

TP2 - Ant Colony Optimization

Felipe Cadar Chamone

Outubro 2020

1 Introdução

Este trabalho se trata da implementação e análise de sensibilidade de parâmetros do algoritmo de otimização usando colônia de formigas (*Ant Colony Optimization* - ACO). O trabalho foi implementado em CUDA, linguagem de programação de GPU da NVIDIA, e portanto necessita de uma placa de vídeo compatível para funcionar, como esse pode não ser o caso sempre, também foi implementada uma versão em Python e também foi disponibilizado um Colab Notebook [1] que consegue compilar e rodar a implementação em CUDA. O repositório no GitHub [2] também pode ser interessante de visitar por possuir um GIF onde é possível visualizar a distribuição de feromônios ao longo das iterações.

2 Modelagem

O problema proposto foi o *longest path problem*, que envolve achar o maior caminho de um grafo que não repete vértices. Isso significa que qualquer vértice do grafo foi ser o início ou o fim do caminho. Inicialmente decidi iterar sobre os nós do grafo de modo a rodar o algoritmo começando de cada um deles, isso implica que grafos muito grandes impossibilitam uma experimentação adequada. Para resolver isso pesquisei alguns artigos da literatura que também usam uma implementação em GPU. A solução encontrada, inspirada no artigos de Dawson and Stewart [3] e Wei2013 [4], foi iniciar cada formiga de um nó aleatório em cada iteração.

2.1 Soluções

Para representar um caminho usamos apenas uma lista com os índices dos nós pertencentes ao trajeto na ordem em que foram visitados. Como uma formiga nunca repete um nó no mesmo caminho e os caminhos podem começar e terminar em qualquer nó, o algoritmo nunca gera soluções inválidas.

2.2 *Fitness*

O objetivo do problema é achar o maior caminho dentro do grafo, então a *fitness* é simplesmente a soma dos pesos das arestas pertencentes ao caminho. Esse valor é usado para variar a quantidade de feromônio deixado por cada formiga em cada iteração.

Mais especificamente, depois que todas as formigas encontram uma solução, cada um aplica nas arestas pertencentes à sua solução uma quantidade de feromônio igual ao peso total do caminho encontrado.

2.3 Detalhes de implementação

O programa principal foi implementado em CUDA, isso significa que temos um enorme potencial de paralelismo de tarefas simples, o que é bem conveniente para esse algoritmo, já que, individualmente, o trabalho de cada formiga é bem simples e pode ser computado em paralelo.

Na implementação inicial, onde algoritmo testava cada nó do grafo como sendo o início do caminho (rodando um ACO para cada nó do grafo), foi usado *OpenMP* e múltiplos *streams* de dados na GPU para processar o máximo de nó iniciais em paralelo, onde cada um desses nós iniciais processa suas formigas em paralelo também. O uso de múltiplas *streams* implica que computações enviadas para *streams* diferentes potencialmente serão executadas em paralelo, aproveitando ao máximo da GPU para a execução do algoritmo. Apesar de efetivo, esse método ocupa muita memória de vídeo, e a quantidade de *streams* está limitada, praticamente, a quantidade de cores disponíveis na CPU. Por esses problemas a implementação não é bem escalável.

Na segunda implementação, onde cada formiga se inicia de um nó aleatório, resolvemos boa parte do problema de tempo e escalabilidade, apesar de precisar de mais formigas para chegar nos mesmos resultados, o que não é um problema pela grande capacidade de paralelismo de tarefas simples da GPU.

Mesmo com todas as vantagens do código em CUDA, ele pode se tornar bem complexo pela indexação não usual de *arrays* enviados à GPU, dificultando a correção do trabalho, por esse motivo também foi implementado um código análogo em Python.

3 Experimentos

Aqui vamos discutir a análise de sensibilidade dos parâmetros do ACO. A seguir listamos os parâmetros e suas configurações padrões.

1. Número de formigas: 100
2. Número de iterações: 100
3. Taxa de evaporação: 0.2
4. Alpha: 1
5. Beta: 1

Para a execução dos experimentos variamos um parâmetro de cada vez enquanto mantemos os outros na configuração padrão. Cada experimento foi executado 30 vezes e tomamos a média dessas 30 execuções para medir os resultados.

3.1 Entrada 1

3.1.1 Número de Formigas

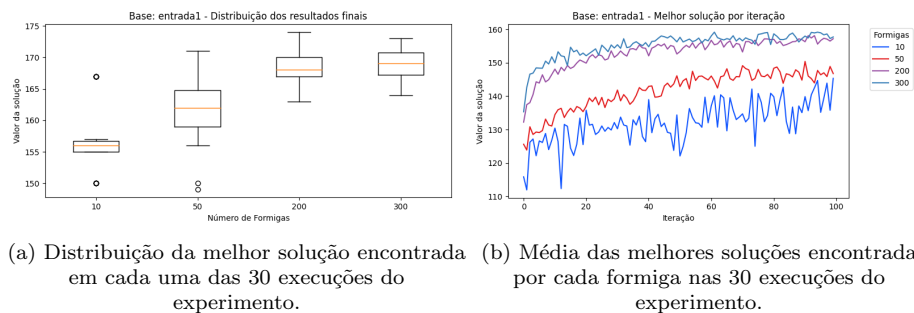


Figure 1: Gráficos variando o número de formigas para a base entrada1

Foram testados os valores [10, 50, 200, 300] para quantidade de formigas. Como podemos ver nos gráficos da figura 1a diferença entre o uso de 200 e 300 formigas é bem pequena, até mesmo no

gráfico (a), de distribuições, as médias e os quartis ficaram bem próximos, por isso foi escolhido o valor 200 para quantidade de formigas.

3.1.2 Número de iterações

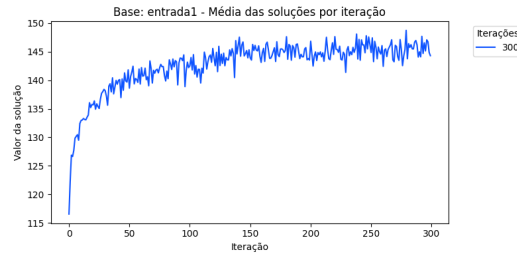
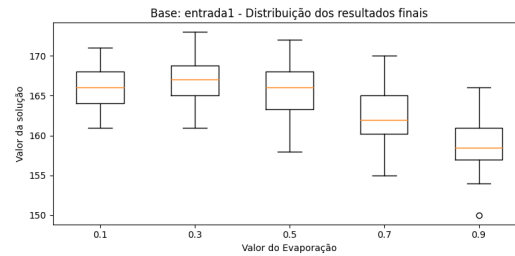


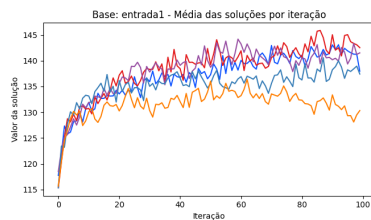
Figure 2: Média das soluções médias encontrada por cada formiga nas 30 execuções do experimento

Foi testado apenas o valor de 300 iterações pois, por se tratar de número de iterações, esse resultado já contém todos os valores anteriores. Como podemos ver no gráfico da figura 2, a curva fica bem lateralizada a partir da iteração de número 100, apresentando uma tendência positiva talvez até o valor 150. Por isso foi escolhido o valor de 150 para quantidade de iterações.

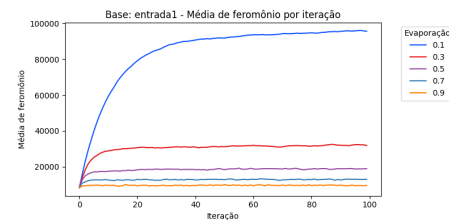
3.1.3 Taxa de Evaporação



(a) Distribuição da melhor solução encontrada em cada uma das 30 execuções do experimento.



(b) Média das melhores soluções encontrada por cada formiga nas 30 execuções do experimento.



(c) Média da quantidade média de feromônio em cada aresta nas 30 execuções do experimento.

Figure 3: Gráficos variando a taxa de evaporação para a base entrada1

Foram testados os valores de $[0.1, 0.3, 0.5, 0.7, 0.9]$ como taxas de evaporação de feromônio. Como podemos observar nos gráficos da figura 3, apesar do gráficos de soluções médias (b) apresentar resultados um pouco confusos, podemos ver no gráfico de distribuições (a) que o valor de 0.3 é o mais indicado para usar como taxa de evaporação da base entrada1. Também podemos observar no gráfico de quantidade média de feromônio (c) que com mesmo com uma valor de evaporação relativamente baixo ocorre uma estabilização da quantidade de feromônio no grafo.

3.1.4 Alpha

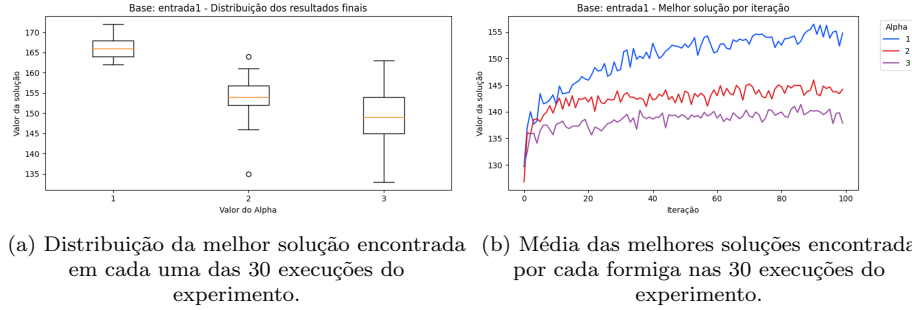


Figure 4: Gráficos variando o parâmetro alpha para a base entrada1

Foram testados valores de alpha [1, 2, 3]. Como podemos ver pelos gráficos da figura 4, o alpha de valor 1 foi melhor. Considerando que nesses experimentos o valor de beta é 1, isso pode significar que selecionar arestas que possuem pesos maiores pode ser uma boa heurística, iremos voltar nesse ponto no experimento do parâmetro beta.

3.1.5 Beta

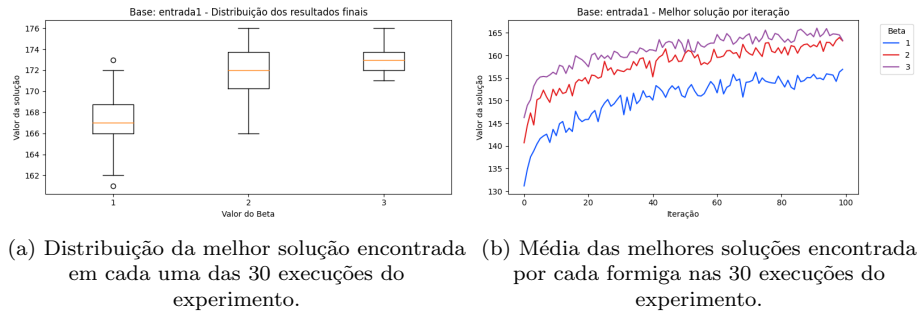


Figure 5: Gráficos variando o parâmetro beta para a base entrada1

Foram testados valores de beta [1, 2, 3]. Como podemos ver pelos gráficos da figura 5, o beta de valor 3 foi melhor, reforçando a hipótese de que o próprio peso das arestas são uma boa indicação de como achar o caminho mais longo do grafo.

3.2 Entrada 2

O grafo da entrada 2 é consideravelmente maior que o anterior, por isso adaptamos um pouco os parâmetros de número de formigas a serem testados.

3.2.1 Número de Formigas

Foram testados os valores [100, 500, 1000, 2000] para quantidade de formigas. Como podemos ver nos gráficos da figura 6, tanto nas distribuições (a) quanto no gráfico de linhas (b), o valor de 2000 mostra uma melhora significativa, por isso ele foi escolhido como número de formigas. Esse parâmetro pode parecer grande, mas a implementação em GPU nos permite usa-lo em tempo hábil.

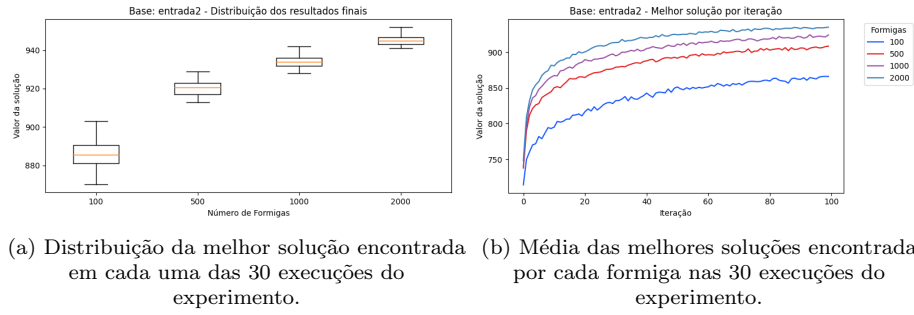


Figure 6: Gráficos variando o número de formigas para a base entrada2

3.2.2 Número de iterações

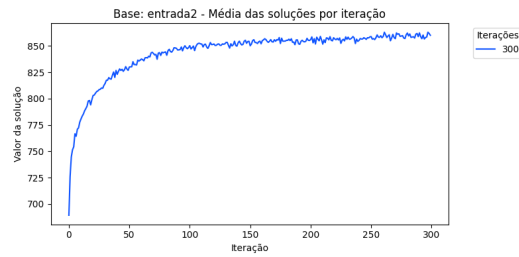
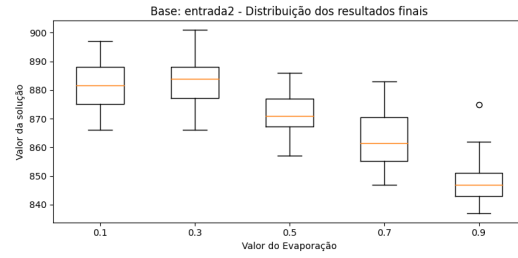


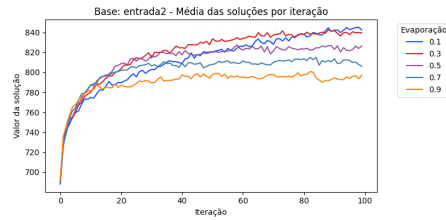
Figure 7: Média das soluções médias encontrada por cada formiga nas 30 execuções do experimento

Assim como na base entrada1, esse parâmetro foi testado apenas o valor de 300. Como podemos ver no gráfico da figura 7, a curva fica bem lateralizada a partir da iteração de número 100, apresentando uma tendência positiva talvez até o valor 150. Por isso foi escolhido o valor de 150 para quantidade de iterações.

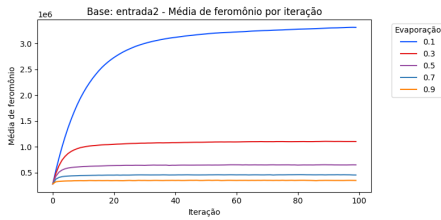
3.2.3 Taxa de Evaporação



(a) Distribuição da melhor solução encontrada em cada uma das 30 execuções do experimento.



(b) Média das melhores soluções encontrada por cada formiga nas 30 execuções do experimento.

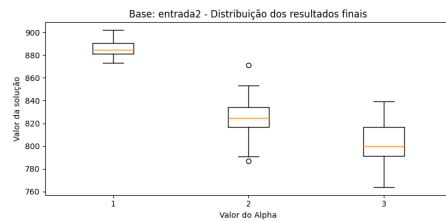


(c) Média da quantidade média de feromônio em cada aresta nas 30 execuções do experimento.

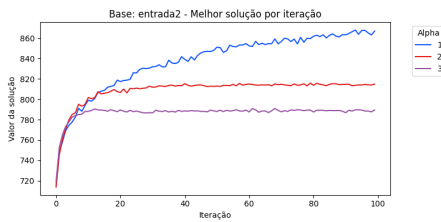
Figure 8: Gráficos variando a taxa de evaporação para a base entrada2

Foram testados os valores de $[0.1, 0.3, 0.5, 0.7, 0.9]$ como taxas de evaporação de feromônio. Como podemos observar nos gráficos da figura 8, apesar do gráficos de soluções médias (b) aproximar bastante as linhas de 0.1 e 0.3, podemos ver no gráfico de distribuições (a) que o valor de 0.3 é o levemente melhor para ser usado como taxa de evaporação da base entrada2. Também podemos observar no gráfico de quantidade média de feromônio (c) que ocorre uma estabilização da quantidade de feromônio no grafo.

3.2.4 Alpha



(a) Distribuição da melhor solução encontrada em cada uma das 30 execuções do experimento.



(b) Média das melhores soluções encontrada por cada formiga nas 30 execuções do experimento.

Figure 9: Gráficos variando o parâmetro alpha para a base entrada2

Foram testados valores de alpha $[1, 2, 3]$. Como podemos ver pelos gráficos da figura 9, o alpha de valor 1 foi melhor. Considerando que nesses experimentos o valor de beta é 1, isso pode significar que selecionar arestas que possuem pesos maiores pode ser uma boa heurística para esse grafo, assim como aconteceu na entrada1.

3.2.5 Beta

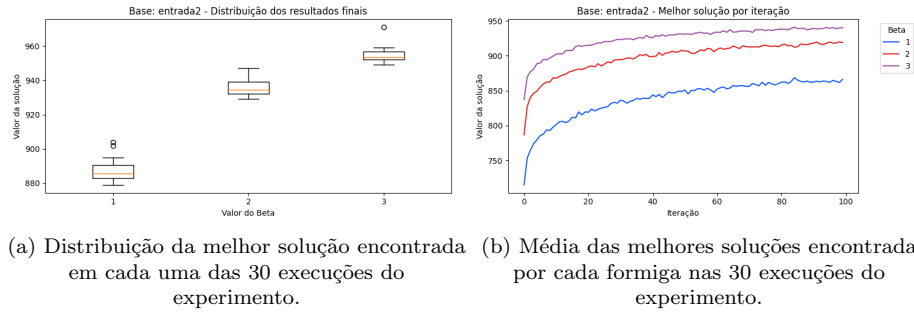


Figure 10: Gráficos variando o parâmetro beta para a base entrada2

Foram testados valores de beta $[1, 2, 3]$. Como podemos ver pelos gráficos da figura 10, o beta de valor 3 foi melhor, reforçando a hipótese de que o próprio peso das arestas são uma boa indicação de como achar o caminho mais longo do grafo para a base entrada2.

3.3 Entrada 3

Assim como o grafo da entrada2, o grafo da entrada 3 é consideravelmente maior que o da entrada1 e por isso adaptamos os parâmetros de quantidade de formigas.

3.3.1 Número de Formigas

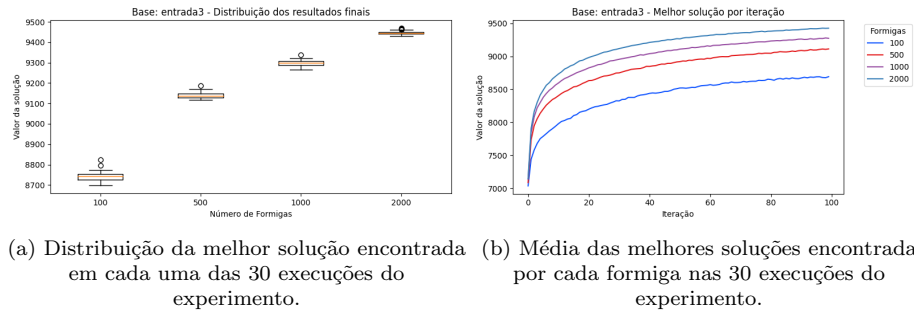


Figure 11: Gráficos variando o número de formigas para a base entrada3

Foram testados os valores $[100, 500, 1000, 2000]$ para quantidade de formigas. Como podemos ver nos gráficos da figura 11, tanto nas distribuições (a) quanto no gráfico de linhas (b), o valor de 2000 mostra uma melhora significativa, por isso ele foi escolhido como número de formigas.

3.3.2 Número de iterações

Assim como nas bases entrada1 e entrada2, esse parâmetro foi testado apenas o valor de 300. Como podemos ver no gráfico da figura 12, a curva fica bem lateralizada a partir da iteração de número 100, apresentando uma tendência positiva talvez até o valor 150. Por isso foi escolhido o valor de 150 para quantidade de iterações.

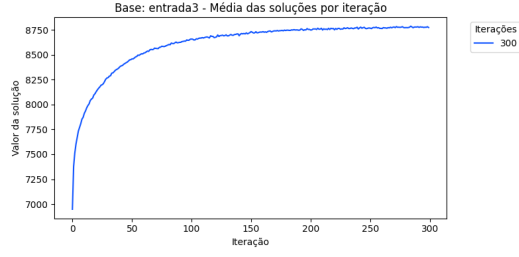
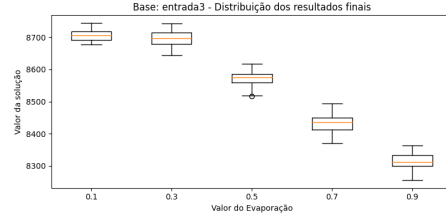
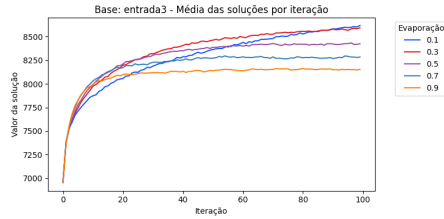


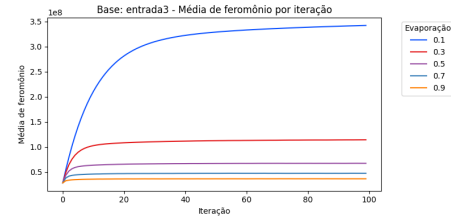
Figure 12: Média das soluções médias encontrada por cada formiga nas 30 execuções do experimento



(a) Distribuição da melhor solução encontrada em cada uma das 30 execuções do experimento.



(b) Média das melhores soluções encontrada por cada formiga nas 30 execuções do experimento.



(c) Média da quantidade média de feromônio em cada aresta nas 30 execuções do experimento.

Figure 13: Gráficos variando a taxa de evaporação para a base entrada3

3.3.3 Taxa de Evaporação

Foram testados os valores de $[0.1, 0.3, 0.5, 0.7, 0.9]$ como taxas de evaporação de feromônio. Como podemos observar nos gráficos da figura 13, apesar do gráficos de soluções médias (b) e o gráfico de distribuições (a) aproximarem bastante os resultados de 0.1 e 0.3, podemos ver que quantidade média de feromônio (c) do valor de 0.1 cresceu muito mais que as outras, podendo causar um convergência muito prematura e mantendo máximos locais no *landscape* da função de fitness, por isso o valor de 0.3 foi escolhido.

3.3.4 Alpha

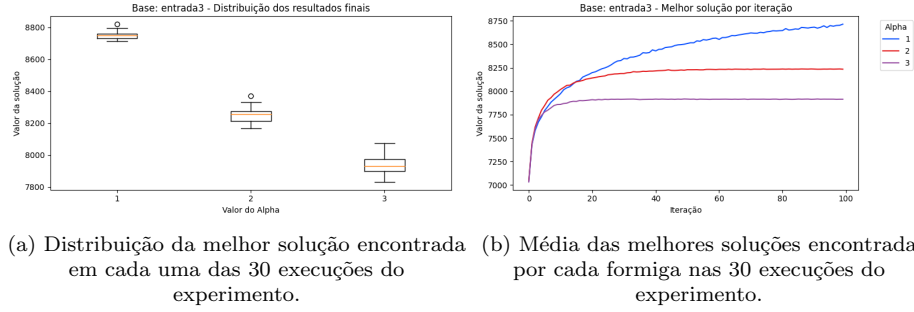


Figure 14: Gráficos variando o parâmetro alpha para a base entrada3

Foram testados valores de alpha $[1, 2, 3]$. Como podemos ver pelos gráficos da figura 14, o alpha de valor 1 foi o melhor e portanto foi escolhido.

3.3.5 Beta

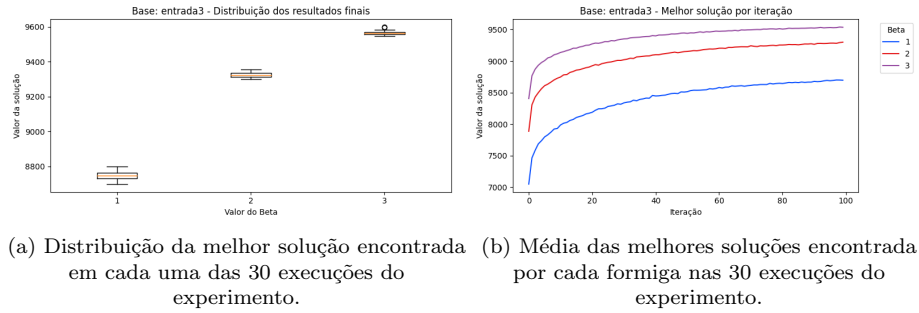


Figure 15: Gráficos variando o parâmetro beta para a base entrada3

Foram testados valores de beta $[1, 2, 3]$. Como podemos ver pelos gráficos da figura 15, o beta de valor 3 foi melhor, e portanto foi escolhido.

3.4 Resumo dos experimentos

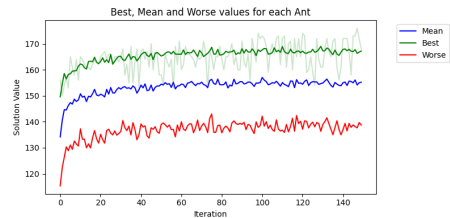
A tabela 1 mostra o resumo das escolhas dos parâmetros para as 3 bases.

	entrada1	entrada2	entrada3
Formigas	200	2000	2000
Iterações	150	150	150
Evaporação	0.3	0.3	0.3
Alpha	1	1	1
Beta	3	3	3

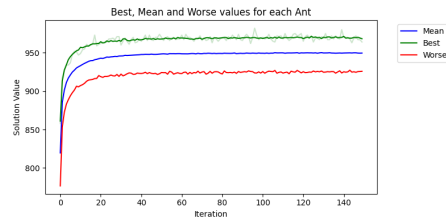
Table 1: Recapitulação dos experimentos.

4 Resultados

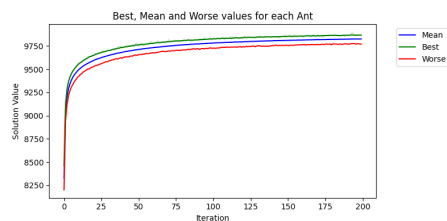
Após realizar a análise de sensibilidade dos parâmetros executamos cada o ACO com os parâmetros otimizados 30 vezes em cada base, obtendo os resultados apresentados nos gráficos da figura 16. Nos gráficos, a linha verde clara representa a execução que gerou o melhor resultado. Os melhores resultados para as bases entrada1, entrada2 e entrada3 foram, 176, 982 e 9899, respectivamente.



(a) Média dos valores máximos, médios e mínimos das 30 execuções do ACO na base entrada1. Valor máximo: 176



(b) Média dos valores máximos, médios e mínimos das 30 execuções do ACO na base entrada2. Valor máximo: 982



(c) Média dos valores máximos, médios e mínimos das 30 execuções do ACO na base entrada3. Valor máximo: 9899

Figure 16: Gráficos variando a taxa de evaporação para a base entrada3

References

- [1] Felipe Cadar Chamone. Colabexample.ipynb - colaboratory. <https://colab.research.google.com/github/felipecadar/AntColonyOptimization-CUDA/blob/main/ColabExample.ipynb>, . (Accessed on 10/17/2020).
- [2] Felipe Cadar Chamone. felipecadar/antcolonyoptimization-cuda: Antcolonyoptimization in cuda. <https://github.com/felipecadar/AntColonyOptimization-CUDA>, . (Accessed on 10/17/2020).
- [3] Laurence Dawson and Iain Stewart. Improving ant colony optimization performance on the GPU using CUDA. In *2013 IEEE Congress on Evolutionary Computation*. IEEE, June 2013. doi: 10.1109/cec.2013.6557791. URL <https://doi.org/10.1109/cec.2013.6557791>.
- [4] Kai-Cheng Wei, Chao-Chin Wu, and Chien-Ju Wu. Using CUDA GPU to accelerate the ant colony optimization algorithm. In *2013 International Conference on Parallel and Distributed Computing, Applications and Technologies*. IEEE, December 2013. doi: 10.1109/pdcat.2013.21. URL <https://doi.org/10.1109/pdcat.2013.21>.