**Relatório Algoritmo Genético**

***Caixeiro viajante***

**Equipe:**

Nome: Lucas Massao RA: 12345

Nome: Eduardo H. K. Shibukawa RA: 62003

Sumário

[1. Introdução 2](#_Toc458608830)

[2. Descrição do problema 2](#_Toc458608831)

[3. Descrição do algoritmo 2](#_Toc458608832)

[4. Resultados 2](#_Toc458608833)

[5. Conclusões 2](#_Toc458608834)

# Introdução

## Algoritmo genético

Algoritmo genético (AG) é um algoritmo heurístico, probabilístico, propostos pelo Professor John Holland (1975) da universidade de Michigan, inspirado na teoria da evolução natural das espécies.

AG é um procedimento iterativo que mantém uma população de estruturas também chamadas de cromossomos ou indivíduos, que representam possíveis soluções para um determinado problema, formadas por genes que são elementos do conjunto de solução, entre os fatores que tem feito o AG uma técnica bem-sucedida, destacam-se a simplicidade e a facilidade de se resolver um problema sem exigir muito conhecimento prévio deste.

## Caixeiro Viajante

Caixeiro viajante ou Traveling Salesman Problem (TSP), é um problema NP (Não Polinomial) no qual um caixeiro deve percorrer um conjunto de “n “ cidades e voltar a sua cidade de origem, passando uma única vez em cada cidade, de modo que a distância total percorrida seja mínima.

# Descrição do problema

Como citado na introdução o problema do TSP, é um NP no qual se deve percorrer todas cidades, voltando a sua origem, passando uma única vez em cada cidade, de modo que a distância total percorrida seja mínima.

Para se ter uma ideia da dificuldade deste problema, o tempo esperado para se resolver este algoritmo com um total de 25 cidades é de mais de 20 vezes a idade do universo, supondo que seja possível calcular 10.000 caminhos por segundo.

# Descrição do algoritmo

O algoritmo utilizado para a resolução do problema é o Algoritmo genético, utilizamos pela simplicidade e facilidade para resolver um problema sem conhecimento prévio do mesmo.

## Estrutura

**Cidades:** São os vértices do nosso problema, ela é composta pelo id da cidade, e as coordenadas X e Y da cidade.

**Indivíduo:** Representa uma solução válida, onde sua estrutura é composta por uma aptidão (Fitness) e de genes, onde o gene é a representação de do id da cidade, e os genes são uma lista de gene que representam um caminho, e a aptidão representa a distância total percorrida para este caminho.

**População:** Representa um conjunto de soluções, composta por vários indivíduos.

## Funcionamento

O algoritmo irá gerar uma população inicial, aletoriamente

## Busca Local

### First Fit

### Hill Climbing

First Fit, é mais rápido para chegar a um resultado, porém tem resultados finais piores

hillClimbing, tem menor velocidade para chegar a um resultado melhor, porém tem resultados finais melhores.

## Gráficos

### Att48

## Tabelas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caso | *MS* | *Alg* | *Alg\_bl* | *GAP*% |
| att48 | 10.628 |  |  |  |
| att535 | 202.339 |  |  |  |
| d657 | 48.912 |  |  |  |

Legenda:

* Caso: colocar o nome do caso, exemplo:
* *MS*: Melhor solução conhecida da literatura.
* *Alg*: Melhor solução obtida com o seu algoritmo.
* *Alg\_bl*: Melhor solução obtida com a combinação do seu algoritmo com busca local.
* *GAP*%: é um parâmetro de desvio relativo calculado da seguinte forma: 

# Resultados

# Conclusões