

# TP2 - Ant Colony Optimization (ACO)

## Longest Path

Yuri Niitsuma

### I. INTRODUÇÃO

O objetivo desse trabalho é desenvolver conceitos chaves para construções de soluções para problemas usando Ant Colony Optimization (ACO), envolvendo o entendimento e a implementação dos componentes básicos de um arcabouço de ACO, bem como a análise de sensibilidade dos seus parâmetros (como eles afetam o resultado final, a natureza de convergência, etc) e procedimentos para avaliação das soluções alcançadas. Para esse trabalho, vocês devem elaborar soluções para o problema conhecido como **longest path problem**.

Dado um grafo  $G(V, E)$ , uma função  $w : E \rightarrow R$  que atribui pesos a cada aresta e dois vértices  $u, v \in V$ , denotaremos como  $\mathcal{P}$  o conjunto de caminhos simples partindo de  $u$  e chegando em  $v$ . O problema consiste então em encontrar  $P^* = \{e_1^*, e_2^*, \dots, e_k^*\}$  tal que

$$P^* = \arg \max_{P \in \mathcal{P}} \sum_{e_i \in P} w(e_i)$$

Ou seja, queremos encontrar o caminho simples de  $u$  a  $v$  que maximize o peso total do caminho.

### II. IMPLEMENTAÇÃO

O ACO foi modelado utilizando as principais bibliotecas:

- **numpy**
- **networkx**: biblioteca para grafos

As bibliotecas instaladas, via **pip**, pelo **virtual-env** encontra-se no arquivo **req.txt**.

Possui características de um ACO clássico do problema do caixeiro viajante. A única diferença ao modelo deste trabalho é a atratividade do caminho  $\eta_{xy}$  e taxa de depósito de feromônio nas arestas  $\Delta\tau_{xy}^k$  que serão descritos nas seções a seguir.

#### A. Grafo

Seja o grafo  $G(V, E)$ , definimos baseado na especificação.

- O grafo  $G$  é direcionado.
- Cada aresta do grafo contém, como atributo, pesos dados como entrada, a taxa de feromônio  $t_{xy}$  e a probabilidade da formiga escolher a aresta  $p_{xy}$  que é atualizado no final de cada iteração. Sendo  $xy$  a direção da aresta do nó  $x$  a  $y$ .
- A taxa de feromônio inicial das arestas é 0.1.
- Como o custo de encontrar um caminho entre os nós  $u$  e  $v$  é caro, assumimos que sempre existe um caminho simples do nó 1 ao  $n$ . Caso contrário, entrará em loop infinito. Isto não acontece com os **datasets graph1, graph2 e graph3**.
- O algoritmo não gera soluções inválidas, se o caminho gerado for inválido, a formiga é descartada e gerada novamente.

#### B. Elitismo

Foi utilizado um parâmetro de elitismo que mantém os  $k$  melhores formigas/caminhos para a próxima iteração.

#### C. Escolha da Aresta

A formiga escolhe a aresta  $xy$  pela seguinte fórmula de probabilidade.

$$p_{xy} = \frac{(\tau_{xy})^\alpha (\eta_{xy})^\beta}{\sum_{e_{xj} \in E} (\tau_{xj})^\alpha (\eta_{xj})^\beta}$$

O denominador representa a soma entre todas as arestas vizinhas e  $x$ .

O termo  $\eta_{xy}$  representa a atratividade do caminho e foi definido no modelo como  $\eta_{xy} = w(x, y)$  em que  $w$  é peso da aresta do nó  $x$  para  $y$ .

#### D. Fitness

A fitness da formiga é o custo do caminho do vértice 1 a  $n$ .

$$fit(P^k) = \sum_i w(e_i^k)$$

#### E. Atualização Feromônio

A atualização do feromônio nas arestas ocorre após a construção do grafo ou ao final de cada iteração.

Dado que:

- $\tau_{xy}$  é a taxa de feromônio depositado na aresta
- $\sigma$  é a taxa de evaporação do feromônio ( $0 < \sigma < 1$ ).
- $\Delta\tau_{xy}^k$  é a quantidade de feromônio que a formiga  $k$  irá depositar na aresta  $xy$ .

A taxa de depósito é:

$$\Delta\tau_{xy}^k = \frac{fit(P^k) - fit(\{e_1^k, e_2^k, \dots, e_{xy}^k\}) + 0.5}{\max(fit(P^k); \text{for all ants})}$$

Assim o feromônio é mais forte no início do caminho e fica mais fraco ao final. Isto ajuda a criar mais caminhos com o prefixo do melhor caminho e tentar variar nas arestas posteriores. Também sofre uma normalização dividindo pela fitness máximo entre todas as formigas, assim penalizando as formigas com fitness mais baixas.

A atualização da taxa de feromônio é dada por:

$$\tau_{xy} \leftarrow \min \left( 100, (1 - \sigma)\tau_{xy} + \sum_k \Delta\tau_{xy}^k \right)$$

Observe que há um teto de valor 100 para evitar que o valor exploda e que mantenha uma certa variabilidade dos caminhos.

### III. EXPERIMENTOS

#### *Variação da quantidade de formigas*

O código foi executado com diferentes parâmetros para **graph1** e **graph2**. Foram eles:

- $n^{\circ}$  de formigas =  $\{50, 100, 200, 1000\}$
- iterações =  $\{100\}$
- $\sigma = \{0.1, 0.3, 0.5, 0.7, 0.9\}$
- $\alpha = \{0.1, 0.5, 1.0, 2.0\}$
- $\beta = \{0.1, 0.5, 1.0, 2.0\}$
- elitismo =  $\{0, 1, 5, 10, 15\}$

Todos os dados dos experimentos estão disponíveis no:

<https://homepages.dcc.ufmg.br/~yuriniitsuma/natcomp/>

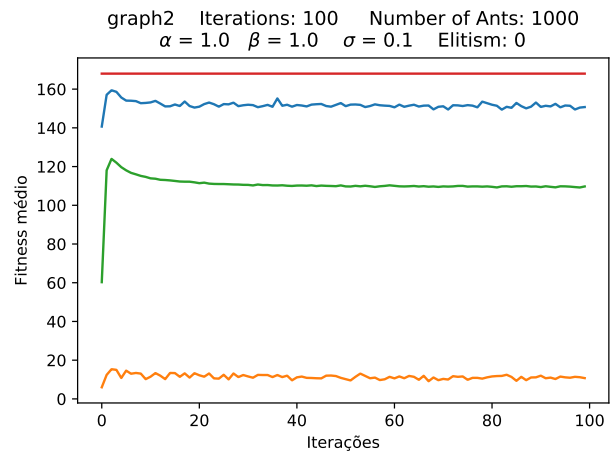
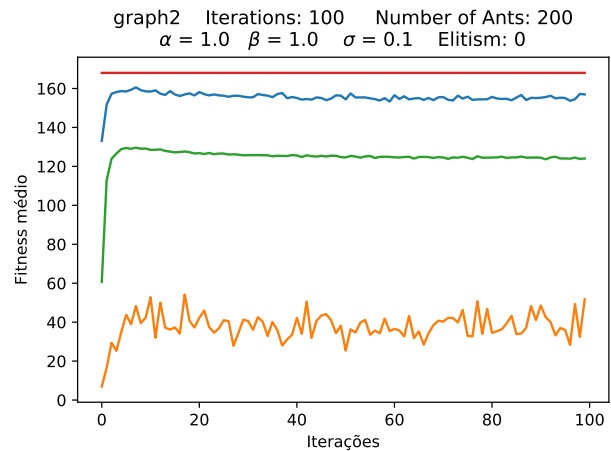
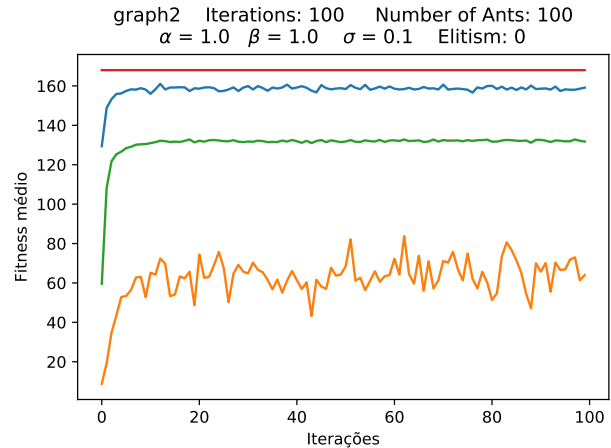
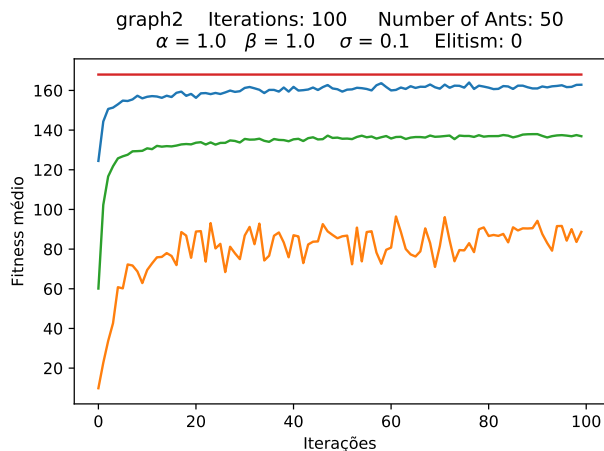
- **out.tar.gz** contendo as saídas com cada combinação de parâmetros.
- **All\_plots.pdf** contendo todos os gráficos (3252) de todos os experimentos. Cada gráfico é a média das saídas de cada combinação dos parâmetros.

No gráfico cada representa:

- **Azul:** melhor fitness
- **Verde:** média das fitness
- **Laranja:** pior fitness
- **Vermelha:** solução ótima

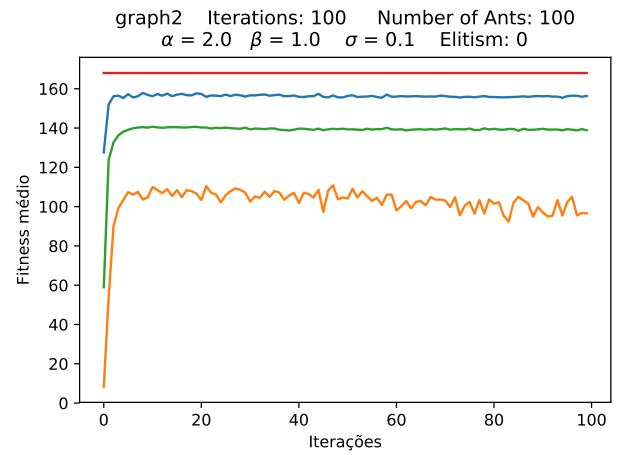
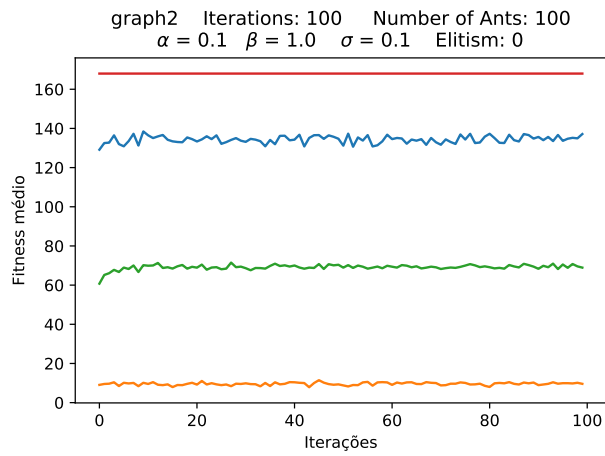
Destacarei os gráficos que representam as análises mais significantes do experimento.

#### A. *graph2*



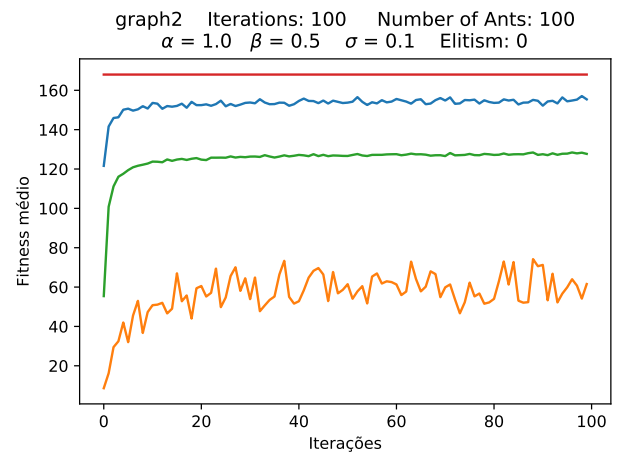
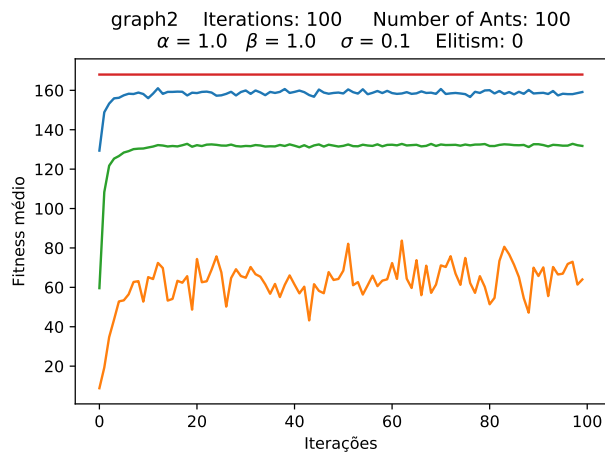
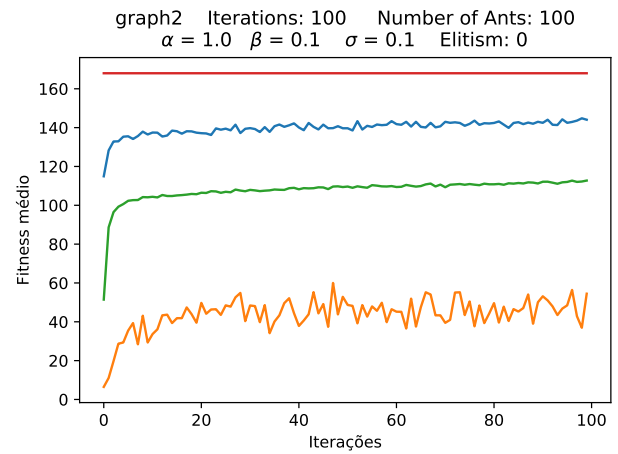
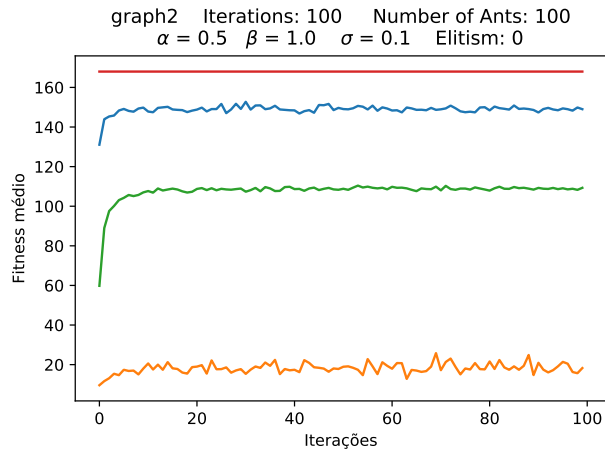
Percebe-se um efeito colateral dado a quantidade de formigas excessivas, pois os caminhos são aleatórios tendo uma pequena chance da convergência ser muito lenta. Manteremos 100 formigas e analisaremos a variação de  $\alpha$ .

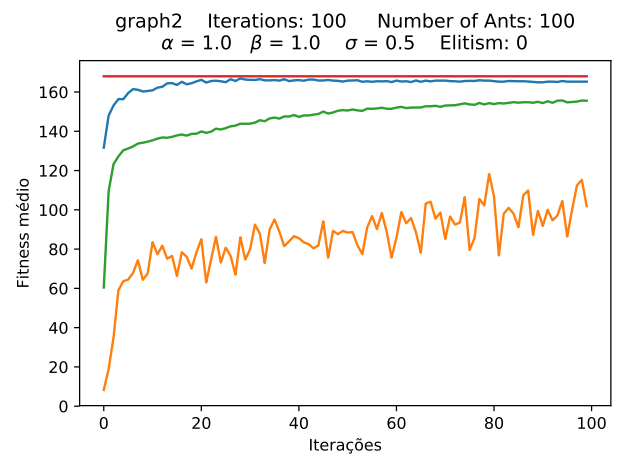
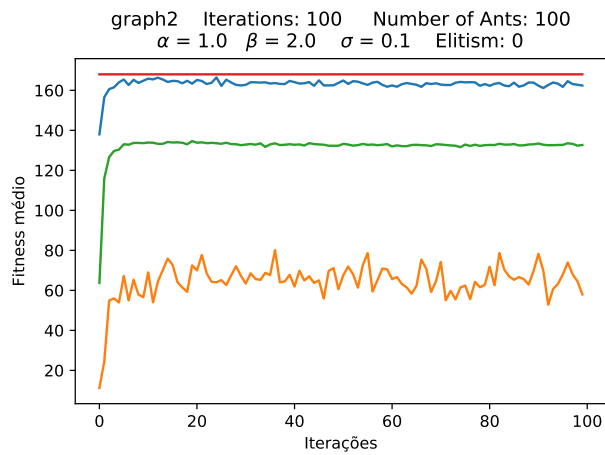
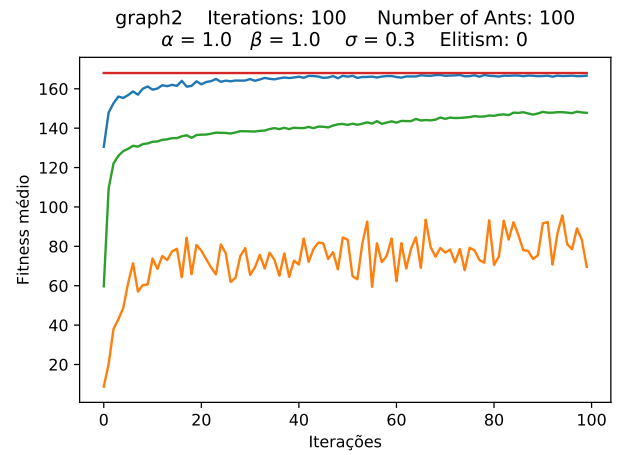
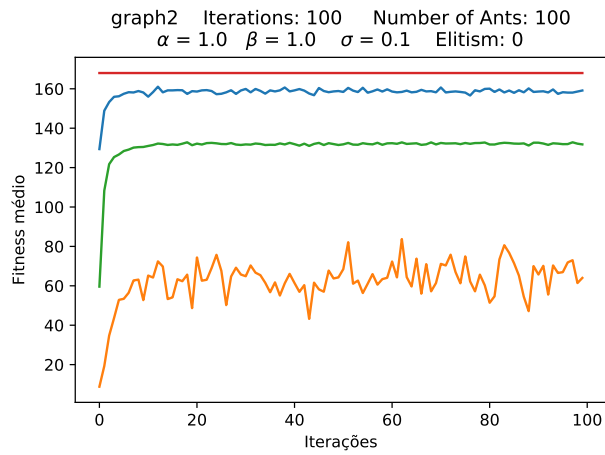
### Variação de $\alpha$



Verificamos que aumenta a taxa de convergência, mas um valor maior aumenta a convergência prematura.

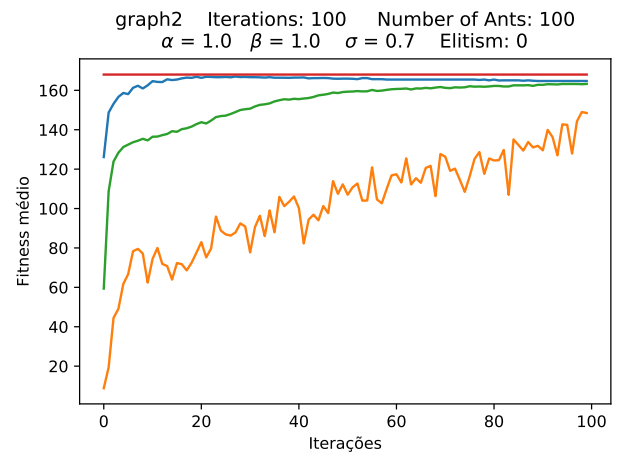
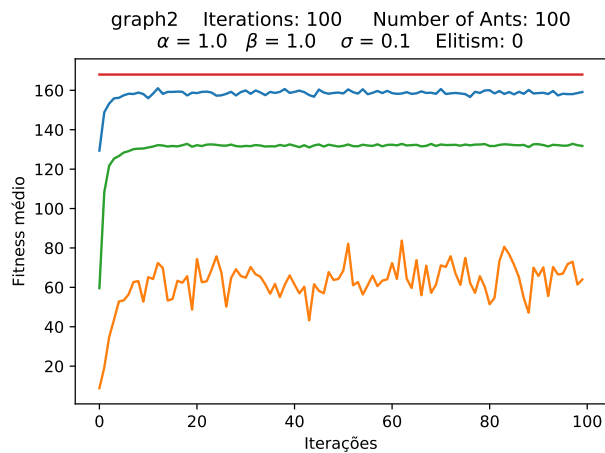
### Variação de $\beta$

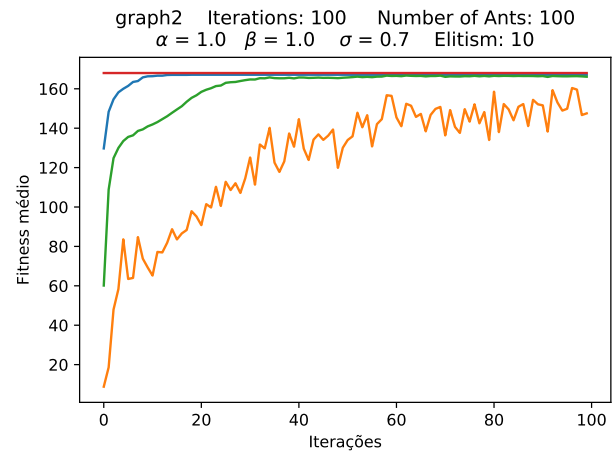
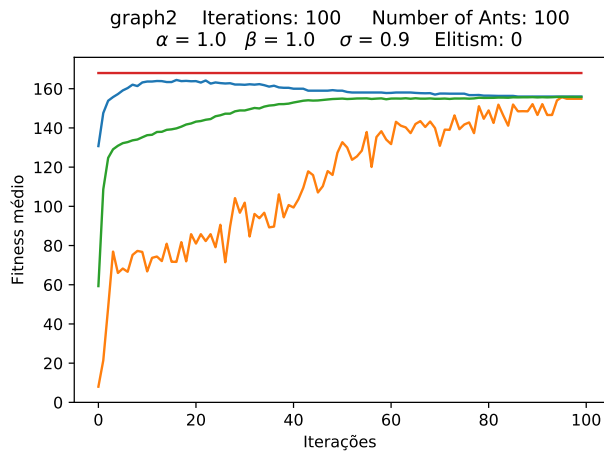




Tem efeitos mais significativos mas não foi encontrado ainda um efeito colateral. Vamos agora variar a evaporação.

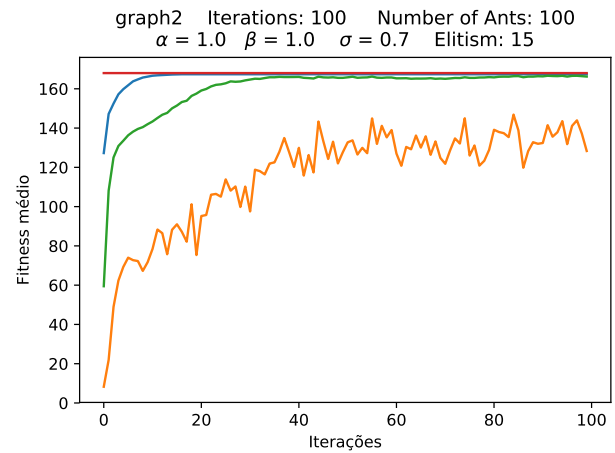
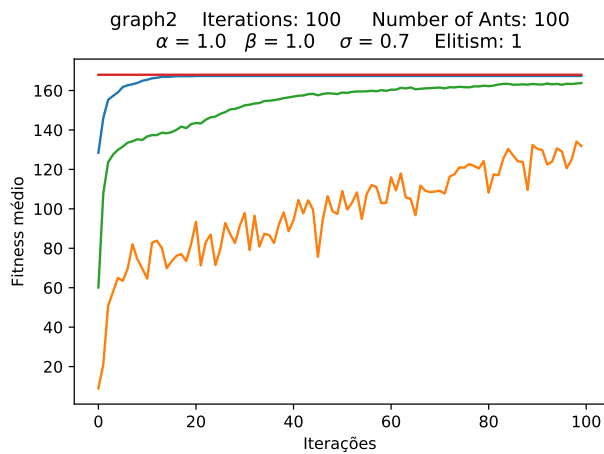
### *Variação de $\sigma$*





Aqui percebe-se um bom ganho das soluções mas que valores muito alto possui efeito colateral que forçam as melhores soluções e a pior convergirem para a média fazendo um efeito sanduiche. Agora vamos analisar o efeito do parâmetro de elitismo.

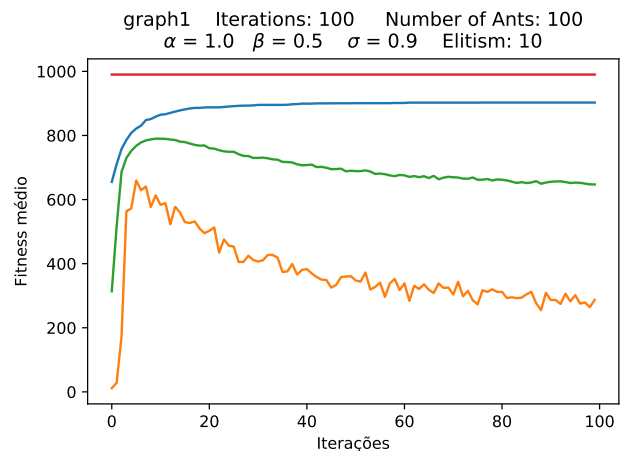
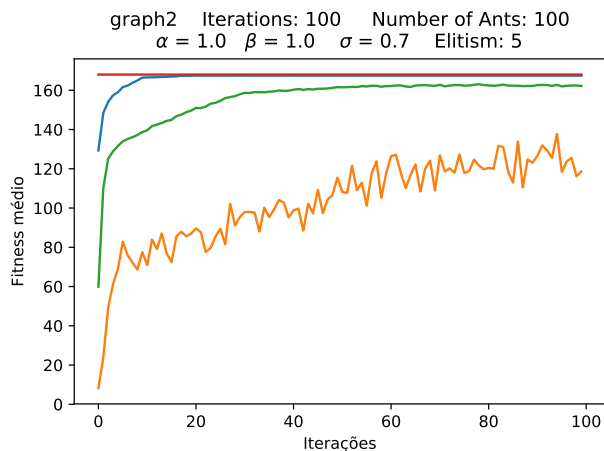
#### *Variação do parâmetro de elitismo*



Percebe-se que a melhor solução converge junto com a média mas ainda mantém alguma variabilidade.

#### **B. graph1**

No *graph1* percebemos que o parâmetro  $\sigma$  é o mais significativo, talvez por alguma característica no grafo.



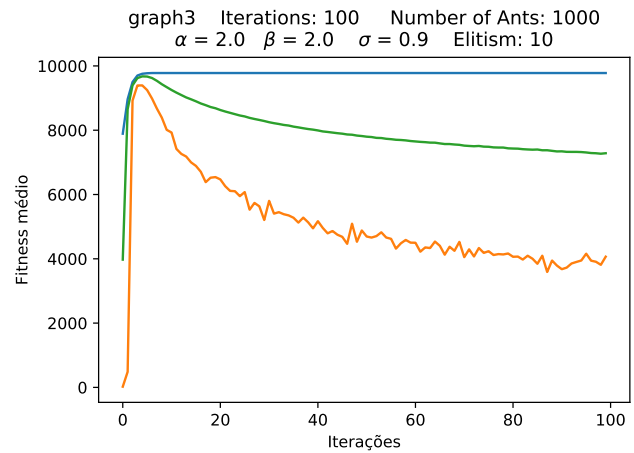
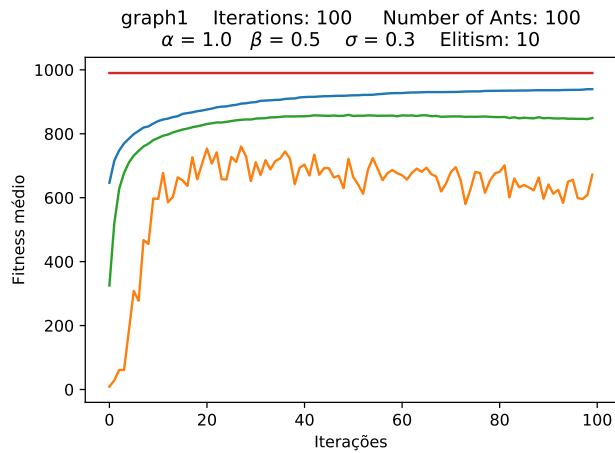


Fig. 2. 9778.75

Valores muito altos de evaporação ocasionam numa maior variabilidade talvez pela quantidade alta de arestas afetam algum efeito de inanição e a aresta acaba sendo inutilizado.

### C. graph3

No **graph3**, os parâmetros que influenciaram na melhor solução foi  $\beta$  e a quantidade máxima de formigas pelo fato do grafo ser mais esperso que os dois anteriores.

As melhores soluções, da melhor formiga, foram:

{9786.16, 9778.75, 9773.58}

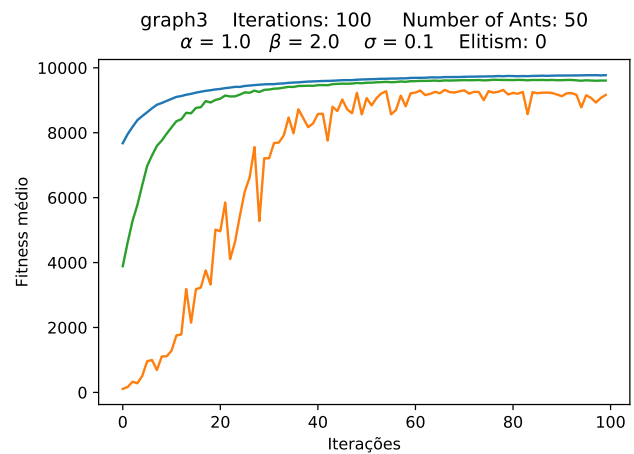


Fig. 3. 9773.58

## IV. CONCLUSÃO

Este trabalho mostrou as capacidades do algoritmo heurístico ACO para encontrar boas soluções do problema *longest path problem*. Mostrou-se também o impacto de alguns dos seus parâmetros no algoritmo e as características de seus grafos.

## V. REFERÊNCIAS

Ant colony optimization: a new meta-heuristic *Dorigo, Marco and Di Caro, Gianni*

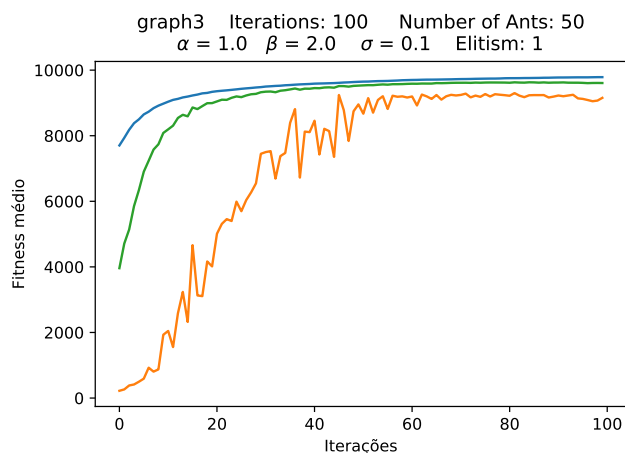


Fig. 1. 9786.16