PROBLEMA DE CAMINHO MÍNIMO NO VALE DO IVAÍ - PR

Juliana Verga Shirabayashi (UFPR)
juliana.verga@ufpr.br
Nadya Zanin Muzulon (UFPR)
nadyamuzulon@gmail.com
Debora Fermino Domiciano (UFPR)
dfermino97.df@gmail.com
BRUNO JOSE RODRIGUES DE CAMOS (UFPR)
brunopicos97@gmail.com
Beatriz Betone de Lima (UFPR)
bia_lima09@hotmail.com

Neste trabalho, estudamos o problema de caminho mínimo na região do Vale do Ivaí - PR através de dois algoritmos clássicos: Dijkstra e Ford-Moore-Bellman. Esta região foi escolhida por ser onde esta localizado um dos Campus da Universidade Federal do Paraná - UFPR, Campus Avançado em Jandaia do Sul. Analisamos a região e utilizamos os algoritmos a fim de encontrar o menor caminho entre quaisquer duas cidades localizadas na mesma. Este trabalho foi motivado pelos estudos de diferentes métodos na disciplina de Tecnologia da Decisão II. Após obter os caminhos mínimos, analisamos os resultados e comparamos com as distâncias obtidas quando utilizamos o Google Maps. Por fim, este estudo permitiu uma análise de questões de logística de tal região

Palavras-chave: caminho mínimo, Vale do Ivaí, algoritmos clássicos

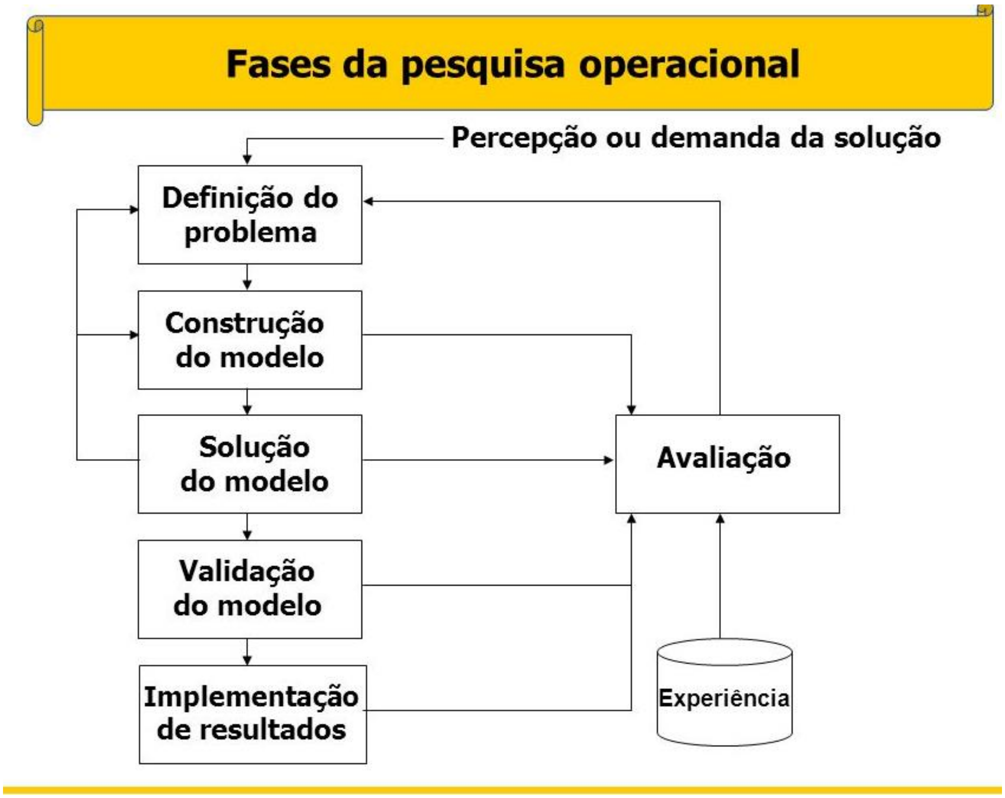
1. Introdução

O surgimento da Pesquisa Operacional (PO) foi em 1938, durante a Segunda Guerra Mundial, onde cientistas de diversas áreas como matemáticos, físicos e engenheiros se uniram para resolver problemas estratégicos, logísticos e táticos, a fim de avaliar e reposicionar adequadamente os radares do sistema de defesa aérea da Grã-Bretanha. Após a guerra os estudiosos envolvidos continuaram a realizar pesquisas e desenvolver modelos para auxiliar na tomada de decisão até mesmo para problemas não militares. Atualmente, este ramo da Matemática é bastante utilizado em diversas áreas. Provavelmente, o desenvolvimento metodológico mais importante do período pós-guerra foi o Método *Simplex*, por George Dantzig, em 1947, para a resolução de problemas de Programação Linear (ARENALES ET AL, 2007).

O surgimento dos computadores em 1950 possibilitou a solução de problemas práticos reais com diversas variáveis através dos algoritmos, e à medida que a capacidade computacional aumenta é possível solucionar problemas cada vez mais complexos.

A solução de um problema por diferentes métodos da PO envolve alguns passos a serem resolvidos, como segue na Figura 1 (ARENALES ET AL, 2007).

Figura 1- Fases da Pesquisa Operacional



Dentre as fases, a definição do problema é a que define os objetivos, as variáveis e as restrições, seguido pela construção do modelo matemático, onde geralmente apenas define-se qual o melhor modelo matemático a ser usado de acordo com o problema. A solução do modelo pode ser feita por diferentes métodos que podem ser métodos de Programação Linear (PL), Programação Não Linear (PNL), Programação Inteira (PI), entre outros.

1.1 Teoria de Grafos

Diferente de muitos dos ramos da Matemática que foram motivados por problemas envolvendo cálculos, o desenvolvimento da Teoria de Grafos se deu através de problemas envolvendo jogos e quebra-cabeças, o que do ponto de vista matemático parecia insignificante, mas apesar da aparente trivialidade, cada vez mais chamava a atenção de matemáticos pelos seus resultados teóricos de uma surpreendente variedade e profundidade.

O artigo de Leonhard Euler, publicado em 1736, sobre o problema das sete pontes de Königsberg, é considerado o primeiro resultado da teoria dos grafos, tendo o primeiro livro didático escrito por Dénes König e publicado em 1936.

A teoria dos grafos é um ramo da matemática que estuda as relações entre os objetos de um determinado conjunto. Para tal são empregadas estruturas chamadas de grafos, G(V,A), onde V é um conjunto não vazio de objetos denominados vértices (ou nós) e A é um subconjunto de pares não ordenados de V, chamados arestas (ou arcos).

Dependendo da aplicação, arestas podem ou não ter direção, pode ser permitido ou não arestas ligarem um vértice a ele próprio e vértices e/ou arestas podem ter um peso numérico associado.

Os grafos são geralmente representados por um círculo para cada vértice, e para cada aresta é desenhado um arco conectando suas extremidades. Se o grafo for direcionado, seu sentido é indicado na aresta por uma seta. Vejamos as figuras:

Figura 2- Grafo não-orientado

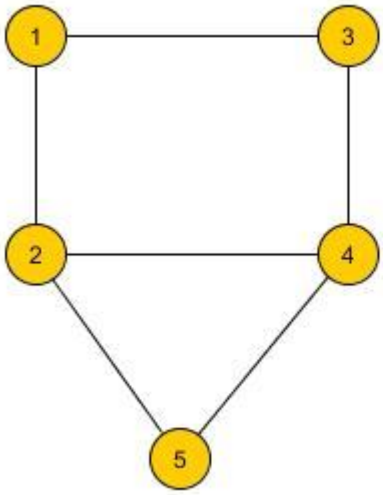
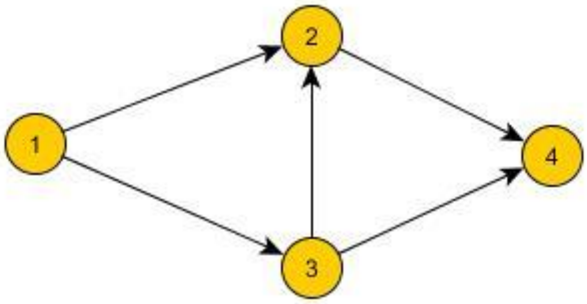


Figura 3- Grafo orientado



Há diversas maneiras de armazenarmos grafos em computadores, a estrutura de dados usada dependerá tanto da estrutura do grafo quanto do algoritmo usado para manipula-lo.

Dado um grafo G com n vértices, podemos representá-lo em uma matriz de $A=[a_{ij}]$ onde o valor a_{ij} guardará informações sobre como os vértices v_i e v_j estão relacionados.

Para representar um grafo não direcionado, simples e sem pesos nas arestas, criamos uma matriz A de adjacência, nesta, basta que as entradas a_{ij} da matriz A contenham 1 se, existe aresta do nó i ao nó j e 0 caso contrário. Se as arestas do grafo tiverem pesos, a_{ij} pode conter, ao invés de 1, o peso dessa mesma aresta.

Para representar um grafo direcionado, criamos uma matriz A de incidência, nesta, basta que as entradas a_{ij} da matriz A contenham 1 se, a aresta j sai do vértice i, -1 se a aresta j chega no vértice i e 0 se a aresta j não é incidente do vértice i.
Estruturas que podem ser representadas por grafos estão em toda parte e muitos problemas de interesse prático podem ser formulados por eles, como o problema de caminho mínimo que é estudado neste trabalho.

1.2 Problema de caminho mínimo

O problema do caminho mínimo (PCM) consiste na minimização do custo de travessia de um grafo entre dois vértices, custo este dado pela soma dos pesos de cada aresta percorrida. Geralmente existe mais de um caminho entre dois vértices específicos, então o problema de caminhos mínimos consiste em encontrar o caminho com menor custo dentre todos os possíveis.

Segue abaixo a modelagem matemática genérica para o PCM.

Modelagem para problema de caminho mínimo

$$\min \quad f(x) = \sum_{(i,j)\in A} c_{ij}x_{ij}$$

$$\text{s.a} \sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} 1, & i = o \\ 0, & i \neq o,\, d \\ -1, & i = d \end{cases}$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A.$$

Dada à modelagem, o problema de caminhos mínimos pode ser resolvido por diferentes algoritmos, entre eles podemos citar o algoritmo de Dijkstra (DIJKSTRA,1959) e Ford-Moore-Bellman (BELLMAN,1958).

1.2.1 Algoritmo de Dijkstra

O algoritmo de Dijkstra, concebido pelo cientista da computação holandês Edsger Dijkstra em 1956 e publicado em 1959 (DIJKSTRA, 1959), soluciona o problema do caminho mais curto num grafo dirigido ou não dirigido com arestas de peso não negativo.

Trata-se de um algoritmo iterativo, que parte de uma estimativa inicial para a distância mínima considerada infinita, e vai sucessivamente ajustando esta distância. Tal algoritmo utiliza-se de uma fórmula de recorrência e considera que um nó é "fechado" quando se encontra o caminho mínimo da origem até este nó e aqueles nós cujos caminhos mínimos ainda não foram encontrados são considerados ativos ou "abertos". O conceito de fechado e aberto está associado à impossibilidade de se encontrar um caminho melhor do que o já encontrado assim enquanto o nó não é fechado (ou rotulado) ainda é possível encontrar um caminho de menor valor da origem até este nó. A seguir estão definidos os passos generalizados do algoritmo:

Passo 1: Atribui-se um valor d(x) para cada um dos nós do grafo sendo: d(*o*) = 0 e d(x) = ∞; $\forall$ x $\neq$ *o*. Considere y o último nó rotulado (fechado). Inicialmente o nó *o* é o único rotulado e y = *o*;

Passo 2: Para cada nó x aberto redefine-se d(x) conforme a expressão $d(x)^{i} = \min \{d(x)^{i-1}, d(y) + d(y; x)\}$, onde: $d(x)^{i}$ é o tamanho do caminho da origem *o* até o nó x (na iteração corrente); d(y) é o tamanho do caminho da origem o até o nó fechado y; d(y; x) é o tamanho do arco (y; x). O nó "aberto" que possuir o menor valor d(x) é "fechado" e faz-se y = x;

Passo 3: Se o nó de destino d foi "fechado" então pare, um caminho de *o* para d foi encontrado. Se o nó o ainda não foi "fechado" volte ao Passo 2.

1.2.2 Algoritmo de Ford-Moore-Bellman

O segundo algoritmo citado, algoritmo de Ford-Moore-Bellman tem-se tal nomenclatura devido aos três autores, Lester Ford (1956), Edward Moore (1957) e Richard Bellman (1958) terem proposto o mesmo algoritmo. Este algoritmo encontra o caminho mais curto entre dois nós, mesmo que haja arcos com custos negativos, sendo uma generalização do algoritmo de Dijkstra.

Sua principal diferença é que ao invés de fechar um vértice por iteração, ele examina todos os vértices do grafo por iteração até que atualizações não sejam possíveis, determinando assim o menor caminho entre um vértice origem de todos os demais vértices do grafo. A seguir estão definidos os três passos pelo qual o algoritmo é formado:

Passo 1: Inicialmente o nó inicial é fechado (R), e os demais abertos (NR). Padronizam-se os valores de distância mínima para cada nó. Percorrem-se todos os vértices e defini-se que a sua distância mínima no momento é infinito, enquanto na origem coloca-se 0. O nó predecessor (p) do nó original também é considerado 0, enquanto os demais nós não possuem predecessor. Considera-se o valor de "a" para cada iteração o último nó incluído nos fechados (R);

Passo2: $\forall$ v $\in$ N, determina-se d(v) = min {d(v), d(a)+c(a; v)} e se faz p(v) = a , onde p(v) é o predecessor do vértice v, caso d(v) = d(a) + c(a; v). Se d(i) = ∞, $\forall$ v $\in$ NR, pare. Se para algum v $\in$ R, d(v) decresceu de valor, então se excluí o nó v de R e inclua em NR;

Passo3: Caso NR $\neq \phi$ determina-se k tal que d(k) = min {d(v), v $\in$ NR}, exclui-se k dos NR e, o inclui, nos R, assim "a" passa a valer k. Quando NR= ϕ recupera-se o caminho mínimo C a partir dos valores armazenados em p.

1.3 Vale do Ivaí

A partir do estudo da Teoria de Grafos e do problema genérico de caminho mínimo, este trabalho tem como objetivo encontrar o caminho de custo mínimo entre pares de cidades do Vale do Ivaí.

O Vale do Ivaí está localizado na região Sul do Brasil, no centro-oeste do Estado do Paraná. Abrange uma área de aproximadamente 7.385,05 km², correspondendo cerca de 3,7% do território Paranaense (IBGE), sendo composto por 28 cidades, entre elas Jandaia do Sul, onde está localizado a Universidade Federal do Paraná, Campus Avançado de Jandaia do Sul.

Figura 4- Mapa do vale do Ivaí

2. Metodologia de Solução

O primeiro passo foi a criação do grafo, onde as arestas estão direcionada em ambos os sentidos, considerando-se o mesmo caminho para ir e voltar dentre as cidades, onde cada cidade foi vinculada a um número (Tabela 1), neste caso, os nós (vértices) são as cidades e o os arcos (arestas) os caminhos com as distâncias entre as mesmas (Figura 4), posteriormente, foi criado a matriz de distâncias, e para resolução foram utilizados os dois algoritmos de caminho mínimo descritos anteriormente.

Figura 4- Grafo representando o Vale do Ívai

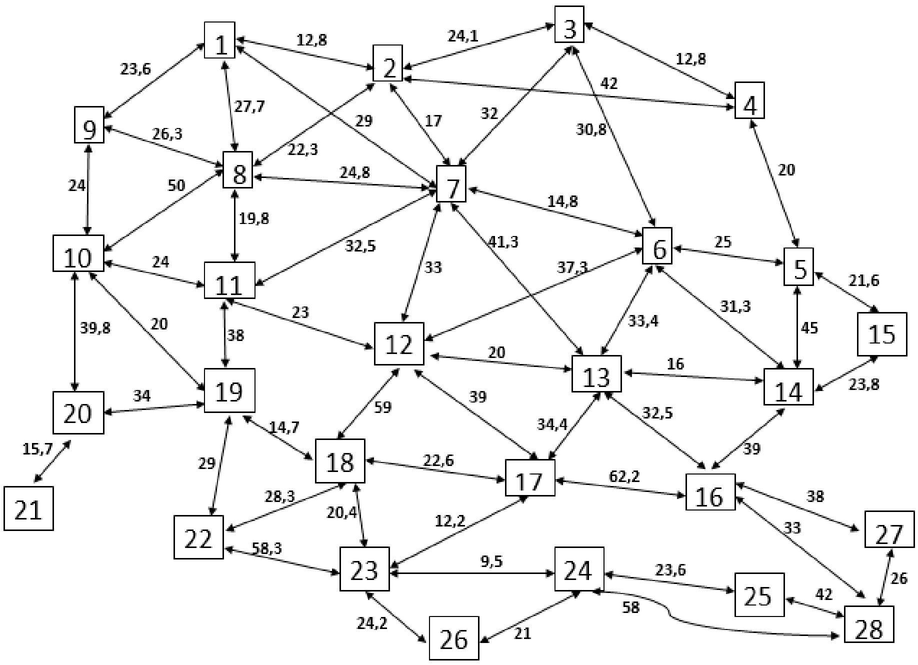


Tabela 1 – Relação entre cidades e números correspondentes

CIDADES	NÚMERO VINCULADO
Jandaia do Sul	1
Cambira	2
Apucarana	3
Califórnia	4
Marilândia do Sul	5
Rio Bom	6
Novo Itacolomi	7
Marumbi	8
Bom Sucesso	9
São Pedro do Ivaí	10
Kaloré	11
Borrazópolis	12
Cruzmaltina	13
Faxinal	14
Mauá da Serra	15

Grandes Rios	16
Lidianópolis	17
Lunardelli	18
São João do Ivaí	19
Barbosa Ferraz	20
Corumbataí do Sul	21
Godoy-Moreira	22
Jardim Alegre	23
Ivaiporã	24
Ariranha do Ivaí	25
Arapuã	26
Rosário do Ivaí	27
Rio Branco do Ivaí	28

A matriz de distâncias criada a partir do grafo é composta pelas distâncias em quilômetros (km) entre as cidades, que neste caso são os nós, tais distâncias foram coletadas por meio do aplicativo Google *Maps*.

Para resolução do problema, foram escolhidos três pares de cidades do Vale do Ivaí, sendo as origens Jandaia do Sul, Apucarana e Ivaiporã, tendo como destino Mauá da Serra. Tais pares de cidades foram escolhidos, pelo fato de existir um grande fluxo de pessoas, em determinadas épocas do ano, com destino ao litoral Paranaense, passando assim pela Cidade Mauá da Serra.

3. Testes computacionais

Os testes computacionais foram realizados através de um notebook Acer, core i3 (2,4GHz), utilizando o software livre (*SciLab*) na plataforma Windows, e como citado anteriormente, por meio dos dois algoritmos obtivemos os seguintes resultados:

Para o primeiro par de cidades, Jandaia do Sul – Mauá da Serra, o caminho mínimo obtido é composto por: Jandaia do Sul → Novo Itacolomi → Rio Bom → Marilândia do Sul → Mauá da Serra, totalizando 90,4 Km.

Para o segundo par de cidades, Apucarana – Mauá da Serra, o caminho mínimo obtido é composto por: Apucarana → Califórnia → Marilândia do Sul → Mauá da Serra, totalizando 54,4 Km.

E por fim, entre o terceiro par de cidades Ivaiporã – Mauá da Serra, o caminho mínimo obtido é composto por: Ivaiporã → Jardim Alegre → Lidianópolis → Cruzmaltina → Faxinal → Mauá da Serra, totalizando 95,9 Km.

Comparando os resultados obtidos com os do Google *Maps*, podemos observar, na tabela a seguir, que os resultados obtidos são bons, haja vista que as distâncias entre as cidades variam de acordo com a localização que consideramos (por exemplo: distância de centro - centro difere da distância entrada – entrada que difere da distância saída – saída)

Tabela 2 – Comparação dos resultados obtidos entre o algoritmo e o *Google Maps*

PARES DE CIDADES	ALGORITMOS	GOOGLE *MAPS*
Jandaia do Sul – Mauá da Serra	90,4 KM	71,4 KM
Apucarana – Mauá da Serra	54,4 KM	51,9 KM
Ivaiporã – Mauá da Serra	95,9 KM	90,6 KM

4. Considerações Finais

Neste trabalho, aplicamos dois algoritmos clássicos de caminhos mínimos na região do Vale do Ivaí, a fim de determinar o menor caminho entre dois pares de cidades quaisquer desta região. Tal região foi escolhida, por ser onde esta localizado um dos Campos da UFPR.

Como era esperado, ambos os algoritmos encontram o menor caminho para qualquer par de cidades e nos testes computacionais especificamos 3 pares de cidades.

Após a obtenção do menor caminho entre as cidades especificadas, analisamos questões relacionadas à logística das rodovias: é possível perceber o porquê de algumas rotas estarem em ótimas condições e outras não. Concluímos que as rotas mais atrativas e com maior fluxo recebem mais investimentos enquanto que aquelas menos utilizadas ficam em segundos planos.

5. Referências

ARENALES, Marcos; ARMENTANO, Vinícius; MORABITO, Reinaldo; YANASSE, Horacio. **Pesquisa Operacional para cursos de Engenharia**. Editora Campus, 523 pp, 2007.

BELLMAN, Richard. On a routing problem. **Quarterly Applied Mathematics**, 16,87-90, 1958.

DIJKSTRA, Edsger. A note on two problems in connexion with graphs. **Numerische Mathematik**. **1**: 269–271, 1959.

CHARTRAND, Gari; LESNIAK, Linda; ZHANG, Ping. ***Graphs & Digraphs***. Editora CRC Press, 640pp, 2004.

SILVA, Braulio Wilker. **Pesquisa Operacional: uma visão geral**. Disponível em <http://www.bwsconsultoria.com/2011/04/pesquisa-operacional-visao-geral.html>. Acesso em 04 maio, 2017.