

UNIVERSIDADE FEDERAL DO ABC  
CENTRO DE MATEMÁTICA, COMPUTAÇÃO E COGNIÇÃO  
PROJETO DE PESQUISA PARA O PROGRAMA IC – EDITAL 04/2022

---

## Algoritmos de Aproximação para o $(1, 2)$ -TSP

---

**Orientadora:** Carla Negri Lintzmayer

**Aluno:** Pedro de Luca Occulate Serra

Setembro/2023

# 1 Introdução

Problemas em Otimização Combinatória têm como objetivo encontrar a melhor solução dentro de um enorme, mas finito, conjunto de soluções possíveis. Em geral, um problema desses possui um conjunto de restrições que define o que é uma solução viável e uma função objetivo que determina o valor de cada solução viável. Um exemplo é o clássico problema do Caixeiro Viajante (TSP, de *Traveling Salesman Problem*), em que temos um conjunto de cidades e os custos entre pares delas que representam, por exemplo, a distância entre duas cidades. O objetivo é encontrar um percurso que passe por todas as cidades exatamente uma vez (Hamiltoniano) e que tenha custo mínimo.

O TSP aparece em diferentes contextos ao longo da história e essa característica contribui para sua importância no estudo de algoritmos. Relatos e documentos que datam do século 19 mostram tentativas de mercadores europeus de criarem mapas com rotas mercantes entre determinadas cidades, a fim de otimizar o percurso no quesito tempo ou dificuldade do trajeto [2]. Destaca-se ainda a campanha publicitária da empresa Procter & Gamble do ano 1960, onde era oferecido um prêmio de \$10,000 à pessoa que apresentasse o menor percurso entre determinadas cidades dos Estados Unidos. O TSP esteve presente em jogos também, como o *The Traveller's Dodecahedron: A Voyage Round the World*. Esse jogo tinha a proposta de criar um ciclo ao longo das arestas de um dodecaedro permitindo, no entanto, a repetição de arestas para tal [10].

Também é possível perceber aplicações do TSP nos dias atuais. Por exemplo, quando se quer projetar a melhor rota de uma linha de ônibus dentro de uma cidade. Ou quando se deseja visitar os principais pontos turísticos em uma viagem, planejando a melhor rota para visitar todos os pontos, ou até mesmo otimizar a quantidade de “giros” que telescópios de larga escala devem fazer para observar um determinado número de galáxias e planetas [2]. Evidencia-se, portanto, a importância e relevância que o TSP possui.

Dentro desse contexto, portanto, serão apresentados nesse relatório alguns dos algoritmos de aproximação que buscam tratar uma variação do TSP, o  $\{1, 2\}$ -TSP, que é um caso especial do TSP métrico. Para tal, são apresentados na Sessão 2 alguns conceitos essenciais que serão utilizados ao longo das demonstrações. Na Sessão 3 apresentaremos os principais problemas que ajudam a desenvolver os algoritmos apresentados nas Sessões 5 e 6. A Sessão 4 apresenta os limitantes inferiores para tais problemas. Por último, o apêndice apresenta um algoritmo de aproximação para o problema da Árvore de Steiner métrico.

## 2 Conceitos Básicos

A seguir são apresentadas as fundamentações teóricas básicas necessárias para a compreensão dos algoritmos apresentados ao longo desse documento.

## 2.1 Problemas de Otimização e Algoritmos de Aproximação

Em **problemas de otimização**, temos três elementos característicos: um conjunto de instâncias ( $I$ ), um conjunto de soluções viáveis para cada uma dessas instâncias ( $S(I)$ ) e uma função que atribui um número a cada uma dessas soluções  $S$ , que é chamado de **valor de  $S$**  denotado  $V(S)$ . Quando  $S(I)$  é vazio, dizemos que a instância é inviável; caso contrário,  $I$  é uma instância viável. [4]

Problemas de otimização podem ser de minimização, onde se interessa buscar as soluções viáveis de valor mínimo, ou de maximização, em que busca-se soluções viáveis de valor máximo. Uma solução viável para uma instância  $I$  que possui valor ótimo é denominada **solução ótima**, onde o valor de qualquer uma das soluções ótimas de uma instância  $I$  é denominado  $OPT(I)$  (Para este relatório, será utilizada a notação  $OPT_X(I)$  para denotar o valor de uma solução ótima de um problema  $X$ ). Problemas desse tipo pertencem à classe NP-difícil e, assumindo a hipótese que  $P \neq NP$ , é impossível encontrar algoritmos eficientes para esses problemas. Sendo assim, uma das abordagens utilizadas para lidar com esses problemas são os algoritmos de aproximação.

**Algoritmos de aproximação** tentam encontrar uma solução em tempo polinomial que possua uma garantia de qualidade, ou seja, um valor que não seja pior que uma fração pré determinada de uma solução ótima. Essa fração é denominada **razão de aproximação**  $\alpha$ , onde  $0 < \alpha < 1$  para problemas de maximização e  $\alpha > 1$  para problemas de minimização. Um algoritmo  $A$  é uma  $\alpha$ -aproximação se, para toda instância viável  $I$ , devolve uma solução viável  $A(I)$  [4] e, se estivermos falando de um problema de minimização, para essa solução deverá ter valer que

$$V(A(I)) \leq \alpha OPT(I) \quad (1)$$

e, se estivermos falando de um problema de maximização, deverá valer que

$$V(A(I)) \geq \frac{1}{\alpha} OPT(I). \quad (2)$$

O interesse por tais algoritmos é oferecer soluções que estão garantidamente dentro de um fator de uma solução ótima de forma eficiente.

## 2.2 Grafos

Um **grafo** é uma representação abstrata de um conjunto de objetos e das relações par-a-par que esses objetos possuem entre si. Define-se um grafo  $G$  (simples) por um par  $(V, E)$ , onde  $V$  é um conjunto finito e não vazio de elementos chamados **vértices**, e  $E$  é um conjunto de pares não ordenados de  $V$  chamados **arestas**. Denominaremos  $V(X)$  e  $E(X)$  o conjunto de vértices e arestas, respectivamente, de um grafo  $X$ , a fim de remover a necessidade de definir o par ordenado para todo grafo a todo momento.

A aresta formada pelo par  $u, v \in V(G)$  será denotada por  $uv$  (ou  $vu$ ), representando o par não ordenado  $\{u, v\} \in E(G)$ . Os vértices  $u$  e  $v$  são **extremos** da aresta  $uv$ , a aresta  $uv$  **incide** em  $u$  e em  $v$ , e os vértices  $u$  e  $v$  são **vizinhos** ou **adjacentes**. O **grau** de um vértice é o número de arestas incidentes a ele. Para um vértice  $u \in V(G)$  seu grau é denotado  $d_G(u)$ .

Dados os grafos  $G$  e  $H$ , dizemos que  $H$  é **subgrafo** de  $G$  se  $V(H) \subseteq V(G)$  e  $E(H) \subseteq E(G)$ , e denotamos essa relação por  $H \subseteq G$ . Se  $V(H) = V(G)$ , dizemos ainda que  $H$  é um **subgrafo gerador** de  $G$ . Ainda dizemos que  $G$  é **supergrafo** de  $H$ . Sejam  $X$  e  $Y$  conjuntos, de modo que  $Y \subseteq X$ . Utilizamos  $X \setminus Y$  para representar os elementos que pertencem apenas ao conjunto  $X$ . Uma maneira interessante de gerar subgrafos é adicionando ou removendo elementos desse grafo. Dado  $S \subseteq V(G)$ , o grafo gerado ao remover  $S$  de  $G$  é denotado  $G - S$ . Formalmente,  $G - S$  é o grafo tal que  $V(G - S) = V(G) \setminus S$  e  $E(G - S) = \{uv \in E(G) : u, v \notin S\}$ . Analogamente,  $G - F$  é o grafo tal que  $V(G - F) = V(G)$  e  $E(G - F) = E(G) \setminus F$ . Podemos adotar um abuso de notação quando  $S$  e  $F$  forem apenas um elemento, de modo que  $G - v = G - \{v\}$  para  $v \in V(G)$  e  $G - e = G - \{e\}$  para  $e \in E(G)$ .

De maneira semelhante, podemos gerar um supergrafo de  $G$  adicionando elementos a  $G$ . Dado  $S \not\subseteq V(G)$ , o grafo gerado ao adicionarmos  $S$  a  $G$  é denotado  $G + S$  de modo que, formalmente,  $V(G + S) = V(G) \cup S$  e  $E(G + S) = E(G)$ . Analogamente, dado  $F \not\subseteq E(G)$ , onde  $F$  é composto por pares de elementos de  $V(G)$ , denotamos como  $G + F$  o grafo gerado ao adicionar  $G$  a  $F$ . Formalmente,  $G + F$  é tal que  $V(G + F) = V(G)$  e  $E(G + F) = E(G) \cup F$ . Utilizaremos também um abuso de notação, quando  $S$  e  $F$  forem apenas um elemento. Dessa forma,  $G + v = G + \{v\}$  para  $v \notin V(G)$  e  $G + e = G + \{e\}$  para  $e \notin E(G)$ , com  $e = xy$  sendo  $x, y \in V(G)$ .

Um **passeio** em um grafo é uma sequência  $W = (v_0, v_1, \dots, v_l)$  de vértices tais que  $v_i v_{i+1} \in E(G)$ , onde  $v_0$  e  $v_l$  são chamados de **vértices extremos** e  $v_1 \dots v_{l-1}$  chamados de **vértices internos**. Um passeio é dito **aberto** se os vértices extremos são distintos e **fechado** se os vértices extremos são idênticos. Um passeio que possui todas as arestas distintas é chamado de **trilha**. Uma trilha que possui todos os vértices distintos é chamada de **caminho**. Um passeio fechado é chamado de **ciclo** se possuir todos os vértices distintos.

Um **ciclo Hamiltoniano** é um ciclo que contém todos os vértices de um grafo  $G$ . Uma **trilha Euleriana** é em uma trilha que contém todas as arestas de  $G$ . Para determinar se um grafo contém uma trilha Euleriana fechada, utilizamos o Teorema 1.

**Teorema 1** ([3]). *Um grafo conexo  $G$  contém uma trilha Euleriana fechada se e somente se todos os vértices de  $G$  tiverem grau par.*

Se o grafo contém uma trilha Euleriana fechada, ele certamente possui uma trilha Euleriana aberta, o Teorema 2 trata dessas trilhas:

**Teorema 2** ([3]). *Um grafo conexo  $G$  contém uma trilha Euleriana aberta se e somente se  $G$  contém exatamente dois vértices de grau ímpar.*

Um grafo é chamado **completo** quando todos os vértices têm grau exatamente  $n - 1$ . Ou seja, um grafo onde todo vértice  $u$  está conectado a todos os vértices  $V(G) \setminus \{u\}$ . Denotamos um grafo completo com  $n$  vértices por  $K_n$ . Um grafo  $G$  é **regular** quando o grau de todos os seus vértices é igual. Formalmente,  $G$  é  **$k$ -regular** se  $d_G(v) = k$  para todo  $v \in V(G)$ . Um exemplo é o grafo completo com  $n$  vértices,  $K_n$ , que é  $(n - 1)$ -regular. Um subgrafo gerador  $k$ -regular de um grafo  $G$  é chamado  **$k$ -fator** de  $G$ . Particularmente, um **2-fator** de  $G$  é uma união de ciclos disjuntos nos vértices.

Um **emparelhamento** em um grafo  $G$  é um conjunto de arestas não adjacentes em  $G$ . Denotando esse conjunto por  $M$ , dizemos que as duas extremidades das  $uv$  arestas com  $uv \in M$  estão **emparelhadas** por  $M$  e que cada vértice incidente em uma aresta de  $M$  está **coberto** por  $M$ . Observe que todo vértice coberto por  $M$  só vê uma aresta de  $M$ . Assim, se  $M$  cobrir todos os vértices de  $G$ , que é chamado de **emparelhamento perfeito**, então podemos construir um 1-fator de  $G$  com as arestas de  $M$ .

Seja  $G$  um grafo e  $w: E(G) \rightarrow \mathbb{R}$  uma função de peso nas arestas. A **desigualdade triangular** ocorre quando, para quaisquer vértices  $u, v, x$  pertencentes a  $V(G)$ , temos que

$$w(uv) \leq w(ux) + w(xv). \quad (3)$$

Dizemos que  $G$  é conexo se existe  $uv$ -caminho para todo par de vértices  $u, v \in V(G)$ . Um grafo  $T$  é uma **árvore** se  $T$  for conexo e não possuir ciclos. Em um grafo  $G$  que não possui ciclos, cada uma de suas componentes conexas é uma árvore, e esse grafo é chamado **floresta**. Dado um grafo  $G$  e um subgrafo  $T \subseteq G$  que é uma árvore,  $T$  é chamada de **geradora** de  $G$  se  $V(T) = V(G)$ . Ou seja,  $T$  é um subgrafo conexo que não possui ciclos e contém todos os vértices de  $G$ .

### 3 Problemas de interesse

Nesta seção vamos apresentar alguns problemas que embasam as análises dos algoritmos estudados, a começar pelo nosso problema principal.

O Problema do Caixeiro Viajante, TSP (*The Traveling Salesman Problem*, em inglês) consiste em encontrar, em um grafo completo  $G$  que tem função de custo  $w$  nas arestas, um ciclo que visite todos os vértices de  $G$ , de forma que esse ciclo possua custo mínimo. O custo é definido pelo somatório do peso das arestas que compõem esse ciclo. Perceba portanto que queremos encontrar em  $G$  um ciclo Hamiltoniano de custo mínimo. Para este relatório, focaremos a atenção sobre uma variação do TSP, o chamado TSP métrico (Problema 1), onde a função de custos  $w$  respeita a desigualdade triangular.

**Problema 1** (TSP Métrico).

**Entrada:**  $\langle G, w \rangle$ , onde  $G$  é um grafo completo e  $w: E(G) \rightarrow \mathbb{R}_{\geq 0}$  é uma função de custo

nas arestas que respeita a desigualdade triangular.

**Saída:** Um ciclo Hamiltoniano  $C$ , tal que o custo de  $C$ ,  $w(C) = \sum_{e \in E(C)} w(e)$  seja mínimo.

Outra variação do TSP, que por sua vez também é variação do TSP métrico, é o  $\{1, 2\}$ -TSP (Problema 2), em que o peso das arestas do grafo  $G$  só pode assumir valores 1 ou 2.

**Problema 2** ( $\{1, 2\}$ -TSP).

**Entrada:**  $\langle G, w \rangle$ , onde  $G$  é um grafo completo e  $w: E(G) \rightarrow \{1, 2\}$ .

**Saída:** Um ciclo Hamiltoniano  $C$ , tal que o custo de  $C$ ,  $w(C) = \sum_{e \in E(C)} w(e)$  seja mínimo.

O problema da Árvore Geradora Mínima (MST, de *Minimum Spanning Tree*) consiste em, dado um grafo conexo  $G$  e uma função de custo  $w$  nas arestas, encontrar uma árvore geradora  $T$  de  $G$  que tenha o custo mínimo.

**Problema 3** (MST - Árvore Geradora Mínima).

**Entrada:**  $\langle G, w \rangle$ , onde  $G$  é um grafo conexo e  $w: E(G) \rightarrow \mathbb{R}$ .

**Saída:** Árvore geradora  $T$  de  $G$  cujo custo  $w(T) = \sum_{e \in E(T)} w(e)$  é mínimo.

O problema do 2-fator consiste em, dado um grafo completo  $G$  e uma função de peso  $w$  nas arestas, encontrar um 2-fator de  $G$  de custo mínimo.

**Problema 4** (2-fator).

**Entrada:**  $\langle G, w \rangle$ , onde  $G$  é um grafo completo e  $w: E(G) \rightarrow \mathbb{R}$ .

**Saída:** Um 2-fator  $S \subseteq G$ , cujo custo  $w(S) = \sum_{e \in E(S)} w(e)$  é mínimo.

## 4 Limitantes inferiores para o valor da solução ótima do TSP

Os problemas da seção anterior nos auxiliam a procurar um algoritmo de aproximação para o problema TSP, nos fornecendo limitantes inferiores que ajudam a analisar o problema. Esses limitantes são descritos nas subseções a seguir.

### 4.1 MST e TSP

Seja  $G$  um grafo completo e  $w: E(G) \rightarrow \mathbb{R}_{\geq 0}$  uma função que satisfaz a desigualdade triangular. Denote por  $S_{ST}$  o conjunto de árvores geradoras de  $G$  (a sigla  $ST$  vem de *spanning tree*). Note que em  $S_{ST}$  existem árvores que são caminhos Hamiltonianos. Denominaremos  $S_{HP}$  esse subconjunto de  $S_{ST}$  ( $HP$  vem de *Hamiltonian Path*). Sendo assim, para obter um ciclo Hamiltoniano, bastaria tomar um caminho em  $S_{HP}$  e adicionar uma aresta que una suas extremidades. Portanto, o conjunto de soluções do problema TSP está, de certa forma, contido

dentro do conjunto de soluções do problema MST. Denotando o conjunto de soluções do TSP como  $S_{TSP}$ , temos:

$$S_{TSP} = \{C + vx : C \in S_{HP} \text{ e } v \text{ e } x \text{ são os extremos do caminho } C\} . \quad (4)$$

Observe que, por definição, o custo de uma árvore geradora mínima é menor que o custo de qualquer árvore em  $S_{ST}$  e, logo, em  $S_{HP}$ . Portanto, o custo de qualquer árvore em  $S_{HP}$  com uma aresta adicionada é sempre maior que o custo ótimo do problema MST. Sendo assim, temos que:

$$OPT_{TSP}(G, w) \geq OPT_{MST}(G, w) . \quad (5)$$

## 4.2 2-FATOR e TSP

Seja  $G$  um grafo completo e  $w: E(G) \rightarrow \mathbb{R}_{\geq 0}$  uma função que satisfaz a desigualdade triangular. Seja  $S_{2fat}$  o conjunto que contém todo possível 2-fator de  $G$ . Esses 2-fatores podem assumir diferentes formas, a depender do número de vértices que cada ciclo possui dentro daquele elemento de  $S_{2fat}$ , sendo 3 o número mínimo de vértices de cada ciclo e  $|V(G)|$  o número máximo. Observe que os de 2-fatores que possuem apenas um ciclo (ou seja,  $|V(G)|$  vértices) também são soluções viáveis do TSP em  $\langle G, w \rangle$ . Em particular, uma solução ótima do TSP (que tem custo  $OPT_{TSP}(G, w)$ ) está contida nesse conjunto. Assim, se esse conjunto é  $S_{TSP}$ , vale que:

$$S_{TSP} \subseteq S_{2fat} . \quad (6)$$

Por definição, o custo  $w(F)$  de qualquer 2-fator  $F$  é maior ou igual ao custo de uma solução ótima do problema 2-FATOR ( $OPT_{2-fator}(G, w)$ ) em  $\langle G, w \rangle$ . Então temos:

$$OPT_{2-fator}(G, w) \leq c(F) \quad \forall F \in S_{2fat} . \quad (7)$$

Analogamente, o custo de qualquer solução do TSP é maior ou igual que o custo da solução ótima do TSP:

$$OPT_{TSP}(G, w) \leq c(C) \quad \forall C \in S_{TSP} . \quad (8)$$

Sendo assim, por conta da expressão (6) e utilizando as inequações (7) e (8), temos a seguinte relação entre a solução ótima do TSP e do 2-FATOR:

$$OPT_{TSP}(G, w) \geq OPT_{2-fator}(G, w) . \quad (9)$$

## 5 Algoritmos para o TSP Métrico

Seja  $G$  um grafo. Uma estratégia interessante, que nos dá um bom fator de aproximação para o TSP, começa buscando uma árvore geradora mínima  $MST(G)$  nesse grafo. Pela definição de árvore geradora mínima, todos os vértices de  $G$  estarão conectados pela árvore, com custo mínimo. Agora, para gerar o ciclo, dobraremos as arestas da forma apresentada na Figura 1.

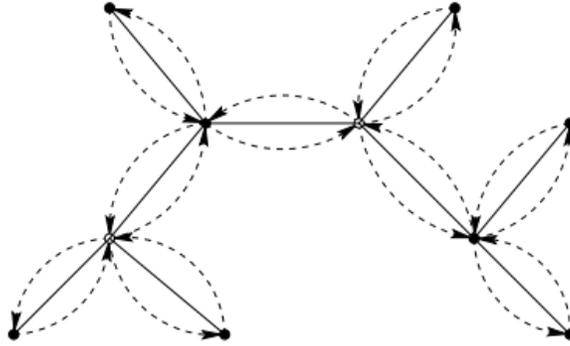


Figura 1: Exemplo de MST com arestas dobradas (Retirado de [11]).

Esse procedimento nos permite obter um subgrafo  $S$  de  $G$  que possui uma trilha Euleriana. Como queremos um ciclo Hamiltoniano, seguimos essa trilha, vértice a vértice, até que cheguemos em uma aresta que levaria a um vértice já visitado. Quando isso ocorrer, utilizamos uma aresta de  $G \setminus S$  para prosseguir a outro vértice não visitado de  $S$  e seguimos, até completar o ciclo. Essa estratégia é chamada de *short cut* (Figura 2) e, como  $G$  respeita a desigualdade triangular, é garantido que esse “atalho” não terá custo maior do que percorrer quaisquer outros vértices.

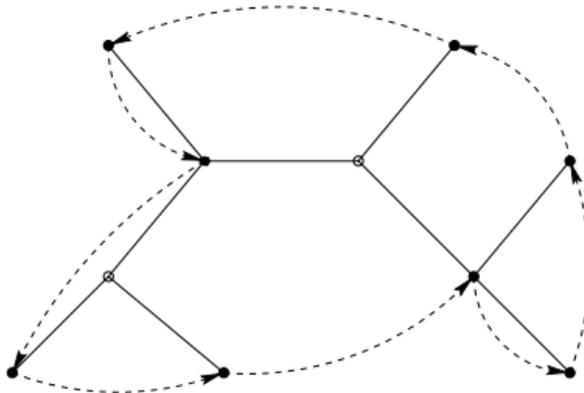


Figura 2: Procedimento de *short cut* na trilha Euleriana (retirado de [11]).

Sendo assim, por termos dobrado uma árvore geradora mínima, essa solução possui custo no máximo  $2 OPT_{MST}(G, w)$ . Pela inequação (5), temos um limitante inferior entre

$OPT_{TSP}(G, w)$  e  $OPT_{MST}(G, w)$ . Sendo assim, esse algoritmo devolve um ciclo Hamiltoniano em  $G$  cujo custo máximo  $\mathcal{A}$  é no máximo o dobro do custo ótimo do  $TSP$ ,

$$\mathcal{A} \leq 2 OPT_{TSP}(G, w), \quad (10)$$

de onde vemos que é uma 2-aproximação para o  $TSP$  métrico.

## 5.1 Melhorando resultado para 3/2-aproximação

Perceba que o método descrito anteriormente propõe encontrar uma trilha Euleriana de baixo custo. Sendo assim, apenas precisamos que as arestas do grafo possuam grau par para que ele funcione corretamente. Uma estratégia para isso foi proposta por Christofides [5]. A melhoria está justamente em como será obtido esse novo grafo. Note que, ao invés de dobrar todas as arestas do grafo, podemos apenas focar nos vértices que possuam grau ímpar em  $MST(G)$ . Seja  $O \subseteq MST(G)$  o subconjunto de vértices da  $MST(G)$  que possuem grau ímpar. Antente-se que  $|O|$  deve ser par, pelo Teorema 3, conhecido como **Teorema do Aperto de Mãos**.

**Teorema 3.** *Para todo grafo  $G$ , vale que  $\sum_{v \in V(G)} d_G(v) = 2|E(G)|$ .*

Note que, se encontrarmos um emparelhamento perfeito de custo mínimo em  $O$  e adicionarmos esse emparelhamento à  $MST(G)$ , vamos obter então um novo grafo que é um grafo Euleriano. Denote por  $M(G)$  esse emparelhamento e denote por  $J$  esse grafo. Sendo assim, seguindo um procedimento análogo ao resultado de 2-aproximação anterior, basta percorrer  $J$  seguindo a trilha Euleriana utilizando *short cut* quando necessário para não repetir vértices. O emparelhamento perfeito de custo mínimo possui uma relação importante com o TSP descrita pelo Teorema 4.

**Teorema 4.** *Seja  $G$  um grafo completo e  $w: E(G) \rightarrow \mathbb{R}_{\geq 0}$  é uma função de custo nas arestas que respeita a desigualdade triangular. Seja  $X \subseteq V(G)$  de modo que  $|X|$  é par, e seja  $M$  um emparelhamento perfeito de custo mínimo em  $X$ . Então  $w(M) \leq OPT_{TSP}(G, w)/2$ .*

*Demonstração.* Considere um TSP ótimo  $S$  em  $G$ . Seja  $S'$  o ciclo apenas em  $X$ , obtido realizando *short cut* em  $S$ . Pela desigualdade triangular,  $w(S') \leq w(S)$ . Note que  $S'$  é a união de dois emparelhamentos perfeitos em  $X$ , cada um consistindo de arestas alternadas de  $S$ . Sendo assim, o emparelhamento que tem menor custo, desses dois, tem custo  $w \leq w(S')/2 \leq OPT_{TSP}(G, w)/2$ .  $\square$

Sendo assim, a trilha Euleriana proposta tem custo  $\mathcal{B}$  tal que

$$\mathcal{B} \leq OPT_{MST}(G, w) + w(M(G)) \leq OPT_{TSP}(G, w) + \frac{OPT_{TSP}(G, w)}{2} = \frac{3}{2} OPT_{TSP}(G, w), \quad (11)$$

e portanto é uma  $3/2$ -aproximação para o TSP métrico.

## 6 Algoritmos para o $\{1, 2\}$ -TSP

Seja  $G$  um grafo completo e  $w: E(G) \rightarrow \{1, 2\}$  uma função de custos nas arestas de  $G$ . Note que  $G$  respeita a desigualdade triangular. Portanto, os algoritmos vistos na Seção 5 garantem aqueles mesmos fatores de aproximação para o  $\{1, 2\}$ -TSP. Um limitante inferior interessante que se apresenta quando tratamos do  $\{1, 2\}$ -TSP é apresentado pela inequação (12), onde  $n$  representa o número de vértices. Essa relação se dá pelo fato de que, no melhor caso, o ciclo será composto apenas pelas arestas de custo 1:

$$OPT_{1,2\text{-TSP}}(G, w) \geq n. \quad (12)$$

Nas seções a seguir, serão apresentados alguns algoritmos que possuem fatores de aproximação muito bons para essa variação específica do TSP. Atualmente, a melhor aproximação que se tem para essa variação é uma  $8/7$ -aproximação [1].

### 6.1 Algoritmo de 2-aproximação para o $\{1, 2\}$ -TSP

Seja  $I = \langle G, w \rangle$  uma instância do  $\{1, 2\}$ -TSP. Esse algoritmo simplesmente devolve qualquer ciclo Hamiltoniano. Se  $n$  é o número de vértices, note que o custo do ciclo obtido pelo algoritmo é no máximo  $2n$ , já que cada aresta tem o custo 1 ou 2. Além disso, pela inequação (12), se  $\mathcal{A}$  é o custo do algoritmo, vale que

$$\mathcal{A} \leq 2n \leq 2 OPT_{\text{TSP}}(G, w), \quad (13)$$

de onde obtemos outra 2-aproximação, porém bem mais simples.

### 6.2 Algoritmo de $5/3$ -aproximação para o $\{1, 2\}$ -TSP

Seja  $I = \langle G, w \rangle$  uma instância do  $\{1, 2\}$ -TSP e seja  $n = |V(G)|$ . Para encontrar um algoritmo com fator de aproximação  $\alpha = 5/3$ , nos utilizaremos da desigualdade  $OPT_{1,2\text{-TSP}}(G, w) \geq OPT_{2\text{-fator}}(G, w)$ , conforme vimos na Equação (9). Certamente nosso grafo  $G$ , por ser completo, possui um 2-fator. O menor número de vértices necessários para formar um ciclo é 3, portanto qualquer 2-fator em  $G$  terá no máximo  $n/3$  ciclos. Esses ciclos são disjuntos e possuem arestas com custo 1 ou 2. Para encontrar um ciclo Hamiltoniano em  $G$  podemos unir tais ciclos, removendo uma aresta de cada ciclo, dois a dois, e unindo esses dois ciclos com outras arestas de custo 1 ou 2, conforme mostrado na Figura 3.

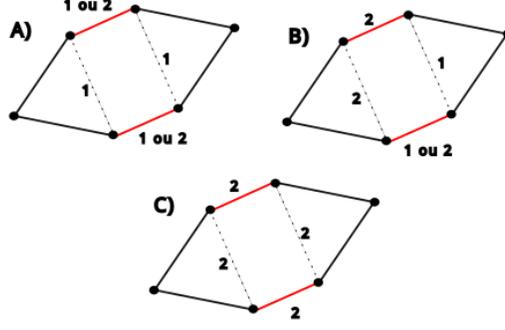


Figura 3: Diferentes possibilidades de unir ciclos de um 2-fator.

Note que ao unir dois ciclos, podemos ter um acréscimo do custo total a depender do caso. Seguindo a notação na Figura 3, caso a união ocorra como em “A”, o acréscimo será de +0, +1 ou +2; agora se ocorrer como em “B”, este poderá ser de +0 ou +1. E, por último, +0 no caso “C”. Seja  $U$  o custo total dos acréscimos após a união de todos os ciclos do 2-fator inicial. Se realizarmos esse procedimento para todos os ciclos, teremos que o custo final do ciclo Hamiltoniano encontrado será

$$S = OPT_{2\text{-fator}}(G, c) + U. \quad (14)$$

Substituindo pelo resultado da Equação (9),

$$OPT_{2\text{-fator}}(G, c) + U \leq OPT_{TSP}(G, w) + U. \quad (15)$$

Como o 2-fator tem no máximo  $n/3$  ciclos, significa que faremos ao todo no máximo  $n/3 - 1$  uniões para gerar o ciclo Hamiltoniano final. Unir ciclos aumenta o custo total do ciclo resultante em no máximo 2 no pior caso. Assim teremos:

$$\begin{aligned} S &\leq OPT_{1,2\text{-TSP}}(G, w) + U \\ &\leq OPT_{1,2\text{-TSP}}(G, w) + 2 \left( \frac{n}{3} - 1 \right) \\ &< OPT_{1,2\text{-TSP}}(G, w) + 2 \frac{n}{3} \\ &\leq OPT_{1,2\text{-TSP}}(G, w) + \frac{2}{3} OPT_{1,2\text{-TSP}}(G, c) \\ &= \frac{5}{3} OPT_{1,2\text{-TSP}}(G, w), \end{aligned} \quad (16)$$

onde usamos o resultado da Equação (12), o que resulta em uma  $5/3$ -aproximação para o  $\{1, 2\}$ -TSP.

### 6.3 Algoritmo de 4/3-aproximação para o $\{1, 2\}$ -TSP

Seja  $I = \langle G, w \rangle$  uma instância do  $\{1, 2\}$ -TSP. Observando com atenção, pode-se perceber que o algoritmo de 5/3-aproximação une ciclos sem critério nenhum. É possível utilizar abordagens mais inteligentes para essas uniões. Perceba que unir os ciclos dois a dois pode acrescer o custo total de diferentes formas, a depender de quais arestas escolhermos substituir em cada união. Se substituirmos duas arestas de custo 1, estaremos sempre adicionando no máximo 2 ao custo total. Note ainda que, sempre que há aresta de custo 2 presente em algum dos ciclos que pretendemos unir, o custo total só poderá ser acrescido em no máximo 1 pois, no pior caso, podemos escolher trocar duas arestas cujo custo total é 3, por duas arestas cujo custo total será 4. Dessa forma, começamos com os ciclos do 2-fator e os juntamos, um após o outro. Perceba que podemos começar adicionando 2 ao custo total, porém a partir desse momento, haverá uma aresta de custo 2 no ciclo resultante e, portanto, estaremos adicionando no máximo 1 ao custo a partir daí. Outra possibilidade é que, sempre que houver um acréscimo de 2 ao custo, esse acréscimo deverá ter sido precedido por um acréscimo de 0 ao custo. Então em média, o acréscimo será sempre de 1. Sendo assim, se começarmos unindo os  $n/3$  ciclos, preferencialmente pelas arestas de custo 2 presentes nos ciclos, teremos

$$OPT_{2\text{-fator}}(G, w) + \frac{n}{3} \leq \frac{3}{3} OPT_{1,2\text{-TSP}}(G, w) + \frac{1}{3} OPT_{1,2\text{-TSP}}(G, w) = \frac{4}{3} OPT_{1,2\text{-TSP}}, \quad (17)$$

sendo, assim uma 4/3-aproximação para o  $\{1, 2\}$ -TSP.

### 6.4 Algoritmo de 5/4-aproximação para o $\{1, 2\}$ -TSP

Seja  $I = \langle G, w \rangle$  uma instância do  $\{1, 2\}$ -TSP. A demonstração de um algoritmo cujo fator de aproximação é 5/4 é muito semelhante à apresentada na Seção 6.2. Novamente, utilizando da desigualdade  $OPT_{\text{TSP}}(G, w) \geq OPT_{2\text{-fator}}(G, w)$ , que apresenta uma relação limitante inferior entre o TSP e o 2-fator (Equação (9)), tome agora, porém, um conjunto formado por ciclos de tamanho pelo menos 4. Sendo assim, o tamanho máximo desse conjunto será  $n/4$ . Esses ciclos são disjuntos e possuem aresta de custo 1 ou 2. Para encontrar um ciclo Hamiltoniano a partir desses ciclos, basta uní-los utilizando o mesmo procedimento apresentado nas sessões anteriores.

Mais uma vez, a cada união, aumentamos o custo do ciclo em até 2. Seja  $U$  o custo total dos acréscimos após a união de todos os ciclos. Se realizarmos esse procedimento para todos os ciclos, teremos que o custo final do ciclo Hamiltoniano encontrado será

$$S = OPT_{2\text{-fator}}(G, w) + U. \quad (18)$$

Substituindo pelo resultado da Equação (9),

$$OPT_{2\text{-fator}}(G, w) + U \leq OPT_{1,2\text{-TSP}}(G, w) + U. \quad (19)$$

Como se trata de no máximo  $n/4$  ciclos, significa que faremos ao todo no máximo  $n/4 - 1$  uniões para gerar o ciclo Hamiltoniano final. Assim, ao final, teremos

$$\begin{aligned} S &\leq OPT_{1,2\text{-TSP}}(G, w) + U \\ &\leq OPT_{1,2\text{-TSP}}(G, w) + \frac{OPT_{1,2\text{-TSP}}(G, w)}{4} \\ &= \frac{5}{4}OPT_{1,2\text{-TSP}}(G, w) \end{aligned}$$

onde, utilizando o resultado da Equação (12), concluímos que o algoritmo é uma  $5/4$ -aproximação para o  $\{1, 2\}$ -TSP.

## 7 Conclusão

O TSP é um problema que possui complexidade enorme, mas também inúmeras aplicações no nosso dia a dia, o que contribui para sua fama e atração por inúmeros pesquisadores e pesquisadoras na área de estudo da computação. Os principais resultados que se destacam para o TSP Métrico são o do fator de inaproximabilidade  $123/122$  de Karpinski [8] e a  $(3/2 - \epsilon)$ -aproximação de Karlin [6] em 2021. Já no  $\{1, 2\}$ -TSP, por se tratar de uma variação mais específica, encontramos fatores de aproximação ainda menores tanto em sua inaproximabilidade, em que foi apresentado um fator de aproximação mínimo de  $535/534$  [7], e na  $8/7$ -aproximação proposta por Adamaszek [1] em 2018.

Os algoritmos apresentados até hoje possuem estratégias muito interessantes e elegantes para tentar solucionar instâncias restritas desse problema. Um possível trabalho futuro seria aprofundar mais a pesquisa desses algoritmos e técnicas, abordando tanto o algoritmo de melhor fator de aproximação apresentado até hoje para o TSP métrico [6] como as estratégias probabilísticas.

A proposta inicial desse relatório era aprofundar os estudos até o algoritmo que apresenta uma  $11/9$ -aproximação para o  $\{1, 2\}$ -TSP [9], o que acabou não sendo possível. Porém, os estudos dos outros algoritmos apresentados aqui tiveram grande contribuição para os estudos do aluno área de algoritmos de aproximação e aumentaram o interesse em aprofundar os estudos em outros problemas dessa área.

## Referências

- [1] A. Adamaszek, M. Mnich, and K. Paluch. New approximation algorithms for  $(1, 2)$ -TSP. In *45th International Colloquium on Automata, Languages, and Programming*, volume 107 of *LIPIcs. Leibniz Int. Proc. Inform.*, page 14. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2018. doi:[10.4230/LIPIcs.ICALP.2018.9](https://doi.org/10.4230/LIPIcs.ICALP.2018.9).
- [2] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton Series in Applied Mathematics. Princeton University Press, 2006. ISBN 9780691129938.
- [3] J. Bondy and U. Murty. *Graph Theory*. Springer Publishing Company, Incorporated, 1st edition, 2008. ISBN 1846289696.
- [4] M. Carvalho, M. Cerioli, R. Dahab, P. Feofiloff, C. Fernandes, C. Ferreira, K. Guimaraes, F. Miyazawa, J. Pina Jr, J. Soares, et al. Uma introdução sucinta a algoritmos de aproximação. *Publicações Matemáticas do IMPA*, 2001.
- [5] N. Christofides. Worst-Case Analysis of a New Heuristic for the Traveling Salesman Problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.
- [6] A. R. Karlin, N. Klein, and S. O. Gharan. A Deterministic Better-than- $3/2$  Approximation Algorithm for Metric TSP. In A. Del Pia and V. Kaibel, editors, *Integer Programming and Combinatorial Optimization*, pages 261–274, Cham, 2023. Springer International Publishing. doi:[10.1007/978-3-031-32726-1\\_19](https://doi.org/10.1007/978-3-031-32726-1_19).
- [7] M. Karpinski and R. Schmied. On Approximation Lower Bounds for TSP with Bounded Metrics, 2012.
- [8] M. Karpinski, M. Lampis, and R. Schmied. New inapproximability bounds for TSP. *Journal of Computer and System Sciences*, 81(8):1665–1677, 2015. doi:[10.1016/j.jcss.2015.06.003](https://doi.org/10.1016/j.jcss.2015.06.003).
- [9] C. H. Papadimitriou and M. Yannakakis. The traveling salesman problem with distances one and two. *Mathematics of Operations Research*, 18(1):1–11, 1993. ISSN 0364-765X. doi:[10.1287/moor.18.1.1](https://doi.org/10.1287/moor.18.1.1).
- [10] E. Pegg Jr. The icosian game, revisited. *The Mathematica Journal*, 11(3):310–314, 2009.
- [11] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, Heidelberg, 2001.

# A Algoritmos para o problema da árvore de Steiner

Nesse apêndice será apresentado um algoritmo para outro problema que se relaciona com o TSP, o problema da árvore de Steiner. Por possuir uma estratégia muito semelhante à abordada na Sessão 7 também proposta por [11], seu estudo teve papel complementar.

O problema da Árvore de Steiner é bastante importante no estudo de algoritmos de aproximação, possuindo uma vasta gama de aplicações. Ele consiste, de forma geral, em encontrar a menor interconexão para um determinado conjunto de objetos. Segue sua descrição formal:

**Problema 5** (Árvore Steiner). *Dado um grafo  $G$  sem direções com arestas de custo não negativos, cujo vértices são separados em dois subconjuntos, requeridos e Steiner, ache uma árvore de custo mínimo no grafo  $G$  que contenha todos os vértices requeridos e qualquer subconjunto dos vértices Steiner.*

## A.1 Relação do problema métrico com o problema geral

A solução para esse problema pode ser encontrada analisando o seu caso métrico (Problema da Árvore Steiner Métrica), ou seja, nas instâncias que possuem a propriedade da **desigualdade triangular**. Essa restrição nos garante mais informações sobre o problema, o que proporciona uma análise melhor. O Teorema 1 garante redução do problema geral para o caso métrico, preservando o fator de aproximação.

**Teorema 5.** *Existe uma redução do problema da Árvore Steiner para o problema da Árvore Steiner Métrica que preserva o fator de aproximação.*

Seja  $G$  um grafo sem direções,  $I$  uma instância do problema da Árvore Steiner,  $w$  o custo das arestas de  $G$  e  $R$  o conjunto de vértices requeridos. Precisamos primeiramente transformar uma instância  $I = (G, w, R)$  em uma instância da Árvore Steiner Métrica  $I' = (G', w', R')$ , em que  $G'$  é um grafo completo e sem direções. Além disso, a divisão de  $V(G)$  em *Requeridos* e *Steiner* é igual em  $I$  e  $I'$  (isto é,  $R' = R$ ). Para isso, o grafo  $G'$  será construído de forma que  $w'(uv)$  em  $G'$  seja definido como o custo do menor  $u - v$  caminho em  $G$ .  $G'$  é, portanto, chamado de *metric closure* (colocar obs) de  $G$ . Essa regra de formação de  $G'$  faz com que qualquer aresta  $uv \in E(G')$  não tenha custo maior que seu custo em  $G$  e, com isso, o custo de qualquer solução ótima em  $I'$  é menor ou igual ao custo de uma solução ótima em  $I$ .

Seja  $T'$  uma Árvore Steiner em  $G'$ . Uma aresta em  $G'$  tem o custo de um caminho de menor custo em  $G$ . Substitua cada aresta  $uv \in T'$  pelo seu caminho correspondente e teremos um subgrafo de  $G$ . Como  $T'$  é uma Árvore Steiner, o subgrafo correspondente em  $G$  também possui todos os vértices *Requeridos* conectados. Caso esse subgrafo possua ciclos, basta remover uma aresta de cada ciclo e com isso teremos uma Árvore Steiner  $T$  em  $G$ .

Esse método permite que se gere  $T$  em  $I$  à partir de uma  $T'$  em  $I'$  em tempo polinomial. Como o custo de qualquer solução em  $I'$  não é maior que o de uma em  $I$ , fica provada a redução.

Sendo assim, a partir do teorema, segue que qualquer fator de aproximação obtido para o problema métrico pode ser utilizado diretamente no caso geral. Portanto os algoritmos que seguem utilizarão apenas instâncias da Árvore Steiner Métrica.

## A.2 Algoritmo de 2-aproximação

A seguir será apresentado um algoritmo de aproximação para encontrar uma árvore de Steiner que utiliza uma árvore geradora mínima de um grafo. Seja  $G, c, R$  uma instância do problema da Árvore de Steiner Métrica. É evidente que uma MST do subgrafo composto pelos vértices *Requeridos*  $R$  do grafo  $G$  é uma árvore de Steiner no grafo, pela própria definição da MST. Uma árvore geradora mínima em  $G[R]$  pode ser encontrada em tempo polinomial, porém esta nem sempre será a solução ótima para a Árvore Steiner. No teorema a seguir, provaremos que ela tem, porém, custo até  $2 OPT_{\text{steiner}}(G, c, R)$ .

**Teorema 6.** *Sejam  $G$  um grafo completo,  $c: E(G) \rightarrow \mathbb{R}^+$  uma função de custos nas arestas que respeita a desigualdade triangular e  $R \subseteq V(G)$  um subconjunto de vértices. O custo de uma árvore geradora mínima em  $G[R]$  é de até  $2 OPT_{\text{steiner}}(G, c, R)$ .*

*Demonstração.* Dados  $G$  e  $R$ , seja  $T^*$  uma árvore de Steiner de custo ótimo  $OPT_{\text{steiner}}(G, c, R)$ . Note que  $T^*$  conecta todos os vértices *Requeridos* e, possivelmente, alguns *Steiner*. Ao dobrar as arestas de  $T^*$ , obteremos um grafo euleriano (todos os vértices de  $\mathcal{E}$  têm grau par). Seja  $H$  esse grafo. Agora ache uma trilha euleriana  $\mathcal{E}$  (que passa por todas as arestas) em  $H$ . Como  $H$  tem o dobro de arestas de  $T^*$ , o custo dessa trilha é  $2 OPT_{\text{steiner}}(G, c, R)$ . Construa agora um ciclo Hamiltoniano (todos os vértices são visitados) a partir de  $\mathcal{E}$ , pulando vértices *Steiner* com a técnica de *short cut* quando necessário. Como  $c$  satisfaz a desigualdade triangular, utilizar *short cuts* não aumenta o custo do ciclo Hamiltoniano. Retire agora uma aresta desse ciclo, e teremos um caminho que conecta todos os vértices de  $R$  com custo no máximo  $2 OPT_{\text{steiner}}(G, c, R)$ . Esse caminho é uma árvore geradora em  $G[R]$ , portanto uma árvore geradora mínima em  $G[R]$  só poderá ter custo no máximo  $2 OPT_{\text{steiner}}(G, c, R)$ .  $\square$