



ALGORITMIA E ESTRUTURA DE DADOS

---

# Travel Salesman Problem

---

*Discente(s):*

Pedro ROLDAN

Leandro MOREIRA

*Docente:*

Doutor Faroq ALTAM

27 de Junho de 2017

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
1.1	Requisitos Minimos . . . . .	2
<b>2</b>	<b>2-Opt</b>	<b>3</b>
2.1	Técnica 2-Opt . . . . .	4
<b>3</b>	<b>Enquadramento</b>	<b>13</b>
3.1	Motivação . . . . .	13
3.2	Objectivos . . . . .	13
<b>4</b>	<b>Conclusões</b>	<b>13</b>
<b>5</b>	<b>Bibliografia</b>	<b>14</b>
<b>6</b>	<b>Anexos</b>	<b>15</b>

# 1 Introdução

O travel salesman problem (TSP) é um problema bastante comum largamente encontrado em diversas aplicações tais como: empresas de transporte (e.g. UPS), escalas de tripulação de companhias aéreas, etc.

Em princípio, um vendedor necessita de efetuar uma viagem por diversas cidades, onde inicia a viagem numa determinada cidade (Casa), visita todas as cidades para vender os seus produtos, e retorna a casa.

## 1.1 Requisitos Minimos

O TSP pode ser representado por uma lista de nós, sendo o objectivo descobrir uma serie de caminhos (Edges) entre cada um dos nós.

Sendo que:

- Cada nó (Cidade) pode ser visitado apenas 1 vez.
- Os caminhos formam uma Tour.
- O custo da Tour deve ser o minimo possivel.

A tour TSP é um grafico direcionado, onde cada nó representa uma cidade, e cada edge representa um caminho entre 2 cidades.

Cada edge têm um peso, que é no seu caso mais simples a distancia euclidiana entre os seus nós.

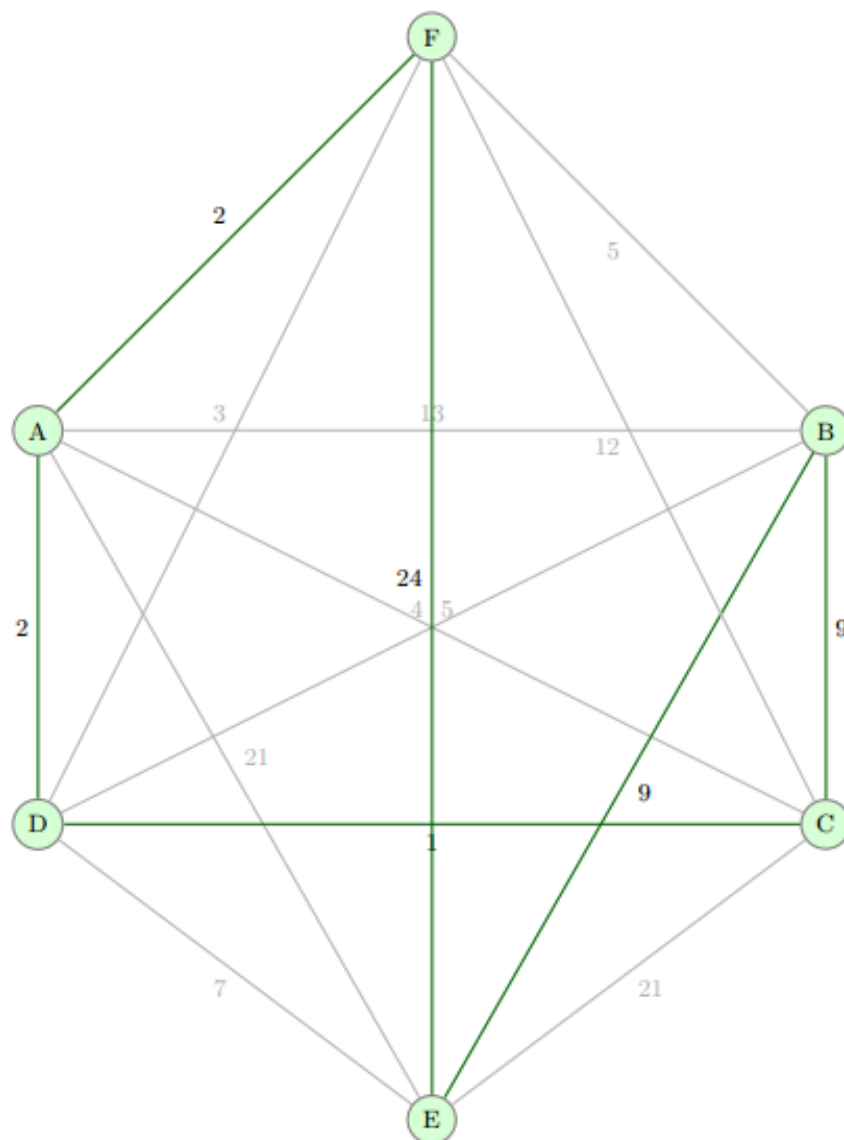
Este peso pode ser composto por diversos fatores, no entanto neste projeto apenas se considera a distancia entre nós.

## 2 2-Opt

O algoritmo 2-Opt foi proposto por Croes em 1958, embora o movimento basico tenha sido sugerido por Flood em 1956.

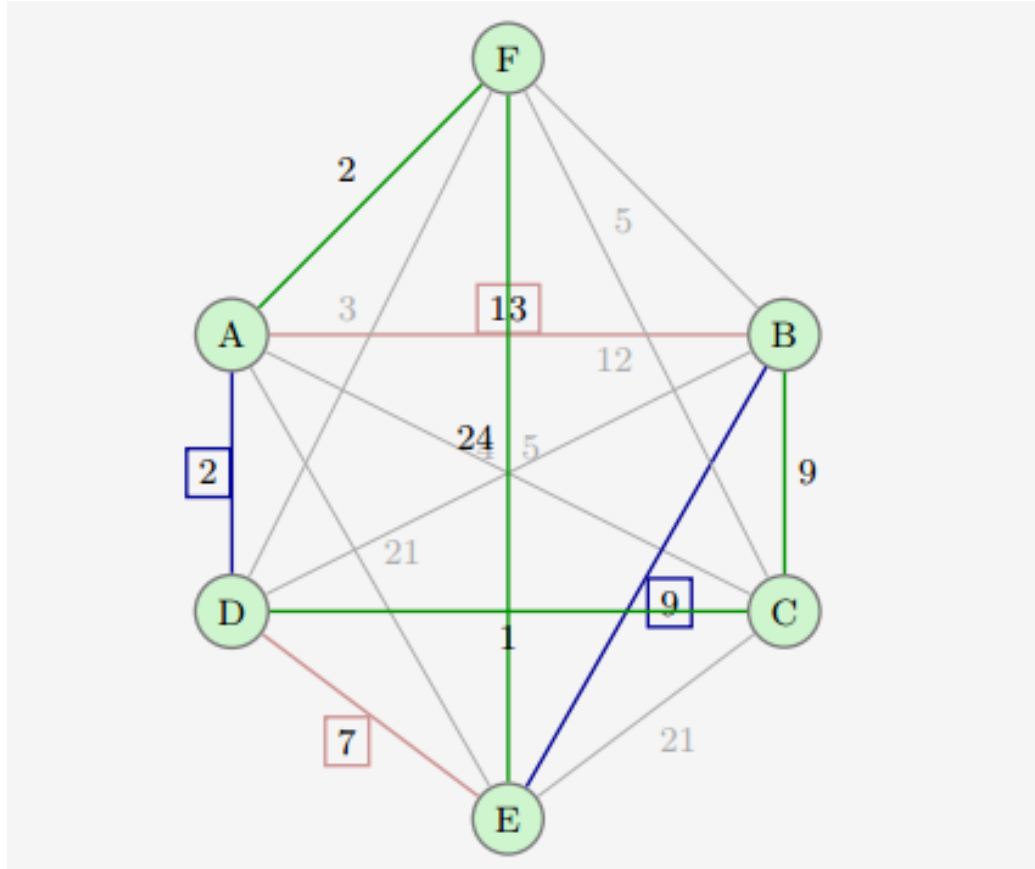
O algoritmo 2-Opt basicamente remove 2 nós (Edges) da tour, e reconnecta os dois caminhos criados. Isto é normalmente referido como um movimento 2-Opt.

## 2.1 Técnica 2-Opt



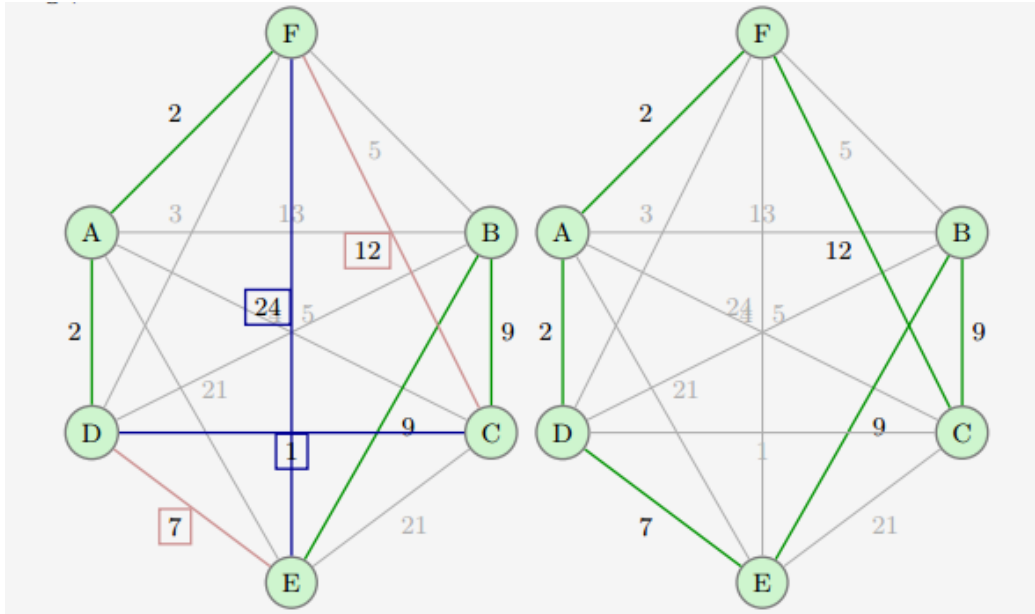
Aqui vamos utilizar uma lista de 6 nós, em que a distancia total da tour é de 47.

Em primeiro lugar selecionamos um edge, neste caso **AD**, vamos de seguida selecionar outro edge que não seja adjacente ao edge em uso, **AD**. Neste exemplo apenas dispomos de um edge possível **BE**. Podemos assim trocar **AD** e **BE** por **AB** e **BE**.



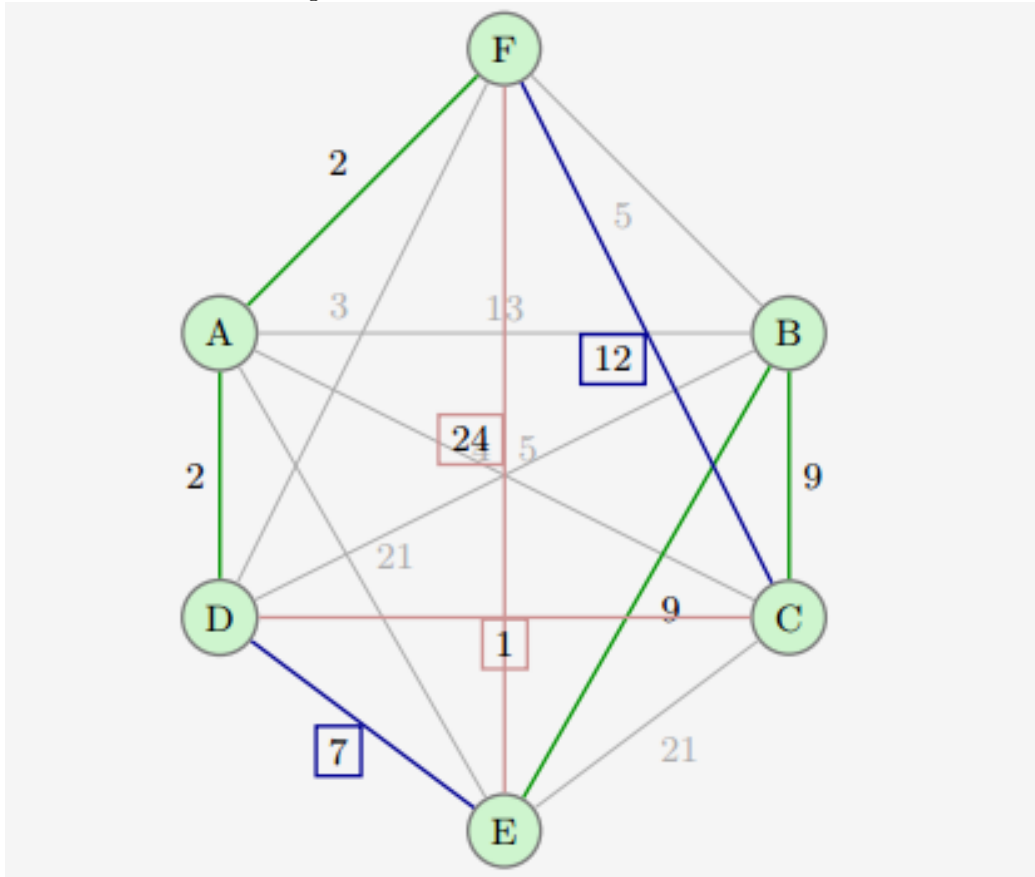
No entanto, a soma das distancias dos edges possiveis de trocar é **maior** que a soma das distancias dos edges originais. Assim sendo não trocamos os edges.

Considerando o proximo edge **DC**, apenas dispoe de 1 edge não-adjacente,



Aqui, como a soma das distancias dos edges originais é **maior** que a soma das distancias dos edges possiveis de trocar, efetuamos a troca de edges.

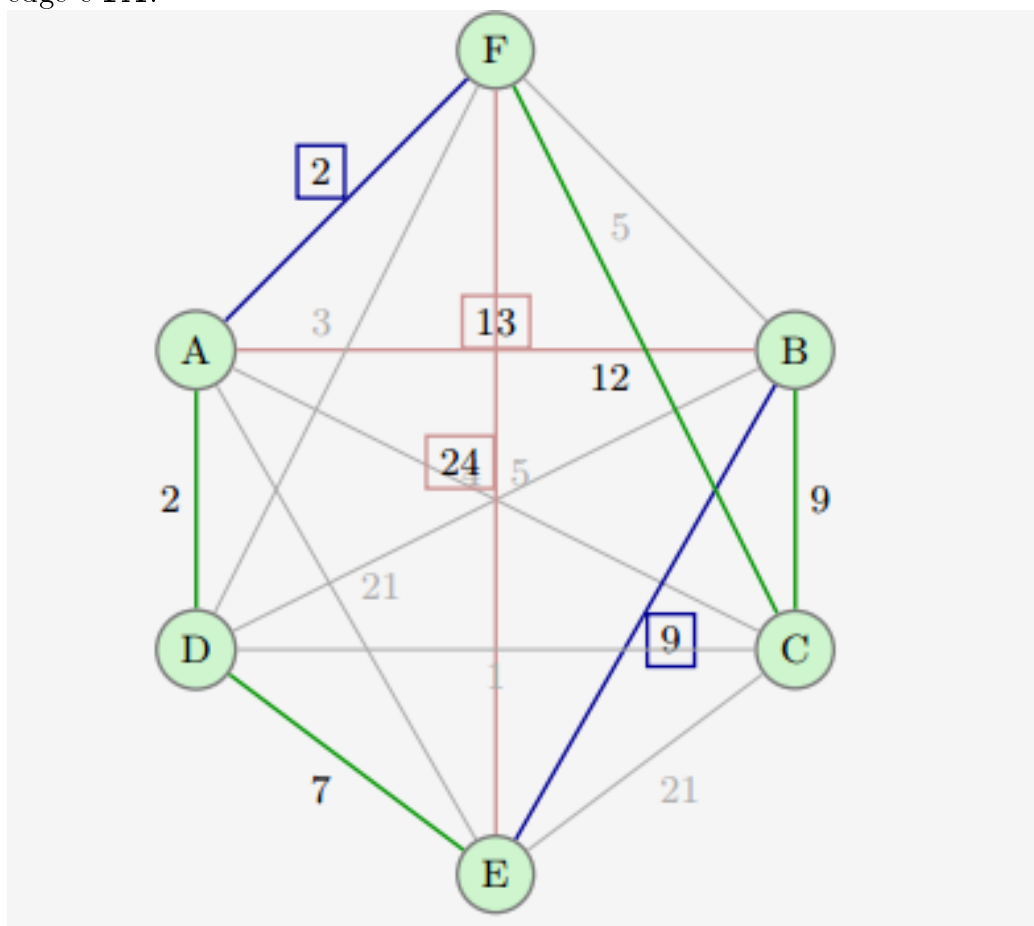
Considerando o edge **DE**, este edge dispõe do edge não-adjacente **CF**.



Como a soma das distancias dos edges possiveis de trocar são maiores que a soma das distancias dos edges originais, não se efetua trocas.

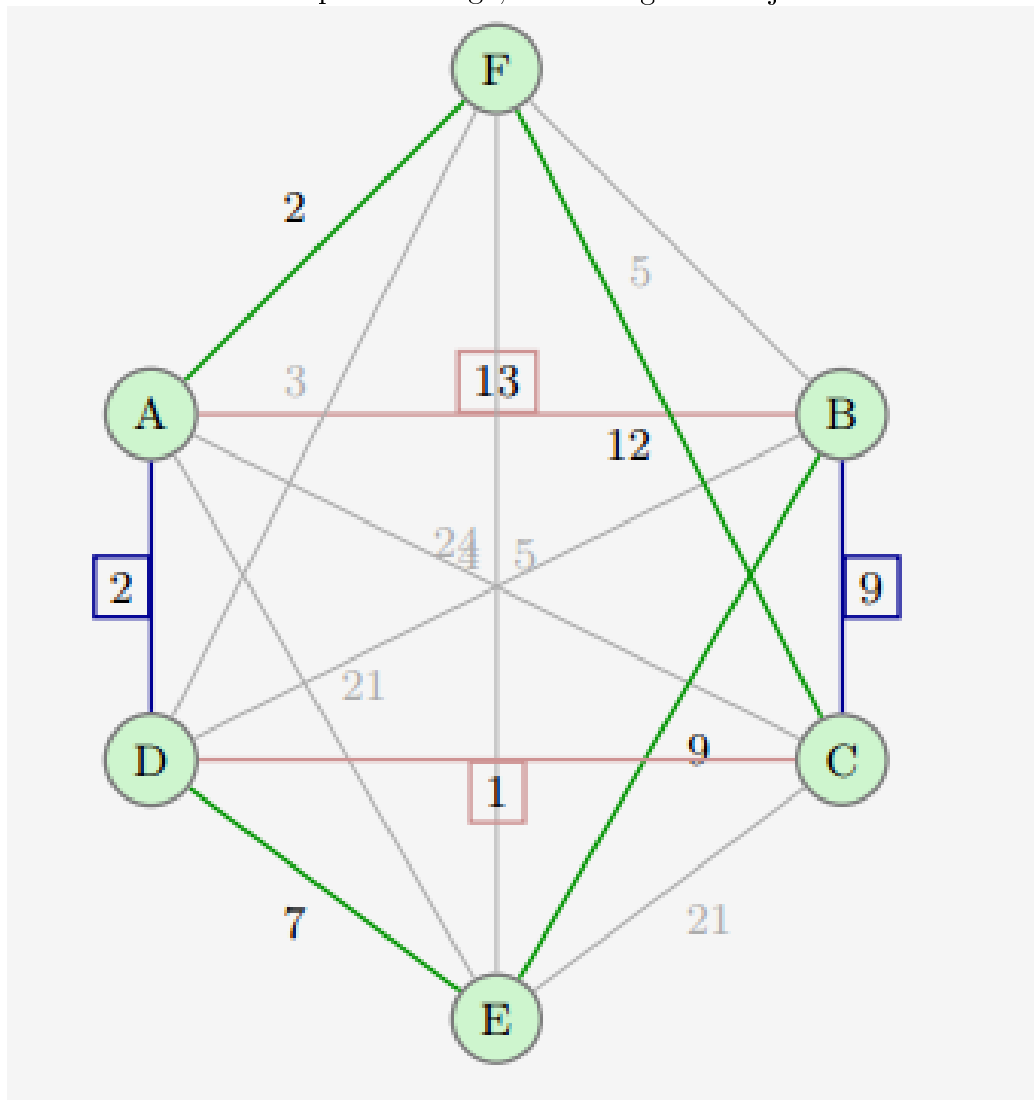


No proximo passo o edge selecionado é **EB**. O edge não-adjacente a este edge é **FA**.



Não é necessário efetuar trocas de edges.

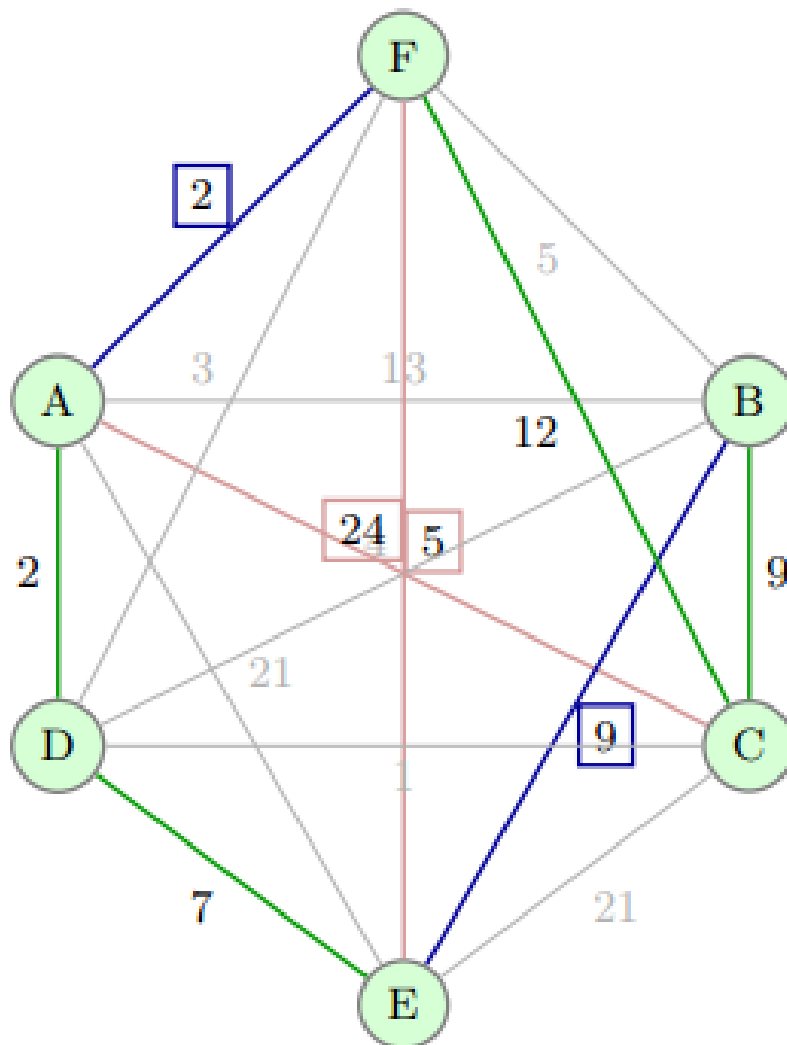
Vamos seleccionar o proximo edge, **BC**. O edge não-adjacente a este é **AD**.



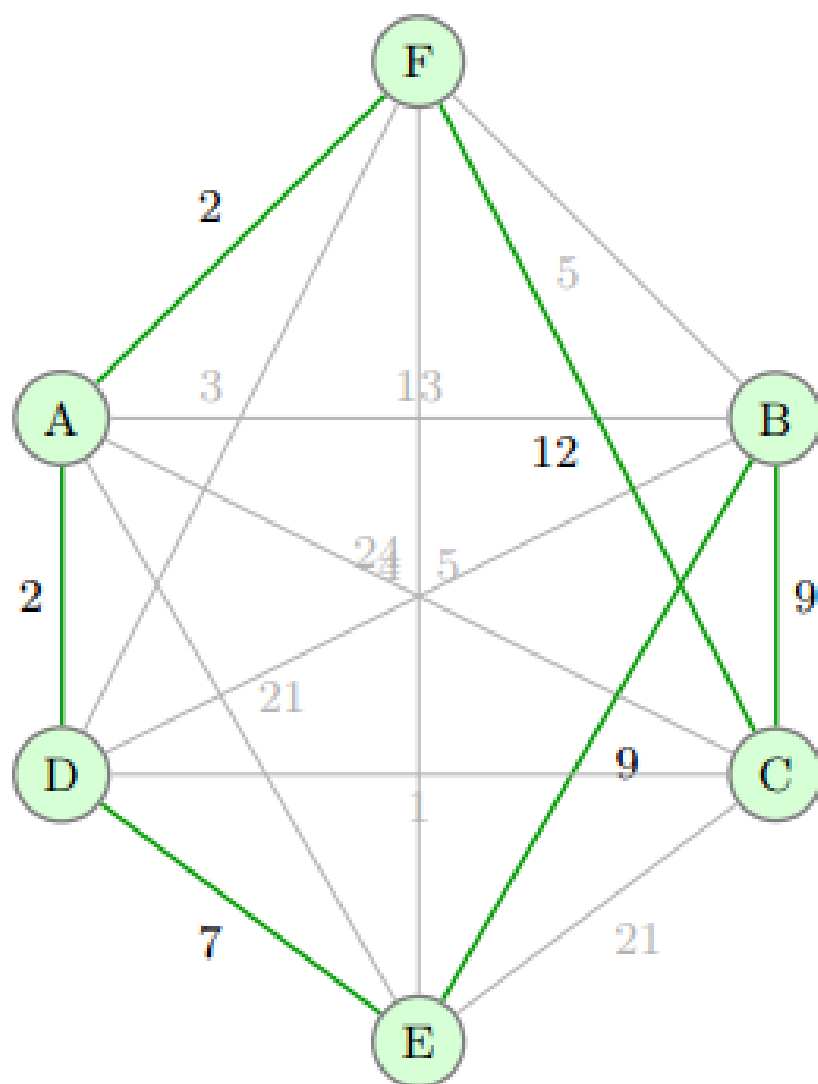
Este cenário não garante alterações ao caminho.

Página 10 de 15

O proximo edge a ser considerado é **FA**. O edge não adjacente é **EB**.



Este também é um caso em que não existe necessidade de se efetuar trocas de edges.



Finalmente chegamos a uma solução otimizada. A solução apresenta um custo de 41, efetivamente inferior ao custo da solução inicial.

## 3 Enquadramento

O trabalho descrito neste relatório foi realizado recorrendo à linguagem de programação ANSI C, assim como os recursos disponibilizados na unidade curricular.

### 3.1 Motivação

A principal motivação para a realização deste trabalho, resulta da importância em implementar o algoritmo 2-Opt ao problema Travel Salesman, assim como demonstrar os conhecimentos alcançados na disciplina de Algoritmia e Estrutura de Dados.

### 3.2 Objectivos

Pretende-se através deste trabalho, atingir uma solução válida para o Travel Salesman Problem, implementando o algoritmo 2-Opt.

## 4 Conclusões

A implementação do algoritmo 2-Opt ao TSP desenvolvida, para além de permitir os requisitos pedidos no enunciado do trabalho prático, permite também a possível implementação de outros algoritmos, pois sendo modular torna-se mais escalável, entre outras funcionalidades.

De frisar que devido à liberdade proporcionada, quer na sua forma de desenvolvimento quer na implementação permitiu desta forma aguçar a curiosidade para o uso de diversos algoritmos para a solução do Travel Salesman Problem.

Foi sem dúvida um desafio interessante, mas que por limitação de tempo, deixa ainda uma larga margem para melhoramentos.

## 5 Bibliografia

### Referências

- [1] <https://en.wikipedia.org/wiki/2-opt>.
- [2] [https://en.wikipedia.org/wiki/travelling salesman problem](https://en.wikipedia.org/wiki/travelling_salesman_problem).
- [3] Doutor Faroq AlTam. *Aulas Teórico-Práticas Algoritmia e Estruturas de Dados 2º ano, 2º semestre da Licenciatura em Engenharia Informática do Instituto Superior Manuel Teixeira Gomes*. ISMAT, 2016-2017.

## **6 Anexos**

Ficheiro "relatorio.pdf" e "ledmatrix.circ" compactado num ficheiro "trabalho.zip".

Não existem quaisquer códigos ou listagens adicionais.