Apoena Revista Eletrônica

Uma visão computacional teórico-prática do emprego da teoria dos grafos na resolução de problemas de fluxo máximo

William da Silva Santos Pinheiro, Caio Eduardo Pinheiro Costa, Rafael Alves Cotrim

Centro Universitário Jorge Amado (Unijorge) – Salvador – BA – Brasil

{Pinheiro, William} williampinheiro18@gmail.com

{Costa, Caio} caio.costa@unijorge.edu.br

{Cotrim, Rafael} rcotrim@unijorge.pro.br

Abstract. *Graph theory is an area of mathematics with great importance for several modern technological applications, such as geolocation and data analysis; its creation is relatively new, but its contributions are felt in several areas of human thought, such as chemistry, biology and mathematics itself. Among the possible uses of graph theory is its use to solve problems in which it is necessary to detect the maximum possible flow in a network that starts at a source and ends at a sink. The general objective of this research is to inform the reader about the use of graph theory in solving the maximum flow problem. The specific objectives are: a) to review the concepts of graph theory to the reader, b) to inform the reader about the maximum flow problem, c) to illustrate, through software, the use of graph theory in solving problems maximum flow. The methodology adopted consists of a literature review based on the reading of books and articles published on the subject. The aim of this work is to inform and exemplify to the reader how graph theory and its algorithms are used to solve maximum flow problems.*

Resumo. *A teoria dos grafos é uma área da matemática com grande importância para diversas aplicações tecnológicas modernas, como a geolocalização e análise de dados; sua criação é relativamente nova, mas suas contribuições são sentidas em diversas áreas do pensamento humano, como a química, biológica e a própria matemática. Dentre os empregos possíveis da teoria dos grafos está a sua utilização para resolver problemas em que seja necessário detectar o fluxo máximo possível em uma rede que se inicia em uma fonte e termine em um sumidouro. O objetivo geral da presente pesquisa é informar ao leitor sobre o uso da teoria dos grafos na resolução do problema de fluxo máximo. Os objetivos específicos são: a) revisar ao leitor sobre os conceitos da teoria dos grafos, b) informar o leitor sobre o problema de fluxo máximo, c) ilustrar, por meio de um software, o uso da teoria dos grafos na resolução de problemas de fluxo máximo. A metodologia adotada consiste em revisão de literatura realizada a partir da leitura de livros e artigos publicados sobre a temática. Espera-se com o presente trabalho informar e exemplificar ao leitor como se dá a utilização da teoria dos grafos e seus algoritmos na resolução de problemas de fluxo máximo.*

1. INTRODUÇÃO

A teoria dos grafos é um ramo da matemática em constante ascensão: grafos servem como modelos matemáticos para analisar com sucesso muitos problemas concretos do mundo real (BALAKRISHNAM et all, 2012, p. 14). Dentre os problemas que a teoria dos grafos pode suportar a resolução, estão aqueles que envolvem a maximização da transmissão de um determinado recurso em uma rede. Por exemplo, para calcular a capacidade máxima de transmissão de água em uma rede de transmissão em uma cidade. [Balakrishnam and Ranganathan 2012].

Problemas como o descrito recebem na teoria dos grafos o nome de problemas de fluxo máximo (ou problema de vazão máxima, **maximum flow problem**), e sua resolução consiste em encontrar um número, representação da solução para a transmissão de um fluxo em uma rede, tal que esse número seja o maior possível levando sempre em conta a capacidade de transmissão da rede. [Szwarcfiter 2018]

Grafos em redes estão presentes em diversas formas e em diferentes problemas do nosso cotidiano. Sua natureza é pervasiva, ou seja, se encaixam facilmente em diferentes problemas e de diferentes maneiras. As redes formadas pelo transporte de mercadorias são talvez a melhor forma de visualização da atuação dos grafos em redes em nosso cotidiano. Por exemplo, em um sistema de logística de entregas de mercadorias, é preciso organizar os pontos por onde passarão os entregadores de tal modo que se realize a maior quantidade de entregas economizando o máximo de combustível possível; para isso, é possível representar os pontos de entregas sendo vértices de um grafo e a trajetória a ser percorrida entre os pontos as arestas desse grafo. A partir dessa representação podemos utilizar um algoritmo, como o de descoberta do caminho mínimo, para realizar os cálculos necessários e encontrar a melhor trajetória para esse problema. [Ahuja and Orlin 1993]

Com o auxílio da teoria dos grafos é possível modelar um problema recorrente como o encontrado na tentativa de distribuição eficiente da tubulação de transporte de água em uma plantação agrícola como uma rede de grafos, onde cada aresta valorada do grafo represente o comprimento da tubulação que passará pelo terreno e os vértices sendo os pontos em que a tubulação precisará fazer um desvio. Para esse tipo de problema, pode-se utilizar uma árvore geradora mínima, ou seja, a árvore cuja soma dos pesos é a menor possível dentro de todas as árvores geradoras encontradas. Tal abordagem resulta em um caminho onde se alcança todos os pontos da plantação, porém economizando a tubulação utilizada. [Amarildo and Luiz 2011]

Comentado [D1]: Procurar uma imagem ilustrativa para isso. Ficaria muito bacana!

Problemas envolvendo maximização de recursos são comuns em nosso cotidiano e, pensando em resolvê-los, podemos utilizar a teoria dos grafos para modelar tais problemas e suportar a sua resolução através dos diversos algoritmos desenvolvidos na área, como o **Algoritmo de Ford-Fulkerson**. [Szwarcfiter 2018]

O presente trabalho tem como objetivo geral informar ao leitor, de forma teórica e prática, sobre o emprego da teoria dos grafos em problemas de maximização de fluxo. Os objetivos específicos do trabalho são: informar o leitor sobre a teoria dos grafos em si, visando conceituá-lo sobre os termos e conceitos que serão utilizados durante todo o trabalho; abordar o problema de fluxo máximo em grafos, desenvolvendo, para este fim, uma aplicação que utilize o algoritmo de Ford-Fulkerson para resolver problemas de fluxo máximo em grafos.

A metodologia adotada para o desenvolvimento desse trabalho consiste em uma revisão de literatura, realizada a partir da leitura de livros e artigos publicados em revistas de caráter científico sobre a temática, de forma qualiquantitativa ou qualitativa.

2. FUNDAMENTAÇÃO TEÓRICA

A teoria dos grafos possui como marco inicial de estudo um problema real e muito específico, cuja solução deu-se na elaboração de um algoritmo eficiente, ou seja, um algoritmo que consegue alcançar seu objetivo no menor tempo e com o menor uso de recursos. O problema em questão, chamado problema das pontes de Königsberg, foi solucionado pelo matemático e físico suíço Leonhard Euler (1707-1783) em 1736. [Szwarcfiter 2018].

Figura 1 Leonhard Euler, quadro a óleo por Johann Georg Brucker (Jakob Emanuel Handmann, domínio público, via Wikimedia Commons, disponível em: https://upload.wikimedia.org/wikipedia/commons/6/60/Leonhard_Euler_2.jpg))

O problema das pontes de Königsberg consistia na seguinte questão: é possível percorrer todas as pontes que divide a Königsberg (atual Kaliningrado) passando por cada ponte uma, e apenas uma vez? Euler resolveu o problema proposto imaginando cada uma das 4 partes da cidade, divididas pelo rio, como sendo um vértice de um grafo; em seguida, visualizou cada ponte como sendo as arestas desse grafo [Koh et al. 2007]. Na figura 2 podemos ver a cidade de Königsberg com suas sete pontes em destaque:

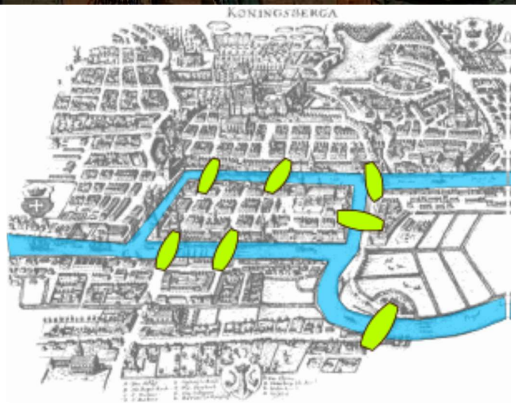

Figura 2. A cidade de Königsberg com suas sete pontes em destaque (By Bogdan Giuşcă- Public domain (PD), based on the image, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=112920)

A partir dessa visualização Euler concluiu, após muitos estudos, que para ser possível realizar tal travessia cada vértice (representação de uma parte da cidade) precisava estar conectado com um número par de partes da cidade, ou seja, para cada vértice é preciso que exista um número par de arestas. Através de sua descoberta, Euler concluiu ser impossível executar a travessia proposta pelo problema, e seu artigo a respeito da questão (denominado ***Solutio problematis ad geometriam situs pertinentes***, que em tradução significa ***A solução de um problema relativo à geometria de posição***) lançou o que viria ser a base da teoria dos grafos. Na figura 3 podemos ver o grafo formado a partir do problema analisado por Euler: [Saoub 2017]

Comentado [D2]: qual nome desse arigo? Precisa evidencia aqui, é a base da teoria do seu trabalho!

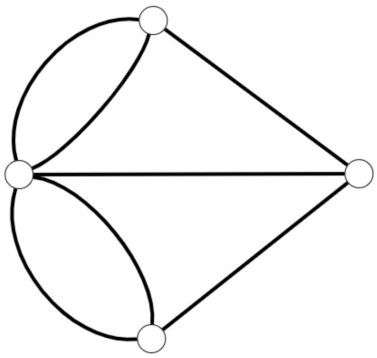

Figura 3 Grafo formado a partir do problema das pontes de Königsberg (elaborado pelo autor)

Grafos são uma forma visual e matemática de representar relações entre elementos de um conjunto utilizando para isso vértices e arestas. Segundo [Saoub 2021] "Um grafo G consiste em dois conjuntos: V(G), chamado conjunto de vértices, e E(G), chamado conjunto de arestas. Uma aresta, denotada por xy, é um par de vértices não ordenado." Informalmente, podemos definir um grafo como um conjunto de vértices interconectados por arestas e que, graficamente, podem ser representados por círculos (indicando os vértices) e linhas (indicando as arestas). Um vértice de um grafo pode incluir um número denominado ***grau do vértice***, que indica a quantidade de arestas que incidem sobre ele.

Na figura 4 temos um exemplo de grafo com 8 vértices e 7 arestas; nesse exemplo, o vértice A possui grau 2, por estar conectado a dois vértices (C e E), enquanto que o vértice C possui grau 3, por estar conectado aos vértices A, E e D:

Comentado [D3]: incluir aqui a definição de grau de um vértice.

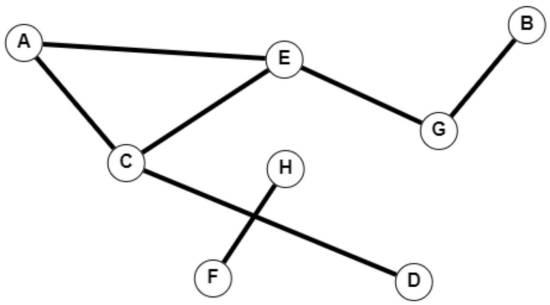

Figura 4 Exemplo de grafo (elaborado pelo autor)

Grafos valorados são grafos nos quais se atribui algum tipo de valor às arestas (ligações entre vértices). Assim, por exemplo, em um grafo representando um mapa, podemos atribuir a cada aresta (representação das ruas) um valor correspondente ao comprimento da rua. A natureza desse tipo de grafo apresentada no exemplo é **estática**, ou seja, seu valor não se altera e não sofre influência dos vértices das suas conexões [Netto and Jurkiewicz 2017]. A figura 5 exemplifica um grafo valorado com 7 vértices e 8 arestas; nota-se que para cada aresta há um valor associado:

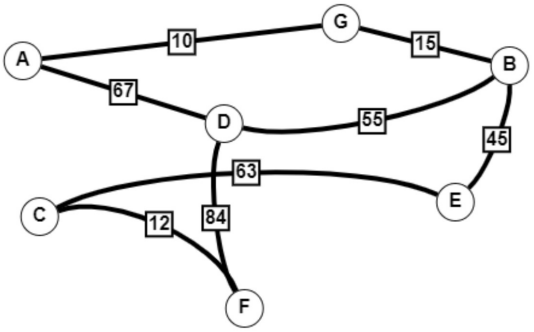

Figura 5 Exemplo de grafo valorado (elaborado pelo autor)

Há ainda um outro tipo de grafo valorado: o que possui natureza **dinâmica**. Por exemplo, em uma distribuidora de água, a tubulação que transporta o líquido para as residências o faz em determinada quantidade de litros por hora; esse líquido transportado pode ser chamado de recurso, e por trafegar ao longo de um meio (não está, portanto, fixo) recebe o nome de **fluxo**. Cada vértice que compõe esse fluxo pode ter um limite diferente de transmissão do recurso; por exemplo, em um vértice x podemos ter a capacidade de transmissão de 100 litros de água por hora, enquanto que em um vértice y, apenas 20 litros. Essa diferença na capacidade de transmissão dos vértices altera a capacidade final de transmissão do fluxo, de onde provêm a necessidade de encontrar a sua capacidade máxima. [Netto and Jurkiewicz 2017].

Ligações entre grafos podem ainda apresentar direções; trata-se então de grafos ***direcionados (ou dirigidos)*** que, por contraste, possuem os ***grafos não direcionados (ou***

não dirigidos). Um ***grafo dirigido*** é aquele em que suas arestas possuem direções; por exemplo, na figura 6, o vértice A está conectado ao vértice B (representado por uma seta), porém, o vértice B não está conectado ao vértice A, mas está conectado ao vértice C, que por sua vez não se conecta a nenhum vértice. Um exemplo da utilização de grafos direcionados está na modelagem de mapas em grafos, pois uma rua desse mapa pode ser de mão única ou mão dupla, ou seja, os carros podem trafegar apenas em uma direção ou podem trafegar em ambas as direções. Netto and Jurkiewicz 2017].

O ***grau de entrada*** de um vértice é o número que indica a quantidade de arestas direcionadas que incidem sobre ele, enquanto que o seu ***grau de saída*** é o número que indica a quantidade de arestas em que este vértice está incidindo. Por exemplo, ainda na figura 6, o grau de entrada do vértice A é 0, pois não há nenhum vértice incidindo sobre ele, enquanto que seu grau de saída é 1, pois o vértice A está incidindo apenas sobre o vértice B, que por sua vez tem grau de entrada 1 (pois A incide sobre B) e grau de saída também 1 (pois B índice sobre C).

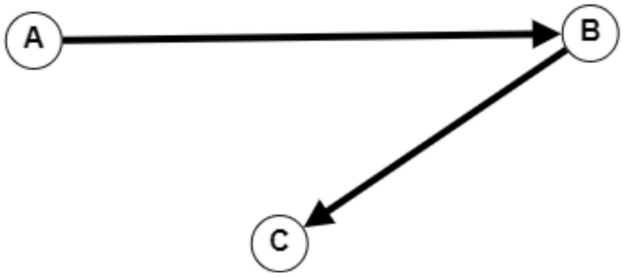

Figura 6 Exemplo de grafo direcionado (elaborado pelo autor)

Um grafo que represente um fluxo possui dois vértices distintos: o vértice **fonte** e o vértice **sumidouro**. Um ***vértice fonte*** é aquele em que o seu grau de entrada é 0, enquanto que o ***vértice sumidouro*** caracteriza-se por ter o grau de saída 0. Por exemplo, na figura 7, o vértice A é uma fonte, possuindo grau de entrada 0 e grau de saída 5, enquanto que o vértice G é um sumidouro de grau de entrada 3 e grau de saída 0. [Saoub 2021]

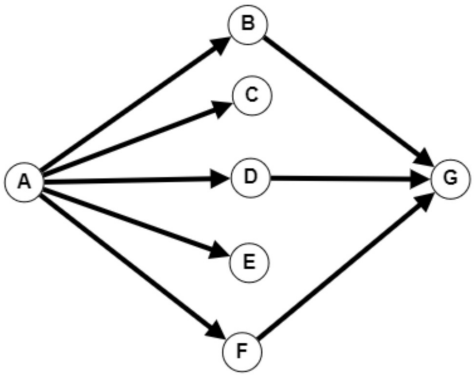

Figura 7 Exemplo de grafo com fonte e sumidouro (elaborado pelo autor)

Nos grafos de fluxo, além da direção das ligações (arestas) também encontramos valores (também chamados pesos), sendo, portanto, grafos valorados e direcionados. Tais valores

Comentado [D4]: incluir a definição de grau de entrada e saída

Comentado [D5]: já deve ter explicado grau, grau de entrada anteriormente.

Comentado [D6]: deve explicar antes, conforme comentários anteriores.

Comentado [D7]: dessa forma, retirar a definição desse parágrafo. Trazer apenas o exemplo explicando.

devem ser inteiros não negativos, e representam a capacidade de transmissão do fluxo pela aresta. Podemos definir uma função que por entrada receba o grafo em fluxo e que por saída resulte no fluxo máximo do grafo. Tal função deve seguir as seguintes restrições: cada aresta do grafo deve possuir fluxo não negativo, o fluxo não deve exceder a capacidade da aresta, o fluxo de entrada deve ser o mesmo de saída do vértice e o fluxo de saída do vértice fonte deve ser o mesmo de entrada do vértice sumidouro. Propõe-se como objetivo que se obtenha o fluxo máximo de transmissão da rede, ou seja, a capacidade máxima de saída do vértice sumidouro tendo como base a capacidade dos vértices que compõem o fluxo. [Saoub 2021]. A figura 8 ilustra um grafo em fluxo; neste exemplo, convenciona-se que o vértice fonte seja representado pela letra S e o vértice sumidouro pela letra T, para melhor identificação.

Comentado [D8]: uma função define ALGO a ALGO. O que seria, neste caso?

Comentado [D9]: essa função é bem específica, não? Tem algum nome?

Comentado [WP10R9]:

Comentado [D11]: isso é algum padrão???

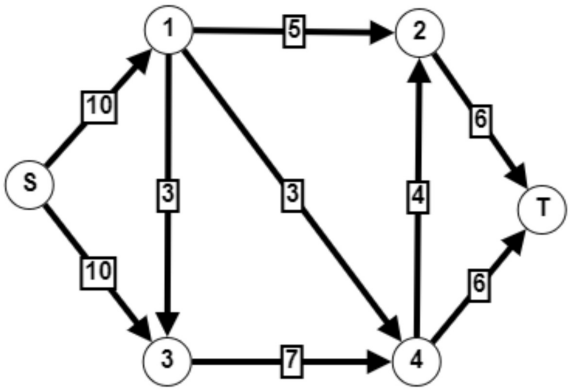

Figura 8 Exemplo de grafo em fluxo (elaborado pelo autor)

O ***algoritmo de Ford-Fulkerson*** (também conhecido como algoritmo dos pseudo-caminhos aumentadores) foi desenvolvido por Lester Randolph Ford, Jr e Delbert Ray Fulkerson, ambos matemáticos. O algoritmo tem como objetivo encontrar o fluxo de valor máximo que faça o melhor aproveitamento possível da capacidade de transmissão da rede. Seu desenvolvimento se deu entre as décadas de 1930 e 1950, através da análise de ferrovias na União Soviética. [Thomas H. Cormen 2001].

Comentado [D12]: ficou ótimo o texto!

O algoritmo de Ford-Fulkerson pode ser descrito da seguinte forma: a) cada iteração do algoritmo terá início com um fluxo F que não ultrapasse a capacidade permitida para cada arco, b) para a primeira iteração teremos um fluxo com capacidade 0, c) enquanto existirem pseudo-caminhos aumentadores, faça:

1. Gere um pseudo-caminho aumentador N,
2. Calcule a capacidade residual X para o pseudo-caminho aumentador N,
3. Transporte X unidades do fluxo em questão ao longo do pseudo-caminho N
4. Atualize o fluxo F

Comentado [D13]: confesso que fiquei confuso. Será que existia algum exemplo ou algo gráfico para visualizar? Se não tiver, tudo bem.

O tempo de execução desse algoritmo (trata-se, portanto de sua ***eficiência algorítmica***) depende da forma de se encontrar o caminho aumentador do mesmo, podendo ser utilizada a ***busca em largura*** (ou seja, o algoritmo utilizado para realizar uma travessia em um grafo do tipo árvore) para que o seu tempo de execução seja polinomial (ou seja, tenha o tempo de execução sendo $O(n^k)$, com k constante). O ***caminho aumentador*** no algoritmo de ford-fulkerson trata-se de um caminho orientado que tem início no vértice de fonte e término no vértice sumidouro, obedecendo à regra de que todo arco presente nesse

Comentado [D14]: do que se trata, mesmo?

Comentado [D15]: pode explicar sobre? Qual a característica de algo polinomial?

Comentado [D16]: vértice fonte?

caminho possua resíduo exclusivamente positivo. A ***Rede residual*** no algoritmo de ford-fulkerson trata-se do grafo cujos valores dos arcos são o resultado da diferença entre a capacidade e o fluxo do arco em questão. O menor valor dentre os presentes na rede residual é chamado "***Capacidade Residual do Caminho Aumentador***" por ser o valor que representa a quantidade de fluxo que pode ser adicionado ao caminho.

Comentado [D17]: tem algo para ilustrar? Confesso ficar um pouco complexo, apenas com teoria. Se não der, ok!

3. O USO DA COMPUTAÇÃO NA RESOLUÇÃO DO PROBLEMA DE FLUXO MÁXIMO

No presente capítulo pretende-se documentar o desenvolvimento de uma aplicação autoral, nomeada pelo autor do artigo como **Max Flow Calculator**, que encontra o fluxo máximo em uma rede de grafos através do algoritmo de Ford-Fulkerson descrito anteriormente. A aplicação permitirá que o usuário forneça os dados do grafo de sua preferência, validando-o em seguida conforme as regras de definição para uma rede de fluxo e, se pertencente a essa categoria de grafo, fornecer a sua resolução através do algoritmo citado.

Comentado [D18]: já existe? Você a nomeou? Você a criará? Importante comentar isso!
"Nomeada pelo autor do artigo como Max..."

A aplicação será desenvolvida para a plataforma web e ficará hospedada em um serviço de hospedagem gratuito para a utilização do público. A escolha por uma aplicação web se deu pela portabilidade da mesma, sendo possível ter acesso ao sistema via smartphones, tablets, desktops etc., o que aumentará a disponibilidade do sistema ao público. Além desse ponto, a alta gama de recursos fornecidos pela web foi de suma importância para a escolha dessa plataforma de desenvolvimento.

A linguagem de programação utilizada no projeto é a linguagem ***Javascript,*** criada por Brendan Eich para ser utilizada em sistemas web. Além da linguagem, outras tecnologias serão utilizadas, tais como serviço de hospedagem, banco de dados para armazenamento de arquivos, etc.

3.1 DIAGRAMAÇÃO EM UML

Comentado [D19]: esse tópico só consiste disso? Para que ele serve, mesmo? pode detalhar porque ele é importante ao seu app.

Segundo [Gilleanes 2009], a ***UML*** (Unified Modeling Language, em tradução, Linguagem de Modelagem Unificada) "é uma linguagem visual utilizada para modelar softwares baseados no paradigma de orientação a objetos". Para a representação visual do sistema, foi construído o diagrama de casos de uso, descrevendo as funções possíveis de serem executadas pelo o usuário no sistema. Na figura 9 o diagrama de casos de uso é descrito com seus principais casos.

Tal diagramação é importante pois retrata o sistema de uma forma abstraída de seus detalhes técnicos (como a linguagem de programação utilizada, por exemplo). No diagrama da figura 9 temos o sistema com suas 3 funções principais: permitir que o usuário desenhe o grafo que será avaliado pelo programa, apagar o grafo para que o sistema seja reiniciado e utilizado novamente e, por fim, calcular a solução para o problema de fluxo máximo. O algoritmo de Ford-Fulkerson será aplicado no terceiro caso de uso; o sistema então se encarregará de exibir em tela o resultado fornecido pelo algoritmo, ou seja, o valor máximo que pode trafegar por este fluxo.

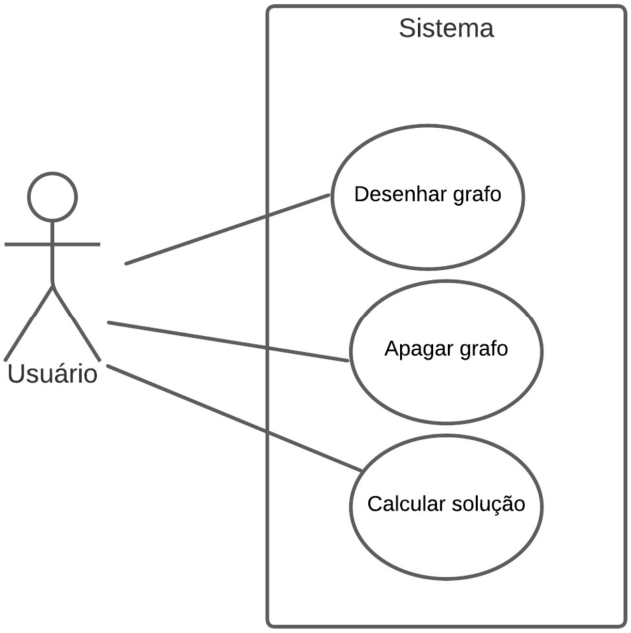

Figura 9 Diagrama de caso de uso (elaborado pelo autor)

3.2 EXEMPLO DA UTILIZAÇÃO DO SISTEMA MAX FLOW CALCULATOR

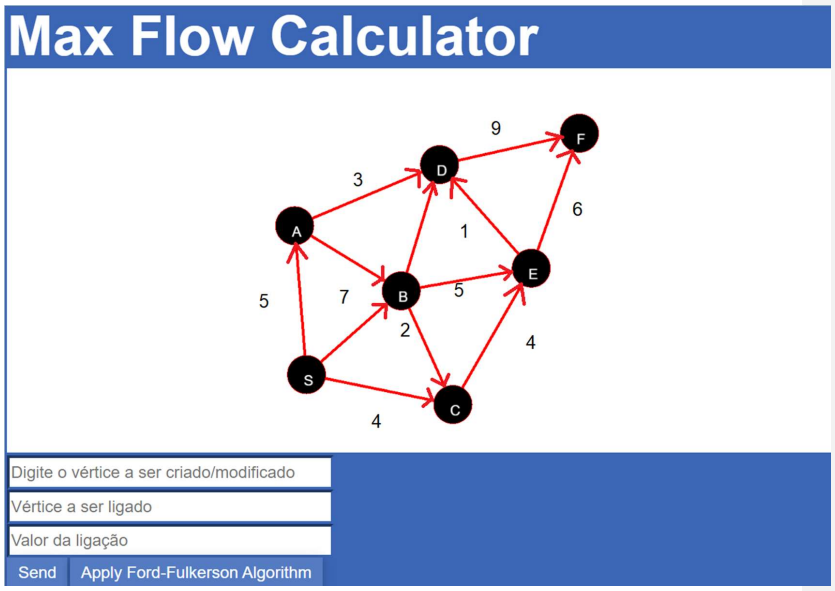

Figura 10 Exemplo da utilização do sistema Max Flow Calculator (elaborado pelo autor)

Na figura 10 podemos ver a interface gráfica do sistema **Max Flow Calculator**. Ela é composta de uma área em branco onde o grafo será desenhado conforme as instruções do usuário; caixas de interação com o sistema e dois botões, sendo o primeiro deles responsável por gerar o grafo e o segundo responsável por aplicar o algoritmo de Ford-Fulkerson no grafo criado, exibindo o resultado do processamento logo em seguida.

O sistema possui três caixas de interação, a primeira delas recebe como valor a identificação do grafo a ser criado ou modificado caso já exista; a segunda caixa de interação recebe como valor o nome do vértice ao qual o vértice criado/modificado terá ligação: por exemplo, podemos especificar aqui que o vértice A terá ligação com o vértice B; a terceira e última caixa de identificação recebe como parâmetro o valor da ligação entre os vértices que foram especificados nos dois campos anteriores: aqui podemos, por exemplo, especificar que o valor da ligação entre o vértice A e o vértice B é 7.

Após especificar todos os campos nas caixas de interação, utilizamos o botão “send” para enviar os dados para o sistema, que então desenhará na área em branco, caso os dados sejam válidos, o vértice e as ligações realizadas pelo usuário.

Quando o usuário já tiver montado o grafo, pode-se utilizar o botão “Apply Ford-Fulkerson Algorithm” para que o sistema aplique sobre o grafo o algoritmo de Ford-Fulkerson e imprima em tela o seu resultado. Caso o grafo montado pelo usuário não corresponda a um grafo de fluxo (por exemplo, se alguma das regras definidas

para esse tipo de grafo for violada) o sistema irá alertar ao usuário por meio de mensagem em tela indicando que há um erro na formação do grafo, o que impede que o algoritmo seja aplicado.

Para possibilitar o desenho do grafo em tela, foi utilizado duas bibliotecas web: **Springy** e **PixiJS**. O ***Springy*** (disponível em http://getspringy.com/) permite a criação de uma estrutura de dados em grafo e a aplicação de uma força direcionada a esse grafo, dando um efeito fisicamente realista ao grafo desenhado e distribui os seus nós corretamente sobre o plano desenhado. Na figura 11 temos um trecho da criação de um grafo utilizando a biblioteca Springy:

```
// Cria-se um grafo
var graph = new Springy.Graph();

// Adiciona-se vértices
var verticeA = graph.newNode({label: 'A'});
var verticeB = graph.newNode({label: 'B'});
var verticeC = graph.newNode({label: 'C'});

// realiza a conexão entre vértices
graph.newEdge(verticeA, verticeB);
graph.newEdge(verticeA, verticeC);
```

Figura 11 Exemplo de criação de grafo utilizando Springy (elaborado pelo autor)

PixiJS (disponível em https://pixijs.com/)é uma biblioteca de código-aberto que permite a criação de aplicações gráficas (como jogos, visualização de dados etc.) baseada em tecnologia web. O sistema **Max Flow Calculator** utiliza essa biblioteca para desenhar a parte gráfica do programa e para tratar eventos realizados pelo usuário (como o clicar do mouse, por exemplo).

4. CONCLUSÃO E TRABALHOS FUTUROS

Ao decorrer do presente trabalho foram apresentados alguns dos principais conceitos relevantes para a compreensão da teoria dos grafos e o seu problema de fluxo máximo, tais como grafos e sua utilidade na modelagem de problemas reais do cotidiano, grafos direcionados, grafos valorados e por fim grafos em fluxo. Em seguida o problema de fluxo máximo em grafos foi aprofundado, e para a sua resolução foi contextualizado e descrito o algoritmo de Ford-Fulkerson, que suporta a resolução do problema. Além da parte teórica, no presente trabalho, foi introduzida uma parte prática, em que se concentrou no desenvolvimento de um sistema de software que tem como objetivo exemplificar o uso do algoritmo de Ford-Fulkerson na resolução de problemas de fluxo máximo.

Tal sistema irá auxiliar aqueles que precisam encontrar o maior valor possível de ser trafegado em uma rede (independentemente do recurso a ser trafegado; água, energia, internet etc.) que possa ser descrita como um grafo de fluxo. A aplicação encontra-se em sua versão inicial, mas pretende-se que futuramente encontre-se à disposição de todos na internet. Pontos futuros a serem trabalhados são o design da aplicação, responsividade para atender a diversas plataformas (mobile ou desktop) e a implementação de uma animação passo a passo da resolução do algoritmo, o que trará maior compreensão para o usuário.

Comentado [D20]: aqui você deve "vender o peixe". Qual a importância de sua aplicação? Onde ela pode ser usada? Quais seus próximos passos? Ou seja, detalhe como tudo deve ser feito!

5. REFERÊNCIAS

Ahuja, R. K. and Orlin, J. B. (1993). In *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall.

Amarildo, V. and Luiz, R. R. (2011). Uma aplicação de grafos a um problema agrícola, envolvendo distribuição de água e transportes. *Engenharia na agricultura, viçosa - mg,* 19(3):203–209.

Balakrishnam, R. and Ranganathan, K. (2012). *In A Textbook of Graph Theory*. Springer, 2nd edition.

Gilleanes, G. T. A. (2009). In *UML 2 Uma Abordagem Prática*. novatec.

Koh, K.-M., Dong, F., and Tay, E. G. (2007*). In Introduction to Graph Theory: H3 Mathematics.* World Scientific Publishing Company.

Netto, P. O. B. and Jurkiewicz, S. (2017). In *Grafos: Introdução e Prática.* Blucher.

Saoub, K. R. (2017). In *A Tour through Graph Theory*. Chapman and Hall/CRC.

Saoub, K. R. (2021). In *Graph Theory: An Introduction to Proofs, Algorithms, and Applications.* Chapman and Hall/CRC, 1st edition.

Szwarcfiter, J. L. (5 abril 2018). In *Teoria computacional de grafos: Os algoritmos*. GEN LTC.

Thomas H. Cormen, Charles E. Leiserson, R. L. R. C. S. (2001). *In Introduction to Algorithms*. The MIT Press; 2nd edition.