**Algoritmos de Melhor Rota**

**Lucas Rabelo Pereira¹, Luiz Fernando Fraga Valença²**

**¹**Centro Universitário Instituto de Educação Superior de Brasília (IESB)

SGAS Quadra 613/614, Via L2 Sul - Asa Sul, Brasília - DF, 70200-730

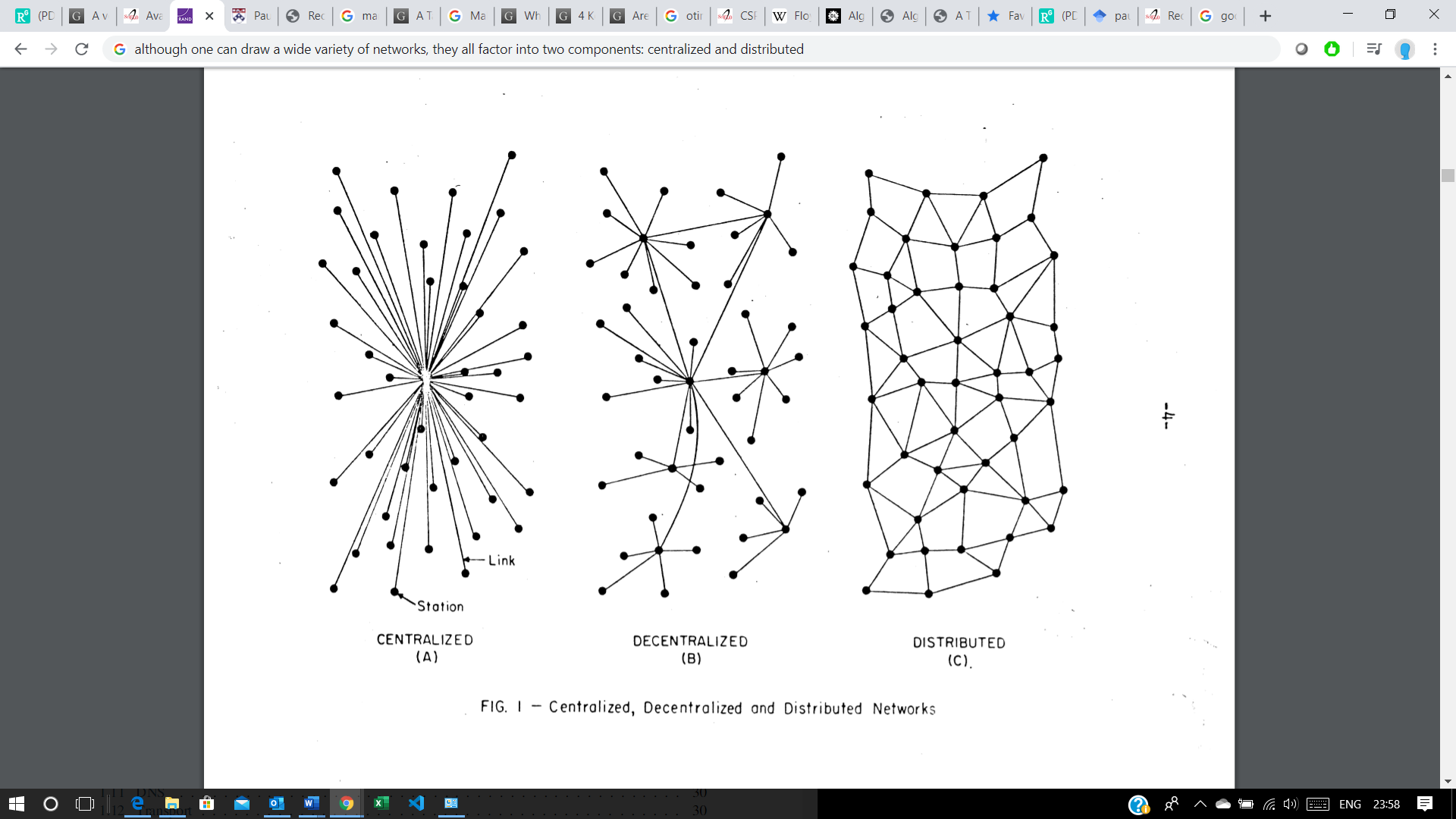
lucas\_rabelo.pereira@hotmail.com, lffvalenca@gmail.com

***Abstract.*** *Road trips are made every day and using a better route becomes necessary, especially when the work period is short. The calculation of the best route can be known using the Floyd-Warshall algorithm. An application uses the Floyd-Warshall algorithm to determine the fastest travel time from the starting point to the desired end point. The data from the desired locations will be collected using the Floyd-Warshall algorithm and then the system will calculate the fastest distance based on your start and end point. This application was developed with the objective of providing a better route for a trip, taking into account the cost-benefit that must return to the user.*

***Resumo.*** *Viagens rodoviárias são feitas todos os dias e a utilização de uma melhor rota se torna algo necessário, principalmente quando o período de trabalho é curto. O cálculo de melhor rota pode ser conhecido utilizando o algoritmo de Floyd-Warshall. A aplicação utiliza o algoritmo de Floyd-Warshall para determinar o tempo de viagem mais rápido do ponto inicial ao ponto final desejado. Os dados dos locais desejados serão coletados utilizando o algoritmo de Floyd-Warshall e, em seguida, o sistema calculará a distância mais rápida com base no seu ponto inicial e final. Esse aplicativo foi desenvolvido com o objetivo de fornecer a melhor rota para uma viagem levando em conta o custo-benefício que esse trajeto irá retornar ao seu usuário.*

1. Introdução

Em 1962, durante o período da guerra fria, Paul Baran planejou um sistema que mitigasse o risco de um ataque nuclear russo nas bases americanas. Tendo esse escopo, para desenvolver sua lógica em seu artigo “On distributed communications network”, classificou as redes em dois tipos: as centralizadas e as distribuídas, e ressaltou a importância da redundância das redes (Baran, 1962).



Em seu artigo, a redundância se refere à conectividade, porém, esse conceito pode ser abstraído para outras necessidades, os links de conectividade podem ser ruas numa cidade, atividades num projeto, pessoas numa empresa, entre outras opções.

“Hoje, o mundo dos negócios é muito mais propenso à rápidas mudanças do que era a vinte anos atrás” (Gallup,2011) e nesse cenário, apenas estarmos conectados não é mais satisfatório, desejamos que esses sistemas sejam inteligentes, eficientes, de baixo custo, curtas e que sejam uma solução para os problemas dos gestores. Para os stakeholders, esses desejos permitem o embasamento das decisões e podem proporcionar a economia de custos operacionais, economia de tempo, eficiência do processo.

Para que exista um problema de decisão é necessário: (1) Que um ou mais objetivos precisem ser alcançados; (2) A existência de várias alternativas que podem ser seguidas; (3) A existência de incerteza sobre qual alternativa irá otimizar os objetivos estabelecidos (Machado, 1976).

Logo, temos como problema um usuário que deseja sair de um ponto inicial e ir até um ponto final, passando ou não por outros pontos da rede. Diferentes opções de caminhos a serem seguidos e a incerteza sobre qual roteamento oferece o caminho mais curto para o usuário.

Visto isso, no contexto desse trabalho, iremos trabalhar as questões logísticas, buscando a menor distância entre dois ou mais nós da rede para propor a rota de menor caminho, também conhecido como caminho mínimo na literatura tradicional.

Como abordado no artigo do Instituto Gallup em 2011, “Nenhum processo isolado é mais crítico para o sucesso ou fracasso de uma organização do que a tomada de decisão.” Assim, devemos pensar em uma solução que envolva a rede.

O objetivo geral desse trabalho é desenvolver um aplicativo de otimização para diminuir o tempo de trajeto que o usuário utilizará para visitar espaços comerciais, será necessário calcular a rota a partir do seu ponto de chegada, até os locais desejados e posteriormente para um ponto final, como o seu hotel, por exemplo.

Este artigo tem como objetivo específico reduzir o tempo de deslocamento entre os pontos desejados, utilizar métodos que visem o custo-benefício afim de resolver a dificuldade de tempo em visitar as localizações planejadas com a melhor rota e que possua abundância de localizações finais.

O presente trabalho está organizado da seguinte forma: a Seção 2 apresenta o problema do caminho mínimo, a Seção 3 explana sobre o Grafos, Seção 4 explica o Algoritmo de Floyd-Warshall, a Seção 5 exibe os trabalhos relacionados, e, encerrando, a Seção 6 traz as considerações finais a respeito do trabalho.

1. Objetivo Geral

Analisar o desempenho do Algoritmo de Floyd-Warshall para o problema do caminho mínimo

1. Objetivos Específicos

Revisar as referências sobre o tema e desenvolver o algoritmo de Floyd-Warshall para solucionar o problema do caminho mínimo entre dois pontos.

1. Problema do Caminho Mínimo

O Problema do caminho mínimo tem como objetivo encontrar um caminho com menor distância, tempo ou custo a partir do nó fonte até o nó destino (Gao, 2011). Esse problema pode ser resolvido através de alguns algoritmos como os propostos ou desenvolvidos por Bellman, Dijkstra, Dreyfus, Floyd (Gao, 2011). Outra opção para a solução desse problema é o uso do método Simplex, no modelo de programação linear ou mesmo, resolver à mão avaliando as alternativas. (p,85. Taha, 2007).

1. Grafos

Basicamente, um grafo, G = (V,E) é um sistema formado por um conjunto V de elementos denominados como vértices, pontos ou nodos, e um conjunto E de pares não ordenados de vértices chamados de linhas ou arestas (Pereira, 2014).

A representação V(G) e E(G) será utilizada para V como o conjunto de vértices e E como o conjunto de arestas de um grafo, além de mais, serão utilizados como padrão os termos vértices e arestas para descrever os grafos.

Normalmente, os grafos são representados por figuras planas constituídas por pontos e arestas, como pode ser visto nas propriedades das relações abaixo (Valnei, 2019):

* Reflexiva: Qualquer vértice do grafo possui uma aresta com origem e destino nele mesmo



* Antirreflexiva ou irreflexiva: Qualquer vértice do grafo não possui aresta com origem e destino nele mesmo:



* Simétrica: Entre dois vértices quaisquer, ou não existe aresta ou existem duas arestas, uma em cada sentido



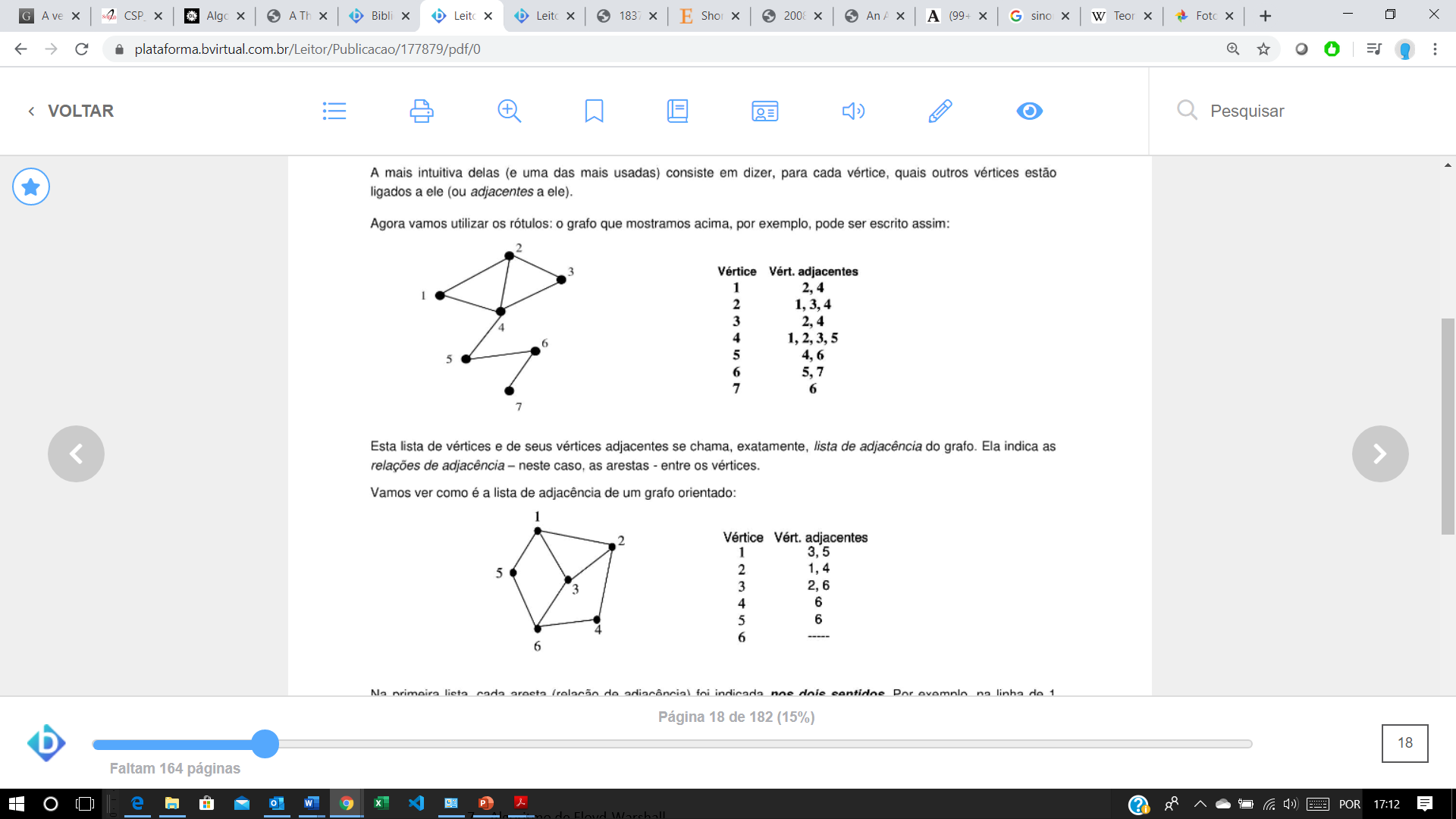
* Antissimétrica: Entre dois vértices quaisquer, existe no máximo uma aresta



* Transitiva: Para qualquer sequência de arestas saindo de um vértice a e chegando a outro vértice z existe uma aresta ligando esses vértices diretamente



Essas propriedades são representadas através de esquemas gráficos, porém, em sua resolução serão utilizados computadores que não entendem desenhos. Além do mais, o esquema de um grafo pode ser desenhado de infinitas maneiras, equivalentes, enquanto o grafo abstrato associado a todas elas serão o mesmo (XXX). Uma das maneiras mais intuitivas de organizar os dados de um grafo para serem introduzidos no computador consiste em dizer, para cada vértice, quais outros vértices estão ligados a ele, ou adjacentes a ele (p.18, Boa Ventura e Jurkiewicz, 2017).



Na lista de adjacência do grafo acima, indica as relações de adjacência entre os vértices do grafo, significando que o vértice 1 se relaciona com os vértices 2 e 4, o vértice 2 se relaciona com os vértices 1,3 e 4 e assim por diante para todos os vértices

1. Linguagem C

C foi uma linguagem criada em 1972 por Dennis Ritchie na empresa AT&T Bell Labs para o desenvolvimento de um sistema operacional e é uma linguagem de programação padronizada pela Organização Internacional para Padronização (ISSO)

Se tornou uma das linguagens de programação mais populares mundialmente e possui uma aplicabilidade muito alta pois existem poucas arquiteturas que não compilem C. A linguagem C tem inspirado outras linguagens de programação, principalmente o C++ que derivou de uma extensão para C.

C é uma linguagem para implementação de sistemas, seus pontos de design foram para ele ser compilado, fornecendo as qualidades de acesso irrestrito à memória e a baixa necessidade de um hardware avançado. Porém, a facilidade de compilação não descarta sua qualidade de ser desenvolvido em alto nível, a linguagem C foi necessária para várias aplicações que foram desenvolvidas originalmente em Assembly.

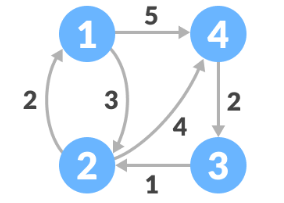
Essa qualidade atribuída não foi por acaso, a linguagem C foi elaborada com o foco principal em mente de facilitar o desenvolvimento de programas extensos com menos erros por meio do paradigma da programação procedural, mas sobrecarregando menos o autor do compilador, do qual o trabalho se complica ao ter de realizar as peculiaridades complexas da linguagem.

1. Algoritmo de Floyd-Warshall

Exibir de forma completa sua história, exemplos, figuras e soluções aplicadas no nosso trabalho através desse algoritmo.

O algoritmo de Floyd-Warshall é uma operação para encontrar o caminho mais curto em uma rede de pares de vértices em um gráfico ponderado. Essa operação funciona para ambos os gráficos, sejam eles ponderados direcionados ou não direcionados. Porém sua funcionalidade se limita a gráficos com ciclos positivos, os com ciclos negativos que, por sua vez, não são calculados.

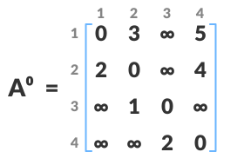
Um gráfico ponderado é um gráfico no qual cada aresta tem um valor numérico associado a ela. O algoritmo de Floyd-Warhshall também é chamado de algoritmo de Floyd, Roy-Floyd, Roy-Warshall ou WFI. Esse algoritmo segue a abordagem de programação dinâmica para encontrar os caminhos mais curtos.



Como exemplificado na imagem acima e seguindo as etapas abaixo, conseguimos encontrar o caminho mais curto entre todos os pares de vértices.

A. Crie uma matriz A1 da dimensão n \* n onde n é o número de vértices. A linha e a coluna são indexadas como i, respectivamente. j são os vértices do gráfico

Cada célula A [i] [j] é preenchida com a distância do i-ésimo vértice ao j-ésimo vértice. Se não houver um caminho do i-ésimo vértice ao j-ésimo vértice, a célula será deixada como infinito.

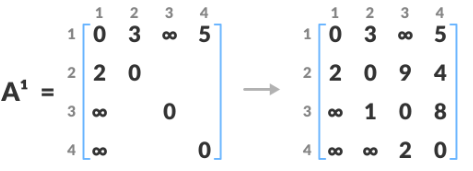


1. Now, create a matrix A1 using matrix A0. The elements in the first column and the first row are left as they are. The remaining cells are filled in the following way.

Let k be the intermediate vertex in the shortest path from source to destination. In this step, k is the first vertex.A[i][j] is filled with (A[i][k] + A[k][j]) if (A[i][j] > A[i][k] + A[k][j]).

That is, if the direct distance from the source to the destination is greater than the path through the vertex k, then the cell is filled with A[i][k] + A[k][j].

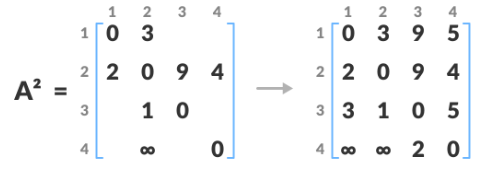
In this step, k is vertex 1. We cacluate the distance from source vertex to destination vertex through this vertex k.



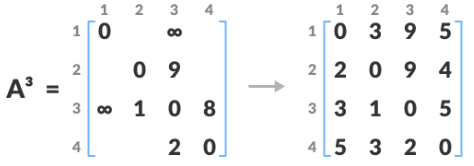
For example: For A1[2, 4], the direct distance from vertex 2 to 4 is 4 and the sum of the distance from vertex 2 to 4 through vertex (ie. from vertex 2 to 1 and from vertex 1 to 4) is 7. Since 4 < 7, A0[2, 4] is filled with 4.

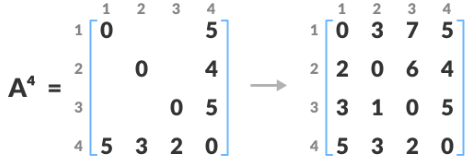
1. In a similar way, A2 is created using A3. The elements in the second column and the second row are left as they are.

In this step, k is the second vertex (i.e. vertex 2). The remaining steps are the same as in step 2.



1. Similarly, A3 and A4 is also created.





1. A4 gives the shortest path between each pair of vertices.
2. Considerações finais
3. Referências

Afonso, Johnnatan (2010). Teoria dos Grafos – Primeiro trabalho prático Caminhos Mínimos.

Souza, Rafael (2015). Uma abordagem paralela do algoritmo de Floyd para solução do problema do caminho mínimo.

Neves, Patricia (2007). Variações e aplicações do algoritmo de Dijkstra.

Risald, Mirino, A. E., & Suyoto. (2017). Best routes selection using Dijkstra and Floyd-Warshall algorithm.

Park, Jaehyun. (2015). Shortest Path Algorithms.

A.Machado, Rogério (1976). Avaliação de alternativas na tomada de decisão.

<http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0034-75901976000200005>

Crabtree, Steve (2011). A Talent for Making Decisions.

<https://news.gallup.com/businessjournal/394/talent-making-decisions.aspx>

https://reader.elsevier.com/reader/sd/pii/S089812211100633X?token=D5B8B863B39AAC21D8C4BB365BB1917416BECC7A36348B262B8DF079DB6355E40677885EDC8F50B37BCF38119BD33922

Pesquisa operacional, 8° edição, taha, Elsevier,

<https://plataforma.bvirtual.com.br/Leitor/Pu'blicacao/689/pdf/0>

<https://plataforma.bvirtual.com.br/Leitor/Publicacao/42049/pdf/0>

https://plataforma.bvirtual.com.br/Leitor/Publicacao/177879/pdf/0