

Problema de Roteirização

Gabriel Campos, Luigi Sorrentino, Matheus Pôssas

¹Pontifícia Universidade Católica de Minas Gerais (PUC-MG)
Belo Horizonte – MG – Brasil

1. Introdução

A roteirização é a atividade que tem por fim buscar o melhor trajeto o qual o algoritmo deve encontrar para o problema através de uma malha. Esta busca, geralmente, tem como objetivo minimizar o tempo ou a distância.

Atuar na decisão de roteirização não significa atuar somente sobre o transporte: a extensão do tempo em que o produto está em trânsito influencia no total de estoque da cadeia, além que uma boa escolha das rotas pode melhorar o nível de serviço prestado ao cliente.

2. O Problema

O problema encontrado nesse trabalho foi o deslocamento eficiente dos alunos de ciência da computação da PUC-MG dentre os prédios de uso dos alunos, para melhor deslocamento entre esses prédios.

3. Objetivo

O objetivo do grupo consiste em determinar a menor rota entre os prédios dos alunos do curso Ciência da Computação da PUC-MG, que são os prédios 12, 13, 34, 43 e o Diretório Acadêmico, pois a maioria dos alunos não conhecem o melhor caminho entre esses prédios. Mesmo sendo uma diferença bem pequena, é sempre bom saber o melhor caminho entre esses prédios por via das dúvidas.

Assim, o artigo busca encontrar boas soluções para o problema, ou seja, ajudar os alunos da Ciência da Computação a encontrar o melhor caminho [Fan and Shi 2010], entre os prédios quando precisarem ter o menor deslocamento possível entre os mesmos.

4. Justificativa

O estudo da roteirização e seus processos permite entender como o setor de logística de uma grande empresa consegue atender apropriadamente as necessidades de seus clientes, fazendo com que a mercadoria esteja no lugar certo e no tempo certo, sempre com o menor custo possível e um nível de serviço adequado.

Outro objetivo é aprofundar o conhecimento sobre esse tipo de problema e por causa de sua grande abrangência em contextos atuais com a utilização de redes complexas.

5. Referencial Teórico

Alguns dos trabalhos utilizados como referência do artigo estarão citados aqui, mas não foram muitos encontrados, então citaremos as referências básicas para o entendimento do problema.

No artigo [Fan and Shi 2010] O autor demonstra os principais conceitos para encontrar o caminho mínimo em um grafo, porém, ele o demonstra usando o algoritmo de Dijkstra, e uma forma melhorada de o utilizar para encontrar esse menor caminho.

Já no artigo [Bixby 1992] é demonstrado como é implementado um modelo simplex, para ter um conhecimento básico desse modelo para o entendimento e modelamento da função objetiva e as restrições do problema de roteirização deste trabalho.

O site [Makhorin] foi utilizado pelo grupo para implementar o GLPK que foi a solução do grupo para o problema citado, pois a maior dificuldade foi encontrar e entender os métodos de chamadas do GLPK, aos quais estão bem explicadas no site acima.

O artigo [Thai and Lam 2002] serviu para demonstração e entendimento da plataforma .NET Framework, que serviu como interface para a modelagem do software de otimização, o GLPK.

6. A metodologia

Este trabalho apresenta uma alternativa simples e eficaz para resolver o problema em questão, onde é proposto um modelo matemático geral para representar o problema.

$$\begin{aligned} FO \rightarrow \text{MIN } Z = & (13.09 * x_{06}) + (13.09 * x_{60}) + (24.95 * x_{17}) + (24.95 * x_{71}) + (43.69 * x_{27}) + (43.69 * x_{72}) + \\ & (64.94 * x_{23}) + (64.94 * x_{32}) + (57.19 * x_{214}) + (57.19 * x_{142}) + (33.41 * x_{34}) + (33.41 * x_{43}) + \\ & (27.13 * x_{38}) + (27.13 * x_{83}) + (62.32 * x_{315}) + (62.32 * x_{153}) + (31.40 * x_{49}) + (31.40 * x_{94}) + \\ & (36.02 * x_{59}) + (36.02 * x_{95}) + (42.67 * x_{67}) + (42.67 * x_{76}) + (20.27 * x_{613}) + (20.27 * x_{136}) + \\ & (26.97 * x_{714}) + (26.97 * x_{147}) + (32.04 * x_{89}) + (32.04 * x_{98}) + (13.91 * x_{910}) + (13.91 * x_{109}) + \\ & (28.93 * x_{1011}) + (28.93 * x_{1110}) + (30.26 * x_{1112}) + (30.26 * x_{1211}) + (32.60 * x_{1314}) + \\ & (32.60 * x_{1413}) + (35.74 * x_{1317}) + (35.74 * x_{1713}) + (25.98 * x_{1415}) + (25.98 * x_{1514}) + \\ & (22.58 * x_{1516}) + (22.58 * x_{1615}) + (67.90 * x_{168}) + (67.90 * x_{816}) + (48.42 * x_{1618}) + (48.42 * x_{1816}) + \\ & (36.17 * x_{1620}) + (36.17 * x_{2016}) + (42.37 * x_{1718}) + (42.37 * x_{1817}) + (13.91 * x_{1819}) + (13.91 * x_{1918}) + \\ & (53.51 * x_{1820}) + (53.51 * x_{2018}) + (98.91 * x_{2010}) + (98.91 * x_{1020}) + (38.14 * x_{1921}) + (38.14 * x_{2119}) + \\ & (97.51 * x_{2122}) + (97.51 * x_{2221}) + (37.28 * x_{2211}) + (37.28 * x_{1122}) \end{aligned}$$

Figura 1. Modelo matemático sem as restrições

A metodologia utilizada pelo o grupo na realização desse problema de roteirização, foi primeiramente entender o tema por completo e todas as suas exigências, entender o modelo de chamada ao GLPK e pegar as coordenadas do prédios previamente ditos.

Para entender o funcionamento do GLPK foi pesquisada através do site da GNU, ao qual foi mencionada no referencial teórico desse artigo, onde foi entendido o seu uso e implementado através do OPTANO.modeling

As coodenadas geográficas e a distâncias tiradas dentre os prédios da PUC-MG foram obtidas através do Google Maps, pois a PUC não liberou as plantas com as distâncias entre esses prédios para o grupo, assim então foi modelado o grafo com os vértices sendo os prédios e as arestas sendo as distâncias entre eles.

O motor da solução foi implementado utilizando o GLPK [Makhorin], através do OPTANO.Modeling, uma API para o .NET Framework [Thai and Lam 2002] que provê

uma interface de modelagem para vários softwares de otimização, como o CPLEX, Gurobi e GLPK.

O back-end consiste em uma Web API desenvolvida em .NET Core que recebe como parâmetros os vértices inicial e final, realiza uma chamada ao motor da solução e retorna uma lista de arestas que configuram a solução ótima do menor caminho entre os dois vértices.

O front-end consiste em uma aplicação web desenvolvida em Ionic 4 e Angular 7 que provê uma interface de utilização da solução e que faz chamadas ao back-end informando o ponto de partida e destino como vértices inicial e final, recebe a lista de arestas e a exibe na tela. A todo momento você pode conferir a interface da figura 1 para lembrar onde fica cada aresta e cada vértice.

6.1. O método

Na teoria dos grafos, o problema do menor caminho é o problema de encontrar um caminho entre dois vértices em um grafo de forma que a soma dos pesos de suas arestas constituintes seja minimizado [Hernandes et al. 2009].

Foi elaborado no problema citado um grafo em que os vértices são as interseções (ou esquinas) entre dois ou mais caminhos (ou ruas) que conectam os prédios da computação.



Figura 2. O Problema

Esse problema poderia ser facilmente resolvido utilizando o algoritmo de Dijkstra para o menor caminho, pois com o grafo da Figura 2, podemos ver que se o usuário entrar com os vértices iniciais e finais, o algoritmo de Dijkstra resolveria facilmente o problema. Porém o grupo decidiu utilizar o modelo Simplex, informando o modelo matemático ao GLPK, onde o mesmo, verificando a função objetiva, as variáveis e as restrições, consegue identificar também a menor distância entre esses vértices.

A modelagem do problema foi feita com o modelo Simplex, informando as variáveis, sendo elas maiores ou iguais a 0, foi feito também a função objetiva, como apresentado na figura 1, cada vértice tem multiplicado o valor de sua aresta e isso foi feito para cada vértice do grafo, e também foi apresentado as restrições do modelo, onde

o mesmo informa qual será o caminho que o problema deseja encontrar, -1 para saída do vértice e 1 para entrada.

7. Desenvolvimento

As teorias utilizadas para o entendimento do problema são as teorias de grafos, onde para encontrar o menor caminho é feito através de algoritmos simples como Dijkstra ou outros parecidos, ou também o Modelo Simplex com as chamadas ao GLPK, o grupo optou pelo modelo Simplex nesse trabalho para maior aprendizado sobre o modelo.

O maior contribuidor para o entendimento do problema foi a teoria dos grafos, onde o menor caminho entre dois vértices de um grafo dirigido ou não dirigido, pois é o método mais simples e eficiente para resolução de problemas dessa maneira.

Outro grande contribuidor para resolver o problema foi o entendimento no modelo Simplex, onde foi informado ao GLPK as informações dos vértices, as entradas e saídas deles, sinalizando a quem o vértice está ligado, as restrições do problema, pois as distâncias entre os vértices tem que ser informada, e também na função objetiva do problema como na Figura 1.

Então foi decidido a implementação do problema com o GLPK como foi dito na seção de metodologia, onde foi gerado uma interface para o usuário decidir os vértices iniciais e finais para encontrar o menor caminho possível entre os prédios, como especificado na Figura 3.

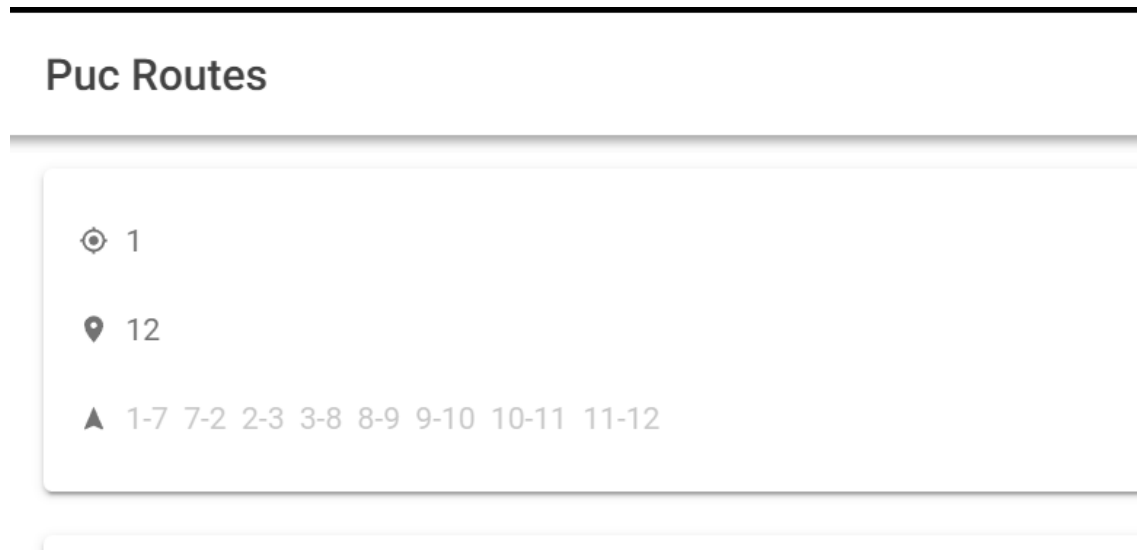


Figura 3. Resultado do Front-End

8. Coleta e avaliação de dados

A coleta dos dados do resultado se dá após a execução do modelo Simplex, onde o mesmo retornará os resultados de cada variável do problema, restando então apenas a avaliação desse resultado para identificar o menor caminho.

A avaliação da resposta gerada pelo Simplex se dá seguinte forma: O algoritmo retorna uma lista contendo o valor de cada variável do problema, 1 se a variável pertence

ao menor caminho e 0 se não pertence. Para encontrar o caminho basta exibir apenas as arestas cujas respectivas variáveis possuem valor 1.

9. Limitações do trabalho

As limitações encontradas nesse trabalho foram, primeiramente, a análise de sensibilidade do projeto, pois o trabalho consiste em variáveis binárias para locomoção entre os prédios da PUC, e em problemas de variáveis binárias não existe a análise de sensibilidade por não ser possível transformá-las em variáveis que podem ser alteradas os valores.

Outra limitação encontrada no projeto consiste em projetar a dualidade do problema, pois o mesmo aumenta muito o tempo de processamento do problema devido ao fato que as restrições se tornam variáveis, contendo 23 restrições, sendo que o tempo de processamento atual do problema já é bem alto.

10. Trabalhos futuros

Para trabalhos futuros o grupo sugere para determinar as rotas de todos os prédio da PUC-MG, tendo em vista que esse pode ser um projeto que ajudará não só os alunos da computação, mas sim a todos da faculdade, sendo assim, projetar e planejar todos os prédios de todos os cursos da PUC-MG na roteirização do problema, e ainda podendo incluir entre eles, as lanchonetes da dita faculdade.

Referências

- Bixby, R. (1992). Implementing the simplex method: The initial basis. *ORSA Journal on Computing*, 4(3):267–284.
- Fan, D. and Shi, P. (2010). Improvement of dijkstra's algorithm and its application in route planning. In *2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, volume 4, pages 1901–1904.
- Hernandes, F. A., Berton, L., and Castanho, M. J. A. d. P. (2009). O problema de caminho mínimo com incertezas e restrições de tempo. *Pesquisa Operacional*, 29:471 – 488.
- Makhorin, A. Glpk (gnu linear programming kit).
Available at <http://www.gnu.org/software/glpk/glpk.html>.
- Thai, T. L. and Lam, H. (2002). *.NET Framework Essentials (2Nd Edition)*. O' Reilly & Associates, Inc.