Solução Exata do Problema de Roteamento de Veículos por Solucionador Genérico

Fernando Bombardelli da Silva
Instituto de Informática – Universidade Federal do Rio Grande do Sul

Introdução

O **problema de roteamento de veículos** (em inglês, *vehicle routing problem*) é um problema de otimização combinatória que objetiva determinar rotas de veículos, buscando soluções que minimizam os custos de operação de tais rotas.

Este problema tem como entrada os dados sobre a frota de veículos disponível e um conjunto de clientes a serem visitados. O objetivo de um solucionador do problema é calcular as rotas ótimas desses veículos, que são aquelas que minimizam o custo (ou tempo) total do serviço, atendendo a todas as requisições, de modo que as restrições do problema sejam satisfeitas. A Figura 1 mostra um exemplo de solução para o problema.

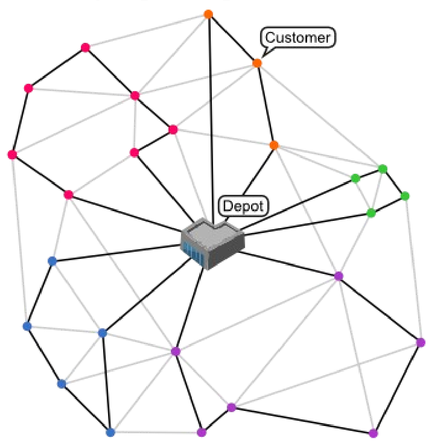

Figura 1. Fonte: http://neo.lcc.uma.es/vrp/wp-content/uploads/vrp.png

Metodologia

Para desenvolver a proposta aqui apresentada, foi utilizado um solucionador genérico de programas lineares e lineares inteiros, que aplica os algoritmos *simplex* e *branch and cut*. O programa em questão se chama GLPK (sigla inglês para *GNU Linear Programming Kit*), um programa de código aberto, parte do projeto GNU da *Free Software Foundation*. O GLPK é distribuído como uma biblioteca dinâmica e uma API (inglês para *Application Programming Interface*) para acesso às suas funções. Essa API é originalmente disponível para a linguagem C, porém existem versões de compatibilidade para aplicações Java. A Figura 2 ilustra a arquitetura da implementação proposta.

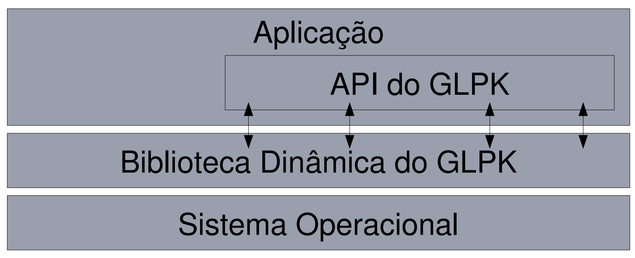

Figura 2

Modelo

A modelagem matemática do problema de otimização pode ser representada pela seguinte fórmula.

$$\begin{aligned} &\textit{minimize} && \boldsymbol{c}^T \boldsymbol{x}, \\ &\textit{sujeito a} && A\boldsymbol{x} \leq \boldsymbol{b}, \\ & && \boldsymbol{x} \geq \boldsymbol{0}. \end{aligned}$$

Onde **x** é o vetor de variáveis de decisão, **c** é o vetor de componentes da função objetivo, *A* é a matriz dos coeficientes dos lados esquerdos das inequações de restrição das variáveis e **b** é o vetor dos valores dos lados direitos dessas.

A partir da leitura da instância do problema, a aplicação cria os vetores **c** e **b**, e a matriz *A*. Então, ela realiza chamadas à API do GLPK. Estas chamadas tem como objetivo: (i) informar os dados ao solucionador, (ii) executar o algoritmo que resolve o problema linear inteiro de maneira exata, e (iii) recuperar o valor das variáveis de decisão **x** que representam a solução ótima para o problema.

Por fim, a aplicação gera uma saída adequada transformando o resultado, ou seja, o vetor **x,** em uma representação das rotas respectivas de cada veículo. A Figura 3 ilustra o fluxo de dados na aplicação.

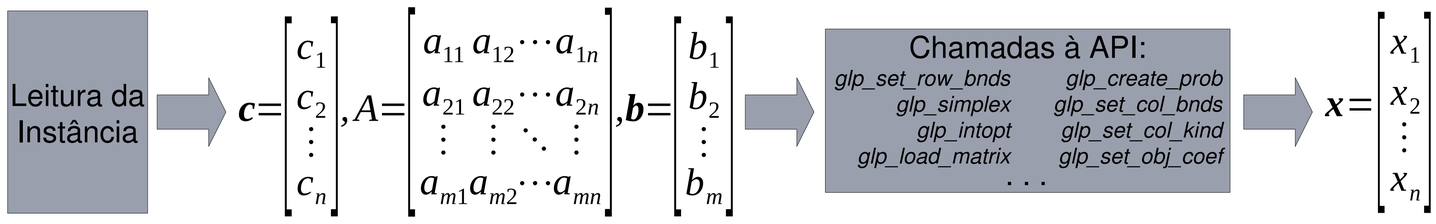

Figura 3

Estudo de Caso

Foi realizado um estudo de caso para resolver o problema do caixeiro viajante assimétrico (ATSP), uma simplificação do problema de roteamento de veículos. Essa versão do problema é formulado matematicamente como segue.

$$\begin{aligned} &\textit{minimize} && \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij}, && \\ &\textit{sujeito a} && \sum_{j \in V} x_{ij} = 1, && \forall\, i \in V, \\ & && \sum_{i \in V} x_{ij} = 1, && \forall\, j \in V, \\ & && u_i - u_j + n\, x_{ij} \leq n-1, && \forall\, i \in V \setminus \{0\}, j \in V \setminus \{0, i\}, \\ & && x_{ij} \in \{0,1\}, && \forall\, i, j \in V, \\ & && u_i \in \mathbb{N}, && \forall\, i \in V \setminus \{0\}. \end{aligned}$$

Onde *V* é o conjunto de locais a serem visitados, d_{ij} indica a distância entre os locais *i* e *j*, e *n* é a cardinalidade do conjunto *V*. *A* variável de decisão x_{ij} tem valor *1* se, e somente se, o veículo atende o local *i* e imediatamente depois *j*. Por fim, u_i é uma variável auxiliar que informa a ordem do local *i* na rota do veículo.

Foram resolvidas **1900** instâncias do problema, com número de locais variando entre 2 e 11. Para todos os casos a aplicação encontrou a solução ótima. A média dos tempos de execução ficou em aproximadamente **0,72 segundos** e o tempo máximo de execução foi de 275 segundos.

Os resultados obtidos pelo solucionador foram comparados com soluções previamente conhecidas.

A Figura 4 mostra um diagrama *box plot*. No eixo X estão representados dois conjuntos de instâncias, que são distinguidos pelo número de clientes. O eixo Y representa a redução da função objetivo (obtida nos experimentos) em relação aos valores das soluções já conhecidas.

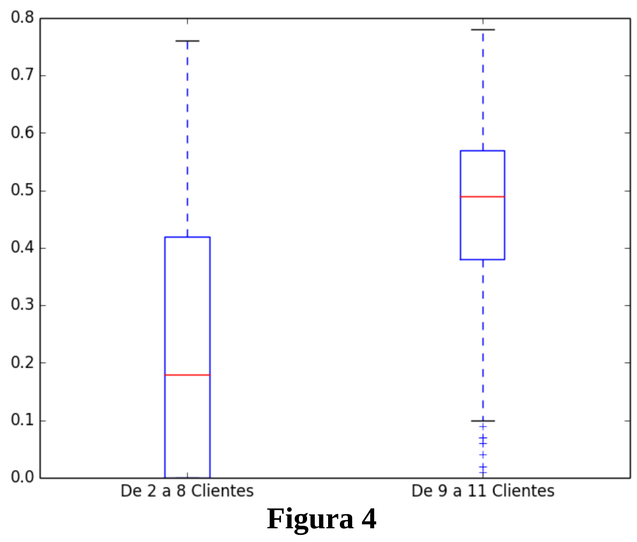

Figura 4

Conclusão

A abordagem proposta neste trabalho apresentou resultados eficientes para a solução de instâncias pequenas de uma forma simplificada do problema de roteamento de veículos.

Trabalhos futuros devem tentar resolver variações do problema que tenham uma frota, potencialmente heterogênea, de diversos veículos e que incluam restrições de capacidade e janelas de tempo para o atendimento a clientes.

Implementação

A proposta foi implementada em *Java SE 7*. O ambiente de execução da aplicação requer a biblioteca dinâmica do *GLPK* (veja https://www.gnu.org/software/glpk/). Foi utilizado um *wrapper* para a API nativa, chamado *GLPK for Java* (veja http://glpk-java.sourceforge.net). Esta ferramenta provê uma vinculação entre a aplicação, que executa sobre a *Java Virtual Machine*, e a biblioteca dinâmica, carregada pelo sistema operacional. Para tal, ela utiliza a *Java Native Interface* (JNI).

Agradecimentos

O trabalho foi desenvolvido no grupo de pesquisa de algoritmos e otimização do Instituto de Informática sob a orientação da **Profª Luciana Buriol**. Agradecimentos especiais também a **Marcelo Friske** e **Carlo Sartori** pelas valiosas ajudas na implementação da proposta e na integração com o ambiente alvo.

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
.inf INSTITUTO DE INFORMÁTICA UFRGS
Salão UFRGS 2016 CONHECIMENTO FORMAÇÃO INOVAÇÃO
paz no plural