as intersecções os vértices do grafo. A classe *Net* passa a ter então uma referência a um objeto de uma classe chamada *NetGraph* que representa o grafo. Esta referência pode ser obtida também através da

leitura dos dados de um grafo já criado anteriormente e armazenado no próprio banco. A classe *Net* também possui um método de nome *mergePoints* que permite a inserção de pontos de interesse como novos vértices do grafo, à partir de uma camada de dados geográficos com esses pontos armazenada no banco. Objetos da classe *NetGraph* por sua vez possuem coleções de referências a objetos das classes chamadas *Edge* e *Vertex*, que representam as arestas e os vértices do grafo. Cada objeto dessas duas classes possui uma coleção de referências à classe *Attribute*, superclasse de três outras classes, cada uma representando um atributo de um determinado tipo de dados, chamadas *DoubleAttribute*, *IntAttribute* e *StringAttribute*.

Uma típica aplicação feita com as classes acima descritas inicialmente cria um objeto da classe *Net*, passando toda informação necessária para conectar a um gerenciador de bancos de dados como parâmetros do construtor. Em seguida, a aplicação escolhe e carrega um dos grafos já criados com o método *loadNetGraph*, ou cria um grafo novo usando o método do *createNetGraph*. A criação do grafo exige diversas informações a respeito da camada de ruas já presente no banco e dos seus atributos. Opcionalmente, a aplicação pode então inserir pontos de interesse como novos vértices do grafo. Finalmente, a aplicação pode invocar algum método de serviço da classe *Net* para resolver um determinado problema a respeito dos pontos e das ruas modelados na estrutura do grafo. Cada método de serviço grava sua solução na forma de uma nova camada de dados no banco. Um software de visualização como, por exemplo, o TerraView pode ser então utilizado.

A seguir estão descritas as funcionalidades implementadas e os algoritmos utilizados.

3.1.Caminho Mínimo

Uma das tarefas as mais comuns executadas com SIG é a determinação do caminho de custo mínimo entre dois vértices de uma rede. Na classe *Net* foi implementado um método chamado o *minimumPath* que executa esta tarefa. Este método aciona uma rotina de BGL que executa o clássico algoritmo de Dijkstra (Dijkstra, 1959). O resultado do caminho mínimo é visualizado através do Terraview. A **Figura 2** mostra parcialmente a tela do TerraView com um caminho entre dois pontos da cidade de São José dos Campos.

Uma outra aplicação dos SIG pode requerer a determinação do caminho de custo mínimo entre diversos pontos do interesse na rede, de modo que cada ponto seja visitado uma única vez. Este problema é equivalente ao clássico Problema do Caixeiro Viajante (PCV) (Lawler *et al.*, 1986), cujo objetivo é encontrar o trajeto de custo mínimo, partindo de um ponto de origem, para visitar exatamente uma vez cada um dos outros pontos de interesse, e retornar à origem. Este problema é uma abstração de diversas aplicações em transporte e logística como, por exemplo, a determinação de rotas de ônibus escolares, rotas para serviços de entregas de produtos, rotas para oferecer transporte a pessoas com necessidades especiais etc.



Figura 2: Caminho de Custo Mínimo.

3.2 Facilidade mais Próxima e Área de Serviço

Dois aplicações imediatas dos caminhos de custo mínimo são também implementadas como métodos da classe *Net*: encontrar a facilidade a mais próxima a uma posição em uma rede, e determinar uma área de serviço em torno de uma posição. O termo “facilidade mais próxima” se refere a um fornecedor de um determinado tipo do serviço que está mais próximo de uma dada localização.

O termo “acessibilidade” refere-se à facilidade de se chegar a uma certa localização e pode ser medida em tempo de viagem ou distância. Examinar a acessibilidade de pontos disponíveis pode ser uma tarefa muito importante no momento de decidir onde instalar um novo negócio que atenda uma certa área de serviço.

A **Figura 3** mostra parcialmente a tela do TerraView com o resultado de uma instância de teste com dois pontos de interesse dentro da região demarcada. O ponto mais ao centro da região é a origem e o outro é o único entre os pontos de interesse que se verificou estar dentro do limite estabelecido no teste.



Figura 3: Polígono de Área de Serviço.

3.3 Agrupamentos de Ruas

Outra funcionalidade implementada permite dividir a rede em regiões de acordo com restrições de capacidade. Este procedimento pode ser útil em aplicações práticas como a coleta de produtos por veículos ou a aquisição de dados por empregados do serviço público.

Osman e Christofides (1994) descrevem o Problema de Agrupamento Capacitado como o problema em que um conjunto de elementos deve ser dividido em um dado número de subconjuntos. O objetivo é encontrar os pontos centrais, e conseqüentemente os subconjuntos, que minimizam a soma total das distâncias para todos os subconjuntos. Este problema é NP-Completo e heurísticas são requeridas.

Para encontrar a solução deste problema é necessário determinar o número de agrupamentos desejados e as posições iniciais dos pontos centrais.

Esses pontos centrais são chamados de “sementes” para formar os agrupamentos. A atribuição de uma rua a uma semente é baseada na distância da semente à rua. As demandas das ruas são dadas por um de seus atributos e cada semente tem uma capacidade dada também por um de seus atributos.

A heurística de busca local move cada semente para o vértice vizinho que produz a maior redução de custo total das distâncias das ruas associadas à semente. A vizinhança é limitada por um raio de distância passado como um parâmetro à rotina. Este processo é repetido até que nenhuma melhoria seja obtida e as sementes se estabilizem. Técnicas similares foram usadas em outros trabalhos (Arakaki, 2002) (Lorena e Pereira, 2002).

As distâncias são calculadas com o algoritmo de Dijkstra, e a associação das ruas às sementes, respeitando suas capacidades, é feita por uma heurística desenvolvida para resolver o chamado Problema Generalizado de Atribuição (Martello e Toth, 1990).

A **Figura 4** mostra parcialmente uma tela do TerraView com três agrupamentos e seus respectivos centros em sua localização final.



Figura 4: Três Agrupamentos de Ruas.

3.4 Problema do Carteiro Chinês

Como abstração consideramos o caso de um carteiro que seja responsável pela entrega do correio em uma determinada área da cidade. O carteiro deve sempre começar sua rota de entrega na posição onde a correspondência é armazenada, deve passar por cada rua em sua área e, eventualmente, deve retornar à origem. O problema que se levanta é encontrar a rota que minimiza a distância total que o carteiro percorre. Este problema é conhecido como o Problema do Carteiro Chinês (PCC), pode ser formulado para grafos direcionados e não direcionados, e é NP-Completo para gráficos mistos (Larson e Odoni, 1981).

Na classe *Net* foi criado um método chamado *chinesePostman* que executa um algoritmo para grafos não direcionados. Este algoritmo usa os conceitos relacionados ao trabalho de Euler. Um ciclo de Euler (ou euleriano) é um percurso que passa por cada aresta de um grafo exatamente uma vez, e começa e termina no mesmo vértice. Nos trabalhos de Euler verifica-se que um grafo conexo possui um ciclo euleriano se, e somente se, o grafo tem todos os vértices com grau par. O ciclo euleriano é uma solução do PCC sobre o grafo.

TerraNetwork usa uma rotina escrita em C, disponível em <http://elib.zib.de/pub/Packages/mathprog/matching/text.en.html> de forma gratuita, para encontrar o *matching* de maior custo num grafo não direcionado (Gabow, 1973). Apesar da

rotina originalmente ser feita para o *matching* máximo, ela possui um parâmetro que permite sua configuração para encontrar o *matching* mínimo.

A **Figura 5** mostra parcialmente a tela do TerraView com uma malha de ruas e o número de ordem de dos segmentos de rua que formam a rota que percorre cada segmento com menor custo total.

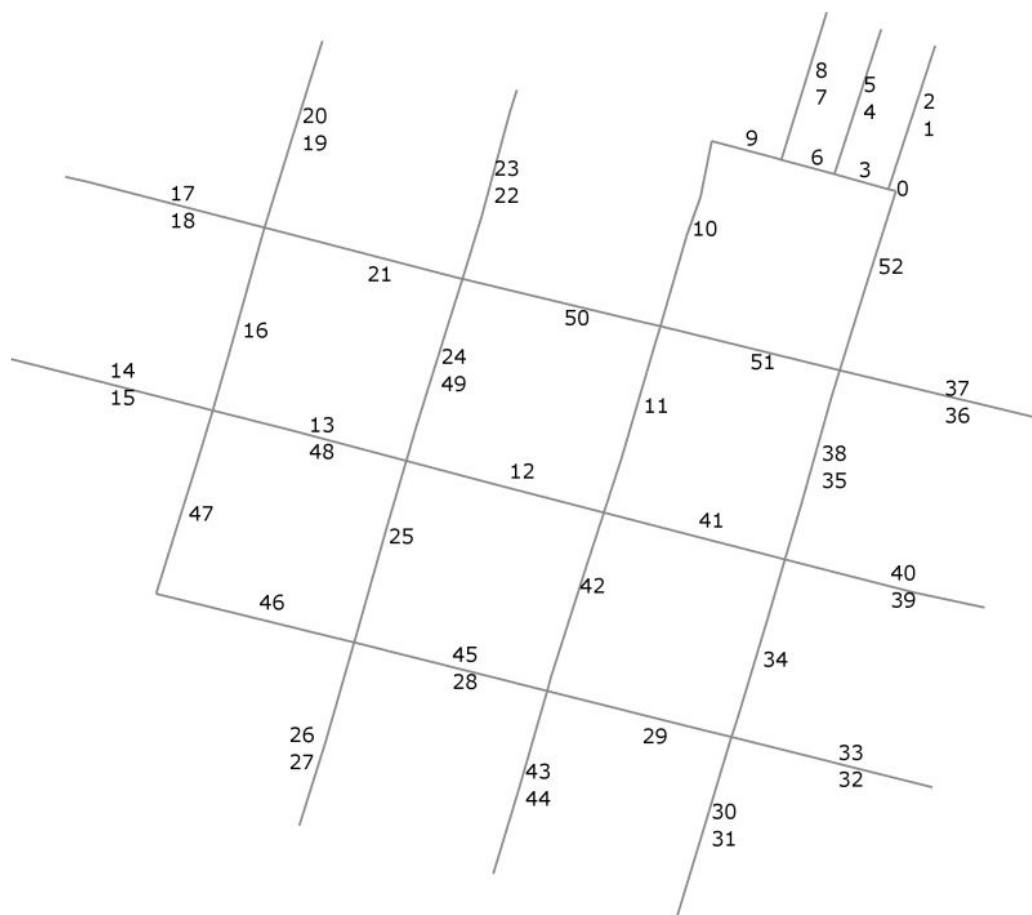


Figura 5: Percurso com Custo Mínimo.

4. Conclusão

Este trabalho apresentou o projeto TerraNetwork, um protótipo de SIG aplicável a problemas sobre redes urbanas. O estágio atual do modelo de classes foi apresentado, juntamente com seu software de suporte (TerraLib, TerraView e BGL). Foram descritos as funcionalidades já implementadas e os algoritmos utilizados. O esforço de pesquisa atual está nos problemas de localização e alocação,.

A relevância dos esforços de pesquisa em software que abordam problemas sobre redes urbanas é inquestionável. O objetivo primário deste trabalho foi mostrar a associação dos SIG com as técnicas de otimização combinatória no tratamento dos problemas relacionados a redes viárias.

Referências

- Arakaki, R. G. I. **Heurística de Localização-Alocação para Problemas de Localizacao de Facilidades**. 2002, 90 p. (INPE-9572-TDI/837) Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais – INPE, São José dos Campos,2002.
- Dijkstra, E. W. **A Note on Two Problems with Connection with Graphs**. Numerische Mathematik, 1959, v.1.
- Gabow, H. **Implementation of Algorithms for Maximum Matching on Nonbipartite Graphs**. Ph.D. thesis - Stanford University, 1973.
- Larson R. C.; Odoni, A. R. **Urban Operations Research**. Prentice Hall, 1981.
- Lawler, E. L.; Lenstra, J. K.; Rinnooykan KAN, A. H. G.; Shmoys, D. B. **The Travelling Salesman Problem**. John Wiley and Sons, 1986.
- Lorena, L. A. N.; Pereira, M. A. A Lagrangean/surrogate heuristic for the maximal covering location problem using Hillsman's edition. **International Journal of Industrial Engineering**, 2002, v. 9, pg. 57-67.
- Martello, S.; Toth, P. **Knapsack Problems: Algorithms and Computer Implementations**. John Wiley and Sons, 1990.
- Osman, I. H., Christofides, N. Capacitated Clustering Problems by Hybrid Simulated Annealing and Tabu Search. **Transactions in Operational Research**, 1994, v. 1, pg. 317-336.

A pesquisa relatada neste artigo teve o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq (processo 506840/2004-2).