

RICARDO NORIO MIYATA

EXPLORANDO A CENTRALIDADE DOS CAMINHOS MÍNIMOS EM GRAFOS.

*(versão pré-defesa, compilada em 13 de dezembro de 2021)*

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Murilo V. G. da Silva.

CURITIBA PR

2021

## **RESUMO**

O objetivo deste trabalho é o estudo da medida de centralidade dos caminhos mínimos. Para o cômputo da medida de centralidade foi executado o algoritmo de Dijkstra para a busca de caminhos mínimos a partir de todos os vértices em um grafo e para armazenamento e manipulação de dados, foi utilizado o uso de dicionários como estrutura de dados mais comum para armazenar os caminhos mínimos com seus respectivos valores de centralidade. O que motiva este trabalho é o fato de que os grafos representam muito bem situações da vida real, como mapas, redes sociais, entre outras. A busca por rotas otimizadas em mapas é um exemplo do escopo de grafos que cobre o problema de caminhos mínimos. Com isso, torna-se muito relevante analisar os valores de centralidades dos caminhos mínimos em um grafo, um exemplo disto, no tráfego de dados em uma rede cabeada, existem pontos (rotas) nesta infraestrutura que devem ter estruturas físicas mais robustas para suportar a alta transferência de dados e afins. Os resultados apresentados neste trabalho permitirão visualizar a teoria algorítmica proposta, juntamente com o comportamento da distribuição dos valores de centralidade.

Palavras-chave: Teoria dos grafos. Caminhos mínimos. Algoritmo de Dijkstra.

## **ABSTRACT**

The objective of this work is to study the measure of centrality of the shortest paths. For the computation of the centrality measure, the Dijkstra algorithm was used to search for shortest paths from all vertices in a graph and for data storage and manipulation, the use of dictionaries was used as the most common data structure to store the shortest paths with their respective values of centrality. What motivates this work is the fact that the graphs represent real life situations very well, such as maps, social networks, among others. Searching for optimized routes on maps is an example of the scope of graphs that covers the shortest path problem. With this, it becomes very relevant to analyze the values of centralities of the shortest paths in a graph, an example of this, in data traffic in a wired network, there are points (routes) in this infrastructure that must have more robust physical structures to support the high data transfer and the like. The results presented in this work will allow us to visualize the proposed algorithmic theory, together with the behavior of the distribution of centrality values.

Keywords: Graph theory. Shortest path. Dijkstra's algorithm.

## LISTA DE FIGURAS

2.1	Representação visual de grafos.. . . . .	12
2.2	Direcionamento de grafos. . . . .	13
2.3	Um grafo direcionado de 8 vértices, com ênfase para o caminho do vértice 0 ao 6.	14
2.4	Um grafo e uma floresta com duas árvores.. . . . .	15
2.5	Uma rede de coautoria de físicos e matemáticos aplicados. . . . .	15
2.6	Representação de uma rede (grafo) de um clube de karatê. . . . .	15
3.1	A execução do algoritmo de Dijkstra. O $s$ de origem é o vértice mais à esquerda. As estimativas de caminho mais curto aparecem dentro dos vértices e as bordas sombreadas indicam os valores predecessores. Os vértices pretos estão no conjunto $S$ e os vértices brancos estão na fila de prioridade mínima $Q = V - S$ . (a) A situação imediatamente antes da primeira iteração do <i>loop while</i> das linhas 5 – 9. O vértice sombreado tem o valor $d$ mínimo e é escolhido como vértice $u$ na linha 6. (b) - (f) A situação após cada iteração sucessiva do laço <i>while</i> . O vértice sombreado em cada parte é escolhido como vértice $u$ na linha 6 da próxima iteração. Os valores $d$ e predecessores mostrados na parte (f) são os valores finais (Cormen et al., 2009). . . . .	17
3.2	Grafo não direcionado com as centralidades de intermediação dos vértices.. . . .	19
3.3	Grafo simples com 5 nós. . . . .	20
5.1	Distribuição dos valores de centralidade, normalizados, dos caminhos mínimos. .	27
5.2	Distribuição dos valores de centralidade, não normalizados e na escala logarítmica, dos caminhos mínimos. . . . .	27

## **LISTA DE TABELAS**

5.1	Detalhamento das redes selecionadas. . . . .	26
5.2	Valores de centralidade do grafo simples, Algoritmo 4.2. . . . .	26

## **LISTA DE ACRÔNIMOS**

DINF	Departamento de Informática
PPGINF	Programa de Pós-Graduação em Informática
UFPR	Universidade Federal do Paraná
APSP	All-pairs Shortest Path

## LISTA DE SÍMBOLOS

$V$	Vértices de um grafo
$E$	Arestas de um grafo
$G(V, E)$	Um grafo com $V$ vértices e $E$ arestas
$d(v)$	Grau do vértice $v$
$\pi$	pi
$T_u$	Árvore de caminhos mínimos de um vértice $u$
$\mathcal{B}_u(v)$	Dado $T_u$ , $\mathcal{B}_u(v)$ é um ramo do vértice $u$ ao vértice $v$

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>9</b>
1.1	JUSTIFICATIVA . . . . .	9
1.2	OBJETIVOS E MOTIVAÇÃO . . . . .	9
1.3	ORGANIZAÇÃO DO TRABALHO . . . . .	10
<b>2</b>	<b>CONCEITOS INTRODUTÓRIOS. . . . .</b>	<b>11</b>
2.1	TEORIA DE GRAFOS . . . . .	11
2.2	REPRESENTAÇÃO VISUAL . . . . .	11
2.3	GRAU DE VÉRTICES E NÚMERO DE ARESTAS . . . . .	12
2.4	DIRECIONAMENTO . . . . .	12
2.5	SUBGRAFOS E SUPERGRAFOS . . . . .	13
2.6	PASSEIOS, CAMINHOS E CICLOS . . . . .	13
2.7	ÁRVORES E FLORESTAS. . . . .	14
2.8	MODELAGENS DE GRAFOS REAIS . . . . .	14
<b>3</b>	<b>CAMINHOS MÍNIMOS . . . . .</b>	<b>16</b>
3.1	ALGORITMO DE DIJKSTRA . . . . .	16
3.2	ÁRVORE DE CAMINHOS MÍNIMOS . . . . .	17
3.3	CAMINHOS MÍNIMOS ENTRE TODOS OS PARES DE VÉRTICES (APSP) .	18
3.4	COMPLEXIDADE DE TEMPO . . . . .	18
3.5	CENTRALIDADE DE INTERMEDIACÃO ( <i>BETWEENNESS CENTRALITY</i> ) .	18
3.6	CENTRALIDADE DE CAMINHOS MÍNIMOS . . . . .	19
3.7	REDUNDÂNCIA DE CAMINHOS MÍNIMOS . . . . .	20
<b>4</b>	<b>METODOLOGIA . . . . .</b>	<b>22</b>
4.1	ALGORITMO . . . . .	22
<b>5</b>	<b>RESULTADOS. . . . .</b>	<b>26</b>
<b>6</b>	<b>CONCLUSÕES . . . . .</b>	<b>28</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>30</b>



# 1 INTRODUÇÃO

## 1.1 JUSTIFICATIVA

As redes estão em todo lugar, desde a internet com a estrutura da World Wide Web, economia, geografia à transmissão de doenças, estes tipos de dados são comumente modelados por grafos, pois são considerados um método muito conveniente para ilustrar visualmente as relações nos dados.

Tendo-se que os grafos têm alta representatividade de dados da vida real, a manipulação e análise destes resultam em inúmeras soluções e facilidades para aplicações na realidade. O problema dos caminhos mínimos, tópico deste trabalho, é encontrar um percurso entre dois vértices (ou nós) em um grafo de forma que a soma dos pesos de suas arestas constituintes seja minimizada, este conceito facilmente representa algumas situações reais como busca por rotas otimizadas em mapas, análise de manutenibilidade de rodovias centrais (rotas mais usadas), estudo de transmissibilidade de surtos de doenças (COVID-19), dentre outros. Dado estes exemplos, a busca por caminhos mínimos mais importantes (caminhos mínimos de alta centralidade) nos permite analisar e definir quais são os meios mais comuns de propagação. No exemplo de rodovias, quais precisarão de maior monitoramento para manutenção, no caso de transmissibilidade viral, tentar mitigar os meios mais comuns de transmissão do vírus (rotas aéreas para regiões com surto da doença, por exemplo).

## 1.2 OBJETIVOS E MOTIVAÇÃO

Este trabalho instiga o estudo da centralidade de caminhos mínimos em um grafo. A análise de centralidade de vértices em grafos tem sido muito discutida e já dispõe de diversos algoritmos cujo entendimento, tanto do ponto de vista teórico como prático, têm sido investigados na literatura (Williams, 2014) (Pettie e Ramachandran, 2002). No escopo de caminhos mínimos o algoritmo que computa os valores exatos de centralidade de caminhos mínimos é extremamente custoso, tornando-o inviável para aplicações em grafos muito grandes, justamente os que, maioritariamente, representam situações da vida real. Contudo, existem linhas de pesquisas para otimização deste processo de computação, o cálculo por amostragem (de Lima et al., 2021) é um exemplo recentemente proposto para a solução do problema. Dito isto, este trabalho irá passar o conceito base para a computação dos valores de centralidade dos caminhos mínimos (algoritmo de força bruta), apresentando resultados, dados numéricos e gráficos visuais para estudo de comportamento e ressalta, adicionalmente, a questão da aleatorização de rotulagem.

### 1.3 ORGANIZAÇÃO DO TRABALHO

De forma geral, o trabalho está dividido em três partes: introdução dos conceitos básicos; apresentação da metodologia para a computação da centralidade dos caminhos mínimos; apresentação dos resultados e suas respectivas análises e conclusões. A introdução visa definir os conceitos principais para o entendimento do algoritmo proposto neste trabalho, para a computação dos valores de centralidades dos caminhos mínimos, definições como, vértices, arestas, direcionamento, árvores, caminhos, dentre outros. O capítulo metodologia apresenta o processo passo a passo do algoritmo para o cálculo dos valores de centralidade dos caminhos mínimos, desde o formato de recepção de dados (dados de entrada) à saída (impressão na saída padrão) dos valores computados. E, por fim, a última parte, resultados e conclusões, apresenta os dados obtidos pelo algoritmo, como as tabelas de centralidade, representação gráfica destes valores, análise de comportamento, juntamente com as conclusões.

## 2 CONCEITOS INTRODUTÓRIOS

Este capítulo introduz, brevemente, os conceitos básicos de grafos que serão utilizados ao longo dessa monografia. Os conceitos apresentados neste capítulo podem ser acessados nas seguintes bibliografias: (Bondy e Murty, 1976); (West, 2002); (Bondy e Murty, 2008); (Feofiloff et al., 2018).

### 2.1 TEORIA DE GRAFOS

Muitas situações do mundo real podem ser convenientemente descritas por meio de um diagrama que consiste em um conjunto de pontos, juntamente com linhas que unem certos pares destes pontos (Bondy e Murty, 1976). Por exemplo, os pontos podem representar pessoas, com linhas unindo pares de amigos ou os pontos podem ser centros de comunicação, com linhas representando links de relacionamento. Tais diagramas buscam saber se dois pontos dados são unidos por uma linha, a maneira pela qual eles são unidos é irrelevante. Uma abstração matemática de situações desse tipo dá origem ao conceito de grafos (Bondy e Murty, 2008).

**Definição 1** Um grafo  $G$  é denotado por um par ordenado  $(V(G), E(G))$ , onde  $V(G)$  é um conjunto de vértices e  $E(G)$  é um conjunto de ligações entre pares de vértices de  $G$ , chamados de arestas. São denotadas por  $n = |V(G)|$  e  $m = |E(G)|$  as cardinalidades dos conjuntos  $V(G)$  e  $E(G)$ , respectivamente.

### 2.2 REPRESENTAÇÃO VISUAL

Os grafos têm esse nome porque podem ser representados graficamente. Cada vértice é indicado por um ponto e cada aresta por uma linha que une os pontos que representam suas extremidades (Bondy e Murty, 1976).

Existem diversas maneiras de representação para um grafo. Os pontos (vértices) e conexões (arestas) são meramente um modelo visual da estrutura de dados, isto é, não têm significado.

A seguir, dois exemplos de representações de dois grafos,  $G$  e  $H$ .

**Exemplo 1** Considere o grafo  $G$ , Figura 2.1(a), onde

$$V(G) = \{u, v, w, x, y\}$$

e

$$E(G) = \{a, b, c, d, e, f, g, h\}$$

**Exemplo 2** Considere o grafo  $H$ , Figura 2.1(b), onde

$$V(H) = \{v_0, v_1, v_2, v_3, v_4, v_5\}$$

$e$

$$E(H) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}\}$$

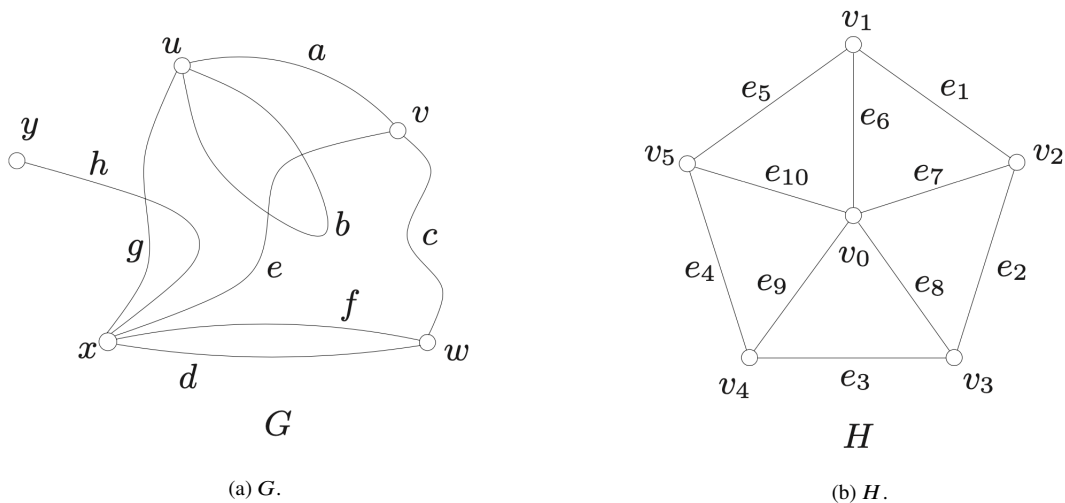


Figura 2.1: Representação visual de grafos.

### 2.3 GRAU DE VÉRTICES E NÚMERO DE ARESTAS

O número de arestas incidentes a um vértice se refere ao grau deste vértice. Algumas bibliografias ainda subdividem este conceito em definições de leques de entrada e saída, ou seja, o grau do leque de saída de um vértice se dá pelas arestas que "saem do vértice" e, de forma intuitiva, o grau de um vértice de entrada é o número de arestas que incidem em um vértice, "entram no vértice". De forma complementar, percebe-se que a soma dos graus de saída de todos os vértices de um grafo é igual ao número de arestas do grafo. A soma dos graus de entrada de todos os vértices também é igual ao número de arestas. Segue daí que um grafo com  $V$  vértices tem no máximo  $V(V - 1)$  arestas. Observe que com isso, um grafo com  $n$  vértices pode ter  $O(n^2)$  arestas.

**Definição 2** Dado um grafo  $G = (V(G), E(G))$ , o grau de um vértice  $v$ , denotado por  $d(v)$ , é o número total de arestas de  $G$ , incidentes a  $v$ .

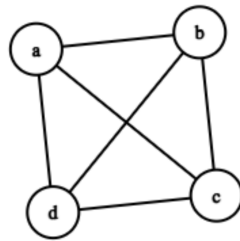
### 2.4 DIRECIONAMENTO

Usando-se das definições anteriores, nesta seção iremos explanar umas das principais propriedades de grafos no que diz respeito ao relacionamento entre vértices de um grafo. Os

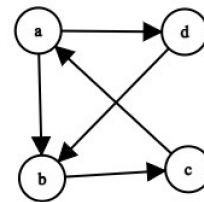
grafos podem ou não ser direcionados, pondo-se em exemplos reais, a relação de amizade em uma rede social por exemplo, pode ser denotada como não direcionada, pois um amigo  $a$  é amigo de  $b$ , assim como o inverso. Já em rotas e trajetos (mapas), as ruas possuem direcionamento, isto é, para um exemplo em que uma rua possui mão única, a relação do ponto de interseção de seu início e fim pode ser dado como de  $a$  para  $b$ , porém não o inverso, pois seria considerado contramão.

Considere a Figura 2.2(a). Este é um exemplo de um grafo  $G$ , não direcionado, onde  $V(G) = \{a, b, c, d\}$  e  $E(G) = \{\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}\}$ . Nota-se que as arestas  $\{a, b\}$  e  $\{b, a\}$  são considerados redundantes, motivo pela qual o conjunto  $E(G)$  apresenta somente uma delas.

Considere a Figura 2.2(b). Este é um exemplo de um grafo  $G$ , direcionado, onde  $V(G) = \{a, b, c, d\}$  e  $E(G) = \{ab, ad, bc, ca, db\}$ . A aresta  $ab$ , indica que há uma relação de  $a$  para  $b$  e o inverso não é verdade.



(a) Grafo não direcionado.



(b) Grafo direcionado.

Figura 2.2: Direcionamento de grafos.

## 2.5 SUBGRAFOS E SUPERGRAFOS

Dado um grafo  $G$ , um subgrafo de  $G$  é um grafo com todos os seus vértices e arestas em  $G$ . Se  $G'$ , é um subgrafo de  $G$ , então  $G$  é um supergrafo de  $G'$ . Um subgrafo de extensão é um subgrafo, contendo todos os pontos de  $G$ . Para qualquer conjunto  $S$  de pontos de  $G$ , o subgrafo induzido  $G[S]$  é o subgrafo máximo de  $G$  com conjunto de pontos  $S$ . Desta forma, dois pontos de  $S$  são adjacentes em  $G[S]$ , se e somente se, eles são adjacentes em  $G$ .

**Definição 3** Um grafo  $H$  é um subgrafo de  $G$ , escrito como  $H \subseteq G$  se  $V(H) \subseteq V(G)$  e  $E(H) \subseteq E(G)$ .

## 2.6 PASSEIOS, CAMINHOS E CICLOS

Previamente, foram definidos alguns princípios básicos dos grafos, vértices e arestas. Considerando os princípios até agora apresentados, esta seção visa definir as noções de passeios e caminhos em um grafo.

Um passeio em um grafo é uma sequência de vértices dotada da seguinte propriedade: se  $v$  e  $w$  são vértices consecutivos na sequência, então  $(v, w)$  é uma aresta do grafo (note que o

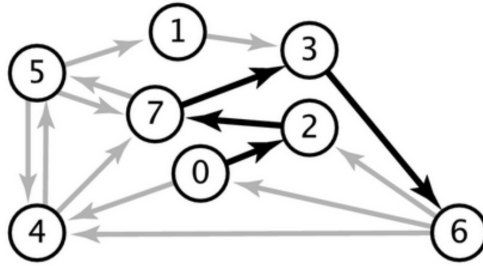


Figura 2.3: Um grafo direcionado de 8 vértices, com ênfase para o caminho do vértice 0 ao 6.

inverso de um passeio não é, em geral, um passeio). Uma aresta do passeio é qualquer aresta  $(v, w)$  do grafo tal que  $w$  é o sucessor de  $v$  no passeio. Um passeio é fechado se tem pelo menos duas arestas e seu primeiro vértice coincide com o último.

Um caminho em um grafo é um passeio sem arestas repetidas, ou seja, um passeio em que as arestas são todas diferentes entre si. Um caminho é *simples* se não tem vértices repetidos. Por exemplo,  $(0, 2, 7, 3, 6)$  é um caminho simples no grafo  $G$  da Figura 2.3.

Todas as arestas de um caminho apontam na mesma direção de um vértice para o seu sucessor. Há quem goste de enfatizar esse fato dizendo caminho dirigido em vez de caminho.

A origem de um caminho é o seu primeiro vértice. O término é o seu último vértice. Se um caminho tem origem  $s$  e término  $t$ , dizemos que vai de  $s$  a  $t$ . O comprimento de um caminho é o número de arestas do caminho. Se um caminho tem  $n$  vértices, seu comprimento é pelo menos  $n - 1$ ; se o caminho é simples, seu comprimento é exatamente  $n - 1$  (Feofiloff et al., 2018).

Tendo-se das definições acima, um ciclo em um grafo nada mais é que um caminho fechado, ou seja, tem comprimento maior que 1, sem arestas repetidas.

**Definição 4** Dado um grafo  $G = (V(G), E(G))$ , um passeio  $P = (v_1, \dots, v_k)$  em  $G$  é uma sequência de vértices de  $G$ , tal que, para todo par consecutivo de vértices  $(v_{i-1}, v_i)$ , com  $2 \leq i \leq k$ , temos que  $v_{i-1}v_i$  é uma aresta de  $G$ . Se não houverem vértices repetidos em  $P$ , dizemos que  $P$  é um caminho. Um ciclo no grafo  $G$  é uma sequência de vértices  $C = (v_1, v_2, \dots, v_k)$  se  $P = (v_1, \dots, v_k)$  é um caminho em  $G$  e  $v_1v_k \in E(G)$ .

## 2.7 ÁRVORES E FLORESTAS

Em grafos, uma árvore é um grafo conexo sem ciclos. Uma floresta é um grafo com cada componente conectado em uma árvore. Uma folha em uma árvore é qualquer vértice de grau 1. A Figura 2.4 apresenta uma árvore e uma floresta com duas árvores.

## 2.8 MODELAGENS DE GRAFOS REAIS

Como mencionado na introdução deste trabalho, grafos são ferramentas matemáticas amplamente utilizadas na modelagem de redes complexas de larga escala. A seguir apresento

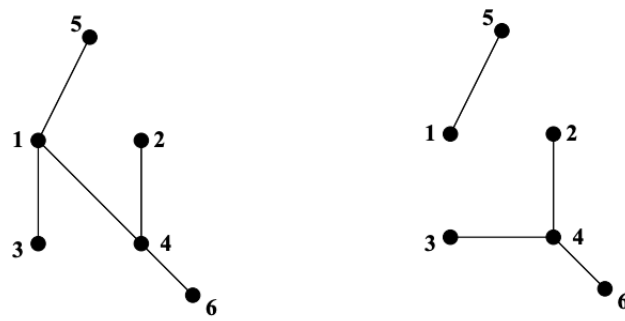


Figura 2.4: Um grafo e uma floresta com duas árvores.

exemplos de representações de aplicações da vida real: o primeiro, grafo da Figura 2.5 descreve uma rede de coautorias entre um conjunto de físicos e matemáticos aplicados (Easley e Kleinberg, 2010); o segundo, grafo da Figura 2.6 se refere à rede de um clube de karatê, uma disputa entre o presidente do clube (nó 34) e o instrutor (nó 1).

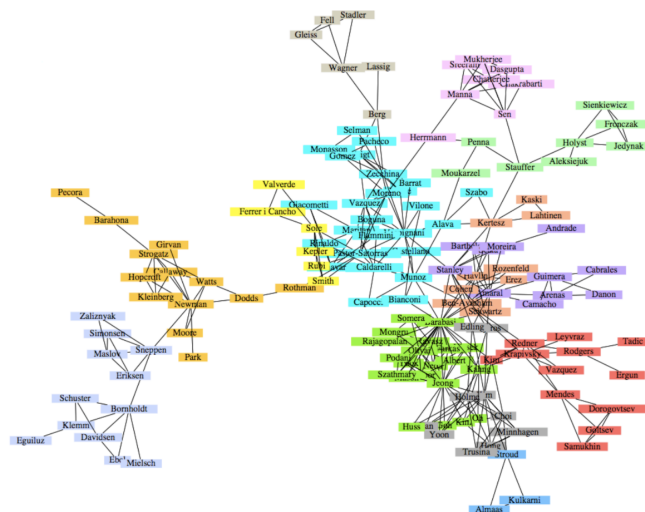


Figura 2.5: Uma rede de coautoria de físicos e matemáticos aplicados.

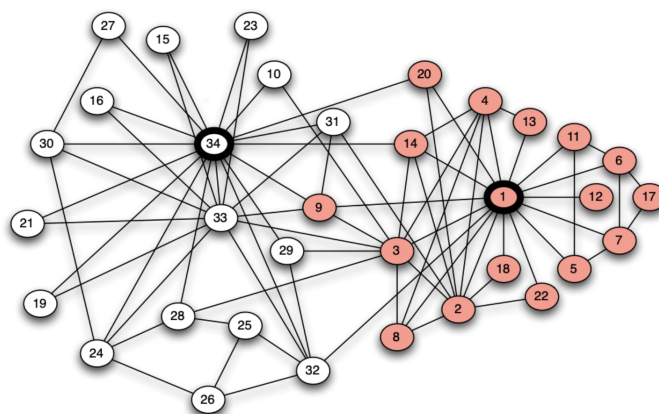


Figura 2.6: Representação de uma rede (grafo) de um clube de karatê.

### 3 CAMINHOS MÍNIMOS

Usando-se das propriedades de grafos até agora citados, caminhos mínimos é mais uma propriedade que visa, dado dois vértices  $v_1$  e  $v_2$ , buscar qual o caminho mais curto de  $v_1$  para  $v_2$ . O *peso* (*ponderação*) de arestas é mais uma propriedade adicional para este capítulo. As arestas de um grafo, além de todas as definições previamente definidas, podem ou não possuir peso, isto é, em um exemplo da vida real, dado um mapa, os pesos das arestas para um grafo que o representasse, poderiam se referir a distância em quilômetros, com isto, o grafo seria classificado como direcionado e ponderado. Quando não se é destacado que o grafo possui pesos, consideramos que todas as suas arestas possuem o mesmo peso ou apenas que não o possuem.

#### 3.1 ALGORITMO DE DIJKSTRA

O algoritmo de Dijkstra resolve o problema de caminhos mínimos de fonte única em um grafo direcionado e ponderado  $G = (V, E)$  para o caso em que todos os pesos das arestas não são negativos.

O algoritmo de Dijkstra mantém um conjunto  $S$  de vértices cujos pesos finais de caminhos mínimos dos  $s$  de origem já foram determinados. O algoritmo seleciona repetidamente o vértice  $u \in V - S$  com a estimativa do caminho mais curto, adiciona-se  $u$  a  $S$  e relaxa todas as arestas, deixando  $u$ . Na implementação a seguir, usa-se uma fila de prioridade mínima  $Q$  de vértices, codificados por seus valores de  $d$  (Cormen et al., 2009).

```

1  DIJKSTRA( $G, w, s$ )
2  INITIALIZE-SINGLE-SOURCE( $G, s$ )
3   $S = \text{\textbackslash emptyset}$ 
4   $Q = G * V$ 
5  while  $Q \neq \text{\textbackslash emptyset}$ 
6       $u = \text{EXTRACT-MIN}(Q)$ 
7       $S = S \cup \{u\}$ 
8      for each vertex  $v \in G.\text{Adj}[u]$ 
9          RELAX( $u, v, w$ )

```

O algoritmo de Dijkstra relaxa as bordas, conforme mostrado na Figura 3.1. A linha 2 inicializa os valores de  $d$  de maneira usual, e a linha 3 inicializa o conjunto  $S$  com o conjunto vazio. O algoritmo mantém a invariante  $Q = V - S$  no início de cada iteração do *loop while* das linhas 5 – 9. A linha 4 inicializa a fila de prioridade mínima  $Q$  para conter todos os vértices em  $V$ ; desde que  $S = \emptyset$ , naquele momento, o invariante é verdadeiro após a linha 4. Cada vez que, através do *loop while* das linhas 5 – 9, a linha 6 extrai um vértice  $u$  de  $Q = V - S$  e a linha 7 o adiciona ao conjunto  $S$ , mantendo assim o invariante. (A primeira vez, através deste loop,  $u = s$ .) O vértice  $u$ , portanto, tem a menor estimativa de caminho mínimo de qualquer vértice em  $V - S$ .



Então, as linhas 8 – 9 relaxam cada aresta  $(u, v)$ , deixando  $u$ , atualizando assim a estimativa  $v.d$  e o predecessor  $v.\pi$  se pudermos melhorar o caminho mínimo encontrado até agora, passando por  $u$ . Observe que o algoritmo nunca insere vértices em  $Q$  após a linha 4 e que cada vértice é extraído de  $Q$  e adicionado a  $S$  exatamente uma vez, de modo que o *loop while* das linhas 5 – 9 itera exatamente  $|V|$  vezes (Cormen et al., 2009).

Como o algoritmo de Dijkstra sempre escolhe o vértice "mais leve" ou "mais próximo" em  $V - S$  para adicionar ao conjunto  $S$ , dizemos que ele usa uma estratégia "gulosa". Este tipo de estratégia nem sempre produz resultados ótimos (Cormen et al., 2009).

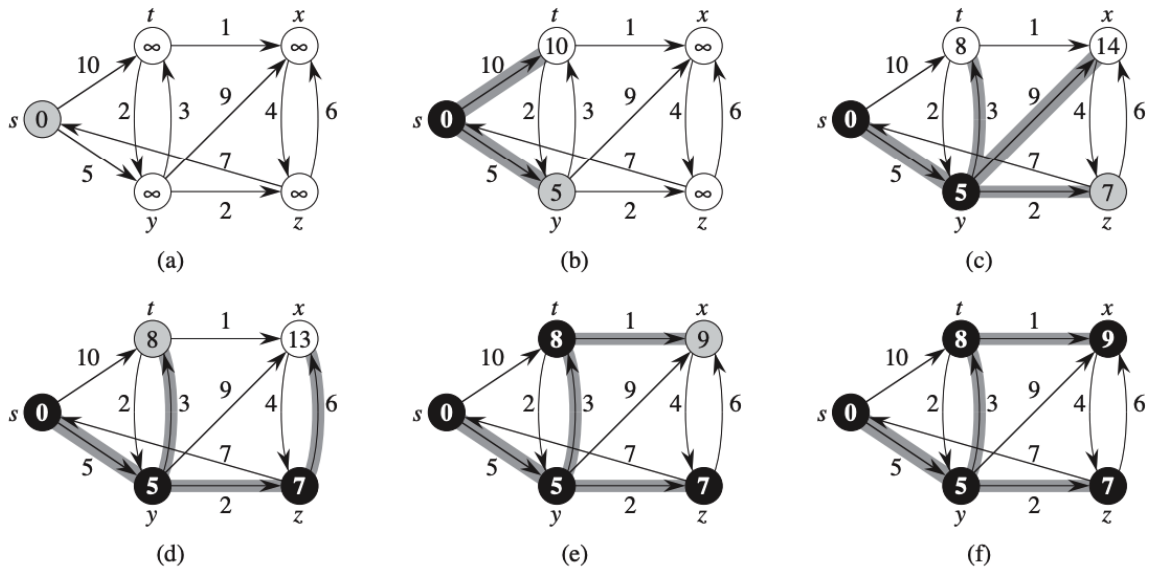


Figura 3.1: A execução do algoritmo de Dijkstra. O  $s$  de origem é o vértice mais à esquerda. As estimativas de caminho mais curto aparecem dentro dos vértices e as bordas sombreadas indicam os valores predecessores. Os vértices pretos estão no conjunto  $S$  e os vértices brancos estão na fila de prioridade mínima  $Q = V - S$ . (a) A situação imediatamente antes da primeira iteração do *loop while* das linhas 5 – 9. O vértice sombreado tem o valor  $d$  mínimo e é escolhido como vértice  $u$  na linha 6. (b) - (f) A situação após cada iteração sucessiva do laço *while*. O vértice sombreado em cada parte é escolhido como vértice  $u$  na linha 6 da próxima iteração. Os valores  $d$  e predecessores mostrados na parte (f) são os valores finais (Cormen et al., 2009).

**Definição 5** O algoritmo de Dijkstra, executado em um grafo direcionado ponderado  $G = (V, E)$  com função de peso não negativa  $w$  e fonte  $s$ , termina com  $u.d = \delta(s, u)$  para todos os vértices  $u \in V$ .

### 3.2 ÁRVORE DE CAMINHOS MÍNIMOS

Uma árvore de caminhos mínimos de um vértice  $u$  é uma árvore geradora  $T_u$  de  $G$ , de modo que o caminho de  $u$  para todos os outros vértices de  $T_u$  é um caminho mínimo em  $G$ . Tomaremos a árvore de caminho mínimos para um vértice  $u$  como sendo  $T_u$ . Um caminho mínimo que começa na raiz de uma árvore Dijkstra é chamado de ramo de  $G$ . Mais formalmente, dado  $T_u$ , para cada  $v \neq u$ , o caminho mínimo de  $u$  a  $v$  é um ramo, denotado por  $\mathcal{B}_u(v)$ . Além

disso, cada subcaminho de  $\mathcal{B}_u(v) = p_{uv}$  é também um caminho mínimo em  $G$ , e denotamos esse conjunto de subcaminhos (incluindo  $p_{uv}$ ) como  $S(p_{uv})$  ou  $S(\mathcal{B}_u(v))$  (ambas as notações são usadas indistintamente, por conveniência). Uma vez que  $G$  não é direcionado, o mesmo se aplica a caminhos em ordem reversa, ou seja, cada subcaminho de  $p_{vu}$  em  $T_u$  também é um caminho mínimo. Com isto, assumimos  $S(p_{vu})$  como sendo este conjunto de caminhos mínimos (de Lima et al., 2021).

### 3.3 CAMINHOS MÍNIMOS ENTRE TODOS OS PARES DE VÉRTICES (APSP)

O APSP, do inglês, *all pairs shortest paths* é o problema de calcular o comprimento mínimo entre cada par de vértices em um grafo ponderado. O problema APSP é muito bem estudado e tem havido resultados recentes para uma variedade de suposições para o grafo de entrada (direcionado, não direcionado, ponderado, não ponderado, etc.) (de Lima et al., 2021) (Williams, 2014) (Pettie e Ramachandran, 2002).

### 3.4 COMPLEXIDADE DE TEMPO

O tempo de execução no pior caso para o algoritmo de Dijkstra em um grafo com  $n$  nós e  $m$  arestas é  $\theta((m + n)\log(n))$ . Ele até encontra os caminhos mais curtos de um nó de origem  $s$  para todos os outros nós no grafo. Isso é basicamente  $O(n \log n)$  para seleção de nó e  $O(m)$  para atualizações de distância. Embora  $O(n \log n)$  seja a melhor complexidade possível para grafos densos, a complexidade pode ser melhorada significativamente para grafos esparsos. Com pequenas modificações, o algoritmo de Dijkstra pode ser usado como um algoritmo reverso que mantém árvores geradoras mínimas para o nó sorvedouro. Com outras modificações, ele pode ser estendido para se tornar bidirecional. O gargalo no algoritmo de Dijkstra, caso o grafo seja esperso, é a seleção de nós. No entanto, usando a implementação de Dial, isso pode ser melhorado significativamente para grafos esparsos (Len et al., 2021).

É possível resolver o problema de encontrar os caminhos mínimos de todos os pares de vértices de um grafo, executando um algoritmo de caminhos mínimos de fonte única  $|V|$  vezes, uma vez para cada vértice como fonte. Se todos os pesos das arestas forem não negativos, podemos usar o algoritmo de Dijkstra. Existem variações deste algoritmo, otimizando a complexidade de tempo, o mais famoso deles, o algoritmo de Floyd-Warshall, cuja complexidade de tempo é  $\theta(n^3)$  e depende apenas de  $n$ .

### 3.5 CENTRALIDADE DE INTERMEDIACÃO (*BETWEENNESS CENTRALITY*)

A centralidade de intermediação é uma maneira de detectar a quantidade de influência que um nó tem sobre o fluxo de informações em um grafo. Em outras palavras, usa-se para localizar os nós que servem como "ponte" de uma parte de um grafo para outra ou que inferem possuir maior "importância".

Com o cálculo de todos os pares de caminhos mínimos em um grafo, os nós recebem uma pontuação, esta se refere a um montante de "importância" do nó em relação ao número de caminhos mínimos que passam pelo nó. Resumindo, os nós que mais frequentemente estiverem presentes nos caminhos mínimos de um grafo terão pontuações de centralidade de intermediação mais altas.

A Figura 3.2 apresenta um grafo não direcionado colorido com base na centralidade da intermediação de cada vértice do menor (vermelho) ao maior (azul).

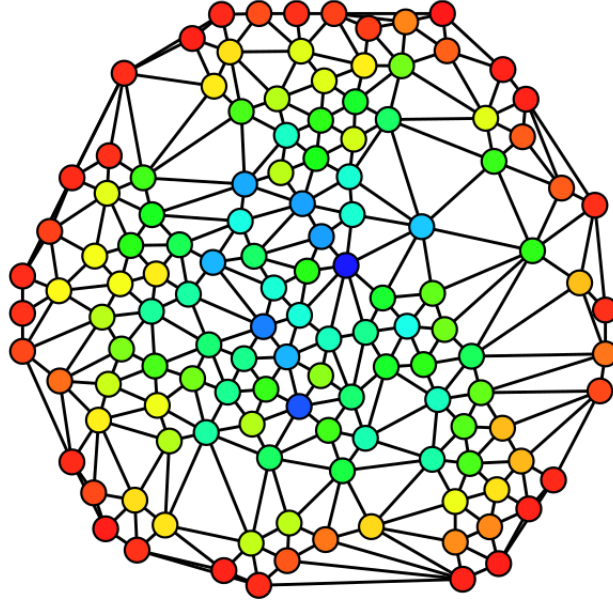


Figura 3.2: Grafo não direcionado com as centralidades de intermediação dos vértices.

Nesta mesma linha, podemos avaliar a "importância" de arestas ou, de forma mais relevante, dos caminhos mínimos, atribuindo-lhes pontuações toda vez que estes estão presentes como sub caminhos mínimos em outros caminhos mínimos. Isto desperta a importância e motiva ao estudo de, assim como na centralidade de intermediação em vértices, a centralidade de caminhos mínimos que será discutido na próxima seção.

### 3.6 CENTRALIDADE DE CAMINHOS MÍNIMOS

A seguir apresento a definição da centralidade de caminhos mínimos de um grafo.

**Definição 6** (*Centralidade de caminhos mínimos*) Dado um grafo não direcionado ponderado  $G = (V, E)$  com  $n = |V|$ , um par  $(u, v) \in V^2$  e a árvore Dijkstra  $T_a$  para cada  $a \in V$ , sejam  $p_{ab} = (a, \dots, b)$  e  $p_{uv} = (u, \dots, v)$  os caminhos mínimos de  $a$  para  $b$  e  $u$  para  $v$ , respectivamente, tais que  $p_{ab}, p_{uv} \in S(G)$ . A centralidade do caminho mínimo de um par  $(u, v) \in V^2$  é definida como

$$c(u, v) = \frac{t_{uv}}{n(n-1)}$$

onde

$$t_{uv} = \sum_{(a,b) \in V^2: a \neq b} \mathbb{1}_{\tau_{uv}(\mathcal{B}_a(b))}$$

A função  $\mathbb{1}_{\tau_{uv}(\mathcal{B}_a(b))}$  retorna 1 se houver algum caminho mínimo de  $u$  a  $v$  como subcaminho do ramo  $\mathcal{B}_a(b) \in S(T_a)$  (e 0 caso contrário). Intuitivamente falando, um par  $(u, v)$  tem alta centralidade de caminho mínimo se o caminho mínimo canônico  $p_{uv} \in S(T_u)$  (e  $S(T_v)$ ) é um subcaminho de um grande número de caminhos canônicos mínimos em  $S(G)$ .

Considere o grafo da Figura 3.3, o percurso mais curto do vértice  $v_1$  para o vértice  $v_3$  é o caminho mínimo  $(v_1, v_3)$ . Calculando-se todos os possíveis pares de caminhos mínimos deste grafo, chegaremos a conclusão de que o caminho mínimo  $(v_1, v_3)$  terá maior centralidade que o caminho mínimo  $(v_1, v_5, v_4)$  (percurso mais curto do vértice  $v_1$  para o  $v_4$ ) por exemplo, isto porque as vezes em que o caminho mínimo  $(v_1, v_3)$  está presente como subcaminho mínimo é maior que as vezes em que o caminho mínimo  $(v_1, v_5, v_4)$  está presente em outros.

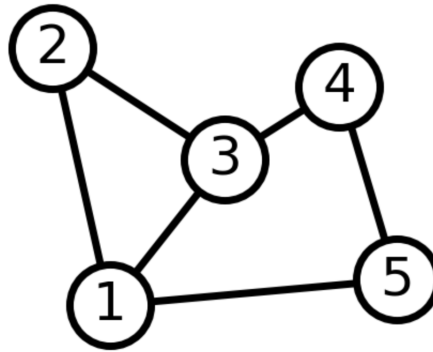


Figura 3.3: Grafo simples com 5 nós.

Assim como ocorre com o problema da centralidade de intermediação, a melhor maneira conhecida de se computar a centralidade de caminhos mínimos também é utilizando algoritmos que resolvem o APSP.

### 3.7 REDUNDÂNCIA DE CAMINHOS MÍNIMOS

Adicionalmente, ainda considerando o grafo da Figura 3.3, do vértice  $v_1$  ao vértice  $v_4$ , um algoritmo de busca por caminhos mínimos pode retornar o percurso mais curto como sendo  $\langle v_1, v_3, v_4 \rangle$  ou  $\langle v_1, v_5, v_4 \rangle$ , porém nos importa apenas uma destas saídas e usaremos a que o algoritmo em questão encontrar. Para a busca de caminhos mínimos, usaremos o algoritmo de Dijkstra e assumiremos seu retorno como verdade. Esta informação é importante e curiosa, pois dependendo de como o algoritmo se comporta/ trata os dados, o resultado do conjunto de caminhos mínimos podem diferir e, consequentemente, os valores de centralidade dos caminhos mínimos nos grafos que possuam mais de uma opção de percurso mais curto de entre um par de vértices. Este aspecto de aleatorização dos rótulos de um grafo é, inclusive, uma linha de pesquisa

neste mesmo escopo, mas algumas bibliografias já indicam que, em grafos muito esparsos, a aleatorização dos rótulos não resulta em diferenças significativas nos valores de centralidade dos caminhos mínimos (de Lima et al., 2021).

## 4 METODOLOGIA

De um ponto de vista algorítmico e de estruturas de dados, os grafos são amplamente utilizados para representar redes em geral, sejam estas representações de mapas, redes sociais ou até mesmo o fluxo de influências numa rede bibliográfica. Com isto, existem muitas e das mais diversas linhas de pesquisas envolvendo algoritmos e teoria dos grafos. Os estudos vão desde análise de uma rota otimizada em um mapa à análise do surto de uma transmissão viral. Seguindo-se nesta linha, o estudo sobre a análise da centralidade de caminhos mínimos é bastante relevante em diversos cenários da vida real. Um exemplo, dado um mapa, existem rotas que sempre são mais comumente usadas por transportes de carga rodoviário, desta forma, espera-se que este trajeto demande de mais frequente manutenção. Este e muitos outros cenários nos levam a entender a importância do estudo e análise da centralidade de caminhos mínimos.

O problema de centralidades em grafos é um problema muito explorado e já existem diversas técnicas para análise da centralidade de vértices, porém a centralidade de arestas, caminhos mínimos, presente neste mesmo escopo pesquisa, ainda carece de estudos. É um problema extremamente interessante de exploração, pois representa um enorme leque de aplicações da vida real.

Neste trabalho foram implementados e executados, experimentos de conceitos introduzidos no trabalho de doutorado da Alane Marie de Lima e publicado em (de Lima et al., 2021). O problema da centralidade de caminhos mínimos é explorado no trabalho da Alane de um ponto de vista de aplicabilidade, pois o algoritmo e todo seu processo, já em teoria, são extremamente custosos em tempo de execução para grafos muito esparsos. A pesquisa dela busca métodos de análise da centralidade de caminhos mínimos de forma mais eficiente.

Dito isto, este trabalho irá mostrar a aplicação de um algoritmo exato em três exemplos de grafos. O objetivo central é investigar, mesmo que preliminarmente, a distribuição dos valores de centralidade de caminhos mínimos. Os grafos que serão utilizados neste trabalho não são direcionados.

Para apresentação visual e conceitual da metodologia serão utilizados: um grafo simples, com 5 vértices; o modelo de grafo do clube de karatê de Zachary, que é uma rede social de um clube universitário de karatê (Zachary, 2009); a rede de jogos de futebol americano entre as faculdades da Divisão IA durante a temporada regular do outono de 2000 (NetworkX, 2021).

### 4.1 ALGORITMO

Dados os grafos de entrada, a ideia geral do trabalho é obter todos os possíveis caminhos mínimos, isto é, executar o algoritmo de Dijkstra para cada um dos  $n$  vértices do grafo. Como já mencionado, o algoritmo de Dijkstra, dado um vértice *target*, constrói uma árvore de caminhos mínimos, sendo cada percurso da raiz até os nós da árvore, um caminho mínimo da raiz ao nó.

Situações em que o grafo é conexo, a quantidade de caminhos mínimos de um vértice para todos os demais é de  $n - 1$ . Este processo será executado  $n$  vezes, isto é, para todos os vértices do grafo.

O trabalho foi desenvolvido em *Python 3*, com uso da biblioteca *NetworkX* (Developers, 2021).

A função *main()*, serve como chamada central de outras subfunções que fazem o tratamento dos dados e que serão explicadas em sequência. De forma geral, esta função faz a construção de um grafo (linhas 2 - 4), determina os caminhos mínimos para todos os vértices (linhas 5 - 8), faz a computação de centralidade dos caminhos mínimos de um grafo (linha 7) e exibe, na saída padrão, todos os caminhos mínimos com seus respectivos valores de centralidade (linha 8).

Listing 4.1: Função central para chamada de subfunções.

```

1  def main() -> None:
2      G = simple_graph_generator()
3      G = karate_club_graph_generator()
4      G = football_graph_generator()
5
6      dijkstraTrees = get_dijkstra_trees_from_a_graph(G)
7      shortestPaths = get_shortest_paths(dijkstraTrees)
8      all_shortest_paths_centrality(shortestPaths)
9      print_all_paths_and_centrality(shortestPaths)

```

Para a geração de grafos de entrada, meu código possui 3 funções: o Algoritmo 4.2 (*simple\_graph\_generator()*), o Algoritmo 4.3 (*karate\_club\_graph\_generator()*) e o Algoritmo 4.4 (*football\_graph\_generator()*). Estas funções usam a biblioteca *NetworkX* para construção manual do grafo simples, chamada do grafo, já predefinido na biblioteca, do clube de karatê e chamada por fonte externa do grafo da rede de jogos de futebol americano.

Listing 4.2: Função para gerar um grafo simples.

```

1  def simple_graph_generator():
2      G = nx.Graph()
3      G.add_edge(1, 2)
4      G.add_edge(1, 3)
5      G.add_edge(1, 5)
6      G.add_edge(2, 3)
7      G.add_edge(3, 4)
8      G.add_edge(4, 5)
9      return G

```

Listing 4.3: Função para gerar o modelo de grafo do clube de karatê de Zachary.

```

1  def karate_club_graph_generator():
2      G = nx.karate_club_graph()
3      for v in G:
4          print(f"{v:4} {G.degree(v):6}")

```

```
5         return (G)
```



```

9         pathDict[path] = newPathInfo
10
11     return pathDict

```

Agora sim, tendo-se os dados refinados, esta etapa visa calcular a centralidades dos caminhos mínimos, Algoritmo 4.7 (*all\_shortest\_paths\_centrality()*). A centralidades dos caminhos mínimos do grafo é calculada da seguinte forma: para cada caminho mínimo  $cm_1$ , verifique em todos os demais caminhos mínimos se  $cm_1$  está presente como subcaminho mínimo (*linhas 2 - 5*); em caso positivo, contabilize +1 na centralidade de  $cm_1$  (*linha 6*); estas etapas são repetidas para todos os demais caminhos mínimos (*linhas 2 - 6*); feito isto, agora a última etapa é referente a normalização dos valores (*linhas 11 - 13*), pois é comumente visto valores de ponderação com valores entre 0 e 1, ou seja, quanto mais próximo de 1, maior a centralidade do caminho mínimo.

Listing 4.7: Função para cálculo da centralidade dos caminhos mínimos.

```

1 def all_shortest_paths_centrality(pathDict: dict) -> dict:
2     # Check if i is contained in j. If it is, plus one
3     for i in pathDict:
4         for j in pathDict:
5             if i in j:
6                 pathDict[i].centrality += 1
7
8     # Get the number of Dijkstra trees
9     numTrees = len(pathDict)
10
11     # Normalizing centrality values
12     for path in pathDict:
13         pathDict[path].centrality /= numTrees * pathDict[path].numNodes

```

## 5 RESULTADOS

Para a execução dos testes, foram selecionados três grafos, sendo um primeiro bastante simples, utilizado para visualização exata dos valores de centralidade para cada caminho mínimo. Os demais dois grafos são modelos da vida real, clube de karatê de Zachary e a rede de jogos de futebol americano. Buscou-se selecionar redes que contivessem número de vértices e arestas distintas, para que fosse possível obter resultados não viciados em relação ao tamanho do grafo sobre os quais seriam executados os testes. Dito isso, as redes selecionadas representam diferentes tipos de redes e possuem quantidades de vértices que variam de 5 a 115. A Tabela 5.1 apresenta a descrição das redes nas quais foram executados os testes propostos.

Tabela 5.1: Detalhamento das redes selecionadas.

Rede	Vértices	Arestas	Tipo	Classificação
Simple	5	6	Não direcionado	Teste
Zachary	34	78	Não direcionado	Social
Football	115	613	Não direcionado	Informação

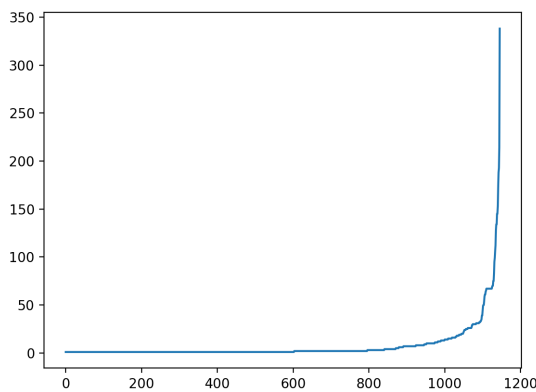
A Tabela 5.2 representa os valores de centralidade calculados para o grafo mais simples, Algoritmo 4.2. A ordem apresentada na tabela depende muito do contexto de análise, para este grafo, foi feita a ordenação de tal forma a agrupar ou deixar mais próximos os vértices iguais, por exemplo, as primeiras linhas desta tabela estão ordenadas primeiro com todos os caminhos mínimos que iniciam no vértice 1 e assim por diante.

Tabela 5.2: Valores de centralidade do grafo simples, Algoritmo 4.2.

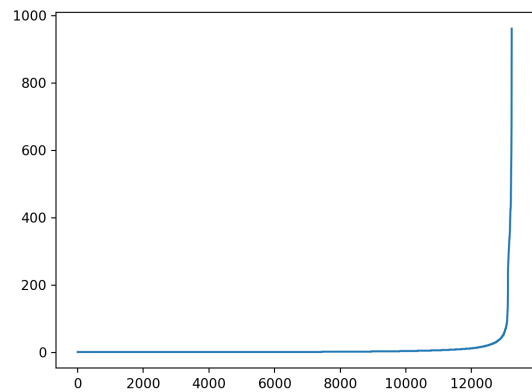
Path	Centrality
[1, 2]	[0.050]
[1, 3]	[0.075]
[1, 5]	[0.075]
[1, 3, 4]	[0.016]
[2, 1]	[0.050]
[2, 3]	[0.050]
[2, 1, 5]	[0.016]
[2, 3, 4]	[0.016]
[3, 1]	[0.075]
[3, 2]	[0.050]
[3, 4]	[0.075]
[3, 1, 5]	[0.016]
[5, 1]	[0.075]
[5, 4]	[0.025]
[5, 1, 2]	[0.016]
[5, 1, 3]	[0.016]
[4, 3]	[0.075]
[4, 5]	[0.025]
[4, 3, 1]	[0.016]
[4, 3, 2]	[0.016]

Já para os grafos do clube de karatê de Zachary e da rede de jogos de futebol americano foram apresentados em gráficos, pois suas tabelas são muito extensas. Serão apresentados dois tipos de gráficos, um com os valores de centralidade dos caminho mínimos normalizados e outro com os valores de centralidade, não normalizados e na escala logarítmica, dos caminhos mínimos.

O Gráfico 5.1(a) e o Gráfico 5.1(b) distribuem os valores de centralidade, normalizados, dos caminhos mínimos no eixo y, de forma ordenada (crescente), enquanto o eixo x representa apenas os caminhos mínimos em si, ou seja, estão rotulados sequencialmente, pois nesta visualização o percurso exato de cada um dos caminhos mínimos não nos interessa.



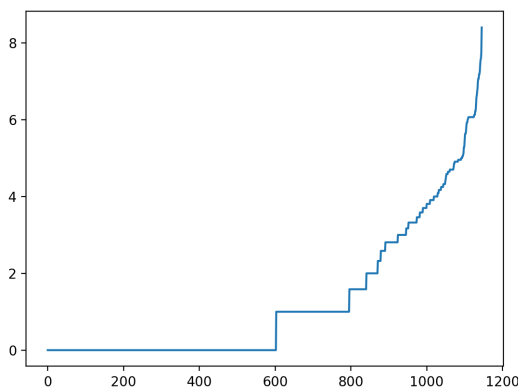
(a) Clube de karatê de Zachary.



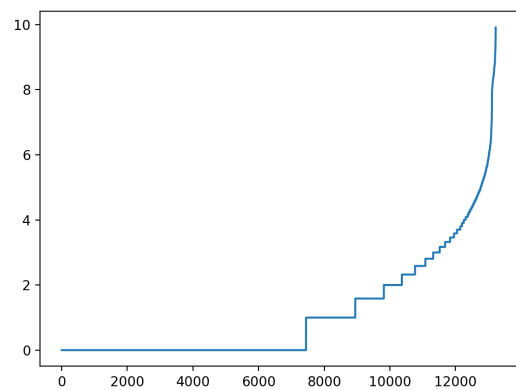
(b) Rede de jogos de futebol americano.

Figura 5.1: Distribuição dos valores de centralidade, normalizados, dos caminhos mínimos.

O Gráfico 5.2(a) e o Gráfico 5.2(b) distribuem os valores de centralidade, não normalizados e na escala logarítmica, dos caminhos mínimos no eixo y, de forma ordenada (crescente).



(a) Clube de karatê de Zachary.



(b) Rede de jogos de futebol americano.

Figura 5.2: Distribuição dos valores de centralidade, não normalizados e na escala logarítmica, dos caminhos mínimos.

## 6 CONCLUSÕES

Neste trabalho, foram usados três tipos de grafos de entrada, um deles é muito simples e não reflete situações da vida real, que são os tipos de aplicações que nos interessam. O propósito do uso de um grafo mais simples neste trabalho é permitir o entendimento mais claro do algoritmo para a computação da centralidade de caminhos mínimos proposto. Já os outros dois grafos, são modelos que refletem condições da vida real, dito isto, podem ser inviáveis ou muito difíceis de analisar visualmente ou por *teste de mesa*. É interessante e útil ver o comportamento e a distribuição dos valores de centralidade dos caminhos mínimos nos grafos selecionados, pois, pelos gráficos obtidos, constata-se que os grafos possuem um comportamento muito parecido entre eles, segundo a relação de caminhos mínimos.

Vale ressaltar que o algoritmo proposto neste trabalho para o problema de caminhos mínimos é muito custoso. Dito isto, pode se tornar inviável para uso em algumas aplicações da vida real, porém este trabalho visa despertar a curiosidade referente a importância e utilidade deste tipo de análise.

Do tempo de execução, pelos testes realizados é possível constatar que o tempo de processamento do grafo de entrada e cálculo dos valores de centralidade para todos os caminhos mínimos, cresce rapidamente conforme o aumento do número de vértices. Importante destacar isto, pois corrobora a necessidade de se explorar algoritmos mais eficientes, inclusive, já existem propostas de algoritmos que calculam o valor da centralidade de caminhos mínimos por meio de amostragem (de Lima et al., 2021).

Da análise dos valores de centralidade dos caminhos mínimos. De forma intuitiva, já se percebe um comportamento e relação entre o tamanho do percurso de um caminho mínimo e seu valor de centralidade, em outras palavras, caminhos mínimos com um percurso maior tendem a ter valores de centralidade menores. Explicando, dado um caminho mínimo, ele só poderá estar contido como subcaminho em algum outro, se e somente se, este conter pelo menos a mesma quantidade de vértices, ou seja, quanto menor o percurso, mais possibilidades de este estar contido como subcaminho mínimo em outros e, conseqüentemente, maior o seu valor de centralidade. Em resumo, pode-se dizer que, quanto maior o percurso de um caminho mínimo, menor seu valores de centralidade e, de forma análoga, quanto menor o percurso de um caminho mínimo, mais provável de seu valor de centralidade ser maior.

Do comportamento do gráfico de distribuição dos valores de centralidade dos caminhos mínimos. O Gráfico 5.1(a) e o Gráfico 5.1(b) apresentam a disposição dos valores de centralidade ordenados de forma crescente, isto nos permite constatar que a curva é sempre crescente. Isto é importante destacar, pois indica que os caminhos mínimos tendem a ser muito centrais ("importantes") ou pouco centrais e, outro fator interessante disto, assim como supracitado, os caminhos mínimos dispostos mais à esquerda, os valores menores, referem-se aos caminhos

mínimos com percurso maior, enquanto que, de forma análoga, os caminhos mínimos dispostos mais à direita dos gráficos, representam os percursos maiores.

Dito tudo isto, vale ressaltar um aspecto em adicional. Sabemos que existem diversas situações em que, dado um grafo, existe a possibilidade de ocorrer mais de uma opção de caminho mínimo entre um par de vértices, isto é um aspecto importante, pois, de forma lógica, os valores de centralidade possivelmente mudem. Em grafos pequenos este aspecto pode ser facilmente percebido e por isso é necessário assumir como verdade o retorno de um algoritmo de busca por caminhos mínimos, pois cada algoritmo pode tratar de forma diferente a busca. Existem algumas linhas de pesquisas que indicam que o problema da aleatorização de rotulagem tem impacto desprezível em grafos muito esparsos, entende-se disto que a aleatorização de rotulagem não é, necessariamente, um empecilho para o estudo da centralidade de caminhos mínimos em um grafo (de Lima et al., 2021).

## REFERÊNCIAS

- Bondy, J. A. e Murty, U. S. R. (1976). *Graph theory with applications*. Elsevier Science Publishing Co., Inc.
- Bondy, J. A. e Murty, U. S. R. (2008). *Graph Theory*. Springer London.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L. e Stein, C. (2009). *Introduction to Algorithms*. Massachusetts Institute of Technology.
- de Lima, A., da Silva, M. e Vignatti, A. (2021). Shortest path centrality and the apsp problem via vc-dimension and rademacher averages. <https://www.inf.ufpr.br/murilo/public/apsp-alg.pdf>. Acessado em 15/11/2021.
- Developers, N. (2021). Networkx. <https://networkx.org/documentation/stable/index.html>. Acessado em 26/11/2021.
- Easley, D. e Kleinberg, J. (2010). *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press.
- Feofiloff, P., Kohayakawa, Y. e Wakabayashi, Y. (2018). Uma introdução sucinta à teoria dos grafos. <https://www.ime.usp.br/~pf/teoriadosgrafos/>. Acessado em 06/11/2021.
- Len, G., Andreas, L. e Eric, W. (2021). Dijkstra's algorithm. <https://mathworld.wolfram.com/DijkstrasAlgorithm.html>. Acessado em 15/11/2021.
- NetworkX (2021). Football. [https://networkx.org/documentation/stable/auto\\_examples/graph/plot\\_football.html](https://networkx.org/documentation/stable/auto_examples/graph/plot_football.html). Acessado em 29/11/2021.
- Pettie, S. e Ramachandran, V. (2002). *Computing shortest paths with comparisons and additions*. Society for Industrial and Applied Mathematics, United States.
- West, D. (2002). *Introduction to Graph Theory*. Pearson Education (Singapore) Pte. Ltd., Indian Branch, 482 F.I.E. Patparganj, Delhi 1 10 092, India.
- Williams, R. R. (2014). *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*. Association for Computing Machinery, New York, NY, United States.
- Zachary, W. W. (2009). *An Information Flow Model for Conflict and Fission in Small Groups*. University of New Mexico.