

Documentação do Trabalho de MOA

Ant System e Simulated Annealing

Alunos: Kamylla Felipin	112676
Kananda da Silva	116382
Leonardo de Souza	98995

Professor: Igor da Penha Natal

1. Ant System

Para o desenvolvimento do Ant System, utilizamos o problema do caixeiro viajante e N Rainhas.

1.1. Caixeiro Viajante:

Configuração 1:

- Alpha: 0.7
- Beta: 0.7
- Evaporation Rate: 0.5
- Máximo de ciclos: 100

1º Execução:

7301.0

[(10, 8), (8, 9), (9, 6), (6, 5), (5, 14), (14, 13), (13, 12), (12, 11), (11, 15), (15, 0), (0, 7), (7, 3), (3, 1), (1, 2), (2, 4), (4, 10)].

2º Ciclo.

2º Execução:

7845.0

[(2, 1), (1, 3), (3, 7), (7, 0), (0, 15), (15, 12), (12, 11), (11, 14), (14, 4), (4, 9), (9, 8), (8, 10), (10, 6), (6, 5), (5, 13), (13, 2)]

3º ciclo

3º execução:

7301.0

[(8, 9), (9, 6), (6, 5), (5, 14), (14, 13), (13, 12), (12, 11), (11, 15), (15, 0), (0, 7), (7, 3), (3, 1), (1, 2), (2, 4), (4, 10), (10, 8)]

2º ciclo

Configuração 2:

- alpha: 0.6
- beta: 0.8
- evaporation rate: 0.5

- máximo de ciclos: 100

1º execução:

7301.0

[(10, 8), (8, 9), (9, 6), (6, 5), (5, 14), (14, 13), (13, 12), (12, 11), (11, 15), (15, 0), (0, 7), (7, 3), (3, 1), (1, 2), (2, 4), (4, 10)]

3º ciclo

2º Execução:

7301.0

[(10, 8), (8, 9), (9, 6), (6, 5), (5, 14), (14, 13), (13, 12), (12, 11), (11, 15), (15, 0), (0, 7), (7, 3), (3, 1), (1, 2), (2, 4), (4, 10)]

2º ciclo

3º execução:

7943.0

[(2, 1), (1, 3), (3, 7), (7, 0), (0, 15), (15, 12), (12, 13), (13, 11), (11, 6), (6, 5), (5, 14), (14, 4), (4, 9), (9, 8), (8, 10), (10, 2)]

1º ciclo

1.2. N Rainhas:

Configuração 1:

- alpha: 2
- beta: 2
- rainhas: 5

1º execução:

gerações executadas: 94

Ants = [5, 3, 1, 4, 2], fitness = 10

vezes encontradas: 5

2º execução:

gerações executadas: 190

Ants = [1, 3, 5, 2, 4], fitness = 10

vezes encontradas: 1

3º execução:

gerações executadas: 117

Ants = [3, 5, 2, 4, 1], fitness = 10

vezes encontradas: 3

Configuração 2:

- alpha: 3
- beta: 4

- rainhas: 5

1º execução:

gerações executadas: 32

Ants = [4, 1, 3, 5, 2], fitness = 10

vezes encontradas: 1

2º execução:

gerações executadas: 46

Ants = [5, 2, 4, 1, 3], fitness = 10

vezes encontradas: 3

3º execução:

gerações executadas: 45

Ants = [3, 1, 4, 2, 5], fitness = 10

vezes encontradas: 1

2. Simulated Annealing

Para o Simulated Annealing, utilizamos o problema do caixeiro viajante.

Configuração 1:

- alpha: 0.995
- cooling_schedule: slow
- operador: inversão
- stop: 100
- t_max: 10
- t_min: 0.0005

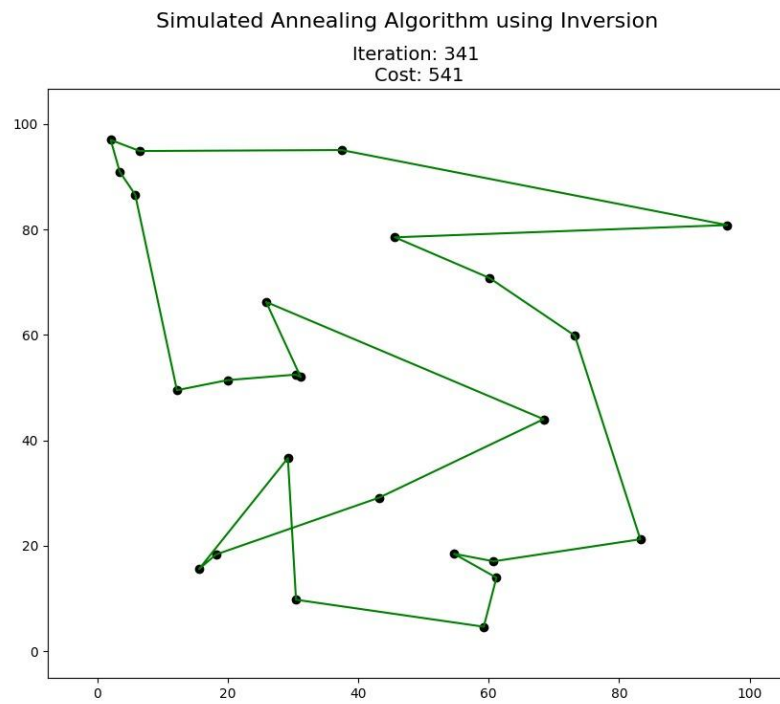


Imagem 1: Configuração 01 - Execução 01

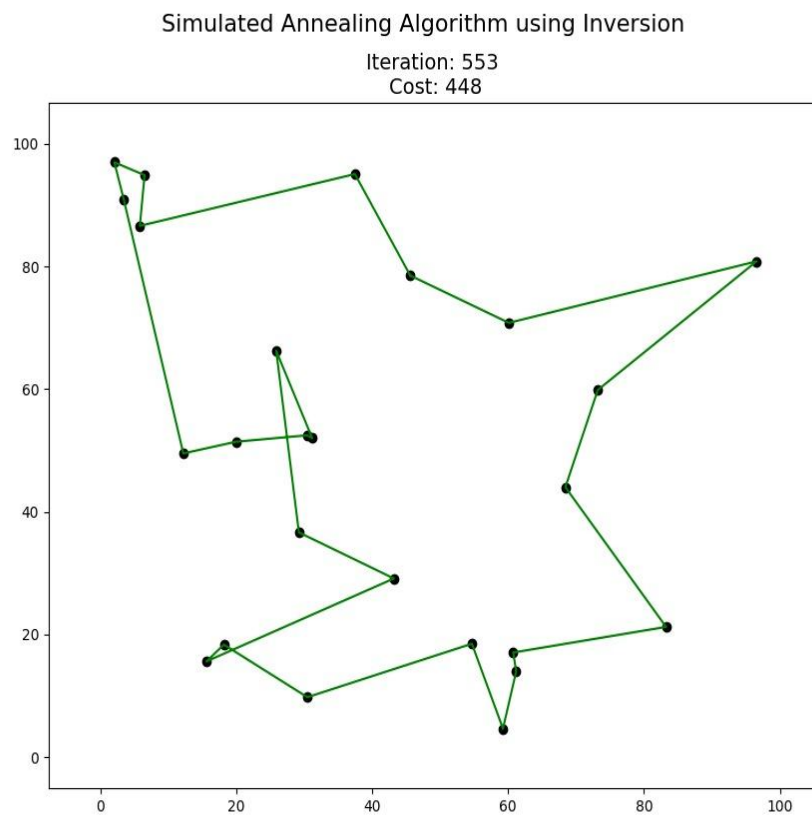


Imagem 2: Configuração 01 - Execução 02

Simulated Annealing Algorithm using Inversion

Iteration: 553

Cost: 448

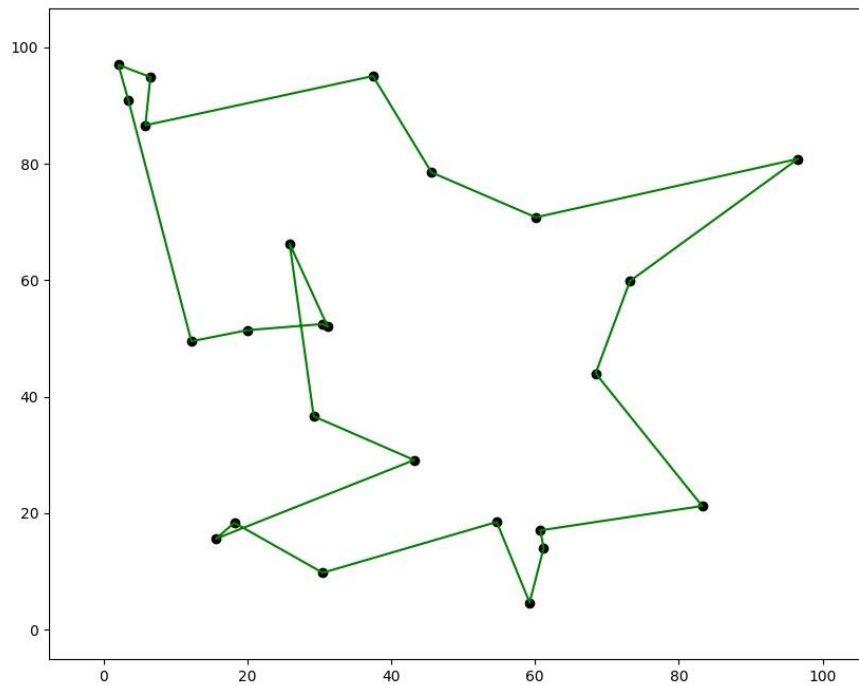


Imagem 3: Configuração 01 - Execução 03

Configuração 2:

- alpha: 0.905
- cooling_schedule: slow
- operador: inversão
- stop: 100
- t_max: 10
- t_min: 0.0002

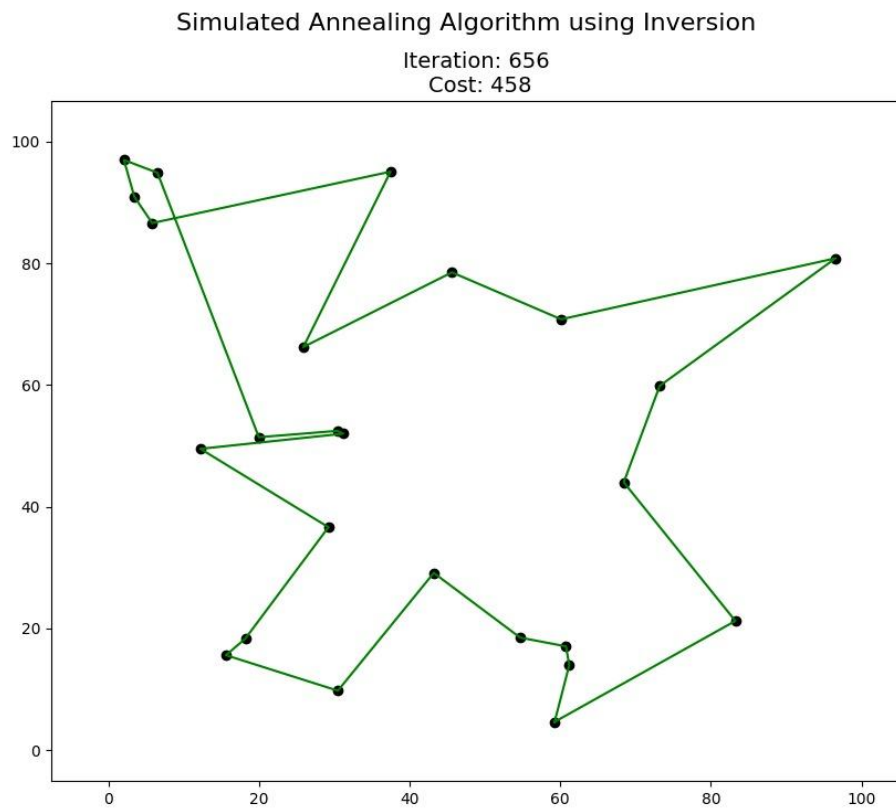


Imagem 4: Configuração 02 - Execução 01

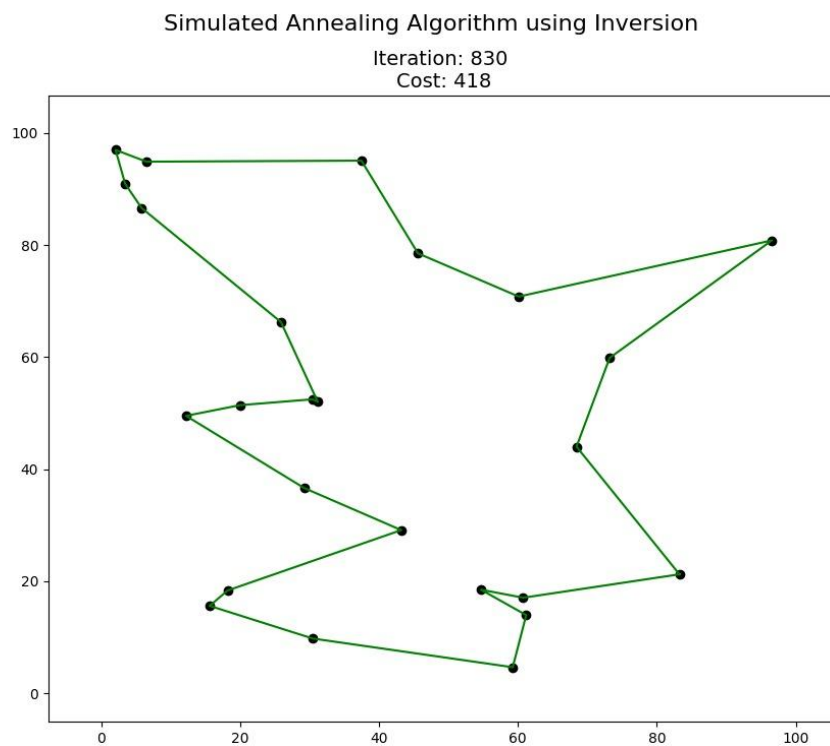


Imagem 5: Configuração 02 - Execução 02

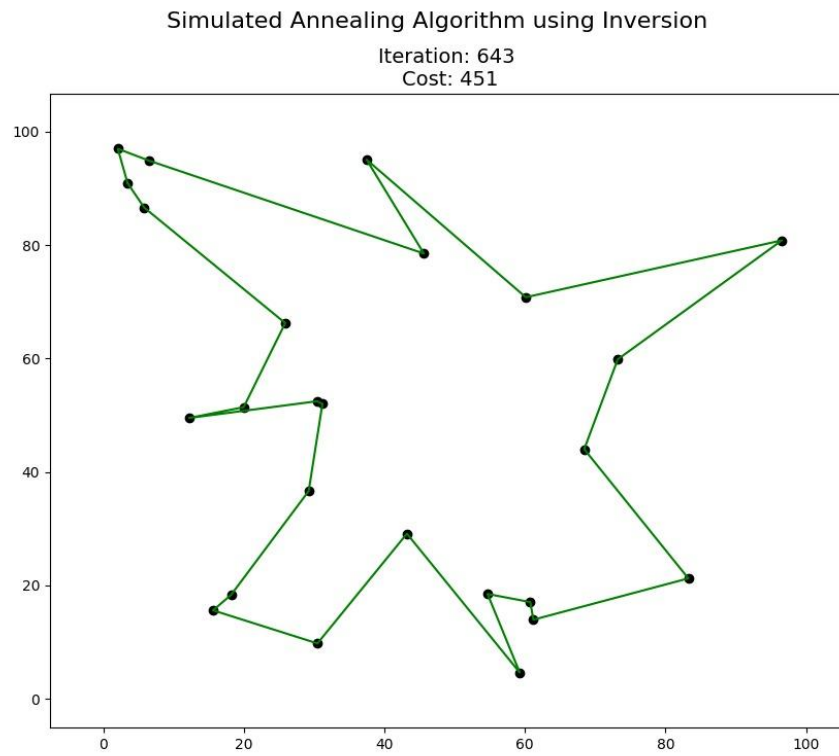


Imagem 6: Configuração 02 - Execução 03