

GRAFOS: algoritmos para calcular os menores caminhos

Gabriela Fernanda Moreira
Graduanda em Matemática - Uni-FACEF
moreirafernandagabriela@gmail.com

Mauricio Conrado Alves
Graduando em Matemática - Uni-FACEF
mconrado193@gmail.com

Dra. Lucinda Maria de Fátima Rodrigues Coelho
Docente do Departamento de Matemática - Uni-FACEF
lucindarcoelho@gmail.com

Resumo

Grafos são representações subjetivas formadas por dois conjuntos: vértices e arestas, comumente usados para representar situações tais como rodovias entre uma cidade e outra; rede de distribuição de energia elétrica interligando casas. Nesse trabalho apresenta-se uma breve introdução à teoria dos grafos, além de relacionar grafos a matrizes de adjacência e de incidência, com o objetivo principal de apresentar a alunos do Ensino Médio os algoritmos de Dijkstra e de Floyd-Warshall para encontrar o menor caminho. Para resolução do problema de caminho mínimo existem vários algoritmos, dentre eles vamos destacar os algoritmos de Dijkstra e de Floyd-Warshall. O algoritmo de Dijkstra é uma solução para o problema do caminho mínimo de origem única, funciona em grafos orientados e não orientados, no entanto, todas as arestas devem conter apenas custos positivos. O algoritmo de Floyd funciona em grafos ponderados seja ele orientado ou não orientado, e trabalha com pesos (ou valores) positivos e negativos. Compreender que a interpretação de um problema pode ser transcrita através de uma modelagem antes de encontrar sua solução, nesse caso a utilização de um desenho (grafo) para modelar problemas enriquece o conhecimento do aluno e oferece ferramentas adequadas para tomada de decisões na busca de uma solução, a qual pode ser obtida manualmente ou com o auxílio de um computador com novas oportunidades de aprendizagem até mesmo em outras disciplinas e assuntos que não sejam a Matemática.

Palavras-chaves: Grafos; Algoritmo de Dijkstra; Algoritmo de Floyd-Warshall; Menor caminho.

1 Introdução

Uma grande complexidade de estruturas do mundo real pode ser representada subjetivamente por meio de grafos. Os grafos são representações abstratas poderosas e flexíveis, possuem características específicas em diferentes áreas.

Segundo Feofiloff (2013), grafos são um bom modelo para muitos problemas de matemática, de computação e de engenharia. Afirma ainda, que a teoria dos grafos não é propriamente uma teoria, mas uma coleção de problemas os quais apresentam interessantes desafios intelectuais e têm importantes aplicações práticas.

Grafos surgiram em 1736, com um problema da época, conhecido como o Problema das sete pontes de Königsberg, criado pelos próprios habitantes da cidade. Grafos estão interligados com a matemática e com várias áreas, tais como: sociologia, biologia, computação, jogos, cinematográfica etc., apesar de grafo estar presente em diferentes áreas, ele possui importâncias específicas dentro de cada uma.

Grafos são estruturas discretas que podem ser utilizadas na representação e resolução de várias classes de problemas algorítmicos. Por exemplo, existem os “problemas de decisão” cujo objetivo consiste em decidir a resposta **sim** ou **não** a uma questão; os de “localização” em que o objetivo é localizar certa estrutura e que satisfaça um conjunto de propriedades dadas e os de “otimização” que envolve critérios de maximização e minimização. Os algoritmos associados a esses problemas permitem verificar a existência de caminhos para ir de um objeto a outro seguindo determinadas conexões.

Um grafo pode ser uma espécie de mapa rodoviário, os vértices são as cidades e as arestas são estradas de mão dupla que ligam as cidades; ou uma rede de distribuição de energia elétrica interligando em casas. Na web, os vértices são representados pelas páginas, e as arestas pelos hiperlinks de uma página para a outra. Nas redes sociais obtêm-se grafos por meio de ciclos de amizades distintas, por exemplo, o A é amigo de B, D, e G, mas o B, não é amigo de D e G, através do A o B começa a ter uma relação mesmo que indireta com D e G, pois ambos possuem uma amizade em comum com A. Outros exemplos poderiam ser expostos, mas o essencial é ter a percepção que os grafos estão presentes em inúmeros contextos.

O objetivo desse trabalho é abordar os algoritmos de Dijkstra e Floyd-Warshall para resolução de problemas que procuram os menores caminhos cujas soluções podem ser modeladas com grafos. Tais algoritmos nos dão um caminho menor por meio de matrizes de adjacência e de incidência e devido a isso, serão abordadas também

O Algoritmo de Dijkstra (E.W. Dijkstra) é um dos algoritmos que calcula o caminho de custo mínimo entre vértices de um grafo. A partir do vértice escolhido como origem da busca, este algoritmo calcula o custo mínimo deste vértice para todos os demais vértices do grafo. É simples e com um bom nível de desempenho, mas não garante, a exatidão da solução caso haja a presença de arcos com valores negativos. Este algoritmo parte de uma estimativa inicial para o custo mínimo e ajusta sucessivamente este cálculo. Ele avalia que um vértice estará fechado quando obtiver um caminho de custo mínimo do vértice tomado como origem da busca até ele, caso contrário será dito estar aberto.

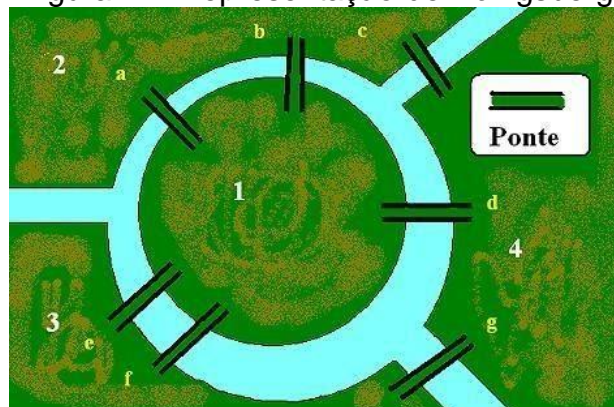
O algoritmo de Floyd-Warshall calcula os caminhos mais curtos entre todos os pares de vértices de um grafo direcionado e ponderado que eventualmente possua arcos com peso negativo, mas que não possua ciclos de custo negativo escolhido um vértice como raiz da busca, este **algoritmo** calcula o custo mínimo deste vértice para todos os demais vértices do grafo. O **algoritmo** pode ser usado sobre grafos orientados (dígrafos), ou não.

2 Grafos

2.1 Desenvolvimento Histórico

Grafos (BOAVENTURA NETTO/JURKIEWICZ, 2009, pag.1,2.) surgem em 1736, por intermédio do problema das Sete pontes de Königsberg, cidade que foi erguida entre o rio Pregel, onde existem duas ilhas e para ajudar a passagem das pessoas e do transporte de cargas, foram construídas sete pontes em diferentes locais como mostra a Figura 1.

Figura 1 – Representação de Königsberg



Fonte: Wikipedia, 2021, online.

Os habitantes queriam saber se era possível fazer um percurso partindo de uma origem e retornando a ela mesma, passando por todas as pontes uma única vez. O matemático suíço Leonhard Euler (1707-1783), foi o primeiro que conseguiu desenvolver uma solução para o tal problema e provar que não era possível realizar o percurso dentro das prescrições que os habitantes sugeriram, além de criar um diagrama para representar a cidade e generalizar o problema (GONSALVES, 2007, p 14).

2.2 Definições de Grafo

Grafos é uma representação subjetiva formada por dois conjuntos: vértices, conhecidos também como nóculo ou nós e arestas também chamadas de arcos.

Uma grande complexidade de estruturas do mundo real pode ser representada subjetivamente por detrás deles. Qualquer aresta é um par não ordenado de vértices, isto é, um conjunto com exatamente dois vértices. Imaginem que um grafo seja uma espécie de mapa rodoviário, os vértices são as cidades e as arestas são estradas de mão dupla que ligam as cidades.

Um grafo é por um par (V, A) , onde V é o conjunto de vértices e A o conjunto das arestas que interligam os pares de vértices.

Quando apenas é admitida no máximo uma aresta entre dois nós, diz-se que o grafo é **simples**. Os **vértices** ligados por uma aresta chamam-se **adjacentes**, como mostra a Figura 2. A **aresta** que emenda dois vértices denomina-se **incidente** a cada um dos vértices, como na Figura 3.

Figura 2

Os vértices X e Y são adjacentes



Fonte: Elaborado pelos autores.

Figura 3

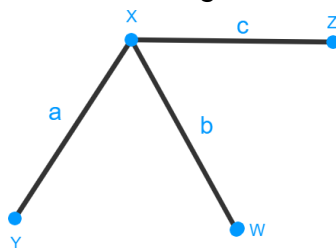
A aresta 'a' é incidente em X e Y



Fonte: Elaborado pelos autores.

O **número de arestas** incidentes no vértice expressa o **grau** desse vértice. O grau máximo do grafo é o maior dos vértices ($X(G)$), por consequência o grau mínimo ($Y(G)$) é o menor. Na Figura 4 o vértice X tem três arestas incidentes, logo o grau é 3. Os vértices Z, W e Y tem uma aresta incidente, sendo assim, ambos são grau 1. Dessa forma, o grau máximo do grafo é 3 e o mínimo é 1.

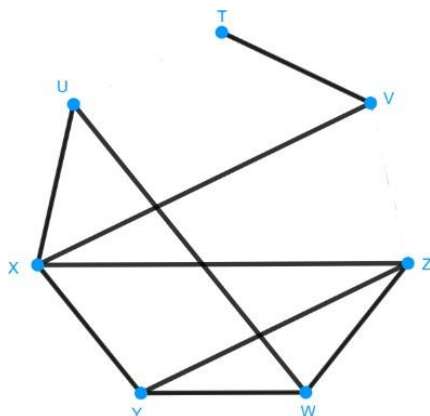
Figura 4 - Grau máximo do grafo é 3 e o mínimo é 1



Fonte: Elaborado pelos autores.

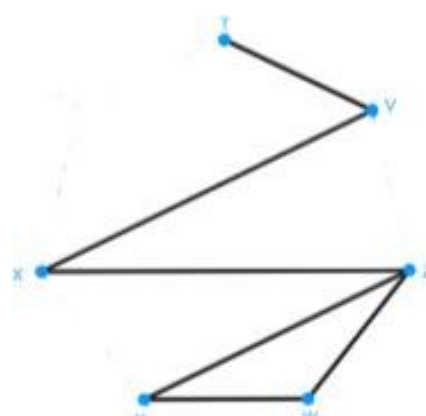
Um **segmento de vértices** em que os vértices sucessivos estão ligados pelas arestas do grafo refere-se a um **caminho**.

Figura 5



Fonte: Elaborado pelos autores.

Figura 6



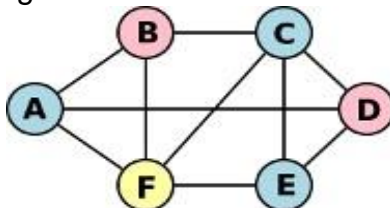
Fonte: Elaborado pelos autores.

Os vértices T-U-V-W-X-Y-Z ilustrados na Figura 5 formam um caminho encontrado no grafo G da Figura 6.

Outras concepções de grafos são de que o conjunto dos vértices pode ser infinito, e nesse caso é chamado de grafo infinito. No momento em que existir mais de uma aresta entre o mesmo par de vértices, tem-se as arestas múltiplas e quando há uma aresta que conecta no mesmo nó, chama-se **laço**.

Os nós podem ser representados de inúmeras formas. A Figura 7 ilustra um grafo colorido não dirigido.

Figura 7 - Grafo colorido não dirigido



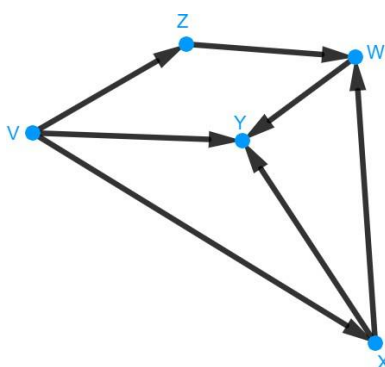
Fonte: Wikipedia, 2021, online.

2.2.1 Dígrafos ou grafos dirigidos

Além do grafo, existe o conceito de dígrafo. A distinção entre os conceitos consiste no fato de que as arestas do grafo são simplesmente formadas por dois vértices, enquanto que no dígrafo, as **arestas** possuem um **vértice inicial** e um **vértice final** (arestas dirigidas).

Quando as arestas de um grafo tiverem **orientação**, chamamos de **grafo dirigido** ou **dígrafo** e quando não tiver direção chamamos de grafo não dirigido ou simplesmente grafos. Grafos orientados, ou dígrafos são utilizados quando suas arestas são direcionadas, como é o caso do grafo da Figura 8. Pode-se perceber que as arestas que ligam os vértices têm indicações de onde saem e para onde vão, simbolizando que não é possível fazer o caminho contrário.

Figura 8 - Grafo dirigido ou dígrafo



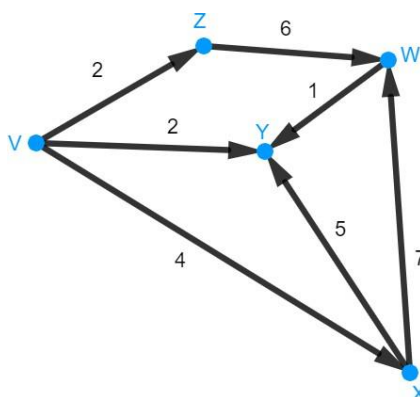
Fonte: Elaborado pelos autores.

Em grafos orientados, o grau de entrada de um vértice é o número de arestas que **entram** no respectivo vértice V e é definido por $d^-(v)$, e o de saída é o número de arestas que **saem** do vértice em direção a outros,

definido por $d^+(v)$. O vértice é chamado de **fonte** quando o grau de entrada é igual a zero e **sumidouro** quando o grau de saída for a igual a zero.

Os **grafos ponderados**, são aqueles que além da orientação recebem valores ou pesos para suas arestas, como ilustrado na figura 9.

Figura 9 - Grafo ponderado



Fonte: Elaborado pelos autores.

2.3 Representação de grafos por matrizes ou tabelas

Nesse modelo de representação criam-se vetores para cada vértice, em que os elementos desses vetores demonstram as ligações entre o vértice representado pelo vetor e cada vértice que possui o grafo.

Para representação de grafos, as formas mais comuns são: matriz de adjacências e de incidência.

Dado um grafo $(G) = (V, A)$ pode se obter desse grafo uma matriz, utilizando-se um grafo não orientado, obtém-se uma matriz de adjacência.

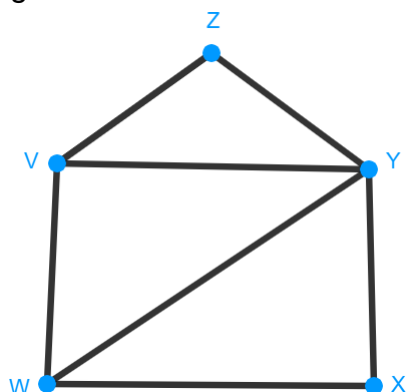
2.3.1 Matriz de adjacência de grafos não orientados

Uma matriz M de ordem $N \times N$, para um grafo não orientado é tal que:

- N = ao número de vértices;
- $M(i, j) = 1$ se existir arestas de i a j ;
- $M(i, j) = 0$ se não existir arestas de i a j .

j.Exemplos:

Figura 10 - Grafo não orientado



Fonte: Elaborado pelos autores.

Tabela 1 – Para a obtenção da Matriz de adjacência do grafo **não** orientado da figura 10.

	V	W	X	Y	Z
V	0	1	0	1	1
W	1	0	1	1	0
X	0	1	0	1	0
Y	1	1	1	0	1
Z	1	0	0	1	0

Fonte: Elaborado pelos autores.

Ou na forma matricial $M = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$

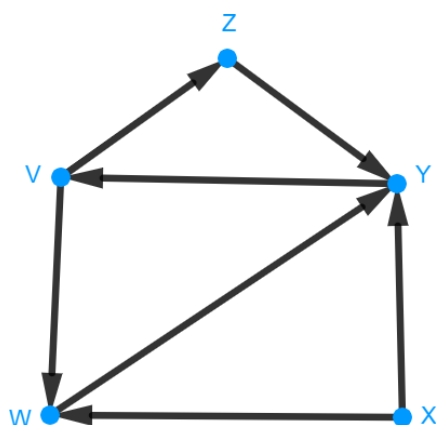
2.3.2 Matriz de adjacência e de incidência de dígrafos (grafos orientados)

Utilizando-se um grafo orientado, pode-se obter uma matriz de adjacência e uma matriz de incidência.

2.3.2.1 Matriz de adjacência M de ordem N x N, para um dígrafo (grafo orientado)

- N = ao número de vértices;
- $M(i, j) = 1$ se existir arestas de i a j;
- $M(i, j) = 0$ se não existir arestas de i a j.

Figura 11 - Dígrafo (grafo orientado)



Fonte: Elaborado pelos autores.

Tabela 2 – Para a obtenção da Matriz de adjacência do **dígrafo (grafos orientado)** da Figura 11.

	V	W	X	Y	Z
V	0	1	0	0	1
W	0	0	0	1	0
X	0	1	0	1	0
Y	1	0	0	0	0
Z	0	0	0	1	0

Fonte: Elaborado pelos autores.

Ou na forma matricial $M = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

2.3.2.2 Matriz de incidência de dígrafos (grafos orientados)

Com uma pequena modificação é possível utilizar a estrutura anterior, para identificar quando uma aresta que incide sobre um vértice converge ou diverge para este vértice.

A matriz de incidência do grafo orientado ou dígrafo, M de ordem N x N, em que N é o número de vértices, é definida como:

- $M(i, j) = +1$ se existir arestas de j até i, isto é, se a aresta converge de j para i;
- $M(i, j) = -1$ se existir arestas de i até j, isto é, se a aresta diverge de j para i;
- $M(i, j) = 0$ se não existir arestas de i a j.

Tabela 3 — Para a obtenção da Matriz de incidência do dígrafo (grafo orientado) da Figura 11.

	V	W	X	Y	Z
V	0	+1	0	-1	+1
W	-1	0	-1	+1	0
X	0	+1	0	+1	0
Y	+1	-1	-1	0	-1
Z	-1	0	0	+1	0

Fonte: Elaborado pelos autores.

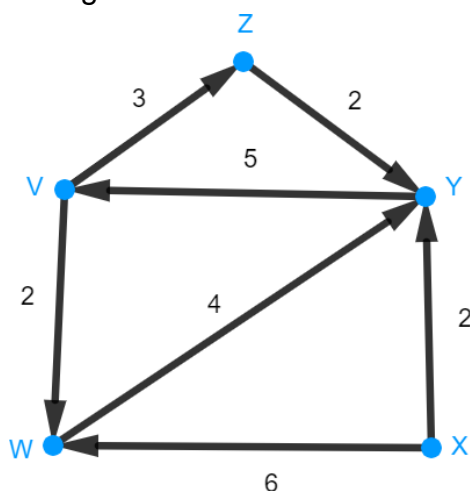
Ou na forma matricial $M =$

$$\begin{bmatrix} 0 & +1 & 0 & 0 & +1 \\ -1 & 0 & +1 & +1 & 0 \\ 0 & +1 & 0 & +1 & 0 \\ +1 & +1 & -1 & 0 & -1 \\ -1 & 0 & 0 & +1 & 0 \end{bmatrix}$$

2.3.3 Matriz de dígrafos (grafos orientados) valorados

Grafo valorado é todo grafo que suas arestas apresentam valores. Isto é, em alguns casos os caminhos que ligam os vértices têm valores diferentes uns dos outros, então se deve atribuir valores às arestas do grafo. Nesse caso dizemos que o grafo é um grafo valorado, veja a Figura 12.

Figura 12- Grafo Valorado



Fonte: Elaborado pelos autores.

Tabela 4 — Para a obtenção da Matriz de incidência do grafo valorado referente à figura 12.

	V	W	X	Y	Z
V	0	2	0	0	3
W	0	0	0	4	0
X	0	6	0	2	0
Y	5	0	0	0	2
Z	0	0	0	2	0

Fonte: Autoria própria.

Ou na forma matricial $M = \begin{bmatrix} 0 & 2 & 0 & 0 & 3 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 6 & 0 & 2 & 0 \\ 5 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 \end{bmatrix}$

As definições apresentadas são os conceitos básicos de grafos necessários para elucidar os problemas nos quais serão modelados nesse trabalho aplicando estes mecanismos.

3 Problema do Caminho Mínimo

O problema do caminho mínimo consiste em encontrar o menor caminho entre dois nós ou vértices, de forma em que a soma dos valores de suas arestas existentes possa ser minimizada. Assim, resolver este problema pode significar determinar o caminho entre dois vértices com o custo mínimo, ou com o menor tempo de viagem ou a menor distância. Por exemplo, um caminhoneiro deseja passar por um caminho mais curto entre dois vértices quaisquer, onde o vértice de origem seja A e o destino seja G, precisa-se analisar o caminho mais curto para passar por todo o percurso de modo que chegue mais rápido no destino, em uma rede qualquer dependendo de seus atributos, pode existir n caminhos entre um par de vértices determinados como origem e destino.

Entre os vários caminhos aquele que possui o menor “valor” é chamado de caminho mínimo. Este valor representa a soma total dos valores das arestas que compõem o caminho, e estes valores podem ser representados pelo tempo de viagem, a distância percorrida ou um custo qualquer da aresta.

3.1 Modelagem matemática

Numa rede qualquer, dependendo das suas características, pode existir vários caminhos entre um par de nós, definidos como origem e destino. Entre os vários caminhos aquele que possui o menor “peso” é chamado de caminho mínimo. Este peso representa a soma total dos valores dos arcos que compõem o caminho e estes valores podem ser: o tempo de viagem, a distância percorrida ou um custo qualquer do arco.

O modelo matemático para o problema de caminho mais curto do vértice 1 ao vértice n de um grafo $G = (V, A)$, $N = \{1, 2, \dots, n\}$.

- Variáveis:
 $x_{ij} \in \{0, 1\}$ = ativação, ou não do arco (i, j)
- Parâmetros:
 c_{ij} = custo unitário do fluxo em (i, j)
 $S(j)$ = é o conjunto dos vértices sucessores de j
 $P(j)$ = é o conjunto dos vértices predecessores de j
- Função objetivo:
$$\min Z = \sum_{i=1}^n \sum_{j \in S(i)} c_{ij} x_{ij}$$

- Restrições = {
$$\begin{aligned} \sum_{j \in S(1)} x_{ij} &= 1 \\ \sum_{i \in P(n)} x_{in} &= 1 \\ \sum_{i \in P(j)} x_{ij} &= \sum_{k \in S(j)} x_{jk} \quad ; j = 2; \dots ; n - 1 \end{aligned}$$

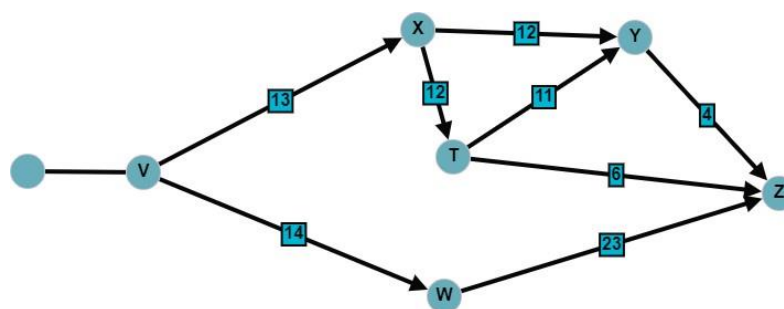
Para resolução do problema de caminho mínimo existem vários algoritmos, dentre eles vamos destacar os algoritmos de Dijkstra e de Floyd-Warshall. O algoritmo de Dijkstra é uma solução para o problema do caminho mínimo de origem única, funciona em **grafos orientados e não orientados**, no entanto, **todas as arestas** devem conter apenas **custos positivos**. O algoritmo de Floyd funciona em **grafos ponderados** seja ele **orientado** ou **não orientado**, e trabalha com **pesos** (ou valores) **positivos e negativos**.

3.2 Algoritmo de Dijkstra

O cientista da computação Edsger W. Dijkstra desenvolveu em 1956 o algoritmo original para encontrar o caminho de custo mínimo entre dois nós ou vértices dados em um grafo ponderado positivamente, mas uma variante mais comum é fixar um único nó como o vértice de origem e encontrar o caminho mais curto dele para qualquer outro vértice em um grafo, a principal ideia do algoritmo de Dijkstra é eliminar continuamente os caminhos mais longos entre o vértice de origem e todos os outros vértices do grafo.

A maior vantagem desse algoritmo é que as distâncias mais curtas de 2 vértices podem ser calculadas em $O(v)$ em vez de $O(3)$, onde v é o número de vértices do grafo.

Figura 13: Exemplo de um Dígrafo valorado



Fonte: autoria própria

O algoritmo de Dijkstra começa com o vértice selecionado como o vértice de origem e examina todo o grafo para determinar o menor caminho da origem para todos os outros vértices do grafo. Como os vértices são processados sequencialmente, encontrar os caminhos mais curtos do vértice que encontramos para outro vértice é feito como uma operação de relaxamento e a distância entre os vértices será maior ou igual à menor distância calculada. Se começarmos com um vértice, ele terá uma distância de 0 e o vértice restante terá distâncias indefinidas ou infinitas até a fonte.

O algoritmo usa os pesos das arestas para encontrar um caminho que minimiza a distância total entre os pesos de um determinado vértice (o vértice de origem) e todos os outros vértices e é só aplicável quando todos os pesos das arestas são positivos, porque durante a execução os pesos são somados para encontrar o caminho mais curto.

Estágios:

- 1º- Marcar todos os vértices como não visitados;
- 2º- marcar o vértice inicial com uma distância atual de **0** e os outros vértices com o símbolo ∞ (infinito);
- 3º- escolher um nó de avaliação do conjunto de nós não visitados, o nó de avaliação deve ser aquele com a menor distância da fonte;
- 4º- calcular novas distâncias para os vizinhos diretos, mantendo a distância mais baixa em cada avaliação;
- 5º- adicionar vizinhos que ainda não foram estabelecidos ao conjunto de nós não visitados.

Essas etapas podem ser divididas em dois processos, Inicialização e Avaliação, a seguir um exemplo de aplicação dessas etapas □.

Exemplo da aplicação para cálculo do menor caminho com o algoritmo de Dijkstra

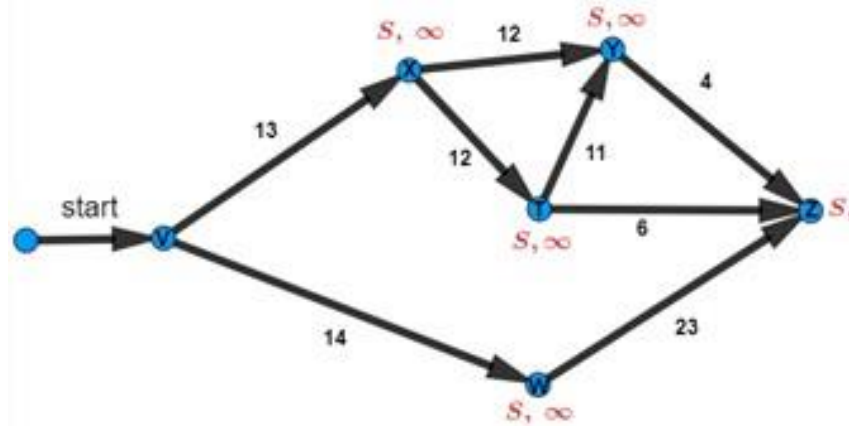
Seja o dígrafo da figura 13

- Inicialização

Marcar todos os nós com uma distância infinita e um predecessor desconhecido, exceto a fonte. Em seguida atribui-se valor ao nó V (entendemos que a distância do nó V ao nó V é zero), deste modo, cada nó no

resto do grafo será distinguido com um predecessor e uma distância, como mostra a figura 14.

Figura 14- Exemplo da aplicação do algoritmo de Dijkstra: 1º passo



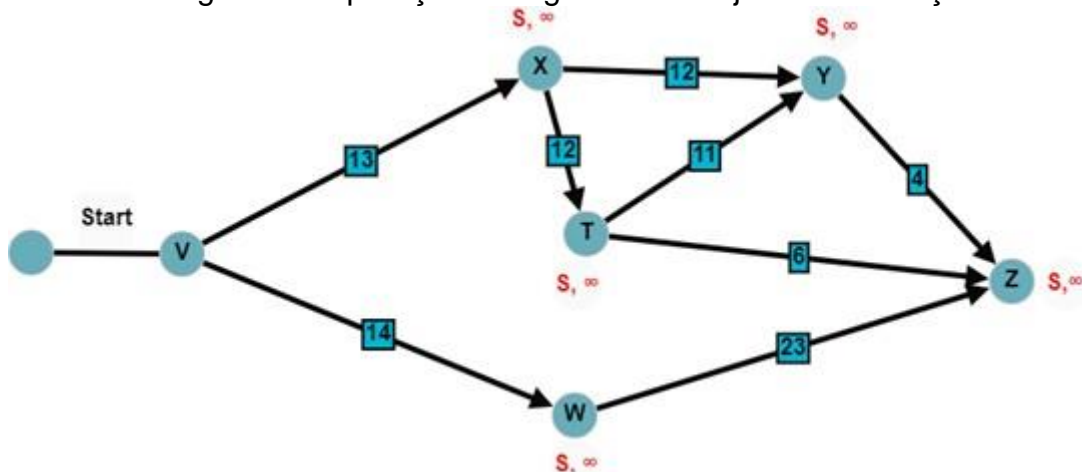
Fonte: Elaborado pelos autores.

Para finalizar o processo da Inicialização, precisamos adicionar o nó V aos nós não definidos dispostos a serem visitados primeiro na etapa da avaliação.

Avaliação

Nesta etapa o nosso grafo já foi inicializado, escolhermos vértices com a menor distância do conjunto não selecionado e, em seguida, avaliaremos todos os vértices adjacentes que não foram selecionados. No caso a seguir estamos calculando a menor distância do vértice V ao vértice Z, onde o vértice V é nosso vértice de início pois possui a menor distância, como mostrar a figura 15.

Figura 15- Aplicação do algoritmo de Dijkstra: 2ª iteração

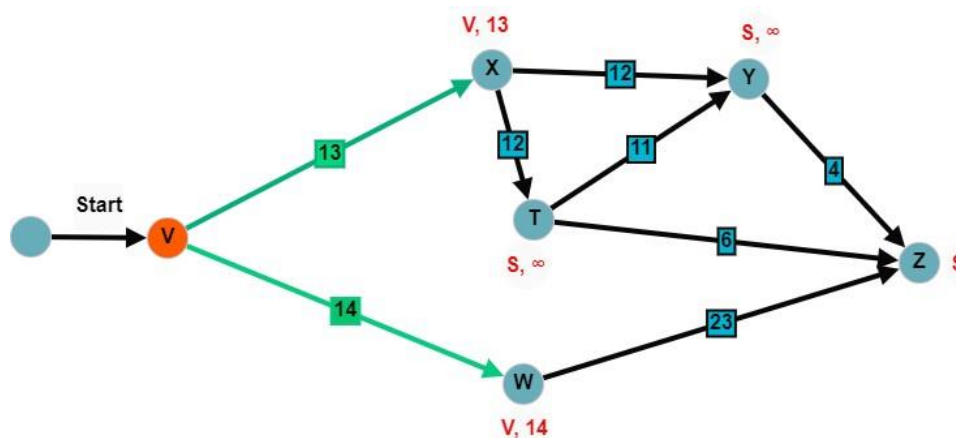


Fonte: Elaborado pelos autores.

O conceito é adicionar o peso da aresta à distância do vértice de avaliação e, em seguida, compará-lo com a distância do destino. por exemplo, para o vértice X, $0 + 13$ é menor que infinito, então a nova distância para o vértice X é 13, e o novo predecessor é V, o mesmo se aplica ao nó W, pois $0 +$

14 é menor que infinito, então a nova distância de W é 14 e seu predecessor é V, como na figura 16.

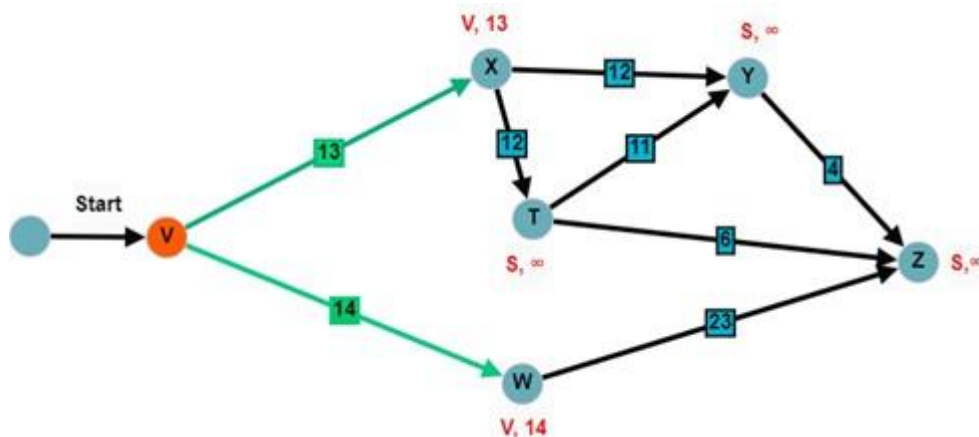
Figura 16- Aplicação do algoritmo de Dijkstra: 2ª iteração



Fonte: Elaborado pelos autores.

Ao ser feita a primeira iteração notamos que a menor distância dos vértices analisando é de $V \rightarrow X$, então X passará a ser o nosso predecessor e a partir do mesmo vamos analisar outros vértices do conjunto não visitados, já que a partir do X temos a possibilidade de dois caminhos $X \rightarrow T$ e $X \rightarrow Y$. A segunda iteração é calcular o menor caminho dentro os caminhos dispostos, que são $X \rightarrow Y$ e $X \rightarrow T$, ao calcular é notório que a distância de ambos são a mesma resultando em 25 a distância, como mostra a figura 18.

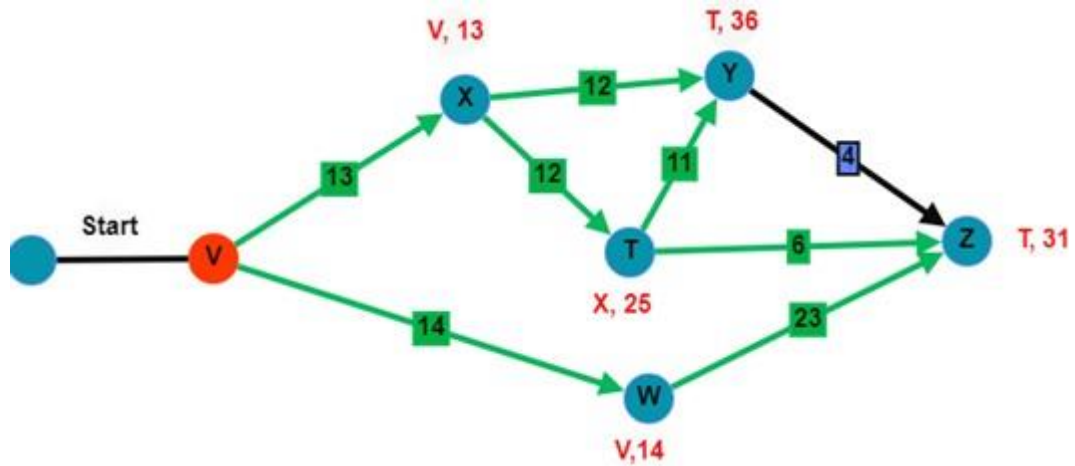
Figura 17- Aplicação do algoritmo de Dijkstra: 2ª iteração



Fonte: Elaborado pelos autores.

Na terceira iteração será analisado vértice W e o caminho disposto a partir do mesmo, se sua distância for menor do que 25, logo o menor caminho de $V \rightarrow Z$ foi encontrado, se a distância for maior que 25 é necessário retornar ao vértice com a menor distância encontrada anteriormente e calcular uma nova rota. Ao se calcular a distância do vértice W, $14 + 23$ resultando é 37, $37 > 25$ sendo assim sua distância é maior do que a distância calculada anteriormente, portanto é necessário retornar a vértice com a menor distância e calcular uma nova rota, como mostra a figura 18.

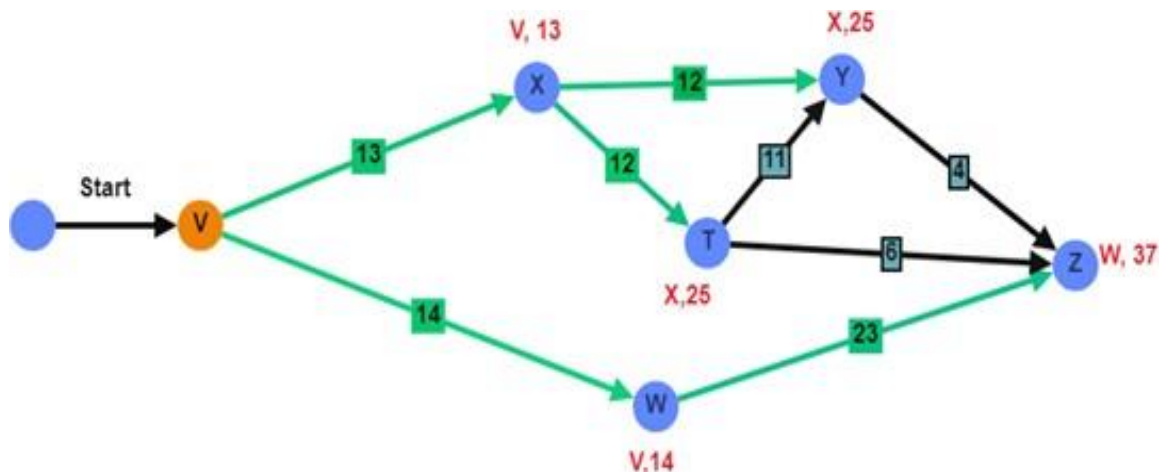
Figura 18- Aplicação do algoritmo de Dijkstra: 3ª iteração



Fonte: Elaborado pelos autores

Na quarta iteração pode ser analisado novas rotas partindo tanto de Y como de T, pois suas distâncias são equivalentes. Sendo T nosso predecessor temos a possibilidade de dois caminhos, um que vai de T diretamente a nosso ponto de destino(Z) e outro que tem como parada o vértice Y. Ao calcular $T \rightarrow Z$ encontramos uma distância maior que 25, pois para o vértice Z, $25 + 6$, resultando em 31, $31 > 25$, portanto é necessário ainda buscar por uma nova rota. Ao se calcular $T \rightarrow Y$ também é encontrado uma distância maior do que 25, pois para Y, $25 + 11$, resultando em 36, $36 > 25$, sendo assim só nos restas calcular a distância de $Y \rightarrow Z$ é avaliar se a iteração pois o menor caminho do grafo.

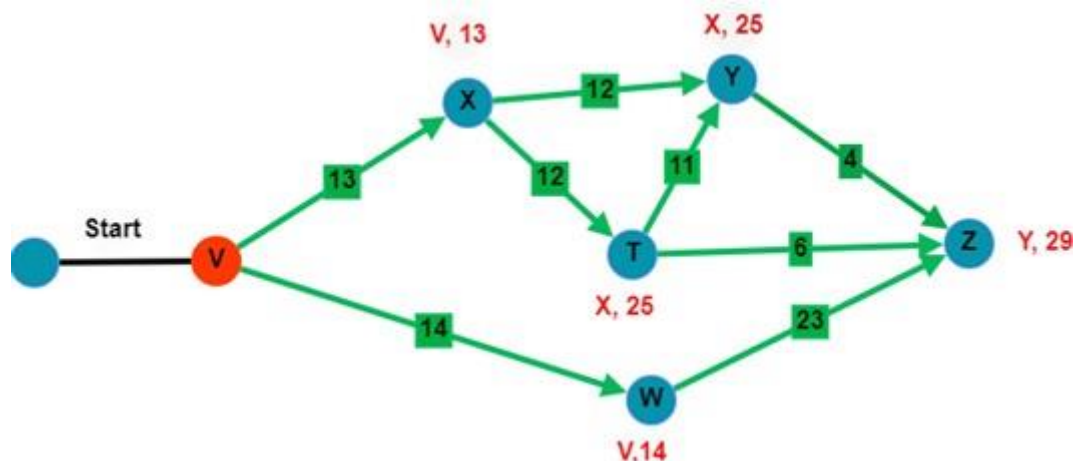
Figura 19- Aplicação do algoritmo de Dijkstra: 4ª iteração



Fontes: Elaborado pelos autores.

Na quinta iteração verificamos por fim o único vértice ainda não visitado, usaremos o menor caminho encontrado até Y, que no caso é 25 tendo X como se predecessor. Com isto podemos calcular de $Y \rightarrow Z$, pois para o vértice Z, $25 + 4$ resultando em 29, sendo assim o menor caminho encontrado do ponto V até Z, como mostra a figura 20.

Figura 20- Aplicação do algoritmo de Dijkstra: 5ª iteração



Fonte: Elaborado pelos autores.

Em seguida constrói-se uma tabela que mostra todas as iterações que foram realizadas no processo de avaliação

Tabela 5 – Tabela de Iterações

Iteração	Não Resolvidos	Assentou	Nó de avaliação	V	W	X	Y	T	Z
1	V	-	V	0	V-14	V-13	V-∞	V-∞	V-∞
2	X, W	V	X	0	V-14	V-13	X-25	X-25	V-∞
3	W, Y, T	V, X	W	0	V-14	V-13	X-25	X-25	T-37
4	T, Z, Y	V, X, W	T	0	V-14	V-13	T-36	X-25	T-31
5	Y, Z	V, X, W, T	Y	0	V-14	V-13	T-36	X-25	Y-29
6	Z	V, X, W, T, Y	Z	0	V-14	V-13	T-36	X-25	Y-29
FINAL	-	TUDO	NENHUM	0	V-14	V-14	T-36	X-25	Y-29

Fonte: Elaborado pelos autores.

A notação X-25, por exemplo significado que no vértice X é o predecessor imediato com uma distância total de 25 do vértice V. Por fim pode-se calcular os caminhos mais curtos do vértice A são os seguintes:

- Vértice X: $V \rightarrow X$ a distância total é 13;
- Vértice W: $V \rightarrow W$ a distância total é 14;
- Vértice T: $V \rightarrow X \rightarrow T$ a distância total é 25;
- Vértice Y: $V \rightarrow X \rightarrow T \rightarrow Y$ a distância total é 25;
- Vértice Z: $V \rightarrow X \rightarrow T \rightarrow Y \rightarrow Z$ a distância total é de 29;

3.4 Algoritmo de Floyd- Warshall

O algoritmo de Floyd-Warshall é usado para encontrar o par de caminhos mais curto de um vértice em um grafo até uma borda do grafo, tendo ele valores positivos ou negativos, supondo que o caminho $T(Q_1, Q_2, Q_j)$ seja o caminho minimamente ponderado do vértice Q ao vértice J . Os pontos intermediários são extraídos de um subconjunto de vértices $(1, 2, \dots, k)$ e o algoritmo funciona examinando os pares de caminhos, usando cada vértice como um ponto intermediário.

O intuito deste algoritmo é averiguar a cada iteração se a inclusão de um vértice k intermediário no percurso de i para j pode diminuir o tamanho de um percurso já determinado.

O algoritmo de Floyd-Warshall é responsável por calcular o menor caminho por meio de todos os pares de vértices em um grafo valorado e orientado. Floyd-Warshall foi propagado pelo informático Robert Floyd em 1966, este algoritmo é o mesmo que foi publicado por Bernard Roy em 1959, e por Stephen Warshall em 1962 a fim de determinar o desfecho transitivo de um grafo. O formato atual do algoritmo com três loops de repetição foi descrito por Peter Ingerman em 1962.

A aplicação do algoritmo ocorre somente no centro de um grafo, e na mediana.

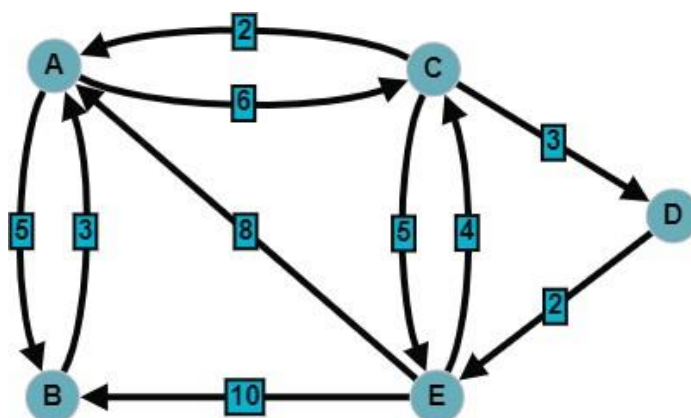
Estruturas do Algoritmo

- Numere os vértices do grafo;
- Defina a matriz inicial (D^0) cujos valores d^{0ij} sejam correspondentes aos valores das arestas i,j se existir arestas no grafo, caso contrário considere $d_{ij} = \infty$, pois pode existir infinitas maneiras formas de solucionar, e faça os elementos da diagonal da matriz $d_{ii} = 0$ para todo i ;
- Para cada $k = n$, determine sucessivamente os elementos da matriz D_k , a partir dos elementos da matriz D_{k-1} , utilizando a expressão 1.

$$D_{ij}^k = \min \{d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1}\} \quad (1)$$

Exemplo da aplicação para cálculo do menor caminho com o algoritmo de Floyd-Warshall

Figura 21- Exemplo de Dígrafo valorado para **cálculo do menor caminho com o algoritmo de Floyd-Warshall**.



Fonte: Elaborado pelos autores.

1ª Etapa: Construa a matriz de **DISTÂNCIA**, com as respectivas distâncias.

$$M^0 = \begin{pmatrix} 0 & 5 & 6 & \infty & \infty \\ 3 & 0 & \infty & \infty & \infty \\ 2 & \infty & 0 & 3 & \infty \\ \infty & \infty & \infty & 0 & 2 \\ h & 8 & 10 & 4 & \infty & 0 \end{pmatrix}$$

2ª Etapa: Construa a matriz M^1

- Para construção de M^1 , repetir somente os valores da **linha A**, da **coluna A** e a **diagonal da matriz M^0** .
- Com isto na matriz M^1 , verifica-se que faltará valores para completar a matriz, para encontrar tal valores terá que ser feita a seguinte verificação:

$$D_{ij} > D_{ik} + D_{kj}$$

□ $\left\{ \begin{array}{l} \text{se for verdade a afirmação, então } D_{ij} = D_{ik} + D_{kj} \\ \text{se Não, mantém - se o valor} \end{array} \right.$

- K, sempre será a linha e a coluna que estamos analisando, por exemplo na matriz M^1 nosso K= A (linha A ; coluna A), na matriz M^2 o K será = B (linha B ; coluna B) e assim sucessivamente.

$$M^1 = \begin{pmatrix} 0 & 5 & 6 & \infty & \infty \\ 3 & 0 & & & \\ 2 & & 0 & & \\ \infty & & & 0 & 2 \\ h & 8 & & & 0 \end{pmatrix}$$

Resolução:

- para se calcular a célula BC (linha B ; coluna C)

$$D_{bc} = \infty \text{ e } K=A$$

$$D_{ik} = D_{ba} = 3 \text{ e } D_{kj} = D_{ac} = 6$$

$$D_{bc} > D_{ba} + D_{ac} \\ \infty > 3 + 6$$

→ $\left\{ \begin{array}{l} \infty > 9 \text{ sim, então o novo valor de } D_{bc} \text{ será } 9 \text{ em } M^1 \end{array} \right.$

- para se calcular a célula BD (linha B ; coluna D)

$$D_{bd} = \infty \text{ e } K=A$$

$$D_{ik} = D_{ba} = 3 \text{ e } D_{kj} = D_{ad} = \infty$$

$$D_{bd} > D_{ba} + D_{ad}$$

→ $\left\{ \right.$

$$\infty > 3 + \infty$$

$\infty > 3 + \infty$? Não , então o valor de D_{bd} se mantém, isto é, ∞ em M^1

- para se calcular a célula BE (linha B ; coluna E)

$$D_{be} = \infty \text{ e } K=A$$

$$D_{ik} = D_{ba} = 3 \text{ e } D_{kj} = D_{ae} = \infty$$

$$D_{be} > D_{ba} + D_{ae}$$

$$\rightarrow \left\{ \begin{array}{l} \infty > 3 + \infty \end{array} \right.$$

$\infty > 3 + \infty$? Não , então o valor de D_{be} se mantém, isto é, ∞ em M^1

□ para se calcular a célula CB (linha C ; coluna B)

$$D_{cb} = \infty \text{ e } K=A$$

$$D_{ik} = D_{ca} = 3 \text{ e } D_{kj} = D_{ab} = \infty$$

$$D_{cb} > D_{ca} + D_{ab}$$

$$\rightarrow \left\{ \begin{array}{l} \infty > 2 + 5 \end{array} \right.$$

$\infty > 7$? Sim , então o valor de D_{cb} será 7, em M^1

□ para se calcular a célula CD (linha C ; coluna D)

$$D_{cd} = 3 \text{ e } K=A$$

$$D_{ik} = D_{ca} = 2 \text{ e } D_{kj} = D_{ad} = \infty$$

$$D_{cd} > D_{ca} + D_{ad}$$

$$\rightarrow \left\{ \begin{array}{l} 3 > 2 + \infty \end{array} \right.$$

$3 > 2 + \infty$? Não , então o valor de D_{cd} se mantém, isto é, 3 em M^1

□ para se calcular a célula CE (linha C ; coluna E)

$$D_{ce} = \infty \text{ e } K=A$$

$$D_{ik} = D_{ca} = 2 \text{ e } D_{kj} = D_{ae} = \infty$$

$$D_{ce} > D_{ca} + D_{ae}$$

$$\rightarrow \left\{ \begin{array}{l} \infty > 2 + \infty \end{array} \right.$$

$\infty > 2 + \infty$? Não , então o valor de D_{ce} se mantém, isto é, ∞ em M^1

□ para se calcular a célula DB (linha D ; coluna B)

$$D_{db} = \infty \text{ e } K=A$$

$$D_{ik} = D_{da} = \infty \text{ e } D_{kj} = D_{ab} = 5$$

$$D_{db} > D_{da} + D_{ab}$$

$$\rightarrow \left\{ \begin{array}{l} \infty > \infty + 5 \end{array} \right.$$

$\infty > \infty + 5$? Não , então o valor de D_{db} se mantém, isto é, ∞ em M^1

□ para se calcular a célula DC (linha D ; coluna C)

$$D_{dc} = \infty \text{ e } K=A$$

$$D_{ik} = D_{da} = \infty \text{ e } D_{kj} = D_{ac} = 6$$

$$D_{dc} > D_{da} + D_{ac}$$

$$\rightarrow \left\{ \begin{array}{l} \infty > \infty + 6 \end{array} \right.$$

$\infty > \infty + 6$? Não , então o valor de D_{dc} se mantém, isto é, ∞ em M^1

- para se calcular a célula DE (linha D ; coluna E)

$$D_{de} = 2 \text{ e } K=A$$

$$D_{ik} = D_{da} = \infty \text{ e } D_{kj} = D_{ae} = \infty$$

$$D_{de} > D_{da} + D_{ae}$$

$$\rightarrow \left\{ \begin{array}{l} 2 > \infty + \infty ? \text{ Não , então o valor de } D_{de} \text{ se mantém, isto é, } 2 \text{ em } M^1 \end{array} \right.$$

- para se calcular a célula EB (linha E ; coluna B)

$$D_{eb} = 10 \text{ e } K=A$$

$$D_{ik} = D_{ea} = 8 \text{ e } D_{kj} = D_{ab} = 5$$

$$D_{eb} > D_{ea} + D_{ab}$$

$$\rightarrow \left\{ \begin{array}{l} 10 > 8 + 5 \\ 10 > 13 ? \text{ Não , então o valor de } D_{eb} \text{ se mantém, isto é, } 10 \text{ em } M^1 \end{array} \right.$$

- para se calcular a célula EC (linha E ; coluna C)

$$D_{ec} = \infty \text{ e } K=A$$

$$D_{ik} = D_{ea} = 8 \text{ e } D_{kj} = D_{ac} = 6$$

$$D_{ec} > D_{ea} + D_{ac}$$

$$\infty > 8 + 6$$

$$\rightarrow \left\{ \begin{array}{l} \infty > 14 ? \text{ Sim , então o valor de } D_{ec} \text{ será } 14 \text{ em } M^1 \end{array} \right.$$

- para se calcular a célula ED (linha E ; coluna D)

$$D_{ed} = \infty \text{ e } K=A$$

$$D_{ik} = D_{ea} = 8 \text{ e } D_{kj} = D_{ad} = \infty$$

$$D_{ed} > D_{ea} + D_{ad}$$

$$\rightarrow \left\{ \begin{array}{l} \infty > 8 + \infty \\ \infty > 8 + \infty ? \text{ Não , então o valor de } D_{ed} \text{ se mantém, isto é, } \infty \text{ em } M^1 \end{array} \right.$$

$$M^1 = \begin{pmatrix} 0 & 5 & 6 & \infty & \infty \\ 3 & 0 & 9 & \infty & \infty \\ 2 & 7 & 0 & 3 & 5 \\ \infty & \infty & \infty & 0 & 2 \\ h 8 & 10 & 4 & \infty & 0 \end{pmatrix}$$

3ª Etapa: Construir a matriz M^2

- Para construção de M^2 , repetir somente os valores da linha B, da coluna B e a diagonal da matriz M^0 .

$$M^2 = \begin{matrix} & 0 & 5 & & & \\ \begin{matrix} | \\ | \\ | \\ | \end{matrix} & 3 & 0 & 9 & \infty & \infty \\ & & 7 & 0 & & | \\ & & \infty & & 0 & 2 \\ h & & 10 & & & 0 \end{matrix}$$

- Verifica-se que na matriz M^2 faltarão valores para completar a matriz, para encontrar tal valores deverão ser feita a verificação:

$$D_{ij} > D_{ik} + D_{kj}$$

□ $\left\{ \begin{array}{l} \text{se for verdade a afirmação, então } D_{ij} = D_{ik} + D_{kj} \\ \text{se Não, mantem - se o valor} \end{array} \right.$

- K, sempre será a linha e a coluna que estamos analisando, na matriz M^2 o K será sempre = B (linha B ; coluna B) .
Realizando os cálculos de forma análoga para construir M^1 obtém-se □

$$M^2 = \begin{matrix} & 0 & 5 & 6 & \infty & \infty \\ & 3 & 0 & 9 & \infty & \infty \\ \begin{matrix} | \\ | \\ | \\ | \end{matrix} & 2 & 7 & 0 & 3 & 5 \\ & \infty & \infty & \infty & 0 & 2 \\ h & 8 & 10 & 4 & \infty & 0 \end{matrix}$$

4ª Etapa: Construir a matriz M^3

- Para construção de M^3 , repetir somente os valores da **linha C**, da **coluna C** e a **diagonal da matriz M^0** .

$$M^3 = \begin{matrix} & 0 & & 6 & & \\ \begin{matrix} | \\ | \\ | \\ | \end{matrix} & & 0 & 9 & & \\ & 2 & 7 & 0 & 3 & 5 \\ & & & \infty & 0 & \\ h & & & 4 & & 0 \end{matrix}$$

- Verifica-se que na matriz M^3 faltarão valores para completar a matriz, para encontrar tal valores deverão ser feita a verificação:

$$D_{ij} > D_{ik} + D_{kj}$$

□ $\left\{ \begin{array}{l} \text{se for verdade a afirmação, então } D_{ij} = D_{ik} + D_{kj} \\ \text{se Não, mantem - se o valor} \end{array} \right.$

- K, sempre será a linha e a coluna que estamos analisando, na matriz M^3 o K será sempre = C (linha C ; coluna C) .
Realizando os cálculos de forma análoga para construir M^1 obtém-se □

$$M^3 = \begin{matrix} & 0 & 5 & 6 & 9 & 11 \\ \mathbf{I} & 3 & 0 & 9 & 18 & 14 \\ \mathbf{I} & 2 & 7 & 0 & 3 & 5 \\ \infty & \infty & \infty & 0 & 2 & \mathbf{I} \\ \mathbf{h} & 6 & 10 & 4 & 7 & 0 \end{matrix})$$

5ª Etapa: Construir a matriz M^4

- Para construção de M^4 , repetir os valores da **linha D**, da **coluna D** e a **diagonal da matriz M^0** .

$$M^4 = \begin{matrix} & 0 & & 9 & & \\ \mathbf{I} & & 0 & & 18 & \\ \mathbf{I} & & & 0 & 3 & \mathbf{I} \\ \infty & \infty & \infty & & 0 & 2 \\ \mathbf{h} & & & 7 & 0 & \end{matrix})$$

- Verifica-se que na matriz M^2 faltarão valores para completar a matriz, para encontrar tal valores deverão ser feita a verificação:

$$D_{ij} > D_{ik} + D_{kj}$$

□ $\left\{ \begin{array}{l} \text{se for verdade a afirmação, então } D_{ij} = D_{ik} + D_{kj} \\ \text{se Não, mantem - se o valor} \end{array} \right.$

- K, sempre será a linha e a coluna que estamos analisando, na matriz M^2 o K será sempre = D (linha D ; coluna D) .

Realizando os cálculos de forma análoga para construir M^1 obtém-se □

$$M^4 = \begin{matrix} & 0 & 5 & 6 & 9 & 11 \\ \mathbf{I} & 3 & 0 & 9 & 18 & 14 \\ \mathbf{I} & 2 & 7 & 0 & 3 & 5 \\ \infty & \infty & \infty & 0 & 2 & \mathbf{I} \\ \mathbf{h} & 6 & 10 & 4 & 7 & 0 \end{matrix})$$

6ª Etapa: Construir a matriz M^5

- Para construção de M^5 , repetir somente os valores da **linha E**, da **coluna E** e a **diagonal da matriz M^0** .

$$M^5 = \begin{matrix} & 0 & & & 11 & \\ \mathbf{I} & & 0 & & 14 & \\ \mathbf{I} & & & 0 & 5 & \mathbf{I} \\ & & & & 0 & 2 \\ \mathbf{h} & 6 & 10 & 4 & 7 & 0 \end{matrix})$$

- Verifica-se que na matriz M^5 faltará valores para completar a matriz, para encontrar tal valores deverão ser feita a verificação:

$$D_{ij} > D_{ik} + D_{kj}$$

□ { *se for verdade a afirmação, então $D_{ij} = D_{ik} + D_{kj}$*
se Não, mantém - se o valor

- K, sempre será a linha e a coluna que estamos analisando, na matriz M^5 o K será sempre = E (linha E ; coluna E) .
 Realizando os cálculos de forma análoga para construir M^1 obtém-se □

$$M^5 = \begin{matrix} & & 0 & 5 & 6 & 9 & 11 \\ & \begin{matrix} | & 3 & 0 & 9 & 18 & 14 & | \end{matrix} \\ & \begin{matrix} | & 2 & 7 & 0 & 3 & 5 & | \end{matrix} \\ & \begin{matrix} | & 16 & 16 & 10 & 0 & 2 & | \end{matrix} \\ h & 6 & 10 & 4 & 7 & 0 &) \end{matrix}$$

Essa M^5 é a matriz que fornece o menor caminho entre todos os vértices pelo **Algoritmo de Floyd-Warshall**, que pode ser melhor representada pela tabela6, a seguir:

Tabela 6: Fornece o menor caminho entre todos os vértices pelo **Algoritmo de Floyd-Warshall**

	A	B	C	D	E
A	0	5	6	9	11
B	3	0	9	18	14
C	2	7	0	3	5
D	16	16	10	0	2
E	6	10	4	7	0

O Algoritmo de Floyd-Warshall é um algoritmo recursivo que utiliza a recursão que é um método de resolução de problemas que envolva desdobrar um problema.

Em sub-problemas menores e assim sucessivamente até chegar a um problema elementar, o suficiente para que este possa ser resolvido facilmente.

4 Considerações finais

Compreender que a interpretação de um problema pode ser transcrita por meio de uma modelagem antes de encontrar sua solução, nesse caso a utilização de um desenho (grafo) para modelar problemas amplia o conhecimento do aluno e oferece ferramentas adequadas para tomada de decisões na busca de uma solução, a qual pode ser obtida manualmente ou com o auxílio de um computador com novas oportunidades de aprendizagem até mesmo em outras disciplinas e assuntos que não sejam a Matemática.

As propostas de atividades levantam a possibilidade de teoria dos grafos ser inserida complementarmente em alguns conteúdos, como conjuntos ou matrizes.

Tendo em vistas as aplicações de grafos e a importância de seu uso em determinadas áreas, desde o tradicional problema das sete pontes de Königsberg, até mesmo com o encontro do mesmo em menores caminhos, acreditamos que a modelagem de um problema em grafos propicia aos alunos a pensar, a entender o abstrato e a inferir sobre várias proposições colocadas tornando-se uma técnica ou procedimento capaz de ajudá-lo a tomar decisões mais adequadas para resolver inúmeros problemas.

Referências

A teoria dos grafos e sua abordagem na sala de aula com recursos educacionais digitais / Flavia Fernanda Favaro. - Rio Claro, 2017. Dissertação (mestrado) - Universidade Estadual Paulista, Instituto de Geociências e Ciências Exatas.

Aplicações da Matemática: Redes Sociais, Jogos, Engenharia, Prof^o Fábio Protti (IC-UFF/RJ).

BOAVENTURA NETTO, Paulo Oswaldo. Teoria e Modelos de Grafos. São Paulo. Editora Edgard Blucher Ltda, 2003.

BOAVENTURA NETTO, Paulo Oswaldo / JURKIEWICZ, Samuel. Grafos: Introdução e prática. - São Paulo. Editora Edgard Blucher, 2009.

BRASIL. Parâmetros Curriculares Nacionais. Ciências da natureza e Matemática e suas tecnologias. MEC. 2000.

BRASIL. Secretária de Educação Básica. Base Nacional Curricular Comum: educação é a base. Brasília: MEC; SEB, 2017.

CAMINHOS e ciclos em grafos. [S. l.], 8 abr. 2017. Disponível em: https://www.ime.usp.br/~pf/algoritmos_para_grafos/aulas/paths-and-cycles.html. Acesso em: 17 maio 2021.

CONCEITO Básicos da Teoria de Grafos. Disponível em: <https://www.inf.ufsc.br/grafos/definicoes/definicao.html>. Acesso em: 14 set. 2021.

DA SILVA, Luiz Fernando. INTRODUÇÃO AO ESTUDO DA TEORIA DOS GRAFOS: UMA PROPOSTA DE SEQUÊNCIA DIDÁTICA PARA O ENSINO MÉDIO1. Curso de Especialização em M, Mídias Digitais, e didática para educação Básica matemática, p. 1-27.

EXERCÍCIOS de Teoria dos Grafos. [S. l.], 27 set. 2017. Disponível em: <https://www.ime.usp.br/~pf/grafos-exercicios/>. Acesso em: 6 out. 2021.

GRAFO 12 jul. 2012. Disponível em: <https://wikiciencias.casadasciencias.org/wiki/index.php/Grafo>. Acesso em: 4 ago. 2021.

GRAFOS (não-dirigidos). [S. l.], 23 nov. 2020. Disponível em: https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/ugraphs.html#exr:chess-knight. Acesso em: 9 jul. 2021.

Grafos e suas aplicações, Fabiana Nascimento Santos Cavalcante e Severino Domingos da Silva (PUC/RS).

MARINS, Fernando Augusto Silva. Introdução à Pesquisa Operacional. Cultura acadêmica, 2011.

Parâmetros curriculares nacionais: Ciências Naturais / Secretaria de Educação Fundamental. Brasília: MEC / SEF, 1998.