

Avaliação de Algoritmo Colônia de Formiga para o Problema do Caixeiro Viajante

Lucas Rocha

9 de junho de 2025

Resumo do Processo de Avaliação

Este relatório apresenta uma análise realizada sobre o Algoritmo de Colônia de Formigas para a solução do problema do Caixeiro Viajante (TSP). Foram utilizadas duas instâncias distintas, ambas descrevendo a estrutura do problema em questão. A primeira instância é a **LAU15**. A partir dela, foi possível obter uma matriz 15×15 que representa as ligações entre as 15 cidades e seus respectivos pesos. A outra instância foi a **SGB128**, que possui 128 cidades. Também foi utilizada uma matriz, neste caso 128×128 , para representar a estrutura do grafo com as ligações entre as cidades.

Para a averiguação da eficiência do algoritmo em questão, foi realizado um teste fatorial completo, testando combinações de parâmetros, com o objetivo de encontrar um conjunto de valores que permitisse extrair o melhor desempenho possível do algoritmo estudado neste trabalho. Os parâmetros utilizados para o teste foram: `evaporationRates`, `maxIterationsSet`, `alphas`, `betas`, `Qs`, `epsilons` e `choiceMethod`.

Foram implementados três métodos de distribuição de feromônios após os percursos das formigas. O primeiro deles distribui feromônios para todos os caminhos percorridos por todas as formigas. O segundo, semelhante ao primeiro, também distribui feromônios dessa forma, mas com um valor adicional: uma pequena taxa contribuidora da melhor formiga naquela iteração, dada pela multiplicação do parâmetro `epsilon` com o valor contribuidor da melhor formiga. Por fim, o terceiro método distribui feromônios apenas para a melhor formiga.

Os conjuntos de parâmetros testados foram:

- `evaporationRates` = {0.1, 0.3, 0.5};
- `maxIterationsSet` = {50, 100, 300};
- `alphas` = {1.0, 2.5};
- `betas` = {1, 2.0, 5.0};
- `Qs` = {10, 100};
- `epsilons` = {3, 5, 10};
- `choiceMethod` = {1, 2, 3};

Não foi possível realizar o teste fatorial para a instância com 128 cidades, pois o código demandava um tempo de execução muito alto, o que obrigaria a realização de um teste fatorial ineficiente na tentativa de rodar em um tempo adequado. Por isso, o teste fatorial foi realizado apenas na instância com 15 cidades.

Os parâmetros considerados “melhores” nesse teste foram utilizados como base para a execução da análise de convergência, a qual, sim, foi realizada para ambas as instâncias.

Resultados e Observações

Para a instância com 15 cidades, o valor ótimo é conhecido e corresponde a 291. A melhor combinação de parâmetros para alcançar esse valor foi:

- `evaporationRate`: 0,3
- `maxIterations`: 50
- `alpha`: 2,5
- `beta`: 5,0
- `epsilon`: 10,0
- `Q`: 10,0
- `method`: 3
- `averageFitness`: 291,0

Além disso segue a abaixo as 5 melhores execução do teste fatorial.

Tabela 1: Top 5 Combinações de Parâmetros para TSP (LAU15)

evaporationRates	maxIteration	alpha	beta	Q	epsilon	method	AverageFitness
0.3	50	2.5	5.0	10.0	10.0	3	291.0
0.3	100	1.0	5.0	10.0	10.0	1	291.0
0.3	100	1.0	5.0	10.0	10.0	3	291.0
0.3	100	1.0	5.0	100.0	3.0	1	291.0
0.3	100	1.0	5.0	100.0	3.0	2	291.0

É perceptível que mesmo ocorrendo um empate entre as 5 melhores combinações, a primeira se torna uma escolha mais viável por possuir o parâmetros `maxIterations` menor. Com isso, menos execuções e mais eficiência.

O método 3, que utiliza a distribuição de feromônios apenas na melhor formiga da iteração, juntamente com o método 1, que distribui igualmente para todas, demonstraram melhor desempenho. Já o método 2, apesar de aparecer entre os cinco melhores, deixa a desejar em uma análise mais global do restante dos resultados obtidos no arquivo de teste.

Análise de Convergência

Uma vez obtida a combinação ótima dos parâmetros, foi possível realizar execuções que permitiram analisar a convergência do algoritmo para o valor ótimo. Como discutido anteriormente, nesta etapa analisamos o desempenho de convergência do algoritmo para as duas instâncias: a de 15 cidades e a de 128 cidades, com uma pequena alteração na instância SGB128.

A instância SGB128, por possuir um número elevado de cidades, não permitiu a realização de um teste fatorial apropriado. Sendo assim, utilizou-se o conjunto de parâmetros obtido no teste fatorial da instância LAU15 como base. Após algumas execuções de teste, foi possível perceber que o valor de `maxIteration` estava muito baixo. Por isso, para a instância SGB128, foi utilizado `maxIteration` igual a 100.

Utilizou-se 20 execuções em ambas as instâncias. A cada execução, foi salvo um vetor de fitness, de modo que cada item no índice *i* desse vetor representava o valor da melhor formiga na iteração *i*.

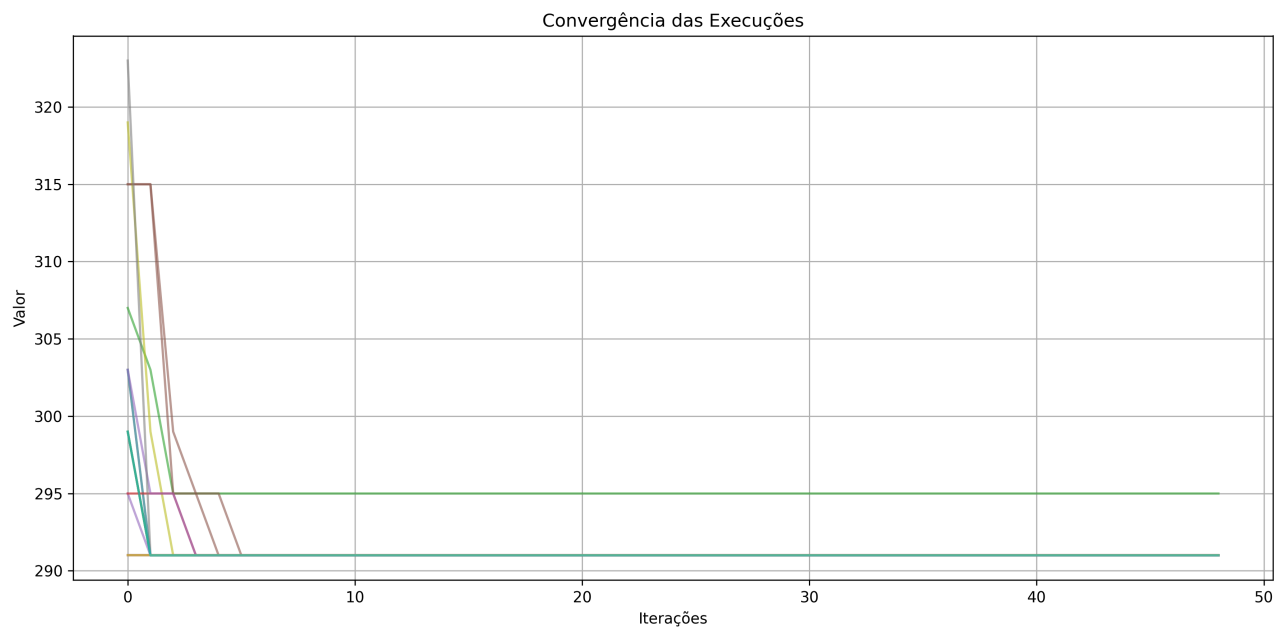


Figura 1: Distribuição dos custos em 20 execuções (**LAU15**)

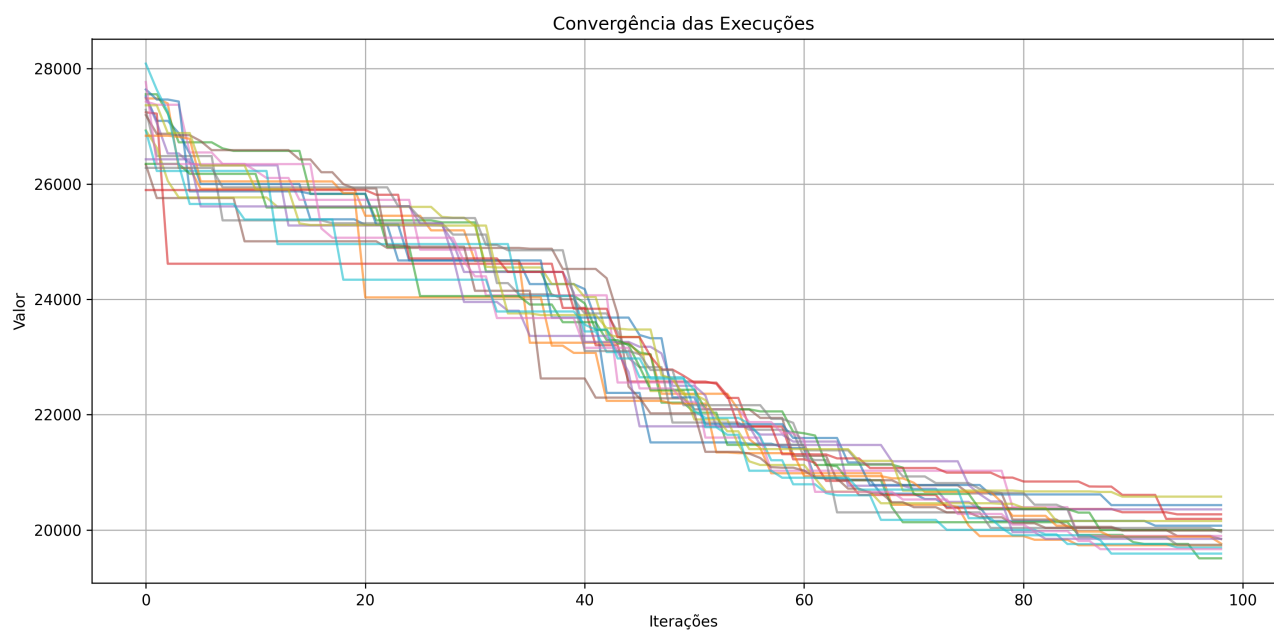


Figura 2: Distribuição dos custos em 20 execuções (**SGB128**)

Discussão

Nota-se a diferença de convergência ao se comparar as duas instâncias. Por LAU15 possuir poucas cidades, o algoritmo rapidamente alcança o valor ótimo — que, neste exemplo, é 291. Todas as execuções tenderam ao ótimo, com exceção de uma delas, que encontrou apenas o valor 295, o que demonstra a natureza estocástica do algoritmo.

Já para a instância SGB128, percebe-se que a convergência foi contínua, com o algoritmo reduzindo seu valor a cada execução. Diferentemente da instância anterior, em que o valor ótimo era encontrado logo no início e mantido constante, aqui o comportamento é distinto. Mesmo ao final das execuções, observa-se uma tendência de melhora. Um ponto positivo é que, em algumas execuções, o algoritmo atingiu valores inferiores a 20.000 — o que é significativo, considerando que, com o algoritmo genético, os melhores resultados até então não ultrapassavam os 22.000.

Isso demonstra que o algoritmo de colônia de formigas, para esse problema, possui uma natureza que favorece a busca por soluções mais próximas do ótimo.